

École doctorale : Mathématiques et informatique de Marseille, ED184.

Unité de recherche : Institut de Mathématiques de Luminy, UMR6206.

Localisation : Campus de Luminy

Directeur de thèse : Laurent Régnier (PR Université de la Méditerranée),
regnier@iml.univ-mrs.fr

Titre de la thèse : Curry-Howard pour la concurrence.

Description. Ce sujet s'inscrit dans la thématique du projet ANR CHOCO (Curry-Howard pour la COncurrence) auquel il emprunte son titre. Le but à long terme est d'étendre la correspondance de Curry-Howard qui établit un pont entre logique mathématique et langages de programmation fonctionnels en vue d'obtenir une compréhension mathématique et logique de la programmation concurrente et des agents mobiles.

De nombreux formalismes ont été définis afin de modéliser la concurrence et la mobilité, l'exemple le plus représentatif étant le π -calcul de Robin Milner (Université de Cambridge). Ces formalismes se basent sur l'expérience acquise dans l'étude des langages fonctionnels (lambda-calcul) mais, à cause des restrictions intrinsèques aux langages séquentiels, s'en écartent notablement.

En conséquence ces formalismes, s'ils disposent déjà d'une boîte à outils théorique conséquente (systèmes de transitions étiquetés, bissimulations, structures d'événements ou de traces, catégories de préfaisceaux, systèmes de typages...) ne satisfont pas une correspondance aussi claire et naturelle que la correspondance preuve/programme entre logique et lambda-calcul. Il existe pourtant un certain nombre d'analogies entre les deux domaines, théorie de la concurrence d'une part et programmation fonctionnelle d'autre part :

- les systèmes de transitions sont les analogues des systèmes de réécritures ;
- les bissimulations sont analogues aux équivalences induites par ces systèmes de réécriture (par exemple la beta-équivalence) ;
- les traces et leurs généralisations (structures d'événements, préfaisceaux) sont analogue à la sémantique des jeux ;
- les logiques spatiales pour les processus sont analogues aux modèles de réalisabilité pour les preuves et les programmes ;
- les processus répliquables sont analogues aux boîtes promotion de la logique linéaire ;
- la localisation dans les calculs de processus est analogue à la localisation en géométrie de l'interaction et/ou en ludique ;
- ...

Le sujet de cette thèse serait, partant d'une ou plusieurs de ces analogies connues, de tenter de les étendre et les unifier dans un cadre commun, en vue d'avancer vers cette fameuse théorie globale. Une piste possible pour démarrer

serait d'utiliser les progrès récents obtenus grâce à la *logique linéaire différentielle*.

En partant de leur travail sur le lambda-calcul différentiel, Ehrhard et Régnier ont introduit une extension de la logique linéaire de Girard ; rappelons que cette dernière, apparue en 1986, s'est avérée un outil fondamental de décomposition et de compréhension des langages fonctionnels. La logique linéaire doit son nom au fait qu'elle met à jour une analogie forte entre les structures des programmes fonctionnels et les structures trouvées en algèbre linéaire. Au niveau syntaxique, elle introduit une nouvelle représentation des preuves, les réseaux de preuves qui sont un formalisme graphique.

La logique linéaire différentielle (DiLL), basé sur les travaux en sémantique de Ehrhard, ajoute à la logique linéaire des opérations s'apparentant à des opérateurs de différentiation formels. L'observation de base est que l'introduction de ces opérations différentielles revient à étendre les modalités exponentielles avec de nouvelles constructions duales de celles déjà existantes : co-affaiblissement, co-contraction et co-dereliction. Les nouvelles règles de réductions des réseaux différentiels sont non-déterministes dans le sens qu'elle transforme les réseaux en *sommes* de réseaux.

Les réseaux différentiels ont été ensuite utilisés par Ehrhard et Laurent pour modéliser un fragment significatif du π -calcul ; les réseaux différentiels apparaissent donc comme un excellent candidat pour étendre la correspondance du Curry-Howard aux calculs de processus : (1) c'est un formalisme logique, étendant la logique linéaire et disposant de toutes les propriétés attendues (un calcul des séquents, une sémantique dénotationnelle, ...) ainsi que de relations avec d'autres branches des mathématiques (ils sont fondés sur une notion de différentiation des programmes, analogue formel de la différentiation en analyse) et (2) c'est un langage de bas niveau qui permet un encodage fidèle d'un fragment significatif du π -calcul.

Connaissances et compétences requises : le candidat devra posséder les bases de la théorie des langages de programmation fonctionnelle (formalismes logique, lambda-calcul, isomorphisme de Curry-Howard, notions de sémantique dénotationnelle) et également celles de la théorie des processus (calcul de processus, bisimulation, systèmes de transitions).