Les preuves vues comme des jeux et réciproquement : sémantique dialogique de langages naturel ou logiques

Davide Catta

THÈSE POUR OBTENIR LE GRADE DE DOCTEUR
DE L'UNIVERSITE DE MONTPELLIER

En informatique

École doctorale : Information, Structures, Systèmes

Unité de recherche UMR5506

Les preuves vues comme des jeux et réciproquement : sémantique dialogique de langages naturel ou logique

Présentée par Davide Catta

Le 23 Novembre 2021

Sous la direction de Christian Retoré, Richard Moot
et Myriam Quatrini

Devant le jury composé de

Vito Michele Abrusci, Professeur émérite, dipartimento di matematica e fisica	Rapporteur
Andreas Herzig, Directeur de Recherches, CNRS, IRIT, Université Paul Sabatier	Rapporteur
David Delahaye, Professeur, LIRMM, Université de Montpellier	Examinateur
Valeria de Paiva, Chargé de recherce, Topos institute, Berkeley	Examinatrice
Jean Yves Marion, Professeur, ENSMN, LORIA	Examinateur

UNIVERSITÉ
DE MONTPELLIER

REMERCIEMENTS

Je tiens d'abord à remercier mon directeur de thèse Christian Retoré et mes deux encadrants Richard Moot et Myriam Quatrini. Leurs suggestions et encouragements ont été très précieux pendant tout mon travail de conception et rédaction.
Je voudrais en suite remercier les deux examinateurs de mon travail de thèse, Michele Abrusci et Andreas Herzig, qui ont lu mon manuscrit en profondeur et qui m'ont données des conseils très ponctuels pour l'améliorer. Je remercie également les examinateurs et l'examinatrice de mon travail qui m'ont posé des questions très stimulantes et pertinentes lors de ma défense de thèse : David Delahaye, Jean Yves Marion, Alexis Saurin et Valeria de Paiva.
Je remercie les membres de l'équipe TEXTE du LIRMM qui m'ont accueilli pendant les dernières trois ans.
Je remercie les collègues qui ont travaillé avec moi pendant ce trois ans et qui m'ont appris beaucoup : Matteo Acclavio, Lutz Straßburger, Mehdi Mirzapour et Symon Jory Stevens-Guille.
Je remercie chaleureusement toutes les personnes que, pendant ce trois ans, m'ont aidé à garder un contact avec la réalité : les membres du politburo, les membres de LCM, Antonio, Pino, Peppe, Marco, Lorenza, Lucia, Gaetan et ma famille qui m'a toujours soutenu(matérialement et moralement) dans mes études. Enfin, je tiens à remercier du fond du cœur ma compagne Nadège qui m'a supporté tout au long de la préparation de la thèse et de la rédaction du manuscrit. La tâche n'était pas facile et je lui en suis infiniment reconnaissant.

Woof-woof pour Hobey.

Mors clés

Théorie de la preuve, sémantique dialogique, sémantique des jeux, traitement automatique du langage naturel, logique mathématique

Keywords

Proof theory, Dialogical Semantics, Game Semantics, Natural Language Processing, Mathematical Logic

RÉSUMÉ DE LA THÈSE

Notre travail de thèse se situe au carrefour de plusieurs disciplines : d'une part, la logique mathématique et l'informatique théorique, d'autre part le traitement automatique du langage naturel et plus particulièrement la sémantique formelle du langage naturel. Le fil conducteur est la présence constante des méthodes logiques issues de la théorie de la preuve et par le problème philosophique qui a motivé notre thèse : quels sont les liens entre la notion de preuve et celle de signification linguistique ou logique ? Plus concrètement, nous étudions des systèmes formels dont les preuves sont vues comme des stratégies gagnantes pour des jeux à deux joueurs. Dans ces jeux, un jouer, appelé Proposant, essaye de construire une justification pour un certain énoncé tandis que l'autre, l'Opposant, essaye de construire une réfutation de cet énoncé. La thèse est composée de trois parties, chaque partie contenant deux ou trois chapitres.

La première partie est propédeutique. Dans les deux chapitres qui la composent nous présentons les outils mathématiques utilisés dans notre thèse ainsi que les principes logiques et philosophiques qui ont guidés nos travaux, notamment la sémantique inférentialiste.

La deuxième partie de notre thèse contient deux longs chapitres, lesquels présentent les résultats de théorie de la démonstration qui constituent le cœur de notre thèse. En particulier, dans le premier chapitre de cette partie, nous définissons précisément un système de logique dialogique pour la logique classique du premier ordre avec termes. Nous montrons que, pour une formule A, l'existence d'une stratégie gagnante pour A équivaut au fait que A est un théorème logique. Bien que des systèmes de logique dialogique pour la logique classique du premier ordre existent depuis les années 1960 il n'existait pas à ce jour de preuve convaincante publiée de ce résultat, notamment en présence de termes. Dans le deuxième chapitre de cette deuxième partie, nous présentons une sémantique dénotationnelle pour la variante constructive de la logique modale K. En particulier notre sémantique dénotationnelle est une sémantique des jeux dans laquelle les preuves de la logique modale sont interprétées par des stratégies gagnantes pour des jeux à deux jouer. Nous montrons que notre sémantique possède une propriété remarquable : elle est 'pleinement adéquate' (fully complete) c'est-à-dire que toute stratégie gagnante est l'interprétation d'au moins une preuve de la logique modale.

La troisième et dernière partie se compose de trois chapitres, chacun étant consacré à une application de nos travaux en théorie de la démonstration à la sémantique du langage naturel. Dans le premier chapitre, nous étudions le rapport entre les analyses syntaxiques catégorielles d'une même phrase et les représentations sémantiques logiques de la phrase analysée. Nous montrons que, lorsque certaines conditions sont respec-

tées, la fonction qui transforme analyses syntaxiques d'une phrase en représentations sémantique logiques est injective. Dans le deuxième chapitre de cette troisième partie, nous appliquons notre système de logique dialogique à la résolution au problème de la reconnaissance d'inférences en langage naturel en utilisant un analyseur syntaxique et sémantique catégoriel. Dans le dernier chapitre de cette partie, nous présentons un système formel pour la résolution d'anaphore et ellipses, problème généralement abordé par des méthodes de théorie des modèles. Nous, au contraire, présentons une solution basée sur la théorie de la démonstration, en développant un système de logique dialogique qui permet de résoudre simplement les anaphores et les ellipses.

Dans la conclusion, nous faisons le bilan de notre travail de thèse et essayons de décrire les développements futurs possibles de notre recherche, tant du point de vue mathématique et logique que du point de vue des applications au langage naturel.

SUMMARY

This thesis is situated at the intersection of several disciplines: on the one hand, mathematical logic and theoretical computer science, on the other hand, natural language processing and formal semantics of natural language. The thread tying these topics together is the constant use of tools and methodologies of proof theory and the philosophical problem that motivated our thesis: what are the links between the notion of proof and that of linguistic meaning? More concretely, we study formal proofs systems. in these systems proofs are seen as winning strategies for two-player games. In the games one player, called the Proponent, tries to construct a justification for a certain statement while the other, the Opponent, tries to refute this statement. Our thesis is composed of three parts, each part containing a maximum of three chapters.

The first part is preparatory. In the two chapters that compose it we present the mathematical tools used in our thesis as well as the philosophical question that underlie our research.

The second part consists of two long chapters and presents the central proof-theoretical results of our thesis. In the first chapter of this part we present a dialogical logic system for classical first order logic. We show that, given a formula A, A is a logical theorem if and only if there is a proponent winning strategy for A. Dialogical logic systems for classical first-order logic have existed since the 1960's. However there is no convincing proof of this result in the literature. In the second chapter of this second part we present a denotational semantics for the constructive variant of the modal logic K. Our denotational semantics is a game semantics: the proofs of modal logic are interpreted by winning strategies for two-player games. We show that our game semantics has a remarkable property; it is 'fully complete': every winning strategy is the interpretation of a proof of modal logic.

The third and last part of our thesis consists of three chapters. Each chapter is devoted to an application of proof theory to the semantics of natural language. In the first chapter, we study the relationship between the categorical syntactic analyses of a sentence and the logical representations of the sentence. We show that, when certain conditions are met, the function that transforms syntactic analyses of a sentence into logical representations is injective. In the second chapter of this third part, we use our dialogical logic system, together with type logical grammars, to solve textual entailment problems. In the last chapter of this section we present a formal system for the resolution of anaphora and ellipsis. This problem is usually addressed by model-theoretic methods. We, on the contrary, present a solution based on proof theory. We develop a dialogical logic system in which anaphora and ellipsis can be solved in a simple way.

In the conclusion, we sketch possible future developments of our research. Both from a mathematical and logical point of view and from the point of view of natural language applications

Contents

Introduction en français

Pour qu'une chose soit intéressante, il suffit de la regarder longtemps

Gustave Flaubert, *Lettre à Alfred Le Poitevin*

Notre travail de thèse se situe au carrefour de plusieurs disciplines : d'une part, la logique mathématique et l'informatique théorique, d'autre part le traitement automatique du langage naturel et plus particulièrement la sémantique formelle du langage naturel. Le fil conducteur est la présence constante des méthodes logiques issues de la théorie de la preuve et par le problème philosophique qui a motivé notre thèse : quels sont les liens entre la notion de preuve et celle de signification linguistique ou logique ? Plus concrètement, nous étudions des systèmes formels dont les preuves sont vues comme des stratégies gagnantes pour des jeux à deux joueurs. Dans ces jeux, un jouer, appelé Proposant, essaye de construire une justification pour un certain énoncé tandis que l'autre, l'Opposant, essaye de construire une réfutation de cet énoncé. La thèse est composée de trois parties, chaque partie contenant deux ou trois chapitres. La première partie est propédeutique. Dans les deux chapitres qui la composent nous présentons les outils mathématiques utilisés dans notre thèse ainsi que les principes logiques et philosophiques qui ont guidés nos travaux, notamment la sémantique inférentialiste. La deuxième partie de notre thèse contient deux longs chapitres, lesquels présentent les résultats de théorie de la démonstration qui constituent le coeur de notre thèse. En particulier, dans le premier chapitre de cette partie, nous définissons précisément un système de logique dialogique pour la logique classique du premier ordre avec termes. Nous montrons que, pour une formule A, l'existence d'une stratégie gagnante pour A équivaut au fait que A est une théorème logique. Bien que des systèmes de logique dialogique pour la logique classique du premier ordre existent depuis les années 1960 il n'existait pas à ce jour de preuve convaincante publiée de ce résultat, notamment en présence de termes. Dans le deuxième chapitre de cette deuxième partie, nous présentons une sémantique dénotationnelle pour la variante constructive de la logique modale K. En particulier notre sémantique dénotationnelle est une sémantique des jeux dans laquelle les preuves de la logique modale sont interprétées par des stratégies gagnantes pour des jeux à deux jouer. Nous montrons que notre sémantique possède une propriété remarquable : elle est 'totalement adéquate' (fully complete) c'est-à-dire que toute stratégie gagnante est l'interprétation d'au moins une preuve de la logique modale. La troisième et dernière partie se compose de trois chapitres, chacun étant consacré à une application de nos travaux en théorie de la démonstration à la sémantique du langage naturel. Dans le premier chapitre, nous étudions le rapport entre les analyses syntaxiques catégorielles d'une même phrase et les représentations sémantiques logiques de la phrase analysée. Nous montrons que, lorsque certaines conditions sont

respectées, la fonction qui transforme analyses syntaxiques d'une phrase en représentations sémantique logiques est injective. Dans le deuxième chapitre de cette troisième partie, nous appliquons notre système de logique dialogique à la résolution au problème de la reconnaissance d'inférences en langage naturel en utilisant un analyseur syntaxique et sémantique catégoriel. Dans le dernier chapitre de cette partie, nous présentons un système formel pour la résolution d'anaphore et ellipses, problème généralement abordé par des méthodes de théorie des modèles. Nous, au contraire, présentons une solution basée sur la théorie de la démonstration, en développant un système de logique dialogique qui permet de résoudre simplement les anaphores et les ellipses. Dans la conclusion, nous faisons le bilan de notre travail de thèse et essayons de décrire les développements futurs possibles de notre recherche, tant du point de vue mathématique et logique que du point de vue des applications au langage naturel.

Notre travail de thèse se situe au carrefour d'au moins deux disciplines : d'une part, la logique mathématique et l'informatique théorique, d'autre part le traitement automatique du langage naturel, la sémantique formelle du langage naturel et la philosophie du langage. Le fil conducteur de ce travail est la théorie de la démonstration, qui nous a offert outils et méthodologies, et permis d'avancer sur une question fondamentale de logique : quels sont les liens entre la notion de preuve et celle de signification linguistique ? Plus précisément : comprendre le sens d'un énoncé A est-ce être capable de fournir une justification de cet énoncé A ?

Notre intérêt pour la philosophie du langage en tant que logicien et informaticien est naturel : l'une des taches centrales de la philosophie du langage n'est-elle pas d'éclairer ou d'expliquer le concept de signification et de l'appliquer au langage ? Pour parvenir à donner une explication du concept de sens les philosophes ont introduit la notion de théorie de la signification. Pour expliquer cela, reprenons les mots de Michael Dummett:

> according to one well known view, the best method of formulating the philosophical problems surrounding the concept of meaning and related notions is by asking what form that should be taken by what is called 'a theory of meaning' for any one entire language; that is a detailed specification of the meanings of all the words and sentence forming operations of the language, yielding a specification of the meaning of every expression and sentence of the language. [41]

Une théorie de la signification n'est donc rien d'autre qu'une spécification de la manière dont nous attribuons du sens et comprenons les expressions d'un langage. Le langage peut être bien sûr *artificiel*, comme le serait C++, Scheme ou un langage logique, mais il peut aussi être *naturel* comme le sont le français ou l'anglais. Bien qu'ils existent différentes théories de la signification, nous pouvons distinguer deux grandes familles :

- D'un côté, nous trouvons les théories de la signification qui ont comme concepts centraux les concepts de *vérité* et de *référence*.
- De l'autre côté, nous trouvons les théories de la signification qui ont comme concept central le concept d'*inférence*.

Théories référentialistes de la signification

Dans la grande famille des théories de la signification basées sur les concepts de vérité et de référence, la signification d'une expression linguistique est spécifiée par les phrases dans lesquels elle apparaît. Plus précisément la signification d'une expression est la contribution que l'expression apporte à la détermination du valeur de vérité d'une phrase dans laquelle elle apparaît. Si, par exemple, nous considérons les deux phrases

(1) Emmanuel Macron est le président de la République Française.

(2) Mario Draghi est le président de la République Française.

nous remarquons que la première phrase est vraie tandis que la seconde est fausse. Selon une théorie de la signification référentialiste ceci est du au fait que les deux références des deux noms propres 'Emmanuel Macron' et 'Mario Draghi' sont différentes : l'un désigne l'individu qui est président de la république Française en ce moment alors que le l'autre ne réfère pas à un tel individu. Considérons la sémantique, au sens de la théorie des modèles, des formules logique. Cette analyse de la signification des phrases (formules) logiques nous offre l'un des exemples paradigmatiques de théorie de la signification référentialiste. Soit $\mathcal{L}$ un langage logique du premier ordre[1]. L'interprétation des termes du langage est spécifiée au moyen d'une fonction associant aux constantes et aux variables d'individus des éléments de la structure. Supposons qu'à chaque terme t du langage corresponde un élément t^I de la structure d'interprétation. Un prédicat du langage est interprété par un sous-ensemble du produit cartésien de la structure. Ainsi une formule atomique $P(t_1, \dots t_n)$ est-elle vraie dans la structure d'interprétation si et seulement si le tuple $(t_1^I, \dots t_n^I)$ appartient au sous-ensemble de la structure qui interprète le prédicat P. Ensuite, lors de l'interprétation des formules complexes, écrites avec les connecteurs logiques et les quantificateurs, on spécifie au moyen de clauses inductives la sémantique : une conjonction $A \wedge B$ est vraie dans l'interprétation si et seulement si la formule A et la formule B sont vraies dans celle interprétation, etc.

La sémantique formelle est une branche de la linguistique formelle qui construit des modèles mathématiques du sens des expressions linguistiques. L'approche dominante en sémantique formelle est *référentialiste* : la signification d'un énoncé E est définie via le concept auxiliaire de *forme logique* de l'énoncé. La forme logique d'un énoncé est une formule F appartenant à un certain langage logique $\mathcal{L}$, et elle est censée capturer l'une des lectures possibles de l'énoncé, éliminant ainsi les ambiguïtés. Par exemple, l'énoncé

(3) Chaque enfant mange une pizza

Peut signifier

(4) Étant donné un enfant x on peut toujours trouver une pizza y telle que x mange y.

Ou bien

(5) Il y a une pizza y telle que tout enfant la mange.

[1]Une définition précise de langage logique, termes et structure d'interprétation sera donnée dans le chapitre 1

La paraphrase 4 est capturée par la formule

$$\forall x\,[\text{enfant}\,(x) \supset (\exists y\,\text{pizza}\,(y) \wedge \text{mange}\,(x, y))]$$

Alors que la paraphrase 5 se formalise ainsi :

$$\exists y\,[\text{pizza}\,(y) \wedge (\forall x\,\text{enfant}\,(x) \supset \text{mange}\,(x, y))]$$

en utilisant le concept de forme logique, on peut définir la signification d'un énoncé E, ou mieux d'une lecture possible de E, comme étant la classe des modèles qui satisfont la forme logique correspondante à la lecture considérée. Pour reprendre l'exemple 3 supra, la signification de l'énoncé par rapport à la lecture 5 est donnée par l'ensemble

$$\{\mathfrak{M} \mid \mathfrak{M} \models \exists y\,[\text{pizza}(y) \wedge (\forall x\,\text{enfant}(x) \supset \text{mange}\,(x, y))]$$

Comme nous l'avons déjà anticipé, ce type de sémantique pour les phrases du langage naturel s'appuie sur le concept de vérité. La signification d'une phrase est l'ensemble des situations qui rendant la phrase vraie, ces conditions de vérité, d'où la terminologie "sémantique vériconditionnelle".

Ce type d'approche de la sémantique formelle est tout à la fois simple et fructueux: on arrive ainsi à donner un sens précis à de substantiels fragments du langage naturel. Cependant, ce type d'approche souffre d'une certains nombre de défauts, lesquels remettent en cause sa pertinence en tant qu'analyse de la signification des phrases du langage naturel.

Une première critique est la pauvreté de cette analyse sémantique, tout au moins d'un certain point de vue. Considérons par exemple, deux phrases qui sont équivalentes en termes des valeurs de vérité :

(6) Cécile est grande et riche

(7) Ce n'est pas le cas que Cécile n'est pas grande ou n'est pas riche.

Les deux phrases peuvent être 'traduites' dans les deux formules logiques qui suivent

(8) $\mathit{grande}\,(\texttt{Cécile}) \wedge \mathit{riche}\,(\texttt{Cécile})$

(9) $\neg(\neg \mathit{grande}\,(\texttt{Cécile}) \vee \neg \mathit{riche}\,(\texttt{Cécile}))$

les deux ensembles de modèles des deux formules ci-dessus sont les mêmes. Malgré cela, les deux formules présentent de notables différences de signification, surtout si nous nous concentrons sur les aspects inférentiels. On peut considérer que l'inférence qui va de la phrase *Cécile est grande et riche* à la phrase *Cécile est riche* est une inférence *élémentaire*. Il est plus difficile de considérer l'inférence qui va de *C'est n'est pas le cas que Cécile n'est pas grande ou n'est pas riche* à la phrase *Cécile est riche* comme une inférence élémentaire.

Mais il y a pire : considérons deux théorèmes logiques quelconques e.g., $(A \supset B) \supset (\neg B \supset \neg A)$ et $A \vee \neg A$. Puisqu'un théorème logique est par définition une formule qui est vraie dans toute structure d'interprétation les deux théorèmes ont la même signification. C'est pour le moins étonnant.

Théories inférentialiste de la signification

De l'autre côté, on trouve les théories inférentialistes de la signification : une théorie inférentialiste de la signification nie que les conditions de vérité jouent le rôle principal en sémantique. Au contraire le concept fondamental pour la signification des expressions est celui d'*inférence* ou de *justification*. Selon les mots de Robert Brandom :

> The standard way [of classical semantics] is to assume that one has a prior grip on the notion of truth, and use it to explain what good inference consists in [...] [I]nferentialist pragmatism reverses this order of explanation [...] It starts with a practical distinction between good and bad inferences, understood as a distinction between appropriate and inappropriate doings, and goes on to understand talk about truth as talk about what is preserved by the good moves. [15]

Une théorie inférentialiste de la signification met l'accent sur les propriétés sémantiques des *énoncés*. Les relations inférentielles existent entre les phrases et non entre expressions sub-sententielles. (On ne peut pas, par exemple, inférer un nom à partir d'un autre). Ainsi, l'inférentialiste n'expliquera pas les propriétés sémantiques des termes singuliers, par exemple, en termes de relations représentationnelles entre ces termes singuliers et les éléments du monde ; il expliquera plutôt ce qui est distinctif des termes singuliers en termes de leur rôle dans certains types d'inférences.
Bien qu'existent des désaccords, parfois majeurs, entre les auteurs défendant la sémantique

inférentialiste [2], on peut affirmer que les deux points qui suivent sont acceptés, à notre connaissance, par tout défenseur du point de vue inférentialiste.

1. La signification d'un énoncé est la connaissance qu'il faut posséder (implicitement ou explicitement) afin de pouvoir comprendre l'énoncé;

2. cette connaissance doit en principe être observable dans les interactions entre l'orateur, l'auditeur et l'environnement

Remarquons que, pour le point 1, la signification d'un énoncé d'un langage (naturel ou artificiel) ne peut pas coïncider avec l'objet capturé par la définition référentialiste de la signification : les locuteurs ne sont capables de stocker qu'une quantité limitée de données. Ceci veut dire que les connaissances nécessaires pour comprendre le sens de la langue elle-même devraient également être limitées ou, au moins, pouvoir être énumérées et décrites de manière récursive à partir d'un ensemble limité de données et de règles. Du moment que, en général, il n'existe pas de moyens finis d'énumérer l'infinité de modèles d'une formule, ni d'énumérer de façon finie l'infinité des individus et des relations dans un seul de ces modèles, un inférentialiste refuse la définition de signification d'une phrase S comme l'ensemble des modèles qui satisfont l'une des lectures de S.

Inférentialisme et dialogues argumentatifs

Les approches référentialistes de la théorie de la signification utilisent les méthodes de la théorie de modèles, une branche de la logique mathématique — ou s'en s'inspirent fortement. Dans notre thèse, nous utiliserons des méthodes et des concepts de la théorie de la démonstration. La théorie de la démonstration, ou théorie de la preuve, est une branche de la logique mathématique qui s'occupe d'étudier les propriétés formelles des preuves, on dit aussi déductions ou dérivations. Comme cela semble naturel, une théorie sémantique basée sur le concept de preuve est étroitement liée à l'étude des propriétés mathématiques des preuves, et les développements formels de la théorie inférentialiste du sens, basés sur les outils de la théorie de la démonstration, ne manquent pas [53]. Les développements formels du paradigme inférentialiste de la signification sont basés sur un système de preuve formelle

[2] Le lecteur intéresse peut consulter les premiers chapitres du livre de C. Cozzo *Meaning and Argument* [30] où un comparaisons détaillé entre différents versions de l'inférentialisme en philosophie du langage est présentée

inventé par Gentzen et étudié en profondeur par Prawitz: la déduction naturelle [117]. Nous choisirons une approche différente, pour mettre en œuvre la théorie inférentialiste de la signification : à savoir la logique dialogique [96, 97]. Notre choix est motivé par deux facteurs distincts.

Premièrement, la déduction naturelle se “comporte mal” par rapport à la logique classique, et des propriétés essentielles de la déduction naturelle ne sont plus vraies. Par exemple, dans un système de déduction naturelle classique, il n'y a pas de propriété de la sous-formule. La logique classique est un outil essentiel pour l'étude du raisonnement et de l'inférence en langage naturel et notre thèse comprend une étude, bien que limitée, de ces sujets.

Ensuite, à notre avis, le lien avec une sémantique basée sur la notion d'argument est plus clair dans le paradigme inférentialiste: un argument en faveur d'une proposition est souvent développé lorsqu'un public critique, réel ou imaginaire, doute de la vérité ou de la plausibilité de la proposition. Dans ce cas, afin de faire valoir avec succès la déclaration, l'orateur ou le promoteur de celle-ci doit être capable de fournir toutes les justifications que le public pourra exiger. En prenant cette idée au sérieux, une approximation du sens d'une phrase dans une situation donnée peut être obtenue en étudiant les dialogues argumentatifs qui surviennent une fois que la phrase est affirmée devant un public critique. Ce type de situation est fidèlement capturé —avec un degré raisonnable d'approximation— par la logique dialogique. La logique dialogique analyse le concept de validité d'une formule à travers le concept de stratégie gagnante dans un type particulier de jeu à deux joueurs. Ce type de jeu n'est rien d'autre qu'un dialogue argumentatif entre un joueur, appelé Proposant, qui affirme la validité d'une certaine formule et un autre joueur, appelé Opposant, qui en conteste la validité. Le Proposant entame le dialogue argumentatif en affirmant une certaine formule. L'opposant attaque à tour de rôle l'affirmation du Proposant en fonction de la forme logique de la formule affirmé. Le Proposant peut, en fonction de la forme de l'attaque faite par l'Opposant, soit défendre son affirmation, soit contre-attaquer. Le débat évolue selon ce schéma. Le Proposant gagne le débat s'il a le dernier mot, c'est-à-dire : la défense contre l'une des attaques de l'Opposant consiste dans l'assertion d'une formule que l'Opposant ne peux pas attaquer. On peut donc énoncer l'idée fondamentale de la sémantique dialogique comme il suit :

> la signification d'un énonce E est donnée par la forme du dialogue dans lequel le locuteur énonçant l'assertion E fournit à un interlocuteur critique toutes les justifications nécessaires à la justification de son assertion E.

Dans notre travail de thèse, nous nous concentrerons sur une représentation dialogique des preuves formelles, et nous étudierons en détail deux systèmes logiques dont les preuves formelles peuvent interpréter comme des stratégies gagnantes pour des jeux: la logique dialogique et la sémantique des jeux.

Résultats du travail de thèse

Dans cette section nous décrivons brièvement les résultats principaux de notre travail de thèse, lesquels sont de deux styles. Il y a d'une part des résultats de théorie de la preuve et d'autre part des résultats sur la modélisation de la syntaxe ou la sémantique du langage naturel.

Théorie de la preuve

Dans notre thèse, nous explorons deux théories assez similaires : la logique dialogique et la sémantique des jeux. Nous exposons brièvement les deux théories et présentons nos propres contributions.

Logique Dialogique

Le premier article sur la logique dialogique est paru en 1958 [96]. Les idées présentées par Lorenzen ont été développées en utilisant les outils de la théorie de jeux par son étudiant Lorenz dans sa thèse de doctorat [97]. Les objets de base de la logique dialogique sont des jeux à deux joueurs et sont une idéalisation d'un dialogue argumentatif. Les deux joueurs sont appelées Proposant (ou **P**) et Opposant (ou **O**). Les deux joueurs prennent la parole à tour de rôle et ont des rôles différents dans le débat : **P** essaye de construire une justification pour une certaine formule F, alors que **O** essaye de construire une réfutation de cette formule F. Chaque connecteur ou quantificateur de la formule peut être attaqué par un joueur et défendu par l'autre. Par conséquent, à chaque formule complexe sont associé des questions en fonction de son connecteur principal tandis que des réponses à ces questions sont attendues.

Un jeu est donc une suite (fini ou infinie) d'interventions de **P** et de **O** à tour de rôle. La première intervention est faite par **P** et c'est l'assertion de la formule F. Toute autre intervention dans le débat doit être "justifiée" au sens suivant : une intervention consiste à poser une question sur une formule affirmée par l'autre joueur, ou à répondre à l'une des questions posées par l'autre joueur. Les jeux sont réglementés par une série de restrictions, par exemple **P** ne peut affirmer une formule atomique que si **O** a déjà affirmé la même formule atomique. Chaque intervention de **O** est une réaction à l'intervention de **P** qui la précède immédiatement. Une stratégie est une fonction qui determiné quel est la prochaine intervention d'un joueur en fonction des interventions précédentes, c'est-à-dire en fonction l'historique du jeu. Une stratégie est dite gagnante lorsque le joueur qui la suit gagne la partie quels que soient les coups de l'autre jouer et l'historique. Nous nous intéressons aux stratégies qui sont gagnantes pour **P**. Notre principale contribution à l'étude de la logique dialogique est la preuve de l'énoncé suivante

> Soit $\mathcal{L}$ un langage du premier ordre *avec termes* et A une formule de $\mathcal{L}$. La formule A est valide (au sens de la logique classique) si et seulement s'il existe une stratégie **P**-gagnante pour A.

Notre preuve de cet énoncé est constructive : nous transformons toute stratégie **P**-gagnante en une preuve du calcul des séquents de $\vdash A$ et vice-versa, nous transformons toute preuve de $\vdash A$ en une stratégie gagnante pour **P**. Bien que le système de logique dialogique classique que nous avons esquissé soit présenté par Lorenzen dans l'article du 1958, à ce jour il n'y avait pas de preuve de cet énoncé. Il existaient des preuves pour le calcul propositionnel comme celle proposée par Herbelin dans sa thèse [68] que notre preuve étend, ou pour des langages du premier ordre restreints dont les seuls termes sont des constantes [25]. Nous appelons SLK le calcul que nous utilisons pour montrer l'équivalence entre validité classique et existence d'une stratégie **P**-gagnante. Ce système SLK est un calcul polarisé. Si $\mathfrak{D}$ est une preuve qui se terminé par une application d'une règle d'introduction droite du quantificateur existentiel ou de la disjonction

$$\dfrac{\dfrac{\begin{array}{c}\mathfrak{D}'\\ \vdots\end{array}}{\Gamma \vdash \underline{A[t/x]}, \exists x A, \Delta}\ \mathsf{R}}{\Gamma \vdash \exists x A, \Delta}\ \exists^{\mathsf{R}} \qquad\qquad \dfrac{\dfrac{\begin{array}{c}\mathfrak{D}'\\ \vdots\end{array}}{\Gamma \vdash \underline{A_i}, A_1 \vee A_2, \Delta}\ \mathsf{R}}{\Gamma \vdash A_1 \vee A_2, \Delta}\ \vee_i^{\mathsf{R}}\ i \in \{1,2\}$$

alors la formule soulignée dans la prémisse de la règle est la formule 'introduite' par la règle R. En effet les stratégies gagnantes pour **P** que nous étudiions dans notre thèse ont été introduites —dans le cas de la logique dialogique intuitionniste— par Felscher [46]. Ces stratégies —que Felscher appelle *formelles*— ont une particularité : le proposant est forcé de répondre à toute attaque dirigée contre une disjonction ou une formule existentielle immédiatement. Ce type de restriction sur les stratégies correspond naturellement à la propriété de polarisation esquissée ci haut.

Sémantiques de jeux

La sémantique des jeux est une forme de sémantique dénotationnelle inspirée par la logique dialogique. Une sémantique dénotationnelle est une manière d'interpréter les preuves d'un système logique par des objets mathématique. Une sémantique dénotationnelle possède une particularité : les preuves que sont égales modulo une notion de réduction (usuellement l'élimination de la coupure ou la β-réduction) sont interprété par le même objet. Les premiers travaux en sémantique des jeux sont apparus au début des années 1990 pour la logique linéaire [1, 76, 89, 13].

Dans notre thèse, nous nous développons une sémantique des jeux pour la variante constructive de la logique modale K (appelée CK). Les formules de CK sont écrites en utilisant les connecteurs $\supset$ et $\wedge$ et les modalités $\Box$ et $\Diamond$. La logique CK peut être définie comme le plus petit ensemble de formules contenant :

- toute tautologie de la logique minimale ;
- chaque instance de l'axiome $\Box(A \supset B) \supset \Box A \supset \Box B$;
- chaque instance de l'axiome $\Box(A \supset B) \supset \Diamond A \supset \Diamond B$;

et clos par :

- nécessitation : si A appartient à CK alors $\Box A$ appartient aussi ;
- modus ponens : si A et $A \supset B$ appartiennent a CK alors B appartient aussi ;

Un calcul des sequents complet pour cette logique —LCK— s'obtient en rajoutant les deux règles suivantes à un calcul des sequents 'standard' pour la logique minimale

$$\frac{\Gamma \vdash C}{\Box\Gamma \vdash \Box C}\,\mathsf{K}^{\Box} \qquad\qquad \frac{\Gamma, B \vdash C}{\Box\Gamma, \Diamond B \vdash \Diamond C}\,\mathsf{K}^{\Diamond}$$

où $\Box\Gamma = \Box A_1, \ldots \Box A_n$ si $\Gamma = A_1, \ldots A_n$. Cette logique est bien comprise tant du point de vue de la théorie des modèles que du point de vue de la théorie de la démonstration. Malgré cela des sémantiques dénotationnelles pour CK n'ont pas été étudié, hormis dans [11] où les auteurs étudient la structure catégorielle des modèles dénotationnels de CK. Dans notre travail de thèse nous construisons un modèle dénotationnel concret pour CK.

Dans la sémantique des jeux, une preuve s'interprète par une stratégie gagnante dans un jeu à deux joueurs dans une arène, laquelle est un graphe orienté représentant la structure d'une certaine formule A. L'arène d'une formule complexe se définit récursivement à partir des arènes des formules atomiques. Soit A une formule modale écrite avec les connecteurs $\supset, \wedge$ et les modalités $\Box, \Diamond$. L'arène d'une formule A, notée $[\![A]\!]$, est un graphe orienté dont les sommets sont les formules atomiques contenues dans A et les occurrences de modalités dans A, et dont les arêtes sont de deux sortes. Une sorte d'arête (dénotée par $\rightarrow$) permet de retrouver les connecteurs binaires de A, tandis que l'autre sorte d'arêtes (dénoté par $\rightsquigarrow$) permet de retrouver la portée des modalités à l'intérieur de la formule A.

Chaque sommet v d'une arène $[\![A]\!]$ a une polarité. Cette polarité, positive ou négative, est la même de l'occurrence de la formule atomique, ou modalité, de A qui décore le sommet v[3]. Un sommet positif v est appelé racine s'il n'existe pas un sommet w tel que $v \rightarrow w$. Un coup est un sommet et il est un coup de **O** s'il est positive, il est un coup de **P** sinon. Les jeux sont des suites finies alternées de coups de **O** et de **P**.

- **O**-commence toujours un jeu : le premier coup d'un jeu est une racine de l'arène.
- Chaque coup w du jeu, sauf le premier, est justifié par un coup précèdent v de l'autre jouer : $w \rightarrow v$ dans l'arène ;
- chaque coup dans un jeu est soit une variable propositionnelle soit un sommet $\Diamond$;
- chaque coup de **O** est justifié par le coup de **P** qui le précède dans le jeu ;
- chaque coup w de **P** porte le même label que le coup v de **O** immédiatement précèdent : si v est une variable propositionnelle a (resp. une modalité $\Diamond$) alors w est une occurrence de la même variable propositionnelle a (resp. une occurrence de $\Diamond$).

[3]une définition précise de la notion de polarité d'une formule est donnée dans le chapitre 1 de notre travail de thèse

Un jeux est gagné par un des deux joueurs lorsque l'autre joueur ne peut plus répondre. Une stratégie gagnante est une arbre dont chaque branche est un jeu gagné par **P**. Les bifurcation de cet arbre résultent des coups de **O**. Pour caractériser le stratégies correspondant à des preuves modales, nous considérerons deux contraintes supplémentaires sur les coups de **P** acceptés. Dans l'arène chaque sommet w possède une hauteur, le nombre de modalités qui portent sur w.

La première contrainte exige que chaque coup de **P** aie la même hauteur le coup précédent de **O**. La deuxième contrainte nous permet de définir une notion de *sous-jeu*: chaque fois qu'un coup de **O** w est dans la portée d'une nouvelle modalité m de polarité positive (une modalité positive de hauteur k qui ne porte pas sur un coup précédent) un nouveau sous-jeu de hauteur k est ouvert. Le sous-jeu se termine lorsque **O** joue un coup v qui n'est pas dans la portée d'une modalité ou bien lorsque **O** joue un coup v qui est dans la portée d'une modalité positive d'auteur k et différent de m. Dans une stratégie gagnante modale tout sous-jeu possède la caractéristique suivante : si w est le premier coup du sous-jeu et m sa modalité positive de hauteur k, les propriétés suivantes sont satisfaites :

- tout x appartenant au sous-jeu est sous la portée d'une modalité de hauteur k;
- Si x appartient au sous-jeu et x est sous la portée d'une modalité positive n de hauteur k, alors $n = m$;
- Si x appartient au sous jeu et x est sous la portée d'une modalité négative n de hauteur k, alors pour tout coup y appartenant à un autre sous-jeu, y n'est pas sous la portée de n;
- si $m = \Diamond$ alors il existe une modalité négative n de hauteur k telle que $n = \Diamond$ et un coup x sous la portée de n et qu'appartient au sous-jeu.

Cette caractérisation des stratégies gagnantes modales permet d'interpréter toute preuve $\mathfrak{D}$ de LCK par une stratégie gagnante modale $\{\!\{\mathfrak{D}\}\!\}$ et nous montrons alors que si $\mathfrak{D}$ se réduit $\mathfrak{D}'$ en appliquant l'algorithme d'élimination de la coupure alors $\{\!\{\mathfrak{D}\}\!\} = \{\!\{\mathfrak{D}'\}\!\}$. En outre, nous montrons que notre sémantique est totalement adéquate, c'est-à-dire

> Soit $\{\!\{-\}\!\}$ la fonction qui associe à une preuve $\mathfrak{D}$ du calcul des sequents LCK une stratégie gagnante modale $\{\!\{\mathfrak{D}\}\!\}$; la fonction $\{\!\{-\}\!\}$ est surjective : pour toute stratégie gagnante $\mathcal{S}$ il existe une preuve $\mathfrak{D}$ telle que $\mathcal{S} = \{\!\{\mathfrak{D}\}\!\}$.

La preuve de l'énoncé en haut et obtenu en utilisant un algorithme de 'sequentialization' : nous montrons comment transformer une stratégie gagnante pour A en une preuve $\mathfrak{D}$ de A.

Modélisation de la syntaxe et de la sémantique du langage naturel

Grammaires Catégorielles

Les grammaires catégorielles permettent d'analyser la syntaxe et la sémantique du langage naturel grâce à la théorie de la démonstration. Étant donné une phrase en langage naturel $w_1, \ldots w_n$ une fonction calculable associe à chaque mot w_i une formule $l(w_i)$ de la logique linéaire intuitionniste multiplicative non commutative (du calcul de Lambek [90] ou d'une variante de celui-ci). La grammaticalité de la phrase correspond à la prouvabilité, dans la logique en question, du jugement $l(w_1), \ldots l(w_n) \vdash s$ (s représente la catégorie grammaticale des phrases). Le même jugement peut avoir plusieurs preuves, et chacune de ces preuves correspond à une analyse syntaxique distincte de la phrase. Par l'isomorphisme de Curry-Howard, chaque analyse syntaxique peut être vue comme un lambda-terme linéaire, lequel fournit des instructions sur la façon de composer le sens des mots individuels, eux aussi décrits par un lambda termes appelés lambda-termes sémantiques. Dans l'analyse syntaxique nous substituons ensuite aux variables libres — correspondant aux entrées lexicales w_i — les lambda termes sémantiques associées pour obtenir, par β-réduction, une représentation du sens logique de la phrase. Une question naturelle est alors la suivante : si une phrase $w_1, \ldots w_n$ possède au moins deux analyses syntaxiques différentes donnent-elles lieu à deux représentations logiques différentes ? Nous montrons que si la question est naïvement formulée, ces analyses sémantiques peuvent être les mêmes. Cependant, en considérant une classe restreinte de λ-termes sémantiques et une notion forte des différences entre λ-terme syntaxiques, les analyses sémantiques restent différentes. Pour cela, il faut se restreindre à des λ-termes sémantiques qui soient λ_I-termes de la forme

$$\lambda x_1, \ldots, \lambda x_n k M_1 \cdots M_m$$

où k est une constante. Pour établir la différence entre représentations logiques d'une phrase , nous définissons une relation de *dominance* entre symboles atomiques d'un λ-terme. Intuitivement la dominance nous permet de reconstruire l'arbre applicatif d'un λ-terme. Nous montrons que la dominance entre constantes est préserve par β-réduction c'est à dire

> Soit M λ_I terme typé qui contient deux occurrences de constantes k et k' telles que k domine k' dans M. Supposons que M β-réduit à M'. Alors chaque trace k_i de k est associée à un ensemble d'occurrences $k_i'^j$ de k' dans M' et k_i domine $k_i'^j$ en M'.

Par conséquent, si deux analyses syntaxiques P_1 et P_2 de la même phrase définissent deux relations de dominance différentes entre leurs variables libres, (deux analyses de la même phrase ont les mêmes variables libres) alors leur forme logique respective est différente. En effet, la forme logique est obtenue en substituant les variables libres de P_1 et P_2 par les mêmes lambda-termes sémantiques. Le symbole de tête d'un terme sémantique est une constante. Si la variable w_1 domine la variable w_2 en P_1 mais pas en P_2 (ou vice-versa) nous pouvons conclure que dans la forme logique de P_1 il y aura une constante k_1 qui domine une constante k_2 et que le couple formé par k_1 et k_2 n'apparaît pas dans la relation de dominance de la forme logique de P_2.

Reconnaissance de l'implication textuelle

On dit qu'un texte A en entraîne un autre B lorsque B est conséquence de A. Une tâche assez standard du traitement automatique du langage naturel, nommée RTE (Recognizing Textual Entailment), consiste à détecter qu'un texte est conséquence logique d'un autre. Par exemple “Chaque canard est un oiseau” a pour conséquence logique “Chaque patte de canard est une patte d'oiseau” alors que, “Chaque Italien aime la pizza et Charles n'est pas italien” n'a pas comme conséquence “Charles n'aime pas la pizza”.

Le data-set FraCaS [28][4] à été crée dans les années 1990. L'objectif était de développer un framework général pour la sémantique computationelle. Le data-set consiste en des problèmes des reconnaissance de l'inférence textuelle. Chaque problème consiste en une ou plusieurs assertions et une question qui à comme réponse ‘oui’ ou ‘non’.

(10) A Swede won a Nobel prize.

(11) Every Swede is a Scandinavian.

(12) Did a Scandinavian won a Nobel prize? [Yes]

[4] il existe aussi une version en français du data-set FraCas [7]

Nous avons appliqué la logique dialogique et les grammaires catégorielles à la reconnaissance de l'implication textuelle dans le corpus FraCas. Les grammaires catégorielles produisent une représentation logique des phrases. Puis on traduit les couples question-réponse du data-set en des affirmations. Dans le cas précédent la couple formé par "Did a Scandinavian won a Nobel Prize" et "no" a été traduit dans la phrase "Some Scandinavian won a Nobel Prize". En traduisant les phrases en formules logiques de l'exemple ci-dessus on obtient

(13) A Swede won a Nobel prize
$\exists\, x_1\, [\, \mathit{Swede}\,(x_1) \wedge (\exists\, x_2\, \mathit{Nobel\text{-}prize}\,(x_2) \wedge \mathit{won}\,(x_1, x_2))]$

(14) Every Swede is a Scandinavian
$\forall\, x_3, [\, \mathit{Swede}\,(x_3) \supset \mathit{Scandinavian}\,(x_3)]$

(15) Some Scandinavian won a Nobel prize
$\exists\, x_4\, [\, \mathit{Scandinavian}\,(x_4) \wedge (\exists\, x_5\, \mathit{Nobel\text{-}prize}\,(x_5) \wedge \mathit{won}\,(x_4, x_5))]$

enfin nous montrons l'existence (ou la non-existence en fonction de cas) d'une stratégie gagnante pour la formule $H_1 \wedge \ldots \wedge H_n \supset C$ dont les H_i sont les formules correspondantes aux phrases présentes dans le data-set et C est la formule correspondante à la phrase obtenue en utilisant le couple question réponse présent dans le data set. Nous traitons dans les détails certains exemples simples : e.g, monotonicité des quantificateurs.

Ensuite nous nous attaquons à un problème légèrement plus complexe. Nous considérons des fragments de texte qui sont dans la relation d'implication textuelle en vertu de la signification des mots non-logiques e.g., par exemple tout locuteur français reconnaît que la phrase 'Jean mange une pomme' est dans la relation d'implication textuelle avec la phrase 'Jean mange un fruit rouge ou bien Jean mange un fruit jaune ou bien Jean mange un fruit vert" ceci car nous pouvons définir le mot pomme par "Fruit du pommier, charnu, de forme plus ou moins arrondie, de couleur verte, jaune ou rouge selon la variété" (dictionnaire Larousse). Afin de reconnaître ce type de relations, nous considérons des jeux dialogiques dans lesquels les joueurs jouent modulo un certains ensembles de définitions i.e., formule logique de la forme

$$\forall x_1, \ldots \forall x_n (Q \iff A)$$

où Q est une formule atomique (appelons cette formule le *definiendum*) et A une formule quelconque (appelons cette formule le *definiens*). Dans les jeux que nous considérons lorsque un jouer affirme une formule atomique Q qui est definiendum, l'autre joueur peut demander d'expliciter le définiens de la formule. Ces jeux module définitions sont appelé jeux de dépliage. La notion de dépliage d'une formule atomique à été étudié en premier par Prawitz [117] dans le contexte de la déduction naturel et développé par Dowek [40] dans le cadre de la Déduction modulo.

Résolution d'anaphores et ellipses

Définir exactement qu'est-ce que c'est est une anaphore et qu'est-ce que c'est est une ellipse est une tache difficile [83]. Nous pouvons sommairement définir l'anaphore comme un phénomène linguistique par lequel l'interprétation d'une occurrence d'une expression dépend de l'interprétation d'une occurrence d'une autre expression. Cette caractérisation permet de reconnaître que dans les deux phrases 'Gertrude mange une pomme. Elle est exquise' le pronom 'elle' est anaphorique. L'interprétation du pronom 'elle' est soit 'une pomme' soit 'Gertrude' (Gertrude pourrait être particulièrement gracieuse lorsqu'elle mange une pomme). Appelons l'expression linguistique dont l'interprétation d'une anaphore dépend *l'antécédent*. Les humains arrivent souvent à résoudre les anaphores, c'est-à-dire trouver un antécédent approprié pour une expression anaphorique. La résolution d'anaphore est une tache difficile dans le traitement automatique du langage naturel. Dans un contexte logique, la résolution des anaphores est souvent traitée par des méthodes de la théorie des modèles [80]. Dans notre travail de thèse, nous introduisons une approche différente au problème de la résolution d'anaphores. En particulier nous l'approchons avec les outils de la théorie de la preuve. Nous introduisons un nouveau quantificateur du premier ordre (que nous écrivons $\mathcal{A}$) dont la signification est donnée par les règles qui concernent son 'utilisation' dans un dialogue argumentatif. Plus concrètement nous allons définir un système de logique dialogique du premier ordre (pour un langage multisorte) dont les formules sont écrites avec occurrence du quantificateur $\mathcal{A}$. Dans un jeu dialogique les formules qui sont affirmées par le joueur **O** ont un statut différent des formules qui sont affirmées par **P**. Les formules affirmées par **O** forment les hypothèses du dialogue i.e., les formules qui peuvent être considérées comme acceptées dans le dialogue. En effet, **O** peut affirmer n'importe quelle formule atomique et ainsi se défendre contre n'importe quel coup d'attaque de **P**. Nous pouvons pousser cette intuition : si dans le jeu **O** affirme qu'un certain individu k possède une certaine propriété A alors *dans le contexte du jeu* l'existence de cet individu est attestée. Appelons l'ensemble des formules affirmées par **O** dans le dialogue *le terrain d'entente* du dialogue. Ce terrain d'entente s'agrandit au fur et à mesure que le jeu progresse. Nous disons qu'une constante k apparaît dans le terrain d'entente si celui-ci contient une formule $B(k)$. Les règles d'attaque et de défense du quantificateur $\mathcal{A}$ sont ainsi définies: si une formule $\mathcal{A}xA$ est affirmée dans le jeu alors cette affirmation peut être attaquée seulement si au moins une constante apparaît dans le terrain d'entente. Si la formule est attaquée par un joueur l'autre joueur peut se défendre en affirmant $A(k)$ pour un certain k qui apparaît dans le terrain d'entente. Intuitivement les variables qui sont liées par une occurrence de $\mathcal{A}$ représentent les expressions anaphoriques. Nous appelons ce type de jeux dialogique *jeux anaphoriques*. Nous montrons que l'ensemble des formules pour lesquelles il existe une stratégie anaphorique est un ensemble consistant : il ne contient pas toutes les formules du langage. Ensuite nous appliquons les stratégies pour les jeux anaphoriques au problème de la reconnaissance de l'inférence textuelle en présence d'anaphores. On traite certains problèmes du data-set FraCas par exemple,

(16) Smith attended a meeting.

(17) She chaired it.

(18) Did Smith chaired a meeting? [Yes]

Afin de réduire l'espace des antécédents possibles pour une expression anaphorique on utilise un langage logique multisorte. Il est naturel, au moins en anglais, de considérer que certaines expressions anaphoriques véhiculent de l'information sur le statut ontologique de leur antécédent[5]. De plus il est naturel que certains verbes, adjectifs, etc. peuvent être appliqués uniquement à certaines sortes d'entité. Si par exemple nous considérons la phrase 'Smith took a train to Baltimora. It whistled at 12 o'clock' il est clair que 'it' fait référence au train. le pronom 'It' désigne une entité inanimée et les villes comme Baltimore ne sifflent pas. On peut donc représenter les phrases de l'exemple ci-dessous par les formules qui suivent :

(19) Smith attended a meeting.
$F_1 = \exists x_1{}^o\,(\mathit{meeting}\,(x_1) \wedge \mathit{attended}\,(\texttt{smith}, x_1))$

[5]En français comme en italien un pronom personnel comme 'elle' réfère à la personne grammaticale et ne véhicule donc pas forcément de l'information sur le statut ontologique de l'entité (être humain, objet inanimé ou événement, etc.). Pour cette raison, on peut considérer que la phrase 'Gertrude mange une pomme. Elle est exquise' est ambiguë entre la lecture dont 'elle' réfère à 'une pomme' ou 'elle' réfère à 'Gertrude'. Évidemment la lecture dont le pronom réfère à 'une pomme' est plus naturelle. Ceci puisqu'on considère l'utilisation littérale de l'adjectif 'exquise'

(20) She chaired it.
$F_2 = \mathcal{A} x_2{}^h \mathcal{A} x_3{}^o \, (\, chaired\,(x_2, x_3))$

(21) Smith chaired a meeting.
$C = \exists x_4{}^o \, (chaired\,(\texttt{smith}, x_4))$

nous pouvons ainsi construire une stratégie gagnante pour la formule $F_1 \wedge F_2 \supset C$. Nous considérons aussi le problème de la résolution d'ellipse en le réduisant au problème de la résolution d'anaphore. En suivant l'analyse des Davidson [35] nous considérons que chaque verbe dans une phrase introduit un évènement. Ainsi la forme 'logique' de la phrase 'Gertrude a coupé une pomme dans la baignoire à minuit' est 'il y a un évènement, de découpage de pomme, qui s'est passé dans la baignoire à minuit et dont l'agent est Gertrude'. Cette stratégie de formalisation des phrases nous permet de considérer certaines ellipses comme des anaphores : ainsi dans la phrase 'Gertrude a rencontré Marie, Antoine aussi' 'aussi' fais référence à l'évènement de rencontrer Marie.

Organisation du manuscrit

Notre travail de thèse est divisé en trois parties. Chaque partie se compose d'un maximum de trois chapitres. Le travail est organisé selon un schema conceptuel qui va d'abstrait considérations philosophiques à des applications plus concrètes aux domains du traitement automatique du traitement du langage naturel en passant par une partie logico-mathématique. Nous allons maintenant présenter au lecteur le contenu des trois parties de notre thèse.

Mathematical and Philosophical foundations

Cette partie de la thèse contient deux chapitres et poses les fondements logique et philosophique de notre travail.

Mathematical preliminaries Le premier chapitre de notre thèse est dédié à poser les fondements mathématiques de notre travail de thèse. Nous rappelons les définitions des outils logiques et mathématiques que nous utiliserons au cours de notre thèse ; par exemple la notion de langage logique du premier ordre, d'arbre de séquence, etc. Nous donnons aussi une brève introduction à deux de trois systèmes de preuve que nous utiliserons dans notre travail de thèse : la déduction naturelle et le calcul des sequents. Le troisième système de preuve que nous étudierons dans notre thèse, la logique dialogique, est présente en détail dans le troisième chapitre. Dans ce premier chapitre introductif nous ne présentons aucun nouveau résultat.

Philosophical foundations Le deuxième chapitre expose plus en détail les thèses et les théories philosophiques que nous avons esquissé dans cette introduction. Nous montrons comment les philosophes et logicien inférentialistes se sont confronté aux problèmes de la signification des constantes et énonce logiques. En particulier nous détaillons la relation entre la théorie inférentialiste de la signification et les deux systèmes de preuve introduits dans le premier chapitre : la déduction naturelle et le calcul des séquents. En particulier nous exposons la théorie philosophique qui veut que les règle d'introduction des constantes logiques en déduction naturelle définissent la signification des constantes logiques. Bien que cette théorie soit pertinente dans le cas de la logique intuitionniste nous montrons qu'elle rencontre des problèmes lorsqu'on s'intéresse à la signification de constantes logique de la logique classique. Nous essayons alors de montrer que le calcul des séquents —ou mieux une lecture dialogique du calcul des séquents— permets de donner une interprétation inférentialiste cohérente de la signification des constantes logique classique.

Logic and DiaLogical Games

Cette partie de notre manuscrit regroupe les résultats théoriques obtenue au cours de notre travail de thèse. Elle est composée de deux longs chapitres.

DiaLogical Games Le premier chapitre de cette partie offre une exposition détaille de la logique dialogique de Lorenzen et Lorenz [96, 97, 46]. D'abord nous définissons les objets centraux de la logique dialogique : les jeux et les stratégies. Ensuite nous présentons un système de calculs de séquents —que nous appelons SLK— et nous étudions ses propriétés. En particulier nous montrons que SLK est complet pour la logique classique du premier ordre : toute formula A qui est prouvable en SLK est valide et, vice-versa, toute formula valide A est prouvable en SLK. Ensuite nous montrons l'équivalence entre la notion classique de validité logique et la notion Dialogique de validité logique, c'est-à-dire, 'une formule A est valide en logique classique lorsqu'il existe une stratégie gagnante classique pour le Proposant'. Une stratégie gagnante étant une fonction qui spécifié, en fonction de l'historique du jeu, les coups que le Proposant doit jouer s'il veut être sûr de gagner le jeu.

Game Semantics for Constructive Modal Logic Dans le quatrième chapitre de notre thèse nous nous concentrerons sur sémantique dénotationnelle qui trouve son inspiration dans la logique dialogique : la sémantique des jeux. En particulier nous définissons une sémantique de jeux pour la variante constructive de la logique modale K (que nous appelons CK). Après avoir présenté un système de calcul des sequents qui est complète pour CK (LCK) nous définissons d'abord les jeux modaux, ensuite les stratégies gagnantes modales, que nous appelons CK-**WIS**s, (CK winning innocent strategies) et nous prouvons qu'elles peuvent être composées. Étant donné deux CK-**WIS**s pour les formules $A \supset B$ et $B \supset C$ nous pouvons construire une CK-**WIS** pour $A \supset C$. En suite, nous montrons comment interpréter les preuves de LCK par des CK-**WIS**. Finalement nous prouvons que notre sémantique des jeux pour CK est 'pleinement adéquat' (fully complete). Tout CK-**WIS** pour une formule A est l'interprétation d'une preuve de $\vdash A$ dans le calcul de sequents LCK.

Natural Language Applications of Proof Theory

Dans la troisième partie de notre thèse, nous nous concentrons sur certaines applications de la théorie de la preuve à l'analyse automatique du langage naturel et à la sémantique formelle.

Type Logical Grammars: a result about the syntax-semantic interface Dans le cinquième chapitre de notre thèse, nous nous intéressons à un problème naturel concernant le rapport entre analyse syntaxique d'une phrase et analyse sémantique d'une phrase dans les grammaires catégorielle. Nous présentons d'abord les grammaires catégorielles en détaillant comment on peut calculer la forme logique d'une phrase en langage naturel en utilisant la théorie de la preuve. Ensuite nous présentons le problème que nous traitons : la forme logique d'une phrase est une formule logique écrite dans le lambda calcul simplement typée. Pour produire la forme logique une grammaire catégorielle passe par une étape intermédiaire : l'analyse syntaxique de la phrase. L'analyse syntaxique de la phrase représente l'arbre grammatical de la phrase. Elle est exprimée par des lambda termes linéaires. Une même phrase peut avoir plusieurs analyses syntaxiques et forme logique différentes. On étudie alors une condition, que nous appelons dominance, que nous garanti que si une grammaire catégorielle attribue a une mémé phrase deux analyses syntaxiques différentes alors elle attribuera à cette phrase deux formes logiques différentes.

Textual Entailement Recognition and DiaLogical Games Dans le sixième chapitre nous nous concentrons sur le problème de la reconnaissance de l'inférence textuelle : étant donné une phrase ou un morceau de texte T et une phrase P nous essayons de juger si nous pouvons inférer à partir de la vérité de T la vérité de P. Nous traiterons ce problème en utilisant la Logique Dialogique ; en utilisant les grammaires catégorielles nous traduisons les phrases en formule logique ; ensuite pour vérifier qu'à partir de T nous pouvons déduire P nous construirons une stratégie gagnante pour la formule $F_T \supset F_P$ (F_T étant la forme logique de T et F_P celle de P). Nous utiliserons d'abord notre méthodologie pour résoudre certain exemple simple d'inférence textuelle issue du date-set FraCas. Ce dernier est expressément conçu pour un traitement logique de l'inférence textuelle. Ensuite nous modifierons légèrement le système de logique dialogique présenté dans le troisième chapitre : nous introduisons la possibilité pour les jouer de 'définir' la signification de certaines formules atomique pendant le jeu. Cette modification nous permet de traiter des inférences qui nécessitent une connaissance de la signification des mots par exemple, la phrase 'Jean consulte un psychiatre' implique 'Jean consulte un docteur'.

Dialogical Games for Anaphora and Ellipsis Resolution Dans le septième et dernier chapitre de notre thèse, on s'occupera d'un problème qui lie la reconnaissance d'inférence textuelle et un problème typique issue de la sémantique formelle : la résolution d'anaphores. En sémantique formelle le problème de la résolution d'anaphore est usuellement traité en utilisant des outils de la théorie des modèles. Dans ce chapitre, nous proposons une approche basée sur la théorie de la preuve. Nous présentons un quantificateur $\mathcal{A}$. La signification d'une formule $\mathcal{A}xF$ est donnée par ses règles d'attaque et défense dans un système Dialogique. Intuitivement la variable x pourra être instancié uniquement par un terme qui apparaît dans le contexte du jeu. Ce dernier est l'ensemble des formules qui sont affirmé par l'opposant **O** pendant le jeu. En utilisant ce système de logique dialogique on résoudra certains problèmes de reconnaissance d'inférence textuelle dont la résolution d'anaphore et ellipse est essentielle.

Introduction

Our thesis work lies at the crossroads of at least two disciplinary fields: on the one hand, mathematical logic and theoretical computer science, on the other hand, natural language processing, formal semantics of natural language and the philosophy of language. The main thread of this work is proof theory, which has offered us tools and methodologies to advance on a fundamental question of logic: what are the links between the notion of proof and that of linguistic meaning? More precisely: to understand the meaning of a statement A is to be able to provide a justification for the assertion of A?

Our interest in the philosophy of language as a logician and computer scientist is natural: isn't one of the central tasks of the philosophy of language to clarify or explain the concept of meaning and apply it to language? To achieve an explanation of the concept of meaning, philosophers have introduced the notion of 'theory of meaning'. Let us take up the words of Michael Dummett:

> according to one well known view, the best method of formulating the philosophical problems surrounding the concept of meaning and related notions is by asking what form should be taken by what is called 'a theory of meaning' for any one entire language; that is a detailed specification of the meanings of all the words and sentence forming operations of the language, yielding a specification of the meaning of every expression and sentence of the language. [41]

A theory of meaning is therefore nothing more than a specification of how we attribute meaning to the expressions of a language and how we understand those expressions. The language can of course be artificial, as would be C++, Scheme or a logical language, but it can also be natural, as are French or English. Although there are different theories of meaning, we can distinguish two main families:

- On the one hand we find the theories of meaning which have as their central concepts the concepts of *truth* and *reference*.
- On the other hand we find the theories of meaning which have as their central concept the concept of *inference*.

Referentialist theories of meaning

In a referentialist meaning theory, the concept of meaning is understood as follows: the meaning of an expression is the contribution that the expression makes to determining the truth value of a sentence in which it appears. Consider the two sentences:

(22) Boris Johnson is the prime minister of the United Kingdom.

(23) Mario Draghi is the prime minister of the United Kingdom.

we note that the first sentence is true while the second is false. According to a referentialist theory of meaning, this is because the two proper names 'Boris Johnson' and 'Mario Draghi' have different references: one name refers to the individual who is prime minister of the United Kingdom, while the other name does not refer to such an individual.

Let us consider the semantics, in the sense of model theory, of logical formulas. This analysis of the meaning of logical sentences (formulas) offers us one of the paradigmatic examples of a referentialist theory of meaning. Let $\mathcal{L}$ be a first order logical language.[6] The interpretation of the language terms is specified by means of a function associating to constants and variables elements of a structure. Suppose that each term t of the language corresponds to an element t^I of the interpretation structure. A predicate of the language is interpreted by a subset of the Cartesian product of the structure. Thus an atomic formula $P(t_1, \ldots t_n)$ is true in the interpreting structure if and only if the tuple $(t_1^I, \ldots t_n^I)$ belongs to the subset of the Cartesian product of the structure interpreting the predicate P. Then, when interpreting complex formulas, written with logical connectives and quantifiers, we specify the semantics by means of inductive clauses: a conjunction $A \wedge B$ is true in the interpretation if and only if the formula A and the formula B are true in this interpretation, etc.

[6] A precise definition of logical language, terms and interpretation structure will be given in chapter 1

Formal semantics is a branch of formal linguistics that constructs mathematical models of the meaning of linguistic expressions. The dominant approach in formal semantics is *referentialist*: the meaning of a sentence E is defined via the auxiliary concept of *logical form* of the sentence. The logical form of a sentence is a formula F belonging to some logical language $\mathcal{L}$. The logical form is supposed to capture one of the possible readings of the sentence, thus eliminating ambiguities. For example the sentence

(24) Every child eats a pizza.

May mean

(25) Given a child x, we can always find a pizza y such that x eats y.

Or

(26) There is a pizza y such that any child eats it.

The paraphrase 25 is captured by the formula

$$\forall x\,[\text{child}\,(x) \supset (\exists y\,\text{pizza}\,(y) \wedge \text{eat}\,(x, y))]$$

While the paraphrase 26 is formalized by the following:

$$\exists y\,[\text{pizza}\,(y) \wedge (\forall x\,\text{child}\,(x) \supset \text{eat}\,(x, y))]$$

Using the concept of logical form, we can define the meaning of a statement E, or better still of a possible reading of E, as being the class of models which satisfy the logical form corresponding to the reading considered. To take the example of 24 above, the meaning of the statement in relation to the reading 26 is given by the set

$$\{\mathfrak{M} \mid \mathfrak{M} \models \exists y\,[\text{pizza}(y) \wedge (\forall x\,\text{child}(x) \supset \text{eat}\,(x, y))]\}$$

As we have already anticipated, this type of semantics for natural language sentences is based on the concept of truth. The meaning of a sentence is the set of situations that make the sentence true. This type of approach to formal semantics is both simple and fruitful: it allows us to give a precise meaning to substantial fragments of natural language. However, this type of approach suffers from a number of shortcomings, which call into question its relevance as an analysis of the meaning of natural language sentences.

A first criticism is the poverty of this semantic analysis, at least from a certain point of view. Consider, for example, two sentences that are equivalent in terms of truth values:

(27) Cecile is tall and rich

(28) It is not the case that Cecile is not tall or not rich.

The two sentences can be 'translated' into the following two logical formulas

(29) $\mathit{tall}(\texttt{Cecile}) \wedge \mathit{rich}(\texttt{Cecile})$

(30) $\neg(\neg \mathit{tall}(\texttt{Cecile}) \vee \neg \mathit{rich}(\texttt{Cecile}))$

both formulas have the same set of models. In spite of this, the two formulas have significant differences in meaning, especially if we focus on inferential aspects. The inference that goes from the sentence "Cecile is tall and rich" to the sentence "Cecile is rich" can be considered an elementary inference. It is more difficult to consider the inference that goes from "It is not the case that Cecile is not tall or not rich" to the sentence "Cecile is rich" as an elementary inference.

But the worst is yet to come: let us consider any two logical theorems e.g., $(A \supset B) \supset (\neg B \supset \neg A)$ and $A \vee \neg A$. Since a logical theorem is by definition a formula which is true in any interpretation structure, both theorems have the same meaning. This is surprising, to say the least.

Inferentialist theories of meaning

On the other hand, we find inferentialist theories of meaning: an inferentialist theory of meaning denies that truth conditions play the main role in semantics. Instead, the fundamental concept which determines the meaning of expressions is that of *inference* or *justification*. In Robert Brandom's words:

> The standard way [of classical semantics] is to assume that one has a prior grip on the notion of truth, and use it to explain what good inference consists in [...] [I]nferentialist pragmatism reverses this order of explanation [...] It starts with a practical distinction between good and bad inferences, understood as a distinction between appropriate and inappropriate doings, and goes on to understand talk about truth as talk about what is preserved by the good moves. [15]

An inferentialist theory of meaning focuses on the semantic properties of statements. Inferential relations exist between sentences, not between sub-sentential expressions. (One cannot, for example, infer one noun from another). Thus, the inferentialist will not explain the semantic properties of singular terms, for example, in terms of the representational relations between those singular terms and the elements of the world; rather, he or she will explain what is distinctive about singular terms in terms of their role in certain types of inference.
Although there are disagreements, sometimes major, between the authors defending inferentialist semantics[7], the following two points are accepted, to the best of our knowledge, by any defender of the inferentialist viewpoint.

1. The meaning of an utterance is the knowledge that must be possessed (implicitly or explicitly) in order to understand the utterance;

2. This knowledge must in principle be observable in the interactions between the speaker, the listener, and the environment

Note that, for the point 1, the meaning of an utterance in a language (natural or artificial) cannot coincide with the object captured by the referentialist definition of meaning: speakers are only able to store a limited amount of data. By consequence, the knowledge needed to understand the meaning of the language itself should also be limited or, at least, could be enumerated and described recursively from a limited set of data and rules. In general, there is no finite way to enumerate the infinite number of models of a formula, nor to finitely enumerate the infinite number of individuals and relations in a single model. Thus, an inferentialist rejects the definition of the meaning of a sentence S as the set of models that satisfy one of the readings of S.

Inferentialism and argumentative dialogues

Referentialist approaches to meaning theory use —or are strongly inspired by—- methods of model theory, a branch of mathematical logic. In our thesis, we will use methods and concepts from proof theory. Proof theory is a branch of mathematical logic that deals

[7] The interested reader can consult the first chapters of Cozzo's book 'Meaning and Argument' [30] where a detailed comparison between different versions of inferentialism in the philosophy of language is presented

with the study of the formal properties of proofs, also called deductions or derivations. As seems natural, a semantic theory based on the concept of proof (or justification) is closely related to the study of the mathematical properties of proofs. There are in fact several approaches to the inferentialist meaning theory based on proof theory [53]. More specifically, they are based on a formal proof system invented by Gentzen and studied in depth by Prawitz: natural deduction [117]. We will choose a different approach, to implement the inferentialist theory of meaning: namely dialogical logic [96, 97]. Our choice is motivated by two distinct factors.

First, natural deduction "misbehaves" with respect to classical logic. For example, in a classical natural deduction system, there is no subformula property. Classical logic is an essential tool for the study of reasoning and inference in natural language and our thesis includes a study, albeit limited, of these topics.

Secondly, in our view, the link to a semantics based on the notion of argument is clearer in the paradigm of dialogical logic: an argument in favor of a statement is often developed when a critical audience, real or imagined, doubts the truth or plausibility of the proposition. In this case, in order to successfully assert the statement, a speaker, or proponent of it, must be capable of providing all the justifications that the audience is entitled to demand. Taking this idea seriously, an approximation of the meaning of a sentence in a given situation can be obtained by studying the *argumentative dialogues* that arise once the sentence is asserted in front of such a critical audience. This type of situation is captured —with a reasonable degree of approximation— by dialogical logic. In the dialogical logic framework, knowing the meaning of a sentence means being able to provide a justification of the sentence to a critical audience. Dialogical logic analyze the concept of validity of a formula F through the concept of winning strategy in a particular type of two-player game. This type of game is nothing more than an argumentative dialogue between a player, called *Proponent*, who affirms the validity of a certain formula F and another player, called *Opponent*, who contests its validity. The Proponent starts the argumentative dialogue by affirming a certain formula. The Opponent takes turns and attacks the claim made by the proponent according to its logical form. The Proponent can, depending on his previous assertion and on the form of the attack made by the Opponent, either defend his previous claim or counter-attack. The debate evolves following this pattern. The proponent wins the debate if he has the last word, i.e., the defense against one of the attacks made by the Opponent is a proposition that the opponent can not attack without violating the debate rules.

The basic idea of dialogical semantics can therefore be stated as follows:

> the meaning of a statement E is given by the form of the dialogue in which the speaker asserting E provides a critical interlocutor with all the justifications necessary to justify his assertion of E.

In the thesis, we will focus on a dialogical representation of formal proofs, and we will study in detail two logical systems whose formal proofs can be interpreted in terms of winning strategies for games: dialogical logic and game semantics.

Thesis's results

In this section we briefly describe the main results of our thesis work, which are of two styles. On the one hand, there are results on proof theory and on the other hand, results about the syntax or semantics of natural language.

Proof theory

In our thesis we explore two rather similar theories: dialogical logic and game semantics. We briefly outline both theories and present our own contributions.

Dialogical logic

The first article on dialogical logic was published in 1958 [96]. The ideas presented by Lorenzen were developed using the tools of game theory by his student Lorenz in his PhD thesis [97]. The base objects of dialogical logic are two-player games and are an idealization of an argumentative dialogue. The two players are called Proponent (or **P**) and Opponent (or **O**). The two players take turns and have different roles in the debate: **P** tries to construct a justification for a certain formula F, while **O** tries to construct a refutation of that formula F. Each connective and quantifier in the formula can be attacked by one player and defended by the other. Therefore, each complex formula is associated with questions according to its main connective and answers to these questions are expected.

A game is an alternated sequence (finite or infinite) of interventions made by **P** and **O**. The Player **P** makes the first intervention of the game. This first intervention is the assertion of the formula F. Any other intervention in the debate must be "justified" in the following sense: each intervention consists in asking a question about a formula asserted by the other player, or in answering one of the questions asked by the other player. Games are regulated by a series of restrictions, for example **P** can assert an atomic formula only if **O** has already asserted the same atomic formula. Each intervention of **O** is a reaction to the intervention of **P** which immediately precedes it. A strategy is a function that determines the next move of a player according to the previous moves, i.e. according to the history of the game. A strategy is said to be winning when the player who follows it wins the game whatever the history of the game is. We are interested in strategies that are winning for **P**. Our main contribution to the study of dialogic logic is the proof of the following statement

> Let $\mathcal{L}$ be a first-order language with terms and A a formula of $\mathcal{L}$. The formula A is valid (in the sense of classical logic) if and only if there exists a **P**-win strategy for A.

Our proof of this statement is constructive: given a winning strategy $\mathcal{S}$ for A we transform it into a proof of $\vdash A$ in the sequent calculus and, vice versa, we transform any proof of $\vdash A$ into a winning strategy. Although the system of classical dialogical logic we have outlined is presented by Lorenzen in the 1958 paper, to date there was no proof of this statement. There were proofs for propositional calculus, like the one proposed by Herbelin in his thesis [68] which our proof extends, or for restricted first-order languages whose only terms are constants [25]. We call SLK the sequent calculus we use to show the equivalence between classical validity and the existence of a **P**-winning strategy. SLK is a polarized sequent calculus system. If $\mathfrak{D}$ is a proof that ends with an application of a right introduction rule of the existential quantifier or the disjunction

$$\dfrac{\dfrac{\begin{matrix}\mathfrak{D}'\\ \vdots\end{matrix}}{\Gamma \vdash \underline{A[t/x]}, \exists x A, \Delta}\ \mathsf{R}}{\Gamma \vdash \exists x A, \Delta}\ \exists^{\mathsf{R}} \qquad\qquad \dfrac{\dfrac{\begin{matrix}\mathfrak{D}'\\ \vdots\end{matrix}}{\Gamma \vdash \underline{A_i}, A_1 \vee A_2, \Delta}\ \mathsf{R}}{\Gamma \vdash A_1 \vee A_2, \Delta}\ \vee_i^{\mathsf{R}}\ i \in \{1,2\}$$

then the underlined formula in the premise of the rule is the formula 'introduced' by rule R. The **P** winning strategies that we study in our thesis were introduced — in the case of intuitionistic dialogical logic — by Felscher [46]. These strategies — called *formal* by Felscher — have a peculiarity: the Proponent is forced to immediately respond to any attack directed against a disjunction or an existential formula. This type of restriction on strategies naturally corresponds to the polarization property sketched above.

Game Semantics

Game semantics is a form of denotational semantics inspired by dialogical logic. In a denotational semantic we interpret proofs of a logical system by mathematical objects. A denotational semantics has a peculiarity: proofs that are equal modulo a notion of reduction (usually cut-elimination or β-reduction) are interpreted by the same object. The first works in game semantics appeared in the early 1990s for linear logic [1, 76, 89, 13].

In our thesis, we develop a game semantics for the constructive variant of the modal logic K (called CK). The formulas of CK are written using the connectives $\supset$ and $\wedge$ and the modalities $\Box$ and $\Diamond$. The logic CK can be defined as the smallest set of formulas containing:

- any tautology of minimal logic [79];
- each instance of the axiom $\Box(A \supset B) \supset \Box A \supset \Box B$;
- each instance of the axiom $\Box(A \supset B) \supset \Diamond A \supset \Diamond B$;

and closed by:

- necessitation: if A belongs to CK then $\Box A$ also belongs;
- modus ponens : if A and $A \supset B$ belong to CK then B belongs too.

A complete sequent calculus system for this logic —that we call LCK—- is obtained by adding the following two rules to a 'standard' sequent calculus system for minimal logic.

$$\frac{\Gamma \vdash C}{\Box\Gamma \vdash \Box C}\,\mathsf{K}^{\Box} \qquad\qquad \frac{\Gamma, B \vdash C}{\Box\Gamma, \Diamond B \vdash \Diamond C}\,\mathsf{K}^{\Diamond}$$

where $\Box\Gamma = \Box A_1, \ldots, \Box A_n$ whenever $\Gamma = A_1, \ldots A_n$. The logic CK is well understood both from the model theoretic point of view and from the proof-theoretic point of view. Despite this, we count just one article in which the denotational semantics of CK is studied [11]. In this article the authors study the categorical structure of denotational models of CK. In our thesis, we construct a concrete denotational model for CK. In game semantics proofs are interpreted as winning strategies for two player games. The games are played over arenas: graphs representing the structure of formulas. The arena of a complex formula is recursively defined from those of atomic formulas. The arena of a formula A, denoted by $[\![A]\!]$, is a directed graph whose vertices are the atomic formulas contained in A and the occurrences of modalities in A, and whose edges are of two kinds. One kind of edge (denoted by $\rightarrow$) allows recovering the binary connectives of the formula A, while the other kind of edge (denoted by $\rightsquigarrow$) allows recovering the range of the modalities inside the formula A. Each vertex v of an arena $[\![A]\!]$ has a polarity. This polarity, positive or negative, is the same as that of the occurrence of the atomic formula, or modality, of A that decorates v. A positive vertex v is called *root* if there is no vertex w such that $v \rightarrow w$. A move is a vertex and it is a **O**-move if it is positive, it is a **P** move otherwise. Games are finite, alternated sequences of **O** and **P** moves.

- **O** always starts a game: the first move of a game is a root of the arena.
- Each move w of the game, except the first, is justified by a preceding move v of the other player: $w \rightarrow v$ in the arena;
- Each move in a game is either a propositional variable or a vertex $\diamond$;
- Each **O**-move is justified by the **P**-move that precedes it in the game;
- each move w of **P** carries the same label of the move v that immediately precedes it: if v is a propositional variable a (resp. a modality $\diamond$) then w is an occurrence of the same propositional variable a (resp. an occurrence of $\diamond$).

A game is won by one of the two players when the other player can no longer answer. A winning strategy is a tree where each branch is a game won by **P**. Bifurcations in this tree result from **O**-moves. To characterize the strategies corresponding to modal proofs, we will consider two additional constraints on the accepted moves of **P**. In the arena, each vertex w has a height: the number of modalities that have w in their scope. The first constraint requires that each **P**-move has the same height as the previous **O**-move. The second constraint permits us to define a notion of sub-game. Let w be a move in a game that is in the scope of some modality m of height k. We say that the modality m is new iff no move that precedes w is in the scope of m. Whenever **O** plays a move that is under the scope of a new modality, a sub-game starts. The sub-game ends whenever **O** plays a move w' that is the scope of no modality or when **O** plays a move that is under the scope of a positive modality $n \neq m$ of height k. In a modal winning strategy any sub-game has the following characteristics: let w be the first move of the sub-game and m its positive modality of height k, then

- each move v of the sub-game is in the scope of a modality of height k;
- if v is the sub-game and v is in the scope of a positive modality n of height k then $n = m$;
- If x belongs to the sub-game and x is within the scope of a negative modality n of height k, then for any move y belonging to another sub-game, y is not within the scope of n;
- if $m = \Diamond$ then there exists a negative modality n of height k such that $n = \Diamond$ and a move x under the scope of n and that belongs to the sub-game.

Using the above characterization, it is possible to interpret any proof $\mathfrak{D}$ of LCK by a modal winning strategy $\{\!\{\mathfrak{D}\}\!\}$. We then show that if $\mathfrak{D}$ reduces to $\mathfrak{D}'$ by applying the cut elimination algorithm then $\{\!\{\mathfrak{D}\}\!\} = \{\!\{\mathfrak{D}'\}\!\}$. Furthermore, we show that our semantics is fully complete, i.e.,

> Let $\{\!\{-\}\!\}$ be the function which associates to any LCK proof $\mathfrak{D}$ a modal winning strategy $\{\!\{\mathfrak{D}\}\!\}$; the function $\{\!\{-\}\!\}$ is surjective: for any winning strategy $\mathcal{S}$ there exists a proof $\mathfrak{D}$ such that $\mathcal{S} = \{\!\{\mathfrak{D}\}\!\}$.

We obtain the proof of the above statement using a sequentialization algorithm: we show how to transform a winning strategy for A into a proof of A.

Natural language syntax and semantics modelisation

Type-logical grammars

Type-logical grammars are a family of frameworks for the analysis of natural language based on proof theory. Given a natural language sentence $w_1, \dots w_n$ a computable function associates to each word w_i a formula $l(w_i)$ of linear, non-commutative, mutipliplicative intuitionistic logic (e.g., the Lambek calculus [91] or one of its variants). The grammaticality of the sentence corresponds to the provability, in the logic in question, of the judgment $l(w_1), \dots l(w_n) \vdash s$ (s represents the grammatical category of sentences). The same judgment can have several proofs, and each of these proofs corresponds to a distinct syntactic analysis of the sentence. By the Curry-Howard isomorphism, each syntactic analysis can be seen as a linear lambda-term, which provides instructions on how to compose the meaning of individual words. The individual words are also described by lambda terms called semantic lambda-terms. In the syntactic analysis, we substitute the free variables — corresponding to the lexical entries w_i — with the associated semantic lambda terms. We then obtain, by β-reduction, a representation of the logical meaning of the sentence. A natural question is then: if a sentence $w_1, \ldots w_n$ has at least two different syntactic analyses, do they give rise to two different logical representations? We show that if the question is naively formulated, these logical representation can be the same. However, considering a restricted class of semantic *lambda*-terms and a strong notion of difference between syntactic *lambda*-terms, the logical representation remain different. In particular, semantic lambda terms will λ_I-terms[8] of the shape

$$\lambda x_1, \ldots \lambda x_n k M_1 \cdots M_m$$

where k is a constant. To establish the difference between logical representations of a sentence, we define a *dominance* relation between atomic symbols of a λ-term. Intuitively, dominance allows us to reconstruct the 'application tree' of a λ-term. We show that dominance between constants is preserved by β-reduction, i.e.

> Let M be a λ_I typed term that contains two occurrences of constants k and k', such that k dominates k' in M. Suppose that M β-reduces to M'. Then each trace k_i of k is associated with a set of occurrences $k_i'^j$ of k' in M' and k_i dominates $k_i'^j$ in M'.

Therefore, if two syntactic analyses P_1 and P_2 of the same sentence define two different dominance relations between their free variables, (two syntactic analysis of the same sentence have the same free variables) then their respective logical form is different. Indeed, the logical form is obtained by substituting the free variables of P_1 and P_2 by the same semantic lambda-terms. The head symbol of a semantic term is a constant. If the variable w_1 dominates the variable w_2 in P_1 but not in P_2 (or vice versa) we can conclude that in the logical form of P_1 there will be a constant k_1 which dominates a constant k_2 and that the couple formed by k_1 and k_2 does not appear in the dominance relation of the logical form of P_2.

[8] the class of λ_I-terms is defined in chapter 1 section 1.9

Textual entailment recognition

One text A is said to entail another B when B is a consequence of A. A fairly standard task in automatic language processing, called textual entailment recognition, is to detect that one text is a logical consequence of another. For example, “Every duck’s leg is a bird’s leg” is logical consequence of “Every duck is a bird” whereas “Charles does not like pizza” is not logical consequence of “Every Italian likes pizza and Charles is not Italian”.

The FraCaS [28] data set was created in the 1990s. The objective was to develop a general framework for computational semantics. The data-set consists of textual inference recognition problems. Each problem consists of one or more assertions and a question with a ‘yes’ or ‘no’ answer.

(31) A Swede won a Nobel prize.

(32) Every Swede is a Scandinavian.

(33) Did a Scandinavian won a Nobel prize? [Yes]

We have applied dialogical logic and type-logical grammars to textual entailment recognition problems in the FraCas corpus. Type-Logical grammars produce a logical representation of sentences. Then the question-answer pairs of the data-set are translated into statements. In the previous case the pair formed by “Did a Scandinavian win a Nobel Prize” and “no” is translated into the sentence “Some Scandinavian won a Nobel Prize”. Consider again the above example. By translating sentences into logical formulas we obtain:

(34) A Swede won a Nobel prize
$\exists x_1 \, [\, \mathit{Swede}\,(x_1) \land (\exists x_2 \, \mathit{Nobel\text{-}prize}\,(x_2) \land \mathit{won}\,(x_1, x_2))]$

(35) Every Swede is a Scandinavian
$\forall x_3, [\, \mathit{Swede}\,(x_3) \supset \mathit{Scandinavian}\,(x_3)]$

(36) Some Scandinavian won a Nobel prize
$\exists x_4 \, [\, \mathit{Scandinavian}\,(x_4) \land (\exists x_5 \, \mathit{Nobel\text{-}prize}\,(x_5) \land \mathit{won}\,(x_4, x_5))]$

Finally, we show the existence (or non-existence, depending on the case) of a winning strategy for the formula $H_1 \land \ldots \land H_n \supset C$ where the H_i are the formulas corresponding to the sentences present in the data-set and C is the formula corresponding to the sentence obtained by using the question-answer pair present in the data set. We treat in detail some simple examples: e.g., quantifiers monotonicity.

Next we tackle a slightly more complex problem. We consider natural language texts that are in the textual entailment relation by virtue of the meaning of non-logical words e.g., any English speaker recognizes that the sentence 'Paul eats an apple' implies the sentence 'Paul eats a red fruit or Paul eats a green fruit'. This is because we can define the word apple as "the round fruit of a tree of the rose family, which typically has thin green or red skin and crisp flesh". To recognize this type of entailment relationship, we consider dialogical games in which players play modulo a certain set of definitions i.e., logical formulas of the form

$$\forall x_1, \dots \forall x_n (Q \iff A)$$

where Q is an atomic formula (let us call this formula the definiendum) and A is any formula (let us call this formula the definiens). In such games when one player asserts an atomic formula Q which is definiendum, the other player can ask to assert the definiens B of Q. Such games modulo definitions are called *unfolding* games. The notion of unfolding an atomic formula was first studied by Prawitz [117] in the context of natural deduction and developed by Dowek [40] in the context of deduction modulo.

Anaphora and ellipsis resolution

Defining exactly what is an anaphora and what is an ellipsis is a difficult task [83].

We can briefly define anaphora as a linguistic phenomenon whereby the interpretation of one occurrence of an expression depends on the interpretation of an occurrence of another expression. This characterization allows us to recognize that in the two sentences 'Gertrude eats an apple, it is delicious' the pronoun 'it' is anaphoric. The interpretation of the pronoun 'it' is the same as the interpretation of the noun 'an apple'. Let us call the linguistic expression on which the interpretation of an anaphora depends the antecedent. Humans are often able to resolve anaphora, i.e., find an appropriate antecedent for an anaphoric expression. Anaphora resolution is a difficult task in natural language processing. In a logical context, anaphora resolution is often handled by model-theoretic methods [80]. In our thesis we introduce a different approach to the problem of anaphora resolution. In particular, we approach it with the tools of proof theory. We introduce a new first-order quantifier (which we write $\mathcal{A}$) whose meaning is given by the rules that concern its 'use' in an argumentative dialogue. More concretely, we will define a first-order dialogical logic system (for a multisorted language) whose formulas are written with occurrences of the quantifier $\mathcal{A}$. In a dialogical game the formulas which are asserted by the player **O** have a different status from the formulas which are asserted by **P**. The formulas asserted by **O** form the hypotheses of the dialogue i.e., the formulas which can be regarded as accepted in the dialogue. Indeed, **O** can assert any atomic formula and thus defend himself against any attack blow from **P**. We can push this intuition: if the player **O** asserts that a certain individual k possesses a certain property A, then the existence of this individual is attested. Let us call the set of formulas affirmed by **O** in the game the *common ground* of the game. This common ground grows as the game progresses. We say that a constant k appears in the common ground iff the common ground contains a formula $B(k)$. The rules for attacking and defending the quantifier $\mathcal{A}$ are defined as follows: if a formula $\mathcal{A}xA$ is asserted in the game, then this assertion can be attacked only if at least one constant appears in the common ground. If one player attacks the formula, then the other player can defend himself by asserting $A(k)$ for some k that appears in the common ground. Intuitively the variables that are linked by an occurrence of $\mathcal{A}$ represent anaphoric expressions. We call this type of dialogical game "anaphoric games". We show that the set of formulas for which there is an anaphoric strategy is a consistent set: it does not contain all the formulas of the language. We then apply strategies for anaphoric games to the problem of textual entailment recognition that depends upon anaphoric resolution. We deal with some problems of the FraCas data-set e.g.,

(37) Smith attended a meeting.

(38) She chaired it.

(39) Did Smith chaired a meeting? [Yes]

In order to reduce the space of possible antecedents for an anaphoric expression, we use a multisorted logical language. It is natural, at least in English, to consider that some anaphoric expressions convey information about the ontological status of their antecedent. Moreover, it is natural that certain verbs, adjectives etc. can be applied only to certain kinds of entity. If, for example, we consider the sentence 'Smith took a train to Baltimore. It whistled at 12 o'clock' it is clear that 'it' refers to the train. The pronoun 'It' refers to an inanimate entity, and cities like Baltimore do not whistle. The sentences in the example below can therefore be represented by the following formulas:

(40) Smith attended a meeting.
$F_1 = \exists x_1{}^o\,(meeting\,(x_1) \wedge attended\,(\mathtt{smith}, x_1))$

(41) She chaired it.
$F_2 = \mathcal{A}\, x_2{}^h\, \mathcal{A}\, x_3{}^o\,(\,chaired\,(x_2, x_3))$

(42) Smith chaired a meeting.
$C = \exists x_4{}^o\,(chaired\,(\mathtt{smith}, x_4))$

we can thus build a winning strategy for the formula $F_1 \wedge F_2 \supset C$. We also consider the ellipsis resolution problem by reducing it to the anaphora resolution problem. Following the analysis of Davidson [35] we consider that each verb in a sentence introduces an event. Thus, the 'logical' form of the sentence 'Gertrude cut an apple in the bathtub at midnight' is 'there is an event, of cutting an apple, which happened in the bathtub at midnight and whose agent is Gertrude'. This sentence formalization strategy allows us to consider certain ellipses as anaphora: for example, in the sentence 'Gertrude met Maria, Bill too' 'too' refers to the event of meeting Maria;

Thesis's organization

Our thesis is divided into three parts. Each part consists of a maximum of three chapters. The work is organized according to a conceptual scheme that goes from abstract philosophical considerations through a logical-mathematical part to more concrete applications to the domains of automatic natural language processing. We will now present to the reader the contents of the three parts of our thesis.

Mathematical and Philosophical foundations

This part of the thesis contains two chapters and lays the logical and philosophical foundations of our work.

Mathematical preliminaries The first chapter of our thesis is dedicated to lay the mathematical foundations of our thesis work. We recall the definition of the logical and mathematical tools that we will use during our thesis; for example the notion of first order logic language, sequence tree etc. We also give a brief introduction to two of the three proof systems we will use in our thesis: natural deduction and the sequence calculus. The third proof system we will study in our thesis, dialogical logic, is presented in detail in the third chapter. In this first introductory chapter we do not present any new results.

Philosophical foundations The second chapter sets out in more detail the philosophical theses and theories that we have outlined in this introduction. We show how inferentialist philosophers and logicians have confronted the problems of the meaning of logical constants. In particular we detail the relationship between the inferentialist meaning theory and the two systems proof system introduced in the first chapter: natural deduction and the sequent calculus. We expose the philosophical theory according to which the meaning of the logical constant is given by their introduction rules in natural deduction. Although this theory is relevant in the case of intuitionistic logic, we show that it encounters problems when we are interested in the meaning of the classical logical constants. We then try to show that the sequent calculus — or better a dialogical lecture of the sequent calculus — allows us to give a coherent inferentialist interpretation of the meaning of classical logic constants.

Logic and DiaLogical Games

This part of our manuscript gathers the theoretical results obtained during our thesis work. It is composed of two long chapters.

DiaLogical Games The first chapter of this part offers a detailed exposition of Lorenzen and Lorenz dialogical logic [96, 97, 46]. First, we define the central objects of dialogical logic: games and strategies. Then we present a sequent calculus system — which we call SLK — and we study its properties. In particular, we show that SLK is complete for classical first-order logic: any formula A that is provable in SLK is valid and, vice versa, any valid formula A is provable in SLK. Then we show the equivalence between the classical notion of logical validity and the Dialogical notion of logical validity i.e., 'a formula A is valid in classical logic iff there exists a classical proponent winning strategy for A'. A winning strategy being a function that specifies, according to the history of the game, the moves that the Proponent must make if she wants to be sure to win the game.

Game Semantic for Constructive Modal Logic In the fourth chapter of our thesis we will focus on denotational semantics, which finds its inspiration in dialogical logic: game semantics. In particular, we define a game semantics for the constructive variant of the modal logic K (which we call CK). After presenting a complete sequent calculus for the logic CK (LCK), we first define modal games, then modal winning strategies — which we call CK-**WIS**s— and we prove that they can be composed. Given two CK-**WIS**s for $A \supset B$ and $B \supset C$ we can construct a CK-**WIS** for $A \supset C$. Then we show how to interpret the proofs of LCK by a CK-**WIS**. Finally, we prove that our game semantics for CK is 'fully complete'. Any CK-**WIS** for a formula A is the interpretation of a proof of $\vdash A$ in the sequent calculus LCK.

Natural Language Applications of Proof Theory

In the third part of our thesis we focus on some applications of proof theory to the automatic analysis of natural language and to formal semantics.

Type Logical Grammars: a result about the syntax-semantic interface In the fifth chapter of our thesis we focus on a natural problem concerning the relationship between the syntactic and semantic parsing of a sentence produced by a Type-logical grammar. We first introduce Type-logical grammars: we detail how one can compute the logical form of a sentence in natural language using proof theory. Then we present the problem we are dealing with: the logical form of a sentence is a logical formula written in the simply typed lambda calculus. To produce the logical form a categorial grammar goes through an intermediate step: the syntactic analysis of the sentence. The syntactic analysis represents the grammatical tree of the sentence. It is expressed by linear lambda terms. The same sentence can have several syntactic analysis and logical forms. We study a condition, which we call dominance, that guarantees that if a Type-logical grammar assigns to a same sentence two different syntactic analysis then it will assign to this sentence two different logical forms.

Textual Entailment Recognition and DiaLogical Games In the sixth chapter we focus on the problem of textual entailment recognition: given a sentence, or piece of text, T and a sentence P we try to judge whether we can infer from the truth of T the truth of P. We will deal with this problem using Dialogical Logic; by using Type-logical grammars we translate sentences into logical formulas; to check that we can infer P from T we construct a winning strategy for the formula $F_T \supset F_P$ (F_T being the logical form of T and F_P that of P). We will first use our methodology to solve some textual inference problem we took from FraCas. In the second part of this chapter, we slightly modify the dialogical logic system presented in the third chapter: we introduce the possibility for players to 'define' the meaning of certain atomic formulas during the game. This modification allows us to deal with inferences that require knowledge of the meaning of words to be solved e.g., the sentence 'John is consulting a psychiatrist' implies 'John is consulting a doctor'.

Dialogical Games for Anaphora and Ellipsis Resolution In the seventh and last chapter of our thesis, we will deal with a problem that links textual entailment recognition and a typical problem from formal semantics: anaphora resolution. In formal semantics the problem of anaphora resolution is usually treated using model-theoretic tools. In this chapter we propose an approach based on proof theory. We present a new first order quantifier $\mathcal{A}$. The meaning of a formula $\mathcal{A}xF$ is given by its attack and defense rules in a Dialogic system. Intuitively the variable x can only be instantiated by a term which appears in the context of the game. This context is the set of formulas that are asserted by the opponent **O** during the game. By using this dialogical logic system we will solve some problems of textual entailment recognition in which anaphora and ellipsis resolution occupy a central part.

Part I

Mathematical and Philosophical foundations

Chapter 1

Mathematical preliminaries

In this first chapter we introduce the formal object that we will be using throughout the dissertation. We first define the notions of first-order languages, sequences, and trees. We then define the model theoretic notion of (classical) interpretation of a first order formula. All this definitions are standard and can be found in any good introductory logic book such as [3, 132, 136]. We then introduce the two proof systems that will be used in our dissertation: the sequent calculus and natural deduction for the implicational fragment of intuitionist logic. We finally give an outline of the simply typed lambda calculus and we conclude by a very brief presentation of the Curry-Howard isomorphism.

1.1 First Order Language

Definition 1.1. *A first order language* $\mathcal{L}$ *is given by a signature* $\Sigma = (\mathfrak{C}, \mathfrak{F}, \mathfrak{R})$ *where the three sets* $\mathfrak{C}$, $\mathfrak{F}$ *and* $\mathfrak{R}$ *are at most countable and pairwise disjoint.*

- *The set* $\mathfrak{C}$ *is a set of constant symbols;*
- *the set* $\mathfrak{F}$ *is the set of function symbols. To each function symbol we associate a strictly positive natural number called* the arity *of the function symbol. The arity of a function symbol specifies the number of argument of the function symbol;*
- *The set* $\mathfrak{R}$ *is the set of relation symbols. We associate a natural number, its arity, to each relation symbol. The arity of a relation symbol is not necessarily positive. In this case we will speak of a propositional constant.*

Unless it is explicitly stated otherwise we consider that each set of relation symbols contains a proposition constant $\perp$ representing an arbitrary false proposition.

1.1.1 First Order Terms

Let $\mathcal{V}$ be a countable set of individual variable symbols. Individual variable symbols (or variables for short) will be denoted by $x, y, z \ldots$ (eventually indexed $x_1, x_2 \ldots$)

Definition 1.2. *The set $\mathcal{T}$ of (first order) terms is the smallest set which includes the set $\mathcal{V}$ of variables and the set $\mathfrak{C}$ of constants. This set is stable for the application of function symbols to terms: if $t_1, \ldots t_n$ are $n \geq 1$ terms and f is a function symbol with arity n then $f(t_1, \ldots t_n)$ is a term.*

Definition 1.3. *The depth $|t|$ of a term t is inductively defined as follows:*

1. *if $t \in \mathcal{V} \cup \mathfrak{C}$ then $|t| = 0$;*
2. $|f(t_1, \ldots t_n)| = max(|t_1|, \ldots |t_n|) + 1.$

.

Definition 1.4. *The set $FV(t)$ of free variables of a term t is inductively defined as follows:*

- *if $t \in \mathcal{V}$ then $FV(t) = \{t\}$;*
- *if $t \in \mathfrak{C}$ then $FV(t) = \emptyset$;*
- $FV(f(t_1, \ldots t_n)) = \bigcup_{i \leq n} FV(t_i)$.

The set of occurrences *of free variables of a term t is obtained by using the disjoint union $\uplus$ instead of the union $\cup$ in the inductive clause. A term t such that $FV(t) = \emptyset$ is said to be* closed.

Definition 1.5. *Let x be an individual variable and t, u be terms. We define the term $t[u/x]$ resulting from the substitution of the term u for all the free occurrences of the variable x in t by induction on the depth of t.*

1. *if t has depth* 0

- *if $t \in \mathfrak{C}$ then $t[u/x] = t$;*
- *if $t \in \mathcal{V}$*
  (a) if $t = x$ then $t[u/x] = u$;
  (b) if $t = y$ and $y \neq x$; then $t[u/x] = t$;

2. *if t has depth n then $t = f(t_1, \ldots t_k)$ and $t[u/x] = f(t_1[u/x], \ldots t_k[u/x])$.*

1.1.2 First Order Formulas

Unless otherwise stated, we consider the following logical constants: the binary logical constants $\wedge$ (conjunction),$\vee$ (disjunction), $\supset$ (implication) and the two quantifiers $\forall$ (universal quantifier), $\exists$ (existential quantifier). We will also use the parenthesis symbols '(' and ')' as well as the brackets '[' and ']'.

The set At of atomic formulas is the smallest set containing all expressions of the form $R(t_1, \ldots t_n)$ where $R \in \mathcal{R}$ is a relational symbol with arity $n \geq 0$, and for all $i \in \{1, \ldots n\}$ $t_i \in \mathcal{T}$.

Definition 1.6 (First order formulas)**.** *The set $\mathcal{F}$ of first order formulas is the smallest set containing the set At of atomic formulas and closed under the following operations:*

- *if $A, B \in \mathcal{F}$ then $A \star B \in \mathcal{F}$ for $\star \in \{\wedge, \vee, \supset\}$;*
- *if $A \in \mathcal{F}$ and $x \in \mathcal{V}$ then $QxA \in \mathcal{F}$ for $Q \in \{\forall, \exists\}$.*

The negation $\neg A$ of a formula A is defined as $\neg A = A \supset \bot$

Remark 1.1. *We can give another equivalent definition of the set of formulas. The set $\mathcal{F}$ of formulas is defined by the following grammar:*

$$\mathcal{F} = At \,|\, \mathcal{F} \wedge \mathcal{F} \,|\, \mathcal{F} \vee \mathcal{F} \,|\, \mathcal{F} \supset \mathcal{F} \,|\, \forall x \mathcal{F} \,|\, \exists x \mathcal{F}$$

where x runs through the set $\mathcal{V}$ of variables. The above expression can be read as follows: the elements of the set $\mathcal{F}$ are either atomic formulas, an expression $A \star B$ where A, B are elements of $\mathcal{F}$ and $\star \in \{\wedge, \vee, \supset\}$ or an expression QxB where $Q \in \{\forall, \exists\}$ $x \in \mathcal{V}$ and B is an element of $\mathcal{F}$.

Definition 1.7. *The depth $|A|$ of a formula A is inductively defined as follows*

- *$A \in At$ then $|A| = 0$;*
- *if A is of the form $B \star C$ for $\star \in \{\wedge, \vee, \supset\}$ then $|A| = max(|B|, |C|) + 1$;*
- *if A is of the form QxB with $Q \in \{\forall, \exists\}$ then $|A| = |B| + 1$;*

Definition 1.8. *The set $FV(A)$ of free variables of a formula A is inductively defined as follows*

- *if $A \in At$ and $A = \bot$ then $FV(A) = \emptyset$*
- *if $A \in At$ and A is of the form $P(t_1, \ldots t_n)$ then $FV(A) = \bigcup_{i \leq n} FV(t_i)$*
- *if A is of the form $B \star C$ with $\star \in \{\wedge, \vee, \supset\}$ then $FV(A) = FV(B) \cup FV(C)$*
- *if A is of the form QxB with $Q \in \{\forall, \exists\}$ then $FV(A) = FV(B)/\{x\}$*

The set of occurrences of free variables of a formula A is obtained by using the disjoint union $\uplus$ instead of the union $\cup$ in the preceding definition. A formula in which the set of free variables is empty is called closed.

Definition 1.9. *Let x be a variable, u a term and A a formula. The formula $A[u/x]$ resulting from the* capture avoiding *substitution of the term u for all the free-occurrences of the variable x in A is defined as follows:*

- *if $A \in At$ and $A = \bot$ then $A[u/x] = A$;*
- *if $A \in At$ and A is of the form $P(t_1, \ldots t_n)$ then $A[t/x] = P(t_1[u/x], \ldots t_n[u/x])$*
- *if A is of the form $B \star C$ with $\star\{\wedge, \vee, \supset\}$ then $A[u/x] = B[u/x] \star C[u/x]$*
- *if A is of the form QyB with $Q \in \{\forall, \exists\}$ then*
  1. *if $y \notin FV(u)$ and $y \neq x$ then $A[u/x] = Qy(B[u/x])$;*
  2. *if $y \in FV(u)$ and $y \neq x$ then $A[u/y] = Qz((B[z/y])[u/x])$ where z is a variable that does not appear in B nor in u;*
  3. *$A[u/x] = QyB$ otherwise.*

Definition 1.10. *Let A be a formula. The set $sub(A)^+$ of positive gentzen subformulas of A and the set $sub(A)^-$ of negative gentzen subformulas of A are defined as follows:*

- *if $A \in At$ then $Sub(A)^+ = \{A\}$ and $Sub(A)^- = \emptyset$;*
- *if A is of the form $B \star C$ with $\star \in \{\wedge, \vee\}$ then $sub(A)^+ = sub(B)^+ \cup sub(C)^+ \cup \{A\}$. $sub(A)^- = sub(B)^- \cup sub(C)^-$;*
- *if A is of the form $B \supset C$ then $sub(A)^+ = sub(C)^+ \cup \{A\}$. $sub(A)^- = sub(B)^-$;*
- *if A is of the form QxB with $Q \in \{\forall, \exists\}$ then $sub(A)^+ = \bigcup_{t \in \mathcal{T}} sub(B[t/x])^+ \cup sub(A)^+$. $sub(A)^- = \bigcup_{t \in \mathcal{T}} sub(B[t/x])^-$.*

  The set of (gentzen) subformulas of a formula A is equal to the union of the set of positive and negatives subformulas of A

1.2 Trees

All the content of this section can be found in [3]. In terms of graph theory, a tree can be defined as an acyclic and connected graph. If we choose a node of this graph and "pull it up", we obtain a "rooted" tree, i.e. an acyclic and connected graph in which one of the nodes has a particular status and is called a root. This structure can then naturally be presented as a particular ordered set, of which the root is the maximum. We recall that an order relation over a set A is a subset $\leq$ of the Cartesian product $A \times A$ (we will also say that $(A, \leq)$ is an ordered set) such that:

- for each $a \in A$ $a \leq a$;
- for each a, b and c in A if $a \leq b$ and $b \leq c$ then $a \leq c$;
- for each a and b in A if $a \leq b$ and $b \leq a$ then $a = b$.

The relation $\leq$ induces a strict order relation $<$ on A defined by $a < b$ iff $a \leq b$ and $a \neq b$. If for all $a, b \in A$ we have that either $a \leq b$ or $b \leq a$ then $\leq$ is *total*. Given $B \subseteq A$ an element $a \in A$ is an upper bound (resp. lower bound) of B if $b \leq a$ (resp. $a \leq b$) for all $b \in B$. An upper bound (resp. lower bound) a_0 of $B \subset A$ is a supremum (resp. infimum) if $a \leq a_0$ (resp. $a_0 \leq a$) for each lower bound (resp. upper bound) a of B. An order relation $\leq$ on a set A is *well-founded* when there is no infinite descending chain of elements of A i.e., there is no subset $\{a_i \mid i \in \mathbb{N}\}$ of A such that $a_{i+1} < a_i$ for each $i \in \mathbb{N}$.

Definition 1.11. *A tree $\mathfrak{T}$ is an ordered set $(T, \leq)$ where T is non-empty and such that:*

1. *for each x and y in T the set $\{x, y\}$ has a supremum;*
2. *for each $x \in T$ the set $\{z \in T \mid z \geq x\}$ is finite;*
3. *for each $x \in T$ the set $\{z \in T \mid z \geq x\}$ is totally ordered.*

By definition each tree $\mathfrak{T} = (T, \leq)$ has a maximal element. This maximal element will be called the *root* of the tree. Indeed fix an element $a_0 \in T$, and consider the set $\{b \in T \mid b \geq a_0\}$. By condition 2 and 3 this set is finite and has a bigger element c. Consider another arbitrary element $d \in T$. By condition 1 the set $\{d, c\}$ has a supremum s that must coincide with c. Elements of T will be called nodes or vertex. An element $(a, b) \in \leq$ will be called and edge whenever $a \neq b$ and there is no c such that $a < c < b$. A tree $\mathfrak{T} = (T, \leq)$ is well-founded iff $\leq$ is well-founded. Let a, b be element of T, b is a daughter of a iff $b < a$ and there is no c such that $b < c < a$. An element $a \in T$ that has no daughters is called *leaf* of $\mathfrak{T}$. A tree $\mathfrak{T}$ is finitely branching iff each element a of T has a finite number of daughters. A path in a tree is a totally ordered subset of the tree. A branch $\mathfrak{B}$ of $\mathfrak{T}$ is a maximal (for the $\subset$ relation) path of $\mathfrak{T}$. Remark that in virtue of condition 3 there is a branch, necessarily unique, from each element $a \in T$ to the root of $\mathfrak{T}$. The height of a node a is the largest number of edges in a path from a leaf node to a. The height of a tree is the height of the root of the tree.

1.3 Sequences

Informally, a sequence is an enumerated collection of objects in which repetitions are allowed and order matters. We formally define a sequence in the following way

Definition 1.12. *A sequence is a function from a subset A of the set $\mathbb{N}$ of natural numbers to a set S.*

Every element of a sequence is thus a pair (n, x) where n is a natural number and x an element of some set. We will denote an element (n, x) as x_n and call n an index. We will consider that if A is a subset of natural numbers that index a set S and that $n \in A$ then $m \in A$ for every $m < n$. Thus, the first element of a sequence will be indexed by 0 the second by 1 the third by 2 and so on. Sequences will be denoted by small case letters of the Greek alphabet σ, ρ, τ etc. If σ is a sequence we write σ_i to denote the element of σ indexed by i. The parity of σ_i is the parity of i, e.g., if $\sigma = M\,O\,T\,H\,E\,R$ then $\sigma_0 = M$ and has parity 0, $\sigma_3 = H$ and has parity 1. We will denote the empty sequence by ϵ. If $\sigma = x_0 x_1 \ldots x_n$ is a sequence and $\rho = y_0 y_1 \cdots y_m$ is another sequence, the expression $\sigma\rho$ will denote the sequence $x_0 x_1 \cdots x_n y_{n+1} \cdots y_{n+m}$. Given a sequence τ and a sequence ρ, τ is a prefix of ρ (written $\tau \sqsubseteq \rho$) iff there is a sequence σ such that $\rho = \tau\sigma$. If $\sigma \neq \epsilon$ then τ is a proper prefix of ρ. Given a sequence τ and a sequence ρ, τ is a suffix of ρ (written $\rho \ll \tau$) if there is a sequence σ such that $\rho = \sigma\tau$. If $\tau \neq \epsilon$ then τ is a proper suffix of ρ.

Let X be a set of sequences. The set X is said to be *prefix closed* whenever if $\rho \in X$ then $\sigma \in X$ for every $\sigma \sqsubseteq \rho$. Let X be a set of sequences, define the relation $\preceq \subseteq X \times X$ by

$$\rho \preceq \tau \equiv \tau \sqsubseteq \rho$$

The relation $\preceq$ is an order relation because $\sqsubseteq$ is also an order relation. Thus, if X is a prefix closed set of sequences the pair $(X, \preceq)$ is a tree in the sense of definition 1.11: the set X it contains at least the empty sequence ϵ. Each sequence ρ has a finite number of prefixes thus the set $\{\tau \in X \mid \tau \succeq \rho\}$ is finite for any $\rho \in X$. Moreover, the set $\{\tau \in X \mid \tau \succeq \rho\}$ is totally ordered. Finally, since X is prefix closed, given ρ and $\tau \in X$ we can always find some $\sigma \in \rho$ such that $\sigma \sqsubseteq \rho$ and $\sigma \sqsubseteq \tau$ (in the worst case $\sigma = \epsilon$) thus the set $\{\rho, \tau\}$ has always a supremum.

1.4 Semantic

In this section we briefly recall the standard definition of first order classical structure.

Definition 1.13. *Let $\mathcal{L}$ be a first order language over a signature $(\mathfrak{C}, \mathfrak{F}, \mathfrak{R})$. A first order structure $\mathfrak{M} = (M, -^{\mathfrak{M}})$ is a pair where M is an arbitrary non-empty set called the base of the structure and $-^{\mathfrak{M}}$ is a function such that*

- *$k^{\mathfrak{M}} \in M$ for all $\mathbf{k} \in \mathfrak{C}$;*
- *$f^{\mathfrak{M}}$ is a (total) function from M^n to M for all $f \in \mathfrak{F}$;*

- $P^{\mathfrak{M}} \subseteq M^n$ *for all* $P \in \mathfrak{R}$.

In order to define the notion of truth of a formula in a first order structure, we need to define what is the value in the structure of the free variables of the formula. We thus define the notion of evaluation.

Definition 1.14. *Let* $\mathfrak{M} = (M, (-)^{\mathfrak{M}})$ *be a first order structure for a first order language* $\mathcal{L}$*. An evaluation* e *is a function from the set* $\mathcal{V}$ *of variables of* $\mathcal{L}$ *to the base set* M *of* $\mathfrak{M}$*. If* e *is a valuation and* a *an element of* M*, we denote by* $\mathrm{e}[a := x]$ *the valuation* e' *such that* $\mathrm{e}'(x) = a$ *and* $\mathrm{e}'(y) = \mathrm{e}(y)$ *for any variable* y *different from* x*.*

Definition 1.15. *Let* $\mathfrak{M}$ *be a first order structure for a first order language* $\mathcal{L}$ *and let* $\mathcal{T}$ *be the set of terms of* $\mathcal{L}$*. The value of the term* t *in* $\mathcal{L}$ *with respect to the evaluation e, denoted* $[\![t]\!]_{\mathrm{e}}$*, is defined as follows*

- $[\![x]\!]_{\mathrm{e}} = \mathrm{e}(x)$ *for any variable* x*;*
- $[\![k]\!]_{\mathrm{e}} = k^{\mathfrak{M}}$ *for any constant* k*;*
- $[\![f(t_1, \dots t_n)]\!]_{\mathrm{e}} = f^{\mathfrak{M}}([\![t_1]\!]_{\mathrm{e}}, \dots [\![t_n]\!]_{\mathrm{e}})$

We can now define the notion of value of a formula in a structure. The value (or reference) of the formula will be a truth value: either true or false.

Definition 1.16. *Let* $\mathfrak{M} = (M, (-)^{\mathfrak{M}})$ *be a first order structure for a language* $\mathcal{L}$*. Let* F *be a formula of* $\mathcal{L}$*. The value of the formula* F *in the structure* $\mathfrak{M}$ *with respect to an evaluation* e*, noted* $[\![F]\!]_{\mathrm{e}}$*, is an element of the set* $\{0, 1\}$ *and is defined by induction on the depth of* F *as follows:*

$$\begin{aligned}
[\![\bot]\!]_{\mathrm{e}} &= 0 \\
[\![P(t_1, \dots t_n)]\!]_{\mathrm{e}} &= 1 \textit{ iff } ([\![t_1]\!]_{\mathrm{e}}, \dots [\![t_n]\!]_{\mathrm{e}}) \in P^{\mathfrak{M}} \\
[\![\neg A]\!]_{\mathrm{e}} &= 1 \textit{ iff } [\![A]\!]_{\mathrm{e}} = 0 \\
[\![A \wedge B]\!]_{\mathrm{e}} &= 1 \textit{ iff } [\![A]\!]_{\mathrm{e}} = 1 \textit{ and } [\![B]\!]_{\mathrm{e}} = 1 \\
[\![A \vee B]\!]_{\mathrm{e}} &= 1 \textit{ iff } [\![A]\!]_{\mathrm{e}} = 1 \textit{ or } [\![B]\!]_{\mathrm{e}} = 1 \\
[\![A \supset B]\!]_{\mathrm{e}} &= 0 \textit{ iff } [\![A]\!]_{\mathrm{e}} = 1 \textit{ and } [\![B]\!]_{\mathrm{e}} = 0 \\
[\![\exists x A]\!]_{\mathrm{e}} &= 1 \textit{ iff there is an } a \in M \textit{ such that } [\![A[a := x]]\!]_{\mathrm{e}} = 1 \\
[\![\forall x A]\!]_{\mathrm{e}} &= 1 \textit{ iff for any } a \in M [\![A[a := x]]\!]_{\mathrm{e}} = 1
\end{aligned}$$

if $\mathfrak{M}$ is a first order structure and F a formula we will write $\mathfrak{M}, e \models F$ whenever $[\![F]\!]_{\mathrm{e}} = 1$ and say that $\mathfrak{M}$ is a model of F. Let Γ be a set of formulas. We have that $\mathfrak{M}\mathrm{e} \models \Gamma$ iff $\mathfrak{M}, \mathrm{e} \models A$ for any $A \in \Gamma$.

Definition 1.17 (Classical Validity, logical consequence)**.** *Let $\mathcal{L}$ be a first order language and F a formula of $\mathcal{L}$. The formula F is valid (noted $\models F$) iff for any structure $\mathfrak{M}$ and valuation* e *we have that $\mathfrak{M}, \mathrm{e} \models F$. We say that a formula A is logical consequence of a set of formulas Γ iff for any structure $\mathfrak{M}$ and valuation e we have that*

$$\mathfrak{M}, e \models \Gamma \textit{ implies } \mathfrak{M}, e \models A$$

1.5 Proofs

In the sections to follow we present the proof formalism that we will exploit in our dissertation: the sequent calculus and natural deduction. Both were invented (or discovered as the reader prefer) by Gerhard Gentzen in is PhD dissertation and both system offers a representation of deductive arguments as trees generated by a set of rules.

1.6 Sequent Calculus

A sequent is an expression $\Gamma \vdash \Delta$ where Γ and Δ are finite (possibly empty) lists of formulas. The formulas in Γ are called the antecedents or hypothesis of the sequent, and the formulas in Δ are called the consequents of the sequent. The intuitive interpretation of a sequent

$$A_1, \ldots, A_n \vdash B_1, \ldots, B_m$$

is that if the conjunction of the antecedents is true then the disjunction of the consequents is true i.e., the sequent above corresponds to the formula

$$A_1 \wedge \cdots \wedge A_n \supset B_1 \vee \cdots \vee B_m$$

A sequent calculus is a formalism to construct formal deductive arguments. The arguments, called derivations or proofs, are obtained through the application of inference rules. Inference rules have a (possibly empty) list of sequents as premise and a sequent as conclusion. A rule is represented, schematically, as follows:

$$\frac{\Gamma_1 \vdash \Delta_1 \qquad \cdots \qquad \Gamma_n \vdash \Delta_n}{\Sigma \vdash \Pi}\ \mathrm{r}$$

where r is the name of the rule, the sequents $\Gamma_i \vdash \Delta_i$ for $i \in \{1, \ldots n\}$ are called the *premises* of the rule and the sequent $\Sigma \vdash \Pi$ is called the *conclusion* of the rule. A rule is classified according to the number of its premises. In this thesis we will work with rules with zero, one or two premises called, respectively, initial rules, unary rules and binary rules. Proofs in the sequent calculus are trees of sequents that are constructed from a given set of rules. The leaves of the proof-trees are obtained from initial rules. There exists numerous sequent calculus systems for many logics [1], here we present the sequent calculus system introduced by Gerhard Gentzen [58]: LK. The rules of the sequent calculus LK are shown in table 1.1

Definition 1.18. *A derivation (or proof) $\mathfrak{D}$ in LK of a sequent $\Gamma \vdash \Delta$ is a finite tree of sequents constructed according to the rules of table 1.1. The root of the tree, also called conclusion, is $\Gamma \vdash \Delta$, the leaves of the tree are instances of the Id-rule or of the $\bot^{\mathrm{L}}$-rule. In the $\exists^{\mathrm{L}}$ and $\forall^{\mathrm{R}}$ rules, the variable y does not appear in the sequent that is the conclusion of the rule, and it is called* proper parameter *or* eigenvariable*. The formulas that appear both in the conclusion and in the premises of a rule (the formulas that are not concerned by the application of the rule) are called context or side formulas.*

Rules in the sequent calculus can be divided in the following manner

Structural rules: these inference-rules do not refer to any logical connective, but instead operates on the 'shape' of the antecedents or succedents of a sequent. In table 1.1 they are shown above the cut-rule. Structural rules mimic intended meta-theoretic properties of the logic e.g., the structural rules of weakening W^{L} and W^{R} reflect the fact the consequence relation of classical logic is *monotone*.

Logical Rules: these inference-rules deals with the connectives and quantifier of a logic: using inference rules, we can introduce a new formula on the left or on the right of $\vdash$-symbol in a sequent. In table 1.1 logical rules are shown below the cut-rule. Remark that every logical symbol as some dedicated left introduction rules and right introduction rules and that those rules does not depend on the rules for another logical symbol.

[1] The interested reader can consult [132] for a presentation of some sequent calculus system for classical, intuitionistic, linear and modal logic

$$\frac{}{A \vdash A}\ Id$$

$$\frac{\Gamma, A, B, \Sigma \vdash \Delta}{\Gamma, B, A, \Sigma \vdash \Delta}\ \mathsf{E}^{\mathsf{L}} \qquad \frac{\Gamma \vdash \Sigma, A, B, \Delta}{\Gamma \vdash \Sigma, B, A, \Delta}\ \mathsf{E}^{\mathsf{R}}$$

$$\frac{\Gamma \vdash \Delta}{\Gamma, A, \vdash \Delta}\ \mathsf{W}^{\mathsf{L}} \qquad \frac{\Gamma \vdash \Delta}{\Gamma \vdash A, \Delta}\ \mathsf{W}^{\mathsf{R}}$$

$$\frac{\Gamma, A, A \vdash \Delta}{\Gamma, A \vdash \Delta}\ \mathsf{C}^{\mathsf{L}} \qquad \frac{\Gamma \vdash A, A, \Delta}{\Gamma \vdash A, \Delta}\ \mathsf{C}^{\mathsf{R}}$$

$$\frac{\Gamma \vdash A, \Delta \qquad \Sigma, A \vdash \Pi}{\Gamma, \Delta \vdash \Sigma, \Pi}\ \mathsf{cut}$$

$$\frac{\Gamma, A \vdash \Delta}{\Gamma \vdash \neg A, \Delta}\ \neg^{\mathsf{L}} \qquad \frac{\Gamma \vdash A, \Delta}{\Gamma, \neg A \vdash \Delta}\ \neg^{\mathsf{R}}$$

$$\frac{}{\bot \vdash}\ \bot^{\mathsf{L}} \qquad \frac{\Gamma \vdash \Delta}{\Gamma \vdash \bot, \Delta}\ \bot^{\mathsf{R}}$$

$$\frac{\Gamma, A \vdash \Delta}{\Gamma, A \wedge B \vdash \Delta}\ \wedge^{\mathsf{L}}_1 \qquad \frac{\Gamma, B \vdash \Delta}{\Gamma, A \wedge B \vdash \Delta}\ \wedge^{\mathsf{L}}_2 \qquad \frac{\Gamma \vdash A, \Delta \qquad \Gamma \vdash B, \Delta}{\Gamma \vdash A \wedge B, \Delta}\ \wedge^{\mathsf{R}}$$

$$\frac{\Gamma, A \vdash \Delta \qquad \Gamma, B \vdash \Delta}{\Gamma, A \vee B \vdash \Delta}\ \vee^{\mathsf{L}} \qquad \frac{\Gamma \vdash A, \Delta}{\Gamma \vdash A \vee B, \Delta}\ \vee^{\mathsf{R}}_1 \qquad \frac{\Gamma \vdash B, \Delta}{\Gamma \vdash A \vee B, \Delta}\ \vee^{\mathsf{R}}_2$$

$$\frac{\Gamma \vdash A, \Delta \qquad \Sigma, B \vdash \Pi}{\Gamma, \Sigma, A \supset B \vdash \Delta, \Pi}\ \supset^{\mathsf{L}} \qquad \frac{\Gamma, A \vdash B, \Delta}{\Gamma \vdash A \supset B, \Delta}\ \supset^{\mathsf{R}}$$

$$\frac{\Gamma, A[y/x] \vdash \Delta}{\Gamma, \exists x A \vdash \Delta}\ \exists^{\mathsf{L}} \qquad \frac{\Gamma \vdash A[t/x], \Delta}{\Gamma \vdash \exists x A, \Delta}\ \exists^{\mathsf{R}}$$

$$\frac{\Gamma, A[t/x] \vdash \Delta}{\Gamma, \forall x A \vdash \Delta}\ \forall^{\mathsf{L}} \qquad \frac{\Gamma \vdash A[y/x], \Delta}{\Gamma \vdash \forall x A, \Delta}\ \forall^{\mathsf{R}}$$

Table 1.1: The sequent calculus LK

Theorem 1.1. *The sequent calculus LK is a sound and complete proof system for classical first order logic: for any formula F, there is a LK-derivation $\mathfrak{D}$ of $\vdash F$ if and only if F is valid.*

Proof. A detailed proof of this classic result can be found in [54]. □

If we give a closer look to the rules of the sequent calculus LK we notice that the only inference rule in which the formulas in the premises of the rule are not Gentzen sub-formulas of the formulas in the conclusion of the rule is the cut-rule. Derivation in which the cut-rule is not used are called *normal* and enjoys an important property called the *sub-formula property*. In a derivation enjoying the sub-formula property, all formulas appearing in some sequent of the derivation are (Gentzen) sub-formulas of the conclusion sequent of the derivation. Such proofs are sometimes also called analytical. A calculus is called analytical in this sense that, given an arbitrary theorem, by analyzing its logical structure we have hope to succeed in a bottom-up search for its proof.

Theorem 1.2. *In the sequent calculus LK the* cut*-rule is redundant: if there is a derivation $\mathfrak{D}$ of the sequent $\Gamma \vdash \Delta$ that contains instances of the* cut*-rule then there is a derivation $\mathfrak{D}'$ of $\Gamma \vdash \Delta$ in which no sequent is obtained by using the* cut*-rule*

There are two ways of proving this important theorem. One, the semantical way, is to prove that every valid sequent has a normal proof. This style of proof is given in the already quoted [54]. Another way, the algorithmic one, is to define a series of transformation on derivation that permits, if applied in a certain order, to transform an arbitrary proof of a sequent in a cut-free proof of the same sequent. This way of proving the theorem was the one originally studied by Gentzen [58] and modern presentations of this proof can be found by the reader in [3, 63]. This kind of proof will be presented in detail for a sequent calculus for the modal logic CK in the third chapter of our dissertation.

We can now refine the statement of theorem 1.1

Theorem 1.3. *The sequent calculus LK without the* cut*-rule is a sound and complete proof system for classical first order logic: for any formula F, there is* cut*-free LK-derivation $\mathfrak{D}$ of $\vdash F$ if and only if F is valid.*

1.7 Variations on the sequent calculus

The sequent calculus LK that we have briefly introduced in the previous section is one of the many sequent calculi that are cut-free sound and complete for first order classical logic. In this section we describe some variants on the sequent calculus that will be used in our dissertation.

1.7.1 Intuitionistic Logic

To obtain a sequent calculus system that is sound and complete for first order intuitionistic logic, one can simply define a sequent to be an expression $\Gamma \vdash \Delta$ where Γ and Δ are finite list of formulas such that the list Δ contains at most one formula and modify the rules of LK accordingly. Such a sequent calculus is called LJ.

1.7.2 Atomic identity rule

In the *Id*-rule the formula A is an arbitrary formula. One can restrict the rule, so that A is an atomic proposition different from $\bot$ without changing the set of sequents that are LK provable. In fact we can always replace an instance of an *Id*-rule with conclusion $\bot \vdash \bot$ with an instance of the $\bot^{\mathsf{L}}$-rule followed by an instance of the $\bot^{\mathsf{R}}$-rule. Moreover, we can expand up-to atomic formulas the non-atomic instances of the *Id*-rules e.g.,

$$\dfrac{}{\neg A \vdash \neg A}\,Id \qquad\qquad \dfrac{\dfrac{\begin{array}{c}\vdots\\ A \vdash A\end{array}}{A, \neg A \vdash}\,\neg^{\mathsf{L}}}{\neg A \vdash \neg A}\,\neg^{\mathsf{R}}$$

$$\dfrac{}{B \supset C \vdash B \supset C}\,Id \qquad\qquad \dfrac{\dfrac{\dfrac{\begin{array}{c}\vdots\\ B \vdash B\end{array} \qquad \begin{array}{c}\vdots\\ C \vdash C\end{array}}{B, B \supset C \vdash C}\,\supset^{\mathsf{L}}}{B \supset C, B \vdash C}\,\mathrm{E}^{\mathsf{L}}}{B \supset C \vdash B \supset C}\,\supset^{\mathsf{R}}$$

$$\dfrac{}{B \wedge C \vdash B \wedge C}\,Id \qquad\qquad \dfrac{\dfrac{\begin{array}{c}\vdots\\ B \vdash B\end{array}}{B \wedge C \vdash B}\,\wedge^{\mathsf{L}}_{1} \qquad \dfrac{\begin{array}{c}\vdots\\ C \vdash C\end{array}}{B \wedge C \vdash C}\,\wedge^{\mathsf{L}}_{2}}{B \wedge C \vdash B \wedge C}\,\wedge^{\mathsf{R}}$$

1.7.3 Negation

By defining the negation of a formula $\neg A$ as $A \supset \bot$ (as we have done in subsection 1.1.2) the rules $\neg^{\mathrm{L}}$ and $\neg^{\mathrm{R}}$ became redundant. Consider the subsystem of LK that contains all the rules but those of the $\neg$ connective. Suppose that we have a derivation $\mathfrak{D}_1$ of $\Gamma, A \vdash \Delta$ and a derivation $\mathfrak{D}_2$ of $\Gamma \vdash A, \Delta$. We can construct the following:

$$\dfrac{\dfrac{\dfrac{\begin{array}{c}\mathfrak{D}_1\\ \vdots\\ \Gamma, A \vdash \Delta\end{array}}{\Gamma, A \vdash \bot, \Delta}\,\bot^{\mathsf{R}}}{\Gamma \vdash A \supset \bot, \Delta}\,\supset^{\mathsf{R}}}{\Gamma \vdash \neg A, \Delta}\,\equiv \qquad\qquad \dfrac{\dfrac{\begin{array}{c}\mathfrak{D}_2\\ \vdots\\ \Gamma \vdash A, \Delta\end{array} \qquad \dfrac{}{\bot \vdash}\,\bot^{\mathsf{L}}}{\Gamma, A \supset \bot \vdash \Delta}\,\supset^{\mathsf{L}}}{\Gamma, \neg A \vdash \Delta}\,\equiv$$

in both derivation we have substituted the formula $A \supset \bot$ by its definendium $\neg A$. We used the $\equiv$-symbol to underline such substitution in the proof-tree

1.7.4 Additive vs Multiplicative

Rules for the logical connectives exists in two formats: the multiplicative and the additive format e.g., the multiplicative rule $\supset^{\mathsf{R}}$ of LK

$$\dfrac{\Gamma \vdash A, \Delta \qquad \Sigma, B \vdash \Pi}{\Gamma, \Sigma, A \supset B \vdash \Delta, \Pi}\ \supset^{\mathsf{L}}$$

in which the side formulas of the two premises does not need to be equals and the side formulas of the conclusion sequent are obtained by concatenation of the list of side formulas of the premises, can be replaced by the following additive version of the rule

$$\dfrac{\Gamma \vdash A, \Delta \qquad \Gamma, B \vdash \Delta}{\Gamma, A \supset B \vdash \Delta}\ \supset^{\mathsf{L}}$$

in which the side formulas in the premises and in the conclusion of the rule are exactly the same. The multiplicative and additive version of the $\supset^{\mathsf{L}}$-rule are equivalent modulo the structural rules. If one has a derivation $\mathfrak{D}_1$ of the premises of the multiplicative version of the rule, one can construct a derivation of the conclusion of the multiplicative version of the rule using the additive version of the rule and the weakening rules. Vice-versa, if one has a derivation $\mathfrak{D}_2$ of the premises of the additive version of the rule one can construct a derivation of the conclusion of the additive rule using the multiplicative version of the rule and the contraction rules.

$$\dfrac{\begin{array}{c}\mathfrak{D}_{1.1}\\ \vdots\\ \Gamma \vdash A, \Delta \\ \hline\hline \Gamma, \Sigma, A \vdash \Delta, \Pi\end{array} \qquad \begin{array}{c}\mathfrak{D}_{1.2}\\ \vdots\\ \Sigma, B \vdash \Pi \\ \hline\hline \Gamma, \Sigma, B \vdash \Delta, \Pi\end{array}}{\Gamma, \Sigma, A \supset B \vdash \Delta, \Pi}\ \supset^{\mathsf{L}} \qquad\qquad \begin{array}{c} \dfrac{\begin{array}{c}\mathfrak{D}_{2.1}\\ \vdots\\ \Gamma \vdash A, \Delta\end{array} \qquad \begin{array}{c}\mathfrak{D}_{2.2}\\ \vdots\\ \Gamma, B, \vdash \Delta\end{array}}{\Gamma, \Gamma, A \supset B \vdash \Delta, \Delta}\ \supset^{\mathsf{L}} \\ \hline\hline \Gamma, A \supset B \vdash \Delta \end{array}$$

In the derivation on the left-side the multiple lines designates multiple application of the weakening rules and exchange rules. In the derivation on the right-side they indicate multiple applications of the contraction and exchanges rules. There exists multiplicative and additive versions of all the rules for the logical connectives. Modulo the structural rules, they are always equivalent in the sense specified above. A sensible difference between the two format of rules appears if we drop some of the structural rules. In particular, if we drop the structural rules of weakening and contraction the additive and multiplicative rules defines *different* connectives. This is one of the main ideas of linear logic.

1.7.5 Absorbing the Exchange rules

The left and right exchange rules E^L and E^R are needed in LK because we have defined a sequent to be an expression $\Gamma \vdash \Delta$ in which Γ and Δ are finite list of formulas. Since the order of the hypothesis in a sequent does not matter for logical consequence, we are forced to introduce the exchange rules to freely permute formulas in a list. If we want to get rid of exchange rules without losing essential properties of derivations, we can resort to two alternatives:

1. We define a sequent as an expression $\Gamma \vdash \Delta$ where Γ and Δ are finite multiset of formulas. A multiset is, informally, a set in which repetition counts or, alternatively, a list in which the order does not count. Formally a multiset is a pair (Γ, m) where Γ is a set and $m : \Gamma \to \mathbb{N}^+$ is a function.

2. Following an idea of Herbelin [68], we define a sequent as an expression $\Gamma \vdash \Delta$ where Γ and Δ are finite set of named formulas. A named formula is a pair (A, n) where A is a formula and n is an arbitrary name. Of course each formula in Γ and Δ have a name that is different from the name of all other formulas in Γ and Δ.

Both alternatives have their advantages and disadvantages, and both will be used in our PhD dissertation.

1.7.6 Absorbing the Weakening rules

To avoid using the weakening rule, one possibility is to push all the occurrences of the weakening rules to the *id* and $\bot^L$ rules and to integrate them there by modifying the *Id* and $\bot^L$ rules in this way:

$$\frac{}{\Gamma, A \vdash A, \Delta}\ Id \qquad\qquad \frac{}{\Gamma, \bot \vdash \Delta}\ \bot^L$$

1.7.7 Absorbing the Contractions rules

it is possible to absorb contraction rules at the level of introduction rules (right and left) for logical connectives. To do so, it is sufficient to modify them so that the formula introduced in the conclusion of a rule already appears in the rule's premises e.g., the rules for the $\exists$ and the additive rules for the $\supset$ connective are modified in the following way

$$\frac{\Gamma, \exists x A, A[x/y] \vdash \Delta}{\Gamma, \exists x A \vdash \Delta} \exists^{\mathsf{L}} \qquad \frac{\Gamma \vdash A[t/x], \exists x A, \Delta}{\Gamma \vdash \exists x A, \Delta} \exists^{\mathsf{R}}$$

$$\frac{\Gamma, A \supset B \vdash A, \Delta \qquad \Gamma, A \supset B, B \vdash \Delta}{\Gamma, A \supset B \vdash \Delta} \supset^{\mathsf{L}} \qquad \frac{\Gamma, A \vdash B, A \supset B, \Delta}{\Gamma \vdash A \supset B, \Delta} \supset^{\mathsf{R}}$$

1.8 Natural Deduction

As it is well explained by Schroeder-Heister [127] natural deduction is based on at least three major ideas:

- *Discharge of hypotheses*: hypotheses can be discharged or eliminated in the course of a proof. Natural deduction deals with proof depending on hypotheses, and this results in a deductive system enjoying important structural properties.
- *Separation*: each primitive rule scheme only concerns a single logical operator (connective or quantifier).
- *Introductions and eliminations*: Each logical operator is given with two rules. Introduction rules for an operator $*$ allow one to infer a sentence whose main logical operator is $*$. Elimination rules for an operator $*$ allow one to infer consequences from a sentence with $*$ as its main operator.

Natural deduction proofs are *finite trees* (with some additional information):

- The leaves, the root and internal nodes are labeled with formulas.
- The root formula is called the conclusion of the proof.
- Branching are denoted by a horizontal line and are indexed by the name of the rule. The formulas above the line are said to be the premise(s) of the rule, and the formula below the line/rule is called the conclusion of the rule.
- The formulas that label the leaves of the tree are called the hypotheses of the proof. There are two kinds of hypotheses. Ones are said to be active or undischarged, others

are said to be discharged or canceled. The discharged hypotheses are written between square brackets, possibly with an index specifying the rule which discharged them: $[H]_i$.

- A tree whose active leaves are $H_1, \ldots, H_n$ and whose root is C is a proof of C from the active (or undischarged) hypotheses $H_1, \ldots, H_n$,

We shall use the graphical notation

$$\begin{array}{c} \vdots \\ C \end{array}$$

to denote a natural deduction proof of C. Natural deduction proofs will be written as trees; the leaves of the tree are obtained by the initial rule of natural deduction derivations, which is called hypothesis rule and permits to prove any formula. In particular the single vertex proof-tree

$$A$$

Is a natural deduction derivation for any formula A. The meaning of such derivation is that we can prove A under the hypothesis that A holds.

Table 1.2 shows the natural deduction rules for the implicational fragment of intuitionistic logic i.e., the fragment of intuitionistic logic in which formulas are constructed only by the connective $\supset$.

$$\dfrac{\begin{array}{c}\vdots\\ A\end{array} \qquad \begin{array}{c}\vdots\\ A \supset B\end{array}}{B}\,[\supset E] \qquad\qquad \dfrac{\begin{array}{c}[A]_n \cdots [A]_n\\ \vdots\mathfrak{D}\\ B\end{array}}{A \supset B}\,[\supset I]_n$$

Table 1.2: natural deduction rules for the $\supset$-fragment of Intuitionistic Propositional Logic

The $\supset I$-rule discharge an arbitrary, finite number (possibly 0) of occurrences of the formula A that are active in the proof of B. These discharged hypothesis are marked by the index n in the proof of $A \supset B$. In the $\supset E$-rule (also called modus ponens) the major premises of the rule is the formula $A \supset B$. The active and discharged hypothesis of this derivation are exactly those of the sub-derivations of A and $A \supset B$. We will write $B_1, \dots B_n \vdash A$ if there is a proof of A with undischarged hypothesis $B_1, \dots B_n$. If all the hypothesis of a proof $\mathfrak{D}$ of A are discharged, we will say that $\mathfrak{D}$ is a *closed* proof of A.

Example 1.1. *We give some examples of natural deduction proofs:*

$$\dfrac{\dfrac{[A]_2}{A \supset A}\,[\supset I]_1}{A \supset (A \supset A)}\,[\supset I]_2 \qquad\qquad \dfrac{\dfrac{[A]_1}{A \supset A}\,[\supset I]_1}{A \supset (A \supset A)}\,[\supset I]_2$$

Remark that in the left-hand proof the first $\supset I$ discharge 0 occurrence of the formula A and the second $\supset I$ rule discharge the unique occurrence of the formula A. In the right-hand proof we face the opposite situation: the first $\supset I$ rule discharge the unique occurrence of the formula A and the second $\supset I$ rule discharge 0 occurrences of A.

$$\dfrac{\dfrac{\dfrac{[A]_1 \qquad [A \supset A]_2}{A}\,[\supset E]}{A \supset A}\,[\supset I]_1}{(A \supset A) \supset (A \supset A)}\,[\supset I]_2 \qquad \dfrac{\dfrac{\dfrac{\dfrac{[A]_1 \qquad [A \supset A]_2}{A}\,[\supset E] \qquad [A \supset A]_2}{A}\,[\supset E]}{A \supset A}\,[\supset I]_1}{(A \supset A) \supset (A \supset A)}\,[\supset I]_2$$

In the left-hand proof we discharge just one occurrence of $A \supset A$ while in the right-hand one we discharge two occurrence of the same formula.

We have introduced a natural deduction system for the implicational fragment of intuitionistic logic; there are two reasons for our choice:

1. in our PhD dissertation natural deduction will be used uniquely to represents proofs for proper fragments of the implicational fragment of proposition intuitionistic logic i.e., intuitionistic multiplicative linear logic and the Lambek Calculus;

2. We will relate natural deduction with the simply typed λ-calculus in which the only complex types are functional types.

1.8.1 Detours and normal proofs

Let $\mathfrak{D}$ be a natural deduction proof. An introduction rules for $\supset$ that introduces the major premise of an elimination rule $\supset E$ is called a *detour* or redex. A derivation with no detours is called normal. There is a reduction rule to eliminates detours in a derivation.

$$\dfrac{\dfrac{\begin{matrix}[A]_n \cdots [A]_n\\ \vdots\mathfrak{D}\\ B\end{matrix}}{A \supset B}[\supset I]_n \qquad \begin{matrix}\vdots\mathfrak{D}_1\\ A\end{matrix}}{B}[\supset E] \quad \rightsquigarrow \quad \begin{matrix}\vdots\mathfrak{D}_1 & \cdots & \vdots\mathfrak{D}_1\\ A & & A\\ & \vdots\mathfrak{D} & \\ & B & \end{matrix}$$

Natural deduction proofs enjoys the following important property (see [63] or [60] for a proof of the following proposition)

Proposition 1.1. *Natural deduction for the $\supset$-fragment of intuitionistic logic enjoys strong normalization: there is no infinite sequence of $\rightsquigarrow$ reduction.*

A reduction process is confluent whenever if $\mathfrak{D}$ reduces to $\mathfrak{D}_1$ and $\mathfrak{D}_2$ in some number of steps, then there is $\mathfrak{D}'$ such that both $\mathfrak{D}_1$ and $\mathfrak{D}_2$ reduces to $\mathfrak{D}'$ in some number of steps. If each time that $\mathfrak{D}$ reduces to $\mathfrak{D}_1$ and $\mathfrak{D}_2$ in *one* step of reduction, then one can find a $\mathfrak{D}'$ such that $\mathfrak{D}_1$ and $\mathfrak{D}_2$ reduces to $\mathfrak{D}'$ in some number of steps, then the reduction process is said to be locally confluent. If a reduction process is both strongly normalizing and locally confluent, then it is confluent. The normalization procedure of natural deduction for $\supset$-fragment is locally confluent (see again [63] or [60]). We can thus conclude that the following proposition holds.

Proposition 1.2. *The reduction process $\rightsquigarrow$ of natural deduction for the $\supset$-connective is locally confluent.*

Remark that by virtue of the two above propositions, every proof $\mathfrak{D}$ containing detours can be transformed to a unique proof $\mathfrak{D}'$ that contains no detour.

1.8.2 Intuitionistic multiplicative linear logic

Linear logic was introduced by Jean-Yves Girard in his seminal work [59]. Linear logic is a refinement of classical and intuitionistic logic. Instead of emphasizing truth, as in classical logic, or proof, as in intuitionistic logic, linear logic emphasizes the role of formulas as resources. The hypothesis of a proof are seen as resources that get consumed to get the conclusion of the proof. The literature on linear logic is massive, and it will be impossible to summarize it all. In our PhD dissertation we will use only the intuitionistic fragment of multiplicative linear logic in which formulas are generated, from a set of propositional variables, by the connective $\multimap$ (linear implication). To obtain a natural deduction system for intuitionistic multiplicative linear logic (IMLL for short) it is enough to restrict the rule $\supset I$ of table 1.2 so that every application of the rule discharge exactly one formula occurrence. The natural deduction system for IMLL is nothing but a subsystem of the natural deduction system presented above, thus proposition 1.1 and 1.2 holds for IMLL as well.

1.9 The Simply Typed Lambda Calculus

In this section we give a very succinct exposition of the simply typed λ-calculus. We will introduce only what is strictly needed for our successive exposition. For an in-depth introduction to the simply typed λ-calculus see [63, 71]. Types of the simply typed λ-calculus are defined from a set of primitive types $\mathcal{P}$ according to the following grammar:

$$\mathcal{T} := \mathcal{P} \mid \mathcal{T} \rightarrow \mathcal{T}$$

any type $T \in \mathcal{T}$ is given a countable set of variables of this type and constant of this type. We will use upper case letters from the roman alphabet $A, B, C, D \ldots$ etc. to denote types.

The set of typed terms, the set $FV(M)$ of free variables of a term M, and the notion of subterm of a term are defined as follows:

- variables: if x is a variable of type A written $x : A$ or x^A, then x is a term of type A. The set $FV(x)$ of free variables of x contains only x and x is the only subterm of itself;
- constants: if k is a constant of type A, written $k : A$ or k^A, then k is a term of type A and the set of free variables of k is empty. The only subterm of k is k itself;
- application: if M is a term of type $A \rightarrow B$, written $M : A \rightarrow B$ or $M^{A \rightarrow B}$, and N is a term of type A, written $N : A$ or N^A, then MN is a term of type B. The set $FV(MN)$

is equal to $FV(M) \cup FV(N)$. The subterms of MN are MN itself, the subterms of M and the subterms of N;

- abstraction: If M is a term of type B and x is a variable of type A then $\lambda x^A M$ is a term of type $A \to B$. The set $FV(\lambda x M)$ is equal to $FV(M) \backslash \{x\}$. The subterms of $\lambda x^A M$ are $\lambda x^A M$ itself and the subterms of M.

Variables and constants will be called atoms. An occurrence of a variable x, which is a subterm of M but does not belong to $FV(M)$, is said to be bound in M and it is associated to a unique lambda binder λx.

Let $M, N_1, \ldots N_n$ be terms and $x_1, \ldots x_n$ be distinct variables, and suppose that for all $i \in \{1, \ldots n\}$ N_i and x_i have the same type. The term $M[N_1/x_1, \ldots N_n/x_n]$ i.e., the result of the *capture avoiding* substitution of N_i to each free occurrence of x_i in M for $i \in \{1, \ldots n\}$, is defined by induction on M.

i $x_i[N_1/x_1, \ldots N_n/x_n] = N_i$;

ii $\alpha[N_1/x_1, \ldots N_n/x_n] = \alpha$ if α is an atom and $\alpha \neq x_i$ for all $i \in \{1, \ldots, n\}$;

iii $PQ[N_1/x_1, \ldots N_n/x_n] = P[N_1/x_1, \ldots N_n/x_n]Q[N_1/x_1, \ldots N_n/x_n]$;

iv $(\lambda z P)[N_1/x_1, \ldots N_n/x_n] = \lambda z P$ if for all $i \in \{1, \ldots, n\}$ $x_i \notin FV(P)$;

v $(\lambda z P)[N_1/x_1, \ldots N_n/x_n] = \lambda z(P[N_1/x_1, \ldots N_n/x_n])$ if for some $i \in \{1, \ldots, n\}$ $x_i \in FV(P)$ and z is not free in any of the N_i such that $x_i \in FV(P)$.

vi $(\lambda z P)[N_1/x_1, \ldots N_n/x_n] = \lambda w(P[w/z][N_1/x_1, \ldots N_n/x_n])$ if for some $i \in \{1, \ldots n\}$ $x_i \in FV(P)$ and z is free in some N_i such that $x_i \in FV(P)$. The variable w appear neither in P nor in any of the N_i;

vii $(\lambda x_i P)[N_1/x_1, \ldots N_n/x_n] = \lambda x_i(P[N_1/x_1, \ldots N_{i-1}/x_{i-1} N_{i+1}/x_{i+1}, \ldots N_n/x_n])$.

A redex is an application MN where M is an abstraction $\lambda x M'$. A term that is redex-free is said to be normal or in normal form. A term M is in *head normal form* iff

$$M = \lambda x_1 \cdots \lambda x_m \alpha N_1 \cdots N_n$$

with $m, n \geq 0$ and α is a (possibly bound) atom. Remark that every normal term is in head normal form but not every term in head normal form is normal e.g., $(k^A((\lambda z^B z)w^B)$ is a term in head normal form but it contains the redex $(\lambda z^B z)w^B$ A one-step beta reduction is defined by $(\lambda x M)N \overset{\beta}{\rightsquigarrow}_1 M[N/x]$ and it preserves typing. The transitive and reflexive closure of $\overset{\beta}{\rightsquigarrow}_1$ will be written $\overset{\beta}{\rightsquigarrow}$. The proof of the following proposition can be found in e.g., [63]

Proposition 1.3. *In the simply typed λ-calculus Beta reduction is strongly normalizing and confluent, that is*

- *strongly normalizing: there is no infinite path of beta reduction in the simply typed lambda calculus;*
- *confluent : if $M \overset{\beta}{\rightsquigarrow} N_1$ and $M \overset{\beta}{\rightsquigarrow} N_2$ then there is N such that $N_1 \overset{\beta}{\rightsquigarrow} N$ and $N_2 \overset{\beta}{\rightsquigarrow} N$.*

As a consequence of the previous proposition, we obtain that every term has a normal form and that this normal form is unique.

We shall consider also eta-expansion, which is defined as follows, where M has type $A \to B$ and $x^A \notin FV(M)$:

$$M \overset{\eta}{\rightsquigarrow} \lambda x^A M x$$

In the following, we will consider β-normal η-long forms that are defined using the auxiliary notion of atomic forms:

i each atom is an atomic form

ii MN is an atomic form if M is an atomic form and N is β-normal and η-long

iii each atom that has an atomic type is β-normal and η-long

iv $\lambda x M$ is β-normal and η-long whenever M is

Finally, we define two subclasses of simply typed λ-term that will be used in the following.

Definition 1.19. *A λ-term M is called a λ_I term [71, 9] iff for each subterm $\lambda x.N$ of M x occurs free in N at least once.*

A λ-term M is called linear *whenever each free variable occurs exactly once, and for each subterm $\lambda x.N$ of M, x has exactly one free occurrence in N*

Let use denote by Λ_I the class of λ_I-terms and by Λ_{lin} the class of linear lambda terms. We have that $\Lambda_{lin} \subset \Lambda_I$; moreover, it is easy to prove that if $M \in \Lambda_I$ (resp. $M \in \Lambda_{lin}$) and $M \overset{\beta}{\rightsquigarrow} N$ then $N \in \Lambda_I$ (resp. $N \in \Lambda_{lin}$).

1.10 The Curry-Howard Correspondence

$$\dfrac{\begin{array}{c}\vdots\\ N:A\end{array} \qquad \begin{array}{c}\vdots\\ M:A\supset B\end{array}}{MN:B}\,[\supset E] \qquad\qquad \dfrac{\begin{array}{c}[y_1:A]_n\cdots[y_n:A]_n\\ \vdots\\ M:B\end{array}}{\lambda x^A(M[x/y_1,\dots x/y_n]):A\supset B}\,[\supset I]_n$$

Table 1.3: natural deduction Rules with λ-terms decorations.

The Curry Howard Correspondence relates proofs in natural deduction and terms in the simply typed λ-calculus. A type is seen as a formula, with $A \supset B$ representing the type $A \to B$ of functions from A to B.

The rules of natural deduction are seen as term-construction rules for the simply typed λ-calculus. A natural deduction derivation $\mathfrak{D}$ of C is seen as a term M of type C in which:

- the free hypothesis of $\mathfrak{D}$ are decorated with the free variables of M;
- the discharged hypothesis of $\mathfrak{D}$ are the bound variables of M.

Natural deduction rules decorated with λ-terms are shown in table 1.3. We suppose that every formula that is a leaf of a derivation $\mathfrak{D}$ is decorated with a different variable.

Example 1.2. *The natural deduction proofs of example 1.1 corresponds to the following λ-terms:*

$$\dfrac{\dfrac{[x:A]_2}{\lambda y^A x:A\supset A}\,[\supset I]_1}{\lambda x^A\lambda y^A x:A\supset(A\supset A)}\,[\supset I]_2 \qquad\qquad \dfrac{\dfrac{[x:A]_1}{\lambda x^A x:A\supset A}\,[\supset I]_1}{\lambda y^A\lambda x^A x:A\supset(A\supset A)}\,[\supset I]_2$$

$$[\supset I]_2\,\dfrac{[\supset I]_1\,\dfrac{[\supset E]\,\dfrac{[x:A]_1 \qquad [f:A\supset A]_2}{fx:A}}{\lambda x^A fx:A\supset A}}{\lambda f^{A\supset A}\lambda x^A fx:(A\supset A)\supset(A\supset A)}$$

$$[\supset I]_2\,\dfrac{[\supset I]_1\,\dfrac{[\supset E]\,\dfrac{\dfrac{[x:A]_1 \qquad [f_1:A\supset A]_2}{f_1x:A}\,[\supset E] \qquad [f_2:A\supset A]_2}{f_2(f_1x):A}}{\lambda x^A f_2(f_1x):A\supset A}}{\lambda f^{A\supset A}\lambda x^A f(fx):(A\supset A)\supset(A\supset A)}$$

The Curry-Howard correspondence does not concern uniquely terms and derivations; also the computational process of the λ-calculus i.e., β-reduction, corresponds to detour-elimination in natural deduction.

We Resume the Curry-Howard Correspondence in the following table

λ-**calculus**	**Natural Deduction**
Type	Formula
Type $A \to B$	Formula $A \supset B$
Term of type A	Proof of A
β-reduction	Detour-elimination

Chapter 2

Philosophical Foundations

Abstract

In this chapter we discuss the philosophical relevance of the two proof system that we introduced in the previous chapter: natural deduction and the sequent calculus. We first detail the relationship between inferentialism and natural deduction. We then highlight some critical aspects of natural deduction as meaning conferring system for classical logic. We then sketch a dialogic interpretation of the sequent calculus that encompass such criticisms.

2.1 Meaning and natural deduction

According to many inferentialist philosophers such as Dummett and Prawitz [115, 116, 118, 41, 42] the meaning of the logical constants can be specified by means of natural deduction systems. Remember that for an inferentialist the meaning of a sentence is given by the way in which we can justify its *assertion*. Inferentialist such as Prawitz and Dummett specify that the meaning of sentences is given by a certain *canonical* way of justifying their assertion. In the case of the logical connectives, the meaning of a sentence A with main connective $\star$ is given by the set of natural deduction proofs of A that ends with a $\star$-introduction. The idea that introduction rules in a natural deduction System are *meaning constitutive* comes directly by a remark made by Gentzen

> The introductions represent, as it were, the "definition" of the symbol concerned, and the eliminations are no more, in the final analysis, then the consequence of these definitions. This fact may be expressed as follows: in elimi-

nating a symbol, we may use the formula with those terminal symbol we are dealing only in the sense afforded it by the introduction of that symbol [58]

In which sense is the meaning conferred by the elimination rules the consequences of the meaning that is given to the logical constants by the introduction rules? To answer this question, consider the natural deduction system we have introduced. Let $\mathfrak{D}$ be a closed proof of an implicational formula $B \supset C$. Suppose that $\mathfrak{D}$ ends in an elimination rule. Thus, $\mathfrak{D}$ has the following form

$$\dfrac{\dfrac{\vdots \mathfrak{D}_1}{A \supset (B \supset C)}\,\mathrm{R} \qquad \begin{matrix}\vdots \mathfrak{D}_2 \\ A\end{matrix}}{B \supset C}\,[\supset E]$$

We affirm that $\mathfrak{D}$ necessarily contains a redex. Suppose, to reach a contradiction, that $\mathfrak{D}$ is normal. Thus, the rule R can neither be a hypothesis rule —because $\mathfrak{D}$ is a closed proof— nor an introduction rule. It follows that R must be an elimination rule $\supset E$. By the fact that $\mathfrak{D}_1$ ends in an elimination rule, we conclude that $\mathfrak{D}_1$ and $\mathfrak{D}$ have the same form i.e., $\mathfrak{D}_1$ is of the form

$$\dfrac{\dfrac{\vdots \mathfrak{D}_{1.1}}{A_1 \supset (A \supset (B \supset C))}\,\mathrm{R} \qquad \begin{matrix}\vdots \mathfrak{D}_{1.2} \\ A_1\end{matrix}}{A \supset (B \supset C)}\,[\supset E]$$

By the same kind of reasoning as above, we again conclude that R must be an elimination rule. Thus, $\mathfrak{D}_{1.1}$ will have the same form as $\mathfrak{D}_1$ and $\mathfrak{D}$, and so on. This means, in particular, that the proof $\mathfrak{D}$ contains an infinite branch which is clearly absurd.

Elimination rules are thus consequences of the introduction rules in the following sense: if we take closed proofs ending with an introduction rules as meaning constitutive, closed proofs that ends in an elimination rule can always be transformed into the former type of proofs. A closed proof $\mathfrak{D}$ that ends in an elimination rule necessarily contain a detour. By applying (a finite number of times) the detour elimination procedure on $\mathfrak{D}$, we obtain a closed normal proof $\mathfrak{D}'$. This last proof necessarily ends in an introduction rule.

If this type of meaning explanation works very well with respect to intuitionistic logic, problems arise if we consider the natural deduction system for (the implicational fragment of) Classical propositional logic. This system is obtained by adding the following two deduction rules to the one shown in table 1.2

$$\dfrac{\begin{array}{c}\vdots\\ \bot\end{array}}{A}\,[\mathrm{Efq}] \qquad\qquad \dfrac{\begin{array}{c}[\neg A]_n \cdots [\neg A]_n\\ \vdots\\ \bot\end{array}}{A}\,[\mathrm{Raa}]_n$$

In this latter system there are closed normal proof of implicational formulas that do not end in an implication introduction. Consider the following closed normal proof of the Peirce law, which exemplifies the foregoing problem (we recall that $\neg A$ is a shortcut for $A \supset \bot$)

$$\dfrac{\dfrac{[(((A \supset B) \supset A) \supset A) \supset \bot]_3 \qquad \dfrac{\dfrac{[(A \supset B) \supset A]_2 \qquad \dfrac{\dfrac{[\supset E]\ \dfrac{\dfrac{[A]_1}{((A \supset B) \supset A) \supset A}\,[\supset I]_0 \qquad [(((A \supset B) \supset A) \supset A) \supset \bot]_3}{\bot}}{B}\,\mathrm{Efq}}{A \supset B}\,[\supset I]_1}{A}\,[\supset E]}{((A \supset B) \supset A) \supset A}\,[\supset I]_2}{\bot}\,[\supset E]}{((A \supset B) \supset A) \supset A}\,[\mathrm{Raa}]_3$$

A partisan of the theory of meaning proposed by Dummett and Prawitz will tend to dismiss this objection by saying that something is wrong with classical logic: namely since we cannot give an inferentialist explanation based on the notion of canonical proofs to the meaning of the classical implication, then the latter connector is simply devoid of intelligible meaning. Said in another way: the meaning of the classical logical constant is obscure since their meaning theory is not satisfactory. In our particular case, the meaning of the classical implication $\supset$ is not entirely determined by its introduction rules. To know the meaning of the logical constant $\supset$, we should also know the meaning of classical negation $\neg$. The reader may wonder 'what is wrong with the proposition that knowing the meaning of one logical constant depends upon knowing the meaning of another logical constant?'. Dummett's answer would be that this kind of explanation is not molecular: we cannot understand the meaning of a sentence solely on the basis of the meaning of its components. We have two answers to this kind of remark. First, it is not clear whether molecularity of meaning really fails for classical logic. As we will see in the next section, the molecularity problem exposed above seems to be tied to the particular format of natural deduction rules. Moreover, even were molecularism to fail, we don't see where the problem is. In our view, a satisfactory theory of meaning should simply be non-holistic: the meaning of a given expression in language should not depend on the meaning of every other expression in language. To exemplify this point, note the meaning of classical implication does not depend upon the meaning of classical disjunction. If we add to natural deduction the rule for disjunction, no new purely implicational formula became provable.

2.2 Meaning and the Sequent Calculus

Compared to the attention that philosophers have paid to natural deduction systems, the attention paid to the sequent calculus is very little. We will try to argue that the the sequent calculus is not philosophically insignificant.

In section 1.6 we have briefly underlined the mathematical importance of normal (or analytical) proofs; Analytical proofs, however, are important also for a philosophical reason that was already remarked by Gentzen

> Perhaps we may express the essential properties of such a normal proof by saying: it is not roundabout. No concepts enter into the proof other than those contained in its final result, and their use was therefore essential to the achievement of that result. [58]

Suppose, as inferentialist do, that the meaning of a formula F is given by the way in which we justify the assertion of F. The meaning of a formula is thus conferred by the inference rules. As we have already remarked, in the sequent calculus every logical symbol came with right and left introduction rules. Moreover, the rules for each logical constant is independent of the rules for another logical constant. A way to rephrase this observation is to say that, if we take the sequent calculus rules as *defining* the meaning of the logical connectives, then the meaning theory thus obtained is not holistic: the meaning of each logical connective is defined independently of the meaning of another logical connective. However, in the presence of the cut-rule can we really rule out the possibility that the meaning of a logical constant depends upon the meaning of another different logical constant? The answer is no: suppose that we restrict our self to the implicational fragment of LK, and we say that the meaning of the logical connective $\supset$ in classical logic is entirely defined by this fragment of LK. Now suppose that we add to this fragment the rules for the conjunction connective and the cut-rule. How can we be sure of the fact that no purely implication formula that was not provable in the implicational fragment of LK became provable in the augmented fragment? We simply cannot: because we might prove it by resorting in an essential way to the conjunction rules and then cutting away the formula introduced by the conjunction rules. However if we stipulate that the meaning conferring object are the normal proofs then our meaning theory for the logical constant is indeed non-holistic. In the previous section we have underlined that, according to inferentialists that take introduction rules in natural deduction as meaning conferring rules, the meaning of the classical logical constant is ill-defined. There are closed proofs of pure implications that do not end in implication introduction. Thanks to the sub-formula property this is not the case in the sequent calculus and it thus enjoys the advantage over natural deduction with respect to being meaning-conferring. There are however two philosophical difficulties in considering the sequent calculus in terms of a meaning conferring proof system.

2.2.1 The role of the structural rules

The first difficulty is given by the fact, remarked on in the foregoing, that there are two kind of rules in the sequent calculus. The logical rules and the structural rules. If we say that the meaning of a logical constant is given by its left and right introduction rules we are not saying enough; in this case the meaning of classical, intuitionistic, relevant and linear implication would be the same since these four connectives have the same left and right introduction rules in the sequent calculus. Intuitionistic logic is obtained by dropping the right structural rules of LK, relevant logic by dropping the right and left weakening rules of LK and linear implication by dropping all the structural rules but the exchange rules. To obtain a proper characterization of the meaning of the logical constants, one has to take into account the role of the structural rules.

2.2.2 Formulas vs sequents

The second difficulty is given by the fact that natural deduction rules concerns formulas while sequent calculus rules concerns sequent. It is quite natural to see a natural deduction proof as a tree of arguments given in support of the formulas that label its vertex. A rule

$$\frac{\begin{matrix}\vdots & & \vdots \\ H_1 & \cdots & H_n\end{matrix}}{C}$$

can be naturally undering in the following terms: ‘to produce an argument that supports the assertion of C you should produce arguments that supports the assertions of all the H_i’. Since the inferentialist believes that the meaning of a proposition is given by the argument that supports the assertion of the proposition, natural deduction proofs have an intuitive appeal: each step of a natural deduction proof can be seen as the justification of a certain assertion (under certain hypotheses). The situation is more complicated in the sequent calculus. A rule is in the following form

$$\frac{\Gamma_1 \vdash \Delta_1 \qquad \cdots \qquad \Gamma_n \vdash \Delta_n}{\Sigma \vdash \Pi}\ \mathrm{r}$$

one immediate reflex would be to interpret such a rule by interpreting sequents as their corresponding formulas. The reading of the above rule would be ‘to justify the assertion of the disjunction of the formula in Π under the hypothesis that the assertion of the conjunction of the formulas in Σ can be justified, justify the assertion of the disjunction of the formulas in Δ_i under the hypothesis that the conjunction of the formulas in Γ_i can be justified for $i \in \{1, \ldots n\}$’. This kind of reading of the rule is indeed natural and has been advocated by some philosophers such as Restall [124]. This notwithstanding, the legitimacy of such a reading has been convincingly questioned by Dummett [42] (p. 87) and Steinberg who writes:

> Moreover, by our inferentialist hypothesis, such a characterization [of the meaning of the logical operators] is to be given within the confines of an interpreted proof system that codifies all meaning-theoretically relevant inferential relations. However, if the only possible (informal) interpretation of our proof-theoretic framework necessitates a prior understanding of certain logical operators, it will not be a suitable medium within which to settle questions of legitimacy of any of the principles containing the logical constants in question [130].

said in a simpler way: to understand the inference rule of the classical sequent calculus we need to grasp the meaning of the logical operator $\vee$. This implies a sort of vicious circle.

2.3 Meaning and argumentation

The idea that the meaning of the logical constants should be specified in terms of argumentation rules in particular types of two-player argumentation games is due to Paul Lorenzen [96] and has been philosophically and mathematically developed by Kuno Lorenz in his PhD dissertation titled ‘Arithmetik und Logik als Spiele’ (Lorenz’s PhD dissertation is contained in [97]). The two authors introduced Dialogical Logic, which will be focus of two chapters of our PhD dissertation. Games in dialogical logic are (schematic) argumentative dialogues between two players: one player — the Proponent — tries to construct a justification for the assertion of a certain formula, while the other player — the Opponent— doubts that such justification actually exists. Each game is a series of attacks (questions directed toward a certain assertion) and defenses (answers to question that have been asked in the course of the dialogue). What count as a question about a certain formula depends upon the logical form of the formula asserted in the game. What counts as an answer for a question depends upon the question e.g., if the formula F is $A \wedge B$ a question on F is either ‘could you assert A?’ or ‘could you assert B?’. An answer to the former question is an A assertion, while an answer to the latter question is a B assertion. In his book ’Making it explicit’ [16] the philosopher Robert Brandom argues that linguistic meaning is determined by inferential practices rather than by truth value. Inferential practices are made explicit i.e., intersubjectively observable and acquirable, because language user are engaged in a perpetual game of ‘giving and asking for reasons’. Brandom’s approach is grounded in the theory of assertions: according to Brandom the speech act of asserting includes the willingness to play this game of ’giving and asking for reasons’. As has already been noted by Marion, Dialogic logic is particularly well suited to formally capture the philosophical remarks about meaning theorized by Brandom

> My suggestion is simply that dialogical logic is perfectly suited for a precisification of these ‘assertion games’. This opens the way to a ‘game-semantical’ treatment of the ‘game of giving and asking for reasons’: ‘asking for reasons’ corresponds to ‘attacks’ in dialogical logic, while ‘giving reasons’ corresponds to ‘defences’ [98].

To resume: an argument in favor of a statement is often developed when a critical audience, real or imaginary, doubts the truth, or the plausibility of the proposition. In this case, in order to successfully assert the statement, a speaker or proponent of it must be capable of providing all the justifications that the audience is entitled to demand. Taking this idea seriously, an approximation of the meaning of a sentence in a given situation can be obtained by studying the *argumentative dialogues* that arise once the sentence is asserted in front of such a critical audience. This type of situation is captured — with a reasonable degree of approximation — by dialogical logic. In the dialogical logic framework, knowing the meaning of a sentence means being able to provide a justification of the sentence to a critical audience. Note that with this type of methodology the requirement of manifestability required to attribute knowledge of the meaning of a sentence to a locutor is automatically met. The locutor who asserts a certain formula is obliged to make his knowledge of the meaning manifest so that he can answer the questions and objections of his interlocutor. In addition, any concessions made by his interlocutor during the argumentative dialogue will form the linguistic context in which to evaluate the initial assertion.

2.4 A DiaLogical interpretation of the sequent calculus

In this section we try to show how sequent calculus proofs can be seen in terms of strategies for a particular type of two player games. The game's participant are the Proponent (**P**) and the Opponent (**O**). The two participants alternate in the game and the proponent starts by asserting a certain formula. Each move of the game is either a question directed toward a formula that has been asserted by the other player or an answer to a question that has been posed by the other player. What answers and question consist of depends upon the form of the formulas that we are considering. If F is of the form $A \supset B$ a question about F would be "could you assert B if I concede that A holds?". An answer to this question is simply an assertion of B. If F is of the form $A \vee B$ a question about F is "could you assert either A or B?". An answer to this question is either an assertion of A or an assertion of B; there is no question directed toward atomic formulas and the Proponent cannot assert the formula that represent absurdity ($\perp$). The behavior of the two players is restricted by some simple rules:

- the Proponent cannot assert an atomic formula unless the Opponent has already asserted it;
- the Opponent must always react to the immediately preceding Proponent Intervention.

The Proponent wins a Game whenever she asserts an atomic formula as an answer to a question asked by the Opponent, or whenever she forces the Opponent to assert the absurd $\bot$. This type of game is biased toward the Proponent. Note that there is nothing to prevent her from answering more than once, and in different ways, the same question of the Opponent.

Now consider the following sequent calculus proof[1] $\mathfrak{D}$ of $\vdash a \lor \neg a$ where a is a propositional variable, and in which we underline the active formula of each rule application.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{a \vdash \underline{a}, a \lor \neg a, \bot}\ Id}{a \vdash \bot, \underline{a \lor \neg a}}\ \lor^{\mathsf{R}}_1}{\vdash \underline{\neg a}, a \lor \neg a}\ \supset^{\mathsf{R}}}{\vdash \underline{a \lor \neg a}}\ \lor^{\mathsf{R}}_2$$

The proof, if read bottom-up, can be interpreted in terms of a game won by the Proponent: the root formula $a \lor \neg a$ correspond to the first assertion of the game. The opponent asks the proponent to justify this latter assertion by requesting the proponent to assert either a or $\neg a$. The Proponent asserts $\neg a \equiv a \supset \bot$. The Opponent asks the proponent to justify this latter formula by asserting a and querying the proponent for an assertion of $\bot$. The proponent cannot assert $\bot$, thus she answers in a different way to the question concerning $a \lor \neg a$. She thus asserts a and wins the game. We can decorate the above proof by signing formulas that are asserted by **P** and **O**. Side formulas that are asserted by **P** (somewhere in the proof) can be re-defended by **P**

$$\dfrac{\dfrac{\dfrac{\dfrac{}{a^{\mathbf{O}} \vdash \underline{a^{\mathbf{P}}}, a \lor \neg a, \bot}\ Id}{a^{\mathbf{O}} \vdash \bot, \underline{(a \lor \neg a)^{\mathbf{P}}}}\ \lor^{\mathsf{R}}_1}{\vdash \underline{\neg a^{\mathbf{P}}}, a \lor \neg a}\ \supset^{\mathsf{R}}}{\vdash \underline{(a \lor \neg a)^{\mathbf{P}}}}\ \lor^{\mathsf{R}}_2$$

By considering a proof in the sequent calculus in terms of a two player game we can give a simple explanation of the sequent calculus rules that does not require a previous understanding of the meaning of the logical constants themselves; in particular side formulas on the right of the $\vdash$ symbol are there to signal that the Proponent has asserted them at some point of the game, that they have being questioned by the Opponent and that can be re-defended by the Proponent.

[1]we use the variant of the sequent calculus in which contraction rules are absorbed in introduction rules. This variant has been briefly described in subsection 1.7.7 of the previous chapter

2.5 Proof Semantics vs Semantics of Proofs

A proof (theoretic) semantics is an explanation of the meaning of formulas in terms of the arguments, or proofs, that could establish the assertability of the formulas.

$$[\![F]\!] = \{\mathfrak{D} \mid \mathfrak{D} \text{ is a proof of } F\}$$

A semantics of proof is an explanation of the meaning of (formal) proofs. The two types of semantics are distinct but enjoy, at least on a philosophical level, a close kinship. As we have already remarked, formal proofs in natural deduction can be easily seen as chaining of arguments. Natural deduction is indeed 'natural' in this sense: formal objects of natural deduction are a good approximation of 'real' mathematical proofs. Let us quote Steinberg once more:

> Only those deductive systems that answer to the use we put our logical vocabulary to fit the bill. After all, it is the practice represented, not the formalism as such, that confers meanings. Therefore, the formalism is of meaning-theoretic significance and hence of interest to the inferentialist only if it succeeds in capturing (in a perhaps idealised form) the relevant meaning-constituting features of our practice [...] It has become customary in the inferentialist tradition to regard Gentzen– Prawitz natural deduction systems as the privileged proof-theoretic framework within which to carry out the inferentialist program. Its alleged "close affinity to actual reasoning" is thought to make natural deduction deserving of the honorific title 'natural'. [130]

The situation is more complicated in the sequent calculus: in order to gain an intuitive understanding of the proofs from the sequent calculus, we interpreted these as winning strategies for a two-player game, i.e., we sketched an informal semantics of proofs for the sequent calculus in order to see this proof system in terms of a proof theoretic semantics for classical logic. Indeed, one of the main contribution of our PhD thesis will be to give a game semantic interpretation of sequent calculus proofs. In light of all the above discussion, our contribution can be considered both a logical and philosophical contribution.

In particular, we will give a game semantic interpretation of proofs in the classical and intuitionistic sequent calculus in terms of Dialogical Games, and we will give an interpretation of *constructive* modal proofs in terms of Game-semantics.

2.5.1 Sequent calculus proofs as dialogic games: other approaches.

Our interpretation of the sequent calculus in terms of dialogic games is hardly the first. According to Girard [62] “It is in Gentzen’s first consistency proof that one can find the first interpretation of a formula of logic – or rather arithmetic – by a game between a player Me and an opponent You.” The games are played on formulas of the form $\forall x_1 \exists y_1, \forall x_n \ldots \exists y_n F$ where F is quantifier free. You (the Opponent) start the game by giving a value n to x_1 in. In this way he challenges Me (the Proponet) to give a proof of $\exists y_1, \ldots \forall x_n \exists y_n F[n/x_1]$. The proponent then propose a value m for y_1 and so on. As in dialogical logic, the proponent has a crucial advantage: at any point of the game she can change her mind and choose a new value for an existential quantifier. If the proponent find, after finitely many attempts, a value n such that a formula $A[n]$ holds, she wins the game. An explicit interpretation in terms of Games and Strategies of Gentzen’s proof is studied by Coquand [29] and by Herbelin in his PhD thesis [68].

Ludics [61] is another main contribution to this kind of reading of sequent calculus proofs as strategies for two player games. In Ludics the main object, design, are a ’syntax free’ counterpart of sequent calculus proofs. In a design, all the information’s about formulas is discarded but the ‘location’ that formulas may occupy during proof search. Design are seen as fallible strategies in a two player game. Moreover, there are numerous scientific works that connect the form of inferentialism theorized by Brandom to Ludics [95, 52, 94]. All of these approaches are interesting and would merit thorough consideration. However, we have chosen to focus on dialogic logic for one reason : its simplicity. Dialogic logic systems have an intuitive appeal, and the treatment of classical logic within them is very simple.

Part II

Logic and DiaLogical Games

Chapter 3

DiaLogical Games

Abstract

In this chapter, we give a proof of the correspondence between the existence of a winning strategy for E-games and classical validity for first order logic. The proof is obtained by a direct mapping between formal E-strategies and derivations in a cut-free complete sequent calculus for first order intuitionistic logic. Our approach builds on the one developed by Herbelin in his PhD dissertation [68]. We detail also a mapping between winning strategies for Intuitionistic E-games and derivations in a cut-free complete sequent calculus for first order intuitionistic logic. Our proof greatly simplifies the proof of correspondence given by Felscher in his classic paper [46]. The result of this chapter already appears in [22] for classical logic and in [21] for intuitionistic logic.

3.1 Introduction

The art of persuasive debate, dialectics, and the science of valid inference, logic, have been intrinsically linked since their beginnings [99, 19, 20, 44]. At the dawn of the modern age the connection between the two disciplines seemed so clear that one of the first sentences pronounced by Doctor Faustus in Marlowe's work goes as follows

> Is, to dispute well, logic's chiefest end? Affords this art no greater miracle?

Despite this ancient connection between the two disciplines, mathematical logic had to wait until the 50s of the last century to determine that the logical concept of validity could be expressed through the use of dialogical concepts and techniques. The German mathematician and philosopher Lorenzen [96] proposed to analyze the concept of validity of a formula A through the concept of winning strategy in a particular type of two-player game. This type of game is nothing more than an argumentative dialogue between a player, called Proponent, who affirms the validity of a certain formula A and another player, called Opponent, who contests its validity. The argumentative dialogue starts by the Proponent affirming a certain formula. The Opponent takes his turn and attacks the claim made by the Proponent according to its logical form. The Proponent can, depending on the form of the attack made by the Opponent, either defend her previous claim or counter-attack. The debate evolves following this pattern. The Proponent wins the debate if she has the last word, i.e., the defense against one of the attacks made by the Opponent is a proposition that the Opponent cannot attack without violating the debate rules.

Dialogical logic was initially conceived by Lorenzen as a foundation for intuitionistic logic (IL). Subsequently, various dialogical logic systems were developed for the most diverse logics e.g., modal logics [50], paraconsistent logics [122] free logics [120] etc. Lorenzen's original idea was the following: it is possible to define a natural class of dialogue games in which given a formula A, the Proponent can always win a game on A, *no matter how the opponent chooses to act in the debate*, if and only if A is IL-valid. This intuition was formalized by saying that, given a certain class of dialogue games, and a formula A

> A is IL valid if, and only if, there is a winning strategy for the proponent for the formula A in the class of games under consideration.

Unfortunately almost 40 years of work were needed to get a first correct proof of the completeness theorem [46]. In this chapter we will focus on classical dialogical logic and intuitionistic dialogical logic. We will see in the next section that, despite the fact that dialogues for classical logic have been known since Lorenzen's very first paper on dialogical logic [96], a convincing proof that, given a first-order formula F, the existence of a winning strategy for F implies that F is classically valid and vice versa is nowhere to be found in the literature. To remedy this shortcoming, we give a formulation of dialogical games that we believe is clearer and more precise. Using this formulation, we show that it is possible to transform a winning strategy for a formula into a proof in a cut free complete sequent calculus of the formula and that, conversely, a derivation of the formula can be transformed into a winning strategy for the formula. We thus obtain a constructive proof of the equivalence between dialogical validity and classical validity. We then show that the exact same method applies to intuitionitistic dialogical logic thus greatly simplifying the notoriously difficult proof of Felscher.

3.1.1 Previous works

We briefly review the previous works on the equivalence between classical validity and dialogical validity. If one is interested in finding a proof of the equivalence between classical validity and dialogical validity for classical *propositional* logic one can consult [48] in which a mapping between winning strategies and derivations in a hyper-sequent calculus is presented or [6] in which some *small* errors of the previous paper are corrected and a mapping of winning strategies to derivations in a Kleene-style sequent calculus is presented. Another similar result is presented in Herbelin's PhD dissertation [68]. Herbelin presents a variant of the LK sequent calculus, called LKQ, and proves that, given a formula A, derivations for A in the sequent calculus LKQ correspond to winning E-strategies for A and vice versa. Herbelin's work only capture the fragment of classical propositional logic without negation but, despite this fact, offers in our opinion the clearest proof of correspondence between dialogical validity and (restricted) classical validity.

On the other hand numerous classes of dialogue games for classical first-order logic have been defined in the literature, but the correspondence between the class of games and classical validity is merely asserted, and not proved. Such previous work asserting a correspondence between first-order classical logic and a particular class of Lorenzen dialogue games without proof includes [26, 55, 86, 123, 129]. We count just one exception: Clerbout [25] proves the correspondence between classical validity and a particular class of games in which the players must declare, at the beginning of each game, the maximum number of attacks they can make on the same formula. Clerbout shows the completeness of his dialogical logic system by transforming winning strategies into proofs of a semantic tableaux system. Clerbout's proof has, in our opinion, two major flaws. The first defect, which is mostly aesthetic, is simply that the transformation of winning strategies into tableaux is no less complicated than the one which can be consulted in Felscher's classic paper for intuitionistic first-order logic [46]; In addition, the length of the proof and the absence of intermediate lemmas make it particularly difficult to understand and assess.

The second defect is more serious: Clerbout's proof is limited to a language in which no functional symbol appears i.e., the proof of completeness concerns a particular and restricted first-order language. Clerbout is aware of this limitation; at the end of his paper he writes

> We have considered the particular case of a first-order language without equality and without complex terms i.e., without function symbols. Furthermore, we have dealt with dialogical games and tableaux for sentences. Hence, future work shall consider a generalization of the result for arbitrary first-order languages and dialogical games and tableaux with free variables. This would be a chance to study the dialogical manifestation in strategical terms of the mechanism of unification in tableaux.

However, Clerbout did not publish further results on this topic.

Concerning intuitionistic logic the classic reference in which a proof of correspondence between the existence of a winning intuitionistic strategy and intuitionistic validity is [46]. Felscher's proof makes use of various intermediate notions that allow a winning strategy to be transformed into a proof in the sequent calculus LJ and vice versa. First, Felscher defines two types of dialogue games, called D-dialogues and E-dialogues. Second, he gives an algorithm that converts D-strategies into formal E-strategies (strategies that respect the eigenvariable condition). Third, algorithms are given which transform derivations of the sequent calculus LJ into what Felscher calls IC-protableaux. Felscher concludes his proof by providing an algorithm to transform an IC-protableaux into an E-strategy. As one can see merely from this description, Felscher's proof is a big nut to crack.

The objective of this chapter is simple: we give a proof of correspondence between the existence of a winning classical strategy and classical validity for a formula A. As in Herbelin's PhD dissertation, we present a complete cut-free sequent calculus system, that we call SLK (strategic LK), and a mapping between winning strategies for A and derivations of A in SLK. We then show how, with slight modifications, the same result can be obtained for intuitionistic winning strategies and intuitionistic validity using the intuitionistic variant SLJ of our SLK sequent calculus. We thus offer a considerable simplification of Felscher's proof.

Organisation Of the Chapter

The rest of the chapter is structured as follows: Sect. 3.2 introduces dialogical logic for classical logic: we define games and strategies and prove some preliminary results about them. Sect. 3.3 introduces the sequent calculus SLK (Strategic LK): we prove some results about SLK, in particular, that SLK is sound and complete for classical first-order logic. In Sect. 3.5 we show how to transform a winning strategy for a formula A into a derivation of A

in the calculus SLK; In Sect. 3.5 we show how to transform a derivation of A in the calculus SLK into a winning strategy for A. In Sect. 3.6 we consider the intuitionistic variant of our games, and we show how a natural correspondence between strategies for intuitionistic games and derivations in the intuitionistic variant SLJ of SLK can be obtained by slightly modifying the one for classical dialogical games. We conclude with some philosophical remarks in Sect. 3.7.

3.2 Dialogical Logic

3.2.1 Argumentative dialogues: informal overview

Before entering into the formal matter of dialogical logic, let us give an informal example of an argumentative dialogue about the validity of a formula. Let A and B stand for two arbitrary atomic formulas.

0. **P**: I affirm that $A \wedge B \supset B$
1. **O**: Let me assume, for the sake of the proof, that $A \wedge B$ holds, can you show that B holds?
2. **P**: You admitted that $A \wedge B$ holds, can you admit that B holds?
3. **O**: Indeed, I must admit that B holds.
4. **P**: Then I have nothing more to prove, you have admitted that B holds, if $A \wedge B$ holds.

We can see that the Proponent and the Opponent alternates in the dialogue. The dialogue is a sequence of interventions. Each intervention but the first is either an attack against a preceding intervention of the other player or a defense against an attack of the other player. For example **O** in intervention 1 attacks intervention 0 by asking **P** to show that B holds provided that $A \wedge B$ holds. **P**'s defense against 1 is the intervention 4. What counts as a question against an asserted formula A, and what counts as an answer to such a question, depends upon the logical form of A. For example in 2, **P** attacks the formula asserted in 1 by asking **O** to assert B. This is because if one presumes that a conjunction holds, one must be ready to concede that both members of the conjunction hold.

Given the foregoing discussion, an argumentative dialogue will be defined in terms of sequences of alternating interventions made by the Proponent and the Opponent. Each intervention in the dialogue is an attack or a defense against a preceding intervention made by the other player. The dialogue ends whenever the Opponent cannot produce a new intervention without contradicting what he already conceded.

The next subsections will be devoted to introducing formal content corresponding to this intuitive discussion. In subsection 3.2.2 we define what a question on a formula is and what counts as an answer to such a question. In subsection 3.2.3 we formally define what it means for an intervention in a dialogue to refer to another preceding intervention in the same dialogue (definitions 3.1 and 3.2). Finally, in subsection 3.2.4, we define (definition 3.3) the class of argumentative dialogues we are interested in (which we call games) and the conditions under which **P** wins in an argumentative dialogue.

3.2.2 Argumentation forms

Let $\mathcal{L}$ be a standard first-order language. We denote by $\mathcal{T}$ the set of terms of $\mathcal{L}$ and by $\mathcal{F}$ the set of formulas of $\mathcal{L}$.

The set of auxiliary symbols *Aux* is the smallest set containing the symbols $\wedge_1, \wedge_2, \vee, \exists$ and the expressions $\forall[t/x]$ for all terms in $\mathcal{T}$ and variables x in $\mathcal{L}$.

Following the terminology of Felscher [47], an argumentation form *Arg* is a function assigning to each non-atomic formula A a set of pairs consisting of one *question* and one *answer* with questions being either formulas or symbols in *Aux* and answers being formulas[1]

$$\begin{aligned}
Arg(A \supset B) &= \{(A, B)\} \\
Arg(A \wedge B) &= \{(\wedge_1, A), (\wedge_2, B)\} \\
Arg(A \vee B) &= \{(\vee, A), (\vee, B)\} \\
Arg(\forall x A) &= \{(\forall[t/x], A[t/x]) \mid t \in \mathcal{T}\} \\
Arg(\exists x A) &= \{(\exists, A[t/x]) \mid t \in \mathcal{T}\}
\end{aligned}$$

[1] The words “question” and “answer” are called “attack” and “defense” by Felscher in [47]; we deviate from this terminology because we will use the terms “attack” and “defense” exclusively for the moves in a game, avoiding possible confusion.

Given a couple $(q, a) \in Arg(A)$, q is called a *question on* A. Given $(q, B) \in Arg(A)$, the formula B is called an *answer to the question* q on A. So, for example, if A is $B \wedge C$, both $\wedge_1$ and $\wedge_2$ are question on A but only B is an answer to $\wedge_1$ and only C is an answer to $\wedge_2$. If $A = B \vee C$, the symbol $\vee$ is a question on A, and both B, C are answers to $\vee$. Consider the case where A is $B \supset C$. In this case B is a question on A and C is an answer to B.

3.2.3 Augmented sequence

A *defense move* is a couple $(!, A)$ where A is a formula. An *attack move* is a couple $(?, s)$ where s is either a formula or an auxiliary symbol. A *move* is either an attack move or a defense move. A move $(\star, A)$, where A is a formula and $\star \in \{?, !\}$, is called assertion move. We will also say that the move asserts the formula A, or that A is the asserted formula of the move. Attack moves of the form $(?, \exists)$ are called *existential attacks*. Attack moves of the form $(?, \vee)$ are called *disjunctive attacks*. Let $\rho = m_0, m_1, \ldots m_n \ldots$ be a sequence of moves. We denote by ρ_i the ith move of the sequence. The parity of ρ_i is the parity of i. An assertion move $\rho_j = (\star, A)$ is called a *reprise* if and only if there is move $\rho_k \in \rho$ with $k < j$ such that $\rho_k = (\star', A)$ and ρ_j, ρ_k have different parities

Definition 3.1. *An augmented sequence is a non-empty sequence of moves ρ together with a function ϕ that is defined on each ρ_i with $i \geq 1$ and such that, for all $i \geq 1$, $\phi(\rho_i) = \rho_j$ for some $j < i$. The move $\phi(\rho_i)$ is called the enabler of ρ_i.*

Definition 3.2. *Let (ρ, ϕ) be an augmented sequence:*

- *an attack move $\rho_i = (?, s)$ is justified whenever $\phi(\rho_i)$ is of the form $(\star, A)$ and s is a question on A;*
- *a defense move $\rho_i = (!, B)$ is justified whenever $\phi(\rho_i)$ is of the form $(?, s)$, $\phi(\rho_i)$ is justified, $\phi(\phi(\rho_i)) = (\star, A)$ and B is an answer to the question s on A.*

We give an example of an augmented sequence (ρ, ϕ); we represent the augmented sequence by a table with two columns and as many rows as there are moves in the sequence of moves. In the first column, we write down the moves of the sequence. In the second column, the value of the function ϕ for the corresponding entry in the first column:

σ	value of ϕ
$\mathfrak{m}_0 = (?, P \wedge Q)$	
$\mathfrak{m}_1 = (!, P)$	$\mathfrak{m}_0$
$\mathfrak{m}_2 = (?, \wedge_1)$	$\mathfrak{m}_1$
$\mathfrak{m}_3 = (?, \wedge_1)$	$\mathfrak{m}_0$
$\mathfrak{m}_4 = (!, P)$	$\mathfrak{m}_3$
$\mathfrak{m}_5 = (!, R \vee Q)$	$\mathfrak{m}_2$
$\mathfrak{m}_6 = (?, \vee)$	$\mathfrak{m}_5$

the first three moves, as well as the move $\mathfrak{m}_5$, are not justified. The moves $\mathfrak{m}_3$ and $\mathfrak{m}_6$ (colored in blue) are both justified attack-moves. The move $\mathfrak{m}_3$ is a justified attack move because $\wedge_1$ is a question on the formulas $P \wedge Q$ asserted by the move $\mathfrak{m}_0$ and $\mathfrak{m}_0$ is the enabler of $\mathfrak{m}_3$. The move $\mathfrak{m}_4$ (colored in red) is both the unique reprise of the augmented sequence, and the unique justified defense move: it is a reprise because it is an assertion move and there is a move with a smaller index of opposite parity i.e., $\mathfrak{m}_1$ that asserts the same formula. It is a justified defense because its enabler (the move $\mathfrak{m}_3$) is a justified attack move and the asserted formula P of $\mathfrak{m}_4$ is an answer to the question $\wedge_1$ on the formula that is asserted by the move $\mathfrak{m}_0$ i.e., $P \wedge Q$.

3.2.4 Games

Let (ρ, ϕ) be an augmented sequence, we say that a formula A appears in the augmented sequence if and only if there is a move $\mathfrak{m} \in \rho$ that asserts A. We say that a variable v appears in ρ whenever v occurs free in some asserted formula or there is a move $\mathfrak{m} = (?, \forall[v/x])$ in ρ. Fix an enumeration $(v_i)_{i \in I}$ of the variables of $\mathcal{L}$.

Definition 3.3 (Game)**.** *A game $\mathcal{G}$ for a formula A is an augmented sequence (ρ, ϕ) such that:*

1. $\rho_0 = (!, A)$ *and for all $i > 0$ the move ρ_i is justified;*
2. $\phi(\rho_i) = \rho_{i-1}$ *if i is odd,* $\phi(\rho_i) = \rho_j$ *with j odd if i is even;*
3. *if $\rho_i = (\star, B)$ with B atomic formula and i even then ρ_i is a reprise and $B \neq \bot$;*

4. *if ρ_i is an attack move of the form $(?, \forall[t/x])$ and i is odd then $t = v_k$; v_k is the first variable in the enumeration $(v_i)_{i\in I}$ that does not appear in the prefix of ρ ending with ρ_{i-1};*

5. *if $\rho_i = (!, B[t/x])$ is a defense move, i is odd and ρ_{i-1} is of the form $(?, \exists)$ then $t = v_k$; v_k is the first variable in the enumeration $(v_i)_{i\in I}$ that does not appear in the prefix of ρ ending with ρ_{i-1};*

6. *if ρ_i and ρ_j are defense moves, i and j are even and $\phi(\rho_j) = \phi(\rho_i) = \rho_k$, then ρ_k is either an existential attack or a disjunctive attack.*

In a game $\mathcal{G}$, moves ρ_i with i even are called **P**-moves. They are called **O**-moves otherwise. If $\mathcal{G}m$ is a game and m is **P**-move, we will write $\mathcal{G}\mathfrak{m}^{\mathbf{P}}$. We will write $\mathcal{G}\mathfrak{m}^{\mathbf{O}}$ otherwise. Let us make some comments about the definition of game. Conditions 1 and 2 assures us that each, but the first, move in a game is justified by a previous move, that **P**-moves are justified by **O**-moves and that **O**-moves are justified by the immediately preceding **P**-move. Condition 3 is usually called *formal* condition. The condition says that **P** can assert an atomic formula only if the formula has already been asserted by **O** during the game. The intuition behind condition 3 is that **P**'s method of arguing in the game is only determined by the meaning of the logical constants and quantifiers. The value of atomic formulas is only known to **O**, and only he can concede that it holds in the given dialogical context. Condition 4 says that **O** requires **P** to develop an argument in favor of a universally quantified formula using an arbitrary element of the discourse domain: the required term is a variable which does not appear free in any formula asserted in the game. Condition 5 is the dual of condition 4. While **O** defends an existentially-quantified formula, he must choose an arbitrary element of the discourse domain to do so. Finally, condition 6 asserts that the **P** can only re-defend existential formulas or disjunctions.

Let $\mathcal{G} = (\rho, \phi)$ be a finite game and $\mathfrak{m}$ be a move. The move $\mathfrak{m}$ is legal for $\mathcal{G}$ if and only if the augmented sequence $(\rho\mathfrak{m}, \psi)$ is a game, $\psi|_\rho = \phi$ and $\psi(\mathfrak{m}) \in \rho$ where $\psi|_\rho$ is the restriction of the function ψ to the sequence ρ.

Definition 3.4. *A game $\mathcal{G}$ is won by* **P** *if and only if it is finite and either*

- *the game is of the form $\mathcal{G}'m^{\mathbf{P}}$ and there is no move $\mathfrak{m}'$ legal for $\mathcal{G}$;*
- *the game is of the form $\mathcal{G}'\mathfrak{m}^{\mathbf{O}}$ and $\mathfrak{m}$ asserts $\bot$.*

In what follows, we will often identify a game with the sequence of its moves by an abuse of notation.

3.2.5 Some examples

We give some examples of games. A game will be represented as a table with two columns and as many rows as there are moves in the game. In the first column of the table, we will write down the moves of the game. In the second column, we will write the value of the function ϕ for the move in the first column. Let a, b, c be propositional variables, P a unary predicate variable and R a binary predicate variable. We recall that we write $\neg A$ as a shortcut for $A \supset \bot$.

$\mathfrak{m}_0 = (!, a \vee \neg a)$	
$\mathfrak{m}_1 = (?, \vee)$	$\mathfrak{m}_0$
$\mathfrak{m}_2 = (!, \neg a)$	$\mathfrak{m}_1$
$\mathfrak{m}_3 = (?, a)$	$\mathfrak{m}_2$
$\mathfrak{m}_4 = (!, a)$	$\mathfrak{m}_1$

$\mathfrak{m}_0 = (!, a \supset b \supset ((b \supset c) \supset (a \supset c)))$	
$\mathfrak{m}_1 = (?, a \supset b)$	$\mathfrak{m}_0$
$\mathfrak{m}_2 = (!, (b \supset c) \supset (a \supset c))$	$\mathfrak{m}_1$
$\mathfrak{m}_3 = (?, b \supset c)$	$\mathfrak{m}_2$
$\mathfrak{m}_4 = (!, a \supset c)$	$\mathfrak{m}_3$
$\mathfrak{m}_5 = (?, a)$	$\mathfrak{m}_4$
$\mathfrak{m}_6 = (?, a)$	$\mathfrak{m}_1$
$\mathfrak{m}_7 = (!, b)$	$\mathfrak{m}_6$
$\mathfrak{m}_8 = (?, b)$	$\mathfrak{m}_3$
$\mathfrak{m}_9 = (!, c)$	$\mathfrak{m}_8$
$\mathfrak{m}_{10} = (!, c)$	$\mathfrak{m}_5$

$\mathfrak{m}_0 = (!, a \vee b \supset b \vee a)$	
$\mathfrak{m}_1 = (?, a \vee b)$	$\mathfrak{m}_0$
$\mathfrak{m}_2 = (?, \vee)$	$\mathfrak{m}_1$
$\mathfrak{m}_3 = (!, a)$	$\mathfrak{m}_2$
$\mathfrak{m}_4 = (!, b \vee a)$	$\mathfrak{m}_1$
$\mathfrak{m}_5 = (?, \vee)$	$\mathfrak{m}_4$
$\mathfrak{m}_6 = (!, a)$	$\mathfrak{m}_5$

$\mathfrak{m}_0 = (!, a \vee b \supset b \vee a)$	
$\mathfrak{m}_1 = (?, a \vee b)$	$\mathfrak{m}_0$
$\mathfrak{m}_2 = (?, \vee)$	$\mathfrak{m}_1$
$\mathfrak{m}_3 = (!, b)$	$\mathfrak{m}_2$
$\mathfrak{m}_4 = (!, b \vee a)$	$\mathfrak{m}_1$
$\mathfrak{m}_5 = (?, \vee)$	$\mathfrak{m}_4$
$\mathfrak{m}_6 = (!, b)$	$\mathfrak{m}_5$

$\mathfrak{m}_0 = (!, a \supset \neg\neg a)$	
$\mathfrak{m}_1 = (?, a)$	$\mathfrak{m}_0$
$\mathfrak{m}_2 = (!, \neg\neg a)$	$\mathfrak{m}_1$
$\mathfrak{m}_3 = (!, \neg a)$	$\mathfrak{m}_2$
$\mathfrak{m}_4 = (!, a)$	$\mathfrak{m}_3$
$\mathfrak{m}_5 = (?, \bot)$	$\mathfrak{m}_4$

$\mathfrak{m}_0 = (!, \neg\neg a \supset a)$	
$\mathfrak{m}_1 = (?, \neg\neg a)$	$\mathfrak{m}_0$
$\mathfrak{m}_2 = (?, \neg a)$	$\mathfrak{m}_1$
$\mathfrak{m}_3 = (?, a)$	$\mathfrak{m}_2$
$\mathfrak{m}_4 = (!, a)$	$\mathfrak{m}_1$

$\mathfrak{m}_0 = (!, (a \wedge \neg a) \supset c)$	
$\mathfrak{m}_1 = (?, a \wedge \neg a)$	$\mathfrak{m}_0$
$\mathfrak{m}_2 = (?, \wedge_1)$	$\mathfrak{m}_1$
$\mathfrak{m}_3 = (!, a)$	$\mathfrak{m}_2$
$\mathfrak{m}_4 = (?, \wedge_2)$	$\mathfrak{m}_1$
$\mathfrak{m}_5 = (!, \neg a)$	$\mathfrak{m}_4$
$\mathfrak{m}_6 = (?, a)$	$\mathfrak{m}_5$
$\mathfrak{m}_7 = (!, \bot)$	$\mathfrak{m}_6$

$\mathfrak{m}_0 = (!, a \vee b \supset a)$	
$\mathfrak{m}_1 = (?, a \vee b)$	$\mathfrak{m}_0$
$\mathfrak{m}_2 = (?, \vee)$	$\mathfrak{m}_1$
$\mathfrak{m}_3 = (!, a)$	$\mathfrak{m}_2$
$\mathfrak{m}_4 = (!, a)$	$\mathfrak{m}_1$

$\mathfrak{m}_0 = (!, \neg\forall x \neg P(x) \supset \exists x P(x))$	
$\mathfrak{m}_1 = (?, \neg\forall x \neg P(x)$	$\mathfrak{m}_0$
$\mathfrak{m}_2 = (?, \forall x \neg P(x)$	$\mathfrak{m}_1$
$\mathfrak{m}_3 = (?, [w/x])$	$\mathfrak{m}_2$
$\mathfrak{m}_4 = (!, \neg P(w))$	$\mathfrak{m}_3$
$\mathfrak{m}_5 = (?, P(w))$	$\mathfrak{m}_4$
$\mathfrak{m}_6 = (!, \exists x P(x))$	$\mathfrak{m}_1$
$\mathfrak{m}_7 = (?, \exists)$	$\mathfrak{m}_6$
$\mathfrak{m}_8 = (!, P(w))$	$\mathfrak{m}_7$

$\mathfrak{m}_0 = (!, \neg\forall x \neg P(x) \supset \exists x P(x))$	
$\mathfrak{m}_1 = (?, \neg\forall x \neg P(x)$	$\mathfrak{m}_0$
$\mathfrak{m}_2 = (?, \forall x \neg P(x)$	$\mathfrak{m}_1$
$\mathfrak{m}_3 = (!, \bot)$	$\mathfrak{m}_2$

$\mathfrak{m}_0 = (!, \exists y \forall x R(x,y) \supset \forall x \exists y R(x,y))$	
$\mathfrak{m}_1 = (?, \exists y \forall x R(x,y))$	$\mathfrak{m}_0$
$\mathfrak{m}_2 = (!, \forall x \exists y R(x,y))$	$\mathfrak{m}_1$
$\mathfrak{m}_3 = (?, \forall [w/x])$	$\mathfrak{m}_2$
$\mathfrak{m}_4 = (!, \exists y R(w,y))$	$\mathfrak{m}_3$
$\mathfrak{m}_5 = (?, \exists)$	$\mathfrak{m}_4$
$\mathfrak{m}_6 = (?, \exists)$	$\mathfrak{m}_1$
$\mathfrak{m}_7 = (!, \forall x R(x,z))$	$\mathfrak{m}_6$
$\mathfrak{m}_8 = (?, \forall [w/z])$	$\mathfrak{m}_7$
$\mathfrak{m}_9 = (!, R(w,z))$	$\mathfrak{m}_8$
$\mathfrak{m}_{10} = (!, R(w,z))$	$\mathfrak{m}_5$

$\mathfrak{m}_0 = (!, \exists x (P(x) \supset \forall y P(y)))$	
$\mathfrak{m}_1 = (?, \exists)$	$\mathfrak{m}_0$
$\mathfrak{m}_2 = (!, P(k) \supset \forall y P(y))$	$\mathfrak{m}_1$
$\mathfrak{m}_3 = (?, P(k))$	$\mathfrak{m}_2$
$\mathfrak{m}_4 = (!, \forall y P(y))$	$\mathfrak{m}_3$
$\mathfrak{m}_5 = (?, \forall [w/y])$	$\mathfrak{m}_4$
$\mathfrak{m}_6 = (!, P(w) \supset \forall y P(y))$	$\mathfrak{m}_1$
$\mathfrak{m}_7 = (?, P(w))$	$\mathfrak{m}_6$
$\mathfrak{m}_8 = (!, P(w))$	$\mathfrak{m}_5$

Remark 1. *All the games are won by the Proponent: they either do not admit further Opponent's moves or they end with an Opponent's assertion of* $\bot$.

The two games for the formula $a \vee b \supset b \vee a$ *have a common prefix, and they first differ on an Opponent's move. In one game the Opponent chooses to assert* a *in the defense move* $\mathfrak{m}_4$ *while in the other game the Opponent chooses to assert* b*. In any case the Proponent wins.*

The Proponent wins the game for the formula $a \vee b \supset a$ *even if this latter formula is not a tautology of first-order classical logic. Note that if the Opponent had chosen to assert the formula* b *instead of the formula* a *on move* 3*, then the Proponent would have had no chance of winning. In the game for the formula* $\exists y \forall x R(x,y) \supset \forall x \exists y R(x,y)$*, the player does not defend immediately against the attack move* $\mathfrak{m}_5$*. Instead, she delays her defense until* *the last move. Remark that the last move of all games won by the Proponent ending in a Proponent's move are defense moves and assertions of an atomic formula. In all games formulas asserted by the Proponent are positive sub-formulas of the formula about which the game is played. Formulas asserted by the Opponent are negative sub-formulas of the formula about which the game is played. In each game atomic formulas asserted by the Proponent are both positive and negative sub-formulas of the formula about which the game is played.*

3.2.6 Properties of games

We systematize the observations on the games just made with some simple propositions

Proposition 3.1. *Let A be an arbitrary formula and $\mathcal{G}$ and arbitrary game for A. If $(\star, B)$ is an assertion move in $\mathcal{G}$, then B is a Gentzen subformula of A.*

Proof. By induction on the length of $\mathcal{G}$. □

Proposition 3.2. *Let A be an arbitrary formula and $\mathcal{G} = \mathcal{G}'\mathfrak{m}^{\mathbf{P}}$ be a finite game for A. if $\mathcal{G}$ is* **won** *by* **P** *then m asserts an atomic Gentzen subformula B of A.*

Proof. Suppose, to reach a contradiction, that the last move m of $\mathcal{G}$ is not a defense move. Then it is an attack move of the form $(?, s)$. By the definition of games there is a preceding **O**-move $\mathfrak{m}_k$ that asserts some formula C, $\phi(\mathfrak{m}) = \mathfrak{m}_k$ and s is a question on C. Then the augmented sequence $\mathcal{G}\mathfrak{n}$ where $\mathfrak{n} = (!, D)$, $\phi(\mathfrak{n}) = \mathfrak{m}$ and D is an answer to the question s on the formula B, is a game. This contradicts the fact that $\mathcal{G}$ is won by **P**. Thus, $\mathfrak{m}$ must be a defense move $(!, B)$. If B it is not atomic we reach again a contradiction: in fact by adding a move $\mathfrak{n} = (?, s)$ to $\mathcal{G}$ where s is a question on B we obtain a game. Thus, we must conclude that B is an atomic formula. □

Proposition 3.3. *Let A be an arbitrary formula and $\mathcal{G}$ an arbitrary game for A. If $(\star, B)$ is an assertion move in $\mathcal{G}$ that is a* **P**-*move (resp. a* **O**-*move) then B is a positive (resp. negative) Gentzen subformula of A.*

Proof. Suppose that the proposition holds for all games $\mathcal{G}$ having length n, and let $\mathcal{G}'$ be a game having length $n + 1$. Let $\mathfrak{m}_n$ be the last move of G'. Suppose that $\mathfrak{m}_n$ is a **P**-move (the argument for **O**-moves runs in a very similar way) We have three cases.

1. If $\mathfrak{m}_n$ is not an assertion, the proposition holds automatically by induction hypothesis.

2. If $\mathfrak{m}_n$ is a defense-move asserting some formula B, then, since $\mathcal{G}'$ is a game, $\mathfrak{m}_n$ is enabled by some **O**-move $\mathfrak{m}_k$ with $(k < n)$. If $m_k := (?, C)$ (the other cases are easier) then it is an attack against $\mathfrak{m}_{k-1}$ and $\mathfrak{m}_{k_1}$ is a **P**-move that asserts $C \supset B$. By induction hypothesis $C \supset B$ is a positive subformula of A, and C is a negative subformula of A. Thus B is a positive subformula of A by definition.

3. If $\mathfrak{m}_n$ is an assertion and an attack, let B be the asserted formula. As before, there must exist an enabler of $\mathfrak{m}_n$, call it m_k $(k < n)$, $\mathfrak{m}_k$ is necessarily a **O**-move that asserts the formula $B \supset C$. By induction hypothesis, this last formula is a negative subformula of A, thus B is a positive subformula of A by definition.

□

Proposition 3.4. *Let A be an arbitrary formula and $\mathcal{G}$ an arbitrary game for A. If $(\star, B)$ is an assertion move made by* **P** *and B is an atomic formula, then B is both a negative and positive Gentzen subformula of A.*

Proof. Direct consequence of Proposition 3.3 and of the condition 3 in the definition of game. □

3.2.7 Strategies

As we have discussed in remark 1 the game for the formula $a \vee b \supset a$ is won by the Proponent but by mere accident: if the Opponent had chosen to assert b instead of a the Proponent would not have had a chance to win. This means that the Proponent *cannot* win a game on that formula *no matter how the Opponent chooses* to act in the game. On the contrary, the Proponent *can* win a game on the formula $a \vee b \supset b \vee a$ *no matter how* the Opponent choose to act in the game. This means that there is a Proponent winning strategy for the formula $a \vee a \supset b \vee a$ and no winning strategy for the formula $a \vee b \supset a$.

Intuitively speaking a strategy for a game $\mathcal{G}$ is a function. A function that specifies, at each moment of the game, which move a player must play according to the moves previously played (the history of the game). A strategy is *winning* when the player that follows the strategy wins whatever the history of the game is. As long as each move of the player that follows the strategy is determined by the strategy itself, it can be concluded that the game history varies only according to the moves of his opponent. We informally describe how a strategy should operate and then formalize this notion. Imagine being engaged in a game $\mathcal{G}$, that the last move of $\mathcal{G}$ was played according to the strategy, and that it is now your opponent's turn to play. Your opponent could extend the game in different ways: for example if you are playing chess, you are white, and you just made your first move by moving a pawn to a certain position of the chessboard, black can in turn move a pawn or move a horse. If you are playing according to the strategy, the strategy should tell you how to react against either type of move. If black moves a pawn to $C6$ and you just moved your pawn to $C3$, then move the horse to $H3$. If black moves a horse to $H6$ and you just moved your pawn to $C3$ then move your pawn to $B4$. Therefore, a strategy can be viewed as tree in which each node is a move in the game, the moves of my opponent have at most one daughter, and my moves have as many daughters as there are available moves for my opponent. A tree can be seen as a prefix-closed set of sequence over an alphabet. Since our games are sequences over the alphabet of moves we can define strategies in the following manner:

Definition 3.5. *A strategy $\mathcal{S}$ for a formula A is a non-empty prefix-closed set of games for A such that:*

1. *if $\mathcal{G}\mathfrak{m}^{\mathbf{P}}$ and $\mathcal{G}\mathfrak{n}^{\mathbf{P}}$ belongs to $\mathcal{S}$ then $\mathfrak{m} = \mathfrak{n}$;*
2. *if $\mathcal{G} = \mathcal{G}'\mathfrak{m}^{\mathbf{P}} \in \mathcal{S}$ then $\mathcal{G}\mathfrak{n}^{\mathbf{O}} \in \mathcal{S}$ for each move $\mathfrak{n}$ legal for $\mathcal{G}$;*
3. *if $\mathcal{G} = \mathcal{G}'\mathfrak{m}^{\mathbf{O}} \in \mathcal{S}$ and $\mathfrak{m} = (?, \exists)$ or $\mathfrak{m} = (?, \vee)$ then, there is an $\mathfrak{n}$ such that $\mathcal{G}\mathfrak{n}^{\mathbf{P}} \in \mathcal{S}$ and $\mathfrak{n}$ is enabled by $\mathfrak{m}$.*

A strategy $\mathcal{S}$ is winning if and only if every maximal, with respect to the prefix order, game of the strategy is won by **P**.

Condition 3 in the definition above precludes the Proponent to delay a defense against an existential attack or a disjunctive attack. Consequently, games like the one for the formula $\exists y \forall x R(x, y) \supset \forall x \exists y R(x, y)$ presented in subsection 3.2.5 cannot belong to a strategy.

Proposition 3.5. *Let $\mathcal{S}$ be an arbitrary strategy and let $\mathcal{G}$ be a game in $\mathcal{S}$ that ends in* **P***-move. The family of moves $(\mathfrak{m}_{k_i})_{(i \in I)}$ such that $\mathcal{G}\mathfrak{m}_{k_i} \in \mathcal{S}$, is a finite family.*

By the proposition above any winning strategy is a finitely branching tree in which each branch is finite, thus by König's lemma any winning strategy is a finite tree.

3.2.8 Some examples of strategies

Let a, b, c and d be propositional variables and R be a binary predicate variable. According to definition 3.5 a strategy for a formula A is a tree in which each branch is a game for A. We will thus represent strategies as trees.

$$\begin{array}{l|l} \mathfrak{m}_0 = (!, a \vee b \supset b \vee a) & \\ \mathfrak{m}_1 = (?, a \vee b) & \mathfrak{m}_0 \\ \mathfrak{m}_2 = (?, \vee) & \mathfrak{m}_1 \end{array}$$

$$\nearrow \qquad\qquad \nwarrow$$

$$\begin{array}{l|l} \mathfrak{m}_3 = (!, b) & \mathfrak{m}_2 \\ \mathfrak{m}_4 = (!, a \vee b) & \mathfrak{m}_1 \\ \mathfrak{m}_5 = (?, \vee) & \mathfrak{m}_4 \\ \mathfrak{m}_6 = (!, b) & \mathfrak{m}_5 \end{array} \qquad \begin{array}{l|l} \mathfrak{m}_3 = (!, a) & \mathfrak{m}_2 \\ \mathfrak{m}_4 = (!, a \vee b) & \mathfrak{m}_1 \\ \mathfrak{m}_5 = (?, \vee) & \mathfrak{m}_4 \\ \mathfrak{m}_6 = (!, a) & \mathfrak{m}_5 \end{array}$$

$$\begin{array}{l|l} \mathfrak{m}_0 = (!, a \vee b \supset a) & \\ \mathfrak{m}_1 = (?, a \vee a) & \mathfrak{m}_0 \\ \mathfrak{m}_2 = (?, \vee) & \mathfrak{m}_1 \end{array}$$

$$\nearrow \qquad\qquad \nwarrow$$

$$\begin{array}{l|l} \mathfrak{m}_3 = (!, a) & \mathfrak{m}_2 \\ \mathfrak{m}_4 = (!, a) & \mathfrak{m}_1 \end{array} \qquad \begin{array}{l|l} \mathfrak{m}_3 = (!, b) & \mathfrak{m}_2 \end{array}$$

$$
\begin{array}{ll|l}
\mathfrak{m}_0 & = (!, ((a \supset (b \supset c)) \supset (((b \supset c) \supset d) \supset (a \supset d)) & \\
\mathfrak{m}_1 & = (?, a \supset (b \supset c)) & \mathfrak{m}_0 \\
\mathfrak{m}_2 & = (!, ((b \supset c) \supset d) \supset (a \supset d) & \mathfrak{m}_1 \\
\mathfrak{m}_3 & = (?, (b \supset c) \supset d) & \mathfrak{m}_2 \\
\mathfrak{m}_4 & = (!, a \supset d) & \mathfrak{m}_3 \\
\mathfrak{m}_5 & = (?, a) & \mathfrak{m}_4 \\
\mathfrak{m}_6 & = (?, b \supset c) & \mathfrak{m}_3
\end{array}
$$

$$
\begin{array}{ll|l}
\mathfrak{m}_7 & = (?, b) & \mathfrak{m}_6 \\
\mathfrak{m}_8 & = (?, a) & \mathfrak{m}_1 \\
\mathfrak{m}_9 & = (!, b \supset c) & \mathfrak{m}_8 \\
\mathfrak{m}_{10} & = (?, b) & \mathfrak{m}_9 \\
\mathfrak{m}_{11} & = (!, c) & \mathfrak{m}_{10} \\
\mathfrak{m}_{12} & = (!, c) & \mathfrak{m}_7
\end{array}
\qquad
\begin{array}{ll|l}
\mathfrak{m}_7 & = (!, d) & \mathfrak{m}_6 \\
\mathfrak{m}_8 & = (!, d) & \mathfrak{m}_5
\end{array}
$$

$$
\begin{array}{ll|l}
\mathfrak{m}_0 & = (!, \exists y \forall x R(x,y) \supset \forall x \exists y R(x,y) & \\
\mathfrak{m}_1 & = (?, \exists y \forall x R(x,y)) & \mathfrak{m}_0 \\
\mathfrak{m}_2 & = (!, \forall x \exists y R(x,y)) & \mathfrak{m}_1 \\
\mathfrak{m}_3 & = (?, \forall [w/x]) & \mathfrak{m}_2 \\
\mathfrak{m}_4 & = (?, \exists) & \mathfrak{m}_1 \\
\mathfrak{m}_5 & = (!, \forall x R(x,z)) & \mathfrak{m}_4 \\
\mathfrak{m}_6 & = (?, \forall [w/x])) & \mathfrak{m}_5 \\
\mathfrak{m}_7 & = (!, R(w,z)) & \mathfrak{m}_6 \\
\mathfrak{m}_8 & = (!, \exists y R(w,y)) & \mathfrak{m}_3 \\
\mathfrak{m}_9 & = (?, \exists) & \mathfrak{m}_8 \\
\mathfrak{m}_{10} & = (!, R(w,z)) & \mathfrak{m}_9
\end{array}
$$

3.3 The sequent calculus SLK

We now present the sequent calculus SLK. (Strategic LK). SLK is a first-order version of the calculus LKQ studied by Herbelin in his PhD dissertation [68]. LKQ is a Kleene style sequent calculus: the active formula of introduction rules is present in the premises of the rule. LKQ differs from a Kleene-style calculus like GKc [132] because of a restriction on the use of the left introduction rule for the implication connective. Our system SLK is obtained from LKQ by adding the quantifier rules and imposing a restriction on the use of the right introduction rule for the disjunction and the existential quantifier connectives.

Definition 3.6. *The sequent calculus SLK is defined by the rules in Table 3.1. A sequent is an expression* $\Gamma \vdash \Delta$ *where* Γ *and* Δ *are finite multisets of formulas. The bold formulas occurrences in the conclusion of a rule are the active formulas of the rule. Greek upper-case letters* $\Gamma, \Delta, \Sigma, \Pi \ldots$ *stand for multisets of formulas. In the Id-rule A is of the form* $P(t_1, \ldots, t_n)$ *where P is a predicate variable with arity* $n \geq 0$ *and the* t_i *are terms, and there is no* $\bot$ *formula in* Γ*. In the* $\forall^{\mathsf{R}}$ *and* $\exists^{\mathsf{L}}$ *rules the variable y does not occur in the conclusion sequent, and it is called the proper parameter of the rule.*[2]

A derivation (or a proof) $\mathfrak{D}$ *of a sequent* $\Gamma \vdash \Delta$ *in SLK is a tree of sequents constructed according to the rules of SLK in which leaves are instances of Id-rules or* $\bot$*L-rules, all sequents of the form* $\Pi, \bot \vdash \Sigma$ *are leaves and whose root also called conclusion, is* $\Gamma \vdash \Delta$ *and in which the following restrictions on the use of the* $\supset^{\mathsf{L}}$, $\exists^{\mathsf{R}}$ *and* $\vee^{\mathsf{R}}$ *rules are respected*

1. *For any application of a left implication introduction rule, the formula occurrence A in the left-hand premise is active.*

$$\frac{\Gamma, A \supset B \vdash \mathbf{A}, \Delta \qquad \Gamma, A \supset B, B \vdash \Delta}{\Gamma, \mathbf{A} \supset \mathbf{B} \vdash \Delta} \supset^{\mathsf{L}}$$

2. *For any application of right existential introduction rule, the formula occurrence* $A[t/x]$ *is active in the premise:*

$$\frac{\Gamma \vdash \exists x A, \mathbf{A}[\mathbf{t}/\mathbf{x}], \Delta}{\Gamma \vdash \exists \mathbf{x} \mathbf{A}, \Delta} \exists^{\mathsf{R}}$$

3. *For any application of a right disjunction introduction rule, the formula occurrence* A_i *with* $i \in \{1, 2\}$ *is active in the premise*

$$\frac{\Gamma \vdash \mathbf{A_i}, A_1 \vee A_2, \Delta}{\Gamma \vdash \mathbf{A_1} \vee \mathbf{A_2}, \Delta} \vee^{\mathsf{R}}$$

[2] We can always suppose that every proper parameter is distinct in a derivation

Table 3.1: The SLK sequent calculus.

$$\frac{}{\Gamma, \boldsymbol{\bot} \vdash \Delta}\ \bot^{\mathsf{L}} \qquad \frac{}{\Gamma, A \vdash \mathbf{A}, \Delta}\ Id$$

$$\frac{\Gamma, A \supset B \vdash A, \Delta \qquad \Gamma, A \supset B, B \vdash \Delta}{\Gamma, \mathbf{A} \supset \mathbf{B} \vdash \Delta}\ \supset^{\mathsf{L}} \qquad \frac{\Gamma, A \vdash B, \Delta}{\Gamma \vdash \mathbf{A} \supset \mathbf{B}, \Delta}\ \supset^{\mathsf{R}}$$

$$\frac{\Gamma, A, A \wedge B \vdash \Delta}{\Gamma, \mathbf{A} \wedge \mathbf{B} \vdash \Delta}\ \wedge^{\mathsf{L}}_1 \qquad \frac{\Gamma, B, A \wedge B \vdash \Delta}{\Gamma, \mathbf{A} \wedge \mathbf{B} \vdash \Delta}\ \wedge^{\mathsf{L}}_2 \qquad \frac{\Gamma \vdash A, \Delta \qquad \Gamma \vdash B, \Delta}{\Gamma \vdash \mathbf{A} \wedge \mathbf{B}, \Delta}\ \wedge^{\mathsf{R}}$$

$$\frac{\Gamma, A \vee B, A \vdash \Delta \qquad \Gamma, A \vee B, B \vdash \Delta}{\Gamma, \mathbf{A} \vee \mathbf{B} \vdash \Delta}\ \vee^{\mathsf{L}}$$

$$\frac{\Gamma \vdash A, A \vee B, \Delta}{\Gamma \vdash \mathbf{A} \vee \mathbf{B}, \Delta}\ \vee^{\mathsf{R}}_1 \qquad \frac{\Gamma \vdash B, A \vee B, \Delta}{\Gamma \vdash \mathbf{A} \vee \mathbf{B}, \Delta}\ \vee^{\mathsf{R}}_2$$

$$\frac{\Gamma, A[y/x], \exists x A \vdash \Delta}{\Gamma, \exists \mathbf{x} \mathbf{A} \vdash \Delta}\ \exists^{\mathsf{L}} \qquad \frac{\Gamma \vdash A[t/x], \exists x A, \Delta}{\Gamma \vdash \exists \mathbf{x} \mathbf{A}, \Delta}\ \exists^{\mathsf{R}}$$

$$\frac{\Gamma, A[t/x], \forall x A \vdash \Delta}{\Gamma, \forall \mathbf{x} \mathbf{A} \vdash \Delta}\ \forall^{\mathsf{L}} \qquad \frac{\Gamma \vdash A[y/x], \Delta}{\Gamma \vdash \forall \mathbf{x} \mathbf{A}, \Delta}\ \forall^{\mathsf{R}}$$

A sequent $\Gamma \vdash \Delta$ *is said to be* derivable *or* provable *in the sequent calculus SLK whenever there exists a proof with conclusion* $\Gamma \vdash \Delta$. *The height* $|\mathfrak{D}|$ *of a proof* $\mathfrak{D}$ *is the number of nodes in the maximal branch of the proof tree minus* 1.

Example 1. *We give two examples of pairs of trees of sequents having the same root. Both trees in a pair are constructed using the rules of SLK. However, in each couple, only the second tree complies with the definition of derivation in SLK. Let* a, b *be propositional variables and* P *be a unary predicate variable.*

$$\dfrac{\dfrac{\dfrac{}{a, a \supset b, \vdash \mathbf{a}, b}\,Id}{a \supset b \vdash a, \mathbf{a \supset b}}\supset R \qquad \dfrac{\dfrac{}{b, a \supset b, a \vdash \mathbf{b}}\,Id}{b, a \supset b \vdash \mathbf{a \supset b}}\supset^{R}}{\dfrac{\mathbf{a \supset b} \vdash a \supset b}{\vdash (\mathbf{a \supset b}) \supset (\mathbf{a \supset b})}\supset^{R}}\supset^{L}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{}{a, a \supset b \vdash \mathbf{a}, b}\,Id \qquad \dfrac{}{b, a, a \supset b \vdash \mathbf{b}}\,Id}{\mathbf{a \supset b}, a \vdash b}\supset^{L}}{a \supset b \vdash \mathbf{a \supset b}}\supset^{R}}{\vdash (\mathbf{a \supset b}) \supset (\mathbf{a \supset b})}\supset^{R}$$

In the first tree for the formula $(a \supset b) \supset (a \supset b)$ *condition 1 of definition 3.6 is not respected. The sequent* $a \supset b \vdash a \supset b$ *is proved by the* $\supset L$ *rule. In the left-hand side premise of the rule the active formula is* $a \supset b$*, instead of* a*, which would respect condition 1 of definition 3.6.*

$$\dfrac{\dfrac{\dfrac{}{\forall x P(x), P(y) \vdash \mathbf{P(y)}, \exists x P(x)}\,Id}{\forall \mathbf{x}\, \mathbf{P(x)} \vdash P(y), \exists x P(x)}\forall^{L}}{\forall x\, P(x) \vdash \exists \mathbf{x}\, \mathbf{P(x)}}\exists^{R}$$

$$\dfrac{\dfrac{\dfrac{}{\forall x P(x), P(y) \vdash \mathbf{P(y)}, \exists x P(x)}\,Id}{\forall x P(x), P(y) \vdash \exists \mathbf{x}\, \mathbf{P(x)}}\exists^{R}}{\forall \mathbf{x}\, \mathbf{P(x)} \vdash \exists x\, P(x)}\forall^{L}$$

In the first tree of sequents, condition 2 of definition 3.6 is not respected. The sequent $\forall\, xP(x) \vdash P(x), \exists x P(x)$ *is the premise of the sequent* $\forall x\, P(x) \vdash \exists\, xP(x)$*. This latter sequent is proved from the former sequent by the use of* $\exists R$*-rule. In the former sequent the active formula is* $\forall x\, P(x)$ *contrarily to what condition 2 demands.*

3.3.1 Properties of SLK

In this section we state some proposition that are true about the sequent calculus SLK; whenever the proof of these proposition is standard, we will omit it. We recall that if $\mathfrak{D}$ is a derivation, then $|\mathfrak{D}|$ denote its height.

Proposition 3.6 (Inversion). *For any formula A and B for any multiset of formulas Γ and Δ:*

1. *if there is a derivation $\mathfrak{D}$ of $\Gamma \vdash A \wedge B, \Delta$, then there are derivation $\mathfrak{D}_1$ of $\Gamma \vdash A, \Delta$ and $\mathfrak{D}_2$ of $\Gamma \vdash B, \Delta$. Moreover, $|\mathfrak{D}_i| \leq |\mathfrak{D}|$ for $i \in \{1, 2\}$;*
2. *if there is a derivation $\mathfrak{D}$ of $\Gamma \vdash A \supset B, \Delta$, then there is a derivation $\mathfrak{D}_1$ of $\Gamma, A \vdash B, \Delta$. Moreover, $|\mathfrak{D}_1| \leq |\mathfrak{D}|$;*
3. *if there is a derivation $\mathfrak{D}$ of $\Gamma \vdash \forall x A, \Delta$, then there is a derivation $\mathfrak{D}_1$ of $\Gamma \vdash A[y/x], \Delta$ where y is a variable that does not appear in Γ Moreover, $|\mathfrak{D}_1| \leq |\mathfrak{D}|$;*
4. *if there is a derivation $\mathfrak{D}$ of $\Gamma, A \vee B \vdash \Delta$, then there are derivation $\mathfrak{D}_1$ of $\Gamma, A \vee B, A \vdash \Delta$ and $\mathfrak{D}_2$ of $\Gamma, A \vee B, B \vdash \Delta$. Moreover, $|\mathfrak{D}_i| \leq |\mathfrak{D}|$ for $i \in \{1, 2\}$.*

Proof. By induction on the height $|\mathfrak{D}|$ of $\mathfrak{D}$. □

Corollary 1. *For any formula A for any multiset of formula Γ if the main connective of A is either $\wedge, \supset$ or $\forall$ and the sequent $\Gamma \vdash A$ is provable, then there is a derivation $\mathfrak{D}$ of $\Gamma \vdash A$ in which A is active.*

Let $\Gamma = A_1, \ldots A_n$ be a multiset of formulas. We denote by $\Gamma[t/x]$ the multiset Γ in which the term t is substituted to each occurrence of x in A_i i.e., $\Gamma[t/x] = A_1[t/x], \ldots A_n[t/x]$.

Proposition 3.7. *For any multiset of formulas Γ and Δ if there is a derivation $\mathfrak{D}$ of the sequent $\Gamma \vdash \Delta$ then there is a derivation $\mathfrak{D}[t/y]$ of $\Gamma[t/y] \vdash \Delta[t/y]$ provided that no variable that is free in t became bound after the substitution and that t does not contain variables that are proper parameters of $\exists^{\mathrm{L}}$ or of $\forall^{\mathrm{R}}$. Moreover $|\mathfrak{D}[t/y]| \leq |\mathfrak{D}|$*

Proof. By induction on the height of $\mathfrak{D}$. □

Proposition 3.8. *Contraction and weakening are height preserving admissible in SLK: for any formula A for any multiset of formulas Γ and Δ:*

- *if there is a derivation $\mathfrak{D}$ of $\Gamma, A, A \vdash \Delta$ (resp. of $\Gamma \vdash A, A, \Delta$) and $\mathfrak{D}$ has height n, then there is a derivation $\mathfrak{D}_1$ of $\Gamma, A \vdash \Delta$ (resp. $\Gamma \vdash A, \Delta$) and the height of $\mathfrak{D}_1$ is (at most) n;*

- *if there is a derivation $\mathfrak{D}$ of $\Gamma \vdash \Delta$ and $\mathfrak{D}$ has height n then there is a derivation $\mathfrak{D}_1$ of $\Gamma, A \vdash \Delta$ (resp. $\Gamma \vdash A, \Delta$) and the height of $\mathfrak{D}_1$ is (at most) n.*

Proof. By induction on the height of $\mathfrak{D}$ using proposition 3.6. □

Proposition 3.9. *For all formula A and all multiset of formulas Γ and Δ there is a derivation $\mathfrak{D}$ of the sequent $\Gamma, A \vdash A, \Delta$. Moreover, in the derivation $\mathfrak{D}$ either the occurrence of A on the left of $\vdash$ is active or the occurrence of A on the right of $\vdash$ is active.*

Proof. Suppose that the proposition holds for all formulas B having depth smaller than n and let A be a formula of height n. We will detail the proof only for the case in which $A = C \supset D$. By induction hypothesis, there are proofs of $\mathfrak{D}_1$ of $\Gamma, C \supset D, C \vdash C, \Delta$ and $\mathfrak{D}_2$ of $\Gamma, C \supset D, D \vdash D, \Delta$. We have a problem if in the derivation $\mathfrak{D}_1$ the active occurrence of C is the one on the left of the turnstile, i.e. $\mathfrak{D}_1$ ends in some left introduction rule. If the main connective of C is $\forall$, $\wedge$ or $\supset$, then by corollary 1 we can conclude that there is a derivation $\mathfrak{D}'_1$ of $\Gamma, C \supset D, C \vdash C$ in which the occurrence of C on the right of the turnstile is active. Thus, we can apply $\supset^{\mathsf{L}}$ on $\mathfrak{D}'_1$ and $\mathfrak{D}_2$ to obtain the wanted result. If the main connective of C is $\exists$ or $\vee$ then C has, respectively, the form $\exists x C_1$ or $C_1 \vee C_2$. Let us consider the second case. By induction hypothesis and weakening admissibility, there are derivations of $\Gamma', C_1 \vee C_2, C_1 \vdash C_1, \Delta, C_1 \vee C_2$ and $\Gamma', C_1 \vee C_2, C_2 \vdash C_2, \Delta, C_1 \vee C_2$ where $\Gamma' = \Gamma, C_1 \vee C_2 \supset D$. First we construct the two following derivations:

$$\mathfrak{D}_A\left\{\dfrac{\dfrac{\vdots}{\Gamma', C_1 \vee C_2, C_1 \vdash C_1, \Delta, C_1 \vee C_2}}{\Gamma', C_1 \vee C_2, C_1 \vdash C_1 \vee C_2, \Delta}\,\vee^{\mathsf{R}}\right. \qquad \dfrac{\qquad}{} \!\!\!\! \begin{array}{c}\vdots\mathfrak{D}'_2 \\ \Gamma, C_1 \vee C_2 \supset D, D, C_1 \vdash D, \Delta\end{array}$$

$$\dfrac{\Gamma', C_1 \vee C_2, C_1 \vdash C_1 \vee C_2, \Delta \qquad \Gamma, C_1 \vee C_2 \supset D, D, C_1 \vdash D, \Delta}{\Gamma, C_1 \vee C_2 \supset D, C_1 \vdash D, \Delta}\,\supset^{\mathsf{L}}$$

$$\mathfrak{D}_B\left\{\dfrac{\dfrac{\vdots}{\Gamma', C_1 \vee C_2, C_2 \vdash C_2, \Delta, C_1 \vee C_2}}{\Gamma', C_1 \vee C_2, C_1 \vdash C_1 \vee C_2, \Delta}\,\vee^{\mathsf{R}}\right. \qquad \begin{array}{c}\vdots\mathfrak{D}''_2 \\ \Gamma, C_1 \vee C_2 \supset D, D, C_2 \vdash D, \Delta\end{array}$$

$$\dfrac{\Gamma', C_1 \vee C_2, C_1 \vdash C_1 \vee C_2, \Delta \qquad \Gamma, C_1 \vee C_2 \supset D, D, C_2 \vdash D, \Delta}{\Gamma, C_1 \vee C_2 \supset D, C_2 \vdash D, \Delta}\,\supset^{\mathsf{L}}$$

where $\mathfrak{D}_2'$ and $\mathfrak{D}_2''$ are obtained from $\mathfrak{D}_2$ by weakening. We can now construct a derivation of $\Gamma, C_1 \vee C_2 \supset D \vdash C_1 \vee C_2 \supset D, \Delta$

$$\dfrac{\dfrac{\overset{\vdots \mathfrak{D}_A}{\Gamma, C_1 \vee C_2 \supset D, C_1 \vdash D, \Delta} \qquad \overset{\vdots \mathfrak{D}_B}{\Gamma, C_1 \vee C_2 \supset D, C_2 \vdash D, \Delta}}{\Gamma, C_1 \vee C_2 \supset D, C_1 \vee C_2 \vdash D, \Delta} \vee^{\mathsf{L}}}{\Gamma, C_1 \vee C_2 \supset D \vdash C_1 \vee C_2 \supset D, \Delta} \supset^{\mathsf{R}}$$

□

Proposition 3.10. *For any formula A and B for any multiset of formulas,* Γ *and* Δ *the sequents:*

1. $\Gamma, A, A \supset B \vdash B, \Delta$
2. $\Gamma, A \vdash A \vee B, \Delta$
3. $\Gamma, B, \vdash A \vee B, \Delta$
4. $\Gamma, A[y/x] \vdash \exists x A, \Delta$

are derivable in SLK. In 4 y is a variable that does not appears in $\Gamma, \exists x A, \Delta$.

Proof. It is an immediate consequence of propositions 3.9. and 3.6.

□

Proposition 3.11. *The cut rule*

$$\dfrac{\Gamma \vdash A, \Delta \qquad \Gamma, A \vdash \Delta}{\Gamma \vdash \Delta} \mathsf{cut}$$

is admissible in SLK : for any formula A for any multiset of formulas Γ *and* Δ*, if the sequents* $\Gamma \vdash A, \Delta$ *and* $\Gamma, A \vdash \Delta$ *are provable, then the sequent* $\Gamma \vdash \Delta$ *is provable.*

Proof. By nested inductions on the depth of A (the cut formula), the height of the derivation $\mathfrak{D}_1$ of $\Gamma \vdash A, \Delta$ and the height of the derivation $\mathfrak{D}_2$ of $\Gamma, A \vdash \Delta$. More precisely, we appeal to the induction hypothesis either with a strictly smaller cut formula, or with an identical cut formula and two derivations, one of which is strictly smaller while the other stays the same; the cut admissibility proof follows the usual path of case analysis on the active formula of

$\mathfrak{D}_1$ and $\mathfrak{D}_2$. We detail two cases. Suppose that the cut-fomula A is $B \supset C$. We have a derivation $\mathfrak{D}_1$ of $\Gamma, B \supset C \vdash \Delta$ and a derivation $\mathfrak{D}_2$ of $\Gamma \vdash B \supset C, \Delta$. Moreover, suppose that in both derivation the cut formula $B \supset C$ is active. This means that $\mathfrak{D}_1$ and $\mathfrak{D}_2$ have the form:

$$\dfrac{\begin{matrix}\vdots\mathfrak{D}_{1.1} \\ \Gamma, B \supset C \vdash B, \Delta\end{matrix} \qquad \begin{matrix}\vdots\mathfrak{D}_{1.2} \\ C, B \supset C, \Gamma \vdash \Delta\end{matrix}}{\Gamma, B \supset C \vdash \Delta} \supset^{\mathsf{L}} \qquad\qquad \dfrac{\begin{matrix}\vdots\mathfrak{D}_{2.1} \\ \Gamma, B \vdash C, \Delta\end{matrix}}{\Gamma \vdash B \supset C, \Delta} \supset^{\mathsf{R}}$$

we obtain a derivation of $\Gamma \vdash \Delta$ As follows: we first construct a derivation $\mathfrak{D}_A$ of $\Gamma \vdash C, \Delta$, and a derivation $\mathfrak{D}_B$ of $C, \Gamma \vdash \Delta$, using the admissibility of the cut rule either on derivations that are smaller than $\mathfrak{D}_1$ or $\mathfrak{D}_2$ or on formulas that are smaller than $B \supset C$. In what follows $\mathfrak{D}_{1.1}{}'$, $\mathfrak{D}_2{}'$ and $\mathfrak{D}_2{}''$ are obtained from respectively $\mathfrak{D}_{1.1}$ and $\mathfrak{D}_2$ by height preserving admissibility of the weakening rule. For the sake of clarity we underline the cut-formula of each cut-rule instance.

$$\left.\dfrac{\dfrac{\begin{matrix}\vdots\mathfrak{D}_{1.1}{}' \\ \Gamma, \underline{B \supset C} \vdash B, C, \Delta\end{matrix} \qquad \begin{matrix}\vdots\mathfrak{D}_2' \\ \Gamma \vdash \underline{B \supset C}, B, C, \Delta\end{matrix}}{\Gamma \vdash \underline{B}, C, \Delta} \qquad \begin{matrix}\vdots\mathfrak{D}_{2.1} \\ \Gamma, \underline{B} \vdash C, \Delta\end{matrix}}{\Gamma \vdash C, \Delta}\right\}\mathfrak{D}_A$$

$$\left.\dfrac{\begin{matrix}\vdots\mathfrak{D}_{1.2} \\ C, \Gamma, \underline{B \supset C} \vdash \Delta\end{matrix} \qquad \begin{matrix}\vdots\mathfrak{D}_2'' \\ C, \Gamma \vdash \underline{B \supset C}, \Delta\end{matrix}}{C, \Gamma \vdash \Delta}\right\}\mathfrak{D}_B$$

We then put together $\mathfrak{D}_A$ and $\mathfrak{D}_B$ using an instance of the cut-rule with cut-formula C. Since C is a formula strictly smaller than $B \supset C$ this application of the cut-rule is allowed by the induction hypothesis.

$$\dfrac{\begin{matrix}\vdots\mathfrak{D}_A \\ \Gamma \vdash \underline{C}, \Delta\end{matrix} \qquad \begin{matrix}\vdots\mathfrak{D}_B \\ \Gamma, \underline{C} \vdash \Delta\end{matrix}}{\Gamma \vdash \Delta}$$

Now suppose that $B \supset C$ is not active in one of the two derivations $\mathfrak{D}_1$ and $\mathfrak{D}_2$. Suppose it is not active in $\mathfrak{D}_2$ e.g., $\mathfrak{D}_2$ as the form:

$$\dfrac{\begin{matrix}\vdots\mathfrak{D}_{2.1} & \vdots\mathfrak{D}_{2.2}\\ \Gamma \vdash F_1, B \supset C, \Delta' & \Gamma \vdash F_2, B \supset C, \Delta'\end{matrix}}{\Gamma \vdash F_1 \wedge F_2, B \supset C, \Delta'}$$

we want to obtain a derivation of the sequent $\Gamma \vdash F_1 \wedge F_2, \Delta'$. A derivation of this sequent can be constructed by the following method: we first construct two derivations $\mathfrak{D}_A$ and $\mathfrak{D}_B$ of the sequents $\Gamma \vdash \Delta', F_1 \wedge F_2, F_1$ and $\Gamma \vdash \Delta', F_1 \wedge F_2, F_1$:

$$\left.\dfrac{\begin{matrix}\vdots\mathfrak{D}'_1 & \vdots\mathfrak{D}_{2.1}{}'\\ \Gamma, \underline{B \supset C} \vdash \Delta', F_1 \wedge F_2, F_1 & \Gamma \vdash F_1, F_1 \wedge F_2, \underline{B \supset C}, \Delta'\end{matrix}}{\Gamma \vdash \Delta', F_1 \wedge F_2, F_1}\right\}\mathfrak{D}_A$$

$$\left.\dfrac{\begin{matrix}\vdots\mathfrak{D}''_1 & \vdots\mathfrak{D}'_{2.2}\\ \Gamma, \underline{B \supset C} \vdash \Delta', F_1 \wedge F_2, F_2 & \Gamma \vdash F_2, F_1 \wedge F_2, \underline{B \supset C}, \Delta'\end{matrix}}{\Gamma \vdash \Delta', F_2 \wedge F_2, F_2}\right\}\mathfrak{D}_B$$

we put together the two derivations $\mathfrak{D}_A$ and $\mathfrak{D}_B$ using the $\wedge R$ rule and we obtain the wanted derivation of the sequent $\Gamma \vdash \Delta', F_1 \wedge F_2$ by admissibility of the contraction rule.

$$\dfrac{\dfrac{\begin{matrix}\vdots\mathfrak{D}_A & \vdots\mathfrak{D}_B\\ \Gamma \vdash \Delta', F_1 \wedge F_2, F_1 & \Gamma \vdash \Delta', F_1 \wedge F_2, F_2\end{matrix}}{\Gamma \vdash \Delta', F_1 \wedge F_2, F_1 \wedge F_2}\ \wedge^{\mathsf{R}}}{\Gamma \vdash \Delta', F_1 \wedge F_2}\ \mathsf{C}^{\mathsf{R}}$$

□

3.3.2 SLK: soundness and completeness

We now prove that SLK is sound and complete for classical logic. In order to prove this fact we show that a sequent $\Gamma \vdash \Delta$ is provable in SLK if and only if, it is provable in the sequent calculus system GKc [132]. GKc is sound and complete for first-order classical logic. To obtain GKc one simply drop the right-rule restrictions of SLK (condition 2 and 3 of definition 3.6) as well as the left-implication rule restriction (condition 1 of definition 3.6). Moreover, one adds the active formula of each right introduction rule to the premises of the rule (as described in subsection 1.7.7 of chapter 1). Thus, one obtains that the $\supset^{\mathsf{R}}$, $\wedge^{\mathsf{R}}$ and $\forall^{\mathsf{R}}$-rules of GKc have the following form:

$$\frac{\Gamma, A \vdash B, A \supset B, \Delta}{\Gamma \vdash A \supset B, \Delta} \supset^{\mathsf{R}} \qquad \frac{\Gamma \vdash A, A \wedge B, \Delta \qquad \Gamma \vdash B, A \wedge B, \Delta}{\Gamma \vdash A \wedge B, \Delta} \wedge^{\mathsf{R}}$$

$$\frac{\Gamma \vdash A[y/x], \forall x A, \Delta}{\Gamma \vdash \forall x A, \Delta} \forall^{\mathsf{R}}$$

Lemma 3.1. *For any multiset of formulas Γ and Δ, the sequent $\Gamma \vdash \Delta$ is provable in SLK if and only if it is provable in GKc.*

Proof. To show that if the sequent $\Gamma \vdash \Delta$ is provable in SLK then it is provable in GKc we first remark that the initial rules $\perp^{\mathsf{L}}$ and *Id*-rules of the two systems are the same. We thus show that any other rule of SLK is admissible in GKc i.e., given a derivation of the premises of a rule R of SLK in the system GKc there is a derivation of the conclusion of the rule R of SLK in the system GKc. This is easily done by case analysis using the fact that weakening and contraction are admissible in GKc [132].

To prove that if the sequent $\Gamma \vdash \Delta$ is provable in GKc then $\Gamma \vdash \Delta$ is provable in SLK, suppose that for each derivation $\mathfrak{D}$ in GKc with height n and conclusion $\Gamma' \vdash C'$ there is a derivation $\mathfrak{D}'$ in SLK having the same conclusion. Let $\mathfrak{D}$ be a derivation of $\Gamma \vdash \Delta$ in GKc having height $n + 1$ and let R be the last rule application of $\mathfrak{D}$. If R is none of $\supset^{\mathsf{L}}$, $\exists^{\mathsf{R}}$ or $\vee^{\mathsf{R}}$ we just have to apply the induction hypothesis on the premises of R and the admissibility of weakening and contraction of SLK.

If R is $\supset^{\mathsf{L}}$ then the conclusion of $\mathfrak{D}$ is $\Gamma, A \supset B \vdash \Delta$ and, by induction hypothesis, we have an SLK derivation $\mathfrak{D}_1$ with conclusion $\Gamma, A \supset B \vdash A, \Delta$ and another SLK derivation $\mathfrak{D}_2$ with conclusion $B, \Gamma, A \supset B \vdash \Delta$. We construct a SLK derivation of the sequent $\Gamma, A \supset B \vdash \Delta$ below.

$$\dfrac{\begin{matrix} \vdots \mathfrak{D}_1 \\ \Gamma, A \supset B \vdash A, \Delta \end{matrix} \qquad \text{cut}\ \dfrac{\begin{matrix} \vdots \mathfrak{D}_A \\ A, \Gamma, A \supset B \vdash B, \Delta \end{matrix} \qquad \begin{matrix} \vdots \mathfrak{D}_2' \\ B, \Gamma, A, A \supset B \vdash \Delta \end{matrix}}{A, \Gamma, A \supset B \vdash \Delta}}{\Gamma, A \supset B \vdash \Delta} \text{cut}$$

Where the derivation $\mathfrak{D}_A$ exists by proposition 3.10, and $\mathfrak{D}_2'$ is obtained from $\mathfrak{D}_2$ by admissibility of weakening

If R is $\exists R$ or $\vee R$ then the conclusion of $\mathfrak{D}$ is (respectively) of the form $\Gamma \vdash \exists x A, \Delta$ or $\Gamma \vdash A_1 \vee A_2, \Delta$. By induction hypothesis, we have a SLK derivation $\mathfrak{D}'$ of its premise $\Gamma \vdash A[t/x], \exists x A, \Delta$ or $\Gamma \vdash A_i, A_1 \vee A_2, \Delta$ respectively, where $i \in \{1, 2\}$. We only provide the first case.

$$\dfrac{\dfrac{\begin{array}{c}\vdots \mathfrak{D}' \\ \Gamma \vdash A[t/x], \exists x A, \Delta\end{array} \qquad \begin{array}{c}\vdots \mathfrak{D}[t/y] \\ \Gamma, A[t/y] \vdash \exists x A, \Delta\end{array}}{\Gamma \vdash \exists x A, \exists x A, \Delta}\text{cut}}{\Gamma \vdash \exists x A, \Delta}\mathsf{C}^{\mathsf{R}}$$

where $\mathfrak{D}$ exists in virtue of proposition 3.10 and $\mathfrak{D}[t/y]$ exists in virtue of proposition 3.7 □

The below theorem follows immediately from the latter lemma, since GKc is sound and complete for classical logic, and we have proved that SLK proves exactly the same sequents that are provable in GKc.

Theorem 3.1. *The sequent system SLK is sound and complete for classical first-order logic.*

3.4 From strategies to derivations

In this section we will prove that given a winning strategy $\mathcal{S}$ for a formula A, $\mathcal{S}$ can be transformed into a SLK derivation of A. In fact, we have designed the calculus SLK in such a way that derivations in SLK have the 'shape' of winning strategies. The fact that all binary rules of SLK are context sharing (or additive) is motivated by the fact that we will recursively associate sequents to the nodes of a strategy starting from the root of the strategy. Using this methodology, it would be hard to split the sequent in the manner required by context splitting (or multiplicative) rules. The fact that all the left introduction rule of SLK carry the active formula of the conclusion in the premises of the rule is motivated by the fact that, as we will see below, left introduction "corresponds" to attack move by **P**. The player **P** can attack the *same* formula many times. This corresponds, in a SLK derivation, to a left introduction rule having the same active formula and being used many times in the derivation. The fact that the only right introduction rules in which the premise carries the active formula of the conclusion are the existential rule and the disjunction rule is motivated by the following fact: right introduction rules correspond to defense moves by **P** and **P** can answer to the same question on a formula many times only if that formula is an existential formula or a disjunction (condition 6 of definition 3.3). The two conditions 2 and 3 in the definition 3.6 of SLK derivation are the sequent-calculus counterparts of condition 3 in the definition 3.5 of Strategy. The attentive reader will surely notice that SLK obeys a *focusing principle* [8]: whenever we apply (bottom-up) an $\exists^{R}$ or a $\vee^{R}$ rule over a sequent $\Gamma \vdash \exists x B, \Delta$ (resp., $\Gamma \vdash A \vee B, \Delta$) we are obliged to apply right-rules until an implication, a conjunction, or a universally quantified formula occupies the position of $\exists x B$ (resp. $A \vee B$).

Let $\mathcal{S}$ be a strategy for a formula F and let $\mathcal{G}$ be a game in S. We define the **O**-sequence $\mathcal{G}|_{\mathbf{O}}$ of $\mathcal{G}$ to be the subsequence of $\mathcal{G}$ obtained by forgetting all its **P**-moves, i.e, if $\mathcal{G} = \mathfrak{m}_0, \mathfrak{m}_1, \dots \mathfrak{m}_n$ the **O**-sequence of $\mathcal{G}$ is $\mathfrak{m}_1, \dots \mathfrak{m}_{n-1}$. We define the **O**-tree $\mathcal{S}|_{\mathbf{O}}$ of a strategy $\mathcal{S}$ to be the prefix closed set of sequence

$$\mathcal{S}|_{\mathbf{O}} = \{\mathcal{G}|_{\mathbf{O}} \, | \mathcal{G} \in \mathcal{S}\}$$

Let $\mathcal{S}$ be a strategy for a formula F. We define a function Φ from $\mathcal{S}|_{\mathbf{O}}$ to a tree of sequents τ. The function Φ associates some sequent $\Gamma_{\mathcal{G}|_{\mathbf{O}}} \vdash \Delta_{\mathcal{G}|_{\mathbf{O}}}$ to each $\mathcal{G}|_{\mathbf{O}}$ in $\mathcal{S}|_{\mathbf{O}}$. Let us denote the empty sequence by ϵ

- if $\mathcal{G}|_{\mathbf{O}} = \epsilon$ then $\Gamma_{\mathcal{G}|_{\mathbf{O}}} = \emptyset$ and $\Delta_{\mathcal{G}|_{\mathbf{O}}} = F$
- if $\mathcal{G}|_{\mathbf{O}} = \mathcal{G}'|_{\mathbf{O}}(!, A)$ then $\Gamma_{\mathcal{G}|_{\mathbf{O}}} = \Gamma_{\mathcal{G}'|_{\mathbf{O}}}, A$ and $\Delta_{\mathcal{G}|_{\mathbf{O}}} = \Delta_{\mathcal{G}'|_{\mathbf{O}}}$.
- if $\mathcal{G}|_{\mathbf{O}} = \mathcal{G}'|_{\mathbf{O}}(?, A)$ then $\Gamma_{\mathcal{G}|_{\mathbf{O}}} = \Gamma_{\mathcal{G}'|_{\mathbf{O}}}, A$ and $\Delta_{\mathcal{G}'|_{\mathbf{O}}}$ have the form $\Sigma, A \supset B$. We put $\Delta_{\mathcal{G}|_{\mathbf{O}}} = \Sigma, B$.
- if $\mathcal{G}|_{\mathbf{O}} = \mathcal{G}'|_{\mathbf{O}}(?, \wedge_1)$ then $\Gamma_{\mathcal{G}|_{\mathbf{O}}} = \Gamma_{\mathcal{G}'|_{\mathbf{O}}}$ and $\Delta_{\mathcal{G}'|_{\mathbf{O}}}$ have the form $\Sigma, A \wedge B$. We put $\Delta_{\mathcal{G}|_{\mathbf{O}}} = \Sigma, A$.
- if $\mathcal{G}|_{\mathbf{O}} = \mathcal{G}'|_{\mathbf{O}}(?, \wedge_2)$ then $\Gamma_{\mathcal{G}|_{\mathbf{O}}} = \Gamma_{\mathcal{G}'|_{\mathbf{O}}}$ and $\Delta_{\mathcal{G}'|_{\mathbf{O}}}$ have the form $\Sigma, A \wedge B$. We put $\Delta_{\mathcal{G}|_{\mathbf{O}}} = \Sigma, B$.
- if $\mathcal{G}|_{\mathbf{O}} = \mathcal{G}'|_{\mathbf{O}}(?, \forall[w/x])$ then $\Gamma_{\mathcal{G}|_{\mathbf{O}}} = \Gamma_{\mathcal{G}'|_{\mathbf{O}}}$ and $\Delta_{\mathcal{G}'|_{\mathbf{O}}}$ have the form $\Sigma, \forall x A$. We put $\Delta_{\mathcal{G}|_{\mathbf{O}}} = \Sigma, A[w/x]$.
- if $\mathcal{G}|_{\mathbf{O}} = \mathcal{G}'|_{\mathbf{O}}(?, \vee)$ then $\Gamma_{\mathcal{G}|_{\mathbf{O}}} = \Gamma_{\mathcal{G}'|_{\mathbf{O}}}$ and $\Delta_{\mathcal{G}'|_{\mathbf{O}}}$ have the form $\Sigma, A_1 \vee A_2$. We put $\Delta_{\mathcal{G}|_{\mathbf{O}}} = \Sigma, A_1 \vee A_2, A_i$ where A_i is the formula asserted by the move $m \in \mathcal{G}$, $\mathcal{G}$ in $\mathcal{S}$ such that $\phi(m) = (?, \vee)$.

- if $\mathcal{G}|_{\mathbf{O}} = \mathcal{G}'|_{\mathbf{O}}(?, \exists)$ then $\Gamma_{\mathcal{G}|_{\mathbf{O}}} = \Gamma_{\mathcal{G}'|_{\mathbf{O}}}$ and $\Delta_{\mathcal{G}'|_{\mathbf{O}}}$ have the form $\Sigma, \exists xA$. We put $\Delta_{\mathcal{G}|_{\mathbf{O}}} = \Sigma, \exists xA, A[t/x]$ where $A[t/x]$ is the formula asserted by the move $m \in \mathcal{G}$, $\mathcal{G}$ in $\mathcal{S}$ such that $\phi(m) = (?, \exists)$.

We now prove that given a winning strategy $\mathcal{S}$, $\Phi(\mathcal{S}|_{\mathbf{O}})$ is *almost* a derivation in SLK; all leaves of $\Phi(\mathcal{S}|_{\mathbf{O}})$ are instances of *Id* rules or $\bot^{\mathrm{L}}$ rules (proposition 3.13) and $\Phi(\mathcal{S}|_{\mathbf{O}})$ respects the variable condition of the $\forall^{\mathrm{R}}$ and $\exists^{\mathrm{L}}$ rules of SLK (proposition 3.14)

Proposition 3.12. *Let $\mathcal{S}$ be an arbitrary winning strategy and $\mathcal{G}$ be an arbitrary game in $\mathcal{S}$. If $\mathcal{G}$ ends in a* **P** *defense move that asserts a formula A then the sequent $\Gamma_{\mathcal{G}|_{\mathbf{O}}} \vdash \Delta_{\mathcal{G}|_{\mathbf{O}}}$ associated to the* **O***-restriction $\mathcal{G}|_{\mathbf{O}}$ of $\mathcal{G}$ by the function Φ is of the form $\Gamma \vdash \Delta, A$*

Proof. By induction on the length of $\mathcal{G}$ □

Proposition 3.13. *Let $\mathcal{S}$, be an arbitrary winning strategy for a formula A and $\mathcal{G}$ an arbitrary maximal branch in $\mathcal{S}$.*

1. *If $\mathcal{G} = \mathcal{G}'\mathfrak{m}^{\mathbf{O}}$ then m asserts $\bot$ and the sequent $\Gamma_{\mathcal{G}|_{\mathbf{O}}} \vdash \Delta_{\mathcal{G}|_{\mathbf{O}}}$ associated to the* **O***-restriction $\mathcal{G}|_{\mathbf{O}}$ of $\mathcal{G}$ by the function Φ is of the form $\Gamma, \bot \vdash \Delta$.*

2. *if $\mathcal{G} = \mathcal{G}'\mathfrak{m}^{\mathbf{P}}$ then $m = (!, C)$ with C atomic gentzen subformula of A and the sequent $\Gamma_{\mathcal{G}|_{\mathbf{O}}} \vdash \Delta_{\mathcal{G}|_{\mathbf{O}}}$ associated to the* **O***-restriction $\mathcal{G}|_{\mathbf{O}}$ of $\mathcal{G}$ by the function Φ is of the form $\Gamma, C \vdash C, \Delta$*

Proof. (1) is a direct consequence of definition 3.4 and of the definition of the function Φ. (2) derives from condition 3 in definition 3.3 and proposition 3.12

□

Proposition 3.14. *let $\mathcal{S}$ be an arbitrary winning strategy and $\mathcal{G}$ be a game in $\mathcal{S}$. Suppose that $\mathcal{G}$ ends in* **O***-move that is either*

1. *an attack against a universal quantifier* $(?, \forall[w/x])$

2. *or a defense against an existential attack* $(!, A[w/x])$.

Then the variable w does not appear in the sequent associated by the function Φ to the **O***-restriction $\mathcal{G}'|_{\mathbf{O}}$ of the proper prefix $\mathcal{G}'$ of $\mathcal{G}$.*

Proof. Both (1) and (2) are granted by the conditions 4 and 5 in definition 3.3 and by condition 3 in definition 3.5.

□

We are now ready to prove the main result of this section. We have just shown that we can associate a tree of sequents with each winning strategy. In addition, we have shown that the above-mentioned sequent tree is almost a proof in SLK: all its leaves are instance of *Id* rules or $\bot^{\mathrm{L}}$ rules of SLK and it respects the variable restriction on the $\forall^{\mathrm{R}}$ and $\exists^{\mathrm{L}}$ rules of SLK.

Theorem 3.2. *Let $\mathcal{S}$ an arbitrary winning strategy and let $\mathcal{S}|_{\mathbf{O}}$ be its* **O***-tree. To each sequence of* **O***-moves $\mathcal{G}|_{\mathbf{O}}$ in $\mathcal{S}|_{\mathbf{O}}$ we can associate a derivation $\mathfrak{D}_{\mathcal{G}|_{\mathbf{O}}}$ of $\Gamma_{\mathcal{G}|_{\mathbf{O}}} \vdash \Delta_{\mathcal{G}|_{\mathbf{O}}}$, where $\Gamma_{\mathcal{G}|_{\mathbf{O}}} \vdash \Delta_{\mathcal{G}|_{\mathbf{O}}}$ is the sequent associated by the function Φ to $\mathcal{G}|_{\mathbf{O}}$.*

Proof. Let $\mathcal{G}|_{\mathbf{O}}$ be an arbitrary element of $\mathcal{S}|_{\mathbf{O}}$. Suppose that the induction hypothesis holds for each suffix $\mathcal{G}'|_{\mathbf{O}}$ of $\mathcal{G}|_{\mathbf{O}}$ in $\mathcal{S}|_{\mathbf{O}}$. We consider the last move **P**-move m_{2n} of the game $\mathcal{G} \in \mathcal{S}$ such that the **O**-restriction of $\mathcal{G}$ is $\mathcal{G}|_{\mathbf{O}}$.

We only prove some of the cases that are not straightforward

1. if m_{2_n} is a defense move $(!, \exists x A)$ then there are many cases, depending on the form of A. We treat only two cases:

   - if A is atomic, then $\mathcal{G}|_{\mathbf{O}}(?, \exists)$ is maximal in $\mathcal{S}|_{\mathbf{O}}$; we associate with $\mathcal{G}|_{\mathbf{O}}$ the following derivation in which A is active.

$$\dfrac{\dfrac{}{\Gamma, A \vdash A, \exists x A, \Delta}\, Id}{\Gamma, A \vdash \exists x A}$$

   - if $A = B \vee C$, then $\mathcal{G}|_{\mathbf{O}}{}' = \mathcal{G}|_{\mathbf{O}}(?, \exists)(?, \vee) \in \mathcal{S}|_{\mathbf{O}}$ because of condition 3 in definition 3.5. This means in particular that the formula $(B \vee C)[t/x]$ is active in the derivation that we associate with $\mathcal{G}|_{\mathbf{O}}$

$$\dfrac{\begin{matrix}\vdots \mathfrak{D}_{\mathcal{G}|_{\mathbf{O}}{}'}\\ \Gamma \vdash (A \vee B)[t/x], \exists x(A \vee B), \Delta\end{matrix}}{\Gamma \vdash \exists x(A \vee B), \Delta}\, \exists^{\mathrm{R}}$$

2. if m_{2_n} is an attack $(?, A)$ on the assertion $A \supset C$, then there are many cases depending on the form of A. We again only treat two cases:

- if A is atomic, then the immediate suffix of $\mathcal{G}|_{\mathbf{O}}$ is $\mathcal{G}|_{\mathbf{O}}(!, C)$ for which the proposition hold by hypothesis. We associate it with the following derivation.

$$\dfrac{\dfrac{}{\Gamma, A \supset C, A \vdash A, \Delta}\text{Id} \qquad \begin{array}{c}\vdots \mathfrak{D}_{\mathcal{G}|_{\mathbf{O}}(!,C)} \\ \Gamma, A \supset C, C \vdash \Delta\end{array}}{\Gamma, A \supset C \vdash \Delta}\supset^{\mathsf{L}}$$

- if $A = (A_1 \supset A_2)$, then $\mathcal{G}|_{\mathbf{O}}$ has two immediate suffixes: $\mathcal{G}|_{\mathbf{O}}, (?, A_1)$ and $\mathcal{G}|_{\mathbf{O}}, (!, C)$, for which the proposition holds by hypothesis. We associate the following derivation to $\mathcal{G}|_{\mathbf{O}}$.

$$\dfrac{\dfrac{\begin{array}{c}\vdots \mathfrak{D}_{\mathcal{G}|_{\mathbf{O}},(?,A_1)} \\ \Gamma, (A_1 \supset A_2) \supset C, A_1 \vdash A_2, \Delta\end{array}}{\Gamma, (A_1 \supset A_2) \supset C \vdash A_1 \supset A_2, \Delta}\supset^{\mathsf{R}} \qquad \begin{array}{c}\vdots \mathfrak{D}_{\mathcal{G}|_{\mathbf{O}},(!,C)} \\ \Gamma, (A_1 \supset A_2) \supset C, C \vdash \Delta\end{array}}{\Gamma, (A_1 \supset A_2) \supset C \vdash \Delta}\supset^{\mathsf{L}}$$

3. If m_{2n} is an existential repetition asserting a formula $A[t/x]$, we proceed as follows: we only consider the case where $A[t/x] = (B \supset C)[t/x]$. By induction hypothesis, there is derivation of the sequent $\Gamma, B[t/x] \vdash \exists x(B \supset C), C[t/x], \Delta$ associated to the direct suffix $\mathcal{G}|_{\mathbf{O}}(?, B[t/x])$ of $\mathcal{G}|_{\mathbf{O}}$. We associate the following derivation.

$$\dfrac{\dfrac{\begin{array}{c}\vdots \mathfrak{D}_{\mathcal{G}|_{\mathbf{O}}(?,B)} \\ \Gamma, B[t/x] \vdash \exists x(B \supset C), C[t/x], \Delta\end{array}}{\Gamma \vdash \exists x(B \supset C), (B \supset C)[t/x], \Delta}\supset^{\mathsf{R}}}{\Gamma \vdash \exists x(B \supset C), \Delta}\exists^{\mathsf{R}}$$

□

3.5 From derivations to strategies

Turning a derivation $\mathfrak{D}$ of a formula F into a winning strategy $\mathcal{S}$ for F is easier. To do so, we describe a procedure, that we call $p2s$ (from a Proof in SLK to a strategy). The procedure $p2s$ explore the proofs $\mathfrak{D}$ starting from the root and proceeding by level order traversal. The order of traversal of daughters is irrelevant. The procedure associate to $\mathfrak{D}$ a prefix closed set of games for the formula F.

Theorem 3.3. *Let F be an arbitrary formula and $\mathfrak{D}$ be an arbitrary derivation of F in SLK. There is a function $p2s$ such that $p2s(\mathfrak{D})$ is a winning strategy $\mathcal{S}$ for F.*

Proof. Let x be an arbitrary node the proof $\mathfrak{D}$ of the formula F having depth n, and let $\Gamma \vdash C$ be the sequent that decorates x. Suppose that

1. the branch $r = x_o, \dots x_n = x$ of the derivation from the root r of $\mathfrak{D}$ to x is already associated with a prefix closed set S_x of games for the formula F. Each $\mathcal{G}$ in S_x in which the last move of **P** is the assertion of a complex formula or an attack move ends in a **O**-move;

2. for each formula B in Γ there is an **O**-assertion move $(\star, B)$ in some game $\mathcal{G}$ in S_x;

3. the prefix closed set $\mathcal{S}_x$ is a strategy for F.

The prefix closed set of games S_{a_1} associated with a_1 where a_1 is any daughter of x is defined as follows:

1. if a_1 is obtained by an identity rule $\Gamma, A \vdash A$, then $S_{a_1} = S_x \cup \{\mathcal{G}(!, A)\}$ where A is the active formula of the identity rule and $\mathcal{G}$ is a maximal game in S_x such that $(!, A)$ is legal for $\mathcal{G}$.

2. If a_1 is labelled with a sequent obtained from a right introduction rule with active formula A:

   (a) if A is $B \supset C$ then $S_{a_1} = S_x \cup \{\mathcal{G}(!, B \supset C)(?, B)\}$ where $\mathcal{G}$ is a maximal game in S_x such that $(!, A)$ is legal for $\mathcal{G}$;

   (b) if A is $\forall xB$, then S_{a_1} is $S_x \cup \{\mathcal{G}, (!, \forall xB), (?, \forall[w/x])\}$ where $\mathcal{G}$ is a maximal game in S_x such that $(!, A)$ is legal for $\mathcal{G}$ and the variable w in $(?, \forall[w/x])$ is the variable that appears in the premise of a_1 but not in a_1;

   (c) if A is $B \wedge C$, then $S_{a_1} = S_x \cup \{\mathcal{G}(!, B \wedge C)(?, \wedge_1)\} \cup \{\mathcal{G}(!, B \wedge C)(?, \wedge_2)\}$ where $\mathcal{G}$ is a maximal game in S_x such that the **P**-move $(!, B \wedge C)$ is legal for $\mathcal{G}$;

   (d) if A is $A_1 \vee A_2$ or $\exists xB$:

      i. if A is not active in a sequent that appears below a_1, then $S_{a_1} = S_x \cup \{\mathcal{G}, (!, A), (?, s)\}$ where $\mathcal{G}$ is a maximal game in S_x such that $(!, A)$ is legal for $\mathcal{G}$ and where $(?, s)$ is the unique attack-move which is legal for $\mathcal{G}(!, A)$;

ii. otherwise, let F be the formula that is active in the premise of a_1. If F is an implication, a universally quantified formula or a conjunction, then S_{a_1} is constructed according to 2a, 2b or 2c. Otherwise, S_{a_1} is constructed according to 2(d)i. Remark $F = B[t/x]$ or $F = A_i$ $i \in \{1, 2\}$ by the definition of SLK-proof.

3. If a_1 is labelled with a sequent obtained from a left introduction rule with active formula A:

   (a) if A is $B_1 \wedge B_2$, then $S_{a_1} = S_x \cup \{\mathcal{G}(?, \wedge_i)(!, B_i)\}$ where B_i is the direct subformula of $B_1 \wedge B_2$ that appears in the premise of a_1 but not in a_1 and $\mathcal{G}$ is a maximal game in S_x such that the **P**-move $(?, \wedge_i)$ is legal for $\mathcal{G}$;

   (b) if A is $\forall xB$, then $S_{a_1} = S_x \cup \{\mathcal{G}(?, \forall[t/x])(!, B[t/x])\}$ where $B[t/x]$ is the formula occurrence that appears in the premise of a_1 but not in a_1 and $\mathcal{G}$ is a maximal game in S_x such that the **P**-move $(?, \forall[t/x])$ is legal for $\mathcal{G}$;

   (c) if A is $B \vee C$, then $S_{a_1} = S_x \cup \{\mathcal{G}(?, \vee)(!, B)\} \cup \{\mathcal{G}(?, \vee)(!, C)\}$ where $\mathcal{G}$ is a maximal game in S_x such that the **P**-move $(?, \vee)$ is legal for $\mathcal{G}$;

   (d) if A is $B \supset C$, then $S_{a_1} = S_x \cup \{\mathcal{G}, (?, B), (?, q_1)\} \cup \ldots \cup \{\mathcal{G}, (?, B), (?, q_n)\}$ $\cup\{(\mathcal{G}, (?, B), (!, C)\}$. Where $\mathcal{G}$ is a maximal game in S_x such that the **P**-move $(?, B)$ is legal for $\mathcal{G}$ each q_i is a question on B;

   (e) if A is $\exists xB$, $S_{a_1} = S_x \cup \{\mathcal{G}(?, \exists)(!, B[w/x])\}$ where $\mathcal{G}$ is a maximal game in S_x such that the **P**-move $(?, \exists)$ is legal for $\mathcal{G}$ and $B[w/x]$ is a formula that appears in the premise of a_1 but not in a_1;

   (f) if A is $\perp$ then $S_{a_1} = S_x$. Remark that by 1 and 2 this clause is well formulated.

It is easy to check that conditions 1,2 and 3 are respected after the application of the procedure. Remark that the procedure jumps from a node v of the proof-tree obtained by a $\supset^{\mathsf{L}}$-rule to the daughter of the daughter of v. The same phenomena occur whenever a right-hand formula is obtained by an $\vee^{\mathsf{R}}$-rule or $\exists^{\mathsf{R}}$-rule as described in 2(d)ii. For each node v in a proof tree, there are finitely many ancestors of v, as a consequence the procedure always ends.

□

3.6 Intuitionistic dialogical games

Games of definition 3.3 are games for first-order *classical* logic. The following is a definition of games for first-order intuitionistic logic. Let (ρ, ϕ) be an augmented sequence and ρ_i an attack-move in ρ we say that ρ_i is answered iff there is no defense-move ρ_j such that $\phi(\rho_j) = \rho_i$

Definition 3.7 (Intuitionistic Games)**.** *An intuitionistic game for a formula A is an augmented sequence (ρ, ϕ) satisfying conditions 1 to 5 in the definition 3.3 of game and in which condition 6 is replaced by the following*

Well-bracketing *If ρ_k is a defense move and k is even then $\sigma(\rho_k) = \rho_j$ is the unanswered attack move having the greatest odd index in the prefix of ρ ending with ρ_{k-1}*

The definition of move m legal for a game $\mathcal{G}$ as well as the definition of game won by **P** remains unchanged. Proposition 3.1,3.2,3.3 and 3.4 holds for intuitionistic games. The following proposition is a direct consequence of the above definition.

Proposition 3.15. *Let $\mathcal{G}$ be an intuitionistic game. For each* **O***-attack-move ρ_j in $\mathcal{G}$ there is at most one defense-move ρ_i such that $\phi(\rho_i) = \rho_j$.*

3.6.1 Some examples of intuitionistic games

We give some examples of games for intuitionistic logic. Let a be a propositional variable, and let P and Q be two unary predicate variables

$$\begin{array}{rcl|l} \mathfrak{m}_0 &=& (!, a \vee \neg a) & \\ \mathfrak{m}_1 &=& (?, \vee) & \mathfrak{m}_0 \\ \mathfrak{m}_2 &=& (!, \neg a) & \mathfrak{m}_1 \\ \mathfrak{m}_3 &=& (?, a) & \mathfrak{m}_2 \end{array}$$

$$\begin{array}{rcl|l} \mathfrak{m}_0 &=& (!, \neg\neg a \supset a) & \\ \mathfrak{m}_1 &=& (?, \neg\neg a) & \mathfrak{m}_0 \\ \mathfrak{m}_2 &=& (?, \neg a) & \mathfrak{m}_1 \\ \mathfrak{m}_3 &=& (?, a) & \mathfrak{m}_2 \end{array} \qquad \begin{array}{rcl|l} \mathfrak{m}_0 &=& (!, \neg\neg a \supset a) & \\ \mathfrak{m}_1 &=& (?, \neg\neg a) & \mathfrak{m}_0 \\ \mathfrak{m}_2 &=& (?, \neg a) & \mathfrak{m}_1 \\ \mathfrak{m}_3 &=& (!, \bot) & \mathfrak{m}_2 \end{array}$$

$\mathfrak{m}_0 = (!, \forall x(P(x) \wedge Q(x)) \supset \forall xP(x) \wedge \forall xQ(x))$	
$\mathfrak{m}_1 = (?, \forall x(P(x) \wedge Q(x)))$	$\mathfrak{m}_0$
$\mathfrak{m}_2 = (!, \forall xP(x) \wedge \forall xQ(x))$	$\mathfrak{m}_1$
$\mathfrak{m}_3 = (?, \wedge_1)$	$\mathfrak{m}_2$
$\mathfrak{m}_4 = (!, \forall xP(x))$	$\mathfrak{m}_3$
$\mathfrak{m}_5 = (?, \forall[w/x])$	$\mathfrak{m}_4$
$\mathfrak{m}_6 = (?, \forall[w/x])$	$\mathfrak{m}_1$
$\mathfrak{m}_7 = (!, P(w) \wedge Q(w))$	$\mathfrak{m}_6$
$\mathfrak{m}_8 = (?, \wedge_1)$	$\mathfrak{m}_7$
$\mathfrak{m}_9 = (!, P(w))$	$\mathfrak{m}_8$
$\mathfrak{m}_{10} = (!, P(w))$	$\mathfrak{m}_5$

$\mathfrak{m}_0 = (!, \neg\forall x\neg P(x) \supset \exists xP(x))$	
$\mathfrak{m}_1 = (?, \neg\forall x\neg P(x)$	$\mathfrak{m}_0$
$\mathfrak{m}_2 = (?, \forall x\neg P(x)$	$\mathfrak{m}_1$
$\mathfrak{m}_3 = (?, [w/x])$	$\mathfrak{m}_2$
$\mathfrak{m}_4 = (!, \neg P(w))$	$\mathfrak{m}_3$
$\mathfrak{m}_5 = (?, P(w))$	$\mathfrak{m}_4$

$\mathfrak{m}_0 = (!, \neg\forall x\neg P(x) \supset \exists xP(x))$	
$\mathfrak{m}_1 = (?, \neg\forall x\neg P(x)$	$\mathfrak{m}_0$
$\mathfrak{m}_2 = (?, \forall x\neg P(x)$	$\mathfrak{m}_1$
$\mathfrak{m}_3 = (!, \bot)$	$\mathfrak{m}_2$

Remark 2. *Games for the formulas $\neg\neg a \supset a$, $a \vee \neg a$ and $\neg\forall x\neg P(x) \supset \exists xP(x)$ were already presented in subsection 3.2.4. The difference here is that* **P** *cannot win a game over this formula no matter how* **O** *chooses to act. Consider for example the game for the formula $a \vee \neg a$; this game is won by* **O** *as it ends in a* **O***-move that is not an assertion of $\bot$. The well-bracketing condition prevents* **P** *to make the move $(!, a)$ as a defense against the move $(?, \vee)$, this means that* **P** *cannot win a game on this formula. Similar phenomena happen in the two left-hand game for the formulas $\neg\neg a \supset a$ and $\neg\forall x\neg P(x) \supset \exists xP(x)$. The proponent cannot assert a (resp. $\exists xP(x)$) because there is an answered question having greater index than the one of $(?, \neg\neg a)$ (resp. $(?, \neg\forall x\neg P(x))$)*

3.6.2 Intuitionistic Strategies

The definition of strategies for Intuitionistic games is the same as the one for classical games i.e., definition 3.5. Of course, an intuitionistic strategy will be a tree of *intuitionistic games*. To prove that a formula is intuitionistic valid if and only if there is a winning intuitionistic strategy for the formula, we proceed as in the classical case. We transform a winning strategy $\mathcal{S}$ for a formula into a derivation $\mathfrak{D}$ for the formula and vice versa. Of course, the derivations we consider are constructed in a cut-free complete sequent calculus for intuitionistic logic. The sequent calculus we are going to consider is the intuitionistic variant of SLK, and we are going to call it SLJ (strategic LJ).

Definition 3.8. *The sequent calculus SLJ is defined by the rules in Table 3.2. A sequent is an expression $\Gamma \vdash C$ where Γ is a finite (possibly empty) multiset of formulas and C is a formula. Greek upper-case letters $\Gamma, \Delta, \Sigma, \Pi \dots$ stand for multisets of formulas. In the Id-rule A is of the form $P(t_1, \dots, t_n)$ where P is a predicate variable with arity $n \geq 0$ and the t_i are terms, moreover $\bot$ is not an element of Γ. In the $\forall^{\mathsf{R}}$ and $\exists^{\mathsf{L}}$ rules, the variable y does not occur in the conclusion sequent. In the $\supset^{\mathsf{L}}$ rule the left-side premise of the rule is obtained by an Id-rule or a right introduction rule. In the $\exists^{\mathsf{R}}$-rule as well as in the $\vee^{\mathsf{R}}$-rule, the premise of the rule is obtained by a textitId-rule or a right introduction rule. The bold formulas are called active formulas.*

A derivation (or a proof) $\mathfrak{D}$ of a sequent $\Gamma \vdash C$ in SLJ is a tree of sequents constructed according to the rules of SLJ in which leaves are instances of Id-rules or $\bot^{\mathsf{L}}$-rules, all sequents of the form $\Sigma, \bot \vdash \pi$ are leaves and whose root, also called conclusion, is $\Gamma \vdash C$. The height of a derivation $\mathfrak{D}$ is the number of nodes in its maximal branch minus 1.

A sequent $\Gamma \vdash C$ is said to be derivable *or* provable *in the sequent calculus LJS whenever there exists a proof with conclusion $\Gamma \vdash \Delta$.*

Remark 3. *SLJ is simply SLK in which the right-hand multiset of formulas of a sequent contains* exactly *one formula. Usually, sequent calculus system for intuitionistic logic have sequents of the form $\Gamma \vdash \Delta$ where Δ contains at most one formula. However, by adopting the $\bot^{\mathsf{L}}$-rule we can consider sequent of the form $\Gamma \vdash C$ (see [132] pp. 72-73 or [68] pp. 13-14 for a discussion)*

Propositions 3.6, 3.8, 3.9, 3.10 and 3.11 presented in section 3.3 also holds for the sequent calculus SLJ If enunciated with the appropriate differences dictated by the particularities of SLJ i.e., The $\wedge^{\mathsf{R}}$, $\vee^{\mathsf{L}}$, $\supset^{\mathsf{R}}$-rules and the $\forall^{\mathsf{R}}$-rule of SLJ are height-preserving reversible, the two rules

$$\frac{\Gamma \vdash C}{\Gamma, A \vdash C}\,\mathsf{W} \qquad \frac{\Gamma, A, A \vdash C}{\Gamma, A \vdash C}\,\mathsf{C}$$

Table 3.2: The SLJ sequent calculus.

$$\frac{}{\Gamma, \boldsymbol{\bot} \vdash C}\ \bot^{\mathsf{L}} \qquad \frac{}{\Gamma, A \vdash \mathbf{A}}\ Id$$

$$\frac{\Gamma, A \supset B \vdash A, C \qquad \Gamma, A \supset B, B \vdash C}{\Gamma, \mathbf{A} \supset \mathbf{B} \vdash C}\ \supset^{\mathsf{L}} \qquad \frac{\Gamma, A \vdash B, \Delta}{\Gamma \vdash \mathbf{A} \supset \mathbf{B}}\ \supset^{\mathsf{R}}$$

$$\frac{\Gamma, A, A \wedge B \vdash \Delta}{\Gamma, \mathbf{A} \wedge \mathbf{B} \vdash C}\ \wedge^{\mathsf{L}}_1 \qquad \frac{\Gamma, B, A \wedge B \vdash \Delta}{\Gamma, \mathbf{A} \wedge \mathbf{B} \vdash C}\ \wedge^{\mathsf{L}}_2 \qquad \frac{\Gamma \vdash A \qquad \Gamma \vdash B}{\Gamma \vdash \mathbf{A} \wedge \mathbf{B}}\ \wedge^{\mathsf{R}}$$

$$\frac{\Gamma, A \vee B, A \vdash C \qquad \Gamma, A \vee B, B \vdash C}{\Gamma, \mathbf{A} \vee \mathbf{B} \vdash C}\ \vee^{\mathsf{L}}$$

$$\frac{\Gamma \vdash A}{\Gamma \vdash \mathbf{A} \vee \mathbf{B}}\ \vee^{\mathsf{R}}_1 \qquad \frac{\Gamma \vdash B}{\Gamma \vdash \mathbf{A} \vee \mathbf{B}}\ \vee^{\mathsf{R}}_2$$

$$\frac{\Gamma, A[y/x], \exists x A \vdash C}{\Gamma, \exists \mathbf{x} \mathbf{A} \vdash C}\ \exists^{\mathsf{L}} \qquad \frac{\Gamma \vdash A[t/x]}{\Gamma \vdash \exists \mathbf{x} \mathbf{A}}\ \exists^{\mathsf{R}}$$

$$\frac{\Gamma, A[t/x], \forall x A \vdash C}{\Gamma, \forall \mathbf{x} \mathbf{A} \vdash C}\ \forall^{\mathsf{L}} \qquad \frac{\Gamma \vdash A[y/x]}{\Gamma \vdash \forall \mathbf{x} \mathbf{A}}\ \forall^{\mathsf{R}}$$

are height preserving admissible, the sequents $\Gamma, A \vdash A$, $\Gamma, A[x/y] \vdash \exists xA$, $\Gamma, A, A \supset B \vdash B$, $\Gamma, A \vdash A \vee B$ and $\Gamma, B \vdash A \vee B$ are provable for all multisets of formulas Γ and formulas A and B. Finally, the cut rule

$$\frac{\Gamma \vdash A \qquad \Gamma, A \vdash C}{\Gamma \vdash C}\,\mathtt{cut}$$

is admissible in SLJ. All the proofs of these facts follows the same pattern that in the classic case. Thanks to these facts, one can prove the following proposition, which implies that SLJ is sound and complete for first-order inutuitionistic logic

Proposition 3.16. *There is a derivation of* $\Gamma \vdash C$ *in G3i if and only if there is a derivation of* $\Gamma \vdash C$ *in SLJ for all multiset of formulas* Γ *and formula* A*.*

Proof. The sequent calculus G3i [132] is nothing but SLJ without the restrictions on the use of the rules $\vee^{\mathrm{R}}, \exists^{\mathrm{R}}$ and $\supset^{\mathrm{L}}$ thus one direction of the proof is for free. For the other direction one proceeds, as in the classical case, using the admissibility of the cut rule and proposition 3.10 □

To transform a winning strategy $\mathcal{S}$ for a formula A into a derivation $\mathfrak{D}$ of A, we follow the same pattern of the classical case. We first recursively associate a sequent to all **O**-move in the strategy. This is done by considering the **O**-tree $\mathcal{S}|_{\mathbf{O}}$ of $\mathcal{S}$ and by defining a function Φ from $\mathcal{S}|_{\mathbf{O}}$ to a tree of sequent τ. The function Φ associate a sequent $\Gamma_{\mathcal{G}|_{\mathbf{O}}} \vdash C_{\mathcal{G}|_{\mathbf{O}}}$ to each $\mathcal{G}|_{\mathbf{O}}$ in $\mathcal{S}|_{\mathbf{O}}$. Let us denote the empty sequence by ϵ

- if $\mathcal{G}|_{\mathbf{O}} = \epsilon$ then $\Gamma_{\mathcal{G}|_{\mathbf{O}}} = \emptyset$ and $C_{\mathcal{G}|_{\mathbf{O}}} = F$;
- if $\mathcal{G}|_{\mathbf{O}} = \mathcal{G}'|_{\mathbf{O}}(!, A)$ then $\Gamma_{\mathcal{G}|_{\mathbf{O}}} = \Gamma_{\mathcal{G}'|_{\mathbf{O}}}, A$ and $C_{\mathcal{G}|_{\mathbf{O}}} = C_{\mathcal{G}'|_{\mathbf{O}}}$;
- if $\mathcal{G}|_{\mathbf{O}} = \mathcal{G}'|_{\mathbf{O}}(?, A)$ then $\Gamma_{\mathcal{G}|_{\mathbf{O}}} = \Gamma_{\mathcal{G}'|_{\mathbf{O}}}, A$ and $C_{\mathcal{G}'|_{\mathbf{O}}}$ have the form $A \supset B$ we put $C_{\mathcal{G}|_{\mathbf{O}}} = B$;
- if $\mathcal{G}|_{\mathbf{O}} = \mathcal{G}'|_{\mathbf{O}}(?, \wedge_1)$ then $\Gamma_{\mathcal{G}|_{\mathbf{O}}} = \Gamma_{\mathcal{G}'|_{\mathbf{O}}}$ and $C_{\mathcal{G}'|_{\mathbf{O}}}$ have the form $A \wedge B$ we put $C_{\mathcal{G}|_{\mathbf{O}}} = A$;
- if $\mathcal{G}|_{\mathbf{O}} = \mathcal{G}'|_{\mathbf{O}}(?, \wedge_2)$ then $\Gamma_{\mathcal{G}|_{\mathbf{O}}} = \Gamma_{\mathcal{G}'|_{\mathbf{O}}}$ and $C_{\mathcal{G}'|_{\mathbf{O}}}$ have the form $A \wedge B$ we put $C_{\mathcal{G}|_{\mathbf{O}}} = B$
- if $\mathcal{G}|_{\mathbf{O}} = \mathcal{G}'|_{\mathbf{O}}(?, \forall[w/x])$ then $\Gamma_{\mathcal{G}|_{\mathbf{O}}} = \Gamma_{\mathcal{G}'|_{\mathbf{O}}}$ and $C_{\mathcal{G}'|_{\mathbf{O}}}$ have the form $\forall xA$ we put $C_{\mathcal{G}|_{\mathbf{O}}} = A[w/x]$;

- if $\mathcal{G}|_{\mathbf{O}} = \mathcal{G}'|_{\mathbf{O}}(?, \vee)$ then $\Gamma_{\mathcal{G}|_{\mathbf{O}}} = \Gamma_{\mathcal{G}'|_{\mathbf{O}}}$ and $C_{\mathcal{G}'|_{\mathbf{O}}}$ have the form $A_1 \vee A_2$ we put $C_{\mathcal{G}|_{\mathbf{O}}} = A_i$ where A_i is the formula asserted by the last move m of $\mathcal{G}$;
- if $\mathcal{G}|_{\mathbf{O}} = \mathcal{G}'|_{\mathbf{O}}(?, \exists)$ then $\Gamma_{\mathcal{G}|_{\mathbf{O}}} = \Gamma_{\mathcal{G}'|_{\mathbf{O}}}$ and $C_{\mathcal{G}'|_{\mathbf{O}}}$ have the form $\exists x A$ we put $C_{\mathcal{G}|_{\mathbf{O}}} = A[t/x]$ where $A[t/x]$ is the formula asserted by the last move m of $\mathcal{G}$.

We then prove the analogous of propositions 3.12, 3.13 and 3.14. These proposition guarantees that given a winning intuitionistic strategy $\mathcal{S}$, $\Phi(\mathcal{S})$ is a tree of sequents in which leaves are of the form $\Gamma, A \vdash A$ with A atomic or of the form $\Gamma, \bot \vdash C$, and in which the variable restriction on the use of $\forall^{\mathrm{R}}$ and $\exists^{\mathrm{L}}$ is respected. These clarifications been made we can state the following theorem the proof of which is entirely similar to the one of theorem 3.2 and will be omitted

Theorem 3.4. *Let $\mathcal{S}$ an arbitrary winning intuitionistic strategy and let $\mathcal{S}|_{\mathbf{O}}$ by its $\mathbf{O}$-tree. To each sequence of $\mathbf{O}$-moves $\mathcal{G}|_{\mathbf{O}}$ in $\mathcal{S}|_{\mathbf{O}}$ we can associate a derivation $\pi_{\mathcal{G}|_{\mathbf{O}}}$ of $\Gamma_{\mathcal{G}|_{\mathbf{O}}} \vdash C_{\mathcal{G}|_{\mathbf{O}}}$, where $\Gamma_{\mathcal{G}|_{\mathbf{O}}} \vdash C_{\mathcal{G}|_{\mathbf{O}}}$ is the sequent associated by the function Φ to $\mathcal{G}|_{\mathbf{O}}$*

To obtain a winning strategy $\mathcal{S}$ for a formula F from a proof $\mathfrak{D}$ of F in SLJ we use the same procedure p2s that has been used in the proof of theorem 3.3

Theorem 3.5. *Let F be an arbitrary formula and $\mathfrak{D}$ be an arbitrary derivation of F in SLJ. There is a function p2s such that p2s($\mathfrak{D}$) is a winning strategy $\mathcal{S}$ for F.*

Proof. The function that associates a strategy for F to a derivation of F in SLJ is the same of the function described in theorem 3.3. We only check that given a node x in π the set of games S_x associated to the branch $F = x_0, x_1 \ldots, x_{n-1}, x_n = x$ contains only games that respect the well-bracketing condition. Let $\mathcal{G} \in S_x$. We can suppose without loss of generality that x_n is decorated by a sequent in which the active formula is on the right. Suppose that $x_n = \Gamma \vdash \mathbf{B} \wedge \mathbf{C}$ then $S_x = S_{x_{n-1}} \cup \{\mathcal{G}(!, B \wedge C)(?, \wedge_1)\} \cup \{\mathcal{G}(!, B \wedge C)(?, \wedge_2)\}$ with $\mathcal{G} \in S_{x_{n-1}}$. The fact that $\mathbf{P}$-move $(!, B \wedge C)$ is justified derives from the subformula property of SLJ and from the definition of the procedure p2s. The well bracketing condition derives from the fact that each SLJ sequent has exactly one formula to the right of the $\vdash$ symbol. □

3.7 Conclusion

We proved that there is a natural correspondence between formal E-strategies for both classical and intuitionistic logic and derivations of two complete cut-free sequent calculus for first-order classical and intuitionistic logic. We hope that the simplicity of our approach will help other researchers to appreciate more the dialogical logic approach.

As we have already said our approach builds on the one developed by Herbelin in his PhD thesis [68] and can be seen as an extension of his result to first-order logic. Being more precise Herbelin shows the correspondence between winning strategies for games respecting the well-bracketing condition (definition 3.7 in this chapter) and derivations in the sequent calculus LJQ (chapter 5, section 2 of his PhD dissertation), and winning strategies for classical games (definition 3.3 in this chapter) and derivation in LKQ (chapter 5, section 3 of his PhD dissertation). As Herbelin states LJQ is a sound and complete sequent calculus for propositional minimal logic [79] i.e., the proper fragment propositional intuitionistic logic where formulas constructed using only the connective $\wedge, \vee$ and $\supset$, the sequents of LJQ have the form $\Gamma \vdash \Delta$ where Δ contains exactly one formula. LKQ is an extension of LJQ in which sequents $\Gamma \vdash \Delta$ are allowed to contain multiple formulas on the right of the $\vdash$ symbol. It should be remarked that is difficult to tell for which logic the sequent system LKQ is sound and complete for: one can prove the Peirce's law i.e., the formula $((X \supset Y) \supset X)) \supset X$, but it is impossible to prove that negation is involutive i.e., $\neg\neg X \supset X$ and that contradiction is explosive i.e., $\bot \supset A$. This is because the system LKQ lacks rules for both the propositional constant $\bot$ and for negation. The sequent calculus SLJ and SLK that we have presented can be regarded as first-order extension of, respectively, LJQ and LKQ, however contrarily to LJQ the propositional fragment of our calculus SLJ is complete for intuitionistic propositional logic, and contrarily to LKQ our calculus SLK is complete for classical logic. Moreover, the two systems we have developed obeys a *focusing principle* [8]: whenever we apply (bottom-up) an $\exists R$ or a $\vee R$ rule over a sequent $\Gamma \vdash \Delta$ we are obliged to apply right-rules until an implication a conjunction or a universally quantified formula is active. It is quite surprising that formal E-strategies naturally corresponds to this type of calculus. The fact that Felscher himself did not notice such correspondence is easily explained by remarking when his paper was published: focusing was not known at the time.

We would like to conclude with some very general remarks on dialogical logic. We have shown that if a formula is classically (resp. intuitionistically) valid then there is a winning strategy (resp. a winning *intuitionistic* strategy) for the formula and that, vice versa, if there is a winning strategy (resp. winning intuitionistic strategy) for a formula then the formula is classically (resp. intuitionistically) valid. This has been proved by showing that

1. There is a function that maps proofs in the sequent calculus to winning strategies

2. *every* winning strategy $\mathcal{S}$ is the image of some proof $\mathfrak{D}$, i.e., the function that goes from proofs to winning strategies is surjective.

If we conceive, as Lorenzen did, dialogical logic as a *semantic* for first-order logic then (1) is a form of soundness : every formula that is provable in first-order logic is dialogically valid. (2) is a form of completeness : every formula that is dialogically valid is provable in first-order logic. By looking at (2) more attentively we realize that it is rather strong form of completeness. Consider the function

$$\psi : \quad A \quad \to \quad \{\mathcal{S} \mid \mathcal{S} \text{ is a winning strategy for } A\}$$

.

that maps a formula A to the set of winning strategies for A. The completeness theorem implies that if A is provable, then $\psi(A) \neq \emptyset$. The form of completeness we have proved says that every element of $\psi(A)$ is the image of a proof $\mathfrak{D}$ of A. This form of completeness is known under the name of *full completeness*. In light of this latter discussion, we think that the reader would be puzzled and surprised by the following selection of quotes

> In logic, the earliest forerunners [of the game-approach to logic] are Lorenzen and his coworkers. They considered the activity of proving a formula as a strategy in what they called a dialogue game between two players: Proponent, who is responsible for building the proof, and Opponent, who chooses to refute the formula or any of the intermediate conclusions in the (unraveling of the) proof. Unfortunately, most of this work focused on provability more than on proofs themselves [31].

> A game semantic approach to proof (or at least provability) was suggested by Lorenzen and his school. His ideas have been developed and made precise by a number of people, and form the basis for a distinctive tradition in philosophical logic [75].

> Lorenzen is presumably the only logician of those times to have fully assumed the dialectic dimension of logic. His School must have been in a cantilevered position w.r.t. the milieu: [...] No conscience of a layer below 1 [provability]; thus Felscher formulated his theorem under the form *if there is a winning strategy, then A is provable*, where one would expect: *if there is a winning strategy, then it comes from a proof* [62].

The reader could object that Curien, Hyland and Girard are not aware of our work. Nonetheless, it should be remarked that Felscher was completely aware of the — so to say — correspondence between proofs and strategies

> It is the purpose of this article to prove an equivalence theorem, saying that every strategy for (certain types of) dialogues can, by a well-defined algorithm, be transformed into a proof in Gentzen's calculus LJ and vice versa [46].

Felscher proof is not an example of mathematical elegance: as we have said many times, it is notoriously known to be lengthy and difficult to understand. But by reading Felscher's words — that one can find in the first page of Felscher's paper— it is rather hard to believe that the German mathematician was only concerned by mere provability as Curien, Hyland and Girard seems to suggest. Faced to this fact, one could conclude that the three authors have simply a wrong view of dialogical logic. However, we do think that in *in some sense* the dialogical logic analysis of proof miss a crucial aspect, the fact that proof can be *composed*: given a proof of $A \supset B$ and a proof of $B \supset C$ we can compose the two proofs obtaining a proof of $A \supset C$. This is the content of the cut rule

$$\dfrac{\begin{array}{c}\vdots\\ A \vdash B\end{array} \qquad \begin{array}{c}\vdots\\ B \vdash C\end{array}}{A \vdash C}\;\text{cut}$$

It is of course possible to show that the cut rule is admissible in our variant of dialogical logic

Proposition 3.17. *if there is a winning strategy for $A \supset B$ and a winning strategy for $B \supset C$, then there is a winning strategy for $A \supset C$.*

Proof. By theorem 3.2 there are SLK derivation $\mathfrak{D}_1$ of $\vdash A \supset B$ and $\mathfrak{D}_2$ of $\vdash B \supset C$, by proposition 3.6 the sequents $A \vdash B$ and $B \vdash C$ are provable. By admissibility of the cut-rule (proposition 3.11) the sequent $A \vdash C$ is provable and thus there is a proof of $\vdash A \supset C$. By theorem 3.2 there is a winning strategy for $A \supset C$ □

But we think that this way of dealing with the cut rule is not satisfactory. We have not shown how to *construct* a strategy for $A \supset C$ from strategies for $A \supset B$ and $B \supset C$ but merely asserted its existence using cut-admissibility in the sequent calculus SLK. It would be much more interesting to define an analogous of the cut-rule directly on strategies. We think that this could be obtained by relaxing the definition of game in order to let the proponent assert, at any point of the game, an arbitrary formula C. After the Proponent assertion of C, the Opponent can continue the game by either attacking C or by asserting C in turn. The cut-admissibility theorem for strategies would be obtained by proving that the set of formulas admitting winning strategies containing this kind of games is equal to the class of formulas admitting 'regular' winning strategies. However, we will not pursue by this path in our thesis. In the next chapter we will present a *semantic of proofs* that is heavily inspired by dialogical logic and in which a satisfying treatment of composition of strategies has already been developed i.e., *game semantics*. In particular, we will define a game semantics that is fully complete for a constructive variant of the basic modal logic K.

Chapter 4

Game Semantic for Constructive Modal Logic

Abstract
In this chapter we provide a game semantics for the constructive modal logic CK. We first study a complete sequent calculus for the modal logic CK and we prove the cut-elimination theorem for this calculus. We then define arenas encoding modal formulas, winning innocent strategies for games on these arenas and we prove their compositionality. Finally, we characterize the winning strategies corresponding to proofs in the logic CK. To prove the full-completeness of our semantics, we provide a sequentialization procedure of winning strategies. All the results of this chapter, but the cut-elimination theorem for the sequent calculus, already appears in [4].

4.1 Introduction

In this chapter we present a denotational semantics for the constructive modal logic CK. The denotational semantics we are going to present is constructed using techniques from *game semantics*.

4.1.1 Generalities about denotational semantics

Semantics is the area of logic concerned with specifying the meaning of the logical constructs. We distinguish between two main kinds of the semantic approach to logic. The first, the model-theoretic approach, is concerned with specifying the meaning of formulas in terms of truth in some model. The second, the semantic of proofs approach, is concerned with specifying the meaning of proofs of the logic: we associate an appropriate mathematical object, such as a number, a tuple, a function, or a graph with each proof of the logic being considered:

$$\begin{array}{rccc} \{\!\{-\}\!\}: & \{\text{ Proofs }\} & \to & \{\text{ Mathematical Objects }\} \\ \\ & \mathfrak{D} & \to & \{\!\{\mathfrak{D}\}\!\} \end{array}$$

the map $\{\!\{-\}\!\}$ from proofs to mathematical objects should respect some minimal requirements:

- the map $\{\!\{-\}\!\}$ is not the identity function;
- the map $\{\!\{-\}\!\}$ is not degenerate: one can find a formula A and two proof $\mathfrak{D}$ and $\mathfrak{D}'$ of A such that $\{\!\{\mathfrak{D}\}\!\}$ is not equal to $\{\!\{\mathfrak{D}'\}\!\}$;
- the map $\{\!\{-\}\!\}$ is congruent : if $\mathfrak{D}$ and $\mathfrak{D}'$ are obtained by an application of the same rule R on $\mathfrak{D}_1$ and $\mathfrak{D}_2$ and $\{\!\{\mathfrak{D}_1\}\!\} = \{\!\{\mathfrak{D}_2\}\!\}$ then $\{\!\{\mathfrak{D}\}\!\} = \{\!\{\mathfrak{D}'\}\!\}$;

A semantic of proofs is said to be *fully complete* whenever the map $\{\!\{-\}\!\}$ is surjective, that is: every object in the image of $\{\!\{-\}\!\}$ is the interpretation of some proof. A semantic of proofs for a logic is called *denotational semantic* whenever the map $\{\!\{-\}\!\}$ respects another important requirement: if there is a notion of transformation, or reduction, between proofs —usually cut or detour-elimination— then proofs that are equal modulo this notion of reduction are interpreted by the same mathematical object.

$$\text{if } \mathfrak{D} \text{ reduces to } \mathfrak{D}' \text{ then } \{\!\{\mathfrak{D}\}\!\} = \{\!\{\mathfrak{D}'\}\!\}$$

Dialogical logic, which we have presented in the previous chapter, is clearly a fully complete semantic of proofs for classical and intuitionistic logic. However, *it is not* a denotational semantics. There is no obvious way to interpret an application of the cut-rule of the sequent calculus, and thus we cannot fulfil the last requirement we have presented. In this chapter we present a fully complete denotational semantics for the constructive variant of the basic modal logic K.

4.1.2 Generalities about constructive modal logics

Modal logics are extensions of classical logic, making use of *modalities* to qualify the truth of a judgement. According to the interpretation of such modalities, modal logics find applications, for example, in knowledge representation [134], artificial intelligence [103] and formal verification [73]. More precisely, modal logics are obtained by extending classical logic with a modality operator $\Box$ (together with its dual operator $\Diamond$), which are usually interpreted as *necessity* (respectively *possibility*).

When we move from the classical to the intuitionistic setting, the modality $\Diamond$ is no longer the dual of the modality $\Box$ and by consequence the classical k-axiom $\Box(A \supset B) \supset (\Box A \supset \Box B)$ is no longer sufficient to express the behavior of the modality $\Diamond$. Depending on the chosen axioms, it is possible to define different flavors of "intuitionistic modal logics" (see, e.g., [49, 117, 114, 128, 12, 36]). In this chapter we consider the basic flavor of intuitionistic modal logic originated in the classic work of Dag Prawitz [117] and now called *constructive modal logics* [12, 66, 101, 45, 85]. In particular, we will be interested in the constructive modal logic K also known as CK. This logic is defined as the set of formulas containing all theorems of minimal logic, each instance of the axioms $\Box(A \supset B) \supset (\Box A \supset \Box B)$ and $\Box(A \supset B) \supset (\Diamond A \supset \Diamond B)$ and closed for the rule of modus ponens and necessitation. The interest of these logics lies in the fact that it is possible to give them a computational interpretation by extending the simple typed lambda calculus with specific constructors for the modalities. The modal logic CK is the basic system of constructive modal logic, in the sense that all other systems are obtained by adding one or more modal axioms to it.

The model theoretic semantics for CK and its extensions is well studied and understood, and it is given, in the usual semantics for modal logic, by a particular class of kripke frames i.e., a set of 'worlds' with specific relations between them. We will talk no more about model theoretic semantics and invite the interested reader to consult [85].

On the contrary, the study of denotational semantics for CK is still rough and the only full complete denotational model for CK is defined by the quotient of its natural deduction proofs with respect to detour elimination [11, 12]. Even if the model presented by Bellin and colleagues is a fully complete denotational model, we cannot be really satisfied with it. As we have said, proofs are interpreted by classes of equivalence on proofs. Each of these class of equivalence contains proofs that are equal modulo the reflexive, symmetric and transitive closure of the relation "the proof $\mathfrak{D}$ reduces in one step of detour elimination to the proof $\mathfrak{D}'$" i.e., the model studied in [11] is the syntactic category obtained by the quotient of proofs modulo detour. It should be clear that a syntactic model of this kind is in some sense not *concrete*: the interpretation of proofs is *almost* the identity function on the set of proofs.

4.1.3 Generalities about game semantics

Modern game semantics has been invented (or discovered) independently and almost simultaneously by Samson Abramsky and Radha Jagadeesan [1] and by Martin Hyland and Luke Ong [76]. The two approaches slightly differ, and in this thesis we will focus on the Hyland-Ong approach. Game semantics has been used to provide denotational models for both programming languages and logical systems. On the logical side originally game semantics were developed as a denotational semantic for the multiplicative fragment of linear logic [1, 76]. On the programming language side game semantics were developed as denotational models for the language PCF [2, 77]. In game semantic, as in dialogical logic, proofs are interpreted as *winning strategies* for a two-player game. As in dialogical logic, the games are sequences of moves made alternately by the two players, which — as in dialogical logic— are called Proponent (**P**) and Opponent (**O**). However, contrarily to dialogical logic, the games are played over an *arena*. Arenas are directed graphs that represent formulas. A game for a formula A will be a sequence of vertices of the arena representing A. At each point of the game, the player who is about to make the move must choose a vertex of the arena that points to one of the vertices that has been already chosen by the other player. Using again the analogy with chess, an arena represent the chessboard and the game the configuration of the chessboard at a given moment. What a player can do at a given moment depends upon the current configuration of the game, the physical structure of the chessboard and — of course— the rules of chess.

In particular, we will use the techniques of game semantics to construct a concrete denotational model for the logic CK. In a concrete, fully complete model, the connection between syntax and semantics is strong without, however, being achieved through a by product of syntax. This point is well explained by Laurent in [92]

> An important topic in the recent developments of denotational semantics has been the quest for stronger and stronger connections between the syntactical systems and the denotational models. Works for making the two notions closer have become from the two sides and can be seen as an attempt to solve the general question "what is a proof?". Full abstraction and full completeness

results have been initiated with game semantics [...] This full completeness property can be considered as a measurement of the precision of the semantics (whatever the syntax might be).

4.1.4 Game semantics for CK

The purpose of this chapter is to provide a fully complete denotational semantics for CK in terms of a *game semantics* [2, 77, 100]. Thereby we provide a *concrete* denotational model for this logic, that is, a model whose elements are not obtained by the quotient on proofs induced by cut-elimination.

As mentioned before: in game semantics proofs are denoted by winning strategies for two-player games played on a graph, called *modal arena*, that encodes a modal formula. We denote the players by **O** (Opponent) and **P** (Proponent).

Each play consists of an alternation of **O**-moves and **P**-moves, that is, a play is represented by a list of occurrences of the vertices in the modal arena. The first move in a play is a **O**-move selected among the $\rightarrow$-roots of the modal arena. Each subsequent move of a player must be *justified* by a previous move of the other player, that is, the selected vertex must be the source of a $\rightarrow$-edge with target a vertex previously played by the other player. The game terminates when one player has no possible moves, losing the play.

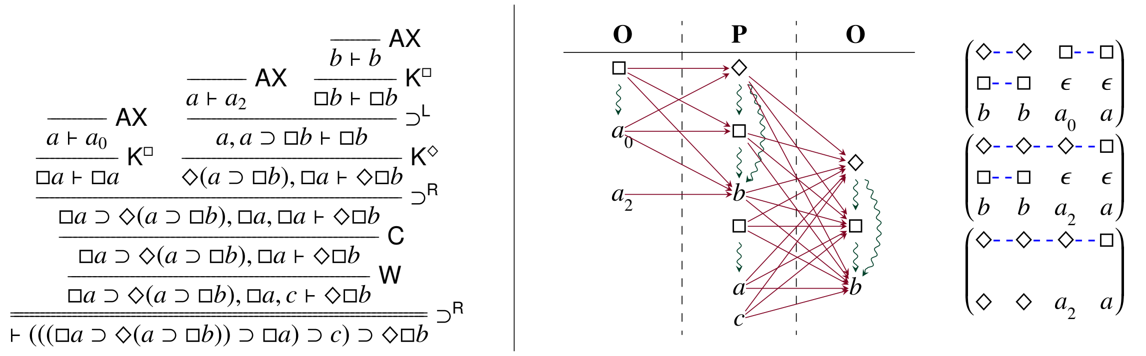
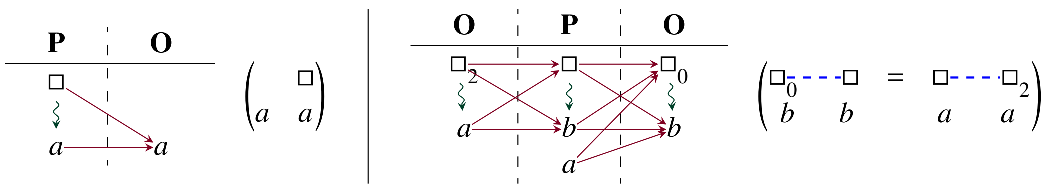


Figure 4.1: A derivation $\mathfrak{D}$ of the formula $F = (((\Box a_0 \supset \Diamond(a_2 \supset b)) \supset \Box a) \supset c) \supset \Diamond b$, the modal arena $[\![F]\!]$, and the maximal batched views in the CK-**WIS** $\{\!\{\mathfrak{D}\}\!\}$ of F. We indexed some occurrences of the atom a to avoid ambiguity in the views.

A *winning innocent strategy* (for **P**) is a set of plays which takes into account every possible **O**-move.

The adjective *innocent* is referred to the play-style of **O** which chooses each of its non-initial moves only according to the previous **P**-move in the play.

De facto, the presence of the modal axioms leads to the need of a new notion of *batches* in a play in order to characterize winning innocent strategies corresponding to proofs in the constructive modal logic CK. By means of example, consider the formulas $\Box a \supset a$ and $(\Box a \supset \Box b) \supset (\Box(a \supset b))$ which are not theorems of CK. Their corresponding modal arenas are pictured below, together with the unique maximal view in their winning innocent strategies. Instead of representing these views, we represent the corresponding *batched views*, which encode the view together with a decoration of each move given by the modalities in whose scope they occur.



(4.1)

The strategies containing these views cannot be considered satisfactory since the modalities are not "properly batched" with respect to the modal rules in the sequent calculus for CK. In fact, the **WIS** containing these maximal views correspond to correct proofs in the intuitionistic propositional logic of the formulas obtained by removing the modalities, that is, $a \supset a$ and $(a \supset b) \supset (a \supset b)$.

In order to recover the correspondence between winning strategies and proofs, it suffices to consider two additional constraints on the accepted **P**-moves. We observe that each modality has a parity (the same of the corresponding node in the modal arena) and a height (defined as the number of the modalities in whose scope it belongs). The first constraint demands that each **P**-move must be in the scope of the same number of modalities of the previous **O**-move, ruling out the leftmost example in Equation (4.1). This constraint allows us to define *sub-plays* (corresponding to sub-proofs): whenever a **O**-move is in the scope of a new **O**-modality, that is, a modality whose scope contains no previous moves of the play, then the successive moves are played in a same sub-play. A sub-play ends when a **O**-move is in the scope of no modalities or in the scope of a new **O**-modality with equal or smaller height with respect to the previous **P**-move. Note that sub-plays can be nested. This allows us to gather modalities having the same height and in whose scope there are moves of a sub-play into batches. The second constraint demands that these batches have a specific shape, that is, the same of the modalities in the rules of the sequent calculus: only one **O** modality occurs, and either all modalities are boxes or there is exactly one **P**-diamond and one **O**-diamond. These conditions rule out the existence of winning strategies for the formulas from Equation (4.1): in the first one the **P**-move has not the same height of the previous **O**-move, in the second one all the modalities are batched in the same set, which includes two **O** modalities.

Contribution of the chapter In this chapter we provide a direct correspondence between the sequent system for CK and our winning innocent strategies (CK-**WIS**). In particular, we show that the set of CK-**WIS**s form a fully complete denotational model for this logic.

Organization of the chapter. In section 4.2 we introduce the constructive modal logic CK; in particular we introduce a sound and complete sequent calculus system for this logic, that we call LCK, and we prove the cut-elimination theorem for this sequent calculus system. In section 4.3 we introduce the graph that will be used to code formulas in our game semantics i.e., *modal arenas*. In section 4.4 we recall the basic definitions of views and winning (innocent) strategies from game semantics and we enrich them in order to properly capture winning strategies corresponding to modal proofs that we call CK-**WIS**s for CK winning innocent strategies. In section 4.5 we prove that we can compose the winning strategies that we have defined in the preceding section i.e., that given two CK-**WIS**s $\mathcal{R}$ and $\mathcal{T}$ for $A \supset B$ and $B \supset C$ we can construct a CK-**WIS** for $A \supset C$ by "gluing" together $\mathcal{R}$ and $\mathcal{T}$ appropriately. In section 4.6 we define the CK-**WIS**s that will be used in section 4.7 to provide a denotational interpretation of LCK derivations. In section 4.8 we prove that the game semantic interpretation of LCK derivations is fully complete: each winning CK strategy is the image of some derivation in the sequent calculus. Section 4.9 sketch how, with a slight modification of the definitions presented in the previous section, we can obtain a fully complete game semantics for the constructive modal logic CD. Section 7.8 concludes the chapter providing some leads for extensions and future works.

4.2 Background

In this section we present the fragment of the constructive modal logic CK in which we are interested. Moreover, we provide a complete sequent calculus system LCK, for the constructive modal logic CK. We prove that this calculus enjoys the cut-elimination theorem.

4.2.1 The constructive modal logic CK

We consider the *(modal) formulas* generated by a countable set of (atomic) formulas $\mathcal{A} = \{1, a, b, \dots\}$[1] and the following grammar

$$\mathcal{F} = \mathcal{A} \mid \mathcal{F} \supset \mathcal{F} \mid \mathcal{F} \wedge \mathcal{F} \mid \Box\mathcal{F} \mid \Diamond\mathcal{F}$$

Arbitrary formulas will be denoted by upper case letters from the roman alphabet. We say that a formula is *modality-free* (respectively *unit-free*) if it contains no occurrences of $\Box$ and $\Diamond$ (respectively no occurrences of 1). A formula is a $\supset$-formula (resp. a $\wedge$ formula) if it is of the form $A \supset B$ (resp. $A \wedge B$).

The constructive modal logic CK is the smallest set of formulas containing

- all tautologies of minimal logic;
- each instance of the axiom k_1 : $\Box(A \supset B) \supset (\Box A \supset \Box B)$;
- each instance of the axiom k_2 : $\Box(A \supset B) \supset \Diamond A \supset \Diamond B$
- closed for *necessitation*: if $A \in \mathsf{CK}$ then $\Box A \in \mathsf{CK}$;
- closed for modus ponens: if $A \in \mathsf{CK}$ and $A \supset B \in \mathsf{CK}$ then $B \in CK$.

4.2.2 The sequent calculus LCK

We now present two sequent calculus system. We call the first one ,presented in Figure 4.2, LCK. We call the second one, presented in figure Figure 4.3, LCK*. Usually a sequent for constructive logic is defined as an expression $\Gamma \vdash C$ where C is a formula and the *context* Γ is either

[1] In this chapter we suppose that the set $\mathcal{A}$ does not contain $\bot$

1. a finite, possibly empty, *list* of formulas.

2. a finite, possibly empty, *multiset* of formulas.

The approach of point 1 is the one that was used by Gentzen in his PhD dissertation in which he introduced the sequent calculus. This definition of sequent has the advantage of making the notion of occurrence of a formula in a context clear, but has a major drawback: we are forced to introduce a structural rule, the exchange rule, to permute formulas in the context. The approach of point 2 is the one we chose in the previous chapter. In this way we do not need the exchange rule, since the order in which formulas appear in the context is irrelevant. Nevertheless, there is a drawback: the concept of occurrence of a formula in a context became fuzzy.

Once again we borrow an idea of Herbelin: suppose that we have two disjoint set $\mathcal{N}_H$ and $\mathcal{N}_C$ of *Hypothesis names* and *Conclusion name*. A named formula is a pair (A, n) where A is a formula and $n \in \mathcal{N}_H \cup \mathcal{N}_C$. We will say that n is the name of A or that A is named by n. A sequent is an expression $\Gamma \vdash C$. The context Γ is a set of named formulas, each name of a formula belongs to $\mathcal{N}_H$, and no two formula are named by the same element of $\mathcal{N}_H$. The conclusion C is a named formula named by an element of $\mathcal{N}_C$. If $\Gamma \vdash C$ is a sequent the expression $\mathcal{N}_\Gamma$ denotes the set of names of the formulas in Γ i.e. $\{n \in \mathcal{N}_H \,|\, (A, n) \in \Gamma\}$. In what follows we will use Greek upper case letters $(\Gamma, \Delta, \Lambda, \Sigma \ldots$ etc.) to denote a set of named formulas, and we will use Roman low-case letters from the end of the alphabet $(x, y, z, w, u \ldots$ etc.) to denote elements of $\mathcal{N}_H \cup \mathcal{N}_C$ Two sequents $\Gamma \vdash C$, $\Delta \vdash D$ are *compatible* iff D and C have different names and $\mathcal{N}(\Gamma) \cap \mathcal{N}(\Delta) = \emptyset$. If $\Gamma = (A_1, x_1), \ldots (A_n, x_n)$ and $\Delta = (B_1, y_1), \ldots (B_m, y_m)$ are two set of named formulas such that $n = m$ and for all $i \in \{1, \ldots n\}$ $A_i = B_i$ and $x_i \neq y_i$, then we will say that they are *twins*. A sequent calculus system is a set of inference rules. In an instance of an inference rule

$$\frac{\Gamma_1 \vdash A_1 \qquad \cdots \qquad \Gamma_n \vdash A_n}{\Delta \vdash C}$$

all the sequents $\Gamma_i \vdash A_i$ for $i \in \{1, \ldots n\}$ are compatible. We call them the *premises* of the rule. We call the unique sequent $\Delta \vdash A$ the *conclusion* of the rule.

A rule is classified with respect to the number of its premises. In the sequent calculi systems that we are going to present, rules have either zero premises, one premise or two premises, and we will call them, respectively, 0-ary rules, unary rules or binary rules. If a named formula A^x appears in the conclusion of a rule R but not in the premises of the rule, then the name of the formula does not appear in the premises of the rule. In this case we will say that R introduces A^x or that A^x is introduced by R. In figure 4.2 and 4.3 the expression $\Box\Gamma$ denotes $(\Box A_1)^{y_1}, \ldots, (\Box A_n)^{y_n}$ whenever Γ is $A_1{}^{x_1}, \ldots, A_n{}^{x_n}$ with $y_i \neq x_i$ for all $i \in \{1, \ldots, n\}$.

In the binary rules for the sequent system $\mathsf{LCK}^\star$, i.e., the two rules

$$\frac{\Gamma \vdash A^x \qquad \Gamma \vdash B^y}{\Gamma \vdash (A \wedge B)^z} \qquad \frac{\Gamma, (A \supset B)^v \vdash A^x \qquad \Gamma, B^y \vdash C^w}{\Gamma, (A \supset B)^z \vdash C^w}$$

The letter Γ represent sets of named formulas that are *twins*.

A derivation (or proof) $\mathfrak{D}$ of a sequent $\Gamma \vdash C^y$ in LCK (resp. $\mathsf{LCK}^\star$) is a finite tree of sequents constructed according to the rules showed in figure 4.2 (resp. in figure 4.3). The leaves of $\mathfrak{D}$ are instances of the 0-ary rules $\mathsf{1}$ or AX (we will sometimes call this rule axiom-rule). The root of the tree, also called conclusion of the proof or simply conclusion, is $\Gamma \vdash C^y$. A sequent $\Gamma \vdash C^y$ is provable in LCK (resp. $\mathsf{LCK}^\star$) iff there is an LCK (resp. $\mathsf{LCK}^\star$) derivation $\mathfrak{D}$ of $\Gamma \vdash C^y$.

The sequent calculus system $\mathsf{LCK}^\star$ of figure 4.3 is sound and complete for the constructive modal logic CK as it is showed in [87].

Moreover, the rules W, C, cut and the generalized axiom-rule

$$\frac{}{\Gamma, A^x \vdash A^y} \ \mathsf{AX}$$

where A is any formula and x, y any names are admissible in $\mathsf{LCK}^\star$. We can thus prove that also LCK is sound and complete for the constructive modal logic CK thanks to the following proposition

Proposition 4.1. *The sequent* $\Gamma \vdash C^y$ *is provable in* LCK *iff it is provable in* $\mathsf{LCK}^\star$

Proof. We prove that all rules of LCK are admissible in $\mathsf{LCK}^\star$ and that all rules of $\mathsf{LCK}^\star$ are admissible in LCK. The proof is by induction on the height $|\mathfrak{D}|$ of the derivation $\mathfrak{D}$ of $\Gamma \vdash C$ in LCK (resp. $\mathsf{LCK}^\star$). For the left to right direction we use the admissibility of the generalized Axiom-rule for the base case, and we use the admissibility of the rules C, W and cut for the inductive cases together with the rules of $\mathsf{LCK}^\star$

For the other direction, we use an instance of AX with A atomic and the W to derive an instance of the $\mathsf{AX}^\star$ and use use instances of C and W together with the rules of LCK for the inductive cases. $\Box$

$$
\frac{}{A^x \vdash A^y}\,\mathsf{AX}
\quad
\frac{\Gamma, B^x \vdash A^y}{\Gamma \vdash (B \supset A)^z}\,{\supset}^{\mathsf{R}}
\quad
\frac{\Gamma, B^x, C^y \vdash A^w}{\Gamma, (B \wedge C)^z \vdash A}\,{\wedge}^{\mathsf{L}}
\quad
\frac{\Gamma \vdash A^x \quad \Delta \vdash B^y}{\Gamma, \Delta \vdash (A \wedge B)^z}\,{\wedge}^{\mathsf{R}}
\quad
\frac{\Gamma \vdash A^x \quad B^y, \Delta \vdash C^w}{\Gamma, \Delta, (A \supset B)^z \vdash C^w}\,{\supset}^{\mathsf{L}}
\quad\Bigg|\quad
\frac{\Gamma \vdash A^x \quad A^y, \Delta \vdash B^w}{\Gamma, \Delta \vdash B^w}\,\mathsf{cut}
$$

$$
\frac{}{\vdash \mathsf{1}^x}\,\mathsf{1}
\quad
\frac{\Gamma, B^x, B^y \vdash A^w}{\Gamma, B^z \vdash A^w}\,\mathsf{C}
\quad
\frac{\Gamma \vdash A^x}{\Gamma, B^y \vdash A^x}\,\mathsf{W}
\quad
\frac{\Gamma \vdash A^x}{\Box\Gamma \vdash (\Box A)^y}\,\mathsf{K}^{\Box}
\quad
\frac{B^y, \Gamma \vdash A^x}{(\Diamond B)^z, \Box\Gamma \vdash (\Diamond A)^w}\,\mathsf{K}^{\Diamond}
$$

Figure 4.2: The rules for the sequent system LCK and the cut-rule

$$
\frac{}{\Gamma, a^x \vdash a^y}\,\mathsf{AX}^{\star}
\quad
\frac{\Gamma, B^x \vdash A^y}{\Gamma \vdash (B \supset A)^z}\,{\supset}^{\mathsf{R}\star}
\quad
\frac{\Gamma, B^x, C^y \vdash A^w}{\Gamma, (B \wedge C)^z \vdash A^w}\,{\wedge}^{\mathsf{L}\star}
\quad
\frac{\Gamma \vdash A^x \quad \Gamma \vdash B^y}{\Gamma, \vdash (A \wedge B)^z}\,{\wedge}^{\mathsf{R}\star}
\quad
\frac{\Gamma, A \supset B^v \vdash A^x \quad \Gamma, B^y \vdash C^w}{\Gamma, (A \supset B)^z \vdash C^w}\,{\supset}^{\mathsf{L}\star}
$$

$$
\frac{}{\Gamma \vdash \mathsf{1}^x}\,\mathsf{1}^{\star}
\quad
\frac{\Gamma \vdash A^x}{\Box\Gamma, \Delta \vdash (\Box A)^y}\,\mathsf{K}^{\Box\star}
\quad
\frac{B^y, \Gamma \vdash A^z}{(\Diamond B)^x, \Box\Gamma, \Delta \vdash (\Diamond A)^w}\,\mathsf{K}^{\Diamond\star}
$$

Figure 4.3: The rules for the sequent system LCK*

From the latter proposition one can easily deduce the following

Theorem 4.1. *The sequent calculus system* LCK *is sound and complete for the constructive modal logic* CK

From now on we will entirely forget the sequent calculus system LCK*. Every time that we will talk about derivations or proofs, we will be talking about proofs and derivations in LCK.

4.2.3 Cut elimination for LCK

We now prove that the cut-rule is redundant in LCK, i.e., that given a derivation $\mathfrak{D}$ of the sequent $\Gamma \vdash C$ containing instances of the cut-rule we can *construct* a derivation $\mathfrak{D}'$ of $\Gamma \vdash C$ in which no cut-rule is used. To do so, as it is usual, we define a series of transformations on quasi cut-free derivations. A quasi-cut free derivation is a derivation in which only one instance of the cut-rule is used, and it is the last rule of the derivation. These transformations, if opportunely applied, permit to transform a quasi cut-free derivation into a cut-free derivation of the same sequent. This technique naturally extends to a derivation $\mathfrak{D}$ in which more than one instance of the cut-rule appears: we first eliminate cuts on quasi cut-free sub-derivations of $\mathfrak{D}$.

Since we are working with named formulas, we must be careful in the definition of the transformation. Some transformation steps duplicate the sequents that are premises of the cut-rule and then apply contraction rules. We thus define the following notion

Definition 4.1. *A proof* $\mathfrak{D}'$ *with conclusion* $\Gamma \vdash C^z$ *is a variant of a proof* $\mathfrak{D}$ *with conclusion* $\Gamma' \vdash C^w$ *iff the sets of named formulas* Γ *and* Γ' *are twins and* $\mathfrak{D}'$ $\mathfrak{D}$ *are obtained by applying the same rules.*

Proposition 4.2. *For every proof* $\mathfrak{D}$ *there is a variant* $\mathfrak{D}'$ *of* $\mathfrak{D}$

Proof. Let A^x be any named formula in the context of the conclusion of $\mathfrak{D}$. The named formula A^x has been introduced in $\mathfrak{D}$ by a rule R. We choose a name v that does not appear in any sequent of $\mathfrak{D}$, and we let R introduce A^v instead of A^x. We repeat this procedure for any formula in the context of the conclusion of $\mathfrak{D}$. □

We now define the transformation on quasi cut-free derivations; for ease of reading (and for ease of writing) we will omit the superscripts of the named formulas, and we will simply signal to the reader when the notions of definition 4.1 are needed. The general form of a cut-rule is the following

$$\dfrac{\dfrac{\begin{matrix}\mathfrak{D}_1\\ \vdots\end{matrix}}{\Gamma \vdash F}\,R_1 \qquad \dfrac{\begin{matrix}\mathfrak{D}_2\\ \vdots\end{matrix}}{\Delta, F \vdash C}\,R_2}{\Gamma, \Delta \vdash C}$$

We define the transformations by looking at R_1 and R_2

($\wedge^{\mathsf{R}}/\wedge^{\mathsf{L}}$) if $R_1 = \wedge^{\mathsf{R}}$ is a rule that introduce $F = A \wedge B$ and $R_2 = \wedge^{\mathsf{R}}$ is a rule that introduce $F = A \wedge B$, then the derivation $\mathfrak{D}$ has the form

$$\dfrac{\dfrac{\begin{matrix}\mathfrak{D}_{1.1}\\ \vdots\\ \Gamma \vdash A\end{matrix} \qquad \begin{matrix}\mathfrak{D}_{1.2}\\ \vdots\\ \Delta \vdash B\end{matrix}}{\Gamma, \Delta \vdash A \wedge B}\,\wedge^{\mathsf{R}} \qquad \dfrac{\begin{matrix}\mathfrak{D}_{2.1}\\ \vdots\\ \Sigma, A, B \vdash C\end{matrix}}{\Sigma, A \wedge B \vdash C}\,\wedge^{\mathsf{L}}}{\Gamma, \Delta \vdash C}\,\mathsf{cut}$$

The transformation associated to $\mathfrak{D}$ is the following proof $\mathfrak{D}'$

$$\dfrac{\begin{matrix}\mathfrak{D}_{1.1}\\ \vdots \\ \Gamma \vdash A\end{matrix} \qquad \dfrac{\begin{matrix}\mathfrak{D}_{1.2}\\ \vdots\\ \Delta\vdash B\end{matrix} \qquad \begin{matrix}\mathfrak{D}_{2.1}\\ \vdots\\ \Sigma, A, B\vdash C\end{matrix}}{\Delta,\Sigma,A\vdash C}\,\mathsf{cut}}{\Gamma,\Delta\vdash C}\,\mathsf{cut}$$

($\supset^{\mathsf{R}}/\supset^{\mathsf{L}}$) if $R_1 = \supset^{\mathsf{R}}$ is a rule that introduce $F = A \supset B$ and $R_2 = \supset^{\mathsf{L}}$ is a rule that introduce $F = A \supset B$, then $\mathfrak{D}$ has the form

$$\dfrac{\dfrac{\begin{matrix}\mathfrak{D}_{1.1}\\ \vdots \\ \Gamma, A \vdash B\end{matrix}}{\Gamma \vdash A \supset B}\,\supset^{\mathsf{R}} \qquad \dfrac{\begin{matrix}\mathfrak{D}_{2.1}\\ \vdots\\ \Delta\vdash A\end{matrix} \qquad \begin{matrix}\mathfrak{D}_{2.2}\\ \vdots\\ B, \Sigma\vdash C\end{matrix}}{A \supset B, \Delta, \Sigma \vdash C}\,\supset^{\mathsf{L}}}{\Gamma,\Delta,\Sigma\vdash C}\,\mathsf{cut}$$

The transformation associated to $\mathfrak{D}$ is the following proof $\mathfrak{D}'$

$$\dfrac{\begin{matrix}\mathfrak{D}_{1.1}\\ \vdots \\ \Delta \vdash A\end{matrix} \qquad \dfrac{\begin{matrix}\mathfrak{D}_{1.2}\\ \vdots\\ \Gamma, A\vdash B\end{matrix} \qquad \begin{matrix}\mathfrak{D}_{2.1}\\ \vdots\\ B, \Sigma\vdash C\end{matrix}}{\Gamma,\Sigma,A\vdash C}\,\mathsf{cut}}{\Gamma,\Delta,\Sigma\vdash C}\,\mathsf{cut}$$

($\mathsf{K}^{\Box}/\mathsf{K}^{\Box}$) if $R_1 = \Box$ is a rule that introduce $F = \Box A$ and $R_2 = \Box A$ is a rule that introduce $F = \Box A$, then $\mathfrak{D}$ has the following form

$$\dfrac{\dfrac{\begin{matrix}\mathfrak{D}_{1.1}\\ \vdots \\ \Gamma \vdash A\end{matrix}}{\Box\Gamma \vdash \Box A}\,\mathsf{K}^{\Box} \qquad \dfrac{\begin{matrix}\mathfrak{D}_{2.1}\\ \vdots\\ \Delta, A\vdash C\end{matrix}}{\Box\Delta, \Box A \vdash \Box C}\,\mathsf{K}^{\Box}}{\Box\Gamma,\Box\Delta\vdash \Box C}\,\mathsf{cut}$$

The transformation associated to $\mathfrak{D}$ is the following proof $\mathfrak{D}'$

$$\dfrac{\dfrac{\begin{matrix}\mathfrak{D}_{1.1}\\ \vdots \\ \Gamma \vdash A\end{matrix} \qquad \begin{matrix}\mathfrak{D}_{2.1}\\ \vdots \\ \Delta, A \vdash C\end{matrix}}{\Gamma, \Delta, \vdash C}\,\mathsf{cut}}{\Box\Gamma, \Box\Delta \vdash \Box C}\,\mathsf{K}^{\Box}$$

($\mathsf{K}^{\Box}/\mathsf{K}^{\Diamond}$) if $R_1 = \mathsf{K}^{\Box}$ is a rule that introduce $F = \Box A$ and $R_2 = \mathsf{K}^{\Diamond}$ is a rule that introduce $F = \Box A$, then $\mathfrak{D}$ has the following form

$$\dfrac{\dfrac{\begin{matrix}\mathfrak{D}_{1.1}\\ \vdots \\ \Gamma \vdash A\end{matrix}}{\Box\Gamma \vdash \Box A}\,\mathsf{K}^{\Box} \qquad \dfrac{\begin{matrix}\mathfrak{D}_{2.1}\\ \vdots \\ \Delta, B, , A \vdash C\end{matrix}}{\Box\Delta, \Diamond B, \Box A \vdash \Diamond C}\,\mathsf{K}^{\Diamond}}{\Box\Gamma, \Box\Delta, \Diamond B \vdash \Diamond C}\,\mathsf{cut}$$

The transformation associate to $\mathfrak{D}$ is the following proof $\mathfrak{D}'$

$$\dfrac{\dfrac{\begin{matrix}\mathfrak{D}_{1.1}\\ \vdots \\ \Gamma \vdash A\end{matrix} \qquad \begin{matrix}\mathfrak{D}_{2.1}\\ \vdots \\ \Delta, B, A \vdash C\end{matrix}}{\Gamma, \Delta, B \vdash C}\,\mathsf{cut}}{\Box\Gamma, \Box\Delta, \Diamond B \vdash \Diamond C}\,\mathsf{K}^{\Diamond}$$

($\mathsf{K}^{\Diamond}/\mathsf{K}^{\Diamond}$) if $R_1 = \mathsf{K}^{\Diamond}$ is a rule that introduces $F = \Diamond A$ and $R_2 = \mathsf{K}^{\Diamond}$ is a rule that introduces $F = \Diamond A$, then $\mathfrak{D}$ has the following form

$$\dfrac{\dfrac{\begin{matrix}\mathfrak{D}_{1.1}\\ \vdots \\ \Gamma, B \vdash A\end{matrix}}{\Box\Gamma, \Diamond B \vdash \Diamond A}\,\mathsf{K}^{\Diamond} \qquad \dfrac{\begin{matrix}\mathfrak{D}_{2.1}\\ \vdots \\ \Delta, A \vdash C\end{matrix}}{\Box\Delta, \Diamond A \vdash \Diamond C}\,\mathsf{K}^{\Diamond}}{\Box\Gamma, \Box\Delta, \Diamond B \vdash \Diamond C}\,\mathsf{cut}$$

The transformation associated to $\mathfrak{D}$ is the following derivation $\mathfrak{D}'$

$$\dfrac{\dfrac{\begin{matrix}\mathfrak{D}_{1.1}\\ \vdots\\ \Gamma, B \vdash A\end{matrix} \qquad \begin{matrix}\mathfrak{D}_{2.1}\\ \vdots\\ \Delta, A \vdash C\end{matrix}}{\Gamma, \Delta, B \vdash C}\,\mathsf{cut}}{\Box\Gamma, \Box\Delta, \Diamond B \vdash \Diamond C}\,\mathsf{K}^{\Diamond}$$

The cases that we have treated will be called *key-cases* or *logical-cases*

(AX) If one between $\mathfrak{D}_1$ and $\mathfrak{D}_2$ is a derivation obtained by an instance of the AX-rule, then it is of conclusion $F \vdash F$. Suppose it is $\mathfrak{D}_1$ (the case in which is $\mathfrak{D}_2$ being perfectly symmetrical). Thus $\mathfrak{D}_2$ has conclusion $\Delta, F \vdash C$ and the transformation associated to $\mathfrak{D}$ is just $\mathfrak{D}_2$

W If $R_2 = \mathsf{W}$ is a rule that introduce F, then $\mathfrak{D}$ has the following form

$$\dfrac{\dfrac{\begin{matrix}\mathfrak{D}_1\\ \vdots\end{matrix}}{\Gamma \vdash F}\,R_1 \qquad \dfrac{\begin{matrix}\mathfrak{D}_2\\ \vdots\\ \Delta \vdash C\end{matrix}}{F, \Delta \vdash C}\,\mathsf{W}}{\Gamma, \Delta \vdash C}\,\mathsf{cut}$$

The transformation associate to $\mathfrak{D}$ is the following proof $\mathfrak{D}'$ in which the multiple lines indicate many instances of the weakening rule

$$\begin{matrix}\mathfrak{D}_2\\ \vdots\\ \Delta \vdash C\\ \hline\hline \Gamma, \Delta \vdash C\end{matrix}$$

C if $R_2 = \mathsf{C}$ is a rule that introduce F, then $\mathfrak{D}$ has the following form

$$\dfrac{\dfrac{\begin{array}{c}\mathfrak{D}_1\\ \vdots\end{array}}{\Gamma \vdash F}R_1 \qquad \dfrac{\begin{array}{c}\mathfrak{D}_2\\ \vdots\\ F, F, \Delta \vdash C\end{array}}{F, \Delta \vdash C}\mathsf{C}}{\Gamma, \Delta \vdash C}\mathsf{cut}$$

The transformation associated to $\mathfrak{D}$ is the following proof $\mathfrak{D}'$

$$\dfrac{\dfrac{\begin{array}{c}\mathfrak{D}_1''\\ \vdots\end{array}}{\Gamma \vdash F}R_1 \qquad \dfrac{\dfrac{\begin{array}{c}\mathfrak{D}_1'\\ \vdots\end{array}}{\Gamma \vdash F}R_1 \qquad \begin{array}{c}\mathfrak{D}_2\\ \vdots\\ F, F, \Delta \vdash C\end{array}}{F, \Gamma, \Delta \vdash C}\mathsf{cut}}{\dfrac{\Gamma, \Gamma, \Delta \vdash C}{\Gamma, \Delta \vdash C}}\mathsf{cut}$$

Here the two proofs $\mathfrak{D}_1{}'$ and $\mathfrak{D}_1{}''$ are two different variants of $\mathfrak{D}_1$ and the renaming of the named formulas in the two occurrences of Γ by the successive application of the contraction rule gives the same named formulas of the conclusion of $\mathfrak{D}_1$

(cc)[2] If one between R_1 and R_2 is a rule that introduces a formula A that is not the cut-formula F, then we remark that neither R_1 nor R_2 can be a modal rules, and we proceed as follows; if it is R_1 then R_1 is one between $\supset^{\mathsf{L}}$, $\wedge^{\mathsf{L}}$ and A is a formula in Γ. In the case $R_1 = \supset^{\mathsf{L}}$, then $A = B \supset D$, $\Gamma = \Gamma_1, \Gamma_2$ and $\mathfrak{D}$ has the following form

$$\dfrac{\dfrac{\begin{array}{c}\mathfrak{D}_{1.1}\\ \vdots\\ \Gamma_1 \vdash B\end{array} \qquad \begin{array}{c}\mathfrak{D}_{1.2}\\ \vdots\\ D, \Gamma_2 \vdash F\end{array}}{\Gamma_1, \Gamma_2, B \supset D \vdash F}\supset^{\mathsf{L}} \qquad \dfrac{\begin{array}{c}\mathfrak{D}_2\\ \vdots\end{array}}{\Delta, F \vdash C}R_2}{\Gamma, \Delta, B \supset D \vdash C}\mathsf{cut}$$

The transformation associated to $\mathfrak{D}$ is the following proof $\mathfrak{D}'$

$$\dfrac{\begin{matrix}\mathfrak{D}_{1.1}\\ \vdots\\ \Gamma_1 \vdash B\end{matrix} \qquad \dfrac{\begin{matrix}\mathfrak{D}_{1.2}\\ \vdots\\ D, \Gamma_2 \vdash F\end{matrix} \qquad \dfrac{\begin{matrix}\mathfrak{D}_2\\ \vdots\end{matrix}}{\Delta, F \vdash C}\, R_2}{\Gamma_2, \Delta, C \vdash C}\,\mathsf{cut}}{\Gamma, \Delta, B \supset C \vdash C}\,\supset^{\mathsf{L}}$$

If $R_1 = \wedge^{\mathsf{L}}$, W or C then $\mathfrak{D}$ has the following schematic form

$$\dfrac{\dfrac{\begin{matrix}\mathfrak{D}_{1.1}\\ \vdots\\ \Gamma_1 \vdash F\end{matrix}}{\Gamma \vdash F}\, R_1 \qquad \dfrac{\begin{matrix}\mathfrak{D}_2\\ \vdots\end{matrix}}{\Delta, F \vdash C}\, R_2}{\Gamma, \Delta \vdash C}\,\mathsf{cut}$$

The transformation associated to $\mathfrak{D}$ is the following proof $\mathfrak{D}'$

$$\dfrac{\dfrac{\begin{matrix}\mathfrak{D}_{1.1}\\ \vdots\\ \Gamma_1 \vdash F\end{matrix} \qquad \dfrac{\begin{matrix}\mathfrak{D}_2\\ \vdots\end{matrix}}{\Delta, F \vdash C}\, R_2}{\Gamma_1, \Delta \vdash C}\,\mathsf{cut}}{\Gamma, \Delta \vdash C}\, R_1$$

If the rule is R_2, we have all the cases we have encountered for R_1 plus the following two : if R_2 is a rule introducing the formula C, then R_2 can be either a $\supset^{\mathsf{R}}$ or a $\wedge^{\mathsf{R}}$. We treat only the case of the $\supset^{\mathsf{R}}$. The derivation $\mathfrak{D}$ has the form

$$\dfrac{\dfrac{\begin{matrix}\mathfrak{D}_1\\ \vdots\end{matrix}}{\Gamma \vdash F}\,R_1 \qquad \dfrac{\begin{matrix}\mathfrak{D}_2\\ \vdots \\ F,\Delta,B\vdash D\end{matrix}}{F,\Delta\vdash B\supset D}\,\supset^{\mathsf{R}}}{\Gamma,\Delta\vdash B\supset B}\,\mathsf{cut}$$

The transformation associated to $\mathfrak{D}$ is the following proof $\mathfrak{D}'$

$$\dfrac{\dfrac{\dfrac{\begin{matrix}\mathfrak{D}_1\\ \vdots\end{matrix}}{\Gamma \vdash F}\,R_1 \qquad \begin{matrix}\mathfrak{D}_2\\ \vdots \\ F,\Delta,B\vdash D\end{matrix}}{\Gamma,\Delta,B\vdash D}\,\mathsf{cut}}{\Gamma,\Delta\vdash B\supset D}\,\supset^{\mathsf{R}}$$

To prove the cut elimination theorem we use a technique developed by Abrusci and Tortora De Falco in their book [3].[3]

We define a set $\mathcal{T}_{\text{glob}}$ of 'global' transformation steps. The global transformation steps allow us to transform any derivation into a derivation without cuts of the same conclusion. The global transformation steps are obtained by composing, in a specific way, the elementary transformation steps we have just introduced.

Remark that we can divide occurrences of the cut-rule into two families

1. Logical cases : these are the cuts $(\wedge^{\mathsf{R}}/\wedge^{\mathsf{L}})$, $(\supset^{\mathsf{R}}/\supset^{\mathsf{L}})$, $(\mathsf{K}^{\Box}/\mathsf{K}^{\Box})$, $(\mathsf{K}^{\Box}/\mathsf{K}^{\Diamond})$ and $(\mathsf{K}^{\Diamond}/\mathsf{K}^{\Diamond})$. A generic logical cut will by denoted by (L)

2. Structural cases: these are all the other cases. A generic structural-cut will by denoted by (S)

[3] We attribute the paternity of this technique to Abrusci and Tortora de Falco, we are unaware whether it occurs somewhere else in the proof-theory literature

Remark that to each logical cut we can apply only one transformation. We call such transformation *logical step*. This is not true of structural cuts: we can apply different transformations — that we will call structural step— but in any case, not a logical transformation. We now define the set $\mathcal{T}_{\text{glob}}$: this set contains each elementary logical step and the structural step obtained by composing the structural steps $(\mathsf{AX}), (\mathsf{C}), (\mathsf{W})$ and (cc). We also use (L) tho denote the generic logical step. We should now define precisely the structural step (S): such a step is applied when no logical step can be applied, that is, when (with the chosen notations) at least one of the two between the rules R_1 and R_2 in $\mathfrak{D}$ is not a logical rule that introduces the cut-formula F. We now reintroduce the names of the formulas, since they are needed to precisely define the structural step (S).

Let us suppose that R_2 is a rule that does not introduce the cut occurrence of formula F^y in $F^y, \Delta \vdash C^w$. We can trace the 'history' of F^y in $\mathfrak{D}_2$. More precisely, we collect the formula occurrences of F that are 'recursively contracted' into F^y together with F^y. First, let us define some auxiliary notions.

Let $\mathfrak{D}$ be a proof, $\Gamma \vdash C^y$ be a sequent in $\mathfrak{D}$ and F^x a formula occurrence that appears in $\Gamma \vdash C^y$. The sequent $\Gamma \vdash C^y$ is the *cradle* of F^x iff F^x does not appear in any sequent above $\Gamma \vdash C^y$.

Definition 4.2. *Let F^x be a formula occurrence and let $\mathfrak{D}$ be a derivation. The history $\mathcal{H}^{\mathfrak{D}}_{F^x}$ of F^x in $\mathfrak{D}$ is the set of formula occurrences defined as follows*

- *if $\mathfrak{D}$ is obtained by an AX-rule or a $\mathsf{1}$-rule and $\mathfrak{D}$ is not the* cradle *of F then $\mathcal{H}^{\mathfrak{D}}_{F} = \emptyset$*
- *If $\mathfrak{D}$ is obtained by an AX-rule or a $\mathsf{1}$-rule and $\mathfrak{D}$ is the* cradle *of F then $\mathcal{H}^{\mathfrak{D}}_{F^x} = \{F^x\}$*
- *if $\mathfrak{D}$ is a proof with immediate sub-proofs $\mathfrak{D}_1, \ldots \mathfrak{D}_n$ and the conclusion of $\mathfrak{D}$ is not the cradle of F^x then $\mathcal{H}^{\mathfrak{D}}_{F^x} = \bigcup_{i \leq n} \mathcal{H}^{\mathfrak{D}_i}_{F^x}$*
- *if $\mathfrak{D}$ is a proof with immediate sub-proofs $\mathfrak{D}_1 \ldots \mathfrak{D}_n$ and the conclusion of $\mathfrak{D}$ is the cradle of F^x we have two sub-cases*
    1. *If the conclusion of $\mathfrak{D}$ is obtained by a logical rule R, or by a weakening rule W then $\mathcal{H}^{\mathfrak{D}}_{F^x} = \bigcup_{i \leq n} \mathcal{H}^{\mathfrak{D}_i}_{F^x} \cup \{F^x\}$*
    2. *If the conclusion of $\mathfrak{D}$ is obtained by a contraction rule C then, by calling $\mathfrak{D}_1$ the immediate sub-proof of $\mathfrak{D}$ with conclusion $\Gamma', F^y, F^z \vdash C'^w$, $\mathcal{H}^{\mathfrak{D}}_{F^x} = \mathcal{H}^{\mathfrak{D}_1}_{F^y} \cup \mathcal{H}^{\mathfrak{D}_1}_{F^z} \cup \{F^x\}$*

A formula occurrence $F^z \in \mathcal{H}^{\mathfrak{D}}_{F^x}$ will be called a leaf *iff $\mathcal{H}^{\mathfrak{D}'}_{F^z} = \{F^z\}$ where $\mathfrak{D}'$ is the sub-proof of $\mathfrak{D}$ with conclusion the cradle of F^z*

Remark that by the definition of $\mathcal{H}^{\mathfrak{D}}_{F^x}$ the leaves of this set will be formula occurrences of F that are introduced by a logical rule, by an axiom rule or by a weakening rule.

We can now define precisely define the step (S).

- for all leaves of $\mathcal{H}^{\mathfrak{D}}_{F^y}$ that are obtained in $\mathfrak{D}_2$ by an AX-rule $F \vdash F$. We substitute a variant of the proof $\mathfrak{D}_1$ of $\Gamma \vdash F^x$ the axiom rule in $\mathfrak{D}_2$.
- for all leaves F^{y_i} of $\mathcal{H}^{\mathfrak{D}_2}_{F^y}$ that are obtained in $\mathfrak{D}_2$ by a logic rules R. Let us denote by α the derivation above R in $\mathfrak{D}_2$. We substitute α in $\mathfrak{D}_2$ with the following derivation

$$\dfrac{\dfrac{\begin{matrix}\mathfrak{D}_1\\ \vdots\end{matrix}}{\Gamma \vdash F^x}\,R_1 \qquad \dfrac{\begin{matrix}\alpha\\ \vdots\end{matrix}}{\Delta', F^{y_i} \vdash C'^{w'}}}{\Gamma, \Delta' \vdash C'^{w'}}\,\mathrm{R}$$

  Where the sequent $\Delta', F^{y_i} \vdash C'^{w'}$ is the cradle of F^{y_i}

- for all leaves of $\mathcal{H}^{\mathfrak{D}}_{F^y}$ that are obtained in $\mathfrak{D}_2$ by a W-rule with premise $\Lambda \vdash C'^{w'}$. We substitute in $\mathfrak{D}$ the instance of W with the number of application of the W needed to obtain the sequent $\Gamma, \Delta \vdash C'^{w'}$. All the name of formulas in Γ being fresh.

By applying this procedure, we obtain that in $\mathfrak{D}$ every leaf of $\mathcal{H}^{\mathfrak{D}_2}_{F^y}$ is substituted everywhere by Γ. Remark that the only rules that are applied to occurrences of F in $\mathfrak{D}_2$ that are not leaf of $\mathcal{H}^{\mathfrak{D}_2}_{F_x}$ are C-rules. We apply those rules to Γ the right number of times and thus obtain a derivation of $\Gamma, \Delta \vdash C$.

It is evident that the step (S) is obtained by composing together in a certain way reduction steps that are not logical steps. Let us now state the following definition.

Definition 4.3. *Let R by a structural* cut*-rule, with cut-formula F, and such that the premises of R are conclusion of a derivation $\mathfrak{D}_1$ and of a derivation $\mathfrak{D}_2$ in which the last rules are, respectively, R_1 and R_2.*

- *if both R_1 and R_2 are rules that do not introduce F then we say that the cut-rule R is of type S_1*
- *it is of type S_2 otherwise.*

The following lemma is evident by the definition of the global structural step (S).

Lemma 4.1. *Let R be a structural cut in an almost cut-free derivation $\mathfrak{D}$ of $\Gamma \vdash C$*

1. *If S is of type S_1 then by applying the global step (S) to R we obtain a derivation $\mathfrak{D}'$ of $\Gamma \vdash C$ in which all instance of the cut-rule (if any) are of type S_2*
2. *If S is of type S_2 then by applying the global step (S) to R we obtain a derivation $\mathfrak{D}'$ of $\Gamma \vdash C$ in which all instance of the cut-rule (if any) are logical cut*

Let $\mathfrak{D}$ and almost cut-free derivation with cut-rule R. The energy $en(\mathfrak{D})$ of $\mathfrak{D}$ is 2 if R is a cut of type S_1, 1 is if R is a cut of type S_2 and 0 otherwise. The degree of a cut-rule is the degree of its cut-formula. The degree $\sigma(\mathfrak{D})$ of an almost cut-free derivation is the degree of the cut-rule of $\mathfrak{D}$ We can finally prove the following

Proposition 4.3. *Let $\mathfrak{D}$ be an almost* cut*-free derivation of the sequent $\Gamma \vdash C$. If we apply only reduction steps in $\mathcal{T}_{glob}$ then $\mathfrak{D}$ can be transformed into a* cut*-free derivation $\mathfrak{D}'$ of $\Gamma \vdash C$.*

Proof. By induction on the couple $(\delta(\mathfrak{D}), en(\mathfrak{D}))$ using lemma 4.1 and the fact that the degree of the cut-formula strictly decreases whenever we apply logical steps of transformation. □

It is now possible to prove the following theorem by induction on the number n of cut-rule in $\mathfrak{D}$.

Theorem 4.2. *There exists a procedure P which associate to each derivation $\mathfrak{D}$ of $\Gamma \vdash C$ a* cut*-free derivation $\mathfrak{D}'$ with the same conclusion $\Gamma \vdash C$.*

4.3 Modal Arenas

In this section we provide the definitions of the graphs that encodes formulas in game semantics. They are called *arenas*. The name is due to the fact that the zero-sum games we are going to define, takes places 'inside' these graphs as, in the roman period, the deadly fights between gladiators took place inside arenas. Arenas are like chessboards: they form the environment in which a game can take place, and their configuration impose restrictions on the type of moves the players can make. Being more precise: the games that we are going to define in the next session are sequences of vertices of arenas. An arena will be defined as a directed graph with two kinds of edges and in which vertex are labelled with either propositional variables, the $\Box$ modality or the $\Diamond$-modality. The two kind of edges takes into account, respectively, the structure of modality free formulas and the scope of modalities.

A *directed graph* $\mathcal{G} = \langle V_{\mathcal{G}}, \overset{\mathcal{G}}{\to} \rangle$ is given by a set of vertices $V_{\mathcal{G}}$ and a set of direct edges $\overset{\mathcal{G}}{\to} \subseteq V_{\mathcal{G}} \times V_{\mathcal{G}}$. A vertex v is a $\overset{\mathcal{G}}{\to}$*-root*, denoted $v \not\to$ if there is no vertex w such that $v \overset{\mathcal{G}}{\to} w$. We denote by $\vec{R}_{\mathcal{G}}$ the set of $\overset{\mathcal{G}}{\to}$-roots of $\mathcal{G}$. A *path* from v to w of length n is a sequence of vertices $x_0 \dots x_n$ such that $v = x_0$, $w = x_n$ and $x_i \overset{\mathcal{G}}{\to} x_{i+1}$ for $i \in \{0, \dots, n-1\}$. We write $v \overset{\mathcal{G}}{\to}{}^* w$ if there is a path from v to w. A *directed acyclic graph* (or dag for short) is a directed graph such that $v \overset{\mathcal{G}}{\to}{}^n v$ implies $n = 0$ for all $v \in V$.

A *two-color directed acyclic graph* (or 2-dag for short) $\mathcal{G} = \langle V_{\mathcal{G}}, \overset{\mathcal{G}}{\to}, \overset{\mathcal{G}}{\rightsquigarrow} \rangle$ is given by a set of vertices $V_{\mathcal{G}}$ and two disjoint sets of edges $\overset{\mathcal{G}}{\to}$ and $\overset{\mathcal{G}}{\rightsquigarrow}$ such that the graph $\langle V_{\mathcal{G}}, \overset{\mathcal{G}}{\to} \cup \overset{\mathcal{G}}{\rightsquigarrow} \rangle$ is acyclic. We denote $\overset{\mathcal{G}}{\leftrightarrow} = \overset{\mathcal{G}}{\to} \cup \overset{\mathcal{G}}{\leftarrow}$, $\overset{\mathcal{G}}{\leftrightsquigarrow} = \overset{\mathcal{G}}{\rightsquigarrow} \cup \overset{\mathcal{G}}{\leftsquigarrow}$. We omit the superscript when clear from context and we denote by $\emptyset$ the empty 2-dag

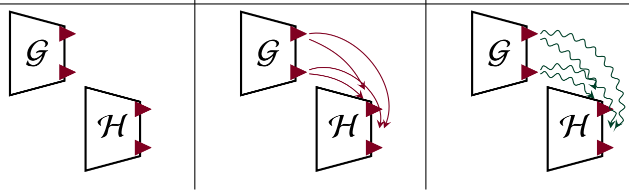
Definition 4.4. *Let $\mathcal{G}$, $\mathcal{H}$ and $\mathcal{F} \neq \emptyset$ be* 2-dag*s, we denote by $R^{\mathcal{G}}_{\mathcal{F}}$ the set of edges from the $\to$-roots of $\mathcal{G}$ to the $\to$-roots of $\mathcal{F}$, that is $R^{\mathcal{G}}_{\mathcal{F}} = \{(u, v) \mid u \in \vec{R}_{\mathcal{G}}, v \in \vec{R}_{\mathcal{F}}\}$.*

We define the following operations on 2-dag*s:*

$$\begin{array}{l}
\mathcal{G} + \mathcal{H} = \langle V_{\mathcal{G}} \uplus V_{\mathcal{H}}, \overset{\mathcal{G}}{\to} \uplus \overset{\mathcal{H}}{\to}, \overset{\mathcal{G}}{\rightsquigarrow} \uplus \overset{\mathcal{H}}{\rightsquigarrow} \rangle \\
\mathcal{G} \rightarrowtriangle \mathcal{F} = \langle V_{\mathcal{G}} \uplus V_{\mathcal{F}}, \overset{\mathcal{G}}{\to} \uplus \overset{\mathcal{F}}{\to} \uplus R^{\mathcal{G}}_{\mathcal{F}}, \overset{\mathcal{G}}{\rightsquigarrow} \uplus \overset{\mathcal{F}}{\rightsquigarrow} \rangle \\
\mathcal{G} \rightsquigarrow\!\triangleright \mathcal{F} = \langle V_{\mathcal{G}} \uplus V_{\mathcal{F}}, \overset{\mathcal{G}}{\to} \uplus \overset{\mathcal{F}}{\to}, \overset{\mathcal{G}}{\rightsquigarrow} \uplus \overset{\mathcal{F}}{\rightsquigarrow} \uplus R^{\mathcal{G}}_{\mathcal{F}} \rangle
\end{array}$$

$$\mathcal{G} \rightarrowtriangle \emptyset = \emptyset \qquad \Box \rightsquigarrow\!\triangleright \emptyset = \emptyset \qquad \Diamond \rightsquigarrow\!\triangleright \emptyset = \Diamond$$

Where $\uplus$ *is the disjoint union i.e.,* $A \uplus B = A \times \{0\} \cup B \times \{1\}$. *The operations can be pictured as follows, with* $\blacktriangleright$ *representing the* $\to$*-roots of each graph.*

$\mathcal{G}+\mathcal{H}$	$\mathcal{G}\rightarrowtriangle\mathcal{H}$	$\mathcal{G}\leadsto\mathcal{H}$	$\mathcal{G}\rightarrowtriangle\emptyset = \Box\leadsto\emptyset$	$\Diamond\leadsto\emptyset$
			$\emptyset$	$\Diamond$

That is :

- The operation $+$ puts the two 2-dags $\mathcal{G}$ and $\mathcal{H}$ side by side without adding any new edge between the two.
- The operation $\rightarrowtriangle$ adds a directed edge $\to$ from every $\to$-root of $\mathcal{G}$ to every $\to$-root of $\mathcal{H}$
- The operation $\leadsto$ adds a directed edge $\rightsquigarrow$ from every $\to$-root of $\mathcal{G}$ to every $\to$-root of $\mathcal{H}$

We can associate to each formula F a $\mathcal{L}$-labeled 2-dag $[\![F]\!]$ as follows by induction on the depth $|F|$ of F.

$$\begin{array}{ccccccccccc} [\![a]\!] & = & a & [\![A \supset B]\!] & = & [\![A]\!]\rightarrowtriangle[\![B]\!] & [\![A \wedge B]\!] & = & [\![A]\!]+[\![B]\!] & [\![1]\!] & = \emptyset \\ & & & [\![\Box A]\!] & = & \Box \leadsto[\![A]\!] & [\![\Diamond A]\!] & = & \Diamond \leadsto[\![A]\!] & & \end{array} \tag{4.2}$$

Definition 4.5. *A* modal arena *is a* dag $\mathcal{G}$ *such that* $\mathcal{G} = [\![F]\!]$ *for a modal formula* F.

if $\Gamma \vdash A$ is a sequent, with $\Gamma = B_1, B_2, \ldots B_{n-1}, B_n$ we denote by $[\![\Gamma \vdash A]\!]$ the modal arena

$$[\![\ (\ldots(B_1 \wedge B_2)\ldots) \wedge B_{n-1}) \wedge B_n) \supset A]\!]$$

By construction, a vertex v in a modal arena $[\![A]\!]$ has the form

$$v = ((\ldots((x, i_1), i_2))\ldots i_n))$$

where $i_j \in \{0, 1\}$ for every $j \in \{1, \ldots n\}$ and $x \in \mathcal{A} \cup \{\Box, \Diamond\}$. We call the vertex x the *label* of the vertex v, and we denote it by $\ell(v)$. We say that v is a modal vertex if $\ell(v) = \Box$ or

$\ell(v) = \Diamond$. We say that v is an atomic vertex otherwise. If $\mathcal{G}$ is a modal arena, we denote by $V^{\Box}_{\mathcal{G}}$ the set of vertices of $\mathcal{G}$ that are labeled by a $\Box$ and by $V^{\Diamond}_{\mathcal{G}}$ the set of vertices of $\mathcal{G}$ that are labeled by a $\Diamond$ modality. We denote by $V^{\mathbf{P}}_{\mathcal{G}}$ the set of odd vertices of $\mathcal{G}$ and by $V^{\mathbf{O}}_{\mathcal{G}}$ the set of even vertices of $\mathcal{G}$.

When no confusion can arise, we will tend to identify a vertex v with its label $\ell(v)$. This will drastically simplify the exposition and the drawings present in this chapter.

Definition 4.6. *Let $\mathcal{G} = \langle V_{\mathcal{G}}, \overset{\mathcal{G}}{\to}, \overset{\mathcal{G}}{\rightsquigarrow} \rangle$ and $\mathcal{H} = \langle V_{\mathcal{H}}, \overset{\mathcal{H}}{\to}, \overset{\mathcal{H}}{\rightsquigarrow} \rangle$ be two* 2-dag *and f a bijective map from $\mathcal{G}$ to $\mathcal{H}$. We say that f is* 2-dag *isomorphism or simply an isomorphism iff it preserves labels and edges i.e. $\ell(x) = \ell(f(x))$, $x \overset{\mathcal{G}}{\to} y$ iff $f(x) \overset{\mathcal{H}}{\to} f(y)$ and $x \overset{\mathcal{G}}{\rightsquigarrow} y$ iff $f(x) \overset{\mathcal{H}}{\rightsquigarrow} f(y)$ for all x, y in $V_{\mathcal{G}}$. We write $\mathcal{G} \overset{f}{\sim} \mathcal{H}$ whenever f is an* 2-dag-*isomorphism $f : V_{\mathcal{G}} \to V_{\mathcal{H}}$*

Proposition 4.4. *[Arena isomorphism] Let A, B and C be three arbitrary formulas. There exists* 2-dag-*isomorphism r_A, nil, g, m_0, c, a and λ such that*

$$[\![A \wedge 1]\!] \overset{r_A}{\sim} [\![A]\!] \qquad [\![A \supset 1]\!] \overset{nil}{\sim} [\![1]\!] \qquad [\![1 \supset A]\!] \overset{g}{\sim} [\![A]\!] \qquad [\![\Box 1]\!] \overset{m_0}{\sim} [\![1]\!]$$
$$[\![A \wedge B]\!] \overset{c}{\sim} [\![B \wedge A]\!] \qquad [\![A \wedge (B \wedge C)]\!] \overset{a}{\sim} [\![(A \wedge B) \wedge C]\!] \qquad [\![(A \wedge B) \supset C]\!] \overset{\lambda}{\sim} [\![A \supset (B \supset C)]\!]$$

Proof. These arena isomorphisms are standard in game semantics and are all quite naturally defined. Recall that a function $f : X \to Y$ is a bijection iff there is a function $g : Y \to X$ such that $f(g(y)) = y$ and $g(f(x)) = x$ for all $x \in X$ and $y \in A$.
Consider the two functions $\lambda : V_{[\![(A \wedge B) \supset C]\!]} \to V_{[\![A \supset (B \supset C)]\!]}$ and $\lambda^{-1} : V_{[\![A \supset (B \supset C)]\!]} \to V_{[\![(A \wedge B) \supset C]\!]}$ defined by

$$\begin{array}{ccccccc} ((x,i),i) & \overset{\lambda}{\to} & (x,i) & \qquad & (x,i) & \overset{\lambda^{-1}}{\longrightarrow} & ((x,i),i) \\ ((x,i),j) & \overset{\lambda}{\to} & ((x,j),i) & \qquad & ((x,j),i) & \overset{\lambda^{-1}}{\longrightarrow} & ((x,i),j) \end{array}$$

Where x is a vertex in one between $V_{[\![A]\!]}$, $V_{[\![B]\!]}$ or $V_{[\![C]\!]}$ and i, j with $i \neq j$ are either 0 or 1. We clearly have that λ^{-1} is the inverse of λ. Moreover, both function trivially satisfies the condition on labels. The proof is concluded by remarking that the only new edges in $[\![A \supset (B \supset C)]\!]$ are the ones from the $\to$ roots of $[\![A]\!]$ and $[\![B]\!]$ to the $\to$-roots of $[\![C]\!]$ and that the same phenomena occurs in $[\![(A \wedge B) \supset C]\!]$.

$\Box$

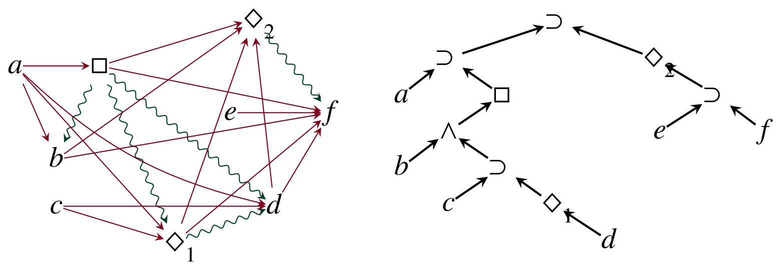
In the introduction we have specified that the notion of winning strategy that we are going to define requires a notion of batches i.e., given a formula A and a modality $\star$

1. a way to recognize and count the modalities such that A is in the scope of these modalities
2. and supposing that $A = \star B$ with $\star$ a modality, a way to recognize the formulas that were introduced in a proof by the same rule that introduced $\star B$, if any.

Definition 4.7. *Let $\mathcal{G} = [\![F]\!]$ be a modal arena and $v \in V_{\mathcal{G}}$. The* address *of v is the unique sequence of modal vertices $\mathsf{add}_v = m_1, \ldots, m_h$ in $V_{\mathcal{G}}$ which corresponds to the sequence of modalities in the path in the formula tree of F connecting the node of v to the root of F.*

If $\mathsf{add}_v = m_1, \ldots, m_h$, we denote by $h_v = |\mathsf{add}_v|$ the length of add_v and by $\mathsf{add}_v^k = m_k$ its k^{th} element.

Example 4.1. *Consider the modal arena and the formula tree of $(a \supset \Box(b \wedge (c \supset \Diamond_1 d))) \supset \Diamond_2(e \supset f)$, then*



$\mathsf{add}_a = \epsilon$
$\mathsf{add}_\Box = \Box$
$\mathsf{add}_b = \Box$
$\mathsf{add}_c = \Box$
$\mathsf{add}_\Diamond = \Diamond\Box$
$\mathsf{add}_d = \Diamond\Box$
$\mathsf{add}_{\Diamond'} = \Diamond'$
$\mathsf{add}_e = \Diamond'$
$\mathsf{add}_f = \Diamond'$

If $\mathcal{G}$ is a modal arena and $v \in V_{\mathcal{G}}$, we define $d(v)$ as the length of the $\rightarrow$-paths from v to a $\rightarrow$-root $w \in \vec{R}_{\mathcal{G}}$. Note that the property that all paths in a modal arena from a vertex to any root have the same length is not trivial, but the proof can be found in [131, Lemma 9]. The *parity of a vertex v* is the parity of $d(v)$, which can be either *even* or *odd*. We denote by $v^{\mathbf{O}}$ and $v^{\mathbf{P}}$ if the parity of v is respectively even or odd. As we will see in the following section, the players **O** and **P** can only play vertices of the corresponding parity, but the parity of the modalities in which the vertex belongs may not be the same as the parity of the move. By means of example, consider the atom a_2 in Figure 4.1 which is **O** but it is in the scope of two **P**-modalities.

4.4 Winning Strategies for CK

In this section we recall the definition of *winning innocent strategy* and we characterize the ones corresponding to correct CK-proofs.

4.4.1 Views

Winning innocent strategies are prefix-closed set of *views*. Views are a particular class of games in which both players can act in a very restricted manner: they are obliged to react, in a sense that we specify below, to the last move of the other player.

Definition 4.8. *Let F be a formula. A* move *is a vertex of* $[\![F]\!]$. *Let* $\mathsf{p} = \mathsf{p}_0 \cdots \mathsf{p}_n$ *be a sequence of distinct* moves *(we denote by* ϵ *the empty sequence). If v and w are two moves in* p*, we say that a vertex w* justifies *v whenever* $v \overset{\mathcal{G}}{\rightarrow} w$*. We call a move* p_i *in* p *a* **O**-move *or* **P**-move *if i is respectively even or odd. The games we will present are called* views *in the literature on game semantics. We conform to this particular nomenclature.*

We say that p *is a* view *in* $[\![F]\!]$ *if the following conditions are fulfilled:*

1. p *is a* play*: if* $\mathsf{p} \neq \epsilon$*, then* $\mathsf{p}_0 \in \overrightarrow{R}_{\mathcal{G}}$*;*
2. p *is* justified*: if* $i > 0$*, then* $\mathsf{p}i \rightarrow \mathsf{p}_{i-2k+1}$ *for a* $k \in \mathbb{N}$
3. p *is* **O**-shortsighted*: if* $\mathsf{p}^{\mathbf{O}}_{i+1}$ *and* $\mathsf{p}^{\mathbf{P}}_{i}$*, then* $\mathsf{p}_{i+1} \rightarrow \mathsf{p}_i$*;*
4. p *is* **P**-uniform*: if* $\mathsf{p}^{\mathbf{P}}_{i+1}$ *and* $\mathsf{p}^{\mathbf{O}}_{i}$*, then* $\ell(\mathsf{p}_{i+1}) = \ell(\mathsf{p}_i)$*.*
5. p *is* modal*:* $\ell(\mathsf{p}_i) \in \mathcal{A} \cup \{\Diamond\}$*.*

Moreover, if p *is a view, we say that*

6. p *is* well-batched*:* $|\mathsf{add}_{\mathsf{p}_{2k}}| = |\mathsf{add}_{\mathsf{p}_{2k+1}}|$ *for every* $2k \in \{0, \ldots, n-1\}$*.*

Let us make some comments on this, otherwise obscure, definition. The first condition impose that the first move of a view in an arena $[\![F]\!]$ is a root of the Arena. Since a root has always **O**-polarity this is another way of saying that the first move of a view is a **O**-move i.e., the **O**-player starts the game. Condition 2 imposes that each move in a game, but the first, is justified by a move that already —up to that point— appears in the game. Moreover, if v is justified by w then there is an even number of moves (possibly zero) between the two moves. Thus condition 2 together with condition 1 implies that a view is an alternated sequence of **O**-moves and **P**-moves i.e., the two players plays by taking turns. In fact : suppose that p is a sequence of vertices of a modal arena respecting the two aforementioned conditions. Suppose moreover that for all $i < j$ p_i **O**-vertex of the arena iff i is even, and it is a **P**-vertex if i is odd. The move that justifies p_{j+1} is an **O**-move is $j+1$ is odd, and it is an **P**-move is $j+1$ is even, we can thus conclude. Condition 3 impose a restriction on the moves that the **O**-player can play. Each of his moves is justified by the immediately preceding move of the other player. Condition 4 puts a restriction on the set of moves that the **P**-player can play. Each move that she plays must be a vertex labeled by the same symbol that labels the vertex of the immediately preceding move of the other player. Condition 5 says that every move in a view is labeled by a propositional variable or by a $\diamond$ modality. Finally, condition 6 imposes another restriction to the moves that are played in a view by **P**. If she plays a move v, then the move must be in the scope of the same number of modalities as the immediately preceding move of the other player.

4.4.2 Winning innocent strategies

We can now define strategies. As in the preceding chapter about dialogical logic, a strategy will be formalized as a tree of views. The tree can branch only after a move of **P** and it will be winning whenever maximal (with respect to the prefix order on sequences) views ends in a **P**-move.

The *predecessor* of a non-empty view p is the sequence obtained by removing the last move in p. The *successor* is the converse relation.

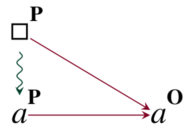
Definition 4.9. *A* winning innocent strategy *(or* **WIS** *for short) for F (or over $[\![F]\!]$) is a finite non-empty set $\mathcal{S}$ of views in $[\![F]\!]$ such that:*

1. *$\mathcal{S}$ is predecessor-closed: if $\mathsf{p} \cdot v \in \mathcal{S}$ then $\mathsf{p} \in \mathcal{S}$;*
2. *$\mathcal{S}$ is* **O**-complete*: if $\mathsf{p} \in \mathcal{S}$ has even length, then every successor of p is in $\mathcal{S}$;*
3. *$\mathcal{S}$ is* **P**-deterministic *and* **P**-total*: if $\mathsf{p} \in \mathcal{S}$ has odd length, then exactly one successor of p is in $\mathcal{S}$.*

We say that a **WIS** *$\mathcal{S}$ is* trivial *if $\mathcal{S} = \{\epsilon\}$ and it is* well-batched *if all its views are. If $\mathcal{S}$ is a winning innocent strategy over $[\![F]\!]$ we will write $\mathcal{S} : [\![F]\!]$.*

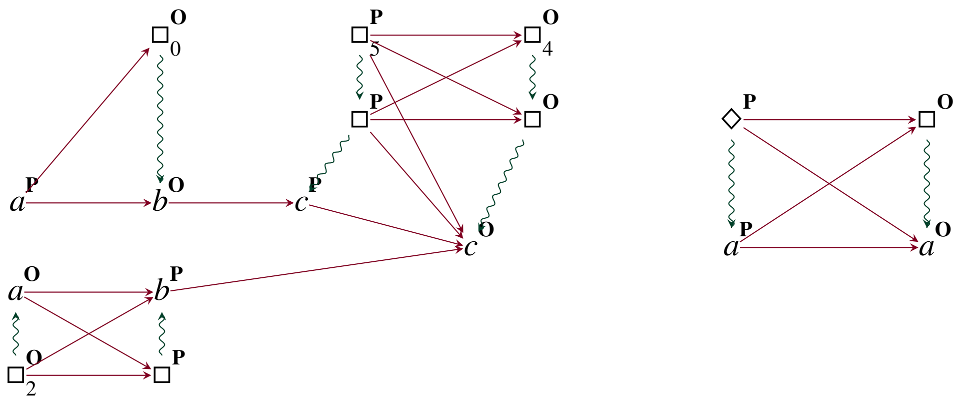
Remark 4.1. *Note that our definition of* **WIS** *on arenas of modality-free formulas is the same of the one given in [34, 113, 131] where the modal condition trivially holds. Remark that in a modal arena, a view can only have a finite number of successors. Thus,* **WIS** *are finitely branching trees. Since each branch is finite we can conclude by König lemma that every* **WIS** *is finite. If $\mathcal{G}$ is a non-empty modal arena, then a* **WIS** *$\mathcal{S}$ on $\mathcal{G}$ must contain all views of the form v with $v \in \overrightarrow{R}_{\mathcal{G}}$, that is, $\mathcal{S}$ is non-trivial.*

Example 4.2. *Consider the formula $(\Box a) \supset a$ where a is a predicate variable. This formula is not provable in* LCK*. The arena of $(\Box a) \supset a$ can be represented as*



The only views over this arena are the empty view ϵ and the two views $\epsilon a^{\mathbf{O}}$ and $\epsilon a^{\mathbf{O}} a^{\mathbf{P}}$. This latter view is not well batched because the address of $a^{\mathbf{O}}$ is $\Box^{\mathbf{O}}$ and $a^{\mathbf{P}}$ has no address. Thus, there are no well-batched **WIS** *over $(\Box a) \supset a$.*

Unfortunately there are well-batched **WIS** *over formulas that are* not *provable in the sequent system* LCK*. Consider the formulas $F = (\Box a \supset \Box b) \supset ((\Box(a \supset b) \supset c) \supset c)$ and $G = \Diamond a \supset \Box a$ whose arenas can be represented as*



where in the arena of F we indexed some occurrences of the $\Box$ modality to speak clearly about them. The maximal, with respect to the prefix order, views in these two arenas are respectively

$$\mathsf{p} = c^{\mathbf{O}} c^{\mathbf{P}} b^{\mathbf{O}} b^{\mathbf{P}} a^{\mathbf{O}} b^{\mathbf{P}} \qquad \sigma = a^{\mathbf{O}} a^{\mathbf{P}}$$

Both p *and* σ *are well-batched views: each* **P***-move in them have an address of the same size of the address of the immediately preceding* **O***-move. In* p $c^{\mathbf{O}} c^{\mathbf{P}}$ *have both an address of size 2.* $\mathsf{add}_{b^{\mathbf{O}}} = \square_0^{\mathbf{O}}$ *and* $\mathsf{add}_{b^{\mathbf{P}}} = \square^{\mathbf{P}}$*. Finally* $\mathsf{add}_{a^{\mathbf{O}}} = \square_2^{\mathbf{O}}$ *and* $\mathsf{add}_{a^{\mathbf{P}}} = \square_0^{\mathbf{O}}$*. In* ρ $\mathsf{add}_{a^{\mathbf{O}}} = \square^{\mathbf{O}}$ *and* $\mathsf{add}_{a^{\mathbf{P}}} = \Diamond^{\mathbf{P}}$*. Thus the set of views containing* p *and any of its prefixes as well as the set of view containing* ρ *and any of its prefixes are* **WIS***s for F and G respectively. This means that well-batched* **WIS***s do not capture theorem-hood in* CK*.*

4.4.3 CK Winning Innocent Strategies

As pointed out in the introduction, in order to characterize the winning strategies corresponding to proofs in LCK we need some additional definitions. The machine we invented is based on the following insight. In an LCK theorem, the modality are partitioned. The partitions of the modality are induced by the introduction rules of the modality themselves. If we decorate the formulas in the rules with their respective polarities, **P** for the negative polarity and **O** for the positive polarity, the two modal rules have the following form

$$\frac{A_1^{\mathbf{P}}, \ldots A_n^{\mathbf{P}} \vdash C^{\mathbf{O}}}{(\square_1 A_1)^{\mathbf{P}}, \ldots (\square_n A_n)^{\mathbf{P}} \vdash (\square C)^{\mathbf{O}}}\,\mathsf{K}^{\square} \qquad \frac{B_1^{\mathbf{P}}, \ldots B_m^{\mathbf{P}}, D^{\mathbf{P}} \vdash F^{\mathbf{O}}}{(\square_1 B_1)^{\mathbf{P}}, \ldots, (\square_n B_n)^{\mathbf{P}}, (\Diamond D)^{\mathbf{P}} \vdash (\Diamond F)^{\mathbf{O}}}\,\mathsf{K}^{\Diamond}$$

each rule 'introduces' exactly one positive modality that can be either an $\Diamond$ or a $\square$ depending on the rule. Moreover, if the rule is a $\mathsf{K}^{\square}$-rule then it introduces also $n \geq 0$ negative boxes while the $\mathsf{K}^{\square}$ introduces *exactly* one negative diamond and $n \geq 0$ negative boxes. Each of the aforementioned partitions contains the modality that are 'introduced' in the proof of the formula by the same instance of a modal rule. We will define the partitions on modalities of a formula using the closure of a relation defined over elements of addresses —addresses are sequence of modalities— of consecutive moves. The element of the addresses must stands at the same depth in addresses of moves. To precisely define this notion we first need the following technical definition: the addresses of a view can have different sizes, i.e., two different addresses can have a different number of elements. We need to line up somehow addresses of different size

Definition 4.10. *Let* $\mathsf{p} = \mathsf{p}_0 \cdots \mathsf{p}_{n-1}$ *be a well-batched view on a modal arena* $\mathcal{G}$. *We write* $h_{\mathsf{p}} = \max\{h_v \mid v \in \mathsf{p}\}$ *and we define the* batched view *of* p *as the* $h_{\mathsf{p}} \times n$ *matrix* $\mathcal{F}(\mathsf{p}) = (\mathcal{F}(\mathsf{p})_0, \ldots, \mathcal{F}(\mathsf{p})_n)$ *with elements in* $V_{\mathcal{G}} \cup \{\epsilon\}$ *such that each column* $\mathcal{F}(\mathsf{p})_i$ *is defined as follows:*

$$\mathcal{F}(\mathsf{p})_i = \begin{pmatrix} \mathcal{F}(\mathsf{p})_i^{h_{\mathsf{p}}} & = & \mathsf{add}_{\mathsf{p}_i}^{h_{\mathsf{p}_i}} \\ & \vdots & \\ \mathcal{F}(\mathsf{p})_i^{h_i+1} & = & \mathsf{add}_{\mathsf{p}_i}^{1} \\ \mathcal{F}(\mathsf{p})_i^{h_i} & = & \epsilon \\ & \vdots & \\ \mathcal{F}(\mathsf{p})_i^{1} & = & \epsilon \\ \mathcal{F}(\mathsf{p})_i^{0} & = & \mathsf{p}_i \end{pmatrix}$$

where a $h_i \in \{0, \ldots, h_{\mathsf{p}}\}$ *defined for each* $i \in \{0, \ldots, n\}$.

Each view induces an equivalence relation $\overset{\mathcal{G}_{\mathsf{p}}}{\sim}$ *over* $V_{\mathcal{G}}$ *generated by the transitive, symmetric and reflexive closure of the following relation:*

$$u \overset{\mathcal{G}_{\mathsf{p}}}{\sim}_1 w \quad \text{iff} \quad \begin{array}{l} u = \mathcal{F}(\mathsf{p})_{2k}^{h} \text{ and } w = \mathcal{F}(\mathsf{p})_{2k+1}^{h} \\ \text{for a } 2k < n-1 \text{ and a } h \leq h_{\mathsf{p}} \end{array}$$

Example 4.3. *The batched view of the view* p *of example 4.2 is*

$$\begin{array}{cccccc} \Box_4^{\mathbf{O}} & \Box_5^{\mathbf{P}} & \Box_0^{\mathbf{O}} & \Box^{\mathbf{P}} & \Box_2^{\mathbf{O}} & \Box_0^{\mathbf{O}} \\ \Box^{\mathbf{O}} & \Box^{\mathbf{P}} & \epsilon & \epsilon & \epsilon & \epsilon \\ c^{\mathbf{O}} & c^{\mathbf{P}} & b^{\mathbf{O}} & b^{\mathbf{P}} & a^{\mathbf{O}} & a^{\mathbf{P}} \end{array}$$

We have the following equivalence classes on this view $E_1 = \{\Box_4^{\mathbf{O}}, \Box_5^{\mathbf{P}}\}$, $E_2 = \{\Box_0^{\mathbf{O}}, \Box^{\mathbf{P}}, \Box_2^{\mathbf{O}}\}$ *and* $E_3 = \{\Box^{\mathbf{O}}, \Box^{\mathbf{P}}\}$. *Remark that* E_2 *contains two positive modalities. This is because* $\Box_0^{\mathbf{O}} \overset{\mathsf{p}}{\sim}_1 \Box_2^{\mathbf{P}}$ *since* $\Box_0^{\mathbf{O}} = \mathsf{add}_{c^{\mathbf{P}}}^2$ *and* $\Box_2^{\mathbf{O}} = \mathsf{add}_{c^{\mathbf{O}}}^2$, *and because* $\Box_0^{\mathbf{O}} \overset{\mathsf{p}}{\sim} \Box_0^{\mathbf{O}}$ *by reflexive closure. We want to exclude strategies in which equivalence classes generated by the* $\sim$ *relation contains more than one positive modality. Moreover, if the positive modality is* $\Box$, *then we want the class to contain only negative boxes. On the contrary, if the positive modality is a* $\Diamond$ *we want the class to contain exactly one negative* $\Diamond$ *and* $n \geq 0$ *negative boxes. We formally define this requirement as follows*

Definition 4.11. *Let* $\mathcal{S}$ *be a well-batched strategy on a modal arena* $\mathcal{G}$. *We say that* $\mathcal{S}$ linked *if for every* $\mathsf{p} \in \mathcal{S}$ *the* $\overset{\mathcal{G}_{\mathsf{p}}}{\sim}$*-classes are of the shape* $\{v_1^{\mathbf{P}}, \ldots, v_n^{\mathbf{P}}, w^{\mathbf{O}}\}$. *This induces the edge-relation* $u \overset{\mathcal{G}_{\mathsf{p}}}{\rightharpoonup} w$ *iff* $u^{\mathbf{P}} \overset{\mathcal{G}_{\mathsf{p}}}{\sim} w^{\mathbf{O}}$.

We say that $\mathcal{S}$ is CK-batched *if each modal vertex $w^{\mathbf{O}}$ occurring in the address of a move in $\mathcal{S}$ the following conditions are fulfilled:*

1. *if $w^{\mathbf{O}} \in V^{\square}_{\mathcal{G}}$ and $v \xrightarrow{\mathcal{G}_{\mathsf{p}}} w$ for a $\mathsf{p} \in \mathcal{S}$, then $v \in V^{\square}_{\mathcal{G}}$*

2. *if $w^{\mathbf{O}} \in V^{\diamond}_{\mathcal{G}}$, then there is a unique $u \in V^{\diamond}_{\mathcal{G}}$ in the set $\{v \in V_{\mathcal{G}} \mid v \xrightarrow{\mathcal{G}_{\mathsf{p}}} w$ for a $\mathsf{p} \in \mathcal{S}\}$.*

We call a CK*-batched* **WIS** *a* CK-winning innocent strategy *(*CK-**WIS** *for short).*

Remark that the view p of example 4.2 cannot belong to a CK-**WIS**. As we have saw in example 3 there is an equivalence class generated by the relation $\overset{\mathsf{p}}{\sim}$ on the addresses of p that contains two positive modalities.

4.5 Compositionality of Winning Strategies

In this section we prove that CK-**WIS** composes i.e., that given a CK-**WIS** $\mathcal{T}$ for $A \supset B$ and a CK-**WIS** $\mathcal{R}$ for $B \supset C$ we can construct a CK-**WIS** for $A \supset C$. This latter strategy on $A \supset C$ will be obtained by playing as $\mathcal{T}$ on A and as $\mathcal{R}$ on C and using the moves in the two occurrences of B as a gluing of the two strategies.

In order to simplify the presentation of our compositionality result, we propose a slightly different approach to the proof of winning strategy's compositionality with respect to the one normally used in the literature, e.g. [77, 100], where proofs are given by reasoning on specific sequences[4] over the arena $[\![A \supset (B \supset C)]\!] \overset{\mathsf{f}}{\sim} [\![A, B \vdash C]\!]$, such that these views can be projected on views over the arenas of $A \vdash B$ and $B \vdash C$. Instead, we here reason directly over the views of a winning strategy over the arena $[\![A, B_1 \supset B_2 \vdash C]\!]$. This allows us to preserve the parities of vertices when performing the projections.

To obtain an intuition behind the idea, consider the additional rule hide removing a formula of the shape $B \supset B$ occurring in the left-hand side of a sequent in order to simulate the cut as shown below.

$$\dfrac{\Gamma \vdash B \quad \Delta, B \vdash C}{\Gamma, \Delta \vdash C}\,\mathsf{cut} \quad \rightsquigarrow \quad \dfrac{\dfrac{\Gamma \vdash B \quad B, \Delta \vdash C}{\Gamma, \Delta, B \supset B \vdash C}\,\supset^{\mathsf{L}}}{\Gamma, \Delta \vdash C}\,\mathsf{hide}$$

[4] Note that these sequences are not views.

This approach complies with the slogan "interaction + hide" often mentioned in the literature, e.g., [2, 100]. Here the interaction is represented by the $\supset^{\mathsf{L}}$-rule, while the hiding is performed by erasing the formula $B \supset B$ using the hide-rule.

In terms of views, our interaction is defined by composing views from the two corresponding strategies by "gluing" them using a *copycat* strategy[5] on the cut-formula while the hiding consist of ignoring the moves in the hidden formulas.

Notation 4.1. *If Δ is a list (of occurrences) of formulas in $\Gamma \vdash A$ and p is a sequence of moves in $[\![\Gamma \vdash A]\!]$, we denote by $\mathsf{p}|_\Delta$ the* projection *of p on Δ, that is, the sequence obtained by erasing from p any move not in Δ. By means of example, if $A = a \supset e$, $B = b \wedge d$ and $C = c$, then $baadcebda|_{A,C} = aacea$.*

Whenever we consider two distinct occurrences B_1 and B_2 of the same formula B, we assume $\cdot^\perp$ to be the bijection between the vertices in $V_{[\![B_1]\!]}$ and in $V_{[\![B_2]\!]}$ corresponding to the same atom/modality in B.

Definition 4.12. *Let $\mathcal{T}$ and $\mathcal{R}$ be CK-**WIS**s respectively for $A \vdash B_1$ and $B_2 \vdash C$ such that B_1 and B_2 are occurrences of the same formula B, and let $\tau \in \mathcal{T}$ and $\rho \in \mathcal{R}$.*

We define the interaction *of τ ad ρ over B as the sequence of moves $\sigma = \tau \overset{B}{\bullet} \rho$ over $[\![A, B_1 \supset B_2 \vdash C]\!]$ following ρ (resp. τ) until a **P**-move b in B_2 (resp. B_1) is reached; then it switches to the corresponding **O**-move $b^\perp$ in τ (resp. ρ), if it exists. That is,*

$$\sigma_0 = \rho_0 \quad \text{and} \quad \sigma_{i+1} = \begin{cases} \tau_{k+1} & \text{where } \sigma_i = \tau_k \text{ is a move in } A \text{ or a } \mathbf{O}\text{-move in } B_1 \\ \rho_{k+1} & \text{where } \sigma_i = \rho_k \text{ is a move in } C \text{ or a } \mathbf{O}\text{-move in } B_2 \\ b^\perp & \text{where } \sigma_i = b \text{ is a } \mathbf{P}\text{-move in } B_1 \text{ and } b^\perp \text{ occurs in } \rho \\ b^\perp & \text{where } \sigma_i = b \text{ is a } \mathbf{P}\text{-move in } B_2 \text{ and } b^\perp \text{ occurs in } \tau \\ \text{not defined} & \text{otherwise} \end{cases}$$

We define the composition *$\tau \overset{B}{\circ} \rho$ of τ and ρ over B as the projection of $\tau \overset{B}{\bullet} \rho$ over A and C, that is, $\tau \overset{B}{\circ} \rho = (\tau \overset{B}{\bullet} \rho)|_{A,C}$. We define the* composition *of $\mathcal{T}$ and $\mathcal{R}$ over B as the following set of sequences over $[\![A \vdash C]\!]$*

$$\mathcal{T} \overset{B}{\circ} \mathcal{R} = \{\tau \overset{B}{\circ} \rho \mid \tau \in \mathcal{T}, \rho \in \mathcal{R}\}$$

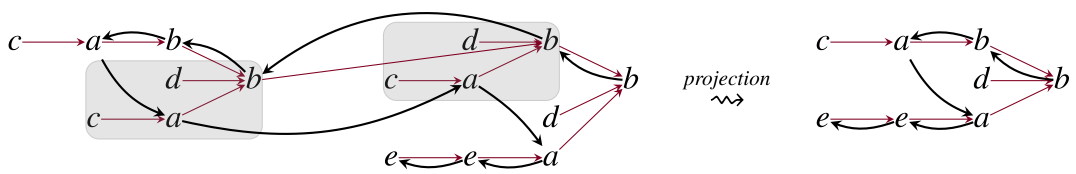
Intuitively, when defining the interaction $\bullet$, the player **O** changes her way of play: whenever the player **P** plays a move b in B_1 (or B_2), its successive **O**-move is the corresponding $b^\perp$ in B_2 (resp. B_1) instead of playing according to **O**-shortsightedness. By

[5] the copy-cut strategy is precisely defined in section 4.6 of the present chapter.

definition $(\tau \overset{B}{\LEFTcircle} \rho)|_{A,B_1} = \tau$ and $(\tau \overset{B}{\LEFTcircle} \rho)|_{B_2,C} = \rho$, hence, $(\tau \overset{B}{\LEFTcircle} \rho)$ is always finite. The rest of this section is devoted to prove that if $\mathcal{T}$ and $\mathcal{R}$ are CK-**WIS**s, then also is $\mathcal{T} \overset{B}{\circ} \mathcal{R}$.

Example 4.4. *Consider the sequents* $A \vdash B_1 = (c \supset a) \supset b \vdash (d \wedge (c \supset a)) \supset b$ *and* $B_2 \vdash C = (d \wedge (c \supset a)) \supset b \vdash (d \wedge ((e \supset e) \supset a)) \supset b$ *and the view* $\tau = bbaacc$ *on* $[\![A \vdash B_1]\!]$ *and the views* $\rho_1 = bbaaee$ *and* $\rho_2 = bbdd$ *on* $[\![B_2 \vdash C]\!]$*. Note that these views are the unique maximal views in the unique* **WIS**s *for these sequents. Then we can picture the construction of* $\tau \overset{B}{\circ} \rho_1$ *as follows, where on the left-hand side we highlight the two occurrences of* $[\![B]\!]$ *on which the views interact, and the black arrows identify the sequences of moves on the arenas.*

$\tau \overset{B}{\LEFTcircle} \rho_1 = bbbbaaaaee$ $\qquad\qquad$ $\tau \overset{B}{\circ} \rho_1 = bbaaee$



Similarly $\tau \overset{B}{\LEFTcircle} \rho_2 = bbbbaa$ *and* $\tau \overset{B}{\circ} \rho_2 = bba$*. Note that in this case the definition of* $\tau \overset{B}{\LEFTcircle} \rho_2$ *stops because the successive should be* $a^{\perp}$ *but it does not occur in* ρ*.*

Remark 4.2. *If* A, B_1, B_2 *and* C *are formulas with* B_1 *and* B_2 *occurrences of the same formula* B*, then atoms and modalities in these formulas have the same parity in* $[\![A, B_1 \supset B_2 \vdash C]\!]$ *and in* $[\![A \vdash B_1]\!]$ *and* $[\![B_2 \vdash C]\!]$*.*

Our definitions allow us to show that the composition of well-batched **WIS**s is a well-batched **WIS**.

Lemma 4.2. *Let* $\mathcal{T}$ *and* $\mathcal{R}$ *be well-batched* **WIS** *for respectively* $A \vdash B_1$ *and* $B_2 \vdash C$ *such that* $B = B_1 = B_2$*. Then* $\mathcal{S} = \mathcal{T} \overset{B}{\circ} \mathcal{R}$ *is a well-batched* **WIS** *for* $A \vdash C$*.*

Proof. We first prove that for each $\tau \in \mathcal{T}$ and $\rho \in \mathcal{R}$ such that $\tau \overset{B}{\equiv} \rho$ we have that $\tau \overset{B}{\circ} \rho$ is a well-batched view over $[\![A \vdash C]\!]$ since it verifies all conditions in Definition 4.8. For any $\sigma = \tau \overset{B}{\circ} \rho$ we have that

1. σ is a play: since $\sigma_0 \in \vec{R}_{[\![B \vdash C]\!]}$ and $\vec{R}_{[\![B \vdash C]\!]} = \vec{R}_{[\![C]\!]} = \vec{R}_{[\![A \vdash C]\!]}$.

2. σ is justified: if a move in $[\![A]\!]$ is justified in τ by a move in $[\![A]\!]$ or if a move in $[\![C]\!]$ is justified in ρ by a move in $[\![C]\!]$, then we can conclude. By definition of $[\![B \vdash C]\!]$ no move in $[\![C]\!]$ can be justified ρ by a moved in B. We conclude by remarking that if a move in $[\![A]\!]$ is justified in τ by a move in $[\![B]\!]$, then this move must be a root of $[\![A]\!]$, and then $v \overset{[\![A\vdash C]\!]}{\rightarrow} \sigma_0$ since $u \overset{[\![A\vdash C]\!]}{\rightarrow} w$ for all $u \in \vec{R}\ [\![A]\!]$ and $w \in \vec{R}\ [\![C]\!]$.

3. σ is **O**-shortsighted: by definition of τ and ρ we must have that both σ_{2k} and σ_{2k+1} are either in $[\![A]\!]$ or in $[\![C]\!]$. We conclude by the hypothesis on σ and τ.

4. σ is **P**-uniform: by induction using the **P**-uniformity of τ and ρ and the fact that $\ell(v) = \ell(v^{\perp})$.

5. σ is modal: follows by the fact that no move in τ or ρ is a $\square$-vertex.

6. σ is well-batched: it suffices to remark that if $\mathsf{add}_v = m_1 \cdots m_k$, then $\mathsf{add}_{v^{\perp}} = m_1^{\perp} \cdots m_k^{\perp}$. We can conclude similarly to the proof of **P**-uniformity since in $\tau \overset{B}{\bullet} \rho$ in all moves in a subsequence in B_1 and B_2 have constant height.

To conclude, we show that that $\mathcal{S}$ is

1. predecessor-closed: it follows by the fact that $\mathcal{T} \overset{B}{\bullet} \mathcal{R} = \{\tau \overset{B}{\bullet} \rho \mid \tau \in \mathcal{T}, \rho \in \mathcal{R} \text{ and } \tau \overset{B}{\equiv} \rho\}$ is predecessor closed

2. **O**-complete: if $\sigma v^{\mathbf{P}} \in \mathcal{S}$ then $v^{\mathbf{P}}$ appears in a view $\tau \in \mathcal{T}$ or in a view $\rho \in \mathcal{R}$ as an **P**-move. We conclude by the definition of the composition $\circ$ and by the fact that $\mathcal{S}$ and $\mathcal{R}$ are **WIS**.

3. **P**-deterministic and **P**-total: each view $\sigma \in \mathcal{S}$ is of the form $\sigma = \tau \overset{B}{\circ} \rho = (\tau \overset{B}{\bullet} \rho)|_{A,C}$ for a $\tau \in \mathcal{T}$ and $\rho \in \mathcal{R}$. By induction on the length of $\tau \overset{B}{\bullet} \rho$ we can prove that $v^{\mathbf{O}} \in \tau \overset{B}{\bullet} \rho$ is followed by a unique **P**-move since $\mathcal{T}$ and $\mathcal{R}$ are **P**-deterministic and **P**-total and each $v^{\perp} \in [\![B_1]\!]$ and $w^{\perp} \in [\![B_2]\!]$ is uniquely determined respectively by a $v \in [\![B_2]\!]$ and a $w \in [\![B_1]\!]$. To conclude we must prove that whenever $s \in \tau \overset{B}{\bullet} \rho$ and

$$s = s' x^{\mathbf{O}} y^{\mathbf{P}} y^{\perp} y z_1, \dots z_n w^{\mathbf{P}} w^{\perp}$$

With $x^{\mathbf{O}} \in [\![C]\!]$ (resp. $x^{\mathbf{O}} \in [\![A]\!]$), $y^{\mathbf{P}} \in [\![B_1]\!]$ (resp. $y^{\mathbf{P}} \in [\![B_2]\!]$) $z_1 \dots z_n$ a (possibly empty) sequence of moves in $[\![A]\!]$ (resp. in $[\![C]\!]$) and $w^{\mathbf{P}} \in [\![B_2]\!]$ (resp. $w^{\mathbf{P}} \in [\![B_1]\!]$)

then there is view $\rho' \in \mathcal{R}$ (resp. $\tau' \in \mathcal{T}$) such that $\tau \overset{B}{\circ} \rho' = s'x^{\mathbf{O}}y^{\mathbf{P}}y^{\perp}yz_1, \dots z_n w^{\mathbf{P}} w^{\perp} z^{\mathbf{P}}$ (resp. $\tau' \overset{B}{\circ} \rho = s'x^{\mathbf{O}}y^{\mathbf{P}}y^{\perp}yz_1, \dots z_n w^{\mathbf{P}} w^{\perp} z^{\mathbf{P}}$) for a certain $z \in [\![B_1 \supset C]\!]$ (resp in $[\![A \supset B_1]\!]$). Suppose that $x^{\mathbf{O}} \in [\![C]\!]$ and $y^{\mathbf{P}} \in [\![B_1]\!]$. The move $w^{\mathbf{P}}$ is in $[\![B_2]\!]$ and thus, by the definition of arena, we must have that $w^{\mathbf{P}} \rightarrow y^{\perp}$. This means, in particular that $w^{\perp}$ is an even vertex in $[\![B_1]\!]$ and that $w^{\perp} \rightarrow y^{\mathbf{P}}$ since $s'x^{\mathbf{O}}y^{\mathbf{P}}|_{C,B_1} = \rho$ is a view in $\mathcal{R}$ by **O**-completeness the view $\sigma w^{\perp} \in \mathcal{R}$, we thus conclude by de totality of $\sigma w^{\perp}$

□

To prove that the composition also preserves CK-framing, we use the following remark and a lemma assuring that each $\sim$-class defined by a well-batched view contains at least one vertex in $V^{\mathbf{O}}$.

Remark 4.3. *If $v^{\mathbf{O}}$, $w^{\mathbf{P}}$, $m^{\mathbf{P}}$ are vertices of a modal arena $\mathcal{G}$ such that $m = \mathsf{add}_v^k$ and $v \rightarrow w$, then $m = \mathsf{add}_w^k$.*

Lemma 4.3. *Let* p *be a view in a well-batched* **WIS** *$\mathcal{S}$ over a modal arena. Then the corresponding $\overset{\mathsf{p}}{\sim}$-classes contain at least one even vertex $v \in V^{\mathbf{O}}$.*

Proof. By definition, a $\overset{\mathsf{p}}{\sim}$-class containing an atomic vertex contains exactly one vertex $v = \mathsf{p}_{2k+1} \in V^{\mathbf{P}}$ and one $w = \mathsf{p}_{2k} \in V^{\mathbf{O}}$.

Otherwise, we can prove that given any vertex $v = \mathsf{add}_{\mathsf{p}_i}^k \in V^{\mathbf{P}}$, then there is a $j < i$ such that $w^{\mathbf{O}} = \mathsf{add}_{\mathsf{p}_j}^k \in V^{\mathbf{O}}$ is in the same $\overset{\mathsf{p}}{\sim}$-class.

In fact, if i it is even, then by Remark 4.3 we have $\mathsf{add}_{\mathsf{p}_i}^k = \mathsf{add}_{\mathsf{p}_{i-1}}^k$. Since for any $r \in \vec{R}$ and $v \in \mathsf{add}_r$ we have $v \in V^{\mathbf{O}}$, then $i - 1 > 0$. If i is odd, then $\mathsf{add}_{\mathsf{p}_i}^k \overset{\mathsf{p}}{\sim} \mathsf{add}_{\mathsf{p}_{i-1}}^k$ and we can repeat the previous argument on $i - 1$. If $j < i$ does not exist, then p should have an infinite prefix. Absurd. □

Let $\mathcal{T}$ a CK-**WIS** for $A \supset B$ and $\mathcal{T}$ be a CK-**WIS** for $B \supset C$. The two following technical lemmas are needed to assure that for each $\mathsf{p} \in \mathcal{S} = \mathcal{T} \overset{B}{\circ} \mathcal{R}$ the corresponding $\overset{\mathsf{p}}{\sim}$-classes contains *exactly* one vertex $v \in V^{\mathbf{O}}$.

Lemma 4.4. *Let $\mathcal{T}$ and $\mathcal{R}$ be two* CK-**WIS** *for, respectively, $A \supset B_2$ and $B_2 \supset C$ with B_1 and B_2 occurrences of the same formula B. Let s be a sequence such that $s \in \mathcal{T} \overset{B}{\circ} \mathcal{R}$ and*

$$s = s'x^{\mathbf{O}}b_1 b_1^{\perp} b_2^{\perp} b_2 b_3 \dots b_n b_n^{\perp} y^{\mathbf{P}} s''$$

with $b_i \in [\![B_k]\!]$ and $b_i^{\perp} \in [\![B_j]\!]$ for $i \in \{1, \dots n\}$, $k \neq j$ and $k, j \in \{1, 2\}$.

1. *if* $y^{\mathbf{P}} \in [\![A]\!]$ *and* $\mathsf{add}^k_x \in V^{\mathbf{O}}_{[\![A]\!]}$ *for some* $k \in \mathbb{N}$*, then* $\mathsf{add}^k_{b_i^\perp} \in V^{\mathbf{P}}_{[\![B_2]\!]}$ *for all* $i \in \{1, \dots n\}$
2. *if* $y^{\mathbf{P}} \in [\![C]\!]$ *and* $\mathsf{add}^k_x \in V^{\mathbf{O}}_{[\![C]\!]}$ *for some* $k \in \mathbb{N}$*, then* $\mathsf{add}^k_{b_i^\perp} \in V^{\mathbf{P}}_{[\![B_1]\!]}$ *for all* $i \in \{1, \dots n\}$

Proof. We prove 1 the proof of 2 being entirely similar. Remark that

$$s|A, B_2 = s'|_{A,B_2} b_1^\perp b_2^\perp, \dots b_n^\perp x^{\mathbf{P}} s''|_{A,B_2}$$

is a view τ in $\mathcal{T}$. It follows that $\mathsf{add}^k_{x^{\mathbf{P}}} \overset{\tau}{\sim} \mathsf{add}^k_{b_n^\perp}$ by the definition of the $\sim$-relation. The fact that $\mathsf{add}^k_{x^{\mathbf{P}}} \overset{\tau}{\sim} \mathsf{add}^k_{b_n^\perp}$ and that $\mathsf{add}^k_{x^{\mathbf{P}}} \in V^{\mathbf{O}}_A$ implies that $\mathsf{add}^k_{b_n^\perp} \in V^{\mathbf{P}}_{B_2}$ otherwise there would be a $\overset{\tau}{\sim}$-class of equivalence containing two distinct even modalities contradicting the hypothesis that $\mathcal{T}$ is a CK-**WIS**. Let $j > 1$ and suppose that, for all $r > k$, $r \in \{j+1, \dots n\}$ we have that $\mathsf{add}^k_{b_r^\perp} \in V^{\mathbf{P}}_{[\![B_2]\!]}$ and $\mathsf{add}^k_{b_r^\perp} \overset{\tau}{\sim} \mathsf{add}^k_{x^{\mathbf{P}}}$.

- If $b_{j+1}^\perp \in V^{\mathbf{P}}_{[\![B_2]\!]}$ then $\mathsf{add}^k_{b_{j+1}^\perp} \overset{\tau}{\sim} \mathsf{add}^k_{b_j^\perp}$. This force $\mathsf{add}^k_{b_j}$ to belong to $V^{\mathbf{P}}_{B_2}$. Otherwise, since $\mathsf{add}^k_{b_{j+1}^\perp} \overset{\tau}{\sim} \mathsf{add}^k_{x^{\mathbf{O}}}$ there would be a $\overset{\tau}{\sim}$-class containing two distinct even modal vertices.
- If $b_{j+1}^\perp \in V^{\mathbf{O}}_{[\![B_2]\!]}$ since $\mathsf{add}^k_{b_{j+1}^\perp} \in V^{\mathbf{P}}_{B_2}$ and $b_{j+1}^\perp \overset{[\![B_2]\!]}{\rightarrow} b_j^\perp$ we conclude, by remark 4.3 that $\mathsf{add}^k_{b_{j+1}^\perp} = \mathsf{add}^k_{b_j^\perp}$ and thus that $\mathsf{add}^k_{b_{j+1}^\perp} \overset{\tau}{\sim} \mathsf{add}^k_{x^{\mathbf{P}}}$

□

Let us state and prove another technical lemma

Lemma 4.5. *Let $\mathcal{T}$ and $\mathcal{R}$ be two* CK-**WIS** *for respectively $A \supset B_2$ and $B_1 \supset C$ with B_1 and B_2 two occurrences of the same formula B. Let $s = \tau \overset{B}{\bullet} \rho \in \mathcal{T} \overset{B}{\bullet} \mathcal{R}$ and call* p *the projection $s|_{A,C}$ of s to A and C. Let $m = \mathsf{add}^k_x$ for a certain $k \in \mathbb{N}$ and $x \in s$ and suppose $x \in \mathsf{p}$. if there is $n = \mathsf{add}^k_y$ with $y \in s$ such that $m \overset{\mathsf{p}}{\sim} n$ then there is $w = \mathsf{add}^k_z$ with $z \in s$ such that either $m \overset{\tau}{\sim} w$ or $m \overset{\rho}{\sim} n$. Moreover, if $n \in V^{\mathbf{O}}$ and $m \neq n$ then $w \in V^{\mathbf{O}}$ and $m \neq w$.*

Proof. The proof is by induction on $|\mathsf{p}|$. Suppose that the proposition holds for all p with length n and consider $\mathsf{p} = \mathsf{p}'xy$ such that $\mathsf{p}x'$ has length n. All cases are trivial except when y is a **P**-move and x, y are in two different components. Suppose without loss of generality that $x \in [\![C]\!]$ and $y \in [\![A]\!]$ then $\mathsf{p} = s = \tau \overset{B}{\bullet} \rho|_{A,C}$ for $\tau \in \mathcal{T}$ and $\rho \in \mathcal{R}$ and s is equal to

$$s' x^{\mathbf{O}} b_1 b_1^\perp b_2^\perp b_2 b_3 \dots b_n b_n^\perp y^{\mathbf{P}} s''$$

by the definition of the $\sim$-relation, we have that $m = \mathsf{add}_x^k \overset{\mathsf{p}}{\sim} \mathsf{add}_y^k = n$. The w we are searching for is $\mathsf{add}_{b_1}^k$. Remark that $\mathsf{add}_{b_1}^k \overset{\rho}{\sim} m$. Suppose that $n \neq m$ and $n \in V^{\mathbf{O}}_{[\![A]\!]}$. By lemma 4.4 for all $i \in \{1, \dots n\}$ the modality $\mathsf{add}_{b_i^\perp}^k \in V^{\mathbf{P}}_{B_2}$. This means that, $\mathsf{add}_{b_1}^k \in V^{\mathbf{O}}_{[\![B_1]\!]}$. Since $w = \mathsf{add}_{b_1}^k$ and $m = \mathsf{add}_x^k$ are vertices in respectively, $[\![B_1]\!]$ and $[\![C]\!]$ we can conclude that $w \neq m$ as we wanted.

□

We are now ready to prove that each $\overset{\mathsf{p}}{\sim}$ class of a view $\mathsf{p} \in \mathcal{T} \overset{B}{\bullet} \mathcal{R}$ contains exactly one positive modality.

Proposition 4.5. *Let $\mathcal{S}$ and $\mathcal{R}$ be two* CK-**WIS** *for respectively $A \supset B_1$ and $B_2 \supset C$ with B_1 and B_2 two occurrences of the same formula B. The strategy $\mathcal{S} = \mathcal{T} \overset{B}{\circ} \mathcal{R}$ is* linked *i.e., for all $\mathsf{p} \in \mathcal{S}$ each $\overset{\mathsf{p}}{\sim}$-class contains exactly one even modality m.*

Proof. By lemma 4.3 we know, since $\mathcal{S}$ is well batched, that each $\overset{\mathsf{p}}{\sim}$-class contains at least one even modality m for $\mathsf{p} \in \mathcal{S}$. Suppose that the class contains another distinct even modal vertex n. Then we have that $m \overset{\mathsf{p}}{\sim} n$. Remark that $\mathsf{p} = \tau \overset{B}{\bullet} \rho$ for some $\tau \in \mathcal{T}$ and $\rho \in \mathcal{R}$. By lemma 4.5 there is a modal vertex $w \neq m$ in either $[\![B_1 \supset C]\!]$ or in $[\![A \supset B_2]\!]$ such that $m \overset{\tau}{\sim} w$ or $m \overset{\rho}{\sim} w$. This contradicts the fact that $\mathcal{T}$ and $\mathcal{R}$ are linked. □

We can now prove that the composition of CK-**WIS**s is a CK-**WIS**.

Theorem 4.3. *Let $\mathcal{T}$ and $\mathcal{R}$ be* CK-**WIS** *for respectively $A \supset B_2$ and $B_1 \supset C$ such that B_1 and B_2 are occurrences of the same formula B. Then $\mathcal{S} = \mathcal{T} \overset{B}{\circ} \mathcal{R}$ is a* CK-**WIS**.

Proof. After Lemma 4.2, and proposition 4.5 it suffices to prove that $\mathcal{S}$ is CK-batched, i.e., by using $\mathcal{G}$ for $[\![A \supset C]\!]$ we must prove that

1. if $w^{\mathbf{O}} \in V^{\Box}_{\mathcal{G}}$ and $v \overset{\mathcal{G}_{\mathsf{p}}}{\rightharpoonup} w$ for a $\mathsf{p} \in \mathcal{S}$, then $v \in V^{\Box}_{\mathcal{G}}$

2. if $w^{\mathbf{O}} \in V^{\Diamond}_{\mathcal{G}}$, then there is a unique $u \in V^{\Diamond}_{\mathcal{G}}$ in the set $\{v \in V_{\mathcal{G}} \mid v \overset{\mathcal{G}_{\mathsf{p}}}{\rightharpoonup} w \text{ for a } \mathsf{p} \in \mathcal{S}\}$.

For this purpose we define for each $\sigma = \tau \circ \rho$ the relation $\overset{\tau \bullet \rho}{\rightharpoonup}$ on the vertices in $[\![A, B_1 \supset B_2 \vdash C]\!]$ as the transitive closure of the following relation

$$v \overset{\tau \bullet \rho}{\rightharpoonup}_1 w \iff v^{\mathbf{P}} \overset{\tau}{\rightharpoonup} w^{\mathbf{O}} \text{ or } v^{\mathbf{P}} \overset{\rho}{\rightharpoonup} w^{\mathbf{O}} \text{ or } v = w \text{ or } v^{\mathbf{O}} = (w^{\mathbf{P}})^{\perp}$$

where we write $w \overset{\perp}{\leftharpoonup} v$ if $v^{\mathbf{O}} = (w^{\mathbf{P}})^{\perp}$.

We use $\overset{\tau\bullet\rho}{\rightharpoonup}$ to prove the properties of $\overset{\sigma}{\rightharpoonup}$. In fact it follows by induction on $\mathsf{p} = \tau \overset{B}{\circ} \rho$ that $v \overset{\mathsf{p}}{\rightharpoonup} w$ iff $v \overset{\tau\bullet\rho}{\rightharpoonup} w$.

Suppose that $v^{\mathbf{P}} \overset{\tau\bullet\rho}{\rightharpoonup} w^{\mathbf{O}}$. We prove 1 by contraposition. If v is labelled by $\diamond$, then also w is labelled by $\diamond$: this follows by induction on the definition of $\overset{\tau\bullet\rho}{\rightharpoonup}$. The base case of the induction follows by the fact that $\mathcal{T}$ and $\mathcal{R}$ are CK-**WIS** and by the fact that $-^{\perp}$ is a bijection that preserves labels.

To prove 2, we remark that $\mathcal{T}$ and $\mathcal{R}$ are linked, then $\overset{\tau}{\rightharpoonup}$ and $\overset{\rho}{\rightharpoonup}$ can be considered as injective functions associating a vertex in $V^{\mathbf{P}}$ a unique vertex in $V^{\mathbf{O}}$. Then also $\overset{\tau\bullet\rho}{\rightharpoonup}$ can be considered as an injective function since $\overset{\perp}{\leftharpoonup}$ is a bijection.

□

Proposition 4.6. *Let A, B, C and D formulas. If $\mathcal{S}$ is a* CK-**WIS** *for $A \vdash B$ and $\mathcal{T}$ is a* CK-**WIS** *for $B \vdash C$ and $\mathcal{R}$ is a* CK-**WIS** *for $C \vdash D$, then $(\mathcal{S} \overset{B}{\circ} \mathcal{T}) \overset{C}{\circ} \mathcal{R} = \mathcal{S} \overset{B}{\circ} (\mathcal{T} \overset{C}{\circ} \mathcal{R})$.*

Proof. The operation $\bullet$ is associative by Definition 4.12. Moreover, for any Δ and Σ sequences of formulas, the projections on Δ and Σ permute, that is, $(s|_{\Delta})|_{\Sigma} = s|_{\Delta,\Sigma} = (s|_{\Sigma})|_{\Delta}$. We conclude by observing that for any $\sigma \in \mathcal{S}$, $\tau \in \mathcal{T}$ and $\rho \in \mathcal{R}$ such that $\sigma \overset{B}{\equiv} \tau$ and $\tau \overset{C}{\equiv} \rho$ we have

$$
\begin{aligned}
\sigma \overset{B}{\circ} (\tau \overset{C}{\circ} \rho) = \quad & (\sigma \overset{A}{\bullet} (\tau \overset{B}{\circ} \rho))|_{A,D} \; = \; (\sigma \overset{B}{\bullet} ((\tau \overset{C}{\bullet} \rho)|_{B,D}))|_{A,D} \quad = \\
= \quad & (\sigma \overset{B}{\bullet} (\tau \overset{C}{\bullet} \rho))|_{A,D} \; = \; ((\sigma \overset{B}{\bullet} \tau) \overset{C}{\bullet} \rho|_{\Gamma,\Delta,C} \quad = \\
= \; & (((\sigma \overset{B}{\bullet} \tau)|_{A,C}) \overset{C}{\bullet} \rho)|_{A,D} \; = \; (((\sigma \overset{B}{\circ} \tau)A, C) \overset{C}{\bullet} \rho)|_{A,D} = (\sigma \overset{B}{\circ} \tau) \overset{C}{\circ} \rho
\end{aligned}
$$

since $\sigma \overset{B}{\circ} \tau \overset{C}{\equiv} \tau \overset{C}{\equiv} \rho$ and $\tau \overset{C}{\circ} \rho \overset{B}{\equiv} \tau \overset{B}{\equiv} \rho$. □

4.6 Some remarkable strategies

In this section we define the strategies that will be used to interpret proofs of LCK. Most of the strategies that we will define are presented in detail in the literature of game semantics [77, 76, 1, 2] where it is proved that they are winning innocent strategies. When the verification of the additional CK-**WIS** condition is trivial, we will omit such verification.

In what follows if $\mathcal{G}$ is an arena then $\mathcal{V}_{\mathcal{G}}$ denotes the set of views over $\mathcal{G}$. Such set is always non-empty because it always contains at least the empty view ϵ.

We start with the copy-cut strategy that will be the interpretation of the identity axiom $A \vdash A$. The copy-cut strategy can informally be explained as follows: I claim that I can always win a game of chess against either Magnus Carlsen or Fabiano Caruana if games are played simultaneously and I am black on one board and white on the other. The strategy that I will follow is simple. Suppose that Carlsen starts as white on a board and make a move m, then i will play m on the other board against Caruana, wait for his move n and play the same move n in the game against Carlsen. Carlsen will reply by a move r, I will make the move r in the chess game against Caruana and so on. Since Chess are zero-sum game I must loose in one of the two games and win in the other. The copy-cut strategy mimics exactly this mechanism. Each time **O** makes a move on one copy of A in the arena $[\![A \supset A]\!]$ then **P** will make the same move in the other copy of A. More formally:

Proposition 4.7. *For any formula A $\mathtt{Id}_A$, the copycat strategy over the arena $[\![A \supset A]\!]$, defined by*

$$\mathtt{Id}_A = \{\rho \in \mathcal{V}_{[\![A_1 \supset A_2]\!]} |\ \ \tau|_{A_1} = \tau|_{A_2}. \text{ for each even-length prefix } \tau \text{ of } \rho\}$$

is a CK-**WIS**. *We use A_1 and A_2 to distinguish the two different occurrences of A*

Proof. The fact that $\mathtt{Id}_A$ is a winning innocent strategy for any A is a classical result of game-semantics and can be found in [75]. The fact that each view $\tau \in \mathtt{Id}_A$ is well batched and CK-framed comes from the fact that if $\tau' x^{\mathbf{O}} y^{\mathbf{P}} \in \mathtt{Id}_A$ then $y = x^{\perp}$ □

Proposition 4.8. *Let $\mathcal{S}$ be a* CK-**WIS** *for $A \supset B$ then $\mathcal{S} = \mathtt{Id}_A \overset{A}{\circ} \mathcal{S} = \mathcal{S} \overset{B}{\circ} \mathtt{Id}_B$*

The CK-**WIS** that interprets a $\wedge^{\mathsf{R}}$ application to some sequents $\Gamma \vdash A$ and $\Delta \vdash B$ is obtained by ‘ putting side by side’ CK-**WIS**s over $[\![\Gamma \vdash A]\!]$ and $[\![\Delta \vdash B]\!]$

Proposition 4.9. *Let $\mathcal{T}$ and $\mathcal{R}$ be* CK-**WIS** *for respectively $A \supset C$ and $B \supset D$ then*

$$\mathcal{T} \wedge \mathcal{R} = \{\sigma \in \mathcal{V}_{[\![A \wedge B \supset C \wedge D]\!]} \,|\, \sigma|_{A,C} \in \mathcal{T} \text{ and } \sigma|_{B,D} \in \mathcal{R}\,\}$$

is a CK-**WIS** *for $A \wedge B \supset C \wedge D$. Moreover $\mathtt{Id}_{A \wedge B} = \mathtt{Id}_A \wedge \mathtt{Id}_B$.*

The CK-**WIS** that will be used to interpret the contraction rule is a CK-**WIS** over $A \supset A \wedge A$. This CK-**WIS** (the diagonal strategy) plays as the copy-cut strategy on A for both occurrences of A in $A \wedge A$.

Proposition 4.10. *Let A be a formula, the following set of views defines a* CK-**WIS** *for $A \supset A \wedge A$*

$$\mathcal{D}_A = \{\sigma \in \mathcal{V}_{[\![A_1 \supset A_2 \wedge A_3]\!]} \mid \sigma|_{A_1,A_3} \in \mathtt{Id}_A \text{ and } \sigma|_{A_1,A_2} \in \mathtt{Id}_A\}$$

Where we used indices to distinguish the three occurrences of A.

If $\mathcal{S}$ is a CK-**WIS** over $\mathcal{G}$, $\sigma_0\sigma_1 \dots \sigma_n = \sigma \in \mathcal{S}$ and f is a function from $\mathcal{G}$ to $\mathcal{H}$ then $f(\sigma)$ is the sequence $f(\sigma_0)f(\sigma_1), \dots f(\sigma_n)$ in $\mathcal{H}$. We write $f(\mathcal{S})$ for

$$f(\mathcal{S}) = \{\tau \in \mathcal{V}_{\mathcal{H}} \mid \tau = f(\sigma) \text{ for a } \sigma \in \mathcal{S}\}$$

Proposition 4.11. *if $\mathcal{S}$ is a* CK-**WIS** *for $(A \wedge B) \supset C$ then $\lambda_B(\mathcal{S})$ is a* CK-**WIS** *for $A \supset (B \supset C)$. If, vice versa, $\mathcal{S}$ is a* CK-**WIS** *for $A \supset (B \supset C)$ then $\lambda^{-1}(\mathcal{S})$ is a* CK-**WIS** *for $(A \wedge B) \supset C$, where λ and $\lambda - 1$ are the* 2-dag*-isomorphism defined in proposition 4.4*

Proof. The proof follows by observing that λ and λ^{-1} are 2-dag isomorphism. They are thus bijections that preserves labels and edges. □

Let $\mathtt{Id}_{B \supset C}$ be the copy-cut strategy for $(B \supset C) \supset (B \supset C)$. Then $\lambda^{-1}(\mathtt{Id}_{B \supset C})$ is a CK-**WIS** for $((B \supset C) \wedge B) \supset C$ that we will call. $\mathtt{Eval}_{B,C}$. Given a CK-**WIS** $\mathcal{S}$ for $(A \wedge B) \supset C$ it is easy to verify that

$$(\lambda_B(\mathcal{S}) \times \mathtt{Id}_B) \overset{(B \supset C) \wedge C}{\circ} \mathtt{Eval}_{B,C} = \mathcal{S}$$

We now define the CK-**WIS**s that will be used to interpret the two modal rules of LCK: the rule $\mathsf{K}^{\Box}$ and the rule $\mathsf{K}^{\Diamond}$. Given a strategy $\mathcal{S}$ over the arena $[\![\Gamma \vdash C]\!]$ the strategy interpreting the $\mathsf{K}^{\Box}$-rule will be –morally– $\mathcal{S}$ played over the arena $[\![\Box\Gamma \vdash \Box C]\!]$ i.e., $\Box\mathcal{S}$ will be $\mathcal{S}$ in which each move in a view σ has a new address $\Box$. The strategy that interprets the $\mathsf{K}^{\Diamond}$-rule is defined similarly : given a strategy $\mathcal{T}$ over the arena $[\![\Gamma, B \vdash C]\!]$ the strategy interpreting the $\mathsf{K}^{\Diamond}$-rule will be —morally— $\mathcal{T} \cup \{\Diamond^{\mathbf{O}}\Diamond^{\mathbf{P}}\}$ i.e., $\mathcal{T}$ played over the arena $[\![\Box\Gamma, \Diamond B \vdash \Diamond C]\!]$.

Proposition 4.12. *Let $\mathcal{S}$ and $\mathcal{T}$ be* CK-**WIS***s over, respectively, the arenas $[\![A_1, \dots A_n \vdash C]\!]$ and $[\![B_1, \dots, B_m, B \vdash C]\!]$, then*

1. $\Box\mathcal{S} = \{\tau \in \mathcal{V}_{[\![\Box A_1, \dots, \Box A_n \vdash \Box C]\!]} \mid \tau|_{A_1,\dots A_n,C} \in \mathcal{S}\}$

2. $\Diamond\mathcal{T} = \{\tau \in \mathcal{V}_{[\![\Box B_1,\dots\Box B_m,\Diamond B\vdash\Diamond C]\!]} \mid \tau|_{B_1,\dots B_m,B,C} \in \mathcal{T}\}$

are CK-**WIS***s over the arenas* $[\![\Box A_1,\dots\Box A_n \vdash \Box C]\!]$ *and* $[\![\Box B_1,\dots,\Box B_m,\Diamond B \vdash \Diamond C]\!]$ *respectively.*

Proof. It is clear that if $\Diamond(\mathcal{S})$ and $\Box(\mathcal{S})$ are **WIS**. We only prove that the **WIS** of 2 is a CK-**WIS**. By writing Γ for $B_1,\dots,B_m$:

- $\Diamond\mathcal{T}$ is well batched : let $\tau \in \Diamond\mathcal{T}$ and let $x^{\mathbf{O}}y^{\mathbf{P}}$ two consecutive moves in τ. We must prove that $|\mathsf{add}_x| = |\mathsf{add}_y|$. We have that in $\tau|_{\Gamma,B,C}$ $|\mathsf{add}_x| = |\mathsf{add}_y|$ because $\mathcal{T}$ is a CK-**WIS**. We can conclude because for $w \in \tau$ $|\mathsf{add}_w| = |\mathsf{add}_w| + 1$ for $w \in \tau|_{\Gamma,B,C}$

- $\Diamond\mathcal{T}$ is linked : condition 1 of definition 4.11 automatically holds: the arena of $\Diamond\mathcal{T}$ and the arena of $\mathcal{T}$ have the same even vertex labeled by a $\Box$. To check that condition 2 of definition 4.11 let us denote by $\Diamond^{\mathbf{O}}$ the vertex v such that $\ell(v) = \Diamond$ and $v \rightsquigarrow [\![C]\!]$ and by $\Diamond^{\mathbf{P}}$ the vertex w such that $\ell(w) = \Diamond$ and $v \rightsquigarrow [\![A]\!]$. By the definition of $\Diamond\mathcal{S}$ the view $\epsilon\Diamond^{\mathbf{O}}\Diamond^{\mathbf{P}}$ belong to $\Diamond(\mathcal{S})$, thus we obtain that $\Diamond^{\mathbf{P}} \rightharpoonup \Diamond^{\mathbf{O}}$. The fact that there is no other vertex v such that $v \rightharpoonup \Diamond^{\mathbf{O}}$ is evident.

□

4.7 Game semantics interpretation of CK

A derivation $\mathfrak{D}$ of $\Gamma \vdash A$ will be interpreted as CK-**WIS** $\{\!\{\mathfrak{D}\}\!\} : [\![\Gamma \vdash A]\!]$. The interpretation is given by induction on the proof $\mathfrak{D}$ of $\Gamma \vdash A$.

If $\mathfrak{D}$ has height 0 then $\mathfrak{D}$ is obtained by the rule AX and has conclusion $A \vdash A$ for some formula A or by the rule 1, and has conclusion $\vdash 1$ In the first case, we define $\{\!\{\mathfrak{D}\}\!\} = \mathtt{Id}_A$. In the second case $\{\!\{\mathfrak{D}\}\!\}$ is the trivial strategy $\{\epsilon\}$. Suppose that for all sequents $\Gamma' \vdash C'$ for all proof $\mathfrak{D}'$ of $\Gamma' \vdash C'$ if $|\mathfrak{D}'| \leq n$ then the interpretation $\{\!\{\mathfrak{D}'\}\!\}$ of $\mathfrak{D}'$ is defined. Let $\mathfrak{D}$ be a derivation with $|\mathfrak{D}| = n + 1$

$\wedge^{\mathsf{R}}$ if $\mathfrak{D}$ is

$$\dfrac{\begin{matrix}\mathfrak{D}_1\\ \vdots\\ \Gamma \vdash A\end{matrix} \qquad \begin{matrix}\mathfrak{D}_1\\ \vdots\\ \Delta \vdash B\end{matrix}}{\Gamma, \Delta \vdash A \wedge B}$$

By induction hypotheses $\{\!\{\mathfrak{D}_1\}\!\} : [\![\Gamma \vdash A]\!]$ and $\{\!\{\mathfrak{D}_2\}\!\} : [\![\Delta \vdash B]\!]$ are CK-**WIS**. we define $\{\!\{\mathfrak{D}\}\!\} = \{\!\{\mathfrak{D}_1\}\!\} \wedge \{\!\{\mathfrak{D}_2\}\!\} : [\![\Gamma \wedge \Delta \vdash A \wedge B]\!]$.

$\wedge^{\mathsf{L}}$ if $\mathfrak{D}$ is

$$\dfrac{\begin{matrix}\mathfrak{D}' \\ \vdots \\ \Gamma, A, B \vdash C\end{matrix}}{\Gamma, A \wedge B \vdash C}$$

By induction hypothesis $\{\!\{\mathfrak{D}'\}\!\} : [\![\Gamma \wedge A \wedge B \vdash C]\!]$ is a CK-**WIS**. We thus define $\{\!\{\mathfrak{D}\}\!\} = \{\!\{\mathfrak{D}'\}\!\}$.

$\supset^{\mathsf{R}}$ If $\mathfrak{D}$ is

$$\dfrac{\begin{matrix}\mathfrak{D}' \\ \vdots \\ \Gamma, B \vdash C\end{matrix}}{\Gamma \vdash B \supset C}$$

By induction hypothesis $\{\!\{\mathfrak{D}'\}\!\} : [\![\Gamma \wedge B \vdash C]\!]$ is a CK-**WIS**. We define $\{\!\{\mathfrak{D}\}\!\} = \lambda(\{\!\{\mathfrak{D}'\}\!\}) : [\![\Gamma \vdash B \supset C]\!]$.

$\supset^{\mathsf{L}}$ if $\mathfrak{D}$ is

$$\dfrac{\begin{matrix}\mathfrak{D}_1 \\ \vdots \\ \Gamma \vdash A\end{matrix} \qquad \begin{matrix}\mathfrak{D}_2 \\ \vdots \\ B, \Delta \vdash C\end{matrix}}{A \supset B, \Gamma, \Delta \vdash C}$$

Then by induction hypothesis $\{\!\{\mathfrak{D}_1\}\!\} : [\![\Gamma \vdash A]\!]$ and $\{\!\{\mathfrak{D}_2\}\!\} : [\![B \wedge \Delta \vdash C]\!]$ are CK-**WIS**s. We construct $\{\!\{\mathfrak{D}\}\!\}$ as follows:

Remark that $g = \mathtt{Id}_{A \supset B} \wedge \{\!\{\mathfrak{D}_1\}\!\} : [\![(A \supset B) \wedge \Gamma \vdash (A \supset B) \wedge A]\!]$ is a CK-**WIS**. Thus, $g \circ \mathtt{eval}_{A,B} : [\![A \supset B \wedge \Gamma \vdash B]\!]$ is also a CK-**WIS**. We finally define $\{\!\{\mathfrak{D}\}\!\} = (g \circ \mathtt{eval}_{A,B}) \wedge \mathtt{Id}_\Delta \circ \{\!\{\mathfrak{D}_2\}\!\} : [\![A \supset B \wedge \Gamma \wedge \Delta \vdash C]\!]$.

$\mathsf{K}^\Box$ If $\mathfrak{D}$ is

$$\dfrac{\begin{array}{c}\mathfrak{D}' \\ \vdots \\ \Gamma \vdash C\end{array}}{\Box\Gamma \vdash \Box C}$$

With $\Gamma = A_1, \ldots A_n$. By induction hypothesis $\{\!\{\mathfrak{D}\}\!\} : [\![A_1, \ldots, A_n \vdash C]\!]$ is a CK-**WIS**. We define $\{\!\{\mathfrak{D}\}\!\} = \Box\{\!\{\mathfrak{D}'\}\!\} : [\![\Box A_1 \ldots \Box A_n \vdash \Box C]\!]$.

$\mathsf{K}^\Diamond$ If $\mathfrak{D}$ is

$$\dfrac{\begin{array}{c}\mathfrak{D}' \\ \vdots \\ \Gamma, B \vdash C\end{array}}{\Box\Gamma, \Diamond B \vdash \Diamond C}$$

with $\Gamma = A_1, \ldots A_n$. By induction hypothesis $\{\!\{\mathfrak{D}\}\!\} : [\![A_1 \wedge \cdots A_n \wedge B \vdash C]\!]$ is a CK-**WIS**. We define $\{\!\{\mathfrak{D}\}\!\} = \Diamond\{\!\{\mathfrak{D}'\}\!\} : [\![\Box A_1, \ldots \Box A_n, \Diamond B \vdash \Diamond C]\!]$.

C If $\mathfrak{D}$ is

$$\dfrac{\begin{array}{c}\mathfrak{D}' \\ \vdots \\ \Gamma, A, A \vdash C\end{array}}{\Gamma, A \vdash C}$$

Then by induction hypothesis $\{\!\{\mathfrak{D}'\}\!\} : [\![\Gamma \wedge A \wedge A \vdash C]\!]$ is a CK-**WIS**. We define $\{\!\{\mathfrak{D}\}\!\} = (\mathtt{Id}_\Gamma \wedge \mathcal{D}_A) \circ \{\!\{\mathfrak{D}'\}\!\} : [\![\Gamma \wedge A \vdash C]\!]$. Where $\mathcal{D}_A$ is the diagonal strategy on $[\![A \supset A \wedge A]\!]$.

W If $\mathfrak{D}$ is

$$\dfrac{\begin{array}{c}\mathfrak{D}\\ \vdots \\ \Gamma \vdash C\end{array}}{\Gamma, B \vdash C}$$

Then by induction hypothesis $\{\!\{\mathfrak{D}'\}\!\} : [\![\Gamma \vdash C]\!]$ is a CK-**WIS**. Remark that for every modal arena $\mathcal{G}$ the trivial CK-**WIS** $\{\epsilon\}$ is a CK-**WIS** over $\mathcal{G} \supset [\![1]\!]$. Call this latter strategy h. We obtain $\{\!\{\mathfrak{D}'\}\!\} \times h : [\![\Gamma \wedge B \vdash C \wedge 1]\!]$. We define $\{\!\{\mathfrak{D}\}\!\} = (\{\!\{\mathfrak{D}'\}\!\} \wedge h) \circ l_{C^\star}$ where $l_{C^\star}$ is $l(\mathtt{Id}_{C \wedge 1})$.

cut Finally, if $\mathfrak{D}$ is

$$\dfrac{\begin{array}{c}\mathfrak{D}_1\\ \vdots \\ \Gamma \vdash F\end{array} \qquad \begin{array}{c}\mathfrak{D}_2\\ \vdots \\ \Delta, F \vdash C\end{array}}{\Delta, \Gamma \vdash C}$$

Then by induction hypothesis $\{\!\{\mathfrak{D}_1\}\!\} : [\![\Gamma \vdash F]\!]$ and $\{\!\{\mathfrak{D}_2\}\!\} : [\![\Delta \wedge F \vdash C]\!]$ are are CK-**WIS**. We put $\{\!\{\mathfrak{D}\}\!\} = (\mathtt{Id}_\Delta \wedge \{\!\{\mathfrak{D}_1\}\!\}) \circ \{\!\{\mathfrak{D}_2\}\!\}$ where $\mathtt{Id}_\Delta = \bigwedge_{i=1}^{m} \mathtt{Id}_{B_i}$ if $\Delta = B_1, \ldots B_m$.

4.7.1 Denotational Model

In this section we show that if a proof $\mathfrak{D}$ reduces to a proof $\mathfrak{D}'$ then $\{\!\{\mathfrak{D}\}\!\} = \{\!\{\mathfrak{D}'\}\!\}$. To show that this property holds it is enough to show that if $\mathfrak{D}'$ is the transformation associated by the cut-elimination procedure detailed in section 4.2.3 to $\mathfrak{D}$ then $\{\!\{\mathfrak{D}\}\!\} = \{\!\{\mathfrak{D}'\}\!\}$.

Proposition 4.13. *Let $\mathfrak{D}$ and $\mathfrak{D}'$ two* LCK *derivations such that $\mathfrak{D}'$ is the transformation associated to $\mathfrak{D}$ by the cut elimination procedure. We have that $\{\!\{\mathfrak{D}\}\!\} = \{\!\{\mathfrak{D}'\}\!\}$.*

Proof. By taking arenas as objects and **WIS**s as arrows one obtain a Cartesian Closed Category [93, 75]. In [110] the authors shows that an arbitrary Cartesian closed category is a denotational model for the sequent calculus LCK restricted to $\supset$ and $\wedge$-formulas. The interpretation of sequent calculus proofs that we have detailed is the same used —in the context of an arbitrary Cartesian closed category— by the authors of the aforementioned paper. By this reason it is enough to consider the three modal cases of cut elimination ($\mathsf{K}^{\Box}/\mathsf{K}^{\Box}$), ($\mathsf{K}^{\Diamond}/\mathsf{K}^{\Diamond}$) and ($\mathsf{K}^{\Box}/\mathsf{K}^{\Diamond}$). Since the three case are almost identical, we detail only one of the three.

($\mathsf{K}^{\Box}/\mathsf{K}^{\Box}$) if $\mathfrak{D}$ is

$$\dfrac{\dfrac{\begin{array}{c}\mathfrak{D}_{1.1}\\ \vdots\\ A_1,\dots A_n \vdash F\end{array}}{\Box A_1,\dots,\Box A_n \vdash \Box F} \qquad \dfrac{\begin{array}{c}\mathfrak{D}_{2.1}\\ \vdots\\ B_1,\dots B_m, F \vdash C\end{array}}{\Box B_1,\dots B_m, \Box F \vdash \Box C}}{\Box B_1,\dots,\Box B_m, \Box A_1,\dots,\Box A_n \vdash \Box C}$$

The transformation associate to $\mathfrak{D}$ is the following proof $\mathfrak{D}'$

$$\dfrac{\dfrac{\begin{array}{c}\mathfrak{D}_{1.1}\\ \vdots\\ A_1,\dots,A_n \vdash F\end{array} \qquad \begin{array}{c}\mathfrak{D}_{2.1}\\ \vdots\\ B_1,\dots B_m, F \vdash C\end{array}}{B_1,\dots,B_m, A_1,\dots,A_n, \vdash C}}{\Box B_1,\dots,\Box B_m, \Box A_1,\dots,\Box A_n \vdash \Box C}$$

the interpretations $\{\!\{\mathfrak{D}\}\!\}$ and $\{\!\{\mathfrak{D}'\}\!\}$[6] of $\mathfrak{D}$ and $\mathfrak{D}'$ are respectively

$$\{\!\{\mathfrak{D}\}\!\} = \Big(\big(\bigwedge_{i=1}^{m} \mathtt{Id}_{\Box B_i}\big) \wedge \Box\{\!\{\mathfrak{D}_{1.1}\}\!\}\Big) \overset{(\bigwedge_{i=1}^{m}\Box B_i)\wedge\Box F}{\circ} \Box\{\!\{\mathfrak{D}_{2.1}\}\!\}$$

[6]Remark that this strategy is well-defined since by definition 4.5, $[\![\bigwedge_{i=1}^{m} B_i \vdash C]\!] = [\![B_1,\dots B_m \vdash C]\!]$

$$\{\!\{\mathfrak{D}'\}\!\} = \Box(((\bigwedge_{i=1}^{m} \mathtt{Id}_{B_i}) \wedge \{\!\{\mathfrak{D}_{1.1}\}\!\}) \overset{(\bigwedge_{i=1}^{m} B_i)\wedge F}{\circ} \{\!\{\mathfrak{D}_{2.1}\}\!\})$$

The result follows immediately by observing that $\{\!\{\mathfrak{D}\}\!\}|_{\Delta,\Gamma,C}$ is equal to $\{\!\{\mathfrak{D}'\}\!\}|_{\Delta,\Gamma,C}$ with $\Delta = B_1, \dots B_m$ and $\Gamma = A_1, \dots A_n$.

□

4.8 Full Completeness

In this section we prove that our game model is fully complete: each CK-**WIS** $\mathcal{S}$ is the interpretation of a proof $\mathfrak{D}$. To prove this result we give a sequenzialization procedure for CK-**WIS** i.e., an algorithm that permits to transform any CK-**WIS** in a LCK-derivation. In particular CK-**WIS** will be transformed into derivation in which leaves obtained from the AX-rule are of the form $a \vdash a$ with a atomic. This is not a limitation, since we can always expand instances of the AX-rule up to atomic formulas.

In order to provide the sequentialization procedure for CK-**WIS** we prove three preliminary lemmas. In the first two lemmas give a way to sequentialize the CK-**WIS**s when a $\wedge$ in the right-hand side of the sequent or a $\supset$ in the left-hand side of the sequent occurs. In the sequent calculus LCK these connective require the use of rules splitting the context. In order to avoid reproving the splitting lemmas from [113], we adopt a simpler approach relying on the presence of W and C in the sequent system. The third result proves that the presence of the two rules $K^{\Box}$ and $K^{\Diamond}$ can be easily recognized and sequentialized by only considering the shape of the conclusion sequent and the CK-framing conditions.

Lemma 4.6. *Let $\Gamma \vdash A_1 \wedge A_2$ such that Γ does not contain $\wedge$-formulas, If $\mathcal{S}$ is CK-**WIS** for $\Gamma \vdash A_1 \wedge A_2$, then there are CK-**WIS**s $\mathcal{S}_1$ and $\mathcal{S}_2$ for $\Gamma \vdash A_1$ and $\Gamma \vdash A_2$.*

Proof. For $i \in \{1, 2\}$ we let $\mathcal{S}_i$ be the set of views in $\mathcal{S}$ starting from a move in A_i plus the empty view, that is, $\mathcal{S}_i = \{\mathsf{p} \in \mathcal{S} \mid \mathsf{p}_0 \in \overrightarrow{R}_{[\![A_i]\!]}\} \cup \{\epsilon\}$. By definition of the arena $[\![\Gamma \vdash A_1 \wedge A_2]\!]$, no move in A_i may occur in a view in $\mathcal{S}_j$ whenever $i \neq j$. Hence $\mathcal{S}_1$ and $\mathcal{S}_2$ are CK-**WIS**s for $\Gamma \vdash A_1$ and $\Gamma \vdash A_2$ respectively. □

Lemma 4.7. *Let $\mathcal{S}$ be a CK-**WIS** for $\Gamma \vdash c$. Suppose that Γ does not contain any $\wedge$-formula. Then either*

- *The unique $c^{\mathbf{P}}$ such that $c^{\mathbf{O}}c^{\mathbf{P}} \in \mathcal{S}$ is en element of $[\![\Gamma]\!]$ and $c^{\mathbf{O}}c^{\mathbf{P}}$ is maximal in $\mathcal{S}$;*

- *or $\Gamma = \Gamma', A \supset B$ for some formulas A and B, the unique $c^{\mathbf{P}}$ such that $c^{\mathbf{O}}c^{\mathbf{P}} \in \mathcal{S}$ belongs the root of $[\![A \supset B]\!]$ and there are two* CK-**WIS** *$\mathcal{S}$ and $\mathcal{R}$ for, respectively, $\Gamma' \vdash A$ and $B, A \supset B, \Gamma' \vdash c$*

Proof. Remark that the view $c^{\mathbf{O}}c^{\mathbf{P}}$ must belong to $\mathcal{S}$ because of totality and $c^{\mathbf{P}}$ is uniquely determined because of determinism. Moreover, $c^{\mathbf{O}}c^{\mathbf{P}}$ is a prefix of every view in $\mathcal{S}$ with length bigger or equal to 2.

We have two possibilities, either $c^{\mathbf{P}} = [\![F]\!]$ for $F \in \Gamma$ or $c^{\mathbf{P}} \in [\![F]\!]$ for some $F \in \Gamma$. In the first case, $\Gamma = \Gamma', c$ and $\mathcal{S} = \{\epsilon, \epsilon c^{\mathbf{O}}, \epsilon c^{\mathbf{O}}c^{\mathbf{P}}\}$; this is because there is no vertex that points $c^{\mathbf{P}}$ in $[\![\Gamma']\!]$.

if we are in the second case i.e., $c^{\mathbf{P}}$ is a vertex in $[\![F]\!]$ for some $F \in \Gamma$ then necessarily, since Γ does not contain $\wedge$ formulas, F is of the form $A \supset B$ and $c^{\mathbf{P}}$ is a root of $[\![B]\!]$. If a view σ in $\mathcal{S}$ contains a move in A, then it is of the form $c^{\mathbf{O}}c^{\mathbf{P}}\tau_0$ with $\tau_0 \in \vec{R}_{[\![A]\!]}$.

We first show that for a $v \in \vec{R}_{[\![A]\!]}$ there is a maximal $\sigma \in \mathcal{S}$ such that $v = \sigma_{2k}$ and σ_i is not a move in $B\{c^{\mathbf{P}}\}$ for any $i > 2k$. First all remark that if $v \in \vec{R}_{[\![A]\!]}$ then there is $\sigma \in \mathcal{S}$ such that $v^{\mathbf{O}}$ in σ because of **O**-completeness. Moreover, by the definition of view $v = \sigma_{2k}$ for some k.
let $i > 2k$ such that σ_i is the first move in $[\![B\{c^{\mathbf{P}}\}]\!]$, hence $\sigma_i \in \vec{R}_{[\![B\{c^{\mathbf{P}}\}]\!]}$. By **O**-completeness, there is a $\sigma' \in \mathcal{S}$ such that $\sigma' = \sigma_0 \cdots \sigma_i v$. Again, $v = \sigma'_{2k'}$ for some $k \in \mathbb{N}$; We can repeat the same reasoning as above. Since views are finite we will find a view $\sigma'' \in \mathcal{S}$ such that $v = \sigma_{2k''}$ and for all $i > 2k''$ σ''_i is not a move in $[\![B]\!]$.

Now observe that since v was an arbitrary root of $[\![A]\!]$ the line of reasoning is valid for any root of $[\![A]\!]$ i.e., there is a σ^w with the same property and such that $\sigma^v_0 \cdots \sigma^v_{2k-1} = \sigma^w_0 \cdots \sigma^w_{2k-1}$ for any $w \in \vec{R}_A$. We define $\mathsf{Split}^A_{\mathcal{S}}$ to be the set containing such a view σ^w for each $w \in \vec{R}_{[\![A]\!]}$. All the σ^w share the same prefix. We use this $\mathsf{Split}^A_{\mathcal{S}}$ to define

$$\begin{aligned} \mathcal{T} &= \left\{\tau \mid \text{there are } \sigma \text{ and } \tau' \text{ such that } \sigma\tau\tau' \in \mathsf{Split}^A_{\mathcal{S}}\right\} \\ \mathcal{R} &= \left\{\rho \mid \text{there is no } \sigma \text{ such that } \rho\sigma \in \mathsf{Split}^A_{\mathcal{S}}\right\} \end{aligned}$$

By definition, $\mathcal{T}$ is a CK-**WIS** for $\Gamma \vdash A$ and $\mathcal{R}$ is a CK-**WIS** for $\Gamma, A \supset B, B \vdash c$ and both are strictly smaller than $\mathcal{S}$. □

Lemma 4.8. *Let $\mathcal{S}$ be a* CK-**WIS***s for $\Gamma' \vdash A'$ such that $A' = \Box A$ or $A' = \Diamond A$, the sequent Γ' is of the form $\Box\Gamma, \Diamond\Delta$. If at least one move from each formula in Γ' occurs in a view in $\mathcal{S}$, then*

- *either* $\Gamma' \vdash A'$ *is of the form* $\Box\Gamma \vdash \Box A$ *and* $\mathcal{S}|_{\Gamma,A}$ *is a* CK-**WIS** *for* $\Gamma \vdash A$
- *or* $\Gamma' \vdash A'$ *is of the form* $\Box\Gamma, \Diamond B \vdash \Diamond A$ *and* $\mathcal{S} = \mathcal{S}|_{\Gamma,B,C}$ *is a* CK-**WIS** *for* $\Gamma, B \vdash A$

Proof. If at least one move from each formula in Γ' occurs in a view in $\mathcal{S}$, then each principal modality of a formula in Γ' must occur in the first row of a batched view of a $\mathsf{p} \in \mathcal{S}$. By CK-batched condition, all the principal modalities of the formulas in Γ' must be in $\xrightarrow{\mathcal{S}}$-relation with the principal modality of A'. Hence, $\Gamma' \vdash A'$ is either of the form $\Box\Gamma \vdash \Box A$ or $\Box\Gamma, \Diamond B \vdash \Diamond A$. In the first case, we conclude by remarking that if we remove the first row in any batched view $\mathcal{F}(\mathsf{p})$ with $\mathsf{p} \in \mathcal{S}$, then we obtain a batched view of the same p, but in $\Gamma \vdash A$; this is exactly to say that $\mathcal{S}|_{\Gamma,A}$ is a CK-**WIS** for $\Gamma \vdash A$

The second case is treated similarly. We should only remark that $\mathcal{S}$ will contain a maximal view of the form $\Diamond\Diamond$ and that this view will not appear in $\mathcal{S}|_{\Gamma,B,C}$

□

Lemma 4.9 (Full Completeness)**.** *If $\mathcal{S}$ is a* CK-**WIS** *over* $[\![\Gamma \vdash A]\!]$ *then there is cut-free proof* $\mathfrak{D}$ *such that* $\{\!\{\mathfrak{D}\}\!\} = \mathcal{S}$.

Proof. The proof is by induction on the triple $(|\mathcal{S}|, ||A||, ||\Gamma||)$ where, $|\mathcal{S}|$ is the cardinality of $\mathcal{S}$, $||A||$ is the number of connectives, modalities and atoms in A and if $\Gamma = A_1, \ldots, A_n$ then $||\Gamma|| = \sum_{i=1}^{n} ||A_i||$. We remark that if in no view in a CK-**WIS** $\mathcal{S}$ for $\Gamma, B \vdash A$ contains moves in B, then $\mathcal{S}$ is a CK-**WIS** also for $\Gamma \vdash A$. Observe that in case of $\Diamond$-formulas occurring in Γ, we may have that only one of these $\Diamond$s occurring in a view. In this case, we expect to observe in the final derivation a $\mathsf{K}^{\Diamond}$-rule preceded (bottom-up) by a W-rule. Moreover, since $[\![\Gamma, B \wedge C \vdash A]\!] = [\![\Gamma, B, C \vdash A]\!]$, then each CK-**WIS** for the first sequent is a CK-**WIS** for the second one, but the size of the lhs sequent decreases. A similar reasoning applies to the sequents $\Gamma \vdash B \supset C$ and $\Gamma, B \vdash C$. We conclude by Lemmas 4.6, 4.7 and 4.8.

In Figure 4.4 we give a table resuming the sequentialization step to apply according to the shape of the sequent and the shape of the CK-**WIS**. The conditions on the sequent (first column) can be checked in the given order, triggering the corresponding sequentialization step. □

By the above lemma, the fact that we can interpret derivations in LCK as CK-**WIS**s and by the fact that proofs that are equal modulo the cut-elimination procedure are interpreted by the same CK-**WIS**, we can conclude that

Theorem 4.4. *The* CK-**WIS**s *form a fully-complete denotational semantics for* CK.

Sequent	Shape of $\mathcal{S}$	Shape of $\mathfrak{D}_{\mathcal{S}}$
$\vdash 1$	$\mathcal{S} = \{\epsilon\}$	$\dfrac{}{\vdash 1}\,\mathsf{1}$
$a \vdash a$	$\mathcal{S} = \{\epsilon, a, aa\}$	$\dfrac{}{a \vdash a}\,\mathsf{AX}$
$\Gamma, B \wedge C \vdash A$	any	$\dfrac{\overset{\mathfrak{D}_{\mathcal{S}}\,\Vert}{\Gamma, B, C \vdash A}}{\Gamma, B \wedge C \vdash A}\,\wedge^{\mathsf{L}}$
$\Gamma \vdash B \supset A$	any	$\dfrac{\overset{\mathfrak{D}_{\mathcal{S}}\,\Vert}{\Gamma, B \vdash A}}{\Gamma \vdash B \supset A}\,\supset^{\mathsf{R}}$
$\Gamma \vdash A_1 \wedge A_2$ Γ contains no $\wedge$-formula	$\mathcal{S} = \mathcal{T} \cup \mathcal{R}$ $\mathcal{T} = \left\{\tau \in \mathcal{S} \mid \tau \text{ contains no moves in } A_2\right\}$ $\mathcal{R} = \left\{\rho \in \mathcal{S} \mid \rho \text{ contains no moves in } A_1\right\}$	$\dfrac{\dfrac{\overset{\mathfrak{D}_{\mathcal{T}}\,\Vert}{\Gamma \vdash A_1} \quad \overset{\mathfrak{D}_{\mathcal{R}}\,\Vert}{\Gamma \vdash A_2}}{\Gamma, \Gamma \vdash A_1 \wedge A_2}\,\wedge^{\mathsf{R}}}{\Gamma \vdash A_1 \wedge A_2}\,\mathsf{C}$
$\Gamma, A \supset B\{c^{\mathbf{P}}\} \vdash c^{\mathbf{O}}$ Γ contains no $\wedge$-formulas c atomic $B\{c^{\mathbf{P}}\}$ contains the atom $c^{\mathbf{P}}$	$c^{\mathbf{O}} c^{\mathbf{P}} \in \mathcal{S}$ $\mathcal{T} = \left\{\tau \in \mathcal{S} \mid c^{\mathbf{O}} c^{\mathbf{P}} \tau \in \mathcal{S} \text{ with } \tau_0 \text{ move in } A\right\}$ $\mathcal{R} = \left\{\rho \in \mathcal{S} \mid \rho \text{ contains no moves in } A\right\}$	$\dfrac{\dfrac{\overset{\mathfrak{D}_{\mathcal{T}}\,\Vert}{\Gamma \vdash A} \quad \overset{\mathfrak{D}_{\mathcal{R}}\,\Vert}{\Gamma, B\{c^{\mathbf{P}}\} \vdash c^{\mathbf{O}}}}{\Gamma, \Gamma, A \supset B\{c^{\mathbf{P}}\} \vdash c^{\mathbf{O}}}\,\supset^{\mathsf{L}}}{\Gamma, A \supset B\{c^{\mathbf{P}}\} \vdash c^{\mathbf{O}}}\,\mathsf{C}$
$\Gamma, B \vdash A$	$\mathcal{S}$ contains no moves on B	$\dfrac{\overset{\mathfrak{D}_{\mathcal{S}}\,\Vert}{\Gamma \vdash A}}{\Gamma, B \vdash A}\,\mathsf{W}$
$\Box\Gamma \vdash \Box A$	at least one move of each formula in $\Box\Gamma$ occurs in $\mathcal{S}$	$\dfrac{\overset{\mathfrak{D}_{\mathcal{S}}\,\Vert}{\Gamma \vdash A}}{\Box\Gamma \vdash \Box A}\,\mathsf{K}^{\Box}$
$\Box\Gamma, \Diamond^{\mathbf{P}} B \vdash \Diamond^{\mathbf{O}} A$	at least one move of each formula in $\Box\Gamma, \Diamond^{\mathbf{P}} B$ occurs in $\mathcal{S}$	$\dfrac{\overset{\mathfrak{D}_{\mathcal{S}\setminus\{\Diamond^{\mathbf{O}}, \Diamond^{\mathbf{O}}\Diamond^{\mathbf{P}}\}}\,\Vert}{\Gamma, B \vdash A}}{\Box\Gamma, \Diamond B \vdash \Diamond A}\,\mathsf{K}^{\Diamond}$

Figure 4.4: Sequentialization procedure

4.9 Bonus: game semantics for CD

The results presented in this chapter can be straightforwardly extended to the constructive modal logic CD, which is obtained by extending CK with the modal axiom d shown below left:

$$\mathsf{d}\colon \Box A \supset \Diamond A \qquad\qquad \frac{\Gamma \vdash A}{\Box\Gamma \vdash \Diamond A}\,\mathsf{D}$$

A sound and complete (cut-free) sequent system for this logic can be obtained by adding the sequent rule above on the right to the sequent system for CK.

In order to define **WIS** capturing proofs in CD we need some additional definitions.

Definition 4.13. *Let $\mathcal{S}$ be a* **WIS** *over an arena $\mathcal{G}$. We say that $\mathcal{S}$ is* CD-batched *if it is* atomic, *that is, the views in $\mathcal{S}$ contains only atomic vertices,* linked, *and if for each modal vertex $w^{\mathbf{O}}$ occurring in the address of a move in $\mathcal{S}$ the following conditions are fulfilled:*

i if $w^{\mathbf{O}} \in V^{\Box}_{\mathcal{G}}$ and $v \xrightarrow{\mathcal{G}_{\mathsf{p}}} w$ for a $\mathsf{p} \in \mathcal{S}$, then $v \in V^{\Box}_{\mathcal{G}}$;

ii if $w^{\mathbf{O}} \in V^{\Diamond}_{\mathcal{G}}$, then there is at most a $u \in V^{\Diamond}_{\mathcal{G}}$ in the set $\{v \in V_{\mathcal{G}} \mid v \xrightarrow{\mathcal{G}_{\mathsf{p}}} w$ for a $\mathsf{p} \in \mathcal{S}\}$.

Note that the first condition is the same first condition from Definition 4.11. The reason why we do not need the information about the diamonds in the strategies for CD depends on a property of the logic (see [5, Theorem 2]). The idea is that an instance of weakening can permute below $\mathsf{K}^{\Diamond}$-rules, transforming it into an instance of the D-rule, as shown below (while in CK the information about the left-hand side diamond must be kept in some way):

$$\dfrac{\dfrac{\overset{\mathcal{D}\,\|}{\Gamma \vdash A}}{B, \Gamma \vdash A}\,\mathsf{W}}{\Diamond B, \Box\Gamma \vdash \Diamond A}\,\mathsf{K}^{\Diamond} \quad \rightsquigarrow \quad \dfrac{\dfrac{\overset{\mathcal{D}\,\|}{\Gamma \vdash A}}{\Box\Gamma \vdash \Diamond A}\,\mathsf{D}}{\Diamond B, \Box\Gamma \vdash \Diamond A}\,\mathsf{W}$$

We then define a CD-**WIS** as a CD-batched **WIS**. This allows to extend theorem 4.4 with no effort, that is

Theorem 4.5. *The* CD-**WIS***s form a full-complete denotational semantics for* CD.

4.10 Conclusion and Future Work

In this chapter we have defined a game semantics for the constructive modal logic CK and have shown how it can be extended for the logic CD. We have proved full completeness and compositionality of our winning strategies, and thus have shown that our model provides a a full complete denotational semantics for CK and CD. The two game semantics that we have defined provides an inferentialist alternative to the model theoretic semantics of the two constructive modal logics [85]. Semantics here is defined in a pragmatist way: it is given by a certain way of playing intuitionist games. In particular, the games for CK and CD can be seen as intuitionistic games in which the Proponent is forced to play inside *contexts* (batches). These contexts are opened by Opponent whenever he plays a move that is under the scope of a positive modality. The proponent can play only moves that are in the currently active context, and only the Opponent can switch to an already opened context or open a new one.

In order to make our alternative approach more attractive we are currently investigating the possibility of extending our semantics to the logics CT and CS4, that are obtained by adding the modal axioms

$$\mathsf{T}: (A \supset \Diamond A) \wedge (\Box A \supset A) \qquad \text{and} \qquad \mathsf{4}: (\Diamond\Diamond A \supset \Diamond A) \wedge (\Box A \supset \Box\Box A)$$

However, the problem that arises is that for these logics also the $\Box$ should be allowed as move in order to keep track of the rules for T and 4. However, the **P**-determinism of winning strategies depends on the fact that atoms and diamonds are paired by the rules which introduce them. This means that when boxes are allowed as moves, determinism cannot hold. We have to leave this issue for future work.

It is worth noticing that our result is strongly related to the game semantics for *light linear logic* as given in [113]. In fact, light linear logic can be seen, in terms of proof system, as a $\Diamond$-free variant of LCK with two modalities ! and § behaving like $\Box$, where weakening and contraction rules are restrained to !-formulas. This suggests the possibility of using our approach to define a new notion of winning innocent strategies capturing the proofs in intuitionistic linear logic.

Finally, we conjecture the existence of a one-to-one correspondence between our CK-**WIS**s and the λ-calculi for constructive modal logics [11, 12]. This investigation will also be object of future research.

Part III

Natural Language Applications of Proof Theory

Chapter 5

Type Logical Grammars: a result about the syntactic-semantic interface.

Abstract
A natural question in categorial grammar is the relation between a syntactic analysis and its logical form, i.e. the logical formula obtained from this syntactic analysis, once provided with semantic lambda terms. More precisely, do different syntactic analyses fed with equal semantic terms, lead to different logical form? We shall show that when this question is too simply formulated, the answer is "no" while with some constraints on semantic lambda terms the answer is "yes". The results of this Chapter already appears in [23]

5.1 Introduction

Type-logical grammars are a family of frameworks for the analysis of natural language based on logic and type theory. Type-logical grammars are generally fragments of intuitionistic linear logic, with the Curry-Howard isomorphism of intuitionistic logic serving as the syntax-semantics interface. Figure 5.1 shows the standard architecture of type-logical grammars.

1. given some input text, a *lexicon* translates words into formulas, resulting in a judgment in some logical calculus, such as the Lambek calculus or some variant/extension of it,

2. the grammaticality of a sentence corresponds to the provability of this statement in the given logic (where different proofs can correspond to different interpretations/readings of a sentence),

3. there is a forgetful mapping from the grammaticality proof into a proof of multiplicative, intuitionistic linear logic,

4. by the Curry-Howard isomorphism, this produces a linear lambda-term representing the derivational meaning of the sentence (that is, it provides instructions for how to compose the meanings of the individual words),

5. we then substitute complex lexical meanings for the free variables corresponding to the lexical entries to obtain a representation of the logical meaning of the sentence,

6. finally, we use standard theorem proving tools (in first- or higher-order logic) to compute entailment relations between (readings of) sentences.

This chapter is divided into two parts. In the first part of the chapter, we introduce the various items that compose a type logical grammar based on the product free Lambek calculus. We explain how one can use such items to obtain, from a syntactic parsing of a natural language sentence, a logical formula.

In the second part of the chapter we will deal with a natural problem in type logical grammar: the relation between a syntactic analysis and its logical form, i.e. the logical formula obtained from this syntactic analysis, once provided with semantic lambda terms. More precisely, do different syntactic analyses fed with equal semantic terms, lead to different logical form? We shall show that when this question is too simply formulated, the answer is “no” while with some constraints on semantic lambda terms the answer is “yes”.

Organization of the chapter In section 5.2 we introduce the Lambek Syntactic calculus. More precisely we introduce a natural deduction system for this logic and then we show how we can analyze natural language phrases by means of Lambek-deduction, moreover we briefly introduce the language of multisorted first order logic and show how formulas of multisorted first order logic can be written in the simply typed λ-calculus. In section 5.3 we show how we can obtain a logical representation of a natural language sentence from a lambek proof using the Curry-Howard isomorphism. In the following section (section 5.4) we introduce the question that we are interested in “does different syntactic parsing of the

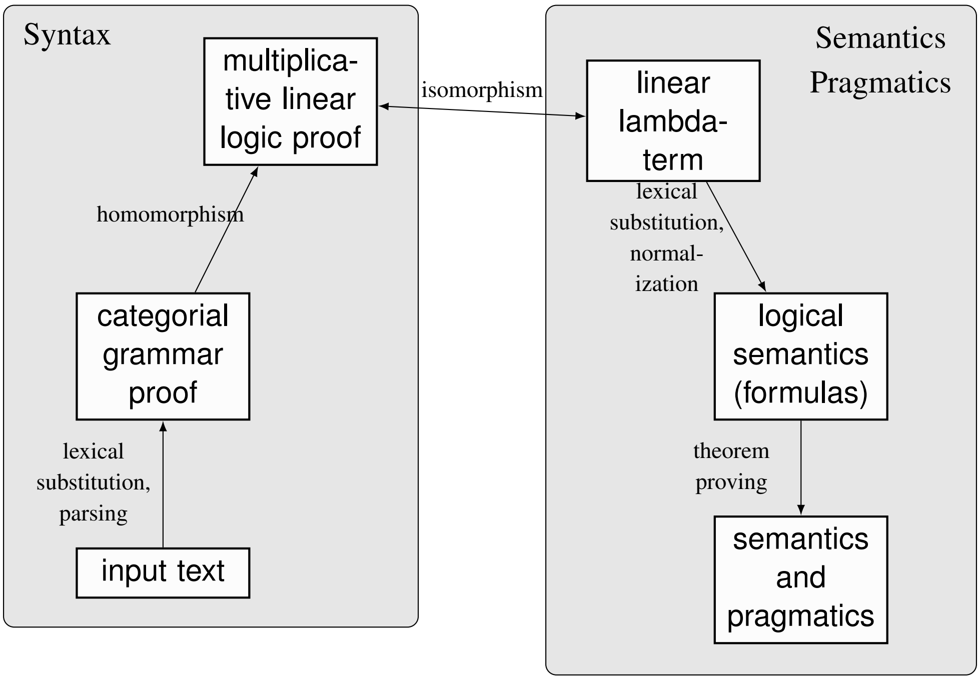


Figure 5.1: The standard architecture of type-logical grammars

same sentence give rise to different logical representation of the same sentence?". We show that this question —if naively formulated— admits several negative answers. In section 5.5 we introduce the notion of *dominance* between atomic terms in a lambda-term. By using this notion, we can reformulate the aforementioned question and obtain a positive answer to it. Section 5.6 concludes this chapter by sketching some possible extension of our result.

5.2 The Lambek Syntactic calculus

In this section we introduce the Lambek Syntactic Calculus. The Lambek calculus [90] is a logic developed for analyzing natural language. For Lambek grammars, the grammaticality of a sentence corresponds to the derivability of a statement in the logical calculus, given a lexicon mapping the words of the sentence to formulas. Lambek calculus proofs correspond to logical formulas in a simple and systematic way [135].

The language of the Lambek calculus is specified as follows: given a set of primitive categories (or primitive formulas) $\mathcal{P} = \{s, n, np, pp, \ldots, \}$ sentence, noun, noun phrases, prepositional phrases, Lambek categories (or formulas) are constructed according to the following grammar:

$$Lp = \mathcal{P} \mid (Lp \backslash Lp) \mid (Lp/Lp)$$

as usual, we will use roman upper case letters from the beginning of the alphabet A, B, C, D etc. to denote arbitrary categories. The intuitive meaning of $A \backslash B$ and B/A is the following: an expression $A \backslash B$ is looking for an expression of type A on its left to produce an expression of type B. Similarly, an expression B/A is looking for an expression of type A on its right to produce an expression of type B.

The Natural Deduction rules for the Lambek calculus are shown in figure 5.2. The elimination rules $/E$ and $\backslash E$ are simply directional versions of the modus ponens rule. The major premise of an elimination rule is the category whose connective is eliminated by the rule application. The introduction rules $/I$ and $\backslash I$ require us the discharge exactly one occurrence of the A category. The introduction rules have the additional condition that the discharged category A must be the leftmost (resp. rightmost) free hypothesis in the sub-proof ending in B for the $\backslash I$ rule (resp. the $/I$ rule) *and* that there must be at least one other category not already discharged.

$$\frac{\begin{matrix}\vdots & \vdots \\ A & A\backslash B\end{matrix}}{B}\,[\backslash E] \qquad\qquad \frac{\begin{matrix}[A]\cdots\cdots \\ \vdots \\ B\end{matrix}}{A\backslash B}\,[\backslash I]$$

$$\frac{\begin{matrix}\vdots & \vdots \\ B/A & A\end{matrix}}{B}\,[/E] \qquad\qquad \frac{\begin{matrix}\cdots\cdots[A] \\ \vdots \\ B\end{matrix}}{B/A}\,[/I]$$

Figure 5.2: Natural deduction rules for L

Remark that contrarily to what happens in a natural deduction calculus for intuitionistic logic (see section 1.8 of chapter 1) there is no need of naming the formula occurrence that is discharged by an introduction rule. As a consequence the formal structure of a deduction is a plain tree with leaves labeled with formulas and with nodes labelled by rules: such a plain tree is enough to reconstruct the deduction, i.e. which hypothesis are free or not and which hypothesis is discharged by which rule.

Let $\mathfrak{D}$ be a derivation. An introduction rule for a connective $\star$ that introduces the major premises for an elimination rule for the same connective $\star$ is called a *detour* (or redex) in the derivation $\mathfrak{D}$. A derivation with no detour is called normal. There are reduction rules to eliminate detours in a derivation. In the case of the Lambek calculus the reduction rules are the following:

$$[/E]\,\frac{\begin{matrix}\vdots\mathfrak{D}_2 \\ A\end{matrix} \qquad \dfrac{\begin{matrix}[A]\cdots\cdots \\ \vdots\mathfrak{D}_1 \\ B\end{matrix}}{A\backslash B}\,[/I]}{B} \quad\rightsquigarrow\quad \begin{matrix}\vdots\mathfrak{D}_2 \\ A\cdots\cdots \\ \vdots\mathfrak{D}_1 \\ B\end{matrix}$$

$$
[/E]\,\dfrac{\dfrac{\begin{array}{c}\cdots\cdots[A]\\ \vdots\mathfrak{D}_1\\ B\end{array}}{B/A}\,[/I] \qquad \begin{array}{c}\vdots\mathfrak{D}_2\\ A\end{array}}{B} \qquad\rightsquigarrow\qquad \begin{array}{c}\vdots\mathfrak{D}_2\\ \cdots\cdots A\\ \vdots\mathfrak{D}_1\\ B\end{array}
$$

An *expansion* of a derivation $\mathfrak{D}$ consists in the replacement of a sub-derivation $\mathfrak{D}'$ of $\mathfrak{D}$ by another sub-derivation according to one of the two following rules

$$
\begin{array}{c}\vdots\mathfrak{D}\\ A\backslash B\end{array} \qquad\hookrightarrow\qquad \dfrac{\dfrac{[A] \qquad \begin{array}{c}\vdots\mathfrak{D}\\ A\backslash B\end{array}}{B}\,[\backslash E]}{B\backslash A}\,[\backslash I]
$$

$$
\begin{array}{c}\vdots\mathfrak{D}\\ B/A\end{array} \qquad\hookrightarrow\qquad \dfrac{\dfrac{\begin{array}{c}\vdots\mathfrak{D}\\ B/A\end{array} \qquad [A]}{B}\,[/E]}{B/A}\,[/I]
$$

the expansion can create new redexes in a derivation. Therefore, we want to allow them only in position where no new redex are created. A formula occurrence A is said to be in end position in a derivation $\mathfrak{D}$ iff it is either the conclusion of $\mathfrak{D}$ or it is not the major premise of an elimination rule. A formula occurrence A is in *minimal* position iff either

- A is the conclusion of an en elimination rule and the premise of an introduction rule for the same connective.
- A is end position and it is also the conclusion of an elimination rule.

Finally, we say that a derivation $\mathfrak{D}$ is in *long normal form* iff its is normal and no expansion rule at minimal position are possible without creating a detour.

5.2.1 Lambek Calculus and Grammar

In order to test if a sequence $w_1, w_2, \ldots w_n$ of words forms a sentence that is grammatically correct we assign to each word in the sequence a *set* of Lambek categories i.e., there is a

$$
\begin{aligned}
lex(John) &= np & lex(ran) &= np\backslash s\\
lex(Mary) &= np & lex(slept) &= np\backslash s\\
lex(the) &= np/n & lex(ate) &= (np\backslash s)/np\\
lex(report) &= n & lex(wrote) &= (np\backslash s)/np\\
lex(student) &= n & lex(everyone) &= s/(np\backslash s)\\
lex(pizza) &= n & lex(someone) &= (s/np)\backslash s\\
lex(who) &= (n\backslash n)/(np\backslash s) & lex(every) &= (s/(np\backslash s))/n\\
lex(whom) &= (n\backslash n)/(s/np) & lex(some) &= ((s/np)\backslash s)/n
\end{aligned}
$$

Table 5.1: Lambek calculus lexicon

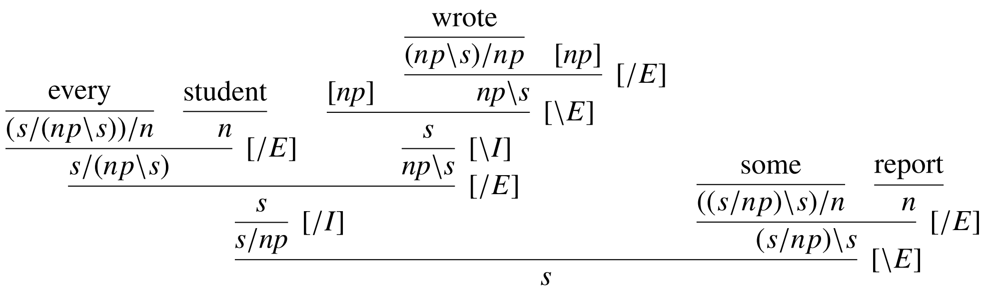


Figure 5.3: Proof of "every student wrote some report"

function *lex* that assigns to each w_i for $i \in \{1, \dots n\}$ a set of Lambek types. In the following if $lex(w)$ is a singleton set $\{A\}$ we will simply write $lex(w) = A$. Table 5.1 gives a simple Lambek calculus lexicon. Some words, like "John" are assigned atomic categories (here np for 'noun phrase'). Similarly, "student" is assigned an atomic category n for 'noun'. The article "the" is assigned a complex category np/n indicating it is looking for a noun n (such as "student") to its right to form a noun phrase. Similarly, "slept" is assigned a complex category $np\backslash s$, indicating is looking for a noun phrase np (such as "the student" or "John") to form a sentence.

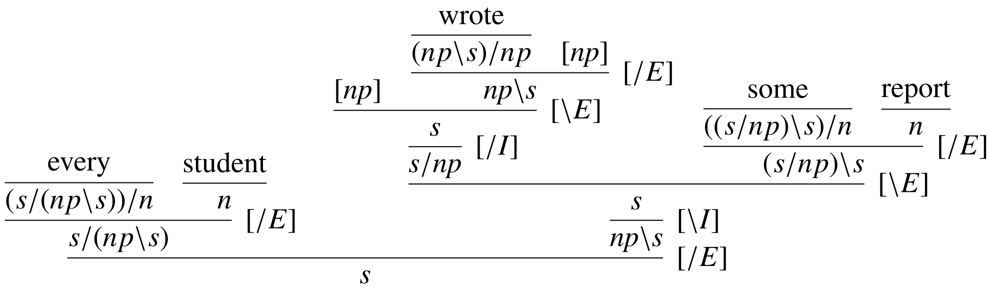


Figure 5.4: Second proof of "every student wrote some report"

Figure 5.3 shows an example proof for the sentence "every student wrote some report". Each non-discharged category in the proof corresponds to a word in the lexicon, and we have written this word above the category. Note that this is a long normal form derivation: we have an elimination rule immediately followed by an introduction rule for the np hypothesis.

It is fairly easy, given a lexicon, to enumerate all long normal form proofs for a sentence [108], and with the lexicon of Table 5.1 the sentence "every student wrote some report" has exactly two such proofs, with the second shown in Figure 5.4.

These two proofs correspond to the two readings of the sentence in a Montague-style treatment of quantification [106]: one where the existential quantifier "some" has wide scope over the universal quantifier "every", and one where "every" outscopes "some".

5.2.2 Multisorted logic and lambda calculus

In this section we briefly introduce the language of multisorted logic and how formulas of multisorted logic can be coded in the simply typed λ-calculus. Given a non-empty set S_0 of basic sorts, sorts are defined by the following grammar

$$S := \{t\} \,|\, S_0 \,|\, S \to S$$

Let $\Sigma = (\mathcal{P}, \mathfrak{F}, S_0)$. where S_0 is an at most countable set of sorts for individuals, $\mathcal{P}$ is a set of predicate variables and $\mathfrak{F}$ is a set of function symbol. The three sets are disjoints and there is a function α that assign a sort $\sigma = s_1 \to s_2 \to \cdots s_n \to t$ (where each

s_i belongs to S_0 and t is the sort of truth values) to each predicate variable, and a sort $\rho = s_1 \to s_2 \to \cdots s_n$ to each function symbols. If $\mathbf{k} \in \mathfrak{F}$ and $\alpha(\mathbf{k}) = s$ with $s \in S_0$ then $\mathbf{k}$ will be called and individual constant. In the same way if $R \in \mathcal{P}$ and $\alpha(P) = t$ then P will be called a predicate constant. Suppose that there is a countable set of individual variables symbols for each $s \in S_0$. We will denote a variable symbol with sort s by x^s. Terms of the language are defined by the following grammar:

$$t := \; x^s \; | f(t_1, \dots t_n)$$

let $\neg, \wedge, \vee, \supset, \forall, \exists$ be the symbols for the usual connective and quantifiers of first order logic, $\perp$ and $\top$ be predicate constants. The set At of atomic formulas is the smallest set containing formulas of the form $P(t_1, \dots t_n)$, where P is a predicate variable with sort $s_1 \to s_2 \to \cdots \to s_n \to t$ and the t_i are terms of the appropriate sort. Formulas are specified by the following grammar:

$$\mathcal{F} := At \mid \neg\mathcal{F} \mid \mathcal{F} \wedge \mathcal{F} \mid \mathcal{F} \vee \mathcal{F} \mid \mathcal{F} \supset \mathcal{F} \mid \forall x^s \mathcal{F} \mid \exists x^s \mathcal{F} \mid$$

The notion of free (resp. bound) variables of a term (resp. formula) will be the usual ones, as well as the notion of subformula, Gentzen-subformula and positive/negative subformula of a formula.

Let $\mathcal{L}$ be a multisorted first order language over the signature $(P, \mathfrak{F}, S_0)$. To represent formulas of this first order multi-sorted language as terms of the simply typed λ-calculus, we need first to assume that the set of base sorts of $\mathcal{L}$ and the set of base types of our simply typed calculus are the same. We then need to assume that

- we have a λ-constant $\neg : t \to t$ as well as three constants $\wedge, \vee \supset$ each one of type $t \to (t \to t)$
- for each base type s we have two constants $\forall$ and $\exists$ both of type $(s \to t) \to t$
- for every relational symbol $R \in \mathcal{R}$ such that $\alpha(R) = \rho$ a λ-constant $R : \rho$
- for every functional symbol $f \in \mathfrak{F}$ such that $\alpha(f) = \sigma$ a λ-constant $f : \sigma$

Example 5.1. *Suppose that h is the sort of human being, nh the sort of non-human being, that donkey is a predicate with sort $nh \to t$, farmer a predicate with sort $h \to t$ and that owns is a predicate with sort $h \to (nh \to t)$. The following formula of multisorted first order logic*

$$\forall\, x^{h}\, (farmer\,(x) \wedge (\exists\, y^{nh}\, (donkey\,(y) \wedge owns\,(x, y)))$$

can be represented by the following λ-term

$$\forall^{(h\to t)\to t}(\lambda x^{h}(\wedge\,(farmer^{h\to t}\;\; x)(\exists^{(nh\to t)\to t}(\lambda y^{nh}(\wedge(donkey^{nh\to t}\;\; y)(owns^{nh\to(h\to t)}\; x\, y))))$$

In general one can always transform a formula F of multisorted first order logic into a λ-term M_F. Moreover, one can show the following proposition by induction on M:

Proposition 5.1. *Let M be a eta-long β-normal λ-term of type s_i with free variables x_k of type s_{i_k}, then M correspond to a term of multisorted logic of type s_i whose free variables are of type x_{i_k}*

We can also show that the following proposition holds by induction on M:

Proposition 5.2. *Let M be a eta-long β-normal λ-term of type t, then M corresponds to a formula F of multisorted first order logic; moreover F and M have the same free variables of type s_i for $s_i \in S$*

the detailed proofs of the two above proposition can be found in the third chapter of [108].

5.3 From Lambek to logic

There is a very direct way to turn the two proofs of the previous section (proofs 5.3 and 5.4) into (lambda term representations) of the logical formulas representing the two possible meanings of the sentence. There is a division of labor here: the Lambek calculus proof specifies how the word in the lexicon are combined (in the form of a linear lambda term) and the lexical entry for each word specifies a (not necessarily linear) lambda term corresponding to the meaning of the word. To obtain a linear lambda term from out of a lambek derivation, one usually pass through the intuitionistic fragment of multiplicative linear logic

5.3.1 From Lambek to imll

Formulas of multiplicative intuitionistic linear logic (imll for short) are constructed from an at most countable set of base, or primitive, types using the connective $\multimap$ (linear implication).

Proofs in imll calculus correspond to linear lambda terms. Table 5.2 shows the natural deduction rules for imll together with term assignment for the proof. The elimination rule has the condition that M and N do not share variables. The introduction rule has the condition that exactly one occurrence of the formula A (with variable x) is discharged. Remark that we need to index the discharged hypothesis an the introduction rule that discharge it i.e., natural deduction proofs for imll are trees decorated with this additional information. We describe a very simple way to turn a Lambek proof into a linear λ-term. Assume that the base type of the lambek calculus are n, np and s. Assume moreover that the only base types of imll are e (the type of entities) and t (the type of truth values). The mapping $.^*$ from lambek categories to imll formulas is specified as follows:

$$n^* = e \multimap t \qquad np^* = e \qquad s^* = t$$

$$(A\backslash B)^* = (B/A)^* = A^* \multimap B^*$$

the mapping $.^*$ has the property that it does not only translate Lambek-types to imll types, but also Lambek derivation rules (and therefore derivations) to imll derivation rules (and derivations).

The proofs in Figures 5.3 and 5.4 correspond to the lambda terms given in 5.1 and 5.2 respectively.

$$\dfrac{\begin{matrix}\vdots \\ N : A\end{matrix} \qquad \begin{matrix}\vdots \\ M : A \multimap B\end{matrix}}{MN : B}\,[\multimap E] \qquad\qquad \dfrac{\begin{matrix}[x : A] \\ \vdots \\ M : B\end{matrix}}{\lambda x^A\, M : A \multimap B}\,[\multimap I]$$

Table 5.2: Natural deduction rules for multiplicative intuitionistic linear logic.

$$(w_4\, w_5)(\lambda y((w_1\, w_2)\,(\lambda x((w_3\, y)\, x)))) \tag{5.1}$$

$$(w_1\, w_2)(\lambda x((w_4\, w_5)\,(\lambda y((w_3\, y)\, x)))) \tag{5.2}$$

Remark that both terms are of type t, the free variables $w_1, \ldots w_5$ of the two terms corresponds to the words *every*, ..., *report* of the two analyzed sentences, and the two bound variables x and y corresponds to the discharged hypothesis of the Lambek derivations. Although we use the Lambek calculus as an example in this chapter, most type-logical grammars (including the multi-modal non associative Lambek calculus) have a similar forgetful mapping from their logical connectives (and the corresponding derivation rules) to imll.

5.3.2 From imll to logical formulas

Finally, to obtain a representation of the meaning of the sentence, we substitute the lexical meaning for each word. Following Montague, we leave some words unanalyzed, using the constant *student* as the meaning of the word "student", and similarly for "wrote" and "report", which are assigned the meaning *write* and *report* respectively. The interesting words in this example are "every" and "some". Using the constants $\forall$ and $\exists$, both of type $(e \rightarrow t) \rightarrow t$, to represent the universal and the existential quantifier, and the constants $\wedge$, $\vee$ and $\supset$ of type $t \rightarrow (t \rightarrow t)$ to represent the binary logical connectives, we can assign the following lambda term to "every" and to "some":

$$\lambda P \lambda Q \forall(\lambda x.(\supset (P\, x))(Q\, x)) \tag{5.3}$$

$$\lambda P \lambda Q \exists(\lambda x.(\wedge (P\, x))(Q\, x)) \tag{5.4}$$

substituting the lexical terms for each of the corresponding variables derived terms 5.1 and 5.2 produces the following two terms.

$$(\lambda P \lambda Q \exists(\lambda z.(\wedge(P\, z))(Q\, z))\; report)(\lambda y((\lambda R \lambda S\, \forall(\lambda v.(\supset (R\, v))(S\;\, v))\; student)\,(\lambda x((write\, y)\, x)))) \tag{5.5}$$

$$(\lambda R \lambda S\, \forall(\lambda v.(\supset (R\, v))(S\;\, v))\; student)(\lambda x((\lambda P \lambda Q \exists(\lambda z.(\wedge(P\, z))(Q\, z))\; report)\,(\lambda y((write\, y)\, x)))) \tag{5.6}$$

these terms normalize to:

$$\exists(\lambda z.(\wedge(report\, z))(\forall(\lambda v.(\supset (student\, v)((write\, z)\, v) \tag{5.7}$$

$$\forall(\lambda v.(\supset (student\, v))(\exists(\lambda z.(\wedge(report\, z))((write\, z)\, v) \tag{5.8}$$

in more standard logical notation, these terms represent the following two formulas:

$$\exists z(report(z) \wedge \forall v.(student(v) \supset write(v, z))) \tag{5.9}$$

$$\forall v(student(v) \supset \exists z.(report(z) \wedge write(v, z))) \tag{5.10}$$

5.4 Syntactic terms and logical readings

In the previous sections we have sketched how, by using type theory, we can produce a logical formula from a given string of lexicalyzed word. In particular, we have shown that the same string of words can correspond to different logical formulas: from the same string of lexicalyzed words we can produce different lambek derivations that will give rise to different linear λ-terms. In the running examples that we have developed through the chapter, we have two Lambek calculus proofs producing two different readings. Now, while it is the case that two different natural deduction proofs for the Lambek calculus always produce two different linear lambda terms (terms like 5.1 and 5.2), the question which will interest us in the rest of this chapter is the following: when can we guarantee that different Lambek calculus proofs (or proofs in another system of type-logical grammar) produce different meanings? By this, we mean different meaning in the sense of different lambda terms after lexical substitution and normalization, and not terms representing logical formulas which are not logically equivalent. This distinction is obvious when we replace our example sentence by "some student wrote some report". Here we still produce two different terms, where the two existential quantifiers have different scope with respect to each other. However, these two terms correspond to logically equivalent formulas. In order to make the question we intend to ask clearer, let us introduce some definitions

Definition 5.1 (Syntactic λ-term)**.** *A syntactic λ-term is a β-normal, simply-typed linear λ-term with one occurrence of each free variable in $w_1, \ldots, w_n$ with $n > 0$ — those free variables are the words of some analyzed sentence.*

We focus on linear lambda terms instead of the more restricted class of lambda terms corresponding to Lambek calculus proofs because:

1. linear lambda terms have a simple inductive definition,

2. semantic phenomena like quantifier scope can be captured using linear lambda terms but not using lambda terms corresponding to Lambek calculus proofs; a simple counting argument shows that there are not enough Lambek calculus proofs to generate the $n!$ readings for a sentence with n quantifiers [109],

3. many modern type-logical grammars also produce linear lambda terms for their syntactic proofs, thereby making our results more generally applicable.

Definition 5.2. *A semantic lambda term is a β-normal simply typed λ-term in which some constants occur.*

5.4.1 The problem

Assume that the sentence $w_1 \cdots w_n$ has two syntactic λ-term P_1 and P_2 associated to it. When replacing each v_i (a free variable representing w_i in the syntactic λ-term) by the associated lexical lambda term N_i (non linear, with constants) in P_1 and in P_2 does beta reduction give different lambda terms (i.e. logical formulas), i.e. does one have

$$P_1[N_1/v_1 \cdots N_n/v_n] \overset{\beta}{\neq} P_2[N_1/v_1 \cdots N_n/v_n] \quad ?$$

where $\overset{\beta}{=}$ is the symmetric, transitive and reflexive closure of $\overset{\beta}{\rightsquigarrow}_1$

5.4.2 Some counterexamples

In its most general form, the answer to our question is negative:

Proposition 5.3. *There exist P_1, P_2 two syntactic λ-terms both of type σ and with the same free variables $w_1, w_2 \ldots w_n$, and and there exist $N_1, N_2 \ldots, N_n$ n semantic λ-terms such*

$$P_1 \overset{\beta}{\neq} P_2 \quad \textit{and} \quad P_1[N_1/w_1 \cdots N_n/w_n] \overset{\beta}{=} P_1[N_1/w_1 \cdots N_n/w_n]$$

Proof. Take

$$P_1 \equiv w_1((w_2 w_3) w_4)$$

$$P_2 \equiv w_1((w_2 w_4) w_3)$$

where $w_1 : t \to t$, $w_2 : e \to (e \to t)$ and w_3, w_4 are both of type e. Moreover take

$$N_1 \equiv \lambda y.k_1 \quad N_2 \equiv \lambda x_1 \lambda x_2((k_2 x_1)x_2)) \quad N_3 \equiv k_3 \quad N_4 \equiv k_4$$

where $k_1 : t$, $y : t$, $k_2 : e \to (e \to t)$ and t_3, t_4, x_1 and x_2 are of type e. Make the following substitution.

$$P_1[N_1/w_1, N_2/w_2, N_3/w_3, N_4/w_4] \qquad P_2[N_1/w_1, N_2/w_2, N_3/w_3, N_4/w_4]$$

Both terms reduce to k_1 □

This proposition (counter example to our question) holds because β-reduction can delete i.e., when we have $M := \lambda x M'$ in which $x \notin Fv(M')$ MN reduces in one step to M' for all N. Therefore, is it essential to exclude such cases if one hopes to give a positive answer to the claim.

However, people who experimented categorial lexicons have probably noticed that semantic lambda terms are λ_I-terms i.e. β reduction never erase any subterm. One may wonder whether semantic lambda terms could be asked to be linear. This would be a too severe constraint, since many common semantic lexical entries such as generalized quantifiers cannot be expressed by linear lambda terms.

$$every : \lambda P \lambda Q \forall(\lambda x.(\supset (P\,x))(Q\,x))$$

Definition 5.3. *A* simple semantic lambda term *is a β-normal η-long λ_I-term M*[1] *with constants whose head symbol is a constant i.e.,*

$$M := \lambda z_1, \cdots z_n k N_1 N_2 \cdots N_m$$

where k is a constant.

We want to preserve the difference between syntactic terms throughout β reduction. Therefore, we just consider the case when free variables in the syntactic terms are substituted with semantic λ-terms with constants as head term. This restriction is severe but still allows the capture of some non-trivial differences between syntactic analysis of the same sentence e.g. scope ambiguity for quantifiers.

[1] Remark that β-normal η-long terms have been defined in chapter 1 (p.70) as well as the notion of λ_I-term (definition 1.19)

Proposition 5.4. *There exist P_1, P_2 two syntactic λ-terms both of type σ and with the same free variables $w_1, w_2, \ldots w_n$, and there exist $N_1, N_2 \ldots, N_n$ n simple semantic λ-terms such that*

$$P_1 \overset{\beta}{\neq} P_2 \quad \text{and} \quad P_1[N_1/w_1, \ldots N_n/w_n] \overset{\beta}{=} P_2[N_1/w_1, \ldots, N_n/w_n]$$

Proof. take

$$P_1 \equiv ((w_1 w_2) w_3)$$

$$P_2 \equiv ((w_1 w_3) w_2)$$

with $w_1 : e \to (e \to t)$ and w_2, w_3 of type e

$$N_1 \equiv \lambda x_1 \lambda x_2((k_1 x_1) x_2) \quad N_2 \equiv k_2 \quad N_3 \equiv k_2$$

with $k_1 : e \to (e \to t)$, x_2, x_2, k_2 of type e and make the following substitution

$$P_1[N_1/w_1, N_2/w_2, N_3/w_3] \quad P_2[N_1/w_1, N_2/w_2, N_3/w_3]$$

After β-reduction the two terms become β-equal.

□

This counterexample shows that we should also require, at least, that the n lexical lambda terms all have a different head-constant. Unfortunately — even with this restriction — if one formalizes the notion of difference between the two syntactic analyses of the sentence $w_1 \cdots w_n$ in terms of β-difference between syntactic terms, one is doomed to failure.

Proposition 5.5. *There exists P_1 and P_2 two syntactic terms, both of type σ and with the same free variables $w_1, \ldots, w_n$. There exists $N_1, N_2 \ldots, N_n$ n simple semantic lambda terms such that whenever $i \neq j$ then the head-constant of N_i is different from the head-constant of N_j. We have that*

$$P_1 \overset{\beta}{\neq} P_2 \quad \text{and} \quad P_1[N_1/w_1, \ldots, N_n/w_n] \overset{\beta}{=} P_1[N_1/w_1, \ldots, N_n/w_n]$$

Proof. take

$$P_1 \equiv w_1(\lambda x \lambda y((w_2 x) y))$$

$$P_2 \equiv w_1(\lambda y \lambda x((w_2 x) y))$$

where $x : e, y : e, w_2 : e \to (e \to t), w_1 : (e \to (e \to t)) \to t$. Take

$$N_1 \equiv \lambda P(k_1((Px)x)) \quad N_2 \equiv (\lambda z \lambda y((k_2 z)y))$$

where $P : (e \to (e \to t)) \to t$, $k_1 : t \to t$, $k_2 : e \to (e \to t)$ and x, z, y are of type e. And make the following substitution

$$P_1[N_1/w_1, N_2/t_2] \quad P_2[N_1/w_1, N_2/w_2]$$

After β-reduction the two terms become β-equal. □

5.5 Dominance

In the last section, we have seen that the question has a negative answer if the difference between the two syntactic analyses is just syntactic difference (up to the renaming of bound variables). By consequence, we can attack the problem by using at least two different strategies.

Strategy 1 Refining our notion of syntactic lambda-terms. We know that not all linear lambda terms have a corresponding Lambek calculus derivation, because Lambek calculus is not commutative and therefore is proofs correspond to a proper subset of the linear lambda terms. Similarly, most other type-logical grammars produce generate a proper subset of the linear lambda terms as well. Although such proofs are possible, it may be hard to give a precise and succinct statement of these classes of lambda terms (for the Lambek calculus, we could follow the ideas of [142] for a directional lambda calculus)

Strategy 2 Define a stronger notion of difference between syntactic analyses and the resulting lambda terms.

The first strategy seems promising: the two syntactic terms $P_1 = w_1(\lambda y \lambda x((w_2 x)y))$ $P_2 = w_1(\lambda y \lambda x((w_2 x)y))$ of Proposition 5.4 cannot correspond to two Lambek calculus parses obtained by the same category assignment to the free variables w_1 and w_2. However, it is really difficult to exactly characterize the subclass of typed linear lambda terms that corresponds to derivations in the Lambek-calculus. The translation from the latter to the former is not injective, and thus some information that could be relevant is lost.

The second strategy could be pursued only if the new notion of difference captures some interesting differences between syntactic analysis of the same sentence e.g., scope ambiguity for quantifiers. We are going to pursue this path by defining a notion of *dominance* between occurrences of atoms in λ-terms.

Definition 5.4 (leftmost-innermost subterm)**.** *The leftmost-innermost subterm of a term M is defined as follows.*

- *If the term M is atomic the leftmost-innermost subterm is M itself.*
- *If the term M is an application M_1M_2, the leftmost innermost subterm of M is the leftmost innermost subterm of M_1.*
- *If the term M is an abstraction $\lambda x.M_1$, the leftmost innermost subterm of M is the leftmost innermost subterm of M_1*

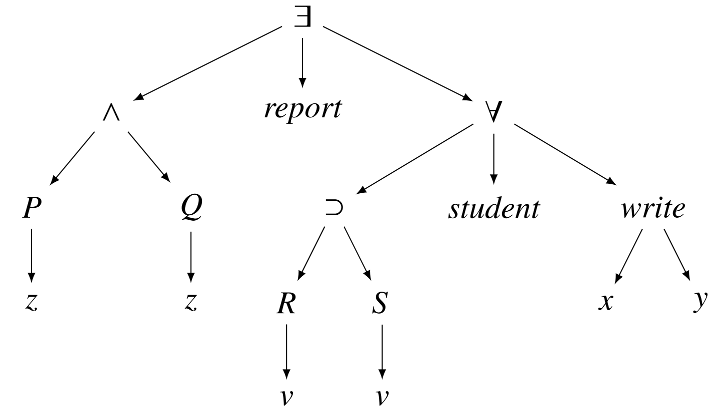
Proposition 5.6. *The leftmost-innermost subterm of a term M is atomic and thus normal.*

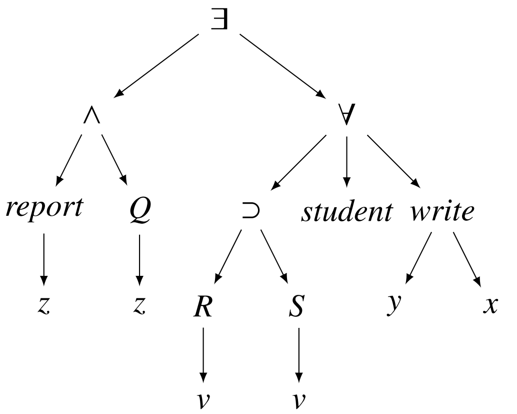
Definition 5.5 (Dominance)**.** *In a term M, occurrences of constants and variables are endowed with a dominance relation as follows.*

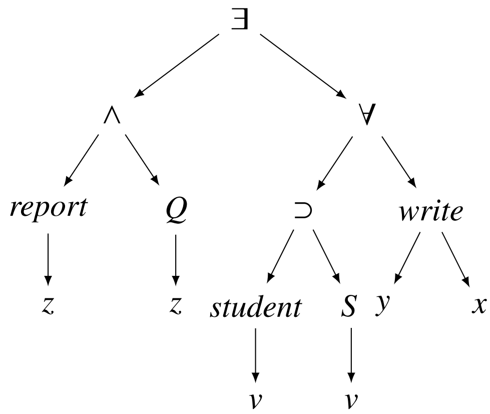
- *If the term is atomic there is no elementary dominance relation.*
- *If the term M is an application M_0M_1, the elementary dominance relations are the union of the ones in each of the M_i, plus the relations: Let M'_0 be the leftmost innermost subterm of M_0. Let M'_1 be the leftmost innermost subterm of M_1 then M'_0 dominates M'_1*
- *If the term M is an abstractions $\lambda x.\ M_1$, then the dominance relations are the ones in M_1.*

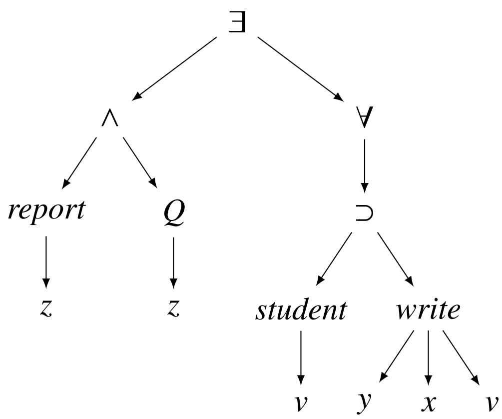
The occurrence of atomic term h elementary dominates the occurrence of atomic term h' is written $h \lhd_1 h'$, and the transitive reflexive closure of $\lhd_1$ is written $\lhd$.

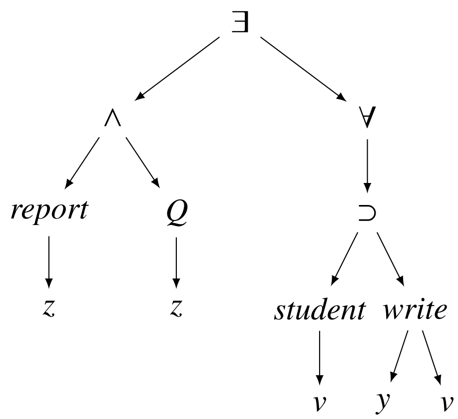
Example 5.2. *Figure 5.5 in the next page shows an example of the dominance relations between occurrences of constants and variables for lambda term 5.5 trough β-reduction. The lambda terms corresponds to the sentence "every student wrote a report" with the existential quantifier having wide scope. Remark that after each step of β we have that $\exists \lhd \forall$.*

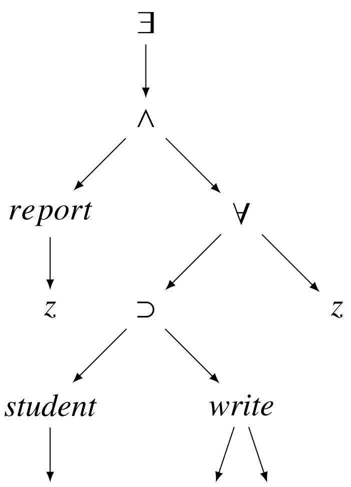

∃
∧ report ∀
P Q ⊃ student write
z z R S x y
v v

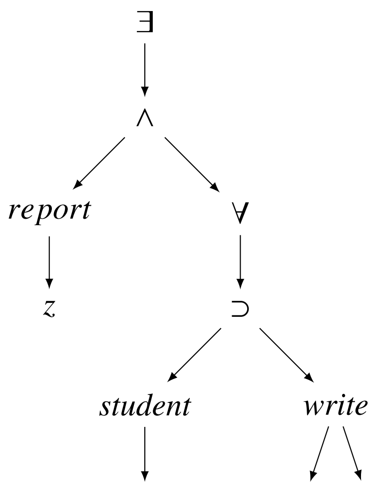

∃
∧ ∀
report Q ⊃ student write
z z R S y x
v v


∃
∧ ∀
report Q ⊃ write
z z student S y x
v v


∃
∧ ∀
report Q ⊃
z z student write
v y x v


∃
∧ ∀
report Q ⊃
z z student write
v y v


∃
∧
report ∀
z ⊃ z
student write
v v x


∃
∧
report ∀
z ⊃
student write
v v z

Before showing that dominance between constant is preserved trough β-reduction, let us state some obvious propositions:

Proposition 5.7. *Let P be a syntactic lambda term with words $w_1, \ldots, w_n$. Let M_i for $1 \leq i \leq n$ be the corresponding simple semantic lambda terms with head constant k_i. If $w_{i_0} \triangleleft w_{i_1}$ in P then $k_{i_0} \triangleleft k_{i_1}$ in $P[N_1/w_1, \ldots N_n/w_n]$.*

Proposition 5.8. *Let $(\lambda x.M)N$ be a redex where $\lambda x.M$ and N are normal terms. Let h_1, h_2 be two atomic terms s.t. h_1 is a subterm of $\lambda x.M$, h_2 is a subterm of N. We have that $h_1 \triangleleft h_2$ iff h_1 is the head variable (constant) of $\lambda x.M$*

Proposition 5.9. *In a term M the leftmost-innermost subterm M_0 of M dominates all occurrences of all the atomic subterm of M.*

The exact formulation of the following theorem is quite cumbersome. However its "moral" content is easy: β-reduction cannot reverse the dominance relation between two occurrences of constants in a λ_I-term. If $k_1 \triangleleft k_2$ in U and U reduces to U' then there are possibly new occurrences of k_1 and k_2 and *between all the occurrences produced by β-reduction* the dominance relation still holds.

Theorem 5.1 (Dominance preservation)**.** *Let M be a typed lambda I term including two occurrences of constants k and k' such that $k \triangleleft k'$ in M. Assume $M \overset{\beta}{\rightsquigarrow} M'$. Then each copy k_i of k is associated with a set of occurrences $k_i'^j$ of k' in M' with $k_i \triangleleft k_i'^j$ in M' — the sets $K_i' = \{k_i'^j\}$ define a partition of copies of k' in M'. In particular, there is never a relation the other way round after reduction: $k_i' \not\triangleleft k_i$ in M' for all i.*

Proof. Let k and k' be occurrences of constants. It is enough to show that whenever $k \triangleleft k'$, then, after one step of innermost β-reduction, $k_i \triangleleft k_i'$. With k_i and k_i' being copies of k and k' as stated above.

A redex in M looks like $M = P[(\lambda x.A)B]$, moreover, we can suppose that both λxA and B are normal. This is enough, since β-reduction is confluent.

For each relation $k \triangleleft_1 k'$ (because of the implicit recursion on the number of β reductions performed so far there might already be several such pairs), because of the definition of dominance k we see that k and k' necessarily in one of the following relations in $M = P[(\lambda x.A)B]$:

1. outside the redex, i.e. elsewhere in P, k and k' do not move

2. both in A. The only interesting case is when we have that $k \triangleleft x \triangleleft k'$ where x is the bound variable in $\lambda x.A$. We can assume that we have $\triangleleft_1$ between the three symbols. Since we are using innermost reduction, B is a term in normal form. This means that it has a head variable (constant) h. In the initial configuration we have -by the definition of dominance- that k is the leftmost innermost subterm of a subterm of $\lambda x A$. Call the subterm of A $A^\star$. $A^\star$ is applied to another normal subterm $A_1^\star$ having x as head variable which is applied to another normal subterm $A_2^\star$ having k' as head constant i.e., the configuration is the following $A^\star(A_1^\star A_2^\star)$. After β-reduction the configuration became $A^\star(A_1^\star[B/x]A_2^\star)$. In this case h — the head variable (constant) of B— is the leftmost-innermost term of $A_1^\star[B/x]$ by the definition of dominance $h \triangleleft_1 k'$. Again by the definition of dominance $k \triangleleft_1 h$

3. both in B, k and k' move inside A being possibly duplicated

4. k is in $\lambda x A$ and k' is in B. Since we are using innermost reduction, both term are in normal form. This implies (Proposition 5.8) that k is the head constant of $\lambda x.A$ and thus is leftmost-innermost subterm. This is still true after β-reduction i.e., k is the leftmost-innermost subterm of $A[B/x]$. This implies (Proposition 5.9) that $k \triangleleft k'$ (for each occurred of k') in $A[B/x]$

In any case $k \triangleleft k'$ also holds after reduction to $A[B/x]$. In case the "initial" k and k' are both in B and A contains several occurrences of x, they have several traces which are in a one to one correspondence with $k \triangleleft k'$ in t'.

Note that having λ_I terms is crucial otherwise both k and k' or just k' may disappear during the β reduction. □

It should be observed that dominance relations, even between occurrences of constants, can be introduced by β-reduction (when a constant becomes the head variable of some term), but this does not prevent the proposition to hold, since we only require the dominance relation to not inverse an already existing relation between two occurrences of constants.

Corollary 2. *Assume two syntactic analyses P_1 and P_2 give opposite dominance relation between two words, $u \triangleleft u'$ in P_1 and $u \triangleleft u'$ in P_2. Whatever the semantic lambda terms for u and u' with different head constant k and k' are, the associated logical forms will be different.*

5.6 Conclusion

In the first part of this chapter, we have introduced the architecture of type-logical grammar. In particular, we have shown how we can assign to a natural language sentence a logical formula. These latter formula corresponds to one of the meaning of the sentence. The architecture we have described can be implemented. By means of example, type logical grammars as described in this chapter are the basis of the wide-coverage French parser which is part of the Grail family of theorem provers [107]: Grail uses a deep learning model to predict the correct formulas for each word, finds the best way to combine these lexical entries and finally produces a representation of a logical formula using the architecture of type logical grammar we have just described. Admittedly the meaning that we assign to natural language sentence is rather simplistic. Logical formulas cannot capture all the subtleties of natural language expressions; however, this simplified meaning assignment has a tremendous advantage: it can be automatically obtained and that it is of exactly the right form for logic-based entailment tasks. Logic-based natural language entailment tasks will be the focus of the next chapter.

In the second part of this chapter we have addressed a natural problem in type logical grammar : do different syntactic analyses, fed with equal semantic terms, lead to different logical forms? We have shown that some quite reasonable formalization of the claim "different syntactic analyses yield different readings" are false. We nevertheless established that with stronger constraints on the allowed semantic terms, and using dominance relations between constants, the claim is true.

One open question, interesting more from a technical point of view than from a natural language semantics point of view, is whether a stronger theorem holds when we assume the syntactic terms are lambda terms obtained from Lambek calculus proofs, i.e. are the commutative "trace" of non-commutative proofs. In proposition 5.5 the two terms P_1 and P_2 cannot correspond to two different lambek proofs of the same sentence i.e., P_1 and P_2 are not the commutative trace of two different non-commutative proofs. We thus may hope for a stronger result.

As said supra, this is of a limited interest for computational semantics. Indeed, plain Lambek calculus is not able to derive some syntactic structures, like the reading of a sentence with three quantifiers, with the middle one taking scope over the other two.

Another open question would be to see how to extend the current results to more general classes of semantic lambda terms, for example by incorporating reflexives (which are assigned semantics terms of the form $\lambda P \lambda x.\ Pxx$, and have no head constant). Observe that some counter examples we gave are using semantic lambda terms whose structures are quite similar to reflexives.

Chapter 6

Textual Entailement Recognition and DiaLogical Games

Abstract
In this chapter we show, by constructing concrete examples, how we can use dialogical logic to solve problems of textual entailment recognition. We start with some simple problems that can be dealt with purely logical means. We then introduce games in which player can use the inferential properties of words. Part of the content of this chapter already appears in [22]

6.1 Introduction

In this chapter we will illustrate the application of the inferentialist view to natural language semantics with a very natural task: the recognition of natural language inference, a task also known as textual entailment. In the current context, we use textual entailment in a more limited sense than it is generally used in natural language processing tasks. Textual entailment in natural language processing generally aims to obtain human-like performance on relating a text and a possible conclusion [32]. Natural language processing systems are evaluated on their ability to approach the performance of humans when deciding between entailment, contradiction and unknown (i.e., neither the entailment relation nor the contradiction relation holds between the text and the given candidate conclusion). We consider textual entailment from a purely logical point of view, taking entailment and contradiction in their strictly logical meanings. In our opinion, a minimal requirement for a textual entailment system should be that it can handle the well-known syllogisms of Aristotle, as well as a number of other patterns [112] with perfect accuracy[1].

The computational correspondence between natural language sentences and logical formulas is obtained using type-logical grammars. In a sense, type-logical grammars are *designed* to produce logical meanings for grammatical sentences. They compute the possible meanings of a sentence viewed as logical formulae.

In the first part of this chapter, we show how we can solve simple instances of textual entailment recognition within the framework of classical dialogical logic. We then introduce dialogical games in which the players can use the inferential information provided by non-logical words. In these latter games, called Unfold-Games, the players plays modulo a fixed set of axioms. Each of these axioms has the form $\forall x_1, \ldots \forall x_m (Q \iff B)$ where Q is a predicate variable (let's call this formula the *definiendum*) and B is an arbitrary formula (let's call this formula *definiens*). Whenever **P** asserted an atomic formula Q that is a definiendum **O** asks **P** to assert the definiens B of Q; If **O** assert an atomic formula Q that is a definiendum **P** may ask **O** to assert the definies of the formula. Said in a less verbose way : we study games in which the players can ask question about the meaning of atomic formulas.

Organization of the chapter In section 6.2 we first briefly present the task of Textual Entailment Recognition and the FraCas Benchmark. We then present, trough examples, our solution to the problem of textual entailment recognition. In the subsequent section 6.3 we discuss entailment relations that holds by virtue of the meaning of non-logical words. To treat these kinds of entailment recognition problems, we define unfold games and prove some properties about these games. We finally show how we can solve textual entailment problem by means of winning U-strategies.

6.2 Textual Entailment Recognition

In natural language processing, *textual entailment* is usually defined as a relation between text fragments that holds whenever the truth of one text fragments follows from another

[1] The patterns in the well-known corpus for testing Textual Entailment recognition called FraCaS [28] greatly vary in difficulty. We expect only some of them (monotonicity, syllogisms) to be handled easily, whereas we expect others (plurals, temporal inference for aspectual classes) to be much more difficult for systems based on automated theorem provers, or, indeed, for any automated system.

text. *Textual entailment recognition* is the task of determining, given text fragments, whether the relation of textual entailment holds between these texts.

Our examples below are taken from the FraCaS benchmark; the FraCaS benchmark was built in the mid 1990s; the aim was developing a general framework from computational semantics. The data set consists of problems, each containing one or more statements and one yes/no-question. An example taken from the date set is the following

(6.1) A Swede won a Nobel prize.

(6.2) Every Swede is a Scandinavian.

(6.3) Did a Scandinavian won a Nobel prize? [Yes]

6.2.1 First Example

We illustrate our methodology to solve inference problems using examples. First we turn the question (6.3) into an assertion, i.e.,

(6.4) Some Scandinavian won a Nobel prize.

We then use the Type-Logical-Grammars to translate the sentences into logical formulas. In the enumeration below we report, in order: the sentence in English and the logical formula that a Type-Logical-Grammar outputs from the input of the latter.

(6.5) A Swede won a Nobel prize
$H_1 = \exists x_1 [\, \mathit{Swede}\,(x_1) \wedge (\exists x_2\, \mathit{Nobel\text{-}prize}\,(x_2) \wedge \mathit{won}\,(x_1, x_2))]$

(6.6) Every Swede is a Scandinavian
$H_2 = \forall x_3, [\, \mathit{Swede}\,(x_3) \supset \mathit{Scandinavian}\,(x_3)]$

(6.7) Some Scandinavian won a Nobel prize
$C = \exists x_4 [\, \mathit{Scandinavian}\,(x_4) \wedge (\exists x_5\, \mathit{Nobel\text{-}prize}\,(x_5) \wedge \mathit{won}\,(x_4, x_5))]$

Finally, we construct a dialogical logic *winning strategy* for the formula $H_1 \wedge \cdots \wedge H_n \supset C$ where each H_i is the logical formula that a type logical grammar associates to each statement from the data set, and C is the formula that associated to the assertion obtained from the pair question-answer in the data-set. Below we show a winning strategy for the formula $H_1 \wedge H_2 \supset C$. In the strategy *Sw*, *N-p* and *Sc* stands for *Swede*, *Nobel-prize* and *Scandinavian*

$$
\begin{array}{ll|l}
\mathfrak{m}_0 = (!, \big(\exists x_1[Sw(x_1) \wedge (\exists x_2\, N\text{-}p(x_2) \wedge w(x_1, x_2))] \;\wedge\; \forall x_3[S\,w(x_3) \supset S\,c(x_3)]\big) & & \\
\quad \supset \exists x_4[(Sc(x_4) \wedge (\exists\, x_5\, N\text{-}p(x_5) \wedge w(x_4, x_5))] & & \\
\mathfrak{m}_1 = (?, \big(\exists x_1[Sw(x_1) \wedge (\exists x_2\, N\text{-}p(x_2) \wedge w(x_1, x_2))] \;\wedge\; \forall x_3[S\,w(x_3) \supset S\,c(x_3)]\big)) & & \mathfrak{m}_0 \\
\mathfrak{m}_2 = (?, \wedge_1) & & \mathfrak{m}_1 \\
\mathfrak{m}_3 = (!, \exists x_1 Sw(x_1) \wedge (\exists x_2\, N\text{-}p(x_2) \wedge w, (x_1, x_2))) & & \mathfrak{m}_2 \\
\mathfrak{m}_4 = (?, \exists) & & \mathfrak{m}_3 \\
\mathfrak{m}_5 = (!, Sw(y_1) \wedge (\exists x_2\, N\text{-}p(x_2) \wedge w\,(y_1, x_2))]) & & \mathfrak{m}_4 \\
\mathfrak{m}_6 = (?, \wedge) & & \mathfrak{m}_5 \\
\mathfrak{m}_7 = (!, Sw(y_1)) & & \mathfrak{m}_6 \\
\mathfrak{m}_8 = (?, \wedge_2) & & \mathfrak{m}_5 \\
\mathfrak{m}_9 = (!, (\exists x_2\, N\text{-}p\,(x_2) \wedge w\,(y_1, x_2))) & & \mathfrak{m}_8 \\
\mathfrak{m}_{10} = (?, \exists) & & \mathfrak{m}_9 \\
\mathfrak{m}_{11} = (!, N\text{-}p\,(y_2) \wedge w\,(y_1, y_2)) & & \mathfrak{m}_{10} \\
\mathfrak{m}_{12} = (?, \wedge_2) & & \mathfrak{m}_1 \\
\mathfrak{m}_{13} = (!, \forall x_3(Sw\,(x_3)\text{ımp}Sc\,(x_3)) & & \mathfrak{m}_{12} \\
\mathfrak{m}_{14} = (?, \forall[y_1/x_3]) & & \mathfrak{m}_{13} \\
\mathfrak{m}_{15} = (!, Sw\,(y_1) \supset Sc\,(y_1)) & & \mathfrak{m}_{14} \\
\mathfrak{m}_{16} = (?, Sw\,(y_1)) & & \mathfrak{m}_{15} \\
\mathfrak{m}_{17} = (!, Sc\,(y_1)) & & \mathfrak{m}_{16} \\
\mathfrak{m}_{18} = (!, \exists x_4[(Sc(x_4) \wedge (\exists\, x_5\, N\text{-}p(x_5) \wedge w(x_4, x_5))]) & & \mathfrak{m}_1 \\
\mathfrak{m}_{19} = (?, \exists) & & \mathfrak{m}_{18} \\
\mathfrak{m}_{20} = (!, (Sc\,(y_1) \wedge (\exists\, x_5\, N\text{-}p\,(x_5) \wedge w\,(y_1, x_5))) & & \mathfrak{m}_{19}
\end{array}
$$

$$
\begin{array}{l|l}
\mathfrak{m}_{2_1} = (?, \wedge_1) & \mathfrak{m}_{20} \\
\mathfrak{m}_{22} = (!, Sc\,(y_1)) & \mathfrak{m}_{21}
\end{array}
\qquad
\begin{array}{l|l}
\mathfrak{m}_{21} = (?, \wedge_2) & \mathfrak{m}_{20} \\
\mathfrak{m}_{22} = (!, (\exists\, x_5\, N\text{-}p\,(x_5) \wedge w\,(y_1, x_5))) & \mathfrak{m}_{21} \\
\mathfrak{m}_{23} = (?, \exists) & \mathfrak{m}_{22} \\
\mathfrak{m}_{24} = (!, N\text{-}p\,(y_2) \wedge w\,(y_1, y_2))) & \mathfrak{m}_{23}
\end{array}
$$

$$
\begin{array}{l|l}
\mathfrak{m}_{25} = (?, \wedge_1) & \mathfrak{m}_{24} \\
\mathfrak{m}_{26} = (!, N\text{-}p\,(y_2)) & \mathfrak{m}_{25}
\end{array}
\qquad
\begin{array}{l|l}
\mathfrak{m}_{25} = (?, \wedge_2) & \mathfrak{m}_{24} \\
\mathfrak{m}_{26} = (!, w\,(y_1, y_2)) & \mathfrak{m}_{24}
\end{array}
$$

6.2.2 Second Example

(6.8) Some Irish delegates finished the survey on time.

(6.9) Did any delegates finish the survey on time? [Yes]

The answer to the question is affirmative. This means that if (6.8) is true, then the sentence *"some delegate finished the survey on time"* must also be true.

(6.10) Some Irish delegates finished the survey on time

$$F_1 = \exists x_1\,[(\mathit{Irish}\,(x_1) \wedge \mathit{delegate}\,(x_1)) \wedge \exists x_2\,(\mathit{survey}\,(x_2) \wedge \mathit{finished\text{-}on\text{-}time}\,(x_1, x_2))]$$

(6.11) Some delegates finished the survey on time.

$$F_2 = \exists x_3\,[\mathit{delegate}\,(x_3) \wedge \exists x_4\,(\mathit{survey}\,(x_4) \wedge \mathit{finished\text{-}on\text{-}time}\,(x_3, x_4))]$$

A winning strategy for the formula $F_1 \supset F_2$ is shown below; *Ir*, *de*, *f-o-t* and *su* stands for *Irish*, *delegate*, *finished-on-time* and *survey* respectively.

$\mathfrak{m}_0 = (!, (\exists x_1\,[(Ir\,(x_1) \wedge de\,(x_1)) \wedge \exists x_2\,(su\,(x_2) \wedge f\text{-}o\text{-}t\,(x_1, x_2))]) \supset \exists x_3\,[de\,(x_3) \wedge \exists x_4\,(su\,(x_4) \wedge f\text{-}o\text{-}t\,(x_3, x_4)))$	
$\mathfrak{m}_1 = (?, \exists x_3\,[de\,(x_3) \wedge \exists x_4\,(su\,(x_4) \wedge f\text{-}o\text{-}t\,(x_3, x_4)))$	$\mathfrak{m}_0$
$\mathfrak{m}_2 = (?, \exists)$	$\mathfrak{m}_1$
$\mathfrak{m}_3 = (!, (Ir\,(y_1) \wedge de\,(y_1)) \wedge \exists x_2\,(su\,(x_2) \wedge f\text{-}o\text{-}t\,(y_1, x_2)))$	$\mathfrak{m}_2$
$\mathfrak{m}_4 = (?, \wedge_1)$	$\mathfrak{m}_3$
$\mathfrak{m}_5 = (!, (Ir\,(y_1) \wedge de\,(y_1)))$	$\mathfrak{m}_4$
$\mathfrak{m}_6 = (?, \wedge_2)$	$\mathfrak{m}_5$
$\mathfrak{m}_7 = (!, de\,(y_1))$	$\mathfrak{m}_6$
$\mathfrak{m}_8 = (?, \wedge_2)$	$\mathfrak{m}_3$
$\mathfrak{m}_9 = (!, \exists x_2\,(su\,(x_2) \wedge f\text{-}o\text{-}t\,(y_1, x_2))))$	$\mathfrak{m}_8$
$\mathfrak{m}_{10} = (?, \exists)$	$\mathfrak{m}_9$
$\mathfrak{m}_{11} = (!, (su\,(y_2) \wedge f\text{-}o\text{-}t\,(y_1, y_2))))$	$\mathfrak{m}_{10}$
$\mathfrak{m}_{12} = (?, \wedge_1)$	$\mathfrak{m}_{11}$
$\mathfrak{m}_{13} = (!, su\,(y_2))$	$\mathfrak{m}_{12}$
$\mathfrak{m}_{14} = (?, \wedge_2)$	$\mathfrak{m}_{11}$
$\mathfrak{m}_{15} = (!, f\text{-}o\text{-}t\,(y_1, y_2))$	$\mathfrak{m}_{14}$
$\mathfrak{m}_{16} = (!, \exists x_3\,[de\,(x_3) \wedge \exists x_4\,(su\,(x_4) \wedge f\text{-}o\text{-}t\,(x_3, x_4)))])$	$\mathfrak{m}_1$
$\mathfrak{m}_{17} = (?, \exists)$	$\mathfrak{m}_{16}$
$\mathfrak{m}_{18} = (!, de\,(y_1) \wedge \exists x_4\,(su\,(x_4) \wedge f\text{-}o\text{-}t\,(y_1, x_4))))$	$\mathfrak{m}_{17}$

Left branch from $\mathfrak{m}_{18}$:

$\mathfrak{m}_{19} = (?, \wedge_1)$	$\mathfrak{m}_{18}$
$\mathfrak{m}_{20} = (!, de\,(y_1))$	$\mathfrak{m}_{19}$

Right branch from $\mathfrak{m}_{18}$:

$\mathfrak{m}_{19} = (?, \wedge_2)$	$\mathfrak{m}_{18}$
$\mathfrak{m}_{20} = (\exists x_4\,(su\,(x_4) \wedge f\text{-}o\text{-}t\,(y_1, x_4))))$	$\mathfrak{m}_{19}$
$\mathfrak{m}_{21} = (?, \exists)$	$\mathfrak{m}_{20}$
$\mathfrak{m}_{22} = (!, (su\,(y_2) \wedge f\text{-}o\text{-}t\,(y_1, y_2)))$	

Left branch from $\mathfrak{m}_{22}$:

$\mathfrak{m}_{23} = (?, \wedge_1)$	$\mathfrak{m}_{22}$
$\mathfrak{m}_{24} = (!, su\,(y_2))$	$\mathfrak{m}_{23}$

Right branch from $\mathfrak{m}_{22}$:

$\mathfrak{m}_{23} = (?, \wedge_2)$	$\mathfrak{m}_{22}$
$\mathfrak{m}_{24} = (!, f\text{-}o\text{-}t\,(y_1, y_2))$	$\mathfrak{m}_{23}$

6.2.3 Third Example

(6.12) No delegate finished the report on time.

(6.13) Did any Scandinavian delegate finished the report on time? [No]

In this example, the question should get a negative reply. A positive answer would be implied by the existence of a Scandinavian delegate who finished the report in the time allotted. Thus, the sentence (6.12) plus the sentence "*Some Scandinavian delegate finished the report on time*" should imply a contradiction. We first translate the two sentences into logical formulas.

(6.14) No delegate finished the report on time.

$$F_1 = \forall x_1 \, (\mathit{delegate}\,(x_1) \supset \neg(\exists x_2 \, (\mathit{report}\,(x_2) \wedge \mathit{finished\text{-}on\text{-}time}\,(x_1, x_2)))$$

(6.15) Some Scandinavian delegate finished the report on time.

$$F_2 = \exists x_3 \, [\mathit{Scandinavian}\,(x_3) \wedge \mathit{delegate}\,(x_3) \wedge \exists x_4 \, (\mathit{report}\,(x_4) \wedge \mathit{finished\text{-}on\text{-}time}\,(x_3, x_4))]$$

The two formulas F_1 and F_2 are contradictory. So there exists a winning strategy for the formula $\neg(F_1 \wedge F_2)$ as shown below. In the strategy *de*, *re*, *f-o-t* and *Sc* stands for, respectively, *delegate*, *report*, *finished-on-time* and *Scandinavian*

$$
\begin{array}{l|l}
\mathfrak{m}_0 = (!, \neg(F_1 \wedge F_2)) & \\
\mathfrak{m}_1 = (?, F_1 \wedge F_2) & \mathfrak{m}_0 \\
\mathfrak{m}_2 = (?, \wedge_2) & \mathfrak{m}_1 \\
\mathfrak{m}_3 = (!, \exists\, x_3\, (Sc\,(x_3) \wedge de\,(x_3) \wedge \exists\, x_4\, (re\,(x_4) \wedge f\text{-}o\text{-}t\,(x_3, x_4)))) & \mathfrak{m}_2 \\
\mathfrak{m}_4 = (?, \exists) & \mathfrak{m}_3 \\
\mathfrak{m}_5 = (!, (Sc\,(y_1) \wedge de\,(y_1)) \wedge \exists\, x_4\, (re\,(x_4) \wedge f\text{-}o\text{-}t\,(y_1, x_4)))) & \mathfrak{m}_4 \\
\mathfrak{m}_6 = (?, \wedge_1) & \mathfrak{m}_5 \\
\mathfrak{m}_7 = (!, Sc\,(y_1) \wedge de\,(y_1)) & \mathfrak{m}_6 \\
\mathfrak{m}_8 = (?, \wedge_2) & \mathfrak{m}_7 \\
\mathfrak{m}_9 = (!, de\,(y_1)) & \mathfrak{m}_8 \\
\mathfrak{m}_{10} = (?, \wedge_2) & \mathfrak{m}_5 \\
\mathfrak{m}_{11} = (!, \exists\, x_4\, (re\,(x_4) \wedge f\text{-}o\text{-}t\,(y_1, x_4)))) & \mathfrak{m}_6 \\
\mathfrak{m}_{12} = (?, \exists) & \mathfrak{m}_{11} \\
\mathfrak{m}_{13} = (!, (re\,(y_2) \wedge f\text{-}o\text{-}t\,(y_1, y_2)))) & \mathfrak{m}_{12} \\
\mathfrak{m}_{14} = (?, \wedge_1) & \mathfrak{m}_{12} \\
\mathfrak{m}_{15} = (!, re\,(y_2)) & \mathfrak{m}_{14} \\
\mathfrak{m}_{16} = (?, \wedge_2) & \mathfrak{m}_{13} \\
\mathfrak{m}_{17} = (!, f\text{-}o\text{-}t\,(y_1, x_4)) & \mathfrak{m}_{16} \\
\mathfrak{m}_{18} = (?, \wedge_1) & \mathfrak{m}_1 \\
\mathfrak{m}_{19} = (!, [\forall\, x_1\, (de\,(x_1) \supset \neg(\exists\, x_2\, (re\,(x_2) \wedge f\text{-}o\text{-}t\,(x_1, x_2)))) & \mathfrak{m}_{18} \\
\mathfrak{m}_{20} = (?, \forall[y_1/x_1]) & \mathfrak{m}_{19} \\
\mathfrak{m}_{21} = (!, (de\,(y_1) \supset \neg(\exists\, x_2\, (re\,(x_2) \wedge f\text{-}o\text{-}t\,(y_1, x_2)))) & \mathfrak{m}_{20} \\
\mathfrak{m}_{22} = (?, de\,(y_1)) & \mathfrak{m}_{21} \\
\mathfrak{m}_{23} = (!, \neg(\exists\, x_2\, (re\,(x_2) \wedge f\text{-}o\text{-}t\,(y_1, x_2))) & \mathfrak{m}_{22} \\
\mathfrak{m}_{24} = (?, \exists\, x_2\, (re\,(x_2) \wedge f\text{-}o\text{-}t\,(y_1, x_2)) & \mathfrak{m}_{23}
\end{array}
$$

$$
\begin{array}{l|l}
\mathfrak{m}_{25} = (?, \exists) & \mathfrak{m}_{24} \\
\mathfrak{m}_{26} = (!, re\,(y_2) \wedge f\text{-}o\text{-}t\,(y_1, y_2)) & \mathfrak{m}_{25}
\end{array}
\qquad
\begin{array}{l|l}
\mathfrak{m}_{25} = (!, \bot) & \mathfrak{m}_{24}
\end{array}
$$

$$
\begin{array}{l|l}
\mathfrak{m}_{27} = (?, \wedge_1) & \mathfrak{m}_{26} \\
\mathfrak{m}_{28} = (!, re\,(y_2)) & \mathfrak{m}_{27}
\end{array}
\qquad
\begin{array}{l|l}
\mathfrak{m}_{27} = (?, \wedge_2) & \mathfrak{m}_{26} \\
\mathfrak{m}_{28} = (!f\text{-}o\text{-}t\,(y_1, y_2)) & \mathfrak{m}_{27}
\end{array}
$$

6.2.4 Fourth Example

In this example we focus on a series of sentences that our system should not solve, because the question asked neither has a positive nor a negative answer.

(6.16) A Scandinavian won a Nobel prize.

(6.17) Every Swede is a Scandinavian

(6.18) Did a Swede win a Nobel prize? [Don't know]

This means that, on the basis of the information provided, we can neither say that a Swede has won a Nobel Prize nor that there are no Swedes who have won a Nobel Prize.

(6.19) A Scandinavian won a Nobel prize.
$F_1 = \exists x_1 [\mathit{Scandinavian}(x_1) \wedge \exists x_2 (\mathit{Nobel\text{-}prize}(x_2) \wedge \mathit{won}(x_1, x_2))]$

(6.20) Every Swede is a Scandinavian
$F_3 = \forall x_3 (\mathit{Swede}(x_3) \supset \mathit{Scandinavian}(x_3))$

In dialogical logic terms, the fact that we do not have enough information to answer the question (6.18), either positively or negatively, means that there is no winning strategy for the formula $F_1 \wedge F_2 \supset F_3$ nor for the formula $F_1 \wedge F_2 \supset \neg F_3$ where the formula F_3 is

$$F_3 = \exists x_4 (\mathit{Swede}(x_4) \wedge \exists x_5 (\mathit{Nobel\text{-}prize}(x_5) \wedge \mathit{won}(x_4, x_5)))$$

We consider what a winning strategy for the formula $F_1 \wedge F_2 \supset F_3$ must look like. We recall that, by definition, a game is won by **P** iff it is finite and either ends with a **P**-move that asserts an atomic formula or ends with an **O**-move that asserts $\perp$. Since there is no occurrence of $\perp$ in the formula $H = F_1 \wedge F_2 \supset F_3$ all games in a winning strategy for H must end with a **P**-assertion of an atomic formula. By proposition 3.4 **P** can assert an atomic formula B in a game for a formula F only if B is both a positive and negative gentzen sub-formula of F. In the case of the formula H, the only atomic formulas that are both positive and negative gentzen subformulas of H are $\mathit{Nobel\text{-}prize}(t)$ for all term t of the language, and $\mathit{won}(t_1, t_2)$ for all terms t_1 and t_2 of the language. In order to assert the formulas $\mathit{Nobel\text{-}prize}(t)$ and $\mathit{won}(t_1, t_2)$ **P** must assert the formula $\exists x_4 (\mathit{Swede}(x_4) \wedge \exists x_5 (\mathit{Nobel\text{-}prize}(x_5) \wedge \mathit{won}(x_4, x_5)))$. If a game $\mathcal{G}, \mathfrak{m}^{\mathbf{P}} \in \mathcal{S}$ where $\mathfrak{m} = (!, \exists x_4 (\mathit{Swede}(x_4) \wedge \exists x_5 (\mathit{Nobel\text{-}prize}(x_5) \wedge \mathit{won}(x_4, x_5))))$ then, by the definition 3.5 of strategy, $\mathcal{S}$ must contain also the game $\mathcal{G}_1 = \mathcal{G}\mathfrak{m}^{\mathbf{P}}\mathfrak{m}'^{\mathbf{O}}\mathfrak{n}^{\mathbf{P}}\mathfrak{n}'^{\mathbf{O}}$ where $\mathfrak{m}' = (?, \exists)$, $\mathfrak{n} = (!, (\mathit{Swede}(t') \wedge \exists x_5 (\mathit{Nobel\text{-}prize}(x_5) \wedge \mathit{won}(t', x_5)))$ where t' is a term of the language, and $\mathfrak{n}' = (?, \wedge_1)$. Remark that there cannot be any move legal for the game $\mathcal{G}_1$: this is because the formula $\mathit{Swede}(t')$ is only a negative gentzen subformula of H for all terms t'. By this we can conclude that there is no winning strategy $\mathcal{S}$ for the formula $H = F_1 \wedge F_2 \supset F_3$.

since $\neg F_3$ is a positive formula is a positive formula of $F_1 \wedge F_2 \supset \neg F_3$ then, if it is asserted in a game $\mathcal{G}$, it is **P** that asserts it. This means that **P** cannot win the game because **O** asserts $\perp$. Thus if there is a winning strategy $\mathcal{S}$ for $F_1 \wedge F_2 \supset \neg F_3$ it should only contain games ending with a **P**-move. This means that the **P**-move that is the last move on each of these games must be a defence-move that asserts some atomic gentzen subformula of $F_1 \wedge F_2 \supset \neg F_3$. Using again proposition 3.4 we must conclude that the only formula that appears both positively and negatively in $F_1 \wedge F_2 \supset \neg F_3$ is $Swede(t)$ for all terms t in the language. Let $\mathcal{G}\mathfrak{m}$ be a game in $\mathcal{S}$ where $\mathfrak{m}$ is an assertion of $swede(t)$ for some term t. We conclude that $\mathfrak{m}$ must be an attack move. If it were a defense instead, this would mean that there must be a sub-formula G of $F_1 \wedge F_2 \supset F_3$ and that G is of of the form $\forall w.Swede(w)$ or $\exists w(swede\,(w))$ or $G' \vee Swede\,(w)$ or $G' \wedge Swede\,(w)$ or $G' \supset Swede\,(t)$ such that **P** asserts G. This implies that this formula G must be a positive Gentzen subformula of $F_1 \wedge F_2 \supset \neg F_3$. But not such formula exists. Thus, the move $\mathfrak{m}$ asserting $Swede\,(t)$ must be an attack. Since the only formula that can be attacked by this means is the formula $Swede(t) \supset Scandinavian(t)$, **O** can answer by asserting $Scandinavian(t)$, and **P** cannot win the game. Thus, there is no winning strategy $\mathcal{S}$ for the formula $F_1 \wedge F_2 \supset \neg F_3$.

6.3 Word Knowledge

In this section we extend our simple model. We take into account inference problems whose solution must invoke some background word knowledge. Consider the following couple of sentences:

(6.21) The cost of living soared.

(6.22) The cost of living raised.

Any English speaker would recognize that if the statement 6.21 is true then the statement 6.22 must also be true. This is simply because "to soar" means "to raise quickly". To solve correctly the inference problem, one need to take into account the meaning of the verb "to soar". We now introduce a way to take into account this kind of implicit word knowledge into the winning strategies of dialogical logic. Our idea to take into account inference that involve word knowledge is very simple: let us put ourselves in the shoes of someone who does not know the meaning of a word e.g., the verb "to soar". How would she react to the assertion of the statement 6.21? Probably she would look confused and simply ask for the meaning of "to soar". If she receives a good explanation she would certainly —at least we hope— conclude that 6.22 is logical consequence of 6.21.

6.3.1 Unfolding

If A and B are formulas we write $A \iff B$ as a shortcut for $(A \supset B) \wedge (B \supset A)$. Let $\mathcal{L}$ be a first order language and consider the following set;

$$\mathcal{AX} = \{F \text{ is a formula of } \mathcal{L} \mid F \text{ is of the form } \forall x_1, \ldots, \forall x_m (P \iff C)\}$$

where $m \geq 0$, P is an atomic formula and C is a formula. The definition to follow is taken from [40]

Definition 6.1. *We say that an atomic proposition A* unfolds *to a proposition B if $A = \theta(P)$ and $B = \theta(C)$ for some $\forall x_1, \ldots \forall x_m(P \iff C) \in \mathcal{AX}$ and some substitution θ. We assume that $\mathcal{AX}$ is such that each atomic formula* unfolds *to at most one formula. Moreover, we suppose that the set $\mathcal{AX}$ is consistent: there is an interpretation structure $\mathfrak{M}$ such that $\mathfrak{M} \models F$ for any $F \in \mathcal{AX}$.*

Remark that —according to the above definition— if P unfolds to C, then the formula C may contain occurrences of P. However, for the applications we are interested in, we can assume that if P unfolds to C then P does not occur in C.

We consider that a set of formulas $\mathcal{AX}$ is given; the set Aux of auxiliary symbols is the smallest containing the symbols $\wedge_1, \wedge_2, \vee, \exists, \mathcal{U}$ and the expression $\forall[t/x]$ for all terms t and variables x. Recall that the function Arg that maps non-atomic formulas to set of pairs in which each pair is composed of a question and an answer has been defined in subsection 3.2.2 as follows:

$$\begin{aligned}
Arg(A \supset B) &= \{(A, B)\} \\
Arg(A \wedge B) &= \{(\wedge_1, A), (\wedge_2, B)\} \\
Arg(A \vee B) &= \{(\vee, A), (\vee, B)\} \\
Arg(\forall x A) &= \{(\forall[t/x], A[t/x]) \mid t \in \mathcal{T}\} \\
Arg(\exists x A) &= \{(\exists, A[t/x]) \mid t \in \mathcal{T}\}
\end{aligned}$$

we now extend this definition

Definition 6.2 (Unfold argumentation form)**.** *An Unfold argumentation form* $\texttt{Arg}^{\mathcal{U}}$ *is a partial function that maps formulas to sets of pairs of question and answers. A question is either a formula or an element of Aux and an answer is a formula.*

- $\texttt{Arg}^{\mathcal{U}}(A) = Arg(A)$ *if A is non-atomic*
- $\texttt{Arg}^{\mathcal{U}}(A) = \{(\mathcal{U}, C)\}$ *if A is atomic and A unfold to C;*
- *It is undefined otherwise.*

A defense move is a pair $(!, A)$ where A is a formula. An attack move is a pair $(?, s)$ where s is either a formula or an auxiliary symbol. The notion of justified attack move and justified defense move are defined as in definition 3.2 with the only difference that what count as a question about a formula and what counts as an answer about that question are now defined with respect to the function $\texttt{Arg}^{\mathcal{U}}$. We define games in which the two player can use the information provided by the set of formulas $\mathcal{AX}$

Definition 6.3 (Unfold Game)**.** *An* Unfold Game *(U-Game for short) is a game* (ρ, ϕ) *in the sense of definition 3.3 in which condition 3 is replaced by*

- *if* $(!, B) = \mathfrak{m}_k \in \rho$ *where B is an atomic formula and k is even, then either* $\mathfrak{m}_k$ *is a repetition and* $B \neq \bot$ *or B unfolds to C.*

The notion of move $\mathfrak{m}$ that is legal for a game $\mathcal{G}$ is defined exactly as in subsection 3.2.4 as well as the notion of game won by **P** (definition 3.4). Games in the sense of definition 3.3 will be called *regular* games in this section.

6.3.2 Some Examples of U-games

Consider a first order language $\mathcal{L}$ in which the only predicate symbols are the binary predicate symbol $\in$ and the binary predicate symbol $\subseteq$. Suppose that the only formula in $\mathcal{AX}$ is $\forall x \forall y (x \subseteq y \iff \forall z (z \in x \supset z \in y))$. We show two games won by **P**

$\mathfrak{m}_0 = (!, x \subseteq x)$	
$\mathfrak{m}_1 = (?, \mathcal{U})$	$\mathfrak{m}_0$
$\mathfrak{m}_2 = (!, \forall z (z \in x \supset z \in x))$	$\mathfrak{m}_1$
$\mathfrak{m}_3 = (?, [y_1/z])$	$\mathfrak{m}_2$
$\mathfrak{m}_4 = (!, y_1 \in x \supset y_1 \in x)$	$\mathfrak{m}_3$
$\mathfrak{m}_5 = (?, y_1 \in x)$	$\mathfrak{m}_4$
$\mathfrak{m}_6 = (!, y_1 \in x)$	$\mathfrak{m}_5$

$\mathfrak{m}_0 = (!, (x_1 \subseteq x_2 \wedge x_2 \subseteq x_3) \supset x_1 \subseteq x_3)$	
$\mathfrak{m}_1 = (?, x_1 \subseteq x_2 \wedge x_2 \subseteq x_3)$	$\mathfrak{m}_0$
$\mathfrak{m}_2 = (!, x_1 \subseteq x_3)$	$\mathfrak{m}_1$
$\mathfrak{m}_3 = (?, \mathcal{U})$	$\mathfrak{m}_2$
$\mathfrak{m}_4 = (!, \forall z (z \in x_1 \supset z \in x_3))$	$\mathfrak{m}_3$
$\mathfrak{m}_5 = (?, \forall [y_1/z])$	$\mathfrak{m}_4$
$\mathfrak{m}_6 = (!, y_1 \in x_1 \supset y_1 \in x_3)$	$\mathfrak{m}_5$
$\mathfrak{m}_7 = (?, y_1 \in x_1)$	$\mathfrak{m}_6$
$\mathfrak{m}_8 = (?, \wedge_1)$	$\mathfrak{m}_1$
$\mathfrak{m}_9 = (!, x_1 \subseteq x_2)$	$\mathfrak{m}_8$
$\mathfrak{m}_{10} = (?, \mathcal{U})$	$\mathfrak{m}_9$
$\mathfrak{m}_{11} = (!, \forall z (z \in x_1 \supset z \in x_2))$	$\mathfrak{m}_{10}$
$\mathfrak{m}_{12} = (?, \forall [y_1/z])$	$\mathfrak{m}_{11}$
$\mathfrak{m}_{13} = (!, y_1 \in x_1 \supset y_1 \in x_2)$	$\mathfrak{m}_{12}$
$\mathfrak{m}_{14} = (?, y_1 \in x_1)$	$\mathfrak{m}_{13}$
$\mathfrak{m}_{15} = (?, y_1 \in x_2)$	$\mathfrak{m}_{14}$
$\mathfrak{m}_{16} = (?, \wedge_2)$	$\mathfrak{m}_1$
$\mathfrak{m}_{17} = (!, x_2 \in x_3)$	$\mathfrak{m}_{16}$
$\mathfrak{m}_{18} = (?, \mathcal{U})$	$\mathfrak{m}_{17}$
$\mathfrak{m}_{19} = (?, \forall z (z \in x_2 \supset z \in x_3))$	$\mathfrak{m}_{18}$
$\mathfrak{m}_{20} = (!, \forall [y_1/z])$	$\mathfrak{m}_{19}$
$\mathfrak{m}_{21} = (?, y_1 \in x_2 \supset y_1 \in x_3)$	$\mathfrak{m}_{20}$
$\mathfrak{m}_{22} = (?, y_1 \in x_2)$	$\mathfrak{m}_{21}$
$\mathfrak{m}_{23} = (?, y_1 \in x_3)$	$\mathfrak{m}_{22}$
$\mathfrak{m}_{24} = (!, y_1 \in x_3)$	$\mathfrak{m}_7$

Consider a multi-sorted first order language $\mathcal{L}$ in which the set of sorts includes the sort e of entities and the sort v of events, `tony` and `christopher` are constants symbols of sorts e, α is a constant symbol of fort v, and in which there are predicate symbols *killing* with sort $v \to t$, *agent* and *patient* with sort $v \to (e \to t)$, *intentionally* with sort $v \to t$, and *murderer* with sort $e \to t$. Finally, suppose that $\mathcal{AX}$ contains the formula

$$\forall x^e \, [murderer(x) \iff \exists z^v \, \exists y^e ((killing\,(z) \wedge intentionally\,(z)) \wedge (agent\,(z, x) \wedge patient\,(z, y))))]$$

$\mathfrak{m}_0 = (!, [(ki\,(\alpha) \wedge in\,(\alpha)) \wedge (ag\,(\alpha, \texttt{tony}) \wedge pa(\alpha, \texttt{christopher}))] \supset mu\,(\texttt{tony}))$	
$\mathfrak{m}_1 = (?, [(ki\,(\alpha) \wedge in\,(\alpha)) \wedge (ag\,(\alpha, \texttt{tony}) \wedge pa(\alpha, \texttt{christopher}))])$	$\mathfrak{m}_0$
$\mathfrak{m}_2 = (!, mu\,(\texttt{tony}))$	$\mathfrak{m}_1$
$\mathfrak{m}_3 = (?, \mathrm{U})$	$\mathfrak{m}_2$
$\mathfrak{m}_4 = (!, \exists z^v\, \exists y^e((ki\,(z) \wedge in\,(z)) \wedge (ag\,(z, \texttt{tony}) \wedge pa\,(z, y)))))$	$\mathfrak{m}_3$
$\mathfrak{m}_5 = (?, \exists)$	$\mathfrak{m}_4$
$\mathfrak{m}_6 = (!, \exists y^e((ki\,(\alpha) \wedge in\,(\alpha)) \wedge (ag\,(\alpha, \texttt{tony},) \wedge pa\,(\alpha, y)))))$	$\mathfrak{m}_5$
$\mathfrak{m}_7 = (?, \exists)$	$\mathfrak{m}_6$
$\mathfrak{m}_8 = (!, (ki\,(\alpha) \wedge in\,(\alpha)) \wedge (ag\,(\alpha, \texttt{tony},) \wedge pa\,(\alpha, \texttt{christopher})))$	$\mathfrak{m}_7$
$\mathfrak{m}_9 = (?, \wedge_2)$	$\mathfrak{m}_8$
$\mathfrak{m}_{10} = (?, \wedge_2)$	$\mathfrak{m}_1$
$\mathfrak{m}_{11} = (!, (ag\,(\alpha, \texttt{tony},) \wedge pa\,(\alpha, \texttt{christopher}))$	$\mathfrak{m}_{10}$
$\mathfrak{m}_{12} = (!, (ag\,(\alpha, \texttt{tony},) \wedge pa\,(\alpha, \texttt{christopher}))$	$\mathfrak{m}_9$
$\mathfrak{m}_{13} = (?, \wedge_1)$	$\mathfrak{m}_{12}$
$\mathfrak{m}_{14} = (?, \wedge_1)$	$\mathfrak{m}_{11}$
$\mathfrak{m}_{15} = (!, ag\,(\alpha, \texttt{tony}))$	$\mathfrak{m}_{14}$
$\mathfrak{m}_{16} = (!, ag\,(\alpha, \texttt{tony}))$	$\mathfrak{m}_{13}$

In the above game *ki* stands for *killing*, *mu* stands for murderer, *ag* stands for *agent* etc.

6.3.3 Properties of U-games

We now prove that from a winning U-game we can construct a regular winning game and vice versa. This result is fairly obvious, and we include the proof for the sake of completeness. Let $\mathcal{G}$ be a U-game, $\mathcal{AX}$ a set of formulas and $H = \forall x_1, \ldots \forall x_m(P \iff C)$ a formula in $\mathcal{AX}$. We say that H is *used* in $\mathcal{G}$ iff there is move $\mathfrak{m} \in \mathcal{G}$ such that $\mathfrak{m} = (!, \theta(C))$ for some substitution θ.

Lemma 6.1. *Let $\mathcal{AX}$ be a set of formulas, A a formula, $\mathcal{G}$ a U-game for A won by* **P** *and $H_1, H_2, \ldots H_n$ formulas of $\mathcal{AX}$ that are used in $\mathcal{G}$. There is a regular game $\mathcal{G}_r$ won by* **P** *for the formula $H_1 \wedge H_2 \wedge \cdots \wedge H_n \supset A$.*

Proof. The regular game $\mathcal{G}_r$ for the formula $H_1 \wedge H_2 \wedge \cdots \wedge H_n \supset A$ is played as follows: after **P**-assertion of $H_1 \wedge H_2 \cdots \wedge H_n \supset A$, the player **O** asserts the formula $H_1 \wedge H_2 \wedge \cdots \wedge H_n$. **P** attacks this latter formula obliging **O** to assert each H_i for $i \in \{1, \ldots n\}$, then:

- if A is an atomic formula then, since there is a winning U-game for A, A must unfold to one of the H_i i.e., there is an $i \in \{1, \ldots n\}$ such that $H_i = \forall x_1, \ldots \forall x_n((B \supset A_1) \wedge$

$(A_1 \supset B))$ and $A = A_1[t_1/x_1, \ldots t_n/x_n]$ where each t_i is a term. **P** attacks the **O**-assertion of the universally quantified formula H_i and oblige **O** to assert the formula $H_i[t_1/x_1]$. Subsequently, she attacks the **O**-assertion of $H_i[t_1/x_1]$ and obliges **O** to assert the formula $H_i[t_1/x_1][t_2/x_2]$ and so on; After $2n$ steps **O**-asserts $[(A_1 \supset B) \wedge (B \supset A_1)][t_i/\vec{x_i}]$. The player **P** forces **O** to assert $B \supset A_1[t_i/\vec{x_i}]$. We can suppose, without loss of generality, that $B_1 = B[t_i/\vec{x_i}]$ is a non-atomic formula. If B_1 was atomic, since there is a U-game for A and A unfolds to B_1, B_1 itself must unfold to another formula C and we can repeat the same line of reasoning as above. We thus let **P** attack the **O**-assertion of $B_1 \supset A$ by asserting, in turn, B_1. We continue the game by letting **O** assert A as a defense against this latter **P** attack. Finally, we let **P** assert A;

- if A is non-atomic we let **P** assert A as a defense against the **O**-assertion of $H_1 \wedge H_2 \wedge \cdots \wedge H_n$ and we play the game $\mathcal{G}_r$ as the U-game $\mathcal{G}$ 'below' A with the following differences:

    1. suppose that **P** asserts (in the U-game $\mathcal{G}$) an atomic formula Q, and that this assertion is not a repetition. This means that Q unfold to some formula B i.e., there is an $i \in \{1, \ldots n\}$ such that $H_i = \forall x_1, \ldots x_n[(B_1 \supset Q_1) \wedge (Q_1 \supset B_1)]$, $Q = Q_1[t_i/\vec{x_i}]$ and $B = B_1[t_i/\vec{x_i}]$. In the game $\mathcal{G}_r$, **P** attacks the assertion of H_i and obliges **O** to assert $H_i[t_1/x_1]$, then she attacks $H_i[t_1/x_1]$ and obliges **O** to assert $H_i[t_1/x_1, t_2/x_2]$ and so on. Once **O** has asserted $H_i[t_i/\vec{x_i}] = (B \supset Q) \wedge (Q \supset B)$, **P** forces **O** to assert $B \supset Q$, then she attack this latter assertion by a move $(?, B)$. The player **O** finally assert Q and **P** asserts Q in turn. The game $\mathcal{G}_r$ continues as $\mathcal{G}$ below the **P**-assertion of Q;
    2. Suppose that (in the U-game $\mathcal{G}$) **O** asserts some formula B as a defense against a **P** attack of the form $(?, \mathcal{U})$, this latter attack being directed against an **O** assertion of some atomic formula P. As usual this means that there is a formula H_i in $\mathcal{AX}$ such that $H_i = \forall \vec{x_i}[(B_1 \supset Q_1) \wedge (Q_1 \supset B_1)]$ and $B = B_1[t_i/\vec{x_i}]$. The regular game $\mathcal{G}_r$ is constructed as follows: after **O** assertion of Q, **P** attacks the formula H_i (in the usual way detailed above) until **O** is forced to assert $(B \supset Q) \wedge (Q \supset B)$. The player **P** forces **O** to assert $Q \supset B$ and then she attacks this latter assertion by asserting, in turn, the formula Q (remark that **O** has already asserted Q). This **P** attack forces **O** to assert B and the game $\mathcal{G}_r$ continues as U-game $\mathcal{G}$ below B.

□

Lemma 6.2. *Let $H_1, \ldots, H_n$ be formulas of the form $\forall x_1, \ldots, \forall x_m(Q \iff C)$ where P is an atomic formula and C is a formula. Let A be a formula and $\mathcal{G}_r$ be a regular game won by* **P** *for the formula $H_1 \wedge \cdots \wedge H_n \supset A$. There is a U-game $\mathcal{G}'$ won by* **P** *for A if $\{H_1, \ldots, H_n\} \subseteq \mathcal{AX}$.*

Proof. Recall that a game is an augmented sequence=(σ, ϕ) where σ is a sequence of moves and ϕ a pointing function. We say that a move $\mathfrak{m}$ is hereditary enabled by a move if $\mathfrak{n}$ is the transitive, reflexive closure of the relation $\mathfrak{m}$ is enabled by $\mathfrak{n}'$. The regular game $\mathcal{G}_r$ starts by a **P**-assertion of $H_1 \wedge \cdots \wedge H_n \supset C$. The second move $\mathfrak{m}_1$ is $(?, H_1 \wedge \cdots \wedge H_n)$. Since the formulas $H_1, \ldots H_n$ are elements of $\mathcal{AX}$ and $\mathcal{AX}$ is consistent, the move $\mathfrak{m} = (!, A)$ belongs to the regular game $\mathcal{G}_r$. Consider the set of moves:

$$\mathcal{M}_{\mathfrak{m}_2} = \{\mathfrak{n} \in \mathcal{G}_r \,|\, \mathfrak{n} \text{ is hereditary enabled by } \mathfrak{m}_2\}$$

And the difference $\mathcal{M}_{\mathfrak{m}_2} - \mathcal{M}_{(!,A)}$ where $\mathcal{M}_{(!,A)}$ is the set of moves hereditary enabled by the move $\mathfrak{m} = (!, A)$. We delete from $\mathcal{G}_r$ all moves that belongs to the aforementioned difference, obtaining an augmented sequence (ρ, ϕ) in which all moves belong to $\mathcal{M}_{(!,A)}$. To transform (ρ, ϕ) in an U-game $\mathcal{G}$ won by **P** for A we use the unfold attacks and defenses to simulate the deductive links that are given by the formulas $H_1, \ldots H_n$.

□

The notion of strategy for U-games is defined exactly as in subsection 3.2.7 (definition 3.5). We say that a formula $H \in \mathcal{AX}$ is used in a strategy $\mathcal{S}$ iff H is used in some game $\mathcal{G} \in \mathcal{S}$. The following theorem directly follows from the two above lemmas.

Theorem 6.1. *Let $\mathcal{AX}$ be a set of formulas of the form $\forall x_1, \ldots \forall x_m(P \iff C)$ where P is an atomic formula and C is a formula. Let A be an arbitrary formula. There is a winning U-strategy $\mathcal{S}$ for A if and only if there is a winning strategy $\mathcal{T}$ for $\bigwedge \Gamma \supset A$ where Γ is the finite subset of formulas of $\mathcal{AX}$ that are used in $\mathcal{S}$.*

6.4 Textual entailment and U-strategies

Consider a first order language $\mathcal{L}$ in which there are four unary predicates *cost-of-living*, *soared*, *raised*, *quickly*. Suppose that the set $\mathcal{AX}$ as the following formula as element

$$\forall\, x_1[\mathit{soared}\,(x_1) \iff \mathit{raised}\,(x_1) \wedge \mathit{quickly}\,(x_1)]$$

the logical form of (6.21) and (6.22) are respectively

(6.23) $F_1 = \exists\, x_1\, [\mathit{cost\text{-}of\text{-}living}\,(x_1) \wedge \mathit{soared}\,(x_1)]$

(6.24) $F_2 = \exists\, x_2\, [\mathit{cost\text{-}of\text{-}living}\,(x_2) \wedge \mathit{raised}\,(x_2)]$

We show a winning U-strategy for $F_1 \supset F_2$:

$\mathfrak{m}_0 = (!, \exists\, x_1\, [\mathit{cost\text{-}of\text{-}living}\,(x_1) \wedge \mathit{soared}\,(x_1)] \supset \exists\, x_2\, [\mathit{cost\text{-}of\text{-}living}\,(x_2) \wedge \mathit{raised}\,(x_2)])$	
$\mathfrak{m}_1 = (?, \exists\, x_1\, [\mathit{cost\text{-}of\text{-}living}\,(x_1) \wedge \mathit{soared}\,(x_1)])$	$\mathfrak{m}_0$
$\mathfrak{m}_2 = (?, \exists)$	$\mathfrak{m}_1$
$\mathfrak{m}_3 = (!, [\mathit{cost\text{-}of\text{-}living}\,(y) \wedge \mathit{soared}\,(y)])$	$\mathfrak{m}_2$
$\mathfrak{m}_4 = (?, \wedge_1)$	$\mathfrak{m}_3$
$\mathfrak{m}_5 = (!, \mathit{cost\text{-}of\text{-}living}\,(y))$	$\mathfrak{m}_4$
$\mathfrak{m}_6 = (?, \wedge_2)$	$\mathfrak{m}_3$
$\mathfrak{m}_7 = (!, \mathit{soared}\,(y))$	$\mathfrak{m}_6$
$\mathfrak{m}_8 = (?, \mathcal{U})$	$\mathfrak{m}_7$
$\mathfrak{m}_9 = (!, \mathit{raised}\,(y) \wedge \mathit{quickly}\,(y))$	$\mathfrak{m}_8$
$\mathfrak{m}_{10} = (?, \wedge_1)$	$\mathfrak{m}_9$
$\mathfrak{m}_{11} = (!, \mathit{raised}\,(y))$	$\mathfrak{m}_{10}$
$\mathfrak{m}_{12} = (!, \exists\, x_2\, [\mathit{cost\text{-}of\text{-}living}\,(x_2) \wedge \mathit{raised}\,(x_2)])$	$\mathfrak{m}_1$
$\mathfrak{m}_{13} = (?, \exists)$	$\mathfrak{m}_{12}$
$\mathfrak{m}_{14} = (!, \mathit{cost\text{-}of\text{-}living}\,(y) \wedge \mathit{raised}\,(y))$	$\mathfrak{m}_{13}$

$\mathfrak{m}_{15} = (?, \wedge_1)$	$\mathfrak{m}_{14}$
$\mathfrak{m}_{16} = (!, \mathit{cost\text{-}of\text{-}living}\,(y))$	$\mathfrak{m}_{15}$

$\mathfrak{m}_{15} = (?, \wedge_2)$	$\mathfrak{m}_{14}$
$\mathfrak{m}_{16} = (!, \mathit{raised}\,(y))$	$\mathfrak{m}_{15}$

Let us now consider a more complex example:

(6.25) Some patient has aphasia and is a child.

(6.26) Every boy or girl suffering from mutism consult a psychiatrist.

(6.27) Does some patient consult a doctor? [Yes]

let $\mathcal{L}$ be a first order language. Let *has-aphasia*, *patient*, *child*, *boy*, *girl*, *speak*, and *write* be unary predicates. Suppose that the elements of $\mathcal{AX}$ are the following formulas

$$\forall x_1\, [(\mathit{has\text{-}aphasia}\,(x_1)) \iff (\neg \mathit{speak}\,(x_1) \wedge \neg \mathit{write}\,(x_1))]$$

$$\forall\, x_2\, [child\,(x_2) \iff (boy\,(x_2) \lor girl\,(x_2))]$$

$$\forall\, x_3\, [has\text{-}mutism\,(x_3) \iff \neg speak\,(x_3)]$$

$$\forall\, x_4\, [psychiatrist\;\,(x_4) \iff (doctor\,(x_4) \land treats\text{-}mental\text{-}issues\,(x_4))]$$

The logical formulas corresponding to (6.25), (6.26) and to the assertion implicit in (6.27) are:

(6.28) Some patient has aphasia and is a child.
$F_1 = \exists\, y_1\, [patient\,(y_1) \land (has\text{-}aphasia\,(x_1) \land child\,(x_1))]$

(6.29) Every boy or girl suffering from mutism consult a psychiatrist.
$F_2 = \forall\, y_2\, [(boy\,(y_2) \lor girl\,(y_2) \land has\text{-}mutism\,(y_2)) \supset$
$\exists\, y_3\, (psychiatrist\,(y_3) \;\land consult\,(y_2, y_3))]$

(6.30) Some patient consults a doctor.
$F_3 = \exists\,(y_4)\, [patient\,(y_4) \land \exists y_5\, (doctor\, y_5 \land consult\,(y_4, y_5))]$

Below we show a winning U-strategy for the formula $F_1 \land F_2 \supset F_3$. For typographical reason the strategy for this formula will be split in two. In the strategy we abbreviate the predicates in the usual manner.

$\mathfrak{m}_0 = (!, F_1 \wedge F_2 \supset F_3)$	
$\mathfrak{m}_1 = (?, F_1 \wedge F_2)$	$\mathfrak{m}_1$
$\mathfrak{m}_2 = (?, \wedge_1)$	$\mathfrak{m}_1$
$\mathfrak{m}_3 = (!, \exists y_1\, [pa\,(y_1) \wedge (h\text{-}a\,(x_1) \wedge ch\,(x_1))])$	$\mathfrak{m}_2$
$\mathfrak{m}_4 = (?, \exists)$	$\mathfrak{m}_3$
$\mathfrak{m}_5 = (!, pa\,(z) \wedge (h\text{-}a\,(z) \wedge ch\,(z)))$	$\mathfrak{m}_4$
$\mathfrak{m}_6 = (?, \wedge_2)$	$\mathfrak{m}_4$
$\mathfrak{m}_7 = (!, h\text{-}a\,(z) \wedge ch\,(z))$	$\mathfrak{m}_6$
$\mathfrak{m}_8 = (?, \wedge_1)$	$\mathfrak{m}_7$
$\mathfrak{m}_9 = (!, h\text{-}a\,(z))$	$\mathfrak{m}_8$
$\mathfrak{m}_{10} = (?, \mathcal{U})$	$\mathfrak{m}_9$
$\mathfrak{m}_{11} = (!, \neg sp\,(z) \wedge \neg wr\,(z))$	$\mathfrak{m}_{10}$
$\mathfrak{m}_{12} = (?, \wedge_1)$	$\mathfrak{m}_{11}$
$\mathfrak{m}_{13} = (!, \neg sp\,(y))$	$\mathfrak{m}_{12}$
$\mathfrak{m}_{14} = (?, \wedge_2)$	$\mathfrak{m}_7$
$\mathfrak{m}_{15} = (!, ch\,(z))$	$\mathfrak{m}_{12}$
$\mathfrak{m}_{16} = (?, \mathcal{U})$	$\mathfrak{m}_{13}$
$\mathfrak{m}_{17} = (!, bo\,(z) \vee gi\,(z))$	$\mathfrak{m}_{14}$
$\mathfrak{m}_{18} = (?, \wedge_2)$	$\mathfrak{m}_1$
$\mathfrak{m}_{19} = (!, \forall y_2[(bo\,(y_2) \vee gi\,(y_2) \wedge h\text{-}m\,(y_2)) \supset \exists y_3\,(ps\,(y_3) \wedge co\,(y_2, y_3))])$	$\mathfrak{m}_{17}$
$\mathfrak{m}_{20} = (?, \forall[z/y_3])$	$\mathfrak{m}_{19}$
$\mathfrak{m}_{21} = (!, (bo\,(z) \vee gi\,(z)) \wedge h\text{-}m\,(z)) \supset \exists y_3\,(ps\,(y_3) \wedge co\,(z, y_3)))$	$\mathfrak{m}_{20}$
$\mathfrak{m}_{22} = (?, (bo\,(z) \vee gi\,(z)) \wedge h\text{-}m\,(z))$	$\mathfrak{m}_{21}$

$\mathfrak{m}_{23} = (?, \wedge_1)$	$\mathfrak{m}_{22}$
$\mathfrak{m}_{24} = (?, \vee)$	$\mathfrak{m}_{15}$

$\mathfrak{m}_{23} = (?, \wedge_2)$	$\mathfrak{m}_{22}$
$\mathfrak{m}_{24} = (!, h\text{-}m\,(z))$	$\mathfrak{m}_{23}$
$\mathfrak{m}_{25} = (?, \mathcal{U})$	$\mathfrak{m}_{24}$
$\mathfrak{m}_{26} = (!, \neg sp\,(z))$	$\mathfrak{m}_{25}$
$\mathfrak{m}_{27} = (?, sp\,(z))$	$\mathfrak{m}_{26}$
$\mathfrak{m}_{28} = (?, sp\,(z))$	$\mathfrak{m}_{13}$
$\mathfrak{m}_{29} = (!, \perp)$	$\mathfrak{m}_2$

$\mathfrak{m}_{25} = (!, bo\,(z))$	$\mathfrak{m}_{24}$
$\mathfrak{m}_{26} = (!, bo\,(z) \vee gi\,(z)$	$\mathfrak{m}_{23}$
$\mathfrak{m}_{27} = (?, \vee)$	$\mathfrak{m}_{26}$
$\mathfrak{m}_{28} = (!, bo\,(z))$	$\mathfrak{m}_{27}$.

$\mathfrak{m}_{25} = (!, gi\,(z))$	$\mathfrak{m}_{24}$
$\mathfrak{m}_{26} = (!, bo\,(z) \vee gi\,(z)$	$\mathfrak{m}_{23}$
$\mathfrak{m}_{27} = (?, \vee)$	$\mathfrak{m}_{26}$
$\mathfrak{m}_{28} = (!, gi\,(z))$	$\mathfrak{m}_{27}$.

$\mathfrak{m}_0 = (!, F_1 \wedge F_2 \supset F_3)$	
$\mathfrak{m}_1 = (?, F_1 \wedge F_2)$	$\mathfrak{m}_1$
$\mathfrak{m}_2 = (?, \wedge_1)$	$\mathfrak{m}_1$
$\mathfrak{m}_3 = (!, \exists y_1\, [pa\,(y_1) \wedge (h\text{-}a\,(x_1) \wedge child\,(x_1))])$	$\mathfrak{m}_2$
$\mathfrak{m}_4 = (?, \exists)$	$\mathfrak{m}_3$
$\mathfrak{m}_5 = (!, pa\,(z) \wedge (h\text{-}a\,(z) \wedge child\,(z)))$	$\mathfrak{m}_4$
$\mathfrak{m}_6 = (?, \wedge_2)$	$\mathfrak{m}_4$
$\mathfrak{m}_7 = (!, h\text{-}a\,(z) \wedge ch\,(z))$	$\mathfrak{m}_6$
$\mathfrak{m}_8 = (?, \wedge_1)$	$\mathfrak{m}_7$
$\mathfrak{m}_9 = (!, h\text{-}a\,(z))$	$\mathfrak{m}_8$
$\mathfrak{m}_{10} = (?, \mathcal{U})$	$\mathfrak{m}_9$
$\mathfrak{m}_{11} = (!, \neg sp\,(z) \wedge \neg write\,(z))$	$\mathfrak{m}_{10}$
$\mathfrak{m}_{12} = (?, \wedge_1)$	$\mathfrak{m}_{11}$
$\mathfrak{m}_{13} = (!, \neg sp\,(y))$	$\mathfrak{m}_{12}$
$\mathfrak{m}_{14} = (?, \wedge_2)$	$\mathfrak{m}_7$
$\mathfrak{m}_{15} = (!, child\,(z))$	$\mathfrak{m}_{12}$
$\mathfrak{m}_{16} = (?, \mathcal{U})$	$\mathfrak{m}_{13}$
$\mathfrak{m}_{17} = (!, bo\,(z) \vee gi\,(z))$	$\mathfrak{m}_{14}$
$\mathfrak{m}_{18} = (?, \wedge_2)$	$\mathfrak{m}_1$
$\mathfrak{m}_{19} = (!, \forall\, y_2[(bo\,(y_2) \vee gi\,(y_2) \wedge h\text{-}m\,(y_2)) \supset \exists\, y_3\,(ps\,(y_3)\ \wedge co\,(y_2, y_3))])$	$\mathfrak{m}_{17}$
$\mathfrak{m}_{20} = (?, \forall[z/y_3])$	$\mathfrak{m}_{19}$
$\mathfrak{m}_{21} = (!, (bo\,(z) \vee gi\,(z)) \wedge h\text{-}m\,(z)) \supset \exists\, y_3\,(ps\,(y_3)\ \wedge co\,(z, y_3)))$	$\mathfrak{m}_{20}$
$\mathfrak{m}_{22} = (?, (bo\,(z) \vee gi\,(z)) \wedge h\text{-}m\,(z))$	$\mathfrak{m}_{21}$
$\mathfrak{m}_{23} = (!, \exists\, y_3\,(ps\,(y_3) \wedge co\,(z, y_3)))$	$\mathfrak{m}_{22}$
$\mathfrak{m}_{24} = (?, \exists)$	$\mathfrak{m}_{23}$
$\mathfrak{m}_{25} = (!, ps\,(w) \wedge co\,(z, y_3))$	$\mathfrak{m}_{24}$
$\mathfrak{m}_{26} = (?, \wedge_1)$	$\mathfrak{m}_{25}$
$\mathfrak{m}_{27} = (!, ps\,(w))$	$\mathfrak{m}_{26}$
$\mathfrak{m}_{28} = (?, \mathcal{U})$	$\mathfrak{m}_{27}$
$\mathfrak{m}_{29} = (?, do\,(w) \wedge t\text{-}m\text{-}h\,(w))$	$\mathfrak{m}_{28}$
$\mathfrak{m}_{30} = (?, \wedge_1)$	$\mathfrak{m}_{29}$
$\mathfrak{m}_{31} = (do\,(w))$	$\mathfrak{m}_{30}$
$\mathfrak{m}_{32} = (!, \exists\,(y_4)\,[pa\,(y_4) \wedge \exists y_5\,(do\, y_5 \wedge co\,(y_4, y_5))])$	$\mathfrak{m}_1$
$\mathfrak{m}_{33} = (?, \exists)$	$\mathfrak{m}_{32}$
$\mathfrak{m}_{34} = (!, pa\,(z) \wedge \exists y_5\,(do\,(y_5) \wedge co\,(z, y_5))$	

$\mathfrak{m}_{35} = (?, \wedge_1)$	$\mathfrak{m}_{34}$
$\mathfrak{m}_{36} = (!, \wedge_1)$	$\mathfrak{m}_5$
$\mathfrak{m}_{37} = (!, pa\,(z)$	$\mathfrak{m}_{36}$
$\mathfrak{m}_{38} = (!, pa\,(z))$	$\mathfrak{m}_{35}$

$\mathfrak{m}_{35} = (?, \wedge_2)$	$\mathfrak{m}_{34}$
$\mathfrak{m}_{36} = (!, \exists\, y_5\,(do\,(y_5) \wedge co\,(z, y_5))$	$\mathfrak{m}_{35}$
$\mathfrak{m}_{37} = (?, \exists)$	$\mathfrak{m}_{36}$
$\mathfrak{m}_{38} = (!, do\,(w) \wedge co\,(z, w))$	$\mathfrak{m}_{37}$

$\mathfrak{m}_{39} = (?, \wedge_1)$	$\mathfrak{m}_{38}$
$\mathfrak{m}_{40} = (?, do\,(w))$	$\mathfrak{m}_{39}$

$\mathfrak{m}_{39} = (?, \wedge_2)$	$\mathfrak{m}_{38}$
$\mathfrak{m}_{40} = (!, co\,(y, w))$	$\mathfrak{m}_{39}$

6.5 Conclusion

In this chapter we have seen how we can solve simple instances of the textual entailment recognition problem using type logical grammars and dialogical logic. The examples are taken from the FraCas dataset and covers simple examples of textual entailment recognition. We also presented dialogical games that are better suited to the treatment of inferences in which the meaning of words is essential. The meaning of certain words (atomic proposition) is given to the two players in the form of explicit definitions. The players can unfold these definitions. In doing so, inferential links that are due solely to the meaning of words can be used in dialogic games. Unfold (and Fold) rules were first studied by Prawitz in the context of Natural Deduction systems [117], then by Schroeder-Heister in the context of the sequent calculus [126] and have been developed extensively by Dowek [40]. Unfold rules permits to considerably reduce the size of formal proofs and for this reason versions of these types of rules are the basis of many approaches to automated proof search in the sequent calculus e.g., [39, 17].

Chapter 7

DiaLogical Games for anaphora and ellipsis resolution

Abstract
In this chapter we develop a proof theoretic approach to anaphora and ellipsis resolution. Our approach is based on the introduction of a new quantifier $\mathcal{A}xF$ whose meaning is specified in the context of particular dialogical games. The bound variable of the quantifier can only be instantiated with terms that already appears in the game. We present in detail some examples of entailment recognition involving anaphora and ellipsis resolution. Much of the content of this chapter already appears in in [24] .

7.1 Introduction

In this chapter we propose a novel solution to anaphora and ellipsis resolution. A meaning η is anaphoric just in case it cannot occur without the co-occurrence of another meaning η' from which it is indistinguishable, dubbed its antecedent. One obvious example of anaphora in English is 'he' in 'John believes he proved the theorem', where 'he' is understood to be John. But ellipsis —the absence of some subcategorized for expression— is a variety of anaphora too [111]. Anaphora resolution is the process by which the meaning of anaphoric expressions is identified with the meaning of their antecedents. Since the space of possible antecedents is in principle wide, but people find the correct (=intended) referent quickly and often without error, anaphora resolution poses an interesting puzzle for any theory of linguistic semantics.

We propose that anaphora can be accounted for by means of proof-theoretic methods in (multi-sorted) first order dialogical logic. We introduce a new quantifier $\mathcal{A}$, which is intuitively understood to correspond to quantification over non-fresh terms. Given a context p which includes a proposition with $\mathcal{A}$ binding an occurrence x, there is a proof of some proposition q which is the result of substituting some term t in the context of p for the x bound by $\mathcal{A}$. This proof-theoretic approach is extended to resolution of post-auxilliary ellipsis (PAE) [104] by the introduction of events into the inventory of sorts. In short, PAE under the present theory is event anaphora.

Organization of the Chapter

In the next section, we introduce the phenomena under study–pronouns and anaphoras– and our conception of their resolution. In the subsequent section we introduce a dialogical logic system for anaphora resolution. Section 7.4 shows how we can solve textual entailment recognition problems involving anaphoric reference by means of our dialogical system. In Section 7.5 we discuss post-auxiliary ellipsis and event semantics. The subsequent section 7.6 presents examples of entailment problems that are solved by our methodology. In section 7.7 we succinctly discuss other approaches to anaphora and ellipsis resolution and the role of sorting in our system. Section 7.8 concludes.

7.2 Pronouns

Since the pioneering work of [67, 80], pronouns in natural language semantics have received much attention. The conception of pronouns in these theories is under girded by the view that some expressions introduce semantic objects dubbed discourse referents (drefs) and others refer back to these drefs. Much of the work in this tradition is concerned with when an antecedent dref is 'accessible' for a pronoun to be resolved to it. Consider the following:

(7.1) Not everyone smiled. *He had a terrible headache.

(7.2) Someone did not smile. He had a terrible headache.

(7.3) Pedro didn't buy a donkey. *It is grey.

(7.4) Bill bought a donkey. It is grey.

(7.5) Bill didn't visit Sue. She is out.

Where a pronoun that cannot reefer to an entity introduced in the previous sentence is preceded by *. According to DRT [81], negation blocks the introduction of (some) drefs in its scope, which are otherwise introduced by indefinites. Names, unlike indefinites, project the dref they introduce outside the scope of negation. Consequently, while 'she' can refer to Sue in (7.5), and 'it' can refer to Bill's donkey in (7.4), 'it' cannot be understood to refer to the donkey Pedro didn't buy in (7.3), which needn't even exist. However, since the scope of indefinites is not restricted to the complement of negation, there is a second reading of (7.3) on which the indefinite outscopes the negation. The two scopes of the indefinite are represented in first order logic below, where p is 'Pedro':

(7.6) $\neg\exists x.donkey(x) \wedge buy(p, x)$

(7.7) $\exists x.donkey(x) \wedge \neg buy(p, x)$

The second reading is sometimes said to be 'specific' in that it is felicitous (=judged coherent) in the context in which Bill bought some gray donkey which Pedro didn't buy. Since there is a gray donkey in this context, one could felicitously utter the discourse in (7.3). A variety of other contexts where drefs don't seem to project have exercised semanticists. Some drefs don't project outside the scope of 'if,then' expressions; others don't project outside the scope of 'every'. Modelling the contexts in which drefs do and don't project is one of the primary projects of those theories of semantics dubbed 'dynamic', which depart from the static Montagovian [105] picture of meaning in viewing the meaning of an utterance first and foremost in terms of how it can change the context of a discourse.

7.3 Dialogical games for Anaphora and Ellipsis resolution: $\mathcal{A}$-games

As we have anticipated in the introduction of this chapter, the solution that we propose rely on the introduction of a new quantifier $\mathcal{A}$. Formulas that are asserted by the Opponent trough a dialogical game $\mathcal{G}$ are considered to be granted; this is the meaning of the formal rule 3 in the definition 3.3 of game. The Proponent can safely assert atomic formulas that the Opponent already asserted because the truth of those formulas is taken for granted i.e., Opponent's assertion forms the *context* or *common ground* of the game. Our idea to treat anaphora resolution is thus simple. Whenever the assertion of formula $F = \mathcal{A} y G$ is attacked, the defense must be the assertion of $G[\mathbf{k}/y]$ where $\mathbf{k}$ is a constant that occurs in some formula that has been previously asserted by the Opponent. The constant k represent an entity whose existence — in the context of the game— is taken for granted.

Given a set S_0 of basic sorts, sorts are defined by the following grammar

$$S := \{t\} \mid S_0 \mid S \rightarrow S$$

where t is the sort of booleans. Consider a multi-sorted first order language $\mathcal{L}$ in which the set $\mathcal{T}$ of terms of $\mathcal{L}$ is the union of the set $\mathfrak{C}$ of sorted constant (for any base sort $s_i \in S_0$ we have arbitrarily many constants of that sort) and of the set $\mathcal{V}$ of sorted variables (for any base sort $s_i \in S_0$ we have countably many variables of that sort). Let $\mathcal{R}$ be an at most countable set of predicate variables. To each predicate variable we associate a sort $s_1 \rightarrow (s_2 \rightarrow \cdots \rightarrow (s_n \rightarrow t))$ with $n \geq 0$ and for all $i \in \{1, \ldots n\}$ $s_i \in S_0$. The set At of atomic formulas has for elements expressions $P(t_1, \ldots t_n)$ where P is a predicate variable with sort $s_1 \rightarrow (s_2 \rightarrow \cdots \rightarrow (s_n \rightarrow t))$ and each t_i for $i \geq 0$ is of sort s_i.

Let $\wedge, \vee, \supset, \forall, \exists$ be the usual connectives and quantifiers of multisorted first order logic, let $\mathcal{A}$ be our new quantifier. Formulas are defined by the following grammar

$$\mathcal{F} = At \mid \mathcal{F} \wedge \mathcal{F} \mid \mathcal{F} \vee \mathcal{F} \mid \mathcal{F} \supset \mathcal{F} \mid \forall x^{\alpha} \mathcal{F} \mid \exists x^{\alpha} \mathcal{F} \mid \mathcal{A} x^{\alpha} \mathcal{F}$$

As usual the negation of a formula is defined as $\neg F = F \supset \bot$. In this section the set Aux of auxiliary symbols is defined ad the smallest set with elements $\wedge_1, \wedge_2, \vee, \exists, \mathcal{A}$ and the expression $\forall[\mathbf{k}^{\alpha}/x^{\alpha}]$ for all constants $\mathbf{k}$ and variables x. The function Arg that maps non-atomic formulas to sets of pairs of auxiliary symbols and formulas is defined as follows:

$$\begin{array}{ll}
Arg(A \supset B) & = \{(A, B)\} \\
Arg(A \wedge B) & = \{(\wedge_1, A), (\wedge_2, B)\} \\
Arg(A \vee B) & = \{(\vee, A), (\vee, B)\} \\
Arg(\forall x^{\alpha} A) & = \{(\forall[\mathbf{k}^{\alpha}/x], A[\mathbf{k}^{\alpha}/x]) \mid \mathbf{k}^{\alpha} \in \mathfrak{C}\} \\
Arg(\exists x^{\alpha} A) & = \{(\exists, A[\mathbf{k}^{\alpha}/x]) \mid \mathbf{k}^{\alpha} \in \mathfrak{C}\} \\
Arg(\mathcal{A} x^{\alpha} A) & = \{(\mathcal{A}, A[\mathbf{k}^{\alpha}/x]) \mid \mathbf{k}^{\alpha} \in \mathfrak{C}\}
\end{array}$$

A defense move is a pair $(!, A)$ where A is a formula. An attack move is a pair $(?, s)$ where s is either a formula or an auxiliary symbol. The notion of justified attack move and justified defense move are defined as in definition 3.2 with the only difference that what count as a question about a formula and what counts as an answer about that question are now defined with respect to the function *Arg* defined above.

Let (ρ, ϕ) be an augmented sequence, we say that a formula A appears in the augmented sequence iff there is a move $m \in \rho$ that asserts A. We say that a constant c appears in ρ whenever $\mathtt{k}$ occurs in some asserted formula, or there is a move $m = (?, \forall[\mathtt{k}/x])$ in ρ. Fix an enumeration $(\mathtt{k}_i)_{i \in I}$ of constants of $\mathfrak{C}$. The definition of games for anaphora is entirely similar to the definition 3.3 of "plain" games. However, we recast it for the sake of intelligibility

Definition 7.1 ($\mathcal{A}$-Game)**.** *An Anaphora Game ($\mathcal{A}$-Game for short) $\mathcal{G}$ for a formula F is an augmented sequence (ρ, ϕ) where $\rho = \mathfrak{m}_0 \cdots \mathfrak{m}_n \cdots$ is non-empty and such that*

1. *$m_0 = (!, F)$ and for all $i > 0$ the move m_i is justified;*
2. *$\phi(m_i) = m_{i-1}$ if i is odd, $\phi(m_i) = m_j$ with j odd if i is even;*
3. *if $m_i = (\star, B)$ with B atomic formula and i even then m_i is a reprise and $B \neq \bot$;*
4. *if m_i is an attack move of the form $(?, \forall[\mathtt{k}^\alpha/x])$ and i is odd then $\mathtt{k}^\alpha$ is the first constant in the enumeration $(\mathtt{k}_i)_{i \in I}$ that does not appear in the prefix of ρ ending in m_{i-1};*
5. *if $m_i = (!, B[\mathtt{k}^\alpha/x])$ is a defense move, i is odd and m_{i-1} is of the form $(?, \exists)$ then $\mathtt{k}^\alpha$ is the first constant in the enumeration $(\mathtt{k}_i)_{i \in I}$ that does not appear in the prefix of ρ ending in m_{i-1};*
6. *if $m_i = (!, B[\mathtt{k}^\alpha/x])$ is a defense move and $\phi(\rho_i)$ is of the form $(?, \mathcal{A})$ then there is an assertion move $m_j = (\star, C)$ with $j < i$ and j odd such that $\mathtt{k}^\alpha$ occurs in C.*
7. *if $\mathfrak{m}_i = (?, \mathcal{A})$ and i is even, then there is an assertion move $\mathfrak{m}_j = (\star, C) \in \rho$ and a constant $\mathtt{k}^\alpha$ such that $\mathtt{k}^\alpha$ occurs in C, $j < i$ and j is odd.*

Condition 4 ensures that **P** must instantiate a universal quantifier with a fresh constant. Condition 5 ensures that **O** must do the same thing with existential quantifiers. Condition 6 determines the behavior of the quantifier $\mathcal{A}$ in a game. This quantifier must be instantiated with a constant that appears in the *common ground* of the game where the common ground is the set of formulas asserted by **O** through the game. Finally, condition 7 assures that **P** can attack a move that asserts a formula $\mathcal{A}xA$ only if **O** can defend against this attack.

$\mathcal{A}$-games winning conditions are defined exactly as in definition 3.4. An A-strategy $\mathcal{S}$ for a formula F is defined as in definition 3.5 with the only difference that $\mathcal{S}$ will be a prefix-closed set of $\mathcal{A}$-games for F.

7.3.1 Properties of $\mathcal{A}$-Games

Propositions 3.1, 3.2, 3.3 and 3.4 of section 3.2.6 holds for $\mathcal{A}$-games. Moreover, we can prove the following

Proposition 7.1. *Let F be a multi-sorted first order formula in which there is no occurrence of the quantifier $\mathcal{A}$. There is a winning $\mathcal{A}$-strategy $\mathcal{S}$ for F if and only if there is a winning strategy $\mathcal{T}$ for F.*

Proof. It is sufficient to remark that each plain game is a $\mathcal{A}$-game and that if F has no occurrence of $\mathcal{A}$ then each $\mathcal{A}$-game for F is a plain game for F. □

A logic can be defined as a set of formulas. We say that a set of formulas is consistent iff there is a formula B that does not belong to it . Define the set

$$\mathtt{Teor}_{\mathcal{A}} = \{F \text{ is a multi-sorted formula} \mid \text{there is a winning } \mathcal{A}\text{-strategy for } F\}$$

Because of proposition 7.1 `Teor` is a consistent set. Thus the logic defined by the set of formulas for which there is a winning A-strategy is consistent. Let Γ be a set of formulas and F a formula. We say that F is an $\mathcal{A}$-consequence of Γ iff there is a winning $\mathcal{A}$-strategy for $\bigwedge\Gamma \supset F$. The consequence relation we have just defined is *non-monotonic* i.e., there are Δ and Γ and F such that F is $\mathcal{A}$-consequence of Δ, $\Delta \subseteq \Gamma$ and there is no winning $\mathcal{A}$-strategy for $\bigwedge\Gamma \supset F$. For a simple example, consider $\Delta = \{P(\mathtt{k}), \mathcal{A}yQ(y)\}$ where $\mathtt{k}_1$ is a constant. The formula $P(\mathtt{k}_1) \wedge Q(\mathtt{k})$ is an $\mathcal{A}$-consequence of Δ, while it is not a $\mathcal{A}$-consequence of $\{P(\mathtt{k}_2), P(\mathtt{k}_1), \mathcal{A}yQ(y)\}$, where $\mathtt{k}_2$ is a constant different from $\mathtt{k}_1$.

7.4 Textual Entailment Recognition and Anaphora Resolution

We present some examples of textual entailment recognition involving anaphoras, and we show how they can be solved using the notions defined in the previous subsection. We choose some examples of anaphora resolution that we find representative. We do not claim that our method can solve all example of anaphora resolution. The extension of this class of problems is unclear, and we content our self to show that some inference problem involving anaphora resolution can be solved using $\mathcal{A}$-games.

7.4.1 First Example Involving Anaphoras

In subsection 7.2 we saw that in sentence (7.1) i.e.,

(7.8) Not everyone smiled. He had an headache.

The pronoun *He* cannot be solved by the subject of *Not everyone smiled*. On the contrary the pronoun in sentence (7.2) i.e.,

(7.9) Someone did not smile. He had an headache.

The pronoun *He* can be solved by *someone*. What is remarkable in this two examples is that the sentences *Someone did not smile* and *Not everyone smiled* are logically equivalent in classical logic. In fact their logical form are, respectively,

$$\neg\forall\, x_1\, smile\,(x_1)$$

$$\exists\, x_1\, \neg smile\,(x_1)$$

In the logic we have introduced in the preceding section the sentence *He had an headache* can be rendered as

$$\mathcal{A}\, y_1\ \textit{had-headache}\,(y_1)$$

[1]

One way of seeing the fact that the pronoun *He* cannot be solved in (7.8) but can be solved in (7.9) is that from (7.8) we cannot derive the sentence *there is someone who did not smile and had a terrible headache* while we can derive it from (7.9). Let us convince our self that there is no winning $\mathcal{A}$-strategy for the formula

$$F = [\neg\forall\, x_1\, smile\,(x_1) \wedge \mathcal{A}\, y_1\ \textit{had-headache}\,(y_1)] \supset \exists\, x_2\, (\neg smile\,(x_2) \wedge \textit{had-headache}\,(x_2))$$

[1] Here the sorting is not strictly required and we suppose that all variables and constants have the same sort

we remark immediately that the formula $\mathcal{A}\, y_1\, \textit{had-headache}\,(y_1)$ is a negative sub-formula of F. By condition 6 in definition 7.1 this means that the quantifier $\mathcal{A}$ cannot be instantiated with the same constant that instantiate $\forall\, x_1\, \textit{smile}\,(x_1)$: this latter formula is a positive sub-formula of F and therefore only **P** can assert it; in particular this means that only **P** can assert the formula $\textit{smile}(x_1)[\mathtt{k}/x_1]$ as a defense against an attack $(?, \forall[\mathtt{k}/x_1])$ made by **O** against the **P** assertion of $\forall\, x_1\, \textit{smile}\,(x_1)$. If **P** asserts the formula $\exists\, x_2\, (\neg \textit{smile}\,(x_2) \wedge \textit{had-headache}\,(x_2))$, **O** would force him to instantiate the existential quantifier with a constant $\mathtt{k}'$ and then the strategy will branch: we will have a branch in which **P** must assert the formula $\textit{had-headache}\,(\mathtt{k}')$ to win the game. For what we said and for condition 3 in definition 7.1 a branch in which **P** asserts the formula $\textit{had-headache}\,(\mathtt{k}')$ for some constant $\mathtt{k}'$ cannot exist.

On the contrary we can easily obtain a winning $\mathcal{A}$-strategy for the formula

$$G = [\exists\, x_1\, \neg \textit{smile}\,(x_1) \wedge \mathcal{A}\, y_1\, \textit{had-headache}\,(y_1)] \supset \exists\, x_2\, (\neg \textit{smile}\,(x_2) \wedge \textit{had-headache}\,(x_2))$$

such a strategy is presented in figure 7.1

$\mathfrak{m}_0 = (!, \exists\, x_1\, \neg \textit{smile}\,(x_1) \wedge \mathcal{A}\, y_1\, \textit{had-headache}\,(y_1)] \supset \exists\, x_2\, (\neg \textit{smile}\,(x_2) \wedge \textit{had-headache}\,(x_2)))$	
$\mathfrak{m}_1 = (?, \exists\, x_1\, \neg \textit{smile}\,(x_1) \wedge \mathcal{A}\, y_1\, \textit{had-headache}\,(y_1))$	$\mathfrak{m}_0$
$\mathfrak{m}_2 = (?, \wedge_1)$	$\mathfrak{m}_1$
$\mathfrak{m}_3 = (!, \exists\, x_1\, \neg \textit{smile}\,(x_1))$	$\mathfrak{m}_2$
$\mathfrak{m}_4 = (?, \exists)$	$\mathfrak{m}_3$
$\mathfrak{m}_5 = (!, \neg \textit{smile}\,(\mathtt{k}_1))$	$\mathfrak{m}_4$
$\mathfrak{m}_6 = (?, \wedge_2)$	$\mathfrak{m}_1$
$\mathfrak{m}_7 = (!, \mathcal{A}\, y_1\, \textit{had-headache}\,(y_1))$	$\mathfrak{m}_6$
$\mathfrak{m}_8 = (?, \mathcal{A})$	$\mathfrak{m}_7$
$\mathfrak{m}_9 = (!, \textit{had-headache}\,(\mathtt{k}_1))$	$\mathfrak{m}_8$
$\mathfrak{m}_{10} = (!, \exists\, x_2\, (\neg \textit{smile}\,(x_2) \wedge \textit{had-headache}\,(x_2)))$	$\mathfrak{m}_1$
$\mathfrak{m}_{11} = (?, \exists)$	$\mathfrak{m}_{10}$
$\mathfrak{m}_{12} = (!, \neg \textit{smile}\,(\mathtt{k}_1) \wedge \textit{had-headache}\,(\mathtt{k}_1))$	$\mathfrak{m}_{11}$

$\mathfrak{m}_{13} = (?, \wedge_1)$	$\mathfrak{m}_{12}$
$\mathfrak{m}_{14} = (!, \neg \textit{smile}\,(\mathtt{k}_1))$	$\mathfrak{m}_{13}$
$\mathfrak{m}_{15} = (?, \textit{smile}\,(\mathtt{k}_1))$	$\mathfrak{m}_{14}$
$\mathfrak{m}_{16} = (?, \textit{smile}\,(\mathtt{k}_1))$	$\mathfrak{m}_5$
$\mathfrak{m}_{17} = (!, \bot)$	$\mathfrak{m}_5$

$\mathfrak{m}_{13} = (?, \wedge_2)$	$\mathfrak{m}_{12}$
$\mathfrak{m}_{14} = (!, \textit{had-headache}\,(\mathtt{k}_1))$	$\mathfrak{m}_{13}$

Figure 7.1: Winning $\mathcal{A}$-strategy for the formula G

7.4.2 Second Example Involving anaphoras

We took the following example from the FraCas data-set.

(7.10) Smith attended a meeting.

(7.11) She chaired it.

(7.12) Did Smith chaired a meeting? [Yes]

The answer to (7.12) is affirmative. This means that if (7.10) and (7.11) are true then the sentence "Smith chaired a meeting" must be true. Let h be the sort of human being and o be the sort of inanimate objects. Let *meeting* be a predicate with sort $o \to t$ and *attended* and *chaired* be predicates with sort $h \to (o \to t)$. Finally let $\mathtt{smith}$ be a constant symbol with sort h.

(7.13) Smith attended a meeting.

$$\exists\, {x_1}^{o}\, (\mathit{meeting}\,(x_1) \land \mathit{attended}\,(\mathtt{smith}, x_1))$$

$\mathfrak{m}_0 = (!, [(\exists\, {x_1}^{o}\, (\mathit{me}\,(x_1)) \land \mathit{at}\,(\mathtt{smith}, x_1)) \land (\mathcal{A}\, {x_2}^{h}\, \mathcal{A}\, {x_3}^{o}\, (\, \mathit{ch}\,(x_2, x_3)))] \supset \exists\, {x_4}^{o}\, (\mathit{ch}\,(\mathtt{smith}, x_4)))$	
$\mathfrak{m}_1 = (?, (\exists\, {x_1}^{o}\, (\mathit{me}\,(x_1)) \land \mathit{at}\,(\mathtt{smith}, x_1)) \land (\mathcal{A}\, {x_2}^{h}\, \mathcal{A}\, {x_3}^{o}\, (\, \mathit{ch}\,(x_2, x_3))))$	$\mathfrak{m}_0$
$\mathfrak{m}_2 = (?, \land_1)$	$\mathfrak{m}_1$
$\mathfrak{m}_3 = (!, \exists\, {x_1}^{o}\, (\mathit{me}\,(x_1)) \land \mathit{at}\,(\mathtt{smith}, x_1)))$	$\mathfrak{m}_2$
$\mathfrak{m}_4 = (?, \exists)$	$\mathfrak{m}_3$
$\mathfrak{m}_5 = (!, (\mathit{me}\,(\mathtt{k}_1)) \land \mathit{at}\,(\mathtt{smith}, \mathtt{k}_1))$	$\mathfrak{m}_4$
$\mathfrak{m}_6 = (?, \land_2)$	$\mathfrak{m}_1$
$\mathfrak{m}_7 = (!, \mathcal{A}\, {x_2}^{h}\, \mathcal{A}\, {x_3}^{o}\, (\, \mathit{ch}\,(x_2, x_3)))$	$\mathfrak{m}_6$
$\mathfrak{m}_8 = (?, \mathcal{A})$	$\mathfrak{m}_7$
$\mathfrak{m}_9 = (!, \mathcal{A}\, {x_3}^{o}\, (\, \mathit{ch}\,(\mathtt{smith}, x_3)))$	$\mathfrak{m}_8$
$\mathfrak{m}_{10} = (?, \mathcal{A})$	$\mathfrak{m}_7$
$\mathfrak{m}_{11} = (!\ \mathit{ch}\,(\mathtt{smith}, \mathtt{k}_1))$	$\mathfrak{m}_{10}$
$\mathfrak{m}_{12} = (!, \exists\, {x_4}^{o}\, (\mathit{ch}\,(\mathtt{smith}, x_4))$	$\mathfrak{m}_1$
$\mathfrak{m}_{13} = (?, \exists)$	$\mathfrak{m}_{12}$
$\mathfrak{m}_{14} = (!, \mathit{ch}\,(\mathtt{smith}, \mathtt{k}_1))$	$\mathfrak{m}_{13}$

Figure 7.2: Winning $\mathcal{A}$-strategy for the second example involving anaphoras

(7.14) She chaired it.
$\mathcal{A}\, x_2{}^h\, \mathcal{A}\, x_3{}^o\, (\, chaired\, (x_2, x_3))$

(7.15) Smith chaired a meeting.
$\exists\, x_4{}^o\, (chaired\, (\texttt{smith}, x_4) \wedge meeting\, (x_4))$

Call the formula in (7.13) F_1, the formula in (7.14) F_2 and the formula in (7.15) F_3. Figure 7.2 shows a winning strategy for the formula $F_1 \wedge F_2 \supset F_3$. In the figure, *me* stands for *meeting*, *at* stands for *attended* and *ch* stands for *chaired*.

7.4.3 Third Example Involving anaphoras

(7.16) Smith delivered a report to ITEL.

(7.17) She also delivered them an invoice.

(7.18) She also delivered them a project proposal.

(7.19) Did Smith delivered a report, an invoice and a project proposal to ITEL? [Yes]

The answer to the question (7.19) is affirmative, this means that if (7.16), (7.18) and (7.19) are true then the sentence "Smith delivered a report an invoice and a project proposal to ITEL" is true. Let h be the sort of humans, c be the sort of companies and i be the sort of information. Let `smith` and `itel` be two constants with, respectively, sort h and c. Let *report, invoice* and *project-proposal* be three predicates with sort $i \rightarrow t$ and *delivered* a predicate with sort $(h \rightarrow (i \rightarrow c)) \rightarrow t$

(7.20) Smith delivered a report to ITEL.
$\exists\, x_1{}^i\, (report\, (x_1) \wedge delivered\, (\texttt{smith}, \texttt{itel}, x_1))$

(7.21) She also delivered them an invoice.
$\exists\, x_2{}^i\, (invoice\, (x_2) \wedge \mathcal{A}\, y_1{}^h\, \mathcal{A}\, y_2{}^c\, delivered\, (y_1, y_2, x_2))$

(7.22) She also delivered them a project proposal.
$\exists\, x_3{}^i\, (project\text{-}proposal\, (x_2) \wedge \mathcal{A}\, y_3{}^h\, \mathcal{A}\, y_4{}^c\, delivered\, (y_3, y_4, x_3))$

$\mathcal{G} =$

$\mathfrak{m}_0 = (!, ((F_1 \wedge F_2) \wedge F_3) \supset F_4)$	
$\mathfrak{m}_1 = (?, ((F_1 \wedge F_2) \wedge F_3))$	$\mathfrak{m}_0$
$\mathfrak{m}_2 = (?, \wedge_1)$	$\mathfrak{m}_1$
$\mathfrak{m}_3 = (!, \exists x_1{}^i\, (re\,(x_1) \wedge de\,(\texttt{sm}, \texttt{it}, x_1)) \wedge \exists x_2{}^i\, (in\,(x_2) \wedge \mathcal{A}\, y_1{}^h\, \mathcal{A}\, y_2{}^c\, de\,(y_1, y_2, x_2)))$	$\mathfrak{m}_2$
$\mathfrak{m}_4 = (?, \wedge_1)$	$\mathfrak{m}_3$
$\mathfrak{m}_5 = (!, \exists x_1{}^i\, (re\,(x_1) \wedge de\,(\texttt{sm}, \texttt{it}, x_1)))$	$\mathfrak{m}_4$
$\mathfrak{m}_6 = (?, \exists)$	$\mathfrak{m}_5$
$\mathfrak{m}_7 = (!, re\,(\texttt{k}_1) \wedge de\,(\texttt{sm}, \texttt{it}(\texttt{k}_1))$	$\mathfrak{m}_6$
$\mathfrak{m}_8 = (?, \wedge_1)$	$\mathfrak{m}_7$
$\mathfrak{m}_9 = (!, re\,(\texttt{k}_1))$	$\mathfrak{m}_8$
$\mathfrak{m}_{10} = (?, \wedge_2)$	$\mathfrak{m}_7$
$\mathfrak{m}_{11} = (!, de\,(\texttt{smith}, \texttt{it}, \texttt{k}_1))$	$\mathfrak{m}_{10}$
$\mathfrak{m}_{12} = (?, \wedge_2)$	$\mathfrak{m}_3$
$\mathfrak{m}_{13} = (!, \exists x_2{}^i\, (in\,(x_2) \wedge \mathcal{A}\, y_1{}^h\, \mathcal{A}\, y_2{}^c\, de\,(y_1, y_2, x_2)))$	$\mathfrak{m}_{12}$
$\mathfrak{m}_{14} = (?, \exists)$	$\mathfrak{m}_{13}$
$\mathfrak{m}_{15} = (!, in\,(\texttt{k}_2) \wedge \mathcal{A}\, y_1{}^h\, \mathcal{A}\, y_2{}^c\, de\,(y_1, y_2, \texttt{k}_2)))$	$\mathfrak{m}_{14}$
$\mathfrak{m}_{16} = (?, \wedge_1)$	$\mathfrak{m}_{15}$
$\mathfrak{m}_{17} = (!, in\,(\texttt{k}_1))$	$\mathfrak{m}_{16}$
$\mathfrak{m}_{18} = (?, \wedge_2)$	$\mathfrak{m}_{15}$
$\mathfrak{m}_{19} = (!, \mathcal{A}\, y_1{}^h\, \mathcal{A}\, y_2{}^c\, de\,(y_1, y_2, \texttt{k}_2))$	$\mathfrak{m}_{18}$
$\mathfrak{m}_{20} = (?, \mathcal{A})$	$\mathfrak{m}_{19}$
$\mathfrak{m}_{21} = (!, \mathcal{A}\, y_2{}^c\, de\,(\texttt{sm}, y_2, \texttt{k}_2))$	$\mathfrak{m}_{20}$
$\mathfrak{m}_{22} = (?, \mathcal{A})$	$\mathfrak{m}_{21}$
$\mathfrak{m}_{23} = (!, de\,(\texttt{sm}, \texttt{it}, \texttt{k}_2))$	$\mathfrak{m}_{22}$
$\mathfrak{m}_{24} = (?, \wedge_2)$	$\mathfrak{m}_1$
$\mathfrak{m}_{25} = (!, \exists x_3{}^i\, (p\text{-}p\,(x_2) \wedge \mathcal{A}\, y_3{}^h\, \mathcal{A}\, y_4{}^c\, de\,(y_3, y_4, x_3)))$	$\mathfrak{m}_{24}$
$\mathfrak{m}_{26} = (?, \exists)$	$\mathfrak{m}_{25}$
$\mathfrak{m}_{27} = (!, p\text{-}p\,(\texttt{k}_2) \wedge \mathcal{A}\, y_3{}^h\, \mathcal{A}\, y_4{}^c\, de\,(y_3, y_4, \texttt{k}_2))$	$\mathfrak{m}_{26}$
$\mathfrak{m}_{28} = (?, \wedge_1)$	$\mathfrak{m}_{27}$
$\mathfrak{m}_{29} = (!, p\text{-}p\,(\texttt{k}_2))$	$\mathfrak{m}_{28}$
$\mathfrak{m}_{30} = (?, \wedge_2)$	$\mathfrak{m}_{29}$
$\mathfrak{m}_{31} = (!, \mathcal{A}\, y_3{}^h\, \mathcal{A}\, y_4{}^c\, de\,(y_3, y_4, \texttt{k}_2))$	$\mathfrak{m}_{30}$
$\mathfrak{m}_{32} = (?, \mathcal{A})$	$\mathfrak{m}_{31}$
$\mathfrak{m}_{33} = (!, \mathcal{A}\, y_4{}^c\, de\,(\texttt{sm}, y_4, \texttt{k}_2))$	$\mathfrak{m}_{32}$
$\mathfrak{m}_{34} = (?, \mathcal{A})$	$\mathfrak{m}_{33}$
$\mathfrak{m}_{35} = (!, de\,(\texttt{sm}, \texttt{it}, \texttt{k}_2))$	$\mathfrak{m}_{34}$
$\mathfrak{m}_{36} = (!, (\exists x_4{}^i\, re\,(x_4) \wedge de(\texttt{sm}, \texttt{it}, x_4)) \wedge (\exists x_5{}^i\, in(x_5) \wedge de(\texttt{sm}, \texttt{it}, x_5))$ $\wedge(\exists x_6{}^i\, p\text{-}p\,(x_6) \wedge de\,(\texttt{sm}, \texttt{it}, x_6)))$	$\mathfrak{m}_1$

Figure 7.3: Winning $\mathcal{A}$-strategy for the third example involving anaphoras. First part

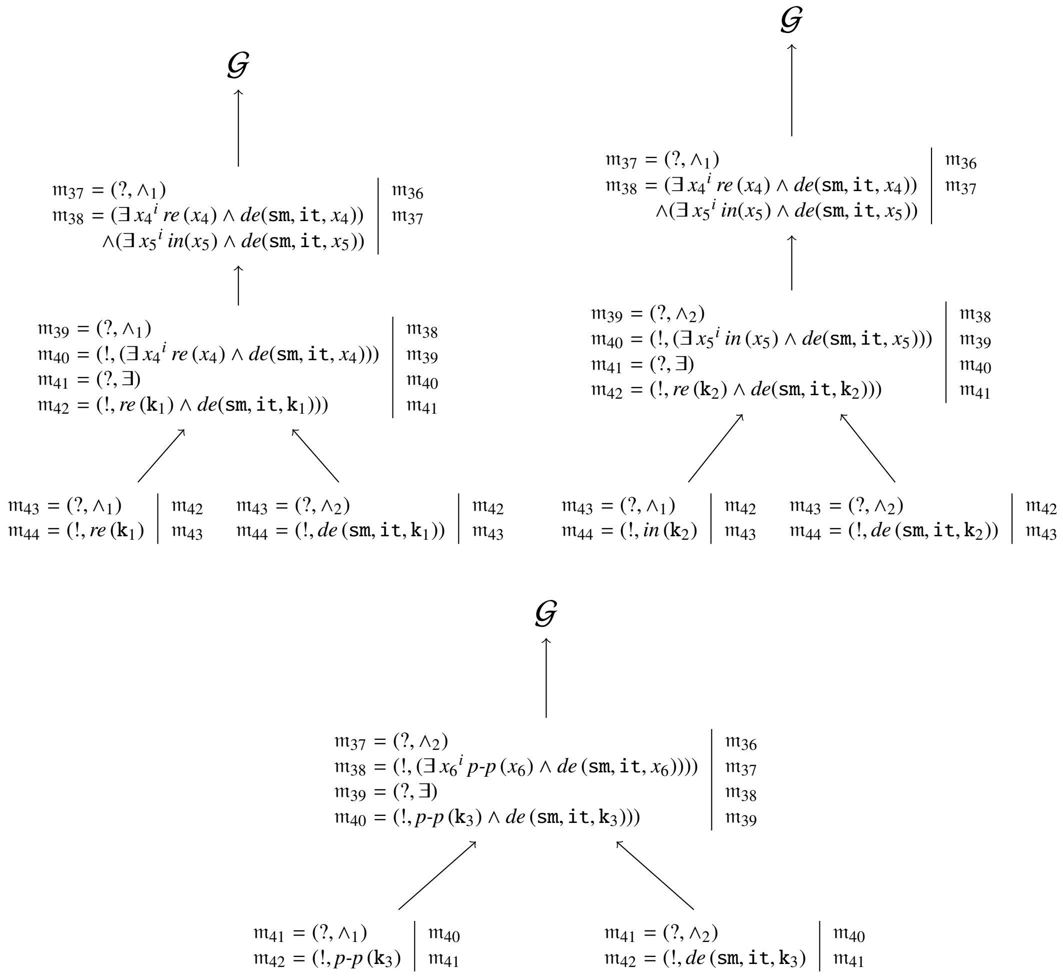


Figure 7.4: Winning $\mathcal{A}$-strategy for the third example involving anaphoras. Second part

(7.23) Smith delivered a report, an invoice and a project proposal to ITEL.
$(\exists x_4{}^i\ \mathit{report}(x_4) \wedge \mathit{delivered}(\mathtt{smith}, \mathtt{itel}, x_4)) \wedge (\exists x_5{}^i\ \mathit{invoice}(x_5) \wedge \mathit{delivered}(\mathtt{smith}, \mathtt{itel}, x_5)) \wedge (\exists x_6{}^i\ \mathit{project\text{-}proposal}(x_6) \wedge \mathit{delivered}(\mathtt{smith}, \mathtt{itel}, x_6))$

Call the formulas in (7.20), (7.21), (7.22) and (7.22) F_1, F_2, F_3 and F_4 respectively. For reasons of space we show a winning $\mathcal{A}$ strategy for the formula $((F_1 \wedge F_2) \wedge F_3) \supset F_4$ in figure 7.3 and figure 7.4. In the figures *re* stands for *report*, *de* for *delivered*, *in* for *invoice*, *p-p* for *project-proposal*, `sm` for `smith` and `it` for `itel`.

7.4.4 Fourth Example Involving Anaphoras: Donkey Anaphora

The following example is not taken from the FraCas test suite. The examples of E-type pronouns and Donkey anaphoras [67] in the FraCas test-suite involve vague quantifiers such as 'several'. However, the example follows the pattern of those from the suite.

(7.24) Every farmer who owns a tractor has a service contract for it.

(7.25) Smith is farmer who owns some tractors.

(7.26) Does Smith has a service contract for all his tractors? [Yes]

The sentence (7.24) is an example of the so-called *donkey sentence*. Donkey sentences are sentences in which a pronoun — the pronoun *it* in our sentence— is semantically bound by a quantifier but lies, syntactically, outside the scope of that quantifier. Barker and Shan [10] defines a Donkey pronoun as

> a pronoun that lies outside the restrictor of a quantifier or the if-clause of a conditional, yet covaries with some quantificational element inside it, usually an indefinite.

A possible translation of 7.24 in multi-sorted first order logic would be

(7.27) $\forall x^h\ [\mathit{farmer}(x) \wedge \exists y^{nh}\ (\mathit{tractor}(y) \wedge \mathit{owns}(x, y)) \supset \mathit{has\text{-}service\text{-}contract}(z^{nh}, x)]$

where h is the sort of humans and nh the sort of non-humans. As the reader can see, the variable z^{nh} lies outside the scope of the quantifier $\exists$, even if it is semantically bound by it. As we have already done, we solve this problem by letting the quantifier $\mathcal{A}$ bind the occurrence of z

$\mathcal{G}_1$

↑

$\mathfrak{m}_{15} = (?, \wedge_2)$	$\mathfrak{m}_{14}$
$\mathfrak{m}_{16} = (?, \wedge_2)$	$\mathfrak{m}_{1}$
$\mathfrak{m}_{17} = (!, \exists x_3{}^{nh}\, ((tr\,(x_3) \wedge ow\,(\mathtt{sm}, x_3)) \wedge fa\,(\mathtt{sm}))$	$\mathfrak{m}_{16}$
$\mathfrak{m}_{18} = (?, \wedge_1)$	$\mathfrak{m}_{17}$
$\mathfrak{m}_{19} = (!, \exists x_3{}^{nh}\, ((tr\,(x_3) \wedge ow\,(\mathtt{sm}, x_3)))$	$\mathfrak{m}_{18}$
$\mathfrak{m}_{20} = (?, \exists)$	$\mathfrak{m}_{15}$
$\mathfrak{m}_{21} = (!, (tr\,(\mathtt{k}_2) \wedge ow\,(\mathtt{sm}, k_2)))$	$\mathfrak{m}_{20}$
$\mathfrak{m}_{22} = (?, \wedge_1)$	$\mathfrak{m}_{21}$
$\mathfrak{m}_{23} = (!, (tr\,(\mathtt{k}_2)))$	$\mathfrak{m}_{22}$
$\mathfrak{m}_{24} = (?, \wedge)$	$\mathfrak{m}_{21}$
$\mathfrak{m}_{25} = (!, ow\,(\mathtt{sm}, k_2))$	$\mathfrak{m}_{24}$
$\mathfrak{m}_{26} = (!, \exists x_2{}^{nh}\, (tr\,(x_2) \wedge ow\,(\mathtt{sm}, x_2)))$	$\mathfrak{m}_{15}$
$\mathfrak{m}_{27} = (?, \exists)$	$\mathfrak{m}_{26}$
$\mathfrak{m}_{28} = (!, (tr\,(\mathtt{k}_2) \wedge ow\,(\mathtt{sm}, k_2)))$	$\mathfrak{m}_{27}$

$\mathfrak{m}_{29} = (?, \wedge_1)$	$\mathfrak{m}_{28}$
$\mathfrak{m}_{30} = (!, tr(\mathtt{k}_2))$	$\mathfrak{m}_{29}$

$\mathfrak{m}_{29} = (?, \wedge_2)$	$\mathfrak{m}_{28}$
$\mathfrak{m}_{30} = (!, ow\,(\mathtt{sm}, \mathtt{k}_2))$	$\mathfrak{m}_{29}$

$\mathcal{G}_1$

↗ ↖

$\mathfrak{m}_{15} = (?, \wedge_1)$	$\mathfrak{m}_{14}$
$\mathfrak{m}_{16} = (?, \wedge_2)$	$\mathfrak{m}_{1}$
$\mathfrak{m}_{17} = (!, \exists x_3{}^{nh}\, ((tr\,(x_3) \wedge ow\,(\mathtt{sm}, x_3)) \wedge fa\,(\mathtt{sm}))$	$\mathfrak{m}_{16}$
$\mathfrak{m}_{18} = (?, \wedge_2)$	$\mathfrak{m}_{17}$
$\mathfrak{m}_{19} = (!, fa\,(\mathtt{sm}))$	$\mathfrak{m}_{18}$
$\mathfrak{m}_{20} = (!, fa\,(\mathtt{sm}))$	$\mathfrak{m}_{15}$

$\mathfrak{m}_{15} = (!, \mathcal{A}\, y_1{}^{nh}\, h\text{-}a\text{-}c\,(\mathtt{sm}, y_1))$	$\mathfrak{m}_{14}$
$\mathfrak{m}_{16} = (?, \mathcal{A})$	$\mathfrak{m}_{15}$
$\mathfrak{m}_{17} = (!, h\text{-}a\text{-}c\,(\mathtt{sm}, \mathtt{k}_1))$	$\mathfrak{m}_{16}$
$\mathfrak{m}_{18} = (!, h\text{-}a\text{-}c\,(\mathtt{sm}, \mathtt{k}_1))$	$\mathfrak{m}_{5}$

Figure 7.5: Winning $\mathcal{A}$-strategy for the fourth example involving anaphoras.

(7.28) Every farmer who owns a tractor has a service contract for it.

$$\forall x_1{}^{h}\, [\mathit{farmer}\,(x_1) \wedge \exists\, x_2{}^{nh}\, (\mathit{tractor}\,(x_2) \wedge \mathit{owns}\,(x_1, x_2)) \supset \mathcal{A}\, y_1{}^{nh}\, \mathit{has\text{-}service\text{-}contract}\,(y_1, x_1)]$$

(7.29) Smith is a farmer who owns some tractors.

$$\exists\, x_3{}^{nh}\, ((\mathit{tractor}\,(x_3) \wedge \mathit{owns}\,(\mathtt{smith}, x_3)) \wedge \mathit{farmer}\,(\mathtt{smith})$$

(7.30) Smith has a service contract for all his tractors

$$\forall\, x_4{}^{nh}\, (\mathit{tractor}\,(x_4) \wedge \mathit{owns}\,(\mathtt{smith}, x_4) \supset \mathit{has\text{-}service\text{-}contract}\,(\mathtt{smith}, x_4))$$

Let $\mathcal{G}_1$ be the following game

$$\mathcal{G}_1 = \begin{array}{l|l}
\mathfrak{m}_0 = (!, F_1 \wedge F_2 \supset F_3) & \\
\mathfrak{m}_1 = (?, F_1 \wedge F_2) & \mathfrak{m}_0 \\
\mathfrak{m}_2 = (!, \forall\, x_4{}^{nh}\, (\mathit{tr}\,(x_4) \wedge \mathit{ow}\,(\mathtt{sm}, x_4) \supset \mathit{h\text{-}s\text{-}c}\,(\mathtt{sm}, x_4))) & \mathfrak{m}_2 \\
\mathfrak{m}_3 = (?, \forall[\mathtt{k}_1/x_4]) & \mathfrak{m}_2 \\
\mathfrak{m}_4 = (!, \mathit{tr}\,(\mathtt{k}_1) \wedge \mathit{ow}\,(\mathtt{sm}, \mathtt{k}_1) \supset \mathit{h\text{-}s\text{-}c}\,(\mathtt{sm}, \mathtt{k}_1))) & \mathfrak{m}_3 \\
\mathfrak{m}_5 = (?, \mathit{tr}\,(\mathtt{k}_1) \wedge \mathit{ow}\,(\mathtt{sm}, \mathtt{k}_1)) & \mathfrak{m}_4 \\
\mathfrak{m}_6 = (?, \wedge_1) & \mathfrak{m}_5 \\
\mathfrak{m}_7 = (!, \mathit{tr}\,(\mathtt{k}_1)) & \mathfrak{m}_6 \\
\mathfrak{m}_8 = (?, \wedge_2) & \mathfrak{m}_5 \\
\mathfrak{m}_9 = (!, \mathit{ow}\,(\mathtt{sm}, \mathtt{k}_1)) & \mathfrak{m}_8 \\
\mathfrak{m}_{10} = (?, \wedge_1) & \mathfrak{m}_1 \\
\mathfrak{m}_{11} = (!, \forall x_1{}^{h}\, [\mathit{fa}\,(x_1) \wedge \exists\, x_2{}^{nh}\, (\mathit{tr}\,(x_2) \wedge \mathit{ow}\,(x_1, x_2)) \supset \mathcal{A}\, y_1{}^{nh}\, \mathit{h\text{-}s\text{-}c}\,(y_1, x_1)]) & \mathfrak{m}_{10} \\
\mathfrak{m}_{12} = (?, \forall[\mathtt{sm}/x_1]) & \mathfrak{m}_{11} \\
\mathfrak{m}_{13} = (!, \mathit{fa}\,(\mathtt{sm}) \wedge \exists\, x_2{}^{nh}\, (\mathit{tr}\,(x_2) \wedge \mathit{ow}\,(\mathtt{sm}, x_2)) \supset \mathcal{A}\, y_1{}^{nh}\, \mathit{h\text{-}s\text{-}c}\,(\mathtt{sm}, y_1)) & \mathfrak{m}_{12} \\
\mathfrak{m}_{14} = (?, \mathit{fa}\,(\mathtt{sm}) \wedge \exists\, x_2{}^{nh}\, (\mathit{tr}\,(x_2) \wedge \mathit{ow}\,(\mathtt{sm}, x_2))) & \mathfrak{m}_{13}
\end{array}$$

A strategy for the formula $F_1 \wedge F_2 \supset F_3$ is shown in figure 7.5

7.5 Ellipsis

We consider the problem of resolving Post-Auxilliary Ellipsis (PAE) [65], more commonly referred to by the term verb-phrase ellipsis (VPE), using the technology used to resolve pronouns.[2] The following discourse exemplifies the phenomenon:

(7.31) John slept.

[2]VPE doesn't describe the whole distribution of PAE, since, unless one extends the notion of VP well beyond its descriptive use, the antecedents for PAE need not be VPs. See [74] for excellent descriptive discussion of the distribution of ellipsis in English.

(7.32) Bill did not.

If the only context for the second sentence is the first sentence, one would correctly infer that what Bill didn't do is sleep. The terseness of the antecedent might give the impression that ellipsis resolution is resolution to some preceding property–here the property of sleeping. This picture is complicated by the possibility of resolving the ellipsis to a modified property, without thereby including the modifier in the resolution. The following discourse from [28] exemplifies the phenomenon:

(7.33) John spoke to Mary at four o'clock.

(7.34) And Bill did at five o'clock.

PAE is a highly studied topic in both theoretical syntax and linguistic semantics [102]. Linguistic theories are often split by whether they presume there is hidden syntactic structure in the ellipse, i.e. whether 'and Bill did at five o'clock' is underlyingly 'and Bill spoke to Mary at five o'clock', and whether the resolution of the antecedent is in terms of (more or less) syntactic or semantic representation. Among the theories proposed, few of them enjoy the rigor of a logic [56, 138, 14, 33] and fewer directly employ the proof theory of the logic [78]. This list is not exhaustive, but it suffices to show the privilege of denotation over deduction in the use of logic for linguistic semantics.

We contend that ellipsis can be resolved by means of deduction in the logic we use to resolve pronouns. We propose to use the notion of an event, common to philosophy since [35] but widely employed in subsequent linguistic theory [133]. Events are denoted by verbs, and therefore provide objects which can be subsequently referred to.

7.5.1 Events

We propose every verb introduces an event, which event may subsequently be referred to. While events are put to a wide variety of uses in philosophy and linguistics, they remain a somewhat vexed notion.[3] Our use of events is modest but deserves some discussion. Events are a sort in the underlying first order language we reason over.

To be more precise about our conception of events, consider the sort of discourse which [35] used to motivate events:

[3] See [133] for an overview of some of the uses of events in linguistics and philosophy. [140] divided events into various subtypes: achievements, activities, accomplishments, and states. We will not be concerned with subtyping events in the present paper.

(7.35) Jones buttered the toast. He did it slowly in the bathroom at midnight.

For Davidson, Jones buttering his toast is an event, which event is referred to by ‘it’ in the subsequent sentence. The verb in the first sentence is conjectured to introduces some object —the event— for further description. This suggests an event is something with time and extension.[4] Now consider the following discourse:

(7.36) John reviewed the paper on Tuesday and Bill did (it) on Wednesday.

It is likely that what Bill did Wednesday is review the paper John reviewed the previous day. But obviously Bill’s reviewing the paper doesn’t require his reviewing to be in or at the time or extension of John’s reviewing the paper. Consequently what ‘did (it)’ refers to need not include the time or extension in which the antecedent of ‘did (it)’ happened. But then, if events are construed in terms of some time and extension, ‘did (it)’ could not refer to reviewing the paper *per se*. This predicament can be resolved without wrecking our folk conception of events or their utility in theories of anaphoric reference.

We propose that ‘did it’ retrieves the event kind of reviewing. There is precedent for the notion of event kind in linguistic theory [57], though we commit to no pre-existing theory of whether or how such kinds could be (re)constructed from the Davidsonian notion of event. Subsequently, we will use the term ‘event kind’ to refer to our theoretic conception of events and the term ‘event proper’ to refer to the tokening of an event kind with respect to some time, extension, and participant(s). When no confusion is possible, we will simply use the term ‘event’. We will subsequently use the Neo-Davidsonian idiom of ‘thematic roles’ when discussing the participants of an event token, which will be modelled in terms of properties of triples of entities, times, and event kinds. We argue that referring back to an event kind needn’t involve referring to those involved in the event proper or the time of the event proper–under our theory these further references correspond to further anaphoras. To see how the time and extension of an event proper are resolved independently of event kinds consider the following:

(7.37) John reviewed the paper on Tuesday at his home. Bill did (it) the same day and at John’s house too.

[4] While ‘do it’ is not uniformly considered anaphoric to some event [27], this view enjoys wide support among both linguists and philosophers. The arguments [27] levies against ‘do it’ referring to some event deserve considered discussion, but space restrictions prevent us from providing such discussion here. Davidson’s argument for events is first and foremost that they provide a solution to the problem of underspecified adicity [82]. [69, 70] argues they provide the means to model some phenomena that otherwise would require higher-order terms. Neither of these points are, we think, rebutted by [27].

The expression ‘same’ is anaphoric to the time of the preceding event proper, while the additive expression ‘too’ is anaphoric to the extension of the previous event proper.[5] Without these explicit devices for referring to parts of the preceding event proper, it would be unjustified to conclude the event kind corresponding to Bill reviewing the paper includes reference to the time or extension of John reviewing the paper.

Given the foregoing, we think our notion of event kind is of merit. Event anaphora is event kind anaphora, while the event proper is determined by specifying the participant(s), time, and extension of the event kind. The object of the reviewing event Bill is involved in will be rolled into a condition on the term corresponding to PAE; we discuss the resolution of objects in the sequel. While we are here concerned with ellipsis, our account could be extended to ‘do it’ and ‘do so’ too.

7.6 Textual Entailment Recognition and Ellipsis Resolution

We present some examples of textual entailment recognition involving ellipsis revolution.

7.6.1 First Example involving Ellipsis: VP-ellipsis

(7.38) John spoke to Mary at four o’clock.

(7.39) Bill did it at five o’clock.

(7.40) Did Bill speak to Mary at five o’clock? [Yes]

The answer to (7.40) is affirmative, this means that if the two sentences (7.38) and (7.39) are true so the sentence “Bill speak to Mary at five o’clock” must be true. As previously specified, we use a neodavidsonian account of event and thus translate the sentences as follows: let mh be the sorts of male humans, fh the sort of female humans, v be the sort of events and n the sort of time moments. Let `john` and `bill` be constants of sort mh, `mary` be a constant of sort fh and `4` and `5` be constants with sort n. The sorts of the predicates can be inferred by the sorts of their arguments.

[5] We are not here concerned with the distinction between anaphora and presupposition. The content of ‘too’ is sometimes said to be a presupposition, while ‘same’ is quite uniformly considered anaphoric. Since [137], presuppositions are often considered a variety of anaphora.

(7.41) John spoke to Mary at four o'clock.
$\exists x_1^v\,(((spoke(x_1) \wedge agent\,(x_1, \texttt{john}, 4)) \wedge (pa(x_1, \texttt{mary}, 4))$

$\mathfrak{m}_0 = (!, [\exists x_1^v\,(((sp(x_1) \wedge ag\,(x_1, \texttt{jo}, 4)) \wedge (pa(x_1, \texttt{ma}, 4)) \wedge \mathcal{A}y_1{}^v\, \mathcal{A}y_2{}^{fh}(ag(y_1, \texttt{bi}, 5) \wedge pa(y_1, y_2, 5))]$ $\supset \exists x_2^v\,((sp\,(x_2) \wedge ag\,(x_2, \texttt{bi}, 5) \wedge pa\,(x_2, \texttt{ma}, 5))$	
$\mathfrak{m}_1 = (?, \exists x_1^v\,(((sp(x_1) \wedge ag\,(x_1, \texttt{jo}, 4)) \wedge (pa(x_1, \texttt{ma}, 4)))) \wedge \mathcal{A}y_1{}^v\, \mathcal{A}y_2{}^{fh}(ag(y_1, \texttt{bi}, 5) \wedge pa(y_1, y_2, 5)))$	$\mathfrak{m}_0$
$\mathfrak{m}_2 = (?, \wedge_1)$	$\mathfrak{m}_1$
$\mathfrak{m}_3 = (!, \exists x_1^v\,(((sp(x_1) \wedge ag\,(x_1, \texttt{jo}, 4)) \wedge (pa(x_1, \texttt{ma}, 4)))))$	$\mathfrak{m}_2$
$\mathfrak{m}_4 = (?, \exists)$	$\mathfrak{m}_3$
$\mathfrak{m}_5 = (!, (((sp(\texttt{k}_1) \wedge ag\,(\texttt{k}_1, \texttt{jo}, 4)) \wedge (pa(\texttt{k}_1, \texttt{ma}, 4)))))$	$\mathfrak{m}_4$
$\mathfrak{m}_6 = (?, \wedge_1)$	$\mathfrak{m}_5$
$\mathfrak{m}_7 = (!, sp(\texttt{k}_1) \wedge ag\,(\texttt{k}_1, \texttt{jo}, 4))$	$\mathfrak{m}_6$
$\mathfrak{m}_8 = (?, \wedge_1)$	$\mathfrak{m}_7$
$\mathfrak{m}_9 = (!, sp(\texttt{k}_1))$	$\mathfrak{m}_8$
$\mathfrak{m}_{10} = (?, \wedge_2)$	$\mathfrak{m}_7$
$\mathfrak{m}_{11} = (!, ag\,(\texttt{k}_1, \texttt{jo}, 4))$	$\mathfrak{m}_{10}$
$\mathfrak{m}_{12} = (?, \wedge_2)$	$\mathfrak{m}_5$
$\mathfrak{m}_{13} = (!, pa\,(\texttt{k}_1, \texttt{ma}, 4))$	$\mathfrak{m}_{12}$
$\mathfrak{m}_{14} = (?, \wedge_2)$	$\mathfrak{m}_1$
$\mathfrak{m}_{15} = (!, \mathcal{A}y_1{}^v\, \mathcal{A}y_2{}^{fh}(ag(\texttt{k}_1, \texttt{bi}, 5) \wedge pa(y_1, y_2, 5)))$	$\mathfrak{m}_{14}$
$\mathfrak{m}_{16} = (?, \mathcal{A})$	$\mathfrak{m}_{15}$
$\mathfrak{m}_{17} = (!, \mathcal{A}y_2{}^{fh}\,(ag\,(\texttt{k}_1, \texttt{bi}, 5) \wedge pa\,(\texttt{k}_1, y_2, 5)))$	$\mathfrak{m}_{16}$
$\mathfrak{m}_{18} = (?, \mathcal{A})$	$\mathfrak{m}_{17}$
$\mathfrak{m}_{19} = (!, ag\,(\texttt{k}_1, \texttt{bi}, 5) \wedge pa\,(\texttt{k}_1, \texttt{ma}, 5))$	$\mathfrak{m}_{18}$
$\mathfrak{m}_{20} = (?, \wedge_1)$	$\mathfrak{m}_{19}$
$\mathfrak{m}_{21} = (!, ag\,(\texttt{k}_1, \texttt{bi}, 5))$	$\mathfrak{m}_{20}$
$\mathfrak{m}_{22} = (?, \wedge_2)$	$\mathfrak{m}_{19}$
$\mathfrak{m}_{23} = (!, pa\,(\texttt{k}_1, \texttt{ma}, 5))$	$\mathfrak{m}_{22}$
$\mathfrak{m}_{24} = (!, \exists x_2^v\,((sp\,(x_2) \wedge ag\,(x_2, \texttt{bi}, 5) \wedge pa\,(x_2, \texttt{ma}, 5))))$	$\mathfrak{m}_1$
$\mathfrak{m}_{25} = (?, \exists)$	$\mathfrak{m}_{24}$
$\mathfrak{m}_{26} = (!, ((sp\,(\texttt{k}_1) \wedge ag\,(\texttt{k}_1, \texttt{bi}, 5) \wedge pa\,(\texttt{k}_1, \texttt{ma}, 5))$	

$\mathfrak{m}_{27} = (?, \wedge_1)$	$\mathfrak{m}_{26}$
$\mathfrak{m}_{28} = (!, sp\,(\texttt{k}_1) \wedge ag\,(\texttt{k}_1, \texttt{bi}, 5))$	$\mathfrak{m}_{27}$

$\mathfrak{m}_{27} = (?, \wedge_2)$	$\mathfrak{m}_{26}$
$\mathfrak{m}_{28} = (!, pa\,(\texttt{k}_1, \texttt{ma}, 5)$	$\mathfrak{m}_{27}$

$\mathfrak{m}_{29} = (\wedge_1)$	$\mathfrak{m}_{28}$
$\mathfrak{m}_{30} = (!, sp\,(\texttt{k}_1))$	$\mathfrak{m}_{29}$

$\mathfrak{m}_{29} = (?, \wedge_2)$	$\mathfrak{m}_{28}$
$\mathfrak{m}_{30} = (!, ag\,(\texttt{k}_1, \texttt{bi}, 5))$	$\mathfrak{m}_{29}$

Figure 7.6: Winning $\mathcal{A}$-strategy for the first example involving ellipsis resolution

(7.42) Bill did it at five o'clock.
$\mathcal{A}y_1{}^v\,\mathcal{A}y_2{}^{fh}(agent(y_1, \texttt{bill}, 5) \wedge patient(y_1, y_2, 5))$

(7.43) Bill spoke to Mary at five o'clock.
$\exists x_2^v\,((spoke\,(x_2) \wedge agent\,(x_2, \texttt{bill}, 5)) \wedge patient\,(x_2, \texttt{mary}, 5))$

We consider that the thematic role *agent* and *patient* conveys temporal information about the event. We also consider that the patient of *did it* is anaphoric. As usual, we call the formulas in (7.41), (7.42) and (7.43) F_1, F_2 and F_3 respectively. Figure 7.6 shows a winning $\mathcal{A}$-strategy for the formula $F_1 \wedge F_2 \supset F_3$.

7.6.2 Second Example Involving Ellipsis

(7.44) John slept.

(7.45) Bill did not.

(7.46) Does Bill slept? [No]

The answer to (7.46) is negative. This means that if (7.44) and (7.45) are true, then the sentence "Bill did not sleep" must be true. If we use a neodavidsonian account of meaning this latter sentence is ambiguous. It could either mean that "there is no event x such that x is a sleeping event and Bill is the agent of x" or "there is an event x, x is a sleeping event and Bill is not the agent of x. The two possible reading of 'Bill did not sleep' are shown in (7.49).

(7.47) John slept.
$F_1 = \exists x_1^v\,(sleep\,(x_1) \wedge agent\,(x_1, \texttt{john}))$

(7.48) Bill did not.
$F_2 = \mathcal{A}y_1^v\,\neg agent\,(y_1, \texttt{bill})$

(7.49) Bill did not slept.
$H = \neg(\exists x_2^v\,(sleep\,(x_1) \wedge agent\,(x_2, \texttt{bill})))$
$G = \exists x_2{}^v\,(sleep\,(x_2) \wedge \neg agent\,(x_2, \texttt{bill}))$

$\mathfrak{m}_0 = (!, [\exists x_1^v\,(sl\,(x_1) \land ag\,(x_1, \mathtt{jo})) \land \mathcal{A} y_1^v\, \neg ag\,(y_1, \mathtt{bi})] \supset \neg(\exists x_2^v\,(sl\,(x_2) \land ag\,(x_2, \mathtt{bi}))))$	
$\mathfrak{m}_1 = (?, \exists x_1^v\,(sl\,(x_1) \land ag\,(x_1, \mathtt{jo})) \land \mathcal{A} y_1^v\, \neg ag\,(y_1, \mathtt{bi}))$	$\mathfrak{m}_0$
$\mathfrak{m}_2 = (!, \neg(\exists x_2^v\,(sl\,(x_1) \land ag\,(x_2, \mathtt{bi}))))$	$\mathfrak{m}_1$
$\mathfrak{m}_3 = (?, (\exists x_2^v\,(sl\,(x_1) \land ag\,(x_2, \mathtt{bi}))))$	$\mathfrak{m}_2$
$\mathfrak{m}_4 = (?, \exists)$	$\mathfrak{m}_3$
$\mathfrak{m}_5 = (!, (sl\,(\mathtt{k}_1) \land ag\,(\mathtt{k}_1, \mathtt{bi})))$	$\mathfrak{m}_4$
$\mathfrak{m}_6 = (?, \land_2)$	$\mathfrak{m}_5$
$\mathfrak{m}_7 = (!, ag\,(\mathtt{k}_1, \mathtt{bi}))$	$\mathfrak{m}_6$
$\mathfrak{m}_8 = (?, \land_2)$	$\mathfrak{m}_1$
$\mathfrak{m}_9 = (!, \mathcal{A} y_1^v\, \neg ag\,(y_1, \mathtt{bi}))$	$\mathfrak{m}_8$
$\mathfrak{m}_{10} = (?, \mathcal{A})$	$\mathfrak{m}_9$
$\mathfrak{m}_{11} = (!, \neg ag\,(\mathtt{k}_1, \mathtt{bi}))$	$\mathfrak{m}_{10}$
$\mathfrak{m}_{12} = (?, ag\,(\mathtt{k}_1, \mathtt{bi}))$	$\mathfrak{m}_{11}$
$\mathfrak{m}_{13} = (!, \bot)$	$\mathfrak{m}_{12}$

Figure 7.7: Winning $\mathcal{A}$-strategy for the first version of the second example involving ellipsis resolution

Figure 7.7 shows a winning $\mathcal{A}$-strategy for the formula $F_1 \land F_2 \supset H$, while figure 7.8 shows a winning $\mathcal{A}$-strategy for the formula $F_1 \land F_2 \supset G$. In the figure *sl*, *ag* and `bi` stands for, respectively, *sleep*, *agent* and `bill`. Bill has sort h (the sort of humans) while v is the sort of events. The predicate's sorts can be inferred by the one of their arguments.

7.7 Discussion

7.7.1 Other Works

The most obvious theories to compare the present work to are Discourse Representation Theory (DRT) [80, 81], Dynamic Predicate Logic (DPL) [64], Dekker's Predicate Logic with Anaphora [37] and Predicate Logic with Indices [38]. We briefly discuss these theories in turn.

DRT is concerned with the construction of Discourse Representation Structures (DRS), which serve to track objects invoked by speakers in the course of producing some discourse. It includes rules for introducing, blocking, and linking (sub)DRSs to one another. The semantics of the DRS language, in its standard version, is given by an embedding function

$\mathfrak{m}_0 = (!, [\exists x_1^v\,(sl\,(x_1) \land ag\,(x_1, \texttt{jo})) \land \mathcal{A} y_1^v\,\neg ag\,(y_1, \texttt{bi})] \supset \exists x_2{}^v\,(sl\,(x_2) \land \neg ag\,(x_2, \texttt{bi})))$	
$\mathfrak{m}_1 = (?, \exists x_1^v\,(sl\,(x_1) \land ag\,(x_1, \texttt{jo})) \land \mathcal{A} y_1^v\,\neg ag\,(y_1, \texttt{bi}))$	$\mathfrak{m}_0$
$\mathfrak{m}_2 = (?, \land_1)$	$\mathfrak{m}_1$
$\mathfrak{m}_3 = (!, \exists x_1^v\,(sl\,(x_1) \land ag\,(x_1, \texttt{jo})))$	$\mathfrak{m}_2$
$\mathfrak{m}_4 = (?, \exists)$	$\mathfrak{m}_3$
$\mathfrak{m}_5 = (!, (sl\,(\texttt{k}_1) \land ag\,(\texttt{k}_1, \texttt{jo})))$	$\mathfrak{m}_4$
$\mathfrak{m}_6 = (?, \land_2)$	$\mathfrak{m}_1$
$\mathfrak{m}_7 = (!, \mathcal{A} y_1^v\,\neg ag\,(y_1, \texttt{bi}))$	$\mathfrak{m}_6$
$\mathfrak{m}_8 = (?, \mathcal{A})$	$\mathfrak{m}_7$
$\mathfrak{m}_9 = (!, \neg ag\,(\texttt{k}_1, \texttt{bi}))$	$\mathfrak{m}_8$
$\mathfrak{m}_{10} = (!, \exists x_2{}^v\,(sl\,(x_2) \land \neg ag\,(x_2, \texttt{bi})))$	$\mathfrak{m}_1$
$\mathfrak{m}_{11} = (?, \exists)$	$\mathfrak{m}_{10}$
$\mathfrak{m}_{12} = (!, sl\,(\texttt{k}_1) \land \neg ag\,(\texttt{k}_1, \texttt{bi}))$	$\mathfrak{m}_{11}$

$\mathfrak{m}_{13} = (?, \land_1)$	$\mathfrak{m}_{12}$
$\mathfrak{m}_{11} = (!, sl\,(\texttt{k}_1))$	$\mathfrak{m}_{13}$

$\mathfrak{m}_{13} = (?, \land_2)$	$\mathfrak{m}_{12}$
$\mathfrak{m}_{14} = (!, \neg ag\,(\texttt{k}_1, \texttt{bi}))$	$\mathfrak{m}_{13}$
$\mathfrak{m}_{15} = (?, ag\,(\texttt{k}_1, \texttt{bi}))$	$\mathfrak{m}_{14}$
$\mathfrak{m}_{16} = (?, ag\,(\texttt{k}_1, \texttt{bi}))$	$\mathfrak{m}_9$
$\mathfrak{m}_{17} = (!, \bot)$	$\mathfrak{m}_{16}$

Figure 7.8: Winning $\mathcal{A}$-strategy for the second version of the second example involving ellipsis resolution

into first order logic (FOL for short). Some attempts have been made to do inference directly over DRSs; [84] provides a brief but non-exhaustive survey of existing inference systems for DRT and DPL circa 2000. Subsequently, he proposes a tableaux theorem proving method for DRT that includes a connective quite close to our $\mathcal{A}$ connective.

DPL, unlike DRT, just is first order logic. However, it effectively redefines the semantics of FOL in order to extend the scope of ∃. It is, in principle, a reconstruction of DRT. While much work develops DPL's novel semantics for FOL [141], the proof theory of the logic receives less study. DPL's proof theory only fully developed some years after its discovery. The most successful study of DPL's proof theory is [139]. However, DPL relies on the linguist to select the correct variable for a pronoun to be resolved by its antecedent and is therefore modelling anaphoric dependence but not resolution.

Dekker develops two logics for anaphora, intending to show the minimum needed to modify FOL to account for pronouns. Dekker's approach is more proof-theoretic in adding pronouns to the inventory of terms, which, in [38], are de Bruijn (pre-)indexed identity functions on (sequences of) the ith antecedent. Such terms are subject to inference rules. Nonetheless, the preindexing corresponds to the choice of variable made by the linguist in choosing the antecedent of a pronoun in DPL.

The price of modelling resolution in our logic is the multiplicity of possible antecedents for an anaphor. If there is a disjunction of possible antecedents, the selection of just one could be enforced by means of further axioms. In fact, the winnowing down of possible antecedents, is present in common symbolic approaches to anaphora resolution. We are presently considering how to implement Centering Theory [18] and/or Coherence Theory [72] in our logic, which we intend to present in future work.

7.7.2 Sorting

The mechanism that we invented to treat anaphora and ellipsis deductively heavily depends on sorting. Consider the sentence

(7.50) Someone entered a room. He whistled.

here the pronoun *he* reefers to *Someone*. Any competent English speaker will recognize this fact. Let o be the sort of inanimate object and h the sort of humans. A possible translation in first order multi-sorted logic of the above sentence may be:

(7.51) $F = (\exists\, x_1^o\, [room\,(x_1) \wedge \exists x_2^h entered\,(x_2, x_1)]) \wedge (\mathcal{A}\, y_1^h\; whistled\,(y_1))$

from this latter formula we can deduce the sentence "someone entered a room and whistled" i.e., there is a winning $\mathcal{A}$-strategy for the formula

$$F \supset \exists\, x_3^o[\, room\,(x_3) \wedge \exists\, x_4^h(entered\,(x_4, x_3) \wedge whistled\,(x_4))]$$

the fact that we can construct a winning $\mathcal{A}$-strategy for the above formula depends upon the following fact. We have attributed the same sort to `paul` and the variable bound by $\mathcal{A}$ and another different sort to the variable bound by $\exists$. If we formalize (7.50) in plain first order logic e.g.,

(7.52) $G = (\exists\, x_1\, [room\,(x_1) \wedge \exists x_2\; entered\,(x_2, x_1)]) \wedge (\mathcal{A}\, y_1\; whistled\,(y_1))$

then we are not able to construct a winning $\mathcal{A}$-strategy for

$$G \supset \exists x_3[\, room\,(x_3) \wedge \exists x_4(entered\,(x_4, x_3) \wedge whistled\,(x_4))]$$

this is because we will have the variable bound by the quantifier $\mathcal{A}$ will have two 'possible antecedents': the instantiation of the variable x_1 bound by the leftmost existential quantifier of G and the instantiation of the variable x_2 bound by the other existential quantifier of G. However, the use of sorting is a fairly natural assumption, at least as far as the formalization of fragments of the English language is concerned. In a sentence like

(7.53) A man took a train to Baltimore. it whistled at twelve o'clock.

we know that the pronoun *it* refers to *a train*. First the pronoun cannot reefer to *a man*, since *it* is a neutral pronoun that —normally— reefers to impersonal physical objects, abstract concepts, situations, actions, characteristics, etc, but not humans. Furthermore, we know that whatever the reference of *it* may be, this reference must be something that *whistles*. A city, like Baltimore, cannot whistle i.e., they are not the kind, or *sort*, of entities that can be in the extension of the predicate *whistles*. Similar considerations stands as basis of the work of Retore [125] on —compositional— lexical semantics by means of the polymorphic lambda calculus. In turn, Retore's work is inspired by Pustejovsky's linguistic theory of the generative lexicon [119].

7.8 Conclusion

This chapter proposes a novel proof-theoretic method of resolving anaphora in first order multi-sorted logic. This method is the introduction of a new quantifier. The semantics of this quantifier is given by the Dialogical Rules that specifies what counts as an attack against the assertion of a formula $\mathcal{A}xA$ and what count as a defense against such an attack. Such rules prescribe that a formula $\mathcal{A}xA$ can only be instantiated in a game by a term t whose already appears in the game i.e., the quantifier $\mathcal{A}$ varies on individuals whose existence has been asserted in the course of the game.

Despite our preference for an inferentialist theory of meaning, we believe that it is necessary to develop a veroconditional semantics for our logic. That is, we would like to understand what is the class of first-order structures that characterize the logic we have introduced.

Although we believe that the dialogical system we created has an inherent logical interest, we cannot deny that it is its application to natural language that makes it potentially attractive. We would therefore like to better understand what the limitation of our system are. Understanding its limitations could help us improve it significantly. Unfortunately, such work requires the analysis of many cases of anaphora and ellipsis resolutions. It would therefore be useful to carry out a comparative study of our system with more semantically oriented theories of anaphora, such as DRT and work in the dynamic tradition. The language of DRT, DPL and of our system is first order logic. So it should be possible to give an overview of predicted accessibility relations and to compare results.

Part IV

Conclusion

Chapter 8

Conclusion

We conclude our thesis by trying to sketch possible future developments of the research reported herein. We divide them into possible developments in the field of proof theory and applications of proof theory to natural language semantics and syntax. We begin with the former.

8.1 Proof theory

8.1.1 Dialogical Logic

Despite the fact that dialogical logic first developed more than 70 years go, dialogical logic is somewhat neglected in proof theory. Its ‘cousins’ i.e., sequent calculus systems and tableaux systems, are much more popular in the proof theory community. Perhaps this is due to the seeming obscurity of work on the topic, which is not renowned for its mathematical elegance. This view was already stated by Felscher in 1985 [46] and we must write with regret that the situation has not evolved much. Although the literature on dialogical logic contains many interesting ideas, we found that articles on this topic often lack proofs of the main results or precise definitions of the objects being manipulated. By means of example in [121] the author develop a dialogical logic system for the propositional fragment of linear logic. The ideas developed in the article are attractive and simple: by attacking the connectives of linear logic a player open a context that can be considered as a sub-game. The nature of these sub-games depends on the logical connective that has been attacked: sub-games for additive connectives can share some information with other sub-games, while those for multiplicative connectives cannot. This idea fits well with the intuitive interpretation of linear logical connectives. According to such interpretation, multiplicative logical connectives express properties of parallel threads of computation in a program. Despite these interesting ideas, the aforementioned paper does not contain a proof of the correspondence between the existence of winning strategies for the games introduced in the paper and provability in linear logic. One could argue that the correspondence is intuitively obvious and that writing a proof of it would only contribute to deforestation. We beg to dissent.

In our thesis we proved that winning strategies in a standard dialogical logic system correspond naturally to proofs in a polarized sequent calculus; we would like to extend this work to dialogical logic system for others logic such as modal logic and linear logic. As we have already argued, sequent systems do not have an intuitive reading in inferential terms. On the contrary, dialogical logic systems have an intuitive and pedagogical appeal. We therefore believe that it would be useful to develop dialogical logic systems for logics that are difficult to understand intuitively, such as modal logics and linear logic.

We also plan to study the counterpart of the cut rule in dialogical logic. We think that this could be obtained by relaxing the definition of game in order to let the proponent assert, at any point of the game, an arbitrary formula C. After the Proponent's assertion of C, the Opponent can continue the game by either attacking C or by asserting C in turn. The cut-admissibility theorem for strategies would be obtained by proving that the set of formulas admitting winning strategies containing this kind of games is equal to the class of formulas admitting 'regular' winning strategies.

We would like to conclude this subsection with some very general remarks on dialogical logic and argumentation. We have many times remarked that the games of dialogical logic are nothing but idealized argumentative dialogues. We use the adjective 'idealized' because argumentative dialogues (or debates) between real people have little chance of resembling those studied in dialogical logic. In a 'real' argumentative dialogue it is unlikely, though not impossible, that if one of the two disputants states "If Bill killed someone then Bill is a murderer" the other disputant will react by saying "Suppose Bill killed someone, can you prove that Bill is a murderer?". The artificial character of the games of dialogical logic is what makes them 'formalizable'. Given a finite sequence of moves, we can always decide whether it is a game in the sense of dialogical logic. We can also always decide whether it is a game won by the proponent. By contrast, we have no precise definition of what a debate between two (or more) people is. We have no clear criteria to determine neither who won a debate between two or more people, nor why a certain intervention counts as an 'attack to' or a 'defense from' another intervention in the debate. This situation is reflected in the very abstract definition of argumentation framework à la Dung [43].

Argumentation frameworks are pairs $(\mathcal{A}, \mathcal{R})$, where $\mathcal{A}$ is a set and $\mathcal{R}$ a binary relation over $\mathcal{A}$. The set $\mathcal{A}$ represents a set of arguments, and the relation $\mathcal{R}$ the 'attack' relation between arguments. In this setting, arguments do not have any internal structure and any couple of argument can be in the attack relation.

A less abstract implementation of argumentation frameworks are the so-called logic based argumentation frameworks, where arguments are formalized as follows: given a finite set Δ of logical formulas, an argument is a pair (Γ, A) where Γ is a minimal (with respect to inclusion) consistent subset of Δ such that $\Gamma \models A$. For example if $\Delta = \{A \wedge B, A \supset C, A \supset A\}$ the pair $(\{A \wedge B, A \supset C\}, C)$ is an argument while the pair $(\{A \supset A, A \wedge B\}, A \supset A)$ is not. In this less abstract setting the notion of 'an argument attacks another arguments' can be defined e.g., an argument (Γ, A) attacks an argument (Δ, C) if $A \models \neg B_1 \wedge \cdots \wedge \neg B_n$ where $\{B_1, \ldots B_n\} \subseteq \Delta$.

Despite the fact that this setting is less abstract and gives us a way to compute (at least in the predicate calculus case) attacks between arguments, we think that there is a problem with it. Intuitively an argument is something that grant the truth of a proposition i.e., albeit arguments are not exactly proof in the mathematical sense, they do share some similarities with proofs. In the setting of logic based argumentation frameworks 'arguments' in our sense are not taken into consideration: a logic based argument is, more or less, a provable sequent. No attention is paid to the proofs by means of which we establish that the sequent is provable. By considering arguments as proofs, new attack relations can arise. For instance: if we consider a valid formula A there still can be 'bad' arguments for A. We encounter such arguments quite often: incorrect 'proofs' of true statements are quite common in mathematical practice. It is by analyzing the structure of such 'proofs' that we find flaws.

To sum it up: dialogical logic games are too 'precise' to be considered a realistic analysis of argumentation: there is too much structure. On the contrary, argumentation frameworks cannot be considered a realistic analysis of argumentation because they are too abstract and permissive since there is not enough structure. We think that it would be interesting to try to find a compromise between the two approaches to argumentation. A starting point could be a 'formalization' of the kind of dialogues that are presented in the classic Book of Latakatos *Proofs and refutations* [88]. The book takes the form of a discussion between a teacher and some pupils. The dialogue participants analyze attempts to prove a mathematical statement. The setting of the dialogue that is presented in Lakatos's book is dialectical: some pupils tries to argue in favor of a given mathematical proposition, some others propose counter-examples or simply point out flaws in previous arguments. We think that it would be interesting and fruitful to develop games in the style of dialogical logic to model the dynamic of proposing arguments and refutations.

8.1.2 Game Semantic

We have presented game semantics for the constructive modal logic CK and CD. We would like to extend our work to some other logics of the so-called modal cube. In particular we would like to develop a game semantics for the constructive version of the modal logic T (CT) and S4 (CS4). These two logic are obtained by adding to the set of formulas of CK the following axiom schemes

$$\Box T\ :\ \Box A \supset A \qquad \Diamond T\ :\ A \supset \Diamond A$$

$$\Box 4\ :\ \Box A \supset \Box\Box A \qquad \Diamond 4\ :\ \Box(A \supset \Diamond B) \supset (\Diamond A \supset \Diamond B)$$

More precisely CT is obtained by adding the axiom schemes $\Box T$ and $\Diamond T$ to CK. CS4 is obtained by adding the axioms schemes $\Box 4$ and $\Diamond 4$ to CT. Apart from pure mathematical interest, there are other reasons that entice us to study the game semantics of these logics. Let us state two:

- The proof theory of CS4 has found applications in the analysis of staged computation i.e., a refined form of partial evaluation, in a functional programming context. The authors of [36] construct an ML-style typing system in which modalities can act on the evaluation properties of programs. The approach of the authors of the aforementioned paper is purely syntactical. We think that it would be interesting, and maybe instructive, to study such evaluation properties from a semantic perspective.

- It is known that the $\Box$-modality of CS4 behaves similarly to the !-modality of (intuitionistic) linear logic. In particular, The $\Box$-modality of the logic CS4 behaves similarly to the $!_x$-modality of bounded linear logic. The meaning of a formula $!_x A$ in bounded linear logic is "the resource A can be reused x times". The (simplified version) of sequent calculus rules for the $!_x$-modality are the following

$$\frac{\Gamma, B \vdash C}{\Gamma, !_1 B \vdash C} \qquad \frac{!_{\vec{y}}\Gamma \vdash C}{!_{x\vec{y}}\Gamma \vdash !_x C}$$

$$\frac{\Gamma \vdash C}{\Gamma, !_0 B \vdash C} \qquad \frac{\Gamma, !_x A, !_y A \vdash C}{\Gamma, !_{x+y} A \vdash C}$$

while the sequent calculus rules for the $\Box$-modality of CS4 are

$$\frac{\Gamma, B \vdash C}{\Gamma, \Box B \vdash C} \qquad \frac{\Box, \Gamma \vdash C}{\Box\Gamma \vdash \Box C}$$

The difference between the $\Box$-modality and the $!_x$-modality is that the latter deals with resources, i.e., contraction and weakening, while $\Box$ does not. The complexity of cut-elimination in bounded linear logic is polynomial i.e., any functional term (proof) of appropriate type actually encodes a polynomial-time algorithm and, conversely, any polynomial-time function can be obtained in this way. Bounded linear logic is generalized by the so-called graded modal type theories. The modality $!_n$ of bounded linear logic is indexed by elements of the semi-ring of natural numbers. The modalities $\Box_i$ of graded modal type theories are indexed by elements of an arbitrary semi-ring. Different graded modalities can be obtained by choosing different semi-rings. Graded modal type theories are used to conduct fine-grained analysis of the use of resources in programs. To our knowledge, no denotational semantic treatment of graded modal types is discussed in the literature. We would like to extend our game semantics of modal logic in order to capture the modalities of bounded linear logic and graded modal type theories. We think that this could be obtained by first studying the conditions that are needed to capture the behavior of the $\Box$-modality of CS4 in our game semantic setting. Once we have captured such behavior we could focus on the "resource-management" aspect of the graded modal types modalities. We believe that the resource management aspect can be captured by imposing restrictions on the repetition of **P**-moves in a game.

8.2 Applications of proof theory to natural language.

8.2.1 Syntactic terms and semantic readings

In the fifth chapter of this thesis we have presented a small result about the syntactic-semantic interface of Type Logical Grammars. This result is based on the introduction of many assumptions about the form of semantic lambda terms. This is due to the fact that we do not have a precise definition of a semantic term and it is also unlikely that one could exist. To prove this result we have moreover established that a syntactic term is nothing else but a linear lambda term in which some variables appear unbounded. Obviously, this characterization is far from optimal. Usually, categorial grammars are proper sub-systems of multiplicative intuitionistic linear logic. Consequently, the classes of lambda terms corresponding to these derivations — the *real* syntactic terms— are proper subclasses of the class of linear lambda terms. We suppose that the assumptions of our theorem can be relaxed if we consider such classes. We conjecture, for example, the following

> let P_1 and P_2 be two linear lambda terms of the same type 'corresponding' to normal proofs in the Lambek calculus. Suppose that $M_1, \ldots M_n$ are n simple semantic lambda terms such that the head constant of M_i is different from the head constant of M_j whenever $i \neq j$.
>
> $$\text{if } P_1 \neq_\beta P_2 \text{ then } P_1[M_1/x_1, \ldots M_n/x_n] \neq_\beta P_2[M_1/x_1, \ldots M_n/x_n]$$
>
> provided that $FV(P_1) = FV(P_2) = \{x_1, \ldots x_n\}$.

We have already obtained a preliminary result that goes in this direction. We have defined the notion of dominance over the proofs of the Lambek calculus and shown that two β-different proofs (in the sense of the lambek calculus) of the same judgment have different dominance relations. This result is false in the case of linear lambda terms. We have neither published this result nor presented it in our thesis because it is still not sufficient to prove the aforementioned conjecture.

8.2.2 Textual Entailment Recognition, anaphora, ellipsis and Dialogical Logic

The problems of textual entailment recognition that we have studied in our thesis are admittedly very simple. We believe that many aspects of inferences in natural language are too subtle to be captured by a logical system. They involve aspects of world knowledge and linguistic idiosyncrasies whose precise definition is difficult. Moreover, even for aspects of natural language that are well understood (e.g., plurals) logical modeling seems to be difficult. We have tried to incorporate more complex elements into the treatment of inferences: the meaning of non-logical words and the resolution of anaphora and ellipses. To obtain a treatment of inferences that are due to the linguistic meaning of the terms used, we introduced *Unfold rules* into dialogical logic. We considered a very simple framework in which the two players share the same meaning of linguistic terms and each atomic proposition is defined by at most another formula. In a sense this is because we assume that the two players are nothing more than two 'perfectly rational' beings. Two model speakers of the same language who know (and agree on) the meaning of each term. This model is clearly unrealistic. People often have a limited understanding of the meaning of words or, worse, associate the same word with definitions that are only partially overlapping or even diametrically opposed. We thus could study systems of Unfold games in which **P** and **O** play by 'using' different set of axioms or in which atomic proposition can unfold to more than one formula. It would be particularly interesting to study such a system to model reasoning and debates in natural language.

Debates often arise because two or more disputants cannot agree on the meaning of the terms used: i.e., they accord to the same linguistic term partially or diametrically opposed meanings. Debates in which the participants do not necessarily agree on the meaning of words have been studied using Ludics [51, 95]. It would then be a way to partially integrate such analysis in the simpler framework of dialogical logic.

We conclude by anticipating some of the work that needs to be done on the treatment of anaphoras and ellipsis by mean of $\mathscr{A}$-games. We would like to understand which first order structure characterize this logic. Even if we tend to prefer inferentialist semantics, we think that a model-theoretical understanding of the quantifier $\mathscr{A}$ is needed. In particular this would facilitate the task of comparing the logic 'defined' by $\mathscr{A}$-games to more traditional approach to anaphora resolution in natural language such as [80]. This comparison is indeed very important to test the limit of our logical system and eventually improve it. There is no precise linguistic definition of either 'anaphora' or 'ellipsis'. If one want to understand the depth of a logical contribution to this subject, empirical work is needed.

Bibliography

[1] S. Abramsky and R. Jagadeesan. Games and full completeness for multiplicative linear logic. *Journal of Symbolic Logic*, 59, 04 2003.

[2] S. Abramsky, P. Malacaria, and R. Jagadeesan. Full abstraction for pcf. In *International Symposium on Theoretical Aspects of Computer Software*, pages 1–15. Springer, 1994.

[3] V. M. Abrusci and L. Tortora De Falco. *Logica Volume 1: Dimostrazioni e modelli al primo ordine*. Springer Verlag, 2014.

[4] M. Acclavio, D. Catta, and L. Straßburger. Game semantics for constructive modal logic. In D. Anupam and N. Sara, editors, *Automated Reasoning with Analytic Tableaux and Related Methods, 30th International Conference, TABLEAUX 2021, Birmingham, UK, September 6–9, 2021, Proceedings*, volume 12842 of *Lecture Notes in Artificial Intelligence*, pages 428–445. Springer International Publishing, 2021.

[5] M. Acclavio, D. Catta, and L. Straßburger. Towards a Denotational Semantics for Proofs in Constructive Modal Logic. preprint, Apr. 2021.

[6] J. Alama, A. Knoks, and S. Uckelman. Dialogues games for classical logic, 2011. preprint https://www.researchgate.net/publication/228823176_Dialogue_games_for_classical_logic.

[7] M. Amblard, C. Beysson, P. de Groote, B. Guillaume, and S. Pogodalla. A French Version of the FraCaS Test Suite. In *LREC 2020 - Language Resources and Evaluation Conference*, page 9, Marseille, France, May 2020.

[8] J.-M. Andreoli. Logic programming with focusing proofs in linear logic. *Journal of Logic and Computation*, 2:297–347, 1992.

[9] H. P. Barendregt. *The lambda calculus: its syntax and semantics*, volume 103 of *Studies in Logic and the Foundations of Mathematics*. Elsevier, 1984.

[10] C. Barker and C. Shan. Donkey anaphora is in-scope binding. *Semantics and Pragmatics*, 1:1–42, 2008.

[11] G. Bellin, V. De Paiva, and E. Ritter. Extended Curry-Howard correspondence for a basic constructive modal logic. In *In Proceedings of Methods for Modalities*, 05 2001.

[12] G. M. Bierman and V. C. de Paiva. On an intuitionistic modal logic. *Studia Logica*, 65(3):383–416, 2000.

[13] A. Blass. A game semantics for linear logic. *Annals of Pure and Applied Logic*, 56(1):183–220, 1992.

[14] J. Bos. Vp ellipsis in a drt-implementation. In *Sixth Conference of the European Chapter of the Association for Computational Linguistics*, 1993.

[15] R. Brandom. *Articulating Reasons: An Introduction to Inferentialism*. Harvard University Press, 2000.

[16] R. B. Brandom. *Making It Explicit: Reasoning, Representing, and Discursive Commitment*. Harvard University Press, 1994.

[17] P. Brauner, C. Houtmann, and C. Kirchner. Principles of superdeduction. In *Proceedings of Symposium on Logic in Computer Science*, page 41–50, 08 2007.

[18] S. E. Brennan, M. W. Friedman, and C. Pollard. A centering approach to pronouns. In *25th Annual Meeting of the Association for Computational Linguistics*, pages 155–162, 1987.

[19] B. Castelnérac and M. Marion. Arguing for inconsistency: Dialectical games in the academy. In G. Primiero, editor, *Acts of Knowledge: History, Philosophy and Logic*. College Publications, 2009.

[20] B. Castelnérac and M. Marion. Antilogic. *The Baltic International Yearbook of Cognition, Logic and Communication*, 8(1), 2013.

[21] D. Catta. From strategies to derivations and back. an easy completeness proof for first order intuitionistic dialogical logic. In A. Piccolomini d'Aragona, editor, *Perspectives on Deduction*. Springer, (To appear).

[22] D. Catta, R. Moot, and C. Retoré. Dialogical argumentation and textual entailment. In R. Loukanova, editor, *Natural Language Processing in Artificial Intelligence—NLPinAI 2020.*, volume 939 of *Studies in Computational Intelligence*. Springer, 2021.

[23] D. Catta, R. Moot, and C. Retoré. Do different syntactic trees yield different logical readings? some remarks on head variables in typed lambda calculus. EasyChair Preprint no. 209, EasyChair, 2018.

[24] D. Catta and S. Stevens-Guille. Lorenzen won the game, lorenzen did too: Dialogical logic for anaphora and ellipsis resolution. In A. Silva, R. Wasserman, and R. de Queiroz, editors, *Logic, Language, Information and Computation. WoLLIC 2021*, volume 13038 of *Lecture Notes in Computer Science*. Springer, 2021.

[25] N. Clerbout. First-order dialogical games and tableaux. *Journal of Philosophical Logic*, 43(4):785–801, 2014.

[26] N. Clerbout, M.-H. Gorisse, and S. Rahman. Context-sensitivity in jain philosophy: A dialogical study of siddarsighani's commentary on the handbook of logic. *Journal of Philosophical Logic*, 40(5):633–662, 2011.

[27] A. Colapinto. Do it anaphora without covert events: In defense of a pro-verb analysis. *Lingua*, 245:102921, 2020.

[28] R. Cooper, D. Crouch, J. Van Eijck, C. Fox, J. V. Genabith, J. Jaspars, H. Kamp, D. Milward, M. Pinkal, M. Poesio, S. Pulman, T. Briscoe, H. Maier, and K. Konrad. Using the framework, 1996. FraCaS deliverable D16.

[29] T. Coquand. A semantics of evidence for classical arithmetic. *Journal of Symbolic Logic*, 60(1):325–337, 1995.

[30] C. Cozzo. *Meaning and Argument: A Theory of Meaning Centred on Immediate Argumental Role*. Stockholm Studies in Philosophy. Almqvist & Wiksell International, 1994.

[31] P. Curien. Notes on game semantics, 2006.
Online version https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.676.7186&rep=rep1&type=pdf.

[32] I. Dagan, D. Roth, M. Sammons, and F. M. Zanzotto. *Recognizing textual entailment: Models and applications*, volume 6(4) of *Synthesis Lectures on Human Language Technologies*. Morgan & Claypool Publishers, 2013.

[33] M. Dalrymple, S. M. Shieber, and F. C. Pereira. Ellipsis and higher-order unification. *Linguistics and philosophy*, 14(4):399–452, 1991.

[34] V. Danos, H. Herbelin, and L. Regnier. Game semantics and abstract machines. In *Proceedings of the 11th Annual IEEE Symposium on Logic in Computer Science*, LICS '96, page 394, USA, 1996. IEEE Computer Society.

[35] D. Davidson. The logical form of action sentences. In N. Rescher, editor, *The Logic of Decision and Action*, pages 81–95. University of Pittsburgh Press, 1967.

[36] R. Davies and F. Pfenning. A modal analysis of staged computation. *Journal of the ACM (JACM)*, 48(3):555–604, 2001.

[37] P. Dekker. Predicate logic with anaphora. In *Dynamic Semantics*, pages 7–47. Springer, 2012.

[38] P. Dekker. Exclusively indexical deduction. *The Review of Symbolic Logic*, 9(3):603–637, 2016.

[39] D. Delahaye. Automated Deduction Modulo. In *Proof-Search in Axiomatic Theories and Type Theories (PSATTT)*, page online, Palaiseau, France, Nov. 2013.

[40] G. Dowek. About folding-unfolding cuts and cuts modulo. *Journal of Logic and Computation*, 11:419–429, 11 2000.

[41] M. A. E. Dummett. What is a theory of meaning? In S. Guttenplan, editor, *Mind and Language*. Oxford University Press, 1975.

[42] M. A. E. Dummett. *The Logical Basis of Metaphysics*. Harvard University Press, 1991.

[43] P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2):321–357, 1995.

[44] C. Dutilh Novaes. Medieval "obligationes" as logical games of consistency maintenance. *Synthese*, 145(3):371–395, 2005.

[45] M. Fairtlough and M. Mendler. Propositional lax logic. *Information and Computation*, 137(1):1–33, 1997.

[46] W. Felscher. Dialogues, strategies, and intuitionistic provability. *Annals of Pure and Applied Logic*, 28(3):217 – 254, 1985.

[47] W. Felscher. Dialogues as a foundation for intuitionistic logic. In D. M. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic*, pages 115–145. Springer Netherlands, Dordrecht, 2002.

[48] C. G. Fermüller. Parallel dialogue games and hypersequents for intermediate logics. In M. Cialdea Mayer and F. Pirri, editors, *Automated Reasoning with Analytic Tableaux and Related Methods*, pages 48–64, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

[49] F. B. Fitch. Intuitionistic modal logic with quantifiers. *Portugaliae mathematica*, 7(2):113–118, 1948.

[50] M. Fontaine and J. Redmond. *Logique Dialogique: une introduction, Volume 1: Méthode de Dialogique: Règles et Exercies*, volume 3 of *Cahiers de logique et d'Epistémologie*. College Publications, 2008.

[51] C. Fouqueré and M. Quatrini. Ludics and Natural Language: First Approaches. In A. D. Denis Béchet, editor, *LACL 2012 seventh International Conference on Logical Aspects of Computational Linguistics*, volume 7351 of *Logical Aspects of Computational Linguistics, 7th International Conference, LACL 2012, Nantes, France, July 2-4, 2012. Proceedings*, pages 21–44, Nantes, France, July 2012. Laboratoire d'Informatique Nantes Atlantique, Springer.

[52] C. Fouqueré and M. Quatrini. Argumentation and inference a unified approach. In *The Baltic International Yearbook of Cognition, Logic and Communication Volume 8: Games, Game Theory and Game Semantics*, pages 1–41. New Paririe Press, 2013.

[53] N. Francez. *Proof Theoretical Semantics*, volume 57 of *Studies in Logic*. College Publication, 2015.

[54] J. H. Gallier. *Logic for Computer Science: Foundations of Automatic Theorem Proving, Second Edition*. Dover Publications, Inc., USA, 2015.

[55] D. Galmiche, D. Larchey-Wendling, and J. Vidal-Rosset. Some Remarks on Relations between Proofs and Games. In P. E. Bour, M. Rebuschi, and L. Rollet, editors, *Construction - Festschrift for Gerhard Heinzmann*. College Publications, 2010.

[56] C. Gardent. Sloppy identity. In *International Conference on Logical Aspects of Computational Linguistics*, pages 188–207. Springer, 1996.

[57] B. Gehrke. Event kinds. *The Oxford handbook of event structure*, 205:233, 2019.

[58] G. Gentzen. Untersuchungen über das logische Schließen. I. *Mathematische Zeitschrift*, 39:176–210, 1935.

[59] J.-Y. Girard. Linear logic. *Theor. Comput. Sci.*, 50(1):1–102, Jan. 1987.

[60] J.-Y. Girard. *Proof Theory and Logical Complexity*. Bibliopolis, Napoli, 1987.

[61] J.-Y. Girard. Locus solum: From the rules of logic to the logic of rules. *Mathematical Structures in Computer Science*, 11(3):301–506, 2001.

[62] J.-Y. Girard. *The blind spot – lectures on logic*. European Mathematical Society, 2011.

[63] J.-Y. Girard, P. Taylor, and Y. Lafont. *Proofs and Types*. Cambridge University Press, USA, 1989.

[64] J. Groenendijk and M. Stokhof. Dynamic predicate logic. *Linguistics and philosophy*, pages 39–100, 1991.

[65] J. Hankamer. On the nontransformational derivation of some null vp anaphors. *Linguistic Inquiry*, 9(1):66–74, 1978.

[66] S. Heilala and B. Pientka. Bidirectional decision procedures for the intuitionistic propositional modal logic IS4. In *International Conference on Automated Deduction*, pages 116–131. Springer, 2007.

[67] I. Heim. *The semantics of definite and indefinite noun phrases*. PhD thesis, University of Massachusetts Amherst, 1982.

[68] H. Herbelin. *Séquents qu'on calcule : de l'interprétation du calcul des séquents comme calcul de λ-termes et comme calcul de stratégies gagnantes*. Phd thesis, Université Paris 7, Janvier 1995.

[69] J. Higginbotham. The logic of perceptual reports: An extensional alternative to situation semantics. *The Journal of Philosophy*, 80(2):100–127, 1983.

[70] J. Higginbotham. Linguistic theory and davidson's program in semantics. In E. LePore, editor, *Truth and Interpretation: Perspectives on the Philosophy of Donald Davidson*, pages 29–48. Cambridge: Blackwell, 1986.

[71] J. R. Hindley. *Basic Simple Type Theory*, volume 42 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 1997. Corrected edition, 2008.

[72] J. R. Hobbs. Coherence and coreference. *Cognitive science*, 3(1):67–90, 1979.

[73] R. Horne, K. Y. Ahn, S.-w. Lin, and A. Tiu. Quasi-open bisimilarity with mismatch is intuitionistic. In *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS '18, page 26–35, New York, NY, USA, 2018. Association for Computing Machinery.

[74] R. Huddleston, G. K. Pullum, et al. The cambridge grammar of english. *Language. Cambridge: Cambridge University Press*, 1:23, 2002.

[75] M. Hyland. Game semantics. In A. Pitts and P. Dybjer, editors, *Semantics and Logics of Computation*, pages 131–182. Cambridge University Press, 1997.

[76] M. Hyland and L. Ong. Fair games and full completeness for multiplicative linear logic without the mix-rule, 1993.

[77] M. Hyland and L. Ong. On full abstraction for PCF: I, II, and III. *Information and Computation*, 163(2):285 – 408, 2000.

[78] G. Jäger. *Anaphora and type logical grammar*, volume 24. Springer Science & Business Media, 2006.

[79] I. Johansson. Der minimalkalkül, ein reduzierter intuitionistischer formalismus. *Compositio Mathematica*, 4:119–136, 1937.

[80] H. Kamp. A theory of truth and semantic representation. In P. Portner and B. H. Partee, editors, *Formal Semantics — the Essential Readings*, pages 189–222. Blackwell, 1981.

[81] H. Kamp and U. Reyle. *From discourse to logic: introduction to model theoretic semantics of natural language, formal logic and discourse representation theory. Part 1*. Kluwer Academic, 1993.

[82] A. Kenny. *Action, emotion and will*. Routledge, 2003.

[83] J. King and K. Lewis. Anaphora. In Edward N., editor, *The Stanford Encyclopedia of Philosophy*. Fall 2018 edition, 2018. https://plato.stanford.edu/archives/fall2018/entries/anaphora/.

[84] M. Kohlhase. Model generation for discourse representation theory. In *ECAI*, pages 441–445, 2000.

[85] K. Kojima. *Semantical study of intuitionistic modal logics*. PhD thesis, Kyoto University, 2012. Ph.D. thesis.

[86] E. C. W. Krabbe. Formal systems of dialogue rules. *Synthese*, 63(3):295–328, 1985.

[87] R. Kuznets, S. Marin, and L. Straßburger. Justification logic for constructive modal logic *. IMLA 2017 - 7th Workshop on Intuitionistic Modal Logic and Applications, July 2017.

[88] I. Lakatos, J. Worrall, and E. Zahar. *Proofs and refutations : the logic of mathematical discovery / Imre Lakatos ; edited by John Worrall and Elie Zahar*. Cambridge University Press Cambridge ; New York, 1976.

[89] F. Lamarche. Sequentiality, games and linear logic (manuscript), 1992.

[90] J. Lambek. The mathematics of sentence structure. *American Mathematical Monthly*, 65:154–170, 1958.

[91] J. Lambek. The mathematics of sentence structure. *The American Mathematical Monthly*, 65(3):154–170, 1958.

[92] O. Laurent. Syntax vs. Semantics: a polarized approach. *Theoretical Computer Science*, 343:177–206, 2005.

[93] O. Laurent. Game semantics for first-order logic. *Logical Methods in Computer Science*, 6(4):3, Oct. 2010.

[94] A. Lecomte. *Meaning, Logic and Ludics*. Imperial College Press, 2011.

[95] A. Lecomte and M. Quatrini. Figures of dialogue: a view from ludics. *Synthese*, 183(1):59–85, 2011.

[96] P. Lorenzen. Logik und agon. *Atti Del XII Congresso Internazionale di Filosofia*, 4:187–194, 1958.

[97] P. Lorenzen and K. Lorenz. *Dialogische Logik*. Wissenschaftliche Buchgesellschaft, [Abt. Verlag], 1978.

[98] M. Marion. Between saying and doing: From lorenzen to brandom and back. In Pierre Edouard Bour, M. Rebuschi, and L. Rollet, editors, *Constructions. Essays in Honour od Gerhard Heinzmann*, pages 489–497. College Publications, London.

[99] M. Marion and H. Rückert. Aristotle on universal quantification: A study from the point of view of game semantics. *History and Philosophy of Logic*, 37(3):201–229, 2016.

[100] G. McCusker. Games and full abstraction for FPC. *Information and Computation*, 160(1):1 – 61, 2000.

[101] M. Mendler and S. Scheele. Cut-free gentzen calculus for multimodal CK. *Information and Computation*, 209(12):1465–1490, 2011.

[102] J. Merchant. Ellipsis: A survey of analytical approaches. *The Oxford handbook of ellipsis*, pages 18–46, 2019.

[103] J.-J. Meyer and F. Veltmanw. Intelligent agents and common sense reasoning. In P. Blackburn, J. Van Benthem, and F. Wolter, editors, *Handbook of Modal Logic*, volume 3 of *Studies in Logic and Practical Reasoning*, pages 991 – 1029. Elsevier, 2007.

[104] P. Miller and G. K. Pullum. Exophoric vp ellipsis. *The core and the periphery: Data-driven perspectives on syntax inspired by Ivan A. Sag*, 5:32, 2013.

[105] R. Montague. The proper treatment of quantification in ordinary english. In *Approaches to natural language*, pages 221–242. Springer, 1973.

[106] R. Montague. The proper treatment of quantification in ordinary English. In R. Thomason, editor, *Formal Philosophy. Selected Papers of Richard Montague*. Yale University Press, New Haven, 1974.

[107] R. Moot. The Grail Theorem Prover: Type Theory for Syntax and Semantics. In Z. Luo and S. Chatzikyriakidis, editors, *Modern Perspectives in Type-Theoretical Semantics*, volume Studies in Linguistics and Philosophy of *Part III*, pages 247–277. Springer, 2017.

[108] R. Moot and C. Retoré. *The Logic of Categorial Grammars: A Deductive Account of Natural Language Syntax and Semantics*. Number 6850 in Lecture Notes in Artificial Intelligence. Springer, 2012.

[109] R. Moot and C. Retoré. Natural language semantics and computability. Technical report, arXiv, 2016.

[110] A. Mori and Y. Matsumoto. Coherence for cartesian closed categories: A sequential approach. In N. Dershowitz and N. Lindenstrauss, editors, *Conditional and Typed Rewriting Systems*, pages 276–295, Berlin, Heidelberg, 1995. Springer Berlin Heidelberg.

[111] G. V. Morrill. *Type logical grammar: Categorial logic of signs*. Springer Science & Business Media, 2012.

[112] L. Moss. Natural logic. In S. Lappin and C. Fox, editors, *Handbook of Contemporary Semantic Theory*, chapter 18, pages 561–592. John Wiley and Sons, second edition, 2015.

[113] A. S. Murawski and C. H. Luke Ong. Evolving games and essential nets for affine polymorphism. In S. Abramsky, editor, *Typed Lambda Calculi and Applications*, pages 360–375, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

[114] G. Plotkin and C. Stirling. A framework for intuitionistic modal logics. In *Proceedings of the 1st Conference on Theoretical Aspects of Reasoning about Knowledge (TARK)*, pages 399–406, 1986.

[115] D. Prawitz. Towards a foundation of general proof theory. In *Įtetsuppes1973*. North Holland, Amsterdam, 1973.

[116] D. Prawitz. On the idea of a general proof theory. *Synthese*, 27(1-2):63–77, 1974.

[117] D. Prawitz. *Natural deduction: A proof-theoretical study*. Courier Dover Publications, 2006.

[118] D. Prawitz. The epistemic significance of valid inference. *Synthese*, 187(3):887–898, Aug 2012.

[119] J. Pustejovsky. The generative lexicon. *Computational Linguistics*, 17, 07 2002.

[120] S. Rahman. On Frege's Nightmare: A Combination of Intuitionistic, Free and Paraconsistent Logics. In *essays on non-classical logic*. 2001.

[121] S. Rahman. Un desafío para las teorías cognitivas de la competencia lógica: los fundamentos pragmáticos de la semántica de la lógica linear. *Manuscrito*, 25:381–432, 2002.

[122] S. Rahman and W. A. Carnielli. The dialogical approach to paraconsistency. *Synthese*, 125(1):201–232, 2000.

[123] S. Rahman, N. Clerbout, and L. Keiff. On Dialogues and Natural Deduction. In G. P. et alii, editor, *Acts of Knowledge: History and Philosophy of Logic*, College Publictions. Tributes, pages 301–336. College Publications, 2009.

[124] G. Restall. Multiple conclusions. In P. Hájek, L. Valdés-Villanueva, and D. Westerståhl, editors, *Logic, Methodology and Philosophy of Science*. College Publications, 2005.

[125] C. Retoré. The montagovian generative lexicon lambda tyn: a type theoretical framework for natural language semantics. In *19th International Conference on Types for Proofs and Programs (TYPES 2013)*, volume 26, pages 202–229, 2014.

[126] P. Schroeder-Heister. Cut-elimination in logics with definitional reflection. In D. Pearce and H. Wansing, editors, *Nonclassical Logics and Information Processing*, volume 619 of *Lecture Notes in Computer Science (Lecture Notes in Artificial Intelligence)*. Springer, Berlin, Heidelberg., 1992.

[127] P. Schroeder-Heister. Proof Theoretic Semantic. In E. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, spring 2018 edition, 2018. https://plato.stanford.edu/archives/spr2018/entries/proof-theoretic-semantics/.

[128] A. K. Simpson. *The proof theory and semantics of intuitionistic modal logic*. PhD thesis, University of Edinburgh. College of Science and Engineering, 1994.

[129] W. Stegmüller. Remarks on the completeness of logical systems relative to the validity-concepts of p. lorenzen and k. lorenz. *Notre Dame Journal of Formal Logic*, 5(2):81–112, 1964.

[130] F. Steinberger. Why conclusions should remain single. *Journal of Philosophical Logic*, (3):333–355, 2011.

[131] L. Straßburger, W. Heijltjes, and D. J. D. Hughes. Intuitionistic proofs without syntax. In *LICS 2019 - 34th Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 1–13, Vancouver, Canada, June 2019. IEEE.

[132] A. S. Troelstra and H. Schwichtenberg. *Basic Proof Theory*. Cambridge University Press, USA, 1996.

[133] R. Truswell. *The Oxford Handbook of Event Structure*. Oxford University Press, 2019.

[134] D. Vakarelov. Modal logics for knowledge representation systems. *Theor. Comput. Sci.*, 90:433–456, 01 1991.

[135] J. van Benthem. *Language in Action: Categories, Lambdas and Dynamic Logic*. MIT Press, Cambridge, Massachusetts, 1995.

[136] D. van Dalen. *Logic and Structure*. Universitext. Springer-Verlag, fifth edition, 2013.

[137] R. A. Van der Sandt. Presupposition projection as anaphora resolution. *Journal of semantics*, 9(4):333–377, 1992.

[138] J. van Eijck and N. Francez. Verb-phrase ellipsis in dynamic semantics. In *Applied Logic: How, What and Why*, pages 29–59. Springer, 1995.

[139] J. van Eijck, J. Heguiabehere, and B. Ó Nualláin. Tableau reasoning and programming with dynamic first order logic. *Logic Journal of IGPL*, 9(3):411–445, 2001.

[140] Z. Vendler. *Linguistics in philosophy*. Cornell University Press, 2019.

[141] C. F. M. Vermeulen. Sequence semantics for dynamic predicate logic. *Journal of Logic, Language and Information*, 2(3):217–254, 1993.

[142] H. Wansing. Formulas-as-types for a hierarchy of sublogics of intuitionistic propositional logic. In D. Pearce and H. Wansing, editors, *Nonclassical Logics and Information Processing*, pages 125–145. Springer, Berlin, Heidelberg, 1992.