

Compte-rendu du 1er séminaire Quotient

5-6 mars à Nancy
7-9 mars au Val d'Ajol

Ce compte-rendu présente – quelques unes – des discussions et des travaux qui ont animées ou été réalisés au cours de ce séminaire. Les participants indiqués sont les participants principaux : d'autres personnes ont pu intervenir de manière ponctuelle.

1 Participants

5-6 mars : Richard Bonichon, Paul Brauner, Damien Doligez, Catherine Du-bois, Frédéric Blanqui, Yves Guiraud, Hélène Kirchner, Pierre-Etienne Moreau, Antoine Reilles, Colin Riba, Pierre Weis

7-9 mars : Richard Bonichon, Damien Doligez, Frédéric Blanqui, Yves Guiraud, Pierre-Etienne Moreau, Pierre Weis

2 Programme

2.1 1ère partie plutôt orientée théorie

2.1.1 Lundi 5 mars (LORIA, salle B013)

10h30 : arrivée du train de Paris à la gare de Nancy

11h00 : début du séminaire - description du programme de la semaine

11h15 : présentation du papier¹ qui va être présenté à ESOP'07 les 28-30 mars prochains, point de départ de l'ARC : "On the implementation of construction functions for non-free concrete data types" (Frédéric Blanqui)

12h30 : repas

13h30 : présentation de l'état actuel de Moca (Pierre Weis)

14h30 : présentation de l'ARC et des différents problèmes à résoudre (Frédéric Blanqui)

15h30 : création de groupes de réflexion sur chaque problème

17h00 : bilan des discussions ensemble

18h00 : fin de la journée

20h00 : repas en ville

¹<http://quotient.loria.fr/publis/esop07.ps.gz>

2.1.2 Mardi 6 mars

09h00 : présentation de l'état actuel de Tom et Gom² (Pierre-Etienne Moreau)

10h00 : utilisation de Tom et Gom pour écrire une procédure de recherche de preuve dans le calcul des structures (Antoine Reilles)

11h00 : présentation de l'état actuel de Zénon³ (Damien Doligez ou Richard Bonichon)

12h00 : reprise des groupes de réflexion

13h00 : repas

14h00 : introduction sur les polygraphes (Yves Guiraud)

15h15 : fin de la journée pour ceux qui doivent rentrer à Paris (départ du train à 15h50)

15h30 : séance de développement pour ceux qui restent

18h00 : fin de la journée

2.2 2ème partie plutôt orientée développement

2.2.1 Mercredi 7 mars

08h30 : rendez-vous à la gare de Nancy

08h45 : départ pour le Val d'Ajol

10h15 : arrivée au Val d'Ajol - installation des machines

11h00 : journée de développement

2.2.2 Jeudi 8 mars

journée de développement

2.2.3 Vendredi 9 mars

09h00 : matinée de développement

12h30 : rangement du matériel

13h00 : repas

14h00 : départ du Val d'Ajol

15h30 : arrivée à Nancy

15h50 : train pour Paris

3 Bilan 1ère partie

3.1 Certifier le calcul des formes normales

Participants : Frédéric Blanqui, Catherine Dubois

Un type OCaml t du premier ordre avec des constructeurs C_1, \dots, C_n :

²<http://tom.loria.fr/>

³<http://focal.inria.fr/zenon/>

```

type t =
| C_1 of t_1 * .. * t_{n_1}
| ...

```

se traduit naturellement dans Coq par un type inductif :

```

Inductive t : Set :=
| C_1 : t_1 -> .. -> t_{n_1} -> t
| ...

```

Les fonctions de constructions f_i générées par Moca semblent pouvoir être facilement formalisées dans Coq à condition de remplacer les exceptions par des tests. Presque tous les appels récursifs sont sur des arguments structurellement plus petits, sauf ceux avec $(f_I x)$. Il faudrait voir comment modifier les fonctions de constructions de façon à n'avoir que des appels récursifs avec des arguments structurellement plus petits.

On peut ensuite définir dans Coq la fonction de normalisation associée :

```

Function f (x : t) : t :=
  match t with
  | C_1 t_1 .. t_{n_1} => f_1 (f t_1) .. (f t_{n_1})
  | ...

```

Maintenant, la théorie équationnelle associée au type t (ensemble d'équations $l_1 = r_1, \dots, l_k = r_k$) peut se traduire dans Coq par un prédicat inductif :

```

Inductive teq : t -> t -> Prop :=
| teq_refl : forall x, teq x x
| teq_sym : forall x y, teq x y -> teq y x
| teq_trans : forall x y z, teq x y -> teq y z -> teq x z
| teq_C_1 : forall x_1 x_1' .. x_{n_1} x_{n_1}',
  teq x_1 x_1' -> .. -> teq x_{n_1} x_{n_1}' ->
  teq (C_1 x_1 .. x_{n_1}) (C_1 x_1' .. x_{n_1}')
| teq_1 : forall x_1 .. x_p (* variables de l_1=r_1 *), teq l_1 r_1
| ...

```

La correction et la complétude de la fonction f peuvent alors s'exprimer ainsi :

Lemma correctness : forall x, teq (f x) x.

Lemma completeness : forall x y, teq x y -> f x = f y.

Maintenant, le fait d'être en forme normale s'exprime simplement comme étant dans l'image de f :

Definition is_nf x := exists x', x = f x'.

Il conviendrait alors de pouvoir montrer automatiquement que les chaque fonction de constructions préserve bien les formes normales :

Lemma preserv_nf_i : forall x, is_nf x -> is_nf (f_i x)

3.2 Filtrage sur les formes normales

Participants : Frédéric Blanqui, Damien Doligez, Pierre Weis

Utilité d'un pattern p_{n+1} après p_1, \dots, p_n patterns : il existe une substitution close σ telle que $p_{n+1}\sigma$ soit en forme normale et ne matche aucun des p_1, \dots, p_n .

Exhaustivité d'un ensemble de patterns p_1, \dots, p_n : pour tout terme clos t , il existe un entier i et une substitution σ tels que $t = p_i\sigma$.

Si C est commutatif alors $C(t, u)$ est en forme normal si $t \leq u$: la commutativité rajoute des contraintes d'ordre.

3.3 Décider l'existence de formes normales

Participants : Yves Guiraud, Pierre-Étienne Moreau et Antoine Reilles

Yves Guiraud, Pierre-Étienne Moreau et Antoine Reilles ont travaillé sur les fonctions de construction associées à la structure de 2-polygraphe. En particulier, étude de la relation d'échange entre la composition parallèle \star_0 et la composition séquentielle \star_1 :

$$(f \star_1 g) \star_0 (h \star_1 k) \equiv (f \star_0 h) \star_1 (g \star_0 k).$$

3.4 Utiliser les propriétés des formes normales

Participants : Richard Bonichon

Richard Bonichon a travaillé sur le développement d'une procédure de recherche de preuve dans le calcul des structures utilisant Moca.

4 Bilan 2ème partie

4.1 Cat

Participants : Yves Guiraud

Yves Guiraud a travaillé sur le développement d'un outil de recherche de preuve de terminaison pour les systèmes de réécriture et les programmes polygraphiques. Réalisation de l'extraction d'un système d'inéquations polynomiales dont les solutions éventuelles prouvent la terminaison du système de réécriture de termes considéré.

4.2 Utilisation de Moca

Participants : Richard Bonichon

Utilisation de Moca pour programmer de petits démonstrateurs automatiques en logique propositionnelle.

- Premier prototype fondé sur le calcul d'inférences profondes KS, en déléguant toute la tâche à Moca, à la fois par les mots-clés et les règles utilisateur. Obtention d'une preuve que KS n'est pas confluent (et que l'on ne peut pas le compléter).
- Deuxième prototype pour KS, où l'application des règles est contrôlée en dehors de Moca, qui reste utilisé pour toute la procédure de canonisation. Travail encore en cours.
- Troisième prototype : mise en forme clausale et canonisation par Moca. Obtention d'une procédure de décision pour la logique propositionnelle au code très compact. Travail en cours pour la méthode des tableaux.

4.3 Zenon

Participants : Damien Doligez, Richard Bonichon

Étude du code et des structures de données employées dans Zenon pour déterminer comment ajouter à Zenon la faculté d'utiliser des systèmes de ré-écritures. Ouverture d'une branche déduction modulo dans le CVS de Zenon.

4.4 Moca

Participants : Frédéric Blanqui, Pierre Weis

Poursuite du développement de Moca. Correction d'erreurs découvertes par Richard Bonichon. Etude de la complétion (avec CiME) de différents systèmes de réécriture.