

Automatic structures, Part 3

Bakh Khoussainov
Computer Science Department, The University of Auckland,
New Zealand

July 23, 2007

Plan:

- Rabin automatic structures
- Scott ranks of automatic structures
- The isomorphism problem
- Heights of automatic well founded relations
- Cantor-Bendixson ranks of trees
- Resource bounded complexity

- Let \mathcal{T} be the binary tree $(\{0, 1\}^*; \text{Left}, \text{Right})$.
- Let $\text{Tree}(\Sigma)$ be the set of all the Σ -labeled trees (\mathcal{T}, ν) , where $\nu : \mathcal{T} \rightarrow \Sigma$.

Definition

A **Rabin automaton** \mathcal{M} is $(S, \iota, \Delta, \mathcal{F})$, where S is a set of **states**, $\iota \in S$ is the **initial state**, $\Delta : S \times \Sigma \rightarrow P(S \times S)$ is the **transition table**, and $\mathcal{F} \subset P(S)$ is the set of **designated subsets**.

Definition

A **run** of \mathcal{M} on (\mathcal{T}, ν) is a mapping $r : \mathcal{T} \rightarrow S$ such that $r(\text{root}) = \iota$, and for each $x \in \mathcal{T}$ we have

$$(r(\text{Left}(x)), r(\text{Right}(x))) \in \Delta(r(x), \nu(x)).$$

The run is **accepting** if for every path η in \mathcal{T} we have

$$\{s \mid s \text{ appears on } \eta \text{ infinitely many times}\} \in \mathcal{F}.$$

Definition

The **language** accepted by the automaton \mathcal{M} , denoted $L(\mathcal{M})$, is the set of all trees (\mathcal{T}, ν) accepted by \mathcal{M} .

The alphabet Σ is $\{0, 1\}$.

- 1 $\{(T, \nu) \mid \text{there is at least one } x \text{ such that } \nu(x) = 1\}$.
- 2 $\{(T, \nu) \mid \nu(x) = 1 \text{ for finitely many } x \in \mathcal{T}\}$.
- 3 $\{(T, \nu) \mid \text{for every node } x \text{ if } \nu(x) = 1 \text{ then the tree below } x \text{ is labeled by 0s only}\}$.

Theorem (Rabin, 1968)

- 1 *The emptiness problem for Rabin automata is decidable.*
- 2 *Rabin automata recognizable languages are closed under Boolean operations.*

- Consider the structure $\mathcal{T} = (\{0, 1\}^*, \text{Left}, \text{Right})$. Consider the MSO logic defined to be the extension of the FO logic with (monadic) variables for subsets over the domain of \mathcal{T} .
- On \mathcal{T} the MSO logic can express many interesting relations such as $X \subseteq Y$, $\text{Finite}(X)$, $\text{Path}(X)$, $\text{Open}(X)$, $\text{Clopen}(X)$, and $\text{PathOrder}(X, Y)$, etc.

Theorem (Rabin, 1968)

- 1 *A relation $R \subseteq P(\mathcal{T})^n$ is definable in the MSO logic if and only if R is Rabin recognizable.*
- 2 *The monadic second order theory of \mathcal{T} , denoted by S2S, is decidable.*

A **finite Σ -tree** is $t : \text{dom}(t) \rightarrow \Sigma$, where $\text{dom}(t)$ is a finite binary tree. A tree language is a set of Σ -trees.

Definition

A **tree automaton** is $M = (S, \iota, \Delta, F)$, where $F \subseteq S$ and the rest are all as for Rabin automata.

Definition

A run of M on t is **accepting** if the last state along each path of the run is in F .

Now one has:

- 1 The emptiness problem for tree automata is decidable.
- 2 Tree automata recognizable languages are closed under Boolean operations.

Definitions of Automatic Structure

Definition

A structure $\mathcal{A} = (A; R_0, R_1, \dots, R_m)$ is **tree automatic** over Σ if its domain A and all relations R_0, R_1, \dots, R_m are all tree automata recognizable (over Σ).

Definition

A structure $\mathcal{A} = (A; R_0, R_1, \dots, R_m)$ is **Rabin automatic** over Σ if its domain A and all relations R_0, R_1, \dots, R_m are all Rabin automata recognizable (over Σ).

- 1 Every word automatic structure is Rabin automatic.
- 2 Every Büchi automatic structure is Rabin automatic.
- 3 If \mathcal{A} is tree automatic then so is its ω -product.
- 4 (ω, \times) is tree automatic.
- 5 The countable atomless boolean algebra is tree automatic.
- 6 Every tree automatic structure is Rabin automatic.

The term algebra example (Niwinski)

The term algebra $\mathcal{F} = (\text{Terms}(X), f)$, where $|X| = \omega$ and f is the binary function symbol, is tree automatic.

Proof. Let A be the set of all trees $t : \text{dom}t \rightarrow \{0, 1\}$. Let t_0 and t_1 be $\{0, 1\}$ -trees. Define $f(t_0, t_1)$ as the tree t such that $t(\text{root}) = 1$, $t(x0) = t_0(x)$ and $t(x1) = t_1(x)$.

It is easy that $(A, F) \cong \mathcal{F}$. □

Theorem (Khoussainov, Nies, 2006)

Let \mathcal{A} be a Rabin automatic structure. Consider

$$A' = \{(T, v) \in A \mid (T, v) \text{ is a regular tree}\}.$$

The structure \mathcal{A}' is a computable elementary substructure of \mathcal{A} .

The situation is similar to word and Büchi automatic structures:

Fact

- 1 *A structure is Rabin automatic iff it is definable in the monadic second order logic of the binary tree \mathcal{T} .*
- 2 *A structure is tree automatic iff it is definable in the weak monadic second order logic of the binary tree \mathcal{T} .* □

Fact

Every Büchi automatic structure is Borel.

Theorem (Khoussainov, Montalban, Nies; 2007)

There exists a Rabin automatic structure that is not Borel.

Let $V = \{(\mathcal{T}, \nu) \mid \text{each path through } \mathcal{T} \text{ has finitely many 1s}\}$.

Lemma

The language V is Rabin recognizable but not Borel.

Proof. Embed $\omega^{<\omega}$ into \mathcal{T} by: $n_1 \dots n_k \rightarrow 1^{n_1} 0 1^{n_2} 0 \dots 1^{n_k}$. A tree S in $\omega^{<\omega}$ has no infinite path if and only if its image (which is a tree) contains finitely many 1s along each path. \square

We code the set V in a Rabin automatic structure.

Outline of the proof

- 1 The domain D of the structure is $\{(\mathcal{T}, \nu) \mid \nu : \mathcal{T} \rightarrow \{0, 1\}\}$.
- 2 The unary predicate $S = \{(\mathcal{T}, \nu) \mid \text{there is a unique } x \text{ for which } \nu(x) = 1\}$.
- 3 The unary predicate V from the lemma above.
- 4 Two operations $Left' : S \rightarrow S$ and $Right' : S \rightarrow S$ mimic the $Left$ and $Right$ operations on the binary tree.

The structure $(D; S, V, Left', Right')$ is Rabin automatic. If it had a Borel copy then the set V would also be Borel. \square

Definition

For tuples $\bar{a}, \bar{b} \in A^n$ define

- $\bar{a} \equiv^0 \bar{b}$ if \bar{a}, \bar{b} determine isomorphic substructures.
- For $\alpha > 0$, $\bar{a} \equiv^\alpha \bar{b}$ if for all $\beta < \alpha$, for each \bar{c} there is \bar{d} such that $\bar{a}, \bar{c} \equiv^\beta \bar{b}, \bar{d}$, and vice versa.

The **Scott rank** of \bar{a} is the least β such that for all $\bar{b} \in A^n$, $\bar{a} \equiv^\beta \bar{b}$ implies that $(\mathcal{A}, \bar{a}) \cong (\mathcal{A}, \bar{b})$. The Scott rank of \mathcal{A} , $\mathcal{SR}(\mathcal{A})$, is the least $\alpha \geq$ the Scott ranks of tuples of \mathcal{A} .

All known examples of automatic structures have had small Scott ranks.

Fact

The Scott rank of any locally finite graphs is at most 1.

Proof. Indeed, let G be such a graph. Then, by König's lemma, there is an automorphism between tuples \bar{a} to \bar{b} if and only if for every n the n -neighborhood of \bar{a} is isomorphic to the n -neighborhood of \bar{b} . Thus, $(G, \bar{a}) \cong (G, \bar{b})$ iff $\bar{a} \equiv^1 \bar{b}$. \square

Corollary

The Scott rank of the configuration space of any Turing machine is at most 1.

Theorem (B. Khoussainov, M. Minnes, 2007)

For each infinite $\alpha \leq \omega_1^{CK} + 1$ there is an automatic structure of Scott rank α .

We now outline the proof of the theorem.

- Let $\mathcal{C} = (C; R)$ be a computable structure.
- We construct an automatic structure \mathcal{A} whose Scott rank is (close to) the Scott rank of \mathcal{C} .
- We assume that $C = \Sigma^*$ for some finite Σ .
- Let \mathcal{M} be a Turing machine for R .

Consider the configuration space $C(\mathcal{M})$ of the machine \mathcal{M} .

Definition

A deterministic Turing machine \mathcal{M} is **reversible** if the in-degree of each vertex in $C(\mathcal{M})$ is at most 1.

Lemma (Bennet, 1973)

Any deterministic Turing machine may be simulated by a reversible Turing machine. □

So, we assume that \mathcal{M} is reversible.

Some assumptions and terminology for $C(\mathcal{M})$

- 1 All the chains in $C(\mathcal{M})$ are of the type ω or ω^* or n .
- 2 \mathcal{M} halts if and only if its output is *yes*.
- 3 **Terminating computation chains:** finite chains whose base is a valid initial configuration.
- 4 **Non-terminating computation chains:** infinite chains whose base is a valid initial configuration.
- 5 **Unproductive chains:** chains whose base is not a valid initial configuration.

Changing the configuration space $C(\mathcal{M})$:

- 1 Add an ω^* -chain below each base of an unproductive chain.
- 2 Add ω -many copies of ω^* and $\omega^* + \omega$.
- 3 Connect to each base of a computation chain a structure which consists of ω many chains of each finite length.
- 4 Connect each tuple (x_1, \dots, x_n) in C to the initial configuration of M determined by the tuple.

Denote the resulting structure by \mathcal{A} .

Lemma

For \bar{x}, \bar{y} from the domain of \mathcal{C} , and for any ordinal α , $\bar{x} \equiv_{\mathcal{C}}^{\alpha} \bar{y}$ implies that $\bar{x} \equiv_{\mathcal{A}}^{\alpha} \bar{y}$. □

Lemma

$SR(\mathcal{C}) \leq SR(\mathcal{A}) \leq 2 + SR(\mathcal{C})$. □

Lemma (Knight, Millar, in print)

For each $\alpha \leq \omega_1^{CK} + 1$ there is a computable structure of Scott rank α . □

Thus, we have proved the theorem.

Corollary

The isomorphism problem for automatic structures is Σ_1^1 -complete.

Proof. The transformation from \mathcal{C} to \mathcal{A} preserves isomorphism types. The isomorphism problem for computable structures is Σ_1^1 -complete. Hence, the theorem reduces the isomorphism problem for computable structures to the isomorphism problem for automatic structures. \square

Recall the following:

Definition

Let $\mathcal{T} = (T, \leq)$ be a tree. $d(\mathcal{T})$ is the subtree of all nodes x such that x belongs to two distinct infinite paths of \mathcal{T} . Set

- $d^{\alpha+1} = d(d^\alpha(\mathcal{T}))$, and
- for limit ordinal α , set $d^\alpha = \bigcap_{\beta < \alpha} d^\beta(\mathcal{T})$.

Definition

The first α for which $d^{\alpha+1}(\mathcal{T}) = d^\alpha(\mathcal{T})$ is called the **CB rank** of \mathcal{T} denoted by $CB(\mathcal{T})$.

Cantor-Bendixson ranks of successor trees

In the second tutorial we proved the following

Theorem (Khoussainov, Rubin, Stephan, 2003)

If \mathcal{T} is automatic partial order tree then $CB(\mathcal{T}) < \omega$.

This theorem fails if we consider automatic successor trees rather than automatic partial order trees.

Theorem (Khoussainov, Minnes, 2007)

For each computable ordinal $\alpha < \omega_1^{CK}$ there is a successor tree of CB rank α .

Heights of automatic well founded relations

The general case

In the second tutorial we proved that heights of automatic well founded po sets are $< \omega^\omega$. However, we have the following:

Theorem (B. Khoussainov, M. Minnes, 2007)

For each computable ordinal $\alpha < \omega_1^{CK}$, there is an automatic well-founded relation (A, R) such that $\alpha \leq h(A) \leq \omega + \alpha$.

This answers Vardi's question.

Let $G = (V, E)$ be an automatic graph. We ask the following:

- **Connectivity Problem.** Is G connected?
- **Reachability Problem.** Is there a path from x to y ?
- **Infinity Testing Problem.** Is the component of x infinite?
- **Infinite Component Problem.** Does G have an infinite component?

All the problems above are *undecidable*.

Theorem (Khoussainov, Liu, Minnes, 2007)

Given a unary automaton \mathcal{A} of size n representing a locally finite graph G :

- 1 The infinite component problem can be solved in $O(n^{\frac{3}{2}})$.
- 2 The infinity testing problem can be solved in $O(n^{\frac{5}{2}})$.
Moreover, when \mathcal{A} is fixed, the infinity testing problem can be solved in constant time.
- 3 The reachability problem can be solved in $O(|v| + |w| + n^{\frac{5}{2}})$.

Conclusion (Open questions):

- 1 Study the isomorphism problem for the classes of
 - Automatic linear orders.
 - Automatic groups.
 - Automatic Abelian groups.
 - Automatic partial orders.
 - Automatic equivalence structures.
- 2 Study computational complexity of computing isomorphism invariants of automatic structures (heights, CB ranks, etc).
- 3 Study computational complexity of the theories of automatic structures.
- 4 etc.