IF122 24/3/21

Sébastien Delpeuch · delpeuch.net · CC BY-NC-SA 4.0

Méthode de résolution de jeux de type $\mathrm{Reach}(F_1) \wedge \mathrm{Reach}(F_2)$, ou $\mathrm{Reach}(F_1) \wedge \mathrm{Avoid}(F_2)$, etc.

→ <u>Jeux d'obligation</u>

Win = combinaison booléenne de cond. $\mathrm{Reach}(F)$, $F \subseteq V$

$= \tilde{F} \subseteq 2^V$

$\tilde{F} \hookrightarrow \{U : U \subseteq V\}$

$= \{U_1, U_2, \dots U_n\}$

Ex $\mathrm{Win} = \mathrm{Reach}(F_1) \wedge \mathrm{Reach}(F_2)$

→ $\tilde{F}_1 = \{ U \subseteq V : U \cap F_1 \neq \emptyset$, et $U \cap F_2 \neq \emptyset \}$

$\mathrm{Win}_2 = \mathrm{Reach}(F_1) \wedge \mathrm{Avoid}(F_2)$

→ $\tilde{F}_2 = \{ U \subseteq V : U \cap F_1 \neq \emptyset, U \cap F_2 = \emptyset \}$

Sébastien Delpeuch · delpeuch.net · CC BY-NC-SA 4.0

Partie $\pi = v_0, v_1, \ldots$ est gagnée par P_0 ssi l'ensemble des états visités dans π app. à F.

Pour résoudre ces jeux, on va passer par les jeux de parité faible.

$A = (V_0, V_1, E)$ + fonction de priorité $p : V_0 \cup V_1 \to \{0, \ldots, K\}$

$p(v)$ = priorité (couleur) de v

Partie $\pi = v_0, v_1, \ldots$ maximale

$p(v_0)\ p(v_1) \ldots$

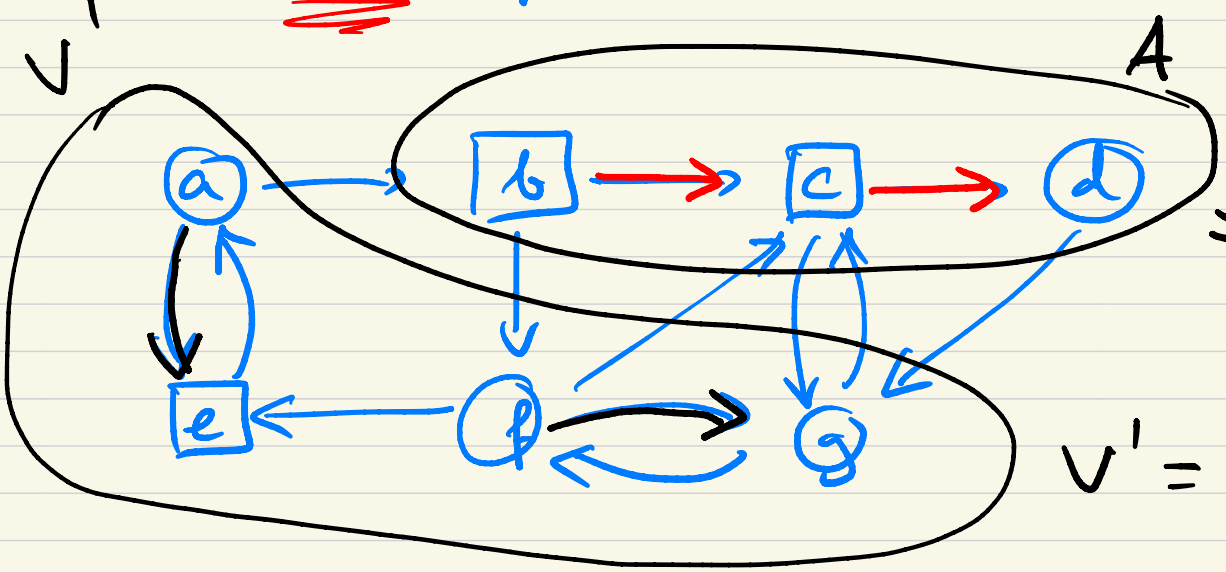
P_0 gagne π si la plus grande priorité vue dans π est paire

(P_1 gagne si elle est impaire).

Sébastien Delpeuch · delpeuch.net · CC BY-NC-SA 4.0

π est gagnée par P_0 si

$\max\{p(v_i) : i \geq 0\}$ est pair



$U = \{d\}$

$= Attr_1(U)$

$= \{d, c, b\}$

$V' = W_0'$

$p(a) = p(e) = 0$ $\quad p(c) = p(g) = 2$

$p(b) = p(f) = 1$ $\quad p(d) = 3$

$d \overset{?}{\in} W_1$ $\quad c \in W_1$ $\quad c \to d \in V_1$

$g \rightleftarrows f$

f g f g ... $\to$ max 2 P_0 gagne

1 2 1 2

[Plus tard : jeu de parité $\to$ do-saurent]

Sébastien Delpeuch · delpeuch.net · CC BY-NC-SA 4.0

On veut montrer que les jeux de parité faible sont :

- déterminés ($V = W_0 \dot\cup W_1$)
- résolubles en temps poly (calcul de W_0, W_1 en PTime)
- les stratégies gagnantes sont positionnelles

$$\sigma_0 : V_0 \to V$$
$$\sigma_1 : V_1 \to V$$

$A = (V_0, V_1, E)$, $p : V \to \{0..,K\}$

Supposons que K est impair

(l'autre cas est symétrique)

$U := \{ v \in V : p(v) = K \} \neq \emptyset$

On note que $U \subseteq W_1$, et aussi $A = Attr_1(U) \subseteq W_1$.

Sébastien Delpeuch · delpeuch.net · CC BY-NC-SA 4.0

On considère $V \setminus A =: V'$

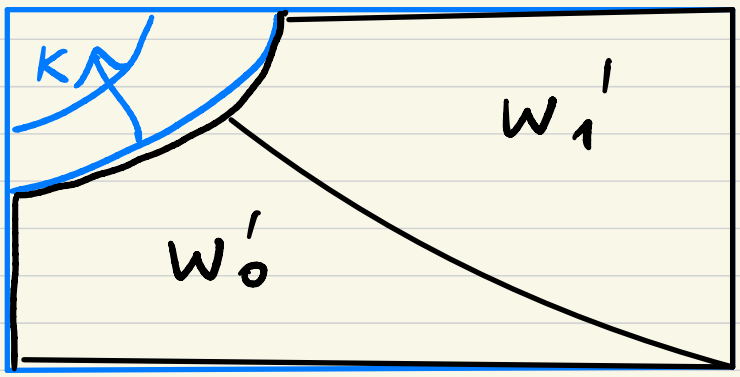
On résout le jeu de parité faible sur la sous-arène de jeu définie par V'

$$\mathcal{A}' = (V_0 \setminus A,\ V_1 \setminus A,\ E \cap (V' \times V'))$$

Par récurrence, on obtient des régions gagnantes $W_0' \subseteq V'$, $W_1' \subseteq V'$,

ainsi que des stratégies gagnantes positionnelles $\sigma_0' : V_0' \to V'$, $\sigma_1' : V_1' \to V'$

et $W_0' \cup W_1' = V'$.

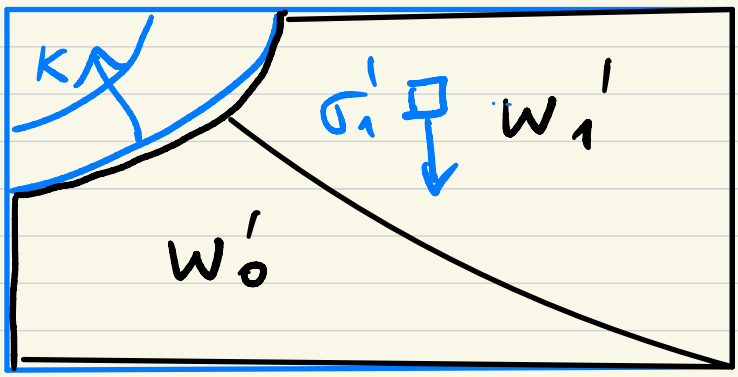


On montre :

$W_0 = W_0'$

$W_1 = A \cup W_1'$

Sébastien Delpeuch · delpeuch.net · CC BY-NC-SA 4.0



A

On montre :

$W_0 = W_0'$

$W_1 = A \cup W_1'$

$W_0' \subseteq W_0$ (Q1 du DM)

parce que P_1 ne peut pas quitter V'.

$A \subseteq W_1$:

Dans A, P_1 applique la stratégie d'attracteur pour aller dans l'ensemble de priorité K, donc K sera la plus grande priorité.

$W_1' \subseteq W_1$

Dans W_1', P_1 applique la stratégie (gagnante sur V') σ_1'.

Si une partie commence dans W_1' et reste dans V', alors elle est gagnée par P_1 (par réc.)

Sébastien Delpeuch · delpeuch.net · CC BY-NC-SA 4.0

Si une partie démarre et reste dans V', alors :

$\Rightarrow$ elle est gagnée par P_0, si elle démarre dans W_0'

resp.

par P_1, si elle démarre dans W_1' (par. hypp.)

Une partie peut quitter V' (à cause de P_0) et aller dans A, dans quel cas elle va être par P_1.

Les stratégies gagnantes :

$\sigma_0 = \sigma_0'$ sur W_0'

$\sigma_1 = \sigma_1'$ sur W_1', attracteur sur A

sont positionnelles.

Sébastien Delpeuch · delpeuch.net · CC BY-NC-SA 4.0

Le temps de calcul sera quadratique :

Calcul de A : $O(|V| + |E|)$

Appel récursif sur V' : $O(|V'| + |E'|)^2$

$$t(V, E) \leq c \cdot (|V| + |E|) + c' \cdot (|V'| + |E'|)^2$$

$$\leq c' (|V| + E|)^2$$

Jeux d'obligation :

$$\mathcal{F} \subseteq 2^V$$

$$\mathcal{F} = \{ U_1, U_2, \ldots U_n \}$$

Il nous reste à réduire les jeux d'obligation à partir faible.

Sébastien Delpeuch · delpeuch.net · CC BY-NC-SA 4.0

On va "mémoriser" les états visités.

Nouvelle arène $V \times 2^V = V'$

2^V ↑ mémoire

avec arêtes $E' \subseteq V' \times V'$:

$(u, X) \to (v, Y)$ si

$(u, v) \in E$ et $Y = X \cup \{u\}$

$v_0 \to v_1 \to v_2 \to \cdots$ partie infinie

$(v_0, \emptyset) \to (v_1, \{v_0\}) \to (v_2, \{v_0, v_1\}) \to \cdots$

partie dans l'arène A'

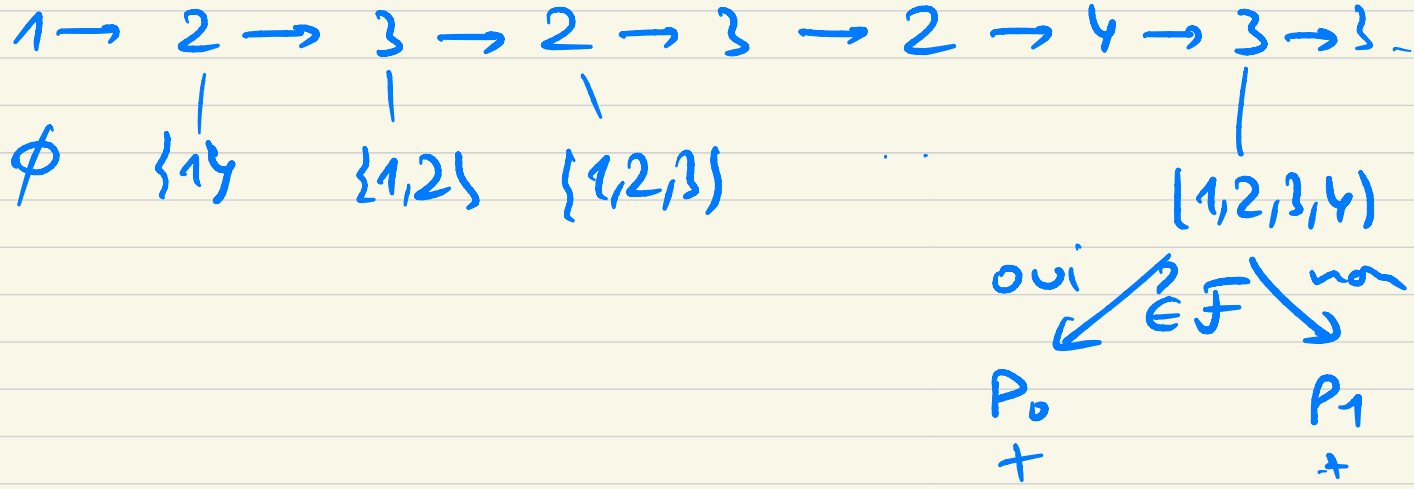
$V_0' = V_0 \times 2^V$

$V_1' = V_1 \times 2^V$

$(v_0, X_0) \to (v_1, X_1) \to \cdots$

$X_0 \subseteq X_1 \subseteq X_2 \cdots$ $\quad X_n = X_{n+1} = \cdots$

Sébastien Delpeuch · delpeuch.net · CC BY-NC-SA 4.0



On déf. $p : V' \to \{0, \dots, 2|V|+2\}$

$$p((u,X)) = \begin{cases} 2|X|+2 & \text{si } X \in F \\ 2|X|+1 & \text{si } X \notin F \end{cases}$$

$|X| \leq |V|$

Rq les priorités au cours d'une partie sont croissantes, parce que

$(u_0, X_0) \to (u_1, X_1) \to \cdots$

$X_0 \subseteq X_1 \subseteq \cdots$

$2|X_0|+1/2 \leq 2|X_1|+1/2 \leq \cdots$

La plus grande priorité est $2|X_n| + 1/2$, où X_n = ens. des sommets visités.

Sébastien Delpeuch · delpeuch.net · CC BY-NC-SA 4.0

Réduction du jeu d'obligation (V_0, V_1, E, F) vers le jeu de parité faible (V_0', V_1', E', p) :

$$f(v) = (v, \phi) \in V'$$
$$v \in V$$

v est gagnant pour P_0 dans le jeu d'obligation $\Leftrightarrow$

(v, ϕ) est gagnant pour P_0 dans le jeu de parité faible.

($\Rightarrow$) P_0 a une stratégie gagnante σ_0

$\Rightarrow$ toute partie conforme à σ_0 :

$$v_0 \to v_1 \to \cdots \quad \text{tq,}$$
$$\{v_i : i \geq 0\} \in F$$

Dans V', P_0 "joue" σ_0 :

Sébastien Delpeuch · delpeuch.net · CC BY-NC-SA 4.0

(v, X) , $v \in V_0$

$\sigma_0'(v, X) := (\sigma_0(v), X \cup \{v\})$

σ_0' est gagnante pour P_0 , parce que la priorité max. est $2|X|+2$, où X = ens. des sommets visités ($X \in F$)

($\Leftarrow$) soit σ_0' = strat. gagn. de P_0 dans le jeu de parité faible

$\sigma_0' : V_0' \to V$ strat. pos.

$\rightsquigarrow$ stratégie σ_0 pour P_0 avec mémoire 2^V dans le jeu d'objectif

$\sigma_0 : V_0 \times M^{2^V} \to V$

$\sigma_0(v, X) := w$ tq. $\sigma_0'(v, X) = (w, Y)$

update $(v, X) := X \cup \{v\}$

Sébastien Delpeuch · delpeuch.net · CC BY-NC-SA 4.0

$\| \; \mathrm{Reach}(F_1) \cap \mathrm{Reach}(F_2)$

$(V_0; V_1, E) \longrightarrow (V_0', V_1', E')$

$V' = V \times 2^V$

Ici : plus simple

ni F_1, F_2 — F_1 pas F_2 — F_1 et F_2

$\phi, \{1\}, \{2\}, \{1,2\}$

$(u, X) \to (v, X')$ si :

$u \to v$ et

$$X' = \begin{cases} X & \text{si } u \notin F_1, u \notin F_2 \\ X \cup \{1\} & \text{si } u \in F_1 \setminus F_2 \\ X \cup \{2\} & F_2 \setminus F_1 \\ X \cup \{1,2\} & u \in F_1 \cap F_2 \end{cases}$$

Réduction vers jeu d'accessibilité

$\mathrm{Reach}(\{(v, \{1,2\}) : v \in V\})$

Sébastien Delpeuch · delpeuch.net · CC BY-NC-SA 4.0