Reachability on prefix-recognizable graphs

Stefan Göller

Institute for Computer Science
University of Leipzig, Germany

Abstract

We prove that on prefix-recognizable graphs reachability is complete for deterministic exponential time matching the complexity of alternating reachability.

Key words: Reachability, Prefix-recognizable graphs, Computational complexity, Alternation

1 Introduction

The *reachability problem* asks, given some graph G and two vertices u and v of G, if there exists a path from u to v in G. The *alternating reachability problem* asks, given two vertices u and v of some game graph, i.e. a graph whose vertices are partitioned into those for player *Eve* and player *Adam*, if *Eve* has a winning strategy in the reachability game from u to v. On finite graphs it is well known that the reachability problem is complete for NL, whereas for the alternating reachability problem the complexity jumps to P-completeness.

It is natural to ask what the complexity status of the (alternating) reachability problem is when the input graph is potentially infinite but finitely presented. On *pushdown graphs* reachability can be solved in deterministic polynomial time [2] and a matching lower bound can easily be shown via a reduction from the nonemptiness problem of context-free grammars. Alternating reachability on pushdown graphs on the other hand is EXP-complete [14]. On *one-counter graphs*, i.e. transition graphs of pushdown automata over a singleton stack alphabet, reachability is NL-complete, whereupon recent results [9,13] imply that alternating reachability is PSPACE-complete. Reachability on order-n pushdown graphs (which generalize pushdown graphs) is (n-1)-EXP-complete [7],

Email address: `goeller@informatik.uni-leipzig.de` (Stefan Göller).

whereas alternating reachability is n-EXP-complete [8,4]. Summarizing, for the above mentioned established classes of infinite graphs, there is an exponential complexity jump from reachability to alternating reachability. A comparable phenomenon can be observed on *regular ground tree rewriting graphs*: Reachability is decidable but already deciding whether all paths starting in some source vertex eventually reach some vertex of a regular set of vertices is undecidable [12]. Even worse, on the more general class of *automatic graphs*, where reachability is Σ^0_1-complete [1], alternating reachability is at least Π^1_1-hard following from a recent result [10], where it is shown inter alia that deciding if an automatic successor tree has an infinite path is Σ^1_1-complete.

Prefix-recognizable graphs, introduced by Caucal [5], form a natural class of infinite graphs with numerous characterizations and a decidable monadic second order theory; [11] gives a nice overview. For instance, they can be characterized as the transition graphs of pushdown automata with ε-transitions, thus generalizing (order-1) pushdown graphs. We prove that already the reachability problem on prefix-recognizable graphs is EXP-hard. Since solving parity games on prefix-recognizable is in EXP [3], it immediately follows that alternating reachability is in EXP too. Thus surprisingly, both reachability *and* alternating reachability on prefix-recognizable graphs are EXP-complete. So alternation on prefix-recognizable graphs comes for free.

2 Preliminaries

For a finite alphabet Σ and languages $U, V, W \subseteq \Sigma^*$, let $(U \times V)W$ denote the binary relation $\{(uw, vw) \mid u \in U, v \in V, w \in W\}$. For simplicity, a *prefix-recognizable graph* is a binary relation $G \subseteq \Sigma^* \times \Sigma^*$ such that for some $k \geq 1$ there exist regular languages $U_1, V_1, W_1, \ldots, U_k, V_k, W_k$ over Σ such that $G = \cup_{i=1}^k (U_i \times V_i)W_i$. We assume that G is represented by a set of triples of regular expressions $(\alpha_i, \beta_i, \gamma_i)_{1 \leq i \leq k}$ where $L(\alpha_i) = U_i$, $L(\beta_i) = V_i$, $L(\gamma_i) = W_i$. The *size* of G is defined as $\sum_{i=1}^k |\alpha_i| + |\beta_i| + |\gamma_i|$.

The *reachability problem* (on prefix-recognizable graphs) is stated as follows:

INPUT: A prefix-recognizable graph G and $u, v \in \Sigma^*$.
QUESTION: $(u, v) \in G^+$?

For proving an EXP lower bound for reachability on prefix-recognizable graphs we need to introduce alternating Turing machines. An *alternating Turing machine (ATM)* is a tuple $\mathcal{M} = (Q, \Gamma, \Lambda, q_0, \delta, \Box)$ where (i) $Q = Q_\forall \uplus Q_\exists$ is a finite set of *states*, which is partitioned into *universal states* $Q_\forall$ and *existential states* $Q_\exists$, (ii) Γ is a finite *tape alphabet*, (iii) $\Lambda \subseteq \Gamma$ is the *input alphabet*, (iv) $q_0 \in Q$ is the *initial state*, (v) $\Box \in \Gamma \setminus \Lambda$ is the *blank symbol*, and (vi) the mapping $\delta : Q \times \Gamma \to (\text{Moves} \times \text{Moves}) \cup \{\bot\}$ with $\text{Moves} = Q \times \Gamma \times \{\leftarrow, \rightarrow\}$

assigns to every pair $(q, a) \in Q \times \Gamma$ either a pair of moves or $\perp$. Hence, we assume that every configuration of an ATM either has two successor configurations or none. An *acceptance tree* T of a configuration u of $\mathcal{M}$ that is in current state $q \in Q$ and scans a symbol $a \in \Gamma$ is a finite tree with root u such that exactly one of the following three conditions holds:

(a) $q \in Q_\forall$, $\delta(q, a) = \perp$ and T is a singleton (with root u).
(b) $q \in Q_\forall$, $\delta(q, a) \neq \perp$, i.e. u has two successor configurations u_1 and u_2, and T consists of the root u together with two subtrees T_1 and T_2, where T_1 (resp. T_2) is some acceptance tree of u_1 (resp. u_2).
(c) $q \in Q_\exists$, $\delta(q, a) \neq \perp$ and T consists of the root u together with one subtree T', where T' is an acceptance tree of one of the two successor configurations of u.

The *height* of an acceptance tree T is the largest distance from some leaf of T to the root of T. We call a configuration of $\mathcal{M}$ *accepting* if it has an acceptance tree. The *language* of $\mathcal{M}$ is defined as $L(\mathcal{M}) = \{w \in \Lambda^* \mid \text{configuration } q_0 w \text{ is accepting}\}$. We denote by EXP the class of problems that are decidable by some $2^{p(n)}$ time bounded deterministic Turing machine for some polynomial p.

3 EXP-hardness of reachability on prefix-recognizable graphs

We proceed with the EXP lower bound for reachability on prefix-recognizable graphs. For this, fix some $p(n)$-space bounded ATM $\mathcal{M} = (Q, \Lambda, \Gamma, q_0, \delta, \Box)$, for some polynomial p, with an EXP-hard membership problem, which exists by [6]. Moreover, let $w \in \Lambda^*$ be an input of length n and let $N = p(n)$. Configurations of $\mathcal{M}$ will be described by the regular language

$$U \quad = \quad \bigcup_{0 \leq i < N} \Gamma^i Q \Gamma^{N-i}.$$

In the following, configurations of $\mathcal{M}$ will be confused with members of U and vice versa. We will construct a polynomial time computable prefix-recognizable graph G such that

$$w \in L(\mathcal{M}) \quad \Leftrightarrow \quad (q_0 w \Box^{N-n}, \varepsilon) \in G^+.$$

We may assume w.l.o.g. that $\mathcal{M}$ never attempts to move left (right) scanning the left-most (right-most) tape cell. Let Moves denote the moves that occur as components in $\delta(Q, \Gamma) \setminus \{\perp\}$. The alphabet of G will be

$$\Sigma \quad = \quad Q \cup \Gamma \cup \{l, r, e\} \cup \{\sigma_{i,j} \mid \sigma \in \text{Moves and } i, j \in [0, N+1]\}.$$

Before defining G, let us introduce a relation $\longrightarrow \subseteq \Sigma^* \times \Sigma^*$ that describes a single step of a depth-first left-to-right traversal of acceptance trees of $\mathcal{M}$. For this, let u range over U being in state $q \in Q$, scanning a symbol $a \in \Gamma$, and where $v \in \Sigma^*$ is arbitrary,

$$uv \longrightarrow v \quad \text{if } q \in Q_\forall \text{ and } \delta(q,a) = \bot, \tag{1}$$

$$xuv \longrightarrow v \quad \text{if } x \in \{e, r\}, \tag{2}$$

$$uv \longrightarrow u'luv \quad \text{if } q \in Q_\forall, \delta(q,a) = (\sigma,\tau) \in \text{Moves}^2, \text{ and } u' \text{ is the } \sigma\text{-successor configuration of } u, \tag{3}$$

$$uv \longrightarrow u'euv \quad \text{if } q \in Q_\exists, \delta(q,a) = (\sigma,\tau) \in \text{Moves}^2 \text{ and } u' \text{ is the } \sigma\text{- or } \tau\text{-successor configuration of } u, \tag{4}$$

$$luv \longrightarrow u'ruv \quad \text{if } q \in Q_\forall, \delta(q,a) = (\sigma,\tau) \in \text{Moves}^2, \text{ and } u' \text{ is the } \tau\text{-successor configuration of } u. \tag{5}$$

Next, we prove that the relation $\longrightarrow$ is as required.

Lemma 1 *For every $u \in U$ the following three statements are equivalent:*

(i) $u \longrightarrow^+ \varepsilon$,
(ii) $uv \longrightarrow^+ v$ *for every* $v \in \Sigma^*$,
(iii) u *is accepting.*

PROOF.
(ii) $\Rightarrow$ (i): Trivial.
(i) $\Rightarrow$ (iii): We prove $u \longrightarrow^k \varepsilon$ implies that u has an acceptance tree by induction on $k \geq 1$. The case $k = 1$ is clear since we can only apply (1) and u has an acceptance tree by (a). Now assume $u \longrightarrow^{k+1} \varepsilon$. Assume u is universal and let $u', u'' \in U$ be the two successor configurations of u. It follows that $u \longrightarrow^{k+1} \varepsilon$ is decomposed as

$$u \xrightarrow{(3)} u'lu \longrightarrow^{k_1} lu \xrightarrow{(5)} u''ru \longrightarrow^{k_2} ru \xrightarrow{(2)} \varepsilon,$$

where $k_1 + k_2 < k$. The latter is due to the observation that every occurrence of the letter l (resp. r) in a word $z \in \Sigma^*$ can only be rewritten by $\longrightarrow$ if l (resp. r) is the first letter of z. Moreover it follows by the definition of $\longrightarrow$ that $u' \longrightarrow^{k_1} \varepsilon$ and $u'' \longrightarrow^{k_2} \varepsilon$. Hence, by induction hypothesis both for u' and u'' there exists an acceptance tree. Thus, by (b), there exists an acceptance tree for u too. The case when u is existential can be proven analogously.
(iii) $\Rightarrow$ (ii): Assume u is accepting. We prove $uv \longrightarrow^+ v$ for all $v \in \Sigma^*$ by induction on the smallest height h of some acceptance tree of u. Assume $h = 0$, hence u is in some state $q \in Q_\forall$ scanning some symbol $a \in \Gamma$ and $\delta(q,a) = \bot$ by (a). Thus, by applying (1), we obtain $uv \longrightarrow v$ for all $v \in \Sigma^*$.

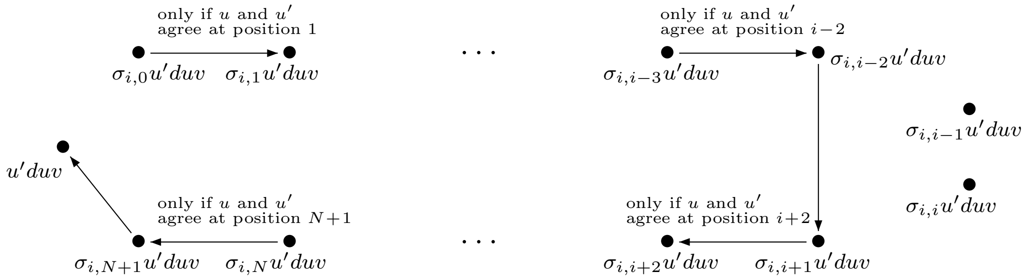

Fig. 1. The graph $G_=$

Now let $h > 0$. In case u is existential, there exists a successor configuration u' of u for which the smallest height of some acceptance tree for u' is strictly less than h. Hence, by induction hypothesis, we have $u'v' \longrightarrow^+ v'$ for all $v' \in \Sigma^*$. By additionally applying (2) and (4) we obtain for every $v \in \Sigma^*$

$$uv \xrightarrow{(4)} u'euv \longrightarrow^+ euv \xrightarrow{(2)} v.$$

Hence $uv \longrightarrow^+ v$ for all $v \in \Sigma^*$. The case when u is universal can be proven analogously. $\square$

Our prefix-recognizable graph G will be the union of several prefix-recognizable graphs we define below.

The prefix-recognizable graph G_1 corresponds to (1):

$$G_1 \quad = \bigcup_{\substack{0 \le i < N \\ q \in Q_\forall, a \in \Gamma : \delta(q,a) = \perp}} \left(\Gamma^i q a \Gamma^{N-i-1} \times \{\varepsilon\}\right) \Sigma^*$$

Similarly, the prefix-recognizable graph G_2 corresponds to (2):

$$G_2 \quad = \quad (\{e, r\} U \times \{\varepsilon\}) \Sigma^*$$

Before defining prefix-recognizable graphs that correspond to (3), (4), and (5) respectively, we need an auxiliary graph $G_=$ that is often used later and has the following property: For each $\sigma \in$ Moves and each $u, u' \in U$ we have $(\sigma_{i,0}u'duv, u'duv) \in G_=^+$ only if u' agrees with u on all positions besides $\{i-1, i, i+1\}$, where $d \in \{l, r, e\}$ and $v \in \Sigma^*$. Figure 1 shows how this can be achieved. Formally, we define

$$
\begin{aligned}
G_{=} \quad = \quad & \bigcup_{\substack{1<i\leq N,\\ \sigma\in\text{Moves}}} \left(\{\sigma_{i,i-2}\}\times\{\sigma_{i,i+1}\}\right)\Sigma^* \;\cup\; \left(\sigma_{i,N+1}\times\{\varepsilon\}\right)\Sigma^* \;\cup \\
& \bigcup_{\substack{0\leq j<i-2 \text{ or } i+1\leq j\leq N\\ c\in\Gamma}} \left(\{\sigma_{i,j}\}\times\{\sigma_{i,j+1}\}\right)\Sigma^j c\Sigma^{N+1}c\Sigma^*.
\end{aligned}
$$

Let us turn to a graph G_3 that corresponds to (3). We make a case distinction. In case the move σ is from $Q\times\Gamma\times\{\leftarrow\}$, we realize (3) by

$$
G_3^{\leftarrow} \quad = \quad \bigcup_{1<i\leq N}\bigcup_{c\in\Gamma}\bigcup_{\substack{q\in Q_\forall, a\in\Gamma\\ \delta(q,a)=(\sigma,\tau):\\ \sigma=(q',b,\leftarrow)}} \left(\{\varepsilon\}\times\sigma_{i,0}\Gamma^{i-2}q'cb\Gamma^{N-i}l\right)\Gamma^{i-2}cqa\Gamma^{N-i}\Sigma^*.
$$

In case σ is from $Q\times\Gamma\times\{\rightarrow\}$, we realize (3) by

$$
G_3^{\rightarrow} \quad = \quad \bigcup_{1<i\leq N}\bigcup_{c\in\Gamma}\bigcup_{\substack{q\in Q_\forall, a\in\Gamma\\ \delta(q,a)=(\sigma,\tau):\\ \sigma=(q',b,\rightarrow)}} \left(\{\varepsilon\}\times\sigma_{i,0}\Gamma^{i-2}bq'c\Gamma^{N-i}l\right)\Gamma^{i-2}qac\Gamma^{N-i}\Sigma^*.
$$

Define $G_3 = G_3^{\leftarrow}\cup G_3^{\rightarrow}$. Note that both $G_3^{\rightarrow}$ and $G_3^{\leftarrow}$ allow to get from vertices uv to vertices $\sigma_{i,0}u'luv$ where u' is not necessarily the σ-sucessor of u. But thanks to $G_{=}$ it is guaranteed that from $\sigma_{i,0}u'luv$ the vertex $u'luv$ can only be reached if u' is indeed the σ-successor of u.

Analogously to (3), the graph G_4 that corresponds to (4) is the union of

$$
\bigcup_{1<i\leq N}\bigcup_{c\in\Gamma}\bigcup_{\substack{q\in Q_\exists, a\in\Gamma\\ \delta(q,a)=(\tau,\tau'):\\ \sigma=(q',b,\leftarrow)\in\{\tau,\tau'\}}} \left(\{\varepsilon\}\times\sigma_{i,0}\Gamma^{i-2}q'cb\Gamma^{N-i}e\right)\Gamma^{i-2}cqa\Gamma^{N-i}\Sigma^*
$$

and

$$
\bigcup_{1<i\leq N}\bigcup_{c\in\Gamma}\bigcup_{\substack{q\in Q_\exists, a\in\Gamma\\ \delta(q,a)=(\tau,\tau'):\\ \sigma=(q',b,\rightarrow)\in\{\tau,\tau'\}}} \left(\{\varepsilon\}\times\sigma_{i,0}\Gamma^{i-2}bq'c\Gamma^{N-i}e\right)\Gamma^{i-2}qac\Gamma^{N-i}\Sigma^*.
$$

Finally, as for (3) and (4), the graph G_5 that realizes (5) is the union of

$$
\bigcup_{1<i\leq N}\bigcup_{c\in\Gamma}\bigcup_{\substack{q\in Q_\forall, a\in\Gamma\\ \delta(q,a)=(\sigma,\tau):\\ \tau=(q',b,\leftarrow)}} \left(\{l\}\times\tau_{i,0}\Gamma^{i-2}q'cb\Gamma^{N-i}r\right)\Gamma^{i-2}cqa\Gamma^{N-i}\Sigma^*
$$

and

$$
\bigcup_{1<i\leq N}\bigcup_{c\in\Gamma}\bigcup_{\substack{q\in Q_\forall, a\in\Gamma\\ \delta(q,a)=(\sigma,\tau):\\ \tau=(q',b,\rightarrow)}} \left(\{l\}\times\tau_{i,0}\Gamma^{i-2}bq'c\Gamma^{N-i}r\right)\Gamma^{i-2}qac\Gamma^{N-i}\Sigma^*.
$$

Define $G = G_1 \cup G_2 \cup G_3 \cup G_4 \cup G_5 \cup G_=$. It is easy to verify that G is computable in time polynomial in $n = |w|$. The following fact is not hard to check.

Fact 2 *Let $A = (U \cup \{\varepsilon\})(\{l, r, e\}U)^*$. Then*

$$\longrightarrow^+ \cap A \times A \quad = \quad G^+ \cap A \times A.$$

The following equivalences finish our EXP lower bound proof:

$$\begin{array}{rcl} w \in L(\mathcal{M}) & \Longleftrightarrow & q_0 w \Box^{N-n} \in U \text{ is accepting} \\ & \overset{\text{Lemma 1}}{\Longleftrightarrow} & \forall v \in \Sigma^* : q_0 w \Box^{N-n} v \longrightarrow^+ v \\ & \overset{\text{Lemma 1}}{\Longleftrightarrow} & q_0 w \Box^{N-n} \longrightarrow^+ \varepsilon \\ & \overset{\text{Fact 2}}{\Longleftrightarrow} & (q_0 w \Box^{N-n}, \varepsilon) \in G^+. \end{array}$$

Since solving parity games on prefix-recognizable graphs is in EXP [3], we get the following theorem.

Theorem 3 *The reachability problem on prefix-recognizable graphs is* EXP*-complete.*

It is a natural question to ask what the complexity of the reachability problem is on *prefix rewrite graphs*, where transitions are given as

$$\bigcup_{1 \le i \le k} (U_i \times V_i)\Sigma^* \quad \text{for regular languages } U_1, V_1, \ldots, U_k, V_k. \qquad (*)$$

Note that in our lower bound proof only $G_=$ was not of the form $(*)$. The latter is inevitable, since there is an easy reduction from the reachability problem on prefix rewrite graphs to the reachability problem on pushdown graphs: Transitions of the kind $(U_i \times V_i)\Sigma^*$ are simulated by a *sequence* of transitions of two nondeterministic finite automata that simulate U_i (resp. V_i) by popping (resp. pushing) symbols from (resp. on) the stack. Thus, reachability on prefix rewrite graphs is P-complete.

Acknowledgments The author would like to thank Markus Lohrey and Christian Mathissen for discussion and proofreading.

References

[1] Achim Blumensath and Erich Grädel. Finite presentations of infinite structures: Automata and interpretations. *Theory of Comput. Syst.*, 37(6):641–674, 2004.

[2] Ahmed Bouajjani, Javier Esparza, and Oded Maler. Reachability analysis of pushdown automata: Application to model-checking. In *CONCUR*, number 1243 in Lecture Notes in Computer Science, pages 135–150. Springer, 1997.

[3] Thierry Cachat. Uniform solution of parity games on prefix-recognizable graphs. *Electr. Notes Theor. Comput. Sci.*, 68(6), 2002.

[4] Thierry Cachat and Igor Walukiewicz. The complexity of games on higher order pushdown automata. *CoRR*, abs/0705.0262, 2007.

[5] Didier Caucal. On infinite transition graphs having a decidable monadic theory. *Theoretical Computer Science*, 290(1):79–115, 2002.

[6] Ashok Chandra, Dexter Kozen, and Larry Stockmeyer. Alternation. *Journal of the Association for Computing Machinery*, 28(1):114–133, 1981.

[7] Joost Engelfriet. Iterated pushdown automata and complexity classes. In *STOC*, pages 365–373. ACM, 1983.

[8] Matthew Hague and Luke Ong. Symbolic backwards-reachability analysis for higher-order pushdown systems. In *FoSSaCS*, volume 4423 of *Lecture Notes in Computer Science*, pages 213–227. Springer, 2007.

[9] Petr Jancar and Zdenek Sawa. A note on emptiness for alternating finite automata with a one-letter alphabet. *Inf. Process. Lett.*, 104(5):164–167, 2007.

[10] Dietrich Kuske and Markus Lohrey. Hamiltonicity of automatic graphs. To appear in Proceedings of IFIP-TCS, 2008.

[11] Martin Leucker. Prefix-recognizable graphs and monadic logic. In *Automata, Logics, and Infinite Games*, Lecture Notes in Computer Science, pages 263–284. Springer, 2001.

[12] Christof Löding. Reachability problems on regular ground tree rewriting graphs. *Theory Comput. Syst.*, 39(2):347–383, 2006.

[13] Olivier Serre. Parity games played on transition graphs of one-counter processes. In *FoSSaCS*, volume 3921 of *Lecture Notes in Computer Science*, pages 337–351. Springer, 2006.

[14] Dejvuth Suwimonteerabuth, Stefan Schwoon, and Javier Esparza. Efficient algorithms for alternating pushdown systems with an application to the computation of certificate chains. In *ATVA*, volume 4218 of *Lecture Notes in Computer Science*, pages 141–153. Springer, 2006.