

Symétrie et interaction : les jeux en logique focalisée

Anne-Laure SCHNEIDER

27 octobre 2009

Résumé

Recherche de preuves et normalisation de preuves, qu'on tend souvent à opposer, se rejoignent en réalité dans un cadre commun, les jeux, qui permettent d'un côté d'améliorer l'efficacité des prouveurs en utilisant de nouvelles structures qui réduisent l'espace de recherche des preuves, et de l'autre de donner une sémantique convaincante à des phénomènes logiques d'une grande diversité. C'est ce que je chercherai à justifier dans ma thèse, en utilisant la richesse des interactions dans les jeux et des théories adaptées comme la logique focalisée, la ludique et la géométrie de l'interaction, pour créer un type de jeu efficace à la fois dans sa recherche de stratégies gagnantes, et dans le cadre du model-checking, pour déterminer la satisfaisabilité ou la réfutabilité d'une proposition.

La recherche en théorie de la démonstration s'articule aujourd'hui autour de deux grands axes : la recherche de preuves (ou la programmation logique) et la normalisation de preuves (ou programmation fonctionnelle). Le premier domaine consiste à chercher des démonstrations sans coupures d'une formule donnée, et le principal objectif est de rendre les prouveurs automatiques plus rapides et plus efficaces. Le second domaine se concentre sur la preuve en tant qu'objet d'étude en elle-même, il s'agit d'obtenir une preuve sans coupures à partir d'une preuve quelconque (ce qui correspond à l'étude de la représentation des termes, et des substitutions explicites). Mais plus généralement, il s'agit d'obtenir un représentant d'une classe de preuves, leur forme normale. La question qui se pose alors est celle de la signification de ce regroupement des preuves en classes : quelles preuves devrait-on identifier, quelles preuves devrait-on distinguer ? Différentes sémantiques sont évidemment possibles, et selon les cas on privilégiera l'aspect opérationnel (si on n'applique pas rigoureusement les mêmes règles dans le même ordre, deux preuves sont différentes¹), ou l'aspect extensionnel (deux preuves sont identiques si elles prouvent la même chose), beaucoup de sémantiques intermédiaires étant possibles.

Ces deux aspects sont cependant complémentaires, car choisir des représentants adaptés pour les preuves et identifier des classes entières de preuves permettent de restreindre l'espace de recherche pour les prouveurs automatiques. Mon objectif est d'améliorer l'efficacité des prouveurs, de bâtir de nouvelles structures pour la recherche de preuves, et de proposer une sémantique adaptée, en me plaçant dans le cadre des jeux. Je chercherai à unifier différentes théories logiques

1. Par exemple si l'on veut prouver $A \wedge B$, commencer par prouver A ou par prouver B donnera deux preuves différentes

comme la logique focalisée, les jeux neutres, la géométrie de l'interaction et la ludique, dans un même jeu qui puisse s'appliquer dans la logique linéaire tout entière. Par ailleurs, je souhaite fortifier les liens entre recherche de preuves et model-checking, à travers des jeux qui permettent facilement les deux : à partir d'une recherche de preuves générale, on pourrait calculer très facilement différentes évaluations d'atomes qui rendraient la proposition vraie ou fausse.

1 Pourquoi utiliser des jeux ?

L'étude de la logique peut se faire à travers différents biais, différents formalismes. De nombreuses équivalences ont été prouvées : grâce à l'équivalence de Curry-Howard on peut considérer indifféremment des preuves et des programmes ; le calcul des séquents, la déduction naturelle ou le calcul des structures sont trois formalismes des mêmes phénomènes ; dans sa thèse, Hugo Herbelin ([Her95]) montre l'équivalence entre le calcul des séquents et le lambda-calcul, ainsi que l'équivalence entre une stratégie gagnante dans un jeu de Lorenzen (jeu de dialogue à base de questions et réponses) et une preuve du calcul des séquents ; de nombreux théorèmes d'équivalence entre certains types de jeux et certains types de logique ont été prouvés (à commencer par le travail fondateur de Jaakko Hintikka dans [Hin73], qui a introduit les jeux de dialogue dans la logique), ainsi que les liens entre les jeux et les programmes (comme la classe des jeux simples qui représente un modèle extensionnel de PCF dans [HOn00]). Tout ceci permet d'envisager un même problème sous des angles très différents. Le formalisme des jeux présente l'avantage d'être à la fois très riche (le principe de base est simplement celui d'une interaction entre joueurs, tout le reste peut être redéfini) et très intuitif. Par ailleurs le jeu donne des intentions aux joueurs (en général l'un essaie de prouver une formule et l'autre de la réfuter, et une stratégie gagnante pour l'un des joueurs constitue une preuve respectivement de la formule ou de sa négation), ce qui est très proche de la réalité d'une démonstration et permet de donner plus de sens à la notion de preuve, ou plutôt d'identité des preuves. Les jeux sont donc un cadre idéal pour unifier des principes et des idées très diverses, tout en gardant une cohérence interne.

2 Simplifier la recherche de preuves : logique focalisée et jeux à approche neutre

Pour rendre un processus plus rapide, différentes pistes sont possibles. L'une d'entre elles est d'essayer d'en éliminer tout indéterminisme : c'est le principe de la logique focalisée développée par Jean-Marc Andreoli dans [And92]. Ses travaux se fondent sur l'idée que certaines règles logiques sont réversibles : par conséquent on peut les appliquer immédiatement sans risquer de perdre de l'information sur la proposition qu'on cherche à prouver. Cela constitue une phase dite asynchrone, où l'on applique en réalité un ensemble de règles, indépendamment de leur ordre. Ensuite on applique une ou plusieurs règles non inversibles, qui impliquent un choix, c'est la phase synchrone. Un prouveur automatique n'a donc pas à se préoccuper de l'ordre des règles inversibles qu'il applique, et identifie des preuves qui ne diffèrent que par une inversion de ces règles. C'est donc un atout à la fois en ce qui concerne la vitesse de la recherche de preuves, et le sens que l'on peut donner aux différentes preuves trouvées (il s'agit d'une sémantique des preuves intermédiaire entre l'extensionnel et l'opérationnel). À l'heure actuelle il n'existe pas de jeux appropriés pour modéliser la logique focalisée dans son ensemble.

Je compte poursuivre les travaux commencé par Dale Miller et Alexis Saurin ([MS05]) et poursuivis par Olivier Deland ([DM08] et [DMS09]) qui ont établi un modèle de jeu neutre (ou symétrique) pour le fragment multiplicatif et additif de la logique linéaire focalisée (MALL). Je

souhaite étendre ce modèle en lui ajoutant les quantificateurs logiques et les exponentielles, pour pouvoir modéliser la logique linéaire focalisée dans son ensemble.

Par ailleurs, cette approche neutre des jeux en est encore à ses balbutiements. L'idée est que dans un jeu le Joueur et son Opposant ont des rôles parfaitement symétriques. On peut donc écrire le jeu sous une forme neutre, qu'il suffit de traduire dans une version positive pour obtenir la vision du Joueur, et dans une version négative pour obtenir celle de l'Opposant. Cette approche, qui a le mérite de considérer un unique jeu pour modéliser la recherche de preuve d'une formule F et de sa négation $\neg F$, n'est cependant pas totalement satisfaisante. Pour rechercher des stratégies gagnantes, on est pour l'instant obligés de commencer par traduire le jeu, et donc de revenir au point de départ. L'approche neutre donne une nouvelle structure et une nouvelle compréhension du jeu, mais on n'a pas encore d'approche neutre de la recherche de stratégies gagnantes. L'un de mes objectifs est donc de chercher à développer cette approche pour déterminer si une stratégie pour un joueur est gagnante ou non, ce qui permettrait, en un seul calcul, d'obtenir une preuve de F ou une preuve de sa négation. Je compte le faire dans un premier temps pour des jeux dans ALL ou MLL, pour lesquels il n'y a que deux possibilités : ou bien le Joueur gagne, ou bien l'Opposant gagne, avant de voir s'il est possible de le généraliser à des cas plus complexes, où il est possible que ni l'un ni l'autre ne gagnent.

3 L'interactivité des jeux en lien avec le model-checking

J'ai déjà obtenu, dans le cadre simplifié de la logique classique focalisée positive (LKF+), des résultats encourageant sur un jeu qui permet à la fois une approche neutre à la Miller-Delande de la recherche de preuves, mais aussi des calculs très simples de satisfaisabilité et de réfutabilité (rapport de recherches disponible sur ma page Internet). Je compte donc poursuivre ce travail en l'étendant à la logique linéaire focalisée, ce qui pose le problème plus complexe des interactions entre connecteurs additifs et multiplicatifs.

Les avantages d'une approche neutre ne doivent pas en effet nous faire oublier tout l'intérêt sémantique des jeux, qui procurent une interactivité entre deux joueurs et sont en cela révélateurs d'une dynamique dans la recherche de preuve. La géométrie de l'interaction introduite par Girard rend compte de la richesse de ces interactions². La ludique ([Gir01],[Cur02] et [Gir03]) est le développement de ces recherches.

Dans celles-ci, Girard considère des pseudo-preuves, ou dérivations, et non pas seulement des preuves complètes. Dans ce cadre, une dérivation peut s'achever avec une règle daimon à la place de la règle d'axiome. Cela signifie simplement que l'on admet un atome ou une formule, plutôt que de le prouver. Les dérivations sont donc en quelque sorte des preuves qui attendent d'être complétées par une évaluation appropriée des atomes, si cela est possible. On peut donc à partir de là obtenir des résultats de satisfaisabilité ou de réfutabilité, comme j'ai pu le faire dans les jeux que j'ai introduits. Je travaillerai à expliquer les liens entre ces jeux et la ludique, afin de pouvoir profiter des avancées effectuées dans les deux domaines simultanément. Je m'inspirerai également des travaux de Curien et Herbelin ([CH05]) sur les arbres de Böhm et les jeux, ainsi que des machines abstraites auxquelles ils ont abouti.

4 Ajouter des quantificateurs

Dès que l'on souhaite utiliser des quantificateurs se pose la question de l'unification. En effet en recherche de preuves, lorsque l'on rencontre un quantificateur existentiel, on lui attribue une

2. Au niveau logique le procédé d'élimination des coupures correspond à la dynamique de recherche de preuves, ou de stratégies gagnantes dans un jeu, comme l'a démontré Saurin dans [Sau08]

variable, et ensuite, selon les propriétés que l'on aura obtenu sur cette variable, on pourra lui donner une valeur. C'est ce processus qui est appelé unification, puisqu'il unifie tout ce qu'on sait sur un élément. Mais il est difficile d'effectuer une unification si l'élément est par exemple variable d'une fonction, etc. Il existe déjà des unifications de plus grand ordre ([Dow01]), mais tout cela est difficile à rajouter aux systèmes déjà existants, et il serait intéressant de pouvoir l'inclure dans mon modèle de jeu.

5 Implémentation

Selon la forme qu'auront mes résultats, je souhaite en faire une implémentation, ou bien en intégrant dans le prouveur de l'équipe Parsifal au LIX, Bedwyr ([BGMNT07]), ou bien en remplaçant Bedwyr par un outil plus performant, car en particulier en ce qui concerne la gestion des quantificateurs, Bedwyr n'est pour l'instant pas adapté, et ne peut pas prouver tout ce qu'on souhaiterait qu'il prouve³. Il peut alors être plus simple d'implémenter une nouvelle théorie de l'unification dans un nouveau prouveur.

L'équipe Parsifal a des collaborateurs à l'Université de Minneapolis et ailleurs dans le monde qui ont déjà participé à l'élaboration de Bedwyr et d'autres prototypes, nous pouvons donc compter sur leur aide pour réaliser une implémentation plus rapidement.

6 Références

[And92] Jean-Marc Andreoli. Logic programming with focussing proofs in linear logic. *Journal of Logic and Computation*, 2(3) : 291-341, 1992.

[BGMNT07] David Baelde, Andrew Gacek, Dale Miller, Gopalan Nadathur, et Alwen Tiu. The Bedwyr system for model checking over syntactic expressions. *CADE 2007 : 21th Conference on Automated Deduction*, Frank Pfenning, editor, LNAI 4603, pages 391-397. Springer, 2007.

[Cur02] Pierre-Louis Curien. Introduction to ludics. *Manuscrit*, 2002.

[CH05] Pierre-Louis Curien et Hugo Herbelin. Abstract machine for dialogue games. A paraître dans *Interactive models of computation and program behaviour* dans la collection *Panoramas et Synthèses*, SMF.

[DM08] Olivier Delande et Dale Miller. A neutral approach to proof and refutation in MALL. *Proceedings of LICS 2008*, édité par F. Pfenning, Pittsburgh, June 2008, pp. 498-508.

[DMS09] Olivier Delande, Dale Miller et Alexis Saurin. Proof and refutation in MALL as a game. *Accepté par les Annals of Pure and Applied Logic*. Brouillon du 20 May 2009.

[Dow01] Gilles Dowek. *Handbook of automated reasoning*, chapitre 16. Elsevier Science Publishers, 2001.

[Gir01] Jean-Yves Girard. Locus Solum : From the rules of logic to the logic of rules. *Mathematical Structures in Computer Science*, 11(3) : 301-507, 2001.

[Gir03] Jean-Yves Girard. From foundation to Ludics. *The Bulletin of Symbolic Logic*, 9(2) : 131-168, juin 2003.

[Her95] Hugo Herbelin. Séquents qu'on calcule. *Thèse de doctorat*, 1995.

[Hin73] Jaakko Hintikka. *Logic, Language-Games and Information : Kantian Themes in the Philosophy of Logic*. Clarendon Press, Oxford 1973.

3. Par exemple il ne peut pas prouver l'assertion $\exists x (x = a \Rightarrow \text{false})$, car il doit alors gérer une infinité de possibilités pour la variable x , et son degré d'unification ne lui permet pas de le faire. Le problème vient de ce que les contraintes sont gérées de façon « multiplicative », c'est-à-dire qu'elles n'interagissent pas assez entre elles.

[HOn00] J.M.E.Hyland et C.-H.L.Ong. On Full Abstraction for PCF : I, II and III. *Information and Computation*, 163, 285-408, 2000.

[MS05] Dale Miller et Alexis Saurin. A game semantics for proof search : Preliminary results. Présenté à MFPS 2005 : *Mathematical Foundations of Programming Semantics*, Birmingham, UK (18-21 Mai 2005).

[Sau08] Alexis Saurin. *Towards Ludics Programming : Interactive Proof Search*. ICLP 2008, *Lecture Notes in Computer Science* 5366, Springer-Verlag, pp. 253-268, 2008.