Bounds for Tree Automata with Polynomial Costs — Part II

Andreas Maletti

GRK-Workshop Gohrisch – February 20, 2004

Joint work of: Zoltán Fülöp, Zsolt Gazdag (University of Szeged, Hungary),
Björn Borchardt, A. M. (University of Technology, Dresden).

1. Tree Automata and Cost Functions
2. Cost-Finiteness
3. Boundedness Results

Tree Automata – Syntax

Definition: A *tree automaton* is a quadruple $M = (Q, \Sigma, \delta, F)$ where

(i) Q is a finite set of *states*,

(ii) Σ is a ranked alphabet of *input symbols*,

(iii) $\delta = \bigcup_{k \in \mathbb{N}} \delta^{(k)}$ with $\delta^{(k)} \subseteq Q^k \times \Sigma^{(k)} \times Q$ is a ranked alphabet of *transitions*, and

(iv) $F \subseteq Q$ is a set of *final states*.

Example: $M_E = (\{q_0, q_1, q, r\}, \{\sigma^{(2)}, \alpha^{(0)}\}, \delta_E, \{q_1, r\})$ with the following set δ_E of transitions.

$$\delta_E = \{\underbrace{(\varepsilon, \alpha, q_0)}_{\tau_1}, \underbrace{(\varepsilon, \alpha, q)}_{\tau_2}, \underbrace{(q_0 q_0, \sigma, q_0)}_{\tau_3}, \underbrace{(q_0 q, \sigma, q_1)}_{\tau_4}, \underbrace{(q_0 q, \sigma, q)}_{\tau_5}, \underbrace{(q q_1, \sigma, r)}_{\tau_6}, \underbrace{(rr, \sigma, r)}_{\tau_7}\}$$

Tree Automata – Semantics

Definition: Let $n \in \mathbb{N}$, $q_1, \ldots, q_n, q \in Q$, and $s \in T_\Sigma(X_n)$. The set $\Psi^q_{q_1 \ldots q_n}(s) \subseteq T_\delta(X_n)$ of $(q_1 \ldots q_n, q)$-*computations of* s is inductively defined by:

(i) Let $s = x_j$ for some $j \in [n]$, then

$$\Psi^q_{q_1 \ldots q_n}(s) = \begin{cases} \{x_j\} & \text{, if } q_j = q \\ \emptyset & \text{, otherwise} \end{cases} .$$

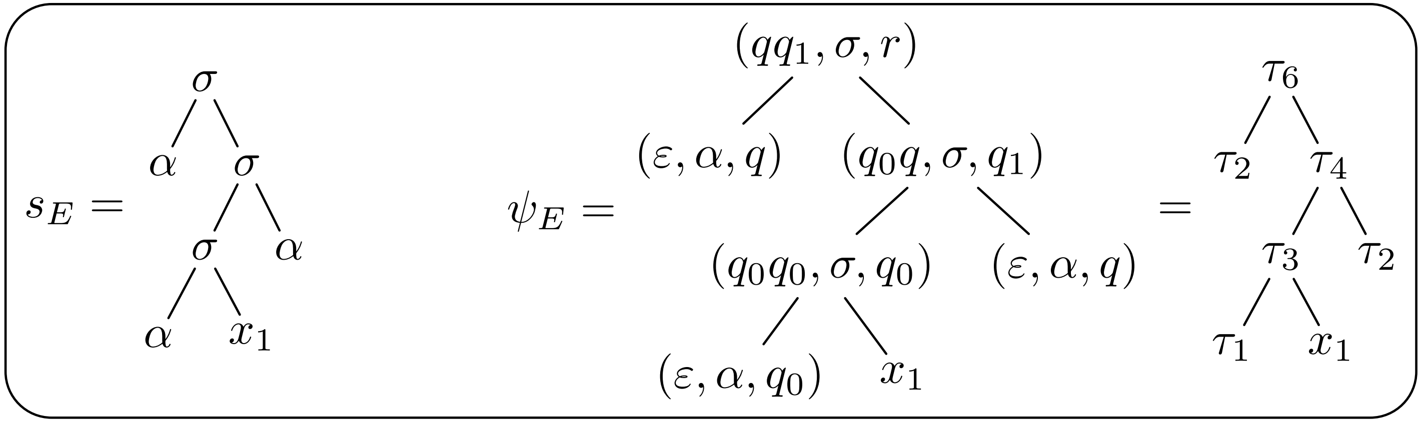
(ii) Let $s = \sigma(s_1, \ldots, s_k)$ for some $k \in \mathbb{N}$, $\sigma \in \Sigma^{(k)}$, $s_1, \ldots, s_k \in T_\Sigma(X_n)$.

$$\Psi^q_{q_1 \ldots q_n}(s) = \left\{ \tau(\psi_1, \ldots, \psi_k) \;\middle|\; \begin{array}{l} \tau = (r_1 \ldots r_k, \sigma, q) \in \delta^{(k)}, \\ \quad (\forall j \in [k]) : \; \psi_j \in \Psi^{r_j}_{q_1 \ldots q_n}(s_j) \end{array} \right\} .$$

Definition: The *language accepted by* M is

$$L(M) = \bigcup_{q \in F} \{ \, s \in T_\Sigma \mid \Psi^q(s) \neq \emptyset \, \} .$$

Tree Automata – Semantics (cont.)



An input tree $s_E \in T_{\{\sigma^{(2)}, \alpha^{(0)}\}}(X_1)$ and a (q_0, r)-computation $\psi_E \in T_{\delta_E}(X_1) \cap \Psi^r_{q_0}(s_E)$ of s.

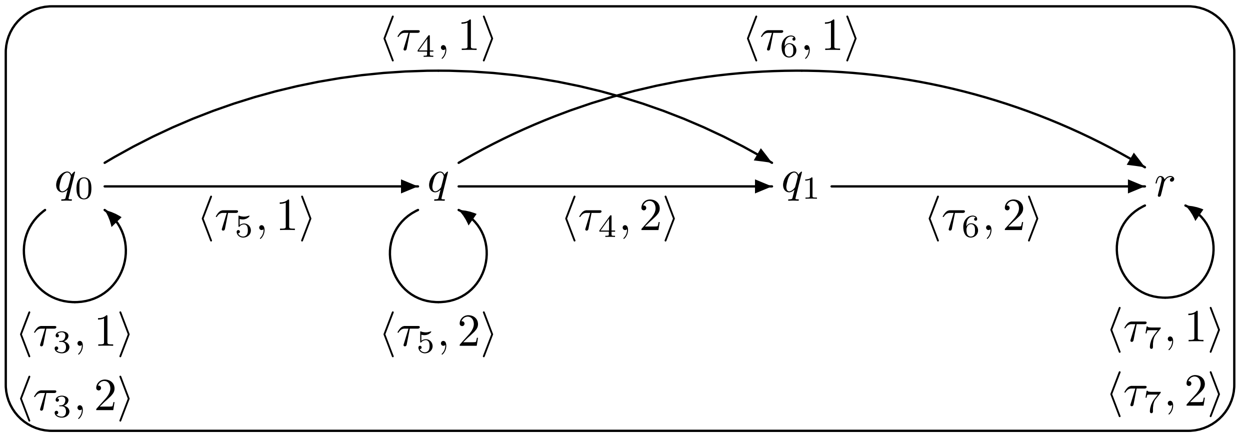
The language accepted by M_E is $L(M_E) = \{\, \sigma(s_1, s_2) \mid s_1, s_2 \in T_\Sigma \,\}$.

Tree Automata – Trace Graph

Definition: The *trace graph of* M is the labeled, directed graph $G(M) = (Q, E)$ where $E \subseteq Q \times (\delta \times \mathbb{N}_+) \times Q$ is

$$E = \left\{ (q', \langle \tau, j \rangle, q) \;\middle|\; k \in \mathbb{N}_+,\ j \in [k],\ \tau = (q_1 \ldots q_k, \sigma, q) \in \delta^{(k)},\ q' = q_j \right\}.$$

Example: The trace graph of M_E



Tree Automata – Trace Graph (cont.)

Definition: We define the equivalence relation $\equiv_M \subseteq Q \times Q$ as follows:

$$q \equiv_M q' \quad \iff \quad \text{if } q \text{ and } q' \text{ are strongly connected in } G(M).$$

Definition: We define the partial order $\leq_M \subseteq Q/_{\equiv_M} \times Q/_{\equiv_M}$ as follows:

$$[q]_{\equiv_M} \leq_M [q']_{\equiv_M} \quad \iff \quad q' \text{ is reachable from } q \text{ in } G(M).$$

Definition: Finally we define the partial order $\leq_M \subseteq Q \times Q$ as follows:

$$q \leq_M q' \quad \iff \quad [q]_{\equiv_M} \leq_M [q']_{\equiv_M} \text{ and } q \notin [q']_{\equiv_M} \setminus \{q'\}.$$

Example: $[q_0]_{\equiv_{M_E}} = \{q_0\}$, $[q]_{\equiv_{M_E}} = \{q\}$, $[q_1]_{\equiv_{M_E}} = \{q_1\}$, and $[r]_{\equiv_{M_E}} = \{r\}$.

$$[q_0]_{\equiv_{M_E}} <_{M_E} [q]_{\equiv_{M_E}} <_{M_E} [q_1]_{\equiv_{M_E}} <_{M_E} [r]_{\equiv_{M_E}} \quad \text{and} \quad q_0 <_{M_E} q <_{M_E} q_1 <_{M_E} r.$$

Decomposition of a Computation

Definition:

$$\Psi^q_{q_1 \ldots q_n} = \bigcup_{s \in T_\Sigma(X_n)} \Psi^q_{q_1 \ldots q_n}(s)$$

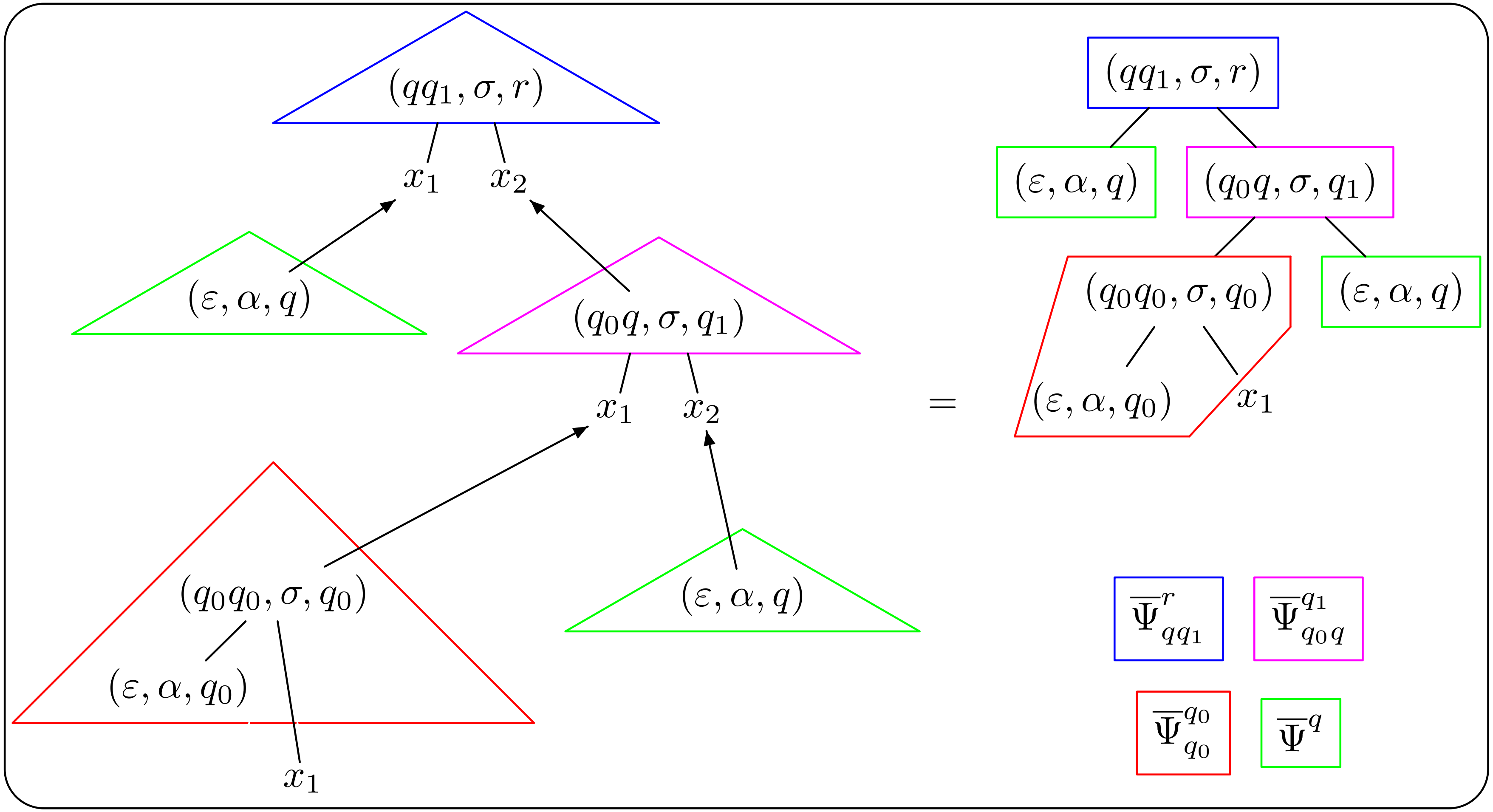
$$\widehat{\Psi}^q_{q_1 \ldots q_n} = \bigcup_{s \in \widehat{T_\Sigma}(X_n)} \Psi^q_{q_1 \ldots q_n}(s)$$

$$\delta_{(q)} = \left\{ (q_1 \ldots q_k, \sigma, q) \in \delta^{(k)} \mid k \in \mathbb{N},\, q_1, \ldots, q_k \in Q \right\}.$$

Definition: For every $n \in \mathbb{N}$, $q, q_1, \ldots, q_n \in Q$ we define the set

$$\overline{\Psi}^q_{q_1 \ldots q_n} = \left\{ \psi \in \Psi^q_{q_1 \ldots q_n} \,\middle|\, (\forall w \in \mathrm{pos}(\psi))(\exists r \in [q]_{\equiv_M}) :\ \mathrm{lab}_\psi(w) \in \delta_{(r)} \cup X_n \right\}.$$

Decomposition of a Computation (cont.)



Cost Functions

Definition: Let $M = (Q, \Sigma, \delta, F)$ be a tree automaton and $\mathcal{A} = (A, \oplus, \odot, \mathbf{0}, \mathbf{1})$ be a semiring. A mapping $c : \delta \longrightarrow A\langle X\rangle$ satisfying for every $k \in \mathbb{N}$ and $\tau \in \delta^{(k)}$ the condition that $c(\tau) \in A\langle X_k\rangle$ is called a *(polynomial) cost function for M*.

Example: The following mapping $c_E : \delta_E \longrightarrow \mathbb{N}\langle X\rangle$ is a cost function for M_E.

$$
\begin{array}{llll}
c_E(\tau_1) = 0 & c_E(\tau_2) = 2 & c_E(\tau_3) = 3x_1 + 4x_2 & c_E(\tau_4) = 3x_1x_2 \\
c_E(\tau_5) = 2x_1 + x_2 & c_E(\tau_6) = 5x_1 & c_E(\tau_7) = x_1 + x_2. &
\end{array}
$$

Definition: We extend c to a mapping $c : T_\delta(X) \longrightarrow A\langle X\rangle$ as follows.

(i) If $\psi = x$ for some $x \in X$, then $c(\psi) = x$.

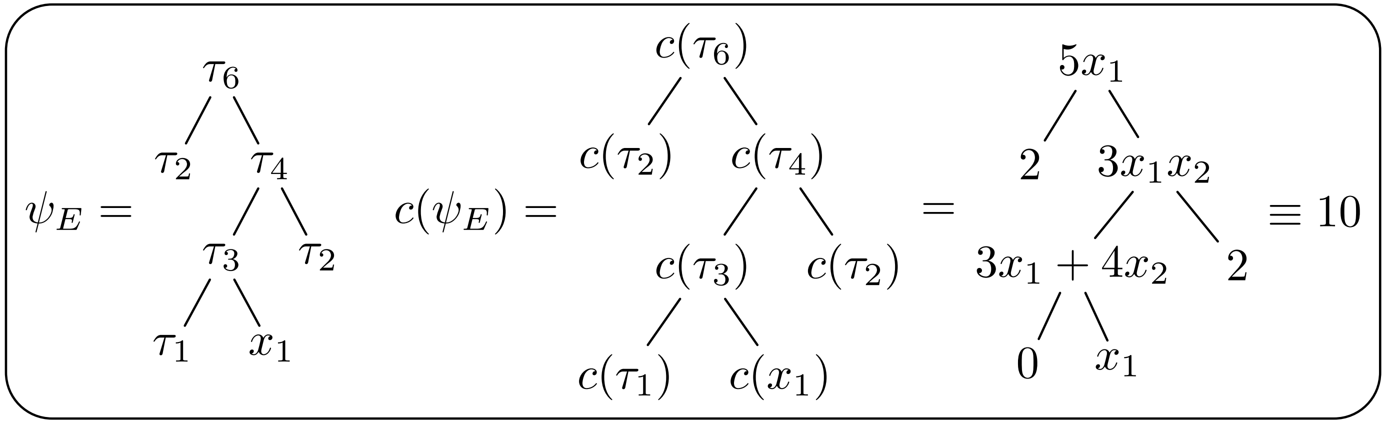
(ii) If $\psi = \tau(\psi_1, \ldots, \psi_k)$ for some $k \in \mathbb{N}$, $\tau \in \delta^{(k)}$, $\psi_1, \ldots, \psi_k \in T_\delta(X)$, then

$$c(\psi) = c(\tau)[c(\psi_1), \ldots, c(\psi_k)].$$

Cost Functions (cont.)

Definition: The *set of accepting costs* is

$$c(M) = \bigcup_{q \in F} c(M)_q = \bigcup_{q \in F} \{\, c(\psi) \mid \psi \in \Psi^q \,\}\,.$$



Example: $c_E(M_E)/_{\equiv} = \{[0]_{\equiv}, [10]_{\equiv}, [20]_{\equiv}, \ldots\}$.

E-states

Observation: Let $q, q' \in Q$, $\psi \in \Psi^q$, $\psi_2 \in \Psi^{q'}$, and $\psi_1 \in \Psi^q_{q'}$ such that $\psi = \psi_1[\psi_2]$. Then $c(\psi) = c(\psi_1)[c(\psi_2)]$.

Definition: For every $E \subseteq A$ we define the set $Q_E \subseteq Q$ of *E-states of M* to be

$$Q_E = \{\, q \in Q \mid (\forall \psi \in \Psi^q)(\exists e \in E) : \; c(\psi) \equiv e \,\} \,.$$

Clearly, $Q_A = Q$ and $Q_\emptyset = \emptyset$, because M is assumed to have no useless states.

Lemma: Provided that $\mathcal{A}$ is positive, one-summand free, and one-product free, we can effectively compute the sets $Q_{\{\mathbf{0}\}}$, $Q_{\{\mathbf{1}\}}$, and $Q_{\{\mathbf{0},\mathbf{1}\}}$.

Example: Using M_E with cost function c_E we observe that $Q_{\{0\}} = \{q_0, q_1\}$.

Algorithm Computing the $\{\mathbf{0}\}$-States

For every set S let $e_S : (X \cup A) \times S \times \mathcal{P}(S) \longrightarrow X \cup A$ be specified for every $z \in X \cup A$, $s \in S$, and $S' \subseteq S$ by

$$e_S(z, s, S') = \begin{cases} z & \text{, if } s \in S' \\ \mathbf{0} & \text{, otherwise} \end{cases} .$$

Require: M has no useless states, $\mathcal{A}$ is positive

$n := 0$, $Q_0 := \emptyset$

repeat

$$Q_{n+1} := Q_n \cup \left\{ q \in Q \;\middle|\; \begin{array}{r} k \in \mathbb{N},\ \tau = (q_1 \ldots q_k, \sigma, q) \in \delta^{(k)}, \\ c(\tau)[e_Q(x_1, q_1, Q_n), \ldots, e_Q(x_k, q_k, Q_n)] \not\equiv \mathbf{0} \end{array} \right\}$$

$n := n + 1$

until $Q_n = Q_{n-1}$

Ensure: $Q_{\{\mathbf{0}\}} = Q \setminus Q_n$

Reduced Tree Automata with Cost Functions

Definition: M is called *reduced*, if

(i) $\perp \in Q$ is the designated zero-state, i.e., $Q_{\{\mathbf{0}\}} = \{\perp\}$,

(ii) M possesses no useless states except potentially $\perp$,

(iii) for every $k \in \mathbb{N}$, $\tau = (q_1 \ldots q_k, \sigma, q) \in \delta^{(k)}$ with $q \in Q_{\{\mathbf{0},\mathbf{1}\}}$ we have $q_j = \perp$ for every $j \in [k]$ and $c(\tau) = e_Q(\mathbf{1}, q, Q \setminus \{\perp\})$,

(iv) for every $k \in \mathbb{N}$, $\tau = (q_1 \ldots q_k, \sigma, q) \in \delta^{(k)}$ we demand for every $j \in [k]$ that

  (a) if $c(\tau) = \mathbf{0}$, then $q_1, \ldots, q_k \in Q_{\{\mathbf{0},\mathbf{1}\}}$, and

  (b) else $c(\tau)$ is a zero-free polynomial and

$$x_j \in \mathrm{var}(c(\tau)) \qquad \Longleftrightarrow \qquad q_j \notin Q_{\{\mathbf{0},\mathbf{1}\}}.$$

Example: Clearly, M_E is not reduced because $Q_{\{0\}} = \{q_0, q_1\}$.

Reduced Tree Automata with Cost Functions (cont.)

Lemma: For every tree automaton M with cost function c over a positive, one-summand free, and one-product free semiring, a reduced tree automaton M' with cost function c' can effectively be constructed such that $c(M) \equiv c'(M')$.

Corollary: Let M be a reduced tree automaton with cost function c. For every $q, q' \in Q \setminus Q_{\{\mathbf{0},\mathbf{1}\}}$, if $\widehat{\Psi}^{q}_{q'} \neq \emptyset$, then there exists $\psi \in \widehat{\Psi}^{q}_{q'}$ such that $c(\psi)$ is zero-free and $x_1 \in \mathrm{var}(c(\psi))$.

Boundedness and Cost-Finiteness

Definition: Let $\preceq \subseteq A \times A$ be a partial order. M is said to be *bounded* (with respect to $\preceq$), if there exists $a \in A$ such that for every $a' \in A$ with $a' \equiv p$ for some $p \in c(M)$ we have $a' \preceq a$.

Definition: M is said to be *cost-finite*, if $c(M)/_{\equiv}$ is finite.

Lemma: Let $\mathcal{A}$ be a naturally ordered and finitely factorizing semiring and $A' \subseteq A$.

$$A' \text{ is finite} \quad \Longleftrightarrow \quad A' \text{ is bounded with respect to the natural order } \sqsubseteq .$$

Proof sketch: (i) $\Rightarrow$ (ii) Take $\sum_{a \in A'} a$ as an upper bound.
(ii) $\Rightarrow$ (i) Let $a \in A$ be such that $a' \sqsubseteq a$ for every $a' \in A'$. Then $A' \subseteq B_1 = \{\, b_1 \in A \mid b_1 \sqsubseteq a \,\}$. Assume that B_1 is infinite, then $B_2 = \{\, (b', b'') \mid a = b' \oplus b'' \,\}$ is also infinite, because

$$b_1 \sqsubseteq a \quad \Longleftrightarrow \quad (\exists b_2 \in A) : \; a = b_1 \oplus b_2.$$

Then $\mathcal{A}$ is not finitely factorizing. Contradiction, hence B_1 and A' are finite.

Cost-Finiteness – Step I

Lemma: Let M be a reduced tree automaton with cost function c over a finitely factorizing semiring.

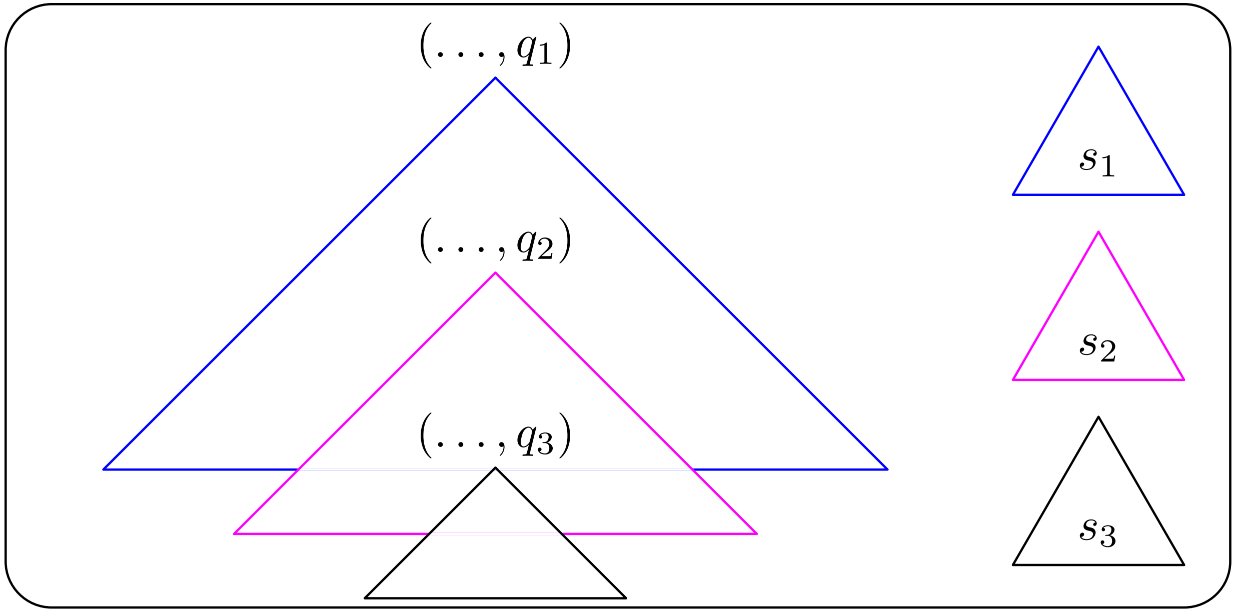
$$M \text{ is cost-finite} \quad \iff \quad (\forall q \in Q)(\forall a \in c(M)_q) : \; c(\widehat{\Psi}^q_q)[a]/_\equiv \text{ is finite}$$

Proof sketch:

$\Rightarrow$: indirect using finitely factorizing and reducedness

$\Leftarrow$: firstly prove that $(\forall q \in Q)(\forall r_1, r_2 \in [q]_{\equiv_M})(\forall a \in c(M)_{r_2}) : \; c(\widehat{\Psi}^{r_1}_{r_2})[a]/_\equiv$ is finite, then perform well-founded induction along $\leq_M$ using the outlined decomposition

Cost-Finiteness – Step I (cont.)



$s_1 \in \widehat{\Psi}^{q_1}_{q_2}$, $s_2 \in \widehat{\Psi}^{q_2}_{q_3}$, and $s_3 \in \Psi^{q_3}$. Note that $q_3 <_M q_2 <_M q_1$.

Cost-Finiteness – Step II

Definition: Condition *(finite)* holds, if for every $q \in Q \setminus Q_{\{\mathbf{0},\mathbf{1}\}}$ and $\psi \in \widehat{\Psi}^q_q$ there exists $a \in A$ such that either

(i) $c(\psi) \equiv x_1 + a$ and ($\mathcal{A}$ is additively idempotent or $a = \mathbf{0}$), or

(ii) $c(\psi) \equiv a$.

Lemma: Let M be a reduced tree automaton with cost function c over a monotonic and finitely factorizing semiring. If $c(\widehat{\Psi}^q_q)[a]/_{\equiv}$ is finite for every $q \in Q$ and $a \in c(M)_q$, then Condition (finite) holds.
Proof sketch: Assume that for some $q \in Q$ and $\psi \in \widehat{\Psi}^q_q$ Condition (finite) does not hold. Next show that for every $b \in A \setminus \{\mathbf{0}, \mathbf{1}\}$ we have $b \prec c(\psi)(b) \prec c(\psi^2)(b) \prec \cdots$ due to strictness with respect to multiplication. Contradiction.

Cost-Finiteness – Step III

Lemma: Let M be a tree automaton fulfilling Condition (finite) with cost function c over a non-idempotent semiring $\mathcal{A}$. Then for every $k \in \mathbb{N}$, $q \in Q$, and $\psi \in \Psi^q$ there exists $\psi' \in \Psi^q$ such that $c(\psi) \equiv c(\psi')$ and $\mathrm{height}(\psi') \leq 2 \cdot \mathrm{card}(Q)$.

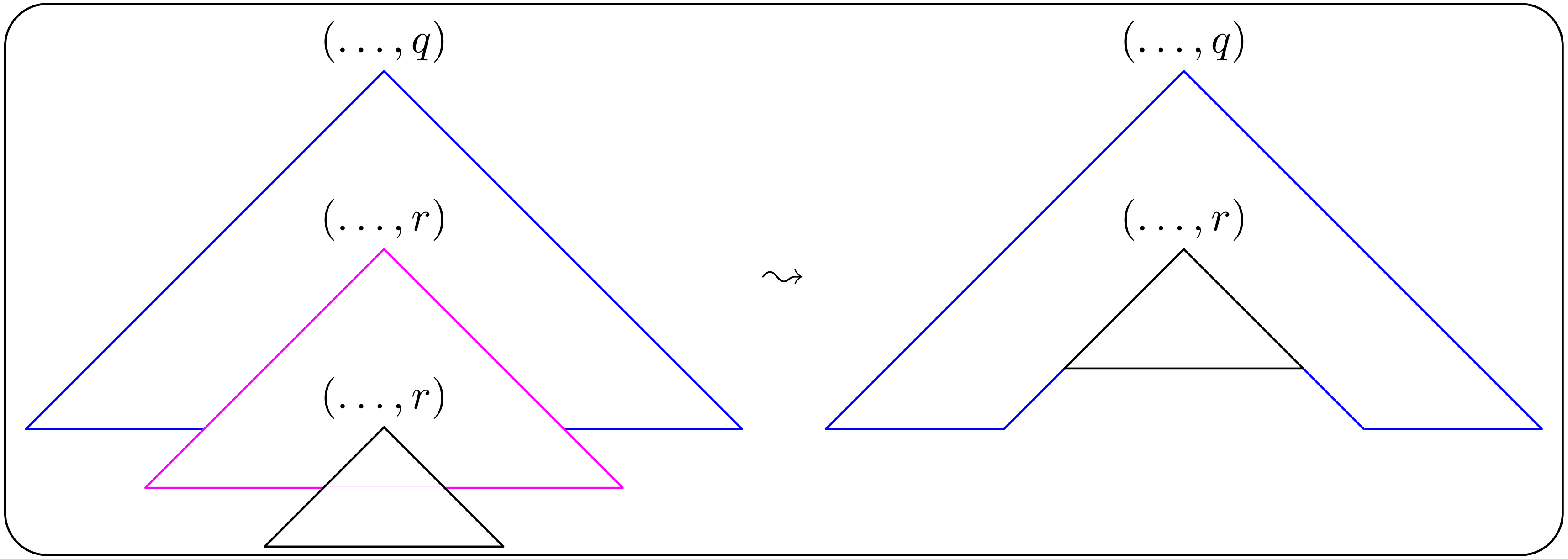
Proof sketch: Let $\psi \in \Psi^q$ be a minimal counterexample with respect to the cardinality of

$$W_\psi = \{\, w \in \mathrm{pos}(\psi) \mid 2 \cdot \mathrm{card}(Q) \leq |w| \,\}\,.$$

Clearly there is a path of length at least $2 \cdot \mathrm{card}(Q)$ in ψ. Consider the prefix of length $\mathrm{card}(Q)$. Now we distinguish two cases.

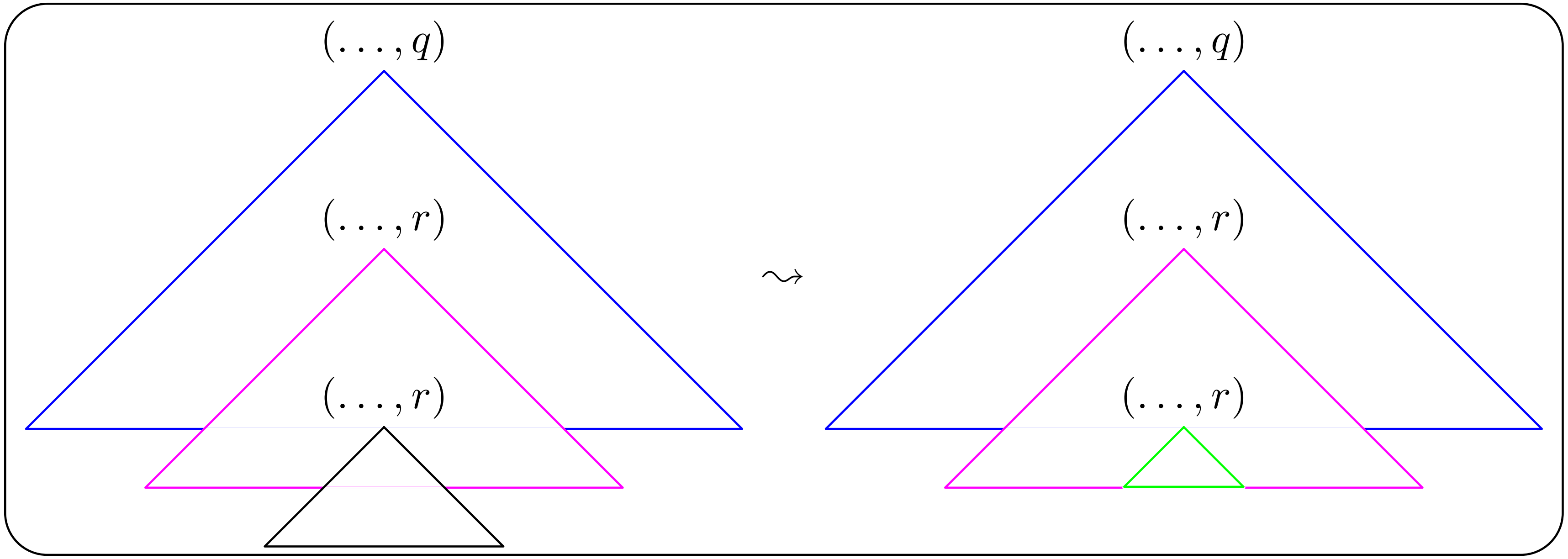
Cost-Finiteness – Step III (cont.)

Case 1: $c(\varphi) = x_1$ with $\varphi \in \widehat{\Psi}^r_r$. The following trees have equivalent costs, but the latter is smaller. Contradiction.



Cost-Finiteness – Step III (cont.)

Case 2: $c(\varphi) = a$ with $\varphi \in \widehat{\Psi}_r^r$. The following trees have equivalent costs, but the latter is smaller. The black subtree has height at least $\mathrm{card}(Q) + 1$ and is replaced by the green tree of height at most $\mathrm{card}(Q)$. Contradiction.



Decidability of Cost-Finiteness

Definition: Let M be a reduced tree automaton with cost function c over a finitely factorizing and monotonic semiring $\mathcal{A}$. Condition *(finite-trans)* holds, if for every $k \in \mathbb{N}_+$, $q \in Q \setminus Q_{\{\mathbf{0},\mathbf{1}\}}$, $i \in [k]$ with $q_i \equiv_M q$, and $\tau = (q_1 \ldots q_k, \sigma, q) \in \delta^{(k)}$ we have either

(i) $c(\tau) \equiv x_i + p$ for some $p \in A\langle X_k \setminus \{x_i\}\rangle$ and ($\mathcal{A}$ is additively idempotent or $p \equiv \mathbf{0}$) or

(ii) $x_i \notin \mathrm{var}(c(\tau))$.

Observation: Condition (finite-trans) is decidable.

Lemma: Condition (finite) and Condition (finite-trans) are equivalent.
Proof sketch:
(finite-trans) $\Rightarrow$ (finite): trivial
(finite-trans) $\Leftarrow$ (finite): apply the closed under decomposition property

Decidability of Cost-Finiteness (cont.)

Theorem: Let M be a tree automaton with cost function c over a monotonic and finitely factorizing semiring $\mathcal{A}$. The following statements are equivalent and decidable.

(i) M is cost-finite.

(ii) For every $q \in Q$ and $a \in c(M)_q$ the set $c(\widehat{\Psi}_q^q)[a]/_\equiv$ is finite.

(iii) Condition (finite) holds.

(iv) Condition (finite-trans) holds.

(v) If $\mathcal{A}$ is naturally ordered, then M is bounded with respect to the natural order $\sqsubseteq$.

Boundedness results

Example: Let M be a tree automaton with cost function c over the naturally ordered semiring $\mathcal{A}$. It is decidable whether M is bounded with respect to the natural order $\sqsubseteq$,

(i) if $\mathcal{A} = \mathrm{Nat}$ is the semiring of the non-negative integers,

(ii) if $\mathcal{A} = \mathrm{Arct}$ is the arctic semiring,

(iii) if $\mathcal{A} = \mathrm{Lang}_\Sigma$ is the finite-language semiring, or

(iv) if $\mathcal{A} = \mathcal{N}$ is the finite-subset (of the non-negative integers) semiring.

Remarks:

(a) The results (i) and (ii) were also obtained in [Sei94].

(b) For results (iii) and (iv) one shows cost-finiteness with the help of a partial order different from $\subseteq$ (both semirings are non-monotonic with respect to $\subseteq$).

(c) The semiring $\mathcal{N}$ was also considered in [Sei94], but with respect to a slightly different problem (cardinality of the elements bounded?).

Remaining Questions

(i) Which properties of monotonic semirings are obsolete when restricting ourselves to tree automata with linear cost functions (or cost functions of a particular type like $a \cdot x_1 \cdot \ldots \cdot x_k$)?

(ii) Can we characterize boundedness of tree automata with cost functions over certain semirings which are not finitely factorizing?

(iii) Can we establish sufficient or necessary criteria for boundedness/unboundedness with less restrictions on the semiring?

References

[Eil74] S. Eilenberg. *Automata, Languages, and Machines – Volume A*, volume 59 of Pure and Applied Mathematics. Academic Press, 1974.

[ES77] J. Engelfriet and E.M. Schmidt. *IO and OI. I*. Journal of Computer and System Sciences, 15(3):328–353, 1977.

[ES78] J. Engelfriet and E.M. Schmidt. *IO and OI. II*. Journal of Computer and System Sciences, 16(1):67–99, 1978.

[GS84] F. Gécseg and M. Steinby. *Tree Automata*. Akadémiai Kiadó, Budapest, 1984.

[HW98] U. Hebisch and H.J. Weinert. *Semirings – Algebraic Theory and Applications in Computer Science*. World Scientific, Singapore, 1998.

[Sei94] H. Seidl. *Finite tree automata with cost functions*. Theoretical Computer Science, 126(1):113-142, 1994.