

Perspectives of Algorithmic Model Theory

Beijing, July 2007

Wolfgang Thomas

`thomas@informatik.rwth-aachen.de`

RWTHAACHEN

Objective

Overview of chapter of model theory motivated by program verification.

Model-Checking Problem: $\mathcal{A} \models \varphi$?

where \mathcal{A} represents a state-based system and φ is a “specification” (requirement).

Classical case: \mathcal{A} is finite Kripke structure.

In this talk, \mathcal{A} is infinite.

Essentially we study decidability of theories of infinite graphs.

The Use of “Syntax”

In the algorithmic context, the finite presentation of infinite models is crucial.

Format of model-checking problem:

- Given a class \mathcal{C} of finitely presented models, given a class \mathcal{L} of sentences
- provide a uniform decision procedure for the problem “ $\mathcal{A} \models \varphi?$ ” for $\mathcal{A} \in \mathcal{C}$, $\varphi \in \mathcal{L}$

Focus of Algorithmic Model Theory

- A jungle of relational models rather than nice algebras like $(\mathbb{N}, +, \cdot)$ and $(\mathbb{R}, +, \cdot)$.
- Fragments of second-order logic rather than first-order logic.
- Operations like unfolding and synchronized products rather than: substructures, reduced products, ultraproducts etc.
- Representation of elements is relevant.
- Results on (efficient) decidability are central.

Models

Transition graphs $G = (V, (E_a)_{a \in \Sigma_1}, (P_b)_{b \in \Sigma_2})$

where $E_a \subseteq V \times V$, $P_b \subseteq V$

$E := \bigcup_{a \in \Sigma_1} E_a$

Examples:

- Kripke structures $G = (V, E, (P_b))$
- the ordering $(\mathbb{N}, <, \mathbb{P})$ of the natural numbers with a designated predicate,
- the binary tree $T_2 = (\{0, 1\}^*, S_0, S_1)$ where $S_i = \{(w, wi) \mid w \in \{0, 1\}^*\}$

Logics

- First-order logic FO,
- first-order logic with reachability FO(R)
(includes relation symbol for E^*),
- monadic second-order logic MSO.

MSO can express

- existence of colorings (e.g. runs of automata),
- the reachability relation E^* .

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

Reachability Problems

for a given set T of vertices:

- Plain reachability
- Universal reachability (“termination”)
- Recurrent reachability
- Controlled reachability (given regular language L)
- Alternating reachability

Overview

- **Connect three approaches to describe infinite models:**
 - internal representation in terms of finite automata
 - external representations by model transformations
 - structural properties
- **Clarify the status of the model-checking problem**

Parts of this talk:

1. **Automatic and prefix-recognizable models**
2. **The Caucal hierarchy**
3. **Generalizing prefix rewriting: Tree and bifix rewriting**

Automatic Transition Graphs

Idea for defining the edge relation(s): **Synchronous scan of word pairs**

$(010, 11011)$ is represented by the word $\begin{pmatrix} 0 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \end{pmatrix} \begin{pmatrix} \$ \\ 1 \end{pmatrix} \begin{pmatrix} \$ \\ 1 \end{pmatrix}$.

$(u, v) \in \Gamma^* \times \Gamma^*$ is represented by a single word over $(\Gamma \cup \{\$\}) \times (\Gamma \cup \{\$\})$.

We speak of an automatic relation, similarly for the n -ary case.

$G = (V, (E_a), (P_b))$ is automatic if for some alphabet Γ

- $V \subseteq \Gamma^*$ and the $P_b \subseteq \Gamma^*$ are regular.
- the E_a are automatic.

Standard closure and decidability properties are preserved.

The FO-theory of an automatic graph is decidable.

Example: The Infinite Grid

$$G_2 = (\mathbb{N} \times \mathbb{N}, E_a, E_b)$$

(with E_a -edges $((i, j), (i, j + 1))$)

and E_b -edges $((i, j), (i + 1, j))$)

G_2 is automatic:

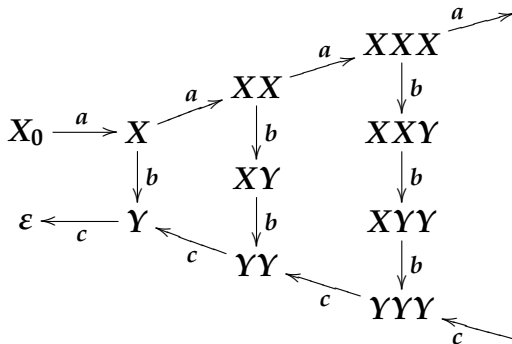
Use $\Gamma = \{X, Y\}$, $V = X^*Y^*$

$$E_a = \{(X^i Y^j, X^i Y^{j+1}) \mid i, j \geq 0\}$$

$$E_b = \{(X^i Y^j, X^{i+1} Y^j) \mid i, j \geq 0\}$$

A typical word in E_b : $\begin{pmatrix} X \\ X \end{pmatrix} \begin{pmatrix} X \\ X \end{pmatrix} \begin{pmatrix} Y \\ X \end{pmatrix} \begin{pmatrix} Y \\ Y \end{pmatrix} \begin{pmatrix} \$ \\ Y \end{pmatrix}$

An infinite acceptor



Accepted language: $\{a^i b^i c^i \mid i > 0\}$

Undecidability of the Reachability Problem

The reachability problem – and hence the MSO-theory – of an automatic graph are in general undecidable.

- Consider the configuration graph of universal Turing machine.
- The one-step relation of pairs $(uqv, u'q'v')$ is automatic.
- Then: Turing machine M accepts word w iff over G from the configuration $q_0\text{code}(M)w$ a halting configuration can be reached.

“Infix rewriting is too strong.”

Pushdown Graphs

$$\begin{array}{ccccccc} q_0 Z_0 & \xrightarrow{a} & q_0 X Z_0 & \xrightarrow{a} & q_0 X X Z_0 & \xrightarrow{a} & q_0 X X X Z_0 \xrightarrow{a} \dots \\ & & \downarrow b & & \downarrow b & & \downarrow b \\ & & q_1 Z_0 & \xleftarrow{b} & q_1 X Z_0 & \xleftarrow{b} & q_1 X X Z_0 \xleftarrow{b} \dots \end{array}$$

$G = (V, (E_a))$ is a pushdown graph if there are

alphabets Q, Γ with $q_0 \in Q, Z_0 \in \Gamma$ and a finite system Δ of rewrite rules $pu_1 \xrightarrow{a} qu_2$ such that

- E_a contains the pairs (pu_1w, qu_2w)
- $V = q_0 Z_0 E^*$.

Prefix-recognizable Graphs

Discard control states.

Represent V by a regular set over some alphabet Γ

Use generalized rewriting rules: $U_1 \xrightarrow{a} U_2$
with regular sets U_1, U_2 of words.

Example: $(\mathbb{N}, +1, <)$, with

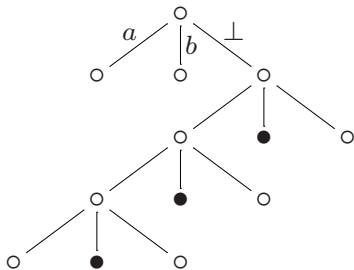
- alphabet $\{a\}$
- $V = a^*$
- E_{+1} defined by $\varepsilon \rightarrow a$
- $<$ defined by $\varepsilon \rightarrow a^+$.

Basic Facts

- The classes of pushdown graphs, prefix recognizable graphs, and automatic graphs form a strict hierarchy.
- The MSO-theory of a prefix recognizable graph is decidable.

Use Rabin's Tree Theorem

Illustration for stack contents $b\perp$, $ba\perp$, $baa\perp$, ...:



Reachability over PD-Graphs

(essentially Büchi 1965)

Given a pushdown automaton $\mathcal{P} = (P, \Sigma, \Gamma, p_0, Z_0, \Delta)$

and $T \subseteq P\Gamma^*$

$\text{pre}^*(T) := \{pv \in P\Gamma^* \mid pv \vdash^* qw \text{ for some } qw \in T\}$

$\text{post}^*(T)$ similarly

If $T \subseteq P\Gamma^*$ is regular, so are $\text{pre}^*(T)$ and $\text{post}^*(T)$.

There is a polynomial-time transformation of finite automata for this.

Connection with language theory

Assume initial and final states are defined by two regular sets I and F .

Then a transition graph “recognizes” a formal language (consisting of the label sequences of paths from I to F)

A language is recognized

- by a finite graph iff it is regular,
- by a pushdown graph (or prefix recognizable graph) iff it is context-free (Muller, Schupp),
- by an automatic graph iff it is context-sensitive (Morvan, Stirling; Rispal).

External Representations

Generate models by transformations

Examples: Interpretations, unfoldings, various products.

Barthelmann, Blumensath, Grädel:

- A graph is prefix-recognizable iff it is MSO-interpretable in T_2 .
- A graph is automatic iff it is FO-interpretable in some tree T_k expanded by partial tree order and equal level predicate.

Structure of Pushdown Graphs

A pointed graph (with designated vertex v) is “finite in the infinite” if

after deletion of the n -neighborhoods of v for increasing n the remaining connected components have only finitely many different isomorphism types.

Example: Binary tree

Counterexample: Infinite $\mathbb{N} \times \mathbb{N}$ grid

Muller, Schupp 1985:

A pointed graph of bounded degree is a pushdown graph iff it is finite in the infinite.

Intermediate Summary

Pushdown graphs can be described via

1. an internal representation (using prefix rewriting)
2. MSO-interpretation in T_2
3. a structural property

Applications:

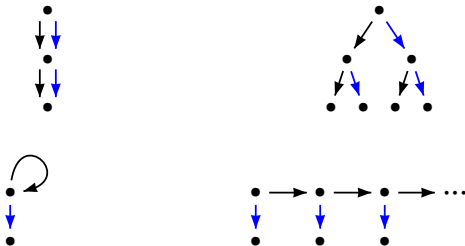
1. Efficient model-checking for special properties
2. Decidability of model-checking for a logic
3. Distinction from other graphs

Unfoldings

The unfolding of a graph $G = (V, (E_a), (P_b))$ from v_0 is the tree T_{G,v_0} whose nodes have the form

$v_0 a_1 v_1 a_2 \dots a_k v_k$ with $(v_{i-1}, v_i) \in E_{a_i}$

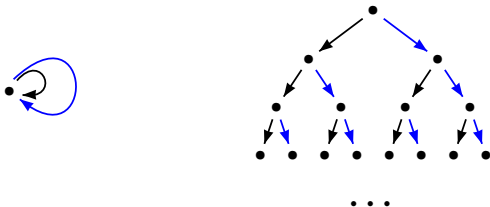
and $v_0 a_1 v_1 a_2 \dots a_k v_k \in P_b$ iff $v_k \in P_b$



MSO-Theory of Unfoldings

Muchnik 1991, Courcelle/Walukiewicz 1998:

If the MSO-theory of G is decidable and v is an MSO-definable vertex of G , then the MSO-theory of $T_{G,v}$ is decidable.



Rabin's Tree Theorem is a special case.

More general construction: Tree model (Muchnik, Walukiewicz)

Caucal's Hierarchy

(in the version of Carayol, Wöhrle 2003)

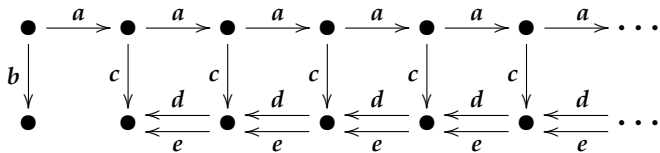
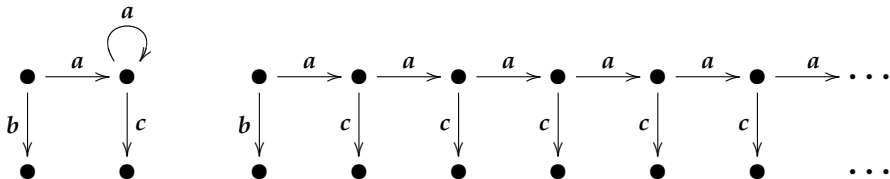
- \mathcal{T}_0 = the class of finite trees
- \mathcal{G}_n = the class of graphs MSO-interpretable in a tree of \mathcal{T}_n
- \mathcal{T}_{n+1} = the class of unfoldings of graphs in \mathcal{G}_n

Note:

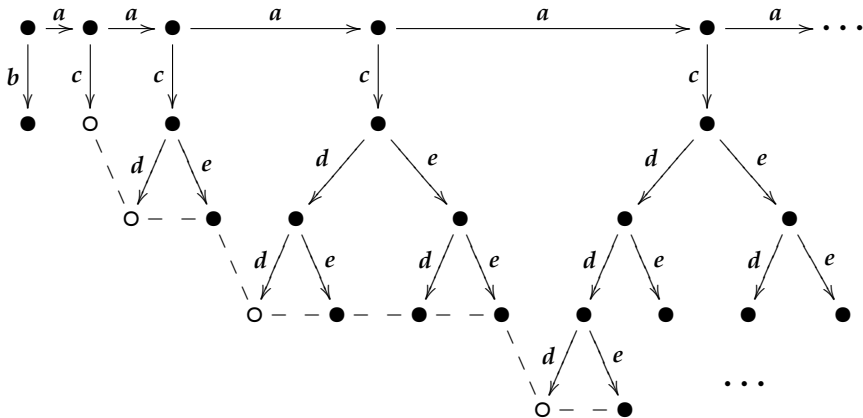
- \mathcal{G}_0 is the class of finite graphs
- \mathcal{T}_1 contains the regular trees
- \mathcal{G}_1 is the class of prefix-recognizable graphs

Each graph in the hierarchy has a decidable MSO-theory.

Example



Application of Unfolding



MSO-interpretation produces $(\mathbb{N}, \text{Succ}, \text{powers of } 2)$.

What about the primes?

Internal Representation

by pushdown systems with nested stacks.

Level-2 stack is a stack where each entry is a stack.

Level-3 stack is a stack of level-2 stacks, etc.

Operations, depending on top symbol of top stack:

- on top level-1 stack as for standard pushdown systems
- deletion of top level-1 stack
- copy (and thus duplication) of top level-1 stack

Internal vs. External Representation

Caucal 2002, Carayol/Wöhrle 2003:

The graphs of the class \mathcal{G}_n of the Caucal hierarchy can be characterized as the transition graphs of level- n pushdown automata (including ε -transitions).

Which transition graphs occur in the Caucal hierarchy?

Despite its richness, the Caucal hierarchy does not contain simple graphs like the infinite grid.

Towards Tree Rewriting Graphs

Idea: Use “prefix rewriting” over trees.

Background:

Ranked alphabets $\Sigma = \Sigma_0 \cup \dots \cup \Sigma_k$.

Tree automata, evaluating input trees from frontier to root.

Obtain “regular tree languages”.

Closure properties and decidability results hold as for words.

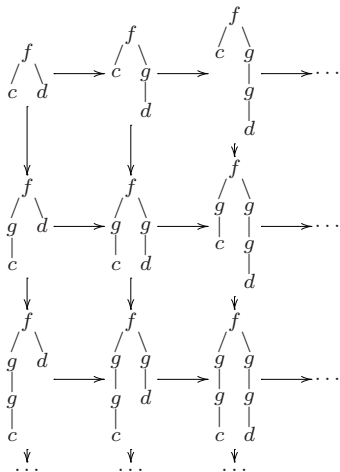
Ground tree rewriting system is a finite set of rules $s \rightarrow_a s'$
with trees s, s'

Application of $s \rightarrow_a s'$:

$t \vdash t'$ iff t' is obtained from t by replacing an occurrence of
ground subtree s by s'

A Ground Tree Rewriting Graph

Rules: $c \rightarrow g(c)$ $d \rightarrow g(d)$



Decidability Results

Let $G(V, \dots)$ be a GTR graph.

If $T \subseteq V$ is regular, so are $\text{pre}^*(T)$ and $\text{post}^*(T)$.

So the reachability problem over GTR graphs is decidable.

Moreover (Dauchet-Tison, Löding):

The FO(R)-theory of a ground tree rewriting graph is decidable.

The (ML+EF+EGF)-theory of a ground tree rewriting graph is decidable.

Here ML+EF+EGF has the following syntax:

T (regular) | $\neg\varphi$ | $\varphi_1 \vee \varphi_2$ | $EX_a\varphi$ | $EF\varphi$ | $EGF\varphi$.

An Undecidability Result

Over a ground tree rewriting graph, the universal reachability problem is undecidable.

(Given a ground tree rewriting graph G , a vertex v and a regular set T of vertices of G , does every path from v through G reach T ?)

Formulation in CTL with state property p_T : $(G, v) \models \text{AF}p_T$?

Reduction of the halting problem for Turing machines

We represent a Turing machine configuration

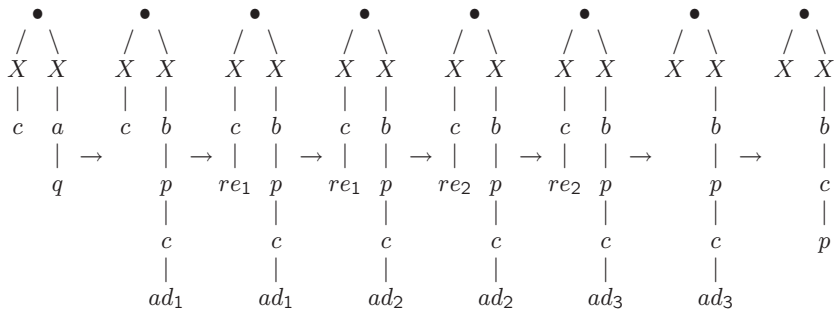
$a_1 \dots a_k q b_1 \dots b_l$ as a two-branch tree

with root \bullet and left branch $Xa_1 \dots a_k$ and right branch

$Xb_1 \dots b_l q$.

TM Step Simulation

Example instruction ($q a b L p$).



Proof Idea

- Dissolve TM simulation step into independently applicable substeps through intermediate situations.
- Define set of intermediate situations as a regular set GOOD of trees.
- Ensure that incorrect substeps lead into a regular set ERROR (= complement of GOOD).
- Consequence: Error-avoiding sequence of situations must simulate TM computation.
- TM halts on empty tape iff each sequence of steps reaches either ERROR or HALT.

Other Cases

The undecidability result extends from universal reachability to

- controlled reachability,
- alternating reachability.

This borderline of undecidability also applies to

- generalized ground tree rewriting graphs,
- unranked ground tree rewriting graphs (Löding, Spelten 2006),
- bifix rewriting graphs (Altenbernd 2006),
- finitely synchronizing graphs (Wöhrle, Th. 2004).

Is there a general principle behind this?

Bifix Rewriting Graphs

Generalize prefix rewriting to mixed prefix-suffix rewriting.

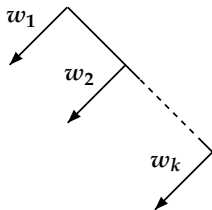
No global control on application of rules!

Further generalization: Bifix rewriting in vector of words.

Obtain rewriting rules over vector (w_1, \dots, w_k) of words.

Reachability is again decidable.

Ground tree rewriting gives dynamic list of stacks:



Summary

- Automata, including tree automata, are a tool for presentation of infinite models that supports model-checking.
- FO(R) is an interesting logic between FO and MSO, allowing decidability results.
- Similar for ML+EF+EGF between ML and CTL.

Perspectives

- Find further adequate logics between FO and MSO.
- Study interesting properties rather than logics.
- Clarify the extensions of the model classes.
- Study dependence of complexity on representation.
- Merge structural properties with internal and external presentations.
- Unfolding is a “sequence model” – consider also “set model” and “function model”.
- Address products and develop the Feferman-Vaught method
- Generalize trees to tree-like structures
- Proceed from graphs to relational structures
- Include “arithmetical” conditions

Conclusion

Automata theory can be a useful and promising framework in building an algorithmic theory of models.