Introduction à la vérification

16

exo 1

$\phi = \underbrace{AF(b \wedge \neg a)}_{\phi_1} \Rightarrow \underbrace{AF\, A(b\, W(\neg a \wedge \neg b))}_{\phi_2}$

$\phi_1 = \neg EG(\neg(b \wedge \neg a))$

$\phi_2 = \neg EF \neg \neg E((b \wedge (\neg a \wedge \neg b)) \cup (\neg b \wedge \neg(\neg a \wedge \neg b)))$

$= \neg EGE(b \cup (\neg b \wedge a))$

$a \Rightarrow b$
$\neg a \vee b$
$\neg(a \wedge \neg b)$

$\phi = \neg(\neg EG(\neg(b \wedge \neg a)) \wedge \underbrace{EGE(b \cup (\neg b \wedge a))}_{\phi_3})$ ✓

exo 1.2

$Sat(a) = \{s_0, s_5\}$

$Sat(b) = \{s_0, s_2, s_3\}$

$Sat(\neg b \wedge a) = \{s_5\} = (S \setminus Sat(b)) \cap Sat(a)$

$Sat(E(b \cup (\neg b \wedge a))) = \{s_5, s_2, s_0\}$

Par plus petit point fixe

$T_0 = Sat(\neg b \wedge a) = \{s_5\}$

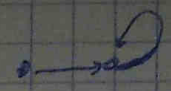
$T_1 = T_0 \cup (Sat(b) \cap Pred(T_0)) = \{s_5\} \cup \{s_2\} = \{s_5, s_2\}$

$T_2 = \{s_5, s_2\} \cup \{s_2, s_0\} = \{s_5, s_2, s_0\}$

$T_3 = \{s_5, s_2, s_0\} \cup \{s_2, s_0\}$ plus petit point fixe atteint $= T_2$

✓

$Sat(EG(\phi_3)) = \{s_0, s_2, s_5\}$



par plus grand pt fixe

~~$Sat(\phi_3) = \{s_0, s_2, s_5\}$~~

✓

	T			V		
	s_0	s_2	s_5			
successeurs	~~3~~	~~2~~	~~2~~	~~s_1~~	~~s_3~~	~~s_4~~
	~~2~~	1	1			
	1					

$Sat(\neg(b \wedge \neg a)) = S \setminus (Sat(b) \cap (S \setminus Sat(a)))$

$= \{s_0, s_1, s_4, s_5\}$

$Sat(EG(\neg(b \wedge \neg a))) = \{s_0, s_1\}$ ✓

	T				V	
	s_0,	s_1,	~~s_4~~,	~~s_5~~	~~s_2~~	~~s_3~~
successeurs	~~3~~	~~2~~	~~2~~	~~2~~	~~s_4~~	
	~~2~~	1		~~2~~	~~s_5~~	
	1					

✓

$Sat(\phi) = S \setminus ((S \setminus \{s_0, s_1\}) \cap \{s_5, s_2, s_0\})$

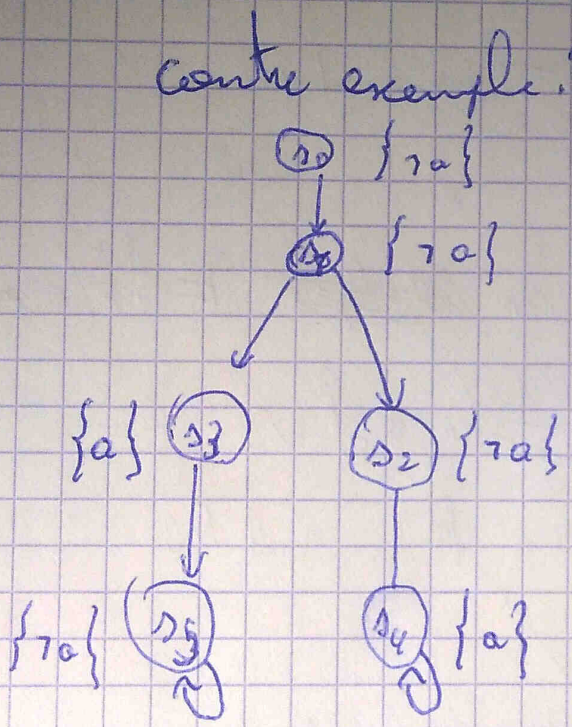
$= \{s_0, s_1, s_3, s_4\}$

✓

exo 2

1. $AFAX_a \overset{?}{\Leftrightarrow} FX_a$

contre exemple :



s_0 satisfait FX_a car s_2 satisfait X_a et s_1 satisfait X_a dans le cas

$s_1 \to s_3$. Tous les chemin satisfont donc la formule.

s_0 ne satisfait pas $AFAX_a$: dans l'exemple

$s_0 \to s_1 \to s_3$ ~~$\to s_5$~~)

car s_1 a s_2 comme successeur qui ne satisfait pas a et donc ne satisfait pas AX_a

Les deux formules ne sont donc pas équivalentes

2.2 $AF(a \wedge AFa) \Leftrightarrow F(a \wedge Fa)$

Soit S un système de transition avec s_0 état initial et $s_0 \models AF(a \wedge AFa)$

par définition :

$\forall$ exécution partant de $s_0 : s_0 \to s_1 \dots$

$\exists n \geq 0$ tq $s_n \models a \wedge AFb$

donc

$\exists m \geq n$ tq $s_m \models b$

donc $\forall \pi$ chemin de S $\exists$ un n et un m tel que précédent

donc $\pi \models F(a \wedge Fb)$ $\checkmark$ $\Rightarrow$ $S \models F(a \wedge Fb)$

Soit S un système de transition où $\forall$ chemin π, $\pi \models F(a \wedge Fb)$

par définition

Il existe un état $s_n \in \pi$ (avec $n \geq 0$) où $s_n \models a \wedge Fb$

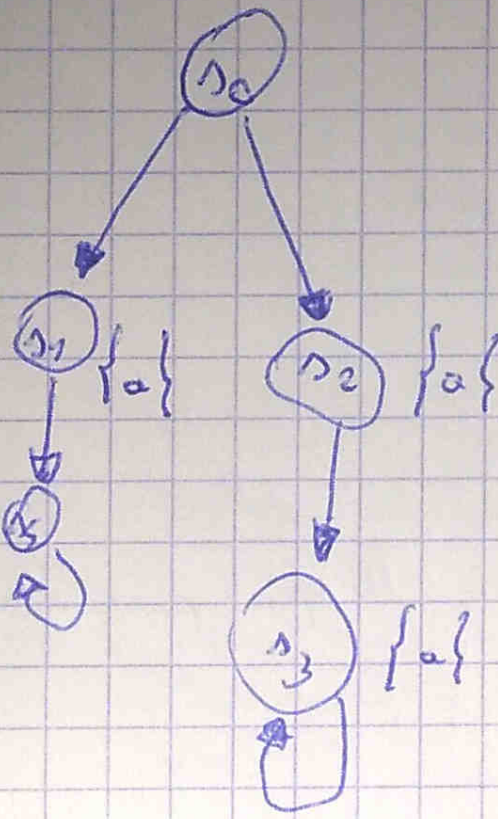
et donc s_m $(m \geq n)$ où $s_m \models b$

ce qui implique que s_0 (premier état de π) $\models AF(a \wedge AFb)$

ça parle d'une exécution → plus que ça !

2.3

$$EG\,AX_a \overset{?}{\iff} GX_a$$



$s_0 \not\models AX_a$

$s_2 \not\models AX_a$

$s_3 \not\models AX_a$

$\Rightarrow s_0 \not\models EG\,AX_a$

le chemin $s_0 \to s_1 \to s_3$ ne satisfait pas GX_a car $s_1 \not\models X_a$

Les deux formules ne sont donc pas équivalentes ✓

3.1

$A(\phi U \varphi)$ peut être défini récursivement :

$A(\phi U \varphi) \Leftrightarrow \varphi \vee (\phi \wedge AX\, A(\phi U \varphi))$ ✓

donc $\boxed{Sat(\varphi) \subseteq T}$

on pose $T_0 = Sat(\varphi)$

à partir de cet ensemble on ajoute itérativement ajoutés les états qui satisfont : si $\vdash \phi$ et $\forall s_j$ successeur de s_i $\Delta_j \subseteq T_{n-1}$

l'algo explique que $Sat(A\phi U \psi)$ satisfait (1) et (2), mais pas la minimalité

l'algorithme s'arrête lorsque $T_n = T_{n-1}$, il n'existe alors plus d'état satisfaisant la formule qui ne soit pas inclus dans T_{n-1}

(invariant : $T_{n-1} \subseteq T_n \subseteq T \subseteq S$)

S'étant fini l'algorithme termine toujours.

Il fallait montrer aussi que $Sat(A(\phi U \psi)) \subseteq T$, pour tout T qui satisfait (1) et (2)

Tous les états ajoutés satisfont donc $Sat(\phi)$ et $post(s) \in T$

l'ensemble est donc bien minimal

3.4

La définition rencontrée de $AG\phi$ est

$$AG\phi \Leftrightarrow \phi \wedge AX AG\phi$$

donc $\boxed{T \subseteq Sat(\phi)}$ ✓

on pose $T_0 = Sat(\phi)$

On retire itérativement les état s tq $succ(s) \not\subseteq T_{n-1}$

Invariants : $T \subseteq T_n \subseteq T_{n-1} \subseteq Sat(\phi) \subseteq S$

On s'arrête lorsque $T_{n-1} = T_n$, il n'y a donc plus d'état qui ne satisfont pas $post(s) \subseteq T_{n-1}$ donc $T_n = T$ et

$\forall s \in T \; post(s) \subseteq T$ (S étant fini l'algorithme termine bien)

On a donc retiré uniquement des états qui ne satisfont pas les formules $s \in Sat(\phi)$ tq $post(s) \subseteq T$

Le nombre d'état retiré est donc minimal et l'ensemble est le plus grand satisfaisant la formule. idem (3.1)

3.3

$$A(\phi \mathcal{U} \varphi) \Leftrightarrow \neg E((\phi \wedge \neg\varphi) U (\neg\phi \wedge \neg\varphi))$$

On va donc procéder par plus petit point fixe sur $\mathcal{P}(T_n (= V_n))$

$V_0 = Sat(\neg\phi \wedge \neg\varphi)$ $\quad T_0 = Sat(\phi \vee \varphi)$

On ajoute itérativement à V_n les états $s \in T_{n-1}$ t.q.

$succ(s) \cap V_{n-1} \neq 0 \wedge s \notin Sat(\varphi)$

invariant : $Sat(\varphi) \subseteq T_n \subseteq T_{n-1} \subseteq Sat(\phi \vee \varphi) \subseteq S$

On s'arrête lorsque $T_n = T_{n-1}$

Tous les états de T satisfont donc $s \in Sat(\varphi) \cup \{s \in Sat(\varphi) \mid succ(s) \subseteq T\}$

~~Donc~~ S'étant fini l'algorithme s'arrête toujours

On a donc retiré uniquement des états qui ne satisfaisaient pas la formule à T, l'ensemble est donc le plus grand possible.

idem

4.1

$S_1 \models \phi \Rightarrow S_2' \models \phi$ est équivalent à

$(\forall s_2 \in I_2 \; s_2 \models \phi) \Leftrightarrow (\forall s_2 \in I_1 \; s_2 \models \phi)$

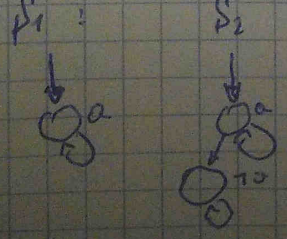
$I \subseteq S$

$a, \neg a$ {
- ou $L_2(s) = L_1(s) \; \forall s \in S_1$ $\quad I_1 = I_2$
- donc $a \in L_1(s) \Leftrightarrow a \in L_2(s)$
- donc $s \models a$ dans $S_2' \Leftrightarrow s \models a$ dans $S_1' \quad \forall s \in S_1'$
- idem pour $s \models \neg a$ (on pose $b = \neg a$)

E {
- $s \models E\varphi$ signifie qu'il existe un chemin qui satisfait φ
- un chemin est constitué de transitions et d'états
- donc si $S_1 \subseteq S_2$ et $\to_1 \subseteq \to_2$
- donc si un chemin satisfait φ dans S_1' alors il est aussi dans S_2'
- donc $s \models E\varphi$ dans $S_1' \Rightarrow s \models E\varphi$ dans $S_2' \quad \forall s \in S_1'$ ✓

$\wedge$ {
- $s \models \phi_1 \wedge \phi_2 \quad \forall s \in S_1$
- $\Leftrightarrow$
- $s \in Sat(\phi_1) \cap Sat(\phi_2)$
- or $Sat(\phi_1)$ dans $S_1' \subseteq Sat(\phi_2)$ dans S_2
- donc
- $s \in Sat(\phi_1) \cap Sat(\phi_2)$ dans $S_1' \subseteq Sat(\phi_1) \cap Sat(\phi_2)$ dans S_2' ✓
- ...

4.2



AGa est une formule CTL valide

$S_1' \not\models AGa$

$S_2 \not\models AGa$ ✓

très bien

$S_2 \subseteq S_2'$ ⇒ contradiction, AGa ne peut pas être exprimé en ELTL