Samuel Goubeau

Projet : Théorie des Langages

**Exercice 2 :**



* Il faut écrire une fonction par état, par conséquent il faut dans ce cas écrire N fonctions.
* Si l’état i est final, alors la fonction reconnaitRec\_i("") retourne vrai. Cependant dans l’autre cas, elle retournera faux.
* Il faut appeler la fonction correspondant à la transition i--[**c**]-->j, soit la fonction reconnaitRec\_ j(). Le paramètre de cette dernière sera alors le mot **mot** auquel on aura retiré la première lettre **c**.

1. Automate Ar :

Une image contenant texte, tableau blanc, signe

Description générée automatiquement

**Exercice 3 :**

1. Pour cette particularité, il va falloir faire de la même manière que le calcul de la hauteur d’un arbre Binaire. Si le mot respect la condition ‘a’, alors j’appellerai à la fois reconnaitRec\_1( ) et reconnaitRec\_2( ). Sachant que ces 2 fonctions vont renvoyer un booléen, alors il me suffira de poser la condition || (soit OU) entre ces 2 valeurs de retour pour avoir celle de reconnaitRec\_0( ) :

|  |
| --- |
| **return ( reconnaitRec\_1(this.mBis) || reconnaitRec\_2(this.mBis) ) ;**  *(où this.mBis est le mot sans sa première lettre)* |

1. Je vais effectuer la même manipulation qu’avec reconnaitRec\_0( ). Ce qui veut dire que si le mot respect l’une des conditions de transition, alors cette fonction appellera non seulement reconnaitRec\_3( ) (pour ‘a’), reconnaitRec\_ 1( ) (pour ‘c’), ou reconnaitRec\_ 4( ) (pour ‘c’), mais aussi reconnaitRec\_2( ) avec le mot complet :

|  |
| --- |
| Cas ‘a’ :  **return ( reconnaitRec\_3(this.mBis) || reconnaitRec\_2(this.m) ) ;**  *(où this.mBis est le mot sans sa première lettre*  *et this.m le mot complet)*  Cas ‘b’ :  **return ( reconnaitRec\_1(this.mBis) || reconnaitRec\_2(this.m) ) ;**  *(où this.mBis est le mot sans sa première lettre*  *et this.m le mot complet)*  Cas ‘c’ :  **return ( reconnaitRec\_4(this.mBis) || reconnaitRec\_2(this.m) ) ;**  *(où this.mBis est le mot sans sa première lettre*  *et this.m le mot complet)* |