

Travaux Dirigés de Compilation n°6

ESIPE - Filière informatique

Analyse syntaxique avec antlr

L'objectif de ce TD est d'apprendre à utiliser `antlr` pour l'analyse syntaxique.

► **Exercice 1.** *Ajouter des règles lexicales à la grammaire `xml_tags.g4`, sans modifier les règles syntaxiques, de façon à obtenir un analyseur syntaxique capable de reconnaître les balises XML dans un fichier d'entrée tel que `weblogic.log` ou `weblogic-stack-trace.log`. On demande que l'analyseur ne provoque aucune erreur de reconnaissance de lexèmes (token recognition error) à l'exécution.*

```
/**
 * parsing xml tags
 */
grammar xml_tags;
file
    : (NO_XML | NAME | VALUE | '/' | '=' | ':' | '>'
      | xml_tag { System.out.println(" tag parsed "); }
      | '</' | '<' )+ ;
xml_tag
    : '<' name attribute* '/'? '>' | '</' name '>' ;
attribute
    : name '=' VALUE ;
name
    : (NAME ':')? NAME ;
```

Indication : Attention à ce que les lexèmes des catégories `NAME` et `VALUE` soient prioritaires sur ceux de la catégorie `NO_XML` quand on est dans une balise. Ce n'est pas facile parce que l'analyseur lexical ne sait pas si on est à l'intérieur ou à l'extérieur des balises.

► **Exercice 2.** *Ajouter des règles syntaxiques à la grammaire `embedded_xml.g4`, sans modifier les règles lexicales, de façon à obtenir un analyseur syntaxique capable de reconnaître un document XML tel que `weblogic-stack-trace.log`. On demande que l'analyseur ne provoque aucune erreur de reconnaissance de lexèmes (token recognition error) à l'exécution.*

```
/**
 * parsing an xml file
 */
grammar embedded_xml;
```

```

file      : xml_prolog ? element ;
xml_prolog: '<?xml' attribute * '?>' ;
element   :
(...)
WS        : [ \t\r\n]+ -> skip ;
QNAME     : (ID ':' ID)? ID ;
ATTRIB_VAL: '"' ~'"'* '"' | '\'' ~'\''* '\'' ;
NO_LESS   : ~'<' ;
fragment
ID        : [\p{Alphabetic}_] [-\p{Alphabetic}_0-9.]* ;

```