AUTOBÓZ 2019

OPEN PROBLEMS IN AUTOMATA, LOGIC, GAMES, AND RELATED TOPICS

FIRBUSH, SCOTLAND

THIS IS AN INTERNAL DOCUMENT FOR AUTOBÓZ 2019, FIRBUSH, SCOTLAND

HTTP://AUTOBOZ.ORG/

Last compilation: 28th June, 2019

Contents

Introduction

How would I describe myself? Three words:
Hard working, alpha male, jack hammer.
Merciless. Insatiable.
Dwight Schrute

The core *raison d'être* of Autobóz is to gather a group of automata enthusiasts around open problems, and let their insatiable curiosity do the rest. For this edition, we offered to put together a short compendium of open problems in automata, logic, games, and related topics. The hope is that this booklet will find a use before, during, and after the workshop.

The problems are divided into topics, but this should be seen as a loose—if not arbitrary—classification. Arguably, lots of these problems could have been filed in another category, although it would have impacted the very nice balance we seem to have achieved. Each problem appears on its own page, leaving some space for scribbles and thoughts.

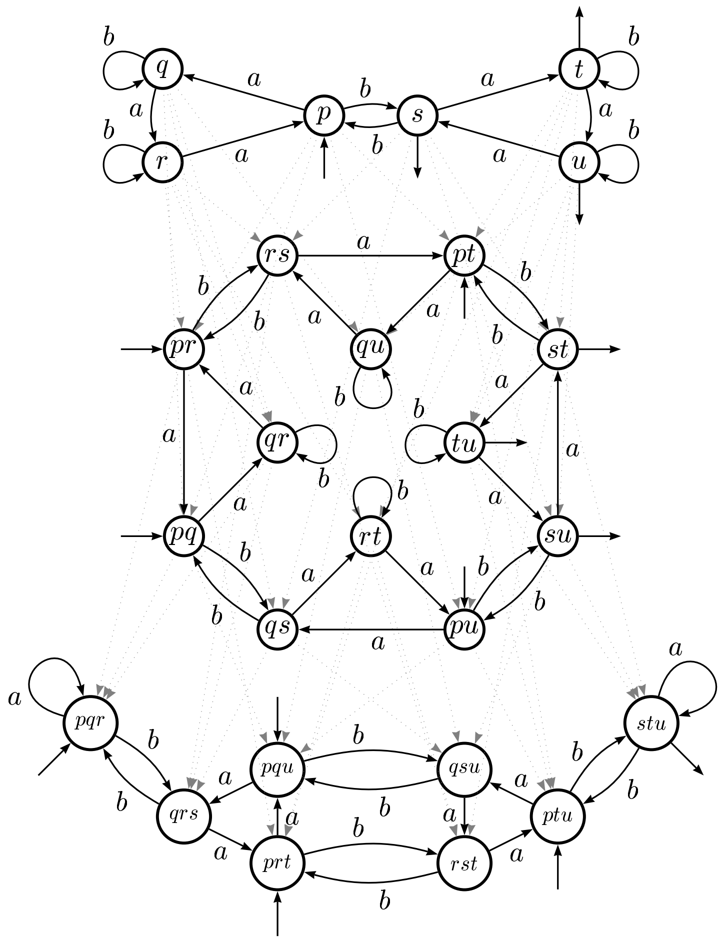
We wish all of you a wonderful Autobóz. Here's to many more!

Michaël Cadilhac
On behalf of Dmitry Chistikov, Patrick Totzke, and me
28th June 2019

List of participants:

- Michael Blondin <michael.blondin@usherbrooke.ca>
- Marie van den Bogaard <marie.van.den.bogaard@ulb.ac.be>
- Michaël Cadilhac <michael@cadilhac.name>
- Dmitry Chistikov <d.chistikov@warwick.ac.uk>
- Lorenzo Clemente <clementelorenzo@gmail.com>
- Wojtek Czerwiński <wczerwin@mimuw.edu.pl>
- Javier Esparza <esparza@in.tum.de>
- Marie Fortin <marie.fortin@lsv.fr>
- Piotrek Hofman <piotrek.hofman@gmail.com>
- Ismaël Jecker <ismael.jecker@gmail.com>
- Sandra Kiefer <kiefer@informatik.rwth-aachen.de>
- Ranko Lazic <r.s.lazic@warwick.ac.uk>
- Karoliina Lehtinen <k.lehtinen@liverpool.ac.uk>
- Nathan Lhote <nlhote@mimuw.edu.pl>
- Emanuel Martinov <eo.martinov@gmail.com>
- Corto Mascle <corto.mascle@ens-paris-saclay.fr>
- Filip Mazowiecki <filip.mazowiecki@u-bordeaux.fr>
- Charles Paperman <charles.paperman@inria.fr>
- Vincent Penelle <vincent.penelle@labri.fr>
- Michał Pilipczuk <michal.pilipczuk@gmail.com>
- Radosław Piórkowski <r.piorkowski@mimuw.edu.pl>
- David Purser <d.j.purser@warwick.ac.uk>
- Sylvain Schmitz <sylvain.schmitz@lsv.ens-cachan.fr>
- Salomon Sickert <sickert@in.tum.de>
- Michał Skrzypczak <mskrzypczak@mimuw.edu.pl>
- Patrick Totzke <totzke@liverpool.ac.uk>
- Georg Zetzsche <georg@mpi-sws.org>

Chapter 1
Automata & Grammars



1.1 Universality for unambiguous automata

By Wojciech Czerwiński `<wczerwin@mimuw.edu.pl>`

Determine the complexity of universality problem for unambiguous finite automata. Problem is known to be in PTIME and (as far as I understand) in NC^2, but is only known to be NL-hard.

1.2 Combining automata

By Wojciech Czerwiński <wczerwin@mimuw.edu.pl>

Assume there are DFAs $A_1, \ldots, A_n$ such that $L(A_i)$ form a partition of Σ^* and each A_i has at most k states. We would like to design a DFA B such that for every i there is set S_i of states of B s.t. $L(A_i) = L(B, S_i)$, where by $L(B, S)$ we denote language of B with set of accepting states S. We can easily design B as product of A_i, then size of B is at most k^n. We can skip A_n and it works as well, then size of B is at most k^{n-1}. Can we do much better in general? Like $k^{\log n}$. This would have interesting consequences.

1.3 Separating words problem

By Wojciech Czerwiński `<wczerwin@mimuw.edu.pl>`

Given two words u, v of length n over $\{a, b\}$. Does there always exist an automaton of size $O(\log n)$ which distinguishes u and v. Or bigger, like $O(n^{1/3})$? Best known bound is about $O(n^{2/5} \log(n))$.

1.4 On unambiguous grammars

By Lorenzo Clemente `<clementelorenzo@gmail.com>`

The universality problem for context-free grammars is undecidable. For deterministic pushdown automata it is PTIME-complete. For unambiguous context-free grammars, which constitute a model of intermediate expressive power, the problem is still decidable. Using the existential theory of the reals (the existential fragment of Tarski's algebra), it can be shown to be in PSPACE. However, no lower-bound better than PTIME is known. Improving either the PSPACE upper or the PTIME lower-bound would be a major advance on this problem!

A related problem is universality of the union of a deterministic pushdown automaton and an unambiguous finite automaton.

While this problem reduces to the universality problem of unambiguous context-free grammars, a reduction in the other direction seems unlikely. The known complexity bounds are the same as for the more general problem.

1.5 Context-freeness of duplications

By Michael Blondin <michael.blondin@usherbrooke.ca>

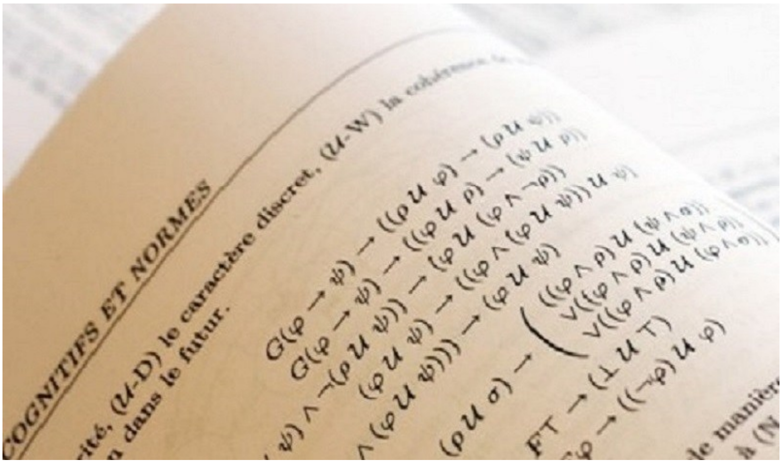
Let Σ be a finite alphabet. For every $L \subseteq \mathbb{N}_+$ and for every $u, v \in \Sigma^+$, let $u \to_L v$ if $v = xyyz$ for some $x, y, z \in \Sigma^*$ such that $|y| \in L$. Let $D_L(u) = \{v : u \xrightarrow{*}_L v\}$.

Is $D_L(w)$ context-free for every $w \in \Sigma^+$ and every $L \in \{\mathbb{N}_+\} \cup \{[n] : n > 0\}$? The case where $L = \mathbb{N}_+$ is open, while the case where $L = [n]$ is claimed to be answered positively in https://doi.org/10.1007/978-3-540-24635-0_22.

In Theorem 4, the authors give a PDA for $D_{[n]}(w)$ which applies contractions until reaching w. However, there appears to be a nontrivial flaw as the "memory" stored in the states of the PDA is unbounded. The same construction is reproduced by a common author in https://doi.org/10.1007/11779148_22.

This question arised from a related bioinformatics question asked by a colleague. I would like to know whether $D_{[n]}(w)$ is indeed context-free. The case of $D_{\mathbb{N}_+}(w)$ is also interesting, but appears to be much more challenging.

Chapter 2
Logic



2.1 HyperLTL satisfiability

By Corto Mascle `<corto.mascle@ens-paris-saclay.fr>`

HyperLTL is an extension of LTL which allows quantification over traces. HyperLTL formulas are generated by the grammar:

$\phi ::= \forall\pi.\phi \mid \exists\pi.\phi \mid \psi$

$\psi ::= a_\pi \mid \neg\psi \mid \psi \vee \psi \mid X\psi \mid \psi U \psi$

with the intuitive semantics. For example the formula

$$\forall\pi.\exists\pi'.F(a_{\pi'} \wedge \neg a_\pi) \wedge G(a_\pi \Rightarrow a_{\pi'})$$

expresses that for all traces there exists another trace satisfying a at strictly more positions; $\{a\}^*\emptyset^\omega$ is a model of this formula. There are two main versions of the satisfiability problem for this logics: Given a formula ϕ, one can ask if there exists any set of traces satisfying ϕ, or if there exists a finite Kripke structure whose runs induce a set of traces satisfying ϕ. Unfortunately, both problems are undecidable in general. This motivates us to look for restrictions on parameters such as temporal depth, number of quantifier alternations or of universal quantifiers which would allow us to reach decidability. For the general problem, only very small fragments are decidable. However there is some hope in the other case. In particular, we do not know if it is decidable, given a HyperLTL formula ϕ of temporal depth one, whether ϕ is satisfied by the set of traces of some Kripke structure (the problem is TOWER-hard). If it is not, then which other restrictions would make it decidable?

2.2 SafeLTL

By Karoliina Lehtinen <k.lehtinen@liverpool.ac.uk>

An ω-word language $L \subseteq \Sigma^\omega$ is a safety language if for all $w \notin L$ there is a prefix $v \in \Sigma^*$ of w such that for all $u \in \Sigma^\omega$, $vu \notin L$.

safeLTL is the Until-free fragment of negation normal form LTL. A safeLTL formula describes a safety property. Oddly, the converse seems to be open and non-trivial: if an LTL formula describes a safety property, is it expressible in safeLTL?

This seems likely, but can we prove it? Better, can we find an effective transformation of safety LTL to safeLTL? Or can we find an LTL expressible safety property which is not expressible in safeLTL? Either outcome would be interesting for the area of runtime verification where safety properties are particularly important and LTL is the dominant formalism.

From an automata perspective, LTL formulas correspond to very weak alternating automata—i.e. alternating automata with only self loops. For safeLTL, all states must be accepting, except for a rejecting sink.

2.3 Membership in logical classes for (succinct) finite automata

By Charles Paperman `<charles.paperman@univ-lille.fr>`

Finite automata can be represented concisely in many way, including enriching their description with bounded-size stacks and registers, by allowing them to be two-way, etc...

Little is known of the complexity of checking membership of those representation of finite automaton into logical formalism (such as LTL, First-Order Logic). Most of the time, classical algorithms relies on testing algebraic properties of the transition semigroup of the automaton. The transition semigroup itself is exponential in the size of the automaton. A straight forward application of those methods to concise representation will lead to tower of exponential algorithms.

For specific kind of concise representations, smarter algorithms could lead to only an exponential blowup.

2.4 Logic and automata for multiply nested-words

By Marie Fortin <marie.fortin@lsv.fr>

Nested-word automata (NWA) are essentially visibly pushdown automata with multiple stacks. They accept languages of (multiply) nested-words, which are words equipped with a fixed number of binary nesting relations, connecting matching push and pop events. In the case of a single stack (that is, a single nesting relation), NWA are equivalent to monadic second-order logic (MSO). The binary predicates used in the logic are the direct successor relation, and the nesting relations. However, in the general case, and even with only two stacks, NWA are not closed under complementation, and strictly less expressive than MSO. In fact, for two stacks, NWA are equivalent to the existential fragment of MSO (EMSO). The equivalence with MSO can be recovered when restricting the set of possible behaviors, one classical example being phase-bounded nested words.

In the general case (more than two stack, arbitrary nested words), the exact expressive power of NWA is not known. In particular, the following question is open: can every first-order formula be translated into an equivalent NWA ?

Chapter 3
Games

© King

3.1 Games with non-classic ω-regular winning conditions

By Salomon Sickert <sickert@in.tum.de>

Classic winning conditions for infinite games on graphs, e.g. safety, Büchi, coBüchi, Rabin, Streett, parity, or Muller, are well-studied and a range of practical algorithms solving games with such acceptance conditions are available. The standard approach to LTL synthesis first translates a LTL specification to a deterministic ω-automaton and then reinterprets the structure of the automaton as a two-player game. Here, Rabin, Streett, parity, and Muller acceptance conditions are predominantly are used, since they can be used to express all LTL-definable languages. However, on deterministic ω-automata with these acceptance conditions (Rabin, Streett, parity, Muller) at least one Boolean operation (union, intersection) cannot be implemented efficiently and incurs an exponential blow-up in the worst-case [1]. This is a challenge for compositional automata constructions, since the final deterministic automaton cannot be constructed by cheap Boolean combinations of smaller automata. "Non-classic" acceptance conditions [1], such as Hyper-Rabin, Hyper-Streett, or Emerson-Lei, allow a higher complexity in the acceptance condition in order to offer union and intersection with only a quadratic cost, which enables the design of compositional constructions. However, at the moment little is known about these winning conditions:

Questions:

- What is the complexity of solving games with non-classic winning conditions, e.g., Hyper-Rabin, Hyper-Streett, or Emerson-Lei?
- How much memory is required for a winning strategy?
- What about practical algorithms?

[1] Udi Boker: Why These Automata Types? LPAR 2018

3.2 *Quantitative termination value approximation in OC-MDPs*

By Emanuel Martinov `<s1133141@sms.ed.ac.uk>`

One-Counter MDPs are a class of infinite-state MDPs, that extend finite-state MDPs via an addition of an unbounded counter, which is kept throughout the game play and that changes by ≤ 1 at each transition. A *configuration* is a pair (q, c) of a control state q and an integer counter value $c \in \mathbb{Z}$. A *run* is an infinite sequence of such configurations $\omega = (q_0, c_0)(q_1, c_1)\ldots$, where (q_0, c_0) is a given starting configuration, and for $j \geq 1 : (q_{j-1}, c_j - c_{j-1}, q_j)$ is a valid transition in the game.

Our focus for this problem is on the *termination* objective, which is the set of runs, whose counter eventually becomes 0, at which point the process terminates, i.e., $Term := \{\omega = (q_0, c_0)(q_1, c_1)\cdots \mid \exists k > 0 \; s.t. \; c_k \leq 0\}$. And in particular, the problem of interest is the *quantitative termination approximation problem*: given a maximizing/minimizing OC-MDP, an initial configuration $(q, c), c > 0$ and a rational $\epsilon > 0$, compute a rational value v, such that $|Val^{Term}(q, c) - v| \leq \epsilon$, where $Val^{Term}(q, c)$ denotes the optimal termination probability starting in (q, c). Furthermore, the goal is also to compute an ϵ-optimal strategy for the player. This problem has been already studied [Brázdil-Brožek-Etessami-Kučera'11], providing a (non-deterministic) exponential time algorithm for (OC-SSGs) OC-MDPs.

The goal of this open problem is to determine whether the complexity can be improved from exponential to polynomial time for the termination value approximation problem for OC-MDPs.

3.3 Solvency games

By Emanuel Martinov `<eo.martinov@gmail.com>`

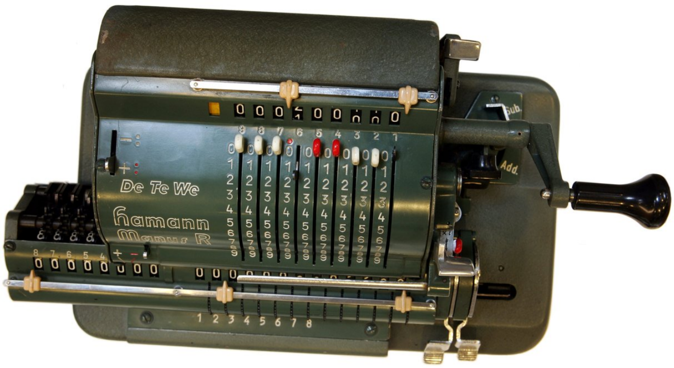
Here is another (subsumed) problem, spawning from a study [Berger-Kapur-Schulman-Vazirani'08].

Solvency games are an interesting sub-class of *minimizing* OC-MDPs. Solvency games are 1-player probabilistic processes with a single control state and a counter, representing the wealth of the player. In the process, actions for the player are presented via a probability distribution over integer payoffs, i.e. at each time t the player changes their wealth (w) by a value, sampled from the distribution associated with their chosen action $a \in A$. The game stops when the player goes bankrupt, i.e., $w \leq 0$. The objective is not to maximize expected long-term profit, but rather to maximize probability of survival (or solvency). That is why the player desires, for a given initial wealth w, to find a strategy π, which assigns choice from a set of actions (A) to each wealth state, such that π minimizes the probability of bankruptcy ($p^{\pi}(w)$).

An essential open question for solvency games is whether an optimal value and an optimal strategy can be computed efficiently. Also, the study shows that in general there is no *rich man's* strategy π, where $\forall w > w' \ : \ \pi(w) = \pi(w')$ (i.e., where there is a clear best choice for every wealth value above a certain threshold). However, they show a special case of restrictions, under which such a *rich man's* strategy does exist and the optimal strategy can be computed in exponential time. Like the previous problem, again we ask whether a more efficient way of computing the optimal strategy exists. But also analysing under what other conditions a *rich man's* strategy exists.

I present both of these (related) open problems, which I have previously studied together with my supervisor, Kousha Etessami.

Chapter 4
Counter Machines



4.1 Variants of one-counter systems universality

By Patrick Totzke `<totzke@liverpool.ac.uk>`

Problem 1:[1] Given a 1-VASS, let L_n be its language where acceptance is by reaching a final state from a fixed initial state and initial counter value n. Does there exist n such that $\Sigma^* = L_n$?

Problem 2: Given a 1-VASS, let L^n be the language of the n-bounded system (the NFA where values 0..n are hard-coded) where acceptance is by reaching a final state from a fixed initial configuration. Does there exist n such that $\Sigma^* \subseteq L^n$?

Questions:

- Are these problems decidable?
- Are they inter-reducible?
- What about trace languages?
- Is there a direct reduction from the seemingly simpler Universality Problem (fixed initial configuration) to either of these problems?

[1] Whether this problem is decidable is also a question suggested by Piotrek Hofman

4.2 Shortest runs in 3-D VASS

By Wojciech Czerwiński `<wczerwin@mimuw.edu.pl>`

Is there a 3-D VASS with shortest run from source to target, which is longer than exponential? Best lower bound is exponential, best upper bound is tower. The case of VASSes with finite reachability set is probably the core of the problem. In my point this is one of the problems we should attack if we want to push theory of VASSes further.

4.3 *Reachability for bounded branching VASS*

By Filip Mazowiecki `<filip.mazowiecki@u-bordeaux.fr>`

Recall that a d-VASS is a finite automaton, where transitions are labelled with d-dimensional vectors over integers. A configuration of a d-VASS is a pair of a state and a d-dimensional vector over naturals. Bounded VASS (BoVASS) are a variant of the classic VASS model where all values in all configurations are upper bounded by a fixed natural number, encoded in binary in the input. It is easy to see that the reachability is in PSPACE for BoVASS and it is not hard to show NP-hardness. In 2013 Fearnley and Jurdziński proved that the reachability problem in this model is PSPACE-hard already in dimension 1. We investigate the complexity of the reachability problem when the BoVASS model is extended with branching transitions (BoBrVASS). Branching transitions create two independent copies of the system splitting the configuration vector among them. For BoBrVASS it is easy to show that the reachability problem is in EXPTIME and PSPACE hardness follows from the results about BoVASS. Recently we proved that the reachability problem is EXPTIME-hard for d-BoBrVASS when $d \geq 2$, leaving the case for $d = 1$ as an open problem for Autobóz.

This problem is also suggested by Michał Pilipczuk.

4.4 *Time complexity of reachability in a variant of 2-D VASS*

By Dmitry Chistikov <d.chistikov@warwick.ac.uk>

Consider a variant of 2-dimensional vector addition systems with states (VASS) in which:

- the first counter is bounded by n and can be tested for 0 and n; and
- the second counter is unbounded and can be tested for 0.

Suppose the system is fixed, and we are given n in unary as the sole input. Can we decide reachability faster than in time $n^3/\mathrm{polylog}(n)$?

This problem is a special case of language recognition for two-way pushdown automata.

4.5 Does a (min, +)-WA preserve REG by inverse image?

By Nathan Lhote <nlhote@mimuw.edu.pl>

We are interested in functions realized by weighted automata over the semiring $\mathbb{N}_{\mathsf{min}} = \langle \mathbb{N} \cup \{\infty\}, \mathsf{min}, +, \infty, 0\rangle$. An $\mathbb{N}_{\mathsf{min}}$-weighted automaton $\mathscr{A}$ over alphabet Σ realizes a partial function $[\![\mathscr{A}]\!] : \Sigma^* \to \mathbb{N}$. We want to decide given such a function if it preserves "simple" sets by inverse image. By simple we mean regular languages ($\mathbb{N}$ is viewed a the set of words over a unary alphabet, hence the regular languages are the semilinear sets). Let us state the decision problem:

Input: $\mathscr{A}$, $\mathbb{N}_{\mathsf{min}}$-weighted automaton
Question: Does it hold that for all $S \subseteq \mathbb{N}$ semilinear, $[\![\mathscr{A}]\!]^{-1}(S)$ is regular?

Remark. This is equivalent to solving the problem over the semiring $\mathbb{N}_{\mathsf{max}} = \langle \mathbb{N} \cup \{-\infty\}, \mathsf{max}, +, -\infty, 0\rangle$.

4.6 Continuous reachability in ordered data VAS

By Piotrek Hofman <piotrek.hofman@gmail.com>

Data VAS is a finite set of finitely supported functions from ordered data set $\mathbb{D}$ to $\mathbb{Z}^k$, where k is a dimension. A configuration/marking is a finitely supported function from $\mathbb{D}$ to $\mathbb{N}^k$.

From a configuration m there is a step to m' if there is a vector $x \in$ VAS and π an ordered preserving bijection (permutation) form $\mathbb{D}$ to $\mathbb{D}$ such that $m + x \circ \pi = m'$. The reachability relation is a transitive closure of the step relation. The reachability problem is undecidable, that is why we are looking for different relaxation of it.

Continuous reachability is a continuous version of the reachability. First, markings are finitely/supported functions from $\mathbb{D}$ to $\mathbb{Q}^k_{\geq 0}$. Further there is a continuous step from m to m' if there are: $x \in$ VAS, an order preserving data permutation π, and a factor $a \in \mathbb{Q}_{\geq 0}$ such that $m + a \cdot x \circ \pi = m'$. A transitive closure of continuous step is a continuous reachability relation.

4.7 Deciding upperboundedness of $\mathbb{Z}$-CCRA

By Michaël Cadilhac <michael@cadilhac.name>

A cost register automaton (CRA) over $(\mathbb{Z}, \min, +)$ is a DFA equipped with a finite number of registers that take values in $\mathbb{Z}$. Each transition induces a transformation of the registers formulated with min and +; for instance $x \leftarrow \min\{x, y+3\} + z$. Accepting states can then output one of the registers.

Consider the model in which:

- Each update uses only min's and addition *with constants*,
- For each transition, no register appears twice on the right-hand side of the updates.

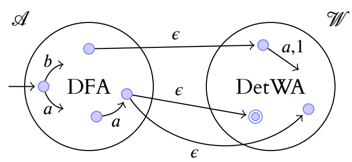
This model, which we call $\mathbb{Z}$-CCRA (the extra C standing for *copyless*), can express for instance:

$$f(a^{i_1}\# \ldots \# a^{i_n}) = \min\{i_1, \ldots, i_n\} \ ,$$

but couldn't express $g(b^i \# w) = i + f(w)$—this would require either copying i, or adding two registers.

In [1], we showed that equivalence is undecidable for that model (even, and this is much more interesting, when restricted to $\mathbb{N}$). We also showed that it is undecidable, given a $\mathbb{Z}$-CCRA, to decide whether the function it expresses is always negative.

For weighted automata people, note that this model is equivalent to this, where $\mathscr{W}$ is a deterministic WA over $(\mathbb{Z}, \min, +)$:



Question: Is it decidable, given a $\mathbb{Z}$-CCRA, whether the function it expresses is upper-bounded? That is, whether $(\exists c)(\forall w)[f(w) \leq c]$.

[1] S. Almagor, M. Cadilhac, F. Mazowiecki, G. A. Pérez. Weak Cost Register Automata are Still Powerful. DLT'18. https://arxiv.org/abs/1804.06336

Chapter 5
Other Models of Computation

5.1 The busy beaver problem for population protocols

By Javier Esparza <esparza@in.tum.de>

Population protocols are a model of distributed computation by indistinguishable agents, close to VASs. We consider population protocols with one input state. These protocols compute predicates $\mathbb{N} \to \{0, 1\}$. (For definitions see https://arxiv.org/abs/1801.00742)

For every $n \geq 1$, let $f(n)$ be the largest number such that some protocol with at most n states computes the predicate $x < f(n)$.

It is shown in the above paper that $f(n) \in 2^{2^{\Omega(n)}}$ for protocols with leaders. This is all we know about $f(n)$.

Open problems:

- Give a bound on $f(n)$ for protocols with leaders.
- Give a bound on $f(n)$ for protocols without leaders.

5.2 *Weak validation of tree-languages by extended automata*

By Charles Paperman `<charles.paperman@univ-lille.fr>`

The weak validation problem is an open problem stated in 2007 by Segoufin and Sirangelo. The problem is about checking if a regular langage of trees, seen as a languages of word (with an XML encoding), can be *validated* by a finite automaton with the assumption that the input is a correct tree encoding. More formally, let T be the set of words encoding correct trees, K be a regular language of tree with XML(K) the associate languages of words. Can we decide if there exists a regular langage of words L such that $L \cap T = \mathrm{XML}(K)$.

Weak validation sounds out of reach and a very difficult question although generalisation of the problem considering extended version of finite automata (e.g. Parikh automaton) could be in same time easier to deal with and provides a more interesting class of weakly validating regular languages of trees.

Some recent work on a close subject:

Eryk Kopczyński. 2016. Invisible Pushdown Languages. LICS '16

5.3 Existence of a universal amplifier of selection

By Ismaël Jecker <ismael.jecker@gmail.com>

Moran Birth-death processes are stochastic processes defined as follows. Consider a connected graph G and a parameter $r \in \mathbb{R}$ strictly greater than 1. We start with a partition of the vertices of G into two types: *mutant* vertices, that have *fitness* r, and *resident* vertices, that have fitness 1. At each step, a random vertex is chosen with probability proportional to its fitness, and spreads its type to an adjacent vertex chosen uniformly at random.

With probability 1, the process eventually reaches either fixation of the mutation (all the vertices are mutant) or extinction (all the vertices are resident). The *fixation probability* of a vertex v of G is the probability that the process starting with a single mutant at v eventually reaches fixation. The fixation probability of each vertex of the complete graph on $n \in \mathbb{N}$ vertices is

$$p_n = \frac{1 - r^{-1}}{1 - r^{-n}}.$$

Open problem: Does there exist a connected graph G with $n \in \mathbb{N}$ vertices such that the fixation probability of each vertex of G is strictly greater than p_n?