

UNIVERSITÀ DEGLI STUDI DI CATANIA

CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA

**Analisi Top-Down LL(1) di grammatiche context-free**

Relazione di Linguaggi e Traduttori

**Docente: Studenti:**

Prof.ssa Vincenza Carchiolo Floriana Leone O55000246

Giuseppe Mastrosimone O55000262

Anno Accademico 2015/2016

**Introduzione**

Lo scopo di questo progetto è quello di implementare il funzionamento di un parser top-down di tipo LL(1), cioè di effettuare l’analisi sintattica di grammatiche non contestuali (context-free) andando a considerare un solo carattere di lookahead.

In particolare, l’applicazione sviluppata è in grado di verificare se una grammatica è context-free e in caso affermativo di procedere con l’analisi LL(1). Il parser implementato è capace di analizzare le grammatiche inserite, verificando di volta in volta se esse soddisfano i requisiti di un parser LL(1) e, in tal caso, di procedere alla generazione delle relative tabelle di parsing.

**Strumenti utilizzati**

L’applicativo è stato implementato utilizzando il linguaggio OO Java. In particolare, per lo sviluppo del codice sono stati usati:

* Eclipse Luna 4.4, come IDE per lo sviluppo del codice Java;
* Libreria Batik, integrata alla piattaforma di sviluppo per la realizzazione delle tabelle di parsing con grafica SVG di tipo XML-like.

**Descrizione del codice**

**Premessa sulla definizione delle grammatiche**

Al fine di effettuare l’analisi LL(1) è necessario per prima cosa definire le grammatiche alle quali è possibile applicarla. Per questo motivo l’utente ha la possibilità di inserire da tastiera i simboli terminali e non terminali e le relative produzioni in modo da definire la grammatica desiderata. In particolare, il formato della grammatica deve essere il seguente: A->aBc, dove la parte sinistra deve contenere solo un terminale (indicato con la lettera maiuscola) e quella destra può contenere invece un numero variabile di terminali e non terminali. Seguendo questo formato viene garantita che la grammatica inserita sia di tipo non contestuale, ma non si ha la certezza che sia di tipo LL(1). Non terminali e terminali sono differenziati dal fatto che solamente i primi possono contenere lettere maiuscole, mentre i secondi possono essere espressi da qualsiasi carattere purché non contenga lettere maiuscole.

**File delle grammatiche**

Una volta definito il formato delle grammatiche supportato dal programma sviluppato, è bene sapere come è possibile leggerlo o definirlo.

L’applicazione permette due modalità di definizione della grammatica di cui si vuole effettuare l’analisi LL(1): lettura della grammatica da file o scrittura della grammatica su file. Il tutto è reso possibile grazie a un menù ben definito che consente all’utente di scegliere l’opzione di sua preferenza.

In particolare, se l’utente sceglie di definire la grammatica inserendo i valori dei terminali, dei non terminali e delle produzioni da tastiera il tutto, oltre ad essere stampato a schermo, viene salvato su 2 file, ovvero: nome\_file.txt dal quale è possibile visualizzare direttamente la grammatica inserita e nome\_file (senza estensione, utilizzato dal programma per analizzare la grammatica in qualunque momento). Questo secondo file è utile quindi nel caso in cui l’utente decida di effettuare l’analisi LL(1) in un secondo momento, cioè se l’utente vuole effettuare l’analisi di una grammatica già precedentemente definita in tal modo.

E’ consentita, inoltre, un’altra modalità di lettura da file della grammatica definita, in questo caso, esternamente dall’utente stesso. Ciò vuol dire che l’utente può anche scegliere di scrivere direttamente su un file esterno, in formato .txt, e poi utilizzare il programma per caricare la grammatica inserita nel file, seguendo le regole di definizione delle grammatiche supportate dal programma.

**Menù**

Come già anticipato, quando il programma viene mandato in esecuzione viene avviato un menù.



**Esempio 1: Inserimento di una grammatica da tastiera:**

Passo 1)

Digitando ‘2’ verrà visualizzata la seguente richiesta di inserimento:



Come è indicato, bisogna digitare il numero di terminali che deve contenere la grammatica da inserire.

Passo 2)

Questo passo verrà ripetuto un numero di volte uguale al valore scelto precedentemente:



Una volta completato il passo 2, tutti i terminali digitati saranno inseriti nella nuova grammatica. I passi successivi, allo stesso modo, permetteranno di inserire i restanti elementi della quadrupla che definisce una grammatica.

Passo 3)



Passo 4)



Passo 5)



In questo passo verrà scelto il simbolo iniziale, che verrà quindi inserito nell’insieme dei non terminali.

Passo 6)



Passo 7)



Qui viene visualizzato a schermo l’insieme dei non terminali che sono stati inseriti in modo da agevolare l’utente nell’inserimento della parte sinistra delle produzioni.

Passo 8)



Passo 9)

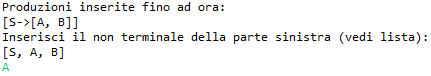


Come per la parte sinistra, viene visualizzato a schermo l’insieme dei simboli che sono stati inseriti in modo da agevolare l’utente nell’inserimento della parte destra delle produzioni. Dopo essere stato inserito l’ultimo simbolo apparirà il seguente messaggio:



Passo 10)

Una volta inserita una produzione, se ne devono essere aggiunte altre, apparirà una schermata di questo tipo:



Vengono cioè visualizzate le produzioni che sono state inserite fino ad ora, così da dare all’utente maggior sicurezza nella prosecuzione dell’esecuzione del programma. Quindi il programma prosegue ripetendo i passi 8 – 10 fino al completamento dell’inserimento delle produzioni. Dopodiché verrà mostrato il messaggio:

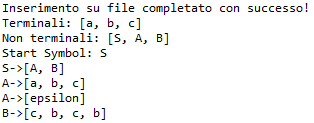


Passo 11)

Adesso deve essere inserito il nome del file sul quale sarà salvata la grammatica appena inserita, in modo da poterla visualizzare ed analizzare in un secondo momento.



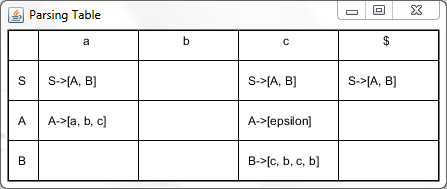
Una volta digitato il nome del file, esso verrà creato e il suo contenuto verrà stampata a schermo:



Inoltre, il programma proseguirà andando ad effettuare l’analisi LL(1) della grammatica, mostrando gli insieme dei first, follow e predict:



Infine, se la grammatica è LL(1), viene generata la tabella di parsing:



A questo punto il programma torna al menù iniziale, permettendo di creare una nuova grammatica da tastiera o di sceglierne da un file già presente.

**Esempio 2: Caricamento di una grammatica da file:**