Expressiveness of Minmax Automata

Amaldev Manuel
LIAFA, Université Paris Diderot

joint work with

Thomas Colcombet
Stefan Göller

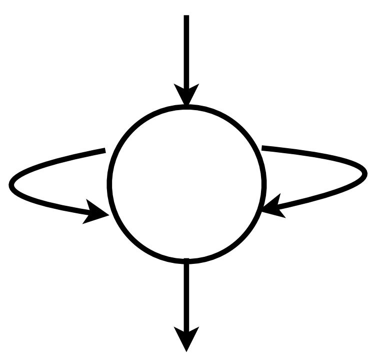
Minmax Automata

- Finite state automaton equipped with +ve-integer registers
- Registers updated using expressions composed of register names, min, max, +1
- Cost of a run = value of the output register at the end
  Cost of a word = minimum value of an accepting run
- Every automaton defines a function from $\mathbf{\Sigma^*}$ to $\mathbb{N} \cup \{\infty\}$

Minmax Automata

- Finite state automaton equipped with +ve–integer registers
- Registers updated using expressions composed of register names, min, max, +1
- Cost of a run = value of the output register at the end
  Cost of a word = minimum value of an accepting run
- Every automaton defines a function from **Σ*** to $\mathbb{N} \cup \{\infty\}$

a, $\texttt{blck} := \texttt{blck} + 1$



$$b,\quad \begin{array}{lcl} \texttt{blck} & := & 0 \\ \texttt{lrgst} & := & \max(\texttt{lrgst},\ \texttt{blck}) \\ \texttt{scndlrgst} & := & \max(\texttt{scndlrgst},\ \min(\texttt{lrgst},\ \texttt{blck})) \end{array}$$

$$a^{n_1} b a^{n_2} b \ldots a^{n_k} b \quad \rightarrow \quad \text{Second-largest}(n_1, n_2, \ldots, n_k)$$

Minmax Automata

- Subclasses : Min automata (Max automata) if expressions use only min (max) and +1
- Variants studied by Alur et. al., Bojańczyk, and Bojańczyk-Toruńczyk.
- Contains distance automata (by powerset like construction) hence equivalence/inclusion of automata is undecidable.

Minmax Automata

- Subclasses : Min automata (Max automata) if expressions use only min (max) and +1
- Variants studied by Alur et. al., Bojańczyk, and Bojańczyk-Toruńczyk.
- Contains distance automata (by powerset like construction) hence equivalence/inclusion of automata is undecidable.

Boundedness

Does there exist a $k \in \mathbb{N}$: for all words $w \in \Sigma^*$ $A(w) \leq k$?

Framework for solving Boundedness (Colcombet)

Cost equivalence

Functions f, g : Σ* → $\mathbb{N}$ ∪ {∞} are **cost equivalent** if over every subset L ⊆ Σ*, f is bounded ⇔ g is bounded

e.g. For $w = a^{n_1} b a^{n_2} b \dots a^{n_k} b$,
$$\begin{aligned} f(w) &= |w| \\ g(w) &= \max(\text{largest}(n_1, \dots, n_k), \#_b(w)) \end{aligned}$$

Framework for solving Boundedness (Colcombet)

Cost equivalence

Functions f, g : $\Sigma^* \to \mathbb{N} \cup \{\infty\}$ are **cost equivalent** if over every subset $L \subseteq \Sigma^*$, f is bounded ⇔ g is bounded

e.g. For $w = a^{n_1} b a^{n_2} b \dots a^{n_k} b$, $\begin{aligned} f(w) &= |w| \\ g(w) &= \max(\text{largest}(n_1, \dots, n_k), \#_b(w)) \end{aligned}$

- To solve boundedness, it is sufficient to consider Automata up to cost equivalence.
- Cost function — A class of the cost equivalence relation

Regularity for cost functions

Regular cost functions (Colcombet)

- Class of cost functions paralleling regular languages
- Strong closure properties (Boolean closure, projections, reversal, …)
- Alternate characterisations (automata, logic, algebra, regular expressions, …)
- decidability — boundedness, equivalence, domination

Regularity for cost functions

Regular cost functions (Colcombet)

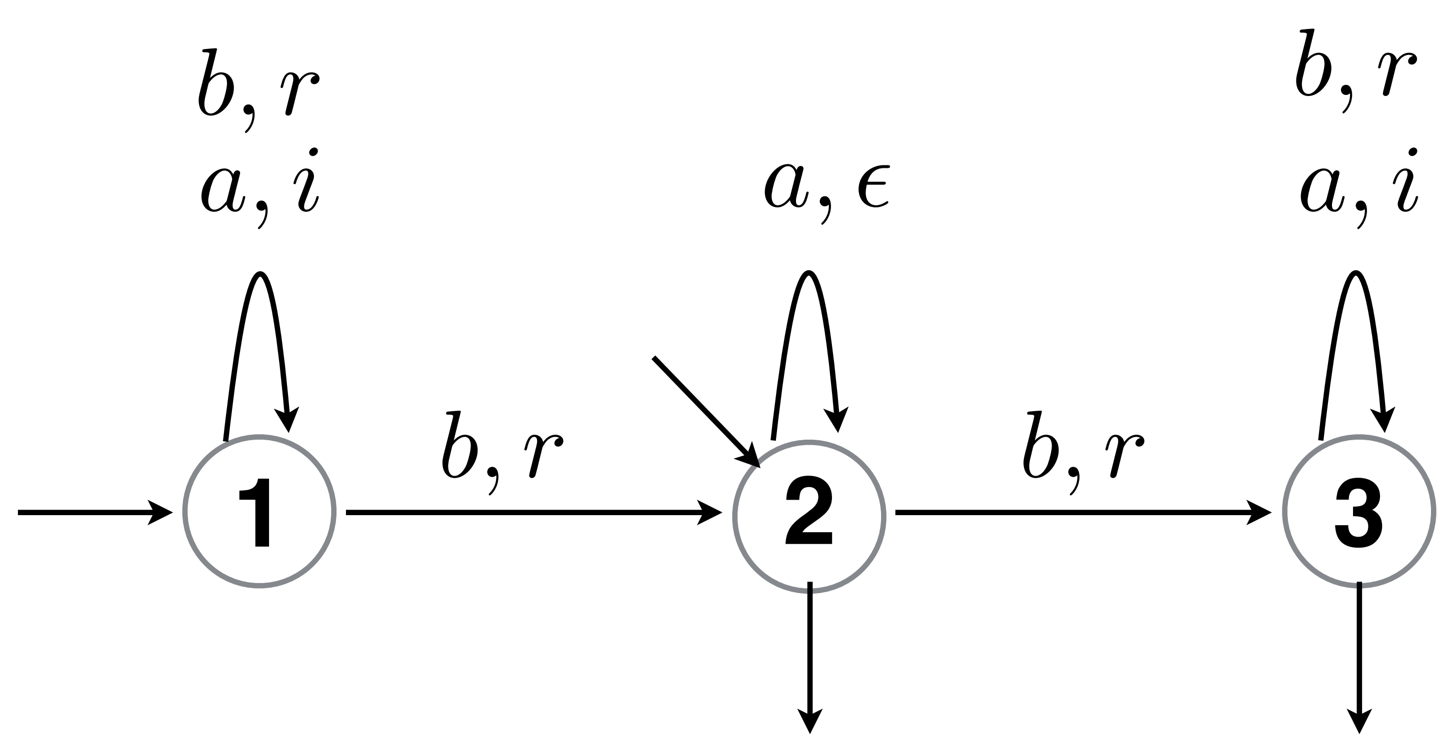
- Class of cost functions paralleling regular languages
- Strong closure properties (Boolean closure, projections, reversal, ...)
- Alternate characterisations (automata, logic, algebra, regular expressions, ...)
- decidability — boundedness, equivalence, domination

B-automaton (Abdulla-Krcal-Yi, Bojańczyk-Colcombet, Kirsten)

- Finite state automata extended with +ve–integer counters
- Operations on counters — increment, epsilon(no op), reset
- Cost of a run = maximal value of any counter during the run
  Cost of a word = minimum value of an accepting run

B-automaton (Abdulla-Krcal-Yi, Bojańczyk-Colcombet, Kirsten)

- Finite state automata extended with +ve-integer counters
- Operations on counters — increment, epsilon(no op), reset
- Cost of a run = maximum value of any counter during the run
  Cost of a word = minimum value of an accepting run



$$a^{n_1} b a^{n_2} b \ldots a^{n_k} b \rightarrow \text{Second-largest}(n_1, n_2, \ldots, n_k)$$

Minmax and B-automata

Theorem

Minmax automata ⊆ B-automata ⊆ history-deterministic Max automata

- History-determinism — nondeterminism can be resolved by looking at the history.
- First inclusion depends on the fact, alternating B-automata = B-automata. (Colcombet-Löding)
- Second inclusion is based on a new semantics for B-automata.

Minmax and B-automata

Theorem

Minmax automata ⊆ B-automata ⊆ history-deterministic Max automata

- History-determinism — nondeterminism can be resolved by looking at the history.
- First inclusion depends on the fact, alternating B-automata = B-automata. (Colcombet-Löding)
- Second inclusion is based on a new semantics for B-automata.

Corollary

Minmax and B-automata

Theorem

Minmax automata ⊆ B-automata ⊆ history-deterministic Max automata

- History-determinism — nondeterminism can be resolved by looking at the history.
- First inclusion depends on the fact, alternating B-automata = B-automata. (Colcombet-Löding)
- Second inclusion is based on a new semantics for B-automata.

Corollary

Boundedness of minmax automata is decidable.

Deterministic Minmax automata

Deterministic minmax automata — when the transition relation is a function.

Open Question. Does det. minmax automata subsume B-automata?

- Distance automata ✔ (1-counter B-automata with no reset)
- Desert automata (Bala) ✔ (1-counter B-automata with no epsilon)
- Distance Desert (Bala-Kirsten) **?** (1-counter B-automata)

Open Question. Does alternating distance automata subsume B-automata?

Det. min automata and det. max automata

- Strictly weaker
- Robust classes with many characterisations
- In particular det. min automata = distance automata.
- Decidable classes — given a B-automaton it is possible to check there is an equivalent det. min (det. max) automaton.
- Provides a way to prove inexpressibility results for weighted automata.

Det. min automata and det. max automata

Theorem (—, Kuperberg-Toruńczyk)

The following classes effectively coincide.

1. Deterministic max-automata.
2. 1-counter S-automata with no reset.
3. Smallest class containing **size** and closed under max, min with regular languages, sup-projections.
4. Cost regular expressions of the form $e_1\ e_2^s\ e_3$ where e_1, e_2, e_3 are regular expressions
5. Cost MSO formulas of the form $\forall X\ (\phi(x) \rightarrow |X| \geq n)$ where $\phi(x)$ is a MSO formula
6. Functions defined by a stab. monoid M and an Ideal I such that M,I has #-reduction : for any #-expression evaluating in I there is an expression obtained by erasing all but one # which is still in I.

The following classes effectively coincide.

1. Deterministic min-automata.
2. Distance automata.
3. Smallest class containing **size** and closed under min, max, inf-projections.
4. Cost regular expressions of the form with no * on top of a B.
5. Cost MSO formulas of the form $\exists X\ (\phi(x) \wedge |X| \leq n)$ where $\phi(x)$ is a MSO formula
6. Stab. monoid satisfying $(xe)^\sharp(xe^\sharp)^\sharp(xe)^\sharp = (xe)^\sharp$

Thank you