Two-way cost automata and cost logics over infinite trees

Achim Blumensath[1], Thomas Colcombet[2],
Denis Kuperberg[3], Pawel Parys[3], and **Michael Vanden Boom**[4]

[1]TU Darmstadt, [2]Université Paris Diderot,
[3]University of Warsaw, [4]University of Oxford

CSL-LICS 2014
Vienna, Austria

Boundedness questions

Finite power property [Simon '78, Hashiguchi '79]

given regular language L of finite words,
is there $n \in \mathbb{N}$ such that $L^* = \{\epsilon\} \cup L^1 \cup L^2 \cup \cdots \cup L^n$?

Star-height problem [Hashiguchi '88, Kirsten '05]

given regular language L of finite words and $n \in \mathbb{N}$,
is there a regular expression for L with at most n nestings of Kleene star?

Fixpoint closure boundedness [Blumensath+Otto+Weyer '09]

given an MSO formula $\varphi(x, X)$ positive in X,
is there $n \in \mathbb{N}$ such that the least fixpoint of φ over finite words
is always reached within n iterations?

Boundedness questions

The **theory of regular cost functions** is an extension of the theory of regular languages that can be used to solve these boundedness questions in a uniform way.

Boundedness questions

The **theory of regular cost functions** is an extension of the theory of regular languages that can be used to solve these boundedness questions in a uniform way.

Boundedness problem

Instance: function $f : \mathcal{D} \to \mathbb{N} \cup \{\infty\}$

($\mathcal{D}$ is set of words or trees over some fixed finite alphabet $\mathbb{A}$)

Question: Is there $n \in \mathbb{N}$ such that for all structures $s \in \mathcal{D}$, $f(s) \leq n$?

Cost functions over finite words [Colcombet'09]

Regular Cost Functions

nondeterministic cost automata
cost MSO
BS expressions
stabilization monoids

Boundedness decidable
[Colcombet'09, Bojańczyk+Colcombet'06]

Cost functions over finite words

Cost monadic second-order logic (CMSO)

Atomic formulas: $a(x)$ $x \in X$ $\underbrace{|X| \leq N}_{\text{must occur positively}}$

Constructors: $\underbrace{\wedge, \vee, \neg}_{\text{Boolean connectives}}$ $\underbrace{\exists x}_{\text{first-order quantification}}$ $\underbrace{\exists X}_{\text{monadic second-order quantification}}$

Cost functions over finite words

Cost monadic second-order logic (CMSO)

Atomic formulas: $a(x)$ $\quad x \in X \quad$ $\underbrace{|X| \leq N}_{\text{must occur positively}}$

Constructors: $\underbrace{\wedge, \vee, \neg}_{\text{Boolean connectives}}$ $\quad \underbrace{\exists x}_{\text{first-order quantification}}$ $\quad \underbrace{\exists X}_{\text{monadic second-order quantification}}$

Semantics $[\![\varphi]\!] : \mathbb{A}^* \to \mathbb{N} \cup \{\infty\}$

$[\![\varphi]\!](u) := \inf \{n : u \vDash \varphi[n/N]\}$

Cost functions over finite words

Cost monadic second-order logic (CMSO)

Atomic formulas: $a(x)$ $x \in X$ $\underbrace{|X| \leq N}_{\text{must occur positively}}$

Constructors: $\underbrace{\wedge, \vee, \neg}_{\text{Boolean connectives}}$ $\underbrace{\exists x}_{\text{first-order quantification}}$ $\underbrace{\exists X}_{\text{monadic second-order quantification}}$

Semantics $[\![\varphi]\!] : \mathbb{A}^* \to \mathbb{N} \cup \{\infty\}$

$[\![\varphi]\!](u) := \inf \{n : u \vDash \varphi[n/N]\}$

Example

If φ is in MSO, then $[\![\varphi]\!](u) := \begin{cases} 0 & \text{if } u \vDash \varphi \\ \infty & \text{otherwise} \end{cases}$

Cost monadic second-order logic (CMSO)

Atomic formulas: $a(x)$ $x \in X$ $\underbrace{|X| \leq N}_{\text{must occur positively}}$

Constructors: $\underbrace{\wedge, \vee, \neg}_{\text{Boolean connectives}}$ $\underbrace{\exists x}_{\text{first-order quantification}}$ $\underbrace{\exists X}_{\text{monadic second-order quantification}}$

Semantics $[\![\varphi]\!] : \mathbb{A}^* \to \mathbb{N} \cup \{\infty\}$

$$[\![\varphi]\!](u) := \inf \{n : u \vDash \varphi[n/N]\}$$

Example

Maximum length of a block of a's

$$\varphi := \forall X \left((\texttt{block}(X) \wedge \forall x (x \in X \to a(x)) \to |X| \leq N \right)$$

Cost functions over finite words [Colcombet'09]

Regular Cost Functions

nondeterministic cost automata
cost MSO
BS expressions
stabilization monoids

Boundedness decidable
[Colcombet'09, Bojańczyk+Colcombet'06]

Cost functions over finite words [Colcombet'09]

Regular Cost Functions

nondeterministic cost automata
cost MSO
BS expressions
stabilization monoids

Boundedness decidable
[Colcombet'09, Bojańczyk+Colcombet'06]

language universality, inclusion, and emptiness decidable

Cost functions over finite words [Colcombet'09]

Regular Cost Functions

nondeterministic cost automata
cost MSO
BS expressions
stabilization monoids

Boundedness decidable
[Colcombet'09, Bojańczyk+Colcombet'06]

language universality, inclusion, and emptiness decidable

finite power property, star height problem, fixpoint closure boundedness, ... decidable

Theory of regular cost functions

The theory of regular cost functions is a robust decidable extension of the theory of regular languages over:

✓ **finite words** [Colcombet '09, Bojanczyk+Colcombet '06]

✓ **infinite words** [Kuperberg+VB'12, Colcombet unpublished]

✓ **finite trees** [Colcombet+Löding '10]

Theory of regular cost functions

The theory of regular cost functions is a robust decidable extension of the theory of regular languages over:

✓ **finite words** [Colcombet '09, Bojanczyk+Colcombet '06]

✓ **infinite words** [Kuperberg+VB'12, Colcombet unpublished]

✓ **finite trees** [Colcombet+Löding '10]

? **infinite trees**

Motivating open problem

Mostowski index problem

Instance: regular language L of infinite trees, and set $\{i, i+1, \ldots, j\}$

Question: Is there a nondeterministic parity automaton $\mathcal{A}$ using only priorities $\{i, i+1, \ldots, j\}$ such that $L = L(\mathcal{A})$?

Motivating open problem

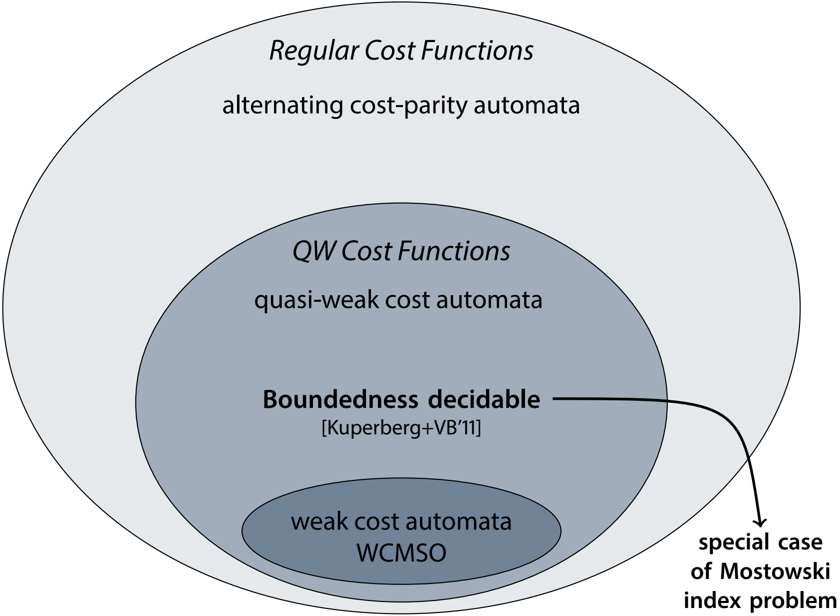
Mostowski index problem

Instance: regular language L of infinite trees, and set $\{i, i+1, \ldots, j\}$

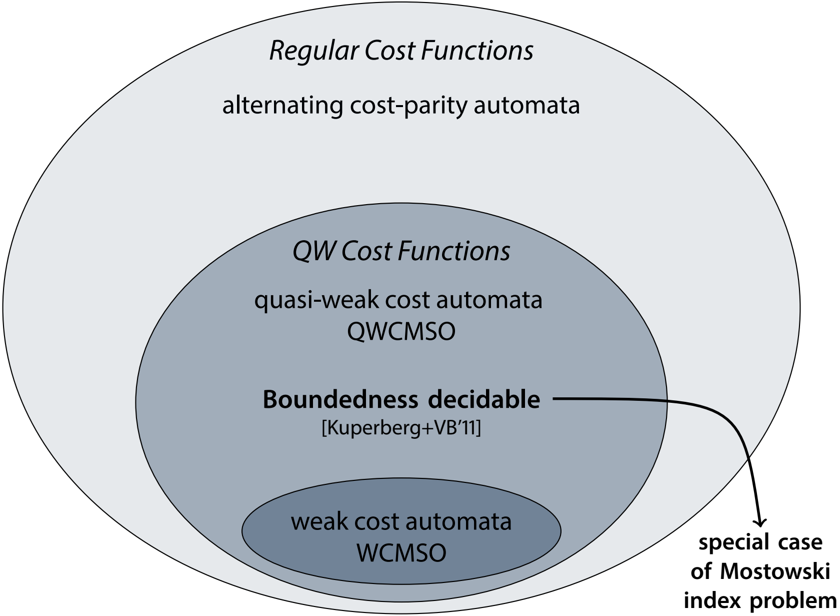
Question: Is there a nondeterministic parity automaton $\mathcal{A}$ using only priorities $\{i, i+1, \ldots, j\}$ such that $L = L(\mathcal{A})$?

Reduced to deciding boundedness for certain cost functions over infinite trees [Colcombet+Löding '08]

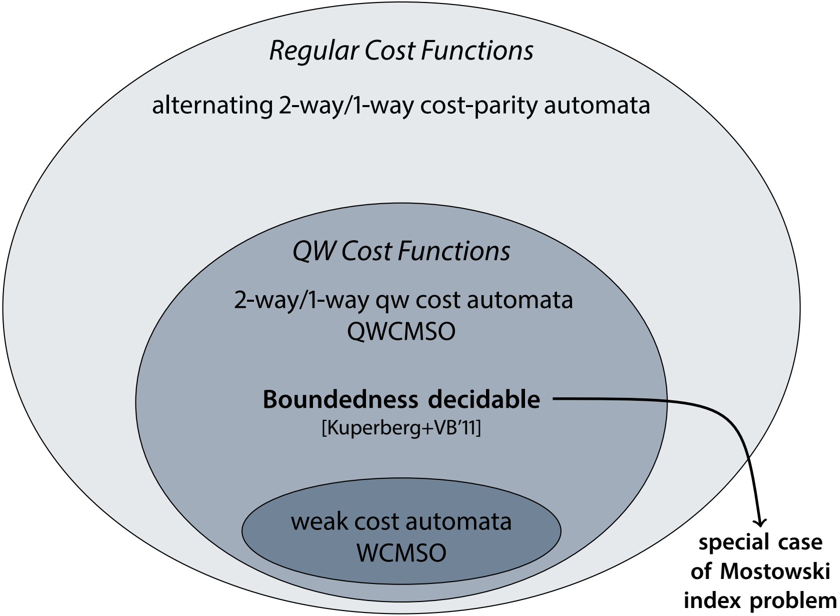
Cost functions over infinite trees



Cost functions over infinite trees



Cost functions over infinite trees



Cost parity automata on infinite trees

$$\mathcal{A} = \langle \mathbb{A}, Q, q_0, \delta, \Omega \rangle$$

δ describes possible moves
for Eve and Adam,
and associated counter actions
(increment, reset, leave unchanged)

$\Omega : Q \to P$
for a finite set of
priorities P

n-acceptance game $\mathcal{A} \times t$

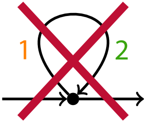
- Positions in the game are $Q \times \mathrm{dom}(t)$.
- Eve and Adam select the next position in the play based on δ.
- Eve is trying to ensure the play has counter value at most n and the maximum priority occurring infinitely often in the play is even.

Semantics

$[\![\mathcal{A}]\!](t) := \inf \{n : \text{Eve wins the } n\text{-acceptance game } \mathcal{A} \times t\}$

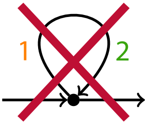
Weak cost automaton

alternating cost-parity automaton such that
no cycle visits both even and odd priorities



Weak cost automaton

alternating cost-parity automaton such that
no cycle visits both even and odd priorities



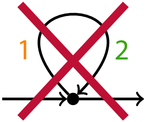
Weak cost monadic second-order logic (WCMSO)

Syntax like CMSO, but interpret second-order quantification over finite sets

Weak cost automata and logic over infinite trees

Weak cost automaton

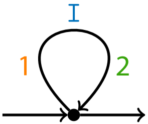
alternating cost-parity automaton such that
no cycle visits both even and odd priorities



Weak cost monadic second-order logic (WCMSO)

Syntax like CMSO, but interpret second-order quantification over finite sets

Quasi-weak cost automaton

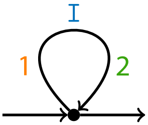


alternating cost-parity automaton such that
in any cycle with both even and odd priorities,
there is a counter which is incremented but not reset

Quasi-weak cost automata and logic over infinite trees

Quasi-weak cost automaton

alternating cost-parity automaton such that
in any cycle with both even and odd priorities,
there is a counter which is incremented but not reset



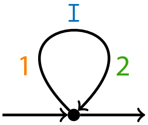
Quasi-weak cost monadic second-order logic (QWCMSO)

Add **bounded expansion operator** to WCMSO:

$$z \in \mu^N Y. \{x : \varphi(x, Y)\}$$

where Y occurs positively in $\varphi(x, Y)$,
and this operator occurs positively in the enclosing formula.

Quasi-weak cost automaton



alternating cost-parity automaton such that
in any cycle with both even and odd priorities,
there is a counter which is incremented but not reset

Quasi-weak cost monadic second-order logic (QWCMSO)

Add **bounded expansion operator** to WCMSO:

$$z \in \mu^N Y. \{x : \varphi(x, Y)\}$$

where Y occurs positively in $\varphi(x, Y)$,
and this operator occurs positively in the enclosing formula.

Example

Maximal size of block of a's on a branch starting at the root:
$\exists w[\mathtt{root}(w) \wedge w \in \mu^N X.\{x : \exists yz[b(x,y,z) \vee (a(x,y,z) \wedge y \in X \wedge z \in X)]\}]$

Bounded expansion operator and 2-way automata

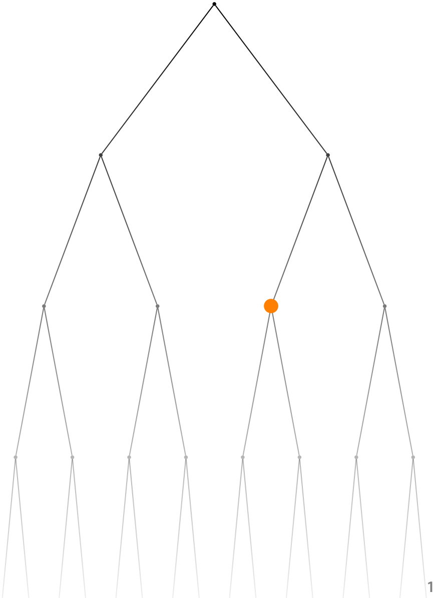
Game for testing
$z \in \mu^N Y.\{x : \varphi(x, Y)\}$ for $n \in \mathbb{N}$.

Initial position $x := z$.

Game from position x:

- Eve chooses set Y such that $\varphi(x, Y)$ holds (if it is not possible, she loses).
- Adam chooses some new $y \in Y$ (if it is not possible, he loses).
- Game continues in next round with $x := y$

If the game exceeds n rounds, Adam wins.



Bounded expansion operator and 2-way automata

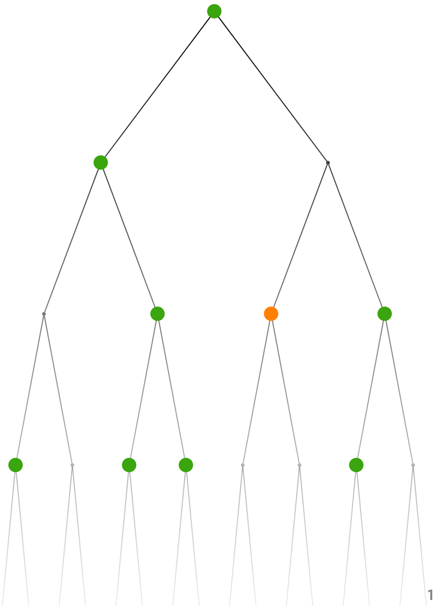
Game for testing
$z \in \mu^N Y.\{x \,:\, \varphi(x, Y)\}$ for $n \in \mathbb{N}$.

Initial position $x := z$.

Game from position x:

- **Eve chooses** set Y such that $\varphi(x, Y)$ holds (if it is not possible, she loses).
- Adam chooses some new $y \in Y$ (if it is not possible, he loses).
- Game continues in next round with $x := y$

If the game exceeds n rounds, Adam wins.



Bounded expansion operator and 2-way automata

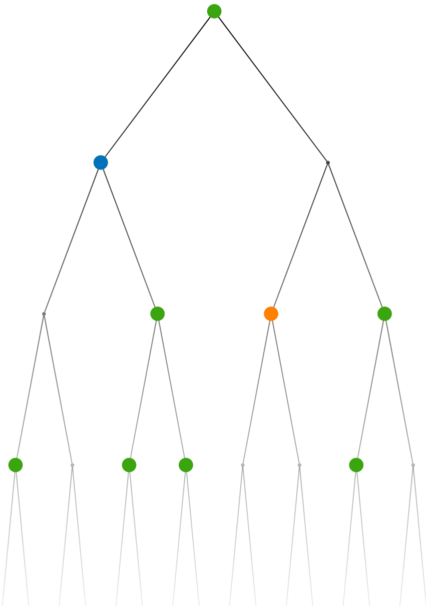
Game for testing
$z \in \mu^N Y.\{x \,:\, \varphi(x, Y)\}$ for $n \in \mathbb{N}$.

Initial position $x := z$.

Game from position x:

- Eve chooses set Y such that $\varphi(x, Y)$ holds (if it is not possible, she loses).
- **Adam chooses** some new $y \in Y$ (if it is not possible, he loses).
- Game continues in next round with $x := y$

If the game exceeds n rounds, Adam wins.



Bounded expansion operator and 2-way automata

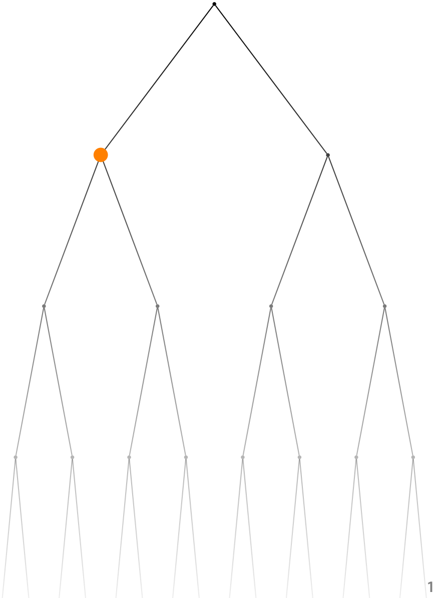
Game for testing
$z \in \mu^N Y.\{x \,:\, \varphi(x, Y)\}$ for $n \in \mathbb{N}$.

Initial position $x := z$.

Game from position x:

- Eve chooses set Y such that $\varphi(x, Y)$ holds (if it is not possible, she loses).
- Adam chooses some new $y \in Y$ (if it is not possible, he loses).
- Game continues in **next round** with $x := y$

If the game exceeds n rounds, Adam wins.



Bounded expansion operator and 2-way automata

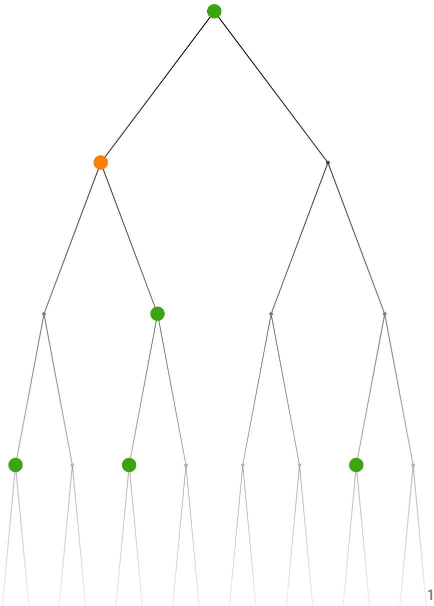
Game for testing
$z \in \mu^N Y.\{x \ : \ \varphi(x, Y)\}$ for $n \in \mathbb{N}$.

Initial position $x := z$.

Game from position x:

- **Eve chooses** set Y such that $\varphi(x, Y)$ holds (if it is not possible, she loses).
- Adam chooses some new $y \in Y$ (if it is not possible, he loses).
- Game continues in next round with $x := y$

If the game exceeds n rounds, Adam wins.



Bounded expansion operator and 2-way automata

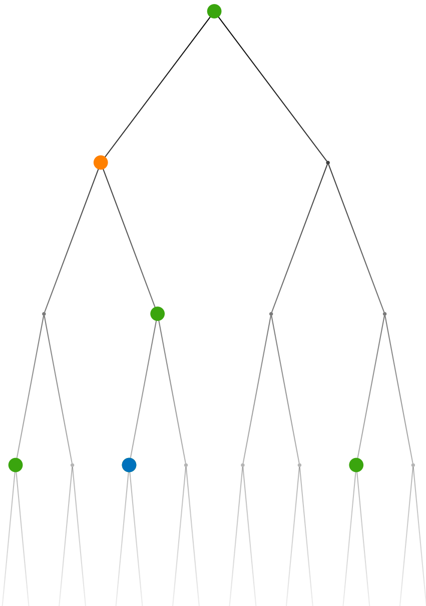
Game for testing
$z \in \mu^N Y.\{x \ : \ \varphi(x, Y)\}$ for $n \in \mathbb{N}$.

Initial position $x := z$.

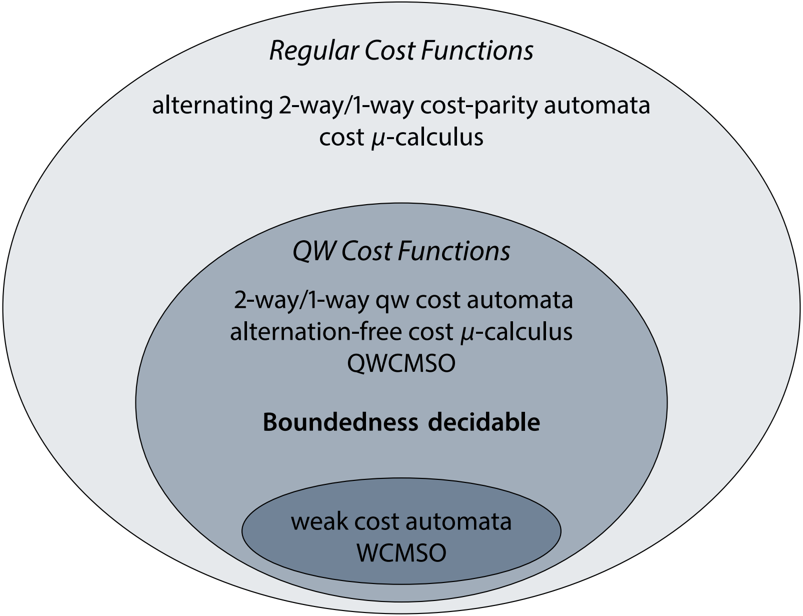
Game from position x:

- Eve chooses set Y such that $\varphi(x, Y)$ holds (if it is not possible, she loses).
- **Adam chooses** some new $y \in Y$ (if it is not possible, he loses).
- Game continues in next round with $x := y$

If the game exceeds n rounds, Adam wins.



Summary


Regular Cost Functions
alternating 2-way/1-way cost-parity automata
cost μ-calculus
QW Cost Functions
2-way/1-way qw cost automata
alternation-free cost μ-calculus
QWCMSO
Boundedness decidable
weak cost automata
WCMSO