

Langages de programmation: approche scientifique

Alain Frisch

IGE 2004 - ENS

29 juin 2004

Présentation au CGTI

- 1 Informatique théorique
- 2 Langages de programmation
- 3 Quelques mots sur la thèse

Une science ?

Chaque science repose un **dogme fondamental**.

Les mathématiques

Les raisonnements formels sont réductibles à de la syntaxe, avec un petit nombre d'axiomes et de règles de déductions universels.

La biologie

La mécanismes de la vie peuvent être étudiés sous l'angle des sciences de la nature.

Informatique théorique: dogme

Il existe une notion abstraite et universelle de **calcul**, qui peut être étudiée par des **méthodes mathématiques**.

Avatars

- Ordinateurs, circuits électroniques, réseaux.
- Machines à calculer mécaniques.
- Cerveaux.

Calcul abstrait

La notion de calcul n'est pas subordonnée à ses réalisations concrètes.

- Pas soumise aux contraintes matérielles.
- Calcul = objet fini, taille non bornée.

Informatique théorique et Mathématiques

Lien historique

- La notion abstraite de **calcul** s'est dégagée en parallèle à la « crise des fondements des mathématiques », et au développement de la logique formelle.
- Pis-aller à l'échec du programme réductionniste de Hilbert.

Informatique théorique \subset Mathématiques

- Cadre universel des mathématiques.
- Exemples: analyse complexe, probas, topologie, arithmétique.

Limites du calcul

La plupart des questions « intéressantes » qui portent sur le calcul ne peuvent pas être traitées par le calcul.

Le théorème de l'arrêt

Il est impossible de décider mécaniquement (par un programme) si un calcul va se terminer ou pas.

Il faut être intelligent.

Les grandes directions

- **Logique.** Étude abstraite de la notion de calcul et de ses liens avec les mathématiques.
- **Sémantique.** Étude théorique des formalismes de calcul (propriétés formelles, expressivité, interprétation mathématique).
- **Algorithmique.** Version effective (calculatoire) des propriétés mathématiques = algorithmes (décidabilité, complexité).

- 1 Informatique théorique
- 2 Langages de programmation
- 3 Quelques mots sur la thèse

Langage de programmation = formalisme pour décrire des calculs
sous forme de **programmes**.

- Programme \neq calcul !
- Problématiques théoriques.
- Aspects non mathématiques, **génie logiciel**.

Rôle de la recherche

La recherche propose/produit:

- de nouveaux **paradigmes** de programmation (impératifs, orientés objets, fonctionnels, logiques, ...) ;
- des langages qui possèdent des propriétés spécifiques (**sûreté**, **sécurité**, **efficacité**, ...) ;
- des techniques pour **corriger/renforcer/comprendre** les langages existants.

Exemple: garantir la sûreté

- Technique « industrielle »: le **test**. La recherche peut guider les tests: quels tests faire? quelle garantie (probabiliste) obtient-on?
- **Spécifications** formelles: dérivation automatique de programmes, ou preuves, avec outils spécifiques.
- Techniques d'**analyses statiques** (test symbolique, interprétation abstraite).
- Conception de langages de sorte à ce que les programmes vérifient certaines propriétés de sûreté **par construction**: typage (spécifications partielles vérifiées automatiquement), ...

Enjeux et impact

- **Fiabilité** des applications = critique (Ariane, Airbus, Météor, mais aussi: flux de données entre entreprises, avec l'administration, les usagers).
- **Cycles de développement** courts et itératifs: importance du choix du langage (facteur de réussite ou d'échec de projets).
- Nouveaux types d'applications (informatique embarquée, temps-réel, distribuée). **Internet** \neq grille-pain.
- Importance stratégique des langages: **domination technologique** via les langages (C#/.NET contre Java).

Souplesse et réactivité pour répondre à nouveaux défis.
Le **savoir-faire** de la recherche française dans le domaine est reconnu.

- 1 Informatique théorique
- 2 Langages de programmation
- 3 Quelques mots sur la thèse

Cadre

- Sous la direction de Giuseppe Castagna (CNRS).
- Département d'Informatique de l'ENS.
- Équipe Langages.

Objet

Conception et réalisation d'un langage de programmation adapté à l'écriture d'applications XML.

Objectifs

- Garantir la **sûreté** et l'**efficacité** des applications.
- Augmenter la **productivité** par un grand pouvoir expressif ; opérations dédiée aux traitements XML, qui structurent les applications.

Problématique centrale

Prendre en compte la notion de **type XML** (DTD, XML-Schema) en tant que type du langage.

Sous-domaines touchés

- Cœur sémantique:
 - Théorie des types, λ -calcul, sémantique.
 - Théorie des automates d'arbres, langages formels.
 - Interactions (dissolution, réinterprétation).
- Algorithmique (des automates, des solveurs de contraintes).
- Théorie des catégories.

Réalisations « recherche »

- **8 publications scientifique** dans des conférences et ateliers internationaux.
- **Exposés** invités ($\simeq 15$), **séjours** dans des labos étrangers (Turin, Tokyo, Cambridge).
- **Encadrement** de 3 stages.

CDuce

Mise en œuvre des résultats de la thèse: conception et réalisation du langage **CDuce** et d'un compilateur.

Logiciel

- Prise en compte des standards XML.
- Diffusion depuis juin 2003, sous forme d'un **logiciel libre**.
- Quelques (heureux) **utilisateurs**.

Projet de recherche

- Deux chercheurs permanents, trois autres thèses, plusieurs stages, et post-docs.
- Collaborations académiques et industrielles.
- Référence citée.

Great work on Cduce, by the way! We use it as a production tool in a web-design company.

Un utilisateur heureux.

Questions ?