

18/20

DM Intro à la Verif

Dimitri Periphanos

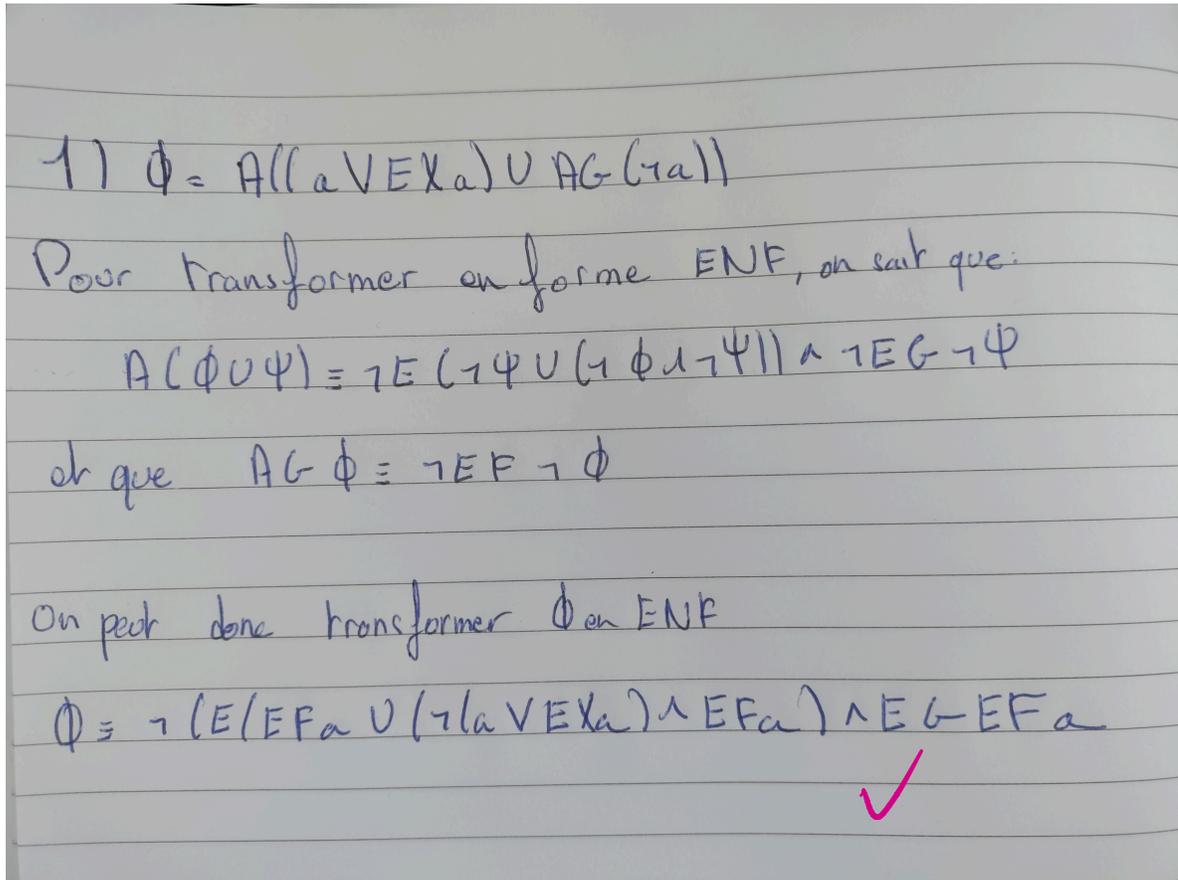
2 avril 2022

Résumé

Ce Dm sera rédigé un peu de manière bâtarde entre le \LaTeX et des photos de rédactions manuscrites, j'ai fait la plupart des exercices sur papier mais je pense que vous préférez avoir un joli PDF avec tout bien organisé plutôt qu'un .zip avec pleins de photos de mes cahiers

1 Exercice 1

1.1 Forme ENF



1.2 Calculer SAT

Pour calculer $Sat(\phi)$, on va découper ϕ en de multiples ϕ_i et calculer $Sat(\phi_i)$ pour tout i .

$$\Phi_1 = EFa \quad \text{Sat}(EFa) = \{s_0, s_1, s_2, s_5\}$$

$$= \{q \in ST \mid \text{post}^*(q) \cap \text{Sat}(a) \neq \emptyset\}$$

$$\Phi_2 = EG EFa \quad \text{Sat}(\Phi_2) = \{s_0, s_1, s_2, s_5\}$$

On restreint le ST aux états qui satisfont EFa et on calcule les SCC non triviales (ici $\{s_0, s_1\}$ et $\{s_2, s_5\}$) ce qui nous donne ce résultat ✓

$$\Phi_3 = \neg EG EFa \quad \text{Sat}(\neg EG EFa) = \{s_3, s_4\}$$

$$\Phi_4 = EXa \quad \text{Sat}(EXa) = \{s_1, s_2\}$$

$$= \{q \in ST \mid \text{post}(q) \cap \text{Sat}(a) \neq \emptyset\}$$

$$\Phi_5 = a \vee EXa \quad \text{Sat}(a \vee EXa) = \text{Sat}(a) \cup \text{Sat}(\Phi_4)$$

$$= \{s_0, s_1, s_2, s_5\}$$

$$\Phi_6 = \neg \Phi_5 \quad \text{Sat}(\Phi_6) = \{s_3, s_4\}$$

$$\Phi_7 = EFa \wedge \Phi_6 \quad \text{Sat}(\Phi_7) = \text{Sat}(\Phi_1) \cap \text{Sat}(\Phi_6) = \emptyset$$

$$\Phi_8 = EFa \cup \Phi_7 \quad \text{Sat}(\Phi_8) = \emptyset$$

En effet, aucun état ne peut satisfaire $a \cup B$ si aucun état ne peut satisfaire B

EF

$$\Phi_9 = E(\Phi_8) \quad \text{Sat}(\Phi_9) = \emptyset = \text{Pre}^*(\text{Sat}(\Phi_8))$$

$$\Phi_{10} = \neg \Phi_9 \quad \text{Sat}(\Phi_{10}) = \{s_0, s_1, s_2, s_3, s_4, s_5\}$$

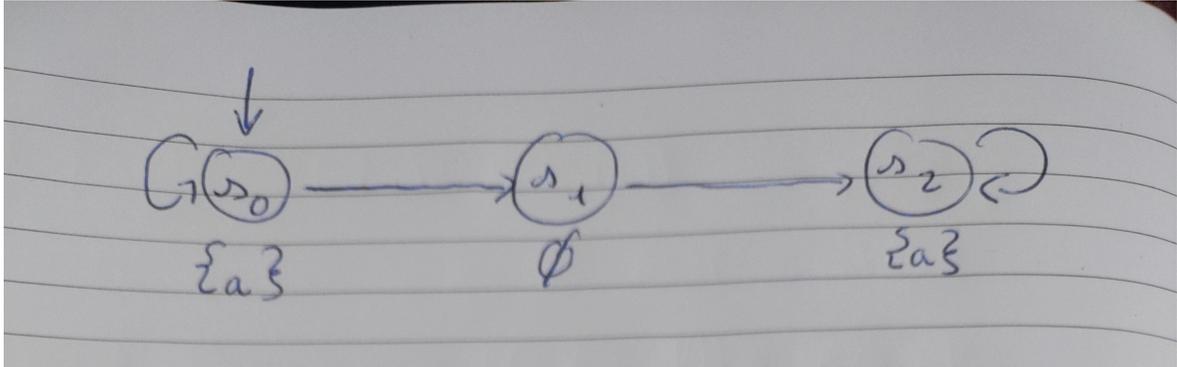
$$\Phi = \Phi_{10} \wedge \Phi_3 \quad \text{Sat}(\Phi) = \text{Sat}(\Phi_{10}) \cap \text{Sat}(\Phi_3) = \{s_3, s_4\}$$

Donc ST $\models \Phi$ ✓

2 Exercice 2

2.1 AF AXa et FXa

Soit le système de transition S suivant :



On remarque bien que $S \models \mathbf{FX}a$ car pour toute exécution partant de s_0 , $s_0 \models \mathbf{FX}a$. Cependant $S \not\models \mathbf{AF AX}a$ car si l'on prends $\pi = s_0, s_0, s_0, \dots$. Alors, comme $s_0 \not\models \mathbf{AX}a$ (car $s_1 \not\models a$), on a bien un chemin pour lequel la condition n'est pas vraie ce qui suffit à contre dire le connecteur universel.

Donc $S \not\models \mathbf{AF AX}a$

Donc $\mathbf{AF AX}a \neq \mathbf{FX}a$

2.2 AF(a ∧ AFb) et F(a ∧ Fb)

2) On cherche à prouver

$$s_0 \models \mathbf{AF}(a \wedge \mathbf{AF}b) \Leftrightarrow s_0 \models \mathbf{F}(a \wedge \mathbf{F}b)$$

i) \Rightarrow

Supposons un système de transition S tq $S \models \mathbf{AF}(a \wedge \mathbf{AF}b)$

Alors pour toute exécution à partir de s_0 ,

$$\exists 0 \leq i \text{ tq } s_i \models a \wedge \mathbf{AF}b$$

~~et $\forall j > i$ $s_j \not\models a \wedge \mathbf{AF}b$ aussi~~

$$\Rightarrow s_i \models a \text{ et } s_i \models \mathbf{AF}b$$

Alors pour toute exécution à partir de s_i ,

$$\exists 0 \leq k \text{ tq } s_{i+k} \models b$$

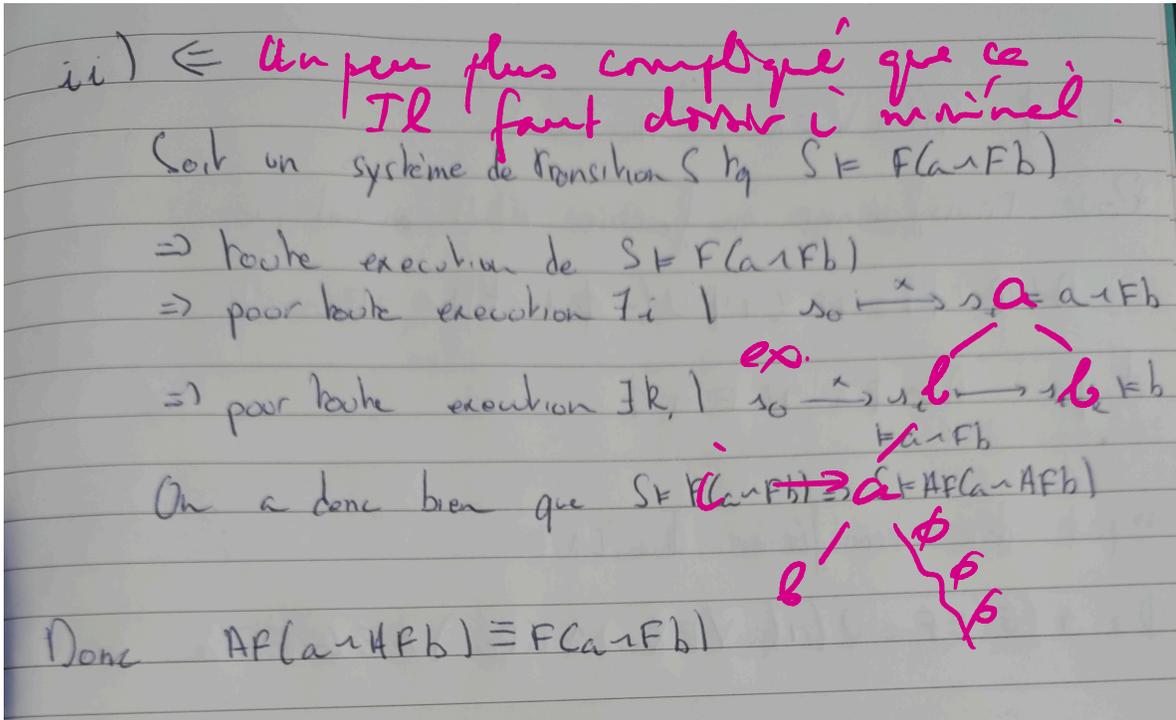
~~et $\forall l > k$ $s_{i+l} \not\models b$ aussi~~

\Rightarrow A partir de s_0 on peut trouver pour toute exécution deux entiers (i, k) tel que

$$s_0 \xrightarrow{x} s_i \xrightarrow{y} s_{i+k} \models b$$

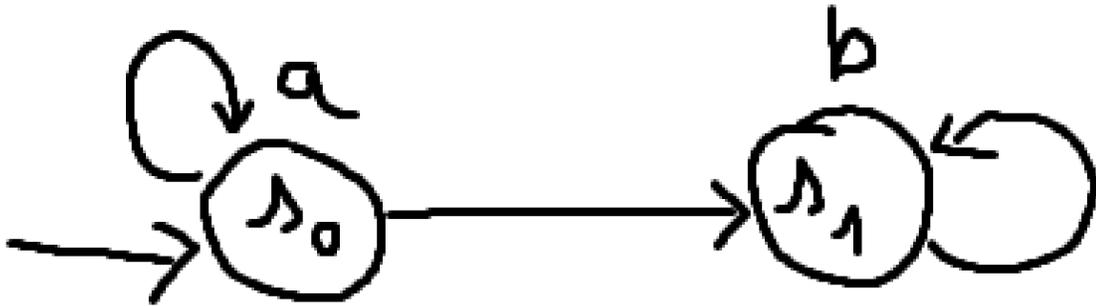
$\models a \wedge \mathbf{F}b$

$$\Rightarrow S \models \mathbf{F}(a \wedge \mathbf{F}b)$$



2.3 $A((EXa) \cup (AGb))$ et $(Xa) \cup (Gb)$

Soit le système de transition S suivant :



On remarque bien que $S \models A((EXa) \cup (AGb))$ car pour toute exécution partant de s_0 ,

$s_0 \models EXa$, grâce au chemin s_0, s_0, s_0, \dots

$s_1 \models AGb$ peu importe le nombre de fois où l'on boucle sur s_1 .

En revanche $S \not\models (Xa) \cup (Gb)$ car $s_0 \not\models Xa$ (car $s_1 \not\models a$) et $s_0 \not\models Gb$ (car $s_0 \not\models b$). ✓

2.4 Ce qu'on peut en déduire

On remarque que pour ces 3 formules, la formule LTL est la même que la formule CTL privée de ces quantificateurs sur les chemins (A et E).

Ce qui nous permet d'affirmer qu'il n'existe pas d'équivalent LTL aux formules CTL 1 et 3.

En effet, on sait que soit ϕ_{CTL} , alors si on nomme ϕ_{LTL} la formule LTL obtenue en supprimant tout les A et le E de ϕ_{CTL} .

Alors on sait que soit $\phi_{CTL} \equiv \phi_{LTL}$ soit il n'existe pas de formule ϕ_{LTL} équivalente à ϕ_{CTL}



3 Exercice 3

3.1 $E(\phi \cup \psi) \equiv \psi \vee (\phi \wedge EXE(\phi \cup \psi))$

Il est assez facile de se convaincre que cette équivalence est vraie, en effet, soit on est dans un état feuille et ψ est valide soit non, alors ϕ est valide et il existe un descendant qui valide $E(\phi \cup \psi)$

Plus formellement on peut prouver la double implication de la sorte :

The handwritten proof on lined paper shows the following steps:

$$E(\phi \cup \psi) \equiv \psi \vee (\phi \wedge EXE(\phi \cup \psi))$$
$$M, s_0 \models E(\phi \cup \psi) \Leftrightarrow s_0 \models \psi \vee (\phi \wedge EXE(\phi \cup \psi))$$

(\Rightarrow)

$$s_0 \models E(\phi \cup \psi)$$
$$\Rightarrow \exists \pi : s_0 \rightarrow s_1 \rightarrow \dots \rightarrow s_n \text{ tq } \pi \models \phi \cup \psi$$
$$\Rightarrow \exists 0 \leq i \leq n \text{ tq } s_i \models \psi$$
$$\text{et } \forall k < i \text{ } s_k \models \phi$$

- Soit $i = 0$
Alors $s_0 \models \psi$
- Soit $i \geq 1$
Alors $s_0 \models \phi$ et $s_1 \models E(\phi \cup \psi)$
 $\Rightarrow s_0 \models \phi \wedge EXE(\phi \cup \psi)$

$$\Rightarrow s_0 \models \psi \vee (\phi \wedge EXE(\phi \cup \psi)) \quad \checkmark$$

$$\Leftrightarrow s_0 \models \psi \vee (\phi \wedge \text{EXE}(\phi \cup \psi))$$

• Soit $s_0 \models \psi$
 $\Rightarrow \forall$ chemin π partant de s_0
 $\pi \models \phi \cup \psi$

$$\Rightarrow s_0 \models \text{E}(\phi \cup \psi) \quad \checkmark$$

• Soit $s_0 \models \phi \wedge \text{EXE}(\phi \cup \psi)$

$$\Rightarrow s_0 \models \phi \text{ et } s_0 \models \text{EXE}(\phi \cup \psi)$$

Alors $\exists \pi : s_0 \rightarrow s_1 \dots \rightarrow s_n$ tq $\pi \models \text{XE}(\phi \cup \psi)$

$$\Rightarrow s_1 \models \text{E}(\phi \cup \psi)$$

$\Rightarrow \exists i \geq 1$ tq $\pi[i] \models \psi$ et $\forall k < i \pi[k] \models \phi$

Or on sait que $\pi[0] \models \phi$
 Donc $\exists i > 0$ tq $\pi[i] \models \psi$ et $\forall k < i \pi[k] \models \phi$

$$\text{donc } s_0 \models \text{E}(\phi \cup \psi) \quad \checkmark$$

3.2 $\mathbf{A}(\phi \cup \psi) \equiv \psi \vee (\phi \wedge \mathbf{AXA}(\phi \cup \psi))$

La preuve est identique à la preuve au dessus si ce n'est qu'il faut remplacer tout les

$$\exists \pi : s_0, s_1, \dots$$

par

$$\forall \pi : s_0, s_1, \dots$$

En effet il faut s'assurer que TOUT les s_1 possible jugent $\mathbf{A}(\phi \cup \psi)$

3.3 $\mathbf{E}(\phi \mathbf{R} \psi) \equiv \phi \wedge (\psi \vee \mathbf{EXE}(\phi \mathbf{R} \psi))$

$$\mathbf{E}(\phi \mathbf{R} \psi) \equiv \phi \wedge (\psi \vee \mathbf{EXE}(\phi \mathbf{R} \psi)) \quad \checkmark$$

$$M_{\phi} \quad s_0 \models \mathbf{E}(\phi \mathbf{R} \psi) \Leftrightarrow s_0 \models \phi \wedge (\psi \vee \mathbf{EXE}(\phi \mathbf{R} \psi))$$

① \Rightarrow Soit S un système de transition tq
$$S \models \mathbf{E}(\phi \mathbf{R} \psi)$$

$\Rightarrow \exists \pi : s_0, \dots$ un chemin partant de s_0 tq

$\pi \models \phi \mathbf{R} \psi \Rightarrow$ $\alpha)$ Soit que $\exists i, \pi[i] \models \phi$ et $\forall j \leq i, \pi[j] \models \psi$ \checkmark

$\beta)$ Soit que $\forall k, \pi[k] \models \psi$ \checkmark

$\alpha)$ Soit $i=0$ alors $s_0 \models \phi \wedge \psi$ et $s_0 \models \mathbf{A}$

$\Rightarrow s_0 \models \phi \wedge (\psi \vee \mathbf{EXE}(\phi \mathbf{R} \psi)) \quad \checkmark$

Soit $i \geq 1$ alors $s_1 \models \mathbf{E}(\phi \mathbf{R} \psi) \Rightarrow$
 $\Rightarrow s_0 \models \mathbf{EXE}(\phi \mathbf{R} \psi)$
et $s_0 \models \psi$ car $\forall j \leq i, s_j \models \psi$

$\Rightarrow s_0 \models \psi \wedge (\phi \vee \mathbf{EXE}(\phi \mathbf{R} \psi)) \quad \checkmark$

$\beta)$ $s_0 \not\models \phi$ et $s_1 \models \mathbf{E}(\phi \mathbf{R} \psi)$ car ϕ hoch le temps vra.

$\Rightarrow s_0 \models \phi \wedge (\psi \vee \mathbf{EXE}(\phi \mathbf{R} \psi)) \quad \checkmark$

ii) \Leftarrow Soit S un système de transition tq

$$S \models \Psi \wedge (\Phi \vee \exists x E(\Phi R \Psi))$$

$$\Rightarrow \alpha) \text{ Soit } S \models \Psi \wedge \Phi$$

$$\beta) \text{ Soit } S \models \Psi \wedge \exists x E(\Phi R \Psi)$$

$$\alpha) S \models \Psi \wedge \Phi \Rightarrow \forall \pi: s_0, s_1, \dots \quad \pi \models \Phi R \Psi$$

$$\Rightarrow s_0 \models E(\Phi R \Psi) \quad \checkmark$$

$$\beta) \Rightarrow s_0 \models \Psi \quad \text{et} \quad \exists \pi: s_0, s_1, \dots \text{ tq } \pi[1] \models E(\Phi R \Psi)$$

$$\Rightarrow \exists i \geq 1 \text{ tq } \pi[i] \models \Phi \quad \text{et} \quad \forall j < i \quad \pi[j] \models \Psi$$

ou $\forall R \geq 1 \quad \pi[R] \models \Psi$

Or on sait que $\pi[0] \models \Psi$ Donc

$$\text{Soit } \exists i \geq 0 \text{ tq } \pi[i] \models \Phi \quad \text{et} \quad \forall j \leq i \quad \pi[j] \models \Psi$$

ou $\forall R \geq 0 \quad \pi[R] \models \Psi$

$$\Rightarrow s_0 \models E(\Phi R \Psi) \quad \checkmark$$

4 Exercice 4

4.1 $Sat(E(aRb))$

1. Pour calculer $SAT(E(aRb))$. On construira un ensemble d'état feuille, puis on ajoutera itérativement tout les états qui permettent d'accéder à ces feuilles.

Plus formellement:

$$SAT(E(aRb)) \subseteq SAT(b)$$

Soit $T \subseteq S$ l'ensemble d'états s tq $s \models E(aRb)$ à la fin de l'exécution

On commence par initialiser $T = \emptyset$ il faut enlever des états $T = SAT(b)$

Puis \forall états $q \in S$ si $q \models a \wedge b$ alors on ajoute q à T : $T = T \cup q$

L'ajout des feuilles est terminé donc maintenant on peut commencer le processus itératif, à chaque itération on a

$$T = T \cup \{q \in S \mid q \models b \text{ et } Post(q) \cap T \neq \emptyset\}$$

Donc si $q \models b$, et qu'il a un descendant qui valide $E(aRb)$ alors il le valide aussi.

Le processus itératif se termine quand T n'évolue plus d'une itération à l'autre

Après relecture je me rends compte que mon algorithme ne me permet pas d'ajouter à T , les états qui permettent des chemins où a est tout le temps faux et b tout le temps vrai, peut-être que les feuilles devraient être les états où b est vrai peu importe a , mais je ne suis pas sûr de moi, j'ai peur que cette simple modification ne soit pas suffisante...

4.2 $Sat(A(aUb))$

Pour calculer $Sat(A(aUb))$. On construira aussi un ensemble de d'états feuille, puis on ajoutera itérativement les états dont TOUT les descendants permettent d'accéder à ses feuilles.

On commence par initialiser $T = \emptyset$

Puis on y ajoute les feuilles

$$T = T \cup \{q \in S \mid q = b\}$$

Maintenant on commence le processus itératif, à chaque étape on a

$$T = T \cup \{q \in S \mid q \neq a \text{ et } \text{Post}(q) \subseteq T\}$$

en effet on s'assure ainsi que TOUT les descendant de q soient soit des feuilles soit d'autres état qui satisfont $A(a \cup b)$

Comme pour le dernier algorithme le processus s'arrête quand T reste inchangé entre deux itérations

5 Exercice 5

5.1 $S_2 \models \phi \implies S_1 \models \phi$

1) Supposons S_1 et S_2 deux systèmes de transitions
avec $S_1 \subseteq S_2$

Supposons maintenant que $S_2 \models \phi$

Alors, soit ϕ est formule d'état (i) soit ϕ est une formule de chemin (ii) \rightarrow ind. sur la structure de la formule : \exists, \forall, \dots

(i) Si ϕ est une formule d'état et que $S_2 \models \phi$ alors comme on a AX, AU, AG
La restriction de L_2 à S_1 et $I_1 = I_2$ on sait que les labels des états initiaux des deux automates seront les mêmes et donc que $S_1 \models \phi$

(ii) Si ϕ est une formule de chemin alors : (car la grammaire ne permet pas le quantificateur existentiel)
 $\forall \pi_2: I_{2,i} s_1 \dots s_n \quad \pi_2 \models \phi$

Or comme ? $S_1 \subseteq S_2$ et $\rightarrow_1 \subseteq \rightarrow_2$ On peut déduire que
 $\forall \pi_1: I_{1,i} s_1 \dots s_n$ on a $\pi_1 \subseteq \pi_2$

et donc que $\pi_1 \models \phi$

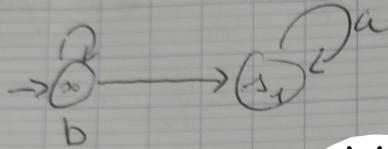
Donc on a bien $\forall \phi \quad S_2 \models \phi \implies S_1 \models \phi$ si $S_1 \subseteq S_2$

(✓)

5.2 Séparation CTL/ACTL

2) Prenons $\phi_{act} = \text{EXEG}a$ et supposons qu'il existe une formule ACTL équivalente

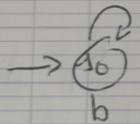
Soit S_2 le ST suivant :



On peut voir que $S_2 \models \text{EXEG}a$ ✓

$\Rightarrow S_2 \models \phi_{act}$ (la formule ACTL dont on suppose l'existence)

Soit S_1 le ST suivant :



On a bien $S_1 \subseteq S_2$ ✓

On sait donc que $\forall \phi, S_2 \models \phi \Rightarrow S_1 \models \phi$

$\Rightarrow S_1 \models \phi_{act} \Rightarrow S_1 \models \text{EXEG}a$

Or aucune exécution ne permet de finir avec que des a
On a donc une contradiction et il n'y a pas de ϕ_{act}
équivalente à $\text{EXEG}a$ ✓