
EXAMEN D'INGENIERIE DES SYSTEMES D'INFORMATION

PREMIERE PARTIE : MODELISATION UML

Exercice 1. RELATIONS STATIQUES – 7 POINTS

Modéliser les phrases suivantes à l'aide d'un diagramme de classes.

- (a) **1 point** – Les enfants sont des personnes.
- (b) **1,5 points** – Les participants sont répartis en groupes de 10 personnes, et chaque groupe choisit parmi ses membres un chef de groupe.
- (c) **1,5 points** – Le panier de bienvenue est constitué de quatre cadeaux, à choisir parmi des fruits, des chocolats, ou des porte-clés lumineux.
- (d) **1,5 points** – Ce livre contient 12 chapitres de 50 pages chacun.
- (e) **1,5 points** – Les étourneaux, qui sont des oiseaux, peuvent voler en formations appelées murmurations.

Exercice 2. UN SYSTEME DE GESTION DE LEGO – 9 POINTS

Un collectionneur souhaite organiser ses briques lego à l'aide d'un système d'information. Ce système doit permettre au collectionneur gérer son stock de briques (voir l'état courant du stock, retirer des briques, ou ajouter des briques), et de consulter les notices des différents modèles (disponibles par une API fournie par le constructeur). Il sera aussi possible, au cours de cette consultation, de sélectionner une notice pour construire le modèle. Le système devra alors sortir de l'inventaire les briques nécessaires à la construction. Enfin, à chaque fois qu'un modèle est construit par le collectionneur, il pourra demander au système d'en capturer une photo, et de la poster sur un réseau social. Le système supporte Instagram, Twitter, et LinkedIn.

Question 1. 2 points – Réaliser un diagramme de cas d'utilisation pour ce système. Attention à bien identifier les acteurs et les cas d'utilisation.

Question 2. 2 points – Présenter sous forme d'un diagramme d'activités système la consultation d'une notice à l'aide du système, en incluant les différents scénarios (en particulier la sélection de notice).

Question 3. 3 points – Décrire en détails la publication de photos. On attend une description textuelle structurée, ainsi qu'un diagramme de séquence système d'un scénario nominal.

Question 4. 2 points – Établir un diagramme de classes métier pour ce système.

DEUXIEME PARTIE : MACHINES DE MEALY ET APPRENTISSAGE DE MODELES

Exercice 3. MACHINES DE MEALY – 4,5 POINTS

Proposer des machines de Mealy réalisant chacune des fonctions suivantes :

- (a) 1 point – Sur l'alphabet $\{a,b,c\}$, remplace les a et les b par 0, et les c par 1.

Exemple :

aababcabccb \mapsto *00000100110*

Entrée	a	a	b	a	b	c	a	b	c	c	b
Sortie	0	0	0	0	0	1	0	0	1	1	0

- (b) 1,5 points – Sur l'alphabet $\{a\}$, sort un 0 chaque position divisible par 3, et un 1 sinon.

Exemple : *aaaaaa* \mapsto *1101101*

Entrée	a	a	a	a	a	a	a	a	a	a
Sortie	1	1	0	1	1	0	1	1	0	1

- (c) 2 points – Sur l'alphabet $\{0,1\}$, encode l'entrée avec le motif 010 comme suit : on itère le motif jusqu'à atteindre la longueur de l'entrée (par exemple pour une entrée de taille 7 on prend 0100100), puis on effectue un « ou exclusif » bit à bit entre entrée et motif itéré.

Exemple : pour *11010101*, on prend le motif itéré *01001001*, et on obtient *10011100*, comme illustré par le tableau suivant :

Entrée	1	1	0	1	0	1	0	1
Motif	0	1	0	0	1	0	0	1
Sortie	1	0	0	1	1	1	0	0

Rappel : l'opérateur « ou exclusif » (aussi appelé xor) est une opération binaire sur les booléens, qui renvoie 1 si et seulement si un de ses arguments vaut 1 et l'autre 0. La table de vérité de cet opérateur est donnée ci-contre.

a	b	a xor b
0	0	0
0	1	1
1	0	1
1	1	0

Exercice 4. REGISTRE PARTAGE – 5,5 POINTS

Considérons un registre partagé par n processus p_1, \dots, p_n . Ce registre peut prendre k valeurs, notées v_1, \dots, v_k . Il contient initialement la valeur v_1 . Chacun des processus peut à tout moment lire la valeur contenue dans le registre : pour cela, p_i peut exécuter l'instruction « *read(i)* », et observer en sortie un message « *val(x)* », où x est la valeur courante du registre. Pour éviter des problèmes liés aux accès concurrents au registre, celui-ci a été équipé d'un verrou. Le registre est initialement déverrouillé. Lorsqu'il souhaite modifier le contenu du registre, le processus p_i doit d'abord verrouiller le registre avec l'instruction « *lock(i)* ». Si le registre était déjà verrouillé, un message « *error* » est produit en sortie. En revanche si le registre était déverrouillé, un message « *ok* » est produit et p_i peut écrire une nouvelle valeur y dans le registre, via l'instruction « *write(i, y)* », qui dans ce cas produit également un message « *ok* ». Lorsque le registre est ainsi verrouillé, seul p_i peut le déverrouiller avec l'instruction « *unlock(i)* ». Toute tentative de verrouillage, ou de déverrouillage ou d'écriture par un autre processus, donne lieu à un message « *error* ».

On souhaite modéliser ce registre par une machine de Mealy.

Question 1. 0,5 points – Quel sont les alphabets d'entrée et de sortie de ce système ?

Question 2. 2 points – Construire une machine de Mealy pour ce système avec $n = k = 2$.

Question 3. 3 points – Décrire les états et les transitions de la machine dans le cas général.

Exercice 5. DEVINER LA MACHINE MYSTERE – 5 POINTS

Construire une machine de Mealy sur l'alphabet $\{a,b\}$ avec aussi peu d'états que possible, et telle que l'image des mots de taille 6 est comme dans le tableau suivant :

aaaaaa	BBBBBB	ababaa	BABABB	babaaa	ABABBB	babbaa	ABABBB
aaaaab	BBBBBA	ababab	BABABA	babaab	ABABBA	babbab	ABABBA
aaaaba	BBBBBAB	ababba	BABABB	bababa	ABABAB	babbaa	ABABAB
aaaabb	BBBBBAB	ababbb	BABABA	bababb	ABABAB	babbbb	ABABAA
aaabaa	BBBABB	abbaaa	BABBBB	babbaa	ABABBB	bbaaaa	AABBBB
aaabab	BBBABA	abbaab	BABBBB	babbab	ABABBA	bbaaab	AABBBB
aaabba	BBBABB	abbaba	BABBBB	babbaa	ABABAB	bbabab	AABBBB
aaabbb	BBBABA	abbabb	BABBAB	babbbb	ABABAA	bbbaab	AABBAB
aabaaa	BBABBB	abbbba	BABABB	baaaaa	ABBBBB	bbabaa	AABABB
aabaab	BBABBA	abbbab	BABABA	baaaab	ABBBBB	bbabab	AABABA
aababa	BBABAB	abbbba	BABAAB	baaaba	ABBBB	bbabba	AABABB
aababb	BBABAB	abbbbb	BABAAA	baaaba	ABBBB	bbabbb	AABABA
aabbaa	BBABBB	baaaaa	ABBBBB	baaabb	ABBBB	bbbbaa	AABBBB
aabbab	BBABBA	baaaab	ABBBB	baabab	ABBABA	bbbbaa	AABBBB
aabbba	BBABAB	baaaba	ABBBB	baabba	ABBABB	bbbaba	AABBBB
aabbbb	BBABAA	baaabb	ABBBB	baabbb	ABBABA	bbbabb	AABBBB
abaaaa	BABBBB	baabaa	ABBABB	baabaa	ABBABB	bbbbaa	AABBBB
abaaaab	BABBBB	baabab	ABBABA	babaab	ABBABA	bbbbaa	AABBBB
abaaaba	BABBAB	baabba	ABBABB	bababa	ABBABA	bbbbaa	AABBBB
abaabb	BABBAB	baabbb	ABBABA	bababb	ABBABA	bbbbaa	AABBBB

Exercice 6. REGISTRE PARTAGE (SEULEMENT POUR LES ING3) – 4 POINTS

Considérons un registre partagé par 2 processus p_1, p_2 . Ce registre peut contenir 1 bit d'information, prenant les valeurs 0 ou 1, et contient initialement la valeur 0. Chacun des processus peut à tout moment lire la valeur contenue dans le registre : pour cela, p_i peut exécuter l'instruction « *read(i)* », et observer en sortie un message « *val(x)* », où x est la valeur courante du registre. Pour éviter des problèmes liés aux accès concurrents au registre, celui-ci a été équipé d'un verrou. Le registre est initialement déverrouillé. Lorsqu'il souhaite modifier le contenu du registre, le processus p_i doit d'abord verrouiller le registre avec l'instruction « *lock(i)* ». Si le registre était déjà verrouillé, un message « *error* » est produit en sortie. En revanche si le registre était déverrouillé, un message « *ok* » est produit et p_i peut écrire une nouvelle valeur y dans le registre, via l'instruction « *write(i,y)* » (qui dans ce cas produit également un message « *ok* ». Lorsque le registre est ainsi verrouillé, seul p_i peut le déverrouiller avec l'instruction « *unlock(i)* ». Toute tentative de verrouillage, ou de déverrouillage ou d'écriture par un autre processus, donne lieu à un message « *error* ».

Modéliser ce registre à l'aide d'un diagramme état-transitions.