LIP6 - UPMC
Master SAR - VFSR

Année 2009–2010
Aide mémoire

Vérification formelle de systèmes répartis

Table des matières

1 Introduction : problématique de la vérification de systèmes réactifs **1**

2 Systèmes de transitions et logiques temporelles **1**
2.1 Structures de Kripke 1
2.2 La logique LTL 2
2.3 La logique CTL 3
2.4 Comparaison 4

3 Automates temporisés **5**
3.1 Systèmes de transitions temporisés 5
3.2 Automates temporisés 6
3.3 Régions et zones 8

1 Introduction : problématique de la vérification de systèmes réactifs

2 Systèmes de transitions et logiques temporelles

La définition usuelle des systèmes de transitions est étendue en ajoutant un étiquetage des états par des propositions atomiques.

2.1 Structures de Kripke

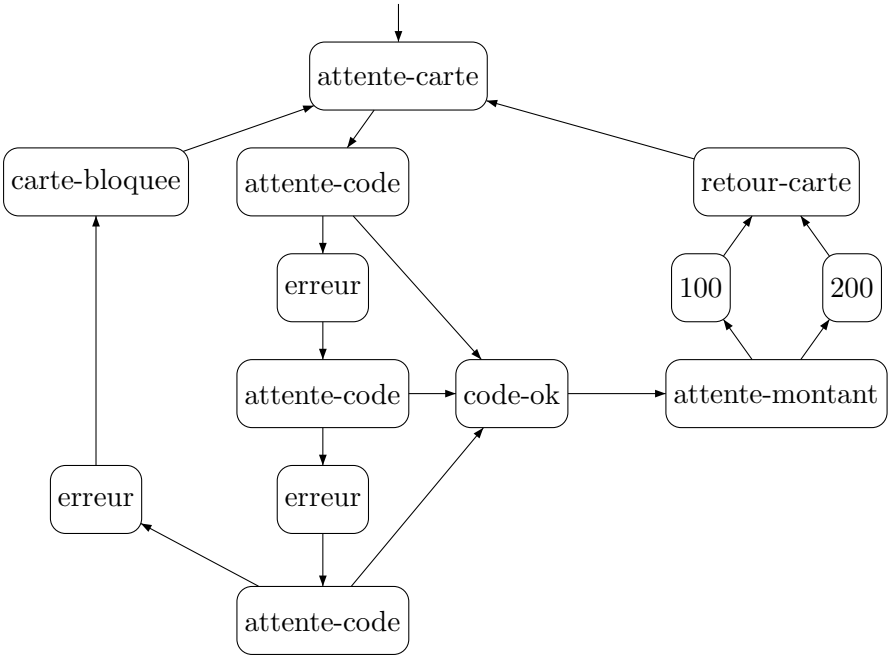
Définition 2.1.
Une structure de Kripke est un quadruplet $\mathcal{K} = (Q, q_0, \Delta, \ell)$ pour lequel $(Q, \{q_0\}, \Delta)$ est un système de transitions fini sur un alphabet d'actions Act (avec Q l'ensemble des états, q_0 l'état initial et Δ l'ensemble des transitions), auquel on adjoint un ensemble de propositions atomiques Prop et une fonction $\ell : S \mapsto 2^{Prop}$ d'étiquetage des configurations par des ensembles de propositions atomiques.

Les actions du système sont très souvent omises (l'ensemble *Act* peut donc être réduit à un singleton), l'attention étant portée principalement sur les propriétés des états. Dans la plupart des cas, on suppose donc que l'ensemble Δ des transitions est un sous-ensemble de $Q \times Q$ au lieu de $Q \times Act \times Q$.

Dans l'exemple ci-dessous, qui décrit un distributeur automatique de billets, les noms des états $(q_0, q_1, \ldots)$ n'ont pas été indiqués, et il y a seulement une proposition atomique par état.

Les logiques temporelles du temps linéaire LTL (Linear Temporal Logic, définie par Pnueli en 1977) et du temps arborescent CTL (Computation Tree Logic, définie par Clarke et Emerson

en 1981) sont deux fragments très fréquemment étudiés d'une logique plus générale, CTL*, introduite plus tardivement (en 1986). Ces logiques s'interprètent sur des structures de Kripke et on considère dans la suite une telle structure $\mathcal{K} = (Q, q_0, \Delta, \ell)$, sur un ensemble Prop de propositions atomiques, où Δ est un sous-ensemble de $Q \times Q$. Dans ce cadre, on s'intéresse aux exécutions infinies de $\mathcal{K}$, qui sont donc des suites infinies d'états $\sigma : q_0 \to q_1 \to q_2 \cdots$ commençant dans l'état initial.



2.2 La logique LTL

Les formules de LTL sont définies par la grammaire :

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

où $P \in$ Prop.

Une formule de LTL s'interpète sur une position i d'une exécution infinie partant de l'état initial : $\sigma : q_0 \to q_1 \to q_2 \cdots$. Pour $i \geq 0$, on note $\sigma(i)$ le $i$ème état de σ (donc ici q_i). La sémantique de LTL est alors définie inductivement par :

$\sigma, i \models P$	si P est dans l'ensemble $\ell(\sigma(i))$ des étiquettes de q
$\sigma, i \models \neg\varphi$	si σ, i ne satisfait pas φ
$\sigma, i \models \varphi \wedge \psi$	si q satisfait à la fois φ et ψ
$\sigma, i \models \mathsf{X}\varphi$	si $\sigma, i+1 \models \varphi$
$\sigma, i \models \varphi \mathsf{U} \psi$	s'il existe $j \geq i$ tel que $\sigma, j \models \psi$ et pour tout k, $i \leq k < j, \sigma, k \models \varphi$

Ainsi, la modalité X, appelée également next, s'interprète naturellement de la façon suivante : $\mathsf{X}\varphi$ est vraie dans une position si φ est vraie dans la position suivante. La formule $\varphi \mathsf{U} \psi$ est vraie dans une position s'il existe une position dans le futur où ψ est vraie, φ restant vraie

jusque là. Outre les opérateurs booléens usuels comme $\vee$ ou $\Rightarrow$, les abréviations suivantes sont couramment utilisées :

- $\mathsf{F}\varphi$, définie par $\mathsf{true}\mathsf{U}\varphi$, exprime que, à partir de la position considérée, φ sera vraie *un jour* (la formule true étant satisfaite dans tout état),
- $\mathsf{G}\varphi$, définie par dualité comme $\neg\mathsf{F}(\neg\varphi)$, exprime que φ est toujours vraie à partir de la position considérée

Exemple : la formule de LTL exprimant que tout signal est suivi d'une réponse est :

$$\mathsf{G}(\mathsf{signal} \Rightarrow \mathsf{F}\ \mathsf{reponse})$$

Etant données une structure de Kripke $\mathcal{K}$ et une formule φ de LTL, on dit que $\mathcal{K}$ satisfait φ, noté $\mathcal{K} \models \varphi$, si l'état initial satisfait φ, c'est-à-dire que pour toute exécution σ de $\mathcal{K}$, on a $\sigma, 0 \models \varphi$.

Le problème du model checking de LTL s'exprime de la façon suivante :
Données : une stucture de Kripke $\mathcal{K}$ et une formule φ de LTL
Question : $\mathcal{K}$ satisfait-elle φ ?

2.3 La logique CTL

Les formules de CTL sont définies par :

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

où $P \in \mathsf{Prop}$.

Une formule de CTL s'interprète sur un état $q \in Q$ de la structure de Kripke. Notons $Exec(q)$ l'arbre des exécutions de $\mathcal{K}$ issues de q. Pour une telle exécution $\sigma : q \to q_1 \to q_2 \cdots$ et pour $i \geq 0$, on note encore $\sigma(i)$ le $i$ème état de σ et on définit la sémantique de manière inductive par :

$q \models P$	si P est dans l'ensemble $\ell(q)$ des étiquettes de q
$q \models \neg\varphi$	si q ne satisfait pas φ
$q \models \varphi \wedge \psi$	si q satisfait à la fois φ et ψ
$q \models \mathsf{EX}\varphi$	s'il existe une exécution σ de $Exec(q)$ telle que $\sigma(1) \models \varphi$
$q \models \mathsf{AX}\varphi$	si toute exécution σ de $Exec(q)$ est telle que $\sigma(1) \models \varphi$
$q \models \mathsf{E}\varphi\mathsf{U}\psi$	s'il existe une exécution σ de $Exec(q)$ et un entier $j \geq 0$ tels que $\sigma(j) \models \psi$ et pour tout $k,\ 0 \leq k < j,\ \sigma(k) \models \varphi$
$q \models \mathsf{A}\varphi\mathsf{U}\psi$	si, pour toute exécution σ de $Exec(q)$ il existe un entier $j \geq 0$ tel que $\sigma(j) \models \psi$ et pour tout $k,\ 0 \leq k < j,\ \sigma(k) \models \varphi$

Dans ces formules, E et A sont respectivement les quantificateurs existentiel et universel sur les exécutions. Ainsi, le fait que l'état q satisfasse la formule $\mathsf{A}\varphi\mathsf{U}\psi$ exprime que ψ sera vraie *plus loin* dans toute exécution partant de q et que φ restera vraie entre temps.

Les abréviations courantes sont par exemple :

– $\mathsf{EF}\varphi$, définie par $\mathsf{E}(\mathsf{true})\mathsf{U}\varphi$, exprime, pour un état q, que la propriété φ sera vraie *un jour* dans au moins une exécution issue de q. Il s'agit donc de l'*accessibilité* d'un état satisfaisant φ.
– $\mathsf{AG}\varphi$, définie par dualité comme $\neg\mathsf{EF}(\neg\varphi)$, exprime, pour un état q, que la propriété φ est *toujours* vraie le long de toute exécution partant de q,
– $\mathsf{AF}\varphi$, définie par $\mathsf{A}(\mathsf{true})\mathsf{U}\varphi$, exprime, pour un état q, que pour toute exécution issue de q, la propriété φ sera vraie *un jour*. La propriété φ est alors dite *inévitable*.

Exemple : la formule correspondant au temps de réponse s'exprime en CTL par :

$$\mathsf{AG}(\mathsf{signal} \Rightarrow \mathsf{AF}\ \mathsf{reponse}).$$

Etant données une structure de Kripke $\mathcal{K}$ et une formule φ de CTL, on dit que $\mathcal{K}$ satisfait φ, noté $\mathcal{K} \models \varphi$, si l'état initial satisfait φ, c'est-à-dire que $q_0 \models \varphi$.

Le problème du model checking de CTL s'exprime de manière similaire par :
Données : une stucture de Kripke $\mathcal{K}$ et une formule φ de LTL
Question : $\mathcal{K}$ satisfait-elle φ ?

2.4 Comparaison

Contrairement à ce que pourrait laisser supposer l'exemple des temps de réponse, les deux logiques LTL et CTL ont des pouvoirs d'expression incomparables. Ces logiques sont deux fragments de CTL^* qui contient des formules d'états utilisant des formules de chemin, avec la syntaxe suivante.

Formules de CTL^* (formules d'état) :

$$\varphi_e, \psi_e ::= P \mid \neg\varphi_e \mid \varphi_e \wedge \psi_e \mid \mathsf{E}\varphi_c \mid \mathsf{A}\varphi_c$$

où φ_c est une formule de chemin, avec pour les formules de chemin :

$$\varphi_c, \psi_c ::= \varphi_e \mid \neg\varphi_c \mid \varphi_c \wedge \psi_c \mid \mathsf{X}\varphi_\mathsf{c} \mid \varphi_\mathsf{c}\mathsf{U}\psi_\mathsf{c}$$

La sémantique se déduit facilement de ce qui a été vu précédemment.

Enfin, concernant les formules des temps de réponse pour ces deux logiques, notons que la durée de réaction ne peut pas être précisée, ce qui a conduit à proposer l'ajout de contraintes quantitatives dans les formules.

3 Automates temporisés

3.1 Systèmes de transitions temporisés

On considère un domaine de temps $\mathbb{T}$ qui peut être $\mathbb{N}$, $\mathbb{Q}$ ou $\mathbb{R}$ et qui sera toujours plongé dans $\mathbb{R}$.

Définition 3.1.
Un système de transitions temporisé est un système de transitions $\mathcal{T} = (S, S_0, T)$ sur l'alphabet $Act \cup \mathbb{T}$. L'ensemble T des transitions est donc un sous-ensemble de $S \times (Act \cup \{\varepsilon\} \cup \mathbb{T}) \times S$, avec deux types de transitions :

- *les transitions d'actions $s \xrightarrow{a} s'$ avec $a \in Act \cup \{\varepsilon\}$,*
- *les transitions de délais $s \xrightarrow{d} s'$ avec $d \in \mathbb{T}$.*

Les transitions $\xrightarrow{d}$, pour $d \in \mathbb{T}$, expriment l'écoulement d'une durée d et doivent de ce fait vérifier des conditions particulières, qui traduisent la compatibilité du système par rapport aux opérations sur le temps :

- *délai nul :* $s \xrightarrow{0} s'$ si et seulement si $s' = s$,
- *additivité :* si $s \xrightarrow{d} s'$ et $s' \xrightarrow{d'} s''$, alors $s \xrightarrow{d+d'} s''$.

De plus, il est parfois exigé du système $\mathcal{T}$ qu'il soit :

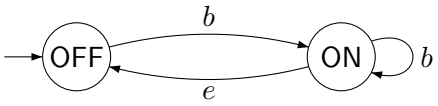
- *déterministe par rapport au temps* : $s \xrightarrow{d} s_1$ et $s \xrightarrow{d} s_2$ implique $s_1 = s_2$,
- *continu* : $s \xrightarrow{d} s'$ implique que, pour tous d' et d'' tels que $d = d' + d''$, il existe s'' tel que $s \xrightarrow{d'} s''$ et $s'' \xrightarrow{d''} s'$.

Dans ce cas, une exécution de $\mathcal{T}$ est un chemin $\rho = s_0 \xrightarrow{d_1} s'_0 \xrightarrow{a_1} s_1 \xrightarrow{d_2} s'_1 \xrightarrow{a_2} \cdots$ partant d'une configuration initiale et où les durées alternent strictement avec les actions. A une telle exécution peuvent être associés :

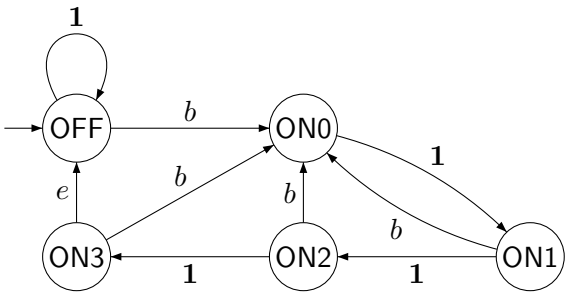
- la suite $(t_i)_{i \geq 0}$ des dates absolues, qui est naturellement donnée par $t_0 = 0$ et $t_i = \sum_{j=0}^{i} d_j$,
- le *mot temporisé* $w = (a_1, t_1)(a_2, t_2)\ldots$, qui correspond à l'observation de la suite des événements, a_i se produisant à la date t_i. C'est donc un mot sur l'alphabet $Act \times \mathbb{T}$ (les paires correspondant à des actions non observables ayant été absorbées).

Le langage associé à un système de transitions temporisé $\mathcal{T}$ est alors l'ensemble des mots temporisés associés à ses exécutions.

Exemple : un interrupteur.



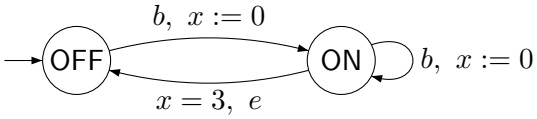
On souhaite ajouter la contrainte : *la lumière reste allumée 3 unités de temps après appui sur le bouton.* Un premier système peut être obtenu par un dépliage en utilisant un temps discret, avec un risque d'explosion combinatoire dû aux constantes impliquées.



3.2 Automates temporisés

Les automates temporisés sont des automates finis qui manipulent des variables appelées horloges, car elles évoluent de manière synchrone avec le temps dans les états. Ces horloges peuvent être testées et remises à zéro lors des transitions.

Ainsi, avec une horloge x, l'interrupteur serait décrit par l'automate temporisé suivant, qui reprend simplement l'automate fini original en le décorant :



On peut aussi adjoindre à l'état ON la contrainte $x \leq 3$ sur l'horloge x pour restreindre la sémantique aux exécutions pour lesquelles la valeur de l'horloge x ne peut pas dépasser 3 dans cet état.

Une exécution dans ce modèle est décrite par l'évolution des états et celle des horloges. Dans l'exemple, on pourrait avoir :

$$(\mathsf{OFF}, 0) \xrightarrow{4.6} (\mathsf{OFF}, 4.6) \xrightarrow{b} (\mathsf{ON}, 0) \xrightarrow{1.7} (\mathsf{ON}, 1.7) \xrightarrow{b} (\mathsf{ON}, 0) \xrightarrow{3} (\mathsf{ON}, 3) \xrightarrow{e} (\mathsf{OFF}, 3)$$

De façon générale, pour un ensemble X d'horloges, on note $\mathcal{C}(X)$ l'ensemble des conjonctions de contraintes atomiques de la forme $x \bowtie c$ où x est une horloge, c une constante (généralement entière) et $\bowtie$ un opérateur de comparaison dans $\{<, \leq, =, \geq, >\}$.

Définition 3.2.
Un automate temporisé *sur un alphabet Act est un quintuplet $\mathcal{A} = (X, Q, q_0, \Delta, Inv)$, où :*
- *X est un ensemble d'horloges,*
- *Q est un ensemble fini d'états (ou modes de contrôle),*
- *$q_0 \in Q$ est l'état initial,*
- *Δ est un sous-ensemble de $Q \times \mathcal{C}(X) \times Act \times 2^X \times Q$,*
- *$Inv : Q \mapsto \mathcal{C}(X)$ associe à chaque état une contrainte de $\mathcal{C}(X)$, appelée* invariant*, formée en utilisant seulement les opérateurs $<$ et $\leq$.*

Une transition de Δ, notée $q \xrightarrow{g,a,r} q'$, exprime un passage possible de q à q' avec l'action a, si la garde g est vraie. Les horloges de $r \subseteq X$ sont alors remises à zéro, ce qui est souvent noté aussi $r := 0$, comme c'est le cas dans l'exemple pour la remise à zéro de l'horloge x.

Les contraintes d'horloges (gardes et invariants) sont interprétées sur des valuations d'horloges, c'est-à-dire des applications de X dans $\mathbb{R}_+$, avec $\mathbf{0}$ représentant la valuation nulle pour toutes les horloges de X. Une valuation v satisfait une contrainte g de la forme $x \bowtie c$, noté $v \models g$, si $v(x) \bowtie c$.

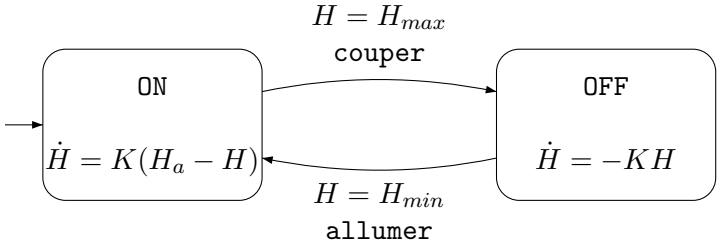
On définit :

– le passage du temps depuis une valuation v par :
$(v + d)(x) = v(x) + d$ pour toute horloge x et tout $d \in \mathbb{T}$, ce qui exprime l'évolution synchrone des valeurs d'horloge lorsqu'une durée d s'écoule,
– la remise à zéro des horloges de $r \subseteq X$ à partir de v par : $v[r \mapsto 0](x) = 0$ si $x \in r$ et 0 sinon.

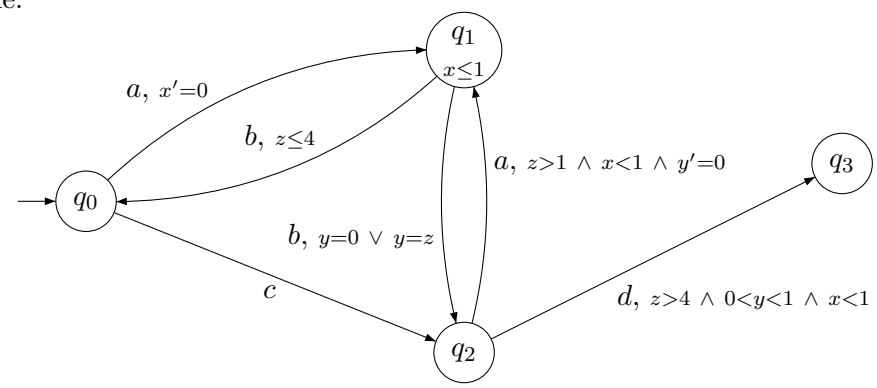
La sémantique d'un automate temporisé $\mathcal{A} = (X, Q, q_0, \Delta, Inv)$ est donnée par un système de transitions temporisé $\mathcal{T}_\mathcal{A} = (S, \{s_0\}, T)$, sur $Act \cup \{\varepsilon\} \cup \mathbb{T}$, avec :
– $S = \{(q, v) \in Q \times \mathbb{R}_+^X \mid v \models Inv(q)\}$. Les configurations sont donc des couples (q, v) où $q \in Q$ et v est une valuation d'horloges satisfaisant l'invariant de l'état.
– La configuration initiale est $s_0 = (q_0, \mathbf{0})$.
– Les transitions de T sont :
• soit $(q, v) \xrightarrow{d} (q, v + d)$, une transition de durée $d \in \mathbb{T}$, possible si $v + d \models Inv(q)$,
• soit $(q, v) \xrightarrow{a} (q', v')$, une transition discrète d'étiquette $a \in Act \cup \{\varepsilon\}$, avec $v' = v[r \mapsto 0]$, possible s'il existe $q \xrightarrow{g,a,r} q'$ dans Δ telle que la valuation v satisfasse la garde g.

Remarque. Ces définitions peuvent être généralisées à des modèles plus puissants, comme les automates hybrides :
– en considérant des gardes plus complexes $\sum_{i=1}^n a_i x_i + b \bowtie 0$,
– ou en considérant des mises à jour au lieu de simples remises à zéro : $x := \sum_{i=1}^n a_i x_i + b$,
– ou encore en considérant des variables dont l'évolution est plus générale que celle des horloges (qui sont affines de pente égale à 1).

Un exemple classique de la littérature est celui du thermostat :



Exercice 9. On considère l'automate hybride linéaire $\mathcal{A}_1$ ci-dessous. Les variables x et y sont des horloges (pente 1 partout) tandis que z a pour pente 1 dans q_1 et 0 dans les autres états de contrôle.



L'état de contrôle q_3 est-il accessible depuis la configuration initiale $(q_0, 0, 0, 0)$? Si oui, donnez une exécution menant à une configuration $(q_3, -, -, -)$.

3.3 Régions et zones

Une méthode classique pour analyser les systèmes de transitions infinis, en particulier pour résoudre le problème de l'accessibilité, consiste à en obtenir un quotient que l'on peut traiter, et qui conserve néanmoins suffisamment d'informations. C'est notamment le cas lorsqu'on peut construire un quotient fini, relié par une équivalence au système de départ.

Considérons un automate temporisé $\mathcal{A} = (X, Q, q_0, \Delta, Inv)$, sur l'aphabet Act avec domaine de temps $\mathbb{T} = \mathbb{R}_+$. La structure de contrôle étant finie, elle peut être conservée telle quelle, et il faut construire une partition de l'ensemble $\mathbb{R}_+^X$ des valuations, telle que la relation d'équivalence $\sim$ associée soit compatible avec les contraintes de $\mathcal{C}(X)$, la progression du temps et les remises à zéro. Pour deux valuations v et v', notons $v \leq v'$ s'il existe d tel que $v' = v + d$. Les propriétés de compatibilité s'expriment alors, pour deux valuations équivalentes $v \sim v'$ par les conditions (*) suivantes :

- pour toute contrainte $g : \ x \bowtie c$, $v \models g$ si et seulement si $v' \models g$,
- si $v \leq v_1$ pour une valuation v_1, alors il existe une valuation v_1' telle que $v' \leq v_1'$ et $v_1 \sim v_1'$,
- pour tout sous ensemble r d'horloges, $v[r \mapsto 0] \sim v'[r \mapsto 0]$.

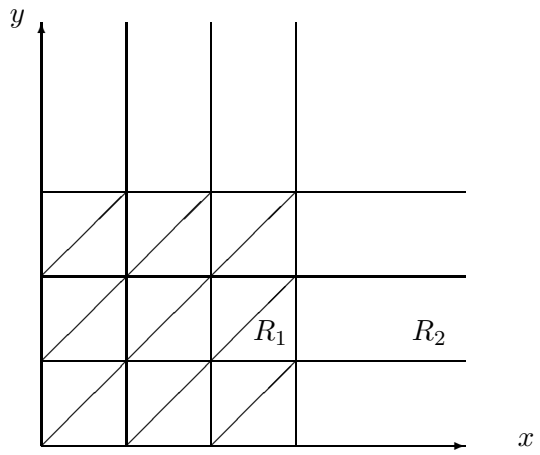
Supposons qu'un tel quotient $\mathcal{R} = (\mathbb{R}_+^X)_{/\sim}$ puisse être construit, nous appelons *région* un élément de $\mathcal{R}$. Les opérations $\leq$ et les remises à zéro peuvent alors être définies sur $\mathcal{R}$ et on peut aussi étendre la définition $R \models g$ pour une région R et une garde g. En réalisant un produit synchronisé avec $\mathcal{A}$, on obtient un nouveau système bisimilaire (au sens (*)) au système de transitions $\mathcal{T}_\mathcal{A} = (S, \{s_0\}, T)$ associé à $\mathcal{A}$: le système de transitions $\mathcal{K}_\mathcal{A} = (Q \times \mathcal{R}, \{(q_0, \mathbf{0})\}, D)$ sur $Act \cup \{\leq\}$. Les états sont de la forme $(q, [v])$ où $[v]$ est la classe d'équivalence d'une valuation v et les transitions de D sont de la forme :

1. $(q, R) \xrightarrow{a} (q', R')$ s'il existe une transition $q \xrightarrow{g,a,r} q' \in \Delta$ avec $R \models g$ et $R' = R[r \mapsto 0]$
2. $(q, R) \xrightarrow{\leq} (q, R')$ si $R \leq R'$.

Dans un article qui a fait date en 1990, Alur et Dill construisent une telle relation pour les automates temporisés, qui est d'index fini. Cette relation utilise la constante maximale m apparaissant dans les contraintes de $\mathcal{A}$, et le quotient est noté $\mathcal{R}_{(X,m)}$. On pose $v \sim v'$ (**) si :

- pour toute horloge x, ou bien les parties entières de $v(x)$ et de $v'(x)$ sont égales, ou bien $v(x) > m$ et $v'(x) > m$,
- pour toute horloge x telle que $v(x) \leq m$, $frac(v(x)) = 0$ si et seulement si $frac(v'(x)) = 0$, où $frac(d)$ désigne la partie fractionnaire du réel d,
- pour tout couple (x, y) d'horloges telles que $v(x) \leq m$ et $v'(x) \leq m$, $frac(v(x)) \leq frac(v(y))$ si et seulement si $frac(v'(x)) \leq frac(v'(y))$.

Pour un automate à deux horloges, avec $m = 3$, on obtient la partition représentée dans la figure ci-dessous. Les régions sont les points à coordonnées entières, les segments ouverts entre deux tels points, les intérieurs des triangles délimités par les segments, les demi-droites ouvertes et les régions ouvertes délimitées par ces demi-droites. Ainsi, les régions notées R_1 et R_2 sur cette figure peuvent être décrites par : $R_1 : (2 < x < 3) \wedge (1 < y < 2) \wedge (0 < frac(y) < frac(x))$ et $R_2 : (x > 3) \wedge (1 < y < 2)$.



Notons que dans la définition des transitions de D, on peut aussi restreindre les transitions de type 2 (correspondant au passage du temps) à celles qui lient une région et son successeur immédiat.

On constate une explosion de la taille de $\mathcal{K}_\mathcal{A}$ puisque le cardinal de $\mathcal{R}_{(X,m)}$ est en $O(|X|! \cdot m^{|X|})$.

Le système $\mathcal{K}_\mathcal{A}$ ainsi obtenu est donc un automate fini, appelé *automate des régions*. Les conditions (*) définissant la bisimulation (avec temps abstrait) impliquent une correspondance naturelle entre les exécutions de $\mathcal{T}_\mathcal{A}$ et celles de $\mathcal{K}_\mathcal{A}$.

Exercice 10. On considère des automates temporisés à deux horloges $X = \{x, y\}$. Déterminer une partition de $\mathbb{R}^2_+$ en régions dans les cas suivants :

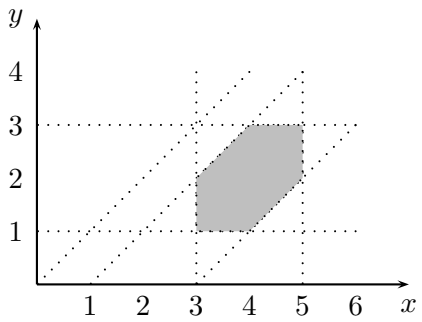
a) $x := 0$, $y := 0$, $x = 1$, $y = 1$
b) $x := 0$, $y := 0$, $y = x$, $x \geq 1$
c) $x := 0$, $y := 0$, $y = x$, $y \leq 2$
d) $x := 0$, $y := 0$, $y = 2x$, $y = 1$

Exercice 11. On considère un ensemble X contenant n horloges et on note $\mathcal{C}_1(X)$ l'ensemble des contraintes qui sont des **conjonctions** de contraintes atomiques $x \bowtie k$ ou $x - y \bowtie k$, avec $x, y \in X$, $k \in \mathbb{Z}$ et $\bowtie \in \{\leq, <, =, >, \geq\}$. Une *zone* est un sous-ensemble de $\mathbb{R}^X_+$ (identifié à $\mathbb{R}^n_+$) défini par une telle contrainte, on note $[[g]] = \{v \in \mathbb{R}^X_+ \mid v \models g$ la zone définie par la contrainte $g \in \mathcal{C}_1(X)$. Pour une zone Z, on définit le *futur* de Z, noté $\overrightarrow{Z}$, comme l'ensemble $\{v + d \mid v \in Z, d \in \mathbb{R}_+\}$.

1. Montrer que l'intersection de deux zones est une zone. La réunion de deux zones est-elle une zone ?

2. Soit $\mathcal{A}$ un automate temporisé et q un état de contrôle de $\mathcal{A}$. On considère une transition $q \xrightarrow{g,a,r} q'$ de $\mathcal{A}$, où g est une contrainte de $\mathcal{C}_1(X)$. On suppose que l'ensemble des valuations v atteintes dans q forme une zone Z. Expliquer pourquoi l'ensemble des valuations que l'on peut atteindre dans q' est $Z' = (\overrightarrow{Z} \cap [[Inv(q)]] \cap [[g]])[r \leftarrow 0]$. Cet ensemble Z' est-il une zone ?

3. Pour $X = \{x, y\}$, on considère la zone Z_0 dessinée ci-dessous.



(a) Donner une conjonction de contraintes qui la définit. Donner les contraintes définissant $\overrightarrow{Z_0}$.

(b) Construire un automate temporisé $\mathcal{A}$ qui génère cette zone, c'est-à-dire que $\mathcal{A}$ comporte un état q pour lequel l'ensemble des valuations v obtenues est exactement Z_0.