



Université d'Ottawa · University of Ottawa  
SCHOOL OF INFORMATION TECHNOLOGY AND ENGINEERING

**COURS:** SEG2501  
**SEMESTRE:** Hiver 2004

**PROFESSEUR:** Gregor v. Bochmann  
**DATE:** 21 février, 2004  
**TEMPS:** 10:00 à 11:20

**Examen mi-session**

**NOM ET NUMÉRO DE L'ÉTUDIANT:** \_\_\_\_\_ / \_\_\_\_\_

**Examen mi-session**

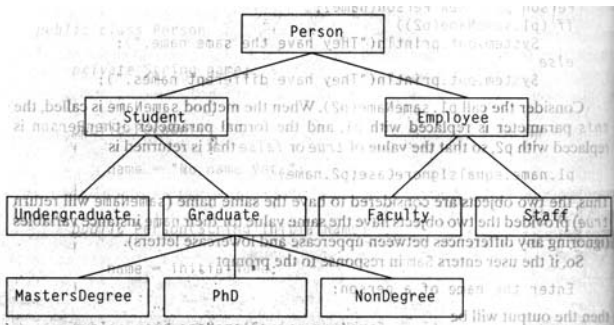
1. Il y a trois types de questions:

<b>Partie 1</b>	<b>Choix multiples</b>	<b>20 points</b>	
<b>Partie 2</b>	<b>Réponses brèves</b>	<b>20 points</b>	
<b>Partie 3</b>	<b>Solution de problèmes</b>	<b>20 points</b>	
<b>Total</b>		<b>60 points</b>	

2. L'espace alloué à chaque question est limité. Si nécessaire, vous pouvez utiliser l'autre côté des pages pour continuer votre réponse.

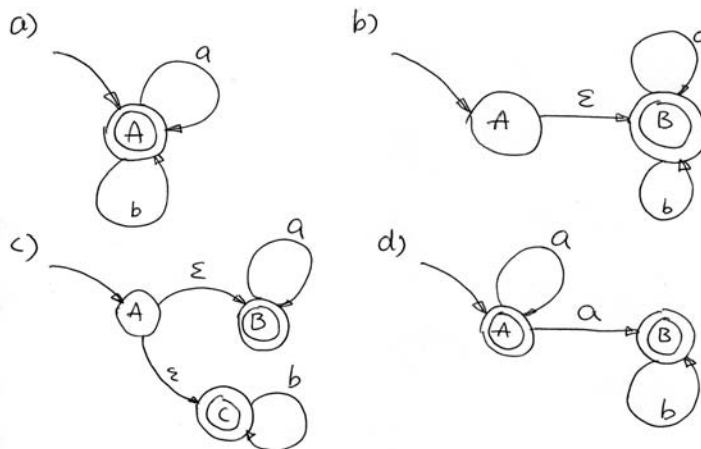
▪ **Questions à choix multiples [2 points chaque, s.v.p. encerclez la meilleur réponse:**

- Lequel des énoncés suivants est correct ?
  - La validation est une activité pour déterminer si le système est conforme à sa spécification.
  - La vérification est une activité pour déterminer si le système satisfait aux besoins des usagers et propriétaires.
  - La qualité du processus correspond à la conformité entre le système et les exigences.
  - Qualité du processus et qualité du système sont deux noms pour la même chose.
- La compagnie ABC réalisait la définition des exigences du 1<sup>er</sup> janvier au 31 mars, la conception fonctionnelle du 1<sup>er</sup> avril au 30 juin, la conception de l'implantation du 1<sup>er</sup> juillet au 30 septembre, et finalement l'implantation du 1<sup>er</sup> octobre au 31 décembre. Lequel des énoncés suivants n'est pas correct ?
  - La période du 1<sup>er</sup> avril au 30 juin est une des phases.
  - Une transition de phase est aussi appelée un "milestone".
  - Une activité sert à produire des résultats.
  - Le 1<sup>er</sup> juillet est une des "baselines".

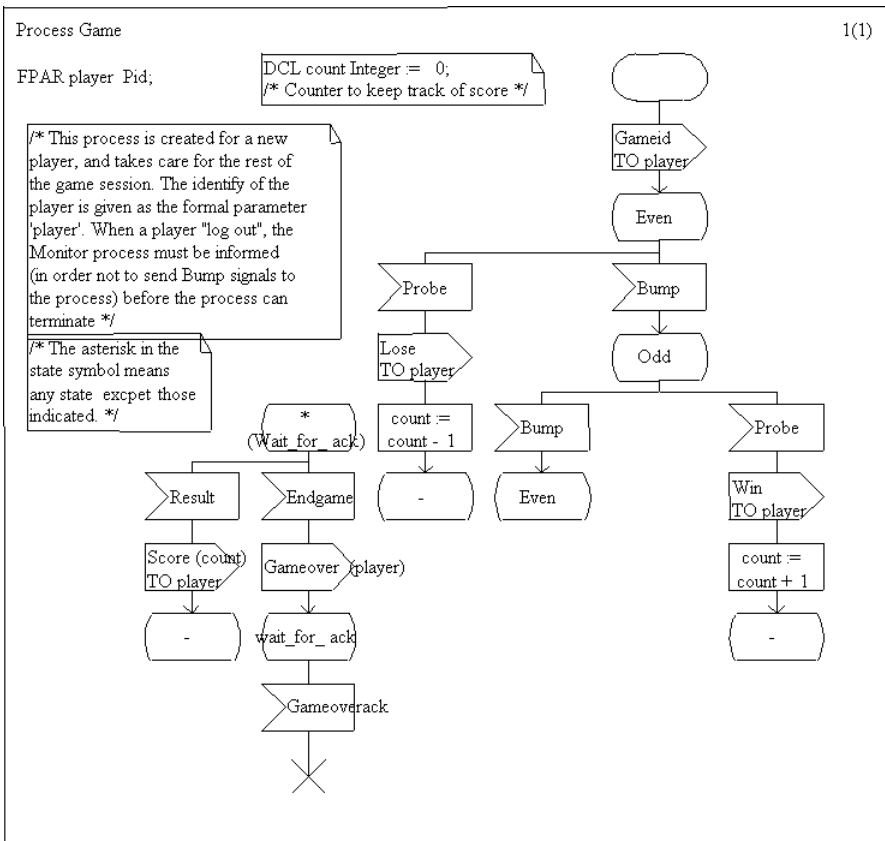
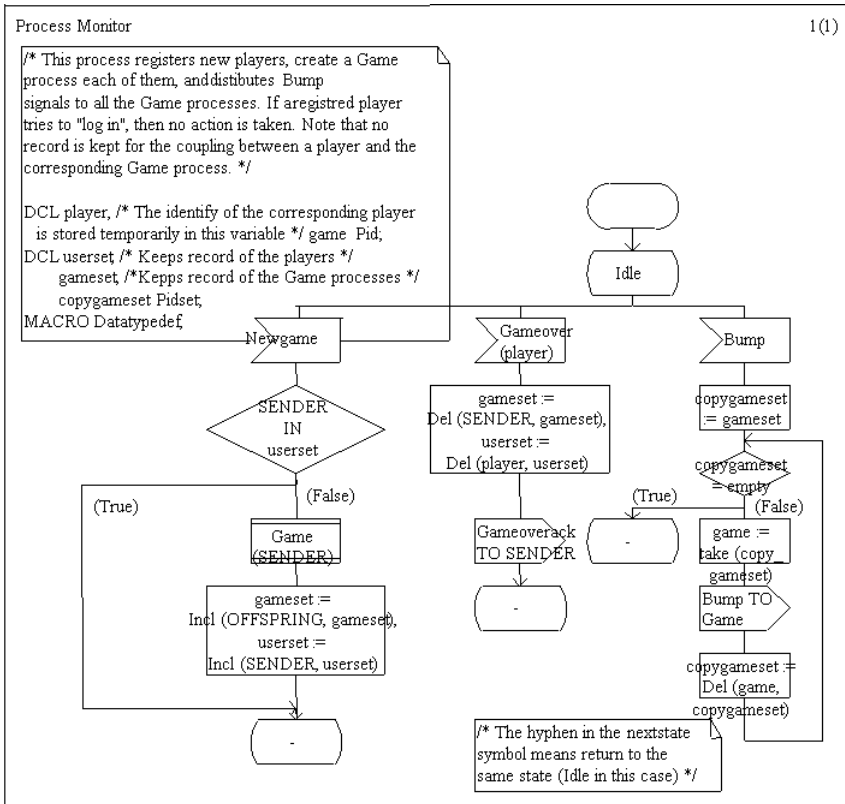


- On se réfère à la figure ci-haut. Nous pouvons définir une classe très générale *Person*, et puis définir des classes plus spécialisées comme *Students*, *PhD*, en ajoutant simplement de nouveaux détails aux classes déjà existantes. Cette façon de procéder peut être décrite par lequel des termes suivants?
  - Projection
  - Agrégation
  - Héritage
  - Partitionnement

- Lequel des diagrammes suivants définit un automate déterministe ?

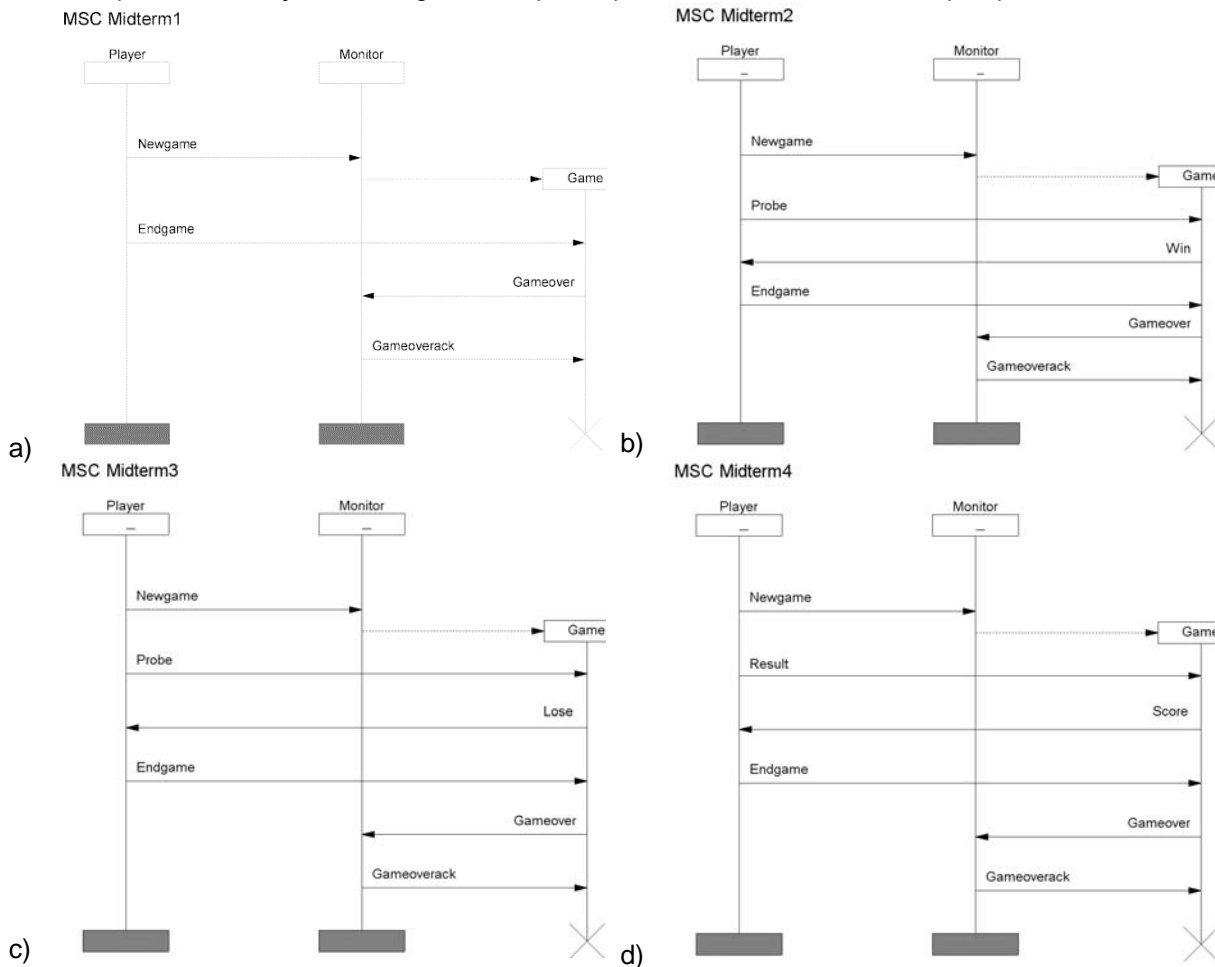


- Référez au processus *Monitor* dans la première figure sur la page suivante. Lequel des énoncés suivants n'est pas correct ?
  - Le processus a un seul état.
  - Le mot "SENDER" dans le symbole SDL de création de processus (`Game (SENDER)`) représente le processus qui a envoyé le signal "Newgame".
  - Le mot "SENDER" dans le symbole de sortie SDL (`GameOver TO SENDER`) représente le processus qui a envoyé le signal "GameOver".
  - Dans le symbole de tâche SDL en dessous du symbole de création de processus (`Game (SENDER)`), OFFSPRING et SENDER représentent le même processus.



6. Référez au processus *Game* dans la figure ci-dessus. Lequel des énoncés suivants n'est pas correct?
- a) Un signal "Result" peut être traité dans l'état "Even".
  - b) Un signal "Endgame" peut être traité dans l'état "Even".
  - c) Un signal "Bump" peut être traité dans l'état "Even".
  - d) Un signal "Gameoverack" peut être traité dans l'état "Even".

7. Référez aux deux processus *Monitor* et *Game* sur la page précédente. Nous supposons que le processus *Game* n'a pas encore reçu aucun signal "Bump". Lequel des MSC suivants n'est pas possible ?



8. Parmi les chaînes suivantes, laquelle ne fait pas partie du langage défini par l'expression régulière  $(a*b)^*a*abb(a|b)^*$  ?

- a) ababbab
- b) abbabbabb
- c) abababab
- d) bbabbabba

9. Parmi les expressions régulières suivantes, laquelle définit un langage différent des autres ?

- a)  $((0|1)^*(10|01))^*1|(11|00)$
- b)  $((0|1)^*(10|01))^*|1|1|00$
- c)  $((0|1)|(0|1)^*(10|01))^*1|11|00$
- d)  $00|11|((1|0)^*(01|10))^*1$

10. Référez au programme Lex de la page suivante. Après la compilation et l'exécution du programme avec l'entrée "abbabbabb hello" (note: les guillemets ne font pas partie de l'entrée), laquelle des lignes suivantes ne peut pas apparaître dans la sortie ?

- a) --- Lines : 1
- b) --- Words : 2
- c) --- Identifiers : 2
- d) --- Pattern : 0

```

/* ----- Definitions space ----- */

%option noyywrap
%{
#include <assert.h>
#include <string.h>
#include <stdio.h>
#include <stdlib.h>

    int numOfChars = 0;
    int numOfWords = 0;
    int numOfID_C = 0;          /*the num of C language identifiers*/
    int numOfLines = 0;
    int numOfOcc = 0;          /*num of occurrence of pattern
(a|b)*abb(a|b)***/

%}

WORD          ([a-zA-Z])+
IDENTIFIER    (_|([A-Z])|([a-z]))(_|([0-9])|([a-z])|([A-Z]))*
PATTERN       (a|b)*abb(a|b)*

/* ----- Rules space ----- */
%%
{PATTERN}/\          {
                    ++numOfOcc;
                    ++numOfWords;
                    ++numOfID_C;
                    numOfChars += yyleng;
                    printf("%s\n",yytext);
                    }
{WORD}/(\ |\\n)      {
                    ++numOfWords;
                    ++numOfID_C;
                    numOfChars += yyleng;
                    printf("%s\n",yytext);
                    }
{WORD}
{IDENTIFIER}/\      {
                    ++numOfID_C;
                    numOfChars += yyleng;
                    printf("%s\n",yytext);
                    }
\\n                  ++numOfLines; ++numOfChars;
.                    ++numOfChars;

%%

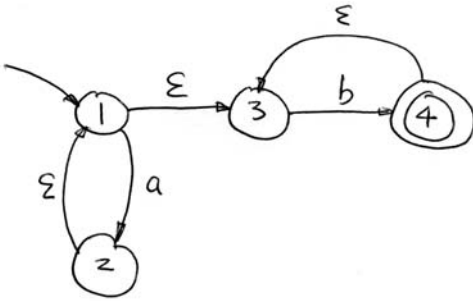
/* ----- User code space ----- */
main()
{
    yylex();
    printf("--- Lines          : %d\n", numOfLines);
    printf("--- Characters     : %d\n", numOfChars);
    printf("--- Words           : %d\n", numOfWords);
    printf("--- Identifiers        : %d\n", numOfID_C);
    printf("--- Pattern           : %d\n", numOfOcc);
}

```

▪ **Questions à réponses brèves**

11. **[4 points]** Suivez l'algorithme 3.3 (voir annexe) pour construire un automate d'états finis non déterministe (NFA) pour l'expression régulière  $(a^*)^*$ .

12. [8 points] Suivez l'algorithme 3.2 (voir annexe) pour convertir l'automate non déterministe suivant en un automate déterministe équivalent (en montrant les différentes étapes de l'algorithme).



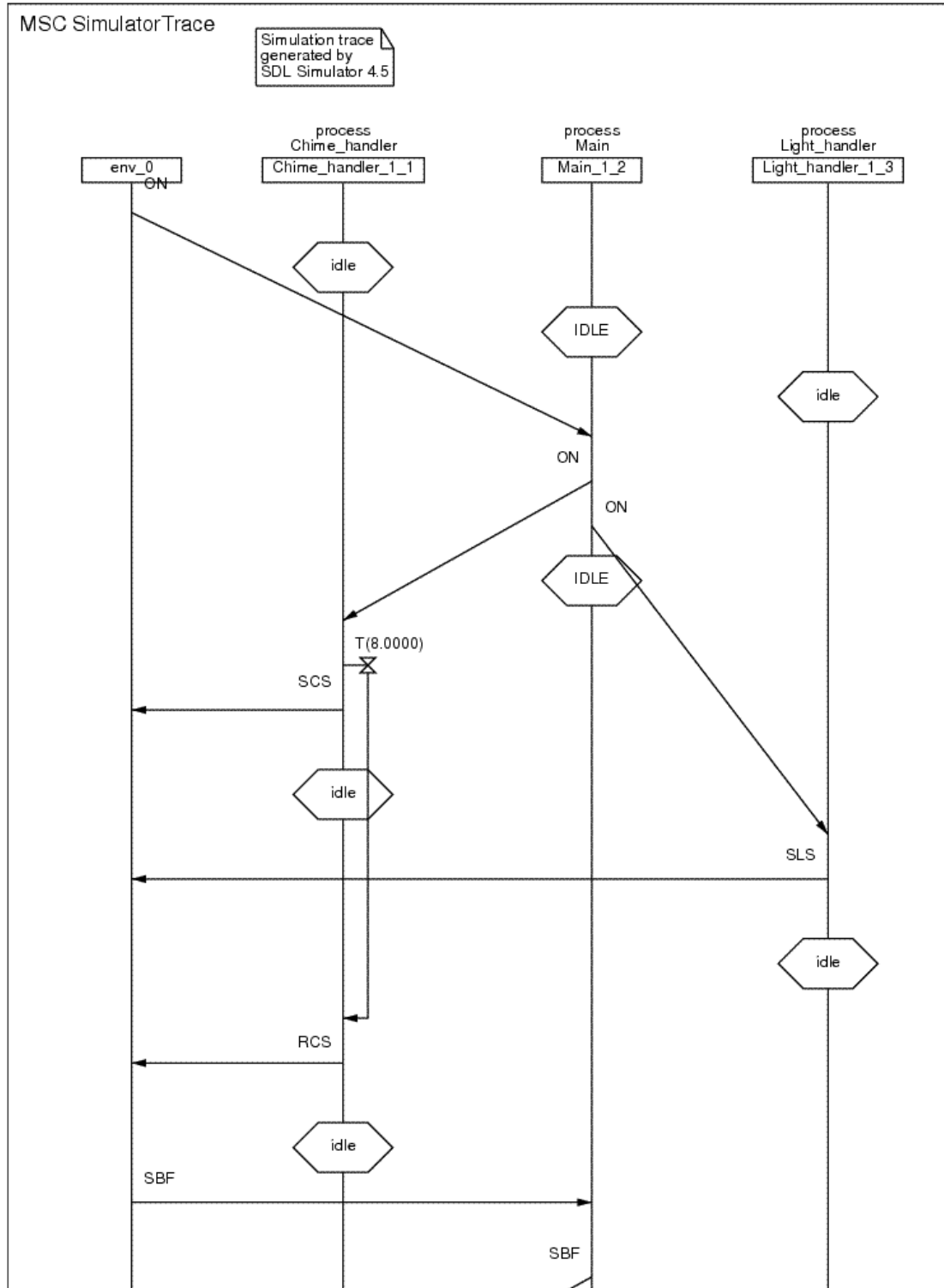
13. [4 points]

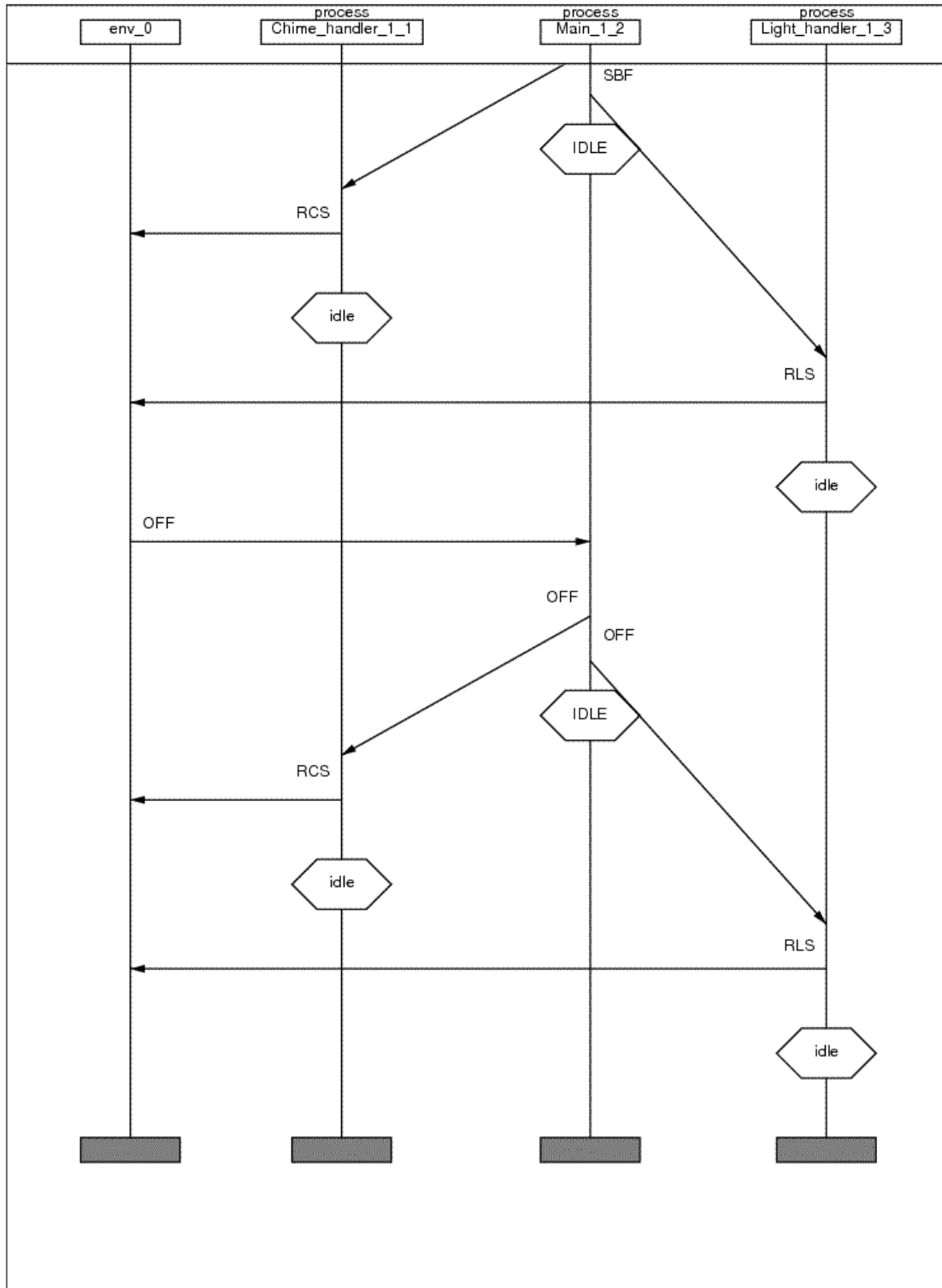
- (a) Écrivez une expression régulière qui définit le langage accepté par l'automate ci-haut.  
(b) Dessinez le diagramme de transitions d'un automate déterministe **minimal** qui accepte le même langage.

14. **[4 points]** Écrivez une définition ou expression régulière qui définit comme langage l'ensemble des chaînes représentant les nombres entiers entre 100 et 300, inclusivement (en notation décimale). On peut supposer que la définition "chiffre = 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 " est déjà donnée.



- **Résolution de problèmes**

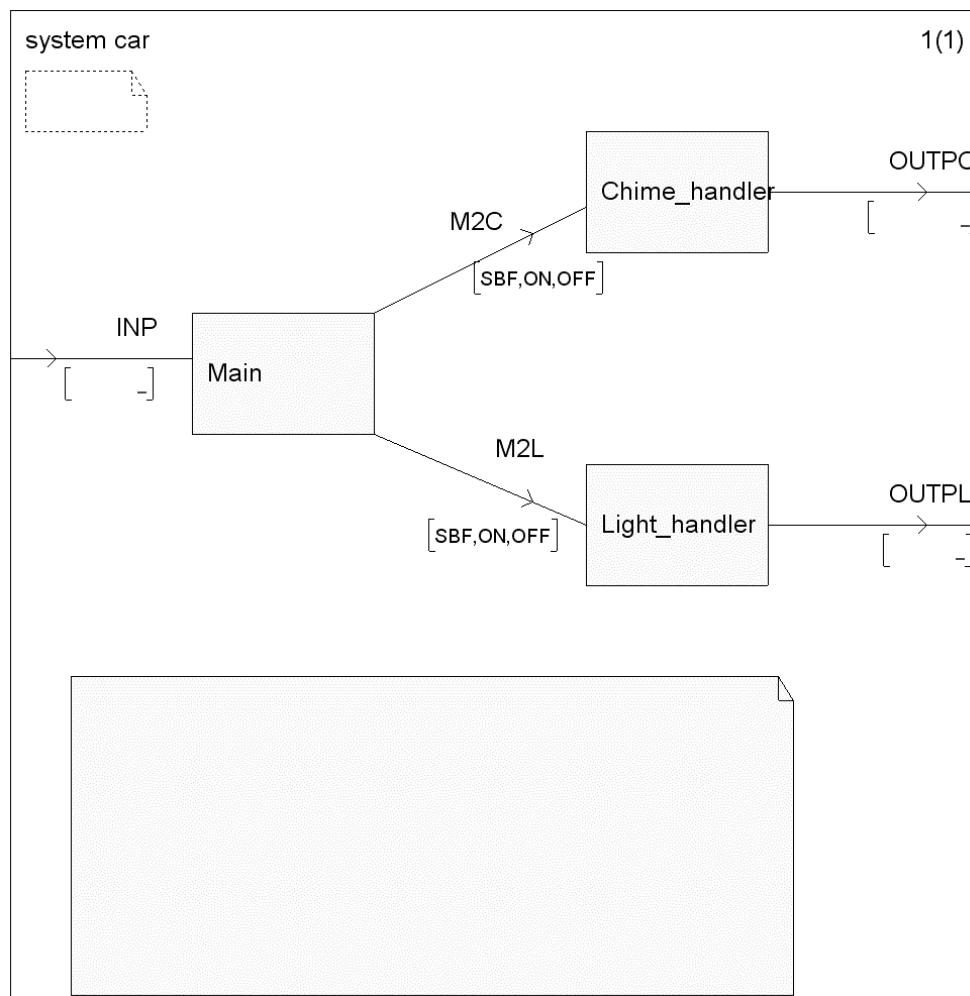




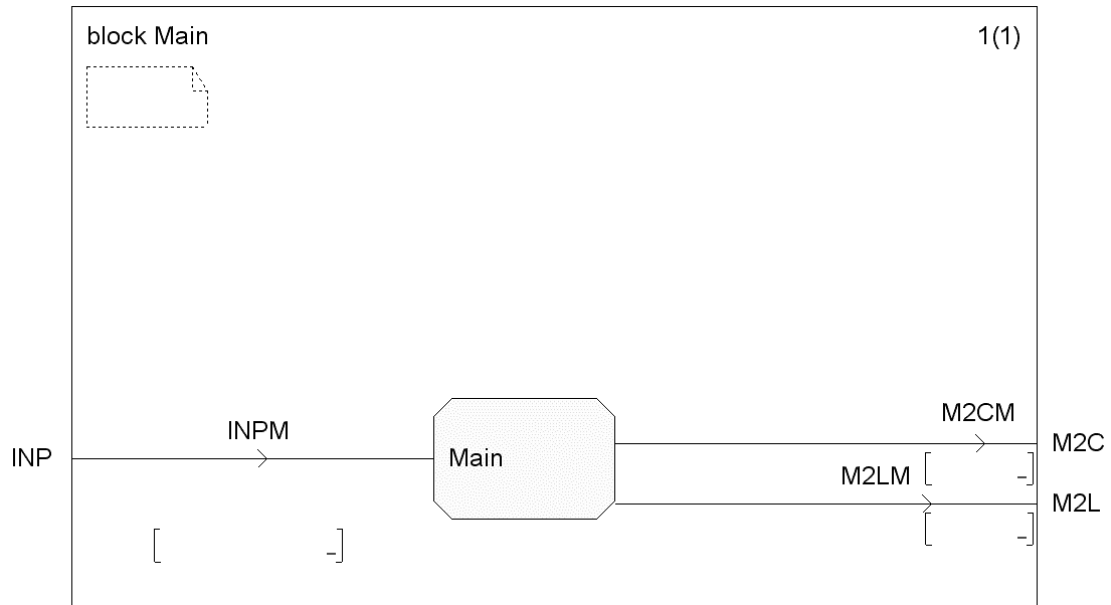
15. Le diagramme ci-haut est un MSC pour le logiciel qui contrôle l'allumage et l'arrêt d'une lumière d'alarme et le signal sonore quand la voiture est utilisée sans que la ceinture de sécurité soit utilisée sur le siège du conducteur. Voici une description de ce système:
- La lumière d'alarme et le signal sonore vous rappellent de mettre la ceinture de sécurité.
  - Si le conducteur ne met pas la ceinture de sécurité avant de mettre la clé de démarrage dans la position ON, alors le signal sonore sonne pendant 8 secondes.
  - Aussi, chaque fois que vous tournez la clé de démarrage à ON, la lumière d'alarme s'allume pour vous rappeler de mettre la ceinture.

- d) Le signal sonore s'arrête aussitôt que la ceinture de sécurité est mise.
- e) L'avertissement par le signal sonore apparaît seulement quand la clé est tournée de OFF à ON (et non quand le conducteur décide de défaire la ceinture pendant que la clé est dans la position ON), et le signal sonore s'arrêtera aussitôt que le conducteur met la clé en position OFF.
- f) Quand la clé de démarrage est dans la position ON, la lumière d'alarme indiquera toujours si la ceinture est mise ou non (i.e. quand la clé est ON, la lumière est allumée si la ceinture est mise, et la lumière est éteinte si la ceinture n'est pas mise).
- g) Il y a des détecteurs dans la voiture qui génèrent les signaux SBF (la ceinture a été mise, ou en anglais: "seat belt has been fastened"), ON (la clé a été tournée de OFF à ON, ou en anglais: "ignition key turned from OFF to ON"), et OFF (la clé a été tournée de ON à OFF, ou en anglais: "ignition turned from ON to OFF"), Ces signaux se propagent des détecteurs (qui sont considérés comme faisant partie de l'environnement du système) vers le système de contrôle.
- h) Le système génère les signaux SCS (commencer le signal sonore, ou en anglais: "set chime sound"), RCS (arrêter le signal sonore, ou en anglais: "reset chime sound"), SLS (allumer la lumière d'alarme, ou en anglais: "set light signal"), and RLS (arrêter la lumière d'alarme, ou en anglais: "reset light signal"). Ces signaux se propagent vers les dispositifs physiques qui génèrent la lumière d'alarme et le signal sonore. Ces dispositifs font partie de l'environnement du système.
- i) Le système consiste de trois blocs SDL: : *Main*, *Chime\_handler*, et *Light\_handler*
- j) Le bloc *Main* contrôle le comportement du mécanisme d'alarme. Il interagit avec l'environnement en acceptant les signaux SBF, ON et OFF.
- k) Le bloc *Chime\_handler* reçoit des instructions du bloc *Main* et contrôle le signal sonore par les signaux SCS et RCS.
- l) Le bloc *Light\_handler* reçoit des instructions du bloc *Main* et contrôle la lumière d'alarme par les signaux SLS et RLS.

15.1. [5 points] Remplissez les blancs dans le diagramme de système suivant:



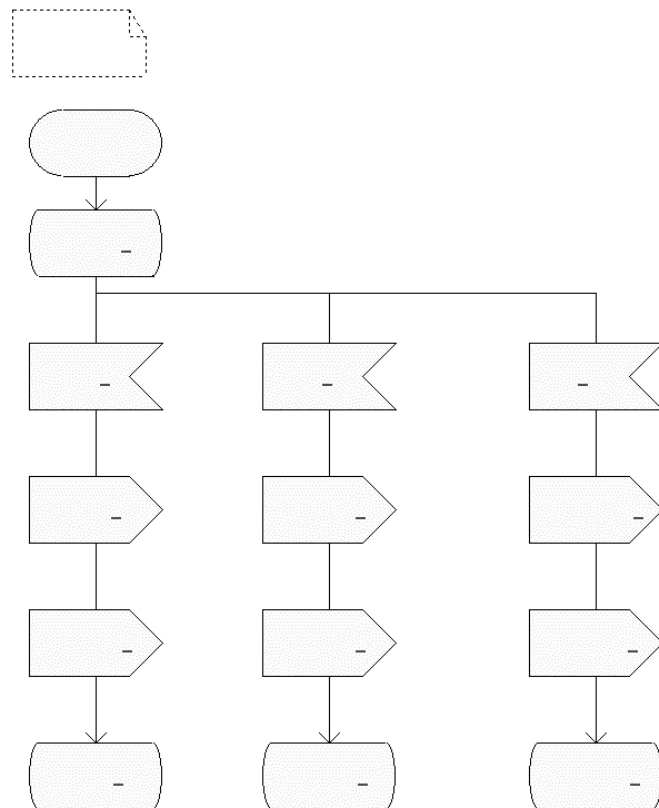
15.2. [3 points] Remplissez les blancs dans le diagramme de bloc suivant:



15.3. [4 points] Remplissez les blancs dans le diagramme de processus suivant:

process Main

1(1)



15.4. [8 points] Remplissez les blancs dans le diagramme de processus suivant:

process Chime\_handler

1(1)

