Preuves et programmes: outils classiques

Alexis SAURIN & Christine TASSON

M2 LMFI 2015-2016

La théorie de la démonstration a connu au moins deux évolutions majeures au cours du siècle dernier suite aux théorèmes d'incomplétude de Gödel. La première a eu lieu dans les années 30, immédiatement après les résultats d'incomplétude, avec l'introduction et l'étude de la déduction naturelle et du calcul des séquents par Gentzen et du lambda-calcul par Church. Church montrait alors l'indécidabilité du calcul des prédicats via le lambda-calcul tout en introduisant un modèle de calcul universel tandis que Gentzen déduisait la consistance de divers systèmes logiques comme corollaire de l'élimination des coupures en calcul des séquents.

La seconde étape a eu lieu dans les années 60 avec la mise en évidence progressive, par le biais de la correspondance de Curry-Howard, des liens profonds entre preuves et programmes, depuis la correspondance entre lambda-calcul simplement typé et déduction naturelle propositionnelle minimale jusqu'aux diverses extensions de cette correspondance au second ordre, à la logique classique et jusqu'à l'émergence de la notion de linéarité en théorie de la démonstration. La logique linéaire a profondément renouvelé les liens entre théorie de la démonstration et théorie de la programmation, conduisant à ce qu'on peut aujourd'hui appeler la ≪logique de programmation≫.

Le cours fondamental a traité de la première étape. Ce cours sera consacré aux développements depuis les années 60 et présentera les outils classiques pour l'étude de la correspondance de Curry-Howard. Après quelques rappels et compléments du cours fondamental, le cours se concentrera sur deux concepts fondamentaux, le second-ordre et la linéarité, et à leurs développements en logique de la programmation. On appliquera notamment les résultats du cours à l'étude de PCF, un langage de programmation idéalisé.

1 Programme

- Introduction et compléments (rappels sur le lambda-calcul et la théorie de la démonstration, interprétation du lambda-calcul simplement typé dans une catégorie cartésienne fermée)
- Second-ordre (Système F, Logique et arithmétique du second-ordre, théorème de normalisation forte, interprétation du système F dans les espaces cohérents et décomposition linéaire)
- Logique linéaire (calcul des séquents linéaire, réseaux de preuves et correction, élimination des coupures, sémantique catégorique de LL, traductions linéaires du lambda-calcul, focalisation et polarisation, logiques allégées et complexité, interprétations interactives)
- PCF (syntaxe et sémantique, le problème de la complète adéquation, sémantique des jeux)

2 Bibliographie

1 R. AMADIO, P.-L. CURIEN : Domains and lambda-calculi (Cambridge Tracts in Theoretical Computer Science 46, Cambridge University Press, 1998).

2 P.-L. CURIEN, H. HERBELIN, J.-L. KRIVINE, P.-A. MELLIES : Interactive Models of Computation and Program Behavior (Panorama et Synthèses, Société mathématique de France, 2009).

3 R. DAVID, K. NOUR, C. RAFFALLI : Introduction à la logique – Théorie de la démonstration (Sciences Sup, Dunod, 2004).

4 J.-Y. GIRARD, Y. LAFONT & P. TAYLOR : Proofs and Types (Cambridge Tracts in Theoretical Computer Science 7, Cambridge University Press, 1989, disponible sur la page de P. Taylor).

5 J.-Y. GIRARD : Le Point Aveugle – Cours de logique, Tomes 1 & 2 (Collection Visions des Sciences, Hermann, 2006-2007).

6 J.-L. KRIVINE : Lambda-calcul : Types et Modèles (Masson, 1990, ou E. HORWOOD : Lambda-calculus : types and models – version augmentée, en anglais, 1992).

7 B.C. PIERCE : Types and Programming Languages (MIT Press, 2002).