

6e séminaire Quotient

Val d'Ajol
19 - 23 novembre 2007

1 Résumé

L'objectif principal de ce 6e séminaire était de préparer une nouvelle distribution de **Moca**.

En dehors de cet objectif, il s'agissait également de continuer le développement de **Moca**, que ce soit en raffinant ou ajoutant certaines options ou par des tests plus poussés sur les fonctionnalités actuelles.

Les sections suivantes détaillent ces différents points.

2 Nouvelle distribution de Moca

Participants: Frédéric Blanqui, Richard Bonichon, Damien Doligez, Pierre Weis.

L'essentiel de cette activité vise à stabiliser le code de **Moca** pour le distribuer. Cela implique notamment :

- Correction des bugs découverts en particulier grâce aux sections 4 et 5 ;
- Mise à jour de la documentation pour refléter les nouveautés introduites dans **Moca** (mots-clés, fonction de comparaison...);
- Introduction des types abbrevs (et patch OCaml 3.10 associé).

La liste des nouveautés sera disponible dans les fichiers distribués et sur le site web de **Moca** (<http://moca.inria.fr>).

3 Développement de Moca

Participants: Frédéric Blanqui, Pierre Weis.

Le développement de **Moca** a continué malgré l'objectif principal qu'était la distribution. Il s'agit notamment d'ajouter des fonctionnalités plus expérimentales.

Hormis les développements mentionnés dans les deux sections 4 et 5 suivantes, l'ajout d'une procédure de complétion de type Knuth-Bendix a commencé. L'objectif à terme est bien entendu de pouvoir compléter certains systèmes

de réécriture donnés par l'utilisateur de façon automatique (ce qui est pour l'instant fait de manière ad-hoc pour certains cas).

Une réflexion sur les mots-clés de *Moca* a été engagée pour permettre de généraliser certaines équations (qui sont des variantes de distributivité) et de réduire (et de clarifier) les mots-clés du langage.

4 Génération automatique de tests

Participants: Laura Lowenthal.

Ce développement a pour objectif de permettre la génération automatique de termes *Moca* en fonction du type relationnel défini. Cela permettra également de tester d'une certaine manière que la génération de code par *Moca* est correcte, c'est-à-dire qu'elle correspond bien à la normalisation attendue.

5 Développement d'exemples

Participants: Frédéric Blanqui, Richard Bonichon, Cody Roux, Pierre Weis.

Moca comporte déjà un certain nombre d'exemples d'utilisation qui permettent en particulier de tester que la génération de code est correcte.

Le développement d'exemples plus conséquents permet de valider expérimentalement le développement de *Moca* dans un cadre plus proche du développement classique d'application.

Les exemples en développement sont les suivants :

- Démonstrateur automatique pour la logique de l'information partielle (LPI) ;
- $\lambda\sigma$ (calcul de substitutions explicites pour le λ -calcul) ;
- Variations autour d'un démonstrateur automatique pour la logique classique propositionnelle fondé sur la méthode des tableaux. *Moca* y est utilisé dans la représentation des données (avec ou sans partage maximal, variadiques) dans laquelle sont intégrées certaines étapes de la preuve : mise en DNF, fermeture de branches, ... ;
- λ -calcul quantique de Dowek-Arrighi.