# Tree Automata and Rewriting

Ralf Treinen

Université Paris Diderot
UFR Informatique
Laboratoire Preuves, Programmes et Systèmes

treinen@pps.jussieu.fr

July 23, 2010

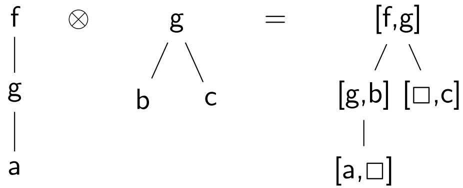
# What happened at the last episode

- Generalization of word automata to trees:
  Rules $q(f(x_1, \ldots, x_n)) \rightarrow f(q_1(x_1), \ldots, q_n(x_n))$
- Closure and decision results as for word automata (beware of non-linearity when generalizing from words to trees)
- Can even be extended to the case of infinite trees

# Relating automata and logic

- A predicate-logic formula $\phi(x_1, \ldots, x_n)$, in a fixed interpretation, denotes a set of $n$-tuples of values: the solutions of the formula.
- A tree automata defines a set of trees.
- A tuple of trees can be encoded as one tree (will be explained soon).
- If we find an encoding of values as trees then we can use a tree automaton to represent a set of tuples of values.
- Use good closure and decision properties of automata to decide validity of formulas in a given interpretation.

# Example: encoding a pair of trees as a tree



# Tuple signatures

Given a signature $\Sigma$, $n \geq 0$ and $\Box \notin \Sigma$,
define $\Sigma_n^{\Box} = \{(f_1, \ldots, f_n) \mid f_i \in \Sigma \cup \{\Box\}\} - \{(\Box, \ldots, \Box)\}$

$$arity((f_1, \ldots, f_n)) = \max\{arity(f_i) \mid f_i \neq \Box\}$$

# Convolution of trees

Given $t_1, \ldots, t_n \in T(\Sigma)$.
Define their convolution $t = t_1 \otimes \cdots \otimes t_n \in T(\Sigma_n^{\square})$ by

- $O(t) = O(t_1) \cup \ldots \cup O(t_n)$
- $t(\pi).i = \begin{cases} t_i(\pi) & \text{if } \pi \in O(t_i) \\ \square & \text{if } \pi \notin O(t_i) \end{cases}$

# Automatic Representation

An automatic representation of a relational structure $\mathcal{A}$ with predicate symbols $R_1, \ldots, R_r$ is given by:

- a finite signature $\Sigma$
- a regular language $L_\delta \subseteq T(\Sigma)$
- an onto function $\nu\colon L_\delta \to \mathcal{A}$
- regular languages $L_i \subseteq T(\Sigma_n^\square)$, $1 \leq i \leq r$, $n = arity(R_i)$, such that all $x_1, \ldots, x_n \in L_\delta$:

$$x_1 \otimes \ldots \otimes x_n \in L_i \text{ iff } (\nu(x_1), \ldots, \nu(x_n)) \in R_i^{\mathcal{A}}$$

A structure is automatic if it has an automatic representation.

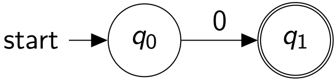
# Example: Presburger Arithmetic

- Presburger Arithmetic: Natural numbers with addition only (no multiplication).
- Presburger (student of Tarski) 1929: Decidability of FO-theory by quantifier elimination.
- Büchi 1960: Decidability by coding in logic WS1S (will be explained later) which is shown to be automatic.
- Boudet&Comon 1996: Direct construction of automatic representation.

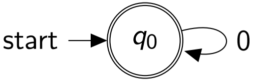
# Automatic Presentation of Presburger Arithmetic

- Structure must be purely relational.
- Choose set of two predicates: $x_1 = 0$ and $x_1 + x_2 = x_3$.
- Choose signature $\Sigma_1 = \{0, 1\}$, $\Sigma_0 = \{\epsilon\}$ (words!). Idea: represent a natural number in binary notation.
- Least or most significant bit first? Least significant bit first, since bits must be aligned for the addition operation!
- Define an onto function $\nu : T(\Sigma) \to \mathbb{N}$: natural interpretation of binary notation.
- $L_\delta = 0 + (0 + 1)^* 1$ (written as regular expression over words)

# Automaton for $x_1 = 0$

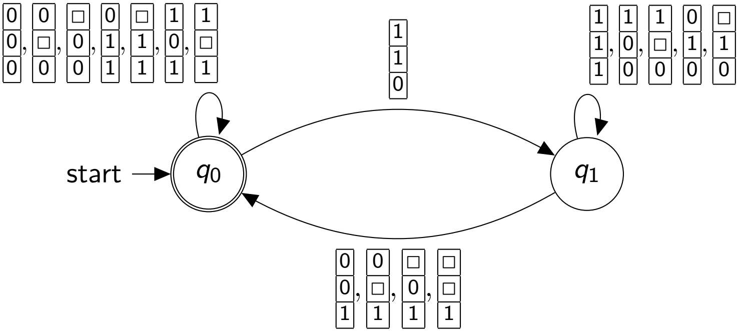



An even simpler automaton?




We only care for $L_\delta$, everything outside $L_\delta$ is junk!

# Automaton for $x_1 + x_2 = x_3$



# FO theory of automatic structures

Büchi 1960, Blumensath&Grädel 2000:

*The first-order theory of any automatic structure is decidable.*

Proof: construct inductively, for any formula $\phi(x_1, \ldots, x_n)$ an automaton $A_\phi$ such that for all $x_1, \ldots, x_n \in L_\delta$ :

$$x_1 \otimes \ldots \otimes x_n \in L_{A_\phi} \text{ iff } (\nu(x_1), \ldots, \nu(x_n)) \in \phi^A$$

# Inductive Construction of $A_\phi$

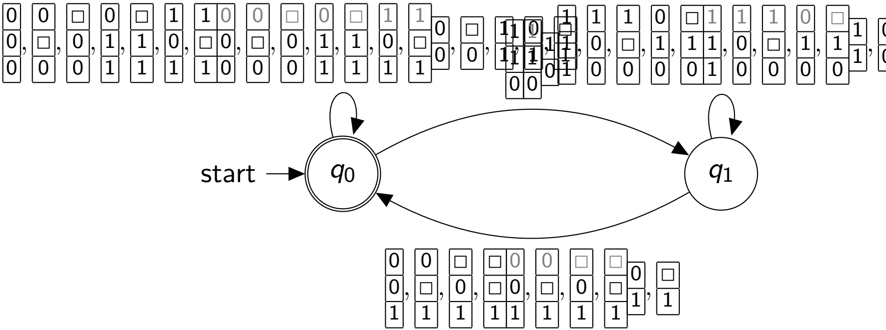
- Base case: $\phi(x_1, \ldots, x_n)$ is a literal $R(x_1, \ldots, x_n)$:
  Automaton $A_\phi$ exists by definition of automatic structures!
- Negation: If $A_\phi$ is the automaton for $\phi(x_1, \ldots, x_n)$:
  then one possible automaton for $A_{\neg\phi}$ is the complement automaton of $\mathcal{A}_\phi$ which recognizes $T(\Sigma^n_\square) \setminus L(A_\phi)$.
  (There may be other automata which differ in the handling of junk.)

# Inductive Construction in case of $\exists$

- Let $A\phi$ be an automaton for $\phi(x_1, \ldots, x_{n+1})$.
- Language recognized by $A_{\exists x_{n+1}\phi}$ ?
- One "forgets" simply the $i+1$-th component in the symbol (projection).
- Linear tree homomorphism: maps $(f_1, \ldots, f_n, f_{n+1})$ to term $(f_1, \ldots, f_n)(x_1, \ldots, x_i)$.
- Use simply the fact that recognizable languages are closed under linear tree homomorphisms!

# Example Projection

Automaton for $\exists x_1 (x_1 + x_2 = x_3)$:



Does this automaton correspond to $x_2 \leq x_3$?

# Inductive Construction in case of $\wedge$

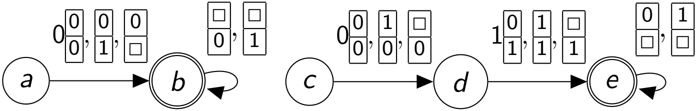
- If $A_1$ is the automaton for $\phi_1$ and $A_2$ the automaton for $\phi_2$, then the automaton for $\phi_1 \wedge \phi_2$ must accept $L(A_1) \cap L(A_2)$, right ?
- If $A_1$ is the automaton for $\phi_1(x_1)$ and $A_2$ the automaton for $\phi_2(x_2)$, then the automaton for $\phi_1(x_1) \wedge \phi_2(x_2)$ must accept $L(A_1) \cap L(A_2)$, right ?
- Of course not in general. We must first assure that both formulas "talk" about the same variables.
- $\phi_1$ and $\phi_2$ must first be "lifted" to the same set of variables $\{x_1, x_2\}$. Only then one can construct the automaton by intersection.

# Cylindrification

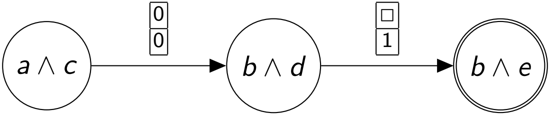
- Here: Given $A$ for $n$ variables, cylindrify to $A^{\uparrow}$ by adding a "bogus" $n+1$-th variable:
- This is exactly the inverse operation of projection, which is described by a tree homomorphism.
- One uses the fact that recognizable languages are closed under inverse tree homomorphisms!

# Example Cylindrification

Automata for $x_1 = 0$ and $x_2 = 2$ cylindrified to $\{x_1, x_2\}$:




Product of the two automata (intersection of languages):



# Finishing up the proof

- Automaton for a closed formula $\phi$ : $\mathcal{A}_\phi$ over alphabet $\Sigma_0^\square$.
- Alphabet $\Sigma_0^\square = ?\emptyset$, since this alphabet contains only tuples with at least one non-blank component!
- Possible languages over alphabet $\emptyset$ ? : $\emptyset$ and $\{\epsilon\}$ !
- $\phi$ is true iff $A_\phi$ recognizes $\{\epsilon\}$
- $\phi$ is false iff $A_\phi$ recognizes $\emptyset$

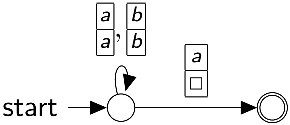
# Exercises on Automatic structures

1. Any automatic structure $\mathcal{A}$ containing the equality relation has an automatic presentation with a one-to-one function $\nu$.
2. For any automatic structure, the theory of the first-order logic extended by the quantifier $\exists^\infty$ is decidable.
   $\exists^\infty x$: there exist infinitely many $x$ such that ...

Solutions: Blumensath&Grädel 2000 paper

# Application 1: Words

- Structure $\{a, b\}^*$, with relations:
  $x_1 = x_2 a$, $x_1 = x_2 b$, $x_1 = a x_2$, $x_1 = b x_2$
- Automatic presentation: $L_\delta = \{a, b\}*$, $\nu = \mathrm{id}$
- Automaton for $x_1 = x_2 a$:




- Automaton for $x_1 = a x_2$: exercise (easy)!
- FO-theory decidable (but not for $x_1 = x_2 x_3$!)

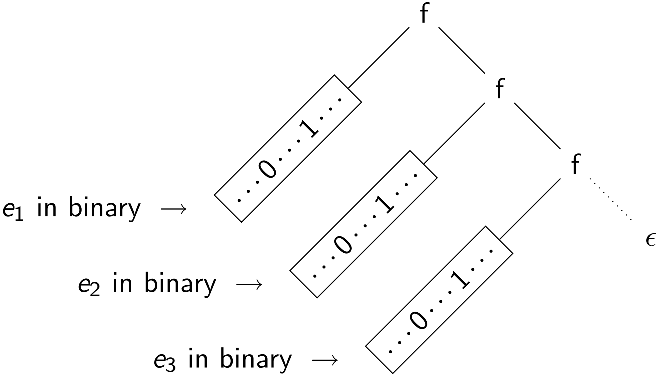
# Application 2: Skolem Arithmetic

- Structure $\mathbb{N}_+\{1,2,3,\ldots\}$, with relations: $x_1 = x_2$, $x = c$ $(c \in \mathbb{N})$, $x_1 * x_2 = x_3$.
- Challenge: find a representation that allows to express multiplication by an automaton!
- Enumeration of prime numbers: $p_1, p_2, p_3, \ldots$
- Represent $n$ as $(e_1, \ldots, e_i)$ where

$$n = p_1^{e_1} * p_2^{e_2} * \ldots * p_i^{e_i}$$

- Multiplication translates to addition of exponents!

# Representation of a number $n = p_1^{e_1} * p_2^{e_2} * p_3^{e_3} * \ldots$



# Application 2: Skolem Arithmetic

- The automaton for $x_1 = x_2 = x_3$ travels down the $f$-spine, and verifies for each branch addition (see the automaton construction for Presburger Arithmetic)
- Consequence: The FO-theory of Skolem Arithmetic is decidable.
- Extension by the relation $x_1 = x_2 + 1$ makes the FO-theory undecidable.

# Application 3: FO-theory of a monadic RPO

- Monadic signature: only constants and unary function symbols
- RPO: Recursive Path Ordering (it does not matter which one when the signature is monadic)
- The structure contains $x \cdot t$ for all $t \in T(\Sigma)$, and $x_1 \prec x_2$.
- Automatic presentation uses trees to represent strings.
- See Narendran&Rusinowitch, ICCL 2000.

# Application 4: multiple equivalence relations

- Structure with universe $T(\Sigma)$
- Multiple congruence relations $=_{E_i}$, for equational theories $E_i$.
- Relations $x = f(y, z)$ not allowed (otherwise FO-theory undecidable, even when all equational theories ground)
- For which classes of equational theories can the FO-theory of this structure be decidable?

# Multiple equivalence relations

Problem with decidability proofs by quantifier elimination (simplification procedure by semantic-preserving rewriting):

$$\frac{\exists x (x =_E y \wedge \phi)}{\phi[y \mapsto x]}$$

is correct only when $=_E$ is congruence w.r.t. all relations in $\phi$. This is in general not the case with several equational theories $E_1, E_2, E_3, \ldots$. Quantifier elimination is not modular!

# Generalized Tree Transducers (GTT)

- A GTT is given by two tree automata $\mathcal{A}_1$ and $\mathcal{A}_2$ over the same signature $\Sigma$, and possibly with shared states.
- The GTT $(\mathcal{A}_1, \mathcal{A}_2)$ recognizes the pair $(t, t') \in T(\Sigma) \times T(\Sigma)$ iff there exists a context $C$, terms $t_i, t_i' \in T(\Sigma)$, and states $q_i$ for $1 \leq i \leq n$, such that $t = C[t_1, \dots, t_n]$, $t' = C[t_1', \dots, t_n']$, $t_i \in L(\mathcal{A}_1, q_i)$ and $t' \in L(A_2, q_i)$ for all $1 \leq i \leq n$.

# Example GTT

- Let $t_1 \to t_2$ be a linear rewrite rule with $V(t_1) \parallel V(t_2)$.
- Tree automaton $\mathcal{A}_1$: recognizes set of ground instances of $t_1$.
- Tree automaton $\mathcal{A}_2$: recognizes set of ground instances of $t_2$.
- The GTT $(\mathcal{A}_1, \mathcal{A}_2)$ recognizes $(t, t')$ iff $t$ transforms to $t'$ in one parallel rewrite step.

# Results about GTTs

- Any relation defined by a GTT is recognizable (by a tree automaton).
- The set of GTT-definable relations is closed under union.
- The set of GTT-definable relations is closed under iteration (Kleene star).

# Application of GTT: multiple equivalence relations

- Let $E$ be a set of linear and variable-disjoint equations (no shared variable on lhs and rhs of an equation).
- $\leftrightarrow^{\|}_{E}$ is GTT-definable. Idea: one automaton recognizes instances of lhs, the other instances of rhs of axioms.
- $=_E$ is the reflexive-transitive closure of that relation, hence recognizable.
- This structure is automatic! (with $\nu = \mathrm{id}$), FO-theory hence decidable.

# Application 5: *WS2S*

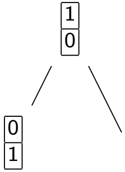
- Weak Second-Order Theory of 2 Successor Functions
- This was the original motivation of Thatcher and Wright to study tree automata
- Two-sorted structure: words $\{0,1\}^*$, and finite sets of words
- Predicates: $x = y \cdot 0$, $x = y \cdot 1$, $x = \epsilon$, $x = y$, $x \in X$.
- FO-theory (even first-order) undecidable with predicate $x = y \cdot z$ (Quine 1946)

# Automatic Presentation of *WS2S*

- Simplify structure: only one sort of finite sets of words.
- Only predicates in the simplified structure:
  $X \subseteq Y$, $S_0(X, Y)$, $S_1(X, Y)$.
- Meaning of $S_0(X, Y)$:
  exists word $w$ with $X = \{w\}$ and $Y = \{w \cdot 0\}$.
- Tree signature is $\Sigma_0 = \{\epsilon\}$, $\Sigma_2 = \{0, 1\}$.
- Tree $t$ represents the set of paths that lead to a 1-node:
  $\nu(t)\{\pi \in O(t) \mid t(\pi) = 1\}$
- One may choose $L_\delta = T(\Sigma)$

# Automatic presentation of the predicates

- $X_1 \subseteq X_2$ : check absence of $\begin{array}{|c|}\hline 1\\\hline 0\\\hline\end{array}$, $\begin{array}{|c|}\hline 1\\\hline \epsilon\\\hline\end{array}$, $\begin{array}{|c|}\hline 1\\\hline \square\\\hline\end{array}$ in the tree.
- $S_0(X_1, X_2)$: Check that tree contains exactly one occurrence of the pattern




and 0, $\epsilon$, $\square$ everywhere else in both components!

# Application 6: $S2S$

- Difference with $WS2S$: sets may be infinite.
- Automatic presentation (with tree automata on infinite trees): exactly as in the finite case.
- Consequence: $S2S$ is decidable.
- Prefix relation can be expressed: $x$ is prefix of $y$ iff

$$\forall S(x \in S \wedge \forall z(x \in S \rightarrow x0 \in S \wedge x1 \in S) \rightarrow y \in S)$$

- Almost all extensions of $S2S$ are undecidable, for instance extension by $|x| = |y|$, extension by suffix relation, or changing $x = y \cdot 1$ into $x = 1 \cdot y$.

# Summary

- Automata can be used (in some cases) to model FO-structures.
- Crucial properties of automata: emptiness decidable, closure under Boolean operations, but also under projection and cylindrification.
- Automata on finite or infinite words or trees can be used.
- Yields decidability of the logic $S2S$, probably the “strongest” known decidability result of a FO theory.

# Literature

- The references of the first lecture
- Achim Blumensath and Erich Grädel: *Automatic Structures*, LICS 2000. Systematic Investigation of automatic structures.
- R.T.: Lecture Notes *Constraint Solving and Decision Problems of FO Theories of Concrete Domains*, chapter 9. See there for detailed references of individual results.