Parity and Exploration Games on Infinite Graphs

Hugo Gimbert

Parity and Exploration Games on Infinite Graphs

Hugo Gimbert

Université Paris 7, LIAFA, case 7014
2, place Jussieu
75251 Paris Cedex 05, France
Hugo.Gimbert@liafa.jussieu.fr

Abstract. This paper examines two players' turn-based perfect-information games played on infinite graphs. Our attention is focused on the classes of games where winning conditions are boolean combinations of the following two conditions: (1) the first one states that an infinite play is won by player 0 if during the play infinitely many different vertices were visited, (2) the second one is the well known parity condition generalized to a countable number of priorities.
We show that, in most cases, both players have positional winning strategies and we characterize their respective winning sets. In the special case of pushdown graphs, we use these results to show that the sets of winning positions are regular and we show how to compute them as well as positional winning strategies in exponential time.

1 Introduction

Two-player games played on graphs have attracted a lot of attention in computer science. In verification of reactive systems it is natural to see the interactions between a system and its environment as a two-person game [19,9], in control theory the problem of controller synthesis amounts often to finding a winning strategy in an associated game [1].

Depending on the nature of the examined systems various types of two-player games are considered. The interactions between players can be turn-based [23,19] or concurrent [7,8], finite like in reachability games or infinite like in parity or Muller games, the players can have perfect or imperfect information about the play. Moreover the transitions may be deterministic or stochastic [6,8] and finally the system itself can be can be finite or infinite.

Another source of diversity comes from players' objectives, i.e. winning conditions.

Our work has as a framework turn-based perfect information infinite games on pushdown graphs. The vertices of such graphs correspond to configurations of a pushdown automaton and edges are induced by push-down automaton transitions. The interest in such games comes, at least in part, from practical considerations, pushdown systems can model, to some extent, recursive procedure calls. On the other hand, pushdown graphs are one of the simplest class of infinite

graphs that admit non trivial positive decidability results and since the seminal paper of Muller and Schupp [14] many other problems are shown to be decidable for this class [2,13,5,18,3,22,4,17].

Let us describe briefly a play of such a game. The set of vertices is partitioned into two sets: vertices belonging to player 0 and vertices belonging to his adversary 1. Initially, a pebble is placed on a vertex. At each turn the owner of the vertex with the pebble chooses a successor vertex and moves the pebble onto it. Then the owner of this new vertex proceeds in the same way, and so on. The successive pebble positions form an infinite path in the graph, this is the resulting play.

In this framework, different objectives have been studied. Such an objective is described in general as the set of infinite plays that are winning for player 0, and it is called a winning condition. A lot of attention has been given to the case where this set is regular, which gives rise to Müller and parity winning conditions [23,22,19] which lie on the level Δ_2 of the Borel hierarchy. However, recently Cachat et al. [5], presented a new winning condition of Borel complexity Σ_3 which still remains decidable. This Σ_3-condition specifies that player 0 wins a play if there is no vertex visited infinitely often. Yet another condition, *unboundedness*, was introduced by Bouquet et al. [3]. The unboundedness condition states that player 0 wins a play if the corresponding sequence of stack heights is unbounded. Obviously the conditions of [5] and [3] are tightly related, if no configuration of the push-down system is visited infinitely often the the stack is unbounded. The converse can be established as well if the winning strategies are memoryless, i.e. do not depend on the past.

In this paper, we first transfer the condition of [3] to arbitrary infinite graphs of finite degree. In the context of arbitrary infinite graphs we examine *Exploration* condition which states that a play is won by player 0 if the pebble visits an infinite number of different vertices. Obviously for the particular case of push-down graphs this gives the same condition as [3]. In fact we go a step further and consider the games whose winning conditions are boolean combinations of Exploration condition and of the classical parity condition. We note respectively $\mathit{Exp} \cup \mathit{Parity}$ and $\mathit{Exp} \cap \mathit{Parity}$ the games obtained by taking the union and the intersection of Exploration and Parity conditions.

We also consider a particular extension of the classical Parity condition to the case with an infinite number of priorities and denote it Parity_∞ (see also [11] for another approach to parity games with an infinity of priorities).

We prove the following results in the context of the games over any infinite graphs:

- Both players have positional winning strategies for the game with the winning condition $\mathit{Exp} \cup \mathit{Parity}$, including the case where there is an infinite number of priorities.
- In the case where there are finitely many priorities, player 1 has also a winning positional strategy in the game where the winning condition for player 0 is of type $\mathit{Exp} \cap \mathit{Parity}$. Moreover, we can easily characterize the winning set of player 0.

Even if general results concerning winning strategies over arbitrary infinite graphs are of some interest we are much more interested in decidability results for the special case of pushdown graphs. In the case where the game graph is a pushdown graph, we prove that the sets of winning configurations (positions) for player 0 (and thus also for player 1) are regular subsets of $Q\Gamma^*$ where Q is the set of states of pushdown system and Γ is the stack alphabet for both types of games $Exp \cup Parity$ and $Exp \cap Parity$. We provide also an algorithm for computing a Büchi automaton with $2^{\mathcal{O}(d^2|Q|^2+|\Gamma|)}$ states recognizing those winning sets, where d is the number of priorities of the underlying parity game and Q and Γ are as stated above. Moreover, we show that for both games and both players, the set of winning positional strategies is regular and recognized by alternating Büchi automata with $\mathcal{O}(d|Q|^2 + |\Gamma|)$ states.

These results constitute an extension of the results of [5,3,22,18,20]: The papers [22,20,18] examine only *Parity* conditions with a finite number of priorities for pushdown games. Bouquet et al. [3] were able to extend the decidability results to the games with the winning condition of the form $Exp \cup Buchi$ or $Exp \cap Buchi$, i.e. union and intersections of Büchi condition with Exploration condition. However this class of conditions is not closed under boolean operations (intersecting Büchi and co-Büchi conditions with an Exploration condition is not in this class). In our paper we go even further since we allow boolean combinations of *Exp* conditions with parity conditions. Since parity conditions, after appropriate transformations, are closed under boolean operations we show in fact that it is decidable to determine a winner for the smallest class of conditions containing Exploration conditions and Büchi conditions and closed under finite boolean operations.

For computing the winning sets and the winning strategies, we make use of tree automata techniques close to the one originated in the paper of Vardi [20] and applied in [16,12]. This is a radical departure from the techniques applied in [21,22,3,18] which are based on game-reductions.

This paper is organized as follows. In the first part, we introduce some basic definitions and the notions of Exploration and Parity games. In the second part, we prove the results concerning the winning strategies for the games $Exp \cup Parity_\infty$ and $Exp \cap Parity$, and make some comments about the $Parity_\infty$ game. In the third part, we describe the construction of automata computing the winning sets and the winning positional strategies. Due to space limitation, most proofs are omitted and can be found in the full version [10].

2 Parity and Exploration Games

In this section, we present basic notions about games and we define different winning conditions.

2.1 Generalities

The games we study are played on oriented graphs of finite degree, with no dead-ends, whose vertices sets is partitioned between the vertices of player 0

and the vertices of player 1. Such a graph is called an arena. At the beginning of a play, a pebble is put on a vertex. During the play, the owner of the vertex with the pebble moves it to one of the successors vertices. A play is the infinite path visited by the pebble. A winning condition determines which player is the winner. Here follows the formal description of these notions.

Notations. Let $G = (V, E)$ be an oriented graph with the set $E \subset V \times V$ of edges. Given a vertex v, vE denotes the set of successors of v, $vE = \{w \in V : (v, w) \in E\}$, whereas Ev is the set of predecessors of v. For a set $H \subseteq E$ of edges, $Dom(H)$, the domain of H, denotes the set of the vertices adjacent to edges of H.

Parity arenas. An arena is a tuple (V, V_0, V_1, E), where (V, E) is a graph of finite degree with no dead-ends and (V_0, V_1) is a partition of V. Let $i \in \{0, 1\}$ be a player. V_i is the set of vertices of player i. We will often say that $G = (V, E)$ itself is an arena, when the partition (V_0, V_1) is obvious. An infinite path in G is called a play, whereas a finite path in G is called a finite play. When the vertices of G are labeled with natural numbers with a map $\phi : V \to \mathbb{N}$, G is said to be a parity arena.

Winning Conditions and Games. A winning condition determines the winner of a play. Formally, it is a subset $\mathrm{Vic} \subseteq V^\omega$ of the set of infinite plays. A game is a couple (G, Vic) made of an arena and a winning condition. Often, when the arena G is obvious, we will say that Vic itself is a game. A play $p \in V^\omega$ is won by player 0 if $p \in \mathrm{Vic}$. Otherwise, if $p \notin \mathrm{Vic}$, it is said to be won by player 1. Vic is said to be *concatenation-closed* if $V^* \mathrm{Vic} = \mathrm{Vic}$.

Strategies, Winning Strategies and Winning Sets. Depending on the finite path followed by the pebble, a strategy allows a player to choose between a restricted number of successor vertices. Let $i \in \{0, 1\}$ be a player. Formally, a strategy for player i is a map σ, which associates to any finite play $v_0 \cdots v_n$ such that $v_n \in V_i$ a nonempty subset $\sigma(v_0 ... v_n) \subseteq v_n E$. A play $p = (v_n)_{n \in \mathbb{N}} \in V^\omega$ is said to be consistent with σ if, for any n such that $v_n \in V_i$, $v_{n+1} \in \sigma(v_0 \cdots v_n)$. Given a subset $X \subseteq V$ of the vertices, A strategy for player i is said to be winning the game (G, Vic) on X if any infinite play starting in X and consistent with this strategy is won by player i. If there exists such a strategy, we say that player i wins (G, Vic) on X. If $X = V$, we simply say that i wins (G, Vic). The winning set of player i is the greatest set of vertices such that i wins Vic on this set.

Positional Strategies. With certain strategies, the choices advised to the player depend only on the current vertex. Such a strategy can be simply described by the set of edges it allows the players to use. $\sigma \subseteq E$ is a positional strategy for player i in the arena G if there is no dead-end in the subgraph $(Dom(\sigma), \sigma)$ induced by σ and σ does not restrict the moves of the adversary: if $v \in V_{1-i} \cap Dom(\sigma)$ then $\{v\} \times vE \subseteq \sigma$. Let $X \subseteq V$ be a subset of vertices. If $Dom(\sigma) = X$, σ is said to be defined on X. We say that a player wins positionally a game Vic on X if he has a positional strategy winning on X.

Subarenas and Traps. Let $X \subseteq V$ be a subset of vertices and $F \subseteq E$ a subset of edges. $G[X]$ denotes the graph $(X, E \cap X^2)$ and $G[X, F]$ denotes the graph $(Dom(F) \cap X, F \cap X^2)$. When $G[X]$ or $G[X, F]$ is an arena, it is said to be a subarena of G. X is said to be a trap for player i in G if $G[X]$ is a subarena and player i can't move outside of X, i.e. $\forall v \in X \cap V_i, vE \subseteq X$.

2.2 Winning conditions.

Let $G = (V, V_0, V_1, E)$ be an arena and $X \subseteq V$. We define various winning conditions.

Attraction game to X. Player 0 wins if the pebble visits X at least once. The corresponding winning condition is $Attraction(X) = V^*XV^\omega$. The winning set for player 0 is denoted by $Att_0(G, X)$ or $Att_0(X)$, when G is obvious. Symmetrically, we define $Att_1(G, X)$ and $Att_1(X)$, the sets of vertices where player 1 can attract the pebble to X. Note that for this game, both players have positional winning strategies.

Trap game and Büchi game to X. Player 0 wins the trap game in X if the pebble stays ultimately in X. The winning condition is $TrapX = V^*X^\omega$. The dual game is the Büchi game to X, where player 0 wins if the pebble visits X infinitely often. The winning condition is $Buchi(X) = (V^*X)^\omega$.

Exploration game. This is a game over an infinite graph, where player 0 wins a play if the pebble visits infinitely many different vertices. The winning condition is $Exp = \{v_0v_1 \cdots \in V^\omega \mid \text{ the set } \{v_0, v_1, \ldots\} \text{ is infinite}\}$.

The exploration condition is an extension of the *Unboundedness* condition introduced in [3]. The Unboundedness condition concerns games played on the configuration graph of a pushdown system. On such a graph, the set of plays is exactly the set of runs of the underlying pushdown automaton, and 0 wins a play if the height of the stack is unbounded, which happens if and only if infinitely many different configurations of the pushdown automaton are visited.

The exploration condition is also closely related to the Σ_3-condition considered in [5], which states that 0 wins a play if every vertex is visited finitely often. Notice that such a play is necessarily also winning for the exploration condition, but the converse is not true. However, given an arena, it is easy to see that each player has the same positional winning strategies for both games. Since the Exploration game is won positionally by both players (cf. Proposition 1), it implies both games have the same winning positions. Hence, in that sense, the Explosion game and the Σ_3-game introduced in [5] are equivalent.

Parity game. G is a parity arena equipped with $\phi : V \to \mathbb{N}$. Player 0 wins a play if there exists a highest priority visited infinitely often and this priority is even, or if the sequence of priorities is unbounded. The winning condition is

$$Parity_\infty = \{(v_i)_{i\in\mathbb{N}} : \overline{\lim_{i\in\mathbb{N}}}\ \phi(v_i) \in \{0, 2, \ldots, +\infty\}\}$$

where $\overline{\lim}_{i\in\mathbb{N}}\ \phi(v_i) = \lim_{i\in\mathbb{N}}\ \sup_{j\geq i}\ \phi(v_j)$ denotes the limit sup of the infinite sequence of visited priorities. If G is in fact labeled with a finite number of priorities, i.e. if there exists $d \in \mathbb{N}$ such that $\phi : V \to [0, d]$, we write also the winning condition as $Parity_d$. In this case, a classical result [9,19,23] states that both players win this game positionally.

3 Playing the games $Exp \cup Parity_\infty$ and $Exp \cap Parity_d$.

In this section we study the winning strategies for the games $Exp \cup Parity_\infty$ and $Exp \cap Parity_d$. We show that both players win positionally $Exp \cup Parity_\infty$. Concerning the game $Exp \cap Parity_d$, we show that player 1 wins it positionally and we give a characterization of the winning set of player 0.

3.1 The game $Exp \cup Parity_\infty$.

G is a parity arena equipped with $\phi : V \to \mathbb{N}$.

Proposition 1. *Each player wins positionally the game* $\mathrm{Exp} \cup \mathrm{Parity}_\infty$ *on his winning set.*

Proof. It is crucial to observe that $Exp \cup Parity_\infty$ can be expressed as the limit of a decreasing sequence of winning conditions:

$$Exp \cup Parity_\infty = \bigcap_{n\in\mathbb{N}} Vic_n,$$

where

$$Vic_n = Attraction(\{n+1, n+2, \ldots\}) \cup Parity_\infty.$$

Moreover, each game (G, Vic_n) is won positionally by players 0 and 1 on their winning sets X_n and $V\backslash X_n$. It is easy to establish that player 1 wins positionally $(G, \bigcap_n Vic_n)$ on $\bigcup_n V\backslash X_n = V\backslash \bigcap_n X_n$. For winning positionally $\bigcap_n Vic_n$ on $\bigcap_n X_n$, player 0 can manage to play in such a way that, as long as the pebble stays in $\{0, 1, \ldots, n\}$, the play is consistent with a winning strategy for Vic_n. Then, if the pebble stays bounded in some set $\{n, n+1, \ldots\}$, the play is won for condition $\bigcap_{m\geq n} Vic_m \subset Parity_\infty$. If the pebble goes out of every set $\{0, \ldots, n\}$, then the play visits infinitely many different vertices and the play is won for *Exp*. □

Since the *Exp* game is a special case of the $Exp \cup Parity_\infty$ game where all the vertices are labeled with priority 1, we get the following corollary.

Corollary 1. *Each player wins positionally the game* Exp *on his winning set.*

3.2 The game $Parity_\infty$.

A natural question that arises is whether the players have some positional strategies for the $Parity_\infty$ game. Notice that $Exp \subseteq \mathrm{Parity}_\infty$ in the special case where, for every priority d, $\phi^{-1}(d)$ is finite. Indeed, any play visiting infinitely many different vertices will visit infinitely many different priorities.

Hence, in this special case, by Proposition 1, the game Parity_∞ is won positionally by both players. It is not true anymore if $\phi^{-1}(d)$ is infinite for some d. Consider the example given by figure 1. The circles are the vertices of player 0 and the squares those of player 1. Player 0 wins Parity_∞ from everywhere but has no positional winning strategy.

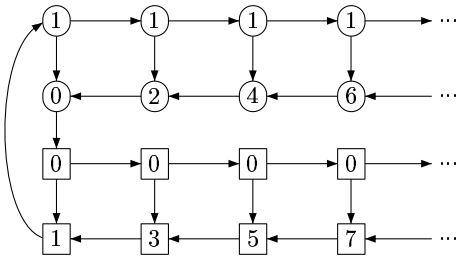

Fig. 1. Player 0's strategy must recall the highest odd vertex reached by player 1 in the lower row in order to answer with a higher even vertex in the second row.

It is interesting to note that if the winning player is determined by the lowest priority visited infinitely often rather than by the greatest one, then both players have positional winning strategies, even if countable many priorities are assumed [11].

3.3 The $Exp \cap Parity_d$ game

The analysis of the $Exp \cap Parity_d$ game is an extension of the results of [23]. In this section, G is a parity arena equipped with $\phi : V \to [0, d]$.

Proposition 2. *Player* 1 *wins positionally the game* $\mathrm{Exp} \cap \mathrm{Parity}_d$ *on his winning set.*

Proof. Without loss of generality, we can assume that player 1 wins everywhere. The proof is by induction on d.

If $d = 0$, it is impossible for player 1 to win any play and his winning set is empty. If d is odd and $d \neq 0$, let W be the attractor for player 1 in the set of vertices coloured by the maximal odd priority d. Since $V \backslash W$ is a trap for player 1 coloured from 0 to $d - 1$, and by inductive hypothesis, player 1 can win positionally $(G[V \backslash W], Exp \cap Parity_{d-1})$ with some strategy $\sigma_{V \backslash W}$. To win, player 1 shall use $\sigma_{V \backslash W}$ inside $V \backslash W$ and shall attract the pebble in a vertex of

colour rd when it reaches the set W. That way, either the plays stay ultimately in $V \backslash W$ and some suffix is consistent with $\sigma_{V \backslash W}$ or it reaches the odd priority d infinitely often. In both cases, player 1 is the winner.

The case where d is even is less trivial. It is easy to prove that there exists the greatest subarena of G where player 1 wins positionally. It remains to prove that this subarena coincides with the whole subarena.

Player 0 does not necessarily have positional winning strategies. For example, in the game of figure 2, player 0 wins $Exp \cap Parity_d$ from any vertex but has no positional winning strategy.

Fig. 2. To win the $Exp \cap Parity_2$ game, player 0 has to visit new vertices arbitrarily far to the right hand side of the arena and has also to visit the unique vertex of color 2 infinitely often.

Nevertheless, we can characterize the arenas in which player 0 wins the game $Exp \cap Parity_d$ from everywhere:

Proposition 3. *Let $G = (V, E)$ be an arena, coloured from 0 to $d > 0$. Let D be the set of vertices coloured by d. Player 0 wins the game $(G, \mathrm{Exp} \cap \mathrm{Parity}_d)$ on V if and only if there exists a subarena $G[W]$, coloured from 0 to $d-1$ such that one of the following conditions holds:*

- **Case d even:** *Player 0 wins the games $(G[W], \mathrm{Exp} \cap \mathrm{Parity}_{d-1})$ and (G, Exp) everywhere and she wins the game $(G, Attraction(D))$ on $V \backslash W$.*
- **Case d odd:** *Player 0 wins the game $(G, Trap(W))$ with a positional strategy $\sigma_{Trap(W)}$ and wins the game $(G[W, \sigma_{Trap(W)}], \mathrm{Exp} \cap \mathrm{Parity}_{d-1})$.*

The conditions of Proposition 3 are illustrated by figure 3.

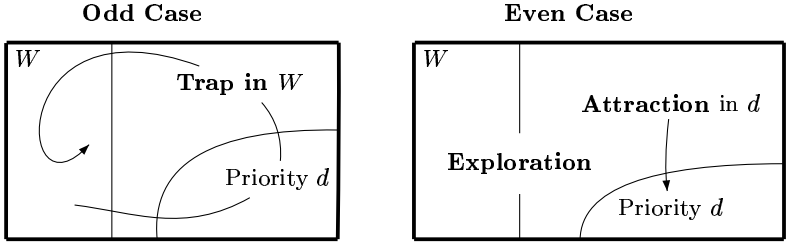

Fig. 3. Conditions of Proposition 3.

Remark 1. Note that winning the game $(G[W, \sigma_{Trap(W)}], Exp \cap Parity_{d-1})$ means that player 0 has a strategy σ_W winning the game $(G[W], Exp \cap Parity_{d-1})$ which advises player 0 to play moves **consistent with the positional strategy** $\sigma_{\mathbf{Trap(W)}}$.

Proof. We sketch the proof of the direct implication. In the case where d is even this proof is simple. Consider $W = V \backslash Att_0(D)$. Since $V \backslash W$ is a trap, player 0 wins $(G[V \backslash W], Exp \cap Parity_d)$. The other claims are trivially true.

The case where d is odd is more tricky. We establish first that the family of subarenas of G where Proposition 3 holds is closed by arbitrary union, then we prove that the maximal such arena is necessarily G itself.

We sketch the proof of the converse implication. We shall construct a strategy σ_G for player 0 winning the game $(G, Exp \cap Parity_d)$. This construction depends on the parity of d.

d Odd. By hypothesis, player 0 has a positional strategy $\sigma_{Trap(W)}$ winning the game $(G, Trap(W))$ and a strategy σ_{Sub} winning the game $(G[W, \sigma_{Trap(W)}], Exp \cap Parity_{d-1})$. σ_G is constructed in the following way:

- If the pebble is not in W, player 0 plays according to her positional strategy $\sigma_{Trap(W)}$.
- If the pebble is in W, player 0 uses her strategy σ_{Sub} in the following way: Let p be the sequence of vertices visited up to now and let p' be the longest suffix of p consisting of vertices of W. Player 0 takes a move according to $\sigma_{Sub}(p')$.

The strategy σ_G is winning the game $(G, Exp \cap Parity_d)$. Indeed, since σ_{Sub} is a strategy in the arena $G[W, \sigma_{Trap(W)}]$, all the moves consistent with σ_G are consistent with $\sigma_{Trap(W)}$. Hence, the play is ultimately trapped in W and is ultimately consistent with σ_W, thus won by player 0.

d Even. By hypothesis and by Corollary 1, player 0 has a positional strategy $\sigma_{Exp} \subseteq E$ winning (G, Exp). She has also a positional strategy $\sigma_{Att} \subseteq E$ winning $(G, Attraction(D))$ on $V \backslash W$ and a strategy σ_{Sub} winning the game $(G[W], Exp \cap Parity_{d-1})$.

σ_G is constructed in the following way. At a a given moment player 0 is in one of the three playing modes: *Attraction*, *Sub* or *Exploration*. It can change the mode when the pebble moves to a new vertex. Player 0 begins to play in *Exploration* mode. Here follows the precise description of the strategy σ_G, summarized by figure 4.

- The playing mode *Exploration* can occur wherever the pebble is. Player 0 plays according to her positional strategy σ_{Exp}. When a new vertex v is visited for the first time the mode is changed. If $v \in W$, it is changed to *Sub* mode. If $v \notin W$, it is changed to *Attraction* mode.
- The playing mode *Attraction* can occur only if the pebble is not in W. Player 0 plays according to her positional strategy σ_{Att}. When a vertex of priority d is visited, the playing mode is set to *Exploration*.

- The playing mode *Sub* can occur only if the pebble is in W. Player 0 plays using her strategy σ_{Sub} in the following way. Let p be the sequence of vertices visited up to now and p' the longest suffix of p consisting of vertices of W. Then 0 takes a move according to $\sigma_{Sub}(p')$. If the pebble goes out of W, the playing mode is set to *Exploration*.

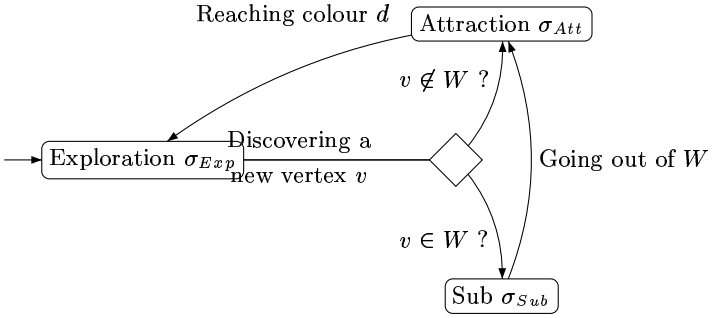

Fig. 4. Rules of transition between playing modes.

Notice that, by definition of σ_{Att} and σ_{Exp}, it is not possible that an infinite play consistent with σ_G stays forever in the playing modes *Attraction* or *Exploration*. Hence, such a play can be of two different types. Either the pebble stays ultimately in the playing mode *Sub* or it goes infinitely often in the modes *Exploration* and *Attraction*. In the first case, it stays ultimately in W and the play is ultimately consistent with σ_{Sub}. In the second case, the pebble visits infinitely often the even priority d and discovers infinitely often a new vertex. In both cases, this play is won by player 0 for the $Exp \cap Parity_d$ condition. □

4 Computation of the winning sets and strategies on pushdown arenas.

In this section, we apply our results to the case where the infinite graph is the graph of the configurations of a pushdown automaton. In this way, we get an algorithm to compute the winning sets. Moreover, in all cases except for player 0 in the game $Exp \cap Parity_d$, we can also compute some winning positional strategies.

Definitions. A pushdown system is a tuple $\mathcal{P} = (Q, \Gamma, \Delta, \bot)$ where Q is a finite set of control states, Γ is a finite stack alphabet, $\bot$ is a special letter called the stack bottom, $\bot \notin \Gamma$ and $\Delta \subseteq Q \times (\Gamma \cup \{\bot\}) \times (\Gamma \cup \{-1\}) \times Q$ is the set of transitions.

The transition $(q, \alpha, \beta, r) \in \Delta$ is said to be a *push transition* if $\beta \in \Gamma$ and a *pop transition* if $\beta = -1$. In both cases, it is said to be an α-transition and a (q, α)-transition. Concerning $\bot$, we impose the restriction that there exists no

pop $\perp$-transition. Moreover, we work only with complete pushdown systems, in the sense that, for every couple $(q, \alpha) \in Q \times (\Gamma \cup \{\perp\})$, there exists at least one (q, α)-transition.

Notice that, in the sense of language recognition, any pushdown automaton is equivalent to one of this kind, and the reduction is polynomial.

A configuration of $\mathcal{P}$ is a sequence $q\gamma$, where $q \in Q$ and $\gamma \in \Gamma^*$. Intuitively, q represents the current state of $\mathcal{P}$ while γ is the stack content above the bottom symbol $\perp$. We assume that the symbols on the right of γ are at the top of the stack. Note that $\perp$ is assumed implicitly at the bottom of the stack, i.e. actually the complete stack content is always $\perp\gamma$.

The set of all configurations of $\mathcal{P}$ is denoted by $V_{\mathcal{P}}$. Transition relation $E_{\mathcal{P}}$ over configurations is defined in the usual way: Let $q\gamma\alpha$, $\alpha \in \Gamma$, be a configuration.

- $(q\gamma\alpha, r\gamma) \in E_{\mathcal{P}}$ if there exists a pop transition $(q, \alpha, -1, r) \in \Delta$,
- $(q\gamma\alpha, r\gamma\alpha\beta) \in E_{\mathcal{P}}$ if there exists a push transition $(q, \alpha, \beta, r) \in \Delta$.

Let $q\epsilon$ be a configuration with empty stack.

- $(q\epsilon, r\beta) \in E_{\mathcal{P}}$ if there exists a push transition $(q, \perp, \beta, r) \in \Delta$.

We will write $q\gamma \xrightarrow{\delta} r\gamma'$ to express that a transition $\delta \in \Delta$ of the pushdown automaton corresponds to an edge $(q\gamma, r\gamma') \in E_{\mathcal{P}}$ between two configurations. The graph $G_{\mathcal{P}} = (V_{\mathcal{P}}, E_{\mathcal{P}})$ is called the *pushdown graph* of $\mathcal{P}$.

If Q is partitioned in (Q_0, Q_1), this partition extends naturally to the set of configurations of $\mathcal{P}$ and $\mathcal{G}_{\mathcal{P}}$ is an arena. Moreover, when the control states Q are labeled by priorities with a map $\phi : Q \to [0, d]$, this labeling extends naturally to $V_{\mathcal{P}}$ by setting $\phi(q\gamma) = \phi(q)$. $G_{\mathcal{P}}$ is then a parity arena.

Subgraph trees and strategy trees. With any subset $\sigma \subseteq E_{\mathcal{P}}$ of the edges of a pushdown arena we associate a tree $T_\sigma : \Gamma^* \to 2^\Delta$ with vertices labeled by sets of transition of $\mathcal{P}$. This construction is illustrated by figure 5.

A vertex of the tree is a stack content of $\mathcal{P}$. A transition $\delta \in \Delta$ is in the label of a vertex $\gamma \in \Gamma^*$ if there exists some state $q \in Q$ and some configuration $r\gamma'$ such that $q\gamma \xrightarrow{\delta} r\gamma'$ and $(q\gamma, r\gamma') \in \sigma$. Such a tree is called the *coding tree* of σ. Notice that the transformation $\sigma \to T_\sigma$ is one-to-one. If σ is a strategy for player i, we call T_σ a *strategy tree* for player i.

The next theorem states that the languages of positional winning strategies is regular. Thus, we can build a Büchi alternating automaton of size $\mathcal{O}(d|Q|^2 + |\Gamma|)$ which recognizes the language of couples (σ_0, σ_1) such that σ_i is a winning positional strategy for player i and the domains of σ_0 and σ_1 are a partition of $V_{\mathcal{P}}$. In the case of the $Parity_d$ and $Exp \cup Parity_d$ games, Proposition 1 establish that this language is non-empty. Hence it is possible to compute a regular tree (σ_0, σ_1) of size $2^{\mathcal{O}(d|Q|^2+|\Gamma|)}$. This regular tree can be seen as the description of a couple of winning stack strategies for both players. This kind of strategy has been defined in [21].

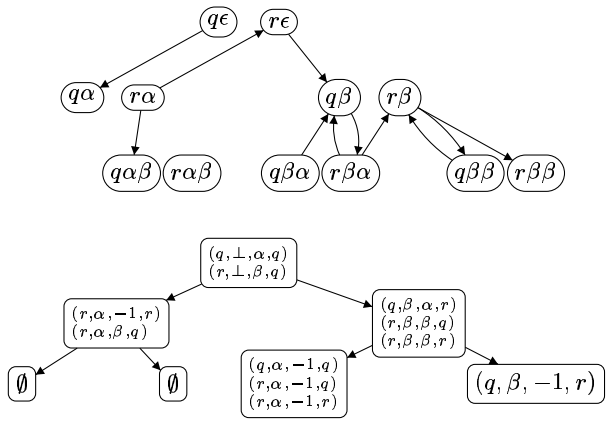

Fig. 5. A finite subset of $E_{\mathcal{P}}$ and its coding tree. Only the labels of the vertices $\{\epsilon, \alpha, \beta, \alpha\alpha, \alpha\beta, \beta\alpha, \beta\beta\}$ are represented. Other vertices of the coding tree are labeled with $\emptyset$.

Theorem 1. *Let i be a player and* $\mathrm{Vic} \in \{\mathrm{Parity}_d, \mathrm{Exp}\cup\mathrm{Parity}_d, \mathrm{Exp}\cap\mathrm{Parity}_d\}$. *The language of strategy trees which correspond to winning positional strategies for player i is regular. One can effectively construct an alternating Büchi automaton $\mathcal{A}_{\mathrm{Vic},i}$ with $\mathcal{O}(d|Q|^2 + |\Gamma|)$ states which recognizes it.*

Proof. The construction of $\mathcal{A}_{\mathrm{Vic},i}$ uses techniques close to the one of [20,16]. Unfortunately, we couldn't manage to make use of the results of those papers about two-way tree automata, because we don't know how to use a two-way automata to detect a cycle in a strategy tree.

Our aim is to construct a tree automaton recognizing a tree $t : \Gamma^* \to 2^\Delta$ iff there exists a winning positional strategy σ such that $t = T_\sigma$. In fact we shall rather construct a Büchi alternating automaton recognizing the complement of the set $\{T_\sigma | \sigma$ winning positional strategy $\}$. First of all it is easy to implement an alternating automaton verifying if the tree t is or is not a strategy tree. It is less trivial to construct the automaton checking if a positional strategy $\sigma \subset E_{\mathcal{P}}$ is winning or not. However, it can be expressed by simple criterion concerning the cycles and the exploration paths of the graph $(Dom(\sigma), \sigma)$ induced by σ. Those criteria are summarized in table 1.

We have to construct automata checking each condition of table 1. They are derived from an automata able to detect the existence of a special kind of finite path called a *jump*. A jump between two vertices with the same stack γ is a path between those vertices, that never pop any letter of γ (see figure 4).

This kind of path is interesting since a cycle is simply a jump from a vertex to itself, and because the existence of an exploration path of priority c is equivalent to the existence of one of the two kinds of paths illustrated by figure 7.

Due to the high computational power of alternation, it is possible to construct automata checking the existence of jumps and detecting the kinds of paths of figure 7, and which have only $\mathcal{O}(d|Q^2| + |\Gamma|)$ control states. □

Winning Condition	i	Condition on cycles	Condition on exploration paths
Parity	0	Even	Even
	1	Odd	Odd
$Exp \cup Parity_d$	0	Even	No condition
	1	Odd	No exploration path
$Exp \cap Parity_d$	0	No cycle	Even
	1	No condition	Odd

Table 1. Characterization of winning positional strategies.

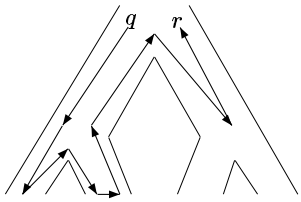

Fig. 6. A jump from $q\gamma$ to $r\gamma$ in a strategy tree.

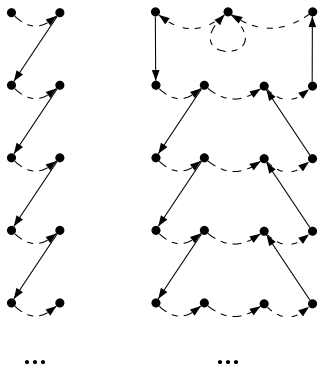

Fig. 7. The dotted arrows are jumps of priority less than c. The top-down regular arrows are push transitions, while the down-top ones are pop-transitions. On the left hand side, infinitely many of the jumps have priority c. On the right hand side, the upper jump, which is a loop, has priority c.

Computation of winning sets. Using the automata recognizing languages of winning positional strategies, it is possible to recognize the language of winning positions. For each player i, Theorem 2 leads to an EXPTIME procedure to compute a regular tree $\Gamma^* \to 2^Q$ of exponential size that associates with a stack γ the set $\{q \in Q : q\gamma \text{ is winning for player } i\}$. Once computed, deciding if a given position is winning for player i can be done in linear time.

Theorem 2. . *For each player i and winning condition* $\mathrm{Vic} \in \{\mathrm{Parity}_d, \mathrm{Exp} \cup \mathrm{Parity}_d, \mathrm{Exp} \cap \mathrm{Parity}_d\}$, *the tree $\Gamma^* \to 2^Q$ which associates with a stack γ the set $\{q \in Q : q\gamma$ is winning for $i\}$ is regular. One can compute a non-deterministic Büchi automata which recognizes it, whose size is $2^{\mathcal{O}(d|Q|^2+|\Gamma|)}$ if* $\mathrm{Vic} \in \{\mathrm{Parity}_d, \mathrm{Exp} \cup \mathrm{Parity}_d\}$ *and $2^{\mathcal{O}(d^2|Q|^2+d|\Gamma|)}$ if* $\mathrm{Vic} = \mathrm{Exp} \cap \mathrm{Parity}_d$.

Proof. For the games $Parity_d$ and $Exp \cup Parity_d$, this Theorem is a direct corollary of Theorem 1. In fact, we can build a Büchi alternating automaton which recognizes the language of couples (σ_0, σ_1) such that σ_i is a winning positional strategy for player i and the domains of σ_0 and σ_1 are a partition of $V_{\mathcal{P}}$. The winning sets are then obtained by projection, which requires a non-determinization, hence an exponential blowup of the state space of the alternating automaton [15].

In the case of the $Exp \cap Parity_d$ game, we use also the characterization of the winning sets given by Proposition 3. We define the notion of winning-proof, which is a tree on Γ^* labeled by tuples of subsets of Δ, and is defined such that the existence of a winning-proof in an arena is equivalent to the conditions of Proposition 3. Here follows the definition of a winning-proof in a subarena G of a pushdown arena $G_{\mathcal{P}}$.

In the case where $d = 0$, it is a strategy tree $T_{\sigma_{Exp}}$ winning the game (G, Exp).

In the case where $d > 0$ and is even, it is a tuple $T_d = (T', T_{\sigma_{Exp}}, T_{\sigma_{Att}}, T_{d-1})$ where

- T' is the coding tree of a subarena G' of G,
- $T_{\sigma_{Exp}}$ is a strategy tree winning the game (G, Exp),
- $T_{\sigma_{Att}}$ is a strategy tree winning the game $(G, Attraction(D))$ on $Dom(G')$,
- T_{d-1} is a $(d-1)$-winning proof in G'.

In the case where d is odd, it is a tuple $T_d = (T', T_{\sigma_{Trap}}, T_{d-1})$ where

- T' is the coding tree of a subarena G' of G,
- $T_{\sigma_{Trap}}$ is a strategy tree winning the game $(G, Trap(Dom(G')))$,
- T_{d-1} is a $(d-1)$-winning proof in G'.

Each one of those conditions $\mathcal{O}(d)$ is checkable with some alternating automaton with $d|Q|^2 + |\Gamma|$ states. The corresponding automata constructions are very close to the ones of Theorem 1. Hence, the language of d-winning proofs is regular and recognized by an alternating automata with $\mathcal{O}(d^2|Q|^2 + d|\Gamma|)$ states.

As in the positional case, by projection, we obtain the desired non-deterministic automaton with $2^{\mathcal{O}(d^2|Q|^2+d|\Gamma|)}$ states. □

5 Conclusions.

The framework of Zielonka has enabled us to prove some characterizations of the $Exp \cap Parity$ and $Exp \cup Parity$ games played on infinite graphs of finite degree. In the special case of pushdown graphs, we have presented algorithms to compute efficiently

References

1. A. Arnold, A. Vincent, and I. Walukiewicz. Games for synthesis of controlers with partial observation. *Theoretical Computer Science*, 303(1):7–34, 2003.
2. A. Bouajjani, J. Esparza, and O. Maler. Reachability analysis of pushdown automata: Applications to model checking. In *CONCUR'97, LNCS*, volume 1243, pages 135–150, 1997.
3. A. Bouquet, O. Serre, and I. Walukiewicz. Pushdown games with unboudedness and regular conditions. In *Proc. of FSTTCS'03, LNCS*, volume 2914, pages 88–99, 2003.
4. T. Cachat. Symbolic strategy for games on pushdown graphs. In *Proc. of 29th ICALP, LNCS*, volume 2380, pages 704–715, 2002.
5. T. Cachat, J. Duparc, and W. Thomas. Pushdown games with a σ_3 winning condition. In *Proc. of CSL 2002, LNCS*, volume 2471, pages 322–336, 2002.
6. K. Chatterejee, M. Jurdziński, and T.A. Henzinger. Simple stochastic parity games. In *CSL'03*, volume 2803 of *LNCS*, pages 100–113. Springer, 2003.
7. L. de Alfaro and T.A. Henzinger. Concurrent ω-regular games. In *LICS'00*, pages 142–154. IEEE Computer Society Press, 2000.
8. L. de Alfaro, T.A. Henzinger, and O. Kupferman. Concurrent reachability games. In *FOCS'98*, pages 564–575. IEEE Computer Society Press, 1998.
9. E. A. Emerson and C. S. Jutla. Tree automata, mu-calculus and determinacy. In *Proc. of 32th FOCS*, pages 368–377. IEEE Computer Society Press, 1991.
10. H. Gimbert. Parity and exploration games on infinite graphs, full version. www.liafa.jussieu.fr/~hugo.
11. E. Grädel. Positional determinacy of infinite games. In *STACS 2004, LNCS*, volume 2996, pages 4–18, 2004.
12. O. Kupferman, N. Piterman, and M. Vardi. Pushdown specifictaions. In *Proc. of LPAR'02, LNCS*, October 2002.
13. O. Kupferman and M.Y. Vardi. An automata-theoretic approach to reasoning about infinite state systems. In *Proc. of CAV'00, LNCS*, volume 1855, pages 36–52, 2000.
14. D. E. Muller and P. E. Schupp. The theory of ends, pushdown automata and second order logic. *Thoretical Computer Science*, 37:51–75, 1985.
15. D. E. Muller and P. E. Schupp. Simulating alternating tree automata by nondeterministic automata. *Theoretical Computer Science*, 141:69–107, 1995.
16. N. Piterman and M.Vardi. From bidirectionnality to alternation. *Theoretical Computer Science*, 295:295–321, 2003.
17. O. Serre. Games with winning conditions of high borel complexity, to appear in proc. of icalp'04.
18. O. Serre. Note on winning positions on pushdown games with omega-regular conditions. *Information Processing Letters*, 85(6):285–291, 2003.

19. W. Thomas. On the synthesis of strategies in infinite games. In *Proc. of STACS'95,LNCS*, volume 900, pages 1–13, 1995.
20. M. Vardi. Reasoning about the past with two-way automata. In *Proc. of ICALP'98, LNCS*, volume 1443, pages 628–641, 1998.
21. I. Walukiewicz. Pushdown processes: Games and model checking. In *Proc. of CAV'96, LNCS*, volume 1102, pages 62–74, 1996.
22. I. Walukiewicz. Pushdown processes: Games and model checking. *Information and Computation*, 164(2):234–263, 2001.
23. W. Zielonka. Infinite games on finitely coloured graphs with applications to automata on infinite trees. *Theoretical Computer Science*, 200:135–183, 1998.

Appendix

A Playing the games $Exp \cup Parity_\infty$ and $Exp \cap Parity_d$.

A.1 Elementary lemmas.

Throughout this appendix, we shall need the following elementary two lemmas.

Lemma 1. *Let T be a trap for player i and $X \subseteq T$.*
Any strategy for i winning (G, Vic) on X wins $(G[T], \mathrm{Vic})$ on X.
Any strategy for $1-i$ winning $(G[T], \mathrm{Vic})$ on X wins (G, Vic) on X.

Lemma 2. *If $\mathrm{Vic} \subseteq V^\omega$ is concatenation closed, the winning set of player 1 is a trap for player 0.*

A.2 Tools for building positional winning strategies.

The following lemmas are useful to prove the existence of positional winning strategies. The first one is used in [23].

Lemma 3. *Let Vic be concatenation closed. Let $W \subseteq V$. If player 0 wins positionally $(G[V \setminus Att_0(W)], \mathrm{Vic})$, then she wins positionally $(G, \mathrm{Vic} \cup \mathrm{Buchi}(W))$.*

Lemma 4. Construction of positional strategies by intersection
Let $(\mathrm{Vic}_n)_{n\in\mathbb{N}}$ be a decreasing sequence of winning conditions and $(X_n)_{n\in\mathbb{N}}$ a sequence of subsets of V such that for any $n \in \mathbb{N}$, player 0 wins positionally the game Vic_n on X_n. Then player 0 wins positionally the game $Exp \cup \bigcap_{n\in\mathbb{N}} \mathrm{Vic}_n$ on $\bigcap_{n\in\mathbb{N}} X_n$.

Proof. It is sufficient to prove this lemma in the case where G is connected. Since G is of finite degree, V is countable and we can suppose that $V = \mathbb{N}$. We shall construct a strategy σ for player 0 winning the game $Exp \cup \bigcap_{n\in\mathbb{N}} Vic_n$ on $\bigcap_{n\in\mathbb{N}} X_n$. The main idea is that 0 will always be playing consistently with an infinite number of positional strategies, each one winning one of the games Vic_n.

To make this construction nicely, we use Koenig's Lemma in a tree whose vertices are positional strategies. Let $Strat_n^0$ be the set of positional strategies winning for 0 on $X = \bigcap_{n\in\mathbb{N}} X_n$ for condition Vic_n, and

$$Strat_n = \{(\sigma \cap [0, n-1]^2, n) : \sigma \in Strat_n^0\}$$

their restrictions to the set of vertices $< n$. Let also

$$Strat = \bigcup_{n\in\mathbb{N}} Strat_n$$

We give to $Strat$ a structure of a directed graph by adding an edge between a vertex $(\sigma, n) \in Strat_n$ and a vertex $(\tau, n+1) \in Strat_{n+1}$ if σ is the restriction of τ to $[0, n-1]^2$. Let $(Strat, F)$ be this graph.

It is straightforward to check that $(Strat, f)$ is a directed tree whose root is $(\emptyset, 0)$ and whose set of vertices of depth n is $Strat_n$. Since, by hypothesis, each $Strat_n^0$ is nonempty, $Strat$ is infinite. Moreover, since for any integer $n \in \mathbb{N}$ $Strat_n \subset 2^{[0,n-1]^2} \times \{n\}$, $(Start, f)$ is a tree of finite degree.

By Koenig's lemma, there exists an infinite path $(\sigma_0, 0), (\sigma_1, 1), ...$ in $(Strat, F)$ starting from the root $(\emptyset, 0)$.

Then it is easy to check that

$$\sigma = \bigcup_{n \in \mathbb{N}} \sigma_n$$

is a positional strategy for player 0, defined at least on X. We shall prove that σ wins $Exp \cup \bigcap_{n \in \mathbb{N}} Vic_n$ on X.

Let p be consistent with σ and starting in X. If p is winning for the game Exp, it is, by hypothesis, won by 0. In the opposite case, it stays in a set $[0, k]$ for some integer k. For $n \in \mathbb{N}$, let $\sigma_n^0 \in Strat_n^0$ such that $\sigma_n = \sigma_n^0 \cap [0, n-1]^2$. Since $\sigma_k \subseteq \sigma_{k+1} \subseteq ...$, the play p is consistent with every σ_n^0, for $n > k$. Thus, $p \in \bigcap\limits_{n \geq k} Vic_n = \bigcap_{n \in \mathbb{N}} Vic_n$. □

Lemma 5. Construction of positional strategies by union.

Let $(\mathrm{Vic}_n)_{n \in I}$ *be a family of winning conditions and* $(Y_n)_{n \in I}$ *a family of subsets of* V *such that for any* $n \in I$*, player* 0 *wins positionally the game* Vic_n *on* Y_n *and such that* $\bigcup_{n \in I} \mathrm{Vic}_n$ *is closed by concatenation. Then player* 0 *wins positionally the game* $\bigcup_{n \in I} \mathrm{Vic}_n$ *on* $\bigcup_{n \in I} Y_n$.

Proof. Let $\preceq$ be a well founded order on I. For each $n \in I$, let σ_n be a positional strategy winning on Y_n the game Vic_n. We can easily build a positional strategy σ winning on $\bigcup\limits_{n \in I} Y_n$ the game $Vic = \bigcup\limits_{n \in I} Vic_n$. For any vertex v, define $index(v) = \min_{\preceq}\{n \in I : v \in Y_n\}$. On a vertex v, σ consists to take a move in $\sigma_{index(v)}(v)$.

Let $p = v_0 v_1 \ldots$ be a play consistent with σ. The Y_n induce traps for 1, thus the sequence $index(v_0), index(v_1), \ldots$ is a $\preceq$-decreasing,hence stationary equal to an index m from a rank N. Since p is ultimately consistent with σ_m, some suffix of p is a play won by 0 for the condition $Vic_m \subset Vic$. The closeness of Vic by concatenation implies that p is won by 0. Finally, σ is a positional strategy winning the game Vic. □

A.3 The game $Exp \cup Parity_\infty$

Proposition 1. *Each player wins positionally the game* $\mathrm{Exp} \cup \mathrm{Parity}_\infty$ *on his winning set.*

Proof. It is sufficient to prove this proposition in the case where G is connected. Since G is of finite degree, V is countable and we can suppose that $V = \mathbb{N}$. Consider, for each vertex $n \in \mathbb{N}$, the game Vic_n where player 0 wins if she reaches a vertex strictly greater than n or if she wins the Parity_∞ game.

$$\mathrm{Vic}_n = \mathit{Attraction}([n+1, +\infty]) \cup \mathrm{Parity}_\infty$$

Notice that player 0 wins a play of the exploration game if and only if the pebble reaches arbitrarily high vertices, hence $\bigcap_{n\in\mathbb{N}} \mathrm{Vic}_n = \mathit{Exp} \cup \mathrm{Parity}_\infty$. For each $n \in \mathbb{N}$, we are going to construct a set of vertices $X_n \subseteq \mathbb{N}$ such that player 0 and player 1 win positionally the game Vic_n on X_n and $V \backslash X_n$, respectively. According to Lemmas 4 and 5, it implies that player 0 and player 1 win positionally $\mathit{Exp} \cup \mathrm{Parity}_\infty$ on $\bigcap_{n\in\mathbb{N}} X_n$ and $V \backslash \bigcap_{n\in\mathbb{N}} X_n$, respectively, which proves Proposition 1.

Construction of $\mathbf{X_n}$. For every subset $W \subseteq \mathbb{N}$ such that $G[W]$ is a subarena, let $\mathit{Par}_0(W)$ denote the winning set of player 0 for the game $(G[W], \mathrm{Parity}_\infty)$. The construction of X_n is illustrated by figure 8.

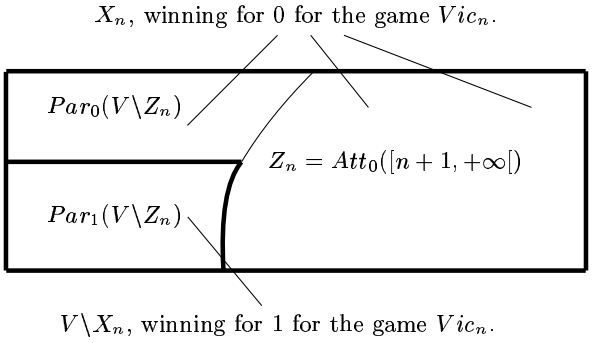

Fig. 8. Construction of X_n.

Since $V \backslash Z_n$ is finite, the game $(G[V \backslash Z_n], \mathrm{Parity}_\infty)$ is the classical parity game with a finite number of priorities and both players win it positionally. Moreover, since $V \backslash Z_n$ is a trap for player 0, and according to Lemma 1, the positional strategy for player 1 that wins $(G[V \backslash Z_n], \mathrm{Parity}_\infty)$ on $\mathit{Par}_1(V \backslash Z_n)$ wins also $(G, \mathrm{Parity}_\infty)$ on $\mathit{Par}_1(V \backslash Z_n)$. The positional strategy for player 0 winning on X_n is defined as follows. On $\mathit{Par}_0(V \backslash Z_n)$, she uses her positional strategy winning the game $(G[V \backslash Z_n], \mathrm{Parity}_\infty)$. Since $\mathit{Par}_0(V \backslash Z_n)$ is a trap for player 1 in $G[V \backslash Z_n]$, if player 1 makes the pebble go out of $\mathit{Par}_0(V \backslash Z_n)$, it necessarily reaches Z_n, and player 0 can use her positional strategy to attract it to $[n+1, \infty[$. □

A.4 The $\mathit{Exp} \cap \mathit{Parity}_d$ game

Proposition 3.

Let $G = (V, E)$ be an arena, coloured from 0 to $d > 0$. Let D be the set of vertices coloured by d. Player 0 wins the game $(G, \mathrm{Exp} \cap \mathrm{Parity}_d)$ on V if and

only if there exists a subarena $G[W]$, coloured from 0 to $d-1$ such that one of the following conditions holds:

- **Case d even:** *Player 0 wins the games* $(G[W], \mathrm{Exp} \cap \mathrm{Parity}_{d-1})$ *and* (G, Exp) *everywhere and she wins the game* $(G, Attraction(D))$ *on* $V \backslash W$.
- **Case d odd:** *Player 0 wins the game* $(G, Trap(W))$ *with a positional strategy* $\sigma_{Trap(W)}$ *and wins the game* $(G[W, \sigma_{Trap(W)}], \mathrm{Exp} \cap \mathrm{Parity}_{d-1})$.

The conditions of Proposition 3 are illustrated by figure 9.

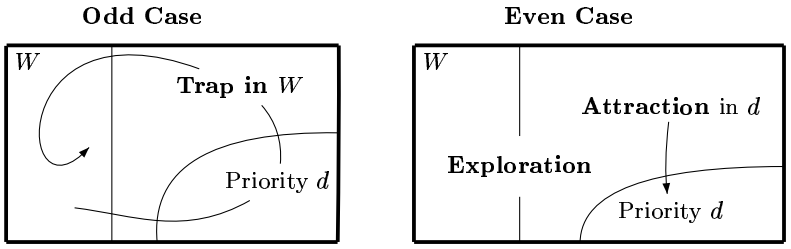

Fig. 9. Conditions of Proposition 3.

Remark 2. Note that winning the game $(G[W, \sigma_{Trap(W)}], Exp \cap Parity_{d-1})$ means that player 0 has a strategy σ_W winning the game $(G[W], Exp \cap Parity_{d-1})$ which advises player 0 to play moves **consistent with the positional strategy** $\sigma_{\mathbf{Trap(W)}}$.

Proof of the converse implication in the case where d is even. By hypothesis and by Corollary 1, player 0 has a positional strategy $\sigma_{Exp} \subseteq E$ winning (G, Exp). She has also a positional strategy $\sigma_{Att} \subseteq E$ winning $(G, Attraction(D))$ on $V \backslash W$ and a strategy σ_{Sub} winning the game $(G[W], Exp \cap Parity_{d-1})$.

σ_G is constructed in the following way. At a a given moment player 0 is in one of the three playing modes: *Attraction*, *Sub* or *Exploration*. It can change the mode when the pebble moves to a new vertex. Player 0 begins to play in *Exploration* mode. Here follows the precise description of the strategy σ_G, summarized by figure 10.

- The playing mode *Exploration* can occur wherever the pebble is. Player 0 plays according to her positional strategy σ_{Exp}. When a new vertex v is visited for the first time the mode is changed. If $v \in W$, it is changed to *Sub* mode. If $v \notin W$, it is changed to *Attraction* mode.
- The playing mode *Attraction* can occur only if the pebble is not in W. Player 0 plays according to her positional strategy σ_{Att}. When a vertex of colour d is visited, the playing mode is set to *Exploration*.

- The playing mode *Sub* can occur only if the pebble is in W. Player 0 plays using her strategy σ_{Sub} in the following way. Let p be the sequence of vertices visited up to now and p' the longest suffix of p consisting of vertices of W. Then 0 takes a move according to $\sigma_{Sub}(p')$. If the pebble goes out of W, the playing mode is set to *Exploration*.

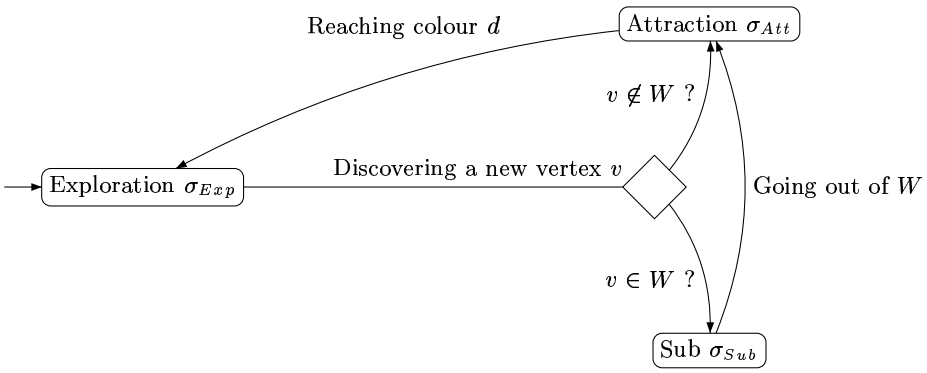

Fig. 10. Rules of transition between playing modes.

Notice that, by definition of σ_{Att} and σ_{Exp}, it is not possible that an infinite play consistent with σ_G stays forever in the playing modes *Attraction* or *Exploration*. Hence, such a play can be of two different types. Either the pebble stays ultimately in the playing mode Sub or it goes infinitely often in the modes *Exploration* and *Attraction*. In the first case, it stays ultimately in W and the play is ultimately consistent with σ_{Sub}. In the second case, the pebble visits infinitely often the even colour d and discovers infinitely often a new vertex. In both cases, this play is won by player 0 for the $Exp \cap Parity_d$ condition. □

Proof of the converse implication in the case where d is odd. By hypothesis, player 0 has a positional strategy $\sigma_{Trap(W)}$ winning the game $(G, Trap(W))$ and a strategy σ_{Sub} winning the game
$(G[W, \sigma_{Trap(W)}], Exp \cap Parity_{d-1})$. σ_G is constructed in the following way:
- If the pebble is not in W, player 0 plays according to her positional strategy $\sigma_{Trap(W)}$.
- If the pebble is in W, player 0 uses her strategy σ_{Sub} in the following way: Let p be the sequence of vertices visited up to now and let p' be the longest suffix of p consisting of vertices of W. Player 0 takes a move according to $\sigma_{Sub}(p')$.

The strategy σ_G is winning the game $(G, Exp \cap Parity_d)$. Indeed, since σ_{Sub} is a strategy in the arena $G[W, \sigma_{Trap(W)}]$, all the moves consistent with σ_G are consistent with $\sigma_{Trap(W)}$. Hence, the play is ultimately trapped in W and is ultimately consistent with σ_W, thus won by player 0.

Proof of the direct implication. Proving the direct implication in the case where d is even is simple. Consider $W = V \setminus Att_0(D)$. Since $V \setminus W$ is a trap, and from

Lemma 1, player 0 wins $(G[V \backslash W], \mathit{Exp} \cap \mathit{Parity}_d)$. The other claims are trivially true.

The proof is harder in the case where d is odd. We establish first that the family of subarenas of G where Proposition 3 holds is closed by arbitrary union, then we prove that the maximal such arena is necessarily G itself.

We need the following notion of a good tuple:

Definition 1. *A tuple (X, W, σ) is good if*

1. *$X \subset V$ is a trap for 1 and $W \subset X$.*
2. *$G[W]$ is a subarena of $G[X]$ coloured from 0 to $d-1$.*
3. *σ is a positional strategy for 0 winning for the game $(G[X], \mathrm{Trap}(W))$.*
4. *Player 0 wins the game $(G[W, \sigma], \mathrm{Exp} \cap \mathrm{Parity}_{d-1})$.*

Let $(X_n, W_n, \sigma_n)_{n \in I}$ be the family of good tuples in G and $\mathbb{X} = (X_n)_{n \in I}$. Then showing the direct implication of Proposition 3 in the case where d is odd is equivalent to showing that $V \in \mathbb{X}$. This is a direct consequence of the following lemmas 6 and 7, which concludes the proof. □

Lemma 6. *Let $(X_n, W_n, \sigma_n)_{n \in I}$ be the family of good tuples in G. Then $\bigcup_{n \in I} X_n$ is also a good tuple.*

Proof. Let $X = \bigcup_{n \in I} X_n$, $W = \bigcup_{n \in I} W_n$ and let $\preceq$ be a well-founded order on I. For any $v \in X$, the *index* of v is $\mathit{index}(v) = \mathit{min}\{n \in I : v \in X_n\}$. Define $\sigma = \{(v, w) : (v, w) \in \sigma_{\mathit{index}(v)}\}$. We shall prove that (X, W, σ) is a good tuple.

Condition 1 is verified because a union of traps is a trap. Condition 2 is trivially verified.

Let us prove that condition 3 is also verified. It is easy to check that σ is a positional strategy for 0 in $G[X]$. Let $p = v_0 v_1 \ldots$ be an infinite play consistent with σ. Since the $X'_n s$ are traps for 1, the sequence $\mathit{index}(v_0)\mathit{index}(v_1)\ldots$ is $\preceq$-decreasing. As $\preceq$ is a well-founded order, this sequence is stationary equal to a level $n(p) \in I$. Hence, p is ultimately played in the arena $X_{n(p)}$, consistently with $\sigma_{n(p)}$, and a suffix of p is ultimately trapped in $W_{n(p)} \subset W$. Hence σ wins $(G[X], \mathit{Trap}(W))$

To achieve the proof that (X, W, σ) is a good tuple, we show that condition 4 is verified, by constructing a strategy τ winning $(G[W, \sigma], \mathit{Exp} \cap \mathit{Parity}_{d-1})$. For each $n \in I$, let τ_n be a strategy for player 0 winning $(G[W_n, \sigma_n], \mathit{Exp} \cap \mathit{Parity}_{d-1})$. Let p be a finite play in $G[W, \sigma]$ ended in a vertex v of player 0. Let p' be its longest suffix which is a play consistent with $\tau_{\mathit{index}(v)}$. Note that p' exists since (v) is such a suffix. Define $\tau(p) = \tau_{\mathit{index}(v)}(p')$. Let q be an infinite play in $G[W, \sigma]$ consistent with the strategy τ. Since τ is consistent with the positional strategy σ, we know (proof of condition 3 above) that q is ultimately trapped in $W_{n(q)}$. Hence q is ultimately consistent with $\tau_{n(q)}$, and q is won by player 0 for the game $(G[W, \sigma], \mathit{Exp} \cap \mathit{Parity}_{d-1})$. □

Lemma 7. *If player 0 wins $(G, \mathrm{Exp} \cap \mathrm{Parity}_d)$ then $\bigcup_{n \in I} X_n = V$.*

Proof. The key of this proof (and of the one of Proposition 2) is Lemma 8. Let $X = \bigcup_{n \in I} X_n$ and suppose that $V \backslash X \neq \emptyset$. We are going to show that it contradicts the maximality of X in $\mathbb{X}$. To achieve this, we shall build a good tuple of the form $(X \cup Y, ., .)$, where $Y \not\subset X$.

First, let's prove that $V \backslash X$ is a trap for 0. Let σ_{Att} be a positional strategy for 0 winning $Attraction(X)$ on $Att_0(X)$. Let σ' be the strategy consisting in playing σ_{Att} on $Att_0(X) \backslash X$ and playing σ on X. Then, since X is a trap for 1, it is straightforward to check that $(Att_0(X), W, \sigma')$ is a good tuple. Hence, by maximality of X in $\mathbb{X}$, $X = Att_0(X)$, which implies that $V \backslash X$ is a trap for 0.

Now, we shall use lemma 8 to contradict the maximality of X in $\mathbb{X}$. Since $V \backslash X$ is a trap for 0 and from Lemma 1, 0 wins $(G[V \backslash X], Exp \cap Parity_d)$. Applying Lemma 8 to $G[V \backslash X]$, we obtain a set of vertices $\emptyset \neq Y \subset V \backslash X$, coloured from 0 to $d-1$, which is a trap for 1 in the arena $G[V \backslash X]$ and such that 0 wins $(G[Y], Exp \cap Parity_d)$. If, during a play in the arena G, the pebble is in Y and 1 moves it out of Y, then, as Y is a trap for 1 in $V \backslash X$, the pebble necessarily goes in X. Hence, $Trap \cup (Y \times (Y \cup X)) \cap E$ is a positional strategy for 0 winning for the game $(G[X \cup Y], Trap(W \cup Y))$. With this remark, since X is a trap for 1, it is straightforward to check that $(X \cup Y, W \cup Y, Trap \cup (Y \times X) \cap E)$ is a good tuple. It contradicts the maximality of X in $\mathbb{X}$. □

Lemma 8. *Let G be a parity arena labeled by $\phi : V \to [0, d]$, such that 0 wins $(G, \mathrm{Exp} \cap \mathrm{Parity}_d)$. Then there exists a non-empty trap Y for 1 in G, coloured from 0 to $d-1$, such that 0 wins the game $(G[Y], \mathrm{Exp} \cap \mathrm{Parity}_{d-1})$.*

Proof (of the Lemma 8.). $X = V \backslash Att_1(D)$ is a trap for 1 coloured from 0 to $d-1$. Let $Y \subset X$ be the winning set for 0 for the game $(G[X], Exp \cap Parity_{d-1})$. According to lemma 2, Y is a trap for 1 in $G[X]$ hence it is a trap for 1 in G. According to lemma 1, 0 wins the game $(G, Exp \cap Parity_{d-1})$ on Y. Now, to achieve the proof, we show that $Y \neq \emptyset$. Let us suppose that $Y = \emptyset$ and find a contradiction. $Y = \emptyset$ means that 1 wins the game $(G[X], Exp \cap Parity_d)$. In this case, applying lemma 3, we deduce that 1 wins the game $(G, Exp \cap Parity_d \cap (V^\omega \backslash Buchi(D))) = (G, Exp \cap Parity_d)$. Since, by hypothesis, 0 wins $(G, Exp \cap Parity_d)$, it is a contradiction □

Proposition 2. *Player 1 wins positionally the game $\mathrm{Exp} \cap \mathrm{Parity}_d$ on his winning set.*

Proof. Without loss of generality, we can assume that player 1 wins everywhere. The proof is by induction on d. Again, $D = \phi^{-1}(d)$ denotes the set of vertices of priority d.

If $d = 0$, it is impossible for player 1 to win any play and his winning set is empty.

If d is odd and $d \neq 0$, consider the set $W = Att_1(D)$.

From Lemma 1, player 1 wins $(G[V \backslash W], Exp \cap Parity_d)$ and we can apply the inductive hypothesis to the arena $G[V \backslash W]$, which is coloured from 0 to $d-1$. Then Lemma 3 shows that player 1 wins the game $Exp \cap Parity_d \cup (V^\omega \backslash Buchi(D)) = Exp \cap Parity_d$.

The case where d is even is less trivial. From Lemma 5, we deduce immediately that there exists the greatest subarena of G where player 1 wins positionally. It remains to prove that this subarena coincides with the whole subarena.

Let $\mathbb{X} = (X_n)_{n \in I}$ be the family of subsets $X_n \subset V$ such that X_n is a trap for 0 and 1 wins positionally the game $Exp \cap Parity_d$ on X_n. According to lemma 5, this family is stable by union. Let $X = \bigcup_{n \in I} X_n$ and σ_X be an associated positional strategy winning the game $Exp \cap Parity_d$ on X.

To achieve the proof, we shall prove that $X = V$. Let $Y = V \backslash X$. Suppose that $Y \neq \emptyset$. To obtain a contradiction, we are going to show that Y is a trap for 1 and that 0 wins $(G[Y], Exp \cap Parity_d)$. According to lemma 1, it implies that 0 wins $(G, Exp \cap Parity_d)$ on Y which contradicts the hypothesis stating that 1 wins $(G, Exp \cap Parity_d)$ on V.

Y is a trap for player 1: Since X is a trap for player 0, player 1 wins positionally $(G, Exp \cap Parity_d)$ on $Att_1(X)$. For that, while the pebble stays in $Att_1(X) \backslash X$, player 1 can attract it positionally in X. When the pebble reaches X, he can then play its positional strategy winning on X. Hence, $Att_1(X) \in \mathbb{X}$ and by maximality of $\mathbb{X}$, $X = Att_1(X)$. It proves that $Y = V \backslash X$ is a trap for player 1.

To show that 0 wins $(G[Y], Exp \cap Parity_d)$ on Y, we prove that the sufficient conditions of Proposition 3 are verified by the arena $G[Y]$, relatively to the set $W = Y \backslash Att_0(G[Y], D)$.

0 wins the game $(\mathbf{G[Y], Exp})$: Let Y_E be the winning set for 1 for the game $(G[Y], Exp)$. According to proposition 1, 1 holds a positional strategy σ_E winning for $(G[Y], Exp)$ on Y_E.

Since X is a trap for player 0, $\sigma_X \cup \sigma_E \cup Y_E \times X$ is a positional strategy for player 1 in G. Moreover, it wins the game $(G, Exp \cap Parity_d)$ on $X \cup Y_E$. In fact, a play p consistent with this strategy can be of two types. Either p doesn't reach X and is consistent with σ_E. In this case, only a finite number of vertices are visited, and the play is lost by player 0 for the $Exp \cap Parity_d$ game. Or the play p reaches X and from this moment stays in X and is consistent with σ_X. In this last case, some suffix of p is lost by player 0for the game $(G[X], Exp \cap Parity_d)$. In both cases, the play is won by player 1. Hence, we proved that 1 wins $(G, Exp \cap Parity_d)$ on $X \cup Y_E$.

By maximality of X in $\mathbb{X}$, $Y_E = \emptyset$. By definition of Y_E, player 0 wins $(G[y], Exp)$.

0 wins the game $(\mathbf{G[W]}, \boldsymbol{Exp \cap Parity_{d-1}})$: Let W_1 be the winning set for 1 for the game $(G[W], Exp \cap Parity_{d-1})$. Since W is coloured from 0 to $d-1$, by inductive hypothesis, 1 holds a positional strategy τ_{W_1} winning for $(G[W], Exp \cap Parity_{d-1})$ on W_1. It is easy to see that $\sigma_W \cup W_1 \times X \cup \sigma_X$ is a positional strategy for 1 winning the game $(G, Exp \cap Parity_d)$ on $X \cup W_1$. By maximality of X in $\mathbb{X}$, $W_1 = \emptyset$.

Hence, the sufficient conditions of Proposition 3 are verified, which proves that 0 wins $(G[Y], Exp \cap Parity_d)$. It is a contradiction with the hypothesis that states that player 1 wins $(G, Exp \cap Parity_d)$. □

B Computing the winning sets and strategies on pushdown arenas.

B.1 Preliminaries.

We recall briefly the definitions of an alternating Büchi or co-Büchi automata, a successful run and of the language recognized by such automata.

Boolean formulas. Given a set S, the set $B^+(S)$ denotes the set of all positive formulas over the set S with $\top\top$ and $\bot\bot$ meaning respectively true and false. We say that a subset $S' \subseteq S$ satisfies a formula $\phi \in B^+(S)$ (denoted $S' \models \phi$) if by assigning $\top\top$ to all members of S' and false to all members of $S \backslash S'$ the formula ϕ evaluates to $\top\top$. Given a formula ϕ, its *dual* $\bar{\phi}$ is obtained by replacing $\wedge$ by $\vee$, $\top\top$ by $\bot\bot$ and vice-versa.

Alternating automata. An alternating Büchi automaton on trees $\Gamma^* \to \Sigma$ is a tuple $\mathcal{A} =< S, s_0, T, F >$, where

- S is a finite set called the set of states,
- $s_0 \in S$ is the initial state,
- $T : S \times \Sigma \to B^+((\Gamma \times S))$ is the set of transitions,
- $F \subset S$ is a set of final states.

A *run* of this automaton over a tree $t : \Gamma^* \to \Sigma$ is a tree $r \subset (\Gamma \times S)^*$, such that

- $(r \cap (\Gamma \times S)) \models T(s_0)$,
- for any word $u \in \mathcal{D}^*$, for any couple $(\alpha, s) \in \Gamma \times Q$,

$$\{(\beta, t) \in \Gamma \times S : u(\alpha, s)(\beta, t) \in r\} \models T(s)$$

We can now define acceptance by alternating automata. A run $r \subset (\Gamma \times S)$ on t is accepting if any infinite branch of r visits the set $\Gamma \times F$ infinitely often.

If there exists an accepting run of $\mathcal{A}$ on t, we say that t is accepted or recognized by the Büchi automaton $\mathcal{A}$. $\mathcal{L}(\mathcal{A})$ is the set of trees accepted by $\mathcal{A}$.

A co-Büchi alternating automaton is defined exactly the same way except that a run is accepting if any branch of it visits the set $\Gamma \times F$ finitely often.

A simple alternating automaton is also defined the same way except that any run is accepting. In fact, a simple alternating automaton is a special case of Büchi automaton, when $F = S$.

Given a simple (resp. Büchi, co-Büchi) alternating automaton $\mathcal{A} =< S, s_0, T, F >$, its dual $\bar{\mathcal{A}}$ is the simple (resp. co-Büchi, Büchi) automaton $\mathcal{A} =< S, s_0, \bar{T}, F >$, where $\bar{T}(s) = (\bar{T(s)})$. The automaton $\mathcal{A}$ and $\bar{\mathcal{A}}$ accept complementary languages [14].

We shall need the following elementary result.

Proposition 4. *Let $\mathcal{A} =< S, s_0, T, F >$ be an alternating co-Büchi automaton. One can compute an alternating Büchi automaton recognizing the same language and whose state space is twice bigger than the state space of $\mathcal{A}$.*

Proof. Koenig Lemma implies that for every accepting computation $r \subset (\Gamma \times S)^*$, of a co-Büchi alternating automaton, there exists some integer $n \in \mathbb{N}$ such that for every $m \geq n$,

$$(\gamma_0, s_0) \dots (\gamma_m, s_m) \in r \implies s_m \notin F$$

Hence, during a successful computation, after some time, every copy of the automaton never meets again a final state. Roughly speaking, the equivalent Büchi automaton shall guess this moment.

To get the Büchi alternating automaton $\mathcal{B}$ equivalent to $\mathcal{A}$, it suffices to duplicate the state space of $\mathcal{A}$. The state space of $\mathcal{B}$ is $S \cup \{new\} \times (Q \backslash F)$. The final states of $\mathcal{B}$ are $\{new\} \times (Q \backslash F)$. The transition function of $\mathcal{B}$ is T' defines as follows. Given a boolean positive formula $f \in B^+()$, (new, f) denotes the formula of $\mathcal{B}^+(\{new\} \times S)$ obtained from f by replacing every occurrence of $s \in S$ by (new, s) if $s \notin F$ and by $\perp\!\!\!\perp$ if $s \in F$. For $s \in S$ and $\alpha \in \Sigma$,

$$T'(s) = T(s) \vee (new, T(s)) \text{ and } T'((new, s)) = (new, T(s)).$$

Non-determinization of alternating automata.

B.2 Computation of the winning sets and strategies.

Theorem 1. *Let i be a player and* $\mathrm{Vic} \in \{\mathrm{Parity}_d, \mathrm{Exp} \cup \mathrm{Parity}_d, \mathrm{Exp} \cap \mathrm{Parity}_d\}$. *The language of strategy trees which correspond to winning positional strategies for i is regular. One can effectively construct an alternating co-Büchi automaton $\mathcal{A}_{\mathrm{Vic},i}$ with $\mathcal{O}(d \mid Q \mid^2 + \mid \Gamma \mid)$ states which recognizes it.*

Proof. The construction of $\mathcal{A}_{\mathrm{Vic},i}$ uses techniques close to the one of [20,16]. Unfortunately, we couldn't manage to make use of the results of those papers about two-way tree automata, because we don't know how to use a two-way automata to detect a cycle in a strategy tree.

Our aim is to construct an alternating co-Büchi automaton accepting a tree $t : \Gamma^* \to 2^\Delta$ iff there exists a winning positional strategy σ such that $t = T_\sigma$. In fact we shall rather construct a Büchi alternating automaton recognizing the complement of the set $\{T_\sigma | \sigma \text{ is a winning positional strategy}\}$.

B.3 Testing whether a tree is a strategy tree.

First of all it is easy to implement an alternating simple automaton verifying if the tree t is or is not a strategy tree. We give the details of this construction, which is based on the following lemma.

Lemma 9. *Let $t : \Gamma^* \to 2^\Delta$ be a tree. Then t is a strategy tree for player i if and only if $\forall \gamma \in \Gamma^*, \forall \alpha, \beta \in \Gamma, \forall q, r \in Q$*

1. *$t(\epsilon)$ only contains $\perp$-transitions and $t(\gamma\alpha)$ only contains α-transitions.*

2. *If $t(\gamma\alpha)$ contains a pop-transition to a state r, then $t(\gamma)$ contains a r-transition.*
3. *If $t(\gamma)$ contains a β-push-transition to a state r then $t(\gamma\beta)$ contains a r-transition.*
4. *If $t(\gamma)$ contains a (q,α)-transition and $q \in Q_{1-i}$ then $t(\gamma)$ contains every (q,α) transitions.*

Moreover, the language of strategy trees for player i is recognized by a Büchi alternating automaton with $2 + |\Gamma| + 2|Q|$ states.

Proof. Condition (1) is equivalent to the fact that there exists a subgraph G' of G_P such that t is the coding-tree of this subgraph. Conditions (2) and (3) are equivalent to the fact that there are no dead-end in G'. Conditions (4) is equivalent to the fact that the moves of player $1-i$ are not restricted.

For each condition $\mathcal{C} \in \{1,2,3,4\}$, we construct an alternating automaton $\mathcal{A}_\mathcal{C}$ which accepts a tree $t : \Gamma^* \to 2^\Delta$ if and only if it verifies the condition $\mathcal{C}$.

Construction of $\mathcal{A}_1$. Instead of constructing the Büchi automaton $\mathcal{A}_1$, we construct its dual $\bar{\mathcal{A}}_1$, which is a co-Büchi automaton. $\bar{\mathcal{A}}_1$ looks for a vertex γ whose label contains a transition not consistent with the top letter of γ. It only needs to keep track of the top of the current stack, hence its state space is $\Gamma \cup \{\bot\}$. The initial state is $\bot$. All the states are final, hence the accepting runs are exactly the finite runs. The transition relation T is defined as follows :

$\forall A \subset \Delta, \forall \alpha \in \Gamma \cup \{\bot\}$,

$$T(\alpha, A) = \begin{cases} \top\top \text{ if } A \not\subseteq \Delta_\alpha \\ \bigvee\limits_{\beta \in \Gamma} (\beta, \beta) \text{ otherwise.} \end{cases}$$

Construction of $\mathcal{A}_2$ and $\mathcal{A}_3$. We only describe the construction of $\mathcal{A}_2$, since $\mathcal{A}_3$ is similar to $\mathcal{A}_2$. Once again, we construct $\bar{\mathcal{A}}_2$ instead of $\mathcal{A}_2$. It looks for a vertex γ, a state r and a letter α, such that a transition $(.,.,-1,r)$ is in $t(\gamma\alpha)$ but no r-transition is in $t(\gamma)$.

Its state space is $\{search\} \cup Q$. *search* is the initial state and once again all the states are final. In a first time, $\bar{\mathcal{A}}_2$ moves non-deterministically to the vertex γ in the state *search*, which is the initial state. Then it guesses some $r \in Q$ and $\alpha \in \Gamma$ such that there is no r-transition in the label of γ and moves to the vertex $\gamma\alpha$ in the state r. There he checks that some transition $(.,.,-1,r)$ is in the label of $\gamma\alpha$. The transition relation T is defined as follows:

$\forall A \subset \Delta$,

$$T(search, A) = \bigvee_{\alpha \in \Gamma} (\alpha, search) \vee \bigvee_{\substack{\{r \in Q : F \cap \Delta_{r,\alpha} = \emptyset\} \\ \alpha \in \Gamma}} (\alpha, r)$$

$$T(r, A) = \begin{cases} \top\top \text{ if a transition } (.,.,-1,r) \text{ is in } A \\ \bot\bot \text{ otherwise.} \end{cases}$$

Construction of $\mathcal{A}_4$. The construction of the dual $\bar{\mathcal{A}}_4$ of $\mathcal{A}_4$ is trivial. We explain it in the case where player i is player 0. This automaton moves nondeterministically in the tree, looking for a vertex γ, such that γ contains a (q,α)-transition but not every (q,α) transition, for some state $q \in Q_1$ and some letter $\alpha \in \Gamma$. It has only one state. □

B.4 Testing Whether a strategy tree is winning.

Now that we have constructed an automaton which checks whether a tree $\Gamma^* \to 2^\Delta$ is a strategy tree, we must construct an automaton checking if the strategy σ associated to such a strategy tree is winning or not. For each winning conditions we study, it can be expressed by some simple conditions about the cycles and the exploration paths of the graph $(Dom(\sigma), \sigma)$ induced by σ. Those criteria are summarized in table 1. An exploration path is a play winning for the game *Exp*, i.e. a path visiting an infinite number of different vertices.

Game	**i**	**Condition on cycles**	**Condition on exploration paths**
Parity	0	Even	Even
	1	Odd	Odd
$Exp \cup Parity_d$	0	Even	No condition
	1	Odd	No exploration path
$Exp \cap Parity_d$	0	No cycle	Even
	1	No condition	Odd

Table 1. Characterization of winning positional strategies.

To prove Theorem 1, we shall first construct some alternating automata for checking the various conditions on cycles and exploration paths listed in table 1. Then, by taking boolean combinations of those alternating automata (which, according to Proposition 4, induces only a linear blowup of the state space), we obtain some automata checking the different conditions of table 1, which proves Theorem 1.

The crucial observation is that any cycle and any exploration path can be factorized with a simple kind of finite path called a *jump*.

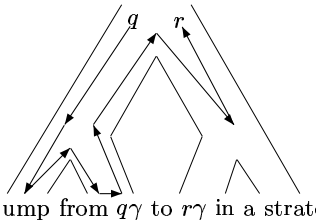

Fig. 11. A jump from $q\gamma$ to $r\gamma$ in a strategy tree.

Factorizing cycles and exploration paths with jumps.

Definition 2. *A jump of priority c from a configuration $q\gamma$ to a configuration $r\gamma$ is a path in σ from $q\gamma$ to $r\gamma$, of length $\neq 0$, such that:*

- *every configuration on the path is of the form $s\gamma\gamma'$ where $s \in Q$ and $\gamma' \in \Gamma^*$.*
- *The maximum priority met along the path is c.*

This definition is illustrated by Figure 9.

Clearly, a cycle of colour c is exactly a jump of colour c from a vertex to itself. The decomposition of exploration paths with jumps is described by the following Lemma and illustrated by fig. 12.

Lemma 10. *Let G be a subgraph of $G_{\mathcal{P}}$. There exists an exploration path of priority c in G if and only if*

1. *Either there exists an infinite sequence $j_0, j_1, \ldots$ of finite nonempty paths such that,*
   (a) *$j_0 j_1 j_2 \cdots$ is a path in G, and for each $i \geq 0$,*
   (b) *j_i is either a jump or a single vertex and $\phi(j_i) \leq c$,*
   (c) *the transition from the last vertex of j_i to the first vertex of j_{i+1} is a push transition,*
   (d) *$\overline{\lim}_{i\in\mathbb{N}} \phi(j_i) = c$.*
2. *or there exists an infinite sequence $j_0, l_0, j_1, l_1, \ldots$ of finite nonempty paths such that,*
   (a) *$j_0 j_1 j_2 \cdots$ and $l_n l_{n-1} \cdots l_0$ are paths in G,*
   (b) *each path j_i and l_i is either a jump or a single vertex of priority less than c,*
   (c) *the transition from the last vertex of j_i to the first vertex of j_{i+1} is a push transition,*
   (d) *the transition from the last vertex of l_{i+1} to the first vertex of l_i is a pop transition,*
   (e) *there exists a jump of priority less than c from the last vertex of j_i to the first vertex of l_i,*
   (f) *the first vertex of j_0 is identical to the last vertex of l_0 and there is a jump of priority c from this vertex to itself.*

Proof. Let $p = (q_0\gamma_0)(q_1\gamma_1)\ldots$ be an exploration path of priority c in G. Then, since $\overline{\lim}\, \phi(q_i) = c$, there exists some index $j \geq 0$ such that the states $q_i, i \geq j$ are of priority equal or less than c. Without restriction, we suppose that $j = 0$, since any suffix of p is also an exploration path of priority c.

There are two cases, depending on whether $\underline{\lim}_{i\in\mathbb{N}} |\gamma_i| = +\infty$.

In the case where $\underline{\lim}_{\mathbf{i}\in\mathbb{N}} |\gamma_\mathbf{i}| = +\infty$, we show that condition 1 of Lemma 10 holds. Let us define inductively the sequence $j_0, j_1, \ldots$, which is simply a factorization of p, i.e. $j_0 \cdots j_i$ is a prefix of p (hence $\overline{\lim}_{i\in\mathbb{N}} \phi(j_i) = \phi(p) = c$). It is defined in such a way that the conditions 1a, 1b, 1c and 1d of Lemma 10 hold and that the stack associated with the last vertex of a j_i is never reached again after $j_0 \cdots j_i$, i.e. $\gamma_i \notin \{\gamma_k : k > |j_0| + \ldots + |j_i|\}$.

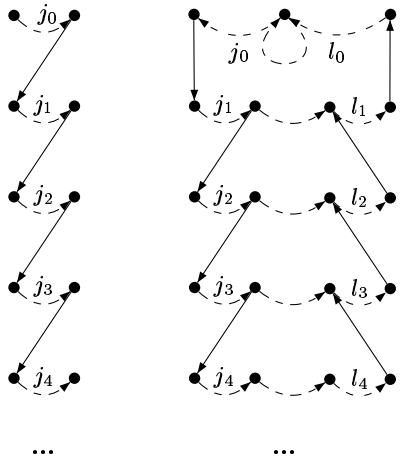

Fig. 12. Two types of exploration paths of priority c. The dotted lines are jumps. All the jumps priorities are less than c. On the left hand side, infinitely many $j_i's$ priority are exactly c. On the right hand side, the upper jump, which is a loop, has priority c.

Suppose that $j_0, \ldots, j_i$ is already constructed and let n be the index of the last vertex of j_i in p. Since $\underline{\lim}_{i\in\mathbb{N}} |\gamma_i| = +\infty$ and $Q \times \{\gamma_{n+1}\}$ is finite, p reaches only finitely often a vertex whose stack is γ_{n+1}. Let $m = \max\{k > n : \gamma_k = \gamma_{n+1}\}$ be the last moment where it occurs and let $j_{i+1} = (q_{n+1}\gamma_{n+1}) \cdots (q_m\gamma_m)$.

By definition of m, the inductive hypothesis holds for the sequence $j_0, j_1, \ldots, j_i$. Since $j_0 j_1 \cdots j_i j_{i+1}$ is a prefix of p, condition 1a holds, and since $\gamma_{n+1} = \gamma_m$, condition 1b holds. Let us check condition 1c. If $(q_n\gamma_n, q_{n+1}\gamma_{n+1})$ were a pop-transition then γ_{n+1} would be a strict prefix of γ_n. Since the set $Q \times Prefix(\gamma_n)$ is finite and p is an exploration path, the stack γ_n should occur one more time in p, which contradicts the inductive hypothesis.

In the case where $\underline{\lim}_{\mathbf{i}\in\mathbb{N}} |\gamma_\mathbf{i}| \in \mathbb{N}$, we show that condition 2 of Lemma 10 holds. In this case, some vertex $q_0\gamma_0$ must occur infinitely often in p. By choosing γ_0 of minimal length and taking some suffix of p, we can restrict to the case where $q_0\gamma_0$ is the first vertex of p and γ_0 is a prefix of every stack $\gamma_i, i \in \mathbb{N}$.

We define inductively the sequence (j_i) and (l_i), as well as some sequence $(q_i, r_i, \gamma_i)_{i\in\mathbb{N}}$ such that

i there is a push transition from the last vertex of j_i to $q_i\gamma_i$,
ii a pop transition from $r_i\gamma_i$ to the first vertex of l_i,
iii and such that there are jumps of arbitrary high depth and of priority $\leq c$ from $q_i\gamma_i$ to $r_i\gamma_i$.

We begin the induction with $j_0 = q_0\gamma_0 = r_0\gamma_0 = l_0$. Now suppose that the sequence has been defined until rank i and that conditions 2 of lemma 10 as well

as conditions (i), (ii) and (iii) hold. Let J_i be the set of jumps of priority $\leq c$ from $q_i\gamma_i$ to $r_i\gamma_i$. Let $j \in J_i$ be a jump of depth $m \geq 2$. Then, by definition of a jump, there exists a factorization $j = uvw$, such that u and w are jumps on the stack γ_i and v is a jump of depth $(m-1)$ from some vertex $q(j)\gamma_i\alpha(j)$ to some vertex $r(j)\gamma_i\alpha(j)$ with $q(j), r(j) \in Q$ and $\alpha(j) \in \Gamma$.

According to condition (iii), J_i is infinite. Since $Q^2 \times \Gamma$ is finite, there exists some triple $(q_{i+1}, r_{i+1}, \alpha_i)$ such that $(q_{i+1}, r_{i+1}, \alpha_i) = (q(j), r(j), \alpha(j))$ for infinitely many $j \in J_i$ of arbitrarily high depth. Let $\gamma_{i+1} = \gamma_i\alpha_i$ and $j_{i+1}vl_{i+1}$ be a factorization of a jump $j \in J_i$ such that v is a jump from $q_{i+1}\gamma_{i+1}$ to $r_{i+1}\gamma_{i+1}$ of priority $\leq c$. By definition of q_{i+1} and r_{i+1}, condition 2e holds. By inductive hypothesis (i) and (ii) $q_i\gamma_i$, which is the first vertex of j_{i+1}, is a successor of the last vertex of j_i, whereas $r_i\gamma_i$, the last vertex of l_{i+1}, is a predecessor of the first vertex of l_i, which proves that condition 2a, 2b, 2c, 2d , (i) and (ii) hold. Condition (iii) hold by definition of the triple $(q_{i+1}, r_{i+1}, \alpha_i)$. □

Testing the existence of a jump in a strategy tree. To prove Theorem 1, we shall first construct an alternating automaton $\mathcal{A}_{jump}$ that can check the existence of a jump. Once done, it is easy to modify it in order to obtain an automaton checking the existence of a jump or of an exploration path of a given priority.

We use the following notations. For $\gamma \in \Gamma^*$ and $t : \Gamma^* \to 2^\Delta$, $t[\gamma]$ denotes the tree $\gamma' \to t(\gamma\gamma')$. Given an automaton $\mathcal{A}$ and a state q of this automaton, $\mathcal{A}[q]$ denotes the automaton obtained by setting the initial state of $\mathcal{A}$ to be q.

Lemma 11. *There exists an alternating Büchi automaton $\mathcal{A}_{jump}$ with state space*
$([0,d] \cup \{pop\}) \times Q^2$ *such that, given $(c,q,r) \in [0,d] \times Q^2$, a coding tree $t : \Gamma^* \to 2^\Delta$ of a subgraph G of $G_\mathcal{P}$ and some vertex γ of t, there exists a jump of priority c from $q\gamma$ to $r\gamma$ in G if and only if $\mathcal{A}_{jump}[(c,q,r)]$ accepts $t[\gamma]$.*

Proof. Suppose that there exists a jump of colour c from $q\gamma$ to $r\gamma$ and let

$$j = (q_0\gamma_0, q_1\gamma_1, \ldots, q_n\gamma_n), \text{ where } q_0\gamma_0 = q\gamma \text{ and } q_n\gamma_n = r\gamma$$

be such a jump. Consider the set $\{i \in [0,n] : \gamma_i = \gamma\}$ of indices where the jump j comes back on the stack γ. We sort it:

$$\{i \in [0,n] : \gamma_i = \gamma\} = \{0 = i_0 < i_1 < \ldots < i_k = n\}.$$

Remark that when we selected a jump of priority c from $q\gamma$ to $r\gamma$, we could have chosen one such that there exists at most one couple $\{(l,l') : l < l'\}$ such that $q_{i_l} = q_{i_{l'}}$. (In fact, we can delete almost all the cycles in j, possibly keeping one such cycle if its priority is c) Suppose that we have selected such a jump, then $k \leq |Q| + 1$.

The jump j can be factorized as

$$j = (q_{i_0}\gamma) \cdot j_0 \cdot (q_{i_1}\gamma) \cdots j_k \cdot (q_{i_k}\gamma). \tag{1}$$

where the j_i's are non-empty paths.

We can remark a few things about this factorization. From the definition of a jump, it appears that, for each $1 \leq i \leq k$, the path j_i is either a single vertex or a jump from a configuration $r_i\gamma\alpha_i$ to another configuration $s_i\gamma\alpha_i$, where $r_i, s_i \in Q$ and $\alpha_i \in \Gamma$. On figure 9, those two cases correspond respectively to the right and left parts of the jump. Moreover, since j is a path in G, we deduce that for each $1 \leq l \leq k-1$, $(q_{i_l}\gamma, r_l\gamma\alpha_l)$ and $(s_l\gamma\alpha_l, q_{i_{l+1}}\gamma)$ are edges of G, corresponding respectively to an α_l-push transition and a pop-transition of $\mathcal{P}$. From the definition of the priority, we get

$$\phi(j) = \max\{\phi(q_{i_l}), c_l : l \in [0,k]\}, \text{ where } c_l = \phi(j_l).$$

The construction of $\mathcal{A}_{jump}$ is directly inspired by those remarks. When $\mathcal{A}_{jump}$ is in the state (c,q,r) on a stack γ, it guesses the sequence

$$(q_{i_l}, r_l, \alpha_l, s_l, q_{i_l+1}, c_l)_{0 \leq l \leq k-1} \in (Q^2 \times \Gamma \times Q^2 \times [0,d])^{\leq |Q|+1},$$

with $k \leq |Q| + 2$. Then $\mathcal{A}_{jump}$ has to verify that this sequence is really the sequence associated with a jump j of priority c from $q\gamma$ to $r\gamma$. For that, it shall check that, for $0 \leq l \leq k-1$:

1. $q_{i_0} = q$ and $q_{i_{k+1}} = r$,
2. $t(\gamma)$ contains a push transition $(q_{i_l}, ., \alpha_l, r_l)$.
3. $(r_l = s_l$ and $\phi(r_l) = c_l)$ or there exists a jump of color c_l from $r_l\gamma\alpha_l$ to $s_l\gamma\alpha_l$,
4. $t(\gamma\alpha_l)$ contains a pop transition $(s_l, \alpha_l, -1, q_{i_l+1})$,
5. $\max\{\phi(q_l), c_l : 0 \leq l \leq k\} = c$.

The verification of conditions 1,2 and 5 can be made without moving from the vertex γ.

For checking condition 4, $\mathcal{A}_{jump}$ shall send a copy of itself in the direction α_l, to check that the transition $(s_l, \alpha_l, -1, q_{l+1})$ is in the label of the vertex $\gamma\alpha_l$. This copy needs to record the couple $(s_l, q_{l+1}) \in Q^2$ and its state is (pop, s, q_{l+1}).

Let consider condition 3. If $r_l = s_l$ and $\phi(r_l) = c_l$, condition 3 holds and $\mathcal{A}_{jump}$ does nothing. In the opposite case, it sends a copy of itself in direction α_l to check the existence of a jump of priority c_l from $r_l\gamma\alpha_l$ to $s_l\gamma\alpha_l$. This copy must record the tuple $(c_l, r_l, s_l) \in [0,d] \times Q^2$ and its state is (c_l, r_l, s_l).

Finally, the state space of $\mathcal{A}_{jump}$ is $\{\text{pop}\} \times Q^2 \cup [0,d] \times Q^2$, it has no final state, and its transition table is:

$\forall A \subseteq \Delta$ and $\forall \alpha \in \Gamma \cup \{\bot\}$,

$$T((pop, s, q), A) = \begin{cases} \top\!\!\top \text{ if some pop-transition } (s, ., -1, q) \text{ is in } A \\ \bot\!\!\bot \text{ otherwise.} \end{cases}$$

$$T((c, r, s), A) = \bigvee_{\substack{(q_l, r_l, \alpha_l, s_l, q_{l+1}, c_l)_{0 \leq l \leq k \leq |Q|} \\ \text{s.t. condition 1,2 and 5 are verified.}}} \bigwedge_{0 \leq l \leq k} \left((\alpha_l, (\text{pop}, s_l, q_{l+1})) \wedge \left((\alpha_l, (c_l, r_l, s_l)) \vee \top\!\!\top_{\substack{r_l = s_l \\ \phi(r_l) = c_l}} \right) \right)$$

$$\text{where} \top\!\!\top_{\mathcal{C}} \text{ means } \begin{cases} \top\!\!\top \text{ if } \mathcal{C} \text{ is a true condition} \\ \bot\!\!\bot \text{ otherwise.} \end{cases}$$

Since there are no final state, an accepting computation of $\mathcal{A}_{jump}$ is exactly a finite computation where all the copies reaches eventually the state $\top\!\!\top$. An induction on the height of this tree and on the height of the jumps proves Lemma 11.

Checking the existence of cycles or exploration paths in strategy trees. Now that $\mathcal{A}_{jump}$ has been constructed, we begin the constructions of the automata checking the conditions of Table 1.

Concerning the conditions on cycles, those constructions consist in trivial modifications of $\mathcal{A}_{jump}$, since a cycle is simply a jump from a vertex to itself (Lemma 10).

We also have to construct an automaton $\mathcal{A}_{path}$ checking the existence of an exploration path of priority c in a strategy tree t. Roughly speaking, $\mathcal{A}_{path}$ tries to guess a sequence $j_0, j_1, j_2 \ldots$ or $j_0, l_0, j_1, l_1 \ldots$ which meet the conditions of Lemma 10.

First case of lemma 10. Let $j_0 j_1 j_2 \ldots$ be as in Lemma 10, δ_i be the push transition between the last vertex $r_i \gamma_i$ of j_i and the first vertex $q_{i+1}\gamma_{i+1}$ of j_{i+1}, and $c_i = \phi(j_i)$. During a computation, one of the copy of $\mathcal{A}_{path}$, that we will call the main copy, tries to guess the sequences $\delta_0, \delta_1, \ldots$ and $c_1, c_2, \ldots$, while the other copies are verifying the existence of the jumps $j_1, j_2, \ldots$.

The main copy starts in an initial state called search and goes non-deterministically in this state to the vertex γ_0. Here, it guesses $\delta_0 = (r_0, ., \alpha_0, q_1), r_1 and$ and c_1, such that δ_0 is in the label of γ_0. Either $q_1 = r_1$, in this case $\phi(q_1)$ should be c_1, or $q_1 \neq r_1$. In this last case, $\mathcal{A}_{path}$ launch a copy of $\mathcal{A}_{jump}$ in the direction α_0 and the state (c_1, q_1, r_1) to check the existence of a jump of priority c_1 from $q_1\gamma_0\alpha_0 = q_1\gamma_1$ to $r_1\gamma_1$.

Then, the main copy goes in the direction α_0. It must record in its state that it is discovering an exploration path that reaches the vertex $r_1\gamma_1$. If $c_1 = c$, its state is a Büchi state called (hit, r_1), otherwise it is a non-Büchi state called $(\text{regular}, r_1)$.

In an accepting computation, infinitely many of the jumps are of priority c, hence the priority of the exploration path is c.

Second case of lemma 10. This construction is similar. Some main copy guesses the sequence of push and pop transitions associated to the sequences $j_0, j_1, \ldots$ and $l_0, l_1, \ldots$, and uses $\mathcal{A}_{jump}$ to check the existence of the different jumps. □

Theorem 2. *For each player i and winning condition* $\mathrm{Vic} \in \{\mathrm{Parity}_d, \mathrm{Exp} \cup \mathrm{Parity}_d, \mathrm{Exp} \cap \mathrm{Parity}_d\}$*, the tree which associates with a stack γ the set $\{q \in Q : q\gamma$ is winning for $i\}$ is regular. One can compute a non-deterministic Büchi automata which recognizes it, whose size is $2^{\mathcal{O}(d|Q|^2+|\Gamma|)}$ if* $\mathrm{Vic} \in \{\mathrm{Parity}_d, \mathrm{Exp} \cup \mathrm{Parity}_d\}$ *and $2^{\mathcal{O}(d^2|Q|^2+d|\Gamma|)}$ if* $\mathrm{Vic} = \mathrm{Exp} \cap \mathrm{Parity}_d$.

Proof. For the games $Parity_d$ and $Exp \cup Parity_d$, this Theorem is a direct corollary of Theorem 1. In fact, we can build a Büchi alternating automaton which recognizes the language of couples (σ_0, σ_1) such that σ_i is a winning positional strategy for player i and the domains of σ_0 and σ_1 are a partition of $V_{\mathcal{P}}$. The winning sets are then obtained by projection, which requires a non-determinization, hence an exponential blowup of the state space of the alternating automaton [15]. For the game $Exp \cap Parity_d$, the situation is less simple since player 0 does not necessarily hold any positional strategy, and we shall make use of Proposition 3, that we recall:

Proposition 3. *Let $G = (V, E)$ be an arena coloured from 0 to $d \neq 0$. Let D be the set of vertices coloured by d. 0 wins the game $(G, \mathrm{Exp} \cap \mathrm{Parity}_d)$ on V if and only if there exists a subarena $G[W]$, coloured from 0 to $d-1$ such that:*

- **Case d even:** *0 wins the game $(G[W], \mathrm{Exp} \cap \mathrm{Parity}_{d-1})$, the game (G, Exp) and wins the game $(G, Attraction(D))$ on $V \backslash W$.*
- **Case d odd:** *0 wins the game $(G, Trap(W))$ with a positional strategy $\sigma_{Trap(W)}$, and wins the game $(G[W, \sigma_{Trap(W)}], \mathrm{Exp} \cap \mathrm{Parity}_{d-1})$.*

We define the notion of winning-proof, which is a tree on Γ^* labeled by tuples of subsets of Δ, and is defined such that the existence of a winning-proof in an arena is equivalent to the conditions of Proposition 3. Here follows the definition of a winning-proof in a subarena G of a pushdown arena $G_{\mathcal{P}}$.

In the case where $d = 0$, it is a strategy tree $T_{\sigma_{Exp}}$ winning the game (G, Exp).

In the case where $d > 0$ and is even, it is a tuple $T_d = (T', T_{\sigma_{Exp}}, T_{\sigma_{Att}}, T_{d-1})$ where

- T' is the coding tree of a subarena G' of G,
- $T_{\sigma_{Exp}}$ is a strategy tree winning the game (G, Exp),
- $T_{\sigma_{Att}}$ is a strategy tree winning the game $(G, Attraction(D))$ on $Dom(G')$,
- T_{d-1} is a $(d-1)$-winning proof in G'.

In the case where d is odd, it is a tuple $T_d = (T', T_{\sigma_{Trap}}, T_{d-1})$ where

- T' is the coding tree of a subarena G' of G,

- $T_{\sigma_{Trap}}$ is a strategy tree winning the game $(G, Trap(Dom(G')))$,
- T_{d-1} is a $(d-1)$-winning proof in G'.

Proposition 3 implies that the language of d-winning proofs in G is non-empty if and only if player 0 wins $(G, Exp \cap Parity_d)$. Moreover, a construction similar to that of the proof of Theorem 1 proves that this language is regular and recognized by an alternating automaton with $\mathcal{O}(d \mid Q \mid^2 + \mid \Gamma \mid)$ states.

Instead of simply constructing an automaton checking whether a tree $T : \Gamma^* \to 2^\Delta$ is a strategy tree, we must construct some automaton recognizing the language of couples $(T, T') : \Gamma^* \to 2^\Delta \times 2^\Delta$ such that T' is the coding tree of a subarena of G_P and T is the coding tree of a subarena of T' (resp. is a strategy in T'). It only requires a few elementary changes in subsection B.1.

Let σ be a positional strategy in an arena G. As in the case of the games $\{Parity_d, Exp \cup Parity_d, Exp \cap Parity_d\}$, there are simple conditions about the cycles and the exploration paths of $(Dom(\sigma), \sigma)$ for deciding whether σ wins the games $(G, Attraction(D))$ and $(G, Trap(Dom(G')))$:

σ does not win the game $(G, Attraction(D))$ if there is a cycle or an exploration path in $(Dom(\sigma), \sigma)$ which visits no vertex of priority d.

σ does not win the game $(G, Trap(Dom(G')))$ if $(Dom(\sigma), \sigma)$ contains a cycle that reach a vertex outside $Dom(G')$ or contains an infinite exploration path that reaches infinitely often a vertex that is not in $Dom(G')$. We modify $\mathcal{A}_{jump}$ and $\mathcal{A}_{path}$ in the following way.

Instead of reading a tree $T : \Gamma^* \to 2^\Delta$, those automata will read a couple of trees $(T, T') : \Gamma^* \to 2^\Delta \times 2^\Delta$, such that T is a strategy tree in the arena G' coded by T'.

We can modify the transition table of $\mathcal{A}_{jump}$ in such a way that they consider the priority of a state $q\gamma \in Q$ to be 1 if there is some q-transition in the label $T'(\gamma)$ and to be 0 otherwise. With this new colouring, an infinite exploration path reaches infinitely often a vertex that is not in $Dom(G')$ if and only if its priority is 0, and we can use $\mathcal{A}_{path}$ to look for those paths.

Finally, to check that some tree is a winning-proof, we need to check $\mathcal{O}(d)$ conditions using an alternating automata with at most $\mathcal{O}(d \mid Q \mid^2 + \mid \Gamma \mid)$ states. We obtain an alternating automaton with $\mathcal{O}(d^2 \mid Q \mid^2 + d \mid \Gamma \mid)$ states.

Thus, we can apply the same technique as in the positional case to construct the desired non-deterministic Büchi automaton. □