Une approche de la théorie de D. Scott et application à la sémantique des types abstraits génériques

Roger Soler

THESE

présentée à

l'Université Scientifique et Médicale de Grenoble

pour obtenir le grade de

DOCTEUR DE TROISIEME CYCLE
"Informatique"

par

Roger SOLER

UNE APPROCHE DE LA THEORIE DE D. SCOTT ET APPLICATION A LA SEMANTIQUE DES TYPES ABSTRAITS GENERIQUES.

Thèse soutenue le 21 septembre 1982 devant la Commission d'Examen :

Monsieur	C. DELOBEL	:	Président
Messieurs	G. BERRY		Examinateurs
	D. BERT		
	C. BOITET		
	Ph. JORRAND		

UNIVERSITE SCIENTIFIQUE ET MEDICALE
DE GRENOBLE

année universitaire 1981-1982

Président de l'Université : M. TANCHE

MEMBRES DU CORPS ENSEIGNANT DE L'U.S.M.G.

PROFESSEURS DE 1ère CLASSE

Mle AGNIUS DELORD Claudine	Biophysique
ALARY Josette	Chimie analytique Fac. LA TRONCHE
MM. AMBLARD Pierre	Clinique dermatologique CHR LES SABLONS
AMBROISE THOMAS Pierre	Parasitologie CHR LES SABLONS
ARNAUD Paul	Chimie organique
ARVIEU Robert	Physique nucléaire I.S.N.
AUBERT Guy	Physique C.N.R.S.
AYANT Yves	Physique appronfondie
MME BARBIER Marie-Jeanne	Electrochimie
MM. BARBIER Jean-Claude	Physique expérimentale C.N.R.S. (labo. de magnétisme)
BARJON Robert	Physique nucléaire I.S.N.
BARNOUD Fernand	Biosynthèse de la cellulose-Biologie
BARRA Jean-René	Statistiques - Maths appliquées
BEAUDOING André	Clinique pédiatrie et puériculture-LATREILLE-
BELORISKY Elie	Physique C.E.N.G. - D.R.F.
BENZADEN Claude	Mathématiques pures
MME BERIEL Hélène	Pharmacodynamie Fac. LA TRONCHE
MM. BERNARD Alain	Mathématiques pures
MME BERTRANDIAS Françoise	Mathématiques pures
MM. BERTRANDIAS Jean-Paul	Mathématiques pures
BEZES Henri	Clinique chirurgicale & traumat. Hôp. SUD
BILLET Jean	Géographie
BONNET Jean-Louis	Clinique ophtalmologie Fac. LES SABLONS
BONNET EYMARD Joseph	Clinique hépato-gastro-entérolo Fac. SABLONS
MME BONNIER Jane Marie	Chimie générale
MM. BOUCHERLE André	Chimie et toxicologie Fac. MEYLAN
BOUCHET Yves	Anatomie Fac. "La Merci"

MM. BOUCHEZ Robert	Physique nucléaire I.S.N.
BRAVARD Yves	Géographie
BUTEL Jean	Orthopédie CHR LES SABLONS
CABANEL Guy	Clinique rhumatologie et hydro. SABLONS -
CARLIER Georges	Biologie végétale
CAU Gabriel	Médecine légale et toxicologie Fac. SABLONS
CAUQUIS Georges	Chimie organique
CHARACHON Robert	Clinique O.R.L. CHR LES SABLONS
CHAMPETIER Jean	Anatomie topographique et app.Fac."La Merci"
CHATEAU Robert	Clinique neurologique CHR LES SABLONS
CHIBON Pierre	Biologie animale
COLIN DE VERDIERE Yves	Mathématiques pures
COUDERC Pierre	Anatomie pathologique CHR LES SABLONS
CRABBE Pierre	C.E.R.M.O.
CUSSAC Max	Chimie thérapeutique Fac. LA TRONCHE
CYROT Michel	Physique du solide
DAUMAS Max	Géographie
DEBELMAS Jacques	Géologie générale
DEGRANGE Charles	Zoologie
DELOBEL Claude	M.I.A.G. Mathématiques appliquées
DELORMAS Pierre	Pneumo-phtisiologie CHR Pav. D1
DEMENGE Charles	Pharmacodynamique Fac. LA TRONCHE
DENIS Bernard	Clinique cardiologique CHR LES SABLONS
DEPORTES Charles	Chimie minérale
DESRE Pierre	Electrochimie
DODU Jacques	Mécanique appliquée IUT 1
DOLIQUE Jean-Michel	Physique des plasmas
DUCROS Pierre	Cristallographie
FAURE Jacques	Médecine légale (interne & toxicologie)SABLON
FONTAINE Jean-Marc	Mathématiques pures
GAGNAIRE Didier	Chimie physique
GASTINEL Noël	Analuse numérique Mathématiques appliquées
GAVEND Jean-Michel	Pharmacologie Fac. "La Merci"
GEINDRE Michel	Electro-radiologie CHR LES SABLONS
GERBER Robert	Mathématiques pures
GERMAIN Jean-Pierre	Mécanique
GIRAUD Pierre	Géologie
IDELMAN Simon	Physiologie animale

MM.	JANIN Bernard	Géographie
	JEANNIN Charles	Pharmacie galénique Fac. MEYLAN
	JOLY Jean-René	Mathématiques pures
	JULLIEN Pierre	Mathématiques appliquées
	KAHANNE André	Physique
Mme	KAHANNE Josette	Physique
MM.	KLEIN Joseph	Mathématiques pures
	KOSZUL Jean-Louis	Mathématiques pures
	KRAKOWIAK Sacha	Mathématiques appliquées
	KUPTA Yvon	Mathématiques pures
	LACAZE Albert	Thermodynamique
	LACHARME Jean	Biologie cellulaire Fac. MEYLAN
Mme	LAJZEROWICZ Jeannine	Physique
MM.	LAJZEROWICZ Joseph	Physique
	LATREILLE René	Chirurgie thoracique CHR LES SABLONS
	LATURAZE Jean	Biochimie pharmaceutique Fac. LA TRONCHE
	LAURENT Pierre	Mathématiques appliquées
	DE LEIRIS Joël	Biologie
	LE NOC Pierre	Bactériologie virologie Fac. "La Merci"
	LLIBOUTRY Louis	Géophysique
	LOISEAUX Jean-Marie	Sciences nucléaires I.S.N.
	LOUP Jean	Géographie
	LUU DUC Cuong	Chimie générale et minérale Fac.LA TRONCHE
	MACHE Régis	Physiologie végétale
	MALINAS Yves	Clinique obstrétricale CHR pav.maternité
Mle	MARIOTTE Anne-Marie	Pharmacognostie Fac. LA TRONCHE
MM.	MAYNARD Roger	Physique du solide
	MAZARE Yves	Clinique médicale A CHR pav. D. Villars
	MICHEL Robert	Minéralogie et pétrographie (géologie)
	MICOUD Max	Clinique maladie infectieuses CHR LES SABLONS
	MOURIQUAND Claude	Histologie Fac. "La Merci"
	NEGRE Robert	Génie civil IUT 1
	NOZIERES Philippe	Spectrométrie - Physique
	OMONT Alain	Astrophysique
	OZENDA Paul	Botanique (biologie végétale)
	PAVAN Jean-Jacques	Mathématiques pures
	PEBAY PEYROULA Jean-Claude	Physique
	PERRET Jean	Sémeiologie médicales (neurologie)
	PERRIAUX Jacques	Géologie

MM. PERRIER Guy	Géophysique
PIERRARD Jean-Marie	Mécanique
RACHAIL Michel	Clinique médicale B CHR Calmettes Guérin
RASSAT André	Chimie systématique
RENARD Michel	Thermodynamique
Mme RENAUDET Jacqueline	Bactériologie Fac. LA TRONCHE
MM. REVOL Michel	Urologie CHR LES SABLONS
RICHARD Lucien	Biologie végétale
Mme RINAUDO Marguerite	Chimie CERMAV
MM. ROCHAT Jacques	Hygiène et hydrologie Fac. LA TRONCHE
DE ROUGEMONT Jacques	Neuro-chirurgie CHR LES SABLONS
SARRAZIN Roger	Clinique chirurgicale B CHR LES SABLONS
Mme SEIGLE MURANDI Françoise	Botanique et crytomanie Fac. MEYLAN
MM. SENGEL Philippe	Biologie animale
SERGERAERT Francis	Mathématiques pures
SIBILLE Robert	Construction mécanique IUT 1
SOUTIF Michel	Physique
TANCHE Maurice	Physiologie Fac. "La Merci"
VAILLANT François	Zoologie
VALENTIN Jacques	Physique nucléaire I.S.N.
VAN CUTSEM Bernard	Mathématiques appliquées
VAUQUOIS Bernard	Mathématiques appliquées
Mme VERAIN Alice	Pharmacie galénique Fac. MEYLAN
MM. VERAIN André	Biophysique Fac."La Merci"
VIGNAIS Pierre	Biochimie médicale Fac. "La Merci"
VIALLON Pierre	Géologie

PROFESSEURS DE 2ème CLASSE

MM. ABIBA Michel	Mathématiques pures
ARMAND Yves	Chimie IUT 1
ARMAND Gilbert	Géographie
AURIAULT Jean-Louis	Mécanique
BEGUIN Claude	Chimie organique
BOEHLER Jean-Paul	Mécanique
BOITET Christian	Mathématiques appliquées
BORNAREL Jean	Physique
BOUTHINON Michel	EEA. IUT 1
BRUGEL Lucien	Energétique IUT 1
CASTAING Bernard	Physique
CHARDON Michel	Géographie

MM.	CHEHIKIAN Alain	EEA IUT 1
	CHENAVAS Jean	Physique IUT 1
	COHENADDAD Jean-Pierre	Physique
	CONTE René	Physique IUT 1
	DENEUVILLE Alain	Physique
	DEPASSEL Roger	Mécanique des fluides
	DOUCE Roland	Physiologie végétale
	DUFRESNOY Alain	Mathématiques pures
	GASPARD François	Physique
	GAUTRON René	Chimie
	GIDON Maurice	Géologie
	GIGNOUX Claude	Sciences nucléaires I.S.N.
	GOSSE Jean-Pierre	EEA IUT 1
	GROS Yves	Physique IUT 1
	GUITTON Jacques	Chimie
	HACQUES Gérard	Mathématiques appliquées
	HERBIN Jacky	Géographie
	HICTER Pierre	Chimie
	JOSELEAU Jean-Paul	Biochimie
	KERCKOVE Claude	Géologie
	KUHN Gérard	Physique IUT 1
	LEBRETON Alain	Mathématiques appliquées
Mme	LONGEQUEUE Nicole	Sciences nucléaires I.N.S.
MM.	LUCAS Robert	Physiques
	LUNA Domingo	Mathématiques pures
	MASCLE Georges	Géologie
	MICHOULIER Jean	Physique IUT 1
Mme	MINIER Colette	Physique IUT 1
MM.	NEMOZ Alain	Thermodynamique (CNRS - CRTBT)
	NOUGARET Marcel	Automatique IUT 1
	OUDET Bruno	Mathématiques appliquées
	PEFFEN René	Métallurgie IUT 1
	PELMONT Jean	Biochimie
	PERARD Jacques	EEA IUT 1
	PERRAUD Robert	Chimie IUT 1
	PERRIN Claude	Sciences nucléaires I.S.N.
	PFISTER Jean-Claude	Physique du solide
	PIBOULE Michel	Géologie
	PIERRE Jean-Louis	Chimie organique

Mle PIERY Yvette	Physiologie animale
MM. RAYNAUD Hervé	Mathématiques appliquées
ROBERT Gilles	Mathématiques pures
ROBERT Jean-Bernard	Chimie physique
ROSSI André	Physiologie végétale
SAKAROVITCH Michel	Mathématiques appliquées
SARROT REYNAUD Jean	Géologie
SAXOD Raymond	Biologie animale
Mle SOUTIF Jeannne	Physique
MM. SCHOOL Pierre-Claude	Mathématiques appliquées
STUTZ Pierre	Mécanique
SUBRA Robert	Chimie
VIDAL Michel	Chimie organique
VIVIAN Robert	Géographie

PROFESSEURS SANS CHAIRE (médecine)

MM. BARGE Michel	Neuro-chirurgie CHR LES SABLONS
BOST Michel	Pédiatrie CHR Enfants 1
BOUCHARLAT Jacques	Psychiatrie Hôpital Sud
CHAMBAZ Edmond	Biochimie (hormonologie) CHR Pav. P. Gérin
COLOMB Maurice	Biochimie Hôpital Sud
COULOMB Max	Radiologie Hôpital Sud
Mme ETERRADOSSI Jacqueline	Physiologie Fac. "La Merci"
MM. GROULADE Joseph	Biochimie (A) CHR LES SABLONS
HOLLARD Daniel	Hématologie CHR LES SABLONS
HUGONOT Robert	Gérontologie CHR Pav. Chatin
JALBERT Pierre	Histologie Fac. "La Merci"
MAGNIN Robert	Hygiène Fac. "La Merci"
PHELIP Xavier	Rhumatologie CHR LES SABLONS
RACINET Claude	Gynécologie obstrétrique Hôpital Sud
REYMOND Jean-Charles	Chirurgie générale 1, rue de la Liberté
STIEGLITZ Paul	Anesthésiologie CHR LES SABLONS
VROUSOS Constantin	Radiothérapie CHR LES SABLONS

MAITRES DE CONFERENCES AGREGES (médecine)

MM. BACHELOT Yvan	Endrocrinologie CHR LES SABLONS
BENABID Alim Louis	Médecine et chirurgie CHR LES SABLONS
BERNARD Pierre	Gynécologie obstrétrique CHR Pav. P. Gérin
CONTAMIN Charles	Chirurgie thoracique CHR LES SABLONS
CORDONNIER Daniel	Néphrologie CHR LES SABLONS

MM. CROUZET Guy	Radiologie CHR LES SABLONS
DEBRU Jean-Luc	Médecine interne CHR LES SABLONS
DUPRE Alain	Chirurgie générale CHR LES SABLONS
DYON Jean-François	Chirurgie infantile CHR LES SABLONS
FAURE Claude	Anatomie et organogénese Fac. "La Merci"
FAURE Gilbert	Urologie CHR LES SABLONS
FLOYRAC Roger	Biophysique Fac. "La Merci"
FOURNET Jacques	Hépato-gastro-entérologie CHR LES SABLONS
GIRARDET Pierre	Anesthésiologie CHR LES SABLONS
GUIDICELLI Henri	Chirurgie générale CHR LES SABLONS
GUIGNIER Michel	Thérapeutique (réanimation) CHR LES SABLONS
JUNIEN-LAVILLAUROY Claude	Clinique ORL CHR LES SABLONS
KOLODIE Lucien	Hématologie biologie CHR LES SABLONS
LETOUBLON Christian	Chirurgie générale CHR LES SABLONS
MASSOT Christian	Médecine interne CHR Pav. D. Villars
MOUILLON Michel	Ophtalmologie CHR LES SABLONS
PARAMELLE Bernard	Pneumologie CHR Pav. D2
RAMBAUD Pierre	Pédiatrie CHR Médecine Néonatale
RAPHAEL Bernard	Stomatologie CHR LES SABLONS
SCHAERER René	Cancérologie CHR LES SABLONS
SEIGNEURIN Jean-Marie	Bactériologie-virologie Fac. "La Merci"
SOTTO Jean-Jacques	Hématologie CHR LES SABLONS
STOEBNER Pierre	Anatomie-pathologique CHR LES SABLONS

Je tiens à remercier Monsieur le Professeur Claude Delobel, Directeur de Laboratoire IMAG, qui m'a fait l'honneur de présider le Jury.

Je remercie Monsieur Gérard Berry, Maître de Recherche à l'Ecole des Mines, qui a bien voulu donner son avis de spécialiste sur mon travail.

Je remercie Monsieur Didier Bert, Chargé de Recherche au CNRS, qui m'a proposé mon sujet de thèse. Il a su orienter mon travail au cours de longues discussions constructives, et ses critiques ont toujours été justifiées. C'est à lui que je dois la "mise en français" de cette thèse, les latino-américanismes ou tournures catalanes restant à ma charge.

Je remercie Monsieur Christian Boitet, Professeur à l'Université de Grenoble, d'avoir bien voulu juger cette thèse. Il sut me communiquer, pendant son cours de DEA, son intérêt pour les aspects théoriques de l'Informatique.

Je remercie Monsieur Philippe Jorrand, Directeur de Recherche au CNRS, pour la confiance qu'il m'a témoignée en m'accueillant dans son équipe dès mon arrivée en France et pour l'intérêt soutenu qu'il a manifesté à l'égard de mon travail.

Cette thèse a profité de l'expérience et des observations pertinentes de mes collègues d'équipe Ricardo Caferra, Paul Jacquet et également de Monsieur Felipe Bracho de l'Universidad Nacional Autónoma de México.

Ces deux dernières années, j'ai effectué mon travail d'enseignant à l'Universidad Central de Venezuela. J'ai bénéficié au cours de cette période, de l'appui logistique en France de mes collègues Mauricio Milchberg et Jacques Voiron.

Je ne voudrais pas dissocier de ces remerciements tous mes collègues du Laboratoire avec qui j'ai souvent eu de fructueuses discussions.

Je remercie Mademoiselle Véronique Catris pour sa patience et son efficacité à transformer mon manuscrit en ce document lisible, je remercie également tout le personnel du Service de reprographie qui a assuré, dans un temps record, la réalisation matériel de cette thèse.

Cette recherche a été effectuée pendant que j'étais boursier du Consejo de Desarrollo Científico y Humanístico de la Universidad Central de Venezuela, et a aussi bénéficié de l'appui du GRECO Programmation CNRS.

Roger Soler Aniorte

" I would like to begin with some personal remarks. When I think of the number of headaches I have caused people in Computer Science who have tried to figure out the mathematical details of the Theory of Domains, I have to cringe. "

Dana S. SCOTT

1er juillet 1982.

TABLE DES MATIERES

0. Introduction 19

1. Domaines 29

1.1. Définitions préliminaires 29
1.2. Dirigé. La notion d'approximation 31
1.3. Treillis 34
1.4. Base. La notion d'élément calculable 40
1.5. Domaines de calcul 42

2. Le concept de fonction (continue, monotone) dans les domaines 49

2.1. Comportement des fonctions sur les bornes supérieures des ensembles dirigés 50
2.2. Continuité dans les domaines de calcul 53
2.3. Points fixes et caractérisation de domaines 56

3. Construction de domaines de calcul 63

3.1. Produit cartésien 63
3.2. Union disjointe ou somme 68
3.3. Espace des fonctions continues ou opération d'exponentiation 73

4. Quelques fonctions (opérateurs, fonctionnelles) continues dans les domaines 83

4.1. La composition de fonctions 83
4.2. Plus petit point fixe d'une fonction continue 86
4.3. Caractérisation de l'espace des fermetures continues sur un domaine 90
4.4. Points fixes d'une fonction continue, engendrés d'une manière continue 94
4.5. Caractérisation au moyen de rétracts à caractère fini des opérations $\times$, $+$, $\rightarrow$. 104

5. Domaines réflexifs et équations de domaines 115

5.1. Connexions de Galois 115
5.2. Domaines universels 122

5.3. Solution des équations de domaines 123
5.4. Le modèle graphique *Pw* 127
5.5. Autre point de vue 132

6. Sur l'interprétation des types de données 137
6.1. Interprétation de la notion de satisfaction 142
6.2. Interprétation des types abstraits algébriques 146
6.3. Interprétation des propriétés 150
6.4. Interprétation des types génériques 151
6.5. Cohérence dans les types génériques 151

7. Conclusion 155

8. Bibliographie 159

INTRODUCTION

0. INTRODUCTION

La sémantique dénotationnelle ou mathématique d'un langage de programmation est le nom de la méthode développée par D. Scott et C. Strachey [ScSt71] pour décrire les concepts qui apparaissent dans les langages de programmation, et consiste à associer à chaque constructeur syntaxique du langage que l'on étudie -langage de programmation- une valeur ou objet mathématique. Cette association se réalise à partir de la morphologie du langage que l'on étudie (syntaxe abstraite). Ces valeurs ou objets mathématiques appartiennent à une théorie mathématique, dans ce cas la théorie des treillis, et c'est cette théorie qui conditionne le méta-langage dans lequel s'exprime la sémantique du langage de programmation. L'exactitude du méta-langage est validée par l'existence d'un modèle, c'est-à-dire un objet mathématique existant, ou qui peut être construit, où sont interprétés les concepts du méta-langage.
A. Tarski [TaA 36] et [TaA 33] propose ce type de schéma pour établir la sémantique des langages formels selon l'ordre logique que ceux-ci possèdent. Dans la théorie que présente D. Scott [ScD 72], il est possible d'établir la sémantique du lambda-calcul, dans lequel l'ordre est infini, sans réaliser une théorie des types dans ce langage formel. Cette étude de Scott a été à l'origine d'une théorie "mathématique" du calcul [ScD 70] et de plusieurs développements pour montrer la validité de certaines méthodes de preuves de programmes [dBJ 80]. Comme dans toute théorie du calcul [McJ 63], on trouve une définition précise des fonctions calculables [ScD 76].

Dans ce travail, nous allons présenter la théorie qui traite des objets mathématiques ou des valeurs qui sont utilisés pour établir la sémantique des langages de programmation avec la méthode dénotationnelle. Ces objets sont les éléments d'une classe d'ensembles, les ensembles eux-mêmes, et les fonctions (continues) entre ces ensembles ; les ensembles sont structurés par une relation d'ordre et possèdent certaines propriétés spéciales qui les caractérisent comme treillis algébriques, et que nous appellerons ici domaines de calcul.

Pour donner une idée du point de vue dénotationnel et des problèmes qui se rencontrent (et se résolvent) en utilisant cette méthode, nous verrons la définition d'un langage qui caractérise les fonctions primitives récursives : le langage LOOP

[MiM 67] [MeRi 67] ; une explication plus détaillée de la définition de ce langage se trouve en [TeR 76], d'où nous avons pris l'exemple.

La syntaxe abstraite du langage LOOP est la suivante :

. Domaines syntaxiques :

$\Xi \in$ Var	variables
$E \in$ Exp	expressions
$\Gamma \in$ Cmd	commandes
$\Psi \in$ Prg	programmes

. Productions :

$E ::= 0$	zéro
$\quad \mid \Xi$	variable
$\quad \mid$ succ E	successeur de E
$\Gamma ::= \Gamma_1 ; \Gamma_2$	séquence
$\quad \mid \Xi := E$	affectation
$\quad \mid$ <u>faire</u> E <u>fois</u> Γ	itération
$\quad \mid (\Gamma)$	parenthésage

Ψ ::= <u>lire</u> Ξ ; Γ ; <u>écrire</u> E

Avec cette partie, nous avons donné la "syntaxe abstraite" du langage, c'est-à-dire, les différents ensembles (domaines syntaxiques : variables, expressions, commandes, programmes) qui devront être interprétés, et comment ils sont engendrés par la grammaire (productions) du langage. La syntaxe abstraite évite les aspects accessoires du langage : comment sont construits les identificateurs des variables n'est pas important. Maintenant nous procédons à la définition des objets mathématiques "dénotés" par les constructeurs du langage (domaines sémantiques) et à la façon dont se fait l'association (fonctions sémantiques).

. Domaines sémantiques :

$\nu \in$ Val = Nat	valeurs = domaines des entiers naturels.
$\sigma \in S$ = Var $\rightarrow$ Val	états de la mémoire = fonctions avec les variables comme domaine et les valeurs comme codomaine.

. Fonctions sémantiques (Interprétation) :

$\mathcal{M}$: Prg $\rightarrow$ [Val $\rightarrow$ Val]

$\mathcal{E}$: Exp $\rightarrow$ [S $\rightarrow$ Val]

$\mathcal{C}$: Cmd $\rightarrow$ [S $\rightarrow$ S]

Ceci est la fonctionnalité des fonctions sémantiques et exprime l'idée suivante : la fonction $\mathcal{M}$ a comme domaine les programmes écrits en LOOP et comme codomaine les fonctions des naturels dans les naturels, dans ce cas : les fonctions primitives récursives. $\mathcal{E}$ est l'évaluation d'une expression $E \in$ Exp avec un état de la mémoire $\sigma \in S$ dans laquelle se trouvent les valeurs des variables qui apparaissent dans E. Finalement, $\mathcal{C}$ associe à chaque commande $\Gamma \in$ Cmd la transformation de l'état de la mémoire que la commande réalise. Maintenant, voyons la définition exacte de ces fonctions sémantiques à la façon de la notation lambda([ChA 41],[HLS 72]).

$$(1)\quad \mathcal{M}[\![\underline{\text{lire}}\ \Xi\ ;\ \Gamma\ ;\ \underline{\text{écrire}}\, E]\!]\nu = \mathcal{E}[\![E]\!]\ (\mathcal{C}[\![\Gamma]\!]\sigma_i)$$

$$\text{où } \sigma_i = \lambda\, \Xi'.\underline{\text{si}}\ \ \Xi' = \Xi\ \ \underline{\text{alors}}\ \nu\ \underline{\text{sinon}}\ 0$$

C'est-à-dire : σ_i est l'état initial de la mémoire où toutes les variables valent 0 ($0 \in$ Val) excepté Ξ qui vaut ν cet état initial est transformé par $\mathcal{C}[\![\Gamma]\!]$, et l'état final est le résultat de cette transformation, dans lequel l'expression E est évaluée. Entre les symboles $[\![$ et $]\!]$, nous indiquons quel est le constructeur syntaxique argument de la fonction. Nous présentons la définition de $\mathcal{E}$ et $\mathcal{C}$ selon les cas qui apparaissent dans les productions :

$$(2)\quad \mathcal{E}[\![0]\!]\ \sigma = 0 \qquad\qquad (0 \in \text{Val})$$

$$(3)\quad \mathcal{E}[\![\Xi]\!]\ \sigma = \sigma(\Xi)$$

$$(4)\quad \mathcal{E}[\![\text{succ } E]\!]\ \sigma = \mathcal{E}[\![E]\!]\ \sigma + 1 \qquad\qquad (1 \in \text{Val})$$

$$(5)\quad \mathcal{C}[\![(\Gamma)]\!]\ \sigma = \mathcal{C}[\![\Gamma]\!]\ \sigma$$

$$(6)\quad \mathcal{C}[\![\Gamma_1 ; \Gamma_2]\!]\ \sigma = \mathcal{C}[\![\Gamma_2]\!]\ (\mathcal{C}[\![\Gamma_1]\!]\ \sigma)$$

$$(7)\quad \mathcal{C}[\![\Xi := E]\!]\ \sigma = \lambda\, \Xi'.\underline{\text{si}}\ \ \Xi' = \Xi\ \ \underline{\text{alors}}\ \mathcal{E}[\![E]\!]\ \sigma\ \underline{\text{sinon}}\ \sigma(\Xi')$$

$$(8)\quad \mathcal{C}[\![\underline{\text{faire}}\ E\ \underline{\text{fois}}\ \Gamma]\!]\ \sigma = \lambda\, \Xi\ .\underline{\text{si}}\ \mathcal{E}[\![E]\!]\ \sigma = 0\ \underline{\text{alors}}\ \sigma(\Xi)$$

$$\underline{\text{sinon}}\ \underbrace{(\mathcal{C}[\![\Gamma]\!]\ldots(\mathcal{C}[\![\Gamma]\!]\ldots(\mathcal{C}[\![\Gamma]\!]\ \sigma)\ldots)}_{\nu\ \text{fois}}(\Xi)$$

$$\text{où } \nu = \mathcal{E}[\![E]\!]\ \sigma$$

Il convient de préciser que les constantes 0 et 1 qui apparaissent à droite dans les définitions des fonctions sémantiques sont les valeurs 0 et 1 des entiers (Val = Nat dans les domaines sémantiques) ; l'opération "+" (équation (4)) est l'opération d'addition dans les entiers naturels.

Telle qu'a été définie la sémantique du langage LOOP, il n'est pas nécessaire que les domaines sémantiques soient structurés ; c'est-à-dire, nous avons seulement demandé que le domaine des valeurs possède deux constantes 0 et 1, et que soient définies les opérations d'addition et d'égalité (équation (8)) ; par rapport aux états de la mémoire -fonctions des variables dans les valeurs-, qu'il existe l'égalité dans l'ensemble syntaxique des variables (équation (7)). Maintenant nous allons redéfinir le fonction sémantique C pour le cas de l'itération (équation (8)) :

Soit $J_S = Nat \times [S \to S] \times S$
et Iter : $J_S \to S$
Définissons Iter de la manière suivante :

$\text{Iter} = \lambda(n,\phi,x).\ \underline{\text{si}}\ n = 0\ \underline{\text{alors}}\ x\ \underline{\text{sinon}}\ \phi(\text{Iter}(n-1,\phi,x))$

Maintenant, soit

$F : [J_S \to S] \to [J_S \to S]$

et $F = \lambda f.\ \lambda(n,\phi,x).\ \underline{\text{si}}\ n = 0\ \underline{\text{alors}}\ x\ \underline{\text{sinon}}\ \phi(f(n-1,\phi,x))$
la fonction Iter est un point fixe de la fonction F, c'est-à-dire :

$F(\text{Iter}) = \text{Iter}$

Le théorème de Tarski (§2.3.1) garantit l'existence de points fixes, quand l'ensemble $[J_S \to S]$ est structuré (est un treillis complet, §1.3.2) et F est monotone (§2.1.1) dans cette structure ; de plus, l'ensemble de points fixes possède aussi la même structure. Dans le cas où F est continue (§2.1.2), nous pouvons définir un opérateur μ, (§4.2.1) tel que μ appliqué à F donne le plus petit point fixe de F, c'est-à-dire

$\mu : ([J_S \to S] \to [J_S \to S]) \to [J_S \to S]$
$\mu F = F(\mu F)$
et $(\forall f)\ F(f) = f \Rightarrow \mu F \sqsubseteq f$
où $\sqsubseteq$ est la relation d'ordre partiel qui structure l'ensemble $[J_S \to S]$ comme treillis complet.

Avec ces définitions, l'équation (8) a la forme suivante :

(8a) $C[\![\underline{\text{faire}}\ E\ \underline{\text{fois}}\ \Gamma]\!]\,\sigma = (\mu F)\ (\mathcal{E}[\![E]\!]\,\sigma,\ C[\![\Gamma]\!],\ \sigma)$

Il est bon de faire remarquer que, dans cet exemple du langage LOOP, nous avons introduit le problème du point fixe d'une manière artificielle ; cependant, le problème apparaît naturellement dans les langages de programmation qui possèdent l'instruction "tantque ... faire ... ", ou la récursivité. Nous noterons aussi que l'ensemble $[J_S \to S]$ (fonctions continues de J_S dans S) est structuré par une relation d'ordre partiel -définie d'une manière naturelle- et possède les propriétés nécessaires pour être un domaine, si J_S et S sont des domaines ; cela signifie que $\to$ construit des domaines à partir de domaines, ainsi que $\times$ et + (produit cartésien et union disjointe, respectivement) ; nous verrons cela plus en détail dans les paragraphes §3.1, 3.2, 3.3.

Maintenant, nous allons introduire dans le langage LOOP la notion de procédure, et avec elle, les domaines définis de façon circulaire. Pour cela, nous définirons un nouveau type d'expression -<u>proc</u>(Γ)- et un nouveau type de commandes -call Ξ - ; l'affectation assigne une dénotation (Ξ ,variable) à ce type d'expression : <u>proc</u>(Γ), qui est une transformation dans les états de la mémoire. Donc, parmi les valeurs dénotables par les variables, nous aurons les procédures. Pour réaliser ceci, nous modifierons le domaine sémantique Val et définirons un nouveau domaine Proc. Les domaines sémantiques sont définis par les équations suivantes :

. Domaines sémantiques :

$$\mathrm{Val} = \mathrm{Nat} + \mathrm{Proc} \qquad (+ : \text{union disjointe})$$

$$\mathrm{Proc} = S \to S$$

$$S = \mathrm{Var} \to \mathrm{Val}$$

Dans ces équations de définition des domaines sémantiques, il faut que Val soit défini de façon circulaire (récursivement) ; de plus nous avons deux "types" de valeurs : Nat et Proc. Les fonctions sémantiques E et C interprètent les nouveaux cas de la manière suivante :

(9) $E [\![\underline{\mathrm{proc}}(\Gamma)]\!] \sigma = \lambda\sigma'. C [\![\Gamma]\!] \sigma'$

(10) $C [\![\mathrm{call}\ \Xi]\!] \sigma = (\sigma(\Xi) \mid \mathrm{Proc})\ \sigma$

où $(\sigma(\Xi) \mid \mathrm{Proc})$ est le résultat de σ sur la composante Proc de Val.

Le reste de la définition de E et C reste identique ; il faut seulement prendre soin, dans les équations (4) et (8), que la valeur de l'expression E en σ ne soit pas de "type" Proc.

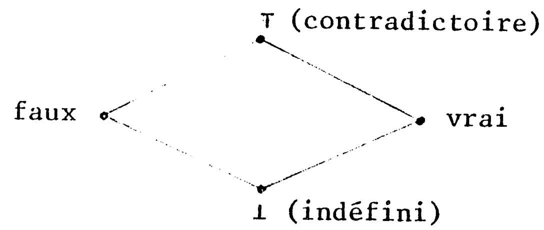
L'équation (10) est très proche du cas de l'application d'une fonction à elle-même, ce qui peut produire des paradoxes ; par exemple :

$\sigma = \lambda x.\ \underline{\mathrm{si}}\ x(x) = 0\ \underline{\mathrm{alors}}\ 1\ \underline{\mathrm{sinon}}\ 0$

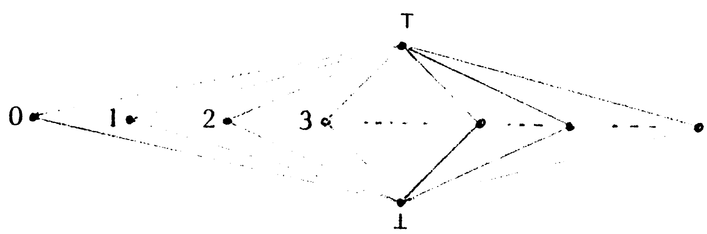
le paradoxe se trouve quand on veut faire $\sigma(\sigma)$.

Un aspect important de la méthode dénotationnelle est l'utilisation du lambda-calcul d'une façon libre (c.a.d. sans types), ce qui, nous l'avons vu, peut créer des paradoxes. Dans ce travail, nous présentons une théorie avec laquelle il est possible d'établir des modèles sémantiques pour le lambda-calcul, et par conséquent pour les langages de programmation. Un autre aspect important de cette méthode est que lorsque l'objet associé est une fonction, on ne donne pas une façon de calculer cette fonction (algorithme) ; au contraire, la fonction se présente sous forme extensive (c.a.d. l'ensemble des paires que représente la fonction). On verra plus clairement cette représentation extensive (et unique) d'une fonction quand nous présenterons les modèles (§ 5.4.). Il est important de remarquer que les constructeurs sémantiques que nous avons montrés (fonctions, domaines, opérations entre domaines, etc..) fournissent une base pour établir des modèles sémantiques concis des langages de programmation [TeR 76].

Pour terminer cette introduction, donnons une idée de la relation d'ordre partiel qui structure les domaines que nous devons manipuler.
Cet ordre est basé sur une certaine notion d'approximation entre les éléments selon l'information qu'ils contiennent. Par exemple, si nous prenons les deux valeurs booléennes : vrai, faux et ajoutons les valeurs indéfini et contradictoire, nous saurons que vrai et faux sont incomparables, mais contiennent plus d'information que indéfini, et ces trois contiennent moins d'information que la valeur contradictoire. Ainsi, nous représenterons graphiquement le domaine des valeurs booléennes de la manière suivante :



La valeur indéfinie sera représentée par le symbole $\perp$ (bottom) et la valeur surdéfinie (dans l'exemple : contradictoire) par $\top$ (top). Avec cette notion d'information, il faut que les entiers naturels soient un domaine avec l'ordre suivant :



Dans le cas de Nat (domaine des entiers naturels), nous pouvons, par exemple, interpréter $\top$ comme la solution de l'équation :

$$x = x + 1$$

c'est-à-dire, le point fixe de la fonction :

$$f = \lambda x.x+1$$

L'idée d'approximation apparaît plus facilement dans le cas des fonctions ; soient f_i : Nat $\rightarrow$ Nat définies par :

$$f_1(x) = \begin{cases} 2 & \text{si } x \text{ est pair} \\ 7 & \text{si } x = 7 \\ \perp & \text{(non définie) sinon } (x \text{ impair et } x \neq 7) \end{cases}$$

$$f_2(x) = \begin{cases} 2 & \text{si } x \text{ est pair} \\ 7 & \text{si } x = 5 \\ 5 & \text{si } x = 7 \\ \perp & \text{sinon} \end{cases}$$

$$f_3(x) = \begin{cases} 2 & \text{si } x \text{ est pair} \\ \perp & \text{sinon} \end{cases}$$

les fonctions f_1 et f_2 sont incomparables, mais sont "plus définies" que f_3 ; nous dirons que f_3 est une approximation de f_1 et f_2.

La fonction "la plus petite" de toutes sera :

$$f(x) = \perp \qquad (\forall x \in \text{Nat})$$

c'est-à-dire, la fonction non-définie pour n'importe quel argument.

C'est avec cette notion d'ordre que nous terminons l'introduction à la méthode dénotationnelle pour établir la sémantique des langages de programmation. Ce travail est divisé en six parties : dans la première, nous présentons les notions générales sur les treillis pour définir les domaines de calcul (treillis algébriques avec base récursive) ; dans la seconde, nous étudions les fonctions entre ces domaines de calcul, le concept de continuité, points fixes, et nous présentons une classe de fonctions qui définissent ces domaines, ces fonctions sont idempotentes et nous les appellerons ici "rétracts" à caractère fini; la troisième partie montre la construction de domaines de calcul à partir d'autres domaines de calcul, ces nouveaux domaines étant le produit cartésien, l'union disjointe, et l'exponentiation entre domaines de calcul ; cette dernière -l'exponentiation- est l'espace des fonctions continues entre deux domaines. Puis, dans la quatrième partie, nous donnons une série d'opérateurs associés aux concepts des langages de programmation, et parmi ces opérateurs nous trouvons les opérateurs de point fixe, opérateurs (rétracts à caractère fini) qui définissent les domaines obtenus dans la troisième partie, et les opérateurs qui produisent des domaines de rétracts à caractère fini (c.a.d. qui établissent un "calcul" de rétracts à caractère fini) ; dans la cinquième partie, nous définissons les domaines réflexifs avec le concept de connexion de Galois, et nous donnons la solution d'équations (circulaires) de domaines ; enfin, dans la partie 5.4., nous exhibons un modèle où sont valides les concepts présentés dans le développement de ce travail ; la sixième partie essaie de résoudre le problème de trouver des domaines de fonctions qui satisfont des équations en vue de donner une sémantique des types abstraits.

CHAPITRE 1.
DOMAINES

1. DOMAINES

Comme nous l'avons déjà mentionné dans l'introduction, les ensembles d'objets que nous manipulons sont structurés par une relation d'ordre partiel. Dans ce chapitre, nous donnons la définition de différents ensembles avec ce type de relation, sur lesquels nous exigerons certaines propriétés, pour arriver aux domaines de calcul, qui seront des treillis algébriques avec une base récursive.

1.1 Définitions préliminaires

1.1.1. Ensemble partiellement ordonné (e.p.o.)

Un ensemble partiellement ordonné est une paire $(P,\sqsubseteq)$, où P est un ensemble non vide, et $\sqsubseteq$ une relation binaire dans P qui satisfait les conditions suivantes :

(i) $(\forall a)\ [a\in P \Rightarrow a \sqsubseteq a]$ réflexive

(ii) $(\forall a,b)\ [a,b\in P \Rightarrow (a \sqsubseteq b \wedge b \sqsubseteq a \Rightarrow a = b)]$ antisymétrique

(iii) $(\forall a,b,c)\ [a,b,c\in P \Rightarrow (a \sqsubseteq b \wedge b \sqsubseteq c \Rightarrow a \sqsubseteq c)]$ transitive

On représentera le fait que deux éléments x et y ne sont pas en relation $(\neg(x\sqsubseteq y) \wedge \neg(y\sqsubseteq x))$ par $x||y$, et on dira qu'ils sont incomparables.

1.1.2. Ensemble totalement ordonné

Un ensemble totalement ordonné ou chaîne est un ensemble partiellement ordonné, où $\sqsubseteq$ satisfait de plus :

(iv) $(\forall a,b)\ [a,b\in P \Rightarrow a \sqsubseteq b \vee b \sqsubseteq a]$ totale

Comme exemples d'ensembles part. ordonnés, nous avons les parties d'un ensemble A ($\mathcal{P}(A)$), où la relation est l'inclusion ; les entiers naturels, où la relation est l'ordre canonique ; ce dernier est aussi une chaîne.

1.1.3. Majorant (minorant)

Soient $(P,\sqsubseteq)$ un e.p.o.et $Q \subseteq P$; $m\in P$ est un majorant (minorant) de Q, s'il existe, et l'on écrira $Q \sqsubseteq m$ $(m \sqsubseteq Q)$ ssi

$(\forall q)\ [q\in Q \Rightarrow q \sqsubseteq m$ (resp. $m \sqsubseteq q)]$

et l'on dira que Q est borné supérieurement (resp. inférieurement).

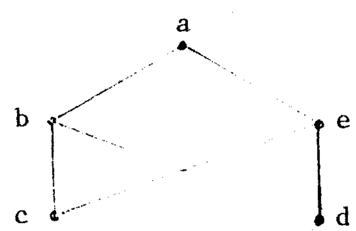
1.1.4. Borne supérieure (borne inférieure)

Soient $(P,\sqsubseteq)$ un e.p.o.et $Q \subseteq P$; $b \in P$, s'il existe, est borne supérieure (inférieure) de Q ssi

$$(\forall p)[p \in P \Rightarrow (b \sqsubseteq p (\text{resp. } p \sqsubseteq b) \Leftrightarrow Q \sqsubseteq p (\text{resp. } p \sqsubseteq Q))]$$

c'est-à-dire, la borne supérieure (inférieure) d'un sous-ensemble d'un e.p.o. est le plus petit (grand), s'il existe, des majorants (minorants) du sous-ensemble. On notera la borne supérieure (inférieure) d'un ensemble Q par $\bigsqcup Q (\bigsqcap Q)$.

Exemple : Soit $(P,\sqsubseteq)$ l'e.p.o. suivant :



soient $Q = \{c,d\}$ et $R = \{c,d,e\}$;
les majorants de Q sont : a, b, e ; et de R : a, e ;
la borne supérieure de $Q(\bigsqcup Q)$ n'existe pas et $\bigsqcup R = e$

1.1.5. Théorèmes sur la borne supérieure

Les trois théorèmes que nous donnons ci-dessous sont conséquence immédiate de la définition de borne supérieure (ils sont valides aussi, en faisant les changements nécessaires -"principe de dualité"- pour la borne inférieure). Le premier et le second théorème expriment l'unicité et la minimalité de la borne supérieure ; le troisième, conséquence du second, la relation qui existe entre les bornes supérieures de deux ensembles, l'un étant sous-ensemble de l'autre ; le quatrième, dont nous donnons la démonstration, exprime la distributivité de la borne supérieure. Nous supposons que les bornes nécessaires existent.

Théorème 1 : $[(P,\sqsubseteq) \text{ e.p.o. } \wedge X \subseteq P \Rightarrow \bigsqcup X \text{ unique}]$

Théorème 2 : $(\forall x)(\exists y)[(P,\sqsubseteq) \text{ e.p.o. } \wedge x \in X \subseteq P \wedge y \in Y \subseteq P \wedge x \sqsubseteq y \Rightarrow \bigsqcup X \sqsubseteq \bigsqcup Y]$

Théorème 3 : $[(P,\sqsubseteq) \text{ e.p.o. } \wedge X \subseteq P \wedge Y \subseteq P \wedge X \subseteq Y \Rightarrow \bigsqcup X \sqsubseteq \bigsqcup Y]$

Théorème 4 : $[(P,\sqsubseteq) \text{ e.p.o. } \wedge R \subseteq \mathcal{P}(P) \Rightarrow \bigsqcup(\bigcup_{X \in R} X) = \bigsqcup\{\bigsqcup X \mid X \in R\}]$

Démonstration :

$(\forall y)[y \in \bigcup_{X \in R} X \Rightarrow (\exists X)(X \in R \wedge y \sqsubseteq \sqcup X)]$

par conséquent (théorème 2) :

$$\sqcup(\bigcup_{X \in R} X) \sqsubseteq \sqcup\{\sqcup X \mid X \in R\}$$

$(\forall X)[X \in R \Rightarrow \sqcup X \sqsubseteq \sqcup(\bigcup_{X \in R} X)]$ (théorème 3)

par la minimalité de la borne supérieure (définition-théorème 2), on a que :

$$\sqcup\{\sqcup X \mid X \in R\} \sqsubseteq \sqcup(\bigcup_{X \in R} X)$$

cqfd

1.2. Dirigé. La notion d'Approximation

L'idée d'approximation que nous avons mentionné dans l'introduction est représentée par les ensembles dirigés. Dans ce paragraphe, nous donnerons sa définition et quelques notions qui dérivent de ce concept.

1.2.1. Ensemble dirigé ([GrG 78],[StJ 77])

Soit $(X,\sqsubseteq)$ un e.p.o. (non vide), X est un ensemble dirigé ssi

$$(\forall Y)(\exists y)[Y \text{ fini} \wedge Y \subseteq X \wedge y \in X \Rightarrow Y \sqsubseteq y]$$

autrement dit, un ensemble est dirigé quand tout sous-ensemble fini de l'ensemble possède un majorant dans cet ensemble.

Exemple 1 : toute chaîne est un ensemble dirigé.

Exemple 2 : dans l'e.p.o. $(P(X),\subseteq)$, $P(X)$ est un dirigé pour n'importe quel ensemble X ; $P_f(X) = \{Y \mid Y \text{ fini} \wedge Y \subseteq X\} \subseteq P(X)$ est aussi un dirigé.

La notion d'approximation est exprimée de la manière suivante : soient $(P,\sqsubseteq)$ un e.p.o., $X \subseteq P$ (X dirigé), et $z \in P$; on dit que le dirigé X est une approximation de z, s'il existe $\sqcup X$ et de plus $z \sqsubseteq \sqcup X$.

Exemple 3 : $P_f(X)$ dans l'ex. 2 est une approximation de X, puisque $X = \sqcup P_f(X) = \bigcup_{Y \in P_f(X)} Y$.

1.2.2. Elément non isolé, ou accessible, ou limite

Une autre notion intéressante est l'approximation exacte, que l'on donne à travers le terme d'élément non-isolé ou accessible ; c'est la suivante :

Soient $(P,\sqsubseteq)$ un e.p.o. et $x \in P$,

x est non isolé $\iff (\exists X)(\exists \sqcup X)[X \subseteq P \wedge X \text{ dirigé} \wedge x = \sqcup X \wedge \sqcup X \notin X]$.

Ce type de dirigé, où la borne supérieure n'appartient pas au dirigé s'appelle dirigé intéressant [StJ 77] ; cela signifie qu'une approximation a une limite qui est à l'extérieur d'elle-même : l'élément accessible. Un dirigé intéressant ne peut être fini, puisque : X dirigé $\wedge$ X fini $\Rightarrow \sqcup X \in X$.

Exemple : Dans les rationnels avec l'ordre habituel, $(Q,\leq)$ est un e.p.o. où tous les éléments distincts de zéro sont accessibles ; un dirigé intéressant pour chaque rationnel q est :

$$\{0, \frac{1}{2}q, \frac{2}{3}q, \ldots \frac{n}{n+1}q, \ldots\}$$

1.2.3. Lemme sur les dirigés

Le lemme que nous allons présenter est un résultat de T. Iwamura [IwT 44] ; ce lemme garantit la décomposition d'un dirigé infini en une chaîne de dirigés de cardinalité plus petite que le premier. Son énoncé est la suivant ([GrG 78], [MaRo 76]) :

$$\begin{aligned} &X \text{ dirigé infini} \wedge |X| = \alpha \\ &\quad \Rightarrow (\exists R)\ X = \bigcup_{Y \in R} Y \\ &\quad \wedge\ R = \{X_\gamma \mid X_\gamma \text{ dirigé} \wedge \gamma < \alpha\} \\ &\quad \wedge\ (\gamma < \delta < \alpha \Rightarrow X_\gamma \subseteq X_\delta) \\ &\quad \wedge\ (\gamma < \alpha \Rightarrow |X_\gamma| < \alpha) \end{aligned}$$

Nous ne donnons pas ici la démonstration de ce lemme, que l'on peut trouver en [MaG 76] pp. 54-55. Ce lemme permet que des propriétés ou définitions qui se basent sur les chaînes, soient aussi valides pour les dirigés, dans certains contextes. L'inverse est toujours valide puisque toute chaîne est un dirigé. Comme exemple d'application de ce lemme, nous aurons le cas de continuité dans les fonctions (§2.1.2., lemme 3).

1.2.4. Notion de partie essentielle ou relativement compacte ([ScD 72], [StJ 77], [MaG 77]).

Nous allons présenter maintenant la notion de partie essentielle (ou relativement compacte) que nous écrivons $x \ll y$ (x est partie essentielle de y ou x est relativement compact à y), qui est liée à l'idée d'approximation de l'information. La définition est la suivante :

Soit $A_y = \{Y | (P,\sqsubseteq)$ e.p.o. $\wedge$ Y dirigé $\wedge$ $Y \subseteq P \wedge y \in P \wedge y \sqsubseteq \sqcup Y\}$

c'est-à-dire, A_y est l'ensemble des approximations de y ;

Soit $x \in P$; alors

$x \ll y \iff (\forall Y)(\exists z)[Y \in A_y \wedge z \in Y \Rightarrow x \sqsubseteq z]$

Cela signifie que, dans toute approximation de l'élément y, il y a un élément qui possède une information égale ou supérieure à x, et donc l'information que contient x est "partie essentielle" de y.

Exemple : $(\mathbb{R},\leq)$ e.p.o. -les réels avec l'ordre habituel- ;

on a : $x \ll y \iff x \leq y \wedge x \neq y, \forall x,y \in \mathbb{R}$

1.2.5. Elément isolé, ou compact, ou fini

La notion (quasi-) inverse à celle d'éléments non-isolé ou accessible (§1.2.2.) est la suivante :

x est isolé (ou compact, ou fini) $\iff x \ll x$

Exemple 1 : Dans $(\mathbb{N},\leq)$, tous les éléments sont compacts.

Exemple 2 : Dans $(\mathbb{R},\leq)$, il n'y a pas d'éléments compacts.

On a : x est isolé $\Rightarrow$ $\neg$(x est accessible) ;
l'inverse n'est pas vrai, sauf dans les treillis continus que nous verrons plus loin (§1.3.4.).

1.2.6. Propriétés de la relation partie essentielle ($\ll$)

Cette notion de relativement compact ou partie essentielle, établit une relation entre les éléments d'un e.p.o. Cette relation satisfait les propriétés suivantes :

Soit $(P,\sqsubseteq)$ un e.p.o.
et soient $x_1,\ldots,x_n, y,z \in P$, on a :

a. $x \ll y \Rightarrow x \sqsubseteq y$
b. $x \ll y \wedge y \sqsubseteq z \Rightarrow x \ll z$
c. $x \sqsubseteq y \wedge y \ll z \Rightarrow x \ll z$
d. $x \ll y \wedge y \ll z \Rightarrow x \ll z$
e. $y = \sqcup\{x_1,\ldots,x_n\} \wedge x_1 \ll z \wedge \ldots \wedge x_n \ll z \Rightarrow y \ll z$

Nous ne donnons pas ici la démonstration de ces propriétés.

Pour terminer cette présentation sur les dirigés, remarquons qu'un dirigé est un ensemble non vide, et que tout ensemble d'un seul élément dans un e.p.o. est un dirigé.

1.3. Treillis

Jusqu'à présent, plusieurs définitions et propriétés que l'on a données dépendaient de l'existence de bornes supérieures ou inférieures dans la structure de l'e.p.o. Les treillis sont des e.p.o. qui garantissent l'existence de ces bornes.

1.3.1. Treillis (simple)

Soit $(P,\sqsubseteq)$ un e.p.o.,

P est un treillis $\Leftrightarrow (\forall X)[X \subseteq P \wedge X \text{ fini} \Rightarrow \bigsqcup X \in P \wedge \bigsqcap X \in P]$

Exemple 1 : Les entiers naturels et les rationnels avec l'ordre canonique sont des treillis.

Exemple 2 : L'e.p.o. de l' exemple en §1.1.4. n'est pas un treillis.

1.3.1.1. Les treillis comme algèbres

Etant donné que, dans un treillis, la borne supérieure et la borne inférieure existent pour tout sous-ensemble fini, on définit deux opérations binaires (totales) $-\cup$ et $\cap$ (union et intersection)- dans l'ensemble qui possède la structure de treillis, comme suit :

Soit P un treillis,

$\forall x,y \in P,\ x \cup y = \bigsqcup\{x,y\}$	union
$x \cap y = \bigsqcap\{x,y\}$	intersection

De cette façon, on a défini une algèbre dans l'ensemble des éléments du treillis, où les opérations sont l'union et l'intersection.

1.3.1.2. Propriétés des opérations d'union et d'intersection

Soit P un treillis, $\forall x,y,z \in P$,

a. $x \cup x = x$	idempotence
b. $x \cup y = y \cup x$	commutativité
c. $x \cup (y \cup z) = (x \cup y) \cup z$	associativité
d. $x \cup (x \cap y) = x$	absorption

Ces propriétés sont aussi valides quand on échange $\cup$ et $\cap$.

1.3.2. Treillis complets

Quand la borne supérieure et inférieure existent pour tout sous-ensemble dans un e.p.o., on dit que l'e.p.o. est un treillis complet ;

Soit $(P,\sqsubseteq)$ un e.p.o.

P est un treillis complet $\Leftrightarrow (\forall X)[X \subseteq P \Rightarrow \bigsqcup X \in P \wedge \bigsqcap X \in P]$

En particulier, on a $\bigsqcup P$ et $\bigsqcap P$, qui sont dénotés respectivement par $\top$ ("top") et $\bot$ ("bottom").

<u>Exemple</u> 1 : si l'on ajoute ∞ (comme $\top$) aux entiers naturels, on obtient un treillis complet avec l'ordre canonique.

<u>Exemple</u> 2 : les rationnels ne sont pas un treillis complet ; il manque les bornes supérieures des approximations des irrationnels ; c'est-à-dire, les bornes sont les irrationnels.

Nous identifierons un treillis complet P de la façon suivante :

$P(\sqsubseteq,\bot,\top,\bigsqcup,\bigsqcap)$.

Le théorème suivant exprime qu'un treillis complet est caractérisé par l'existence de la borne supérieure.

<u>Théorème</u> :

Soit $(P,\sqsubseteq)$ e.p.o.

$(\forall X)[X \subseteq P \wedge \bigsqcup X \in P \Rightarrow P$ est un treillis complet$]$

Démonstration :

Soit $Y = \{y | y \sqsubseteq X\}$ (l'ensemble des minorants (§1.1.3) de X)

et soit $Z = \{z | Y \sqsubseteq z\}$ (l'ensemble des majorants (§1.1.3) de Y)

par la définition de la borne supérieure (§1.1.4), on a que :

$\bigsqcup Y \sqsubseteq Z \wedge Y \sqsubseteq \bigsqcup Y$

mais,

$X \subseteq Z \Rightarrow \bigsqcup Y \sqsubseteq X$

donc $\bigsqcup Y = \bigsqcap X$

en particulier $\bigsqcup \emptyset = \bot$ cqfd

1.3.3. Sous-treillis (complets)

Un sous-treillis est un sous-ensemble d'un treillis complet qui est à son tour un treillis complet, avec la même relation d'ordre : c'est-à-dire, si

$P(\sqsubseteq,\bot,\top,\sqcup,\sqcap)$, $S(\sqsubseteq,\bot',\top',\sqcup',\sqcap')$ et $S \subseteq P$, alors S est un sous-treillis (de P). Les primes dans le treillis complet S indiquent que les bornes (en S) des sous-ensembles de S peuvent être distinctes des bornes des mêmes sous-ensembles en P. Plus précisément, la relation entre les bornes de P et S (en P) est la suivante :

- $\bot \sqsubseteq \bot' \wedge \top' \sqsubseteq \top$
- $(\forall X)[X \subseteq S \subseteq P \Rightarrow \sqcup X \sqsubseteq \sqcup' X \wedge \sqcap' X \sqsubseteq \sqcap X]$

Exemple : $P_\omega(\subseteq,\emptyset,\mathbb{N},\sqcup,\sqcap)$

- P_ω est l'ensemble des parties des naturels ($P(\mathbb{N})$)
- $\subseteq$ est l'inclusion comme relation d'ordre partiel
- $\emptyset$ l'ensemble vide est $\bot$
- $\mathbb{N}$ l'ensemble des entiers naturels est $\top$
- $\sqcup$ $(\forall X)[X \subseteq P(\mathbb{N}) \Rightarrow \sqcup X = \bigcup_{x \in X} x]$
- $\sqcap$ $(\forall X)[X \subseteq P(\mathbb{N}) \Rightarrow \sqcap X = \bigcap_{x \in X} x]$

Si on enlève les ensembles infinis d'entiers en P_ω, sauf l'élément $\top$, on obtient un treillis complet $P_{\omega f}(\subseteq,\emptyset,\mathbb{N},\sqcup',\sqcap)$, où

- $\sqcup'$ $(\forall X)[X \subseteq P_{\omega f} \wedge X \text{ fini} \Rightarrow \sqcup' X = \bigcup_{x \in X} x]$
  $(\forall X)[X \subseteq P_{\omega f} \wedge X \text{ infini} \Rightarrow \sqcup' X = \top]$

On a alors que $P_{\omega f}$ est un sous-treillis de P_ω, et pour quelques cas dans les sous-ensembles infinis de $P_{\omega f}$, la borne supérieure n'est pas la même dans les deux treillis. Prenons par exemple :

$$X \subseteq P_{\omega f} \subseteq P_\omega$$

$$\text{et } X = \{X_f \mid X_f \text{ fini} \wedge [x \in X_f \Leftrightarrow x \text{ est pair}]\}$$

on a :

$$\sqcup' X = \top$$

$$\text{et } \sqcup X = \{x \mid x \text{ est pair}\}$$

Prenant en compte cette différence entre les bornes d'un treillis et celles du sous-treillis, on note trois cas ([SaL 77], [McN 79]) :

Soient $R(\sqsubseteq,\bot,\top,\sqcup,\sqcap)$, $S(\sqsubseteq,\bot',\top',\sqcup',\sqcap')$, $S \subseteq R$

a) S est <u>$\sqcup$-fermé</u> $\Leftrightarrow (\forall X)[X \subseteq S \Rightarrow \sqcup' X = \sqcup X]$

b) S est <u>$\sqcap$-fermé</u> $\Leftrightarrow (\forall X)[X \subseteq S \Rightarrow \sqcap' X = \sqcap X]$

c) S est de <u>caractère fini</u> $\Leftrightarrow (\forall X)[X \subseteq S \wedge X \text{ dirigé} \Rightarrow \sqcup' X = \sqcup X]$

Dans l'exemple, $P_{\omega f}$ est $\sqcap$-fermé.

1.3.4. Treillis continu

Soit le treillis complet $C(\sqsubseteq, \bot, \top, \bigsqcup, \bigsqcap)$,

C est continu $\Longleftrightarrow (\forall y)\ y \in C \Rightarrow y = \bigsqcup\{x \mid x \in C \wedge x \ll y\}$

Cette définition garantit que tout élément d'un treillis continu est un élément accessible (ou limite) ou est un élément compact (ou isolé), formellement :

Soit $C(\sqsubseteq, \bot, \top, \bigsqcup, \bigsqcap)$ continu

et soit $RC(y) = \{x \mid x \in C \wedge x \ll y\} \quad (\forall y \in C)$

on a : $RC(y)$ dirigé d'après §1.2.6.e.,

$y \in RC(y) \Longleftrightarrow y$ compact,

$y \notin RC(y) \Longleftrightarrow y$ accessible (ou limite)

Exemple 1 : $\mathbb{R}^+ (\leq, 0, \top, \bigsqcup, \bigsqcap)$ où $\mathbb{R}^+ = \{r \mid r \in \mathbb{R} \wedge r \geq 0\} \cup \top$ est un treillis continu ; le seul élément compact est 0.

Exemple 2 : $P_{\omega f}$ tel que défini au §1.3.3. -exemple, est un treillis complet qui n'est pas continu, puisque l'unique élément accessible est $\top$, et l'unique élément compact est $\bot$. Les autres éléments, c'est-à-dire les ensembles finis non vides de naturels (en $P_{\omega f}$) ne sont ni accessibles ni compacts). Il est clair que ces éléments ne peuvent pas être accessibles (limites), puisque tout dirigé tel que la borne supérieure soit un de ses éléments est un dirigé fini, et par conséquent, la borne supérieure appartient au dirigé (non intéressant) ; ils ne sont pas des éléments compacts (ou isolés) puisque :

$(\forall x)\ x \in P_{\omega f} \wedge x \neq \bot \wedge x \neq \top$,

définissons

$\bar{X} = \{y \mid y \in P_{\omega f} \wedge y \neq \top \wedge y \cap x = \emptyset\}$

alors :

$\bar{X}$ dirigé $\wedge\ x \sqsubseteq \bigsqcup \bar{X} = \top \wedge x \notin \bar{X}$

1.3.5. Treillis algébrique

Soit le treillis complet $A\ (\sqsubseteq, \bot, \top, \bigsqcup, \bigsqcap)$,

A est algébrique $\Longleftrightarrow (\forall y)[y \in A \Rightarrow y = \bigsqcup\{x \mid x \in A \wedge x \ll x \wedge x \sqsubseteq y\}]$

Cela signifie que tout élément d'un treillis algébrique est la borne supérieure des éléments compacts qui sont inférieurs à cet élément ; par conséquent, on dira que les éléments compacts sont "denses" dans un treillis algébrique.

La définition de treillis algébrique satisfait la définition de treillis continu, puisque : $x \ll x \wedge x \sqsubseteq y \Rightarrow x \ll y$. Mais on peut donner une condition suffisante sur les treillis continus pour qu'ils soient algébriques [Sal 77], cette condition est que la relation de partie essentielle ($\ll$) soit bien fondée, c'est-à-dire

$$(\nexists \{x_i\}_{i \in \mathbb{N}})[x_{i+1} \ll x_i \wedge x_{i+1} \neq x_i]$$

Lemme 1 : $C(\sqsubseteq, \bot, \top, \sqcup, \sqcap)$ continu $\wedge \ll$ bien fondée
$\Rightarrow C(\sqsubseteq, \bot, \top, \sqcup, \sqcap)$ algébrique

Démonstration :

Supposons que C ne soit pas algébrique et soit $\bar{G}$ l'ensemble des éléments de C qui ne sont pas engendrés par des compacts, c'est-à-dire,

$\bar{G} = \{y \mid y \in C \wedge y \neq \{x \mid x \in C \wedge x \ll x \wedge x \sqsubseteq y\}\}$

Soit y' un élément minimal en $\bar{G}$ ($\ll$ bien fondé)

$y' = \sqcup RC(y') \wedge RC(y') = \{x' \mid x' \in C \wedge x' \ll y'\}$ (C continu)

$y' = \sqcup R' \wedge R' = \bigcup_{x' \in RC(y')} RC(x')$ (T.4, §1.1.5)

Soit $R = \{v \mid v = z \cup z' \wedge z \in R' \wedge z' \in R'\} \cup R'$, R est dirigé

et $y' = \sqcup R$ (§1.2.6.e)

$y' \in \bar{G} \Rightarrow (\exists v')[v' \in R' \wedge \neg(v \ll v)]$

$v' \in R' \Rightarrow v' \in RC(\underline{x}') \wedge \underline{x}' \in \bar{G}$ (pour quelque $\underline{x}'$)

mais $\underline{x}' \sqsubseteq y' \wedge \underline{x}' \neq y'$

donc y' n'est pas minimal, et $\ll$ n'est pas bien fondée. cqfd

Exemple : $P_\omega(\sqsubseteq, \emptyset, \mathbb{N}, \sqcup, \sqcap)$ est un treillis algébrique, avec :

A) x est compact (ou isolé, ou fini) si x est un ensemble fini de naturels ; en effet :

Soit $x \in P_\omega$, x un ensemble fini de naturels et soit $X \subseteq P_\omega$, X dirigé, $x \sqsubseteq \sqcup X = \bigcup_{y \in X} y$.

On a deux cas :

1) X fini $\Rightarrow \sqcup X \in X$,
donc il existe $y \in X$ (c.a.d. $y = \sqcup X$) et $x \sqsubseteq y$

2) X infini $\wedge$ x fini (c.a.d. $|x| = n < \omega$) $\wedge\ x \subseteq \bigcup_{y \in X} y$
$\Rightarrow (\exists y_i)(i=1,\dots,n)[y_i \in X \wedge x \subseteq \sqcup\{y_1,\dots,y_n\}]$
$\wedge$ X dirigé $\Rightarrow (\exists y')[y' \in X \wedge \sqcup\{y_1,\dots,y_n\} \subseteq y']$
donc il existe $y' \in X$ et $x \sqsubseteq y'$;
alors x est compact ($x \ll x$).

B) x est accessible (ou non isolé, ou limité) si x est un ensemble infini de naturels ; en effet :

Soit $x \in P_\omega$, un ensemble infini de naturels

et soit $X = \{y \mid y \text{ est fini} \wedge y \sqsubseteq x\} = \{y \mid y \ll y \wedge y \sqsubseteq x\}$,

on a :

$$X \text{ dirigé} \wedge x = \bigsqcup X \wedge \bigsqcup X \notin X$$

donc, x est accessible.

Nous appellerons les treillis algébriques <u>domaines</u> ou <u>domaines à caractère fini</u>, puisqu'ils sont "engendrés" par les éléments finis (ou compacts).

Terminons ce paragraphe avec le lemme suivant :

<u>Lemme</u> 2 :

Soient, . $A(\sqsubseteq, \bot, \top, \sqcup, \sqcap)$ domaine (treillis algébrique) ;

. $A' = \{x \mid x \in A \wedge x \ll x\}$ les éléments compacts de A ;

. $F_x = \{y \mid y \in A \wedge x \in A' \wedge x \sqsubseteq y\}$ l'ensemble des éléments plus grands ou égaux au compact x.

Alors : (1) F_x est un domaine (treillis algébrique)

(2) $(\forall A'')[A'' \subseteq A \Rightarrow \{\bot\} \cup (\bigcup_{x \in A''} F_x)$ est aussi un domaine (treillis algébrique)$]$

démonstration : (nous donnons seulement la démonstration pour (1) $-F_x-$, puisque celle pour (2) est identique).

Soit $y \in F_x \subseteq A$

alors $y = \bigsqcup\{z \mid z \in A' \wedge z \sqsubseteq y\}$ (puisque A est un domaine)

et $y = y \cup x = \bigsqcup\{\bigsqcup\{z \mid z \in A' \wedge z \sqsubseteq y\}, x\}$ (puisque $x \sqsubseteq y$)

$y = \bigsqcup(\{z \mid z \in A' \wedge z \sqsubseteq y\} \cup \{x\})$ (T.4, §1.1.5)

$y = \bigsqcup(\bigcup_{z \in A' \wedge z \sqsubseteq y} \{z, x\})$

$y = \bigsqcup\{\bigsqcup\{z, x\} \mid z \in A' \wedge z \sqsubseteq y\}$ (T 4, §1.1.5)

mais $\bigsqcup\{z, x\} = z \cup x$ est compact (puisque $x \ll x \wedge z \ll z \Rightarrow x \cup z \ll x \cup z$ de §1.2.6, c et e)

de plus $\bigsqcup\{z, x\} \in F_x$ (puisque $x \sqsubseteq z \cup x = \bigsqcup\{z, x\}$) cqfd.

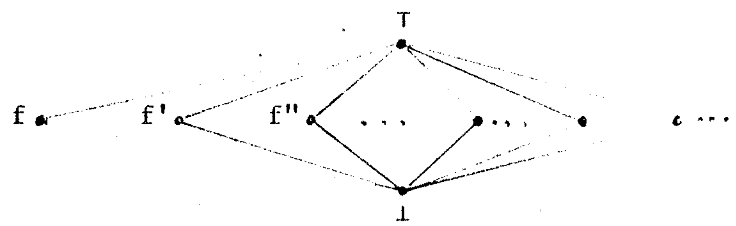
1.3.6. <u>Treillis plat</u>

Un type de domaine (ou treillis algébrique) "simplifié" est le treillis plat, qui est défini de la façon suivante :

$F(\sqsubseteq, \bot, \top, \sqcup, \sqcap)$ est un <u>treillis plat</u>

$\Leftrightarrow (\forall x)(\forall y)[x \in F \wedge y \in F \wedge x \neq y \wedge x \neq \bot \wedge x \neq \top \wedge y \neq \bot \wedge y \neq \top \Rightarrow x \not\sqsubseteq y \wedge y \not\sqsubseteq x]$

c'est-à-dire, tous les éléments distincts (et différents de $\perp$ et $\top$) sont incomparables ; on obtient le graphique :



$(\perp,\top,f,f',\ldots\in F)$

On peut observer que dans un treillis plat, tous les éléments sont compacts. C'est une façon de transformer tout ensemble F en un treillis algébrique : il suffit d'ajouter $\perp$ et $\top$; aussi on appellera <u>$\perp$ et $\top$ éléments impropres</u>. Il est à noter que le treillis plat F est un sous-treillis du treillis -algébrique lui aussi- $P(F)$; de plus, il est $\sqcap$-fermé.

1.4. Base - La notion d'élément calculable

Nous donnons maintenant le concept de génération d'un treillis à partir d'un sous-ensemble de celui-ci.

1.4.1. Base dans un treillis complet

Soit $P(\sqsubseteq,\perp,\top,\sqcup,\sqcap)$ et $B\subseteq P$; on dit que <u>B est une base du treillis complet P</u> ssi :

$$(\forall x)[x\in P \Rightarrow \quad (1)\ (\exists B')[B'\subseteq B \wedge x = \sqcup B']$$
$$\quad (2)\ (\forall b)(\forall b')[b\in B \wedge b'\in B \Rightarrow b\cup b'\in B]]$$

si la condition (2) n'est pas satisfaite, on dira que <u>B est une sous-base de P</u>. Comme conséquence de ces conditions, on déduit le lemme suivant :

<u>Lemme</u> : B base $\wedge\ B' = \{b \mid b\in B \wedge b \sqsubseteq x\} \Rightarrow B'$ dirigé et $x = \sqcup B'$

Exemple : en $P_{\omega f}$, une base est l'ensemble $P_{\omega f}$ moins l'élément $\top$, et une sous-base est constituée des ensembles d'un seul élément et du vide ($\perp$).

1.4.2. Base dans un treillis continu

Si le treillis est continu et possède une base, on a le lemme suivant :

<u>Lemme</u> : $C(\sqsubseteq,\perp,\top,\sqcup,\sqcap)$ continu $\wedge\ B\subseteq C \wedge B$ base
$\Rightarrow (\forall x)[x\in C \Rightarrow x = \sqcup\{b \mid b\in B \wedge b \ll x\}]$

Démonstration :

C continu $\Rightarrow \forall x\in C,\ x = \sqcup\{y \mid y\in C \wedge y \ll x\}$

et B base => $\forall x \in C$, $x = \sqcup\{\sqcup\{b \mid b\in B \wedge b \sqsubseteq y\} \mid y \ll x\}$;

par le théorème 4 (§1.1.5), on a :

$$\begin{aligned} x &= \sqcup\{\sqcup\{b \mid b\in B \wedge b \sqsubseteq y\} \mid y \ll x\} \\ &= \sqcup(\bigcup_{y \ll x}\{b \mid b \sqsubseteq y\}) \\ &= \sqcup\{b \mid b \sqsubseteq y \ll x\} \end{aligned}$$

par les théorèmes 3 et 1 (§1.1.5),

$$x = \sqcup\{b \mid b \ll x\}$$ cqfd

Cela signifie que tout élément $x\in C$ est la borne supérieure des éléments de la base $B\subseteq C$ qui sont partie essentielle de x.

Exemple : les rationnels positifs ($\mathbb{Q}^+ = \{q \mid q\in\mathbb{Q} \wedge q\geq 0\}$) sont une base du treillis continu des réels positifs ($\mathbb{R}^+ = \{r \mid r\in\mathbb{R} \wedge r\geq 0\} \cup \top$)

1.4.3. Base dans les treillis algébriques

Dans la définition du treillis algébrique (§1.3.5), tout élément du treillis est la borne supérieure des compacts inférieurs à cet élément ; cela signifie que l'ensemble des compacts est une base du treillis algébrique. Le lemme suivant nous dit que cette base est minimale.

Lemme : Soient $A(\sqsubseteq,\bot,\top,\sqcup,\sqcap)$ algébrique,
et $C = \{x \mid x\in A \wedge x \ll x\}$
et B une base de A,
alors $C\subseteq B$

Démonstration :

$(\forall x)$ $x\in A \Rightarrow x = \sqcup B_x \wedge B_x = \{b \mid b\in B \wedge b \ll x\}$ dirigé

$(\forall x)$ $x\in C \Rightarrow (\exists b')[b'\in B_x \wedge x \sqsubseteq b']$ (puisque $x \ll x$)

$x = \sqcup B_x \Rightarrow x = b' \in B_x \subseteq B$ cqfd

1.4.4. Bases récursives ([VuJ 74])

On dit qu'une base B d'un treillis $P(\sqsubseteq,\bot,\top,\sqcup,\sqcap)$ est une base récursive ssi :

(1) B est en bijection avec un sous-ensemble récursif I ($I\subseteq\mathbb{N}$) des entiers naturels ;

et (2) la relation ternaire $J \subseteq I \times I \times I$, définie de la manière suivante :

$(n,m,k) \in J \iff b_n \cup b_m = b_k \wedge b_n, b_m, b_k \in B$

est aussi en bijection avec un sous-ensemble récursif des entiers naturels.

De cette façon, on a imposé une énumération (un ensemble récursif d'indices) aux éléments de la base ; c'est-à-dire :
$B = \{b_i \mid b_i \in P \wedge i \in I \wedge I \text{ récursif}\}$ est la base. De plus, le problème de décider si $b_i \sqsubseteq b_j$ ou $b_i \cup b_j = b_k$ est décidable ; dans le cas où les ensembles I et J sont récursivement énumérables, ce problème de décision est semi-décidable ([DaM 58], [RoH 67]).

Exemple : $\mathbb{Q}^+$ dans $\mathbb{R}^+$ est une base énumérable, en prenant par exemple la codification de Cantor (c.a.d. : $q = n/m = r_i$, $i = \frac{1}{2}(m+n)(m+n+1)+m$)

1.4.5. Elément calculable dans un treillis

On associe aussi à tout élément d'un treillis complet avec base récursive, un ensemble d'indices :

Soient $P(\sqsubseteq, \top, \bot, \sqcup, \sqcap)$, B base récursive de P et $I \subseteq \mathbb{N}$ l'ensemble récursif associé à B,

et $(\forall x \in P)\ B_x = \{b \mid b \in B_x \wedge b \sqsubseteq x\}\ (x = \sqcup B_x)$;

on dira que I_x est l'ensemble d'indices associé à x ssi

$(\forall i)\ [i \in I_x \subseteq I \iff b_i \in B_x]$

La notion d'élément calculable est la suivante :

$(\forall x \in P)[\underline{x \text{ est calculable}} \iff I_x \text{ est récursivement énumérable}]$

1.5. Domaines de Calcul

Les ensembles qui nous intéressent, et que nous utiliserons dans le reste du travail, seront appelés domaines de calcul. En voici la définition.

1.5.1. Définition

Un ensemble D est un domaine de calcul, s'il satisfait les conditions suivantes :

(1) D est un treillis algébrique avec base récursive

(2) l'ensemble des compacts de D constitue sa base récursive.

<u>Note</u> : dans tout treillis algébrique (continu ou complet), l'élément $\perp$ est compact ; cependant, on ne considèrera pas cet élément dans la base.

<u>Exemple</u> : P_ω est un domaine de calcul, et on peut affecter aux éléments de la base (les ensembles finis de naturels) l'indice calculé de la manière suivante :

$(\forall x \in P_\omega)[x \text{ fini (ou compact)} \Rightarrow x = b_i \wedge i = \sum_{j \in x} 2^j]$

1.5.2. Interprétation topologique

Maintenant, nous présentons les considérations topologiques (topologie de Scott) que l'on peut faire sur les domaines de calcul. Avant de donner une série de définitions de la topologie, nous donnons la notion de filtre associé à l'élément x dans un domaine :

Soient D un domaine, $x \in D$ un élément et $F_x \subseteq D$,
on dit
F_x <u>filtre associé à x</u> $\Leftrightarrow F_x = \{y \mid y \in D \wedge x \sqsubseteq y\}$
et on a la propriété suivante,
$z \in D \wedge y \in F_x \Rightarrow z \sqcup y \in F_x$
(le filtre est la notion duale de <u>idéal</u> dans les treillis).

Un <u>espace topologique</u> est une paire (S,T), où S est un ensemble quelconque, $T \subseteq P(S)$ une topologie sur S, et où les conditions suivantes sont satisfaites :

(1) $S \in T \subseteq P(S)$
(2) $(\forall O_i)[O_i \in T \wedge i \in I \wedge I \text{ fini} \Rightarrow \bigcap_{i \in I} O_i \in T]$
(3) $(\forall O_i)[O_i \in T \wedge i \in I \Rightarrow \bigcup_{i \in I} O_i \in T]$

C'est-à-dire, T est fermé pour les intersections finies et pour les unions quelconques. Les membres de T s'appellent <u>ouverts</u>. Le complément (par rapport à S) d'un ouvert s'appelle un <u>fermé</u>. Une <u>base</u> $B \subseteq T$ dans un espace topologique est une famille d'ensembles non vides, et telle que tout ouvert est l'union de membres de la base. Un sous-ensemble $C (C \subseteq S)$ est <u>compact</u>, ssi

$C \subseteq \bigcup_{i \in I} O_i \wedge O_i \in T \Rightarrow C \subseteq \bigcup_{i \in I'} O_i \wedge I' \subseteq I \wedge I' \text{ fini}.$

On définit la topologie "induite" dans l'ensemble D par la structure du domaine de calcul D, en donnant les ouverts qui forment une base de cette topologie induite.

Soit le triplet (D, T_i, B)
où . D ensemble (dans lequel existe une structure de domaine de calcul)
. $T_i \subseteq P(D)$

. $B \subseteq \mathcal{T}_i$

avec la définition suivante de B,

$$O \in B \iff (\exists x)[x \in D \wedge x \ll x \wedge O = F_x]$$

Autrement dit, les membres de B (base) sont les filtres associés aux éléments compacts de D. De plus, B est en bijection avec $J \subseteq \mathbb{N}$, et J est récursif puisque l'ensemble des compacts (base de D) est récursif.

Les membres de $\mathcal{T}_i$ sont définis par :

$$(\forall O)[O \in \mathcal{T}_i \iff O = \bigcup_{j \in J' \subseteq J} O_j \wedge B = \{O_j\}_{j \in J}]$$

Notons que, par le lemme 2 du §1.3.5, les ouverts de la base (filtres associés aux éléments compacts de D) sont des domaines, et que $\{\bot\} \cup O$ $(O \in \mathcal{T}_i)$ est aussi un domaine.

Il est facile de vérifier que $\mathcal{T}_i$ (la topologie induite) satisfait les conditions (1) à (3) d'une topologie. Ainsi on a que $(D, \mathcal{T}_i)$ est un espace topologique avec une base récursive. De plus, cette topologie satisfait l'axiome de séparation le plus faible, celui qui caractérise les <u>topologies-T_0</u> ; cet axiome est le suivant :

Soit $(S, \mathcal{T})$ un espace topologique ; $\mathcal{T}$ est une topologie-T_0, si en plus des conditions (1) à (3), elle satisfait

(4) $(\forall x)(\forall y)(\exists O)[x \in S \wedge y \in S \wedge x \neq y \wedge O \in \mathcal{T} \Rightarrow (x \in O \wedge y \notin O) \vee (x \notin O \wedge y \in O)]$

Pour montrer que $\mathcal{T}_i$ satisfait la condition (4), il suffit de prendre le filtre associé à un compact W défini par :

Soient $B_x = \{z \mid z \ll z \sqsubseteq x\}$ et $B_y = \{z \mid z \ll z \sqsubseteq y\}$

alors,

$$W \in (B_x \cup B_y) - (B_x \cap B_y)$$

et on a que :

$$W \in B_x \Rightarrow O \text{ ouvert} \wedge O = F_W \text{ filtre} \wedge x \in O \wedge y \notin O$$
$$\vee\ W \in B_y \Rightarrow O \text{ ouvert} \wedge O = F_W \text{ filtre} \wedge x \notin O \wedge y \in O$$

Etant données les définitions des ouverts en $\mathcal{T}_i$ (ce sont des sous-domaines), on peut donner la caractérisation suivante :

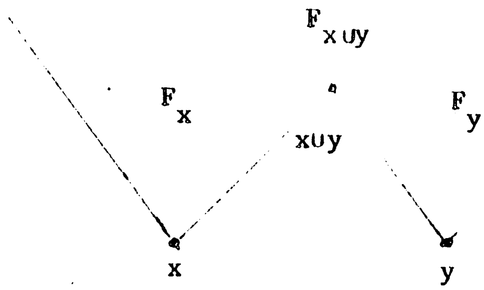
$$(\forall O)[O \in \mathcal{T}_i \iff (\forall x)(\exists y)(x \in O \iff y \in O \wedge y \ll y \sqsubseteq x)]$$

C'est-à-dire, pour tout élément x d'un ouvert, il existe un élément compact (ou fini) plus petit ou égal à x, et qui appartient aussi à l'ouvert. Cela est connu sous le nom de <u>caractère fini des ouverts</u> ([ScD 76]).

Les ouverts $O \in \mathcal{T}_i$ possèdent aussi la propriété de <u>monotonie</u> :

$$(\forall x)(\forall y)[x \sqsubseteq y \wedge x \in O \Rightarrow y \in O].$$

Graphiquement, deux éléments de l'ensemble D et les filtres associés dans l'espace topologique $(D,\mathcal{T}_i)$ sont représentés par :



Notons que : $(\forall x,y\in D)[F_x \cap F_y = F_{x\cup y} \neq \emptyset]$. De plus, tout filtre est un ensemble compact, dans le sens topologique, dans l'ensemble D de l'espace topologique $(D,\mathcal{T}_i)$.

La notion d'élément compact s'exprime en termes du concept de filtre associé à un élément, de la manière suivante :

$x \ll x$ (compact ou fini) $\Leftrightarrow$

$(\forall Y)(\exists Y')[Y \text{ dirigé} \wedge \bigcap_{y\in Y} F_y \subseteq F_x \Rightarrow \bigcap_{y\in Y'\subseteq Y} F_y \subseteq F_x \wedge Y' \text{ fini}]$

1.5.3. Autres domaines

Dans la littérature sur les domaines, on trouve d'autres définitions pour ces structures, habituellement basées sur la notion d'ensemble inductif ou "cpo". La notion d'ensemble inductif est :

I ensemble inductif $\Leftrightarrow$ $(I,\sqsubseteq)$ est un e.p.o.

$(\exists\bot)[\bot = \sqcap I]$

$(\forall C)[C \text{ chaîne} \wedge C \subseteq I \Rightarrow \sqcup C \in I]$

Note : une conséquence du lemme de Iwamura (§1.2.3) est que X dirigé, I cpo, et $X \subseteq I$ implique $\sqcup X \in I$.

Un avantage de travailler avec les ensembles inductifs, est que cela n'exige pas l'existence de l'élément $\top$ ("top") dont l'interprétation est difficile dans quelques cas.

D. Scott en [ScD 80] définit les domaines de caractère fini, comme ensembles inductifs, où les éléments finis (compacts) forment une base récursive infinie, et dans lesquels, pour tout sous-ensemble borné (qui possède des majorants), la borne supérieure existe. Ces domaines sont aussi connus comme domaines algébriques sous conditions [BeG 79]. Une autre définition des domaines à caractère fini (en anglais : finitary domain) est la suivante [BrF 81] :

D domaine à caractère fini $\Leftrightarrow$

$(D,\sqsubseteq)$ e.p.o. non vide

$\wedge$ $E = \{e \mid e\in D \wedge e \ll e\}$

$$\wedge \quad (\forall x)[x \in D \Rightarrow x = \bigsqcup\{e \mid e \in E \wedge e \sqsubseteq x\}]$$
$$\wedge \ (\forall C)(\forall C_f)(\exists y)[C_f \text{ fini} \wedge C_f \subseteq C \subseteq D \wedge y \in D \wedge C_f \sqsubseteq y \Rightarrow \bigsqcup C \in D]$$

Dans cette dernière condition, C représente les sous-ensembles consistants en D (i.e. : tout sous-ensemble fini de C possède un majorant), ce qui exprime que tout ensemble consistant a une borne supérieure. Les autres conditions exprimant que D est un ensemble non vide, partiellement ordonné, et qui possède une base (récursive) des éléments finis. Notons que le fait qu'un ensemble soit consistant est une condition nécessaire pour qu'il soit borné, dirigé, chaîne. Les domaines de calcul (§1.5.1) sont un cas spécial de ces domaines à caractère fini.

Une notion importante, pour la calculabilité, est celle de ω-cpo, qui sont des ensembles inductifs, où la troisième condition ($\bigsqcup C \in I$) est valide pour les chaînes énumérables. Notons que dans les ω-cpo, qui sont une notion plus générale que celle d'ensemble inductif, les dirigés énumérables ont la borne supérieure dans l'ensemble. On définit aussi le concept de ω-continuité dans les fonctions (§2.1.2) avec cette idée d'énumérabilité.

CHAPITRE 2.
LE CONCEPT DE FONCTION (CONTINUE, MONOTONE) DANS LES DOMAINES

2. LE CONCEPT de FONCTION (CONTINUE-MONOTONE) dans les DOMAINES

Les autres "objets mathématiques" qui nous intéressent sont les fonctions entre domaines, et en particulier, les fonctions qui possèdent un certain comportement sur les bornes supérieures des dirigés. Dans ce paragraphe, nous donnons un bref rappel de la notion de fonction, quelques notations, et quelques concepts liés à cette notion.

Une fonction f d'un ensemble D dans un ensemble D', notée $f : D \to D'$, est un sous-ensemble de $D \times D'$ (c.a.d. $f \subseteq D \times D'$) qui vérifie la propriété suivante :

$$(\forall \langle x,y\rangle)(\forall \langle u,v\rangle)[\langle x,y\rangle \in f \wedge \langle u,v\rangle \in f \wedge y \neq v \Rightarrow x \neq u] \qquad (1)$$

Dans le cas où $\langle x,y\rangle \in f$, on dit que y est l'image de x par f, symboliquement : $y = f(x)$.

Une fonction $f : D \to D'$, D et D' ensembles, est <u>totale</u> ssi :

$$(\forall x)(\exists y)[x \in D \wedge y \in D' \wedge y = f(x)] \qquad (2)$$

Quand cette propriété (2) n'est pas satisfaite, on dit que la fonction est partielle (non totale), c'est-à-dire, f "n'est pas définie" en $X \subseteq D$, où :

$$X = \{x \mid x \in D \wedge (\forall y)[y \in D' \wedge \neg(\langle x,y\rangle \in f)]\} \qquad (3)$$

Ici, nous ne traiterons uniquement que des fonctions totales entre domaines sans perte de généralité ; les fonctions partielles sont rendues totales de la manière suivante :

Soit $f : D \to D'$, D et D' domaines

et soit $X \subseteq D$ l'ensemble où f n'est pas définie (3) alors, nous posons $f(x) = \perp'$, pour tout $x \in X$; c'est-à-dire f prend la valeur <u>indéfinie</u> $\perp'$ (bottom) en D', là où elle n'est pas définie.

Généralisant la notion d'image d'un élément, on a :

Soit $f : D \to D'$, D et D' domaines, et soit $X \subseteq D$; l'image de X par f est donnée par :

$$f(X) = \{f(x) \mid x \in X\} \subseteq D'$$

et Image (f) est exprimée par :

$$\mathrm{Image}(f) = f(D) \subseteq D'$$

Une fonction $f : D \to D'$, D et D' domaines, est <u>injective</u> ou <u>1-1</u> ssi :

$$(\forall x)(\forall y)[x \in D \wedge y \in D \wedge x \neq y \Rightarrow f(x) \neq f(y)] \qquad (4)$$

f est <u>surjective</u> ssi :

$$\mathrm{Image}(f) = D' \; ; \qquad (5)$$

Si f est injective et surjective (c.a.d. <u>bijective</u>), il existe une fonction <u>inverse</u> (totale) f^{-1} définie par :

$$f^{-1} : D' \to D \quad f^{-1}(x) = y \ (\text{où } x = f(y)) \tag{6}$$

2.1. Comportement des fonctions sur les bornes supérieures des ensembles dirigés

Nous donnons maintenant la définition de deux classes de fonctions selon la relation qui existe entre l'image de la borne supérieure d'un ensemble dirigé, et la borne supérieure de l'image du dirigé.

2.1.1. Fonctions monotones

Soit $f : D \to D'$, où $D(\sqsubseteq, \bot, \top, \sqcup, \sqcap)$ et $D'(\sqsubseteq', \bot', \top', \sqcup', \sqcap')$ sont des treillis complets. On a :

<u>f monotone</u> $\iff (\forall X)[X \subseteq D \wedge X \text{ dirigé} \Rightarrow \sqcup' f(X) \sqsubseteq' f(\sqcup X)]$

<u>Lemme 1</u> : f monotone $\iff (\forall x,y \in D)[x \sqsubseteq y \Rightarrow f(x) \sqsubseteq' f(y)]$

Démonstration :

$(\Rightarrow)$ $x \sqsubseteq y \Rightarrow \{x,y\}$ dirigé et $\sqcup\{x,y\} = y$

$\wedge\ \sqcup' f(\{x,y\}) = \sqcup'\{f(x),f(y)\} \sqsubseteq' f(\sqcup\{x,y\}) = f(y) \Rightarrow f(x) \sqsubseteq' f(y)$

$(\Leftarrow)$ $(\forall x)[x \in X \Rightarrow x \sqsubseteq \sqcup X] \Rightarrow (\forall x \in X)[(f(x) \sqsubseteq' f(\sqcup X)]$

$\Rightarrow \sqcup' f(X) \sqsubseteq' f(\sqcup X)$ cqfd

Notons que la partie $(\Leftarrow)$ de la preuve n'a pas exigé que X soit dirigé, c'est-à-dire : f monotone $\Rightarrow (\forall X)[X \subseteq D \Rightarrow \sqcup' f(X) \sqsubseteq' f(\sqcup X)]$

<u>Lemme 2</u> : f monotone $\wedge$ X dirigé $\Rightarrow$ f(X) dirigé

c'est-à-dire, l'image d'un dirigé par une fonction monotone est un dirigé.

Exemple : Soit $f : \mathcal{P}\omega \to \mathcal{P}\omega$ définie par :

$$f(x) = \begin{cases} x & \text{si } x \text{ est fini} \\ \top & \text{si } x \text{ est infini} \end{cases}, \qquad \text{Image}(f) = \mathcal{P}\omega_f$$

Parmi les fonctions monotones, on trouve les fonctions strictement momotones et les isomorphismes :

2.1.1.1. Fonction strictement monotone

Soit $f : D \to D'$, D et D' treillis complets, alors :

f <u>strictement monotone</u> <=> $(\forall x,y \in D)[x \sqsubseteq y \iff f(x) \sqsubseteq' f(y)]$

<u>Lemme</u> : f strictement monotone => f injective

Démonstration : $f(x) = f(y) \Rightarrow f(x) \sqsubseteq' f(y) \wedge f(y) \sqsubseteq' f(x) \Rightarrow x \sqsubseteq y \wedge y \sqsubseteq x$
$\Rightarrow x = y$ cqfd

2.1.1.2. Isomorphisme

Soit $f : D \to D'$, D et D' treillis complets, alors :

f <u>isomorphisme</u> <=> f strictement monotone et surjective

<u>Lemme</u> : f isomorphisme <=> f^{-1} isomorphisme
(où f^{-1} dénote l'isomorphisme inverse de f).

Dans ce qui suit, nous étudierons une autre classe de fonctions monotones entre domaines : les fonctions continues.

2.1.2. Fonctions continues

Soit $f : D \to D'$ où $D(\sqsubseteq,\bot,\top,\sqcup,\sqcap)$ et $D'(\sqsubseteq',\bot',\top',\sqcup',\sqcap')$ sont des treillis complets, on a :

f <u>continue</u> <=> $(\forall X)[X \subseteq D \wedge X \text{ dirigé} \Rightarrow \sqcup' f(X) = f(\sqcup X)]$

<u>Lemme 1</u> : f continue => f monotone.

<u>Lemme 2</u> : f monotone $\wedge$ $(\forall X)[X \text{ dirigé} \Rightarrow \sqcup X \in X]$ => f continue.

Le lemme 1 nous indique que les fonctions continues sont un sous-ensemble des fonctions monotones ; et le lemme 2, que les fonctions continues se différencient des monotones par leur comportement aux limites (§1.2.2).

Le concept de continuité d'une fonction peut être fondé sur les chaînes au lieu des dirigés :

Soit $f : D \to D'$, où $D(\sqsubseteq,\bot,\top,\sqcup,\sqcap)$ et $D'(\sqsubseteq',\bot',\top',\sqcup',\sqcap')$ treillis complets, alors

f <u>est chaîne-continue</u> <=> $(\forall C)[C \subseteq D \wedge C \text{ chaîne} \Rightarrow \sqcup' f(C) = f(\sqcup C)]$

<u>Lemme 3</u> : f continue <=> f chaîne-continue [MaG 76]

Démonstration :

(=>) immédiat, puisque toute chaîne est un dirigé.

(<=) a) X dirigé $\wedge$ <u>X fini</u>, on peut décomposer X en un nombre fini de n chaînes finies C_i $(i=1,\ldots,n)$, telles que

$$X = \bigcup_{i=1}^{n} C_i \wedge \sqcup X = \sqcup C_i \quad (i=1,\ldots,n)$$

par conséquent :

$$f(\sqcup X) = f(\sqcup C_i) = \sqcup\{f(c) \mid c\in C_i\} \quad (i=1,\ldots,n)$$

$$f(\sqcup X) = \sqcup'\{\sqcup'\{f(c) \mid c\in C_i\} \mid i=1,\ldots,n\}$$

$$= \sqcup'(\bigcup_{i=1}^{n}\{f(c) \mid c\in C_i\}) \qquad \text{(th.4 §1.1.5)}$$

$$= \sqcup'\{f(x) \mid x\in X\} = \sqcup' f(X)$$

b) X dirigé $\wedge$ <u>X infini</u> ; par le lemme de Iwamura (§1.2.3), X peut être décomposé en une séquence de dirigés, qui forment une chaîne pour l'inclusion, et sont de cardinalité strictement inférieure à celle de X ; c'est-à-dire :

soit $|X| = \alpha$, alors

$$X = \bigcup_{\gamma<\alpha} \{X_\gamma \mid X_\gamma \text{ dirigé} \wedge |X_\gamma| = \gamma<\alpha \wedge (\gamma_1<\gamma_2<\alpha \Rightarrow X_{\gamma_1} \subseteq X_{\gamma_2} \subseteq X)\}$$

par conséquent : $\sqcup X = \sqcup\{\sqcup X_\gamma\}$ (th.4 §1.1.5)

où $\{\sqcup X_\gamma\}$ est une chaîne,

ainsi, on a : $f(\sqcup X) = \sqcup'\{f(\sqcup X_\gamma)\}$, et par induction transfinie, on obtient :

$f(\sqcup X) = \sqcup'\{f(x) \mid x\in X\}$ cqfd

Un autre concept de continuité d'une fonction est celui de ω-continuité, et se réfère au fait que le dirigé et/ou la chaîne est énumérable ; c'est-à-dire :

f <u>ω-continue</u> $\Leftrightarrow (\forall X)[X \text{ dirigé} \wedge X \text{ énumérable} \Rightarrow f(\sqcup X) = \sqcup' f(X)]$

Au contraire des fonctions monotones, f continue n'implique pas que $(\forall X)[X\subseteq D \Rightarrow \sqcup' f(X) = f(\sqcup X)]$. Si cette propriété est vérifiée pour une fonction f, alors f est <u>additive</u>.

Exemple : soit $f : P\omega \to P\omega$ définie par :

$$f(x) = \begin{cases} \bot & \text{si } x \text{ est } \bot \\ x & \text{si le cardinal de } x = 1, \\ \top & \text{si le cardinal de } x > 1 \end{cases} \qquad \text{Image}(f) = \text{Nat}$$

Cette fonction est continue et non additive, mais $f : P\omega \to \text{Nat}$ est additive.

Dans les trois lemmes suivants, nous mettons en relation quelques unes des définitions que nous avons données.

<u>Lemme 4</u> : f additive => f continue.

<u>Lemme 5</u> : f isomorphisme => f additive

Démonstration :

f isomorphisme $\Rightarrow \sqcup' f(X) \sqsubseteq' f(\sqcup X)$ $(\forall X \subseteq D)$ (f monotone)

$\Rightarrow f^{-1}(\sqcup' f(X)) \sqsubseteq f^{-1}(f(\sqcup X))$ (lemme §2.1.1.2)

$\Rightarrow \sqcup f^{-1}(f(X)) \sqsubseteq f^{-1}(\sqcup' f(X)) \sqsubseteq f^{-1}(f(\sqcup X)) = \sqcup X$ (f^{-1} monotone)

$\Rightarrow \sqcup X \sqsubseteq f^{-1}(\sqcup' f(X)) \sqsubseteq f^{-1}(f(\sqcup X)) = \sqcup X$

$\Rightarrow f^{-1}(\sqcup' f(X)) = f^{-1}(f(\sqcup X))$

$\Rightarrow \sqcup' f(X) = f(\sqcup X)$ (f^{-1} injective, lemme §2.1.1.1)

$\Rightarrow$ f additive

cqfd

<u>Lemme 6</u> : f additive $\wedge$ injective $\Rightarrow$ f strictement monotone.

Démonstration :

$f(x) \sqsubseteq' f(y) \Rightarrow f(y) = \sqcup'\{f(x), f(y)\}$

$\Rightarrow f(y) = f(x \cup y)$ (f additive)

$\Rightarrow y = x \cup y$ (f injective)

$\Rightarrow x \sqsubseteq y$

cqfd

Pour continuer, on établit la notion de domaine "isomorphe" ; c'est-à-dire, chaque élément d'un domaine possède les mêmes propriétés que son image dans l'autre domaine et vice-versa. Les deux domaines ont la même cardinalité.

<u>Lemme 7</u> :

Soient $D(\sqsubseteq, \perp, \top, \sqcup, \sqcap)$ et $D'(\sqsubseteq', \perp', \top', \sqcup', \sqcap')$ des treillis complets et soit $f : D \to D'$ un isomorphisme (c.a.d. D <u>isomorphe</u> à D') alors : D domaine $\Leftrightarrow$ D' domaine.

Démonstration :

($\Rightarrow$) (1) x fini en D(c.a.d. $x \ll x$) $\Rightarrow f(x)$ fini en D' (c.a.d. $f(x) \ll' f(x)$) ;

soit $Y \subseteq D' \wedge Y$ dirigé $\wedge f(x) \sqsubseteq' \sqcup' Y$, alors :

$f^{-1}(f(x)) = x \sqsubseteq f^{-1}(\sqcup' Y)$ (f^{-1} isomorphisme, lemme §2.1.1.2)

$x \ll x \wedge x \sqsubseteq f^{-1}(\sqcup' Y) \Rightarrow (\exists y)[y \in Y \wedge x \sqsubseteq f^{-1}(y)]$

$y \in Y \wedge x \sqsubseteq f^{-1}(y) \Rightarrow f(x) \sqsubseteq' y \wedge y \in Y \Rightarrow f(x) \ll' f(x)$

(2) pour tout $y \in D'$

$f^{-1}(y) \in D \Rightarrow f^{-1}(y) = \sqcup\{x \mid x \sqsubseteq f^{-1}(y) \wedge x \ll x\}$ (D domaine)

$\Rightarrow f(f^{-1}(y)) = y = \sqcup'\{f(x) \mid x \sqsubseteq f^{-1}(y) \wedge x \ll x\}$ (lemmes 4 et 5)

$\Rightarrow y = \sqcup'\{z \mid z \sqsubseteq' y \wedge z \ll' z\}$ (par (1) et f isomorphisme)

$\Rightarrow$ D' domaine.

De plus, on a :

(3) $z \ll' z \wedge z \sqsubseteq' y \Rightarrow f^{-1}(z) \sqsubseteq f^{-1}(y) \wedge f^{-1}(z) \ll f^{-1}(z)$ (par (2) et (1))

($\Leftarrow$) la démonstration est équivalente à ($\Rightarrow$).

cqfd

2.2. <u>Continuité dans les domaines de calcul</u>

Dans ce paragraphe, nous présentons trois résultats (théorèmes) qui établissent l'équivalence entre différentes façons de définir le concept de continuité des fonctions dans les domaines de calcul.

2.2.1. Théorème

Soit $f : D \to D'$, où D et D' sont des domaines de calcul, avec bases B et B' respectivement, alors on a :

$$\underline{f \text{ continue}} \Longleftrightarrow \underline{(\forall x)[x \in D \wedge B_x = \{e \mid e \in B \wedge e \sqsubseteq x\}} \; \underline{\Rightarrow f(x) = \sqcup' f(B_x)]}$$

Démonstration :

($\Rightarrow$) immédiat, puisque B_x dirigé $\wedge \sqcup B_x = x$ (lemme §1.4.1).

($\Leftarrow$) $\bigcup_{x \in X} B_x = \bigcup_{x \in X}\{e \mid e \in B \wedge e \sqsubseteq x\} \subseteq \{e \mid e \in B \wedge e \sqsubseteq \sqcup X\} = B_{\sqcup X}$

(puisque $x \in X$, $e \sqsubseteq x \Rightarrow e \sqsubseteq \sqcup X$) (i.e : $\bigcup_{x \in X} B_x \sqsubseteq B_{\sqcup X}$, t.3 §1.1.

mais,

$e \in B_{\sqcup X} \Rightarrow e \ll e \sqsubseteq \sqcup X \Rightarrow \exists x \in X \wedge e \sqsubseteq x$ (i.e : $B_{\sqcup X} \sqsubseteq \bigcup_{x \in X} B_x$, t.2 §1.1.5)

donc $\bigcup_{x \in X} B_x = B_{\sqcup X}$

$$\begin{aligned} f(\sqcup X) &= \sqcup' f(B_{\sqcup X}) = \sqcup' f(\bigcup_{x \in X} B_x) = \sqcup'(\bigcup_{x \in X} f(B_x)) \\ &= \sqcup'\{\sqcup' f(B_x) \mid x \in X\} \qquad \text{(Théorème 4, §1.1.5)} \\ &= \sqcup'\{f(x) \mid x \in X\} \\ &= \sqcup' f(X) \end{aligned}$$

cqfd

Ce théorème nous garantit que, dans les domaines de calcul, on peut se limiter aux dirigés composés des éléments de la base.

2.2.2. Théorème

Soit $f : D \to D'$, où D et D' sont des domaines de calcul avec bases B et B' respectivement, alors on a :

$$\underline{f \text{ continue}} \Longleftrightarrow \underline{(\forall x \in D)(\forall c' \in B')(\exists e \in B)[e' \sqsubseteq' f(x) \Longleftrightarrow e \sqsubseteq x \wedge e' \sqsubseteq' f(e)]}$$

ou, sous forme équivalente (théorème 2.2.1),

$$(\forall x \in D)[f(x) = \sqcup'\{f(e) \mid e \in B \wedge e \sqsubseteq x\} \Longleftrightarrow (\forall e' \in B')(\exists e \in B)(e' \sqsubseteq' f(x) \Longleftrightarrow e \sqsubseteq x \wedge e' \sqsubseteq f(e))]$$

Démonstration :

$(\Leftarrow)$ (1) $(\forall e')(\exists e)[e' \sqsubseteq' f(x) \Rightarrow e \sqsubseteq x \wedge e' \sqsubseteq' f(e)] \wedge f(x) = \bigsqcup'\{e' | e' \sqsubseteq f(x)\}$

$\Rightarrow f(x) \sqsubseteq' \bigsqcup'\{f(e) | e \sqsubseteq x\}$ $\qquad (\forall x)$

(2) $B_{y_1} = \{e | e \sqsubseteq y_1\} \wedge B_{y_2} = \{e | e \sqsubseteq y_2\} \wedge y_1 \sqsubseteq y_2 \Rightarrow B_{y_1} \subseteq B_{y_2}$,

$(\forall x)(\forall e')(\exists e)[e \sqsubseteq x \wedge e' \sqsubseteq' f(e) \Rightarrow e' \sqsubseteq' f(x)]$

$\Rightarrow \{e' | e' \sqsubseteq' f(y_1)\} \subseteq \{e' | e' \sqsubseteq f(y_2)\}$

par conséquent,

$f(y_1) = \bigsqcup'\{e' | e' \sqsubseteq' f(y_1)\} \sqsubseteq' \bigsqcup'\{e' | e' \sqsubseteq' f(y_2)\} = f(y_2)$

alors, $\bigsqcup'\{f(e) | e \sqsubseteq x\} \sqsubseteq' f(x)$

(1) $\wedge$ (2) $\Rightarrow f(x) = \bigsqcup'\{f(e) | e \sqsubseteq x\}$

$(\Rightarrow)$ D' domaine $\Rightarrow f(x) = \bigsqcup'\{e' | e' \ll' f(x)\}$

(lemme 2 §2.1.1.) $\wedge$ f continue $\Rightarrow \{f(e) | e \sqsubseteq x\}$ dirigé $\wedge\ f(x) = \bigsqcup'\{f(e) | e \sqsubseteq x\}$

$(\forall e')(e' \ll' f(x) \Leftrightarrow (\exists f(e))(e' \sqsubseteq' f(e)) \wedge e \sqsubseteq x$ $\qquad$ cqfd

Notons que ce théorème a été démontré avec D' treillis continu.
Ce théorème nous dit que les fonctions continues sont celles pour lesquelles on obtient une partie finie (e') du résultat (f(x)), chaque fois que la fonction est évaluée pour une partie finie (e) de l'argument (x). Cela signifie que cette définition de la continuité est reliée à une certaine notion de "calcul" [StJ 77]. Avec cette définition de fonction continue, semblable à celle ε-δ des espaces métriques, on va montrer l'équivalence avec la définition de fonction continue des espaces topologiques.

2.2.3. Théorème

Soient (D,T_i) et (D',T_i') des espaces topologiques (comme en §1.5.2.), $f : D \to D'$, alors :

f <u>continue</u> $\Leftrightarrow$ <u>$(\forall O')[O' \in T_i' \Rightarrow f^{-1}(O') \in T_i]$</u>

Démonstration :

$(\Leftarrow)$ Soit $O' \in B' \subseteq T_i'$ (B base de l'espace D'), $O' = F_{e'}$ (filtre associé à l'élément e' de la base du domaine D'), de plus $f^{-1}(F_{e'}) = \bigcup_{e \in B_{e',f}} F_e$,

où $B_{e',f} \subseteq B \subseteq T_i$. On a que $\forall F_{e'}$, les implications suivantes sont valides :

(1) $(\forall x) \quad f(x) \in F_{e'} \Rightarrow e' \sqsubseteq' f(x)$ $\qquad$ ($F_{e'}$ filtre-ouvert)

(2) $(\forall x) \quad x \in f^{-1}(F_{e'}) \Rightarrow (\exists e)(e \in f^{-1}(F_{e'}) \wedge e \sqsubseteq x)$ $\qquad$ (caractère fini des ouverts)

(3) $(\forall e) \ e \in f^{-1}(F_{e'}) \Rightarrow f(e) \in F_{e'} \wedge e' \sqsubseteq' f(e)$ $\qquad$ (hypothèse et $F_{e'}$ filtre)

autrement dit, l'hypothèse implique :

$(\forall e')(\forall x)(\exists e)[e' \sqsubseteq' f(x) \wedge e \sqsubseteq x \wedge e' \sqsubseteq' f(e)]$

qui est une condition suffisante pour obtenir :

$(\forall x)(\forall e')(\exists e)[e' \sqsubseteq' f(x) \Leftrightarrow e \sqsubseteq x \wedge e' \sqsubseteq' f(e)]$

$(\Rightarrow)$ Supposons que $f^{-1}(O') \notin T_i$ (n'est pas ouvert)

(1) $f^{-1}(O') \notin T_i \Rightarrow (\exists \bar{x})(\forall e)[e \sqsubseteq \bar{x} \Rightarrow e \notin f^{-1}(O')$

($f^{-1}(O')$ ne possède pas la propriété de caractère fini)

(2) $f(\bar{x}) \in T_i' \Rightarrow (\exists e')[e' \in O' \wedge e' \sqsubseteq' f(\bar{x})]$ (caractère fini des ouverts)

(3) f continue $\Rightarrow (\forall e')(\exists \underline{e})[e' \sqsubseteq' f(\bar{x}) \Leftrightarrow \underline{e} \sqsubseteq \bar{x} \wedge e' \sqsubseteq' f(\underline{e})]$

(4) $e' \sqsubseteq' f(\underline{e}) \wedge e' \in O' \Rightarrow f(\underline{e}) \in O'$

(par (2), (3) et monotonie des ouverts)

$f(\underline{e}) \in O' \wedge \underline{e} \sqsubseteq \bar{x}$ contredit (1) puisque $\underline{e} \in f^{-1}(O')$ cqfd.

2.3. Points fixes et caractérisation de domaines

Un des problèmes que l'on a cité au commencement de ce travail est l'existence du point fixe - $x=f(x)$ - d'une fonction $f:D \to D$. Le théorème que l'on va donner maintenant affirme que pour les domaines et les fonctions que l'on a présentés, ces points fixes existent.

2.3.1. Théorème de Tarski [TaA 55]

Soit $D(\sqsubseteq, \bot, \top, \sqcup, \sqcap)$ un treillis complet, et soit $f:D \to D$ une fonction monotone, alors :

$FIX(f) = \{x | x \in D \wedge x=f(x)\}$ n'est pas vide,

et $FIX(f)(\sqsubseteq, \bot', \top', \sqcup', \sqcap')$ est un sous-treillis complet de D.

Cela signifie que, outre que les points fixes existent, ceux-ci forment un sous-treillis complet de D avec la même relation d'ordre.

En [Co PR 77] se trouve une démonstration de ce théorème qui, par son caractère constructif (schémas itératifs) s'adapte aux problèmes de calcul.

Dans le paragraphe §4.2.1., on donnera une façon d'obtenir le plus petit point fixe ($\bot'$ en FIX(f)), mais avec l'hypothèse que f est continue.

Nous présentons ci-dessous un corollaire du théorème de Tarski, en supposant que la fonction est continue.

Corollaire : $D(\sqsubseteq, \bot, \top, \sqcup, \sqcap) \wedge f : D \to D$ continue

$\Rightarrow$ FIX(f)($\sqsubseteq, \bot', \top', \sqcup', \sqcap'$) sous-treillis de D, de caractère fini (§1.3.3.).

Cela signifie : $(\forall X)[X$ dirigé $\wedge X \subseteq \mathrm{Fix}(f) \subseteq D \Rightarrow \sqcup' X = \sqcup X]$

Démonstration :

$f(\sqcup X) = \sqcup\{f(x) \mid x \in X\} \wedge X \subseteq \mathrm{FIX}(f) \Rightarrow f(\sqcup X) = \sqcup\{x \mid x \in X\} = \sqcup X$

c'est-à-dire : $\sqcup X \in \mathrm{FIX}(f)$ et $\sqcup' X = \sqcup X$ cqfd.

2.3.2. Rétractions

Une fonction continue $f : D \to D$, D treillis complet, est idempotente ssi

$(\forall x)[x \in D \Rightarrow (f \circ f)(x) = f^2(x) = f(f(x)) = f(x)]$

c'est-à-dire :

$f(D) = \mathrm{Image}(f) = \mathrm{FIX}(f) \Leftrightarrow f^2 = f \circ f = f$

Alors, par le corollaire (§2.3.1.) du paragraphe précédent, on a que :

Image(f)($\sqsubseteq, \bot', \top', \sqcup', \sqcap'$) est sous-treillis complet de caractère fini du treillis complet $D(\sqsubseteq, \bot, \top, \sqcup, \sqcap)$, si $f : D \to D$ est continue et idempotente ($f \circ f = f$).

En raison de cela, on dira que f caractérise un sous-treillis de D

Avec le théorème suivant, on voit comment se comporte une fonction continue et idempotente quand le treillis est continu.

Théorème :

Soit $D(\sqsubseteq, \bot, \top, \sqcup, \sqcap)$ treillis continu, et soit $f : D \to D$, f continue, $f \circ f = f$ (idempotente), alors :

Image(f) ($\sqsubseteq, \bot', \top', \sqcup', \sqcap'$) est sous-treillis continu de caractère fini de D.

Démonstration :

Comme Image(f) est sous-treillis complet de caractère fini de D, on va montrer : D continu $\Rightarrow$ Image(f) continu.

$(\forall y)(y \in \mathrm{Image}(f) \subseteq D)$:

(1) $x \ll y \Rightarrow (\forall Z)(\exists z)[Z$ dirigé $\wedge Z \subseteq D \wedge y \sqsubseteq \sqcup Z \Rightarrow z \in Z \wedge x \sqsubseteq z$

(partie essentielle §1.2.4.)

en particulier, de(1) on a

$(\forall Z)[Z \subseteq \mathrm{Image}(f) \Rightarrow f(x) \sqsubseteq f(z) = z]$ (f monotone)

donc, comme $f(x) \in \mathrm{Image}(f)$

(2) $x << y \Rightarrow f(x) <<' y$ (f(x) partie essentielle de y dans Image (f))

(3) $y = f(y) = f(\bigsqcup\{x | x << y\} = \bigsqcup\{f(x) | x << y\}$ (D continu et f continu)

(2) $\Rightarrow \{f(x) | x << y\} \subseteq \{f(x) | f(x) <<' y\}$

$\{f(x) | x << y\} \subseteq \{v <<' y \wedge v \in \text{Image}(f)\}$

(3)$\Rightarrow$ $y = \bigsqcup\{f(x) | x << y\} \sqsubseteq \bigsqcup\{v | v <<' y\}$

donc

$$y = \bigsqcup\{v | v <<' y\} \qquad \text{(puisque } v <<' y \Rightarrow v \sqsubset y)$$

Comme le résultat est vrai, $\forall y$, $y \in \text{Image}(f) \subseteq D$, on a que

Image(f)$(\sqsubseteq, \bot', \top', \bigsqcup', \sqcap')$ treillis continu

cqfd.

Dans la littérature sur les domaines, on dit que f est la rétraction ("retract") qui caractérise le sous-treillis continu de D égal à Image(f).

2.3.3. Fermetures

Une classe particulière de rétractions est la classe des fermetures.
Leur définition est la suivante :
Une fonction continue et idempotente $g : D \to D$, D treillis complet, est une fermeture ssi

$$(\forall x)[x \in D \Rightarrow x \sqsubseteq g(x) = g(g(x))]$$

Dans le cas où D est un domaine - treillis algébrique -, on a le résultat suivant :

Théorème :
Soit $D(\sqsubseteq, \bot, \top, \bigsqcup, \sqcap)$ treillis algébrique, et soit $g : D \to D$ une fermeture continue, alors :

Image(g)$(\sqsubseteq, \bot', \top', \bigsqcup', \sqcap')$ est un sous-treillis algébrique de caractère fini, $\sqcap$-fermé de D (c.à.d. $\sqcap' = \sqcap$), et $\top' = \top$.

Démonstration :
Par le théorème du paragraphe antérieur (§2.3.2.), on sait que Image(g) est un sous-treillis continu, puisque D domaine (treillis algébrique) implique D continu. Ainsi on va montrer que pour tout compact en D, son image en g(D) = Image(g) est aussi compacte.

(1) $(\forall x)[x \in D \wedge x << x \wedge x \sqsubseteq g(x) \Rightarrow x << g(x)]$ (g fermeture $\wedge$ §1.2.6.b)

(2) $(\forall y)[y \in \text{Image}(g) \subseteq D \wedge x << y \Rightarrow g(x) <<' y]$ (théorème §2.3.2.)

comme $g(x) \in \text{Image}(g)$, par (1) et (2), on a que :

$(\forall x)\ [x \in D \wedge x << x \Rightarrow g(x) <<' g(x)]$

par conséquent,

Image(g) est un sous-domaine de caractère fini de D.

Soit $X \subseteq$ Image(g), on sait que :

$\sqcap' X \sqsubseteq \sqcap X$ (en D)

mais $\sqcap X \sqsubseteq g(\sqcap X) \wedge g(\sqcap X) \in$ Image(g) (g fermeture)

et $(\forall x)[x \in X \Rightarrow g(\sqcap X) \sqsubseteq g(x) = x]$ (g monotone et idempotente)

alors, $\sqcap' X = g(\sqcap X)$

comme $\sqcap' X \sqsubseteq \sqcap X \sqsubseteq g(\sqcap X)$ on a $\sqcap' X = \sqcap X$

c'est-à-dire, Image(g) est un sous-domaine $\sqcap$-fermé de D.

Il est facile de voir que $\top' = \top$, puisque $\top' \sqsubseteq \top \sqsubseteq g(\top) = \top'$

cqfd.

Exemple 1 : en $P\omega$, soit $S = \{x \mid x \in P\omega \wedge \text{cardinalité de } x = 1\}$,
une fermeture continue en $P\omega$ (dépendant de $S' \subseteq S$) est :

$$g_{S'}(x) = \begin{cases} \bot & \text{si } x = \bot \\ x & \text{si } x \in S' \subseteq S \\ \top & \text{si } x \notin S' \subseteq S \end{cases}$$

Cette fermeture nous fournit les treillis plats, à partir des ensembles unitaires de $P\omega$.

Dans ce qui suit, on identifiera par fermeture, indistinctement, la fonction et le domaine qu'elle caractérise.

Il y a une autre classe de rétractions qui s'appellent <u>projections</u>, et sont le "dual" des fermetures ; leur définition est :

p projection $\Leftrightarrow$ p monotone $\wedge$ $(\forall x)(p(p(x)) = p(x) \sqsubseteq x)$

Si le domaine possède la propriété : $x \sqsubseteq y \wedge y$ fini $\Rightarrow$ x fini
(notons que cette propriété est une condition nécessaire pour que la relation de partie essentielle soit bien fondée §1.3.5.), alors l'image d'une projection dans le domaine est aussi un domaine.

Le domaine caractérisé par la projection sera $\sqcup$-fermé.

Exemple 2 : en $P\omega$, soit le sous-domaine engendré par les ensembles finis d'entiers de cardinalité supérieure ou égale à 2, c'est-à-dire : $P\omega$-S
(S de l'exemple 1). Ce sous-domaine de $P\omega$ est caractérisé par la projection continue en $P\omega$:

$$p(x) = \begin{cases} \bot & \text{si } x \in S \vee x = \bot \\ x & \text{si } x \notin S \end{cases}$$

La fonction identité I (c.à.d. : $(\forall x)(I(x) = x)$), est à la fois une fermeture et une projection.

2.3.4. Rétracts à caractère fini

En général, les rétracts définis sur un domaine et dont l'image est aussi un domaine, seront appelés <u>rétracts à caractère fini</u>. Avec le théorème suivant, nous verrons que les éléments finis dans l'image d'un rétract défini sur un domaine, sont les images par le rétract des éléments finis du domaine.

<u>Théorème</u> :

Soient : . D domaine et E les éléments finis en D

. $f : D \to D \wedge f^2 = f \wedge f$ continu

alors : $(\forall x)(\exists e)\ [x \in \text{Image}(f) \wedge x <<' x$

$\Rightarrow e \in E \wedge e \sqsubseteq x \wedge x = f(e) \wedge e \sqsubseteq f(e)]$

(où <<' est la relation de partie essentielle dans Image(f))

Démonstration : soit $E_x = \{e | e \in E \wedge e \sqsubseteq x\}$

alors $x = \bigsqcup E_x \wedge x \in \text{Image}(f)$

$\Rightarrow x = \bigsqcup f(E_x) \wedge f(E_x)$ dirigé $\wedge f(E_x) \subseteq \text{Image}(f)$

(f^2=f et lemme 2 §2.1.1.)

on a :

$x <<' x \wedge x = \bigsqcup f(E_x) \Rightarrow (\exists f(e))(f(e) \in f(E_x) \wedge x \sqsubseteq f(e))$

mais : $f(e) \sqsubseteq f(x) \wedge f(x) = x \wedge e \sqsubseteq x$

par conséquent : $x = f(e) \wedge e \sqsubseteq x \wedge e \sqsubseteq f(e)$

cqfd.

CHAPITRE 3.
CONSTRUCTION DE DOMAINES DE CALCUL

3. CONSTRUCTION DE DOMAINES DE CALCUL

Les opérations (constructeurs) de produit cartésien ($\times$), union disjointe (+) et espace de fonctions continues ($\rightarrow$) entre les domaines sont fermées ; c'est-à-dire que ces opérations "construisent" de nouveaux domaines. Dans cette partie du travail, nous présentons les domaines que créent ces constructeurs.

3.1. Produit cartésien

Soient D_1, $D_2 \ldots D_n$ des domaines ; le produit cartésien est défini comme suit :

$$<x_1,x_2,\ldots x_n> \in D_1 \times D_2 \times \ldots D_n \Leftrightarrow (\, i = 1,\ldots n)\ x_i \in D_i$$

Dans les n-uplets ordonnés $<x_1,x_2,\ldots x_n> \in D_1 \times D_2 \times \ldots \times D_n$, on a l'opération de projection $\downarrow$, définie par les n-fonctions suivantes :

$$<x_1,x_2\ldots,x_n> \downarrow j = x_j \in D_j \qquad (j = 1,2,\ldots n)$$

3.1.1. Définitions en $D_1 x D_2 x \ldots x D_n$ (D_i domaines)

Nous donnons une série de définitions qui vérifient les concepts que nous avons donnés pour les domaines. Pour abréger, nous représenterons un n-uplet ordonné $<x_1,x_2\ldots x_n>$ par $<x_i>$.

3.1.1.1. Relation d'ordre partiel

$$<x_i>,\ <y_i> \in D_1 \times \ldots \times D_n$$

$$<x_i> \sqsubseteq <y_i> \iff <x_i> \downarrow j \sqsubseteq <y_j> \downarrow j \quad (\text{en } D_j) \qquad (j = 1,\ldots,n)$$

C'est-à-dire, chaque composante de $<x_i>$ est plus petite ou égale à la composante en $<y_i>$.

3.1.1.2. Borne supérieure et inférieure, $\top$ et $\bot$, dirigé.

Soit $X \subseteq D = D_1 \times D_2 \times \ldots \times D_n$,

et soit $X_j = \{z \mid z \in D_j \wedge (\exists <x_i>)(<x_i> \in X \wedge z = <x_i> \downarrow j)\} \qquad (j = 1,\ldots,n)$,

c'est-à-dire, X_j est l'ensemble des éléments correspondant à la composante j des éléments de X.

On a :

- $\bigsqcup_D X = \langle \bigsqcup_{D_1} X_1,\ \bigsqcup_{D_2} X_2, \ldots, \bigsqcup_{D_n} X_n \rangle$

- $\bigsqcap_D X = \langle \bigsqcap_{D_1} X_1,\ \bigsqcap_{D_2} X_2, \ldots, \bigsqcap_{D_n} X_n \rangle$

- $\top_D = \langle \top_{D_1},\ \top_{D_2}, \ldots \top_{D_n} \rangle$

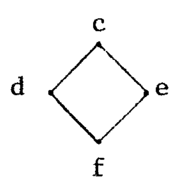
- $\bot_D = \langle \bot_{D_1},\ \bot_{D_2}, \ldots \bot_{D_n} \rangle$

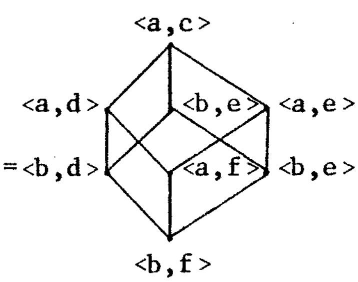
(les indices en $\bigsqcup$, $\bigsqcap$, $\bot$ et $\top$ dénotent les domaines où sont prises les bornes).
Il est clair que ces définitions satisfont le concept de borne ;
cependant, notons que $X \subseteq X_1 \times X_2 \times \ldots \times X_n$.

Aussi, on a que :
X dirigé en $D = D_1 \times D_2 \times \ldots \times D_n \Rightarrow X_j$ dirigé en D_j $(j = 1, \ldots n)$
le contraire n'est pas toujours vrai, mais :
X_j dirigé en D_j $(j = 1, \ldots, n) \Rightarrow X_1 \times X_2 \times \ldots \times X_n$ dirigé en $D = D_1 \times D_2 \times \ldots \times D_n$.

Exemple : soient,

$D_1 =$ 

, $D_2 =$ 

, $D_1 \times D_2 =$ 

$X = \{\langle a,c \rangle, \langle a,d \rangle, \langle b,c \rangle, \langle a,e \rangle, \langle a,f \rangle\} \subseteq D_1 \times D_2$, X dirigé.

$Y = \{\langle b,c \rangle, \langle b,d \rangle, \langle a,f \rangle, \langle b,e \rangle, \langle b,f \rangle\} \subseteq D_1 \times D_2$, Y non dirigé.

cependant : $X_1 = Y_1 = D_1$, $X_2 = Y_2 = D_2$ et $D_1 \times D_2$ est dirigé.

3.1.1.3. Partie essentielle - Elément compact

$\langle x_i \rangle, \langle y_i \rangle \in D_1 \times D_2 \times \ldots \times D_n$

. $\langle x_i \rangle \ll \langle y_i \rangle \iff \langle x_i \rangle \downarrow j \ll \langle y_i \rangle \downarrow j$ (en D_j) $(j = 1, \ldots, n)$

. $\langle x_i \rangle$ compact $\iff \langle x_i \rangle \downarrow j$ compact en D_j $(j = 1, \ldots, n)$

3.1.1.4. Base

Soient $B_1 \subseteq D_1$, $B_2 \subseteq D_2, \ldots, B_n \subseteq D_n$ des bases (§1.4), alors $B_1 \times B_2 \times \ldots \times B_n$ est une base de $D_1 \times D_2 \times \ldots \times D_n$.

3.1.1.5. Base récursive

Soient $B_1 \subseteq D_1$, $B_2 \subseteq D_2$, $\ldots$ $B_n \subseteq D_n$ des bases récursives (§1.4.4) et soient $I_1, I_2, \ldots, I_n$ les ensembles récursifs d'indices ($I_i \subseteq \mathbb{N}$) associés aux bases $B_1, B_2, \ldots, B_n$; il suffit de prendre une fonction de codification de $\mathbb{N}^n$ dans $\mathbb{N}$ qui soit injective et récursive, par exemple la fonction [RoH 67] :

$$\tau_2(x,y) = \frac{1}{2}\left((x+y)^2 + 3x + y\right)$$

qui est une fonction injective et récursive de $\mathbb{N}^2 \to \mathbb{N}$; elle est étendue pour les puissances finies de $\mathbb{N}$ de la manière suivante :

$$\tau_1 = \lambda x.x \qquad \text{(identité)}$$

$$\tau_{k+1} = \lambda x_1 \ldots x_{k+1}.\ \tau_2(\tau_k(x_1, \ldots, x_k), x_{k+1})$$

Cette fonction est aussi injective et récursive.
Avec ces définitions, il est facile de montrer le lemme qui suit.

3.1.2. Lemme

Soient $D_1, D_2, \ldots D_n$ des treillis complets (- continus) (- algébriques) (domaines de calcul), alors
$D_1 \times D_2 \times \ldots \times D_n$ est un treillis complet (- continu) (- algébrique) (domaine de calcul).

3.1.3. Fonctions de plusieurs variables - Continuité

Grâce au concept de dirigé et de borne supérieure dans un domaine obtenu comme produit cartésien d'autres domaines (§3.1.1.2.), on a le théorème suivant :

Théorème :

Soit $f : D \to D'$, $D = D_1 \times D_2 \times \ldots \times D_n$ et D' domaines, alors
f continue (en D) <=> f continue dans chaque D_j $(j=1,\ldots,n)$ séparément.

Démonstration :
Nous donnons la démonstration pour deux variables, qui se généralise facilement à n variables $(D_1 \times D_2 \times \ldots \times D_n = D_1 \times (D_2 \times \ldots \times D_n))$.
Le théorème devient :
soit $g : D \times D' \to D''$
et soit $Z \subseteq D \times D'$ dirigé,
c'est-à-dire $Z_1 = X \subseteq D$ dirigé et $Z_2 = Y \subseteq D'$ dirigé (§3.1.1.2.),
alors :

$$g(\underline{\sqcup}Z) = \sqcup'' g(Z) \iff (1)\ g(\langle \sqcup X,y\rangle) = \sqcup'' g(X,y)$$
$$\text{et}\quad (2)\ g(\langle x,\sqcup' Y\rangle) = \sqcup'' g(x,Y)$$

(où $\underline{\sqcup}$ dénote la borne supérieure en $\underline{D} = D \times D'$, et
$g(X,y) = \{g(\langle x,y\rangle) \mid x\in X\}$, et $g(x,Y) = \{g(\langle x,y\rangle) \mid y\in Y\}$).

(=>) immédiat, puisque :
$Z = (X,y) = \{\langle x,y\rangle \mid x\in X\} = X \times \{y\}$ implique $\underline{\sqcup}\, Z = \langle \sqcup X,y\rangle$
et $Z = (x,Y) = \{\langle x,y\rangle \mid y\in Y\} = \{x\} \times Y$ implique $\underline{\sqcup} Z = \langle x,\sqcup' Y\rangle$ (§3.1.1.2.)
(<=) Soit $\underline{\sqcup} Z = \langle \sqcup X,\sqcup' Y\rangle = \underline{\sqcup}\ (X \times Y)$ (§3.1.1.2)

$$\begin{aligned}
g(\underline{\sqcup}Z) &= g(\langle \sqcup X,\sqcup' Y\rangle) \\
&= \sqcup''\{g(\langle x,\sqcup' Y\rangle) \mid x \in X\} && \text{(par (1))} \\
&= \sqcup''\ \{\sqcup''\{g(\langle x,y\rangle) \mid y \in Y\} \mid x \in X\} && \text{(par (2))} \\
&= \sqcup''\ (\bigcup_{x\in X}\ \{g(\langle x,y\rangle) \mid y \in Y\}) && \text{(théorème 4, §1.1.5)} \\
&= \sqcup''\{g(\langle x,y\rangle) \mid x \in X \wedge y \in Y\} \\
&= \sqcup''\{g(\langle x,y\rangle) \mid \langle x,y\rangle \in X \times Y\}
\end{aligned}$$

On sait que : $Z \subseteq X \times Y \wedge Z_1 = X \wedge Z_2 = Y$ (§3.1.1.2),
par conséquent (théorème 3 §1.1.5) :

$\sqcup'' g(Z) = \sqcup''\{g(<x,y>) \mid <x,y> \in Z\} \sqsubseteq g(\sqcup Z)$

et $g(\sqcup Z) = \sqcup''\{g(<x,y>) \mid <x,y> \in X \times Y\} = \sqcup'' g(X \times Y)$

Maintenant, il faut montrer :

$$g(\sqcup Z) = \sqcup'' g(X \times Y) \sqsubseteq \sqcup'' g(Z) \qquad \text{(ineg-1)}$$

Soit $g(<x,y>) \in g(X \times Y)$

comme $Z_1 = X \wedge Z_2 = Y$, il existe $<x,y'> \in Z \wedge <x',y> \in Z$;

mais Z dirigé implique

$$(\exists <\bar{x},\bar{y}>)\ <\bar{x},\bar{y}> \in Z \wedge <x,y'> \sqsubseteq <\bar{x},\bar{y}> \wedge <x',y> \sqsubseteq <\bar{x},\bar{y}>$$

par monotonie de g sur chaque variable ((1) et (2)) :

$$g(<x,y>) \sqsubseteq'' g(<\bar{x},\bar{y}>) \wedge g(<\bar{x},\bar{y}>) \in g(Z)$$

et donc ineg-1 est vérifiée (théorème 2 §1.1.5)

cqfd

Exemple 1 : l'opération d'union "$\cup$" (§ 1.3.1.1.) dans les treillis complets, définie par :

$\cup : D \times D \to D$, D treillis complet, et $\cup(x,y) = \sqcup\{x,y\}$,

est une fonction continue. La démonstration que cette fonction est continue, et même additive, est une conséquence du théorème 4 (§ 1.1.5) (voir aussi la démonstration du lemme § 1.3.5).

La projection, au § 3.1, est continue.

Exemple 2 : l'opération d'intersection "$\cap$" (§ 1.3.1.1) dans les treillis continus est une fonction continue. Cette fonction est définie de la manière suivante :

$\cap : D \times D \to D$, D treillis continu et $\cap(x,y) = \sqcap\{x,y\}$;

Comme l'intersection est une fonction commutative (c'est-à-dire $x \cap y = y \cap x$) et en vertu du théorème précédent, nous allons démontrer :

$(\forall Z)[Z \subseteq D \wedge D$ continu $\wedge$ Z dirigé $\Rightarrow y \cap (\sqcup Z) = \sqcup\{y \cap z \mid z \in Z\}$.

De plus, l'intersection est monotone :

$$x \sqsubseteq y \Rightarrow z \cap x \sqsubseteq z \cap y$$

par conséquent :

$$y \cap (\sqcup z) \sqsupseteq \sqcup\{y \cap z \mid z \in Z\}$$

Ainsi donc, il suffit de démontrer :

$$y \cap (\sqcup z) \sqsubseteq \sqcup\{y \cap z \mid z \in Z\}$$

comme D est un treillis continu (domaine), on a :

$$y \cap (\sqcup z) = \sqcup RC(y \cap \sqcup Z) \text{ avec } RC(y \cap \sqcup Z) = \{x \mid x \ll y \cap \sqcup Z\}$$

et

$(\forall x)[x \in RC(y \sqcap \bigsqcup Z) \Rightarrow x \ll y \wedge x \ll \bigsqcup Z]$

$(\forall x)[x \ll \bigsqcup Z \Rightarrow (\exists z)(z \in Z \wedge x \sqsubseteq z)$

$(\forall x)(\exists z)\ [x \in RC(y \sqcap \bigsqcup Z) \Rightarrow z \in Z \wedge x \sqsubseteq y \wedge x \sqsubseteq z]$

$(\forall x)\ [x \in RC(y \sqcap \bigsqcup Z) \Rightarrow (\exists z)(z \in Z \wedge x \sqsubseteq y \sqcap z)]$

c'est-à-dire, pour tout x dans $RC(y \sqcap \bigsqcup Z)$, il existe un élément v en $\{y \sqcap z | z \in Z\}$ tel que $x \sqsubseteq v$; par conséquent (théorème 2 §1.1.5) :

$$y \sqcap \bigsqcup Z = \bigsqcup RC(y \sqcap \bigsqcup Z) \sqsubseteq \bigsqcup \{y \sqcap z | z \in Z\}$$

cqfd

Exemple 3 : $P_{\omega f}$ (dans l'exemple du § 1.3.3) est un treillis complet non continu ; dans ce treillis, l'opération d'intersection n'est pas continue :
Soit Z l'ensemble des ensembles finis d'impairs et soit $y = \{2\}$, on a :

$$y \sqcap \bigsqcup Z = y \qquad \text{(puisque } \bigsqcup Z = \top)$$

$$\bigsqcup \{y \sqcap z | z \in Z\} = \bot$$

3.2. Union disjointe ou somme

Soient $D_1, D_2, \ldots, D_n$ des domaines, on définit l'union disjointe (ou somme) comme suit :

$$x \in D_1 + D_2 + \ldots + D_n \iff x \in D_j \quad \text{(pour un certain } j = 1,2,\ldots,n)$$

Notons qu'un même élément peut apparaître plusieurs fois (dans des composants distincts de la somme) dans l'union "disjointe" ; de même, le même domaine peut apparaître plusieurs fois comme composant de l'union disjointe. Ainsi donc, un même élément est différent s'il se trouve dans deux composants distincts de la somme. Nous verrons cela plus précisément avec la définition de la relation d'ordre partiel (§ 3.2.3.1).

3.2.1. $\bot$ et $\top$ dans l'union disjointe

Soit $D = D_1 + D_2 + \ldots + D_n$, on a :

- $\top_D = \top$ en D implique $\top_{Di} \sqsubseteq \top_D \qquad (i = 1,2,\ldots,n)$
- $\bot_D = \bot$ en D implique $\bot_D \sqsubseteq \bot_{Di} \qquad (i = 1,2,\ldots,n)$

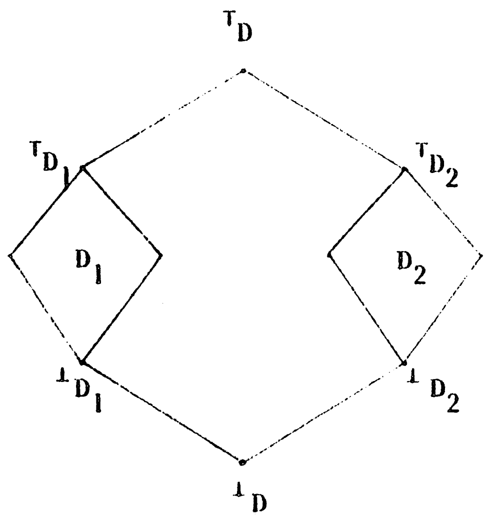
en particulier, on distingue deux cas :

(1) $[\top_{D_i} = \top_D \wedge \bot_D = \bot_{D_i}]$ (i = 1, 2, ..., n)

et (2) $[\top_{D_i} \neq \top_D \wedge \bot_D \neq \bot_{D_i}]$ (i = 1, 2, ..., n)

Le second cas est connu comme la somme séparée. Graphiquement, nous pouvons représenter la somme séparée de la manière suivante :

soit $D = D_1 + D_2$,



La somme séparée a l'avantage de permettre la distinction des $\bot$ (et $\top$) de chaque domaine de la somme.

3.2.2. Opérations sur l'union disjointe

Nous donnons maintenant trois opérations (continues) qui permettent d'identifier des éléments dans l'union disjointe.

3.2.2.1. Injection (in)

Soit la somme $D = D_1 + D_2 + \ldots + D_n$, l'opération d'injection est donnée par les n-fonctions (continues) suivantes :

$\underline{D_i \text{ in } D} : D_i \rightarrow D$ (i = 1,...,n)

avec $\underline{D_i \text{ in } D(x)} = (x_i \text{ in } D)$ dénote l'élément correspondant à $x \in D_i$ dans la somme D.

3.2.2.2. Projection ($|$).

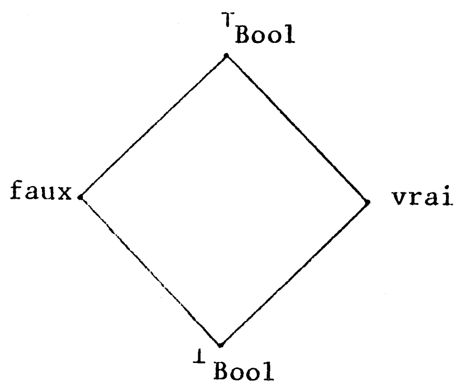
Soit la somme $D = D_1 + D_2 + \dots + D_n$, les n-projections (fonctions continues) sont : $\quad |D_i : D \to D_i \qquad (i = 1, \dots, n)$

$$\text{avec} \quad |D_i(x) = x|D_i = \begin{cases} \top_{D_i} & \text{si } x = \top_D \\ y & \text{si } x = (y_i \text{ in } D) = \underline{D_i \text{ in } D}\ (y) \\ \perp_{D_i} & \text{sinon} \end{cases}$$

3.2.2.3. Appartenance à un composant de la somme ("E" ou "est") et <u>sib</u>.

Soit Bool le domaine des valeurs booléennes :

<u>Bool</u>



et soit $D = D_1 + D_2 + \dots + D_n$; l'appartenance d'un élément à un composant de la somme est défini de la manière suivante :

$$x \text{ est } D_i = x \vec{E}\, D_i = \begin{cases} \perp_{Bool} & \text{si } x \in D \wedge x = \perp_D \\ \top_{Bool} & \text{si } x \in D \wedge x = \top_D \\ \text{vrai} & \text{si } x \in D \wedge x = (y_i \text{ in } D) \\ \text{faux} & \text{si } x \in D \wedge x = (y_j \text{ in } D) \wedge i \neq j \end{cases}$$

A partir de ce prédicat en <u>Bool</u>, définissons la fonction (continue) de condition :

$\underline{\text{sib}} : \text{Bool} \times D \times D \to D \qquad$ (D domaine)

où $\quad \underline{\text{sib}}\ (\top_{Bool}, x, y) = \top_D$

$\underline{\text{sib}}\ (\perp_{Bool}, x, y) = \perp_D$

$\underline{\text{sib}}\ (\text{vrai}, x, y) = x$

$\underline{\text{sib}}\ (\text{faux}, x, y) = y$

3.2.3. Définitions en $D_1 + D_2 + \dots + D_n$

Comme au § 3.1.1., nous donnons maintenant une série de définitions qui vérifient les concepts présentés dans les domaines.

3.2.3.1. Relation d'ordre partiel ($\sqsubseteq_D$)

Soit $D = D_1 + D_2 + \dots + D_n$, D_i domaines $(i = 1,\dots,n)$

$$(\forall x)(\forall y)[x \in D \wedge y \in D \wedge x \to D_i \wedge y \to D_j \wedge (1 \le i,j \le n) \Rightarrow$$
$$x \sqsubseteq_D y \Leftrightarrow i=j \wedge x \sqsubseteq_{D_i} y]$$

c'est-à-dire, deux éléments de la somme sont comparables ssi ils appartiennent au même composant et sont comparables dans ce composant.

3.2.3.2. Borne supérieure et inférieure

Soit $D = D_1 + D_2 + \dots + D_n$, D_i domaines $(i = 1,\dots,n)$;
soit $X \subseteq D$, et soit I l'ensemble des indices des composants associés à X de la manière suivante :

$$i \in I \Leftrightarrow (\exists x)[x \in X \wedge x \to D_i]$$

on a deux cas :

(1) $|I| = 1$ (c.à.d. : $I = \{k\}$)
$$\Rightarrow \bigsqcup_D X = (\bigsqcup_{D_k} X \text{ in } D) \text{ et } \bigsqcap_D X = (\bigsqcap_{D_k} X \text{ in } D)$$

(2) $|I| > 1 \Rightarrow \bigsqcup_D X = \top_D$ et $\bigsqcap_D X = \bot_D$

c'est-à-dire, si X est contenu dans un composant, sa borne est la borne dans ce composant ; dans le cas contraire, les bornes supérieures et inférieures de X seront respectivement $\top$ et $\bot$ de D. Notons qu'un dirigé est "intéressant" (c.à.d. ne contient pas la borne supérieure), s'il est contenu dans un composant de la somme et s'il est intéressant dans ce composant.

3.2.3.3. Partie essentielle - Elément compact ($\ll_D$).

Soit la somme $D = D_1 + D_2 + \dots + D_n$, D_i domaines $(i = 1,\dots,n)$

$$(\forall x)(\forall y)[x \in D \wedge y \in D \wedge x \to D_i \wedge y \to D_j \wedge (1 \le i, j \le n) \Rightarrow$$
$$x \ll_D y \Leftrightarrow i = j \wedge x \ll_{D_i} y]$$

Cela signifie que la propriété d'être partie essentielle ou compact se maintient quand un domaine est injecté dans une somme.

3.2.3.4. Base

Soit $D = D_1 + D_2 + \dots + D_n$, D_i domaines $(i = 1,\dots,n)$
et soient $B_i \subseteq D_i$ $(i = 1,\dots,n)$ bases,
une base B de D $(B \subseteq D)$ est :

$$B = B_1 + B_2 + \dots B_n.$$

Il est clair que n'importe quel élément de D peut être engendré à partir des éléments de B. Mais tous les sous-ensembles finis de B n'ont pas leur borne supérieure en B (second cas du § 3.2.3.2), sauf si nous définissons la somme $B = B_1 + B_2 + \dots + B_n$ avec un élément $\top_B$.

3.2.3.5. Base récursive

Soit la somme $D = D_1 + D_2 + \dots + D_n$, D_i domaines $(i = 1,\dots,n)$, soient B_i $(i=1,\dots,n)$ les bases de chaque $D_i (i=1,\dots,n)$, soient $I_i (i=1,\dots,n)$ les ensembles récursifs des indices associés aux bases B_i, et soit $B = B_1 + B_2 + \dots + B_n$ (comme au § 3.2.3.4.) ; pour trouver l'ensemble récursif d'indices associés à B, il suffit de trouver une fonction de codification qui soit injective et récursive de $\mathbb{N} \times n$ dans $\mathbb{N}$; par exemple :

$$\tau(i,j) = (i-1) \times n + j \qquad (\text{où } i \in I_j \wedge 1 \leq j \leq n)$$

Avec cette série de définitions données sur la somme $D = D_1 + D_2 + \dots + D_n$, il est facile de démontrer le lemme suivant.

3.2.4. Lemme

Soient $D_1, D_2, \dots D_n$ des treillis complets (-continus)(-algébriques)(domaines de calcul) alors, $D_1 + D_2 + \dots + D_n$ est un treillis complet (-continu)(-algébrique)(domaine de calcul).

3.3. Espace des fonctions continues ou opération d'exponentiation

Dans ce paragraphe, nous nous limitons à présenter l'espace des fonctions continues entre les domaines - treillis algébriques - ou domaines de calcul - treillis algébriques avec base récursive -.

Soient D et D' des domaines, nous dénotons l'espace des fonctions continues de D en D' par $[D \rightarrow D']$. Par conséquent, si $f : D \rightarrow D'$ est continue, on a : $f \in [D \rightarrow D']$.

3.3.1. Définitions en $[D \rightarrow D']$

Les définitions suivantes caractérisent $[D \rightarrow D']$ comme un domaine (de calcul) si D et D' le sont.

3.3.1.1. Relation d'ordre partiel ($\sqsubseteq''$)

Soient $D(\sqsubseteq, \perp, \top, \sqcup, \sqcap)$ et $D'(\sqsubseteq', \perp', \top', \sqcup', \sqcap')$ des treillis algébriques - domaines - ou domaines de calcul, et soit $[D \rightarrow D']$ l'espace des fonctions continues entre D et D'. La relation d'ordre partiel ($\sqsubseteq''$) en $[D \rightarrow D']$ est donnée par :

$$(\forall f)(\forall g)\ [f \in [D \rightarrow D'] \wedge g \in [D \rightarrow D'] \Rightarrow$$

$$f \sqsubseteq'' g \Leftrightarrow (\forall x \in D)\ f(x) \sqsubseteq' g(x)]$$

3.3.1.2. Borne supérieure et inférieure - $\top$ et $\perp$.

Soit $F \subseteq [D \rightarrow D']$, on a :

- $\sqcup'' F = \lambda x.\sqcup' \{f(x) \mid x \in D \wedge f \in F\}$
- $\sqcap'' F = \lambda x.\sqcup' \{g(x) \mid x \in D \wedge g \sqsubseteq'' F\}$ (théorème § 1.3.2.)
- $\top'' = \lambda x.\top'$ $(x \in D)$
- $\perp'' = \lambda x.\perp'$ $(x \in D)$

Il est clair que ces définitions satisfont les propriétés des bornes ; le point intéressant est que les fonctions ainsi définies sont aussi continues. Il est immédiat que $\top''$ et $\bot''$ sont des fonctions continues, puisque ce sont des fonctions constantes. Avec le lemme suivant, on voit que $\bigsqcup''F$ est aussi continue.

Lemme : Soient $F \subseteq [D \to D']$, $D(\sqsubseteq, \bot, \top, \bigsqcup, \bigsqcap)$, $D'(\sqsubseteq', \bot', \top', \bigsqcup', \bigsqcap')$;
et soit $\bigsqcup''F = \lambda x.\bigsqcup'\{f(x) \mid x \in D \wedge f \in F\}$,
on a :
$(\forall X)[X \text{ dirigé} \wedge X \subseteq D \Rightarrow \bigsqcup'' F(\bigsqcup X) = \bigsqcup'(\bigsqcup'' F(X))]$

Démonstration :

$$\begin{aligned}
\bigsqcup''F(\bigsqcup X) &= \bigsqcup'\{f(\bigsqcup X) \mid f \in F\} \\
&= \bigsqcup'\{\bigsqcup' f(X) \mid f \in F\} && \text{puisque f continue } (\forall f \in F) \\
&= \bigsqcup'(\bigcup_{f \in F} f(X)) && \text{théorème 4 § 1.1.5.} \\
&= \bigsqcup'(\bigcup_{\substack{f \in F \\ x \in X}} \{f(x)\}) && \text{puisque } f(X) = \{f(x) \mid x \in X\} = \bigcup_{x \in X} \{f(x)\} \\
&= \bigsqcup'(\bigcup_{x \in X} \{f(x) \mid f \in F\}) \\
&= \bigsqcup'\{\bigsqcup'\{f(x) \mid f \in F\} \mid x \in X\} && \text{théorème 4, § 1.1.5.} \\
&= \bigsqcup'\{\bigsqcup''F(x) \mid x \in X\} \\
&= \bigsqcup'(\bigsqcup'' F(X))
\end{aligned}$$

cqfd.

Corollaire : $\bigsqcap''F = \bigsqcup''G$ $\quad (G = \{g \mid g \in [D \to D'] \wedge g \sqsubseteq'' F\})$ est continue.

Dans ce paragraphe, nous avons montré que $[D \to D']$ (D et D' domaines ou treillis complets) est un treillis complet. Maintenant, nous verrons quels sont les éléments finis de $[D \to D']$ à partir des éléments finis de D et D'.

3.3.1.3. Elément compact ou fini.

Soient $D(\sqsubseteq, \bot, \top, \bigsqcup, \bigsqcap)$ et $D'(\sqsubseteq', \bot', \top', \bigsqcup', \bigsqcap')$ des domaines (treillis algébriques) ; soient $B = \{e \mid e \in D \wedge e \ll e\}$ et $B' = \{e' \mid e' \in D' \wedge e' \ll e'\}$
les ensembles d'éléments finis de chaque domaine ; et soit l'ensemble de fonctions :

$$S = \{\langle e;e' \rangle \mid e \in B \wedge e' \in B' \wedge \langle e;e' \rangle : D \to D'\}$$

où chaque fonction $\langle e;e' \rangle$ est définie de la manière suivante :

$$(\forall x \in D) \quad \langle e;e' \rangle(x) = \begin{cases} e' & \text{si } e \sqsubseteq x \\ \bot' & \text{sinon} \end{cases}$$

On a les lemmes suivants :

Lemme 1 : $S \subseteq [D \to D']$ ($[D \to D']$ espace des fonctions continues)

c'est-à-dire, toutes les fonctions dans S sont continues :

$$(\forall X)[X \text{ dirigé} \wedge X \subseteq D \Rightarrow \langle e;e'\rangle(\sqcup X) = \sqcup' \langle e;e'\rangle (X)]$$

Démonstration : (il y a deux cas)

(1) $(\sqcup X \sqsubseteq e \wedge \sqcup X \neq e) \vee (e \| \sqcup X)$ il est immédiat que $\langle e;e'\rangle(\sqcup X) = \bot'$ et $(\forall x \in X)$ $\langle e;e'\rangle (x) = \bot'$

(2) $(e \sqsubseteq \sqcup X)$

comme $e \ll e$ alors $(\exists x \in X)$ tel que $e \sqsubseteq x$

et comme $(\forall x)$ $[e \sqsubseteq x \Rightarrow \langle e;e'\rangle (x) = e']$

alors $\langle e;e'\rangle (\sqcup X) = \sqcup' \langle e;e'\rangle (X) = e'$

et par conséquent :

$$S \subseteq [D \to D']$$

cqfd.

Avec le lemme suivant, nous donnons une caractérisation de l'ordre partiel des fonctions continues par rapport aux éléments de S :

Lemme 2 :

$$(\forall \langle e;e'\rangle)(\forall f) [\langle e;e'\rangle \in S \subseteq [D \to D'] \wedge f \in [D \to D']$$
$$\Rightarrow \langle e;e'\rangle \sqsubseteq'' f \Leftrightarrow e' \sqsubseteq' f(e)]$$

Démonstration :

$(\Rightarrow)$ immédiat, puisque $e' = \langle e;e'\rangle(e) \sqsubseteq' f(e)$

$(\Leftarrow)$ (il y a deux cas)

(1) $(x \sqsubseteq e \wedge x \neq e) \vee x \| e$, comme $\langle e;e'\rangle (x) = \bot'$ alors $\langle e;e'\rangle (x) \sqsubseteq' f(x)$

(2) $(e \sqsubseteq x)$ comme $\langle e;e'\rangle (x) = e' \wedge f$ monotone (continue)

$\wedge\ e' \sqsubseteq f(e) \Rightarrow \langle e;e'\rangle (x) = e' \sqsubseteq' f(e) \sqsubseteq' f(x)$

de (1) et (2) on a :

$$(\forall x) \quad \langle e;e'\rangle (x) \sqsubseteq' f(x)$$

et par conséquent, $\langle e;e'\rangle \sqsubseteq'' f$

cqfd.

Lemme 3 : $(\forall F)(\forall x)$ $[F \text{ dirigé} \wedge F \subseteq [D \to D'] \wedge x \in D$

$\Rightarrow F(x) = \{f(x) | f \in F\}$ est un dirigé en $D']$

c'est-à-dire, l'image en un point d'un ensemble dirigé de fonctions continues est aussi un dirigé.

Démonstration :

La définition de dirigé (§ 1.2.1.) est :

F dirigé $\Leftrightarrow (\forall f)(\forall g)(\exists h)[f \in F \wedge g \in F \wedge h \in F \wedge f \sqsubseteq'' h \wedge g \sqsubseteq'' h]$

alors, pour tout $x \in D$, on a :

$$(\forall y)(\forall z)(\exists v)[y \in F(x) \text{ (c.à.d. } y = f(x)) \wedge z \in F(x) \text{ (c.à.d. } z = g(x))$$
$$\wedge\ v \in F(x) \text{ (c.à.d. } v = h(x)) \wedge y \sqsubseteq' v \wedge z \sqsubseteq' v]$$

par conséquent,

$F(x)$ dirigé en D' ($x \in D$)

cqfd.

Lemme 4 : $(\forall \langle e;e'\rangle)[\langle e;e'\rangle \in S \Rightarrow \langle e;e'\rangle$ fini $(\langle e;e'\rangle \ll'' \langle e;e'\rangle)]$

Démonstration :

Soit $F \subseteq [D \to D'] \wedge F$ dirigé

et soit $\langle e;e'\rangle \sqsubseteq'' \bigsqcup F$

alors, $e' \sqsubseteq' F(e)$ (lemme 2)

et $F(e) = \{f(e) \mid f \in F\}$ dirigé (lemme 3)

mais

e' fini $\Rightarrow (\exists \hat{f}(e))[\hat{f}(e) \in F(e) \wedge e' \sqsubseteq' \hat{f}(e)]$

et $e' \sqsubseteq' \hat{f}(e) \Rightarrow \langle e;e'\rangle \sqsubseteq'' \hat{f}$ (lemme 2)

par conséquent,

$\hat{f} \in F \wedge \langle e;e'\rangle \sqsubseteq'' \hat{f} \Rightarrow \langle e;e'\rangle$ fini

cqfd.

Tous les éléments finis de l'espace des fonctions continues $[D \to D']$ n'appartiennent pas à S. Par exemple, la fonction suivante :

$$(\langle e;e'\rangle \cup \langle \hat{e};\hat{e}'\rangle)\ (x) = \begin{cases} e' \cup \hat{e}' & \text{si } e \cup \hat{e} \sqsubseteq x \\ e' & \text{si } e \neq \hat{e} \wedge e \sqsubseteq x \sqsubseteq \hat{e} \wedge x \neq \hat{e} \\ \hat{e}' & \text{si } e \neq \hat{e} \wedge \hat{e} \sqsubseteq x \sqsubseteq e \wedge x \neq e \\ \perp' & \text{sinon} \end{cases}$$

il est clair que $\langle e;e'\rangle \cup \langle \hat{e};\hat{e}'\rangle$ est finie puisque $\langle e;e'\rangle$ et $\langle \hat{e};\hat{e}'\rangle$ le sont, c'est-à-dire :

$x \ll x \wedge y \ll y \Rightarrow x \cup y \ll x \cup y$ (de § 1.2.6. c et e)

et cette fonction $(\langle e;e'\rangle \cup \langle \hat{e};\hat{e}'\rangle)$ n'appartient pas à S quand $e \neq \hat{e}$.

3.3.1.4. Base

Bien que tout élément fini de $[D \to D']$ n'appartienne pas à S, S constitue une sous-base de l'espace des fonctions continues ; c'est-à-dire :

Lemme :

Soient $D(\sqsubseteq, \bot, \top, \sqcup, \sqcap)$ et $D'(\sqsubseteq', \bot', \top', \sqcup', \sqcap')$ des domaines (treillis algébriques) ; soient $B \subseteq D$ et $B' \subseteq D'$ les ensembles des éléments finis de chaque domaine ; et soit $S \subseteq [D \to D']$ le sous-ensemble d'éléments finis (fonctions finies) comme nous l'avons défini au paragraphe précédent ; alors :

$(\forall f)[f \in [D \to D'] \Leftrightarrow f = \sqcup''\{<e;e'> \mid <e;e'> \in S \wedge <e,e'> \sqsubseteq'' f\}]$

Démonstration :

$(\Leftarrow)$ immédiat, puisque $\sqcup''\{g \mid g \in S' \subseteq S \subseteq [D \to D']\}$ est continue (lemme §3.3.1.2.)

$(\Rightarrow)$ on a $f \in [D \to D']$ (c.à.d. f est continue) ssi

(1) $(\forall x \in D)(\forall e' \in B')(\exists e \in B)[e' \sqsubseteq' f(x) \Leftrightarrow e \sqsubseteq x \wedge e' \sqsubseteq' f(e)]$ (§2.2.2.)

par le lemme 2 (§3.3.1.3.) nous avons :

$$<e;e'> \sqsubseteq'' f \Leftrightarrow e' \sqsubseteq' f(e)$$

alors, par (1) :

$(\forall e')(\exists e)\ [e' \sqsubseteq' f(x) \Leftrightarrow <e;e'> \sqsubseteq'' f \wedge e \sqsubseteq x]$

par conséquent :

$$f(x) = \sqcup'\{e' \mid e' \sqsubseteq' f(x)\} = \sqcup'\{<e;e'>(x) \mid <e;e'> \sqsubseteq'' f \wedge e \sqsubseteq x\}$$

comme $<e;e'>(x) = \bot'$ si $(x \sqsubseteq e \wedge x \neq e) \vee x \| e$, nous avons :

$$f(x) = \sqcup'\{<e;e'>(x) \mid <e;e'> \sqsubseteq'' f\}$$

c'est-à-dire,

$$f = \lambda x. \sqcup'\{<e;e'>(x) \mid <e;e'> \sqsubseteq'' f\}$$

alors, par la définition de la borne supérieure (§3.3.1.2.) :

$$f = \sqcup'' S_f \wedge S_f = \{<e;e'> \mid <e;e'> \in S \wedge <e;e'> \sqsubseteq'' f\}$$

cqfd.

Avec ce lemme, nous avons montré que toute fonction continue est la borne supérieure d'un sous-ensemble de fonctions finies continues dans S ; autrement-dit, S "engendre" l'espace des fonctions continues.

Une base pour l'espace des fonctions continues $[D \to D']$ où D et D' sont des domaines, est la suivante :

$$E = \{\sqcup\hat{S} \mid \hat{S} \subseteq S \wedge \hat{S} \text{ fini}\}$$

Il est clair que E satisfait les conditions d'une base ; de plus : $S \subseteq E \subseteq [D \to D']$. C'est-à-dire, les éléments de E sont finis et continus.

3.3.1.5. Base récursive

Soient B et B' des bases récursives (§1.4.4.) des domaines de calcul D et D' respectivement (c.à.d. $B \subseteq D \wedge B' \subseteq D'$) ; et soient I et I' les ensembles récursifs d'indices associés aux bases B et B'. L'ensemble récursif J associé à $S = \{<e_n;e'_m> \mid e_n \in B \wedge n \in I \wedge e'_m \in B' \wedge m \in I'\}$ est défini de la manière suivante : à chaque $<e_n;e'_m>$ on associe l'indice $\tau_2(n,m)$ (τ_2 est la fonction de codification en § 3.1.1.5.) ; par conséquent :

$$J = \{\tau_2(n,m) \mid n \in I \wedge m \in I'\}$$

Pour les éléments $e \in E = \{\sqcup\hat{S} \mid \hat{S} \subseteq S \wedge \hat{S} \text{ fini}\}$, l'indice associé k est le suivant :

. soit $\hat{J}$ le sous-ensemble (fini) de J associé à $\hat{S}$

. alors $e_k = \sqcup \hat{S} \wedge k = \sum_{i \in \hat{J}} 2^i$

3.3.1.6. Fonction calculable

Ici, nous répétons la définition présentée au § 1.4.5., mais dans ce cas-ci, l'élément calculable est une fonction continue, et nous employons le concept de sous-base :

Soient D et D' des domaines de calcul, S la sous-base récursive (§ 3.3.1.5.) de $[D \to D']$, J l'ensemble récursif d'indices associé à S, et pour tout $f \in [D \to D']$ soit $S_f = \{<e_n;e'_m> \mid <e_n;e'_m> \sqsubseteq'' f\} \subseteq S$, tel que dans le lemme § 3.3.1.4. ; nous disons que J_f est l'ensemble d'indices associés à f ssi :

$$(\forall j)[j \in J_f \subseteq J \wedge j = \tau_2\ (n,m) \Leftrightarrow <e_n;e'_m> \in S_f] \quad ;$$

la notion de fonction continue calculable est la suivante :

$(\forall f \in [D \to D'])$ [<u>f est calculable</u> $\Leftrightarrow$ J_f est récursivement énumérable].

Avec les résultats présentés à partir du § 3.3.1., nous avons montré le lemme suivant pour les espaces de fonctions continues entre domaines (de calcul).

3.3.2. Lemme

Soient D et D' domaines (domaines de calcul) alors, $[D \to D']$, l'espace des fonctions continues entre D et D' est un domaine (domaine de calcul).

Dans cette partie du travail (§ 3), on a montré que les ensembles construits avec $\times$, $+$, et $\to$ à partir de domaines, sont aussi des domaines ; par conséquent, on a agrandi la classe des ensembles qui satisfont les propriétés des domaines. Aussi nous pouvons dire que la classe des domaines est fermée par rapport aux opérations de produit cartésien ($\times$), union disjointe ou somme ($+$) et exponentiation ($\to$).

CHAPITRE 4.
QUELQUES FONCTIONS (OPERATEURS, FONCTIONNELLES) CONTINUES DANS LES DOMAINES

4. QUELQUES FONCTIONS (OPERATEURS, FONCTIONNELLES) CONTINUES dans les DOMAINES

Les opérateurs que l'on présente dans cette partie, avec ceux du § 5, représentent les développements fondamentaux de la méthode dénotationnelle pour établir la sémantique des languages de programmation. On utilise les termes opérateurs, fonctions ou fonctionnelles indistinctement, étant donné que les domaines, comme nous l'avons vu au § 3, peuvent être des espaces de fonctions continues (produit cartésien ou somme).

4.1. La composition de fonctions

Le concept de séquence d'exécution dans les languages de programmation est associé dans la méthode dénotationnelle à la composition des fonctions. Dans cette partie, nous avons différencié, pour faciliter la suite, la composition de deux fonctions de celle de plusieurs fonctions. Après avoir montré la continuité de la composition (application) de fonctions, on définit la puissance d'une fonction, pour établir un lemme sur les ensembles dirigés de fonctions.

4.1.1. La composition de fonctions continues $\circ$

Le théorème suivant établit que l'opérateur de composition de fonctions (continues) est continu.

Théorème :

Soient . D, D', D'' des domaines,

. (la composition) $\circ : ([D \to D'] \times [D' \to D'']) \to [D \to D'']$

. $\circ$ définie par :

$(\forall f)(\forall g)(\forall x)\ [f \in [D \to D'] \wedge g \in [D' \to D''] \wedge x \in D$

$$\Rightarrow \quad \circ\,(f,g) = g \circ f = \lambda x.g\,(f(x))]$$

alors : (la composition) $\circ$ est continue.

Démonstration :

D'abord, nous savons que $\circ(f,g)$ est continue (c.a.d. $g \circ f \in [D \to D'']$) ,

puisque : soit $X \subseteq D \wedge X$ dirigé

$$\begin{aligned} \text{alors, } g \circ f(\sqcup X) &= g(f(\sqcup X)) \\ &= g(\sqcup' f(X)) \quad \text{(f continue)} \\ &= \sqcup''(g(f(X)) \quad \text{(g continue et f(X) dirigé - lemme 2 § 2.1.1.)} \\ &= \sqcup''(g \circ f(X)) \end{aligned}$$

(Dans ce qui suit, nous dénotons par $\sqcup_1, \sqcup_2, \sqcup_3$ les bornes supérieures dans $[D \to D']$, $[D' \to D'']$, $[D \to D'']$ respectivement).

Soit $F \subseteq [D \to D'] \wedge F$ dirigé

$$\begin{aligned} \text{alors, } (g \circ \sqcup_1 F)(x) &= g((\sqcup_1 F)(x)) \\ &= g(\sqcup' \{f(x) \mid f \in F\}) \quad \text{(§ 3.3.1.2.)} \\ &= \sqcup'' \{g(f(x)) \mid f \in F\} \quad (\{f(x) \mid f \in F\} \text{ dirigé - L3 - §3.3.1.3.)} \\ &= \sqcup_3 \{g \circ f \mid f \in F\} (x) \quad \text{(§3.3.1.2.)} \end{aligned}$$

Soit $G \subseteq [D' \to D'']$

$$\begin{aligned} \text{alors, } (\sqcup_2 G \circ f)(x) &= \sqcup_2 G(f(x)) \\ &= \sqcup'' \{g(f(x)) \mid g \in G\} \\ &= \sqcup_3 \{g \circ f \mid g \in G\} (x) \end{aligned}$$

(notons que la composition est additive en g)

par conséquent (théorème § 3.1.3.)

(la composition) $\circ$ est continue cqfd.

4.1.2. Composition de n fonctions ($\circ_n$)

Le théorème suivant est une généralisation naturelle du théorème précédant. Nous n'en donnons pas la démonstration (qui utilise l'induction simple).

Théorème

Soient
- . $n \in \mathbb{N} \wedge n \geq 1$
- . $D_1, D_2, \ldots D_{n+1}$, domaines
- . (la composition n) $\circ_n : ([D_1 \to D_2] \times \ldots \times [D_n \to D_{n+1}]) \to [D_1 \to D_{n+1}]$

ou $\circ_n : \overset{n}{\underset{i=1}{\times}} [D_i \to D_{i+1}] \to [D_1 \to D_{n+1}]$

. avec la définition récursive suivante :

$$\circ_1(f) = f \qquad (f \in [D_1 \to D_2])$$

$$\circ_{n+1}(<f_1, f_2 \dots f_n>) = \circ(\circ_{n-1}(<f_1, \dots f_{n-1}>), f_n)$$

$$f_i \in [D_i \to D_{i+1}] \quad (i-1, \dots n)$$

autrement dit :

$$\circ_n(<f_1, f_2, \dots f_n>) = f_n \circ(f_{n-1} \circ(\dots \circ(f_2 \circ f_1)\dots))$$

$$= \lambda x.f_n(\dots (f_2(f_1(x)))\dots) \quad (x \in D_1)$$

alors :(1) $\circ_n(<f_1, f_2, \dots, f_n>) \in [D_1 \to D_{n+1}]$ est continue et

(2) $\circ_n$ est continue

c'est-à-dire (2) :

$$F \subseteq \overset{n}{\underset{i=1}{X}} [D_i \to D_{i+1}] \wedge F \text{ dirigé} \Rightarrow \circ_n(\sqcup F) = \sqcup(\circ_n(F))$$

4.1.3. Puissance d'une fonction (f^n)

On définit la puissance n ($n \geq 1$) d'une fonction continue f de la manière suivante :

$$f^n = \circ_n(<f, f, \dots f>) \qquad (f \in [D \to D])$$

c'est-à-dire, la fonction appliquée n fois.

Le lemme suivant établit que la puissance n de la borne supérieure d'un ensemble dirigé F de fonctions continues, est égal à la borne supérieure d'un ensemble dirigé de fonctions continues, et dans cet ensemble de fonctions, chaque fonction est la puissance n d'une fonction en F.

Lemme :

Soit D un domaine ,

(1) $f \in [D \to D] \Rightarrow f^n \in [D \to D]$ (f^n continue).

(2) $(\forall F)\ [F \subseteq [D \to D] \wedge F$ dirigé

$\Rightarrow$ (a) $\{f^n \mid f \in F\}$ dirigé

et (b) $(\sqcup'' F)^n = \sqcup'' \{ f^n \mid f \in F\}$

Démonstration :

(1) immédiat $(f^n = \circ_n(<f,\dots f>))$

(2) (a) comme F est dirigé, nous avons :

$(\forall f)(\forall g)(\exists h)[f \in F \wedge g \in F \wedge h \in F \wedge f \sqsubseteq'' h \wedge g \sqsubseteq'' h]$

De plus $f^n \sqsubseteq'' h^n \wedge g^n \sqsubseteq'' h^n$ (par continuité - induction)

par conséquent :

$\{f^n \mid f \in F\}$ est dirigé

(2) (b) comme $\circ_n$ est continue, nous avons :

$(\sqcup''F)^n = \sqcup''\{\circ_n(<f_1,f_2,\dots f_n>) \mid <f_1,\dots f_n> \in F^n\}$

$= \sqcup''\circ_n(F^n)$

mais $\{f^n \mid f \in F\} \subseteq \circ_n(F^n)$

alors $\sqcup'' \{f^n \mid f \in F\} \sqsubseteq'' (\sqcup''F)^n$ (théorème 3 §1.1.5.)

comme $\{f^n \mid f \in F\}$ est dirigé, nous avons :

pour tout $\circ_n(< f_1,f_2,\dots f_n >) \in \circ_n(F^n)$

il existe $j^n \in \{f^n \mid f \in F\}$ tel que $\circ_n(< f_1,f_2,\dots f_n >) \sqsubseteq'' j^n$

c'est-à-dire : $f_i \sqsubseteq'' j \wedge f_i \in F \wedge j \in F$ $(i=1,\dots n)$

alors (théorème 2 § 1.1.5.) : $\sqcup''\circ_n(F^n) \sqsubseteq \sqcup'' \{f^n \mid f \in F\}$

par conséquent : $(\sqcup''F)^n = \sqcup'' \{f^n \mid f \in F\}$ cqfd.

En particulier, nous avons le corollaire suivant :

Corollaire :

Soit $E \subseteq [D \to D]$ la base de l'espace des fonctions continues dans le domaine D, alors :

$(\forall f)\ [f \in [D \to D] \Rightarrow f^n = \sqcup''\{e^n \mid e \in E \wedge e \sqsubseteq'' f\}]$

4.2. Plus petit point fixe d'une fonction continue

Par le théorème de Tarski (§2.3.1.), toute fonction monotone définie dans un treillis complet a un ensemble non vide de points fixes qui est aussi un treillis complet ; donc il existe un point fixe minimal :

la borne inférieure du treillis complet de points fixes. On présente maintenant un opérateur qui fournit ce plus petit point fixe, mais avec l'hypothèse que la fonction est continue. Cet opérateur est défini sur la base d'un schéma itératif, et a son origine dans le premier théorème de récursion de Kleene ([KlS52],[CoPR77]). Dans l'introduction, nous avons mentionné que la notion de point fixe est associée au concept de récursion dans les languages de programmation.

4.2.1. Opérateur de plus petit point fixe (par rapport à $\perp$) d'une fonction continue (μ ou fix ou Y) ([LiC78],[StJ77])

Théorème :

Soient
- . $D(\sqsubseteq, \perp, \top, \sqcup, \sqcap)$ domaine (ou treillis complet)
- . $[D \to D]$ l'espace des fonctions continues
- . $\mu : [D \to D] \mapsto D$
- . μ définie par :

$$(\forall f)\ [f \in [D \to D] \Rightarrow \mu(f) = \sqcup(\bigcup_{n=0}^{\infty} \{f^n(\perp)\})]$$

où $f^0 = I$ (c.a.d. $(\forall x \in D)\ [f^0(x) = x]$)

alors, (a) $\mu(f) = f(\mu(f))$ ($\mu(f)$ point fixe de f)

(b) $(\forall x)\ [x \in D \wedge x = f(x) \Rightarrow \mu(f) \sqsubseteq x]$

($\mu(f)$ est le plus petit point fixe)

(c) μ est continue.

Démonstration :

(a) comme $f^0(\perp) = \perp \sqsubseteq f(\perp)$

on a : $f^n(\perp) \sqsubseteq f^{n+1}(\perp)$ $(\forall n)$ (puisque f continue, monotone)

alors,

(1) $\bigcup_{n=0}^{\infty} \{f^n(\perp)\}$ est une chaine (dirigé)

de plus :

(2) $\sqcup(\bigcup_{n=0}^{\infty} \{f^n(\perp)\}) = \sqcup(\bigcup_{n=1}^{\infty} \{f^n(\perp)\})$ (puisque $f^0(\perp) = \perp \sqsubseteq f^n(\perp) (\forall n \geq 0)$)

$= \sqcup(\bigcup_{n=0}^{\infty} \{f(f^n(\perp))\})$

$= f(\sqcup(\bigcup_{n=0}^{\infty} \{f^n(\perp)\}))$ (puisque f continue et (1)) par conséquent :

$\mu(f) = f(\mu(f))$

(b) soit $x = f(x)$,

comme $f^0(\perp) = \perp \sqsubseteq x = f(x)$

on a : $f^n(\perp) \sqsubseteq f^{n+1}(x) = x \qquad (\forall n \geq 0)$

alors, x est un majorant de $\bigcup_{n=0}^{\infty} \{f^n(\perp)\}$

par conséquent :,

$\mu(f) = \bigsqcup(\bigcup_{n=0}^{\infty} \{f^n(\perp)\}) \sqsubseteq x$

(c) ($\bigsqcup'$ dénote la borne supérieure dans $[D \to D]$)

Soit $F \subseteq [D \to D] \wedge F$ dirigé

alors, $\mu(\bigsqcup' F) = \bigsqcup(\bigcup_{n=0}^{\infty} \{(\bigsqcup' F)^n(\perp)\})$

$$
\begin{aligned}
\mu(\textstyle\bigsqcup' F) &= \textstyle\bigsqcup(\bigcup_{n=0}^{\infty} \{\bigsqcup'\{f^n \mid f \in F\}(\perp)\}) && \text{(lemme § 4.1.3)}\\
&= \textstyle\bigsqcup(\bigcup_{n=0}^{\infty} \{\bigsqcup\{f^n(\perp) \mid f \in F\}\}) && \text{(définition de la borne § 3.3.1.2)}\\
&= \textstyle\bigsqcup(\bigcup_{n=0}^{\infty} \{f^n(\perp) \mid f \in F\}) && \text{(théorème 4 § 1.1.5)}\\
&= \textstyle\bigsqcup(\bigcup_{n=0}^{\infty} (\bigcup_{f \in F} \{f^n(\perp)\}))\\
&= \textstyle\bigsqcup(\bigcup_{f \in F} (\bigcup_{n=0}^{\infty} \{f^n(\perp)\}))\\
&= \textstyle\bigsqcup\{\bigsqcup(\bigcup_{n=0}^{\infty} \{f^n(\perp)\}) \mid f \in F\} && \text{(théorème 4 § 1.1.5)}\\
&= \textstyle\bigsqcup\{\mu(f) \mid f \in F\} = \bigsqcup \mu(F)
\end{aligned}
$$

cqfd.

4.2.2. Opérateur de plus petit point fixe (par rapport à un élément fini) d'une fonction continue (Y_e) ([BrF 81])

L'opérateur (continu) suivant fournit le point fixe le plus proche (supérieurement) d'un élément fini e du domaine, si et seulement si la fonction est extensive en e (c.a.d. $e \sqsubseteq f(e)$). Cet opérateur, dû à D. Park, est semblable à μ.

Théorème :

Soient :
- . $D(\sqsubseteq, \perp, \top, \sqcup, \sqcap)$ domaine,
- . $E = \{e \mid e \in D \wedge e \ll e\} \subseteq D$ les éléments finis de D,
- . $[D \to D]$ l'espace des fonctions continues,
- . $Y_e : [D \to D] \to D$

. Y_e définie par :

$(\forall f)(\forall e)$ $[f \in [D \to D] \wedge e \in E$

$$\Rightarrow \; Y_e(f) = \begin{cases} \bigsqcup\left(\bigcup_{n=o}^{\omega} \{f^n(e)\}\right) & \text{ssi} \quad e \sqsubseteq f(e) \\ \mu(f) \quad (= Y_\perp(f)) & \text{sinon} \end{cases} \;]$$

alors :

(a) $Y_e(f) = f(Y_e(f))$

et (b1) $(\forall x)$ $[x \in D \wedge e \sqsubseteq x \wedge x = f(x) \wedge e \sqsubseteq f(e) \Rightarrow Y_e(f) \sqsubseteq x]$

et (b2) $(\forall x)$ $[x \in D \wedge x = f(x) \wedge \neg(e \sqsubseteq f(e)) \Rightarrow Y_e(f) \sqsubseteq x]$

et (c) Y_e est continue

et (d) Y_e (f) est fini dans le treillis complet FIX(f).

Démonstration :

La démonstration de (a) (b1) (b2) (c) est identique à celle de μ (théorème § 4.2.1) ; notons que $F_e = \{x \mid x \in D \wedge D \text{ domaine} \wedge e \sqsubseteq x\}$ est un domaine (lemme 2 § 1.3.5), de plus : $e \sqsubseteq f(e) \Leftrightarrow f(F_e) \subseteq F_e$.

(d) $Y_e(f)$ est fini en FIX(f)

e fini $\Leftrightarrow (\forall S)(\exists s)[\, s \in S \subseteq D \wedge S$ dirigé $\wedge\, e \sqsubseteq \bigsqcup S \Rightarrow e \sqsubseteq s\,]$

en particulier $(\forall S)[S \subseteq \text{FIX}(f) \subseteq D \wedge S$ dirigé$]$

il existe $s' \in S \wedge e \sqsubseteq s'$, c'est-à-dire $s' \in F_e$

mais (par (b1) et (b2)) $Y_e(f) \sqsubseteq \text{FIX}(f) \cap F_e$

par conséquent ,

$Y_e(f) \sqsubseteq s'$, $Y_e(f)$ fini en FIX(f) cqfd.

Cet opérateur permet de trouver les éléments finis dans le sous-treillis complet FIX(f) $(\sqsubseteq, \perp', \top', \sqcup', \sqcap')$ du domaine $D(\sqsubseteq, \perp, \top, \sqcup, \sqcap)$. Le corollaire suivant caractérise l'union de deux éléments finis, qui est aussi un élément fini (c.a.d. : $x \ll x \wedge y \ll y \Rightarrow x \cup y \ll x \cup y$, § 1.2.6, b et e)

Corollaire :

Soient D,E,f,Y_e du théorème précédent, et $\text{FIX}(f)(\sqsubseteq, \perp', \top', \sqcup', \sqcap')$ le sous-treillis complet du domaine $D(\sqsubseteq, \perp, \top, \sqcup, \sqcap)$ alors :

$(\forall e_1)(\forall e_2)$ $[e_1 \in E \wedge e_2 \in E \Rightarrow Y_{e_1}(f) \cup' Y_{e_2}(f) = Y_{e_1 \cup e_2}(f)]$

Démonstration :

Il suffit de montrer cela pour le cas $e_1 \sqsubseteq f(e_1) \wedge e_2 \sqsubseteq f(e_2)$, et pour ce cas, on remarque que :

$$e_1 \sqsubseteq f(e_1) \sqsubseteq f(e_1 \cup e_2) \wedge e_2 \sqsubseteq f(e_2) \sqsubseteq f(e_1 \cup e_2) \Rightarrow e_1 \cup e_2 \sqsubseteq f(e_1 \cup e_2)$$

cqfd.

4.3. Caractérisation de l'espace des fermetures continues sur un domaine

L'opérateur que nous présentons dans ce paragraphe, et ceux que nous donnerons dans les paragraphes suivants (§ 4.4. et § 4.5.), ont comme objectif d'établir un "calcul" des fermetures (§ 2.3.3) ou rétracts à caractère fini (§ 2.3.4). Comme nous l'avons vu, cette classe de fonctions caractérisent des domaines et correspondent au concept de "type" dans les langages de programmation (ensembles de valeurs, ex : Nat § 2.1.2). L'idée du calcul des rétracts à caractère fini est d'obtenir la solution des équations de domaines qui se posent dans la méthode dénotationnelle, ainsi que nous l'avons présenté dans l'introduction. Cette sorte d'équations apparait aussi dans la définition de types plus complexes. Avec les domaines réflexifs (§5) nous donnerons un outil fondamental pour la solution d'équations de domaines. Ensuite nous verrons comment obtenir les fermetures, et que celles-ci forment un domaine.

Soit R le sous-ensemble de fonctions continues définies sur un domaine D, qui sont des rétractions, c'est-à-dire :

$$R = \{f \mid f \in [D \to D] \wedge f^2 = f\}$$

R est un treillis complet avec la relation d'ordre ($\sqsubseteq'$) définie sur $[D \to D']$ (§ 3.3.1.1) puisque par le théorème de Tarski (§ 2.3.1), on a :

$$R = \text{Image }(r) = \text{FIX}(r)$$

où

$$r = [D \to D] \to [D \to D]$$

est définie par

$$r = \lambda f.\ f \circ f \qquad (\forall f \in [D \to D])$$

On a montré que R n'est pas un domaine (ni treillis algébrique, ni continu)(cf. [ScD 76]). Cependant, nous verrons que si r est restreinte à l'espace des fonctions extensives (c.a.d. $(\forall x)\ x \sqsubseteq f(x)$), alors FIX(r) est un domaine ; autrement dit, les fermetures forment un domaine.

4.3.1. Opérateur d'extension

Soit ε l'opérateur suivant :

$\varepsilon : [D \to D] \to [D \to D]$

$\varepsilon(f) = \sqcup' \{I, f\}$ ($\sqcup'$ borne supérieure en $[D \to D]$)

c'est-à-dire :

$$\varepsilon(f)(x) = \begin{cases} f(x) & \text{si } x \sqsubseteq f(x) \\ x & \text{sinon} \end{cases}$$

$\varepsilon(f)$ est une fonction extensive ; de plus elle est continue (§ 3.3.1.2); ε est aussi continue (exemple 1 § 3.1.3), et est une fermeture :

$(\forall f)\ [f \in [D \to D] \Rightarrow f \sqsubseteq' \varepsilon(f) \wedge \varepsilon(\varepsilon(f)) = \varepsilon(f)]$.

Par conséquent (théorème § 2.3.3), ε caractérise un domaine : le domaine des fonctions continues extensives ; c'est-à-dire :

Image (ε) = FIX(ε) = $\{f \mid f \in [D \to D] \wedge I_D \sqsubseteq' f\}$

est un domaine (treillis algébrique).

Nous noterons ce domaine par $[D \to D]_\varepsilon$ = FIX$(\varepsilon) \subseteq [D \to D]$

Le fait qu'une fonction f soit extensive implique que $f \sqsubseteq' f^2$; une fonction g qui satisfait cette propriété (c.a.d. : $g \sqsubseteq' g^2$) est appelée progressive.

Notons que, dans le domaine $[D \to D]_\varepsilon$, l'élément minimal est la fonction identité (I_D), et l'élément maximal $(\top')$ est la fonction constante $\lambda x.\top$ $(\forall x \in D)$, et les deux fonctions sont des fermetures.

4.3.2. Opérateur qui donne toutes les fermetures (Γ)

([ScD 76],[McN 79])

Il est clair que FIX(r) (où $r = \lambda f.\ f \circ f \wedge r : [D \to D]_\varepsilon \to [D \to D]_\varepsilon$) est l'ensemble des fermetures, et est un sous-treillis complet de $[D \to D]_\varepsilon$ avec le même élément maximal $(\lambda x.\top)$ et le même élément minimal $(\bot' = I_D)$.

Nous voyons maintenant que l'ensemble des fermetures $C = FIX(r) \subseteq [D \to D]_\varepsilon$ est un domaine.

Théorème :

Soit l'opérateur Γ,

$\Gamma: [D \to D]_\varepsilon \to [D \to D]_\varepsilon$

$\Gamma = \lambda f. \sqcup'(\bigcup_{n=o}^{\infty} \{f^n\})$ ($\sqcup'$ borne supérieure en $[D \to D]_\varepsilon$)

alors :

(1) $\Gamma(f) \in [D \to D]_\varepsilon$ ($\Gamma(f)$ est continue et définie en $[D \to D]_\varepsilon$)

(2) $\Gamma(f) \in C = \{c \mid I_D \sqsubseteq' c = c^2\} = FIX(r)$

(3) Γ est continue

(4) Γ est une fermeture

(5) image (Γ) = C est un domaine

Démonstration :

(1) est immédiat, puisque f est extensive et $[D \to D]_\varepsilon$ est domaine

(2) $\Gamma(f)\ (\Gamma(f)(x)) = \Gamma(f)\ ((\sqcup'(\bigcup_{n=o}^{\infty} \{f^n\})(x))$

$= \Gamma(f)(\sqcup(\bigcup_{n=o}^{\infty} \{f^n(x)\}))$

$= \sqcup\ (\bigcup_{n=o}^{\infty} \{\Gamma(f)(f^n(x)\ )\})$ ($\{f^n(x)\}$ dirigé et Γ (f) continue)

$= \sqcup\ (\bigcup_{n=o}^{\infty} \{\sqcup(\bigcup_{m=o}^{\infty} \{f^{n+m}(x)\})\})$

$= \sqcup\ (\bigcup_{n=o}^{\infty}\ (\bigcup_{m=o}^{\infty} \{f^{n+m}(x)\}\ ))$ (théorème 4 § 1.1.5)

$= \sqcup(\bigcup_{n=o}^{\infty} \{f^n(x)\})$

$= \sqcup'(\bigcup_{n=o}^{\infty} \{f^n\})(x) = \Gamma(f)(x) \sqsupseteq x$ (puisque f extensive)

(3) Soit F dirigé $\wedge F \subseteq [D \to D]_\varepsilon$

alors : $\Gamma(\sqcup' F) = \sqcup'(\bigcup_{n=o}^{\infty} \{\sqcup' F^n\})$

$= \sqcup'(\bigcup_{n=o}^{\infty} (\sqcup'\{f^n | f \in F\}))$ (lemme § 4.1.3)

$= \sqcup'(\bigcup_{n=o}^{\infty} \{f^n | f \in F\})$ (théorème 4 § 1.1.5)

$= \sqcup'(\bigcup_{f \in F} (\bigcup_{n=o}^{\infty} \{f^n\}))$

$= \sqcup'\{\sqcup'(\bigcup_{n=o}^{\infty} \{f^n\}) | f \in F\}$ (théorème 4 § 1.1.5)

$= \sqcup'\{\Gamma(f) | f \in F\}$

$= \sqcup' \Gamma(F)$

(4) $\Gamma(\Gamma(f)) = \sqcup'(\bigcup_{n=o}^{\infty} \{(\Gamma(f))^n\})$

$= \sqcup'(\bigcup_{n=o}^{\infty} \{\Gamma(f)\})$ (puisque, par (2), $\forall n, (\Gamma(f))^n = \Gamma(f)$)

$= \sqcup'(\Gamma(f))$

$= \Gamma(f) \supseteq f$ (puisque $f \in [D \to D]_\varepsilon$, progressive)

(5) par (2), on a :

Image $(\Gamma) \subseteq C$

il est immédiat que :

$c \in C \Rightarrow \Gamma(c) = c$ (puisque $\forall n$, $c^n = c$)

par conséquent :

Image $(\Gamma) = C$

comme Γ est une fermeture (4) et par le théorème § 2.3.3, on a que C est un domaine.

cqfd.

4.3.3. Les éléments de la base (finis) dans le domaine des fermetures C

Soit $E \subseteq [D \to D]$ l'ensemble des éléments finis en $[D \to D]$. Par les thèorèmes §2.3.3. et 4, $\varepsilon(E) \subseteq [D \to D]_\varepsilon$ sont les éléments finis dans l'espace (domaine) des fonctions continues extensives. Par les mêmes thèorèmes, les éléments finis de C (les fermetures finies) sont données par :

$E_C = \Gamma(\varepsilon(E))$

Par le théorème § 4.2.2. (partie d), on a :

$(\forall e)[e \in \varepsilon(E) \wedge f = \lambda g.g\circ g \Rightarrow Y_e(f) \in E_C]$

il est facile de voir que

$(\forall e)\ [e \in \varepsilon(E) \wedge f = \lambda g.g\circ g \Rightarrow Y_e(f) = \Gamma(e) = \sqcap_C\ (F_e \cap C)]$

4.3.4. Propriétés du domaine des fermetures $C(\sqsubseteq, \perp, \top, \sqcup, \sqcap)$

Signalons deux propriétés :

(1) $(\forall c_1)(\forall c_2)\ [c_1 \in C \wedge c_2 \in C \wedge c_1 \sqsubseteq c_2 \Rightarrow \mathrm{Image}(c_2) \subseteq \mathrm{Image}(c_1)]$

(2) $(\forall c_1)(\forall c_2)\ [c_1 \in C \wedge c_2 \in C \Rightarrow (c_1 \circ c_2 \in C \Leftrightarrow c_1 \circ c_2 = c_2 \circ c_1)]$

4.4. Points fixes d'une fonction continue, engendrés d'une manière continue

([BrF 81])

La notion générale d'un opérateur continu (F) de point fixe sur des fonctions continues (f) définies dans un domaine (D) est la suivante :

$F \in [[D \to D] \to D]$, D domaine

F <u>opérateur continu de point fixe</u> (ocpf)

$\Leftrightarrow\ (\forall f)[f \in [D \to D] \Rightarrow F(f) = f(F(f))]$

On définit l'ensemble des points fixes, engendrés continuement, d'une fonction continue f, comme :

$FEC(f) = \{x \mid x \in D \wedge\ x = F(f) \wedge F \text{ ocpf}\}$

ainsi, on a :

$FEC(f) \subseteq FIX(f) = \{x \mid x \in D \wedge x = f(x)\}$

Nous savons que FIX(f) est un treillis complet, ou un sous-treillis complet de caractère fini de D (corollaire § 2.3.1), mais non nécessairement un treillis algébrique. Cependant, FEC(f) est en vérité un domaine. Dans cette partie, nous définissons un opérateur qui donne tous les éléments de FEC(f). Le lemme suivant établit la forme que possèdent (nécessairement) les éléments de FEC(f).

4.4.1. Lemme

Soient . $D(\sqsubseteq,\bot,\top,\sqcup,\sqcap)$ domaine,

. $E\subseteq D$ la base - éléments finis - de D,

. $[D\to D](\sqsubseteq',\bot',\top',\sqcup',\sqcap')$ l'espace des fonctions continues sur D,

alors,

$$(\forall f)(\forall x)[f\in[D\to D]\wedge x\in FEC(f)$$

$$\Rightarrow\quad x=\sqcup\{e\mid e\in E\wedge e\sqsubseteq x\wedge e\sqsubseteq f(e)\}]$$

Démonstration :

Soit $\overline{E}\subseteq[D\to D]$ la base - éléments finis - de $[D\to D]$
et soit F ocpf, on a :

$$f=\sqcup'\{\overline{e}\mid\overline{e}\in\overline{E}\wedge\overline{e}\sqsubseteq'f\}$$

et $F(f)=\sqcup\{F(\overline{e})\mid\overline{e}\in\overline{E}\wedge\overline{e}\sqsubseteq'f\}$

(puisque $\{\overline{e}\mid\overline{e}\in\overline{E}\wedge\overline{e}\sqsubseteq'f\}$ dirigé et F continue)

de plus,

$$(\forall\overline{e}\in\overline{E})\ F(\overline{e})=\overline{e}(F(\overline{e}))\in E$$

(puisque F ocpf et $(\forall x)\overline{e}(x)$ est fini § 3.3.1.3)
comme $\overline{e}\sqsubseteq'f$, alors :

$$F(\overline{e})=\overline{e}(F(\overline{e}))\sqsubseteq f(F(\overline{e}))$$

c'est-à-dire $e\sqsubseteq f(e)$, où $e=F(\overline{e})\wedge e\in E$
par conséquent,

$$F(f)=\sqcup\{e\mid e\in\overline{E}\wedge e\sqsubseteq f(e)\wedge e\sqsubseteq F(f)\}$$ cqfd

4.4.2. Opérateur de points fixes engendrés continuement (Cgn)

L'opérateur qui donne les éléments de FEC(f) est le suivant :

Théorème :

Soient . $D(\sqsubseteq,\bot,\top,\sqcup,\sqcap)$ domaine,

. $E = \{e \mid e \in D \wedge e \ll e\} \subseteq D$ les éléments finis (base) de D,

. $[D \to D]$ l'espace des fonctions continues,

. $Cgn : [D \to D] \to [D \to D]$

. Cgn définie par :

$(\forall f)(\forall x)\ [f \in [D \to D] \wedge x \in D$

$\Rightarrow\ Cgn(f)(x) = \sqcup(\bigcup_{n=0}^{\infty} \{f^n(e) \mid e \in E \wedge e \sqsubseteq x \wedge e \sqsubseteq f(e)\})]$

alors, (1) $\lambda x.Cgn(f)(x)$ est continue

(2) $\lambda x.Cgn(f)(x)$ est idempotente

(3) $(\forall x)\ f(Cgn(f)(x)) = Cgn(f)(x)$

(4) $\lambda f.Cgn(f)(x)$ est continue

(5) Image $(\lambda x.Cgn(f)(x))$ = FEC(f) et $\lambda x.Cgn(f)(x)$ est un rétract à caractère fini

(6) f est un rétract à caractère fini $\Leftrightarrow$ Image $(\lambda x.Cgn(f)(x))$ = Image(f)

Démonstration :

Pour la démonstration du théorème, nous utiliserons une autre caractérisation de la fonction $\lambda x.Cgn(f)(x)$ (conséquence du théorème 4 § 1.1.5 et du § 4.2.2) donnée par :

$$\lambda f.\ \lambda x.Cgn(f)(x) = \lambda f.\ \lambda x.\sqcup\{Y_e(f) \mid e \in E \wedge e \sqsubseteq x\}$$

(1) Soit X dirigé $\wedge\ X \subseteq D$

$$\begin{aligned}
Cgn(f)(\sqcup X) &= \sqcup\{Y_e(f) \mid e \sqsubseteq \sqcup X\} \\
&= \sqcup(\bigcup_{e \sqsubseteq \sqcup X} \{Y_e(f)\}) \\
&= \sqcup(\bigcup_{x \in X} \{Y_e(f) \mid e \sqsubseteq x\}) \\
&\qquad \text{(puisque e fini} \Rightarrow \{e \mid e \sqsubseteq \sqcup X\} = \{e \mid e \sqsubseteq x \wedge x \in X\}) \\
&= \sqcup\{\sqcup\{Y_e(f) \mid e \sqsubseteq x\} \mid x \in X\} \quad \text{(théorème 4 § 1.1.5)}
\end{aligned}$$

$= \bigsqcup\{Cgn(f)(x) \mid x \in X\}$

$= \bigsqcup Cgn(f)(X)$

(2) Maintenant montrons que $\lambda x.Cgn(f)(x)$ est idempotente

$(\lambda x.Cgn(f)(x))(Cgn(f)(x)) = Cgn(f)\ (Cgn(f)(x))$

$= \bigsqcup\{Y_e(f) \mid e \subseteq \bigsqcup\{Y_e(f) \mid e \subseteq x\}\}$

Il faut montrer :

$\bigsqcup\{Y_e(f) \mid e \subseteq \bigsqcup\{Y_e(f) \mid e \subseteq x\}\} = \bigsqcup\{Y_e(f) \mid e \subseteq x\}\quad (= Cgn(f)(x))$

($\supseteq$) immédiat, puisque $\{Y_e(f) \mid e \subseteq x\} \subseteq \{Y_e(f) \mid e \subseteq \bigsqcup\{Y_e(f) \mid e \subseteq x\}\}$,
qui est dû à ce que $e \subseteq f(e) \Rightarrow e \subseteq Y_e(f)$ et par th.3 § 1.1.5

($\subseteq$) par la démonstration du corollaire § 4.2.2., on a :
$\{Y_e(f) \mid e \subseteq x\}$ est dirigé, et comme e est fini, alors
$e \subseteq \bigsqcup\{Y_e(f) \mid e \subseteq x\} \Rightarrow (\exists e')(e \subseteq Y_{e'}(f) \wedge e' \subseteq x)$

de plus : $Y_e(f) \subseteq Y_{e'}(f)$ (théorème § 4.2.2 - b1,b2)
et le résultat se déduit du théorème 2 § 1.1.5.
Par conséquent, $\lambda x.Cgn(f)(x)$ est un rétract (c.a.d. idempotent)

(3) Comme f est continue, $\{Y_e(f) \mid e \subseteq x\}$ dirigé, et par le théorème § 4.2.2, on a

$f(Cgn(f)(x)) = f(\bigsqcup\{Y_e(f) \mid e \subseteq x\})$

$= \bigsqcup\{f(Y_e(f)) \mid e \subseteq x\}$

$= \bigsqcup\{Y_e(f) \mid e \subseteq x\}$

$= Cgn(f(x))$ (c.a.d. $(\forall x)\ Cgn(f)(x)$ est point fixe de f)

(4) Soit $\bigsqcup'$ la borne supérieure dans $[D \to D]$, et soit F dirigé et $F \subseteq [D \to D]$, alors :

$Cgn(\bigsqcup' F)(x) = \bigsqcup\{Y_e(\bigsqcup' F) \mid e \subseteq x\}$

$= \bigsqcup\{\bigsqcup Y_e(F) \mid e \subseteq x\}$ (Y_e continue, th. § 4.2.2)

$= \bigsqcup(\bigcup_{e \subseteq x}\{Y_e(f) \mid f \in F\})$ (th. 4 § 1.1.5)

$= \bigsqcup(\bigcup_{f \in F}\bigcup_{e \subseteq x}\{Y_e(f)\})$

$= \bigsqcup(\bigcup_{f \in F} \{Y_e(f) \mid e \sqsubseteq x\})$

$= \bigsqcup\{ \bigsqcup\{ Y_e(f) \mid e \sqsubseteq x\} \mid f \in F\}$ (théorème 4 § 1.1.5)

$= \bigsqcup\{Cgn(f)(x) \mid f \in F\}$

$= \bigsqcup Cgn(F)(x)$

(5) par (4) et (3), nous savons que $\lambda f.Cgn(f)(x)$ est un opérateur continu de point fixe (ocpf - § 4.4), par conséquent :

Image $(\lambda x.Cgn(f)(x)) \subseteq FEC(f) \subseteq FIX(f)$

Maintenant voyons que $FEC(f) \subseteq$ Image $(\lambda x.Cgn(f)(x))$:

$x \in FEC(f) \Rightarrow x = \bigsqcup\{e \mid e \in E \wedge e \sqsubseteq x \wedge e \sqsubseteq f(e)\}$ (lemme § 4.4.1)

comme $\{e \mid e \in E \wedge e \sqsubseteq x \wedge e \sqsubseteq f(e)\}$ est dirigé, et x est point fixe de f continue, et par le lemme 2 § 2.1.1, on a :

$(\forall n \geq 0)\ x = f^n(x) = \bigsqcup\{f^n(e) \mid e \in E \wedge e \sqsubseteq x \wedge e \sqsubseteq f(e)\}$

et nous obtenons : $x \in FEC(f) \Rightarrow$

$x = \bigsqcup(\bigcup_{n=0}^{\infty} \{f^n(e) \mid e \in E \wedge e \sqsubseteq x \wedge e \sqsubseteq f(e)\})$

$= Cgn(f)(x) \in$ Image$(\lambda x.Cgn(f)(x))$

par conséquent :

Image $(\lambda x.Cgn(f)(x)) = FEC(f) \subseteq FIX(f)$

Il est facile de voir que Image$(\lambda x.Cgn(f)(x))$ est un sous-treillis algébrique de D, puisque ses éléments sont engendrés par les éléments finis de FIX(f) (c.a.d. $Y_e(f)$ théorème § 4.2.2(d)), et ces éléments finis appartiennent aussi à Image $(\lambda x.Cgn(f)(x))$. Ainsi, nous avons que $\lambda x.Cgn(f)(x)$ est un rétract à caractère fini (§ 2.3.4), et par conséquent FEC(f) est un domaine.

(6) $(\Leftarrow)$ immédiat, par (5) et puisque $FEC(f) \subseteq FIX(f) \subseteq$ Image(f)

$(\Rightarrow)$ il est clair que:

Image $(\lambda x.Cgn(f)(x)) = FEC(f) \subseteq FIX(f) =$ Image (f)

Pour montrer que : Image(f) $\subseteq$ Image $(\lambda x.Cgn(f)(x))$, il suffit de voir que les éléments finis de Image (f) (c.a.d. $x \ll' x$) correspondent aux éléments finis de Image $(\lambda x.Cgn(f)(x))$ (c.a.d. $Y_e(f)$), et cela est une conséquence du théorème § 2.3.4 cqfd.

4.4.3. Expression de Γ en fonction de Cgn

Nous montrons maintenant que le domaine des fermetures (C § 4.3) est un ensemble de points fixes engendrés continuement de la fonction $r = \lambda f.f^2$, où f appartient au domaine des fonctions extensives: $[D \to D]_\varepsilon$ - § 4.3.1.

Théorème :

Soient . $[D \to D]_\varepsilon(\sqsubseteq, \bot, \top, \sqcup, \sqcap)$ le domaine des fonctions extensives (§ 4.3.1)

. $E \subseteq [D \to D]_\varepsilon$, les éléments finis de $[D \to D]_\varepsilon$

. $r : [D \to D]_\varepsilon \to [D \to D]_\varepsilon$, $r = \lambda f.f^2$

. Γ(définie dans le théorème § 4.3.2)

alors :

$$\text{Image } (\lambda f.\text{Cgn}(r)(f)) = \text{FEC}(r) = \text{Image } (\Gamma) = C$$

Démonstration :

La démonstration est immédiate à partir des théorèmes § 4.3.2 et § 4.4.2.

De plus, il est facile de voir que :

$(\forall f)\ [f \in [D \to D]_\varepsilon \Rightarrow \lambda f.\text{Cgn}(r)(f) = \lambda f.\Gamma(f)]$ cqfd.

4.4.4. FEC($\lambda g.\ g \circ g$) : le domaine des rétracts engendrés continuement

Dans le paragraphe précédent (§ 4.4.3) nous avons vu que le domaine FEC($\lambda g.g \circ g$), où g appartient au domaine des fonctions extensives coïncide avec le domaine des fermetures. Maintenant, nous analyserons le domaine FEC($\lambda g.g \circ g$), où g appartient au domaine des fonctions continues. Nous appellerons ce domaine: le domaine des rétracts engendrés continuement. Nous verrons que les éléments de ce domaine sont des rétracts à caractère fini (th. § 4.4.4.2), et que ses éléments finis sont les limites des éléments finis de l'espace des fonctions continues qui sont progressives (§ 4.4.4.1). De plus, la fonction qui produit les points fixes engendrés continuement d'une fonction continue f (i.e.: $\lambda x.\text{Cgn}(f)(x)$) appartient à ce domaine (théorème § 4.4.4.3).

4.4.4.1. Rétracts finis

Le domaine des rétracts engendrés continuement et ses éléments finis, ou rétracts finis, sont définis en accord avec le théorème § 4.4.2 (5) de la manière suivante :

Soient . D domaine et E l'ensemble des éléments finis de D ;

. $\underline{E}$ les éléments finis du domaine des fonctions continues (c.a.d. $\underline{E} \subseteq [D \to D]$)

On a :

$$\lambda g.g \circ g : [D \to D] \to [D \to D]$$

$$FEC(\lambda g.g\circ g) = Image(\lambda f.Cgn(\lambda g.g\circ g)(f)) \quad (\text{où } f \in [D \to D])$$

et l'ensemble des éléments finis (rétracts finis) du domaine $FEC(\lambda g.g\circ g)$ est donné par :

$$\{Y_{\underline{e}}(\lambda g.g\circ g) \mid \underline{e} \in \underline{E}\} = \{\bigsqcup(\bigcup_{n=o}^{\infty} \{\underline{e}^{2^n} \mid \underline{e} \sqsubseteq \underline{e}^2\}) \mid \underline{e} \in \underline{E}\}$$

Par conséquent, chaque élément fini est de la forme :

$$(\forall w)(\exists \underline{e})\ [w \in FEC(\lambda g.g\circ g) \wedge \underline{e} \in \underline{E} \Rightarrow$$
$$(w \text{ est fini en } FEC(\lambda g.g\circ g) \Leftrightarrow w = \bigsqcup(\bigcup_{n=o}^{\infty}\{\underline{e}^{2^n} \mid \underline{e} \sqsubseteq \underline{e}^2\}))]$$

Le point intéressant ici est que les éléments finis dans le domaine des rétracts engendrés continuement peuvent être reformulés comme suit :

$$w \text{ fini en } FEC(\lambda g.g\circ g) \Leftrightarrow$$
$$(\exists \underline{e})(\exists k)\ [\underline{e} \in \underline{E} \wedge k \in \mathbb{N}$$
$$\wedge w = \bigsqcup(\bigcup_{n=o}^{\infty}\{\underline{e}^{2^n} \mid \underline{e} \sqsubseteq \underline{e}^2\}) = \underline{e}^k = \bigsqcup(\bigcup_{n=o}^{\infty}\{\underline{e}^n \mid \underline{e} \sqsubseteq \underline{e}^2\})]$$

c'est-à-dire, les éléments de la base qui sont progressifs (c.a.d. $\underline{e} \sqsubseteq \underline{e}^2$) atteignent leur limite (deviennent des rétracts) dans une puissance finie (k). Le lemme suivant nous donne ce résultat.

Lemme :

$$(\forall \underline{e})(\exists k)\ [\underline{e} \in \underline{E} \wedge \underline{e} \sqsubseteq \underline{e}^2 \wedge k \in \mathbb{N} \Rightarrow \underline{e}^k = \underline{e}^{k+1}]$$

Démonstration :

$\underline{e}$ fini implique que $\underline{e}$ ne peut prendre qu'une quantité finie de valeurs, disons k ; comme la succession : $\underline{e}, \underline{e}^2, \ldots \underline{e}^k, \underline{e}^{k+1}, \ldots$ est une suite croissante de fonctions (valeurs), puisque $\underline{e}$ est progressive ; on a :

$\underline{e} \sqsubseteq \underline{e}^2 \sqsubseteq \ldots \sqsubseteq \underline{e}^k \sqsubseteq \underline{e}^{k+1} \sqsubseteq \ldots$;

par conséquent : $\underline{e} \sqsubseteq \underline{e}^2 \sqsubseteq \ldots \sqsubseteq \underline{e}^k = \underline{e}^{k+1} = \ldots$ cqfd.

4.4.4.2. Rétracts à caractère fini dans FEC($\lambda g.g\circ g$)

Le théorème suivant établit que les rétracts engendrés continuement sont des rétracts à caractère fini.

Théorème :

$(\forall f)$ [$f \in |D \to D| \wedge D$ domaine

$\Rightarrow$ $Cgn(\lambda g.g\circ g)(f)$ est un rétract à caractère fini]

Démonstration :

Par le théorème § 4.4.2 partie (6), il suffit de montrer l'égalité suivante entre ensembles :

$(\forall f)(\forall x)$ [$f \in |D \to D| \wedge x \in D$ $\Rightarrow$

Image $(\lambda x.(Cgn(\lambda g.g\circ g)(f))(x))$ = Image $(\lambda x.Cgn(Cgn(\lambda g.g\circ g)(f))(x))$]

Pour abréger, posons :

$f' = Cgn(\lambda g.g\circ g)(f)$

nous savons que : $f'\circ f' = f'$ (théorème § 4.4.2. partie (3))

($\supseteq$) il est clair que :

Image $(f') \supseteq$ Image $(\lambda x.Cgn(f')(x))$

puisque : Image $(\lambda x.Cgn(f')(x))$ = FEC$(f') \subseteq$ FIX(f') = Image (f')

($\subseteq$) pour voir que :

Image $(f') \subseteq$ Image$(\lambda x.Cgn(f')(x))$

il suffit de montrer :

$(\forall x)$ [$x \in D \Rightarrow Cgn(f')(f'(x)) = f'(x)$]

Soient E et $\underline{E}$ les ensembles des éléments finis dans les domaines D et $|D \to D|$ respectivement, alors :

$f' = Cgn(\lambda g.g\circ g)(f)$

$= \sqcup(\bigcup_{n=o}^{\infty}\{ \underline{e}^{2^n} \mid \underline{e} \in \underline{E} \wedge \underline{e} \sqsubseteq f \wedge \underline{e} \sqsubseteq \underline{e}^2 \})$

$= \sqcup(\bigcup_{n=o}^{\infty}\{ \underline{e}^{n} \mid \underline{e} \in \underline{E} \wedge \underline{e} \sqsubseteq f \wedge \underline{e} \sqsubseteq \underline{e}^2 \})$ (lemme § 4.4.4.1)

$$= \sqcup(\bigcup_{n=o}^{\infty} \{\underline{e}^n \mid \underline{e} \in \underline{E} \wedge \underline{e} \sqsubseteq f' \wedge \underline{e} \sqsubseteq \underline{e}^2\}) \quad (\underline{e}^n \sqsubseteq f' \wedge \underline{e} \sqsubseteq \underline{e}^2)$$

comme $\bigcup_{n=o}^{\infty} \{\underline{e}^n \mid \underline{e} \in \underline{E} \wedge \underline{e} \sqsubseteq f' \wedge \underline{e} \sqsubseteq \underline{e}^2\}$ est dirigé

et $f' = f' \circ f'$, on a :

$$f' = \sqcup(\bigcup_{n=o}^{\infty} \{f' \circ \underline{e}^n \mid \underline{e} \in \underline{E} \wedge \underline{e} \sqsubseteq f' \wedge \underline{e} \sqsubseteq \underline{e}^2\})$$

mais :

$$(\forall n \in \mathbb{N}) [\underline{e} \in \underline{E} \wedge \underline{e} \sqsubseteq f' \wedge \underline{e} \sqsubseteq \underline{e}^2 \Rightarrow \underline{e}^n \sqsubseteq \underline{e} \circ \underline{e}^n \sqsubseteq f' \circ \underline{e}^n]$$

de plus :

$$(\forall n \in \mathbb{N})(\forall x \in D) [\underline{e} \in \underline{E} \wedge \underline{e} \sqsubseteq f' = f' \circ f' \Rightarrow \underline{e}^n(x) \in \underline{E} \wedge \underline{e}^n(x) \sqsubseteq f'(x)]$$

par conséquent :

$$\begin{aligned} f'(x) &= \sqcup(\bigcup_{n=o}^{\infty} \{f'(\underline{e}^n(x)) \mid \underline{e} \in \underline{E} \wedge \underline{e} \sqsubseteq f' \wedge \underline{e} \sqsubseteq \underline{e}^2\}) \\ &= \sqcup(\bigcup_{n=o}^{\infty} \{f'(e) \mid e \in E \wedge e \sqsubseteq f'(x) \wedge e \sqsubseteq f'(e)\}) \\ &= Cgn(f')(f'(x)) \qquad \text{cqfd.} \end{aligned}$$

De cette façon, nous avons établi que le domaine des rétracts engendrés continuement est composé de rétracts à caractère fini. D. Scott a identifié ces rétracts par :

$$\rho(f) = \sqcup\{\underline{e}^{\infty} \mid \underline{e} \in \underline{E} \wedge \underline{e} \sqsubseteq f \wedge \underline{e} \sqsubseteq \underline{e}^2\}$$
$$(\text{où } \underline{e}^{\infty} = \underline{e}^k \text{ et } \underline{e} \sqsubseteq \underline{e}^2 \sqsubseteq \ldots \sqsubseteq \underline{e}^k = \underline{e}^{k+1} = \ldots)$$

c'est-à-dire :

$$\rho(f) = Cgn(\lambda g.g \circ g)(f)$$

4.4.4.3. Quelques éléments du domaine FEC ($\lambda g.g \circ g$)

Nous présentons maintenant quelques éléments particuliers (rétracts à caractère fini) du domaine des rétracts engendrés continuement. Ces rétracts sont les fonctions obtenues à partir de l'opérateur de points fixes engendrés continuement (Cgn).

Théorème :

$(\forall f)(\forall x)[$ D domaine $\wedge\, x \in D \wedge f \in [D \to D]$

$\Rightarrow\ Cgn(\lambda g.g\circ g)(\lambda x.Cgn(f)(x)) = \lambda x.Cgn(f)(x)]$

Démonstration :

Soit $f'' = \lambda x.Cgn(f)(x)$ et soient E et $\underline{E}$ les ensembles des éléments finis des domaines D et $[D\to D]$ respectivement ; nous savons que :

$$Cgn(\lambda g.g\circ g)(f'') = \sqcup(\bigcup_{n=o}^{\infty}\{\underline{e}^n \mid \underline{e}\in E \wedge \underline{e} \sqsubseteq f'' \wedge \underline{e} \sqsubseteq \underline{e}^2\})$$

(lemme § 4.4.4.1)

et $f'' = (f'')^n = \sqcup\{\underline{e}^n \mid \underline{e}\in \underline{E} \wedge \underline{e} \sqsubseteq f''\}$

par conséquent :

$Cgn(\lambda g.g\circ g)(f'') \sqsubseteq f''$ (lemme § 4.4.4.1)

maintenant nous devons montrer :

$f'' \sqsubseteq Cgn(\lambda g.g\circ g)(f'')$

pour celà, rappelons-nous que :

$f'' = \sqcup S_{f''}$ (lemme § 3.3.1.4)

(où $S_{f''} = \{<e_1;e_2> \mid e_1 \in E \wedge e_2 \in E \wedge <e_1;e_2> \in \underline{E} \wedge <e_1;e_2> \sqsubseteq f''\}$)

et $<e_1;e_2> \sqsubseteq f'' \Leftrightarrow e_2 \sqsubseteq f''(e_1)$ (lemme 2 § 3.3.1.3)

Comme f" est un rétract à caractère fini, alors :

$f''(x) \in FEC(f'')$ (théorème § 4.4.2 (6))

par conséquent :

$f''(x) = \sqcup\{e \mid e \in E \wedge e \sqsubseteq f''(x) \wedge e \sqsubseteq f''(e)\}$ (lemme § 4.4.1)

en particulier, pour chaque $<e_1;e_2> \in S_{f''}$, on a :

$e_2 \sqsubseteq f''(e_1) = \sqcup\{e \mid e \in E \wedge e \sqsubseteq f''(e_1) \wedge e \sqsubseteq f''(e)\}$

Comme $e_2 \in E$ (c,a.d. est fini) et $\{e \mid e \in E \wedge e \sqsubseteq f''(e_2) \wedge e \sqsubseteq f''(e)\}$

est dirigé, alors :

$$(\exists e')\ [e' \in E \wedge e' \sqsubseteq f''(e_1) \wedge e' \sqsubseteq f''(e') \wedge e_2 \sqsubseteq e']$$

de plus :

$$<e_1;e_2> \sqsubseteq <e_1;e'> \sqsubseteq f'' \wedge <e';e'> \sqsubseteq f'' \qquad \text{(lemme 2 § 3.3.1.3)}$$

Soit $\underline{e}' = <e_1;e'> \cup <e';e'>$

on a :

$$\underline{e}' \in E \wedge <e_1;e_2> \sqsubseteq \underline{e}' \wedge \underline{e}' \sqsubseteq \underline{e}' \circ \underline{e}' \wedge \underline{e}' \sqsubseteq f''$$

Soit F" l'ensemble des $\underline{e}'$ finis, qui sont construits pour chaque $<e_1;e_2> \in S_{f''}$,alors :

$$f'' = \bigsqcup S_{f''} \wedge \bigsqcup S_{f''} \sqsubseteq \bigsqcup F'' \qquad \text{(théorème 2 § 1.1.5)}$$

et $\bigsqcup F'' \sqsubseteq Cgn(\lambda g.g\circ g)(f'')$ (théorème 3 § 1.1.5)

par conséquent

$$f'' \sqsubseteq Cgn(\lambda g.g\circ g)(f'') \qquad \text{cqfd.}$$

4.5. <u>Caractérisation au moyen de rétracts à caractère fini des opérations ×, +, →</u> .

Dans les lemmes § 3.1.2 , § 3.2.4, § 3.3.2, nous avons vu que les opérations ×, +, → respectivement, construisent de nouveaux domaines. Maintenant, nous allons voir que si ces opérations se réalisent entre domaines caractérisés par des rétracts à caractère fini (§ 2.3.4), alors le domaine résultat de l'opération est caractérisé par un rétract à caractère fini (r.c.f.). Les trois théorèmes suivants établissent la caractérisation des opérations données (constructeurs) au moyen des opérateurs r.c.f.

4.5.1. Opérateur - r.c.f. - du produit cartésien ($a \boxtimes b$)

Théorème :

Soient
- . $D_1(\sqsubseteq, \perp, \top, \sqcup, \sqcap)$ et D_2 $(\sqsubseteq', \perp', \top', \sqcup', \sqcap')$ domaines,
- . $a \in [D_1 \to D_1]$ et $b \in [D_2 \to D_2]$ rétracts à caractère fini,
- . $D_a = a(D_1) \subseteq D_1$ et $D_b = b(D_2) \subseteq D_2$ les domaines caractérisés par a et b respectivement

et soit $a \boxtimes b : D_1 \times D_2 \to D_1 \times D_2$

défini par : $a \boxtimes b = \lambda x. < a(x \downarrow 1), b(x \downarrow 2) >$

alors :

(1) $a \boxtimes b$ est continue

(2) $a \boxtimes b$ est un rétract (c.a.d. idempotent)

(3) Image $(a \boxtimes b)$ = FIX$(a \boxtimes b) = D_a \times D_b$

(4) $a \boxtimes b$ est un rétract à caractère fini (r.c.f.)

(5) a fermeture et b fermeture $\Rightarrow$ $a \boxtimes b$ fermeture.

Démonstration :

(1) immédiat, puisque $\downarrow 1$, $\downarrow 2$ a et b sont des fonctions continues ;

(2)
$$(a \boxtimes b)^2 = (\lambda x. < a(x\downarrow 1), b(x\downarrow 2) >)\ (\lambda x. < a(x\downarrow 1), b(x\downarrow 2) >)$$
$$= < a((\lambda x. < a(x\downarrow 1), b(x\downarrow 2)>) \downarrow 1),$$
$$b((\lambda x. < a(x\downarrow 1), b(x\downarrow 2) >)\downarrow 2)>$$

$= \lambda x. < a((< a(x\downarrow 1), b(x\downarrow 2) >)\downarrow 1),$ $b((<a(x\downarrow 1), b(x\downarrow 2)>) \downarrow 2)>$ (puisque x est la même dans les deux composantes)

$= \lambda x. < a(a(x\downarrow 1)), b(b(x \downarrow 2))>$ application $\downarrow 1$ et $\downarrow 2$)

$= \lambda x. < a(x \downarrow 1), b(x \downarrow 2)>$ (a et b rétracts)

$= a \boxtimes b$

(3) $(a \boxtimes b)(D_1 \times D_2) = <a((D_1 \times D_2)\downarrow 1), b(D_1 \times D_2)\downarrow 2)>$

$= <a(D_1), b(D_2)>$

$= \{ <x,y> \mid x \in a(D_1) \wedge y \in b(D_2)\}$

$= D_a \times D_b$

par le lemme § 3.1.2, $D_a \times D_b$ est un domaine; et de (2), on a le résultat.

(4) immédiat : $(2) \wedge (3) \Rightarrow (4)$

(5) $(a \boxtimes b) <x,y> = < a(x), b(y) >$

comme a et b sont des fermetures, on a :

$x \sqsubseteq a(x)$ et $y \sqsubseteq' b(y) \Rightarrow <x,y> \sqsubseteq'' <a(x), b(y)>$

($\sqsubseteq''$ relation d'ordre dans $D_1 \times D_2$ - § 3.1.1.1)

par conséquent : $<x,y> \sqsubseteq'' (a \boxtimes b) <x,y>$ $(\forall <x,y> \in D_1 \times D_2)$ cqfd.

4.5.2. Opérateur - r.c.f. - de l'union disjointe ou somme $(a \boxplus b)$

Théorème :

Soient : D_1, D_2, a, b, D_a et D_b comme définis dans l'hypothèse du théorème précédent (§ 4.5.1) et soit : $a \boxplus b : D_1 + D_2 \to D_1 + D_2$

définie par : $a \boxplus b = \lambda x.\ \underline{sib}\ (x \,\mathrm{E}\, D_1, (a(x|D_1))_1 \text{ in } D_1 + D_2,$
$(b(x \mid D_2))_2 \text{ in } D_1 + D_2)$

alors : (1) $a \boxplus b$ est continue

(2) $a \boxplus b$ est un rétract

(3) Image $(a \boxplus b)$ = FIX $(a \boxplus b) = D_a + D_b$

(4) $a \boxplus b$ est un rétract à caractère fini

(5) a fermeture $\wedge$ b fermeture => $a \boxplus b$ fermeture

<u>Démonstration</u> :

(1) Immédiat, puisque : <u>sib</u>, E, $|D_1$, $|D_2$, in sont des fonctions continues.

(2) $$(a \boxplus b)^2 = (\lambda x.\underline{sib}\ (x\mathrm{E}D_1, (a(x \mid D_1))_1 \text{ in } D_1 + D_2, (b(x \mid D_2))_2 \text{ in } D_1 + D_2))(a \boxplus b)$$
$$= \underline{sib}((a \boxplus b)\,\mathrm{E}\,D_1,\ (a((a \boxplus b) \mid D_1))_1 \text{ in } D_1 + D_2, (b((a \boxplus b) \mid D_2))_2 \text{ in } D_1 + D_2)$$

Comme la variable liée x en $a \boxplus b$ est la même dans les trois composants de <u>sib</u>, on a :

$$(a \boxplus b)^2 = \lambda x.\underline{sib}\ ((a \boxplus b)(x)\,\mathrm{E}\,D_1, (a((a \boxplus b)(x) \mid D_1))_1 \text{ in } D_1 + D_2, (b((a \boxplus b)(x) \mid D_2))_2 \text{ in } D_1 + D_2)$$

Dans le premier composant de <u>sib</u> (c.a.d. $(a \boxplus b)(x)\,\mathrm{E}\,D_1$), on a :

$$(a \boxplus b)(x)\,\mathrm{E}\,D_1 = \underline{sib}\ (x\mathrm{E}D_1, (a(x \mid D_1))_1 \text{ in } D_1 + D_2, (b(x \mid D_2))_2 \text{ in } D_1 + D_2)\,\mathrm{E}\,D_1$$

par la définition de in (§ 3.2.2.1), E et <u>sib</u> (§ 3.2.2.3), il est facile de voir que :

$$(a \boxplus b)(x)\,\mathrm{E}\,D_1 = x\,\mathrm{E}\,D_1$$

par conséquent, en substituant et en développant $(a \boxplus b)(x)$:

$$\begin{aligned}(a \boxplus b)^2 = \lambda x.\underline{sib}\ (x\,\mathrm{E}\,D_1,\ & (a((\underline{sib}(x\,\mathrm{E}\,D_1, (a(x \mid D_1))_1 \text{ in } D_1 + D_2, (b(x \mid D_2))_2 \text{ in } D_1 + D_2)) \mid D_1))_1 \text{ in } D_1 + D_2,\\ & (b((\underline{sib}\ (x\,\mathrm{E}\,D_1, (a(x \mid D_1))_1 \text{ in } D_1 + D_2, (b(x \mid D_2))_2 \text{ in } D_1 + D_2)) \mid D_2))_2 \text{ in } D_1 + D_2)\end{aligned}$$

La fonction $(a \boxplus b)^2$ peut prendre quatre valeurs selon la définition de E et $\underline{\text{sib}}$:

(2a) $\quad x \, E \, D_1 = \bot_{Bool} \Rightarrow (a \boxplus b)^2(x) = \bot_{D_1 + D_2}$

(2b) $\quad x \, E \, D_1 = \top_{Bool} \Rightarrow (a \boxplus b)^2(x) = \top_{D_1 + D_2}$

(2c) $\quad x \, E \, D_1 = \text{vrai} \Rightarrow$

$$
\begin{aligned}
(a \boxplus b)^2(x) &= (a(((a(x \mid D_1))_1 \text{ in } D_1 + D_2) \mid D_1))_1 \text{ in } D_1 + D_2 \\
&= (a(a(x)))_1 \text{ in } D_1 + D_2 \qquad (\S\ 3.2.2.2) \\
&= (a(x))_1 \text{ in } D_1 + D_2 \qquad (a^2 = a) \\
&= (a(x \mid D_1))_1 \text{ in } D_1 + D_2
\end{aligned}
$$

(2d) $\quad x \, E \, D_1 = \text{faux} \Rightarrow$

$$
\begin{aligned}
(a \boxplus b)^2(x) &= (b(((b(x \mid D_2))_2 \text{ in } D_1 + D_2) \mid D_2))_2 \text{ in } D_1 + D_2 \\
&= (b(x \mid D_2))_2 \text{ in } D_1 + D_2 \qquad (\text{comme en 2c})
\end{aligned}
$$

Par conséquent :

$$
\begin{aligned}
(a \boxplus b)^2 &= \lambda x.\underline{\text{sib}}\ (x E\ D_1, (a(x \mid D_1))_1 \text{ in } D_1 + D_2, \\
&\qquad (b(x \mid D_2))_2 \text{ in } D_1 + D_2) \\
&= a \boxplus b
\end{aligned}
$$

(3)

$$
\begin{aligned}
(a \boxplus b)(D_1 + D_2) &= \underline{\text{sib}}((D_1 + D_2)\, E\, D_1, (a((D_1 + D_2) \mid D_1))_1 \text{in } D_1 + D_2, \\
&\qquad (b((D_1 + D_2) \mid D_2))_2 \text{ in } D_1 + D_2) \\
&= \underline{\text{sib}}((D_1 + D_2)\, E\, D_1,\ (a(D_1))_1 \text{ in } D_1 + D_2, \\
&\qquad (b(D_2))_2 \text{ in } D_1 + D_2) \\
&= \underline{\text{sib}}\ ((D_1 + D_2)\, E\, D_1,\ (D_a)_1 \text{ in } D_1 + D_2, \\
&\qquad (D_b)_2 \text{ in } D_1 + D_2)
\end{aligned}
$$

si nous identifions $\perp_{D_1 + D_2}$ avec $\perp_{D_a + D_b}$ et $\top_{D_1 + D_2}$ avec $\top_{D_a + D_b}$, il est facile de voir que :

$$(a \boxplus b)\ (D_1 + D_2) = D_a + D_b$$

Par le lemme § 3.2.4, $D_a + D_b$ est un domaine ; et de (2) on a le résultat.

(4) immédiat, (2) $\wedge$ (3) $\Rightarrow$ (4)

(5) on a :

$x \in D_1 \wedge x \sqsubseteq a(x) \Rightarrow x \sqsubseteq_{D_1 + D_2} (a \boxplus b)(x)$ (§ 3.2.3.1)

et $x \in D_2 \wedge x \sqsubseteq b(x) \Rightarrow x \sqsubseteq_{D_1 + D_2} (a \boxplus b)(x)$ (idem)

Par conséquent :

a fermeture $\wedge$ b fermeture $\Rightarrow x \sqsubseteq_{D_1 + D_2} (a \boxplus b)(x)$ cqfd.

4.5.3. Opérateur - r.c.f. - de l'espace des fonctions continues $(a \circ\!\!\to b)$

Théorème :

Soient D_1, D_2, a, b, D_a et D_b comme définis dans l'hypothèse du théorème § 4.5.1

et soit : $a \circ\!\!\to b : [D_1 \to D_2] \to [D_1 \to D_2]$

définie par $a \circ\!\!\to b = \lambda f.b \circ f \circ a$

alors : (1) $a \circ\!\!\to b$ est continue

(2) $a \circ\!\!\to b$ est un rétract

(3) Image $(a \circ\!\!\to b)$ = FIX$(a \circ\!\!\to b) = [D_a \to D_b]$

(4) $a \circ\!\!\to b$ est un rétract à caractère fini

(5) a fermeture $\wedge$ b fermeture $\Rightarrow a \circ\!\!\to b$ fermeture

Démonstration :

(1) immédiat (§ 4.1.2)

$$\begin{aligned}(2)\ (a \circ\!\!\to b)^2 &= (\lambda f.b\circ f\circ a)(\lambda f.b\circ f\circ a)\\ &= b\circ(\lambda f.b\circ f\circ a)\circ a\\ &= \lambda f.b\circ(b\circ f\circ a)\circ a\\ &= \lambda f.(b\circ b)\circ f\circ(a\circ a)\\ &= \lambda f.b\circ f\circ a \qquad (a^2 = a \wedge b^2 = b)\\ &= a \circ\!\!\to b\end{aligned}$$

(3) $(a \circ\!\!\to b)([D_1 \to D_2]) = b \circ [D_1 \to D_2] \circ a$

Comme a "réduit" D_1 en a (D_1) et b réduit D_2 en $b(D_2)$, il est facile de voir que :

$$(a \circ\!\!\to b)([D_1 \to D_2]) = [a(D_1) \to b(D_2)] = [D_a \to D_b]$$

Par le lemme § 3.3.2., $[D_a \to D_b]$ est un domaine ; et de (2), on a le résultat.

(4) immédiat, (2) $\wedge$(3)=>(4)

(5) $x \sqsubseteq a(x) \Rightarrow (b \circ f)(x) \sqsubseteq' (b \circ f \circ a)(x)$ (b et f monotones et continues)

$y \sqsubseteq' b(y) \Rightarrow f(x) \sqsubseteq' (b \circ f)(x)$

$f(x) \sqsubseteq' (b \circ f)(x) \sqsubseteq' (b \circ f \circ a)(x) \Rightarrow f(x) \sqsubseteq' (b \circ f \circ a)(x)$

Comme $f(x) \sqsubseteq' (b \circ f \circ a)(x)$ pour tout $x \in D_1$ et pour tout $f \in [D_i \to D_2]$, on a par le paragraphe § 3.3.1.1 :

$$f \sqsubseteq'' (a \circ\!\!\to b)(f)$$

($\sqsubseteq''$ est la relation d'ordre partiel en $[D_1 \to D_2]$) cqfd.

4.5.4. Continuité des opérateurs $\boxtimes$, $\boxplus$ et $\circ\!\!\to$

Dans les paragraphes précédents, nous avons vu que les opérateurs a $\boxtimes$ b, a $\boxplus$ b, et a $\circ\!\!\to$b sont des opérateurs continus, où a et b représentent des rétracts à caractère fini. Nous allons voir, maintenant, ce qui arrive quand a et b sont des variables qui appartiennent aux domaines des rétracts engendrés continuement (§ 4.4.4) :

$$\text{FEC } (\lambda g.g^2)_{D_1} = \text{Cgn}(\lambda g.g^2)([D_1 \to D_1])$$

et $\text{FEC } (\lambda g.g^2)_{D_2} = \text{Cgn}(\lambda g.g^2)([D_2 \to D_2])$ respectivement.

<u>Théorème</u> :

Soient les domaines : $D_1(\sqsubseteq_1, \perp_1, \top_1, \sqcup_1, \sqcap_1)$, $D_2(\sqsubseteq_2, \perp_2, \top_2, \sqcup_2, \sqcap_2)$

. $\text{FEC}(\lambda g.g^2)_{D_1} (\sqsubseteq_3, \perp_3, \top_3, \sqcup_3, \sqcap_3)$

. $\text{FEC}(\lambda g.g^2)_{D_2} (\sqsubseteq_4, \perp_4, \top_4, \sqcup_4, \sqcap_4)$

. $[(D_1 \times D_2) \to (D_1 \times D_2)]$ $(\sqsubseteq_5, \perp_5, \top_5, \sqcup_5, \sqcap_5)$

. $[(D_1 + D_2) \to (D_1 + D_2)]$ $(\sqsubseteq_6, \perp_6, \top_6, \sqcup_6, \sqcap_6)$

. $[[D_1 \to D_2] \to [D_1 \to D_2]]$ $(\sqsubseteq_7, \perp_7, \top_7, \sqcup_7, \sqcap_7)$

et soient les opérateurs :

$$. \boxtimes : FEC(\lambda g.g^2)_{D_1} \times FEC(\lambda g.g^2)_{D_2} \to [(D_1 \times D_2) \mapsto (D_1 \times D_2)]$$

$$. \boxplus : FEC(\lambda g.g^2)_{D_1} \times FEC(\lambda g.g^2)_{D_2} \to [(D_1 + D_2) \to (D_1 + D_2)]$$

$$. \circ\!\!\to : FEC(\lambda g.g^2)_{D_1} \times FEC(\lambda g.g^2)_{D_2} \to [[D_1 \to D_2] \to [D_1 \to D_2]]$$

définis par :

$. \boxtimes (y,z) = y \boxtimes z$ (théorème § 4.5.1)

$. \boxplus (y,z) = y \boxplus z$ (théorème § 4.5.2)

$. \circ\!\!\to (y,z) = y \circ\!\!\to z$ (théorème § 4.5.3)

alors : (1) $\boxtimes$ est continue

(2) $\boxplus$ est continue

(3) $\circ\!\!\to$ est continue

<u>Démonstration</u> :

Nous présentons uniquement la démonstration pour (1) ; dans les cas (2) et (3), elle est similaire. Pour (2), il suffit de voir que l'injection (§ 3.2.2.1) et <u>sib</u> (§ 3.2.2.3) sont continues ; et pour (3), que la composition est continue (théorème § 4.1.1).

(1) $\boxtimes$ est continue, c'est-à-dire :

Soient $Y \subseteq FEC(\lambda g.g^2)_{D_1} \wedge Z \subseteq FEC(\lambda g.g^2)_{D_2} \wedge Y,Z$ dirigés, il suffit de montrer (théorème § 3.1.3) :

(1a) $\boxtimes (\sqcup_3 Y, z) = \sqcup_5 (\boxtimes(Y,z))$

(1b) $\boxtimes (y, \sqcup_4 Z) = \sqcup_5 (\boxtimes(y,Z))$

1a) (la démonstration est la même pour 1b)

$\boxtimes\ (\sqcup_3 Y, z) = \sqcup_3 Y \boxtimes z = \lambda x.<(\sqcup_3 Y)(x \downarrow 1), z(x \downarrow 2)>$ (théorème § 4.5.1)

$= \lambda x.<\sqcup_1 (Y(x \downarrow 1)), z(x \downarrow 2)>$ (définition borne § 3.3.1.2)

$= \lambda x.\sqcup <Y(x \downarrow 1), z(x \downarrow 2)>$ ($\sqcup$ est la borne en $D_1 \times D_2$, déf. § 3.1.1.2)

$= \sqcup_5 (\boxtimes(Y, z))$ (définition borne § 3.3.1.2)

cqfd.

CHAPITRE 5.
DOMAINES REFLEXIFS ET EQUATIONS DE DOMAINES

5 - DOMAINES REFLEXIFS et EQUATIONS de DOMAINES

Dans les chapitres précédents nous avons montré que les opérateurs qui caractérisent le produit cartésien, la somme, et l'espace des fonctions continues sont des rétracts à caractère fini. Nous avons vu aussi comment calculer les fermetures (§4.3.2.) et les rétracts à caractère fini (§4.4.2.), et nous avons présentés des domaines dont les éléments sont des rétracts à caractère fini (le domaine C - §4.3. -, et le domaine FEC ($\lambda g.\ g \circ g$) - §4.4.4.). Maintenant, nous allons voir comment utiliser ces opérateurs dans certains domaines spéciaux : les domaines universels réflexifs, dans le but de trouver une solution aux équations de domaine. Ces équations sont de la forme :

$Y = F\ (X)$

où F est une expression (fonction) formée des opérations de produit cartésien ($\times$), somme (+), et exponentiation ($\rightarrow$) entre domaines, X étant le domaine variable et Y le domaine résultat. Par exemple :

$F(X) = A + [B \rightarrow X]$

dans ce cas, $F(X)$ est le domaine égal à la somme du domaine A et de l'espace (domaine) des fonctions continues du domaine B dans le domaine X (variable liée). Comme nous l'avons dit dans l'introduction, la méthode dénotationnelle conduit à des définitions récursives (circulaires) de domaines, c'est-à-dire $X = F\ (X)$. En général, nous appellerons ce type de domaines (i.e. : $X = F(X)$) des <u>domaines réflexifs</u>, et le problème est de trouver des points fixes (domaines) de la fonction F ; c'est-à-dire, des domaines qui possèdent comme sous-espace leur propre domaine et/ou des domaines construits avec les opérations $\times$, +, $\rightarrow$, où leur propre domaine est un opérande, par exemple : $X = X \times [X \rightarrow X]$. En réalité, nous verrons que ces domaines (X dans l'exemple) possèdent comme sous-domaines, des domaines <u>isomorphes</u> au sous-espace demandé ($[X \rightarrow X]$ dans l'exemple).

5.1. Connexions de Galois ([Sal 77])([Or 0 44])

Pour continuer, nous présentons la caractérisation de domaines via des isomorphismes (§2.1.1.2.). Pour celà, nous utilisons la notion de <u>connexion de Galois</u>, qui est définie de la manière suivante :

Soient : . $D(\sqsubseteq,\perp,\top,\sqcup,\sqcap)$ et $D'(\sqsubseteq',\perp',\top',\sqcup',\sqcap')$ des domaines,
. $f : D \to D'$, une fonction monotone (ou continue, i.e. $f \in [D \to D']$),
. $g : D' \to D$, une fonction monotone (ou continue, i.e. $g \in [D' \to D]$),
alors, la paire (f,g) forme une connexion de Galois entre les domaines D et D' ssi : (1) $f \circ g \circ f = f$ et (2) $g \circ f \circ g = g$.
Nous dénoterons une connexion de Galois par le quadruple ordonné $\langle f,D,D',g \rangle$, c'est-à-dire : $\langle f,D,D',g \rangle \Leftrightarrow f \circ g \circ f = f \wedge g \circ f \circ g = g$
et nous dirons que la connexion $\langle f,D,D'g \rangle$ est continue si f et g sont continues.
Comme conséquence immédiate de la notation prise pour représenter les connexions, on a le lemme suivant :

Lemme 1 : $\langle f,D,D',g \rangle \Leftrightarrow \langle g,D',D,f \rangle$

Un point important dans les connexions est donné par le lemme suivant :

Lemme 2 : $\langle f,D,D',g \rangle \Leftrightarrow f \circ g$ et $g \circ f$ idempotentes.

Démonstration :

$(f \circ g)^2 = f \circ g \circ f \circ g = f \circ (g \circ f \circ g) = f \circ g$ cqfd.

Cela signifie que la composition des fonctions qui forment une connexion, caractérisent des sous-espaces (à travers des rétracts) dans les espaces qui interviennent dans la connexion.

Le théorème suivant établit l'isomorphisme entre les sous-espaces (treillis complets) images de la connexion.

Théorème :

Soit la connexion $\langle f,D,D',g \rangle$,
et soient $f_{g(D')}$ et $g_{f(D)}$ les fonctions f et g restreintes aux espaces $g(D') \subseteq D$ et $f(D) \subseteq D'$ respectivement, c'est-à-dire :
$f_{g(D')} = f : \text{Image }(g) \to \text{Image }(f)$
et $g_{f(D)} = g : \text{Image }(f) \to \text{Image }(g)$
alors :

(1) $f_{g(D')}$ et $g_{f(D)}$ sont surjectives,

(2) Image (f) et Image (g) sont des treillis complets,

(3) $f_{g(D')}$ et $g_{f(D)}$ sont strictement monotones,

(4) $g_{f(D)}$ est l'isomorphisme inverse de $f_{g(D')}$.

<u>Démonstration</u> :

(1) $f_{g(D')}$ surjective, c'est-à-dire : Image $(f_{g(D')})$ = Image (f) ; il est clair que : Image $(f_{g(D')}) \subseteq$ Image (f) ; montrons l'inverse (c.a.d.: Image (f) $\subseteq$ Image $(f_{g(D')})$).

Image (f) = $f(D) = f \circ g \circ f\ (D)$ $\quad$ ($\langle f,D,D',g\rangle$ connexion)

$\qquad = f \circ g\ (f(D)) \subseteq f \circ g\ (D')$

Image (f) $\subseteq f(g(D'))$ = f (Image (g)) = Image $(f_{g(D')})$.

(La démonstration est semblable pour $g_{f(D)}$).

(2) Il est clair que $g \circ f$ est monotone (continue) ; par le lemme 2 précédent (c.a.d. : $(f \circ g)^2 = f \circ g$) et le théorème de Tarski (§2.3.1.), nous avons que Image $(f \circ g)$ est un treillis complet, de plus, par la partie (1) de ce théorème, nous savons que :

Image $(f \circ g) = f(g(D'))$ = f(Image (g)) = Image $(f_{g(D')})$ = Image (f).

(La démonstration pour Image (g) est semblable).

(3) $f_{g(D')}$ est strictement monotone, c'est-à-dire :

$(\forall x,y \in \text{Image } (g))\ [x \sqsubseteq y \Leftrightarrow f_{g(D')}(x) \sqsubseteq' f_{g(D')}(y)]$

($\Rightarrow$) immédiat, puisque $f_{g(D')}$ est monotone(continue)

($\Leftarrow$) $f_{g(D')}(x) \sqsubseteq' f_{g(D')}(y)$

$\qquad \Rightarrow (g_{f(D)} \circ f_{g(D')})(x) \sqsubseteq (g_{f(D)} \circ f_{g(D')})(y)$ $\quad$ ($g_{f(D)}$ monotone)

$\qquad \Rightarrow (g_{f(D)} \circ f_{g(D')})(g(x')) \sqsubseteq (g_{f(D)} \circ f_{g(D')})(g(y'))$ $\quad$ ($x = g(x')$ et $y = g(y')$)

$\qquad \Rightarrow (g \circ f \circ g)(x') \sqsubseteq (g \circ f \circ g)(y')$

$\qquad \Rightarrow g(x') \sqsubseteq g(y')$

$\qquad \Rightarrow x \sqsubseteq y$

(La démonstration est semblable pour $g_{f(D)}$).

(4) Notons que, en (3), nous avons montré aussi que :

$(\forall x \in \text{Image }(g))\ [(g_{f(D)} \circ f_{g(D')})\ (x) = x]$

c'est-à-dire : $g_{f(D)} = f^{-1}_{g(D')}$;

Par (1) et (3) nous avons que $f_{g(D')}$ est un isomorphisme.

Par conséquent, Image (f) est isomorphe à Image (g). cqfd.

Avec ce théorème, nous avons vu que les sous-espaces -treillis complets - caractérisés par la connexion (i.e. : Image (f) et Image (g)) sont isomorphes.

Par conséquent, si $f \circ g = I_{D'}$ alors $g \circ f$ est un rétract à caractère fini puisque : Image ($g \circ f$) = Image (g) (th. § 5.1(2)) est isomorphe au domaine D' (lemme 7 §2.1.2.) (i.e. : D' = Image ($I_{D'}$) = Image ($f \circ g$) = Image (f)). De cette façon, nous avons que $g \circ f$ caractérise un sous domaine du domaine D, même si $g \circ f$ n'est pas continue ; cependant si g est continue alors elle est même additive (lemme 5 § 2.1.2.) puisque : $f \circ g = I_{D'}$ implique $g = g_{f(D)}$.

Par conséquent : si f est continue, alors $g \circ f$ est continue .

Avec ces considérations, nous donnons la définition suivante :

5.1.1. Les connexions comme rétracts à caractère fini

La connexion <f,D,D',g>est une connexion à caractère fini (donnée par le rétract à caractère fini $g \circ f$) ssi

(1) f est continue

et (2) $f \circ g = I_{D'}$

Le domaine caractérisé par la connexion est $D_{g \circ f} = (g \circ f)\ (D) \subseteq D$, et est isomorphe au domaine D'.

Le lemme suivant nous indique ce qui se produit avec les sous-domaines de D' caractérisés par des rétracts à caractère fini.

Lemme :

Soit <f,D,D',g >une connexion à caractère fini, et soit la fonction :

$H : [D' \to D'] \to [D \to D]$

définie par :

$H = \lambda h.\ g \circ h \circ f$

alors :

(1) H est bien définie et est continue,

(2) $(\forall h)$ $[h \in [D' \to D'] \wedge h^2 = h$ (i.e. h idempotente)

$\Rightarrow$ (2a) $\langle h\circ f, D, D', g\circ h\rangle$ est une connexion

(2b) et $H(h) = g\circ h\circ f$ est idempotente $]$,

(3) $(\forall h)$ $[h \in [D' \to D'] \wedge h$ rétract à caractère fini

$\wedge\ D'' = h(D') =$ Image (h) (i.e. : D'' domaine)

$\Rightarrow$ (3a) $\langle h\circ f, D, D'', g\circ h\rangle$ est une connexion à caractère fini

(3b) et Image (h) est isomorphe à Image $(g\circ h\circ f)]$.

Démonstration :

(1) immédiat, par le théorème §4.1.1.

(2a) $(h\circ f)\circ(g\circ h)\circ(h\circ f) = h\circ(f\circ g)\circ(h\circ h)\circ f$

$= h\circ I_{D'}\circ h\circ f \qquad (f\circ g = I_{D'} \wedge h^2 = h)$

$= (h\circ h)\circ f = h\circ f$

$(g\circ h)\circ(h\circ f)\circ(g\circ h) = g\circ(h\circ h)\circ(f\circ g)\circ h = g\circ h\circ h = g\circ h$

(2b) $(g\circ h\circ f)^2 = g\circ h\circ f\circ g\circ h\circ f = g\circ h\circ(f\circ g)\circ h\circ f = g\circ(h\circ h)\circ f = g\circ h\circ f$

(3a) Il est facile de voir que : $h\circ f \in [D \to D''] \wedge g\circ h \in [D'' \to D]$,
alors par (2a), on a que $\langle h\circ f, D, D'', g\circ h\rangle$ est aussi une connexion. Il nous manque à montrer que : $(h\circ f)\circ(g\circ h) = I_{D''}$, c'est-à-dire :

$(h\circ f)\circ(g\circ h) = h\circ(f\circ g)\circ h$

$= h\circ I_{D'}\circ h$ ($\langle f, D, D', g\rangle$ est à caractère fini)

$= h\circ h = h$

mais,

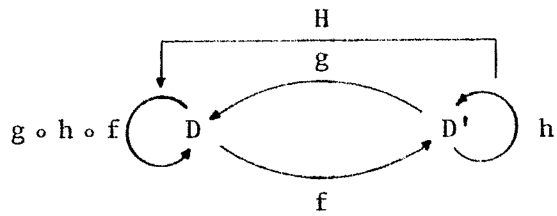
$(\forall x)$ $[x \in D'' = h(D') \wedge h^2 = h \Rightarrow h(x) = x]$

donc : $h = I_{D''}$

(3b) par (3a) nous savons que :
Image $((g\circ f)\circ(h\circ f))$ est isomorphe à $D'' = h(D') =$ Image (h)
mais Image $((g\circ h)\circ(h\circ f)) =$ Image $(g\circ(h\circ h)\circ f) =$ Image $(g\circ h\circ f)$;
ainsi donc, $g\circ h\circ f = H(h)$ est un rétract à caractère fini, puisque son image est isomorphe au domaine caractérisé par le rétract à caractère fini h.

cqfd.

Graphiquement :



5.1.2. Représentations

Une classe de connexions de Galois sont les représentations, et sont définies par :

Soient : . D et D' des domaines,

. $f : D \to D' \wedge g : D' \to D$ des fonctions monotones,

alors :

$\ll f,D,D',g \gg$ représentation $\Leftrightarrow f \circ g \sqsubseteq I_{D'} \wedge I_D \sqsubseteq g \circ f$

Lemme : $\ll f,D,D',g \gg$ représentation $\Rightarrow \langle f,D,D',g \rangle$ connexion.

Démonstration :

$f \circ g \sqsubseteq I_{D'} \Rightarrow f \circ g \circ f \sqsubseteq f \wedge g \circ f \circ g \sqsubseteq g$ (1)

$I_D \sqsubseteq g \circ f \Rightarrow f \sqsubseteq f \circ g \circ f \wedge g \sqsubseteq g \circ f \circ g$ (2)

(1) $\wedge$ (2) $\Rightarrow f = f \circ g \circ f \wedge g = g \circ f \circ g$ cqfd .

De même que pour lès connexions, nous définissons les représentations à caractère fini, mais cette fois nous obtenons des fermetures.

5.1.3. Les représentations comme rétracts à caractère fini

Nous disons que $\ll f,D,D',g \gg$ est une représentation à caractère fini (donnée par la fermeture continue $g \circ f$) ssi :

(1) $\langle f,D,D',g \rangle$ est une connexion à caractère fini

et (2) $I_D \sqsubseteq g \circ f$

Le domaine caractérisé par la représentation est $D_{g \circ f} = (g \circ f)(D)$, lequel est isomorphe à D'.

Il est clair qu'une représentation à caractère fini est une représentation.

Notons que si la représentation est continue (i.e : f et g continues), la fermeture $g \circ f$ caractérise toujours un sous-domaine de D (théorème §2.3.3.). Cependant, nous nous intéressons au cas où $f \circ g = I_{D'}$, puisque de cette façon, le sous-domaine caractérisé par $g \circ f$ est isomorphe ("représente") le domaine D'.

En guise d'exemple, nous présentons maintenant un sous-domaine du domaine $D \times D$ (D domaine), qui est isomorphe à la somme séparée D + D.

5.1.4. D + D comme sous-domaine de D x D

Soit $D(\sqsubseteq, \perp, \top, \sqcup, \sqcap)$ domaine, et soit $<\Theta, D \times D, D_1 + D_2, \nu>$ une connexion, avec les domaines $D \times D(\sqsubseteq', \perp', \top', \sqcup', \sqcap')$ et $D_1 + D_2(\sqsubseteq, \perp'', \top'', \sqcup, \sqcap)$ les indices 1,2, indiquent uniquement l'opérande gauche et l'opérande droit respectivement dans la somme, i.e : $D_1 = D_2 = D$).

Définissons cette connexion de la façon suivante :
soient e_1 et e_2 deux éléments finis de D tels que e_1 et e_2 soient incomparables et leur intersection soit $\perp$, formellement :

$$e_1 \ll e_1 \in D \wedge e_2 \ll e_2 \in D \wedge e_1 || e_2 \wedge e_1 \sqcap e_2 = \perp \;;$$

On a :

$\Theta : D \times D \rightarrow D_1 + D_2$

$\nu : D_1 + D_2 \rightarrow D \times D$

définies par :

$$\Theta(x) = \begin{cases} \perp'' & \text{si } x{\downarrow}2 = \perp \\ (x{\downarrow}1)_1 \text{ in } D_1 + D_2 & \text{si } x{\downarrow}2 \sqsubseteq e_1 \wedge x{\downarrow}2 \neq \perp \\ (x{\downarrow}1)_2 \text{ in } D_1 + D_2 & \text{si } x{\downarrow}2 \sqsubseteq e_2 \wedge x{\downarrow}2 \neq \perp \\ \top'' & \text{sinon} \end{cases}$$

et $\nu(x) = \underline{\text{sib}}\ (y \in D_1, <y|D_1, e_1>, <y|D_2, e_2>)$.

Avec ces définitions de Θ et ν, on a que $<\Theta, D \times D, D_1 + D_2, \nu>$ est une connexion à caractère fini, et par conséquent : $(\nu \circ \Theta)(D \times D)$ est isomorphe à D + D.

5.2. Domaines Universels

On dit qu'un domaine U est un domaine universel ou un domaine universel reflexif s'il possède des sous-domaines qui sont isomorphes aux domaines $U \times U$, $U + U$, et $[U \to U]$. Les isomorphismes sont donnés par des connexions (représentations) à caractère fini que nous identifions à partir d'ici par :

(1) $< \alpha, U, U \times U, \beta >$

(2) $< \gamma, U, U + U, \delta >$

(3) $< \phi, U, [U \to U], \psi >$

Notons que si le domaine U a deux éléments finis comme ceux mentionnés dans le paragraphe précédent (§5.1.4.), alors la connexion $< \gamma, U, U + U, \delta>$ – 2 – s'obtient à partir de $< \alpha, U, U \times U, \beta >$ – (1) – et de $< \Theta, U \times U, U + U, \nu >$ – (§ 5.1.4.) – en faisant $\gamma = \Theta \circ \alpha$ et $\delta = \beta \circ \nu$, puisque :

(i) $\gamma = \Theta \circ \alpha$ est continue

et (ii) $\gamma \circ \delta = \Theta \circ \alpha \circ \beta \circ \nu = \Theta \circ (\alpha \circ \beta) \circ \nu = \Theta \circ I_{U \times U} \circ \nu = \Theta \circ \nu = I_{U+U}$

Par conséquent : $< \Theta \circ \alpha, U, U + U, \beta \circ \nu >$ est une connexion à caractère fini (§5.1.1.)

Graphiquement :

$$\begin{tikzcd} U \arrow[r, bend right, "\alpha"'] & U \times U \arrow[l, bend right, "\beta"'] \arrow[r, bend right, "\Theta"'] & U + U \arrow[l, bend right, dotted, "\nu"'] \end{tikzcd}$$

Comme nous l'avons fait dans le paragraphe précédent (§5.1.4.), nous pourrions essayer de trouver les conditions que doit remplir le domaine U pour être universel ; c'est-à-dire les conditions pour pouvoir définir de façon précise les fonctions : $\alpha, \beta, \gamma, \delta, \phi$ et ψ. Nous ne ferons pas cela ici, simplement noussupposerons qu'elles existent, et nous verrons dans le modèle (§5.4) comment sont définies ces connexions pour ce domaine – le modèle – en particulier.

Nous représentons le fait que deux domaines sont isomorphes par le symbole $\approx$. Ainsi dans le domaine universel U, nous avons que :

$(\beta \circ \alpha)(U) \approx U \times U$, $(\delta \circ \gamma)(U) \approx U+U$, et $(\psi \circ \phi)(U) \approx [U \rightarrow U]$.

Les fonctions α, γ, ϕ sont appelées <u>fonctions d'interprétation</u> ; et les fonctions β, δ, ψ <u>fonctions de représentation</u>. Dans le premier cas : α, γ, ϕ "interprètent" les éléments du domaine U comme éléments dans les domaines $U \times U$, $U + U$, $[U \rightarrow U]$ respectivement ; dans le second : β, δ, ψ "représentent" les éléments des domaines $U \times U$, $U + U$, $[U \rightarrow U]$ respectivement, dans les éléments du domaine U.

5.3. Solution des équations de domaines

Supposant l'existence des domaines universels réflexifs, nous allons montrer comment obtenir des solutions aux équations de domaines. L'idée est de définir des fonctions continues qui sont des rétracts à caractère fini, dont les images sont isomorphes aux expressions de domaines $F(X)$ (§5). Il est clair que si ces fonctions sont définies dans un domaine, nous pouvons en trouver des points fixes, et ainsi donner des solutions aux équations circulaires de domaines, par exemple : $X = [X \rightarrow X]$. Par conséquent, le premier pas est de donner le domaine où sont définies ces fonctions - rétracts à caractère fini - ; ce domaine est celui des rétracts engendrés continuement (§4.4.4.) et définis dans le domaine universel réflexif U. Nous appellerons ce domaine REC, c'est-à-dire :

$$\begin{aligned} REC &= FEC\ (\lambda g.g^2)_U \\ &= Image\ (\lambda f.\ Cgn\ (\lambda g.g^2)(f)) \\ &\qquad\qquad (\text{où } f \in [U \rightarrow U]\ ) \\ &= Cgn\ (\lambda g.g^2)([U \rightarrow U]) \subseteq [U \rightarrow U] \end{aligned}$$

5.3.1. Caractérisation des opérations x, + et → dans REC

Comme nous avons déjà vu (§5), les expressions de domaines sont formées avec les opérations de produit cartésien ($\times$), somme (+), et exponentation ($\rightarrow$) entre domaines. Nous définissons maintenant les <u>opérateurs "isomorphes" à ces opérations</u> dans le domaine REC, qui se construisent à partir des trois connexions (représentations) à caractère fini qui ont été définies dans le domaine universel U (§5.2).

<u>Lemme 1</u> : <u>(Produit cartésien $\otimes$)</u>

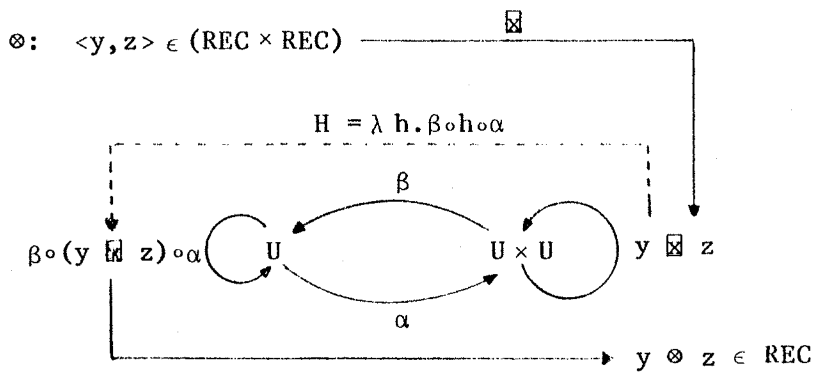
Soit $\otimes$: (REC $\times$ REC) $\to$ REC

définie par : $\otimes\,(y,z) = y \otimes z$

$$= \lambda w.\ \mathrm{Cgn}\ (\beta \circ \boxtimes\ (y,z) \circ \alpha)\ (w)$$

(où $w \in U$) ;

Graphiquement :



$\lambda f.\lambda w.\mathrm{Cgn}(f)(w)$

alors :

(1) $\otimes$ est continue,

(2) Image $(y \otimes z) \approx$ Image $(y) \times$ Image (z)

démonstration :

(1) Comme $\boxtimes$ (théorème §4.5.4.), $\lambda f.\lambda w.\mathrm{Cgn}\,(f)(w)$ (théorème (4.4.2)

et $\lambda h.\beta \circ h \circ \alpha$ (lemme §5.1.1.) sont des fonctions continues, alors :

$\otimes = (\lambda f.\lambda w.\mathrm{Cgn}(f)(w)) \circ (\lambda h.\beta \circ h \circ \alpha\ )\ (\boxtimes)$ est continue ; et par le théorème

§4.4.4.3.; il est facile de voir que $y \otimes z \in$ REC

(2) Image $(y \otimes z)$ = Image $(\beta \circ (y \boxtimes z) \circ \alpha)$ (théorème § 4.4.2. (6))

$\approx$ Image $(y \boxtimes z)$ (lemme §5.1.1.)

et Image $(y \boxtimes z)$ = Image $(y) \times$ Image (z) (théorème §4.5.1.) cqfd .

Un raisonnement semblable nous amène à établir les deux lemmes suivants sur la somme et l'exponentiation.

Lemme 2 : (Somme $\oplus$)

Soit $\oplus$: (REC $\times$ REC) $\rightarrow$ REC

définie par : $\oplus\,(y,z) = y \oplus z$

$$= \lambda w.\mathrm{Cgn}\ (\delta \circ \boxplus(y,z) \circ \gamma)(w)$$

(où $w \in U$)

alors :

(1) $\oplus$ est continue,

(2) Image $(y \oplus z) \approx$ Image (y) + Image (z)

démonstration :

(semblable à celle du lemme 1) cqfd .

Lemme 3 : (Exponentiation $\circ\!\!\rightarrow$)

Soit $\circ\!\!\rightarrow$: (REC $\times$ REC) $\rightarrow$ REC

définie par : $\circ\!\!\rightarrow(y,z) = y \circ\!\!\rightarrow z$

$$= \lambda w.\mathrm{Cgn}\ (\psi \circ \square\!\!\rightarrow(y,z) \circ \phi)(w)$$

(où $w \in U$)

alors :

(1) $\circ\!\!\rightarrow$ est continue,

(2) Image $(y \circ\!\!\rightarrow z)$ $\approx$[Image (y) $\rightarrow$ Image (z)]

démonstration :

(semblable à celle du lemme 1). cqfd .

Notons que : Image (y), Image (z), Image $(y \otimes z)$, Image $(y \oplus z)$, et Image $(y \circ\!\!\rightarrow z)$ sont tous des sous-domaines du domaine universel U.
Avec cette caractérisation des opérations $\times$,+, et $\rightarrow$ en REC, nous pouvons représenter les sous-domaines de U qui sont isomorphes aux expressions de domaines.

5.3.2. Caractérisation des expressions de domaines $F(x)$

Etant donné une expression de domaines $F(X)$, le rétract à caractère fini $\hat{F}(x)$ dont l'image est isomorphe à l'expression de domaines $F(X)$ est obtenu de la façon suivante :

(1) Substituer les variables libres dans $F(X)$ par les rétracts en REC qui caractérisent en U ces variables libres ;

(2) Substituer les opérations $\times$,+, et $\rightarrow$ en $F(X)$ par les opérations respectives (i.e : $\otimes$, $\oplus$, et $\circ\!\!\rightarrow$) en REC ;

(3) Substituer la variable liée X (s'il y en a) par le rétract x ($x \in$REC), unique variable liée en $\hat{F}$.

$\hat{F}$ construite de cette façon est une fonction continue définie en REC (i.e : $\hat{F} \in [\mathrm{REC} \rightarrow \mathrm{REC}]$) , puisqu'il n'y a qu'une seule variable liée. S'il n'y a pas de variable liée, alors $\hat{F}$ est une expression constante en REC.

Exemple :

(Supposons pour simplifier que les opérations $\times$,+ et $\rightarrow$ sont de même priorité, aussi nous devons utiliser une notation complètement parenthésée) .

Soit $F(X) = [(A + (B \times X)) \rightarrow X]$

et soient $a(U) = A \wedge b(U) = B$, $a,b \in$ REC ;

on a: $\hat{F} = \lambda x.(a \oplus (b \otimes x)) \circ\!\!\rightarrow x$ (où $x \in$ REC)

$\wedge\ \hat{F} \in [\mathrm{REC} \rightarrow \mathrm{REC}]$

$\wedge\ (\forall x)\ [x \in \mathrm{REC} \wedge X = x(U) \Rightarrow (\hat{F}(x))(U) \approx F(X)]$

5.3.3. Equations (circulaires) de domaines

Lorsque $X = F(X)$, nous obtenons $\hat{F}$ comme dans le paragraphe précédent (§ 5.3.2.), et comme $\hat{F} \in [\mathrm{REC} \rightarrow \mathrm{REC}]$, alors :

$(\forall x)\ [x \in \mathrm{Cgn}\ (\hat{F})(\mathrm{REC}) \Rightarrow x = \hat{F}(x)]$ (théorème §4.4.2. (3))

et par conséquent, nous pouvons, pour tout $x \in \mathrm{Cgn}\ (\hat{F})(\mathrm{REC})$ interpréter dans le domaine x(U) l'expression de domaines $F(x(U))$, c'est-à-dire :

$$x(U) = (\hat{F}(x))(U) \approx F(x(U))$$

Exemple :

Soit $F(X) = [(A + (B \times X)) \rightarrow X]$

$\wedge\ \hat{F} = \lambda x.(a \oplus (b \otimes x)) \circ\!\!\rightarrow x$ (où $x \in$ REC)

alors :

$(\forall x)(x \in Cgn\ (\hat{F})(REC) \wedge X = x(U) \Rightarrow X \approx [\ (A + (B \times X)) \to X\ ]$

5.4. Le modèle graphique $P\omega$ ([ScD76],[StJ77],[BaH81])

Pour terminer ce chapitre, nous montrons l'existence d'un modèle où sont inclus les différents concepts qui ont été présentés au cours de ce travail. Avec ce modèle sera démontrée la validité du language que nous fournit la théorie des domaines, comme méta-langage pour établir la sémantique des langages de programmation.

Dans le §1.3, nous avons vu que $P\omega$ - l'ensemble des parties des naturels - est un treillis algébrique, où la relation d'ordre partiel est l'inclusion des ensembles et les éléments finis ou compacts sont les ensembles finis de naturels. Ainsi on a :

- $P\omega\ (\subseteq\ ,\ \emptyset,\ \mathbb{N},\ \cup,\ \cap)$
  - $\subseteq$ l'inclusion est la relation d'ordre partiel,
  - $\emptyset$ l'ensemble vide est $\bot$,
  - $\mathbb{N}$ est l'élément $\top$,
  - $\cup$ l'union est la borne supérieure ($\sqcup$)
  - $\cap$ l'intersection est la borne inférieure ($\sqcap$)
- $(\forall x)[\,x \in P\omega \Rightarrow (x \ll x \Leftrightarrow x$ est un sous-ensemble fini de $\mathbb{N})\,]$

Plus loin, en §1.5.1., nous avons vu que $P\omega$ est un domaine de calcul, où l'énumération des éléments de la base (i.e : les éléments finis de $P\omega$) est donnée par :

- Si x fini en $P\omega$, alors $x = b_i$ et $i = \sum_{j \in x} 2^j$

Il nous manque à montrer que $P\omega$ est un domaine universel réflexif (§5.2). Pour celà, nous allons définir des connexions à caractère fini (§5.1.1.) qui caractérisent les sous-domaines de $P\omega$ qui sont isomorphes aux domaines $P\omega \times P\omega$, $P\omega + P\omega$, et $[P\omega \to P\omega]$.

5.4.1. Connexion qui caractérise le produit cartésien (x)

Les fonctions α et β (§5.2) sont définies de la façon suivante :

. $\alpha : P\omega \to P\omega \times P\omega$

$\alpha = \lambda x. <\{n \mid 2n \in x\}, \{m \mid 2m + 1 \in x\}>$

$\beta : P\omega \times P\omega \to P\omega$

$\beta = \lambda x. \{n \mid n = 2i \wedge i \in x \downarrow 1\} \cup \{m \mid m = 2j + 1 \wedge j \in x \downarrow 2\}$

Lemme :

Soient α, β (définies ci-dessus)

alors

$< \alpha, P\omega, P\omega \times P\omega, \beta >$ est une connexion à caractère fini.

Démonstration :

Il est facile de vérifier les assertions suivantes :

(1) α et β sont bien définies et sont monotones,

(2) $\alpha \circ \beta = I_{P\omega \times P\omega}$ et $\beta \circ \alpha = I_{P\omega}$

Par (1) et (2), on a que α et β forment une représentation (§5.1.2.), c'est-à-dire $<< \alpha, P\omega, P\omega \times P\omega, \beta >>$, et par le lemme §5.1.2., elles forment aussi une connexion de Galois. Par le théorème §5.1. (partie (3)) et (2) - ci-dessus - α et β sont des isomorphismes, avec $\alpha = \beta^{-1}$, $\beta = \alpha^{-1}$, et par conséquent sont des fonctions additives (continues) (lemmes 4 et 5, §2.1.2.). Ainsi on a que : $< \alpha, P\omega, P\omega \times P\omega, \beta >$ est une connexion à caractère fini (§ 5.1.1.). On a même le résultat plus fort que α et β forment une représentation à caractère fini entre $P\omega$ et $P\omega \times P\omega$ (§ 5.1.3.) cqfd.

De cette façon : $P\omega \times P\omega \approx \beta \circ \alpha\ (P\omega) = P\omega$

5.4.2. Connexion qui caractérise la somme séparée (+)

Pour définir la connexion à caractère fini $< \gamma, P\omega, P\omega + P\omega, \delta >$ nous établissons d'abord $< \Theta, P\omega \times P\omega, P\omega_1 + P\omega_2, \nu >$ (§ 5.1.4.) avec $e_1 = \{1\}$ et $e_2 = \{2\}$. C'est-à-dire :

$.\theta : P\omega \times P\omega \rightarrow P\omega_1 + P\omega_2$

$$\theta = \begin{cases} \lambda x.\ \text{"}\bot\text{"} & \text{si } x{\downarrow}2 = \bot \\ \lambda x.\ (x{\downarrow}1)_1 \text{ in } P\omega_1 + P\omega_2 & \text{si } x{\downarrow}2 = \{1\} \\ \lambda x.\ (x{\downarrow}1)_2 \text{ in } P\omega_1 + P\omega_2 & \text{si } x{\downarrow}2 = \{2\} \\ \lambda x.\ \text{"}\top\text{"} & \text{sinon} \end{cases}$$

$.\nu : P\omega_1 + P\omega_2 \rightarrow P\omega \times P\omega$

$\nu = \lambda y.\ \underline{\text{sib}}\ (y \in P\omega_1, < y \mid P\omega_1, \{1\} >, < y \mid P\omega_2, \{2\} >)$

Avec la connexion à caractère fini $< \Theta, P\omega \times P\omega, P\omega_1 + P\omega_2, \nu >$ et la connexion du paragraphe précédent (§5.4.1.) $< \alpha, P\omega, P\omega \times P\omega, \beta >$, nous construisons γ et δ comme au § 5.2., c'est-à-dire :

$$\begin{tikzcd}
P\omega \arrow[r, bend right, "\alpha"'] \arrow[rr, bend right=50, "\gamma=\Theta\circ\alpha"'] & P\omega\times P\omega \arrow[l, bend right, "\beta"'] \arrow[r, bend right, "\Theta"'] & P\omega+P\omega \arrow[l, bend right, "\nu"'] \arrow[ll, bend right=50, "\delta=\beta\circ\nu"']
\end{tikzcd}$$

Ainsi on a :

$(\delta\circ\gamma)(P\omega) = (\beta\circ\nu\circ\Theta\circ\alpha)(P\omega) \approx P\omega_1 + P\omega_2$

Dans ce cas, $\delta\circ\gamma$ n'est pas une fermeture, et par conséquent n'est pas une représentation à caractère fini.

5.4.3. Connexion qui caractérise l'exponentiation $(\rightarrow)$

La connexion à caractère fini entre les domaines $P\omega$ et $[P\omega \rightarrow P\omega]$ (i.e. : $< \phi, P\omega, [P\omega \rightarrow P\omega], \psi >$) est définie de la façon suivante :

Soient : $P\omega\ (\subseteq, \bot = \emptyset, \top = \mathbb{N}, \cup, \cap)$

$[P\omega \rightarrow P\omega]\ (\sqsubseteq, \lambda x.\bot, \lambda x.\top, \sqcup, \sqcap)$

alors :

$.\phi : P\omega \to [P\omega \to P\omega]$

$\phi = \lambda w. \bigsqcup\{\lambda x. < b_n, b_m >(x) \mid \tau_2\ (m,n) \in \underline{w}\}$

$.\psi : [P\omega \to P\omega] \to P\omega$

$\psi = \lambda f.\ \{\tau_2(m,n) \mid b_m \subseteq f(b_n)\}$

où :

$-\tau_2$ est la fonction de codification définie au § 3.1.1.5., c'est-à-dire :
$\tau_2 : \mathbb{N}^2 \to \mathbb{N},\ \tau_2 = \lambda x \lambda y\ .\ \frac{1}{2}\ ((x+y)^2 + 3x+y)$, et il existe
$\tau_2^{-1} : \mathbb{N} \to \mathbb{N} \times \mathbb{N},\ \tau_2^{-1} \circ \tau_2 = I_{\mathbb{N} \times \mathbb{N}} \wedge \tau_2 \circ \tau_2^{-1} = I_{\mathbb{N}}$.

$-< b_n, b_m >$ sont les éléments finis de $[P\omega \to P\omega]$ (§ 3.3.1.3.) et le b_i sont les éléments finis de $P\omega$(i est l'indice de l'énumération, § 1.5.1. ou § 5.4.).
$.\underline{w} \in P\omega$ est défini à partir de $w \in P\omega$ par $\underline{w} = w \cup \{0\}$.

L'idée est que $\psi(f)$ est un élément de $P\omega$ qui représente l'ensemble des éléments finis (dans S, § 3.3.1.3.) de $[P\omega \to P\omega]$ qui sont "moins définis" que la fonction $f \in [P\omega \to P\omega]$, en codifiant les indices de ces éléments finis. En d'autres termes, ψ donne la représentation par <u>extension</u> et <u>unique</u> d'une fonction continue en $P\omega$, c'est-à-dire : son <u>graphe</u>. Il est clair que tous les éléments de $P\omega$ ne correspondent pas à des graphes de fonctions continues en $P\omega$, par exemple : les éléments finis de $P\omega$, puisque les graphes sont infinis (i.e. : $\psi([P\omega \to P\omega]) \subset P\omega$). D'un autre côté, $\phi(w)$ est une fonction continue qui est la borne supérieure des éléments finis de $[P\omega \to P\omega]$ dont les indices correspondent aux éléments de l'ensemble w . Pour que ϕ soit totale, c'est-à-dire soit définie quand $w = \bot = \emptyset$,nous ajoutons à w l'élément "0" qui correspond à la fonction $\lambda x.\bot$ (i.e. : $< b_{\tau_2^{-1}(0)\downarrow 2}, b_{\tau_2^{-1}(0)\downarrow 1} > = < b_0, b_0 > = <\bot,\bot> = \lambda x.\bot$).

Les deux fonctions : ϕ et ψ sont connues dans la littérature sur les domaines sous les noms de <u>fun</u> et <u>graph</u> respectivement.
Le lemme suivant établit que ϕ et ψ forment une connexion entre les domaines $P\omega$ et $[P\omega \to P\omega]$.

Lemme :

Soient ϕ , ψ (définies ci-dessus),

alors :

$<\phi\ ,\ P\omega,\ [P\omega \to P\omega]\ ,\ \psi>$ est une connexion à caractère fini.

Démonstration :

Il suffit de montrer :

(1) ϕ et ψ sont bien définies et sont monotones,

(2) $\phi \circ \psi = I_{[P\omega \to P\omega]}$,

(3) ϕ est continue.

Notons que (1) et (2) impliquent que ϕ et ψ forment une connexion de Galois (§ 5.1.)

(1) est immédiat,

(2) $\phi \circ \psi = (\lambda w. \bigsqcup \{\lambda x . <b_n;b_m> (x) \mid \tau_2(m,n) \in \underline{w}) \circ$

$(\lambda f . \{\tau_2(m,n) \mid <b_n;b_m> \sqsubseteq f \})$ (lemme 2 § 3.3.1.3.)

$= \lambda f . \bigsqcup \{\lambda x . <b_n;b_m>(x) \mid \tau_2(m,n) \in \{\tau_2(m,n) \mid <b_n;b_m> \sqsubseteq f\}\}$

(puisque $\forall f,\ \lambda x.\bot \sqsubseteq f \Rightarrow \underline{w} = w$)

$= \lambda f . \bigsqcup \{\lambda x . <b_n;b_m> (x) \mid <b_n,b_m> \sqsubseteq f\}$

$= \lambda f . f$ (puisque $f \in [P\omega \to P\omega]$ et lemme § 3.3.1.4.)

$= I_{[P\omega \to P\omega]}$

(3) soit $W \subseteq P\omega$, $\cup W = \cup\{w \mid w \in W\}$

$\phi(\cup W) = (\lambda w . \bigsqcup \{\lambda x . <b_n;b_m> (x) \mid \tau_2(m,n) \in \underline{w}\})(\cup W)$

$= \bigsqcup\{\lambda x . <b_n;b_m> (x) \mid \tau_2(m,n) \in \underline{\cup W} \}$

$= \bigsqcup\{\lambda x . <b_n;b_m> (x) \mid \tau_2(m,n) \in \cup\{\underline{w} \mid w \in W\}\}$

$= \bigsqcup (\bigcup_{w \in W} \{\lambda x . <b_n;b_m> (x) \mid \tau_2 (m,n) \in \underline{w} \})$

$= \bigsqcup\{ \bigsqcup\{\lambda x . <b_n;b_m> (x) \mid \tau_2(m,n) \in \underline{w} \} \mid w \in W \}$ (théorème 4 § 1.1.5)

$= \bigsqcup\{\phi (w) \mid w \in W\}$ (définition de ϕ)

$= \bigsqcup \phi (W)$

Cela signifie que ϕ est non seulement continue, mais additive (§ 2.1.2.) de même que ψ, puisque la démonstration vient d'être faite pour n'importe quel $W \subseteq P\omega$ et non pour les dirigés cqfd

On peut montrer aussi que $I_{P\omega} \sqsubseteq \psi \circ \phi$, ce qui prouve que ϕ et ψ forment des représentations à caractère fini.
Par le lemme précédent, on a :

$$[P\omega \to P\omega] \approx \quad \psi \circ \phi \; (P\omega) \subseteq P\omega$$

Par conséquent, avec les définitions données pour les fonctions $\alpha, \beta, \gamma, \delta, \phi$ et ψ dans les paragraphes précédents, nous avons montré que $P\omega$ est un domaine universel. Il existe d'autres modèles pour les domaines universels, en particulier : T^{ω} [PoG 78] , où est développé un modèle semblable à celui de $P\omega$, mais avec la notions d'ensemble inductif (§ 1.5.3.).

Un aspect important du modèle $P\omega$ est que l'on peut définir d'une façon précise les fonctions calculables comme un sous-ensemble des fonctions continues de $P\omega$ dans $P\omega$. L'idée consiste à définir les éléments de $P\omega$ qui correspondent aux ensembles récursivement énumérables, et ceci se fait à partir d'un groupe de fonctions continues données dans LAMBDA [ScD 76] et l'opérateur de plus petit point fixe. Ces fonctions correspondent dans la représentation aux ensembles récursivement énumérables.

5.5. Autre point de vue

Dans les paragraphes précédents, nous avons décrit la solution des équations de domaines (§ 5.2.1.3.), en nous fondant sur l'existence d'un domaine universel U (§ 5.2.), pour lequel nous avons trouvé un exemple dans le modèle $P\omega$(§ 5.4.). Le problème de trouver des solutions aux équations de domaines a aussi été abordé en utilisant les concepts de la théorie des catégories. L'idée est la suivante : les opérations de produit cartésien, somme et exponentiation sont des foncteurs dans une certaine catégorie.
De cette façon, dans les équations de domaines $X \approx F(X)$, on considère F comme comme un foncteur, et on cherche à montrer l'existence et la façon d'obtenir un "foncteur inverse" de F , qui soit un isomorphisme. Les détails pour

trouver la solution des équations de domaines en utilisant les concepts de la théorie des catégories, n'entrent pas dans les objectifs ni dans la portée de ce travail. Les principales références sur ce point de vue sont les suivantes : [BeG 79], [SmPo 77] et [WaM 79].

CHAPITRE 6.
SUR L'INTERPRETATION DES TYPES DE DONNEES

6 SUR L'INTERPRETATION des TYPES de DONNEES

Dans le paragraphe § 4.3 nous avons vu que les fermetures, comme fonctions qui caractérisent des domaines, correspondent au concept de type de données dans les languages de programmation, c'est-à-dire : type comme la notion d'un ensemble de "valeurs" - le domaine des valeurs - qui peuvent être affectés à un identificateur. L'idée de type comme dénotation de fermeture se trouve dans [ScD 76] .
Avec cette correspondance : type-fermeture, on résout les problèmes reliés à la sémantique des types, en particulier la vérification de la cohérence des types [DoJ 79] .

En général, nous dirons qu'un type dénote un rétract à caractère fini - r.c.f. - (§ 2.3.4). Pour les types simples : bool, nat, ... l'interprétation (r.c.f. associé) dans le modèle $P\omega$ (§ 5.4) pourrait être la suivante :

Exemple 1 : Bool $\in$ $[P\omega \rightarrow P\omega]$

$$\begin{aligned}\text{Bool} = \lambda x.\ &\text{si } x = \bot \text{ alors } \bot \\ &\text{sinon si } x = \{0\} \text{ alors } \{0\} \\ &\qquad \text{sinon si } x = \{1\} \text{ alors } \{1\} \\ &\qquad\qquad \text{sinon } \top\end{aligned}$$

(où {0} interprête la valeur faux , {1} la valeur vrai, $\bot$ et $\top$ les valeurs indéfini et surdéfini respectivement). De cette façon, bool dénote la fermeture Bool en $P\omega$ (i.e. : Interprétation $[\![\text{bool}]\!]$ = Bool = $(\text{Bool})^2$). De façon semblable, on pourrait obtenir nat (voir Nat dans ex. § 2.1.2.).
Les opérateurs -r.c.f. - que nous avons présentés dans les paragraphes § 4.5. - 5.3.1., interprètent les types suivants dans les langages de programmation : soient a et b (r.c.f.), qui caractérisent (interprètent) les types t1 et t2 respectivement, alors :

- a $\boxtimes$ b interprête le type <t1, t2> (type structuré) ;
- a $\circ\!\!\rightarrow$ b interprête le type des fonctions dont le domaine est l'ensemble des valeurs de type t1, et dont le résultat est l'ensemble des valeurs de type t2 ;
- a $\boxplus$ b interprête l'union de types, aussi bien que la surcharge (anglais : overloading) d'un identificateur d'opérateur, par exemple : + entre entiers, + entre réels, ...

Les types de données plus complexes : ensnat (ensemble de naturels), listnat (liste de naturels), ... sont interprétés comme r.c.f. résultats d'équations de domaines (§ 5.3.2.). Ainsi on a :

Exemple 2 : Interprétation $[\![\text{ensnat}]\!]$ = Ensnat
et Ensnat = Nat $\circ\!\!\rightarrow$ Bool
(où les fonctions du type Nat $\circ\!\!\rightarrow$ Bool correspondent aux fonctions caractéristiques des éléments - ensembles de naturels - du type ensnat).

Exemple 3 : Interprétation $[\![\text{listnat}]\!]$ = Listnat
et Listnat = <> $\oplus$ (Nat $\otimes$ Listnat)
(où <> interprète la liste vide).

Dans ce dernier exemple, nous avons un type dont l'interprétation est définie par une équation récursive de domaines (§ 5.3.3.)

L'inclusion des types de données abstraits dans les langages de programmation (CLU[LSAS77], ALPHARD[SWL 77], EUCLID[Laal 77] , ADA[ADA 81] , RUSSELL[DeDo 79] , ML[GMW 79]) établit la notion de type comme une classe d'objets - valeurs - qui sont caractérisés par les opérations disponibles sur ces objets. Cela signifie que le type de données est associé à l'ensemble des fonctions qui opèrent sur les données. Dans ce cas, la sémantique du type de données est définie par la sémantique des fonctions associées au type. Par exemple, pour le type listnat, les fonctions associées sont : lvide, cons, tête, reste, vide, long, appartient. Ces fonctions sont définies dans le formalisme de ML de la façon suivante :

Exemple 4 :

```
absrectype listnat = . + nat # listnat
with lvide = abslistnat (inl () )
and cons (i,l) = abslistnat (inr (i,l) )
and tête l = fst (outr (replistnat (l) ))
and reste l = snd (outr (replistnat (l) ))
and vide l = isl (replistnat (l) ) ;;
letrec long (l) = if vide (l) then 0 else 1 + long (reste (l) ) ;;
letrec appartient (i,l) = if vide (l) then false
            else i = tête (l) or appartient (i, reste (l)) ;;
```

D'une manière équivalente, on peut présenter le même exemple avec les opérateurs présentés au cours de ce travail :

Exemple 5 :

Soit le domaine $D_1 = \overset{T}{\underset{\bot}{\vdots}}$, caractérisé par le r.c.f. : <>, où "." est la liste vide, et soit D_2 = Nat ⊗ Listnat, alors :

- . Listnat : <>⊕ (Nat ⊗ Listnat)
  (i.e. : Listnat point fixe de $\lambda\rho.\diamond\oplus(Nat\otimes\rho)$, §5.3.3.),
- . lvide = $\lambda().\delta(._1 in D_1 + D_2)$
- . cons = $\lambda <i,l> . \delta(\beta(<i,l>_2 in D_1 + D_2)$
- . tête = $\lambda l. (\alpha (\gamma (l)| D_2))\downarrow 1$
- . reste = $\lambda l. (\alpha (\gamma (l)| D_2))\downarrow 2$
- . vide = $\lambda l.l \in D_1$
- . long = $\mu (\lambda f. \lambda l.$ <u>sib</u> (vide (l), 0, 1 + f (reste (l))))
- . appartient = $\mu (\lambda f. \lambda <i,l>.$<u>sib</u> (vide (l), faux, (i = tête (l)) ∨ f(i, reste(l))))

où : α, β, γ, δ, sont les fonctions des connexions présentées dans le paragraphe § 5.2. - domaines universels -, les autres opérateurs étant définis dans les paragraphes : § 3.1 - 3.2 - 4.2.1 - 5.3.1.

De cette façon, la sémantique de listnat est donnée par le tuple :

< lvide, cons, tête, reste, vide, long, appartient > ∈

[→Listnat] × [Nat × Listnat → Nat] × [Listnat → Nat] × [Listnat → Listnat] × [Listnat → Bool] × [Listnat → Nat] × [Nat × Listnat → Bool] .

Ce tuple de fonctions et les domaines caractérisés par Listnat, Nat et Bool, forment une algèbre hétérogène [BiLi 70].

De là vient le point de vue algébrique [GuHo 78] où on établit la sémantique des fonctions associées au type abstrait de données au moyen d'axiomes.

Normalement, dans l'approche algébrique, on distingue deux parties dans la spécification : la syntaxe et la sémantique. Dans le cas du type abstrait listnat, on a :

Exemple 6 :

. syntaxe (1) lvide : → listnat

(2) cons : nat × listnat → listnat

(3) tête : listnat → nat

(4) reste : listnat → listnat

(5) vide : listnat → bool

(6) long : listnat → nat

(7) appartient : nat × listnat → bool

. sémantique

(1) vide (lvide) = vrai

(2) vide (cons(i,l)) = faux

(3) tête (cons(i,l)) = i

(4) reste (cons (i,l)) = l

(5) long (lvide) = 0

(6) long (cons(i,l)) = 1 + long(l)

(7) appartient (j, lvide) = faux

(8) appartient (j, cons(i,l)) = si i=j alors vrai
sinon appartient (j,l)

Les fonctions associées au type abstrait listnat (i.e. : lvide, cons, ...) sont du "type" exprimé dans la partie syntaxique, et doivent "satisfaire" les huit équations (axiomes) présentés dans la partie sémantique. Il est facile de vérifier que les fonctions définies dans l'exemple 5 satisfont l'axiomatisation de l'exemple 6.

En [GTW 78], on précise la sémantique exprimée par les axiomes au moyen de la notion d' "initialité", qui consiste à définir dans l'algèbre des termes (univers de Herbrand), donnée par la partie syntaxique, les classes de congruence engendrées par les équations - partie sémantique - (§ 6.2.1).

Toutes ces approches de spécification de la sémantique des types abstraits sont étendues pour admettre la paramétrisation des types (et des fonctions) par des types. De tels types de données paramétrés ont été appelés "polymorphes" [MiR 77] , ou "génériques"[BeJa 77] utilisant la terminologie des langages de programmation. Par exemple, le type listnat peut admettre l'abstraction du type des éléments de la liste, et l'on obtient le type générique : liste [elem] . Sa spécification est la suivante :

Exemple 7 :

. syntaxe (1) lvide :→liste [elem]

(2) cons : elem x liste [elem]→ liste [elem]

(3) tête : liste [elem]→ elem

(4) reste : liste [elem]→ liste elem

(5) vide : liste [elem]→ bool

(6) long : liste [elem]→ nat

. sémantique

identique aux axiomes (1) à (6) qui apparaissent dans la partie sémantique de l'exemple 6.

Pour la fonction appartient associée à listnat, nous ne pouvons pas avoir l'abstraction du type des éléments, à moins que nous ayons aussi l'abstraction de la fonction "=", qui est, elle, associée au type des éléments. En faisant ces deux abstractions dans la fonction appartient, nous pouvons compléter l'exemple 7 avec :

. syntaxe (7) appartient : elem × liste [elem] × [elem × elem → bool] → bool

. sémantique (7) appartient (j, lvide, f) = faux

(8) appartient (j, cons(i,l),f) = si f(i,j) alors vrai sinon appartient (j,l,f)

Le problème qui existe avec cette abstraction de la fonction "=" dans appartient est que seul est exprimé le type de la fonction f associé au type des éléments (i.e. : f est de type [elem × elem → bool]), c'est-à-dire, l'abstraction est syntaxique. Pour réaliser l'"abstraction sémantique" de la fonction f, on définit la notion de propriété [BeJa 78]. Dans l'exemple, f doit satisfaire la propriété de l'égalité, c'est-à-dire, f doit être la fonction d'égalité associée au type des éléments. De cette façon, on dit que le type générique liste[elem] exige la propriété d'égalité sur le type effectif qui est substitué au type formel elem. Le point important est que les propriétés peuvent être spécifiées de façon semblables aux types abstraits. Ainsi la propriété égalité sur elem avec la fonction eq peut s'exprimer par :

Exemple 8 :

. syntaxe (1) eq : elem $\times$ elem $\rightarrow$ bool

. sémantique

(1) eq(x,x) = vrai

(2) eq(x,y) = eq(y,x)

(3) eq(x,y) $\wedge$ eq(y,z) $\wedge \neg$ eq(x,z) = faux

La sémantique des types génériques et la paramétrisation a été étudiée dans divers travaux ([ReJ 74] ,[McN 79] ,[KaA 78] ,[TWW 78] , [BeD 79]). Les langages de spécification qui incluent la définition de types abstraits, fondent la sémantique sur la notion de théorie ([NHM 77] , [BuGo 80]).

Ici, nous allons présenter une interprétation des types de données, en tentant de préciser la notion de satisfaction avec la théorie (théorie de Scott-Strachey) que nous avons développé dans ce travail. Comme il vient d'être montré, la notion de satisfaction est exprimée par le fait qu'une fonction $f \in [D_1 \rightarrow D_2]$ (ou n-uplet de fonctions) vérifie une équation de la forme $G(f) = F(f)$, où on a que $G,F \in [[D_1 \rightarrow D_2] \rightarrow D_3]$. Dans [BeSo 81] nous avons montré la nécessité de manier cette notion de satisfaction pour capturer le concept de "généricité" (abstraction sémantique) dans les types abstraits paramétrés (types génériques), en ayant la possibilité de réaliser de cette manière un typage des types (i.e. : propriétés), ce qui amène à une vérification de la cohérence des types à un autre niveau.

6.1. Interprétation de la notion de satisfaction ($\models$)

Dans ce paragraphe, nous allons associer à la notion de satisfaction un rétract à caractère fini (r.c.f.) dont l'image (les points fixes) contient les fonctions qui vérifient une équation (les axiomes) et qui peut être obtenu par les opérateurs continus de points fixes (ocpf - § 4.4). L'idée est la suivante :

Soient . $G, F \in [D \rightarrow D]$, D domaine

. $f \in D$

alors, le prédicat suivant est vrai :

$(\forall f)\ [f \in FIX\ (G) \cap FIX\ (F) \Rightarrow G(f) = F(f)]$

c'est-à-dire, f satisfait l'équation $G(f) = F(f)$ (i.e. : $f \models G(f) = F(f)$).

Maintenant, nous présentons un lemme dans lequel nous donnons un rétract à caractère fini dont l'image contient l'intersection des images des deux rétracts à caractère fini. Dans notre cas, ces deux images correspondent à FEC (G) $\subseteq$ FIX (G) et FEC (F) $\subseteq$ FIX (F) (§ 4.4), c'est-à-fire les points fixes engendrés continuement des fonctions G et F. Le lemme est le suivant :

Lemme 1 :

Soient . $G,F \in [D \to D]$, D domaine

. $f \in D$

alors :

$$(\forall f)\ [f \in FEC\ (G) \cap FEC\ (F) \Rightarrow f \in FEC\ (G \cup F)]$$

Démonstration :

On a, $FEC(G \cup F) = \text{Image}\ (\lambda f.Cgn\ (G \cup F)(f))$ (théorème 4.4.2. (5))

$= \text{Image}\ (\lambda f.\bigsqcup(\bigcup_{n=o}^{\infty}\ \{(G \cup F)^n(e)\ |e \in E \wedge e \sqsubseteq f \ \ \wedge e \sqsubseteq (G \cup F)\ (e)\}\ ))$ (où E est la base de D)

de plus, $f \in FEC(G) \cap FEC(F)$

(1) $f = \bigsqcup E(G,f) \wedge f = \bigsqcup E(F,f)$ (lemme 4.4.1)

(où : $E(G,f) = \{e\ |e \in E \wedge e \sqsubseteq f \wedge\ e \sqsubseteq G(e)\}$

$E(F,f) = \{e\ |e \in E \wedge e \sqsubseteq f \wedge e \sqsubseteq F(e)\}$)

$\wedge$ (2) $G(f) = F(f) = f$ (théorème 4.4.2 (3))

De (1) et par définition de $\bigsqcup$ (i.e. : $G \sqsubseteq' G \cup F \wedge F \sqsubseteq' G \cup F$), on a :

(3) $e \in E(G,f) \vee e \in E(F,f) \Rightarrow e \in E(G \cup F,f)$

(où $E(G \cup F,f) = \{e\ |e \in E \wedge e \sqsubseteq f \wedge e \sqsubseteq (G \cup F)(e)\}$)

De (2) on obtient :

(4) $G(f) = F(f) = f \Rightarrow (G \cup F)(f) = G(f) \cup F(f)$ (§ 3.3.1.2)

$= f \cup f = f$

Par (3) : $f = \bigsqcup E\ (G,F) \sqsubseteq \bigsqcup E(G\cup F),f)$ (théorème 3 § 1.1.5)

$\sqsubseteq \bigsqcup(\bigcup_{n=o}^{\infty}\ \{(G\cup F)^n\ (e)|\ e \in E(G\cup F,f)\ \})$ (théorème 2 § 1.1.5)

$\sqsubseteq Cgn\ (G\cup F)(f)$

Par (4) : $(\forall e)(\forall n)\ [n \in \mathbb{N} \wedge e \sqsubseteq f \Rightarrow (G \cup F)^n(e) \sqsubseteq F]$

de celà on déduit : Cgn $(G \cup F)(f) \sqsubseteq f$

ainsi on a : $f = \text{Cgn } (G \cup F)(f)$ (i.e. : $f \in \text{FEC } (G \cup F)$)

par conséquent : $\text{FEC } (G) \cap \text{FEC } (F) \subseteq \text{FEC } (G \cup F)$ cqfd

De cette façon, nous interprétons l'intersection de deux domaines caractérisés par des r.c.f. par un r.c.f. dont l'image contient l'intersection. Dans quelques cas, il est possible de caractériser exactement l'intersection, par exemple, quand G ou F est l'identité, ou une fonction bijective. Cela signigie qu'il faut connaître avec plus d'exactitude la structure de G,F,f et D.

Dans l'image du r.c.f. que nous avons obtenu (i.e. : $\lambda f.\text{Cgn}(G \cup F)(f)$) nous avons montré que l'on trouve des f qui sont les points fixes engendrés continuement des fonctions G et F, et par conséquent vérifient l'équation $G(f) = F(f)$. Cependant, il y a d'autres f qui peuvent vérifier $G(f) = F(f)$ et ne sont pas des points fixes de G et F. De plus, nous avons vu que la notion de satisfaction est donnée par le fait qu'une fonction $f \in D_1 = [D_1' \to D_1'']$ vérifie une équation $G(f) = F(f)$, où $G,F \in [D_1 \to D_2]$. Maintenant, nous donnons un lemme qui garantit l'existence de deux r.c.f. : $G',F' \in [D \to D]$, tels que l'intersection de leur image contient uniquement <u>toutes</u> les fonctions f qui vérifient $G(f) = F(f)$.

<u>Lemme 2</u> :

Soient
- . D_1, D_2 domaines,
- . $f \in D_1$,
- . $G,F \in [D_1 \to D_2]$ (i.e. :G,F continues)
- . $D = D_1 \times D_2$
- . $G' = D \to D \wedge G' = \lambda x. \langle I(x \downarrow 1), G(x \downarrow 1)\rangle$
- . $F' = D \to D \wedge F' = \lambda x. \langle I(x \downarrow 1), F(x \downarrow 1)\rangle$

(I est l'identité en D_1)

alors : (1) G' et F' sont bien définies et sont continues,

(2) G' et F' sont des rétracts à caractère fini,

(3) $(\forall f)\ [f \models G(f) = F(f) \Longleftrightarrow f \in (\text{FEC } (G') \cap \text{FEC } (F'))\downarrow 1]$

<u>Démonstration</u> :

(1) Immédiat, il suffit de voir la définition de $\sqcup$ en § 3.1.1.2

(2) Notons que : $<I,G> : D_1 \to (D_1 \times D_2)$

$$<I,G> = \lambda f.\ <I(f),\ G(f)> = \lambda f.\ <f,\ G\ (f)>$$

est continue (voir (1)) ;

et $\downarrow 1 : D_1 \times D_2 \to D_1$, $\downarrow 1 = \lambda <f,x>\ .f$

(i.e. : $\downarrow 1(<f,x>) = <f,x>\downarrow 1 = f$)

est continue (§ 3.1 et 3.1.1.2.);

et de plus :

$$<\downarrow 1,\ D_1 \times D_2,\ D_1, <I,G>>$$

est une connexion à caractère fini (§ 5.1.1), puisque :

$$\downarrow 1 \circ <I,G> = I$$

Par conséquent : $G' = <I,G> \circ \downarrow 1$ est un rétract à caractère fini, et le domaine caractérisé par G' est isomorphe à D_1.
La démonstration pour F' est similaire, et le domaine caractérisé par F' est aussi isomorphe à D_1. Et les deux fonctions (G' et F') caractérisent des sous-domaines de $D = D_1 \times D_2$.

(3) Immédiat, puisque :

$(\forall f)\ [f \in D_1 \Longleftrightarrow <f,G(f)> \in D \wedge <f,F(f)> \in D$

$\wedge\ <f,G(f)> \in FIX(G') = FEC(G')$ (théorème 4.4.2.(6))

$\wedge\ <f,F(f)> \in FIX(F') = FEC(F')]$ (th. 4.4.2.(6))

de plus,

$(\forall x)\ [x \in D \Longrightarrow (x \in FEC(G') \cap FEC(F')$

$\Longleftrightarrow <x\downarrow 1,\ x\downarrow 2> \in FEC(G') \wedge <x\downarrow 1, x\downarrow 2> \in FEC(F')$

$\Longleftrightarrow G(f) = F(f)]$ cqfd.

Comme conséquence des lemmes 1 et 2, nous interprèterons la notion de satisfaction par la condition nécessaire que nous donne le rétract à caractère fini $\lambda f.\ S\,(G,F)(f)$.

En d'autres mots :

$$\text{Interprétation}[\![G(f) = F(f)]\!] = S(G,F)$$

et on a :

$$f \models G(f) = F(f) \Longrightarrow S(G,F)(f) = f$$

où : $S(G,F) = \downarrow 1 \circ (\lambda\langle f,y\rangle.\mathrm{Cgn}(\lambda\langle f,y\rangle.\mathrm{Cgn}(\langle I,G\rangle \circ \downarrow 1)(\langle f,y\rangle)$
$\cup\ \lambda\langle f,y\rangle.\mathrm{Cgn}(\langle I,F\rangle \circ \downarrow 1)(\langle f,y\rangle))(\langle f,y\rangle)) \circ (\lambda f.\langle f,y\rangle)$
(où $y \in D_2$) (voir lemme §5.1.1 avec $D=D_1$ et $D'=D_1 \times D_2$).

6.2. Interprétation des types abstraits algébriques

Dans l'exemple 6 (§6), on a donné une spécification algébrique du type listnat. Maintenant, nous présentons le formalisme algébrique pour la spécification des types de données abstraits. Puis nous donnerons une interprétation des types abstraits dans le domaine universel U (§ 5.2.).

6.2.1. Définition algébrique des types abstraits [GTW 78]

La spécification d'un type abstrait ι est donnée par le triplet : $\langle S, \Sigma, E \rangle$

où : . S est un ensemble d'indices (anglais : sorts),

. Σ est une famille d'ensembles d'opérateurs, où l'ensemble d'indices est $S^* \times S$, S^* est l'ensemble des chaines finies incluant le vide $\langle\rangle$ d'éléments de S. C'est-à-dire, $\Sigma = \langle \Sigma_i \rangle \quad (i \in S^* \times S)$

tel que,

$$\sigma \in \Sigma_{w,s} \wedge w \in S^* \wedge s \in S \Longrightarrow \sigma : w \to s$$

i.e. : σ est un opérateur de domaine w et de codomaine s.

. E est un ensemble d'équations (lois, identités, axiomes), que doivent satisfaire les opérateurs en Σ.

Le type abstrait ι appartient à S^+ (les chaines finies et non vides d'éléments de S). Par exemple, pour listnat (ex.6 - §6), on a :

$S = \{listnat, bool, nat\}$, $\tau = listnat$

$\Sigma_{<>, listnat} = \{lvide\}$

$\Sigma_{<nat, listnat>, listnat} = \{cons\}$

$\Sigma_{<listnat>, nat} = \{tête, long\}$

...

$E = \{vide\ (lvide) = vrai, \ldots\}$.

Comme nous l'avons dit, le type abstrait correspond à la notion d'algèbre (hétérogène). Pour le moment, on a les noms (indices) des ensembles (S), les identificateurs des opérateurs (Σ), leur fonctionnalité donnée par l'ensemble d'indices ($S^{*} \times S$), et les équations (E) que doivent satisfaire les opérateurs. Il manque à donner les ensembles où sont définis les opérateurs, c'est-à-dire, la famille d'ensembles dont les indices appartiennent à S, et la définition des opérateurs sur ces ensembles. Dans l'approche "initiale" [GTW 78], cette famille est constituée des ensembles de termes (univers de Herbrand) qui sont construits à partir de Σ, et sont dénotés par $T_\Sigma = \langle T_{\Sigma,s} \rangle_{s \in S}$

C'est-à-dire :

(c1) $\quad \sigma \in \Sigma_{<>,s} \Rightarrow \sigma \in T_{\Sigma,s}$

(c2) $\quad \sigma \in \Sigma_{<s1,\ldots sn>,s} \wedge t_1 \in T_{\Sigma,s1} \wedge \ldots \wedge t_n \in T_{\Sigma,sn}$

$$\Rightarrow \sigma(t_1,\ldots,t_n) \in T_{\Sigma,s}$$

Par la définition des opérateurs (Σ_T) en T_Σ, on a :

(o1) $\quad \sigma \in \Sigma_{<>,s} \Rightarrow \sigma_T \in \Sigma_T \wedge \sigma_T : \to T_{\Sigma,s}$

$$\wedge\ \sigma_T = \sigma \in T_{\Sigma,s}$$

(o2) $\quad \sigma \in \Sigma_{<s_1,\ldots,sn>,s} \wedge t_1 \in T_{\Sigma,s_1} \wedge \ldots\ t_n \in T_{\Sigma,sn}$

$$\Rightarrow \sigma_T \in \Sigma_T \wedge \sigma_T : T_{\Sigma,s1} \times \ldots \times T_{\Sigma,sn} \to T_{\Sigma,s}$$

$$\wedge \sigma_T\, [t_1,\ldots,t_n] = \sigma(t_1,\ldots t_n) \in T_{\Sigma,s}$$

On appelle T_Σ (i.e. : $\langle T_\Sigma\ ,\ \Sigma_T \rangle$) la Σ algèbre initiale.

Le terme "initial" se réfère à l'objet initial d'une catégorie. Nous ne donnerons pas ici ce concept (voir : [ArMa75] , [GTWW77]).

Pour que les opérateurs (Σ_T) satisfassent les équations (E), on construit l'algèbre quotient :

$T_{\Sigma,E} = T_\Sigma / \equiv_E$, où $\equiv_E$ est la relation de congruence engendrée par E.
Le type abstrait τ est défini comme le type d'"intérêt" [GuHo78] de l'algèbre $T_{\Sigma,E}$.

6.2.2. Interprétation d'un type abstrait

Dans le paragraphe précédent, nous avons vu l'approche algébrique, où la sémantique d'un type abstrait est donnée par l'algèbre des termes $T_{\Sigma,E}$. Maintenant, nous associons aux éléments de Σ (opérations) et E (équations) de la spécification d'un type abstrait $\tau = (S,\Sigma,E)$ un objet (r.c.f.) dans le domaine universel. C'est-à-dire, de même que l'ensemble d'indices S (noms des types) dénotent des ensembles de valeurs données par des rétracts à caractère fini -r.c.f.- (i.e. : Nat, Bool ; ex.1, §6), la paire (Σ,E) dénotera le r.c.f. $S(G,F)$ -§6.1- où se trouvent les fonctions (Σ) qui satisfont les équations (E), et sont définies dans les domaines dénotés par les éléments de S. De cette façon, l'ensemble des équations E est transformé en une équation de la forme G(f) = F(f) de la manière suivante :

. Soient $r_{s_1}, r_{s_2}, \dots, r_{s_k}$ les r.c.f. associés à chaque indice de S (i.e. : $S = \{s_1, s_2, \dots, s_k\}$) ; on a :

(1) pour $w \in S^* \wedge w \neq \langle\rangle \wedge w = s_1 s_2 \dots s_n$, le r.c.f. associé est

$r_w = r_{s_1} \otimes r_{s_2} \otimes \dots \otimes r_{s_n}$,

(2) pour $w \in S^* \wedge w = \langle\rangle$, le r.c.f. associé est
$r_w = r_{\langle\rangle}$ et Image $(r_{\langle\rangle}) = \begin{smallmatrix}\top\\ \vdots\\ \bot\end{smallmatrix}$

(3) pour $\langle w,s\rangle \in S^* \times S$, le r.c.f. associé est
$r_{w,s} = r_w \circ\!\!\rightarrow r_s$

. Soient $\sigma_1, \sigma_2, \dots, \sigma_n$ les opérateurs en Σ

(i.e. : $\bigcup_{i \in S^* \times S} \Sigma_i = \{\sigma_1, \sigma_2, \dots, \sigma_n\}$) on a,

si $\sigma_j \in \Sigma_{w,s}$ alors $\sigma_j' \in$ Image $(r_{w,s})$, où σ_j' est l'interprétation de σ_j, i.e. : $r_{w,s}$ est le type de l'opérateur σ_j.

(Dans ce qui suit, $j(j=1,\dots,n)$ identifie les paires w,s ; c'est-à-dire : σ_j' $\in$ Image (rj), $j = 1,\dots,n$)

. Ainsi nous obtenons l'équation

$$G(f) = F(f)$$

où . $f \in D_1 \approx \text{Image } (r_1 \otimes r_2 \otimes \dots \otimes r_n)$

(i.e. : Interprétation $[\![<\sigma_1,\sigma_2,\dots,\sigma_n>]\!] = f$)

. $G,F \in [D_1 \to D_2]$

$G = <G_1,G_2,\dots,G_m>$, $F = <F_1,F_2,\dots,F_m>$

avec, $G_i,F_i \in [D_1 \to D_{2,i}]$ $(i = 1,2,\dots,m)$

qui correspondent aux m équations en E, et $<G_1,G_2,\dots,G_m>$,

$<F_1,F_2,\dots,F_m> \in [D_1 \to D_{2,1} \times \dots \times D_{2,m}]$, c'est-à-dire :

$D_2 = D_{2,1} \times \dots \times D_{2,m}$.

Comme nous l'avons vu dans le modèle $P\omega$ - §5.4 -, les fonctions continues comprennent les fonctions calculables (§ 5.4.3). Par conséquent, nous pouvons exprimer les équations en E à travers les fonctions continues G_i,F_i (i.e. : $G_i(f) = F_i(f)$).

De cette façon, l'interprétation d'un type abstrait τ avec spécification ou dénotation (S,Σ,E) est :

$$\begin{aligned} \text{Interprétation } [\![\tau]\!] &= \text{Interprétation } [\![(S,\Sigma,E)]\!] \\ &= \text{Interprétation } [\![S]\!] \otimes \text{Interprétation } [\![(\Sigma,E)]\!] \\ &= (r_{s_1} \otimes r_{s_2} \otimes \dots \otimes r_{s_k}) \otimes \mathcal{S}(G,F) \end{aligned}$$

Pour continuer, nous donnons une condition qui permet d'identifier les fonctions qui appartiennent à $\mathcal{S}(G,F)$, et qui, prises avec les domaines caractérisés par les r.c.f. associés à S, sont des algèbres isomorphes à l'algèbre initiale $T_{\Sigma,E}$.

<u>Condition d'initialité</u>

Soit $f \in$ Image $(\mathcal{S}(G,F))$, tel que $f \models G(f) = F(f)$;

nous disons que $f = <\sigma'_1,\sigma'_2,\dots,\sigma'_n>$ est une <u>interprétation initiale</u> de $<\sigma_1,\sigma_2,\dots,\sigma_n>$ ssi

$$\text{Image } (\sigma'_j) \approx \{\sigma_j(t) \mid \sigma_j(t) \in T_{\Sigma,E} \wedge t \in T^*_{\Sigma,E}\} \quad (j = 1,2,\dots,n)$$

6.3. Interprétation des propriétés

Dans l'exemple 8 (§6) nous avons vu que les propriétés sont utilisées pour spécifier l'abstraction sémantique (et syntaxique) des fonctions, afin d'obtenir les types génériques.

La notion de propriété est celle de "type des types", c'est-à-dire, la propriété est vue comme un ensemble de types de données abstraits, dont les fonctions, ou sous-ensembles de celles-ci, satisfont les schémas d'équations qui apparaissent dans la propriété, et c'est précisément cela l'interprétation que nous allons donner aux propriétés.

Comme nous l'avons vu dans le paragraphe précédent, les types abstraits sont interprétés par des rétracts à caractère fini. Par conséquent, REC (§5.3) est l'ensemble de tous les types de données, c'est-à-dire, la propriété à un seul type formel, dans laquelle il n'y a pas d'opérateurs ni d'axiomes. Ainsi, les propriétés sont interprétés pas des r.c.f. dont le domaine de définition est REC.

La spécification d'une propriété π est semblable à celle d'un type abstrait :

$$\pi = \lambda t.(S,\Sigma,E)$$

où l'indice t ($t \in S$) est formel (i.e. : est une variable de type). L'interprétation d'une propriété π est la suivante :

$$\text{Interprétation } [\![\pi]\!] = \text{Interprétation } [\![\lambda t.(S,\Sigma,E)]\!]$$
$$= \lambda\tau'.(c_1(\tau'\downarrow 1) \otimes \mathcal{S}(G_\pi,F_\pi)(c_2(\tau'\downarrow 2)))$$

où . τ' est l'interprétation d'un type abstrait,

(i.e. : τ' = Interprétation $[\![\tau]\!] = (r_{s_1} \otimes \ldots \otimes r_{s_k}) \otimes \mathcal{S}(F,G)$;

. $\tau' \downarrow 1 = r_{s_1} \otimes \ldots \otimes r_{s_k}$ et $\tau' \downarrow 2 = \mathcal{S}(F,G)$;

. c_1 et c_2 sont les fonctions de sélection qui donnent les r.c.f. (associés aux indices) et fonctions qui entrent dans les propriétés ;

. $\mathcal{S}(G_\pi,F_\pi)$ le r.c.f. associé aux opérations et équations de la propriété.

De cette façon, on a qu'un type abstrait τ satisfait la propriété π (i.e. : $\tau \models \pi$) ssi

$$\text{Interprétation } [\![\pi]\!] \ (\text{Interprétation } [\![\tau]\!])$$
$$= c_1 \ (\text{Interprétation } [\![\tau]\!] \downarrow 1)$$
$$\otimes \ c_2 \ (\text{Interprétation } [\![\tau]\!] \downarrow 2)$$

Nous pouvons aussi établir une relation d'ordre partiel entre les propriétés [BeD 79] de la façon suivante :

$$\pi_1 \le \pi_2 \iff \text{Image } (\text{Interprétation } [\![\pi_2]\!])$$
$$\subseteq \text{Image } (\text{Interprétation } [\![\pi_1]\!])$$

C'est-à-dire : les types abstraits qui satisfont la propriété π_2 satisfont aussi la propriété π_1, en formule :

$$\pi_1 \le \pi_2 \iff (\forall \tau) \ [\tau \models \pi_2 \Rightarrow \tau \models \pi_1]$$

6.4. Interprétation des types génériques

La spécification d'un type générique (ex.7; §6) est semblable à celle présentée pour les types abstraits et les propriétés :

$$\lambda t.\tau[t{:}\pi] = \lambda t.(S,\Sigma,E,\pi)$$

où t est une variable de type qui doit être substituée par un type effectif τ_0, et celui-ci doit satisfaire la propriété π. De cette façon, l'interprétation d'un type générique est une fonction définie dans l'ensemble des types abstraits, c'est-à-dire :

$$\text{Interprétation } [\![\lambda t.\tau[t{:}\pi]]\!] : \text{REC} \rightarrow \text{REC}$$

et $$\text{Interprétation } [\![\lambda t.\tau[t{:}\pi]]\!] \ (\text{Interprétation } [\![\tau_0]\!])$$
$$= \text{Interprétation } [\![\tau[\tau_0]]\!]$$

Cette dernière application exprime l'"instanciation" d'un type générique par le type effectif τ_0. Comme nous l'avons mentionné plus haut, l'instanciation est possible si τ_0 satisfait la propriété π. D'où le résultat suivant qui nous permet de vérifier statiquement la validité de l'instanciation d'un type générique.

6.5. Cohérence dans les types génériques

Soit le type générique $\lambda t.\tau[t{:}\pi] = \lambda t.(S,\Sigma,E,\pi)$, l'interprétation d'une instanciation par τ_0 est définie ssi τ_0 satisfait la propriété π. C'est-à-dire :

$$\text{Interprétation } [\![\lambda t.\tau[t:\pi]]\!] \; (\text{Interprétation } [\![\pi]\!] \; (\text{Interprétation } [\![\tau_0]\!])) = \text{Interprétation } [\![\tau[\tau_0]]\!] \; .$$

CONCLUSION

7. CONCLUSION

Au cours de ce travail, on a donné les éléments d'un meta-langage pour la description sémantique des langages de programmation. En particulier, ce méta-langage décrit les langages où les calculs se réalisent de façon essentiellement séquentielles [BeG 78]. Un point important que nous avons vu de cette théorie dans laquelle est basé le méta-langage, est qu'elle est développée indépendamment de son application postérieure comme outil de desciption sémantique. Une nouvelle présentation vient d'ailleurs d'en être faite par D. Scott lui-même, d'une manière plus axiomatique [ScD 82]. Les concepts fondamentaux de la théorie ont été : l'existence de points fixes dans les fonctions (continues), comment les calculer, et la possibilité de représenter les fonctions comme éléments des ensembles mêmes (domaines universels) dans lesquels elles sont définies; c'est-à-dire : les éléments de ces domaines peuvent être interprétés comme fonctions d'un ordre quelconque. Ces concepts permettent d'établir la sémantique des langages de programmation, et de plus fournissent une méthodologie pour la conception de ces langages [TeR 77]. On a aussi réalisé des travaux dans lesquels on engendre un compilateur pour un langage, à partir de sa description sémantique par la méthode dénotationnelle ([MoP 75],[MoP 79]).

Avec cette théorie, on a essayé d'exprimer le concept de satisfaction (§ 6.1) à travers l'intersection de rétracts à caractère fini. Ce problème est simple quand on travaille avec des fermetures (§ 4.3.4), et il faudrait en faire une étude plus approfondie pour le cas général (c.à.d. rétracts à caractère fini).

Une suite du travail que nous avons présenté est de trouver une structure convenable pour les types abstraits, structure qui tienne compte des axiomes à satisfaire. Par exemple :

$$(\forall \tau_1, \tau_2)[\tau_1 \in REC \wedge \tau_1 \models \pi_1 \wedge \tau_2 \in REC \wedge \tau_2 \models \pi_2 \wedge \pi_1 \sqsubseteq \pi_2$$
$$\Rightarrow \tau_1 \sqsubseteq \tau_2]$$

De cette façon, nous avons une relation d'ordre pour les types de données abstraits, et cette relation est induite par les propriétés. Avec ces notions, nous pouvons essayer de trouver des solutions aux équations de types de la forme :

$$t = \tau[t]$$

où la spécification de τ est donnée à travers des axiomes.

BIBLIOGRAPHIE

8 BIBLIOGRAPHIE

|ADA 81]

"Reference Manual for the Ada Programming Language" ; Lecture Notes in Computer Science, 106 ; Springer-Verlag ; 1981.

|ArMa 75]

Arbib M.A., Manes E.G. ; Arrows Structures and Functors : The Categorical Imperative ; Academic Press ; 1975.

|BaH 81]

Barendregt H. ; The Lambda Calculus, Its Syntax and Semantics ; North-Holland ; 1981.

|BeD 79]

Bert D. ; La Programmation Générique : Construction de Logiciel, Spécification Algébrique et Vérification ; Thèse d'Etat, Université de Grenoble I ; 1979.

|BeG 78]

Berry G. ; "Séquentialité de l'évaluation formelle des λ-expressions ; Proc. 3ème Colloque International sur la Programmation, Paris ; DUNOD ; 1978.

|BeG 79]

Berry G. ; Modèles Complètement Adéquats et Stables des Lambda-Calculs Typés ; Thèse d'Etat, Université de Paris VII ; 1979.

|BeJa 77]

Bert D., Jacquet P. ; "Generic abstract data types" ; Proc. 5th Annual III Conference, WG 2.1, IFIP ; 1977.

|BeJa 78]

Bert D., Jacquet P. ; "Some validation problems with parameterized types and generic functions" ; 3rd International Symposium on Programming ; IP., Paris ; Dunod ; 1978.

|BeSo 81]

Bert D., Soler R. ; "About Data Type Genericity" ; International Colloquium on Formalization of Programming Concepts, Lecture Notes in Computer Science, 107 ; 1981.

|BiLi 70]

Birkhoff G., Lipson J.D. ; "Heterogeneous Algebras" ; J. Combinatorial Theory, 8 ; pp. 115-133 ; 1970.

[BrF 81]

Bracho, F. ; Finitary Retracts and Fixed Points ; Comunicaciones Técnicas N°. 257, IMAS-UNAM ; 1981 (Presentado en International Colloquium on Formalization of Programming Concepts, Peñiscola-España, Abril 1981).

[BuGo 80]

Burstall R., Goguen J.A. ; "The Semantics of CLEAR, a specification Language" ; Lecture Notes in Computer Science, 86 ; Springer-Verlag ; 1980.

[ChA 41]

Church, A. ; The Calculi of Lambda-Conversion ; Princenton University Press, 1941.

[CoPR 77]

Cousot P., Cousot R. ; Constructive Versions of Tarski's Fixed Point Theorems ; RR 85, Laboratoire IMAG - Grenoble ; 1977.

[DaM 58]

Davis, M. ; Computability & Unsolvability ; Mc Graw-Hill ; 1978.

[dBJ 80]

de Bakker, J. ; Mathematical Theory of Program Correctness ; Prentice-Hall ; 1980.

[DeDo 79]

Demers A., Donahue J. ; Revised Report on Russell ; TR 79-389 ; Cornell University, Dpt. of Computer Science ; 1979.

[DoJ 79]

Donahue, J. ; "On the Semantics of Data Type" ; Siam J. Comput, 8 ; pp. 546-560 ; 1979.

[GMW 79]

Gordon M.J., Milner A.J., Wadsworth C.P. ; "Edimburgh LCF" ; Lecture Notes in Computer Science, 78 ; Springer-Verlag ; 1979.

[GrG 78]

Grätzer, G. ; General Lattice theory ; Academic Press ; 1978.

[GTWW 77]

Goguen J.A., Thatcher J.W., Wagner E.G., Wright J.B. ; Initial Algebra Semantics and Continuous Algebras ; JACM, 24 ; pp. 68-95 ; 1977.

[GTW 78]
Goguen J.A., Thatcher J.W., Wagner E.G. ; "An initial algebra approach to the specification, correctness, and implementation of abstract data types"; Current Trends in Programming Methodology, IV; (ed. Yeh) Prentice-Hall;1978.

[GuHo 78]
Guttag J.V., Horning J.J. ; "The algebraic specification of abstract data types" ; Acta Informatica, 10 ; pp. 27-52 ; 1978.

[HLS 72]
Hindley J., Lercher B., Seldin J. ; Introduction to Combinatory Logic ; London Mathematical Society Lecture Notes Series, 7 ; Cambridge University Press ; 1972.

[IwT 44]
Iwamura, T. ; "A lemma on directed sets" ; Zenhoku Shijo Sugaku Danwakai 262, pp. 107-111 ; 1944.

[KaA 78]
Kanda A. ; "Data Types as Initial Algebras : An Unification of Scottery and ADJery", 19th Annual Symposium on Foundations of Computer Science ; pp. 221-230 ; 1978.

[KLS 52]
Kleene, S. ; Introduction to metamathematics ; D. Van Nostrand Co. Inc. ; 1952.

[Laal 77]
Lampson B.W., Horning J.J., London M.L., Mitchell J.G., Popek G.L. ; "Report on the programming Language EUCLID" ; SIGPLAN Notices, 12:2 ; 1977.

[LiC 78]
Livercy, C. ; Théorie des Programmes - Schémas, preuves, sémantique ; DUNOD ; 1978.

[LSAS 77]
Liskov B., Snyder A., Atkinson R., Schaffert C. ; "Abstraction Mechanisms in CLU ; CACM, 20 ; p. 564 ; 1977.

[MaG 76]
Markowsky, G. ; "Chain-complete Posets and Directed Sets with Applications" ; Algebra Univ., 6 ; pp. 53-68 ; 1976.

[MaG 77]
Markowsky, G. ; A motivation and Generalization of Scott's Notion of a continuous Lattice ; RC 6617 (#28520) ; IBM Research Division ; 1977.

[MaRo 76]

Markowsky G., Rosen B. ; "Bases for Chain-Complete Posets" ; IBM J. RES. DEVELOP ; pp. 138-147 ; 1976.

[McJ 63]

Mc Carthy, J. ; "A Basis for a Mathematical Theory of Computation" ; Computer Programming and Formal Systems (Ed. Braffort-Hirschberg) ; pp. 33-69 ; North-Holland ; 1963.

[McN 79]

Mc Cracken, N. ; An Investigation of a Programming Language with a Polymorphic Type Structure ; Ph.D. Syracuse University ; 1979.

[MeRi 67]

Meyer A., Ritchie D. ; "The complexity of loop programs", Proc. 22nd National ACM Conference ; pp. 465-470 ; 1967.

[MiM 67]

Minsky, M. ; Computation : Finite and Infinite Machines ; Prentice-Hall ; 1967.

[MiR 77]

Milner R. ; "A Theory of Type Polymorphism in Programming" ; Report CSR-9-77 ; Univ. of Edinburgh, Dpt. of Computer Science ; 1977.

[MoP 75]

Mosses, P. ; Mathematical Semantics and Compiler Generations ; Ph.d. Thesis, University of Oxford ; 1975.

[MoP 79]

Mosses, P. ; SIS-Semantics Implementation System ; AARHUS UNIVERSITY, DAIMI MD-30 ; 1979.

[NHN 77]

Nakajima R., Honda M., Nakahara H. ; "Describing and Verifying Programs with Abstract Types" ; Formal Description of Programming Concepts ; (ed. Nevhold) North-Holland ; 1977.

[OrO 44]

Ore, O. ; "Galois Conexions" ; Trans. Amer. Math. Soc., 55 ; pp. 493-513 ; 1944.

|PoG 78|

Plotkin, G. "T^{ω} as a Universal Domain" ; JOURNAL OF COMPUTER AND SYSTEM SCIENCES, 17; pp. 209-236 ; 1978.

|ReJ 74|

Reynolds J.C. ; "Towards a Theory of Type structure" ; Colloquium of Programming ; Paris ; 1974.

|RoH 67|

Rogers, H. ; Theory of Recursive Functions and Effective Computability ; Mc Graw-Hill ; 1967.

|SaL 77|

Sanchis, L. ; "Data Types as Lattices : Retractions, Closures and Projections" ; RAIRO Informatique Théorique, 11 ; pp. 329-344 ; 1977.

|ScD 70|

Scott, D. ; "Outline of a Mathematical Theory of Computation" ; Proceedings of the Fourth Annual Princeton Conference on Information Sciences and Systems ; pp. 169-176 ; Princenton University ; 1970.

|ScD 72|

Scott, D. ; "Continuous Lattices" ; Toposes, Algebraic Geometry and Logic ; pp. 97-136 ; Springer-Verlag, Berlin ; 1972.

|ScD 76|

Scott, D. ; "Data Types as Lattices" ; SIAM Journal on Computing, 5, pp. 522-587 ; 1976.

|ScD 80|

Scott, D. ; "Retracts as Objects of Computation" ; Bulletin of the European Association for Theoretical Computer Science, 11 ; 1980.

|ScD 82|

Scott, D. ; "Domains for Denotational Semantics"; ICALP'82 ; Lecture Notes in Computer Science, 140 ; 1982.

|ScSt 71|

Scott D., Strachey C. ; "Toward a mathematical semantics for computer languages" ; Proceedings of the symposium on Computers and Automata ; pp. 19-46, Polytechnic Press of the Polytechnic Institute of Brooklyn ; 1971.

[SmPo 77]
Smyth M., Plotkin G. ; "The Category-Theoretic Solution of Recursive Domain Equations" ; Proc. 18th Annual IEEE Symp. on Foundations of Computer Science, Providence, R.I. ; 13-17 ; 1977.

[StJ 77]
Stoy, J. ; Denotational Semantics : The Scott-Strachey Approach to Programming Language Theory ; MIT Press ; 1977.

[SWL 77]
Shaw M., Wulf W.A., London R.L. ; "Abstraction and Verification in ALPHARD : Defining and Specifying Iteration and generators ; CACM, 20 ; p. 553 ; 1977.

[TaA 33]
Tarski, A. ; "The Concept of Truth in Formalized Languages" ; Logic, Semantics, Metamathematics - Papers from 1923 to 1938 by A. Tarski (Translated by J.H. Woodger) ; pp. 152-278 ; Oxford at the Clarendon Press ; 1956.

[TaA 36]
Tarski, A. ; "The Establishment of Scientific Semantics" ; Logic, Semantics, Metamathematics - Papers from 1923 to 1938 by A. Tarski (Translated by J.H. Woodger ; pp. 401-408 ; Oxford at the Clarendon Press ; 1956.

[TaA 55]
Tarski, A. ; "A lattice theoretical fixpoint theorem and its applications" ; Pacific Journal of Mathematics, 5 ; 1955.

[TeR 76]
Tennent, R. ; "The Denotational Semantics of Programming Languages" ; CACM, 19 ; pp. 437-453 ; 1976.

[TeR 77]
Tennent, R. ; "Language Design Methods Based on Semantic Principles" ; ACTA INFORMATICA, 8 ; pp. 97-112 ; 1977.

[TWW 78]
Thatcher J.W., Wagner E.G., Wright J.; "Data-type specification : parametrization and the power of specification techniques" ; Proc. of the SIGACT, 10th Annual Symposium on Theory of Computing.

[VuJ 74]
Vuillemin, J. ; Syntaxe, Semantique et Axiomatique d'un langage de Programmation Simple ; Thèse d'Etat, Université de Paris ; 1974.

[WaM 79]
Wand, M. ; "Fixed-Point Constructions in Order-Enriched Categories" ;
Theoretical Computer Science, 8 ; pp. 13-30 ; 1979.

Dernière page d'une thèse

VU

Grenoble, le

Le Président de la thèse

Vu, et permis d'imprimer,

Grenoble, le

Le Président de l'Université Scientifique et Médicale

M. TANCHE