MPRI 2006–2007
Jeux pour la théorie des automates, la vérification et l'internet

Olivier Serre
serre@liafa.jussieu.fr
`www.liafa.jussieu.fr/~serre`

5 décembre 2006

Table des matières

1 Définitions de base **3**
- 1.1 Graphe, graphe de jeu, jeu 3
- 1.2 Conditions de gain 4
- 1.3 Stratégies et positions gagnantes 6
  - 1.3.1 Positions gagnantes 7
  - 1.3.2 Quelques stratégies particulières 7

2 Jeux sur des graphes finis **9**
- 2.1 Jeux d'accessibilité 9
- 2.2 Jeux de Büchi 10
- 2.3 Jeux de parité 11
  - 2.3.1 Définitions de base 11
  - 2.3.2 Preuve du Théorème 2.3 12
- 2.4 Jeux de Muller 13
  - 2.4.1 Les stratégies gagnantes peuvent requérir de la mémoire 14
  - 2.4.2 Résolution des jeux de Muller : le last appearance record 14
- 2.5 Jeux munis de conditions (externe) régulières 16

3 Jeux sur des graphes de processus à pile **17**
- 3.1 Définitions 17
  - 3.1.1 Graphe associé à un processus à pile 17
  - 3.1.2 Graphe de jeu et jeux associés 18
- 3.2 Résolution des jeux d'accessibilité 21
  - 3.2.1 Factorisation des parties 22
  - 3.2.2 Simulation 22
  - 3.2.3 La réduction 26
  - 3.2.4 Conclusion 31
- 3.3 Résolution des jeux de Parité 32
- 3.4 Résolution des jeux d'explosion 32
  - 3.4.1 Stratégies gagnantes 32
  - 3.4.2 Réduction vers un jeu de sûreté 34
  - 3.4.3 Complexité 36
- 3.5 Ensembles de positions gagnantes 36
  - 3.5.1 $\mathcal{P}$-automates 36
  - 3.5.2 Deux propriétés sur les conditions de gain 37

3.5.3 Jeux conditionnés et ensembles de retour 40
3.5.4 Cas particulier des conditions invariantes par translations 41
3.5.5 Régularité des ensembles de positions gagnantes 44

4 Jeux, logique et automates d'arbres **46**
4.1 μ-calcul 46
4.1.1 Syntaxe 47
4.1.2 Sémantique 47
4.1.3 Exemples de formules 48
4.2 Des jeux au μ-calcul 49
4.3 Automates d'arbres et jeux de parité 50
4.3.1 Définitions 50
4.3.2 Exemples d'automates 52
4.4 Du μ-calcul aux automates d'arbre alternants/aux jeux 52
4.5 Retour sur le Théorème 4.1 55
4.6 Conclusion 55

Chapitre 1

Définitions de base

1.1 Graphe, graphe de jeu, jeu

Définition 1.1 *Un* ***graphe*** *est un couple $G = (V, E)$ où V est un ensemble de sommets et E un ensemble d'arc. Si le graphe est non étiqueté, E est un sous-ensemble de $V \times V$. Dans le cas étiqueté, E est un sous-ensemble de $V \times A \times V$.*

Un graphe est fini ssi V est fini.

La représentation graphique d'un graphe est donné dans l'exemple suivant.

Exemple 1.1 La Figure 1.1 donne la représentation graphique du graphe $\{a, b\}$-étiqueté $G = (V, E)$ où :

- $V = \{1, 2, 3, 4\}$.
- $E = \{(1, a, 2), (1, b, 2), (1, a, 4), (2, a, 1), (2, b, 2), (4, a, 2)\}$.

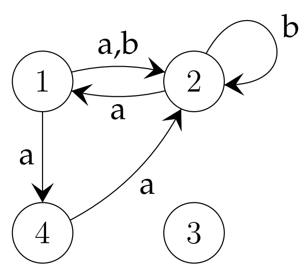


FIG. 1.1 – Représentation graphique d'un graphe

Les deux protagoniste d'un jeu, à savoir les joueurs, seront la plupart du temps appelé Eve et Adam (Alternativement 0 et 1, Éloïse et Abelard, joueur existentiel et joueur universel...).

On se fixe un graphe $G = (V, E)$.

On considère une partition des sommets $V = V_{\mathbf{E}} \cup V_{\mathbf{A}}$ entre les deux joueurs : les sommets (positions) dans $V_{\mathbf{E}}$ sont celles (contrôlés par) d'Eve tandis que les sommets dans $V_{\mathbf{A}}$ appartiennent à Adam.

Un triplet $\mathcal{G} = (G, V_{\mathbf{E}}, V_{\mathbf{A}})$ est appelé un **graphe de jeu** (ou une **arène**). La représentation graphique est la même que pour un graphe à la différence près que les sommets d'Eve sont représentés par des cercles et ceux d'Adam par des carrés. La Figure 1.2 donne la représentation du graphe de jeu $\mathcal{G} = (G, V_{\mathbf{E}}, V_{\mathbf{A}})$ où $V_{\mathbf{E}} = \{2, 4, 7, 8\}$ et $V_{\mathbf{A}} = \{1, 3, 5, 6\}$.

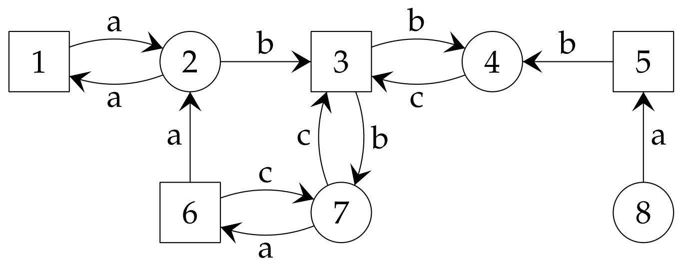


FIG. 1.2 – Exemple de graphe de jeu

1.2 Conditions de gain

Une **condition de gain** sur $\mathcal{G}$ est un sous-ensemble Ω de E^ω (ou de façon équivalente de V^ω si le graphe n'est pas étiqueté).

Enfin, un **jeu à deux joueurs** sur un graphe de jeu $\mathcal{G}$ est un couple $\mathbb{G} = (\mathcal{G}, \Omega)$, où $\mathcal{G}$ est un graphe de jeu et Ω une condition de gain sur $\mathcal{G}$.

Une **partie** dans un jeu $\mathbb{G} = (\mathcal{G}, \Omega)$ débutant en v_0 est définie comme suit. Le joueur qui contrôle le sommet initial v_0 choisit, si c'est possible, un arc $e_1 = (v_0, a_0, v_1)$ d'origine v_0. Si un tel arc n'existe pas, la partie se termine. Sinon le joueur qui contrôle v_1 choisit l'arc suivant $e_2 = (v_1, a_1, v_2)$ s'il en existe un, et ainsi de suite. Une partie est donc un chemin maximal (fini ou infini) dans G : $(v_0, a_0, v_1)(v_1, a_1, v_2) \cdots$. Par **partie partielle** on désignera le préfixe d'une partie, c'est à dire un chemin fini dans G. Autant que peut se faire, on réservera les lettres Λ et λ pour les parties (partielles).

Dans le graphe de jeu donné dans la figure 1.2,
$(1, a, 2)(2, b, 3)(3, b, 7)(7, a, 6)(6, a, 2)((2, a, 1)(1, a, 2))^\omega$ est une partie.

Eve **remporte** une partie infinie ssi elle appartient à Ω, sinon c'est Adam qui gagne. Dans le cas d'une partie finie (qui termine donc dans un cul-de-sac), le joueur qui est bloqué perd la partie.

On va considérer principalement deux types de conditions de gain : les conditions de gains internes et les conditions de gain externes.

Définition 1.2 *Une **condition de gain interne** sur un graphe de jeu $\mathcal{G}$ de sommets V est un sous-ensemble Ω of V^ω. Une partie $\Lambda = (v_0, a_0, v_1)(v_1, a_1, v_2) \cdots$ est remportée par Eve ssi $v_0 v_1 v_2 \cdots$ appartient à Ω.*

Définition 1.3 *Une **condition de gain externe** sur un graphe de jeu A-étiqueté $\mathcal{G}$est un sous-ensemble Ω de A^ω. Une partie $\Lambda = (v_0, a_0, v_1)(v_1, a_1, v_2) \cdots$ est remportée par Eve ssi $a_0 a_1 a_2 \cdots$ appartient à Ω.*

Etant donnée une condition de gain Ω, on considère la **condition duale** $\overline{\Omega}$ définie comme $\overline{\Omega} = E^\omega \setminus \Omega$. Il est clair que la condition duale d'une condition interne est une condition interne, et que la condition duale d'une condition externe est une condition externe.

Donnons maintenant quelques exemples de conditions de gains. On commence par les conditions d'accessibilité et de sûreté

Définition 1.4 *Soit un sous-ensemble $F \subseteq V$ de sommets, appelés **sommets finaux**. La condition interne V^*FV^∞ est une condition dite d'**accessibilité**. Ainsi, Eve remporte une partie si un état final est visité au cours de celle-ci.*

*La condition duale $(V \setminus F)^\omega$ d'une condition d'accessibilité est une condition dite de **sûreté**. On parle alors pour F de sommets **interdits**. Ainsi, Eve remporte une partie si on ne visite pas d'état interdit au cours de celle-ci.*

Pour les condition de gains suivantes, on adopte la convention qu'une partie finie est perdue par le joueur qui ne peut bouger. On définit alors les conditions de Büchi, de co-Büchi, de parité et de Muller.

Définition 1.5 *Soit un sous-ensemble $F \subseteq V$ de sommets, appelés **sommets finaux**. La condition interne $(V^*F)^\omega$ est une condition dite de **Büchi**. Ainsi, Eve remporte une partie si on visite infiniment souvent des états finaux au cours de celle-ci.*

*La condition duale $\bigcup_{i \geq 0}(V^*F)^i(V \setminus F)^\omega$ d'une condition de Büchi est une condition dite de **co-Büchi**. On parle alors pour F de sommets **interdits**. Ainsi, Eve remporte une partie si on ne visite qu'un nombre fini d'états interdits au cours de celle-ci.*

Définition 1.6 *Soit un sous-ensemble fini $C \subseteq \mathbb{N}$ d'entiers positifs appelés **couleurs** et soit une application col de V dans C, appelée **fonction de coloriage**. La condition interne $\{v_0v_1v_2\cdots \in V^\omega \mid \liminf(col(v_i))_{i\geq 0}$ est paire$\}$ est une condition dite de **parité**. Ainsi, Eve remporte une partie si la plus petite couleur infiniment souvent visitée est paire.*

*La condition duale $\{v_0v_1v_2\cdots \in V^\omega \mid \liminf(col(v_i))_{i\geq 0}$ est impaire$\}$ est une condition dite de **co-parité**. Ainsi, Eve remporte une partie si la plus petite couleur infiniment souvent visitée est impaire.*

Remarque 1.1 Considérons une condition Ω de co-parité sur un graphe de jeu coloré par une fonction de coloriage col. On considère alors la fonction de coloriage col' définie par $col'(v) = col(v) + 1$. Il est facile de voir que la condition de parité Ω' sur le graphe de jeu précédent coloré par la fonction col', est telle que $\Omega = \Omega'$.

Définition 1.7 *Soit un sous-ensemble fini $C \subseteq \mathbb{N}$ d'entiers positifs appelé **couleurs**, et soit une application col de V dans C, appelée **fonction de coloriage**. Soit une collection $\mathcal{F}$ de sous-ensembles de C, appelés **ensembles finaux**. La condition interne $\{v_0v_1v_2\cdots \in V^\omega \mid \{c \mid \exists^\infty i,\ col(v_i) = c\} \in \mathcal{F}\}$ est une condition dite de **Muller**. Ainsi, Eve remporte une partie si l'ensemble des couleurs infiniment souvent visitées est dans $\mathcal{F}$.*

La condition duale d'une condition de Muller d'ensembles finaux $\mathcal{F}$ est également une condition de Muller avec la même fonction de coloriage et avec $\mathcal{P}(C) \setminus \mathcal{F}$ pour ensembles finaux.

Donnons enfin un exemple de condition externe.

Définition 1.8 *Considérons un langage ω-régulier L sur un alphabet d'étiquetage A. Soit un graphe de jeu A-étiqueté. Le langage $\Omega = L$ est une condition dite **ω-régulière**. La condition duale d'une condition ω-régulière est également ω-régulière, par clôture par complémentation des langages réguliers.*

1.3 Stratégies et positions gagnantes

Dans toute ce paragraphe, on se donne un graphe de jeu $\mathcal{G} = (G, V_{\mathbf{E}}, V_{\mathbf{A}})$, où $G = (V, E)$.

Stratégies

Qui dit jeu dit envie de gagner, et donc stratégie. Une stratégie consiste à considérer les coups déjà joués ainsi que la position courante pour décider quel sommet atteindre. Plus formellement, une **stratégie** pour Eve est une fonction $\varphi : E^* \to E$ telle que, pour toute partie partielle λ se terminant en $v \in V_{\mathbf{E}}$, $\varphi(\lambda)$ a pour origine v. De même, on définit les stratégies pour Adam comme des fonctions $\psi : E^* \to E$ telles que, pour toute partie partielle λ se terminant en $v \in V_{\mathbf{A}}$, $\psi(\lambda)$ a pour origine v.

Il existe d'autres définitions plus générales pour les stratégies. En particulier, on peut les définir non plus comme des fonctions à valeur dans E, mais comme des applications à valeur dans $\mathcal{P}(E)$. Une **stratégie non déterministe** pour Eve est une application $\varphi : E^* \to \mathcal{P}(E)$ telle que, pour toute partie partielle λ se terminant en $v \in V_{\mathbf{E}}$, les éléments de $\varphi(\lambda)$ ont pour origine v. De même, on définit les stratégies pour Adam comme des applications $\psi : E^* \to E$ telles que, pour toute partie partielle λ se terminant en $v \in V_{\mathbf{A}}$, les éléments de $\psi(\lambda)$ ont pour origine v.

Les stratégies non déterministes généralisent bien les stratégies déterministes. En effet, étant donnée une stratégie déterministe φ, on définit un stratégie φ' équivalente en posant $\varphi'(\lambda) = \emptyset$ pour tout λ si φ n'est pas définie en λ, et $\varphi'(\lambda) = \{\varphi(\lambda)\}$ sinon. Pour la suite des définitions, on se place dans le cadre des stratégies non déterministes.

Etant donnée une stratégie φ pour Eve, cette dernière peut la **respecter** en choisissant toujours de prendre un arc donné par φ sur le préfixe de partie déjà joué. Plus formellement, on dira qu'Eve *respecte* φ au cours d'une partie $\lambda = e_1 e_2 \cdots$ si, pour tout $0 \leq i < |\lambda|$, $e_{i+1} \in \varphi(e_1 \cdots e_i)$ dès lors que e_i termine par un sommet de $V_{\mathbf{E}}$. De même, on définit le fait qu'Adam respecte une stratégie ψ.

Les stratégies intéressantes sont celles que l'on peut suivre tout au long d'une partie. Plus précisément, on dira qu'une stratégie φ pour Eve est **praticable** depuis un sommet v si, pour toute partie partielle λ d'origine v où Eve respecte φ, et se terminant par un arc d'extrémité dans $V_{\mathbf{E}}$, $\varphi(\lambda)$ est non vide. De même, on définit les stratégies praticables pour Adam.

Remarque 1.2 Evoquons maintenant le cas particulier des graphes non étiquetés. Nous avions noté que l'on peut représenter les chemins par des mots sur l'alphabet des sommets. Dans ce formalisme, une stratégie pour Eve est alors une application φ de V^∞ dans $\mathcal{P}(V)$ telle que, pour toute partie partielle λ se terminant dans un sommet $v \in V_{\mathbf{E}}$ et pour tout $v' \in \varphi(\lambda)$, $(v, v') \in E$. De façon symétrique, on définit les stratégies pour Adam.

Etant donnée une stratégie φ, Eve respecte φ lors d'une partie $\lambda = v_1 v_2 v_3 \cdots$ si, pour tout $1 \leq i < |\lambda|$, $v_i \in V_{\mathbf{E}} \Rightarrow v_{i+1} \in \varphi(v_1 v_2 \cdots v_i)$.

Considérons maintenant un jeu $\mathbb{G} = (\mathcal{G}, \Omega)$ sur $\mathcal{G}$, une position initiale v_0 et une stratégie φ pour Eve. On dira que φ est une **stratégie gagnante pour Eve dans $\mathbb{G}$ depuis** v_0 si toute partie au départ de v_0 où Eve respecte φ, est remportée par Eve. On définit de même les stratégies gagnantes pour Adam. Le jeu $\mathbb{G}$ est **déterminé** si, pour toute position initiale, l'un des deux joueurs possède une stratégie gagnante depuis cette position dans $\mathbb{G}$.

Remarque 1.3 Etant donné un jeu, l'un des deux joueurs au plus a une stratégie gagnante. En effet, si les deux joueurs possédaient une stratégie gagnante, la partie obtenue quand ces derniers respectent leurs stratégies gagnantes respectives serait à la fois dans et hors de Ω.

Tous les jeux considérés dans ce cours seront déterminés.

Remarque 1.4 Etant donné une stratégie φ gagnante pour Eve depuis un sommet v dans un jeu $\mathbb{G}$, il est facile de voir qu'Eve possède une stratégie gagnante déterministe depuis v. En effet, il suffit de considérer la stratégie φ' définie pour tout $\lambda \in E^\infty$ tel que $\varphi(\lambda)$ est non vide, par $\varphi'(\lambda) = e_\lambda$, où e_λ est un élément quelconque de $\varphi(\lambda)$. On vérifie alors facilement que φ' est gagnante pour Eve.

1.3.1 Positions gagnantes

Une position v dans un jeu $\mathbb{G}$ est une **position gagnante**pour Eve si celle-ci possède une stratégie gagnante dans $\mathbb{G}$ depuis v. L'ensemble des positions gagnantes pour Eve sera généralement noté $W_{\mathbf{E}}$. On définit de même l'ensemble $W_{\mathbf{A}}$ des positions gagnantes pour Adam.

1.3.2 Quelques stratégies particulières

On se fixe maintenant un graphe de jeu $\mathcal{G} = ((V, E), V_{\mathbf{E}}, V_{\mathbf{A}})$, et un jeu $\mathbb{G} = (\mathcal{G}, \Omega)$ sur $\mathcal{G}$. Dans ce paragraphe, nous discutons de quelques types particuliers de stratégies possibles et de leurs représentations respectives.

Evoquons le cas des stratégies dites *sans mémoire* ou *positionnelles*. Une stratégie est sans mémoire si le choix du coup à jouer ne dépend pas du passé de la partie mais uniquement de la position courante.

Définition 1.9 *Une stratégie φ est* ***sans mémoire*** *(ou* ***positionnelle****) si pour toutes parties partielles λ et λ' de même extrémité, $\varphi(\lambda) = \varphi(\lambda')$. Ainsi, les stratégies sans mémoire pour Eve (respectivement pour Adam) sont exactement l'ensemble des applications φ de $V_{\mathbf{E}}$ (respectivement de $V_{\mathbf{A}}$) dans $\mathcal{P}(E)$ telles que pour tout $v \in V$, les éléments de $\varphi(v)$ ont pour origine v. Dans le cas des graphes non étiquetés, ce sont les fonctions de $V_{\mathbf{E}}$ (respectivement de $V_{\mathbf{A}}$) dans $\mathcal{P}(V)$, telles que pour tout $v \in V_{\mathbf{E}}$ et tout $v' \in \varphi(v)$, $(v, v') \in E$.*

Dans le cas où le graphe de jeu est fini, une stratégie sans mémoire est donc représentable de façon finie. Dans le cas où le graphe est infini, ce n'est pas le cas en général, puisque le graphe lui-même peut ne pas être finiment représentable. C'est pourquoi on préférera parler de stratégie sans mémoire pour les jeux sur des graphes finis, et de stratégie positionnelle pour les jeux sur des graphes infinis.

Entre les deux extrêmes que sont les stratégies générales et les stratégies positionnelles, le concept de *stratégies avec mémoire* permet de définir les deux notions précédentes mais aussi de les raffiner.

Définition 1.10 *Soit M un ensemble que l'on appelle* mémoire. *Une* stratégie à mémoire M *pour Eve est un triplet (m_0, φ, up) formé d'un élément $m_0 \in M$ appelé* mémoire initiale, *d'une application φ de $V \times M$ dans $\mathcal{P}(E)$ telle que pour tout couple (v, m), tout élément de $\varphi(v, m)$ est d'origine v, et d'une application up de $M \times E$ dans M appelée* application de mise à jour. *Eve respecte φ au cours d'une partie $\lambda = e_1 e_2, \cdots$ si pour tout $i < |\lambda|$, $e_{i+1} \in \varphi(v_i, m_i)$, où v_i est l'extrémité de e_i si $i \geq 1$ et où l'on pose $m_i = up(m_{i-1}, e_i)$ pour tout $i \geq 1$. En d'autres termes, quand Eve doit jouer, elle regarde la valeur de la mémoire ainsi que le sommet courant pour déterminer quel arc emprunter. Après chaque coup (effectué par Eve ou par Adam), Eve met à jour sa mémoire à l'aide de la fonction up en considérant l'ancienne valeur et le dernier coup joué. La stratégie φ est gagnante si toutes les parties où Eve respecte φ sont gagnantes. On définit également le caractère praticable d'une stratégie à mémoire. Il est alors facile de définir les mêmes notions pour Adam.*

Lorsque M est réduit à un seul élément (et n'est donc pas utile), on retrouve les stratégies sans mémoire (d'où leur nom).

Soit une stratégie φ générale et soit v une position initiale. Considérons la stratégie à mémoire M, (m_0, ψ, up) où l'on pose $M = E^\infty$, $m_0 = \varepsilon$, $\psi(v, m) = \varphi(m)$ et $up(e_1 \cdots e_n, e_{n+1}) = (e_1 \cdots e_n e_{n+1})$. La stratégie ci-dessus stocke donc dans sa mémoire le préfixe de partie déjà joué. Dès lors il est évident que l'ensemble des parties où Eve respecte la stratégie (m_0, ψ, up) est exactement l'ensemble des parties où Eve respecte φ.

Un cas particulièrement intéressant est celui des *stratégies à mémoire finie*, c'est-à-dire des stratégies à mémoire M où M est fini. Lorsque le graphe de jeu est fini, une telle stratégie est réalisée par un transducteur (c'est-à-dire un automate fini avec sortie).

Chapitre 2

Jeux sur des graphes finis

On considère diverses conditions de gain pour des jeux sur des graphes finis et on en donne la résolution.

2.1 Jeux d'accessibilité

On se fixe un graphe non étiqueté $G = (V, E)$ *sans cul-de-sac*, une partition $V_{\mathbf{E}} \cup V_{\mathbf{A}}$ de V qui définit un graphe de jeu $\mathcal{G}$. On se donne un ensemble $F \subseteq V$ de sommets finaux et on note $\mathbb{G}$ le jeu d'accessibilité induit.

On a alors le résultat suivant :

Théorème 2.1 *On peut calculer l'ensemble des positions gagnantes dans $\mathbb{G}$ et construire des stratégies sans mémoire pour les deux joueurs.*

Le reste de la section est dévolu à la preuve de ce résultat. On définit par induction la suite croissante de sous-ensembles de V.

$Attr_0^{\mathbf{E}}(F) = F$ et
$Attr_i^{\mathbf{E}}(F) = Attr_i^{\mathbf{E}}(F) \cup \{v \in V_{\mathbf{E}} \mid \exists v' \in Attr_i^{\mathbf{E}}(F)$ t.q. $(v, v') \in E\}$
$\cup \{v \in V_{\mathbf{A}} \mid (v, v') \in E \Rightarrow v' \in Attr_i^{\mathbf{E}}(F)\}$

Comme la suite précédente et croissante et bornée, elle converge : on note $Attr^{\mathbf{E}}(F)$ sa limite et on va montrer que $Attr^{\mathbf{E}}(F) = W_{\mathbf{E}}$. L'ensemble $Attr^{\mathbf{E}}(F)$ est appelé **Attracteur pour Eve de** F et son complément est qualifié de **piège pour Eve** .

Il est facile de voir que pour tout $v \in Attr^{\mathbf{E}}(F)$, si $v \in V_{\mathbf{E}}$ alors soit $v \in F$ soit v a un successeur v' dans $Attr^{\mathbf{E}}(F)$ tel que $rg(v') < rg(v)$. Si $v \in V_{\mathbf{A}}$ alors soit $v \in F$ soit pour tout successeur v' de v, v' est dans $Attr^{\mathbf{E}}(F)$ et $rg(v') < rg(v)$.

Soit $v \in Attr^{\mathbf{E}}(F)$. On définit $rg(v) = \min\{i \mid v \in Attr_i^{\mathbf{E}}(F)$. On définit enfin une stratégie sans mémoire φ pour Eve en définissant pour tout $v \in Attr^{\mathbf{E}}(F) \cap V_{\mathbf{E}}$, $\varphi(v) = v'$ pour un v' tel que $rg(v') < rg(v)$ (il est facile de voir qu'un tel v' existe bien).

On vérifie que si Eve respecte φ dans une partie débutant dans un sommet de $Attr^{\mathbf{E}}(F)$, alors la partie reste dans $Attr^{\mathbf{E}}(F)$ jusqu'à atteindre F : en effet, soit une telle partie $\lambda = v_0 v_1 \cdots$ alors pour tout i tel que $v_j \notin F$ pour tout $j < i$, on a $rg(v_{i+1}) < rg(v_i)$: en $rg(v_0)$ étapes on arrive dans F. La stratégie φ est donc gagnante depuis tout sommet dans $Attr^{\mathbf{E}}(F)$.

Il reste à prouver que pour tout sommet dans $Attr^{\mathbf{E}}(F)$, Adam a une stratégie gagnante sans mémoire.

Il est facile de voir que pour tout $v \notin Attr^{\mathbf{E}}(F)$, si $v \in V_{\mathbf{A}}$ alors v a un successeur v' qui n'est pas dans $Attr^{\mathbf{E}}(F)$. Si $v \in V_{\mathbf{E}}$ alors pour tout successeur v' de v, v' n'est pas dans $Attr^{\mathbf{E}}(F)$.

On définit une stratégie sans mémoire ψ pour Adam en définissant pour tout $v \notin Attr^{\mathbf{E}}(F) \cap V_{\mathbf{A}}$, $\psi(v) = v'$ pour un v' tel que $v' \notin Attr^{\mathbf{E}}(v)$ (un tel v' existe bien d'après ce qui précède).

On vérifie que si Adam respecte ψ dans une partie débutant dans un sommet pas dans $Attr^{\mathbf{E}}(F)$, alors la partie reste hors de $Attr^{\mathbf{E}}(F)$. En effet, soit une partie $\lambda = v_0 v_1 \cdots$ où Adam respecte ψ débutant dans un $v_0 \notin Attr^{\mathbf{E}}(F)$. Par induction on prouve que $v_i \notin Attr^{\mathbf{E}}(F)$ pour tout i : c'est vrai pour $i = 0$ et ça vient de la définition de ψ pour v_{i+1} si $v_i \in V_{\mathbf{A}}$ et de la remarque initiale si $v_i \in V_{\mathbf{E}}$.

2.2 Jeux de Büchi

On se fixe un graphe non étiqueté $G = (V, E)$ *sans cul-de-sac*, une partition $V_{\mathbf{E}} \cup V_{\mathbf{A}}$ de V qui définit un graphe de jeu $\mathcal{G}$. On se donne un ensemble $F \subseteq V$ de sommets finaux et on note $\mathbb{G}$ le jeu de Büchi induit.

On a alors le résultat suivant :

Théorème 2.2 *On peut calculer l'ensemble des positions gagnantes dans $\mathbb{G}$ et construire des stratégies sans mémoire pour les deux joueurs.*

Tout d'abord on commence par définir une variante de l'attracteur. Soit S un sous-ensemble de sommets. On définit la suite croissante suivante :$Attr_0^{+\mathbf{E}}(S) = S$ et $Attr_{i+1}^{+\mathbf{E}}(S) = Attr_i^{+\mathbf{E}}(S) \cup \{v \in V_{\mathbf{E}} \mid \exists v' \in Attr_i^{+\mathbf{E}}(S) \text{ t.q. } (v, v') \in E\}$
$\cup \{v \in V_{\mathbf{A}} \mid (v, v') \in E \Rightarrow v' \in Attr_i^{+\mathbf{E}}(S)\}$. On note $Attr^{+\mathbf{E}}(S)$ sa limite et on parle alors d'attracteur strict.

Une simple adaptation du résultat sur les jeux d'accessibilité permet de montrer qu'un sommet est dans $Attr^{+\mathbf{E}}(S)$ ssi Eve a une stratégie pour atteindre en au moins un coup S.

On définit maintenant la suite (décroissante) suivante : $Z_0 = V$, $Z_{i+1} = Attr^{+\mathbf{E}}(Z_i) \cap F$. On note Z_∞ sa limite. En particulier $Z_\infty = Attr^{+\mathbf{E}}(Z_\infty)$ et $Z_\infty \subseteq F$.

On va montrer que $W_{\mathbf{E}} = Attr^{\mathbf{E}}(Z_\infty)$. Pour cela, on définit la stratégie sans mémoire φ suivante pour Eve. Si $v \in Z_\infty$ on pose $\varphi(v) = v'$ où v' est donné par une stratégie sans mémoire pour l'attracteur strict vers Z_∞ ; en particulier $v' \in W_{\mathbf{E}}$. Si $v \notin Z_\infty \cap W_{\mathbf{E}}$, on pose $\varphi(v) = v'$ où v' est donné par une stratégie sans mémoire pour l'attracteur vers Z_∞. Il est clair qu'une partie où Eve respecte φ et qui débute dans $W_{\mathbf{E}}$ atteint Z_∞ et ensuite visite infiniment $Z_\infty \subseteq F$ et donc F : la partie est donc gagnante pour Eve.

Considérons maintenant $W_{\mathbf{A}} = V \setminus W_{\mathbf{E}}$ et montrons qu'Adam possède une stratégie gagnante sans mémoire pour $W_{\mathbf{A}}$. Soit $v \in W_{\mathbf{A}} \cap V_{\mathbf{A}}$: on montre facilement qu'il existe i tel que $v \notin Attr(Z_i)$. On considère le plus petit i tel que $v \in Attr(Z_i) \notin Attr(Z_{i+1})$. On considère le successeur de v par une stratégie d'Adam pour éviter Z_{i+1} si $v \notin F$: on pose $\psi(v) = v'$. Si $v \in F$, comme $v \notin Z_\infty$ il existe un successeur $v' \notin Attr^{\mathbf{E}}(Z_\infty)$ (donc

$v' \in W_{\mathbf{A}}$) : on pose $\psi(v) = v'$. Par induction sur i, on montre facilement que ψ est une stratégie gagnante pour Adam sur $W_{\mathbf{A}}$.

2.3 Jeux de parité

On se fixe un graphe non étiqueté $G = (V, E)$ *sans cul-de-sac*, une partition $V_{\mathbf{E}} \cup V_{\mathbf{A}}$ de V qui définit un graphe de jeu $\mathcal{G}$. On se donne un ensemble de couleurs $C = \{0, \ldots, d\}$ et une fonction de coloriage $col : V \to C$. On appelle alors $\mathbb{G}$ le jeu de parité induit.

On a alors le résultat suivant dont la preuve occupe le reste de cette section.

Théorème 2.3 *On peut calculer l'ensemble des positions gagnantes dans $\mathbb{G}$ et construire des stratégies sans mémoire pour les deux joueurs.*

Pour un joueur σ (Eve ou Adam), on désignera par $\overline{\sigma}$ l'autre joueur.

2.3.1 Définitions de base

On commence par quelques définitions de base utiles pour la suite.

Définition 2.1 (Sous-graphe de jeu) *Soit $\mathcal{G} = (G, V_{\mathbf{E}}, V_{\mathbf{A}})$ un graphe de jeu avec $G = (V, E)$, et soit $U \subseteq V$. Soit $col : V \to C$ une fonction de coloriage sur G. On considère le sous-graphe de jeu coloré (en restreignant col à U) de $\mathcal{G}$ induit par U, $\mathcal{G}[U] = (G[U], V_{\mathbf{E}} \cap U, V_{\mathbf{A}} \cap U)$ où $G[U] = (U, E \cap (U \times U))$. On dira que $\mathcal{G}[U]$ est un **sous-graphe de jeu** si et seulement s'il est sans cul-de-sac,* i.e. *pour tout $u \in U$, il exists $u' \in U$ tel que $(u, u') \in E$.*

Fait 2.1 *Un sous-graphe de jeu d'un sous-graphe de jeu est un sous-graphe de jeu.*

Définition 2.2 (Piège) *Soit un joueur σ et soit $\mathcal{G} = (G, V_{\mathbf{E}}, V_{\mathbf{A}})$ un graphe de jeu avec $G = (V, E)$. Un **piège pour** σ (ou σ-piège) dans $\mathcal{G}$ est un sous-ensemble de sommets $U \subseteq V$ tel que*
- *pour tout $v \in U \cap V_\sigma$, $(v, v') \in E \Rightarrow v' \in U$ (σ ne peut sortir de U) ;*
- *pour tout $v \in U \cap V_{\overline{\sigma}}$, il existe $v' \in U$ tel que $(v, v') \in E$ ($\overline{\sigma}$ peut rester dans U) ;*

Fait 2.2 *Le joueur $\overline{\sigma}$ possède une stratégie sans mémoire pour maintenir une partie dans un σ-piège.*

Preuve. Ceci vient du fait que rester dans un piège est la condition duale d'atteindre un sommet hors du piège, qui est une condition d'accessibilité. Le résultat découle alors du fait que les joueurs ont des stratégies sans mémoire pour les jeux d'accessibilité. ■

Fait 2.3 *Soit U un piège pour σ dans un graphe de jeu $\mathcal{G}$. Alors $\mathcal{G}[U]$ est un sous-graphe de jeu.*

Preuve. Immédiat.

Définition 2.3 (Attracteur) *Soit un joueur σ et soit $\mathcal{G} = (G, V_{\mathbf{E}}, V_{\mathbf{A}})$ un graphe de jeu avec $G = (V, E)$. L'**attracteur** de $U \subseteq V$ pour σ (σ-attracteur), noté $Attr^\sigma(\mathcal{G}, U)$ ($\mathcal{G}$ est omis si le contexte est clait), est défini comme la limite de la suite (croissante et bornée) :*
- *$Attr_0^\sigma(U) = U$;*

– $\forall i \geq 0$, $Attr^{\sigma}_{i+1}(U) = Attr^{\sigma}_{i}(U) \cup \{v \in V_{\sigma} \mid \exists v' \in Attr^{\sigma}_{i}(U) \text{ t.q. } (v,v') \in E\} \cup \{v \in V_{\overline{\sigma}} \mid (v,v') \in E \Rightarrow v' \in Attr^{\sigma}_{i}(U)\}$

Fait 2.4 *L'ensemble $Attr^{\sigma}(\mathcal{G}, U)$ est l'ensemble des positions gagnantes pour σ dans le jeu d'accessibilité vers U.*

Le complémentaire d'un attracteur pour σ est un piège pour σ.

L'attracteur pour σ d'un piège pour $\overline{\sigma}$ est un piège pour $\overline{\sigma}$

Preuve. Immédiats ou conséquences directes des résultats sur les jeux d'accessibilité.

2.3.2 Preuve du Théorème 2.3

On peut enfin prouver le Théorème 2.3. On va prouver en plus du résultat annoncé que W_{σ} est un piège pour $\overline{\sigma}$.

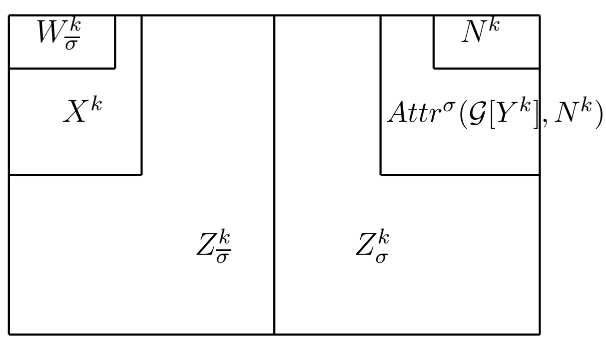
La preuve est faite par induction sur le nombre n de couleurs. Le cas de base est celui où $n = 1$ et le résultat est alors immédiat : selon la parité de l'unique couleur, Eve (paire) ou Adam (impaire) gagne partout et peut appliquer n'importe quelle stratégie (en particulier, une stratégie positionnelle).

On suppose le résultat prouvé pour un certain $n - 1 \geq 1$ et l'on considère le cas où il y a n couleurs. On note pour le reste de la preuve par σ le joueur qui gagne si la plus petite couleur c est infiniment souvent visitée (*i.e.* Eve si σ est paire, Adam sinon).

On construit alors une suite de sous-ensembles de V que l'on note $(W^k_{\overline{\sigma}})$ et une suite de stratégies positionnelles $\varphi^k_{\overline{\sigma}}$ pour $\overline{\sigma}$ telles que :

1. Pour tout k, $W^k_{\overline{\sigma}}$ est un piège pour σ et $\varphi^k_{\overline{\sigma}}$ est une stratégie gagnante pour $\overline{\sigma}$ sur $W^k_{\overline{\sigma}}$.
2. La suite des $W^k_{\overline{\sigma}}$ est croissante, et chaque $\varphi^{k+1}_{\overline{\sigma}}$ étend $\varphi^k_{\overline{\sigma}}$

Pour cela on commence par poser $W^0_{\overline{\sigma}} = \emptyset$ et les propriétés sont alors vérifiées. On suppose maintenant que $W^k_{\overline{\sigma}}$ et $\varphi^k_{\overline{\sigma}}$ sont construits et on définit $W^{k+1}_{\overline{\sigma}}$ et $\varphi^{k+1}_{\overline{\sigma}}$.



On commence par poser $X^k = Attr^{\overline{\sigma}}(\mathcal{G}, W^k_{\overline{\sigma}})$. Il s'agit d'un piège pour σ car c'est l'attracteur d'un piège pour σ (Hypothèse d'induction).

Le complémentaire $Y^k = V \setminus X^k$ de X^k est un piège pour $\overline{\sigma}$ car c'est le complémentaire d'un $\overline{\sigma}$-attracteur. C'est en particulier un sous-graphe de jeu que l'on note alors $\mathcal{G}[Y^k]$.

On regarde alors l'ensemble des sommets de Y^k de couleur c : $N^k = \{v \in Y^k \mid \rho(v) = c\}$ et enfin on retire à Y^k l'attracteur pour σ de N^k dans le sous-graphe de jeu :

$$Z^k = Y_k \setminus Attr^{\sigma}(\mathcal{G}[Y^k], N^k)$$

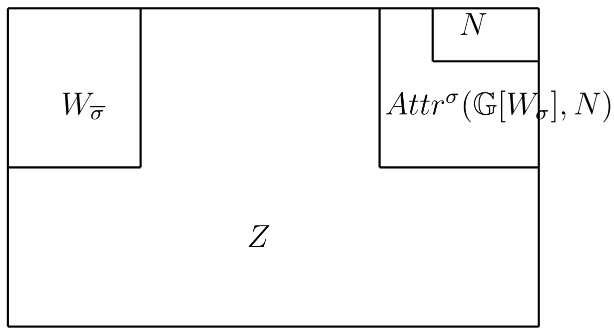
L'ensemble Z^k ainsi obtenu est un piège pour σ (complémentaire d'un σ attracteur) et définit un sous-graphe de jeu (sous-graphe de jeu d'un sous-graphe de jeu). De plus tous les sommets dans Z^k ont une couleur supérieure ou égale à $c+1$.

En applicant l'hypothèse de récurrence, on définit Z^k_σ et $Z^k_{\overline{\sigma}}$. Enfin, on pose $W^{\overline{\sigma}}_{k+1} = X^k \cup Z^k_{\overline{\sigma}}$, et on définit $\varphi^{k+1}_{\overline{\sigma}}$ comme l'union des stratégies (positionnelles sur ces deux ensembles). Cette dernière est clairement gagnante sur $W^{k+1}_{\overline{\sigma}}$ et étend $\varphi^{k+1}_{\overline{\sigma}}$.

Enfin, le fait que $W^{k+1}_{\overline{\sigma}}$ est un piège est immédiat.

Maintenant, on prend pour $W_{\overline{\sigma}}$ la limite de la suite précédente et de même pour $\varphi_{\overline{\sigma}}$. Il est alors clair que les points (1) et (2) pour $\overline{\sigma}$ sont vérifiés ici.

On pose alors $W_\sigma = V \setminus W_{\overline{\sigma}}$. Le fait que W_σ soit un piège pour $\overline{\sigma}$ vient du fait qu'il est le complémentaire d'un $\overline{\sigma}$-attracteur car $W_{\overline{\sigma}} = Attr^{\overline{\sigma}}(W_{\overline{\sigma}})$. Il reste donc à prouver que σ y possède une stratégie gagnante positionnelle.



Pour cela, on pose $N = \{v \in W_\sigma \mid \rho(v) = c\}$ et on pose

$$Z = W_\sigma \setminus Attr^\sigma(\mathcal{G}[W_\sigma], N)$$

Par définition de $W_{\overline{\sigma}}$, $\overline{\sigma}$ ne peut gagner sur Z (sinon $W_{\overline{\sigma}}$ ne serait pas la limite de la suite $W^k_{\overline{\sigma}}$). Ainsi σ a une stratégie positionnelle sur Z (HR). On définit alors la stratégie φ_σ de la façon suivante :

– Suivre la stratégie gagnante positionnelle quand on est dans Z.
– Dans $Attr^\sigma(\mathcal{G}[W_\sigma], N)$ suivre la stratégie positionnelle d'attracteur.
– Dans N rester dans l'ensemble gagnant.

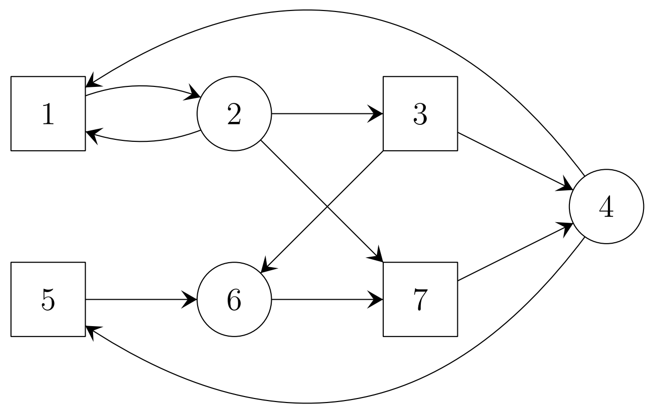
On vérifie facilement que cette stratégie est gagnante et positionnelle.

2.4 Jeux de Muller

On se fixe un graphe non étiqueté $G = (V, E)$ *sans cul-de-sac*, une partition $V_{\mathbf{E}} \cup V_{\mathbf{A}}$ de V qui définit un graphe de jeu $\mathcal{G}$. On se donne un ensemble de sous-ensembles d'états $\mathcal{F} = \{F_1, \cdots, F_i\}$, $F_i \subseteq V$ pour tout i. On appelle alors $\mathbb{G}$ le **jeu de Muller** induit : Eve gagne une partie ssi l'ensemble des sommets visités infiniment souvent au cours de la partie est dans $\mathcal{F}$.

2.4.1 Les stratégies gagnantes peuvent requérir de la mémoire

On considère le graphe de jeu suivant et on prend $\mathcal{F} = \{\{1, 2, 3, 4\}, \{1, 2, 3, 4, 5, 6, 7\}\}$.



Il est facile de vérifier qu'une stratégie sans mémoire pour Eve ne peut être gagnante : depuis 2, Eve doit aller vers 3 (qui doit dans tous les cas être infiniment souvent visité), et si elle choisit d'aller depuis 4 vers 1, Adam peut jouer de sorte à ce que tous les sommets soient visités souvent infiniment sauf 5, et si elle choisit d'aller depuis 4 en 5, Adam peut jouer de sorte à ce que les sommets visités infiniment souvent soient $\{4, 5, 6, 7\}$.

Une stratégie gagnante avec mémoire pour Eve est la suivante :
- Depuis 2 toujours aller vers 3
- Depuis 4 si l'on vient de 3, aller vers 1.
- Depuis 4, si l'on vient depuis 7 via 3 et 6, aller vers 5 puis 6 puis 7 puis 4 puis 1.

En d'autres termes, Eve regarde en 4 quels sont les deux derniers états impairs visités :
- $(1, 3)$ ou $(5, 7)$: aller vers 1.
- $(3, 7)$: aller vers 5

Il est à notre que cette information est facilement constructible par un automate fini et qu'une telle machine peut donc réaliser une stratégie gagnante pour Eve.

On généralise ces idées dans la suite.

2.4.2 Résolution des jeux de Muller : le last appearance record

On identifie dans la suite V avec $\{1, \ldots, n\}$. On appelle **last appearance record (LAR)** une permutation de V avec un entier dans $\{1, \ldots, n\}$ (le **hit**). On représentera cela par un n-uplet dont le j-ème élément est souligné si le hit vaut j, et on note $LAR(V)$ l'ensemble des LAR sur V. Le LAR est défini par induction pour toute suite (finie) de sommets (en particulier toute partie partielle) en posant : $LAR(\varepsilon) = (1, \ldots, \underline{n})$ et en posant si $LAR(\lambda) = (v_1, \cdots, v_n)$, $LAR(\lambda \cdot v) = (v_1, \cdots v_{j-1}, \underline{v_{j+1}}, v_{j+2}, \cdots, v_n, v)$ si $v = v_j$: v est mis au fond et la position qu'il occupait devient le hit. Ainsi $LAR(\lambda \cdot v)$ *ne dépend que* de v et de $LAR(\lambda)$.

Il est facile de voir que l'ensemble des sommets infiniment souvent répétés est F au cours d'une partie **ssi** au bout d'un certain moment, le LAR a toujours un hit supérieur ou égal à $|F|$ et que infiniment souvent le hit vaut $|F|$ et alors les $|F|$ derniers sommets du LAR forment une permutation de F.

On considère un nouveau graphe de jeu dans lequel les sommets vont en plus contenir une information sur le LAR. Plus formellement, on considère le graphe $G' = (V \times LAR(V), E')$ où $((v,\tau),(v',\tau')) \in E$ ssi $(v,v') \in E$ et τ' est le LAR obtenu en allant vers v' avec le LAR τ. On considère la partition de V' donnée par $V'_{\mathbf{E}} = V_{\mathbf{E}} \times LAR(V)$, et on appelle $\mathcal{G}'$ le graphe de jeu associé. Enfin, on définit la condition de gain suivante sur $\mathcal{G}'$: pour tout $i = 1, \cdots n$, on note E_i l'ensemble des sommets de V' avec un hit $< i$ et par F_i l'ensemble des sommets de E_i augmenté des sommets dont le hit vaut i et dont les $n - i + 1$ derniers sommets du LAR (*i.e.* ceux à partir du hit) forment une permutation d'un ensemble présent dans $\mathcal{F}$. On a donc une chaîne $E_1 \subseteq F_1 \subseteq E_2 \subseteq \cdots F_n$. Afin de rendre cette dernière stricte, on fusionne éventuellement E_j et E_{j+1} si $E_j = F_j$, et F_j et F_{j+1} si $E_{j+1} = F_j$. La condition de gain sur $\mathcal{G}'$ est la suivante : Eve gagne si et seulement s'il existe un j tel que E_j est finiment visité et F_j est infiniment souvent visité, et on note $\mathbb{G}'$ le jeu ainsi obtenu. La condition de gain précédente est qualifiée de **condition en Chaîne de Rabin**.

On a alors les deux résultats suivants qui permettent de conclure :

Lemme 2.1 *Un sommet v est gagnant pour Eve dans $\mathbb{G}$ ssi le sommet $(v, LAR(\varepsilon))$ est gagnant pour Eve dans $\mathbb{G}'$*

Preuve. Considérons une partie $\lambda = v_0 v_1 \cdots$ dans $\mathbb{G}$. On considère alors la partie $\lambda' = v'_0 v'_1 \cdots$ dans $\mathbb{G}$ définie par $v'_i = (v_i, LAR(v_0 \cdots v_{i-1})) \cdots$ pour tout i. Il est alors facile de voir que λ est gagnante pour Eve dans $\mathbb{G}$ ssi λ' est gagnante pour Eve dans $\mathbb{G}'$. Ceci vient de la caractérisation donnée des sommets infiniment souvent répétés en terme de LAR, et de la définition des E_i/F_i.

On note τ la fonction (bijection en fait) telle que $\tau(\lambda) = \lambda'$ et π la fonction inverse qui projette λ' en λ, et on considère les versions naturelles de ces fonctions définie sur les parties partielles. Maintenant, si φ est une stratégie d'Eve dans $\mathbb{G}$ on définit la stratégie φ' pour Eve dans $\mathbb{G}'$ en posant $\varphi'(\lambda') = last(\tau(\varphi(\pi(\lambda))))$, où $last$ est la fonction qui associe à une partie son dernier sommets. Il est alors facile de vérifier que φ' est gagnante depuis $(v, LAR(\varepsilon))$ si φ est gagnante depuis v. Comme cette construction peut être également faite pour une stratégie gagnante d'Adam, cela conclut la preuve. ■

Lemme 2.2 *La condition en chaîne de Rabin peut être exprimée comme une condition de parité. Ainsi, les jeux munis d'une condition en chaîne de Rabin admettent des stratégies gagnantes sans mémoire.*

Preuve. On considère le coloriage suivant : $col(v) = 2i - 1$ pour tout $v \in E_i \setminus F_{i-1}$ (avec $F_0 = \emptyset$) et $col(v) = 2i$ pour tout $v \in F_i \setminus E_i$. On conclut alors facilement. ■

On a donc en résumé le théorème suivant :

Théorème 2.4 *On peut calculer l'ensemble des positions gagnantes dans un jeu de Muller ainsi que des stratégies gagnantes qui nécessitent une mémoire de taille $k.k!$ où k est le nombre de sommets du graphe de jeu considéré.*

Preuve. La première partie du résultat est une conséquence directe des deux lemmes précédents. La seconde partie (taille de la mémoire) est laissée en exercice. ■

2.5 Jeux munis de conditions (externe) régulières

On se fixe un graphe **étiqueté** $G = (V, E \subseteq V \times A \times V)$ *sans cul-de-sac*, une partition $V_{\mathbf{E}} \cup V_{\mathbf{A}}$ de V qui définit un graphe de jeu $\mathcal{G}$. On se donne un langage régulier $\Omega \subseteq A^\omega$ que l'on voit comme une condition de gain externe. On appelle alors $\mathbb{G}$ le jeu induit.

Dans ce qui suit, on considérera le cas particulier où Ω peut être décrit par un automate *déterministe complet* de Büchi[1] que l'on note $\mathcal{A} = (Q, A, \delta, q_{in}, F)$. On rappelle ici que Q est un ensemble fini d'états, que $q_{in}Q$ est l'état initial, $F \subseteq Q$ est un ensemble d'états finaux, A l'alphabet d'entrée et $\delta : Q \times A \to Q$ est la fonction de transition. Enfin, on rappelle qu'un mot (infini) est accepté par $\mathcal{A}$ ssi l'unique calcul de $\mathcal{A}$ débutant en q_{in} visite infiniment souvent l'état F.

On a alors le résultat suivant dont la preuve occupe le reste de cette section.

Théorème 2.5 *On peut calculer l'ensemble des positions gagnantes dans $\mathbb{G}$ et construire des stratégies avec mémoire fini pour les deux joueurs.*

On considère pour cela le graphe non étiqueté $G' = (V' = V \times Q, E')$ où l'on définit $E' = \{((v_1, q_1), (v_2, q_2)) \mid \exists a \in A \text{ t.q. } (v_1, a, v_2) \in E \text{ et } q_2 = \delta(q_1, a)\}$. On partitionne V' en $V'_{\mathbf{E}} \cup V'_{\mathbf{A}}$ en posant $V'_{\mathbf{E}} = V_{\mathbf{E}} \times Q$, en on note $F' = V \times F$. Enfin, on considère le jeu de Büchi $\mathbb{G}'$ sur le graphe de jeu $\mathcal{G}'$ induit par la précédente partition et l'ensemble F'.

On a alors le résultat suivant : pour tout sommet $v \in V$, v est gagnant pour Eve dans $\mathbb{G}$ ssi (v, q_{in}) est gagnant pour Eve dans $\mathbb{G}'$. Pour cela, on considère un sommet v tel que (v, q_{in}) soit gagnant pour Eve dans $\mathbb{G}'$ et une stratégie sans mémoire φ' associée. On définit alors une stratégie avec mémoire[2] φ pour Eve dans $\mathbb{G}$ depuis v. La mémoire utilisée par φ' est un élément de Q, et est initialisée au départ à q_{in}. Dans un sommet $v \in V_E$, avec une mémoire q, $\varphi(v, q) = (v, a, v')$ et $up(q, (v, a, v')) = q'$ où l'on pose $\varphi'((v, q)) = (v', q')$, et où l'on prend pour a n'importe quelle lettre telle que $\delta(q, a) = q'$ et $(v, a, v') \in E$ (par définition de E', un tel a existe toujours). Il est alors facile de voir que φ est bien une stratégie à mémoire finie. Maintenant, si l'on considère une partie λ débutant en v où Eve respecte φ, et que l'on note $\lambda = (v_0, a_0, v_1)(v_1, a_1, v_2) \cdots$ et $q_0 q_1 \cdots$ le calcul de $\mathcal{A}$ sur $a_0 a_1 \cdots$ débutant en $q_0 = q_{in}$, on a alors que $(v_0, q_0)(v_1, q_1) \cdots$ est une partie dans $\mathcal{G}'$ où Eve respecte φ'. En particulier cette partie est gagnante, ce qui veut dire qu'il y a une infinité de i tels que $q_i \in F$, ce qui veut dire que $a_0 a_1 \in \Omega$, et donc que λ est remportée par Eve. On en conclut donc que φ est gagnante.

Réciproquement, on construit, de la même façon, une stratégie gagnante à mémoire finie pour Adam dans $\mathbb{G}$ depuis une stratégie gagnante dans $\mathbb{G}'$.

[1] Ceci n'est pas toujours possible, puisqu'il faut en général, pour garder le déterminisme, considérer une condition de parité (ou de façon équivalente de Muller). Pour éviter d'introduire des notions peut être peu familière pour certains, on se concentrera sur le cas d'une condition d'acceptation de Büchi en remarquant que la preuve proposée par la suite se généralise facilement pour le cas général.

[2] Voir définition 1.10

Chapitre 3

Jeux sur des graphes de processus à pile

L'objet de ce chapitre est l'étude de jeux sur des graphes infinis. L'approche étant résolument algorithmique, elle suppose une description finie des entrées, et en partie des graphes considérés. Dès lors, les graphes infinis considérés seront finiment représentables.

Pour cela, nous allons considérer les graphes associés à des automates à pile. La construction est simple : étant donné un automate à pile, les sommets du graphe associé sont les configurations de l'automate à pile, alors que les arcs du graphe correspondent à l'existence d'une transition entre deux configurations dans l'automate à pile.

3.1 Définitions

3.1.1 Graphe associé à un processus à pile

Dans ce cadre, on considère une version édulcorée des automates à pile dans laquelle on supprime les concepts d'états initiaux et finaux. On parle alors de **processus à pile**

Définition 3.1 *Un **processus à pile** $\mathcal{P}$ est un quintuplet $\langle Q, A, \Gamma, \perp, \Delta\rangle$ où Q est un ensemble fini d'états, A est un alphabet fini d'entrée, Γ est un alphabet fini de pile, $\perp$ est un symbole de Γ appelé symbole de fond de pile et Δ est une application de $Q\times\Gamma\times A$ dans $\mathcal{P}(\{skip(q),\ pop(q),\ push(q,\gamma) \mid q \in Q,\ \gamma \in \Gamma\backslash\{\perp\}\})$ appelée fonction de transition et telle que $\delta(p,\perp,a) \subseteq \{push(q,\gamma),\ skip(q) \mid q \in Q,\ \gamma \in \Gamma \setminus \{\perp\}\}$, pour tout état $p \in Q$ et toute lettre $a \in A$.*

*On appelle $Q \times (\Gamma \setminus \{\perp\})^*\perp$ l'ensemble des **configurations** de $\mathcal{P}$, c'est à dire l'ensemble des paires formées d'un état de contrôle et d'un mot de pile. Soit une configuration $(q,\gamma\sigma)$ de l'automate à pile, et soit $a \in A$ une lettre. A la lecture de a, $\mathcal{P}$ peut passer de la configuration $(q,\gamma\sigma)$ aux configurations suivantes :*

- *$(q',\gamma\sigma)$ pour tout $skip(q') \in \Delta(q,\gamma,a)$.*
- *(q',σ) pour tout $pop(q') \in \Delta(q,\gamma,a)$.*
- *$(q',\gamma'\gamma\sigma)$ pour tout $push(q',\gamma') \in \Delta(q,\gamma,a)$.*

On dit alors qu'il y a une transition d'étiquette a de la première configuration C à la seconde C', ce que l'on note $C \xrightarrow{a} C'$.

Etant donné un processus à pile, on lui associe un graphe infini de la façon suivante.

Définition 3.2 *Soit un processus à pile $\mathcal{P} = \langle Q, A, \Gamma, \bot, \Delta \rangle$, et soit V l'ensemble de ses configurations. Soit $\rightarrow$ la relation de transition introduite dans la définition 3.1. On considère la relation $E \subseteq V \times A \times V$ donnée par $E = \{(v, a, v') \mid v \xrightarrow{a} v'\}$. Le graphe A-étiqueté $G = (V, E)$ est le* ***graphe associé au processus à pile*** *$\mathcal{P}$.*

Si l'on souhaite ignorer l'étiquetage sur les arcs, il suffit de partir d'un processus à pile dont l'alphabet d'entrée est réduit à une lettre, qui du coup étiquette tous les arcs du graphe, et qui peut alors être ignorée. Dans ce cadre, on omettra l'alphabet d'entrée dans la définition du processus à pile sous-jacent. Lorsque le contexte est clair, nous ne préciserons pas systématiquement si l'on considère des processus à pile avec ou sans étiquetage. En fait, dans tout ce qui va suivre, on se placera dans le cadre où il n'y a pas d'étiquette.

Exemple 3.1 La figure 3.3 illustre le graphe engendré par un processus à pile.

$\mathcal{P} = \langle Q, A, \Gamma, \bot, \Delta \rangle$ avec :

$$
\begin{aligned}
&Q = \{p, q, r\}\\
&\Gamma = \{\alpha, \beta, \bot\}\\
&\Delta(p, \alpha) = \{pop(q), push(r, \alpha)\}\\
&\Delta(q, \alpha) = \{pop(p), push(r, \alpha)\}\\
&\Delta(r, \alpha) = \{pop(p), pop(r)\}\\
&\Delta(p, \beta) = \{pop(p), push(p, \alpha)\}\\
&\Delta(q, \beta) = \{pop(p), push(r, \alpha)\}\\
&\Delta(r, \beta) = \{pop(p), push(q, \alpha)\}\\
&\Delta(p, \bot) = \{push(p, \alpha), push(r, \beta)\}\\
&\Delta(q, \bot) = \{push(q, \alpha), push(r, \alpha)\}\\
&\Delta(r, \bot) = \{push(q, \beta), push(r, \alpha)\}
\end{aligned}
$$

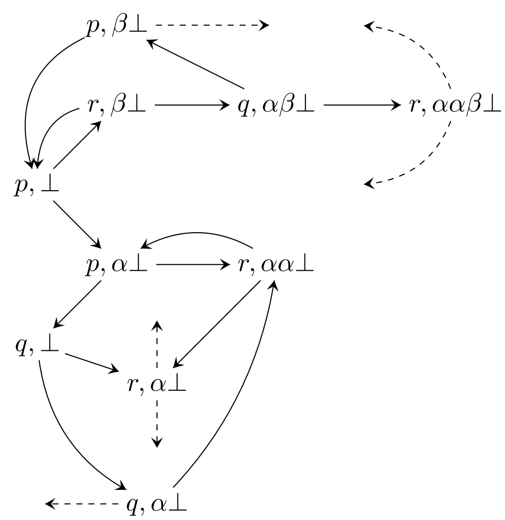


FIG. 3.1 – Exemple de graphe de processus à pile

3.1.2 Graphe de jeu et jeux associés

Dans le paragraphe 3.1.1 nous avons montré comment construire, à partir d'un processus à pile, un graphe infini. On explique maintenant comment définir, à partir d'un graphe de processus à pile, un graphe de jeu. On explique ensuite comment définir les conditions de gain classiques et l'on introduit également de nouvelles conditions de gains.

Graphe de jeu

Définition 3.3 *Soit un processus à pile $\mathcal{P} = \langle Q, A, \Gamma, \bot, \Delta \rangle$ et soit $G = (V, E)$ le graphe engendré par $\mathcal{P}$. Soit une partition $Q_{\mathbf{E}} \cup Q_{\mathbf{A}}$ de Q entre Eve et Adam. La partition précédente induit* *une partition $V_{\mathbf{E}} \cup V_{\mathbf{A}}$ de V définie par $V_{\mathbf{E}} = Q_{\mathbf{E}} \times (\Gamma \setminus \{\bot\})^* \bot$ et $V_{\mathbf{A}} = Q_{\mathbf{A}} \times (\Gamma \setminus \{\bot\})^* \bot$: les configurations d'Eve sont celles dont l'état de contrôle est dans $Q_{\mathbf{E}}$. Le graphe de jeu $\mathcal{G} = (G, V_{\mathbf{E}}, V_{\mathbf{A}})$ est appelé **graphe de jeu sur le processus à pile $\mathcal{P}$ induit par** $Q_{\mathbf{E}} \cup Q_{\mathbf{A}}$.*

Considérons un processus à pile $\mathcal{P} = \langle Q, A, \Gamma, \bot, \Delta \rangle$, une partition $Q_{\mathbf{E}} \cup Q_{\mathbf{A}}$ de Q, et appelons $\mathcal{G}$ le graphe de jeu associé. Etant donnée une configuration $v = (q, \sigma)$, on appelle **hauteur** de v, que l'on note $|v|$, la taille $|\sigma|$ du contenu de pile dans v.

Les hauteurs de pile vont jouer un rôle crucial dans la résolution des jeux sur des graphes de processus à pile. On s'intéressera essentiellement à savoir si dans une configuration de hauteur donnée, on revient un jour dans une configuration de même hauteur. Pour illustrer les raisonnements, on adoptera la représentation d'une partie par un graphique défini comme suit : l'axe des abscisses représente le temps tandis que l'axe des ordonnées représente la hauteur de pile. Un tel graphique permet donc de lire l'évolution de la hauteur de pile au cours de la partie. La figure 3.2 donne un exemple d'une telle représentation : on commence par empiler puis dépiler (on parle de bosse). Ensuite on empile deux symboles puis un troisième que l'on dépile. Ensuite on empile un symbole puis un second que l'on dépile…

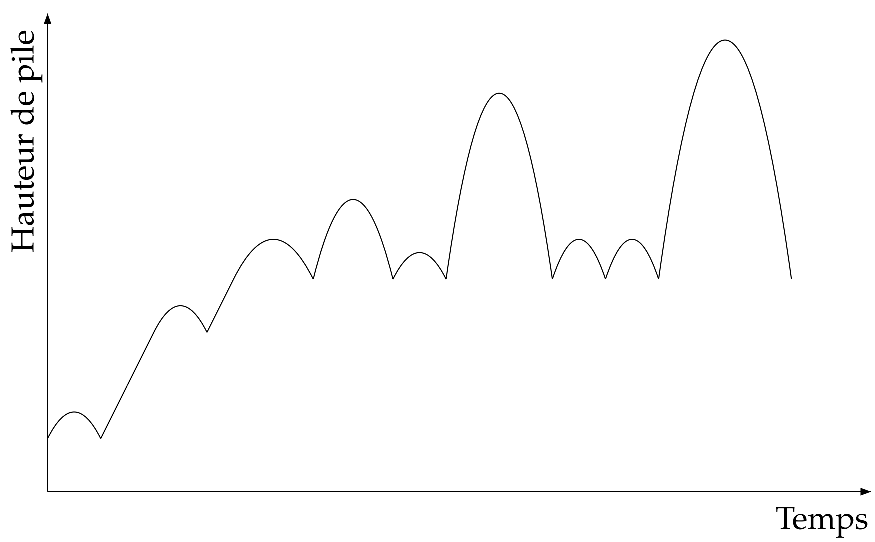


FIG. 3.2 – Représentation graphique d'une partie

Conditions d'accessibilité, de Büchi et de parité

Dans la construction du graphe de jeu, on a utilisé les états de contrôle pour partager les configurations entre les deux joueurs. La même idée nous permet de définir des configurations finales et des fonctions de coloriage. Les jeux de Büchi et de parité ne considérant pas l'étiquetage des arcs, on considère ici des graphes de jeu engendrés par des processus à pile non étiqueté.

Définition 3.4 (Jeu d'accessibilité, de Büchi) *Soit un graphe de jeu engendré par un processus à pile $\mathcal{P} = \langle Q, \Gamma, \perp, \Delta \rangle$ et une partition $Q_{\mathbf{E}} \cup Q_{\mathbf{A}}$ de Q. Soit $\mathcal{G}$ le graphe de jeu engendré par $\mathcal{P}$ et la partition précédente. Soit un sous-ensemble F de Q. L'ensemble F induit alors un ensemble $V_F = F \times (\Gamma \setminus \{\perp\})^* \perp$ de configurations finales.*

Soit Ω_{Acc} la condition d'accessibilité associée à V_F sur $\mathcal{G}$. Le jeu $\mathbb{G} = (\mathcal{G}, \Omega_{Acc})$ est qualifié de ***jeu d'accessibilité sur un graphe de processus à pile****.*

Soit Ω_{Buc} la condition de Büchi associée à V_F sur $\mathcal{G}$. Le jeu $\mathbb{G} = (\mathcal{G}, \Omega_{Buc})$ est qualifié de ***jeu de Büchi sur un graphe de processus à pile****.*

Définition 3.5 (Jeu de Parité) *Soit un graphe de jeu engendré par un processus à pile $\mathcal{P} = \langle Q, \Gamma, \perp, \Delta \rangle$ et une partition $Q_{\mathbf{E}} \cup Q_{\mathbf{A}}$ de Q. Soit $\mathcal{G} = ((V, E), V_{\mathbf{E}}, V_{\mathbf{A}})$ le graphe de jeu engendré par $\mathcal{P}$ et la partition précédente. Soit une fonction de coloriage col de Q dans un ensemble de couleur C. On étend col en une fonction de V dans C en posant $col'((q, \sigma)) = col(q)$ pour tout état q et tout contenu de pile σ. On appelle Ω la condition de parité associée à col' sur $\mathcal{G}$. Le jeu $\mathbb{G} = (\mathcal{G}, \Omega)$ est qualifié de* ***jeu de parité sur un graphe de processus à pile****.*

Exemple 3.2 La figure 3.3 illustre le graphe de jeu et le coloriage engendrés par un processus à pile.

$\mathcal{P} = \langle Q, A, \Gamma, \perp, \Delta \rangle$ avec :

$$
\begin{aligned}
Q &= \{p, q, r\} \\
\Gamma &= \{\alpha, \beta, \perp\} \\
\Delta(p, \alpha) &= \{pop(q), push(r, \alpha)\} \\
\Delta(q, \alpha) &= \{pop(p), push(r, \alpha)\} \\
\Delta(r, \alpha) &= \{pop(p), pop(r)\} \\
\Delta(p, \beta) &= \{pop(p), push(p, \alpha)\} \\
\Delta(q, \beta) &= \{pop(p), push(r, \alpha)\} \\
\Delta(r, \beta) &= \{pop(p), push(q, \alpha)\} \\
\Delta(p, \perp) &= \{push(p, \alpha), push(r, \beta)\} \\
\Delta(q, \perp) &= \{push(q, \alpha), push(r, \alpha)\} \\
\Delta(r, \perp) &= \{push(q, \beta), push(r, \alpha)\}
\end{aligned}
$$

$Q_E = \{p, q\}$ et $Q_A = \{r\}$

$\rho(p) = 0, \rho(q) = 2, \rho(r) = 1$

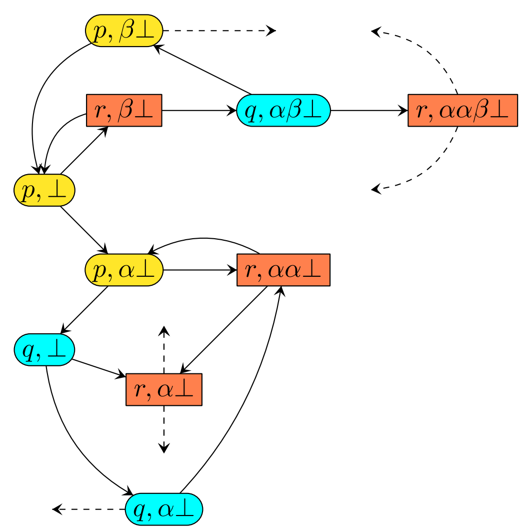


FIG. 3.3 – Exemple de jeu de parité engendre par un processus à pile

Conditions de bornage

Les processus à pile permettent de modéliser des programmes autorisant des appels récursifs, stockés dans la pile. Dans ce formalisme, la hauteur de pile représente le nombre d'appels récursifs en cours. Il est alors naturel de se demander si la pile est bornée ou pas au cours d'une partie, ce qui exprime le fait que le nombre d'appels récursifs soit borné ou non.

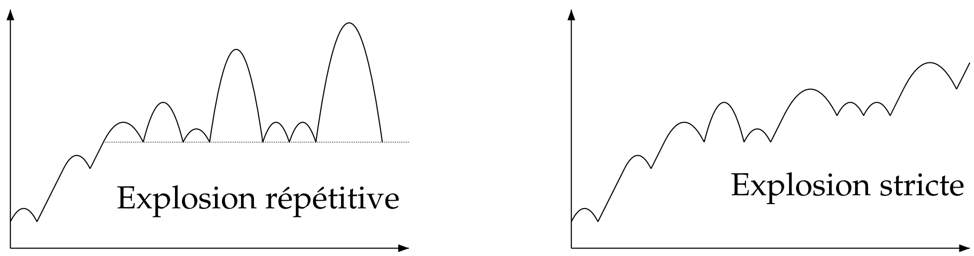


FIG. 3.4 – Les deux types de parties gagnantes pour l'explosion

On considère un processus à pile $\mathcal{P} = \langle Q, \Gamma, \perp, \Delta \rangle$ et une partition $Q_{\mathbf{E}} \cup Q_{\mathbf{A}}$ de Q. On appelle $\mathcal{G} = ((V, E), V_{\mathbf{E}}, V_{\mathbf{A}})$ le graphe de jeu engendré par $\mathcal{P}$ et la partition précédente, et on suppose qu'il est sans cul-de-sac. Pour tout entier $n \geq 1$, on note V_n l'ensemble des configurations de $\mathcal{G}$ de hauteur de pile égale à n, par $V_{\leq n}$ l'ensemble des configurations de $\mathcal{G}$ de hauteur de pile inférieure ou égale à n et par $V_{\geq n}$ l'ensemble des configurations de $\mathcal{G}$ de hauteur de pile supérieure ou égale à n.

La condition de gain $\Omega_{Exp} = \bigcap_{n \geq 1} [(V^* V_n) V^\omega]$ est qualifiée de **condition d'explosion**. Le jeu $\mathbb{G} = (\mathcal{G}, \Omega_{Exp})$ est qualifié de **jeu d'explosion**.

La condition duale $\Omega_{Bor} = \bigcup_{n \geq 1} V_{\leq n}^\omega$ de la condition d'explosion est qualifiée de **condition de bornage**.

Une partie est remportée par Eve dans un jeu d'explosion si et seulement si la pile est non bornée. On peut distinguer deux types possibles de partie gagnantes pour Eve, selon qu'un niveau de pile est infiniment souvent visité ou non. Ces deux cas sont représentés dans la figure 3.4. Une partie est remportée par Eve pour la condition de bornage si et seulement si la hauteur de pile est bornée.

On définit une variante de la condition d'explosion, la condition d'explosion stricte (dont l'interprétation graphique est donnée dans la figure 3.4).

La condition de gain $\Omega_{ExpSt} = \bigcap_{n \geq 1} V^* V_{\geq n}^\omega$ est qualifiée de **condition d'explosion stricte**. Le jeu $\mathbb{G} = (\mathcal{G}, \Omega_{ExpSt})$ est qualifié de **jeu d'explosion stricte**.

La condition duale Ω_{Rep} de la condition d'explosion stricte, que l'on appelle **condition de répétition** est donnée par $\bigcup_{n \geq 1} \bigcap_{m \geq 0} [(V^m V^* V_n) V^\omega]$.

Une partie est remportée par Eve dans un jeu d'explosion stricte si et seulement si toute hauteur de pile est un jour quittée pour toujours. Une partie est remportée par Adam si et seulement si un niveau de pile est infiniment souvent visité, ou de façon équivalente si et seulement si une configuration est infiniment souvent visitée.

3.2 Résolution des jeux d'accessibilité

Dans tout ce qui suit on se donne un graphe de jeu engendré par un processus à pile $\mathcal{P} = \langle Q, \Gamma, \perp, \Delta \rangle$ et une partition $Q_{\mathbf{E}} \cup Q_{\mathbf{A}}$ de Q. Soit $\mathcal{G}$ le graphe de jeu engendré par $\mathcal{P}$ et la partition précédente. Soit un sous-ensemble F de Q et soit $\mathbb{G}$ le jeu d'accessibilité associé.

3.2.1 Factorisation des parties

Considérons une partie Λ dans $\mathbb{G}$ et sa représentation graphique, comme illustré dans la figure 3.2.

L'idée dans ce qui suit va être de décomposer une telle partie. Etant donnée une partie (partielle ou non) $\Lambda = v_0 v_1 v_2 \cdots$ dans $\mathbb{G}$, et un entier $h \geq 1$, un facteur $v_i \cdots v_j$ de Λ est une **bosse sur le niveau** h si et seulement si $|v_i| = |v_j| = h$ et pour tout $i < k < j$, $|v_k| > h$. Par opposition aux bosses, on appelle **marche quittant le niveau** h un facteur de Λ, $v_i v_{i+1}$, de longueur 2, tel que $|v_i| = h$, $|v_{i+1}| = h+1$, et pour tout $i+1 \leq k < |\Lambda|$, $|v_k| > h$. En d'autres termes, une marche est un facteur de longueur 2 qui n'est pas préfixe d'une bosse.

Il est alors naturel, étant donnée une partie Λ dans $\mathbb{G}$, de considérer les positions suivantes.

Définition 3.6 *Etant donnée une partie (partielle ou non) $\Lambda = v_0 v_1 v_2 \cdots$ dans $\mathbb{G}$, on note $Steps_\Lambda$ le sous-ensemble d'indices correspondant à des configurations de hauteur de pile minimale par rapport au futur de la partie. Plus précisément :*

$$Steps_\Lambda = \{n \in \mathbb{N} \mid \forall m \geq n,\ |v_m| \geq |v_n|\},$$

où $|v_n|$ désigne la hauteur de la pile dans la configuration v_n.

Ainsi, l'ensemble $Steps_\Lambda$ induit une factorisation naturelle de Λ en bosses et en marches.

Définition 3.7 (M/B factorisation) *Etant donnée une partie (partielle ou non) $\Lambda = v_0 v_1 v_2 \cdots$, on appelle **Marche/Bosse factorisation** de Λ (ou plus simplement **M/B factorisation**) la suite (finie or infinie) $(\Lambda_i)_{i \geq 0}$ de facteurs de Λ définis de la façon suivante. Soit $Steps_\Lambda = \{n_0 < n_1 < n_2 < \cdots\}$, on pose alors, pour tout $0 \leq i < |Steps_\Lambda|$, $\Lambda_i = v_{n_i} \cdots v_{n_{i+1}}$. Ainsi, pour tout $i \geq 0$, le premier sommet de Λ_{i+1} est le dernier sommet de Λ_i. De plus $\Lambda = \Lambda_1 \odot \Lambda_2 \odot \Lambda_3 \odot \cdots$ où $\Lambda_i \odot \Lambda_{i+1}$ désigne la concaténation de Λ_i et de Λ_{i+1} privée de son premier sommet.*

*Enfin, on qualifiera un facteur Λ_i de **marche** si $|v_{n_{i+1}}| = |v_{n_i}| + 1$ et sinon de **bosse** (et dans ce cas, $|v_{n_{i+1}}| = |v_{n_i}|$).*

La figure 3.5 illustre la M/B factorisation de la partie représentée dans la figure 3.2.

Considérons plus précisément une condition d'accessibilité. Il est alors facile de voir le point suivant.

Fait 3.1 *Soit Λ une partie remportée par Eve. Il existe alors un préfixe de Λ qui est une partie partielle telle que le dernier sommet dans cette partie est final et aucun avant ne l'est.*

Toute l'idée de la suite est de simuler la M/B factorisation d'une telle partie partielle jusqu'à éventuellement tomber sur un sommet final.

3.2.2 Simulation

De façon assez informelle, imaginons maintenant un nouveau jeu dans lequel Eve et Adam se mettent directement d'accord sur la M/B factorisation. Cette simulation est illustrée par la bande dessinée qui suit.

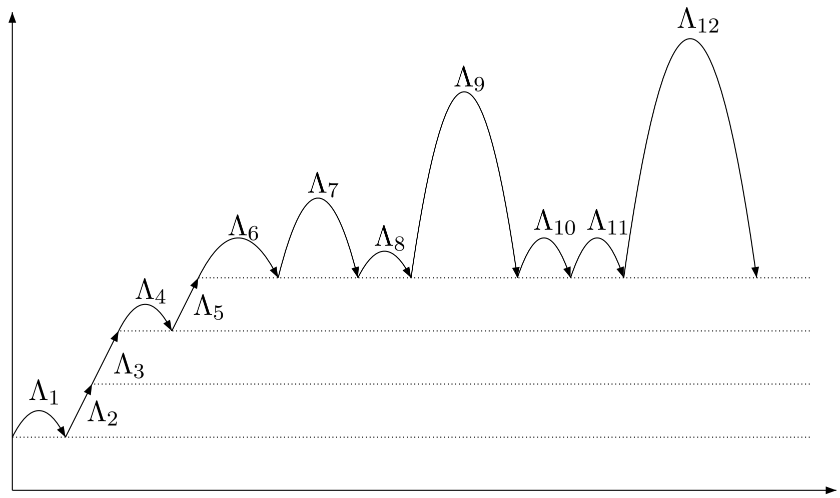
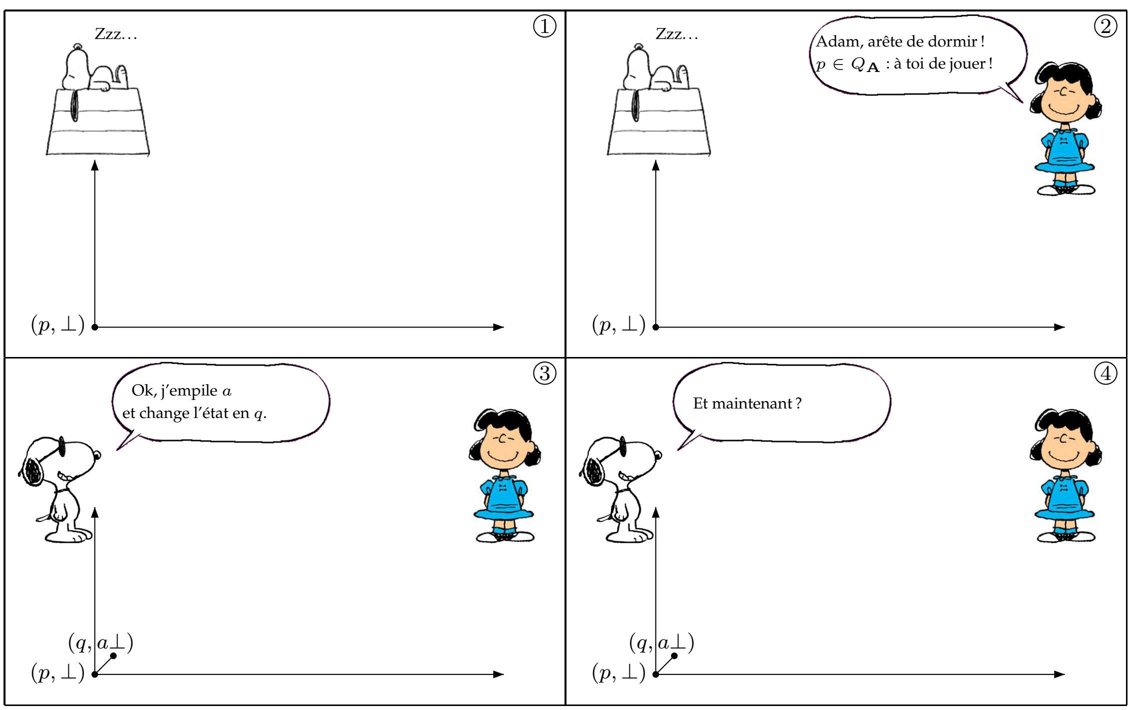
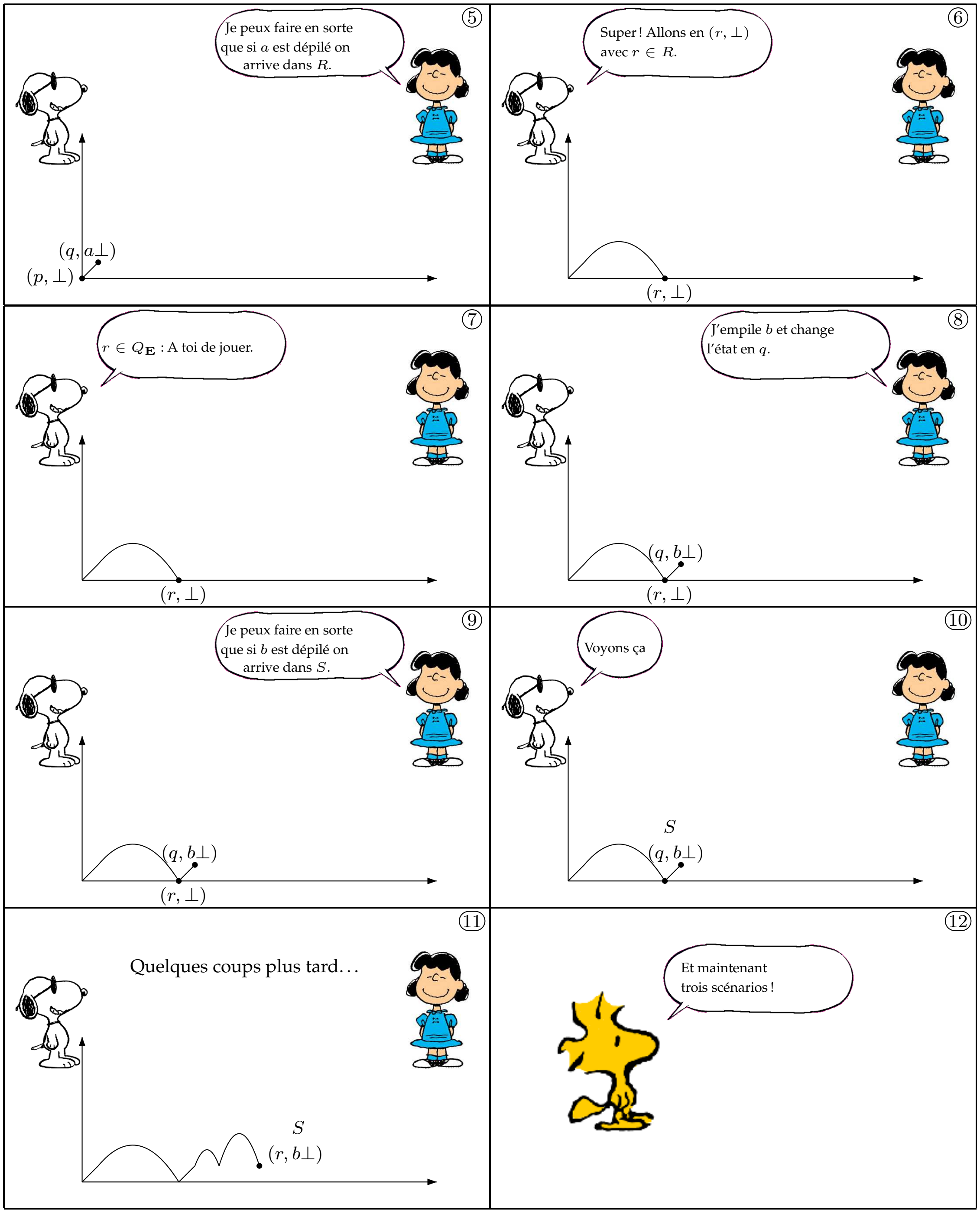
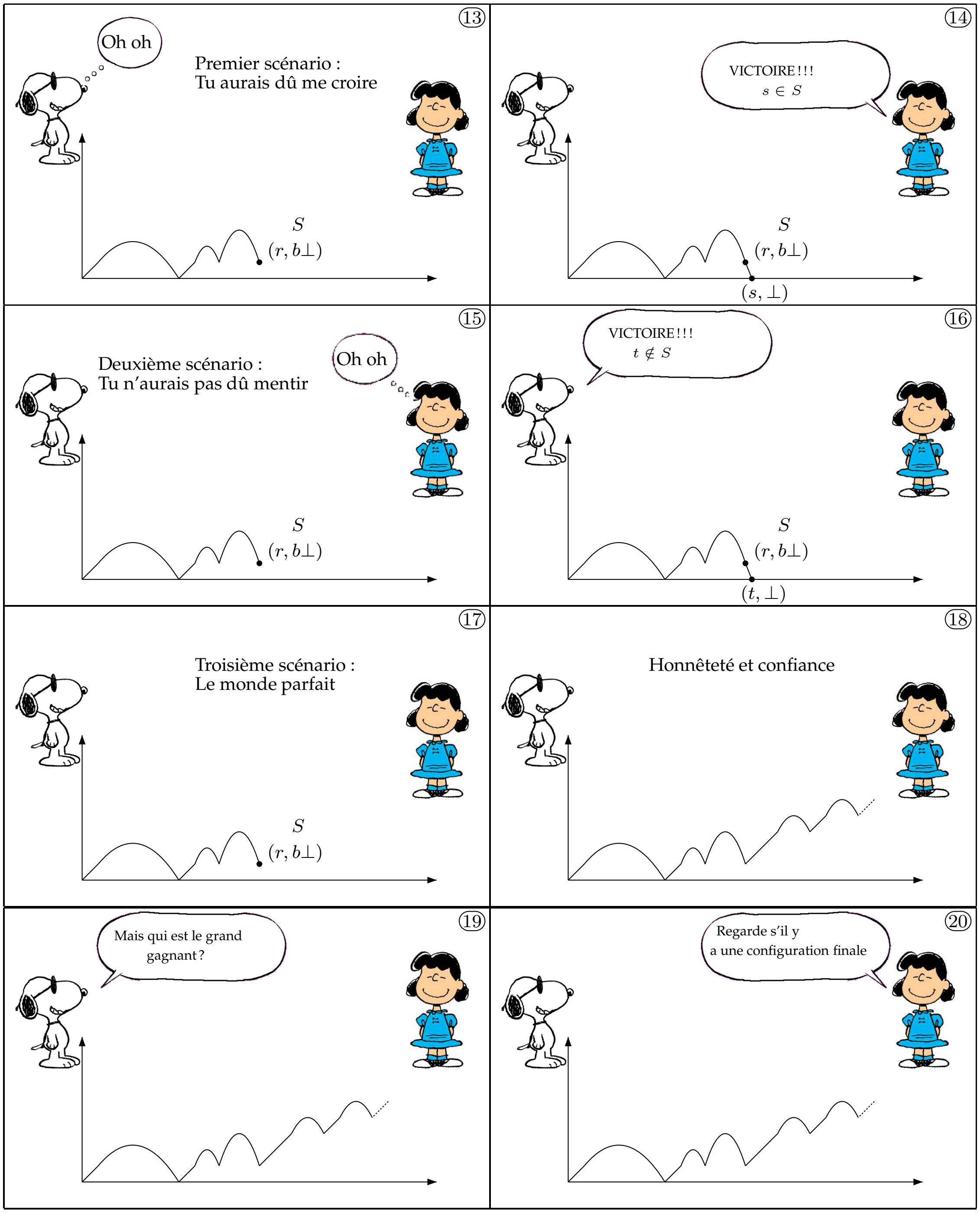


FIG. 3.5 – M/B factorisation




⑤
Je peux faire en sorte que si a est dépilé on arrive dans R.
$(q, a\perp)$
$(p, \perp)$
⑥
Super ! Allons en $(r, \perp)$ avec $r \in R$.
$(r, \perp)$
⑦
$r \in Q_{\mathbf{E}}$: A toi de jouer.
$(r, \perp)$
⑧
J'empile b et change l'état en q.
$(q, b\perp)$
$(r, \perp)$
⑨
Je peux faire en sorte que si b est dépilé on arrive dans S.
$(q, b\perp)$
$(r, \perp)$
⑩
Voyons ça
S
$(q, b\perp)$
⑪
Quelques coups plus tard…
S
$(r, b\perp)$
⑫
Et maintenant trois scénarios !



Afin de simuler une partie dans le jeu $\mathbb{G}$ depuis une configuration $(p, \bot)$, Eve et Adam procèdent de la façon suivante. Le joueur à qui appartient la configuration $(p, \bot)$ peut choisir, a priori, d'empiler, de dépiler, ou de ne pas modifier la pile (vignettes 1 à 2).

Dans le cas où il ne modifie pas la pile, c'est-à-dire qu'une règle de la forme $skip(q) \in \Delta(p, \bot)$ est appliquée, le facteur correspondant est la bosse triviale
$(p, \bot)(q, \bot)$.

Dans le cas où il décide d'empiler (vignette 3), c'est-à-dire d'appliquer une règle de la forme $push(q, a) \in \Delta(p, \perp)$, on souhaite décider si $(p, \perp)(q, a\perp)$ sera une marche ou le préfixe d'une bosse. Bien sûr, cela dépend de la manière dont Eve et Adam vont décider de jouer. Eve donne alors à Adam des informations sur la stratégie qu'elle va adopter, à savoir un sous-ensemble $R \subseteq Q$ d'états r pouvant être atteints si le a est dépilé et qu'aucun n'état final n'est visité entre temps (vignette 5). Adam peut alors prolonger la partie en une bosse (vignette 6), et donc continuer la partie depuis une position, de son choix, $(r, \perp)$ avec $r \in R$ (vignette 7 à 9), ou continuer avec la nouvelle lettre comme sommet de pile (vignettes 10 avec b comme sommet de pile et S comme ensemble). Dans ce dernier cas, la partie se prolonge depuis $(q, b\perp)$ si b est le nouveau sommet de pile. Si b est un jour dépilé, Eve remporte la partie (vignettes 13 et 14) si l'état de contrôle est dans S (S est l'ensemble annoncé par Eve précédemment), et sinon c'est Adam qui gagne (vignettes 15 et 16). Ainsi, la nature de ce second choix est double du point de vue d'Adam : il peut lui permettre de prouver que l'ensemble S donné par Eve est trop petit (il existe un état qui n'est pas dans S et qui est obtenu en dépilant b), ou simplement de simuler une partie où b ne sera jamais dépilé.

Quel que soit le choix d'Adam, les deux joueurs se sont mis d'accord sur le premier facteur de la M/B factorisation. De plus, il est facile pour eux de décider s'il s'agit d'une marche ou d'une bosse, et de savoir si le dernier sommet est final. Si la partie est infinie (vignettes 17 à 20), ils peuvent décider qui est le gagnant, en regardant si un jour un sommet final a été visité.

Dans ce qui suit, on va prouver que ces deux jeux sont *équivalents* (en un sens à préciser).

3.2.3 La réduction

Dans le jeu précédemment décrit, on peut remarquer qu'à tout moment la seule information utile est le sommet de pile, l'état de contrôle de la configuration courante, ainsi que le vecteur de sous-ensembles de Q fourni par Eve. Cette information étant finie, l'idée est de coder le jeu précédent dans un jeu sur un graphe fini.

On considère le graphe de jeu $\widetilde{\mathcal{G}}$ illustré dans la figure 3.6, et dont voici une description :

- les sommets principaux de $\widetilde{\mathcal{G}}$ sont ceux de la forme (p, a, R), où $p \in Q$, $a \in \Gamma$ et $R \in \mathcal{P}(Q)$. Un sommet (p, a, R) représente une partie partielle Λ dans $\mathbb{G}$ telle que :
  - le dernier sommet de Λ est de la forme (p, au) pour un certain $u \in \Gamma^*$.
  - Eve prétend pouvoir jouer depuis Λ de telle sorte que si a est un jour dépilé et qu'on ne voit pas d'état final entre temps, l'état de contrôle atteint après avoir dépilé a est dans R.

  Un sommet de la forme (p, a, R) appartient à Eve si et seulement si $p \in Q_{\mathbf{E}}$.
- les états *tt* et *ff* sont là pour s'assurer que les ensembles R dans les sommets principaux sont corrects. Le sommet *tt* appartient à Adam, tandis que *ff* appartient à Eve. Il y a des boucles sur ces deux sommets, et *tt* est final. Ainsi une partie qui arrive dans *tt* sera remportée par Eve tandis qu'une partie arrivant dans *ff* sera remportée par Adam.

  Il y a une transition d'un sommet (p, a, R) vers *tt*, si et seulement s'il existe une règle de la forme $pop(r) \in \Delta(p, a)$, telle que $r \in R$ (ce qui traduit le fait que R est correct par rapport à cette règle). Symétriquement, il y a une transition d'un sommet (p, a, R)

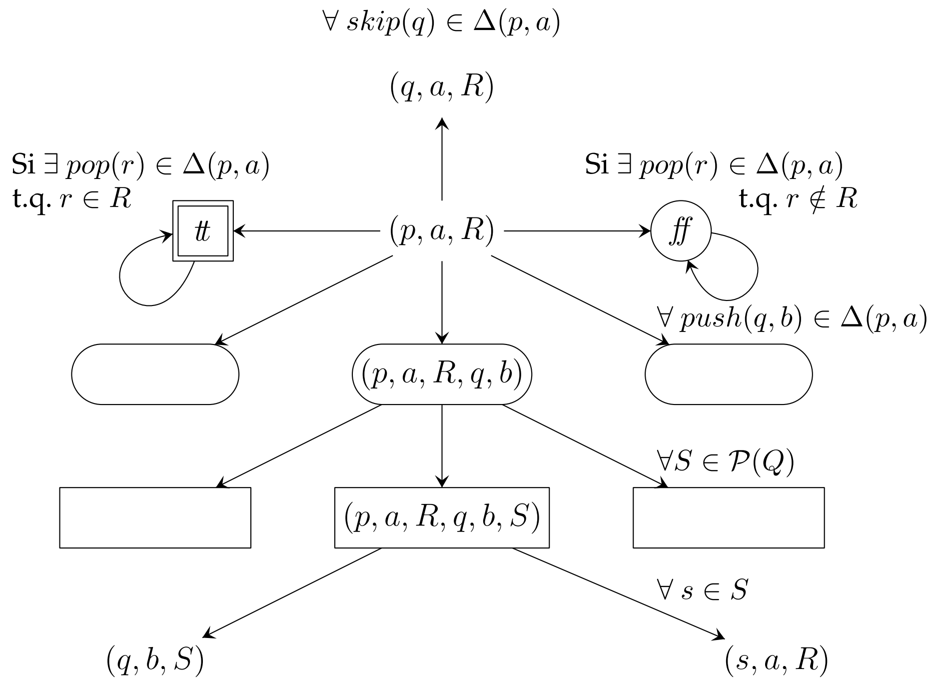


FIG. 3.6 – Structure locale de $\widetilde{\mathcal{G}}$.

vers *ff* si et seulement s'il existe une règle de la forme $pop(r) \in \Delta(p,a)$, telle que $r \notin R$ (ce qui traduit le fait que R n'est pas correct par rapport à cette règle).
- afin de simuler l'application d'une transition $skip(q) \in \Delta(p,a)$, le joueur à qui appartient (p,a,R) va en (q,a,R).
- afin de simuler l'application d'une transition $push(q,b) \in \Delta(p,a)$, le joueur à qui appartient (p,a,R) va en (p,a,R,q,b). Ce dernier sommet appartient à Eve et celle-ci doit donc choisir un ensemble $S \in \mathcal{P}(Q)$ décrivant les états pouvant être atteints si b est un jour dépilé. Eve va alors dans le sommet (p,a,R,q,b,S).
  Ce dernier appartient à Adam. Depuis (p,a,R,q,b,S), Adam choisit s'il veut simuler une bosse ou une marche. Dans le premier cas, il va dans un sommet (s,a,R) avec $s \in S$. Dans le second cas, Adam va dans le sommet (q,b,S).

Le graphe de jeu $\widetilde{\mathcal{G}}$ a pour sommet finaux : *tt* et tout sommet de la forme (p,a,R) avec p final. Pour ces derniers sommets, on supposera qu'il y a une boucle dessus et que c'est la seule transition qui en part.

Si l'on appelle $\widetilde{\mathbb{G}}$ le jeu d'accessibilité sur $\widetilde{\mathcal{G}}$, on a le résultat suivant.

Théorème 3.1 *Soit $p_{in} \in Q$. Eve possède une stratégie gagnante dans $\mathbb{G}$ depuis $(p_{in},\perp)$ si et seulement si elle possède une stratégie gagnante dans $\widetilde{\mathbb{G}}$ depuis $(p_{in},\perp,\emptyset)$.*

Le reste de cette section est consacré à la preuve du Théorème 3.1.

Préliminaires : factorisation dans $\widetilde{\mathbb{G}}$ Remarquons tout d'abord qu'une partie dans $\widetilde{\mathbb{G}}$ visite *régulièrement*, au plus tous les trois coups, des sommets de la forme (p, a, R). On qualifie de **round** la séquence de sommets entre deux passages dans de telles positions. Un round peut alors être de plusieurs types :

- il est de la forme $(p, a, R)(q, a, R)$ et correspond donc à la simulation d'une règle de type skip. On parle alors de *bosse (triviale)*.
- il est de la forme $(p, a, R)(p, a, R, q, b)(p, a, R, q, b, S)(s, a, R)$ et correspond donc à la simulation d'une règle empilant un b suivie d'une série de coups se terminant par le dépilement du b. On le qualifie de *bosse*.
- il est de la forme $(p, a, R)(p, a, R, q, b)(p, a, R, q, b, S)(q, b, S)$ et correspond donc à la simulation d'une règle d'empilement d'un b qui ne sera pas dépilé. On le qualifie de *marche*.

Etant donnée une partie $\lambda = v_0 v_1 v_2 \cdots$ dans $\widetilde{\mathbb{G}}$, on peut considérer le sous-ensemble d'indices correspondant à des sommets de la forme (p, a, R). Plus précisément :

$$Steps_\lambda = \{n \in \mathbb{N} \mid v_n = (p, a, R),\ p \in Q,\ a \in \Gamma,\ R \in \mathcal{P}(Q)\}$$

Tout comme dans le cas des jeux sur un graphe de processus à pile, l'ensemble $Steps_\lambda$ induit une factorisation naturelle d'une partie λ en bosses et en marches.

Définition 3.8 (M/B factorisation) *Etant donnée une partie, partielle ou non, $\lambda = v_0 v_1 v_2 \cdots$ dans $\widetilde{\mathbb{G}}$, on appelle **Marche/Bosse factorisation** de λ, ou plus simplement **M/B factorisation**, la suite, finie ou infinie, $(\lambda_i)_{i \geq 0}$ de rounds de λ définis de la façon suivante. Soit $Steps_\lambda = \{n_0 < n_1 < n_2 < \cdots\}$, on pose alors, pour tout $0 \leq i < |Steps_\lambda|$, $\lambda_i = v_{n_i} \cdots v_{n_{i+1}}$.*

Ainsi, pour tout $i \geq 0$, le premier sommet de λ_{i+1} est le dernier sommet de λ_i. De plus, $\lambda = \lambda_1 \odot \lambda_2 \odot \lambda_3 \odot \cdots$, où $\lambda_i \odot \lambda_{i+1}$ désigne la concaténation de λ_i et de λ_{i+1} privé de son premier sommet.

Implication directe Supposons que la configuration $(p_{in}, \perp)$ est gagnante pour Eve dans le jeu $\mathbb{G}$ et considérons une stratégie Φ gagnante pour Eve dans $\mathbb{G}$ depuis $(p_{in}, \perp)$

A partir de Φ, on définit une stratégie φ pour Eve dans $\widetilde{\mathbb{G}}$ depuis $(p_{in}, \perp, \emptyset)$. La stratégie utilise une mémoire qui contient une partie partielle dans $\mathbb{G}$, c'est-à-dire un élément de V^*, et son contenu sera noté Λ. Au départ, Λ est réduite au sommet $(p_{in}, \perp)$. On commence par décrire φ, puis on explique comment Λ est mise à jour. La stratégie φ, ainsi que la mise à jour de Λ sont décrites pour un round.

Choix du coup : On suppose donc que l'on est dans une configuration de la forme (p, a, R) avec $p \in Q_\mathbf{E} \setminus F$ (si $p \in F$, on n'a qu'un coup possible, qui est une boucle, donc la définition de φ est triviale). Le coup donné par φ dépend alors de $\Phi(\Lambda)$:

- si $\Phi(\Lambda) = pop(r)$, alors Eve va dans $\sharp$ (la proposition 3.1 montrera que ce coup est toujours possible).
- si $\Phi(\Lambda) = skip(q)$, alors Eve va dans (q, a, R).
- si $\Phi(\Lambda) = push(q, b)$, alors Eve va dans (p, a, R, q, b).

Dans ce dernier cas, ou lorsque $p \in Q_\mathbf{A}$ et qu'Adam est allé en (p, a, R, q, b), nous devons également expliquer comment Eve joue depuis (p, a, R, q, b). Elle doit choisir un ensemble $S \in \mathcal{P}(Q)$ qui décrit les états atteignables si b est dépilé sans que l'on voit un état final. Afin de définir S, Eve considère l'ensemble des parties possibles dans $\mathbb{G}$, où elle respecte sa stratégie Φ, et qui commencent par $\Lambda \cdot (q, bau)$, si (p, au) désigne le dernier sommet de Λ. Pour chacune de ces parties, Eve regarde si une configuration de la forme (s, au) apparaît après $\Lambda \cdot (q, bau)$, c'est-à-dire si b est un jour dépilé. Si tel est le cas, Eve considère la première configuration (s, au) apparaissant après $\Lambda \cdot (q, bau)$ et regarde si l'on a visité une configuration finale lorsque b se trouvait dans la pile. S est exactement l'ensemble des $s \in Q$, tel que l'on ait la propriété précédente et qu'on ne voit pas d'état final entre temps. Eve joue alors vers (p, a, R, q, b, S).

Mise à jour de Λ **:** La mise à jour de Λ est effectuée chaque fois que l'on arrive dans un sommet de la forme (p, a, R). On a alors trois cas, selon la nature du round :

- le round est une bosse triviale et l'on vient donc de simuler une action $skip(q)$. Si l'on appelle (p, au) le dernier sommet de Λ, on augmente Λ du sommet (q, au).
- le round est une bosse et l'on vient donc de simuler une bosse commençant par une action $push(q, b)$ et se terminant dans un état s. Si l'on appelle (p, au) le dernier sommet de Λ, on met à jour Λ en y ajoutant (q, bau) suivi d'une série de coups, où Eve respecte Φ, et qui conduisent à dépiler b et à arriver dans (s, au) sans visiter de configuration finale entre temps. Par définition de φ, une telle suite de sommets existe toujours.
- le round est une marche et l'on vient donc de simuler une action $push(q, b)$. Si l'on appelle (p, au) le dernier sommet de Λ, on augmente Λ du sommet (q, bau).

Dès lors, à toute partie partielle λ dans $\widetilde{\mathbb{G}}$ dans laquelle Eve respecte sa stratégie φ, est associée une partie partielle Λ dans $\mathbb{G}$. Une récurrence immédiate permet de prouver que Λ est une partie dans laquelle Eve respecte Φ. Le même raisonnement peut être mené lorsque λ est une partie infinie ne visitant pas d'état final, et la partie Λ associée est alors infinie, commence en $(p_{in}, \perp)$ et lors de celle-ci, Eve respecte sa stratégie gagnante Φ. En particulier Λ visite une configuration finale.

La proposition suivante découle directement de la définition de φ.

Proposition 3.1 *Soit λ une partie partielle dans $\widetilde{\mathbb{G}}$ commençant en $(p_{in}, \perp, \emptyset)$, se terminant dans un sommet de la forme (p, a, R), ne visitant pas d'état final, et où Eve respecte φ. Soit Λ la partie associée à λ construite par la stratégie φ. On a alors les points suivants :*

1. *Λ est une partie où Eve respecte Φ.*
2. *Λ se termine dans un sommet de la forme (p, au) pour un certain $u \in \Gamma^*$.*
3. *si la partie Λ est prolongée en une partie où Eve respecte Φ, alors, si a est dépilé et qu'aucune configuration finale n'est visitée entre temps, la configuration (r, u) alors atteinte est telle que $r \in R$.*

Remarque 3.1 La proposition 3.1 implique que la stratégie φ est bien définie lorsqu'elle donne un coup vers *tt*. De même, on en déduit que si Eve respecte φ, le sommet *ff* ne peut être atteint.

Des définitions de $\widetilde{\mathcal{G}}$ et de φ, on déduit la proposition suivante.

Proposition 3.2 *Soit λ une partie infinie dans $\widetilde{\mathbb{G}}$ commençant en $(p_{in}, \perp, \emptyset)$, ne visitant pas de configuration finale et où Eve respecte φ. Soit Λ la partie associée à λ construite par la stratégie φ. Soient $(\lambda_i)_{i \geq 1}$ et $(\Lambda_i)_{i \geq 1}$ les M/B factorisations respectives de λ et Λ. On a alors pour tout $i \geq 1$:*

1. *Λ est une partie où Eve respecte Φ.*
2. *λ_i est un bosse si et seulement si Λ_i est une bosse.*
3. *Si λ_i est une bosse, Λ_i ne contient pas de configuration finale.*

Considérons une partie (infinie) λ dans $\widetilde{\mathbb{G}}$ commençant en $(p_{in}, \perp, \emptyset)$ où Eve respecte φ. Si la partie est perdante pour Eve, il en va de même pour sa partie associée Λ car aucune de ses bosses ne contient une configuration finale (Proposition 3.2) et qu'aucune des extrémités de ses facteur n'est final (Proposition 3.1). Ceci contredit le fait que Φ est gagnante et conclu l'implication directe.

Implication réciproque Supposons maintenant que la configuration $(p_{in}, \perp)$ est gagnante pour Adam dans le jeu $\mathbb{G}$ et considérons une stratégie Φ gagnante pour Adam dans $\mathbb{G}$ depuis $(p_{in}, \perp)$.

A partir de Φ, on définit une stratégie φ pour Adam dans $\widetilde{\mathbb{G}}$ depuis $(p_{in}, \perp, \emptyset)$. La stratégie utilise une mémoire qui contient une partie partielle dans $\mathbb{G}$, c'est-à-dire un élément de V^*, et son contenu sera noté Λ. Au départ, Λ est réduite au sommet $(p_{in}, \perp)$. On commence par décrire φ, puis on explique comment Λ est mise à jour. La stratégie φ, ainsi que la mise à jour de Λ sont décrites pour un round. [1]

Choix du coup : On suppose donc que l'on est dans une configuration de la forme (p, a, R) avec $p \in Q_{\mathbf{A}} \setminus F$ (si $p \in F$, on n'a qu'un coup possible, qui est une boucle, donc la définition de φ est triviale). Le coup donné par φ dépend alors de $\Phi(\Lambda)$:

- si $\Phi(\Lambda) = pop(r)$, alors Adam va dans *ff* (la proposition 3.3 montrera que ce coup est toujours possible).
- si $\Phi(\Lambda) = skip(q)$, alors Adam va dans (q, a, R).
- si $\Phi(\Lambda) = push(q, b)$, alors Adam va dans (p, a, R, q, b).

Dans ce dernier cas, ou lorsque $p \in Q_{\mathbf{E}}$ et qu'Eve est allée en (p, a, R, q, b) puis en (p, a, R, q, b, S) pour un $S \in \mathcal{P}(Q)$, on doit expliquer comment Adam joue dans (p, a, R, q, b, S). Adam considère l'ensemble des parties possibles dans $\mathbb{G}$, où il respecte sa stratégie Φ, et qui commencent par $\Lambda \cdot (q, bau)$, si (p, au) désigne le dernier sommet de Λ. Pour chacune de ces parties, Adam regarde si une configuration de la forme (s, au) apparaît après $\Lambda \cdot (q, bau)$, c'est-à-dire si b est un jour dépilé. Si tel est le cas, Adam considère la première configuration (s, au) apparaissant après $\Lambda \cdot (q, bau)$ et regarde si l'on a visité une configuration finale lorsque b se trouvait dans la pile. On définit l'ensemble T comme l'ensemble des $s \in Q$, tel que l'on ait la propriété précédente et qu'on ne voit pas d'état final entre temps. Adam joue alors vers (t, a, R) s'il existe un $t \in S \cap T$, et Adam joue vers (q, b, S) sinon.

Mise à jour de Λ : La mise à jour de Λ est effectuée chaque fois que l'on arrive dans un sommet de la forme (p, a, R). On a alors trois cas, selon la nature du round :

- le round est une bosse triviale et l'on vient donc de simuler une action $skip(q)$. Si l'on appelle (p, au) le dernier sommet de Λ, on augmente Λ du sommet (q, au).

[1] Cette réciproque a pour avantage d'être pratiquement duale de la preuve directe. Cependant, elle possède un défaut majeur, puisque ni dans le sens direct ni dans le sens réciproque on ne montre comment construire une stratégie pour l'un des deux joueurs à partir d'une stratégie dans le jeu fini (stratégie que l'on sait calculer), ce qui donnerait alors une preuve constructrice en terme de stratégie. Il est à noter qu'une telle preuve existe mais, étant nettement plus compliquée nous nous contenterons ici de la preuve non constructive.

– le round est une bosse et l'on vient donc de simuler une bosse commençant par une action $push(q, b)$ et se terminant dans un état s. Si l'on appelle (p, au) le dernier sommet de Λ, on met à jour Λ en y ajoutant (q, bau) suivi d'une série de coups, où Adam respecte Φ, et qui conduisent à dépiler b et à arriver dans (s, au) sans visiter de configuration finale entre temps. Par définition de φ, une telle suite de sommets existe toujours.
– le round est une marche et l'on vient donc de simuler une action $push(q, b)$. Si l'on appelle (p, au) le dernier sommet de Λ, on augmente Λ du sommet (q, bau).

Dès lors, à toute partie partielle λ dans $\widetilde{\mathbb{G}}$ dans laquelle Adam respecte sa stratégie φ, est associée une partie partielle Λ dans $\mathbb{G}$. Une récurrence immédiate permet de prouver que Λ est une partie dans laquelle Adam respecte Φ.

La proposition suivante découle directement de la définition de φ.

Proposition 3.3 *Soit λ une partie partielle dans $\widetilde{\mathbb{G}}$ commençant en $(p_{in}, \perp, \emptyset)$, se terminant dans un sommet de la forme (p, a, R), ne visitant pas de sommet final sauf le dernier, et où Adam respecte φ. Soit Λ la partie associée à λ construite par la stratégie φ. On a alors les points suivants :*

1. *Λ est une partie où Eve respecte Φ.*
2. *Λ se termine dans un sommet de la forme (p, au) pour un certain $u \in \Gamma^*$.*
3. *si la partie Λ est prolongée en une partie où Adam respecte Φ, alors, si a est dépilé et qu'aucune configuration finale n'est visitée entre temps, la configuration (r, u) alors atteinte est telle que $r \notin R$.*

Preuve. Le premier point est évident. Pour le second, il suffit de remarquer que si la partie est dans (p, a, R), Adam a choisi à un moment de réaliser une bosse, qui dans Λ se traduit par le fait de quitter le niveau de pile de contenu u. Ce choix a été effectué car aucun prolongement de Λ où Adam respectait Φ ne conduisait dans une configuration (r, u) avec $r \in R$. ■

Le second point de la proposition 3.3 implique que la stratégie φ est bien définie lorsqu'elle donne un coup vers *ff*. De même, on en déduit que si Adam respecte φ, le sommet *tt* ne peut être atteint.

Le premier point de la proposition 3.3 implique qu'une partie où Adam respecte φ ne peut visiter de sommet (p, a, R) avec $p \in F$ (car sinon la partie associée dans $\mathbb{G}$ visite une configuration finale ce qui contredit le fait que φ soit gagnante pour Adam).

Ainsi, une partie où Adam respecte φ ne visite pas de sommet final, ce qui signifie exactement que φ est une stratégie gagnante pour Adam, et conclut la preuve.

3.2.4 Conclusion

Le Théorème 3.1 implique la décidabilité des jeux d'accessibilité sur des graphes de processus à pile.

Théorème 3.2 *Pour tout jeu d'accessibilité sur un graphe de processus à pile, on peut décider, pour tout configuration, qui possède une stratégie gagnante. Ce problème peut être résolu en temps exponentiel.*

Preuve. Le Théorème 3.1 donne la solution pour les configurations de pile vide : il suffit pour cela d'utiliser la réduction vers un jeu sur un graphe fini et de résoudre ce jeu. Ce nouveau jeu possède un nombre exponentiel (dans la taille de Q) de sommets et est équipé d'une condition d'accessibilité. Il faut donc un temps exponentiel pour résoudre ce jeu (algorithme polynomial sur un graphe de jeu exponentiel), ce qui répond à la question.

Pour ce qui est du cas d'une configuration quelconque, il suffit de modifier le processus à pile de sorte à ce qu'il commence par atteindre la configuration en question depuis une configuration donnée de pile vide, et qu'ensuite il simule le processus à pile de départ. Ainsi, décider le gagnant depuis la configuration donnée dans le processus à pile de départ équivaut à décider le gagnant depuis une configuration initiale de pile vide dans le nouveau jeu. ■

En fait on a un résultat plus précis (dont on ne donne pas la preuve ici).

Théorème 3.3 *Décider le gagnant dans un jeu d'accessibilité sur un graphe de processus à pile depuis une configuration de pile vide est un problème* EXPTIME*-complet.*

3.3 Résolution des jeux de Parité

La technique présentée pour les jeux d'accessibilité peut être généralisée pour des conditions de gains plus complexes comme le montre le résultat suivant (dont nous ne donnons pas la preuve ici).

Théorème 3.4 *Pour tout jeu de parité sur un graphe de processus à pile, on peut décider, pour tout configuration, qui possède une stratégie gagnante. Ce problème peut être résolu en temps exponentiel en le nombre d'état de contrôle et le nombre de couleurs, et en temps polynomial en la taille de l'alphabet de pile.*

3.4 Résolution des jeux d'explosion

Dans tout ce qui suit on se donne un graphe de jeu engendré par un processus à pile $\mathcal{P} = \langle Q, \Gamma, \perp, \Delta \rangle$ et une partition $Q_{\mathbf{E}} \cup Q_{\mathbf{A}}$ de Q. Soit $\mathcal{G}$ le graphe de jeu engendré par $\mathcal{P}$ et la partition précédente. On note $\mathbb{G}$ (respectivement $\mathbb{G}_s$) le jeu d'explosion (resp. explosion stricte) associé.

3.4.1 Stratégies gagnantes

On prouve ici le résultat suivant.

Théorème 3.5 *Eve possède des stratégies sans mémoire pour gagner dans $\mathbb{G}$. Les ensembles de positions gagnantes sont donc les mêmes dans $\mathbb{G}$ et $\mathbb{G}_s$ et Eve possède donc des stratégies gagnantes sans mémoire dans les deux jeux. Le même résultat est vrai pour Adam.*

Preuve.

On note $W_{\mathbf{E}}$ et $W_{\mathbf{A}}$ les ensembles de positions gagnantes pour Eve et pour Adam dans le jeu $\mathbb{G}$.

Considérons une position gagnante v pour Eve dans $\mathbb{G}$ et une stratégie gagnante φ pour Eve dans $\mathbb{G}$ depuis v. En particulier, pour tout $i \geq 1$, φ est une stratégie gagnante dans le jeu d'accessibilité $\mathbb{G}_i$ vers l'ensemble $Q \times \Gamma^{\geq i}$ des configurations de hauteur de pile supérieure ou égale à i. Si l'on note $W_{\mathbf{E}}^i$ et $W_{\mathbf{A}}^i$ les ensembles de positions gagnantes pour Eve et Adam dans $\mathbb{G}_i$, on a donc $W_E ve_I \subseteq \bigcap_{i \geq 0} W_{\mathbf{E}}^i$. Dans ce qui suit, on prouve l'implication réciproque et on en déduit des stratégies sans mémoire pour Eve dans $\mathbb{G}$.

Soit une configuration $v \in \bigcap_{i \geq 0} W_{\mathbf{E}}^i$. Pour tout $i \geq 1$, la configuration v étant gagnante dans le jeu d'accessibilité $\mathbb{G}_i$, Eve possède une stratégie sans mémoire ψ_i depuis v dans ce jeu[2]. De plus, il est clair que ψ_i est une stratégie sans mémoire gagnante pour Eve depuis v dans les jeux $\mathbb{G}_j$ pour $j \leq i$.

A partir de la suite $(\psi_i)_{i \geq 1}$, il n'est pas difficile de construire une stratégie sans mémoire gagnante dans $\mathbb{G}$ depuis v. Pour cela on ordonne les sommets de $\mathcal{G}$ par taille de pile croissante : on obtient alors une suite $(v_j)_{j \geq 0}$, telle que $\{v_j \mid j \in \mathbb{N}\} = Q \times \Gamma^*$ (les premiers sommets sont ceux de pile vide, puis viennent ceux de hauteur de pile égale à 1 et ainsi de suite). Appelons enfin $I_0 = \mathbb{N}$.

Si $v_0 \in Q_{\mathbf{A}}$, il n'y a rien à faire pour définir ψ dessus (car le sommet n'est pas contrôlé par Adam). Sinon, comme v_0 n'a qu'un nombre **fini** de successeurs possibles, il en possède un, w_0, tel que $\psi_i(v_0) = w_0$ pour une infinité d'indices $i \in I_0$. Appelons I_1 cet ensemble infini d'indices et posons $\psi(v_0) = w_0$. Si $v_0 \in Q_{\mathbf{A}}$, il n'y a rien à faire pour définir ψ dessus, Sinon, comme v_1 n'a qu'un nombre fini de successeurs possibles, il en possède un, w_1, tel que $\psi_i(v_1) = w_1$ pour une infinité d'indices $i \in I_1$. Appelons I_2 cet ensemble infini d'indices et posons $\psi(v_1) = w_1$.

En raisonnant de la sorte, on définit une stratégie sans mémoire ψ, ainsi qu'une suite infinie décroissante d'ensembles infinis d'indices $(I_i)_{i \geq 0}$, tels que pour tout entier k, $\psi(v_j) = \psi_i(v_j)$ pour tout $0 \leq j \leq k$ et tout $i \in I_k$.

Par l'absurde, supposons que ψ soit une stratégie perdante pour Eve dans $\mathbb{G}$ depuis v. Dès lors, il existe une partie Λ dans $\mathbb{G}$ où Eve respecte ψ et dans lequel la hauteur de pile est bornée par un entier N. Appelons k le plus petit indice tel que v_k soit de hauteur de pile égale à $N+1$. En d'autres termes, $\{v_j \mid j < k\} = Q \times \Gamma^{\leq N}$. Considérons enfin un entier i dans I_k tel que $i \geq N+1$ (un tel entier existe puisque I_k est infini). Comme $\psi(v_j) = \psi_i(v_j)$ pour tout $0 \leq j \leq k$ et comme toutes les configurations de Λ appartiennent à $\{v_j \mid j < k\} = Q \times \Gamma^{\leq N}$, Λ peut être vue comme une partie dans laquelle Eve respecte sa stratégie positionnelle ψ_i. Mais, comme ψ_i est gagnante dans le jeu d'accessibilité vers $Q \times \Gamma^{\geq i}$, Λ atteint un jour une configuration de hauteur de pile supérieure ou égale à $i \geq N+1$, ce qui est contradictoire avec le fait que la hauteur de pile soit bornée par N

[2]Ce résultat peut être prouvé en généralisant les résultats sur les jeux d'accessibilité sur des graphes finis aux graphes infinis. Plus précisément en généralisant les notions d'attracteurs et en utilisant une induction transfinie. On prouve alors que les jeux d'accessibilité (en fait le résultat est encore vrai pour des jeux de parité) sur des graphes infinis de degré dénombrable admettent des stratégies sans mémoire.
On peut cependant faire plus simple ici, en remarquant que la structure du graphe est très particulière et en considérant l'attracteur de $Q \times \Gamma^i$ dans le sous-graphe $\mathcal{G}[Q \times \Gamma^{\leq i}]$

dans Λ.

Dès lors, on en conclut donc que ψ est une stratégie sans mémoire gagnante pour Eve depuis v dans $\mathbb{G}$.

Par dualité, un sommet v est gagnant pour Adam s'il existe un i tel que v soit gagnant pour Adam dans le jeu $\mathbb{G}_i$. De plus une stratégie pour Adam dans $\mathbb{G}_i$ est aussi gagnante dans $\mathbb{G}$, et on peut donc la choisir sans mémoire.

Les ensembles gagnants dans $\mathbb{G}$ et $\mathbb{G}_s$ coïncident car le résultat précédent prouve que si Eve est capable d'assurer l'explosion, elle peut le faire en utilisant une stratégie sans mémoire. Cela implique que chaque sommet ne sera visité qu'au plus une fois dans une partie où elle suit une telle stratégie (sinon Adam peut forcer la partie à boucler infiniment et donc à être de hauteur de pile bornée). En particulier, dans une telle partie, la pile explose de façon stricte. ■

3.4.2 Réduction vers un jeu de sûreté

Le Théorème 3.5 implique que l'on peut raisonner uniquement sur la condition d'explosion stricte. De plus, on peut supposer que les deux joueurs suivent des stratégies sans mémoire. En particulier, s'il y a une boucle qui est effectuée, on peut considérer que la partie va boucler infiniment (les joueurs suivent des stratégies sans mémoire) et donc qu'Eve va perdre. Si aucune boucle n'est effectuée, on visite une infinité de configurations distinctes et donc la hauteur de pile converge vers $+\infty$ (car il n'y a qu'un nombre fini de configurations de hauteur inférieure ou égale à une borne donnée).

Comme il n'est pas aisé de détecter des boucles dans le cas général, on va se restreindre à un type particulier que l'on qualifie de bosse bouclante.

Définition 3.9 *Soit $\lambda = v_0 v_1 \cdots$ une partie dans $\mathcal{G}$. Un facteur $v_k \cdots v_{k+h}$ est une* bosse bouclante *si $v_k = v_{k+h}$ et $|v_k| \leq |v_{k+i}|$ pour tout $0 \leq i \leq h$.*

On considère maintenant la condition de gain Ω' sur $\mathcal{G}$ donnée par
$\Omega' = \{\lambda \mid \lambda \text{ ne contient pas de bosse bouclante}\}$, et on note $\mathbb{G}' = (\mathcal{G}, \Omega')$.

On a alors l'équivalence suivante

Proposition 3.4 *Les jeux $\mathbb{G}$, $\mathbb{G}_s$ et $\mathbb{G}'$ ont les mêmes ensembles de positions gagnantes. De plus, Eve et Adam possèdent tous deux des stratégies sans mémoire dans $\mathbb{G}'$.*

Preuve. Il est clair qu'une position gagnante pour Eve dans $\mathbb{G}$ est gagnante pour Eve dans $\mathbb{G}'$, car une stratégie gagnante sans mémoire pour Eve assure qu'il n'y a pas de boucle dans une partie où Eve la respecte, et donc en particulier pas de bosse bouclante. Réciproquement, une stratégie gagnante pour Eve dans $\mathbb{G}'$ assure que tout niveau de pile est finiment souvent visité. En effet, si tel n'est pas le cas, il existe une plus petite hauteur de pile infiniment souvent visitée, et à partir d'un moment cette hauteur de pile est la plus petite visitée (par minimalité) : comme il n'y a qu'un nombre fini d'états de contrôle, il y a une configuration qui est infiniment souvent visitée sur ce niveau, ce qui implique en particulier qu'il y a une bosse bouclante. ■

L'idée dans ce qui suit est de simuler le jeu $\mathbb{G}'$ par un jeu sur un graphe de processus à pile plus grand qui simule $\mathcal{P}$ dans lequel on détecte les bosses bouclantes. Le nouveau processus à pile sera noté $\mathcal{P}''$, le graphe de jeu qu'il engendre $\mathcal{G}''$ et le jeu de sûreté qui sera associé sera noté $\mathbb{G}''$. De plus $\mathcal{P}''$ pourra effectuer des transitions de type $rew(q, b)$ qui signifie que l'on va dans un état q et que l'on réécrit le sommet de pile en b. Il est alors facile de voir que l'on peut adapter la preuve du Théorème 3.2 pour obtenir un résultat en tout point similaire quand on considère des jeux engendrés par des processus à pile autorisant une telle réécriture du sommet de pile.

L'alphabet de pile de $\mathcal{P}''$ est $\Gamma'' = \Gamma \times 2^Q$: la seconde composante sert à stocker les états de contrôles déjà vu avec le contenu de pile courant. Comme cette information doit être mise à jour, les états états de contrôle de $\mathcal{P}$ seront $Q'' = \{q, q_{up}, q_{bad} \mid q \in Q\}$: l'état q_{up} signifie que l'on simule l'état q mais qu'il reste à mettre à jour l'information sur les états de contrôle déjà visités, et l'état q_{bad} signifie que l'on simule q, que l'information est à jour mais que l'on vient d'effectuer une bosse bouclante. La fonction de transition Δ'' simule Δ et réalise la mise à jour de l'information sur les états visités ainsi que la détection des bosses bouclantes. Plus précisément,

- $\Delta''(p, (a, S)) = \Delta''(p_{bad}, (a, S))$ contiennent exactement les éléments suivants
  - $push(q, (b, \{q\}))$ ssi $push(q, b) \in \Delta(p, a)$ (on initialise l'ensemble des états visités à $\{q\}$).
  - $rew(q, (b, S \cup \{q\}))$ ssi $skip(q, b) \in \Delta(p, a)$ et $q \notin S$ (pas de bosse bouclante et mise à jour de S).
  - $rew(q_{bad}, (b, S))$ ssi $skip(q, b) \in \Delta(p, a)$ et $q \in S$ (bosse bouclante détectée).
  - $pop(q_{up})$ ssi $pop(q) \in \Delta(p, a)$ (on va devoir mettre à jour l'information sur les états visités et éventuellement détecter une bosse bouclante).
- $\Delta''(q_{up}, (a, S)) = \{rew(q, (a, S \cup \{q\}))\}$ si $q \notin S$ (mise à jour de S) et $\Delta''(q_{up}, (a, S)) = \{rew(q_{bad}, (a, S))\}$ sinon (bosse bouclante détectée).

On partitionne Q'' en $Q''_{\mathbf{E}} \cup Q''_{\mathbf{A}}$ en posant $Q''_{\mathbf{E}} = \{q, q_{up}, q_{bad} \mid q \in Q_{\mathbf{E}}\}$ Enfin, le jeu $\mathbb{G}''$ dénote le jeu sur $\mathcal{G}''$

On a alors le résultat suivant :

Proposition 3.5 *Pour tout $q \in Q$, Eve a une stratégie gagnante dans $\mathbb{G}'$ depuis $(q, \perp)$ ssi elle a une stratégie gagnante dans $\mathbb{G}''$ depuis $(q, (\perp, \{q\}))$.*

Preuve. A partir d'une stratégie gagnante sans mémoire φ' pour Eve dans $\mathbb{G}'$ on construit une stratégie gagnante φ'' pour Eve dans $\mathbb{G}''$. Dans une configuration d'état $q_u p$, il n'y a qu'une transition possible donc φ'' est trivialement définie. Sinon, φ'' simule le coup donné par φ' sur la configuration obtenu en ignorant les informations sur les états visité et en remplaçant éventuellement q_{bad} par q. A une partie $\lambda'' = v''_0 v''_1 \cdots$ respectant φ'' on associe une partie $\lambda' = v'_0 v'_1 \cdots$ où $v'_i = \varepsilon$ (le mot vide) ssi v''_i a un état de contrôle de la forme q_{up} et où $v'_i = (q, \tau(u))$ si l'état de v''_i est q ou q_{bad} et où τ est le morphisme de $\Gamma \times 2^Q$ dans Γ défini par $\tau((a, S)) = a$ pour tout $(a, S) \in \Gamma \times 2^Q$. Il est alors clair que λ' est une partie dans $\mathbb{G}'$ où Eve respecte φ'. En particulier il n'y a pas de bosse bouclante et donc on n'atteint jamais d'état q_{bad} dans λ''.

La même construction peut être faite à partir d'une stratégie d'Adam, ce qui termine la preuve. ■

3.4.3 Complexité

Concernant la complexité, on a le résultat suivant.

Théorème 3.6 *Décider le gagnant dans un jeu d'explosion est un problème* EXPTIME*-complet.*

Preuve. Les réductions précédentes n'augmentent de façon exponentielle que la taille de l'alphabet de pile. Comme résoudre un jeu de sûreté est polynomial dans la taille de l'alphabet de pile et exponentiel dans la taille de l'ensemble des états de contrôle, il s'en suit que le jeu $\mathbb{G}''$ se résout en temps exponentiel en Q et polynomial en Γ, ce qui termine la preuve. ■

3.5 Ensembles de positions gagnantes

3.5.1 $\mathcal{P}$-automates

L'objet de cette section est de calculer des ensembles de positions gagnantes dans des jeux sur des graphes d'automates à pile, c'est à dire une partie de $Q \times \Gamma^*$. Les automates permettent de représenter des langages rationnels et une représentation naturelle d'une partie P de $Q \times \Gamma^*$ est donnée par $(L_q)_{q \in Q}$ où $L_q = \{u \in \Gamma^* \mid (q, u) \in P\}$. Dès lors, si L_q est rationnel pour tout $q \in Q$, la partie P peut être représentée par un ensemble fini d'automates $(\mathcal{A}_q)_{q \in Q}$. Dans ce cas, on dira que l'ensemble P est **régulier**.

Les $\mathcal{P}$-automates sont des objets naturels pour reconnaître de tels ensembles. Nous les définissons dans le cadre le plus général, à savoir celui de l'alternance.

Définition 3.10 *Un $\mathcal{P}$-**automate** est un quintuplet $\mathcal{A} = \langle Q, \Gamma, I, F, \delta \rangle$, où Q est un ensemble fini d'états, A est un alphabet fini, $I \subseteq Q$ est un sous-ensemble d'états de Q appelé ensemble des états initiaux, $F \subseteq Q$ est un sous-ensemble de Q appelé ensemble des états finaux, et δ est une application de $Q \times \Gamma$ dans $\mathcal{P}(Q)$, appelée fonction de transition.*

La différence vient du fait qu'un $\mathcal{P}$-automate accepte des couples de la forme (q, u) où $q \in I$ et $u \in \Gamma^*$. Pour cela, le calcul de l'automate commence en q et ensuite se comporte normalement. Ainsi, la première composante du couple à reconnaître détermine l'état initial du calcul qui se déroule ensuite sur la seconde composante.

Définition 3.11 *Considérons un $\mathcal{P}$-automate $\mathcal{A} = \langle Q, \Gamma, I, F, \delta \rangle$. Considérons un mot $u = a_0 a_2 \cdots a_n$ sur l'alphabet Γ et un état $q \in I$. Un calcul de $\mathcal{A}$ depuis un état p sur u est une suite d'états $q_0, \cdots q_n$ vérifiant les propriétés suivantes :*

- *$q_0 = p$.*
- *pour tout $i 0 \leq i \leq n-1$, $q_{i+1} \in \delta(q_i, a_i)$.*

*Enfin, un **calcul** de $\mathcal{A}$ sur le couple (q, u) est un calcul de $\mathcal{A}$ depuis l'état initial q.*

Précisons enfin comment sont définies les configurations acceptées par un $\mathcal{P}$-automate alternant.

Définition 3.12 *Considérons un $\mathcal{P}$-automate $\mathcal{A} = \langle Q, \Gamma, I, F, \delta \rangle$ et un couple $(q, u) \in I \times \Gamma^*$. Un calcul de $\mathcal{A}$ sur (q, u) est **acceptant** s'il se termine par un état final. Un couple (q, u) est* *accepté par un automate alternant $\mathcal{A}$ s'il existe un calcul acceptant de $\mathcal{A}$ sur (q, u). On note $L(\mathcal{A})$ l'ensemble des couples acceptés par $\mathcal{A}$. Une partie P de $I \times A^*$ est reconnue par $\mathcal{A}$ si tout couple (q, u) de P est accepté par $\mathcal{A}$.*

Il est alors facile d'établir le résultat suivant.

Proposition 3.6 *Soit Q un ensemble fini et soit Γ un alphabet fini. L'ensemble des parties régulières de $Q \times \Gamma^*$ est exactement l'ensemble des parties reconnues par des $\mathcal{P}$-automates alternants (ou non déterministes).*

3.5.2 Deux propriétés sur les conditions de gain

Imaginons maintenant un jeu sur un graphe de processus à pile avec la condition de gain suivante : Eve remporte la partie à condition qu'à aucun moment la pile ne contienne un b. Imaginons de surcroît qu'Eve peut, depuis une position $(p, a\perp)$, faire infiniment la boucle $(p, a\perp) \cdot (p, aa\perp) \cdot (p, a\perp)$. Ainsi, Eve possède une stratégie gagnante depuis $(p, a\perp)$ qui ne vide jamais la pile. Cependant, il est clair que toute les positions de la forme $(p, au\perp)$ ne sont pas gagnantes (songez par exemple à $(p, ab\perp)$). Le raisonnement précédent ne tient donc plus et cela pour une raison simple : les parties gagnantes ne sont pas invariantes par ajout d'un mot en fond de pile dans toutes les positions de la partie.

On en vient donc à introduire les définitions et les propriétés suivantes (qui sont illustrées par les figures 3.7 et 3.8) :

Définition 3.13 *Soit Γ un alphabet de pile, de symbole de fond de pile $\perp$, et soit Q un ensemble d'états. Soit u un mot sur l'alphabet $(\Gamma \setminus \{\perp\})$. On définit l'application τ_u de $Q \times (\Gamma \setminus \{\perp\})^+\perp$ dans lui-même, par $\tau_u(q, w\perp) = (q, wu\perp)$ pour tout état $q \in Q$ et tout mot $w \in (\Gamma \setminus \{\perp\})^+$. On l'étend alors en un endomorphisme lettre à lettre de $(Q \times (\Gamma \setminus \{\perp\})^+\perp)^*$. Ainsi, si l'on considère un jeu sur un graphe de processus à pile d'alphabet de pile Γ, τ_u associe à toute partie où la pile n'est jamais vidée la partie (valide) obtenue en ajoutant, dans toutes les configurations, u au fond de la pile.*

Définition 3.14 *Soit Γ un alphabet de pile, de symbole de fond de pile $\perp$, et soit Q un ensemble d'états. Soit u un mot sur l'alphabet $(\Gamma \setminus \{\perp\})$. On introduit l'application $\tau_{u^{-1}}$ de $Q \times (\Gamma \setminus \{\perp\})^+u\perp$ dans $Q \times (\Gamma \setminus \{\perp\})^+\perp$, définie par $\tau_{u^{-1}}(q, wu\perp) = (q, w\perp)$ pour tout état $q \in Q$ et tout mot $w \in (\Gamma \setminus \{\perp\})^+$. Ainsi, $\tau_{u^{-1}}$ associe à toute configuration contenant u en fond de pile avec des lettres au-dessus, la configuration obtenue en supprimant u.*

On étend alors $\tau_{u^{-1}}$ en un morphisme de $(Q \times (\Gamma \setminus \{\perp\})^+u\perp)^*$ dans $(Q \times (\Gamma \setminus \{\perp\})^+\perp)^*$. Ainsi, si l'on considère un jeu sur un graphe de processus à pile d'alphabet de pile Γ, $\tau_{u^{-1}}$ associe à toute partie où u est toujours présent dans la pile avec des lettres au-dessus de lui la partie (valide), obtenue en supprimant u du fond de la pile de toutes les configurations.

Définition 3.15 *Soit Ω une condition de gain pour un jeu sur un graphe de processus à pile et soit Γ l'alphabet de pile sous-jacent à Ω. On dit que Ω est **invariante par translations verticales** si et seulement si pour toute partie $\Lambda = v_0v_1v_2\cdots$ de Ω on a :*

1. **Invariance par translation vers le haut.** *Si Λ ne visite pas de configuration de pile vide, pour tout mot fini $u \in \Gamma^*$ ne contenant pas le fond de pile, $\tau_u(\Lambda)$ est dans Ω.*

2. **Invariance par translation vers le bas.** *Pour tout mot fini $u \in \Gamma^*$ ne contenant pas le fond de pile, et tel que pour toute position $v_i = (p_i, u_i\bot)$ visitée dans Λ, u est un suffixe strict de u_i, $\tau_{u^{-1}}(\Lambda)$ est dans Ω.*

Les figures 3.7 et 3.8 donnent une interprétation graphique de ces invariances.

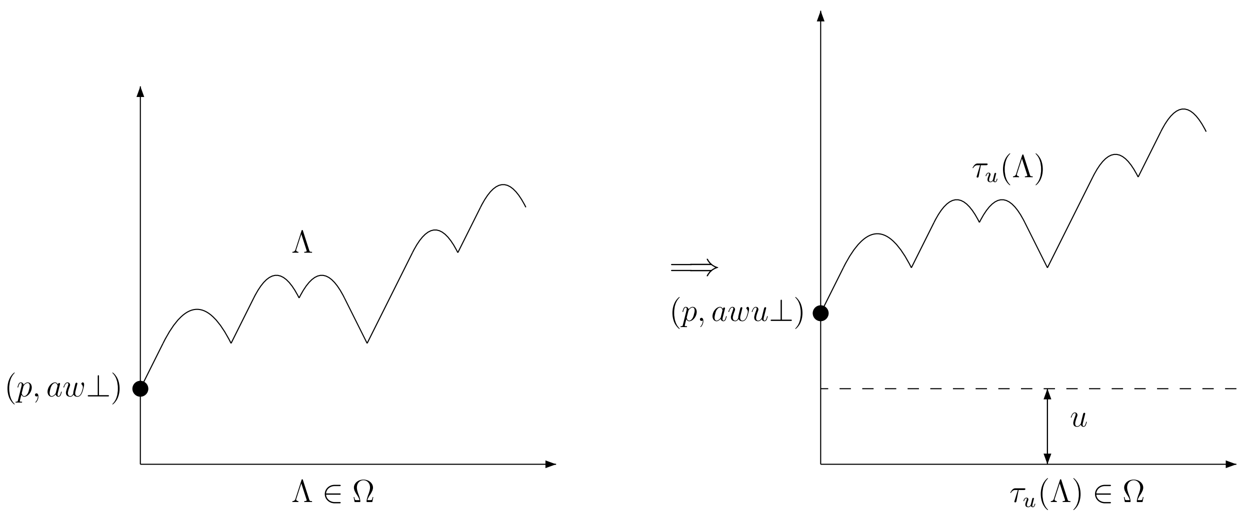


FIG. 3.7 – Invariance par translation vers le haut

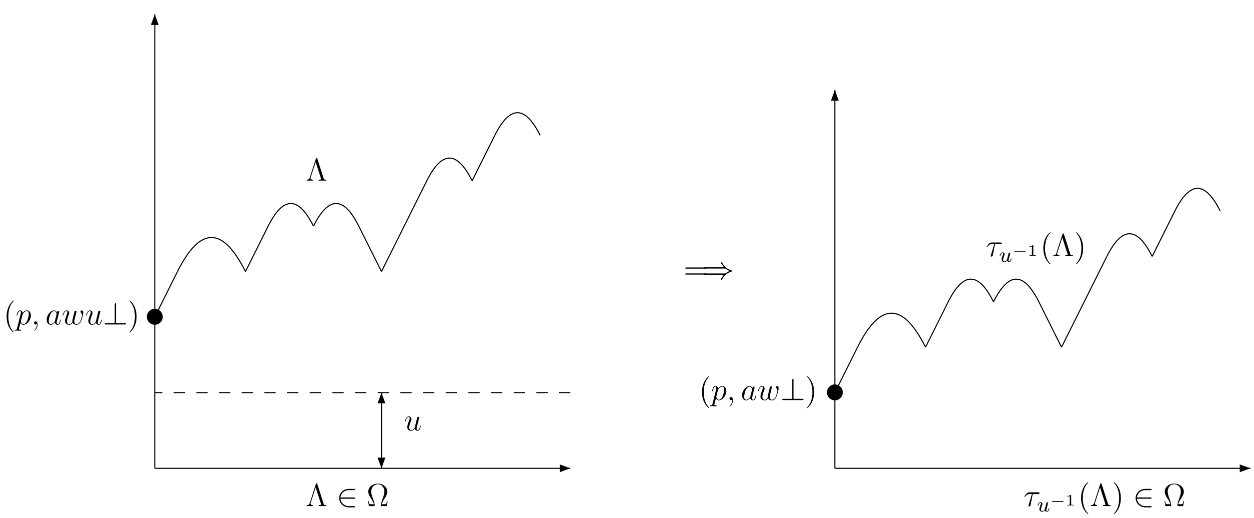


FIG. 3.8 – Invariance par translation vers le bas

Exemple 3.3 Il est facile de voir que toute condition de gain qui ne considère pas le contenu de la pile, c'est-à-dire qui ne dépend que de la suite des états de contrôle, est invariante par translations horizontales. C'est le cas en particulier des conditions d'accessibilité, de Büchi, de parité, ou de Muller portant sur les états de contrôle, mais aussi de conditions plus exotiques qui peuvent, par exemple demander, que la suite des états de contrôle appartienne à un langage (quelconque).

D'autres conditions naturelles pour les jeux sur un graphe de processus à pile sont celles concernant le bornage (ou le non-bornage) de la pile. Elles vérifient l'invariance par translations verticales.

Dans la suite, on va s'intéresser à la composition des stratégies. Sans entrer dans le détail, il s'agit de suivre une stratégie, jusqu'à atteindre une certaine position à partir de laquelle on suit une autre stratégie qui assure la victoire. L'idée sous-jacente est la suivante : si l'on est capable de forcer à atteindre une position gagnante, alors la position de départ est gagnante. C'est le principe de base de la notion de point fixe (ou d'attracteur) utilisée pour résoudre les jeux munis d'une condition d'accessibilité ou de Büchi. La propriété des conditions de gain pour pouvoir enchaîner les parties partielles sans changer l'issue de la partie est leur invariance par ajout ou suppression d'un préfixe de partie. Plus formellement, on définit la propriété suivante.

Définition 3.16 *Soit Ω une condition de gain pour un jeu sur un graphe de processus à pile, de sommets V. On dit que Ω est* ***invariante par translations horizontales*** *si et seulement si, pour toute partie Λ dans Ω, on a :*

1. **Invariance par translation vers la gauche.** *Tout suffixe de Λ est dans Ω.*
2. **Invariance par translation vers la droite.** *Toute partie de $V^*\Lambda$ est dans Ω.*

Les figures 3.9 et 3.10 donnent une interprétation graphique de ces invariances.

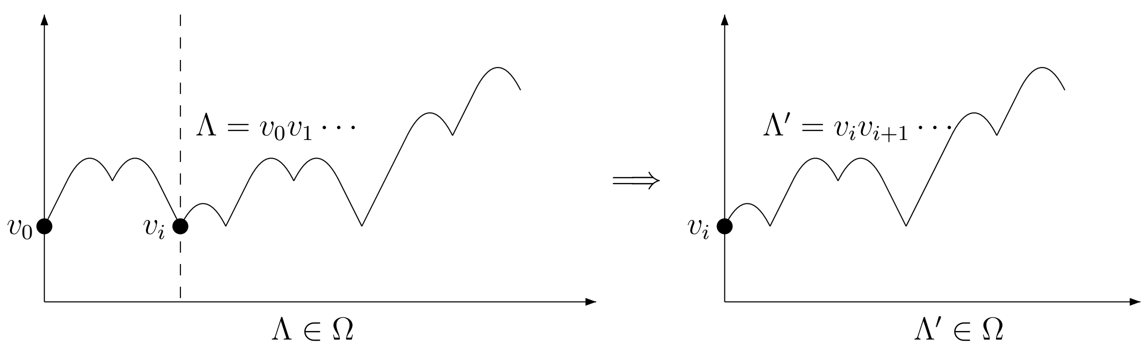


FIG. 3.9 – Invariance par translation vers la gauche

Exemple 3.4 Toute condition de gain ne parlant que d'un comportement à l'infini est invariante par translations horizontales. C'est la cas par exemple des conditions sur l'ensemble des états infiniment répétés comme les conditions de Büchi, de parité, ou plus généralement de Muller.

Les conditions concernant le bornage (ou le non-bornage) de la pile vérifient l'invariance par translations horizontales.

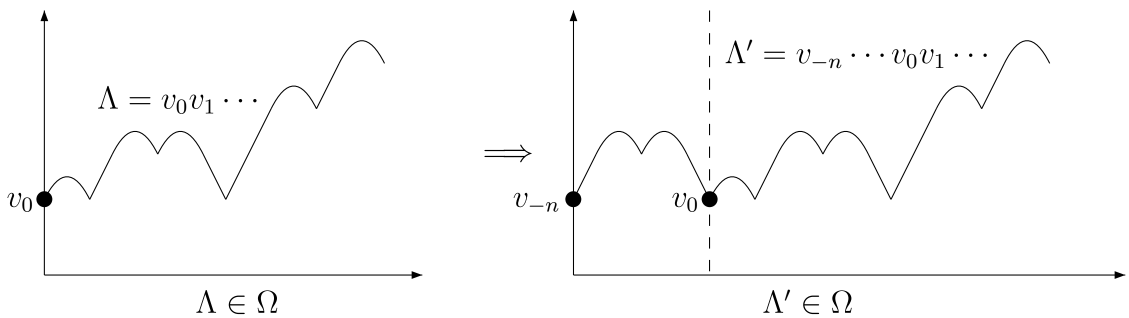


FIG. 3.10 – Invariance par translation vers la droite

En revanche, la condition d'accessibilité (respectivement la condition de sûreté) n'est pas invariante par translation vers la gauche (respectivement vers la droite).

3.5.3 Jeux conditionnés et ensembles de retour

On s'intéresse à la question suivante : Eve peut-elle gagner tout en assurant que la pile ne soit jamais vidée ? Plus exactement, la question est de savoir si, pour une lettre a en sommet de pile, Eve peut gagner tout en assurant que ce a ne sera jamais dépilé. Si cela n'est pas possible, Eve aura tout intérêt à jouer de la façon suivante : soit forcer le gain sans dépiler a, soit, si elle est contrainte à dépiler a, atteindre en contrepartie un état favorable. Pour formaliser ces intuitions, introduisons les notions de **jeu conditionné** et d'**ensemble de retour**.

Définition 3.17 (Jeu conditionné) *Soit $\mathcal{G} = (G, V_{\mathbf{E}}, V_{\mathbf{A}})$ un graphe de jeu sur un graphe $G = (V, E)$ engendré par un processus à pile $\mathcal{P} = \langle Q, \Gamma, \perp, \Delta\rangle$. Soit $\mathbb{G} = (\mathcal{G}, \Omega)$ un jeu sur $\mathcal{G}$. Pour tout sous-ensemble R de Q, on considère une nouvelle condition de gain, notée $\Omega(R)$ et définie par :*

$$\Omega(R) = (\Omega \setminus V^*(Q \times \{\perp\})V^\omega) \cup (V \setminus (Q \times \{\perp\}))^*(R \times \{\perp\})V^\omega$$

En d'autres termes, les parties gagnantes pour $\Omega(R)$ sont :
- *les parties de Ω dans lesquelles aucune configuration de pile vide n'est visitée.*
- *les parties qui visitent une configuration de pile vide et telles que l'état de contrôle, dans la première configuration de pile vide visitée, appartient à R.*

*Le jeu $\mathbb{G}(R) = (\mathcal{G}, \Omega(R))$ est appelé **jeu conditionné** par R.*

Définition 3.18 (Ensemble de retour) *Soit un jeu $\mathbb{G} = (\mathcal{G}, \Omega)$ sur un graphe de jeu engendré par un processus à pile $\mathcal{P} = \langle Q, \Gamma, \perp, \Delta\rangle$. Soit $p \in Q$ un état de contrôle et soit $a \in \Gamma$ une lettre différente de $\perp$. Soit $R \subseteq Q$, un sous-ensemble d'états de contrôle. On dit que R est un **ensemble de retour** pour (p, a) dans $\mathbb{G}$, si Eve possède une stratégie gagnante dans le jeu $\mathbb{G}(R)$ depuis $(p, a\perp)$. On note $\mathcal{R}(p, a)$ l'ensemble des ensembles de retour pour (p, a).*

On a alors facilement la propriété suivante :

Propriété 3.1 *Soit un jeu $\mathbb{G} = (\mathcal{G}, \Omega)$ sur un graphe de jeu engendré par un processus à pile $\mathcal{P} = \langle Q, \Gamma, \bot, \Delta \rangle$. Soit $p \in Q$ et soit $a \in \Gamma$. Si R est un ensemble de retour pour (p, a), et si $R' \supseteq R$, alors R' est un ensemble de retour pour (p, a).*

Preuve. En effet, si $R \subseteq R'$, $\Omega(R) \subseteq \Omega(R')$ ■

3.5.4 Cas particulier des conditions invariantes par translations

Dans tout ce paragraphe, on se donne un jeu $\mathbb{G} = (\mathcal{G}, \Omega)$ sur un graphe de jeu engendré par un processus à pile $\mathcal{P} = \langle Q, \Gamma, \bot, \Delta \rangle$ et une partition $Q_{\mathbf{E}} \cup Q_{\mathbf{A}}$ de Q. On suppose de plus que Ω est invariante par translations verticales et horizontales.

Ensembles de retour et positions gagnantes

On commence par le cas particulier où Eve peut gagner sans dépiler :

Lemme 3.1 *Soit p un état de contrôle de Q et soit a une lettre de Γ différente de $\bot$. Si $\emptyset \in \mathcal{R}(p, a)$, alors pour tout mot u sur l'alphabet $(\Gamma \setminus \{\bot\})$, $(p, au\bot)$ est une position gagnante pour Eve dans le jeu $\mathbb{G} = (\mathcal{G}, \Omega)$.*

Preuve. Soit $\varphi_\emptyset$ une stratégie gagnante pour Eve dans le jeu $\mathbb{G}(\emptyset)$ depuis $(p, a\bot)$. On définit à partir de $\varphi_\emptyset$ une stratégie φ pour Eve dans $\mathbb{G}$ depuis $(p, au\bot)$. Soit $\Lambda = v_0 v_1 v_2 \cdots v_n$ une partie partielle. On définit alors $\varphi(\Lambda)$ de la façon suivante :

- si toute configuration v_i apparaissant dans Λ est de la forme $v_i = (p_i, u_i u\bot)$, où u_i est non vide, on définit alors $\varphi(\Lambda) = \tau_u(\varphi_\emptyset(\tau_{u^{-1}}(\Lambda)))$.
- sinon, $\varphi(\Lambda)$ n'est pas défini.

On montre que si Λ est une partie partielle dans $\mathbb{G}$, commençant en $(p, au\bot)$, où Eve respecte φ, alors toutes les configurations visitées dans Λ sont de la forme $v_i = (p_i, u_i u\bot)$, où u_i est non vide (en particulier, cela implique que si Eve respecte φ, $\varphi(\Lambda)$ est toujours défini) et de plus, $\tau_{u^{-1}}(\Lambda)$ est une partie partielle dans $\mathbb{G}(\emptyset)$ commençant en $(p, a\bot)$ où Eve respecte $\varphi_\emptyset$. Pour cela, on procède par induction :

- la propriété est vraie si $\Lambda = (p, au\bot)$.
- supposons maintenant qu'elle soit vraie pour une partie partielle Λ, et montrons qu'elle l'est encore pour un partie Λ' prolongeant Λ d'une configuration.
  Commençons par le cas où Λ se termine dans une position d'Eve. Comme, par hypothèse d'induction, $\tau_{u^{-1}}(\Lambda)$ est une partie dans $\mathbb{G}(\emptyset)$, où Eve respecte $\varphi_\emptyset$, $\varphi_\emptyset(\tau_{u^{-1}}(\Lambda))$ est une configuration où la pile n'est pas vide et dès lors $\varphi(\Lambda) = \tau_u(\varphi_\emptyset(\tau_{u^{-1}}(\Lambda)))$ est de la forme $(p_i, u_i u\bot)$, où u_i est non vide. De plus, il est clair que $\tau_{u^{-1}}(\Lambda')$ est une partie partielle dans $\mathbb{G}(\emptyset)$, où Eve respecte $\varphi_\emptyset$.
  Considérons le cas où Λ se termine par une position d'Adam. Les dernières positions de Λ et de $\tau_{u^{-1}}(\Lambda)$ ont le même état de contrôle et le même sommet de pile. Dès lors, la transition appliquée par Adam une fois Λ jouée est également applicable dans la dernière position de $\tau_{u^{-1}}(\Lambda)$. Comme, par hypothèse d'induction, $\tau_{u^{-1}}(\Lambda)$

est une partie dans $\mathbb{G}(\emptyset)$ où Eve respecte $\varphi_\emptyset$, le prolongement de $\tau_{u^{-1}}(\Lambda)$ ainsi obtenu est aussi une partie partielle conforme à $\varphi_\emptyset$. La configuration ainsi atteinte n'est donc pas de pile vide. Dès lors, la dernière configuration de Λ' est bien de la forme $(p_i, u_i u\bot)$, où u_i est non vide.

En conclusion, si Λ est une partie dans $\mathbb{G}$ où Eve respecte φ, alors $\tau_{u^{-1}}(\Lambda)$ est une partie dans $\mathbb{G}(\emptyset)$ depuis $(p, a\bot)$, où Eve respecte $\varphi_\emptyset$. Or, $\varphi_\emptyset$ est gagnante, et donc $\tau_{u^{-1}}(\Lambda)$ est gagnante. Dès lors, $\Lambda = \tau_u(\tau_{u^{-1}}(\Lambda))$ est également gagnante.

Ainsi, φ est une stratégie gagnante pour Eve dans $\mathbb{G}$ depuis la position $(p, au\bot)$. ■

De façon symétrique, on a le résultat suivant :

Lemme 3.2 *Soit p un état de contrôle de Q et soit a une lettre de Γ différente de $\bot$. Si $\mathcal{R}(p, a) = \emptyset$, alors pour tout mot u sur l'alphabet $(\Gamma \setminus \{\bot\})$, $(p, au\bot)$ est une position gagnante pour Adam dans le jeu $\mathbb{G} = (\mathcal{G}, \Omega)$.*

Que se passe-t-il maintenant pour les couples $(p, a) \in Q \times (\Gamma \setminus \{\bot\})$ pour lesquels $\mathcal{R}(p, a)$ ne contient pas $\emptyset$ ou n'est pas vide ? Considérons un tel couple (p, a) et un élément $R \in \mathcal{R}(p, a)$. Eve possède une stratégie gagnante φ_R dans le jeu $\mathbb{G}(R)$ depuis $(p, a\bot)$. A partir de cette stratégie, on peut, comme dans la preuve du lemme 3.1, construire une stratégie (partiellement définie) pour Eve dans $\mathbb{G}$, depuis n'importe quelle configuration de la forme $(p, au\bot)$. Une telle stratégie ne permet pas d'assurer que le a ne sera jamais dépilé et n'est plus définie lorsqu'une telle chose se produit. Cependant, dans le cas où l'on ne dépile pas le a on gagne et sinon on atteint une configuration de la forme $(r, u\bot)$ avec $r \in R$. Pour l'emporter, il faut alors être capable de gagner depuis cette position. Plus formellement on a l'équivalence suivante qui généralise les lemmes 3.1 et 3.2 :

Lemme 3.3 *Soit p un état de contrôle de Q et soit a une lettre de Γ différente de $\bot$. Soit u un mot sur l'alphabet $(\Gamma \setminus \{\bot\})$, alors $(p, au\bot)$ est une position gagnante pour Eve dans le jeu $\mathbb{G}$, si et seulement s'il existe $R \in \mathcal{R}(p, a)$ tel que $(r, u\bot)$ soit gagnant pour Eve dans $\mathbb{G}$ pour tout $r \in R$.*

Preuve. Implication réciproque : On commence par démontrer la réciproque, qui est une généralisation des lemmes 3.1 et 3.2. On suppose donc qu'il existe $R \in \mathcal{R}(p, a)$ tel que $(r, u\bot)$ est gagnant dans $\mathbb{G}$ pour tout $r \in R$. Soit φ_R une stratégie gagnante pour Eve depuis $(p, a\bot)$ dans le jeu $\mathbb{G}(R)$. Soient $(\varphi_r)_{r \in R}$ des stratégies gagnantes dans $\mathbb{G}$ depuis les positions $((r, u\bot))_{r \in R}$.

On définit une stratégie φ pour Eve dans le jeu $\mathbb{G}$ depuis la position $(p, au\bot)$. Soit $\Lambda = v_0 v_1 v_2 \cdots v_n$ une partie partielle, $\varphi(\Lambda)$ est alors défini de la façon suivante :

- si toute configuration v_i apparaissant dans Λ est de la forme $(p_i, u_i u\bot)$, où u_i est non vide, on pose $\varphi(\Lambda) = \tau_u(\varphi_R(\tau_{u^{-1}}(\Lambda)))$. Ce cas correspond à une partie Λ où a n'a pas été dépilé.
- s'il existe un entier j tel que $v_j = (r, u\bot)$ pour un certain $r \in R$ et tel que, pour tout $i < j$, v_i est de la forme $(p_i, u_i u\bot)$, on pose $\varphi(\Lambda) = \varphi_r(v_j v_{j+1} \cdots v_n)$. Dans ce cas, on a dépilé le a juste après v_{j-1}.
- sinon $\varphi(\Lambda)$ n'est pas défini.

En raisonnant comme dans la preuve du lemme 3.1, on prouve que si Λ est une partie partielle dans $\mathbb{G}$ commençant en $(p, au\bot)$, où Eve respecte φ, et telle que toutes les configurations visitées dans Λ sont de la forme $(p_i, u_i u\bot)$, où u_i est non vide, alors :

– $\tau_{u^{-1}}(\Lambda)$ est une partie partielle dans $\mathbb{G}(R)$ commençant en $(p, a\bot)$, où Eve respecte φ_R, et dans laquelle la pile n'est jamais vidée.
– si Λ se prolonge en une partie où le a est dépilé, alors la configuration ainsi atteinte est de la forme $(r, u\bot)$ avec $r \in R$.

Dès lors, si Eve respecte φ, $\varphi(\Lambda)$ est toujours défini. Afin de montrer que la stratégie φ est gagnante, considérons une partie Λ respectant φ. La partie Λ peut être de deux formes :

– dans Λ, toutes les configurations visitées sont de la forme $(p_i, u_i u\bot)$, où u_i est non vide. La partie Λ est donc l'image par τ_u d'une partie Λ' dans $\mathbb{G}(R)$ où la pile n'est jamais vidée et où Eve respecte φ_R. Ainsi, Λ' est dans $\Omega(R)$ et donc dans Ω (car la pile n'est jamais vidée). Dès lors, Λ est également dans Ω
– $\Lambda = \Lambda' \cdot \Lambda_r$, où Λ_r est une partie dans $\mathbb{G}$, commençant par $(r, u\bot)$ pour un certain $r \in R$. De plus, Eve respecte φ_r dans Λ_r, ce qui implique que Λ_r est dans Ω. Dès lors, on en conclut que Λ est gagnante pour Eve.

La stratégie φ est donc bien gagnante pour Eve dans $\mathbb{G}$ depuis $(p, au\bot)$.

Implication directe : Soit φ une stratégie gagnante pour Eve dans $\mathbb{G}$ depuis $(p, au\bot)$. Considérons l'ensemble $\mathcal{L}_\varphi$ des parties possibles dans $\mathbb{G}$ depuis $(p, au\bot)$, où Eve respecte φ. On définit alors un sous-ensemble R de Q de la façon suivante : pour tout état $r \in Q$, $r \in R$ s'il existe une partie Λ de $\mathcal{L}_\varphi$ de la forme $\Lambda' \cdot (r, u\bot) \cdot \Lambda''$, et où toutes les configurations dans Λ' sont de la forme $(p_i, v_i u\bot)$, où v_i est un mot non vide. Ainsi, R est l'ensemble des états de contrôle pouvant être atteints juste après avoir dépilé a dans une partie où Eve respecte φ.

Montrons que R est un ensemble de retour pour (p, a). Pour cela, on définit une stratégie φ_R pour Eve dans $\mathbb{G}(R)$ depuis $(p, a\bot)$. Pour toute partie partielle Λ dans $\mathbb{G}(R)$ depuis $(p, a\bot)$, on définit $\varphi_R(\Lambda)$ comme suit :

– si dans Λ aucune configuration de pile vide n'est visitée, $\varphi_R(\Lambda) = \tau_{u^{-1}}(\varphi(\tau_u(\Lambda)))$.
– si dans Λ une configuration de pile vide est visitée, $\varphi_R(\Lambda)$ peut prendre n'importe quelle valeur parmi celles qui sont possibles.

On a alors le résultat suivant : pour toute partie partielle Λ dans $\mathbb{G}(R)$ commençant en $(p, a\bot)$ où Eve respecte φ_R, on a :

1. si dans Λ la pile n'est jamais vidée, $\tau_u(\Lambda)$ est une partie partielle dans $\mathbb{G}$ commençant en $(p, au\bot)$, où Eve respecte φ.
2. si dans Λ une configuration de pile vide est visitée, alors $\Lambda = \Lambda' \cdot (r, \bot) \cdot \Lambda''$, où Λ' est une partie partielle dans laquelle la pile n'est jamais vide, et où $r \in R$.

Pour établir le résultat précédent, on raisonne par induction sur Λ :

– au départ, $\Lambda = (p, a\bot)$. La propriété est donc établie.
– soit maintenant une partie partielle Λ, pour laquelle on suppose le résultat établi. Soit Λ' un prolongement de Λ par un coup (d'Eve ou d'Adam).
Si Λ contenait déjà une configuration de pile vide, la seconde propriété étant vraie pour Λ, elle l'est encore pour Λ'.
Si Λ ne visite pas de configuration de pile vide, et si Λ' ne se termine pas par une configuration de pile vide, le premier point ne pose pas de problème : si Eve joue, cela vient de la définition de φ_R. Si c'est Adam qui joue, cela vient du fait que la

dernière position de Λ et la dernière position de $\tau_u(\Lambda)$ ont le même état de contrôle et le même sommet de pile (les mêmes types de transitions peuvent donc être effectuées).

Enfin, si Λ ne visite pas de configuration de pile vide, et si Λ' se termine par une configuration de pile vide, en raisonnant comme ci-dessus, on établit que $\tau_u(\Lambda')$ est une partie partielle dans $\mathbb{G}$ commençant en $(p, au\bot)$, où Eve respecte φ. De plus, $\tau_u(\Lambda)$ est préfixe d'un élément de $\mathcal{L}_\varphi$. Par définition, l'état de contrôle dans la dernière position de $\tau_u(\Lambda')$ (qui prolonge $\tau_u(\Lambda)$) est donc dans R. Or, l'état de contrôle dans la dernière position de Λ' est le même que dans la dernière position de $\tau_u(\Lambda')$, ce qui prouve le second point.

Ainsi, une partie Λ dans $\mathbb{G}(R)$, où Eve respecte φ_R, est gagnante pour Eve. En effet, si la pile est vidée, l'état de contrôle dans la première configuration de pile vide visitée est un élément de R. Si la pile n'est jamais vidée, $\tau_u(\Lambda)$ est une partie de $\mathbb{G}$ commençant en $(p, au\bot)$, où Eve respecte φ. C'est donc une partie gagnante pour Eve. Or, $\Lambda = \tau_{u^{-1}}(\tau_u(\Lambda))$, et comme Ω est invariante par translation vers le bas, on en déduit que Λ est également dans Ω, et donc dans $\Omega(R)$ (puisque toutes les configurations visitées sont de pile non vide dans Λ).

L'ensemble R est donc bien un ensemble de retour pour (p, a) dans $\mathbb{G}$.

Il nous reste donc à établir que, pour tout $r \in R$, $(r, u\bot)$ est une position gagnante pour Eve dans $\mathbb{G}$.

Soit donc $r \in R$. Par définition de R, il existe une partie partielle, où Eve respecte φ, et qui est de la forme $\Lambda' \cdot (r, u\bot)$, où Λ' ne visite pas de configuration de pile vide. On définit une stratégie gagnante φ_r pour Eve dans $\mathbb{G}$ depuis $(r, u\bot)$ de la façon suivante : pour toute partie partielle Λ commençant dans $(r, u\bot)$, on pose $\varphi_r(\Lambda) = \varphi(\Lambda' \cdot \Lambda)$. Ainsi, pour toute partie Λ jouée selon φ_r, $\Lambda' \cdot \Lambda$ est une partie jouée selon φ et commençant dans $(p, au\bot)$. Dès lors, $\Lambda' \cdot \Lambda$ est une partie gagnante pour Eve, et donc Λ est également une partie gagnante. Dès lors $(r, u\bot)$ est gagnante pour Eve dans $\mathbb{G}$. ■

3.5.5 Régularité des ensembles de positions gagnantes

Considérons une configuration $(p, u\bot)$ dans le jeu $\mathbb{G}$ et posons $u = a_1 a_2 \cdots a_n$. En utilisant le lemme 3.3, on en déduit que $(p, u\bot)$ est gagnant pour Eve si et seulement s'il existe $R \in \mathcal{R}(p, a_1)$ tel que pour tout $r \in R$, (r, u') soit gagnant pour Eve, où l'on pose $u' = a_2 a_3 \cdots a_n$. Ainsi, on doit deviner un ensemble de retour R pour (p, a_1) et vérifier pour tout $r \in R$, que les configurations (r, u') sont gagnantes, ce que l'on fait à nouveau en devinant des ensembles de retour, et ainsi de suite jusqu'à devoir vérifier que des configurations de la forme $(s, \bot)$ sont gagnantes. En d'autres termes, on a donc une procédure de décision qui lit le mot de pile de haut en bas (ou de gauche à droite), en alternant des actions existentielles (choix des ensembles de retour) avec des actions universelles (calculs en parallèle pour tous les états d'un ensemble de retour). Un tel comportement peut être facilement réalisé par un automate non déterministe dont les états de contrôles codent une partie de Q.

On définit alors un $\mathcal{P}$-automate $\mathcal{A} = \langle Q, \Gamma, I, F, \delta \rangle$ où :

– Les états de $\mathcal{A}$ sont les parties de Q.
– L'alphabet d'entrée est Γ, l'alphabet de pile de $\mathcal{P}$.

– $I = \{\{q\} \mid q \in Q\}$: les états initiaux sont les singletons. Par la suite on identifiera $\{q\}$ et q.
– $F = \mathcal{P}(\{p \mid (p, \perp)$ gagnante pour Eve$\})$.
– la fonction de transition δ est définie par :

$$\delta(S, a) = \{S' \mid \exists R_p \in \mathcal{R}(p, a) \forall p \in S, \text{ t.q. } S' = \bigcup_{p \in S} R_p\}$$

On a alors le résultat suivant :

Théorème 3.7 *Le $\mathcal{P}$-automate $\mathcal{A}$ reconnaît l'ensemble des positions gagnantes pour Eve dans le jeu $\mathbb{G}$.*

Preuve. On remarque tout d'abord le point suivant : un mot u est accepté en partant dans un état R ssi il est accepté en partant de tout état $r \in R$. Cette propriété est une conséquence directe de la définition de δ.

Montrons dans un premier temps que toute position gagnante pour Eve est acceptée par $\mathcal{A}$. On raisonne par récurrence sur la longueur du mot de pile. Soit donc une configuration $(p, u\perp)$ gagnante pour Eve dans $\mathbb{G}$.

– si $|u| = 0$, la configuration est donc de la forme $(p, \perp)$ et elle est acceptée par définition de F.
– supposons le résultat établi pour toutes les configurations gagnantes de mot de pile, de longueur inférieure ou égale à un certain entier $n \geq 1$. Considérons une configuration gagnante pour Eve de longueur égale à $n + 1$. Une telle configuration est de la forme $(p, au\perp)$, avec $|u| = n - 1$. Comme $(p, au\perp)$ est gagnante pour Eve, il existe, d'après le lemme 3.3, un ensemble de retour R pour (p, a) tel que $(r, u\perp)$ est gagnant pour tout $r \in R$. Or, par hypothèse de récurrence, on sait que $(r, u\perp)$ est accepté par $\mathcal{A}$ pour tout $r \in R$. Dès lors, le calcul de $\mathcal{A}$, qui consiste à choisir la transition initiale allant vers $R \in \delta(\{p\}, a)$, et ensuite de *composer* avec les calculs acceptants de $\mathcal{A}$ sur les $(r, u\perp)$, est un calcul acceptant.

Réciproquement, montrons que tout configuration acceptée par $\mathcal{A}$ est gagnante pour Eve dans le jeu $\mathbb{G}$. On raisonne à nouveau par récurrence sur la longueur du mot de pile. Soit donc une configuration $(p, u\perp)$ acceptée par $\mathcal{A}$.

– si $|u| = 0$, la configuration est donc de la forme $(p, \perp)$ et elle est gagnante par définition de F.
– supposons le résultat établi pour toutes les configurations gagnantes de mot de pile de longueur inférieure ou égale à un certain entier $n \geq 1$. Considérons une configuration de longueur égale à $n + 1$ et acceptée par $\mathcal{A}$. Une telle configuration est de la forme $(p, au\perp)$, avec $|u| = n - 1$. Par définition de δ, il existe un ensemble de retour R pour (p, a) tel que, pour tout $r \in R$, $(r, u\perp)$ est accepté par $\mathcal{A}$. La configuration $(r, u\perp)$ est donc gagnant pour Eve par hypothèse de récurrence. Dès lors, d'après le lemme 3.3, $(p, au\perp)$ est une position gagnante pour Eve dans $\mathbb{G}$.

■

Chapitre 4

Jeux, logique et automates d'arbres

Dans ce qui suit, nous allons considérer des problèmes de model-checking pour des structures de Kripke finies. Ces notions vont être définies dans ce qui suit, mais pour faire court, une structure de Kripke est un graphe pour lequel on possède une fonction de valuation, associant à chaque commet du graphe un ensemble de propositions atomiques vraie dans ce sommet ; le model-checking, consiste à décider pour une structure de Kripke donnée, un sommet dans cette dernière et une formule logique, si la formule est vraie dans ce sommet.

Dans ce qui suit, nous montrerons comment le problème du model-checking pour une certaine logique (le μ-calcul) peut être réduit à décider de l'acceptation d'une structure de Kripke par un automate d'arbre alternant, problème qui peut enfin être ramené à décider le gagnant dans un jeu de parité. En fait, cette série de transformation fonctionne également dans le sens contraire et les trois problèmes suivants sont polynomialement équivalents :

1. Étant donnée une formule φ du μ-calcul, une structure de Kripke $\mathcal{K}$ et un sommet s de $\mathcal{K}$, φ est-elle satisfaite en s dans la structure $\mathcal{K}$?
2. Étant donné un automate d'arbre alternant $\mathcal{A}$, une structure de Kripke $\mathcal{K}$ et un sommet s de $\mathcal{K}$, $\mathcal{A}$ accepte-t-il la structure $\mathcal{K}$ avec s pour origine ?
3. Étant donné un jeu de parité $\mathbb{G}$ et un sommet s dans ce jeu, Eve possède-t-elle une stratégie gagnante dans $\mathbb{G}$ depuis s ?

4.1 μ-calcul

On se fixe pour la suite un ensemble $\mathbf{P}$ de variables propositionnelles.

Définition 4.1 *Une **structure de Kripke** est un triple $\mathcal{K} = (S, R, \lambda)$ où*
- *S est un ensemble d'états,*
- *$R \subseteq S \times S$ est une relation sur S, et*
- *$\lambda : \mathbf{P} \to \mathcal{P}(S)$ est une fonction qui associe à toute variable propositionnelle un sous-ensemble d'états (où elle est vraie).*

Dans le cas où l'on regarde $\gamma : S \to \mathcal{P}(\mathbf{P})$ définit par $\gamma(s) = \{p \in \mathbf{P} \mid s \in \lambda(p)\}$, on peut voir une structure de Kripke comme un graphe orienté dont chaque sommet est étiqueté par l'ensemble des propositions atomiques qui y sont vraies.

Une structure de Kripke ***pointée*** *est un couple $(\mathcal{K}, s_i)$ où s_i est un état particulier de la structure que l'on qualifiera d'initial. Enfin, une structure de Kripke sera dite finie si elle possède un nombre fini d'états.*

Le μ-calcul modal est une logique dite de point fixe qui permet d'exprimer des propriétés des structures de Kripke (pointées).

4.1.1 Syntaxe

On note L_μ l'ensemble des formules du μ-calcul modal et on le définit par induction :
- $\bot$ et $\top$ appartiennent à L_μ.
- Pour tout $p \in \mathbf{P}$, p et $\neg p$ appartiennent à L_μ.
- Si φ et ψ sont deux formules de L_μ, il en est de même pour $\varphi \wedge \psi$ et $\varphi \vee \psi$.
- Si φ est une formule de L_μ, il en est de même pour $\Box\varphi$ et $\diamond\varphi$.
- Si $p \in \mathbf{P}$ et $\varphi \in L_\mu$, et si p n'apparaît que positivement dans φ, alors $\mu p\varphi$ et $\nu p\varphi$ appartiennent à L_μ.

Les opérateurs μ et ν sont qualifiés d'opérateurs de point fixe. Étant donnée une formule φ de L_μ, on définit l'ensemble $free(\varphi)$ des variables libres de φ par induction :
- $free(\bot) = free(\top) = \emptyset$,
- $free(p) = free(\neg p) = \{p\}$,
- $free(\varphi \vee \psi) = free(\varphi \wedge \psi) = free(\varphi) \cup free(\psi)$,
- $free(\Box\varphi) = free(\diamond\varphi) = free(\varphi)$,
- $free(\mu p\varphi) = free(\nu p\varphi) = free(\varphi) \setminus \{p\}$.

Enfin, on note F_μ et F_ν les ensembles de μ- et ν-formules :
- $F_\mu = \{\mu p\varphi \mid \varphi \in L_\mu\}$ et
- $F_\mu = \{\nu p\varphi \mid \varphi \in L_\mu\}$.

Les formules de $F_\mu \cup F_\nu$ sont qualifiées de formules de point fixe.

4.1.2 Sémantique

Nous commençons par donner (sans preuve) une version du théorème de Knaster-Tarski :

Proposition 4.1 (Knaster-Tarski) *Soit S un ensemble et soit $g : \mathcal{S} \to \mathcal{S}$ une fonction monotone pour l'inclusion. Alors g possède un plus petit point fixe μg et un plus grand point fixe νg qui satisfont les équations suivantes :*

$$\mu g = \bigcap\{S' \subseteq S \mid g(S') \subseteq S'\}$$

$$\nu g = \bigcap\{S' \subseteq S \mid g(S') \supseteq S'\}$$

Afin de définir la sémantique du μ-calcul, nous avons besoin de la notion de substitution dans une structure de Kripke.

Définition 4.2 *Soit une structure de Kripke $\mathcal{K} = (S, R, \lambda)$. Soit $S' \subseteq S$ et soit une variable propositionnelle $p \in \mathbf{P}$. On définit alors*

$$\lambda[p \mapsto S'](p') = \begin{cases} S' & si\ p' = p \\ \lambda(p') & si\ p' \neq p \end{cases}$$

Enfin, on définit la structure de Kripke $\mathcal{K}[p \mapsto S']$ comme

$$\mathcal{K}[p \mapsto S'] = (S, R, \lambda[p \mapsto S'])$$

Les formules de L_μ sont évaluées sur des structures de Kripke et donc la sémantique d'une formule φ sur une structure $\mathcal{K} = (S, R, \lambda)$, notée $||\varphi||_\mathcal{K}$ est un sous-ensemble de S qui décrit les états où φ est vraie. La sémantique est donnée par induction :

- $||\bot||_\mathcal{K} = \emptyset$ et $||\top||_\mathcal{K} = S$
- $||p||_\mathcal{K} = \lambda(p)$ et $||\neg p||_\mathcal{K} = S \setminus \lambda(p)$
- $||\varphi \wedge \psi||_\mathcal{K} = ||\varphi||_\mathcal{K} \cap ||\psi||_\mathcal{K}$ et $||\varphi \vee \psi||_\mathcal{K} = ||\varphi||_\mathcal{K} \cup ||\psi||_\mathcal{K}$
- $||\Box\varphi||_\mathcal{K} = \{s \in S \mid \forall s' \in S,\ (s, s') \in R \Rightarrow s' \in ||\varphi||_\mathcal{K}\}$
- $||\diamond\varphi||_\mathcal{K} = \{s \in S \mid \exists s' \in S,\ (s, s') \in R$ et $s' \in ||\varphi||_\mathcal{K}\}$
- $||\mu p\varphi||_\mathcal{K} = \bigcap\{S' \subseteq S \mid ||\varphi||_{\mathcal{K}[p \mapsto S']} \subseteq S'\}$
- $||\nu p\varphi||_\mathcal{K} = \bigcup\{S' \subseteq S \mid ||\varphi||_{\mathcal{K}[p \mapsto S']} \supseteq S'\}$

Nous écrirons par la suite $(\mathcal{K}, s) \models \varphi$ pour signifier que $s \in ||\varphi||_\mathcal{K}$. Enfin, on peut remarquer que dans la définition précédente, $||\mu p\varphi||_\mathcal{K}$ et $||\nu p\varphi||_\mathcal{K}$ sont respectivement le plus petit et le plus grand point fixe de la fonction (monotone) suivante

$$g : \mathcal{P}(S) \to \mathcal{P}(S), \quad S' \mapsto ||\varphi||_{\mathcal{K}[p \mapsto S']}$$

Remarquons qu'une autre façon de calculer les points fixes précédents est de considérer, pour le plus petit point fixe la limite de la suite (croissante) définie par $S_0 = \emptyset$ et $S_{i+1} = g(S_i)$, et pour le plus grand point fixe de prendre la limite de la suite (décroissante) donnée par $S_0 = S$ et $S_{i+1} = g(S_i)$.

4.1.3 Exemples de formules

Certains des exemples qui suivent ne comportent pas de preuve. Elles sont laissées en guise d'exercice.

Exemple 4.1 On considère la formule $\varphi_0 = \mu q_0(\Box q_0)$. Pour toute structure de Kripke $\mathcal{K}$, $||\varphi_0||_\mathcal{K}$ est l'ensemble des états depuis lesquels il n'existe pas de chemin infini.

Considérons maintenant la formule $\varphi_1 = \mu q_1(q_0 \vee \diamond q_1)$. Pour toute structure de Kripke $\mathcal{K}$, $||\varphi_1||_\mathcal{K}$ est l'ensemble des états depuis lesquels il existe un chemin (fini) aboutissant dans un état où q_0 est vrai.

La formule $\varphi'_1 = \diamond\varphi_1$ est elle vraie dans tout état depuis lequel il existe un chemin de longueur non nulle aboutissant dans un état où q_0 est vrai.

Enfin, posons $\varphi_2 = \nu q_2(\diamond\mu q_1((q_2 \wedge q_0) \vee \diamond q_1))$. Remarquons tout d'abord que $\diamond\mu q_1((q_2 \wedge q_0) \vee \diamond q_1)$ correspond à la formule φ'_1 précédente dans laquelle q_0 est remplacée par $q_2 \wedge q_0$: elle est donc vraie dans les états depuis lesquels il existe un chemin de longueur au moins 1 aboutissant dans un état où q_2 et q_0 sont vrais. Enfin, la formule φ_2 est vrai dans tout état depuis lequel il existe un chemin infini le long duquel q_0 est infiniment souvent vrai.

Exemple 4.2 Considérons la formule $\varphi_3 = \mu q_1(\nu q_2(q_0 \wedge \diamond q_2) \vee \Box q_1)$. Cette formule est vrai dans un état ssi tous les chemins qui en partent aboutisse,t au bout d'un temps fini dans un état depuis lequel il existe un chemin le long duquel q_0 est toujours vrai.

4.2 Des jeux au μ-calcul

Exemple 4.3 Considérons la formule $\varphi_4 = \mu q(q_0 \vee (q_{\mathbf{E}} \wedge \diamond q) \vee (\neg q_{\mathbf{E}} \wedge \Box q))$. Prenons maintenant une structure de Kripke représentant un graphe de jeu de la façon suivante : $q_{\mathbf{E}}$ est vrai dans un état donné si et seulement si celui-ci appartient à Eve. Il est facile de voir que φ est vrai exactement dans les états où Eve a une stratégie gagnante dans le jeu d'accessibilité vers un sommet dans lequel q_0 est vrai.

Exemple 4.4 Considérons la formule

$$\varphi_5 = \mu s_1 \nu s_0 \left(\bigvee_{i<2} (q_{\mathbf{E}} \wedge q_i \wedge \diamond s_i) \vee \bigvee_{i<2} (\neg q_{\mathbf{E}} \wedge q_i \wedge \Box s_i) \right)$$

On l'interprète sur une structure de Kripke représentant un graphe de jeu coloré, où un état satisfait q_i ssi sa couleur est i.

Tout d'abord, $\nu s_0 \left(\bigvee_{i<2} (q_{\mathbf{E}} \wedge q_i \wedge \diamond s_i) \vee \bigvee_{i<2} (\neg q_{\mathbf{E}} \wedge q_i \wedge \Box s_i) \right)$ est vraie ssi Eve peut forcer de ne voir soit que la couleur 0 soit voir la couleur 1 mais à l'état suivant de satisfaire s_1. Maintenant, la formule φ_5 est vraie ssi Eve peut forcer de ne voir que finiment la couleur 1, et décrit donc l'ensemble des positions gagnantes pour un jeu de max-parité à deux couleurs, 0 et 1.

Exemple 4.5 Considérons la formule

$$\varphi_6 = \nu s_2 \mu s_1 \nu s_0 \left(\bigvee_{i<3} (q_{\mathbf{E}} \wedge q_i \wedge \diamond s_i) \vee \bigvee_{i<2} (\neg q_{\mathbf{E}} \wedge q_i \wedge \Box s_i) \right)$$

On l'interprète sur une structure de Kripke représentant un graphe de jeu coloré, où un état satisfait q_i ssi sa couleur est i.

Tout d'abord, $\nu s_0 \left(\bigvee_{i<3} (q_{\mathbf{E}} \wedge q_i \wedge \diamond s_i) \vee \bigvee_{i<3} (\neg q_{\mathbf{E}} \wedge q_i \wedge \Box s_i) \right)$ est vraie ssi Eve peut forcer de ne voir soit que la couleur 0 soit voir la couleur 1 mais à l'état suivant de satisfaire s_1 soit voir la couleur 2 mais à l'état suivant satisfaire s_2. Maintenant, la formule $\mu s_1 \nu s_0 \left(\bigvee_{i<3} (q_{\mathbf{E}} \wedge q_i \wedge \diamond s_i) \vee \bigvee_{i<3} (\neg q_{\mathbf{E}} \wedge q_i \wedge \Box s_i) \right)$ est vraie ssi Eve peut forcer de ne voir à partir d'un moment que la couleur 0, ou de voir infiniment q_1 mais de voir aussi infiniment souvent s_2 (après chaque visite à q_1 on doit voir plus tard s_2). Enfin, la formule φ_6 est vraie ssi Eve peut jouer de sorte que si 1 est infiniment souvent vérifié, on voit infiniment souvent la couleur 2, c'est à dire qu'Eve possède une stratégie gagnante au jeu de max-parité dont les couleurs sont 0, 1 et 2.

Plus généralement, on a :

Théorème 4.1 *La formule φ suivante est vraie dans tout sommet depuis lequel Eve possède une stratégie gagnante dans le jeu de max-parité de couleurs $0, 1, \cdots n$ (où η vaut ν si n est pair et μ sinon) :*

$$\eta s_n \cdots \nu s_2 \mu s_1 \nu s_0 \left(\bigvee_{i \leq n} (q_{\mathbf{E}} \wedge q_i \wedge \diamond s_i) \vee \bigvee_{i \leq n} (\neg q_{\mathbf{E}} \wedge q_i \wedge \Box s_i) \right)$$

La preuve peut être obtenue en transformant la formule en automate de parité comme indiqué par la suite et en vérifiant que celui-ci reconnaît bien les structures de Kripke pointées voulues. Celle-ci est donnée plus loin.

Ce résultat, prouve que l'on peut réduire les problèmes du vide pour les automates d'arbre alternants (voir plus bas), et de décision du gagnant dans un jeu de parité à celui du model-checking pour une formule du μ-calcul. Dans la suite, on prouve la réciproque.

4.3 Automates d'arbres et jeux de parité

4.3.1 Définitions

Maintenant, nous donnons la définition d'un automate d'arbre. Il existe de nombreuses définitions dans la littérature et nous adaptons ici celle des automates d'arbres comme des machines acceptant des structures de Kripke pointées. L'*arbre* est alors le dépliage de la structure de Kripke.

Considérons un ensemble Q de symboles. Les **conditions de transition** TC^Q sur Q sont définies comme suit :

- Les symboles 0 et 1 sont des conditions de transition.
- Pour tout $p \in \mathbf{P}$, p et $\neg p$ sont des conditions de transition.
- Pour tout $q \in Q$, q, $\Box q$ et $\diamond q$ sont des conditions de transition.
- Pour tout $q_1, q_2 \in Q$, $q_1 \wedge q_2$ et $q_1 \vee q_2$ sont de conditions de transition.

Définition 4.3 *Un* ***automate d'arbre alternant*** *est un quadruplet $\mathcal{A} = (Q, q_i, \delta, col)$ où*

- *Q est un ensemble fini d'états,*
- *$q_i \in Q$ est l'état initial,*
- *$\delta : Q \to TC^Q$ est la fonction de transition, et*
- *$col : Q \to \mathbb{N}$ est la fonction de coloriage.*

On adoptera les notations suivantes :

$$Q_\Box = \{q \in Q \mid \exists q' : \delta(q) = \Box q'\}$$

$$Q_\diamond = \{q \in Q \mid \exists q' : \delta(q) = \diamond q'\}$$

Pour un état $q \in Q$, on définira $\overrightarrow{q} = q'$ si $\delta(q) = \Box q'$ ou $\delta(q) = \diamond q'$. Dans les autres cas $\overrightarrow{q}$ n'est pas défini.

Considérons une structure de Kripke pointée $(\mathcal{K} = (S, R, \lambda), s_i)$ et un automate d'arbre alternant $\mathcal{A} = (Q, q_i, \delta, col)$. Afin de définir le comportement de $\mathcal{A}$ sur $(\mathcal{K}, s_i)$, on considère des suites de paires dans $Q \times S$, c'est à dire des mots sur l'alphabet $Q \times S$. Le **comportement** de $\mathcal{A}$ sur $(\mathcal{K}, s_i)$ est le plus petit language $V \subseteq (Q \times S)^*$ avec $(q_i, s_i) \in V$ et tel que pour tout mot $v(q, s) \in V$, on a :

– Si $\delta(q) = q'$ pour un $q' \in Q$, alors $v(q,s)(q',s) \in V$.
– Si $\delta(q) = q_1 \wedge q_2$ ou $\delta(q) = q_1 \vee q_2$ pour $q_1, q_2 \in Q$, alors $v(q,s)(q_1,s) \in V$ et $v(q,s)(q_2,s') \in V$.
– Si $\delta(q) = \Box q'$ ou $\delta(q) = \diamond q'$ pour un $q' \in Q$, alors $v(q,s)(q',s') \in V$ pour tout s' tel que $(s,s') \in R$.

Enfin l'acceptation est définie en considérant un jeu de parité. Pour cela, on considère le graphe $G = (V, E)$ où $E = \{(v, v(q,s)) \mid v(q,s) \in V,\ v \neq \varepsilon\}$. Ensuite on définit un graphe de jeu $\mathcal{G} = (G, V_{\mathbf{E}}, V_{\mathbf{A}})$ par $v(q,s) \in V_{\mathbf{E}}$ ssi :
– $\delta(q) = 0$,
– $\delta(q) = p$ et $s \notin \lambda(p)$,
– $\delta(q) = \neg p$ et $s \in \lambda(p)$,
– $\delta(q) = q'$,
– $\delta(q) = q_1 \vee q_2$ pour $q_1, q_2 \in Q$, ou
– $\delta(q) = \diamond q'$.

et $v(q,s) \in V_{\mathbf{A}}$ ssi :
– $\delta(q) = 1$,
– $\delta(q) = p$ et $s \in \lambda(p)$,
– $\delta(q) = \neg p$ et $s \notin \lambda(p)$,
– $\delta(q) = q_1 \wedge q_2$ pour $q_1, q_2 \in Q$, ou
– $\delta(q) = \Box q'$.

Enfin, on définit la condition de coloriage sur V par $col(v(q,s)) = col(q)$ pour tout $v(q,s) \in V$, et on note $\mathbb{G}$ le jeu de parité sur $\mathcal{G}$.

L'automate $\mathcal{A}$ **accepte** la structure de Kripke pointée $(\mathcal{K}, s_i)$ ssi Eve possède une stratégie gagnante dans $\mathbb{G}$ depuis le sommet (q_i, s_i). Remarquons que le graphe de jeu $\mathcal{G}$ contient des cul-de-sac : une partie aboutissant dans un tel sommet est perdu par le joueur qui contrôle le cul-de-sac.

Le principal inconvénient de cette définition est que le graphe de jeu $\mathcal{G}$ est infini. Cependant, il est facile de construire un jeu équivalent qui cette fois est joué sur un graphe fini. Pour cela on considère le graphe $G' = (V', E')$ où $V' = Q \times S$ et où pour tout $(q,s) \in V'$,
– Si $\delta(q) = q'$ pour un $q' \in Q$, alors $((q,s),(q',s)) \in E'$.
– Si $\delta(q) = q_1 \wedge q_2$ ou $\delta(q) = q_1 \vee q_2$ pour $q_1, q_2 \in Q$, alors $((q,s)(q_1,s)) \in E'$ et $((q,s)(q_2,s)) \in E'$.
– Si $\delta(q) = \Box q'$ ou $\delta(q) = \diamond q'$ pour un $q' \in Q$, alors $((q,s),(q',s')) \in E'$ pour tout s' tel que $(s,s') \in R$.

Ensuite V' est partitionné en $V'_{\mathbf{E}} \cup V'_{\mathbf{A}}$ comme précédemment pour V, et col est défini par $col((q,s)) = col(q)$. On note enfin $\mathbb{G}'$ le jeu de parité associé.

On a alors facilement l'équivalence suivante :

Proposition 4.2 *Eve a une stratégie gagnante depuis (q_i, s_i) dans $\mathbb{G}$ ssi Eve a une stratégie gagnante depuis (q_i, s_i) dans $\mathbb{G}'$.*

On en déduit donc le résultat suivant :

Théorème 4.2 *On peut décider étant donné un automate alternant d'arbre et une structure de Kripke pointée si cette dernière est acceptée par l'automate. De plus la complexité est la même que celle consistant à résoudre un jeu de parité (de même taille)*

4.3.2 Exemples d'automates

On donne quelques exemples d'automate reliés aux formules vues dans la section 4.1.3.

Exemple 4.6 Considérons l'automate $\mathcal{A}_0$ à un seul état s et dont la fonction de transition est donnée par $\delta(s) = \Box s$. On suppose que l'état s est de couleur 1. Il est facile de voir qu'une structure ne peut être acceptée que s'il n'existe pas de chemin infini dans celle-ci. En d'autre terme cet automate correspond à la formule φ_0 précédente.

On donne maintenant un automate $\mathcal{A}_1$ qui correspond à la formule φ_1, c'est à dire qui accepte une structure ssi il existe un chemin vers un état où q_0 est vrai. A nouveau, un seul état s suffit mais cette fois la fonction de transition est donnée par $\delta(s) = q_0 \vee \diamond s$. Ainsi si l'automate est dans un état où q_0 est vrai, il va pouvoir s'arrêter sinon, il va continuer son calcul depuis un état successeur.

On termine en donnant un automate $\mathcal{A}_2$ qui correspond à la formule φ_2 c'est à dire qui accepte une structure ssi l'on peut atteindre un état depuis lequel q_0 est toujours vrai. Cette fois on a besoin de deux états s_0 et s_1 de couleurs respectives 0 et 1 : ainsi une structure est acceptée si toute partie dans le jeu associé voit 0 infiniment souvent. L'idée est de propager l'information concernant le fait que q_0 soit vrai ou non. Pour cela on pose $\delta(s_0) = \delta(s_1) = (q_0 \wedge \diamond s_0) \vee (\neg q_0 \wedge s_1)$.

En fait les deux automates $\mathcal{A}_1$ et $\mathcal{A}_2$ ne correspondent pas exactement à la définition donnée car on utilise des transitions qui sont plus générales que les conditions de transition. Cependant il est facile de voir que l'on peut ajouter des états intermédiaires pour se conformer à la définition. Ainsi, pour l'automate $\mathcal{A}_1$, on peut ajouter deux états s_l et s_r de priorité 2, assigner la priorité 1 à s et définir $\delta(s) = s_l \vee s_r$, $\delta(s_l) = q_0$ et $\delta(s_r) = \diamond s$

4.4 Du μ-calcul aux automates d'arbre alternants/aux jeux

Dans cette section, on prouve que pour toute formule φ de L_μ on peut construire un automate d'arbre alternant $\mathcal{A}_\varphi$ tel que $\mathcal{A}$ accepte une structure pointée $(\mathcal{K}, s)$ ssi $(\mathcal{K}, s) \models \varphi$.

On se fixe donc une formule φ de L_μ et on considère l'automate suivant $\mathcal{A}_\varphi = (Q, q_i, \delta, col)$ où :

- Q est l'ensemble qui contient pour toute sous-formule ψ de φ (incluant φ) un état noté $\langle\psi\rangle$.
- $q_i = \langle\varphi\rangle$.
- $\delta : Q \to TC^Q$ est définit par :
  - $\delta(\langle\bot\rangle) = 0$ et $\delta(\langle\top\rangle) = 1$.
  - $\delta(\langle p\rangle) = \begin{cases} p \text{ si } p \in free(\varphi) \\ \langle\varphi_p\rangle \text{ si } p \notin free(\varphi). \end{cases}$, où (dans le second cas) φ_p désigne l'unique sous-formule de φ de la forme $\mu p\psi$ ou $\nu p\psi$.
  - $\delta(\langle\neg p\rangle) = \neg p$.
  - $\delta(\langle\psi_1 \wedge \psi_2\rangle) = \langle\psi_1\rangle \wedge \langle\psi_2\rangle$ et $\delta(\langle\psi_1 \vee \psi_2\rangle) = \langle\psi_1\rangle \vee \langle\psi_2\rangle$
  - $\delta(\langle\Box\psi\rangle) = \Box\langle\psi\rangle$ et $\delta(\langle\diamond\psi\rangle) = \diamond\langle\psi\rangle$.
  - $\delta(\langle\mu p\psi\rangle) = \langle\psi\rangle$ et $\delta(\langle\nu p\psi\rangle) = \langle\psi\rangle$.

– La fonction de priorité col est donnée par :
  – $col(\langle\psi\rangle) = 2[\alpha(\psi)/2] - 1$ si $\psi \in F_\mu$
  – $col(\langle\psi\rangle) = 2[\alpha(\psi)/2]$ si $\psi \in F_\nu$
  – $col(\langle\psi\rangle) = 2[\alpha(\varphi)/2]$ si $\psi \notin F_\mu \cup F_\nu$

Ici la fonction α attribue à chaque sous-formule de φ qui est un point fixe un entier plus petit que celui attribué à toute ses sous-formules (la plus grande formule de point fixe reçoit la valeur 1, puis la seconde reçoit la valeur 2 et ainsi de suite).

Exemple 4.7 Reprenons l'exemple de la formule φ_1 et montrons que l'on retrouve l'automate $\mathcal{A}_1$ précédent. On a $\varphi_1 = \mu q_1(q_0 \vee \diamond q_1)$. L'automate $\mathcal{A}_{\varphi_1}$ a pour états : $\langle \mu q_1(q_0 \vee \diamond q_1)\rangle$, $\langle (q_0 \vee \diamond q_1)\rangle$, $\langle q_0\rangle$, $\langle \diamond q_1\rangle$ et $\langle q_1\rangle$. L'état initial est $\langle \mu q_1(q_0 \vee \diamond q_1)\rangle$, tous les états ont la couleur 2 sauf $\langle \mu q_1(q_0 \vee \diamond q_1)\rangle$ qui a la couleur 1.

La fonction de transition est donnée par :
– $\delta(\langle \mu q_1(q_0 \vee \diamond q_1)\rangle) = \langle q_0 \vee \diamond q_1\rangle$,
– $\delta(\langle q_0 \vee \diamond q_1\rangle) = \langle q_0\rangle \vee \langle \diamond q_1\rangle$,
– $\delta(\langle q_0\rangle) = q_0$,
– $\delta(\langle \diamond q_1\rangle) = q_1$,
– $\delta(\langle q_1\rangle) = \langle \mu q_1(q_0 \vee \diamond q_1)\rangle$.

Les états $\langle (q_0 \vee \diamond q_1)\rangle$ et $\langle q_1\rangle$ peuvent être ignorés en posant :
– $\delta(\langle \mu q_1(q_0 \vee \diamond q_1)\rangle) = \langle q_0\rangle \vee \langle \diamond q_1\rangle$,
– $\delta(\langle q_0\rangle) = q_0$,
– $\delta(\langle \diamond q_1\rangle) = \diamond\langle \mu q_1(q_0 \vee \diamond q_1)\rangle$.

Enfin, si l'on identifie $\langle \mu q_1(q_0 \vee \diamond q_1)\rangle$ avec s, $\langle q_0\rangle$ avec s_l et $\langle \diamond q_1\rangle$ avec s_r, on retrouve l'automate $\mathcal{A}_2$.

On a alors les lemmes suivants :

Lemme 4.1 *Soit ψ_1 et ψ_2 deux formules de L_μ. On a alors :*
– $L(\mathcal{A}_{\psi_1\wedge\psi_2}) = L(\mathcal{A}_{\psi_1}) \cap L(\mathcal{A}_{\psi_2})$,
– $L(\mathcal{A}_{\psi_1\vee\psi_2}) = L(\mathcal{A}_{\psi_1}) \cup L(\mathcal{A}_{\psi_2})$,

Preuve. On ne considère que la première égalité, la seconde se prouvant de la même façon.

Soit $(\mathcal{K}, s) \in L(\mathcal{A}_{\psi_1\wedge\psi_2})$: Eve possède une stratégie sans mémoire dans le jeu associé. Cette stratégie est alors également gagnante dans les jeux associés à $\mathcal{A}_{\psi_1}$ et $\mathcal{A}_{\psi_2}$ (qui se jouent sur des sous-graphes de jeu du jeu précédent). Réciproquement, si $(\mathcal{K}, s) \in L(\mathcal{A}_{\psi_1}) \cap L(\mathcal{A}_{\psi_2})$, on définit une stratégie gagnante pour Eve dans le jeu associé à $\mathcal{A}_{\psi_1\wedge\psi_2}$ de la façon suivante. Dans ce jeu, le sommet initial $(\langle\psi_1 \wedge \psi_2\rangle, s)$ est contrôlé par Adam et celui-ci peut aller soit en $(\langle\psi_1\rangle, s)$ soit en $(\langle\psi_2\rangle, s)$: dans le premier cas Eve suit sa stratégie gagnante dans le jeu associé à $\mathcal{A}_{\psi_1}$ et dans le second cas elle suit sa stratégie gagnante associée à $\mathcal{A}_{\psi_2}$ ■

Lemme 4.2 *Soit une formule ψ de L_μ. On a alors :*
– $L(\mathcal{A}_{\square\psi}) = \{(\mathcal{K}, s) \mid \forall s',\ (s, s') \in R \Rightarrow (\mathcal{K}, s') \in L(\mathcal{A}_\psi)\}$,
– $L(\mathcal{A}_{\diamond\psi}) = \{(\mathcal{K}, s) \mid \exists s',\ (s, s') \in R \textit{ et } (\mathcal{K}, s') \in L(\mathcal{A}_\psi)\}$.

Preuve. Les résultats se prouvent de la même façon que dans le lemme précédent. Une stratégie dans le premier jeu donne une stratégie dans le second jeu pour une (toute) structure pointée depuis s'. Réciproquement, à partir de stratégies pour une (toute) structure pointée depuis s' on reconstruit une stratégie dans le premier jeu. ■

On a enfin le résultat voulu :

Théorème 4.3 *Soit une formule φ de L_μ, et soit $(\mathcal{K}, s)$ une structure de Kripke pointée. On a alors l'équivalence suivante :*

$$(\mathcal{K}, s) \models \varphi \textit{ ssi } (\mathcal{K}, s) \in L(\mathcal{A}_\varphi)$$

Preuve. La preuve est faite par induction sur φ. Les seuls cas qui ne sont pas immédiats ou traités par les deux lemmes précédents sont ceux où φ est une formule de point fixe.

On commence par le cas où $\varphi = \mu p \psi$ et on pose $\mathcal{K} = (S, R, \lambda)$. On considère la fonction suivante $g : S' \mapsto ||\psi||_{\mathcal{K}[p \mapsto S']}$ qui par hypothèse d'induction peut être vue comme $g : S' \mapsto \{s' \mid (\mathcal{K}[p \mapsto S'], s') \in L(\mathcal{A}_\psi)\}$.

Comme on a que $(\mathcal{K}, s) \models \mu p \psi$ ssi $s \in \mu g$ et que $(\mathcal{K}, s) \in L(\mathcal{A}_{\mu p \psi})$ ssi Eve gagne dans le jeu associé à $\mu p \psi$ et $(\mathcal{K}, s)$, il faut donc montrer que

$$\mu g = \{s \in S \mid \text{Eve gagne dans le jeu associé à } \mu p \psi \text{ et } (\mathcal{K}, s)\}$$

On note S_μ la partie droite de cette équation.

Afin de montrer que $\mu g \subseteq S_\mu$, comme $\mu g = \bigcap\{S' \subseteq S \mid g(S') \subseteq S'\}$, il suffit de montrer que $g(S_\mu) \subseteq S_\mu$. Pour cela, soit $s \in g(S_\mu)$: Eve possède donc une stratégie gagnante dans le jeu $\mathbb{G}'$ associé à ψ et $(\mathcal{K}[p \mapsto S_\mu], s)$. Il faut alors montrer qu'Eve possède également une stratégie gagnante dans le jeu $\mathbb{G}$ associé à $\mu p \psi$ et $(\mathcal{K}, s)$.

Dans $\mathbb{G}$, le sommet initial est $(\langle \mu p \psi \rangle, s)$ et ce dernier possède une transition vers $(\langle \psi \rangle, s)$, le sommet initial de $\mathbb{G}'$. De plus toutes les transitions de $\mathbb{G}'$ se trouvent dans $\mathbb{G}$ et les sommets sont contrôlés par le même joueur, à l'exception de ceux de la forme $(\langle p \rangle, s')$ qui sont maintenant contrôlés par Eve et qui possède un arc vers $(\langle \mu p \psi \rangle, s')$. La stratégie pour Eve dans $\mathbb{G}$ est la suivante : elle va de $(\langle \mu p \psi \rangle, s)$ en $(\langle \psi \rangle, s)$ et ensuite respecte sa stratégie gagnante dans $\mathcal{G}'$ jusqu'à ce que l'on atteigne un sommet de la forme $(\langle p \rangle, s')$. Si cela n'arrive jamais, elle remporte la partie, sinon on remarque que $s' \in S_\mu$ car le sommet $(\langle p \rangle, s')$ est un cul-de-sac dans $\mathbb{G}'$ et est donc contrôlé par Adam (on suit une stratégie gagnante). Dès lors, Eve va en $(\langle \mu p \psi \rangle, s')$ et joue selon sa stratégie gagnante depuis ce point ($s' \in S_\mu$).

Considérons la réciproque. Pour cela, on prouve que tout point fixe de g contient S_μ, c'est à dire que pour tout $S' \subseteq S$, si $g(S') = S'$ alors $S_\mu \subseteq S'$. On se donne un point fixe S' de g et on suppose qu'il existe $s_0 \in S_\mu$ avec $s_0 \notin S' = g(S')$. Comme $s_0 \in S_\mu$, Eve possède une stratégie gagnante f dans le jeu $\mathbb{G}$ associé à $\mu p \psi$ et $(\mathcal{K}, s_0)$. Comme $s_0 \notin S'$, cette stratégie ne peux être gagnante dans le jeu $\mathbb{G}'$ associé à ψ et $(\mathcal{K}[p \mapsto S_\mu], s_0)$ et il existe donc une partie π_0 consistente avec f et perdante pour Eve dans $\mathbb{G}'$. On voit facilement que π_0 est finie et que son dernier sommet est de la forme $(\langle p \rangle, s_1)$ et donc que $s_1 \notin S'$ (c'est Adam qui gagne π_0). On peut voir π_0 comme le préfixe d'une partie gagnante π pour Eve dans $\mathbb{G}$ et dans π le successeur de $(\langle p \rangle, s_1)$ ne peut être que $(\langle \mu p \psi \rangle, s_1)$. On voit alors facilement que $s_1 \in S_\mu$. Comme $s_1 \in S_\mu$ et $s_1 \notin S'$, on se retrouve dans la même situation, et on peut alors construire un s_2 dans le même cas et ainsi de suite. Mais alors, on obtient une partie π infinie dans $\mathbb{G}$ qui respecte f mais visite infiniment souvent des sommets de la forme $(\langle \mu p \psi \rangle, s_i)$: la plus petite couleur infiniment visitée dans π est donc $col(\langle \mu p \psi \rangle)$ qui est impaire, ce qui est une contradiction.

Le cas des plus grands points fixe se prouve de *plus ou moins* façon duale. On peut aussi donner une preuve en réduisant ce cas au cas précédent en complémentant la formule φ (ce qui transforme les ν en μ) et en complémentant $\mathcal{A}_\varphi$ et enfin en vérifiant que l'on retrouve alors l'automate associé à $\mathcal{A}_\varphi$ ■

4.5 Retour sur le Théorème 4.1

Rappelons tout d'abord l'énoncé du Théorème 4.1 :

Théorème 4.1 *La formule φ suivante est vraie dans tout sommet depuis lequel Eve possède une stratégie gagnante dans le jeu de max-parité de couleurs $0, 1, \cdots n$ (où η vaut ν si n est pair et μ sinon) :*

$$\eta s_n \cdots \nu s_2 \mu s_1 \nu s_0 \left(\bigvee_{i \leq n} (q_{\mathbf{E}} \wedge q_i \wedge \diamond s_i) \vee \bigvee_{i \leq n} (\neg q_{\mathbf{E}} \wedge q_i \wedge \Box s_i) \right)$$

Afin d'en donner une preuve, il suffit de considérer l'automate $\mathcal{A}_\varphi$ associé à φ : celui-ci reconnaît les structures de Kripke pointées pour lesquelles φ est vraie. Il suffit donc de montrer que $\mathcal{A}_\varphi$ reconnaît en fait les structures de Kripke pointées représentant un jeu de parité tel qu'Eve possède une stratégie gagnante depuis l'état initial. En fait il est très facile de voir que $\mathcal{A}_\varphi$ possède cette propriété car ce dernier possède une structure très simple : au départ l'automate va faire du surplace (*i.e.* il reste sur le même état de la structure) jusqu'à arriver dans l'état $\langle (\bigvee_{i \leq n} (q_{\mathbf{E}} \wedge q_i \wedge \diamond s_i) \vee \bigvee_{i \leq n} (\neg q_{\mathbf{E}} \wedge q_i \wedge \Box s_i)) \rangle$: dans celui-ci Eve doit choisir le nouvel état qui – si elle ne veut pas perdre – est forcément $(q_{\mathbf{E}} \wedge q_i \wedge \diamond s_i)$ si elle contrôle l'état courant et que celui-ci a couleur i et est forcément $(q_{\mathbf{A}} \wedge q_i \wedge \Box s_i)$ si Adam contrôle l'état courant et que celui-ci a couleur i. Ensuite Eve ou Adam choisit un successeur dans la structure et ensuite Adam– s'il ne veut pas perdre – va dans l'état q_i.Enfin, on repart sur une formule de point fixe, fait du surplace et retombe sur l'état $\langle (\bigvee_{i \leq n} (q_{\mathbf{E}} \wedge q_i \wedge \diamond s_i) \vee \bigvee_{i \leq n} (\neg q_{\mathbf{E}} \wedge q_i \wedge \Box s_i)) \rangle$. Maintenant, si l'on regarde les couleurs visitées dans cette dernière série de mouvement, elles sont exactement i,...,n. Ainsi, il est clair que l'automate va accepter une structure de Kripke pointée ssi Eve a une stratégie gagnante dans le jeu associé depuis l'état initial.

4.6 Conclusion

On a donc prouvé le théorème suivant :

Théorème 4.4 *Les trois problèmes suivants sont polynomialement équivalents :*

1. *Étant donnée une formule φ du μ-calcul, une structure de Kripke $\mathcal{K}$ et un sommet s de $\mathcal{K}$, φ est-elle satisfaite en s dans la structure $\mathcal{K}$?*
2. *Étant donné un automate d'arbre alternant $\mathcal{A}$, une structure de Kripke $\mathcal{K}$ et un sommet s de $\mathcal{K}$, $\mathcal{A}$ accepte-t-il la structure $\mathcal{K}$ avec s pour origine ?*

3. *Étant donné un jeu de parité* $\mathbb{G}$ *et un sommet* s *dans ce jeu, Eve possède-t-elle une stratégie gagnante dans* $\mathbb{G}$ *depuis* s *?*