

# Habilitation à diriger les recherches en Informatique :

## *Logique linéaire, Types et Complexité Implicite*

soutenue par Patrick Baillot

le mardi 11 mars 2008, à 14h  
Amphi EULER, Institut Galilée

### Jury :

Pierre-Louis	CURIEN	CNRS/Université Paris 7, <i>Rapporteur</i>
Christophe	FOUQUERÉ	Université Paris 13
Jean-Yves	GIRARD	CNRS/Université Aix-Marseille 2
Martin	HOFMANN	LMU Munich, <i>Rapporteur</i>
Jean-Yves	MARION	LORIA INPL Nancy
C.-H. Luke	ONG	Oxford University, <i>Rapporteur</i>
Simona	RONCHI DELLA ROCCA	Università di Torino
Jacqueline	VAUZEILLES	Université Paris 13

**Résumé :** Les travaux présentés dans ce mémoire s'inscrivent dans le cadre de l'approche logique de la programmation, par la *correspondance de Curry-Howard*, et visent à rendre compte dans cette perspective de la complexité calculatoire, et en particulier du calcul en temps polynomial (*Ptime*). La correspondance de Curry-Howard, ou *correspondance preuves-programmes*, relie l'étude des preuves formelles à celle du  $\lambda$ -calcul, qui constitue le fondement des langages de programmation fonctionnelle comme CAML. Nous abordons le thème de la *complexité implicite*, qui cherche à définir des calculs ou des langages de programmation pour lesquels tous les programmes admettent par construction une borne de complexité donnée, telle que *Ptime*. Notre outil principal est la *logique linéaire*, qui rend finement compte des opérations sur les données, en particulier au moyen de *modalités* spécifiques pour la duplication. Des variantes de cette logique, appelées *logiques light*, comme les systèmes LLL (Girard) ou SLL (Lafont), fournissent des systèmes de complexité implicite caractérisant la classe de complexité *Ptime*.

Nos contributions ont porté sur plusieurs aspects de l'étude des logiques light.

En amont d'une part, nous avons étudié la sémantique et les possibilités calculatoires de ces systèmes. Nous avons pour cela donné un modèle dénotationnel de LLL, exploré l'extension de ce système avec des points fixes de types ainsi que son expressivité, défini et étudié un  $\lambda$ -calcul étendu correspondant à SLL, le  *$\lambda$ -calcul soft*.

D'autre part nous avons analysé l'application de ces logiques comme systèmes de types pour le  $\lambda$ -calcul. Dans un premier temps, nous avons considéré le système LAL (variante de LLL), dans lequel si un programme de  $\lambda$ -calcul est bien typé, alors il peut être exécuté en temps polynomial (*Ptime*). Nous avons montré que le fait de déterminer si un programme admet un type (*inférence de type*) dans LAL (sans polymorphisme) est un problème décidable.

Dans un deuxième temps, nous avons défini à partir de LAL un nouveau système de types, DLAL, admettant de meilleures propriétés vis-à-vis du  $\lambda$ -calcul, et certifiant aussi une complexité *Ptime*. Nous avons considéré l'inférence de type pour un autre système, EAL, garantissant une complexité élémentaire, et fourni un algorithme efficace (*Ptime*), utilisant la résolution de contraintes linéaires. À parti de cette procédure, nous avons alors élaboré un algorithme *Ptime* pour l'inférence dans DLAL.

Enfin, nous avons étudié l'évaluation des programmes de EAL et LAL par le biais de la *réduction optimale* : cette méthode s'appuie sur la réécriture de graphes et décompose l'exécution en étapes purement locales. Nous avons démontré qu'elle permettait bien d'évaluer les programmes de EAL et LAL avec les bornes de complexité attendues, respectivement élémentaire et *Ptime*.

En conclusion, nos travaux ont contribué à replacer l'étude des logiques light dans le domaine du  $\lambda$ -calcul et à montrer qu'elle permet de définir des systèmes de types pour la complexité avec une inférence efficace.

Ce mémoire s'appuie notamment sur des travaux en collaboration avec V. Atassi, P. Coppola, U. Dal Lago, V. Mogbil, K. Terui.