Deciding the weak definability of Büchi definable tree languages

Michael Vanden Boom

Department of Computer Science
University of Oxford

Queen Mary
Theory Group Seminar

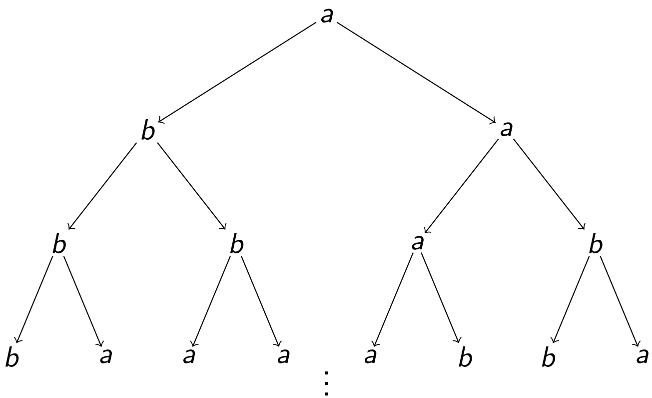
13 November 2013

Joint work with Thomas Colcombet, Denis Kuperberg, and Christof Löding

Setting

Infinite labelled tree: model of possible execution of a system where

- branching represents non-determinism in system, or
  different possibilities when the environment interacts with the system;
- label describes behavior of the system.



Setting

Let $L(\varphi)$ denote the set of infinite trees over some fixed finite alphabet $\mathbb{A}$ that satisfy some property φ.

Question

Given some property φ, is there a "simpler" φ' such that $L(\varphi) = L(\varphi')$?

Setting

Let $L(\varphi)$ denote the set of infinite trees over some fixed finite alphabet $\mathbb{A}$ that satisfy some property φ.

Question

Given some property φ, is there a "simpler" φ' such that $L(\varphi) = L(\varphi')$? $\longrightarrow$

smaller size,
restricted set of operations,
different specification language,
...

Setting

Let $L(\varphi)$ denote the set of infinite trees over some fixed finite alphabet $\mathbb{A}$ that satisfy some property φ.

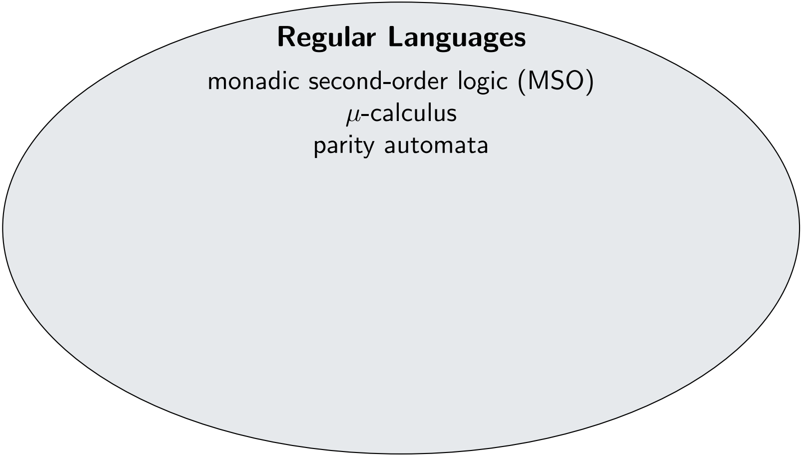
Question

Given some property φ, is there a "simpler" φ' such that $L(\varphi) = L(\varphi')$? $\longrightarrow$

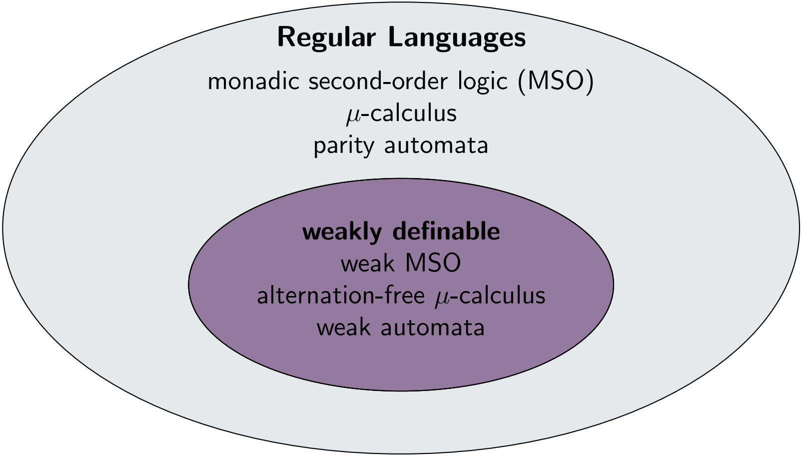
smaller size,
restricted set of operations,
different specification language,
...

Goal: analyze/decide questions like this for **regular languages** $L(\varphi)$.

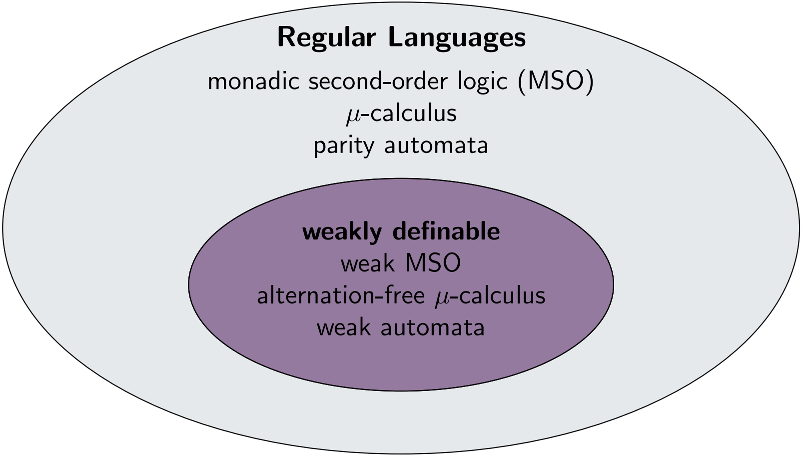
Regular languages of infinite trees



Regular languages of infinite trees



Regular languages of infinite trees



Weakly definable languages are expressive (subsuming CTL), but still have good computational properties (model-checking can be done in linear time).

Alternating parity automata on infinite trees

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

δ describes possible moves
for Eve and Adam

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

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 satisfies the **parity condition**:
  the maximum priority occurring infinitely often in the play is even.

Alternating parity automata on infinite trees

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

δ describes possible moves for Eve and Adam

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

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 satisfies the **parity condition**: the maximum priority occurring infinitely often in the play is even.

$$L(\mathcal{A}) := \{t : \text{Eve has a winning strategy in } \mathcal{A} \times t\}$$

Example

$L_1 := \{t : \text{there is some } a \text{ in } t \text{ with no } b \text{ below it}\}$.

Construct $\mathcal{A}_1$ with $Q = \{q_0, q_a, q_\perp\}$ and $\Omega : q_0, q_\perp \mapsto 1; q_a \mapsto 2$.

- In state q_0, Eve selects a path in the tree.
  If she sees an a, Eve can choose to switch to state q_a.
- In state q_a, Adam selects a path in the tree.
  If he sees a b, then he can switch to a sink state $q_\perp$.

Example

$L_1 := \{t : \text{there is some } a \text{ in } t \text{ with no } b \text{ below it}\}$.

Construct $\mathcal{A}_1$ with $Q = \{q_0, q_a, q_\perp\}$ and $\Omega : q_0, q_\perp \mapsto 1; q_a \mapsto 2$.

- In state q_0, Eve selects a path in the tree.
  If she sees an a, Eve can choose to switch to state q_a.
- In state q_a, Adam selects a path in the tree.
  If he sees a b, then he can switch to a sink state $q_\perp$.

$L_2 := \{t : \text{every } a \text{ in } t \text{ has a } b \text{ below it}\}$.

Construct $\mathcal{A}_2$ with $Q = \{q_0, q_b, q_\top\}$ and $\Omega : q_0, q_\top \mapsto 2; q_b \mapsto 1$.

- In state q_0, Adam selects a path in the tree.
  If he sees an a, Adam can choose to switch to state q_b.
- In state q_b, Eve selects a path in the tree.
  If she sees a b, then she can switch to a sink state $q_\top$.

Special types of alternating parity automata

Büchi automaton

parity automaton using only priorities $\{1, 2\}$
(we call states of priority 2 the accepting states
and states of priority 1 the non-accepting states)

Special types of alternating parity automata

Büchi automaton

parity automaton using only priorities $\{1, 2\}$
(we call states of priority 2 the accepting states
and states of priority 1 the non-accepting states)

Nondeterministic Büchi automaton

alternating Büchi automaton such that
a strategy for Eve in an acceptance game
is just a labelling of the input tree with states (called a **run**)

Special types of alternating parity automata

Büchi automaton

parity automaton using only priorities $\{1, 2\}$
(we call states of priority 2 the accepting states
and states of priority 1 the non-accepting states)

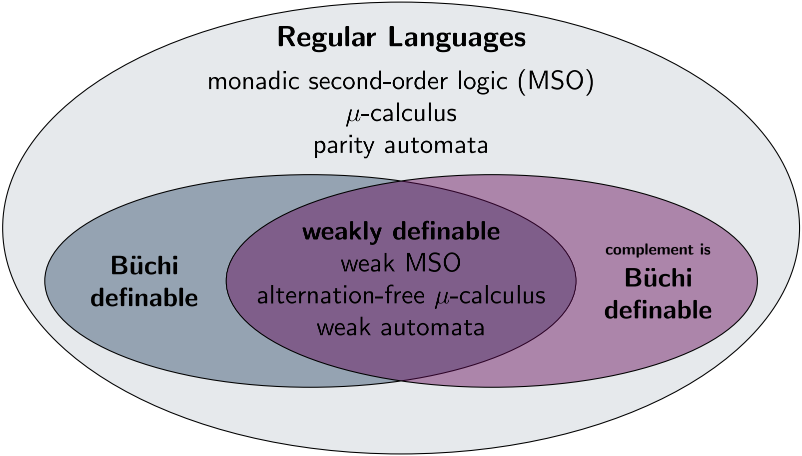
Nondeterministic Büchi automaton

alternating Büchi automaton such that
a strategy for Eve in an acceptance game
is just a labelling of the input tree with states (called a **run**)

Alternating weak automaton

alternating Büchi automaton such that
no cycle visits both accepting and non-accepting states

Regular languages of infinite trees



Theorem [Rabin '70, Kupferman+Vardi '99]

A language L is weakly definable iff L and $\overline{L}$ are Büchi definable.

Weak definability problem

Weak definability decision problem

INPUT: parity automaton $\mathcal{U}$

OUTPUT: `YES` if there exists weak automaton $\mathcal{W}$ with $L(\mathcal{W}) = L(\mathcal{U})$, `NO` otherwise

Weak definability problem

Weak definability decision problem

INPUT: parity automaton $\mathcal{U}$
OUTPUT: `YES` if there exists weak automaton $\mathcal{W}$ with $L(\mathcal{W}) = L(\mathcal{U})$,
`NO` otherwise

Theorem [Niwiński+Walukiewicz '05]

The weak definability problem is decidable if $L(\mathcal{U})$ is **deterministic**.

Weak definability problem

Weak definability decision problem

INPUT:	parity automaton $\mathcal{U}$
OUTPUT:	`YES` if there exists weak automaton $\mathcal{W}$ with $L(\mathcal{W}) = L(\mathcal{U})$, `NO` otherwise

Theorem [Niwiński+Walukiewicz '05]

The weak definability problem is decidable if $L(\mathcal{U})$ is **deterministic**.

Theorem [Facchini+Murlak+Skrzypczak '13]

The weak definability problem is decidable if $L(\mathcal{U})$ is a **game language**.

Weak definability problem

Weak definability decision problem

INPUT: parity automaton $\mathcal{U}$

OUTPUT: `YES` if there exists weak automaton $\mathcal{W}$ with $L(\mathcal{W}) = L(\mathcal{U})$,
`NO` otherwise

Theorem [Niwiński+Walukiewicz '05]

The weak definability problem is decidable if $L(\mathcal{U})$ is **deterministic**.

Theorem [Facchini+Murlak+Skrzypczak '13]

The weak definability problem is decidable if $L(\mathcal{U})$ is a **game language**.

Theorem [Colcombet,Kuperberg,Löding,VB '13]

The weak definability problem is decidable if $\mathcal{U}$ is **Büchi**.

Cost automata

Finite state automaton $\mathcal{A}$

$+$ finite set of counters (initialized to 0, values range over $\mathbb{N}$)
$+$ counter operations on transitions (increment $\mathtt{i}$, reset $\mathtt{r}$, no change ε)

Semantics

$[\![\mathcal{A}]\!]$: infinite trees $\to \mathbb{N} \cup \{\infty\}$

$$[\![\mathcal{A}]\!](t) := \min\{n : \exists \text{ winning strategy for Eve in } \mathcal{A} \times t \text{ such that every play has counter values at most } n\}$$

Example

$f(t) := \min\{n : \text{every } a \text{ has a } b \text{ at most } n \text{ nodes below it}\}$.

Construct $\mathcal{A}$ with $Q = \{q_0, q_b, q_\top\}$, $\Omega : q_0, q_\top \mapsto 2; q_b \mapsto 1$, 1 counter.

- In state q_0, Adam selects a path in the tree.
  The counter operation is ε.
  If he sees an a, Adam can choose to switch to state q_b.
- In state q_b, Eve selects a path in the tree.
  If she sees an a, then the counter is incremented.
  If she sees a b, then she can switch to a sink state $q_\top$.

Cost automata

Finite state automaton $\mathcal{A}$

$+$ finite set of counters (initialized to 0, values range over $\mathbb{N}$)
$+$ counter operations on transitions (increment $\mathtt{i}$, reset $\mathtt{r}$, no change ε)

Semantics

$[\![\mathcal{A}]\!]$: infinite trees $\to \mathbb{N} \cup \{\infty\}$

$$[\![\mathcal{A}]\!](t) := \min\{n : \exists \text{ winning strategy for Eve in } \mathcal{A} \times t \text{ such that every play has counter values at most } n\}$$

Boundedness with respect to language K (written $[\![\mathcal{A}]\!] \approx \chi_K$)

$[\![\mathcal{A}]\!] \approx \chi_K$ if there is bound $n \in \mathbb{N}$ such that $[\![\mathcal{A}]\!](t) \leq n$ if $t \in K$ and
$[\![\mathcal{A}]\!](t) = \infty$ if $t \notin K$

Decidability of boundedness for cost automata

Decidability of $\approx$ is known for some types of cost automata.

- cost automata over **finite words**
  [Colcombet '09, Bojanczyk+Colcombet '06]

- cost automata over **infinite words**
  [Colcombet unpublished]

- cost automata over **finite trees**
  [Colcombet+Löding '10]

Decidability of boundedness for cost automata

Decidability of $\approx$ is known for some types of cost automata.

- cost automata over **finite words**
  [Colcombet '09, Bojanczyk+Colcombet '06]

- cost automata over **infinite words**
  [Colcombet unpublished]

- cost automata over **finite trees**
  [Colcombet+Löding '10]

- counter-weak automata over **infinite trees**
  [Kuperberg+VB '11]

Reduction to boundedness

Many problems for a regular language L have been reduced to deciding $\approx$ for special types of cost automata.

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

  is there some n such that $L^* = \{\epsilon\} \cup L^1 \cup L^2 \cup \cdots \cup L^n$?

- **Star-height problem**
  [Hashiguchi '88, Kirsten '05, Colcombet+Löding '08]

  given n, is there a regular expression for L with at most n nestings of Kleene star?

- **Parity-index problem**
  [reduction in Colcombet+Löding '08, decidability open]

  given $i < j$, is there a nondeterministic parity automaton for L which uses only priorities $\{i, i+1, \ldots, j\}$?

Reduction to boundedness

Many problems for a regular language L have been reduced to deciding $\approx$ for special types of cost automata.

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

  is there some n such that $L^* = \{\epsilon\} \cup L^1 \cup L^2 \cup \cdots \cup L^n$?

  distance

- **Star-height problem**
  [Hashiguchi '88, Kirsten '05, Colcombet+Löding '08]

  given n, is there a regular expression for L with at most n nestings of Kleene star?

  nested distance-desert

- **Parity-index problem**
  [reduction in Colcombet+Löding '08, decidability open]

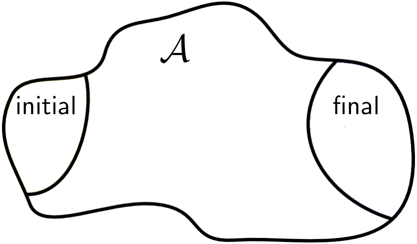
  given $i < j$, is there a nondeterministic parity automaton for L which uses only priorities $\{i, i+1, \ldots, j\}$?

  cost-parity

Reduction to boundedness (example)

Finite power property decision problem

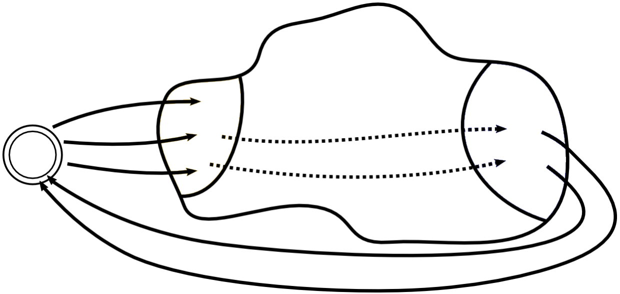
INPUT: Finite state automaton $\mathcal{A}$ over finite words with $L = L(\mathcal{A})$
OUTPUT: YES if there is $n \in \mathbb{N}$ with $L^* = \{\epsilon\} \cup L^1 \cup L^2 \cup \cdots \cup L^n$,
NO otherwise



Reduction to boundedness (example)

Finite power property decision problem

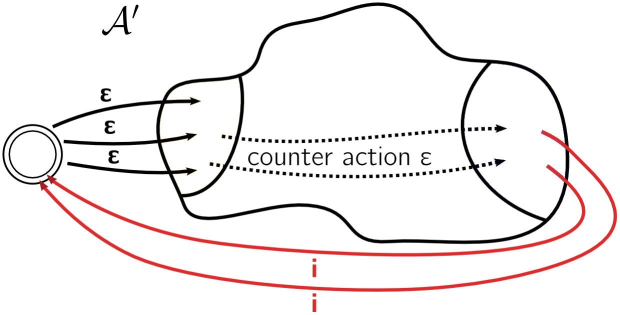
INPUT:	Finite state automaton $\mathcal{A}$ over finite words with $L = L(\mathcal{A})$
OUTPUT:	YES if there is $n \in \mathbb{N}$ with $L^* = \{\epsilon\} \cup L^1 \cup L^2 \cup \cdots \cup L^n$,
	NO otherwise



Reduction to boundedness (example)

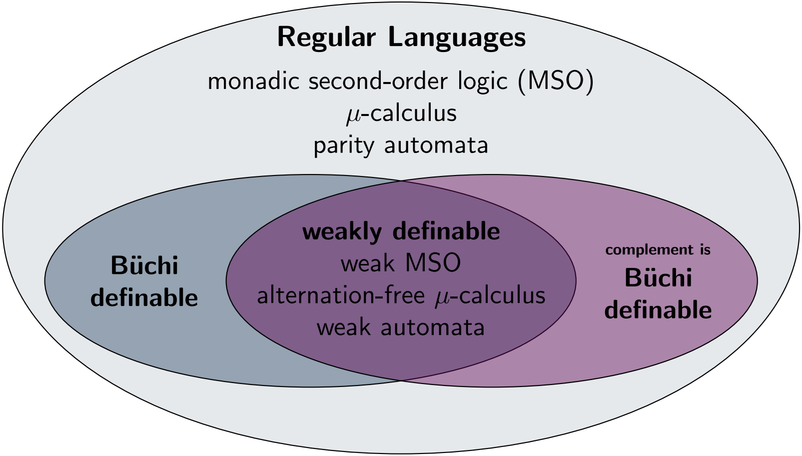
Finite power property decision problem

INPUT: Finite state automaton $\mathcal{A}$ over finite words with $L = L(\mathcal{A})$
OUTPUT: $\texttt{YES}$ if there is $n \in \mathbb{N}$ with $L^* = \{\epsilon\} \cup L^1 \cup L^2 \cup \cdots \cup L^n$,
$\texttt{NO}$ otherwise

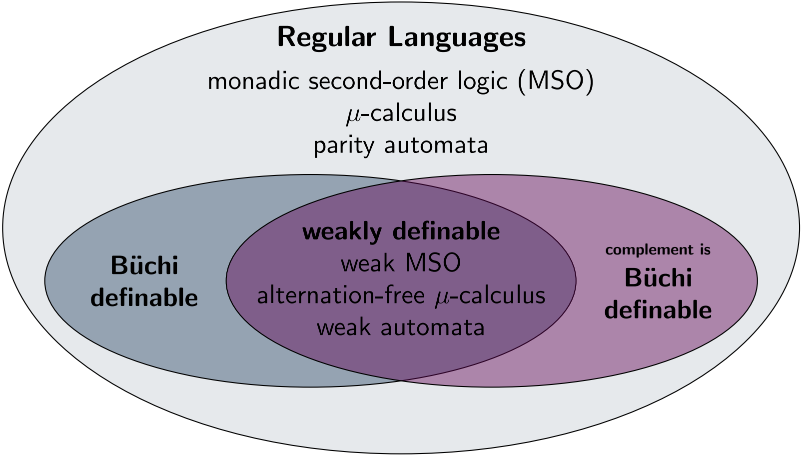


Finite power property holds iff $[\![\mathcal{A}']\!] \approx \chi_{L^*}$

Reduction of weak definability to boundedness



Reduction of weak definability to boundedness



Goal

Given Büchi automaton $\mathcal{U}$, construct cost automaton $\mathcal{Q}$ such that $[\![\mathcal{Q}]\!] \approx \chi_{\overline{L(\mathcal{U})}}$ iff $L(\mathcal{U})$ is weakly definable.

Block counting

Given nondeterministic Büchi automata $\mathcal{U}$ and $\mathcal{V}$

- fix some tree t and let $\rho_{\mathcal{U}}$ and $\rho_{\mathcal{V}}$ be runs of $\mathcal{U}$ and $\mathcal{V}$ on t, with accepting states marked with ◯

$$
\begin{array}{lccccccccccc}
\rho_{\mathcal{U}} & q_0 & q_1 & q_2 & q_3 & q_4 & q_5 & q_6 & q_7 & q_8 & q_9 & \cdots \\
\rho_{\mathcal{V}} & r_0 & r_1 & r_2 & r_3 & r_4 & r_5 & r_6 & r_7 & r_8 & r_9 & \cdots
\end{array}
$$

- divide each branch in the composed run into **blocks** containing accepting state for $\mathcal{V}$ followed by accepting state for $\mathcal{U}$

Block counting

Given nondeterministic Büchi automata $\mathcal{U}$ and $\mathcal{V}$

- fix some tree t and let $\rho_{\mathcal{U}}$ and $\rho_{\mathcal{V}}$ be runs of $\mathcal{U}$ and $\mathcal{V}$ on t, with accepting states marked with $\bigcirc$

$$\begin{array}{lcccc|cc|cccc|c}
\rho_{\mathcal{U}} & q_0 & q_1 & q_2 & q_3 & q_4 & q_5 & q_6 & q_7 & q_8 & q_9 & \cdots \\
\rho_{\mathcal{V}} & r_0 & r_1 & r_2 & r_3 & r_4 & r_5 & r_6 & r_7 & r_8 & r_9 & \cdots \\
 & & & & 1 & & 2 & & & & 3 &
\end{array}$$

- divide each branch in the composed run into **blocks** containing accepting state for $\mathcal{V}$ followed by accepting state for $\mathcal{U}$

Block counting

Given nondeterministic Büchi automata $\mathcal{U}$ and $\mathcal{V}$

- fix some tree t and let $\rho_{\mathcal{U}}$ and $\rho_{\mathcal{V}}$ be runs of $\mathcal{U}$ and $\mathcal{V}$ on t, with accepting states marked with ◯

$$\begin{array}{lcccc|cc|cccc|l} \rho_{\mathcal{U}} & q_0 & q_1 & q_2 & q_3 & q_4 & q_5 & q_6 & q_7 & q_8 & q_9 & \cdots \\ \rho_{\mathcal{V}} & r_0 & r_1 & r_2 & r_3 & r_4 & r_5 & r_6 & r_7 & r_8 & r_9 & \cdots \\ & & & & 1 & & 2 & & & & 3 & \end{array}$$

- divide each branch in the composed run into **blocks** containing accepting state for $\mathcal{V}$ followed by accepting state for $\mathcal{U}$

Theorem [Rabin '70]

If there are at least $m = |Q_{\mathcal{U}}| \cdot |Q_{\mathcal{V}}| + 1$ blocks on every branch in the composed run, then $L(\mathcal{U}) \cap L(\mathcal{V}) \neq \emptyset$.

Weak automaton construction [Kupferman+Vardi '99]

Given nondeterministic Büchi automata $\mathcal{U}$ and $\mathcal{V}$ with $L(\mathcal{U}) = \overline{L(\mathcal{V})}$

Construct weak automaton $\mathcal{W}$ such that $L(\mathcal{W}) = L(\mathcal{V})$

- Adam selects transition from $\Delta_{\mathcal{U}}$
- Eve selects transition from $\Delta_{\mathcal{V}}$ and direction

Adam $\rho_{\mathcal{U}}$	q_0	q_1	q_2	q_3	q_4	q_5	q_6	q_7	q_8	q_9	$\cdots$
Eve $\rho_{\mathcal{V}}$	r_0	r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_8	r_9	$\cdots$

Weak automaton construction [Kupferman+Vardi '99]

Given nondeterministic Büchi automata $\mathcal{U}$ and $\mathcal{V}$ with $L(\mathcal{U}) = \overline{L(\mathcal{V})}$

Construct weak automaton $\mathcal{W}$ such that $L(\mathcal{W}) = L(\mathcal{V})$

- Adam selects transition from $\Delta_{\mathcal{U}}$
- Eve selects transition from $\Delta_{\mathcal{V}}$ and direction
- Accept/reject depending on occurrences of ◯

Adam $\rho_{\mathcal{U}}$	q_0	q_1	q_2	q_3	q_4	q_5	q_6	q_7	q_8	q_9	$\cdots$
Eve $\rho_{\mathcal{V}}$	r_0	r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_8	r_9	$\cdots$

Weak automaton construction [Kupferman+Vardi '99]

Given nondeterministic Büchi automata $\mathcal{U}$ and $\mathcal{V}$ with $L(\mathcal{U}) = \overline{L(\mathcal{V})}$

Construct weak automaton $\mathcal{W}$ such that $L(\mathcal{W}) = L(\mathcal{V})$

- Adam selects transition from $\Delta_{\mathcal{U}}$
- Eve selects transition from $\Delta_{\mathcal{V}}$ and direction
- Accept/reject depending on occurrences of ◯
- Store the block number in the state, up to value $m := |Q_{\mathcal{U}}| \cdot |Q_{\mathcal{V}}| + 1$
  once m blocks have been witnessed, stabilize in rejecting state

Adam $\rho_{\mathcal{U}}$	q_0	q_1	q_2	q_3	|	q_4	q_5	|	q_6	q_7	q_8	q_9	| $\cdots$
Eve $\rho_{\mathcal{V}}$	r_0	r_1	r_2	r_3	|	r_4	r_5	|	r_6	r_7	r_8	r_9	| $\cdots$
					1			2					3

Reduction of weak definability to boundedness

Given nondeterministic Büchi automaton $\mathcal{U}$

Construct cost automaton $\mathcal{Q}$ s.t. $[\![\mathcal{Q}]\!] \approx \chi_{\overline{L(\mathcal{U})}}$ iff $L(\mathcal{U})$ is weakly definable

- Adam selects transition from $\Delta_{\mathcal{U}}$
- Eve selects direction and guesses whether to output ◯
- Accept/reject depending on occurrences of ◯

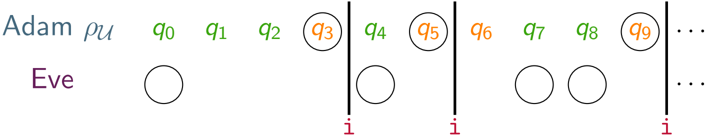
Adam $\rho_{\mathcal{U}}$	q_0	q_1	q_2	q_3	q_4	q_5	q_6	q_7	q_8	q_9	$\cdots$
Eve	◯				◯			◯	◯		$\cdots$

Reduction of weak definability to boundedness

Given nondeterministic Büchi automaton $\mathcal{U}$

Construct cost automaton $\mathcal{Q}$ s.t. $[\![\mathcal{Q}]\!] \approx \chi_{\overline{L(\mathcal{U})}}$ iff $L(\mathcal{U})$ is weakly definable

- Adam selects transition from $\Delta_{\mathcal{U}}$
- Eve selects direction and guesses whether to output $\bigcirc$
- Accept/reject depending on occurrences of $\bigcirc$
- Store the block number in the counter



Decidability of boundedness for cost automata

Decidability of $\approx$ for cost automata over infinite trees is open in general, but is known in some special cases.

Theorem [Kuperberg+VB '11]

The boundedness relation $\approx$ is decidable for **counter-weak cost automata** over infinite trees.

Counter-weak cost automaton

alternating cost-Büchi automaton such that in any cycle with both accepting and non-accepting states, there is a counter which is incremented but not reset

Deciding weak definability for Büchi input

Theorem

Given Büchi automaton $\mathcal{U}$, we can construct a counter-weak cost automaton $\mathcal{Q}$ such that the following are equivalent:

- $L(\mathcal{U})$ is weakly definable;
- $[\![\mathcal{Q}]\!] \approx \chi_{\overline{L(\mathcal{U})}}$.

$$\downarrow$$

Theorem [Kuperberg+VB '11]

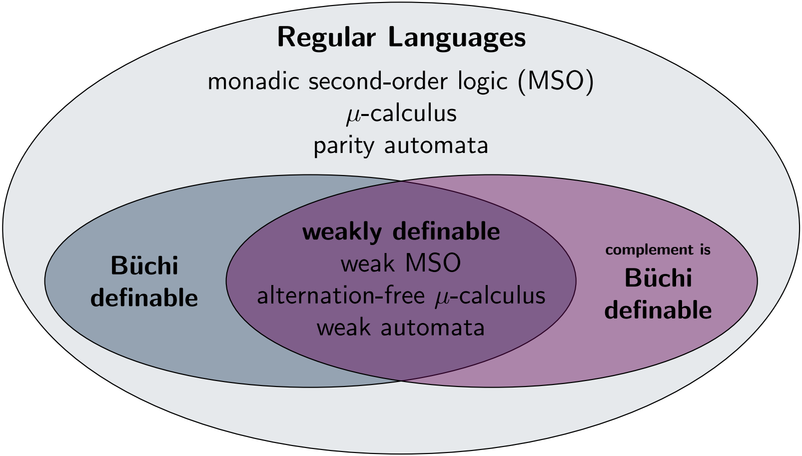
The boundedness relation $\approx$ is decidable for counter-weak cost automata.

$$\downarrow$$

Theorem [Colcombet+Kuperberg+Löding+VB '13]

Given Büchi automaton $\mathcal{U}$, it is decidable whether $L(\mathcal{U})$ is weakly definable.

Regular languages of infinite trees



Conclusion

There are many questions related to determining whether there is "simpler" way to define some regular language.

Cost automata can be used to help prove the decidability of some definability problems for regular languages of infinite trees.

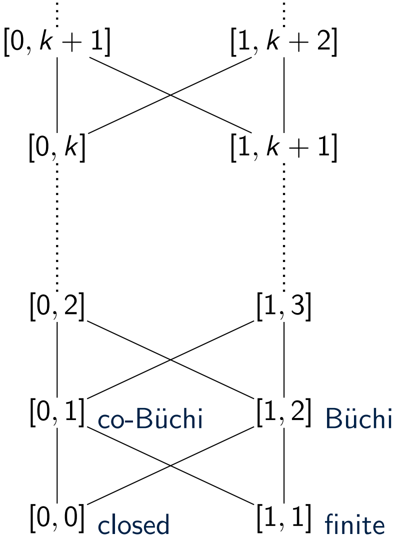
- The **weak definability problem** is decidable when the input is a Büchi automaton.
- The **co-Büchi definability problem** is decidable when the input is a parity automaton.

Open questions

Can we use cost automata to solve other questions like this?
(e.g., the nondeterministic parity index problem)

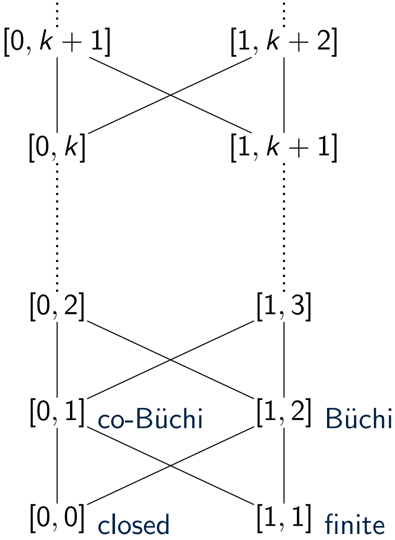
Is $\approx$ decidable for larger classes of cost automata over infinite trees?

Nondeterministic Mostowski hierarchy



A language L has **index** $[i, j]$ if there is some nondeterministic parity automaton using priorities from $\{i, i+1, \ldots, j\}$ that recognizes L.

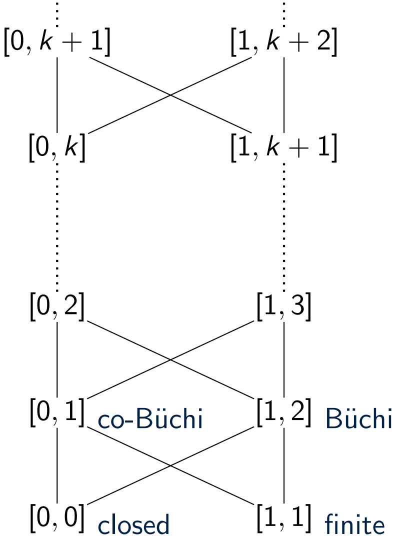
Nondeterministic Mostowski hierarchy



A language L has **index** $[i, j]$ if there is some nondeterministic parity automaton using priorities from $\{i, i+1, \ldots, j\}$ that recognizes L.

Hierarchy is **strict** over infinite trees [Niwinski '86]

Nondeterministic Mostowski hierarchy



A language L has **index** $[i, j]$ if there is some nondeterministic parity automaton using priorities from $\{i, i+1, \ldots, j\}$ that recognizes L.

Hierarchy is **strict** over infinite trees [Niwinski '86]

Parity index problem
Given a parity automaton $\mathcal{A}$ and index $[i, j]$, determine whether $L(\mathcal{A})$ has index at most $[i, j]$.

Deciding co-Büchi definability

Theorem [Colcombet+Löding '08]

Given parity automaton $\mathcal{U}$ and index $[i, j]$, we can construct a cost-parity automaton $\mathcal{B}$ using priorities $[i, j]$ such that the following are equivalent:

- $L(\mathcal{U})$ has index $[i, j]$;
- $[\![\mathcal{B}]\!] \approx \chi_{L(\mathcal{U})}$.

Deciding co-Büchi definability

Theorem [Colcombet+Löding '08]

Given parity automaton $\mathcal{U}$ and index $[i, j]$, we can construct a cost-parity automaton $\mathcal{B}$ using priorities $[i, j]$ such that the following are equivalent:

- $L(\mathcal{U})$ has index $[i, j]$;
- $[\![\mathcal{B}]\!] \approx \chi_{L(\mathcal{U})}$.

$$\downarrow$$

Theorem [VB '11]

$[\![\mathcal{B}]\!] \approx \chi_L$ is decidable for cost-parity automata using priorities $\{0, 1\}$ and regular languages L.

$$\downarrow$$

Theorem

Given parity automaton $\mathcal{U}$, it is decidable whether $L(\mathcal{U})$ has index $[0, 1]$.