Jeux Distribués Asynchrones

P. Gastin, B. Lerman, M. Zeitoun

LIAFA, CNRS & Univ. Paris 7

Plan

- ⋆ Motivation et Présentation des jeux distribués asynchrones.
- ⋆ Comparaison avec les jeux distribués de [MW03].
- ⋆ Décidabilité du problème d'accessibilité sur les alphabets série-parallèles.

Motivations et Présentation

Motivations

- Les jeux sont des outils intéressants pour la synthèse de contrôleurs
  - On représente l'environnement par un joueur
  - On représente le système distribué par une équipe
  - Le bon comportement est la condition de gain
  - Résoudre le jeu revient à trouver un contrôleur

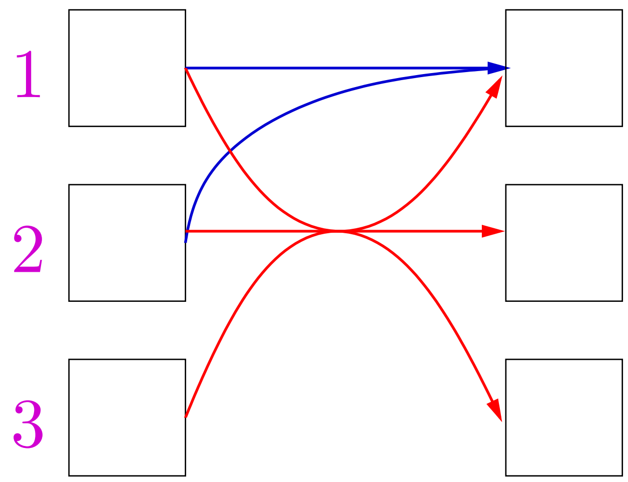
- Les jeux séquentiels sont peu adéquats pour les systèmes distribués
  - Une stratégie gagnante séquentielle ne donne pas facilement de contrôleur distribué

Architecture distribuée (Zielonka)

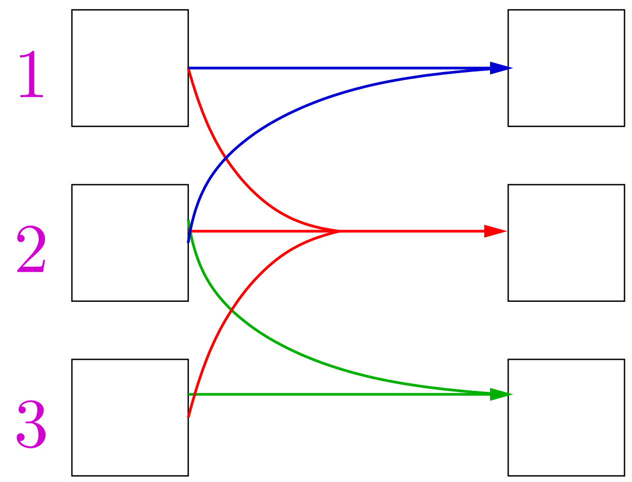
$(\Sigma, \mathcal{P}, R, W)$:

- Σ : ensemble fini de joueurs
- $\mathcal{P}$: ensemble fini de places mémoires
- $R : \Sigma \rightarrow 2^{\mathcal{P}}$ assigne à chaque joueur de $a \in \Sigma$ son domaine de lecture $R(a)$
- $W : \Sigma \rightarrow 2^{\mathcal{P}}$ assigne à chaque joueur de $a \in \Sigma$ son domaine d'écriture $W(a)$

$\Sigma = \{a \;\; b\}$



$\Sigma = \{a \;\; b \;\; c\}$ (cellulaire)



Jeux distribués asynchrones sur des traces

- (Σ, D) : alphabet de dépendance.
- $\Sigma = \Sigma_0 \uplus \Sigma_1$: Partition de l'ensemble des joueurs en deux équipes 0 et 1.

  Les joueurs de l'équipe 0 coopèrent contre l'équipe 1.

Jeux distribués asynchrones sur des traces

- (Σ, D) : alphabet de dépendance.
- $\Sigma = \Sigma_0 \uplus \Sigma_1$: Partition de l'ensemble des joueurs en deux équipes 0 et 1.

  Les joueurs de l'équipe 0 coopèrent contre l'équipe 1.

$\perp$

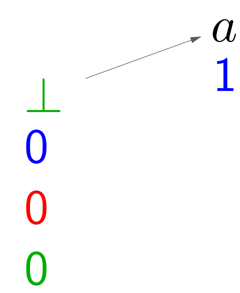
0

0

0

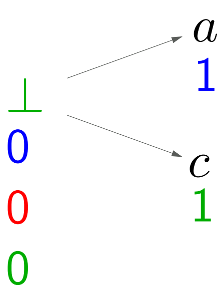
Jeux distribués asynchrones sur des traces

- (Σ, D) : alphabet de dépendance.
- $\Sigma = \Sigma_0 \uplus \Sigma_1$: Partition de l'ensemble des joueurs en deux équipes 0 et 1.
  Les joueurs de l'équipe 0 coopèrent contre l'équipe 1.



Jeux distribués asynchrones sur des traces

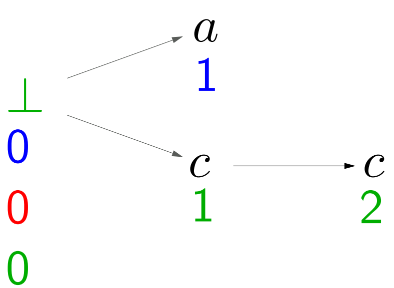
- (Σ, D) : alphabet de dépendance.
- $\Sigma = \Sigma_0 \uplus \Sigma_1$: Partition de l'ensemble des joueurs en deux équipes 0 et 1.
  Les joueurs de l'équipe 0 coopèrent contre l'équipe 1.



Jeux distribués asynchrones sur des traces

- (Σ, D) : alphabet de dépendance.
- $\Sigma = \Sigma_0 \uplus \Sigma_1$: Partition de l'ensemble des joueurs en deux équipes 0 et 1.

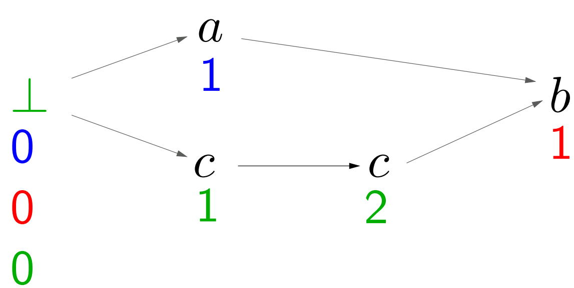
  Les joueurs de l'équipe 0 coopèrent contre l'équipe 1.



Jeux distribués asynchrones sur des traces

- (Σ, D) : alphabet de dépendance.
- $\Sigma = \Sigma_0 \uplus \Sigma_1$: Partition de l'ensemble des joueurs en deux équipes 0 et 1.

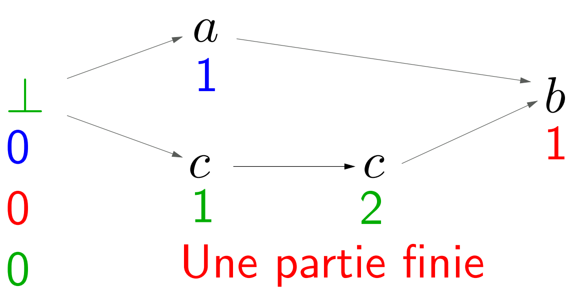
  Les joueurs de l'équipe 0 coopèrent contre l'équipe 1.



Jeux distribués asynchrones sur des traces

- (Σ, D) : alphabet de dépendance.
- $\Sigma = \Sigma_0 \uplus \Sigma_1$: Partition de l'ensemble des joueurs en deux équipes 0 et 1.

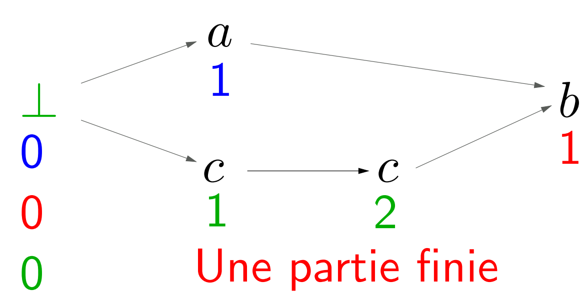
  Les joueurs de l'équipe 0 coopèrent contre l'équipe 1.



Jeux distribués asynchrones sur des traces

- (Σ, D) : alphabet de dépendance.
- $\Sigma = \Sigma_0 \uplus \Sigma_1$: Partition de l'ensemble des joueurs en deux équipes 0 et 1.

  Les joueurs de l'équipe 0 coopèrent contre l'équipe 1.



Une partie finie

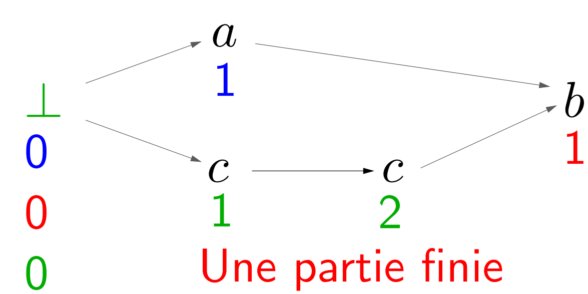
- Partie : une trace finie ou infinie de $\mathbb{R}(\Sigma', D)$.

$$\Sigma' = \{(a, p) \mid a \in \Sigma, p \in Q_{W(a)}\} \cup \{(\perp, q^0)\}, \qquad (a, p)\,\mathrm{D}\,(b, q) \Leftrightarrow a\,\mathrm{D}\,b$$

- Les règles du jeu pour le joueur a sont données par $T_a \subseteq Q_{R(a)} \times Q_{W(a)}$.

Jeux distribués asynchrones sur des traces

- (Σ, D) : alphabet de dépendance.
- $\Sigma = \Sigma_0 \uplus \Sigma_1$: Partition de l'ensemble des joueurs en deux équipes 0 et 1. Les joueurs de l'équipe 0 coopèrent contre l'équipe 1.
- Partie : une trace finie ou infinie de $\mathbb{R}(\Sigma', D)$.



Une partie finie

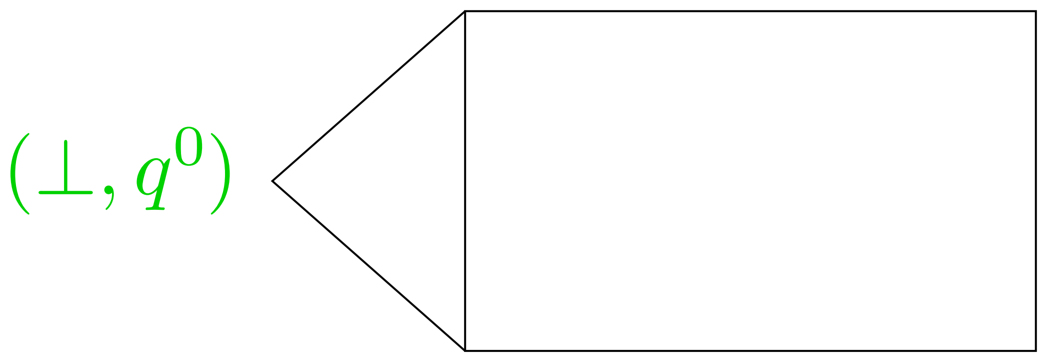
$$\Sigma' = \{(a,p) \mid a \in \Sigma, p \in Q_{W(a)}\} \cup \{(\perp, q^0)\}, \qquad (a,p)\,\mathrm{D}\,(b,q) \Leftrightarrow a\,\mathrm{D}\,b$$

- Les règles du jeu pour le joueur a sont données par $T_a \subseteq Q_{R(a)} \times Q_{W(a)}$.
- Condition de gain : un ensemble de traces finies et/ou infinies $\mathcal{W} \subseteq \mathbb{R}(\Sigma', \mathrm{D})$. L'équipe 0 gagne sur les parties de $\mathcal{W}$ et perd sur les parties de $\mathbb{R}(\Sigma', D) \setminus \mathcal{W}$.
- Remarque Il existe des positions sur lesquelles les deux équipes peuvent jouer.

On retrouve l'état global à partir d'une partie distribuée.

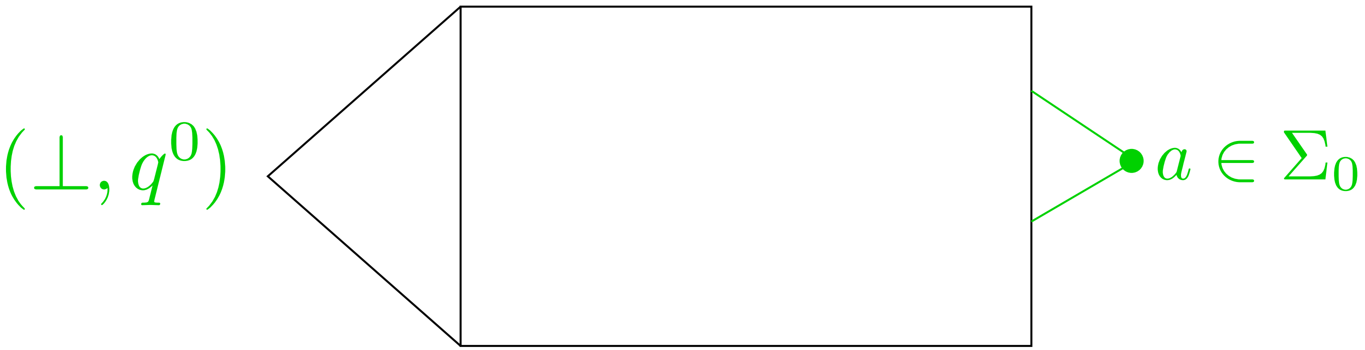
Stratégies distribuées de Σ_0 avec mémoire parfaite

Une partie distribuée est une trace enracinée de $\mathbb{R}(\Sigma', \mathrm{D})$.



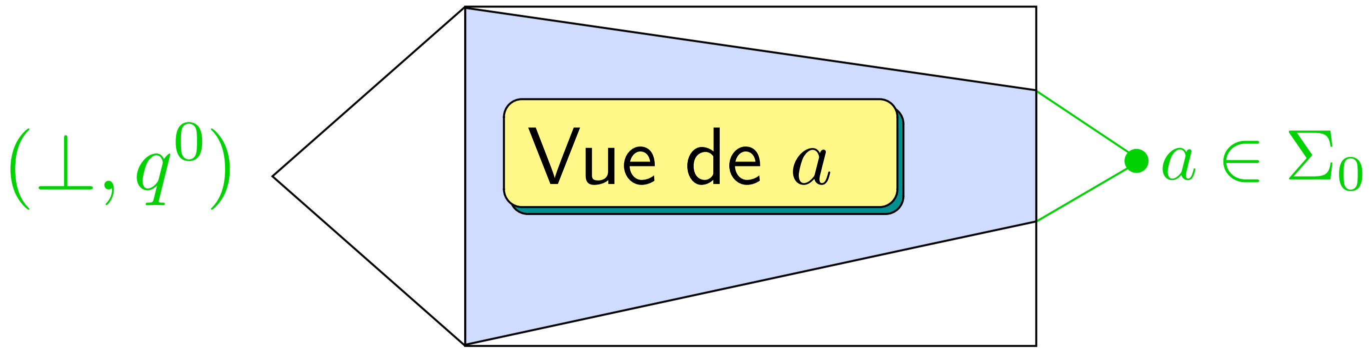
Stratégies distribuées de Σ_0 avec mémoire parfaite

Une partie distribuée est une trace enracinée de $\mathbb{R}(\Sigma', \mathrm{D})$.



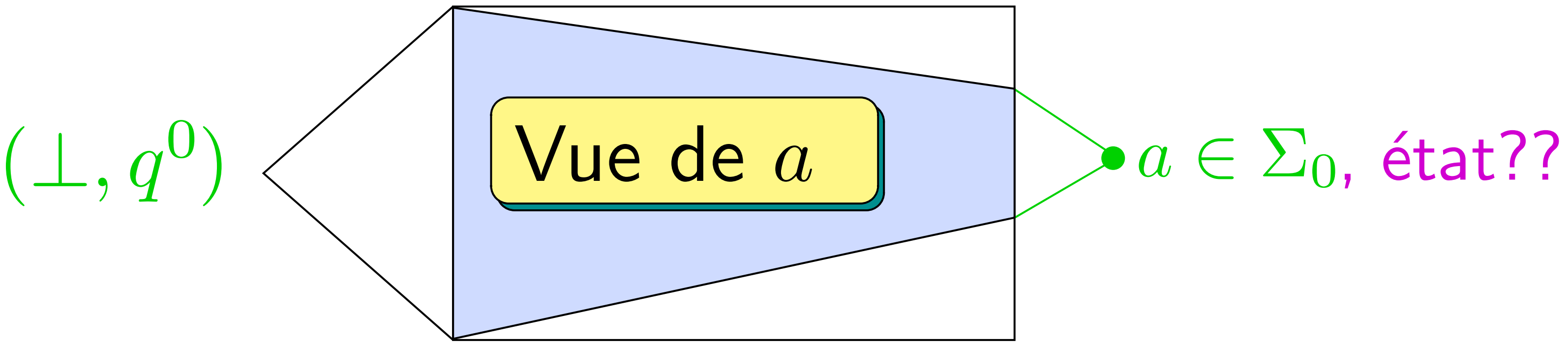
Stratégies distribuées de Σ_0 avec mémoire parfaite

Une partie distribuée est une trace enracinée de $\mathbb{R}(\Sigma', \mathrm{D})$.



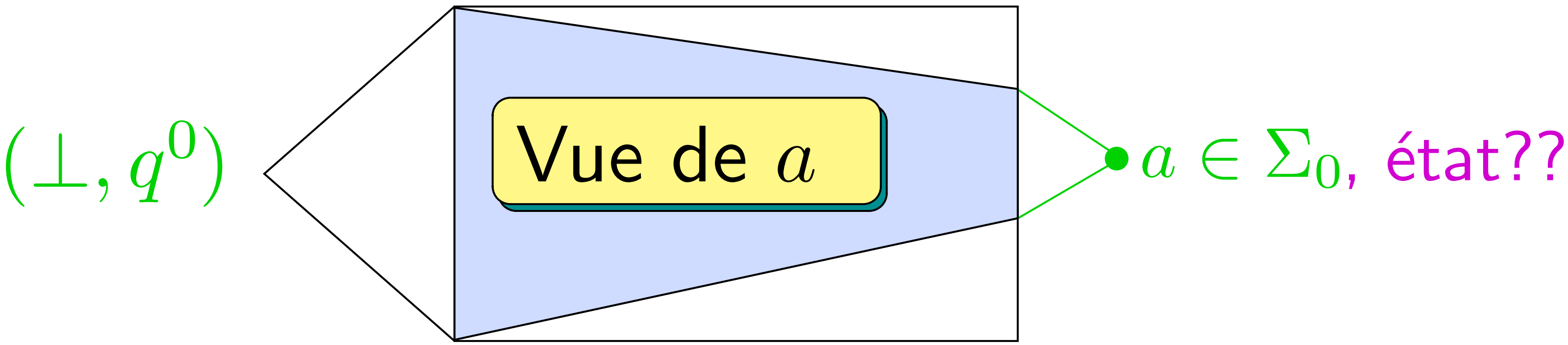
Stratégies distribuées de Σ_0 avec mémoire parfaite

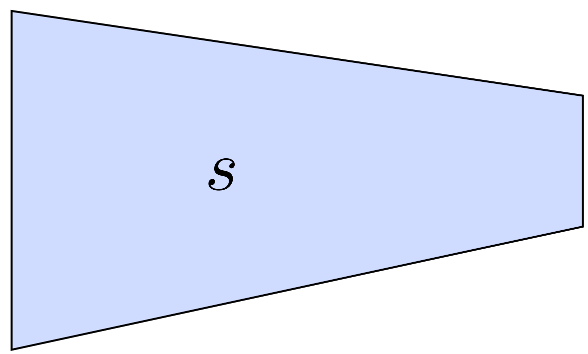
Une partie distribuée est une trace enracinée de $\mathbb{R}(\Sigma', \mathrm{D})$.



Stratégies distribuées de Σ_0 avec mémoire parfaite

Une partie distribuée est une trace enracinée de $\mathbb{R}(\Sigma', \mathrm{D})$.



f : 

, $a \longmapsto Q_{W(a)} \cup \{\text{STOP}\}$

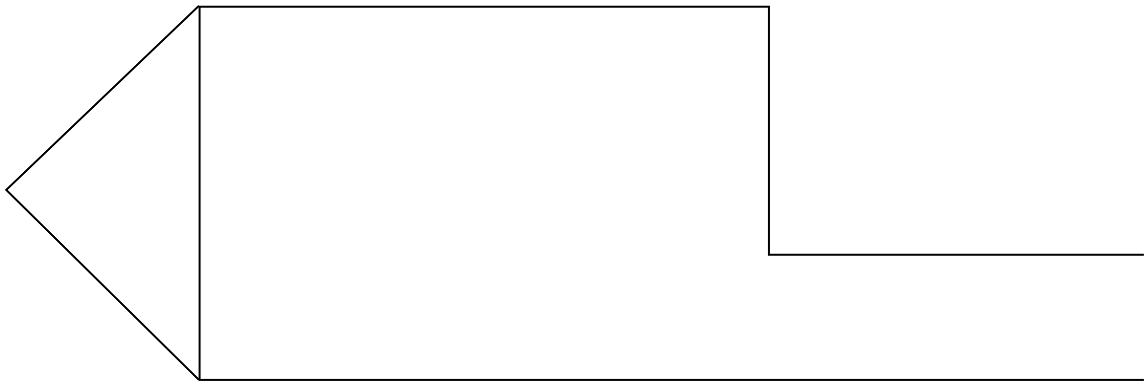
- La stratégie de a n'utilise pas toute la partie, uniquement la vue de a.
- La stratégie peut permettre à a de s'arrêter.
- Toute stratégie distribuée est une abstraction d'une de ces stratégies.

Stratégies distribuées : condition de maximalité

Soit f une stratégie distribuée.

Une partie distribuée est f-maximale si aucun Σ_0-coup compatible avec f ne peut être joué.

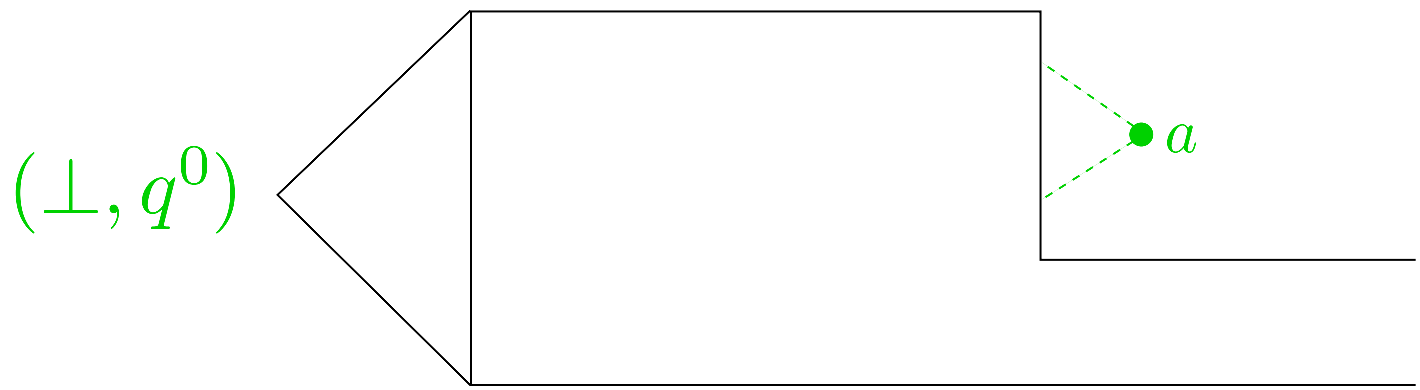
$(\perp, q^0)$



Stratégies distribuées : condition de maximalité

Soit f une stratégie distribuée.

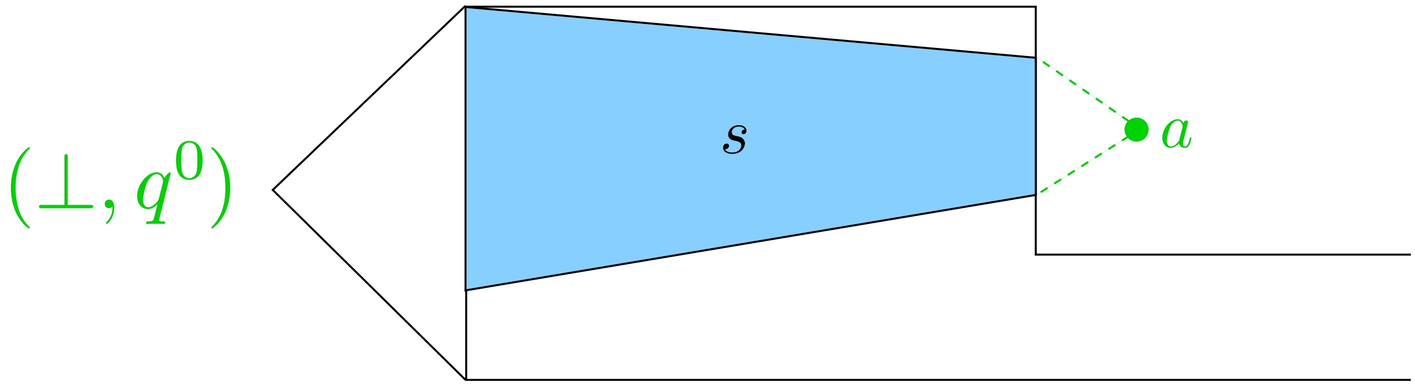
Une partie distribuée est f-maximale si aucun Σ_0-coup compatible avec f ne peut être joué.



Stratégies distribuées : condition de maximalité

Soit f une stratégie distribuée.

Une partie distribuée est f-maximale si aucun Σ_0-coup compatible avec f ne peut être joué.

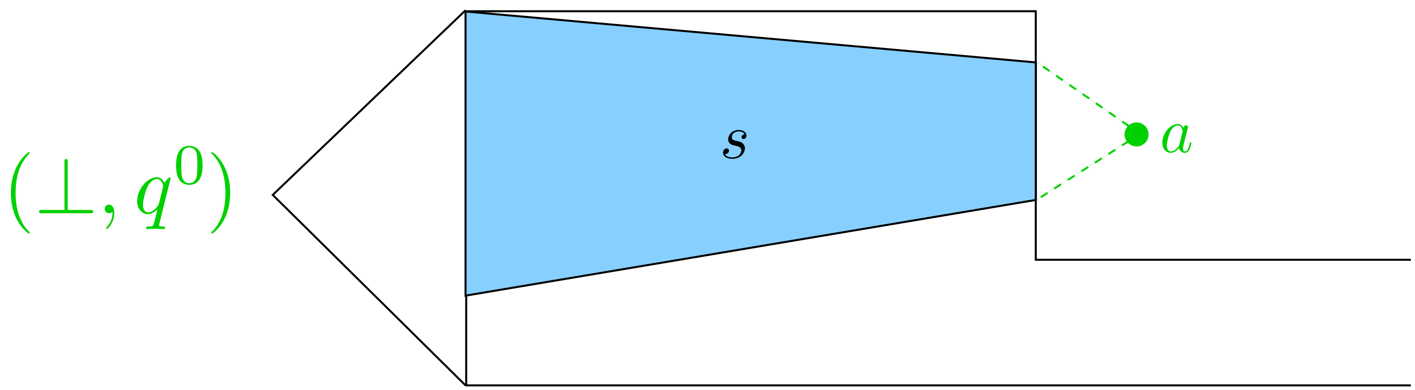


$$a \in \Sigma_0 \Longrightarrow f(s,a) = \text{STOP}$$

Stratégies distribuées : condition de maximalité

Soit f une stratégie distribuée.

Une partie distribuée est f-maximale si aucun Σ_0-coup compatible avec f ne peut être joué.



$$a \in \Sigma_0 \Longrightarrow f(s,a) = \text{STOP}$$

La condition de gain ne concerne que les parties f-maximales.

La stratégie f est gagnante si toutes les f-parties f-maximales sont dans $\mathcal{W}$.

Mémoires génériques

- On associe à chaque joueur $a \in \Sigma$ une mémoire M_a.
- On choisit D' telle que $a \subseteq \mathrm{D}'(a) \subseteq \mathrm{D}(a)$.
- $\mathrm{D}'(a)$ représente les joueurs qui transmettent des informations à a.
- Mise à jour de la mémoire : $Q_{R(a)} \times M_{\mathrm{D}'(a)} \to M_a$.
- Si $\forall a$, $\mathrm{D}'(a) = \{a\}$, on parlera de mémoire locale.
- Si $\forall a$, $\mathrm{D}'(a) = \mathrm{D}(a)$, on parlera de mémoire causale.
- Si f est une stratégie avec une mémoire M, on a $f_a : M_a \times Q_{R(a)} \to Q_{W(a)}$.

Justification de la mémoire causale

- Modélisation
  - Certains systèmes permettent de forcer l'environnement à faire passer les informations nécessaires.
- Calculabilité
  - La mémoire causale représente la plus grosse mémoire envisageable (tout en restant distribuée).
  - L'existence d'une stratégie globale gagnante une condition nécessaire à l'existence d'une stratégie distribuée gagnante.
  - L'existence d'une stratégie gagnante avec mémoire causale est une condition nécessaire à l'existance d'une stratégie distribuée gagnante.

Remarques

- Étant donnés une mémoire M et un jeu asynchrone distribué G, il existe un jeu asynchrone distribué $G \times M$ tel que Σ_0 a une stratégie gagnante sur G si et seulement si Σ_0 a une stratégie gagnante sans mémoire sur $G \times M$.

- Si de plus G et M sont finis, alors $G \times M$ est également fini.

- Étant donné un jeu asynchrone distribué G, il est indécidable de déterminer si Σ_0 à une stratégie gagnante dès que $\mathcal{W}$ est un ensemble rationnel de traces de $\mathbb{R}(\Sigma', \mathrm{D})$.

Comparaison avec les jeux distribués de [MW03]

Autre type de jeux distribués

Jeux distribués déjà étudiés : [PR89], [MT01,02], [MW03]...

Construits à partir de jeux locaux $G_i = \langle P_i, E_i, \mathrm{Tr}_i, q_i^0 \rangle$, $\mathrm{Tr}_i \subseteq (P_i \times E_i)$.

Positions Joueurs : $P = \prod_i (P_i \cup E_i) \setminus E$. Environnement : $E = \prod_i E_i$.

Transitions : $\mathrm{Tr} = \mathrm{Tr}_p \uplus \mathrm{Tr}_e$.

Joueurs : Produit cartésien : $\mathrm{Tr}_p = (\prod_i (\mathrm{Tr}_i \cup \Delta_i)) \cap (P \times E)$ Δ_i: diagonale

Environnement : Globales, données par un sous-ensemble Tr_e de $E \times P$.

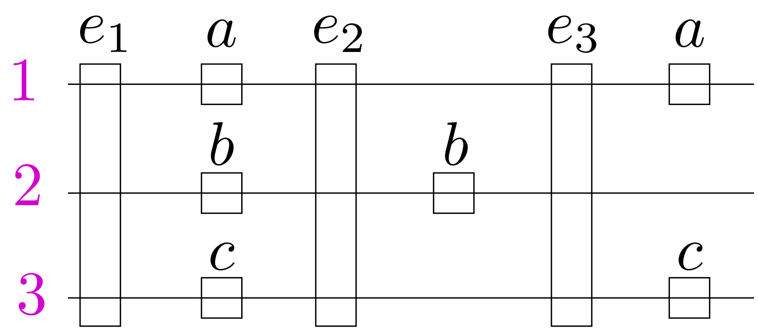
$\rightsquigarrow$ Les informations entre les joueurs transitent par l'environnement.

Les coups des joueurs et de l'environnement alternent.

Condition de gain : $\mathrm{Acc} \subseteq (E \cdot P)^\omega$.

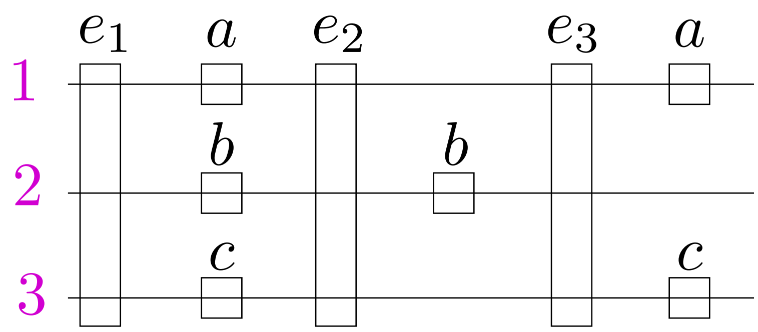
Jeux distribués de [MW03]

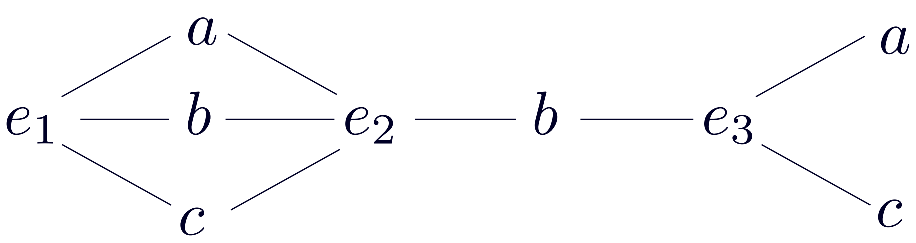
Les parties de ces jeux distribués peuvent être représentées par des traces.



Jeux distribués de [MW03]

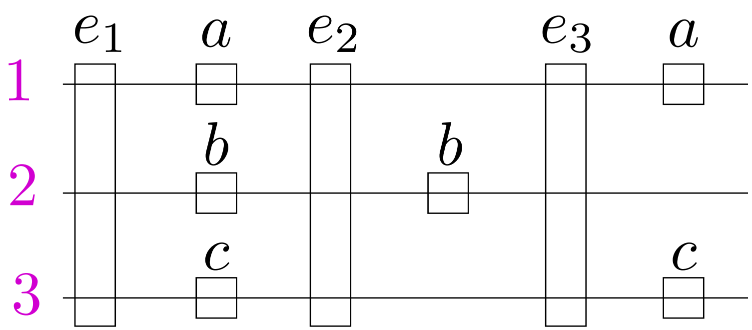
Les parties de ces jeux distribués peuvent être représentées par des traces.

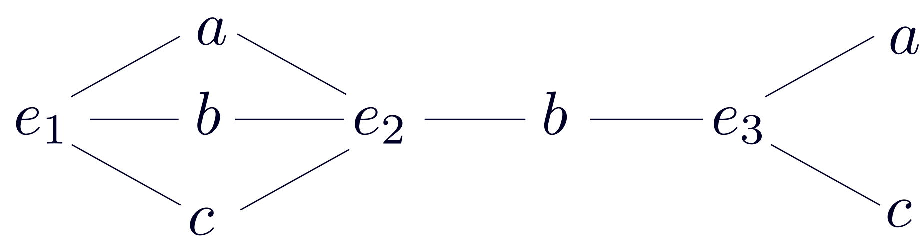




Jeux distribués de [MW03]

Les parties de ces jeux distribués peuvent être représentées par des traces.



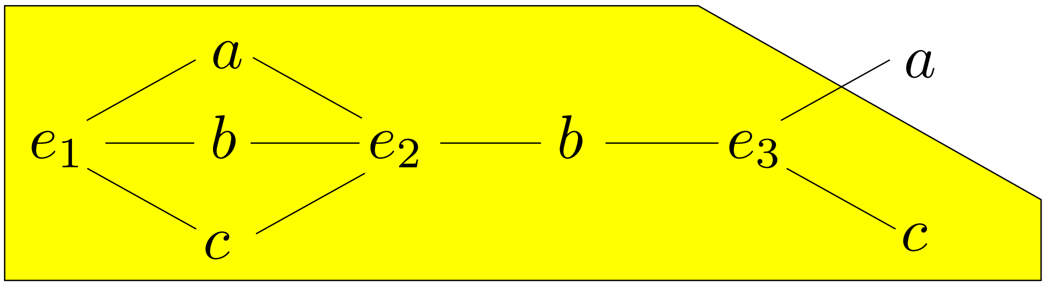


Remarque : l'alphabet est série-parallèle.

Important : Les stratégies utilisent une mémoire locale.

La stratégie d'un joueur ne dépend que de la suite $e_1 p_1 e_2 p_2 \cdots$ qu'il a observé sur sa place mémoire : sa vue est locale et non causale.

Avec une stratégie causale, un joueur pourrait détecter les coups asynchrones.

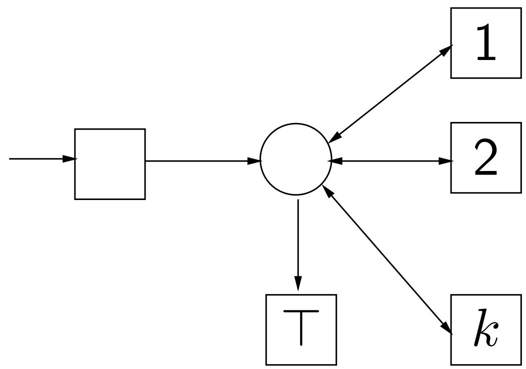


Indécidabilité des jeux distribués [BJW03]

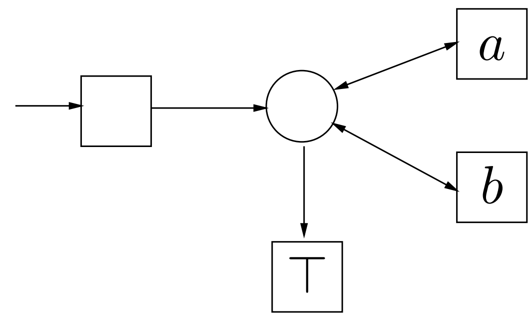
Théorème Le jeu d'accessibilité est indécidable pour les jeux distribués.

À partir d'une instance $u_1, v_1, \ldots, u_k, v_k \in \{a, b\}^*$ du PCP, on construit un jeu à 3 joueurs.

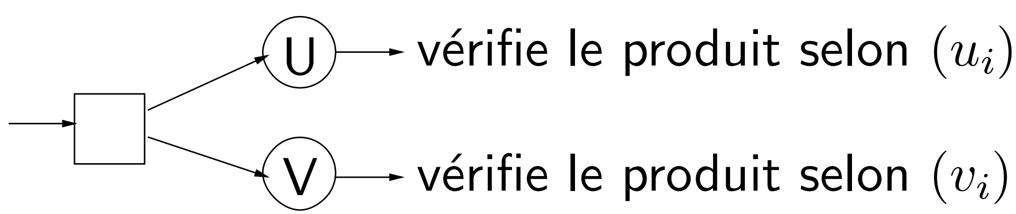
Joueur 1: génère les indices



Joueur 2: génère les lettres



Joueur 3: reçoit de l'environnement ces suites d'indices et de lettres et vérifie que le produit est correct par rapport à (u_i) ou (v_i).



Preuve de la décidabilité du problème d'accessibilité sur les alphabets série-parallèles

Quelques remarques sur les jeux d'accessibilité

Plusieurs types de jeux d'accessibilité

- Global : atteindre un état global vs. Local : atteindre un état local.
- Strict: atteindre et s'arrêter vs. Laxiste : juste atteindre.
  $\mathcal{W} \in \mathsf{Rec}(\mathbb{M}(\Sigma', D))$

Jeux distribués de [MW03] : il est indécidable de déterminer si l'équipe 0 a une stratégie gagnante pour un jeu d'accessibilité dès que l'équipe 0 contient au moins 3 joueurs.

Jeux asynchrones distribués : l'existence d'une stratégie gagnante à mémoire causale pour l'équipe 0 dans un jeu d'accessibilité global et strict est décidable dès que l'alphabet est série-parallèle.

Preuve (1) : stratégies à mémoire finie

Proposition Si Σ_0 a une stratégie f distribuée gagnante sur le jeu d'accessibilité strict, alors l'ensemble des f-parties f-maximales est fini.

En particulier, f est à mémoire finie.

Preuve Lemme de König sur l'arbre de la stratégie.

- Sans borne, ce résultat ne fournit pas d'algorithme.
- On va calculer une borne pour la mémoire ne dépendant que de $|\Sigma|$ et de $|Q|$.

Remarque Pour tout M > 0 il existe un jeu d'accessibilité strict/laxiste global/local pour lequel Σ_0 a besoin d'une mémoire de taille au moins M.

Les mémoires suivantes sont insuffisantes :

⋆ L'état global de la vue de a.

⋆ Des approximations finies de la vue de a.

Preuve (2) : ensemble de f-parties

- Si f est une stratégie, on nomme $P(f)$ l'ensemble des f-parties f-maximales.
- Par la remarque précédente, $P(f)$ est fini.
- Si t est une partie, on nomme $\bar{\sigma}(t)$ l'état global en t.
- Si P est un ensemble de parties, il existe une stratégie f telle que $P = P(f)$ si et seulement si :
  - P vérifie les règles du jeu,
  - P contient tous les coups possibles de l'équipe 1,
  - L'équipe 0 joue de façon distribuée dans P,
  - L'équipe 0 joue de façon déterministe dans P.

Preuve (3) : induction

Proposition On peut calculer $S_M : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ tel que quelque soit Σ, Q, δ, q_0, F, si il existe une stratégie gagnante f pour l'équipe 0 dans le jeu $(\Sigma, Q, \delta, q_0, F)$, alors il existe une stratégie gagnante de mémoire inférieure à $S_M(|\Sigma|, |Q|)$.

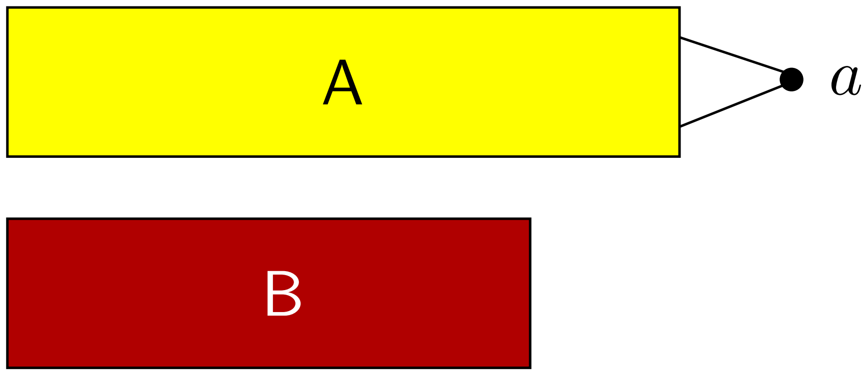
Preuve On procède par induction sur $|\Sigma|$.

À partir de f, on calcule f' à mémoire au plus $S_M(|\Sigma|, |Q|)$.

Pour $|\Sigma| = 1$, le jeu est un jeu séquentiel.

Preuve (4) : cas $\Sigma = A \parallel B$

Pour $\Sigma = A \parallel B$, on montre que $\bar{\sigma}(P(f))$ est un produit direct $\mathcal{W}_A \times \mathcal{W}_B$



On peut considérer les projections f_A et f_B de f sur G_A et G_B.

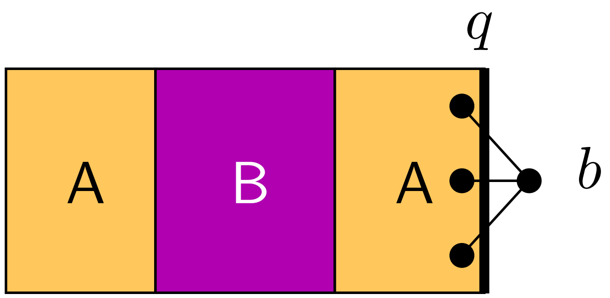
L'hypothèse d'induction fournit f'_A sur G_A de mémoire inférieure à $S_M(|A|, |Q|)$

f'_B sur G_B de mémoire inférieure à $S_M(|B|, |Q|)$.

On en déduit que $f' = f'_A \times f'_B$ est gagnante sur G et de mémoire inférieure à : $S_M(|A|, |Q|) + S_M(|B|, |Q|)$.

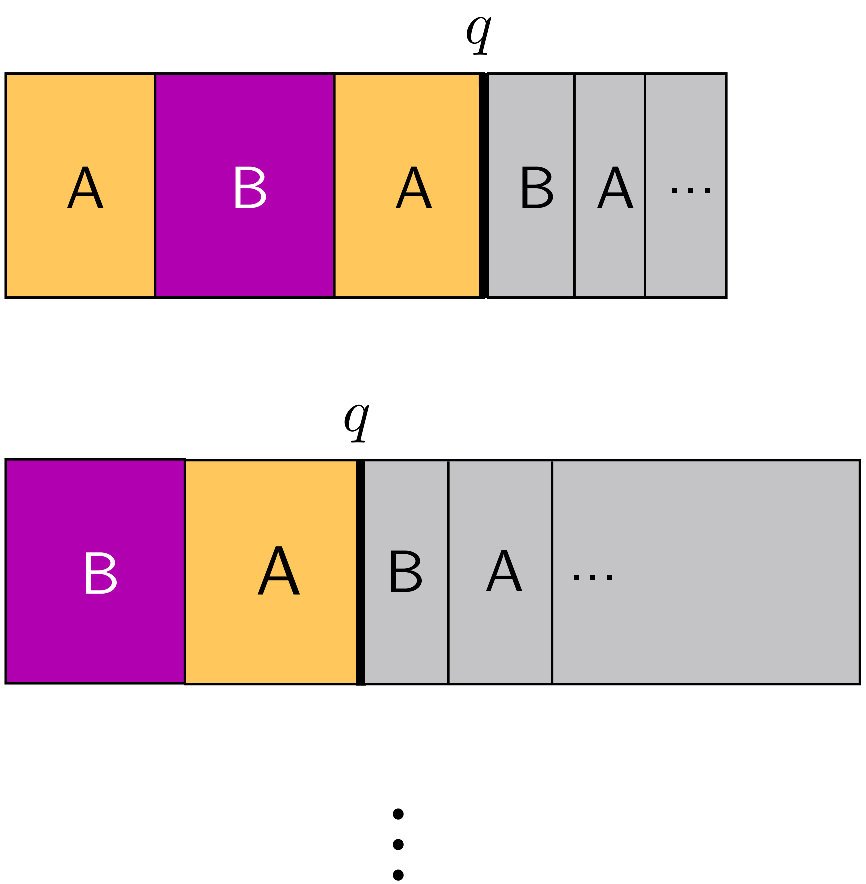
Preuve (5) : cas $\Sigma = A \otimes B$

Pour $\Sigma = A \otimes B$, on transforme $P(f)$ en P'.



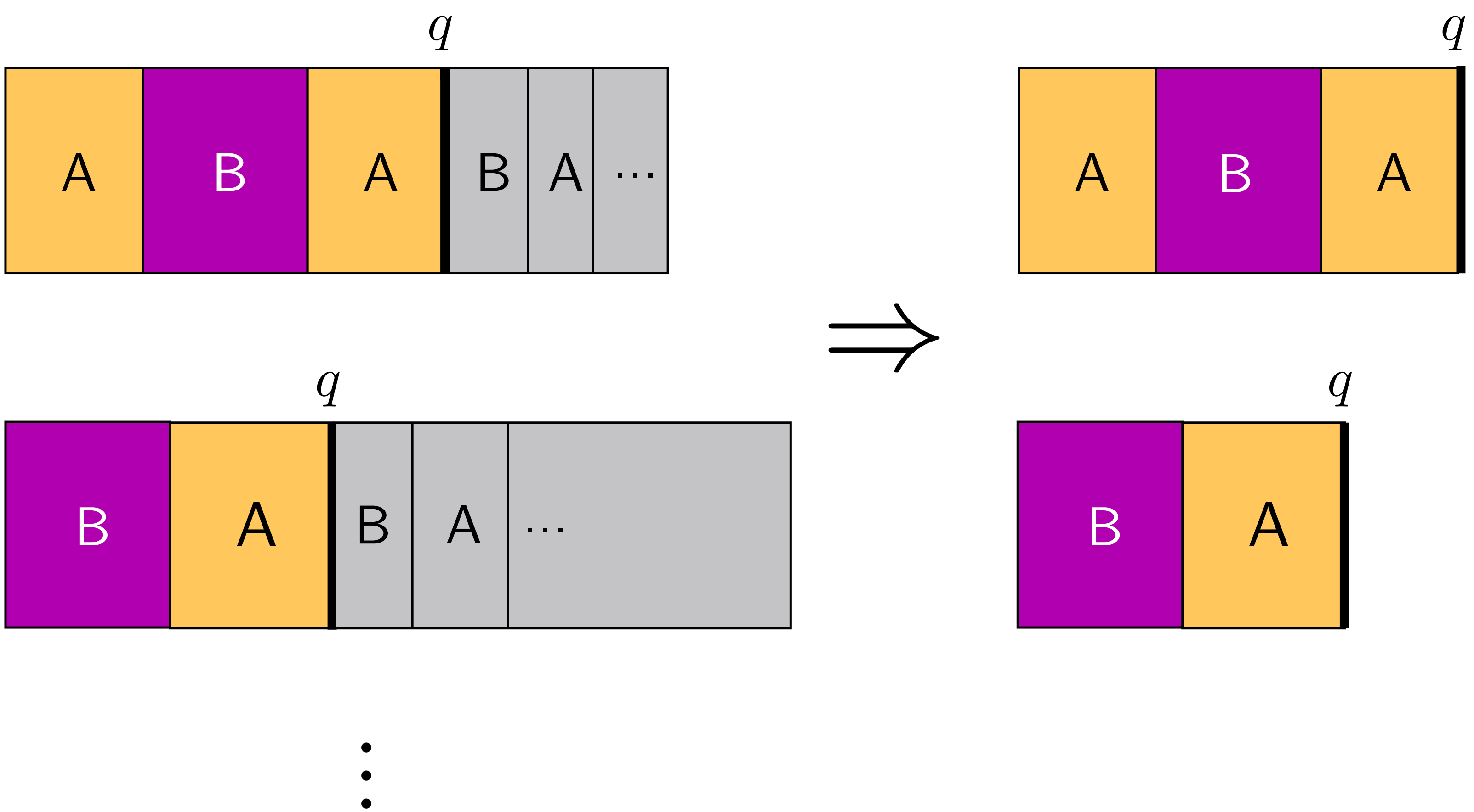
Preuve (5) : cas $\Sigma = A \otimes B$

Pour $\Sigma = A \otimes B$, on transforme $P(f)$ en P'.



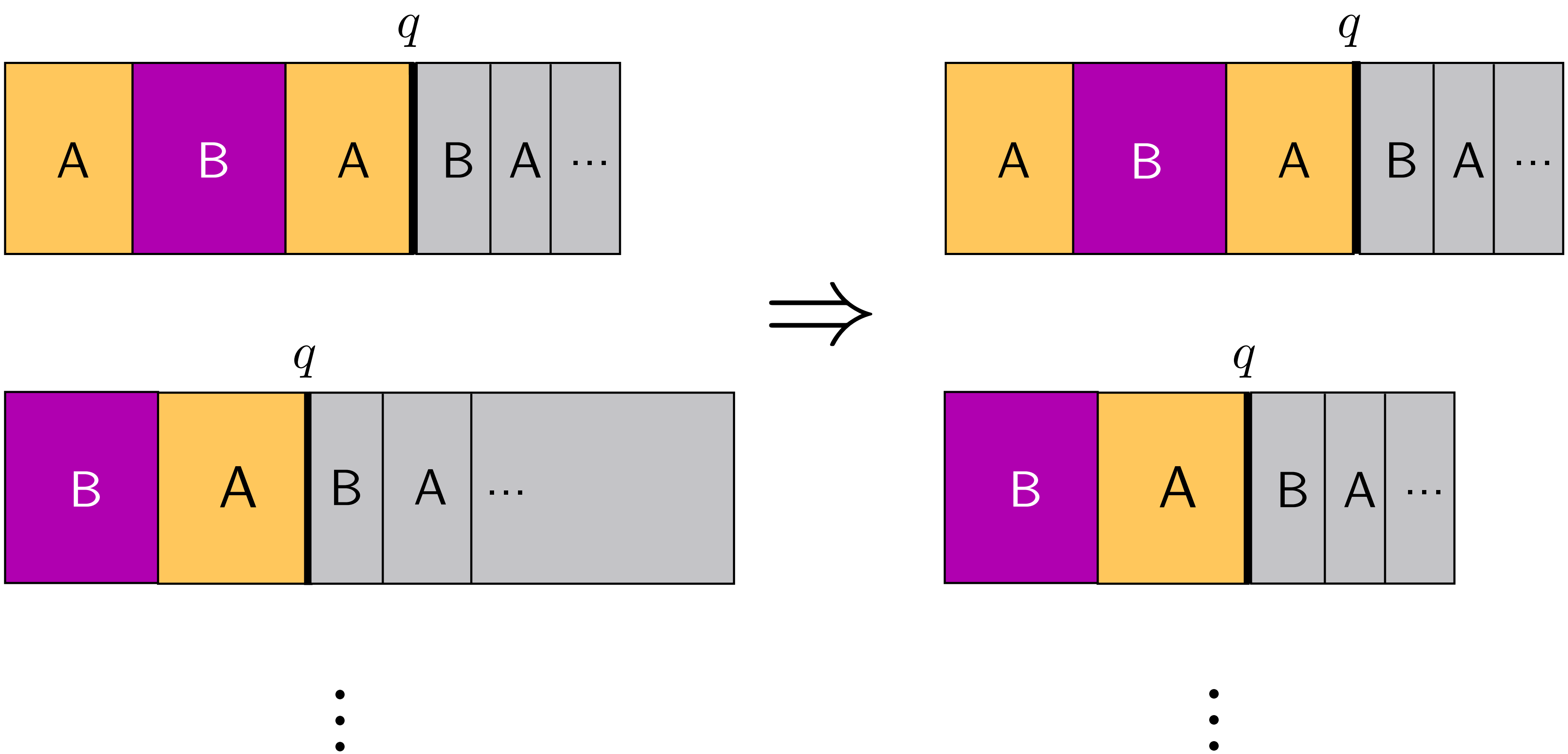
Preuve (5) : cas $\Sigma = A \otimes B$

Pour $\Sigma = A \otimes B$, on transforme $P(f)$ en P'.



Preuve (5) : cas $\Sigma = A \otimes B$

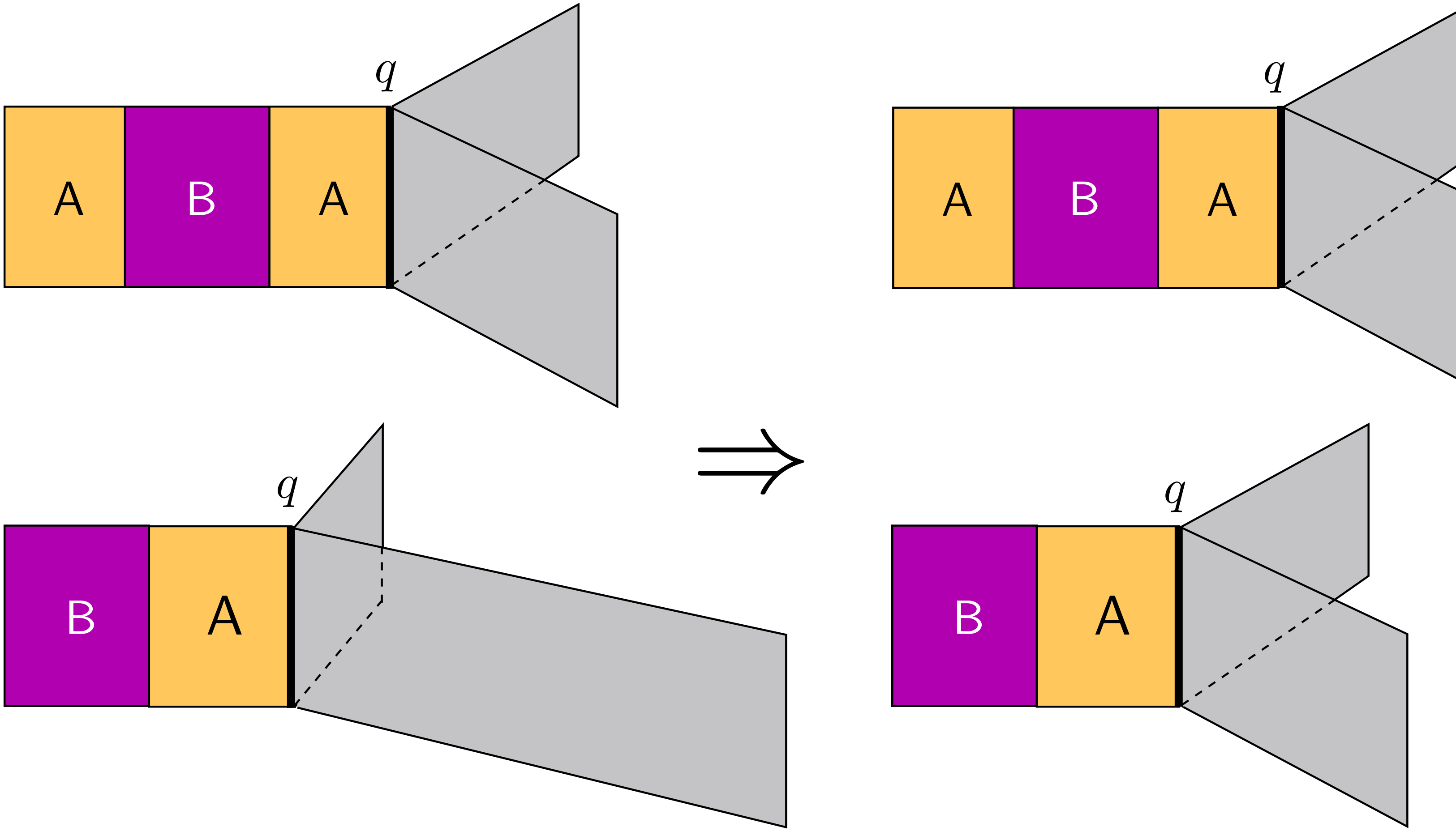
Pour $\Sigma = A \otimes B$, on transforme $P(f)$ en P'.



Preuve (6) : cas $\Sigma = A \otimes B$ (suite)

Ces découpages et recollages sont faits uniformément sur tous les suffixes qui prolongent un contexte (A, q).

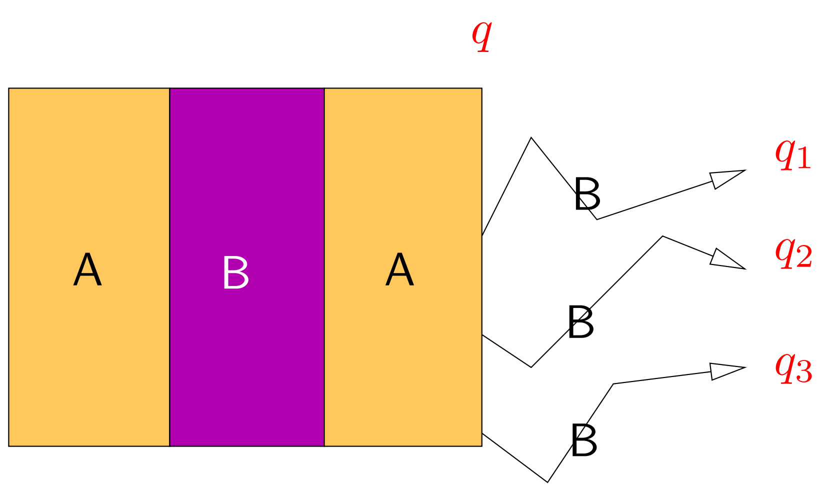
On garde l'ensemble de suffixes qui gagne le plus rapidement dans le cas le pire.



Preuve (7)

On obtient une nouvelle stratégie f' telle que $P' = P(f')$.

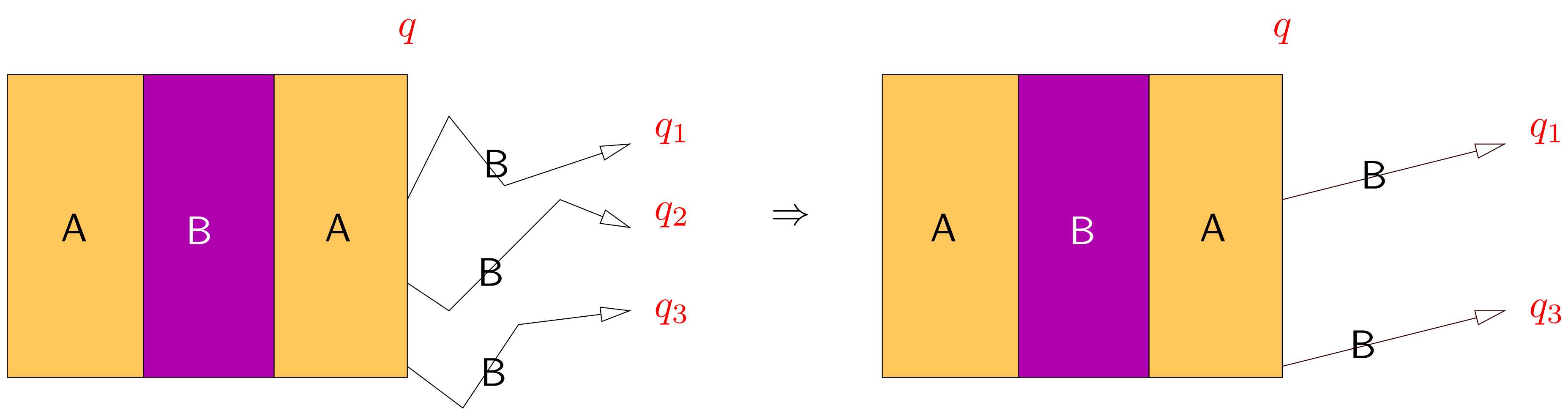
On transforme modulairement P' à l'aide de l'hypothèse d'induction sur chacun des alphabet A et B.



Preuve (7)

On obtient une nouvelle stratégie f' telle que $P' = P(f')$.

On transforme modulairement P' à l'aide de l'hypothèse d'induction sur chacun des alphabet A et B.



On obtient $P'' = P(f'')$ de mémoire inférieure à :
$\max(S_M(|A|, |Q|), S_M(|B|, |Q|)) + |Q|$.

Conclusion et perspectives

⋆ Les jeux asynchrones distribués à mémoire causale donnent des conditions nécessaires à l'existence de stratégies gagnantes

⋆ L'existence d'une stratégie gagnante est décidable pour une condition de gain reconnaissable de traces finies sur les cographes

⋆ Quelle est la complexité du problème, est-il possible d'améliorer la recherche de stratégie gagnante ?

⋆ Qu'en est-il pour les conditions reconnaissables sur les traces infinies ou sur un alphabet quelconque ?