



Direction et administration

Directeur : Frédéric Magniez

Directeur adjoint : Giuseppe Castagna

Responsable administrative : Eva Rycckelynck

Tél. : +33(0)1 57 27 92 74

Université Paris Diderot, Case 7014, Bâtiment Sophie Germain, 75205 Paris Cedex 13

Web : <https://www.irif.univ-paris-diderot.fr/>

Effectifs et équipes

L'IRIF compte actuellement une centaine de membres permanents, se répartissant en environ 54 enseignants-chercheurs, 30 chercheurs CNRS, 6 chercheurs INRIA et 5 personnels administratifs ou techniques (en janvier 2018). L'effectif total de l'IRIF, incluant doctorants, postdoctorants, et visiteurs de longue durée s'éleva à près de deux cents personnes.

L'IRIF est structuré en neuf équipes-thématiques regroupées en trois pôles de recherche :

Pôle Algorithmes et structures discrètes. Responsable: Sylvie Corteel

- Algorithmes et complexité. Responsable: Sophie Laplante

- Combinatoire. Responsable: Guillaume Chapuy

- Systèmes complexes, réseaux, calcul distribué. Responsable: Adrian Kosowski

- Théorie et algorithmique des graphes. Responsable: Pierre Charbit

Pôle Automates, structures et vérification. Responsable: Ahmed Bouajjani

- Automates et applications. Responsable: Thomas Colcombet

- Modélisation et vérification. Responsable: Constantin Enea

Pôle Preuves, programmes et systèmes. Responsable: Yves Guiraud

- Algèbre et calcul. Responsable: Christine Tasson

- Analyse et conception de systèmes. Responsable: Ralf Treinen

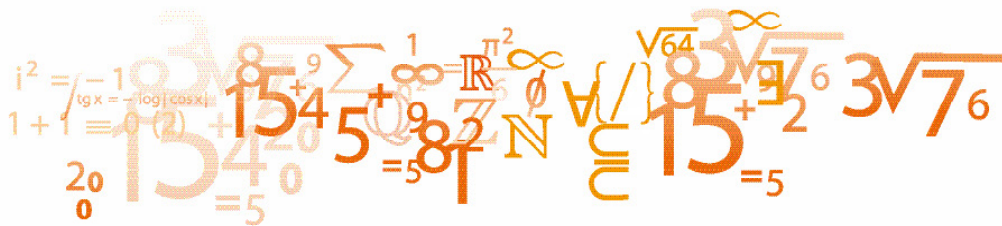
- Preuves et programmes. Responsable: Hugo Herbelin

Thèmes de recherche

Algorithmes et complexité

La théorie des algorithmes efficaces est le socle commun de nos axes de recherche, à la fois en calcul classique et quantique. Nous cherchons à relever les défis algorithmiques émergents et à comprendre les limites des nouveaux modèles de calcul. Nous identifions des problèmes centraux de l'algorithmique, puis nous élaborons et analysons des solutions efficaces tout en cherchant à montrer leur optimalité en établissant des bornes inférieures sur la complexité des problèmes traités.

En calcul classique, nous étudions plus particulièrement les modèles de calcul qui considèrent divers accès restreints à leur entrée, dont les algorithmes de streaming, en ligne, distribués ou encore de



effets faisant appel à des états internes ou à de la mémoire modifiable), et aux catégories de dimension supérieure pour modéliser la théorie des types dans un cadre homotopique. Enfin, l'étude de la réécriture est un objectif majeur de cette équipe thématique, notamment pour ses applications à l'algèbre et aux catégories, mais aussi aux différents types de lambda-calcul, à l'étude des systèmes concurrents et des langages probabilistes, dont elle permet de décrire naturellement la sémantique opérationnelle.

Analyse et conception de systèmes

Notre objectif est de développer des méthodes formelles ou mathématiques pour modéliser des systèmes existants, naturels ou artificiels, et pour résoudre des problèmes concrets. Cette thématique est très liée à l'IRILL, avec laquelle nous développons des méthodes pour rendre plus sûre et plus efficace la distribution de systèmes logiciels de grande dimension. Nous développons des langages de programmation adaptés à des domaines d'application particuliers, comme les bases de données ou le web. Nous mettons aussi au point des outils probabilistes, graphiques, à base de réécriture ou d'algèbres de processus pour modéliser des systèmes concurrents.

En application, l'équipe souhaite poursuivre et développer son activité pluridisciplinaire tournée vers la biologie des systèmes et la modélisation. Nos activités sont aussi historiquement connectées à la fondation Software Heritage, créée par des membres de PPS, et dont nous soutenons le développement.

Preuves et programmes

Notre méthodologie repose principalement sur la correspondance de Curry-Howard entre preuves et programmes, dans laquelle la réécriture joue un rôle essentiel, représentant à la fois l'élimination des coupures des preuves et la sémantique opérationnelle des programmes. L'objet de base est le lambda-calcul, une théorie très pure de la programmation fonctionnelle, et ses extensions : par des systèmes de types, par des raffinements linéaires ou différentiels, par des mécanismes de pattern-matching et par différentes sortes d'effets non purement fonctionnels (choix non déterministe ou probabiliste, continuations, états internes, locaux ou non, etc.). On obtient alors des formalismes intéressants pour leurs propriétés logiques, donnant par exemple un sens calculatoire à des concepts standard de la logique (tels que le choix ou le raisonnement par l'absurde). On obtient aussi des représentations abstraites des langages de programmation réels, dans un cadre rendu aussi indépendant que possible des implémentations, et permettant d'en analyser la sémantique. Parmi ces extensions, le calcul des constructions inductives est à la fois une logique et un langage de programmation, sur laquelle est fondé l'assistant de preuve Coq.

