

# Lison Blondeau-Patissier

👤 ATER à l'ENS de Lyon  
✉ [lison.blondeau-patissier@protonmail.com](mailto:lison.blondeau-patissier@protonmail.com)  
🏠 31 rue de la commune de Paris, 69600 Oullins



## Parcours Universitaire

### Doctorat à l'université d'Aix-Marseille

- 2025 • **Doctorat en Mathématiques. Université d'Aix-Marseille.**  
Titre : *Jeux Concurrents à Pointeurs et Calcul à Ressources*.  
École doctorale : Mathématiques et informatique (ED 184).  
Laboratoires : *Institut de mathématiques de Marseille* et *Laboratoire d'Informatique et des Systèmes*.  
**Directeur de thèse** : Lionel Vaux Auclair.  
*Maître de conférences à Aix-Marseille Université, équipe Logique de la Programmation à l'I2M.*  
**Codirecteur de thèse** : Pierre Clairambault.  
*Directeur de recherche CNRS à Aix-Marseille Université, équipe Logic, Semantics and Categories au LIS.*  
**Rapporteurs** : Giulio Manzonetto (*professeur des universités à Paris Cité*) et Guy McCusker (*professor of Computing à University of Bath*). **Examinatrices** : Claudia Faggian (*chargée de recherche CNRS à Paris Cité*) et Marie Kerjean (*chargée de recherche CNRS à Sorbonne Paris Nord*). **Président du jury** : Stefano Guerrini (*professeur des universités à Sorbonne Paris Nord*).

### Scolarité à l'École Normale Supérieure de Lyon

- 2020 • **Master de sciences, technologies, santé.** École Normale Supérieure de Lyon, UCBL.  
Mention Informatique; Parcours type Informatique Fondamentale.  
Sujet de stage : *Positional Injectivity for Innocent Strategies*.  
Stage encadré par Pierre Clairambault, au sein de l'équipe **PLUME** du **LIP**.
- 2017 • **Licence d'Informatique Fondamentale.** École Normale Supérieure de Lyon, UCBL.  
• **Licence de Mathématiques.** Université Claude Bernard Lyon 1.

## Expérience professionnelle

- 2024–2026 • **ATER à l'École Normale Supérieure de Lyon.**  
Attachée Temporaire d'Enseignement et de Recherche au sein de l'équipe **PLUME** du **LIP**.  
Service de 192h d'enseignement.

## Activités d'enseignement

### Enseignements en ATER (ENS de Lyon, 2024–2025)

- 2025–2026 • **Sémantique et Vérification** (20h TD).  
TD niveau M1, École Normale Supérieure de Lyon.  
*Programme* : modèles mathématiques des programmes et des systèmes, propriétés vérifiables sur ces modèles, topologie, systèmes de transitions, logique temporelle linéaire, automates de Büchi, bisimulations, logique modale, etc.
- **Leçons pour la préparation à l'agrégation** (30h TD).  
Niveau M2 FEADep, École Normale Supérieure de Lyon.  
Corrections de partiels, leçons, etc. notamment en logique.

## Activités d'enseignement (suite)

- **Fondements de l'Informatique** (30h TD).  
TD niveau L3, École Normale Supérieure de Lyon.  
*Programme* : automates finis, langages rationnels, grammaires, langages algébriques et automates à pile, machines de Turing, calculabilité, fonctions récursives et récursives primitives, complexité,  $\lambda$ -calcul, etc.
- **Logique** (30h TD).  
TD niveau L3, École Normale Supérieure de Lyon.  
*Programme* : calcul propositionnel, systèmes de déduction, sémantique, théorie des modèles, représentation, incomplétude, théorie des ensembles, etc.
- **Probabilités** (30h TD).  
TD niveau L3, École Normale Supérieure de Lyon.  
*Programme* : probabilités discrètes, événements, espérance et espérance conditionnelle, variance, variables aléatoires, probabilités continues, inégalités de Markov, Chebyshev et Chernoff, graphes aléatoires, méthode probabiliste, loi des grands nombres, convergences de variables, chaînes de Markov, etc.
- **Programmation** (30h TD).  
TP niveau L3, École Normale Supérieure de Lyon.  
*Programme* : initiation à C, git et Python ; brève histoire de la programmation.  
*Évaluation* : TP noté en C ; projet de programmation en groupe de deux ou trois (langage libre, principalement C et Python) ; projet pédagogique individuel (mini cours, vidéo ou modification/traduction/création d'une page wikipedia sur un aspect de la programmation).
- 2024–2025 • **Sémantique et Vérification** (20h TD).  
TD niveau M1, École Normale Supérieure de Lyon.  
*Programme* : modèles mathématiques des programmes et des systèmes, propriétés vérifiables sur ces modèles, topologie, systèmes de transitions, logique temporelle linéaire, automates de Büchi, bisimulations, logique modale, etc.
- **Fondements de l'informatique** (30h TD).  
TD/TP niveau L3, École Normale Supérieure de Lyon.  
*Programme* : automates finis, langages rationnels, grammaires, langages algébriques et automates à pile, machines de Turing, calculabilité, fonctions récursives et récursives primitives, complexité,  $\lambda$ -calcul, etc.
- **Logique** (30h TD).  
TD niveau L3, École Normale Supérieure de Lyon.  
*Programme* : calcul propositionnel, systèmes de déduction, sémantique, théorie des modèles, représentation, incomplétude, théorie des ensembles, etc.
- **Probabilités** (30h TD).  
TD niveau L3, École Normale Supérieure de Lyon.  
*Programme* : probabilités discrètes, événements, espérance et espérance conditionnelle, variance, variables aléatoires, probabilités continues, inégalités de Markov, Chebyshev et Chernoff, graphes aléatoires, méthode probabiliste, loi des grands nombres, convergences de variables, chaînes de Markov, etc.
- **Projet fonctionnel** (10h TD).  
TD niveau L3, École Normale Supérieure de Lyon.  
*Programme* :  $\lambda$ -calcul,  $\beta$ -réduction, types simples, déduction naturelle, Curry-Howard, preuves, déduction naturelle intuitionniste, polymorphisme, négation, etc.

## Activités d'enseignement (suite)

- **Théorie de la programmation** (30h TD).  
TD/TP niveau L3, École Normale Supérieure de Lyon.  
*Programme* : inductions, expressions arithmétiques simples, sémantiques à petits pas et grands pas, typage, sémantiques opérationnelles et dénotationnelle, coinduction, logique de Floyd-Hoare, logique de séparation, réécriture, terminaison, confluence, etc. Initiation à Rocq.

## Enseignements durant le doctorat (Aix-Marseille Université, 2021–2024)

- 2023–2024
- **Complexité** (20h TD et 14hTD TP).  
TD/TP niveau M1, Aix-Marseille Université.  
*Programme* : analyse de complexité, classes P et NP, NP-complétude, mini-projets de programmation (Python, Java ou C).
  - **Aspects probabilistes pour l'info.** (5,87hTD TP).  
TP niveau M1, Aix-Marseille Université.  
*Programme* : Programmation d'algorithmes probabilistes.
  - **Automates finis** (15h TD et 6hTD TP).  
TD/TP niveau L2, Aix-Marseille Université.  
*Programme* : AFD-AFN, déterminisation, minimisation, langages réguliers, etc.
- 2022–2023
- **Mise en oeuvre informatique** (15h TD et 29hTD TP).  
TD/TP niveau L1, Aix-Marseille Université.  
*Programme* : programmation simple en Python, notions d'algorithme et de représentation des données.
- 2021–2022
- **Introduction à l'analyse** (33hTD CM et 18h TD).  
Cours/TD niveau L1 (cycle préparatoire [PEIP](#)) à Polytech Marseille.  
*Programme* : applications, quantificateurs, ensembles, bijectivité, fonctions trigonométriques, limites, continuité, dérivées, primitives.

## Encadrement, service pédagogique, médiation scientifique, etc.

- nov. 2025
- Encadrement de la visite de laboratoires à Sophia-Antipolis, visite de deux jours de découverte d'un laboratoire de recherche, niveau L3 (10hTD). École Normale Supérieure de Lyon.
- 2024–2026
- Soutien dans diverses matières niveau L3 (12hTD). École Normale Supérieure de Lyon.
  - Jury de soutenance de stage niveau L3 et M1. École Normale Supérieure de Lyon.
- jan. 2025
- Encadrement de la *semaine sport études*, une semaine de conférences d'introduction à la recherche en informatique niveau L3 (24hTD). École Normale Supérieure de Lyon.
- 2021–2023
- Tutrice lors de stages *Hippocampe* (36hTD). Stages d'initiation à la recherche en science sur trois jours pour une classe de 3ème ou de lycée, organisés par l'[Institut de Recherche pour l'Enseignement des Sciences](#). Aix-Marseille Université.
- avril 2022
- « Speed-meetings » des *Journées Filles, Maths et Info* organisées par [Animath](#) et [femmes & mathématiques](#). Aix-Marseille Université.
- fév. 2020
- « Speed-meetings » des *Rendez-vous des Jeunes Mathématiciennes* organisés par [Animath](#) et [femmes & mathématiques](#). École Normale Supérieure de Lyon.
- août 2019
- Encadrement de *Girls Can Code*. Stage d'une semaine d'initiation à l'informatique pour collégiennes et lycéennes, organisé par [Prologin](#). École Normale Supérieure de Lyon.
- 2018–2019
- Cours particuliers (mathématiques, physique, science de la vie et de la terre, français, histoire-géographie) niveau lycée. Lyon.
- déc. 2017
- Encadrement des *Rendez-vous des Jeunes Mathématiciennes* organisés par [Animath](#) et [femmes & mathématiques](#). École Normale Supérieure de Lyon.

## Projet d'enseignement

J'apprécie beaucoup l'enseignement, et plus généralement les échanges autour des mathématiques et de l'informatique, que ce soit dans un cadre universitaire ou plus informellement lors de stages de découverte.

J'ai notamment donné des cours, TD et TP auprès de publics de niveau licence (cycle préparatoire PEIP, L1, L2, et L3) et master (M1), que ce soit en mathématiques, en informatique fondamentale, ou en programmation, et je me sens donc capable d'assurer des enseignements variés dans ces domaines.

## Projet de recherche

### Contexte général

Je m'intéresse aux sémantiques dénotationnelles des langages de programmation, en particulier à la sémantique des jeux. On étudie donc des propriétés formelles du comportement des programmes, en représentant les exécutions par des objets mathématiques afin de mieux comprendre leurs structures et leurs propriétés.

**Sémantique des jeux.** La sémantique des jeux est née dans les années 1990 afin d'étudier les programmes dans PCF (*Programming Computable Function*, un langage fonctionnel typé – voir [10] ou [13]). Certains des premiers modèles pleinement abstraits pour PCF (c'est à dire tels que tous les programmes observationnellement équivalents ont la même sémantique) sont des modèles de jeux : les jeux Hyland-Ong/Nickau (HO/N), introduits indépendamment dans [8] et dans [9], et les jeux Abramsky-Jagadeesan-Malacaria, dans [7]. Ce domaine de recherche a depuis connu de nombreux développements, avec par exemples des extensions à des modèles non-déterministe, probabilistes, etc.

Les exécutions de programmes sont représentés par des interactions entre un programme et son environnement, modélisées comme des **parties** d'un « jeu » entre **Joueur**, le programme, et **Opposant**, l'environnement. Un programme est alors représenté par une **stratégie**, l'ensemble des parties modélisant les exécutions possibles du programme. On s'intéressera ici en particulier aux stratégies **innocentes**, qui correspondent aux programmes purs n'ayant pas d'accès à l'état (ou aux  $\lambda$ -termes sans références). Enfin, les types sont représentés par des **arènes**, qui donnent les « règles du jeu », c'est-à-dire les interactions possibles en fonction du type.

**Modèle relationnel.** Une autre approche en sémantique est de s'intéresser aux aspects *statiques* des programmes uniquement, par exemple avec le modèle relationnel [11] qui représente les types comme des ensembles et les programmes comme des relations. On peut voir les éléments de ce modèle comme des parties déséquentialisées, où on oublie l'information temporelle.

Les stratégies innocentes oubliant déjà une partie de l'information temporelle (l'ordonnancement des actions d'Opposant), on peut s'intéresser à la *positionnalité* des stratégies innocentes : le comportement d'une stratégie dépend-t-il uniquement de l'ensemble des actions déjà jouées, sans aucune temporelle ? C'est le cas pour les jeux asynchrones [5]. Pour les jeux Hyland-Ong, on sait que les stratégies innocentes sont **positionnellement injectives**, c'est-à-dire caractérisées uniquement par leur image dans le modèle relationnel [2], cependant cette correspondance nécessite d'interpréter les types de base avec des ensembles dénombrables, afin d'encoder l'information causale dans l'interprétation. **Question 1 :** *Les stratégies innocentes sont-elles positionnellement innocentes si on interprète les types de bases par des singletons ?*

**Calcul à ressources.** Le  $\lambda$ -calcul, introduit par Church dans les années 1930 (voir [14] pour la référence historique et [12] pour une introduction plus détaillée) est un langage de programmation formel, qui permet d'étudier des propriétés des langages de programmation dans un cadre mathématique très simplifié. En particulier, il n'y a pas de borne sur le nombre de fois où un programme peut appeler son argument. Le calcul à ressource permet de représenter des exécutions finies, en comptant précisément ces appels et en considérant chaque terme comme une *ressource* qui doit être utilisée une seule et unique fois.

On s'intéresse au lien entre les termes à ressources et les parties en sémantique des jeux. Il existe déjà une bijection entre les termes en forme  $\beta$ -normale et  $\eta$ -longue d'une part, et les parties quotientées par une relation d'homotopie (définie en [5]) d'autre part [2] ; mais cette bijection passe par le modèle relationnel. **Question 2 :** *Peut-on comprendre cette correspondance entre jeux et termes à ressources de façon plus directe ?* De plus, elle se concentre sur l'aspect statique de l'interprétation, et ne détaille pas les aspects dynamiques. **Question 3 :** *Que dire de l'interprétation des termes qui ne sont pas en forme normale ?*

## Travail de recherche réalisé durant la thèse

**Jeux Concurrents à Pointeurs (JCP).** Puisque les stratégies innocentes « n'ont pas besoin de toute l'information temporelle », l'information essentielle dans une partie n'est pas la chronologique exacte mais uniquement les liens de causalité de la stratégie. Nous avons donc dans un premier temps développé un modèle de jeux, inspiré des jeux concurrents (présentés en détail dans [1]), où les objets principaux sont les **augmentations**, qui correspondent aux parties quotientées par l'ordonnancement des actions d'Opposant. On a alors un cadre de travail adapté à l'étude des structures causales des stratégies innocentes, tout en conservant un lien détaillé avec les jeux Hyland-Ong.

**Injectivité positionnelle.** Avec cette première formulation statique des JCP, on montre que les stratégies innocentes *totale et finies* sont positionnellement injectives au sens de la [Question 1](#). En revanche, les stratégies innocentes partielles infinies ne le sont pas en général. Ces résultats sont d'abord obtenus pour les JCP puis traduits pour les jeux Hyland-Ong.

**Lien statique avec le calcul à ressources.** En définissant un calcul à ressources *extensionnel*, c'est-à-dire typé de façon à ce que les termes en forme normale soient également en forme  $\eta$ -longue, on montre que les termes normaux sont en bijection avec les (classes d'isomorphisme des) augmentations dans les JCP – ce qui nous donne la correspondance directe attendue à la [Question 2](#).

**Dynamique des JCP.** On étend les JCP avec une notion de composition. La composition d'augmentations produit une somme d'augmentation – ce qui rappelle les sommes obtenues *via* la substitution dans le calcul à ressources. On va donc définir les stratégies dans JCP comme des sommes d'augmentations, et on montre que les JCP forment une catégorie symétrique monoïdale fermée (avec les stratégies pour morphismes et les arènes pour objets). La correspondance entre PCG et HO s'étend en un foncteur cartésien fermé strict.

**Catégories à ressources.** Pour étudier l'interprétation du calcul à ressources dans PCG, on cherche à comprendre plus précisément sa structure catégorique. Les termes du calcul à ressources sont généralement interprétés dans des catégories *différentielles* [4], mais contrairement aux morphismes des catégories différentielles et aux termes à ressources, les stratégies des JCP n'ont *pas* en général un comportement linéaire. En s'inspirant des catégories différentielles, on définit donc les *catégories à ressources*, qui nous permettent à la fois de considérer des morphismes « non-linéaires » en général et d'identifier les morphismes qui *ont* un comportement « linéaire ». On obtient une interprétation correcte (compatible avec la  $\beta$ -réduction) des termes à ressource dans une catégorie à ressources.

**Interprétation.** Enfin, on vérifie que les JCP forment bien une catégorie à ressources. On a alors l'interprétation dynamique voulue à la [Question 3](#) ; et on vérifie de plus que l'interprétation d'un terme en forme normale coïncide avec la bijection établie précédemment. Autrement dit, quand on normalise un terme à ressource, on calcule sa sémantique dans les jeux concurrents à pointeurs.

## Projets de recherche futurs

Je souhaite poursuivre les travaux réalisés durant ma thèse sur les liens entre syntaxes et sémantiques, et entre différents modèles des langages de programmation. En particulier, des recherches sont en cours sur deux axes dans le prolongement de la thèse : développement de Taylor et calculs non typés.

**Développement de Taylor.** Le développement de Taylor [3] d'un  $\lambda$ -terme est la somme des ses approximations sous forme de termes à ressources. On aimerait donc étendre l'interprétation du calcul à ressource dans les JCP aux  $\lambda$ -termes, de sorte que l'interprétation d'un  $\lambda$ -terme soit égale à l'interprétation de son développement de Taylor. Cela nous demande de définir un développement de Taylor pour le calcul à ressource extensionnel, ainsi que l'interprétation des  $\lambda$ -termes dans les JCP.

**Calculs non typés.** Jusqu'à présent, les travaux présentés portent sur le cas *typé*. On sait qu'on peut interpréter les termes *non typés* dans les jeux Hyland-Ong [6], mais qu'en est-il des termes à ressources ? Nous travaillons à une version non typée du calcul à ressources extensionnel, avec le développement de Taylor correspondant.

Plus généralement, j'aimerais étudier les liens entre ces modèles et d'autres, par exemple les jeux concurrents, des modèles probabilistes, d'autres variantes du  $\lambda$ -calcul, etc.

## Publications

---

### Revue

- [BCVA25] • *Strategies as Resources Terms, and Their Categorical Semantics*. (Version longue de [BCVA23].)  
Lison Blondeau-Patissier, Pierre Clairambault et Lionel Vaux Auclair.  
Logical Methods in Computer Science (LMCS), 2025.  
**Résumé :** On définit le calcul à ressources extensionnel et on montre la bijection entre augmentations (introduites en [BC21]) et termes en forme normale. On définit la composition dans les jeux concurrents à pointeurs, afin de montrer que les JCP forment une catégorie. On définit les catégories à ressources, avec l'interprétation du calcul à ressources. Enfin, on montre que les JCP forment une catégorie à ressources, et que l'interprétation d'un terme calcule sa forme normale.

### Actes de conférences internationales

- [BCVA23] • *Strategies as Resource Terms, and Their Categorical Semantics*.  
Lison Blondeau-Patissier, Pierre Clairambault et Lionel Vaux Auclair.  
Formal Structures for Computation and Deduction (FSCD), 2023.
- [BC21] • *Positional Injectivity for Innocent Strategies*.  
Lison Blondeau-Patissier and Pierre Clairambault.  
Formal Structures for Computation and Deduction (FSCD), 2021.  
**Résumé :** On présente les augmentations, afin d'étudier les liens entre les stratégies innocentes dans les jeux Hyland-Ong et le modèle relationnel. On montre que les stratégies innocentes totales finies sont positionnellement injectives (d'abord à l'aide des augmentations, puis dans les jeux Hyland-Ong). On donne également un contre-exemple à l'injectivité positionnelle dans le cas des stratégies innocentes (partielles et infinies).

### Actes de conférences nationales

- [B24] • *Resource Categories from Differential Categories*.  
Lison Blondeau-Patissier.  
Journées Francophones des Langages Applicatifs (JFLA), 2024.  
**Résumé :** On présente en détail les catégories à ressources, et on étudie les liens avec les catégories différentielles – en particulier, on montre comment construire une catégorie à ressources à partir d'une catégorie additive monoïdale avec stockage.

## Exposés

---

### Catégories à ressources et catégories différentielles

- jan. 2024 • Journées Francophones des Langages Applicatifs (conférence nationale), Saint-Jacut-de-la-Mer.

### Développement de Taylor et sémantique des jeux

- jan. 2024 • GALOP (workshop international), London (Royaume-Uni).

### Sémantique des jeux, termes à ressources et sémantique catégorique

- oct. 2023 • Séminaire d'équipe LIMD, Chambéry.  
sep. 2023 • Chocla (séminaire national), Lyon.  
sep. 2023 • Séminaire d'équipe LoVE, Paris Nord.  
jui. 2023 • Formal Structures for Computation and Deduction (conférence internationale), Roma (Italie).



## Exposés (suite)

---

juin 2023 • Séminaire d'équipe **LDP**, Marseille.

### **Calcul à ressources extensionnel et sémantique catégorique**

mars 2022 • **Journées Nationales du GDR IM** (présentation de poster), Lille.

### **Introduction à la sémantique des jeux**

oct. 2021 • Séminaire des doctorant-e-s de l'**I2M**, Marseille.

### **Injectivité positionnelle des stratégies innocentes**

juin 2022 • Séminaire d'équipe **PPS**, Paris.

nov. 2021 • **Journées du GT Scalp** (conférence nationale), Fontainebleau.

sep. 2021 • Séminaire d'équipe **LDP**, Marseille.

jui. 2021 • **Formal Structures for Computation and Deduction** (conférence internationale), Buenos Aires (Argentine) (en ligne).

mai 2021 • Séminaire d'équipe **PLUME**, Lyon (en ligne).

## Bibliographie

---

- 1 P. CLAIRAMBAULT, « Causal Investigations in Interactive Semantics, » Habilitation à diriger des recherches, Aix-Marseille Université, 2024.  adresse : <https://hal.science/tel-04523273>.
- 2 T. TSUKADA et C. L. ONG, « Plays as Resource Terms via Non-idempotent Intersection Types, » in *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science, LICS '16, New York, NY, USA, July 5-8, 2016*, M. GROHE, E. KOSKINEN et N. SHANKAR, éd., ACM, 2016, p. 237-246.  DOI : [10.1145/2933575.2934553](https://doi.org/10.1145/2933575.2934553).
- 3 T. EHRHARD et L. REGNIER, « Uniformity and the Taylor expansion of ordinary lambda-terms, » *Theoretical Computer Science*, t. 403, n° 2-3, p. 347-372, 2008.  DOI : [10.1016/j.tcs.2008.06.001](https://doi.org/10.1016/j.tcs.2008.06.001).
- 4 R. BLUTE, J. R. B. COCKETT et R. A. G. SEELY, « Differential categories, » *Mathematical Structures in Computer Science*, t. 16, p. 1049-1083, 2006.  adresse : <https://api.semanticscholar.org/CorpusID:11818257>.
- 5 P. MELLIÈS, « Asynchronous games 2 : The true concurrency of innocence, » *Theor. Comput. Sci.*, t. 358, n° 2-3, p. 200-228, 2006.  DOI : [10.1016/j.tcs.2006.01.016](https://doi.org/10.1016/j.tcs.2006.01.016).
- 6 A. D. KER, H. NICKAU et C. L. ONG, « Innocent game models of untyped lambda-calculus, » *Theor. Comput. Sci.*, t. 272, n° 1-2, p. 247-292, 2002.  DOI : [10.1016/S0304-3975\(00\)00353-4](https://doi.org/10.1016/S0304-3975(00)00353-4).
- 7 S. ABRAMSKY, R. JAGADEESAN et P. MALACARIA, « Full Abstraction for PCF, » *Inf. Comput.*, t. 163, n° 2, p. 409-470, 2000.  DOI : [10.1006/inco.2000.2930](https://doi.org/10.1006/inco.2000.2930).
- 8 J. M. E. HYLAND et C. L. ONG, « On Full Abstraction for PCF : I, II, and III, » *Inf. Comput.*, t. 163, n° 2, p. 285-408, 2000.  DOI : [10.1006/inco.2000.2917](https://doi.org/10.1006/inco.2000.2917).
- 9 H. NICKAU, « Hereditarily Sequential Functionals, » in *Logical Foundations of Computer Science, Third International Symposium, LFCS'94, St. Petersburg, Russia, July 11-14, 1994, Proceedings*, A. NERODE et Y. V. MATIYASEVICH, éd., sér. Lecture Notes in Computer Science, t. 813, Springer, 1994, p. 253-264.  DOI : [10.1007/3-540-58140-5\\_25](https://doi.org/10.1007/3-540-58140-5_25).
- 10 D. S. SCOTT, « A Type-Theoretical Alternative to ISWIM, CUCH, OWHY, » *Theoretical Computer Science*, t. 121, n° 1&2, p. 411-440, 1993.  DOI : [10.1016/0304-3975\(93\)90095-B](https://doi.org/10.1016/0304-3975(93)90095-B).

- 11 J.-Y. GIRARD, « Normal functors, power series and  $\lambda$ -calculus, » *Ann. Pure Appl. Log.*, t. 37, p. 129-177, 1988.  
🔗 adresse : <https://api.semanticscholar.org/CorpusID:5877007>.
- 12 H. P. BARENDREGT, *The lambda calculus - its syntax and semantics* (Studies in logic and the foundations of mathematics). North-Holland, 1984, t. 103, ISBN : 978-0-444-86748-3.
- 13 G. D. PLOTKIN, « LCF Considered as a Programming Language, » *Theoretical Computer Science*, t. 5, n° 3, p. 223-255, 1977. 🔗 DOI : [10.1016/0304-3975\(77\)90044-5](https://doi.org/10.1016/0304-3975(77)90044-5).
- 14 A. CHURCH, « A Formulation of the Simple Theory of Types, » *The Journal of Symbolic Logic*, t. 5, n° 2, p. 56-68, 1940, ISSN : 00224812. 🔗 adresse : <http://www.jstor.org/stable/2266170> (visité le 29/07/2025).