

Synthèse TP Automates

Ce TP présente différents automates à réaliser et le but est de vérifier si selon la chaîne de caractères passée si l'automate l'accepte ou non.

Le programme commence d'abord le menu de l'application (voir image ci-dessous) où l'utilisateur doit choisir l'automate dans lequel il va envoyer la chaîne de caractères à analyser.

```
----- My TP menu -----
1. Smiley (recognize smileys)
2. HH:MM (recognize a well-formed hour)
3. DD/MM/YYYY (recognize a well formed date)
4. Email address (recognize a well-formed email address)
5. Polynomial (recognize a well-formed polynomial under the form  $aX^2+bX+c$  with a, b and c between 1 and 25)
9. Stop
Your choice (1-9) ?
I will then ask you for the channel to analyze, thank you.
```

Une fois l'automate choisi, l'utilisateur va devoir indiquer la chaîne de caractère à analyser puis l'application lancera la classe demandée qui va créer l'automate puis assurer la vérification de la chaîne de caractères.

Si la chaîne de caractères correspond bien à l'automate, il sera indiqué que la chaîne de caractères est correcte et dans le cas contraire, qu'elle est incorrecte.

```
}
Please enter the character string to analyze, thank you.
{-}
The parsed character string is correct.
```

Quand une classe d'automate est appelée, celle-ci va créer les différents états par lesquels on peut passer, elle va créer des transitions entre les états et définir un état de départ et un voire des états de fin.

```
public Smiley(){
    automate = new Automate( nom: "Smiley");
    State E0 = new State( newnom: "E0");
    State A1 = new State( newnom: "A1");
    State A2 = new State( newnom: "A2");
    State B3 = new State( newnom: "B3");
    State C4 = new State( newnom: "C4");
    automate.addTransition(E0, Caractere: ':', A1);
    automate.addTransition(E0, Caractere: '-', A2);
    automate.addTransition(A1, Caractere: '-', B3);
    automate.addTransition(A1, Caractere: '=', B3);
    automate.addTransition(A2, Caractere: '-', B3);
    automate.addTransition(B3, Caractere: '(', C4);
    automate.addTransition(B3, Caractere: ')', C4);
    automate.finalstate(C4);
    automate.statel(E0);
}
```

Ensuite, la méthode de vérification sera appelée et va vérifier chaque caractère pour savoir s'il devrait être là ou non. Dans le cas où un caractère n'est pas bon, l'exception « AutomateException » sera levée et la chaîne de caractères sera considérée comme incorrecte.

Le programme continue jusqu'à ce que l'utilisateur demande le contraire (choix 9 dans le menu).

En conclusion, ce programme permet de se rendre compte de l'importance des automates de nos applications d'aujourd'hui et de commencer à en créer pour que l'on puisse potentiellement créer des logiciels plus tard.