Le problème de l'équivalence des automates à pile déterministes est décidable.

8 juillet 1998

1

Résumé: Depuis le milieu des années 60, les informaticiens théoriciens cherchaient à résoudre le "problème de l'équivalence des automates à pile déterministes". Il s'agissait de déterminer s'il existe (ou non) un algorithme résolvant le problème suivant: étant donnés deux automates (à mémoire pile et déterministes), reconnaissent-ils le même langage?
Il s'avère qu'un tel algorithme existe: cela a été démontré en Janvier 1997 par un chercheur de l'équipe "Langages Formels et Graphes" du LaBRI (LAboratoire Bordelais de Recherche en Informatique), à Bordeaux.

La décidabilité L'un des objectifs généraux que poursuivent les informaticiens théoriciens est de classifier les problèmes calculatoires selon qu'ils sont *décidables* ou, au contraire, *indécidables*. La notion formelle de problème décidable a été définie bien avant l'invention des ordinateurs, dans les années 30, par les spécialistes de logique mathématique. En termes modernes et informels, un problème est *décidable* si l'on peut le résoudre par un algorithme (un programme) qui serait exécuté sur un ordinateur dont le temps de calcul et l'espace-mémoire ne seraient pas limités à l'avance.

Les automates Les travaux de linguistique mathématique de Chomsky et Schützenberger (1963) ont abouti à diverses notions de *grammaire formelle*, en particulier la notion de grammaire *algébrique* (ou encore *sans-contexte* dans la terminologie anglo-saxonne). Les *automates à pile* ont la propriété de "correspondre aux grammaires algébriques" en ce sens qu'un langage L (i.e. un ensemble de mots sur un alphabet X) peut être décrit par une telle grammaire G si et seulement si le problème de l'appartenance d'un mot à L peut être résolu par un automate à pile A. On dit alors que L est le langage *reconnu* par l'automate A. (La notion de "pile" utilisée ici, correspond à une "structure de donnée ", c'est à dire à l'ensemble des opérations algébriques autorisées dans la mémoire de l'automate). Les travaux de Knuth (1965) ont mis en évidence la sous-classe des grammaires $LR(k)$ qui correspondent aux automates à pile *déterministes*. L'immense majorité des langages de programmation actuels sont spécifiés par des grammaires LR(k) et leurs compilateurs effectuent la phase d'analyse syntaxique au moyen d'un automate à pile .

En termes mathématiques: l'automate est une action à droite $\odot$ du monoïde $(X^*, \cdot, 1)$ sur la mémoire de A et , si l'on note a le contenu initial de la mémoire de A, L est l'ensemble des mots $u \in X^*$ tels que $a \odot u = 1$. La simulation d'une telle action à droite $\odot$ permet de résoudre en temps linéaire le problème de l'appartenance à L.

Le "problème de l'équivalence des automates à pile déterministes", posé en 1966 par Ginsburg et Greibach, est le suivant:
peut-on décider, étant donnés deux automates (à pile déterministes) A, B, si les langages qu'ils reconnaissent sont identiques?
(ce que nous abrégerons par $L(A) = L(B)$? ou encore $A \equiv B$?)

Résultats antérieurs De nombreux travaux antérieurs à Janvier 1997 avaient démontré la décidabilité de $L(A) = L(B)$? lorsque A, B vérifient des hypothèses particulières. Citons pour mémoire le cas des automates simples (A, B n'ont qu'un état et pas de transition spontanée) et le cas des automates à nombre fini de pics (A, B ont plusieurs états et des transitions spontanées, mais leur pile ne change qu'un nombre borné de fois de sens de variation).

Les méthodes utilisées se répartissaient en deux groupes principaux:

1. les méthodes de "simulation en parallèle" des deux automates A, B (initiées par Valiant, 1973). Elles reposent sur une analyse fine des similitudes que les deux automates doivent présenter, s'il est bien vrai qu'ils sont équivalents; on utilise ensuite ces similitudes pour démontrer que la simulation de A et B en parallèle peut être réalisée par un autre automate C, lui-même dans une classe d'automate "suffisamment régulière".

2. les méthodes de "transformations d'équations" (initiées par Hopcroft et Korenjac, 1966). Elles consistent à développer l'équation $A \equiv B$ en "équations résiduelles" obtenues en faisant agir des mots de longueur croissante sur A et B, puis à simplifier les équations obtenues dès qu'elles deviennent trop "grosses".

Une solution La solution complète trouvée par G. Sénizergues ([1]) consiste tout d'abord à généraliser le problème initial en examinant l'ensemble des "séries rationnelles déterministes" écrites sur un alphabet V (essentiellement l'alphabet de pile de A, B). L'ensemble de ces séries forme une structure algébrique nouvelle, nommée *espace déterministe*, qui présente des analogies avec la notion classique d'espace vectoriel (existence de bases dans le cas de type fini, possibilité de trianguler les sytèmes linéaires) et aussi avec la notion d' automate: l'espace E des "séries rationnelles déterministes" sur V est muni d'une action à droite $\odot$ qui prolonge celle des mots sur la mémoire des automates A, B. Ces espaces permettent en particulier une formalisation rigoureuse de la notion d'indépendance linéaire de langages, introduite par Meitus (1989).

Il exhibe ensuite un "système axiomatique " $\mathcal{D}$ qui engendre certaines identités "$C \equiv D$" entre de telles séries. Une telle identité est valide lorsque

$$\{u \in X^*, C \odot u = 1\} = \{u \in X^*, D \odot u = 1\}.$$

Le type de système axiomatique utilisé fait directement suite aux systèmes étudiés par Harrison-Havel-Yehudai (1979) et généralisés par Courcelle (1983). Le système $\mathcal{D}$ n'engendre que des identités valides et en outre il est *complet*, c'est à dire qu'il engendre *toutes* les identités valides. Etablir cette propriété de complétude: c'est là le point crucial et aussi la difficulté technique majeure de cette solution.

Si en fait $A \not\equiv B$, on peut trouver un mot u tel que $(A \odot u = 1 \Leftrightarrow B \odot u \neq 1)$ et si au contraire $A \equiv B$, alors la complétude de $\mathcal{D}$ entraine que l'on peut trouver une preuve finie (dans ce système $\mathcal{D}$) de l'identité $A \equiv B$. La décidabilité de $A \equiv B$? résulte facilement de ces remarques.

Applications et perspectives Depuis 1966, de nombreux avatars du "problème de l'équivalence des automates à pile déterministes" ont été identifiés, et sont ipso facto résolus par l'algorithme ci-dessus convenablement composé avec des réductions (au sens de Turing). L'un des plus significatifs est le problème de "l'équivalence sémantique des schémas de programme": peut-on décider, étant donnés deux programmes A, B, s'ils calculent la *même* fonction pour *toute* interprétation possible des symboles d'opérations qu'ils utilisent (par exemple $0, 1, +, *$, $\mathtt{if} - \mathtt{then} - \mathtt{else}$)? En composant le théorème de Sénizergues avec des réductions dûes à Courcelle (1978) ou Gallier (1981), on obtient un algorithme qui résout ce problème. L'extension de ces méthodes à d'autres types d'automates ou de schémas de programme semble possible et a d'ores et déjà conduit à de nouveaux résultats ([2]).

Bibliographie

[1] G. Sénizergues. L(A) = L(B)? Technical report, LaBRI, Université Bordeaux I, rapport nr1161-97, disponible à l'URL, http://www.labri.u-bordeaux.fr/, 1997.

Version très courte dans les actes de ICALP'97, LNCS 1256, p.671-681;
Version courte dans les actes de INFINITY'97, p.1-26, URL: http://www.elsevier.nl/locate/entcs/volume9.html
Version longue, Avril 1998, soumise à Theoretical Computer Science, disponible à l'URL, http://www.labri.u-bordeaux.fr/~ges, pages 1-166.

[2] G. Sénizergues. Decidability of the bisimulation problem for equational graphs of finite out-degree. Technical report, LaBRI, Université Bordeaux I, rapport nr 1183-97, disponible à l'URL, http://dept-info.labri.u-bordeaux.fr/ ges, 1997. Pages 1-113,
Version courte à paraître dans les actes de FOCS'98.

2 Contacts

Chercheur: Géraud Sénizergues, LaBRI, Université Bordeaux 1
tel:05-56-84-60-48
fax:05-56-84-66-69
adresse électronique: ges@labri.u-bordeaux.fr
http://dept-info.labri.u-bordeaux.fr/~ges/
Équipe: Langages Formels et Graphes,
Responsable: Bruno Courcelle,
http://www.labri.u-bordeaux.fr/Equipe/LangForG/
Laboratoire: LaBRI,UMR 5800/Bordeaux1/ENSERB.
351 Cours de la liberation -33405-Talence Cedex FRANCE
Directeur: André Arnold,
http://www.labri.u-bordeaux.fr/