Automata and Logic on Trees

Monadic Second Order Logic on Trees

Wim Martens[1] Stijn Vansummeren[2]

[1]University of Dortmund, Germany
[2]Hasselt University, Belgium

Our story so far

- **Tree automata**
  deterministic, bottum-up, top-down, constructions, ...
- **Decision problems for tree automata**
  Equivalence, universality, emptiness, intersection emptiness, ...,

Our story so far

- **Tree automata**
  deterministic, bottum-up, top-down, constructions, ...
- **Decision problems for tree automata**
  Equivalence, universality, emptiness, intersection emptiness, ...,

Automata and Logic on Trees
Where's the logic?

Outline

Outline

Trees as structures

Recall that in mathematical logic

- A relational vocabulary is a sequence of relation names $(R, \ldots, S)$ with associated arities $\text{arity}(R), \ldots, \text{arity}(S)$.
- A structure $\mathscr{S}$ over $(R, \ldots, S)$ is a tuple

$$(D, R^{\mathscr{S}}, \ldots, S^{\mathscr{S}})$$

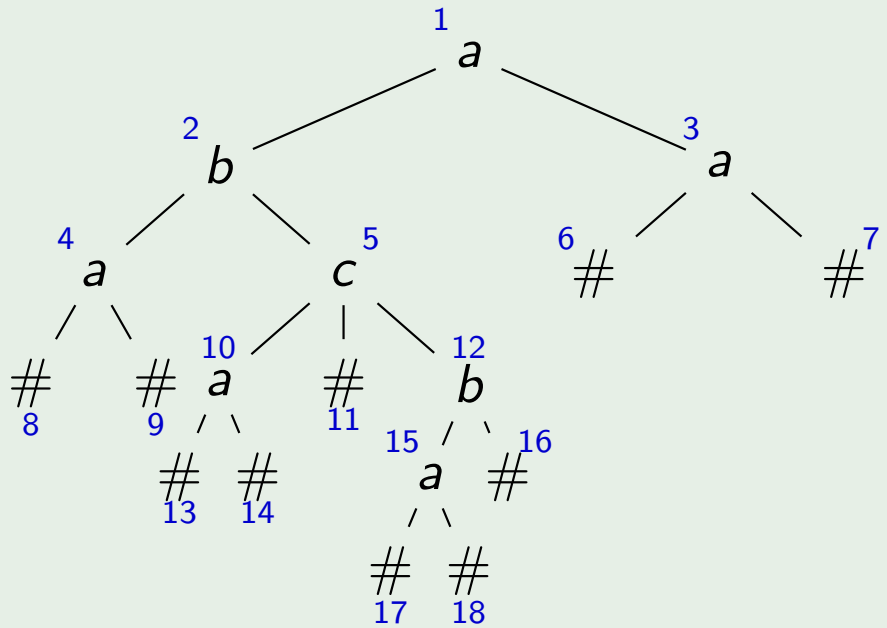
with D a finite set, and $R^{\mathscr{S}} \subseteq D^{\text{arity}(R)}, \ldots, S^{\mathscr{S}} \subseteq D^{\text{arity}(S)}$

Trees as structures

A tree t over a ranked alphabet $\Sigma = \{a, \ldots, b\}$ naturally corresponds to a structure $\underline{t}$ over the vocabulary $V_\Sigma = (E, <, L_a, \ldots, L_b)$

Example

Tree



Structure

- $D^{\underline{t}} = \{1, 2, 3, \ldots\}$
- $E^{\underline{t}}(1,2), E^{\underline{t}}(1,3), E^{\underline{t}}(2,4), \ldots$
- $2 <^{\underline{t}} 3, 4 <^{\underline{t}} 5, \ldots$
- $L_a = \{1, 3, 4, 10, 15\}$
- $L_b = \{2, 12\}$
- $\ldots$

Outline

First Order Logic over Trees

FO over the vocabulary $V_\Sigma = (E, <, L_a, \ldots, L_b)$:

$$\begin{array}{rcl} \phi & ::= & x = y \mid E(x,y) \mid x < y \mid L_a(x) \mid \cdots \mid L_b(y) \\ & \mid & \phi \wedge \phi \mid \neg\phi \mid \exists x\, \phi \end{array}$$

with the usual abbreviations $\phi \vee \phi$, $\phi \to \phi$, $\forall x\, \phi$, ...

Example

All a-labeled nodes in t have a b-labeled child if, and only if,

$$\underline{t} \models \forall x(L_a(x) \to \exists y(L_b(y) \wedge E(x,y)))$$

First Order Logic over Trees

FO over the vocabulary $V_\Sigma = (E, <, L_a, \ldots, L_b)$**:**

$$\begin{array}{rcl} \phi & ::= & x = y \mid E(x,y) \mid x < y \mid L_a(x) \mid \cdots \mid L_b(y) \\ & \mid & \phi \wedge \phi \mid \neg\phi \mid \exists x\, \phi \end{array}$$

with the usual abbreviations $\phi \vee \phi$, $\phi \rightarrow \phi$, $\forall x\, \phi$, ...

Notation and terminology

- If ϕ is a sentence over V_Σ then

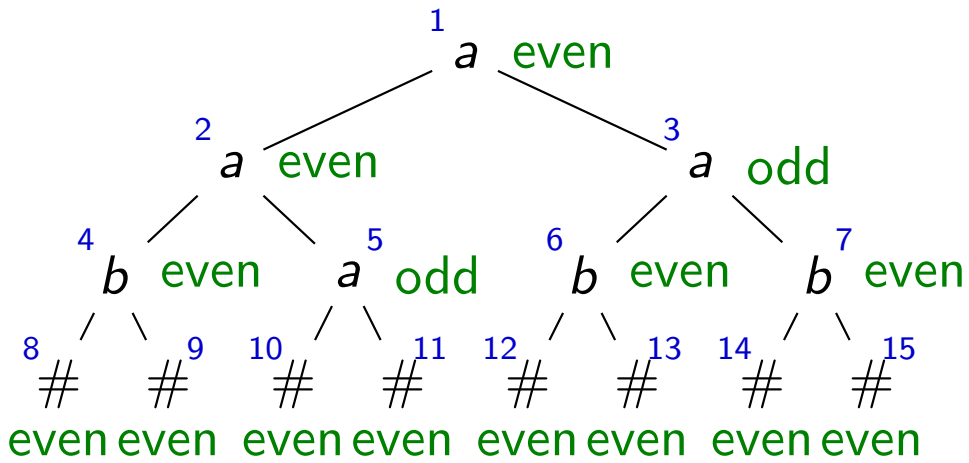
$$\text{Language}(\phi) := \{t \mid t \text{ a ranked tree over } \Sigma \text{ such that } \underline{t} \models \phi\}$$

- A tree language S is FO-definable if there is some ϕ with $S = \text{Language}(\phi)$

First Order Logic over Trees

Why consider logic on trees?

- A formula describes a specification for a language
- An automaton gives an algorithm for recognizing a language



$$\forall x(L_a(x) \rightarrow \exists y(L_b(y) \wedge E(x,y)))$$

First Order Logic over Trees

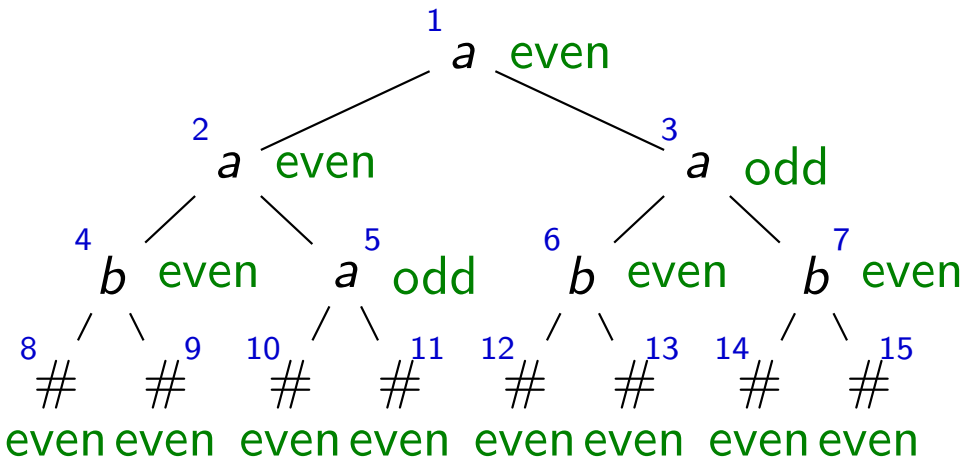
Question: is every regular tree language FO-definable?

(a) Yes (b) No (c) Only if P = NP

First Order Logic over Trees

Theorem

The language L consisting of all trees with an even number of a-nodes is not FO-definable



Intuitively, this is because FO cannot "count"
Formal argumentation trough Ehrenfeucht-Fraïssé games

First Order Logic over Trees

Theorem

The language L consisting of all trees with an even number of a-nodes is not FO-definable

Proof sketch:

- Fact: if ϕ is a first-order sentence with quantifier depth k and the duplicator wins the Ehrenfeucht-Fraïssé game of k rounds on t and t' then both $\underline{t} \models \phi$ and $\underline{t'} \models \phi$ or both $\underline{t} \not\models \phi$ and $\underline{t'} \not\models \phi$
- Then show that for every quantifier depth k you can find t and t' such that $t \in L$, $t' \notin L$, and the duplicator wins the k-round Ehrenfeucht-Fraïssé game on t and t'

Outline

Monadic Second Order Logic

MSO is the extension of FO with set variables X:

$$\begin{array}{rcl} \phi & ::= & x = y \mid E(x,y) \mid x < y \mid L_a(x) \mid \cdots \mid L_b(y) \\ & \mid & \phi \wedge \phi \mid \neg\phi \mid \exists x\, \phi \mid X(x) \mid \exists X\, \phi \end{array}$$

with the usual abbreviations $\phi \vee \phi$, $\phi \to \phi$, $\forall x\, \phi$, $\forall X\, \phi$, ...

Example

Every a-labeled node in t has a b-labeled descendant if, and only if

$$\underline{t} \models \forall n\, L_a(n) \to \exists X\, (\phi \wedge \psi \wedge \rho)$$

where

$$\phi := X(n) \wedge \forall m \forall m' \Big(E(m,m') \wedge X(m) \to X(m') \Big)$$

$$\psi := \exists m (X(m) \wedge L_b(m))$$

$$\rho := \forall Y \big(\phi(Y) \to \forall m (X(m) \to Y(m)) \big)$$

Monadic Second Order Logic

MSO is the extension of FO with set variables X:

$$\begin{array}{rcl} \phi & ::= & x = y \mid E(x,y) \mid x < y \mid L_a(x) \mid \cdots \mid L_b(y) \\ & \mid & \phi \wedge \phi \mid \neg\phi \mid \exists x\, \phi \mid X(x) \mid \exists X\, \phi \end{array}$$

with the usual abbreviations $\phi \vee \phi$, $\phi \rightarrow \phi$, $\forall x\, \phi$, $\forall X\, \phi$, ...

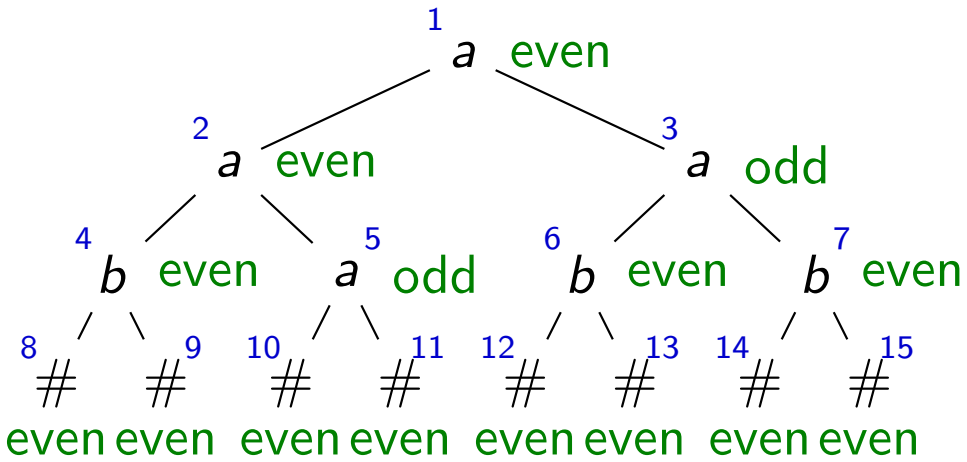
Notation and terminology

- If ϕ is an MSO sentence over V_Σ then

$$\text{Language}(\phi) := \{t \mid t \text{ a ranked tree over } \Sigma \text{ such that } \underline{t} \models \phi\}$$

- A tree language S is MSO-definable if there is some MSO formula ϕ with $S = \text{Language}(\phi)$

Exercise: construct an MSO formula ϕ that recognizes the set of all trees with an even number of a's

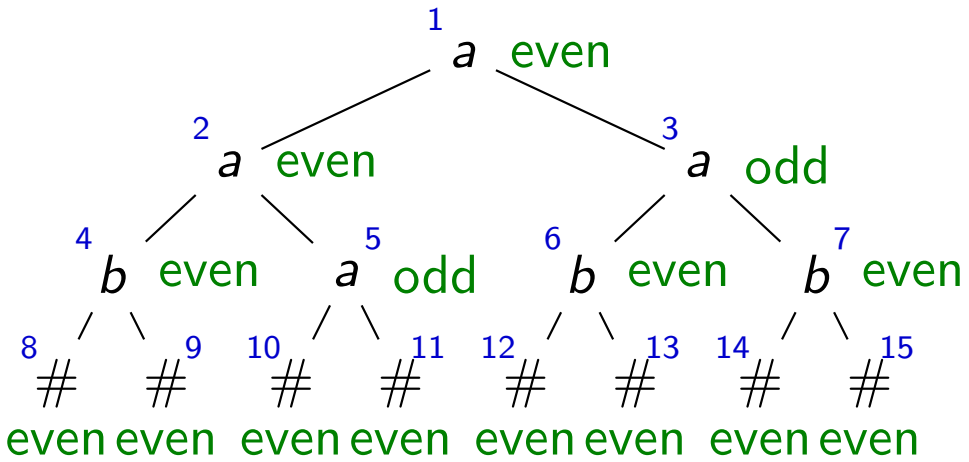


Regularity implies MSO-definability

Theorem: Every ranked regular tree language is MSO-definable

For every tree automaton A there exists an MSO formula ϕ with $\text{Language}(A) = \text{Language}(\phi)$.

Observation 1. If $\text{States}(A) = \{q_1, \ldots, q_n\}$ then every run of A on a tree t can be represented by sets of nodes $Q_1, \ldots, Q_n$



EVEN $:= \{1, 2, 4, 6, 7, \ldots\}$
ODD $:= \{3, 5\}$

Regularity implies MSO-definability

Theorem: Every ranked regular tree language is MSO-definable

For every tree automaton A there exists an MSO formula ϕ with $\text{Language}(A) = \text{Language}(\phi)$.

Observation 2. We can define that $Q_1, \ldots, Q_n$ represents an accepting run in FO.

- Every node is assigned at most one state:

$$\phi_1 := \bigwedge_{i \neq j} \forall x \Big(Q_i(x) \rightarrow \neg Q_j(x) \Big)$$

Regularity implies MSO-definability

Theorem: Every ranked regular tree language is MSO-definable

For every tree automaton A there exists an MSO formula ϕ with $\text{Language}(A) = \text{Language}(\phi)$.

Observation 2. We can define that $Q_1, \ldots, Q_n$ represents an accepting run in FO.

- The root node is assigned an accepting state:

$$\phi_2 := \forall x \left(\neg(\exists y\, E(y,x)) \rightarrow \bigvee_{q_i \in \text{Final}(A)} Q_i(x) \right)$$

Regularity implies MSO-definability

Theorem: Every ranked regular tree language is MSO-definable

For every tree automaton A there exists an MSO formula ϕ with $\text{Language}(A) = \text{Language}(\phi)$.

Observation 2. We can define that $Q_1, \ldots, Q_n$ represents an accepting run in FO.

- Leaf nodes are assigned a state in accordance with $\text{Rules}(A)$:

$$\phi_3 := \bigwedge_{\substack{a \in \text{Alphabet}(A) \\ \text{rank}(a)=0}} \forall x \left(L_a(x) \rightarrow \bigvee_{\varepsilon \xrightarrow{a} q_i \in \text{Rules}(A)} Q_i(x) \right)$$

Regularity implies MSO-definability

Theorem: Every ranked regular tree language is MSO-definable

For every tree automaton A there exists an MSO formula ϕ with $\text{Language}(A) = \text{Language}(\phi)$.

Observation 2. We can define that $Q_1, \ldots, Q_n$ represents an accepting run in FO.

- Internal nodes are assigned a state in accordance with $\text{Rules}(A)$:

$$\phi_4 := \bigwedge_{\substack{a \in \text{Alphabet}(A) \\ \text{rank}(a)=n}} \forall x \forall y_1 \ldots \forall y_n$$

$$\Big(L_a(x) \wedge E(x, y_1) \wedge \cdots \wedge E(x, y_n) \wedge y_1 < y_2 \wedge \cdots \wedge y_{n-1} < y_n \Big)$$

$$\rightarrow \bigvee_{(q_{i_1}, \ldots, q_{i_n}) \xrightarrow{a} q_i \in \text{Rules}(A)} \left(Q_{i_1}(y_1) \wedge \cdots \wedge Q_{i_n}(y_n) \wedge Q_i(x) \right)$$

Regularity implies MSO-definability

Theorem: Every ranked regular tree language is MSO-definable

For every tree automaton A there exists an MSO formula ϕ with $\text{Language}(A) = \text{Language}(\phi)$.

Observation 3. We can guess $Q_1, \ldots, Q_n$ in MSO:

$$\phi := \exists Q_1 \ldots \exists Q_n \, \phi_1 \wedge \phi_2 \wedge \phi_3 \wedge \phi_4$$

Clearly, $\text{Language}(A) = \text{Language}(\phi)$.

Hence, every ranked regular tree language is MSO-definable

First Order Logic over Trees

Question: is every MSO-definable tree language regular?

(a) Yes
(b) No
(c) Only if P = NP
(d) Who cares?

Does MSO-definability imply regularity?

Question: Is Language(ϕ) regular for every MSO sentence ϕ?

Observation 1: ϕ is equivalently expressed by a formula

$$\begin{array}{rcl} \psi & := & X \subseteq Y \mid \mathrm{Sing}(X) \mid E(X,Y) \mid X < Y \mid X \subseteq L_a \mid \cdots \mid X \subseteq L_b \\ & \mid & \psi \wedge \psi \mid \neg\psi \mid \exists X\, \phi \end{array}$$

where

- the X's range over sets of nodes
- $\mathrm{Sing}(X)$ indicates that X is a singleton
- $E(X,Y)$ indicates that X and Y are singletons $\{x\}$ and $\{y\}$ with x a parent of y
- $X < Y$ indicates that X and Y are singletons $\{x\}$ and $\{y\}$ with x a left sibling of y
- $X \subseteq L_a$ indicates that all nodes in X are labeled a

Does MSO-definability imply regularity?

Question: Is Language(ϕ) regular for every MSO sentence ϕ?

Observation 1: ϕ is equivalently expressed by a formula

$$\begin{array}{rcl} \psi & := & X \subseteq Y \mid \mathrm{Sing}(X) \mid E(X,Y) \mid X < Y \mid X \subseteq L_a \mid \cdots \mid X \subseteq L_b \\ & \mid & \psi \wedge \psi \mid \neg\psi \mid \exists X\, \phi \end{array}$$

Example

$$t \models \forall x(L_a(x) \rightarrow \exists y(E(x,y) \wedge L_b(y)))$$

if and only if

$$t \models \forall X((\mathrm{Sing}(X) \wedge X \subseteq L_A) \rightarrow \exists Y\, E(X,Y) \wedge Y \subseteq L_b)$$

Does MSO-definability imply regularity?

Question: Is Language(ϕ) regular for every MSO sentence ϕ?

Observation 1: ϕ is equivalently expressed by a formula

$$\begin{array}{rcl} \psi & := & X \subseteq Y \mid \mathrm{Sing}(X) \mid E(X,Y) \mid X < Y \mid X \subseteq L_a \mid \cdots \mid X \subseteq L_b \\ & \mid & \psi \wedge \psi \mid \neg\psi \mid \exists X\, \phi \end{array}$$

Example

$$t \models \forall x(L_a(x) \rightarrow \exists y(E(x,y) \wedge L_b(y)))$$

if and only if

$$t \models \forall X((\mathrm{Sing}(X) \wedge X \subseteq L_A) \rightarrow \exists Y\, E(X,Y) \wedge Y \subseteq L_b)$$

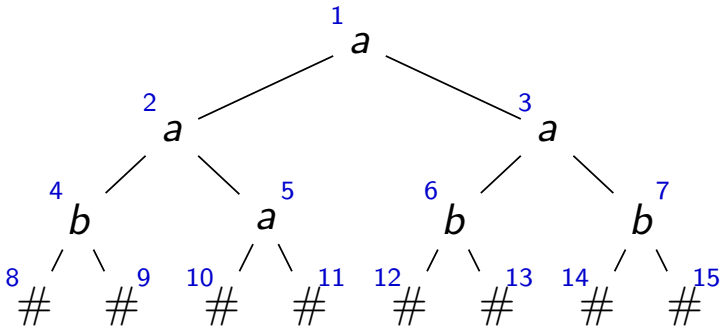
Notation

Denote this logic by MSO_0

Does MSO-definability imply regularity?

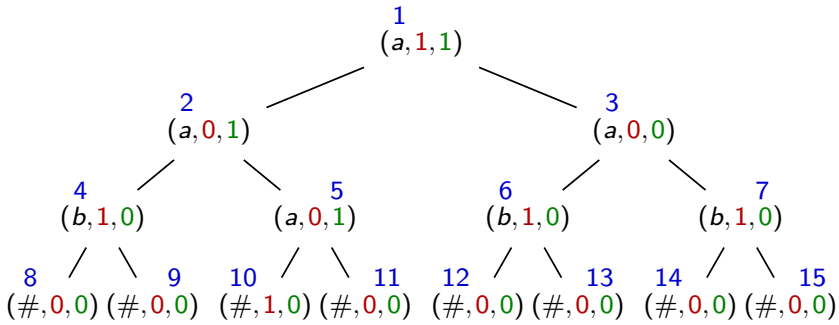
New question: Is Language(ψ) regular for every MSO_0 sentence ψ?

Observation 2: We can view a formula $\psi(X_1, \ldots, X_n)$ as defining a tree language over the extended alphabet $\Sigma \times \{0,1\}^n$



$V_1 := \{1,4,10\}$ $V_2 := \{1,2,5\}$

Tree t + two sets of nodes V_1 and V_2



Single tree $t[V_1, V_2]$ over $\Sigma \times \{0,1\}^2$

Does MSO-definability imply regularity?

New question: Is Language(ψ) regular for every MSO_0 sentence ψ?

Observation 2: We can view a formula $\psi(X_1, \ldots, X_n)$ as defining a tree language over the extended alphabet $\Sigma \times \{0,1\}^n$

Definition

- For a formula $\psi(X_1, \ldots, X_n)$, define

$$\text{Language}(\psi; X_1, \ldots, X_n) := \{\underline{t[V_1, \ldots, V_n]} \mid \underline{t} \models \psi(V_1, \ldots, V_n)\}$$

- In particular, when ψ is a sentence we have

$$\text{Language}(\psi;) = \text{Language}(\psi).$$

Does MSO-definability imply regularity?

New question: Is Language$(\psi; X_1, \ldots, X_n)$ regular for every MSO_0 formula ψ?

Answer: Yes, by induction on ψ:

- **Case** $\psi = X_i \subseteq X_j$.

  Exercise: construct the automaton for Language$(X_i \subseteq X_j; X_1, \ldots, X_n)$. It can be done using only two states.

Does MSO-definability imply regularity?

New question: Is Language$(\psi; X_1, \ldots, X_n)$ regular for every MSO_0 formula ψ?

Answer: Yes, by induction on ψ:

- **Case** $\psi = X_i \subseteq X_j$.

  Construct A_ψ with $\mathrm{States}(A_\psi) = \{\mathrm{ok}, \mathrm{notok}\}$, $\mathrm{Final}(A_\psi) = \{\mathrm{ok}\}$, and rules of the form

$$(\mathrm{ok}, \ldots, \mathrm{ok}) \xrightarrow{(a, b_1, \ldots, b_n)} \begin{cases} \mathrm{notok} & \text{when } b_i = 1 \text{ but } b_j = 0 \\ \mathrm{ok} & \text{otherwise} \end{cases}$$

$$(\ldots, \mathrm{notok}, \ldots) \xrightarrow{(a, b_1, \ldots, b_n)} \mathrm{notok}$$

- **Case** $\psi = X_i \subseteq L_\sigma$ is similar.

Does MSO-definability imply regularity?

New question: Is Language$(\psi; X_1, \ldots, X_n)$ regular for every MSO_0 formula ψ?

Answer: Yes, by induction on ψ:

- **Case** $\psi = \mathrm{Sing}(X_i)$.

  Construct A_ψ with $\mathrm{States}(A_\psi) = \{0, 1, \mathrm{many}\}$, $\mathrm{Final}(A_\psi) = \{1\}$, and rules of the form

$$(q_1, \ldots, q_k) \xrightarrow{(a, b_1, \ldots, b_n)} \begin{cases} 0 & \text{when every } q_l = 0 \text{ and } b_i = 0 \\ 1 & \text{when every } q_l = 0 \text{ and } b_i = 1 \\ 1 & \text{when exactly one } q_l = 1 \text{ and } b_i = 0 \\ \mathrm{many} & \text{otherwise} \end{cases}$$

Does MSO-definability imply regularity?

New question: Is Language$(\psi; X_1, \ldots, X_n)$ regular for every MSO_0 formula ψ?

Answer: Yes, by induction on ψ:

- **Case** $\psi = E(X_i, X_j)$. Construct A with States$(A) = \{0,1\} \times \{\mathrm{ok}, \mathrm{notok}\}$,

  Final$(A) = \{0,1\} \times \{\mathrm{ok}\}$, and rules of the form

$$\varepsilon \xrightarrow{(a,b_1,\ldots,b_n)} (b_j, \mathrm{notok})$$

$$(q_1, \ldots, q_k) \xrightarrow{(a,b_1,\ldots,b_n)} \begin{cases} (b_j, \mathrm{ok}) & \text{if } b_i = 1 \text{ and some } q_l = (1, \cdot) \\ (b_j, \mathrm{ok}) & \text{if some } q_l = (\cdot, \mathrm{ok}) \\ (b_j, \mathrm{notok}) & \text{otherwise} \end{cases}$$

  Then Language(Sing$(X_i)) \cap$ Language(Sing$(X_j)) \cap$ Language(A) equals Language$(\psi; X_1, \ldots, X_n)$. The former is regular since regular languages are closed under intersection.

Does MSO-definability imply regularity?

New question: Is Language($\psi; X_1, \ldots, X_n$) regular for every MSO_0 formula ψ?

Answer: Yes, by induction on ψ:

- **Case** $\psi = X_i < X_j$.

Construct A with States$(A) = \{0,1\}^2 \times \{\mathrm{ok}, \mathrm{notok}\}$, Final$(A) = \{0,1\}^2 \times \{\mathrm{ok}\}$, and rules of the form

$$(q_1, \ldots, q_k) \xrightarrow{(a, b_1, \ldots, b_n)} \begin{cases} (b_i, b_j, \mathrm{ok}) & \text{if some } q_l = (1, \cdot, \cdot) \\ & \text{and } q_{l+1} = (\cdot, 1, \cdot) \\ (b_i, b_j, \mathrm{ok}) & \text{if some } q_l = (\cdot, \cdot, \mathrm{ok}) \\ (b_i, b_j, \mathrm{notok}) & \text{otherwise} \end{cases}$$

Then Language(Sing(X_i)) $\cap$ Language(Sing(X_j)) $\cap$ Language(A) equals Language$(\psi; X_1, \ldots, X_n)$. The former is regular since regular languages are closed under intersection.

Does MSO-definability imply regularity?

New question: Is Language$(\psi; X_1, \ldots, X_n)$ regular for every MSO_0 formula ψ?

Answer: Yes, by induction on ψ:

- **Case** $\psi = \psi_1 \wedge \psi_2$.

  Construct A_1 for ψ_1 and A_2 for ψ_2. Clearly,

  $$\mathrm{Language}(A_1) \cap \mathrm{Language}(A_2) = \mathrm{Language}(\psi_1 \wedge \psi_2; X_1, \ldots, X_n).$$

  The former is regular since regular languages are closed under intersection.

Does MSO-definability imply regularity?

New question: Is Language$(\psi; X_1, \ldots, X_n)$ regular for every MSO_0 formula ψ?

Answer: Yes, by induction on ψ:

- **Case** $\psi = \neg\psi_1$.

  Language$(\psi; X_1, \ldots, X_n)$ is the complement of Language$(\psi_1; X_1, \ldots, X_n)$.

Does MSO-definability imply regularity?

New question: Is Language$(\psi; X_1, \ldots, X_n)$ regular for every MSO_0 formula ψ?

Answer: Yes, by induction on ψ:

- **Case** $\psi = \exists Y\, \psi_1$.

  Adapt automaton A for Language$(\psi_1; Y, X_1, \ldots, X_n)$ by replacing every

  $$(q_1, \ldots, q_k) \xrightarrow{(a,b,b_1,\ldots,b_n)} q \qquad \text{by} \qquad (q_1, \ldots, q_k) \xrightarrow{(a,b_1,\ldots,b_n)} q$$

  Observe that this makes A **non-deterministic**

MSO-definability implies regularity!

Theorem in conclusion

Every MSO-definable ranked regular tree language is regular. That is, for every MSO sentence ϕ there exists a tree automaton A with

$$\text{Language}(A) = \text{Language}(\phi).$$

Observe that A can effectively be computed given ϕ!

Size?

Question: the automaton A_ϕ we construct for ϕ is

(a) as big as ϕ
(b) smaller than ϕ
(c) bigger than King Kong
(d) a character of Star Trek

Size?

Question: the automaton A_ϕ we construct for ϕ is

(a) as big as ϕ
(b) smaller than ϕ
(c) bigger than King Kong
(d) a character of Star Trek

Answer: a lot bigger than King Kong!

Stockmeyer and Meyer [1973]: For every n there exists

$$\text{MSO}_0 \text{ formula } \psi = \exists X_1 \neg \exists Y_1 \exists X_2 \neg \exists Y_2 \ldots \exists X_n \neg \exists Y_n \psi'$$

with ψ' quantifier free, such that for every A recognizing the same language as ψ we have:

$$\text{size}(A) \geq \left. 2^{2^{\cdots^{2^{\text{size}(\psi)}}}} \right\} n \text{ times}$$

An MSO Normal Form

By inspection of our construction:

$$\phi \quad \rightarrow \quad A \quad \rightarrow \quad \exists Q_1, \ldots, \exists Q_n \phi'$$

with ϕ' first order.

Immediate corollary

Every MSO formula ϕ is equivalent to a formula $\exists Q_1, \ldots, \exists Q_n \phi'$ with ϕ' first order.