1 Module "Méthodes formelles et fiabilité du logiciel"

Intervenants: Isabelle Attali, Robert de Simone, Joëlle Despeyroux

Objectif: Former des chercheurs et des ingénieurs de haut niveau à la spécification et aux preuves sur machine. Les laboratoires de recherche des industries un peu partout dans le monde sont actuellement très demandeurs de personnel qualifié dans ce secteur. Et cette demande devrait encore grandir, vu que l'informatique est de plus en plus utilisée dans des domaines critiques pour la vie humaine (comme le nucléaire, l'industrie avionique ou ferrovière), ou bien dans des domaines où les enjeux financiers sont importants (comme le spatial, ou le commerce electronique).

Les méthodes d'analyse et de preuves de propriétés de programmes se divisent en deux grandes familles complémentaires: les méthodes interactives (preuves par induction) et les méthodes automatiques (vérification exhaustive par modèles finis, réécriture). Les preuves de propriétés sémantiques et les preuves de correction de programmes utilisent des méthodes et des outils différents. On distingue aussi les preuves sur le logiciel des preuves sur le hardware; les preuves sur des langages impératifs, applicatifs, langages à objets, langages temps réels, etc...

L'objectif de ce module est de couvrir le domaine des spécifications et des preuves sur machine le plus complètement possible, par la présentation (méthodes et outils) des approches les plus modernes et les plus utilisées et par la pratique de certaines d'entre elles.

Plan du cours :

1. Introduction: Pourquoi formaliser? Les enjeux technologiques, économiques.
2. Formalismes de spécification:
   - Systèmes de transition, SOS, Sémantique Naturelle, (ex: CCS)
   - logique de Hoare, invariants (méthode B), abstraction (interpretation abstraite), TLA,
   - langages réactifs asynchrones: CCS, SDL, Lotos,
     langages réactifs synchrones: Esterel, Statecharts
3. Analyses par modèles:
   - *model-checking* et exploration exhaustive (logique temporelle), abstraction,
     applications: circuits digitaux, protocoles.
4. Preuves de correction de programmes:
   - langages fonctionnels, concurrents et à objets, logique du 1er ordre, éléments de théorie des types.
   - les assistants à la preuve (les systèmes Coq, PVS, Isabelle et HOL),
   - applications: formalisation des langages, preuves de propriétés de langages, preuves de correction de programmes,
   - application: sécurité.

Bibliographie et liens URL

- Méthodes formelles: `http://www.comlab.ox.ac.uk/archive/formal-methods.html`
- J.F. Monin, *Comprendre les méthodes formelles*, Masson, 1996.
- G. Winskel, *The Formal Semantics of Programming Languages, an introduction*, Foundations of computing, MIT Press, 1993.
- Jean-Yves Girard, Paul Taylor et Yves Lafont, *Proofs and Types*, Cambridge University Press, 1989.
- Joëlle Despeyroux, *Sémantique Naturelle: Spécifications et Preuves*, Notes de cours. Rapport de recherche INRIA RR-3359, février 98. `http://www.inria.fr/RRRT/RR-3359.html`.