Automates temporisés

Formation sur les Systèmes à Événements Discrets (SED)

2e édition
Mars 2025
Nantes

Société d'Automatique,
de Génie Industriel & de Productique

Plan

1. Définition des automates temporisés
2. Evolutions des automates temporisés
3. Analyse
4. Mises en application théorique puis sur le logiciel UppAal

1 - Définition des automates temporisés

1.1 - Définition

Un automate temporisé est un automate fini classique étendu avec

- des horloges à valeurs réelles évoluant en continu,
- des transitions munies :
  - d'une garde sur les horloges qui indique quand la transition peut s'effectuer
  - d'une remise à zéro des horloges lors du tir de la transition
- des localités pouvant porter des invariants, c'est à dire, des contraintes qui limitent la durée de séjour dans celle-ci.

Intérêt : Ce modèle, introduit par Alur et Dill (années 1991), permet de modéliser le comportement temporel de systèmes temps-réel (ex. protocoles temps-réel, systèmes embarqués) en capturant des exigences du type "événement B doit survenir dans les t secondes suivant l'événement A".

1.1 - Définition

Sémantique :

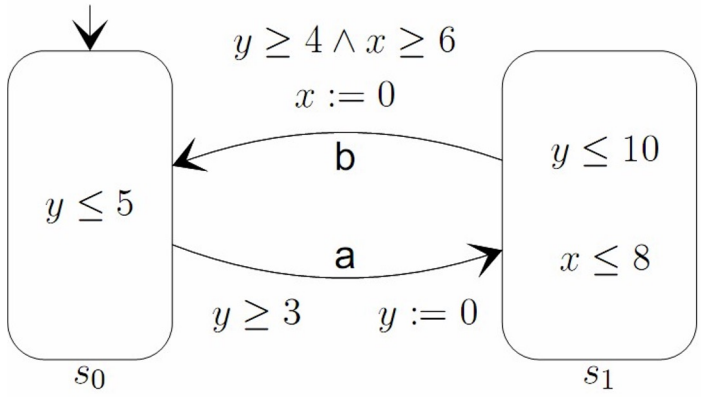
- un état d'un automate temporisé est une paire (localité, valuation)
  - localité est un état de contrôle
  - valuation assigne une valeur à chaque horloge.
- Le temps s'écoule dans une localité (les horloges augmentent à vitesse uniforme) jusqu'à ce qu'une transition soit déclenchable
  - Les horloges évoluent en même temps et mesurent le temps depuis leur dernière initialisation
  - mais elles peuvent être réinitialisées à 0 indépendamment
- une transition peut se déclencher instantanément, changeant de localité et réinitialisant certaines horloges.
- un invariant de la localité courante doit rester satisfait pendant l'écoulement du temps, sinon une transition sortante doit être prise avant qu'il ne soit violé

1.2 - Définition des automates temporisés

Première définition par Alur et Dill en 1991

Définition formelle des automates temporisés : $A = (\Sigma, L, L_0, H, I, T)$, avec

- Σ: ensemble fini d'événements possibles
- L : ensemble fini de localités
- $L_0 \in L$: localité initiale
- H : ensemble fini d'horloge à valeur dans $\mathbb{R}+$
- $I : L \rightarrow C(H)$ Invariant d'une localité
- $T \subset L \times \Sigma \times C(H) \times 2H \times L$: ensemble des transitions.



- Chaque $e =< l, a, \phi, \lambda, l' > \in T$ correspond une transition entre la localité l et la localité l', gardé par la contrainte ϕ , étiqueté par $a \in \Sigma$ et qui réinitialise les variables $\lambda \subset L$

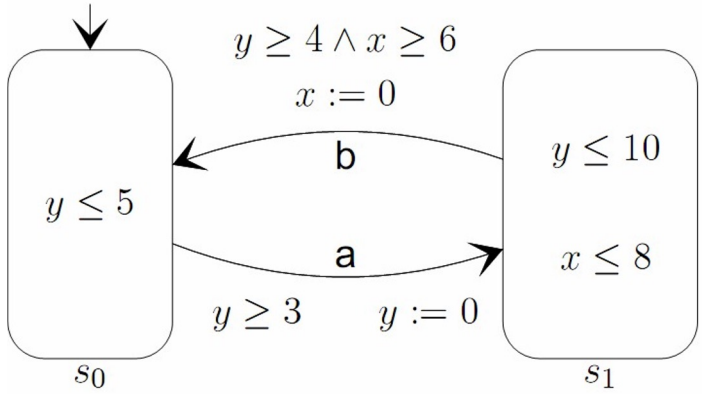
1.2 - Définition des automates temporisés

Exemple. Soit un automate A avec L=(s_0, s_1) et H=(x,y)

La transition s_0 → s_1 est étiquetée par

l'événement a, et conditionnée par la garde y ≥ 3.

Au franchissement, l'horloge y est remise à 0.



s_0 possède un invariant y ≤ 5 , c'est-à-dire qu'en s_0 l'horloge y ne peut pas être supérieure à 5

s_1 en possède deux, y ≤ 10 et x ≤ 8, c'est-à-dire qu'en s_1 l'horloge y ne peut pas être supérieure à 10 et l'horloge x ne peut pas être supérieure à 8

2 - Evolution des automates temporisés

2.1 - Evolution des automates temporisés

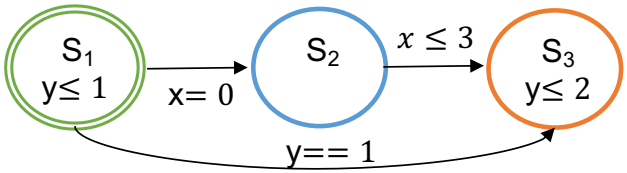
Un automate temporisé peut évoluer de la manière suivante :

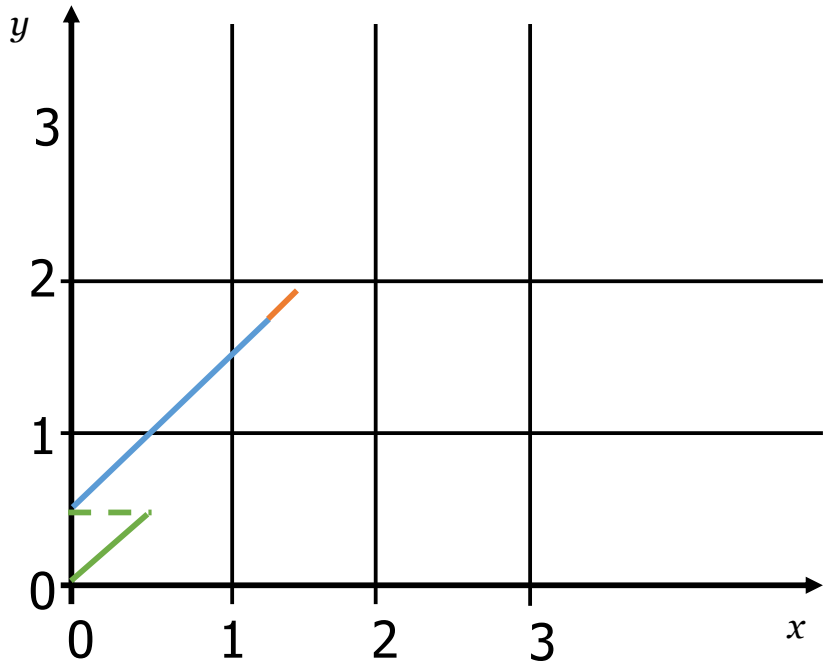
- Rester dans la localité courante et laisser le temps s'écouler
- Quitter la localité courante avant que l'invariant ne soit violé et franchir une transition
  - Si sa garde est vraie et que l'invariant de la localité de destination est satisfait
  - Suite à l'occurrence d'un événement
  - Et Remettre à zéro des horloges

Le franchissement d'une transition entre deux localités est considéré instantané

2.1 - Evolution des automates temporisés

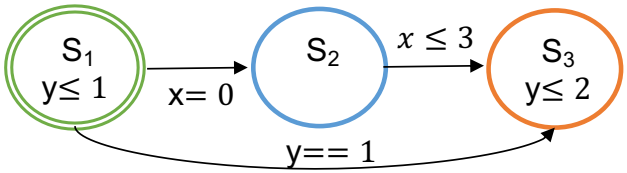
- *Exemple.* Evolutions d'un automate temporisé

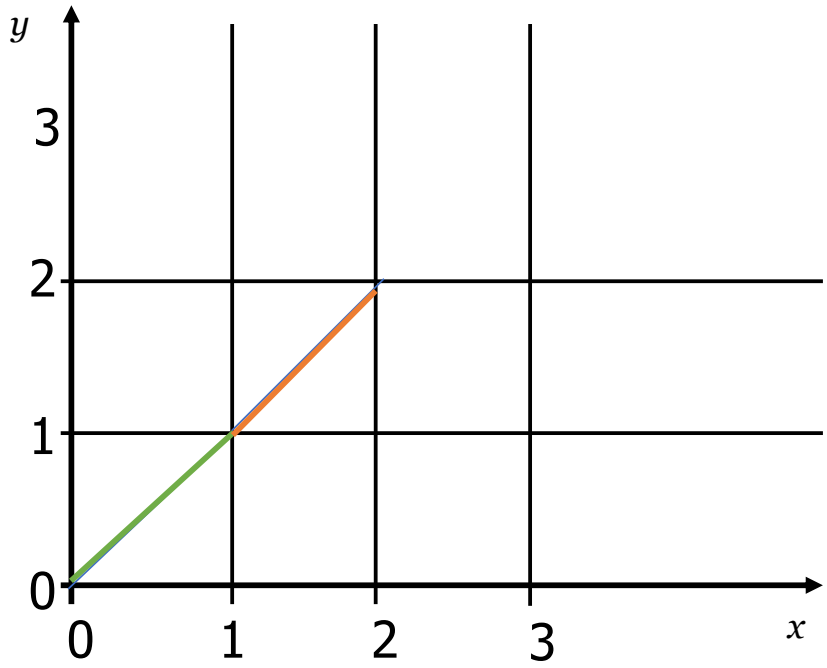




2.1 - Evolution des automates temporisés

- *Exemple.* Evolutions d'un automate temporisé





2.2 – Incohérences temporelles

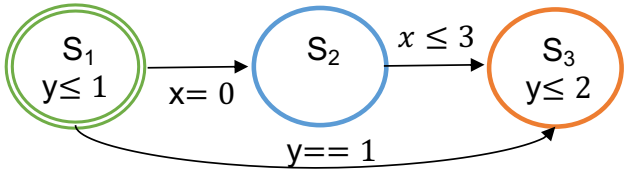
Définition : Une incohérence apparait lorsque le système atteint un état avec des valeurs des horloges telles qu'**aucune garde des transitions sortantes n'est ou ne sera jamais vérifiée**. **Conséquence** : Blocage permanent dans cet état.

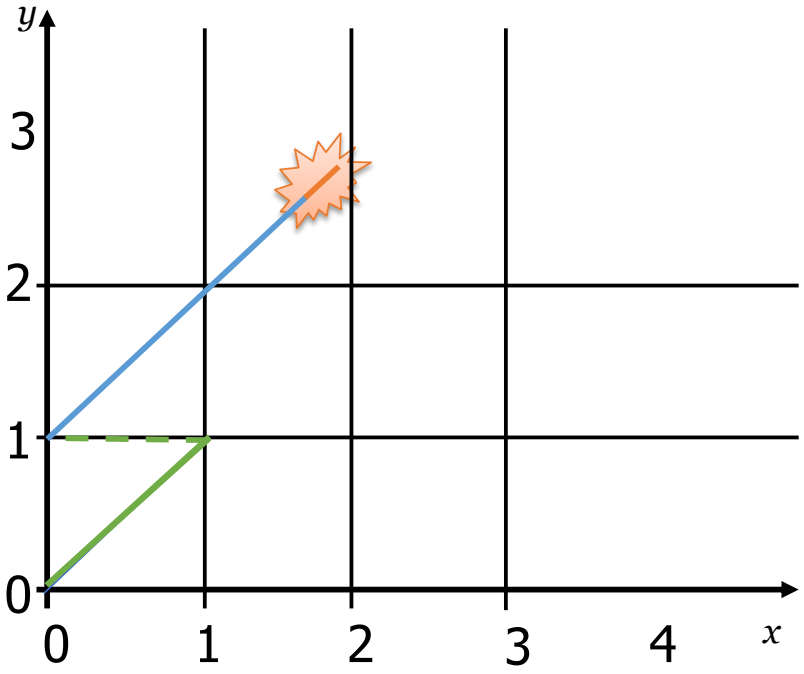
Il existe deux catégories d'incohérences temporelles :

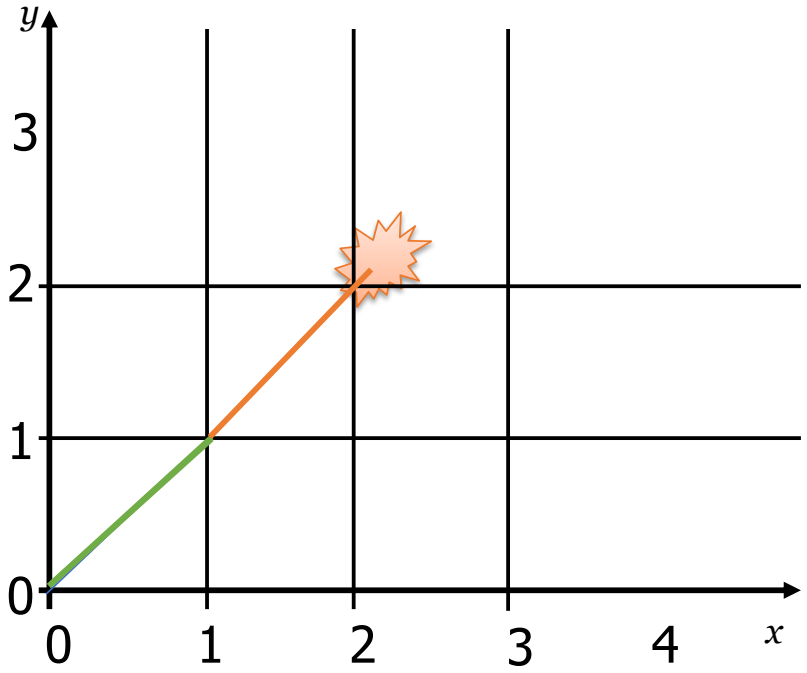
- Les **incohérences temporelles simples** apparaissent lorsqu'un automate ne peut pas franchir une transition en raison de ses contraintes temporelles (blocage local).
- Les **incohérences temporelles complexes** apparaissent lorsque les automates composés imposent des contraintes incompatibles sur un même événement commun (blocage global).

2.2 – Incohérences temporelles

- *Exemple.* Incohérences temporelles

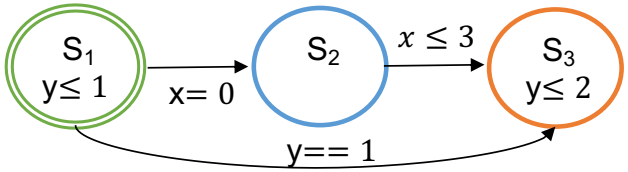


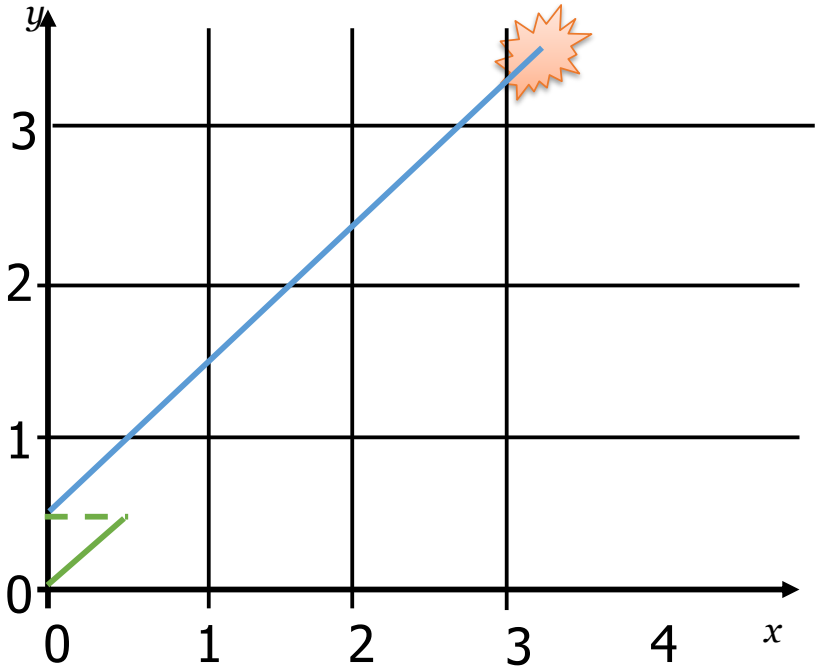




2.2 – Incohérences temporelles

- *Exemple.* Incohérences temporelles





3 - Analyse des automates temporisés

3.1 - Analyse des automates temporisés

L'analyse des automates temporisés est basée sur les mêmes propriétés que les automates à états finis :

- Un état q de A est accessible à partir d'un état p s'il existe une trajectoire dans A dont l'origine est p et dont q est l'extrémité. L'état q est dit accessible s'il est accessible à partir de l'état initial ;
- Un état p est co-accessible à un état q s'il existe une trajectoire dans A dont l'origine est p et dont q est l'extrémité. L'état p est dit co-accessible s'il est co-accessible à un état marqué.

Problème : Un automate temporisé génère un espace d'états continu (infini)
→ Impossible à vérifier directement

- Solution : Représentation symbolique : Graphe des régions (théorique) ou Graphe de zones (pratique)

3.2 - Définition d'un graphe des régions

L'automate des régions est l'abstraction finie obtenue en partitionnant l'ensemble des valuations d'horloges en classes d'équivalence appelées régions

Deux valuations d'horloges sont équivalentes (même région) si, intuitivement, l'automate temporisé aura le même comportement futur à partir de ces deux valuations

- Une région est définie par
  - des valeurs entières pour les horloges (bornées par les constantes maximales du modèle)
  - des informations sur les parts fractionnaires (ordre entre horloges atteignant une valeur entière, etc.)

3.2 - Définition d'un graphe des régions

- Partition de l'espace des horloges :
  - Régions définies par grilles (valeurs entières + fractionnaires)
  - Résultat : nombre élevé de régions, même pour des petits modèles

- Construction : Le graphe des régions se construit en prenant chaque localité de l'automate temporisé et en le dupliquant pour chacune des régions possibles des horloges.
  - Les nœuds du graphe des régions sont des couples (localité, région)
  - Les transitions entre ces nœuds reflètent les pas de temps ou les franchissements de transitions de l'automate temporisé :
    - (i) des arcs de temps relient (l, R) à (l, R') si en laissant le temps s'écouler depuis n'importe quelle valuation de la région R on atteint une valuation dans R' (sans changer de localité l),
    - (ii) des arcs de transition relient (l, R) à (l', R') s'il existe une transition depuis la localité l franchissable pour une valuation dans R et menant à la localité l' avec une nouvelle valuation dans R'

3.2 - Définition d'un graphe des régions

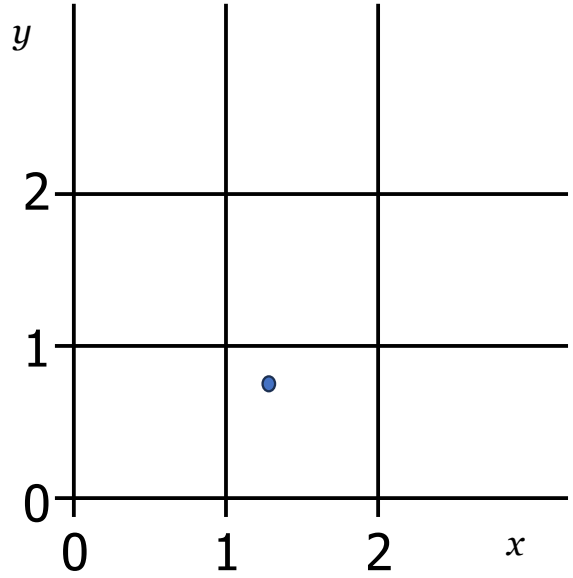
- Propriétés :
  - Nombre fini de régions
  - Sémantique exacte (bisimulation)
  - Décidabilité garantie

- Propriété de bisimulation entre l'automate temporisé et son graphe de régions : toute exécution de l'automate temporisé se reflète dans une suite de nœuds (q,R) dans le graphe des régions, et réciproquement toute exécution dans le graphe des régions correspond à une exécution concrète possible.

3.3 - Définition des régions

En fonction des valeurs des horloges, le franchissement d'une transition ou l'accès à une localité n'est pas toujours autorisé.

Pour revenir à un automate à états, il faut définir le graphe des régions

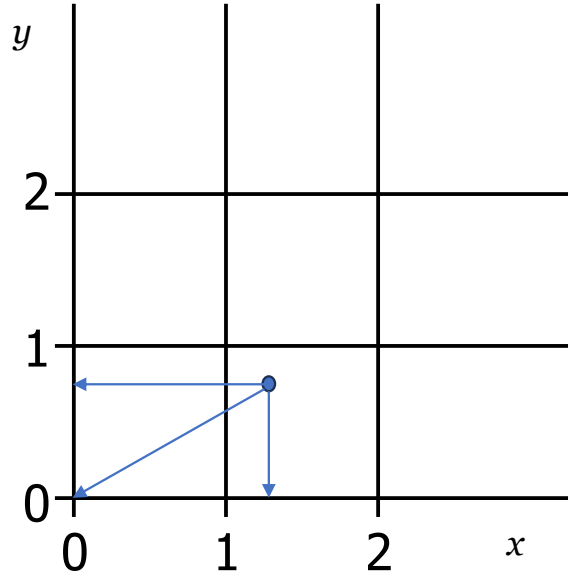


A un instant t, avec des valeurs d'horloge pour x et y, 2 évolutions possibles :

3.3 - Définition des régions

En fonction des valeurs des horloges, le franchissement d'une transition ou l'accès à une localité n'est pas toujours autorisé.

Pour revenir à un automate à états, il faut définir le graphe des régions



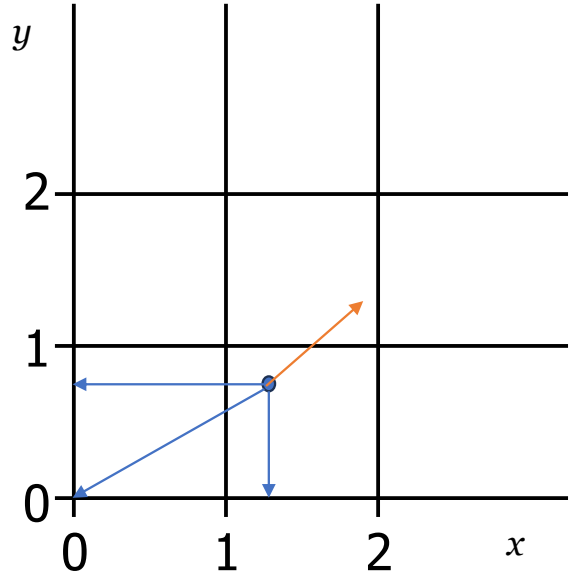
A un instant t, avec des valeurs d'horloge pour x et y, 2 évolutions possibles :

- Remise à zéro

3.3 - Définition des régions

En fonction des valeurs des horloges, le franchissement d'une transition ou l'accès à une localité n'est pas toujours autorisé.

Pour revenir à un automate à états, il faut définir le graphe des régions



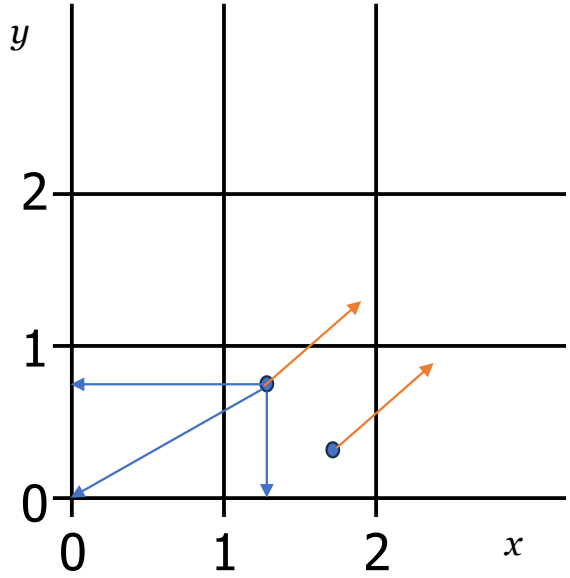
A un instant t, avec des valeurs d'horloge pour x et y, 2 évolutions possibles :

- Remise à zéro
- Ecoulement du temps

3.3 - Définition des régions

En fonction des valeurs des horloges, le franchissement d'une transition ou l'accès à une localité n'est pas toujours autorisé.

Pour revenir à un automate à états, il faut définir le graphe des régions



A un instant t, avec des valeurs d'horloge pour x et y,
2 évolutions possibles :

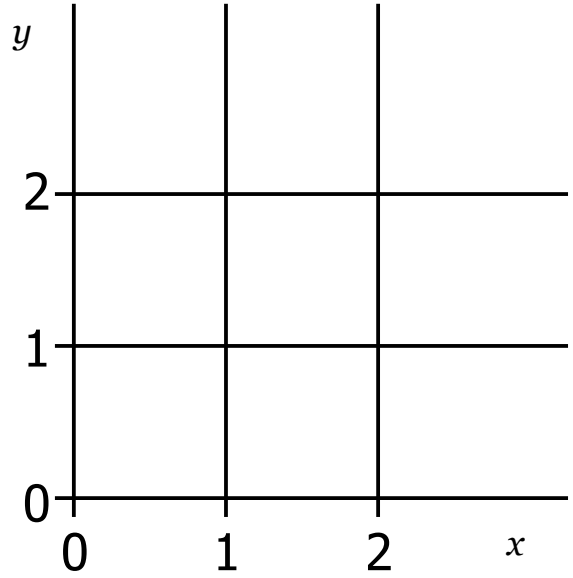
- Remise à zéro
- Ecoulement du temps

En fonction de l'instant t, les évolutions peuvent être différentes

3.3 - Définition des régions

En fonction des valeurs des horloges, le franchissement d'une transition ou l'accès à une localité n'est pas toujours autorisé.

Pour revenir à un automate à états, il faut définir le graphe des régions



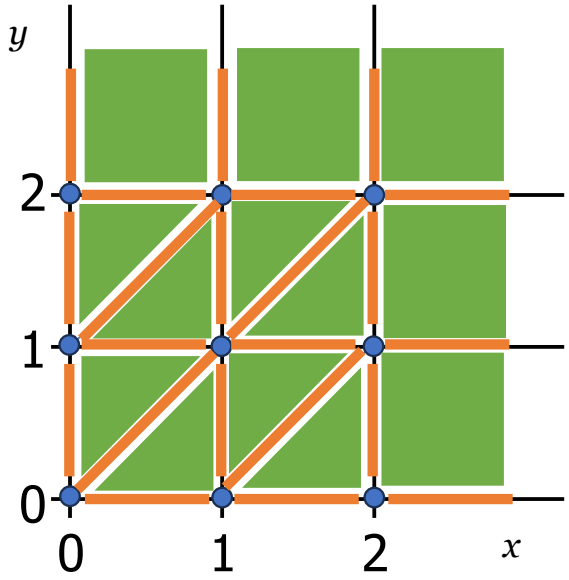
Le graphe des régions : discrétiser le domaine des horloges en un nombre fini de classes d'équivalence pour les évaluations des horloges, **à partir des contraintes d'horloges (**constantes entières)

3.3 - Définition des régions

- Le graphe des régions est une structure théorique fixe, déterminée uniquement par :
  - le nombre d'horloges
  - les constantes maximales des gardes/invariants
  - Il ne dépend pas des valeurs initiales concrètes des horloges.
  - Mais certaines régions ne seront jamais atteintes en pratique

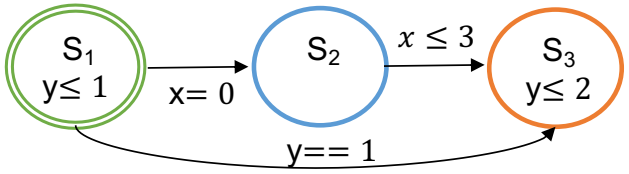
Objectif : Trouver les évolutions d'horloges
Les régions d'horloges peuvent être présentées par :

- Des points ● par exemple $x=1 \wedge y=1$
- Des segments ／ par exemple $(0<x<1) \wedge y=1$
- Des régions ouvertes ◢ par exemple $(0<x<1) \wedge (0<y<1)$

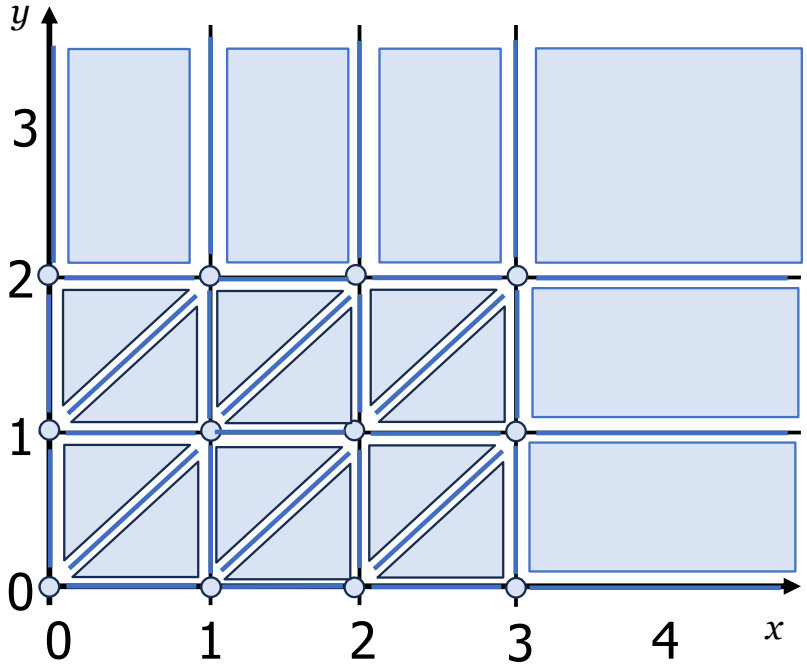


3.3 - Définition des régions

- *Exemple.* Incohérences temporelles



- Automate temporisé à 2 horloges (x et y)
- Contraintes temporelles liées aux gardes ou aux invariants :
  - $x \leq 3$
  - $y \leq 2$
- Nombre de régions : 60 régions
  - 12 points
  - 12 triangles
  - 17 segments + 6 segments diagonaux
  - 7 lignes infinies + 6 rectangles infinis



3.4 - Définition du graphe de régions accessibles

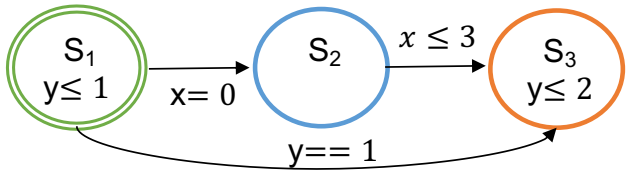
Une région R est dite **accessible** si et seulement s'il existe une exécution valide (écoulement du temps + transitions discrètes) depuis la localité initiale (l_0,u_0) telle que :

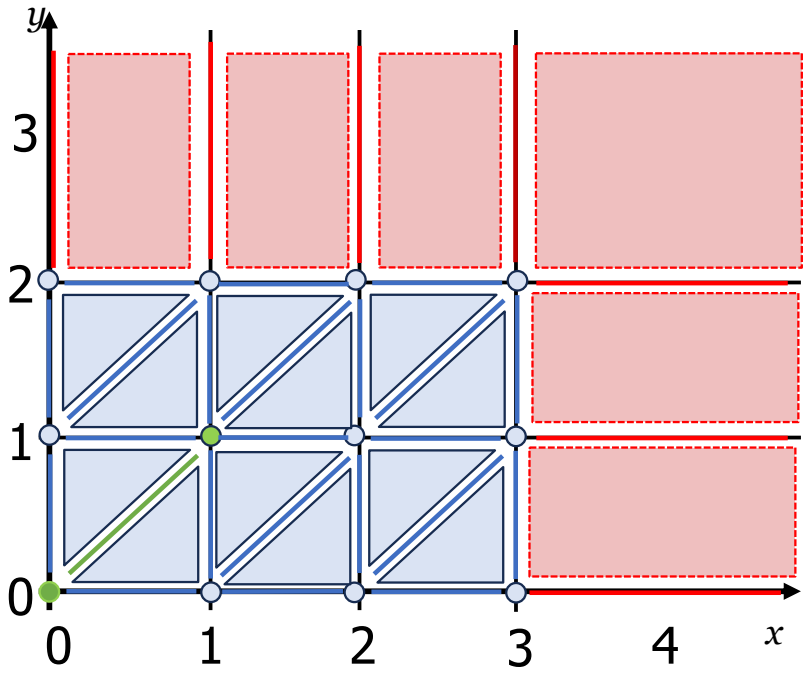
$$(l_0, u_0) \xrightarrow{\text{temps/transitions}}{}^{*} (l, u) \quad \text{avec} \quad u \in R$$

- (l_0, u_0) : état initial (localité initiale + valeurs initiales des horloges)
- (l, u) : état atteint par une séquence d'écoulements temporels et transitions
- R : région contenant la valuation u

3.4 - Définition du graphe de régions accessibles

- *Exemple.* Régions inaccessibles

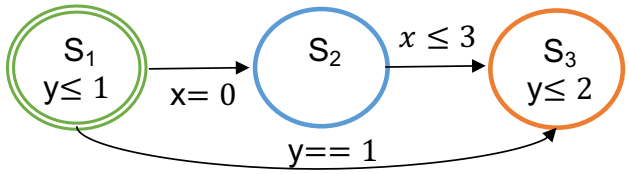




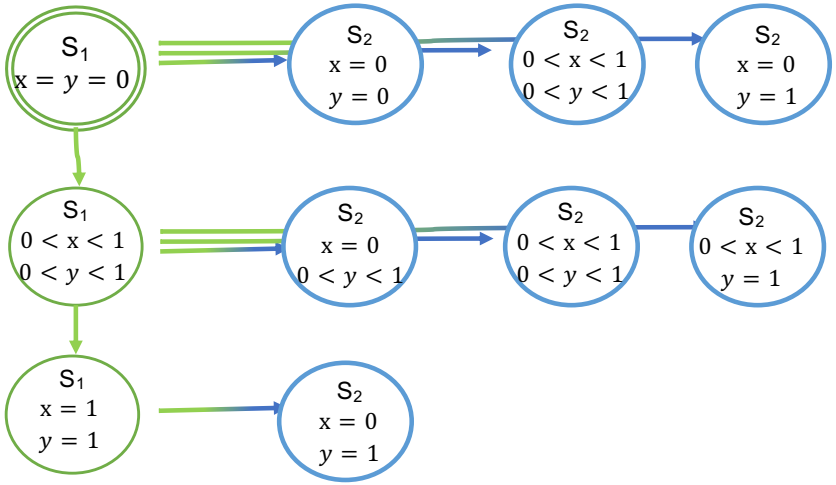
  - Contraintes temporelles liées aux gardes ou aux invariants :
    - $x \leq 3$
    - $y \leq 2$
  - Les régions en rouge ne sont pas accessibles

3.4 - Définition du graphe de régions accessibles

- *Exemple. Extrait du graphe de régions accessibles*



- Transformation de l'évolution S1, vers S2



3.5 - Graphe des zones

- Représentation **symbolique** des ensembles de valuations d'horloges atteignables. Une *zone* est un ensemble de valuations défini par une conjonction de contraintes linéaires sur les horloges
- Géométriquement, une zone correspond à une région convexe de l'espace des horloges, c'est-à-dire une union de plusieurs régions élémentaires du graphe des régions :
  - Une **région** correspond à une classe d'équivalence très fine des valuations temporelles.
  - Une **zone** regroupe plusieurs régions adjacentes qui n'ont pas besoin d'être distinguées pour les **analyses futures,** car elles partagent le même ensemble de transitions possibles et contraintes temporelles à venir.

- **Avantages** : Le graphe des zones est une abstraction du graphe des régions, qui évite de générer explicitement toutes les régions possibles. En pratique, cela réduit le nombre d'états explorés.

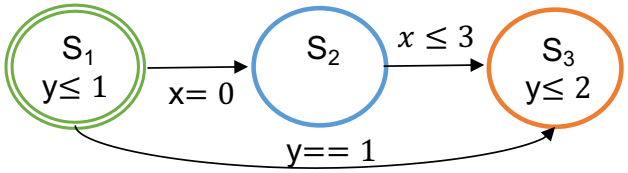
3.5 - Graphe des zones

Construction du graphe de zones

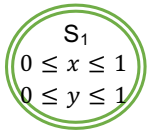
- Chaque nœud est un couple (localité, zone) où localité est une localité de l'automate temporisé et zone une contrainte représentant *toutes* les valuations possibles des horloges dans cet état.
- L'état initial aura pour zone l'ensemble des valuations initiales (souvent toutes horloges à 0) et la localité initiale.
- Les nœuds successeurs en appliquant symboliquement les mêmes pas que dans le graphe des régions :
  - (i) **évolution du temps** : on fait évoluer la zone actuelle en ajoutant le passage du temps
  - (ii) **franchissement d'une transition** : on intersecte la zone courante avec la garde de transition (pour conserver uniquement les valuations habilitant la transition), puis on applique le reset des horloges de la transition (on remet ces horloges à 0 dans la zone)
- Une nouvelle **zone** sera définie si les **analyses futures sont différentes de la zone courante.**

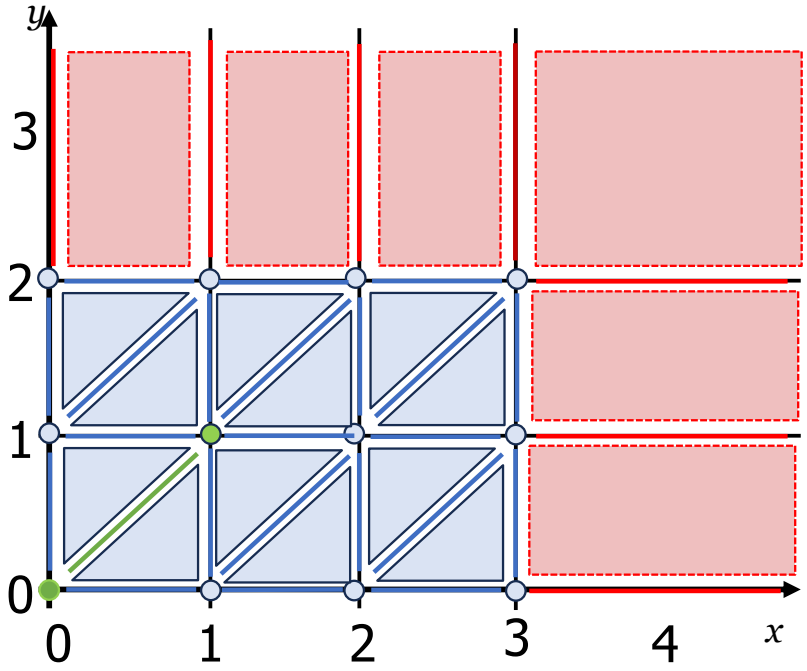
3.5 - Graphe des zones

- *Exemple. Construction du graphe de zones*



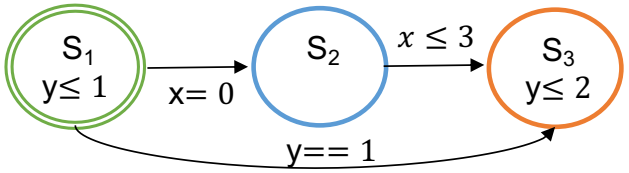
  - Zone verte : être en S1 et attendre, évolutions futures :
    - Passage à la zone bleue entre 0 et 1
    - Passage à la zone orange à y=1
    - Pour y>1 invariant violé

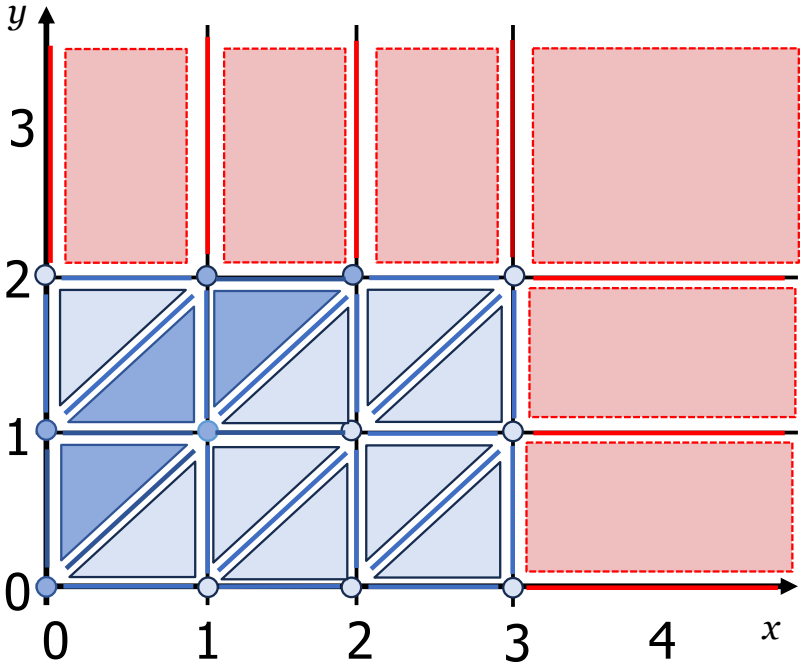




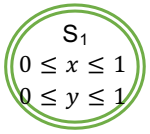
3.5 - Graphe des zones

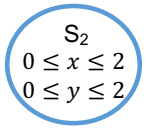
- *Exemple. Construction du graphe de zones*

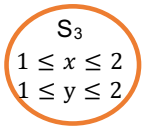




  - Zone bleue : être en S2 et attendre
    - Passage à la zone orange avant que y>2
    - Passage vers une zone bleue bloquante car y>2 viole l'invariant de S3

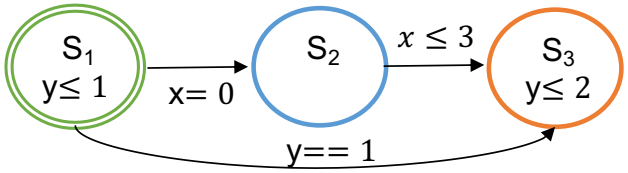


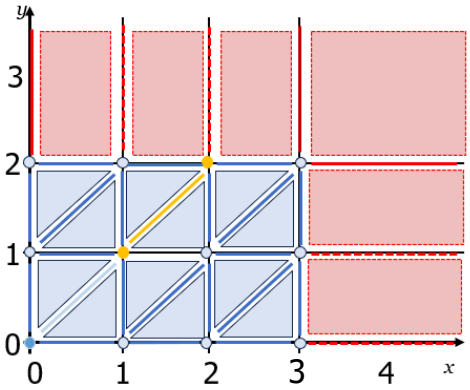




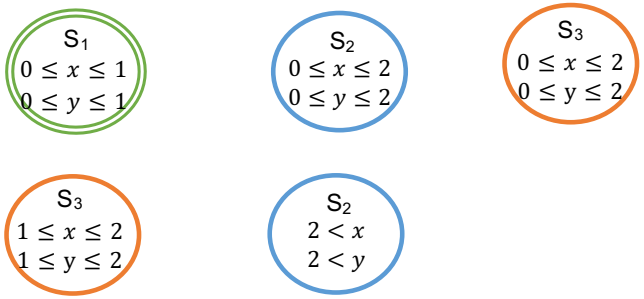
3.5 - Graphe des zones

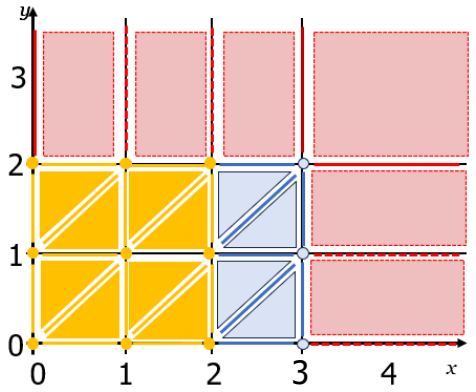
- *Exemple. Construction du graphe de zones*



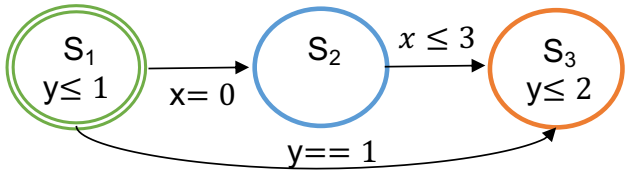


- Zone orange : être en S3 et attendre

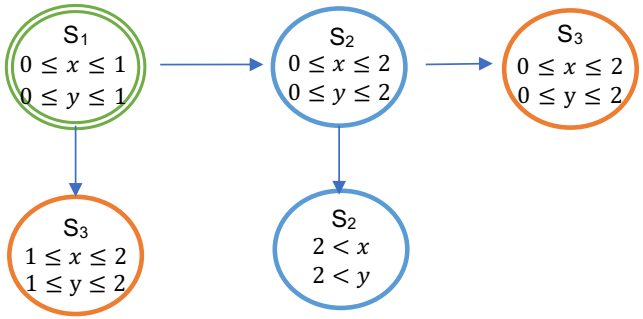




3.5 - Graphe des zones



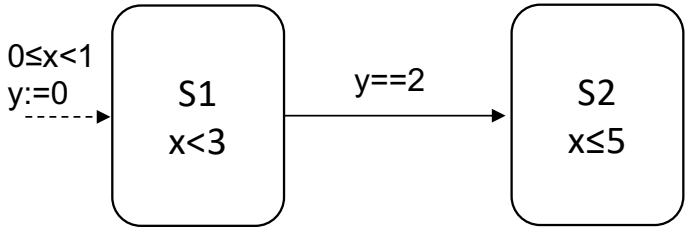
- *Exemple. Construction du graphe de zones*



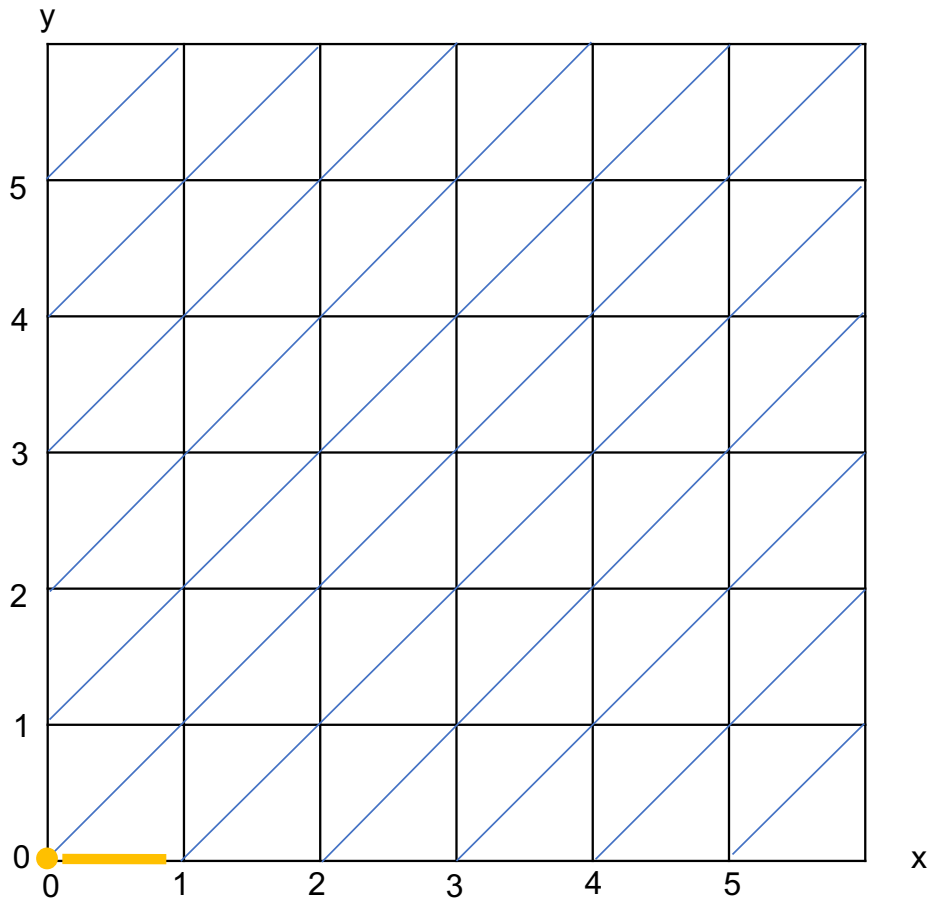
4 – Mises en application

4.1 – Exercice transformations vers graphe de zones

Exemple. Soit un extrait d'automate A avec X=(s_1, s_2) et H=(x,y)

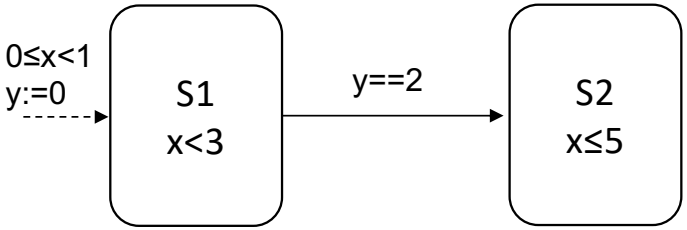


Contrainte temporelle : 0 ≤x<1 ∧ y=0 => Zone Jaune en S1

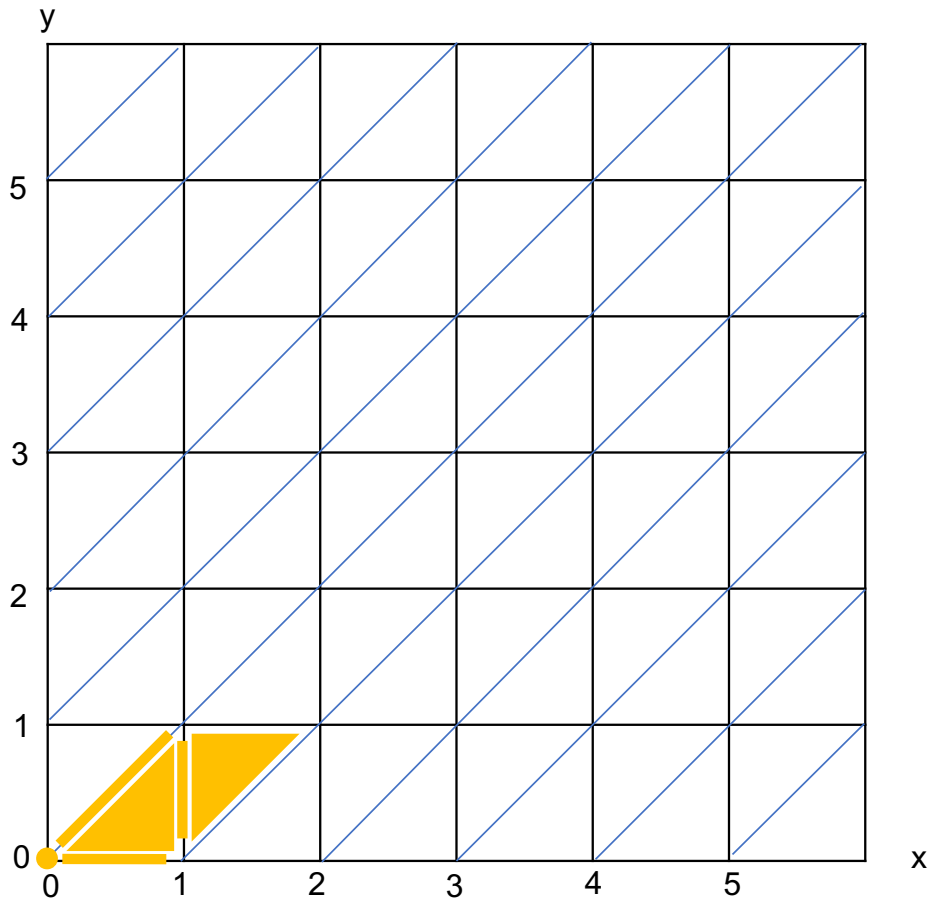


4.1 – Exercice transformations vers graphe de zones

Exemple. Soit un extrait d'automate A avec X=(s_1, s_2) et H=(x,y)

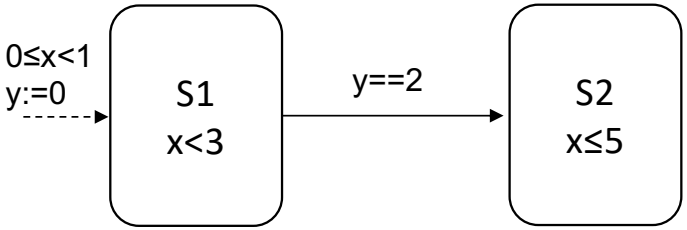


Contrainte temporelle : 0 ≤x<1 ∧ y=0 => Zone Jaune en S1
0≤x<2 ∧ 0≤y<1 => Zone jaune en S1

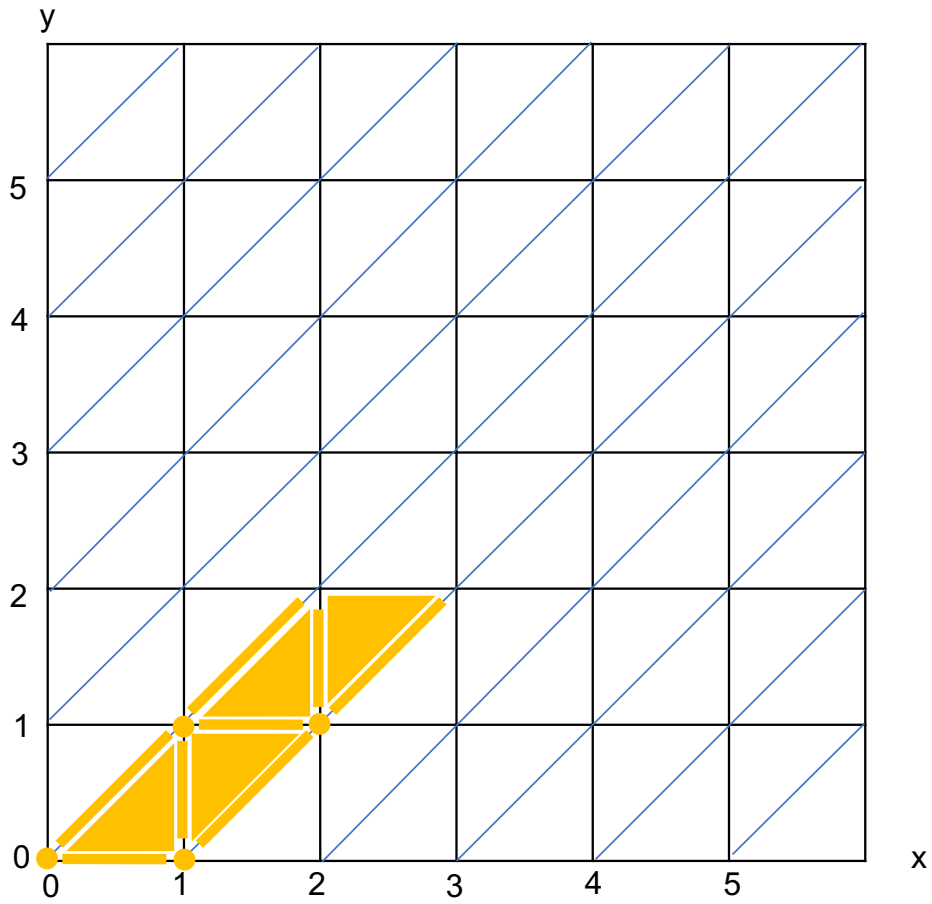


4.1 – Exercice transformations vers graphe de zones

Exemple. Soit un extrait d'automate A avec X=(s_1, s_2) et H=(x,y)

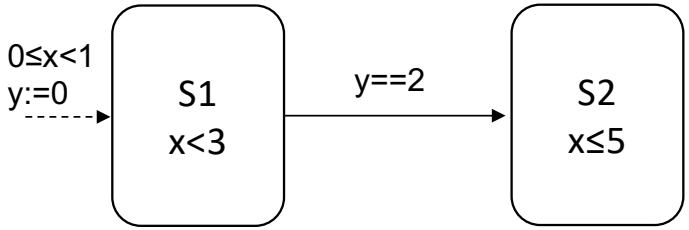


Contrainte temporelle : 0 ≤x<1 ∧ y=0 => Zone Jaune en S1
0≤x<2 ∧ 0≤y<1 => Zone jaune en S1
0≤x<3 ∧ 0≤y<2 => Zone jaune en S1



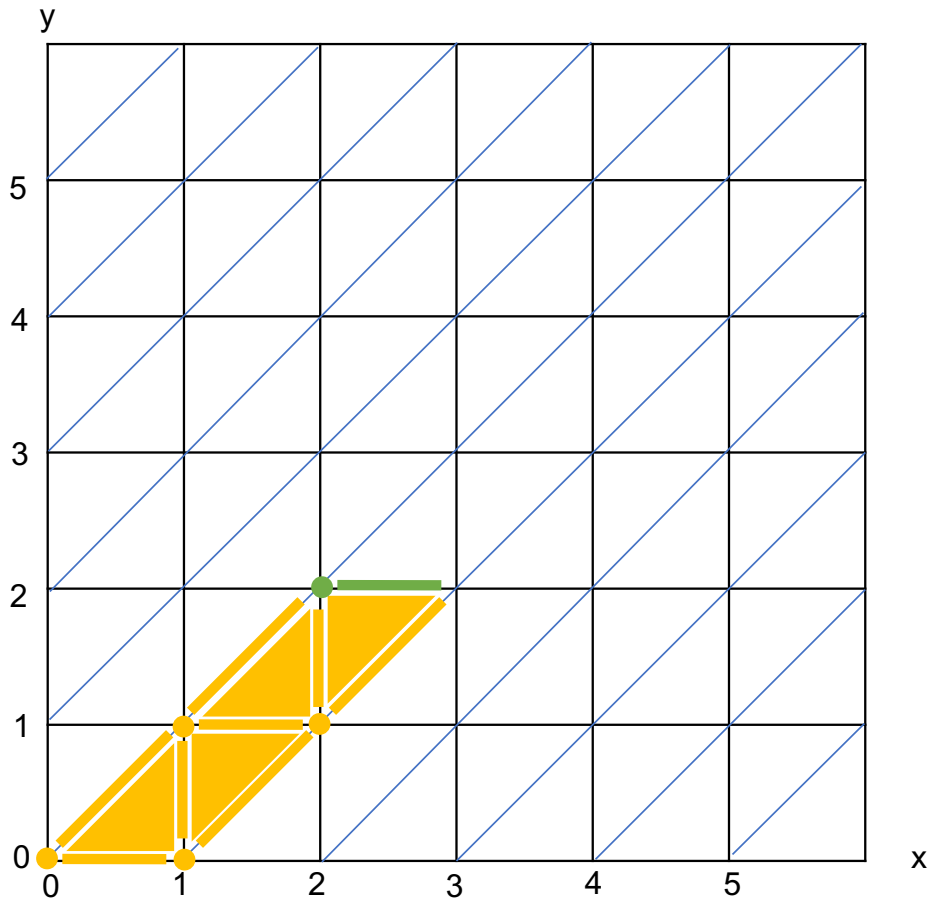
4.1 – Exercice transformations vers graphe de zones

Exemple. Soit un extrait d'automate A avec X=(s_1, s_2) et H=(x,y)



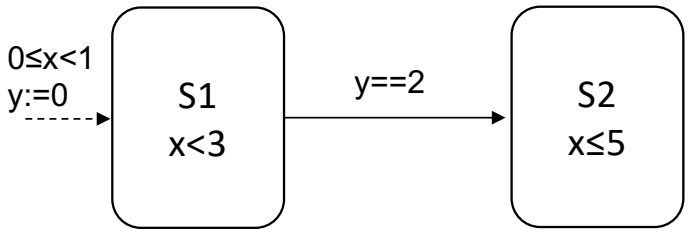
Contrainte temporelle : 0 ≤x<1 ∧ y=0 => Zone Jaune en S1
0≤x<2 ∧ 0≤y<1 => Zone jaune en S1
0≤x<3 ∧ 0≤y<2 => Zone jaune en S1

2≤x<3 ∧ y=2 => Zone Verte : transition franchissable



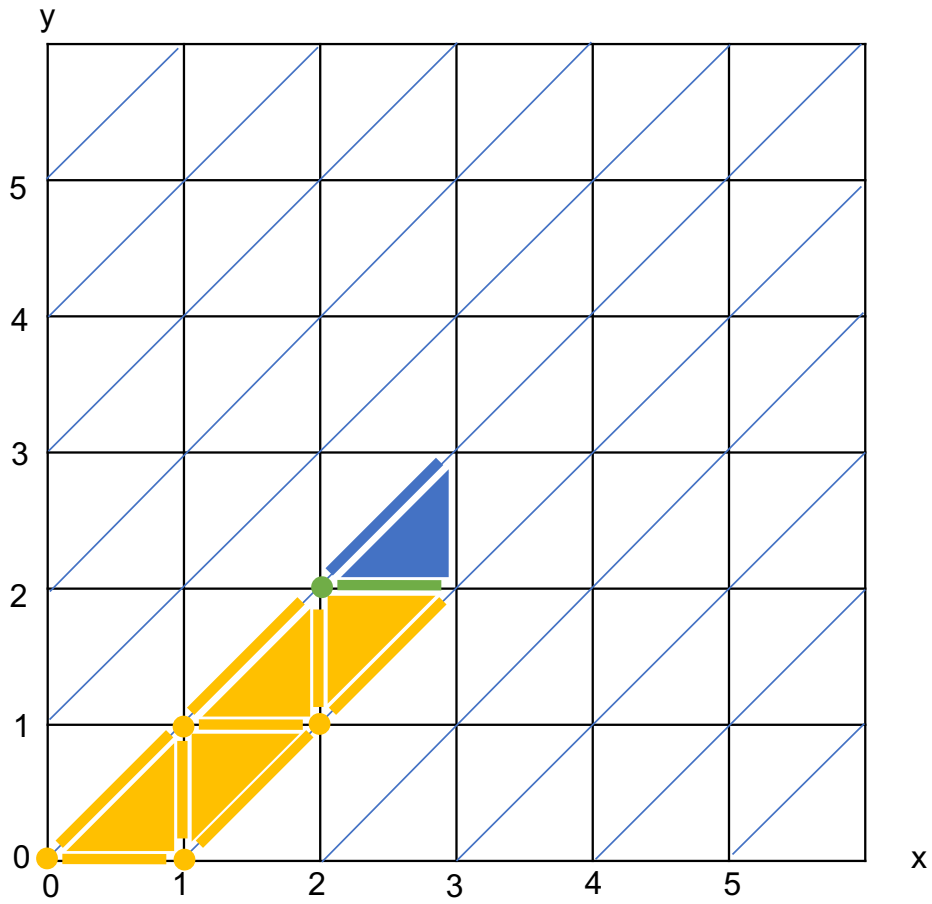
4.1 – Exercice transformations vers graphe de zones

Exemple. Soit un extrait d'automate A avec X=(s_1, s_2) et H=(x,y)



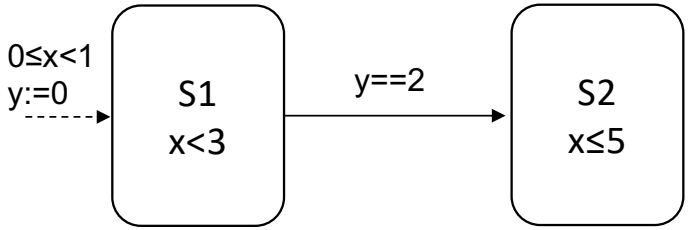
Contrainte temporelle : 0 ≤x<1 ∧ y=0 => Zone Jaune en S1
0≤x<2 ∧ 0≤y<1 => Zone jaune en S1
0≤x<3 ∧ 0≤y<2 => Zone jaune en S1

2≤x<3 ∧ y=2 => Zone Verte : transition franchissable
1) 2<x<3 ∧ 2<y<3 => Zone Bleue : transition non franchie => deadlock



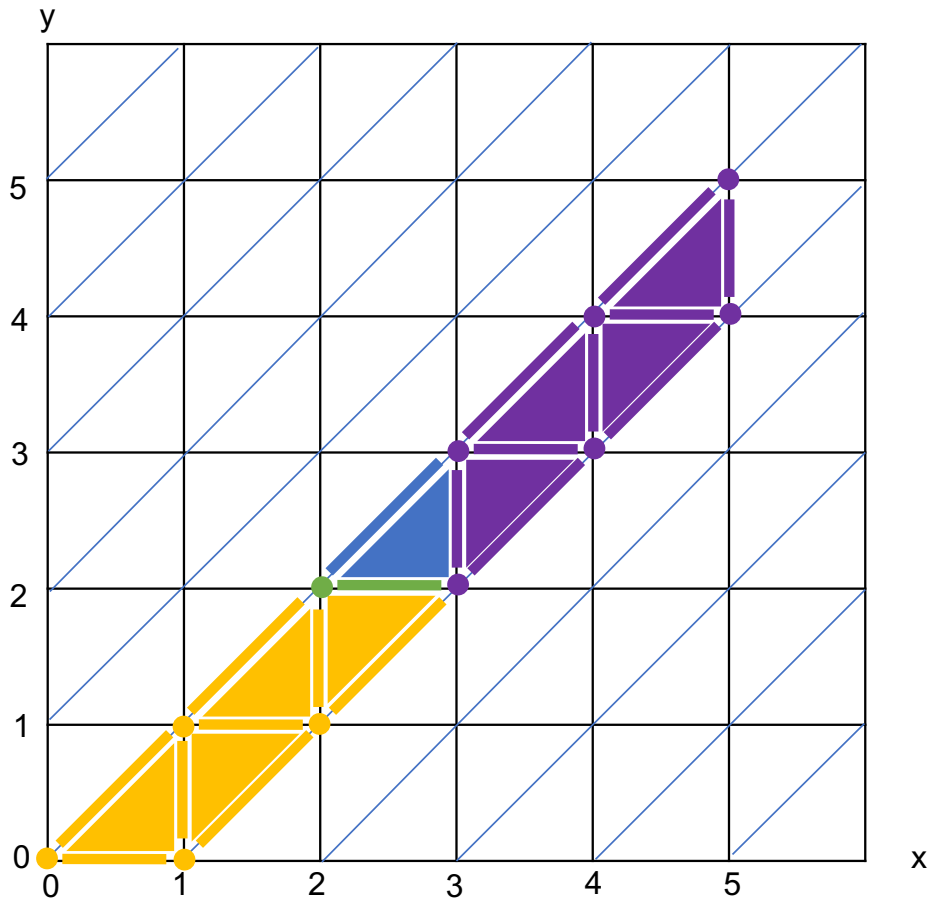
4.1 – Exercice transformations vers graphe de zones

Exemple. Soit un extrait d'automate A avec X=(s_1, s_2) et H=(x,y)



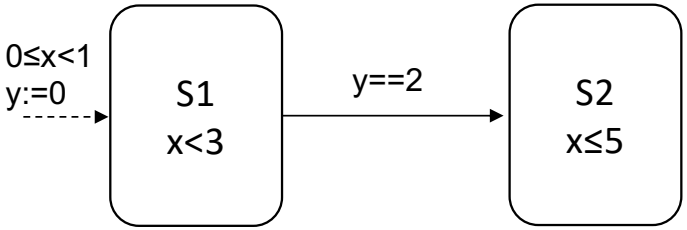
Contrainte temporelle : 0 ≤x<1 ∧ y=0 => Zone Jaune en S1
0≤x<2 ∧ 0≤y<1 => Zone jaune en S1
0≤x<3 ∧ 0≤y<2 => Zone jaune en S1

2≤x<3 ∧ y=2 => Zone Verte : transition franchissable
1) 2<x<3 ∧ 2<y<3 => Zone Bleue : transition non franchie => deadlock
2) 2≤x≤5 ∧ 2≤y≤5 => Zone Bleue + Violette (+Verte) : transition franchie => en S2



4.1 – Exercice transformations vers graphe de zones

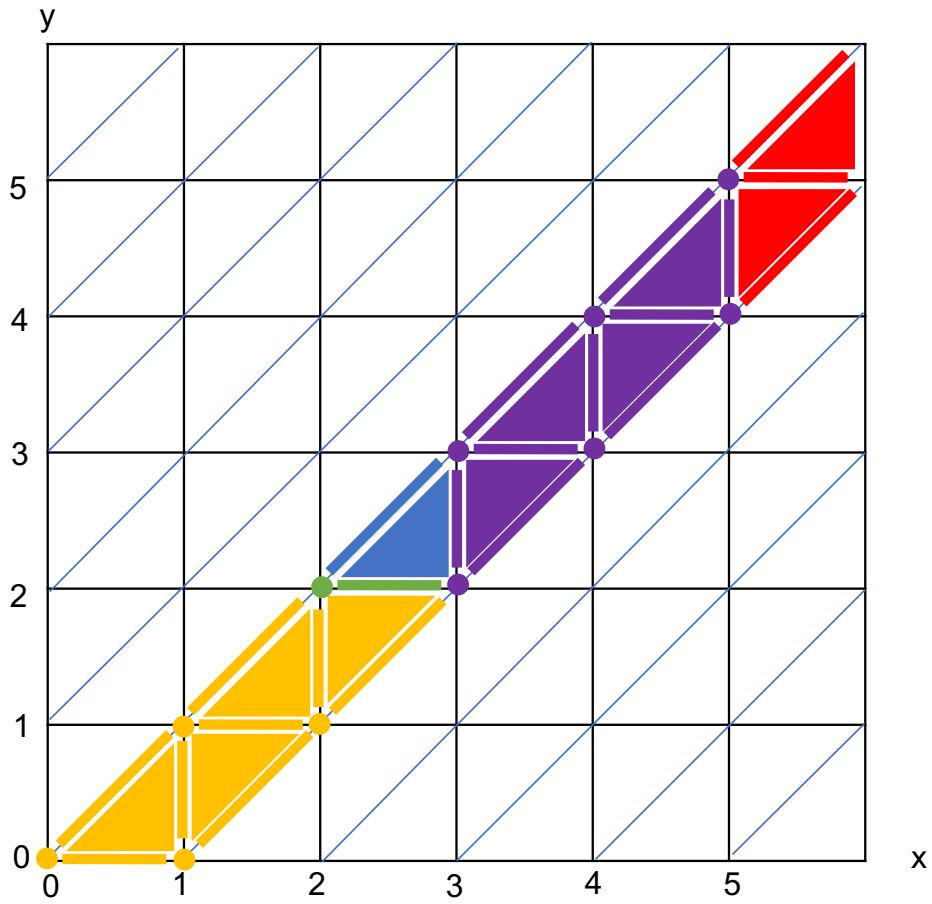
Exemple. Soit un extrait d'automate A avec X=(s_1, s_2) et H=(x,y)



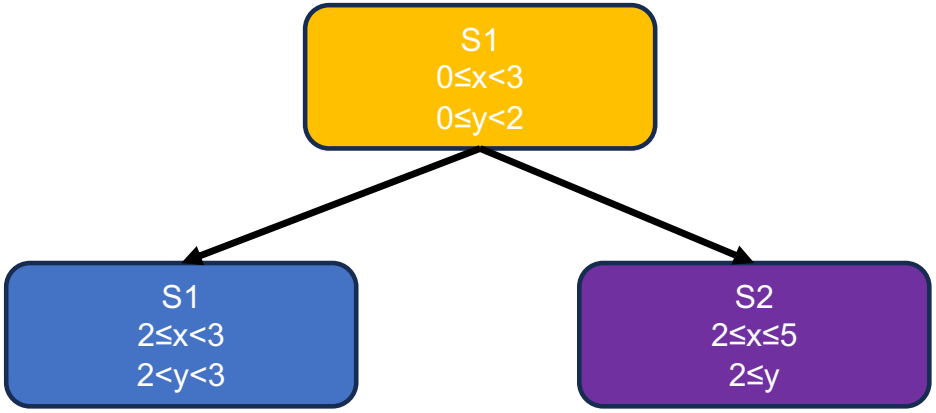
Contrainte temporelle : 0 ≤x<1 ∧ y=0 => Zone Jaune en S1
0≤x<2 ∧ 0≤y<1 => Zone jaune en S1
0≤x<3 ∧ 0≤y<2 => Zone jaune en S1

2≤x<3 ∧ y=2 => Zone Verte : transition franchissable
1) 2<x<3 ∧ 2<y<3 => Zone Bleue : transition non franchie => deadlock
2) 2≤x≤5 ∧ 2≤y≤5 => Zone Bleue + Violette (+Verte) : transition franchie => en S2

x>5 : incohérence temporelle => Zone Rouge



Automate à états équivalent



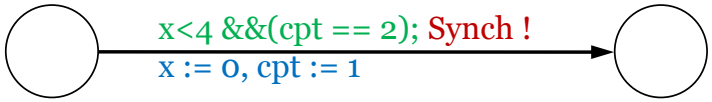
Exercice UppAal

4.2 Outil utilisé : UppAal

Caractéristiques :

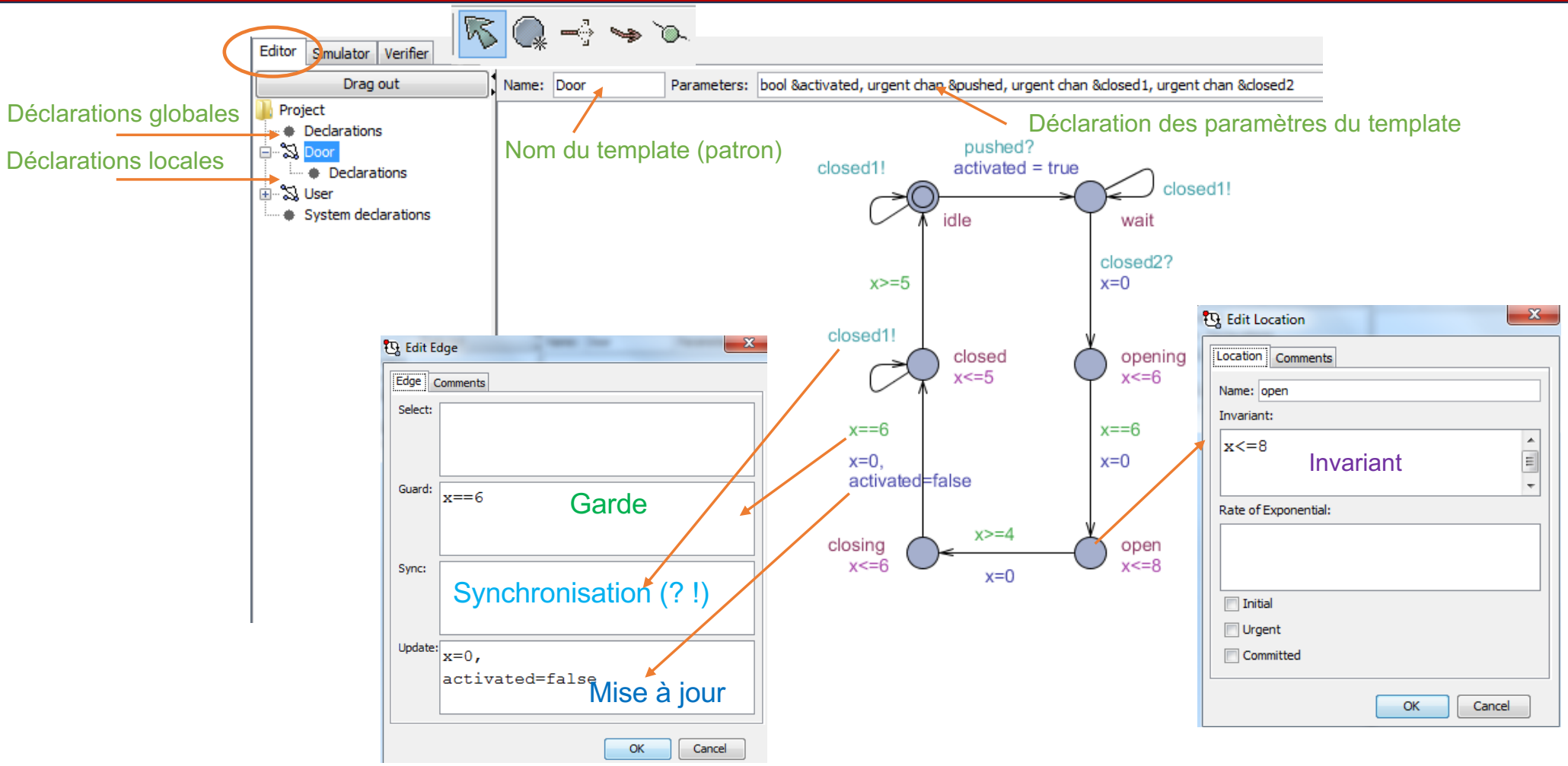
- Outil pour modéliser, simuler et vérifier des systèmes en temps réel
- Modélise les systèmes comme une collection d'automates :
  - non déterministes
  - temporisés par des horloges à valeurs réelles
  - Mise à jour d'autres variables que des horloges
  - communiquant par des canaux de messages ou des variables partagées

- Automates finis + variables (horloges + entiers) + synchronisation

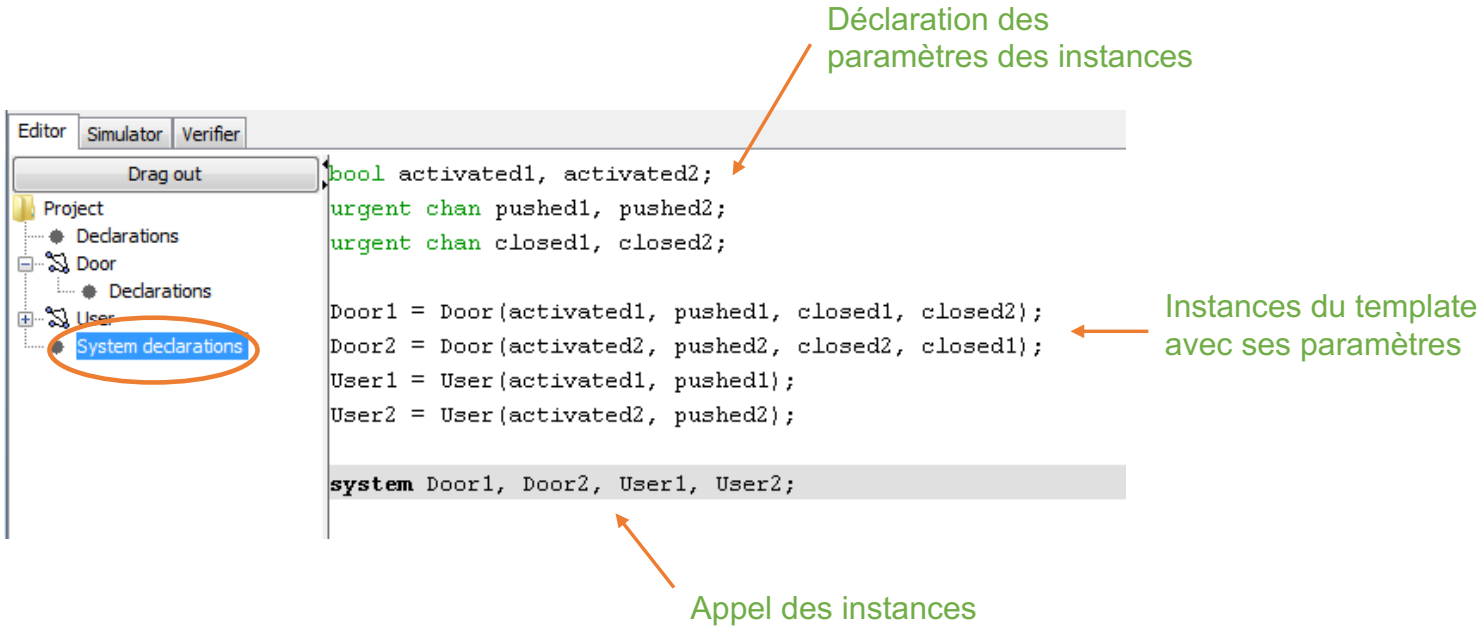


- A chaque transition peuvent être associés :
  - des canaux (quand on veut synchroniser des automates)
  - des gardes (= des conditions sur les valeurs des horloges ou des variables)
  - des actions de remise à jour des horloges (attention, elles ne peuvent qu'être remises à zéro) ou des variables

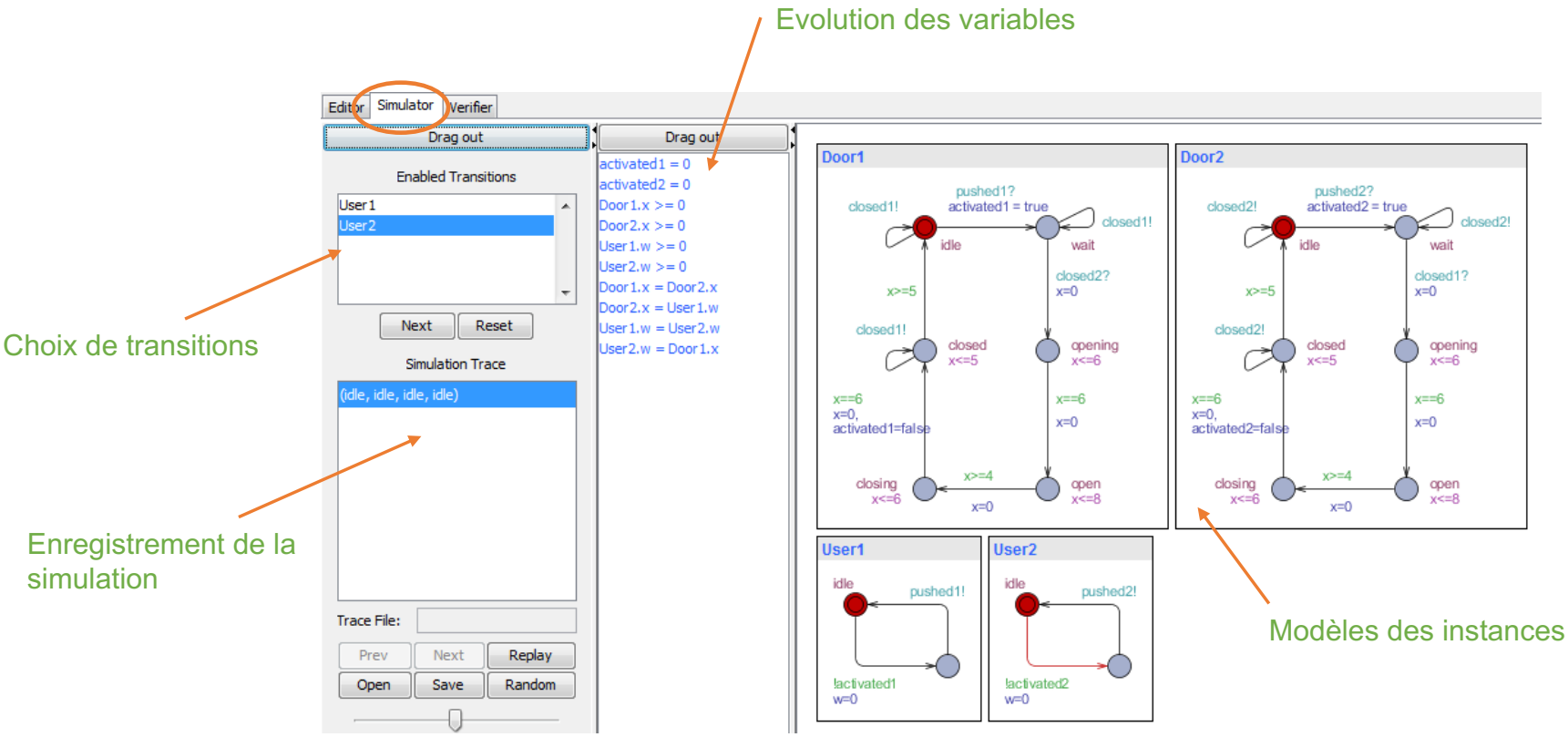
4.2 Outil utilisé : UppAal



4.2 Outil utilisé : UppAal

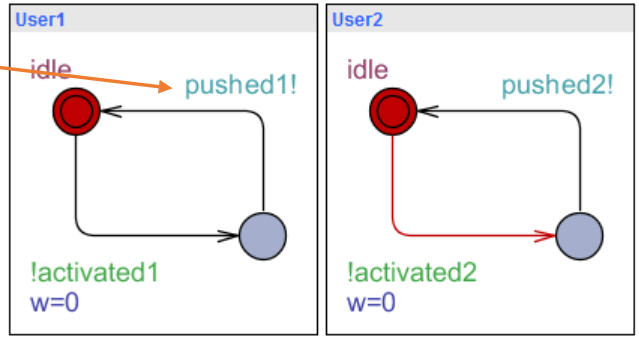


4.2 Outil utilisé : UppAal

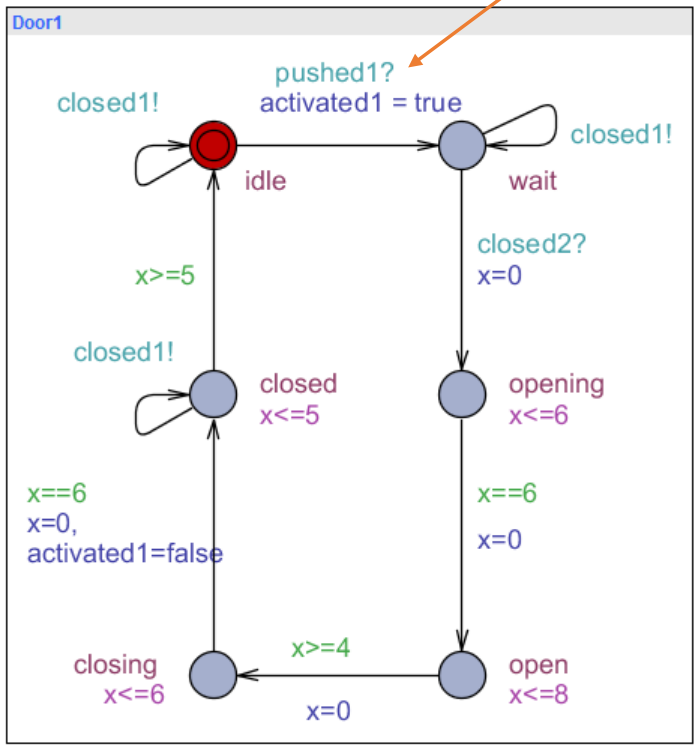


4.2 Outil utilisé : UppAal

Envoi d'un message !

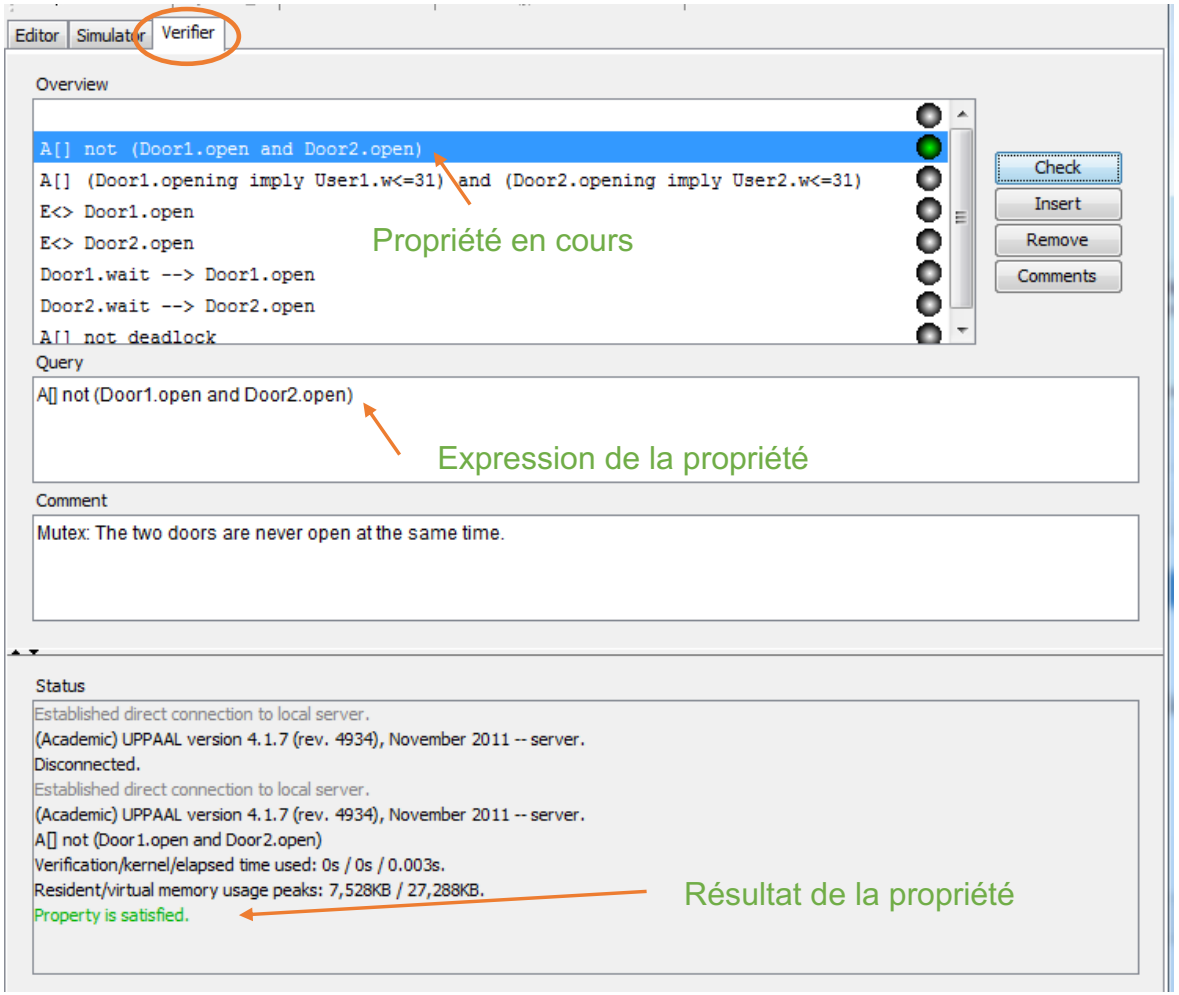


Réception d'un message ?



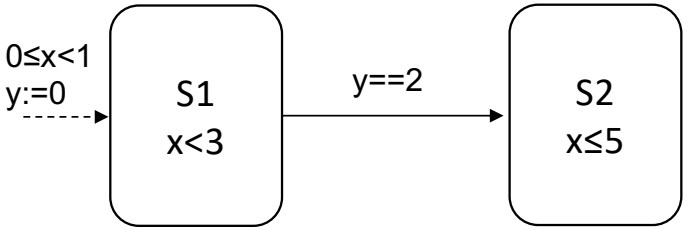
4.2 Outil utilisé : UppAal

Module
Model Checking



4.2 – Vérification du graphe de zones

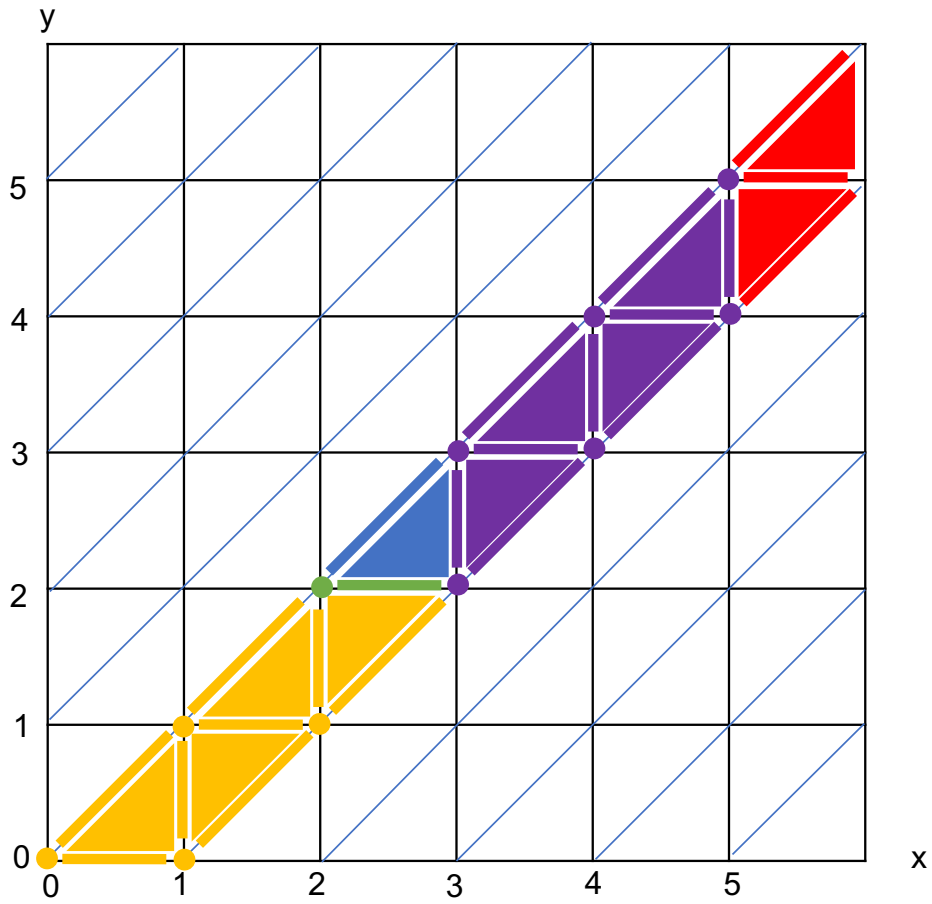
Exemple. Soit un extrait d'automate A avec X=(s_1, s_2) et H=(x,y)



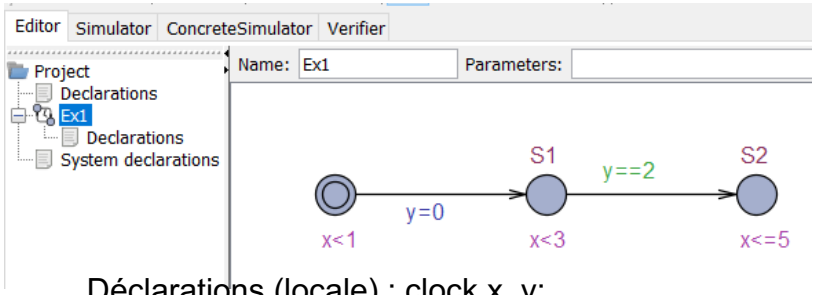
Contrainte temporelle : 0 ≤x<1 ∧ y=0 => Zone Jaune en S1
0≤x<2 ∧ 0≤y<1 => Zone jaune en S1
0≤x<3 ∧ 0≤y<2 => Zone jaune en S1

2≤x<3 ∧ y=2 => Zone Verte : transition franchissable
1) 2<x<3 ∧ 2<y<3 => Zone Bleue : transition non franchie => deadlock
2) 2≤x≤5 ∧ 2≤y≤5 => Zone Bleue + Violette (+Verte) : transition franchie => en S2

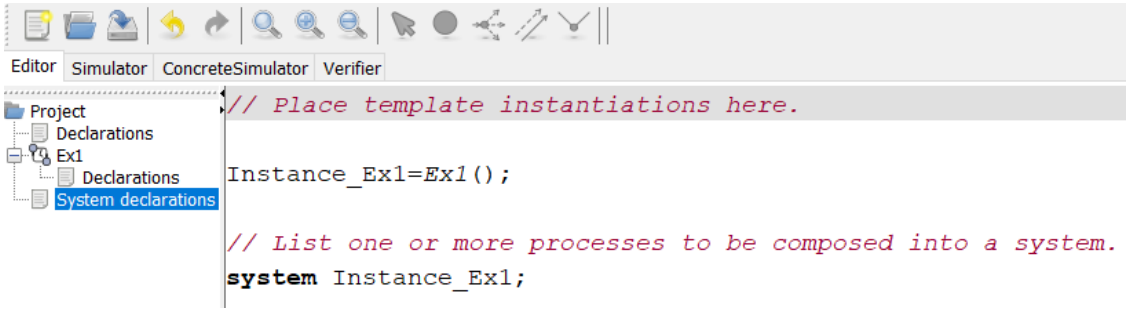
x>5 : incohérence temporelle => Zone Rouge



4.2 – Vérification des évolutions via le graphe de zones



Déclarations (locale) : clock x, y;



4.3 Exemple d'un passage à niveau

Soit un système de passage à niveau, afin de vérifier la sécurité de tronçon de route, nous souhaitons modéliser :

CI : Barrière ouverte et train éloigné

- Le train
  - Signale son approche au moins 1 secondes avant la section gardée
  - Parcours la distance entre l'approche et la zone gardée dans un intervalle compris entre 4 et 5 secondes pour un TER, (et entre 1 et 5 secondes pour un TGV)
  - Signale son départ de la zone gardée entre 2 secondes et 4 secondes
- Système de pilotage de la barrière
  - Quand le train signal son approche, la demande de descente est envoyée à la barrière instantanément
  - Sa descente doit avoir lieu entre 1 et 2 secondes
  - Quand le train sort, la demande de levée est envoyée instantanément
  - Sa montée doit avoir lieu entre 1 et 2 secondes
  - Durant sa montée, si un nouveau train approche, celle-ci doit alors redescendre

4.3 Exemple d'un passage à niveau

Présentation exemple

- Modéliser le comportement du train TER/TGV
- Modéliser le comportement de la barrière avec 1 voie

- Simuler les évolutions sur la base du graphe de zone avec UppAal
- Analyser

Références bibliographiques

Références bibliographiques

- Cassandras, C. G. and Lafortune, S. (2008). *Introduction to discrete event systems.* Springer.
- Alur R., Dill D.L., (1994). *A theory of timed automata*. Theoretical Computer Science, Volume 126, Issue 2, 1994, Pages 183-235, ISSN 0304-3975, https://doi.org/10.1016/0304-3975(94)90010-8
- J. Bengtsson et Wang Yi. *Timed Automata Semantics, Algorithms and Tools.* Lecture Notes on Concurrency and Petri Nets, Springer (2004)
- G. Behrmann, A. David et K. Larsen (2006). *A Tutorial on UPPAAL.* Formal Methods for the Design of Real-Time Systems, Springer LNCS 3185, pp. 200-236, http://www.uppaal.com/admin/anvandarfiler/filer/uppaal-tutorial.pdf

Remerciements et crédits

Équipe pédagogique

Auteur.rice.s : Alexandre Philippot et Pascale Marangé

Intervenant.e.s : Alexandre Philippot et Pascale Marangé

Crédits