Pareto-Optimalité pour la Vérification et la Synthèse dans les Jeux sur Graphe

Clément Tamines

Septembre 2022

Sommaire

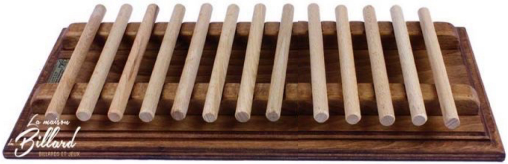
1. Pareto-Optimalité pour la Vérification et la Synthèse dans les **Jeux sur Graphe**
2. Pareto-Optimalité pour la **Vérification et la Synthèse** dans les Jeux sur Graphe
3. **Pareto-Optimalité** pour la Vérification et la Synthèse dans les Jeux sur Graphe

Sommaire

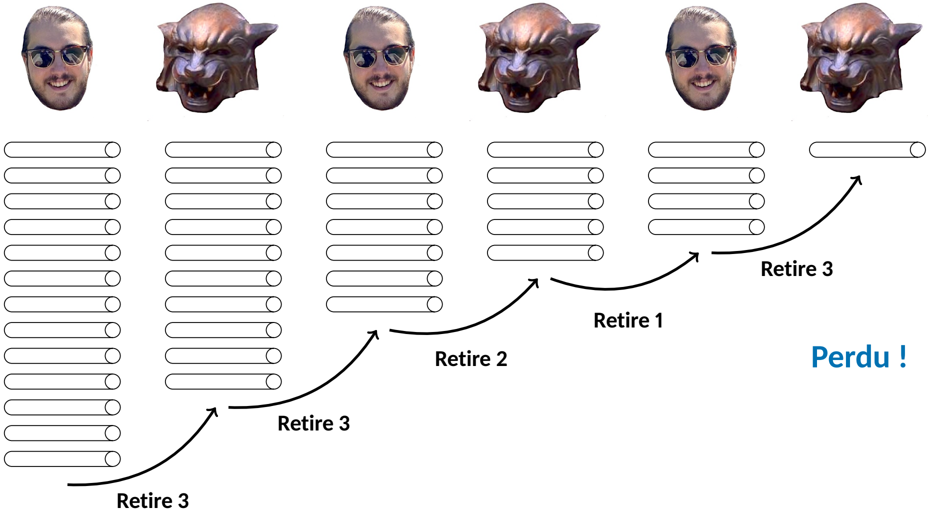
Jeu de Nim

Jeu à **deux joueurs**

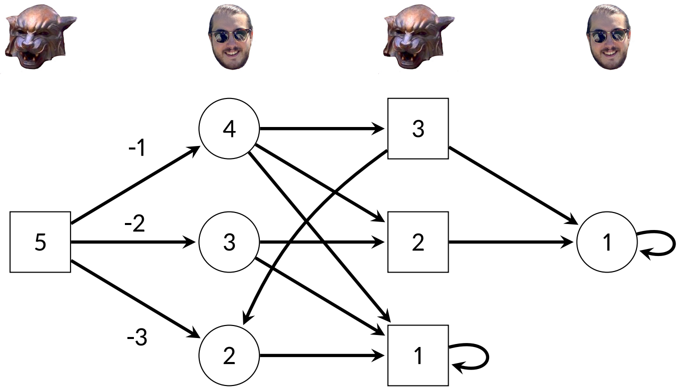
- commence avec **13** bâtonnets
- **à tour de rôle** on retire **1, 2, ou 3** bâtonnets
- la personne qui retire **le dernier** bâtonnet a **perdu**



Déroulement d’une Partie



Modéliser avec un Graphe

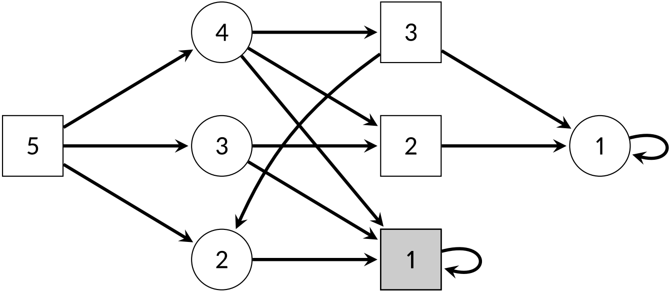


Graphe dirigé: **nœuds** (V) et **arcs** (E)

Deux joueurs: **Joueur 0** (cercles) et **Joueur 1** (carrés)

Arène de jeu: $G = (V, V_0, V_1, E, v_0)$ avec (V, E) un graphe dirigé

Partie et Objectifs



Partie: chemin infini depuis le **noeud initial**: $\rho = \boxed{5}\ (4)\ \boxed{1}\ \boxed{1}\ \boxed{1} \ldots$

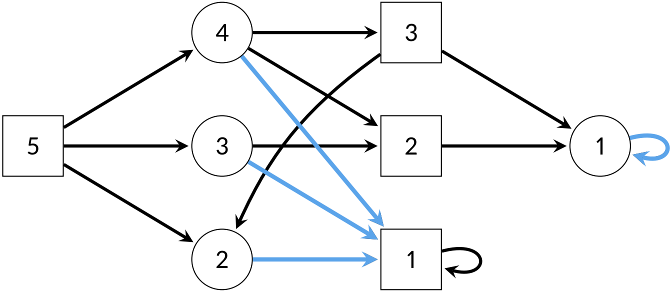
Objectif Ω_i pour le Joueur $i \in \{0, 1\}$:

- sous-ensemble de parties, ρ **satisfait** Ω_i si $\rho \in \Omega_i$
- $\text{Reach}(T)$: parties qui *visitent* $T \subseteq V$

Stratégie

Stratégie $\sigma_0 : V^* \times V_0 \to V$ **dicte les choix** du Joueur 0

$\to$ depuis $\underbrace{v_0 v_1 \ldots}_{h} \underbrace{v_k}_{\in V_0}$ dicte v_{k+1} avec hv_k (mémoire) ou v_k (sans mémoire)

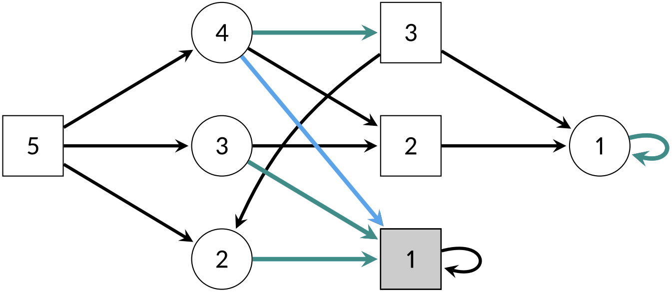


Une partie est **conforme** à σ_0 si $v_{k+1} = \sigma_0(v_0 \ldots v_k)\ \forall k \in \mathbb{N}, \forall v_k \in V_0$

Stratégie

On considère **l'ensemble des parties conformes** à la stratégie σ_0

$$\text{Plays}_{\sigma_0} = \{\boxed{5}\ 4\ \boxed{1}^{\omega}, \boxed{5}\ 3\ \boxed{1}^{\omega}, \boxed{5}\ 2\ \boxed{1}^{\omega}\}$$
$$\text{Plays}_{\sigma'_0} = \{\boxed{5}\ 4\ \boxed{3}\ 1^{\omega}, \boxed{5}\ 4\ \boxed{3}\ 2\ \boxed{1}^{\omega}, \boxed{5}\ 3\ \boxed{1}^{\omega}, \boxed{5}\ 2\ \boxed{1}^{\omega}\}$$

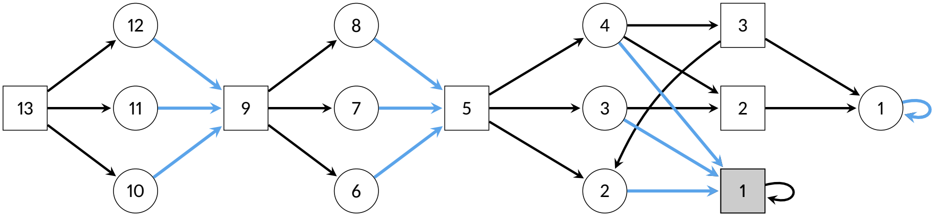


Stratégie σ_0 est **gagnante** pour Joueur 0 si $\forall \rho \in \text{Plays}_{\sigma_0}, \rho \in \Omega_0$

Résumé

On peut créer **l'arène complète** du jeu de Nim

→ Joueur 0 a une **stratégie gagnante** depuis le nœud initial



Récapitulons

- notions d'arène de jeu, de partie, et d'objectif
- notion de stratégie gagnante (pour gagner à coup sûr au conseil)

Sommaire

1. Pareto-Optimalité pour la Vérification et la Synthèse dans les **Jeux sur Graphe**
2. Pareto-Optimalité pour la **Vérification et la Synthèse** dans les Jeux sur Graphe
3. **Pareto-Optimalité** pour la Vérification et la Synthèse dans les Jeux sur Graphe

Systèmes Réactifs

Systèmes informatiques qui **interagissent** avec leur **environnement**

- ils doivent pouvoir **réagir** aux évènements dans cet environnement
- il faut donc considérer **tous les évènements possibles** de celui-ci

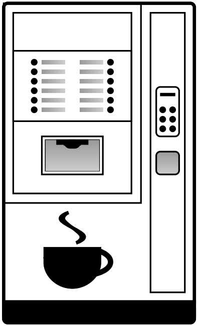
Ces systèmes sont **grands et complexes** et donc **sujets aux erreurs**

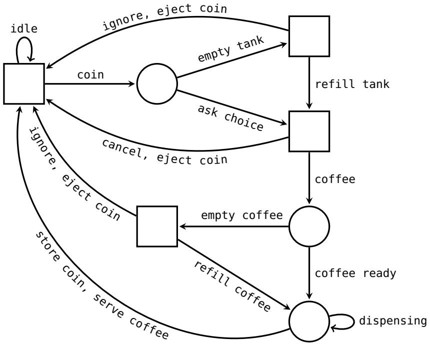
- nécessité de nombreux **tests** pour assurer leur bon fonctionnement
- il est **difficile de tester** toutes les possibilités d'évènements
- certains systèmes sont **critiques**, ne **tolèrent pas les bugs**

Vérification Formelle

On aimerait une **preuve formelle** du bon fonctionnement du système

- montrer que le système a **toujours un comportement correct**
- **quels que soient** les évènements de l'environnement





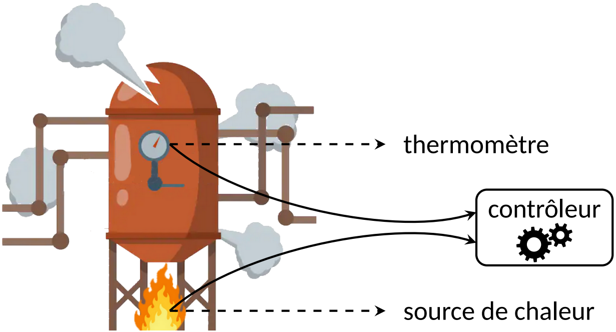
arrête de servir $\longrightarrow$ formule LTL $\phi = \neg(\Diamond\Box\mathtt{dispensing})$

Synthèse de Contrôleur

On aimerait **générer** un contrôleur qui **fait respecter** la spécification

- depuis un **modèle de l'environnement**
- et une **spécification** que notre système **doit respecter**

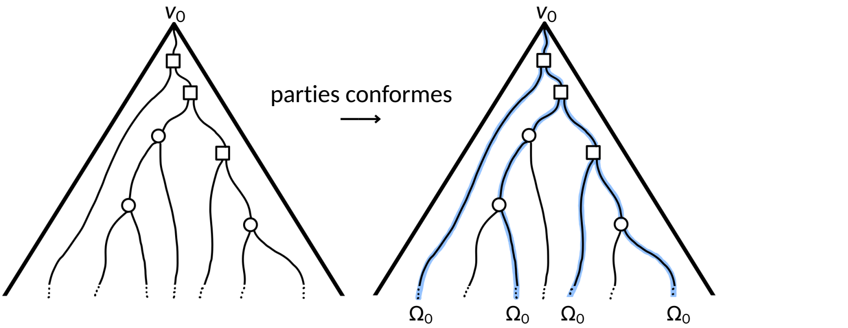
Contrôleur pour **réguler la température** où la spécification est $[T_{min}, T_{max}]$



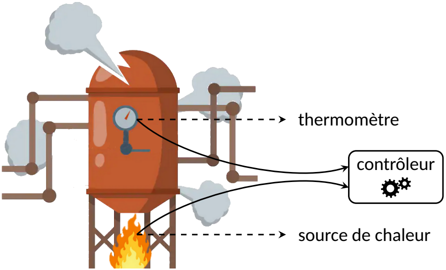
Modèle Classique [GTW02]

Approche pour la **synthèse** et la **vérification**: jeux à **somme nulle** (G, Ω_0)

- **système** = Joueur 0, **environnement** = Joueur 1
- **but du système**: bien fonctionner, i.e., **respecter la spécification**
- **environnent antagoniste**: son objectif est **opposé** $\Omega_1 = \neg\Omega_0$
- **stratégie gagnante** pour Joueur 0 = **contrôleur** pour système

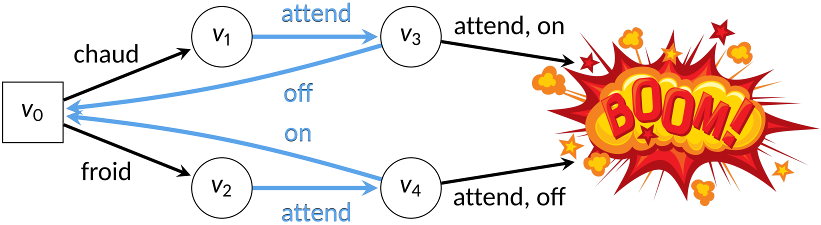


Retour à la Synthèse de Contrôleur



Approche

- modéliser le problème **par un jeu**
- trouver une **stratégie gagnante**
- on obtient ainsi un **contrôleur**

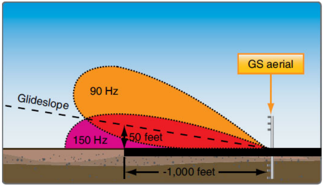


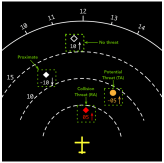
Limitations et Alternatives

Environnement complètement antagoniste: **abstraction de la réalité**

- suppose que le **seul but** de l'environnement = **faire échouer le système**
- l'environnent peut être composé de **plusieurs** composants
- chacun avec son **propre but**, autre que faire échouer le système

Alternative: jeux à **somme non-nulle**



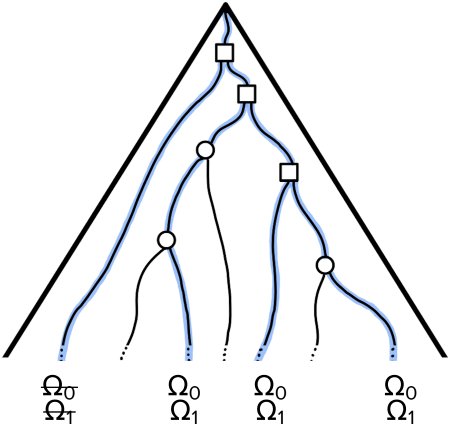


Rationalité de l'Environnement

Seul but de l'environnement **faire échouer le système**

⟶ environnement suit **n'importe quelle** partie consistante: **stratégie gagnante**

Supposons maintenant que l'environnement a son **propre objectif Ω_1**



Sommaire

Contributions

Introduire un **nouveau modèle** de jeux

- représente un environnement avec **plusieurs composants**
- une **notion appropriée de rationalité** pour l'environnement

Étudier le problème de **synthèse** dans ce contexte

$\longrightarrow$ synthèse où Joueur 0 satisfait Ω_0 contre un **environnement rationnel**

Étudier le problème de **vérification** dans ce contexte

$\longrightarrow$ vérification contre un **environnement rationnel**

$\longrightarrow$ vérification de **plusieurs comportements** pour le système

Notre Nouveau Modèle

Jeux de Stackelberg-Pareto (jeu SP): $\mathcal{G} = (G, \Omega_0, \Omega_1, \ldots, \Omega_t)$

- Joueur 0 (système): objectif Ω_0
- Joueur 1 (environnement): **plusieurs objectifs** $\Omega_1, \ldots, \Omega_t$
- **somme non-nulle**: environnement avec plusieurs composants

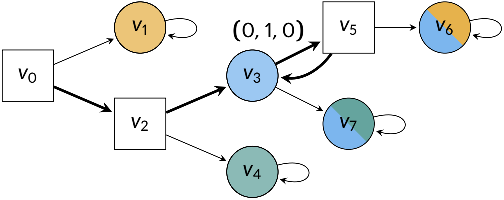
Payoff de ρ pour Joueur 1 est le **vecteur de Booléens** $\text{pay}(\rho) \in \{0, 1\}^t$

- **ordre** $\leq$ sur les payoffs, e.g., $(0, 1, 0) < (0, 1, 1)$

$\Omega_1 = \text{Reach}(\{v_4, v_7\})$

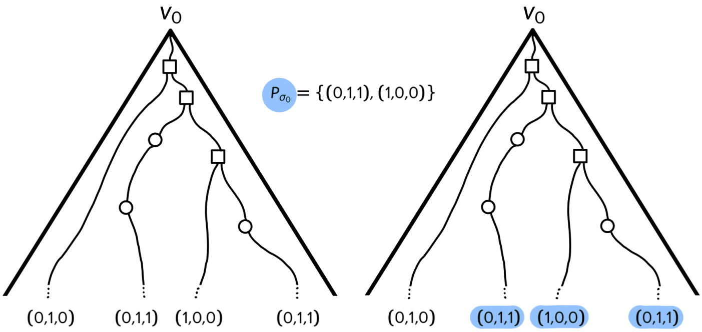
$\Omega_2 = \text{Reach}(\{v_3, v_6, v_7\})$

$\Omega_3 = \text{Reach}(\{v_1, v_6\})$



Payoffs Pareto-Optimaux

1. Joueur 0 **annonce** sa stratégie σ_0
2. Joueur 1 **considère** Plays_{σ_0}
   - **ensemble de payoffs** correspondant $\{\text{pay}(\rho) \mid \rho \in \text{Plays}_{\sigma_0}\}$
   - identifie payoffs **Pareto-optimaux** (PO) (maximaux pour $\leq$) : P_{σ_0}

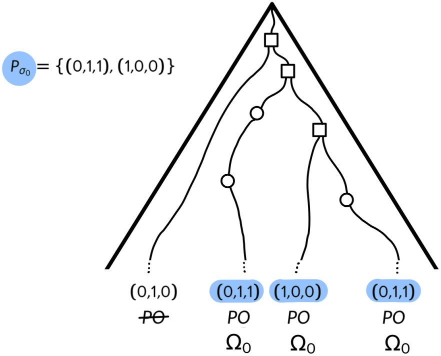


Problème de Synthèse Stackelberg-Pareto (problème SSP)

Le problème SSP consiste à décider si il existe une stratégie σ_0 pour le Joueur 0 t.q. pour toute partie $\rho \in \mathsf{Plays}_{\sigma_0}$ avec $\mathsf{pay}(\rho) \in P_{\sigma_0}$, on a $\rho \in \Omega_0$

Environnement **rationel** répond à σ_0 **pour obtenir un payoff Pareto-optimal**

⟶ Joueur 0 doit **satisfaire** Ω_0 dans toute **réponse rationnelle**



Nos Résultats sur le Problème SSP (1/3)

Etudie la **complexité du problème** pour une série d'objectifs

Complexité du Problème SSP

Objectif	Classe de complexité
Reach (arbres), Büchi	NP-complet
co-Büchi	NEXPTIME, NP-difficile
Reach, Safe, BooleanBüchi, Parity	NEXPTIME-complet
Muller, Streett, Rabin	NEXPTIME-complet
LTL	2EXPTIME-complet

Nos Résultats sur le Problème SSP (2/3)

Étudié la **complexité paramétrique** du problème

Complexité FPT du Problème SSP

Résoudre le problème SSP est FPT

Un problème est **fixed-parameter tractable** (FPT) pour un paramètre k si

- il existe une solution qui tourne en $f(k) \times n^{\mathcal{O}(1)}$
- où f est une fonction de k indépendante de n

Intuition: résoudre est polynomial en la taille de l'entrée **exponentiel en** k

$\longrightarrow$ résoudre est fixed-parameter tractable (facile si on fixe un petit k)

Nos Résultats sur le Problème SSP (3/3)

Complexité FPT du Problème SSP

Objectif	Complexité en les paramètres
Reach	double exp en t
Safe	double exp en t
Büchi	double exp en t
co-Büchi	double exp en t
BooleanBüchi	double exp en t, exp en $\sum_{i=0}^{t} m_i$, et poly en $\sum_{i=0}^{t} \lvert\phi_i\rvert$
Parity	double exp en t et exp en $\sum_{i=0}^{t} d_i$
Muller	double exp en t, exp en $\sum_{i=0}^{t} d_i$, et poly en $\sum_{i=0}^{t} \lvert Q_i\rvert$
Streett	double exp en t et exp en $\sum_{i=0}^{t} m_i$
Rabin	double exp en t et exp en $\sum_{i=0}^{t} m_i$

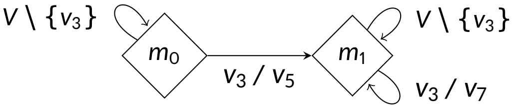
En pratique on peut supposer que ces paramètres ont **de petites valeurs**

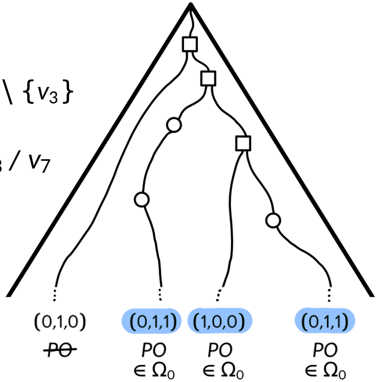
Problème de Vérification Pareto-Rationnelle (problème PRV)

Etant donné une stratégie à mémoire finie σ_0 encodée par $\mathcal{M}$, vérifier si toute partie $\rho \in \mathrm{Plays}_{\sigma_0}$ avec $\mathrm{pay}(\rho) \in P_{\sigma_0}$ est t.q. $\rho \in \Omega_0$

Environnement **rationnel** répond à σ_0 **pour obtenir un payoff Pareto-optimal**

$\longrightarrow$ Vérifier que σ_0 **satisfait** Ω_0 dans toute **réponse rationnelle**

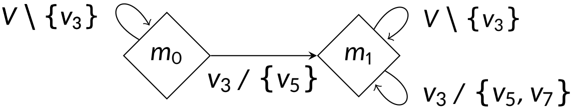


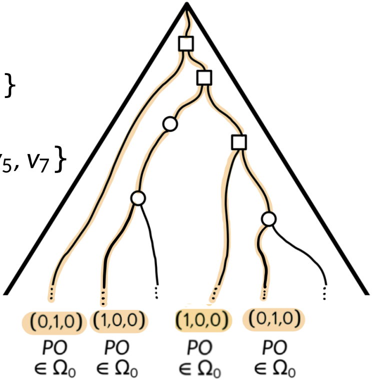


Vérification Pareto-Rationnelle Universelle (problème UPRV)

Etant donné une machine de Moore non-déterministe $\mathcal{M}$, vérifier si pour toute stratégie $\sigma_0 \in [\![\mathcal{M}]\!]$, toute partie $\rho \in \mathsf{Plays}_{\sigma_0}$ avec $\mathsf{pay}(\rho) \in P_{\sigma_0}$ est t.q. $\rho \in \Omega_0$

Généralisation du problème précédent à **plusieurs stratégies**





Nos Résultats sur le Problème PRV et UPRV (1/2)

Étudié les deux problèmes **parity**, **Boolean Büchi**, **LTL** (+ résultats partiels)

Problème PRV

Objectif	Classe de complexité
Parity	co-NP-complet
Boolean Büchi	Π_2P-complet
LTL	PSPACE-complet

Problème UPRV

Objective	Classe de complexité
Parity	PSPACE, NP-difficile, co-NP-difficile
Boolean Büchi	PSPACE-complet
LTL	2EXPTIME-complet

Nos Résultats sur le Problème PRV et UPRV (2/2)

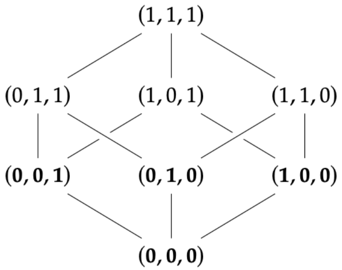
Problème PRV et UPRV

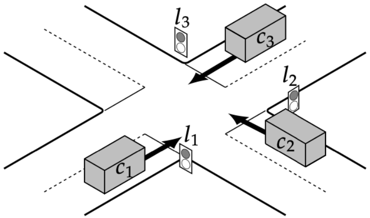
Les deux problèmes sont fixed-parameter tractable (FPT)

Algorithme pour PRV **construit une antichaine** depuis le **treillis de payoffs**

Algorithme supplémentaire: basé sur les **contre-exemples**

→ **implémentés et comparés** sur un exemple et des instances aléatoires





Le Problème SSP est NP-Difficile sur les Arbres

Cadre simple des **arbres**

Utilise le **problème de Couverture par Ensemble** (NP-complet) [Kar72]

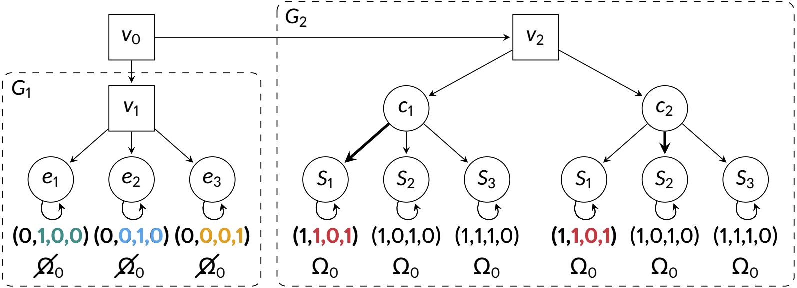
- $C = \{e_1, e_2, \ldots, e_n\}$ avec n **éléments**
- m **sous-ensembles** $S_1, S_2, \ldots, S_m$ t.q. $S_i \subseteq C$
- un **entier k** $\leq m$
- **trouver** k **indices** $i_1, i_2, \ldots, i_k$ t.q. $C = \bigcup_{j=1}^{k} S_{i_j}$.

Construire un **jeu SP** tel que:

solution au problème de couverture $\Longleftrightarrow$ Joueur 0 a **solution** au problème SSP

Réduction au Problème de Couverture par Ensemble

$C = \{e_1, e_2, e_3\}$, $S_1 = \{e_1, e_3\}$, $S_2 = \{e_2\}$, $S_3 = \{e_1, e_2\}$, $k = 2$



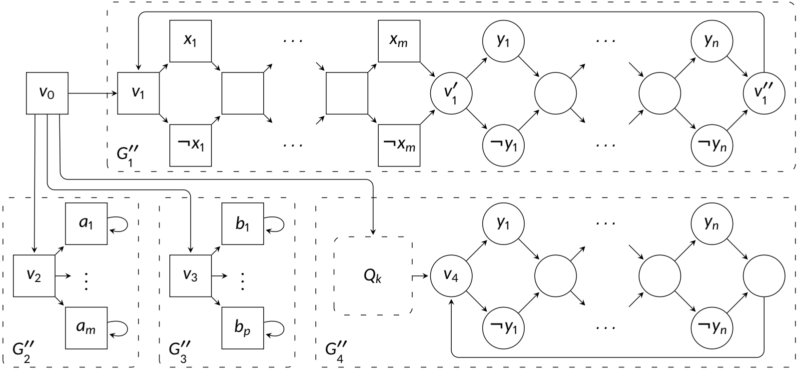
Toute partie de G_1 est **conforme à toute stratégie** du Joueur 0 et $\notin \Omega_0$

$\longrightarrow$ dans une solution, payoffs de G_1 **ne peuvent pas être Pareto-optimaux**

Chaque payoff de G_1 doit être $<$ **qu'un payoff de G_2** (un ensemble)

NEXPTIME-Difficile

Construction compliquée, plus complexe pour parité que pour accessibilité



Merci pour votre attention !

Bibliography I

[FKL10] Dana Fisman, Orna Kupferman, and Yoad Lustig.

Rational synthesis.

In Javier Esparza and Rupak Majumdar, editors, *Tools and Algorithms for the Construction and Analysis of Systems, 16th International Conference, TACAS 2010, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2010, Paphos, Cyprus, March 20-28, 2010. Proceedings*, volume 6015 of *Lecture Notes in Computer Science*, pages 190–204. Springer, 2010.

Bibliography II

[GTW02] Erich Grädel, Wolfgang Thomas, and Thomas Wilke, editors.

Automata, Logics, and Infinite Games: A Guide to Current Research [outcome of a Dagstuhl seminar, February 2001], volume 2500 of *Lecture Notes in Computer Science*. Springer, 2002.

[Kar72] Richard M. Karp.

Reducibility among combinatorial problems.

In Raymond E. Miller and James W. Thatcher, editors, *Proceedings of a symposium on the Complexity of Computer Computations, held March 20-22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, USA*, The IBM Research Symposia Series, pages 85–103. Plenum Press, New York, 1972.

Bibliography III

[KPV16] Orna Kupferman, Giuseppe Perelli, and Moshe Y. Vardi.
Synthesis with rational environments.
Ann. Math. Artif. Intell., 78(1):3–20, 2016.

[Pap94] Christos H. Papadimitriou.
Computational complexity.
Addison-Wesley, 1994.

[vS37] Heinrich Freiherr von Stackelberg.
Marktform und Gleichgewicht.
Wien und Berlin, J. Springer, 1937.