M1 internship

Emptiness problem for p-automata

Author :
Engel Lefaucheux

Supervisor:
Nir Piterman

ENS CACHAN

Juin/Juillet 2012

Contents

Introduction

The second year internship in the ENS offers the possibilty to get a better understanding of one particular domain of informatics. In my case, it was the occasion to see more of game and automata theories. Thanks to the advice of D. Berwanger and N. Fijalkow, I contacted N. Piterman who offered me to work on p-automaton, a kind of automata that uses games in its acceptance condition.

p-automata were created by N. Piterman, M. Huth et D. Wagner in 2010 in [1]. This new type of automata generalises alternating automata to read Markov chain. The acceptance condition is a two player game that checks whether the paths of the input Markov chain satisfy the p-automaton. As p-automata are recent, they are still evolving a little.In [2], N. Piterman simplifies the acceptance game.
My internship aimed to deal with an important question about p-automata: the emptiness problem (Given one p-automata, is the accepted language of this automaton empty?).

An application of p-automata can be found in verification. In [1], it is proved that given a PCTL formula, we can create a p-automaton that accepts the Markov chains that satisfy the formula. Thus the satisfiability problem (Given a PCTL formula, does there exists a Markov chain satisfying that formula?) can be easily reduced to the emptiness problem.

N. Piterman made me learn the concept of p-automaton by successive steps, by proving theorem on structures increasingly close to p-automaton. In the first part, I will describe some of those steps.
Then, you will learn what a p-automaton is and I will describe the main problem that was tackled during the internship: the emptiness problem for qualitative p-automaton.
Some problems were interesting but could not be finished in time, I will describe them in the last part.

1 From word automaton to p-automaton

1.1 Automata on infinite words

Automata on finite words were supposed to be known, so the first thing I had to learn was the generalisation to infinite words. Automata on infinite words (called ω-automata) mostly use four different acceptance conditions: Büchi, Rabin, Street and Parity conditions.

Definition 1: An infinite word on an alphabet Σ is a function from $\mathbb{N}$ to Σ.

Definition 2: A non-deterministic Büchi Automaton over words is a tuple $M = (\Sigma, S, I, \rho, F)$, where S is a finite set of states, Σ is an alphabet, $\rho : S \times \Sigma \to 2^S$ is the transition relation, $I \subseteq S$ is the set of initial states, and $F \subseteq S$ is the set of final states.
A run of a non-deterministic Büchi Automaton $M = (\Sigma, S, I, \rho, F)$ on a word ω is a word m on S such that $m(0) \in I$ and for all $i \in \mathbb{N}, m(i+1) \in \rho(m(i), \omega(i))$.
A run m on ω is accepted iff there is an infinite number of $i \in \mathbb{N}$ such that $m(i) \in F$.
The language accepted by a non-deterministic Büchi Automaton M is the set L(M) of all infinite words ω such that there is an accepting run of M on ω.

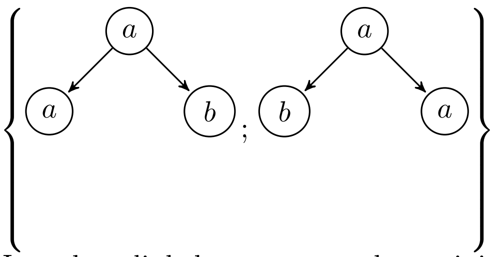
Definition 3: A deterministic Rabin automaton over words is a tuple $M = (\Sigma, S, s_0, \rho, F)$, where S and Σ are the same as in the definition of non-deterministic Büchi automaton, $\rho : S \times \Sigma \to S$ is a partial function called the transition relation, $s_0 \in S$ is the initial state, and F is a set of pair of set of states F $= \{(E_1, F_1), ..., (E_n, F_n)\}$.
A run of a deterministic Rabin automaton $M = (\Sigma, S, s_0, \rho, F)$ on a word ω is a word m on S such that $m(0) = s_0$ and for all $i \in \mathbb{N}, m(i+1) = \rho(m(i), \omega(i))$.
A run m on ω is accepted iff there exists i such that F_i is visited infinitely often whereas E_i is not. The language accepted by a deterministic Rabin automaton M is the set L(M) of all infinite words ω such that there is an accepting run of M on ω.

Theorem 1: Let $M = (\Sigma, S, I, \rho, F)$ be a non-deterministic Büchi automaton, there exists a deterministic Rabin automaton $M' = (\Sigma, S', s_0, \rho', F')$ such that $\mathcal{L}(M) = \mathcal{L}(M')$.

The proof that I made of this theorem can be found in the appendix 1. This theorem was previously prooved by S. Safra in [3]. An interesting fact is that non-deterministic Büchi, Rabin, Street, Parity automata and deterministic Rabin, Street and Parity automata recognize the same languages but deterministic Büchi automata are strictly weaker. Proofs of these can be found in [4].

1.2 Automata on trees

We want to get an automaton that reads Markov chains, as words are far from Markov chain, the next kind of automaton I worked with read trees. Deterministic automaton on trees are really weak, they cannot for example recognize the language



.

Hence instead of making a determinisation theorem, I made a link between non-deterministic automata and alternating automata (which are one step closer to p-automata).

Definition 4: Let $D \subseteq \mathbb{N}^+$ be the set of arities such that if $d \in D$ then for every $0 < d' < d$ we have $d' \in D$. A D-tree is a set $T \subseteq D^*$ such that if $x.n \in T$, for $x \in \mathbb{N}^*, n \in \mathbb{N}$, then $x \in T$, $n \in D$ and $x.m \in T$ for all $0 < m \leq n$.
A Σ-labeled D-tree is a pair (T, V), where T is a D-tree and $V : T \rightarrow \Sigma$.
An infinite path or branch on a D-tree T is a set $P \subseteq T$ such that $\varepsilon \in P$ and for every $x \in P$ there exists a unique $i \in \mathbb{N}$ such that $x.i \in P$.
In a tree (T,V), we define for every node of T $arity(x) = |\{i|x.i \in \mathbb{N}\}|$.

Definition 5: A non-deterministic Büchi Automaton over D-trees is a tuple $M = (\Sigma, D, S, I, \rho, F)$, where S is a finite set of states, Σ is an alphabet, ρ is the transition relation such that for every $s \in S, \sigma \in \Sigma$, and $d \in D$ we have $\rho(s, \sigma, d) \subseteq (S^d)^*$, $I \subseteq S$ is the set of initial states, and $F \subseteq S$ is the set of final states.
A run of a non-deterministic Büchi Automaton $M = (\Sigma, D, S, I, \rho, F)$ on a Σ-labeled D-tree (T, V) is a S-labeled D-tree (T, r) such that $r(\varepsilon) \in I$ and for all $x \in T, \exists d \in D$, x has arity d and $(r(x.1), ..., r(x.d)) \in \rho(r(x), V(x), d)$.
We focus on two acceptance conditions of a run:

- Finite tree acceptance : a run (T,r) is accepting if and only if there is no infinite path and every leaf x verifies $r(x) \in F$ and no $false$ transition are taken.
- Büchi acceptance : a run (T,r) is accepting if and only if there is no finite path and for every infinite path P in (T, r), there exist infinitely many $x \in P$ such that $r(x) \in F$ and P does not take any $false$ transition.

The language accepted by a non-deterministic Büchi Automaton M is the set L(M) of all Σ-labeled D-trees (T, V) such that there is an accepting run of M on (T, V).

Definition 6: Let $B^+(S)$ be the set of formulae using $true$, $false$, $\vee$, $\wedge$ and propositional variables from S.
An Alternating Büchi Automaton over D-trees is a tuple $A = (\Sigma, D, S, s_0, \rho, F)$, where S is a finite set of states, Σ is an alphabet, D is the set of arities of nodes, F is a set of accepting states, $s_0 \in S$ is an initial state and $\rho : S \times \Sigma \times D \rightarrow \mathcal{B}^+(\mathbb{N} \times S)$ is the transition function such that $\rho(s, \sigma, d) \in B^+(\{1, \ldots, d\} \times S)$. .
A run of an alternating Büchi Automaton $A = (\Sigma, D, S, s_0, \rho, F)$ on a Σ-labeled D-tree (T, V) is a $T \times S$-labeled tree (T_r, r) such that:

1. $\varepsilon \in T_r$ and $r(\varepsilon) = (\varepsilon, s_0)$.
2. if $y \in T_r$ with $r(y) = (x, s)$ and $\rho(s, V(x), n) = \theta$ (where n is the arity of x), then there is a (possibly empty) set $H = \{(c_1, s_1), ..., (c_n, s_n)\} \subseteq \mathbb{N} \times S$, such that the following hold:
   - H satisfies θ, which means that θ is $true$ if we set every state in H as $true$ and every state in S\H as $false$.
   - for all $1 \leq i \leq n, y.i \in T_r$, and $r(y.i) = (x.c_i, s_i)$.

We focus on two acceptance conditions of a run:

- Finite tree acceptance : a run (T_r, r) is accepting if and only if there is no infinite path and every leaf x veryfies $r(x) \in F$ and no $false$ transition are taken.
- Büchi acceptance : A run (T_r, r) is accepting if and only if for every infinite path P in (T_r, r), there exist infinitely many $x \in P$ such that $r(x) \in F$and every finite path either reads a transition true or ends on an accepting state.

The language accepted by an Alternating Büchi Automaton A is the set $L(A)$ of all Σ-labeled D-trees (T, V) such that there is an accepting run of A on (T, V).

Alternating automata and non-deterministic automata recognize the same languages. The proofs of the three following theorems are in the appendix 2.

Theorem 2: For every non-deterministic Büchi automaton on D-trees $\mathcal{A}$, there exists an alternating Büchi automaton on D-trees $\mathcal{A}'$ such that L($\mathcal{A}$)= L($\mathcal{A}'$).

Theorem 3: For every alternating Büchi automaton on D-trees with finite tree acceptance $\mathcal{A}$, there exists a non-deterministic Büchi automaton on D-trees with finite tree acceptance $\mathcal{A}'$ such that L($\mathcal{A}$)= L($\mathcal{A}'$).

Theorem 4: For every alternating Büchi automaton on D-trees with Büchi acceptance $\mathcal{A}$, there exists a non-deterministic Büchi automaton on D-trees with Büchi acceptance $\mathcal{A}'$ such that L($\mathcal{A}$)= L($\mathcal{A}'$).

Once again, those results were already known, check [5] for an alternative proof.

1.3 Emptiness problem for alternating Büchi tree automata

p-automata are an adaptation of alternating automata to Markov chain. It was thus natural to work a little on the emptiness problem of alternating Büchi tree automata (ABT). For this purpose, we created a game such that the input automaton accepts a tree if and only if player 0 has a winning strategy. It is left to decide whether player 0 has a winning strategy but this has already been done for games with Büchi condition (see [6]).

Definition 7: An **arena** is a triplet $A = (V_0, V_1, E)$ where V_σ is the set of vertices of the player σ, V_0 and V_1 are disjoint, we denote $V = V_0 \cup V_1$ and $E \subseteq V \times V$ is the edge relation.
The set of successors of $v \in V$ is $vE = \{v' \in V | (v, v') \in E\}$.
An infinite **play** in an arena is a path $\pi = v_0 v_1 ... \in V^\omega$ such that for all $i, v_{i+1} \in v_i E$.

Definition 8: A **game** G = (A,W) is defined by an arena A and a winning set $W \subseteq V^\omega$. player 0 is called the winner of a play π iff $\pi \in W$, otherwise player 1 is the winner.
Let $X : V \to C$ be a function mapping the vertices to a finite number of colors C. This function is called the coloring function. We extend X to play such that $X(\pi) = X(x_0)X(x_1)...$.
The winning set W_X *derived* from the Büchi condition is the set that consists of all the infinite plays π where $\mathrm{Inf}(X(\pi)) \cap F \neq \emptyset$ for a chosen set $F \subseteq C$ of accepting colors.
The winner of a finite play is the player whose opponent is unable to move.

Definition 9: Let A be an arena, $\sigma \in \{0, 1\}$, and $f_\sigma : V^* V_\sigma \to V$ a partial function. A prefix of a play $\pi = v_0 v_1 ... v_l$ is said to be conform with f_σ if, for every i with $0 \leq i < l$ such that

$v_i \in V_\sigma$, the function f_σ is defined at $v_0...v_i$ and we have $v_{i+1} = f_\sigma(v_0...v_i)$. We call f_σ a **strategy** on $U \subseteq V$ if it is defined for every prefix of a play that is conform with it and starts in a vertex in U.
For a game $G = (A, W)$, a strategy f_σ on U is a winning strategy if all plays that are conform with f_σ and start in a vertex in U are winning for player σ (i.e. in W iff $\sigma = 0$).

Let $A = (\Sigma, D, S, s_0, \rho, F)$ be an alternating Büchi automaton.
We construct the game $G_A = ((V_0, V_1, E), W)$ where:

- $V_0 = \{((\alpha_1, ..., \alpha_k), (\alpha_{k+1}, ...\alpha_{k'}))|\forall i, \alpha_i \in \mathcal{B}^+(\mathbb{N} \times S)$ and $\exists i, \alpha_i \notin \{true\} \cup \{false\} \cup \mathbb{N} \times S\} \cup \{((\beta_1, ..., \beta_k), (\beta_{k+1}, ...\beta_{k'})|\forall i, \beta_i \in S\} \cup \{false\}$.
- $V_1 = \{((\alpha_1, ..., \alpha_k), (\alpha_{k+1}, ...\alpha_{k'}))|\forall i, \alpha_i \in \{true\} \cup \{false\} \cup \mathbb{N} \times S\} \cup \{true\}$
- $E = \{(((\gamma_1, ..., \gamma_k), (\gamma_{k+1}, ...\gamma_{k'})), ((\gamma'_1, ..., \gamma'_k), (\gamma'_{k+1}, ...\gamma'_{k'})))|$
$\exists i \in \mathbb{N}, m \geq 2, \phi_1, ...\phi_m \in \mathcal{B}^+(\mathbb{N} \times S), \gamma_i = \bigvee_{n=1...m} \phi_n$ and $\exists j, \gamma'_i = \phi_j . \forall n \neq i, \gamma'_i = \gamma_i\}$
$\cup\{(((\gamma_1, ..., \gamma_k), (\gamma_{k+1}, ...\gamma_{k'})), ((\gamma'_1, ..., \gamma'_k), (\gamma'_{k+1}, ...\gamma'_{k'})))|$
$\exists i \in \mathbb{N}, m \geq 2, \phi_1, ...\phi_m \in \mathcal{B}^+(\mathbb{N} \times S), \gamma_i = \bigwedge_{n=1...m} \phi_n$ and $\gamma'_i = \phi_1, \phi_2, ..., \phi_m . \forall n \neq i, \gamma'_i = \gamma_i\}$
$\cup\{(((\alpha_1, ..., \alpha_k), (\alpha_{k+1}, ...\alpha_{k'})), ((\beta_1, ..., \beta_n), (\beta_{n+1}, ...\beta_{n'})))|\forall i, \alpha_i \in \{true\} \cup \{false\} \cup \mathbb{N} \times S$, and there exists i, $\{\beta_1, ..., \beta_n\} = \{q_j | j \leq k$ and $\alpha_j = (i, q_j)\}$ and $\{\beta_{n+1}, ..., \beta_{n'}\} = \{q_j | j > k$ and $\alpha_j = (i, q_j)\}\}$
$\cup\{(((\alpha_1, ..., \alpha_k), (\alpha_{k+1}, ...\alpha_{k'})), false)|\exists i, \alpha_i = false\}$
$\cup\{(((\alpha_1, ..., \alpha_k), (\alpha_{k+1}, ...\alpha_{k'})), true)|\forall i, \alpha_i = true\}$
$\cup\{(((q_1, ..., q_k), (q_{k+1}, ...q_{k'})), ((\phi_1, ..., \phi_n), (\phi_{n+1}, ...\phi_{k'})))|$ there exists $a \in \Sigma$ of arity d such that for all i, there exists j such that $\phi_j = \rho(q_i, a, d)$ and either k>0 then we have $j > n$ iff $i > k$ or $q_i \in F$ or $k = 0$ then we have $j \leq n\}$.
- The winning condition in this game is derived from a Büchi condition, the coloring function is X such that $X(s) = 0$ iff $s = ((), (s_1, ..., s_k))$ for some $s_1, ..., s_k \in S$ and $k \in \mathbb{N}^*$. 0 is the only accepting color.

Theorem 5: The language of the automaton A is not empty iff player 0 has a winning strategy starting from the state $((s_0), ())$.

The proof is in the appendix 3.

2 The emptiness problem of qualitative p-automata

2.1 P-automata and acceptance games

Definition 10: A p-automaton A is a tuple $\langle \Sigma, Q, \delta, \phi^{in}, \alpha \rangle$, where Σ is a finite input alphabet, Q is a finite set of states, $\delta : Q \to B^{+}(Q \cup \llbracket Q \rrbracket)$, where $\llbracket Q \rrbracket = \{\llbracket q \rrbracket_{>0}, \llbracket q \rrbracket_{\geq 1} | q \in Q\}$, is the transition fuction, $\phi^{in} \in B^{+}(\llbracket Q \rrbracket)$ the initial condition and $\alpha \subseteq Q$ an acceptance condition.

Intuitively, a state $q \in Q$ of a p-automaton and its transition structure model a probabilistic path set. So $\llbracket q \rrbracket_{\bowtie p}$ holds in location s if the measure of paths that begin in s and satisfy q is $\bowtie p$ (a path is winning if it visits α infinitely often).
Formally, for a p-automaton $A = \langle \Sigma, Q, \delta, \phi^{in}, \alpha \rangle$ and a Markov chain $M = (S, P, L, s^{in})$, we decide whether A accepts M by a reduction to a turn-based stochastic Büchi game with obligations (this reduction was introduced in [2]).
We construct a game $G_{M,A}$. A configuration of $G_{M,A}$ corresponds to a subformula appearing in the transition of A and a location in M. Configurations with a term of the form $\llbracket q \rrbracket_{\bowtie p}$ correspond to obligations. All other configurations have no obligations. Let $G_{M,A} = ((V, E), (V_0, V_1, V_p), \kappa, G)$, where :

- $V = S \times cl(\delta(Q, \Sigma))$ with $cl(\Phi) = \sum_{\phi \in \Phi} cl(\varphi)$ for Φ a set of formulae and for φ a formula $cl(\varphi)$ is the set of subformulae of φ.
- $V_0 = \{(s, \psi_1 \vee \psi_2) | s \in S \text{ and } \psi_1 \vee \psi_2 \in cl(\delta(Q, \Sigma))\}$.
- $V_1 = \{(s, \psi_1 \wedge \psi_2) | s \in S \text{ and } \psi_1 \wedge \psi_2 \in cl(\delta(Q, \Sigma))\}$.
- $V_p = S \times (Q \cup \llbracket Q \rrbracket)$ represents the set of probabilistic states. From $c \in V_p$, we have a probability $\kappa(c, c')$ to go in c'.
- $E = \{(s, \psi_1 \wedge \psi_2); (s, \psi_i)) | i \in \{1, 2\}\} \cup \{(s, \psi_1 \vee \psi_2); (s, \psi_i)) | i \in \{1, 2\}\} \cup$ $\{(s, \llbracket q \rrbracket_{\bowtie p}), (s', \delta(q, L(s)))) | s' \in succ(s)\} \cup \{(s, q), (s', \delta(q, L(s)))) | s' \in succ(s)\}$.
- $\kappa((s; q); (s', \delta(q, L(s)))) = \kappa(s, \llbracket q \rrbracket_{\bowtie p}), (s', \delta(q, L(s)))) = P(s, s')$.
- $G = (\alpha', O)$, where
  - For $q \in Q$ and $p \in [0; 1]$ we have $(s, q) \in \alpha' \Leftrightarrow q \in \alpha$, $(s, \llbracket q \rrbracket_{\bowtie p}) \in \alpha' \Leftrightarrow q \in \alpha$.
  - For $q \in Q$ and $p \in [0; 1]$ we have $O(s, \llbracket q \rrbracket_{\bowtie p}) = \bowtie p$. For every other configuration c, we have $O(c) = \bot$. An obligation raised in a configuration $(s, \llbracket q \rrbracket_{\bowtie p})$ is satisfied if, with the chosen strategy, the measure of paths starting in this configuration and that reach a satisfied obligation or that visit α' infinitely often without raising a new obligation is $\bowtie p$.

The Markov chain is accepted if and only if player 0 has a strategy from the configuration $(\phi^{in}; s^{in})$ in $G_{M,A}$ such that the first obligation raised (there is one as ϕ^{in} is a conjonction of elements of $\llbracket Q \rrbracket$) is satisfied. The language of A is $L(A) = \{M | M \text{ is accepted}\}$.
We call qualitative a p-automaton were $\bowtie p$ is restricted to $\{> 0; \geq 1\}$.

Theorem 7: Given a p-automaton $A = \langle \Sigma, Q, \delta, \phi^{in}, \alpha \rangle$, its language $L(A)$ is well defined.

Theorem 8: Let $A = (\Sigma, Q, \delta, \phi^{in}, \alpha)$ be a p-automaton, there exists a finite set of p-automata $A_i = (\Sigma, Q, \delta, \phi_i^{in}, \alpha)$ such that ϕ_i^{in} is a conjunction over $\llbracket Q \rrbracket$ and $L(A) \neq \emptyset$ iff $\exists i, L(A_i) \neq \emptyset$. Furthermore, the number of automata A_i is at most exponential in the number of states of A.

Theorem 9: Let $A = (\Sigma, Q, \delta, \phi^{in}, \alpha)$ be a qualitative p-automaton where ϕ^{in} is a conjunction over $[\![Q]\!]$ then there is a p-automaton $A' = (\Sigma, Q \cup \{q_0\}, \delta, \phi_a^{in}, \alpha)$ such that $L(A) \neq \emptyset$ iff $L(A') \neq \emptyset$ and ϕ^{in} is a conjunction over $[\![Q]\!]$ that contains at most one literal in $\{[\![q]\!]_{>0} | q \in Q\}$.

From now on, when dealing with a qualitative p-automaton, we assume that ϕ^{in} is a conjunction over $[\![Q]\!]$ and contains at most one literal in $\{[\![q]\!]_{>0} | q \in Q\}$.

Definition 11: A *choice* is $G \in 2^{Q \times (Q \cup [\![Q]\!])}$. For a choice G and $q \in Q$, let $G(q) = \{c | (q, c) \in G\}$. For an automaton $A = \langle \Sigma, Q, \delta, \phi^{in}, \alpha \rangle, G \subseteq Q \times (Q \cup [\![Q]\!])$ is feasible for a letter a iff $\forall q \in Q, G(q) = \emptyset$ or $G(q) \models \delta(q, a)$.
Let $\mathfrak{G}$ denote the set of all feasible choices.

Definition 12: A *p-obligation* over a set of states Q and an alphabet Σ is a label of the form (O, Z, k) where O, Z are subsets of $Q \cup [\![Q]\!]$, $k \in \Sigma$.
Clearly the number of such p-obligations is exponential in the number of states of Q and linear in the size of Σ.

2.2 A complex emptiness game

Let $A = (\Sigma, Q, \delta, \phi^{in}, \alpha)$ be a qualitative p-automaton.
We construct the game $G_A = ((V_0, V_1, E), W)$ where:

- $V_0 = 2^{Q \cup [\![Q]\!]} \times 2^{Q \cup [\![Q]\!]} \times \{c, n\} \times 2^{Q \times (Q \cup [\![Q]\!])} \cup 2^{Q \cup [\![Q]\!]} \times 2^{Q \cup [\![Q]\!]} \times \{c, n\} \times \Sigma$.
- $V_1 = 2^{Q \cup [\![Q]\!]} \times 2^{Q \cup [\![Q]\!]} \times \{p_1\} \times \Sigma$.
- $E =$

$$\begin{aligned}&\{((O,Z,k,G),(O,Z,p_1,a)) \mid a \in \Sigma\ \}\\ &\cup \left\{ ((O,Z,p_1,a),(O',Z',n,a)) \;\middle|\; \begin{array}{l} \exists [\![q]\!]_{>0} \in O \cup Z, Z' = \{q\}, \\ O' = O \cap (Q \cup [\![Q]\!]_{\geq 1}) \cup Z \cap [\![Q]\!]_{\geq 1} \end{array} \right\}\\ &\cup \left\{ ((O,Z,p_1,a),(O',Z',c,a)) \;\middle|\; \begin{array}{l} Z' = Z \cap Q, \\ O' = O \cap (Q \cup [\![Q]\!]_{\geq 1}) \cup Z \cap [\![Q]\!]_{\geq 1} \end{array} \right\}\\ &\cup \left\{ ((O,Z,k,a),(O',Z',k,G)) \;\middle|\; \begin{array}{l} k \in \{c,n\}, G \text{ is a choice feasible for } a \text{ where for all } t \in O \cup Z, \\ q = t \text{ if } t \in Q, [\![q]\!]_{\bowtie p} = t \text{ if } t \in [\![Q]\!], G(q) \models \delta(q,a) \\ Z' = \bigcup_{q \in Z \cap Q} G(q) \\ O' = (\bigcup_{t \in O \cap Q} G(t)) \cup (\bigcup_{q \in (O \cup Z) \cap [\![Q]\!]_{\geq 1}} G(q)) \end{array} \right\}\end{aligned}$$

Intuitively, configurations of the form (O, Z, k, G) are player 0 states, which correspond to a location of a Markov chain. From such configurations, player 0 chooses a letter to label the location.
In such a configuration, player 0 needs to show that all the unbounded states in O accept with probability 1, that all the unbounded states in Z accept with positive probability, and that all bounded states in both O and Z satisfy their obligations.
Accordingly, after player 0 chooses the letter, the configuration is a player 1 configuration of the form (O, Z, p_1, a) and it is the choice of player 1 whether to *continue* with the same >0 obligation (going to a configuration of the form (O, Z, c, a)) or to follow a *new* >0 obligation (going to a configuration of the form (O, Z, n, a)).
After the choice of player 1, it is the turn of player 0 to choose how to satisfy the transitions of all the states of A she is following simultaneously. player 0 does this by making a choice G feasible for a and moving to a configuration of the form (O, Z, k, G).

- Consider a play
$\pi = (O_0, Z_0, c, \emptyset), (O_0, Z_0, p_1, a_0), (O'_0, Z'_0, k_1, a_0), (O_1, Z_1, k_1, G_1), (O_1, Z_1, p_1, a_1)(O'_1, Z'_1, k_1)...$
A run $t_0t_1t_2...$ is possible in π iff $\forall i \geq 0, t_i \in O_i \cup Z_i$ and for $t_i = q_i$ or $t_i = [\![q_i]\!]_{\bowtie p}$ we have $(q_i, t_{i+1}) \in G_{i+1}$.
A run $t_0t_1t_2...$ is bounded iff it is possible in π and $t_i = [\![q_i]\!]_{\bowtie p}$ infinitely often ($\bowtie p \in \{>0, \geq 1\}$).
A run $t_0t_1t_2...$ is unbounded iff it is possible in π and there exists i such that $\forall j > i, t_j = q_j$. That is, a run is unbounded iff it is not bounded.
A run $t_0t_1t_2...$ is existential iff it is possible in π, unbounded and the last time that $t_i \in [\![Q]\!]$, we have $t_i = [\![q_i]\!]_{>0}$.
A run $t_0t_1t_2...$ is universal iff it is possible in π, unbounded and the last time that $t_i \in [\![Q]\!]$, we have $t_i = [\![q_i]\!]_{\geq 1}$.
A play is winning for player 0 if all of the following hold:

  1. Every bounded run $t_0t_1t_2...$ verifies $t_i \in \alpha \cup [\![\alpha]\!]$ infinitely often.
  2. Every existential run $t_0t_1t_2...$ verifies $t_i \in \alpha \cup [\![\alpha]\!]$ infinitely often.
  3. Every universal run $t_0t_1t_2...$ verifies $t_i \in \alpha \cup [\![\alpha]\!]$ infinitely often or $k_i = n$ infinitely often.

  Otherwise it is winning for player 1.

Theorem 10: The language of A is not empty iff player 0 has a winning strategy starting from a location $(O, \emptyset, c, \emptyset)$ where $O \models \phi^{in}$.

It is the most important result of my internship, you can find the proof in the appendix 4.

2.3 Translation to a Parity game

Unfortunately, this game does not follow an usual winning condition. We decided to transform it to a parity game by making the product of G_A with an automaton on words that reads the play, decides if the play is winning and follows a parity condition.
We first construct three automata, each checking one kind of run.

Part 1: the bounded runs

We construct the non-deterministic Parity automaton on words $B_b = (\Sigma', Q', \delta', s_0, F)$ where :

- $Q' = (Q \times \{v, b\}) \cup \{s_0, s_{reject}\}$
- $\Sigma' = V_0 \cup V_1$
- For $q \in Q$, a p-obligation (O, Z, k), a choice G and $a \in \Sigma$,
$\delta'(s_0, (O, Z, k, G)) = \{(q, b) | [\![q]\!]_{\bowtie p} \in O\}$ if $(O, Z, k, G) = (O, \emptyset, c, \emptyset)$ where $O \models \phi^{in}$,
$\delta'(s_0, (O, Z, k, G)) = \{s_{reject}\}$ in the other case.
$\delta'((q, v), (O, Z, k, G)) = \{(q', v) | q' \in G(q) \cap Q\} \cup \{(q', b) | [\![q']\!]_{\bowtie p} \in G(q)\}$.
$\delta'((q, b), (O, Z, k, G)) = \{(q', v) | q \in \alpha \text{ and } q' \in G(q) \text{ or } [\![q']\!]_{\bowtie p} \in G(q)\} \cup \{(q', b) | q \notin \alpha \text{ and } q' \in G(q) \text{ or } [\![q']\!]_{\bowtie p} \in G(q)\}$.
$\delta'(s_{reject}, (O, Z, k, G)) = \{s_{reject}\}$.
For $i \in \{v, b\}$, $\delta'((q, i), (O, Z, k, a)) = \{(q, i)\}$.
we are in (q, b) if we raised a new obligation recently, we are in (q, v) if we visited α recently.

- $Q \times \{v\}$ has parity 3, $Q \setminus \alpha \times \{b\}$ has parity 2 and $\alpha \times \{b, v\}$ has parity 1.

Theorem 11: Let π be an infinite play, the word associated to π is accepted by B_b iff there is a bounded run in π that is winning for player 1.

Part 2: the universal runs

We construct the non-deterministic co-Büchi automaton on words $B_u = (\Sigma', Q', \delta', s_0, F)$ where :

- $Q' = (Q \times \{b, c, n\}) \cup \{s_0, s_{reject}\}$
- $\Sigma' = V_0 \cup V_1$
- For $q \in Q$, a p-obligation (O, Z, k), a choice G and $a \in \Sigma$,
  $\delta'(s_0, (O, Z, k, G)) = \{(q, b) | [\![q]\!]_{\bowtie p} \in O\} \cup \{(q, c) | [\![q]\!]_{\geq 1} \in O\}$ if $(O, Z, k, G) = (O, \emptyset, c, \emptyset)$ where $O \models \phi^{in}$, $\delta'(s_0, (O, Z, k, G)) = \{s_{reject}\}$ in the other case.
  $\delta'((q, b), (O, Z, k, G)) = \{(q', b) | q' \in G(q) \cap Q$ or $[\![q']\!]_{\bowtie p} \in G(q)\} \cup \{(q', c) | [\![q']\!]_{\geq 1} \in G(q)\}$.
  $\delta'((q, c), (O, Z, k, G)) = \{(q', c) | q' \in G(q) \cap Q, k \neq n\} \cup \{(q', n) | q' \in G(q) \cap Q, k = n\}$.
  $\delta'((q, n), (O, Z, k, G)) = \{(q', c) | q' \in G(q) \cap Q, k \neq n\} \cup \{(q', n) | q' \in G(q) \cap Q, k = n\}$.
  $\delta'(s_{reject}, (O, Z, k, G)) = \{s_{reject}\}$.
  For $i \in \{b, c, n\}$, $\delta'((q, i), (O, Z, k, a)) = \{(q, i)\}$.
  we are in (q, b) until we raise the obligation that player 1 wants to follow, then we are in (q, n) or (q, c) depending on if the last location visited had $k = n$ or not.
- $F = (Q \times \{b, n\}) \cup (\alpha \times \{c\})$

Theorem 12: Let π be an infinite play, the word associated to π is accepted by B_u iff there is an universal run in π that is winning for player 1.

Part 3: the existential runs

We construct the non-deterministic co-Büchi automaton on words $B_e = (\Sigma', Q', \delta', s_0, F)$ where :

- $Q' = (Q \times \{b, e\}) \cup \{s_0, s_{reject}\}$
- $\Sigma' = V_0 \cup V_1$
- For $q \in Q$, a p-obligation (O, Z, k), a choice G and $a \in \Sigma$,
  $\delta'(s_0, (O, Z, k, G)) = \{(q, b) | [\![q]\!]_{\bowtie p} \in O\} \cup \{(q, e) | [\![q]\!]_{>0} \in O\}$ if $(O, Z, k, G) = (O, \emptyset, c, \emptyset)$ where $O \models \phi^{in}$, $\delta'(s_0, (O, Z, k, G)) = \{s_{reject}\}$ in the other case.
  $\delta'((q, b), (O, Z, k, G)) = \{(q', b) | q' \in G(q) \cap Q$ or $[\![q']\!]_{\bowtie p} \in G(q)\} \cup \{(q', e) | [\![q']\!]_{>0} \in G(q)\}$.
  $\delta'((q, e), (O, Z, k, G)) = \{(q', e) | q' \in G(q)\}$.
  $\delta'(s_{reject}, (O, Z, k, G)) = \{s_{reject}\}$.
  For $i \in \{b, e\}$, $\delta'((q, i), (O, Z, k, a)) = \{(q, i)\}$.
  we are in (q, b) until we raise the obligation that player 1 wants to follow, then we are in (q, e).
- $F = (Q \times \{b\}) \cup (\alpha \times \{e\})$

Theorem 13: Let π be an infinite play, the word associated to π is accepted by B_e iff there is an existential run in π that is winning for player 1.

Part 4: application to the game

Let C be the automaton resulting of the complementation of the union of B_b, B_e and B_u.

Theorem 14: Let π be an infinite play, the word associated to π is accepted by C iff this play is winning for player 0.

Theorem 15: The language of A is not empty iff player 0 has a winning strategy in $G_A \times D$ starting from a location $((O, \emptyset, c, \emptyset), s_0)$ where $O \models \phi^{in}$.

As this game follows an usual acceptance condition, there already exists an algorithm solving this kind of game.
So, what is the complexity of the algorithm we made for the emptiness problem? If we can prove that player 0 has a winning strategy in $G_A \times D$ if and only if he has a memoryless winning strategy (we are convinced that it is true but we did not prove it yet), then this algorithm is in EXPTIME. Moreover, [1] gives a polynomial reduction of the satisfiability problem of a PCTL formula to the emptiness problem of a p-automaton. As the satisfiability problem for qualitative PCTL is EXPTIME-complete (see [7]), this implies that the emptiness problem for qualitative p-automaton is EXPTIME-hard. Thus, assuming that our conjecture holds, it is EXPTIME-complete.

3 Linked problems

The former theorem gives an algorithm that decides if a Markov chain is accepted by a qualitative p-automaton. We then tried to make games that give a more precise answer.

3.1 Emptiness game, finite case

Can we change the former game so that player 0 would have a winning strategy in the game if and only if the p-automaton accepts a finite Markov chain? Here is the game that we created to answer this question.

Let $A = (\Sigma, Q, \delta, \phi^{in}, \alpha)$ be a qualitative p-automaton.
We construct the game $G_A = ((V_0, V_1, E), W)$ where:

- $V_0 = 2^{Q \cup [\![Q]\!]} \times 2^{Q \cup [\![Q]\!]} \times \{c, n, c_e, n_e\} \times 2^{Q \times (Q \cup [\![Q]\!])} \cup 2^{Q \cup [\![Q]\!]} \times 2^{Q \cup [\![Q]\!]} \times \{c, n, c_e, n_e\} \times \Sigma$.
- $V_1 = 2^{Q \cup [\![Q]\!]} \times 2^{Q \cup [\![Q]\!]} \times \{p_1, o, e\} \times \Sigma$.
- $E =$

$$
\begin{aligned}
&\left\{((O,Z,k,G),(O,Z,p_1,a)) \,\middle|\, k \in \{c,n\}, a \in \Sigma \right\}\\
&\cup \left\{((O,Z,k,G),(O,Z,o,a)) \,\middle|\, k \in \{c,n\}, a \in \Sigma \right\}\\
&\cup \left\{((O,Z,p_1,a),(O',Z',c,a)) \,\middle|\, \begin{array}{l} Z' = Z \cap Q, \\ O' = O \cap (Q \cup [\![Q]\!]_{\geq 1}) \cup Z \cap [\![Q]\!]_{\geq 1} \end{array} \right\}\\
&\cup \left\{((O,Z,p_1,a),(O',Z',n,a)) \,\middle|\, \begin{array}{l} \exists [\![q]\!]_{>0} \in O \cup Z, Z' = \{q\}, \\ O' = O \cap (Q \cup [\![Q]\!]_{\geq 1}) \cup Z \cap [\![Q]\!]_{\geq 1} \end{array} \right\}\\
&\cup \left\{((O,Z,k,a),(O',Z',k,G)) \,\middle|\, \begin{array}{l} k \in \{c,n\}, G \text{ is a choice feasible for } a \text{ where for all } t \in O \cup Z, \\ q = t \text{ if } t \in Q, [\![q]\!]_{\bowtie p} = t \text{ if } t \in [\![Q]\!], G(q) \models \delta(q,a) \\ Z' = \bigcup_{q \in Z \cap Q} G(q) \\ O' = (\bigcup_{t \in O \cap Q} G(t)) \cup (\bigcup_{q \in (O \cup Z) \cap [\![Q]\!]_{\geq 1}} G(q)) \end{array} \right\}\\
&\cup \left\{((O,Z,o,a),(O',Z',c,a)) \,\middle|\, \begin{array}{l} Z' = Z \cap Q, \\ O' = O \cap (Q \cup [\![Q]\!]_{\geq 1}) \cup Z \cap [\![Q]\!]_{\geq 1} \end{array} \right\}\\
&\cup \left\{((O,Z,o,a),(O',Z',n,a)) \,\middle|\, \begin{array}{l} \exists [\![q]\!]_{>0} \in O \cup Z, Z' = \{q\}, \\ O' = O \cap (Q \cup [\![Q]\!]_{\geq 1}) \cup Z \cap [\![Q]\!]_{\geq 1} \end{array} \right\}\\
&\cup \left\{((O,Z,o,a),(O',Z',\emptyset,c_e,a)) \,\middle|\, \begin{array}{l} Z' = Z \cap Q, \\ O' = O \cap (Q \cup [\![Q]\!]_{\geq 1}) \cup Z \cap [\![Q]\!]_{\geq 1} \end{array} \right\}\\
&\cup \left\{((O,Z,o,a),(O',Z',Z'',n_e,a)) \,\middle|\, \begin{array}{l} Z' = Z \cap Q, \exists [\![q]\!]_{>0} \in O \cup Z, Z'' = \{q\}, \\ O' = O \cap (Q \cup [\![Q]\!]_{\geq 1}) \cup Z \cap [\![Q]\!]_{\geq 1} \end{array} \right\}\\
&\cup \left\{((O,Z_1,Z_2,k,G),(O,Z_1,Z_2,e,a)) \,\middle|\, k \in \{c_e,n_e\}, a \in \Sigma \right\}\\
&\cup \left\{((O,Z_1,Z_2,e,a),(O',Z_1',Z_2',c_e,a)) \,\middle|\, \begin{array}{l} Z_i' = Z_i \cap Q, \\ O' = O \cap (Q \cup [\![Q]\!]_{\geq 1}) \cup Z \cap [\![Q]\!]_{\geq 1} \end{array} \right\}\\
&\cup \left\{((O,Z_1,Z_2,e,a),(O',Z_1',Z_2',n_e,a)) \,\middle|\, \begin{array}{l} Z_1' = Z_1 \cap Q, \exists [\![q]\!]_{>0} \in O \cup Z_1 \cup Z_2, Z_2' = \{q\}, \\ O' = O \cap (Q \cup [\![Q]\!]_{\geq 1}) \cup (Z_1 \cup Z_2) \cap [\![Q]\!]_{\geq 1} \end{array} \right\}\\
&\cup \left\{((O,Z_1,Z_2,k,a),(O',Z_1',Z_2',k,G)) \,\middle|\, \begin{array}{l} k \neq e, G \text{ is a choice feasible for } a \text{ where for all} \\ t \in O \cup Z_1 \cup Z_2, \\ q = t \text{ if } t \in Q, [\![q]\!]_{\bowtie p} = t \text{ if } t \in [\![Q]\!], G(q) \models \delta(q,a) \\ Z_i = \bigcup_{q \in Z_i \cap Q} G(q) \\ O' = (\bigcup_{t \in O \cap Q} G(t)) \cup (\bigcup_{q \in (O \cup Z_1 \cup Z_2) \cap [\![Q]\!]_{\geq 1}} G(q)) \end{array} \right\}
\end{aligned}
$$

The game starts similarly as the previous one: in (O, Z, k, G), player 0 chooses a letter that leads to (O, Z, p_1, a) where player 1 chooses the >0 obligation that he wants to follow and

in (O, Z, k, a), player 0 creates the choice feasible for a.
But from a configuration (O, Z, k, G), player 0 can also go to a configuration of the form (O, Z, o, a). This offers the choice to player 1 to either continue the first phase by going to a location (O', Z', k, a) with $k \in \{c, n\}$ or to go to the second phase by choosing a location (O', Z', Z'', k, a) with $k \in \{c_e, n_e\}$. From this point onwards, the only configurations visited have the form (O, Z_1, Z_2, k, a) and (O, Z_1, Z_2, k, G) with $k \in \{c_e, n_e, e\}$ corresponding to an *end component* of the input Markov chain. Player 0 chooses a letter in (O, Z_1, Z_2, k, G) and a choice in (O, Z_1, Z_2, k, a) with $k \in \{c_e, n_e\}$ and player 1 chooses which obligation he wants to follow in Z_2. It is similar to the previous game except that player 1 can not change the obligation in Z_1 and he can add another obligation in Z_2, player 0 has to follow Z_1 and Z_2 simultaneously.

- The definitions of play, possible, bounded, unbounded, universal and existential run are as before.
  A play is winning for player 0 if all of the following hold:
  1. If there is an existential run in this play, there exists $i \in \mathbb{N}$ such that $k_i = e$ or $k_i = o$ infinitely often.
  2. Every bounded run $t_0 t_1 t_2 ...$ verifies $t_i \in \alpha \cup [\![\alpha]\!]$ infinitely often.
  3. Every existential run $t_0 t_1 t_2 ...$ verifies $t_i \in \alpha \cup [\![\alpha]\!]$ infinitely often.
  4. Every universal run $t_0 t_1 t_2 ...$ verifies $t_i \in \alpha \cup [\![\alpha]\!]$ infinitely often or $k_i = n$ or $k_i = n_e$ infinitely often.

  Otherwise it is winning for player 1.

We are currently trying to prove the following conjecture: the language of A contains a finite Markov chain iff player 0 has a winning strategy starting from a location $(O, \emptyset, c, \emptyset)$ where $O \models \phi^{in}$.

3.2 Emptiness game, bounded case

A bounded Markov chain is a potentially infinite Markov chain such that there exists $\epsilon > 0$ inferior to the probability of every transition of the Markov chain. So the question is now : can we change the former game so that player 0 would have a winning strategy in the game if and only if the p-automaton accepts a bounded Markov chain? Here is the game that we created to answer this question.

Let $A = (\Sigma, Q, \delta, \phi^{in}, \alpha)$ be a qualitative p-automaton.
We construct the game $G_A = ((V_0, V_1, E), W)$ where:

- $V_0 = 2^{Q \cup [\![Q]\!]} \times 2^{Q \cup [\![Q]\!]} \times 2^{Q \cup [\![Q]\!]} \times \{p_0^1, p_0^2\} \times 2^{Q \times (Q \cup [\![Q]\!])} \cup 2^{Q \cup [\![Q]\!]} \times 2^{Q \cup [\![Q]\!]} \times 2^{Q \cup [\![Q]\!]} \times \{p_0^1, p_0^2\} \times \Sigma$.
- $V_1 = 2^{Q \cup [\![Q]\!]} \times 2^{Q \cup [\![Q]\!]} \times 2^{Q \cup [\![Q]\!]} \times \{p_1^1, p_1^2\} \times \Sigma$.

- $E =$

$$
\begin{aligned}
&\left\{((O,Z,\emptyset,p_0^1,G),(O,Z,\emptyset,p_1^1,a)) \;\middle|\; a\in\Sigma \right\}\\
&\cup\left\{((O,Z,\emptyset,p_0^1,G),(O,Z,\emptyset,p_1^2,a)) \;\middle|\; a\in\Sigma \right\}\\
&\cup\left\{((O,Z,\emptyset,p_0^1,a),(O',Z',\emptyset,p_0^1,G)) \;\middle|\; \begin{array}{l} G \text{ is a choice feasible for } a \text{ where for all } t\in O\cup Z,\\ q=t \text{ if } t\in Q, [\![q]\!]_{\bowtie p}=t \text{ if } t\in [\![Q]\!], G(q)\models \delta(q,a)\\ Z'=\bigcup_{q\in Z\cap Q} G(q)\\ O'=(\bigcup_{t\in O\cap Q} G(t))\cup(\bigcup_{q\in (O\cup Z)\cap [\![Q]\!]_{\geq 1}} G(q)) \end{array}\right\}\\
&\cup\left\{((O,Z,\emptyset,p_1^1,a),(O',Z',\emptyset,p_0^1,a)) \;\middle|\; \begin{array}{l} Z'=Z\cap Q \text{ or } \exists [\![q]\!]_{>0}\in O\cup Z, Z'=\{q\},\\ O'=O\cap(Q\cup [\![Q]\!]_{\geq 1})\cup Z\cap [\![Q]\!]_{\geq 1} \end{array}\right\}\\
&\cup\left\{((O,Z_1,Z_2,p_0^2,G),(O,Z_1,Z_2,p_1^2,a)) \;\middle|\; a\in\Sigma \right\}\\
&\cup\left\{((O,Z_1,Z_2,p_1^2,a),(O',Z_1',Z_2',p_0^2,a)) \;\middle|\; \begin{array}{l} Z_1'=Z_1 \text{ or } Z_1'=Z_2\\ \text{ or } \exists [\![q]\!]_{>0}\in O\cup Z_1\cup Z_2, Z_1'=\{q\}\\ \text{ and } Z_2'=Z_2 \text{ or } Z_2'=\emptyset\\ \text{ or } \exists [\![q]\!]_{>0}\in O\cup Z_1\cup Z_2, Z_2'=\{q\} \end{array}\right\}\\
&\cup\left\{((O,Z_1,Z_2,p_0^2,a),(O',Z_1',Z_2',p_0^2,G)) \;\middle|\; \begin{array}{l} G \text{ is a choice feasible for } a \text{ where for all } t\in O\cup Z,\\ q=t \text{ if } t\in Q, [\![q]\!]_{\bowtie p}=t \text{ if } t\in [\![Q]\!], G(q)\models \delta(q,a)\\ Z_1'=(\bigcup_{q\in Z_1\cap Q} G(q)), Z_2'=(\bigcup_{q\in Z_2\cap Q} G(q))\\ O'=(\bigcup_{q\in O\cap Q} G(q))\cup(\bigcup_{[\![q]\!]_{\geq 1}\in (O\cup Z_1\cup Z_2)} G(q)) \end{array}\right\}
\end{aligned}
$$

  This game has two phases. In the first phase, player 0 chooses the letter and creates the choice while player 1 decides which >0 obligation he wants to follow. It is as before except that we do not check whether the >0 obligation is a new one or the continuation of an old one. Thus we only use p_0^1 and p_1^1 to determine if it iss a player 0 or player 1 location.
  After a finite time, player 0 decides to start the second phase by going in a configuration of the form $(O,Z,\emptyset,p_1^2,a)$ from a configuration of the form $(O,Z,\emptyset,p_0^1,G)$. In this second phase, player 1 can follow a second obligation in Z_2. player 0 must be able to win (by visiting α infinitely often in every run) if player 1 decides to keep only Z_1 or Z_2.

- The definition of play, possible, bounded, unbounded, universal and existential run is as before.
  A play is winning for player 0 if all of the following hold:

  1. There exists $i\in\mathbb{N}$ such that $k_i=p_1^2$.
  2. Every possible run $t_0t_1t_2...$ verifies $t_i\in\alpha\cup[\![\alpha]\!]$ infinitely often.

  Otherwise it is winning for player 1.

We are currently trying to prove the following conjecture: the language of A contains a bounded Markov chain iff player 0 has a winning strategy starting from a location $(O,\emptyset,\emptyset,p_0^1,\emptyset)$ where $O\models\phi^{in}$.

3.3 Emptiness game for p-automata

We made games for qualitative p-automata, can we make a game for automata without the qualitative hypothesis? Here are some ideas to prove the question undecidable. My goal was to reduce the acceptance of p-automata to a variant of the Post Correspondence Problem.

Let Σ be an alphabet.

We define a function h such that for $w \in \Sigma^*, a \in \Sigma$, f, g functions from Σ^* to $\{1;2\}^*$ and $k = max_{a\in\Sigma, v\in\{f;g\}}(|v(a)|+1)$, $h(f,g,\varepsilon) = 0$ and $h(f,g,aw) = i(f(a)) + \frac{h(f,g,w)}{3^k}(1 - i(f(a)) - i(g(a)))$ where for $a_1...a_n \in \{1;2\}, i(a_1...a_n) = \frac{1}{2}\sum_{i=0}^{n}\frac{a_i}{3^i}$.

The following conjecture is close to the Post Correspondence Problem but we were not able to prove it for now.

Conjecture: The following problem is undecidable: Given two morphisms f and g from words on a finite alphabet Σ to $\{1;2\}^*$, does there exist $n \in \mathbb{N}$, words $w_1, ...w_n \in \Sigma^+$ and probabilities $\alpha_1...\alpha_n \in \mathbb{R}^+$ such that $\alpha_1 h(f,g,w_1) + ...\alpha_n h(f,g,w_n) = \alpha_1 h(g,f,w_1)...\alpha_n h(g,f,w_n)$?

For two morphisms f and g from words on a finite alphabet Σ to $\{1;2\}^*, k = max_{a\in\Sigma, v\in\{f;g\}}(|v(a)|+1)$, we create the p-automaton $A = \{\{p, p', c, c', f, f'\} \cup \Sigma, Q, \delta, \phi^{in}, \emptyset\}$ where:

- $Q = \{q, q', q_f, q_{f'}, q_c, q_p, q_{p'}, q_{c'}, q_l\}$
- $\phi^{in} = [\![q_l]\!]_{\geq 1} \wedge [\![q]\!]_{\geq \frac{1}{2}} \wedge [\![q']\!]_{\geq \frac{1}{2}}$
- $\delta(q, f) = \delta(q, p) = \delta(q, c) = true = \delta(q', f') = \delta(q', p') = \delta(q', c')$
  $\forall a \in \Sigma, \delta(q, a) = q,\ \delta(q', a) = q'$
  $\delta(q_f, f) = true = \delta(q_{f'}, f') = \delta(q_c, c) = \delta(q_{c'}, c') = \delta(q_p, p) = \delta(q_{p'}, p')$
  For every $a \in \Sigma, \delta(q_l, a) = *([\![q_c]\!]_{\geq h(f,g,a)}, [\![q_{c'}]\!]_{\geq h(g,f,a)}, [\![q_p]\!]_{\geq \frac{3^k-1}{2}\frac{1-h(f,g,a)-h(g,f,a)}{3^k}},$
  $[\![q_{p'}]\!]_{\geq \frac{3^k-1}{2}\frac{1-h(f,g,a)-h(g,f,a)}{3^k}}, [\![q_l]\!]_{\geq \frac{1-h(f,g,a)-h(g,f,a)}{3^k}}) \vee *([\![q_c]\!]_{\geq h(f,g,a)}, [\![q_{c'}]\!]_{\geq h(g,f,a)},$
  $[\![q_p]\!]_{\geq \frac{3^k-1}{2}\frac{1-h(f,g,a)-h(g,f,a)}{3^k}}, [\![q_{p'}]\!]_{\geq \frac{3^k-1}{2}\frac{1-h(f,g,a)-h(g,f,a)}{3^k}}, [\![q_f]\!]_{\geq \frac{1-h(f,g,a)-h(g,f,a)}{2*3^k}}, [\![q_{f'}]\!]_{\geq \frac{1-h(f,g,a)-h(g,f,a)}{2*3^k}})$.
  Every non notified transition is $false$.

Theorem 16: The p-automaton A recognizes a Markov chain iff there exist $w_1, ...w_n \in \Sigma^+$, $\alpha_1, ..., \alpha_n \in \mathbb{R}^+$ such that $\alpha_1 h(f,g,w_1) + ... + \alpha_n h(f,g,w_n) = \alpha_1 h(g,f,w_1) + ... + \alpha_n h(g,f,w_n)$.

Thus if the previous conjecture is right we get:

Corolary: The emptiness problem for p-automaton is undecidable.

While trying to prove the conjecture we discovered the two following theorems.

Theorem 17: There exist $n \in \mathbb{N}^*, w_1, ...w_n \in \Sigma^+$ and $\alpha_1, ..., \alpha_n \in \mathbb{R}^+$ such that

$$\alpha_1 h(f,g,w_1) + ... + \alpha_n h(f,g,w_n) = \alpha_1 h(g,f,w_1) + ... + \alpha_n h(g,f,w_n)$$

iff there exist $m \in \mathbb{N}^*$ and $w_1, ...w_m \in \Sigma^+$, such that

$$h(f,g,w_1) + ... + h(f,g,w_m) = h(g,f,w_1) + ... + h(g,f,w_m).$$

Quick proof: As the words are finite, $\forall w \in \Sigma^*, h(f,g,w) \in \mathbb{Q}$. Consequently, if there exist n,α_i and w_i such that $\alpha_1 h(f,g,w_1) + ... + \alpha_n h(f,g,w_n) = \alpha_1 h(g,f,w_1) + ... + \alpha_n h(g,f,w_n)$, we could choose such α_i in $\mathbb{Q}$. By multiplying both sides by the denominator, the α_i could be in $\mathbb{N}$. We could thus emulate α_i by repeating w_i.

Theorem 18: If the alphabet Σ (used by f and g) is possibly infinite, then for every $n \in \mathbb{N}$, the following question is undecidable: does there exist $m \leq n, w_1, ...w_m \in \Sigma^+$, such that $h(f, g, w_1) + ... + h(f, g, w_m) = h(g, f, w_1) + ... + h(g, f, w_m)$? .

Quick proof: Let Σ a finite alphabet. Let $f, g : \Sigma^* \to \{1, 2\}^*$ be two morphisms. The words on Σ are numerable. We give them an order: $\Sigma^* = \{w_1, w_2, ...\}$, we will call the alphabet where the w_i are the letters Σ'. We create two morphisms f' and g' from Σ' to $\{1, 2\}^*$ such that $f'(w_i) = 1^{max(|f(w_1)|,|g(w_1)|)} 1^n 1^{max(|f(w_2)|,|g(w_2)|)} 1^n ... 1^{max(|f(w_{i-1})|,|g(w_{i-1})|)} 1^n f(w_i)$.
There exist words $m \leq n$ and $w_1, ..., w_m \in \Sigma'$ such that $h(f', g', w_1) + ... + h(f', g', w_m) = h(g', f', w_1) + ... + h(g', f', w_m)$ iff there exists one word $w \in \Sigma^+$ such that $f(w) = g(w)$.

Conclusion

We made the first algorithm dealing with the emptiness of a qualitative p-automaton. This algorithm is likely to be in EXPTIME, thus the problem would be EXPTIME-complete. This algorithm can also be used to decide the satisfiability problem of a qualitative PCTL formula. But the question for a general p-automaton stays open. That is not surprising as the satisfiability problem for PCTL formulae is a long-standing open problem. There is still a lot to do on this one and some optimization can still be done in the qualitative case.

This internship not only allowed me to confirm my interest in game and automata theory but also to see the importance of writing clearly the proofs. As we aim to write an article around the emptiness problem for p-automaton when the bounded and finite case will be dealt with, we had to make sure that our proofs were good obviously but also easy to understand. Thus I had to redo my drafts again and again. And I am still redoing them now.

References

[1] M. Huth, N. Piterman, D. Wagner *p-Automata: New Foundations for Discrete-Time Probabilistic Verification*, In 7th International Conference on Quantitative Evaluation of SysTems, © IEEE press, 2010.

[2] N. Piterman *p-Automata and Obligation Games*, In 18th International Symposium on Temporal Representation and Reasoning, IEEE, 2011.

[3] S. Safra *On the complexity of ω-automata*, in Proc. 29th IEEE Symp. on Foundations of Computer Science, pages 319-327, 1988.

[4] C. Löding, *Methods for the Transformation of Omega-Automata: Complexity and Connection to Second Order Logic*, Diplomarbeit, Christian-Albrechts-Universität zu Kiel, 1998

[5] S. Miyano, T. Hayashi *Alternating Finite Automata on omega-Words* Theor. Comput. Sci. 32: 321-330, 1984

[6] K. Chatterjee, T. Henzinger, N. Piterman *Algorithms for Buchi Games* Games for Design and Verification (GDV), 2006.

[7] T. Brázdil, V. Forejt, J. Kretínský, A. Kučera *The Satisfiability Problem for Probabilistic CTL* in Proceedings of 23rd Annual IEEE Symposium on Logic in Computer Science (LICS 2008), pages 391-402, IEEE Computer Society, 2008.

Appendix 1: Determinisation on infinite words.

Theorem 1: Let $M = (\Sigma, S, I, \rho, F)$ be a non-deterministic Büchi automaton, there exists a deterministic Rabin automaton $M' = (\Sigma, S', s_0, \rho', F')$ such that $\mathcal{L}(M) = \mathcal{L}(M')$.

In order to get rid of the non-determinism, we have to follow all the runs of the automaton (which already implies an exponential blowup of the number of states). But we also want to see clearly which run visits an accepting state, which is a "good" thing, and which run cannot be followed anymore and gets erased, which is a "bad" thing. We keep this information in memory in the form of a tree; a node gets a son when it goes through an accepting state.

Definition: A memory tree over a set of states S is a tree whose nodes have names in $\{1, ..., |S|\}$ and labelled in $(2^S \setminus \emptyset) \times \{0; 1\}$, if (u,n) is a label of a node and (v,m) the label of one of its children then v $\subsetneq$ u and two brothers have disjoint labels.

Theorem: Those trees are well defined; they can not have more than $|S|$ nodes.

Proof: We proceed by induction on $n = |S|$.
If n=1, the only possible memory tree is the root labelled by the only state of S.
If the theorem is true for $n \in \mathbb{N}$, let T be a memory tree on S where $|S| = n + 1$. Let s be the root of T and $(s_1, ..., s_k)$ be his children. The subtrees T_i of T whose roots are s_i have disjoint labels. Thus we can create sets of states $S_1, ..., S_k$ such that $\forall i \neq j, S_i \cap S_j = \emptyset$, $S_i \subsetneq S$ and T_i is a memory tree over S_i. By induction, as $|S_i| \leq n$, the subtree T_i has at most $|S_i|$ nodes. As the label of s is a strict superset of the label of its children, $|S_1| + |S_2| + ... + |S_k| \leq n$. Thus T has at most n+1 nodes.

Corollary: There is a finite number of memory trees over a set of states S.

Theorem: Let $M = (\Sigma, S, I, \rho, F)$ be a non-deterministic Büchi automaton, there exists a deterministic Rabin automaton $M' = (\Sigma, S', s_0, \rho', F')$ such that $\mathcal{L}(M) = \mathcal{L}(M')$.

Proof:
We construct an equivalent deterministic Rabin automaton $M' = (\Sigma, S', s_0, \rho', F')$ where :

- S' is the set of memory trees over S. We assume that $S = \{1, ..., n\}$.
- s_0 is the memory tree with only the root labelled by (I,0).
- We realize the transition ρ' at a state t for an input a following these steps :
  1. Replace the label (u, k) of every node by$(\bigcup_{q \in u} \rho(q, a), 0)$.
  2. For every node with label (u, k) where $u \cap F \neq \emptyset$, we create a new child whose name has to be higher than all existing brothers and label is $u \cap F$. We can temporarily give a name that's over n.
  3. If s and t are brothers with label (u, k) and (v,m) and s is the older one (its name is smaller), we replace v by $v - u$. We also remove the states in u from the labels of the descendants of t.
  4. Remove the nodes with empty label.
  5. If a node s with label (u, k) verifies u is equal to the union of the first term of the labels of its children, remove its descendants and replace k by 1.

6. Change the name of the nodes so as to leave no gaps while conserving the order (this way a node's name can only decrease).

- $F = \{(E_1, F_1), ..., (E_n, F_n)\}$ where E_i is the set of memory trees without node i and F_i is the set of memory tree where the label (u, k) of the node i verifies $k = 1$.

The transition function gives a memory tree. Indeed, step 3 assures that brothers have disjoint labels, step 4 assures that every label is in $2^S \setminus \emptyset$, step 5 assures that a father is a proper superset of its children and step 6 puts their names in $\{1, ..., n\}$.

Moreover, the language of M and M' are the same.
$\mathcal{L}(M) \subseteq \mathcal{L}(M')$: let σ be an accepting run of M on a word w. The root always exists as he always contains at least the state of the run σ. If the label (u, k) of the root verifies infinitely often $k = 1$, the run of M' on w is accepting. If not, there exists a time t_0 after which $k \neq 1$. As σ visits infinitely many accepting states, the root will have at least one child after t_0, let us say at a time t_1. From this moment onwards, the root will always have at least one child. Indeed, the older child can not be removed except through the application of 5 which would assign k to 1. The states followed by the run σ can move with the application of 3 to an older brother only. It will thus end in a node that will never be removed. By repeating this process, if we do not find a node with label (v,m) such that v is infinitely often equal to 1, we get a tree with infinite depth. The depth of the tree is bounded as there is a finite number of memory trees. Therefore we reach such a node and the run is accepting.

$\mathcal{L}(M') \subseteq \mathcal{L}(M)$: Let σ' be an accepting run of M' on a word w. There is in σ' a node v with label (u(t),k(t)) at time t that has infinitely often $k(t) = 1$ and is finitely often removed. Thus there exists a time t_0 such that v always exists after t_0 and $k(t_0) = 1$. As v is labelled by the iteration of application of ρ and the deletion of some of the elements created, for every state $s \in u(t_0)$, there exists a run σ on M such that $\sigma(t_0) = s$. Let t_1 be a time posterior to t_0 where $k = 1$. For every state $t \in u(t_1)$ there exists a run σ such that $\sigma(t_0) \in u(t_0), \sigma(t_1) = t$ and σ visited F between t_0 and t_1 because $k(t_1) = 1 = k(t_0)$.
Let $t_0, t_1, ...$ be the time where $k = 1$. By iterating the process, for every $n \in \mathbb{N}$ and state $s \in t_n$ we can construct a run σ such that $\sigma(t_n) = s$ and σ visited F at least n times. By taking the limit, we can create a run visiting F infinitely often; $w \in \mathcal{L}(M)$.

Appendix 2: Equivalence between alternating and non-deterministic automata on trees

$\mathcal{L}$(non-deterministic) $\subseteq$ $\mathcal{L}$(alternating)

Theorem: For every non-deterministic Büchi automaton on D-trees $\mathcal{A}$ there exists an alternating Büchi automaton on D-trees $\mathcal{A}'$ such that L($\mathcal{A}$)= L($\mathcal{A}'$).

Proof:
Let $\mathcal{A} = (\Sigma, D, S, S^0, \rho, F)$ be a non-deterministic Büchi automaton on D-trees.
We construct an equivalent alternating Büchi automaton $\mathcal{A}' = (\Sigma, D, S \cup \{s_0\}, s_0, \rho', F')$ where

- $F' = F$ if $S^0 \cap F = \emptyset$, otherwise $F' = F \cup \{s_0\}$.
- for $s \in S, a \in \Sigma$ of arity k, $\rho(s, a, k) = \bigcup_{j=1...n} \{(s_1^j, ...s_k^j)\}$,

we set $\rho'(s,a,k) = \bigvee_{j=1...n} \bigwedge_{i=1...k} (i, s_i^j)$.

and $\rho'(s_0,a,k) = \bigvee_{\{s_1^j,...,s_k^j\}\in\rho(s,a), s\in S^0} \bigwedge_{i=1...k} (i, s_i^j)$.

$\mathcal{A}'$ is an alternating automata on D-trees and $\mathcal{L}(\mathcal{A}') = \mathcal{L}(\mathcal{A})$. This construction works for finite and infinite trees. Indeed, except for the initial state, the run trees are identical. The tree ε is accepted in $\mathcal{A}$ iff it is accepted in $\mathcal{A}'$ (we added s_0 to F if it was necessary).

$\mathcal{L}(alternating) \subseteq \mathcal{L}(non-deterministic)$

Finite acceptance condition

Theorem: For every alternating automaton on finite D-trees $\mathcal{A}$ there exists a non-deterministic automaton on D-trees $\mathcal{A}'$ such that L($\mathcal{A}$)= L($\mathcal{A}'$).

Proof:
Let $\mathcal{A} = (\Sigma, D, S, s_0, \rho, F)$ be an alternating automaton.
We construct an equivalent non-deterministic Büchi automaton $\mathcal{A}' = (\Sigma, D, S', \{\{s_0\}\}, \rho', F')$ such that

- $S' = 2^S$.
- $F' = 2^F$.
- For every $Q \in S', a \in \Sigma$ and arity $k \in D$,
  $\rho'(Q,a,k) = \{(Q_1,...,Q_k) | \exists Y \subseteq D \times S, Y \models \bigwedge_{q\in Q} \rho(q,a,k)$ and $Q_i = \{s | (d_i, s) \in Y\}\}$.

$\mathcal{L}(\mathcal{A}') = \mathcal{L}(\mathcal{A})$.
Indeed : let (T,V) be a $\Sigma-$labelled D-tree and (T_f, f) an accepting run tree on (T,V) of $\mathcal{A}$.
Let (T,r) the labelled D-tree over S' such that :

- $r(\varepsilon) = \{s_0\}$
- Let x a node of T. There exists $Y = \{(c_1, s_1^1), ..., (c_1, s_1^{m_1}), ..., (c_n, s_n^{m_n})\}$ such that $\forall s \in r(x), Y \models \rho(s, V(x))$. And $\forall i = 1...n, r(x.i) = \{s \in S | (c_i, s) \in Y\}$.

This labelled D-tree over S' is a possible run of (T,V) in $\mathcal{A}'$.

Conversely, we can convert a run of $\mathcal{A}'$ to a run of $\mathcal{A}$:
Let (T,V) be a $\Sigma-$labelled D-tree and (T, r) an accepting run tree on (T,V) of $\mathcal{A}'$.
Let (T_f, f) be the $T \times S-$labelled D-tree such that :

- $f(\varepsilon) = (\varepsilon, s_0)$
- Let x a node of T, f(x) = (b,a). There exist $S_1, ..., S_n$ such that $\rho(a, V(b)) = (S_1, ..., S_n)$. And $D_i = \{(i,s) | s \in S_i\}$. $D = \bigcup_{i=1...n} D_i = ((c_1, s_1), (c_2, s_2), ..., (c_k, s_k))$. Finally, $\forall i = 1...k, f(x.i) = (b.c_i, s_i)$.

This labelled D-tree over $T \times S$ is a possible run of (T,V) in $\mathcal{A}$.

This run is accepting in $\mathcal{A}$ iff it is accepted in $\mathcal{A}' : \forall x \in T, r(x) = \{s \in S | (x,s) \in T_f\}$. Thus every leaf is accepting in (T,r) iff every leaf is accepting in (T_f, f).

Büchi acceptance condition

Theorem: For every alternating Büchi automaton on D-trees $\mathcal{A}$ there exists a non-deterministic Büchi automaton on D-trees $\mathcal{A}'$ such that L($\mathcal{A}$)= L($\mathcal{A}'$).

Proof:
Every branch must either stop by reading true or visit infinitely many accepting state. But nothing ensures that they visit the accepting state at the same time. We change the automaton so that it remembers which branches recently went through an accepting state and when every one went through an accepting state of the previous construction, they visit an accepting state of the new automaton. If every branch visits an infinity of previously accepting state, the new branches will visit an infinite number of now accepting state.
Let $\mathcal{A} = (\Sigma, D, S, s_0, \rho, F)$ be an alternating Büchi automaton.
Let $\mathcal{A}' = (\Sigma, D, S', (\{s_0\}, \emptyset), \rho', F')$ such that

- $S' = 2^S \times 2^S$
- $F' = \{\emptyset\} \times 2^S$
- For $U \neq \emptyset, V \subseteq S$, $\rho'((U,V),a,k) = \{(U',V')|$ there exist $X_i, Y_i \subseteq S$ such that $(X_1, ..., X_k) \in \rho'(U,a,k)$ and $(Y_1,...,Y_k) \in \rho'(V,a,k)$ with the ρ' defined in the finite case. And $\exists i, U' = X_i \cap \overline{F}, V' = Y_i \cup (X \cap F)\} \cap \overline{U_i'}$.
  $\rho'((\emptyset, V), a, k) = \{(U',V')|$ there exist $Y_i \subseteq S$ such that $(Y_1, ..., Y_k) \in \rho'(V,a,k)$ with the ρ' defined in the finite case. And $U' = Y_i \cap \overline{F}, V' = Y_i \cap F \cap \overline{U_i'}\}$.

Remark: As we only use the state (U,V) such that $U \cap V = \emptyset$. We can reduce S to only have $\sum_{k=0}^{n} \binom{n}{k} 2^{n-k} = (2+1)^k = 3^k$ states.

Then $\mathcal{L}(\mathcal{A}') = \mathcal{L}(\mathcal{A})$:
Let (T,V) a labelled D-tree over Σ. (T_f, f) the run tree of (T,V) in $\mathcal{A}$.
Let (T,r) the labelled D-tree over S' such that :

- $r(\varepsilon) = (\{s_0\}, \emptyset)$
- Let x be a node of T, r(x) = (P,Q). There exists $X = \{(c_1, s_1^1), ..., (c_1, s_1^{m_1}), ..., (c_n, s_n^{m_n})\}$ such that
  $\forall s \in P, X \models \rho(s, V(x))$, $Y = \{(c_1', s_1'^1), ..., (c_1', s_1'^{m_1'}), ..., (c_n', s_n'^{m_n'})\}$ such that
  $\forall s \in Q, Y \models \rho(s, V(x))$.
  If $P = \emptyset$, then $\forall i = 1...n, r(x.i) = (\{s \in S - F | (c_i, s) \in Y\}, \{s \in S \cap F | (c_i, s) \in Y\})$.
  If $P \neq \emptyset$, then $\forall i = 1...n, r(x.i) = (\{s \in S - F | (c_i, s) \in X\}, \{s \in S \cap F | (c_i, s) \in X\} \cup \{s \in S | (c_i, s) \in Y\})$.

This labelled D-tree over S' is a possible run of (T,V) in $\mathcal{A}'$.

Conversely, we can convert a run in $\mathcal{A}'$ in a run in $\mathcal{A}$:
Let (T,V) a labelled D-tree over Σ. (T,r) the run tree of (T,V) in $\mathcal{A}'$.
Let (T_f, f) the labelled D-tree over $T \times S$ such that :

- $f(\varepsilon) = (\varepsilon, s_0)$
- Let x a node of T, f(x) = (b,a). There exists $S_1, ..., S_n, T_1, ..., T_n$ such that $\rho(a, V(b)) = ((S_1, T_1), ..., (S_n, T_n))$. And $D_i = \{(i,s) | s \in S_i \cup T_i\}$. $D = \bigcup_{i=1...n} D_i = ((c_1, s_1), (c_2, s_2), ..., (c_k, s_k))$. Finally, $\forall i = 1...k, f(x.i) = (b.c_i, s_i)$.

This labelled D-tree over $T \times S$ is a possible run of (T,V) in $\mathcal{A}$.

This run is accepting in $\mathcal{A}$ iff it is accepted in $\mathcal{A}'$.
Indeed, let (T,V) a labelled D-tree over Σ acceted by $\mathcal{A}$. Let P be an infinite branch of (T,V). As $\mathcal{A}$ accepts (T,V), for every infinite branch Q in (T_f, f) of projection P on T, there exist infinitely many $x \in Q$ such that $r(x) \in F$. Let (T,r) the run on $\mathcal{A}'$. Suppose that there exists i_0 such that for every $i \geq i_0$, $r(x_i) = (U_i, V_i)$ où x_i is the ith node of the branch and $U_i \neq$. Thus there exists $s \in U_{i+1}$ whose parent is in U_i for every $i \geq i_0$. Those states represent an infinite branch of (T_f, f) that must be accepting. Thus it is impossible. U_i is infinitely often empty.
Conversely, let (T,V) a labelled D-tree over Σ acceted by $\mathcal{A}'$. Let P be an infinite branch of (T_f, f). This branch corresponds to an infinite branch Q in (T,V) that is accepted in $\mathcal{A}$. Moreover, for every $x \in P$, f(x) = (a,s) and $s \in r(a)$. As there are an infinity of those s that are in an accepting state of $\mathcal{A}$, this branch is accepted in $\mathcal{A}'$.

Appendix 3: emptiness of alternating Büchi tree automata

Theorem 5: The language of the automaton A is not empty iff player 0 has a winning strategy starting from the state $((s_0), ())$.

Intuitively, this game lets player 1 choose which branch of the tree to follow while player 0 chooses the letter of the tree and take care of the disjunction in the formulae. A conjunction is not a choice of player 1: we follow all conjunctions simultaneously.

Player 1 choices are limited to a number i or a boolean. For a play π, we will denote by w_n^π what was chosen by player 1 up until its n'th choice in a game (i.e. : $w_n^\pi = b \in \{true, false\}$ if the last choice of player 1 was b and $w_n^\pi = \epsilon i_1 ... i_n$ if player 1 chose n numbers $i_1, ..., i_n$). We can remark that if w_i^π is not a boolean, it directs us to the i'th node of the tree that is explored or to the associated set of nodes in the tree representing the run of the alternating automata.

If (T_r, r) is an accepting run of (T,V), then there exists a strategy f for player 0 such that for every play π conform to f, if n is the number of times player 1 played, then the actual position is either

- $true$,

- $((\phi_1, ..., \phi_k), (\phi_{k+1}, ..., \phi_{k'}))$ where $\forall i, \phi_i$ is one of the states of the nodes represented by w_n^π in (T_r, r)
- or $((\phi_1, ..., \phi_k), (\phi_{k+1}, ..., \phi_{k'}))$ which verifies : let $H = \{q_1, ..., q_m\}$ be the label of the children of w_n^π in (T_r, r), for every $i = 1...k', H \models \phi_i$.

In the initial position (q_0 is the state of $\varepsilon = w_0^\pi$), player 0 can choose the letter $a = V(\epsilon) \in \Sigma$ and goes to $((\rho(s_0, a, arity(\epsilon))), ())$. This formula is satisfied by the sons of ϵ because (T_r, r) is an accepting run.
If the current position is true, then we cannot move from it.
If the current position represents the states of w_n^π in (T_r, r): $(q_1, ..., q_{k'})$, then we can only go to positions of the form $((\rho(q_1, a, d), ..., \rho(q_k, a, d)), (\rho(q_{k+1}, a, d), ..., \rho(q_{k'}, a, d)))$ where a is a letter of arity d. As (T_r, r) is an accepted run and w_n^π represents the set of states associated with one particular node of T called x, there exists $a \in \Sigma$ such that all the formulae created are satisfied by the sons of w_n^π: $a = V(x)$.
In the last case, if the position belongs to player 0, either all the formulae are *true* then we go to *true* (there cannot be any *false*), or one formula has the form $\phi_i = \bigvee_{n=1...m} \psi_n$, as ϕ_i is satisfied, there exists j such that ψ_j is satisfied and player 0 can choose to go in this state or one formula has the form $\phi_i = \bigwedge_{n=1...m} \psi_n$ and as ϕ_i is satisfied, every ψ_j is satisfied, we can replace ϕ_i by all its components.
If the position belongs to player 1, then player 1 will go to *true* or choose a direction i and keep all the states that are in this direction. As the states are satisfied by the children of w_n^π, they are all children of w_n^π. Keeping the children in the direction i means keeping states represented in (T_r, r) by w_{n+1}^π.

This strategy is winning because for one play π conform to this strategy, if we do not end in *true*, then we visit the branch $P = \{w_i^\pi | i \in \mathbb{N}\}$ of T. Every branch associated to P in T_r visits F infinitely often. Thus supposing that the play stop going to states of the form $((), (s_1, ..., s_k))$ (for $s_i \in S$) means there is a path that stays on the first part and do not go to any accepting state anymore which is not possible.

If player 0 has a winning strategy f from the state $((s_0), ())$, then we create an accepting run (T_r, r) where :

- the initial state is labelled (ε, s_0).
- for a state labelled by (x, q) where x is associated to $\varepsilon i_1...i_k$ in T, let π be a play conform to f such that either there exists n < k where $w_n^\pi = \varepsilon i_1...i_n$ and $w_n^\pi = true$ or $w_k^\pi = \varepsilon i_1...i_k$. In the first case we do not need to add children to this node, in the second we choose for children in the direction i the states that appears in the location reached next if player 1's choice is i.

This run is accepting because if a branch is not acepting, then there would be a play conform to f where this branch is followed among others and after a time, the state in this branch would never meet an accepting state and thus stay in the left part of the position of the play making this play a winning play for player 1.

Appendix 4: a complex emptiness game

Theorem 10: The language of A is not empty iff player 0 has a winning strategy starting from a location $(O, \emptyset, c, \emptyset)$ where $O \models \phi^{in}$.

If the language of the p-automaton A is not empty then we can find a winning strategy for player 0.

Let $M = (S, P, L, s^{in})$ be a Markov chain accepted by A. Let val(s,q) denote the value of the configuration (s,q) in the game $G_{A\times M}$ showing that A accepts M. Let g be a winning strategy of player 0 in $G_{A\times M}$.
During one play, we associate every location visited in G_A with a set H of configurations in $G_{A\times M}$ and a path Γ in the Markov chain. The initial location is $(O, \emptyset, c, \emptyset)$ such that $O \models \phi^{in}$, for every $[\![q]\!]_{\geq 1} \in O, val(s^{in}, q) = 1$ and for every $[\![q]\!]_{>0} \in O, val(s^{in}, q) > 0$. Such O exist as M is accepted by A. We associate this location with the set $H = \{(s^{in}, t) | t \in O\}$ and with the path $\Gamma = s^{in} s_1 s_2 ...$ such that every play in $G_{A\times M}$ conforming with g, which starts in locations of H and which takes the probabilistic transitions that follow the path Γ, is winning.
Such a Γ exists. Indeed, there is at most one >0 obligation $[\![q]\!]_{>0}$, the measure of the set H' of paths starting in s^{in} and winning for q is $\alpha = val(s^{in}, q) > 0$ (if q does not exist, then $\alpha = 1$). For every ≥ 1 obligation q' we remove from H' the paths that are not winning for q'. The set of removed paths has measure 0 as q' wins on a set of measure 1. So we remove a finite number of set of paths from H', every one of which has measure 0, thus we obtain a set H'' which has measure α too. As this set has strictly positive measure, it contains at least one path Γ. This path satisfies every obligation as it is in H''.

For a location (O, Z, k, G), H the associated set of configurations and Γ the associated path, player 0 has to choose a letter a. Let s be the first state of the path Γ. We choose the label of s for the letter a. We associate the location reached with the same set of configuration and the same path.

For a location (O, Z, p_1, a), Γ the associated path and H the associated set of configurations, player 1 will choose which >0 obligation he wants to follow and thus choose a new location (O', Z', k', a) with $k' \in \{c, n\}$. If $k' = c$, then the new associated path Γ' is $s_1 s_2 ...$ with $\Gamma = s_0 s_1 s_2 ....$ If $k' = n$, then there exists one path $\Gamma'' = s_0 s'_1 s'_2 ...$ satisfying the new >0 obligations and all the ≥ 1 obligation, we associate the new location with $\Gamma' = s'_1 s'_2 ....$ We also associate the new location with the same set of configurations where s' is the first state of Γ'.

For a location (O, Z, k, a) with $k \in \{c, n\}$, H the associated set of configurations and Γ the associated path, player 0 has to choose a choice G feasible for a and apply this choice. For every configuration $(s, q) \in H$, as M is accepting, g is a winning strategy from this location. The next step of the game can be $(s', \delta(q, L(s)))$ for s' the second state of Γ. From this, the application of the winning strategy will lead to locations (s', t') where $t' = q'$ or $t' = [\![q']\!]_{\bowtie}$ for some $q' \in Q$. For every such location (s', t'), (q, t') must be in G. Moreover every element of G has to be required by a configuration of $G_{A\times M}$.
Such choice G is feasible. Indeed for $q \in Q$ either $G(q) = \emptyset$ or $(s, q) \in H$. In the second case, $G(q) \models \delta(q, L(s))$ as $G(q)$ contains (q, t') for every t' reached by the winning strategy g in the game $G_{A\times M}$, if it did not satisfy $\delta(q, L(s))$, then the strategy would be losing on the path Γ. We associate the location reached with the same path Γ and the set of configurations $H' = \{(s', q) | q \in O' \cup Z'\}$ where s' is the first state of Γ.

This method creates a strategy f that is winning for player 0. Indeed, let π be a play:

- Every unbounded run of π is fair. Indeed, if one is not, then the path in the game $G_{A\times M}$ corresponding to the run does not satisfy the winning condition of the p-automaton as it does not visit α infinitely often.
- The existential run of π (if there is one) is fair. If there is one, we followed a path Γ that is fair for the existential obligation in $G_{A\times M}$. Thus this obligation visits α infinitely often.
- Every universal run of this play is fair or has $k = n$ infinitely often. If we follow a >0 obligation, then as the set of paths on which this obligation is winning has strictly positive measure, we showed that we could choose to follow a path Γ so that the universal and the existential runs are fair in $G_{A\times M}$. Thus they visit α infinitely often. If there is no >0 obligation at all, then we also follow a path Γ where all the universal runs are winning in $G_{A\times M}$, thus visiting α infinitely often. The last possibility is that we change of >0 obligation infinitely often. We do not stay on one path, we have to change infinitely often so we may not visit α infinitely often, but it also implies that $k = n$ infinitely often as $k = n$ everytime we change of >0 obligation.

In the other direction, let f: $(V_0(V_1V_0V_0)^* \to V_1) \cup (V_0(V_1V_0V_0)^*V_1V_0 \to V_0)$ be a winning strategy for player 0, we can construct a Markov chain that is accepted by A.
Let $M = (S, P, L, s^{in})$ be the Markov chain where:

- $S = V_0V_1(V_0V_0V_1)^*$
- For $s = wu_1 \in S$, every state $s' \in S$ such that $P(s, s') > 0$ verifies $s' = su_0u_0'u_1'$ with $u_0, u_0' \in V_0$, $u_1' \in V_1$, $(u_1, u_0) \in E$, $u_0' = f(su_0)$ and $u_1' = f(su_0u_0')$.
  Let $s_1, ..., s_n$ be the states such that $P(s, s_i) > 0$,
  let $s_i = s(O', Z', k_i, a)(O_i, Z_i, k_i, G_i)(O_i, Z_i, 2, a_i)$ and let $h \in \mathbb{N}$ such that $s \in V_0V_1(V_0V_0V_1)^h$. There exists an integer j such that $k_j = c$. If $n = 1$, then $P(s, s_1) = 1$. If not, then $P(s, s_j) = 1 - \frac{1}{4^{h+1}}$ and for every $m \neq j$, $P(s, s_m) = \frac{1}{4^{h+1}(n-1)}$. We say that the transition from s to s_m for $m \neq j$ corresponds to a new >0 obligation and that the transition from s to s_j corresponds to a continuation of the >0 obligation.
- For $w(O, Z, p_1, a) \in S, L(w(O, Z, p_1, a)) = a$.
- The initial state is $s^{in} = v_o f(v_0)$ where v_0 is an initial location winning with f in G_A.

This choice gives a zero probability to every path that corresponds to new >0 obligations infinitely often and a strictly positive probability to the others.
Indeed, let Γ be a path in the Markov chain. If it changes infinitely often to a new >0 obligation, then considering the construction above, every time $k = n$ the path took an edge that has a probability inferior or equal to $\frac{1}{4}$, thus the probability of this path is 0. If it changes to a new >0 obligation finitely often, then let s be one state of the path after which every state is annoted by $k = c$. There is a probability α to get to this state, $\alpha > 0$ as the prefix of this path is finite, then we take edges with probability $p_1, p_2, ...$ where $\forall i, p_i \geq 1 - \frac{1}{4^i}$, the path thus has a probability at least $\alpha \prod_{k=1}^{\infty}(1 - \frac{1}{4^k}) > 0$.

M is accepted by A. Indeed, we create the strategy g in $G_{A\times M}$ that maintains the invariant: for every configuration (s, t) reached where $t \in Q \cup [\![Q]\!]$ we have $s = w.(O, Z, p_1, a)$ and $t \in O \cup Z$.

We start in (s^{in}, ϕ^{in}), by assumption, ϕ^{in} is a conjunction over $[\![Q]\!]$ so player 1 chooses a location (s, t). Moreover as $s^{in} = (O, Z, c, \emptyset)(O, Z, p_1, a)$ is the initial state of the Markov chain, $(O, Z, c, \emptyset)$ is the initial location of the game G_A, thus $t \in O \cup Z$.
From a location (s, t) where $s = w(O, Z, p_1, a)$ and $t = [\![q]\!]_{\bowtie p}$ or $t = q$ we reach a location $(s', \delta(q, L(s)))$ with a probabilistic transition.
If $t = [\![q]\!]_{>0}$, then the state $s'' = s(O', Z', n, a)(O'', Z'', n, G')(O'', Z'', p_1, a')$ where $Z' = q$ satisfies $Z'' = G'(q) \models \delta(q, L(s))$ thus we can make g such that if we reach a location (s'', t') with $t' \in Q \cup [\![Q]\!]$, then $t' \in Z''$. If $s' \neq s''$, player 0 forfeits.
If $t = [\![q]\!]_{\geq 1}$, then let $s' = su_0(O', Z', k', G')(O', Z', p_1, a')$, $O' \models \delta(q, L(s))$ thus we can choose g such that if we reach a location (s', t') with $t' \in Q \cup [\![Q]\!]$, then $t' \in O'$. It is not possible to have $O' \not\models \delta(q, L(s))$ as q appears as a $\geq$1 obligation in O or Z, and thus has to be satisfied in every next location.
If $t = q$ then if t is in O then we deal with this state as if $t = [\![q]\!]_{\geq 1}$. If t is in Z, then the state $s'' = su_0(O', Z', n, a)(O'', Z'', n, G')(O'', Z'', p_1, a')$ where $q \in Z'$ satisfies $Z'' \models \delta(q, L(s))$ thus we can make g such that if we reach a location (s'', t') with $t' \in Q \cup [\![Q]\!]$, then $t \in Z''$. If $s' \neq s''$, player 0 forfeits.

This strategy is winning for player 0. Indeed, let π be a play where player 0 did not forfeit, the locations in $S \times (Q \cup [\![Q]\!])$ visited are $(s^{in}, t_0), (s_1, t_1), \ldots$. Let π' be the play in G_A such that for every $i \in \mathbb{N}$, s_i is a prefix of π'. As the choices made in the creation of the Markov chain are conform to f, π' is a play conform to f, thus is winning. The run $t_0 t_1 t_2 \ldots$ is a possible run in π'.
If this run is bounded or existential, then it visits α infinitely often, which implies that π is winning.
If this run is universal, then it might not visit α infinitely often, it can also be winning by having $k = n$ infinitely often. This is not an issue as the set of paths H that visit states of M with $k = n$ infinitely often has measure 0, allowing universal obligation to be satisfied with probability 1. Indeed, let $\epsilon = \prod_{i=1}^{\infty}(1 - \frac{1}{4^i}) > 0$, we suppose that the set of loosing paths has probability $\alpha > 0$, there exists $n \in \mathbb{N}$ such that $(1-\epsilon)^n < \alpha$. Let H^i be the set of paths that verify $k = 1$ for at least i states, those sets are clearly supersets of H, moreover those sets have probability inferior to $(1-\epsilon)^i$:
H^0 includes all paths and thus has probability 1.
Whenever one path Γ of H^i takes a transition from a state s to s' where the location reached in s' isthe $i+1$'th location marked by $k = 1$ of the play, then s has a successor s" that is not marked by $k = 1$. From s to s" and onwards, there is a probability at least ϵ to follow a new path Γ'' that does not visit $k = 1$ anymore. Indeed, the path that always goes to the state of M where $k = 0$ has probability at least ϵ from s. Thus at least a measure ϵ of the path of H^i are not in H^{i+1}. As the measure of H^i is known by induction to be less than $(1-\epsilon)^i$, the measure of H^{i+1} is at most $(1-\epsilon)^{i+1}$.
By induction, H^n has measure at most $(1-\epsilon)^n$ but as H^n is a superset of H, H has measure at most $(1-\epsilon)^n < \alpha$ which is impossible. Thus H has measure 0.

In other words, the only plays that are not winning are forfeited plays and plays that follow a $\geq$1 obligation and have measure 0. The forfeited plays follow a $>$0 obligation only, so the measure of plays satisfying a $\geq$1 obligation is 1. Moreover, for every $>$0 obligation there is one winning play, the one that only visits states with $k = 0$ after raising the obligation. This path has strictly positive probability.
Thus every obligation met (in non forfeited plays) are satisfied; g is a winning strategy for player 0.

Appendix 5: Translation to a parity game

Theorem 11: Let π be an infinite play, the word associated to π is accepted by B_b iff there is a bounded run in π that is winning for player 1.

Proof:
Let $\pi = (O_0, Z_0, c, \emptyset), (O_0, Z_0, p_1, a_0), (O'_0, Z'_0, k_1, a_0), (O_1, Z_1, k_1, G_1), (O_1, Z_1, p_1, a_1)(O'_1, Z'_1, k_1, a_1)...$ be an infinite play with a bounded run winning for player 1. We create the word $w = \pi$. This word is accepted by B_b:
let $r = t_0t_1t_2...$ be the bounded run winning for player 1. Let $\forall n, t_n = q_n \in Q$ or $t_n = [\![q_n]\!]_{\bowtie p}$.
There exists a run on w that follows r: as $t_0 \in O_0$, by reading $(O_0, Z_0, c, \emptyset)$ in s_0, we can get to (q_0, b). From a state (q_n, m), we can get to (q_{n+1}, m') with $m' = b$ iff $t_{n+1} \neq q_{n+1}$ by reading $(O_{n+1}, Z_{n+1}, k_{n+1}, G_{n+1})$ because $(q_n, t_{n+1}) \in G_{n+1}$.
As r is winning for player 1, he only visits α finitely often thus the run that follows r visits $\alpha \times \{b\}$ finitely often. Moreover, we make infinitely many bound so we visit $Q \times \{b\}$ infinitely often and even $Q \setminus \alpha \times \{b\}$ infinitely often. Thus the run following r is an accepting run of B_b on w, w is accepted by B_b.

In the other direction, let $w = u_0v_0u'_0u_1v_1u'_1...$ be an infinite word accepted by the automaton such that $\pi = u_0v_0u'_0u_1v_1u'_1...$ is a play of G_A. Then π is winning for player 1. Indeed, as w is accepted, there is an accepting run $r = (t_0, m_0)(t_1, m_1), ...$ on w. $\forall i, (q_i, t_{i+1}) \in G_i$ thus $t_0t_1t_2...$ is a possible run in π. Moreover r visits $Q \setminus \alpha \times \{b\}$ infinitely often and $\alpha \times \{b, v\}$ finitely often. Thus $t_0t_1t_2...$ is bounded and visits α finitely often. Thus the run $t_0t_1t_2...$ is a bounded run winning for player 1.

Theorem 12: Let π be an infinite play, the word associated to π is accepted by B_u iff there is an universal run in π that is winning for player 1.

Proof:
Let $\pi = (O_0, Z_0, c, \emptyset), (O_0, Z_0, p_1, a_0), (O'_0, Z'_0, k_1, a_0), (O_1, Z_1, k_1, G_1), (O_1, Z_1, p_1, a_1)(O'_1, Z'_1, k_1, a_1)...$ be an infinite play with an universal run winning for player 1. We create the word $w = \pi$. This word is accepted by B_u:
let $r = t_0t_1t_2...$ be the universal run winning for player 1. Let $\forall n, t_n = q_n \in Q$ or $t_n = [\![q_n]\!]_{\bowtie p}$.
As this run is universal and winning for player 1, there exist i and j such that $t_i = [\![q_i]\!]_{\geq 1}$, $\forall n > i, t_n \in Q$ and if $n > j$, then $t_n \notin \alpha$ and $k_n = c$.
There exists a run on w that follows r: as $t_0 \in O_0$, by reading $(O_0, Z_0, c, \emptyset)$ in s_0, we can get to (q_0, b) (or if $i = 0$ to (q_0, c)). From a state (q_n, b), if $n < (i-1)$, we can get to (q_{n+1}, b) by reading $(O_{n+1}, Z_{n+1}, k_{n+1}, G_{n+1})$ because $(q_n, t_{n+1}) \in G_{n+1}$. If $n = i-1$, then by reading (O_i, Z_i, k_i, G_i), we can go to (q_i, c). From a state (q_n, m), $m \in \{c, n\}$, for $n \geq i$, we can get to (q_{n+1}, m') with $m' = c$ iff $k = c$ then by reading $(O_{n+1}, Z_{n+1}, k_{n+1}, G_{n+1})$ because $(q_n, q_{n+1}) \in G_{n+1}$.
After i, we leave definitively $Q \times \{b\}$ and after j we will not go to $\alpha \times \{c\}$ or $Q \times \{n\}$ anymore. Thus the run following r is an accepting run of B_u on w, w is accepted by B_u.

In the other direction, let $w = u_0v_0u'_0u_1v_1u'_1...$ be an infinite word accepted by the automaton such that $\pi = u_0v_0u'_0u_1v_1u'_1...$ is a play of G_A. Then π is winning for player 1. Indeed, as w is accepted, there is an accepting run $v = (t_0, m_0)(t_1, m_1), ...$ on w. $\forall i, (q_i, t_{i+1}) \in G_i$ thus $r = t_0t_1t_2...$ is a possible run in π. Moreover, r is an universal run as v leaves $Q \times \{b\}$ by reading a $\geq$1 obligation. As v visits $\alpha \times \{c\}$ and $Q \times \{n\}$ finitely often, r visits α and $k = n$ finitely often. Thus the run $t_0 t_1 t_2 ...$ is an universal run winning for player 1.

Theorem 13: Let π be an infinite play, the word associated to π is accepted by B_e iff there is an existential run in π that is winning for player 1.

Proof:
Let $\pi = (O_0, Z_0, c, \emptyset), (O_0, Z_0, p_1, a_0), (O'_0, Z'_0, k_1, a_0), (O_1, Z_1, k_1, G_1), (O_1, Z_1, p_1, a_1)(O'_1, Z'_1, k_1, a_1)...$ be an infinite play with an existential run winning for player 1. We create the word $w = \pi$. This word is accepted by B_e:
let $r = t_0 t_1 t_2 ...$ be the existential run winning for player 1. Let $\forall n, t_n = q_n \in Q$ or $t_n = [\![q_n]\!]_{\bowtie p}$. As this run is existential and winning for player 1, there exist i and j such that $t_i = [\![q_i]\!]_{>0}$, $\forall n > i, t_n \in Q$ and if $n > j$, then $t_n \notin \alpha$.
There exists a run on w that follows r: as $t_0 \in O_0$, by reading $(O_0, Z_0, c, \emptyset)$ in s_0, we can get to (q_0, b) (or if $i = 0$ to (q_0, e)). From a state (q_n, b), if $n < (i-1)$, we can get to (q_{n+1}, b) by reading $(O_{n+1}, Z_{n+1}, k_{n+1}, G_{n+1})$ because $(q_n, t_{n+1}) \in G_{n+1}$. If $n = i-1$, then by reading (O_i, Z_i, k_i, G_i), we can go to (q_i, e). From a state (q_n, e), for $n \geq i$, we can get to (q_{n+1}, e) by reading $(O_{n+1}, Z_{n+1}, k_{n+1}, G_{n+1})$ because $(q_n, q_{n+1}) \in G_{n+1}$.
After i, we leave definitively $Q \times \{b\}$ and after j we will not go to $\alpha \times \{e\}$ anymore. Thus the run following r is an accepting run of B_e on w, w is accepted by B_e.

In the other direction, let $w = u_0 v_0 u'_0 u_1 v_1 u'_1 ...$ be an infinite word accepted by the automaton such that $\pi = u_0 v_0 u'_0 u_1 v_1 u'_1 ...$ is a play of G_A. Then π is winning for player 1. Indeed, as w is accepted, there is an accepting run $v = (t_0, m_0)(t_1, m_1), ...$ on w. $\forall i, (q_i, t_{i+1}) \in G_i$ thus $r = t_0 t_1 t_2 ...$ is a possible run in π. Moreover, r is an existential run as v leaves $Q \times \{b\}$ by reading a $>$0 obligation. As v visits $\alpha \times \{e\}$ finitely often, r visits α finitely often. Thus the run $t_0 t_1 t_2 ...$ is an existential run winning for player 1.

Theorem 14: Let π be an infinite play, the word associated to π is accepted by C iff this play is winning for player 0.

Proof: Let π be an infinite play, if it is winning for player 1 then there exists a run in π winning for player 1. As this run is either bounded, existential or universal, it will lead to the acceptance of π by B_b, B_e or B_u. Thus π will not be accepted by C.

In the other direction, let w a word associated to a play π that is not accepted by C, then w is accepted by B_b, B_e or B_u. Thus player 1 has a winning run in π, π is winning for player 1.

Let D be the determinisation of C. We construct the parity game $G_A \times D$ where there is a transition from (g, q) to (g', q') iff there is a transition from g to g' in G_A and q' is reached from q by reading g in D. A play $(g_1, q_1)(g_2, q_2)...$ is winning iff $q_1 q_2 ...$ is winning in D.

Theorem 15: The language of A is not empty iff player 0 has a winning strategy in $G_A \times D$ starting from a location $((O, \emptyset, c, \emptyset), s_0)$ where $O \models \phi^{in}$.

Proof: Let $(O, \emptyset, c, \emptyset)$ be a location of G_A where $O \models \phi^{in}$. If there exists a winning strategy g for player 0 starting in this location, then there is a winning strategy for player 0 in $G_A \times D$ starting from the location $((O, \emptyset, c, \emptyset), s_0)$.
Indeed, let $\pi = u_0 u_1 ...$ be a play in G_A starting in $(O, \emptyset, c, \emptyset)$ and conform with g, π is winning as g is a winning strategy. Let $\pi' = (u_0, s_0)(u_1, s_1)...$ be a play in $G_A \times D$. π' is winning: $u_0 u_1 ...$ is winning thus there is a winning run reading π in C (theorem 7) which implies that the run reading $u_0 u_1 ...$ in D is winning. As $s_0 s_1 ...$ is this run, it is winning. As a consequence, the strategy in $G_A \times D$ that can go from (u_n, s_n) to (u_{n+1}, s_{n+1}) after a play $(u_0, s_0)(u_1, s_1)...(u_n, s_n)$ iff there is a play conform with g in G_A starting by $u_0 u_1 ... u_n u_{n+1}$ is well defined (as the transition of $G_A \times D$ are the same as the transitions of G_A) and winning for player 0.

In the other direction, let $((O, \emptyset, c, \emptyset), s_0)$ be a location of $G_A \times D$ where $O \models \phi^{in}$. If there exists g a winning strategy for player 0 starting in this location then there is a winning strategy for player 0 in G_A starting from a location $(O, \emptyset, c, \emptyset)$.
Indeed, let $\pi = (u_0, s_0)(u_1, s_1)...$ be a play in $G_A \times D$ starting in $((O, \emptyset, c, \emptyset), s_0)$ and conform with g, π is winning as g is a winning strategy. Let $\pi' = u_0 u_1 ...$, it is a play in $G_A \times D$. π' is winning : $s_0 s_1 ...$ is winning in D and as D is the determinisation of C, every run in π' are winning for player 0 (theorem 7). As a consequence, the strategy in G_A that can go from u_n to u_{n+1} after a play $u_0 u_1 ... u_n$ iff there is a play conform with g in $G_A \times D$ starting by $(u_0, s_0)(u_1, s_1)...(u_n, s_n)(u_{n+1}, s_{n+1})$ is well defined (as the transition of $G_A \times D$ are the same as the transitions of G_A) and winning for player 0.