Università degli studi di Bergamo

Facoltà di Ingegneria

Corso di Laurea Magistrale in Ingegneria Informatica

Linguaggi formali e compilatori

(COD.CORSO: 38070)

Anno accademico:	**2018/2019**
Settore scientifico-disciplinare:	**ING-INF/05**
Dipartimento:	**Ingegneria gestionale, dell'informazione e della produzione**

Titolo:

Riconoscitore di grammatiche LR(1)

Autori:

Luca Filice	**Matr: 1026838**
Matteo Gusmini	**Matr: 1035426**
Davide Presciani	**Matr: 1035189**

Indice

Introduzione

In informatica e in linguistica, una grammatica libera dal contesto è una grammatica formale in cui ogni regola sintattica è espressa sotto forma di derivazione di un simbolo a sinistra a partire da uno o più simboli a destra. Ciò può essere espresso con due simbolismi equivalenti:

1) S := α
2) S → α

dove S è un simbolo *non terminale* e α è una sequenza di simboli *terminali* e *non terminali*.

L'espressione "libera dal contesto" si riferisce al fatto che il simbolo *non terminale* S può sempre essere sostituito da α, indipendentemente dai simboli che lo precedono o lo seguono. Un linguaggio formale si dice libero dal contesto se esiste una grammatica libera dal contesto che lo genera.

Le grammatiche libere dal contesto sono abbastanza potenti da descrivere la sintassi della maggior parte dei linguaggi di programmazione; al tempo stesso, sono abbastanza semplici da consentire un parsing molto efficiente.

La notazione formale di Backus-Naur (BNF) è la sintassi più comunemente usata per descrivere grammatiche libere dal contesto.

Un parser LR è un parser di tipo Bottom-up per grammatiche libere da contesto, usate molto di frequente nei compilatori dei linguaggi di programmazione. Un Parser LR legge il proprio input partendo da sinistra verso destra, producendo una derivazione destra. A volte questo parser viene anche indicato col nome “Parser LR(k)” dove k si riferisce al numero di simboli letti (ma non "consumati") utilizzati per prendere le decisioni di parsing.

Funzionamento

Questo riconoscitore di grammatiche LR(1) permette di verificare se una data grammatica in ingresso è o meno di tipo LR(1).

Inoltre, questo riconoscitore genera un elenco degli stati, ognuno con relative regole di core e completamento complete di look-ahead, e una lista delle transizioni tra i vari stati.

I caratteri accettati dal riconoscitore, con le rispettive categorie, sono i seguenti:

Simbolo	Categoria	Caratteri
SZ	Stato iniziale	S0
EQ	Simboli di derivazione	-> | :=
NT	Non terminali	A … Z
CT	Caratteri terminali	a … z | 0 … 9 | + | - | * | /
TER	Terminatori	/swa | /cjswa
SC	Fine regola	;

Questi caratteri sono utilizzati per comporre i due diversi tipi di regole di produzione riconosciuti dal parser:

- una prima regola *pr*, che ha come elemento di sinistra il *non terminale* S0

  SZ EQ NT TER SC

- tutte le altre regole di produzione *ar*, che formano il resto della grammatica

  NT EQ (NT|CT)* SC

Per procedere all'analisi è sufficiente creare un file di tipo txt denominato "input.txt" nella cartella dove è presente il file *jar* "riconoscitoreLR1.jar"; il file deve contenere una grammatica che segua nella forma la struttura definita in precedenza, come nell'esempio qui riportato.

```
S0->S/cjswa;
S->P;
P->L;
L->IL;
L->;
I->vit;
I->iei;
I->iefi;
I->wioLc;
```

Definito tale file è sufficiente aprire il prompt dei comandi nella cartella contenente il file *jar* "riconoscitoreLR1.jar" e digitare il comando "`java -jar riconoscitoreLR1.jar`".

A questo punto il riconoscitore, qualora non siano individuati errori lessicali, sintattici e/o semantici, costruirà l'automa di tipo LR(1) relativo alla grammatica fornita, andando a individuare eventuali conflitti che rendano la grammatica stessa non LR(1).

Al termine dell'analisi, il programma fornisce in output un elenco degli stati formati da regole core e regole di completamento, con i relativi look-ahead, insieme a un elenco delle transizioni che indicano quale carattere viene consumato per passare da uno stato all'altro.

Viene infine riportato se la grammatica è o meno LR(1) e, in caso non lo sia, vengono riportati gli stati che contengono i conflitti che rendono la grammatica non LR(1).

Possibili sviluppi futuri

Il progetto potrebbe presentare alcuni interessanti sviluppi futuri, tra questi si sottolineano:

- creazione di una GUI per l'interazione con l'applicativo, sia per l'inserimento delle grammatiche che per l'avvio dell'analisi;
- visualizzazione dei singoli stati o dell'automa completo LR(1) in formato grafico.

Bibliografia e riferimenti

https://it.wikipedia.org/wiki/Grammatica_libera_dal_contesto
https://it.wikipedia.org/wiki/Parser_LR