The Reachability Problem over Infinite Graphs

Wolfgang Thomas

RWTH Aachen University, Lehrstuhl Informatik 7, 52056 Aachen, Germany
thomas@informatik.rwth-aachen.de

Abstract. We survey classical and selected recent work on the reachability problem over finitely presented infinite graphs. The problem has a history of 100 years, and it is central for automatic verification of infinite-state systems. Our focus is on graphs that are presented in terms of word or tree rewriting systems.

1 Historical Introduction

In a rather unknown paper of 1910, titled *Die Lösung eines Spezialfalles eines allgemeinen logischen Problems* [20], Axel Thue introduced trees as representations of terms, tree rewriting rules, and the problem of deciding whether from a given term one can reach another given term by a finite number of rewrite steps. He treats a "special case" of this "general logical problem" since he is doubtful about solvability in general. His prophetical statement on the matter is: "Eine Lösung dieser Aufgabe im allgemeinsten Falle dürfte vielleicht mit unüberwindlichen Schwierigkeiten verbunden sein" ("A solution of this problem in the most general case may perhaps be connected with unsurmountable difficulties"). Thue's problem can be stated as the reachability problem over an infinite directed graph; this graph has terms as vertices, and the edge relation is defined by term rewriting rules.

Thue's comment marks the beginning of a long and as yet unfinished track of research – aiming at finding infinite graphs over which the reachability problem "Given vertices u, v, is there a path from u to v?" is decidable. A central motivation today is infinite-state system verification. Another motivation is more general, to develop an algorithmic theory of infinite models. In this context, the reachability problem is the most basic of a family of model-checking problems, also covering several logics in which reachability can be expressed.

The first undecidability results (in fact, showing also undecidability of Thue's problem) follow from the fundamental work of Turing and Post in the 1930's and 1940's. A Turing machine defines an infinite graph consisting of the words that represent Turing machine configurations and where an edge corresponds to a step from one configuration to the next. Assuming a unique acceptance configuration s, and considering a Turing machine that accepts a non-recursive (but recursively enumerable) language, the reachability problem "Is there a path from the initial configuration for the input word w to the configuration s?" is undecidable.

The configuration graph of a Turing machine can be presented with a regular set of words (configurations) as vertex set and an infix rewriting system defining

the edge relation. As Post showed (see e.g. [16]), the reachability problem stays undecidable if instead one uses an edge relation defined by a coordinated prefix-suffix rewriting. Here (namely, in the normal form of Post's canonical systems) a rule consists of deleting a certain prefix of a word while adding a certain suffix.

There are several approaches for detecting classes of graphs where the reachability problem is decidable. An important branch of research has its focus on models that share certain monotonicity properties and are often called "well-structured" (see e.g. [10]). Petri nets and lossy channel systems are prominent cases of this type where reachability was shown decidable.

Here we pursue another approach, where different forms of word and tree rewriting (or corresponding automata theoretic concepts) are used for the presentation of graphs. The reachability problem serves here as a core problem in an algorithmic model theory (of graphs). As an example for the automata theoretic presentation of graphs we mention the *automatic graphs* (see [3]): Here the vertex set is given by a regular language (over a suitable alphabet), and the set of word pairs (u, v) that belong to the edge relation are defined by an automaton that scans u and v synchronously, letter by letter.

It is easy to see that both the infix rewriting and the coordinated prefix-suffix rewriting as mentioned above (for Turing machines and Post's canonical systems, respectively) lead to automatic graphs. The first-order theory of an automatic graph is decidable, but – as seen above – an extension of first-order logic by the reachability relation leads to an undecidable theory. So the leading question for the sequel has to focus on subclasses of the class of automatic graphs.

2 Variants of the Reachability Problem

We consider directed graphs in the format $G = (V, E)$ with vertex set V and edge relation E, and in the format $G = (V, (E_a)_{a\in\Sigma})$ with labelled edges (where E_a contains the edges labelled with a). Let us list a number of reachability problems over such graphs G.

1. *Plain reachability:* Given u and a (finite representation of a) set $T \subseteq V$ of "target vertices", is there a path from u to some vertex of T? (A special case deals with singleton sets $T = \{v\}$.)
2. *Termination:* Given u and a set $T \subseteq V$, does each path from u eventually meet T?
3. *Alternating reachability:* Here we consider a game where two players $1, 2$ choose edges in turn, thus building up a path: Given u and a set $T \subseteq V$, is there a strategy (say for Player 2) to build a path from u that reaches T?
4. *FO(E^*) model-checking:* Here we refer to the expansion of graphs to structures $G' = (V, E, E^*)$ and ask whether the first-order theory of G' is decidable.
5. *FO(Reg) model-checking:* This is defined like FO(E^*), but we refer to edge labeled graphs and the operators E_r with a regular expression r rather than E^* – $E_r(x, y)$ meaning that there is a path from x to y labeled with a word that satisfies r.

6. *MSO model-checking:* Is the monadic second-order theory of G is decidable? (Note that MSO-logic allows to express, e.g., plain reachability of T from u by the formula saying that each set X containing u and closed under E intersects T.)

3 Prefix Rewriting

Consider a graph $G = (V, (E_a)_{a \in \Sigma})$ where V consists of words $qw \in Q \cdot \Gamma^*$ that represent configurations (control state, stack content) of a pushdown automaton, and where E_a consists of those configuration pairs that are induced by an a-transition of the pushdown automaton. Thus one calls G a "pushdown graph". It is known (e.g., from Büchi's analysis of "regular canonical systems" [4]) that the configurations reachable from an initial configuration $q_0\gamma_0$ of a pushdown automaton form a regular set whose representation (e.g., by a finite automaton) can be computed from the pushdown automaton. More precisely, and more abstractly: Given a prefix rewriting system S over the alphabet Γ, if $W \subseteq \Gamma^*$ is regular, then also the set $\mathrm{post}^*(W)$ of words reachable from words in W using rules from S is again regular. The modern proof of this result is done by the "saturation method" ([2, 9]), which basically extends a given finite automaton (recognizing W) by extra transitions to a new finite automaton that recognizes $\mathrm{post}^*(W)$.

This basic result has been extended in several ways. In this section we discuss two extensions that adhere to the idea of prefix rewriting. In the next section other extensions are treated that involve more general mechanisms of rewriting.

First, one can generalize the prefix rewriting rules $u \to v$ (with words u, v) to rules $U \to V$ with regular sets U, V of words instead. An application as a prefix rewrite rule is a step from a word uw to a word vw where $u \in U, v \in V$. The graphs generated by these generalized prefix rewriting systems are called "prefix recognizable" ([5]); in contrast to pushdown graphs they may have vertices of infinite degree.

The second generalization rests on the idea of nested stacks, leading to the so-called higher-order pushdown automata and their associated configuration graphs. We do not give definitions here (see [6, 7]) but just note that for each k one can introduce "level-k pushdown automata" that work over nested stacks of level k; here a standard stack is of level 1, a stack of stacks is of level 2, etc. It is known that a highly complex landscape of (level-k pushdown-) graphs is generated in this way. The two generalizations may also be merged by allowing "regular rules" in the definition of higher-order pushdown automata.

It is interesting to note that for the basic model of pushdown graphs, and for all generalizations of it mentioned above, a much stronger decidability result than that for reachability can be shown: In all these cases of graphs, the MSO-theory is decidable.

4 Ground Tree Rewriting, Mixed Prefix-Suffix Rewriting

The MSO-theory of a graph is of interest in connection to automata theory. Typical sentences expressible in MSO-logic formulate the existence of a "run" (or a "coloring") of some device over the considered structure. However, in verification other properties are more significant; they are related to existence of paths rather than the existence of colorings. It is therefore useful to consider structures that fall outside the general domain of prefix rewriting but nevertheless admit a decision procedure e.g. for plain reachability.

A first natural approach is the idea of ground tree rewriting. We are given a finite tree rewriting system, and the application of a rule $t \to t'$ to a tree s means to replace a subtree t of s by t'. In term rewriting, t, t' are called "ground terms". Considering ground terms as "prefixes" of terms, we can still speak of a kind of "prefix rewriting". We can use these conventions to define naturally the edge relation of a graph whose vertices are trees. In this way we build the co-called ground tree rewriting graphs. The concept of regular tree language can be invoked to single out vertex sets of graphs (as needed for the specification of instances of the reachability problem).

A simple example shows that these graphs go beyond the graphs obtained by prefix rewriting. Using the initial term (= tree) $f(c, d)$ and the two rules $c \to g(c)$ and $d \to g(d)$ we obtain a copy of the infinite grid, with vertices shortly denoted as $f(g^i(c), g^j(d))$. Since it is well-known that the MSO-theory of the infinite grid is undecidable, the MSO model-checking problem over ground tree rewriting graphs is in general undecidable. However, it turns out that the saturation method can be applied again for trees rather than words; so the plain reachability problem over ground tree rewriting graphs is decidable. Even the FO(E^*)-theory of a ground tree rewriting graph is decidable ([8]). An analysis by Löding [14] showed that slightly more ambitious reachability problems are undecidable, namely the termination problem, the alternating reachability problem, and the FO(Reg)-theory. For the extension of these results to graphs that arise from unranked trees (where there is no uniform bound on the branching index) see [15].

The second approach to be discussed here refers to words, but involves rewriting of either prefixes and suffixes. The difference to Post's canonical systems is the lack of coordination between applying prefix rewriting rules and suffix rewriting rules; the order in which they are applied is arbitrary. Not only is the reachability problem decidable over the corresponding graphs; the reachability relation between words turns out to be rational [11, 12]. The same is true when the rules are regular (i.e., involving regular sets rather than single words), see [1]. Finally, all the undecidability results mentioned above for ground tree rewriting graphs also holds for graphs generated by mixed prefix-suffix rewriting.

Despite this correspondence of results for the two types of graphs, there are clearly intuitive differences between them. With the tree structure one can realize a dynamic sequence of stacks; see the left of Figure 1 below where two stacks are seen (the top positions being the leafs), and where an additional stack can be opened at the rightmost leaf. Only for two stacks this can directly be realized

also over words, using mixed prefix-suffix rewriting; see the right of the figure where the symbol Z serves as a separator. On the other hand, mixed prefix-suffix rewriting allows to hand over information say from left to right (by adding symbols on the left and deleting symbols on the right), which corresponds to a migration of information from the bottom of one stack to the bottom of the other. This migration is not directly possible in the representation with ground tree rewriting rules.

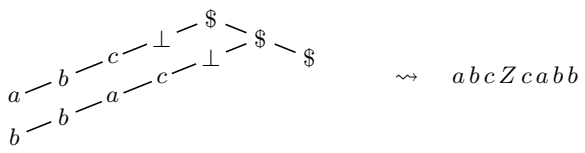

Fig. 1. Two realizations of a pair of stacks

These differences can be fixed also in a more precise sense, by showing that the two classes of ground tree rewriting graphs and mixed prefix-suffix rewriting graphs are indeed incompatible (w.r.t inclusion). First, ground tree rewriting can generate graphs with finite but unbounded degree (just consider the graph generated from $f(d, d)$ by the rule $d \to f(d, d)$). Secondly, we mentioned that the FO(E^*)-theory of a ground tree rewriting graph is decidable whereas it turns out to be undecidable for mixed prefix-suffix rewriting graphs. The latter claim is verified using a simple rewriting system with two rules, one that allows to delete a letter on the left, and one that allows to delete a letter on the right. The transitive closure of the rewrite relation E is $E^* = \{(u, v) \mid u \text{ has infix } v\}$. Now one can apply a result of Quine [18] that the first-order theory of the structure $(\Gamma^*, \text{is infix of})$ is undecidable.

5 Concluding Remarks

These concepts and examples (presented here with a bias to work done in the author's group) are a first step into a widely open field that amounts to new kind of model theory in which algorithmic results are central and where concepts of automata theory are useful.

In this overview we concentrated on the study of graphs and the reachability problem (in different variants). Let us end by mentioning two related aspects that suggest further tracks of investigation.

It would be nice to have structural characterizations of the graph classes discussed above. For a single class a very elegant characterization is known, namely for the pushdown graphs. A theorem of Muller and Schupp [17] characterizes these graphs as the directed graphs G with an origin v_0 and with bounded degree which have only "finitely many end types". An "end" of G is here a connected graph H that is obtained as follows: Delete, for some $n \geq 0$, all vertices

of distance $\leq n$ from v_0 and let H be a connected component of the remaining graph. An end type is an isomorphism type of an end H. – For other types of graphs, in particular for the automatic graphs, such a description is lacking.

A more modest method to classify graphs is to compare their recognition power as acceptors of languages. In this case we deal with labelled graphs $G = (V, (E_a)_{a \in \Sigma})$ which are viewed as (possibly infinite) automata with labelled transitions. Assuming that the "state set" V is represented by a regular language (say of words), one introduces two further regular sets $I, F \subseteq V$ and declares a word w accepted if there is a w-labelled path from I to F (see [19]). Under these conventions, it is known that pushdown graphs with ε-transitions recognize precisely the context-free languages and automatic graphs precisely the context-sensitive languages. For the languages recognized by ground tree rewriting graphs and for mixed prefix-suffix rewriting graphs, only partial results are known (see [13]).

References

1. J. Altenbernd, On bifix systems and generalizations, *Proc. 2nd Int. Conf. on Language, Automata Theory and Applications, LATA 2008*, volume 5196 of Lecture Notes in Computer Science, Springer-Verlag, Berlin 2008, pp. 40-51.
2. A. Bouajjani, J. Esparza, O. Maler, Reachability Analysis of Pushdown Automata: Application to Model-Checking, *CONCUR 1997*, vol. 1243 of Lecture Notes in Computer Science, Springer-Verlag, Berlin 1997, pp. 135-150.
3. A. Blumensath, E. Grädel, Finite Presentations of Infinite Structures: Automata and Interpretations, *Theory of Computing Systems* 37 (2004), 641 674.
4. J. R. Büchi, *Finite Automata, Their Algebras and Grammars* (Ed. D. Siefkes), Springer-Verlag, New York 1989.
5. D. Caucal, On the Regular Structure of Prefix Rewriting, *Theor. Comput. Sci.* 106 (1992), 61-86.
6. D. Caucal, On infinite graphs having a decidable monadic theory, in: *Proc. 27th MFCS* (K. Diks, W. Rytter, Eds.), Springer LNCS 2420 (2002), 165-176.
7. A. Carayol, S. Wöhrle, The Caucal hierarchy of infinite graphs in terms of logic and higher-order pushdown automata, in: *Proc. 23rd Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2003* LNCS 2914 (2003), 112-123.
8. M. Dauchet, S. Tison, Th. Heuillard, P. Lescanne, Decidability of the Confluence of Ground Term Rewriting Systems. *Proc. LICS 1987*, 353-359.
9. J. Esparza, D. Hansel, P. Rossmanith, S. Schwoon, Efficient Algorithms for Model Checking Pushdown Systems. *Proc. Computer Aided Verification, 12th Int. Conf. CAV 2000*, Lecture Notes in Computer Science 1855, Springer-Verlag, Berlin 2000, pp. 232-247.
10. A. Finkel, Ph. Schnoebelen: Well-structured transition systems everywhere!, *Theor. Comput. Sci.* 256 (2001), 63-92.
11. J. Karhumäki, M. Kunc, A. Okhotin, Communication of Two Stacks and Rewriting, *Automata, Languages and Programming, 33rd Int. Coll. ICALP 2006, Part II*, Lecture Notes in Computer Science 4052, Springer-Verlag, Berlin 2006, pp. 468-479.

12. J. Karhumäki, M. Kunc, A. Okhotin, Computing by commuting, *Theor. Comput. Sci.* 356 (2006), pp. 200-211.
13. C. Löding, *Infinite graphs generated by tree rewriting*, PhD thesis, RWTH Aachen, Germany, 2003.
14. C. Löding, Reachability problems on regular ground tree rewriting graphs, *Theory of Computing Systems* 39 (2006), 347-383.
15. C. Löding, A. Spelten, Transition Graphs of Rewriting Systems over Unranked Trees, *Proc. 32nd Int. Symp. on Mathematical Foundations of Computer Science, MFCS 2007*, volume 4708 of Lecture Notes in Computer Science, Springer-Verlag, Berlin 2007, pp. 67-77.
16. M. Minsky, *Computation: Finite and Infinite Machines*, Prentice-Hall, N.J., 1967.
17. D. E. Muller, P. E. Schupp, The theory of ends, pushdown automata, and second-order logic, *Theor. Comput. Sci.* 37 (1985), 51-75,
18. W. V. Quine, Concatenation as a Basis for Arithmetic, *J. Symb. Logic* 11 (1946), 105-114.
19. W. Thomas, A short introduction to infinite automata. In: *Proc. 5th International Conference "Developments in Language Theory"*, Springer LNCS 2295 (2002), 130-144.
20. A. Thue, Die Lösung eines Spezialfalles eines allgemeinen logischen Problems, *Kra. Videnskabs-Selskabets Skrifter, I. Mat. Nat. Kl. 1910, No. 8*, Kristiania 1910. Reprinted in: T. Nagel et. al. (Eds.), *Selected Mathematical Papers of Axel Thue.* Universitetsforlaget, Oslo 1977.