Construction de $\mathscr{A}_\phi$ - 1

Etant donnée une formule de LTL ϕ, on veut construire un automate $\mathscr{A}_\phi$ qui reconnait le langage mod(ϕ).

$\mathscr{A}_\phi = (Q, q_0, \rightarrow, \mathscr{F})$ sera un automate de Büchi généralisé.
kesako ?

Un automate de Büchi reconnait des mots infinis: un mot w est accepté si il existe un chemin dans l'automate dont l'étiquetage correspond à w et si le chemin passe infiniment souvent par un des états acceptants (un sous-ensemble de Q).

Dans un automate de Büchi généralisé, les états acceptants sont donnés par un ensemble de sou-ensemble $\mathscr{F}=\{\mathscr{F}_1,\ldots,\mathscr{F}_k\}$: un « bon » chemin doit passer infiniment souvent par un des états de chaque $\mathscr{F}_i$…

Construction de $\mathscr{A}_\phi$ - 2

$$\mathscr{A}_\phi = (Q, Q_0, \rightarrow, \mathscr{F})$$

Chaque état de Q sera associé (défini) par un sous-ensemble de sous-formules de ϕ.
idée de la construction: depuis un état associé à l'ensemble de sous-formules $\{\psi_1,\ldots,\psi_n\}$, on reconnait des mots vérifiant chacune de ses sous-formules.

Comme $\mathscr{A}_\phi$ doit reconnaître les modèles de ϕ, l'ensemble des états initiaux Q_0 contiendra tous les états de Q contenant la sous-formule ϕ…

Construction de $\mathscr{A}_\phi$ - 3

$\mathscr{A}_\phi = (Q, Q_0, \rightarrow, \mathscr{F})$

Comment définir les états Q ? Quels ensembles de sous-formules de ϕ choisir ?

On va choisir des sous-ensembles cohérents (logiquement), maximaux et conforme à la sémantique de LTL.

Soit S_ϕ l'ensemble des sous-formules de ϕ et leur négation.

Exemple:
$\phi = a \mathbf{U} (\mathbf{X}\, b)$
$S_\phi = \{a, \neg a, b, \neg b, \mathbf{X}\, b, \neg\mathbf{X}\, b, a \mathbf{U} (\mathbf{X}\, b), \neg(a \mathbf{U} (\mathbf{X}\, b))\}$

Construction de $\mathscr{A}_\phi$ - 4

Comment définir les états Q ?

$\mathscr{A}_\phi = (Q, Q_0, \rightarrow, \mathscr{F})$

Les états q sont des sous-ensembles cohérents, maximaux et conforme à la sémantique de LTL…

- **Cohérents:**

Si $\psi_1 \wedge \psi_2 \in q \Rightarrow \psi_1, \psi_2 \in q$,

si $\neg(\psi_1 \wedge \psi_2) \in q \Rightarrow (\psi_1 \notin q$ ou $\psi_2 \notin q)$,

Si $\psi_1 \vee \psi_2 \in q \Rightarrow (\psi_1 \in q$ ou $\psi_2 \in q)$,

si $\neg(\psi_1 \vee \psi_2) \in q \Rightarrow (\psi_1 \notin q$ et $\psi_2 \notin q)$,

Si $\psi \in q$, alors $\neg\psi \notin q$.

- **Maximaux:**

Dans tout état, pour chaque sous-formule ψ, on met soit ψ, soit $\neg\psi$.

Construction de $\mathscr{A}_\phi$ - 5

Comment définir les états Q ? $\mathscr{A}_\phi = (Q, Q_0, \rightarrow, \mathscr{F})$

Les états q sont des sous-ensembles cohérents, maximaux et conforme à la sémantique de LTL…

- **Conforme à la sémantique de LTL:**

Dans tout état q, si la sous-formule $\psi_1 \mathbf{U} \psi_2$ est présente, alors on a soit ψ_1, soit ψ_2 dans l'état q.

Si $\psi_1 \mathbf{U} \psi_2 \in S\phi$, alors si ψ_2 est dans un état q, $\psi_1 \mathbf{U} \psi_2 \in q$

Et les états initiaux Q_0 sont ceux contenant ϕ.

Construction de $\mathscr{A}_\phi$ - 6

Comment définir les états Q ? $\mathscr{A}_\phi = (Q, Q_0, \rightarrow, \mathscr{F})$

Ce sont des sous-ensembles cohérents, maximaux et conforme à la sémantique de LTL…

Exemple:

$\phi = a \mathbf{U} (b \wedge c)$

$S_\phi = \{a, \neg a, b, \neg b, c, \neg c, b\wedge c, \neg(b\wedge c), a \mathbf{U} (b\wedge c), \neg(a \mathbf{U} (b\wedge c)) \}$

$q=\{a,\neg b,c,\neg(b\wedge c),a \mathbf{U} (b\wedge c)\}$ ou

$q'=\{a,\neg b,c,\neg(b\wedge c),\neg (a \mathbf{U} (b\wedge c))\}$ sont ok !

Mais $r=\{a,b,\neg b,c,\neg(b\wedge c),a \mathbf{U} (b\wedge c)\}$,

$r'= \{a,b,c,\neg(b\wedge c),a \mathbf{U} (b\wedge c)\}$ ou

$r''= \{a,b,c,(b\wedge c),\neg(a \mathbf{U} (b\wedge c))\}$ ne sont pas bien formés !

Construction de $\mathscr{A}_\phi$ - 7

$\mathscr{A}_\phi = (Q, Q_0, \rightarrow, \mathscr{F})$

Comment définir les transitions de l'automate $\mathscr{A}_\phi$?

On met une transition $(q,\sigma,q') \in Q\text{x}2^{AP}\text{x}Q$ si et seulement si:

- $\sigma = q \cap AP$ (*ie* les prop. atomiques de q)
- $\forall\, \mathbf{X}\psi \in S_\phi$, $\mathbf{X}\psi \in q \iff \psi \in q'$
- $\forall\, \psi_1\, \mathbf{U}\, \psi_2 \in S_\phi$, $\psi_1\, \mathbf{U}\, \psi_2 \in q \iff (\psi_2 \in q \vee (\psi_1 \in q \wedge \psi_1\, \mathbf{U}\, \psi_2 \in q'))$

Exemple:

$\phi = a\, \mathbf{U}\, (b \wedge c)$

$S_\phi = \{a, \neg a, b, \neg b, c, \neg c, b\wedge c, \neg(b\wedge c), a\, \mathbf{U}\, (b\wedge c), \neg(a\, \mathbf{U}\, (b\wedge c))\}$

$q=\{a,\neg b,c,\neg(b\wedge c),a\, U\, (b\wedge c)\}$

$q'=\{\neg a,b,c,(b\wedge c),(a\, U\, (b\wedge c))\}$

On a : q —- {a,c} ——> q'

Construction de $\mathscr{A}_\phi$ - 8

$\mathscr{A}_\phi = (Q, Q_0, \rightarrow, \mathscr{F})$

Comment définir les conditions d'acceptation $\mathscr{F}$?

Pour chaque sous-formule $\psi_1\mathbf{U}\psi_2$, on a un ensemble $\mathscr{F}_{\psi_1\mathbf{U}\psi_2}$ défini par:

$$\mathscr{F}_{\psi_1\mathbf{U}\psi_2} = \{\, q \in Q \mid \psi_1\mathbf{U}\psi_2 \notin q \;\vee\; \psi_2 \in q \,\}$$

idée: un état contenant $\psi_1\mathbf{U}\psi_2$ doit reconnaître les modèles de $\psi_1\mathbf{U}\psi_2$ et donc visiter un jour un état contenant ψ_2.

Pour en être sûr, on impose de visiter infiniment souvent des états contenant ψ_2... ou infiniment souvent des états contenant $\neg\, \psi_1\mathbf{U}\psi_2$...

Dans les deux cas, on est sûr de ne pas attendre indéfiniment la satisfaction de ψ_2.

$\mathscr{A}_\phi$ et les modèles de ϕ

$\mathscr{A}_\phi = (Q, Q_0, \rightarrow, \mathscr{F})$

Prenons un chemin dans $\mathscr{A}_\phi$ $q_1 \rightarrow q_2 \rightarrow q_3 \rightarrow q_4 \rightarrow \dots$
étiqueté par le mot $\sigma_1 \sigma_2 \sigma_3 \sigma_4 \dots$ de $(2^{AP})^\omega$

Alors on a:

$$\forall\, \psi \in q_i, \quad \sigma_i \sigma_{i+1} \sigma_{i+2} \dots \models \psi$$

Construction de $\mathscr{A}_\phi$ - exemple

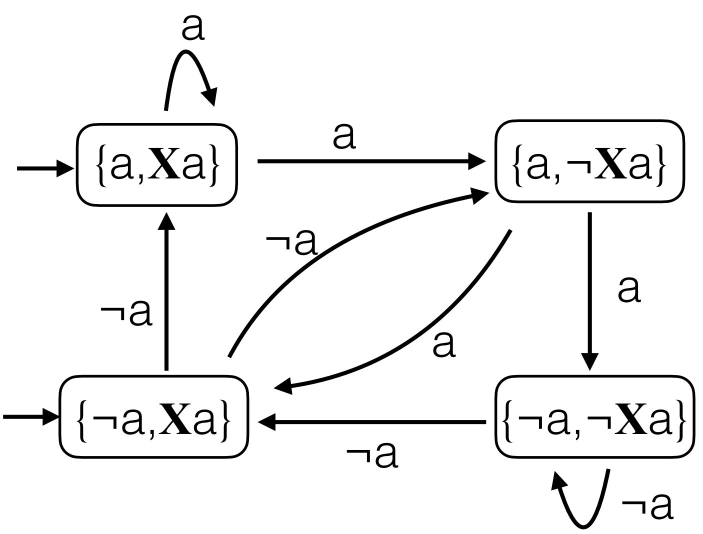
$\phi = \mathbf{X}\, a$

$\mathscr{A}_\phi = (Q, Q_0, \rightarrow, \mathscr{F})$

$S_\phi = \{a, \neg a, \mathbf{X}a, \neg\mathbf{X}a\}$

$\mathscr{F} = \{Q\}$

$q_1 = \{a, \mathbf{X}a\}$,
$q_2 = \{a, \neg\mathbf{X}a\}$
$q_3 = \{\neg a, \mathbf{X}a\}$
$q_4 = \{\neg a, \neg\mathbf{X}a\}$



$\{a\}\{a\}\{\neg a\}\dots \qquad \{\neg a\}\{a\}\{a\}\dots \qquad \{a\}\{a\}\{a\}\dots$

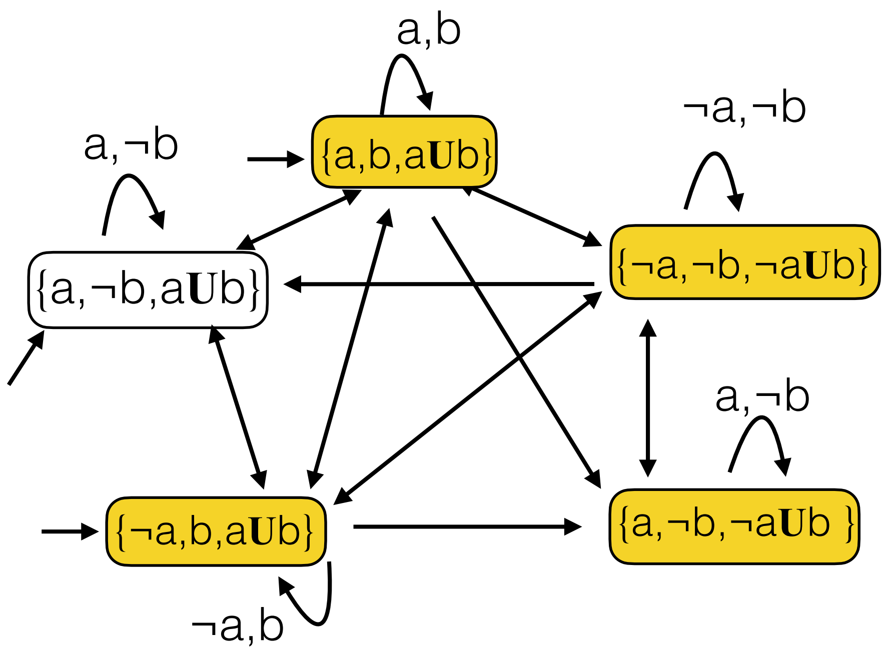
Construction de $\mathscr{A}_\phi$ - exemple

$\phi = a \,\mathbf{U}\, b$

$\mathscr{A}_\phi = (Q, Q_0, \rightarrow, \mathscr{F})$

$S_\phi = \{a, \neg a, b, \neg b, a\mathbf{U}b, \neg a\mathbf{U}b\}$

$q_1 = \{a, b, a\mathbf{U}b\}$,
$q_2 = \{\neg a, b, a\mathbf{U}b\}$
$q_3 = \{a, \neg b, a\mathbf{U}b\}$
$q_4 = \{\neg a, \neg b, \neg a\mathbf{U}b\}$
$q_5 = \{a, \neg b, \neg a\mathbf{U}b\}$



état acceptant

$\{a,\neg b\}\{a,\neg b\}\{a,b\}\ldots$

Construction de $\mathscr{A}_\phi$ - exemple

$\phi = \mathbf{G}\,(a \Rightarrow \mathbf{F}\, b)$

$\mathscr{A}_\phi = (Q, Q_0, \rightarrow, \mathscr{F})$

$S_\phi = \{a, \neg a, b, \neg b, \mathbf{F}b, \neg\mathbf{F}b, \neg a \vee \mathbf{F}b, a \wedge \neg\mathbf{F}b, \phi, \neg\phi\}$

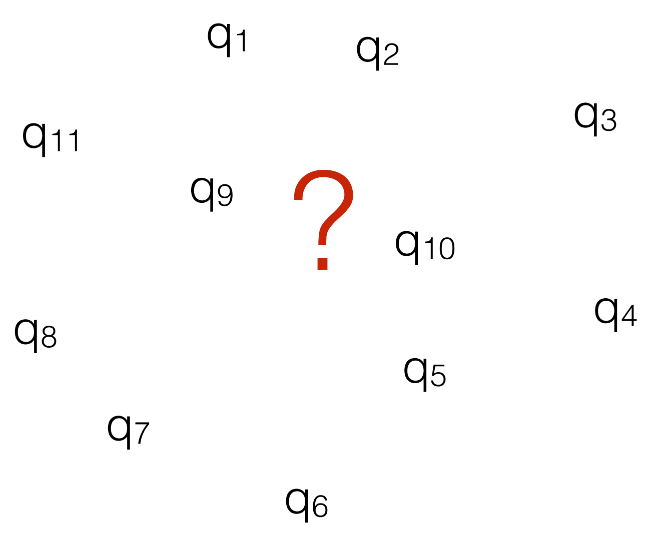
if $q_1 = \{a, b, \mathbf{F}b, a\Rightarrow\mathbf{F}b, \phi\}$,
f $q_2 = \{a, b, \mathbf{F}b, a\Rightarrow\mathbf{F}b, \neg\phi\}$,
i $q_3 = \{a, \neg b, \mathbf{F}b, a\Rightarrow\mathbf{F}b, \phi\}$,
$q_4 = \{a, \neg b, \mathbf{F}b, a\Rightarrow\mathbf{F}b, \neg\phi\}$,
f $q_5 = \{a, \neg b, \neg\mathbf{F}b, \neg(a\Rightarrow\mathbf{F}b), \neg\phi\}$,
if $q_6 = \{\neg a, b, \mathbf{F}b, a\Rightarrow\mathbf{F}b, \phi\}$,
f $q_7 = \{\neg a, b, \mathbf{F}b, a\Rightarrow\mathbf{F}b, \neg\phi\}$,
i $q_8 = \{\neg a, \neg b, \mathbf{F}b, a\Rightarrow\mathbf{F}b, \phi\}$,

$q_9 = \{\neg a, \neg b, \mathbf{F}b, a\Rightarrow\mathbf{F}b, \neg\phi\}$,
if $q_{10} = \{\neg a, \neg b, \neg\mathbf{F}b, a\Rightarrow\mathbf{F}b, \phi\}$,
f $q_{11} = \{\neg a, \neg b, \neg\mathbf{F}b, a\Rightarrow\mathbf{F}b, \neg\phi\}$.

i : états initiaux
f : états acceptant

Construction de $\mathscr{A}_\phi$ - exemple

$\phi = \mathbf{G}\,(a \Rightarrow \mathbf{F}\, b)$ $\qquad \mathscr{A}_\phi = (Q, Q_0, \rightarrow, \mathscr{F})$



On remarque ici que on ne peut jamais aller d'un état contenant ϕ à un état contenant $\neg\phi$…

Construction de $\mathscr{A}_\phi$ - exemple

{a,b}{a,b}{a,¬b}{a,¬b}{a,¬b}{¬a,¬b}{¬a,b}…

$\phi = \mathbf{G}\,(a \Rightarrow \mathbf{F}\, b)$ $\qquad \mathscr{A}_\phi = (Q, Q_0, \rightarrow, \mathscr{F})$

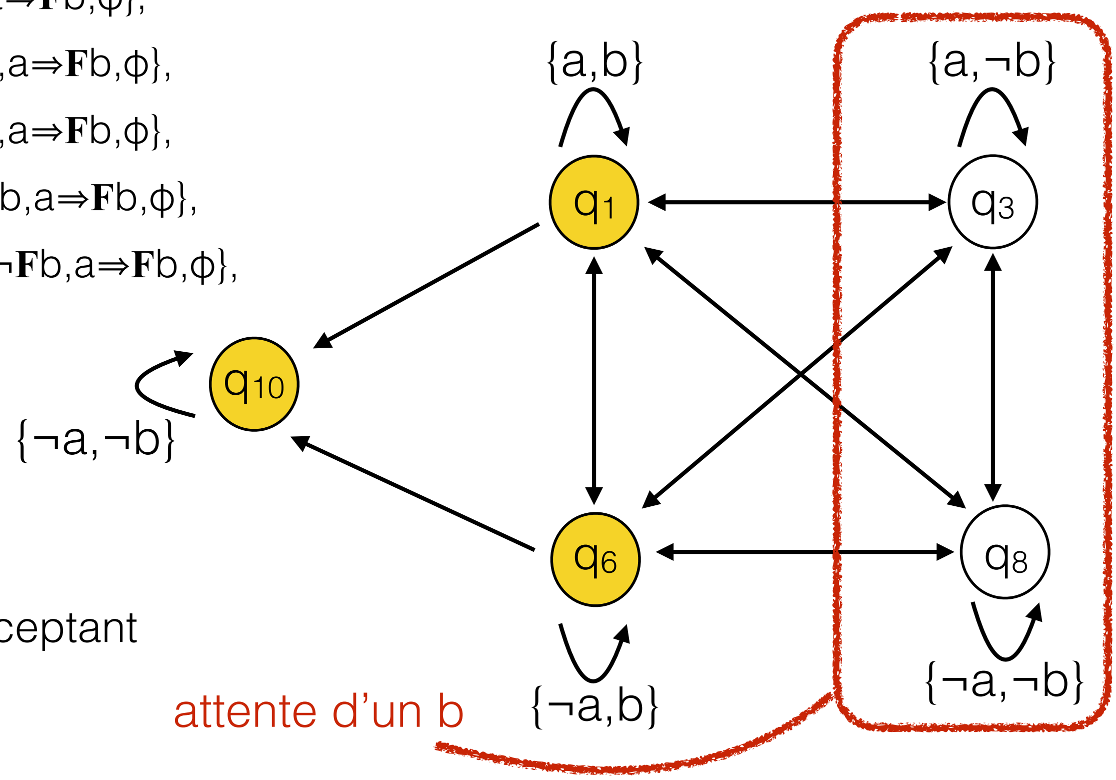
$q_1 = \{a, b, \mathbf{F}b, a \Rightarrow \mathbf{F}b, \phi\}$,

$q_3 = \{a, \neg b, \mathbf{F}b, a \Rightarrow \mathbf{F}b, \phi\}$,

$q_6 = \{\neg a, b, \mathbf{F}b, a \Rightarrow \mathbf{F}b, \phi\}$,

$q_8 = \{\neg a, \neg b, \mathbf{F}b, a \Rightarrow \mathbf{F}b, \phi\}$,

$q_{10} = \{\neg a, \neg b, \neg\mathbf{F}b, a \Rightarrow \mathbf{F}b, \phi\}$,

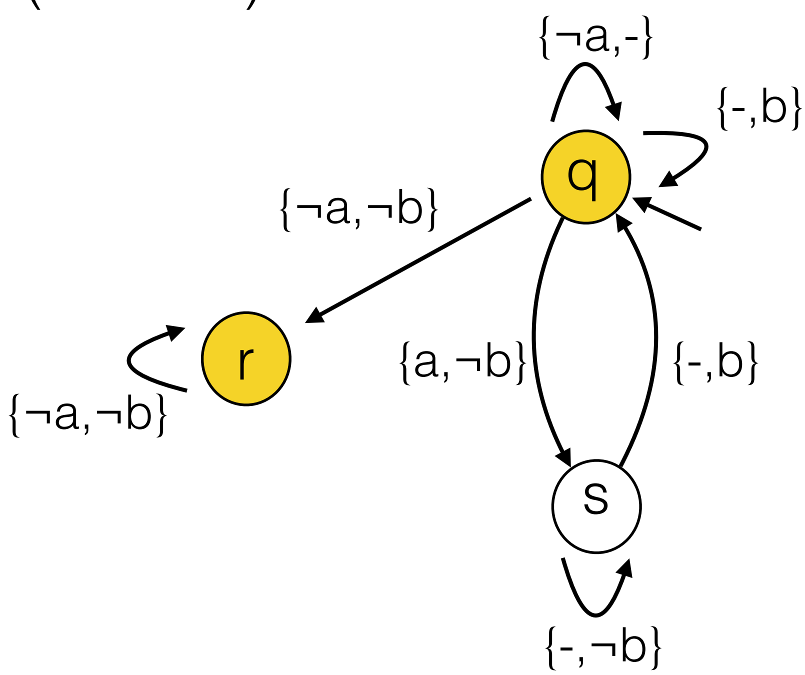


état acceptant

Construction de $\mathscr{A}_\phi$ - exemple

(version simplifiée)

$\phi = \mathbf{G}\,(a \Rightarrow \mathbf{F}\, b)$



{a,b}(a,b}{a,¬b}{a,¬b}{a,¬b}{¬a,¬b}{¬a,b}…

Correction de la construction

Théorème 1:
soit $w=w_0w_1\ldots \in (2^{AP})^\omega$ et $\rho = q_0q_1\ldots$ une exécution acceptante de $\mathscr{A}_\phi$ sur le mot w, alors on a:
$\forall\, i\geq 0,\ \forall\, \psi \in S_\phi, \quad (\psi \in q_i \iff w,i \models \psi)$

(Preuve par induction structurelle sur ψ)

Corolaire: $\mathscr{L}(\mathscr{A}_\phi) \subseteq \mathrm{mod}(\phi)$

Théorème 2:
soit $w=w_0w_1\ldots \in (2^{AP})^\omega$ t.q. $w,0 \models \phi$, alors on a: $w \in \mathscr{L}(\mathscr{A}_\phi)$.

(Preuve: on construit une exécution acceptante sur w…)

Corolaire: $\mathrm{mod}(\phi) \subseteq \mathscr{L}(\mathscr{A}_\phi)$

Problèmes de vérification pour LTL

1) $\mathbf{S} \models \phi$? Traces($\mathbf{S}$) $\subseteq$ mod(ϕ)

$$\mathscr{L}(\mathscr{A}_{\mathbf{S}}) \cap \mathscr{L}(\mathscr{A}_{\neg\phi}) = \varnothing \ ?$$

2) $\mathbf{S} \models \neg\, \phi$? Traces($\mathbf{S}$) $\cap$ mod(ϕ) $= \varnothing$

$$\mathscr{L}(\mathscr{A}_{\mathbf{S}}) \cap \mathscr{L}(\mathscr{A}_{\phi}) = \varnothing \ ?$$

3) Est-ce que ϕ est satisfaisable ?

$$\mathscr{L}(\mathscr{A}_{\phi}) \neq \varnothing$$

Problèmes de vérification pour LTL

La taille de l'automate $\mathscr{A}_{\phi}$ est exponentiel dans $|\phi|$!
Ces problèmes sont donc difficiles !

→ Tester si $\mathbf{S} \models \phi$ ou si ϕ est satisfaisable sont des problèmes PSPACE-complet.

Problèmes de vérification pour LTL

NuSMV

- $\mathbf{S} \models \phi$?

Oui. On définit $\mathbf{S}$ et ϕ…

- Est-ce que ϕ est satisfaisable ?
  Avec NuSMV, on peut tester si une formule est valide (*ie* vraie pour tout modèle).

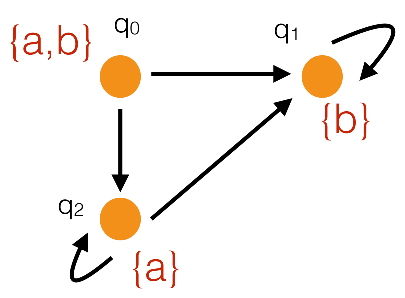
  ϕ est satisfaisable ssi $\neg\phi$ n'est pas valide (c'est-à-dire si il existe des modèles où $\neg\phi$ n'est pas vraie, donc où ϕ est vraie)…

Problèmes de vérification pour LTL

NuSMV

- $\mathbf{S} \models \phi$?

$\mathbf{S}$



$$\phi = \mathbf{G}\,(a \Rightarrow \mathbf{F}\, b)$$

```
MODULE main
VAR
    etat : {q0, q1, q2};
ASSIGN
    init(etat) := q0;
    next(etat) :=
            case
             etat=q0 : {q1,q2};
             etat=q1 : q1;
             etat=q2 : {q1, q2};
            esac;
DEFINE
    a := (etat=q0) | (etat=q2);
    b := (etat=q0) | (etat=q1);
LTLSPEC NAME
prop := G( a -> F b)
```

Problèmes de vérification pour LTL
NuSMV

- $\models \phi$?

$\phi = \mathbf{G}\,\mathbf{F}\,a \wedge \mathbf{G}\,\mathbf{F}\,b$

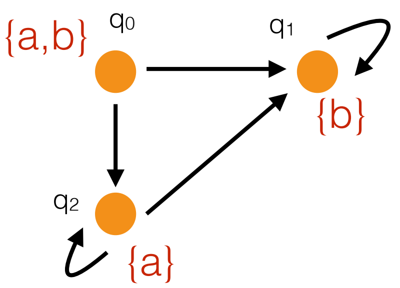
```
MODULE main
VAR
    a : boolean;
    b : boolean;
LTLSPEC NAME
prop1 :=  !(G F a & G F b);
```

Problèmes de vérification pour LTL
Prism

Model-checking

- $\mathbf{S} \models \phi$?

S



$\phi = \mathbf{G}\,(a \Rightarrow \mathbf{F}\,b)$

```
mdp

label "a" = (q=0 | q=2);
label "b" = (q=0) | (q=1);

module K
        q : [0..2] init 0;
    [] q=0 -> (q'=1);
    [] q=0 -> (q'=2);
    [] q=1 -> true;
    [] q=2 -> true;
    [] q=2 -> (q'=1);
endmodule
```

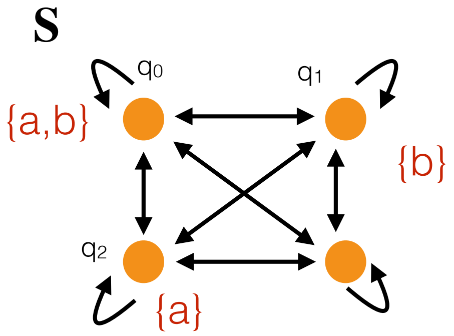
A [G (« a » => (F « b »))]

Problèmes de vérification pour LTL Prism

Satisfaisabilité

- $\models \phi$?

$\phi = \mathbf{G}\ \mathbf{F}\ a \wedge \mathbf{G}\ \mathbf{F}\ b$



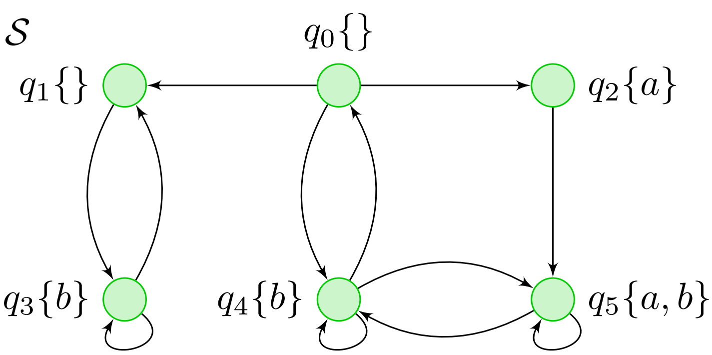

```
mdp

label "a" = (va=true);
label "b" = (vb=true);

module General
        va : bool init false;
        vb : bool init false;
    [] true -> (va'=true) & (vb'=true);
    [] true -> (va'=false) & (vb'=true);
    [] true -> (va'=true) & (vb'=false);
    [] true -> (va'=false) & (vb'=false);
endmodule
```

E [X ((G F « a ») & (G F « b »))]

Problèmes de vérification pour LTL Prism

TD 2

- $\mathbf{S} \models \phi$?




```
mdp

label "a" = (q=2 | q=5);
label "b" = (q=3 | q=4 | q=5);

module General
      q : [0..5];
   [] (q=0) -> (q'=1);
   [] (q=0) -> (q'=2);
   [] (q=0) -> (q'=4);
   [] (q=1) -> (q'=3);
   [] (q=3) -> (q'=3);
   [] (q=3) -> (q'=1);
   [] (q=2) -> (q'=5);
   [] (q=4) -> (q'=0);
   [] (q=4) -> (q'=4);
   [] (q=4) -> (q'=5);
   [] (q=5) -> (q'=4);
   [] (q=5) -> (q'=5);
endmodule
```