Note on winning positions on pushdown games with omega-regular winning conditions

Olivier Serre

Note on Winning Positions on Pushdown Games with ω-Regular Conditions

Olivier Serre [1]

LIAFA, Université Paris VII
2, place Jussieu, case 7014
F-75251 Paris Cedex 05

Abstract

We consider infinite two-player games on pushdown graphs. For parity winning conditions, we show that the set of winning positions of each player is regular and we give an effective construction of an alternating automaton recognizing it. This provides a DEXPTIME procedure to decide whether a position is winning for a given player. Finally, using the same methods, we show, for any ω-regular winning condition, that the set of winning positions for a given player is regular and effective.

Keywords: Automata, Games, Infinite Graphs, Pushdown Processes.

1 Introduction

Two-player games on finite and infinite graphs are being studied for several years and one of the central questions is to decide the winner in such a game. This problem closely depends on the winning condition that is considered. For reachability and Büchi winning conditions there are beautiful methods based on fix-points to compute the winning positions [10]. For parity conditions more complicated methods are also known [7,12]. Parity conditions are specially interesting as deciding the winner in such game is equivalent to the mu-calculus model checking problem [4]. On the other hand, pushdown systems define an interesting class of infinite graphs enjoying several nice characterizations [9] and they provide a natural model for infinite systems. All these reasons motivate the study of games on pushdown graphs.

[1] This research has been partially supported by the European Community Research Training Network "Games and Automata for Synthesis and Validation" (GAMES), (contract HPRN-CT-2002-00283), see www.games.rwth-aachen.de.

Walukiewicz provided in [11] a DEXPTIME algorithm to decide the winner in a pushdown game, and he showed also a DEXPTIME lower bound for this problem. As his construction is not uniform (a different finite game has to be considered for any configuration), Cachat recently proposed in [2] a uniform construction for reachability and Büchi conditions, by showing that the set of winning positions is regular. More precisely, he proposed an effective construction of an alternating automaton recognizing precisely the winning configurations. Cachat also states in [1] that the set of winning positions for parity games is regular. The proof of [1] is only sketched and the cmoplexity of the computation is not clear. Note that the results of this paper were found independently of [1].
In this paper we extend Cachat's results, proving that the set of winning positions is regular for any ω-regular winning condition. We provide an effective and efficient construction for these conditions, that leads in the special case of parity conditions to a uniform DEXPTIME algorithm to decide whether a configuration is winning. For this we use slightly different methods from Cachat's techniques, which were essentially based on a fix-point approach that cannot be extended to parity games. The main idea is to consider conditional games and Walukiewicz's decidability results for parity games on pushdown processes. This yields a very general method for showing the regularity and the decidability of ω-regular games played on pushdown graphs.
Note that another interesting method was given by Kupferman and Vardi in [6]. They noted that two-way alternating automata on trees allow a simpler and more natural approach to parity games on pushdown graphs. They use the decidability of the emptiness problem for alternating two-way tree automata.

2 Notations and Definitions

2.1 Pushdown Processes

Throughout the paper, Γ represents a finite alphabet and Γ^* stands for the set of finite words over Γ. The empty word is denoted by ε. Finally, we set $\Gamma^+ = \Gamma\Gamma^*$ and we denote by Γ^∞ the set of finite and infinite words over Γ.

Definition 1 (Pushdown Process) *A* pushdown process *is a tuple* $\mathcal{A} = (Q, \Gamma, \bot, \hookrightarrow)$ *where* Q *is the finite set of states,* Γ *is the finite set of stack symbols,* $\bot \in \Gamma$ *is a special stack symbol (bottom) and* $\hookrightarrow$ *is the transition relation. A configuration of* $\mathcal{A}$ *is a pair* (q, u) *with* $q \in Q$ *and* $u \in (\Gamma \setminus \{\bot\})^*\bot$ *(the top stack symbol is the leftmost symbol of* u*). The bottom stack symbol* $\bot$ *is never put nor removed from the stack. The rewriting rules that can be applied to the set of configurations of* $\mathcal{A}$ *are of the following form:*

- *Push:* $(p, a) \hookrightarrow (q, ba)$*, where* $p, q \in Q$*,* $a \in \Gamma$ *and* $b \in (\Gamma \setminus \{\bot\})$.
- *Pop:* $(p, a) \hookrightarrow (q, \varepsilon)$*, where* $p, q \in Q$ *and* $a \in (\Gamma \setminus \{\bot\})$.

By $(p, v) \to (q, w)$ we mean that from the configuration (p, v) the pushdown process can go in one step to (q, w). Note that the emptiness test of the stack corresponds to a push rule with $a = \bot$. This naturally leads to associate with any pushdown process an infinite graph as follows:

Definition 2 (Pushdown Graph) *With any pushdown process* $\mathcal{A}$ *one can associate a* pushdown graph *defined as a directed graph having the set of configurations of* $\mathcal{A}$ *as nodes and the edges of which are given by the relation* $\to$.

2.2 Playing on a Pushdown Graph

To define a *two-player game* on a pushdown graph $G = (V, \to)$ associated with a pushdown process $\mathcal{A}$, we need a partition $Q_I \sqcup Q_{II}$ of the set of states Q. From this partition follows a partition $V_I \sqcup V_{II}$ of the nodes V of G among the two-player: the nodes of player I are those whose control state belongs to Q_I and the others are player II's nodes. A *play* from a node v proceeds as follow: if $v \in V_I$, player I chooses a successor v' such that $v \to v'$ in G. Otherwise it is player II's turn to choose such a successor. If there exists no such v' the play ends, otherwise the play goes on from v'. Therefore a play can be either finite or infinite and is represented by a (finite or infinite) word on V.
All the winning conditions considered in this paper depend on a *coloring function*. We consider a finite set C of positive integers that we call colors and we define a coloring function c as a function assigning to any state in Q a color in C. This function is naturally extended into a coloring function on V, also noted c, by setting $c(q, v) = c(q)$. Therefore with any play $v_1 v_2 \dots$ we can associate a word $\gamma \in C^\infty$: $\gamma = c(v_1)c(v_2)\dots$. The winner of a play $v_1 v_2 \dots$ is determined by a condition on γ, for instance by a *reachability condition* (Does a given color eventually appear?), a *Büchi condition* (Does a given color appear infinitely often?), a *parity condition* (Is the parity of the smallest color appearing infinitely often even?) or more generally an *ω-regular condition* (see Section 4). In the special case of finite plays, the winner can be determined considering the last state reached or another condition (for instance player I wins whenever a node with no successor is reached). In the following we will assume that all plays are infinite as, adding loops allows us to transform any finite play into a looping infinite play (and coloring in accordance with the conditions on finite plays does not change the winner).

2.3 Strategies and Determinacy

A *strategy* φ for player X ($X = I$ or II) is a partial function $\varphi : V^*V_X \to V$ such that for any word $\alpha \in V^*$, and any node $v \in V_X$ having a successor in G we have, $v \to \varphi(\alpha \cdot v)$. In other words a strategy is a function that, with the beginning of a play, associates a valid successor (if it exists). A player plays according to a strategy φ if whenever it is his turn to choose a move he chooses $\varphi(\alpha)$, where α is the current prefix of the actual play.
A strategy φ is a *winning strategy* from a node v for a player if, in a play starting from v, whatever his adversary plays, playing according to φ is always possible (that is φ is defined) and insures the victory. Finally, a position $v \in V$ is a *winning position* for a player, if he has a winning strategy in the game starting from v. We denote by W_I and W_{II} the respective sets of winning positions of players I and II.
An important question is the *determinacy* of a game, that is to decide if one of the players has a winning strategy. If this holds, the game is *determined*. In the special case of *parity games* (that is, games where player I wins if and only if the smallest color infinitely repeated is even) on arbitrary infinite graphs we have the following result [12]:

Theorem 1 *Parity games are determined. Moreover, for any winning position for player X, there exists a* memoryless strategy*, that is a winning strategy $\varphi : V_X \to V$ that only depends on the current position.*

2.4 $\mathcal{P}$-Alternating Automata

In the following we are interested in recognizing with a finite automaton winning positions on a game graph generated by a pushdown process, that is we want to recognize pairs of the form (q, u) where $q \in Q$ and $u \in \Gamma^*$. For this we use the notion of *$\mathcal{P}$-automaton* [5]. A $\mathcal{P}$-automaton works exactly as a finite automaton except that it does not have initial states: it accepts a pair $(q, u) \in Q \times \Gamma^*$ if there exists, from q, an accepting run on the word u.
For convenience, we will use *alternating $\mathcal{P}$-automata*. An alternating $\mathcal{P}$-automaton is a tuple $\mathcal{B} = (Q, \Gamma, \delta, F)$, where Q is a finite set of states, F is the set of final states and $\delta : Q \times \Gamma \to \mathcal{B}^+(Q)$ is the transition function, where $\mathcal{B}^+(Q)$ is the set of all negation-free boolean formulas over Q.
A *run* of an alternating automaton is a finite tree whose nodes are labeled with states of Q. The level of a node is the length of the word labeling the path from the root to this node. A run associated with a configuration $(q, u = a_1a_2 \cdots a_n)$ is defined by induction:

(1) The root is labeled by q.

(2) The nodes of level n are leaves (i.e. they have no sons). Leaves labeled by a final state are marked as accepting (otherwise, they are marked as rejecting).
(3) If q labels a node v of level $i-1 < n$ and $\delta(q, a_i) = C_1 \vee C_2 \vee \cdots \vee C_m$ with $C_j = q_{j,1} \wedge q_{j,2} \wedge \cdots \wedge q_{j,n_j}$ then v has n_j sons for some j, $1 \leq j \leq m$, labeled by $q_{j,1}, \ldots, q_{j,n_j}$. That is, v must have as sons all the states appearing in one of the conjunctions C_j. In the special case where $\delta(q, a_i) = tt$ (tt is the true formula), v is a leaf marked as accepting. Symmetrically if $\delta(q, a_i) = ff$ (ff is the false formula), v is a leaf marked as rejecting.

A pair (q, u) is *accepted* by $\mathcal{B}$ if there exists a run r associated with (q, u) such that all leaves are marked as accepting.

3 Games on Pushdown Graphs

In this section our aim is to extend to parity games the following result stated by T. Cachat in [2] which generalizes the construction proposed by Esparza et al. for reachability for 1-player games on pushdown graphs [5]:

Theorem 2 *For any two-player game on a pushdown graph constructed from a pushdown process $\mathcal{A} = (Q, \Gamma, \bot, \hookrightarrow)$ with a reachability or Büchi winning condition one can construct a $\mathcal{P}$-alternating automaton recognizing the set of winning configurations for player I. Moreover the automaton is of exponential size and can be constructed in time $\mathcal{O}(|\mathcal{A}|2^{c|Q|^2})$, where c is a constant.*

We therefore consider a parity game played on a pushdown graph G constructed from a pushdown process $\mathcal{A} = (Q, \Gamma, \bot \hookrightarrow)$. To construct a $\mathcal{P}$-alternating automaton recognizing W_I we first need some preliminary work.

3.1 Conditional Games and Preliminary Results

First we introduce some conditional games: for any subset $R \subseteq Q$ we consider the parity game played on the graph $G(R)$ defined by the pushdown process $\mathcal{A}(R)$ obtained from $\mathcal{A}$ as follows:

- We add two states, w and l of respective colors 0 et 1 (they may belong to any of the two players). w stands for winning and l stands for *loosing*
- We suppress all rules $(p, \bot) \hookrightarrow (q, a\bot)$.
- We add a rule $(r, \bot) \hookrightarrow (w, \bot)$ for any $r \in R$.
- We add a rule $(t, \bot) \hookrightarrow (l, \bot)$ for any $t \notin R$.
- Finally we add the loops $(w, \bot) \hookrightarrow (w, \bot)$ and $(l, \bot) \hookrightarrow (l, \bot)$.

By construction the game $G(R)$ has the same winning condition as the initial game G for plays where the stack is never emptied (that is where a configuration $(q, \perp)$ is never reached). In the case where the stack is emptied, the first player wins if and only if the control state belongs to R. In this game we denote by $W_I(R)$ and $W_{II}(R)$ the respective sets of winning positions for player I and player II. Note that the game $G(R)$ is determined.
We have the following result:

Proposition 1 *For any state p and any letter $a \in \Gamma$, we have:*

(1) If $(p, a\perp) \in W_I(\emptyset)$ then for all words $u \in \Gamma^*$, $(p, au\perp) \in W_I$.
(2) If $(p, a\perp) \in W_{II}(Q)$ then for all words $u \in \Gamma^*$, $(p, au\perp) \in W_{II}$.

Proof: We just prove (1) as (2) follows from symmetry. Let $(p, a\perp) \in W_I(\emptyset)$ and let $u \in \Gamma^*$. As $(p, a\perp) \in W_I(\emptyset)$, player I has a winning strategy in $G(\emptyset)$ from $(p, a\perp)$. This strategy ensures that starting from $(p, a\perp)$, the stack will never be emptied, whatever player II plays. Thus, we obtain a corresponding winning strategy in G from $(p, au\perp)$: player I forgets u and plays the same way as in $G(\emptyset)$ from $(p, a\perp)$. This leads to a winning play where au will never be removed from the stack. ■

Note that taking $u = \varepsilon$ in the preceding proposition directly implies that $W_I(\emptyset) \cap W_{II}(Q) = \emptyset$. The last case to consider is the one of configurations $(p, a\perp) \notin W_I(\emptyset) \cup W_{II}(Q)$, that is configurations $(p, a\perp) \in W_I(Q) \cap W_{II}(\emptyset)$. In this case, we must specify which states player I can impose to reach when popping a. We need for generalizing Proposition 1 the following set of goals:

$$\mathcal{R}(p, a) = \{R \subseteq Q \mid (p, a\perp) \in W_I(R) \text{ and } (p, a\perp) \notin W_I(R') \text{ for any } R' \subsetneq R\}$$

Note that $\mathcal{R}(p, a) = \{\emptyset\}$ if and only if $(p, a\perp) \in W_I(\emptyset)$ and $\mathcal{R}(p, a) = \emptyset$ if and only if $(p, a\perp) \notin W_I(Q)$ that is $(p, a\perp) \in W_{II}(Q)$.

Proposition 2 *Let $u \in \Gamma^*$, $p \in Q$, $a \in \Gamma$. Then we have $(p, au\perp) \in W_I$ if and only if there exists some $R \in \mathcal{R}(p, a)$ such that $(q, u\perp) \in W_I$ for all $q \in R$.*

Proof: Let $(p, au\perp) \in W_I$ and let φ be the associated winning strategy of player I in G. In any play where I plays according to φ consider the state q (if it exists) such that $(q, u\perp)$ is the configuration reached when a is popped for the first time. Let R be the set of such states q. We can always choose φ such that this set R is minimal among the sets constructed from all winning strategies. First note that $(p, a\perp) \in W_I(R)$. In order to win from $(p, a\perp)$ in $G(R)$, player I plays as in G from $(p, au\perp)$: either he wins without a being popped, or a is popped and then the play goes to a state $(r, \perp)$ with $r \in R$ and then loops in $(w, \perp)$. Finally, for any $q \in R$, there exists, by definition of R, a play where player I plays according to φ such that the configuration $(q, u\perp)$

is reached. Therefore, the strategy φ is also a winning strategy for player I from $(q, u\perp)$ in game G and thus $(q, u\perp) \in W_I$.

Conversely, assume that there exists $R \in \mathcal{R}(p, a)$ such that for any $q \in R$, $(q, u\perp) \in W_I$. If R is empty this implies that $(p, a\perp) \in W_I(\emptyset)$ and then $(p, au\perp) \in W_I$ by Proposition 1. Otherwise, as $(p, a\perp) \in W_I(R)$ player I has a winning strategy in $G(R)$ from $(p, a\perp)$. herefore, in order to win from $(p, au\perp)$ in G player I forgets u and plays as in $G(R)$ from $(p, a\perp)$. Either a is never popped and then player I wins, or a is popped and then a configuration $(r, u\perp)$ is reached with $r \in R$ and then player I has a winning strategy by hypothesis. ■

We define now a $\mathcal{P}$-alternating automaton $\mathcal{B} = (Q, \Gamma, \delta, F)$ recognizing the winning configurations in a parity game on a pushdown graph defined by $\mathcal{A} = (Q, \Gamma, \perp, \hookrightarrow)$:

- $\delta(p, a) = \bigvee_{R \in \mathcal{R}(p,a)} \bigwedge_{q \in R} q$, for every $p \in Q$, $a \in \Gamma$ and $\delta(p, \perp) = p$.
  Note the following special cases:
  · If $\mathcal{R}(p, a) = \{\emptyset\}$, then the above definition corresponds to $\delta(p, a) = tt$. As in this case we have $(p, a\perp) \in W_I(\emptyset)$, the definition is consistent with Proposition 1.
  · If $\mathcal{R}(p, a) = \emptyset$, then the above definition corresponds to $\delta(p, a) = ff$. As in this case we have $(p, a\perp) \in W_{II}(Q)$, the definition is consistent with Proposition 1.
- $F = \{f \in Q \mid (f, \perp) \in W_I\}$.

The algorith proposed in [11] by Walukiewicz can be used to compute the sets $\mathcal{R}(p, a)$. Therefore we have:

Theorem 3 *For any two-player game on a pushdown graph with a parity winning condition one can construct a $\mathcal{P}$-alternating automaton $\mathcal{B}$ that recognizes the set W_I of winning positions for player I. Moreover, $\mathcal{B}$ gives an exponential-time procedure to decide whether a configuration is winning for player I.*

Proof: The automaton $\mathcal{B}$ is the one described above. We reason by induction on the length l of the stack to prove that the set of pairs $(p, u\perp)$ where $|u| = l$ recognized by $\mathcal{B}$ is exactly the set of configurations in W_I of stack's height l.

- The case $l = 0$ follows from the definition of F.
- Let $l \geq 0$ such that the result holds. Let us prove that it is also true for $l + 1$. Let $(p, au\perp)$ be accepted by $\mathcal{B}$ such that $|u| = l$. As $(p, au\perp)$ is recognized, there exists $R \in \mathcal{R}(p, a)$ such that for any $q \in R$, $(q, u\perp)$ is recognized by $\mathcal{B}$ and then, by induction hypothesis, $(q, u\perp) \in W_I$. Consequently, $(p, au\perp) \in W_I$ by Proposition 2.

Conversely, let $(p, au\bot)$ be in W_I such that $|u| = l$. Using Proposition 2, there exists $R \in \mathcal{R}(p, a)$ such that $\forall q \in R$, $(q, u\bot) \in W_I$. If $R = \emptyset$, then $\delta(p, a) = \not\Vdash$ and $(p, au\bot)$ is recognized by $\mathcal{B}$. Otherwise for all $q \in R$, $(q, u\bot) \in W_I$ and then (q, u) is recognized by $\mathcal{B}$ by induction hypothesis. Therefore, we have an accepting run in $\mathcal{B}$ for $(p, au\bot)$: we choose for successors of p the states of R and then we have an accepting run for each of them while reading u. Thus $(p, au\bot)$ is recognized by $\mathcal{B}$.

Once $\mathcal{B}$ is constructed we can decide in time $\mathcal{O}(|\mathcal{B}| + |u|)$ whether a configuration (p, u) is accepted by $\mathcal{B}$ [3]. Note that the number of states of $\mathcal{B}$ is linear, however the number of transitions can be exponential. Moreover based on Walukiewicz's algorithm [11] $\mathcal{B}$ can be constructed in time $\mathcal{O}(k^n 2^{cmn^2})$, where $k = |\Gamma|$, c is a constant, $m = |Q|$ and $n = |C|$. Effectively, there are $\mathcal{O}(mk2^m)$ problems of the form *Does R belong to $\mathcal{R}(p, a)$?* or of the form *Does $(f, \bot)$ belong to W_I?* and all these problems reduce to a unique problem on parity game on a finite graph that is solved in time $\mathcal{O}(k^n 2^{cmn^2})$ [11]. This gives us a DEXPTIME procedure to decide if a configuration is winning. ■

4 Regularity of Winning Set for ω-Regular Conditions

To generalize Theorem 3 we introduce a more general winning condition:

Definition 3 (ω-Regular Conditions) *An ω-regular condition $\mathcal{L}$ is an ω-regular language on the alphabet of colors C. A play $v_1 v_2 \ldots$ is won by player I under condition $\mathcal{L}$ if and only if $\gamma = c(v_1)c(v_2)\ldots$ belongs to $\mathcal{L}$.*

Theorem 4 *Let G be a pushdown graph constructed from a pushdown process $\mathcal{A}$ and let $\mathcal{L}$ be a regular condition. The set W_I of winning positions for player I in the game played on G with the winning condition $\mathcal{L}$ is regular and can be effectively constructed. Moreover each player has a winning strategy realized by a pushdown automaton.*

Proof: Let set $\mathcal{A} = (Q, \Gamma, \bot, \hookrightarrow)$. As $\mathcal{L}$ is regular, it is recognized by a finite *deterministic* parity automaton [8] $\mathcal{B} = (Q', C, q_0, \cdot, c', C')$ with alphabet C (the set of colors of $\mathcal{A}$) and where Q' is the set of states of $\mathcal{B}$, q_0 is the initial state, $\cdot$ is the transition function, $C' \subset \mathbb{N}$ is a finite set of colors and $c' : Q' \to C'$ is the coloring function of $\mathcal{B}$. The automaton $\mathcal{B}$ accepts an infinite word $\alpha \in C^\infty$ if and only if the smallest color appearing infinitely often while reading α is even.
The main idea is to define a new pushdown game by composing $\mathcal{B}$ with $\mathcal{A}$ in order to compute on-the-fly the behavior of $\mathcal{B}$ on γ while playing the game and therefore to express the acceptance condition of $\mathcal{B}$ on γ as a parity winning condition on this new game. We thus define a new pushdown process $\widetilde{\mathcal{A}} =$

$(Q\times, Q', \bot, \hookrightarrow\!\!\!\!\!\rightarrow)$ where $\hookrightarrow\!\!\!\!\!\rightarrow$ is defined by: $((p,p'),a) \hookrightarrow\!\!\!\!\!\rightarrow ((q,q'),u)$ if and only if $(p,a) \hookrightarrow (q,u)$ and $p' \cdot c(p) = q'$. We finally consider the parity game $\widetilde{G}$ defined by $\widetilde{\mathcal{A}}$ with the coloring function $\widetilde{c}((p,p')) = c'(p')$. The result follows from the fact that a configuration (p,u) is winning in G for player I if and only if the configuration $((p,q_0),u)$ is winning in $\widetilde{G}$ for player I.
Consider a winning position (p,u) for player I in G. Therefore, $((p,q_0),u)$ is a winning position in $\widetilde{G}$ and, as $\widetilde{G}$ is a parity game played on a pushdown graph, it follows from [11] that there is a winning strategy realized by a pushdown automaton. But, with a winning strategy in $\widetilde{G}$ from $((p,q_0),u)$, it is not difficult to associate a winning strategy in G from (p,u) by constructing on-the-fly the associated play in $\widetilde{G}$ and then by playing in accordance with the corresponding winning strategy in $\widetilde{G}$. Thus it follows that the winning strategy in G can be given by a pushdown automaton. ■

5 Conclusion

We have shown how to construct, for any pushdown game with an ω-regular winning condition, an alternating automaton recognizing the winning positions of a given player. Unfortunately some costly and complicated precomputing steps are needed. Future work aims at simplifying these steps as they concern similar instances of the same problem and as the algorithm used to solve them was designed for a more general problem. However, the global complexity of deciding the winner will not be improved, as Walukiewicz proved the DEXPTIME completeness of deciding the winner even with the weak winning condition of reachability.

In fact, Proposition 2 is still true for any Borel condition specified by a language $L \subseteq C^\omega$ such that any suffix of a word in L is in L and any word obtained by adding a finite prefix to a word in L is in L. Therefore, it follows directly that under such condition, the set of winning positions is regular (without being necessarily computable). Consequently, another interesting investigation would be to look at more general Borel conditions than regular conditions.

I gratefully acknowledge the many helpful suggestions of Anca Muscholl during the preparation of the paper. I also wish to express my thanks to Marcin Jurdzinski, Marc Zeitoun and Wieslaw Zielonka for their remarks on the paper, and to the anonymous referees.

References

[1] T. Cachat. Uniform solution of parity games on prefix-recognizable graphs. To appear in ENTCS. www-i7.informatik.rwth-aachen.de/ cachat.

[2] T. Cachat. Symbolic strategy synthesis for games on pushdown graphs. In *Proceedings of ICALP'02*, volume 2380 of *Lecture Notes in Computer Science*, pages 704–715. Springer, 2002.

[3] K. Chandra, D. Kozen, and J. Stockmeyer. Alternation. *Journal of the Association of Computing Machinery*, 28(1):114–133, 1981.

[4] E.A. Emerson and C.S. Jutla. Tree automata, mu-calculus and determinacy (extended abstract). In *FOCS: IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 368–377, 1991.

[5] J. Esparza, D. Hansel, P. Rossmanith, and S. Schwoon. Efficient algorithms for model checking pushdown systems. In *Proceedings of CAV'00*, volume 1855 of *Lecture Notes in Computer Science*, pages 232–247. Springer, 2000.

[6] O. Kupferman and M.Y. Vardi. An automata-theoretic approach to reasoning about infinite-state systems. In *Proceedings of CAV'00*, volume 1855 of *Lecture Notes in Computer Science*, pages 36–52. Springer-Verlag, 2000.

[7] R. McNaughton. Infinite games played on finite graphs. *Annals of Pure and Applied Logic*, 65:149–184, 1993.

[8] A.W. Mostowski. Regular expressions for infinite trees and a standard form for automata. In *Computation theory*, volume 208 of *Lecture Notes in Computer Science*, pages 157–168. Springer, Berlin, 1984.

[9] D. Muller and P. Schupp. The theory of ends, pushdown automata, and second-order logic. *Theoretical Computer Science*, 37:51–75, 1985.

[10] W. Thomas. On the synthesis of strategies in infinite games. In *Proceedings of STACS '95*, volume 900 of *Lecture Notes in Computer Science*, pages 1–13. Springer, 1995.

[11] I. Walukiewicz. Pushdown processes: games and model checking. *Information and Computation*, 157:234–263, 2000.

[12] W. Zielonka. Infinite games on finitely coloured graphs with applications to automata on infinite trees. *Theoretical Computer Science*, 200(1-2):135–183, 1998.