



# Laboratoire PPS

## Preuves, Programmes et Systèmes

Université Paris 7 Denis Diderot, 2 Place Jussieu, 75251 PARIS



### A l'origine de PPS, la rencontre de deux mondes : preuves mathématiques et programmes informatiques

Les travaux en logique des années 70-80 ont jeté un éclairage nouveau sur le raisonnement mathématique : derrière chaque abstraction mathématique se cache un programme informatique qui réalise cette abstraction. Cette découverte lève une partie du mystère de la "déraisonnable efficacité des mathématiques" (Wigner). L'analogie entre preuves et programmes est dans certains cas si précise qu'on parle d'**isomorphisme** (dit de Curry-Howard). Cette correspondance permet le transfert de nombreuses notions et outils via un dictionnaire dont voici une partie :

Énoncé	↔	Type (= spécification)
Preuve	↔	Programme
Preuve d'un lemme	↔	Sous-routine
Preuve par récurrence	↔	Programme récursif

On peut alors certifier la correction d'un programme, voire même le fabriquer à partir d'une preuve (extraction).

Un énoncé	$A \wedge B \Rightarrow B \wedge A$ c'est-à-dire : "Si $A$ et $B$ , alors $B$ et $A$ "
Preuve informelle	Pour montrer la conclusion " $B$ et $A$ ", il nous faut montrer $B$ , puis montrer $A$ . Or de l'hypothèse " $A$ et $B$ " on peut déduire $B$ . De cette même hypothèse on déduit également $A$ .
Arbre de preuve	$\frac{\frac{A \wedge B}{B} \quad \frac{A \wedge B}{A}}{B \wedge A}$
Preuve formelle Coq	Goal $A \wedge B \rightarrow B \wedge A$ . Proof. intro hyp. destruct hyp as (a,b). exact (b,a). Qed.
Explications	L'hypothèse initiale est en fait un couple (a,b) formé d'une preuve de $A$ et d'une preuve de $B$ . En assemblant en sens inverse ces deux preuves, on obtient un nouveau couple (b,a) qui prouve la formule " $B$ et $A$ ".
Programme Ocaml	function (a,b) -> (b,a)

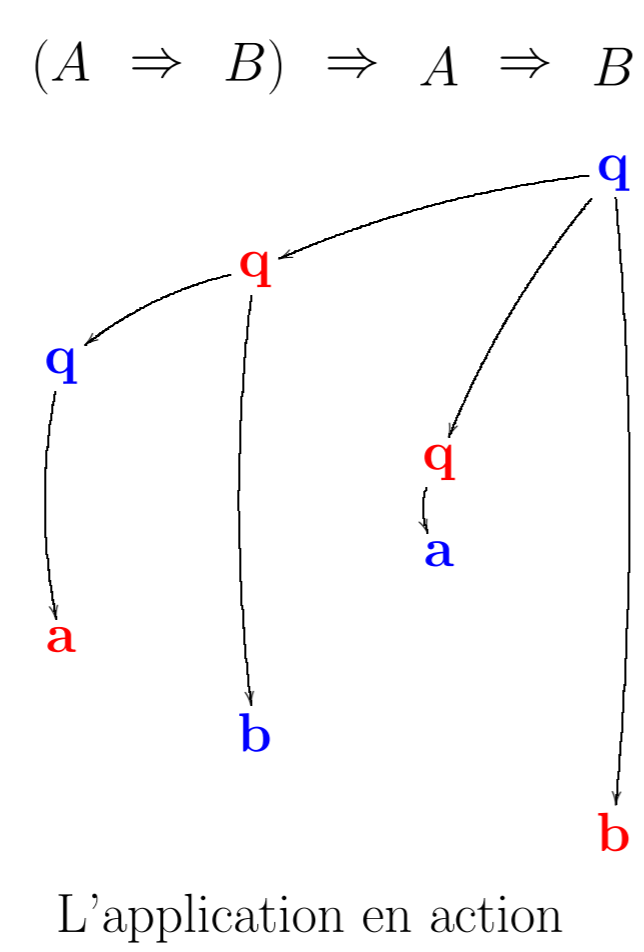
L'UMR 7126 **Preuves, Programmes et Systèmes** est un laboratoire d'interface. Il regroupe à la fois

- des mathématiciens, pour la plupart spécialistes de logique mathématique
- des informaticiens intéressés à la compréhension, à la conception et à l'implantation des langages de programmation.

De structure mono-équipe mais d'une grande variété, PPS est axé autour des thèmes suivants :

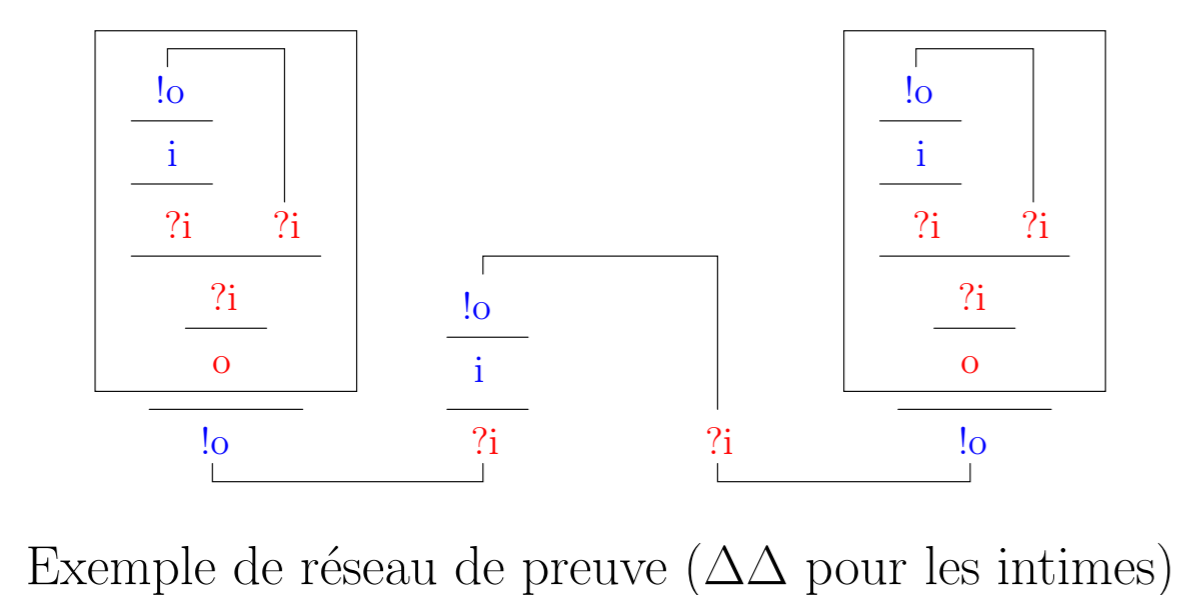
### Jeux et modèles de la programmation

Pour étudier les langages de programmation, il faut leur donner un sens : c'est le but de la **sémantique**. Elle introduit des **modèles**, c'est-à-dire des mondes mathématiques qui correspondent aux programmes de façon abstraite. Pour décrire comment les programmes interagissent avec leur environnement, de nouveaux modèles ont été créés depuis quelques années, notamment à PPS : ils prennent la forme de **stratégies** pour gagner à certains jeux.



### Théorie de la démonstration et $\lambda$ -calcul

Dans ce thème, intimement lié au thème ci-dessus, l'accent est mis sur la syntaxe. L'outil central est alors le  **$\lambda$ -calcul**, charpente syntaxique épurée de la logique intuitionniste et de la programmation fonctionnelle. On étudie ici notamment **l'élimination des coupures** (la contre-partie logique de l'évaluation des programmes) et les **réseaux de preuve** (qui sont des représentations géométriques des preuves que la logique linéaire a permis de construire).



### PPS en chiffres

- Création en **1999**
- Effectif : environ **55** (le double de l'effectif de 1999)
- **31** chercheurs et enseignants-chercheurs
- **17** doctorants
- **3** post-doc / ATER
- **1** gestionnaire
- **1** ingénieur
- **2** CDD sur projet
- Participation à **15** projets nationaux et internationaux
- Publications (2000-2003) :
  - **4** livres
  - **68** articles
- **3** récompenses récentes :
  - J.-L. Krivine : prix du rayonnement français (2004)
  - D. Kesner & S. Lengrand : meilleur article (RTA 2005)
  - A.-G. Bosser : prix jeune chercheur (Imagina 2005)

### Programmation

Le laboratoire est doté d'un véritable pôle de programmation, qui développe des produits logiciels, principalement autour du langage de programmation fonctionnelle **Ocaml**. Par exemple, ce langage est à la base d'**Ocsigen**, une nouvelle technique de développement web. On le retrouve également, couplé au prouveur **Coq**, dans des activités autour de la compilation et de l'assembleur certifié correct. Le laboratoire développe aussi une activité systèmes autour de **Jaluna/Chorus**, et une activité protocoles réseaux autour de **IPv6**. Près de la moitié des membres de PPS consacrent une part importante de leur temps à programmer, soit dans le cadre de leurs recherches, soit comme activité annexe.

### Réécriture

La réécriture repose sur l'idée de manipuler et de transformer des expressions, des valeurs, des propriétés et des méthodes plutôt que de décrire des séquences d'instructions machine. Cette approche très générale permet de modéliser quantité de systèmes informatiques ou logiques. Les systèmes de réécriture les plus riches, dit **d'ordre supérieur** (c'est-à-dire avec variables liées), permettent ainsi l'étude de fonctions définies par cas à l'aide de motifs, de procédures de normalisation liées aux logiciels d'aide à la preuve, ou encore de calculs avec substitutions explicites.

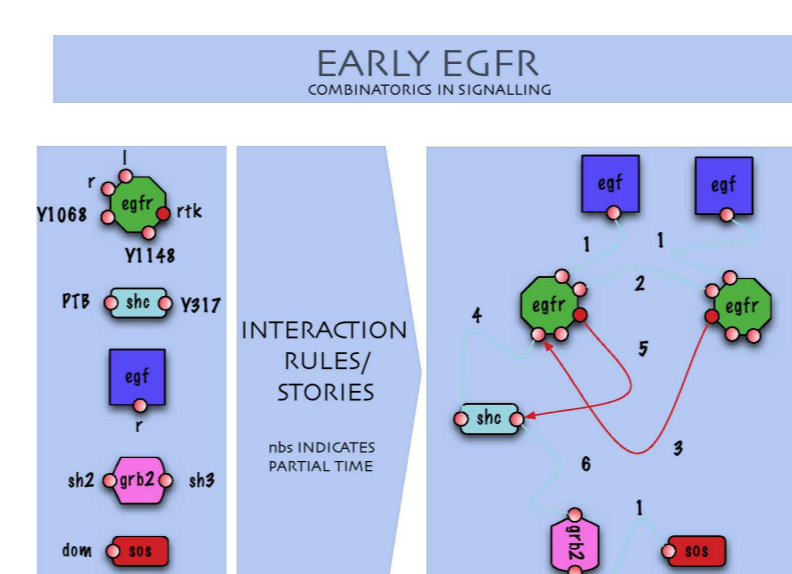
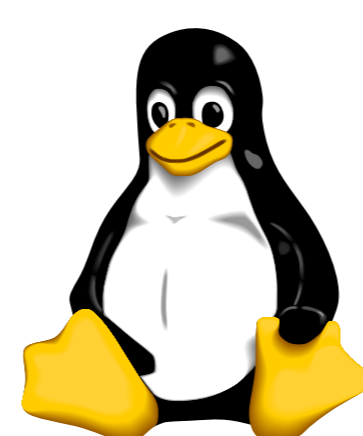


### Spécifications et réalisabilité

L'ambition ici est de débusquer quel programme se cache derrière tel ou tel théorème des mathématiques. Il fallait commencer par la logique classique, puis prendre les axiomes de la théorie des ensembles. Le principal outil pour cette étude est la **réalisabilité**, qui prend un peu le contre-pied des deux thèmes ci-dessus en ce qu'elle ne suppose pas les formules avant les preuves, mais au contraire considère les formules comme des spécifications, ou ensembles de programmes ayant le même comportement.

### Thème d'avenir : le logiciel libre

Les logiciels libres partagent avec la science de nombreuses valeurs humanistes telles que le libre accès à la connaissance. Par ailleurs, la maintenance et la fiabilisation de bases de codes vastes et en continuelle évolution pose de nombreuses questions théoriques. Sans surprise, le monde de la recherche est donc un acteur majeur du logiciel libre. En particulier, PPS est actuellement fortement impliqué dans un projet de centre commun recherche-industrie autour du logiciel libre, le **CIRILL**.



La connaissance des voies de signalisation en biologie moléculaire est fragmentée et en perpétuelle évolution. La start-up **Plectix Biosystems** travaille avec certains membres de PPS à utiliser des langages de processus spécifiques à la fois visuels, formels, et riches en détails dans une plate-forme collaborative destinée à intégrer ces données de nature dynamique.