

Théorie des Types et Réécriture

Frédéric Blanqui
blanqui@lri.fr
<http://www.lri.fr/~blanqui/>

Cette thèse traite de logique, sous la forme de la théorie des types ou logique d'ordre supérieur, et de calcul, sous la forme de réécriture, un paradigme de calcul Turing-complet, c'est-à-dire, par lequel on peut exprimer toutes les fonctions calculables.

Une démonstration mathématique est constituée de raisonnements ingénieux et de calculs mécanisables. Or les systèmes de développement de preuves actuels (Coq, Lego, Nuprl, HOL, etc.) ont un mécanisme de calcul assez rudimentaire. Bien qu'ils permettent d'exprimer de très nombreuses fonctions (toute fonction dont l'existence est prouvable en arithmétique intuitionniste d'ordre supérieur), les définitions peuvent être malaisées et parfois inefficaces. En effet, celles-ci doivent suivre un schéma très restrictif semblable à la récursion primitive. Ces systèmes ne permettent donc pas de définir fonctions et prédicats de la manière que l'on veut, ce qui les rend difficiles à utiliser par des non-spécialistes et limite l'aspect "calcul": des propositions qui pourraient être prouvées de manière automatique, par un simple calcul, nécessitent des déductions compliquées. Par contre, avec une notion de calcul plus riche mais telle que l'égalité de deux termes soit décidable, il n'est pas nécessaire de mémoriser les étapes de calcul. Les preuves se trouvent ainsi réduites aux "raisonnements". Or, actuellement, la taille des preuves est un problème critique dans les systèmes de développement de preuves.

Les définitions à l'aide de règles de réécriture sont à la fois plus générales, plus aisées et peuvent être plus efficaces. Mais considérer des fonctions ou des prédicats définis par un ensemble de règles de réécriture *a priori* quelconques pose un problème essentiel: peut-on vérifier si une preuve est correcte ou non? C'est la "décidabilité du typage". Ce problème est difficile car, pour vérifier qu'une preuve est correcte, on doit être capable de décider si deux termes sont égaux modulo la réécriture. Pour cela, il est suffit que la réécriture soit:

- confluente: l'ordre des réécritures n'a pas d'importance,
- fortement normalisante: un terme ne peut pas être récrit indéfiniment.

Il existe de nombreux résultats sur la confluence et la normalisation forte de la réécriture dite du premier ordre, c'est-à-dire, s'appliquant aux termes d'une algèbre engendrée par un ensemble de symboles et un ensemble de variables. Mais, dans le cadre du Calcul des Constructions de T. Coquand et G. Huet, un système de types dépendants et polymorphes qui permet d'exprimer les propositions et les preuves de la logique d'ordre supérieur et qui généralise le λ -calcul polymorphe de J.-Y. Girard, on peut vouloir définir des fonctions polymorphes d'ordre supérieur, c'est-à-dire, paramétrées par des types et des fonctions comme, par exemple, une fonction de tri paramétrée par le type des éléments à trier et une fonction de comparaison sur ce type. Cela permet d'écrire des programmes génériques, une propriété importante en génie logiciel.

Or, il existe très peu de résultats sur la confluence et la normalisation forte de ce type de réécriture. De plus, jusqu'à maintenant, aucun travail n'a considéré une telle forme de réécriture au niveau des types et des prédicats (seul le Calcul des Constructions Inductives autorise la définition de prédicats par récurrence, une forme très restreinte de réécriture). Pourtant, la réécriture au niveau des types permet d'augmenter la puissance logique du formalisme et aussi de formaliser des procédures de décision.

Dans notre thèse, nous commençons par étudier les propriétés des Systèmes de Types Purs, une classe très générale de systèmes de types dont fait partie le Calcul des Constructions, quand ceux-ci sont étendus par de la réécriture. En particulier, nous montrons qu'il est possible de linéariser certaines règles de réécriture (une règle de réécriture est linéaire si une variable n'apparaît pas deux fois dans le membre gauche) sans mettre en péril la préservation du typage ou la normalisation forte. Cela a deux conséquences pratiques importantes: puisqu'il n'y a pas de tests d'égalité pour appliquer une règle, d'une part, la réécriture est plus efficace et, d'autre part, la confluence est bien plus facile à montrer.

Ensuite, pour assurer la confluence et la normalisation forte dans le cas du Calcul des Constructions, nous donnons des conditions très générales sur les règles de réécriture dont la vérification peut être automatisée. Celles-ci généralisent tous les résultats antérieurs sur la combinaison Calcul des Constructions et réécriture, y compris le Calcul des Constructions Inductives. Nous montrons également que ces conditions peuvent s'appliquer à la Dédution Naturelle Modulo.

Bien sûr, ces conditions peuvent encore être généralisées. Par exemple, nous n'avons pas considéré de réécriture modulo certaines théories équationnelles comme l'associativité ou la commutativité, très utiles en pratique. Et des restrictions trop importantes pèsent sur les règles au niveau des types. Par ailleurs, nous n'avons pas abordé le problème de la cohérence logique qui, déjà, en l'absence de réécriture, n'admet pas de solution générale. Tous ces problèmes font partie de nos projets de recherche.