Queue Automata of Constant Length

Sebastian Jakobi[1,⋆] Katja Meckel[1,⋆]
Carlo Mereghetti[2,⋆,⋆⋆] Beatrice Palano[2,⋆,⋆⋆]

[1] Institut für Informatik, Universität Giessen, Arndtstr. 2, 35392 Giessen, Germany
{jakobi,meckel}@informatik.uni-giessen.de
[2] Dip. Informatica, Univ. degli Studi di Milano, v. Comelico 39, 20135 Milano, Italy
{mereghetti,palano}@di.unimi.it

Abstract. We introduce and study the notion of *constant length queue automata*, as a formalism for representing regular languages. We show that their descriptional power outperforms that of traditional finite state automata, of constant height pushdown automata, and of straight line programs for regular expressions, by providing optimal exponential and double-exponential size gaps. Moreover, we prove that constant height pushdown automata can be simulated by constant length queue automata paying only by a linear size increase, and that removing nondeterminism in constant length queue automata requires an optimal exponential size blow-up, against the optimal double-exponential cost for determinizing constant height pushdown automata.

1 Introduction

It is well known that *computational power* can be tuned by restricting the way memory is accessed. To get a quick overview of this phenomenon, one may start from the traditional model of one-way Turing machines, where memory is modeled by a potentially unbounded working tape that can be freely accessed. If we impose a LIFO usage of the working tape, still keeping unboundedness, then we obtain *pushdown automata*, whose computational power (context-free languages) is strictly lower. On the other hand, by imposing a FIFO access policy, we get *queue automata*, whose power gets back to that of Turing machines.

In all cases, by fixing a *constant* bound — i.e., not depending on the input length — on the amount of available memory, the computational power boils down to that of finite state automata, regardless of memory usage mode. For *constant memory machines*, it is then worth investigating how the way in which memory is accessed affects their *descriptional power*, in other words, their capability of succinctly representing regular languages.

This line of research is settled in [7], where the notion of a *constant height pushdown automaton* is introduced and studied from a descriptional complexity

⋆ Partially supported by CRUI/DAAD under the project "Programma Vigoni: Descriptional Complexity of Non-Classical Computational Models".
⋆⋆ Partially supported by MIUR under the project "PRIN: Automi e Linguaggi Formali: Aspetti Matematici e Applicativi".

perspective. Roughly speaking, this device is a traditional pushdown automaton (see, e.g., [8]) with a built-in constant limit on the height of the pushdown. Optimal exponential and double-exponential gaps are proved between the size of constant height deterministic and nondeterministic pushdown automata (DPDAs and NPDAs, respectively), deterministic and nondeterministic finite state automata (DFAs and NFAs), and that of classical regular expressions. Moreover, also the notion of a straight line program for regular expressions (SLP) is introduced, as a formalism equivalent to a constant height NPDA from a size point of view. In [3, 2], the fundamental problem of removing nondeterminism in constant height NPDAs is tackled, and a double-exponential size blow-up for determinization is emphasized. Finally, the descriptional cost of boolean operations on constant height DPDAs and NPDAs is analyzed in [4, 6].

We investigate the descriptional advantages of substituting the pushdown with a *queue* storage of fixed size by introducing the notion of a *constant length queue automaton*. Basically, this device is a traditional queue automaton (see, e.g., [1, 5]), where the length of the queue cannot grow beyond a fixed constant limit.

2 Results

For the sake of conciseness, we will be using the designation "constant memory automaton" to denote either a constant height NPDA or a constant length NQA. For constant memory automata, a fair *size measure* (see, e.g., [4, 7]) takes into account all the components the device consists of, i.e.: (i) the number of states of its finite control, (ii) the size of the memory alphabet, and (iii) the memory limit.

As for constant height pushdown automata, we single out optimal exponential and double-exponential gaps between the size of constant height deterministic and nondeterministic queue automata (DQAs and NQAs, respectively) and DFAs and NFAs. However, differently from constant height pushdown automata, we prove that a queue storage enables a size-efficient handling of nondeterminism. Precisely, we show that NFAs can be simulated by constant length DQAs paying by only a *linear* size increase. This, in turn, leads us to prove that the optimal cost of removing nondeterminism in constant length NQAs is only *exponential*, in sharp contrast with the optimal double-exponential blow-up above pointed out for the pushdown case.

The higher descriptional power of a queue vs. a pushdown storage in a constant setting is also emphasized when we show that constant height NPDAs (resp., DPDAs) can be simulated by constant length NQAs (resp., DQAs), paying by only a *linear* size increase. On the other hand, the opposite simulations have optimal exponential costs; this is witnessed by proving, that constant length DQAs can be exponentially smaller than equivalent SLPs, and this gap is optimal.

For reader's ease of mind, we sum up in Figure 1 the main relations on the sizes of the different types of formalisms for regular languages considered in this paper:

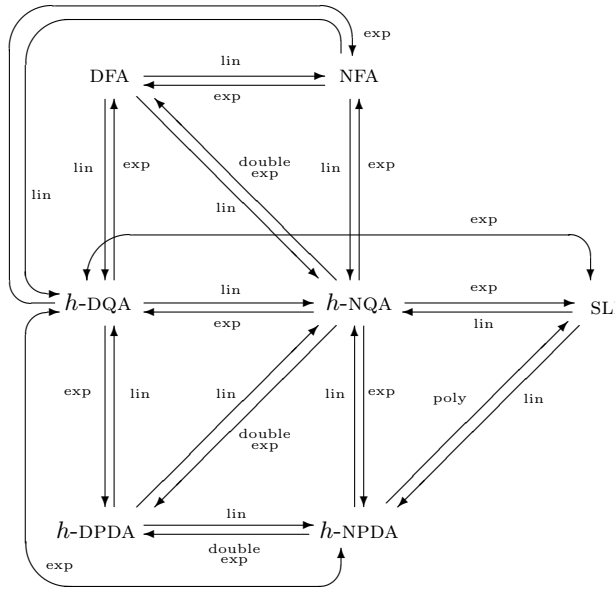

Fig. 1. Costs of simulations among different types of formalisms defining regular languages. Here h-DPDA (h-NPDA) denotes constant height DPDA (NPDA, respectively), while h-DQA (h-NQA) denotes constant length DQA (NQA, respectively). An arc labeled by lin (poly, exp, double exp) from a vertex A to a vertex B means that, given a representation of type A, we can build an equivalent representation of type B, paying by a linear (polynomial, exponential, double-exponential, respectively) increase in the size.

References

1. E. Allevi, A. Cherubini, and S. Crespi-Reghizzi. Breadth-first phrase structure grammars and queue automata. In *MFCS*, volume 324 of *Lecture Notes in Computer Science*, pages 162–170. Springer, 1988.
2. Z. Bednárová, V. Geffert, C. Mereghetti, and B. Palano. Removing nondeterminism in constant height pushdown automata. *Submitted for publication.*
3. Z. Bednárová, V. Geffert, C. Mereghetti, and B. Palano. Removing nondeterminism in constant height pushdown automata. In *DCFS*, volume 7386 of *Lecture Notes in Computer Science*, pages 76–88. Springer, 2012.
4. Z. Bednárová, V. Geffert, C. Mereghetti, and B. Palano. The size-cost of boolean operations on constant height deterministic pushdown automata. *Theor. Comput. Sci.*, 449:23–36, 2012.
5. A. Cherubini, C. Citrini, S. Crespi-Reghizzi, and D. Mandrioli. Qrt fifo automata, breath-first grammars and their relations. *Theor. Comput. Sci.*, 85(1):171–203, 1991.
6. V. Geffert, Z. Bednárová, C. Mereghetti, and B. Palano. Boolean language operations on nondeterministic automata with a pushdown of constant height. In *CSR*, volume 7913 of *Lecture Notes in Computer Science*, pages 100–111. Springer, 2013.
7. V. Geffert, C. Mereghetti, and B. Palano. More concise representation of regular languages by automata and regular expressions. *Inf. Comput.*, 208(4):385–394, 2010.
8. J. E. Hopcroft, R. Motwani, and J. D. Ullman. *Introduction to automata theory, languages, and computation - (2. ed.).* Addison-Wesley series in computer science. Addison-Wesley-Longman, 2001.