Contraintes de sous-typage dans les quasi-treillis

Emmanuel Coquery et François Fages
Projet Contraintes, INRIA Rocquencourt,
BP 105, 78153 Le Chesnay Cedex, FRANCE
{Emmanuel.Coquery, Francois.Fages} @inria.fr

9 mars 2003

Résumé

Dans cet article nous montrons la décidabilité des contraintes de sous-typage non-structurel dans les quasi-treillis. Ce problème posé par Smolka en 1988 est important pour le typage des langages logiques et fonctionnels. Nous généralisons l'algorithme de Trifonov et Smith dans les treillis, au cas des quasi-treillis avec une complexité en temps en $O(k_{min}^v * k_{max}^v * n^3)$, où k_{min} (resp. k_{max}) est le nombre d'éléments minimaux (resp. maximaux) du quasi-treillis, v est le nombre de variables non bornées, et n est le nombre de contraintes. Nous étendons de la même manière l'algorithme de Pottier de calcul explicite de solutions au cas des quasi-treillis. Nous évoquons ensuite les applications de ces résultats, notamment au système TCLP de typage des programmes logiques avec contraintes.

1 Introduction

La recherche de systèmes de types de plus en plus flexibles pour les langages de programmation, va de paire avec la recherche d'algorithmes de résolution de contraintes de typage dans des structures de types de plus en plus complexes. Dans le cadre de nos travaux sur le système de typage TCLP des programmes logiques avec contraintes [2], nous nous intéressons à la résolution de contraintes de sous-typage non structurel dans des structures de types plus générales que les treillis.

Le sous-typage non-structurel permet d'exprimer des relations de sous-typage comme par exemple list$(\alpha) \leq$ term qui permet de voir une liste (homogène) comme un terme. Les algorithmes de vérification et d'inférence des types d'un programme se ramènent essentiellement à résoudre des systèmes de contraintes de sous-typage de la forme $\exists X \bigwedge_{i=1}^{n} t_i \leq t_i'$ où les t_i,t_i' sont des types et X est l'ensemble des variables apparaissant dans le système.

Lorsque les constructeurs de types forment un treillis, Trifonov et Smith [?] ont donné un algorithme simple de décomposition, ayant une complexité en temps en $O(n^3)$, pour tester la satisfiabilité des contraintes de sous-typage non-structurel dans le treillis des types infinis ou réguliers (dans les types finis, la décidabilité de ces contraintes est un problème ouvert). Pottier [5] a étendu cet algorithme pour calculer explicitement des solutions lorsqu'il en existe. La structure de treillis des constructeurs de types imposent cependant qu'il existe un élément minimum $\perp$ et un élément maximum $\top$, et ne traite pas le typage par $\perp$ comme une erreur.

De façon à nous affranchir de ces restrictions, nous nous intéressons au cas où les constructeurs de types forment un quasi-treillis, c'est-à-dire un ensemble partiellement ordonné dont toute partie finie admettant un minorant (resp. majorant) admet une borne inférieure (resp. supérieure). La résolution des contraintes de sous-typage non-structurel dans les quasi-treillis est un problème ouvert mentionné dans la thèse de Smolka [**?**]. Dans cet article, nous apportons une réponse positive à ce problème en généralisant les algorithmes de Trifonov-Smith et Pottier aux hypothèses de quasi-treillis.

Le plan de l'article est le suivant. Dans la prochaine section, on définit l'ensemble ordonné des types infinis formés sur un quasi-treillis de constructeurs, et on montre que cet ensemble a une structure de quasi-treillis. Dans la section suivante on montre que les systèmes de contraintes clos au sens des règles de décomposition de Trifonov et Smith sont encore satisfiables sous les hypothèses de quasi-treillis, et on donne un algorithme de test de satisfiabilité des contraintes en $O(k_{min}^v * k_{max}^v * n^3)$, où k_{min} (resp. k_{max}) est le nombre d'éléments minimaux (resp. maximaux) du quasi-treillis, v est le nombre de variables non bornées, et n est le nombre de contraintes. Dans la section 4, on généralise l'algorithme de Pottier de calcul explicite de solutions aux hypothèses de quasi-treillis. Enfin la section 5 présente des applications de ces résultats et nous concluons.

2 Types infinis

2.1 Préliminaires

Soit $(E, \leq)$ un ensemble partiellement ordonné. Soit $S \subseteq E$. On note $\mathrm{Mi}_E(S)$ l'ensemble des minorants de S et $\mathrm{Ma}_E(S)$ l'ensemble des majorants de S. On note $\mathrm{bi}_E(S)$ (resp. $\mathrm{bs}_E(S)$) la borne inférieure (resp. supérieure) de S si elle existe. On note $\mathrm{Mi}_E(\{a,b\}) = \mathrm{Mi}_E(a,b)$ et $\mathrm{Ma}_E(\{a,b\}) = \mathrm{Ma}_E(a,b)$. Un inf-quasi-treillis (resp. sup-quasi-treillis) est un ensemble partiellement ordonné dont toute partie finie admettant un minorant (resp. un majorant) admet une borne inférieure (resp. supérieure). Un quasi-treillis est un inf et sup-quasi-treillis.

Définition 2.1 (Quasi-treillis complet) *Un ensemble partiellement ordonné* $(E, \leq)$ *est un* quasi-treillis complet *(au sens des ensembles), si :*

- *pour tout* $S \neq \emptyset \subseteq E$, *si* $\mathrm{Mi}_E(S) \neq \emptyset$, *alors* $\mathrm{bi}_E(S)$ *existe.*
- *pour tout* $S \neq \emptyset \subseteq E$, *si* $\mathrm{Ma}_E(S) \neq \emptyset$, *alors* $\mathrm{bs}_E(S)$ *existe.*

2.2 Etiquettes

Comme mentionné dans l'introduction, nous nous intéressons à des langages de types qui autorisent des relations de sous-typage entre des constructeurs d'arité différente, comme $\text{list}(\alpha) \le \text{term}$. De façon générale, ces relations de sous-typage spécifient les relations entre les arguments des constructeurs, par exemple dans $k_1(\alpha,\beta) \le k_2(\beta)$, on spécifie que les types construits avec k_1 sont des sous-types de k_2 lorsque le deuxième argument de k_1 et l'argument de k_2 correspondent, le premier argument de k_1 est quant à lui oublié.

D'un point de vue formel, il est plus simple (et plus général) d'exprimer ces correspondances entre arguments, en travaillant avec une structure de termes étiquetés, comme développé dans la thèse de Pottier [3]. Dans ce formalisme, chaque argument d'un constructeur est désigné par une étiquette et non une position. On distingue de plus des étiquettes positives et négatives pour exprimer la covariance ou la contravariance des arguments, vis à vis de la relation de sous-typage.

Soient donc $\mathcal{L}^+$ et $\mathcal{L}^-$ deux ensembles disjoints dénombrables d'*étiquettes*, on pose $\mathcal{L} = \mathcal{L}^+ \uplus \mathcal{L}^-$. Soit $(\mathcal{K},\le_{\mathcal{K}})$ un quasi-treillis complet de *constructeurs* de types. Soit arg une fonction de $\mathcal{K}$ dans les parties finies de $\mathcal{L}$.

Définition 2.2 (Signature) $(\mathcal{K},\le_{\mathcal{K}},\mathcal{L}^+,\mathcal{L}^-,\text{arg})$ *est une* signature *si*

1. *pour tout* $k_1 \le_{\mathcal{K}} k_2 \le_{\mathcal{K}} k_3$, $\text{arg}(k_1) \cap \text{arg}(k_3) \subseteq \text{arg}(k_2)$
2. *pour tout* $S \subseteq \mathcal{K}$, *si* $\text{bi}_{\mathcal{K}}(S)$ *existe, alors* $\text{arg}(\text{bi}_{\mathcal{K}}(S)) \subseteq \bigcup_{s\in S} \text{arg}(s)$
3. *pour tout* $S \subseteq \mathcal{K}$, *si* $\text{bs}_{\mathcal{K}}(S)$ *existe, alors* $\text{arg}(\text{bs}_{\mathcal{K}}(S)) \subseteq \bigcup_{s\in S} \text{arg}(s)$
4. *pour tout* $k_1 \le_{\mathcal{K}} k_2$, *il existe* k *tel que* $k_1 \le_{\mathcal{K}} k \le_{\mathcal{K}} k_2$ *et* $\text{arg}(k) = \text{arg}(k_1) \cap \text{arg}(k_2)$.

Les conditions 1), 2), 3) de cohérences des étiquettes vis à vis de la relation d'ordre sont similaires à celles que l'on trouve dans [4] pour les treillis. La condition 4) est spécifique aux quasi-treillis. Elle a pour but de rejeter des signatures comme $k_1(\alpha) \le k_2(\beta)$ qui n'induisent pas une structure de quasi-treillis pour les types. En effet $k_2(k_3)$ et $k_2(k_4)$ admettent des minorants comme par exemple $k_1(k_3)$ ou $k_1(k_4)$ mais n'admettent pas de borne inférieure.

Pour une signature $(\mathcal{K},\le_{\mathcal{K}},\mathcal{L}^+,\mathcal{L}^-,\text{arg})$, on note $\mathcal{L}^*$ l'ensemble des chaînes finies d'étiquettes, ϵ la chaîne vide, "." la concaténation et $\|w\|$ la longueur de w. Nous nous intéressons aux types infinis formés sur $\mathcal{K}$, dans lequels les positions sont définies par des chaînes d'étiquettes :

Définition 2.3 (Types infinis) *Soit* $(\mathcal{K},\le_{\mathcal{K}},\mathcal{L}^+,\mathcal{L}^-,\text{arg})$ *une signature.*

Un type t *est une fonction partielle de* $\mathcal{L}^*$ *dans* $\mathcal{K}$ *telle que :*

1. *Son domaine est clos par préfixe :* $\forall w = w_1.w_2 \in \text{dom}(t), w_1 \in \text{dom}(t)$
2. $\epsilon \in \text{dom}(t)$
3. *pour tout* $w \in \text{dom}(t)$, *pour tout* $l \in \mathcal{L}$, $w.l \in \text{dom}(t)$ *si et seulement si* $l \in \text{arg}(t(w))$.

On note $\mathcal{T}(\mathcal{S})$ l'ensemble des types construits sur la signature $\mathcal{S}$. Par la suite, on supposera $\mathcal{S} = (\mathcal{K},\le_{\mathcal{K}},\mathcal{L}^+,\mathcal{L}^-,\text{arg})$ fixée et on notera $\mathcal{T} = \mathcal{T}(\mathcal{S})$. On note t/w

le type $t' : v \mapsto t(w.v)$. On note U/l, l'ensemble des sous-termes des types dans $U \subseteq \mathcal{T}$ à la position $l \in \mathcal{L}$, c'est-à-dire $U/l = \{t/l \mid t \in U \wedge l \in \arg(t(\epsilon))\}$.

Les notions suivantes seront utiles à la suite. On définit tout d'abord les minorants (ou les majorants) d'un constructeur k qui ont un ensemble restreint d'étiquettes communes avec k :

Définition 2.4 (Minorants et majorants par rapport à un ensemble d'étiquettes) *On dit que $k' \in \mathcal{K}$ est un* minorant *(resp.* majorant*) de $k \in \mathcal{K}$ par rapport à $L \in \mathcal{L}$ si $k' \leq_{\mathcal{K}} k$ (resp. $k \leq_{\mathcal{K}} k'$) et $\arg(k) \cap \arg(k') \subseteq L$.*

On note $\mathrm{Mi}_{\mathcal{K},\mathcal{L}}(k,L)$ (resp. $\mathrm{Ma}_{\mathcal{K},\mathcal{L}}$) l'ensemble des minorants (resp. majorants) de k par rapport à L.

On définit ensuite le sous-ensemble des étiquettes de k apparaissant dans $\mathrm{Mi}_{\mathcal{K},\mathcal{L}}(k,L)$

Définition 2.5 (Etiquettes significatives) *Le sous-ensemble des étiquettes significatives de $L \in \mathcal{L}$ au-dessous (resp. au-dessus) de k est définit par :*

$$\mathrm{ESMi}_{\mathcal{K},\mathcal{L}}(k,L) = \{l \in L \mid \exists k' \in \mathrm{Mi}_{\mathcal{K},\mathcal{L}}(k,L), l \in \arg(k) \cap \arg(k')\}$$

(resp. $\mathrm{ESMa}_{\mathcal{K},\mathcal{L}}(k,L) = \{l \in L \mid \exists k' \in \mathrm{Ma}_{\mathcal{K},\mathcal{L}}(k,L), l \in \arg(k) \cap \arg(k')\}$).

On vérifie aisément en utilisant les conditions de la définition d'un signature que

Propriete 2.6 *Si $\mathrm{Mi}_{\mathcal{K},\mathcal{L}}(k,L) \neq \emptyset$, alors $\mathrm{Mi}_{\mathcal{K},\mathcal{L}}(k,L)$ a un maximum $k' = \mathrm{bs}_{\mathcal{K}}(\mathrm{Mi}_{\mathcal{K},\mathcal{L}}(k,L))$ et $\arg(k') = \mathrm{ESMi}_{\mathcal{K},\mathcal{L}}(k,L)$.*
De même, si $\mathrm{Ma}_{\mathcal{K},\mathcal{L}}(k,L) \neq \emptyset$, alors $\mathrm{Mi}_{\mathcal{K},\mathcal{L}}(k,L)$ a un minimum $k' = \mathrm{bi}_{\mathcal{K}}(\mathrm{Ma}_{\mathcal{K},\mathcal{L}}(k,L))$ et $\arg(k') = \mathrm{ESMa}_{\mathcal{K},\mathcal{L}}(k,L)$.

2.3 Ordre sur les types

On définit l'ordre $\leq_{\mathcal{T}}$ sur les types par coinduction, comme l'intersection d'une suite $(\leq_n)$ de préordres sur les types définie par :

- $\leq_0 = \mathcal{T} \times \mathcal{T}$
- $t \leq_{n+1} t'$ si $t(\epsilon) \leq_{\mathcal{K}} t'(\epsilon)$ et pour tout $l \in \arg(t(\epsilon)) \cap \arg(t'(\epsilon))$:
  - soit $l \in \mathcal{L}^+$ et $t/l \leq_{\mathcal{K}} t'/l$
  - soit $l \in \mathcal{L}^-$ et $t'/l \leq_{\mathcal{K}} t/l$

Définition 2.7 (Ordre sur les types)

$$\leq_{\mathcal{T}} = \bigcap_{n \in \mathbb{N}} \leq_n$$

Propriete 2.8 *$\leq_{\mathcal{T}}$ est un ordre sur $\mathcal{T}$*

Démonstration : On montre par récurrence que pour tout $n \in \mathbb{N}$, $\leq_n$ est un préordre et on en déduit la transitivité et la réflexivité pour $\leq_{\mathcal{T}}$. Pour démontrer l'anti-symétrie, on commence par montrer, par récurrence, que pour tout $n \in \mathbb{N}$, si $t_1 \leq_{n+1} t_2$ et $t_2 \leq_{n+1} t_1$ alors pour tout $w \in \mathrm{dom}(t_1)$ tel que $\|w\| \leq n$, on a $w \in \mathrm{dom}(t_2)$ et $t_1(w) = t_2(w)$. Il suffit ensuite de prendre $t_1 \leq_{\mathcal{T}} t_2 \leq_{\mathcal{T}} t_1$: si

$t_1 \neq t_2$, alors il existe un w de taille minimale, telle que $t_1(w) \neq t_2(w)$. Or $t_1 \leq_{\|w\|+1} t_2 \leq_{\|w\|+1} t_1$ et donc $t_1(w) = t_2(w)$. $\square$

On vérifie de la même manière que pour deux types $t_1, t_2 \in \mathcal{T}$,

Propriete 2.9 *$t_1 \leq_{\mathcal{T}} t_2$ si et seulement si $t_1(\epsilon) \leq_{\mathcal{K}} t_2(\epsilon)$ et $\forall l \in \arg(t_1(\epsilon)) \cap \arg(t_2(\epsilon))$:*

- *soit $l \in \mathcal{L}^+$ et $t_1/l \leq_{\mathcal{T}} t_2/l$*
- *soit $l \in \mathcal{L}^-$ et $t_2/l \leq_{\mathcal{T}} t_1/l$*

Le but à présent est de montrer que l'ensemble des types ainsi ordonné est un quasi-treillis. Nous définissons tout d'abord l'ensemble des étiquettes utilisables sous un ensemble de type S comme l'ensemble des étiquettes l telles que S/l soit minoré (ou majoré, selon le cas) :

Définition 2.10 (Ensemble des étiquettes utilisables) *Soit $S \subseteq \mathcal{T}$. L'ensemble des* étiquettes utilisables au-dessus (resp . au-dessous) de *S est l'ensemble $l \in \mathcal{L}$ tels que :*

1. *$S/l \neq \emptyset$*
2. *l'une de ces deux affirmations est vraie*
   - *$l \in \mathcal{L}^+$ et $\mathrm{Mi}_{\mathcal{T}}(S/l) \neq \emptyset$ (resp. $\mathrm{Ma}_{\mathcal{T}}(S/l) \neq \emptyset$)*
   - *$l \in \mathcal{L}^-$ et $\mathrm{Ma}_{\mathcal{T}}(S/l) \neq \emptyset$ (resp. $\mathrm{Mi}_{\mathcal{T}}(S/l) \neq \emptyset$)*

On note $\mathrm{EUSous}(S)$ l'ensemble des étiquettes utilisables au-dessous de S et $\mathrm{EUSur}(S)$ l'ensemble des étiquettes utilisables au-dessus de S. On définit à présent ce qui sera le constructeur de tête des bornes inférieres et supérieures dans $\mathcal{T}$.

Définition 2.11 (Constructeur de borne) *Le* constructeur de borne inférieure *(resp.* supérieure*) d'un ensemble de types $S \subseteq \mathcal{T}$, noté $\mathcal{K}\mathrm{bi}(S)$ (resp. $\mathcal{K}\mathrm{bs}(S)$) est définit par :*

$$\begin{aligned} \mathcal{K}\mathrm{bi}(S) &= \mathrm{bs}_{\mathcal{K}}(\mathrm{Mi}_{\mathcal{K},\mathcal{L}}(\mathrm{bi}_{\mathcal{K}}(\{s(\epsilon) \mid s \in S\}), \mathrm{EUSous}(S))) \\ \mathcal{K}\mathrm{bs}(S) &= \mathrm{bi}_{\mathcal{K}}(\mathrm{Ma}_{\mathcal{K},\mathcal{L}}(\mathrm{bs}_{\mathcal{K}}(\{s(\epsilon) \mid s \in S\}), \mathrm{EUSur}(S))) \end{aligned}$$

On définit maintenant des suites de types qui approximent les bornes jusqu'à une profondeur donnée :

Définition 2.12 (Borne inférieure (resp. supérieure) de rang n) *La borne inférieure (resp. supérieure) de rang n d'un ensemble $S \subseteq \mathcal{T}$ avec $S \neq \emptyset$, notée $\mathrm{bi}_{\mathcal{T}n}(S)$ (resp. $\mathrm{bs}_{\mathcal{T}n}(S)$), est définie comme suit :*
Soit $\eta \in \mathcal{T}$ un type fixé quelconque[1] *:*

- $\mathrm{bi}_{\mathcal{T}0}(S) = \eta$
- $\mathrm{bs}_{\mathcal{T}0}(S) = \eta$
- $\mathrm{bi}_{\mathcal{T}n+1}(S)(\epsilon) = \mathcal{K}\mathrm{bi}(S)$ *et* $\forall l \in \arg(\mathcal{K}\mathrm{bi}(S))$ *:*
  - *soit $l \in \mathcal{L}^+$ et $\mathrm{bi}_{\mathcal{T}n+1}(S)/l = \mathrm{bi}_{\mathcal{T}n}(S/l)$*
  - *soit $l \in \mathcal{L}^-$ et $\mathrm{bi}_{\mathcal{T}n+1}(S)/l = \mathrm{bs}_{\mathcal{T}n}(S/l)$*

1. L'idée ici c'est d'avoir une définition correctement fondée pour les $\mathrm{bi}_{\mathcal{T}n}/\mathrm{bs}_{\mathcal{T}n}$. η n'est comparé qu'à lui même et son choix est donc complètement arbitraire

– $\mathrm{bs}_{\mathcal{T}n+1}(S)(\epsilon) = \mathcal{K}\mathrm{bs}(S)$ *et* $\forall l \in \arg(\mathcal{K}\mathrm{bs}(S))$ *:*
  – *soit* $l \in \mathcal{L}^+$ *et* $\mathrm{bi}_{\mathcal{T}n+1}(S)/l = \mathrm{bs}_{\mathcal{T}n}(S/l)$
  – *soit* $l \in \mathcal{L}^-$ *et* $\mathrm{bi}_{\mathcal{T}n+1}(S)/l = \mathrm{bi}_{\mathcal{T}n}(S/l)$

On peut à présent définir les candidats pour les bornes inférieures et supérieures dans $\mathcal{T}$:

Définition 2.13 ($\mathrm{bi}_{\mathcal{T}}$ et $\mathrm{bs}_{\mathcal{T}}$) *On défini la fonction partielle* $\mathrm{bi}_{\mathcal{T}} : \wp(\mathcal{T}) \to \mathcal{T}$ *(resp.* $\mathrm{bs}_{\mathcal{T}}$*) comme suit :*
Pour tout $n \in \mathbb{N}$*, pour tout* $w \in \mathrm{dom}(\mathrm{bi}_{\mathcal{T}n+1})$ *(resp.* $w \in \mathrm{dom}(\mathrm{bs}_{\mathcal{T}n+1})$*) tel que* $\|w\| = n$ *:*

$$\mathrm{bi}_{\mathcal{T}}(S)(w) = \mathrm{bi}_{\mathcal{T}n+1}(S)(w)$$

$$(\text{resp. } \mathrm{bs}_{\mathcal{T}}(S)(w) = \mathrm{bs}_{\mathcal{T}n+1}(S)(w)\ )$$

Propriete 2.14 *Soit* $S \subseteq \mathcal{T}$ *avec* $S \neq \emptyset$*. Si* $\mathrm{Mi}_{\mathcal{T}}(S) \neq \emptyset$ *(resp.* $\mathrm{Ma}_{\mathcal{T}}(S) \neq \emptyset$*) alors pour tout* $n \in \mathbb{N}$*,* $\forall s \in S, \mathrm{bi}_{\mathcal{T}n}(S) \leq_n s$ *(resp.* $\mathrm{bs}_{\mathcal{T}n}(S) \geq_n s$*).*

Propriete 2.15 *Soit* $S \subseteq \mathcal{T}$ *avec* $S \neq \emptyset$ *tel que* $\mathrm{Mi}_{\mathcal{T}}(S) \neq \emptyset$ *(resp.* $\mathrm{Mi}_{\mathcal{T}}(S) \neq \emptyset$*).*
$\forall s \in S, \mathrm{bi}_{\mathcal{T}}(S) \leq_{\mathcal{T}} s$ *(resp.* $\mathrm{bs}_{\mathcal{T}}(S) \geq_{\mathcal{T}} s$*)*

Propriete 2.16 *Soit* $S \subseteq \mathcal{T}$ *avec* $S \neq \emptyset$ *tel que* $\mathrm{Mi}_{\mathcal{T}}(S) \neq \emptyset$ *(resp.* $\mathrm{Mi}_{\mathcal{T}}(S) \neq \emptyset$*).*
$\forall t \in \mathrm{Mi}_{\mathcal{T}}(S), t \leq_{\mathcal{T}} \mathrm{bi}_{\mathcal{T}}(S)$ *(resp.* $\forall t \in \mathrm{Ma}_{\mathcal{T}}(S), t \geq_{\mathcal{T}} \mathrm{bs}_{\mathcal{T}}(S)$*).*

Théorème 2.17 $(\mathcal{T}, \leq_{\mathcal{T}})$ *est un quasi-treillis complet où* $\mathrm{bi}_{\mathcal{T}}$ *désigne les bornes inférieures et* $\mathrm{bs}_{\mathcal{T}}$ *les bornes supérieures.*

Démonstration : Soit $S \subseteq \mathcal{T}$ et $S \neq \emptyset$. Si S est minoré alors, par la propriété A.8, $\mathrm{bi}_{\mathcal{T}}(S)$ existe. Par 2.15, $\forall s \in S, \mathrm{bi}_{\mathcal{T}}(S) \leq_{\mathcal{T}} s$ et par 2.16 $\forall t \in \mathcal{T}, (\forall s \in S, t \leq_{\mathcal{T}} s) \Rightarrow t \leq_{\mathcal{T}} \mathrm{bi}_{\mathcal{T}}(S)$. Donc $\mathrm{bi}_{\mathcal{T}}(S)$ est la borne inférieure de S. De même, si S est majoré alors $\mathrm{bs}_{\mathcal{T}}(S)$ est défini et est la borne supérieure de S. Donc $(\mathcal{T}, \leq_{\mathcal{T}})$ est un quasi-treillis complet. □

On note $\mathcal{R}$ l'ensemble des termes réguliers (c'est-à-dire ayant un ensemble fini de sous-termes) de $\mathcal{T}$.

Propriete 2.18 (Bornes de deux types réguliers) *Soit* t_1 *et* t_2 *deux types réguliers. Si* $\mathrm{bi}_{\mathcal{T}}(t_1, t_2)$ *est définie alors c'est un type régulier. De même, si* $\mathrm{bs}_{\mathcal{T}}(t_1, t_2)$ *est définie alors c'est un type régulier.*

Théorème 2.19 $(\mathcal{R}, \leq_{\mathcal{T}})$ *est un quasi-treillis.*

Démonstration : Par le théorème 2.17, $(\mathcal{T}, \leq_{\mathcal{T}})$ est un quasi-treillis. Par la propriété 2.18, on a que si $r_1 \in \mathcal{R}$ et $r_2 \in \mathcal{R}$, et si $\exists r, r \leq_{\mathcal{T}} r_1 \wedge r \leq_{\mathcal{T}} r_2$, alors $\mathrm{bi}_{\mathcal{T}}(r_1, r_2) \in \mathcal{R}$. De même si $\exists r, r \geq_{\mathcal{T}} r_1 \wedge r \geq_{\mathcal{T}} r_2$, alors $\mathrm{bs}_{\mathcal{T}}(r_1, r_2) \in \mathcal{R}$. Donc $(\mathcal{R}, \leq_{\mathcal{T}})$ est un quasi-treillis dont la borne inférieure est définie par $\mathrm{bi}_{\mathcal{T}}$ et la borne supérieure par $\mathrm{bs}_{\mathcal{T}}$. □

Remarque : $(\mathcal{R}, \leq_{\mathcal{T}})$ n'est pas forcément un quasi-treillis *complet.* Considérons $\mathcal{K} = \{a, b\}$ avec $a \leq_{\mathcal{K}} b$ et $\arg(a) = \arg(b) = \{l\}$. Considérons la suite de types $(u_n)_n$ définie comme suit : $u_n(w) = b$ si $\|w\| = \frac{n(n+1)}{2}$, et $u_n(w) = a$ sinon. On peut vérifier qu'il n'existe pas de borne inférieure de $(u_n)_n$ dans $\mathcal{R}$.

3 Algorithme de test de satisfiabilité des contraintes de sous-typage

Soit $\mathcal{V}$ un ensemble dénombrable de variables, notées α,β,.... On définit les types avec variables comme l'ensemble des types formés à partir de la signature $(\mathcal{K} \cup \mathcal{V}, \leq_{\mathcal{K}}, \mathcal{L}^+, \mathcal{L}^-, \text{arg})$. On note cet ensemble $\mathcal{T}_{\text{VAR}}$.

Une contrainte de sous-typage est de la forme $t_1 \leq t_2$, avec $t_1,t_2 \in \mathcal{T}_{\text{VAR}}$ finis. Soit C un système de contraintes de sous-typage. On note VAR(C), le sous-ensemble de $\mathcal{V}$ qui apparaît dans C.

Définition 3.1 *On dit qu'une substitution $\rho : \mathcal{V} \rightarrow \mathcal{T}$* satisfait *la contrainte $t_1 \leq t_2$, noté $\rho \models t_1 \leq t_2$, si $\rho(t_1) \leq_{\mathcal{T}} \rho(t_2)$.*

La contrainte de sous-typage $t_1 \leq t_2$ est satisfiable *si et seulement si il existe une substitution ρ qui satisfait $t_1 \leq t_2$.*

Dans un but de simplification, nous supposons, sans perte de généralité, que les systèmes de contraintes considérés dans la suite ne contiennent que des petits termes. Un *petit terme* est un terme de pronfondeur maximale 1, dont toutes les feuilles sont des variables. Par exemple $\mathit{list}(\alpha)$ et α sont des petits termes mais $\mathit{list}(\mathit{int})$ n'en est pas un. Clairement, étant donné un système de contraintes, on peut trouver un système de contraintes équivalent dont tous les termes sont des petits termes, en introduisant des variables pour les arguments de termes qui ne sont pas des petits termes, et en introduisant des contraintes d'égalités (doubles inégalités) entre ces variables et les arguments correspondants.

3.1 Systèmes clos

On définit tout d'abord les systèmes prés-clos comme les ensembles de contraintes dont les variables sont bornées :

Définition 3.2 (Système pré-clos) *Un système de contraintes C est dit* pré-clos supérieurement *si pour toute variable $\alpha \in \text{VAR}(C)$, il existe t tel que $\alpha \leq t \in C$. C est dit* pré-clos inférieurement *si pour toute variable $\alpha \in \text{VAR}(C)$, il existe t tel que $t \leq \alpha \in C$. Un système de contraintes C est dit* pré-clos *s'il est pré-clos supérieurement et inférieurement.*

On définit à présent dans la table 1 la fonction subc de décomposition des contraintes, de l'algorithme de Trifonov et Smith [6].

Définition 3.3 (Système clos) *On dit qu'un système de contraintes C est* clos *s'il est pré-clos et si pour toute contrainte c,* $\text{subc}(c) \in C$*, pour tout $\{t_1 \leq \alpha, \alpha \leq t_2\} \subseteq C$,* $\text{subc}(t_1 \leq t_2)$ *est définie et inclue dans C et si pour toute variable α de C, il existe $t_1,t_2 \notin \mathcal{V}$ tels que $\{t_1 \leq \alpha, \alpha \leq t_2\} \subseteq C$.*

Nous introduisons à présent une série de notions et de lemmes techniques visant à démontrer qu'un système clos est satisfiable, théorème 3.7.

On définit $C\uparrow(\alpha) = \{t \mid t \notin \mathcal{V}, \alpha \leq t \in C\}$ et $C\downarrow(\alpha) = \{t \mid t \notin \mathcal{V}, t \leq \alpha \in C\}$. On étend ces fonctions aux ensembles de variables comme suit : $C\uparrow(A) =$

$$\begin{array}{rcl}
\mathrm{subc}(\alpha \leq \beta) & = & \{\alpha \leq \beta\} \\
\mathrm{subc}(\alpha \leq t) & = & \{\alpha \leq t\} \\
\mathrm{subc}(t \leq \alpha) & = & \{t \leq \alpha\} \\
\text{si } t_1(\epsilon) \leq_{\mathcal{K}} t_2(\epsilon) \ : & & \\
\mathrm{subc}(t_1 \leq t_2) & = & \displaystyle\bigcup_{l \in \arg(t_1(\epsilon)) \cap \arg(t_2(\epsilon)) \cap \mathcal{L}^+} \{t_1/l \leq t_2/l\} \cup \\
& & \displaystyle\bigcup_{l \in \arg(t_1(\epsilon)) \cap \arg(t_2(\epsilon)) \cap \mathcal{L}^-} \{t_2/l \leq t_1/l\}
\end{array}$$

TAB. 1 – *Fonction* subc *[5, 6]*

$\bigcup_{\alpha \in A} C{\uparrow}(\alpha)$ et $C{\downarrow}(A) = \bigcup_{\alpha \in A} C{\downarrow}(\alpha)$. On introduit également la notation suivante : $\mathrm{mi}_{\mathcal{K},\mathcal{K}}(k_1,k_2) = \mathrm{bs}_{\mathcal{K}}(\mathrm{Mi}_{\mathcal{K},\mathcal{L}}(k_1,\arg(k_2) \cap \arg(k_1)))$.

On définit la fonction sol sur les couples (A,B) d'ensembles de variable de C clos. Cette fonction permet de bâtir une solution ρ de C. On peut la voir comme une fonction donnant le constructeur de tête d'un type qui serait compris entre les $\rho(\alpha)$ et les $\rho(\beta)$, pour α dans A et β dans B.

Définition 3.4 *Soit C un système de contraintes clos. On introduit la fonction partielle* $\mathrm{sol} : \wp(\mathrm{VAR}(C)) \times \wp(\mathrm{VAR}(C)) \to \mathcal{K}$ *définie comme suit : Soit $A,B \subseteq \mathrm{VAR}(C)$, tels que $A \neq \emptyset$, $B \neq \emptyset$ et $\forall \alpha \in A, \forall \beta \neq \alpha \in B, \alpha \leq \beta \in C$. Soit $S = \{t(\epsilon) \mid t \in C{\uparrow}(B)\}$ et $I = \{t(\epsilon) \mid t \in C{\downarrow}(A)\}$. Si on note $k_S = \mathrm{bi}_{\mathcal{K}}(S)$ et $k_I = \mathrm{bs}_{\mathcal{K}}(I)$ alors $\mathrm{sol}(A,B) = \mathrm{mi}_{\mathcal{K},\mathcal{K}}(k_S,k_I)$.*

Notons que si C est clos, $A \neq \emptyset$, $B \neq \emptyset$ et pour tout $\alpha \in A, \beta \in B$ avec $\alpha \neq \beta$, $\alpha \leq \beta \in C$, alors $\mathrm{sol}(A,B)$ est bien défini. Soit $t \in C{\downarrow}(A)$ et $t' \in C{\uparrow}(B)$. Comme C est clos, $\mathrm{subc}(t \leq t')$ est définie, c'est-à-dire $t(\epsilon) \leq_{\mathcal{K}} t'(\epsilon)$. Donc $\forall k \in I, \forall k' \in S, k \leq_{\mathcal{K}} k'$. Comme C est clos, $C{\downarrow}(A) \neq \emptyset$ et $C{\uparrow}(B) \neq \emptyset$. Donc k_S et k_I sont bien définis, de plus $k_I \leq_{\mathcal{K}} k_S$.

Lemme 3.5 *Soit C un système de contraintes clos et $A,B \subseteq \mathrm{VAR}(C)$. Si $\mathrm{sol}(A,B)$ est défini, alors $\forall l \in \arg(\mathrm{sol}(A,B))$, si $l \in \mathcal{L}^+$, $\mathrm{sol}(C{\downarrow}(A)/l, C{\uparrow}(B)/l)$ est défini, si $l \in \mathcal{L}^-$, $\mathrm{sol}(C{\uparrow}(B)/l, C{\downarrow}(A)/l)$ est défini.*

Lemme 3.6 *Soit C un système de contraintes clos. Soit $A,B,E,F \subseteq \mathrm{VAR}(C)$. Si $C{\downarrow}(A) \subseteq C{\downarrow}(E)$ et $C{\uparrow}(F) \subseteq C{\uparrow}(B)$ et $\mathrm{sol}(A,B)$ et $\mathrm{sol}(E,F)$ sont définis, alors $\mathrm{sol}(A,B) \leq_{\mathcal{K}} \mathrm{sol}(E,F)$.*

Théorème 3.7 *Tout système de contraintes clos est satisfiable.*

Démonstration : On considère la fonction partielle $\gamma : \wp(\mathrm{VAR}(C)) \times \wp(\mathrm{VAR}(C)) \to \mathcal{T}$ définie sur le domaine de sol comme suit : $\gamma(A,B)(\epsilon) = \mathrm{sol}(A,B)$. Si $\gamma(A,B)(w) = \mathrm{sol}(E,F)$ alors, $\forall l \in \arg(\mathrm{sol}(E,F)), l \in \mathcal{L}^+, \gamma(A,B)(w.l) = \mathrm{sol}(C{\downarrow}(E)/l, C{\uparrow}(F)/l)$ et si $l \in \mathcal{L}^-$, $\gamma(A,B)(w.l) = \mathrm{sol}(C{\uparrow}(F)/l, C{\downarrow}(E)/l)$. Par induction, et utilisant le lemme 3.5, on vérifie que $\gamma(A,B)$ est un type. On pose la substitution $\rho(\alpha) = \gamma(\{\alpha\},\{\alpha\})$

Reste à vérifier que $\rho \models C$. On montre par récurrence que $\forall n \in \mathbb{N}, \forall t_1 \leq t_2 \in C, \rho(t_1) \leq_n \rho(t_2)$ et que pour tout $A,B,E,F \in \mathrm{VAR}(C)$ tels que $C{\downarrow}(A) \subseteq C{\downarrow}(E)$, $C{\uparrow}(F) \subseteq C{\uparrow}(B)$ et $\mathrm{sol}(A,B)$ et $\mathrm{sol}(E,F)$ sont définis, $\gamma(A,B) \leq_n \gamma(E,F)$.

Le cas $n = 0$ est trivialement vrai. Supposons à présent que l'hypothèse est vrai pour le cas n.

Considérons le cas $t_1 = \gamma(A,B) \leq_{n+1} \gamma(E,F) = t_2$. Par le lemme 3.6, $k_1 = \gamma(A,B)(\epsilon) \leq_{\mathcal{K}} \gamma(E,F)(\epsilon) = k_2$. Soit $l \in \arg(k_1) \cap \arg(k_2)$, $l \in \mathcal{L}^+$. Comme $C{\downarrow}(A) \subseteq C{\downarrow}(E)$, $C{\downarrow}(A)/l \subseteq C{\downarrow}(E)/l$ et donc $C{\downarrow}(C{\downarrow}(A)/l) \subseteq C{\downarrow}(C{\downarrow}(E)/l)$. De même $C{\uparrow}(C{\uparrow}(F)/l) \subseteq C{\uparrow}(C{\uparrow}(B)/l)$. Donc, par récurrence, $t_1/l = \gamma(C{\downarrow}(A)/l, C{\uparrow}(B)/l) \leq_n \gamma(C{\downarrow}(E)/l, C{\uparrow}(F)/l) = t_2/l$. Si $l \in \mathcal{L}^-$, comme $C{\downarrow}(A) \subseteq C{\downarrow}(E)$, $C{\downarrow}(A)/l \subseteq C{\downarrow}(E)/l$ et donc $C{\uparrow}(C{\downarrow}(A)/l) \subseteq C{\uparrow}(C{\downarrow}(E)/l)$. De même $C{\downarrow}(C{\uparrow}(F)/l) \subseteq C{\downarrow}(C{\uparrow}(B)/l)$. Donc, par récurrence, $t_2/l = \gamma(C{\uparrow}(F)/l, C{\downarrow}(E)/l) \leq_n \gamma(C{\uparrow}(B)/l, C{\downarrow}(A)/l) = t_1/l$. D'où $t_1 \leq_{n+1} t_2$.

Considérons $\rho(\alpha) \leq \rho(\beta)$. Comme $\alpha \leq \beta \in C$ et comme C est clos, on a $C{\downarrow}(\alpha) \subseteq C{\downarrow}(\beta)$ et $C{\uparrow}(\beta) \subseteq C{\uparrow}(\alpha)$. On peut donc appliquer le résultat ci-dessus pour obtenir $\rho(\alpha) = \gamma(\{\alpha\},\{\alpha\}) \leq_{n+1} \gamma(\{\beta\},\{\beta\}) = \rho(\beta)$.

Considérons $\alpha \leq t$, avec $t \notin \mathcal{V}$. On a $\rho(\alpha)(\epsilon) = \mathrm{sol}(\{\alpha\},\{\alpha\}) \leq_{\mathcal{K}} \mathrm{bi}_{\mathcal{K}}(\{t'(\epsilon) \mid t' \in C{\uparrow}(\alpha)\}) \leq_{\mathcal{K}} t(\epsilon)$ car $t \in C{\uparrow}(\alpha)$. Soit $l \in \arg(\rho(\alpha)(\epsilon)) \cap \arg(t(\epsilon))$, $l \in \mathcal{L}^+$. On a $t/l = \beta$ pour un certain β, $\rho(t)/l = \gamma(\{\beta\},\{\beta\})$ et $\rho(\alpha)/l = \gamma(C{\downarrow}(\alpha)/l, C{\uparrow}(\alpha)/l)$. On a $\beta \in C{\uparrow}(\alpha)/l$, donc $C{\uparrow}(\beta) \subseteq C{\uparrow}(C{\uparrow}(\alpha)/l)$. Comme C est clos, on a $\forall \alpha' \in C{\downarrow}(\alpha)/l, \alpha' \leq \beta \in C$. Comme C est clos, on a $C{\downarrow}(C{\downarrow}(\alpha)/l) \subseteq C{\downarrow}(\beta)$. On obtient donc $\rho(\alpha)/l = \gamma(C{\downarrow}(\alpha)/l, C{\uparrow}(\alpha)/l) \leq_n \gamma(\{\beta\},\{\beta\}) = \rho(t)/l$. Si $l \in \mathcal{L}^-$, on a $\rho(\alpha)/l = \gamma(C{\uparrow}(\alpha)/l, C{\downarrow}(\alpha)/l)$. On a $\beta \in C{\downarrow}(\alpha)/l$, donc $C{\downarrow}(\beta) \subseteq C{\downarrow}(C{\downarrow}(\alpha)/l)$. Comme C est clos, $\forall \alpha' \in C{\uparrow}(\alpha)/l, \beta \leq \alpha' \in C$ et $C{\uparrow}(C{\uparrow}(\alpha)/l) \subseteq C{\uparrow}(\beta)$. Donc $\rho(t)/l = \gamma(\{\beta\},\{\beta\}) \leq_n \gamma(C{\uparrow}(\alpha)/l, C{\downarrow}(\alpha)/l) = \rho(\alpha)/l$. Donc $\rho(\alpha) \leq_{n+1} \rho(t)$

On traite de manière similaire le cas $t \leq \alpha$.

Reste le cas $t_1 \leq t_2$ avec $t_1, t_2 \notin \mathrm{VAR}(C)$. Comme C est clos $t_1(\epsilon) \leq_{\mathcal{K}} t_2(\epsilon)$. Soit $l \in \arg(t_1(\epsilon)) \cap \arg(t_2(\epsilon))$, $l \in \mathcal{L}^+$. Comme C est clos, en posant $\alpha = t_1/l$ et $\beta = t_2/l$, on a $\alpha \leq \beta \in C$ et par récurrence, $\rho(t_1)/l = \rho(\alpha) \leq_n \rho(\beta) = \rho(t_2)/l$. De même si $l \in \mathcal{L}^-$, on a $\rho(t_2)/l \leq_n \rho(t_1)/l$. Donc $\rho(t_1) \leq_{n+1} \rho(t_2)$.

Donc pour tout $n \in \mathbb{N}$, pour tout $t_1 \leq t_2 \in C$, $\rho(t_1) \leq_n \rho(t_2)$, c'est-à-dire $\rho(t_1) \leq_{\mathcal{T}} \rho(t_2)$. Donc $\rho \models C$. □

3.2 Algorithme de clôture

Théorème 3.8 (Clôture) *Soit $C = C^0$ un système de contraintes pré-clos. Soit*

$$C^{n+1} = C^n \cup \bigcup_{c \in C} \mathrm{subc}(c) \cup \bigcup_{\{t \leq \alpha, \alpha \leq t'\} \subseteq C} \mathrm{subc}(t \leq t')$$

Si la suite $C, C^1, C^2, \ldots$ est infinie, alors elle atteint un point fixe C^∞, qui est le plus petit système de contraintes clos contenant C. Son ensemble de solutions est égal à celui de C et C^∞ est satisfiable. Sinon C n'a pas de solution. On appelle C^∞ la clôture de C.

Démonstration : Pour un C^n quelconque, il est clair que C^{n+1} est équivalent à C^n s'il est défini et que C^n n'a pas de solution dans le cas contraire (c'est-à-dire si on applique essaie d'appliquer subc sur une contrainte $t_1 \leq t_2$ alors que $t_1(\epsilon)\not\leq_{\mathcal{K}}t_2(\epsilon)$, ce qui signifie que $t_1 \leq t_2$ n'est pas satisfiable). Donc si un élément de la suite n'est pas défini, C n'est pas satisfiable. Sinon, la suite doit atteindre un point fixe C^∞ car toute les contraintes ajoutées dans C^{n+1} utilisent des termes dans C^n, c'est-à-dire dans C, et qu'il n'y a q'un nombre fini des telles contraintes. Il est clair que C^∞ est clos et donc satisfiable. □

L'algorithme ci-dessus requiert un que le système de contraintes auquel il est appliqué soit pré-clos. Cette condition est automatiquement remplie dans [5, 6], de par l'existence des types $\bot$ et $\top$ respectivement plus petit et plus grand que tous les autres types. Dans ce cas, il suffit en effet d'ajouter les contraintes $\bot \leq \alpha$ et $\alpha \leq \top$ à C pour le rendre pré-clos, sans changer son ensemble de solutions. Nous énonçons dans le théorème 3.9 des conditions suffisantes sur $\mathcal{K}$ avec lesquelles on peut décider de la satisfiabilité d'un système de contraintes non-préclos :

Théorème 3.9 (Pré-clotûre) *Soit $\mathcal{K}max$ l'ensemble des maximums de $\mathcal{K}$ et $\mathcal{K}min$ l'ensemble de ses minimums. Si $\mathcal{K}$ vérifie les conditions suivantes :*

1. $\forall k \in \mathcal{K}max \cup \mathcal{K}min, \arg(k) = \emptyset$
2. *Pour tout $k \in \mathcal{K}$, il existe $k_1 \in \mathcal{K}min$ et $k_2 \in \mathcal{K}max$ tels que $k_1 \leq_{\mathcal{K}} k \leq_{\mathcal{K}} k_2$.*

alors pour tout système de contraintes C on définit l'ensemble de ses pré-clotûres $\mathrm{pc}(C)$ *comme :*

$$\mathrm{pc}(C) = \left\{ C \cup \bigcup_{\alpha \in \mathrm{VAR}(C)} \{t_\alpha \leq \alpha, \alpha \leq t'_\alpha\} \mid t_\alpha(\epsilon) \in \mathcal{K}min, t'_\alpha(\epsilon) \in \mathcal{K}max \right\}$$

Tous les éléments de $\mathrm{pc}(C)$ *sont clos et l'union de leurs ensembles de solutions est égal à l'ensemble des solutions de C.*

Démonstration : Comme $\forall k \in \mathcal{K}max \cup \mathcal{K}min, \arg(k) = \emptyset$, pour tout $C' \in \mathrm{pc}(C), \mathrm{VAR}(C') = \mathrm{VAR}(C)$. Donc, par construction, les éléments de $\mathrm{pc}(C)$ sont pré-clos. Pour tout $C' \in \mathrm{pc}(C)$, on a $C \subseteq C'$, donc $\rho \models C' \Rightarrow \rho \models C$. Reste à montrer que si $\rho \models C$ alors il existe $C' \in \mathrm{pc}(C)$ tels que $\rho \models C'$. Par la condition 2) tout $\alpha \in \mathrm{VAR}(C)$, on peut trouver $k_\alpha \in \mathcal{K}min$ et $k'_\alpha \in \mathcal{K}max$ tels que $\rho \models t \leq \alpha, \alpha \leq t'$ avec $t(\epsilon) = k_\alpha$ et $t'(\epsilon) = k'_\alpha$. Donc il existe $C' \in \mathrm{pc}(C)$ tel que $\rho \models C'$. Donc l'union des ensembles de solutions des éléments de $\mathrm{pc}(C)$ est égal à l'ensemble des solutions de C. □

Si $\mathcal{K}max$ et $\mathcal{K}min$ sont finis, il est possible d'énumérer les éléments de $\mathrm{pc}(C)$. Comme ces éléments sont clos, ont peut tester leur satisfiabilité avec l'algorithme du théorème 3.8. Cela nous donne un algorithme pour tester la satisfiabilité des systèmes de contraintes non clos. La complexité du test de satisfiabilité devient alors en $O(n^3 k_{min}^v k_{max}^v)$ où n est la taille du système de contraintes, k_{min} est la taille de $\mathcal{K}min$ et k_{max} celle de $\mathcal{K}max$, et v est le nombre de variables non bornées.

4 Calcul explicite de solutions

Dans [4], Pottier décrit un algorithme de simplification des ensembles contraintes de sous-typage dans les treillis permettant de calculer des bornes pour les variables apparaissant dans les contraintes, et de calculer des solutions par identification des variables avec ces bornes. Nous étendons cet algorithme au cas des quasi-treillis.

Nous supposerons, comme dans la section 3 que les contraintes sont formées de petits termes. Pour résoudre un système de contraintes C dans un quasi-treillis $\mathcal{T}(\mathcal{K})$, nous complétons $\mathcal{K}$ en un treillis $\mathcal{K}^{\perp,\top}$ en lui ajoutant $\perp$ et $\top$, avec $\forall k \in \mathcal{K}, \perp \leq_{\mathcal{K}^{\perp,\top}} k \leq_{\mathcal{K}^{\perp,\top}} \top$ et $ar(\top) = \arg(\perp) = \emptyset$. Nous résolvons ensuite C dans $\mathcal{T}(\mathcal{K}^{\perp,\top})$ avec l'algorithme de Pottier, obtenant un système de contraintes C' équivalent à C et simplifié. Nous appliquons ensuite un ensemble de règles sur C', obtenant ainsi un système de contraintes dans lesquelles $\perp$ et $\top$ n'apparaissent plus.

Nous décrivons la forme du système de contraintes C' résultant de l'application de l'algorithme sur un ensemble de contraintes de sous-typage C dans le treillis $\mathcal{T}(\mathcal{K}^{\perp,\top})$. Lors de l'application de l'algorithme de Pottier, certaines variables ont été ajoutées. Elle représentent les "bornes" supérieures et inférieures de certains ensembles de variables de C. Nous nommerons γ_A la variable représentant la borne inférieure de A et λ_A celle représentant sa borne supérieure. On dira que les variables de C sont des variables "originelles" et que autres sont des variables "introduites". C' vérifie les propriétés indiquées dans la figure 1 :

1. Pour tout $\alpha \in \mathrm{VAR}(C')$ $\exists! t \notin \mathrm{VAR}(C')$ tel que $t \leq \alpha \in C'$ et $\exists! t' \notin \mathrm{VAR}(C')$ tel que $\alpha \leq t' \in C'$. On notera $t = C'{\downarrow}(\alpha)$ et $t' = C'{\uparrow}(\alpha)$.
2. Pour tout $\{\alpha \leq \beta, \beta \leq \delta\} \subseteq C'$, $\alpha \leq \delta \in C'$.
3. Pour tout $\alpha \in \mathrm{VAR}(C')$, $\forall l \in \arg(C'{\uparrow}(\alpha)(\epsilon))$, soit $l \in \mathcal{L}^+$ et $\exists A, t/l = \gamma_A$, soit $l \in \mathcal{L}^-$ et $\exists A, t/l = \lambda_A$. $\forall l \in \arg(C'{\downarrow}(\alpha)(\epsilon))$, soit $l \in \mathcal{L}^+$ et $\exists A, t/l = \lambda_A$, soit $l \in \mathcal{L}^-$ et $\exists A, t/l = \gamma_A$.
4. Si $\alpha \leq \beta \in C'$, ou si $\alpha \equiv \gamma_A$ et $\beta \equiv \gamma_B$ avec $B \subseteq A$ alors $k_\alpha = C'{\uparrow}(\alpha)(\epsilon) \leq_{\mathcal{K}^{\perp,\top}} k_\beta = C'{\uparrow}(\beta)(\epsilon)$ et pour tout $l \in \arg(k_\alpha) \cap \arg(k_\beta)$, si $l \in \mathcal{L}^+$, $C'{\uparrow}(\alpha)/l = \gamma_E$, $C'{\uparrow}(\beta)/l = \gamma_F$, alors $F \subseteq E$. Si $l \in \mathcal{L}-$, $C'{\uparrow}(\alpha)/l = \lambda_E$, $C'{\uparrow}(\beta)/l = \lambda_F$, alors $F \subseteq E$.
5. Si $\alpha \leq \beta \in C'$, ou si $\alpha \equiv \lambda_A$ et $\beta \equiv \lambda_B$ avec $A \subseteq B$ alors $k_\alpha = C'{\downarrow}(\alpha)(\epsilon) \leq_{\mathcal{K}^{\perp,\top}} k_\beta = C'{\downarrow}(\beta)(\epsilon)$ et pour tout $l \in \arg(k_\alpha) \cap \arg(k_\beta)$, si $l \in \mathcal{L}^+$, $C'{\downarrow}(\alpha)/l = \lambda_E$, $C'{\downarrow}(\beta)/l = \lambda_F$, alors $E \subseteq F$. Si $l \in \mathcal{L}-$, $C'{\downarrow}(\alpha)/l = \gamma_E$, $C'{\downarrow}(\beta)/l = \gamma_F$, alors $E \subseteq F$.
6. $\forall \gamma_A, C'{\uparrow}(\gamma_A) \neq \top$ et $\forall \lambda_A, C'{\downarrow}(\lambda_A) \neq \perp$.
7. Pour tout $\alpha \in \mathrm{VAR}(C)$, $\mathrm{subc}(C'{\downarrow}(\alpha) \leq C'{\uparrow}(\alpha))$ est défini et inclus dans C'.
8. $\forall t, t'$, $t \leq t' \notin C'$

FIG. 1 – *Propriétés vérifiées par le resultat de l'algorithme de Pottier.*

Il est clair que C' est clos donc satisfiable dans $\mathcal{T}(\mathcal{K}^{\perp,\top})$.

La figure 2 présente les règles permettant le calcul des bornes des solutions à C dans le quasi-treillis. D désigne un système de contraintes.

(Down $\perp$)	$D,\gamma_A \leq \perp,\alpha \leq t \rightarrow D,\gamma_A \leq \perp,\alpha \leq t'$ Si $t/l = \gamma_A$. $t'(\epsilon) = \mathrm{bs}_{\mathcal{K}}(\mathrm{Mi}_{\mathcal{K},\mathcal{L}}(t(\epsilon),\arg(t(\epsilon)) \setminus \{l\}))$ si elle est définie, $t'(\epsilon) = \perp$ sinon. Pour tout $l' \in t'(\epsilon)$, $t'/l' = t/l'$.
(Down $\top$)	$D,\top \leq \lambda_A,\alpha \leq t \rightarrow D,\top \leq \lambda_A,\alpha \leq t'$ Si $t/l = \lambda_A$. $t'(\epsilon) = \mathrm{bs}_{\mathcal{K}}(\mathrm{Mi}_{\mathcal{K},\mathcal{L}}(t(\epsilon),\arg(t(\epsilon)) \setminus \{l\}))$ si elle est définie, $t'(\epsilon) = \perp$ sinon. Pour tout $l' \in t'(\epsilon)$, $t'/l' = t/l'$.
(Up $\top$)	$D,\top \leq \lambda_A,t \leq \alpha \rightarrow D,\top \leq \lambda_A,t' \leq \alpha$ Si $t/l = \lambda_A$. $t'(\epsilon) = \mathrm{bi}_{\mathcal{K}}(\mathrm{Ma}_{\mathcal{K},\mathcal{L}}(t(\epsilon),\arg(t(\epsilon)) \setminus \{l\}))$ si elle est définie, $t'(\epsilon) = \perp$ sinon. Pour tout $l' \in t'(\epsilon)$, $t'/l' = t/l'$.
(Up $\perp$)	$D,\gamma_A \leq \perp,t \leq \alpha \rightarrow D,\gamma_A \leq \perp,t' \leq \alpha$ Si $t/l = \gamma_A$. $t'(\epsilon) = \mathrm{bi}_{\mathcal{K}}(\mathrm{Ma}_{\mathcal{K},\mathcal{L}}(t(\epsilon),\arg(t(\epsilon)) \setminus \{l\}))$ si elle est définie, $t'(\epsilon) = \perp$ sinon. Pour tout $l' \in t'(\epsilon)$, $t'/l' = t/l'$.

FIG. 2 – *Règles de calcul des bornes dans le quasi-treillis*

Propriete 4.1 *L'application des règles de la figure 2 préservent les solutions dont le co-domaine est inclus dans $\mathcal{T}(\mathcal{K}) \cup \{\perp,\top\}$.*

Démonstration : Condidérons le cas (Down $\perp$), les autres cas étant similaires.

Supposons $\rho(\alpha) \models D,\alpha \leq t,\gamma_A \leq \perp$ avec $\mathrm{dom}(\rho) \subseteq \mathcal{T}(\mathcal{K}) \cup \{\perp,\top\}$ et montrons que $\rho \models D,\gamma_A \leq \perp,\alpha \leq t'$. Clairement, $\rho(\gamma_A) = \perp$. Comme pour tout $l' \in t'(\epsilon),t/l' = t'/l'$, il suffit de montrer que $\rho(\alpha)(\epsilon) \leq_{\mathcal{K}} t'(\epsilon)$. On a $t/l = \gamma_A$, donc $\rho(t)/l = \perp$. Comme il n'existe pas de type dans $\mathcal{T}(\mathcal{K})$ qui soit plus petit que $\perp$, et par la propriété 2.9, on a $l \notin \arg(\rho(\alpha)(\epsilon))$. Donc $\rho(\alpha)(\epsilon) \leq_{\mathcal{K}^{\perp,\top}} \mathrm{bs}_{\mathcal{K}}(\mathrm{Mi}_{\mathcal{K},\mathcal{L}}(t(\epsilon),\arg(t(\epsilon)) \setminus \{l\}))$ si elle existe et $\rho(\alpha)(\epsilon) \leq_{\mathcal{K}^{\perp,\top}} \perp$ sinon. Donc $\rho(\alpha)(\epsilon) \leq_{\mathcal{K}^{\perp,\top}} t'(\epsilon)$. □

Propriete 4.2 *Le système de réécriture décrit par les règles de la figure 2 termine et est convergent.*

Démonstration : De par la propriété 2.6, si $D,\alpha \leq t \rightarrow D,\alpha \leq t'$ alors $\arg(t'(\epsilon)) \subset \arg(t(\epsilon))$, donc On ne peut appliquer la règles (Down $\perp$) et (Down $\top$) qu'un nombre fini de fois par variable. De même, ne peut appliquer la règles (Up $\perp$) et (Up $\top$) qu'un nombre fini de fois par variable. Comme aucune règle ne fait augmenter l'arité des bornes des variables le système de réécriture termine, en $O(n)$ étapes où n est la taille du système de contraintes. La convergence se vérifie simplement en remarquant que $\mathrm{bs}_{\mathcal{K}}(\mathrm{Mi}_{\mathcal{K},\mathcal{L}}(\mathrm{bs}_{\mathcal{K}}(\mathrm{Mi}_{\mathcal{K},\mathcal{L}}(k,L)),L')) = \mathrm{bs}_{\mathcal{K}}(\mathrm{Mi}_{\mathcal{K},\mathcal{L}}(k,L \cap L')) = \mathrm{bs}_{\mathcal{K}}(\mathrm{Mi}_{\mathcal{K},\mathcal{L}}(\mathrm{bs}_{\mathcal{K}}(\mathrm{Mi}_{\mathcal{K},\mathcal{L}}(k,L')),L))$. □

Propriete 4.3 *Soit C' le résultat de l'algorithme de Pottier appliqué à C, et soit $C' \rightarrow^* C'' \not\rightarrow$. Alors C'' vérifie les propriétés de la figure 1.*

Démonstration : On vérifie que les propriétés 1), 2), 3), 6), 7) et 8) sont conservées lors de l'application des règles de la figure 2. On montre ensuite si l'application d'une règle brise la propriété 4) ou 5), il est possible de la rétablir en ré-appliquant cette règle. Comme $\rightarrow^*$ est convergent, on obtient que C'' vérifie les propriétés 4) et 5). □

Théorème 4.4 *Soit C' le résultat de l'algorithme de Pottier appliqué à C, et soit $C' \rightarrow^* C'' \not\rightarrow$. Si $\mathcal{L}^- = \emptyset$ alors les $C''\uparrow(\alpha)$ (resp. $C''\downarrow(\alpha)$) de C'' définissent une solution maximale (resp. minimale) de C dans $\mathcal{T}(\mathcal{K})$.*

Démonstration : Soit le système d'équations suivant : $\{\alpha = t \mid \alpha \leq t \in C''\}$. Ce système admet une solution unique ρ car chaque variable n'apparaît qu'une fois à gauche du signe $=$. Montrons que ρ est une solution de C''. Pour cela on montre par récurrence que pour tout $n \in \mathbb{N}$, pour tout $t \leq t' \in C''$, $\rho(t) \leq_n \rho(t')$, que si $A \subseteq B$, $\rho(\gamma_B) \leq_n \rho(\gamma_A)$. Le cas $n = 0$ est trivalement vérifié. Soit $\alpha \leq \beta \in C''$. Par la propriété 4.3, C'' vérifie les propriétés de la figure 1. Donc $\rho(\alpha)(\epsilon) = C''\uparrow(\alpha)(\epsilon) \leq_{\mathcal{K}} C''\uparrow(\beta)(\epsilon) = \rho(\alpha)(\epsilon)$. Soit $l \in \arg(\rho(\alpha)(\epsilon)) \cap \arg(\rho(\beta)(\epsilon))$. On a $\rho(\alpha)/l = \gamma_A$ pour un certain A et $\rho(\beta)/l = \gamma_B$ pour un certain B avec $B \subseteq A$. Par récurrence $\rho(\gamma_A) \leq_n \rho(\gamma_B)$. Donc $\rho(\alpha) \leq_{n+1} \rho(\beta)$. De la même manière, si $A \subseteq B$, $\rho(\gamma_B) \leq_{n+1} \rho(\gamma_A)$. Soit $\alpha \leq t \in C''$. On a $\rho(\alpha) = \rho(t)$ donc $\rho(\alpha) \leq_{n+1} \rho(t)$. Soit $t \leq \alpha \in C''$. Comme C'' vérifie la propriété 7) de la figure 1, on a $\rho(t)(\epsilon) \leq_{\mathcal{K}} \rho(\alpha)(\epsilon)$. De plus, pour tout $l \in \arg(rho(t)(\epsilon)) \cap \arg(\rho(\alpha)(\epsilon))$, $t/l \leq C''\uparrow(\alpha) \in C$, d'où $\rho(t)/l \leq_n \rho(\alpha)/l$. Donc $\rho(t) \leq_{n+1} \rho(\alpha)$. Donc pour tout $t \leq t' \in C''$, $\rho(t) \leq_{\mathcal{T}} \rho(t')$, donc $\rho \models C''$. De plus, comme $\mathcal{L}^- = \emptyset$, il n'y a que des variables de la forme γ_A qui apparaissent à droite dans les équation du système qui définit ρ, donc, en utilisant la propriété 6) de la figure 1 et le fait que C soit pré-clos avec des bornes dans $\mathcal{K}$, on obtient que pour tout $\alpha \in \mathrm{VAR}(C)$, $\rho(\alpha) \in \mathcal{T}(\mathcal{K})$. On vérifie ensuite par récurrence que pour tout $\rho' \models C''$, $\forall \alpha \in \mathrm{VAR}(C''), \rho'(\alpha) \leq_{\mathcal{T}} \rho(\alpha)$. ρ est donc une solution maximale de C dans $\mathcal{T}(\mathcal{K})$. □

Notons que dans le cas ou il existe des constructeurs de type avec des étiquettes contravariantes (dans $\mathcal{L}^-$), il n'y a pas forcément de solution maximale. Par exemple, prenons $\mathcal{K} = \{int, float, \rightarrow\}$ avec $int \leq float$, $\mathcal{L}^+ = \{r\}$, $\mathcal{L}^- = \{a\}$, $\arg(int) = \arg(float) = \emptyset$, $\arg(\rightarrow) = \{a,r\}$. Soit $C = \{\alpha \rightarrow \alpha \leq \beta, \beta \leq \alpha \rightarrow \alpha, int \leq \alpha, \alpha \leq float\}$. C est pré-clos et a deux solutions : $\rho(\beta) = int \rightarrow int, \rho(\alpha) = int$ et $\rho'(\beta) = float \rightarrow float, \rho'(\alpha) = float$, et ces deux solutions sont incomparables.

En combinant l'algorithme de simplification de Pottier avec les règles de la figure 1, nous obtenons donc un algorithme de simplification pour les systèmes pré-clos dans les quasi-treillis. De plus, dans le cas où tous les constructeurs de types sont covariants ($\mathcal{L}^- = \emptyset$), il fournit des solutions minimales et maximales. Grâce au théorème 3.9, on peut obtenir un ensemble de solutions maximales et minimales pour les systèmes non pré-clos.

5 Applications

Une première application de l'algorithme de résolution des contraintes de sous-typage dans les quasi-treillis est notre système TCLP [2, 1] de typage des programmes logique avec contraintes. Cet algorithme permet de s'affranchir de l'élément $\perp$ dans la structure des types. Il permet également de disposer de structures de types plus complexes que celles utilisées jusqu'à présent, notament en autorisant des structures de types privées du type *term* de méta-programmation. Nous avons implanté en CHR [**?**] dans TCLP l'algorithme de résolution des contraintes par une simple modification du précédent solveur de contraintes de TPLP dans les treillis [**?**].

Une seconde application se situe dans le cadre de l'inférence de types avec sous-typage dans les langages à la ML. Dans [4] Pottier utilise les contraintes de sous-typage dans les treillis pour l'inférence de type dans un langage à la ML avec rangées. Cependant, dans un treillis, le type inféré pour une fonction peut être de la forme $\perp \rightarrow \tau$, ce qui signifie que cette fonction n'est applicable à aucun argument. L'algorithme de résolution de contraintes décrit dans ce papier permet d'utiliser comme structure de types le quasi-treillis obtenu en supprimant l'élément $\perp$ du treillis. Ceci permet de produire une erreur lors du typage des fonctions de la forme précédente.

6 Conclusion

Dans cet article nous avons défini des relations de sous-typage non structurel très générales dans l'ensemble des types infinis (réguliers) définis sur un quasi-treillis de constructeurs, et avons montré que cet ensemble ainsi ordonné a une structure de quasi-treillis. Nous avons généralisé l'algorithme de Trifonov et Smith de test de satisfiabilité des contraintes de sous-typage dans les treillis, au cas des quasi-treillis avec une complexité en $O(k_{min}^v * k_{max}^v * n^3)$ où k_{min} (resp. k_{max}) est le nombre d'éléments minimaux (resp. maximaux) du quasi-treillis et v le nombre de variables non bornées. La complexité de cet algorithme est donc inchangée en $O(n^3)$ pour les systèmes dont toutes les variables sont données avec des bornes. Nous avons également étendu l'algorithme de Pottier de calcul de solutions au cas des quasi-treillis, et montré que les solutions calculées sont minimales (resp. maximales) si tous les constructeurs sont covariants. Enfin, nous avons évoqué les applications de ces extensions au système de typage TCLP ainsi qu'à l'inférence de type dans des langages à la ML.

Références

[1] E. Coquery. `Tclp`: a generic type checker for constraint logic programs. `http://pauillac.inria.fr/~coquery/tclp/`.

[2] F. Fages and E. Coquery. Typing constraint logic programs. *Theory and Practice of Logic Programming*, 1, November 2001.

[3] F. Pottier. *Synthèse de types en présence de sous-typage: de la théorie à la pratique*. PhD thesis, Université Paris 7, July 1998.

[4] F. Pottier. A versatile constraint-based type inference system. *Nordic Journal of Computing*, 7(4):312–347, November 2000.

[5] F. Pottier. Simplifying subtyping constraints: a theory. *Information & Computation*, 170(2):153–183, November 2001.

[6] V. Trifonov and S. Smith. Subtyping constrained types. In *Proc. 3rd Int'l Symposium on Static Analysis*, number 1145 in LNCS, pages 349–365. Springer, 1996.

A Démonstrations de la section 2

Lemme A.1 *Soit $S \subseteq \mathcal{T}$ avec $S \neq \emptyset$.*
Si $t \in \mathrm{Mi}_{\mathcal{T}}(S)$, alors $t(\epsilon) \in \mathrm{Mi}_{\mathcal{K},\mathcal{L}}(\mathrm{bi}_{\mathcal{K}}(\{s(\epsilon) \mid s \in S\}),\mathrm{EUSous}(S))$. De même, si $t \in \mathrm{Ma}_{\mathcal{T}}(S)$, alors $t(\epsilon) \in \mathrm{Ma}_{\mathcal{K},\mathcal{L}}(\mathrm{bs}_{\mathcal{K}}(\{s(\epsilon) \mid s \in S\}),\mathrm{EUSur}(S))$.

Démonstration : On vérifie que $t(\epsilon) \leq_{\mathcal{K}} \mathrm{bi}_{\mathcal{K}}(\{s(\epsilon) \mid s \in S\})$ et que $\arg(t(\epsilon)) \cap \arg(\mathrm{bi}_{\mathcal{K}}(\{s(\epsilon) \mid s \in S\})) \subseteq \mathrm{EUSous}(S)$. □

Corolaire A.2 *Soit $S \subseteq \mathcal{T}$ avec $S \neq \emptyset$. Si $\mathrm{Mi}_{\mathcal{T}}(S) \neq \emptyset$ (resp. $\mathrm{Ma}_{\mathcal{T}}(S) \neq \emptyset$) alors $\mathcal{K}\mathrm{bi}(S)$ (resp. $\mathcal{K}\mathrm{bs}(S)$) est bien défini.*

Démonstration : Comme $\mathrm{Mi}_{\mathcal{T}}(S) \neq \emptyset$, il existe $t \in \mathrm{Mi}_{\mathcal{T}}(S)$ et par le lemme A.1 $t(\epsilon) \in \mathrm{Mi}_{\mathcal{K},\mathcal{L}}(\mathrm{bi}_{\mathcal{K}}(\{s(\epsilon) \mid s \in S\}),\mathrm{EUSous}(S))$. Donc $\mathrm{Mi}_{\mathcal{K},\mathcal{L}}(\mathrm{bi}_{\mathcal{K}}(\{s(\epsilon) \mid s \in S\}),\mathrm{EUSous}(S)) \neq \emptyset$ et par définition, il est majoré. Donc il admet une borne supérieure $\mathcal{K}\mathrm{bi}(S)$. □

Corolaire A.3 *Soit $S \subseteq \mathcal{T}$ avec $S \neq \emptyset$. Si $\mathrm{Mi}_{\mathcal{T}}(S) \neq \emptyset$ (resp. $\mathrm{Ma}_{\mathcal{T}}(S) \neq \emptyset$) alors pour tout $n \in \mathbb{N}$, $\mathrm{bi}_{\mathcal{T}n}(S)$ (resp. $\mathrm{bs}_{\mathcal{T}n}(S)$) est bien défini.*

Démonstration : On le démontre par récurrence sur n : le cas $n = 0$ est trivial. Pour le cas $n+1$, on considère $\mathrm{bi}_{\mathcal{T}n+1}(S)$ (le cas de $\mathrm{bs}_{\mathcal{T}n+1}(S)$ est similaire). Par le corolaire A.2, $\mathcal{K}\mathrm{bi}(S)$ est bien défini. Soit $l \in \arg(\mathcal{K}\mathrm{bi}(S))$, par la propriété 2.6 $l \in \mathrm{ESMi}_{\mathcal{K},\mathcal{L}}(k,\mathrm{EUSous}(S)) \subseteq \mathrm{EUSous}(S)$. Donc S/l est minoré (ou majoré selon le signe de l). Donc, par récurrence, $\mathrm{bi}_{\mathcal{T}n+1}(S)/l$ est bien défini. □

Lemme A.4 *Soit $S \subseteq \mathcal{T}$ avec $S \neq \emptyset$. $\forall t \in \mathcal{T}$, si $t \in \mathrm{Mi}_{\mathcal{T}}(S)$ (resp. $t \in \mathrm{Ma}_{\mathcal{T}}(S)$) alors $t(\epsilon) \leq_{\mathcal{K}} \mathcal{K}\mathrm{bi}(S)$ (resp. $\mathcal{K}\mathrm{bs}(S) \leq_{\mathcal{K}} t(\epsilon)$) .*

Démonstration : Soit $t \in \mathrm{Mi}_{\mathcal{T}}(S)$.
Par le lemme A.1, on a $k \in \mathrm{Mi}_{\mathcal{K},\mathcal{L}}(\mathrm{bi}_{\mathcal{K}}(\{s(\epsilon) \mid s \in S\}),\mathrm{EUSous}(S))$. Par définition de $\mathcal{K}\mathrm{bi}(S)$, on a donc $k \leq_{\mathcal{K}} \mathcal{K}\mathrm{bi}(S)$.
La démonstration est similaire pour $\mathcal{K}\mathrm{bs}(S)$. □

Corolaire A.5 *Soit $S \subseteq \mathcal{T}$ avec $S \neq \emptyset$. $\forall t \in \mathrm{Mi}_{\mathcal{T}}(S)$ (resp. $t \in \mathrm{Ma}_{\mathcal{T}}(S)$), $\forall n \in \mathbb{N}$ $t \leq_n \mathrm{bi}_{\mathcal{T}n}(S)$ (resp. $\mathrm{bi}_{\mathcal{T}n}(S) \leq_n t$).*

Démonstration : Par récurrence sur n. Le cas $n = 0$ est trivial. Considérons $t \leq_{n+1} \mathrm{bi}_{\mathcal{T}n+1}(S)$. Comme $t \in \mathrm{Mi}_{\mathcal{T}}(S)$ avec $t(\epsilon) = k$ alors, par le lemme A.4, $k \leq_{\mathcal{K}} \mathcal{K}\mathrm{bi}(S)$. Soit $l \in \arg(t(\epsilon)) \cap \arg(\mathcal{K}\mathrm{bi}(S))$. Si $l \in \mathcal{L}^+$, alors , comme $t \in \mathrm{Mi}_{\mathcal{T}}(S)$, par la propriété 2.9, $t/l \in \mathrm{Mi}_{\mathcal{T}}(S/l)$. Par recurrence, on obtient $t/l \leq_n \mathrm{bi}_{\mathcal{T}n}(S/l) = \mathrm{bi}_{\mathcal{T}n+1}(S)/l$. Le cas $l \in \mathcal{L}^-$, est similaire. On a donc $t \leq_{n+1} \mathrm{bi}_{\mathcal{T}n+1}(S)$. De même, si $t \in \mathrm{Ma}_{\mathcal{T}}(S)$, alors $t \geq_{n+1} \mathrm{bs}_{\mathcal{T}n+1}(S)$. □

Lemme A.6 *Soit $S \subseteq \mathcal{T}$ avec $S \neq \emptyset$. Si $\mathrm{Mi}_{\mathcal{T}}(S) \neq \emptyset$ (resp. $\mathrm{Ma}_{\mathcal{T}}(S) \neq \emptyset$) alors $\forall s \in S, \mathcal{K}\mathrm{bi}(S) \leq_{\mathcal{K}} s(\epsilon)$ (resp. $\mathcal{K}\mathrm{bs}(S) \geq_{\mathcal{K}} s(\epsilon)$)*

Démonstration : Comme $\exists t \in \mathrm{Mi}_{\mathcal{T}}(S)$, $\mathcal{K}\mathrm{bi}(S)$ est bien défini. On note $K_S = \{s(\epsilon) \mid s \in S\}$. On a $\mathcal{K}\mathrm{bi}(S) = \mathrm{bs}(\mathrm{Mi}_{\mathcal{K},\mathcal{L}}(\mathrm{bi}_{\mathcal{K}}(K_S),\mathrm{EUSous}(S))$.
De plus $\forall s \in S, \mathrm{bi}_{\mathcal{K}}(K_S) \leq_{\mathcal{K}} s(\epsilon)$. Or, par définition, $\mathcal{K}\mathrm{bi}(S) \leq_{\mathcal{K}} \mathrm{bi}_{\mathcal{K}}(K_S)$.
Donc $\forall s \in S, \mathcal{K}\mathrm{bi}(S) \leq_{\mathcal{K}} s$. La démonstration pour $\mathcal{K}\mathrm{bs}(S)$ est similaire. □

Démonstration de la propriété 2.14 : Par récurrence sur n. Le cas $n = 0$ est trivial. Soit $s \in S$. On note $k_s/L_s = s(\epsilon)$. Considérons $\mathrm{bi}_{\mathcal{T}n+1}(S) \leq_{n+1} s$. On a $\mathrm{bi}_{\mathcal{T}n+1}(S)(\epsilon) = \mathcal{K}\mathrm{bi}(S)$, et par le lemme A.6, $\mathcal{K}\mathrm{bi}(S) \leq_{\mathcal{K}} s(\epsilon)$. Soit $l \in \arg(\mathcal{K}\mathrm{bi}(S)) \cap \arg(s(\epsilon))$. Par la propriété 2.6 $l \in \mathrm{ESMi}_{\mathcal{K},\mathcal{L}}(k,\mathrm{EUSous}(S)) \subseteq \mathrm{EUSous}(S)$. Si $l \in \mathcal{L}^+$, $\mathrm{bi}_{\mathcal{T}n+1}(S)/l = \mathrm{bi}_{\mathcal{T}n}(S/l)$. On a $s/l \in S/l$. Donc, par récurrence, $\mathrm{bi}_{\mathcal{T}n+1}(S)/l = \mathrm{bi}_{\mathcal{T}n}(S/l) \leq_n s/l$. Le cas $l \in \mathcal{L}^-$ est symétrique. De même, $\mathrm{bs}_{\mathcal{T}n+1}(S) \geq_{n+1} s$. □

Le lemme suivant exprime que $\mathrm{bi}_{\mathcal{T}m}(S)$ est égal $\mathrm{bi}_{\mathcal{T}n}(S)$ jusqu'à la profondeur $\min(m,n)$.

Lemme A.7 *Soit $S \subseteq \mathcal{T}$ avec $S \neq \emptyset$ tel que $\mathrm{Mi}_{\mathcal{T}}(S) \neq \emptyset$ (resp. $\mathrm{Ma}_{\mathcal{T}}(S) \neq \emptyset$). Soit $m,n \in \mathbb{N}$, $m \leq n$, $\forall w \in \mathbb{N}^*$, si $\|w\| < m$ alors $w \in \mathrm{dom}(\mathrm{bi}_{\mathcal{T}m}(S)) \Leftrightarrow w \in \mathrm{dom}(\mathrm{bi}_{\mathcal{T}n}(S))$ et $\mathrm{bi}_{\mathcal{T}m}(S)(w) = \mathrm{bi}_{\mathcal{T}n}(S)(w)$ (resp. $w \in \mathrm{dom}(\mathrm{bs}_{\mathcal{T}m}(S)) \Leftrightarrow w \in \mathrm{dom}(\mathrm{bs}_{\mathcal{T}n}(S))$ et $\mathrm{bs}_{\mathcal{T}m}(S)(w) = \mathrm{bs}_{\mathcal{T}n}(S)(w)$).*

Démonstration : Par récurrence sur m : Le cas $n = 0$ est trivial car il n'y a aucun w tel que $\|w\| < 0$. Considérons le cas $m+1 \leq n+1$. Si $w = \epsilon$: Par définition on a $\epsilon \in \mathrm{dom}(\mathrm{bi}_{\mathcal{T}m+1}(S))$, $\epsilon \in \mathrm{dom}(\mathrm{bi}_{\mathcal{T}n+1}(S))$ et $\mathrm{bi}_{\mathcal{T}m+1}(S)(\epsilon) = \mathcal{K}\mathrm{bi}(S) = \mathrm{bi}_{\mathcal{T}n+1}(S)(\epsilon)$. Si $w = l.w'$: Comme $\mathrm{bi}_{\mathcal{T}m+1}(S)(\epsilon) = \mathcal{K}\mathrm{bi}(S) = \mathrm{bi}_{\mathcal{T}n+1}(S)(\epsilon)$ avec $\arg(\mathcal{K}\mathrm{bi}(S)) = L$, on a $l \in L$. Si $l \in \mathcal{L}^+$, alors $\mathrm{bi}_{\mathcal{T}m+1}(S)/l = \mathrm{bi}_{\mathcal{T}m}(S/l)$ et $\mathrm{bi}_{\mathcal{T}n+1}(S)/l = \mathrm{bi}_{\mathcal{T}n}(S/l)$. Par récurrence on a $w' \in \mathrm{dom}(\mathrm{bi}_{\mathcal{T}m}(S/l)) \Leftrightarrow w' \in \mathrm{dom}(\mathrm{bi}_{\mathcal{T}n}(S/l))$ Comme $l \in L$, on a donc $w = l.w' \in \mathrm{dom}(\mathrm{bi}_{\mathcal{T}m+1}(S)) \Leftrightarrow w \in \mathrm{dom}(\mathrm{bi}_{\mathcal{T}n+1}(S))$. Par récurrence, on a $\mathrm{bi}_{\mathcal{T}m}(S/l)(w') = \mathrm{bi}_{\mathcal{T}n}(S/l)(w')$, donc $\mathrm{bi}_{\mathcal{T}m+1}(S)(l.w') = (\mathrm{bi}_{\mathcal{T}m+1}(S)/l)(w') = \mathrm{bi}_{\mathcal{T}m}(S/l)(w') = \mathrm{bi}_{\mathcal{T}n}(S/l)(w') =$
$(\mathrm{bi}_{\mathcal{T}n+1}(S)/l)(w') = \mathrm{bi}_{\mathcal{T}n+1}(S)(l.w')$. Le cas $l \in \mathcal{L}^-$ est symétrique. La démonstration est similaire pour $\mathrm{bs}_{\mathcal{T}m+1}$ □

Le propriété suivante établit la validité de la définition 2.13 :

Propriete A.8 *Soit* $S \subseteq \mathcal{T}$ *avec* $S \neq \emptyset$*. Si* $\mathrm{Mi}_{\mathcal{T}}(S) \neq \emptyset$ *(resp.* $\mathrm{Ma}_{\mathcal{T}}(S) \neq \emptyset$*) alors* $\mathrm{bi}_{\mathcal{T}}(S)$ *(resp.* $\mathrm{bs}_{\mathcal{T}}(S)$*) est bien défini et est un type.*

Démonstration : Comme $\mathrm{Mi}_{\mathcal{T}}(S) \neq \emptyset$, par le corolaire A.3, pour tout $n \in \mathbb{N}$, $\mathrm{bi}_{\mathcal{T}n}(S)$ est bien défini. Reste à montrer que $\mathrm{bi}_{\mathcal{T}}(S)$ est un type :

- Soit $w.l \in \mathrm{dom}(\mathrm{bi}_{\mathcal{T}}(S))$. Alors $w.l \in \mathrm{dom}(\mathrm{bi}_{\mathcal{T}\|w.l\|+1})$. Donc $w \in \mathrm{dom}(\mathrm{bi}_{\mathcal{T}\|w.l\|+1})$. Par le lemme A.7 on obtient $w \in \mathrm{dom}(\mathrm{bi}_{\mathcal{T}\|w\|+1})$, d'où $w \in \mathrm{dom}(\mathrm{bi}_{\mathcal{T}}(S))$.
- $\epsilon \in \mathrm{dom}(\mathrm{bi}_{\mathcal{T}1}(S))$, donc $\epsilon \in \mathrm{dom}(\mathrm{bi}_{\mathcal{T}}(S))$
- Soit $w \in \mathrm{dom}(\mathrm{bi}_{\mathcal{T}}(S))$ tel que $\mathrm{bi}_{\mathcal{T}}(S)(w) = k/L = \mathrm{bi}_{\mathcal{T}\|w\|+1}(S)(w)$. On a $w.l \in \mathrm{dom}(\mathrm{bi}_{\mathcal{T}}(S)) \Leftrightarrow w.l \in \mathrm{dom}(\mathrm{bi}_{\mathcal{T}\|w.l\|+1}(S))$. Par le lemme A.7, $\mathrm{bi}_{\mathcal{T}\|w.l\|+1}(S)(w) = \mathrm{bi}_{\mathcal{T}\|w\|+1}(S)(w) = k/L$. Donc $w.l \in \mathrm{dom}(\mathrm{bi}_{\mathcal{T}\|w.l\|+1}(S)) \Leftrightarrow l \in L$.

□

Propriete A.9 *Soit* $S \subseteq \mathcal{T}$ *avec* $S \neq \emptyset$ *tel que* $\mathrm{Mi}_{\mathcal{T}}(S) \neq \emptyset$*.*
On a $\forall l \in \arg(\mathrm{bi}_{\mathcal{T}}(S)(\epsilon))$ *:*

- *Soit* $l \in \mathcal{L}^+$ *et* $\mathrm{bi}_{\mathcal{T}}(S)/l = \mathrm{bi}_{\mathcal{T}}(S/l)$
- *Soit* $l \in \mathcal{L}^-$ *et* $\mathrm{bi}_{\mathcal{T}}(S)/l = \mathrm{bs}_{\mathcal{T}}(S/l)$

De même si $\mathrm{Ma}_{\mathcal{T}}(S) \neq \emptyset$ *alors* $\forall l \in \arg(\mathrm{bs}_{\mathcal{T}}(S)(\epsilon))$ *:*

- *Soit* $l \in \mathcal{L}^+$ *et* $\mathrm{bi}_{\mathcal{T}}(S)/l = \mathrm{bs}_{\mathcal{T}}(S/l)$
- *Soit* $l \in \mathcal{L}^-$ *et* $\mathrm{bi}_{\mathcal{T}}(S)/l = \mathrm{bi}_{\mathcal{T}}(S/l)$

Démonstration : $(\mathrm{bi}_{\mathcal{T}}(S)/l)(w) = \mathrm{bi}_{\mathcal{T}}(S)(l.w) = \mathrm{bi}_{\mathcal{T}\|l.w\|+1}(S)(l.w)$. Si $l \in \mathcal{L}^+$ alors $\mathrm{bi}_{\mathcal{T}\|l.w\|+1}(S)/l = \mathrm{bi}_{\mathcal{T}\|w\|+1}(S/l)$. Dans ce cas $\mathrm{bi}_{\mathcal{T}\|l.w\|+1}(S)(l.w) = (\mathrm{bi}_{\mathcal{T}\|l.w\|+1}(S)/l)(w) = \mathrm{bi}_{\mathcal{T}\|w\|+1}(S/l)(w) = \mathrm{bi}_{\mathcal{T}}(S/l)(w)$. Le cas $l \in \mathcal{L}^-$ est symétrique. On démontre de la même manière la propriété pour $\mathrm{bs}_{\mathcal{T}}(S)$ □

Lemme A.10 *Soit* $S \subseteq \mathcal{T}$ *avec* $S \neq \emptyset$ *tel que* $\mathrm{Mi}_{\mathcal{T}}(S) \neq \emptyset$ *(resp.* $\mathrm{Mi}_{\mathcal{T}}(S) \neq \emptyset$*).* $\forall n \in \mathbb{N}$*,* $\mathrm{bi}_{\mathcal{T}n}(S) \leq_n \mathrm{bi}_{\mathcal{T}}(S)$ *et* $\mathrm{bi}_{\mathcal{T}n}(S) \geq_n \mathrm{bi}_{\mathcal{T}}(S)$ *(resp.* $\mathrm{bs}_{\mathcal{T}n}(S) \leq_n \mathrm{bs}_{\mathcal{T}}(S)$ *et* $\mathrm{bs}_{\mathcal{T}n}(S) \geq_n \mathrm{bs}_{\mathcal{T}}(S)$*).*

Démonstration : Par récurrence sur n : le cas $n = 0$ est trivial. Considérons le cas $\mathrm{bi}_{\mathcal{T}}(S) \leq_{n+1} \mathrm{bi}_{\mathcal{T}n+1}(S)$. Par le lemme A.10, $\mathrm{bi}_{\mathcal{T}}(S)(\epsilon) = \mathrm{bi}_{\mathcal{T}1}(S)(\epsilon) = \mathrm{bi}_{\mathcal{T}n+1}(S)(\epsilon)$. On a déjà $\mathrm{bi}_{\mathcal{T}}(S)(\epsilon) \leq_{\mathcal{K}} \mathrm{bi}_{\mathcal{T}n+1}(S)(\epsilon)$ et $\mathrm{bi}_{\mathcal{T}}(S)(\epsilon) \geq_{\mathcal{K}} \mathrm{bi}_{\mathcal{T}n+1}(S)(\epsilon)$. Soit $l \in \arg(\mathrm{bi}_{\mathcal{T}}(S)(\epsilon))$. Si $l \in \mathcal{L}^+$, alors, par la propriété A.9, $\mathrm{bi}_{\mathcal{T}}(S)/l = \mathrm{bi}_{\mathcal{T}}(S/l)$. On a également $\mathrm{bi}_{\mathcal{T}n+1}(S)/l = \mathrm{bi}_{\mathcal{T}n}(S/l)$. Par récurrence $\mathrm{bi}_{\mathcal{T}}(S/l) \leq_n \mathrm{bi}_{\mathcal{T}n}(S/l)$ et $\mathrm{bi}_{\mathcal{T}}(S/l) \geq_n \mathrm{bi}_{\mathcal{T}n}(S/l)$. Donc $\mathrm{bi}_{\mathcal{T}n+1}(S)/l \leq_n \mathrm{bi}_{\mathcal{T}}(S)/l$ et $\mathrm{bi}_{\mathcal{T}n+1}(S)/l \geq_n \mathrm{bi}_{\mathcal{T}}(S)/l$. Le cas $l \in \mathcal{L}^-$ est symétrique. On en déduit que $\mathrm{bi}_{\mathcal{T}}(S) \leq_{n+1} \mathrm{bi}_{\mathcal{T}n+1}(S)$ et $\mathrm{bi}_{\mathcal{T}}(S) \geq_{n+1} \mathrm{bi}_{\mathcal{T}n+1}(S)$. De même $\mathrm{bs}_{\mathcal{T}}(S) \leq_{n+1} \mathrm{bs}_{\mathcal{T}n+1}(S)$ et $\mathrm{bs}_{\mathcal{T}}(S) \geq_{n+1} \mathrm{bs}_{\mathcal{T}n+1}(S)$. □

Démonstration de la propriété 2.15 : Soit $s \in S$. Soit $n \in \mathbb{N}$. Par le corolaire 2.14, $\mathrm{bi}_{\mathcal{T}n}(S) \leq_n s$. Par le lemme A.10, $\mathrm{bi}_{\mathcal{T}}(S) \leq_n \mathrm{bi}_{\mathcal{T}n}(S)$. On a donc $\mathrm{bi}_{\mathcal{T}}(S) \leq_n s$. D'où $\mathrm{bi}_{\mathcal{T}}(S) \leq_{\mathcal{T}} s$. De même $\mathrm{bs}_{\mathcal{T}}(S) \geq_{\mathcal{T}} s$. □

Démonstration de la propriété 2.16 : Soit $t \in \mathrm{Mi}_{\mathcal{T}}(S)$. Soit $n \in \mathbb{N}$. Par le corolaire A.5, $t \leq_n \mathrm{bi}_{\mathcal{T}n}(S)$ et par le lemme A.10, $\mathrm{bi}_{\mathcal{T}n}(S) \leq_n \mathrm{bi}_{\mathcal{T}}(S)$ et donc

$t \leq_n \mathrm{bi}_{\mathcal{T}}(S)$ D'où $t \leq_{\mathcal{T}} \mathrm{bi}_{\mathcal{T}}(S)$. De même, si $t \in \mathrm{Ma}_{\mathcal{T}}(S)$, alors $t \geq_{\mathcal{T}} \mathrm{bs}_{\mathcal{T}}(S)$. $\square$

Démonstration de la propriété 2.18 : Soit S l'ensemble des sous-termes de t_1 et de t_2. Soit $S_{\mathrm{bi}} = \{\mathrm{bi}_{\mathcal{T}}(u,v) \mid u \in S \wedge v \in S\}$ et $S_{\mathrm{bs}} = \{\mathrm{bs}_{\mathcal{T}}(u,v) \mid u \in S \wedge v \in S\}$.

On montre que pour tout $t \in S_{\mathrm{bi}} \cup S_{\mathrm{bs}}$, pour tout $w \in \mathrm{dom}(t)$, $t/w \in S_{\mathrm{bi}} \cup S_{\mathrm{bs}}$, par récurrence sur w : si $w = \epsilon$ alors $t/\epsilon = t \in S_{\mathrm{bi}} \cup S_{\mathrm{bs}}$. Si $w = l.w'$: Supposons $t = \mathrm{bi}_{\mathcal{T}}(u,v)$, avec $u \in S$ et $v \in S$ (la demonstration est similaire pour $t = \mathrm{bs}_{\mathcal{T}}(u,v)$). Supposons que $l \in \mathcal{L}^+$ (la demonstration est similaire pour $\in \mathcal{L}^-$). On a, par la propriété A.9, $t/l = \mathrm{bi}_{\mathcal{T}}(\{u,v\})/l = \mathrm{bi}_{\mathcal{T}}(\{u,v\}/l)$.

- Si u/l est défini et pas v/l alors $t/l = u/l \in S \subseteq S_{\mathrm{bi}} \cup S_{\mathrm{bs}}$
- Si v/l est défini et pas u/l alors $t/l = v/l \in S \subseteq S_{\mathrm{bi}} \cup S_{\mathrm{bs}}$
- Si u/l et v/l sont définis, lors $u/l \in S$ et $v/l \in S$ et $t/l = \mathrm{bi}_{\mathcal{T}}(u/l,v/l) \in S_{\mathrm{bi}} \cup S_{\mathrm{bs}}$.

Donc $t/l \in S_{\mathrm{bi}} \cup S_{\mathrm{bs}}$. Par recurrence, $(t/l)/w' \in S_{\mathrm{bi}} \cup S_{\mathrm{bs}}$, d'où $t/w \in S_{\mathrm{bi}} \cup S_{\mathrm{bs}}$.

Comme t_1 et t_2 sont réguliers, S est fini et donc $S_{\mathrm{bi}} \cup S_{\mathrm{bs}}$ est fini. Or $\mathrm{bi}_{\mathcal{T}}(t_1,t_2) \in S_{\mathrm{bi}}$, donc tous ses sous-termes sont dans $S_{\mathrm{bi}} \cup S_{\mathrm{bs}}$, donc il y en a un nombre fini, donc $\mathrm{bi}_{\mathcal{T}}(t_1,t_2)$ est régulier. De même $\mathrm{bs}_{\mathcal{T}}(t_1,t_2)$ est régulier. $\square$

B Démonstrations de la section 3

Démonstration du lemme 3.5 : Comme les termes présents dans C sont des petits termes, $C{\downarrow}(A)/l \subseteq \mathrm{VAR}(C)$ et $C{\uparrow}(B)/l \subseteq \mathrm{VAR}(C)$. De plus, comme C est clos et $\forall\alpha \in A, \forall\beta \in B, \alpha \leq \beta \in C$, pour tout $t \in C{\downarrow}(A), t' \in C{\uparrow}(B)$, $\mathrm{subc}(t \leq t')$ est défini et inclus dans C. Soit $l \in \arg(\mathrm{sol}(A,B))$ et $l \in \mathcal{L}^+$. Soit $\alpha \in C{\downarrow}(A)/l$ et $\beta \neq \alpha \in C{\uparrow}(B)/l$. Il existe $t \in C{\downarrow}(A)$ tel que $t/l = \alpha$ et $t' \in C{\uparrow}(B)$ tel que $t'/l = \beta$. Comme $\mathrm{subc}(t \leq t') \subseteq C$ et $l \in \arg(t(\epsilon)) \cap \arg(t'(\epsilon))$, $\alpha \leq \beta \in C$. Donc $\mathrm{sol}(C{\downarrow}(B)/l, C{\uparrow}(A)/l)$ est défini. De même, si $l \in \mathcal{L}^-$, $\mathrm{sol}(C{\uparrow}(B)/l, C{\downarrow}(A)/l)$ est défini. $\square$

Lemme B.1 *Soit $k_1,k_2,k_3,k_4 \in \mathcal{K}$ tels que $k_1 \leq_{\mathcal{K}} k_2$,, $k_1 \leq_{\mathcal{K}} k_3$, $k_2 \leq_{\mathcal{K}} k_4$ et $k_3 \leq_{\mathcal{K}} k_4$. Alors $\mathrm{mi}_{\mathcal{K},\mathcal{K}}(k_2,k_1) \leq_{\mathcal{K}} \mathrm{mi}_{\mathcal{K},\mathcal{K}}(k_4,k_3)$.*

Démonstration : On pose $k = \mathrm{bs}_{\mathcal{K}}(\mathrm{Mi}_{\mathcal{K},\mathcal{L}}(k_4, \arg(k_4) \cap \arg(k_3)))$ et $k' = \mathrm{bs}_{\mathcal{K}}(\mathrm{Mi}_{\mathcal{K},\mathcal{L}}(k_2, \arg(k_2) \cap \arg(k_1)))$, $L_i = \arg(k_i)$, $L = \arg(k)$ et $L' = \arg(k')$. Par la propriété 2.6, on a $L' = \mathrm{ESMi}_{\mathcal{K},\mathcal{L}}(k_2, L_2 \cap L_1) \subseteq L_2 \cap L_1$. Comme $k_1 \leq_{\mathcal{K}} k_3 \leq_{\mathcal{K}} k_4$, $L_1 \cap L_4 \subseteq L_3$. Donc $L' \cap L_4 \subseteq L_2 \cap L_1 \cap L_4 \subseteq L_2 \cap L_3 \cap L_4 \subseteq L_3 \cap L_4$. De plus $k' \leq_{\mathcal{K}} k_2 \leq_{\mathcal{K}} k_4$, donc $k' \in \mathrm{Mi}_{\mathcal{K},\mathcal{L}}(k_4, L_4 \cap L_3)$, d'où $k' \leq_{\mathcal{K}} k$. $\square$

Démonstration du lemme 3.6 : Soit $S_B = \{t(\epsilon) \mid t \in C{\uparrow}(B)\}$, $I_A = \{t(\epsilon) \mid t \in C{\downarrow}(A)\}$, $S_F = \{t(\epsilon) \mid t \in C{\uparrow}(F)\}$, $I_E = \{t(\epsilon) \mid t \in C{\downarrow}(E)\}$. On a $I_A \subseteq I_E$ et $S_F \subseteq S_B$. Donc $\mathrm{bs}_{\mathcal{K}}(I_A) \leq_{\mathcal{K}} \mathrm{bs}_{\mathcal{K}}(I_E)$ et $\mathrm{bi}_{\mathcal{K}}(S_B) \leq_{\mathcal{K}} \mathrm{bi}_{\mathcal{K}}(S_F)$. De plus $\mathrm{bs}_{\mathcal{K}}(I_A) \leq_{\mathcal{K}} \mathrm{bi}_{\mathcal{K}}(S_B)$ et $\mathrm{bs}_{\mathcal{K}}(I_E) \leq_{\mathcal{K}} \mathrm{bi}_{\mathcal{K}}(S_F)$. Donc, par le lemme B.1, $\mathrm{sol}(A,B) = \mathrm{mi}_{\mathcal{K},\mathcal{K}}(\mathrm{bi}_{\mathcal{K}}(S_B), \mathrm{bs}_{\mathcal{K}}(I_A)) \leq_{\mathcal{K}} \mathrm{mi}_{\mathcal{K},\mathcal{K}}(\mathrm{bi}_{\mathcal{K}}(S_F), \mathrm{bs}_{\mathcal{K}}(I_E)) = \mathrm{sol}(E,F)$. $\square$

C Démonstrations de la section 4

Démonstration détaillée de la propriété 4.3 : Les propriétés 1), 2), 3) et 7) ne sont pas modifiées par l'application des règles de la figure 2. Considérons l'application de la règle (Down $\perp$) $C' \rightarrow^* D_1 = D,\alpha \leq t \rightarrow D_2 = D,\alpha \leq t'$. Par la propriété 4.1, $D,\alpha \leq t'$ est satisfiable. Donc il existe ρ tel $t_i = \rho(D_1{\downarrow}(\alpha)) \leq_{\mathcal{T}} t_\alpha = \rho(\alpha) \leq_{\mathcal{T}} \rho(t)$ et $t_i \leq_{\mathcal{T}} t_\alpha \leq_{\mathcal{T}} \rho(t')$. On a alors $t_i(\epsilon) \leq_{\mathcal{K}} t_\alpha(\epsilon) \leq_{\mathcal{K}} t'(\epsilon)$, d'où $D_2{\downarrow}(\alpha) \leq_{\mathcal{K}} D_2{\uparrow}(\alpha)$. Comme pour tout $l' \in \arg(t'(\epsilon)), t'/l' = t/l$, on conserve la propriété 6) (le traitement des autres règles est similaire). Démontrons que C'' vérifie la propriété 4). On considère la réduction suivante par (Down $\perp$) : $C' \rightarrow^* D_1 = D,\beta \leq t \rightarrow D_2 = D,\beta \leq t'$. Cette transition peut briser la condition 4). Prenons par exemple $\alpha \leq \beta \in D_1$ (les autres cas sont similaires) et montrons qu'il est possible d'effectuer une transition (Down $\perp$) pour rétablir la condition 4) vis à vis de α et β. La seule manière dont la condition 4) peut être brisée par (Down $\perp$) est que $k_\alpha = D_2{\uparrow}(\alpha)(\epsilon) \not\leq_{\mathcal{K}} t'(\epsilon)$. Soit l l'étiquette utilisée lors de l'application de (Down $\perp$). On pose $t'(\epsilon) = k'_\beta$ et $t(\epsilon) = k_\beta$. Si $l \notin \arg(k_\alpha)$, alors $k_\alpha \in \mathrm{Mi}_{\mathcal{K},\mathcal{L}}(k_\beta, \arg(k_\beta) \setminus \{l\})$ et $k_\alpha \leq_{\mathcal{K}} k'_\beta$. Sinon, on a $t/l \neq \beta$, car $t \neq \perp$. Donc la propriété 4) est vérifiée pour $D_2{\uparrow}(\alpha)/l$ et $D_1{\uparrow}(\beta)/l$, donc $D_2{\uparrow}(D_2{\uparrow}(\alpha)/l) = \perp$ et on peut appliquer la règle (Down $\perp$) sur D_2 et α, obtenant ainsi D_3. On a alors que $l \notin \arg(D_3{\uparrow}(\alpha)(\epsilon))$, d'où $D_3{\uparrow}(\alpha)(\epsilon) \leq_{\mathcal{K}} k'_\beta$. On peut donc toujours appliquer la règle (Down $\perp$) pour rétablir la propriété 4). Il en est de même pour les autre règles et pour la propriété 5). Comme par la propriété 4.2, le système de réécriture converge et termine, C'' vérifie bien les propriétés 4) et 5). $\square$