Cours 10/3/21

Bisimulation $R \subseteq S_1 \times S_2$

⇝ comparaison de systèmes de transition (ST) : T_1, T_2

$T_i = (S_i, \Sigma, \delta_i, q_0^i)$

$$\begin{array}{ccc} s_1 & R & s_2 \\ \forall\ \downarrow a & & \downarrow a\ \exists \\ s_1' & R & s_2' \end{array} \qquad \Bigg| \qquad \begin{array}{ccc} s_1 & R & s_2 \\ \exists\ \downarrow a & & \downarrow a\ \forall \\ s_1' & R & s_2' \end{array}$$

<u>Ex</u> : Bakery algorithm (exclusion mutuelle)

$\boxed{\text{Proc 1}}$ Init : $x_1 = x_2 = 0$ $\boxed{\text{Proc 2}}$

symm.

...

<u>while</u> <u>true</u> <u>do</u>

n_1 : $x_1 := x_2 + 1$

w_1 : <u>wait until</u> ($x_2 = 0$ OR $x_1 < x_2$)

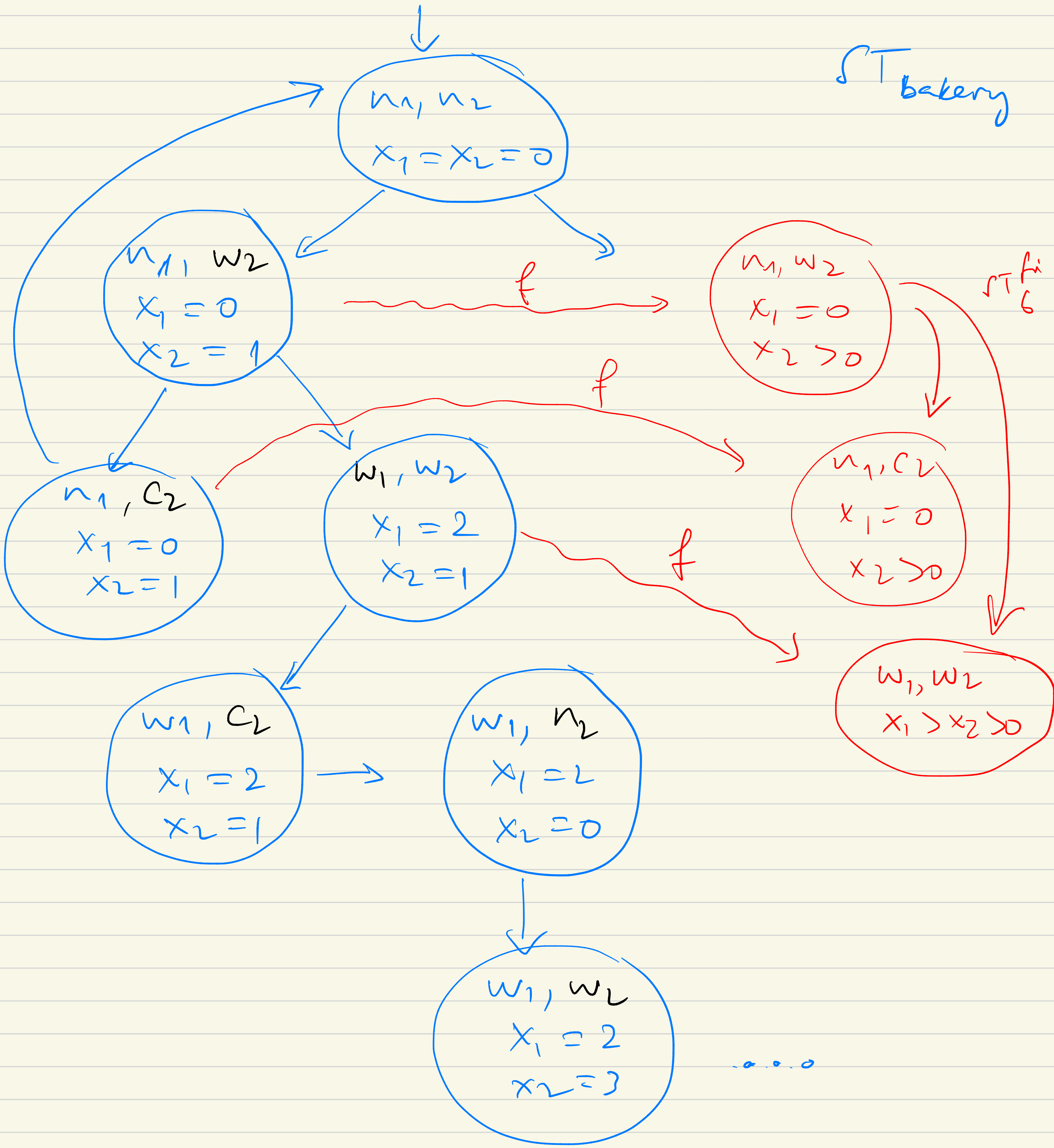
critical section

c_1 : $x_1 := 0$;

<u>od</u>

le ST associé est infini :

ST_{bakery}



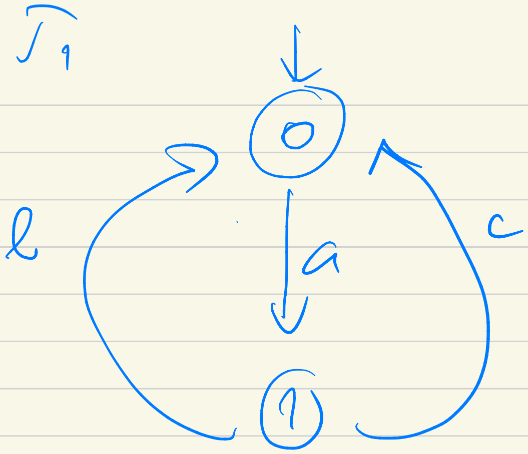
Mais il ex. un $\boxed{ST \text{ fini}}$ $ST^{\text{fini}}_{\text{bakery}}$

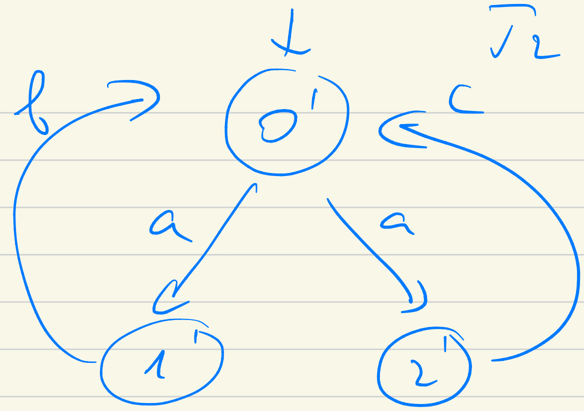
tq. $ST_{\text{bakery}} \sim_{\text{bisim.}} ST^{\text{fi}}_{\text{bakery}}$

Etat $s = (l_1, l_2, x_1 = b_1, x_2 = b_2)$ de

ST_{bakery}, $l_i \in \{n_i, w_i, c_i\}$

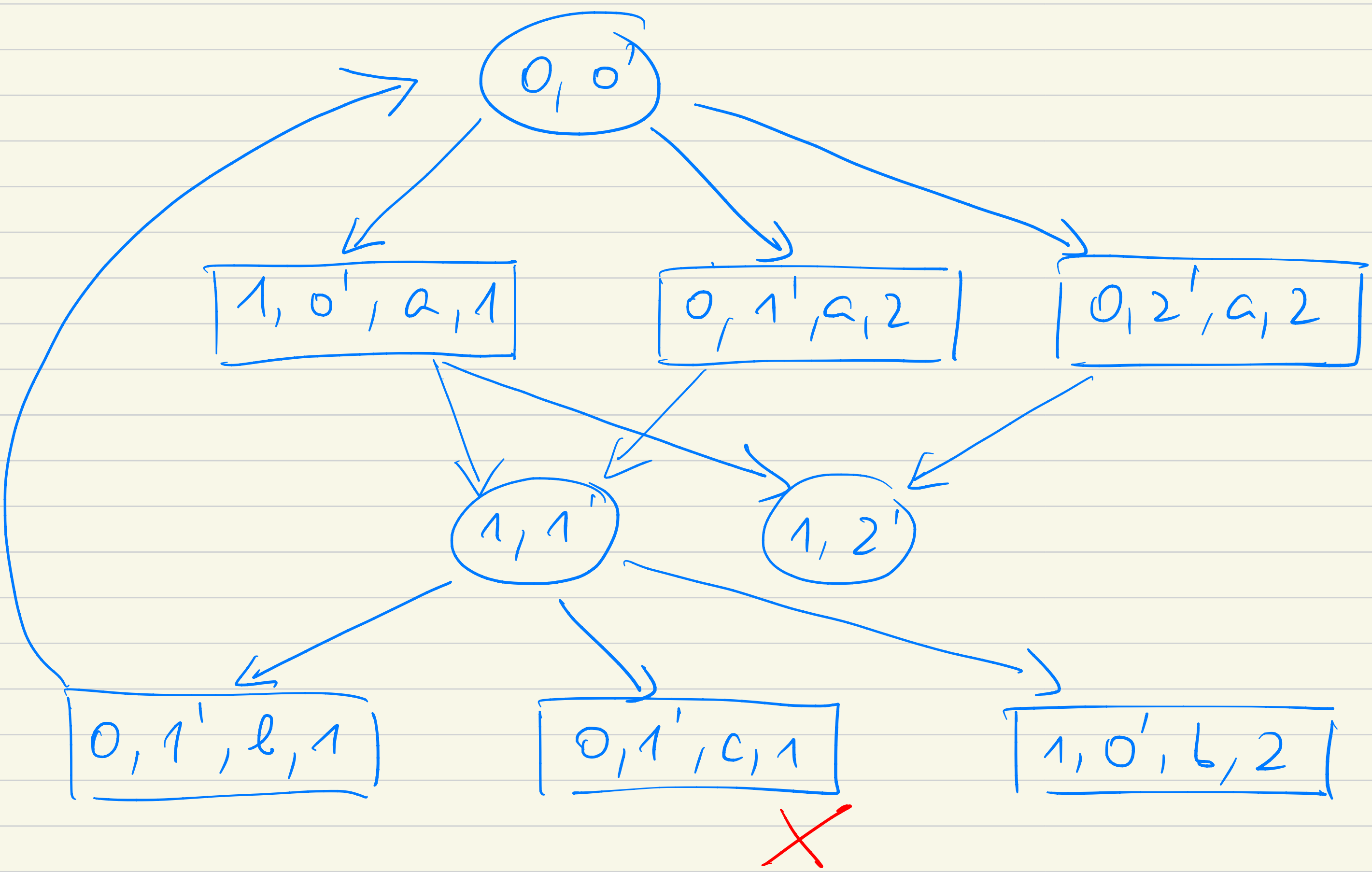
$$f(s) = \begin{cases} (l_1, l_2, x_1 = 0, x_2 = 0) & \text{si } b_1 = b_2 = 0 \\ (l_1, l_2, x_1 = 0, x_2 > 0) & \text{si } b_1 = 0, b_2 > 0 \\ (l_1, l_2, x_1 > 0, x_2 = 0) & \text{si } b_1 > 0, b_2 = 0 \\ (l_1, l_2, x_1 > x_2 > 0) & \text{si } b_1 > b_2 > 0 \\ (l_1, l_2, x_2 > x_1 > 0) & \text{si } b_2 > b_1 > 0 \end{cases}$$





$F_1 = \{0\}$ $\qquad$ $F_2 = \{0'\}$

$$L(T_1) = L(T_2) = (ab + ac)^*$$



Spoiler gagne en essayant d'aller vers $\langle 0, 1', c, 1 \rangle$ ou $\langle 0, 2', b, 1 \rangle$

$\longrightarrow$ jeu d'accessibilité

Ça donne un algo polynomial pour savoir si deux ST sont bisimilaires.

[Il y a bcp d'algorithmes très efficaces pour la bisimulation.

Complexité : $O(|T_1| \cdot |T_2|)$ pour $|T_1|, |T_2| \leq n \rightarrow O(n \log n)$]

Aujourd'hui : comment résoudre des jeux dont la cond. de victoire est une combinaison booléenne de cond. d'accessibilité :

ex P_0 veut visiter F, tout en évitant B. $F, B \subseteq V$.

$$\mathrm{Reach}(F) \cap \mathrm{Avoid}(B)$$

Préparation

Q Etant donnés $F_1, F_2 \subseteq V$

?? $\mathrm{Attr}_0(F_1 \cup F_2) \overset{\supseteq}{=} \mathrm{Attr}_0(F_1) \cup \mathrm{Attr}_0(F_2)$??

F $\quad 1 \leftarrow \boxed{0} \rightarrow 2 \quad$ $F_1 = \{1\}$, $F_2 = \{2\}$

$\mathrm{Attr}_0(F_1) = \{1\}$

$\qquad F_2 = \{2\}$

$\mathrm{Attr}_0(F_1 \cup F_2) = \{0, 1, 2\}$

Rq $\mathrm{Attr}_0(F_1) \cap \mathrm{Attr}_0(F_2) \overset{?}{=} \mathrm{Attr}_0(F_1 \cap F_2)$

$$1 \leftarrow 0 \longrightarrow 2 \qquad F_1 = \{1\}, \quad F_2 = \{2\}$$

$$\mathrm{Attr}_0(F_1) = \{0, 1\}$$
$$\mathrm{Attr}_0(F_2) = \{0, 2\}$$
$$\mathrm{Attr}_0(F_1 \cap F_2) = \emptyset$$

ex $A = (V_0, V_1, E)$ $\quad F_1, F_2 \subseteq V$

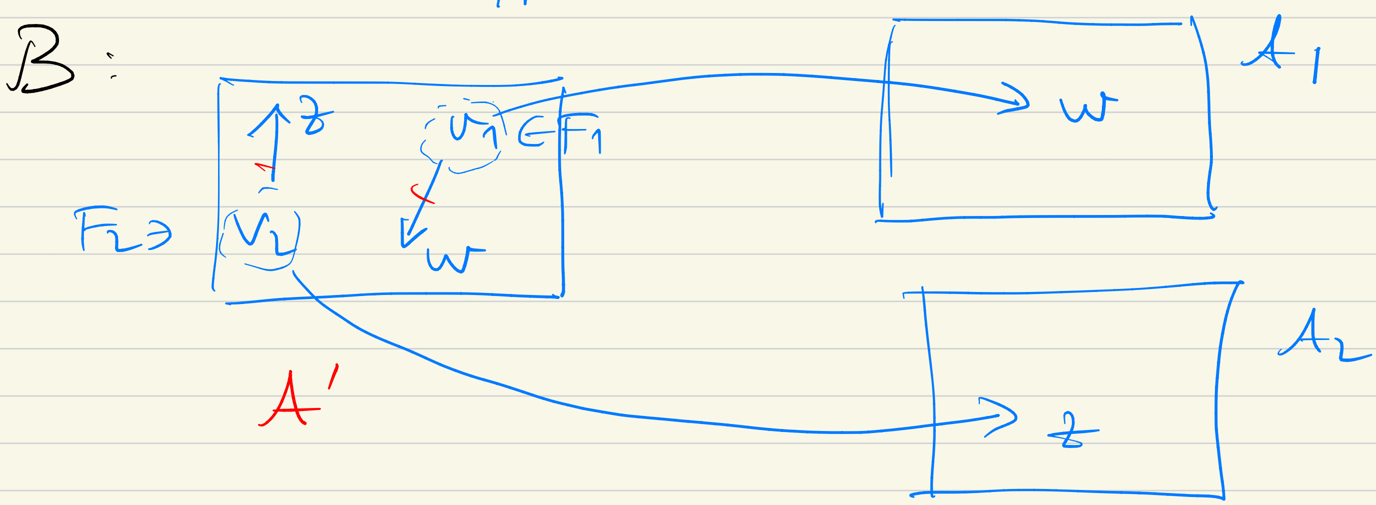
$$\boxed{\mathrm{Reach}(F_1) \cap \mathrm{Reach}(F_2)}$$

Une partie $\pi = (v_0, v_1 \ldots)$ est gagnée par P_0 s'il ex. i, j tq. $v_i \in F_1$, $v_j \in F_2$

solution : construire une arène plus grande sur laquelle on se souvient si on a déjà visité F_1 (ou F_2). Et utiliser un jeu d'accessibilité sur l'arène plus grande.

$\longrightarrow$ réduction de jeu

On suppose : $F_1 \cap F_2 = \emptyset$



A_1, A_2 : copies de A

A' : A où les arcs sortants de $F_1 \cup F_2$ sont redirigés vers A_1, resp. A_2

Sur B on joue jeu d'accessibilité

$$\mathrm{Reach}\big(\underbrace{F_2^{(1)}}_{\text{"}A_1\text{"}} \cup \underbrace{F_1^{(2)}}_{\text{"}A_2\text{"}}\big)$$

B : $V' = V \cup (V \times \{1\}) \cup (V \times \{2\})$

$$E' = \big(E \cap (V \setminus (F_1 \cup F_2)) \times V\big) \cup E^{(1)} \cup E^{(2)} \cup$$

$$\{(u, v^{(1)}) : u \in F_1,\ (u,v) \in E\} \cup$$

$$\{(u, v^{(2)}) : u \in F_2,\ (u,v) \in E\}$$

$X = \mathrm{Attr}_0 \left(F_1^{(2)} \cup F_2^{(1)} \right) \leftarrow$ dans le jeu $\mathcal{B}$

$v \in X \cap V \iff$ dans le jeu original, P_0 gagne à partir de v pour la condition $\mathrm{Reach}(F_1) \cap \mathrm{Reach}(F_2)$

Deux jeux $\mathcal{G}_1$, $\mathcal{G}_2$. $\mathcal{G}_i = (\underbrace{A_i}_{\text{arène}}, \mathrm{Win}_i)$

$f : V(A_1) \to V(A_2)$.

f est <u>réduction de $\mathcal{G}_1$ vers $\mathcal{G}_2$</u> si pour tout $v \in V(A_1)$:

v gagnant pour P_0 par rapport à Win_1 $\iff$

$f(v)$ gagnant pour P_0 par rapport à Win_2

Dans l'exemple :

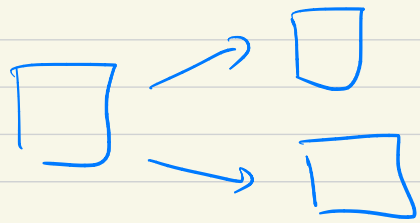
A_1 = arène initiale A

σ_0, σ_1

$\uparrow$

σ_0', σ_1'

A_2 =



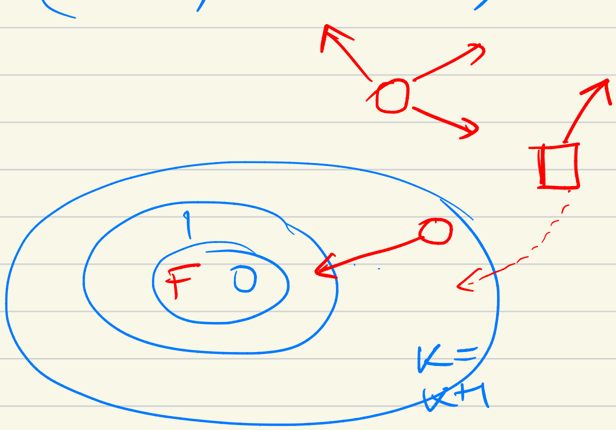
$$f(v) = v$$

$v \in V(A)$ dans A'

En résolvant le jeu (A_2, Win_2) on obtient une stratégie gagnante pour P_0, (à partir des sommets gagnants)

$(A_2, Win_2) \longrightarrow$ W_0', W_1' } régions gagnantes

σ_0', σ_1' stratégies gagnantes



σ_0' : strat. d'attracteur.

σ_1' : strat. de piège

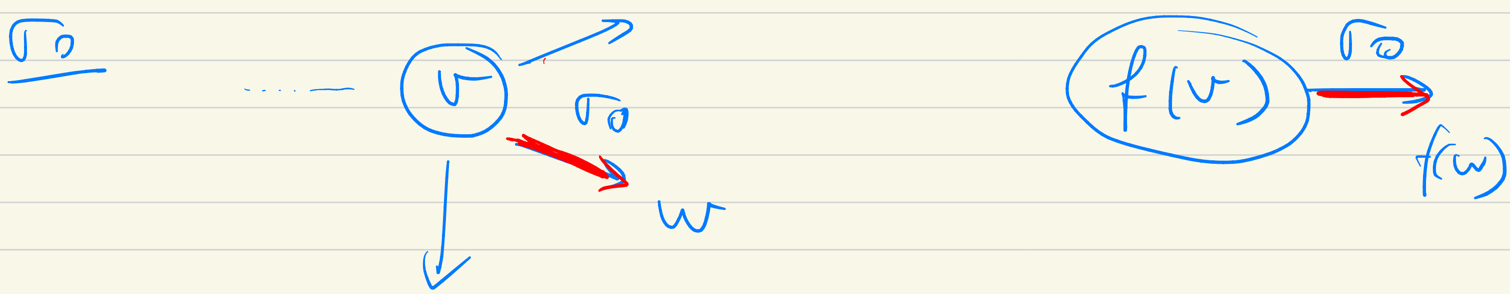
$$(A_1, Win_1) \xrightarrow{f} (A_2, Win_2)$$

$$W_0, W_1 \longleftarrow W_0', W_1'$$

$$\sigma_0, \sigma_1 \qquad \sigma_0', \sigma_1'$$

$$W_0 = \{ v \in V(A_1) : f(v) \in W_0' \}$$

$$W_1 = \{ v \in V(A_1) : f(v) \in W_1' \}$$



$\sigma_0(v) = w$ di $\sigma_0'(f(v)) = f(w)$

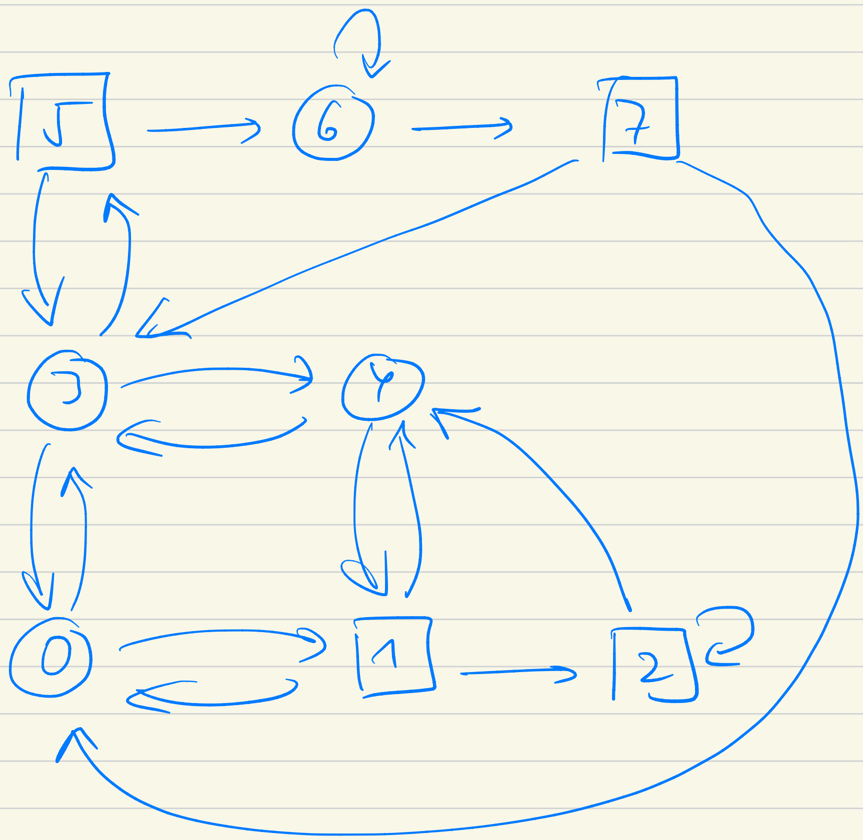
Reach(3,4) $\wedge$ Reach(0,6)

$\downarrow$

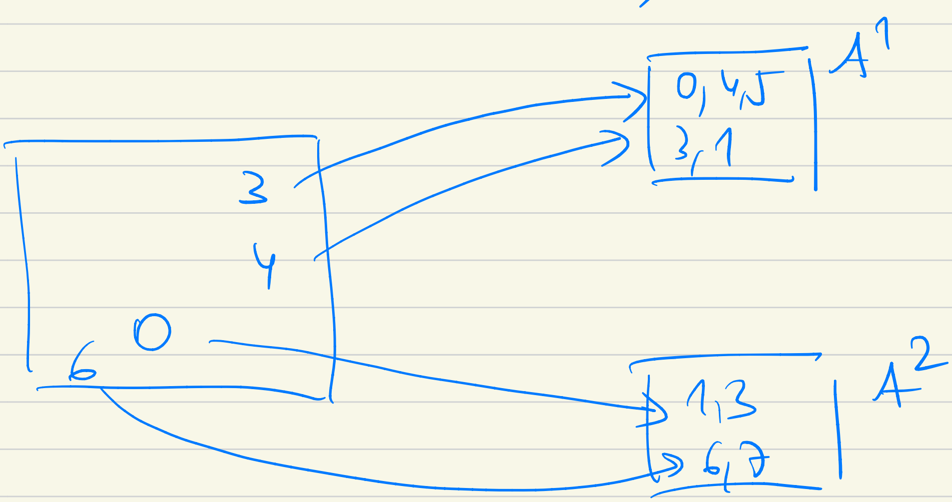
σ_0 utilise de la mémoire

$7 \in W_0$

A



Reach({3,4}) $\cap$ Reach({0,6})



$A^1 \cup A^1 \cup A^2$

B

$\sigma_0'(\overset{A^1}{(3,1)}) = (0,1)$

Dès que P_0 atteint 3

→ état de mémoire 1

$\sigma_0(3,1) = 0$

<u>Rq</u> : dans le jeu $\mathcal{B}$: P_0 a stratégie sans mémoire (strat. d'attracteur). σ_0'

<u>σ_0'</u> $\longrightarrow$ σ_0 est une strat. gagnante avec mémoire

$$\sigma_0 : V_0 \times \{0, 1, 2\} \longrightarrow V$$

0 : ni F_1, ni F_2

1 : F_1, pas F_2

2 : F_2, pas F_1

$$\sigma_0'(\underbrace{v}_{\in A'}) = (w, 1)$$

$$\sigma_0(v, 0) = w$$

0 : mémoire

update $0 \longrightarrow 1$

Stratégies (pour P_0) :

$$\sigma_0 : \underbrace{V^*}_{\text{historique}} V_0 \to V$$

Stratégies sans mémoire :

$$\sigma_0 : V_0 \to V$$

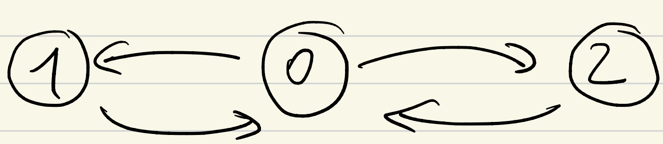
Stratégies avec mémoire finie M

$$\sigma_0 : V_0 \times M \to V$$

$$\text{update} : V_0 \times M \to M$$

<u>Q</u> Est-ce que P_0 a besoin de mémoire pour gagner un jeu $\text{Reach}(F_1) \cap \text{Reach}(F_2)$?

$\boxed{\text{Oui}}$:



$F_1 = \{1\}$

$F_2 = \{2\}$

mémoire $0, 1, 2$

$\sigma_0(0,0) = 1$; $upd(0,0) = 1$

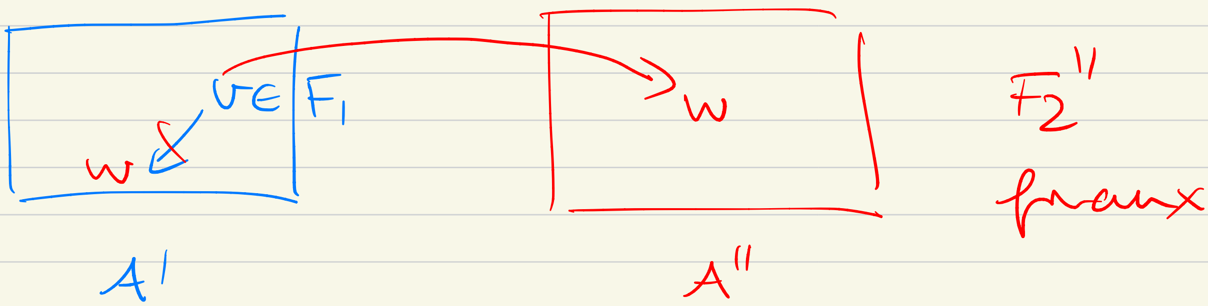
$\sigma_0(0,1) = 2$; $upd(0,1) = 2$

$F_1 \cap F_2 = \emptyset$

$\boxed{\text{Win : voir } F_1 \text{, et ensuite } F_2}$

$\pi = v_0, v_1, \ldots$ est gagnante pour P_0 si il ex. $i < j$ tq.

$v_i \in F_1$, $v_j \in F_2$



P_0 a toujours besoin de mémoire

Mais : est-ce vrai aussi pour P_1 ?