

DEVOIR SURVEILLÉ

SPÉCIFICATION ET VÉRIFICATION FORMELLES

Enseignantes	: Mme. M. Fourati & Mme. E. Menif	Nom	:
Niveau	: 3 ^{ème} année ingénierie des systèmes intelligents	Prénom	:
Documents et calculatrices	: Non autorisées	Groupe	:

SPÉCIFICATION FORMELLE: (10 points)

Exercice 1 : Questions de cours (2 points)

	Vrai	Faux
1. Un langage de programmation peut être un langage de spécification formelle	<input type="checkbox"/>	<input type="checkbox"/>
2. Les besoins énoncés sont le fruit de la phase d'analyse des besoins	<input type="checkbox"/>	<input type="checkbox"/>
3. La spécification est conduite au début de chaque phase de du cycle de vie d'un logiciel	<input type="checkbox"/>	<input type="checkbox"/>
4. La sur-spécification est un élément de la spécification formelle ne correspondant pas à une caractéristique du problème mais plutôt de la solution	<input type="checkbox"/>	<input type="checkbox"/>
5. En présence de la spécification formelle, la spécification en langage naturelle n'est plus nécessaire	<input type="checkbox"/>	<input type="checkbox"/>
6. $\text{seq } X \triangleq \{f: \mathbb{N} \rightarrow X\}$	<input type="checkbox"/>	<input type="checkbox"/>
7. L'approche axiomatique fait partie des méthodes orientées modèle	<input type="checkbox"/>	<input type="checkbox"/>
8. Dans Z, tout ensemble est un type	<input type="checkbox"/>	<input type="checkbox"/>

Exercice 2 (8 points)

On s'intéresse à la spécification du jeu de société Scrabble. Ce jeu consiste à former des mots entrecroisés sur une grille avec des lettres de valeurs différentes. Le vainqueur est le joueur qui cumule le plus grand nombre de points à l'issue de la partie. Le jeu se compose, notamment :

- d'un plateau de 225 cases ;
- de 102 jetons dont 2 jokers; un jeton correspond à une lettre et un ombre de points
- de chevalets sur lesquels chaque joueur pose ses lettres.



On ne va spécifier qu'une version simplifiée de ce jeu. On fera abstraction des jokers et des cases multiplicatrices du plateau.

- Le score d'un coup est calculé en additionnant la valeur de toutes les lettres des nouveaux mots formés.
- Pour commencer, chaque joueur va tirer au hasard 7 jetons dans le sac. Le joueur peut avoir plusieurs jetons identiques
- Chaque joueur doit former un mot (suite de jetons) à partir des jetons qu'il détient.
- Après avoir placé un mot, le joueur doit piocher autant de jetons que ceux placés sur le plateau pour qu'il ait toujours un nombre total de 7 jetons sur son chevalet.

A ₁	B ₄	C ₄	D ₄	E ₄	F ₄
G ₂	H ₄	I ₁	J ₈	K ₅	L ₁
M ₃	N ₁	O ₁	P ₃	Q ₁₀	R ₄
S ₄	T ₁	U ₁	V ₄	W ₄	X ₈
	Y ₄	Z ₁₀			

On considère les types suivants :

Lettre ::= A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z

Points $\triangleq \{1,2,3,4,8,10\}$

Jetons : Lettre \rightarrow Points

Nous supposons avoir les opérations **first**, **second** et **Somme**. L'opération **first** prend un couple et retourne le premier élément du couple. L'opération **second** prend un couple et retourne le deuxième élément du couple. L'opération **Somme** qui prend un ensemble d'entiers et retourne la somme de ces entiers.

1. Définir les ensembles **Voyelles** et **Consonnes** à partir de l'ensemble **Lettre**.

2. Définir l'opération **Jouer** qui prend l'ensemble des jetons que possède un joueur, l'ensemble des jetons qu'il va jouer pour former son mot et retourne l'ensemble des jetons qui lui reste.

.....
.....
.....
.....
.....

3. Définir l'opération **NombreJetons** qui prend l'ensemble des jetons en possession du joueur et retourne le nombre des jetons.

.....
.....
.....
.....
.....

4. Définir l'opération **PrendreJetons** qui prend l'ensemble des jetons en possession du joueur après avoir joué, l'ensemble des jetons qu'il a pioché et retourne l'ensemble des jetons qui seront sur son chevalet.

.....
.....
.....
.....
.....

5. En considérant le mot formé par un joueur comme une suite de jetons. Définir l'opération **ScoreMot** qui prend le mot composé par le joueur et retourne le nombre de points correspondant à ce mot.

.....
.....
.....
.....
.....

6. Définir l'opération **VoyellesMot** qui prend l'ensemble des jetons en possession du joueur et retourne le nombre des voyelles présentes

VÉRIFICATION FORMELLE : (10 points)

Exercice 1 : Questions de cours (1.5 points) 0.25/rép

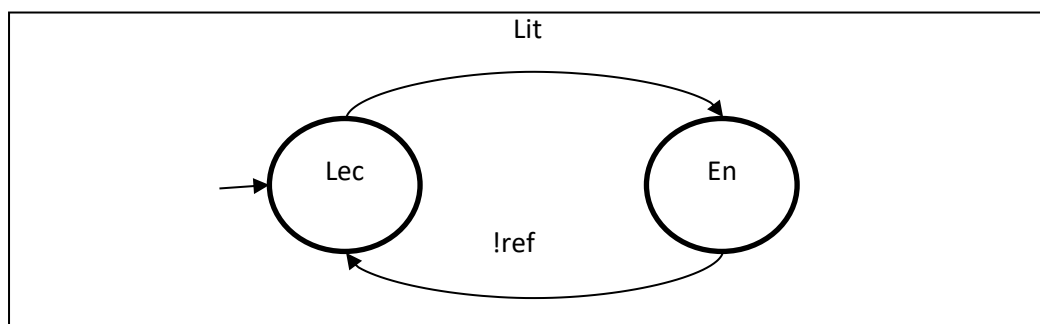
	Vrai	Faux
• La logique LTL est plus expressive que la logique CTL.	<input type="checkbox"/>	<input checked="" type="checkbox"/>
• La logique CTL est plus expressive que la logique LTL.	<input type="checkbox"/>	<input checked="" type="checkbox"/>
• $\mathbf{F}p$ est vraie lorsque p est vrai dans l'état courant.	<input checked="" type="checkbox"/>	<input type="checkbox"/>
• $p\mathbf{A}Uq$ est vraie lorsque q est vrai dans l'état courant.	<input checked="" type="checkbox"/>	<input type="checkbox"/>
• $\mathbf{GF}p$ est vrai s'il existe une exécution dans laquelle p est toujours vrai et à un certain moment, p devient toujours faux.	<input type="checkbox"/>	<input checked="" type="checkbox"/>
• Le model checking consiste à donner une preuve formelle de la satisfaction d'une formule.	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Exercice 2 : Modélisation et produit d'automates (5 points)

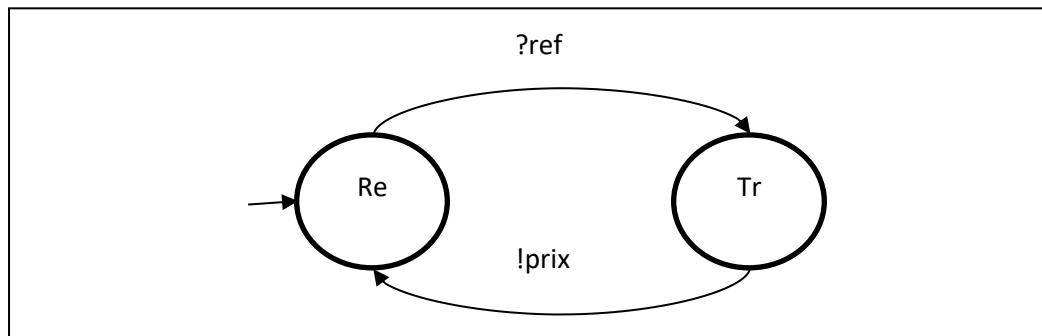
Considérons un système (très simplifié) de gestion des prix des produits dans un supermarché. Le système consiste en trois processus : un lecteur de code à barre (LCB) qui lit le code et communique sa référence au programme d'attribution des prix (AP). En recevant cette référence, le programme AP transmet le prix de l'article à l'imprimante (IMP) qui génère un ticket (contenant la référence de l'article et son prix). Uniquement le ticket est important, nous ne nous intéressons pas à son contenu.

Sachez que le but de l'exercice est de synchroniser les trois processus LCB, AP et IMP par envoie/réception de message et qu'il peut y avoir des actions d'envoi/réception et d'autres internes aux processus.

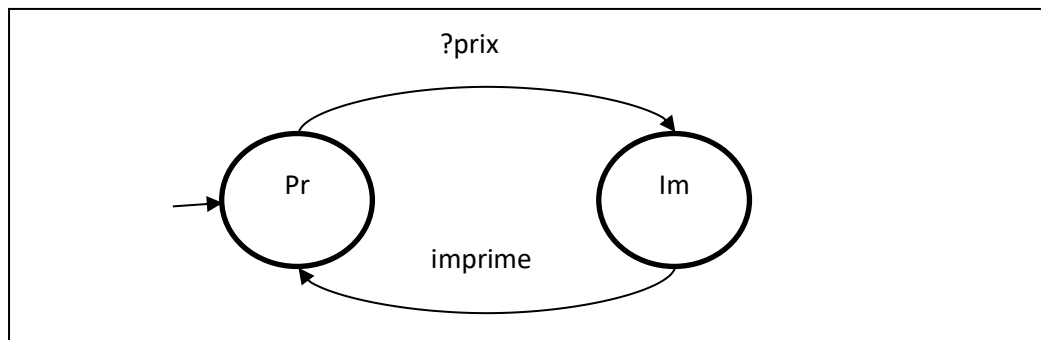
1. Modéliser le processus LCB par une structure de Kripke. Indiquer l'état initial et les actions de ce dernier. **1 point**



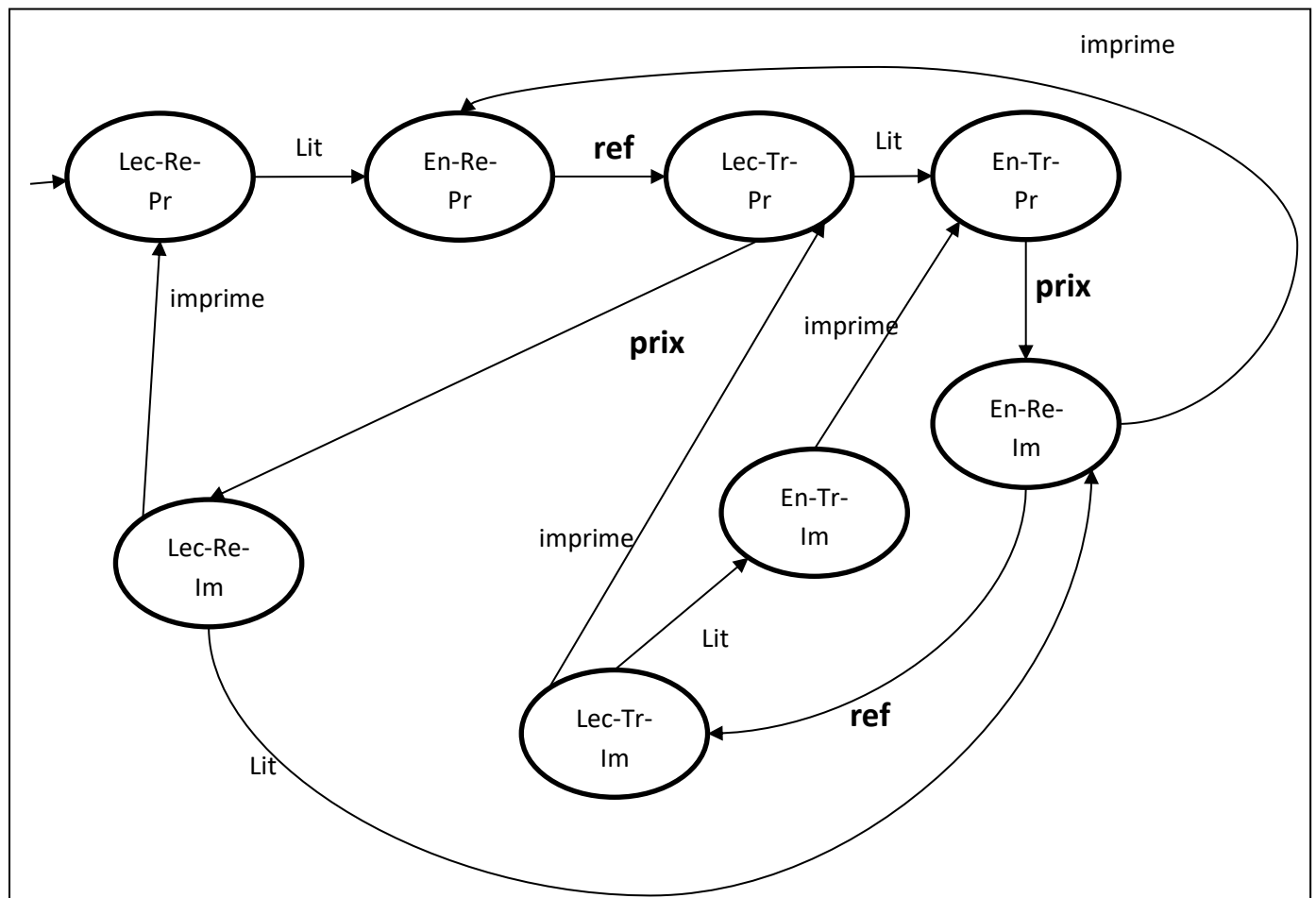
2. Modéliser le processus AP par une structure de Kripke. Indiquer l'état initial et les actions de ce dernier. **1 point**



3. Modéliser le processus IMP par une structure de Kripke. Indiquer l'état initial et les actions de ce dernier. **1 point**



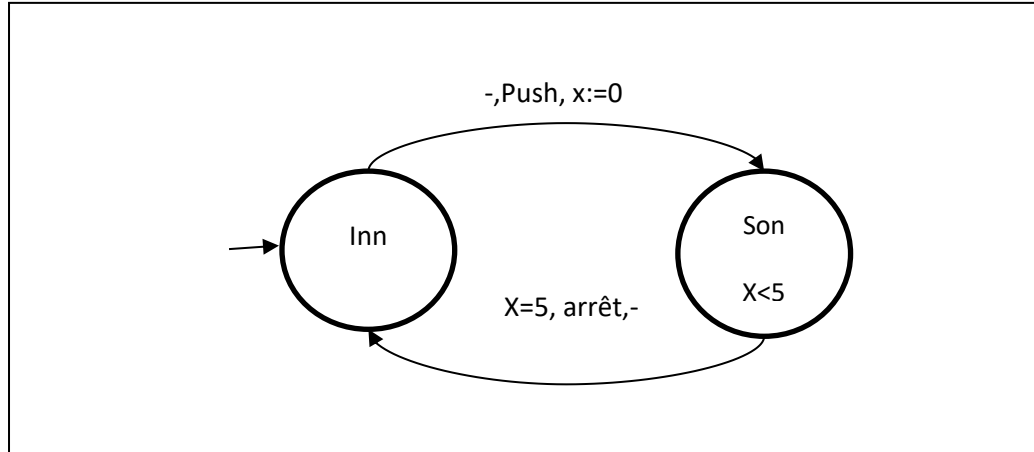
4. Synchroniser les trois processus par envoi/réception de messages. **3 points**



Exercice 2 : Automate temporisé (2.5 points)

Nous voulons modéliser une sonnerie, lorsque l'utilisateur appuie sur le bouton *push*, la sonnerie va sonner durant exactement 5 secondes. Modéliser la sonnerie par un automate temporisé en évitant d'aboutir à un blocage sur les actions franchissables.

1. Modélisez par un automate temporisé M la sonnerie. **2 points**



2. Donnez une exécution de l'automate M contenant exactement 4 transitions (5 configurations) et qui revient à l'état initial. **0.5 point**

$$(Inn, 0) \xrightarrow{3} (Inn, 3) \xrightarrow{push} (Son, 0) \xrightarrow{4} (Son, 4) \xrightarrow{1} (Inn, 5)$$