

## Table des matières

<b>1</b>	<b>Curriculum vitae</b>	<b>2</b>
1.1	Informations personnelles . . . . .	2
1.2	Titres universitaires . . . . .	2
1.3	Parcours professionnel . . . . .	3
1.4	Distinctions . . . . .	3
<b>2</b>	<b>Activité d'enseignement</b>	<b>3</b>
2.1	Resumé . . . . .	3
2.2	Tableaux récapitulatifs . . . . .	4
2.3	Bilan et perspectives . . . . .	7
<b>3</b>	<b>Activité de recherche</b>	<b>8</b>
3.1	Domaine et contributions . . . . .	8
3.1.1	Introduction . . . . .	8
3.1.2	La séquentialité (1989-1993) . . . . .	9
3.1.3	Degrés de définissabilité (1994-2004) . . . . .	11
3.1.4	Bi-domaines et collapses extensionnels (1995-1998) . . . . .	13
3.1.5	La logique linéaire indexée (1998-2001) . . . . .	14
3.1.6	Expressivité des types intersection (1999-2000) . . . . .	15
3.1.7	Modèles du $\lambda$ -calcul pur (2002-2008) . . . . .	15
3.1.8	Extensions . . . . .	16
3.1.9	Perspectives . . . . .	18
3.2	Activité d'encadrement . . . . .	19
3.3	Publications . . . . .	19
<b>4</b>	<b>Tâches administratives et collectives</b>	<b>23</b>
4.1	Résponsabilité pédagogiques . . . . .	23
4.2	Charges collectives . . . . .	23
4.3	Animation scientifique . . . . .	23

# 1 Curriculum vitae

## 1.1 Informations personnelles

Antonio Bucciarelli  
né le 17/06/1963 à Chiaravalle (Italie), marié, trois enfants  
Adresse personnelle : 7 rue Emile Dubois, 75014 Paris  
Telephone personnel : 06 11 80 81 75  
Telephone professionnel : 01 57 27 92 17  
E-mail : buccia@pps.jussieu.fr  
Web : www.pps.jussieu.fr/~buccia  
Etablissement : Université Paris Diderot  
Grade : maître de conférence

## 1.2 Titres universitaires

### Doctorat :

Université :	Pise
Specialité :	Informatica
Directeur :	Giuseppe Longo
Titre :	Sequential models of PCF : some contribution to the domain theoretic approach to full abstraction
Date et lieu de soutenance :	15 janvier 1993, Pisa

Université :	Paris 11
Specialité :	Informatique
Directeurs :	Giuseppe Longo et Pierre-Louis Curien
Titre :	Modèles dénotationnels des langages séquentiels
Rapporteurs :	Gerard Berry et Albert Meyer
Examineurs :	Terese Hardin, Jean-Pierre Jouannaud et Laurence Puel
Date et lieu de soutenance :	26 octobre 1994, Orsay

### Habilitation à diriger des recherches :

Université :	Paris Diderot
Titre :	Sur la $\lambda$ -définissabilité
Rapporteurs :	Pierre-Louis Curien, Achim Jung et Luke Ong
Examineurs :	Jean-Louis Krivine, Yves Lafont et Giuseppe Longo
Date et lieu de soutenance :	15 décembre 2009, Paris

### 1.3 Parcours professionnel

- 1992-93 : ATER en informatique, Université Paris 11, Laboratoire de Recherche en Informatique.
- 1993-95 : post-doc du programme CE “Capital Humain et Mobilité”. Laboratoire d’Informatique de l’ENS Paris.
- 1995 (juillet-décembre) : chargé de recherche au département d’informatique de l’Imperial College de Londres.
- 1995-96 : ATER en informatique (mi-temps), Université Paris 13, Institut Galilée.
- 1996-1999, enseignant-chercheur à l’Université de Rome-La Sapienza, département d’informatique.
- Depuis 1999, maître de conférence en informatique à l’Université Paris Diderot.

### 1.4 Distinctions

- Bourse “per laureandi” du *Consiglio Nazionale delle Ricerche*, en 1986.
- *Diploma di laurea* en informatique décerné *cum laude*, à Pisa en 1987.
- Bourse individuelle dans le cadre du programme européen Capital Humain et Mobilité, août 1993-juillet 1995.
- Titre de docteur en informatique de l’Univ. Paris 11 décerné avec les félicitations du jury, 1994.
- En 2001-02 : délégation au CNRS, section 01.
- Depuis 2004 : prime d’encadrement doctorale et de recherche.

## 2 Activité d’enseignement

### 2.1 Résumé

Mes premières expériences d’enseignement ont été des vacances dans des lycées en Italie, avant de commencer ma thèse. Pendant la thèse, j’ai participé de façon sporadique à des enseignements de troisième cycle à l’Univ. de Pisa (le système italien ne prévoit pas une activité équivalente au monitorat). À partir de 1992, j’ai régulièrement enseigné l’informatique à l’université.

Dans la section suivante, je décris analytiquement mes enseignements (intitulé, charge, public, volume) année par année. Ci-dessous, un résumé des fonctions exercées et du volume d’enseignement :

- ATER à l’Université Paris 11 (1992-93)
- Vacataire à l’Université de Marne-la-Vallée (1993-95)

- ATER à l'Université Paris 13 (1995-96)
  - Enseignant-chercheur à l'Université de Roma - La Sapienza (1996-99)
  - Maître de conférence à l'Université Paris Diderot (depuis 1999)
- un volume globale d'environ 2500 heures d'enseignement (CM-TD-TP) en informatique, du premier au troisième cycle. Principaux sujets traités : introduction à l'informatique, langages de programmation, compilation, mathématiques pour l'informatique.

## 2.2 Tableaux récapitulatifs

### 1992-93, Université Paris 11, ATER

Outils mathématiques pour l'informatique	TD	L3	60h
Théorie des langages	TD	M1	50h
Théorie de la calculabilité et complexité	TD	M1	30h

### 1993-95, Université de Marne-la-Vallée, vacataire

Logique mathématique	TD	L3	30h
Programmation fonctionnelle	TD	L2	30h
Programmation fonctionnelle	TP	L3	12h

### 1995-96, Université Paris 13, ATER

Compilation	CM	École Ing. Telecom. 2ème année	28h
Compilation	TD	EIT2	28h
Compilation	TP	EIT2	28h

### 1996-99, Univ. di Roma "La Sapienza", ricercatore universitario

#### •1996-97 :

Programmation (Pascal, ML)	TD	L1	30h
Programmation (Pascal, ML)	TP	L1	30h

#### •1997-98 :

Programmation (Pascal, ML)	TD	L1	30h
Programmation (Pascal, ML)	TP	L1	30h
Architecture (68000)	TP	L1	30h

#### •1998-99 :

Programmation (Pascal, ML)	TP	L1	30h
Programmation fonctionnelle	CM	L2	30h

**Dépuis 1999, Université Paris Diderot - Paris 7, maître de conférences**

## ●1999-2000 :

Introduction à l'informatique (C)	TD	L1	26h
Introduction à l'informatique (C)	TP	L1	26h
Prog. fonctionnelle et logique	TD	L3	28h
Grammaires et analyse	TD	L3	28h
Compilation	TD	M1	56h
Sémantique des lang. de prog.	TD	M1	28h

## ●2000-01 :

Introduction à l'informatique(C)	TD	L1	26h
Introduction à l'informatique (C)	TP	L1	26h
Prog. fonctionnelle et logique	TD	L3	28h
Grammaires et analyse	TD	L3	28h
Compilation	TD	M1	56h
Sémantique des lang. de prog.	TD	M1	28h

## ●2001-02 :

Compilation	CM	M1	28h
Compilation	TD	M1	56h

## ●2002-03 :

Logique et circuits	TD	L2	26h
Compilation	CM	M1	28h
Compilation	TD	M1	28h
Domaines, catégories, sémantique	CM	M2	12h

## ●2003-04 :

Logique et circuits	TD	L2	26
Prog. fonctionnelle et logique	TD	L3	56h
Compilation	TD	M1	56h
Sémantique des lang. de prog.	TD	M1	28h
Domaines, catégories, sémantique	CM	M2	12h

## ●2004-05 :

Logique et circuits	TD	L2	26h
Logique et circuits	TP	L2	26h
Prog. fonctionnelle	TD	L3	56h
Prolog et prog. par contraintes	CM	M1	14h
Prolog et prog. par contraintes	TD	M1	14h
Sémantique des lang. de prog.	TD	M1	28h
Domaines, catégories, sémantique	CM	M2	12h

●2005-06 :

Prog. fonctionnelle	TD	L3	52h
Analyse syntaxique et compilation	TD	L3	52h
Prolog et prog. par contraintes	CM	M1	14h
Prolog et prog. par contraintes	TD	M1	26h
Sémantique des lang. de prog.	TD	M1	26h
Domaines, catégories, sémantique	CM	M2	12h

●2006-07 :

Prog. fonctionnelle	TD	L3	52h
Mathématiques discrètes	TD	L3	26h
Prolog et prog. par contraintes	CM	M1	14h
Prolog et prog. par contraintes	TD	M1	26h
Programmation 2 (Ocaml)	CM	M1 Math-Info	26h
Domaines, catégories, sémantique	CM	M2	12h

●2007-08 :

Prog. fonctionnelle	TD	L3	56h
Mathématiques discrètes	TD	L3	26h
Prolog et prog. par contraintes	CM	M1	14h
Prolog et prog. par contraintes	TD	M1	26h
Programmation 2 (Ocaml)	CM	M1 Math-Info	26h

●2008-09 :

Programmation (Java)	TD	L3 Math-Info	30h
Programmation (Java)	TP	L3 Math-Info	24h
Prog. fonctionnelle	TD	L3	26h
Mathématiques discrètes	TD	L3	26h
Prolog et prog. par contraintes	CM	M1	14h
Prolog et prog. par contraintes	TD	M1	26h
Programmation 2 (Ocaml)	CM	M1 Math-Info	26h

●2009-10 :

Outils Logiques	TD	L2	26h
Outils Logiques	TP	L2	26h
Prog. fonctionnelle	TD	L3	26h
Prolog et prog. par contraintes	CM	M1	14h
Prolog et prog. par contraintes	TD	M1	26h
Programmation 2 (Ocaml)	CM	M1 Math-Info	26h

●2010-11 :

Introduction à l'informatique	CM-TD	L1	52h
Outils Logiques	TD	L2	26h
Outils Logiques	TP	L2	26h
Prolog et prog. par contraintes	CM	M1	14h
Prolog et prog. par contraintes	TD	M1	13h
Programmation 2 (Ocaml)	CM	M1 Math-Info	26h

●2011-12 :

Introduction à l'informatique	CM-TD	L1	52h
Outils Logiques	TD	L2	26h
Outils Logiques	TP	L2	26h
Prolog et prog. par contraintes	CM	M1	14h
Prolog et prog. par contraintes	TD	M1	13h
Introduction à l'I.A.	TD	M1	26h

### 2.3 Bilan et perspectives

Au cours de bientôt vingt ans d'expérience, dans plusieurs établissements et à différents niveaux de responsabilité, j'ai essayé de construire une certaine cohérence dans mes enseignements, centrés autour des langages de programmation. Ce thème fondamentale de notre discipline comprend, de manière non exhaustive :

- des aspects mathématiques tels que la théorie des langages, la sémantique formelle, la logique, la calculabilité, la complexité,
- la compilation, les machines abstraites,
- l'étude des langages de bas niveau et des architectures,
- le génie logiciel,
- l'algorithmique,

et bien sûr l'étude des différents paradigmes de programmation, et des principaux langages qui les implémentent.

Ces sont les principaux sujets que j'ai enseignés.

Au niveau BAC+1 j'ai surtout participé à des cours d'introduction à la programmation ; à l'Univ. "La Sapienza" de Rome, dès la première année on introduisait la programmation fonctionnelle, (ML), à côté de la programmation impérative (Pascal). À Paris Diderot, j'ai repris cette année le cours d'introduction à l'informatique, auquel j'avais participé pendant deux ans lors de mon recrutement. Ce cours est suivi par tous les étudiants du département de sciences exactes ; nous y enseignons en particulier les bases de la programmation, en utilisant JAVA.

En L2 et L3, j'ai beaucoup enseigné la programmation fonctionnelle, surtout en OCaml, la logique, les mathématiques discrètes, mais aussi la programmation

en JAVA et la compilation.

Le premier cours dont je été le responsable a été d'ailleurs un cours de compilation pour des élève ingénieurs telecom de deuxième année, à Paris 13, en 1996.

Parmi les cours dont je suis ou j'ai été responsable, et que j'ai mis en place à Paris Diderot, deux cours de M1 Informatique sur la programmation logique et la compilation, respectivement, un cours de M1 Math-Info sur la programmation fonctionnelle et la partie "théorie des domaines" d'un cours de sémantique en deuxième année du Master Parisien de Recherche en Informatique.

En ce qui concerne mon activité future : enseigner la programmation et les sujets qui s'y rapportent cités ci-dessus me permettrait de bénéficier pleinement des compétences et des expériences acquises. Au même temps, je suis convaincu du fait que s'approprier un domaine dont on n'est pas spécialiste, pour monter un cours, soit bénéfique à la fois pour l'enseignant qui relève ce défi, et pour ses élèves, car le cours qui en résulte est en général très vivant et riche. Avoir expérimenté plusieurs fois cet état de choses, et la perspective de continuer à le faire, est l'une des raisons qui me font apprécier le métier d'enseignant.

## **3 Activité de recherche**

### **3.1 Domaine et contributions**

#### **3.1.1 Introduction**

Mon domaine principal de recherche est la sémantique dénotationnelle des langages de programmation, qui a comme objet l'étude des invariants des processus de calcul, ou de normalisation. Les objets syntaxiques sont, par leur nature, peu canoniques, redondants, contenant de l'arbitraire. Différentes notions de forme normale peuvent aider à les classer, et éventuellement à définir des notions convenables d'équivalence entre les programmes, mais en générale une bonne notion de *modèle* est une composante essentielle de notre compréhension du langage. Un modèle est une structure mathématique qui permet d'interpréter les programmes du langage, de telle sorte que deux programmes qui calculent la même valeur, ou plus généralement deux fragments de programme qui ont la même forme normale, aient la même interprétation.

L'enjeu de la sémantique dénotationnelle est de fournir des outils pour raisonner sur les programmes (correction, optimisation, . . .), à travers les modèles.

La richesse de ce domaine de recherche a, je crois, deux motivations principales. En premier lieu, le "feed back" réciproque entre l'étude de la syntaxe d'un langage et de ses modèles : un modèle peut suggérer des extensions, cohérentes, du langage, tandis que la modélisation d'une propriété syntaxique peut amener à

la découverte de nouveaux cadres sémantiques. Ce processus peut s’itérer, et mener à des situations où la correspondance entre langage et modèle est parfaite : le modèle fournit alors une description *complètement adéquate* du langage, et réciproquement, rendant parfois floue la frontière entre syntaxe et sémantique.

Deuxièmement, ce domaine de recherche a grandement bénéficié de l’éclairage apporté par la découverte du fait fondamental, connu comme *isomorphisme de Curry-Howard*, que les activités de *programmation* d’un côté et de *démonstration* de l’autre sont intimement liées : un programme fonctionnel codifie une preuve, son type étant la formule prouvée. Ce paradigme, découvert dans le cadre du  $\lambda$ -calcul simplement typé, côté programmation, et du calcul propositionnel minimal intuitionniste, côté logique, a connu un vif succès et a été appliqué à un très grand nombre de langages.

Des extensions désormais classiques sont celles concernant la logique classique et les langages ayant des traits de programmation impérative, ou la logique linéaire et les langages à complexité bornée.

Plus récemment, avec l’extension différentielle de la logique linéaire, ce paradigme a été étendu aux calculs de processus.

Disposer constamment de cette double perspective sur les objets qu’on étudie est une source d’inspiration d’une grande richesse. Les exemples de “cross fertilisation” sont très nombreux.

Concernant ma propre activité de recherche, je pourrais l’exposer en utilisant le vocabulaire de la logique linéaire ou celui du (ou plutôt des)  $\lambda$ -calcul(s). J’ai privilégié cette deuxième approche, plus fidèle à son déroulement.

### 3.1.2 La séquentialité (1989-1993)

C’est le thème que j’ai abordé pour ma thèse, et qui a durablement marqué mon activité de recherche. Un problème qui se posait suite aux travaux fondamentaux de D. Scott, G. Plotkin et R. Milner [Sco93, Plo77, Mil77] sur la sémantique des langages issus du  $\lambda$ -calcul de Church, était de définir un modèle d’un de ces langages, PCF<sup>1</sup>, dont tous les éléments soient définissables dans le langage PCF lui-même<sup>2</sup>. Les motivations pour cette démarche étaient multiples, notamment la caractérisation de l’équivalence opérationnelle du langage, mais de manière plus fondamentale il s’agissait de cerner la notion de fonction séquentiellement cal-

---

1. PCF est un acronyme de *Programming language for Computable Functions* ; il s’agit d’un  $\lambda$ -calcul simplement typé, avec des constantes arithmétique et logiques et de combinateurs de point fixe à tous les types ; c’est un paradigme des langages de programmation purement fonctionnels.

2. Dans la suite, je vais le citer comme “problème de la séquentialité” ou “de l’adéquation complète pour PCF”.

culable, aux types supérieurs<sup>3</sup>. Dans la décennie précédente, plusieurs tentatives avaient échoué, tout en ouvrant des voies de recherche très fécondes. Je citerai ici le modèle stable de G. Berry<sup>4</sup> [Ber78] et le modèle des algorithmes séquentiels de G. Berry et P.-L. Curien [BC82]. Sans entrer dans les détails, on peut dire que la stabilité se révéla être une contrainte trop faible pour caractériser la séquentialité, tandis que la notion d’algorithme séquentiel est trop forte. En collaboration avec T. Ehrhard, nous avons commencé par généraliser la notion d’algorithme séquentiel, en la définissant sur des structures moins contraintes que les *concrete data structures* de G. Kahn et G. Plotkin [BE93]. Cette généralisation nous a permis ensuite de réaliser que la notion de fonction séquentielle peut s’exprimer par des contraintes algébriques proches de celles définissant la stabilité. Nous avons ainsi introduit le modèle des fonctions fortement stables [BE91b], qui s’est par la suite imposé comme un bon paradigme pour la description du calcul séquentiel aux types supérieurs<sup>5</sup>.

T. Ehrhard a ensuite introduit une version “linéarisée” de ce modèle, basée sur des structures appelées *hypercoherences*, et montré qu’elle fournit un modèle de la logique linéaire de Girard [Gir87].

Toutefois, le problème de la séquentialité n’était pas résolu, notamment à cause du fait que, pour obtenir la fermeture cartésienne, on avait dû munir les fonctions fortement stables de l’ordre *stable* de Berry. Cet ordre est strictement inclus dans l’ordre point par point, ou *extensionnel*, qu’on utilise généralement pour comparer les fonctions en théorie des domaines. A titre d’exemple, une fonction de  $\mathbb{N}$  dans  $\mathbb{N}$  qui est “constante par vocation”, comme celle définie par le terme  $\lambda n. 0$ , n’est pas un majorant stable de la “constante par calcul” correspondante, définie par exemple par le terme  $\lambda n. n - n$ . Le fait que ces deux variantes de la fonction constante soient non corrélées, permet de définir, au type  $(\mathbb{N} \rightarrow \mathbb{N}) \rightarrow \mathbb{N}$ , une fonction qui les “sépare”, par exemple en renvoyant 0 sur la première et 1 sur la seconde. Une telle fonction n’est pas définissable dans un langage fonctionnelle : les informations qu’on peut obtenir en interrogeant la constante par vocation subsument celles qu’on peut obtenir de la constante par calcul, donc le résultat renvoyé sur la première doit majorer celui renvoyé sur la seconde<sup>6</sup>. Cette contrainte intuitive s’exprime techniquement par le fait que le modèle complètement adéquat de PCF est

3. Pour les fonctions de  $\mathbb{N}^k$  dans  $\mathbb{N}$ , il s’agit de la notion classique de calculabilité.

4. Par ailleurs, la notion de stabilité a été introduite indépendamment par J.-Y. Girard dans son étude des dilatateurs, et a mené à la découverte de la logique linéaire, via le modèle des espaces de cohérence.

5. Il est intéressant de remarquer que la notion de stabilité forte a été “redécouverte” plusieurs fois par la suite, par des méthodes différentes [CE94, vO99, Lon02].

6. L’ordre sur  $\mathbb{N}$  que l’on considère ici n’est pas l’ordre usuel : deux entiers différents sont incomparables, et maximaux, et il existe un minimum, noté  $\perp$ , qui dénote “l’absence d’information”.

*extensionnel pour l'ordre*. Pour résoudre ce problème, nous avons inventé une technique de plongement du modèle fortement stable dans le modèle de Scott<sup>7</sup>. Nous avons ainsi défini le modèle des *plongements extensionnels* [BE91a], et montré qu'il est "meilleur" que le modèle de Scott dans le sens que la théorie du premier inclut celle du deuxième, et qu'elle s'approche donc d'avantage de l'équivalence opérationnelle entre les termes de PCF [BE94].

Mais le modèle des plongements extensionnels contient lui aussi des éléments non définissables par des termes de PCF. Les exemples, subtils, sont des fonctionnelles séquentielles et extensionnelles pour l'ordre, qui ne respectent pas une certaine "discipline" dans l'exploration de leur argument. Les spécificités de cette discipline ne seront mises au clair qu'avec la définition des *modèles de jeux* [AJM00, HO00, Nic94]<sup>8</sup>, qui fournissent une solution du problème de la séquentialité<sup>9</sup>.

En marge du travail sur la stabilité forte, j'ai montré dans cette période qu'une contribution de Kleene au problème de la calculabilité aux types supérieurs, exposé dans une série d'articles datant du début des années 1980, est essentiellement une reformulation de la théorie des algorithmes séquentiels [Buc93a].

### 3.1.3 Degrés de définissabilité (1994-2004)

Une critique adressée à l'approche par jeux à la sémantique des langages de programmation était que les stratégies sont d'un aussi bas niveau que les programmes qu'elles dénotent, et que donc elles ne fournissent pas une représentation suffisamment abstraite de ces derniers. A titre d'exemple, considérons la suite de termes de type  $bool \rightarrow bool$  suivante :

$$T_0 = \lambda x. \text{ si } x \text{ alors } x \text{ sinon } x$$

$$T_{n+1} = \lambda x. \text{ si } (T_n x) \text{ alors } x \text{ sinon } x$$

tous ces termes sont interprétés par la fonction identité dans les modèles "standard" (de Scott, stable, fortement stable, ...). Par contre les stratégies correspondantes sont de plus en plus complexes,  $n$  croissant, car elles reflètent l'imbrication des *si... alors... sinon... L'ensemble des stratégies entre deux types finis est donc infini en général. Le modèle complètement adéquat, qui est au contraire fini au type  $bool \rightarrow bool$ , s'obtient ensuite en égalisant les stratégies qui donnent le même résultat face à tout "opposant" possible. Ce passage au quotient non effectif, appelé *collapse extensionnel*, rend le modèle difficilement exploitable.*

7. Le modèle de Scott étant extensionnel pour l'ordre.

8. La sémantique des jeux avait déjà été appliquée avec succès à des fragments de la logique linéaire.

9. Toutefois, le débat autour de ce problème demeura vivace dans la communauté scientifique, comme en verra dans la section qui suit.

En particulier, le modèle des jeux ne permet pas de décider si deux programmes sont opérationnellement équivalents, même pour des types finis. Le débat sur le problème de la séquentialité se cristallisa alors autour des questions suivantes :

- L'équivalence opérationnelle de PCF finitaire<sup>10</sup>, est-elle décidable ?
- La définissabilité dans le modèle de Scott de PCF finitaire, est-elle décidable ?

Une réponse positive à l'une ou l'autre de ces questions, strictement reliées, aurait dans la substance rouvert le problème de la séquentialité.

Inspiré par les travaux de K. Sieber sur les *relations de séquentialité* [Sie92], j'ai commencé donc à m'intéresser au problème de la définissabilité, et de la définissabilité relative, dans le modèle de Scott de PCF finitaire.

La définissabilité relative est une notion bien connue en théorie de la calculabilité : s'il existe une machine de Turing qui calcule  $f$  en utilisant un oracle pour  $g$ , on dit que  $f$  est définissable relativement à  $g$ , ou  $g$ -définissable. Par analogie, pour deux fonctions  $f$  et  $g$  du modèle de Scott de PCF,  $f$  est dite  $g$ -définissable s'il existe un terme de PCF étendu avec une constante désignant  $g$  qui définit  $f$ <sup>11</sup>.

Avant d'illustrer mes contributions sur ce thème, il convient de citer ici un résultat important obtenu par R. Loader en 1996 [Loa01], qui répond par la négative aux questions de décidabilité posées ci-dessus : l'équivalence opérationnelle de PCF finitaire est indécidable (et donc la PCF-définissabilité dans le modèle de Scott l'est aussi). Ce résultat négatif marqua la fin de la quête d'un modèle standard, complètement adéquat de PCF.

Le problème de la définissabilité relative des fonctions continues de Scott, bien qu'indécidable en général, demeurait intéressant et très largement inexploré.

Un *degré de définissabilité* (ou de *parallélisme*) étant une classe d'équivalence de la relation d'inter-définissabilité, on voit que les degrés de définissabilité forment un sup semi-treilli, dont le plus grand élément est la classe de la fonction *ou parallèle*<sup>12</sup>, et le plus petit est l'ensemble des fonctions PCF-définissables.

A partir de là, ma contribution se base sur une découverte assez surprenante : on peut associer à chaque fonction  $f$  du *premier ordre*<sup>13</sup> du modèle de Scott de PCF finitaire un hypergraphe  $H_f$ , de telle sorte que les morphismes d'hypergraphes entre  $H_f$  et  $H_g$  fournissent des informations sur les degrés de parallélisme de  $f$  et de  $g$ . L'hypergraphe en question a pour sommets les éléments de la *trace* de  $f$ , et pour arêtes des ensembles de sommets qui représentent en quelques sortes des

---

10. PCF finitaire est la variante de PCF qui n'a que *bool* comme type de base, et les constantes *vrai*, *faux*, *si... alors... sinon...* et  $\Omega$ , désignant un terme qui boucle indéfiniment.

11. Il est clair qu'on peut parler plus généralement de définissabilité relative dans un modèle  $\mathcal{M}$  d'un langage  $L$ .

12. Le fait que le degré du ou parallèle soit maximum est non triviale, et il est dû à G. Plotkin.

13. Un type du premier ordre étant de la forme  $bool \rightarrow (bool \rightarrow \dots (bool \rightarrow bool) \dots)$ ; c'est le type des *circuits booléens*.

“informations cohérentes maximales” sur l’argument de  $f$ .

La pertinence de  $H_f$  pour l’étude du degré de parallélisme de  $f$  a été mise à jour en plusieurs étapes :

- Dans [Buc97b], j’ai exhibé une hiérarchie bidimensionnelle  $h_{n,m}$  d’éléments du modèle de Scott à l’intérieur de laquelle on trouve à la fois de chaînes croissantes et décroissantes infinies, et des anti-chaînes infinies de degrés de parallélisme. Pour les éléments de cette hiérarchie, un morphisme d’hypergraphe entre  $H_f$  et  $H_g$  existe si et seulement si  $f$  est  $g$ -définissable.
- En collaboration avec P. Malacaria, nous avons ensuite généralisé la partie “seulement si” de ce résultat en montrant que, pour tout  $f, g$ , l’existence d’un morphisme d’hypergraphe entre  $H_f$  et  $H_g$  implique que  $g$  est  $f$ -définissable [BM02].
- En général, il se peut que  $f$  soit  $g$ -définissable sans qu’il existe de morphisme entre  $H_f$  et  $H_g$ . En collaboration avec B. Leperchey, nous avons proposé une notion faible de morphisme d’hypergraphe, appelé *morphisme temporisé*, et montré qu’elle est correcte et complète pour la caractérisation de la définissabilité relative dans un sous-ensemble significatif du modèle de Scott, composé des fonctions *sous-séquentielle*<sup>14</sup> [BL04]. Nous avons aussi montré que le problème de la définissabilité relative entre fonctions sous-séquentielles du premier ordre est NP-complet.

En collaboration avec B. Leperchey et V. Padovani, nous avons étudié la définissabilité relative dans le modèle de Scott d’un langage encore moins expressif que PCF finitaire (*PCF unaire*). Nous avons montré que la hiérarchie des degrés de définissabilité est non triviale, même dans ce cadre extrêmement restreint [BLP03].

### 3.1.4 Bi-domaines et collapsés extensionnels (1995-1998)

J’ai évoqué plus haut un problème des modèles ordonnés par l’ordre stable vis à vis de l’adéquation complète : ils font apparaître aux types supérieures des éléments non définissables, comme la fonctionnelle qui sépare les fonctions constantes “par vocation” et “par calcul”. Pour résoudre ce problème, plusieurs définitions de *bi-domaines* ont été proposées, à partir de celle présentée dans la thèse de G. Berry [Ber79]<sup>15</sup>. Dans les bi-domaines, les relations d’ordre stable et extensionnel coexistent. A titre d’exemple, la fonctionnelle décrite ci-dessus n’est pas un morphisme de bi-domaines car elle n’est pas monotone pour l’ordre extensionnel.

Il s’agit donc de “combiner” deux modèles différents pour en définir un troisième, qui en soit la synthèse.

---

14. Une fonction est sous-séquentielle si elle est majoré point par point par une fonction séquentielle.

15. Le modèle des plongements extensionnelles, déjà à cité, s’inscrit dans cette démarche.

Une façon alternative d’obtenir cette synthèse, que j’ai proposée et étudiée dans [Buc97c, Buc97a, Buc96], est la suivante : on considère deux modèles  $\mathcal{M}$  et  $\mathcal{N}$ , dont on suppose qu’ils interprètent de la même manière les types de base du langage (entiers, booléens,...) <sup>16</sup>. ; on définit une famille de relations binaires, indexée par les types, qui mettent en relation les éléments de  $\mathcal{M}$  et ceux de  $\mathcal{N}$ , comme suit :

- Aux types de base, la relation est l’identité.
- Au type  $\sigma \rightarrow \tau$ , la relation est définie selon le paradigme de la “réductibilité”, ou des “relations logiques” : deux morphismes sont reliés si et seulement s’ils envoient des arguments reliés sur des valeurs reliées.

Cette relations de *collapse extensionnel hétérogène* permet de définir un modèle quotient  $\mathcal{M}/\mathcal{N}$ , dont la théorie inclut celles de  $\mathcal{M}$  et de  $\mathcal{N}$ , réalisant ainsi de façon systématique l’objectif affiché, et pas toujours atteint, par les constructions de bi-domaines.

Ce thème, qui se situe pour ses motivations initiales dans la continuité du problème de la séquentialité, est en réalité un axe transversal, qui continue d’inspirer mon travail de recherche [Buc11, Buc].

En effet, le champ d’application de la construction de collapse extensionnel hétérogène est vaste, et les résultats qui en découlent peuvent être surprenants : à titre d’exemple, en collaboration avec I. Salvo de l’Univ. “La Sapienza” de Rome, nous avons montré que les circuits booléens s’obtiennent par collapse extensionnel du modèle de Scott via la relation de *totalité* [BS98] : la construction est celle décrite ci-dessus, mais la relation au type de base est l’inclusion de  $\{0, 1\}$  vers  $\{\perp, 0, 1\}$ , au lieu de l’identité. Ainsi, on a pu associer à chaque circuit booléen  $c$  un treillis de fonctions booléennes du modèle de Scott, la *classe de totalité* de  $c$ , et montrer que la taille de cette classe de totalité est liée à (une mesure de) la complexité de  $c$ .

### 3.1.5 La logique linéaire indexée (1998-2001)

Nous avons vu que les relations logique ont été un outil performant dans l’étude du problème de la séquentialité <sup>17</sup>. En février 1998, lors d’un séjour à Marseille suite à une invitation de T. Ehrhard, nous avons initié un programme qui consistait en appliquer la méthode des relations logiques à l’étude de la définissabilité dans le modèle canonique de la logique linéaire : les espaces de cohérence.

En étudiant le fragment sans exponentiels, MALL, nous avons commencé par remarquer que, pour définir une notion adéquate de relation logique sur le modèle des espaces de cohérence, il fallait enrichir ce modèle d’un monoïde commutatif,

16. Cette hypothèse nous permet de simplifier la présentation, mais elle n’est pas nécessaire.

17. Par ailleurs, elles ont inspiré une solution du problème de la séquentialité, [OR95], alternative a celles basées sur les jeux.

suivant une construction due à Girard. En cherchant à caractériser la définissabilité dans le modèle étendu, nous avons défini une version indexée de la logique linéaire, MALL(I), qui a la propriété remarquable suivante : un ensemble de points du modèle étendu est contenu dans l'interprétation d'une preuve de MALL<sup>18</sup> si et seulement si la formule de MALL(I) qui lui est associé est prouvable. Cette correspondance entre la sémantique des phases de la logique indexée et la sémantique dénotationnelle de MALL va plus loin : on peut associer à tout modèle de phases de MALL(I) un modèle dénotationnel de MALL. Ces résultats sont présentés dans [BE00], et étendus à la logique linéaire avec exponentiels dans [BE01].

### 3.1.6 Expressivité des types intersection (1999-2000)

Les systèmes de typage basés sur les *types intersection* sont plus expressifs que ceux basés sur les *types simples*, dans le sens qu'ils permettent de typer plus de termes purs (non typés). Cette simple constatation ne permet pas vraiment de comparer le pouvoir expressif des deux systèmes, car pour cela il faudrait exhiber un terme typable dans le système plus puissant, dont le comportement opérationnel ne puisse pas être simulé par un terme typable dans le système plus faible, ou bien montrer qu'un tel terme n'existe pas. Typiquement, il faudrait exhiber une fonction de  $\mathbb{N}^K$  dans  $\mathbb{N}$  qui soit définissable dans l'un et pas dans l'autre.

D. Leivant avait conjecturé qu'il n'y a pas de telle fonction, c'est à dire que les deux systèmes définissent le même ensemble de fonctions numériques<sup>19</sup>.

Avec deux collègues de Roma "La Sapienza", A. Piperno et I. Salvo, nous avons démontré cette conjecture [BPS03, BLPS99].

Techniquement, la preuve se base sur une traduction des *dérivations de typage* dans les types intersection vers des *termes* simplement typables, qui a de bonnes propriétés vis à vis de la  $\beta$ -réduction. Un corollaire intéressant en est une nouvelle preuve de normalisation forte pour le système des types intersection, qui repose uniquement sur la normalisation forte du système des types simples.

### 3.1.7 Modèles du $\lambda$ -calcul pur (2002-2008)

Les modèles stables du  $\lambda$ -calcul pur, ou non typé, ont été mon tout premier sujet de recherche [Buc87].

À partir de 2002, en collaboration avec A. Salibra de l'Université Cà Foscari de Venise, nous avons revisité une classe bien connue de modèles du  $\lambda$ -calcul pur, les *modèles de graphe*, ou modèles de Plotkin-Scott-Engeler définis dans les années 1970.

---

18. On dit dans ce cas que l'ensemble en question est sous-définissable

19. Dans les deux cas il s'agit d'un sous-ensemble des *fonctions élémentaires*.

En particulier nous avons montré que cette classe de modèles admet une plus petite théorie, et que cette théorie est strictement incluse dans la  $\beta$ -équivalence [BS03].

Ensuite, nous nous sommes intéressés à la plus grande  $\lambda$ -théorie issue des modèles de graphe, parmi les théories *sensibles*<sup>20</sup>, et montré qu'elle coïncide avec la théorie  $\mathcal{B}$  des *arbres de Böhm* [BS04].

Nos contributions sur ce sujet sont résumées dans [BS08].

Plus récemment, en collaboration avec T. Ehrhard et G. Manzonetto, nous avons défini un modèle  $\mathcal{D}$  du calcul non typé dans la catégorie des ensembles et des relations [BEM07]. Il s'agit d'un modèle très général, qui accommode un certain nombre d'extensions du  $\lambda$ -calcul que je vais décrire dans la section suivante.

En particulier, ce modèle *n'a pas assez de points*, au sens catégorique, ce qui le rend bien adapté aux extensions différentielles ou "avec ressources" du calcul.

Par ailleurs, l'ensemble  $\mathcal{P}(\mathcal{D})$  est naturellement muni de deux opérations qui permettent d'interpréter la somme non déterministe et la composition parallèle, respectivement.

### 3.1.8 Extensions

Ma recherche récente porte principalement sur des extensions du calcul fonctionnel et déterministe, selon plusieurs axes.

**Calcul non-déterministe et parallèle :** un premier axe, liée au modèle  $\mathcal{D}$  mentionné ci-dessus, concerne une extension du  $\lambda$ -calcul avec non-déterminisme et parallélisme [BEM09a, BEM09b]. Ces extensions se modélisent aisément dans le modèle  $\mathcal{D}$  : le non-déterminisme, *may*, est interprété par l'union ensembliste, tandis que le parallélisme, *must*, est interprété par une opération de "mélange", stricte dans ses deux arguments, qui est liée à l'interprétation de la règle MIX de la logique linéaire. Nous avons un modèle simple qui n'est toutefois pas complètement adéquat, car par exemple les interprétations des termes  $Id|Id$ , obtenu par composition parallèle de deux copies de la fonction identité, et le terme  $Id$  tout court, sont différentes, tandis que ces termes ne peuvent pas être séparés par un contexte du langage. La version extensionnelle de  $\mathcal{D}$ , définie par T. Ehrhard dans [Ehr09], est un candidat à fournir un modèle complètement adéquat. Dans l'autre sens, nous verrons plus bas un langage suffisamment expressif pour définir tous les éléments de  $\mathcal{D}$ , pour lequel  $\mathcal{D}$  est complètement adéquat.

---

20. A savoir, les théories qui égalisent tous les termes *non résolubles*, représentant l'indéfini.

**Calcul avec ressources :** le  $\lambda$ -calcul différentiel et les réseaux différentiels de T. Ehrhard et L. Regnier [ER03, ER06] ont été très étudiés ces dernières années, car ils fournissent un cadre adéquat pour analyser des aspects quantitatifs du calcul, comme la réplification des ressources nécessaire à son déroulement. Récemment, R. Blute, R. Cockett et R. Seely ont défini le cadre catégorique des modèles du calcul différentiel. En collaboration avec T. Ehrhard et G. Manzonetto [BEM10], nous avons raffiné leur notion de *catégorie cartésienne différentielle*, défini celle de  *$\lambda$ -catégorie différentielle* et montré que cette dernière fournit un modèle du  $\lambda$ -calcul différentiel simplement typé, et aussi d’autres instances de calculs avec ressources, comme ceux basés sur le  *$\lambda$ -calcul avec multiplicités* de G. Boudol. Nous montrons que les espaces de finitude de Ehrhard, et le modèle relationnel standard de la logique linéaire sont deux instances de  $\lambda$ -catégorie différentielle.

**Calcul avec tests :** en collaboration avec A. Carraro, T. Ehrhard et G. Manzonetto nous avons récemment défini une extension du  $\lambda$ -calcul avec ressources pour laquelle le modèle  $\mathcal{D}$  est complètement adéquat [BCEM11, ?]. La spécificité de ce langage est de posséder un mécanisme rudimentaire d’exceptions, dont nous montrons qu’il permet de définir tous les éléments finis de  $\mathcal{D}$ . C’est un exemple de langage directement issu d’un modèle spécifique, dont il reste largement à explorer les aspects syntaxiques et calculatoires.

**Calcul avec motifs :** le *filtrage de motifs* (pattern matching) est un mécanisme à la fois simple et puissant de passage des paramètres, utilisé en programmation fonctionnelle. Son étude formelle a débuté récemment avec la définition de *calculs avec motifs* (pattern calculi), où le filtrage de motifs s’intègre au cadre bien établi de la réécriture et du  $\lambda$ -calcul [JK06]. En collaboration avec D. Kesner et S. Ronchi nous étudions le cas très simple, où les motifs sont uniquement des variables ou des couples de motifs [BKdR]. L’étude des propriétés de normalisation des termes, en particulier de la *solvability*, y est déjà complexe, à cause des blocages possibles au cours de la réduction<sup>21</sup>. Nous nous intéressons à un système de typage tels que les termes typables soient exactement les résolubles (*solvable*).

**Linear information systems :** la “linéarisation” des modèles du  $\lambda$ -calcul a produit plusieurs modèles de la logique linéaire : un exemple déjà cité est celui des hypercoherences. En collaboration avec A. Carraro et T. Ehrhard, nous avons appliqué cette méthode aux *systèmes d’information* de D. Scott. Ceci nous a mené

---

21. Par exemple le terme  $(\lambda\langle x, y \rangle.\Delta)z\Delta$ , où  $\Delta$  est le duplicateur  $\lambda u.uu$ , n’est pas résoluble, car si  $z$  est substitué par un terme qui se réduit à un couple, le terme tout entier réduit à  $\Omega$ , sinon il est bloqué.

à définir la catégorie des *systèmes d'informations linéaires* [BCES10], équivalents aux domaines de Scott premiers algébriques, qui généralisent un modèle connu de la logique linéaire classique, celui des treillis premiers algébriques et des fonctions continues.

**Situations de collapse extensionnel :** nous avons vu que un *collapse extensionnel* est la relation logique binaire sur un modèle (non extensionnel) définie comme l'identité aux types de base. Mon objectif avec la définition de *situation de collapse extensionnel* [Buc11, Buc] est de généraliser cette notion selon deux directions.

Premièrement, la relation n'est pas forcément l'identité aux types de base, mais plus généralement une identité partielle, qui élimine des éléments correspondant à des comportements qu'on veut interdire par collapse. Par exemple, la relation de totalité sur le domaine plat des valeurs de vérité, indéfinie sur  $\perp$ , dont j'ai déjà à parlé.

Deuxièmement, je considère ces relations en tant que *pre-logical relations*, [HS02] dont l'avantage principal est d'être closes par composition (relationnelle). On peut donc itérer le processus de collapse. Par exemple, le modèle de PCF finitaire dont les objets sont des treillis<sup>22</sup> collapse, via l'inclusion de  $\{\perp, 0, 1, \top\}$  dans  $\{\perp, 0, 1\}$ , sur le modèle de Scott. Ensuite, comme on l'a déjà vu, le modèle de Scott collapse sur le modèle ensembliste, dont les éléments aux types du premier ordre sont les circuits booléens. Par composition, on obtient une relation de collapse du modèle des treillis sur le modèle ensembliste. On sait par ailleurs que le modèle relationnel, qui prend en compte des aspects du calcul autres que la non terminaison et les erreurs, collapse sur le modèle des treillis [Ehr09]. Par composition, on obtient une relation de collapse du modèle relationnel sur le modèle ensembliste.

L'intérêt de cette démarche réside surtout dans le fait que, si on a une situation de collapse extensionnel entre deux modèles  $\mathcal{M}$  et  $\mathcal{N}$ , les éléments de  $\mathcal{N}$  peuvent être vus comme des algorithmes qui implémentent les éléments de  $\mathcal{M}$ , et utilisés pour analyser ces derniers. Une situation de ce type a été illustrée plus haut, pour les circuits booléens et les fonctions monotones qui les implémentent, dans le cas du collapse par totalité. Les notions d'algorithme et de fonctions sont ici relatives aux modèles qu'on considère, qui déterminent où se situe la frontière entre l'une et l'autre.

### 3.1.9 Perspectives

Mon projet de recherche à court terme se base sur les thèmes exposés dans la section précédente. En font partie en particulier l'étude des calculs avec ressources

---

22. Le plus grand élément du treillis des valeurs de vérité,  $\top$ , dénote typiquement une erreur.

et avec motifs, et du non déteterminisme, dans un cadre fonctionnel. D'autres thèmes proches, comme les extensions probabilistes du calcul, conventionnel ou avec ressources, m'intéressent. Plus généralement, les extensions qui rélevent de l'aller-retour fructueux entre syntaxe et sémantique que j'ai cité dans l'introduction à ce texte, s'integrent assez naturellement dans mon projet.

Au-delà, un programme à plus long termes peut porter sur des applications des modèles dénotationnels, comme l'interprétation abstraite ou l'étude de la complexité.

### 3.2 Activité d'encadrement

- 2002-2005 : direction du stage de DEA, puis de la thèse de Benjamin Leperchey. Titre de la thèse : Sur la notion d'observation en sémantique. Thèse soutenue en décembre 2005.
- 2004-2006 : co-direction (avec D. Kesner) du stage de DEA, puis de la thèse de Caroline Priou. Titre de la thèse : Sémantique des motifs. Abandon en 2006.
- 2006-2009 : co-direction (avec Lorenzo Tortora) de la thèse de Paolo Tranquilli, en co-tutelle avec l'Université de Rome 3. Titre de la thèse : Nets between Determinism and Nondeterminism. Thèse soutenue en avril 2009.
- 2008-2011 : co-direction (avec Antonino Salibra) de la thèse de Alberto Carraro. Titre de la thèse : Models and theories of pure and resource lambda calculi. Thèse soutenue en avril 2011.
- depuis 2011 : direction de la thèse de Flavien Breuvert. Titre de la thèse : Syntaxe et sémantique des tests en  $\lambda$ -calcul.
- direction de "Travaux de Recherche Encadrés" au niveau M1 à Paris Diderot, et de "Tesi di Laurea" à l'Université de Roma - La Sapienza.

### 3.3 Publications

#### Revues internationales :

[BE93] Antonio Bucciarelli and Thomas Ehrhard. A theory of sequentiality. *Theor. Comput. Sci.*, 113(2):273–291, 1993

[BE94] Antonio Bucciarelli and Thomas Ehrhard. Sequentiality in an extensional framework. *Inf. Comput.*, 110(2):265–296, 1994

[Buc97b] Antonio Bucciarelli. Degrees of parallelism in the continuous type hierarchy. *Theor. Comput. Sci.*, 177(1):59–71, 1997

[Buc97a] Antonio Bucciarelli. Bi-models: Relational versus domain-theoretic approaches. *Fundam. Inform.*, 32(3-4):251–266, 1997

- [**BE00**] Antonio Bucciarelli and Thomas Ehrhard. On phase semantics and denotational semantics in multiplicative-additive linear logic. *Ann. Pure Appl. Logic*, 102(3):247–282, 2000
- [**BE01**] Antonio Bucciarelli and Thomas Ehrhard. On phase semantics and denotational semantics: the exponentials. *Ann. Pure Appl. Logic*, 109(3):205–241, 2001
- [**BM02**] Antonio Bucciarelli and Pasquale Malacaria. Relative definability of boolean functions via hypergraphs. *Theor. Comput. Sci.*, 278(1-2):91–110, 2002
- [**BPS03**] Antonio Bucciarelli, Adolfo Piperno, and Ivano Salvo. Intersection types and lambda-definability. *Mathematical Structures in Computer Science*, 13(1):15–53, 2003
- [**BS08**] Antonio Bucciarelli and Antonino Salibra. Graph lambda theories. *Mathematical Structures in Computer Science*, 18(5):975–1004, 2008
- [**BEM12**] Antonio Bucciarelli, Thomas Ehrhard, and Giulio Manzonetto. A relational semantics for parallelism and non-determinism in a functional setting. *Ann. Pure Appl. Logic*, 163(7):918–934, 2012

#### **Conférences internationales :**

- [**BE91b**] Antonio Bucciarelli and Thomas Ehrhard. Sequentiality and strong stability. In *Logic in Computer Science*, pages 138–145. IEEE Computer Society, 1991
- [**BE91a**] Antonio Bucciarelli and Thomas Ehrhard. Extensional embedding of a strongly stable model of PCF. In Javier Leach Albert, Burkhard Monien, and Mario Rodríguez-Artalejo, editors, *ICALP*, volume 510 of *Lecture Notes in Computer Science*, pages 35–46. Springer, 1991
- [**Buc93a**] Antonio Bucciarelli. Another approach to sequentiality: Kleene’s unimotone functions. In *Mathematical Foundations of Programming Semantics*, volume 802 of *Lecture Notes in Computer Science*, pages 333–358. Springer, 1993
- [**Buc97c**] Antonio Bucciarelli. Logical reconstruction of bi-domains. In Philippe de Groote, editor, *TLCA*, volume 1210 of *Lecture Notes in Computer Science*, pages 99–111. Springer, 1997
- [**BS98**] Antonio Bucciarelli and Ivano Salvo. Totality, definability and boolean circuits. In Kim Guldstrand Larsen, Sven Skyum, and Glynn Winskel, editors, *ICALP*, volume 1443 of *Lecture Notes in Computer Science*, pages 808–819. Springer, 1998
- [**BLPS99**] Antonio Bucciarelli, Silvia De Lorenzis, Adolfo Piperno, and Ivano Salvo. Some computational properties of intersection types. In *Logic in Computer Science*, pages 109–118, 1999

[BLP03] Antonio Bucciarelli, Benjamin Leperchey, and Vincent Padovani. Relative definability and models of unary pcf. In Martin Hofmann, editor, *Typed Lambda-Calculi and Applications*, volume 2701 of *Lecture Notes in Computer Science*, pages 75–89. Springer, 2003

[BS03] Antonio Bucciarelli and Antonino Salibra. The minimal graph model of lambda calculus. In Branislav Rován and Peter Vojtás, editors, *Mathematical Foundations of Computer Science*, volume 2747 of *Lecture Notes in Computer Science*, pages 300–307. Springer, 2003

[BL04] Antonio Bucciarelli and Benjamin Leperchey. Hypergraphs and degrees of parallelism: A completeness result. In Igor Walukiewicz, editor, *FoSSaCS*, volume 2987 of *Lecture Notes in Computer Science*, pages 58–71. Springer, 2004

[BS04] Antonio Bucciarelli and Antonino Salibra. The sensible graph theories of lambda calculus. In *Logic in Computer Science*, pages 276–285. IEEE Computer Society, 2004

[BEM07] Antonio Bucciarelli, Thomas Ehrhard, and Giulio Manzonetto. Not enough points is enough. In Jacques Duparc and Thomas A. Henzinger, editors, *CSL*, volume 4646 of *Lecture Notes in Computer Science*, pages 298–312. Springer, 2007

[BEM09a] Antonio Bucciarelli, Thomas Ehrhard, and Giulio Manzonetto. A relational model of a parallel and non-deterministic lambda-calculus. In Sergei N. Artëmov and Anil Nerode, editors, *LFCS*, volume 5407 of *Lecture Notes in Computer Science*, pages 107–121. Springer, 2009

[BEM10] Antonio Bucciarelli, Thomas Ehrhard, and Giulio Manzonetto. Categorical models for simply typed resource calculi. In *Mathematical Foundations of Programming Semantics*, volume 265 of *ENTCS*, pages 213–230, 2010

[BCEM11] Antonio Bucciarelli, Alberto Carraro, Thomas Ehrhard, and Giulio Manzonetto. Full abstraction for resource calculus with tests. In Marc Bezem, editor, *CSL*, volume 12 of *LIPICs*, pages 97–111. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2011

#### **Workshops internationaux avec actes :**

[Buc94a] Antonio Bucciarelli. Degrees of parallelism in the continuous type hierarchy. Proc. 10th Int. Workshop on Mathematical Foundation of Progr. Semantics, Manhattan (KS), 1994

[BM96] Antonio Bucciarelli and Pasquale Malacaria. Relative definability of boolean functions via hypergraphs. Proc. 12th Int. Workshop on Mathematical Foundation of Progr. Semantics, Boulder (CO), 1996

[Buc96] Antonio Bucciarelli. Logical relations and lambda-theories. In *3rd Imperial-College Workshop on the Advances in Theory and Formal Methods of Computing*, pages 37–48. Imperial College Press, 1996

[**BCES10**] Antonio Bucciarelli, Alberto Carraro, Thomas Ehrhard, and Antonino Salibra. On linear information systems. In *First International Workshop on Linearity*, volume 22 of *EPTCS*, pages 38–48, 2010

#### **Mémoires :**

[**Buc87**] Antonio Bucciarelli. Teoria generalizzata della ricorsività in alcuni modelli del  $\lambda$ -calcolo. Tesi di Laurea, Università di Pisa, 1987

[**Buc93b**] Antonio Bucciarelli. *Sequential models of PCF: some contributions to the domain theoretic approach to full abstraction*. PhD thesis, TD-3/93, Dipartimento di informatica, Università di Pisa, 1993

[**Buc94b**] Antonio Bucciarelli. *Modèles denotationnels des langages séquentiels*. PhD thesis, Université Paris 11, 1994

[**Buc09**] Antonio Bucciarelli. Sur la  $\lambda$ -définissabilité. Mémoire d’habilitation à diriger les recherches, Université Paris Diderot, 2009

#### **Notes, Rapports techniques**

(autres que versions préliminaires d’articles ci-dessus) :

[**Buc88**] Antonio Bucciarelli. An extension of PCF. Unpublished manuscript, 1988

[**Buc95**] Antonio Bucciarelli. Event structures representing domains with coherence. Rapport de recherche du laboratoire d’informatique de l’ENS Paris, LIENS-95-16, 1995

[**Buc11**] Antonio Bucciarelli. Extensional collapse situations i: non-termination and unrecoverable errors. *CoRR*, abs/1101.4465, 2011

#### **Articles soumis :**

[**BCEM**] Antonio Bucciarelli, Alberto Carraro, Thomas Ehrhard, and Giulio Manzonetto. Full abstraction for resource calculus with tests. submitted

#### **Articles en préparation :**

[**Buc**] Antonio Bucciarelli. Extensional Collapse Situations II: linearity.

[**BKdR**] Antonio Bucciarelli, Delia Kesner, and Simona Ronchi della Rocca. Solvability in a simple pattern calculus

[**BDJ**] Antonio Bucciarelli, Gilles Dowek, and Ying Jiang. On the expressive power of schemes

## **4 Tâches administratives et collectives**

### **4.1 Responsabilité pédagogiques**

- 2002-03 : responsable du tutorat, UFR d’informatique, Paris Diderot.
- 2005-06 : responsable des stages de M1, UFR d’informatique, Paris Diderot.
- depuis 2006 : responsable des relations internationales, UFR d’informatique, Paris Diderot.

### **4.2 Charges collectives**

- 1997-99 : responsable du projet “biblioteca digitale”, département d’informatique, Univ. de Rome “La Sapienza”.
- 1999-2003 : membre du conseil de l’UFR d’informatique, Paris Diderot.
- 1999-2007 : membre de la commission des spécialistes, section 27, Paris Diderot (suppléant de 2004 à 2007).

### **4.3 Animation scientifique**

- 1999-2006 : responsable du séminaire hebdomadaire, puis du groupe de travail *Sémantique* du laboratoire Preuves, Programmes et Systèmes,
- (co-) organisation de colloques, rencontres : Annual Meeting du projet Categorical Logic in Comp. Science, Paris, 1994 ; Final Meeting du projet Gentzen, Roma ; 1997 ; 12th Int. Workshop on Unification (Unif 98), Roma, 1998 ; Final Meeting du projet Logique et sémantique des langages de programmation, Paris 2003 ; Rencontre du projet Geocal, Paris, 2005 ; Federated Conference on Rewriting, Deduction and Programming (RDP 2007), Paris, 2007.

## Références

- [AJM00] Samson Abramsky, Radha Jagadeesan, and Pasquale Malacaria. Full abstraction for PCF. *Inf. Comput.*, 163(2) :409–470, 2000.
- [BC82] Gérard Berry and Pierre-Louis Curien. Sequential algorithms on concrete data structures. *Theor. Comput. Sci.*, 20 :265–321, 1982.
- [BCEM] Antonio Bucciarelli, Alberto Carraro, Thomas Ehrhard, and Giulio Manzonetto. Full abstraction for resource calculus with tests. submitted.
- [BCEM11] Antonio Bucciarelli, Alberto Carraro, Thomas Ehrhard, and Giulio Manzonetto. Full abstraction for resource calculus with tests. In Marc Bezem, editor, *CSL*, volume 12 of *LIPICs*, pages 97–111. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2011.
- [BCES10] Antonio Bucciarelli, Alberto Carraro, Thomas Ehrhard, and Antonino Salibra. On linear information systems. In *First International Workshop on Linearity*, volume 22 of *EPTCS*, pages 38–48, 2010.
- [BDJ] Antonio Bucciarelli, Gilles Dowek, and Ying Jiang. On the expressiva power of schemes.
- [BE91a] Antonio Bucciarelli and Thomas Ehrhard. Extensional embedding of a strongly stable model of PCF. In Javier Leach Albert, Burkhard Monien, and Mario Rodríguez-Artalejo, editors, *ICALP*, volume 510 of *Lecture Notes in Computer Science*, pages 35–46. Springer, 1991.
- [BE91b] Antonio Bucciarelli and Thomas Ehrhard. Sequentiality and strong stability. In *Logic in Computer Science*, pages 138–145. IEEE Computer Society, 1991.
- [BE93] Antonio Bucciarelli and Thomas Ehrhard. A theory of sequentiality. *Theor. Comput. Sci.*, 113(2) :273–291, 1993.
- [BE94] Antonio Bucciarelli and Thomas Ehrhard. Sequentiality in an extensional framework. *Inf. Comput.*, 110(2) :265–296, 1994.
- [BE00] Antonio Bucciarelli and Thomas Ehrhard. On phase semantics and denotational semantics in multiplicative-additive linear logic. *Ann. Pure Appl. Logic*, 102(3) :247–282, 2000.
- [BE01] Antonio Bucciarelli and Thomas Ehrhard. On phase semantics and denotational semantics : the exponentials. *Ann. Pure Appl. Logic*, 109(3) :205–241, 2001.

- [BEM07] Antonio Bucciarelli, Thomas Ehrhard, and Giulio Manzonetto. Not enough points is enough. In Jacques Duparc and Thomas A. Henzinger, editors, *CSL*, volume 4646 of *Lecture Notes in Computer Science*, pages 298–312. Springer, 2007.
- [BEM09a] Antonio Bucciarelli, Thomas Ehrhard, and Giulio Manzonetto. A relational model of a parallel and non-deterministic lambda-calculus. In Sergei N. Artëmov and Anil Nerode, editors, *LFCS*, volume 5407 of *Lecture Notes in Computer Science*, pages 107–121. Springer, 2009.
- [BEM09b] Antonio Bucciarelli, Thomas Ehrhard, and Giulio Manzonetto. A relational semantics of parallelism and non-determinism in a functional setting. submitted, 2009.
- [BEM10] Antonio Bucciarelli, Thomas Ehrhard, and Giulio Manzonetto. Categorical models for simply typed resource calculi. In *Mathematical Foundations of Programming Semantics*, volume 265 of *ENTCS*, pages 213–230, 2010.
- [BEM12] Antonio Bucciarelli, Thomas Ehrhard, and Giulio Manzonetto. A relational semantics for parallelism and non-determinism in a functional setting. *Ann. Pure Appl. Logic*, 163(7) :918–934, 2012.
- [Ber78] Gérard Berry. Stable models of typed lambda-calculi. In Giorgio Ausiello and Corrado Böhm, editors, *ICALP*, volume 62 of *Lecture Notes in Computer Science*, pages 72–89. Springer, 1978.
- [Ber79] Gérard Berry. *Modèles complètement adéquats et stables des lambda-calculs typés*. PhD thesis, Université Paris VII, 1979.
- [BKdR] Antonio Bucciarelli, Delia Kesner, and Simona Ronchi della Rocca. Solvability in a simple pattern calculus.
- [BL04] Antonio Bucciarelli and Benjamin Leperchey. Hypergraphs and degrees of parallelism : A completeness result. In Igor Walukiewicz, editor, *FoSSaCS*, volume 2987 of *Lecture Notes in Computer Science*, pages 58–71. Springer, 2004.
- [BLP03] Antonio Bucciarelli, Benjamin Leperchey, and Vincent Padovani. Relative definability and models of unary pcf. In Martin Hofmann, editor, *Typed Lambda-Calculi and Applications*, volume 2701 of *Lecture Notes in Computer Science*, pages 75–89. Springer, 2003.
- [BLPS99] Antonio Bucciarelli, Silvia De Lorenzis, Adolfo Piperno, and Ivano Salvo. Some computational properties of intersection types. In *Logic in Computer Science*, pages 109–118, 1999.

- [BM96] Antonio Bucciarelli and Pasquale Malacaria. Relative definability of boolean functions via hypergraphs. Proc. 12th Int. Workshop on Mathematical Foundation of Progr. Semantics, Boulder (CO), 1996.
- [BM02] Antonio Bucciarelli and Pasquale Malacaria. Relative definability of boolean functions via hypergraphs. *Theor. Comput. Sci.*, 278(1-2) :91–110, 2002.
- [BPS03] Antonio Bucciarelli, Adolfo Piperno, and Ivano Salvo. Intersection types and lambda-definability. *Mathematical Structures in Computer Science*, 13(1) :15–53, 2003.
- [BS98] Antonio Bucciarelli and Ivano Salvo. Totality, definability and boolean circuits. In Kim Guldstrand Larsen, Sven Skyum, and Glynn Winskel, editors, *ICALP*, volume 1443 of *Lecture Notes in Computer Science*, pages 808–819. Springer, 1998.
- [BS03] Antonio Bucciarelli and Antonino Salibra. The minimal graph model of lambda calculus. In Branislav Rován and Peter Vojtás, editors, *Mathematical Foundations of Computer Science*, volume 2747 of *Lecture Notes in Computer Science*, pages 300–307. Springer, 2003.
- [BS04] Antonio Bucciarelli and Antonino Salibra. The sensible graph theories of lambda calculus. In *Logic in Computer Science*, pages 276–285. IEEE Computer Society, 2004.
- [BS08] Antonio Bucciarelli and Antonino Salibra. Graph lambda theories. *Mathematical Structures in Computer Science*, 18(5) :975–1004, 2008.
- [Buc] Antonio Bucciarelli. Extensional Collapse Situations II : linearity.
- [Buc87] Antonio Bucciarelli. Teoria generalizzata della ricorsività in alcuni modelli del  $\lambda$ -calcolo. Tesi di Laurea, Università di Pisa, 1987.
- [Buc88] Antonio Bucciarelli. An extension of PCF. Unpublished manuscript, 1988.
- [Buc93a] Antonio Bucciarelli. Another approach to sequentiality : Kleene’s unimotone functions. In *Mathematical Foundations of Programming Semantics*, volume 802 of *Lecture Notes in Computer Science*, pages 333–358. Springer, 1993.
- [Buc93b] Antonio Bucciarelli. *Sequential models of PCF : some contributions to the domain theoretic approach to full abstraction*. PhD thesis, TD-3/93, Dipartimento di informatica, Università di Pisa, 1993.
- [Buc94a] Antonio Bucciarelli. Degrees of parallelism in the continuous type hierarchy. Proc. 10th Int. Workshop on Mathematical Foundation of Progr. Semantics, Manhattan (KS), 1994.

- [Buc94b] Antonio Bucciarelli. *Modèles denotationnels des langages séquentiels*. PhD thesis, Université Paris 11, 1994.
- [Buc95] Antonio Bucciarelli. Event structures representing domains with coherence. Rapport de recherche du laboratoire d’informatique de l’ENS Paris, LIENS-95-16, 1995.
- [Buc96] Antonio Bucciarelli. Logical relations and lambda-theories. In *3rd Imperial-College Workshop on the Advances in Theory and Formal Methods of Computing*, pages 37–48. Imperial College Press, 1996.
- [Buc97a] Antonio Bucciarelli. Bi-models : Relational versus domain-theoretic approaches. *Fundam. Inform.*, 32(3-4) :251–266, 1997.
- [Buc97b] Antonio Bucciarelli. Degrees of parallelism in the continuous type hierarchy. *Theor. Comput. Sci.*, 177(1) :59–71, 1997.
- [Buc97c] Antonio Bucciarelli. Logical reconstruction of bi-domains. In Philippe de Groote, editor, *TLCA*, volume 1210 of *Lecture Notes in Computer Science*, pages 99–111. Springer, 1997.
- [Buc09] Antonio Bucciarelli. Sur la  $\lambda$ -définissabilité. Mémoire d’habilitation à diriger les recherches, Université Paris Diderot, 2009.
- [Buc11] Antonio Bucciarelli. Extensional collapse situations i : non-termination and unrecoverable errors. *CoRR*, abs/1101.4465, 2011.
- [CE94] Loïc Colson and Thomas Ehrhard. On strong stability and higher-order sequentiality. In *Proceedings of the 9th Annual IEEE Symposium on Logic in Computer Science*. IEEE Computer Society, 1994.
- [Ehr09] Thomas Ehrhard. The Scott model of linear logic is the extensional collapse of its relational model. Technical report, Preuves, Programmes et Systèmes, 2009. Submitted for publication.
- [ER03] Thomas Ehrhard and Laurent Regnier. The differential lambda-calculus. *Theoretical Computer Science*, 309(1-3) :1–41, 2003.
- [ER06] Thomas Ehrhard and Laurent Regnier. Differential interaction nets. *Theoretical Computer Science*, 364(2) :166–195, 2006.
- [Gir87] Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50 :1–102, 1987.
- [HO00] Martin Hyland and Luke Ong. On Full Abstraction for PCF : I, II, and III. *Inf. Comput.*, 163(2) :285–408, 2000.
- [HS02] Furio Honsell and Donald Sannella. Prelogical relations. *Inf. Comput.*, 178(1) :23–43, 2002.

- [JK06] C. Barry Jay and Delia Kesner. Pure pattern calculus. In Peter Sestoft, editor, *ESOP*, volume 3924 of *Lecture Notes in Computer Science*, pages 100–114. Springer, 2006.
- [Loa01] Ralph Loader. Finitary PCF is not decidable. *Theor. Comput. Sci.*, 266(1-2) :341–364, 2001.
- [Lon02] John Longley. The sequentially realizable functionals. *Ann. Pure Appl. Logic*, 117(1-3) :1–93, 2002.
- [Mil77] Robin Milner. Fully abstract models of typed *lambda*-calculi. *Theor. Comput. Sci.*, 4(1) :1–22, 1977.
- [Nic94] Hanno Nickau. Hereditarily sequential functionals. In Anil Nerode and Yuri Matiyasevich, editors, *LFCS*, volume 813 of *Lecture Notes in Computer Science*, pages 253–264. Springer, 1994.
- [OR95] Peter W. O’Hearn and Jon G. Riecke. Kripke logical relations and PCF. *Information and Computation*, 120(1) :107–116, 1995.
- [Plo77] Gordon D. Plotkin. LCF considered as a programming language. *Theor. Comput. Sci.*, 5(3) :225–255, 1977.
- [Sco93] Dana S. Scott. A type-theoretical alternative to ISWIM, CUCH, OWHY. *Theor. Comput. Sci.*, 121(1&2) :411–440, 1993.
- [Sie92] Kurt Sieber. Reasoning about sequential functions via logical relations. In M. Fourman, P. Johnstone, and A.Pitts, editors, *Proceedings of the LMS Symposium on Applications of Categories in Computer Science*, volume 177 of *LMS Lecture Notes Series*. Cambridge University Press, 1992.
- [vO99] Jaap van Oosten. A combinatory algebra for sequential functionals of finite types. In S.B. Cooper and J.K. Truss, editors, *Models and Computability*, pages 389–406. Cambridge University Press, 1999.