18/20

DM Intro à la Verif

Dimitri Periphanos

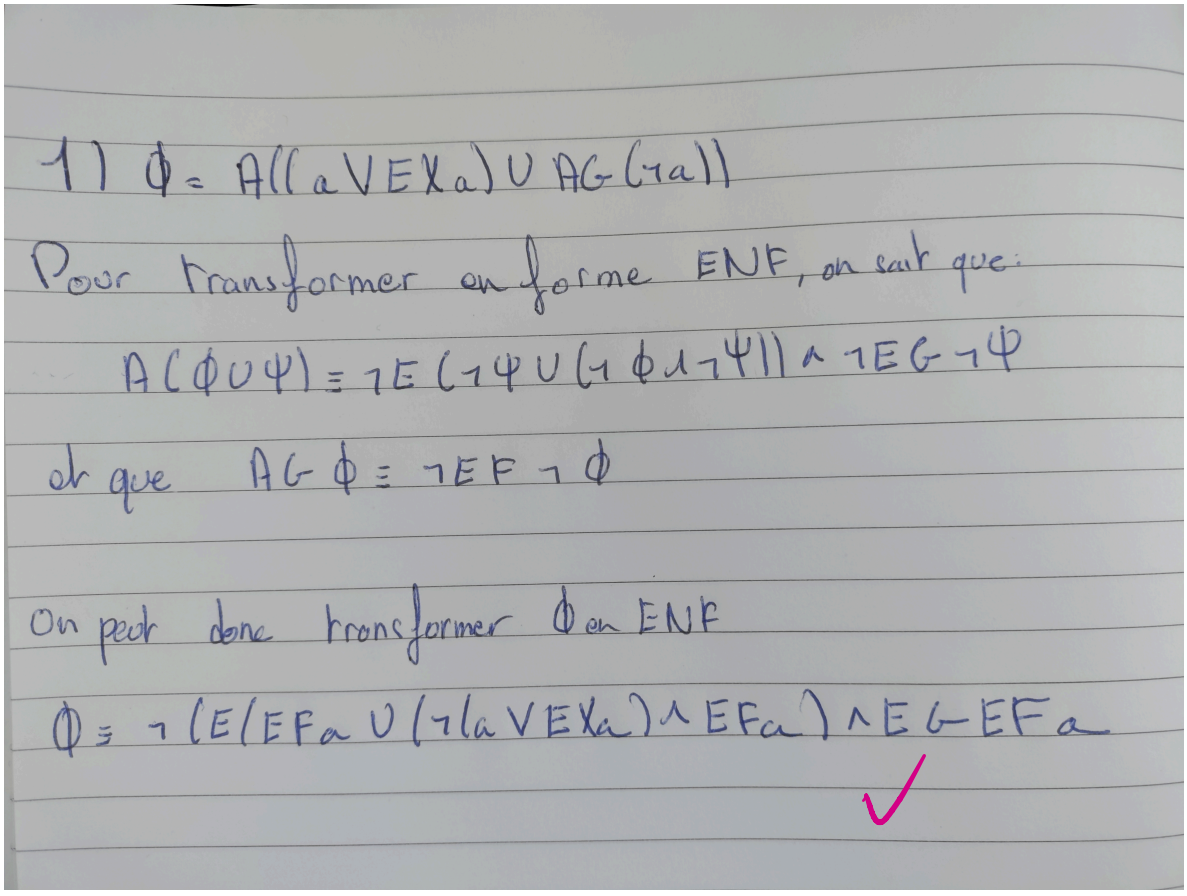
2 avril 2022

Résumé

Ce Dm sera rédigé un peu de manière bâtarde entre le LaTeXet 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.

$\phi_1 = EFa$ $\quad Sat(EFa) = \{s_0, s_1, s_2, s_5\}$

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

$\phi_2 = EGEFa$ $\quad Sat(\phi_2) = \{s_0, s_1, s_2, s_5\}$

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 ✓

$\phi_3 = \neg EGEFa$ $\quad Sat(\neg EGEFa) = \{s_3, s_4\}$

$\phi_4 = EXa$ $\quad Sat(EXa) = \{s_1, s_2\}$

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

$\phi_5 = a \vee EXa$ $\quad Sat(a \vee EXa) = Sat(a) \cup Sat(\phi_3)$

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

$\phi_6 = \neg \phi_5$ $\quad Sat(\phi_6) = \{s_3, s_4\}$

$\phi_7 = EFa \wedge \phi_6$ $\quad Sat(\phi_7) = Sat(\phi_1) \cap Sat(\phi_6) = \emptyset$

$\phi_8 = EFa \, U \, \phi_7$ $\quad Sat(\phi_8) = \emptyset$ ✓

En effet, aucun état ne peut satisfaire $\alpha U \beta$ si aucun état ne peut satisfaire β

EF

$\phi_9 = E(\phi_8)$ $\quad Sat(\phi_9) = \emptyset = Pre^*(Sat(\phi_8))$

$\phi_{10} = \neg \phi_9$ $\quad Sat(\phi_{10}) = \{s_0, s_1, s_2, s_3, s_4, s_5\}$

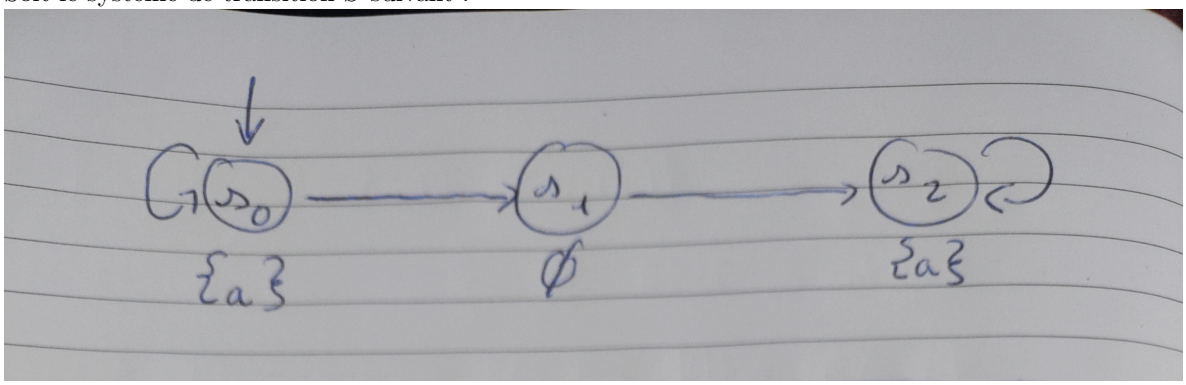
$\phi = \phi_{10} \wedge \phi_3$ $\quad Sat(\phi) = Sat(\phi_{10}) \cap Sat(\phi_3) = \{s_3, s_4\}$ ✓

Donc $ST \neq \emptyset$

2 Exercice 2

2.1 AF AXa et FXa

Soit le système de transition S suivant :

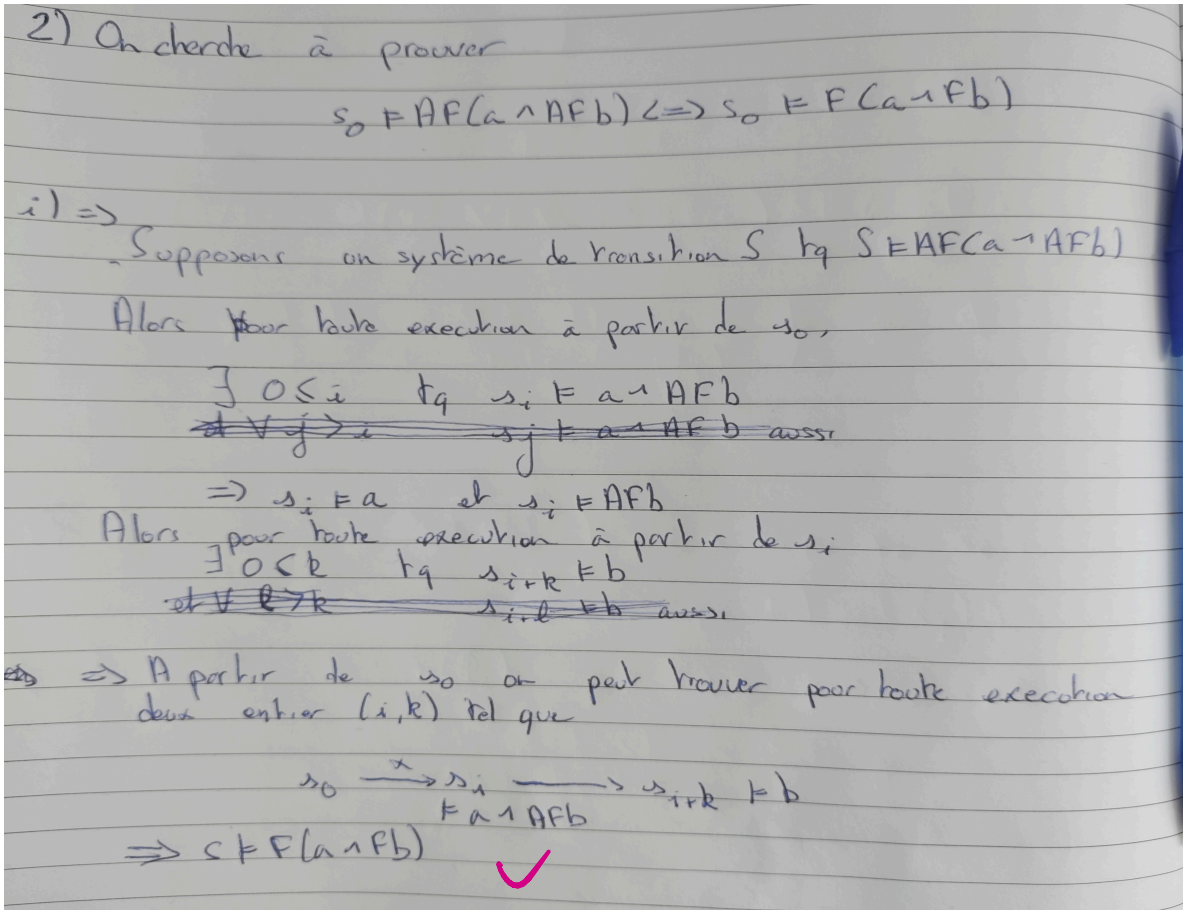


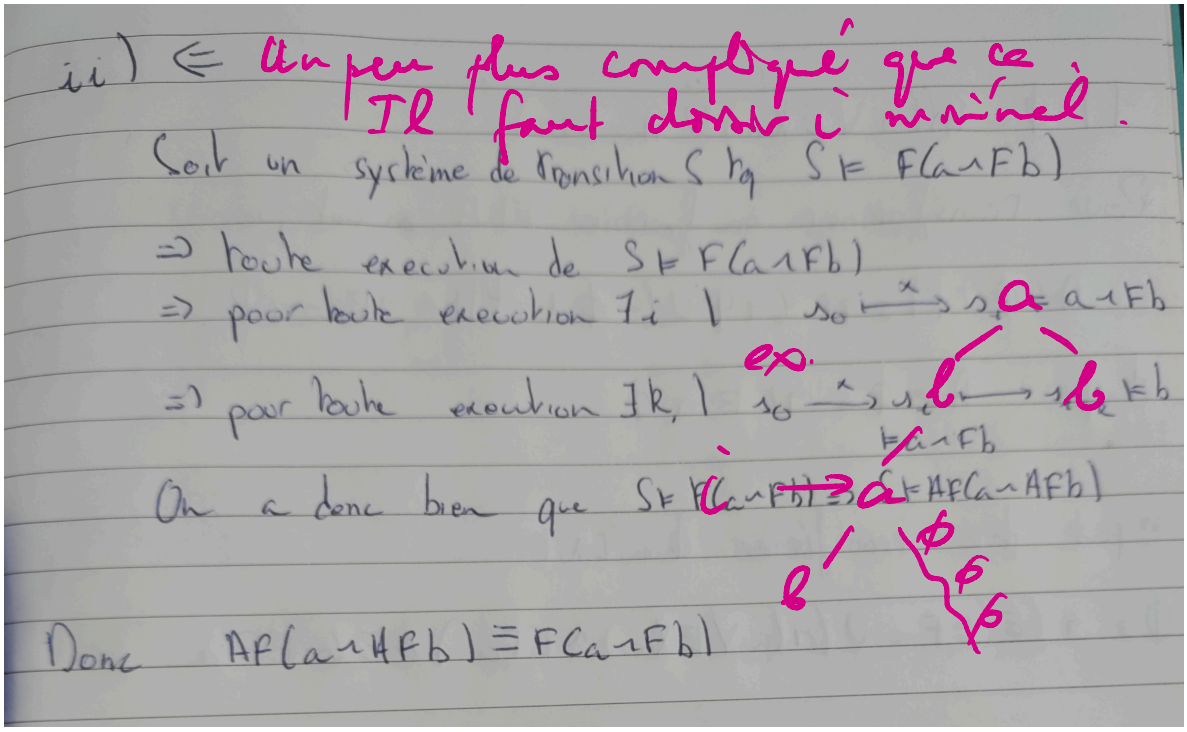
On remarque bien que $S \models \mathbf{FX}a$ car pour toute éxécution partant de s_0, $s_0 \models \mathbf{FX}a$ Cependant $S \not\models \mathbf{AF}\ \mathbf{AX}a$ car si l'on prends $\pi = s_0, s_0, s_0, ...$ Alors, comme $s_0 \not\models \mathbf{AX}a$ (car $s_1 \not\models a$), on a bien un chemin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

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

Donc $\mathbf{AF}\ \mathbf{AX}a \not\equiv \mathbf{FX}a$

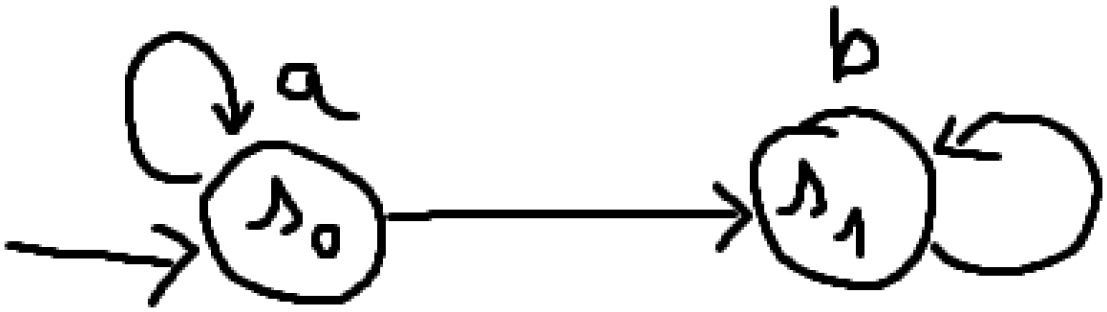
2.2 AF$(a \wedge$ AF$b)$ et F$(a \wedge$ F$b)$





2.3 $\mathbf{A}((\mathbf{EX}a) \cup (\mathbf{AG}b))$ et $(\mathbf{X}a) \cup (\mathbf{G}b)$

Soit le système de transition S suivant :



On remarque bien que $S \models \mathbf{A}((\mathbf{EX}a) \cup (\mathbf{AG}b))$ car pour toute éxécution partant de s_0,
$s_0 \models \mathbf{EX}a$,grâce au chemin s_0, s_0, s_0....
$s_1 \models \mathbf{AG}b$ peut importe le nombre de fois où l'on boucle sur s_1.
En revanche $S \not\models (\mathbf{X}a) \cup (\mathbf{G}b)$ car $s_0 \not\models \mathbf{X}a$(car $s_1 \not\models a$) et $s_0 \not\models \mathbf{G}b$ (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}

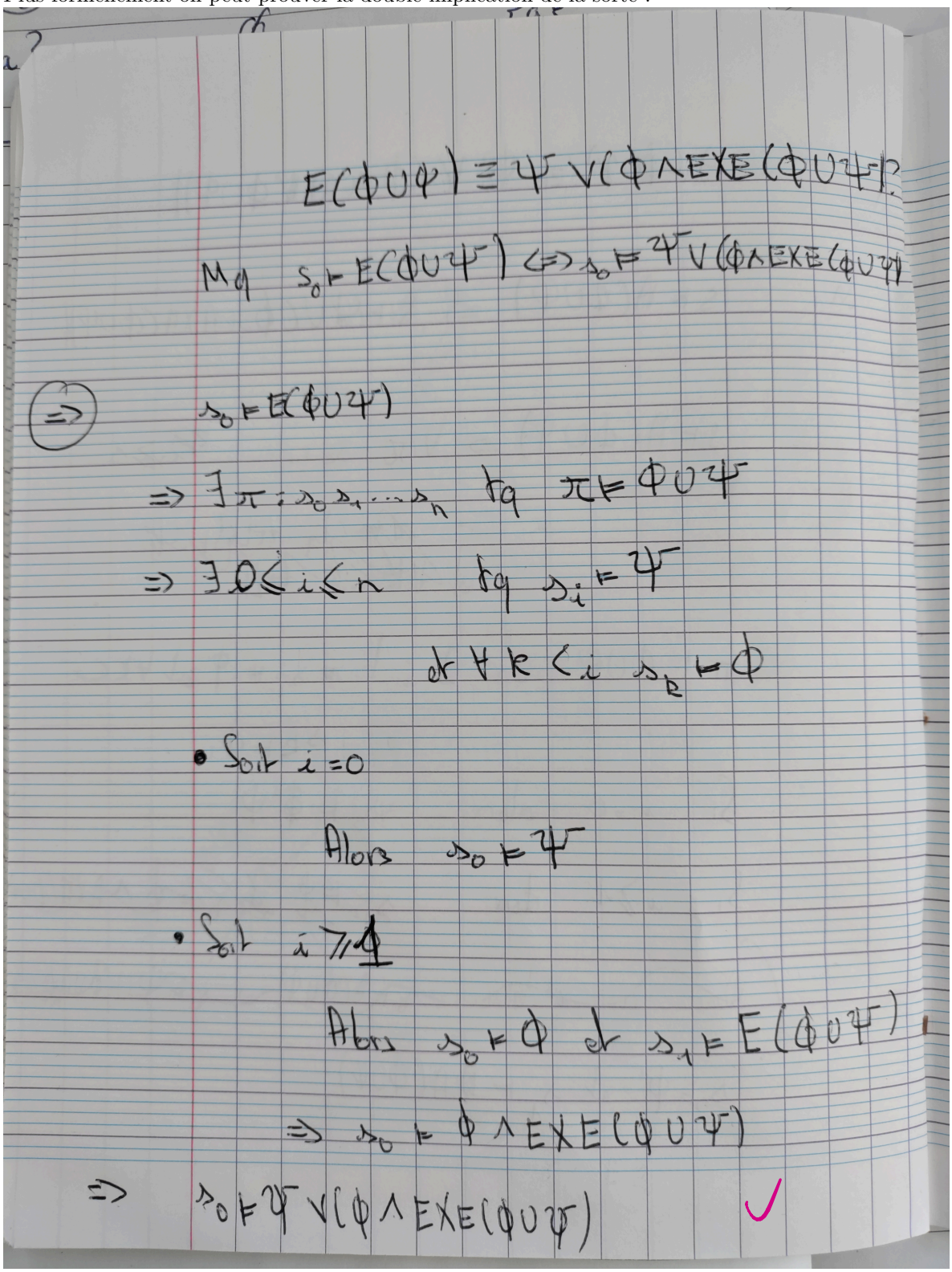
✓

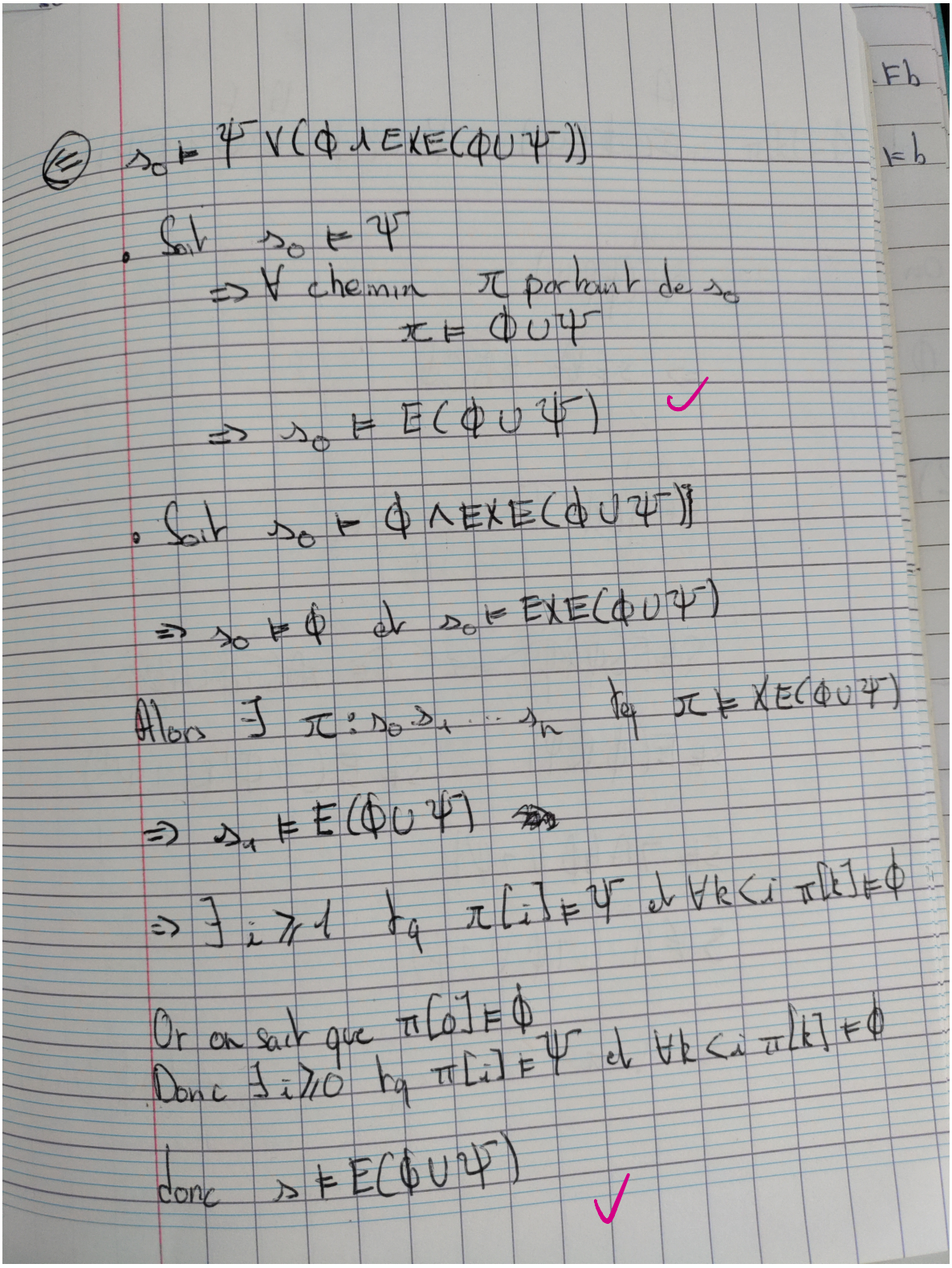
3 Exercice 3

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

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 $\mathbf{E}(\phi \cup \psi)$

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





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

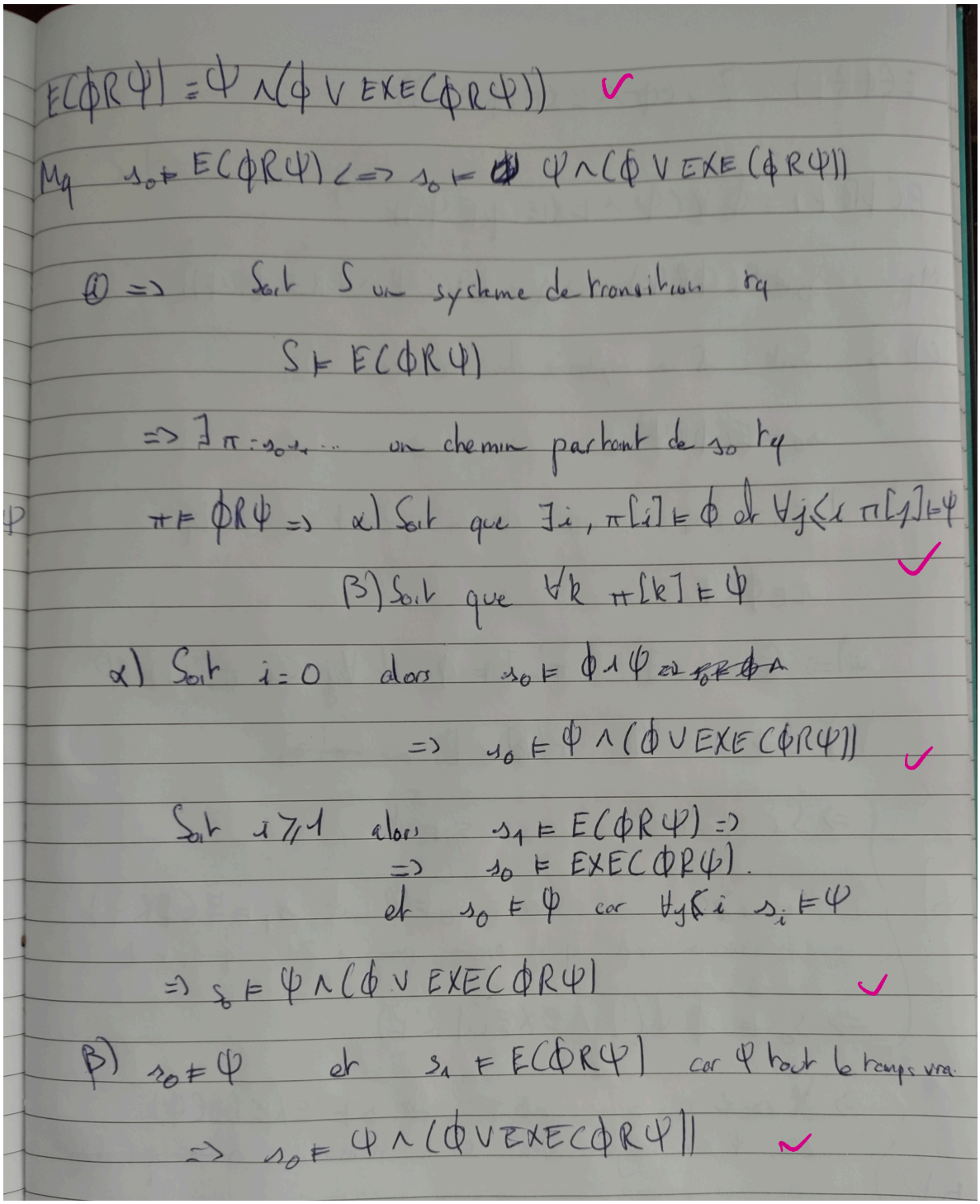
$$\exists \pi : s_0, s_1, ..$$

par

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

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

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



ii) $\Leftarrow$ Soit S un système de transition tq

$$S \models \psi \wedge (\phi \vee EXE(\phi R \psi))$$

$\Rightarrow$ α) Soit $s \models \psi \wedge \phi$

β) Soit $s \models \psi \wedge EXE(\phi R \psi)$

α) $s \models \psi \wedge \phi \Rightarrow \forall \pi : s_0 s_1 \dots \quad \pi \models \phi R \psi$

$\Rightarrow s_0 \models E(\phi R \psi)$ ✓

β) $\Rightarrow s_0 \models \psi$ et $\exists \pi : s_0 s_1 \dots$ tq $\pi[1] \models E(\phi R \psi)$

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

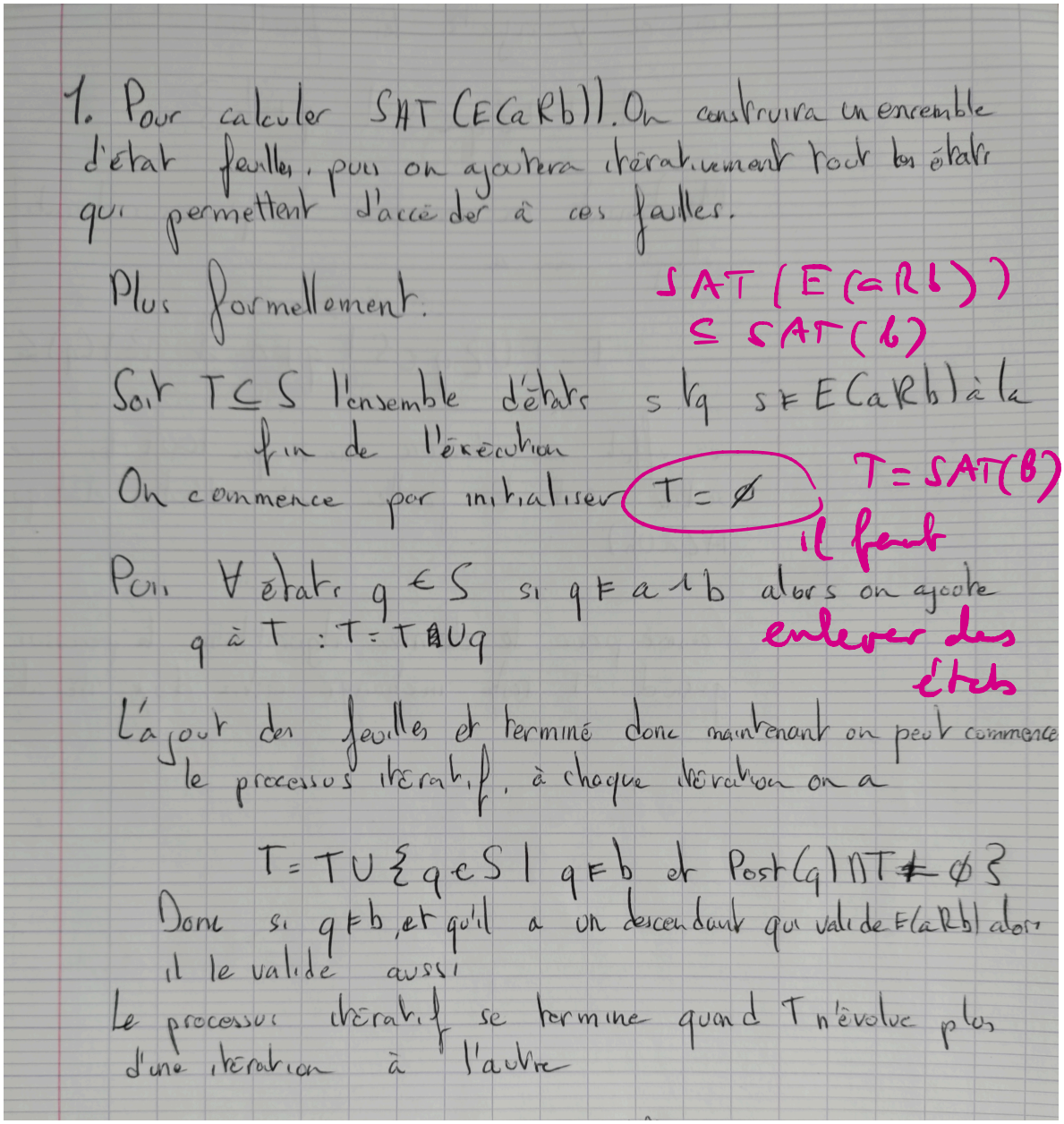
Or on sait que $\pi[0] \models \psi$ Donc

Soit $\exists i \geq 0$ tq $\pi[i] \models \phi$ et $\forall j < i \ \pi[i] \models \psi$
ou $\forall k \geq 0 \ \pi[k] \models \psi$

$\Rightarrow s_0 \models E(\phi R \psi)$ ✓

4 Exercice 4

4.1 $Sat(\mathbf{E}(a\mathbf{R}b)$



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 $Sat(\mathbf{A}(a\mathbf{U}b)$

Pour calculer $Sat(\mathbf{A}(a\mathbf{U}b)$. On construira aussi un ensemble de d'états feuille, puis on ajoutera itérativement les états dont TOUT les descendants permettent d'acceder à ses feuilles.

On commence par initialiser $T = \emptyset$

Puis on y ajoute les feuilles

$$T = T \cup \{ q \in S \mid q \not\models b \}$$

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

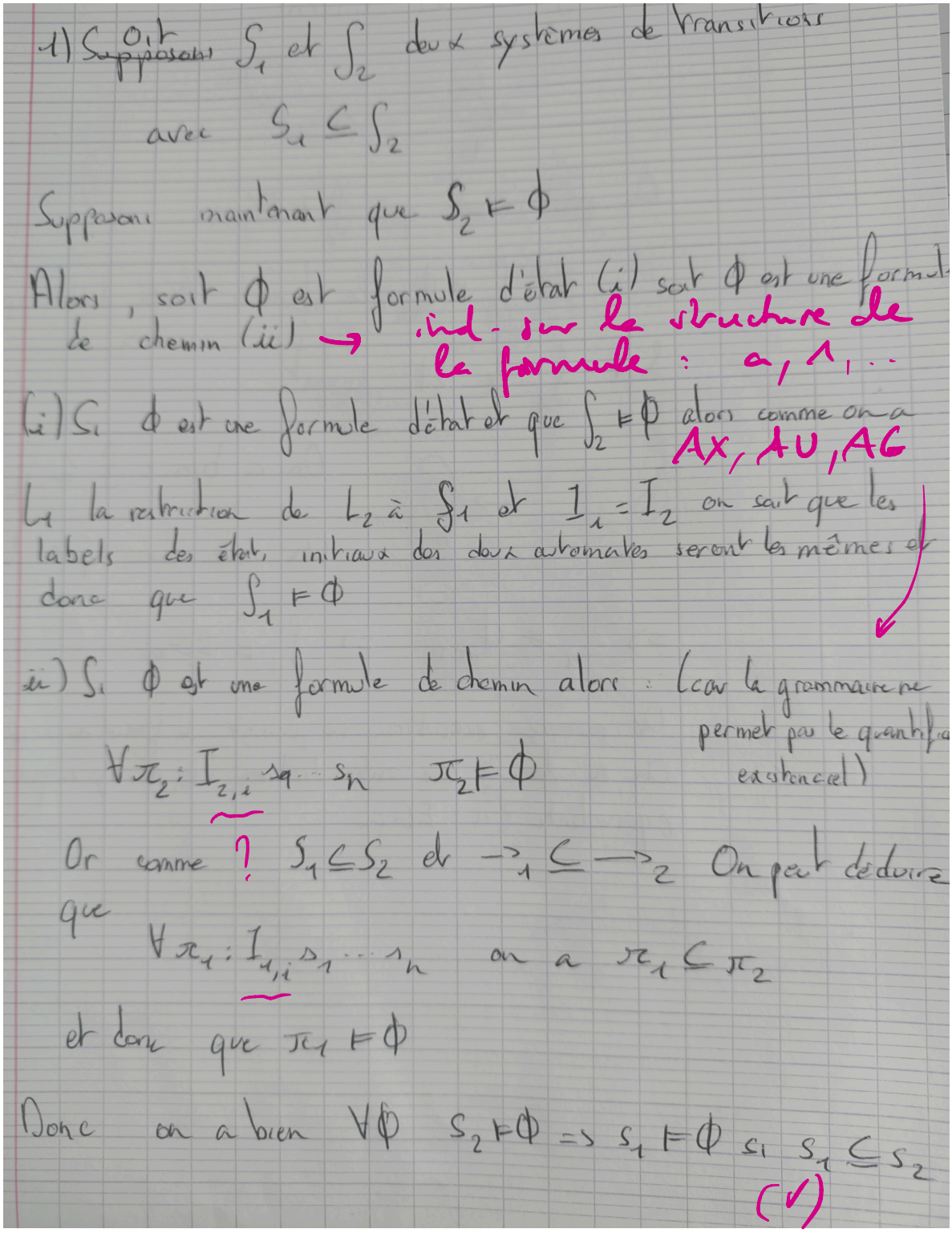
$$T = T \cup \{ q \in S \mid q \not\models a \text{ et } Post(q) \subseteq T \}$$

en effet on s'assure ainsi que TOUT les descendant de q soient soit des feuilles soit d'autres etat 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 $S_2 \models \phi \implies S_1 \models \phi$



5.2 Séparation CTL/ACTL