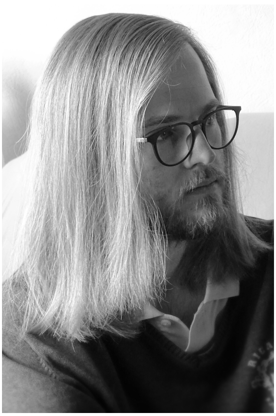
Automates Distribués et Logique

Fabian Reiter [1]

Fabian Reiter a soutenu sa thèse [2] en décembre 2017 à l'université Paris-Diderot, thèse préparée à l'Institut de recherche en informatique fondamentale (IRIF), sous la direction d'Olivier Carton. Il effectue actuellement un stage postdoctoral au sein de l'équipe de Javier Esparza à Munich.



De manière très synthétique, on peut dire que ma recherche vise à développer une forme de *complexité descriptive* pour le *calcul distribué*.

Qu'est-ce que cela veut dire ?

Dans le calcul séquentiel classique, la complexité descriptive est un domaine bien établi qui compare les puissances expressives de certaines classes de *formules* logiques avec celles de certaines classes d'*algorithmes*, généralement représentés comme des machines abstraites. Le « Graal », pour ainsi dire, est d'établir des équivalences de la forme :

(1) « *La classe de formules Φ a la même puissance que la classe d'algorithmes $\mathscr{A}$.* »

1. fabian.reiter@gmail.com
2. « Distributed Automata and Logic », https://hal.archives-ouvertes.fr/tel-01827435.

Le domaine est né dans les années 1970 avec le théorème de Fagin [Fag74], qui affirme (de manière légèrement simplifiée) qu'une propriété de graphe (telle que « être hamiltonien ») peut être définie par une formule de la logique existentielle du second ordre si et seulement si elle peut être reconnue par une machine de Turing non déterministe en temps polynomial. Ce théorème fournit ainsi une caractérisation logique – et de ce fait indépendante de toute notion de machine – de la classe de complexité NP. Par la suite, de nombreuses autres classes connues, dont P, PSPACE, et EXPTIME, ont été caractérisées de façon similaire (voir par exemple les livres [GKL$^+$07, Imm99, Lib04]). Toutes ces caractérisations peuvent être représentées par une image comme celle de la figure 1. Dans la partie gauche, nous considérons un formalisme logique qui nous permet de décrire des propriétés de graphes (ou, plus généralement, des propriétés de structures relationnelles). À droite, nous avons une notion de machine abstraite qui nous permet de vérifier de telles propriétés sur des graphes encodés sous forme de chaînes binaires.

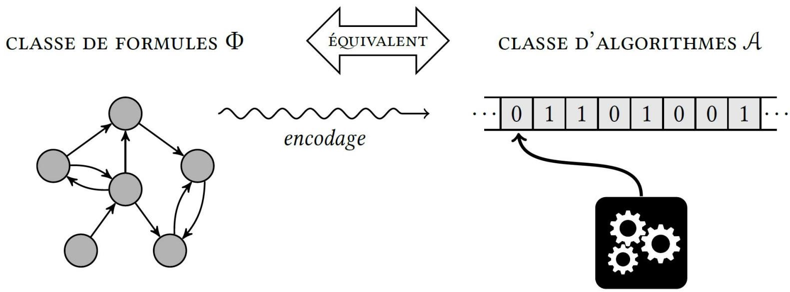


FIGURE 1. La complexité descriptive classique.

Le calcul distribué [Lyn96, Pel00], en revanche, concerne des réseaux composés de plusieurs processeurs interconnectés qui partagent un but commun. Les processeurs communiquent entre eux en transmettant des messages le long des liens du réseau pour résoudre collectivement un problème algorithmique. Dans beaucoup de cas, il s'agit d'un problème de graphe, où l'instance de problème considérée est précisément le graphe défini par le réseau lui-même. Tous les processeurs exécutent le même algorithme simultanément, et souvent ne font aucune hypothèse préalable sur la taille et la topologie du graphe. Quelques problèmes typiques qui peuvent être résolus par de tels *algorithmes distribués* sont la coloration de graphe, l'élection d'un leader, et les constructions d'arbre couvrant et d'ensemble indépendant maximal.

Bien qu'elle soit bien développée dans le cas classique, la théorie de la complexité descriptive n'a jusqu'à présent pas reçu beaucoup d'attention dans le contexte du calcul distribué. À ma connaissance, le premier pas dans cette direction a été fait par Hella et al. dans [HJK$^+$12, HJK$^+$15], où ils ont montré que la *logique modale* de base évaluée sur des graphes finis a la même puissance expressive qu'une classe particulière d'*automates distribués* fonctionnant en temps constant. Ces automates constituent un modèle faible de calcul distribué, où tous les nœuds exécutent de manière synchrone la même machine à états finis et communiquent entre eux en diffusant leur état actuel à leurs voisins pour un nombre constant de tours de communication. Bien que ce modèle soit assez simple, le résultat est novateur dans le sens qu'il démontre comment, en principe, l'approche descriptive peut être transférée de la complexité classique (séquentielle) à la complexité distribuée. C'est peut-être le premier exemple d'une équivalence de la forme :

> (2) « *La classe de formules Φ a la même puissance que la classe d'algorithmes* distribués $\mathscr{A}$. »

Une telle équivalence peut être représentée par une image comme dans la figure 2. Le côté gauche est exactement le même que celui de la figure 1, c'est-à-dire que nous considérons à nouveau un formalisme logique qui nous permet de décrire des propriétés de graphe. Mais à droite, nous avons maintenant plusieurs machines abstraites qui communiquent afin de vérifier certaines propriétés du graphe sous-jacent au réseau de communication. Ces machines abstraites constituent notre représentation formelle d'algorithmes distribués. Notons que contrairement à la figure 1, le graphe n'a pas besoin d'être encodé, puisqu'il est donné implicitement par la topologie du réseau (qui pourrait même être infini).

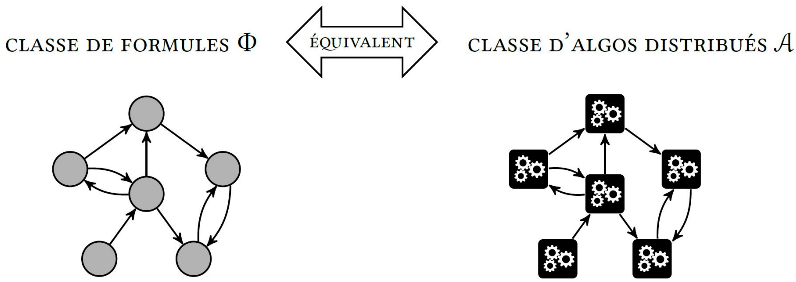


FIGURE 2. La complexité descriptive dans le contexte du calcul distribué.

Comme je l'ai mentionné plus haut, mon principal objectif jusqu'à présent a été de développer davantage la complexité distribuée descriptive. Cela signifie qu'une grande partie de mon activité de recherche a été consacrée à l'établissement d'équivalences de la forme (2).

Pourquoi est-ce intéressant ?

Tout d'abord, une complexité descriptive pour le calcul distribué offrirait les mêmes bénéfices que son homologue classique pour le calcul séquentiel :

(a) Si la classe d'algorithmes distribués $\mathscr{A}$ s'avère équivalente à la classe de formules Φ, alors cela témoigne du caractère naturel des deux classes. En effet, la définition de n'importe quel objet mathématique peut, en elle-même, sembler arbitraire. *Pourquoi les machines distribuées devraient-elles communiquer précisément de cette manière ? Pourquoi les formules logiques devraient-elles contenir précisément ces composantes ?* Mais si deux objets qui semblent très différents au premier abord se révèlent être des descriptions d'exactement la même chose, alors il est peu probable qu'il s'agisse d'une pure coïncidence.

(b) La mise en relation de deux domaines qui paraissent indépendants – ici, le calcul distribué et la logique – peut apporter de nouvelles perspectives dans les deux domaines. Certaines démonstrations pourraient être plus faciles à réaliser si l'on adopte le point de vue de l'un plutôt que celui de l'autre. En outre, certaines questions ouvertes dans un domaine pourraient déjà avoir des réponses connues dans l'autre. En particulier, le calcul distribué pourrait en bénéficier, étant donné que ce domaine a plus d'un siècle de moins que la logique formelle, et a donc eu moins de temps pour évoluer.

Par ailleurs, le calcul distribué apporte également une nouvelle dimension intéressante à la complexité descriptive elle-même :

(c) Les algorithmes distribués peuvent être évalués sur les mêmes entrées que les formules logiques, sans avoir besoin d'encoder ces entrées. Plus précisément, le réseau dans lequel un algorithme distribué est exécuté peut être considéré comme identique à la structure sur laquelle la valeur de vérité d'une formule correspondante est évaluée. Ceci contraste fortement avec la complexité descriptive classique, où les équivalences sont toujours exprimées relativement à un encodage (cf. les figures 1 et 2).

Mes contributions

Dans le cadre de ma thèse, j'ai obtenu des caractérisations logiques similaires à celle de Hella et al. pour deux classes d'automates distribués plus expressives.

[Rei15] La première classe étend les automates fonctionnant en temps constant avec une condition d'acceptation globale et la capacité d'alterner entre des modes de calcul non-déterministe et parallèle. J'ai montré que cette classe est équivalente à la logique monadique du second ordre sur les graphes. En me restreignant à des transitions non-déterministes ou déterministes, j'ai également obtenu deux variantes d'automates strictement plus faibles pour lesquelles le problème du vide est décidable.

[Rei17] La seconde classe adapte la notion standard d'algorithme asynchrone au cadre des automates distribués non-locaux. Les machines résultantes se sont avérées équivalentes à un petit fragment de la logique de point fixe, et plus précisément, à une variante restreinte du μ-calcul modal qui autorise les plus petits points fixes mais interdit les plus grands points fixes. Profitant du lien avec la logique, j'ai aussi montré que la puissance expressive de ces automates asynchrones est indépendante du fait que des messages puissent être perdus ou non.

En dehors des connexions entre le calcul distribué et la logique, j'ai aussi travaillé sur les deux sujets séparément.

[KR17] Du côté distribué, dans le cadre d'une collaboration avec Antti Kuusisto, nous avons étudié la décidabilité du problème du vide pour plusieurs classes d'automates non-locaux. Nous avons montré que le problème est indécidable en général, en simulant une machine de Turing par un automate distribué qui échange les rôles de l'espace et du temps. En revanche, le problème s'est révélé décidable en LOGSPACE pour une classe d'automates oublieux, où les nœuds voient les messages reçus de leurs voisins, mais ne se souviennent pas de leur propre état.

[Rei16] Du côté logique, j'ai élaboré de nouvelles preuves de séparation pour plusieurs hiérarchies d'alternance de quantificateurs basées sur la logique modale.

Quelles sont les perspectives ?

Bien que toutes les caractérisations logiques de calcul distribué mentionnées ci-dessus soient intéressantes du point de vue théorique, elles partagent toutes le même inconvénient majeur : un modèle de calcul distribué très faible qui se base sur des machines à états finis. Ainsi, le principal défi reste d'établir des caractérisations logiques de modèles plus complexes, suffisamment puissants pour exécuter le type d'algorithmes habituellement considérés dans le calcul distribué. Pour avoir un intérêt pratique, une telle caractérisation devrait être en termes de formules *finies*, exactement comme celle fournie par le théorème de Fagin pour les machines de Turing non déterministes à temps polynomial. Récemment, dans une collaboration avec Benedikt Bollig et Patricia Bouyer, nous avons commencé à caractériser des automates munis de registres qui peuvent stocker des identifiants uniques. À plus long terme, le but serait de trouver des caractérisations similaires pour des machines de Turing distribuées soumises à certaines contraintes de temps et d'espace.

Références

[Fag74] Ronald Fagin. Generalized first-order spectra and polynomial-time recognizable sets. In Richard M. Karp, editor, *Complexity of Computation*, volume 7 of *SIAM-AMS Proceedings*, pages 43–73, 1974.

[GKL+07] Erich Grädel, Phokion G. Kolaitis, Leonid Libkin, Maarten Marx, Joel Spencer, Moshe Y. Vardi, Yde Venema, and Scott Weinstein. *Finite Model Theory and Its Applications*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2007.

[HJK+12] Lauri Hella, Matti Järvisalo, Antti Kuusisto, Juhana Laurinharju, Tuomo Lempiäinen, Kerkko Luosto, Jukka Suomela, and Jonni Virtema. Weak models of distributed computing, with connections to modal logic. In Darek Kowalski and Alessandro Panconesi, editors, *ACM Symposium on Principles of Distributed Computing, PODC '12, Funchal, Madeira, Portugal, July 16-18, 2012*, pages 185–194. ACM, 2012.

[HJK+15] Lauri Hella, Matti Järvisalo, Antti Kuusisto, Juhana Laurinharju, Tuomo Lempiäinen, Kerkko Luosto, Jukka Suomela, and Jonni Virtema. Weak models of distributed computing, with connections to modal logic. *Distributed Computing*, 28(1) :31–53, 2015.

[Imm99] Neil Immerman. *Descriptive complexity*. Graduate texts in computer science. Springer, 1999.

[KR17] Antti Kuusisto and Fabian Reiter. Emptiness problems for distributed automata. In Patricia Bouyer, Andrea Orlandini, and Pierluigi San Pietro, editors, *Proceedings Eighth International Symposium on Games, Automata, Logics and Formal Verification, GandALF 2017, Roma, Italy, 20-22 September 2017.*, volume 256 of *EPTCS*, pages 210–222, 2017.

[Lib04] Leonid Libkin. *Elements of Finite Model Theory*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2004.

[Lyn96] Nancy A. Lynch. *Distributed Algorithms*. Morgan Kaufmann, 1996.

[Pel00] David Peleg. *Distributed Computing : A Locality-Sensitive Approach*, volume 5 of *SIAM Monographs on Discrete Mathematics and Applications*. Society for Industrial and Applied Mathematics (SIAM), 2000.

[Rei15] Fabian Reiter. Distributed graph automata. In *30th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2015, Kyoto, Japan, July 6-10, 2015*, pages 192–201. IEEE Computer Society, 2015.

[Rei16] Fabian Reiter. Alternating set quantifiers in modal logic. *CoRR*, abs/1602.08971, 2016.

[Rei17] Fabian Reiter. Asynchronous distributed automata : A characterization of the modal mu-fragment. In Ioannis Chatzigiannakis, Piotr Indyk, Fabian Kuhn, and Anca Muscholl, editors, *44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, July 10-14, 2017, Warsaw, Poland*, volume 80 of *LIPIcs*, pages 100 :1–100 :14. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017.