Modélisation de la langue naturelle et Grammaires Catégorielles Abstraites

Florent Pompigne

florent.pompigne@loria.fr
LORIA/Université de Lorraine

11 décembre 2013

Contexte

- Domaine : Traitement Automatique de la Langue

Contexte

- Domaine : Traitement Automatique de la Langue
- Niveaux linguistiques : syntaxe et sémantique

Contexte

- Domaine : Traitement Automatique de la Langue
- Niveaux linguistiques : syntaxe et sémantique
- Approche : méthodes symboliques

Contexte

- Domaine : Traitement Automatique de la Langue
- Niveaux linguistiques : syntaxe et sémantique
- Approche : méthodes symboliques
- Famille : Grammaires Catégorielles

Contexte

- Domaine : Traitement Automatique de la Langue
- Niveaux linguistiques : syntaxe et sémantique
- Approche : méthodes symboliques
- Famille : Grammaires Catégorielles
- Formalisme : Grammaires Catégorielles Abstraites (ACG)

Motivations et contributions

Extension du système de typage des ACG

Motivations et contributions

Extension du système de typage des ACG

Contributions :

- Modélisation : dépendances à distance

Motivations et contributions

Extension du système de typage des ACG

Contributions :

- Modélisation : dépendances à distance
- Preuve des propriétés de réduction du calcul

Motivations et contributions

Extension du système de typage des ACG

Contributions :

- Modélisation : dépendances à distance
- Preuve des propriétés de réduction du calcul
- Étude de ces propriétés si le calcul est affiné (η-conversion)

Grammaires Catégorielles Abstraites

- Un exemple introductif
- Langage généré

Extension du système de typage

- Principe de l'extension
- Contrôle des mouvements explicites

Propriétés du calcul

- Propriétés de $\longrightarrow_{\beta\gamma}$
- Propriétés de $\longrightarrow_{\beta\gamma\eta}$

Signature

Liste de constantes typées

Signature

Liste de constantes typées

Σ_1 (structures de dérivation) :

MARIE	: $\boldsymbol{NP}$
CHAT	: $\boldsymbol{N}$
UN	: $\boldsymbol{N} \multimap \boldsymbol{NP}$
VOIT	: $\boldsymbol{NP} \multimap \boldsymbol{NP} \multimap \boldsymbol{S}$

Signature

Liste de constantes typées

Σ_1 (structures de dérivation) :

MARIE	: $\boldsymbol{NP}$
CHAT	: $\boldsymbol{N}$
UN	: $\boldsymbol{N} \multimap \boldsymbol{NP}$
VOIT	: $\boldsymbol{NP} \multimap \boldsymbol{NP} \multimap \boldsymbol{S}$

Σ_2 (formes de surface) :

/Marie/, /chat/, /un/, /voit/ : **string**

Lexique

Morphisme d'un niveau abstrait vers un niveau objet

Lexique

Morphisme d'un niveau abstrait vers un niveau objet

$\mathscr{L}_{\text{Syn}}$:

$$\boldsymbol{S}, \boldsymbol{NP}, \boldsymbol{N} \quad :=_{\text{Syn}} \quad \textbf{string}$$

Lexique

Morphisme d'un niveau abstrait vers un niveau objet

$\mathscr{L}_{\text{Syn}}$:

$$
\begin{array}{lll}
\boldsymbol{S}, \boldsymbol{NP}, \boldsymbol{N} & :=_{\text{Syn}} & \textbf{string} \\
\textsc{Marie} & :=_{\text{Syn}} & /\text{Marie}/ \\
\textsc{chat} & :=_{\text{Syn}} & /\text{chat}/ \\
\textsc{un} & :=_{\text{Syn}} & \lambda^0 n.(/\text{un}/ + n) \\
\textsc{voit} & :=_{\text{Syn}} & \lambda^0 o, s.(s + /\text{voit}/ + o)
\end{array}
$$

Lexique

Morphisme d'un niveau abstrait vers un niveau objet

$\mathscr{L}_{\text{Syn}}$:

$$
\begin{array}{lll}
\boldsymbol{S}, \boldsymbol{NP}, \boldsymbol{N} & :=_{\text{Syn}} & \textbf{string} \\
\textsc{Marie} & :=_{\text{Syn}} & /\text{Marie}/ \\
\textsc{chat} & :=_{\text{Syn}} & /\text{chat}/ \\
\textsc{un} & :=_{\text{Syn}} & \lambda^0 n.(/\text{un}/ + n) \\
\textsc{voit} & :=_{\text{Syn}} & \lambda^0 o, s.(s + /\text{voit}/ + o)
\end{array}
$$

Condition :
$c : \alpha \Rightarrow \mathscr{L}(c) : \mathscr{L}(\alpha)$

Langage abstrait

Σ_1 : MARIE : $\boldsymbol{NP}$
CHAT : $\boldsymbol{N}$
UN : $\boldsymbol{N} \multimap \boldsymbol{NP}$
VOIT : $\boldsymbol{NP} \multimap \boldsymbol{NP} \multimap \boldsymbol{S}$

VOIT (UN CHAT) MARIE

Langage abstrait

Σ_1 : MARIE : $\boldsymbol{NP}$
CHAT : $\boldsymbol{N}$
UN : $\boldsymbol{N} \multimap \boldsymbol{NP}$
VOIT : $\boldsymbol{NP} \multimap \boldsymbol{NP} \multimap \boldsymbol{S}$

$$\text{VOIT } \underbrace{(\text{UN CHAT})}_{\boldsymbol{NP}} \text{ MARIE}$$

Langage abstrait

$$\begin{array}{lll} \Sigma_1 : & \textsc{Marie} & : \boldsymbol{NP} \\ & \textsc{chat} & : \boldsymbol{N} \\ & \textsc{un} & : \boldsymbol{N} \multimap \boldsymbol{NP} \\ & \textsc{voit} & : \boldsymbol{NP} \multimap \boldsymbol{NP} \multimap \boldsymbol{S} \end{array}$$

$$\underbrace{\textsc{voit}\ \underbrace{(\textsc{un chat})}_{\boldsymbol{NP}}\ \textsc{Marie}}_{\boldsymbol{S}}$$

Langage abstrait

$$\Sigma_1 : \quad \begin{array}{ll} \text{MARIE} & : \boldsymbol{NP} \\ \text{CHAT} & : \boldsymbol{N} \\ \text{UN} & : \boldsymbol{N} \multimap \boldsymbol{NP} \\ \text{VOIT} & : \boldsymbol{NP} \multimap \boldsymbol{NP} \multimap \boldsymbol{S} \end{array}$$

$$\underbrace{\text{VOIT } \underbrace{(\text{UN CHAT})}_{\boldsymbol{NP}} \text{ MARIE}}_{\boldsymbol{S}} \in \mathcal{A}(\mathscr{G})$$

Langage objet

$t : \boldsymbol{s}$

Langage objet

$$t : \boldsymbol{S} := \mathscr{L}(t) : \mathscr{L}(\boldsymbol{S})$$

Langage objet

$$t : \boldsymbol{S} := \mathscr{L}(t) : \mathscr{L}(\boldsymbol{S}) \longrightarrow_{\beta}^{*} \mathit{FN}(\mathscr{L}(t)) : \mathscr{L}(\boldsymbol{S})$$

Langage objet

$$t : \boldsymbol{S} \;:=\; \mathscr{L}(t) : \mathscr{L}(\boldsymbol{S}) \;\longrightarrow^{*}_{\beta}\; \mathit{FN}(\mathscr{L}(t)) : \mathscr{L}(\boldsymbol{S})$$

VOIT (UN CHAT) MARIE

$:=_{\mathrm{Syn}}$

Langage objet

$$t : \boldsymbol{S} := \mathscr{L}(t) : \mathscr{L}(\boldsymbol{S}) \longrightarrow^*_\beta FN(\mathscr{L}(t)) : \mathscr{L}(\boldsymbol{S})$$

VOIT (UN CHAT) MARIE

$$:=_{\text{Syn}} (\lambda^0 o, s.(s + /\text{voit}/ + o))$$

Langage objet

$$t : \boldsymbol{S} := \mathscr{L}(t) : \mathscr{L}(\boldsymbol{S}) \longrightarrow^*_\beta FN(\mathscr{L}(t)) : \mathscr{L}(\boldsymbol{S})$$

VOIT (UN CHAT) MARIE

$$:=_{\text{Syn}} (\lambda^0 o, s.(s + /\text{voit}/ + o))\ ((\lambda^0 n.(/\text{un}/ + n))$$

Langage objet

$$t : \boldsymbol{S} := \mathscr{L}(t) : \mathscr{L}(\boldsymbol{S}) \longrightarrow^*_\beta FN(\mathscr{L}(t)) : \mathscr{L}(\boldsymbol{S})$$

VOIT (UN CHAT) MARIE

$$:=_{\text{Syn}} (\lambda^0 o, s.(s + /\text{voit}/ + o))\ ((\lambda^0 n.(/\text{un}/ + n))\ /\text{chat}/)$$

Langage objet

$$t : \boldsymbol{S} := \mathscr{L}(t) : \mathscr{L}(\boldsymbol{S}) \longrightarrow_{\beta}^{*} FN(\mathscr{L}(t)) : \mathscr{L}(\boldsymbol{S})$$

VOIT (UN CHAT) MARIE

$$:=_{\text{Syn}} (\lambda^0 o, s.(s + /\text{voit}/ + o))\ ((\lambda^0 n.(/\text{un}/ + n))\ /\text{chat}/)\ /\text{Marie}/$$

Langage objet

$$t : \boldsymbol{S} := \mathscr{L}(t) : \mathscr{L}(\boldsymbol{S}) \longrightarrow^*_\beta FN(\mathscr{L}(t)) : \mathscr{L}(\boldsymbol{S})$$

VOIT (UN CHAT) MARIE

$$:=_{\text{Syn}} (\lambda^0 o, s.(s + /\text{voit}/ + o))\ ((\lambda^0 n.(/\text{un}/ + n))\ /\text{chat}/)\ /\text{Marie}/$$

$$\longrightarrow_\beta (\lambda^0 o, s.(s + /\text{voit}/ + o))\ (/\text{un}/ + /\text{chat}/)\ /\text{Marie}/$$

Langage objet

$$t : \boldsymbol{S} := \mathscr{L}(t) : \mathscr{L}(\boldsymbol{S}) \longrightarrow^*_\beta FN(\mathscr{L}(t)) : \mathscr{L}(\boldsymbol{S})$$

VOIT (UN CHAT) MARIE

$$:=_{\text{Syn}} (\lambda^0 o, s.(s + /\text{voit}/ + o))\ ((\lambda^0 n.(/\text{un}/ + n))\ /\text{chat}/)\ /\text{Marie}/$$

$$\longrightarrow_\beta (\lambda^0 o, s.(s + /\text{voit}/ + o))\ (/\text{un}/ + /\text{chat}/)\ /\text{Marie}/$$

$$\longrightarrow_\beta (\lambda^0 s.s + /\text{voit}/ + /\text{un}/ + /\text{chat}/)\ /\text{Marie}/$$

Langage objet

$$t : \boldsymbol{S} := \mathscr{L}(t) : \mathscr{L}(\boldsymbol{S}) \longrightarrow^*_\beta FN(\mathscr{L}(t)) : \mathscr{L}(\boldsymbol{S})$$

VOIT (UN CHAT) MARIE

$$:=_{\text{Syn}} (\lambda^0 o, s.(s + /\text{voit}/ + o))\ ((\lambda^0 n.(/\text{un}/ + n))\ /\text{chat}/)\ /\text{Marie}/$$

$$\longrightarrow_\beta (\lambda^0 o, s.(s + /\text{voit}/ + o))\ (/\text{un}/ + /\text{chat}/)\ /\text{Marie}/$$

$$\longrightarrow_\beta (\lambda^0 s.s + /\text{voit}/ + /\text{un}/ + /\text{chat}/)\ /\text{Marie}/$$

$$\longrightarrow_\beta /\text{Marie}/ + /\text{voit}/ + /\text{un}/ + /\text{chat}/$$

Langage objet

$$t : \boldsymbol{S} := \mathscr{L}(t) : \mathscr{L}(\boldsymbol{S}) \longrightarrow^*_\beta FN(\mathscr{L}(t)) : \mathscr{L}(\boldsymbol{S})$$

VOIT (UN CHAT) MARIE

$$:=_{\text{Syn}} (\lambda^0 o, s.(s + /\text{voit}/ + o))\ ((\lambda^0 n.(/\text{un}/ + n))\ /\text{chat}/)\ /\text{Marie}/$$

$$\longrightarrow_\beta (\lambda^0 o, s.(s + /\text{voit}/ + o))\ (/\text{un}/ + /\text{chat}/)\ /\text{Marie}/$$

$$\longrightarrow_\beta (\lambda^0 s.s + /\text{voit}/ + /\text{un}/ + /\text{chat}/)\ /\text{Marie}/$$

$$\longrightarrow_\beta /\text{Marie}/ + /\text{voit}/ + /\text{un}/ + /\text{chat}/ \in \mathcal{O}(\mathscr{G})$$

Constructions ajoutées

- Produit cartésien & :
  morph $=[\mathtt{g} : \textbf{genre}, \mathtt{n} : \textbf{nombre}] : \mathtt{type}$
  $[\mathtt{g} = f, \mathtt{n} = s]$ est un terme de type **morph**

Constructions ajoutées

- Produit cartésien & :
  morph $= [\mathrm{g} : \textbf{genre}, \mathrm{n} : \textbf{nombre}] : \texttt{type}$
  $[\mathrm{g} = f, \mathrm{n} = s]$ est un terme de type **morph**

- Somme disjointe $\oplus$:
  genre $= \{m \mid f\} : \texttt{type}$ **nombre** $= \{s \mid p\} : \texttt{type}$
  $union = \lambda^0 x, y.\{case\ x.\mathrm{g}\ of\ m \rightarrow m \mid f \rightarrow y.\mathrm{g}\}$

Constructions ajoutées

- Produit cartésien & :
  morph $= [\mathrm{g} : \textbf{genre}, \mathrm{n} : \textbf{nombre}] : \texttt{type}$
  $[\mathrm{g} = f, \mathrm{n} = s]$ est un terme de type **morph**

- Somme disjointe $\oplus$:
  genre $= \{m \mid f\} : \texttt{type}$ **nombre** $= \{s \mid p\} : \texttt{type}$
  $\mathit{union} = \lambda^0 x, y.\{\mathit{case}\ x.\mathrm{g}\ \mathit{of}\ m \to m \mid f \to y.\mathrm{g}\}$

- Produit dépendant Π :
  N, ***NP*** : (**morph**) `type`
  PAUL : ***NP*** $[\mathrm{g} = m, \mathrm{n} = s]$
  ET : $(\Pi x, y : \textbf{morph})\ \mathit{NP}\ x \multimap \textbf{\textit{NP}}\ y \multimap \textbf{\textit{NP}}\ [\mathrm{g} = \mathit{union}\ x\ y, \mathrm{n} = p]$

Calculs sur les types

ET $[\mathrm{g} = f, \mathrm{n} = s]$ $[\mathrm{g} = m, \mathrm{n} = s]$ MARIE PAUL

Calculs sur les types

ET $[\mathrm{g} = f, \mathrm{n} = s]$ $[\mathrm{g} = m, \mathrm{n} = s]$ MARIE PAUL

est de type

NP $[\mathrm{g} = \mathit{union}\ [\mathrm{g} = f, \mathrm{n} = s]\ [\mathrm{g} = m, \mathrm{n} = s], \mathrm{n} = p]$

Calculs sur les types

ET $[\mathrm{g} = f, \mathrm{n} = s]$ $[\mathrm{g} = m, \mathrm{n} = s]$ MARIE PAUL

est de type

NP $[\mathrm{g} = \mathit{union}\ [\mathrm{g} = f, \mathrm{n} = s]\ [\mathrm{g} = m, \mathrm{n} = s], \mathrm{n} = p]$

$=_\beta$ ***NP*** $[\mathrm{g} = m, \mathrm{n} = p]$

Mouvement explicite

Exemple : proposition relative

Marie aime Jean **S**

Mouvement explicite

Exemple : proposition relative

Marie aime $\quad \boldsymbol{NP \multimap S}$

Mouvement explicite

Exemple : proposition relative

Marie aime $\quad\quad$ $\boldsymbol{NP \multimap S}$

Jean que$_1$ *Marie aime* t_1

Mouvement explicite

Exemple : proposition relative

Marie aime	$\boldsymbol{NP} \multimap \boldsymbol{S}$
Jean que$_1$ *Marie aime* t_1 *dort*	$\boldsymbol{S}$

Mouvement explicite

Exemple : proposition relative

Marie aime $\quad\quad \boldsymbol{NP} \multimap \boldsymbol{S}$

Jean que$_1$ *Marie aime* t_1 *dort* $\quad\quad \boldsymbol{S}$

QUE : $(\boldsymbol{NP} \multimap \boldsymbol{S}) \multimap \boldsymbol{NP} \multimap \boldsymbol{NP}$

Mouvement explicite

Exemple : proposition relative

Marie aime $\quad \boldsymbol{NP} \multimap \boldsymbol{S}$

Jean que$_1$ *Marie aime* t_1 *dort* $\quad \boldsymbol{S}$

QUE : $(\boldsymbol{NP} \multimap \boldsymbol{S}) \multimap \boldsymbol{NP} \multimap \boldsymbol{NP}$

QUE $:=_{\text{Syn}} \lambda^0 pn.(n + /\text{que}/ + (p\ \epsilon))$

Surgénération

Exemple : les îlots d'extraction

Jean dit que Marie aime Paul

Surgénération

Exemple : les îlots d'extraction

Jean dit que Marie aime

Surgénération

Exemple : les îlots d'extraction

Paul que Jean dit que Marie aime

Surgénération

Exemple : les îlots d'extraction

Paul que Jean dit que Marie aime *marche*

Surgénération

Exemple : les îlots d'extraction

(1) *Paul que Jean dit que Marie aime marche*
MARCHE (QUE($\lambda^0 x$.DIT QUE (AIME x MARIE) JEAN) PAUL)

Surgénération

Exemple : les îlots d'extraction

(1) *Paul que Jean dit que Marie aime marche*
MARCHE (QUE($\lambda^0 x$.DIT QUE (AIME x MARIE) JEAN) PAUL)

Jean dit que Paul aime Marie

Surgénération

Exemple : les îlots d'extraction

(1) *Paul que Jean dit que Marie aime* *marche*
MARCHE (QUE($\lambda^0 x$.DIT QUE (AIME x MARIE) JEAN) PAUL)

Jean dit que *aime Marie*

Surgénération

Exemple : les îlots d'extraction

(1) *Paul que Jean dit que Marie aime* *marche*
MARCHE (QUE($\lambda^0 x$.DIT QUE (AIME x MARIE) JEAN) PAUL)

Paul qui Jean dit que *aime Marie*

Surgénération

Exemple : les îlots d'extraction

(1) *Paul que Jean dit que Marie aime* *marche*
MARCHE (QUE($\lambda^0 x$.DIT QUE (AIME x MARIE) JEAN) PAUL)

**Paul qui Jean dit que* *aime Marie marche*

Surgénération

Exemple : les îlots d'extraction

(1) *Paul que Jean dit que Marie aime marche*
$\text{MARCHE}\,(\text{QUE}(\lambda^0 x.\text{DIT QUE}\,(\text{AIME}\,x\,\text{MARIE})\,\text{JEAN})\,\text{PAUL})$

(2) **Paul qui Jean dit que aime Marie marche*
$^*\text{MARCHE}\,(\text{QUI}(\lambda^0 x.\text{DIT QUE}\,(\text{AIME MARIE}\,x)\,\text{JEAN})\,\text{PAUL})$

Surgénération

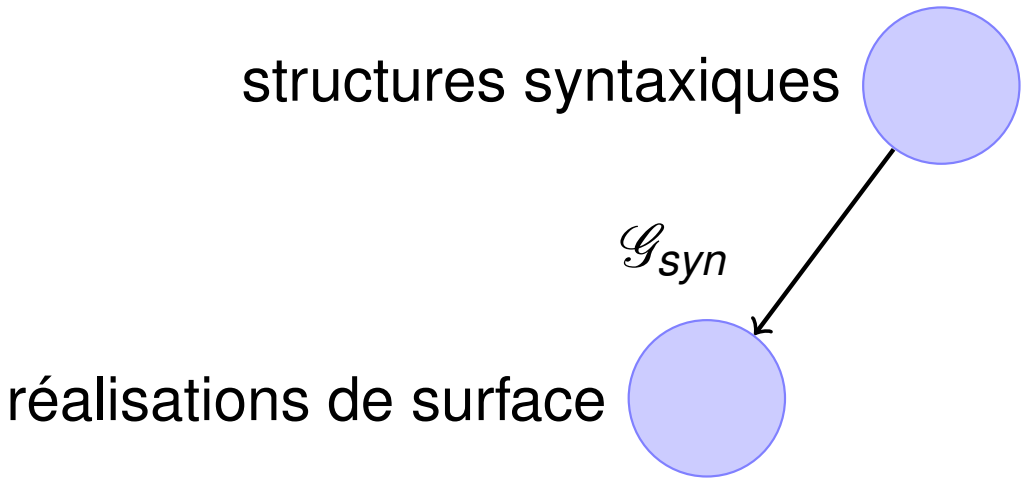
Exemple : les îlots d'extraction

(1) *Paul que Jean dit que Marie aime marche*
$\text{MARCHE}\,(\text{QUE}(\lambda^0 x.\text{DIT QUE}\,(\text{AIME}\,x\,\text{MARIE})\,\text{JEAN})\,\text{PAUL})$

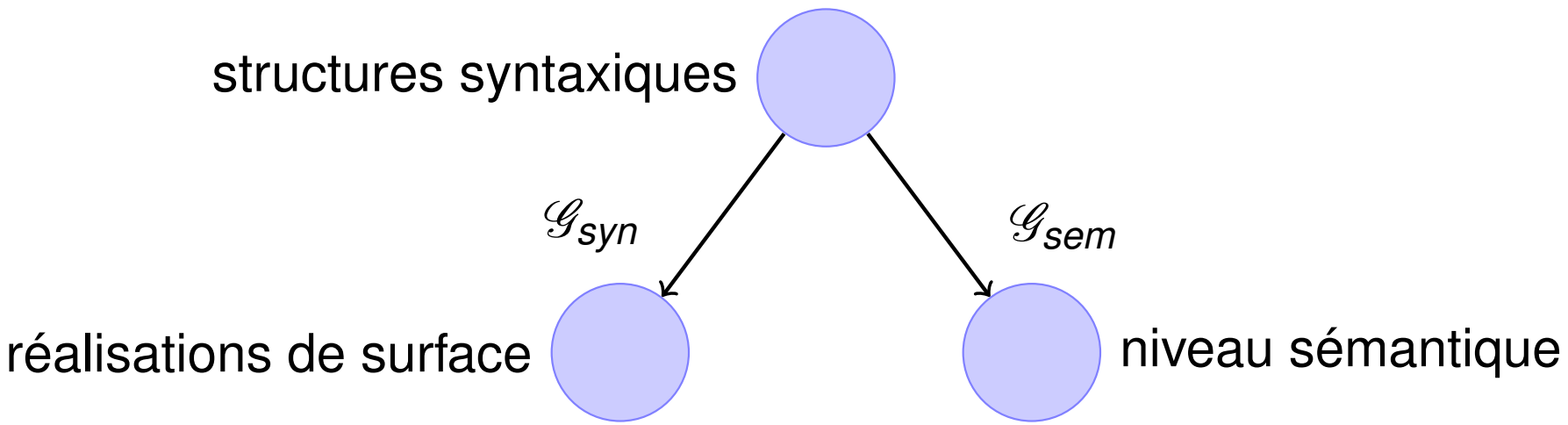
(2) **Paul qui Jean dit que aime Marie marche*
$^*\text{MARCHE}\,(\text{QUI}(\lambda^0 x.\text{DIT QUE}\,(\text{AIME MARIE}\,x)\,\text{JEAN})\,\text{PAUL})$

$\longrightarrow$ les dérivations doivent être contrôlées.

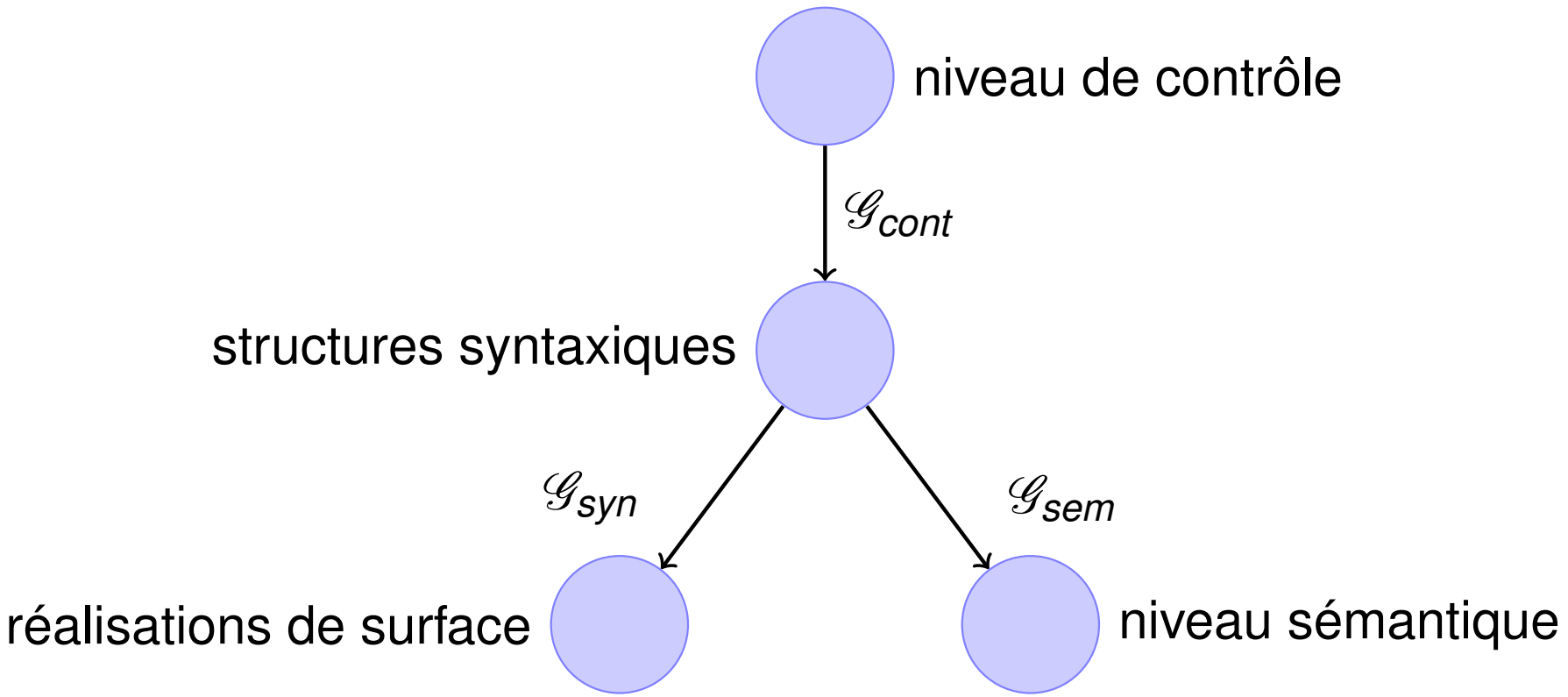
Architecture de contrôle



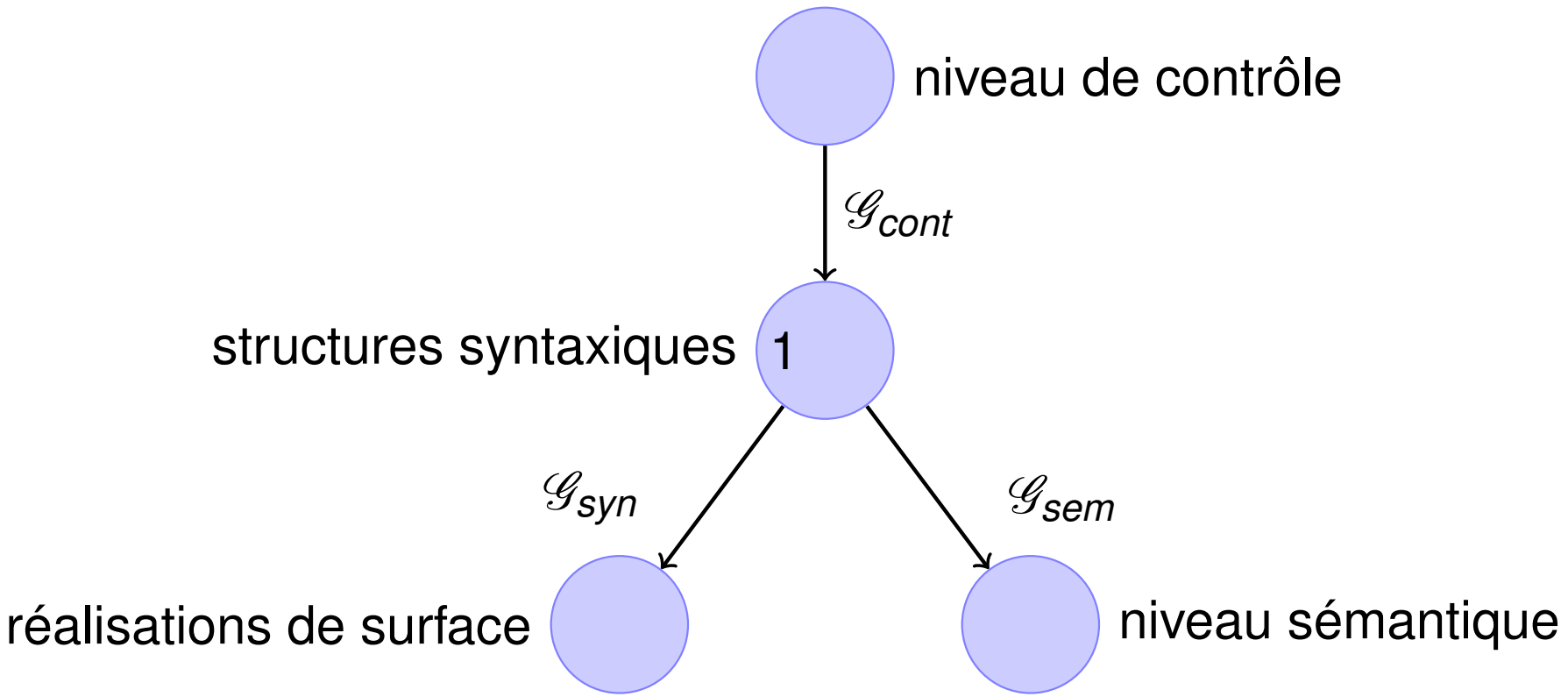
Architecture de contrôle



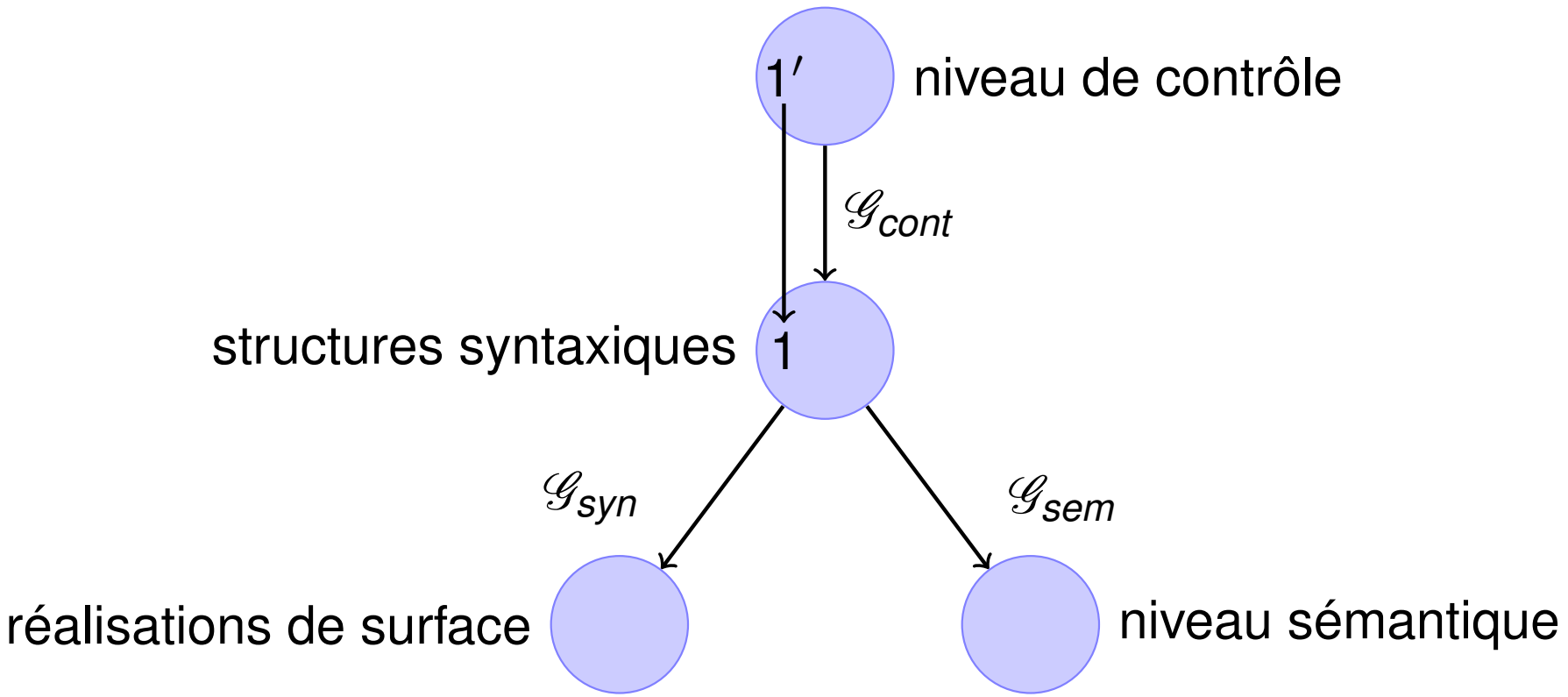
Architecture de contrôle



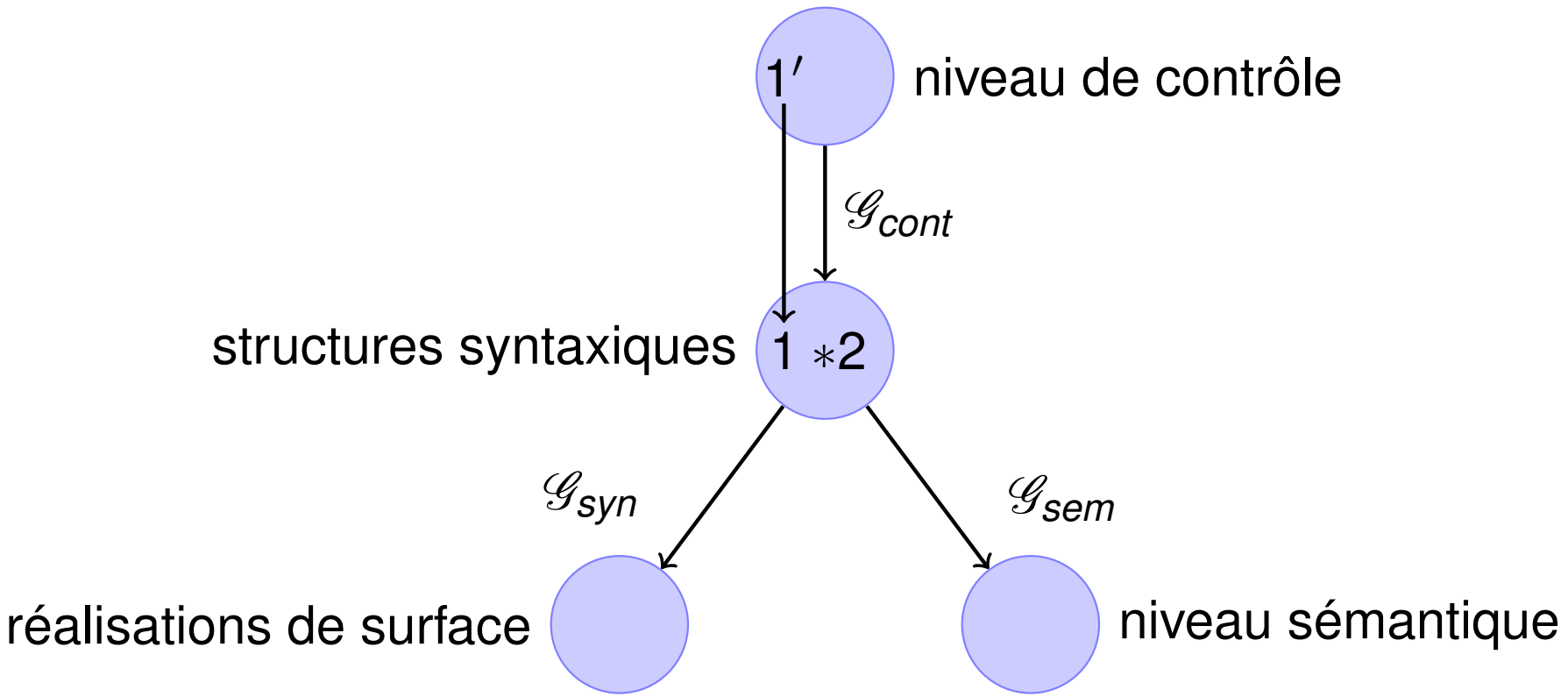
Architecture de contrôle



Architecture de contrôle



Architecture de contrôle



Signature de contrôle

$$
\begin{array}{ll}
\Sigma : & \\
\mathbf{ext} = \{0 \mid 1 \mid 2\} & : \mathtt{type} \\
\boldsymbol{S}, \boldsymbol{NP} & : (\mathbf{ext})\ \mathtt{type}
\end{array}
$$

Signature de contrôle

$$
\begin{array}{ll}
\Sigma : & \\
\mathbf{ext} = \{0 \mid 1 \mid 2\} & : \texttt{type} \\
\boldsymbol{S}, \boldsymbol{NP} & : (\mathbf{ext})\ \texttt{type} \\
\textsc{Marie}', \ldots & : \boldsymbol{NP}\ 0
\end{array}
$$

Signature de contrôle

$$
\begin{array}{ll}
\Sigma : & \\
\mathbf{ext} = \{0 \mid 1 \mid 2\} & : \texttt{type} \\
\boldsymbol{S}, \boldsymbol{NP} & : (\mathbf{ext})\ \texttt{type} \\
\textsc{Marie}', \ldots & : \boldsymbol{NP}\ 0 \\
\textsc{qui}' & : (\Pi x : \mathbf{ext})\ (\boldsymbol{NP}\ 1 \multimap \boldsymbol{S}\ 1) \multimap \boldsymbol{NP}\ x \multimap \boldsymbol{NP}\ x
\end{array}
$$

Signature de contrôle

$$
\begin{array}{ll}
\Sigma : & \\
\mathbf{ext} = \{0 \mid 1 \mid 2\} & : \texttt{type} \\
\boldsymbol{S}, \boldsymbol{NP} & : (\mathbf{ext})\ \texttt{type} \\
\textsc{Marie}', \ldots & : \boldsymbol{NP}\ 0 \\
\textsc{qui}' & : (\Pi x : \mathbf{ext})\ (\boldsymbol{NP}\ 1 \multimap \boldsymbol{S}\ 1) \multimap \boldsymbol{NP}\ x \multimap \boldsymbol{NP}\ x \\
\textsc{que}' & : (\Pi x : \mathbf{ext})\ (\boldsymbol{NP}\ 1 \multimap \boldsymbol{S}\ 1) \multimap \boldsymbol{NP}\ x \multimap \boldsymbol{NP}\ x
\end{array}
$$

Signature de contrôle

$$
\begin{array}{ll}
\Sigma : & \\
\mathbf{ext} = \{0 \mid 1 \mid 2\} & : \texttt{type} \\
\boldsymbol{S}, \boldsymbol{NP} & : (\mathbf{ext})\ \texttt{type} \\
\textsc{Marie}', \ldots & : \boldsymbol{NP}\ 0 \\
\textsc{qui}' & : (\Pi x : \mathbf{ext})\ (\boldsymbol{NP}\ 1 \multimap \boldsymbol{S}\ 1) \multimap \boldsymbol{NP}\ x \multimap \boldsymbol{NP}\ x \\
\textsc{que}' & : (\Pi x : \mathbf{ext})\ (\boldsymbol{NP}\ 1 \multimap \boldsymbol{S}\ 1) \multimap \boldsymbol{NP}\ x \multimap \boldsymbol{NP}\ x \\
\textsc{que}' & : (\Pi x : \mathbf{ext})\ (\boldsymbol{NP}\ 1 \multimap \boldsymbol{S}\ 2) \multimap \boldsymbol{NP}\ x \multimap \boldsymbol{NP}\ x
\end{array}
$$

Signature de contrôle

$$
\begin{array}{ll}
\Sigma : & \\
\mathbf{ext} = \{0 \mid 1 \mid 2\} & : \texttt{type} \\
\boldsymbol{S}, \boldsymbol{NP} & : (\mathbf{ext})\ \texttt{type} \\
\textsc{Marie}', \ldots & : \boldsymbol{NP}\ 0 \\
\textsc{qui}' & : (\Pi x : \mathbf{ext})\ (\boldsymbol{NP}\ 1 \multimap \boldsymbol{S}\ 1) \multimap \boldsymbol{NP}\ x \multimap \boldsymbol{NP}\ x \\
\textsc{que}' & : (\Pi x : \mathbf{ext})\ (\boldsymbol{NP}\ 1 \multimap \boldsymbol{S}\ 1) \multimap \boldsymbol{NP}\ x \multimap \boldsymbol{NP}\ x \\
\textsc{que}' & : (\Pi x : \mathbf{ext})\ (\boldsymbol{NP}\ 1 \multimap \boldsymbol{S}\ 2) \multimap \boldsymbol{NP}\ x \multimap \boldsymbol{NP}\ x \\
\textsc{marche}' & : (\Pi x : \mathbf{ext})\ \boldsymbol{NP}\ x \multimap \boldsymbol{S}\ x \\
\textsc{aime}' & : (\Pi x, y : \mathbf{ext})\ (\boldsymbol{NP}\ x \multimap \boldsymbol{NP}\ y \multimap \boldsymbol{S}\ (max\ x\ y))
\end{array}
$$

Signature de contrôle

$$
\begin{array}{ll}
\Sigma : & \\
\mathbf{ext} = \{0 \mid 1 \mid 2\} & : \texttt{type} \\
\boldsymbol{S}, \boldsymbol{NP} & : (\mathbf{ext})\ \texttt{type} \\
\textsc{Marie}', \ldots & : \boldsymbol{NP}\ 0 \\
\textsc{qui}' & : (\Pi x : \mathbf{ext})\ (\boldsymbol{NP}\ 1 \multimap \boldsymbol{S}\ 1) \multimap \boldsymbol{NP}\ x \multimap \boldsymbol{NP}\ x \\
\textsc{que}' & : (\Pi x : \mathbf{ext})\ (\boldsymbol{NP}\ 1 \multimap \boldsymbol{S}\ 1) \multimap \boldsymbol{NP}\ x \multimap \boldsymbol{NP}\ x \\
\textsc{que}' & : (\Pi x : \mathbf{ext})\ (\boldsymbol{NP}\ 1 \multimap \boldsymbol{S}\ 2) \multimap \boldsymbol{NP}\ x \multimap \boldsymbol{NP}\ x \\
\textsc{marche}' & : (\Pi x : \mathbf{ext})\ \boldsymbol{NP}\ x \multimap \boldsymbol{S}\ x \\
\textsc{aime}' & : (\Pi x, y : \mathbf{ext})\ (\boldsymbol{NP}\ x \multimap \boldsymbol{NP}\ y \multimap \boldsymbol{S}\ (max\ x\ y)) \\
\textsc{dit que}' & : (\Pi x, y : \mathbf{ext})\ (\boldsymbol{S}\ x \multimap \boldsymbol{NP}\ y \multimap \boldsymbol{S}\ (max\ (s\ x)\ y))
\end{array}
$$

où $s : \left\{ \right.$

Signature de contrôle

$$
\begin{array}{ll}
\Sigma : & \\
\mathbf{ext} = \{0 \mid 1 \mid 2\} & : \texttt{type} \\
\boldsymbol{S}, \boldsymbol{NP} & : (\mathbf{ext})\ \texttt{type} \\
\textsc{Marie}', \ldots & : \boldsymbol{NP}\ 0 \\
\textsc{qui}' & : (\Pi x : \mathbf{ext})\ (\boldsymbol{NP}\ 1 \multimap \boldsymbol{S}\ 1) \multimap \boldsymbol{NP}\ x \multimap \boldsymbol{NP}\ x \\
\textsc{que}' & : (\Pi x : \mathbf{ext})\ (\boldsymbol{NP}\ 1 \multimap \boldsymbol{S}\ 1) \multimap \boldsymbol{NP}\ x \multimap \boldsymbol{NP}\ x \\
\textsc{que}' & : (\Pi x : \mathbf{ext})\ (\boldsymbol{NP}\ 1 \multimap \boldsymbol{S}\ 2) \multimap \boldsymbol{NP}\ x \multimap \boldsymbol{NP}\ x \\
\textsc{marche}' & : (\Pi x : \mathbf{ext})\ \boldsymbol{NP}\ x \multimap \boldsymbol{S}\ x \\
\textsc{aime}' & : (\Pi x, y : \mathbf{ext})\ (\boldsymbol{NP}\ x \multimap \boldsymbol{NP}\ y \multimap \boldsymbol{S}\ (max\ x\ y)) \\
\textsc{dit que}' & : (\Pi x, y : \mathbf{ext})\ (\boldsymbol{S}\ x \multimap \boldsymbol{NP}\ y \multimap \boldsymbol{S}\ (max\ (s\ x)\ y))
\end{array}
$$

où $s : \left\{ \begin{array}{l} 0 \longrightarrow 0 \\ \\ \end{array} \right.$

Signature de contrôle

$$
\begin{array}{ll}
\Sigma : & \\
\mathbf{ext} = \{0 \mid 1 \mid 2\} & : \texttt{type} \\
\boldsymbol{S}, \boldsymbol{NP} & : (\mathbf{ext})\ \texttt{type} \\
\textsc{Marie}', \ldots & : \boldsymbol{NP}\ 0 \\
\textsc{qui}' & : (\Pi x : \mathbf{ext})\ (\boldsymbol{NP}\ 1 \multimap \boldsymbol{S}\ 1) \multimap \boldsymbol{NP}\ x \multimap \boldsymbol{NP}\ x \\
\textsc{que}' & : (\Pi x : \mathbf{ext})\ (\boldsymbol{NP}\ 1 \multimap \boldsymbol{S}\ 1) \multimap \boldsymbol{NP}\ x \multimap \boldsymbol{NP}\ x \\
\textsc{que}' & : (\Pi x : \mathbf{ext})\ (\boldsymbol{NP}\ 1 \multimap \boldsymbol{S}\ 2) \multimap \boldsymbol{NP}\ x \multimap \boldsymbol{NP}\ x \\
\textsc{marche}' & : (\Pi x : \mathbf{ext})\ \boldsymbol{NP}\ x \multimap \boldsymbol{S}\ x \\
\textsc{aime}' & : (\Pi x, y : \mathbf{ext})\ (\boldsymbol{NP}\ x \multimap \boldsymbol{NP}\ y \multimap \boldsymbol{S}\ (max\ x\ y)) \\
\textsc{dit que}' & : (\Pi x, y : \mathbf{ext})\ (\boldsymbol{S}\ x \multimap \boldsymbol{NP}\ y \multimap \boldsymbol{S}\ (max\ (s\ x)\ y))
\end{array}
$$

où $s : \begin{cases} 0 \longrightarrow 0 \\ 1 \longrightarrow 2 \end{cases}$

Signature de contrôle

$$
\begin{array}{ll}
\Sigma : & \\
\mathbf{ext} = \{0 \mid 1 \mid 2\} & : \texttt{type} \\
\boldsymbol{S}, \boldsymbol{NP} & : (\mathbf{ext})\ \texttt{type} \\
\textsc{Marie}', \ldots & : \boldsymbol{NP}\ 0 \\
\textsc{qui}' & : (\Pi x : \mathbf{ext})\ (\boldsymbol{NP}\ 1 \multimap \boldsymbol{S}\ 1) \multimap \boldsymbol{NP}\ x \multimap \boldsymbol{NP}\ x \\
\textsc{que}' & : (\Pi x : \mathbf{ext})\ (\boldsymbol{NP}\ 1 \multimap \boldsymbol{S}\ 1) \multimap \boldsymbol{NP}\ x \multimap \boldsymbol{NP}\ x \\
\textsc{que}' & : (\Pi x : \mathbf{ext})\ (\boldsymbol{NP}\ 1 \multimap \boldsymbol{S}\ 2) \multimap \boldsymbol{NP}\ x \multimap \boldsymbol{NP}\ x \\
\textsc{marche}' & : (\Pi x : \mathbf{ext})\ \boldsymbol{NP}\ x \multimap \boldsymbol{S}\ x \\
\textsc{aime}' & : (\Pi x, y : \mathbf{ext})\ (\boldsymbol{NP}\ x \multimap \boldsymbol{NP}\ y \multimap \boldsymbol{S}\ (max\ x\ y)) \\
\textsc{dit que}' & : (\Pi x, y : \mathbf{ext})\ (\boldsymbol{S}\ x \multimap \boldsymbol{NP}\ y \multimap \boldsymbol{S}\ (max\ (s\ x)\ y))
\end{array}
$$

$$
\text{où } s : \begin{cases} 0 \longrightarrow 0 \\ 1 \longrightarrow 2 \\ 2 \longrightarrow 2 \end{cases}
$$

Lexique (effacement)

$$
\begin{array}{ll}
\boldsymbol{S} & :=_{Cont} \lambda x.\, \boldsymbol{S} \\
\boldsymbol{NP} & :=_{Cont} \lambda x.\, \boldsymbol{NP} \\
\textsc{aime}' & :=_{Cont} \lambda xy.\textsc{aime} \\
\textsc{Marie}' & :=_{Cont} \textsc{Marie} \\
\mathrm{T}' & :=_{Cont} \lambda x.\mathrm{T}
\end{array}
$$

Lexique (effacement)

$$
\begin{array}{ll}
\boldsymbol{S} & :=_{Cont} \lambda x.\, \boldsymbol{S} \\
\boldsymbol{NP} & :=_{Cont} \lambda x.\, \boldsymbol{NP} \\
\textsc{aime}' & :=_{Cont} \lambda xy.\textsc{aime} \\
\textsc{Marie}' & :=_{Cont} \textsc{Marie} \\
\mathrm{T}' & :=_{Cont} \lambda x.\mathrm{T}
\end{array}
$$

$\textsc{aime}'\ 0\ 0\ \textsc{Marie}'\ \textsc{Jean}'$

Lexique (effacement)

$\boldsymbol{S}$	$:=_{Cont}\ \lambda x.\,\boldsymbol{S}$
$\boldsymbol{NP}$	$:=_{Cont}\ \lambda x.\,\boldsymbol{NP}$
$\textsc{aime}'$	$:=_{Cont}\ \lambda xy.\textsc{aime}$
$\textsc{Marie}'$	$:=_{Cont}\ \textsc{Marie}$
T'	$:=_{Cont}\ \lambda x.\mathrm{T}$

$\textsc{aime}'\ 0\ 0\ \textsc{Marie}'\ \textsc{Jean}'$
$:=_{Cont}\ (\lambda xy.\textsc{aime})\ 0\ 0\ \textsc{Marie}\ \textsc{Jean}$

Lexique (effacement)

$\boldsymbol{S}$	$:=_{Cont}\ \lambda x.\,\boldsymbol{S}$
$\boldsymbol{NP}$	$:=_{Cont}\ \lambda x.\,\boldsymbol{NP}$
$\textsc{aime}'$	$:=_{Cont}\ \lambda xy.\textsc{aime}$
$\textsc{Marie}'$	$:=_{Cont}\ \textsc{Marie}$
T'	$:=_{Cont}\ \lambda x.\mathrm{T}$

$\textsc{aime}'\ 0\ 0\ \textsc{Marie}'\ \textsc{Jean}'$
$:=_{Cont}\ (\lambda xy.\textsc{aime})\ 0\ 0\ \textsc{Marie}\ \textsc{Jean}$
$=_{\beta}\ \textsc{aime}\ \textsc{Marie}\ \textsc{Jean}$

Langage accepté

(1) Paul que$_1$ Jean dit que Marie aime t_1 marche

Langage accepté

$$\text{MARCHE}\,(\text{QUE}(\lambda^0 x.\text{DIT QUE}\,(\text{AIME}\,x\,\text{MARIE})\,\text{JEAN})\,\text{PAUL})$$

$$\downarrow \mathscr{G}_{syn}$$

(1) Paul que$_1$ Jean dit que Marie aime t_1 marche

Langage accepté

$$\text{MARCHE}'\,0\,(\text{QUE}'\,0\,(\lambda^0 x.\,\text{DIT QUE}'\,1\,0(\text{AIME}'\,1\,0 \quad x \quad \text{MARIE}')\,\text{JEAN}')\,\text{PAUL}')$$

$$\downarrow \mathscr{G}_{cont}$$

$$\text{MARCHE}\,(\text{QUE}(\lambda^0 x.\text{DIT QUE}\,(\text{AIME}\,x\,\text{MARIE})\,\text{JEAN})\,\text{PAUL})$$

$$\downarrow \mathscr{G}_{syn}$$

(1) Paul que$_1$ Jean dit que Marie aime t_1 marche

Langage accepté

$$\text{QUE'} : (\Pi x : \mathit{ext})\ (\boldsymbol{NP}\ 1 \multimap \boldsymbol{S}\ 2) \multimap \boldsymbol{NP}\ x \multimap \boldsymbol{NP}\ x$$

$$\overbrace{\text{MARCHE}'\ 0\ (\text{QUE}'\ 0\ (\lambda^0 x.\ \underbrace{\text{DIT QUE}'\ 1\ 0\ (\text{AIME}'\ 1\ 0\ \overbrace{x}^{\boldsymbol{NP}\ 1}\ \text{MARIE}')\ \text{JEAN}'}_{\boldsymbol{S}\ 2})\ \text{PAUL}')}^{\boldsymbol{S}\ 0}$$

$$\downarrow \mathscr{G}_{cont}$$

$$\text{MARCHE}\ (\text{QUE}(\lambda^0 x.\text{DIT QUE}\ (\text{AIME}\ x\ \text{MARIE})\ \text{JEAN})\ \text{PAUL})$$

$$\downarrow \mathscr{G}_{syn}$$

(1) Paul que$_1$ Jean dit que Marie aime t_1 marche

Exemple de terme éliminé

(2) *Paul qui$_1$ Jean dit que t_1 aime Marie marche

Exemple de terme éliminé

$$*\text{MARCHE}\,(\text{QUI}(\lambda^0 x.\text{DIT QUE}\,(\text{AIME MARIE}\,x)\,\text{JEAN})\,\text{PAUL})$$

$$\downarrow \mathscr{G}_{syn}$$

(2) *Paul qui$_1$ Jean dit que t_1 aime Marie marche

Exemple de terme éliminé

$$\textsc{marche}'\,0\,(\textsc{qui}'\,0\,(\lambda^0 x.\,\textsc{dit que}'\,1\,0\,(\textsc{aime}'\,0\,1\,\textsc{Marie}'\quad x\quad)\,\textsc{Jean}')\,\textsc{Paul}')$$

$$\downarrow \mathscr{G}_{cont}$$

$$*\textsc{marche}\,(\textsc{qui}(\lambda^0 x.\textsc{dit que}\,(\textsc{aime}\,\textsc{Marie}\,x)\,\textsc{Jean})\,\textsc{paul})$$

$$\downarrow \mathscr{G}_{syn}$$

(2) *Paul qui$_1$ Jean dit que t_1 aime Marie marche

Exemple de terme éliminé

$$\textsc{qui'} : (\Pi x : \mathit{ext})\ (\boldsymbol{NP}\ 1 \multimap \boldsymbol{S}\ 1) \multimap \boldsymbol{NP}\ x \multimap \boldsymbol{NP}\ x$$

$$\overbrace{\textsc{marche}'\ 0\ (\textsc{qui}'\ 0\ (\lambda^0 x.\ \underbrace{\textsc{dit que}'\ 1\ 0\ (\textsc{aime}'\ 0\ 1\ \textsc{Marie}'\ \overbrace{x}^{\boldsymbol{NP}\ 1}\ )\ \textsc{Jean}'}_{\boldsymbol{S}\ 2})\ \textsc{Paul}')}^{\textbf{non typable}}$$

$$\downarrow \mathscr{G}_{cont}$$

$$*\textsc{marche}\ (\textsc{qui}(\lambda^0 x.\textsc{dit que}\ (\textsc{aime}\ \textsc{Marie}\ x)\ \textsc{Jean})\ \textsc{paul})$$

$$\downarrow \mathscr{G}_{syn}$$

(2) *Paul qui$_1$ Jean dit que t_1 aime Marie marche

Phénomènes modélisés

- Îlots de mouvements explicites :
  **Jean que$_1$ Paul qui$_2$ t_2 aime t_1 dort marche*

Phénomènes modélisés

- Îlots de mouvements explicites :
  **Jean que$_1$ Paul qui$_2$ t_2 aime t_1 dort marche*
- Contraintes de cas :
  **Paul que aime Marie dort*

Phénomènes modélisés

- Îlots de mouvements explicites :
  **Jean que$_1$ Paul qui$_2$ t_2 aime t_1 dort marche*
- Contraintes de cas :
  **Paul que aime Marie dort*
- Îlots pour la prise de portée de quantificateurs :
  Quelqu'un dit que tout le monde aime Marie
  $^*\forall y.\exists x.\textbf{dit que}\ x\ (\textbf{aime}\ y\ \textbf{m})$

Phénomènes modélisés

- Îlots de mouvements explicites :
  **Jean que$_1$ Paul qui$_2$ t_2 aime t_1 dort marche*
- Contraintes de cas :
  **Paul que aime Marie dort*
- Îlots pour la prise de portée de quantificateurs :
  Quelqu'un dit que tout le monde aime Marie
  $^*\forall y.\exists x.$**dit que** x (**aime** y **m**)
- Contraintes d'imbrication des interrogatives multiples :
  Which$_1$ problems does John know whom$_2$ to talk to t_2 about t_1 ?
  **Whom$_1$ does John know which$_2$ problems to talk to t_1 about t_2 ?*

Phénomènes modélisés

- Îlots de mouvements explicites :
  **Jean que$_1$ Paul qui$_2$ t_2 aime t_1 dort marche*
- Contraintes de cas :
  **Paul que aime Marie dort*
- Îlots pour la prise de portée de quantificateurs :
  Quelqu'un dit que tout le monde aime Marie
  $^*\forall y.\exists x.$**dit que** x (**aime** y **m**)
- Contraintes d'imbrication des interrogatives multiples :
  Which$_1$ problems does John know whom$_2$ to talk to t_2 about t_1 ?
  **Whom$_1$ does John know which$_2$ problems to talk to t_1 about t_2 ?*

+ modélisations en serbo-croate (Mihalicek, 2012)

Problématique

$t : \boldsymbol{S}$

Problématique

$$t : \boldsymbol{S} \;:=\; \mathscr{L}(t) : \mathscr{L}(\boldsymbol{S})$$

Problématique

$$t : \boldsymbol{S} := \mathscr{L}(t) : \mathscr{L}(\boldsymbol{S}) \longrightarrow_{\beta}^{*} \mathit{FN}(\mathscr{L}(t)) : \mathscr{L}(\boldsymbol{S})$$

Problématique

$$t : \boldsymbol{S} := \mathscr{L}(t) : \mathscr{L}(\boldsymbol{S}) \longrightarrow_{\beta}^{*} \mathit{FN}(\mathscr{L}(t)) : \mathscr{L}(\boldsymbol{S})$$

- Normalisation

Problématique

$$t : \boldsymbol{S} := \mathscr{L}(t) : \mathscr{L}(\boldsymbol{S}) \longrightarrow_{\beta}^{*} \mathit{FN}(\mathscr{L}(t)) : \mathscr{L}(\boldsymbol{S})$$

- Normalisation
  - forte : toutes les chaînes de réduction sont finies

Problématique

$$t : \boldsymbol{S} := \mathscr{L}(t) : \mathscr{L}(\boldsymbol{S}) \longrightarrow_{\beta}^{*} \mathit{FN}(\mathscr{L}(t)) : \mathscr{L}(\boldsymbol{S})$$

- Normalisation
  - forte : toutes les chaînes de réduction sont finies
  - faible : une chaîne de réduction produit une forme normale

Problématique

$$t : \boldsymbol{S} := \mathscr{L}(t) : \mathscr{L}(\boldsymbol{S}) \longrightarrow_{\beta}^{*} \mathit{FN}(\mathscr{L}(t)) : \mathscr{L}(\boldsymbol{S})$$

- Normalisation
  - forte : toutes les chaînes de réduction sont finies
  - faible : une chaîne de réduction produit une forme normale
- Confluence : 2 façons de réduire un terme convergent vers un résultat commun

Problématique

$$t : \boldsymbol{S} := \mathscr{L}(t) : \mathscr{L}(\boldsymbol{S}) \longrightarrow_{\beta}^{*} \mathit{FN}(\mathscr{L}(t)) : \mathscr{L}(\boldsymbol{S})$$

- Normalisation
  - forte : toutes les chaînes de réduction sont finies
  - faible : une chaîne de réduction produit une forme normale
- Confluence : 2 façons de réduire un terme convergent vers un résultat commun
- Réduction du sujet : le typage est conservé par la réduction.

Problématique

$$t : \boldsymbol{S} := \mathscr{L}(t) : \mathscr{L}(\boldsymbol{S}) \longrightarrow_{\beta}^{*} \mathit{FN}(\mathscr{L}(t)) : \mathscr{L}(\boldsymbol{S})$$

- Normalisation
  - forte : toutes les chaînes de réduction sont finies
  - faible : une chaîne de réduction produit une forme normale
- Confluence : 2 façons de réduire un terme convergent vers un résultat commun
- Réduction du sujet : le typage est conservé par la réduction.

Produit dépendant : vrai (Cervesato et Pfenning, 1996)

Problématique

$$t : \boldsymbol{S} := \mathscr{L}(t) : \mathscr{L}(\boldsymbol{S}) \longrightarrow_{\beta}^{*} \mathit{FN}(\mathscr{L}(t)) : \mathscr{L}(\boldsymbol{S})$$

- Normalisation
  - forte : toutes les chaînes de réduction sont finies
  - faible : une chaîne de réduction produit une forme normale
- Confluence : 2 façons de réduire un terme convergent vers un résultat commun
- Réduction du sujet : le typage est conservé par la réduction.

Produit dépendant : vrai (Cervesato et Pfenning, 1996)
Somme disjointe : vrai (Lindley, 2007)

Problématique

$$t : \boldsymbol{S} := \mathscr{L}(t) : \mathscr{L}(\boldsymbol{S}) \longrightarrow_{\beta}^{*} \mathit{FN}(\mathscr{L}(t)) : \mathscr{L}(\boldsymbol{S})$$

- Normalisation
  - forte : toutes les chaînes de réduction sont finies
  - faible : une chaîne de réduction produit une forme normale
- Confluence : 2 façons de réduire un terme convergent vers un résultat commun
- Réduction du sujet : le typage est conservé par la réduction.

Produit dépendant : vrai (Cervesato et Pfenning, 1996)
Somme disjointe : vrai (Lindley, 2007)
Les 2 ensemble : ?

Réduction $\longrightarrow_{\beta\gamma}$

- $\longrightarrow_{\beta}$:
  $(\lambda x.t)\ u \longrightarrow_{\beta} t[x := u]$

Réduction $\longrightarrow_{\beta\gamma}$

- $\longrightarrow_{\beta}$:

$$(\lambda x.t)\ u \longrightarrow_{\beta} t[x := u]$$

$$[l_i = t_i]_{i=1}^{n}.l_j \longrightarrow_{\beta} t_j$$

$$\{case\ c_j\ k\ \text{of}\ (c_i\ x_i) \rightarrow t_i\}_{i=1}^{n} \longrightarrow_{\beta} t_j[x_j := k]$$

Réduction $\longrightarrow_{\beta\gamma}$

- $\longrightarrow_{\beta}$:
  $(\lambda x.t)\ u \longrightarrow_{\beta} t[x := u]$

  $[l_i = t_i]_{i=1}^{n}.l_j \longrightarrow_{\beta} t_j$

  $\{case\ c_j\ k\ \text{of}\ (c_i\ x_i) \rightarrow t_i\}_{i=1}^{n} \longrightarrow_{\beta} t_j[x_j := k]$

- $\longrightarrow_{\gamma}$:
  $\{case\ u\ \text{of}\ (c_i\ x_i) \rightarrow \lambda y_i.t_i\}_{i=1}^{n}\ v$

Réduction $\longrightarrow_{\beta\gamma}$

- $\longrightarrow_{\beta}$:
  $(\lambda x.t)\ u \longrightarrow_{\beta} t[x := u]$

  $[l_i = t_i]_{i=1}^{n}.l_j \longrightarrow_{\beta} t_j$

  $\{case\ c_j\ k\ \text{of}\ (c_i\ x_i) \to t_i\}_{i=1}^{n} \longrightarrow_{\beta} t_j[x_j := k]$

- $\longrightarrow_{\gamma}$:
  $\{case\ u\ \text{of}\ (c_i\ x_i) \to \lambda y_i.t_i\}_{i=1}^{n}\ v$
  $\longrightarrow_{\gamma}\ \{case\ u\ \text{of}\ (c_i\ x_i) \to (\lambda y_i.t_i)\ v\}_{i=1}^{n}$

Réduction $\longrightarrow_{\beta\gamma}$

- $\longrightarrow_{\beta}$:
  $(\lambda x.t)\ u \longrightarrow_{\beta} t[x := u]$

  $[l_i = t_i]_{i=1}^{n}.l_j \longrightarrow_{\beta} t_j$

  $\{case\ c_j\ k\ \text{of}\ (c_i\ x_i) \to t_i\}_{i=1}^{n} \longrightarrow_{\beta} t_j[x_j := k]$

- $\longrightarrow_{\gamma}$:
  $\{case\ u\ \text{of}\ (c_i\ x_i) \to \lambda y_i.t_i\}_{i=1}^{n}\ v$
  $\longrightarrow_{\gamma}\ \{case\ u\ \text{of}\ (c_i\ x_i) \to (\lambda y_i.t_i)\ v\}_{i=1}^{n}$
  $\longrightarrow_{\beta}\ \{case\ u\ \text{of}\ (c_i\ x_i) \to t_i[y_i := v]\}_{i=1}^{n}$.

Normalisation de $\longrightarrow_{\beta\gamma}$

Principe de la preuve :
Le calcul est traduit vers un calcul plus simple sans produit dépendant, pour lequel la normalisation est connue.

Normalisation de $\longrightarrow_{\beta\gamma}$

Principe de la preuve :
Le calcul est traduit vers un calcul plus simple sans produit dépendant, pour lequel la normalisation est connue.

$|(\Pi x : \alpha)\,\beta| ::= x_\alpha\ |\alpha|\ (\lambda x.|\beta|)$

Normalisation de $\longrightarrow_{\beta\gamma}$

Principe de la preuve :
Le calcul est traduit vers un calcul plus simple sans produit dépendant, pour lequel la normalisation est connue.

$|(\Pi x : \alpha)\,\beta| ::= x_\alpha\ |\alpha|\ (\lambda x.|\beta|)$

La traduction doit :

- conserver le fait d'être bien typé

Normalisation de $\longrightarrow_{\beta\gamma}$

Principe de la preuve :
Le calcul est traduit vers un calcul plus simple sans produit dépendant, pour lequel la normalisation est connue.

$|(\Pi x : \alpha)\,\beta| ::= x_\alpha\ |\alpha|\ (\lambda x.|\beta|)$

La traduction doit :

- conserver le fait d'être bien typé
- conserver les étapes de réduction

Confluence de $\longrightarrow_{\beta\gamma}$

Preuve standard par établissement

- de la confluence de $\longrightarrow_{\beta}$
- de la confluence de $\longrightarrow_{\gamma}$
- de la commutativité entre $\longrightarrow_{\beta}$ et $\longrightarrow_{\gamma}$

Confluence de $\longrightarrow_{\beta\gamma}$

Preuve standard par établissement

- de la confluence de $\longrightarrow_{\beta}$
- de la confluence de $\longrightarrow_{\gamma}$
- de la commutativité entre $\longrightarrow_{\beta}$ et $\longrightarrow_{\gamma}$

Corollaire

Tout terme typable a une unique forme normale pour $\longrightarrow_{\beta\gamma}$
L'équivalence de 2 termes typables est donc décidable

η-conversion

$$f =_\eta \lambda x.f\ x \quad \text{car} \quad (\lambda x.f\ x)\ u \longrightarrow_\beta f\ u$$

η-conversion

$f =_\eta \lambda x.f\ x$ car $(\lambda x.f\ x)\ u \longrightarrow_\beta f\ u$
(égalité extensionnelle)

η-conversion

$f =_\eta \lambda x.f\ x$ car $(\lambda x.f\ x)\ u \longrightarrow_\beta f\ u$
(égalité extensionnelle)

Ambiguités dans les ACG :
$f : \alpha \multimap \beta$ et $\mathscr{L}(f) = \lambda^0 y.t \quad \Rightarrow \quad \mathscr{L}(\lambda^0 x.f\ x) =_\beta \mathscr{L}(f)$

η-conversion

$f =_\eta \lambda x.f\ x$ car $(\lambda x.f\ x)\ u \longrightarrow_\beta f\ u$
(égalité extensionnelle)

Ambiguités dans les ACG :
$f : \alpha \multimap \beta$ et $\mathscr{L}(f) = \lambda^0 y.t \quad \Rightarrow \quad \mathscr{L}(\lambda^0 x.f\ x) =_\beta \mathscr{L}(f)$

$t =_\eta \lambda x.t\ x$
$t =_\eta [l_i = t.l_i]_{i=1}^n$
$t =_\eta \{case\ t\ \text{of}\ (c_i\ x_i) \rightarrow (c_i\ x_i)\}_{i=1}^n$

η-conversion

$f =_\eta \lambda x.f\ x$ car $(\lambda x.f\ x)\ u \longrightarrow_\beta f\ u$
(égalité extensionnelle)

Ambiguités dans les ACG :
$f : \alpha \multimap \beta$ et $\mathscr{L}(f) = \lambda^0 y.t \quad \Rightarrow \quad \mathscr{L}(\lambda^0 x.f\ x) =_\beta \mathscr{L}(f)$

$t =_\eta \lambda x.t\ x$
$t =_\eta [l_i = t.l_i]_{i=1}^n$
$t =_\eta \{case\ t\ \text{of}\ (c_i\ x_i) \to (c_i\ x_i)\}_{i=1}^n$

Orientation possible en réduction ou en expansion

Contraintes sur l'orientation

- Réduction $\lambda x.f\ x \longrightarrow_{\eta} f$:

$$\lambda x.\{\mathit{case}\ t\ \text{of}\ (c_i\ x_i) \rightarrow c_i\ x_i\}_{i=1}^{n}$$

Contraintes sur l'orientation

- Réduction $\lambda x.f\ x \longrightarrow_\eta f$:

$$\lambda x.\{\mathit{case}\ t\ \text{of}\ (c_i\ x_i) \to c_i\ x_i\}_{i=1}^{n}$$
$$\swarrow_\eta$$
$$\lambda x.t$$

Contraintes sur l'orientation

- Réduction $\lambda x.f\ x \longrightarrow_\eta f$:

$$\lambda x.\{\mathit{case}\ t\ \text{of}\ (c_i\ x_i) \to c_i\ x_i\}_{i=1}^n$$

$$\swarrow_\eta \qquad \searrow_\gamma$$

$$\lambda x.t \qquad \{\mathit{case}\ t\ \text{of}\ (c_i\ x_i) \to \lambda x.c_i\ x_i\}_{i=1}^n$$

Contraintes sur l'orientation

- Réduction $\lambda x.f\ x \longrightarrow_\eta f$:

$$\lambda x.\{\mathit{case}\ t\ \text{of}\ (c_i\ x_i) \to c_i\ x_i\}_{i=1}^n$$

$$\swarrow_\eta \qquad \searrow_\gamma$$

$$\lambda x.t \qquad \{\mathit{case}\ t\ \text{of}\ (c_i\ x_i) \to \lambda x.c_i\ x_i\}_{i=1}^n$$

$$\lambda x.(\{\mathit{case}\ t\ \text{of}\ (c_i\ x_i) \to t_i\}_{i=1}^n\ x)$$

Contraintes sur l'orientation

- Réduction $\lambda x.f\ x \longrightarrow_\eta f$:

$$\lambda x.\{\mathit{case}\ t\ \text{of}\ (c_i\ x_i) \to c_i\ x_i\}_{i=1}^n$$

$$\swarrow_\eta \qquad \searrow_\gamma$$

$$\lambda x.t \qquad \{\mathit{case}\ t\ \text{of}\ (c_i\ x_i) \to \lambda x.c_i\ x_i\}_{i=1}^n$$

$$\lambda x.(\{\mathit{case}\ t\ \text{of}\ (c_i\ x_i) \to t_i\}_{i=1}^n\ x)$$

$$\swarrow_\eta$$

$$\{\mathit{case}\ t\ \text{of}\ (c_i\ x_i) \to t_i\}_{i=1}^n$$

Contraintes sur l'orientation

- Réduction $\lambda x.f\ x \longrightarrow_\eta f$:

$$\lambda x.\{\mathit{case}\ t\ \text{of}\ (c_i\ x_i) \to c_i\ x_i\}_{i=1}^n$$

$$\swarrow_\eta \qquad \searrow_\gamma$$

$$\lambda x.t \qquad \{\mathit{case}\ t\ \text{of}\ (c_i\ x_i) \to \lambda x.c_i\ x_i\}_{i=1}^n$$

$$\lambda x.(\{\mathit{case}\ t\ \text{of}\ (c_i\ x_i) \to t_i\}_{i=1}^n\ x)$$

$$\swarrow_\eta \qquad \searrow_\gamma$$

$$\{\mathit{case}\ t\ \text{of}\ (c_i\ x_i) \to t_i\}_{i=1}^n \qquad \lambda x.\{\mathit{case}\ t\ \text{of}\ (c_i\ x_i) \to t_i\ x\}_{i=1}^n$$

Contraintes sur l'orientation

- Réduction $\lambda x.f\ x \longrightarrow_\eta f$:

$$\lambda x.\{\mathit{case}\ t\ \text{of}\ (c_i\ x_i) \to c_i\ x_i\}_{i=1}^n$$

$$\swarrow_\eta \qquad \searrow_\gamma$$

$$\lambda x.t \qquad \{\mathit{case}\ t\ \text{of}\ (c_i\ x_i) \to \lambda x.c_i\ x_i\}_{i=1}^n$$

$$\lambda x.(\{\mathit{case}\ t\ \text{of}\ (c_i\ x_i) \to t_i\}_{i=1}^n\ x)$$

$$\swarrow_\eta \qquad \searrow_\gamma$$

$$\{\mathit{case}\ t\ \text{of}\ (c_i\ x_i) \to t_i\}_{i=1}^n \qquad \lambda x.\{\mathit{case}\ t\ \text{of}\ (c_i\ x_i) \to t_i\ x\}_{i=1}^n$$

- Expansion $f \longrightarrow_\eta \lambda x : \alpha.f\ x$:

Contraintes sur l'orientation

- Réduction $\lambda x.f\ x \longrightarrow_\eta f$:

$$\lambda x.\{\mathit{case}\ t\ \text{of}\ (c_i\ x_i) \to c_i\ x_i\}_{i=1}^n$$

$$\swarrow_\eta \qquad \searrow_\gamma$$

$$\lambda x.t \qquad \{\mathit{case}\ t\ \text{of}\ (c_i\ x_i) \to \lambda x.c_i\ x_i\}_{i=1}^n$$

$$\lambda x.(\{\mathit{case}\ t\ \text{of}\ (c_i\ x_i) \to t_i\}_{i=1}^n\ x)$$

$$\swarrow_\eta \qquad \searrow_\gamma$$

$$\{\mathit{case}\ t\ \text{of}\ (c_i\ x_i) \to t_i\}_{i=1}^n \qquad \lambda x.\{\mathit{case}\ t\ \text{of}\ (c_i\ x_i) \to t_i\ x\}_{i=1}^n$$

- Expansion $f \longrightarrow_\eta \lambda x : \alpha.f\ x$:
  - les étiquettes doivent apparaître
  - créées en forme normale pour un critère syntaxique

Normalisation et confluence pour $\longrightarrow_{\beta\gamma\eta}$

Principe de la preuve : établissement

- d'un lemme de substitution : propriétés de $u[x := t]$ selon u et t.

Normalisation et confluence pour $\longrightarrow_{\beta\gamma\eta}$

Principe de la preuve : établissement

- d'un lemme de substitution : propriétés de $u[x := t]$ selon u et t.
  sous 2 hypothèses :

Normalisation et confluence pour $\longrightarrow_{\beta\gamma\eta}$

Principe de la preuve : établissement

- d'un lemme de substitution : propriétés de $u[x := t]$ selon u et t.
  sous 2 hypothèses :
  - l'existence d'une forme normale syntaxique

Normalisation et confluence pour $\longrightarrow_{\beta\gamma\eta}$

Principe de la preuve : établissement

- d'un lemme de substitution : propriétés de $u[x := t]$ selon u et t.
  sous 2 hypothèses :
  - l'existence d'une forme normale syntaxique
  - la décomposition des types fonctionnels

Normalisation et confluence pour $\longrightarrow_{\beta\gamma\eta}$

Principe de la preuve : établissement

- d'un lemme de substitution : propriétés de $u[x := t]$ selon u et t.
  sous 2 hypothèses :
  - l'existence d'une forme normale syntaxique
  - la décomposition des types fonctionnels
- de la réduction du sujet

Normalisation et confluence pour $\longrightarrow_{\beta\gamma\eta}$

Principe de la preuve : établissement

- d'un lemme de substitution : propriétés de $u[x := t]$ selon u et t.
  sous 2 hypothèses :
  - l'existence d'une forme normale syntaxique
  - la décomposition des types fonctionnels
- de la réduction du sujet
- de la commutativité entre $\longrightarrow_{\beta\gamma}$ et $\longrightarrow_{\eta}$

Normalisation et confluence pour $\longrightarrow_{\beta\gamma\eta}$

Principe de la preuve : établissement

- d'un lemme de substitution : propriétés de $u[x := t]$ selon u et t.
  sous 2 hypothèses :
  - l'existence d'une forme normale syntaxique
  - la décomposition des types fonctionnels
- de la réduction du sujet
- de la commutativité entre $\longrightarrow_{\beta\gamma}$ et $\longrightarrow_{\eta}$
- de l'unicité des formes normales

Normalisation et confluence pour $\longrightarrow_{\beta\gamma\eta}$

Principe de la preuve : établissement

- d'un lemme de substitution : propriétés de $u[x := t]$ selon u et t.
  sous 2 hypothèses :
  - l'existence d'une forme normale syntaxique
  - la décomposition des types fonctionnels
- de la réduction du sujet
- de la commutativité entre $\longrightarrow_{\beta\gamma}$ et $\longrightarrow_{\eta}$
- de l'unicité des formes normales
- de la normalisation faible de $\longrightarrow_{\eta}$

Normalisation et confluence pour $\longrightarrow_{\beta\gamma\eta}$

Principe de la preuve : établissement

- d'un lemme de substitution : propriétés de $u[x := t]$ selon u et t.
  sous 2 hypothèses :
  - l'existence d'une forme normale syntaxique
  - la décomposition des types fonctionnels
- de la réduction du sujet
- de la commutativité entre $\longrightarrow_{\beta\gamma}$ et $\longrightarrow_{\eta}$
- de l'unicité des formes normales
- de la normalisation faible de $\longrightarrow_{\eta}$
- de la normalisation faible et la confluence de $\longrightarrow_{\beta\gamma\eta}$

Contributions

- Application d'une extension des ACG au contrôle des mouvements grammaticaux

Contributions

- Application d'une extension des ACG au contrôle des mouvements grammaticaux
- Preuve formelle de la confluence et de la normalisation de la relation de réduction de ce calcul

Contributions

- Application d'une extension des ACG au contrôle des mouvements grammaticaux
- Preuve formelle de la confluence et de la normalisation de la relation de réduction de ce calcul
- La preuve lorsque la relation est étendue avec l'η-conversion est ramenée à l'établissement de 2 hypothèses

Perspectives

- Prouver les hypothèses subsistantes

Perspectives

- Prouver les hypothèses subsistantes
- Identifier les fragments décidables de ces extensions

Perspectives

- Prouver les hypothèses subsistantes
- Identifier les fragments décidables de ces extensions
- Donner un algorithme pour la vérification de type

Perspectives

- Prouver les hypothèses subsistantes
- Identifier les fragments décidables de ces extensions
- Donner un algorithme pour la vérification de type
- Étendre la couverture des phénomènes modélisés

Références

Iliano Cervesato et Frank Pfenning. A linear logical framework. In *LICS*, pages 264–275, 1996.

Sam Lindley. Extensional rewriting with sums. In *Proceeding TLCA'07 Proceedings of the 8th international conference on Typed lambda calculi and applications*, pages 255–271. Springer-Verlag Berlin, 2007.

V. Mihalicek. *Serbo-croatian Word Order: A Logical Approach*. PhD thesis, 2012.