# Systèmes de transitions - Modélisation TLA+

Durée 1h45 - Documents autorisés

18 mars 2013

## 1 Questions de cours (4 points)

Soit $x$ et $y$ deux variables. Répondre aux questions suivantes en respectant la syntaxe TLA+.

1. Donner un prédicat qui teste si $x$ est un nombre premier (aucun diviseur sauf 1 et lui-même) ;
   $Cardinality(\{i \in Nat : x\%i = 0\}) = 2 \text{ } ou \text{ } \forall i \in 2..(x-1) : x\%i \neq 0$
2. Donner un invariant qui énonce que $x$ est un entier naturel et $y$ un ensemble d'entiers naturels qui contient la valeur de $x$ ;
   $\Box(x \in Nat \land y \in \text{SUBSET } Nat \land x \in y)$
3. Écrire une action exécutable uniquement si $x$ n'est pas dans l'ensemble $y$, et qui dans ce cas change $x$ en une valeur prise dans $y$ ;
   $x \notin y \land x' \in y \land \text{UNCHANGED } y$
4. Donner une expression correspondant à la fonction définie sur l'ensemble $y$ et dont la valeur en $i$ est $x + i$.
   $[i \in y \mapsto x + i]$

## 2 Exercice (6 points)

Soit le module ci-dessous définissant le système de transitions `Spec`.

1. Donner le graphe d'exécution du système de transitions (16 états).
   (merci de limiter le plat de spaghettis : il est possible de le dessiner sans aucun croisement)

```
% Vertical : swap
% --> incrY  (sauf autour de 3/1)
% <-- decrY  (idem)
% Bégaiement partout, non dessiné

% 1/0 <--> 1/1 <--> 1/2 <-----------------------------> 1/3
%  ^                 ^                                  ^
%  |                 |                                  |
%  v                 v                                  v
% 0/1 <-   2/0 <--> 2/1 <--> 2/2 <--     3/3 <-> 3/2 <-> 3/1 <--> 3/0
%  |    \   ^                     \              ^                 ^
%  |     \  |                      \             |                 |
%  |      \ v                       \            v                 v
%  v       0/2                       -------> 2/3                0/3
% 0/0                                                             |
%                                                         (0/2) <--
```

2. Indiquer, en le *justifiant*, si les propriétés suivantes, exprimées en logique LTL ou CTL, sont vérifiées ou pas par `Spec`.

   (a) $\exists\Diamond(y > 5)$
   (b) $\exists\Box(y = 0)$
   (c) $\exists\Box(x + y \leq 1)$
   (d) $\exists\Diamond\forall\Box(x + y \geq 3)$
   (e) $\Box(x + y \leq 6)$
   (f) $\Diamond(y \neq 0)$
   (g) $\Box\Diamond(x = 0)$
   (h) $x > y \leadsto x \leq y$

   *(a) KO ($\Box(y \leq 3)$ d'après le graphe)*
   *(b) KO : WF(swap) $\Rightarrow$ impossible de bégayer sur l'état initial $\Rightarrow$ on va en 1/1 ou 0/1*
   *(c) OK $((1/0) \rightarrow^{swap} (0/1))^\omega$*
   *(d) KO : réinitialisable (1/0 accessible) depuis tout état, sauf 0/0 pas bon non plus.*
   *(e) OK d'après graphe, ou d'après état initial + IncrY $\Rightarrow \Box(y \leq 3)$ et swap $\Rightarrow \Box(x \leq 3)$*
   *(f) OK : WF swap interdit le bégaiement $\Rightarrow$ on va en 1/1 ou 0/1*
   *(g) KO $((1/0) \rightarrow (1/1)^\omega)$*
   *(h) OK : WF swap fait que si $x > y$, on finira par soit incrémenter $y$, soit échanger*

**MODULE** *examen12_test*

EXTENDS *Naturals*
VARIABLES $x$, $y$

$$TypeInvariant \triangleq \land x \in Nat \land y \in Nat$$

$$Swap \triangleq \land x' = y \land y' = x$$
$$IncrY \triangleq \land x \neq 0 \land y' = y + 1 \land y' \leq 3 \land \text{UNCHANGED } x$$
$$DecrY \triangleq \land y > 0 \land y' = y - 1 \land \text{UNCHANGED } x$$

$$Fairness \triangleq \mathrm{WF}_{\langle x,\, y\rangle}(Swap)$$
$$Spec \triangleq \land x = 1 \land y = 0 \land \Box[Swap \lor IncrY \lor DecrY]_{\langle x,\, y\rangle} \land Fairness$$

## 3 Problème (10 points)

On souhaite modéliser et vérifier le problème dit de l'extinction. On considère un système constitué de `NBNODES` sites. Initialement, chaque site possède une couleur, rouge ou vert. Un site vert ne change jamais de couleur. Quand un site rouge contacte un autre site rouge, il devient vert. Quand un site rouge contacte un site vert, il conserve sa couleur. L'objectif est d'étudier les conditions nécessaires et suffisantes pour qu'il ne reste finalement qu'un seul site rouge.

Pour décrire l'interconnexion (le fait qu'un site puisse contacter un autre), on utilise une variable *network* qui contient un **ensemble de couples**, indiquant qu'un site (le premier membre du couple) a la possibilité de contacter un autre site (le second membre). Ainsi, si $\langle i, j\rangle \in network$, cela signifie que le site $i$ peut contacter le site $j$. Dans ce cas, on dit que $i$ est connecté à $j$.

Dans un premier temps, le réseau est fixe (initialisé au départ et inchangé ensuite) puis on considère un réseau évolutif. Un squelette de module TLA+ `redgreen.tla` est fourni en 3.5.

**Note** L'objectif du problème est la modélisation, pas la connaissance des arcanes de TLA+. Si vous ne savez pas comment écrire un opérateur vous pouvez utiliser une syntaxe de votre choix, à condition d'expliquer clairement la signification de cet opérateur.

## 3.1 Module complet

1. Définir le prédicat de transitions `Next` qui représente toutes les transitions possibles.
   $Next \triangleq \exists i, j \in Nodes : turnoff(i, j) \lor addlink(i, j) \lor removelink(i, j)$
2. Définir la propriété `Spec` qui décrit le système de transitions (sans équité).
   $Spec \triangleq Init \land \Box[Next]_{\langle color, network \rangle}$

## 3.2 Propriétés attendues

3. Énoncer une propriété exprimant qu'un site vert le reste définitivement.
   $\forall i \in Nodes : \Box(colors[i] = \text{"}green\text{"} \Rightarrow \Box(colors[i] = \text{"}green\text{"}))$
   *ou* $\forall i \in Nodes : \Box(colors[i] = \text{"}green\text{"} \Rightarrow (colors[i]' = \text{"}green\text{"}))$
4. Énoncer une propriété exprimant que le nombre de sites rouges ne peut pas croître.
   *Avec* $NumberOfRed \triangleq Cardinality(\{i \in Nodes : color[i] = \text{"}red\text{"}\})$ *:*
   $\forall k \in Nat : \Box(NumberOfRed = k \Rightarrow \Box(NumberOfRed \leq k))$
   *ou :* $\Box(NumberOfRed' \leq NumberOfRed)$
5. Énoncer une propriété `OnlyOneRed` exprimant qu'il finit par n'y avoir qu'un seul site rouge.
   *Selon l'interprétation de la phrase :*
   $\Diamond(Cardinality(\{i \in Nodes : color[i] = \text{"}red\text{"}\}) = 1)$
   $\Diamond\Box(Cardinality(\{i \in Nodes : color[i] = \text{"}red\text{"}\}) = 1)$
6. Énoncer une propriété `Connected` exprimant que tout couple de sites $\langle i, j \rangle$ est infiniment souvent connecté.
   $\forall i, j \in Nodes : \Box\Diamond(\langle i, j \rangle \in network)$

## 3.3 Cas 1 : réseau maillé constant

On considère que tout site est connecté à tout autre site (y compris lui-même), et que le réseau est stable (`addlink` et `removelink` restent donc tels que fournis, sans effet).

7. Compléter le prédicat `Init` pour initialiser correctement `network`.
   $network = Nodes \times Nodes$
   *ou* $network = \{\langle i, j \rangle : i, j \in Nodes\}$
8. Expliquer pourquoi il est impossible que la propriété `OnlyOneRed` soit vérifiée sans contrainte d'équité.
   *bégaiement sur l'état initial*
9. Énoncer une propriété d'équité suffisante pour vérifier `OnlyOneRed`. Expliquer informellement pourquoi.
   $\forall i, j \in Nodes : WF_{\langle color, network \rangle}(turnoff(i, j))$
   *turnoff est continûment faisable* $\Rightarrow$ *équité faible suffit.*
10. Expliquer informellement pourquoi `Connected` est trivialement vraie.
    *network est constant et la propriété est initialement vraie* $\Rightarrow$ *toujours vraie*

## 3.4 Cas 2 : réseau évolutif

On considère un réseau initialement vide, dans lequel des connections peuvent être ajoutées ou supprimées. Arbitrairement, un lien absent peut être ajouté, et un lien présent peut être supprimé.

11. Modifier l'action `addlink` pour ajouter un lien non présent.
    $addlink(i, j) \triangleq \langle i, j \rangle \notin network \land network' = network \cup \{\langle i, j \rangle\} \land \text{UNCHANGED } color$

12. Modifier l'action `removelink` pour supprimer un lien présent.
$removelink(i,j) \triangleq \langle i,j\rangle \in network \land network' = network \setminus \{\langle i,j\rangle\} \land \text{UNCHANGED } color$

13. Ajouter une contrainte d'équité pour que `Connected` soit vraie, en la justifiant.
*La seule contrainte sur addlink est que le lien ne soit pas déjà là : addlink est donc continûment faisable si un lien n'est pas présent, donc WF suffit :*
$\forall i,j \in Nodes : WF_{\langle color,network\rangle}(addlink(i,j))$

14. Énoncer et justifier la contrainte d'équité nécessaire pour que la propriété `OnlyOneRed` soit vérifiée.
*Connected est infiniment souvent vrai, mais pas continûment vraie, donc SF :*
$\forall i,j \in Nodes : SF_{\langle color,network\rangle}(turnoff(i,j))$

15. Modifier le modèle (les actions) pour qu'aucune équité forte ne soit nécessaire.
*Ne pas enlever de lien à un site rouge :*
$removelink(i,j) \triangleq$
$\quad color[i] \neq \text{"}red\text{"} \land$
$\quad \langle i,j\rangle \in network \land network' = network \setminus \{\langle i,j\rangle\} \land \text{UNCHANGED } color$
*Ou ne pas enlever de lien entre deux sites rouges.*
*Ou . . .*

### 3.5 Module fourni : `RedGreen.tla`

MODULE *examen12_redgreen*

EXTENDS *Naturals*, *FiniteSets*

CONSTANTS *NBNODES*
$Nodes \triangleq 1 \,..\, NBNODES$
$Colors \triangleq \{\text{"red"}, \text{"green"}\}$

VARIABLES *color*, *network*

$TypeInvariant \triangleq$
$\quad \Box(\land\ color \in [Nodes \to Colors]$
$\qquad \land\ network \in \text{SUBSET } (Nodes \times Nodes))$

---

$Init \triangleq$
$\quad \land\ color \in [Nodes \to Colors]$
$\quad \land\ \exists\, i \in Nodes : color[i] = \text{"red"}$ at least one red
$\quad \land\ network = \{\}$ TO BE CHANGED

$turnoff(i, j) \triangleq$ node $i$ contacts node $j$, and turns off its red color.
$\quad \land\ i \neq j$
$\quad \land\ \langle i, j\rangle \in network$
$\quad \land\ color[i] = \text{"red"}$
$\quad \land\ color[j] = \text{"red"}$
$\quad \land\ color' = [color \text{ EXCEPT } ![j] = \text{"green"}]$
$\quad \land\ \text{UNCHANGED } \langle network\rangle$

$addlink(i, j) \triangleq$ Constant network
$\quad \land\ \text{UNCHANGED } \langle color, network\rangle$

$removelink(i, j) \triangleq$ Constant network
$\quad \land\ \text{UNCHANGED } \langle color, network\rangle$