Regular Cost Functions on Infinite Trees

Christof Löding

RWTH Aachen University

Workshop TCRF at Highlights 2018

Berlin, September 21, 2018

Outline

1. Introduction: Parity Index and Cost Functions

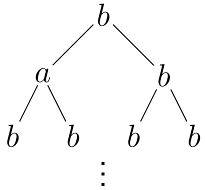
2. Regular Cost Functions over Infinite Words and Trees

3. Transforming between different automaton models

Parity Nondeterministic Tree Automata (PNTA)

- Automata $\mathcal{A} = (Q, \Sigma, q_0, \Delta, pri)$ on infinite binary trees
- Transitions of the form (q, a, q', q'')

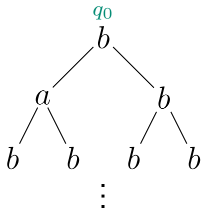
Run of $\mathcal{A}$:



Parity Nondeterministic Tree Automata (PNTA)

- Automata $\mathcal{A} = (Q, \Sigma, q_0, \Delta, pri)$ on infinite binary trees
- Transitions of the form (q, a, q', q'')

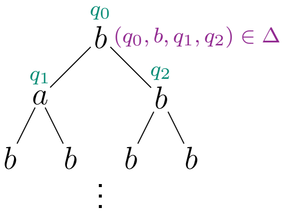
Run of $\mathcal{A}$:



Parity Nondeterministic Tree Automata (PNTA)

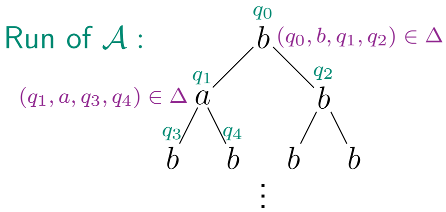
- Automata $\mathcal{A} = (Q, \Sigma, q_0, \Delta, pri)$ on infinite binary trees
- Transitions of the form (q, a, q', q'')

Run of $\mathcal{A}$:



Parity Nondeterministic Tree Automata (PNTA)

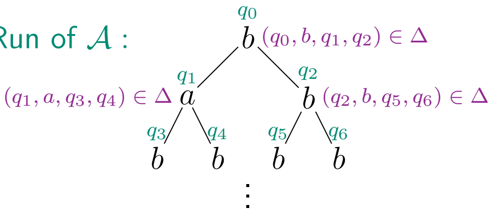
- Automata $\mathcal{A} = (Q, \Sigma, q_0, \Delta, pri)$ on infinite binary trees
- Transitions of the form (q, a, q', q'')



Parity Nondeterministic Tree Automata (PNTA)

- Automata $\mathcal{A} = (Q, \Sigma, q_0, \Delta, pri)$ on infinite binary trees
- Transitions of the form (q, a, q', q'')

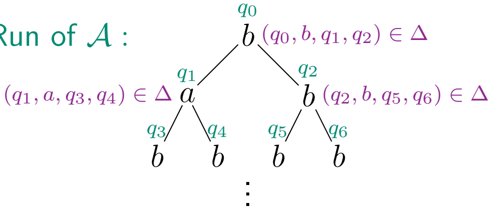
Run of $\mathcal{A}$:



Parity Nondeterministic Tree Automata (PNTA)

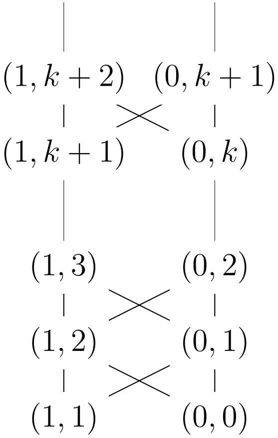
- Automata $\mathcal{A} = (Q, \Sigma, q_0, \Delta, pri)$ on infinite binary trees
- Transitions of the form (q, a, q', q'')

Run of $\mathcal{A}$:



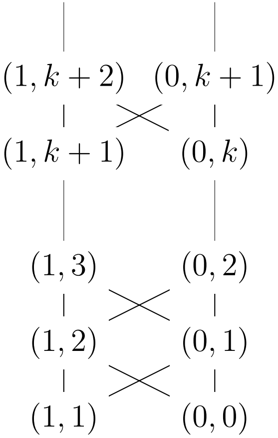
- Priority function $pri : Q \to \mathbb{N}$
- Run accepting if on each path the maximal priority appearing infinitely often is even.

Mostowski Hierarchy



The parity index of $\mathcal{A}$ is the pair (i, j) of highest and lowest priority it uses

Mostowski Hierarchy



The parity index of $\mathcal{A}$ is the pair (i, j) of highest and lowest priority it uses

Deciding the levels of the hierarchy:

Given: Language T and $i < j$
Question: Is there an automaton of index (i, j) for T?

An Approach for a Decision Procedure

Theorem [CL08a]. The nondeterministic parity index problem can be effectively reduced to a limitedness problem for regular cost functions over infinite trees (defined by B-parity tree automata).

Given: PNTA $\mathcal{A}$ and target interval $[i, j]$

An Approach for a Decision Procedure

Theorem [CL08a]. The nondeterministic parity index problem can be effectively reduced to a limitedness problem for regular cost functions over infinite trees (defined by B-parity tree automata).

Given: PNTA $\mathcal{A}$ and target interval $[i, j]$

$$\downarrow$$

B-PNTA $\mathcal{B}$ with priorities $[i, j]$ s.t.
function computed by $\mathcal{B}$ is bounded

$$\Leftrightarrow$$

$L(\mathcal{A})$ can be accepted by PNTA with priorities $[i, j]$

Outline

1 Introduction: Parity Index and Cost Functions

2 Regular Cost Functions over Infinite Words and Trees

3 Transforming between different automaton models

Costs, counters, B-, and S-semantics

- set $\Gamma = \{1, \ldots, k\}$ of counters
- set $Op = \{\mathtt{i}, \mathtt{r}, \mathtt{c}\}$ of operations increment, reset, and check
- sequence w of counter operations $\gamma : \Gamma \to Op^*$

  B-semantics:

  $cost^{\mathsf{B}}(w) = \sup\{$counter values on $\mathtt{c}$-positions in $w\}$

  S-semantics:

  $cost^{\mathsf{S}}(w) = \inf\{$counter values on $\mathtt{c}$-positions in $w\}$

Costs, counters, B-, and S-semantics

- set $\Gamma = \{1, \ldots, k\}$ of counters
- set $Op = \{\mathtt{i}, \mathtt{r}, \mathtt{c}\}$ of operations increment, reset, and check
- sequence w of counter operations $\gamma : \Gamma \to Op^*$

  B-semantics:

  $cost^{\mathsf{B}}(w) = \sup\{\text{counter values on } \mathtt{c}\text{-positions in } w\}$

  S-semantics:

  $cost^{\mathsf{S}}(w) = \inf\{\text{counter values on } \mathtt{c}\text{-positions in } w\}$
- Adding an acceptance condition, β infinite sequence of priorities:

  B-semantics:

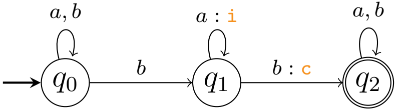
  $$cost^{\mathsf{B}}(\beta, w) = \begin{cases} \infty \text{ if } \beta \text{ does not satisfy the parity condition} \\ \sup\{\text{counter values on } \mathtt{c}\text{-positions}\} \text{ otherwise} \end{cases}$$

  S-semantics:

  $$cost^{\mathsf{S}}(\beta, w) = \begin{cases} 0 \text{ if } \beta \text{ does not satisfy the parity condition} \\ \inf\{\text{counter values on } \mathtt{c}\text{-positions}\} \text{ otherwise} \end{cases}$$

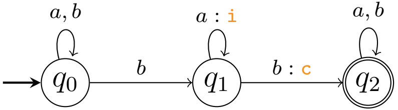
Cost functions on infinite words

Extend transitions of nondeterministic Büchi, parity, ... automata over infinite words by counter operations: (q, a, q', γ) with $\gamma : \Gamma \to Op^*$



Cost functions on infinite words

Extend transitions of nondeterministic Büchi, parity, ... automata over infinite words by counter operations: (q, a, q', γ) with $\gamma : \Gamma \to Op^*$



Cost of an infinite input word α is

$$\text{B-semantics: } \mathcal{A}^{\mathsf{B}}(\alpha) = \inf_{\rho} cost^{\mathsf{B}}(\rho)$$

$$\text{S-semantics: } \mathcal{A}^{\mathsf{S}}(\alpha) = \sup_{\rho} cost^{\mathsf{S}}(\rho)$$

(where ρ ranges over the runs on α)

Equivalence of cost functions

Let $f, g : D \to \mathbb{N} \cup \{\infty\}$ be two cost functions over the same domain.

Domination and equivalence of cost functions:

$$\begin{array}{lcl} f \preceq g & \Leftrightarrow & \forall X \subseteq D : \; g \text{ bounded on } X \; \to f \text{ bounded on } X \\ f \approx g & \Leftrightarrow & f \preceq g \text{ and } g \preceq f \end{array}$$

Example: $f, g : \{a, b\}^* \to \mathbb{N} \cup \{\infty\}$ with

$$\begin{array}{lcl} f(w) & = & 2 \cdot |w|_a \\ g(w) & = & |w| \end{array}$$

Then $f \preceq g$ but not $g \preceq f$ because f is bounded on b^* but g is not.

History determinism

In the classical theory of automata on infinite trees, deterministic ω-automata play an important role because they can be composed with tree automata and games.

Cost-ω-automata cannot be determinized, in general.

There is the weaker notion of history-determinism (or good-for-games) that still admits composition with tree automata and games.

A construction of history-deterministic cost-automata on finite words is given in [CF16] (see talk of N. Fijalkow).

Theorem (Colcombet, unpublished). For each nondeterministic B- or S-parity automaton over infinite words, one can construct an equivalent history-deterministic B- or S-parity automaton.

Cost-parity alternating tree automata (cost-PATA)

- transition function of the form $\delta(q,a) = \varphi$ where φ is a positive Boolean combination of atoms (d, q, γ) with

  direction $d \in \{0,1\}$, state q, and counter action $\gamma : \Gamma \to Op^*$

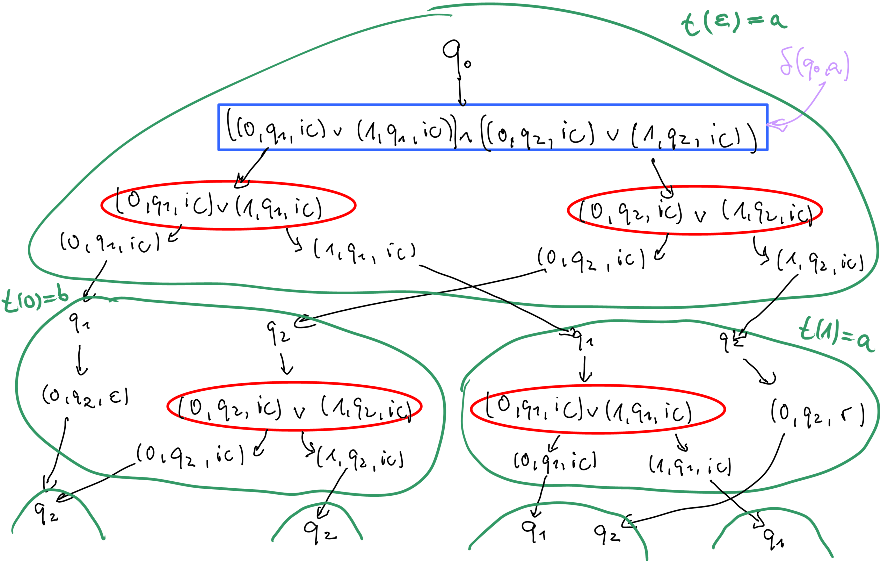
Cost-parity alternating tree automata (cost-PATA)

- transition function of the form $\delta(q,a) = \varphi$ where φ is a positive Boolean combination of atoms (d, q, γ) with

  direction $d \in \{0,1\}$, state q, and counter action $\gamma : \Gamma \to Op^*$

- semantics on a tree t via a game $\mathcal{G}_{\mathcal{A},t}$:
  - Positions of the form (q, u) with state q and tree node u.
  - Let $\delta(q, t(u)) = \varphi$.
  - Players Automaton and Pathfinder determine an atom of the formula where Automaton resolves disjunctions, and Pathfinder resolves conjunctions.
  - If an atom (d, q', γ) is reached, the game continues in (q', ud).
- We write σ_A, σ_P for strategies of Automaton, Pathfinder.

Illustration of $\mathcal{G}_{\mathcal{A},t}$



Semantics of cost-PATA

- A play α in $\mathcal{G}_{\mathcal{A},t}$ has a corresponding path, state sequence, and sequence of counter actions.

  B-semantics:

  $$cost^{\mathsf{B}}(\alpha) = \begin{cases} \infty \text{ if } \alpha \text{ does not satisfy the parity condition} \\ \sup\{\text{counter values on c-positions}\} \text{ otherwise} \end{cases}$$

  S-semantics:

  $$cost^{\mathsf{S}}(\alpha) = \begin{cases} 0 \text{ if } \alpha \text{ does not satisfy the parity condition} \\ \inf\{\text{counter values on c-positions}\} \text{ otherwise} \end{cases}$$

- B- and S-semantics $\mathcal{A}^{\mathsf{B}}, \mathcal{A}^{\mathsf{S}}$: trees $\to \mathbb{N} \cup \{\infty\}$

$$\begin{aligned} \mathcal{A}^{\mathsf{B}}(t) &= \inf_{\sigma_A} \sup_{\alpha \in \sigma_A} cost^{\mathsf{B}}(\alpha) \\ \mathcal{A}^{\mathsf{S}}(t) &= \sup_{\sigma_A} \inf_{\alpha \in \sigma_A} cost^{\mathsf{S}}(\alpha) \end{aligned}$$

Domination and limitedness

Domination problem:

Given two automata computing cost functions f_1 and f_2, decide if

$$f_1 \preceq f_2$$

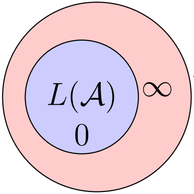
Domination and limitedness

Domination problem:

Given two automata computing cost functions f_1 and f_2, decide if

$$f_1 \preceq f_2$$

For example, the limitedness problem for a B-PATA $\mathcal{A}$ is the domination problem

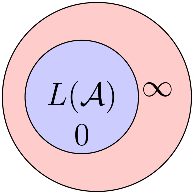
$\mathcal{A}^{\mathsf{B}} \preceq$  ?

Domination and limitedness

Domination problem:
Given two automata computing cost functions f_1 and f_2, decide if

$$f_1 \preceq f_2$$

For example, the limitedness problem for a B-PATA $\mathcal{A}$ is the domination problem

$\mathcal{A}^{\mathsf{B}} \preceq$  ?

Open question: Is the domination problem for cost-PATA decidable?

A first decidability result

Theorem ([CL10,VB11,VB12]). The domination problem $f_1 \preceq f_2$ is decidable if f_1 is given by an S-PNTA $\mathcal{A}_1$, and f_2 is given by a B-PNTA $\mathcal{A}_2$.

(The result in this form is stated in [VB11]. A detailed proof can be found [VB12]. The same technique is used in [CL10] for finite trees.)

A first decidability result

Theorem ([CL10,VB11,VB12]). The domination problem $f_1 \preceq f_2$ is decidable if f_1 is given by an S-PNTA $\mathcal{A}_1$, and f_2 is given by a B-PNTA $\mathcal{A}_2$.

(The result in this form is stated in [VB11]. A detailed proof can be found [VB12]. The same technique is used in [CL10] for finite trees.)

Idea: The situation is similar to checking $L_1 \subseteq L_2$ for standard regular tree languages if

- L_1 is given by a PNTA, and
- L_2 is given by a PNTA for the complement $\overline{L_2}$ of L_2.

$\rightsquigarrow$ intersection emptiness game for $L_1 \cap \overline{L_2}$ on the product automaton.

A first decidability result

Theorem ([CL10,VB11,VB12]). The domination problem $f_1 \preceq f_2$ is decidable if f_1 is given by an S-PNTA $\mathcal{A}_1$, and f_2 is given by a B-PNTA $\mathcal{A}_2$.

(The result in this form is stated in [VB11]. A detailed proof can be found [VB12]. The same technique is used in [CL10] for finite trees.)

Idea: The situation is similar to checking $L_1 \subseteq L_2$ for standard regular tree languages if

- L_1 is given by a PNTA, and
- L_2 is given by a PNTA for the complement $\overline{L_2}$ of L_2.

$\rightsquigarrow$ intersection emptiness game for $L_1 \cap \overline{L_2}$ on the product automaton.

Consequence: If we can transform arbitrary cost-PATAs into S-PNTAs and B-PNTAs, then we can solve the general domination problem.

Overview

decidable

S-PNTA $\leq$ B-PNTA

S-PATA

B-PATA

Outline

1 Introduction: Parity Index and Cost Functions

2 Regular Cost Functions over Infinite Words and Trees

3 Transforming between different automaton models

Transformations between alternating automata

For alternating automata, one can effectively translate between S- and B- automata.

Theorem ([CL10,VB11,VB12]). Given a B-, or S-PATA, one can effectively construct an S-, or B-PATA defining an equivalent cost function.

(The result in this form is stated in [VB11]. A detailed proof can be found [VB12]. The same technique is used in [CL10] for finite trees.)

Idea for B-PATA $\to$ S-PATA

$$\mathcal{A}^{\mathsf{B}}(t) = \inf_{\sigma_A} \underbrace{\sup_{\alpha \in \sigma_A} cost^{\mathsf{B}}(\alpha)}_{\text{value of } \sigma_A}$$

- For $N \in \mathbb{N}$ let $\mathcal{G}^N_{\mathcal{A},t}$ be the win-lose game that is won by Automaton if she has a strategy of value $\leq N$.
- By determinacy of $\mathcal{G}^N_{\mathcal{A},t}$: $\mathcal{A}^{\mathsf{B}}(t) = \sup_{\sigma_P} \inf_{\alpha \in \sigma_P} cost^{\mathsf{B}}(\alpha)$

Idea for B-PATA $\to$ S-PATA

$$\mathcal{A}^{\mathsf{B}}(t) = \inf_{\sigma_A} \underbrace{\sup_{\alpha \in \sigma_A} cost^{\mathsf{B}}(\alpha)}_{\text{value of } \sigma_A}$$

- For $N \in \mathbb{N}$ let $\mathcal{G}^N_{\mathcal{A},t}$ be the win-lose game that is won by Automaton if she has a strategy of value $\leq N$.
- By determinacy of $\mathcal{G}^N_{\mathcal{A},t}$: $\mathcal{A}^{\mathsf{B}}(t) = \sup_{\sigma_P} \inf_{\alpha \in \sigma_P} cost^{\mathsf{B}}(\alpha)$
- Obtain $\widetilde{\mathcal{A}}$ from $\mathcal{A}$ by exchanging $\wedge$ and $\vee$ in the transition function $\rightsquigarrow$ strategies σ_P of Pathfinder in $\mathcal{G}_{A,t}$ become strategies $\widetilde{\sigma}_A$ of Automaton in $\mathcal{G}_{\widetilde{\mathcal{A}},t}$, and thus $\mathcal{A}^{\mathsf{B}}(t) = \sup_{\widetilde{\sigma}_A} \inf_{\alpha \in \widetilde{\sigma}_A} cost^{\mathsf{B}}(\alpha)$

Idea for B-PATA $\to$ S-PATA

$$\mathcal{A}^{\mathsf{B}}(t) = \inf_{\sigma_A} \underbrace{\sup_{\alpha \in \sigma_A} cost^{\mathsf{B}}(\alpha)}_{\text{value of } \sigma_A}$$

- For $N \in \mathbb{N}$ let $\mathcal{G}^N_{\mathcal{A},t}$ be the win-lose game that is won by Automaton if she has a strategy of value $\leq N$.
- By determinacy of $\mathcal{G}^N_{\mathcal{A},t}$: $\mathcal{A}^{\mathsf{B}}(t) = \sup_{\sigma_P} \inf_{\alpha \in \sigma_P} cost^{\mathsf{B}}(\alpha)$
- Obtain $\widetilde{\mathcal{A}}$ from $\mathcal{A}$ by exchanging $\wedge$ and $\vee$ in the transition function $\leadsto$ strategies σ_P of Pathfinder in $\mathcal{G}_{A,t}$ become strategies $\widetilde{\sigma}_A$ of Automaton in $\mathcal{G}_{\widetilde{\mathcal{A}},t}$, and thus
  $\mathcal{A}^{\mathsf{B}}(t) = \sup_{\widetilde{\sigma}_A} \inf_{\alpha \in \widetilde{\sigma}_A} cost^{\mathsf{B}}(\alpha)$

  This is almost the S-semantics of $\widetilde{\mathcal{A}}$, except for $cost^{\mathsf{B}}(\alpha)$.
- Compose $\widetilde{\mathcal{A}}$ with a history deterministic ω-automaton that computes $cost^{\mathsf{B}}(\alpha)$ using S-semantics.

Overview

decidable

S-PNTA $\leq$ B-PNTA

S-PATA $\longleftrightarrow$ B-PATA

dual $\wedge \leftrightarrow \vee$
+ hd-automata for
$cost^B(\alpha)$ / $cost^S(\alpha)$

From alternating to nondeterministic B-automata

Semantics of a B-PATA: $\mathcal{A}^{\mathsf{B}}(t) = \inf_{\sigma_A} \sup_{\alpha} cost^{\mathsf{B}}(\alpha)$

From alternating to nondeterministic B-automata

Semantics of a B-PATA: $\mathcal{A}^{\mathsf{B}}(t) = \inf_{\sigma_A} \sup_{\alpha} cost^{\mathsf{B}}(\alpha)$

If we restrict player Automaton to finite memory strategies $\sigma_{A,m}$ with memory size m, we obtain the cost function

$$\mathcal{A}^{\mathsf{B}}_{A,m}(t) = \inf_{\sigma_{A,m}} \sup_{\alpha} cost^{\mathsf{B}}(\alpha)$$

From alternating to nondeterministic B-automata

Semantics of a B-PATA: $\mathcal{A}^{\mathsf{B}}(t) = \inf_{\sigma_A} \sup_{\alpha} cost^{\mathsf{B}}(\alpha)$

If we restrict player Automaton to finite memory strategies $\sigma_{A,m}$ with memory size m, we obtain the cost function

$$\mathcal{A}^{\mathsf{B}}_{A,m}(t) = \inf_{\sigma_{A,m}} \sup_{\alpha} cost^{\mathsf{B}}(\alpha)$$

Theorem. For a given B-PATA $\mathcal{A}$, and a given memory size m, one can construct a B-PNTA for the cost function $\mathcal{A}^{\mathsf{B}}_{A,m}$.

Proof: Same technique as for de-alternation of standard PATAs: Write strategy on the tree. Build B-PNTA computing value of tree with strategy annotation, using a history deterministic ω-automaton running along the path. Project away strategy annotation. $\square$

From alternating to nondeterministic B-automata

Semantics of a B-PATA: $\mathcal{A}^{\mathsf{B}}(t) = \inf_{\sigma_A} \sup_{\alpha} cost^{\mathsf{B}}(\alpha)$

If we restrict player Automaton to finite memory strategies $\sigma_{A,m}$ with memory size m, we obtain the cost function

$$\mathcal{A}^{\mathsf{B}}_{A,m}(t) = \inf_{\sigma_{A,m}} \sup_{\alpha} cost^{\mathsf{B}}(\alpha)$$

Theorem. For a given B-PATA $\mathcal{A}$, and a given memory size m, one can construct a B-PNTA for the cost function $\mathcal{A}^{\mathsf{B}}_{A,m}$.

Proof: Same technique as for de-alternation of standard PATAs: Write strategy on the tree. Build B-PNTA computing value of tree with strategy annotation, using a history deterministic ω-automaton running along the path. Project away strategy annotation. □

(This technique is used for finite trees in [CL10] and for weak automata on infinite trees in [VB11]/[VB12].)

Overview

decidable

S-PNTA $\leq$ B-PNTA

if fin. mem. strat. for Automaton in $\mathcal{G}_{A,t}$

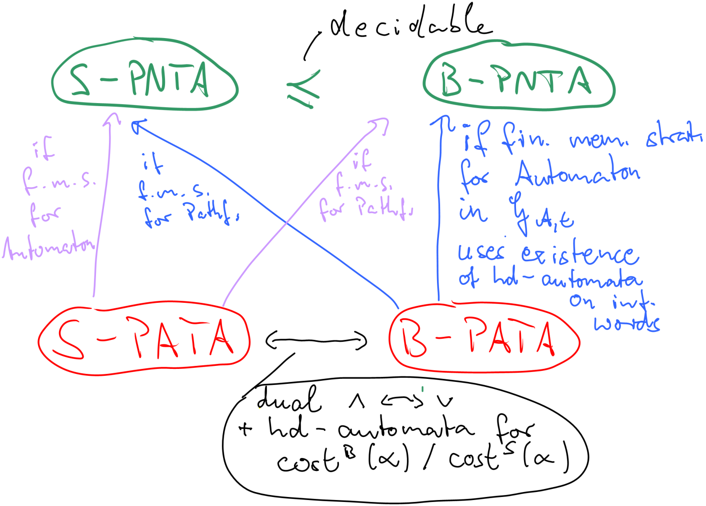
uses existence of hd-automata on inf. words

S-PATA $\longleftrightarrow$ B-PATA

dual $\wedge \leftrightarrow \vee$
+ hd-automata for $cost^B(\alpha)$ / $cost^S(\alpha)$

Overview

The same technique works for the other conversions:



Known finite memory results

Original conjecture: Games with B-parity conditions admit finite memory strategies on arbitrary arenas, with a memory size only depending on the number of counters and priorities.

Has been falsified [FHKS15] (see the talk of M. Skrzypczak)

Known finite memory results

Original conjecture: Games with B-parity conditions admit finite memory strategies on arbitrary arenas, with a memory size only depending on the number of counters and priorities.

Has been falsified [FHKS15] (see the talk of M. Skrzypczak)

Positive finite memory results:

Player I = Automaton

Player II = Pathfinder

player	counter		condition	arena	reference
I/II	arbitrary	B	safety	finite acyclic	[CL08b]

Known finite memory results

Original conjecture: Games with B-parity conditions admit finite memory strategies on arbitrary arenas, with a memory size only depending on the number of counters and priorities.

Has been falsified [FHKS15] (see the talk of M. Skrzypczak)

Positive finite memory results:

Player I = Automaton
Player II = Pathfinder

player	counter		condition	arena	reference
I/II	arbitrary	B	safety	finite acyclic	[CL08b]
I/II	1, no reset	B	parity	arbitrary	[Col13]/[VB12]
I	1	B	parity	chronological	[CKL10]/[VB12]

Known finite memory results

Original conjecture: Games with B-parity conditions admit finite memory strategies on arbitrary arenas, with a memory size only depending on the number of counters and priorities.

Has been falsified [FHKS15] (see the talk of M. Skrzypczak)

Positive finite memory results:
Player I = Automaton
Player II = Pathfinder

player	counter		condition	arena	reference
I/II	arbitrary	B	safety	finite acyclic	[CL08b]
I/II	1, no reset	B	parity	arbitrary	[Col13]/[VB12]
I	1	B	parity	chronological	[CKL10]/[VB12]
I	arbitrary	B	Büchi	chronological	[VB12]
II	arbitrary	B	co-Büchi	chronological	[VB12]

Known finite memory results

Original conjecture: Games with B-parity conditions admit finite memory strategies on arbitrary arenas, with a memory size only depending on the number of counters and priorities.

Has been falsified [FHKS15] (see the talk of M. Skrzypczak)

Positive finite memory results:

Player I = Automaton
Player II = Pathfinder

player	counter		condition	arena	reference
I/II	arbitrary	B	safety	finite acyclic	[CL08b]
I/II	1, no reset	B	parity	arbitrary	[Col13]/[VB12]
I	1	B	parity	chronological	[CKL10]/[VB12]
I	arbitrary	B	Büchi	chronological	[VB12]
II	arbitrary	B	co-Büchi	chronological	[VB12]
I	arbitrary	B	parity	thin tree	[FHKS15]

Implications for decidability

Relying on the finite memory results from [VB12], the following theorem is stated in [CKLVB13]:

Theorem. The domination problem $f_1 \preceq f_2$ is decidable if f_1 is given by a B-coBüchi ATA, and f_2 by a B-Büchi ATA.

Implications for decidability

Relying on the finite memory results from [VB12], the following theorem is stated in [CKLVB13]:

Theorem. The domination problem $f_1 \preceq f_2$ is decidable if f_1 is given by a B-coBüchi ATA, and f_2 by a B-Büchi ATA.

Weak automata are a special case of Büchi and co-Büchi automata:

Theorem [VB11]. The domination problem is decidable for cost functions defined by weak alternating tree automata, or equivalently by weak monadic second-order cost logic (cost WMSO).

Implications for decidability

Relying on the finite memory results from [VB12], the following theorem is stated in [CKLVB13]:

Theorem. The domination problem $f_1 \preceq f_2$ is decidable if f_1 is given by a B-coBüchi ATA, and f_2 by a B-Büchi ATA.

Weak automata are a special case of Büchi and co-Büchi automata:

Theorem [VB11]. The domination problem is decidable for cost functions defined by weak alternating tree automata, or equivalently by weak monadic second-order cost logic (cost WMSO).

The result has been strengthened to an extension of cost WMSO with an additional fixpoint operation [BCKPVB14]. The proof uses quasi weak alternating automata [KVB11] and a transformation of two-way to one-way automata.

Conclusion

- Research on boundedness problems for tree automata, originally motivated by the parity index problem.
- Many interesting techniques have been developed in analogy to the standard theory of automata on infinite trees:
  - Composition of cost-ω-automata with games/tree automata: history-determinism
  - B-PATA $\leftrightarrow$ S-PATA: like complementation of alternating tree automata by dualization
  - cost-PATA $\rightarrow$ cost-PNTA: same proof strategy as in classical theory, requires finite memory determinacy
- Partial results, like decidability of domination for weak alternating cost automata and weak cost-MSO.
- The general problem of decidability of domination remains open.

References

[CL08a] Th. Colcombet, C. Löding.
The non-deterministic Mostowski hierarchy and distance-parity automata. ICALP 2008

[CL08b] Th. Colcombet, C. Löding.
The nesting-depth of disjunctive mu-calculus for tree languages and the limitedness problem. CSL 2008

[CL10] Th. Colcombet, C. Löding.
Theory of Regular Cost Functions over Finite Trees. LICS 2010

[VB11] M. Vanden Boom.
Weak Cost Monadic Logic Over Infinite Trees. MFCS 2011

[KVB11] D. Kuperberg, M. Vanden Boom.
Quasi-Weak Cost Automata: A New Variant of Weakness. FSTTCS 2011

[VB12] M. Vanden Boom.
Weak Cost Automata over Infinite Trees. PhD Thesis, 2012

[CKLVB13] Th. Colcombet, D. Kuperberg, C. Löding., M. Vanden Boom.
Deciding the weak definability of Büchi definable tree languages. CSL 2013

[Col13] Th. Colcombet.
Fonctions régulières de coût. Habilitation Thesis, 2013

[BCKPVB14] A. Blumensath, Th. Colcombet, D. Kuperberg, P. Parys, M. Vanden Boom.
Two-Way Cost Automata and Cost Logics over Infinite Trees. LICS 2014.

[FHKS15] N. Fijalkow, F. Horn, D. Kuperberg, M. Skrzypczak.
Trading Bounds for Memory in Games with Counters. ICALP 2015

[CF16] Th. Colcombet, N. Fijalkow.
The Bridge Between Regular Cost Functions and Omega-Regular Languages. ICALP 2016.