**SUJET D’EXAMEN GL 40 – 26 juin 2018**

*(Aucun document autorisé)*

**Exercice 1**. Questions de cours.

Considérons les trois langages suivants sur l’alphabet {a, b, c} :

Quel est leur type respectif selon la hiérarchie de Chomsky ? Rappeler l’argument pour le prouver dans chaque cas.

Considérons l’alphabet des parenthèses ouvrante/fermante {(, )}, donner une grammaire pour le langage des parenthèses, i.e. qui contient les systèmes de parenthèses bien formés tels que par exemple le mot (()(()())). Rappeler quel est le type de la grammaire.

**Exercice 2**. Vous devez réaliser un éditeur graphique pour un nouveau langage de programmation que vous avez créé de toute pièce, néanmoins inspiré de C++ ou Java. Pour atteindre ce but, il conviendra de pouvoir analyser la syntaxe d’un programme en édition de manière à mettre en évidence la visualisation à l’écran des structures et des mots clés du langage. Pour cela, nous allons définir une grammaire du langage par étapes. Le langage contient des mots-clés, appelés lexicaux, tels que les termes begin, end, while, for, if, else, etc. Il contient des représentations de nombres entiers, des expressions arithmétiques, booléennes, et des structures de blocs, avec des instructions, séparées en séquence avec le « ; ». Nous n’incorporons pas la notion de procédure ici, le langage est relativement simple et contient le programme suivant à titre d’exemple minimal :

**begin**

**int** a**;**

**int** b**;**

a **:=** 10**;**

b **:=** 1**;**

**while**(a ***>*** 1) **do**

b **:=** a **\*** b**;**

a **:=** a **-** 1**;**

**od**;

**print** b**;**

**end**

1. Considérons quelques mots clés du langage, appelés lexicaux, pris isolément tels que « int » « while », « do », donner un automate d’état fini déterministe et aussi une grammaire qui les reconnait à partir de l’alphabet terminal usuel.
2. Donner une grammaire pour la représentation des nombres entiers, puis l’étendre pour y incorporer les expressions arithmétiques usuelles.
3. Donner une grammaire pour les expressions booléennes manipulées qui peuvent inclure les symboles et opérations « true », « false », ET, OU, NOT, « < », « == », etc.
4. Donner maintenant la grammaire des structures de bloc du langage, et réutilisant les parties précédentes au besoin. Ainsi un bloc débute et termine par les mots clés « begin » et « end », il contient une listes d’instructions, qui peuvent être des définitions (int a ;), des affectations ( :=), des boucles (while), des instructions conditionnelles (if, then, else, endif), des instructions en séquences (;), et pourquoi pas des blocs eux-mêmes. Les instructions manipulent des expressions et lexicaux reconnus aux étapes précédentes. Rappeler quel est le type de cette grammaire.

**Exercice 3.** Questions diverses.

1. Une machine qui exécute des calculs peut-elle penser ? Si oui donner un exemple, sinon, justifier votre réponse.
2. Les opérations d’une machine de Turing sont définies en terme de syntaxe sur un alphabet. Rappeler en quoi consiste une transition élémentaire de la machine de Turing sur son entrée. Faites un croquis.
3. Considérons le langage de programmation de la question 2. Il est constitué des programmes syntaxiquement corrects, i.e. générés par la grammaire réalisée dans l’exercice. Supposons que vous devez réaliser un compilateur de ce langage. Quelle fonction réalise un compilateur. Caractériser ses entrées et sorties en termes de langage et ce qui les différencie.

**Exercice 4.** Une société fait appel à vous pour la réalisation d’une application logicielle d’édition collective de comptes-rendus de réunion (CR). Il s’agit de fournir une application qui permettrait à l’ensemble des participants à une réunion, de rédiger en séance le compte-rendu de celle-ci. Bien entendu, nous ne considérerons dans la suite que l’IHM de l’application.



Les exigences sur l’utilisation et les fonctionnalités de l’application sont les suivantes :

* Chaque participant peut intervenir dans la rédaction. Quand c’est le cas, ses initiales (nom, prénom) s’inscrivent automatiquement dans le CR. Un CR se compose : du thème, d’une date, de la liste des participants présents, de l’ordre du jour, du texte de chaque point de l’ordre du jour et de la date de la prochaine réunion.
* Les fonctions d’éditions habituelles : rédaction, modification, suppression et les fonctions de gestion de CR habituelles : création, ouverture, fermeture, enregistrement.

Votre travail consiste à appliquer la démarche de conception d’IHM proposée en cours : 1) analyse ergonomique, 2) modélisation de l’interface, 3) proposition d’une maquette. Les questions suivantes vous serviront de guide pour construire étape par étape votre solution.

1. En tenant compte de l’expression des besoins formulée ci-dessus, déterminer les postes de travail, le modèle des données et un modèle SADT de la tâche collective, proposer un raffinement en sous-tâches par étapes successives de manière à mettre en évidence la tâche individuelle et son intégration dans le collectif*.*
2. A partir du modèle de tâches précédent proposer un modèle PAC de l’interface utilisateur. Représentez sur votre modèle PAC les interactions entre objets graphiques lors de l’édition collective d’un champ de texte de l’ordre du jour. Expliciter les abstractions et les vues, notamment, dessinez une maquette de votre solution (ensemble des vues graphiques).