Results of the Golden 1960's

Wolfgang Thomas

RWTH AACHEN

Francqui Lecture, Mons, April 2013

Golden Times

Overview

1. Background: MSO-Logic
2. Büchi automata
3. Determinization
4. Tree automata
5. Rabin’s Tree Theorem
6. Regular trees

Background: MSO-Logic

Tarski's Problem

Gödel's and Turing's results implied:

The first-order theory of $(\mathbb{N}, +, \cdot, 0, 1)$ is undecidable.

Alfred Tarski asked:

Is the monadic second-order theory of $(\mathbb{N}, +1, 0)$ decidable?

Today we call this a model-checking problem:

Is the model-checking problem

$$(\mathbb{N}, +1, 0) \models \varphi?$$

w.r.t. MSO -logic decidable?

Other names: S1S, SC, Büchi's arithmetic

Alfred Tarski

MSO Logic over $(\mathbb{N}, +1, 0)$

We have

- **first-order variables $x, y, z, \ldots$ ranging over natural numbers**
- **set variables $X, Y, Z, \ldots$ ranging over sets of natural numbers**
- **terms formed from first-order variables and 0 by application of "$+1$"**
- **atomic formulas $s = t$ and $X(t)$ for terms s, t and set variables X**
- **connectives $\neg, \vee, \wedge, \rightarrow, \leftrightarrow$ and quantifiers $\exists, \forall$**

Example Formulas

- **Over graphs (V, E), we can express 3-colorability:**

  $\exists X_1 \exists X_2 \exists X_3 (\text{Partition}(X_1, X_2, X_3)$
  $\wedge \forall x \forall y (E(x, y) \rightarrow \bigvee_{i \neq j} (X_i(x) \wedge X_j(y))))$

- **Over $(\mathbb{N}, +1, 0)$ the induction axiom:**

  $\forall X (X(0) \wedge \forall y (X(y) \rightarrow X(y+1)) \rightarrow \forall z X(z))$

- **Over $(\mathbb{N}, +1, 0)$ the existence of automaton runs (e.g., for three states):**

  $\exists X_1 \exists X_2 \exists X_3$
  $(\text{Partition}(X_1, X_2, X_3)$
  $\wedge$ **transition and acceptance condition**$)$

Transitive Closure

The relation $\leq$ is the transitive closure of successor.

We have $x \leq y$ iff for all sets X containing x and closed under successor, $X(y)$ holds

Notation: $x < y$, $\exists^\omega y \ldots$... for $\forall x \exists y (x < y \wedge \ldots$, etc.

Taking closure under predecessor starting from y, a quantifier of finite sets suffices (weak MSO logic).

For any MSO-formula $\varphi(z, z')$, we write

$$\varphi^*(x, y) \; :=$$

$$\forall X (X(x) \wedge \forall z, z' (X(z) \wedge \varphi(z, z') \rightarrow X(z')) \; \rightarrow \; X(y))$$

Example

"Each set with two successive elements contains an even number"

First define "y is even":

Set $\varphi_2(z, z') \;\; := \;\; (z+1)+1 = z'$

$\mathrm{Even}(y) := \varphi_2^*(0, y)$

Then we take the following formula:

$$\forall X (\exists x (X(x) \wedge X(x+1)) \rightarrow \exists y (X(y) \wedge \mathrm{Even}(y)))$$

Büchi Automata

SECTION

I

MATHEMATICAL LOGIC

Symposium on Decision Problems

ON A DECISION METHOD IN RESTRICTED SECOND ORDER ARITHMETIC

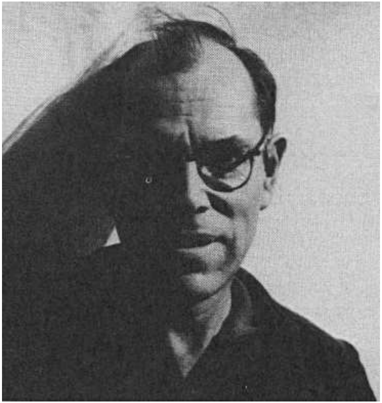
J. RICHARD BÜCHI

University of Michigan, Ann Arbor, Michigan, U.S.A.

Let SC be the interpreted formalism which makes use of individual variables t, x, y, z, . . . ranging over natural numbers, monadic predicate variables q( ), r( ), s( ), i( ), . . . ranging over arbitrary sets of natural numbers, the individual symbol 0 standing for zero, the function symbol ′ denoting the successor function, propositional connectives, and quantifiers for both types of variables. Thus SC is a fraction of the restricted second order theory of natural numbers, or of the first order theory of real numbers. In fact, if predicates on natural numbers are interpreted as binary expansions of real numbers, it is easy to see that SC is equivalent to the first order theory of [Re, +, Pw, Nn], whereby Re, Pw, Nn are, respectively, the sets of non-negative reals, integral powers of 2, and natural numbers.

The purpose of this paper is to obtain a rather complete understanding of definability in SC, and to outline an effective method for deciding truth

This work was done under a grant from the National Science Foundation to the Logic of Computers Group, and with additional assistance through contracts with the Office of Naval Research, Office of Ordnance Research, and the Army Signal Corps.



J. Richard Büchi

Sets versus Words

A set $K \subseteq \mathbb{N}$ can be identified with the infinite 0-1-word α_K where $\alpha_K(i) = 1$ iff $i \in K$.

$\alpha_{\mathbb{P}} = 0\,0\,1\,1\,0\,1\,0\,1\,\ldots$

A tuple $(K_1, \ldots K_n)$ corresponds to an ω-word over $\{0,1\}^n$

$\alpha_{\text{Even},\mathbb{P}} = \binom{1}{0}\binom{0}{0}\binom{1}{1}\binom{0}{1}\binom{1}{0}\binom{0}{1}\ \ldots$

An MSO-formula $\varphi(X_1, \ldots, X_n)$ defines an ω-language:

$L(\varphi) = \{\alpha_{(K_1,\ldots,K_n)} \mid (\mathbb{N}, +1, 0) \models \varphi[K_1, \ldots, K_n]\}$

L is MSO definable (over $(\mathbb{N}, +1, 0)$) iff $L = L(\varphi)$ for some MSO-formula φ.

Consider alphabets $\Sigma = \{0,1\}^n$ for notational simplicity.

Büchi's Version of "Büchi Automaton"

$\Sigma_1^\omega : (\exists r) \cdot A[r(0)] \wedge \forall t\, B[i(t), r(t), r(t')] \wedge (\exists^\omega t)\ C[r(t)]$

Büchi showed closure properties of this formula class and derived that this is a normal form of formulas of S1S.

Consequence: Each formula of S1S can be transformed into a Büchi automaton. MTh$(\mathbb{N}, +1, 0)$ is decidable.

This was new kind of "quantifier elimination".

Büchi-Automata

Format: $\mathcal{A} = (Q, \Sigma, q_0, \Delta, F)$ **with**

- **finite state-set** Q**, initial state** q_0**, set** $F \subseteq Q$ **of final states,**
- **transition relation** $\Delta \subseteq Q \times \Sigma \times Q$

$\mathcal{A}$ **accepts the input word** $\alpha \in \Sigma^\omega$ **if there is a run** ϱ **of** $\mathcal{A}$ **on** α **such that** $\exists^\omega i\ \varrho(i) \in F$

$L(\mathcal{A}) := \{\alpha \in \Sigma^\omega \mid \mathcal{A}$ **accepts** $\alpha\}$

is the ω**-language recognized by** $\mathcal{A}$**.**

L **is called Büchi recognizable if** $L = L(\mathcal{A})$ **for some Büchi automaton** $\mathcal{A}$**.**

Periodicity

Given $\mathcal{A} = (Q, \Sigma, q_0, \Delta, F)$ **define**

$$W_{pq} = \{w \in \Sigma^* \mid \mathcal{A} : p \xrightarrow{w} q\}$$

Then

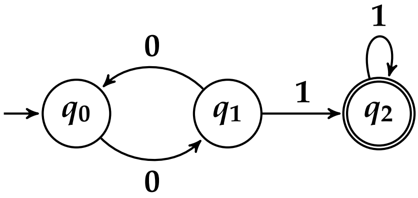
$$L(\mathcal{A}) \; = \; \bigcup_{q \in F} \; W_{q_0 q} \cdot W_{q,q}^{\omega}$$

An ω-language is Büchi recognizable iff it is a finite union of ω-languages $U \cdot V^{\omega}$ with regular $U, V \subseteq \Sigma^*$

Büchi's Theorem:

An ω-language is MSO-definable iff it is Büchi recognizable

From Automata to MSO-Logic



$$\varphi_{\mathcal{A}}(X) :=$$

$$\exists Y_0 \exists Y_1 \exists Y_2 (\mathbf{Partition}(Y_0, Y_1, Y_2) \land Y_0(0)$$

$$\begin{aligned} \land \forall x (& (Y_0(x) \land \neg X(x) \land Y_1(x+1)) \\ & \lor (Y_1(x) \land \neg X(x) \land Y_0(x+1)) \\ & \lor (Y_1(x) \land X(x) \land Y_2(x+1)) \\ & \lor (Y_2(x) \land X(x) \land Y_2(x+1))) \end{aligned}$$

$$\land \forall x \exists y (x < y \land Y_2(y)))$$

From MSO-Logic to Automata

Proceed essentially by induction on formulas

The difficult point is complementation.

Given Büchi's Theorem, we have two immediate applications:

1. **The MSO-theory of $(\mathbb{N}, +1, 0)$ is decidable.**

2. **MSO-formulas can be rewritten as EMSO-formulas.**

Complementation

Büchi's approach to complementation for his Σ_1^ω-formulas:

Represent the complement-ω-language as a finite union of sets $U \cdot V^\omega$ with regular U, V.

As U, V he used equivalence classes of an equivalence relation:

$$u \sim_{\mathcal{A}} v \;\; :\Leftrightarrow \mathcal{A} : p \xrightarrow{u} q \Leftrightarrow \mathcal{A} : p \xrightarrow{v} q$$

$$\text{and } \mathcal{A} : p \xrightarrow{u} q \text{ via } F \text{ iff } \mathcal{A} : p \xrightarrow{v} q \text{ via } F$$

- **$\sim_{\mathcal{A}}$ is a finite congruence, and each $\sim_{\mathcal{A}}$-class is a regular.**
- **For $\sim_{\mathcal{A}}$-classes U, V either $UV^\omega \subseteq L(\mathcal{A})$ or $UV^\omega \cap L(\mathcal{A}) = \emptyset$**
- **Then: $\overline{L(\mathcal{A})} = \bigcup \{UV^\omega | UV^\omega \cap L(\mathcal{A}) = \emptyset\}$**

Ramsey's Theorem

Given a coloring of all pairs (i, j) of natural numbers with $i < j$, there is an infinite subset $H \subseteq \mathbb{N}$ and a fixed color c such that each pair (i, j) with $i, j \in H, i < j$ is colored with c.

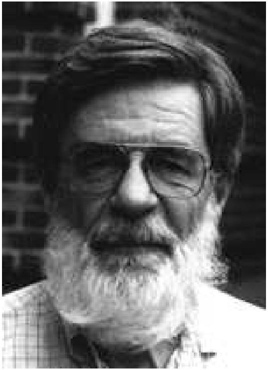
Given $\sim_{\mathcal{A}}$ take as color for (i, j) the $\sim_{\mathcal{A}}$-class of $\alpha[i, j)$

The coloring is additive: the colors of (i, j) and (j, k) determine the color of (i, k).

Consequence: Each ω-word belongs to a set UV^{ω} where U, V are $\sim_{\mathcal{A}}$-classes and moreover $V \cdot V \subseteq V$.

Determinization

McNaughton's Theorem



R. McNaughton

Each Büchi automaton an be transformed into a (deterministic) Muller automaton.

Muller Automata

Format: $\mathcal{A} = (Q, \Sigma, q_0, \delta, \mathcal{F})$

with $\delta : Q \times \Sigma \to Q$, $\mathcal{F} = \{F_1, \ldots, F_k\}$ **where** $F_i \subseteq Q$

Acceptance: $\mathcal{A}$ **accepts** α **iff for the unique run** ϱ **we have**

$$\bigvee_{F \in \mathcal{F}} \Big(\bigwedge_{q \in F} \exists^\omega i\ \varrho(i) = q \wedge \bigwedge_{q \in Q \setminus F} \neg \exists^\omega i\ \varrho(i) = q \Big)$$

Write $\mathcal{A}_q$ **for the det. Büchi automaton** $(Q, \Sigma, q_0, \delta, \{q\})$.

$$L(\mathcal{A}) = \bigcup_{F \in \mathcal{F}} \Big(\bigcap_{q \in F} L(\mathcal{A}_q) \cap \bigcap_{q \in Q \setminus F} \overline{L(\mathcal{A}_q)} \Big)$$

L is Muller recognizable iff L is a Boolean combination of deterministic-Büchi recognizable ω-languages.

Deterministic Büchi Automata in Logic

Given a finite automaton $\mathcal{A}$.

There is a monadic second-order formula $\varphi(y)$ which expresses over an ω-word α:

“the initial segment up to position y is accepted by $\mathcal{A}$”

In $\varphi(y)$ one uses quantifiers “bounded by y”:

$\exists x(x \leq y \wedge \ldots)$, $\exists X(\forall z(X(z) \rightarrow z \leq y) \wedge \ldots)$,

similarly for $\forall$.

L is deterministic-Büchi recognizable iff it is definable in the form

$\forall x \exists y(x < y \wedge \varphi(y))$ **where $\varphi(y)$ is bounded in y.**

There are only two unbounded quantifiers (x and y), all other quantifiers are bounded to a finite domain.

McNaughton's Theorem Logically

Given a Büchi recognizable ω-language of the form

$U \cdot V^{\omega}$ **with** $\sim_{\mathcal{A}}$**-classes** U, V **where** $VV \subseteq V$

The task is to express "$\alpha \in U \cdot V^{\omega}$**"**

by a Boolean combination of formulas

$\forall x \exists y (x < y \wedge \varphi(y))$ **where** $\varphi(y)$ **is bounded in** y

This amounts to a drastic reduction of quantifier complexity.

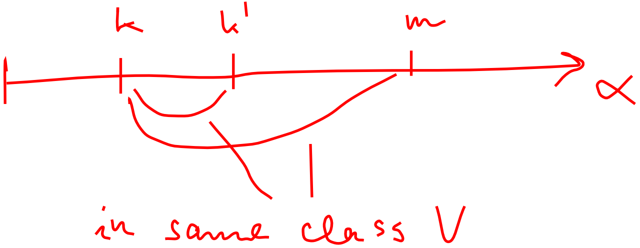
The Merge Relation

Given Büchi automaton $\mathcal{A}$ and an ω-word α:

$k \simeq_\alpha k'(m)$ **means:** $\alpha[k,m) \sim_{\mathcal{A}} \alpha[k',m)$

"k, k' merge at m"

[For the following, more details are in: W. Th., Automata on Infinite Objects, Handbook of Theor. Comput Sci., Elsevier 1990]



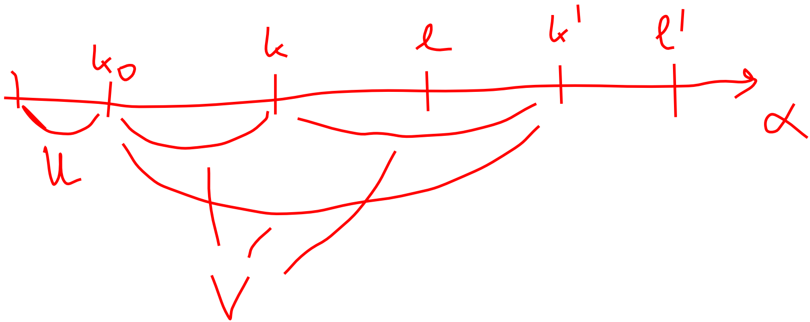
Down to Three Quantifiers

U, V **stand for** $\sim_{\mathcal{A}}$**-classes, and** $V \cdot V \subseteq V$

Then: $\alpha \in U \cdot V^{\omega}$

iff

$$\exists k_0(\alpha[0,k_0) \in U \wedge \exists^{\omega} k \exists \ell(\alpha[k_0,k) \in V \wedge k_0 \sim_{\mathcal{A}} k(\ell)))$$



A Syntactic Detail

$$\exists k_0(\alpha[0,k_0) \in U \wedge \exists^\omega k \exists \ell(\alpha[k_0,k) \in V \wedge k_0 \sim_{\mathcal{A}} k(\ell)))$$

We want a formula

$\exists k_0(\alpha[0,k_0) \in U \wedge \exists^\omega \ell C(k_0,\ell)$ **with** C **bounded in** ℓ

Set $C(k_0,\ell) :=$

$\exists k(k_0 < k < \ell \wedge \alpha[k_0,k) \in V \wedge k_0 \sim_{\mathcal{A}} k(\ell)$

$\wedge$ **there is no** $m < \ell$ **with** $k_0 \sim_{\mathcal{A}} k(m))$

Down to Two Quantifiers

Consider the condition

$\exists k_0 (\alpha[0, k_0) \in U \wedge \exists^\omega \ell C(k_0, \ell))$

We want to exchange $\exists k_0$ and $\exists^\omega \ell$.

Natural idea: Say $\exists^\omega \ell \exists k_0 < \ell \; C(k_0, \ell)$ and k_0 is minimal.

But the minimal k_0 with $\alpha(0, k_0) \in U$ may be incorrectly chosen;

we can take one among those minimal k which finally do not merge.

Two Unbounded Quantifiers Suffice

Let N be the numbers of $\sim_{\mathcal{A}}$-classes.

The previous condition is equivalent to:

$\bigvee_{r=1}^{N}$ **[there are infinitely may ℓ such that among the smallest $(k_1, \ldots, k_r) \leq \ell$ with $\alpha[0, k_i) \in U$ that do not merge at ℓ we have $C(k_i, \ell)$ for some i**

$\wedge$

there are only finitely many such ℓ where the maximum of those smallest $(k_1, \ldots, k_r) \leq \ell$ that do not merge at ℓ actually increases]

A Combinatorial Formulation

Let C be an additive finite coloring of pairs over $\mathbb{N}$.

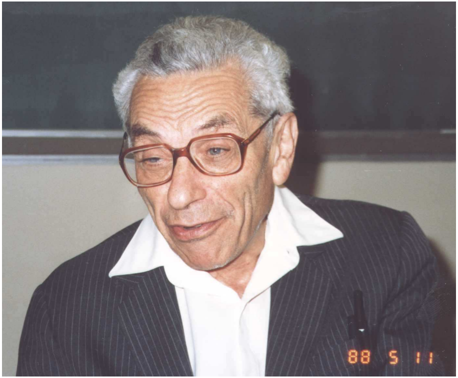
There are bounded formulas $\varphi_{c,\ell}(y)$ and $\psi_{\ell,c,d}(y)$ such that

A set $\{k_0 < k_1 \dots\}$ exists with $C(0, k_0) = c$ and $C(k_i, k_{i+1} = d$

iff

$$(\mathbb{N}, <, \overline{C}) \models \bigvee_{\ell=1}^{|C|} (\exists x \forall y > x \, \varphi_{c,\ell}(y) \wedge \forall x \exists y > x \, \psi_{c,d,\ell}(y))$$

A "combinatorial result"?



Paul Erdös

Parity Automata

accept with a special format of Boolean combination about infinitely many visits to states:

Given a coloring $c : Q \to \{0, \ldots, m\}$

a run ϱ **is accepting if the maximal color occurring infinitely often in it is even**

We show later:

Muller automata can be transformed into parity automata.

Three Problems

Problem 1. Let SC^2 be like SC, except that the functions $2x+1$ and $2x+2$ are taken as primitives in place of $x+1$. Is SC^2 decidable?

This is of some interest, because the functions $2x+1$ and $2x+2$ can be interpreted as the right-successor functions x1 and x2 on the set of all words on two generators 1 and 2.

Problem 2. Let $SC(\alpha)$ be like SC, except that the domain of individuals is the ordinal α, and the well ordering on α is added as a primitive. Is $SC(\omega^2)$ decidable?

As outlined in the introduction, Theorem 2 may be interpreted as a method for deciding whether or not a given finite automaton satisfies a given condition in SC.

Problem 3. Is there a solvability algorithm for SC, i.e., is there a method which applies to any formula $C(\mathbf{i}, \mathbf{u})$ of SC and decides whether or not there is a finite automata recursion $A(\mathbf{i}, \mathbf{r}, \mathbf{u})$ which satisfies the condition C (i.e., $A(\mathbf{i}, \mathbf{r}, \mathbf{u}) \supset C(\mathbf{i}, \mathbf{u})$)?

Tree Automata

The Model T_2

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 second-order theory (short: the monadic theory) of the infinite binary tree, denoted by $\mathrm{MTh}_2(\underline{T_2})$

A labelled binary tree can be presented as $T : \{0,1\}^* \to \Sigma$.

Example Formulas

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

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

$$\forall X\big((X(y) \wedge \forall z(X(z0) \rightarrow X(z)) \wedge \forall z(X(z1) \rightarrow X(z))) \rightarrow X(x)\big)$$

$\mathrm{Chain}(X)$ **("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 Formulas

$x < y$ **(“x is lexicographically before y”):**

$$\exists z(z0 \preceq x \wedge z1 \preceq y) \vee (x \preceq y \wedge x \neq y)$$

$\mathrm{Finite}(X)$:

“each subset Y of X has a minimal and a maximal element w.r.t. $<$”

$$\forall Y((Y \subseteq X \wedge Y \neq \emptyset) \rightarrow$$
$$(\exists y \text{ “}y \text{ is } <\text{-minimal in } Y\text{”} \wedge \exists y \text{ “}y \text{ is } <\text{-maximal in } Y\text{”}))$$

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 Büchi / Muller / parity tree automaton $\mathcal{A} = (Q, \Sigma, q_0, \Delta, F/\mathcal{F}/c)$ accepts the tree t

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

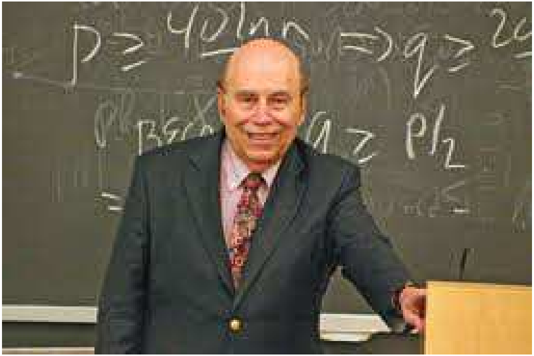
Example

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

recognized by a Büchi tree automaton:

Guess an appropriate path and on it check that infinitely often b occurs by visiting in the next step a final state.

Rabin's Tree Theorem



Michael O. Rabin

Equivalence Logic vs. Automata

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

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

We shall use theorems on infinite games, proved later (in the first lecture on games).

Acceptance via Games

With any parity tree automaton $\mathcal{A} = (Q, \Sigma, q_0, \Delta, c)$

and any input tree t

associate a game $\Gamma_{\mathcal{A},t}$ between two players "Automaton" and "Pathfinder" on the tree 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. We speak of a “parity game”.

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 a "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 $(\varepsilon, t(\varepsilon), q_0)$

Three Results on Parity Games

1. **Parity games are positionally determined: From a given start position one of the two players has a winning strategy, which moreover is positional.**

2. **The set of positions of a parity game graph from which a given player wins is MSO-definable (in the MSO-language for game graphs).**

3. **For parity games over finite game graphs one can decide for any position who wins from this position.**

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.***

More Detail

A reformulation of this yields:

(1) There is an I-labelled tree s such that $s^\wedge t$ satisfies:
(2) for all $\pi \in \{0,1\}^\omega$
(3) for all $\tau_0\tau_1 \ldots \in \Delta^\omega$
(4) if the sequence $s|\pi$ of local instructions applied to the sequence of tree labels $t|\pi$ and to the transition sequence $\tau_0\tau_1 \ldots$ indeed produces the path π, then the state sequence determined by $\tau_0\tau_1 \ldots$ violates the parity condition.

Condition (4) describes a property of ω-words over
$I \times \Sigma \times \Delta \times \{0,1\}$

which obviously can be checked by a sequential parity automaton $\mathcal{M}_4$, independently of t.

The Input-free Case

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

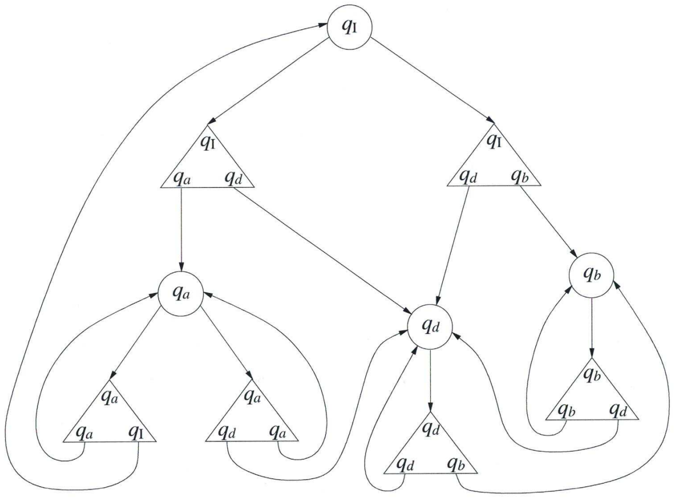
Automaton has positions in in Q and chooses transitions from $Q \times Q \times Q$

Pathfinder has positions in Δ and chooses directions.

The corresponding game graph is finite!

Run Lemma (input-free case): $\mathcal{A}$ admits at least one successful run iff Automaton has a winning strategy in $\Gamma_{\mathcal{A}}$ from position q_0.

The first condition is checked effectively by the 3rd result on parity games.


q_I
q_I
q_a q_d
q_I
q_d q_b
q_a
q_d
q_b
q_a
q_a q_I
q_a
q_d q_a
q_d
q_d q_b
q_b
q_b q_d

Rabin’s Tree Theorem

Rabin’s Tree Theorem

The theory S2S is decidable.

Proof

Consider an S2S-sentence φ

It can be transformed into an input-free parity tree automaton $\mathcal{A}$ such that

the unlabelled infinite binary tree $\underline{T_2}$ satisfies φ iff $\mathcal{A}$ has some successful run.

The second condition can be checked effectively.

Regular Trees

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.

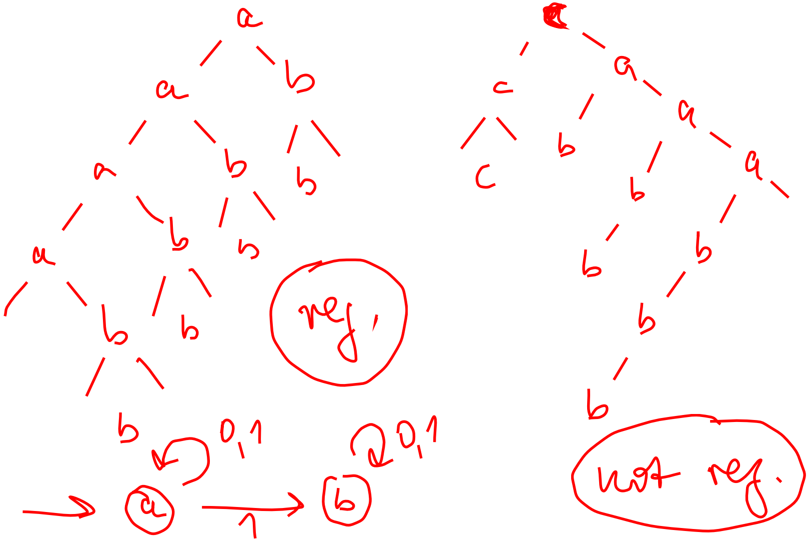
Rabin's Basis Theorem

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 $t(w)$).

Examples



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 $Q \times \Sigma$:

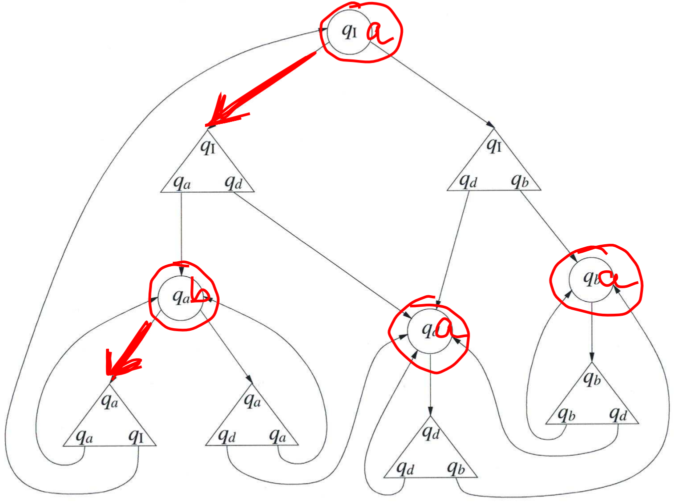
$\mathcal{A}'$ guesses an input tree and works on it as $\mathcal{A}$ does.

$\mathcal{A}'$ 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.



q_I
q_I
q_a q_d
q_I
q_d q_b
q_a
q_d
q_b
q_a
q_a q_I
q_a
q_d q_a
q_d
q_d q_b
q_b
q_b q_d
a
b
a
a

Looking Back

Büchi automata, Muller automata, and parity tree automata provide different versions of quantifier elimination:

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

Tree automata provide a less radical way of quantifier elimination than Büchi automata:

An S2S-formula $\varphi(X_1, \ldots, X_n)$ can be transformed into a formula with two second-order quantifiers:

“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.