

Esercitazione Parser Generators

Fondamenti di informatica II (Modelli)

28-05-2021

Scopo di questa esercitazione è prendere confidenza con i tool Bison ed Antlr per la generazione del codice sorgente di parser, in vari linguaggi di programmazione.

L'esercitazione può essere svolta in Java e Python con Antlr, e in C e Java con Bison; può essere ripetuta, senza sostanziali modifiche, per generare parser scritti in altri linguaggi (Bison: C++; Antlr: C#, JavaScript, C++, Swift, Go, PHP).

Nella cartella *documenti* c'è un file chiamato *Installazione.pdf* che spiega come preparare il pc ad usare gli strumenti previsti in questa esercitazione. Si assumerà che il prerequisito sia soddisfatto (lo era usando i PC del laboratorio).

Task 1. Riprodurre l'esempio riportato nei file di istruzioni

In questa attività preliminare è richiesto di leggere e capire gli esempi riportati nei file di istruzioni. Eseguire quindi alcuni esperimenti, tramite il file *test*, modificando più volte la stringa su cui effettuare il parsing, prevedendo, nei vari esperimenti, stringhe del linguaggio e non.

- Antlr:
 1. Leggere e capire il file *Antlr_Istruzioni*
 2. Aprire il file **test** dove inserire la stringa su cui effettuare il parsing
 3. Compilare ed eseguire (come descritto nelle istruzioni)
- Bison:
 1. Leggere e capire il file *BisonC_Istruzioni* (*BisonJavaIstruzioni*)
 2. Aprire il file **test** dove inserire la stringa su cui effettuare il parsing
 3. Compilare ed eseguire (come descritto nelle istruzioni)

Task 2. Parentesi bilanciate

Seguendo i passi all'interno dei file di istruzioni, fare un nuovo esperimento con il linguaggio delle *Parentesi Bilanciate*, generato dalla seguente grammatica:

```
1 | S → (S)
2 | S → SS
3 | S → ()
```

- Antlr:
 1. Inserire nel file **BalPar.g4** le regole per il parser e per il lexer
 2. Modificare il file **Main.java** (*main.py*) per effettuare l'esperimento di parsing
 3. Usare il file **test** inserendo la stringa su cui effettuare il parsing
 4. Compilare ed eseguire
- Bison:
 1. Inserire nel file **BalPar.l** le regole per il lexer
 2. Inserire nel file **BalPar.y** le regole per il parser
 3. Usare il file **test** inserendo la stringa su cui effettuare il parsing
 4. Compilare ed eseguire

Task 3. Alberi binari

Ripetere il task 2 questa volta utilizzando il linguaggio degli *Alberi Binari*, generato dalla seguente grammatica:

```
1 | T → ()
2 | T → (key T T)
```

Dove *key* è un numero intero qualsiasi. Come nell'esercizio precedente modificare i file presenti nella distribuzione ed eseguire i test.

N.B. Questo è un esempio *importante*, per il suo uso nella (de)serializzazione degli alberi binari.

Task 4. (Per casa)

Stringhe Palindrome

Effettuare un esperimento di parsing utilizzando il linguaggio delle *Stringhe Palindrome*. Data la seguente grammatica:

1	$S \rightarrow aSa$
2	$S \rightarrow bSb$
3	$S \rightarrow aa$
4	$S \rightarrow bb$
5	$S \rightarrow a$
6	$S \rightarrow b$

Modificare i file presenti nella distribuzione ed eseguire i test. Si rileveranno problemi (perlopiù con Bison) dovuti al non determinismo del linguaggio.

3-CNF

Effettuare un esperimento di parsing utilizzando il linguaggio delle *formule 3-CNF*, generato dalla seguente grammatica:

- 1 $F \rightarrow F \wedge C$
- 2 $F \rightarrow C$
- 3 $C \rightarrow (L \vee L \vee L)$
- 4 $L \rightarrow \text{id}$
- 5 $L \rightarrow \neg \text{id}$

Modificare i file presenti nella distribuzione ed eseguire i test.

Questa esercitazione è stata organizzata con il valido supporto del tutor Valerio Coretti.