Cons-free Programs and Complexity Classes between LOGSPACE and PTIME (invited talk)

Neil D. Jones
Computer Science Department
University of Copenhagen
neil@diku.dk

Siddharth Bhaskar
Computer Science Department
University of Copenhagen
sbhaskar@di.ku.dk

Cynthia Kop
Department of Software Science
Radboud University. Nijmegen
C.Kop@cs.ru.nl

Jakob Grue Simonsen
Computer Science Department
University of Copenhagen
simonsen@di.ku.dk

Programming language concepts are used to give some new perspectives on a long-standing open problem: is LOGSPACE = PTIME ?

Introduction

"P =? NP" is an archetypical question in computational complexity theory, unanswered since its formulation in the 1970s. The question: Is the computional power of polynomially time-bounded programs increased by adding the ability to "guess" (i.e., nondeterminism) ? This is interesting because "polynomial time" is a plausible candidate for "feasibly solvable".

Perhaps the second most important question is "L =? P": whether LOGSPACE = PTIME. Here L is the set of problems solvable by *cursor programs*. These also run in polynomial time, but have no rewritable storage[1]. Both questions remain open since Cook and Savitch's pathbreaking papers in the 1970s [1, 7].

We investigate the question "L =? P" from the viewpoint of *functional programming languages*: a different viewpoint than Turing machines. The link is earlier characterisations of L and P by "cons-free" programs [3, 4, 5]. The net result: a deeper and finer-grained analysis, illuminated by perspectives both from programming languages and complexity theory.

Some new definitions and theorems give fresh perspectives on the question L =? P. We use programs to define and study complexity classes between the two. By [3, 4, 5] cursor programs exactly capture the problem class L; and cursor programs with *recursive function definitions* exactly capture the problem class P. A drawback though is that recursive cursor programs *can run for exponential time*, even though they exactly capture the *decision problems* that can be solved in polynomial time by Turing machines.

The goal of this talk is to better understand the problems in the interval between classes L and P. Problem class NL is already-studied in this interval, and it is the logspace analog of similar long-standing open problems. Kuroda's two "LBA problems" posed in 1964 [6]: (1) Is DSPACE(n) =? NSPACE(n) and (2) Is NSPACE(n) closed under complementation? After both stood unresolved for 23 years, (2) was finally answered "yes" (independently in 1987) by Immerman and by Szelepcsényi [2, 8]: NL and larger nondeterministic space classes (with constructive bounds) are closed under complementation.[2]

[1] One take: a cursor program is a multihead two-way read-only finite automaton. A more classical but equivalent version: a 2-tape Turing machine with n-bit read-only input tape 1, that uses at most $O(\log n)$ bits of storage space on read-write tape 2.

[2] Kuroda's other LBA problem DSPACE(n) =? NSPACE(n) is still open, as well as the question L =? NL.

Submitted to:
VPT 2020

We study the problems solvable by an in-between class CFpoly: recursive cursor programs that *run in polynomial time*. Recursion is in some sense orthogonal to the ability to make nondeterministic choices, i.e., to "guess". The class CFpoly seems more natural than NL from a programming perspective.

References

[1] S. A. Cook (1971): *Characterizations of pushdown machines in terms of time-bounded computers*. *Journal of the ACM* 18(1), pp. 4–18.

[2] Neil Immerman (1988): *Nondeterministic space is closed under complementation*. *SIAM J. Comput.* 17(5), pp. 935–938.

[3] Neil D. Jones (1997): *Computability and complexity - from a programming perspective*. Foundations of computing, MIT Press. 1 edition.

[4] Neil D. Jones (1999): *LOGSPACE and PTIME characterized by programming languages*. *Theoretical Computer Science* 228(1-2), pp. 151–174.

[5] Neil D. Jones (2001): *The expressive power of higher-order types or, life without CONS*. *Journal of Functional Programming* 11(1), pp. 55–94.

[6] Sige-Yuki Kuroda (1964): *Classes of languages and linear-bounded automata*. *Information and Control* 7(2), pp. 207–223.

[7] Walter J. Savitch (1970): *Relationships between nondeterministic and deterministic tape complexities*. *J. Comput. Syst. Sci.* 4(2), pp. 177–192.

[8] Róbert Szelepcsényi (1988): *The method of forced enumeration for nondeterministic automata*. *Acta Inf.* 26(3), pp. 279–284.