programmation examen septembre

Nous nous intéressons aux transitions d'un automate à états finis. Une transition est un triplet *(état de départ, symbole, état d'arrivée).* Certaines transitions sont appelées *ε-transitions* et sont représentées par un symbole qui est le caractère de code 0.

```
class Transition{
   private :
      int etatDepart, etatArrivee;
      char symbole ;
   public :
      Transition( int ed, char s, int ea);
      ~Transition() ;
      bool operator==(const Transition& t);
      bool estEpsilon() ;
      int getDepart(); int getArrivee() ;
      char getSymbole();

} ;
```

Question 1. (3 points)

Ecrire les définitions du constructeur, du destructeur et des fonctions membres de la classe *Transition*. La fonction *estEpsilon* renvoie vrai si la transition est une *ε-transition*.

Un ensemble de transitions est représenté par une instance de la classe *Transitions.* Chaque transition est représentée une seule fois dans le vecteur.

```
class Transitions{
 private :
   vector <Transition> lesTransitions ;
 public :
   Transitions();
   ~Transitions() ;
   bool existeEpsilontransition(int & ed, int &ea);
   bool existeTransition(const Transition & t);

   void ajoutTransition(const Transition &t) ;
   void suppTransition(const Transition &t) ;
   void supprimeEpsilonTransition() ;
} ;
```

Question 2. (2 points)

Ecrire les définitions du constructeur et du destructeur de la classe *Transitions*.

Question 3. (5 points)

Ecrire la définition de la fonction *existeEpsilonTransition(int & ed, int &ea)*, qui renvoie vrai si le vecteur contient au moins une *ε-transition* (dans ce cas *ed* contient l'état de départ et *ea* l'état d'arrivée de la première *ε-transition* trouvée) et faux sinon (*ed* et *ea* sont indéterminés). Ecrire la définition de la fonction *existeTransition(const Transition & t),* qui renvoie vrai si le vecteur contient la transition *t* et faux sinon.

<u>Question 4.</u> (5 points)

Ecrire la définition des fonctions *ajoutTransition(const Transition &t),* et *supprimeTransition(const Transition &t)* qui ajoute et supprime une transition du vecteur.

<u>Question 5.</u>(5 points)

Ecrire la définition de la fonction *supprimeEpsilonTransition()*, qui supprime les *ε-transitions* d'une instance de la classe *Transitions*, sans modifier le fonctionnement de l'automate.

On appliquera l'algorithme suivant : pour éliminer la transition *(ed, ε, ea),* on cherche toutes les transitions *(ex, x, ed)* qui arrivent en *ed*, et on ajoute pour chacune la transition *(ex, x, ea)*. On veillera à ne pas ajouter des *ε-transitions* qui bouclent sur un état.