

Programmation logique, unification et espace logarithmique

Séminaire d'équipe L.D.P. de l'I2M

C. Aubert ← M. Bagnol ← P. Pistone ← Th. Seiller →



← Équipe *Logique De la Programmation*
Aix Marseille Université, CNRS, I2M UMR 7373
→ Institut des Hautes Études Scientifiques

12 juin 2014



Sources

- GIRARD « Normativity in Logic » (2012)
- AUBERT et SEILLER « Characterizing co-NL by a group action » (2012)
- AUBERT et SEILLER « Logarithmic Space and Permutations » (2013)
- AUBERT et BAGNOL « Unification and Logarithmic Space » (2014)

Introduction : À la croisée des chemins

Logique	G.d.l.	Programmation Logique
Axiome	Flot	Clause de Horn
Preuve	Câblage	Programme logique
Élimination des coupures	Équation de rétroaction	Résolution (unification)
Forte normalisation	Nilpotence	<i>Boundedness</i>

Outils

Algèbre, restrictions syntaxiques, réduction entre problèmes, etc.

➤ Résultats

Caractérisation du calcul en espace logarithmique

➤ Avancées

Nouvelles correspondances ?

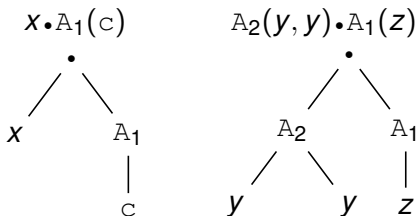
Programmation logique

Programmation logique : Termes du premier ordre

Définition (Termes du premier ordre)

t, u	$:=$	c, d, \dots	$\in C$
		x, y, z, \dots	$\in V$
		$A_n(t_1, \dots, t_n)$	$\forall n \in \mathbb{N}^*$
		$t \cdot u$	avec $t \cdot u \cdot v := t \cdot (u \cdot v)$

Exemple

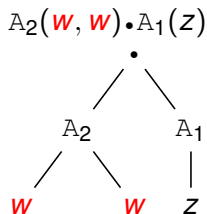
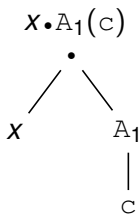


Programmation logique : Termes du premier ordre

Définition (Termes du premier ordre)

t, u := $c, d, \dots \in C$
| $x, y, z, \dots \in V$
| $A_n(t_1, \dots, t_n) \quad \forall n \in \mathbb{N}^*$
| $t.u$ avec $t.u.v := t.(u.v)$

Exemple

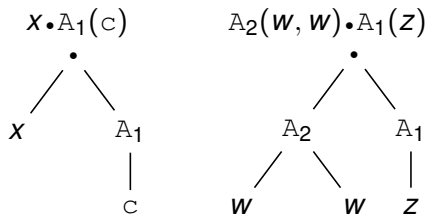


Programmation logique : Termes du premier ordre

Définition (Termes du premier ordre)

$$\begin{array}{l|l} t, u & := \quad c, d, \dots & \in C \\ & | \quad x, y, z, \dots & \in V \\ & | \quad A_n(t_1, \dots, t_n) & \forall n \in \mathbb{N}^* \\ & | \quad t \cdot u & \text{avec } t \cdot u \cdot v := t \cdot (u \cdot v) \end{array}$$

Exemple



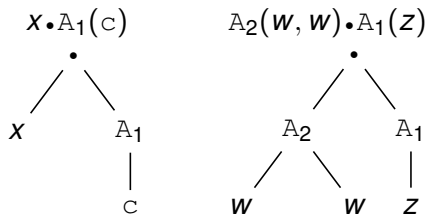
Unifiable ?

Programmation logique : Termes du premier ordre

Définition (Termes du premier ordre)

$$\begin{array}{l|l} t, u & := \quad c, d, \dots & \in C \\ & | \quad x, y, z, \dots & \in V \\ & | \quad A_n(t_1, \dots, t_n) & \forall n \in \mathbb{N}^* \\ & | \quad t \cdot u & \text{avec } t \cdot u \cdot v := t \cdot (u \cdot v) \end{array}$$

Exemple



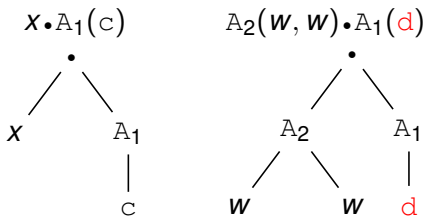
Unifiable? \checkmark : $\theta = [x/A_2(w, w) ; z/c]$

Programmation logique : Termes du premier ordre

Définition (Termes du premier ordre)

t, u	$:=$	c, d, \dots	$\in C$
		x, y, z, \dots	$\in V$
		$A_n(t_1, \dots, t_n)$	$\forall n \in \mathbb{N}^*$
		$t.u$	avec $t.u.v := t.(u.v)$

Exemple



Unifiable? $\times : c \neq d$

Définition (Flots, câblages)

Un *flot* est une paire $t \leftarrow u$ avec $\text{Var}(t) \subseteq \text{Var}(u)$.

Un *câblage* est un ensemble fini de flots.

Définition (Composition de flots)

Soient $u \leftarrow v$ et $t \leftarrow w$ deux flots, $\text{Var}(v) \cap \text{Var}(w) = \emptyset$,

$$(u \leftarrow v)(t \leftarrow w) := \begin{cases} u\theta \leftarrow w\theta & \text{si } v\theta = t\theta \\ \text{non défini} & \text{sinon} \end{cases}$$

Exemples

$$(f(x) \leftarrow x)(f(y) \leftarrow g(y)) = f(f(y)) \leftarrow g(y)$$

$$(x \cdot c \leftarrow (y \cdot y) \cdot x)((c \cdot c) \cdot x \leftarrow y \cdot x) = x \cdot c \leftarrow c \cdot x$$

$$(f(x \cdot c) \leftarrow x \cdot d)(d \cdot d \leftarrow \star) = f(d \cdot c) \leftarrow \star$$

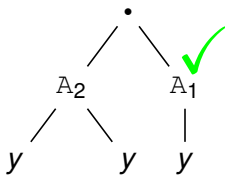
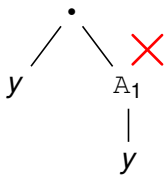
Programmation logique : Une nouvelle contrainte

Définition (Balance)

« Un terme est *balancé* si toutes les occurrences d'une même variable apparaissent à la même hauteur. »

« Un flot est *balancé* si toutes les occurrences des mêmes variables apparaissent dans les deux termes à la même hauteur. »

Exemples



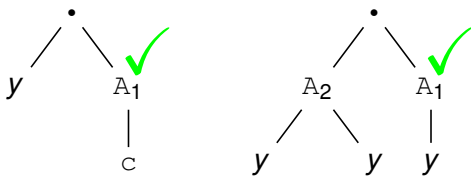
Programmation logique : Une nouvelle contrainte

Définition (Balance)

« Un terme est *balancé* si toutes les occurrences d'une même variable apparaissent à la même hauteur. »

« Un flot est *balancé* si toutes les occurrences des mêmes variables apparaissent dans les deux termes à la même hauteur. »

Exemples



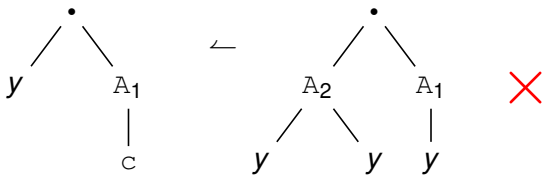
Programmation logique : Une nouvelle contrainte

Définition (Balance)

« Un terme est *balancé* si toutes les occurrences d'une même variable apparaissent à la même hauteur. »

« Un flot est *balancé* si toutes les occurrences des mêmes variables apparaissent dans les deux termes à la même hauteur. »

Exemples



Fondations algébriques

Lemme

L'ensemble des câblages \mathcal{U} est un demi-anneau avec

- $1 := x \leftarrow x$,
- $0 := \emptyset$,
- *le produit est la composition,*
- *la somme (notée $+$) est la réunion.*

(Il n'y a pas d'inverse pour l'addition.)

Lemme

L'ensemble des câblages balancés \mathcal{U}_b est un demi-anneau.

Définition (Produit tensoriel)

Soient $u \leftarrow v$ et $t \leftarrow w$ deux flots, \mathcal{A}, \mathcal{B} deux demi-anneaux,

$$(u \leftarrow v) \dot{\otimes} (t \leftarrow w) := u \cdot t \leftarrow v \cdot w$$

Le produit $\mathcal{A} \dot{\otimes} \mathcal{B}$ est également un demi-anneau :

$$\mathcal{A} \dot{\otimes} \mathcal{B} := \left\{ \sum_i F_i \dot{\otimes} G_i \mid F_i \in \mathcal{A}, G_i \in \mathcal{B} \right\}$$

On note $\mathcal{A} \dot{\otimes} \mathcal{B} \dot{\otimes} \mathcal{C} := \mathcal{A} \dot{\otimes} (\mathcal{B} \dot{\otimes} \mathcal{C})$.

Exemple

$$\begin{aligned} (f(x) \cdot y \leftarrow y \cdot x) \dot{\otimes} (x \leftarrow g(x)) &= (f(x) \cdot y \leftarrow y \cdot x) \dot{\otimes} (z \leftarrow g(z)) \\ &= (f(x) \cdot y) \cdot z \leftarrow (y \cdot x) \cdot g(z) \end{aligned}$$

Fondations algébriques : Des mots, des observations

Soit E un ensemble de termes clos, on note $E^{\leftarrow} := \{ \sum_i t_i \leftarrow u_i \mid t_i, u_i \in E \}$

Définition (Demi-anneaux de mots et d'observations)

Étant donnés deux ensembles de symboles P et S , un symbole de fonction M et $LR := \{L, R\}$, on définit :

$$\mathcal{W} := \mathcal{I} \dot{\otimes} \mathcal{I} \dot{\otimes} M(P)^{\leftarrow} \quad \text{et} \quad \mathcal{O} := S^{\leftarrow} \dot{\otimes} \mathcal{U}_b \setminus P$$

et $\mathcal{M}_{\Sigma}(\mathcal{A}) := (\Sigma \cup \{\star\})^{\leftarrow} \dot{\otimes} LR^{\leftarrow} \dot{\otimes} \mathcal{A}$.

Fondations algébriques : Des mots, des observations

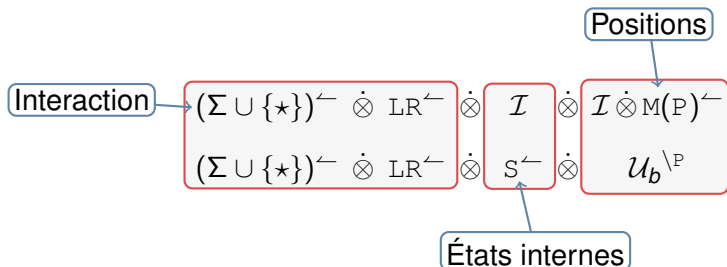
Soit E un ensemble de termes clos, on note $E^{\leftarrow} := \{ \sum_i t_i \leftarrow u_i \mid t_i, u_i \in E \}$

Définition (Demi-anneaux de mots et d'observations)

Étant donnés deux ensembles de symboles P et S , un symbole de fonction M et $LR := \{L, R\}$, on définit :

$$\mathcal{W} := \mathcal{I} \dot{\otimes} \mathcal{I} \dot{\otimes} M(P)^{\leftarrow} \quad \text{et} \quad \mathcal{O} := S^{\leftarrow} \dot{\otimes} \mathcal{U}_b \setminus^P$$

et $\mathcal{M}_{\Sigma}(\mathcal{A}) := (\Sigma \cup \{\star\})^{\leftarrow} \dot{\otimes} LR^{\leftarrow} \dot{\otimes} \mathcal{A}$.



Définition (Observation)

Une *observation* est un élément O de $\mathcal{M}_\Sigma(\mathcal{O})$.

Définition (Langage d'une observation)

Soit O une observation sur l'alphabet Σ , son langage est

$$\mathcal{L}(O) := \left\{ W \in \Sigma^* \mid \exists k \in \mathbb{N}^*, (O\bar{W})^k = 0 \right\}$$

Définition (Observation déterministe)

Soit $O = \sum_i u_i \leftarrow t_i$ une observation sur l'alphabet Σ , si tous les renommages des t_i sont deux à deux non-unifiables, alors O est déterministe.

Pointeurs et complexité

Théorème (Complétude (*Completeness*))

*Si $L \in \text{co-NL}$, alors il existe une observation O t.q. $\mathcal{L}(O) = L$.
Si $L \in L$, O peut être choisie déterministe.*

Démonstration.

Par encodage dans une observation d'un automate bi-directionnel à plusieurs têtes qui décide L . □

Pointeurs et complexité : Un exemple de dialogue

Soient $h_0, x, y \in \text{Var}$, $p_0, p_1, A_0 \in C$ et $\Sigma = \{0, 1\}$.

$$\star.L.\mathbf{init}.A_0.M(h_0) \leftarrow \star.R.\mathbf{init}.A_0.M(h_0) \quad (O_1)$$

$$\star.R.x.y.M(p_0) \leftarrow 1.L.x.y.M(p_1) \quad (W_1)$$

$$1.L.\mathbf{init}.A_0.M(h_0) \leftarrow 1.L.\mathbf{b}.A_0.M(h_0) \quad (O_2)$$

$$1.L.x.y.M(p_1) \leftarrow \star.R.x.y.M(p_0) \quad (W_2)$$

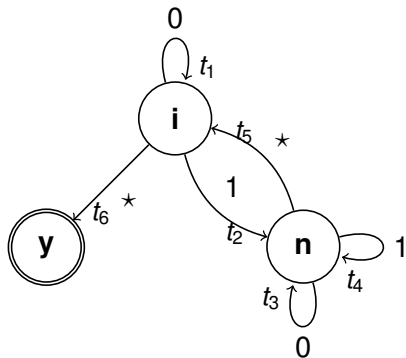
Par unification,

$$\star.L.\mathbf{init}.A_0.M(p_0) \leftarrow 1.L.\mathbf{init}.A_0.M(p_1) \quad (O_1 W_1)$$

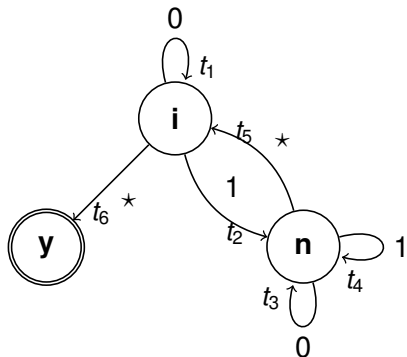
$$\star.L.\mathbf{init}.A_0.M(p_0) \leftarrow 1.L.\mathbf{b}.A_0.M(p_1) \quad (O_1 W_1 O_2)$$

$$\star.L.\mathbf{init}.A_0.M(p_0) \leftarrow \star.R.\mathbf{b}.A_0.M(p_0) \quad (O_1 W_1 O_2 W_2)$$

Pointeurs et complexité : Encoder un automate

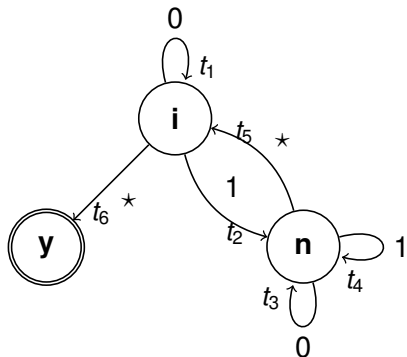


Pointeurs et complexité : Encoder un automate



$$\begin{aligned}0 \cdot \mathbf{R.i.v.M}(Z) &\leftarrow 0 \cdot \mathbf{L.i.v.M}(Z) \\1 \cdot \mathbf{R.i.v.M}(Z) &\leftarrow 1 \cdot \mathbf{L.n.v.M}(Z) \\0 \cdot \mathbf{R.n.v.M}(Z) &\leftarrow 0 \cdot \mathbf{L.n.v.M}(Z) \\1 \cdot \mathbf{R.n.v.M}(Z) &\leftarrow 1 \cdot \mathbf{L.n.v.M}(Z) * \cdot \mathbf{R.n.v.M}(Z) &\leftarrow * \cdot \mathbf{L.i.v.M}(Z) * \cdot \mathbf{R.i.v.M}(Z) &\leftarrow * \cdot \mathbf{L.y.v.M}(Z)\end{aligned}$$

Pointeurs et complexité : Encoder un automate



$$\begin{aligned}0 \cdot \mathbf{R.i.v.M}(Z) &\leftarrow 0 \cdot \mathbf{L.i.v.M}(Z) \\1 \cdot \mathbf{R.i.v.M}(Z) &\leftarrow 1 \cdot \mathbf{L.n.v.M}(Z) \\0 \cdot \mathbf{R.n.v.M}(Z) &\leftarrow 0 \cdot \mathbf{L.n.v.M}(Z) \\1 \cdot \mathbf{R.n.v.M}(Z) &\leftarrow 1 \cdot \mathbf{L.n.v.M}(Z) * \cdot \mathbf{R.n.v.M}(Z) &\leftarrow * \cdot \mathbf{L.i.v.M}(Z) * \cdot \mathbf{R.i.v.M}(Z) &\leftarrow 0\end{aligned}$$

Câblages et graphes

Théorème (Correction (*Soundness*))

Si O est une observation, alors $\mathcal{L}(O) \in \text{co-NL}$.

Si de plus O est déterministe, alors $\mathcal{L}(O) \in \text{L}$.

Techniques

- Réduction au problème d'acyclicité d'un graphe dirigé, connu pour être dans co-NL .
- Nécessite un cas particulier d'unification (« *matching* »), qui est dans L .

Démonstration.

Étant donné un mot W , on fabrique \bar{W} et on pose $O\bar{W} = F$.

Étant donné un câblage F de hauteur $h(F)$,

- L'ensemble des faits de hauteur $h(F)$ est l'espace de calcul, noté **Comp**(F),
- cet ensemble est séparant : pour tout $(u \leftarrow \star) \in \mathbf{Comp}(F)$,
 - $F(u \leftarrow \star) \in \mathbf{Comp}(F)$,
 - $F^n(u \leftarrow \star) = 0$ implique $F^n = 0$.

Étant donné **Comp**(F), on peut fabriquer un graphe qui est acyclique ssi F est nilpotent.



Démonstration.

Étant donné un mot W , on fabrique \bar{W} et on pose $O\bar{W} = F$.

Étant donné un câblage F de hauteur $h(F)$,

- L'ensemble des faits de hauteur $h(F)$ est l'espace de calcul, noté **Comp**(F),
- cet ensemble est séparant : pour tout $(u \leftarrow \star) \in \mathbf{Comp}(F)$,
 - $F(u \leftarrow \star) \in \mathbf{Comp}(F)$,
 - $F^n(u \leftarrow \star) = 0$ implique $F^n = 0$.

Étant donné **Comp**(F), on peut fabriquer un graphe qui est acyclique ssi F est nilpotent.

On peut faire tout ça en espace logarithmique. □

- Automates uni-directionnels
- Et la théorie de la démonstration ?
- Transferts de connaissance

- Automates uni-directionnels
- Et la théorie de la démonstration ?
- Transferts de connaissance

☞ Merci !