

Théorie des langages (INF203), Télécom ParisTech

TP

Bogdan Cautis (bogdan.cautis@telecom-paristech.fr)
Pierre Senellart (pierre.senellart@telecom-paristech.fr)

25 novembre 2009

1 Expressions rationnelles en Java

Le but de cet exercice est de réaliser une petite application Java qui extrait les URL et les adresses emails depuis une page Web donnée, en utilisant des expressions rationnelles.

Les expressions rationnelles peuvent être manipulées en Java grâce aux classes du paquetage `java.util.regex`. L'utilisation de base est la suivante :

```
Pattern expression=Pattern.compile("a*b+");
Matcher m=p.matcher("aaabbbbbbb");
if(m.matches()) {
    ...
}
```

La syntaxe des expressions rationnelles comprise par Java est donnée dans la documentation en ligne de la classe `Pattern`.

1. Dans un programme Java élémentaire, créer un objet de type `Pattern` correspondant au langage des URL, et le tester.
2. Faire un programme Java aussi simple que possible, qui extrait toutes les URL présentes à l'intérieur d'une page Web donnée (p.ex., `http://www.slashdot.org/`). On pourra utiliser les éléments suivants, et se référer à la documentation en ligne de l'API Java :
 - On peut créer un objet de type `java.net.URL` avec un constructeur prenant en entrée une URL sous forme de chaîne de caractères.
 - On peut récupérer un `java.io.InputStream` contenant le contenu d'une page Web à partir d'un objet `url` de type `java.net.URL` avec l'appel `url.openConnection().getInputStream()`.
 - On peut créer un objet `java.util.Scanner` à partir d'un `java.io.InputStream is` avec l'appel `new Scanner(is, "UTF-8")`.
 - On peut parcourir l'ensemble des correspondances d'un `java.util.regex.Pattern p` à l'intérieur d'une page parcourue par un `java.util.Scanner scanner` avec :

```
while( (match=scanner.findWithinHorizon(p,0))!=null) { ... }
```

où `match` est de type `java.lang.String`.
3. Faire en sorte de ne pas afficher deux fois une même URL.
4. Faire la même chose pour les adresses emails.

2 JavaCC

JavaCC est un programme permettant de créer automatiquement un analyseur syntaxique pour un langage décrit sous forme de grammaire hors-contexte. Le principe est le suivant :

- On crée un fichier `MaClasse.jj` décrivant le langage et le traitement à effectuer dessus.
- On appelle `javacc`¹ sur ce fichier pour créer un fichier `MaClasse.java` avec un programme Java d'analyse du langage, ainsi que divers fichiers annexes.
- On appelle `javac` sur ce fichier pour compiler le programme Java.
- On lance la classe Java ainsi obtenue avec `java MaClasse`.

Un fichier `.jj` de description de langage comprend des règles de production, similaires aux règles de production des grammaires hors contexte, entrecoupées d'instructions Java qui indiquent le comportement à adopter quand telle production de la grammaire est utilisée.

1. Télécharger depuis <http://pierre.senellart.com/enseignement/2009-2010/inf203/> le fichier `JavaCC Calculatrice.jj` qui comprend une description pour JavaCC du langage des expressions arithmétiques avec les opérateurs `+` et `*`. Étudier ce fichier. Le transformer en classe Java avec la procédure ci-dessus et tester le programme (il faut démarrer le programme à l'intérieur d'un terminal, taper une expression arithmétique, puis un retour à la ligne et `CTRL+D`). On pourra éventuellement consulter la documentation en ligne de JavaCC sur <https://javacc.dev.java.net/doc>.
2. Ajouter les opérateurs `-` et `/` au langage. Tester.
3. Ajouter la possibilité d'avoir des expressions parenthésées. Tester.
4. Ajouter la possibilité d'avoir des nombres négatifs. Tester.
5. Ajouter la possibilité d'avoir des nombres décimaux. Tester.

¹Sur les machines de la salle de TP, vous appellerez `javacc` avec `~/senellar/bin/javacc`.