Cost Functions and Value 1 problem in practice

Nathanaël Fijalkow, **Denis Kuperberg**

LIAFA/Institute of Informatics, Warsaw University, Poland
ONERA/IRIT, Toulouse

December 18th, 2014

An algorithmic back-end: Stabilization Monoids

What?

- An algebraic structure with two operations: a binary composition and a unary operator $\sharp$,
- Generalizes the transition monoid of a non-deterministic automaton to two weighted settings.

Where? When?

- First appeared in the Theory of Regular Cost Functions [Colcombet 2009],
- Later used for Probabilistic Automata [Fijalkow, Gimbert, Oualhadj 2012].

Regular Cost functions

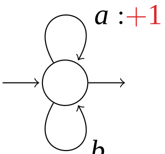
***B*-automata**

- Non-deterministic, and are enriched with *counters* with operations: $+1$, r
- Introduced to generalize proofs on regular languages: finite substitution, star-height.
- Semantic is a function : $A^* \to \mathbb{N} \cup \{\infty\}$,

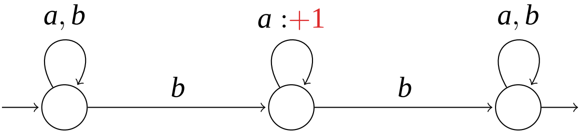
Questions on *B*-automata

- **Equivalence**: bounded on same sets of words,
- **Boundedness**: equivalent to 0 (bounded).

Examples of B-Automata



$\mathcal{A}_1$: number of a

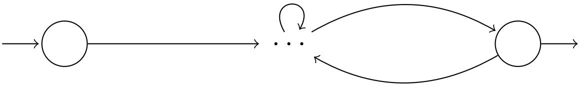


$\mathcal{A}_2$: smallest block of a

Not equivalent: $\mathcal{A}_1$ is not bounded on $(ab)^*$ but $\mathcal{A}_2$ is.

Reductions from Finite Power to Boundedness

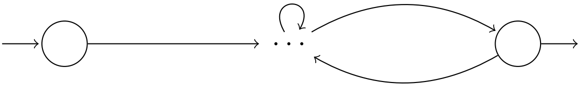
Problem: Given L, is there $n \in \mathbb{N}$ such that $L^* = (L + \epsilon)^n$?



Reductions from Finite Power to Boundedness

Problem: Given L, is there $n \in \mathbb{N}$ such that $L^* = (L + \epsilon)^n$?
Solution:

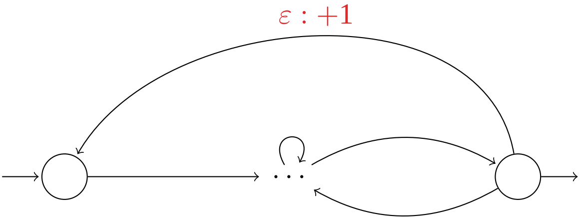
1. Start with an automaton for L.



Reductions from Finite Power to Boundedness

Problem: Given L, is there $n \in \mathbb{N}$ such that $L^* = (L + \epsilon)^n$?
Solution:

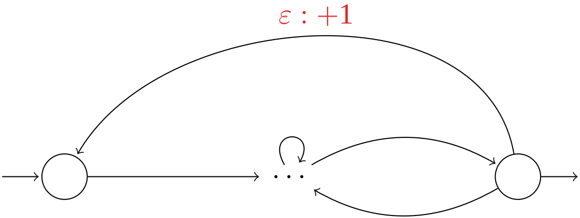
1. Start with an automaton for L.
2. Add increment ε-transitions from final states to initial.



Reductions from Finite Power to Boundedness

Problem: Given L, is there $n \in \mathbb{N}$ such that $L^* = (L + \epsilon)^n$?
Solution:

1. Start with an automaton for L.
2. Add increment ε-transitions from final states to initial.
3. Decide boundedness



Problems solved using counters

- **Finite Power** (finite words) [Simon '78, Hashiguchi '79]
  Is there n such that $(L+\varepsilon)^n = L^*$?

- **Fixed Point Iteration** (finite words)
  [Blumensath+Otto+Weyer '09]
  Bound on the number of fixpoint iterations in a MSO formula?

- **Star-Height** (finite words/trees)
  [Hashiguchi '88, Kirsten '05, Colcombet+Löding '08]
  Given n, is there an expression for L, with at most n nesting of Kleene stars?

- **Parity Rank** (infinite trees)
  [reduction in Colcombet+Löding '08, deterministic input Niwinski+Walukiewicz '05, Büchi input K.+Vanden Boom '11 CKLV '13]
  Given $i < j$, is there a parity automaton for L using ranks $\{i, i+1, \ldots, j\}$?

Problem for probabilistic Automata

Probabilistic automata

- Transitions are distributions $\delta(q, a) : Q \to [0, 1]$,
- $P_{\mathcal{A}}(w)$ is the probability that $\mathcal{A}$ accepts w.

Value 1 problem: Given $\mathcal{A}$, is there a sequence w_n such that $P_{\mathcal{A}}(w_n) \to 1$?

Problem for probabilistic Automata

Probabilistic automata

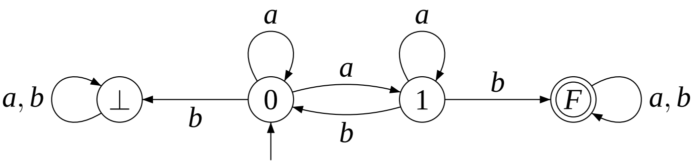
- Transitions are distributions $\delta(q, a) : Q \to [0, 1]$,
- $P_{\mathcal{A}}(w)$ is the probability that $\mathcal{A}$ accepts w.

Value 1 problem: Given $\mathcal{A}$, is there a sequence w_n such that $P_{\mathcal{A}}(w_n) \to 1$?

Undecidable, but decidable in a restricted case: leaktight automata.

Intuition: leaks allow ”competitive” behaviours, outcome depends on fine tuning of transitions.

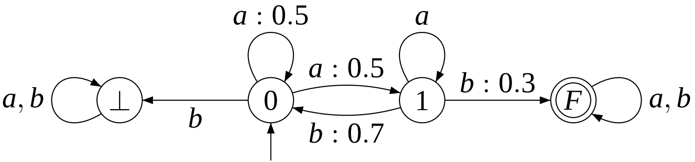
Matrix Representation



$$\langle a \rangle = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad \langle b \rangle = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$I \cdot \langle u \rangle \cdot F = 1$ if and only if u is accepted.

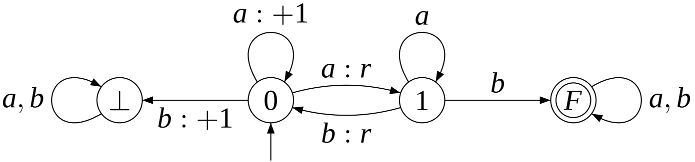
Probabilistic automata



$$\langle a \rangle = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.5 & 0.5 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad \langle b \rangle = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0.7 & 0 & 0.3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$I \cdot \langle u \rangle \cdot F = P_{\mathcal{A}}(u)$$

B-automata



$$\langle a \rangle = \begin{pmatrix} 0 & \bot & \bot & \bot \\ \bot & 1 & r & \bot \\ \bot & \bot & 0 & \bot \\ \bot & \bot & \bot & 0 \end{pmatrix} \qquad \langle b \rangle = \begin{pmatrix} 0 & \bot & \bot & \bot \\ 1 & \bot & \bot & \bot \\ \bot & r & \bot & 0 \\ \bot & \bot & \bot & 0 \end{pmatrix}$$

$$I \cdot \langle u \rangle \cdot F = f(u)$$

Computing in Infinite Semirings

Consider either the rational semiring $(\mathbb{Q}, +, \times)$ or the tropical semiring $(\mathbb{N} \cup \{\infty\}, \min, +)$:

- An automaton $\mathcal{A}$ is given by a matrix $\langle a \rangle$ for each letter $a \in A$,
- We would like to finitely represent $\{\langle u \rangle \mid u \in A^*\}$.

Computing in Infinite Semirings

Consider either the rational semiring $(\mathbb{Q}, +, \times)$ or the tropical semiring $(\mathbb{N} \cup \{\infty\}, \min, +)$:

- An automaton $\mathcal{A}$ is given by a matrix $\langle a \rangle$ for each letter $a \in A$,
- We would like to finitely represent $\{\langle u \rangle \mid u \in A^*\}$.

So we abstract away the precise values and consider two operators:

- a binary composition law: matrix multiplication,
- a stabilization unary operator $\sharp$.

Computing in Infinite Semirings

Consider either the rational semiring $(\mathbb{Q}, +, \times)$ or the tropical semiring $(\mathbb{N} \cup \{\infty\}, \min, +)$:

- An automaton $\mathcal{A}$ is given by a matrix $\langle a \rangle$ for each letter $a \in A$,
- We would like to finitely represent $\{\langle u \rangle \mid u \in A^*\}$.

So we abstract away the precise values and consider two operators:

- a binary composition law: matrix multiplication,
- a stabilization unary operator $\sharp$.

Intuitively, $\langle u \rangle^\sharp$ represents $\lim_n \langle u^n \rangle$.

Stabilization operations

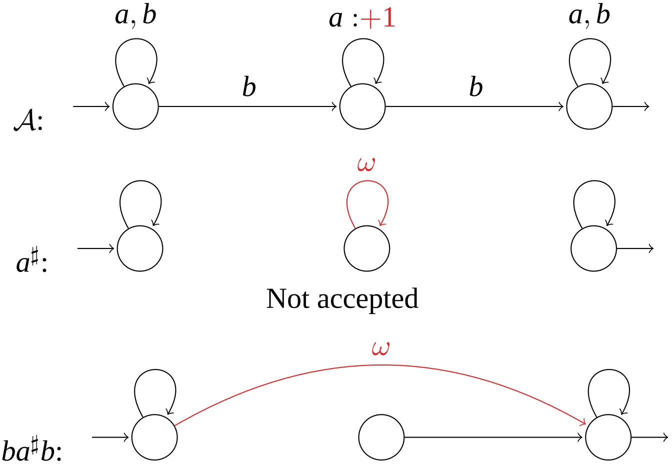
Only defined on idempotents.

B-**automata:**
Replace $+1$ by ω in the matrix, meaning “unbounded”.

Probabilisatic Automata:
Model long runs $\rightarrow$ Keep only edges going to ergodic strongly connected components.

No reduction from one framework to the other on the automata level $\rightarrow$ versatility of stabilization monoids.

Example of computation



Not accepted

Accepted, but not within some bound.

Formal definition

Formally, a stabilization monoid is $(M, \cdot, \sharp, \leq)$ such that:

- $(M, \cdot, \leq)$ is an ordered monoid (associativity, monotonicity of $\cdot$),
- $\sharp$ is a fonction from idempotents to idempotents,
- $x \leq x^\sharp$,
- $x^\sharp = xx^\sharp = (x^\sharp)^\sharp$.
- $(xy)^\sharp x = x(yx)^\sharp$

The order puts a constraint on accepting sets: it has to be upwards-closed.

Stabilization Monoid of an Automaton

Definition

The Stabilization Monoid of $\mathcal{A}$ is the closure of $\{\langle a\rangle \mid a \in A\}$ under both operators.

The Stabilization Monoid of $\mathcal{A}$ contains a lot of informations about $\mathcal{A}$!

Using the Stabilization Monoid

B-Automata

- Decide whether a B-automaton is bounded,
- Decide whether two B-automata are equivalent.

Probabilistic Automata

- Decide whether a probabilistic automaton has (probably) value 1,
- Decide whether a probabilistic automaton is leaktight.

Algorithmics elements of ACME

Saturation

- Hashtable for elements obtained, and queue for new candidates.
- Worst-case complexity EXPTIME.
- Current improvements: keep short names for elements and rewriting rules.

Minimization

- needed for equivalence checking,
- general Myhill-Nerode Equivalence, needs a new operator $\omega\sharp$.
- Union-find structure for partitions,
- Computation of abstract types,
- Polynomial in $|M|$ (exponential in $|\mathcal{A}|$).

Currently migrating from OCaml to C++, with new optimizations.

Demo of ACME

[Demo of ACME]

The end.

Thank you for your attention!