

Abstract State Machines

Thomas Seiller et Jean-Yves Moyen

Sujet

Les Abstract State Machines (ASM) ont été introduites par [Yuri Gurevich](#) au début des années 1990, dans l'intention d'obtenir un modèle de calcul satisfaisant une version étendue de la thèse de Church-Turing. En effet, là où la thèse de Church-Turing se contente d'énoncer que toute *fonction* calculable est calculable par une machine de Turing (ou un terme du λ -calcul), la thèse de Church-Turing étendue s'intéresse aux *algorithmes*. En termes d'ASM, celle-ci énonce que tout algorithme peut être simulé pas-à-pas par une ASM.

Le but du stage sera dans un premier temps d'étudier le modèle des ASM et le résultats de Gurevich prouvant la thèse de Church-Turing étendue pour les algorithmes séquentiels [2]. Si le temps le permet, et selon les intérêts du stagiaire, il sera également possible de travailler sur l'extension de ce résultat aux algorithmes parallèles [1] ou d'étudier une caractérisation, pouvant être caractérisée d'*implicite*, des algorithmes PTIME basée sur les ASM [4, 3]. Cette caractérisation diffère d'autres caractérisations (implicites) de PTIME en cela qu'elle est complète, autrement dit qu'elle caractérise tous les algorithmes polynomiaux.

Environnement de Travail

Ce stage sera co-encadré par [Thomas Seiller](#), postdoctorant à l'Université de Copenhague, et [Jean-Yves Moyen](#), maître de conférence à l'Université Paris 13 et actuellement détaché à l'Université de Copenhague. Le stage aura lieu dans le département d'informatique de l'Université de Copenhague ([DIKU](#)), au sein d'une équipe internationale travaillant sur divers aspects de la calculabilité et la complexité implicite.

References

- [1] Andreas Blass and Yuri Gurevich. Abstract state machines capture parallel algorithms. *ACM Trans. Comput. Log.*, 4(4):578–651, 2003.
- [2] Yuri Gurevich. Sequential abstract-state machines capture sequential algorithms. *ACM Trans. Comput. Log.*, 1(1):77–111, 2000.
- [3] Yoann Marquer. *Caractérisation impérative des algorithmes séquentiels en temps quelconque, primitif récursif ou polynomial*. PhD thesis, Université Paris Est, 2015.
- [4] Yoann Marquer and Pierre Valarcher. An algorithmic complete programming language for a large class of algorithms computable in polynomial time. *DICE 2015*. Extended Abstract.