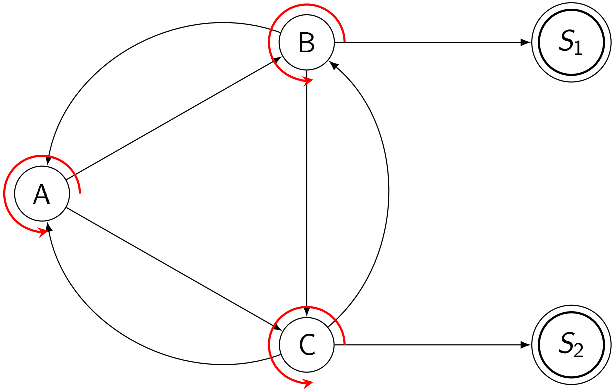
Jeu d'accessibilité lié à une marche de rotors sur un graphe

Pierre Coucheney, David Auger et Loric Duhazé

Laboratoire DAVID, UVSQ

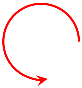
juillet 2022

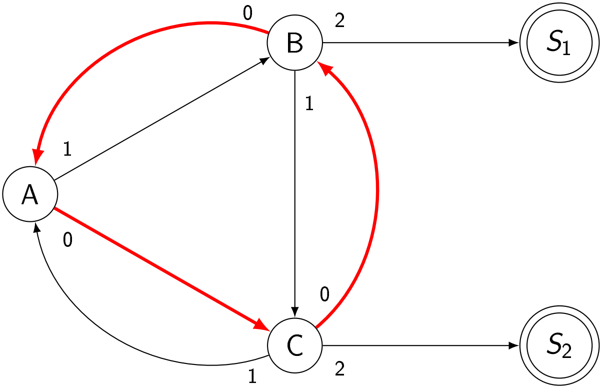
On considère un graphe orienté et on fixe un **ordre cyclique (=rotor)** sur les arcs sortants de chaque sommet



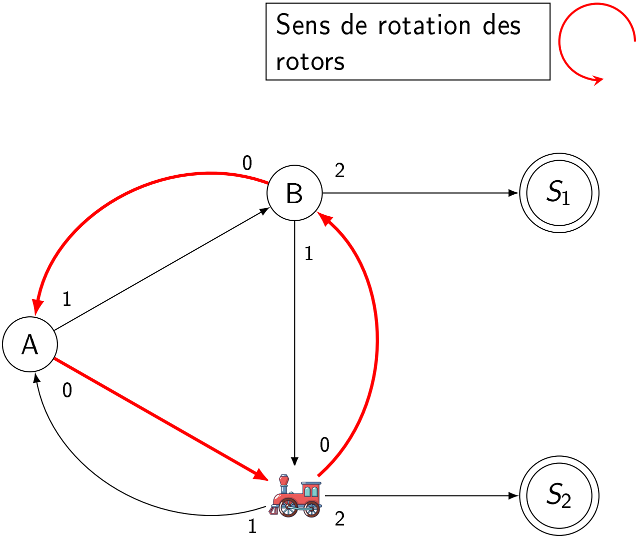
Configuration de Rotors

Sens de rotation (de l'ordre) des rotors





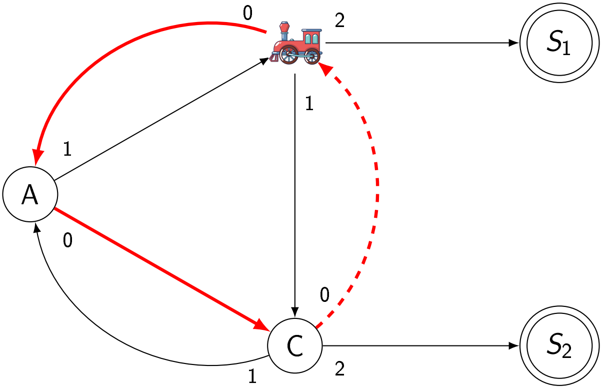
Marche de Rotors



Marche de Rotors

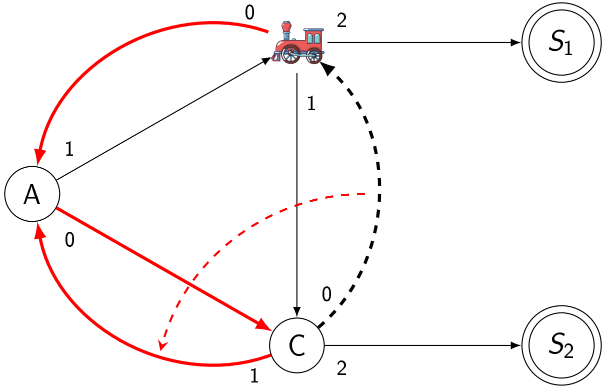
Mouvement de train

Sens de rotation des rotors

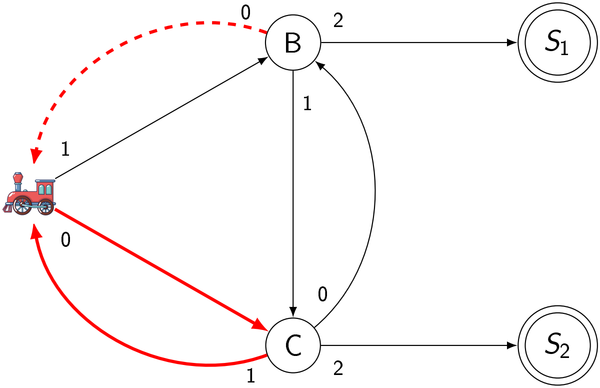


Mouvement de train

Sens de rotation des rotors

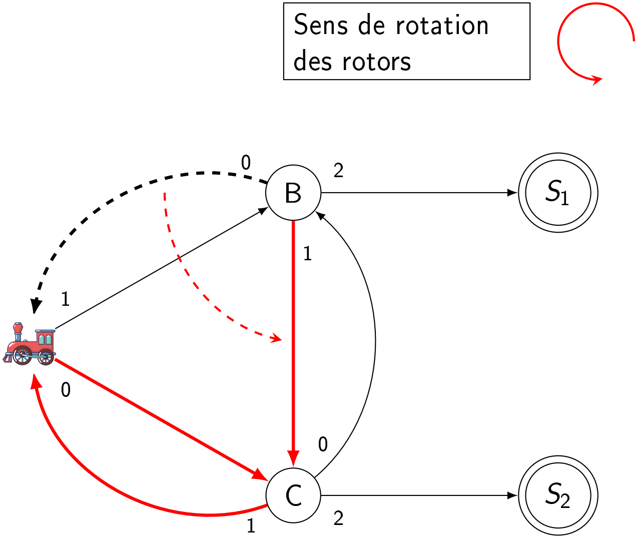


Sens de rotation des rotors

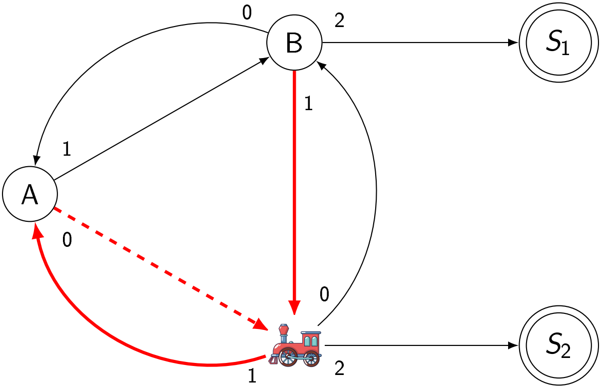


Marche de Rotors

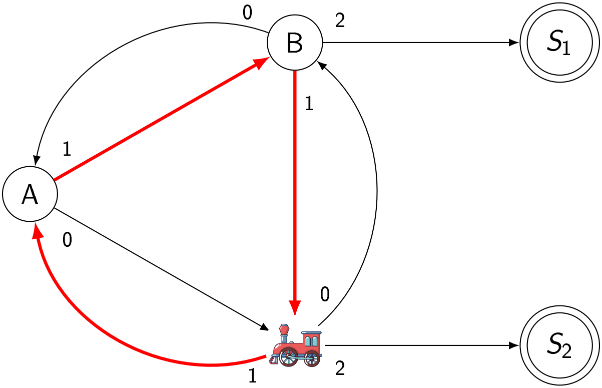
Sens de rotation
des rotors



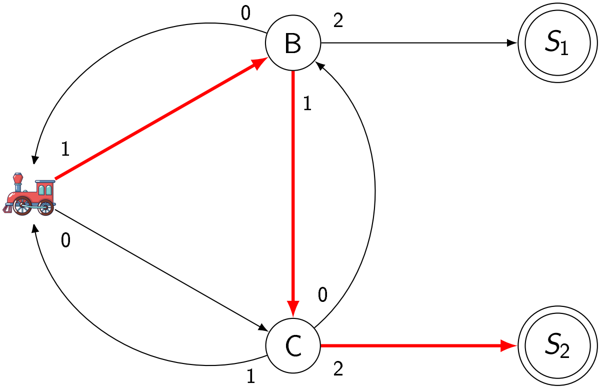
Sens de rotation des rotors



Sens de rotation des rotors

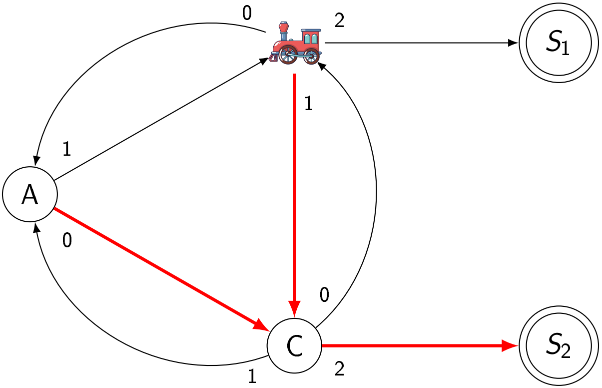


Sens de rotation des rotors

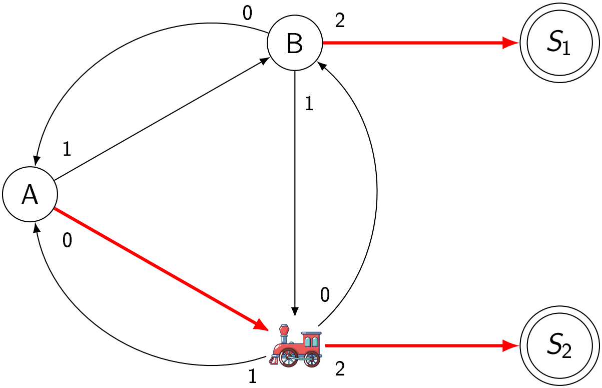


Marche de Rotors

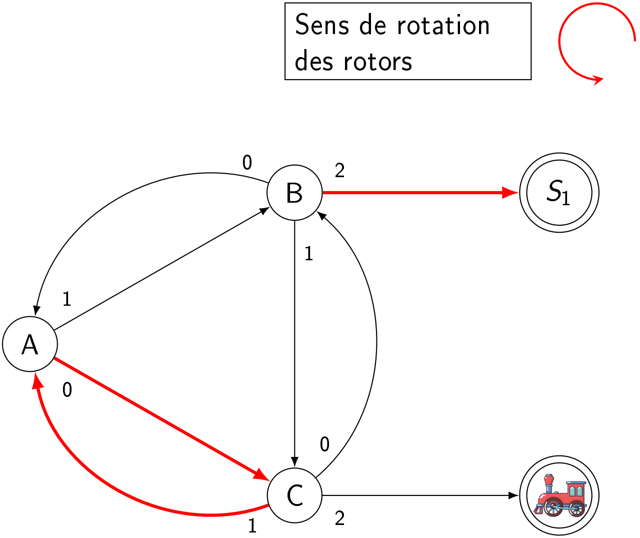
Sens de rotation des rotors



Sens de rotation des rotors



Marche de Rotors



Quelques résultats liés aux marches de rotors (aussi appelées machine de Propp, marches eulériennes)

- sous certaines conditions, convergence rapide vers des quantités naturelles associées aux marches aléatoires (par ex. probabilité d'atteinte d'un sommet) pour certaines familles de graphes
- algorithmes d'équilibrage de charge

Soit $G = (V_0 \cup S_0, A)$ un graphe orienté :

- S_0 : ensemble de puits
- chaque sommet $v \in V_0$ a un chemin vers S_0

Rotor : pour chaque sommet $v \in V_0$, ordre cyclique sur les arcs sortants $A^+(v)$; si $a \in A^+(v)$, on note a^+ le successeur selon cet ordre.

Configuration de Rotors :

$$\begin{array}{rcl} \rho : V_0 & \to & A \\ v & \mapsto & \rho(v) \in A^+(v) \end{array}$$

Marche de Rotors

$$(\rho_0, v_0), (\rho_1, v_1), \dots, (\rho_T, v_T), T \in \mathbb{N} \cup \{+\infty\}$$

telle que

- $(v_t, v_{t+1}) = \rho(v_t)$
- $\rho_{t+1}(v) = \begin{cases} \rho_t^+(v) & \text{si } v = v_t \\ \rho_t(v) & \text{sinon} \end{cases}$

$$(\rho_0, v_0), (\rho_1, v_1), \ldots, (\rho_T, v_T), T \in \mathbb{N} \cup \{+\infty\}$$

telle que

- $(v_t, v_{t+1}) = \rho(v_t)$
- $\rho_{t+1}(v) = \begin{cases} \rho_t^+(v) & \text{si } v = v_t \\ \rho_t(v) & \text{sinon} \end{cases}$

Propriété

Toute marche de rotors est finie.

Preuve :

- le nombre de passages sur un puits est fini
- vrai aussi pour les voisins des puits
- et pour les voisins des voisins, etc.

Nombre d'étapes au plus exponentiel.

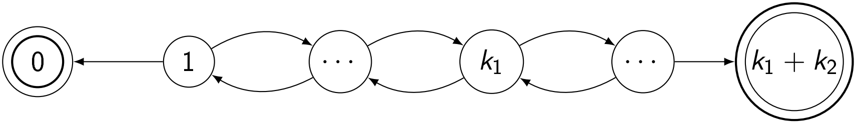
Problème de décision ARRIVAL (Dohrau et al, 2017)

Etant donnés :

- une configuration de rotors ρ_0
- un sommet v_0 de départ
- un puits $s \in S_0$

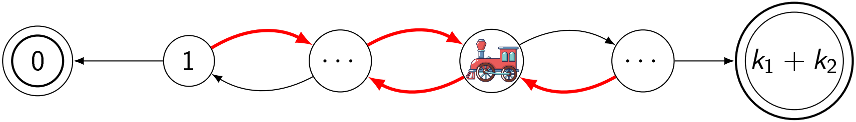
Est-ce que la marche de rotors démarrant en (ρ_0, v_0) termine sur le puits s ?

- Alice a k_1 euros
- Bob a k_2 euros
- itérativement, avec proba p Alice prend 1 euro à Bob, et avec proba $1 - p$ Bob prend 1 euro à Alice
- quelle est la probabilité que Alice soit ruinée ?



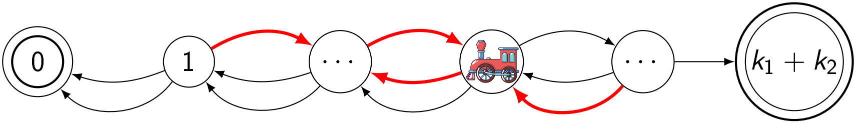
Exemple : ruine du joueur revisitée

- Alice a k_1 euros
- Bob a k_2 euros
- itérativement, avec proba p Alice prend 1 euro à Bob, et avec proba $1 - p$ Bob prend 1 euro à Alice
- quelle est la probabilité que Alice soit ruinée ?

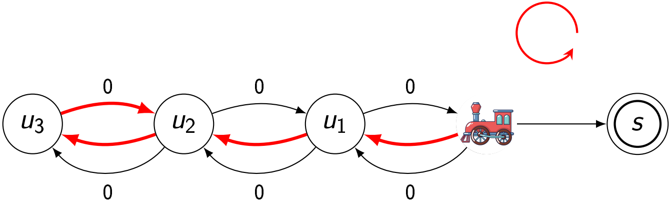


Exemple : ruine du joueur revisitée

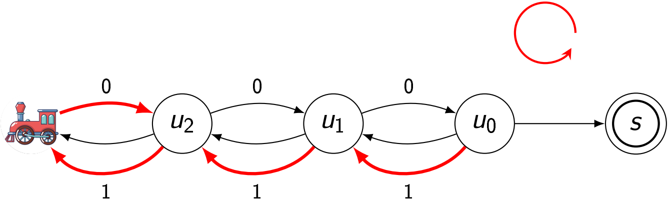
- Alice a k_1 euros
- Bob a k_2 euros
- itérativement, avec proba p Alice prend 1 euro à Bob, et avec proba $1 - p$ Bob prend 1 euro à Alice
- quelle est la probabilité que Alice soit ruinée ?



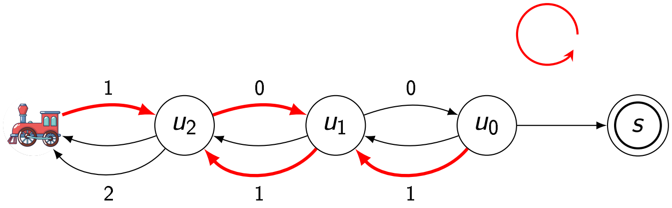
Un routage exponentiel



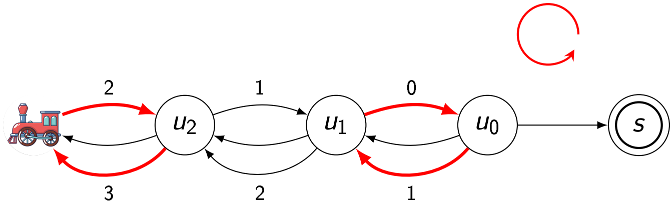
Un routage exponentiel



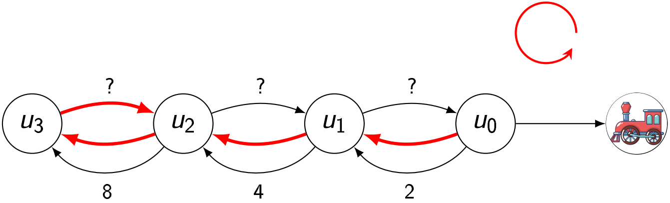
Un routage exponentiel



Un routage exponentiel



Un routage exponentiel



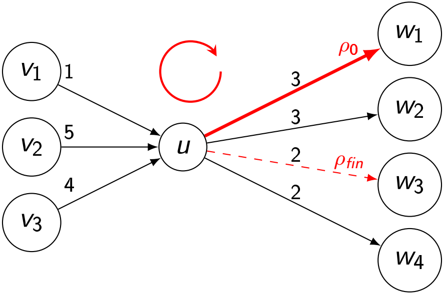
Certificat vérifiable en temps polynomial que la marche de rotor débutant en (ρ_0, v_0) termine en s :

On donne $(f_a)_{a \in A}$ un flot sur les arcs du graphe qui vérifie :

- $f_a \in \mathbb{N}$
- équation de balance locale sur chaque sommet v

$$\sum_{a \in A^-(v)} f_a + \mathbb{1}_{v=v_0} = \sum_{a \in A^+(v)} f_a + \mathbb{1}_{v=s}$$

- compatibilité avec l'ordre de rotor :



Un tel flot certifie la terminaison en s

- existence du flot : compter le nombre de passages sur chaque arc de la marche de rotors
- suffisant pour prouver la terminaison en s

Flot certificat (suite)

Un tel flot certifie la terminaison en s

- existence du flot : compter le nombre de passages sur chaque arc de la marche de rotors
- suffisant pour prouver la terminaison en s

Peut être étendu à plusieurs marches de rotors (plusieurs trains), ce qui prouve la commutativité.

ARRIVAL : résultats connus

- Ce problème est dans NP $\cap$ co-NP

ARRIVAL : résultats connus

- Ce problème est dans NP $\cap$ co-NP
- Pas d'algorithme polynomial connu en général

- Ce problème est dans NP $\cap$ co-NP
- Pas d'algorithme polynomial connu en général
- Pour les digraphes Eulériens $O(n^3)$ en effectuant la marche de rotors (Priezzhev, 96)

ARRIVAL : résultats connus

- Ce problème est dans NP $\cap$ co-NP
- Pas d'algorithme polynomial connu en général
- Pour les digraphes Eulériens $O(n^3)$ en effectuant la marche de rotors (Priezzhev, 96)
- Algorithme sous-exponentiel (Gärtner et al, 2021) : calculer le flot par un algorithme de point fixe.

Cycle pushing (Wilson, 96)

Le **cycle push** d'un cycle $C \in \rho$ transforme ρ en ρ' tel que :

$$\rho'(v) = \begin{cases} \rho^+(v) & \text{si } v \in C \\ \rho(v) & \text{sinon} \end{cases}$$

Cycle pushing (Wilson, 96)

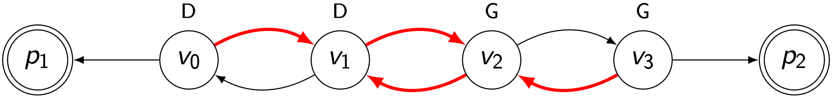
Le **cycle push** d'un cycle $C \in \rho$ transforme ρ en ρ' tel que :

$$\rho'(v) = \begin{cases} \rho^+(v) & \text{si } v \in C \\ \rho(v) & \text{sinon} \end{cases}$$

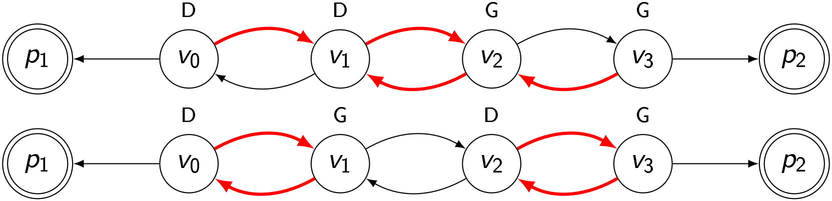
Propriétés :

- ne change pas le puits dans lequel termine la marche de rotor
- toute séquence maximale de cycle push termine sur une configuration acyclique, appelée **forêt de destination**, qui ne dépend pas de l'ordre de ces opérations
- la forêt de destination donne les puits de sortie pour chaque sommet de départ

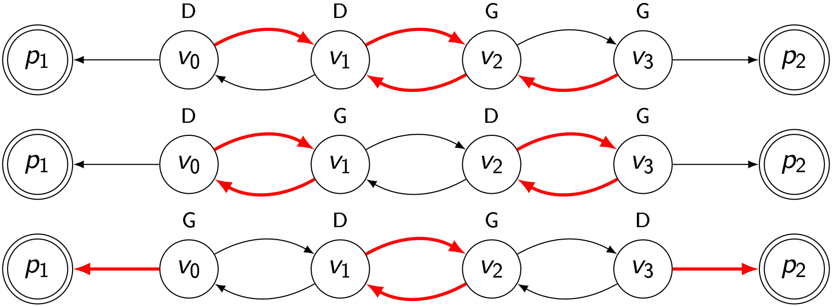
Ruine du joueur revisitée

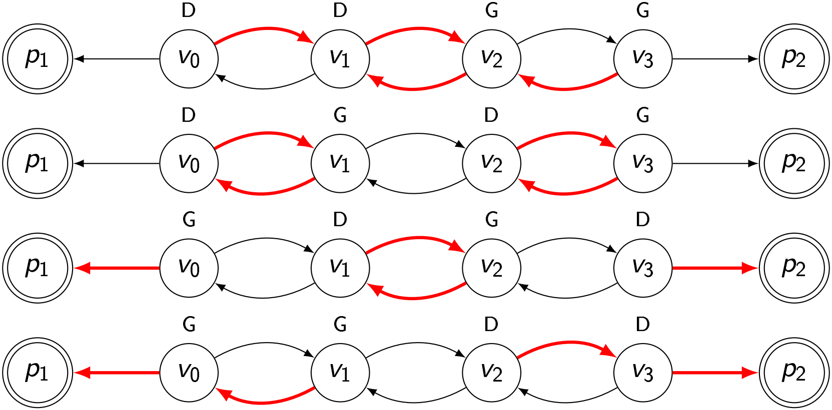


Ruine du joueur revisitée

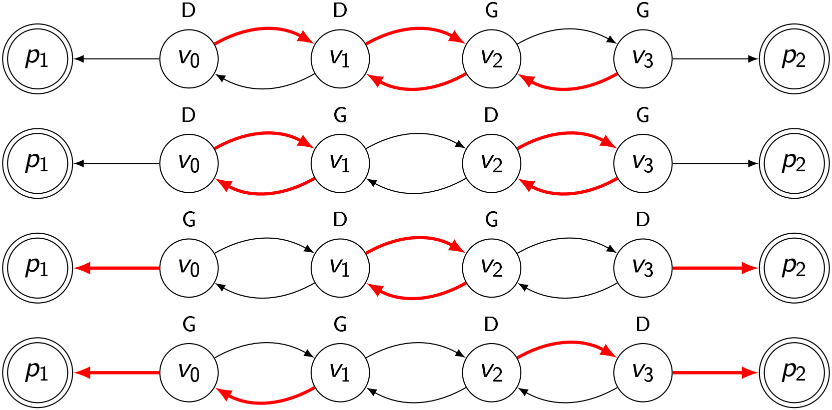


Ruine du joueur revisitée



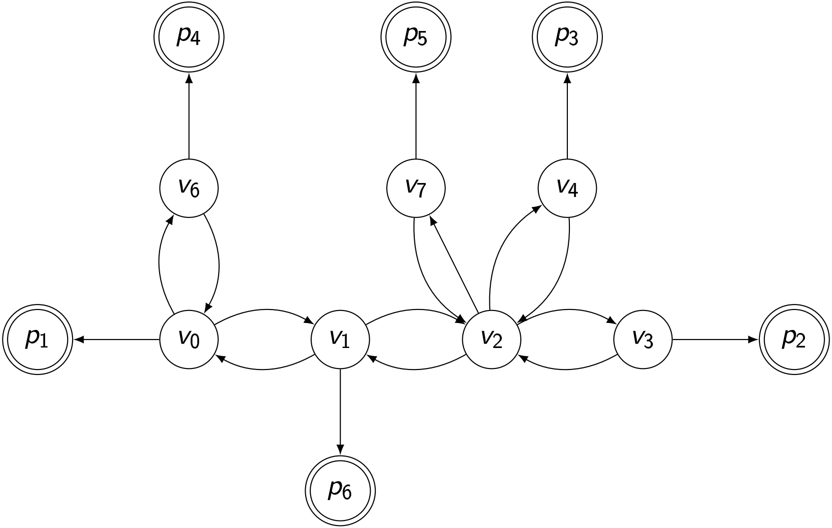

D D G G
p1 v0 v1 v2 v3 p2
D G D G
p1 v0 v1 v2 v3 p2
G D G D
p1 v0 v1 v2 v3 p2
G G D D
p1 v0 v1 v2 v3 p2

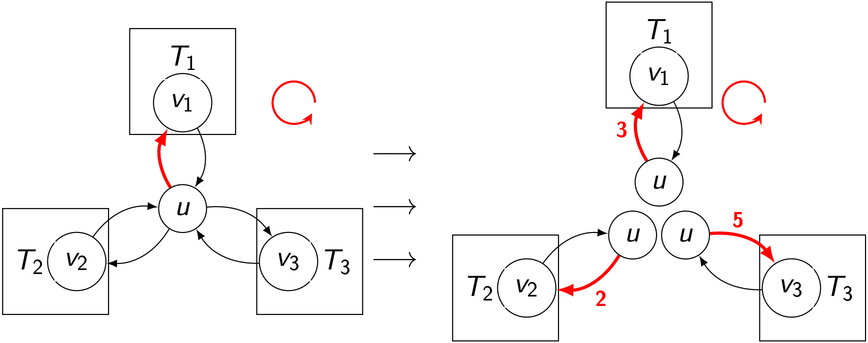
Ruine du joueur revisitée



Nous allons calculer la forêt de destination dans les **multigraphes en forme d'arbre**, et ainsi résoudre ARRIVAL pour tous les sommets de départ en même temps.

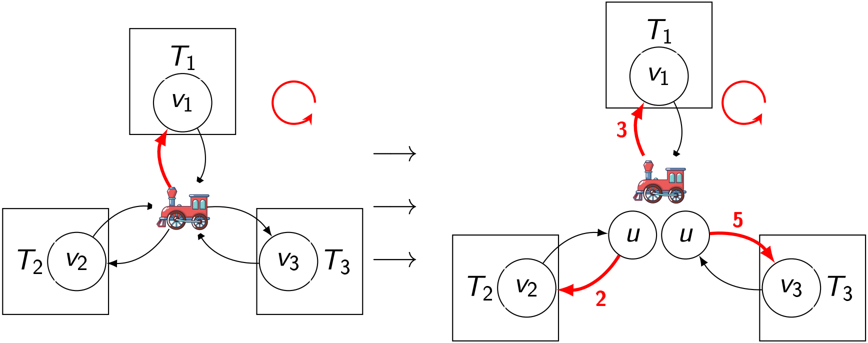
(Multi) Graphe en forme d'arbre





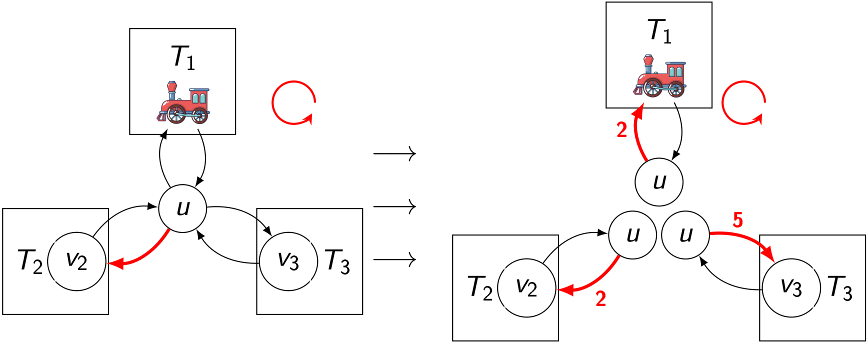
Flot de retour : flot sur l'arête partant de u pour chaque sous-arbre (valeur écrite en rouge).

Flot de retour pour les graphes en forme d'arbre



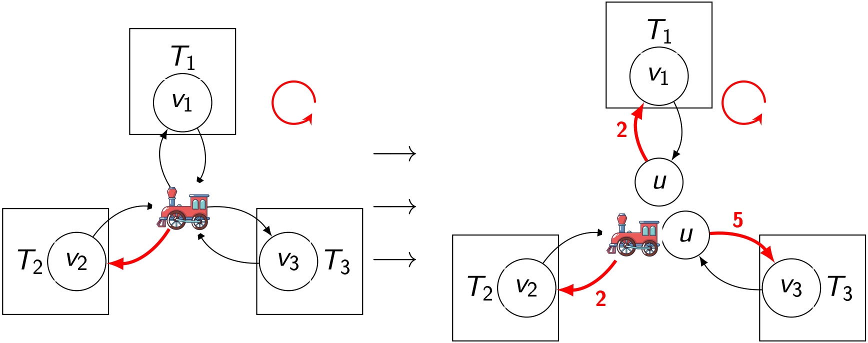
Flot de retour : flot sur l'arête partant de u pour chaque sous-arbre (valeur écrite en rouge).

Flot de retour pour les graphes en forme d'arbre



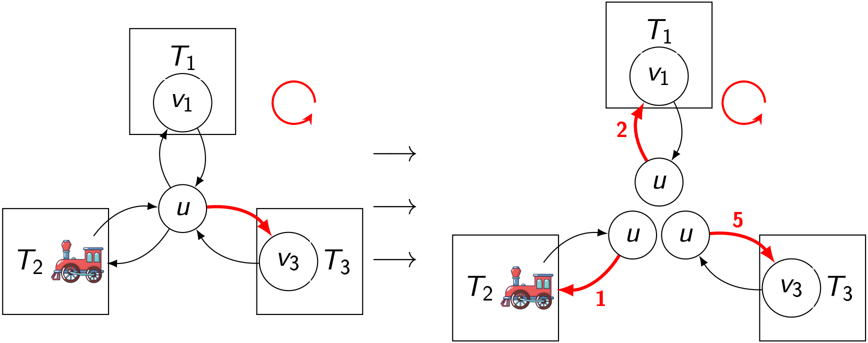
Flot de retour : flot sur l'arête partant de u pour chaque sous-arbre (valeur écrite en rouge).

Flot de retour pour les graphes en forme d'arbre



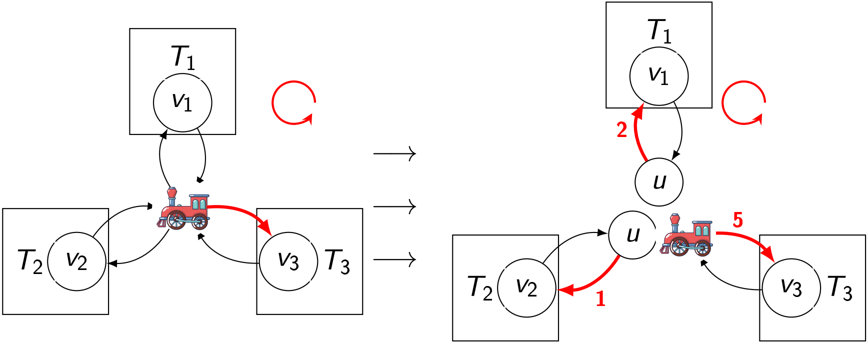
Flot de retour : flot sur l'arête partant de u pour chaque sous-arbre (valeur écrite en rouge).

Flot de retour pour les graphes en forme d'arbre



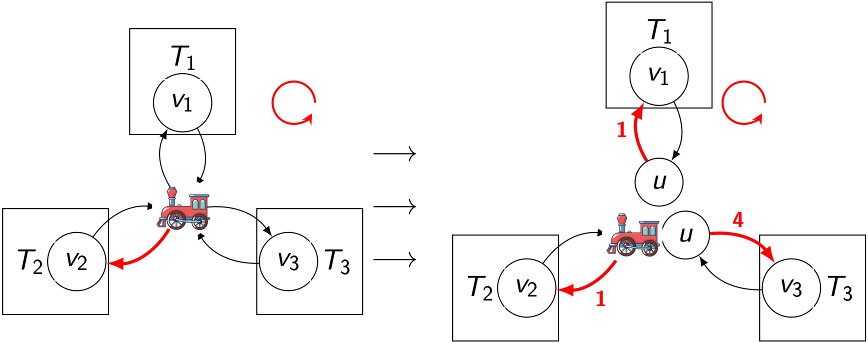
Flot de retour : flot sur l'arête partant de u pour chaque sous-arbre (valeur écrite en rouge).

Flot de retour pour les graphes en forme d'arbre



Flot de retour : flot sur l'arête partant de u pour chaque sous-arbre (valeur écrite en rouge).

Flot de retour pour les graphes en forme d'arbre

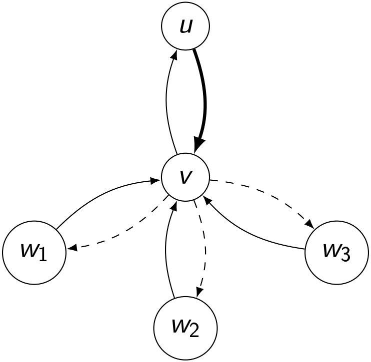


Flot de retour : flot sur l'arête partant de u pour chaque sous-arbre (valeur écrite en rouge).

Arc de la forêt de destination issu de u : $(u, v2)$

Pour les graphes simples, calcul récursif linéaire depuis les puits :

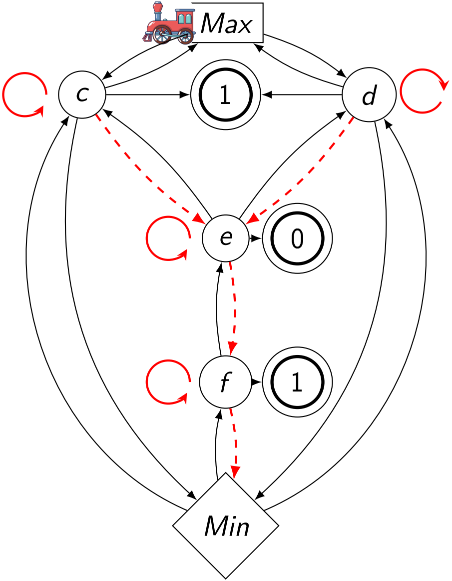
$$r_\rho(u,v) = \begin{cases} 1 & \text{si } v \in S_0 \\ \min_{w \in \Gamma^+(v)\setminus\{u\}} r_\rho(v,w) + \mathbb{1}_{\rho(v)\leq(v,u)\leq(v,w)} & \text{sinon} \end{cases}$$



Résultats

	Marche de rotors	0 joueur	1 joueur	2 joueurs
Graphe quelconque	exponentiel	NP $\cap$ coNP	NP-complet	PSPACE-complet
Multigraphe en forme d'arbre	exponentiel	$O(\|A\|)$	$O(\|A\|)$	$O(\|A\|)$
Modèle aléatoire (G. quelconque)		Poly($\|A\|$)	Poly($\|A\|$)	NP $\cap$ coNP

Jeu d'accessibilité à 2 joueurs



Equivalent au jeu "matching pennies" : pas d'équilibre en stratégies pures.