Logic and Automata II

Wolfgang Thomas

RWTH AACHEN

EATCS School Telc, July 2014

The Plan

1. Infinite games
2. Automata on infinite trees
3. Rabin’s Tree Theorem
4. Infinite structures with undecidable MSO-theory
5. Infinite structures with decidable MSO-theory

Infinite Games

Alonzo Church (1903-1995)

APPLICATION OF RECURSIVE ARITHMETIC TO THE PROBLEM OF CIRCUIT SYNTHESIS

Alonzo Church

RESTRICTED RECURSIVE ARITHMETIC

Primitive symbols are individual (i.e., numerical) variables x, y, z, t,..., singulary functional constants $i_1, i_2, \ldots, i_\mu$, the individual constant 0, the accent ' as a notation for <u>successor</u> (of a number), the notation ( ) for application of a singulary function to its argument, connectives of the propositional calculus, and brackets [ ].

Axioms are all tautologous wffs. Rules are modus ponens; substitution for individual variables; mathematical induction,

$$\text{from } P \supset S^{a}_{a'}P| \quad \text{and} \quad S^{a}_{0}P| \quad \text{to infer } P;$$

and any one of several alternative recursion schemata or sets of recursion schemata.

A Citation

Alonzo Church

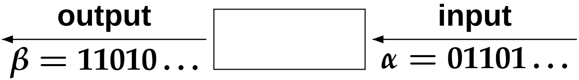
at the "Summer Institute of Symbolic Logic"

Cornell University, 1957:

"Given a requirement which a circuit is to satisfy, we may suppose the requirement expressed in some suitable logistic system which is an extension of restricted recursive arithmetic. The *synthesis problem* is then to find recursion equivalences representing a circuit that satisfies the given requirement (or alternatively, to determine that there is no such circuit)."

(By "circuits", Church means finite automata with output.)

Precise Formulation



Player 1: $a_0 \quad a_1 \quad a_2 \quad a_3 \cdots = \alpha$

Player 2: $b_0 \quad b_1 \quad b_2 \quad b_3 \cdots = \beta$

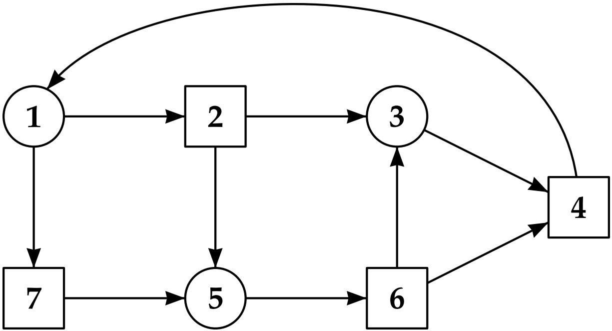
Consider this as a game between two players, called 1 and 2.
A play is a sequence $a_0 b_0 a_1 b_1 a_2 b_2 \ldots$.

A winning condition specifies for a play when Player 2 wins the play.

Questions:

- **Does one of the two players have a winning strategy?**
- **Can one decide who wins?**
- **Can one exhibit a winning strategy?**

Muller Games: Arena and Muller Condition



A **Muller game** is given as a pair $(G, \mathcal{F})$ of an arena $G = (V, V_1, V_2, E)$ with $\mathcal{F}$, a family of subsets of V.

A play is an infinite path through G.
Player 2 wins a play if the vertices visited infinitely often form a set in $\mathcal{F}$.

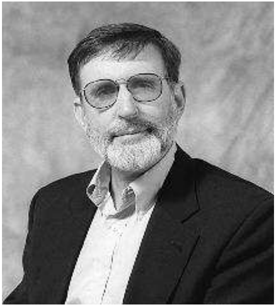
Solution of Church's Problem

Büchi-Landweber Theorem (1969)

For each finite-state Muller game with a designated initial vertex:

- **either Player 1 or Player 2 has a winning strategy (i.e., the game is "determined"),**
- **it is decidable who wins,**
- **and a finite-state winning strategy for the respective winner can be computed.**

J.R. Büchi



L.H. Landweber

Applications

- **Complementation results from determinacy**
- **Model Checking (μ-calculus)**
- **Controller synthesis**

Some Remarks

1. If no initial vertex is specified, the question who wins is generalized: What are the winning regions of the two players?
   (These collect the vertices from where the player under consideration has a winning strategy.)

2. In Muller games, winning strategies may need memory.
   (If one proceeds to the game tree, full memory is built into the positions and memoryless strategies suffice.)

3. If memory is not needed (and the current position suffices to determine the next move), we speak of a positional strategy.

Parity Condition

We assume a coloring $c : V \to \{1, \dots, k\}$ of the game graph.

A play $\varrho \in V^\omega$ satisfies the parity condition iff the maximal color occurring infinitely often in ϱ is even.

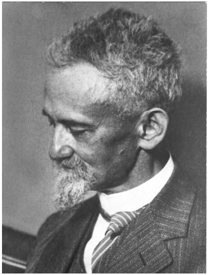
Formally: max(Inf$(c(\varrho))$) is even

$$\bigvee_{j \text{ even}} (\exists^\omega i : c(\varrho(i)) = j \ \wedge \neg \exists^\omega i : c(\varrho(i)) > j)$$

A parity game is given by a game graph with finite coloring and the parity condition as winning condition for player 2.

History

The parity condition goes back to F. Hausdorff 1914



Felix Hausdorff (1868-1942)

and was re-introduced as “Rabin chain condition“
by A.W. Mostowski 1985,
rediscovered as “parity condition” by Emerson and Jutla 1991

Reducing Muller to Parity Games

Add extra memory which allows in a play ϱ to capture the set $\mathrm{Inf}(\varrho)$ of states visited infinitely often in ϱ:

Keep a list of the states in the order of their last visits.

Update by

- **shifting the current state to front**
- **marking the position where the current state was taken from.**

Latest Appearance Record

Visited state	LAR
A	$\underline{A}BCD$
C	$CA\underline{B}D$
C	$\underline{C}ABD$
D	$DCA\underline{B}$
B	$BDC\underline{A}$
D	$D\underline{B}CA$
C	$CD\underline{B}A$
D	$D\underline{C}BA$
D	$\underline{D}CBA$

Analyzing the LAR

The maximal position m marked again and again is the number of states visited infinitely often.

and the set of states seen infinitely often is then listed up to this position m.

We color an LAR by $2k$ if k is the marked position and the states listed up to position k belongs to the Muller acceptance system $\mathcal{F}$, otherwise we use color $2k-1$.

This allows to reformulate the Muller winning condition:

The highest LAR-color occurring infinitely often is even

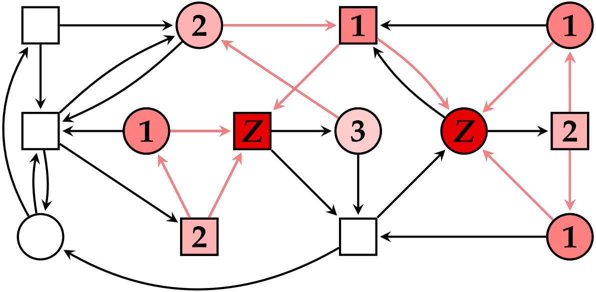
So we can change an arena over V into an arena over LAR(V) while replacing the Muller winning condition by the parity condition.

Positional Determinacy of Parity Games

Theorem (Emerson-Jutla 1991)

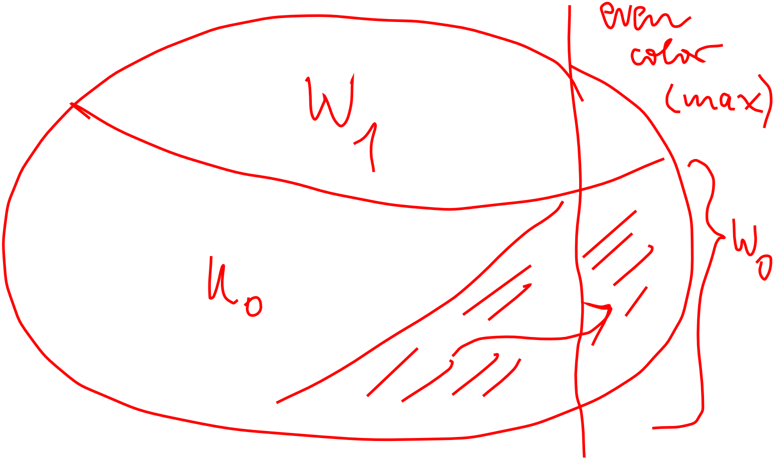
- **Parity games (even over infinite arenas) are determined, and the winner from a given vertex has a positional winning strategy.**
- **Over finite arenas, the winning regions and winning strategies of the two players can be computed.**

Simple Case: Reachability



Positional Determinacy of Parity Games

by induction on number of colors, say with highest color even



Finite Arenas: Computable Solution

For Player 2:

1. **Guess the winning regions and positional strategies over them.**
2. **Check that these strategies are winning for the respective player.**

Step 2 can be done in polynomial time: Analyze the loops that the other player can realize.

The problem to determine the winner from a given vertex of a parity game is in NP $\cap$ co-NP.

Parity games are treated in detail in Nir Piterman's tutorial.

Automata on Infinite Trees

The Model T_2 and S2S

The structure of the infinite binary tree is

$$T_2 = (\{0,1\}^*, S_0, S_1, \varepsilon)$$

where S_i is the i-th successor function:

$$S_0(u) = u0, \quad S_1(u) = u1$$

We study the MSO-theory of T_2.

Our aim is

Rabin's Tree Theorem: The MSO-theory of T_2 is decidable.

Michael O. Rabin

Honorary Doctorate at University of Wroclaw (2007)

Example Formulas

Definition of $x \preceq y$ ("node x is prefix of node y"):

$\varphi_s^*(x,y)$ **with** $\varphi_s(z,z') := z0 = z' \vee z1 = z'$

$\mathrm{Chain}(X)$ **("X is linearly ordered by $\preceq$"):**

$$\forall x \forall y((X(x) \wedge X(y)) \to (x \preceq y \vee y \preceq x))$$

$\mathrm{Path}(X)$ **("X is a path, i.e. a maximal chain"):**

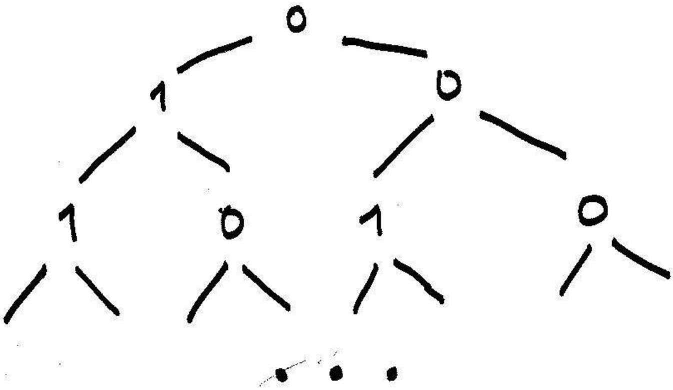
$$\mathrm{Chain}(X) \wedge \neg \exists Y(X \subseteq Y \wedge X \neq Y \wedge \mathrm{Chain}(Y))$$

$X \subseteq Y$**:** $\forall z(X(z) \to Y(z))$

$X = Y$**:** $\forall z(X(z) \leftrightarrow Y(z))$

Further definable relations: "x is lexicographically before y", "X is finite"

A labelled tree



General Format of S2S-Formulas

A formula $\varphi(X_1, \ldots, X_n)$ defines a set of $\{0,1\}^n$-labelled trees.

Example:

$\varphi_0(X_1)$ might express

“there is a path on which infinitely many X_1-elements are located.”

Rabin introduced tree automata equivalent to S2S.

Format of Tree Automata

$\mathcal{A} = (Q, \Sigma, q_0, \Delta, \mathrm{Acc})$ where

$$\Delta \subseteq Q \times \Sigma \times Q \times Q$$

A transition (q, a, q_1, q_2) allows the automaton in state q at an a-labelled node u to proceed to states q_1, q_2 at the two successor nodes $u0, u1$

A parity tree automaton $\mathcal{A} = (Q, \Sigma, q_0, \Delta, c)$ accepts a tree t

if there exists a run ϱ of $\mathcal{A}$ on t such that on each path of ϱ the parity condition is satisfied.

(Similarly for Büchi and Muller acceptance.)

Example

$T_1 = \{t \in T^{\omega}_{\{0,1\}} \mid \exists$**path through** t **with infinitely many** $1\}$

is recognized as follows.

“Guess an appropriate path and on it check that infinitely often 1 occurs on it.”

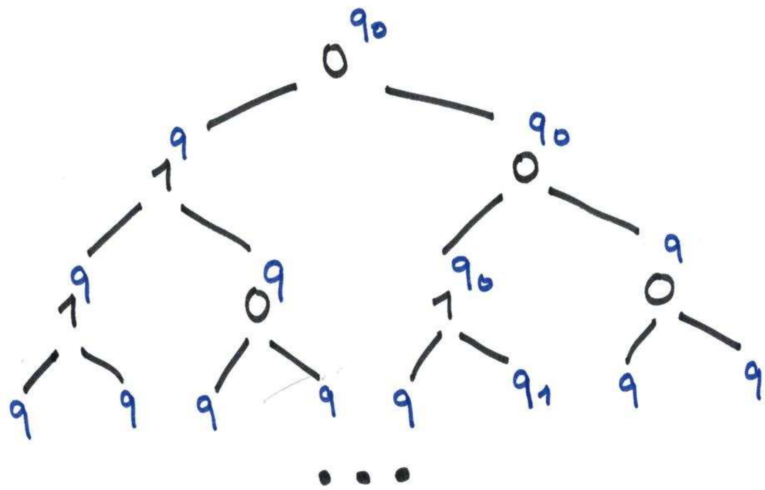
Use states q_0, q_1 **for the path to guessed, otherwise** q**.**

q_0 **is initial state.**

q **has color 0,** q_0 **has color 1, and** q_1 **color 2.**

Transitions: $(q_0, 0/1, q_{0/1}, q)$, $(q_0, 0/1, q, q_{0/1})$,
$(q_1, 0/1, q_{0/1}, q)$, $(q_1, 0/1, q, q_{0/1})$,
finally $(q, a/b, q, q)$

A Run



Rabin's Tree Theorem

(a) A tree language is definable in S2S iff it is recognizable by a parity tree automaton.

(b) The nonemptiness problem for parity tree automata "Given $\mathcal{A}$, does $\mathcal{A}$ accept some tree?" is decidable.

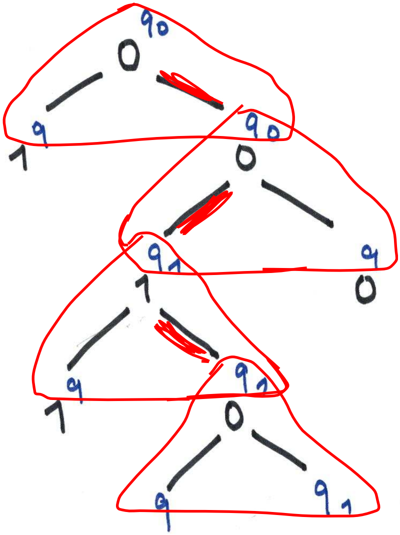
Consequence (from (b) for input-free tree automata):

Rabin's Tree Theorem: MTh(T_2) is decidable.

Everything works as before, but complementation and nonemptiness test are now more difficult.

We use positional determinacy of parity games.

A Play of the Game $\Gamma_{\mathcal{A},t}$



$\Gamma_{\mathcal{A},t}$

- First Automaton picks a transition from Δ which can serve to start a run at the root of the input tree.
- Then Pathfinder decides on a direction (left or right) to proceed to a son of the root.
- Then Automaton chooses again a transition for this node (compatible with the first transition and the input tree).
- and so on in alternation

Automaton wins the play iff the constructed state sequence satisfies the parity condition.

Game Positions

Positions of Automaton are the triples

$($tree node w, tree label $t(w)$, state q at $w)$

Positions of Pathfinder are the triples

$($tree node w, tree label $t(w)$, transition τ at $w)$

We obtain an infinite game graph.

The tree automaton $\mathcal{A}$ accepts the input tree t iff in the parity game $\Gamma_{\mathcal{A},t}$ there is a positional winning strategy for player Automaton from the initial position $(\varepsilon, t(\varepsilon), q_0)$.

Complementation Proof: Outline

Complementation of tree automata means to express the condition that a given automaton $\mathcal{A}$ does not accept t by acceptance of another automaton.

Non-acceptance by $\mathcal{A}$ means non-existence of a winning strategy for Automaton in $\Gamma_{\mathcal{A},t}$.

Determinacy implies existence of a winning strategy for Pathfinder.

We convert this strategy into an automaton strategy in a different game $\Gamma_{\mathcal{B},t}$.

This gives the desired complement automaton $\mathcal{B}$.

Applying Determinacy (Step 1)

Proof: Let $\mathcal{A} = (Q, \Sigma, q_0, \Delta, c)$ be a parity tree automaton.

We find a parity tree automaton $\mathcal{B}$ accepting precisely the trees $t \in T^{\omega}_{\Sigma}$ which are not accepted by $\mathcal{A}$

Start with the following equivalences: For any tree t,

$\mathcal{A}$ does not accept t

iff

Automaton has no winning strategy from the initial position $(\varepsilon, t(\varepsilon), q_0)$ in the parity game $\Gamma_{\mathcal{A},t}$

iff (by Determinacy Theorem)

(*) in $\Gamma_{\mathcal{A},t}$, Pathfinder has a positional winning strategy from $(\varepsilon, t(\varepsilon), q_0)$

Step 2

Reformulate (*) in the form

"$\mathcal{B}$ accepts t" for some tree automaton $\mathcal{B}$

Pathfinder's strategy is a function f from the set $\{0,1\}^* \times \Sigma \times \Delta$ of his vertices into the set $\{0,1\}$ of directions.

Decompose this function into a family

$$(f_w : \Sigma \times \Delta \to \{0,1\})$$

of "local instructions", parameterised by $w \in \{0,1\}^*$

The set I of possible local instructions $i : \Sigma \times \Delta \to \{0,1\}$ is finite,

Thus Pathfinder's winning strategy can be coded by the I-labelled tree s with $s(w) = f_w$

Step 3

Let $s^\wedge t$ be the corresponding $(I \times \Sigma)$-labelled tree with

$s^\wedge t(w) = (s(w), t(w))$ for $w \in \{0,1\}^*$

Now (*) amounts to the following:

There is an I-labelled tree s such that for all sequences $\tau_0\tau_1\ldots$ of transitions chosen by Automaton and for all (in fact for the unique) $\pi \in \{0,1\}^\omega$ determined by $\tau_0\tau_1\ldots$ via the strategy coded by s, the generated state sequence violates the parity condition.

This can be checked by a nondeterministic parity tree automaton $\mathcal{B}$, the desired complement automaton for $\mathcal{A}$.

Equivalence Theorem

A formula $\varphi(X_1, \ldots, X_n)$ defines a set of $\{0,1\}^n$-labelled trees.

A set T of $\{0,1\}^n$-labelled trees is MSO-definable iff it is recognized by a parity tree automaton.

The proof is a copy of the proof for ω-languages, except for the complementation of automata.

The Input-free Case

An S2S-sentence without free variables will lead to an input-free tree automaton.

An input-free parity tree automaton $\mathcal{A} = (Q, q_0, \Delta, c)$ with $\Delta \subseteq Q \times Q \times Q$ defines the simpler game $\Gamma_{\mathcal{A}}$:

Automaton has positions in in Q and chooses transitions from $Q \times Q \times Q$

Pathfinder has positions in Δ and chooses directions.

$\mathcal{A}$ admits at least one successful run iff Automaton has a winning strategy in the game $\Gamma_{\mathcal{A}}$ from position q_0.

This is a parity game on a finite (!) arena; so the condition can be decided effectively.

Rabin's Basis Theorem

Recall: **A nonempty regular ω-language contains an ultimately periodic ω-word.**

A corresponding result holds for nonempty tree languages which are recognized by parity tree automata.

Rabin's Basis Theorem: **A nonempty tree language recognized by a parity tree automaton contains a regular tree.**

A tree $t \in T_\Sigma^\omega$ is called regular if it is "finitely generated" in the following sense:

There is a deterministic finite automaton equipped with output which tells for any given input $w \in \{0,1\}^*$ which label is at node w (i.e. the value $t(w)$).

Rabin's Basis Theorem: Proof

Assume $\mathcal{A} = (Q, \Sigma, q_0, \Delta, c)$ is a parity tree automaton.

Proceed to an "input-guessing" (and input-free) tree automaton $\mathcal{A}'$ with states in $Q \times \Sigma$:

$\mathcal{A}'$ guesses an input tree and works on it as $\mathcal{A}$ does.

$\mathcal{A}'$ may have several initial states.

Then:

The input-free automaton $\mathcal{A}'$ admits a successful run iff $T(\mathcal{A}) \neq \emptyset$, and a tree in $T(\mathcal{A})$ is extracted from the second components of the $\mathcal{A}'$-run.

Thus a regular tree is generated.

Looking Back

Büchi automata, Muller automata, and parity automata over infinite words provide different versions of quantifier complexity:

from Σ^1_n to Σ^1_1, to $\mathrm{Bool}(\Pi^0_2)$.

Tree automata provide a less radical way of quantifier elimination:

"There is a run on the tree given by $X_1, \ldots, X_n$ such that on each path the acceptance condition is satisfied."

In logical terminology this is a Σ^1_2-condition.

Infinite Structures with Undecidable MSO-Theory

Interpretations

An MSO-interpretation of $\mathcal{A}$ in $\mathcal{B}$ is a description of of a copy of $\mathcal{A}$ in $\mathcal{B}$

- **by an MSO-formula $\varphi(x)$ defining the domain of the $\mathcal{A}$-copy in $\mathcal{B}$**
- **for each (say binary) relation R, resp. each (say unary) function of $\mathcal{A}$ a corresponding defining MSO-formula $\psi(x, y)$**

Assume there is an MSO-interpretation of $\mathcal{A}$ in $\mathcal{B}$. Then

- **MSO-Th$(\mathcal{B})$ decidable $\Rightarrow$ MSO-Th$(\mathcal{A})$ decidable.**
- **MSO-Th$(\mathcal{A})$ undecidable $\Rightarrow$ MSO-Th$(\mathcal{B})$ undecidable.**

The Infinite Grid

$G_2 = (\mathbb{N} \times \mathbb{N}, (0,0), S_1, S_2)$

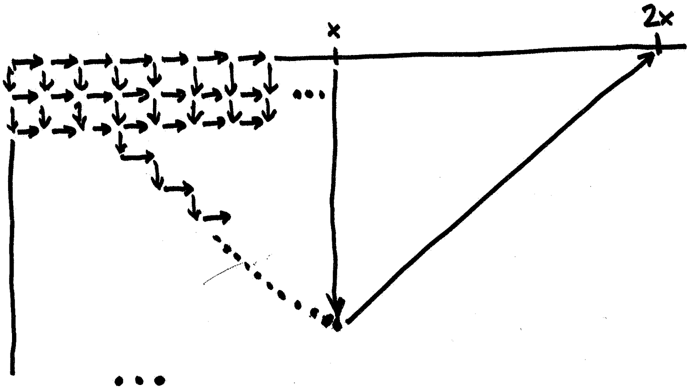
where $S_1(i,j) = (i+1, j), \quad S_2(i,j) = (i, j+1)$

The monadic second-order theory of the infinite grid is undecidable.

Proof:

We recall that MSO-Th$(\mathbb{N}, +1, 2x, 0)$ is undecidable.

We describe $(\mathbb{N}, +1, 2x, 0)$ in G_2.



x
2x

Adding the Equal Level Relation

Consider the binary tree with equal level relation E

$$E(u,v) \quad :\Leftrightarrow \quad |u| = |v|$$

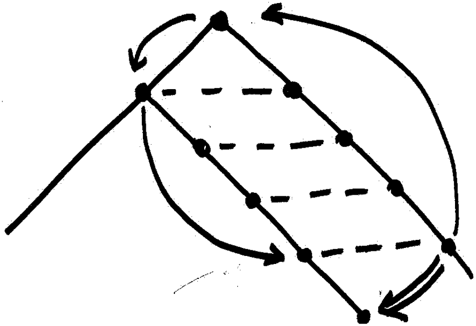
Obtain (T_2, E).

The MSO-theory of (T_2, E) is undecidable.

Proof:

We describe G_2 in (T_2, E).

Proof by Picture



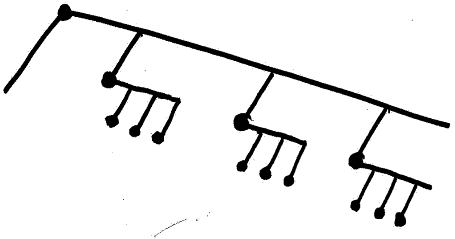
Infinite Structures with Decidable MSO-Theory

A First Example

Show Rabin's Tree Theorem for $T_3 = (\{0,1,2\}^*, S_0^3, S_1^3, S_2^3)$.

Idea: Describe T_3 in T_2:

Consider the T_2-vertices in $(10 + 110 + 1110)^*$.



Interpretation: Details

The element 02211 of T_3 is coded by

1011101110110110 in T_2.

Define the set of codes by

$\varphi(x)$: "x is in the closure of ε under 10-, 110-, and 1110-successors"

Define the 0-th, 1-st 2-nd successors by
$\psi_0(x, y), \psi_1(x, y), \psi_2(x, y)$

Pushdown Graphs

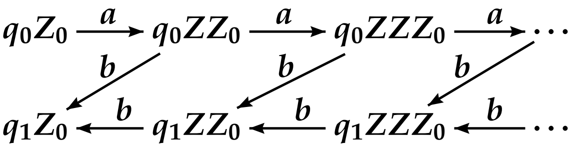
Consider $\mathcal{A}$ for language $L = \{a^n b^n \mid n \geq 0\}$:

$\mathcal{A} = (\{q_0, q_1\}, \{a, b\}, \{Z_0, Z\}, q_0, Z_0, \Delta)$ **with**

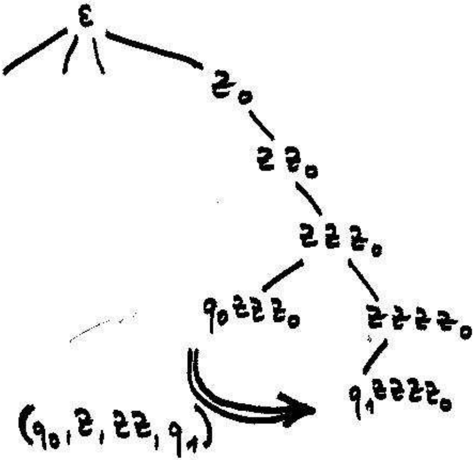
$$\Delta = \left\{ \begin{array}{ll} (q_0, Z_0, a, q_0, ZZ_0), & (q_0, Z, a, q_0, ZZ), \\ (q_0, Z, b, q_1, \varepsilon), & (q_1, Z, b, q_1, \varepsilon) \end{array} \right\}$$

Initial and final configuration: $q_0 Z_0$

The associated pushdown graph (of reachable configurations only) is:



Interpreting a PDG in a Tree T_n



Formal Details

A pushdown graph is MSO-interpretable in T_2

Given pushdown automaton $\mathcal{A}$ with stack alphabet $\{1, \ldots, k\}$ and states $q_1, \ldots, q_m$.

Let $G_{\mathcal{A}} = (V_{\mathcal{A}}, E_{\mathcal{A}})$ be the corresponding PD graph.
$n := \max\{k, m\}$

Find an MSO-interpretation of $G_{\mathcal{A}}$ in T_n.

Represent configuration $(q_j, i_1 \ldots i_r)$ by the vertex $i_r \ldots i_1 j$.

$\mathcal{A}$-steps lead to local moves in T_n.

E.g. a push step from vertex $i_r \ldots i_1 j$ to $i_r \ldots i_1 i_0 j'$.

These edges are easily definable in MSO.

Hence: The MSO-theory of a PD graph is decidable.

Unfoldings

Given a graph $(V, (E_a)_{a \in \Sigma}, (P_b)_{b \in \Sigma'})$

the unfolding of G from a given vertex v_0 is the following tree $T_G(v_0) = (V', (E'_a)_{a \in \Sigma}, (P'_b)_{b \in \Sigma'})$:

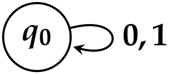
- V' consists of the vertices $v_0 a_1 v_1 \ldots a_r v_r$ with $(v_{i-1}, v_i) \in E_{a_i}$,
- E'_a contains the pairs $(v_0 a_1 v_1 \ldots a_r v_r, v_0 a_1 v_1 \ldots a_r v_r a v)$ with $(v_r, v) \in E_a$,
- P'_b the vertices $v_0 a_1 v_1 \ldots a_r v_r$ with $v_r \in P_b$.

Unfolding Preserves Decidability

Theorem (Muchnik, Courcelle/Walukiewicz)

If the MSO-theory of G is decidable and v_0 is an MSO-definable vertex of G, then the MSO-theory of $T_G(v_0)$ is decidable.

An innocent example:



Caucal's Proposal

We have now two processes which preserve decidability of MSO-theory:

- **interpretation (transforming a tree into a graph)**
- **unfolding (transforming a graph into a tree)**

Let us apply them in alternation!

We obtain the Caucal hierarchy or pushdown hierarchy.

Definition

- $\mathcal{T}_0 =$ the class of finite trees
- $\mathcal{G}_n =$ the class of graphs which are MSO-interpretable in a tree of $\mathcal{T}_n$
- $\mathcal{T}_{n+1} =$ the class of unfoldings of graphs in G_n

Each structure in the pushdown hierarchy has a decidable MSO-theory.

Nontrivial fact:

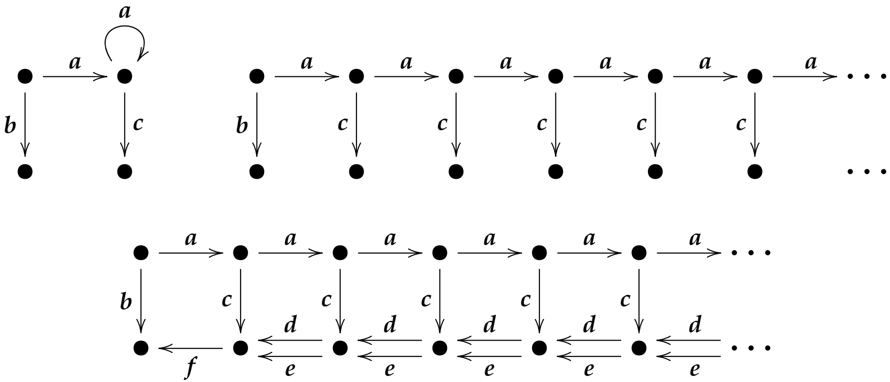
The sequence $\mathcal{G}_0, \mathcal{G}_1, \mathcal{G}_2, \ldots$ is strictly increasing.

The First Levels

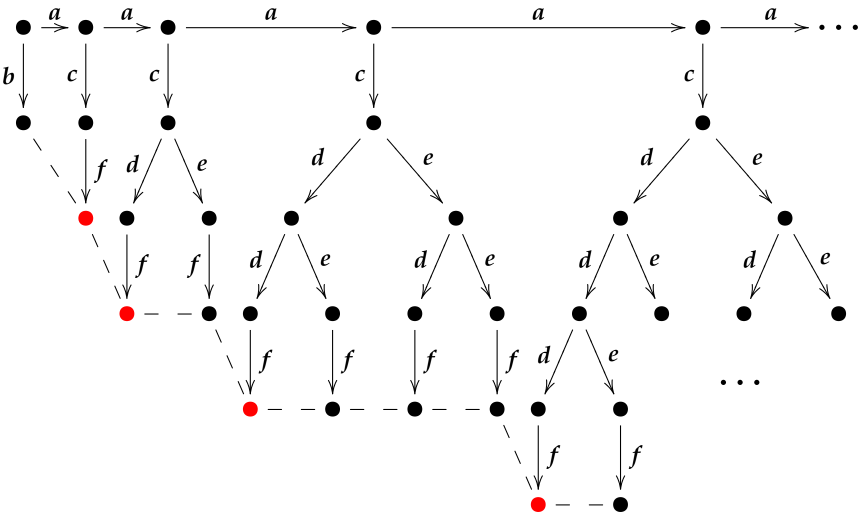
- $\mathcal{T}_0$ **is the class of finite trees.**
- $\mathcal{G}_0$ **is the class of finite graphs.**
- $\mathcal{T}_1$ **is the class of regular trees.**
- $\mathcal{G}_1$ **is the class of prefix-recognizable graphs (including the pushdown graphs)**

Higher levels are not yet well understood.

A Finite Graph, a Regular Tree, a PD Graph



Unfolding Again



Looking at the Bottom Line

The sequence of leaves defines a copy of the successor structure of the natural numbers.

The red points give the numbers that are a power of 2.

Altogether we get $(\mathbb{N}, +1, 0, \mathrm{Pow}_2)$ as a structure in $\mathcal{G}_2$.

Similar constructions give more structures of this form, e.g. with the factorial predicate replacing Pow_2.

On Structures $(\mathbb{N}, +1, 0, P)$ with $P \subseteq \mathbb{N}$

1. $(\mathbb{N}, +1, 0, P)$ belongs to $\mathcal{G}_1$ iff P is ultimately periodic.
2. $(\mathbb{N}, +1, 0, P)$ belongs to $\mathcal{G}_2$ iff P is morphic.

What about $P_{\sqrt{2}}$, P_π, $\mathbb{P}$?

An Example

$P_{\exp}$:= set of iterated 2-powers 1, 2, 2^2, 2^{2^2}, etc.

1. $(\mathbb{N}, +1, 0, P_{\exp})$ is outside the pushdown hierarchy.
2. MSO-Th$(\mathbb{N}, +1, 0, P_{\exp})$ is decidable.

Method for 2:

The MSO-theory of $(\mathbb{N}, +1, 0, P)$ is decidable iff the following "acceptance problem for Büchi automata" is decidable:

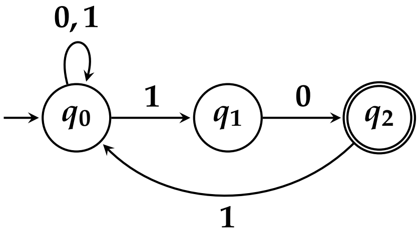
Given a Büchi automaton $\mathcal{A}$ over $\{0, 1\}$, does $\mathcal{A}$ accept α_P?

(Here α_P is the 0-1-sequence with $\alpha(i) = 1$ iff $i \in P$.)

On the Prime Predicate

Consider $\alpha_{\mathbb{P}} = 00110101000101\ldots$

Does the following Büchi automaton accept $\alpha_{\mathbb{P}}$**?**



Outlook

The pushdown hierarchy is a very rich class of structures all of which have a decidable MSO-theory.

Some open questions:

- **Understand which structures belong to the hierarchy.**
- **Compute the smallest level on which a structure occurs.**
- **Find ways to extend the hierarchy while keeping decidability of the MSO-theory.**
- **How to obtain a richer class of models when restricting to a weaker logic, e.g. monadic transitive logic?**