Games for the existence of bounds

Thomas Colcombet

Friday, January 22nd 2010
Journées Nationales d'Informatique Mathématique 2010

1. Tree automata and games
2. Some boundedness questions
3. Toward a generic framework
4. Regular Cost functions
5. Conclusion

Finite tree automata

We work with rooted finite binary tree labelled by $\mathbb{A}$ (root ε, left child $x0$, right child $x1$). A language is a set of trees.

A tree automaton is a finite device which test the existence of a labelling of the tree by finite information satisfying some local constraints.

Finite tree automata

We work with rooted finite binary tree labelled by $\mathbb{A}$ (root ε, left child $x0$, right child $x1$). A language is a set of trees.

A tree automaton is a finite device which test the existence of a labelling of the tree by finite information satisfying some local constraints.

E.g., Trees such that ‘all occurrences of ‘a’ occur on the same branch’:
Guess a set of positions B such that:

- for $x \in B$ non-leaf, exactly one child of x belongs to B,
- for $x \in B$ non-root, the parent of x belongs to B,
- all occurrences of a belong to B.

Finite tree automata

We work with rooted finite binary tree labelled by $\mathbb{A}$ (root ε, left child $x0$, right child $x1$). A language is a set of trees.

A tree automaton is a finite device which test the existence of a labelling of the tree by finite information satisfying some local constraints.

E.g., Trees such that ‘all occurrences of ‘a’ occur on the same branch’:
Guess a set of positions B such that:

- for $x \in B$ non-leaf, exactly one child of x belongs to B,
- for $x \in B$ non-root, the parent of x belongs to B,
- all occurrences of a belong to B.

Formally, one guesses a run r, labeling nodes by states in Q such that $r(\varepsilon) \in \mathit{Init}$, $(r(x), t(x)) \in \mathit{Fin}$ if x is a leaf, and $(r(x), t(x), r(x0), r(x1)) \in \Delta$ if x is not a leaf.

Theorem (Thatcher and Wright68)

Tree languages accepted by tree automata are closed under union, intersection, complementation, projection. Emptiness is decidable.

Link between automata and games (Gurevich&Harrington)

Given $\mathcal{A}$ and t, the acceptance game $G(\mathcal{A}, t)$ is:

- two opponent players Inside/Outside
- at the begining of the game, Inside chooses a state $q \in Init$, and we proceed to position (ε, q),
- when in position (x, q), for x not a leaf,
  - Inside chooses a transition of the form $(q, t(x), q_0, q_1)$,
  - Outide chooses to proceed either in $(x0, q_0)$ (left) or $(x1, q_1)$ (right)
- when in position (x, q), for x a leaf, Inside chooses $(q, t(x)) \in Fin$, and wins.
- if Inside can't make a move, Outside wins.

Link between automata and games (Gurevich&Harrington)

Given $\mathcal{A}$ and t, the acceptance game $G(\mathcal{A}, t)$ is:

- two opponent players Inside/Outside
- at the begining of the game, Inside chooses a state $q \in Init$, and we proceed to position (ε, q),
- when in position (x, q), for x not a leaf,
  - Inside chooses a transition of the form $(q, t(x), q_0, q_1)$,
  - Outide chooses to proceed either in $(x0, q_0)$ (left) or $(x1, q_1)$ (right)
- when in position (x, q), for x a leaf, Inside chooses $(q, t(x)) \in Fin$, and wins.
- if Inside can't make a move, Outside wins.

Fact

Inside wins the game $G(\mathcal{A}, t)$ iff $\mathcal{A}$ accepts t.

Sketch: One identifies a run with a winning strategy for Inside

Sketch of a game theoretic proof of complementation

Problem of complement

$t \in L$ iff "there exists a run"

yield by complement "for all run"

but must be accepted by "there exists a run."

Complementation is a problem of quantification

Sketch of a game theoretic proof of complementation

$t \notin L(\mathcal{A})$

iff all strategies of Inside in $G(\mathcal{A}, t)$ are loosing

Sketch of a game theoretic proof of complementation

$t \notin L(\mathcal{A})$

iff all strategies of Inside in $G(\mathcal{A}, t)$ are loosing

Determinacy

In a game of finite duration, one of the players has a winning strategy.

iff there exists a winning strategy of Outside in $G(\mathcal{A}, t)$

Sketch of a game theoretic proof of complementation

$t \notin L(\mathcal{A})$

iff all strategies of Inside in $G(\mathcal{A}, t)$ are loosing

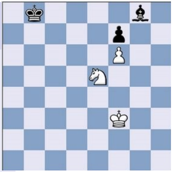
Determinacy

In a game of finite duration, one of the players has a winning strategy.

iff there exists a winning strategy of Outside in $G(\mathcal{A}, t)$

Memoryless strategies

In a reachability game, the winner has a memoryless strategy: he can decide its moves independently of the prefix of the play.



Remark

A game for which the objective for a player is to 'see as many events A events B' does not have memoryless strategies.

Sketch of a game theoretic proof of complementation

$t \notin L(\mathcal{A})$

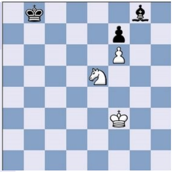
iff all strategies of Inside in $G(\mathcal{A}, t)$ are loosing

Determinacy

In a game of finite duration, one of the players has a winning strategy.

iff there exists a winning strategy of Outside in $G(\mathcal{A}, t)$

Memoryless strategies



In a reachability game, the winner has a memoryless strategy: he can decide its moves independently of the prefix of the play.

iff there exists $\sigma : (x, (q, a, q_0, q_1)) \to \{0, 1\}$ describing a winning strategy for Outside: this labels nodes by finite information

Sketch of a game theoretic proof of complementation

$t \notin L(\mathcal{A})$

iff all strategies of Inside in $G(\mathcal{A}, t)$ are loosing

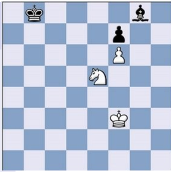
Determinacy

In a game of finite duration, one of the players has a winning strategy.

iff there exists a winning strategy of Outside in $G(\mathcal{A}, t)$

Memoryless strategies

In a reachability game, the winner has a memoryless strategy: he can decide its moves independently of the prefix of the play.



iff there exists $\sigma : (x, (q, a, q_0, q_1)) \to \{0, 1\}$ describing a winning strategy for Outside: this labels nodes by finite information

iff t is accepted by a suitable automaton, guessing σ and a 'proof' that σ is winning for Outside

1. Tree automata and games
2. Some boundedness questions
3. Toward a generic framework
4. Regular Cost functions
5. Conclusion

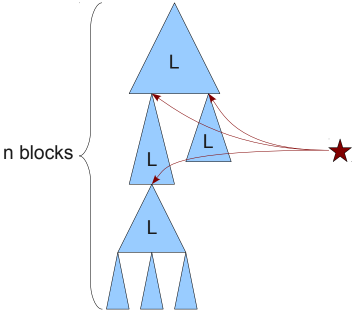
Finite power universality problem

Given a regular tree language L with possibly $\star$ labelling leaves. $L \cdot_\star K$ is obtained by substituting a tree in K to each occurrence of $\star$ in a tree of L. $L^n = L \cdot_\star \cdots \cdot_\star L$.

Problem (variant of the finite power property)

Input: L with $\star \in L$. Is there n such that L^n contains all trees?

I.e., there is n such that every tree can be decomposed as:



Finite power universality problem

Given a regular tree language L with possibly $\star$ labelling leaves. $L \cdot_\star K$ is obtained by substituting a tree in K to each occurrence of $\star$ in a tree of L. $L^n = L \cdot_\star \cdots \cdot_\star L$.

Problem (variant of the finite power property)

Input: L with $\star \in L$. Is there n such that L^n contains all trees?

Theorem (Hashiguchi, Simon)

The finite power property over words is decidable.

Game for the finite power universality

Given an automaton $\mathcal{A}$ accepting L and a tree t. The finite power game $FP(\mathcal{A}, t)$ is as follows.

- two opponent players Inside/Outside
- Inside chooses a state $q \in Init$, yielding to position (ε, q),
- when in position (x, q), for x not a leaf,
  - if $(q, \star) \in Fin$, Inside can choose to go to (x, p) for $p \in Init$, and pays 1 coin,
  - Inside chooses a transition of the form $(q, t(x), q_0, q_1)$,
  - Outide chooses to proceed either in $(x0, q_0)$ (left) or $(x1, q_1)$ (right)
- when in position (x, q), for x a leaf, Inside chooses $(q, t(x)) \in Fin$, and wins.

Game for the finite power universality

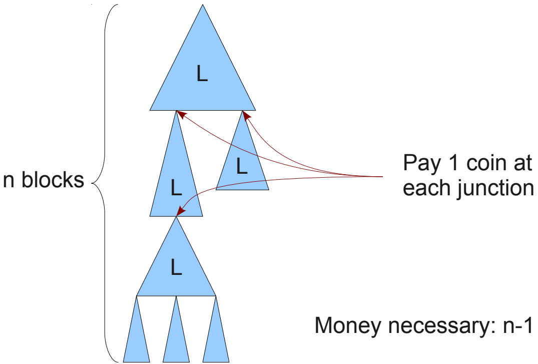
Given an automaton $\mathcal{A}$ accepting L and a tree t. The finite power game $FP(\mathcal{A}, t)$ is as follows.

- two opponent players Inside/Outside
- Inside chooses a state $q \in Init$, yielding to position (ε, q),
- when in position (x, q), for x not a leaf,
  - if $(q, \star) \in Fin$, Inside can choose to go to (x, p) for $p \in Init$, and pays 1 coin,
  - Inside chooses a transition of the form $(q, t(x), q_0, q_1)$,
  - Outide chooses to proceed either in $(x0, q_0)$ (left) or $(x1, q_1)$ (right)
- when in position (x, q), for x a leaf, Inside chooses $(q, t(x)) \in Fin$, and wins.

Fact

Inside wins the game $FP(\mathcal{A}, t)$ with $n - 1$ coins, iff $t \in L^n$.

Game for the finite power universality



Fact

Inside wins the game $FP(\mathcal{A}, t)$ with $n-1$ coins, iff $t \in L^n$.

Decidability of the finite power universality

Theorem

The finite power universality problem is decidable.

Sketch: L does not have the finite power universality property iff for all n, there is a tree t such that every strategy for Inside is loosing in $FP(\mathcal{A},t)$ with n coins

Lemma (Memoryless)

Distance games have memoryless winning strategies.

iff for all n, there is a tree t, and there is a memoryless strategy for Outside winning in $FP(\mathcal{A},t)$ against n coins

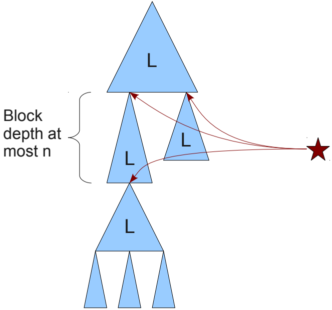
The later problem is logically simpler, and can be solved using algebraic methods (stabilisation monoids).

Game for the finite subset universality

Problem (finite subset universality)

Input L. Decide if K^* contains all trees for some finite $K \subseteq L$.
Equivalently, there exists n such that all trees belong to K_n^*?
With $K_n = \{t \in L \ : \ height(t) \leq n\}$.

I.e., there is n such that every tree can be decomposed as:



Game for the finite subset universality

The finite subset game $FS(\mathcal{A}, t)$ is as follows:

- two opponent players Inside/Outside, one starts with n coins
- Inside chooses a state $q \in Init$, yielding to position (ε, q),
- when in position (x, q), for x not a leaf,
  - if $(q, \star) \in Fin$, Inside can choose to go to (x, p) and get its n coins back
  - pay 1 coin, and Inside chooses a transition of the form $(q, t(x), q_0, q_1)$,
  - Outide chooses to proceed either in $(x0, q_0)$ (left) or $(x1, q_1)$ (right)
- when in position (x, q), for x a leaf, Inside chooses $(q, t(x)) \in Fin$, and wins.

Game for the finite subset universality

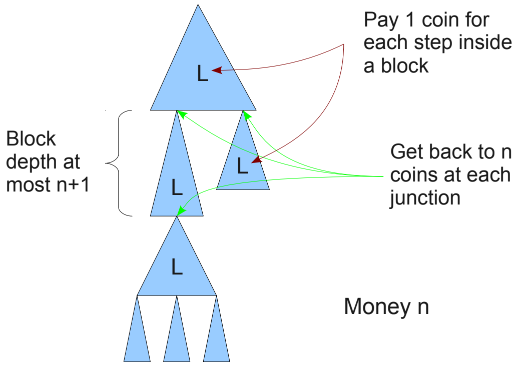
The finite subset game $FS(\mathcal{A}, t)$ is as follows:

- two opponent players Inside/Outside, one starts with n coins
- Inside chooses a state $q \in Init$, yielding to position (ε, q),
- when in position (x, q), for x not a leaf,
  - if $(q, \star) \in Fin$, Inside can choose to go to (x, p) and get its n coins back
  - pay 1 coin, and Inside chooses a transition of the form $(q, t(x), q_0, q_1)$,
  - Outide chooses to proceed either in $(x0, q_0)$ (left) or $(x1, q_1)$ (right)
- when in position (x, q), for x a leaf, Inside chooses $(q, t(x)) \in Fin$, and wins.

Fact

Inside wins the game $FS(\mathcal{A}, t)$ with n coins, iff $t \in K^*$ for $K = \{t \in L \ : \ \textit{height}(t) \leq n + 1\}$.

Game for the finite subset universality



Fact

Inside wins the game $FS(\mathcal{A}, t)$ with n coins, iff $t \in K^*$ for $K = \{t \in L \ : \ \text{height}(t) \leq n+1\}$.

Decidability of the finite subset universality

Theorem

The finite subset universality problem is decidable.

Sketch: L does not have the finite power property
iff for all n, there is a tree t such that every strategy for Inside is loosing in $FS(\mathcal{A}, t)$ with n coins

Lemma (Memoryless)

Desert games have memoryless winning strategies.

iff for all n, there is a tree t, and there is a memoryless strategy for Outside winning in $FS(\mathcal{A}, t)$ against n coins

The later problem is logically simpler, and can be solved using algebraic methods (stabilisation monoids).

Other problems, same approach

Problem (Eggan63)

The star height problem, asks whether a language can be described by a regular expressions with at most k nesting of Kleene stars.

Theorem (C., Löding 08)

The star height problem is decidable for trees (previously known for words [Hashiguchi88,Kirsten05]).

Problem

The boundedness problem asks whether the fix-point of a logic formula can be reached in a bounded number of steps.

Theorem (Blumensath, Otto, Weyer09)

The boundedness problem for monadic second-order logic is decidable over the class of finite words. (Trees to come.)

1 Tree automata and games

2 Some boundedness questions

3 Toward a generic framework

4 Regular Cost functions

5 Conclusion

Principle

Our objective is to propose:

- a strict extension of the notion of regularity, in which languages are replaced by functions from trees to $\mathbb{N} \cup \{\infty\}$,
- in which strong properties remain valid, such as the equivalence with monadic logic,
- for which the existence of bounds is decidable.

Consider $fp(t) = \inf\{n \ : \ t \in L^n\}$
Then $L(\mathcal{A})$ has the finite power universality property iff fp is bounded.

Consider $fs(t) = \inf\{n \ : \ t \in K^*,\ K \subseteq L,\ depth(K) \leq n\}$
Then $L(\mathcal{A})$ has the finite subset universality property iff fs is bounded.

Monadic (second-order) logic

The monadic logic the following syntax:

$$\begin{aligned}\Phi ::= \exists x.\Phi \quad &| \quad \exists X.\Phi \\ | \quad \Phi \vee \Phi \quad &| \quad \neg\Phi \\ | \quad R(x_1, \ldots, x_n) \quad &| \quad x \in X\end{aligned}$$

where x ranges over elements, X over sets (monadic variable).

Example (reachability using edges ending in X)

$$\texttt{reach}(x, y, Z) =^{\mathsf{def}} \forall W. \\ \big[x \in W \wedge \forall z, z'.\ \big((z \in W \wedge z' \in Z \wedge \mathrm{edge}(z, z')) \to z' \in W\big)\big] \\ \to \quad y \in W$$

Equivalence with regular languages

Theorem (Büchi60&62,Thatcher and Wright68, Rabin69)

It is equivalent for a language of finite words, of words of length ω, of trees, of infinite trees, to be:

1. *regular (i.e., defined by non-deterministic automata),*
2. *definable in monadic logic (using order, and letter predicates).*

From 1 to 2. use existential monadic quantifications for guessing a run, and check that the run is valid and accepting.

Equivalence with regular languages

Theorem (Büchi60&62,Thatcher and Wright68, Rabin69)

It is equivalent for a language of finite words, of words of length ω, of trees, of infinite trees, to be:

1. *regular (i.e., defined by non-deterministic automata),*
2. *definable in monadic logic (using order, and letter predicates).*

From 1 to 2. use existential monadic quantifications for guessing a run, and check that the run is valid and accepting.
From 2 to 1. Given Φ, consider the language $L_\Phi = \{u \ : \ u \models \Phi\}$. One proves by induction on Φ that L_Φ is regular:

Equivalence with regular languages

Theorem (Büchi60&62,Thatcher and Wright68, Rabin69)

It is equivalent for a language of finite words, of words of length ω, of trees, of infinite trees, to be:

1. *regular (i.e., defined by non-deterministic automata),*
2. *definable in monadic logic (using order, and letter predicates).*

From 1 to 2. use existential monadic quantifications for guessing a run, and check that the run is valid and accepting.
From 2 to 1. Given Φ, consider the language $L_\Phi = \{u \ : \ u \models \Phi\}$.
One proves by induction on Φ that L_Φ is regular:
$L_{\Phi\vee\Psi} = \{u \ : \ u \models \Phi \ or \ u \models \Psi\} = L_\Phi \cup L_\Psi$ (closure union)

Equivalence with regular languages

Theorem (Büchi60&62,Thatcher and Wright68, Rabin69)

It is equivalent for a language of finite words, of words of length ω, of trees, of infinite trees, to be:

1. *regular (i.e., defined by non-deterministic automata),*
2. *definable in monadic logic (using order, and letter predicates).*

From 1 to 2. use existential monadic quantifications for guessing a run, and check that the run is valid and accepting.
From 2 to 1. Given Φ, consider the language $L_\Phi = \{u \ : \ u \models \Phi\}$.
One proves by induction on Φ that L_Φ is regular:
$L_{\Phi\vee\Psi} = \{u \ : \ u \models \Phi \textit{ or } u \models \Psi\} = L_\Phi \cup L_\Psi$ (closure union)
$L_{\neg\Phi} = \{u \ : \ u \models \neg\Phi\} = \complement(L_\Phi)$ (closure complement)

Equivalence with regular languages

Theorem (Büchi60&62, Thatcher and Wright68, Rabin69)

It is equivalent for a language of finite words, of words of length ω, of trees, of infinite trees, to be:

1. *regular (i.e., defined by non-deterministic automata),*
2. *definable in monadic logic (using order, and letter predicates).*

From 1 to 2. use existential monadic quantifications for guessing a run, and check that the run is valid and accepting.

From 2 to 1. Given Φ, consider the language $L_\Phi = \{u : u \models \Phi\}$.

One proves by induction on Φ that L_Φ is regular:

$L_{\Phi\vee\Psi} = \{u : u \models \Phi \textit{ or } u \models \Psi\} = L_\Phi \cup L_\Psi$ (closure union)

$L_{\neg\Phi} = \{u : u \models \neg\Phi\} = \complement(L_\Phi)$ (closure complement)

$L_{\exists X.\Phi}$ ... (closure projection)

Equivalence with regular languages

Theorem (Büchi60&62,Thatcher and Wright68, Rabin69)

It is equivalent for a language of finite words, of words of length ω, of trees, of infinite trees, to be:

1. *regular (i.e., defined by non-deterministic automata),*
2. *definable in monadic logic (using order, and letter predicates).*

From 1 to 2. use existential monadic quantifications for guessing a run, and check that the run is valid and accepting.
From 2 to 1. Given Φ, consider the language $L_\Phi = \{u : u \models \Phi\}$.
One proves by induction on Φ that L_Φ is regular:
$L_{\Phi\vee\Psi} = \{u : u \models \Phi \textit{ or } u \models \Psi\} = L_\Phi \cup L_\Psi$ (closure union)
$L_{\neg\Phi} = \{u : u \models \neg\Phi\} = \complement(L_\Phi)$ (closure complement)
$L_{\exists X.\Phi}$ ... (closure projection)
etc... Consequence of the closure properties of regular languages.

Equivalence with regular languages

Theorem (Büchi60&62,Thatcher and Wright68, Rabin69)

It is equivalent for a language of finite words, of words of length ω, of trees, of infinite trees, to be:

1. *regular (i.e., defined by non-deterministic automata),*
2. *definable in monadic logic (using order, and letter predicates).*

Corollary

The problem of satisfiability of monadic formula over the class of words/ω-words/tree/infinite trees is decidable.

Cost monadic logic

Monadic logic

$$\begin{aligned}\Phi ::= \exists x.\Phi \quad &| \quad \exists X.\Phi \\ | \quad \Phi \vee \Phi \quad &| \quad \neg\Phi \\ | \quad R(x_1, \dots, x_n) \quad &| \quad x \in X\end{aligned}$$

Cost monadic logic

Monadic logic, negation pushed to the leaves

$$\begin{aligned}\Phi ::= \exists x.\Phi \quad &| \quad \exists X.\Phi \quad | \quad \forall x.\Phi \quad | \quad \forall X.\Phi \\ | \quad \Phi \vee \Phi \quad &| \quad \cancel{\neg\Phi} \quad | \quad \Phi \wedge \Phi \\ | \quad R(x_1, \ldots, x_n) \quad &| \quad x \in X \quad | \quad \neg R(x_1, \ldots, x_n) \quad | \quad x \notin X\end{aligned}$$

Cost monadic logic

Monadic logic, negation pushed to the leaves, with predicate $|X| \leq n$

$$\begin{aligned}\Phi ::= \exists x.\Phi \quad | \quad \exists X.\Phi \quad | \quad \forall x.\Phi \quad | \quad \forall X.\Phi \\ | \quad \Phi \vee \Phi \quad | \quad \Phi \wedge \Phi \\ | \quad R(x_1,\dots,x_n) \quad | \quad x \in X \quad | \quad \neg R(x_1,\dots,x_n) \quad | \quad x \notin X \\ | \quad |X| \leq N\end{aligned}$$

in which N is a new variable ranging over non-negative integers.

Cost monadic logic

Monadic logic, negation pushed to the leaves, with predicate $|X| \leq n$

$$\begin{aligned}\Phi ::= \exists x.\Phi \quad | \quad \exists X.\Phi \quad | \quad \forall x.\Phi \quad | \quad \forall X.\Phi \\ | \quad \Phi \vee \Phi \quad | \quad \Phi \wedge \Phi \\ | \quad R(x_1,\dots,x_n) \quad | \quad x \in X \quad | \quad \neg R(x_1,\dots,x_n) \quad | \quad x \notin X \\ | \quad |X| \leq N\end{aligned}$$

in which N is a new variable ranging over non-negative integers.

Remark that if $k \leq l$ then $\mathcal{S}, N = k \models \Phi$ implies $\mathcal{S}, N = l \models \Phi$, one sets:

$$[\![\Phi]\!](\mathcal{S}) = \inf\{n \;:\; \mathcal{S}, N = n \models \Phi\} \in \mathbb{N} \cup \{\infty\}\ .$$

Example of cost monadic logic over graphs

$$[\![\Phi]\!](\mathcal{S}) = \inf\{n \ : \ \mathcal{S}, N = n \models \Phi\} \in \mathbb{N} \cup \{\infty\}$$

Example

$$\texttt{diameter} ::= \forall x, y. \exists Z. |Z| \leq N \wedge \texttt{reach}(x, y, Z).$$

Given a (directed) graph G,

$[\![\texttt{diameter}]\!](G) =$ diameter of G, ∞ if not strongly connected.

Remark

If Φ is a monadic sentence, then

$$[\![\Phi]\!](\mathcal{S}) = \begin{cases} 0 & \textit{if } \mathcal{S} \models \Phi \\ \infty & \textit{otherwise.} \end{cases}$$

Examples over words

Number of occurrences of letter a:

$\exists Z.|Z| \leq N \wedge \forall x.a(x) \rightarrow x \in Z$

Minimal distance between two b's $(+1)$:

$$
\begin{aligned}
\mathtt{interval}(Z) &= \forall x \in Z.\forall z \in Z.(x \leq y \wedge y \leq z) \rightarrow y \in Z \\
\mathtt{dist-b} &= \exists Z.\mathtt{interval}(Z) \wedge |Z| \leq N \\
&\quad \wedge \exists x \in Z.\exists y \in Z.\ \ x < y \wedge b(x) \wedge b(y)
\end{aligned}
$$

Length of maximal segment of consecutive a's:

$\mathtt{max-seg} \;\; = \forall Z.|Z| \leq N \wedge \mathtt{interval}(Z) \wedge \forall x.a(x) \rightarrow x \in Z$

Remark

Every rational power series over the tropical semiring is definable in cost monadic logic.
The functions fp and fs are definable.

On the limits of decidability over words

Remark

Given Φ, n, it is decidable whether $[\![\Phi]\!](u) = n$ for some u.

Theorem (Consequence of Krob92)

The equivalence between cost monadic definable functions over words is undecidable.

Problem (boundedness)

Given Φ, decide the existence of n, such that $[\![\Phi]\!](u) \leq n$ for all u.

1 Tree automata and games

2 Some boundedness questions

3 Toward a generic framework

4 Regular Cost functions

5 Conclusion

Cost functions

Idea: consider functions modulo an equivalence preserving bounds.

Definition

Given $f, g : E \to \mathbb{N} \cup \{\infty\}$,

$f \preccurlyeq g$ if for all $X \subseteq E$, $g|_X$ bounded implies $f|_X$ bounded.

$f \approx g$ if $f \preccurlyeq g$ and $g \preccurlyeq f$

A cost function over E is an equivalence class for $\approx$.

The objective is to circumvent Krob's undecidability result by reasoning modulo $\approx$.

The existence of bounds is preserved (f is bounded iff $f \preccurlyeq 0$).

Cost functions

Idea: consider functions modulo an equivalence preserving bounds.

Definition

Given $f, g : E \to \mathbb{N} \cup \{\infty\}$,

$f \preccurlyeq g$ if for all $X \subseteq E$, $g|_X$ bounded implies $f|_X$ bounded.

$f \approx g$ if $f \preccurlyeq g$ and $g \preccurlyeq f$

A cost function over E is an equivalence class for $\approx$.

Lemma

It is equivalent for two functions $f, g : E \to \mathbb{N} \cup \{\infty\}$:

- $f \preceq g$,
- $\forall m. \exists n. \forall x \in E.\ g(x) \leq m \to f(x) \leq n$,
- $f \leq \alpha \circ g$ *for some* $\alpha : \mathbb{N} \to \mathbb{N}$ *extended with* $\alpha(\infty) = \infty$.

Examples of cost functions

All bounded functions are equivalent.

$|\cdot|_a$ is incomparable with $|\cdot|_b$: $|\cdot|_a$ bounded over b^*, $|\cdot|_b$ isn't.

Over binary trees $\mathrm{height} \approx \mathrm{size}$ since:

$$\mathrm{height}(t) \leq \mathrm{size}(t) \qquad \text{and} \qquad \mathrm{size}(t) \leq 2^{\mathrm{height}(t)}$$

For Φ, Ψ monadic, $[\![\Phi]\!] \preccurlyeq [\![\Psi]\!]$ iff $\Psi \Rightarrow \Phi$ over trees.

Key results

One defines cost automata accepting functions such that:

Theorem (unpublished, with Löding)

It is equivalent for a cost function over words/finite trees, to be:

- *definable in cost monadic logic,*
- *accepted by cost automata,*

Such cost functions are called regular ([Col09] for words).

Ideas: cost automata have counting features (multiple counters, that can be incremented and reset).

One proves that cost automata have suitable closure properties (min, max, etc...) modulo $\approx$.

This requires the use of games and memoryless strategies, as well as deterministic-like automata over words, algebra, etc...

Key results

One defines cost automata accepting functions such that:

Theorem (unpublished, with Löding)

It is equivalent for a cost function over words/finite trees, to be:

- *definable in cost monadic logic,*
- *accepted by cost automata,*

Such cost functions are called regular ([Col09] for words).

Theorem (unpublished, with Löding)

The relation $\preccurlyeq$ is decidable over regular cost functions of finite trees ([Col09] for words).

Corollary

The relation $\preccurlyeq$ is decidable for cost monadic logic definable functions, and hence boundedness also.

1 Tree automata and games

2 Some boundedness questions

3 Toward a generic framework

4 Regular Cost functions

5 Conclusion

Conclusion and perspectives

The theory of regular cost functions is a strong extension of the notion of regular languages to a quantitative setting.

Games play an important role in the proofs in the tree case.

Main open question

Infinite trees?

The main obstacle is a question of existence of memoryless strategies in some games...

Solving this question would also solve the deep open problem of deciding the Mostowski hierarchy [C., Löding08].