Model checking temporisé

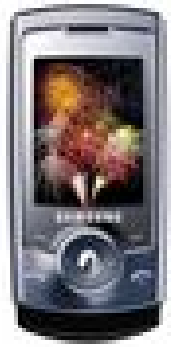
Béatrice Bérard

LAMSADE
Université Paris-Dauphine & CNRS
berard@lamsade.dauphine.fr

ETR'07, 5 septembre 2007

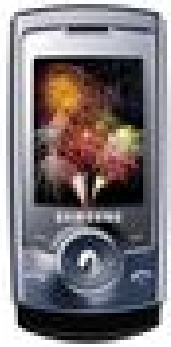
Nécessité de vérifier

des systèmes...



Nécessité de vérifier

des systèmes...



critiques

Nécessité de vérifier

avec des méthodes formelles

- génération de tests
- démonstration assistée
- model-checking

Nécessité de vérifier

avec des méthodes formelles

- génération de tests
- démonstration assistée
- model-checking

Nécessité de vérifier

avec des méthodes formelles

- génération de tests
- démonstration assistée
- model-checking

Nécessité de vérifier

avec des méthodes formelles

- génération de tests
- démonstration assistée
- model-checking

Le système satisfait-il sa spécification ?

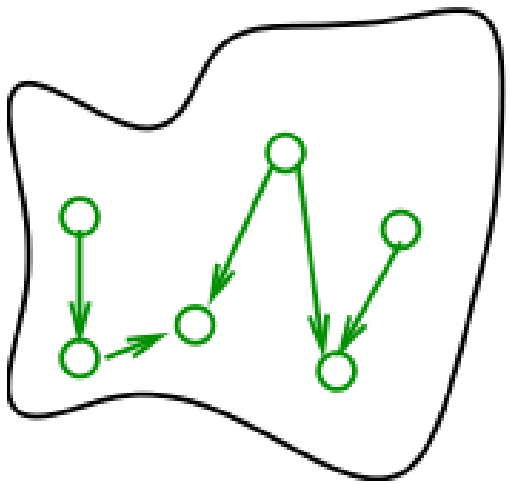
Nécessité de vérifier

avec des méthodes formelles

- génération de tests
- démonstration assistée
- model-checking

Le système satisfait-il sa spécification ?

Modélisation



φ

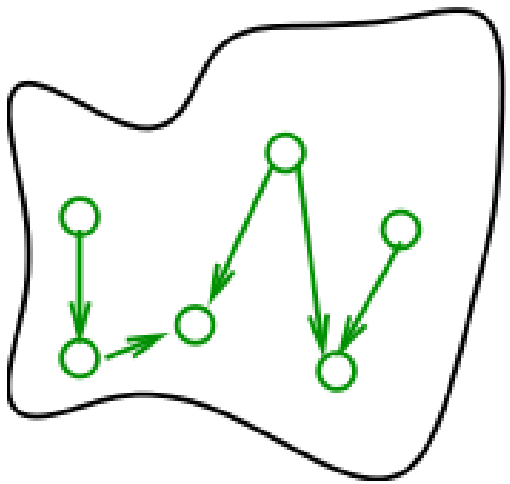
Nécessité de vérifier

avec des méthodes formelles

- génération de tests
- démonstration assistée
- model-checking

Le système satisfait-il sa spécification ?

Modélisation



$\models$

φ

algorithme de model-checking

Difficultés

pour chacune des trois étapes

- Construction d'un modèle M avec un système de transitions :
  expressivité, taille du modèle (explosion combinatoire du nombre d'états).
- Choix d'une logique dans laquelle exprimer la formule φ :
  expressivité, facilité d'utilisation.
- Algorithme de model-checking ($M \models \varphi\,?$) pour une classe de modèles et un langage de spécification :
  décidabilité, complexité, efficacité des outils associés.

Difficultés

pour chacune des trois étapes

- Construction d'un modèle M avec un système de transitions :
  expressivité, taille du modèle (explosion combinatoire du nombre d'états).
- Choix d'une logique dans laquelle exprimer la formule φ :
  expressivité, facilité d'utilisation.
- Algorithme de model-checking ($M \models \varphi\,?$) pour une classe de modèles et un langage de spécification :
  décidabilité, complexité, efficacité des outils associés.

Difficultés

pour chacune des trois étapes

- Construction d'un modèle M avec un système de transitions :
  expressivité, taille du modèle (explosion combinatoire du nombre d'états).
- Choix d'une logique dans laquelle exprimer la formule φ :
  expressivité, facilité d'utilisation.
- Algorithme de model-checking ($M \models \varphi$?) pour une classe de modèles et un langage de spécification :
  décidabilité, complexité, efficacité des outils associés.

Vérification quantitative

en ajoutant des caractéristiques

- temporisées,
- probabilistes,
- de coût,
- ...

aux modèles : par exemple en tenant compte des délais dans des protocoles de communication,
aux logiques : par exemple pour exprimer des propriétés de temps de réponse,

tout en conservant des algorithmes de model-checking.

Vérification quantitative

en ajoutant des caractéristiques

- temporisées,
- probabilistes,
- de coût,
- ...

aux modèles : par exemple en tenant compte des délais dans des protocoles de communication,
aux logiques : par exemple pour exprimer des propriétés de temps de réponse,

tout en conservant des algorithmes de model-checking.

Vérification quantitative

en ajoutant des caractéristiques

- temporisées,
- probabilistes,
- de coût,
- ...

aux modèles : par exemple en tenant compte des délais dans des protocoles de communication,
aux logiques : par exemple pour exprimer des propriétés de temps de réponse,

tout en conservant des algorithmes de model-checking.

Plan

Plan

Systèmes de transitions

Définition

Act alphabet d'actions,
Prop ensemble de propositions atomiques,
$\mathcal{T} = (S, s_0, L, E)$ système de transitions

- S ensemble de configurations, s_0 configuration initiale,
- $L : S \mapsto 2^{\text{Prop}}$, étiquetage des configurations par des ensembles de propositions atomiques,
- $E \subseteq S \times Act \times S$ contient

  des transitions d'action : $s \xrightarrow{a} s'$, exécution instantanée de l'action a

Exemple : une structure de Kripke

Une exécution :

$$\underset{\text{ok}}{q_0} \to \underset{\text{fault}}{q_1} \to \underset{\text{ok}}{q_0} \to \underset{\text{fault}}{q_1} \to \underset{\text{alarm}}{q_2} \to \underset{\text{ok}}{q_0} \to \dots$$

Systèmes de transitions

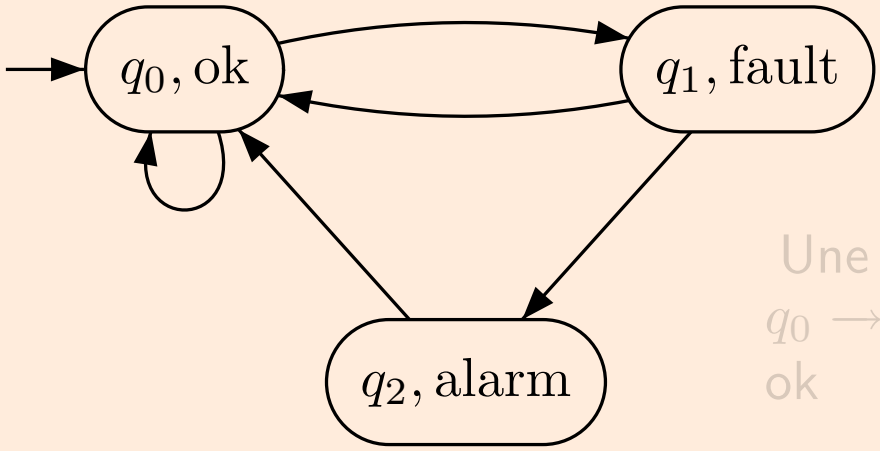
Définition

Act alphabet d'actions,
Prop ensemble de propositions atomiques,
$\mathcal{T} = (S, s_0, L, E)$ système de transitions

- S ensemble de configurations, s_0 configuration initiale,
- $L : S \mapsto 2^{\text{Prop}}$, étiquetage des configurations par des ensembles de propositions atomiques,
- $E \subseteq S \times Act \times S$ contient

  des transitions d'action : $s \xrightarrow{a} s'$, exécution instantanée de l'action a

Exemple : une structure de Kripke



un seul type d'action (omise)

ok, fault, alarm : prop. atomiques

Une exécution :

$q_0 \to q_1 \to q_0 \to q_1 \to q_2 \to q_0 \to \dots$
ok fault ok fault alarm ok

Systèmes de transitions

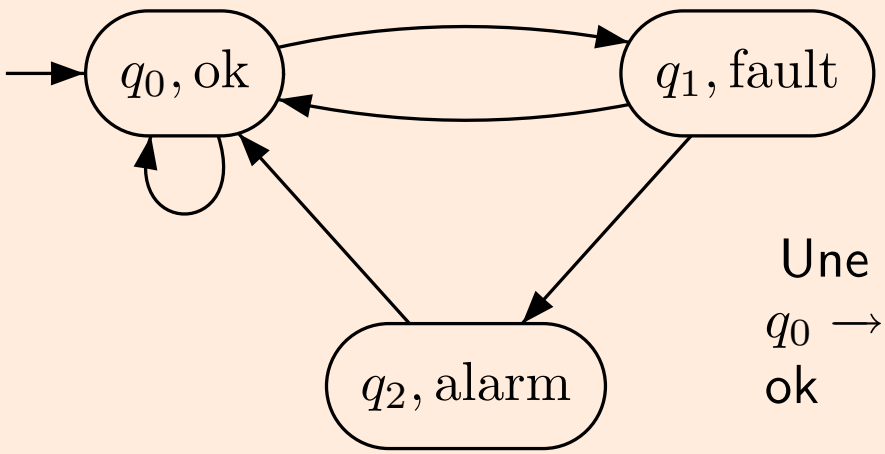
Définition

Act alphabet d'actions,
Prop ensemble de propositions atomiques,
$\mathcal{T} = (S, s_0, L, E)$ système de transitions

- S ensemble de configurations, s_0 configuration initiale,
- $L : S \mapsto 2^{\text{Prop}}$, étiquetage des configurations par des ensembles de propositions atomiques,
- $E \subseteq S \times Act \times S$ contient

  des transitions d'action : $s \xrightarrow{a} s'$, exécution instantanée de l'action a

Exemple : une structure de Kripke



un seul type d'action (omise)

ok, fault, alarm : prop. atomiques

Une exécution :

$q_0 \rightarrow$	$q_1 \rightarrow$	$q_0 \rightarrow$	$q_1 \rightarrow$	$q_2 \rightarrow$	$q_0 \rightarrow \ldots$
ok	fault	ok	fault	alarm	ok

Systèmes de transitions

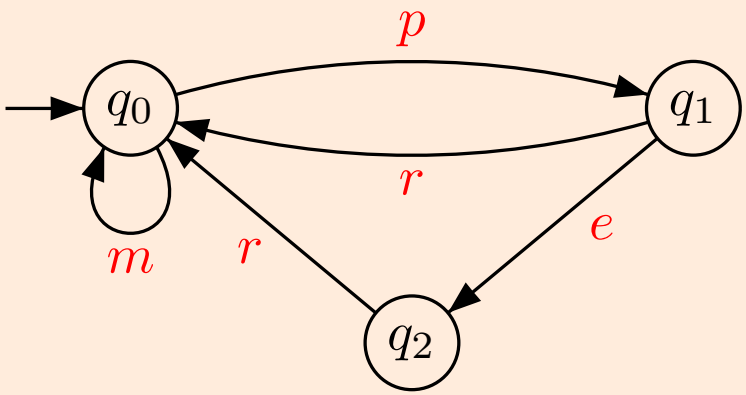
Définition

Act alphabet d'actions,
Prop ensemble de propositions atomiques,
$\mathcal{T} = (S, s_0, L, E)$ système de transitions

- S ensemble de configurations, s_0 configuration initiale,
- $L : S \mapsto 2^{\text{Prop}}$, étiquetage des configurations par des ensembles de propositions atomiques,
- $E \subseteq S \times Act \times S$ contient

  des transitions d'action : $s \xrightarrow{a} s'$, exécution instantanée de l'action a

Exemple : un automate (fini)



prop. atomiques : noms des états

m, p, r, e : actions

Une exécution :

$q_0 \xrightarrow{p} q_1 \xrightarrow{r} q_0 \xrightarrow{p} q_1 \xrightarrow{e} q_2 \xrightarrow{r} q_0 \cdots$

Systèmes de transitions

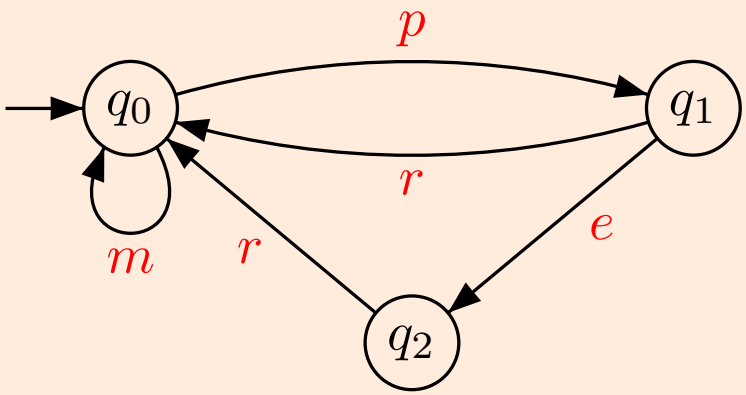
Définition

Act alphabet d'actions,
Prop ensemble de propositions atomiques,
$\mathcal{T} = (S, s_0, L, E)$ système de transitions

- S ensemble de configurations, s_0 configuration initiale,
- $L : S \mapsto 2^{\text{Prop}}$, étiquetage des configurations par des ensembles de propositions atomiques,
- $E \subseteq S \times Act \times S$ contient

  des transitions d'action : $s \xrightarrow{a} s'$, exécution instantanée de l'action a

Exemple : un automate (fini)



prop. atomiques : noms des états

m, p, r, e : actions

Une exécution :

$q_0 \xrightarrow{p} q_1 \xrightarrow{r} q_0 \xrightarrow{p} q_1 \xrightarrow{e} q_2 \xrightarrow{r} q_0 \cdots$

Systèmes de transitions temporisés

Définition

Act alphabet d'actions,
Prop ensemble de propositions atomiques,
$\mathcal{T} = (S, s_0, L, E)$ système de transitions

- S ensemble de configurations, s_0 configuration initiale,
- $L : S \mapsto 2^{\text{Prop}}$, étiquetage des configurations par des ensembles de propositions atomiques,
- $E \subseteq S \times Act \times S$ contient

  des transitions d'action : $s \xrightarrow{a} s'$, exécution instantanée de l'action a

  des transitions de durée : $s \xrightarrow{d} s'$, écoulement d'une durée d.

Systèmes de transitions temporisés

Définition

Act alphabet d'actions, $\mathbb{T}$ domaine de temps plongé dans $\mathbb{R}_{\geq 0}$,
Prop ensemble de propositions atomiques,
$\mathcal{T} = (S, s_0, L, E)$ système de transitions temporisé

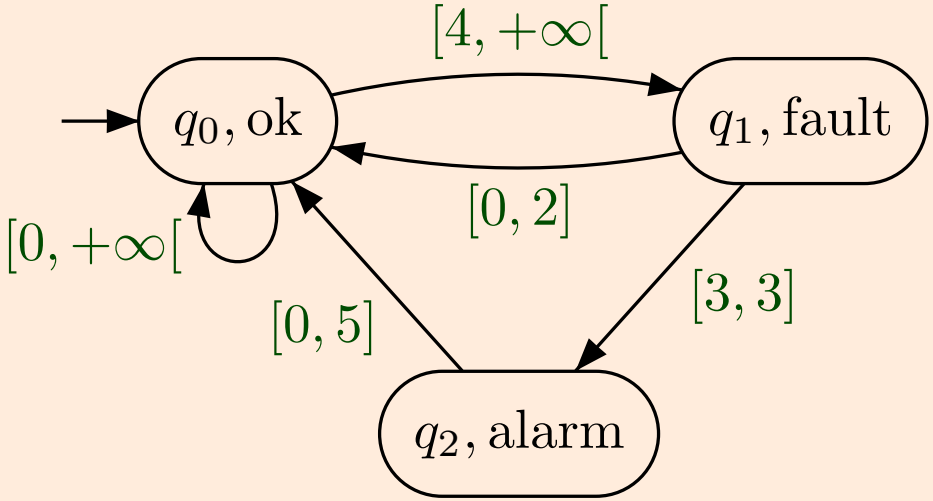
- S ensemble de configurations, s_0 configuration initiale,
- $L : S \mapsto 2^{\text{Prop}}$, étiquetage des configurations par des ensembles de propositions atomiques,
- $E \subseteq S \times (Act \cup \mathbb{T}) \times S$ contient

  des transitions d'action : $s \xrightarrow{a} s'$, exécution instantanée de l'action a

  des transitions de durée : $s \xrightarrow{d} s'$, écoulement d'une durée d.

Exemple 1 : structures de Kripke avec durées

Une variante de [Emerson et al. 1992]



ok, fault, alarm : prop. atomiques

domaine de temps : $\mathbb{N}$

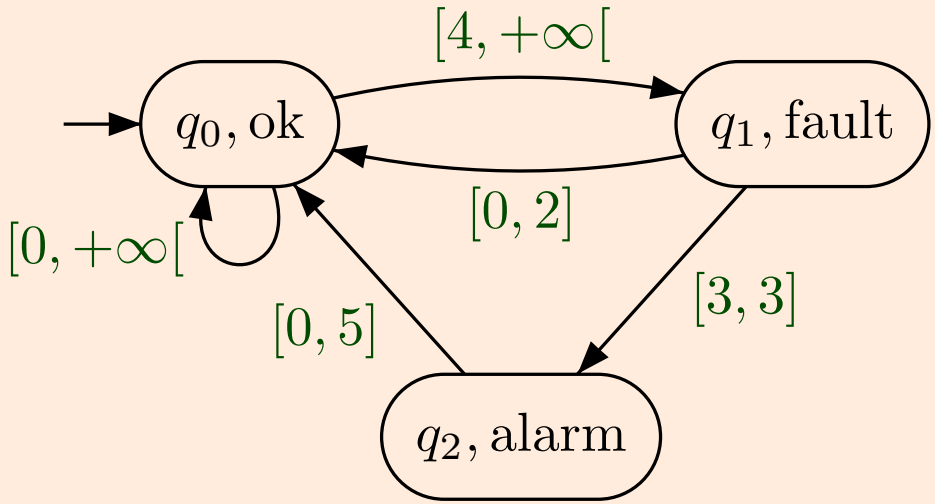
contrainte : un intervalle de $\mathbb{N}$

Une exécution : $q_0 \xrightarrow{15} q_1 \xrightarrow{1} q_0 \xrightarrow{8} q_1 \xrightarrow{3} q_2 \xrightarrow{4} q_0 \cdots$

Rq : uniquement des transitions de durées (Act est vide).

Exemple 1 : structures de Kripke avec durées

Une variante de [Emerson et al. 1992]



ok, fault, alarm : prop. atomiques

domaine de temps : $\mathbb{N}$

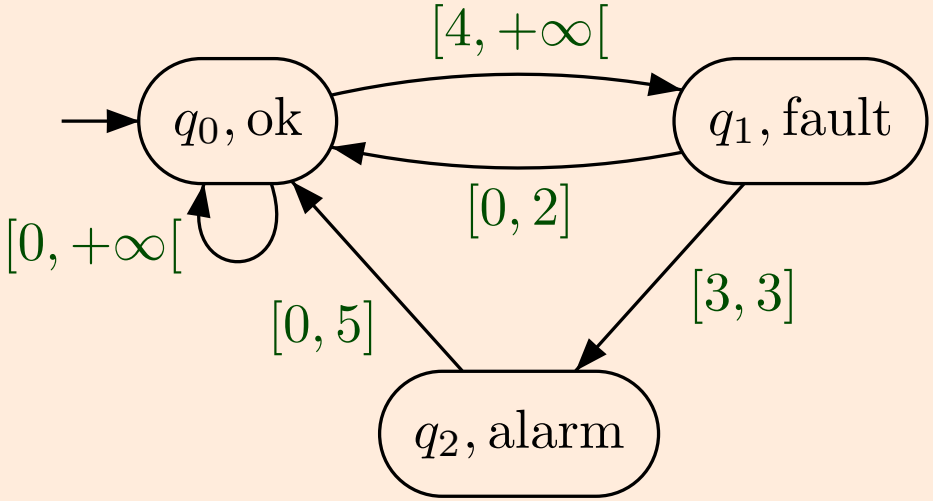
contrainte : un intervalle de $\mathbb{N}$

Une exécution : $q_0 \xrightarrow{15} q_1 \xrightarrow{1} q_0 \xrightarrow{8} q_1 \xrightarrow{3} q_2 \xrightarrow{4} q_0 \cdots$

Rq : uniquement des transitions de durées (Act est vide).

Exemple 1 : structures de Kripke avec durées

Une variante de [Emerson et al. 1992]



ok, fault, alarm : prop. atomiques

domaine de temps : $\mathbb{N}$

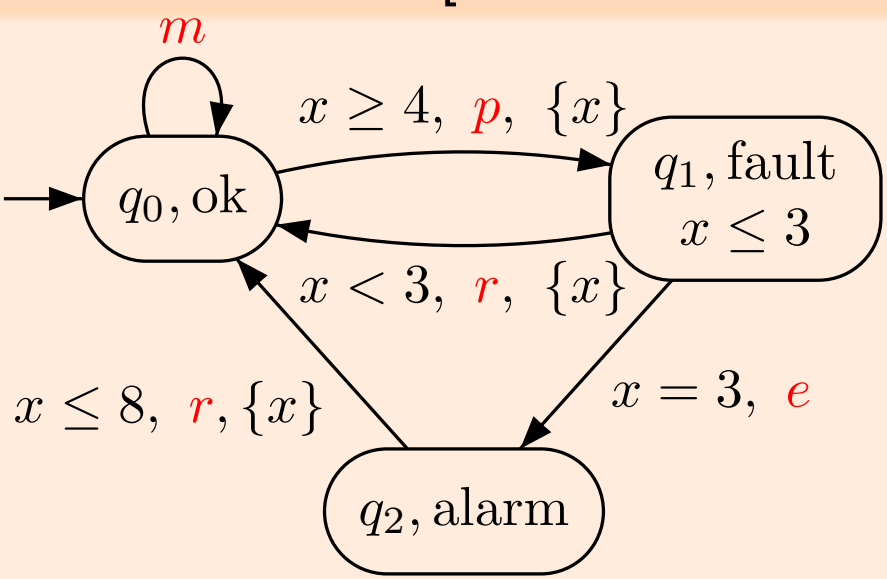
contrainte : un intervalle de $\mathbb{N}$

Une exécution : $q_0 \xrightarrow{15} q_1 \xrightarrow{1} q_0 \xrightarrow{8} q_1 \xrightarrow{3} q_2 \xrightarrow{4} q_0 \cdots$

Rq : uniquement des transitions de durées (Act est vide).

Exemple 2 : automates temporisés

Une variante de [Alur et Dill 1990]



x : horloge à valeurs dans $\mathbb{R}_{\geq 0}$

m, p, r, e : actions

ok, fault, alarm : prop. atomiques

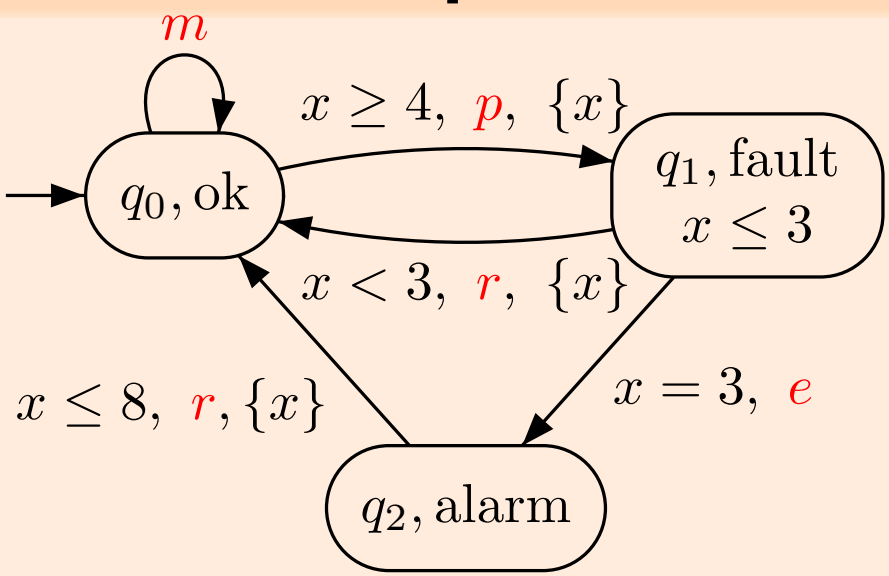
$x < 3$, $x = 3$, $x \geq 4$: gardes

$x \leq 3$: invariant

$\{x\}$: remise à zéro de x

Exemple 2 : automates temporisés

Une variante de [Alur et Dill 1990]



x : horloge à valeurs dans $\mathbb{R}_{\geq 0}$

m, p, r, e : actions

ok, fault, alarm : prop. atomiques

$x < 3$, $x = 3$, $x \geq 4$: gardes

$x \leq 3$: invariant

$\{x\}$: remise à zéro de x

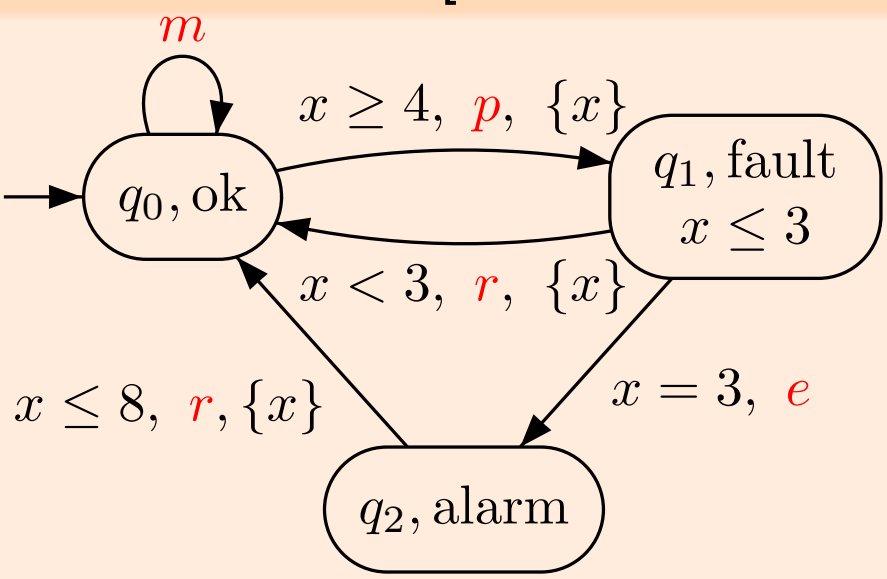
Contraintes et valuations d'horloges

X ensemble d'horloges, valuation $v : X \mapsto \mathbb{R}_{\geq 0}$,

$\mathcal{C}(X)$ ensemble de contraintes d'horloges : conjonctions de contraintes atomiques de la forme $x \bowtie c$, pour une constante c et $\bowtie$ dans $\{<, \leq, =, \geq, >\}$.

Exemple 2 : automates temporisés

Une variante de [Alur et Dill 1990]



x : horloge à valeurs dans $\mathbb{R}_{\geq 0}$

m, p, r, e : actions

ok, fault, alarm : prop. atomiques

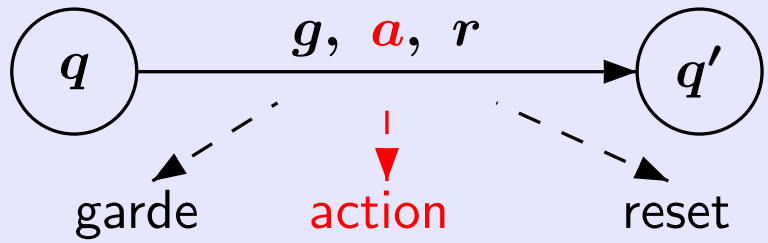
$x < 3$, $x = 3$, $x \geq 4$: gardes

$x \leq 3$: invariant

$\{x\}$: remise à zéro de x

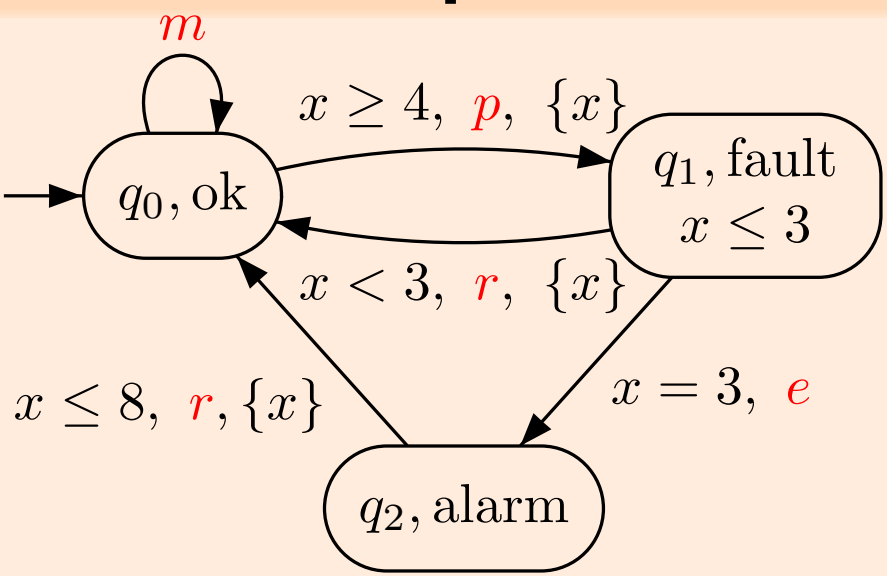
Automate temporisé $\mathcal{A} = (Q, q_0, \ell, Inv, \Delta)$

- Q ensemble d'états, q_0 état initial,
- ℓ fonction d'étiquetage, Inv associe un invariant à chaque état
- Δ contient des transitions :



Exemple 2 : automates temporisés

Une variante de [Alur et Dill 1990]



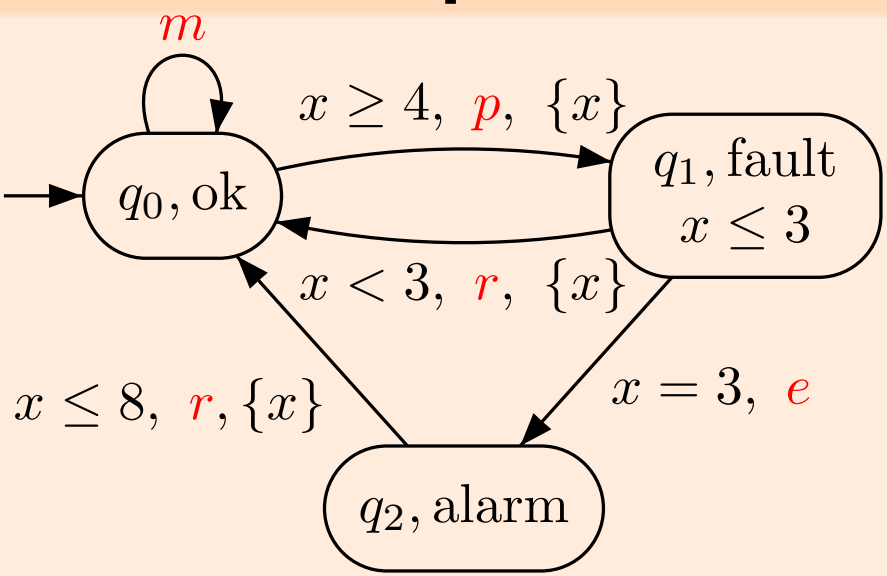
Une exécution : $(q_0, [0]) \xrightarrow{8.3} (q_0, [8.3]) \xrightarrow{p} (q_1, [0]) \xrightarrow{3} (q_1, [3]) \xrightarrow{e} (q_2, [3]) \xrightarrow{2.1} (q_2, [5.1]) \xrightarrow{r} (q_0, [0]) \cdots$

Mot temporisé : $(p, 8.3)(e, 11.3)(r, 13.4) \ldots$

Etiquetage : ok, fault, alarm, ok $\ldots$

Exemple 2 : automates temporisés

Une variante de [Alur et Dill 1990]



Configurations : (q, v)

v valeur de x satifaisant l'invariant

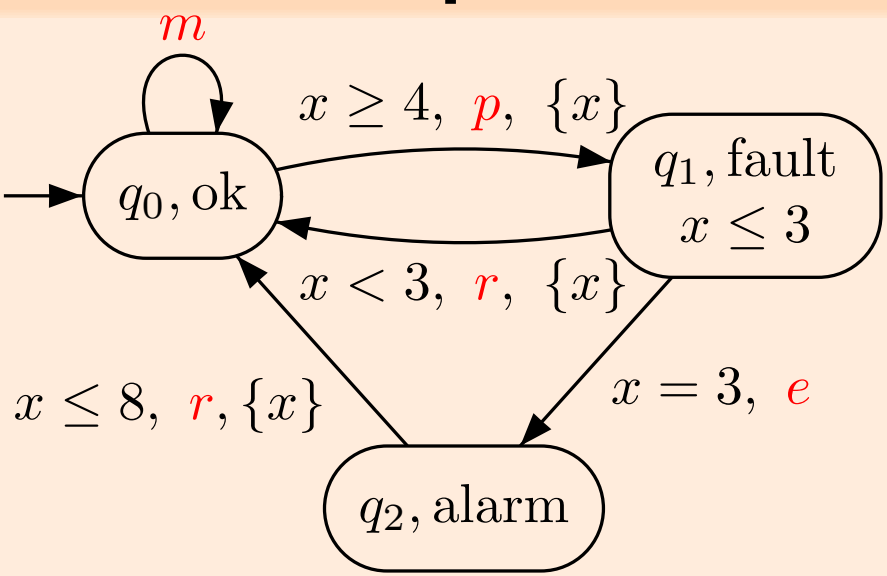
Une exécution : $(q_0, [0]) \xrightarrow{8.3} (q_0, [8.3]) \xrightarrow{p} (q_1, [0]) \xrightarrow{3} (q_1, [3]) \xrightarrow{e} (q_2, [3]) \xrightarrow{2.1} (q_2, [5.1]) \xrightarrow{r} (q_0, [0]) \cdots$

Mot temporisé : $(p, 8.3)(e, 11.3)(r, 13.4) \ldots$

Etiquetage : $\text{ok},\ \text{fault},\ \text{alarm},\ \text{ok} \ldots$

Exemple 2 : automates temporisés

Une variante de [Alur et Dill 1990]



Configurations : (q, v)

v valeur de x satifaisant l'invariant

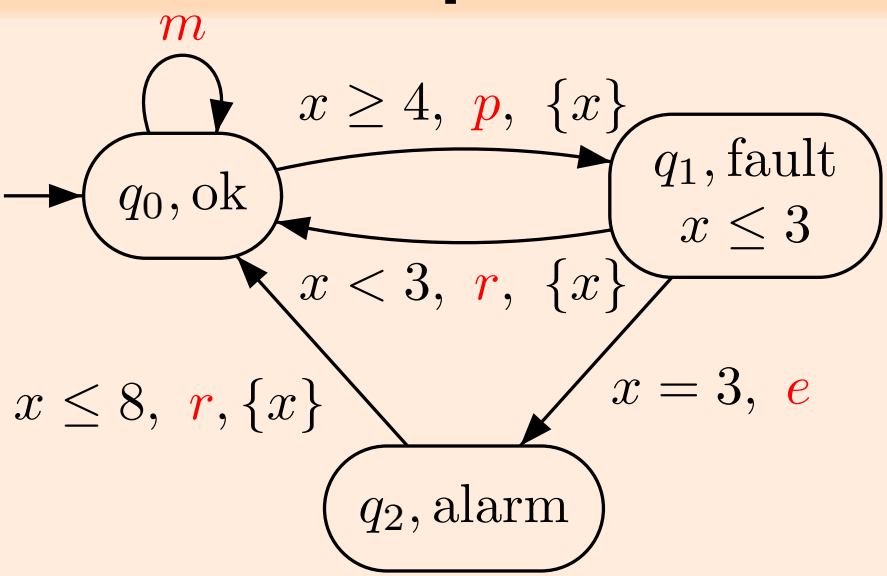
Une exécution : $(q_0, [0]) \xrightarrow{8.3} (q_0, [8.3]) \xrightarrow{p} (q_1, [0]) \xrightarrow{3} (q_1, [3]) \xrightarrow{e} (q_2, [3]) \xrightarrow{2.1} (q_2, [5.1]) \xrightarrow{r} (q_0, [0]) \cdots$

Mot temporisé : $(p, 8.3)(e, 11.3)(r, 13.4) \ldots$

Etiquetage : $\text{ok},\ \text{fault},\ \text{alarm},\ \text{ok} \ldots$

Exemple 2 : automates temporisés

Une variante de [Alur et Dill 1990]



Configurations : (q, v)

v valeur de x satifaisant l'invariant

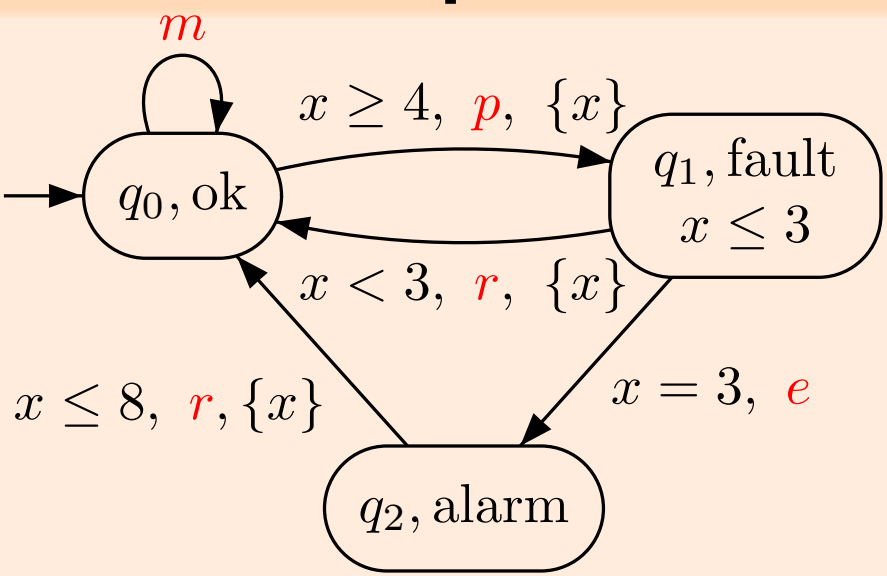
Une exécution : $(q_0, [0]) \xrightarrow{8.3} (q_0, [8.3]) \xrightarrow{p} (q_1, [0]) \xrightarrow{3} (q_1, [3]) \xrightarrow{e} (q_2, [3]) \xrightarrow{2.1} (q_2, [5.1]) \xrightarrow{r} (q_0, [0]) \cdots$

Mot temporisé : $(p, 8.3)(e, 11.3)(r, 13.4) \ldots$

Etiquetage : $\text{ok},\ \text{fault},\ \text{alarm},\ \text{ok} \ldots$

Exemple 2 : automates temporisés

Une variante de [Alur et Dill 1990]



Configurations : (q, v)

v valeur de x satifaisant l'invariant

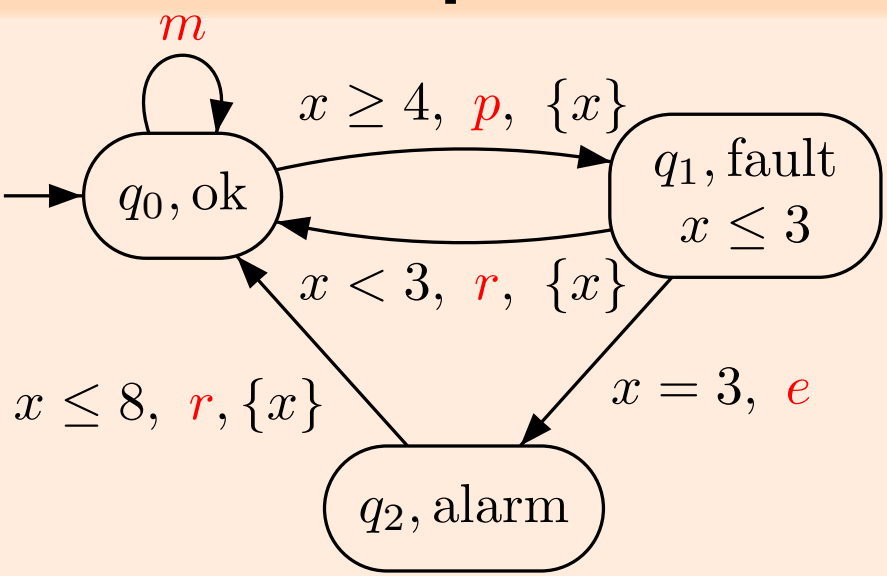
Une exécution : $(q_0, [0]) \xrightarrow{8.3} (q_0, [8.3]) \xrightarrow{p} (q_1, [0]) \xrightarrow{3} (q_1, [3]) \xrightarrow{e} (q_2, [3]) \xrightarrow{2.1} (q_2, [5.1]) \xrightarrow{r} (q_0, [0]) \cdots$

Mot temporisé : $(p, 8.3)(e, 11.3)(r, 13.4) \ldots$

Etiquetage : $\text{ok},\ \text{fault},\ \text{alarm},\ \text{ok} \ldots$

Exemple 2 : automates temporisés

Une variante de [Alur et Dill 1990]



Configurations : (q, v)

v valeur de x satifaisant l'invariant

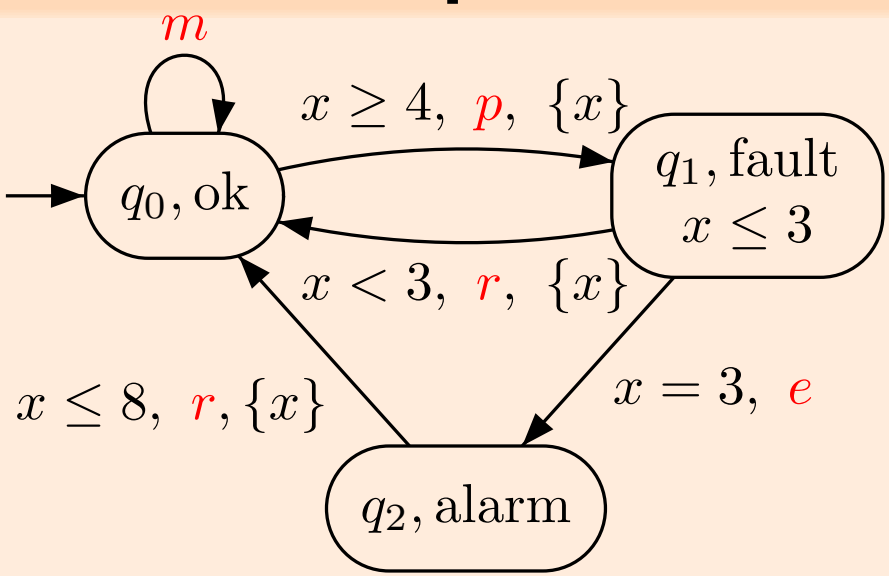
Une exécution : $(q_0, [0]) \xrightarrow{8.3} (q_0, [8.3]) \xrightarrow{p} (q_1, [0]) \xrightarrow{3} (q_1, [3]) \xrightarrow{e} (q_2, [3]) \xrightarrow{2.1} (q_2, [5.1]) \xrightarrow{r} (q_0, [0]) \cdots$

Mot temporisé : $(p, 8.3)(e, 11.3)(r, 13.4) \ldots$

Etiquetage : $\text{ok},\ \text{fault},\ \text{alarm},\ \text{ok} \ldots$

Exemple 2 : automates temporisés

Une variante de [Alur et Dill 1990]



Configurations : (q, v)

v valeur de x satifaisant l'invariant

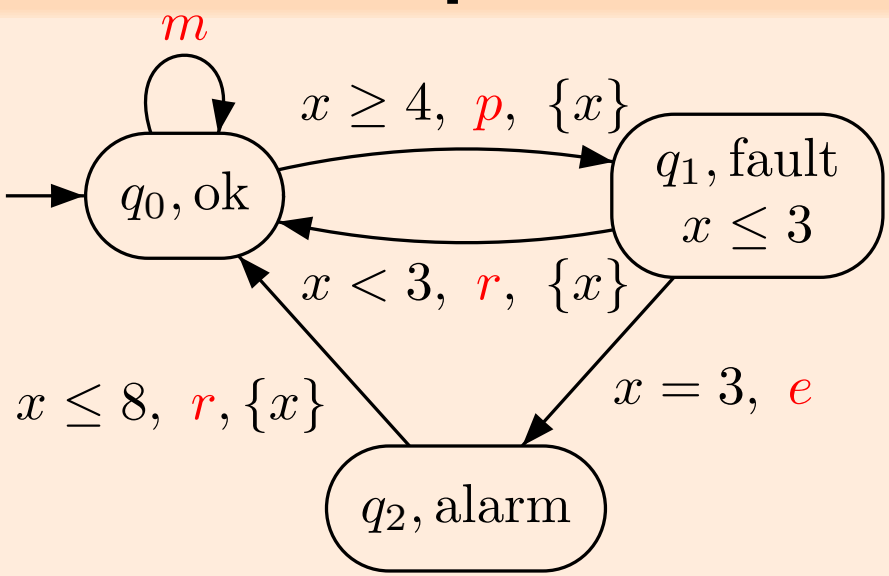
Une exécution : $(q_0, [0]) \xrightarrow{8.3} (q_0, [8.3]) \xrightarrow{p} (q_1, [0]) \xrightarrow{3} (q_1, [3]) \xrightarrow{e} (q_2, [3]) \xrightarrow{2.1} (q_2, [5.1]) \xrightarrow{r} (q_0, [0]) \cdots$

Mot temporisé : $(p, 8.3)(e, 11.3)(r, 13.4) \ldots$

Etiquetage : $\text{ok},\ \text{fault},\ \text{alarm},\ \text{ok} \ldots$

Exemple 2 : automates temporisés

Une variante de [Alur et Dill 1990]



Configurations : (q, v)

v valeur de x satifaisant l'invariant

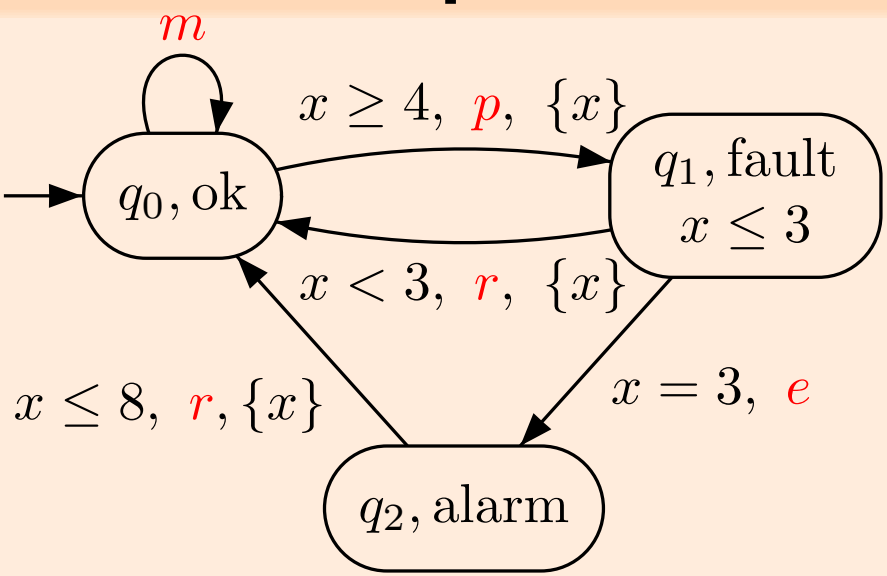
Une exécution : $(q_0, [0]) \xrightarrow{8.3} (q_0, [8.3]) \xrightarrow{p} (q_1, [0]) \xrightarrow{3} (q_1, [3]) \xrightarrow{e} (q_2, [3]) \xrightarrow{2.1} (q_2, [5.1]) \xrightarrow{r} (q_0, [0]) \dots$

Mot temporisé : $(p, 8.3)(e, 11.3)(r, 13.4) \dots$

Etiquetage : ok, fault, alarm, ok $\dots$

Exemple 2 : automates temporisés

Une variante de [Alur et Dill 1990]



Configurations : (q, v)

v valeur de x satifaisant l'invariant

Une exécution : $(q_0, [0]) \xrightarrow{8.3} (q_0, [8.3]) \xrightarrow{p} (q_1, [0]) \xrightarrow{3} (q_1, [3]) \xrightarrow{e} (q_2, [3]) \xrightarrow{2.1} (q_2, [5.1]) \xrightarrow{r} (q_0, [0]) \dots$

Mot temporisé : $(p, 8.3)(e, 11.3)(r, 13.4) \dots$

Etiquetage : $\text{ok},\ \text{fault},\ \text{alarm},\ \text{ok} \dots$

Sémantique d'un automate temporisé (1)

Opérations sur les valuations

X ensemble d'horloges. Pour une valuation v :

- pour un sous-ensemble r de X, on obtient $v[r \mapsto 0]$ en remettant à 0 les horloges de r et en conservant les autres valeurs,
- pour une durée d, on obtient $v + d$ en ajoutant d à toutes les valeurs d'horloges.

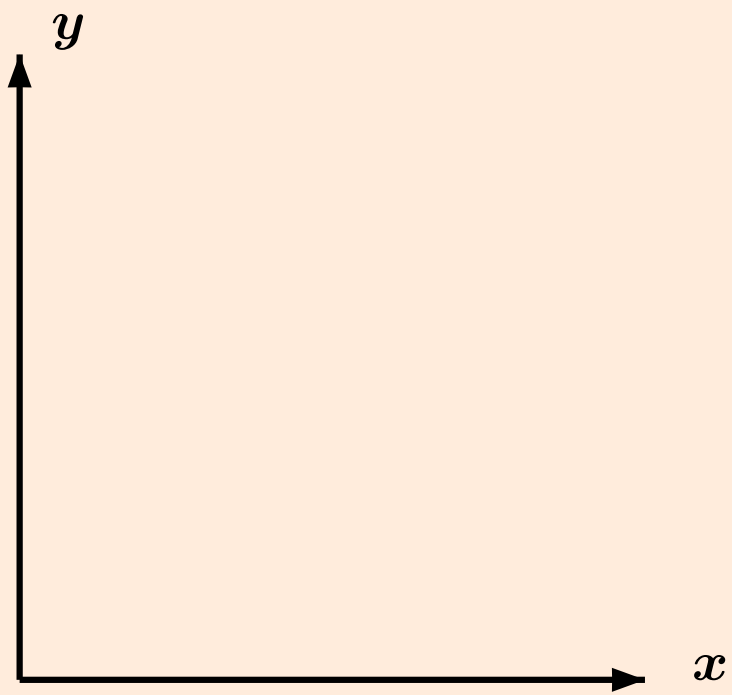
Sémantique d'un automate temporisé (1)

Opérations sur les valuations

X ensemble d'horloges. Pour une valuation v :

- pour un sous-ensemble r de X, on obtient $v[r \mapsto 0]$ en remettant à 0 les horloges de r et en conservant les autres valeurs,
- pour une durée d, on obtient $v + d$ en ajoutant d à toutes les valeurs d'horloges.

Vue géométrique avec deux horloges x et y



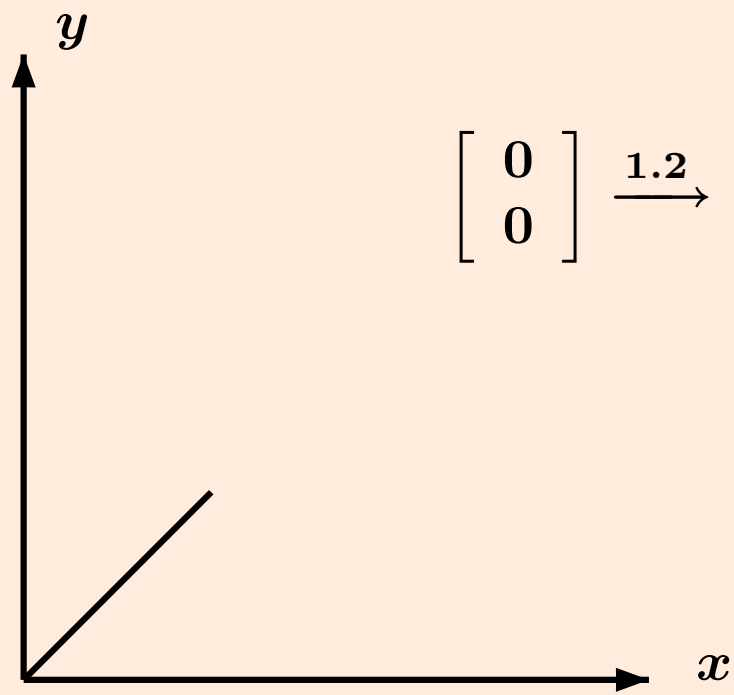
Sémantique d'un automate temporisé (1)

Opérations sur les valuations

X ensemble d'horloges. Pour une valuation v :

- pour un sous-ensemble r de X, on obtient $v[r \mapsto 0]$ en remettant à 0 les horloges de r et en conservant les autres valeurs,
- pour une durée d, on obtient $v + d$ en ajoutant d à toutes les valeurs d'horloges.

Vue géométrique avec deux horloges x et y



$$\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix} \xrightarrow{\mathbf{1.2}} \begin{bmatrix} \mathbf{1.2} \\ \mathbf{1.2} \end{bmatrix}$$

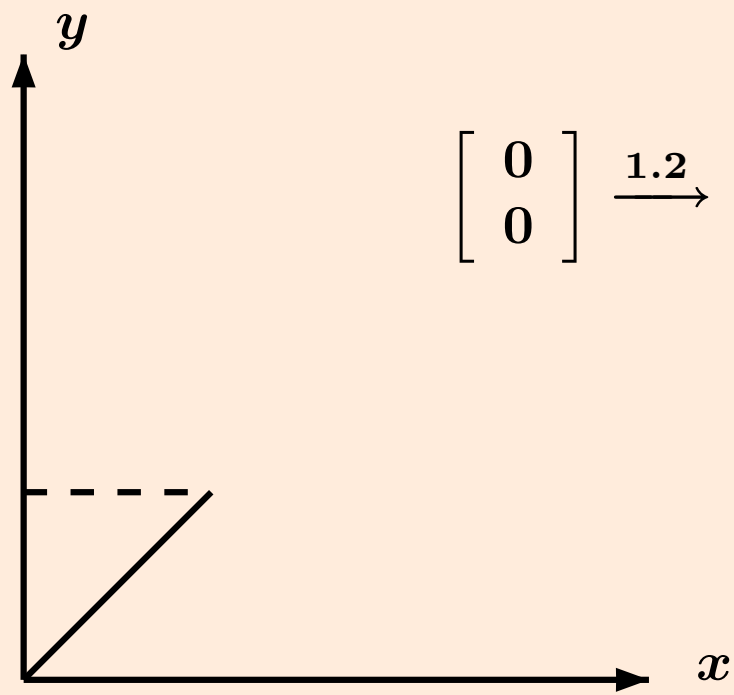
Sémantique d'un automate temporisé (1)

Opérations sur les valuations

X ensemble d'horloges. Pour une valuation v :

- pour un sous-ensemble r de X, on obtient $v[r \mapsto 0]$ en remettant à 0 les horloges de r et en conservant les autres valeurs,
- pour une durée d, on obtient $v + d$ en ajoutant d à toutes les valeurs d'horloges.

Vue géométrique avec deux horloges x et y



$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} \xrightarrow{1.2} \begin{bmatrix} 1.2 \\ 1.2 \end{bmatrix} \xrightarrow{x:=0} \begin{bmatrix} 0 \\ 1.2 \end{bmatrix}$$

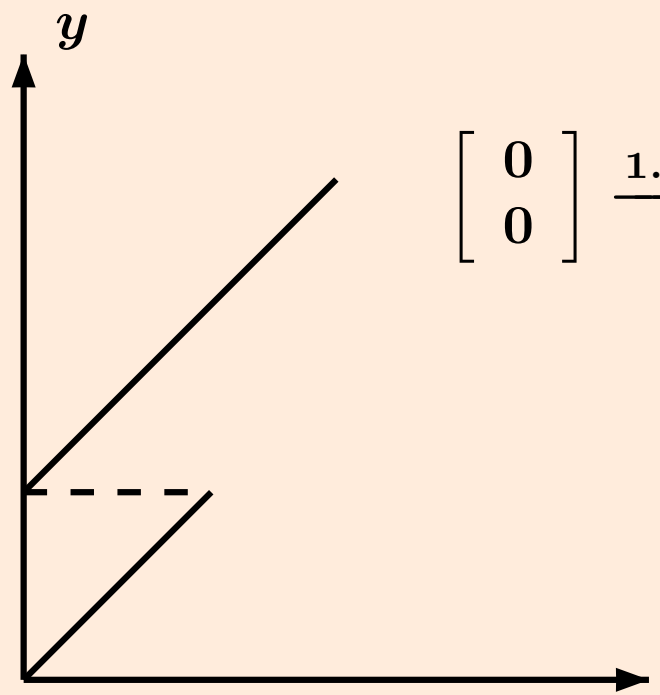
Sémantique d'un automate temporisé (1)

Opérations sur les valuations

X ensemble d'horloges. Pour une valuation v :

- pour un sous-ensemble r de X, on obtient $v[r \mapsto 0]$ en remettant à 0 les horloges de r et en conservant les autres valeurs,
- pour une durée d, on obtient $v + d$ en ajoutant d à toutes les valeurs d'horloges.

Vue géométrique avec deux horloges x et y



$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} \xrightarrow{1.2} \begin{bmatrix} 1.2 \\ 1.2 \end{bmatrix} \xrightarrow{x:=0} \begin{bmatrix} 0 \\ 1.2 \end{bmatrix} \xrightarrow{2} \begin{bmatrix} 2 \\ 3.2 \end{bmatrix}$$

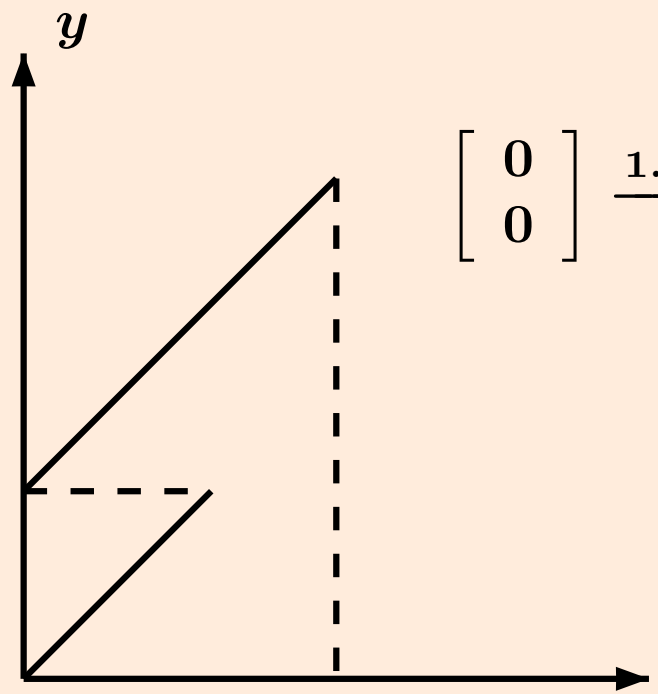
Sémantique d'un automate temporisé (1)

Opérations sur les valuations

X ensemble d'horloges. Pour une valuation v :

- pour un sous-ensemble r de X, on obtient $v[r \mapsto 0]$ en remettant à 0 les horloges de r et en conservant les autres valeurs,
- pour une durée d, on obtient $v + d$ en ajoutant d à toutes les valeurs d'horloges.

Vue géométrique avec deux horloges x et y



$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} \xrightarrow{1.2} \begin{bmatrix} 1.2 \\ 1.2 \end{bmatrix} \xrightarrow{x:=0} \begin{bmatrix} 0 \\ 1.2 \end{bmatrix} \xrightarrow{2} \begin{bmatrix} 2 \\ 3.2 \end{bmatrix} \xrightarrow{y:=0} \begin{bmatrix} 2 \\ 0 \end{bmatrix}$$

Sémantique d'un automate temporisé (2)

Définition

Pour un automate temporisé $\mathcal{A} = (Q, q_0, \ell, Inv, \Delta)$, le système de transitions est $\mathcal{T} = (S, s_0, L, E)$ avec :

- ensemble de configurations $S = \{(q, v) \in Q \times \mathbb{R}_{\geq 0} \mid v \models Inv(q)\}$,
- configuration initiale $s_0 = (q_0, \mathbf{0})$,
- étiquetage $L(q, v) = \ell(q)$
- transitions
  d'action $(q, v) \xrightarrow{a} (q', v')$, s'il existe une transition $q \xrightarrow{g,a,r} q'$ de l'automate telle que $v \models g$ et $v' \models Inv(q')$, avec $v' = v[r \mapsto 0]$,
  de durée $(q, v) \xrightarrow{d} (q, v + d)$ si $v + d \models Inv(q)$.

Sémantique d'un automate temporisé (2)

Définition

Pour un automate temporisé $\mathcal{A} = (Q, q_0, \ell, Inv, \Delta)$, le système de transitions est $\mathcal{T} = (S, s_0, L, E)$ avec :

- ensemble de configurations $S = \{(q, v) \in Q \times \mathbb{R}_{\geq 0} \mid v \models Inv(q)\}$,
- configuration initiale $s_0 = (q_0, \mathbf{0})$,
- étiquetage $L(q, v) = \ell(q)$
- transitions
  d'action $(\boldsymbol{q}, \boldsymbol{v}) \xrightarrow{\boldsymbol{a}} (\boldsymbol{q'}, \boldsymbol{v'})$, s'il existe une transition $q \xrightarrow{g,a,r} q'$ de l'automate telle que $v \models g$ et $v' \models Inv(q')$, avec $v' = v[r \mapsto 0]$,
  de durée $(\boldsymbol{q}, \boldsymbol{v}) \xrightarrow{\boldsymbol{d}} (\boldsymbol{q}, \boldsymbol{v} + \boldsymbol{d})$ si $v + d \models Inv(q)$.

Sémantique d'un automate temporisé (2)

Définition

Pour un automate temporisé $\mathcal{A} = (Q, q_0, \ell, Inv, \Delta)$, le système de transitions est $\mathcal{T} = (S, s_0, L, E)$ avec :

- ensemble de configurations $S = \{(q, v) \in Q \times \mathbb{R}_{\geq 0} \mid v \models Inv(q)\}$,
- configuration initiale $s_0 = (q_0, \mathbf{0})$,
- étiquetage $L(q, v) = \ell(q)$
- transitions
  d'action $(\boldsymbol{q}, \boldsymbol{v}) \xrightarrow{\boldsymbol{a}} (\boldsymbol{q'}, \boldsymbol{v'})$, s'il existe une transition $q \xrightarrow{g,a,r} q'$ de l'automate telle que $v \models g$ et $v' \models Inv(q')$, avec $v' = v[r \mapsto 0]$,
  de durée $(\boldsymbol{q}, \boldsymbol{v}) \xrightarrow{\boldsymbol{d}} (\boldsymbol{q}, \boldsymbol{v} + \boldsymbol{d})$ si $v + d \models Inv(q)$.

Plan

Exemple : temps de réponse

Logiques temporelles

Tout signal est suivi d'une réponse

en CTL : $AG(\text{signal} \Rightarrow AF\ \text{reponse})$

en LTL : $G(\text{signal} \Rightarrow F\ \text{reponse})$

CTL et LTL, deux sous-classes incomparables de CTL*

Logiques temporisées

Tout signal est suivi d'une réponse en moins de 5 unités de temps

Comment étendre les modalités des logiques temporelles avec un temps explicite ?

Exemple : temps de réponse

Logiques temporelles

Tout signal est suivi d'une réponse

en CTL : $\mathsf{AG}(\text{signal} \Rightarrow \mathsf{AF}\ \text{reponse})$

en LTL : $\mathsf{G}(\text{signal} \Rightarrow \mathsf{F}\ \text{reponse})$

CTL et LTL, deux sous-classes incomparables de CTL*

Logiques temporisées

Tout signal est suivi d'une réponse en moins de 5 unités de temps

Comment étendre les modalités des logiques temporelles avec un temps explicite ?

Exemple : temps de réponse

Logiques temporelles

Tout signal est suivi d'une réponse

en CTL : $\mathsf{AG}(\text{signal} \Rightarrow \mathsf{AF}\ \text{reponse})$

en LTL : $\mathsf{G}(\text{signal} \Rightarrow \mathsf{F}\ \text{reponse})$

CTL et LTL, deux sous-classes incomparables de CTL*

Logiques temporisées

Tout signal est suivi d'une réponse en moins de 5 unités de temps

Comment étendre les modalités des logiques temporelles avec un temps explicite ?

Exemple : temps de réponse

Logiques temporelles

Tout signal est suivi d'une réponse

en CTL : $\mathsf{AG}(\text{signal} \Rightarrow \mathsf{AF}\ \text{reponse})$

en LTL : $\mathsf{G}(\text{signal} \Rightarrow \mathsf{F}\ \text{reponse})$

CTL et LTL, deux sous-classes incomparables de CTL*

Logiques temporisées

Tout signal est suivi d'une réponse en moins de 5 unités de temps

Comment étendre les modalités des logiques temporelles avec un temps explicite ?

Exemple : temps de réponse

Logiques temporelles

Tout signal est suivi d'une réponse

en CTL : $AG(\text{signal} \Rightarrow AF\ \text{reponse})$

en LTL : $G(\text{signal} \Rightarrow F\ \text{reponse})$

CTL et LTL, deux sous-classes incomparables de CTL*

Logiques temporisées

Tout signal est suivi d'une réponse en moins de 5 unités de temps

Comment étendre les modalités des logiques temporelles avec un temps explicite ?

Exemple : temps de réponse

Logiques temporelles

Tout signal est suivi d'une réponse

en CTL : $\mathsf{AG}(\text{signal} \Rightarrow \mathsf{AF}\ \text{reponse})$

en LTL : $\mathsf{G}(\text{signal} \Rightarrow \mathsf{F}\ \text{reponse})$

CTL et LTL, deux sous-classes incomparables de CTL*

Logiques temporisées

Tout signal est suivi d'une réponse en moins de 5 unités de temps

Comment étendre les modalités des logiques temporelles avec un temps explicite ?

CTL

Syntaxe de CTL

$$\varphi, \psi ::= P \mid \neg\varphi \mid \mathsf{EX}\varphi \mid \mathsf{AX}\varphi \mid \varphi \wedge \psi \mid \mathsf{E}\varphi\mathsf{U}\psi \mid \mathsf{A}\varphi\mathsf{U}\psi$$

où P est une proposition atomique dans Prop.

Interprétation : sur les états d'une structure de Kripke

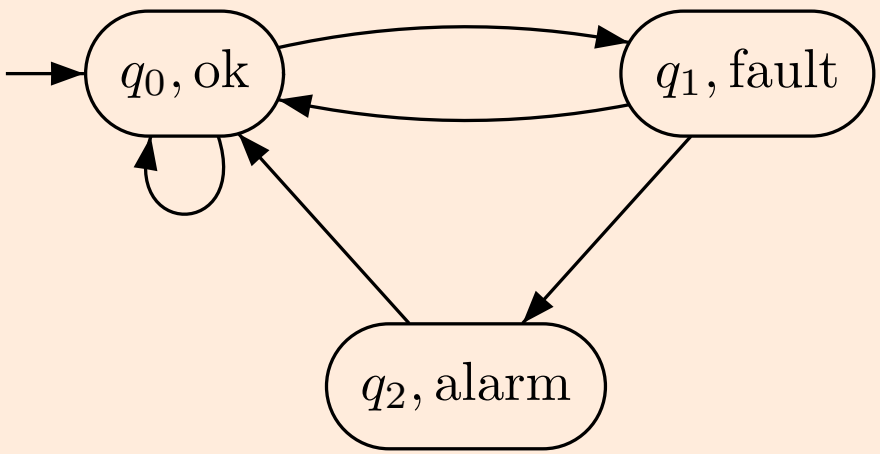
CTL

Syntaxe de CTL

$$\varphi, \psi ::= P \mid \neg\varphi \mid \mathsf{EX}\varphi \mid \mathsf{AX}\varphi \mid \varphi \wedge \psi \mid \mathsf{E}\varphi\mathsf{U}\psi \mid \mathsf{A}\varphi\mathsf{U}\psi$$

où P est une proposition atomique dans Prop.

Interprétation : sur les états d'une structure de Kripke



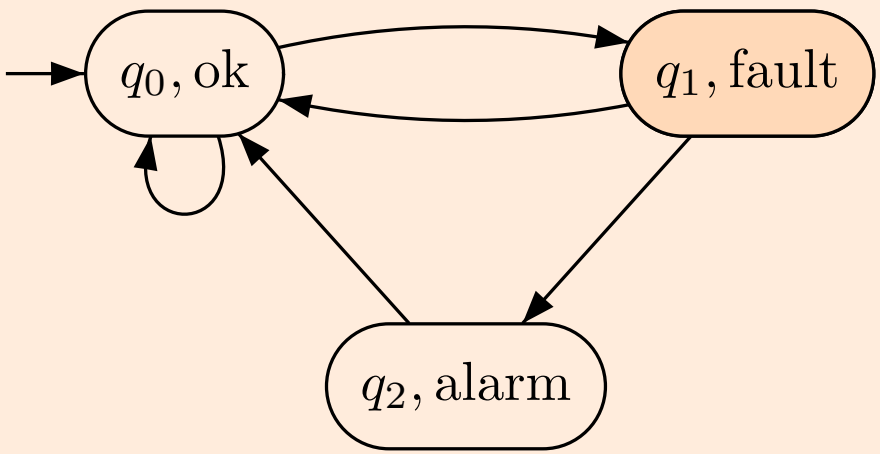
CTL

Syntaxe de CTL

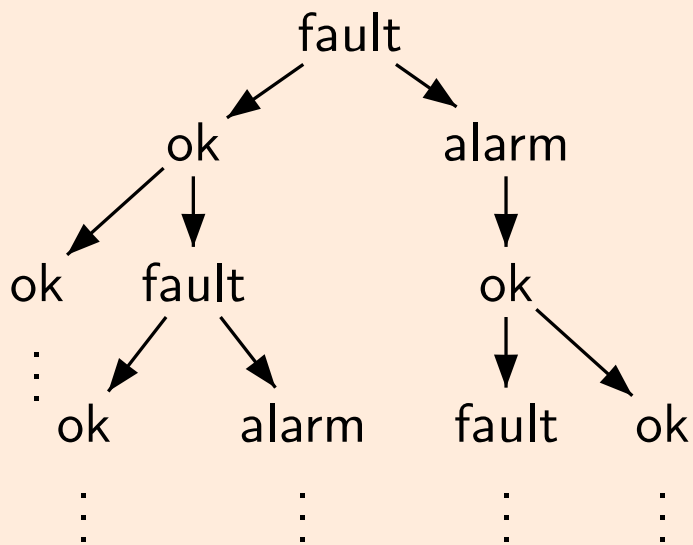
$$\varphi, \psi ::= P \mid \neg\varphi \mid \mathsf{EX}\varphi \mid \mathsf{AX}\varphi \mid \varphi \wedge \psi \mid \mathsf{E}\varphi\mathsf{U}\psi \mid \mathsf{A}\varphi\mathsf{U}\psi$$

où P est une proposition atomique dans Prop.

Interprétation : sur les états d'une structure de Kripke



avec un arbre d'exécution



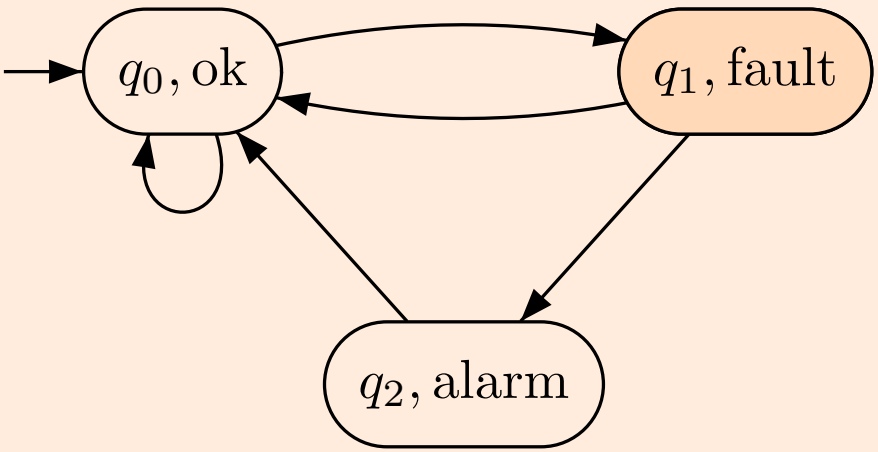
CTL

Syntaxe de CTL

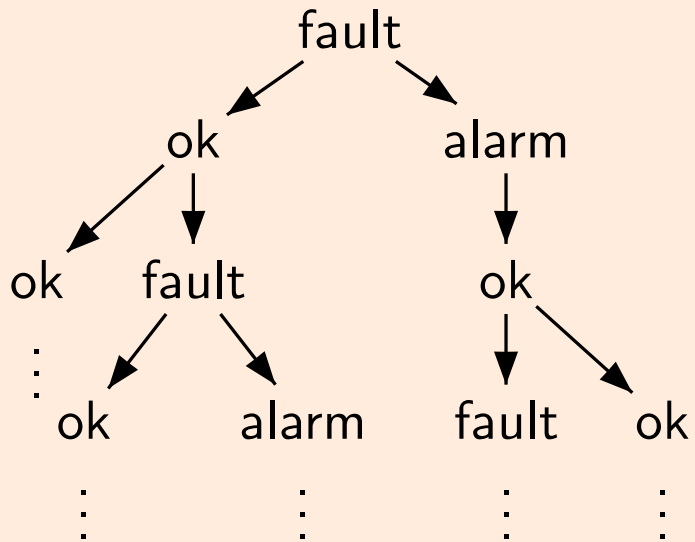
$$\varphi, \psi ::= P \mid \neg\varphi \mid \mathsf{EX}\varphi \mid \mathsf{AX}\varphi \mid \varphi \wedge \psi \mid \mathsf{E}\varphi\mathsf{U}\psi \mid \mathsf{A}\varphi\mathsf{U}\psi$$

où P est une proposition atomique dans Prop.

Interprétation : sur les états d'une structure de Kripke



avec un arbre d'exécution



E : il existe un chemin

X : position suivante (next)

$q_1 \models \mathsf{EX}\mathrm{ok}$

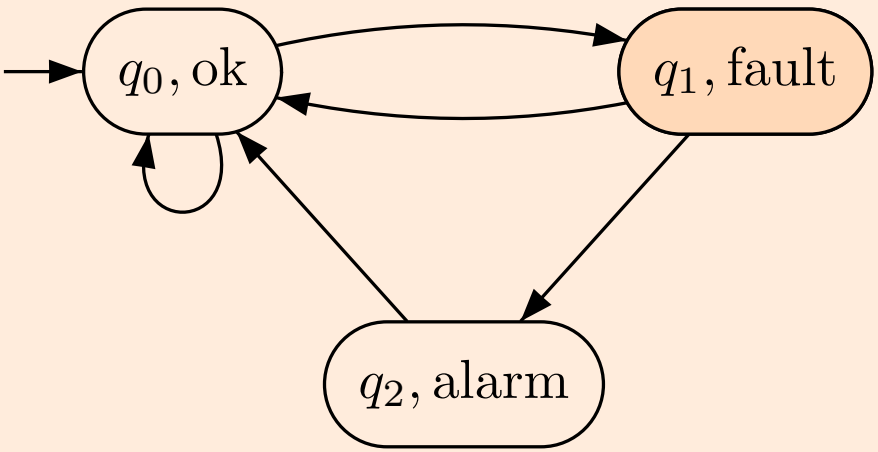
CTL

Syntaxe de CTL

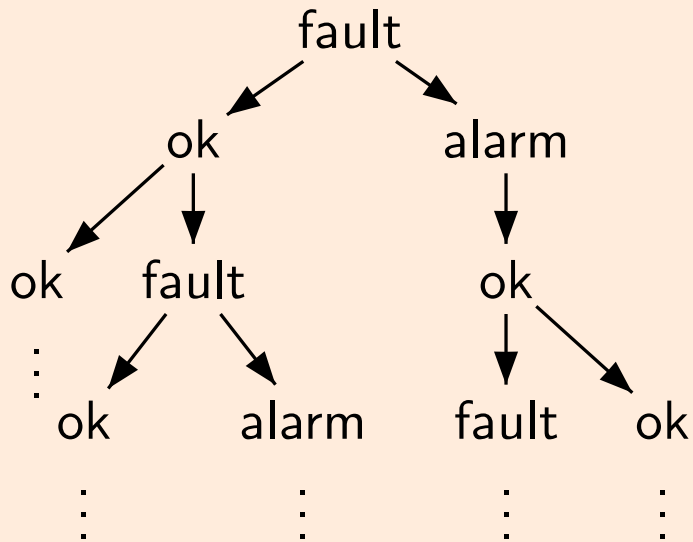
$$\varphi, \psi ::= P \mid \neg\varphi \mid \mathsf{EX}\varphi \mid \mathsf{AX}\varphi \mid \varphi \wedge \psi \mid \mathsf{E}\varphi\mathsf{U}\psi \mid \mathsf{A}\varphi\mathsf{U}\psi$$

où P est une proposition atomique dans Prop.

Interprétation : sur les états d'une structure de Kripke



avec un arbre d'exécution



A : pour tout chemin

$\varphi\mathsf{U}\psi$: until

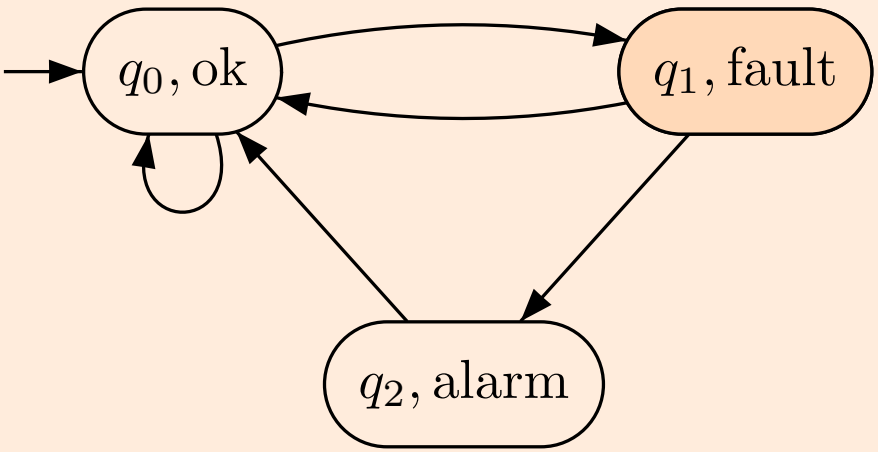
CTL

Syntaxe de CTL

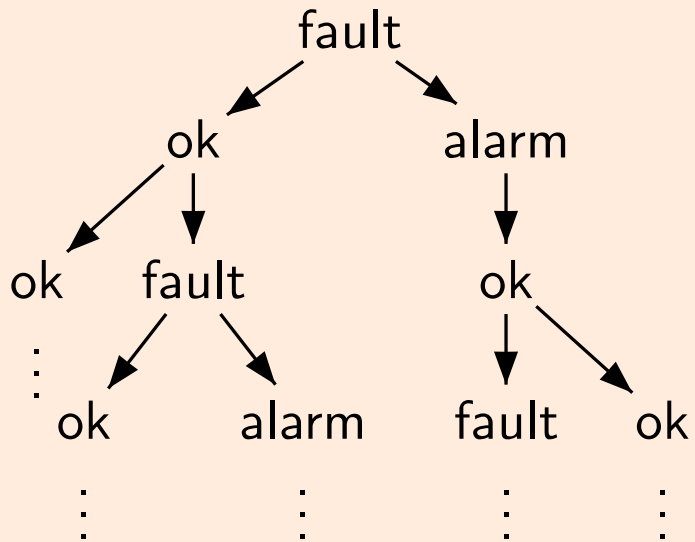
$$\varphi, \psi ::= P \mid \neg\varphi \mid \mathsf{EX}\varphi \mid \mathsf{AX}\varphi \mid \varphi \wedge \psi \mid \mathsf{E}\varphi\mathsf{U}\psi \mid \mathsf{A}\varphi\mathsf{U}\psi$$

où P est une proposition atomique dans Prop.

Interprétation : sur les états d'une structure de Kripke



avec un arbre d'exécution



Abréviations :

AFψ pour A $true$ Uψ

$q_1 \models \mathsf{AF}\mathrm{ok}$

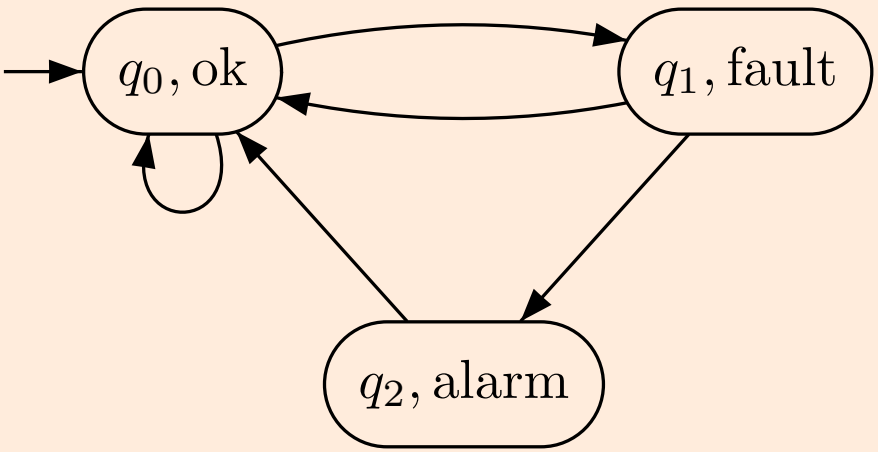
CTL

Syntaxe de CTL

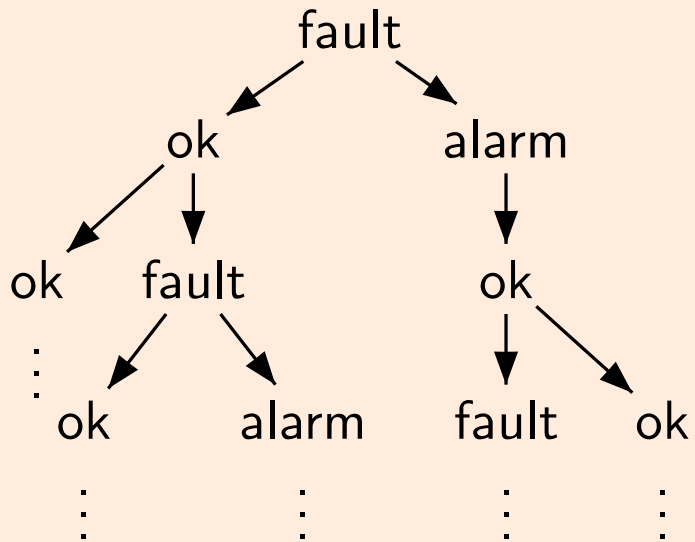
$$\varphi, \psi ::= P \mid \neg\varphi \mid \mathsf{EX}\varphi \mid \mathsf{AX}\varphi \mid \varphi \wedge \psi \mid \mathsf{E}\varphi\mathsf{U}\psi \mid \mathsf{A}\varphi\mathsf{U}\psi$$

où P est une proposition atomique dans Prop.

Interprétation : sur les états d'une structure de Kripke



avec un arbre d'exécution



Abréviations :

- EFψ pour E $true$ Uψ
- AGψ pour $\neg$EF($\neg\varphi$)

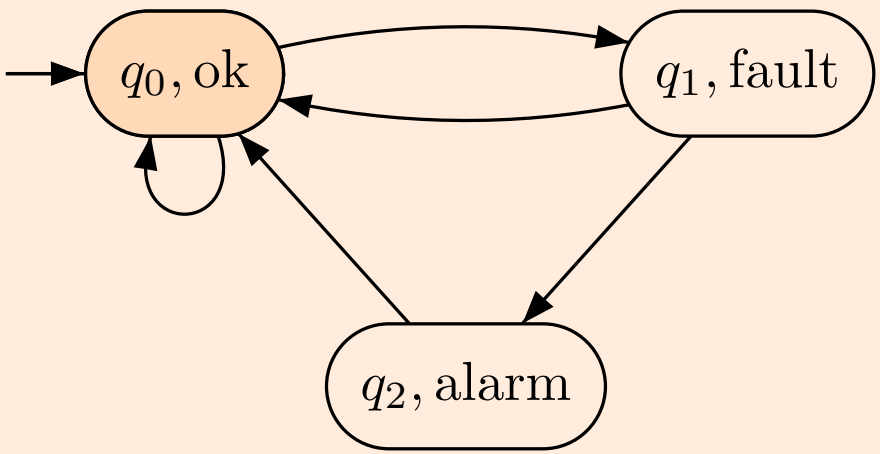
CTL

Syntaxe de CTL

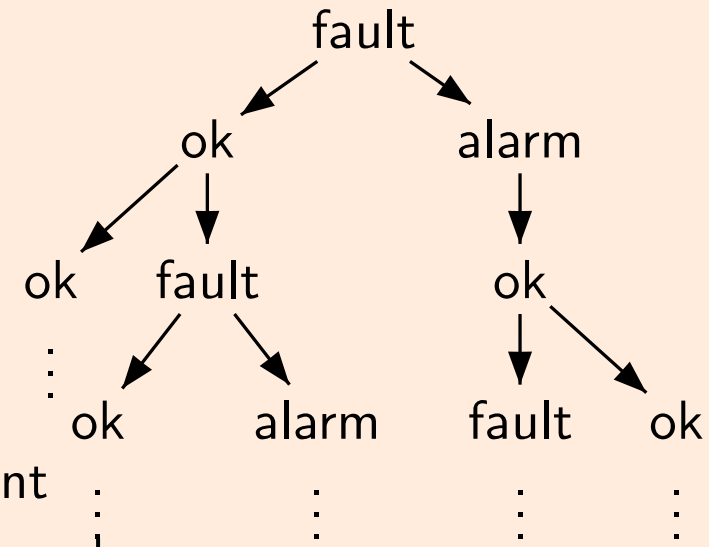
$$\varphi, \psi ::= P \mid \neg\varphi \mid \mathsf{EX}\varphi \mid \mathsf{AX}\varphi \mid \varphi \wedge \psi \mid \mathsf{E}\varphi\mathsf{U}\psi \mid \mathsf{A}\varphi\mathsf{U}\psi$$

où P est une proposition atomique dans Prop.

Interprétation : sur les états d'une structure de Kripke



avec un arbre d'exécution



$q_0 \models \mathsf{AG}(\mathsf{AF}\mathrm{ok})$

de tout état, sur tout chemin partant de cet état, il est possible d'atteindre ok

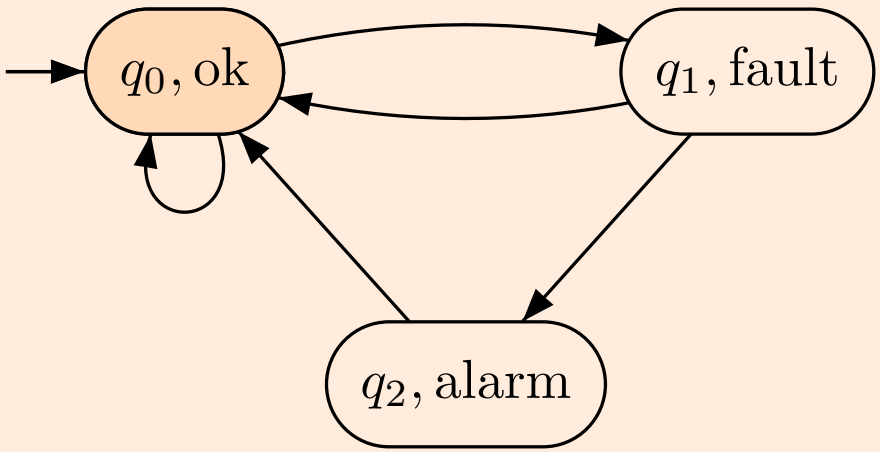
CTL

Syntaxe de CTL

$$\varphi, \psi ::= P \mid \neg\varphi \mid \mathsf{EX}\varphi \mid \mathsf{AX}\varphi \mid \varphi \wedge \psi \mid \mathsf{E}\varphi\mathsf{U}\psi \mid \mathsf{A}\varphi\mathsf{U}\psi$$

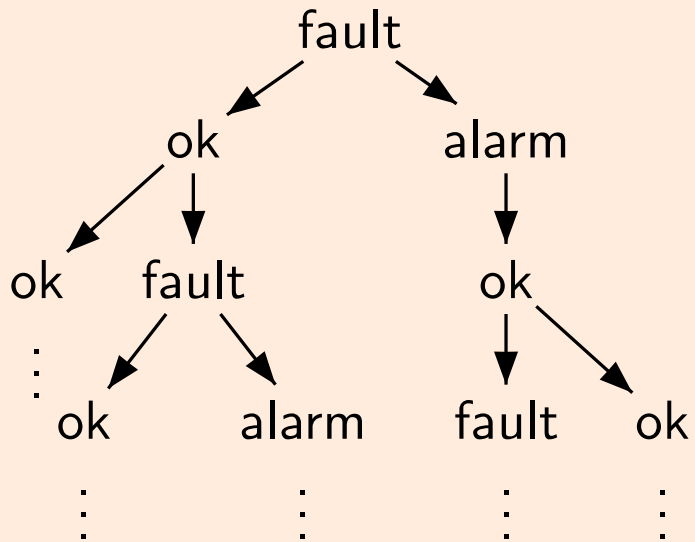
où P est une proposition atomique dans Prop.

Interprétation : sur les états d'une structure de Kripke



$\mathcal{K} \models \mathsf{AG}(\mathsf{AF}\mathrm{ok})$

avec un arbre d'exécution



CTL+temps : TCTL

Syntaxe de TCTL [Alur - Henzinger 1991]

$$\varphi, \psi ::= P \mid \neg\varphi \mid \varphi \wedge \psi \mid \mathsf{E}\varphi\mathsf{U}_{\bowtie c}\psi \mid \mathsf{A}\varphi\mathsf{U}_{\bowtie c}\psi$$

où P est une proposition atomique dans Prop, c est une constante dans $\mathbb{N}$ et l'opérateur $\bowtie$ est dans $\{<, >, \leq, \geq, =\}$.

Interprétation : sur des configurations d'un STT

CTL+temps : TCTL

Syntaxe de TCTL [Alur - Henzinger 1991]

$$\varphi, \psi ::= P \mid \neg\varphi \mid \varphi \wedge \psi \mid \mathsf{E}\varphi\mathsf{U}_{\bowtie c}\psi \mid \mathsf{A}\varphi\mathsf{U}_{\bowtie c}\psi$$

où P est une proposition atomique dans Prop, c est une constante dans $\mathbb{N}$ et l'opérateur $\bowtie$ est dans $\{<, >, \leq, \geq, =\}$.

Interprétation : sur des configurations d'un STT

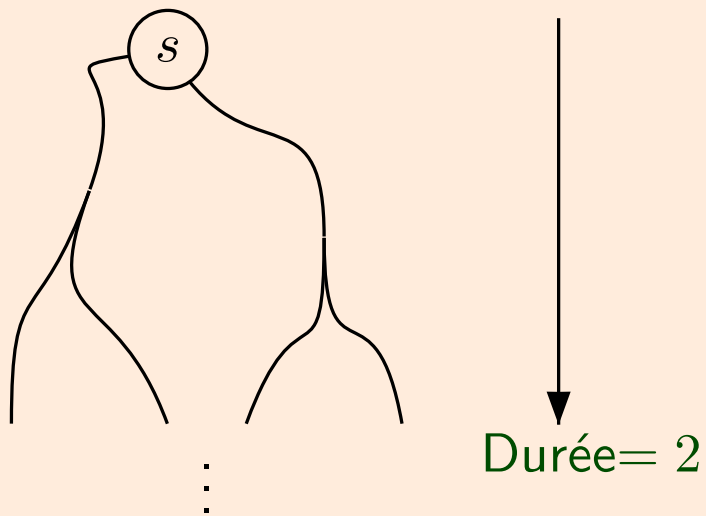
CTL+temps : TCTL

Syntaxe de TCTL [Alur - Henzinger 1991]

$$\varphi, \psi ::= P \mid \neg\varphi \mid \varphi \wedge \psi \mid \mathsf{E}\varphi\mathsf{U}_{\bowtie c}\psi \mid \mathsf{A}\varphi\mathsf{U}_{\bowtie c}\psi$$

où P est une proposition atomique dans Prop, c est une constante dans $\mathbb{N}$ et l'opérateur $\bowtie$ est dans $\{<, >, \leq, \geq, =\}$.

Interprétation : sur des configurations d'un STT



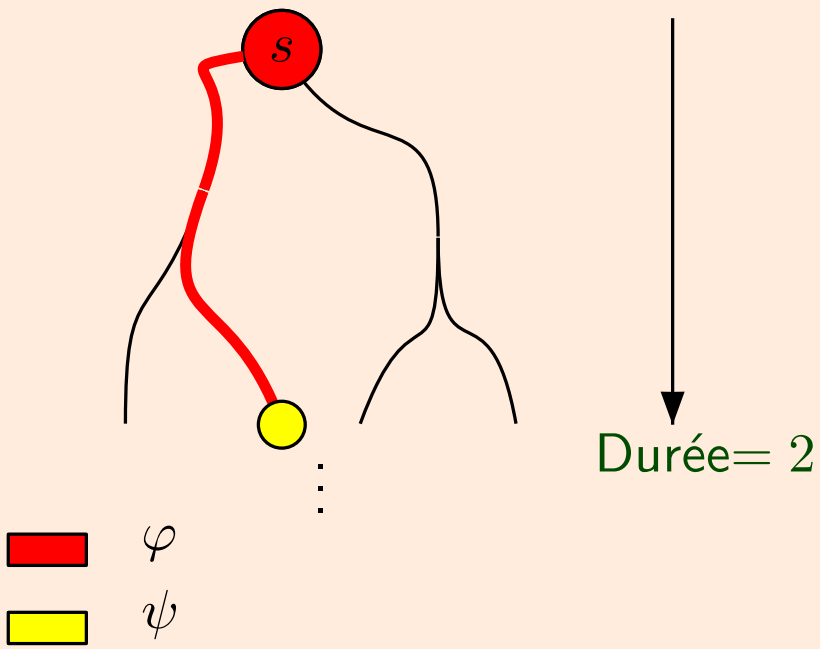
CTL+temps : TCTL

Syntaxe de TCTL [Alur - Henzinger 1991]

$$\varphi, \psi ::= P \mid \neg\varphi \mid \varphi \wedge \psi \mid \mathsf{E}\varphi\mathsf{U}_{\bowtie c}\psi \mid \mathsf{A}\varphi\mathsf{U}_{\bowtie c}\psi$$

où P est une proposition atomique dans Prop, c est une constante dans $\mathbb{N}$ et l'opérateur $\bowtie$ est dans $\{<, >, \leq, \geq, =\}$.

Interprétation : sur des configurations d'un STT



$$s \models \mathsf{E}\varphi\mathsf{U}_{\leq 2}\psi$$

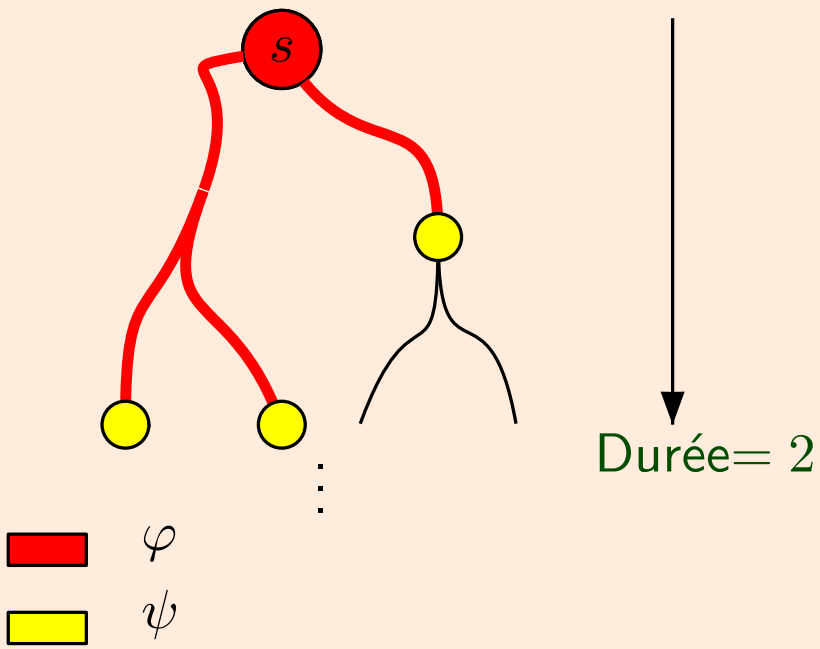
CTL+temps : TCTL

Syntaxe de TCTL [Alur - Henzinger 1991]

$$\varphi, \psi ::= P \mid \neg\varphi \mid \varphi \wedge \psi \mid \mathsf{E}\varphi\mathsf{U}_{\bowtie c}\psi \mid \mathsf{A}\varphi\mathsf{U}_{\bowtie c}\psi$$

où P est une proposition atomique dans Prop, c est une constante dans $\mathbb{N}$ et l'opérateur $\bowtie$ est dans $\{<, >, \leq, \geq, =\}$.

Interprétation : sur des configurations d'un STT



$$s \models \mathsf{A}\varphi\mathsf{U}_{\leq 2}\psi$$

CTL+temps : TCTL

Syntaxe de TCTL [Alur - Henzinger 1991]

$$\varphi, \psi ::= P \mid \neg\varphi \mid \varphi \wedge \psi \mid \mathsf{E}\varphi\mathsf{U}_{\bowtie c}\psi \mid \mathsf{A}\varphi\mathsf{U}_{\bowtie c}\psi$$

où P est une proposition atomique dans Prop, c est une constante dans $\mathbb{N}$ et l'opérateur $\bowtie$ est dans $\{<, >, \leq, \geq, =\}$.

Interprétation : sur des configurations de $\mathcal{T} = (S, s_0, L, E)$

$s \models \mathsf{E}\varphi\mathsf{U}_{\bowtie c}\psi$ s'il existe une exécution ρ partant de s telle que $\rho \models \varphi\mathsf{U}_{\bowtie c}\psi$

$s \models \mathsf{A}\varphi\mathsf{U}_{\bowtie c}\psi$ si, pour toute exécution ρ partant de s, $\rho \models \varphi\mathsf{U}_{\bowtie c}\psi$

avec

$\rho \models \varphi\mathsf{U}_{\bowtie c}\psi$ s'il existe une position p de ρ telle que $\mathsf{Dur}(\rho^{\leq p}) \bowtie c$, $s_p \models \psi$ et pour toute position $p' <_\rho p$, $s_{p'} \models \varphi$

TCTL

et temps de réponse

Tout signal est suivi d'une réponse en moins de 5 unités de temps :

$$\mathsf{AG}(\text{signal} \Rightarrow \mathsf{AF}_{\leq 5}\ \text{reponse})$$

et automates temporisés

$\mathcal{A} = (Q, q_0, \ell, Inv, \Delta)$ un automate temporisé,
$\mathcal{T} = (S, s_0, L, E)$ le système de transitions associé avec :

- la configuration initiale $s_0 = (q_0, v_0)$ où $v_0 = \mathbf{0}$,
- des exécutions de la forme

$$(q_0, v_0) \xrightarrow{d_1} (q_0, v_0 + d_1) \xrightarrow{a_1} (q_1, v_1) \xrightarrow{d_2} (q_1, v_1 + d_2) \xrightarrow{a_2} (q_2, v_2) \cdots$$

Pour une formule φ de TCTL, $\mathcal{A} \models \varphi$ si $s_0 \models \varphi$.

TCTL

et temps de réponse

Tout signal est suivi d'une réponse en moins de 5 unités de temps :

$$\mathsf{AG}(\text{signal} \Rightarrow \mathsf{AF}_{\leq 5}\ \text{reponse})$$

et automates temporisés

$\mathcal{A} = (Q, q_0, \ell, Inv, \Delta)$ un automate temporisé,
$\mathcal{T} = (S, s_0, L, E)$ le système de transitions associé avec :

- la configuration initiale $s_0 = (q_0, v_0)$ où $v_0 = \mathbf{0}$,
- des exécutions de la forme

$$(q_0, v_0) \xrightarrow{d_1} (q_0, v_0 + d_1) \xrightarrow{a_1} (q_1, v_1) \xrightarrow{d_2} (q_1, v_1 + d_2) \xrightarrow{a_2} (q_2, v_2) \cdots$$

Pour une formule φ de TCTL, $\mathcal{A} \models \varphi$ si $s_0 \models \varphi$.

TCTL

et temps de réponse

Tout signal est suivi d'une réponse en moins de 5 unités de temps :

$$\text{AG}(\text{signal} \Rightarrow \text{AF}_{\leq 5}\ \text{reponse})$$

et automates temporisés

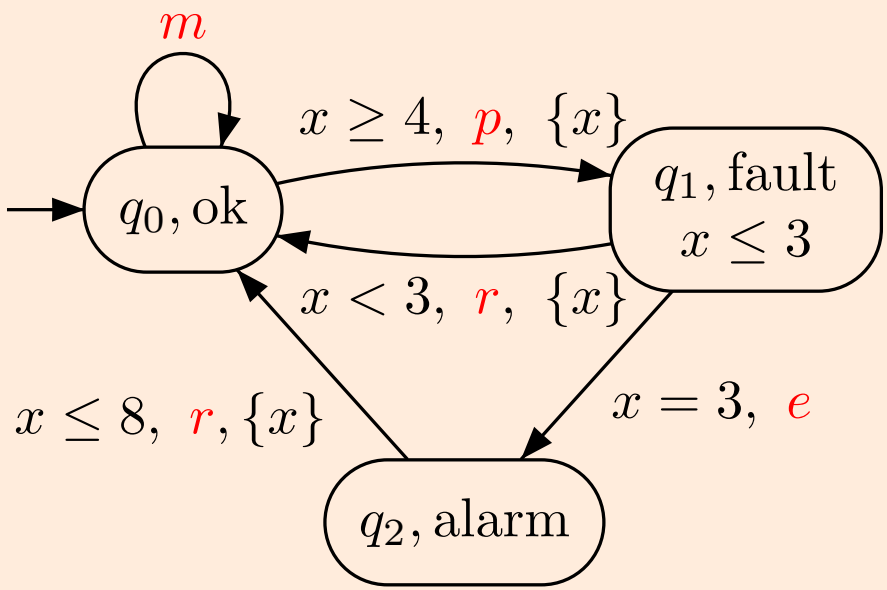
$\mathcal{A} = (Q, q_0, \ell, Inv, \Delta)$ un automate temporisé,
$\mathcal{T} = (S, s_0, L, E)$ le système de transitions associé avec :

- la configuration initiale $s_0 = (q_0, v_0)$ où $v_0 = \mathbf{0}$,
- des exécutions de la forme

$$(q_0, v_0) \xrightarrow{d_1} (q_0, v_0 + d_1) \xrightarrow{a_1} (q_1, v_1) \xrightarrow{d_2} (q_1, v_1 + d_2) \xrightarrow{a_2} (q_2, v_2) \cdots$$

Pour une formule φ de TCTL, $\mathcal{A} \models \varphi$ si $s_0 \models \varphi$.

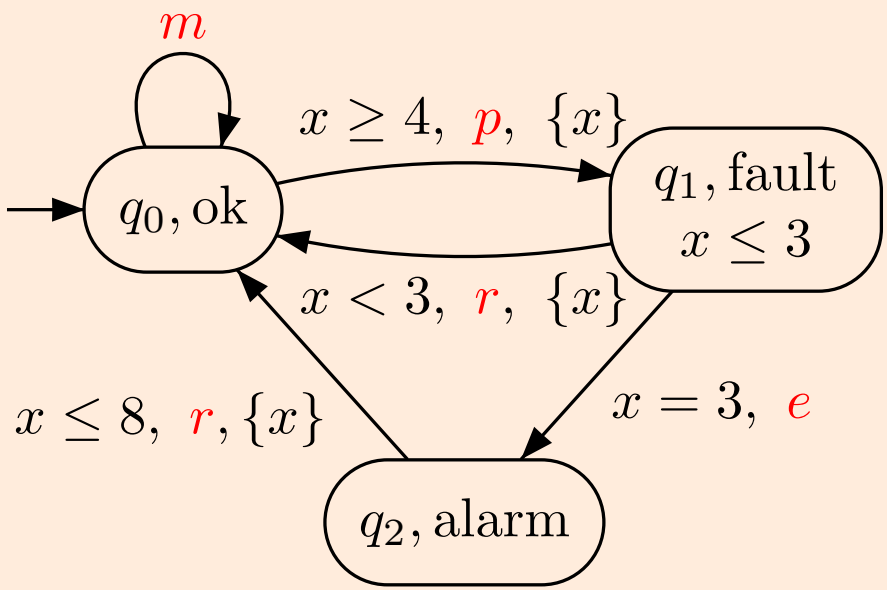
Exemple



satisfait

$$\mathsf{AG}(\mathsf{fault} \Rightarrow \mathsf{AF}_{\leq 8}\ \mathsf{ok})$$

Exemple



satisfait

$$\mathsf{AG}(\mathsf{fault} \Rightarrow \mathsf{AF}_{\leq 8}\, \mathsf{ok})$$

D'autres logiques

du temps arborescent

- Extension de TCTL avec horloges

$$\mathsf{AG}(\text{signal} \Rightarrow z \cdot \mathsf{AF}(z \leq 5 \ \wedge \ \text{reponse}))$$

z horloge de formule remise à zéro par "$z\cdot$"

- Logique T_μ, avec par exemple l'opérateur $\triangleright$, "until en un pas"

$$z \cdot (\text{true} \triangleright (P \wedge z \leq 5))$$

P deviendra vraie en une seule transition, exécutée avant 5 unités de temps.

- etc.

D'autres logiques

du temps arborescent

- Extension de TCTL avec horloges

$$\mathsf{AG}(\text{signal} \Rightarrow z \cdot \mathsf{AF}(z \leq 5 \ \wedge \ \text{reponse}))$$

  z horloge de formule remise à zéro par "$z\cdot$"

- Logique T_μ, avec par exemple l'opérateur $\triangleright$, "until en un pas"

$$z \cdot (\mathsf{true} \triangleright (P \wedge z \leq 5))$$

  P deviendra vraie en une seule transition, exécutée avant 5 unités de temps.

- etc.

D'autres logiques

du temps arborescent

- Extension de TCTL avec horloges

$$\mathsf{AG}(\text{signal} \Rightarrow z \cdot \mathsf{AF}(z \leq 5 \ \wedge \ \text{reponse}))$$

z horloge de formule remise à zéro par "$z\cdot$"

- Logique T_μ, avec par exemple l'opérateur $\triangleright$, "until en un pas"

$$z \cdot (\mathsf{true} \triangleright (P \wedge z \leq 5))$$

P deviendra vraie en une seule transition, exécutée avant 5 unités de temps.

- etc.

LTL

Syntaxe de LTL

$$\varphi, \psi ::= P \mid \neg\varphi \mid \varphi \wedge \psi \mid \mathsf{X}\varphi \mid \varphi \mathsf{U} \psi$$

où P est une proposition atomique dans Prop.

Interprétation : pour une structure de Kripke

sur une position le long d'une exécution

LTL

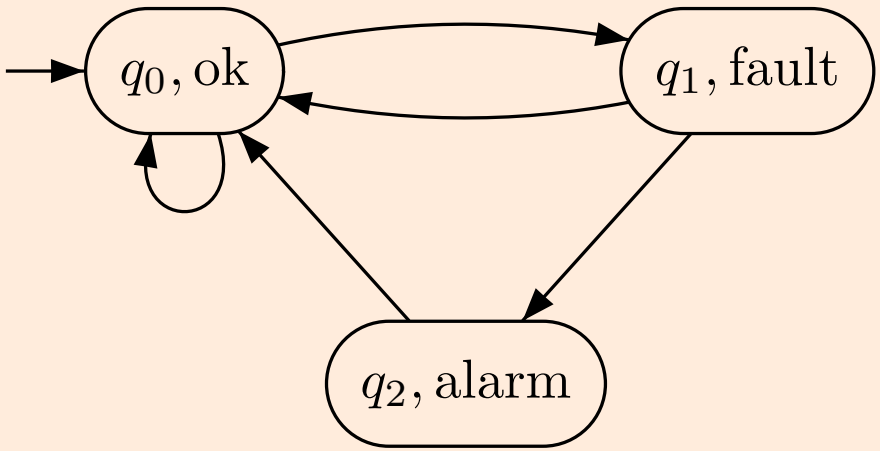
Syntaxe de LTL

$$\varphi, \psi ::= P \mid \neg\varphi \mid \varphi \wedge \psi \mid \mathsf{X}\varphi \mid \varphi \mathsf{U} \psi$$

où P est une proposition atomique dans Prop.

Interprétation : pour une structure de Kripke

sur une position le long d'une exécution



LTL

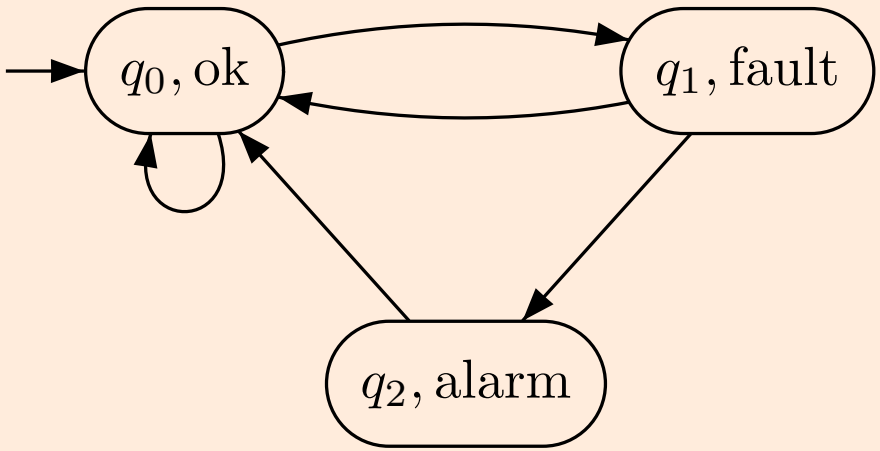
Syntaxe de LTL

$$\varphi, \psi ::= P \mid \neg\varphi \mid \varphi \wedge \psi \mid \mathsf{X}\varphi \mid \varphi \mathsf{U} \psi$$

où P est une proposition atomique dans Prop.

Interprétation : pour une structure de Kripke

sur une position le long d'une exécution



$\sigma \;:\; q_0, q_0, q_1, q_0, q_1, q_2, q_0, \ldots$

$\sigma, 2 \models \mathsf{fault} \wedge \mathsf{X}\,\mathsf{ok}$

$\sigma, 4 \models \mathsf{F}\,\mathsf{ok}$

LTL

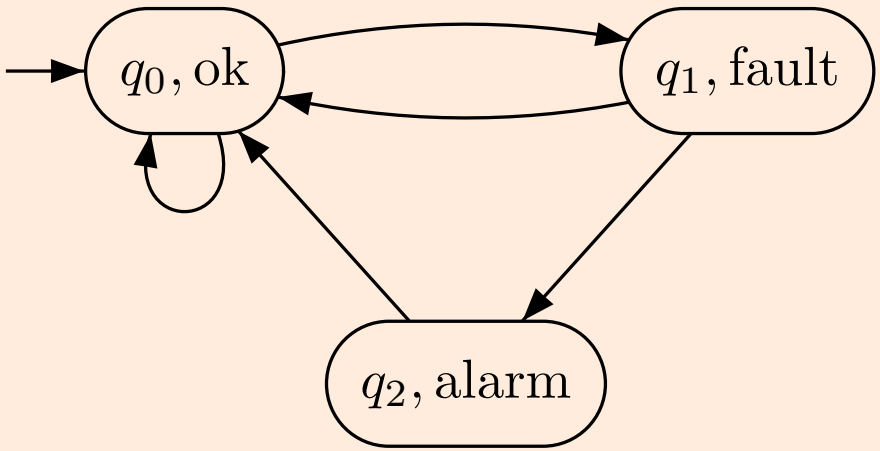
Syntaxe de LTL

$$\varphi, \psi ::= P \mid \neg\varphi \mid \varphi \wedge \psi \mid \mathsf{X}\varphi \mid \varphi \mathsf{U} \psi$$

où P est une proposition atomique dans Prop.

Interprétation : pour une structure de Kripke

sur une position le long d'une exécution



$\sigma \;:\; q_0, q_0, q_1, q_0, q_1, q_2, q_0, \ldots$

$\sigma, 2 \models \text{fault} \wedge \mathsf{X}\,\text{ok}$

$\sigma, 4 \models \mathsf{F}\,\text{ok}$

$\sigma, 0 \models \mathsf{GF}\,\text{ok}$

$\mathcal{K} \models \mathsf{GF}\,\text{ok}$

pour toute exécution σ de $\mathcal{K}$, $\sigma, 0 \models \mathsf{GF}\,\text{ok}$

LTL+temps : MTL

Syntaxe de MTL [Koymans 1990]

$$\varphi, \psi ::= P \mid \neg\varphi \mid \varphi \wedge \psi \mid \varphi \mathsf{U}_I \psi$$

où P est une proposition atomique et I un intervalle.

MITL : restriction à des intervalles non réduits à un point

Sémantique

Une formule s'interprète sur une exécution $s \xrightarrow{d_1} s'_1 \xrightarrow{a_1} s_2 \xrightarrow{d_2} s'_2 \xrightarrow{a_2} s_3 \cdots$ partant d'une configuration quelconque s d'un système de transition temporisé.

Pour une formule φ de TCTL, et pour un automate temporisé $\mathcal{A} = (Q, q_0, \ell, Inv, \Delta)$, avec le système de transitions temporisé associé $\mathcal{T} = (S, s_0, L, E)$:

$\mathcal{A} \models \varphi$ si toute exécution partant de la configuration initiale $s_0 = (q_0, v_0)$ avec $v_0 = \mathbf{0}$ satisfait φ.

Temps de réponse

$$\mathsf{G}(\text{signal} \Rightarrow \mathsf{F}_{[0,5]}\ \text{reponse})$$

LTL+temps : MTL

Syntaxe de MTL [Koymans 1990]

$$\varphi, \psi ::= P \mid \neg\varphi \mid \varphi \wedge \psi \mid \varphi \mathsf{U}_I \psi$$

où P est une proposition atomique et I un intervalle.

MITL : restriction à des intervalles non réduits à un point

Sémantique

Une formule s'interprète sur une exécution $s \xrightarrow{d_1} s'_1 \xrightarrow{a_1} s_2 \xrightarrow{d_2} s'_2 \xrightarrow{a_2} s_3 \cdots$ partant d'une configuration quelconque s d'un système de transition temporisé.

Pour une formule φ de TCTL, et pour un automate temporisé $\mathcal{A} = (Q, q_0, \ell, Inv, \Delta)$, avec le système de transitions temporisé associé $\mathcal{T} = (S, s_0, L, E)$:

$\mathcal{A} \models \varphi$ si toute exécution partant de la configuration initiale $s_0 = (q_0, v_0)$ avec $v_0 = \mathbf{0}$ satisfait φ.

Temps de réponse

$$\mathsf{G}(\text{signal} \Rightarrow \mathsf{F}_{[0,5]}\ \text{reponse})$$

LTL+temps : MTL

Syntaxe de MTL [Koymans 1990]

$$\varphi, \psi ::= P \mid \neg\varphi \mid \varphi \wedge \psi \mid \varphi \mathsf{U}_I \psi$$

où P est une proposition atomique et I un intervalle.

MITL : restriction à des intervalles non réduits à un point

Sémantique

Une formule s'interprète sur une exécution $s \xrightarrow{d_1} s'_1 \xrightarrow{a_1} s_2 \xrightarrow{d_2} s'_2 \xrightarrow{a_2} s_3 \cdots$ partant d'une configuration quelconque s d'un système de transition temporisé.

Pour une formule φ de TCTL, et pour un automate temporisé $\mathcal{A} = (Q, q_0, \ell, Inv, \Delta)$, avec le système de transitions temporisé associé $\mathcal{T} = (S, s_0, L, E)$:

$\mathcal{A} \models \varphi$ si toute exécution partant de la configuration initiale $s_0 = (q_0, v_0)$ avec $v_0 = \mathbf{0}$ satisfait φ.

Temps de réponse

$$\mathsf{G}(\text{signal} \Rightarrow \mathsf{F}_{[0,5]}\ \text{reponse})$$

LTL+temps : MTL

Syntaxe de MTL [Koymans 1990]

$$\varphi, \psi ::= P \mid \neg\varphi \mid \varphi \wedge \psi \mid \varphi \mathsf{U}_I \psi$$

où P est une proposition atomique et I un intervalle.

MITL : restriction à des intervalles non réduits à un point

Sémantique

Une formule s'interprète sur une exécution $s \xrightarrow{d_1} s'_1 \xrightarrow{a_1} s_2 \xrightarrow{d_2} s'_2 \xrightarrow{a_2} s_3 \cdots$ partant d'une configuration quelconque s d'un système de transition temporisé.

Pour une formule φ de TCTL, et pour un automate temporisé $\mathcal{A} = (Q, q_0, \ell, Inv, \Delta)$, avec le système de transitions temporisé associé $\mathcal{T} = (S, s_0, L, E)$:

$\mathcal{A} \models \varphi$ si toute exécution partant de la configuration initiale $s_0 = (q_0, v_0)$ avec $v_0 = \mathbf{0}$ satisfait φ.

Temps de réponse

$$\mathsf{G}(\text{signal} \Rightarrow \mathsf{F}_{[0,5]}\ \text{reponse})$$

LTL+temps : MTL

Syntaxe de MTL [Koymans 1990]

$$\varphi, \psi ::= P \mid \neg\varphi \mid \varphi \wedge \psi \mid \varphi \mathsf{U}_I \psi$$

où P est une proposition atomique et I un intervalle.

MITL : restriction à des intervalles non réduits à un point

Sémantique

Une formule s'interprète sur une exécution $s \xrightarrow{d_1} s'_1 \xrightarrow{a_1} s_2 \xrightarrow{d_2} s'_2 \xrightarrow{a_2} s_3 \cdots$ partant d'une configuration quelconque s d'un système de transition temporisé.

Pour une formule φ de TCTL, et pour un automate temporisé $\mathcal{A} = (Q, q_0, \ell, Inv, \Delta)$, avec le système de transitions temporisé associé $\mathcal{T} = (S, s_0, L, E)$:

$\mathcal{A} \models \varphi$ si toute exécution partant de la configuration initiale $s_0 = (q_0, v_0)$ avec $v_0 = \mathbf{0}$ satisfait φ.

Temps de réponse

$$\mathsf{G}(\text{signal} \Rightarrow \mathsf{F}_{[0,5]} \text{ reponse})$$

D'autres logiques

du temps linéaire

- TPTL, extension de LTL avec horloges

$$\mathsf{G}x \cdot (\text{signal} \Rightarrow \mathsf{F}y \cdot (\text{reponse} \wedge \mathsf{y} \leq \mathsf{x} + 5))$$

  x et y horloges de formule, "gelées" par "$x\cdot$" et "$y\cdot$"

- Variantes avec temps discret ou interprétation sur des séquences discrètes
- etc.

D'autres logiques

du temps linéaire

- TPTL, extension de LTL avec horloges

$$\mathsf{G}x \cdot (\text{signal} \Rightarrow \mathsf{F}y \cdot (\text{reponse} \wedge \mathsf{y} \leq \mathsf{x} + 5))$$

  x et y horloges de formule, "gelées" par "$x\cdot$" et "$y\cdot$"

- Variantes avec temps discret ou interprétation sur des séquences discrètes

- etc.

D'autres logiques

du temps linéaire

- TPTL, extension de LTL avec horloges

$$\mathsf{G}x \cdot (\text{signal} \Rightarrow \mathsf{F}y \cdot (\text{reponse} \wedge \mathsf{y} \leq \mathsf{x} + 5))$$

  x et y horloges de formule, "gelées" par "$x\cdot$" et "$y\cdot$"

- Variantes avec temps discret ou interprétation sur des séquences discrètes
- etc.

Plan

Model-checking de LTL

Theorème [Sistla et Clarke 1985, Lichtenstein et al. 1985]

Le problème du model-checking pour LTL sur des structures de Kripke est PSPACE-complet.

Procédure de décision

Données : une formule φ de LTL et une structure de Kripke $\mathcal{K}$

1. Construction d'un automate (de Büchi) $\mathcal{A}_{\neg\varphi}$ dont les exécutions sont exactement les séquences d'étiquettes satisfaisant la formule $\neg\varphi$.
2. Construction de l'automate $\mathcal{H} = \mathcal{K} \otimes \mathcal{A}_{\neg\varphi}$ acceptant exactement les exécutions communes à $\mathcal{K}$ et $\mathcal{A}_{\neg\varphi}$ (intersection).
3. Test du vide pour $\mathcal{H}$.

Difficulté : construire l'automate $\mathcal{A}_\varphi$ caractérisant φ

Model-checking de LTL

Theorème [Sistla et Clarke 1985, Lichtenstein et al. 1985]

Le problème du model-checking pour LTL sur des structures de Kripke est PSPACE-complet.

Procédure de décision

Données : une formule φ de LTL et une structure de Kripke $\mathcal{K}$

1. Construction d'un automate (de Büchi) $\mathcal{A}_{\neg\varphi}$ dont les exécutions sont exactement les séquences d'étiquettes satisfaisant la formule $\neg\varphi$.
2. Construction de l'automate $\mathcal{H} = \mathcal{K} \otimes \mathcal{A}_{\neg\varphi}$ acceptant exactement les exécutions communes à $\mathcal{K}$ et $\mathcal{A}_{\neg\varphi}$ (intersection).
3. Test du vide pour $\mathcal{H}$.

Difficulté : construire l'automate $\mathcal{A}_\varphi$ caractérisant φ

Model-checking de LTL

Theorème [Sistla et Clarke 1985, Lichtenstein et al. 1985]

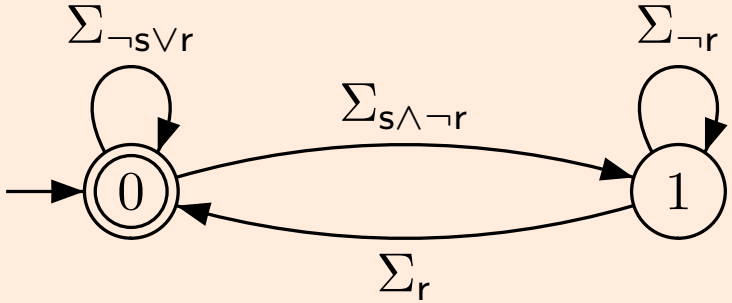
Le problème du model-checking pour LTL sur des structures de Kripke est PSPACE-complet.

Procédure de décision

Données : une formule φ de LTL et une structure de Kripke $\mathcal{K}$

1. Construction d'un automate (de Büchi) $\mathcal{A}_{\neg\varphi}$ dont les exécutions sont exactement les séquences d'étiquettes satisfaisant la formule $\neg\varphi$.
2. Construction de l'automate $\mathcal{H} = \mathcal{K} \otimes \mathcal{A}_{\neg\varphi}$ acceptant exactement les exécutions communes à $\mathcal{K}$ et $\mathcal{A}_{\neg\varphi}$ (intersection).
3. Test du vide pour $\mathcal{H}$.

Difficulté : construire l'automate $\mathcal{A}_\varphi$ caractérisant φ



$$\Sigma = 2^{\mathsf{Prop}}$$

$$\Sigma_r = \{\alpha \in \Sigma \mid r \in \alpha\}$$

$$\Sigma_{\neg r} = \Sigma \setminus \Sigma_r, \text{ etc.}$$

Model-checking de LTL

Theorème [Sistla et Clarke 1985, Lichtenstein et al. 1985]

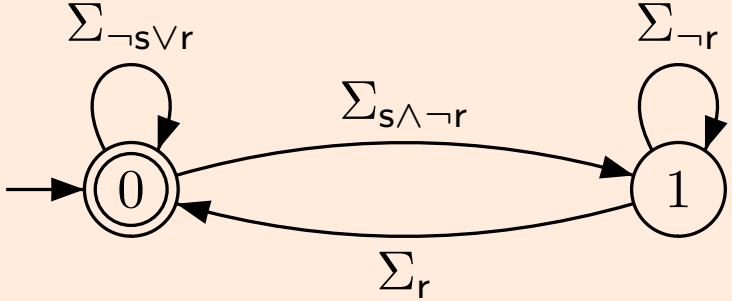
Le problème du model-checking pour LTL sur des structures de Kripke est PSPACE-complet.

Procédure de décision

Données : une formule φ de LTL et une structure de Kripke $\mathcal{K}$

1. Construction d'un automate (de Büchi) $\mathcal{A}_{\neg\varphi}$ dont les exécutions sont exactement les séquences d'étiquettes satisfaisant la formule $\neg\varphi$.
2. Construction de l'automate $\mathcal{H} = \mathcal{K} \otimes \mathcal{A}_{\neg\varphi}$ acceptant exactement les exécutions communes à $\mathcal{K}$ et $\mathcal{A}_{\neg\varphi}$ (intersection).
3. Test du vide pour $\mathcal{H}$.

Difficulté : construire l'automate $\mathcal{A}_\varphi$ caractérisant φ



pour la formule du temps de réponse :

$$G(s \Rightarrow F\, r)$$

Model-checking de LTL

Theorème [Sistla et Clarke 1985, Lichtenstein et al. 1985]

Le problème du model-checking pour LTL sur des structures de Kripke est PSPACE-complet.

Procédure de décision

Données : une formule φ de LTL et une structure de Kripke $\mathcal{K}$

1. Construction d'un automate (de Büchi) $\mathcal{A}_{\neg\varphi}$ dont les exécutions sont exactement les séquences d'étiquettes satisfaisant la formule $\neg\varphi$.
2. Construction de l'automate $\mathcal{H} = \mathcal{K} \otimes \mathcal{A}_{\neg\varphi}$ acceptant exactement les exécutions communes à $\mathcal{K}$ et $\mathcal{A}_{\neg\varphi}$ (intersection).
3. Test du vide pour $\mathcal{H}$.

Difficulté : construire l'automate $\mathcal{A}_\varphi$ caractérisant φ

Pire cas : $O(2^{|\varphi|})$

Model-checking de MTL et MITL

Theorèmes

- Le problème du model-checking pour MTL sur des automates temporisés est indécidable [Henzinger 1991].
- Le problème du model-checking pour MITL sur des automates temporisés est EXPSPACE-complet [Alur et al. 1996].

Procédure de décision pour MITL

Données : une formule φ de MITL et un automate temporisé $\mathcal{A}$

1. Construction d'un automate temporisé $\mathcal{A}_{\neg\varphi}$ dont les exécutions sont exactement celles qui satisfont la formule $\neg\varphi$.
2. Construction de l'automate temporisé $\mathcal{B} = \mathcal{A} \otimes \mathcal{A}_{\neg\varphi}$ acceptant les exécutions communes à $\mathcal{A}$ et $\mathcal{A}_{\neg\varphi}$.
3. Test du vide pour $\mathcal{B}$.

Remarque

Les points 1 et 3 sont plus difficiles pour des automates temporisés.

Model-checking de MTL et MITL

Theorèmes

- Le problème du model-checking pour MTL sur des automates temporisés est indécidable [Henzinger 1991].
- Le problème du model-checking pour MITL sur des automates temporisés est EXPSPACE-complet [Alur et al. 1996].

Procédure de décision pour MITL

Données : une formule φ de MITL et un automate temporisé $\mathcal{A}$

1. Construction d'un automate temporisé $\mathcal{A}_{\neg\varphi}$ dont les exécutions sont exactement celles qui satisfont la formule $\neg\varphi$.
2. Construction de l'automate temporisé $\mathcal{B} = \mathcal{A} \otimes \mathcal{A}_{\neg\varphi}$ acceptant les exécutions communes à $\mathcal{A}$ et $\mathcal{A}_{\neg\varphi}$.
3. Test du vide pour $\mathcal{B}$.

Remarque

Les points 1 et 3 sont plus difficiles pour des automates temporisés.

Model-checking de MTL et MITL

Theorèmes

- Le problème du model-checking pour MTL sur des automates temporisés est indécidable [Henzinger 1991].
- Le problème du model-checking pour MITL sur des automates temporisés est EXPSPACE-complet [Alur et al. 1996].

Procédure de décision pour MITL

Données : une formule φ de MITL et un automate temporisé $\mathcal{A}$

1. Construction d'un automate temporisé $\mathcal{A}_{\neg\varphi}$ dont les exécutions sont exactement celles qui satisfont la formule $\neg\varphi$.
2. Construction de l'automate temporisé $\mathcal{B} = \mathcal{A} \otimes \mathcal{A}_{\neg\varphi}$ acceptant les exécutions communes à $\mathcal{A}$ et $\mathcal{A}_{\neg\varphi}$.
3. Test du vide pour $\mathcal{B}$.

Remarque

Les points 1 et 3 sont plus difficiles pour des automates temporisés.

Model-checking de MTL et MITL

Theorèmes

- Le problème du model-checking pour MTL sur des automates temporisés est indécidable [Henzinger 1991].
- Le problème du model-checking pour MITL sur des automates temporisés est EXPSPACE-complet [Alur et al. 1996].

Procédure de décision pour MITL

Données : une formule φ de MITL et un automate temporisé $\mathcal{A}$

1. Construction d'un automate temporisé $\mathcal{A}_{\neg\varphi}$ dont les exécutions sont exactement celles qui satisfont la formule $\neg\varphi$.
2. Construction de l'automate temporisé $\mathcal{B} = \mathcal{A} \otimes \mathcal{A}_{\neg\varphi}$ acceptant les exécutions communes à $\mathcal{A}$ et $\mathcal{A}_{\neg\varphi}$.
3. Test du vide pour $\mathcal{B}$.

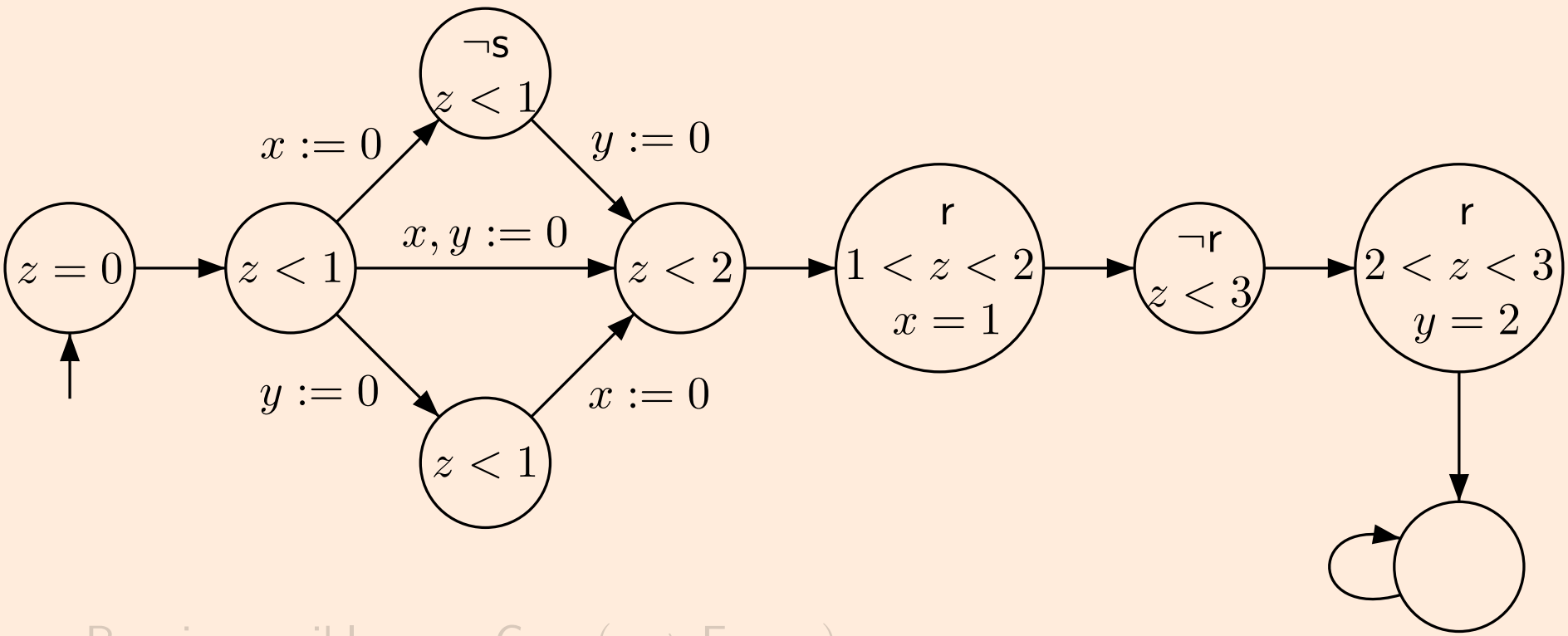
Remarque

Les points 1 et 3 sont plus difficiles pour des automates temporisés.

Exemple de construction

de $\mathcal{A}_\varphi$ pour $\varphi : \mathsf{G}_{]0,1]}(\mathsf{s} \Rightarrow \mathsf{F}_{[1,2]}\ \mathsf{r})$

On suppose que s et r sont vraies seulement sur des intervalles réduits à un point, avec au moins une configuration vérifiant r dans $]1,2[$ et dans $]2,3[$, et que r est fausse au temps 2.

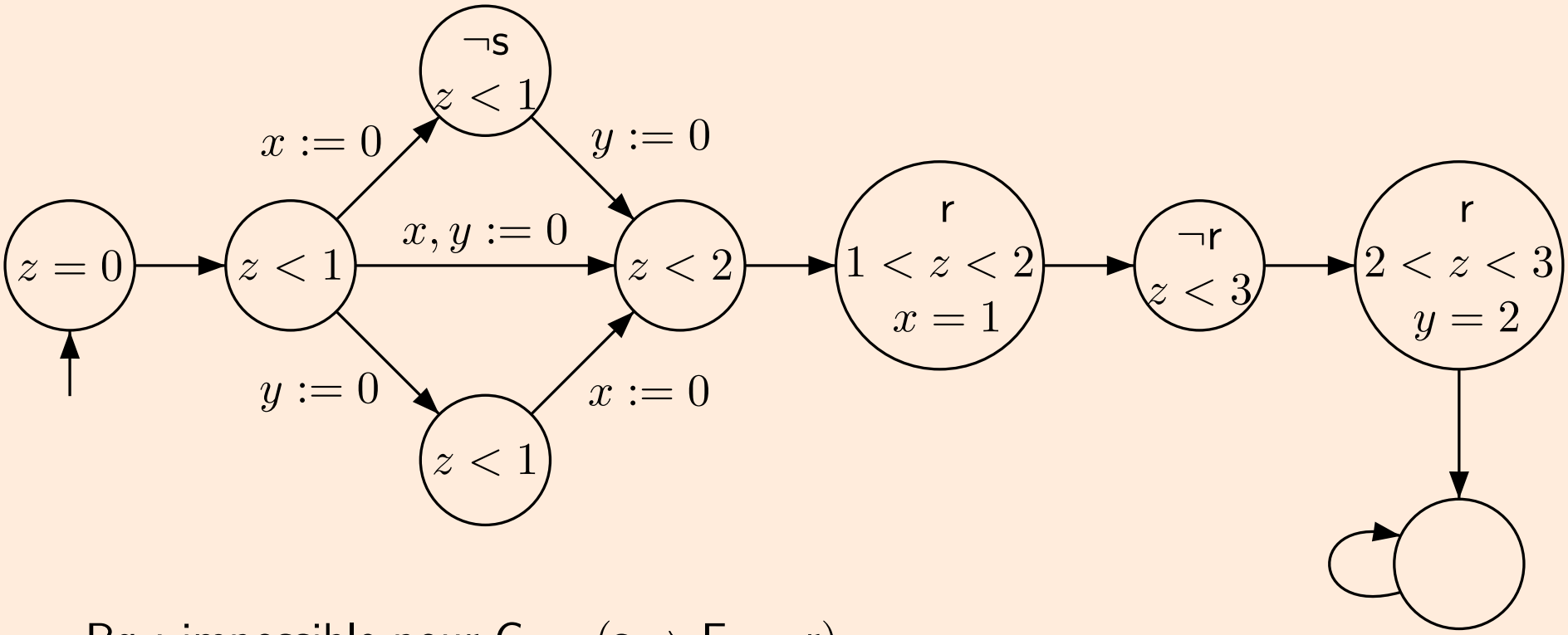


Rq : impossible pour $\mathsf{G}_{]0,1[}(\mathsf{s} \Rightarrow \mathsf{F}_{[1,1]}\ \mathsf{r})$

Exemple de construction

de $\mathcal{A}_\varphi$ pour φ : $\mathsf{G}_{]0,1]}(\mathsf{s} \Rightarrow \mathsf{F}_{[1,2]}\ \mathsf{r})$

On suppose que s et r sont vraies seulement sur des intervalles réduits à un point, avec au moins une configuration vérifiant r dans $]1,2[$ et dans $]2,3[$, et que r est fausse au temps 2.



Rq : impossible pour $\mathsf{G}_{]0,1[}(\mathsf{s} \Rightarrow \mathsf{F}_{[1,1]}\ \mathsf{r})$

Test du vide

Theorème [Alur et Dill 1990]

Le test du vide pour les automates temporisés est PSPACE-complet.

Procédure de décision

Donnée : un automate temporisé $\mathcal{A} = (Q, q_0, \ell, Inv, \Delta)$ sur un ensemble d'horloges X, avec domaine de temps $\mathbb{R}_{\geq 0}$

- Construction d'un automate (de Büchi) non temporisé $\mathcal{H}$, tel que :
  $\mathcal{A}$ n'a aucune exécution $\Leftrightarrow$ $\mathcal{H}$ n'a aucune exécution.
- Test du vide pour $\mathcal{H}$.

Test du vide

Theorème [Alur et Dill 1990]

Le test du vide pour les automates temporisés est PSPACE-complet.

Procédure de décision

Donnée : un automate temporisé $\mathcal{A} = (Q, q_0, \ell, Inv, \Delta)$ sur un ensemble d'horloges X, avec domaine de temps $\mathbb{R}_{\geq 0}$

- Construction d'un automate (de Büchi) non temporisé $\mathcal{H}$, tel que :
  $\mathcal{A}$ n'a aucune exécution $\Leftrightarrow$ $\mathcal{H}$ n'a aucune exécution.
- Test du vide pour $\mathcal{H}$.

Test du vide

Theorème [Alur et Dill 1990]

Le test du vide pour les automates temporisés est PSPACE-complet.

Procédure de décision

Donnée : un automate temporisé $\mathcal{A} = (Q, q_0, \ell, Inv, \Delta)$ sur un ensemble d'horloges X, avec domaine de temps $\mathbb{R}_{\geq 0}$

- Construction d'un automate (de Büchi) non temporisé $\mathcal{H}$, tel que :
  $\mathcal{A}$ n'a aucune exécution $\Leftrightarrow$ $\mathcal{H}$ n'a aucune exécution.
- Test du vide pour $\mathcal{H}$.

$\mathcal{T} = (S, s_0, L, E)$

système de transition associé à $\mathcal{A}$

états : (q, v)

$q \in Q$, $v \in \mathbb{R}^X_{\geq 0}$

$\xrightarrow{quotient}$

$\mathcal{H}$

automate des régions de $\mathcal{A}$

états : $(q, [v])$

$q \in Q$, $[v]$ classe d'équivalence

pour une relation $\sim$ sur $\mathbb{R}^X_{\geq 0}$

Construction d'un quotient (1)

avec les propriétés suivantes :

Pour deux valuations équivalentes $v \sim v'$

1. si une transition d'action $q \xrightarrow{g,a,r} q'$ est possible depuis v, alors la même transition est possible depuis v' et les valuations obtenues $v[r \mapsto 0]$ et $v'[r \mapsto 0]$ sont équivalentes,
2. si une transition de délai d est possible depuis v, alors une transition de délai d' est possible depuis v' et les valuations obtenues $v + d$ et $v' + d'$ sont équivalentes.

Remarques

- Cette relation est une bisimulation avec abstraction du temps entre les états de $\mathcal{T} : Q \times \mathbb{R}^X_{\geq 0}$ et ceux de $\mathcal{H} : Q \times \mathbb{R}^X_{\geq 0}/\sim$.
- Pour la première condition, il suffit de considérer les contraintes $x \bowtie k$, pour les horloges de X et les constantes $0 \leq k \leq m$, où m est la plus grande constante apparaissant dans les contraintes de $\mathcal{A}$.

Construction d'un quotient (1)

avec les propriétés suivantes :

Pour deux valuations équivalentes $v \sim v'$

1. si une transition d'action $q \xrightarrow{g,a,r} q'$ est possible depuis v, alors la même transition est possible depuis v' et les valuations obtenues $v[r \mapsto 0]$ et $v'[r \mapsto 0]$ sont équivalentes,
2. si une transition de délai d est possible depuis v, alors une transition de délai d' est possible depuis v' et les valuations obtenues $v + d$ et $v' + d'$ sont équivalentes.

Remarques

- Cette relation est une bisimulation avec abstraction du temps entre les états de $\mathcal{T} : Q \times \mathbb{R}^X_{\geq 0}$ et ceux de $\mathcal{H} : Q \times \mathbb{R}^X_{\geq 0}/\sim$.
- Pour la première condition, il suffit de considérer les contraintes $x \bowtie k$, pour les horloges de X et les constantes $0 \leq k \leq m$, où m est la plus grande constante apparaissant dans les contraintes de $\mathcal{A}$.

Construction d'un quotient (1)

avec les propriétés suivantes :

Pour deux valuations équivalentes $v \sim v'$

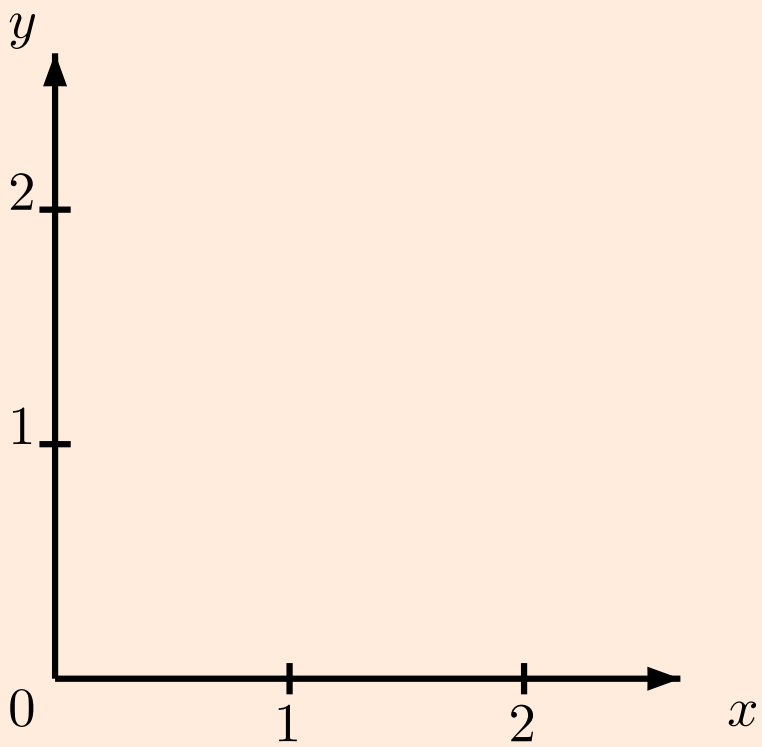
1. si une transition d'action $q \xrightarrow{g,a,r} q'$ est possible depuis v, alors la même transition est possible depuis v' et les valuations obtenues $v[r \mapsto 0]$ et $v'[r \mapsto 0]$ sont équivalentes,
2. si une transition de délai d est possible depuis v, alors une transition de délai d' est possible depuis v' et les valuations obtenues $v + d$ et $v' + d'$ sont équivalentes.

Remarques

- Cette relation est une bisimulation avec abstraction du temps entre les états de $\mathcal{T} : Q \times \mathbb{R}^X_{\geq 0}$ et ceux de $\mathcal{H} : Q \times \mathbb{R}^X_{\geq 0}/\sim$.
- Pour la première condition, il suffit de considérer les contraintes $x \bowtie k$, pour les horloges de X et les constantes $0 \leq k \leq m$, où m est la plus grande constante apparaissant dans les contraintes de $\mathcal{A}$.

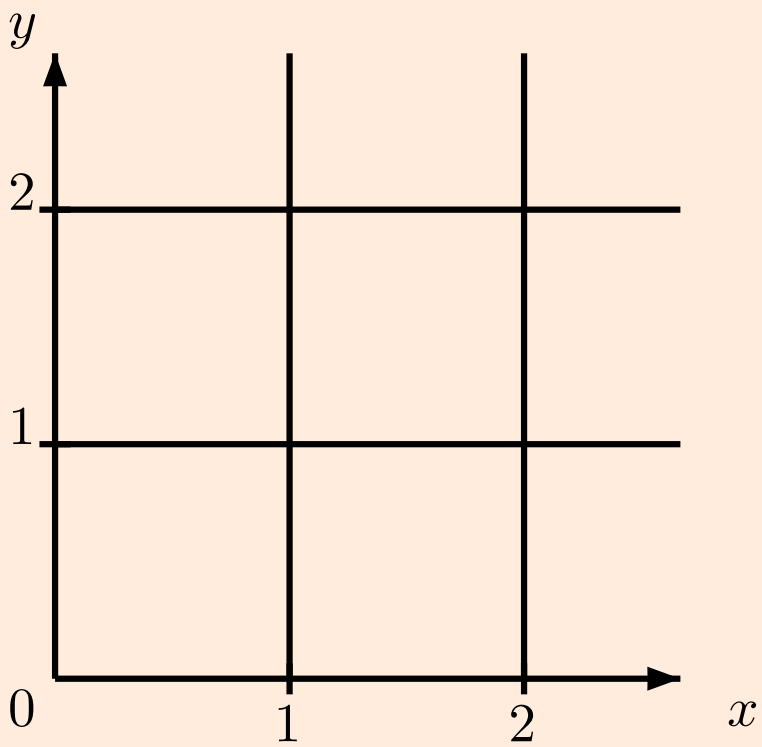
Construction d'un quotient (2)

Vue géométrique avec deux horloges x et y, pour $m = 2$



Construction d'un quotient (2)

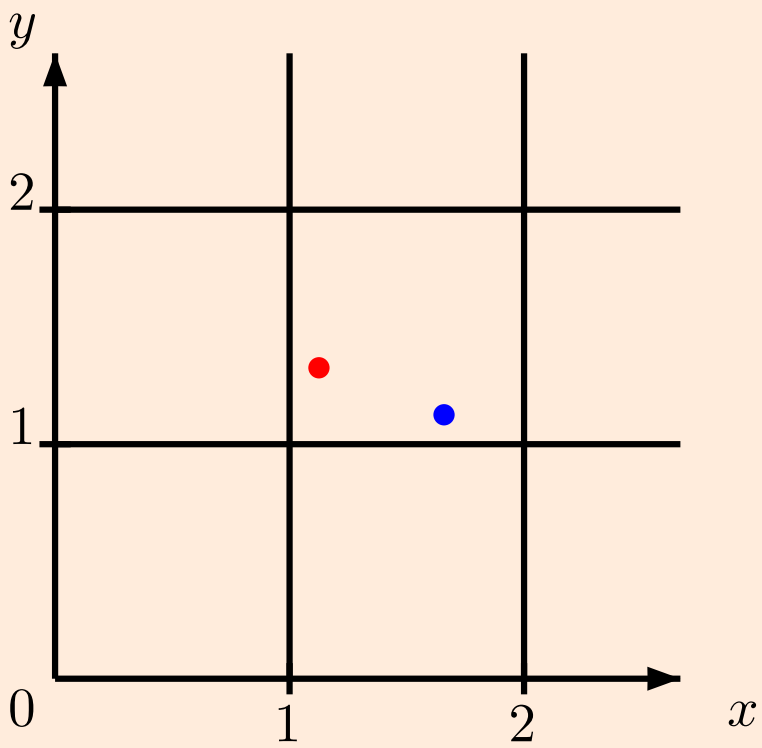
Vue géométrique avec deux horloges x et y, pour $m = 2$



- Compatibilité de valuations équivalentes avec les contraintes $x \bowtie k$

Construction d'un quotient (2)

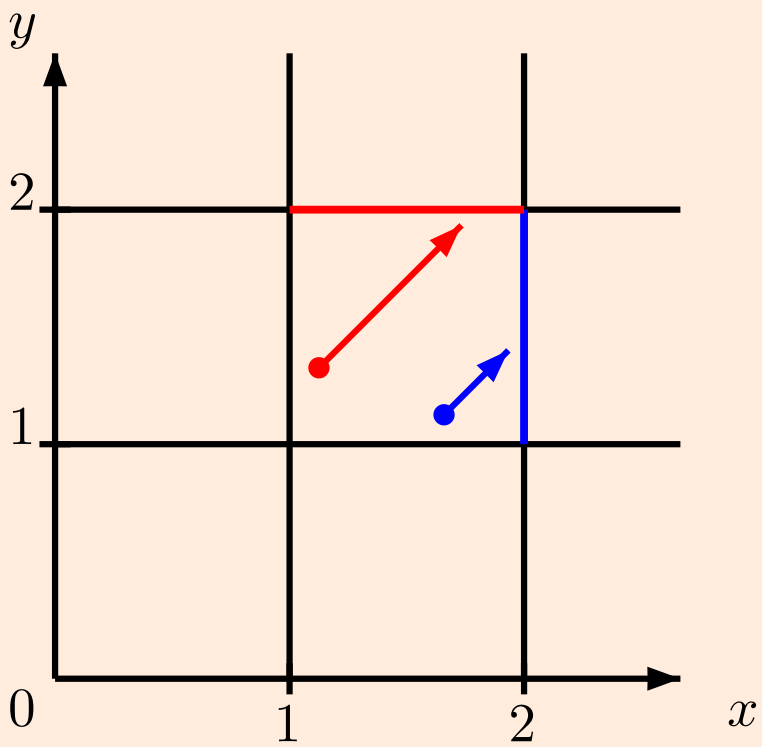
Vue géométrique avec deux horloges x et y, pour $m = 2$



- Compatibilité de valuations équivalentes avec les contraintes $x \bowtie k$
- Compatibilité de valuations équivalentes avec l'écoulement du temps

Construction d'un quotient (2)

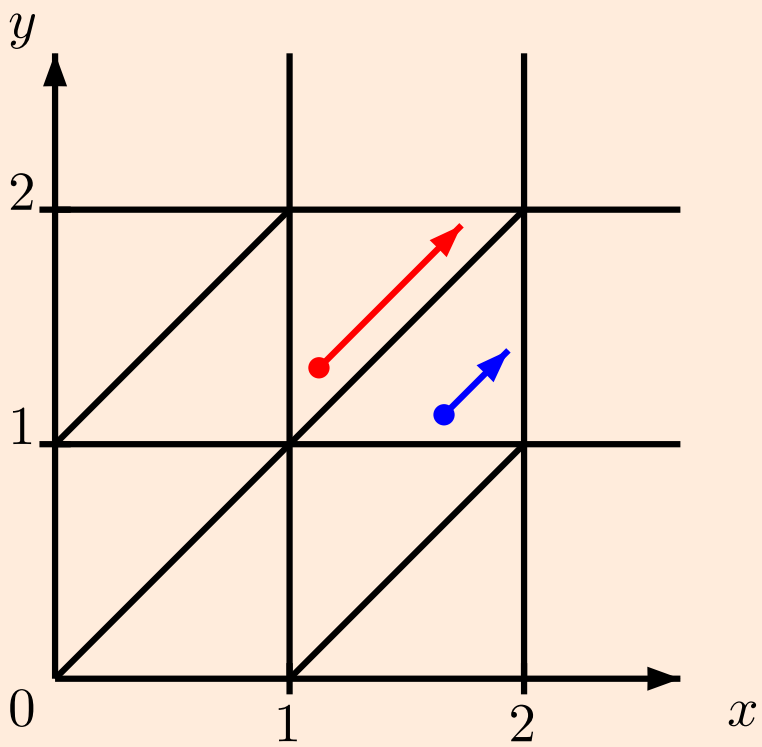
Vue géométrique avec deux horloges x et y, pour $m = 2$



- Compatibilité de valuations équivalentes avec les contraintes $x \bowtie k$
- Compatibilité de valuations équivalentes avec l'écoulement du temps

Construction d'un quotient (2)

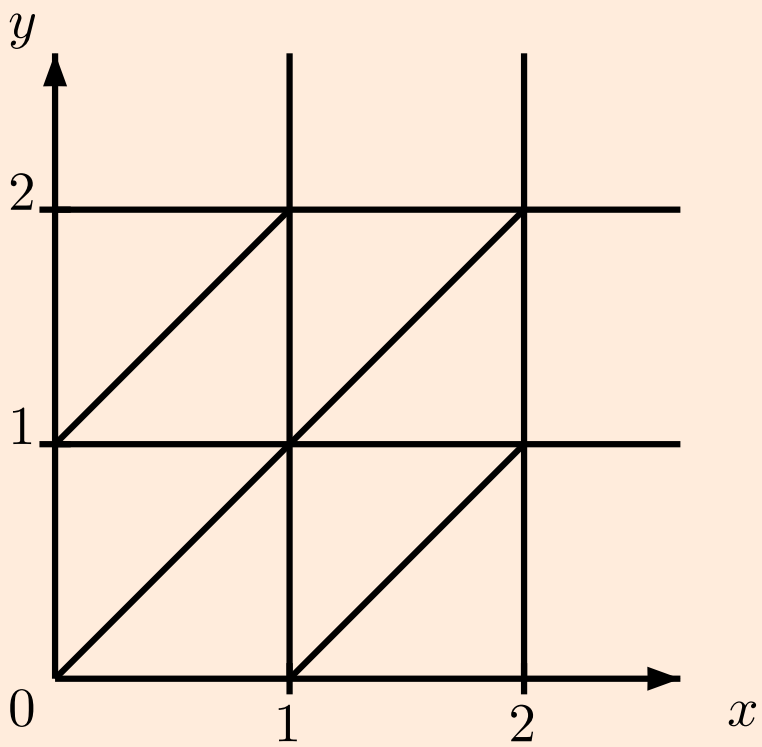
Vue géométrique avec deux horloges x et y, pour $m = 2$



- Compatibilité de valuations équivalentes avec les contraintes $x \bowtie k$
- Compatibilité de valuations équivalentes avec l'écoulement du temps

Construction d'un quotient (2)

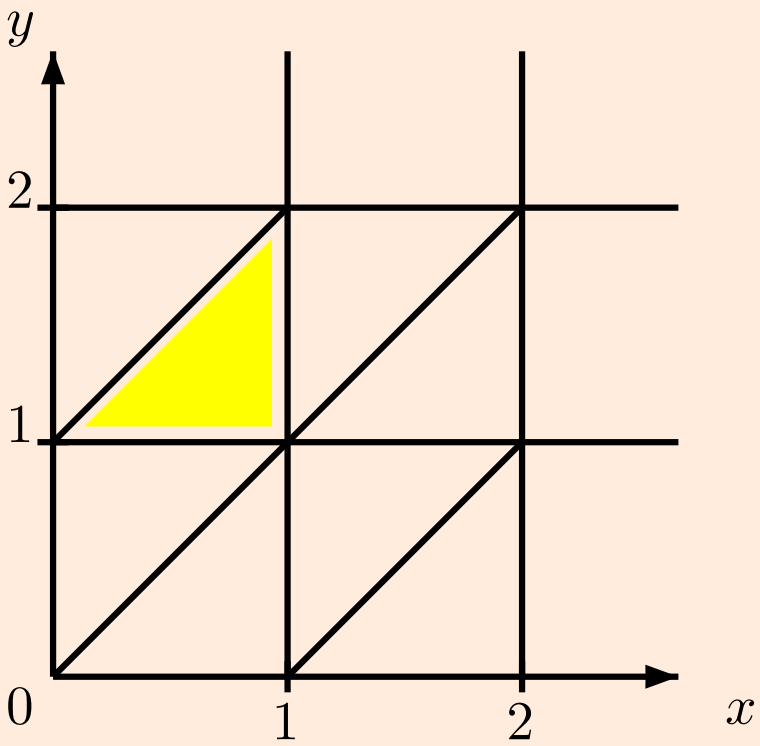
Vue géométrique avec deux horloges x et y, pour $m = 2$



- Compatibilité de valuations équivalentes avec les contraintes $x \bowtie k$
- Compatibilité de valuations équivalentes avec l'écoulement du temps

Construction d'un quotient (2)

Vue géométrique avec deux horloges x et y, pour $m = 2$

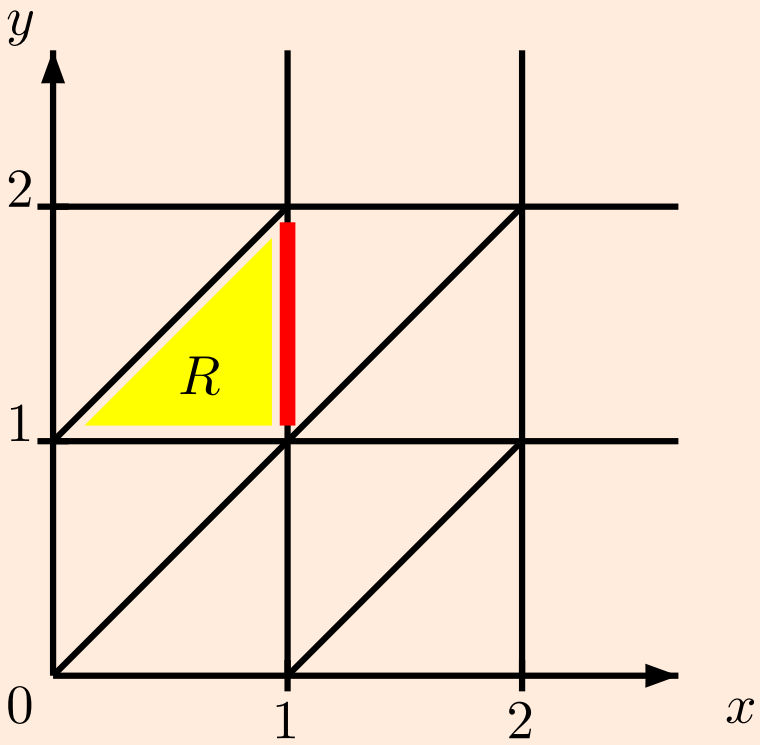


région R définie par
$I_x =]0;1[$, $I_y =]1;2[$
$frac(x) > frac(y)$

- Compatibilité de valuations équivalentes avec les contraintes $x \bowtie k$
- Compatibilité de valuations équivalentes avec l'écoulement du temps

Construction d'un quotient (2)

Vue géométrique avec deux horloges x et y, pour $m = 2$



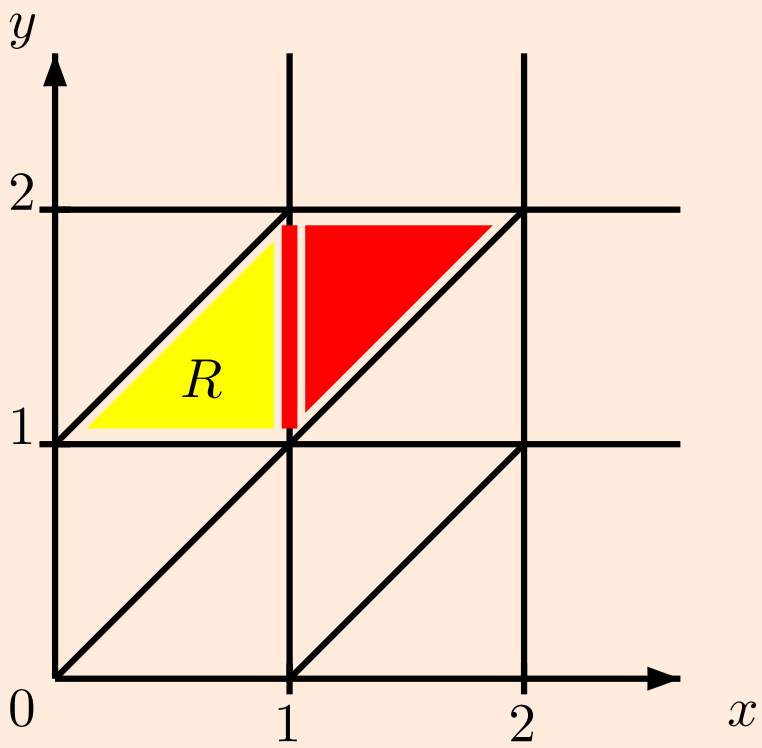
région R définie par
$I_x =]0;1[$, $I_y =]1;2[$
$frac(x) > frac(y)$

Successeur de R pour le temps
$I_x = [1;1]$, $I_y =]1;2[$

- Compatibilité de valuations équivalentes avec les contraintes $x \bowtie k$
- Compatibilité de valuations équivalentes avec l'écoulement du temps

Construction d'un quotient (2)

Vue géométrique avec deux horloges x et y, pour $m = 2$



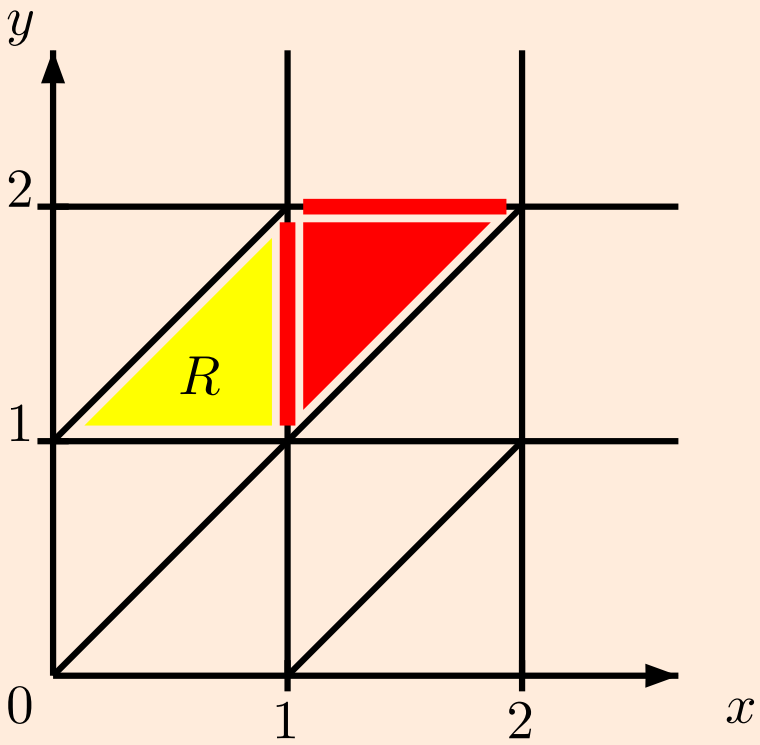
région R définie par
$I_x =]0;1[$, $I_y =]1;2[$
$frac(x) > frac(y)$

Successeur de R pour le temps
$I_x = [1;1]$, $I_y =]1;2[$

- Compatibilité de valuations équivalentes avec les contraintes $x \bowtie k$
- Compatibilité de valuations équivalentes avec l'écoulement du temps

Construction d'un quotient (2)

Vue géométrique avec deux horloges x et y, pour $m = 2$



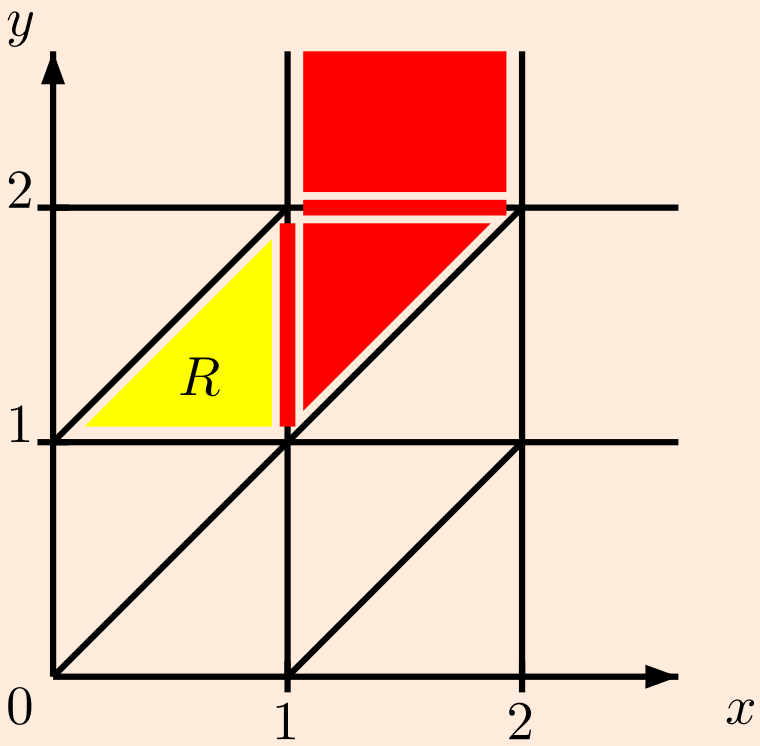
région R définie par
$I_x =]0;1[$, $I_y =]1;2[$
$frac(x) > frac(y)$

Successeur de R pour le temps
$I_x = [1;1]$, $I_y =]1;2[$

- Compatibilité de valuations équivalentes avec les contraintes $x \bowtie k$
- Compatibilité de valuations équivalentes avec l'écoulement du temps

Construction d'un quotient (2)

Vue géométrique avec deux horloges x et y, pour $m = 2$



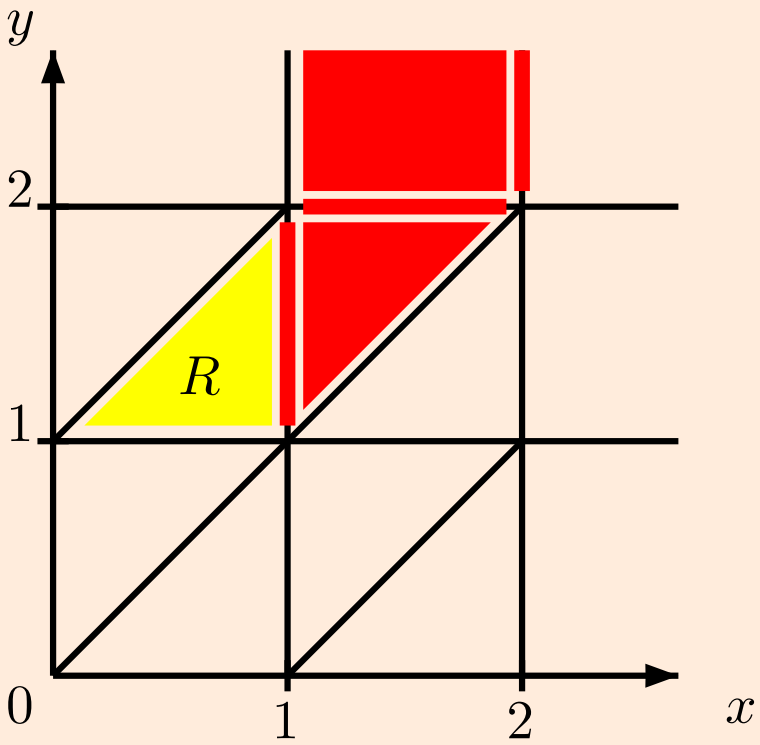
région R définie par
$I_x =]0;1[$, $I_y =]1;2[$
$frac(x) > frac(y)$

Successeur de R pour le temps
$I_x = [1;1]$, $I_y =]1;2[$

- Compatibilité de valuations équivalentes avec les contraintes $x \bowtie k$
- Compatibilité de valuations équivalentes avec l'écoulement du temps

Construction d'un quotient (2)

Vue géométrique avec deux horloges x et y, pour $m = 2$



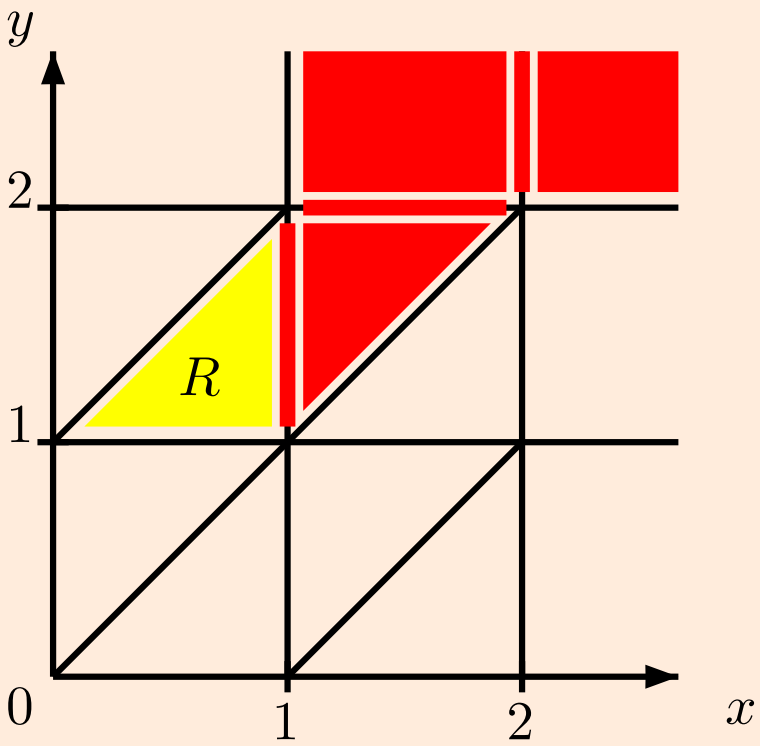
région R définie par
$I_x =]0;1[$, $I_y =]1;2[$
$frac(x) > frac(y)$

Successeur de R pour le temps
$I_x = [1;1]$, $I_y =]1;2[$

- Compatibilité de valuations équivalentes avec les contraintes $x \bowtie k$
- Compatibilité de valuations équivalentes avec l'écoulement du temps

Construction d'un quotient (2)

Vue géométrique avec deux horloges x et y, pour $m = 2$



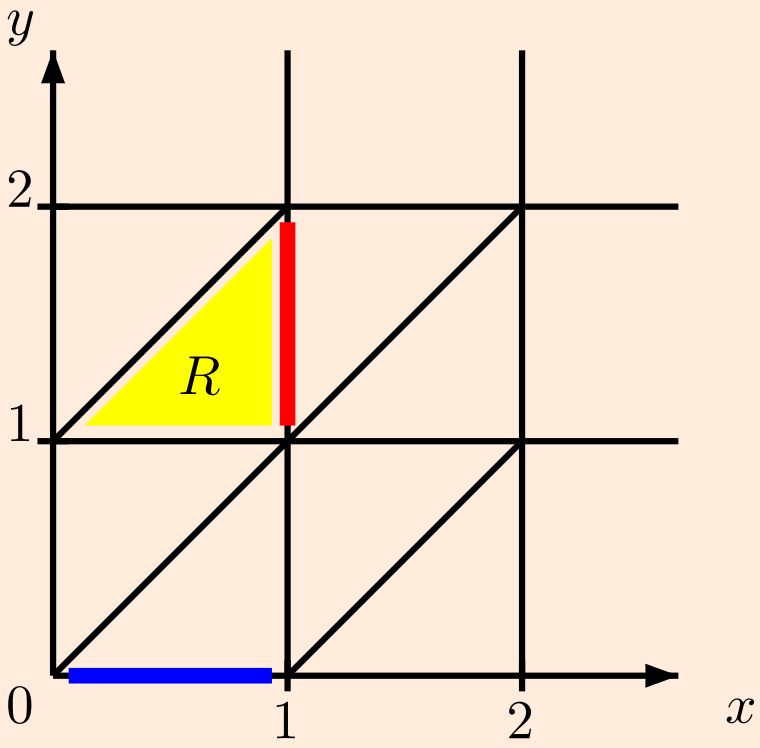
région R définie par
$I_x =]0;1[$, $I_y =]1;2[$
$frac(x) > frac(y)$

Successeur de R pour le temps
$I_x = [1;1]$, $I_y =]1;2[$

- Compatibilité de valuations équivalentes avec les contraintes $x \bowtie k$
- Compatibilité de valuations équivalentes avec l'écoulement du temps

Construction d'un quotient (2)

Vue géométrique avec deux horloges x et y, pour $m = 2$



région R définie par
$I_x =]0;1[$, $I_y =]1;2[$
$frac(x) > frac(y)$

Successeur de R pour le temps
$I_x = [1;1]$, $I_y =]1;2[$

Successeur de R pour une action
avec $y := 0$
$I_x =]0;1[$, $I_y = [0;0]$

- Compatibilité de valuations équivalentes avec les contraintes $x \bowtie k$
- Compatibilité de valuations équivalentes avec l'écoulement du temps

Construction d'un quotient (3)

Automate des régions $\mathcal{H}_{\mathcal{A}}$

pour l'automate temporisé $\mathcal{A} = (Q, q_0, \ell, Inv, \Delta)$,
avec ensemble d'horloges X et constante maximale m,
quotient $\mathbb{R}^X_{\geq 0}/\sim$ noté $\mathcal{R}_{X,m}$.

- états $Q \times \mathcal{R}_{X,M}$
- transitions de délai (abstraites) : $(q, R) \xrightarrow{\leq} (q, succ(R))$
- transitions d'action : $(q, R) \xrightarrow{a} (q', R')$
  s'il existe une transition $q \xrightarrow{g,a,r} q'$ de $\mathcal{A}$ avec $R \models g$ et $R' = R[r \mapsto 0]$

Nombre d'états

La taille de $\mathcal{R}_{X,m}$ est en $\mathcal{O}(|X|! \cdot m^{|X|})$, à multiplier par $\mid Q \mid$.

Construction d'un quotient (3)

Automate des régions $\mathcal{H}_{\mathcal{A}}$

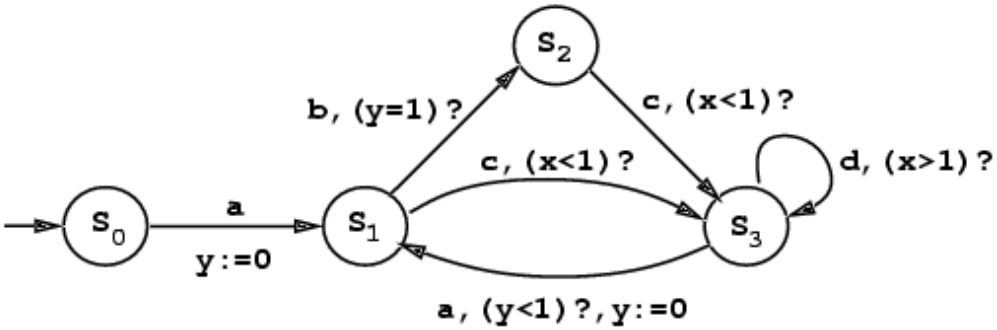
pour l'automate temporisé $\mathcal{A} = (Q, q_0, \ell, Inv, \Delta)$,
avec ensemble d'horloges X et constante maximale m,
quotient $\mathbb{R}^X_{\geq 0}/\sim$ noté $\mathcal{R}_{X,m}$.

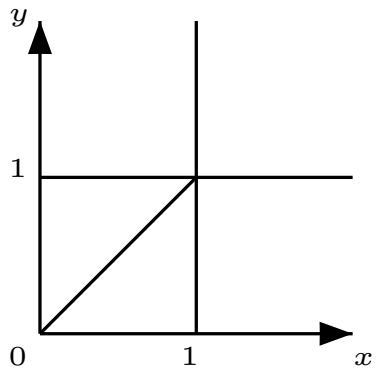
- états $Q \times \mathcal{R}_{X,M}$
- transitions de délai (abstraites) : $(q, R) \xrightarrow{\leq} (q, succ(R))$
- transitions d'action : $(q, R) \xrightarrow{a} (q', R')$
  s'il existe une transition $q \xrightarrow{g,a,r} q'$ de $\mathcal{A}$ avec $R \models g$ et $R' = R[r \mapsto 0]$

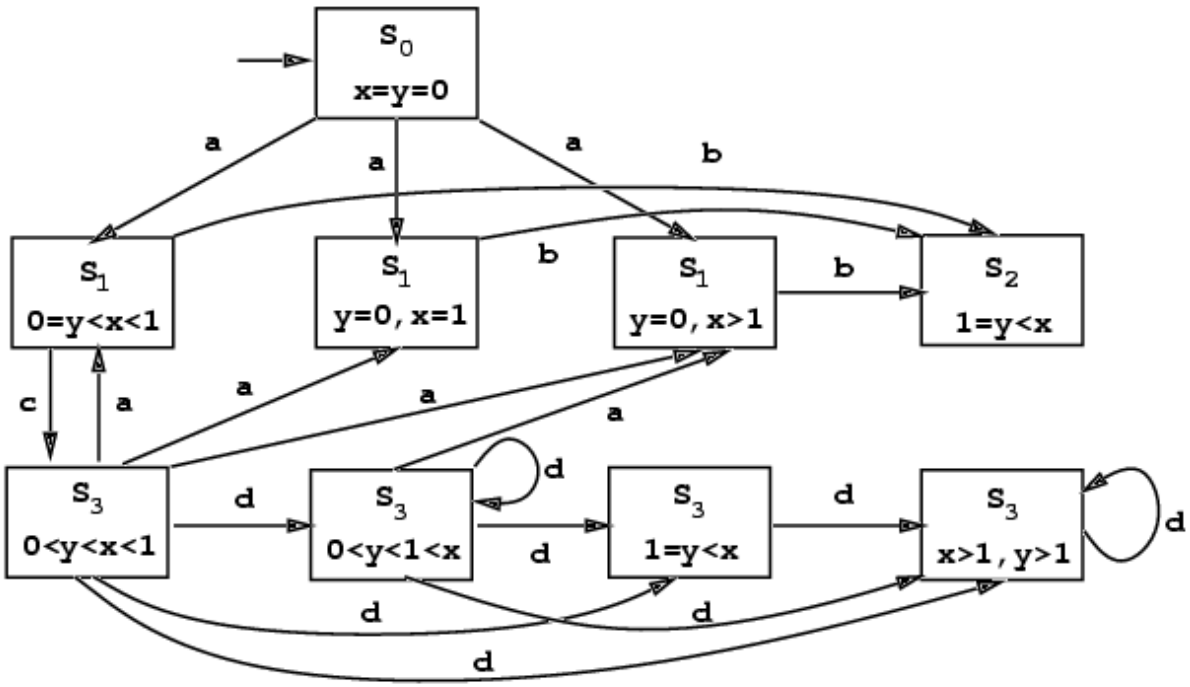
Nombre d'états

La taille de $\mathcal{R}_{X,m}$ est en $\mathcal{O}(|X|! \cdot m^{|X|})$, à multiplier par $| Q |$.

Exemple [Alur et Dill, 1990]







Autres résultats

- L'accessibilité d'un état de contrôle pour un automate temporisé $\mathcal{A}$ se ramène au test du vide sur $\mathcal{A}$.
  Ce problème reste PSPACE-complet, même pour des automates à 3 horloges, ou pour des automates dont les constantes sont dans $\{0, 1\}$.

- Résultats de décidabilité pour des variantes (TPTL, coFlat-MTL) ou une sémantique plus simple pour MTL, avec :
  - un temps discret [Alur et Henzinger, 1989, 1990],
  - une sémantique discrète [Ouaknine et Worrell, 2005],

  [Bouyer et al. 2005, 2007].

Autres résultats

- L'accessibilité d'un état de contrôle pour un automate temporisé $\mathcal{A}$ se ramène au test du vide sur $\mathcal{A}$.
  Ce problème reste PSPACE-complet, même pour des automates à 3 horloges, ou pour des automates dont les constantes sont dans $\{0, 1\}$.

- Résultats de décidabilité pour des variantes (TPTL, coFlat-MTL) ou une sémantique plus simple pour MTL, avec :
  - un temps discret [Alur et Henzinger, 1989, 1990],
  - une sémantique discrète [Ouaknine et Worrell, 2005],

  [Bouyer et al. 2005, 2007].

Model-checking de CTL

Theorème [Clarke et Emerson 1981], [Queille et Sifakis 1982]

Le problème du model-checking pour une formule φ de CTL sur une structure de Kripke $\mathcal{K}$ est décidable en temps $\mathcal{O}(|\mathcal{K}| \cdot |\varphi|)$.

Procédure de décision

Une procédure récursive de marquage des états de la structure de Kripke : pour chaque sous-formule ψ de φ et chaque état q, la procédure affecte vrai à $q \cdot \varphi$ si $q \models \psi$, et faux sinon.
Le marquage de φ est alors calculé selon les cas : φ est une proposition atomique, $\varphi = \neg\psi$, $\varphi = \psi_1 \wedge \psi_2$, $\varphi = \mathrm{EX}\psi$, etc.

Un cas simple : $\varphi = \mathrm{EX}\psi$

Marquage(ψ) ;
pour tout état $q \in Q$, $q \cdot \varphi :=$ faux ;
pour toute transition $q \to q'$, si $q'.\psi$ alors $q.\varphi :=$ vrai ;

Model-checking de CTL

Theorème [Clarke et Emerson 1981], [Queille et Sifakis 1982]

Le problème du model-checking pour une formule φ de CTL sur une structure de Kripke $\mathcal{K}$ est décidable en temps $\mathcal{O}(|\mathcal{K}| \cdot |\varphi|)$.

Procédure de décision

Une procédure récursive de marquage des états de la structure de Kripke : pour chaque sous-formule ψ de φ et chaque état q, la procédure affecte vrai à $q \cdot \varphi$ si $q \models \psi$, et faux sinon.
Le marquage de φ est alors calculé selon les cas : φ est une proposition atomique, $\varphi = \neg\psi$, $\varphi = \psi_1 \wedge \psi_2$, $\varphi = \mathsf{EX}\psi$, etc.

Un cas simple : $\varphi = \mathsf{EX}\psi$

Marquage(ψ) ;
pour tout état $q \in Q$, $q \cdot \varphi :=$ faux ;
pour toute transition $q \to q'$, si $q'.\psi$ alors $q.\varphi :=$ vrai ;

Model-checking de CTL

Theorème [Clarke et Emerson 1981], [Queille et Sifakis 1982]

Le problème du model-checking pour une formule φ de CTL sur une structure de Kripke $\mathcal{K}$ est décidable en temps $\mathcal{O}(|\mathcal{K}| \cdot |\varphi|)$.

Procédure de décision

Une procédure récursive de marquage des états de la structure de Kripke : pour chaque sous-formule ψ de φ et chaque état q, la procédure affecte vrai à $q \cdot \varphi$ si $q \models \psi$, et faux sinon.
Le marquage de φ est alors calculé selon les cas : φ est une proposition atomique, $\varphi = \neg\psi$, $\varphi = \psi_1 \wedge \psi_2$, $\varphi = \mathsf{EX}\psi$, etc.

Un cas simple : $\varphi = \mathsf{EX}\psi$

Marquage(ψ) ;
pour tout état $q \in Q$, $q \cdot \varphi :=$ faux ;
pour toute transition $q \to q'$, si $q'.\psi$ alors $q.\varphi :=$ vrai ;

Model-checking de TCTL

Theorème [Alur et al. 1993]

Le problème du model-checking de TCTL pour les automates temporisés est PSPACE-complet.

Procédure de décision

Elle utilise la propriété de l'équivalence :
si $v \sim v'$ alors pour toute formule φ de TCTL, $(q, v) \models \varphi \Leftrightarrow (q, v') \models \varphi$

Données : une formule φ de TCTL et un automate temporisé $\mathcal{A}$

Appliquer la procédure de marquage à une variante de l'automate des régions.

- L'automate des régions $\mathcal{H}$ est construit avec :
  l'ensemble d'horloges $X^* = X \cup \{z_\varphi\}$, pour une nouvelle horloge z_φ,
  la constante $M = \max(m_\varphi, m_\mathcal{A})$
  $m_\mathcal{A}$: constante maximale associée à $\mathcal{A}$
  m_φ : constante maximale dans les contraintes temporelles de φ.
- L'étiquetage utilise de nouvelles propositions atomiques $p_{\bowtie c}$:
  $p_{\bowtie c}$ est vraie dans une configuration si $z_\varphi \bowtie c$.

Model-checking de TCTL

Theorème [Alur et al. 1993]

Le problème du model-checking de TCTL pour les automates temporisés est PSPACE-complet.

Procédure de décision

Elle utilise la propriété de l'équivalence :
si $v \sim v'$ alors pour toute formule φ de TCTL, $(q, v) \models \varphi \Leftrightarrow (q, v') \models \varphi$

Données : une formule φ de TCTL et un automate temporisé $\mathcal{A}$

Appliquer la procédure de marquage à une variante de l'automate des régions.

- L'automate des régions $\mathcal{H}$ est construit avec :
  l'ensemble d'horloges $X^* = X \cup \{z_\varphi\}$, pour une nouvelle horloge z_φ,
  la constante $M = \max(m_\varphi, m_\mathcal{A})$
  $m_\mathcal{A}$: constante maximale associée à $\mathcal{A}$
  m_φ : constante maximale dans les contraintes temporelles de φ.
- L'étiquetage utilise de nouvelles propositions atomiques $p_{\bowtie c}$:
  $p_{\bowtie c}$ est vraie dans une configuration si $z_\varphi \bowtie c$.

Model-checking de TCTL

Theorème [Alur et al. 1993]

Le problème du model-checking de TCTL pour les automates temporisés est PSPACE-complet.

Procédure de décision

Elle utilise la propriété de l'équivalence :
si $v \sim v'$ alors pour toute formule φ de TCTL, $(q, v) \models \varphi \Leftrightarrow (q, v') \models \varphi$

Données : une formule φ de TCTL et un automate temporisé $\mathcal{A}$

Appliquer la procédure de marquage à une variante de l'automate des régions.

- L'automate des régions $\mathcal{H}$ est construit avec :
  l'ensemble d'horloges $X^* = X \cup \{z_\varphi\}$, pour une nouvelle horloge z_φ,
  la constante $M = \max(m_\varphi, m_\mathcal{A})$
  $m_\mathcal{A}$: constante maximale associée à $\mathcal{A}$
  m_φ : constante maximale dans les contraintes temporelles de φ.
- L'étiquetage utilise de nouvelles propositions atomiques $p_{\bowtie c}$:
  $p_{\bowtie c}$ est vraie dans une configuration si $z_\varphi \bowtie c$.

Model-checking de TCTL

Theorème [Alur et al. 1993]

Le problème du model-checking de TCTL pour les automates temporisés est PSPACE-complet.

Procédure de décision

Elle utilise la propriété de l'équivalence :
si $v \sim v'$ alors pour toute formule φ de TCTL, $(q, v) \models \varphi \Leftrightarrow (q, v') \models \varphi$

Données : une formule φ de TCTL et un automate temporisé $\mathcal{A}$

Appliquer la procédure de marquage à une variante de l'automate des régions.

- L'automate des régions $\mathcal{H}$ est construit avec :
  l'ensemble d'horloges $X^* = X \cup \{z_\varphi\}$, pour une nouvelle horloge z_φ,
  la constante $M = \max(m_\varphi, m_\mathcal{A})$
  $m_\mathcal{A}$: constante maximale associée à $\mathcal{A}$
  m_φ : constante maximale dans les contraintes temporelles de φ.
- L'étiquetage utilise de nouvelles propositions atomiques $p_{\bowtie c}$:
  $p_{\bowtie c}$ est vraie dans une configuration si $z_\varphi \bowtie c$.

Model-checking de TCTL

Theorème [Alur et al. 1993]

Le problème du model-checking de TCTL pour les automates temporisés est PSPACE-complet.

Procédure de décision

Elle utilise la propriété de l'équivalence :
si $v \sim v'$ alors pour toute formule φ de TCTL, $(q, v) \models \varphi \Leftrightarrow (q, v') \models \varphi$

Données : une formule φ de TCTL et un automate temporisé $\mathcal{A}$

Appliquer la procédure de marquage à une variante de l'automate des régions.

- L'automate des régions $\mathcal{H}$ est construit avec :
  l'ensemble d'horloges $X^* = X \cup \{z_\varphi\}$, pour une nouvelle horloge z_φ,
  la constante $M = \max(m_\varphi, m_\mathcal{A})$
  $m_\mathcal{A}$: constante maximale associée à $\mathcal{A}$
  m_φ : constante maximale dans les contraintes temporelles de φ.
- L'étiquetage utilise de nouvelles propositions atomiques $p_{\bowtie c}$:
  $p_{\bowtie c}$ est vraie dans une configuration si $z_\varphi \bowtie c$.

Exemple

pour la formule $\mathsf{AF}_{<3}\mathsf{ok}$

- Etats de $\mathcal{H}$: $(q, [v, \gamma])$ pour une configuration (q, v) de $\mathcal{A}$ et une classe étendue $[v, \gamma]$ où γ est une valeur de z_φ.
- Alors :

$$(q, v) \models \mathsf{AF}_{<3}\mathsf{ok} \;\Leftrightarrow\; (\mathsf{q}, [\mathsf{v}, 0]) \models \mathsf{AF}(\mathsf{ok} \wedge \mathsf{z}_\varphi < 3)$$

  A partir d'un état où la valeur de z_φ est égale à 0, on peut toujours atteindre un état où ok est vraie en même temps que la valeur de z_φ est strictement inférieure à 3.

- En utilisant la nouvelle proposition atomique $p_{<3}$, on obtient :

$$(q, v) \models \mathsf{AF}_{<3}\mathsf{ok} \;\Leftrightarrow\; (\mathsf{q}, [\mathsf{v}, 0]) \models \mathsf{AF}(\mathsf{ok} \wedge \mathsf{p}_{<3})$$

  ce qui ramène au model-checking de CTL sur une structure finie.

Exemple

pour la formule $\mathsf{AF}_{<3}\mathsf{ok}$

- Etats de $\mathcal{H}$: $(q, [v, \gamma])$ pour une configuration (q, v) de $\mathcal{A}$ et une classe étendue $[v, \gamma]$ où γ est une valeur de z_φ.
- Alors :

$$(q, v) \models \mathsf{AF}_{<3}\mathsf{ok} \;\Leftrightarrow\; (\mathsf{q}, [\mathsf{v}, 0]) \models \mathsf{AF}(\mathsf{ok} \wedge \mathsf{z}_\varphi < 3)$$

A partir d'un état où la valeur de z_φ est égale à 0, on peut toujours atteindre un état où ok est vraie en même temps que la valeur de z_φ est strictement inférieure à 3.

- En utilisant la nouvelle proposition atomique $p_{<3}$, on obtient :

$$(q, v) \models \mathsf{AF}_{<3}\mathsf{ok} \;\Leftrightarrow\; (\mathsf{q}, [\mathsf{v}, 0]) \models \mathsf{AF}(\mathsf{ok} \wedge \mathsf{p}_{<3})$$

ce qui ramène au model-checking de CTL sur une structure finie.

Exemple

pour la formule $\mathsf{AF}_{<3}\mathsf{ok}$

- Etats de $\mathcal{H}$: $(q, [v, \gamma])$ pour une configuration (q, v) de $\mathcal{A}$ et une classe étendue $[v, \gamma]$ où γ est une valeur de z_φ.
- Alors :

$$(q, v) \models \mathsf{AF}_{<3}\mathsf{ok} \;\Leftrightarrow\; (\mathsf{q}, [\mathsf{v}, 0]) \models \mathsf{AF}(\mathsf{ok} \wedge \mathsf{z}_\varphi < 3)$$

  A partir d'un état où la valeur de z_φ est égale à 0, on peut toujours atteindre un état où ok est vraie en même temps que la valeur de z_φ est strictement inférieure à 3.

- En utilisant la nouvelle proposition atomique $p_{<3}$, on obtient :

$$(q, v) \models \mathsf{AF}_{<3}\mathsf{ok} \;\Leftrightarrow\; (\mathsf{q}, [\mathsf{v}, 0]) \models \mathsf{AF}(\mathsf{ok} \wedge \mathsf{p}_{<3})$$

  ce qui ramène au model-checking de CTL sur une structure finie.

Complexité

- La complexité de la vérification est plus élevée dans le cas temporisé. L'ajout d'une horloge revient à l'ajout d'un composant :

Vérification	$\mathcal{T} \models \varphi$	"syst. non plat" $\models \varphi$
Accessibilité	NLOGSPACE-C	PSPACE-C
model-checking de CTL	P-C	PSPACE-C

- Des problèmes souvent même indécidables : model-checking de MTL, satisfaisabilité de TCTL, ou extensions du modèle des automates temporisés.

Des algorithmes plus simples

par restriction : logique $\mathrm{TCTL}_{\leq,\geq}$ (sans égalité)

- pour des structures de Kripke avec durées, model-checking en temps polynomial en $|\mathcal{K}| \cdot |\varphi|$ [Laroussinie et al. 2002]
- pour des automates temporisés à une seule horloge, model-checking P-complet [Laroussinie et al. 2004]

Complexité

- La complexité de la vérification est plus élevée dans le cas temporisé. L'ajout d'une horloge revient à l'ajout d'un composant :

Vérification	$\mathcal{T} \models \varphi$	"syst. non plat" $\models \varphi$
Accessibilité	NLOGSPACE-C	PSPACE-C
model-checking de CTL	P-C	PSPACE-C

- Des problèmes souvent même indécidables : model-checking de MTL, satisfaisabilité de TCTL, ou extensions du modèle des automates temporisés.

Des algorithmes plus simples

par restriction : logique $\mathrm{TCTL}_{\leq,\geq}$ (sans égalité)

- pour des structures de Kripke avec durées, model-checking en temps polynomial en $|\mathcal{K}| \cdot |\varphi|$ [Laroussinie et al. 2002]
- pour des automates temporisés à une seule horloge, model-checking P-complet [Laroussinie et al. 2004]

Complexité

- La complexité de la vérification est plus élevée dans le cas temporisé. L'ajout d'une horloge revient à l'ajout d'un composant :

Vérification	$\mathcal{T} \models \varphi$	"syst. non plat" $\models \varphi$
Accessibilité	NLOGSPACE-C	PSPACE-C
model-checking de CTL	P-C	PSPACE-C

- Des problèmes souvent même indécidables : model-checking de MTL, satisfaisabilité de TCTL, ou extensions du modèle des automates temporisés.

Des algorithmes plus simples

par restriction : logique $\mathrm{TCTL}_{\leq,\geq}$ (sans égalité)

- pour des structures de Kripke avec durées, model-checking en temps polynomial en $|\mathcal{K}| \cdot |\varphi|$ [Laroussinie et al. 2002]
- pour des automates temporisés à une seule horloge, model-checking P-complet [Laroussinie et al. 2004]

Complexité

- La complexité de la vérification est plus élevée dans le cas temporisé. L'ajout d'une horloge revient à l'ajout d'un composant :

Vérification	$\mathcal{T} \models \varphi$	"syst. non plat" $\models \varphi$
Accessibilité	NLOGSPACE-C	PSPACE-C
model-checking de CTL	P-C	PSPACE-C

- Des problèmes souvent même indécidables : model-checking de MTL, satisfaisabilité de TCTL, ou extensions du modèle des automates temporisés.

Des algorithmes plus simples

par restriction : logique $\text{TCTL}_{\leq,\geq}$ (sans égalité)

- pour des structures de Kripke avec durées, model-checking en temps polynomial en $|\mathcal{K}| \cdot |\varphi|$ [Laroussinie et al. 2002]
- pour des automates temporisés à une seule horloge, model-checking P-complet [Laroussinie et al. 2004]

Complexité

- La complexité de la vérification est plus élevée dans le cas temporisé. L'ajout d'une horloge revient à l'ajout d'un composant :

Vérification	$\mathcal{T} \models \varphi$	"syst. non plat" $\models \varphi$
Accessibilité	NLOGSPACE-C	PSPACE-C
model-checking de CTL	P-C	PSPACE-C

- Des problèmes souvent même indécidables : model-checking de MTL, satisfaisabilité de TCTL, ou extensions du modèle des automates temporisés.

Des algorithmes plus simples

par restriction : logique $\text{TCTL}_{\leq,\geq}$ (sans égalité)

- pour des structures de Kripke avec durées, model-checking en temps polynomial en $|\mathcal{K}| \cdot |\varphi|$ [Laroussinie et al. 2002]
- pour des automates temporisés à une seule horloge, model-checking P-complet [Laroussinie et al. 2004]

En pratique

De nombreux outils

souvent efficaces malgré la complexité :

- KRONOS, UPPAAL, pour les automates temporisés
- HCMC, HYTECH, pour des automates temporisés étendus (hybrides)
- TSMV pour les structures de Kripke avec durées
- Roméo, TINA, pour les réseaux de Petri temporels
- ...

Des structures de données adaptées

- au codage des régions ou des zones : les DBM (Difference Bounded Matrices) et des variantes (CDD, NDD, etc.)
- au codage de polyèdres

Des heuristiques pour les algorithmes

- analyse à la volée
- méthodes compositionnelles
- résolution de contraintes

En pratique

De nombreux outils

souvent efficaces malgré la complexité :

- KRONOS, UPPAAL, pour les automates temporisés
- HCMC, HYTECH, pour des automates temporisés étendus (hybrides)
- TSMV pour les structures de Kripke avec durées
- Roméo, TINA, pour les réseaux de Petri temporels
- ...

Des structures de données adaptées

- au codage des régions ou des zones : les DBM (Difference Bounded Matrices) et des variantes (CDD, NDD, etc.)
- au codage de polyèdres

Des heuristiques pour les algorithmes

- analyse à la volée
- méthodes compositionnelles
- résolution de contraintes

En pratique

De nombreux outils

souvent efficaces malgré la complexité :

- KRONOS, UPPAAL, pour les automates temporisés
- HCMC, HYTECH, pour des automates temporisés étendus (hybrides)
- TSMV pour les structures de Kripke avec durées
- Roméo, TINA, pour les réseaux de Petri temporels
- ...

Des structures de données adaptées

- au codage des régions ou des zones : les DBM (Difference Bounded Matrices) et des variantes (CDD, NDD, etc.)
- au codage de polyèdres

Des heuristiques pour les algorithmes

- analyse à la volée
- méthodes compositionnelles
- résolution de contraintes

Plan

Conclusion

De nombreux autres travaux dans ce domaine

- pour d'autres modèles et d'autres langages de spécification
- pour des extensions quantitatives avec poids, coûts, probabilités, etc.
- en relation avec les problèmes de contrôle et les jeux.

Perspectives

Théorique : affiner les frontières de la décidabilité.
Pratique : contourner le problème de l'explosion combinatoire.

- spécifications et modèles adaptés à des contextes particuliers, avec des algorithmes de vérification plus simples
- structures de données adaptées au mélange continu + discret
- techniques d'abstraction

Conclusion

De nombreux autres travaux dans ce domaine

- pour d'autres modèles et d'autres langages de spécification
- pour des extensions quantitatives avec poids, coûts, probabilités, etc.
- en relation avec les problèmes de contrôle et les jeux.

Perspectives

Théorique : affiner les frontières de la décidabilité.
Pratique : contourner le problème de l'explosion combinatoire.

- spécifications et modèles adaptés à des contextes particuliers, avec des algorithmes de vérification plus simples
- structures de données adaptées au mélange continu + discret
- techniques d'abstraction

MERCI