Logic and Automata III: Tree Automata and Decidability of Monadic Theories

Wolfgang Thomas

RWTH AACHEN

Computability in Europe, Budapest, June 2014

Automata on Infinite Trees

The Model T_2 and S2S

The structure of the infinite binary tree is

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

where S_i is the i-th successor function:

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

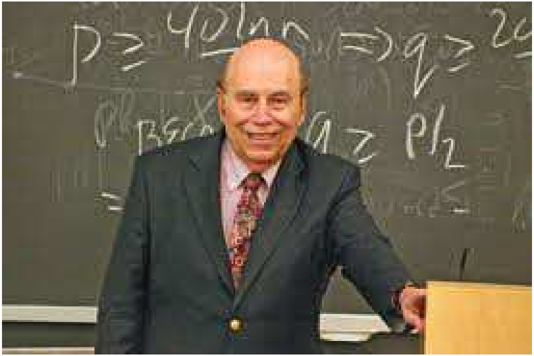
The theory S2S is set of S2S-sentences which are true in $\underline{T_2}$

It is also called the monadic theory of the binary tree,

denoted by $\mathrm{MTh}_2(\underline{T_2})$

Our aim is

Rabin's Tree Theorem: The MSO-theory of $\underline{T_2}$ ist decidable.



Michael O. Rabin

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 $\preceq$"):

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

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

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

$X \subseteq Y$: $\forall z(X(z) \rightarrow Y(z))$

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

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

General Format of S2S-Formulas

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

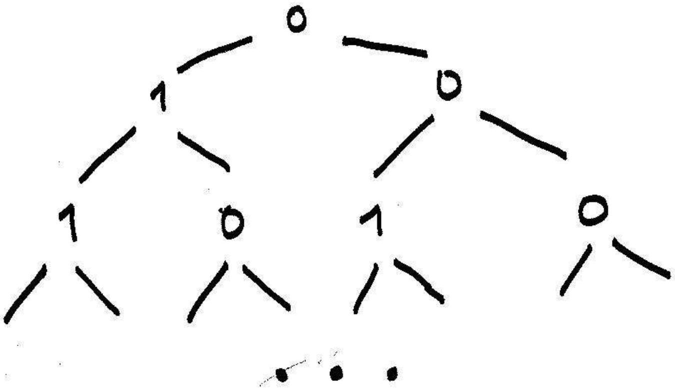
Example:

$\varphi_0(X_1)$ might express

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

Rabin introduced tree automata equivalent to S2S.

A labelled tree



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."

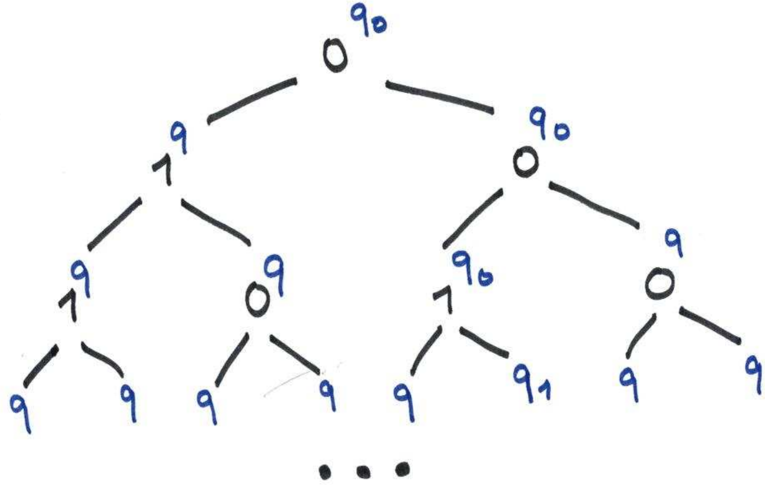
Use states q_0, q_1 **for the path to guessed, otherwise** q**.**

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.

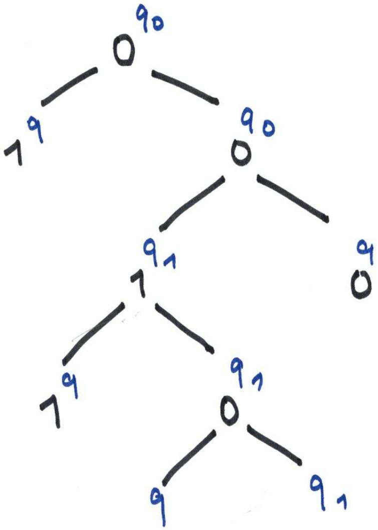
Consequence (from (b) for input-free tree automata):

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).

Then Pathfinder reacts again by branching left or right from the momentary node, etc.

Play gives a sequence of transitions (and hence a state sequence from Q), built up along a path chosen by Pathfinder.

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 $w)$

By choice of a transition τ of the form $(q, t(w), q', q'')$, a position of Pathfinder is reached.

Positions of Pathfinder are the triples

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

These positions with the moves define an infinite game graph.

Run Lemma: **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 $t \in T_\Sigma^\omega$ which are not accepted by $\mathcal{A}$

Start with the following equivalences: For any tree t,

$\mathcal{A}$ does not accept t

iff (by Run Lemma)

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

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

Now (*) is equivalent 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 id 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.

Run Lemma (input-free case): $\mathcal{A}$ 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 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:

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.

Undecidability: The Grid

The Infinite Grid

$G_2 = (\mathbb{N} \times \mathbb{N}, (0,0), S_1, S_2)$ where $S_1(i,j) = (i+1,j)$, $S_2(i,j) = (i,j+1)$

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

Proof

by reduction of the halting problem for Turing machines:

For any TM M construct a sentence φ_M of the monadic second-order language of G_2 such that

M halts when started on the empty tape iff $G_2 \models \varphi_M$.

Configurations of M

Assume that M works on a left-bounded tape.

A halting computation of M can be coded by a finite sequence of configuration words

$C_0, C_1, \ldots, C_m$.

We can arrange the configurations row by row in a right-infinite rectangular array:

$$\begin{array}{cccccccc} q_0 & a_0 & a_0 & a_0 & a_0 & a_0 & a_0 & \ldots \\ a_1 & q_1 & a_0 & a_0 & a_0 & a_0 & a_0 & \ldots \\ q_0 & a_1 & a_2 & a_0 & a_0 & a_0 & a_0 & \ldots \\ a_3 & q_2 & a_2 & a_0 & a_0 & a_0 & a_0 & \ldots \end{array}$$

etc.

Describing an M-Run

The sentence φ_M will express over G_2 the existence of such an array of configurations.

$a_0, \ldots, a_n$ are the tape symbols (a_0 is the blank)
$q_0, \ldots, q_k$ are the states of M, special halting state q_s
We use set variables $X_0, \ldots, X_n, Y_0, \ldots, Y_k$
X_i collects the grid positions where a_i occurs,
Y_i collects the grid positions where state q_i occurs.
φ_M : $\exists X_0, \ldots, X_n, Y_0, \ldots, Y_k$ (Partition$(X_0, \ldots, Y_k)$

- $\wedge$ "the first row is the initial M-configuration"
- $\wedge$ "a successor row is the successor configuration of the preceding one"
- $\wedge$ "at some position the halting state is reached")

A Hidden Grid

Consider the binary tree with Equal-Level Predicate E

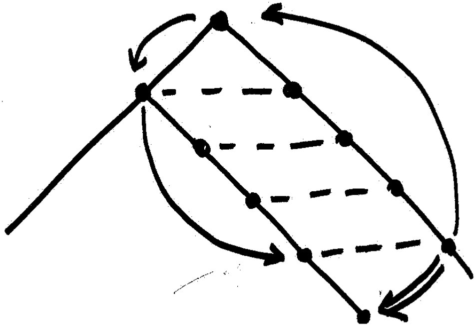
$$E(u, v) \quad :\Leftrightarrow \quad |u| = |v|$$

Obtain (T_2, E).

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

Proof: Use E to define again the grid 0^*1^*.

Proof by Picture



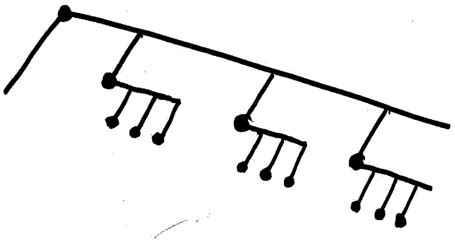
Decidability Results

A First Example

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

Idea: Obtain a copy of T_3 in T_2:

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



Interpretation: Details

The element $i_1 \dots i_m$ of T_3 is coded by

$1^{i_1+1}0 \dots 1^{i_m+1}0$ 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
$\psi_0(x,y), \psi_1(x,y), \psi_2(x,y)$

The structure $(\varphi^{T_2}, (\psi_i^{T_2})_{i=0,1,2}, \varepsilon)$ is isomorphic to T_3.

Pushdown Graphs

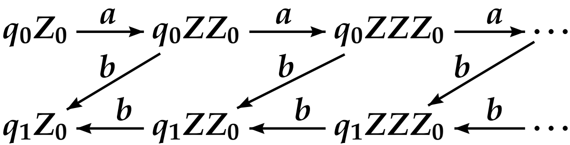
Consider $\mathcal{A}$ for language $L = \{a^n b^n \mid n \geq 0\}$:

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

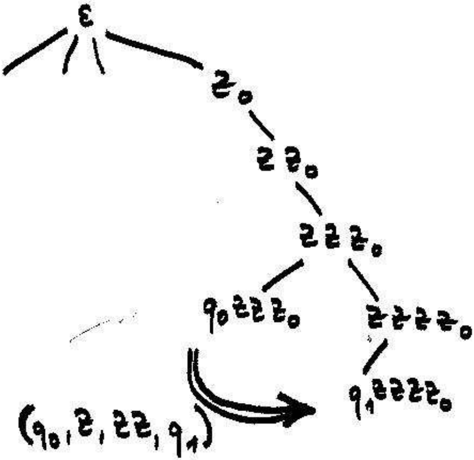
$$\Delta = \left\{ \begin{array}{ll} (q_0, Z_0, a, q_0, ZZ_0), & (q_0, Z, a, q_0, ZZ), \\ (q_0, Z, b, q_1, \varepsilon), & (q_1, Z, b, q_1, \varepsilon) \end{array} \right\}$$

Initial and final configuration: $q_0 Z_0$

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



Interpreting a PDG in T_n



Interpretation: Second Example

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** $T_G(v_0) = (V', (E'_a)_{a \in \Sigma}, (P'_b)_{b \in \Sigma'})$**:**

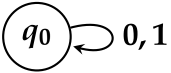
- V' **consists of the vertices** $v_0 a_1 v_1 \ldots a_r v_r$ **with** $(v_{i-1}, v_i) \in E_{a_i}$,
- E'_a **contains the pairs** $(v_0 a_1 v_1 \ldots a_r v_r, v_0 a_1 v_1 \ldots a_r v_r a v)$ **with** $(v_r, v) \in E_a$,
- P'_b **the vertices** $v_0 a_1 v_1 \ldots a_r v_r$ **with** $v_r \in P_b$.

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.

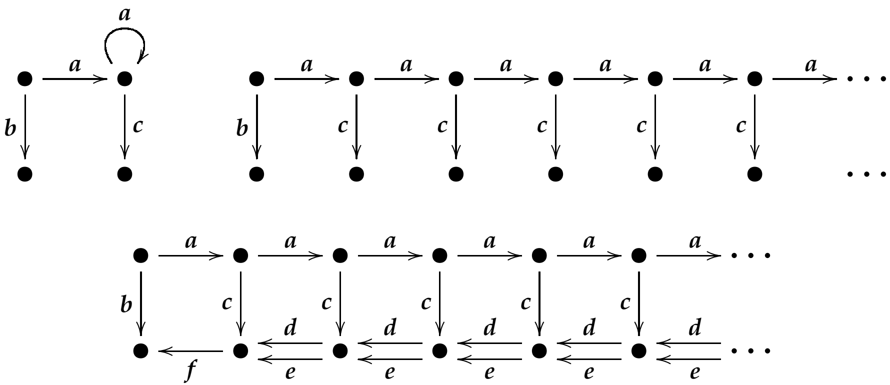
Nontrivial fact:

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

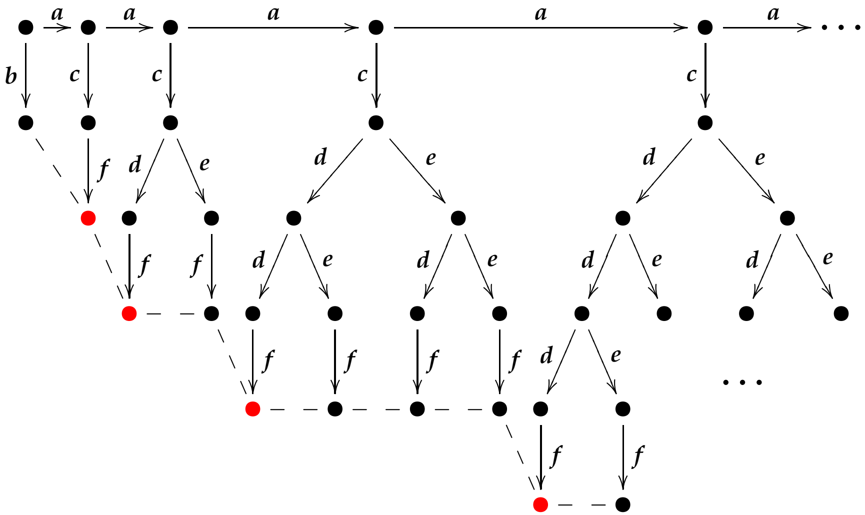
The First Levels

- $\mathcal{G}_0$ is the class of finite graphs.
- $\mathcal{T}_1$ contains the regular trees.
- $\mathcal{G}_1$ contains the prefix-recognizable graphs.

A Finite Graph, a Regular Tree, a PD Graph



Unfolding Again



Interpretation of 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 from an MSO-interpretation in the "comb-tree-structure" the structure

$(\mathbb{N}, +1, 0, \mathrm{Pow}_2)$ **as a structure in the Caucal hierarchy.**

Similar constructions give many interesting structures of this form, with some predicate P replacing Pow_2,

e.g., the predicate to be a factorial.

Scope of Hierarchy?

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

Open questions:

- **Understand which structures belong to the hierarchy**
- **Compute the smallest level on which a structure occurs**

There are structures $\mathcal{S}$ which have a decidable monadic theory but do not belong to the hierarchy.

(Example: Consider the set P of iterated 2-powers 1, 2, 2^2, 2^{2^2}, etc., and take $(\mathbb{N}, +1, 0, P)$.)

The Prime Predicate $\mathbb{P}$

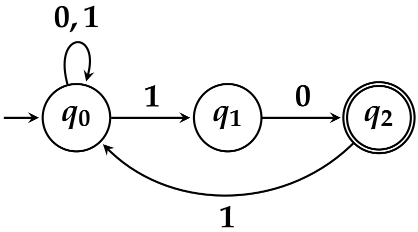
Consider $\alpha_{\mathbb{P}} = 00110101000101\ldots$

By Büchi's Theorem we know:

The MSO-theory of $(\mathbb{N}, +1, 0, \mathbb{P})$ 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 $\alpha_{\mathbb{P}}$?

Decidability of this acceptance problem is open.



Some Survey References

W. Thomas: Automata on Infinite Objects. Handbook of Theoretical Computer Science, Volume B: Formal Models and Semantics (B) Elsevier 1990, pp. 133-192.

W. Thomas: Languages, Automata, and Logic. Handbook of Formal Language Theory, Vol. 3, Springer 1997, pp. 389-455.

E. Grädel, W. Thomas, Th. Wilke (Eds.): Automata, Logics, and Infinite Games: A Guide to Current Research. Springer LNCS 2500 (2002).

W. Thomas: Church's Problem and a Tour through Automata Theory. Pillars of Computer Science, Festschrift for B. A. Trakhtenbrot, Springer LNCS 4800 (2008), pp. 635-655.