

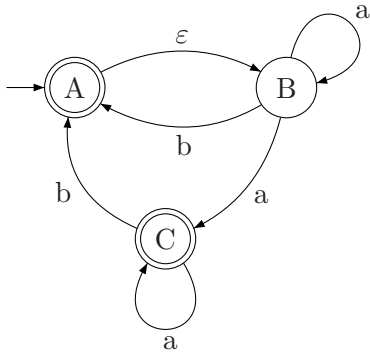
Bases Formelles du T.A.L. - LI063(6)  
Deuxième Session - Devoir sur table  
Documents non autorisés

Durée 1h50

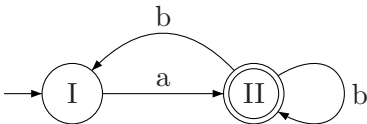
23 Juin 2010

## 1 Automates à États Finis

Soit l'automate suivant :



1. Donnez l'ensemble des raisons pour lesquelles cet automate n'est pas déterministe.
2. Déterminez cet automate. Détaillez l'ensemble de vos étapes.
3. Appliquez l'algorithme de McNaughton & Yamada pour obtenir l'expression rationnelle qui décrit le même langage que cet automate.
4. Construisez l'automate qui reconnaît l'intersection du langage décrit par l'automate ci-dessus et l'automate suivant :



5. Construisez la grammaire de réécriture équivalente à l'automate déterminisé de la question 2.
6. Donnez la dérivation de deux mots reconnus par cette grammaire.

## 2 Grammaires

1. On s'intéresse (du point de vue théorique) à l'implémentation informatique d'une grammaire hors-contexte.
  - (a) Proposez une structure de données qui va représenter une grammaire. Par exemple, dans un pseudo-code inspiré de Java, vous décrirez les attributs d'une classe `Grammaire` qui permettent de capturer toutes les propriétés d'une grammaire. Vous apporterez une attention particulière à la structure de données que vous utiliserez pour représenter les règles de réécriture ainsi qu'à commenter votre structure de données pour la rendre claire.
  - (b) Écrivez le pseudo-code d'une méthode qui permet de rajouter une règle de réécriture valide dans la grammaire.
  - (c) Écrivez le pseudo-code d'une méthode utilisant l'algorithme de suppression des  $\varepsilon$ . La méthode devra fournir en sortie un nouvel objet `Grammaire` représentant la grammaire débarrassée de ses  $\varepsilon$ -productions.
  - (d) Dans cette question on suppose que les grammaires décrites sont des grammaires **régulières** sans  $\varepsilon$ . Donnez le pseudo-code de l'algorithme de reconnaissance d'un mot par la grammaire.