

Méthodes Formelles pour la Mobilité

RTP 23, Action spécifique 92 (2002–2003) :
bilan et perspectives

Coordinateur : Roberto AMADIO

Laboratoire d'Informatique Fondamentale de Marseille
(UMR-CNRS 6166)

<http://www.cmi.univ-mrs.fr/~amadio/as-www/index.html>

Participants

Site	Responsable
LIENS, Paris	G. CASTAGNA
LIF, Marseille	R. AMADIO
LIP, Lyon	D. HIRSCHKOFF
LIPN, Villetaneuse	P. BAILLOT
PPS, Paris	V. DANOS

Thèmes

Fondements Logiques	Applications à la mobilité
Logiques Spatiales	Langages de requête pour Données sémi-structurées
Programmation Fonctionnelle et Complexité	Contrôle de ressources et Code Mobile

Perspectives

Fondements Logiques	Applications à la mobilité	
Logiques Spatiales	Langages de requête pour Données sémi-structurées	ACI MD TRALALA (04-07)
Programmation Fonctionnelle et Complexité	Contrôle de ressources et Code Mobile	ACI SI CRISS (03-06)
ACI NIM GEOCAL (03-06)		

Dans la suite, un survol du travail effectué dans l'AS

- Souvent il s'agit d'un travail commencé dans l'AS et continué dans une des ACI.
- Certaines contributions sont omises.

Logiques spatiales

- En logique *temporelle* (arborescente) :

$$p \models \langle a \rangle \phi \text{ si } p \xrightarrow{a} p' \text{ et } p' \models \phi$$

- En logique *spatiale* :

$$p \models a[\phi] \quad \text{si } p \equiv a[p'] \text{ et } p' \models \phi$$

$$p \models \phi_1 \mid \phi_2 \quad \text{si } p \equiv p_1 \mid p_2 \text{ et } p_i \models \phi_i, i = 1, 2$$

- On peut décrire la *structure d'un système* à une équivalence près.

Historique

- Différents ‘formalismes’ pour modéliser la *migration* du calcul : $D\pi$, D-join, calcul des ambients, . . . (1995-2000)
- Un langage de *spécification* pour décrire la distribution.
- Première proposition :
L. Cardelli, A. Gordon. *Anytime, anywhere : modal logic for mobile ambients*, POPL 2000.

Réorientation

La partie *purement spatiale* est suffisamment *intéressante*!

- Les modèles de cette logique sont essentiellement des *arbres étiquetés* où l'ordre des fils ne compte pas.
- Connexion avec les *modèles de documents au format XML*.
- Connexion avec la *logique de séparation* (separation logic) de Reynolds, O'Hearn,...

Questions abordées dans l'AS

Expressivité Quelles sont les *équivalences induites* par les logiques ? Quels sont les *connecteurs essentiels* ?

Décidabilité *Vérification de modèle* et *satisfaction*.

Langages de requête Les logiques spatiales comme *langages de requêtes* pour une base de documents XML.

Références

- E. Lozes, *Expressivité des logiques d'espaces*, Thèse ENS-Lyon, 2004.
- S. Dal Zilio, D. Lugiez, C. Meyssonier, *A logic you can count on*, POPL 2004.
- V. Benzaken, G. Castagna, A. Frisch, *CDUCE : an XML centric general purpose language*, ACM-FP 2003.

Un exemple de résultat (Dal Zilio et al.)

Décidabilité de la vérification de modèle et satisfaction pour la logique spatiale suivante :

$$A ::= 0 \mid a[A] \mid (A \mid A) \mid (A \triangleright A) \mid A^* \mid \dots$$

où :

$$0 \models 0$$

$$p \models A \triangleright B \quad \text{si } \forall p' \models A \quad (p \mid p') \models B$$

$$p \models A^* \quad \text{si } p \models 0 \vee A \vee (A \mid A) \vee \dots$$

Méthode de preuve

1. Introduction d'une logique intermédiaire (*sheaf logic*) pour faciliter la représentation de multi-ensembles.
2. Une formule dans la logique est un *vecteur de formules* (la base) et un *vecteur de multiplicités* avec *contraintes arithmétiques* (de Presburger).
3. Définition d'une classe d'*automates d'arbres finis* avec contraintes qui reconnaît les modèles de la logique intermédiaire.

NB La complexité dépend fortement de la complexité des contraintes.

Contrôle de ressources pour programmes fonctionnels du premier ordre

Thèse Contrôle de ressources = terminaison+borne sur la taille des données (cf. théorème de Cobham, Bellantoni-Cook,...).

Des questions informatiques

- Couverture algorithmique.
- Inférence automatique des bornes.

Approche *Quasi-interprétation* : méthode modulaire pour déterminer la borne. Combinée avec RPO elle produit des bornes polynômiales (Marion et al.).

Contributions

- *Petits polynômes* souvent suffisants.
- *Synthèse* de quasi-interprétations *max-plus* (NP-dur).
- Mise en oeuvre d'une heuristique.

Référence R. Amadio. *Synthesis of max-plus quasi-interpretations*. Fundamenta Informaticae, 2004 (à paraître).

Ordre supérieur et logique linéaire.

Approche La *logique linéaire* est un bon cadre pour contrôler la complexité de fonctions d'*ordre-supérieur*.

Contributions

- *DLAL* (*dual light affine logic*) (variation sur *LAL*, *light affine logic*).
- *DLAL* assure une borne polynômiale en temps par rapport à une variété de stratégies de réduction (par opposition à *LAL*).
- L'inférence de type pour *DLAL* semble plus simple (comparable à celle d'*EAL*, *elementary affine logic*).

Référence P. Baillot, V. Mogbil. *Soft lambda-calculus : a language for polynomial time computation*. FOSSACS 2004.

Contrôle de ressources et code mobile

Un scénario applicatif :

- Un modèle globalement *asynchrone*, localement *synchrone* et *coopératif* (GALS).
- Le code octet de nouvelles *threads* est chargé dynamiquement et exécuté.
- Méthode d'*analyse statique* (autant que possible) pour *contrôler les ressources* et éviter des attaques du type *déni de service*.

Deux problèmes étudiés

- *Vérifier* les certificats au niveau du *code octet*.
- Aller du modèle *fonctionnel* au modèle *coopératif synchrone*.

Génération de certificats et validation de la machine virtuelle

Approche *Typed Assembly Language* = code octet avec annotations.

Contribution *Analyse de flot* qui permet de *décompiler* le code et de vérifier les certificats pour la *taille* et la *terminaison* au niveau du code octet.

Expérimentation Mise en oeuvre de la *machine virtuelle* et de l'*analyse de flot*. Validation en COQ.

Référence R. Amadio, S. Coupet-Grimal, S. Dal Zilio, L. Jakubiec. *A functional scenario for bytecode verification of resource bounds*. CSL 2004.

Contrôle de ressources pour systèmes synchrones

Approche *Extension* des méthodes développées dans le cadre fonctionnel du premier ordre.

Contributions Analyse de flot *modulaire* qui permet d'assurer :

1. la *terminaison* de chaque instant et
2. des *bornes dans un instant* (contrôle dynamique nécessaire à la fin de l'instant).

NB Modulaire = incrémentale et linéaire dans le nombre de threads.

Référence R. Amadio, S. Dal Zilio. *Resource Control for Synchronous Cooperative Threads*. CONCUR 2004.

Méta-conclusion

théorie + expérience > théorie + théorie

Fondements Logiques	Applications à la mobilité	
Logiques Spatiales	Langages de requête pour Données sémi-structurées	ACI MD TRALALA
Programmation Fonctionnelle et Complexité	Contrôle de ressources et Code Mobile	ACI SI CRISS
ACI NIM GEOCAL		