

Epreuve Écrite



La clarté de la rédaction et la justification des réponses sont des éléments essentiels de l'appréciation. Le devoir devra être remis en fin de séance via Arche et via le mail. Le sujet du message devra être aspd2020 car je filtrerai vos messages en fonction de ce mot. Lorsque je vous demanderai de répondre en utilisant la plateforme toolbox, il faudra me rendre l'archive complète du projet (cf mon fichier joint il faut le fichier .tla mais aussi le dossier .toolbox avec les tests expliqués). Le nom de l'archive que vous rendrez devra être sous la forme votrenom-votreprenom.zip et surtout pas de compresseurs exotiques. De plus, vous aurez besoin de trois fichiers tla appq3.tla, appq41.tla, appq42.tla et donc dans votre réponse vous ajouterez le dossier .toolbox. Pour répondre aux questions textuellement, vous pouvez utiliser ce que vous voulez dans la mesure où le fichier remis sera un fichier pdf produit à partir de votre éditeur. Si vous avez produit un fichier pur texte, c'est aussi acceptable sans produire le fichier pdf.

Ecrit

Exercice 1 *Un ensemble de processus P_1, \dots, P_n partagent une ressource R qui n'est utilisable que par un seul processus à la fois. Pour utiliser cette ressource, un serveur Q gère la ressource et son utilisation selon les règles données. Les processus P_1, \dots, P_n communiquent par envoi de messages à Q .*

*Proposer une technique qui organise la gestion de cette ressource par Q . Vous pouvez utiliser des primitives comme $\text{send}(\text{mes}, Q)$ pour envoyer le message mes à Q et $\text{receive}(\text{mes}, \text{cond}(\text{mes}), Q)$ pour attendre de recevoir un message mes satisfaisant $\text{cond}(\text{mes})$. Pour écrire les algorithmes, vous utiliserez les constructions classiques *if*, *while*, ...*

Exercice 2 *On considère le problème de l'exclusion mutuelle pour un ensemble de processus connectés tous les uns aux autres par un réseau fiable.*

Question 2.1 *Donner le nombre de messages nécessaires pour entrer en section critique dans le cas de l'algorithme de Lamport.*

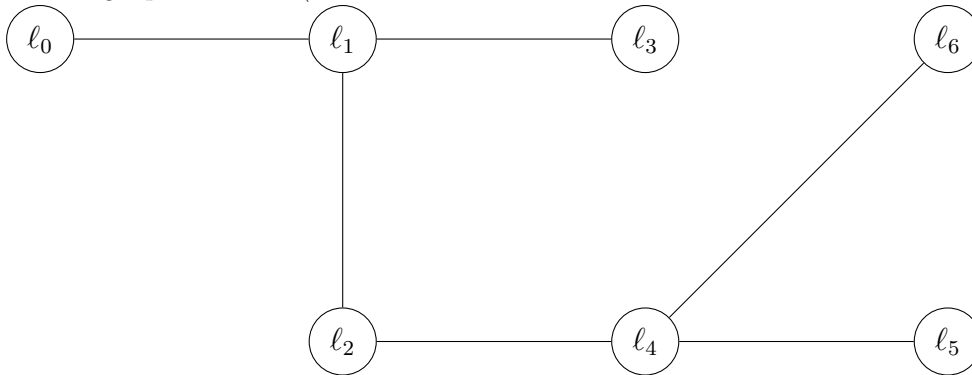
Question 2.2 *Donner le nombre de messages nécessaires pour entrer en section critique dans le cas de l'algorithme de Ricart et Agrawala.*

Question 2.3 *Donner le nombre de messages nécessaires pour entrer en section critique dans le cas de l'algorithme de Carvahlo et Roucairol.*

Question 2.4 *Donner le nombre de messages nécessaires pour entrer en section critique dans le cas de l'algorithme de l'exercice 1.*

Exercice 3 L'exercice consiste à étudier la version TLA/TLA⁺ écrite à partir des modèles Event-B permettant de produire un algorithme local de l'élection du leader dans un graphe acyclique. Le module TLA est donné et est dans le fichier `ieee2020.tla`, sur arche. Il existe des fichiers `tla` sur le web qui peuvent contenir les mêmes informations mais pas les mêmes questions et donc il est hasardeux de chercher la solution sur Internet.

Soit le graphe suivant(:

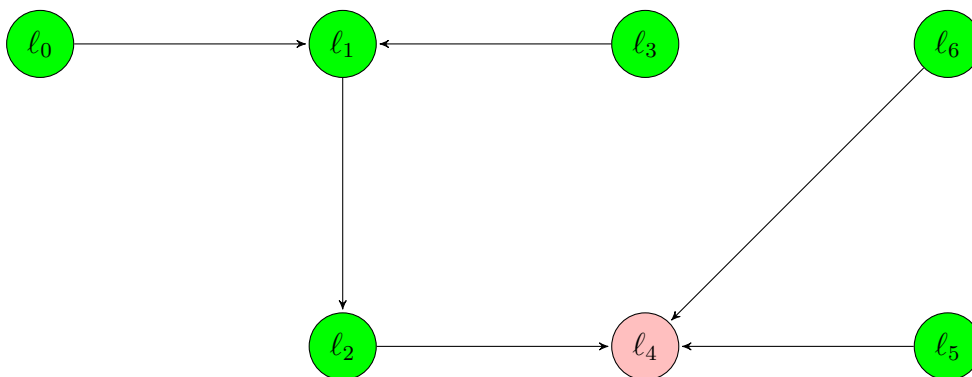


Question 3.1 Configurer le modèle TLA pour ce graphe en accord avec sa représentation.

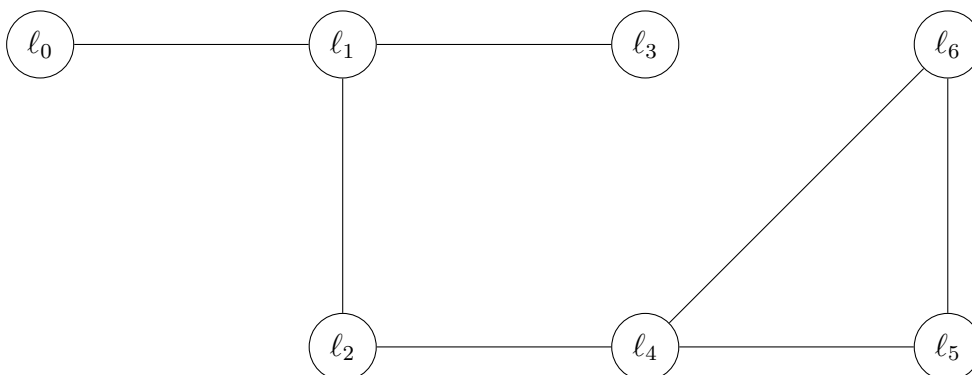
Question 3.2 Vérifier qu'il est possible d'élire un leader pour ce graphe en posant une question de sûreté.

Question 3.3 En utilisant les transparents du cours, écrire une propriété invariante et vérifier cette propriété avec l'outil.

Question 3.4 Modifier une ou plusieurs actions pour forcer l'élection d'un des nœuds et démontrer expérimentalement que cela fonctionne. Pour cela, on pourra introduire une constante `ROOTFORCE` pour permettre le choix d'un nœud parmi l'ensemble des nœuds possibles. On donne par exemple `ROOTFORCE = 4` et on obtient le graphe orienté suivant.



Question 3.5 Nous avons maintenant un nouveau problème à résoudre. En effet, le graphe est supposé acyclique mais est en fait cyclique. Il faut donc détecter cette propriété et pour cela on peut placer une variable `time` qui croît à chaque transition du système tant qu'elle est plus petite qu'une valeur de référence pour la convergence notée `TIMEOUT`. Modifier la solution en intégrant ces données et proposer une solution à la détection.



Exercice 4 Question 4.1 On décide d'élire un leader en résolvant la contention par échange de identités supposées connues. Dans notre exemple, les identités sont les numéros des étiquettes $\ell_0 \dots \ell_6$. Modifier la solution initiale en utilisant la valeur des nœuds en contention et en choisissant toujours soit le plus petit soit le plus grand comme leader. Valider votre solution par un argumentaire fondé sur une analyse avec l'outil Toolbox. On appellera appq41.tla ce nouveau module.

Question 4.2 (Exercice Kangourou)

On se pose une autre question qui est d'utiliser l'un des algorithmes que nous avons décrit en l'appliquant sur un réseau mobile. Un réseau mobile est en fait un réseau qui est soumis à des actions de destruction d'un lien entre deux nœuds ou création d'un lien entre deux nœuds. Un lien non orienté créé signifie que les deux nœuds sont proches et donc il faut ensuite orienter le lien pour avoir un arbre. Complétez la définition de ce nouvel algorithme en créant un nouveau module appq42.tla. Pour vous aider, nous donnons les opérations nouvelles autorisées qui modifient le réseau en créant ou supprimant la connexion et les nouvelles opérations de réorientations. Nous donnons aussi un scénario qui explique ce qui se passe. Vous devez toujours donner tous les éléments d'un projet toolbox. Il faut donc créer un modèle.

```
axm1 :  $g \in N \leftrightarrow N$ 
axm2 :  $g = g^{-1}$ 
axm3 :  $id \cap g = \emptyset$ 
inv1 :  $t \in N \rightarrow N$ 
inv3 :  $t \cap t^{-1} = \emptyset$ 
inv11 :  $\forall S. (S \subseteq N \wedge S \subseteq t^{-1}[S] \Rightarrow S = \emptyset)$ 
inv21 :  $t \subseteq g$ 
inv22 :  $token \subseteq N$ 
inv23 :  $dom(t) \subseteq N \setminus token$ 
```

```
EVENT operation
BEGIN
    act1 :  $g, t, token : |$ 
    
$$\left( \begin{array}{l} g' \in N \leftrightarrow N \wedge \\ g' = g'^{-1} \wedge \\ id \cap g' = \emptyset \wedge \\ t' \in N \rightarrow N \wedge \\ t' \cap t'^{-1} = \emptyset \wedge \\ \forall S. (S \subseteq N \wedge \\ S \subseteq t'^{-1}[S] \Rightarrow S = \emptyset) \wedge \\ t' \subseteq g' \wedge token' \subseteq N \wedge \\ dom(t') \subseteq N \setminus token' \end{array} \right)$$

END
```

```
EVENT removing
ANY
 $x, y$ 
WHERE
    grd1 :  $x \mapsto y \in g$ 
THEN
    act1 :  $g := g \setminus \{x \mapsto y, y \mapsto x\}$ 
    act2 :  $t := t \setminus \{x \mapsto y, y \mapsto x\}$ 
END
```

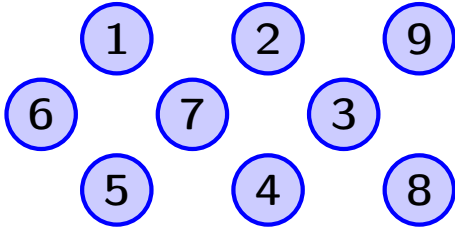
```
EVENT adding
ANY
 $x, y$ 
WHERE
    grd1 :  $x \in N$ 
    grd2 :  $y \in N$ 
    grd3 :  $x \mapsto y \notin g$ 
    grd4 :  $x \neq y$ 
THEN
    act1 :  $g := g \cup \{x \mapsto y, y \mapsto x\}$ 
END
```

```
EVENT op1
REFINES operation
ANY
 $x, y$ 
WHERE
    grd1 :  $x \mapsto y \in g$ 
    grd2 :  $x \notin dom(t)$ 
    grd3 :  $y \notin dom(t)$ 
THEN
    act1 :  $t := t \cup \{x \mapsto y\}$ 
    act2 :  $token := token \setminus \{x\}$ 
END
```

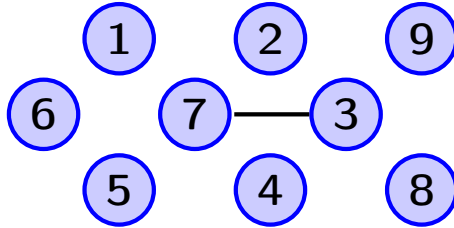
```
EVENT op2
REFINES operation
ANY
 $x, y$ 
WHERE
    grd1 :  $x \mapsto y \in t$ 
    grd2 :  $y \notin dom(t)$ 
THEN
    act1 :  $t := (t \setminus \{x \mapsto y\}) \cup \{y \mapsto x\}$ 
    act2 :  $token := (token \setminus \{y\}) \cup \{x\}$ 
END
```

```
EVENT op3
REFINES operation
ANY
 $x$ 
WHERE
    grd1 :  $x \notin token$ 
    grd2 :  $x \notin dom(t)$ 
    grd3 :  $x \notin dom(g)$ 
THEN
    act1 :  $token := token \cup \{x\}$ 
END
```

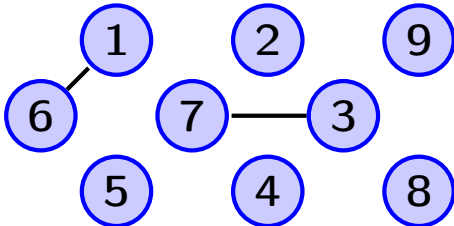
(1) each node has a token



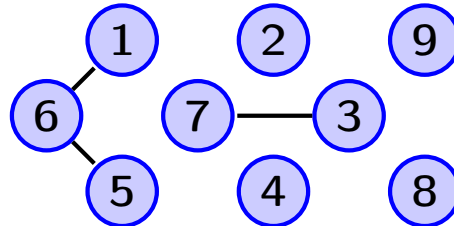
(2) a link is created between two nodes



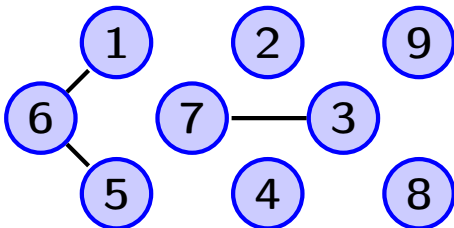
(3) a link is created between two nodes



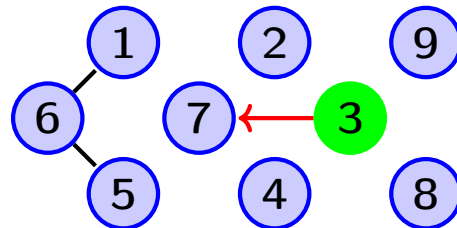
(4) a link is created between two nodes



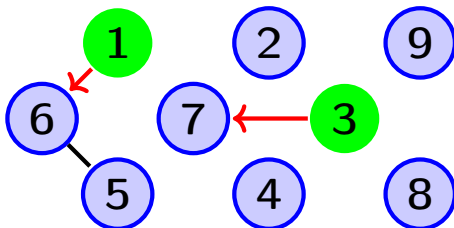
(5) a link is created between two nodes



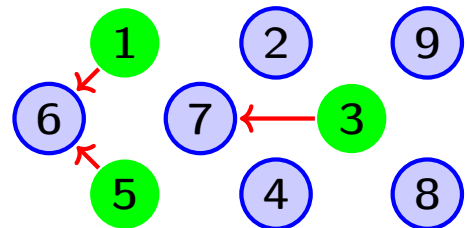
(6) op1 is applied between two nodes 7 and 3



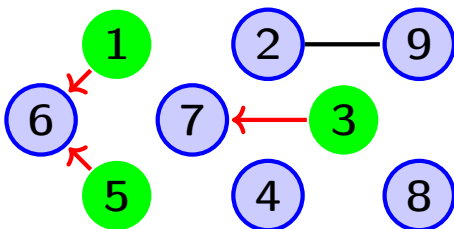
(7) op1 is applied between two nodes 6 and 1



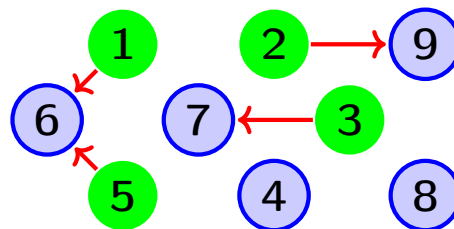
(8) op1 is applied between two nodes 6 and 5



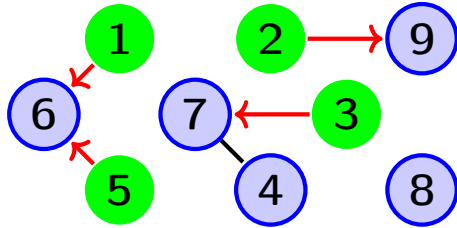
(9) creating a link between two nodes



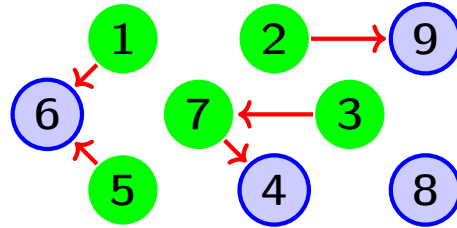
(10) op1 is applied between two nodes 9 and 2



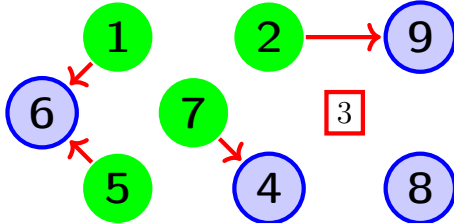
(11) a link is created between two nodes



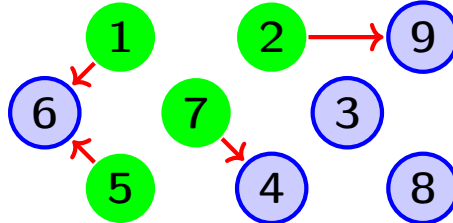
(12) op1 is applied on 7 and 4



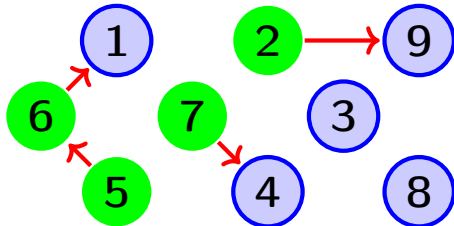
(13) deleting the link between 7 and 3



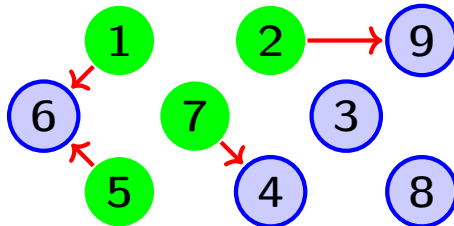
(14) op3 is applied on 3



(15) op2 is applied on 6 and 1



(16) op2 is applied on 6 and 1



Fin de l'énoncé