Projet de Thèse (cotutelle avec l'université Roma Tre)

Directeurs de thèse:
S. Guerrini (Prof., lipn), T. Seiller (cr cnrs, lipn), L. Tortora de Falco (Assoc. Prof., Roma 3)

Organisation:
Il est prévu que le candidat passe deux semestres à l'université Roma Tre. A cela s'ajoutera des périodes de travail avec l'encadrant italien lors de séjours en France prévus durant la période (prochain séjour: visite de six semaines au printemps 2023). Les premiers mois en France seront également l'occasion pour le candidat d'échanger avec B. Eng, doctorant encadré par D. Mazza et T. Seiller, sur le modèle de la syntaxe transcendantale que ce dernier a formalisé durant sa thèse et qui sert de point de départ au présent projet.

Éléments de contexte:
Des projets collaboratifs et des cotutelles ont été montés par le passé entre l'Université Roma Tre et l'équipe LoVe du LIPN, mais ces activités ont cessé les dernières années. Cette co-tutelle permettra de redynamiser les liens entre ces deux partenaires. Cette collaboration s'inscrira également dans le cadre du GDRI franco-italien Linear Logic.

Le projet repose sur les expertises complémentaires des co-encadrants: réseaux de preuves, géométrie de l'interaction, et syntaxe transcendantale et logiques linéaires pour la complexité (T. Seiller). Le candidat ayant fait une partie de ses études à Roma Tre, notamment un stage encadré par L. Tortora de Falco, il retrouvera rapidement ses marques dans le pays et dans les échanges scientifiques.

Projet Scientifique.
Titre: Leçons de la syntaxe transcendantale: nouveaux réseaux et complexité.

Ce projet de thèse trouve ses origines dans la correspondance de Curry-Howard entre preuves mathématiques et programmes informatiques, et plus particulièrement la logique linéaire introduite par Jean-Yves Girard en 1987 [12]. La logique linéaire a eu un impact considérable sur la recherche en logique et en théorie des langages de programmation. Au-delà de ces échanges fructueux, la logique linéaire a eu un impact sur de nombreux domaines au-delà de la correspondance de Curry-Howard, notamment en philosophie (par exemple l'ANR GoA en 2021 et 2025) et en linguistique (par exemple via l'ANR LOCI entre 2012 et 2017).

En logique linéaire l'implication traditionnelle $A \Rightarrow B$, où A, B sont des formules, est remplacée par $!A \multimap B$, où ! est un connecteur *exponentiel* permettant la duplication de la formule A et $\multimap$ est une implication "linéaire". C'est au travers de la correspondance entre preuves et programmes que cette décomposition prends tout son sens (et c'est par ailleurs là qu'elle trouve son origine): utiliser la logique linéaire comme système de types pour les programmes permet de traduire et contrôler l'utilisation des ressources. On peut alors définir des variantes de la logique linéaire caractérisant des classes de complexité [13, 17, 20]. Ces résultats théoriques ont des applications pratiques, par exemple au travers de la conception de systèmes de types permettent d'assurer des bornes polynomiales sur le temps d'execution des programmes typables [1, 4, 5, 6, 8, 11]. Par ailleurs ces caractérisations pourraient ouvrir de nouvelles voies pour obtenir des bornes inférieures en complexité [23], l'un des problèmes les plus difficiles et important en informatique théorique à l'heure actuelle.

D'un point de vue purement logique, la logique linéaire vient avec deux systèmes de représentation des preuves, dont les *réseaux de preuves* qui sont des graphes satisfaisant une propriété particulière nommée *critère de correction*. Les réseaux de preuves ont eu un impact très important dans la communauté travaillant sur la correspondance de Curry-Howard en s'imposant rapidement comme le formalisme privilégié lors des discussions scientifiques et pour la diffusion des idées, et ce malgré quelques imperfections du formalisme pour la représentation des connecteurs *exponentiels* et des *quantificateurs*.

Après avoir introduit la logique linéaire et les réseaux de preuve, J.-Y. Girard a développé le programme de recherche de *géométrie de l'interaction*, ayant mené récemment au modèle de *syntaxe transcendantale*, pour reconstruire la logique à partir de modèles mathématiques des programmes. Au lieu de contraindre

les programmes par la logique (via les systèmes de types), on cherche à définir la logique induite par les interactions entre programme – celle-ci devient donc le langage adapté pour décrire ces derniers [18]. Les formules/types sont dans ce cas définis via une notion de *tests*: un programme est d'un type A donné s'il passe l'ensemble de tests correspondant T_A.

Récemment, T. Seiller et B. Eng (doctorant en troisième année) ont formalisé et étendu le modèle de syntaxe transcendantale esquissé par J.-Y. Girard [14, 15, 16]. Ils ont en particulier montré comment le critère de correction des réseaux de preuve se retrouve dans le modèle pour une restriction de la logique linéaire; celui-ci se traduit par des ensembles de tests définissant les types usuels de la logique linéaire [10]. Cependant, cette correspondance ne s'étend pas naturellement à l'ensemble de la logique linéaire (connecteurs exponentiels, quantificateurs). Ce projet de thèse prends son origine dans ce décalage entre les réseaux de preuves (avec leurs imperfections concernant les connecteurs exponentiels et les quantificateurs) et le modèle de syntaxe transcendantale qui propose une reconstruction de la logique en termes de tests. Il s'agira donc de développer une nouvelle notion de réseaux de preuves plus satisfaisante en traduisant les tests issus de la syntaxe transcendantale, et d'étudier des caractérisations de classes de complexité dans ce cadre.

On distinguera pour cela trois objectifs:

1. **(Réseaux pour les exponentielles et la quantification du second ordre.)** Il s'agira d'étudier les constructions des connecteurs exponentielles et des quantificateurs du second ordre en syntaxe transcendantale pour en extraire une notion de réseau de preuve correspondante. Ces connecteurs sont définis en terme de tests, comme expliqué ci-dessus, et ont été étudiés par B. Eng et T. Seiller. Ce premier objectif consistera donc à comprendre si l'on peut proposer un nouveau critère de correction (sans doute associé à une notion légèrement différente de réseau de preuve) qui traduirait ces tests. Les expertises complémentaires de T. Seiller sur la syntaxe transcendantale et de L. Tortora de Falco sur les réseaux de preuve seront ici essentielles.
2. **(Complexité.)** Il s'agira d'obtenir des caractérisations de classes de complexité dans la syntaxe transcendantale. On se basera sur plusieurs travaux antérieurs: les logique linéaires pour la complexité [7, 21, 22], des caractérisations de complexité dans les modèles de géométrie de l'interaction [2, 3, 24, 25], ainsi que des caractérisations de classes de complexité utilisant des pavages [19] – ces derniers étant naturellement représentés dans la syntaxe transcendantale [10]. Enfin, on établira ces caractérisations en les définissant à l'aide de tests (donc comme des types) afin de pouvoir les interpréter au niveau de la logique considérée, et éventuellement les exprimer dans la nouvelle syntaxe de réseaux de preuve. La bonne réalisation de cet objectif reposera sur les expertises conjointes des encadrants.
3. **(Quantificateurs du premier ordre.)** En parallèle de ces travaux se basant sur des résultats bien établis, on formalisera également l'extension du modèle de syntaxe transcendantale à la logique du premier ordre, esquissée par J.-Y. Girard [15]. Cela permettra dans un premier temps d'en tirer une syntaxe de réseaux pour le premier ordre (une extension du point 1), mais également un nouveau calcul des séquents. Le calcul des séquents est une syntaxe standard en logique mathématique. Si le modèle de syntaxe transcendantale pour les connecteurs exponentiels et les quantificateurs du second ordre n'induit pas un calcul des séquents modifiés, il est attendu que le traitement du premier ordre ait un impact sur celui-ci même dans le cas de la logique classique. Dans un second temps, on utilisera cette extension pour obtenir de nouvelles caractérisations de classes de complexité (une extension du point 2) inspirées des caractérisations connues dans le domaine de la complexité descriptive et fondées sur la logique du premier ordre [9].

References

[1] A. Asperti. Light affine logic. In *Proceedings. Thirteenth Annual IEEE Symposium on Logic in Computer Science (Cat. No. 98CB36226)*, pages 300–308. IEEE, 1998.

[2] C. Aubert, M. Bagnol, P. Pistone, and T. Seiller. Logic programming and logarithmic space. In J. Garrigue, editor, *Programming Languages and Systems - 12th Asian Symposium, APLAS 2014, Singapore, November 17-19, 2014, Proceedings*, volume 8858 of *Lecture Notes in Computer Science*, pages 39–57. Springer, 2014.

[3] C. Aubert, M. Bagnol, and T. Seiller. *Foundations of Software Science and Computation Structures: 19th International Conference, FOSSACS 2016, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2016, Eindhoven, The Netherlands, April 2-8, 2016, Proceedings*, chapter Unary Resolution: Characterizing Ptime, pages 373–389. Springer Berlin Heidelberg, 2016.

[4] P. Baillot. Checking polynomial time complexity with types. In *Foundations of Information Technology in the Era of Network and Mobile Computing*, pages 370–382. Springer, 2002.

[5] P. Baillot, M. Gaboardi, and V. Mogbil. A polytime functional language from light linear logic. In *European Symposium on Programming*, pages 104–124. Springer, 2010.

[6] P. Baillot and A. Ghyselen. Combining linear logic and size types for implicit complexity. *Theoretical Computer Science*, 813:70–99, 2020.

[7] P. Baillot and D. Mazza. Linear logic by levels and bounded time complexity. *Theoretical Computer Science*, 411(2):470–503, 2010.

[8] G. Curzi and L. Roversi. Probabilistic soft type assignment. *arXiv preprint arXiv:2007.01733*, 2020.

[9] A. Durand, N. D. Jones, J. A. Makowsky, and M. More. Fifty years of the spectrum problem: survey and new results. *The Bulletin of Symbolic Logic*, 18(4):505–553, 2012.

[10] B. Eng and T. Seiller. Multiplicative linear logic from logic programs and tilings. hal-02895111, 2021.

[11] M. Gaboardi and S. R. Della Rocca. A soft type assignment system for λ-calculus. In *International Workshop on Computer Science Logic*, pages 253–267. Springer, 2007.

[12] J.-Y. Girard. Linear logic. *Theoretical Computer Science*, 50(1):1–101, 1987.

[13] J.-Y. Girard. Light linear logic. In *Selected Papers from the International Workshop on Logical and Computational Complexity*, LCC '94, pages 145–176, London, UK, UK, 1995. Springer-Verlag.

[14] J.-Y. Girard. Transcendental syntax ii: non deterministic case. *Logical Methods in Computer Science*, 2016.

[15] J.-Y. Girard. Transcendental syntax iii: equality. *Logical Methods in Computer Science*, 2016.

[16] J.-Y. Girard. Transcendental syntax i: deterministic case. *Mathematical Structures in Computer Science*, 27(5):827–849, 2017.

[17] J.-Y. Girard, A. Scedrov, and P. J. Scott. Bounded linear logic: a modular approach to polynomial-time computability. *Theor. Comput. Sci.*, 97(1):1–66, Apr. 1992.

[18] J.-B. Joinet and T. Seiller. From abstraction and indiscernibility to classification and types: revisiting hermann weyl's theory of ideal elements. *Kagaku tetsugaku*, 53(2):65–93, 2021.

[19] N. Jonoska and G. L. McColm. Complexity classes for self-assembling flexible tiles. *Theor. Comput. Sci.*, 410(4–5):332–346, Feb. 2009.

[20] Y. Lafont. Soft linear logic and polynomial time. *Theor. Comput. Sci.*, 318(1-2):163–180, June 2004.

[21] D. Mazza. Linear logic and polynomial time. *Mathematical Structures in Computer Science*, 16(6):947–988, 2006.

[22] D. Mazza. Simple parsimonious types and logarithmic space. In *Proceedings of CSL*, pages 24–40, 2015.

[23] T. Seiller. Towards a complexity-through-realizability theory. *submitted*, 2015.

[24] T. Seiller. Interaction graphs: Nondeterministic automata. *ACM Transactions in Computational Logic*, 19(3), 2018.

[25] T. Seiller. Probabilistic complexity classes through semantics. Submitted, 2020.