Recherche de preuves en logique spatiale

Rapport de stage

Samuel MIMRAM

Stage effectué sous la direction de
Daniel Hirschkoff
et
Étienne Lozes

Équipe PLUME

LIP – ENS Lyon

Résumé

Le π-calcul est un formalisme qui permet de modéliser la programmation concurrente. Notre travail a consisté en la conception et la mise en œuvre d'une procédure permettant de décider, et ce de façon automatique, la validité de certaines propriétés structurelles et comportementales des processus du π-calcul. Ces propriétés sont formulées dans une logique spatiale adaptée au π-calcul et les preuves sont construites dans un calcul de séquents pour cette logique. Les contributions principales de ce stage ont été la définition d'un système de séquents que nous avons montré être correct et complet, et l'implémentation d'une procédure de recherche de preuves pour ce système de séquents.

Le stage a été effectué à l'ÉNS Lyon sous la direction de messieurs Daniel Hirschkoff et Étienne Lozes.

1 Introduction

Le π-calcul est un outil pour modéliser la programmation concurrente. Ce calcul introduit un formalisme qui permet de rendre compte de l'indéterminisme au niveau de l'exécution de programmes parallèles ; la mobilité et la distribution des calculs y sont également bien modélisables. Il est brièvement introduit en §2.1. Il est intéressant par ce qu'il est riche du fait du passage de noms qu'il contient. En effet, le λ-calcul, les calculs orientés objet ou encore les calculs impératifs peuvent y être implémentés (cf. en particulier [Mil91] et [Mil99]).

Il est intéressant de pouvoir automatiquement prouver des propriétés sur les processus du π-calcul en vue, par exemple, de pouvoir assurer certaines propriétés de programmes, et ce de façon transparente pour le programmeur. Pour pouvoir statiquement exprimer ces propriétés, plusieurs approches sont possibles. On peut typer les processus ce qui est en général peu coûteux en temps de calcul mais les types manquent souvent d'expressivité ; on peut utiliser des logiques modales qui permettent bien de décrire les interactions entre processus mais pas leur structure ; on peut enfin utiliser des logiques spatiales qui permettent généralement de décrire à la fois la structure des processus et les interactions entre eux – c'est l'approche que nous avons choisie.

Une logique spatiale et un système de séquents correct pour cette logique ont été proposés par Caires et Cardelli dans [CC03]. Cette logique est globalement indécidable, c'est pourquoi nous nous sommes intéressés à un sous-ensemble de celle-ci qui est présenté en §2.2. Celui-ci n'est pas réduit au point d'être trivialisé ; il contient en particulier les opérateurs $|$ (qui rend compte de la concurrence) et $\triangleright$ (qui permet d'exprimer des propriétés sur l'environnement d'un processus). Des extensions de la logique ont de plus été proposées pour exprimer des propriétés sur un modèle plus proche du π-calcul traditionnel (cf. §6).

Notre travail a tout d'abord consisté à lire et comprendre des articles sur le sujet (cf. bibliographie). Il nous a fallu intégrer les implications profondes de la concurrence présente dans le π-calcul ainsi que tenter d'appréhender l'expressivité de la logique étudiée, due à l'introduction d'opérateurs nouveaux – en particulier l'opérateur $\triangleright$, qui cache une quantification d'ordre supérieur, est loin d'être anodin. Il a été montré dans [CCG02] que le sous-ensemble de la logique que nous avons étudié était décidable et nous avons cherché à la décider de façon constructive en utilisant les séquents. Une partie de notre travail a en effet consisté en la réalisation d'un programme en OCaml permettant de décider de la validité d'une proposition logique et qui donne une dérivation permettant de prouver la formule en cas de succès. Les preuves sont construites dans un calcul de séquents que nous avons défini et prouvé être correct et complet. Ce sont l'élaboration de ce système de séquents et la mise au point des preuves associées à ce système qui ont constitué la majeure et la plus dure partie de notre travail.

Une application de ce que nous avons fait pourrait être la mise au point d'un système de types évolué (les types peuvent être vus comme des formules logiques exprimant des propriétés sur les objets qu'ils typent) et partiellement automatiquement inféré dans lequel on pourrait par exemple imposer à une fonction f d'avoir un type de la forme $(0||0) \to \top$ *i. e.* son unique

argument doit être constitué d'au plus un thread (cf. *infra*). Lors d'un appel à la fonction f, il faudrait vérifier que l'argument qui lui est fourni est bien constitué d'un seul thread, soit encore que le type de cet argument est un sous-type de $(0||0)$. La relation de sous-typage nécessiterait alors de s'appuyer sur des preuves de validité : on a $T_1 <: T_2$ si et seulement si $T_1 \Rightarrow T_2$, ce qui pourrait être fait statiquement et automatiquement par notre procédure (de même que par exemple OCaml est en mesure d'assurer statiquement la sûreté du typage grâce à son système d'inférence de types).

Dans la suite, nous commençons par introduire informellement le π-calcul (§2.1), la logique spatiale étudiée (§2.2) et les séquents (§2.3) puis nous exposons trois systèmes de séquents (§4) que nous avons étudiés – le dernier (§4.3), que nous avons montré être correct et complet, étant celui utilisé pour la recherche de preuves par notre programme – après avoir établi les notions fondamentales nécessaires pour étudier ces systèmes (§3). Enfin, nous proposons un dernier système de règles de séquents (§6) qui est une extension du troisième système à une logique dont les modèles sont plus proches des processus habituels du π-calcul.

2 Cadre théorique

2.1 Introduction au π-calcul

Nous nous contentons ici d'un bref rappel de ce qu'est le π-calcul, le lecteur intéressé par une présentation plus formelle pourra consulter le livre de Milner [Mil99].

Les acteurs du π-calcul sont les *processus*. Ils communiquent par l'intemédiaire de canaux dont les noms sont notés a, b, c, etc. Un processus peut :

- ne rien faire (c'est le processus vide, noté 0) ;
- émettre le nom de canal b sur le canal a, ce que l'on note $a\langle b\rangle$;
- recevoir un nom de canal sur le canal a, ce que l'on note $a(x)$ (après réception, x sera remplacé par le nom reçu).

L'opérateur $|$ permet de spécifier qu'un processus est composé de deux processus qui sont simultanément (parallèlement) actifs. Tout processus peut être considéré comme s'exécutant en parallèle avec un processus vide. Si un processus envoie un nom de canal sur un canal et qu'un autre processus est en écoute sur ce même canal une réduction a lieu, notée $\rightarrow$. Ainsi on a la réduction suivante :

$$a\langle b\rangle \mid a(x).c\langle x\rangle \rightarrow 0 \mid c\langle b\rangle$$

Le lecteur attentif aura certainement remarqué lors d'une telle réduction, dans le processus récepteur, la variable x est remplacée par le nom de canal reçu.

La congruence structurelle $\equiv$ quotiente les processus par associativité et commutativité de l'opérateur $|$, ainsi que par ajout d'un processus vide en parallèle. Les processus sont étudiés modulo un quotient par cette relation.

Si deux processus veulent en même temps écrire ou lire sur le même canal, la réduction n'est plus déterministe ; c'est en cela que le π-calcul permet de modéliser la concurrence. On peut par exemple avoir :

$$a\langle b\rangle \mid a(x).c\langle x\rangle \mid a(x).d\langle x\rangle \rightarrow c\langle b\rangle \mid a(x).d\langle x\rangle$$

ou

$$a\langle b\rangle \mid a(x).c\langle x\rangle \mid a(x).d\langle x\rangle \rightarrow a(x).c\langle x\rangle \mid d\langle b\rangle$$

D'autres opérateurs existent en π-calcul, comme par exemple ν qui permet de restreindre l'utilisation d'un nom de canal à un processus particulier. Nous ne détaillerons pas ici ces opérateurs qui ne sont pas pris en compte par le sous-ensemble étudié de la logique.

2.2 La logique spatiale

La logique spatiale utilisée au cours du stage est un sous-ensemble de celle définie dans [CC03] ; elle est plus formellement présentée en §4.1.

Cette logique définit, en plus des opérateurs habituels $\mathbf{F}$, $\wedge$ et $\Rightarrow$, des opérateurs propres à l'étude des processus du π-calcul : 0 est une formule qui n'est satisfaite que par un processus vide, l'opérateur de composition parallèle $|$ (un processus P vérifie $\mathcal{A}|\mathcal{B}$ si et seulement s'il est congru à $Q|R$ avec Q vérifiant $\mathcal{A}$ et R vérifiant $\mathcal{B}$; d'où l'adjectif *spatial* qualifiant la logique) et l'opérateur de garantie $\triangleright$ entre autres. Elle est rendue très expressive en particulier grâce à cet opérateur $\triangleright$: un processus P satisfait la formule $\mathcal{A} \triangleright \mathcal{B}$ si et seulement si, pour tout processus Q qui satisfait $\mathcal{A}$, le processus $P|Q$ satisfait $\mathcal{B}$. Le connecteur $\triangleright$ permet donc d'exprimer des propriétés sur l'environnement du processus P. Ainsi, on pourrait par exemple imaginer exprimer des propriétés de sécurité d'un programme par des formules de la forme

$$\top \triangleright \neg\Diamond(\top|\mathcal{A}_{\text{sécu}})$$

où $\mathcal{A}_{\text{sécu}}$ est une formule qui exprimerait qu'une information sensible se retrouverait publiquement disponible (la formule $\top$ est vérifiée par tout processus et un processus vérifie la formule $\Diamond\mathcal{A}$ si et seulement s'il peut se réduire en un processus vérifiant $\mathcal{A}$). Ainsi, intuitivement, un programme vérifiant cette propriété garantit qu'il ne va pas communiquer d'informations avec un pirate. On peut aussi exprimer des propriétés sur la structure d'un programme ; la formule suivante permet d'exprimer le fait que le programme n'exécutera jamais plus d'un thread à la fois :

$$\neg\Diamond(\neg 0|\neg 0)$$

Il a été montré dans [CT01] que la validité dans une logique spatiale est, lorsqu'on considère celle-ci dans son ensemble, indécidable. Elle l'est en particulier lorsque l'on se restreint aux opérateurs $\mathbf{F}$, $\wedge$, $\Rightarrow$, 0, $|$, $\triangleright$, $b.$ (nous définissons ce dernier opérateur en §6) et que l'on s'autorise une quantification universelle sur les noms de canaux. Nous nous sommes donc restreints aux opérateurs $\mathbf{F}$, $\wedge$, $\Rightarrow$, 0, $|$ et $\triangleright$ pour l'implémentation (une extension à CCS, qui est un sous-ensemble du π-calcul, a de plus été proposée, cf. §6). Cette restriction peut s'interpréter en termes d'unions finies d'intervalles de $\mathbb{N}$ (cf. remarque 2) mais nous avons raisonné dans l'optique de réaliser un prouveur qui pourra aisément être enrichi par de nouveaux opérateurs – il serait en particulier intéressant d'enrichir le prouveur de l'opérateur ν ainsi que du passage de nom ou d'une restriction décidable de ces opérateurs. Quelques exemples de formules sont donnés en §3.3 et §5.3.

Les problèmes classiques du model checking et de la validité sont interdérivables dans cette logique restreinte. En effet, un processus P valide une formule $\mathcal{A}$ si et seulement si la formule $\chi(P) \Rightarrow \mathcal{A}$ est valide (où $\chi(P)$ est la formule caractéristique de P, cf. la définition 13). Réciproquement, l'opérateur dérivé $\sim$ (cf. définition 15) permet d'exprimer la validité d'une formule $\mathcal{A}$: la formule $\mathcal{A}$ est valide si et seulement si pour un processus P quelconque (0 par exemple) on a $P \models \neg(\sim \mathcal{A})$. Là encore, dans le but de réaliser un prouveur le plus complet et efficace possible, nous avons considéré le problème sous l'angle de la validité car il nous permet d'une part d'exploiter le formalisme introduit par les séquents – ce qui permet de réduire le nombre de cas à traiter – et permet une conception que nous pensons facilement enrichissable avec d'autres opérateurs logiques, en particulier $b.$ et $\Diamond$ (cf. §6) et peut-être aussi $\mathcal{U}$ (cf. [CC02] et [CC03] pour la définition de cet opérateur).

2.3 Prouver la validité, séquents pour la logique spatiale

Les séquents sont une présentation formelle d'un système de déduction qui a été introduite par Gentzen (cf. [Gal87], par exemple). Ils présentent une façon de manipuler les opérateurs tout en préservant la validité des jugements. Une grosse partie de notre travail théorique a consisté à trouver et à prouver la consistance et la complétude des règles de séquents que nous avons été amenés à introduire, justifiant ainsi que notre programme s'arrête toujours, qu'il trouve une preuve d'un séquent si et seulement si celui-ci est valide, et que les preuves qu'il fournit sont correctes. Les séquents sont bien adaptés à la logique avec laquelle nous avons travaillé – qui est classique – car ils rendent compte du tiers-exclu.

Un séquent se présente sous la forme $\langle S \rangle\ \Gamma \vdash \Delta$. Γ et Δ sont des ensembles contenant des couples notés $u : \mathcal{A}$ qui signifient « l'index u vérifie la propriété $\mathcal{A}$ » (un index peut

s'interpréter comme une variable portant sur les processus du π-calcul). L'ensemble Γ peut se lire comme une conjonction d'hypothèses et Δ comme une disjonction de propriétés à prouver (c'est cette disjonction qui permet aux séquents de rendre compte des logiques classiques). L'ensemble S a été rajouté aux séquents dans [CC03] et est nouveau ; il fait bien partie des séquents et n'est pas un ensemble de *side conditions*. C'est un ensemble de contraintes, appelé *théorie de contraintes*, de la forme $u \doteq X_1|\dots|X_n$ (les X_i sont des variables de processus). L'intérêt de la logique étudiée par rapport, par exemple, aux logiques modales, c'est qu'elle permet d'étudier des propriétés structurelles des processus ; et cette structure est stockée dans S sous forme de contraintes sur les index (qui peuvent s'interpréter comme « u est composé de n processus en parallèle – ces derniers étant éventuellement nuls »). Intuitivement, ces contraintes expriment la « granularité » avec laquelle nous allons pouvoir observer les processus ; dans le cas de l'index u mentionné ci-dessus, nous n'allons pouvoir en distinguer que n composantes.

Enfin, l'élimination des coupures est vérifiée pour notre système de séquents, ce qui rend la réalisation d'un prouveur envisageable. En effet, il a été montré dans [CC03] que s'il existe une dérivation permettant de prouver un séquent qui utilise la règle de coupure, à savoir

$$\frac{\langle S\rangle\ \ \Gamma \vdash u : \mathcal{A}, \Delta \quad \langle S\rangle\ \ \Gamma, u : \mathcal{A} \vdash \Delta}{\langle S\rangle\ \ \Gamma \vdash \Delta}\,(\text{Cut})$$

alors il existe une dérivation n'utilisant pas la règle de coupure et permettant de prouver ce même séquent[1]. L'on peut donc implémenter le prouveur sans avoir à se préoccuper de cette règle qui aurait grandement augmenté la difficulté de la réalisabilité d'un prouveur (voire l'aurait rendue impossible) car elle introduirait une recherche exhaustive sur les formules.

3 Définitions préliminaires

Nous commençons par introduire formellement les notions fondamentales qui nous seront nécessaires par la suite.

3.1 Processus, formules

Définition 1 (Processus). L'ensemble des processus du π-calcul est défini[2] inductivement par

$$P, Q ::= 0 \quad | \quad b\,(x)\,.P \quad | \quad b\,\langle x\rangle\,.P \quad | \quad P|Q \quad | \quad \nu x.P$$

où $b \in \mathcal{N}$ est un nom de canal et $x \in \mathcal{N}'$ est une variable de nom de canal.

Définition 2 (Processus insécable). Un processus P du π-calcul est dit insécable si et seulement si

$$P \equiv Q|R \quad \Rightarrow \quad Q \equiv 0 \text{ ou } R \equiv 0$$

(la congruence structurelle $\equiv$ ici utilisée est une extension de celle introduite à la définition 4 ; elle est formellement définie dans [CC02] par exemple).

Lemme 1 (Processus insécables). *L'ensemble des processus insécables du π-calcul est l'ensemble des processus de la forme : 0 ou $b\,(x)\,.P$ ou $b\,\langle x\rangle\,.P$ ou $\nu x.(P_1|\dots|P_n)$ avec, pour tout $i \in [\![1, n]\!]$, P_i insécable non nul et $x \in \mathrm{fcv}(P_i)$ où $\mathrm{fcv}(P)$ désigne l'ensemble des variables de nom de canal apparaissant libres[3] dans le processus P.*

[1] En réalité, l'élimination des coupures n'a été montrée dans [CC03] que pour le système de séquent présenté dans ce papier. Cependant elle est modulaire et s'adapte aisément pour montrer que les systèmes de séquents étudiés éliminent aussi les coupures.

[2] Nous ne considérons que les processus finis *i. e.* définis sans l'opérateur !.

[3] Nous ne précisons pas la définition de $\mathrm{fcv}(P)$ car elle ne présente que peu d'intérêt pour la suite.

La logique que nous allons étudier (cf. définition 5 et suivantes) ne contient que | et 0 comme connecteurs permettant d'exprimer des propriétés structurelles sur les processus. Elle ne sera donc pas à même de « distinguer » deux processus insécables non nuls (*i. e.* plus formellement, si $\mathcal{A}$ est une formule, C un contexte de processus, et P et Q sont deux processus insécables non nuls alors on a : $C[P] \models \mathcal{A}$ si et seulement si $C[Q] \models \mathcal{A}$). Notre logique ne permet que « d'observer » si un processus est nul ou s'il est composé de un ou plusieurs processus insécables en parallèle (ce qui revient à « compter » sur $\mathbb{N}$, mais – comme nous l'avons déjà mentionné – nous n'adopterons pas ce point de vue car notre objectif est de réaliser un prouveur facilement extensible). Nous pouvons donc, pour simplifier les raisonnements et les preuves, et sans perdre en généralité, nous restreindre à un modèle plus simple dans lequel les processus insécables non nuls sont tous représentés par une unique constante arbitrairement nommée 1.

Définition 3 (Processus). L'ensemble $\mathcal{P}$ des processus est défini inductivement par

$$P, Q ::= 0 \quad | \quad 1 \quad | \quad P|Q$$

Définition 4 (Congruence structurelle sur les processus). La congruence structurelle, notée $\equiv$, est la plus petite relation d'équivalence telle que :

$$\begin{aligned} &P|0 \equiv P \\ &P|Q \equiv Q|P \\ &P|(Q|R) \equiv (P|Q)|R \end{aligned}$$

Dans la suite, le modèle implicitement utilisé pour la logique (en particulier lorsque nous parlerons de complétude) est l'ensemble $\mathcal{P}$ des processus, quotienté par congruence structurelle.

Définition 5 (Formule). L'ensemble $\mathcal{F}$ des formules est défini inductivement par :

$$\mathcal{A}, \mathcal{B} ::= \mathbf{F} \quad | \quad A_w \quad | \quad \mathcal{A} \wedge \mathcal{B} \quad | \quad \mathcal{A} \Rightarrow \mathcal{B} \quad | \quad 0 \quad | \quad \mathcal{A}|\mathcal{B} \quad | \quad \mathcal{A} \triangleright \mathcal{B}$$

où A est un élément de l'ensemble χ des variables propositionnelles et w un entier qui représente la « taille maximale » d'une formule (la taille est à prendre au sens de la définition 10) que la variable propositionnelle A peut représenter.

L'ensemble $\mathcal{F}_0$ des formules closes est l'ensemble des formules dans lesquelles n'apparaissent pas de variables propositionnelles.

Les définitions suivantes sont mutuellement récursives (mais on peut montrer qu'elles ont un sens *i. e.* qu'elles terminent).

Définition 6 (Ensemble de propriété descriptible par une formule close de taille w). L'ensemble $\mathbb{F}_w$ des ensembles de propriété descriptible par une formule close de taille au plus w est défini par $\mathbb{F}_w = \{[\![\mathcal{A}]\!] \quad / \quad \mathcal{A} \in \mathcal{F}_0 \wedge \|\mathcal{A}\| \leq w\}$ (la taille sur les formules ici utilisée est formellement introduite à la définition 10).

On note $\mathbb{F}$ l'ensemble limite $\mathbb{F} = \bigcup_{w>0} \mathbb{F}_w$.

Remarque 1. Les domaines de validité des $\mathcal{A}$ ne dépendent pas de la valuation choisie car la formule $\mathcal{A}$ est supposée close c'est pourquoi nous n'avons pas indexé le domaine de validité de $\mathcal{A}$.

Lemme 2. *Si $w' \leq w$ alors $\mathbb{F}_{w'} \subseteq \mathbb{F}_w$.*

Définition 7 (Valuation). Une valuation v est une fonction d'un sous-ensemble fini de χ vers $\mathbb{F}$ qui à une variable propositionnelle A_w associe un élément de $\mathbb{F}_w$.

Définition 8 (Domaine de validité, validation). Le domaine de validité d'une formule sous la valuation v est défini inductivement par :

$$\begin{array}{l}
[\![\mathbf{F}]\!]_v = \emptyset \\
[\![0]\!]_v = \{0\} \\
[\![A_w]\!]_v = v(A_w) \\
[\![\mathcal{A} \wedge \mathcal{B}]\!]_v = [\![\mathcal{A}]\!]_v \cap [\![\mathcal{B}]\!]_v \\
[\![\mathcal{A} \Rightarrow \mathcal{B}]\!]_v = \left(\complement_{\mathcal{P}} [\![\mathcal{A}]\!]_v\right) \cup [\![\mathcal{B}]\!]_v \\
[\![\mathcal{A}|\mathcal{B}]\!]_v = \{P|Q \quad / \quad P \in [\![\mathcal{A}]\!]_v \wedge Q \in [\![\mathcal{B}]\!]_v\} \\
[\![\mathcal{A} \triangleright \mathcal{B}]\!]_v = \{P \in \mathcal{P} \quad / \quad \forall Q \in [\![\mathcal{A}]\!]_v, P|Q \in [\![\mathcal{B}]\!]_v\}
\end{array}$$

On dit qu'un processus P valide une formule $\mathcal{A}$ sous la valuation v (noté $P \models_v \mathcal{A}$) si et seulement si $P \in [\![\mathcal{A}]\!]_v$.

Une formule $\mathcal{A}$ est dite valide (noté $\models \mathcal{A}$) si et seulement si elle est validée par tout processus sous toute valuation.

Remarque 2. On peut aussi prendre pour modèle de cette logique l'ensemble des unions finies d'intervalles de $\mathbb{N}$ (les processus du π-calcul ne deviennent indispensables que lorsque l'on introduit des opérateurs supplémentaires, comme en §6) :

$$\begin{array}{l}
[\![\mathbf{F}]\!]_v = \emptyset \\
[\![0]\!]_v = \{0\} \qquad (\textit{i. e.} \text{ le singleton constitué de l'entier zéro}) \\
[\![A_w]\!]_v = v(A_w) \\
[\![\mathcal{A} \wedge \mathcal{B}]\!]_v = [\![\mathcal{A}]\!]_v \cap [\![\mathcal{B}]\!]_v \\
[\![\mathcal{A} \Rightarrow \mathcal{B}]\!]_v = \left(\complement_{\mathbb{N}} [\![\mathcal{A}]\!]_v\right) \cup [\![\mathcal{B}]\!]_v \\
[\![\mathcal{A}|\mathcal{B}]\!]_v = \{n + m \quad / \quad n \in [\![\mathcal{A}]\!]_v \wedge m \in [\![\mathcal{B}]\!]_v\} \\
[\![\mathcal{A} \triangleright \mathcal{B}]\!]_v = \{n \in \mathbb{N} \quad / \quad \forall m \in [\![\mathcal{A}]\!]_v, n + m \in [\![\mathcal{B}]\!]_v\}
\end{array}$$

L'opérateur $\blacktriangleright$ (cf. définition 14) permet d'obtenir la soustraction :

$$[\![\mathcal{A} \blacktriangleright \mathcal{B}]\!]_v = \{m - n \quad / \quad n \in [\![\mathcal{A}]\!]_v \wedge m \in [\![\mathcal{B}]\!]_v\}$$

3.2 Taille, indiscernabilité

Nous allons définir une taille des processus et des formules. Puis nous allons rendre rigoureuse l'idée intuitive qu'une formule trop petite ne peut pas faire la différence entre deux processus trop grands c'est-à-dire que pour une formule $\mathcal{A}$ donnée, si P et Q sont deux processus de taille supérieure à un entier ne dépendant que de $\mathcal{A}$ alors soit P et Q valident tous les deux $\mathcal{A}$ soit aucun d'entre eux ne valide $\mathcal{A}$.

Définition 9 (Taille d'un processus). La taille des processus est définie inductivement par

$$\begin{array}{l}
\|0\| = 0 \\
\|1\| = 1 \\
\|P|Q\| = \|P\| + \|Q\|
\end{array}$$

Définition 10 (Taille d'une formule). La taille des formules, notée $\|\ \|$, est définie inductivement sur l'ensemble des formules par :

$$\begin{array}{l}
\|\mathbf{F}\| = 0 \\
\|0\| = 1 \\
\|A_w\| = w \\
\|\mathcal{A} \wedge \mathcal{B}\| = \|\mathcal{A} \Rightarrow \mathcal{B}\| = \|\mathcal{A} \triangleright \mathcal{B}\| = \max(\|\mathcal{A}\|, \|\mathcal{B}\|) \\
\|\mathcal{A}|\mathcal{B}\| = \|\mathcal{A}\| + \|\mathcal{B}\|
\end{array}$$

On remarque que la taille correspondant au connecteur $|$ est définie comme la somme des tailles de ses arguments et non pas comme le maximum, comme c'est le cas pour les autres connecteurs d'arité deux. Intuitivement, c'est en effet cet opérateur qui donne le pouvoir discriminant des formules spatiales.

Définition 11 (w-équivalence). Deux processus P et Q sont dit w-équivalents, ce que l'on note $P \sim_w Q$, si et seulement si :
- soit $\|P\| \geq w$ et $\|Q\| \geq w$;
- soit $P \equiv Q$.

Les résultats qui suivent concernant l'équivalence $\sim_w$ sont démontrés dans [CCG02].

Proposition 1 (Indiscernabilité). *Soient P et Q deux processus de $\mathcal{P}$ et $\mathcal{A}$ une formule. Si $P \sim_w Q$ et $\|\mathcal{A}\| \leq w$ alors P valide $\mathcal{A}$ si et seulement si Q valide $\mathcal{A}$.*

Démonstration. Par induction sur la structure de $\mathcal{A}$. □

Lemme 3. *Si $P \sim_w Q$ alors $P|R \sim_w Q|R$.*

Lemme 4. *Si $w' \leq w$ alors $P \sim_w Q \Rightarrow P \sim'_w Q$.*

Proposition 2 (Élagage). *Soit $P \in \mathcal{P}$ un processus. Alors il existe un processus $Q \in \mathcal{P}$ tel que $\|Q\| \leq w$ et $P \sim_w Q$. On note $P_{/w}$ un tel processus.*

Démonstration. Si $\|P\| \leq w$ alors $Q \equiv P$ convient. Sinon, $Q \equiv \underbrace{1|\ldots|1}_{n \text{ fois}}$ convient. □

Remarque 3. C'est pour pouvoir utiliser dans tous les cas l'équivalence $\sim_w$ que nous avons été amenés à introduire une taille maximale sur les variables propositionnelles (l'indice w de A_w).

3.3 Opérateurs dérivés, expressivité de la logique

D'autres opérateurs peuvent être dérivés à partir de ceux déjà définis. Il permettent de mettre en évidence certains aspects de l'expressivité de la logique.

Définition 12 (Opérateur $||$). L'opérateur dérivé $||$ est défini par

$$P \models \mathcal{A}||\mathcal{B} \quad \text{si et seulement si} \quad P \models \neg(\neg\mathcal{A}|\neg\mathcal{B})$$

Cet opérateur permet par exemple de fabriquer la formule $(\mathcal{A}||\mathbf{F})$, encore notée $\mathcal{A}^{\forall}$, qui n'est validée que par des processus dont toutes les composantes vérifient la formule $\mathcal{A}$.

Définition 13 (Formule caractéristique). La formule caractéristique $\chi(P)$ d'un processus P tel que $\|P\| = w$ est

$$\chi(P) = \underbrace{(0||\ldots||0)}_{w+1 \text{ fois}} \wedge \neg \underbrace{(0||\ldots||0)}_{w \text{ fois}}$$

La formule caractéristique $\chi(E)$ d'un ensemble E de processus est définie inductivement par

$$\begin{aligned} &\chi(\emptyset) = \mathbf{F} \\ &\chi(\{P\}) = \chi(P) \\ &\chi(E' \cup E'') = \chi(E') \oplus \chi(E'') \end{aligned}$$

avec l'opérateur $\oplus$ défini par $\mathcal{A} \oplus \mathcal{B} = (\mathcal{A} \vee \mathcal{B}) \wedge \neg(\mathcal{A} \wedge \mathcal{B})$.

Lemme 5. *On a*

$$Q \models \chi(P) \quad \textit{si et seulement si} \quad Q \equiv P$$

et

$$Q \models \chi(E) \quad \textit{si et seulement si} \quad \exists P \in E, Q \equiv P$$

Définition 14 (Opérateur $\blacktriangleright$). L'opérateur dérivé $\blacktriangleright$ est défini par

$$P \models \mathcal{A} \blacktriangleright \mathcal{B} \quad \text{si et seulement si} \quad P \models \neg(\mathcal{A} \triangleright \neg\mathcal{B})$$

La formule $\mathcal{A} \blacktriangleright \mathcal{B}$ est vérifiée par tout processus P tel qu'il existe un processus Q vérifiant $\mathcal{A}$ tel que le processus $P|Q$ vérifie $\mathcal{B}$.

Définition 15 (Opérateur $\sim$). L'opérateur dérivé $\sim$ est défini par

$$P \models \sim \mathcal{A} \quad \text{si et seulement si} \quad P \models \mathcal{A} \triangleright \mathbf{F}$$

Tentons d'un peu mieux appréhender l'expressivité de la logique introduite principalement par l'opérateur $\triangleright$ (et ses connecteurs dérivés). Par exemple, la formule

$$\neg(\sim \mathcal{A})$$

permet d'exprimer la propriété « il existe un modèle de la formule $\mathcal{A}$ ».

On peut encore créer une formule dans laquelle une variable propositionnelle A n'admettra que des modèles de taille paire :

$$\sim (A \Rightarrow A|A) \wedge \sim (\chi(2) \Rightarrow A)$$

où $\chi(2)$ désigne la formule caractéristique d'un processus de taille deux (*i. e.* $(0||0||0) \wedge \neg(0||0)$).

3.4 Séquents

Définition 16 (Congruence $\doteq$). La congruence $\doteq$, qui permet de définir les théories de contraintes, est la plus petite relation d'équivalence telle que

$$\begin{aligned} u|0 &\doteq 0 \\ u|v &\doteq v|u \\ u|(v|t) &\doteq (u|v)|t \end{aligned}$$

On utilise la notation $u \doteq_S v$ lorsque S est une théorie (un ensemble) de contraintes et $u \doteq v \in S$.

Définition 17 (Séquent). Un séquent est un triplet noté $\langle S \rangle \ \Gamma \vdash \Delta$, où S est une théorie de contraintes de la forme $u \doteq X_1|\ldots|X_n$ avec $u \in \mathcal{I}$ ($\mathcal{I}$ est l'ensemble des index) et $X_i \in \mathcal{Z}$ ($\mathcal{Z}$ est l'ensemble des variables de processus) et Γ et Δ sont des contextes (ensembles de couples de $\mathcal{I} \times \mathcal{F}$ notés $u : \mathcal{A}$).

Définition 18 (Interprétation). Une interprétation $\mathscr{I}$ est une fonction qui :
- à toute variable propositionnelle A associe un ensemble de propriété de $\mathbb{F}$;
- à toute variable de processus X associe un processus de $\mathcal{P}$.

Définition 19 (Validation d'une théorie de contraintes). On dit que $\mathscr{I}$ valide la théorie de contraintes S, ce que l'on note $\mathscr{I} \models \langle S \rangle$ si et seulement si $\mathscr{I}$ est prolongée en un $|$-morphisme sur les index et est telle que $u \doteq_S v$ si et seulement si $\mathscr{I}(u) \equiv \mathscr{I}(v)$.

On dit que $\mathscr{I}$ est une interprétation du séquent $\langle S \rangle \ \Gamma \vdash \Delta$ si et seulement si $\mathscr{I}$ valide $\langle S \rangle$.

Lemme 6 (Consistance). *Si S est une théorie de contraintes et $\mathscr{I}$ une interprétation satisfaisant S alors : si $u \doteq_S v$ alors $\mathscr{I}(u) \equiv \mathscr{I}(v)$.*

Définition 20 (Validation d'un séquent). On dit qu'une interprétation $\mathscr{I}$ du séquent $\langle S \rangle \ \Gamma \vdash \Delta$ le valide si, si $\mathscr{I}$ satisfait S et $\mathscr{I}$ satisfait tout Γ (*i. e.* pour tout $u : \mathcal{A}$ dans Γ on a $\mathscr{I}(u) \in [\![\mathcal{A}]\!]_{\mathscr{I}}$, ce que l'on note $\mathscr{I} \models_- \Gamma$) alors $\mathscr{I}$ satisfait partiellement Δ (*i. e.* il existe $u : \mathcal{A}$ dans Δ tel que $\mathscr{I}(u) \in [\![\mathcal{A}]\!]_{\mathscr{I}}$, ce que l'on note $\mathscr{I} \models_+ \Delta$).

Un séquent est dit valide si et seulement s'il est validé par toute interprétation.

Un système de preuve est déterminé par la donnée d'un ensemble de règles d'inférence qui servent à « construire » (*i. e.* dériver) des séquents.

Définition 21 (Séquent prouvable). Un séquent est dit prouvable dans un système de règles d'inférence si et seulement s'il est dérivable en utilisant les règles d'inférence de ce système.

4 Trois systèmes de séquents

Notre prouveur a pour rôle de tenter de « remonter » dans les règles des séquents jusqu'à trouver une dérivation valide. Il part d'un séquent et recherche tous les séquents antécédents possibles pour toutes les règles. Pour que la terminaison d'un tel processus soit assurée, il faut que le système de règles soit, en plus d'être correct, complet *i. e.* que la recherche effectuée par notre programme termine et que cette recherche aboutisse pour tout séquent valide.

Nous avons donc été amenés à étudier trois systèmes de séquents ; le dernier – que nous avons utilisé dans l'implémentation – étant correct et complet.

4.1 Système $\mathcal{S}_0$

Définition 22 (Système $\mathcal{S}_0$). Le système S_0 de règles d'inférence est un sous-ensemble de celui décrit dans [CC03]. Ses règles d'inférence sont :

$$\frac{u \doteq_S v}{\langle S\rangle\ \Gamma, u : A \vdash v : A, \Delta}\,(\mathrm{Id})$$

$$\frac{}{\langle S\rangle\ \Gamma, u : \mathbf{F} \vdash \Delta}\,(\mathbf{F}\mathrm{L}) \qquad \frac{\langle S\rangle\ \Gamma \vdash \Delta}{\langle S\rangle\ \Gamma \vdash u : \mathbf{F}, \Delta}\,(\mathbf{F}\mathrm{R})$$

$$\frac{\langle S\rangle\ \Gamma, u : \mathcal{A}, u : \mathcal{B} \vdash \Delta}{\langle S\rangle\ \Gamma, u : \mathcal{A} \wedge \mathcal{B} \vdash \Delta}\,(\wedge\mathrm{L}) \qquad \frac{\langle S\rangle\ \Gamma \vdash u : \mathcal{A}, \Delta \quad \langle S\rangle\ \Gamma \vdash u : \mathcal{B}, \Delta}{\langle S\rangle\ \Gamma \vdash u : \mathcal{A} \wedge \mathcal{B}, \Delta}\,(\wedge\mathrm{R})$$

$$\frac{\langle S\rangle\ \Gamma \vdash u : \mathcal{A}, \Delta \quad \langle S\rangle\ \Gamma, u : \mathcal{B} \vdash \Delta}{\langle S\rangle\ \Gamma, u : \mathcal{A} \Rightarrow \mathcal{B} \vdash \Delta}\,(\Rightarrow\mathrm{L}) \qquad \frac{\langle S\rangle\ \Gamma, u : \mathcal{A} \vdash u : \mathcal{B}, \Delta}{\langle S\rangle\ \Gamma \vdash u : \mathcal{A} \Rightarrow \mathcal{B}, \Delta}\,(\Rightarrow\mathrm{R})$$

$$\frac{\langle S, u \doteq 0\rangle\ \Gamma \vdash \Delta}{\langle S\rangle\ \Gamma, u : 0 \vdash \Delta}\,(0\mathrm{L}) \qquad \frac{u \doteq_S 0}{\langle S\rangle\ \Gamma \vdash u : 0, \Delta}\,(0\mathrm{R})$$

$$\frac{\langle S\rangle\ \Gamma, u : \mathcal{A}, u : \mathcal{A} \vdash \Delta}{\langle S\rangle\ \Gamma, u : \mathcal{A} \vdash \Delta}\,(\mathrm{CL}) \qquad \frac{\langle S\rangle\ \Gamma \vdash u : \mathcal{A}, u : \mathcal{A}, \Delta}{\langle S\rangle\ \Gamma \vdash u : \mathcal{A}, \Delta}\,(\mathrm{CR})$$

$$[\mathcal{X} \textit{ et } \mathcal{Y} \textit{ non libres dans la conclusion}]\quad \frac{\langle S, u \doteq \mathcal{X}|\mathcal{Y}\rangle\ \Gamma, \mathcal{X} : \mathcal{A}, \mathcal{Y} : \mathcal{B} \vdash \Delta}{\langle S\rangle\ \Gamma, u : \mathcal{A}|\mathcal{B} \vdash \Delta}\,(|\mathrm{L}) \qquad \frac{\langle S\rangle\ \Gamma \vdash v : \mathcal{A}, \Delta \quad \langle S\rangle\ \Gamma \vdash t : \mathcal{B}, \Delta \quad u \doteq_S v|t}{\langle S\rangle\ \Gamma \vdash u : \mathcal{A}|\mathcal{B}, \Delta}\,(|\mathrm{R})$$

$$\frac{\langle S\rangle\ \Gamma \vdash t : \mathcal{A}, \Delta \quad \langle S\rangle\ \Gamma, v : \mathcal{B} \vdash \Delta \quad v \doteq_S t|u}{\langle S\rangle\ \Gamma, u : \mathcal{A} \triangleright \mathcal{B} \vdash \Delta}\,(\triangleright\mathrm{L}) \qquad [\mathcal{X} \textit{ non libre dans la conclusion}]\quad \frac{\langle S\rangle\ \Gamma, \mathcal{X} : \mathcal{A} \vdash v : \mathcal{B}, \Delta \quad v \doteq_S \mathcal{X}|u}{\langle S\rangle\ \Gamma \vdash u : \mathcal{A} \triangleright \mathcal{B}, \Delta}\,(\triangleright\mathrm{R})$$

Les règles des opérateurs $\mathbf{F}$, $\wedge$, $\Rightarrow$ sont les mêmes qu'en logique propositionnelle classique nous n'y reviendrons donc pas.

La règle (|R) peut se lire : « si je sais montrer qu'une partie v de u vérifie $\mathcal{A}$ et que l'autre partie t vérifie $\mathcal{B}$ alors je sais montrer que u vérifie $\mathcal{A}|\mathcal{B}$ ». Les règles à gauche, quant à elles, peuvent se lire comme des négations (la règle dérivée pour $\neg$ qui est exposée en §B.2 permet de mieux comprendre pourquoi). Ainsi la règle ($\triangleright$L) peut se lire : « si je sais exhiber un index t qui vérifie $\mathcal{A}$ et est tel que $t|u$ ne vérifie pas $\mathcal{B}$ alors je sais nier le fait que u vérifie $\mathcal{A} \triangleright \mathcal{B}$ ». Les règles (|R) et ($\triangleright$L) représentent des quantifications existentielles ($u : \mathcal{A}|\mathcal{B}$ peut se lire : « il existe une décomposition de u en v et t telle que $v : \mathcal{A}$ et $t : \mathcal{B}$ ») alors que les règles (|L) et ($\triangleright$R) représentent des quantifications universelles ce qui explique la nécessité d'avoir recours à des variables fraîches ($\mathcal{X}$ et $\mathcal{Y}$). En effet, par exemple pour la règle (|L), on veut que toute décomposition de u soit de la forme $v|t$ avec $v \models \mathcal{A}$ et $t \models \mathcal{B}$. L'introduction de

variables fraîches permet de s'assurer que toutes les décompositions seront prises en compte ce qui pourrait ne pas être le cas si l'on avait utilisé des variables existantes qui auraient pu vérifier des propriétés particulières.

Les théories de contraintes permettent, comme nous l'avons déjà mentionné p. 4, de stocker la structure des processus. Cela est effectivement rendu possible grâce aux règles (0L) et (|L) qui permettent de faire passer les informations structurelles sur les index, des ensembles Γ ou Δ vers la théorie de contraintes S ; et vice-versa grâce aux règles (0R) et (|R). Ainsi par exemple la règle (|L) permet, si un index u doit vérifier la formule $\mathcal{A}|\mathcal{B}$, de rajouter dans la théorie de contraintes le fait que u est formé de deux composantes.

Enfin, les règles (CL) et (CR) posent problème car elles rendent la recherche de preuves clairement non terminante donc non complète.

4.2 Système $\mathcal{S}_1$

Le système $\mathcal{S}_0$ est celui qui découle immédiatement de la logique et nous nous dirigeons vers des séquents qui nous permettent de décrire un algorithme. Pour pouvoir trouver un algorithme qui recherche des preuves, nous avons cherché à établir un système de règles de séquents qui soit correct et complet. Nous y sommes parvenus avec le système $\mathcal{S}_2$ présenté plus loin. Mais avant d'arriver à le mettre au point, nous avons étudié le système $\mathcal{S}_1$ – qui est encore une fois simplement correct (la preuve est donnée dans l'annexe A) – qui constitue une sorte de « transition » entre le système $\mathcal{S}_0$ et le système $\mathcal{S}_2$. Voici les règles de $\mathcal{S}_1$ qui diffèrent de celles $\mathcal{S}_0$:

$$\frac{\langle S, u \doteq v|t, v \doteq X_1'|\ldots|X_n', t \doteq X_1''|\ldots|X_n'', X_1 \doteq X_1'|X_1'', \ldots, X_n \doteq X_n'|X_n''\rangle \;\; \Gamma, v : \mathcal{A}, t : \mathcal{B} \vdash \Delta}{\langle S, u \doteq X_1|\ldots|X_n\rangle \;\; \Gamma, u : \mathcal{A}|\mathcal{B} \vdash \Delta} \;(|\mathrm{L})$$

$$\frac{\langle S\rangle \;\; \Gamma \vdash v : \mathcal{A}, u : \mathcal{A}|\mathcal{B}, \Delta \quad \langle S\rangle \;\; \Gamma \vdash t : \mathcal{B}, u : \mathcal{A}|\mathcal{B}, \Delta \quad u \doteq_S \overline{v}^S|\overline{t}^S}{\langle S\rangle \;\; \Gamma \vdash u : \mathcal{A}|\mathcal{B}, \Delta} \;(|\mathrm{R})$$

$$\frac{\langle S\rangle \;\; \Gamma, u : \mathcal{A} \triangleright \mathcal{B} \vdash X : \mathcal{A}, \Delta \quad \langle S\rangle \;\; \Gamma, X|u : \mathcal{B}, u : \mathcal{A} \triangleright \mathcal{B} \vdash \Delta}{\langle S\rangle \;\; \Gamma, u : \mathcal{A} \triangleright \mathcal{B} \vdash \Delta} \;(\triangleright\mathrm{L})$$

$$\frac{\begin{array}{c}[\mathcal{X} \textit{ non libre dans la conclusion}] \\ \langle S\rangle \;\; \Gamma, \mathcal{X} : \mathcal{A} \vdash v : \mathcal{B}, \Delta \quad v \doteq_S \mathcal{X}|\overline{u}^S\end{array}}{\langle S\rangle \;\; \Gamma \vdash u : \mathcal{A} \triangleright \mathcal{B}, \Delta} \;(\triangleright\mathrm{R})$$

Les règles (CL) et (CR) ont été enlevées. On peut en effet montrer qu'on peut restreindre leur utilisation à avant une règle (|R) ou ($\triangleright$L) [4] ; nous les avons donc intégrées aux nouvelles règles (|R) et ($\triangleright$L) – qui, à leur tour, rendent le système clairement non terminant. Enfin, la règle ($\triangleright$L) a été modifiée : nous cherchons maintenant les découpages possibles de u uniquement parmi le découpage actuel de u (via la contrainte à laquelle il est soumis) en raffinant celui-ci.

4.3 Système $\mathcal{S}_2$

4.3.1 Définitions préliminaires

Nous avons vu (§3.2) que pour une formule donnée, à partir d'une certaine taille les processus sont indiscernable pour la formule. La taille d'une formule $\mathcal{A}$ à prouver va donc nous permettre de déterminer avec quelle « granularité » nous allons pouvoir étudier les processus, limitant ainsi le nombre de composantes distinguables dans les processus que nous allons être amenés à étudier, et ce en préservant la validité des séquents manipulés.

[4] La nécessité d'utiliser des règles de contraction est similaire au comportement de la quantification existentielle en logique propositionnelle (cf. par exemple [Gal87]) c'est pourquoi elle ne s'applique qu'aux règles (|R) et ($\triangleright$L).

Notation 1. On note $\widehat{u}^S$ l'ensemble

$$\{(X_1|\dots|X_k, X_{k+1}|\dots|X_n) \quad / \quad u \doteq_S X_1|\dots|X_n\}$$

(attention, les permutations des X_i sont implicitement prises en compte dans la précédente définition).

Intuitivement $\widehat{u}^S$ désigne l'ensemble des façons de couper en deux le processus désigné par u.

Définition 23 (Imposition). On dit que la théorie de contraintes S impose partiellement l'ensemble de conditions E, ce que l'on note $S \Vdash E$ si et seulement si pour toute interprétation $\mathscr{I}$ validant S, $\mathscr{I}$ valide au moins une des conditions de E.

Notation 2 ($\between$). Le symbole $\between$ ne fait pas partie de la logique. La notation $u \between v : \mathcal{A} \between \mathcal{B}$ indique que l'on a $u \models \mathcal{A}$ et $v \models \mathcal{B}$ (la définition formelle est donnée par la règle ($|\mathrm{R}'$)). Les règles ($|\mathrm{R}$) et ($|\mathrm{R}'$) – dont la définition est donnée ci-dessous – ne forment en réalité qu'une seule et unique règle mais cette notation permet d'éviter d'introduire une quantification universelle au niveau méta sur les découpages de u dans les prémisses de la règle.

Notation 3 ($\widetilde{Z}$). Soit $\mathcal{S} = \langle S \rangle \ \Gamma \vdash \Delta$ un séquent. La notation $\mathcal{S} = \langle S \rangle \ \Gamma \vdash \Delta, \widetilde{Z}$ suppose que $\widetilde{Z}$ ne contient que des contraintes de la forme $u : 0$ et que l'on a $\forall u : \mathcal{A} \in \Gamma \cup \Delta, \mathcal{A} \neq 0$ (ce qui revient à supposer que l'on ne peut plus appliquer la règle (0L)).

4.3.2 Règles de séquents

Définition 24 (Système S_2). Il est identique à S_0 sauf :

$$\frac{\begin{array}{c}[\sigma \text{ est la subsitution telle que } \forall i \in [\![1,n]\!], \sigma(X_i) = X_i^1|X_i^2] \\ \left\langle \sigma(S), v \doteq X_1^1|\dots|X_n^1, t \doteq X_1^2|\dots|X_n^2 \right\rangle \ \Gamma, v : \mathcal{A}, t : \mathcal{B} \vdash \Delta \quad u \doteq_S X_1|\dots|X_n\end{array}}{\langle S \rangle \ \Gamma, u : \mathcal{A}|\mathcal{B} \vdash \Delta} \ (|\mathrm{L})$$

$$\frac{\langle S \rangle \ \Gamma \vdash v : \mathcal{A}, \Delta \quad \langle S \rangle \ \Gamma \vdash t : \mathcal{B}, \Delta}{\langle S \rangle \ \Gamma \vdash v \between t : \mathcal{A} \between \mathcal{B}, \Delta} \ (|\mathrm{R}')$$

$$\frac{\left\langle S, v_i \doteq \overline{v}_i^S, t_i \doteq \overline{t}_i^S \right\rangle \ \Gamma \vdash v_1 \between t_1 : \mathcal{A} \between \mathcal{B}, \dots, v_n \between t_n : \mathcal{A} \between \mathcal{B}, \Delta \quad \text{avec } \{(v_i, t_i) \quad / \quad i \in [\![1,n]\!]\} = \widehat{u}^S}{\langle S \rangle \ \Gamma \vdash u : \mathcal{A}|\mathcal{B}, \Delta} \ (|\mathrm{R})$$

$$\frac{\begin{array}{c}[Avec\ \mathcal{A} \in \mathcal{F}_0,\ n = \|\mathcal{A} \triangleright \mathcal{B}\| \text{ et } J \subseteq [\![1,n]\!]] \\ \begin{array}{ll}\text{Pour tout } j \in J & \langle S, t_1 \doteq Y_1, \dots, t_j \doteq Y_1|\dots|Y_j, y_1 \doteq Y_1, \dots, y_j \doteq Y_j \rangle \ \Gamma \vdash t_j : \mathcal{A}, y_1 : 0, \dots, y_j : 0, \Delta \\ \text{Pour tout } j \in [\![1,n]\!] \setminus J & \langle S, t_1 \doteq Y_1, \dots, t_j \doteq Y_1|\dots|Y_j, y_1 \doteq Y_1, \dots, y_j \doteq Y_j \rangle \ \Gamma, t_j : \mathcal{A} \vdash y_1 : 0, \dots, y_j : 0, \Delta \\ & \left\langle S, \underbrace{v_j \doteq u|t_j, t_j \doteq Y_1|\dots|Y_j}_{\text{pour tout } j \in J}, y_1 \doteq Y_1, \dots, y_n \doteq Y_n \right\rangle \ \Gamma, \underbrace{v_j : \mathcal{B}}_{j \in J} \vdash y_1 : 0, \dots, y_n : 0, \Delta\end{array}\end{array}}{\langle S \rangle \ \Gamma, u : \mathcal{A} \triangleright \mathcal{B} \vdash \Delta} \ (\triangleright\mathrm{L})$$

$$\frac{\left\langle S, v \doteq_S \overline{t}^S|\overline{u}^S, t = Y_1|\dots|Y_{\|\mathcal{A}\triangleright\mathcal{B}\|} \right\rangle \ \Gamma, t : \mathcal{A} \vdash v : \mathcal{B}, \Delta}{\langle S \rangle \ \Gamma \vdash u : \mathcal{A} \triangleright \mathcal{B}, \Delta} \ (\triangleright\mathrm{R})$$

$$\frac{S \Vdash u_1 \sim_{\|\mathcal{A}_1\|} v_1, \dots, u_n \sim_{\|\mathcal{A}_n\|} v_n, Z \quad \text{avec } \{(u_i : \mathcal{A}_i, v_i : \mathcal{A}_i) \quad / \quad i \in [\![1,n]\!]\} \in \Gamma \times \Delta}{\langle S \rangle \ \Gamma \vdash \Delta, \widetilde{Z}} \ (\widetilde{\mathrm{Id}})$$

Les règles (CL) et (CR) ont de plus été retirées.

Dans ce dernier système, la règle ($|\mathrm{R}$) n'est plus clairement non terminante. En effet, on ne recherche cette fois les candidats pour prouver la formule $\mathcal{A}|\mathcal{B}$ que parmi les découpages de u

i. e. les sections en deux de la contrainte actuelle de u. La règle (|L) n'a pas fondamentalement changé par rapport au système $\mathcal{S}_0$, nous avons simplement adopté une notation plus proche de l'implémentation effective. Il est à noter que ces deux règles (c'est surtout visible sur (|R)) dépendent de la contrainte déjà présente sur u en ce qui concerne la correction et la complétude. C'est en ce sens que les contraintes déterminent la « granularité » (la finesse) avec laquelle nous allons pouvoir observer les processus. Ainsi, intuitivement, si u représente un processus qui contient au moins cinq composantes non nulles en parallèle et que S contient la contrainte $u \doteq_S X_1|X_2|X_3$ alors la règle (|R) ne pourra pas tester tous les découpages du processus désigné par u et ne sera plus correcte ; le découpage de u en trois n'est alors, en ce sens, pas assez « fin ». De même, pour la règle (|L), si le découpage n'est pas assez fin, il se peut que nous passions à côté d'un découpage du processus désigné par u qui nous aurait permis de montrer que u ne valide pas $\mathcal{A}|\mathcal{B}$, mettant ainsi à mal la complétude du système. C'est pourquoi nous avons été amenés à introduire une notion de taille des formules qui correspond intuitivement au nombre de composantes que la formule va être à même de distinguer ; et celle-ci nous permet de déterminer la granularité initiale à introduire au niveau des contraintes. De plus cette taille doit intervenir dans la formulation des règles.

Le même type de remarque s'applique aussi aux règles (▷L) et (▷R). La règle (▷L) peut s'expliquer de la façon suivante : les t_j sont des représentants d'une partition des processus selon les classes d'équivalence de $\sim_{\|A \triangleright B\|}$. En effet, les contraintes de la forme $y_i : 0$ à droite, « imposent » aux interprétations des formules d'interpréter les Y_i par 1 (le cas où l'un des Y_i est interprété par 0 est trivial). L'ensemble (fini) des t_j représentent l'ensemble des processus, modulo la relation d'équivalence $\sim_{\|A \triangleright B\|}$ et l'ensemble J est imposé par les deux premiers séquents antécédents comme l'ensemble $J = \{j \quad / \quad t_j \models \mathcal{A}\}$. Le troisième séquent antécédent nous permet de nous assurer que $\forall j \in J, t_j|u \models \mathcal{B}$; ce qui revient à écrire, sachant que les $(t_j)_{j\in J}$ représentent l'ensemble des processus, que pour toute interprétation $\mathscr{I}$ du séquent conséquent on a $\forall P \in \mathcal{P}, P \models \mathcal{A} \Rightarrow P|\mathscr{I}(u) \models \mathcal{B}$. On a alors bien $u \models \mathcal{A} \triangleright \mathcal{B}$, et nous aurons montré cela en n'utilisant qu'un nombre fini de tests. La règle (▷R) est presque une transcription exacte de la règle (▷R) du système $\mathcal{S}_1$ à ceci près que l'on impose la granularité $\|\mathcal{A} \triangleright \mathcal{B}\|$, que l'on sait être suffisante, à l'index de test t.

Enfin, la règle ($\widetilde{\text{Id}}$) est nécessaire pour pouvoir traiter des formules comme $(1 \wedge A)|(1 \wedge \neg A) \Rightarrow \mathbf{F}$ (où 1 désigne la formule $\chi(1)$) qui amène à prouver le séquent (qui est valide)

$$\langle u = X_1, v = X_2 \rangle \;\; u : A \vdash v : A, u : 0, v : 0$$

La validité de ce séquent provient du fait que les index u et v ont une granularité de un et donc ne peuvent être bool-interprétés que par 1 ou 0. Dans tous les cas, on vérifie que l'interprétation valide bien le séquent. À utiliser en dernier recours, la règle ($\widetilde{\text{Id}}$) cache derrière l'opérateur $\Vdash$ une phase de *model-checking* nécessaire pour traiter de tels cas.

4.3.3 Propriétés du système de séquents

Les démonstrations de correction et de complétude de ce système se trouvent en annexe (§B.3).

Définition 25 (Bool-interprétation). Une interprétation booléenne $\mathscr{I}$ est une fonction qui :
- à toute variable propositionnelle A associe une partie de l'ensemble $\{0, 1\}$;
- à toute variable de processus X associe soit 0, soit 1.

Remarque 4. 0 et 1 ne sont pas ici des entiers mais bien des processus.

Définition 26 (Séquent équivalidable). Un séquent est dit *équivalidable* si et seulement s'il est de la forme

$$\left\langle u_1 \doteq X_1^1|\ldots|X_{p_1}^1, \ldots, u_n \doteq X_1^n|\ldots|X_{p_n}^n \right\rangle \;\; u_1 : \mathcal{A}_1, \ldots, u_k : \mathcal{A}_k \vdash u_{k+1} : \mathcal{A}_{k+1}, \ldots, u_n : \mathcal{A}_n$$

où les X_i^j vérifient

$$\forall (j_1, j_2) \in (\llbracket 1, n \rrbracket)^2, \forall i_1 \in \llbracket 1, p_{j_1} \rrbracket, \forall i_2 \in \llbracket 1, p_{j_2} \rrbracket, i_1 \neq i_2 \wedge j_1 \neq j_2 \Rightarrow X_{i_1}^{j_1} \neq X_{i_2}^{j_2}$$

et les p_i vérifient

$$\forall i \in \llbracket 1, n \rrbracket, p_i \leq \max \{ \|\mathcal{A}\| \quad / \quad u_i : \mathcal{A} \in \Gamma \cup \Delta \}$$

Proposition 3. *Un séquent équivalidable est valide si et seulement s'il est bool-valide (i. e. est validé par toute interprétation booléenne).*

Démonstration. Une bool-interprétation est en particulier une interprétation donc tout séquent valide est bool-valide.

Réciproquement, soit $\mathcal{S}$ un séquent équivalidable bool-valide de la même forme que dans la définition précédente. Montrons qu'il est valide. Soit $\mathscr{I}$ une interprétation telle que $\mathscr{I} \models \langle S \rangle$ et $\mathscr{I} \models_- \Gamma$. Montrons que $\mathscr{I} \models_+ \Delta$. Soit $\mathscr{I}'$ la bool-interprétation définie pour tout j dans $\llbracket 1, n \rrbracket$ par

$$\mathscr{I}'(X_i^j) = \begin{cases} 1 & \text{si } i \leq \|\mathscr{I}(u_j)\| \\ 0 & \text{sinon} \end{cases}$$

On a alors $\mathscr{I}'(u_j) \sim_m \mathscr{I}(u_j)$ avec $m = \max \{ \|\mathcal{A}\| \quad / \quad u : \mathcal{A} \in \Gamma \cup \Delta \}$. Comme $\mathscr{I} \models \langle S \rangle$ et $\mathscr{I} \models_- \Gamma$, on a aussi, d'après la proposition 1, $\mathscr{I}' \models \langle S \rangle$ et $\mathscr{I}' \models_- \Gamma$. Le séquent $\mathcal{S}$ étant supposé bool-validable, on a donc $\mathscr{I}' \models_+ \Delta$. Donc $\mathscr{I} \models_+ \Delta$. Le séquent $\mathcal{S}$ est donc valide. □

Proposition 4 (Bool-correction de $\mathcal{S}_2$). *Le système $\mathcal{S}_2$ est bool-correct i. e. toutes ses règles préservent la bool-validité et ses axiomes sont bool-valides.*

Démonstration. La démonstration complète est proposée en annexe (§B.3.1). □

Théorème 1 (Correction). *Si le sequent $\langle u \doteq X_1 | \ldots | X_{\|\mathcal{A}\|} \rangle \vdash u : \mathcal{A}$ est prouvable alors il est valide.*

Démonstration. C'est un cas particulier de la proposition précédente. □

Proposition 5 (Bool-complétude de $\mathcal{S}_2$). *Le système $\mathcal{S}_2$ est bool-complet i. e. si un séquent est bool-valide alors il est prouvable dans $\mathcal{S}_2$.*

Démonstration. La démonstration complète est proposée en annexe (§B.3.2). □

Théorème 2 (Complétude). *Si le sequent $\langle u \doteq X_1 | \ldots | X_{\|\mathcal{A}\|} \rangle \vdash u : \mathcal{A}$ est valide alors il est prouvable dans $\mathcal{S}_2$.*

Démonstration. C'est un cas particulier de la proposition précédente. □

Remarque 5 *($\mathcal{A} \notin \mathcal{F}_0$ pour ($\triangleright$L)).* En réalité la condition $\mathcal{A} \in \mathcal{F}_0$ pour la règle du ($\triangleright$L) n'est restrictive qu'en pratique (et pas théoriquement). On peut en effet montrer que le séquent antécédent de la règle ($\triangleright$L) est valide pour $\mathcal{A}$ formule non close si et seulement s'il l'est pour toute formule close $\mathcal{A}'$ avec $\mathcal{A}'$ obtenue en remplaçant dans $\mathcal{A}$ chaque variable propositionnelle A_w par une formule de $\mathbb{F}_w$.

Pour w entier fixé, l'ensemble $\mathbb{F}_w$ est fini donc le nombre de tests à effectuer est fini – mais grand.

Théorème 3 (Correction et complétude). *Le sequent $\langle u \doteq X_1 | \ldots | X_{\|\mathcal{A}\|} \rangle \vdash u : \mathcal{A}$ est prouvable dans le système $\mathcal{S}_2$ si et seulement s'il est valide.*

Corollaire 1. *Si un séquent est prouvable dans $\mathcal{S}_0$ alors il l'est dans $\mathcal{S}_2$.*

Démonstration. Si un séquent $\mathcal{S}$ prouvable dans $\mathcal{S}_0$ alors il est valide. Le système $\mathcal{S}_2$ étant complet, $\mathcal{S}$ est prouvable dans $\mathcal{S}_2$. □

5 Implémentation de l'inférence

Nous avons implémenté un prouveur dans le système $\mathcal{S}_2$. Pour prouver une formule $\mathcal{A}$, on essaye de montrer la validité du séquent $\langle u \doteq X_1 | \ldots | X_{\|\mathcal{A}\|} \rangle \vdash u : \mathcal{A}$. Notre prouveur est correct et complet d'après le théorème 3.

5.1 Organisation générale

L'architecture logicielle a été conçue pour être, autant que possible, modulaire. Nous nous sommes efforcés pour ce faire d'utiliser des modules déclarés dans des fichiers séparés, afin de maximiser modularité et la clarté du code. Voici les pricipaux modules :

- Le module `Varset` permet de définir un ensemble de variables (dont le type sera `t` et le type des variables sera `var`). Il permet en particulier de générer des variables fraîches à l'intérieur de cet ensemble. En pratique (tout ceci est masqué par des types abstraits), les variables sont des entiers. À chaque demande de variable fraîche, le module incrémente un compteur interne.
- Le module `Logic` contient les fonctions relatives à la logique. Il contient en particulier la définition abstraite de ce qu'est une formule, une fonction qui permet de savoir si deux formules sont identiques et une fonction permettant de calculer la taille d'une formule.
- Les modules `Lexer` et `Parser` – vous l'aviez déjà deviné – contiennent les fonctions pemettant de faire l'analyse lexicale et syntaxique de l'entrée saisie par l'utilisateur. Il sont automatiquement générés par les outils OCamlLex et OCamlYacc.
- Le module `Constraintset` permet de définir et manipuler des théories de contraintes (les « S » des séquents). Il définit le type `t` des théories de contraintes. Celles-ci contiennent leurs propres ensembles d'index et de variables de processus de type `Varset.t`. Le module définit aussi le type des contraintes :

  ```
  type cstr = Cvoid | Cpar of cpar | Cpv of proc_var
  ```

  Le type `cpar` est en réalité égal au type `int` (cf. lemme 7 pour les explications). Enfin sont définies les fonctions permettant de manipuler les théories de contraintes : ajout, modification ou suppression de contraintes, itérateurs sur l'ensemble des contraintes ou sur les découpages d'un index, « découpage » en deux de chacune des variables de processus apparaissant dans la contrainte d'un index.
- Le module `Output` gère les sorties en mode texte et en LaTeX.
- Le module `Prover` contient l'implémentation des règles d'inférence du système $\mathcal{S}_2$. La fonction `check` les teste successivement pour voir si elles sont applicables au séquent à prouver et ce, dans un ordre aisément redéfinissable. À titre d'exemple, voici l'implémentation de la règle ($\Rightarrow$R) ; on peut constater que c'est presque une copie directe de la règle d'inférence correspondante (modulo les conventions sytaxiques de OCaml) :

  ```
  let check_imp_r (s, g, d) =
    list_iter_hole
      (fun k (u, a) l ->
         match a with
           | Imp (a, b) ->
               let a = (s, (u, a) :: g, (u, b) :: k @ l) in
               let r = check a in
                 if r <> Not_proved then raise (Proved_with (Imp_r (a, r)))
           | _ -> ()
      ) d;
    Not_proved
  ```
- Le module `Seqproof` maintient la boucle d'interaction pour l'utilisateur, lance l'analyseur syntaxique lorsque l'utilisateur entre une formule, puis tente de la prouver avec `Prover.check`.

Le prouveur peut être lancé avec l'option `-f` auquel cas il ne cherche que des preuves sans contraction ; les preuves qui ne nécessitent pas de contraction sont alors trouvées plus

rapidement – et des preuves ainsi trouvées sont correctes – mais le prouveur n'est plus complet *i. e.* un séquent valide peut ne pas être prouvé.

La commande `dvi` permet d'afficher la dérivation (l'arbre de preuve) ayant permis de prouver une formule en cas de succès (cf. exemple plus loin).

5.2 Quelques détails concernant l'implémentation

5.2.1 Codage des contraintes

Lemme 7 (Forme des contraintes). *Les contraintes sont toujours le la forme* $u \doteq X_1^{\alpha_1} | \ldots | X_n^{\alpha_n}$ *avec* $\forall i \in [\![1, n]\!], \alpha_i \in \{0, 1\}$.

Démonstration. Par induction simple sur la stucture de la preuve. □

Ceci justifie l'utilisation d'entiers (`int`) pour coder les contraintes dans lesquels la valeur du i-ème bit indique si la variable de processus X_i fait partie de la contrainte. Ce codage a l'inconvénient de limter le nombre de variables de processus disponibles à 30 (ou 64 suivant le type d'entier utilisé) – ce qui est en pratique largement suffisant – mais se manipule très facilement et surtout rapidement.

5.2.2 Connecteurs dérivés

En plus des connecteurs déjà présentés, des connecteurs dérivés, ainsi que les règles correspondantes, ont été implémentés :

- le connecteur $\neg$ défini par : $P \models \neg\mathcal{A}$ si et seulement si $P \models \mathcal{A} \Rightarrow \mathbf{F}$;
- le connecteur $\top$ défini par : $P \models \top$ si et seulement si $P \models \neg\mathbf{F}$;
- le connecteur $\vee$ défini par : $P \models \mathcal{A} \vee \mathcal{B}$ si et seulement si $P \models \neg(\neg\mathcal{A} \wedge \neg\mathcal{B})$;
- le connecteur $||$ (cf. définition 12).

Le lecteur intéressé par les règles dérivées correspondant à ces nouveaux connecteurs pourra les consulter en annexe en §B.2.

5.3 Exemples de formules prouvées

Nous donnons ici quelques exemples de formules qui ont été prouvées par notre programme (sans contraction).

Voici d'abord quelques propriétés montrant que la congruence structurelle sur les processus se répercute au niveau des formules (la mise en parallèle avec un processus vide et la commutativité et l'associativité de l'opérateur | en particulier) :

```
Welcome to SeqProof 0.3 by Samuel Mimram

# A | 0 => A
<> :- u : (A | 0) => A
* Proved with 4 proof attempts.
# A => A | 0
<> :- u : A => (A | 0)
* Proved with 10 proof attempts.
# A | (B | C) => (A | B) | C
<> :- u : (A | (B | C)) => ((A | B) | C)
* Proved with 6238 proof attempts.
# A | B => B | A
<> :- u : (A | B) => (B | A)
* Proved with 25 proof attempts.
```

Les *proof attempts* indiquent le nombre de séquents dont le programme a cherché à montrer la validité.

Voici ensuite quelques formules simples mettant en jeu l'opérateur $\triangleright$ (représenté par `|>`) :

```
# 0 |> A => A
<> :- u : (0 |> A) => A
* Proved with 4 proof attempts.
# A => (0 |> A)
<> :- u : A => (0 |> A)
* Proved with 4 proof attempts.
# A |> A | T
<> :- u : A |> (A | T)
* Proved with 8 proof attempts.
# 0 => A |> A
<> :- u : 0 => (A |> A)
* Proved with 4 proof attempts.
# B => A |> A | B
<> :- u : B => (A |> (A | B))
* Proved with 29 proof attempts.
# F |> A
<> :- u : F |> A
* Proved with 2 proof attempts.
```

On peut forcer les interprétations de u à être égales au processus vide par la formule 0 :

```
# 0 ^ A ^ B => A | B
<> :- u : ((0 ^ A) ^ B) => (A | B)
* Proved with 7 proof attempts.
```

Les règles dérivées pour l'opérateur || conservent bien la sémantique de cet opérateur (le symbole « ‘ » représente l'opérateur $\neg$) :

```
# 0 || 0 => ‘(‘0 | ‘0)
<> :- u : (0 || 0) => ‘(‘0 | ‘0)
* Proved with 176 proof attempts.
# ‘(‘0 | ‘0) => (0 || 0)
<> :- u : ‘(‘0 | ‘0) => (0 || 0)
* Proved with 31 proof attempts.
```

Comme indiqué plus haut, notre programme peut fournir les dérivations utilisées pour prouver les formules. Voici par exemple celle de $B \Rightarrow A \triangleright A|B$:

$$\dfrac{\dfrac{\dfrac{}{\langle u_0 = X_0|X_1, u_1 = X_2|X_3, u_2 = X_0|X_1|X_2|X_3, u_3 = X_2|X_3\rangle \quad u_1 : A, u_0 : B \vdash u_3 : A}\,(\mathrm{Id}) \qquad \dfrac{}{\langle u_0 = X_0|X_1, u_1 = X_2|X_3, u_2 = X_0|X_1|X_2|X_3, u_3 = X_0|X_1\rangle \quad u_1 : A, u_0 : B \vdash u_3 : B}\,(\mathrm{Id})}{\langle u_0 = X_0|X_1, u_1 = X_2|X_3, u_2 = X_0|X_1|X_2|X_3\rangle \quad u_1 : A, u_0 : B \vdash u_2 : A|B}\,(|\mathrm{R})}{\langle u_0 = X_0|X_1\rangle \quad u_0 : B \vdash u_0 : A \triangleright (A|B)}\,(\triangleright\mathrm{R})}{\langle u_0 = X_0|X_1\rangle \quad \vdash u_0 : B \Rightarrow (A \triangleright (A|B))}\,(\Rightarrow\mathrm{R})$$

Et voici un exemple de preuve nécessitant la règle (|R) *avec* contraction :

```
# (A v 0) | (‘A v 0)
<> :- u : (A v 0) | (‘A v 0)
* Proved with 50 proof attempts.
```

Intuitivement, on constate bien que pour montrer cette proposition nous allons être amenés à utiliser le caractère classique de la logique par le biais du tiers-exclu. En effet, en notant P le processus désigné par l'index u, on est amené à distinguer deux cas : soit P valide la formule $\mathcal{A}$, soit P ne valide pas la formule $\mathcal{A}$; et dans chacun des cas on peut montrer que P valide bien la formule $(\mathcal{A} \vee 0)|(\neg\mathcal{A} \vee 0)$ en utilisant la propriété $P \equiv P|0 \equiv 0|P$. La démonstration est très instructive car elle fait bien sentir que c'est la contraction qui va permettre de tenir un raisonnement utilisant le tiers-exclu. Malheureusement cette preuve est trop grande pour pouvoir tenir dans cette feuille[5].

[5] Monsieur Fermat sera certainement d'accord avec moi.

6 Extension de la logique à CCS (système $\mathcal{S}_{\mathbf{ccs}}$)

Cette partie présente des idées qui étaient en cours de formalisation à la fin du stage. La preuve du système proposée n'a pas été complètement élaborée mais elle semble faisable, quitte à modifier quelques détails dans les règles d'inférence présentées.

Pour étendre la logique de sorte qu'elle puisse admettre comme modèle des processus de CCS (sans l'opérateur de restriction ν), nous sommmes amenés à redéfinir quelques notions. Cette extension de la logique permet de se rapprocher de l'observation de propriétés liées à l'interaction entre processus, ce qui justifie le fait de garder une algèbre de processus comme modèle de la logique (et non pas l'ensemble des unions finies d'intervalles de $\mathbb{N}$) ; ainsi nous pouvons faire des observations sur les processus qui soient à la fois comportementales et spatiales.

Définition 27 (Processus). L'ensemble $\mathcal{P}$ des processus est défini inductivement par

$$P, Q ::= 0 \quad | \quad b.P \quad | \quad P|Q$$

où b appartient à l'ensemble $\mathcal{N} \cup \overline{\mathcal{N}}$ des noms de canaux éventuellements utilisés en *input* (un canal ainsi utilisé est noté $\overline{b}$ où $\overline{b}$ est un élément de $\overline{\mathcal{N}}$).

Définition 28 (Formule). L'ensemble $\mathcal{F}$ des formules est défini inductivement par

$$\mathcal{A}, \mathcal{B} ::= \mathbf{F} \quad | \quad A_w \quad | \quad \mathcal{A} \wedge \mathcal{B} \quad | \quad \mathcal{A} \Rightarrow \mathcal{B} \quad | \quad 0 \quad | \quad \mathcal{A}|\mathcal{B} \quad | \quad \mathcal{A} \triangleright \mathcal{B} \quad | \quad b.\mathcal{A} \quad | \quad \Diamond\mathcal{A}$$

Définition 29 (Domaine de validité). La définition du domaine de validité est identique à la définition 8 à laquelle on a ajouté les définitions

$$[\![b.\mathcal{A}]\!]_v = \{b.P \quad / \quad P \in [\![A]\!]_v\}$$
$$[\![\Diamond\mathcal{A}]\!]_v = \{b.P|\overline{b}.Q|R \quad / \quad b \in \mathcal{N} \quad \wedge \quad P|Q|R \in [\![A]\!]_v\}$$

6.1 Règles d'inférence pour les préfixes

Dans les règles qui suivent b et c sont soit des préfixes d'entrée (*input*), soit des préfixes de sortie (*output*).

$$\frac{[\mathcal{X} \text{ et } \mathcal{Y} \text{ non libres dans la conclusion}] \quad \langle S, u \doteq b.\mathcal{X}|\mathcal{Y}\rangle \ \Gamma, \mathcal{X}|\mathcal{Y} : \mathcal{A} \vdash \Delta}{\langle S\rangle \ \Gamma, u : b.\mathcal{A} \vdash \Delta}\,(b\text{L}) \qquad \frac{\langle S, u \doteq b.v|t\rangle \ \Gamma \vdash v|t : \mathcal{A}, u : b.\mathcal{A}, \Delta}{\langle S\rangle \ \Gamma \vdash u : b.\mathcal{A}, \Delta}\,(b\text{R})$$

$$\frac{[\mathcal{X}, \mathcal{Y} \text{ et } \mathcal{Z} \text{ non libres dans la conclusion}] \quad \text{Pour tout } b \in \mathcal{N} \quad \langle S, u \doteq b.\mathcal{X}|\overline{b}.\mathcal{Y}|\mathcal{Z}, v \doteq \mathcal{X}|\mathcal{Z}\rangle \ \Gamma, v : \mathcal{A} \vdash \Delta}{\langle S\rangle \ \Gamma, u : \Diamond\mathcal{A} \vdash \Delta}\,(\Diamond\text{L}) \qquad \frac{\langle S\rangle \ \Gamma \vdash s : \mathcal{A}, \Delta \quad u \doteq_S b.v|\overline{b}.w|t \quad s \doteq_S v|w}{\langle S\rangle \ \Gamma \vdash u : \Diamond\mathcal{A}, \Delta}\,(\Diamond\text{R})$$

$$\frac{\langle S, u \doteq b.v|c.v'|t'', t \doteq c.v'|t'', t' \doteq b.v'|t''\rangle \ \Gamma \vdash \Delta}{\langle S, u \doteq b.v|t, u \doteq c.v'|t'\rangle \ \Gamma \vdash \Delta}\,(Sbc) \qquad \frac{\langle S, u \doteq b.v|t, v' \doteq v, t' \doteq t\rangle \ \Gamma \vdash \Delta}{\langle S, u \doteq b.v|t, u \doteq b.v'|t'\rangle \ \Gamma \vdash \Delta}\,(Sbb)$$

$$\frac{\langle S\rangle \ \Gamma, u : \mathbf{F} \vdash \Delta}{\langle S, u \doteq b.v, u \doteq 0\rangle \ \Gamma \vdash \Delta}\,(Sb0)$$

Une version du système de règles plus proche de l'implémentation que l'on pourrait en faire dans le prouveur ainsi que des éléments de preuve quant à la correction et à la complétude de ce nouveau système se trouvent en annexe C.

7 Remerciements

Je tiens à remercier messieurs Daniel Hirschkoff et Étienne Lozes pour leur aide précieuse, leur grande disponibilité, ainsi que pour leur humour au niveau des blagues ; l'ÉNS pour m'avoir accueilli dans ses locaux, me permettant ainsi de découvrir le monde merveilleux de l'entreprise ; LaTeX et mon sponsor officiel, la table complète des symboles LaTeX ([Pak02]), sans laquelle je n'aurais jamais pu faire des notations aussi surchargées.

Références

[Cai02] Luís Caires. Model-Checking of Spatial Properties in the π-Calculus. octobre 2002.

[CC02] Luís Caires and Luca Cardelli. A Spatial Logic for Concurrency (Part I), 2002.

[CC03] Luís Caires and Luca Cardelli. A Spatial Logic for Concurrency (Part II), mai 2003.

[CCG02] C. Calcagno, L. Cardelli, and A. Gordon. Deciding validity in a spatial logic for trees. Technical Report MSR–TR–2002, Microsoft Research, 2002.

[CG00] Luca Cardelli and Andrew D. Gordon. Anytime, Anywhere – Modal Logics for Mobile Ambients, 2000.

[CT01] Witold Charatonik and Jean-Marc Talbot. The Decidability of Model Checking Mobile Ambients. *Lecture Notes in Computer Science*, 2142, 2001.

[Gal87] J. H. Gallier. *Logic for Computer Science – Foundations of Automatic Theorem Proving*. Harper and Row Computer Science and Technology Series, 1987.

[Mil91] Robin Milner. The Polyadic π-Calculus : a Tutorial, octobre 1991.

[Mil99] Robin Milner. Commmunicating and Mobile Systems : the π-Calculus, 1999.

[Pak02] Scott Pakin. The Comprehensive LaTeX Symbol List, 2002.

A Système $\mathcal{S}_1$

Notation 4. Si $u \doteq_S X_1|\dots|X_n$ alors $X_1|\dots|X_n$ peut être noté $\overline{u}^S$.

Définition 30 (Système $\mathcal{S}_1$). Il est identique au système $\mathcal{S}_0$ sauf pour les règles suivantes (les règles (CL) et (CR) ont de plus été retirées) :

$$\frac{\langle S, u \doteq v|t, v \doteq X_1'|\dots|X_n', t \doteq X_1''|\dots|X_n'', X_1 \doteq X_1'|X_1'', \dots, X_n \doteq X_n'|X_n''\rangle \ \Gamma, v : \mathcal{A}, t : \mathcal{B} \vdash \Delta}{\langle S, u \doteq X_1|\dots|X_n\rangle \ \Gamma, u : \mathcal{A}|\mathcal{B} \vdash \Delta} \ (|\text{L})$$

$$\frac{\langle S\rangle \ \Gamma \vdash v : \mathcal{A}, u : \mathcal{A}|\mathcal{B}, \Delta \quad \langle S\rangle \ \Gamma \vdash t : \mathcal{B}, u : \mathcal{A}|\mathcal{B}, \Delta \quad u \doteq_S \overline{v}^S|\overline{t}^S}{\langle S\rangle \ \Gamma \vdash u : \mathcal{A}|\mathcal{B}, \Delta} \ (|\text{R})$$

$$\frac{\langle S\rangle \ \Gamma, u : \mathcal{A} \triangleright \mathcal{B} \vdash X : \mathcal{A}, \Delta \quad \langle S\rangle \ \Gamma, X|u : \mathcal{B}, u : \mathcal{A} \triangleright \mathcal{B} \vdash \Delta}{\langle S\rangle \ \Gamma, u : \mathcal{A} \triangleright \mathcal{B} \vdash \Delta} \ (\triangleright\text{L})$$

$$\frac{\begin{array}{c}[\mathcal{X} \textit{ non libre dans la conclusion}] \\ \langle S\rangle \ \Gamma, \mathcal{X} : \mathcal{A} \vdash v : \mathcal{B}, \Delta \quad v \doteq_S \mathcal{X}|\overline{u}^S\end{array}}{\langle S\rangle \ \Gamma \vdash u : \mathcal{A} \triangleright \mathcal{B}, \Delta} \ (\triangleright\text{R})$$

Proposition 6 (Correction). *Le système S_1 est correct i. e. tout séquent prouvable est valide.*

Démonstration. Par induction sur l'arbre de preuve utilisé.

Montrons que pour toute règle $\dfrac{\mathcal{S}'}{\mathcal{S}}$, si $\mathcal{S}'$ est validée par toute interprétation alors $\mathcal{S}$ l'est aussi.

Par exemple, traitons le cas de la règle (|L). Soit

$$\mathcal{S} = \langle S, u \doteq X_1|\dots|X_n\rangle \ \Gamma, u : \mathcal{A}|\mathcal{B} \vdash \Delta$$

un séquent. Supposons que le séquent

$$\mathcal{S}' = \langle S, u \doteq v|t, v \doteq X_1'|\dots|X_n', t \doteq X_1''|\dots|X_n'', X_1 \doteq X_1'|X_1'', \dots, X_n \doteq X_n'|X_n''\rangle \ \Gamma, v : \mathcal{A}, t : \mathcal{B} \vdash \Delta$$

soit validé par toute interprétation. Soit $\mathscr{I}$ une interprétation de $\mathcal{S}$ telle que $\mathscr{I} \models_- \Gamma, u : \mathcal{A}|\mathcal{B}$ (et $\mathscr{I} \models \langle S, u \doteq X_1|\dots|X_n\rangle$). On a $\mathscr{I}(u) \models \mathcal{A}|\mathcal{B}$ et $u \doteq_S X_1|\dots|X_n$ donc il existe deux processus P et Q tels que $\mathscr{I}(u) \equiv P|Q$, $P \models \mathcal{A}$ et $Q \models \mathcal{B}$. De plus, $\mathscr{I}(u) \equiv \mathscr{I}(X_1|\dots|X_n) \equiv \mathscr{I}(X_1)|\dots|\mathscr{I}(X_n)$. On peut donc construire une interprétation $\mathscr{I}'$ de $\mathcal{S}'$ telle que $\forall i \in [\![1, n]\!], \mathscr{I}(X_i) \equiv \mathscr{I}'(X_i')|\mathscr{I}'(X_i'')$, $\mathscr{I}'(v) \equiv \mathscr{I}'(X_1'|\dots|X_n') \models \mathcal{A}$ et $\mathscr{I}'(t) \equiv \mathscr{I}'(X_1''|\dots|X_n'') \models \mathcal{B}$. $\mathscr{I}'$ vérifie alors $\mathscr{I}' \models_- \Gamma, v : \mathcal{A}, t : \mathcal{B}$. Donc, par $\mathcal{S}'$, on a $\mathscr{I}' \models_+ \Delta$. Or on a $\forall i \in [\![1, n]\!], \mathscr{I}(X_i) \equiv \mathscr{I}'(X_i')|\mathscr{I}'(X_i'') \equiv \mathscr{I}'(x_i)$. On en déduit que $\mathscr{I} \models_+ \Delta$. Donc $\mathscr{I}$ valide $\mathcal{S}$. Le séquent $\mathcal{S}$ est donc valide.

Les autres cas se traitent de façon analogue. □

B Système $\mathcal{S}_2$

B.1 Règles

$$\frac{u \doteq_S v}{\langle S\rangle\ \Gamma, u : A \vdash v : A, \Delta}\,(\mathrm{Id})$$

$$\frac{}{\langle S\rangle\ \Gamma, u : \mathbf{F} \vdash \Delta}\,(\mathbf{F}\mathrm{L}) \qquad \frac{\langle S\rangle\ \Gamma \vdash \Delta}{\langle S\rangle\ \Gamma \vdash u : \mathbf{F}, \Delta}\,(\mathbf{F}\mathrm{R})$$

$$\frac{\langle S\rangle\ \Gamma, u : \mathcal{A}, u : \mathcal{B} \vdash \Delta}{\langle S\rangle\ \Gamma, u : \mathcal{A} \wedge \mathcal{B} \vdash \Delta}\,(\wedge\mathrm{L}) \qquad \frac{\langle S\rangle\ \Gamma \vdash u : \mathcal{A}, \Delta \quad \langle S\rangle\ \Gamma \vdash u : \mathcal{B}, \Delta}{\langle S\rangle\ \Gamma \vdash u : \mathcal{A} \wedge \mathcal{B}, \Delta}\,(\wedge\mathrm{R})$$

$$\frac{\langle S\rangle\ \Gamma \vdash u : \mathcal{A}, \Delta \quad \langle S\rangle\ \Gamma, u : \mathcal{B} \vdash \Delta}{\langle S\rangle\ \Gamma, u : \mathcal{A} \Rightarrow \mathcal{B} \vdash \Delta}\,(\Rightarrow\mathrm{L}) \qquad \frac{\langle S\rangle\ \Gamma, u : \mathcal{A} \vdash u : \mathcal{B}, \Delta}{\langle S\rangle\ \Gamma \vdash u : \mathcal{A} \Rightarrow \mathcal{B}, \Delta}\,(\Rightarrow\mathrm{R})$$

$$\frac{\langle S, u \doteq 0\rangle\ \Gamma \vdash \Delta}{\langle S\rangle\ \Gamma, u : 0 \vdash \Delta}\,(0\mathrm{L}) \qquad \frac{u \doteq_S 0}{\langle S\rangle\ \Gamma \vdash u : 0, \Delta}\,(0\mathrm{R})$$

$$\frac{\begin{array}{c}[\sigma \text{ est la subsitution telle que } \forall i \in [\![1,n]\!], \sigma(X_i) = X_i^1 | X_i^2] \\ \left\langle \sigma(S), v \doteq X_1^1 | \ldots | X_n^1, t \doteq X_1^2 | \ldots | X_n^2 \right\rangle\ \Gamma, v : \mathcal{A}, t : \mathcal{B} \vdash \Delta \quad u \doteq_S X_1 | \ldots | X_n\end{array}}{\langle S\rangle\ \Gamma, u : \mathcal{A}|\mathcal{B} \vdash \Delta}\,(|\mathrm{L})$$

$$\frac{\langle S\rangle\ \Gamma \vdash v : \mathcal{A}, \Delta \quad \langle S\rangle\ \Gamma \vdash t : \mathcal{B}, \Delta}{\langle S\rangle\ \Gamma \vdash v \between t : \mathcal{A} \between \mathcal{B}, \Delta}\,(|\mathrm{R}')$$

$$\frac{\left\langle S, v_i \doteq \overline{v}_i^S, t_i \doteq \overline{t}_i^S \right\rangle\ \Gamma \vdash v_1 \between t_1 : \mathcal{A} \between \mathcal{B}, \ldots, v_n \between t_n : \mathcal{A} \between \mathcal{B}, \Delta \quad \text{avec } \{(v_i, t_i) \quad / \quad i \in [\![1,n]\!]\} = \widehat{u}^S}{\langle S\rangle\ \Gamma \vdash u : \mathcal{A}|\mathcal{B}, \Delta}\,(|\mathrm{R})$$

$$\frac{\begin{array}{ll} & [\textit{Avec } \mathcal{A} \in \mathcal{F}_0,\ n = \|\mathcal{A} \triangleright \mathcal{B}\| \textit{ et } J \subseteq [\![1,n]\!]] \\ \text{Pour tout } j \in J & \langle S, t_1 \doteq Y_1, \ldots, t_j \doteq Y_1 | \ldots | Y_j, y_1 \doteq Y_1, \ldots, y_j \doteq Y_j\rangle\ \Gamma \vdash t_j : \mathcal{A}, y_1 : 0, \ldots, y_j : 0, \Delta \\ \text{Pour tout } j \in [\![1,n]\!] \setminus J & \langle S, t_1 \doteq Y_1, \ldots, t_j \doteq Y_1 | \ldots | Y_j, y_1 \doteq Y_1, \ldots, y_j \doteq Y_j\rangle\ \Gamma, t_j : \mathcal{A} \vdash y_1 : 0, \ldots, y_j : 0, \Delta \\ & \left\langle S, \underbrace{v_j \doteq u | t_j, t_j \doteq Y_1 | \ldots | Y_j}_{\text{pour tout } j \in J}, y_1 \doteq Y_1, \ldots, y_n \doteq Y_n \right\rangle\ \Gamma, \underbrace{v_j : \mathcal{B}}_{j \in J} \vdash y_1 : 0, \ldots, y_n : 0, \Delta \end{array}}{\begin{array}{c}\langle S\rangle\ \Gamma, u : \mathcal{A} \triangleright \mathcal{B} \vdash \Delta \\ \hline \left\langle S, v \doteq_S \overline{t}^S | \overline{u}^S, t = Y_1 | \ldots | Y_{\|\mathcal{A} \triangleright \mathcal{B}\|} \right\rangle\ \Gamma, t : \mathcal{A} \vdash v : \mathcal{B}, \Delta \\ \hline \langle S\rangle\ \Gamma \vdash u : \mathcal{A} \triangleright \mathcal{B}, \Delta\end{array}}\,(\triangleright\mathrm{L})\,(\triangleright\mathrm{R})$$

$$\frac{S \Vdash u_1 \sim_{\|\mathcal{A}_1\|} v_1, \ldots, u_n \sim_{\|\mathcal{A}_n\|} v_n, Z \quad \text{avec } \{(u_i : \mathcal{A}_i, v_i : \mathcal{A}_i) \quad / \quad i \in [\![1,n]\!]\} \in \Gamma \times \Delta}{\langle S\rangle\ \Gamma \vdash \Delta, \widetilde{Z}}\,(\widetilde{\mathrm{Id}})$$

B.2 Règles des connecteurs dérivés

$$\frac{\langle S\rangle\ \Gamma \vdash \Delta}{\langle S\rangle\ \Gamma, u : \top \vdash \Delta}\,(\top\mathrm{L}) \qquad \frac{}{\langle S\rangle\ \Gamma \vdash u : \top, \Delta}\,(\top\mathrm{R})$$

$$\frac{\langle S\rangle\ \Gamma, u : \mathcal{A} \vdash \Delta \quad \langle S\rangle\ \Gamma, u : \mathcal{B} \vdash \Delta}{\langle S\rangle\ \Gamma, u : \mathcal{A} \vee \mathcal{B} \vdash \Delta}\,(\vee\mathrm{L}) \quad \frac{\langle S\rangle\ \Gamma \vdash u : \mathcal{A}, u : \mathcal{B}, \Delta}{\langle S\rangle\ \Gamma \vdash u : \mathcal{A} \vee \mathcal{B}, \Delta}\,(\vee\mathrm{R})$$

$$\frac{\langle S\rangle\ \Gamma \vdash u : \mathcal{A}, \Delta}{\langle S\rangle\ \Gamma, u : \neg\mathcal{A} \vdash \Delta}\,(\neg\mathrm{L}) \qquad \frac{\langle S\rangle\ \Gamma, u : \mathcal{A} \vdash \Delta}{\langle S\rangle\ \Gamma \vdash u : \neg\mathcal{A}, \Delta}\,(\neg\mathrm{R})$$

$$\frac{\left\langle S, v_i \doteq \overline{v}_i^S, t_i \doteq \overline{t}_i^S \right\rangle\ \Gamma, v_1 : \mathcal{A}, \ldots, v_n : \mathcal{A} \vdash \Delta \quad \langle S\rangle\ \Gamma, t_1 : \mathcal{B}, \ldots, t_n : \mathcal{B} \vdash \Delta \quad \text{avec } \{(v_i, t_i) \quad / \quad i \in [\![1, n]\!]\} = \widehat{u}^S}{\langle S\rangle\ \Gamma, u : \mathcal{A}||\mathcal{B} \vdash \Delta}\,(||\mathrm{L})$$

$$\frac{\begin{array}{c}[\sigma \textit{ est la substitution telle que } \forall i \in [\![1, n]\!], \sigma(X_i) = X_i^1|X_i^2] \\ \left\langle \sigma(S), v \doteq X_1^1|\ldots|X_n^1, t \doteq X_1^2|\ldots|X_n^2 \right\rangle\ \Gamma \vdash v : \mathcal{A}, t : \mathcal{B}, \Delta\end{array}}{\langle S\rangle\ \Gamma \vdash u : \mathcal{A}||\mathcal{B}, \Delta}\,(||\mathrm{R})$$

Ces règles se déduisent des règles de $\mathcal{S}_2$ en utilisant les définitions des connecteurs dérivés (cf. [CC03] pour les démonstrations).

B.3 Démonstrations

B.3.1 Démonstration de la proposition 4 (bool-correction de $\mathcal{S}_2$)

On traite chaque règle ou axiome séparément :

- **Axiome** (**Id**). Soit $\mathcal{S} = \langle S\rangle\ \Gamma, u : \mathcal{A} \vdash v : \mathcal{A}, \Delta$ un séquent avec S tel que $u \doteq_S v$. Soit $\mathscr{I}$ une bool-interprétation de $\mathcal{S}$ telle que $\mathscr{I} \models_- \Gamma, u : \mathcal{A}$. Comme $u \doteq_S v$ et que $\mathscr{I}$ valide $\langle S\rangle$, on a $\mathscr{I}(u) \equiv \mathscr{I}(v)$. De plus, par hypothèse, $\mathscr{I}$ valide $u : \mathcal{A}$ donc $\mathscr{I}$ valide $v : \mathcal{A}$, soit encore $\mathscr{I} \models v : \mathcal{A}, \Delta$. Donc le séquent $\mathcal{S}$ est bool-valide.
- **Axiome** (**F**L). Soit $\mathcal{S} = \langle S\rangle\ \Gamma, u : \mathbf{F} \vdash \Delta$ un séquent. Aucune bool-interprétation $\mathscr{I}$ ne peut être telle que $\mathscr{I} \models \langle S\rangle$ et $\mathscr{I} \models_- \Gamma, u : \mathbf{F}$ car $[\![\mathbf{F}]\!]_{\mathscr{I}} = \emptyset$. Donc le séquent $\mathcal{S}$ est bool-valide.
- **Règle** (**F**R). Soient $\mathcal{S} = \langle S\rangle\ \Gamma \vdash u : \mathbf{F}, \Delta$ et $\mathcal{S}' = \langle S\rangle\ \Gamma \vdash \Delta$ deux séquents. On suppose que $\mathcal{S}'$ est bool-valide. Soit $\mathscr{I}$ une interprétation de $\mathcal{S}'$ telle que $\mathscr{I} \models_- \Gamma$. Comme $\mathcal{S}'$ est valide, on a donc $\mathscr{I} \models_+ \Delta$ donc $\mathscr{I} \models_+ u : \mathbf{F}, \Delta$. Le séquent $\mathcal{S}$ est donc bool-valide.
- **Règle** ($\wedge$L). Soient $\mathcal{S} = \langle S\rangle\ \Gamma, u : \mathcal{A} \wedge \mathcal{B} \vdash \Delta$ et $\mathcal{S}' = \langle S\rangle\ \Gamma, u : \mathcal{A}, u : \mathcal{B} \vdash \Delta$ deux séquents. On suppose que $\mathcal{S}'$ est bool-valide. Soit $\mathscr{I}$ une bool-interprétation de $\mathcal{S}$ vérifiant $\mathscr{I} \models_- \Gamma, u : \mathcal{A} \wedge \mathcal{B}$. C'est aussi une interprétation de $\mathcal{S}'$ et elle vérifie $\mathscr{I}(u) \models \mathcal{A}$ et $\mathscr{I}(u) \models \mathcal{B}$ donc $\mathscr{I} \models_- \Gamma, u : \mathcal{A}, u : \mathcal{B}$. $\mathcal{S}'$ étant supposé bool-valide, on a donc $\mathscr{I} \models_+ \Delta$. Donc le séquent $\mathcal{S}$ est bool-valide.
- **Règle** ($\wedge$R). Soient $\mathcal{S} = \langle S\rangle\ \Gamma \vdash u : \mathcal{A} \wedge \mathcal{B}, \Delta$, $\mathcal{S}' = \langle S\rangle\ \Gamma \vdash u : \mathcal{A}, \Delta$ et $\mathcal{S}'' = \langle S\rangle\ \Gamma \vdash u : \mathcal{B}, \Delta$ trois séquents. On suppose que $\mathcal{S}'$ et $\mathcal{S}''$ sont bool-valides. Soit $\mathscr{I}$ une bool-interprétation de $\mathcal{S}$ telle que $\mathscr{I} \models_- \Gamma$ et $\mathscr{I} \not\models \Delta$. C'est aussi une interprétation de $\mathcal{S}'$ et $\mathcal{S}''$ donc $\mathscr{I}(u) \models \mathcal{A}$ et $\mathscr{I}(u) \models \mathcal{B}$ car $\mathscr{I} \not\models_+ \Delta$. Donc $\mathscr{I}(u) \models \mathcal{A} \wedge \mathcal{B}$ et $\mathcal{S}$ est un séquent bool-valide.
- **Règle** ($\Rightarrow$L). Soient $\mathcal{S} = \langle S\rangle\ \Gamma, \mathcal{A} \Rightarrow \mathcal{B} \vdash \Delta$, $\mathcal{S}' = \langle S\rangle\ \Gamma \vdash u : \mathcal{A}, \Delta$ et $\mathcal{S}'' = \langle S\rangle\ \Gamma, u : \mathcal{B} \vdash \Delta$ trois séquents. On suppose que $\mathcal{S}'$ et $\mathcal{S}''$ sont bool-valides. Soit $\mathscr{I}$ une interprétation de $\mathcal{S}$ telle que $\mathscr{I} \models_- \Gamma, u : \mathcal{A} \Rightarrow \mathcal{B}$. $\mathscr{I}$ est aussi une interprétation de $\mathcal{S}'$, qui est supposé

bool-valide, et $\mathscr{I} \models_- \Gamma$ donc $\mathscr{I} \models_+ u : \mathcal{A}, \Delta$. Si $\mathscr{I} \models_+ \Delta$ la démonstration est terminée. Supposons que $\mathscr{I}(u) \models \mathcal{A}$. On a, par hypothèse, $\mathscr{I}(u) \models \mathcal{A} \Rightarrow \mathcal{B}$. On en déduit que $\mathscr{I}(u) \models \mathcal{B}$. Donc $\mathscr{I} \models \Gamma, u : \mathcal{B}$. Et, d'après le séquent $\mathcal{S}''$ supposé bool-valide, on en déduit que $\mathscr{I} \models_+ \Delta$. Dans tous les cas le séquent $\mathcal{S}$ est donc bool-valide.

- **Règle** ($\Rightarrow$R). Soient $\mathcal{S} = \langle S \rangle\ \Gamma \vdash u : \mathcal{A} \Rightarrow \mathcal{B}, \Delta$ et $\mathcal{S}' = \langle S \rangle\ \Gamma, u : \mathcal{A} \vdash u : \mathcal{B}, \Delta$ deux séquents. On suppose que $\mathcal{S}'$ est valide. Soit $\mathscr{I}$ une interprétation de $\mathcal{S}$ telle que $\mathscr{I} \models_- \Gamma$ et $\mathscr{I} \not\models \Delta$. Si $\mathscr{I}(u) \not\models \mathcal{A}$ alors $\mathscr{I}(u) \models \mathcal{A} \Rightarrow \mathcal{B}$ et la démonstration est terminée. Supposons que $\mathscr{I}(u) \models \mathcal{A}$. Dans ce cas, $\mathscr{I} \models_- \Gamma, u : \mathcal{A}$ et, par le séquent $\mathcal{S}'$ supposé bool-valide, on déduit que $\mathscr{I} \models u : \mathcal{B}, \Delta$. Or par hypothèse, $\mathscr{I} \not\models \Delta$. Donc $\mathscr{I}(u) \models \mathcal{B}$. Et, comme $\mathscr{I}(u) \models \mathcal{A}$, on a $\mathscr{I}(u) \models \mathcal{A} \Rightarrow \mathcal{B}$ donc $\mathscr{I} \models_+ u : \mathcal{A} \Rightarrow \mathcal{B}, \Delta$. Le séquent $\mathcal{S}$ est donc bool-valide.
- **Règle** (0L). Soient $\mathcal{S} = \langle S \rangle\ \Gamma, u : 0 \vdash \Delta$ et $\mathcal{S}' = \langle S, u \doteq 0 \rangle\ \Gamma \vdash \Delta$ deux séquents. On suppose que $\mathcal{S}'$ est bool-valide. Soit $\mathscr{I}$ une bool-interprétation de $\mathcal{S}$ telle que $\mathscr{I} \models_- \Gamma, u : 0$. C'est en particulier une bool-interprétation de $\mathcal{S}'$ qui staisfait tout Γ donc elle satisfait partiellement Δ car $\mathcal{S}'$ est supposé bool-valide. Donc le séquent $\mathcal{S}$ est bool-valide.
- **Axiome** (0R). Soit $\mathcal{S} = \langle S \rangle\ \Gamma \vdash u : 0, \Delta$ un séquent tel que $u \doteq_S 0$. Soit $\mathscr{I}$ une interprétation de $\mathcal{S}$. Cette interprétation valide en particulier la contrainte $u \doteq_S 0$, donc $\mathscr{I}(u) \models 0$. D'où $\mathscr{I} \models_+ u : 0, \Delta$. Le séquent $\mathcal{S}$ est donc bool-valide.
- **Règle** (|L). Soient $\mathcal{S} = \langle S \rangle\ \Gamma, u : \mathcal{A}|\mathcal{B} \vdash \Delta$ et

$$\mathcal{S}' = \left\langle S\left[X_i \leftarrow X_i^1|X_i^2\right]_{i\in[\![1,k]\!]}, v \doteq X_1^1|\ldots|X_k^1, t \doteq X_1^2|\ldots|X_k^2 \right\rangle\ v : \mathcal{A}, t : \mathcal{B}, \Gamma \vdash \Delta$$

deux séquents avec S tel que $u \doteq_S X_1|\ldots|X_k$. On suppose que $\mathcal{S}'$ est bool-valide. Soit $\mathscr{I}$ une bool-interprétation de $\mathcal{S}$ telle que $\mathscr{I} \models \langle S \rangle$ et $\mathscr{I} \models_- \Gamma, u : \mathcal{A}|\mathcal{B}$. $\mathscr{I}$ valide $u \doteq_S X_1|\ldots|X_k$ et $u : \mathcal{A}|\mathcal{B}$ donc, quitte à réindexer les X_i, il existe p tel que $\mathscr{I}(X_1|\ldots|X_p) \models \mathcal{A}$ et $\mathscr{I}(X_{p+1}|\ldots|X_k) \models \mathcal{B}$.Soit $\mathscr{I}'$ la bool-interprétation de $\mathcal{S}'$ telle que $\mathscr{I}'(X_i^1) = \begin{cases} \mathscr{I}(X_i) & \text{si } i \leq p \\ 0 & \text{sinon} \end{cases}$, $\mathscr{I}'(X_i^2) = \begin{cases} \mathscr{I}(X_i) & \text{si } i > p \\ 0 & \text{sinon} \end{cases}$ et $\mathscr{I}'(X) = \mathscr{I}(X)$ sinon. On a alors $\mathscr{I}'(v) = \mathscr{I}(X_1|\ldots|X_p) \models \mathcal{A}$ et de même $\mathscr{I}'(t) \models \mathcal{B}$. Donc $\mathscr{I}' \models_- v : \mathcal{A}, t : \mathcal{B}, \Gamma$. Le séquent $\mathcal{S}'$ étant bool-valide, on en déduit que $\mathscr{I}' \models_- \Delta$. Donc $\mathscr{I} \models_- \Delta$ car pour tout i dans $[\![1,k]\!]$, $\mathscr{I}(X_i) = \mathscr{I}'(X_i^1|X_i^2)$ par définition (ceci a du sens car X_i^1 et X_i^2 ne sont jamais en même temps égaux à 1). Le séquent $\mathcal{S}$ est donc valide.
- **Règle** (|R′). C'est la définition de $u \between v : \mathcal{A} \between \mathcal{B}$.
- **Règle** (|R). Soient $\mathcal{S} = \langle S \rangle\ \Gamma \vdash u : \mathcal{A}|\mathcal{B}, \Delta$ et

$$\mathcal{S}' = \left\langle S, v_i \doteq \overline{v}_i^S, t_i \doteq \overline{t}_i^S \right\rangle\ \Gamma \vdash v_1 \between t_1 : \mathcal{A} \between \mathcal{B}, \ldots, v_n \between t_n : \mathcal{A} \between \mathcal{B}, \Delta$$

des séquents avec S tel que pour tout i dans $[\![1,n]\!]$ on ait $u \doteq_S \overline{v}_i^S|\overline{t}_i^S$. On suppose que $\mathcal{S}'$ est bool-valide. Soit $\mathscr{I}$ une bool-interprétation de $\mathcal{S}$ telle que $\mathscr{I} \models_- \Gamma$ et $\mathscr{I} \not\models_+ \Delta$. $\mathscr{I}$ est aussi une interprétation de $\mathcal{S}'$, qui est supposé bool-valide, et est telle que $\mathscr{I} \models \Gamma$. Donc $\mathscr{I} \models_+ v_1 \between t_1 : \mathcal{A} \between \mathcal{B}, \ldots, v_n | t_n : \mathcal{A} \between \mathcal{B}, \Delta$, d'où $\mathscr{I} \models_+ v_1 \between t_1 : \mathcal{A} \between \mathcal{B}, \ldots, v_n \between t_n : \mathcal{A} \between \mathcal{B}$ car, par hypothèse, $\mathscr{I} \not\models \Delta$. Ainsi, il existe k tel que $\mathscr{I}(v_k) \models \mathcal{A}$ et $\mathscr{I}(t_k) \models \mathcal{B}$ et, de plus, $u \doteq_S v_k|t_k$, donc $\mathscr{I}(u) \equiv \mathscr{I}(v_k|t_k) \equiv \mathscr{I}(v_k)|\mathscr{I}(t_k) \models \mathcal{A}|\mathcal{B}$ soit encore $\mathscr{I}(u) \models_+ u : \mathcal{A}|\mathcal{B}, \Delta$. Le séquent $\mathcal{S}'$ est donc bool-valide.
- **Règle** ($\triangleright$L). On remarque que l'ensemble $J = \{j \in [\![1,n]\!] \quad / \quad \underbrace{1|\ldots|1}_{j \text{ fois}} \models \mathcal{A}\}$, avec $n = \|\mathcal{A} \triangleright \mathcal{B}\|$, convient (pour les séquents $\mathcal{S}_j^+$ et $\mathcal{S}_j^-$ qui suivent). Soient

$$\mathcal{S}_j^+ = \langle S, t_1 \doteq Y_1, \ldots, t_j \doteq Y_1|\ldots|Y_j, y_1 \doteq Y_1, \ldots, y_j \doteq Y_j \rangle\ \Gamma \vdash t_j : \mathcal{A}, y_1 : 0, \ldots, y_j : 0, \Delta$$

avec $j \in J$,

$$\mathcal{S}_j^- = \langle S, t_1 \doteq Y_1, \ldots, t_j \doteq Y_1|\ldots|Y_j, y_1 \doteq Y_1, \ldots, y_j \doteq Y_j \rangle\ \Gamma, t_j : \mathcal{A} \vdash y_1 : 0, \ldots, y_j : 0, \Delta$$

avec $j \in [\![1,n]\!] \setminus J$,

$$\mathcal{S}' = \left\langle S, \underbrace{v_j \doteq u|t_j, t_j \doteq Y_1|\dots|Y_j}_{\text{pour tout } j \in J}, y_1 \doteq Y_1, \dots, y_n \doteq Y_n \right\rangle \ \Gamma, \underbrace{v_j : \mathcal{B}}_{j\in J} \vdash y_1 : 0, \dots, y_n : 0, \Delta$$

et $\mathcal{S} = \langle S \rangle\ \Gamma, u : \mathcal{A} \triangleright \mathcal{B} \vdash \Delta$ des séquents avec $\mathcal{A} \in \mathcal{F}_0$. On suppose que les $\mathcal{S}_j^+$, les $\mathcal{S}_j^-$ et $\mathcal{S}'$ sont bool-valides. Soit $\mathscr{I}$ une bool-interprétation de $\mathcal{S}$ telle que $\mathscr{I} \models_- \Gamma, u : \mathcal{A} \triangleright \mathcal{B}$ et $\mathscr{I} \not\models_+ \Delta$. Soit $\mathscr{I}'$ une bool-interprétation des $\mathcal{S}_j^\pm$ qui prolonge $\mathscr{I}$ et telle que $\forall i \in [\![1,j]\!], \mathscr{I}'(Y_i) = 1$. Puisque $\mathcal{S}'$ est bool-valide et que $\mathscr{I}' \not\models_+ \Delta, y_1 : 0, \dots, y_j : 0$, nécessairement, on a $\exists j_0 \in [\![1,j]\!], \mathscr{I}'(t_{j_0}) \not\models \mathcal{B}$. Or $j_0 \in J$ donc $\mathscr{I}'(t_{j_0}) \models \mathcal{A}$ mais $\mathscr{I}'(u)|\mathscr{I}'(t_{j_0}) \equiv \mathscr{I}'(u|t_{j_0}) \equiv \mathscr{I}'(v_{j_0}) \not\models \mathcal{B}$ et $\mathscr{I}'(u) \models \mathcal{A} \triangleright \mathcal{B}$. Il y a contradiction. Donc $\mathscr{I}' \models_+ \Delta$ d'où $\mathscr{I} \models_+ \Delta$ et $\mathcal{S}$ est bool-valide.

– **Règle** ($\triangleright$R). Soient $\mathcal{S} = \langle S \rangle\ \Gamma \vdash u : \mathcal{A} \triangleright \mathcal{B}, \Delta$ et

$$\mathcal{S}' = \left\langle S, v \doteq_S \overline{t}^S|\overline{u}^S, t = Y_1|\dots|Y_{\|\mathcal{A}\triangleright\mathcal{B}\|} \right\rangle\ \Gamma, t : \mathcal{A} \vdash v : \mathcal{B}, \Delta$$

des séquents. On suppose que $\mathcal{S}'$ est bool-valide. Soit $\mathscr{I}$ une interprétation de $\mathcal{S}$ telle que $\mathscr{I} \models_- \Gamma$ et $\mathscr{I} \not\models \Delta$. Soit $P \in \mathcal{P}$ un processus tel que $P \models \mathcal{A}$. Alors, d'après la proposition 2, il existe $Q \in \mathcal{P}$ tel que $P \sim_{\|\mathcal{A}\triangleright\mathcal{B}\|} Q$ et $\|Q\| \leq \|\mathcal{A} \triangleright \mathcal{B}\|$. On a alors aussi, par propriété, $P \sim_{\|\mathcal{A}\|} Q$ (car $\|\mathcal{A}\| \leq \|\mathcal{A} \triangleright \mathcal{B}\|$) et $P|\mathscr{I}(u) \sim_{\|\mathcal{A}\triangleright\mathcal{B}\|} Q|I(u)$. D'après la proposition 1, $Q|\mathscr{I}(u) \models \mathcal{B}$ si et seulement si $P|\mathscr{I}(u) \models \mathcal{B}$. Soit $\mathscr{I}'$ une interprétation de $\mathcal{S}'$ telle que $\mathscr{I}'$ prolonge $\mathscr{I}$ et $\mathscr{I}'(t) \equiv Q$ (ce qui est possible car $t \doteq_S Y_1|\dots|Y_{\|\mathcal{A}\triangleright\mathcal{B}\|}$ et $\|Q\| \leq \|\mathcal{A} \triangleright \mathcal{B}\|$. Comme $P \sim_{\|\mathcal{A}\|} Q$ et $P \models \mathcal{A}$, on a $\mathscr{I}'(t) \equiv Q \models \mathcal{A}$. Donc $\mathscr{I}'$ est une interprétation du séquent $\mathcal{S}'$, supposé valide, telle que $\mathscr{I}' \models_- \Gamma, t : \mathcal{A}$ donc $\mathscr{I}' \models_+ v : \mathcal{B}, \Delta$ d'où $\mathscr{I}'(v) \equiv \mathscr{I}'(u)|\mathscr{I}'(t) \models \mathcal{B}$ car, par hypothèse, $\mathscr{I} \not\models \Delta$.

– **Règle** ($\widetilde{\textbf{Id}}$). Soit $\mathcal{S} = \langle S \rangle\ \Gamma \vdash \Delta, \widetilde{Z}$ un séquent. On suppose que $S \Vdash u_1 \sim_{\|\mathcal{A}_1\|} v_1, \dots, u_n \sim_{\|\mathcal{A}_n\|} v_n, Z$ avec $\{(u_i : \mathcal{A}_i, v_i : \mathcal{A}_i) \ / \ i \in [\![1,n]\!]\} \in \Gamma \times \Delta$. Soit $\mathscr{I}$ une bool-interprétation de $\mathcal{S}$ telle que $\mathscr{I} \models \Gamma$. Si $\mathscr{I} \Vdash Z$ alors la démonstration est terminée. Supposons donc que $\mathscr{I} \not\Vdash Z$. Dans ce cas, il existe $(u : \mathcal{A}, v : \mathcal{A}) \in \Gamma \times \Delta$ tel que $\mathscr{I} \models u \doteq v$, soit encore $\mathscr{I}(v) = \mathscr{I}(u) \models \mathcal{A}$. Donc $\mathscr{I} \models_+ \Delta$. Le séquent $\mathcal{S}$ est donc bool-valide.

B.3.2 Démonstration de la proposition 5 (bool-complétude de $\mathcal{S}_2$)

Le système $\mathcal{S}_2$ est terminant. En effet, chaque règle fait décroître la taille de $\Gamma \cup \Delta$ pour l'ordre multiensemble induit par la taille des formules.

Pour chaque règle, si le séquent prouvé est valide alors le(s) séquent(s) antécédent(s) est (sont) valide(s) :

– **Règle** (**F**R). Soient $\mathcal{S} = \langle S \rangle\ \Gamma \vdash u : \mathbf{F}, \Delta$ et $\mathcal{S}' = \langle S \rangle\ \Gamma \vdash \Delta$ deux séquents. On suppose que $\mathcal{S}$ est bool-valide. Soit $\mathscr{I}$ une bool-interprétation de $\mathcal{S}'$ telle que $\mathscr{I} \models_- \Gamma$. C'est aussi une interprétation de $\mathcal{S}$ et l'on a donc $\mathscr{I} \models_+ u : \mathbf{F}, \Delta$. Or $[\![\mathbf{F}]\!] = \emptyset$ donc $\mathscr{I}(u) \not\models \mathbf{F}$. Donc $\mathscr{I} \models_+ \Delta$ et $\mathcal{S}'$ est bool-valide.

– **Règle** ($\wedge$L). Soient $\mathcal{S} = \langle S \rangle\ \Gamma, u : \mathcal{A} \wedge \mathcal{B} \vdash \Delta$ et $\mathcal{S}' = \langle S \rangle\ \Gamma, u : \mathcal{A}, u : \mathcal{B} \vdash \Delta$ deux séquents. On suppose que $\mathcal{S}$ est bool-valide. Soit $\mathscr{I}$ une interprétation de $\mathcal{S}'$ telle que $\mathscr{I} \models_- \Gamma, u : \mathcal{A}, u : \mathcal{B}$. On a $\mathscr{I}(u) \models \mathcal{A}$ et $\mathscr{I}(u) \models \mathcal{B}$ donc $\mathscr{I}(u) \models \mathcal{A} \wedge \mathcal{B}$. $\mathscr{I}$ est donc une bool-interprétation de $\mathcal{S}$ qui valide tout $\Gamma, u : \mathcal{A} \wedge \mathcal{B}$. Donc $\mathscr{I} \models_+ \Delta$ et $\mathcal{S}'$ est bool-valide.

– **Règle** ($\wedge$R). Soient $\mathcal{S} = \langle S \rangle\ \Gamma \vdash u : \mathcal{A} \wedge \mathcal{B}, \Delta, \mathcal{S}' = \langle S \rangle\ \Gamma \vdash u : \mathcal{A}, \Delta$ et $\mathcal{S}'' = \langle S \rangle\ \Gamma \vdash u : \mathcal{B}, \Delta$ trois séquents. On suppose que $\mathcal{S}$ est bool-valide. Soit $\mathscr{I}$ une bool-interprétation $\mathcal{S}'$ telle que $\mathscr{I} \models_- \Gamma$ et $\mathscr{I} \not\models_+ \Delta$. $\mathscr{I}$ est aussi une interprétation de $\mathcal{S}$ et l'on a $\mathscr{I} \models_+ u : \mathcal{A} \wedge \mathcal{B}, \Delta$, soit encore $\mathscr{I}(u) \models \mathcal{A} \wedge \mathcal{B}$ car par hypothèse $\mathscr{I} \not\models_+ \Delta$. Le séquent $\mathcal{S}'$ est donc valide.
On montre de même que le séquent $\mathcal{S}''$ est valide.

- **Règle** ($\Rightarrow$L). Soient $\mathcal{S} = \langle S \rangle \ \Gamma, \mathcal{A} \Rightarrow \mathcal{B} \vdash \Delta, \mathcal{S}' = \langle S \rangle \ \Gamma \vdash u : \mathcal{A}, \Delta$ et $\mathcal{S}'' = \langle S \rangle \ \Gamma, u : \mathcal{B} \vdash \Delta$ trois séquents. On suppose que $\mathcal{S}$ est bool-valide. Soit $\mathscr{I}$ une bool-interprétation de $\mathcal{S}'$ telle que $\mathscr{I} \models_- \Gamma$. Si $\mathscr{I}(u) \models \mathcal{A}$ alors $\mathscr{I}$ valide $\mathcal{S}'$. Supposons que $\mathscr{I}(u) \not\models \mathcal{A}$. Dans ce cas, $\mathscr{I}(u) \models \mathcal{A} \Rightarrow \mathcal{B}$ et $\mathscr{I}$ est une interprétation de $\mathcal{S}$ telle que $\mathscr{I} \models_- \Gamma, u : \mathcal{A} \Rightarrow \mathcal{B}$. Donc $\mathscr{I} \models \Delta$ et $\mathscr{I}$ valide $\mathcal{S}'$. Le séquent $\mathcal{S}'$ est donc bool-valide.
  Soit $\mathscr{I}'$ une bool-interprétation de $\mathcal{S}''$ telle que $\mathscr{I}' \models_- \Gamma, u : \mathcal{B}$. $\mathscr{I}'$ est alors une interprétation de $\mathcal{S}$ telle que $\mathscr{I}' \models_- \Gamma, u : \mathcal{A} \Rightarrow \mathcal{B}$ donc $\mathscr{I}' \models_+ \Delta$. Donc $\mathscr{I}'$ valide $\mathcal{S}''$. Le séquent $\mathcal{S}''$ est donc bool-valide.
- **Règle** ($\Rightarrow$R). Soient $\mathcal{S} = \langle S \rangle \ \Gamma \vdash u : \mathcal{A} \Rightarrow \mathcal{B}, \Delta$ et $\mathcal{S}' = \langle S \rangle \ \Gamma, u : \mathcal{A} \vdash u : \mathcal{B}, \Delta$ deux séquents. On suppose que $\mathcal{S}$ est bool-valide. Soit $\mathscr{I}$ une bool-interprétation de $\mathcal{S}'$ telle que $\mathscr{I} \models_- \Gamma, u : \mathcal{A}$ et $\mathscr{I} \not\models_+ \Delta$. Par le séquent $\mathcal{S}$ on déduit que $\mathscr{I}(u) \models \mathcal{A} \Rightarrow \mathcal{B}$. Or $\mathscr{I}(u) \models \mathcal{A}$ donc $\mathscr{I}(u) \models \mathcal{B}$. Le séquent $\mathcal{S}'$ est donc bool-valide.
- **Règle** (0L). Soient $\mathcal{S} = \langle S \rangle \ \Gamma, u : 0 \vdash \Delta$ et $\mathcal{S}' = \langle S, u \doteq 0 \rangle \ \Gamma \vdash \Delta$ deux séquents. On suppose que $\mathcal{S}$ est bool-valide. Soit $\mathscr{I}$ une bool-interprétation de $\mathcal{S}'$ telle que $\mathscr{I} \models_- \Gamma$. Comme $u \doteq_S 0$, on a aussi $\mathscr{I}(u) \models 0$. Donc par $\mathcal{S}$ on déduit $\mathscr{I} \models_+ \Delta$. Donc $\mathcal{S}'$ est bool-valide.
- **Règle** ($|$L). Soient $\mathcal{S} = \langle S \rangle \ \Gamma, u : \mathcal{A}|\mathcal{B} \vdash \Delta$ et

$$\mathcal{S}' = \left\langle S\left[X_i \leftarrow X_i^1|X_i^2\right]_{i\in[\![1,n]\!]}, v \doteq X_1^1|\dots|X_n^1, t \doteq X_1^2|\dots|X_n^2 \right\rangle \ v : \mathcal{A}, t : \mathcal{B}, \Gamma \vdash \Delta$$

  deux séquents avec S tel que $u \doteq_S X_1|\dots|X_k$. On suppose que $\mathcal{S}$ est valide. Soit $\mathscr{I}$ une bool-interprétation de $\mathcal{S}'$ telle que $\mathscr{I} \models_- \Gamma, v : \mathcal{A}, t : \mathcal{B}$. Soit $\mathscr{I}'$ la bool-interprétation de $\mathcal{S}$ telle que $\mathscr{I}'$ soit identique à $\mathscr{I}$ pour les variables communes à $\mathcal{S}$ et $\mathcal{S}'$ et $\forall i \in [\![1,n]\!], \mathscr{I}'(X_i) \doteq \mathscr{I}(X_i^1|X_i^2)$ (d'après la proposition 1, on peut supposer sans perdre de généralité que l'on a toujours $\mathscr{I}(X_i^1|X_i^2) \in \{0,1\}$ car $\|\mathcal{A}|\mathcal{B}\| = \|\mathcal{A}\| + \|\mathcal{B}\|$, donc cette définition a du sens). L'interprétation $\mathscr{I}'$ vérifie $\mathscr{I}' \models_- \Gamma, u : \mathcal{A}|\mathcal{B}$ donc, par le séquent $\mathcal{S}$, on a $\mathscr{I}' \models_+ \Delta$. D'où $\mathscr{I} \models_+ \Delta$. Le séquent $\mathcal{S}'$ est donc bool-valide.
- **Règle** ($|$R$'$). C'est la définition de $u\between v : \mathcal{A}\between\mathcal{B}$.
- **Règle** ($|$R). Soient $\mathcal{S} = \langle S \rangle \ \Gamma \vdash u : \mathcal{A}|\mathcal{B}, \Delta$ et

$$\mathcal{S}' = \left\langle S, v_i \doteq \overline{v}_i^S, t_i \doteq \overline{t}_i^S \right\rangle \ \Gamma \vdash v_1\between t_1 : \mathcal{A}\between\mathcal{B}, \dots, v_n\between t_n : \mathcal{A}\between\mathcal{B}, \Delta$$

  des séquents. On suppose que $\mathcal{S}$ est valide. Soit $\mathscr{I}$ une bool-interprétation de $\mathcal{S}'$ telle que $\mathscr{I} \models_- \Gamma$ et $\mathscr{I} \not\models_+ \Delta$. $\mathscr{I}$ est aussi une interprétation de $\mathcal{S}$ d'où $\mathcal{I} \models_- v|t : \mathcal{A}|\mathcal{B}, \Delta$ donc $\mathscr{I}(u) \models \mathcal{A}|\mathcal{B}$ car $\mathscr{I} \not\models_+ \Delta$. Supposons que $u \doteq_S X_1|\dots|X_n$. $\mathscr{I}$ étant une bool-interprétation, les X_i sont insécables, donc, quitte à réindexer les X_i, il existe $k \in [\![1,n]\!]$ tel que $\mathscr{I}(X_1|\dots|X_k) \models \mathcal{A}$ et $\mathscr{I}(X_{k+1}|\dots|X_n) \models \mathcal{B}$. Par définition des (v_i, t_i), il existe $i_0 \in [\![1,n]\!]$ tel que $v_{i_0} \doteq_S X_1|\dots|X_k$ et $t_{i_0} \doteq_S X_{k+1}|\dots|X_n$ et l'on a $\mathscr{I}(v_{i_0}\between t_{i_0}) \models \mathcal{A}\between\mathcal{B}$. Donc $\mathscr{I}$ valide $\mathcal{S}'$. Ainsi $\mathcal{S}'$ est bool-valide.
- **Règle** ($\triangleright$L). Posons $J = \{j \in [\![1,n]\!] \quad / \quad \underbrace{1|\dots|1}_{j \text{ fois}} \models \mathcal{A}\}$, avec $n = \|\mathcal{A} \triangleright \mathcal{B}\|$; l'ensmble J est bien défini car la formule $\mathcal{A}$ est supposée close donc sont domaine de validité ne dépend pas de l'interprétation choisie. Soient

  $$\mathcal{S}_j^+ = \langle S, t_1 \doteq Y_1, \dots, t_j \doteq Y_1|\dots|Y_j, y_1 \doteq Y_1, \dots, y_j \doteq Y_j \rangle \ \Gamma \vdash t_j : \mathcal{A}, y_1 : 0, \dots, y_j : 0, \Delta$$

  avec $j \in J$,

  $$\mathcal{S}_j^- = \langle S, t_1 \doteq Y_1, \dots, t_j \doteq Y_1|\dots|Y_j, y_1 \doteq Y_1, \dots, y_j \doteq Y_j \rangle \ \Gamma, t_j : \mathcal{A} \vdash y_1 : 0, \dots, y_j : 0, \Delta$$

  avec $j \in [\![1,n]\!] \setminus J$,

  $$\mathcal{S}' = \left\langle S, \underbrace{v_j \doteq u|t_j, t_j \doteq Y_1|\dots|Y_j}_{\text{pour tout } j \in J}, y_1 \doteq Y_1, \dots, y_n \doteq Y_n \right\rangle \ \Gamma, \underbrace{v_j : \mathcal{B}}_{j\in J} \vdash y_1 : 0, \dots, y_n : 0, \Delta$$

et $\mathcal{S} = \langle S \rangle\ \Gamma, u : \mathcal{A} \triangleright \mathcal{B} \vdash \Delta$ des séquents avec $\mathcal{A} \in \mathcal{F}_0$. On suppose que $\mathcal{S}$ est bool-valide. $\mathscr{I}$ une bool-interprétation d'un $\mathcal{S}_{j_0}^+$ telle que $\mathscr{I} \models_- \Gamma$ et $\mathscr{I} \not\models_+ \Delta$. S'il existe $i \in [\![1, j]\!]_0$ tel que $\mathscr{I}(y_i) \models 0$ alors l'on a terminé de traiter ce cas. Supposons donc que $\forall i \in [\![1, n]\!], \mathscr{I}(Y_i) = 1$. Dans ce cas, on a $\mathscr{I}(t_{j_0}) = \underbrace{1|\ldots|1}_{j_0 \text{ fois}}$ avec $j_0 \in J$ donc $\mathscr{I}(t_{j_0}) \models \mathcal{A}$. Dans tous les cas $\mathscr{I}$ valide $\mathcal{S}_{j_0}^+$. Donc $\mathcal{S}_{j_0}^+$ est bool-valide. De même on montre que les $\mathcal{S}_j^-$ sont bool-valides.
Soit $\mathscr{I}$ une interprétation de $\mathcal{S}'$ telle $\mathscr{I} \models_- \Gamma, \underbrace{v_j : \mathcal{B}}_{j \in J}$. S'il existe $i \in [\![1, n]\!]$ tel que $\mathscr{I}(y_i) \models 0$ ou si $\mathscr{I} \models_+ \Delta$ alors on a terminé de traiter ce cas. Supposons donc $\forall i \in [\![1, n]\!], \mathscr{I}(Y_i) = 1$, soit encore que $\forall i \in [\![1, n]\!], \mathscr{I}(t_i) = \underbrace{1|\ldots|1}_{i \text{ fois}}$. Soit $P \in \mathcal{P}$ tel que $P \models \mathcal{A}$. D'après la proposition 2, il existe $j_0 \in J$ tel que $P \sim_{\|\mathcal{A} \triangleright \mathcal{B}\|} \underbrace{1|\ldots|1}_{j \text{ fois}} = \mathscr{I}(t_{j_0})$.
On a alors $\mathscr{I}(t_{j_0}) \models \mathcal{A}$ (car $\|\mathcal{A}\| \leq \|\mathcal{A} \triangleright \mathcal{B}\|$ donc $P \sim_{\|\mathcal{A}\|} \mathscr{I}(t_{j_0})$) et $\mathscr{I}(u)|\mathscr{I}(t_{j_0}) \equiv \mathscr{I}(u|t_{j_0}) \equiv \mathscr{I}(v_{j_0}) \models \mathcal{B}$. De même, on a $P \sim_{\|\mathcal{B}\|} \mathscr{I}(t_{j_0})$ donc $P|I(u) \sim_{\|\mathcal{B}\|} \mathscr{I}(t_{j_0})|\mathscr{I}(u)$ d'où $P|\mathscr{I}(u) \models \mathcal{B}$. On en déduit que $\mathscr{I}(u) : \mathcal{A} \triangleright \mathcal{B}$. Donc $\mathscr{I} \models_- \Gamma, u : \mathcal{A} \triangleright \mathcal{B}$. Par $\mathcal{S}$, on déduit que $\mathscr{I} \models_+ \Delta$. Le séquent $\mathcal{S}'$ est donc bool-valide.

– **Règle** ($\triangleright$R). Soient $\mathcal{S} = \langle S \rangle\ \Gamma \vdash u : \mathcal{A} \triangleright \mathcal{B}, \Delta$ et

$$\mathcal{S}' = \left\langle S, v \doteq_S \overline{t}^S | \overline{u}^S, t = Y_1 | \ldots | Y_{\|\mathcal{A} \triangleright \mathcal{B}\|} \right\rangle\ \Gamma, t : \mathcal{A} \vdash v : \mathcal{B}, \Delta$$

des séquents. On suppose $\mathcal{S}$ bool-valide. Soit $\mathscr{I}$ une bool-interprétation de $\mathcal{S}'$ telle que $\mathscr{I} \models_- \Gamma, t : \mathcal{A}$ et $\mathscr{I} \not\models_+ \Delta$. $\mathscr{I}$ est en particulier une interprétation de $\mathcal{S}$ d'où $\mathscr{I}(u) \models \mathcal{A} \triangleright \mathcal{B}$ car, par hypothèse, $\mathscr{I} \not\models_+ \Delta$. Or $\mathscr{I}(t) \models \mathcal{A}$ et $\mathscr{I}(v) \equiv \mathscr{I}(u|t) \equiv \mathscr{I}(u)|\mathscr{I}(t)$ donc $\mathscr{I}(v) \models \mathcal{B}$. Ainsi $\mathscr{I}$ valide $\mathcal{S}'$. Donc $\mathcal{S}'$ est valide.

– **Règle** ($\widetilde{\textbf{Id}}$). Soit $\mathcal{S} = \langle S \rangle\ \Gamma \vdash \Delta, \widetilde{Z}$ un séquent supposé valide. Montrons que $S \Vdash u_1 \sim_{\|\mathcal{A}_1\|} v_1, \ldots, u_n \sim_{\|\mathcal{A}_n\|} v_n, Z$ avec $\{(u_i : \mathcal{A}_i, v_i : \mathcal{A}_i) \quad / \quad i \in [\![1, n]\!]\} \in \Gamma \times \Delta$. Raisonnons par l'absurde et prenons $\mathscr{I}$ une bool-interprétation de $\mathcal{S}$ qui ne valide pas $S \Vdash u_1 \sim_{\|\mathcal{A}_1\|} v_1, \ldots, u_n \sim_{\|\mathcal{A}_n\|} v_n, Z$. On a $\mathscr{I} \models S$, $\mathscr{I} \not\models_+ u_1 \sim_{\|\mathcal{A}_1\|} v_1, \ldots, u_n \sim_{\|\mathcal{A}_n\|} v_n$ et $\mathscr{I} \not\models_+ Z$. Supposons de plus que $\mathscr{I}(\mathcal{A}) = \left\{\mathscr{I}(u)_{/\|\mathcal{A}\|} \quad / \quad u : \mathcal{A} \in \Gamma\right\}$. En utilisant la proposition 1, on montre aisément que $\mathscr{I} \models \Gamma$. Si l'on avait aussi $\mathscr{I} \models \Delta$, il existerait $v : \mathcal{B} \in \Delta$ tel que $\mathscr{I}(v) \models \mathcal{B}$ et $u : \mathcal{B} \in \Gamma$ tel que $\mathscr{I}(u) \sim_{\|\mathcal{B}\|} \mathscr{I}(v)$. Donc $\mathcal{S}$ n'est pas valide. Par contraposée, cette règle est complète.

On vérifie de plus que les seuls cas terminaux sont les axiomes.

C Système $\mathcal{S}_{\mathbf{ccs}}$

Ici aussi le travail qui est présenté est un travail qui était en cours *à la fin du stage et que nous n'avons pas eu le temps de véritablement approfondir ; nous ne présentons donc que des pistes sur ce que pourraient être les preuves.*

Les tailles maintenant utilisées sur les processus et les formules sont des extensions de celles précédemment introduites. Elles prennent en compte la hauteur et la profondeur. Elles sont formellement définies dans [CCG02] ainsi que la nouvelle relation d'équivalence associée (notée $\sim_{w,h}$).

C.1 Règles implémentables

Pour l'implémentation, on part d'un arbre de taille maximale (en largeur et en profondeur) qui va « s'annuler » petit à petit. On introduit pour ce faire des variables de canaux (notées i, j) et on engendre des contraintes sur ces variables de canaux (du type $i \ddot{=} b$ ou $i \ddot{=} \varepsilon$). Une nouvelle relation d'équivalence $\doteq$ qui permet d'exprimer des contraintes sur ces variables de canaux est définie par :

$$
\begin{array}{l}
i \ddot{=} j \in S \Leftrightarrow i \ddot{=}_S j \\
i \ddot{=}_S b \text{ et } j \ddot{=}_S b \Rightarrow i \ddot{=}_S j \\
\overline{\overline{b}} \ddot{=}_S b \\
\overline{\varepsilon} \ddot{=}_S \varepsilon \\
i \ddot{=}_S i \\
i \ddot{=}_S j \Rightarrow j \ddot{=}_S i \\
i \ddot{=}_S j, j \ddot{=}_S k \Rightarrow i \ddot{=}_S k
\end{array}
$$

On étend la congruence sur les contraintes $\doteq$ en y ajoutant les règles :

$$
\begin{array}{l}
i \ddot{=}_S j \Rightarrow i.u \doteq_S j.u \\
\varepsilon.0 \doteq_S 0
\end{array}
$$

Les règles d'inférence sont :

$$
\frac{[\mathcal{X} \text{ et } \mathcal{Y} \text{ non libres dans la conclusion}]\quad \langle S, u \doteq i.\mathcal{X}|\mathcal{Y}, i \ddot{=} b\rangle\ \Gamma, \mathcal{X}|\mathcal{Y} : \mathcal{A} \vdash \Delta}{\langle S\rangle\ \Gamma, u : b.\mathcal{A} \vdash \Delta}\,(b\mathrm{L}) \qquad \frac{\langle S, u \doteq i.v|t, i \ddot{=} b\rangle\ \Gamma \vdash v|t : \mathcal{A}, u : b.\mathcal{A}, \Delta}{\langle S\rangle\ \Gamma \vdash u : b.\mathcal{A}, \Delta}\,(b\mathrm{R})
$$

$$
\frac{[\mathcal{X}, \mathcal{Y} \text{ et } \mathcal{Z} \text{ non libres dans la conclusion}]\quad \langle S, u \doteq i.\mathcal{X}|\overline{j}.\mathcal{Y}|\mathcal{Z}, i \ddot{=} j\rangle\ \Gamma, \mathcal{X}|\mathcal{Y}|\mathcal{Z} : \mathcal{A} \vdash \Delta}{\langle S\rangle\ \Gamma, u : \Diamond\mathcal{A} \vdash \Delta}\,(\Diamond\mathrm{L}) \qquad \frac{\langle S, u \doteq i.v|\overline{j}.w|t, i \ddot{=} j\rangle\ \Gamma \vdash v|w|t : \mathcal{A}, u : \Diamond\mathcal{A}, \Delta}{\langle S\rangle\ \Gamma \vdash u : \Diamond\mathcal{A}, \Delta}\,(\Diamond\mathrm{R})
$$

Enfin, les règles permettant de maintenir la cohérence sont :

$$
\frac{\langle S, u \doteq i.v|j.v'|t'', t \doteq j.v'|t'', t' \doteq i.v'|t''\rangle\ \Gamma \vdash \Delta}{\langle S, u \doteq i.v|t, u \doteq j.v'|t'\rangle\ \Gamma \vdash \Delta}\,(Sbc) \qquad \frac{\langle S, u \doteq i.v|t, v' \doteq v, t' \doteq t\rangle\ \Gamma \vdash \Delta}{\langle S, u \doteq i.v|t, u \doteq i.v'|t'\rangle\ \Gamma \vdash \Delta}\,(Sbb)
$$

$$
\frac{[\text{avec } b \neq c]\quad \langle S\rangle\ \Gamma, u : \mathbf{F} \vdash \Delta}{\langle S, i \ddot{=} b, i \ddot{=} c\rangle\ \Gamma \vdash \Delta}\,(Sibc) \qquad \frac{\langle S\rangle\ \Gamma, u : 0, v : 0 \vdash \Delta}{\langle S, u \doteq \varepsilon.v\rangle\ \Gamma \vdash \Delta}\,(S\varepsilon 0)
$$

La règle suivante est dérivable des précédentes :

$$
\frac{\langle S, u \doteq v, v \doteq 0, i \ddot{=} \varepsilon\rangle\ \Gamma \vdash \Delta}{\langle S, u \doteq i.v, u \doteq 0\rangle\ \Gamma \vdash \Delta}\,(Sib0)
$$

C.2 Indications sur les preuves

Définition 31 (Interprétation insécable). Une interprétation insécable $\mathscr{I}$ est une fonction qui :

- à toute variable propositionnelle A associe une partie de l'ensemble $\{0\} \cup \{b.0 \;/\; b \in \mathcal{N}\}$;
- à toute variable d'index i associe un nom de canal b de $\mathcal{N}$;
- à toute variable de processus X associe un processus de $\mathcal{P}$.

Remarque 6. On peut montrer que la validité sur les interprétations $\mathscr{I}$ définies comment précédemment est équivalente à la validité sur les interprétations qui à une variable propositionnelle A associent une partie de l'ensemble $\{0\} \cup \{b.0 \;/\; b \in \mathcal{N}'\}$ où $\mathcal{N}'$ est égal à l'ensemble des noms de canaux apparaissant dans la formule à valider plus un nom frais. L'ensemble $\mathcal{N}'$ ainsi défini est fini.

Les preuves de correction et de complétude du nouveau système de règles sont des extensions de celles faites en §B.3.1 et B.3.2. Elles se font en remplaçant les bool-interprétations par des interprétations insécables et la règle ($\triangleright$L) par une règle qui rende compte de la nouvelle relation d'équivalence utilisée, à savoir $\sim_{w,h}$.

Table des matières