

Théorie des langages et automates

TP 3

Automates finis non déterministes avec ϵ -transitions

Dans ce TP, nous allons implémenter un programme Java pour définir et exécuter les automates finis non déterministes avec ϵ -transitions. Pour définir un automate, il est suffisant de remplir la table de transitions et le programme peut la suivre pour voir si un mot entré peut être accepté par l'automate.

Les classes suivantes sont définies dans le package `tla.core` servant à des éléments de base d'automates :

- `State` : l'état ;
- `StateSet` : un ensemble d'états, étendue de la classe `Set<State>` ;
- `Symbol` : le symbol ;
- `TransitionTable` : la table de transitions, étendue de la classe `TreeMap<State, <Symbol, StateSet>>` ;
- `Word` : le mot, étendue de la classe `ArrayList<Symbol>`.

Toutes les méthodes marquées TODO dans la classe `TransitionTable` doivent être déjà implémentées dans le TP précédent.

Les classes suivantes sont définies dans le package `tla.fsa` :

- `Automaton` : la classe abstraite la plus générale pour définir les automates ;
- `AutomatonNfaEpsilon` : la classe abstraite la plus générale pour définir les automates finis déterministes avec ϵ -transitions.

Les méthodes `Delta(w)`, `accepts(w)`, `closure(q)` et `closure(q, E)` dans la classe `AutomatonNfaEpsilon` doivent être implémentées.

La classe `SimpleNFAe` définit un simple automate fini non déterministe avec ϵ -transitions pour un test et la classe `Main` permet de lancer le programme.