Les jeux d'accessibilité généralisée

Nathanaël Fijalkow[1,2], **Florian Horn**[1]

1. *Laboratoire d'Informatique Algorithmique: Fondements et Applications (LIAFA), Université Denis Diderot - Paris 7*
   florian.horn@liafa.univ-paris-diderot.fr
2. *Institut d'Informatique, Université de Varsovie*
   nath@mimuw.edu.pl

RÉSUMÉ. *Dans cet article, nous étudions les jeux d'accessibilité généralisée : ce sont des jeux à deux joueurs à somme nulle, pour lesquels l'objectif du premier joueur, Ève, est donné par une conjonction d'objectifs d'accessibilité. Dans un premier temps, nous montrons que le problème de décider si Ève a une stratégie gagnante est* PSPACE*-complet, mais polynomial en fixant pour paramètre le nombre d'objectifs. De plus, nous montrons que pour les deux joueurs, les stratégies gagnantes nécessitent beaucoup de mémoire : des bornes supérieures et inférieures sont présentées. Dans un second temps, nous nous intéressons à des restrictions de ces jeux pour lesquels nous cherchons des algorithmes efficaces pour décider du gagnant et construire des stratégies gagnantes.*

ABSTRACT. *In this paper, we consider two-player zero-sum games with generalized reachability objectives, defined as conjunctions of reachability objectives. We first prove that deciding the winner in such games is* PSPACE*-complete, although it is fixed-parameter tractable with the number of reachability objectives as parameter. Moreover, we consider the memory requirements for both players and give matching upper and lower bounds on the size of winning strategies. In order to construct more efficient algorithms, we consider subclasses of generalized reachability games.*

MOTS-CLÉS : *Jeux sur les graphes, Jeux d'accessibilité, Jeux pour la vérification, Conjonctions d'objectifs.*

KEYWORDS: *Games on graphs, Reachability games, Games for verification, Conjunction of objectives.*

DOI:10.3166/TSI.33.1-19

1. Introduction

Les jeux sur les graphes pour la synthèse de systèmes ouverts. Les jeux sur les graphes permettent de modéliser un cerain nombre de problèmes touchant aux mathématiques et à l'informatique théorique. L'exemple que nous développons ici est celui de la synthèse de systèmes ouverts, c'est-à-dire la construction (automatique) d'un contrôleur satisfaisant une spécification donnée. Un jeu se déroule sur un graphe dont les sommets représentent les différents états du système et les arêtes les transitions possibles entre différents états. Si dans un état donné, le contrôleur peut choisir l'évolution du système, alors le sommet correspondant est contrôlé par le premier joueur, Ève. Sinon, le système évolue de manière incertaine : nous considérons le scénario pessimiste où un adversaire, Adam, contrôle les sommets correspondants. À une exécution du système correspond une partie dans le jeu : on place un jeton sur le sommet initial, puis Ève et Adam déplacent le jeton le long des arêtes, construisant un chemin infini. La spécification du système donne l'objectif qu'Ève tâche de remplir sur ce chemin. En vue de construire un contrôleur correct, on s'intéresse à deux questions : qui gagne le jeu, et comment constuire (algorithmiquement) une stratégie gagnante. Une présentation plus complète de ce champ d'applications est donnée dans (Grädel *et al.*, 2002).

Spécifications de systèmes. Une spécification est donnée par un ensemble de chemins infinis, représentant les comportements corrects du système. Étant donné un chemin infini, on extrait une quantité finie d'informations pour déterminer si ce chemin vérifie la spécification. Par exemple, observer l'ensemble des sommets visités infiniment souvent permet de spécifier les propriétés ω-régulières, à savoir les objectifs de Büchi, parité, Streett, Rabin et Müller. D'autres informations peuvent être pertinentes, par exemple l'ordre dans lequel les sommets sont visités, pour spécifier des objectifs de logique temporelle, comme dans (Kupferman *et al.*, 2007 ; Horn *et al.*, 2008 ; Zimmermann, 2011). Dans cet article, nous observons l'ensemble des sommets visités (au moins une fois), ce qui permet de spécifier des combinaisons booléennes d'objectifs d'accessibilité, aussi appelés "objectifs faibles".

Objectifs d'accessibilité généralisée. Un objectif d'accessibilité (simple) est donné par un sous-ensemble F de sommets, et il est rempli si un sommet de F a été visité au cours de la partie. Ces objectifs permettent de spécifier qu'une propriété (représentée par F) est satisfaite le long de l'exécution. Pour étendre le pouvoir d'expression, nous considérons au lieu d'un seul objectif d'accessibilité une conjonction de k objectifs d'accessibilité. Dans ce contexte, on associe à chaque objectif d'accessibilité une couleur : l'objectif d'accessibilité généralisée est de voir chacune des k couleurs au moins une fois.

Plan. Les définitions sont données dans la partie 2. La partie 3 est dédiée à une étude générale des jeux d'accessibilité généralisée : complexité de résolution de ces jeux à deux joueurs et à un joueur, puis bornes de mémoire des stratégies gagnantes pour les deux joueurs. Dans la partie 4, nous considérons des sous-classes de ces jeux, avec pour objectif d'obtenir des algorithmes efficaces pour résoudre ces jeux et construire des stratégies gagnantes.

2. Définitions

Les jeux que nous considérons se jouent sur une *arène* $\mathcal{A} = (V, (V_\circ, V_\Box), E)$, qui est un graphe fini et orienté (V, E) enrichi d'une partition $(V_\circ, V_\Box)$ de l'ensemble des sommets V: les sommets de $V_\circ$, représentés par un cercle, sont contrôlés par Ève, et ceux de $V_\Box$, représentés par un carré, par Adam. On note n le nombre de sommets et m le nombre d'arêtes.

Une partie se déroule ainsi : on place un jeton sur un sommet initial v_0, puis le joueur qui contrôle ce sommet choisit une arête et déplace le jeton le long de cette arête jusqu'au prochain sommet, et ainsi de suite. Il en résulte une suite (infinie) de sommets $\pi = v_0, v_1, \ldots$, tel que pour tout i, on a $(v_i, v_{i+1}) \in E$: π est un chemin infini dans le graphe. Nous notons Π l'ensemble de ces parties, et définissons un objectif pour un joueur par l'ensemble des parties gagnantes $\Phi \subseteq \Pi$. Les jeux sont à somme nulle, ce qui signifie que si Ève a pour objectif Φ, alors Adam a pour objectif $\Pi \setminus \Phi$: Adam gagne si et seulement si Ève perd.

La donnée d'un jeu est un couple $\mathcal{G} = (\mathcal{A}, \Phi)$ où $\mathcal{A}$ est une arène et Φ un objectif.

Une *stratégie* est une fonction qui décrit, en consultant l'historique (fini) de la partie, le prochain coup à jouer. Formellement, une stratégie pour Ève est une fonction $\sigma : V^* \cdot V_\circ \to V$ telle que pour une histoire finie $w \in V^*$ et une position courante $v \in V_\circ$, le coup prescrit est légal : $(v, \sigma(w \cdot v)) \in E$. Les stratégies pour Adam sont définies de manière similaire, et souvent dénotées par τ. Étant donné un jeu $\mathcal{G} = (\mathcal{A}, \Phi)$, un sommet initial v_0 et deux stratégies, σ pour Ève et τ pour Adam, ceci définit de manière unique une partie, notée $\pi(v_0, \sigma, \tau)$. Cette partie est gagnante pour Ève si elle appartient à Φ. La phrase "Ève gagne depuis v_0" signifie qu'elle a une stratégie gagnante depuis v_0, en d'autres termes une stratégie σ assurant que contre toute stratégie τ d'Adam, la partie $\pi(v_0, \sigma, \tau)$ est gagnante. Le premier problème que l'on considère est de "résoudre le jeu", c'est-à-dire de décider si Ève gagne depuis un sommet v_0 initial fixé. Nous notons $\mathcal{W}_E(\mathcal{G})$ l'ensemble des positions gagnantes d'Ève, qui est l'ensemble des sommets depuis lesquels Ève gagne, et symétriquement pour Adam. Dans les jeux d'accessibilité généralisée, nous avons $\mathcal{W}_E(\mathcal{G}) \cup \mathcal{W}_A(\mathcal{G}) = V$: depuis n'importe quelle position, l'un des deux joueurs a une stratégie gagnante. On dit que ces jeux sont *déterminés*.

Les stratégies définies précédemment sont des objets infinis. En effet, pour choisir le prochain coup, Ève peut considérer tout l'historique de la partie, qui est arbitrairement grand. Afin de décrire des objets finiment représentables, nous définissons des *stratégies à mémoire finie*. Formellement, une *structure de mémoire* $\mathcal{M} = (M, m_0, \mu)$ pour une arène $\mathcal{A}$ est donnée par un ensemble M d'états de mémoire, un état de mémoire initial $m_0 \in M$ et une fonction de mise à jour $\mu : M \times E \to M$. Une structure de mémoire peut être vue comme un automate synchronisé avec l'arène : partant de m_0, il lit la suite des arêtes produites. Lorsqu'une arête est empruntée, l'état courant est mis à jour par la fonction μ. Une stratégie à mémoire $\mathcal{M}$ a accès, au moment de choisir le prochain coup, seulement à deux informations : le sommet courant et l'état de mémoire courant. Elle est donc décrite par une fonction

$\nu : V_\circ \times M \to V$. Formellement, étant données une structure de mémoire $\mathcal{M}$ et une fonction ν, on définit la stratégie σ pour Ève par $\sigma(w \cdot v) = \nu(v, \mu^+(w \cdot v))$. (La fonction de mise à jour est étendue en une fonction $\mu^+ : V^+ \to M$ par $\mu^+(v) = m_0$ et $\mu^+(w \cdot u \cdot v) = \mu(\mu^+(w \cdot u), (u, v))$.) Une stratégie à mémoire $\mathcal{M}$ est à mémoire finie si M est un ensemble fini. Elle est sans mémoire, ou positionnelle, si M est un singleton : dans ce cas, le choix du coup suivant ne dépend que du sommet courant. De manière plus concise, une stratégie sans mémoire est donnée par une fonction $\sigma : V_\circ \to V$. Toutes ces définitions se transposent pour Adam.

On peut rendre explicite le produit synchronisé entre l'arène $\mathcal{A}$ et la structure de mémoire $\mathcal{M}$: définissons l'arène enrichie $\mathcal{A} \times \mathcal{M} = (V \times M, (V_\circ \times M, V_\square \times M), E \times \mu)$ où $E \times \mu$ est définie par $((v, m), (v', m')) \in E \times \mu$ si $(v, v') \in E$ et $\mu(m, (v, v')) = m'$. Il y a une bijection naturelle entre les parties de $\mathcal{A}$ et de $\mathcal{A} \times \mathcal{M}$, ainsi qu'entre les stratégies sans mémoire de $\mathcal{A} \times \mathcal{M}$ et les stratégies à mémoire $\mathcal{M}$ de $\mathcal{A}$. Il s'ensuit que si un joueur a une stratégie gagnante sans mémoire dans l'arène $\mathcal{A} \times \mathcal{M}$, alors il a une stratégie gagnante à mémoire $\mathcal{M}$ dans l'arène $\mathcal{A}$. Cette propriété importante sera utilisée plusieurs fois dans l'article.

Un *objectif d'accessibilité* pour un sous-ensemble de sommets F est rempli lorsqu'un des sommets de F a été visité : $\mathrm{Reach}(F) = \{v_0, v_1 \ldots \mid \exists p \in \mathbb{N}, v_p \in F\} \subseteq \Pi$. Les jeux de la forme $\mathcal{G} = (\mathcal{A}, \mathrm{Reach}(F))$ sont appelés jeux d'accessibilité. Pour déterminer si Ève gagne un jeu d'accessibilité, on calcule l'attracteur de F, dont le principe est réminiscent de la résolution des jeux à durée finie par “backward induction” due à (Zermelo, 1913). Définissons la suite $(\mathrm{Attr}_i^E(F))_{i \geq 0}$:

$$\begin{array}{lcll} \mathrm{Attr}_0^E(F) & = & F & \\ \mathrm{Attr}_{i+1}^E(F) & = & \mathrm{Attr}_i^E(F) & \cup \quad \{u \in V_\circ \mid \exists (u, v) \in E, v \in \mathrm{Attr}_i^E(F)\} \\ & & & \cup \quad \{u \in V_\square \mid \forall (u, v) \in E, v \in \mathrm{Attr}_i^E(F)\} \end{array}$$

Cette suite de sous-ensembles de V est croissante au sens de l'inclusion, donc stationnaire (car V est fini). Sa limite est notée $\mathrm{Attr}^E(F)$, “l'attracteur de F”, et est atteinte en au plus $n - 1$ itérations. Énonçons ses propriétés :

– il existe une stratégie sans mémoire pour Ève qui assure depuis $\mathrm{Attr}_i^E(F)$ d'atteindre F en au plus i coups,

– il existe une stratégie sans mémoire pour Adam qui assure depuis $V \setminus \mathrm{Attr}^E(F)$ de toujours rester dans $V \setminus \mathrm{Attr}^E(F)$, et donc de ne jamais visiter F.

Ceci implique en particulier que $\mathcal{W}_E(\mathcal{A}, \mathrm{Reach}(F))$ est exactement $\mathrm{Attr}^E(F)$, donc que l'on peut déterminer le gagnant dans un jeu d'accessibilité en calculant l'attracteur (en temps $O(n + m)$), mais également que les deux joueurs ont des stratégies gagnantes sans mémoire.

Un *objectif d'accessibilité généralisée* est une conjonction de k objectifs d'accessibilité. Ainsi, la donnée de k sous-ensembles de sommets $F_1, \ldots, F_k$ définit l'objectif d'accessibilité généralisée

$$\mathrm{GenReach}(F_1, \ldots, F_k) = \{v_0, v_1 \ldots \mid \forall i \in \{1, \ldots, k\}, \exists p_i \in \mathbb{N}, v_{p_i} \in F_i\}.$$

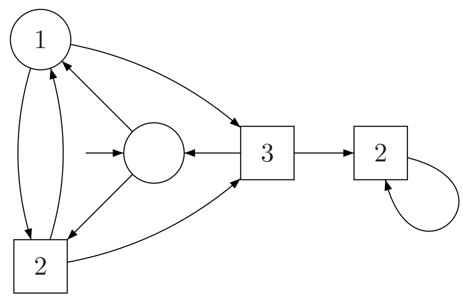

FIGURE *1. Un exemple de jeu d'accessibilité généralisée.*

Souvent, nous associerons à chaque ensemble F_i une couleur; dans ce cadre, l'objectif d'accessibilité généralisée est de voir chacune des k couleurs au moins une fois, dans n'importe quel ordre. Les jeux de la forme $\mathcal{G} = (\mathcal{A}, \mathrm{GenReach}(F_1, \ldots, F_k))$ sont appelés jeux d'accessibilité généralisée. Si $V_\Box$ ou $V_\circ$ est vide, on parle de jeu d'accessibilité généralisée à un joueur.

EXEMPLE 1. — Considérons l'arène représentée en Figure 1. On obtient un jeu d'accessibilité généralisée en considérant l'objectif $\mathrm{GenReach}(F_1, F_2, F_3)$. L'objectif d'Ève est de visiter le sommet de F_1, l'un des deux sommets de F_2 et le sommet de F_3. On peut vérifier qu'Ève gagne depuis le sommet initial central marqué par une flèche entrante. $\Box$

Contributions. Les résultats que nous présentons dans cet article sont les suivants :

– Nous commençons par prouver que résoudre les jeux d'accessibilité généralisée est un problème PSPACE-complet. Avec les mêmes idées, nous déduisons que dans le cas où Ève contrôle tous les sommets, le problème devient NP-complet, et qu'il est polynomial si Adam contrôle tous les sommets. Cependant, si l'on fixe le nombre de couleurs, alors le problème à deux joueurs devient polynomial.

– Nous étudions la mémoire utilisée pour implémenter des stratégies gagnantes pour les deux joueurs, en fonction du nombre k de couleurs. Nous présentons des bornes supérieures et inférieures : pour toute arène, si Ève a une stratégie gagnante, alors elle a une stratégie gagnante qui utilise $2^k - 1$ états de mémoire, et cette borne est atteinte. Les résultats sont les mêmes pour Adam, pour la borne $\binom{k}{\lfloor k/2 \rfloor}$.

– Ensuite, nous considérons des sous-classes de jeux où chaque couleur ne peut apparaître qu'un nombre constant de fois. Ceci révèle une trichotomie : si chaque couleur apparaît au plus trois fois, alors la résolution de ces jeux est, comme le cas général, PSPACE-complet. Cependant, si chaque couleur apparaît une seule fois, alors le problème devient polynomial. Si chaque couleur apparaît au plus deux fois, le problème est polynomial dans le cas où Ève contrôle tous les sommets, mais la complexité exacte dans le cas à deux joueurs est laissée ouverte.

3. Complexité des jeux d'accessibilité généralisée

Dans cette partie nous commençons par prouver que le problème de résoudre un jeu d'accessibilité généralisée est PSPACE-complet. La borne inférieure de complexité résulte d'une réduction depuis le problème d'évaluation d'une formule quantifiée booléenne sous forme normale conjonctive, noté QBF. Malgré sa complexité dans le cas général, nous montrons ensuite qu'en fixant le nombre de couleurs, décider le gagnant peut être fait en temps polynomial : le problème est dans la classe de complexité FPT pour le paramètre "nombre de couleurs".

Nous nous intéressons ensuite aux cas où un seul des deux joueurs n'a de choix. Si Ève contrôle tous les sommets, le problème est NP-complet, tandis que si Adam contrôle tous les sommets, il est polynomial.

Enfin, nous considérons la mémoire nécessaire pour les deux joueurs. Nous présentons des bornes optimales : Ève a besoin de $2^k - 1$ états de mémoire et Adam de $\binom{k}{\lfloor k/2 \rfloor}$, où k est le nombre de couleurs.

3.1. Complexité des jeux d'accessibilité généralisée

Nous présentons dans cette sous-partie un résultat surprenant : alors qu'un jeu d'accessibilité se résout en temps $O(n+m)$, les jeux d'accessibilité généralisée requièrent un espace polynomial ! La difficulté de ces jeux provient de l'invariance des rôles des k objectifs d'accessibilité : puisqu'ils peuvent être remplis dans un ordre quelconque, une stratégie gagnante va devoir considérer simultanément plusieurs scénarios, selon les coups de l'adversaire. Nous illustrerons ces propos dans l'exemple 2.

Les jeux d'accessibilité généralisée peuvent être utilisés pour évaluer des formules quantifiées booléennes. Nous définissons une réduction depuis le problème QBF vers le problème de décider le gagnant dans un jeu d'accessibilité généralisée, d'abord dans le cas général, puis illustré sur un exemple.

Considérons une formule quantifiée booléenne $Q_1 x_1\ Q_2 x_2\ \ldots Q_n x_n\ \phi$, où pour chaque i, Q_i est un quantificateur universel ($\forall$) ou existentiel ($\exists$), et ϕ est une formule propositionnelle, conjonctions de 3-clauses, *i.e*

$$\phi = \bigwedge_{i \in \{1,\ldots,k\}} \ell_{i,1}\ \vee\ \ell_{i,2}\ \vee\ \ell_{i,3}$$

et $\ell_{i,j}$ est le littéral x_v ou $\neg x_v$ pour un certain v. Construisons un jeu d'accessibilité généralisée dans lequel Ève gagne si et seulement si la formule est vraie. Intuitivement, les deux joueurs vont choisir séquentiellement les valeurs des variables, en suivant l'ordre de quantification depuis la variable la plus externe. Ève choisit les valeurs des variables existentielles, et Adam celles de variables universelles. Formellement, le jeu est le suivant :

– pour chaque variable x_i, on a deux sommets x_i et $\overline{x_i}$ et un sommet v_i qui mène vers x_i et $\overline{x_i}$. Ce sommet appartient à Ève si x_i est quantifiée existentiellement, et à Adam si elle est quantifiée universellement;

– pour chaque variable x_i avec $i < n$, les sommets x_i et $\overline{x_i}$ ont une unique arête vers le sommet de choix suivant, v_{i+1};

– les sommets x_n et $\overline{x_n}$ bouclent sur eux-mêmes;

– pour chaque clause $\{\ell_{i,1}, \ell_{i,2}, \ell_{i,3}\}$, il y a un objectif d'accessibilité F_i qui contient les sommets correspondants;

– l'objectif d'accessibilité généralisée est donné par $\mathrm{GenReach}(F_1, \ldots, F_k)$.

Le sommet initial est v_1. Il y a une bijection naturelle entre les distributions de valeurs de vérité des variables et les parties dans le jeu ; de plus une distribution de valeurs de vérité satisfait la formule ϕ si et seulement si la partie satisfait l'objectif d'accessibilité généralisée. L'ordre d'évaluation des variables étant identique dans la formule et dans le jeu, ceci implique qu'Ève a une stratégie gagnante si et seulement si la formule est vraie.

Le problème QBF étant PSPACE-difficile, il en est de même du problème de résolution des jeux d'accessibilité généralisée.

EXEMPLE 2. — Appliquons la construction précédente sur la formule quantifiée booléenne suivante

$$\forall x \, \forall y \, \exists z \, \exists t \qquad (x \vee y \vee t) \wedge (\neg x \vee \neg t) \wedge (y \vee \neg z \vee \neg t) \wedge (\neg y \vee z) \; .$$

Le jeu construit est illustré en Figure 2. L'objectif d'accessibilité généralisée est $\mathrm{GenReach}(\{x, y, t\}, \{\overline{x}, \overline{t}\}, \{y, \overline{z}, \overline{t}\}, \{\overline{y}, z\})$. Le premier ensemble d'accessibilité est représenté par des cercles en gras, le deuxième par des pointillés, le troisième par des double cercles et le quatrième par des cercles grisés. Dans ce jeu, Ève gagne, voici une stratégie gagnante :

– choisir t si Adam a choisi $\overline{x}$ et $\overline{t}$ s'il a choisi x;

– choisir z si Adam a choisi y et $\overline{z}$ s'il a choisi $\overline{y}$.

Dans cet exemple, Ève gagne mais elle ne peut pas prévoir à l'avance dans quel ordre les objectifs d'accessibilité vont être remplis.

□

Nous décrivons maintenant un algorithme pour résoudre les jeux d'accessibilité généralisée. L'idée développée ici est de réduire un jeu d'accessibilité généralisée à un jeu d'accessibilité (simple), exponentiellement plus grand. Cette réduction sera utilisée plusieurs fois dans cet article, pour des résultats de complexité mais également pour capturer la quantité exacte de mémoire nécessaire pour chaque joueur.

De manière informelle, au moment d'effectuer un coup, la seule information pertinente (pour les deux joueurs) est l'ensemble des objectifs qui ont déjà été rem-

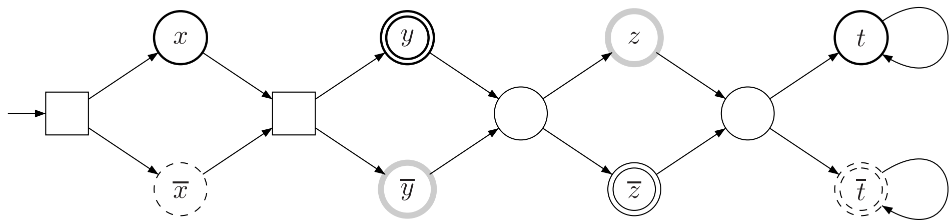

FIGURE 2. *Illustration de la réduction depuis* QBF *vers les jeux d'accessibilité généralisée.*

plis. Nous définissons une structure de mémoire qui maintient cette information, et construisons le produit synchronisé de l'arène avec cette structure de mémoire.

Soit $\mathcal{G} = (\mathcal{A}, \mathrm{GenReach}(F_1, \ldots, F_k))$ un jeu d'accessibilité généralisée et v_0 un sommet initial. La structure de mémoire $\mathcal{M}$ est définie par (M, m_0, μ), où M est l'ensemble des sous-ensembles de $\{1, \ldots, k\}$, m_0 est $\{i \mid v_0 \in F_i\}$, et $\mu(S, (v, v')) = S \cup \{i \mid v' \in F_i\}$. Définissons le jeu d'accessibilité $\mathcal{G} \times \mathcal{M} = (\mathcal{A} \times \mathcal{M}, \mathrm{Reach}(F))$ où $F = \{(v, S) \mid S = \{1, \ldots, k\}\}$. Une partie dans $\mathcal{G}$ depuis v_0 est gagnante si et seulement si la partie correspondante est gagnante dans $\mathcal{G} \times \mathcal{M} = (\mathcal{A} \times \mathcal{M}, \mathrm{Reach}(F))$ depuis (v_0, m_0). Il s'ensuit qu'Ève gagne depuis v_0 dans $\mathcal{G}$ si et seulement elle gagne depuis (v_0, m_0) dans $\mathcal{G} \times \mathcal{M}$.

Cette construction est utilisée dans un premier temps pour obtenir le résultat suivant.

LEMME 3. — *Soit $\mathcal{G} = (\mathcal{A}, \mathrm{GenReach}(F_1, \ldots, F_k))$ un jeu d'accessibilité généralisée et v_0 un sommet initial. Si Ève gagne depuis v_0 dans $\mathcal{G}$, alors elle a une stratégie gagnante qui remplit son objectif en au plus $n \cdot k$ coups.*

PREUVE 4. — Considérons le jeu d'accessibilité $\mathcal{G} \times \mathcal{M}$, alors Ève gagne depuis (v_0, m_0) dans $\mathcal{G} \times \mathcal{M}$, soit $\sigma_{\mathcal{M}}$ une stratégie gagnante positionnelle.

Remarquons qu'au cours d'une partie dans $\mathcal{G} \times \mathcal{M}$, la suite des états de mémoire est croissante pour l'inclusion (c'est l'ensemble des objectifs remplis jusque là). En particulier, ceci implique qu'une partie de $n \cdot k$ coups contient au maximum $k+1$ états de mémoire différents. Supposons par l'absude qu'il existe une partie où Ève joue la stratégie $\sigma_{\mathcal{M}}$ et qui ne remplit pas son objectif en $n \cdot k$ coups, alors l'état de mémoire $\{1, \ldots, k\}$ n'est pas atteint, et cette partie contient au maximum k états de mémoire différents. Ceci implique qu'il existe un sommet (v, m) dans cette partie qui est répété deux fois. La stratégie $\sigma_{\mathcal{M}}$ étant positionnelle, répéter cette boucle à l'infini décrit une partie où Ève joue la stratégie $\sigma_{\mathcal{M}}$, mais dont l'état de mémoire se stabilise à m, donc la partie est perdante, contradiction.

La stratégie $\sigma_{\mathcal{M}}$ assure de remplir l'objectif d'accessibilité en au plus $n \cdot k$ coups dans $\mathcal{G} \times \mathcal{M}$, donc la stratégie correspondante σ utilisant $\mathcal{M}$ pour structure de mémoire assure de remplir l'objectif d'accessibilité généralisée en au plus $n \cdot k$ coups dans $\mathcal{G}$. ■

THÉORÈME 5 (Complexité des jeux d'accessibilité généralisée). — *Le problème de résoudre un jeu d'accessibilité généralisée est* PSPACE-*complet.*

PREUVE 6. — La réduction décrite ci-dessus depuis QBF implique que le problème est PSPACE-difficile.

Montrons que, réciproquement, il existe un algorithme pour résoudre un jeu d'accessibilité généralisée en espace polynomial. L'argument que nous développons ici est assez général. Tout d'abord, nous utilisons un résultat classique de complexité, à savoir $\mathrm{PSPACE} = \mathrm{APTIME}$: les machines de Turing alternantes fonctionnant en temps polynomial ont exactement le même pouvoir d'expression que les machines de Turing fonctionnant en espace polynomial.

Les deux propriétés importantes des jeux d'accessibilité généralisée qui permettent de les résoudre en temps polynomial alternant sont les suivantes :

– si Ève gagne, alors elle a une stratégie gagnante qui remplit son objectif en au plus $n \cdot k$ coups (Lemme 3),

– étant donné un chemin de longueur $n \cdot k$, on peut décider en temps polynomial s'il remplit l'objectif d'accessibilité généralisée.

En s'appuyant sur ces deux propriétés, pour résoudre un jeu d'accessibilité généralisée il suffit de le simuler pendant $n \cdot k$ coups avec la machine de Turing alternante : les sommets contrôlés par Ève sont interprétés par des disjonctions et ceux d'Adam par des conjonctions. Un chemin est accepté s'il remplit l'objectif d'accessibilité généralisée, c'est-à-dire s'il visite chaque F_i. Ainsi, cette machine accepte si et seulement si Ève gagne. ■

Nous donnons une autre application de la réduction décrite ci-dessous vers les jeux d'accessibilité simples: il existe un algorithme pour résoudre les jeux d'accessibilité généralisée qui fonctionne en temps polynomial si l'on suppose fixé le nombre de couleurs. Ce résultat entre en analogie avec le résultat suivant : le problème QBF est dans la classe de complexité FPT pour le paramètre "nombre de clauses".

THÉORÈME 7 (Jeux d'accessibilité généralisée avec un nombre constant de couleurs). — *Le problème de résoudre les jeux d'accessibilité généralisée est* FPT *(fixed-parameter tractable, soluble à paramètre fixé), pour le paramètre "nombre de couleurs".*

PREUVE 8. — Remarquons que le jeu d'accessibilité (simple) $\mathcal{G} \times \mathcal{M}$ est de taille $n \cdot 2^k$, donc polynomiale si k est fixé. Puisque l'on sait résoudre les jeux d'accessibilité (simples) en temps linéaire, on peut résoudre les jeux d'accessibilité généralisée en temps $2^k \cdot O(n+m)$. ■

3.2. Résoudre les jeux à un joueur

Dans cette sous-partie, on s'intéresse aux cas particuliers où $V_\circ$ ou $V_\Box$ est vide.

THÉORÈME 9 (Complexité des jeux d'accessibilité généralisée à un joueur). —

– *Résoudre les jeux d'accessibilité généralisée où Ève contrôle tous les sommets est* NP*-complet.*

– *Résoudre les jeux d'accessibilité généralisée où Adam contrôle tous les sommets est dans* PTIME*.*

PREUVE 10. — Commençons par le cas des jeux d'accessibilité généralisée où Ève contrôle tous les sommets, *i.e* $V_\Box = \emptyset$. Dans la réduction précédente depuis QBF, considérons le cas où toutes les variables de la formule sont quantifiées existentiellement : c'est le problème 3-SAT, satisfaisabilité d'une formule booléenne sous forme normale conjonctive (non quantifiée), qui est NP-difficile. Les jeux obtenus en appliquant cette réduction sont des jeux à un joueur, où Ève contrôle tous les sommets. Il en résulte que résoudre les jeux d'accessibilité généralisée à un joueur est NP-difficile.

Réciproquement, décrivons un algorithme non-déterministe qui fonctionne en temps polynomial, résolvant les jeux d'accessibilité généralisée à un joueur. C'est un cas particulier de l'algorithme décrit dans la preuve du théorème 5 : tous les sommets étant contrôlés par Ève, la machine de Turing n'utilise pas tout le pouvoir d'une machine alternante, puisque tous ses états ont une sémantique disjonctive. En d'autres termes, l'algorithme devine un chemin de longueur $n \cdot k$ et vérifie qu'il satisfait l'objectif d'accessibilité généralisée. Ceci montre que résoudre les jeux d'accessibilité généralisée à un joueur est NP-complet.

Considérons à présent le cas où Adam contrôle tous les sommets. Notons $\mathrm{GenReach}(F_1, \ldots, F_k)$ l'objectif d'accessibilité généralisée, alors la région gagnante d'Adam est $V \setminus \bigcap_i \mathrm{Attr}^E(F_i)$, qui est calculable en temps quadratique. ■

3.3. Quantité de mémoire nécessaire et suffisante

Dans cette sous-partie, nous nous intéressons à l'implémentation des stratégies gagnantes pour les deux joueurs. Nous considérons des stratégies à mémoire finie, pour lesquelles on veut minimiser le nombre d'états de mémoire. Notre démarche s'abstrait du graphe, et étudie la quantité de mémoire nécessaire et suffisante pour implémenter des stratégies gagnantes en fonction de k, le nombre de couleurs.

Présentons pour commencer des bornes supérieures :

LEMME 11 (Bornes supérieures de mémoire). — *Pour tout jeu d'accessibilité généralisée* $\mathcal{G} = (\mathcal{A}, \mathrm{GenReach}(F_1, \ldots, F_k))$,

– *si Ève gagne, alors elle a une stratégie gagnante avec* $2^k - 1$ *états de mémoire;*

– *si Adam gagne, alors il a une stratégie gagnante avec* $\binom{k}{\lfloor k/2 \rfloor}$ *états de mémoire.*

Pour obtenir ce résultat, nous nous appuyons à nouveau sur la structure de mémoire décrite dans la section précédente. Rappelons sa définition : $\mathcal{M} = (M, m_0, \mu)$ est définie par M est l'ensemble des sous-ensembles de $\{1, \ldots, k\}$, $m_0 = \{i \mid v_0 \in F_i\}$ et $\mu(S, (v, v')) = S \cup \{i \mid v \in F_i\}$. La propriété importante est la suivante : une partie du jeu $\mathcal{G}$ depuis v_0 est gagnante si et seulement si la partie correspondante est gagnante dans le jeu d'accessibilité $\mathcal{G} \times \mathcal{M} = (\mathcal{A} \times \mathcal{M}, \mathrm{Reach}(F))$ depuis (v_0, m_0), pour $F = \{(v, S) \mid S = \{1, \ldots, k\}\}$.

PREUVE 12. — Soit $\mathcal{G} = (\mathcal{A}, \mathrm{GenReach}(F_1, \ldots, F_k))$ un jeu d'accessibilité généralisée et v_0 sommet initial. Dans le jeu d'accessibilité (simple) $\mathcal{G} \times \mathcal{M}$, chaque joueur a une stratégie gagnante sans mémoire, donc dans $\mathcal{G}$ chaque joueur a une stratégie gagnante à mémoire $\mathcal{M}$.

Les deux stratégies utilisent donc 2^k états de mémoire. Pour obtenir les bornes annoncées pour chacun des deux joueurs, il faut être plus précis :

– Ève n'a pas besoin d'un état de mémoire particulier pour se souvenir que toutes les couleurs ont été visitées, puisque dans ce cas elle a déjà gagné. Donc, elle peut toujours gagner avec $2^k - 1$ états de mémoire: la structure de mémoire $\mathcal{M}$ est la même, sauf que l'état $\{1, \ldots, k\}$ n'existe plus, et dans la fonction de mise à jour, $\{1, \ldots, k\}$ est remplacé par n'importe quel état, ce qui n'importe pas puisqu'à ce stade la partie est déjà gagnée.

– Pour Adam, l'observation principale est la suivante : s'il gagne dans $\mathcal{G} \times \mathcal{M}$ depuis (v, S), alors pour tout $S' \subseteq S$, il gagne depuis (v, S') *en jouant la même stratégie*.

Soit $v \in V$, considérons l'ensemble des sous-ensembles S tel que (v, S) soit gagnant pour Adam. Ses éléments maximaux (pour l'inclusion) sont incomparables, donc il y en a au plus $\binom{k}{\lfloor k/2 \rfloor}$, notons-les $S_1(v), \ldots, S_p(v)$. L'idée est la suivante : depuis v, Adam ne considère que l'un des p "pire cas", à savoir $S_1(v), \ldots, S_p(v)$.

Définissons une structure de mémoire $\mathcal{M}' = (M', i_0, \mu')$. D'abord, $M' = \{1, \ldots, \binom{k}{\lfloor k/2 \rfloor}\}$. Si le sommet initial est v_0, alors (v_0, m_0) est gagnant pour Adam, donc il existe i_0 tel que $S_{i_0}(v_0)$ contienne m_0, l'état de mémoire initial est un tel i_0. Définissons la fonction de mise à jour par $\mu'(i, (v, v'))$ est un j tel que $S_j(v')$ contienne $\mu(S_i(v), (v, v'))$, s'il existe.

Construisons à présent la fonction ν choisissant le coup suivant depuis (v, i). Pour tout (v, S) gagnant pour Adam, il existe une transition vers $(v', \mu(S, (v, v')))$ également gagnant pour Adam. Appliqué à $(v, S_i(v))$, ceci assure l'existence d'un sommet v' : on définit $\nu(v, i) = v'$.

Pour toute partie (finie) w finissant en $v \in V$ où Adam joue cette stratégie, l'état de mémoire obtenu (*i.e* $\mu'^+(w)$) est un i tel que $S_i(v)$ contienne l'ensemble des couleurs visitées, *i.e* $\mu^+(w) \subseteq S_i(v)$. Vérifions cette propriété par récurrence. Elle est vraie initialement par construction, car $m_0 = \mu^+(v_0) \subseteq S_{i_0}(v_0)$. Supposons que la dernière transition est $(v, v') \in E$. Puisque Adam gagne depuis $(v, S_i(v))$, il gagne également depuis $(v', \mu(S_i(v), (v, v')))$: soit parce qu'Ève ne peut pas sortir de la région gagnante d'Adam, ou par construction de la stratégie d'Adam. Donc il existe j tel que

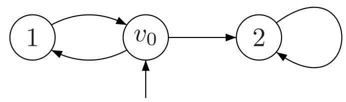

FIGURE 3. *Ève ne peut pas jouer dans le meilleur des cas.*

$S_j(v')$ contienne $\mu(S_i(v),(v,v'))$, autrement dit $\mu'(i,(v,v'))$ est bien défini. Par hypothèse d'induction, $\mu^+(w) \subseteq S_i(v)$, donc $\mu^+(w \cdot v') \subseteq \mu(S_i(v),(v,v')) \subseteq S_j(v')$.
L'invariant implique que la partie reste pour toujours dans la région gagnante d'Adam, donc est gagnante pour Adam. Le nombre d'états de mémoire utilisé est $\binom{k}{\lfloor k/2 \rfloor}$.

■

Remarquons que la propriété essentielle qui permet de fusionner des états de mémoire pour Adam ne se transpose pas pour Ève : il est vrai que si Ève gagne dans $\mathcal{G} \times \mathcal{M}$ depuis (v,S), et que $S \subseteq S'$, alors elle gagne depuis (v,S'), mais pas nécessairement en jouant la même stratégie. Ceci est illustré par la Figure 3, où Ève gagne en allant d'abord à gauche puis à droite, pour voir à la fois 1 et 2. Pour gagner depuis $(v_0,\{1\})$, Ève doit aller à droite, mais depuis $(v_0,\emptyset)$ elle doit aller à gauche. Elle gagne dans les deux états de mémoire $\emptyset$ et $\{1\}$, mais ne peut pas jouer la même chose.

Les bornes supérieures présentées sont atteintes :

LEMME 13 (Bornes inférieures de mémoire). — *Pour tout $k \geq 1$,*

– *il existe $\mathcal{G} = (\mathcal{A}, \mathrm{GenReach}(F_1,\ldots,F_k))$ jeu d'accessibilité généralisée, où Ève a besoin de $2^k - 1$ états de mémoire pour gagner;*

– *il existe $\mathcal{G} = (\mathcal{A}, \mathrm{GenReach}(F_1,\ldots,F_k))$ jeu d'accessibilité généralisée, où Adam a besoin de $\binom{k}{\lfloor k/2 \rfloor}$ états de mémoire pour gagner.*

PREUVE 14. — Décrivons d'abord un jeu d'accessibilité généralisée où Ève a besoin de $2^k - 1$ états de mémoire pour gagner. L'arène est illustrée en Figure 4, pour $k = 5$, et apparaît dans (Chatterjee *et al.*, 2011).

Un sommet étiqueté par i appartient à F_i, et un sommet étiqueté par $\overline{i}$ appartient à tous les F_j sauf F_i. Une partie commence au cœur c de la fleur ; d'abord Adam choisit un pétale i, puis Ève choisit soit de voir la couleur i avant de revenir au cœur, (alors la partie continue), soit de voir toutes les couleurs sauf i et d'arrêter la partie. Ève gagne avec la stratégie suivante : la première fois qu'Adam choisit le pétale i, elle retourne au cœur, la deuxième fois, elle arrête la partie. Cette stratégie est facilement implémentée avec 2^k états de mémoire. Un de ces états est en fait inutile : c'est celui qui correspond au cas où tous les pétales ont été vus une fois, alors Ève a déjà gagné. Montrons qu'il n'y a pas de stratégie gagnante pour Ève qui utilise moins de $2^k - 1$ états de mémoire. Soit σ une stratégie à mémoire $\mathcal{M}$ avec moins de $2^k - 1$ états de mémoire, définie par la fonction ν. Pour chaque état de mémoire m, considérons $S_m = \{i \mid \nu(v_i, \mu(m,(c,v_i))) = \overline{i}\}$, l'ensemble des pétales depuis lesquels Ève

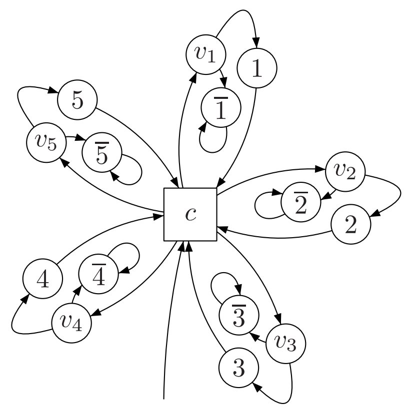

FIGURE 4. *Un jeu d'accessibilité généralisée où Ève a besoin de* $2^k - 1$ *états de mémoire pour gagner.*

choisirait d'arrêter le jeu si Adam proposait ce pétale. Puisqu'il y a moins de $2^k - 1$ états de mémoire, il existe un sous-ensemble strict X de $\{1, \ldots, k\}$ qui n'est pas de la forme S_m pour un état de mémoire m. Adam gagne contre la stratégie σ en choisissant à chaque étape, un pétale dans la différence symétrique de X et de S_m, où m l'état de mémoire courant d'Ève. En effet :

– si Adam joue pour toujours des pétales de X, alors Ève n'arrêtera jamais la partie et seulement les couleurs de X seront vues,

– ou alors lorsque Ève arrête la partie, la dernière fois que le jeton était sur le cœur, l'état de mémoire d'Ève était un m tel que $X \subsetneq S_m$, et le pétale choisi est un i qui appartient à $S_m \setminus X$, donc la couleur i n'a jamais été vue.

Décrivons à présent un jeu d'accessibilité généralisée où Adam a besoin de $\binom{k}{\lfloor k/2 \rfloor}$ états de mémoire pour gagner. Soit $k = 2p + 1$. La Figure 5 illustre un gadget où un joueur (ici, Ève) choisit (exactement) p couleurs parmi $2p + 1$. Ce gadget est utilisé trois fois :

1. d'abord Ève choisit p couleurs,
2. puis Adam choisit p couleurs,
3. puis Ève choisit p couleurs.

Pour gagner, Adam doit visiter exactement les mêmes couleurs qu'Ève a vues à la première étape, ce qui nécessite $\binom{k}{\lfloor k/2 \rfloor}$ états de mémoire, sinon au moins $p + 1$ couleurs ont été vues lorsque la troisième étape commence, et Ève gagne en choisissant les couleurs qui n'ont pas été vues, qui sont en nombre au plus p. ∎

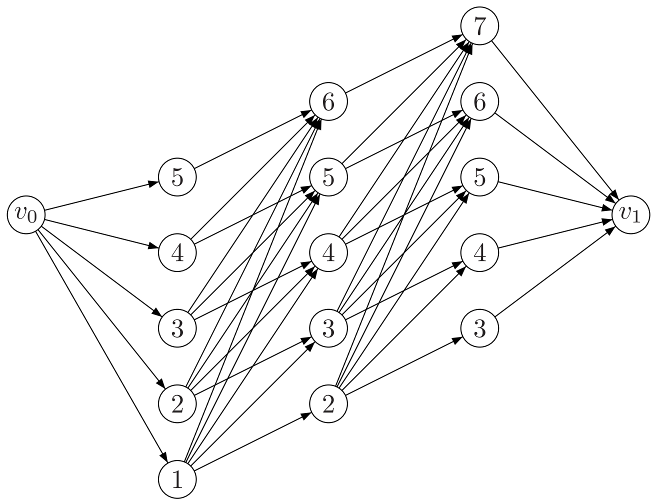

FIGURE 5. *Un gadget où Ève choisit exactement p couleurs parmi $2p + 1$.*

4. Restreindre la taille des objectifs d'accessibilité

Dans la partie précédente, nous montrons que dans le cas général, les jeux d'accessibilité généralisée sont d'une part difficiles à résoudre d'un point de vue complexité (PSPACE-complet), et d'autre part que la mémoire nécessaire pour implémenter des stratégies gagnantes est exponentielle (en le nombre de couleurs) pour les deux joueurs.

Dans cette partie, nous nous attachons donc à décrire des restrictions de ces jeux pour lesquelles à la fois la complexité de résolution et la mémoire utilisée par les stratégies gagnantes sont plus raisonnables.

Pour cela, examinons à nouveau la réduction depuis le problème QBF. Ce problème est PSPACE-difficile si les clauses contiennent trois littéraux, mais se résout en temps polynomial si on ne considère que des clauses de deux littéraux. Remarquons qu'à la taille des clauses d'une formule correspond la taille des objectifs d'accessibilité dans le jeu construit par la réduction. En particulier, si l'on suppose que chaque couleur apparaît au plus deux fois dans l'arène, alors la réduction depuis QBF n'implique pas que ce problème soit PSPACE-difficile. Nous considérons donc dans cette partie les deux classes de jeux suivantes :

– chaque couleur apparaît une seule fois dans l'arène;

– chaque couleur apparaît au plus deux fois dans l'arène.

4.1. Chaque couleur apparaît une seule fois

Dans ce cas, on peut résoudre le jeu en temps polynomial :

THÉORÈME 15 (Jeux d'accessibilité généralisée où chaque couleur apparaît une seule fois). — *Il existe un algorithme polynomial pour résoudre les jeux d'accessibilité généralisée où chaque couleur apparaît une seule fois.*

PREUVE 16. — Soit $\mathcal{G} = (\mathcal{A}, \mathrm{GenReach}(F_1, \dots, F_k))$ un jeu d'accessibilité généralisée où chaque couleur apparaît une seule fois. Notons v_i le sommet de F_i, pour tout i, et F l'union de F_i. Pour qu'Ève gagne, il faut et il suffit que le préordre défini sur F par $v \preceq v'$ si $v \in \mathrm{Attr}^E(v')$ est total. Lorsque c'est le cas, il existe une permutation f de $\{1, \dots, k\}$ telle que $v_{f(i)} \preceq v_{f(i+1)}$ pour tout $i \in \{1, \dots, k-1\}$, ce qui décrit une stratégie gagnante qui assure de voir $v_{f(1)}$, puis $v_{f(2)}$, etc.

Considérons deux cas :

– Si le préordre $\preceq$ est total, alors montrons que l'ensemble $\mathcal{W}_E(\mathcal{G})$ des positions gagnantes d'Ève est $\cap_i \mathrm{Attr}^E(v_i)$. Puisque $\preceq$ est total, il existe une permutation f de $\{1, \dots, k\}$ tel que pour tout $1 \leq i \leq k-1$, on a $v_{f(i)} \preceq v_{f(i+1)}$, *i.e* $v_{f(i)} \in \mathrm{Attr}^E(v_{f(i+1)})$. Remarquons que ceci implique $\mathrm{Attr}^E(v_{f(i)}) \subseteq \mathrm{Attr}^E(v_{f(i+1)})$, donc $\cap_i \mathrm{Attr}^E(v_i)$ est égal à $\mathrm{Attr}^E(v_{f(1)})$. Soit v dans $\cap_i \mathrm{Attr}^E(v_i)$, construisons une stratégie gagnante depuis v : elle joue successivement une stratégie d'attracteur vers $v_{f(1)}$, puis depuis ce sommet une stratégie d'attracteur vers $v_{f(2)}$, et ainsi de suite. Remarquons que cette stratégie est implémentable avec k états de mémoire. Réciproquement, si v n'est pas dans $\cap_i \mathrm{Attr}^E(v_i)$, alors Ève ne peut pas gagner, puisqu'Adam peut l'empêcher de voir une couleur.

– Si le préordre $\preceq$ n'est pas total, alors il existe v_i et v_j tel que $v_i \notin \mathrm{Attr}^E(v_j)$ et $v_j \notin \mathrm{Attr}^E(v_i)$. Dans ce cas Adam gagne partout, avec la stratégie suivante : “depuis v_i, éviter v_j, et réciproquement”. Remarquons que cette stratégie ne requiert que 2 états de mémoire.

Vérifier que l'ordre $\preceq$ est total se fait en temps quadratique, ce qui donne un algorithme polynomial pour résoudre les jeux d'accessibilité généralisée où chaque couleur apparaît une seule fois. ■

Un corollaire de ce théorème (ou plutôt de sa preuve) concerne des bornes supérieures de mémoire : si chaque couleur apparaît une seule fois, alors Ève a une stratégie gagnante utilisant k états de mémoire, et Adam une stratégie gagnante utilisant 2 états de mémoire. Il n'est pas difficile de voir que ces bornes sont atteintes.

4.2. Chaque couleur apparaît au plus deux fois

Considérons à présent le cas où chaque couleur apparaît au plus deux fois. Notre première étape est d'étendre la technique utilisée dans le cas précédent, où : “Ève gagne si et seulement s'il existe un ordre total sur les sommets coloriés”. Une approche similaire permet de résoudre les jeux en temps polynomial, mais seulement dans le cas où Ève joue seule, en réduisant ce problème à 2-SAT (il est connu que ce dernier problème est décidable en temps polynomial, et même linéaire).

THÉORÈME 17 (Jeux d'accessibilité généralisée à un joueur où chaque couleur apparaît au plus deux fois). — *Résoudre les jeux d'accessibilité généralisée à un joueur où chaque couleur apparaît au plus deux fois se fait en temps polynomial.*

PREUVE 18. — De même que dans la preuve précédente, considérons le préordre défini par $v \preceq v'$ si $v \in \mathrm{Attr}^E(v')$. Remarquons que dans le cas où Ève contrôle tous les sommets, $v \in \mathrm{Attr}^E(v')$ se réduit à "il existe un chemin de v vers v'".

Soit $F_i = \{x_i, y_i\}$ les objectifs d'accessibilité, et v_0 sommet initial. Quitte à écrire $F_i = \{x_i, x_i\}$, nous supposons que chaque couleur apparaît exactement deux fois. On dira qu'un sommet est colorié s'il appartient à au moins l'un des F_i.

Sans perte de généralités, supposons que pour tout i, il existe un chemin depuis v_0 vers x_i ou vers y_i. Ceci se vérifie facilement en temps polynomial, et si ce n'est pas vrai alors Ève ne peut pas gagner.

Il est clair que Ève gagne depuis v_0 si et seulement s'il existe $v_1, \ldots, v_k$ des sommets coloriés tel que :

1. pour tout i tel que $1 \leq i \leq k-1$, on a $v_i \preceq v_{i+1}$, et
2. chaque couleur apparaît dans $\{v_1, \ldots, v_k\}$.

Exprimons cette condition par une formule booléenne où chaque clause est de taille 2. Nous avons $2k$ variables X_i et Y_i, qui correspondent aux sommets x_i et y_i. Définissons la formule ϕ par :

$$\underbrace{\bigwedge\{(\neg X \vee \neg Y) \mid \text{ si } x \not\preceq y \text{ et } y \not\preceq x\}}_{(a)} \wedge \underbrace{\bigwedge_{1 \leq i \leq k} (X_i \vee Y_i)}_{(b)},$$

où x, y sont pris parmi les sommets coloriés.

Montrons qu'Ève gagne depuis v_0 si et seulement si ϕ est satisfaisable. Supposons qu'Ève gagne depuis v_0: soit $v_1, \ldots, v_k$ donnés par l'équivalence précédente, considérons la distribution de valeurs de vérité qui rend vraies les variables correspondant à $v_1, \ldots, v_k$, et fausses les autres. Cette distribution satisfait la formule ϕ. En effet, la condition 2. assure que les clauses de (b) sont satisfaites, et pour les clauses de (a), soit x, y tel que $x \not\preceq y$ et $y \not\preceq x$, si x est l'un des v_i, alors y ne peut l'être, donc $\neg X \vee \neg Y$ est vraie. Réciproquement, supposons que ϕ soit satisfaisable, et fixons une distribution de valeurs de vérité la satisfaisant. Les clauses de (a) assurent que l'ordre $\preceq$ est total parmi les sommets dont les variables correspondantes sont vraies. Les clauses de (b) assurent qu'il existe au moins un sommet de chaque couleur dont la variable correspondante est vraie. La conjonction de ces deux propriétés implique la condition décrite précédemment, donc Ève gagne.

Pour décider si Ève gagne depuis v_0, il suffit donc de décider si la formule ϕ est satisfaisable, ce qui se fait en temps polynomial. ■

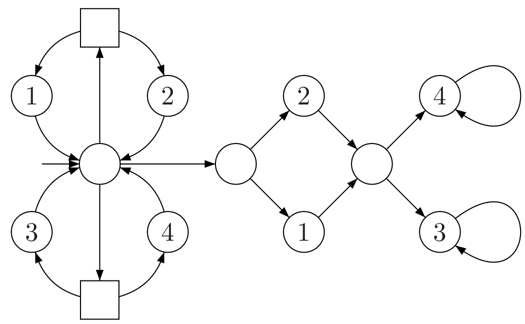

FIGURE 6. *Un jeu d'accessibilité généralisée où chaque couleur apparaît au plus deux fois et où Ève a besoin de* $2^{\lfloor k/2 \rfloor + 1} - 1$ *états de mémoire pour gagner.*

Cette technique ne semble pas s'étendre au cas général des jeux à deux joueurs. Nous ne connaissons pas la complexité exacte des jeux d'accessibilité généralisée où chaque couleur apparaît au plus deux fois. Dans le reste de cette sous-partie, nous donnons quelques pistes de recherche, en considérant la quantité de mémoire nécessaire pour implémenter des stratégies gagnantes dans ce cas.

La mémoire nécessaire pour implémenter une stratégie gagnante pour Ève est encore exponentielle, ainsi que l'illustre la Figure 6 pour $k = 4$. Les deux sommets étiquetés i sont ceux de couleur i. Dans le jeu d'accessibilité décrit, chaque couleur apparaît au plus deux fois, et Ève a besoin de $2^{\lfloor k/2 \rfloor + 1} - 1$ états de mémoire pour gagner. L'arène est divisée en deux parties : à gauche, une "fleur" avec $\lfloor k/2 \rfloor$ pétales, et à droite une arène où Ève contrôle tous les sommets. Le jeu commence au cœur de la fleur. Pour gagner, Ève doit d'abord proposer chaque pétale : pour chacun, Adam choisit une couleur parmi les deux associées à ce pétale. Ensuite, Ève déplace le jeton vers la partie de droite, où elle peut voir exactement une des deux couleurs pour chaque pétale. Elle doit donc se souvenir des $\lfloor k/2 \rfloor$ choix qu'Adam a fait, pour les inverser : si Adam a choisi la couleur 1, alors la couleur 2 n'a pas été vue, donc Ève doit la choisir. Proposer chaque pétale et se souvenir de tous ces choix nécessite $2^{\lfloor k/2 \rfloor + 1} - 1$ états de mémoire, c'est la taille de l'arbre binaire complet de profondeur $\lfloor k/2 \rfloor + 1$.

La situation est différente pour Adam ; la mémoire nécessaire et suffisante dans ce cas demeure une question ouverte. La Figure 7, proposée par Christof Loeding, montre un jeu d'accessibilité généralisée où chaque couleur apparaît au plus deux fois, où Adam a besoin de 4 états de mémoire pour gagner. Les deux sommets étiquetés i sont ceux de couleur i. Le jeu commence par le sommet tout à gauche. Dans la première étape, Ève choisit trois des quatre couleurs, puis envoie le jeton vers le sommet de droite, l'unique contrôlé par Adam. Là, il a quatre options, chacune laissant à Ève l'opportunité de voir toutes les couleurs sauf une. Pour gagner, Adam doit se souvenir des quatre éventualités, ce qui nécessite 4 états de mémoire.

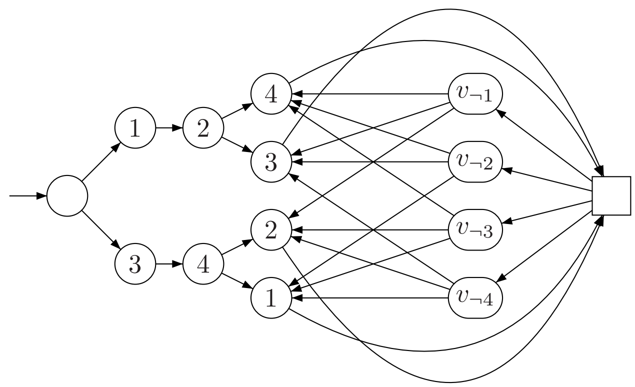

FIGURE 7. *Un jeu d'accessibilité généralisée où chaque couleur apparaît au plus deux fois et où Adam a besoin de* 4 *états de mémoire pour gagner.*

De manière surprenante, nous ne savons pas comment généraliser cet exemple pour obtenir une borne inférieure plus grande que 4. Nous ne savons pas si cette borne est suffisante, c'est-à-dire si dans toute arène, Adam a une stratégie gagnante utilisant 4 états de mémoire, ce qui serait surprenant mais demeure plausible. Ceci impliquerait que la résolution de ces jeux est dans la classe de complexité $\mathrm{coNP}^{\mathrm{NP}}$: pour décider si Adam gagne, on devine une stratégie gagnante utilisant 4 états de mémoire pour Adam, puis on compose cette stratégie avec le jeu, et on résout le jeu à un joueur obtenu.

Problèmes ouverts. La complexité exacte de la résolution des jeux d'accessibilité généralisée où chaque couleur apparaît au plus deux fois reste ouverte. Montrer que dans ce cas, Adam a une stratégie gagnante avec un nombre constant d'états de mémoire constituerait une première étape vers la résolution de cette question.

Bibliographie

Chatterjee K., Henzinger T. A., Horn F. (2011). The complexity of request-response games. In A. H. Dediu, S. Inenaga, C. Martín-Vide (Eds.), *LATA*, vol. 6638, p. 227-237. Springer.

Grädel E., Thomas W., Wilke T. (Eds.). (2002). *Automata, Logics, and Infinite Games: A Guide to Current Research [outcome of a Dagstuhl seminar, February 2001]* (vol. 2500). Springer-Verlag.

Horn F., Thomas W., Wallmeier N. (2008). Optimal Strategy Synthesis in Request-Response Games. In *Proceedings of the 6th International Symposium on Automated Technology for Verification and Analysis, ATVA'08*, vol. 5311, p. 361–373. Springer-Verlag.

Kupferman O., Piterman N., Vardi M. Y. (2007). From Liveness to Promptness. In *Proceedings of the 19th International Conference on Computer Aided Verification, CAV'07*, vol. 4590, p. 406–419. Springer-Verlag.

Zermelo E. (1913). Uber eine Anwendung der Mengenlehre auf die Theorie des Schachspiels. In *Proceedings of the 5th International Congress of Mathematicians, ICM'13*, vol. 2, p. 501–504. Cambridge University Press.

Zimmermann M. (2011). Optimal bounds in parametric LTL games. In *International Symposium on Games, Automata, Logics and Formal Verification, GandALF'11*, p. 146-161.

Reçu le 6 décembre 2011
Accepté le 29 avril 2013

Nathanaël Fijalkow. *Doctorant en co-tutelle au LIAFA (Paris) et à l'Institut d'Informatique de l'Université de Varsovie.*

Florian Horn. *Chercheur au CNRS, rattaché au LIAFA (Paris).*