

Riconoscitore di grammatiche LR(1)

Manuale di utilizzo

Luca Filice, Matteo Gusmini, Davide Presciani



Università degli Studi di Bergamo
Dipartimento di Ingegneria Gestionale, dell'Informazione e della trasmissione

Indice

1. Introduzione	2
2. Struttura corretta della grammatica	2
2.1 Errori lessicali	2
2.2 Errori sintattici	2
2.3 Errori semantici	2
2.4 Esempio di struttura corretta della grammatica	3
3. Requisiti di sistema	3
4. Manuale di utilizzo	4
4.1 Selezione grammatica da analizzare	4
4.2 Esempio grammatica LR(1)	5
4.3 Esempio grammatica non LR(1)	5
4.4 Esempio grammatica con errori	6
4.5 Generazione e salvataggio del grafo	7

1. Introduzione

Il seguente documento illustra il manuale di utilizzo del riconoscitore di grammatiche LR(1).

2. Struttura corretta della grammatica

Il programma deve riconoscere come formalmente corrette (quindi prive di errori sintattici, lessicali e/o grammaticali) soltanto grammatiche che presentino la seguente struttura:

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

$$SZ \ EQ \ NT \ TER \ SC$$

- altre $n \geq 1$ regole di produzione *ar*, che formano il resto della grammatica, definite come segue

$$NT \ EQ \ (NT|CT)^* \ SC$$

I blocchi componenti le regole appena definite sono così traducibili:

Simbolo	Caratteri
<i>SZ</i>	<i>S0</i>
<i>EQ</i>	<code>- > :=</code>
<i>NT</i>	<i>A ... Z</i>
<i>CT</i>	<i>a ... z</i> <i>0 ... 9</i> <code>+</code> <code>-</code> <code>*</code> <code>/</code>
<i>TER</i>	<code>/swa</code> <code>/cjswa</code>
<i>SC</i>	<code>;</code>

Tabella 1: Corrispondenza tra caratteri della grammatica e blocchi di definizione delle regole

Nota: Per la definizione della struttura delle regole è stata utilizzata la notazione formale di Backus-Naur estesa (EBNF).

2.1 Errori lessicali

L'utilizzo di qualsiasi carattere non riconducibile alla colonna "Caratteri" della Tabella 1 corrisponde a un errore lessicale.

2.2 Errori sintattici

Gli errori sintattici sono dati dal mancato rispetto della struttura delle regole *pr* e *ar* come definite nel paragrafo "Struttura corretta della grammatica" a pagina 2.

2.3 Errori semantici

Gli errori semantici si verificano nei seguenti casi:

- nella grammatica è presente un carattere non terminale che non presenta regole di produzioni associate;
- nella grammatica è presente una regola duplicata (nota bene questo non è un errore bloccante).

2.4 Esempio di struttura corretta della grammatica

Un esempio di struttura corretta della grammatica, che dovrà essere inserita all'interno di un file .txt per permetterne la sua analisi, è la seguente:

```
S0->S/cjswa;  
S->X;  
X->aXbd;  
X->bXad;  
X->dX;  
X->;
```

Figura 1: Esempio di struttura corretta della grammatica

Easter Egg: Il carattere terminatore può essere scritto come /swa (abbreviazione di swarrow, ossia il nome in LaTeX della classica freccia utilizzata come terminatore) oppure, come in questo caso, come /cjswa in onore del Capitano Jack Sparrow (per la somiglianza tra swarrow e Sparrow).

3. Requisiti di sistema

Il programma è eseguibile su più SO tra cui Windows, Linux e Mac OS. L'unico requisito necessario per l'utilizzo del programma è l'installazione del Java Development Kit oppure del Java Runtime Environment più recente.

4. Manuale di utilizzo

4.1 Selezione grammatica da analizzare

All'avvio del programma ci si ritroverà davanti alla seguente pagina iniziale:

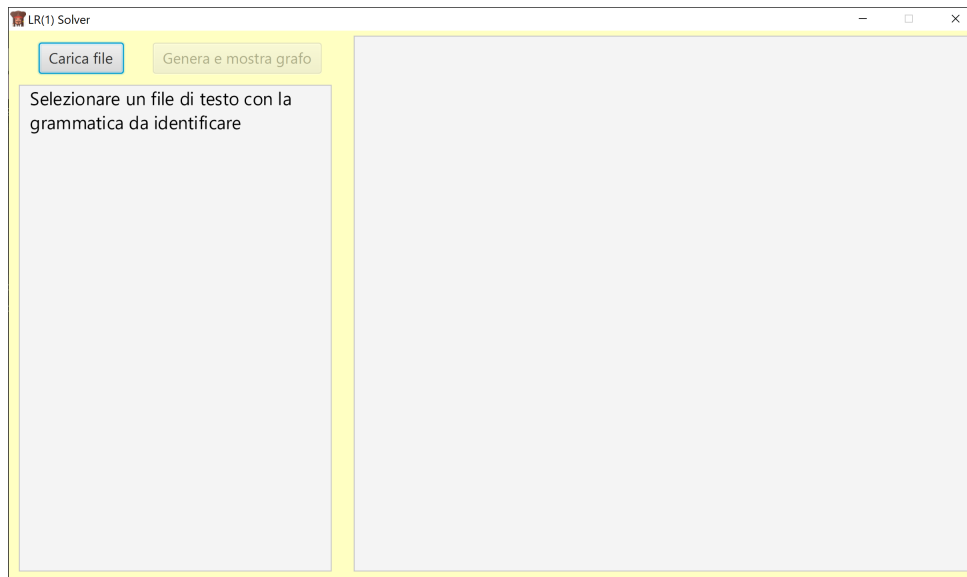


Figura 2: Pagina iniziale

Da questa finestra è possibile premere il tasto "Carica file" per selezionare il file .txt con la grammatica da analizzare.

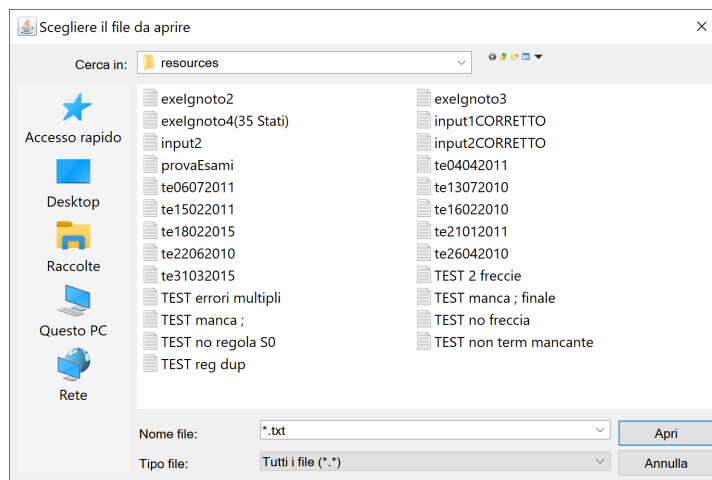


Figura 3: Finestra per la selezione del file

Una volta selezionato il file desiderato, il programma procederà all'analisi della grammatica, informando l'utente se essa sia o non sia di tipo LR(1) oppure se il file contenga degli errori.

4.2 Esempio grammatica LR(1)

Se la grammatica inserita è di tipo LR(1) verrà visualizzato un messaggio come il seguente:



Figura 4: Finestra con messaggio grammatica LR(1)

4.3 Esempio grammatica non LR(1)

Se la grammatica inserita non è di tipo LR(1) verrà visualizzato un messaggio come il seguente, dove verranno inoltre riportati i nomi degli stati in errore:

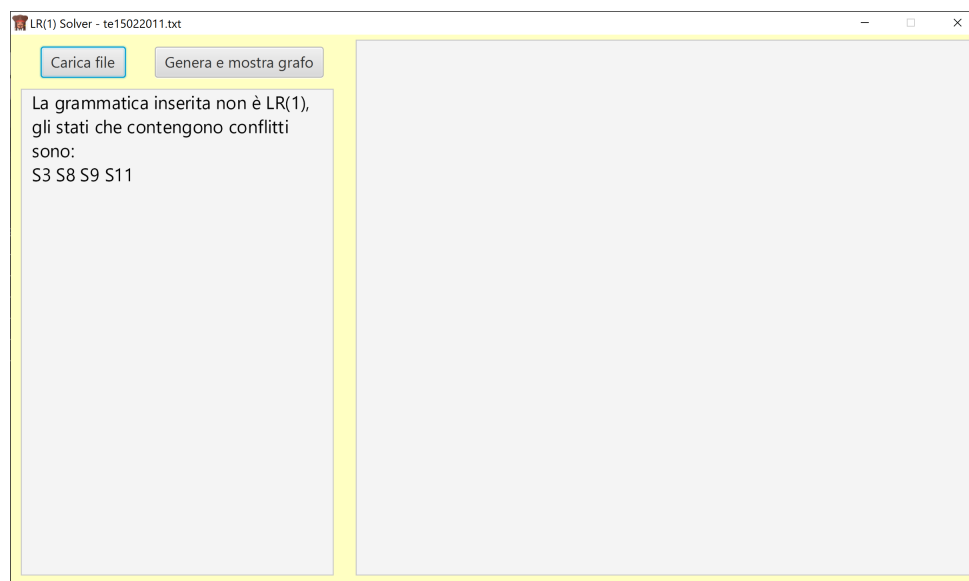


Figura 5: Finestra con messaggio grammatica non LR(1)

4.4 Esempio grammatica con errori

Se la grammatica inserita contiene errori verrà visualizzato un messaggio con il tipo di errore presente:

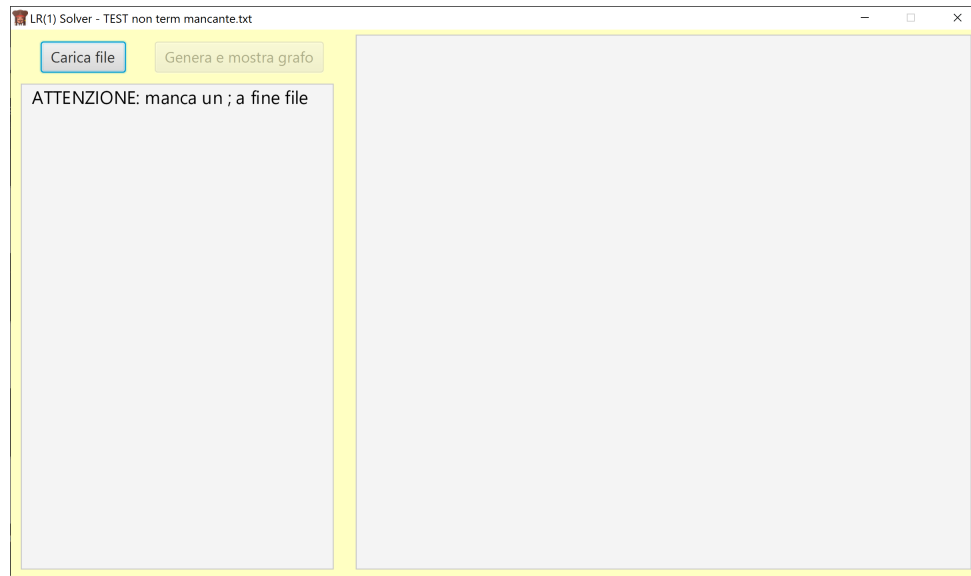


Figura 6: Finestra con messaggio grammatica errata

Se il tipo di errore è risolvibile automaticamente dal programma, verrà mostrato se la grammatica è o non è di tipo LR(1) oltre al tipo di errore presente.

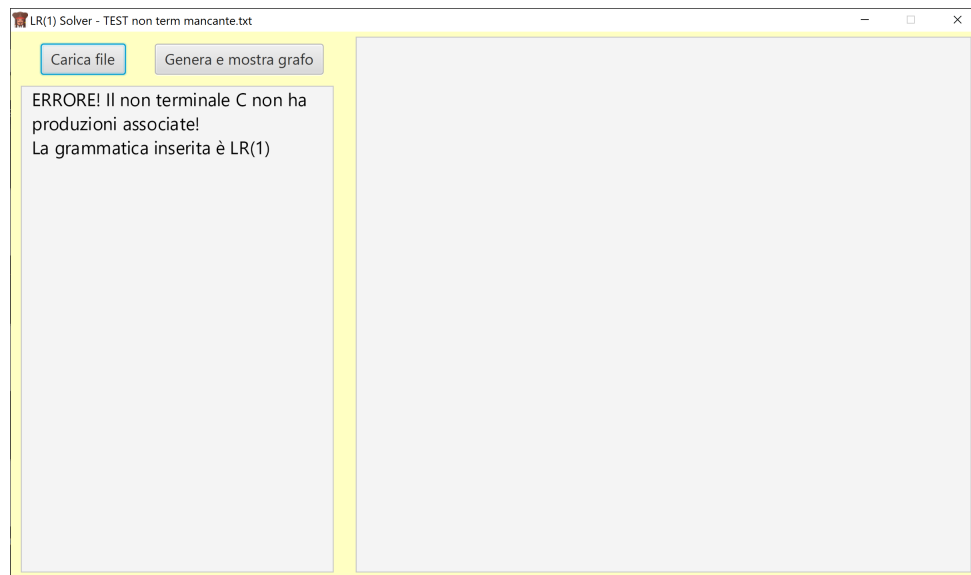


Figura 7: Finestra con messaggio grammatica LR(1), ma errata

4.5 Generazione e salvataggio del grafo

Una volta analizzata la grammatica, se essa non presenta errori bloccanti, sarà possibile, premendo l'apposito tasto in alto, generare e mostrare il grafo. Una volta premuto il tasto si aprirà una finestra, come la seguente, per scegliere dove salvare l'immagine generata (può essere salvata sia in .jpg che .png):

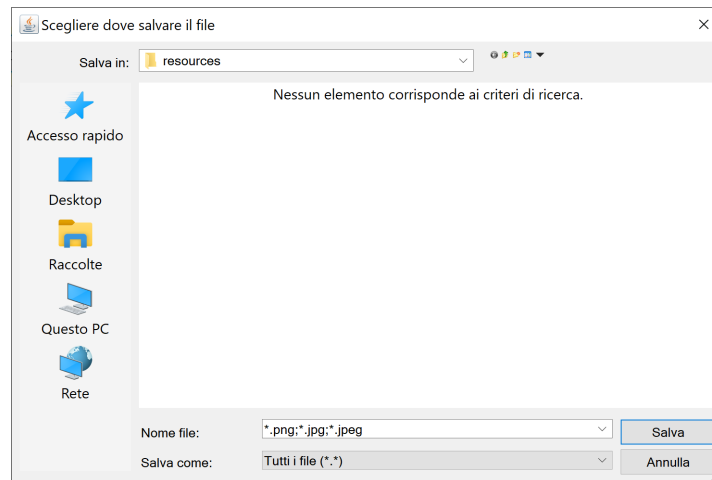


Figura 8: Finestra di salvataggio immagine grafo

A questo punto nella parte destra del programma verrà visualizzata l'immagine del grafo:

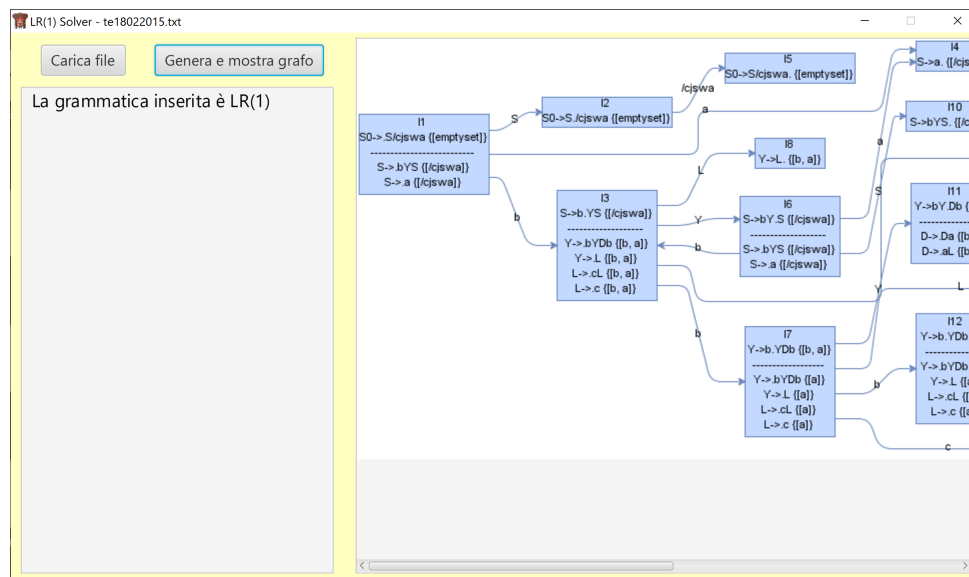


Figura 9: Grafo di una grammatica LR(1)

Se qualche stato dovesse invece presentare errori, essi sarebbero colorati di rosso:

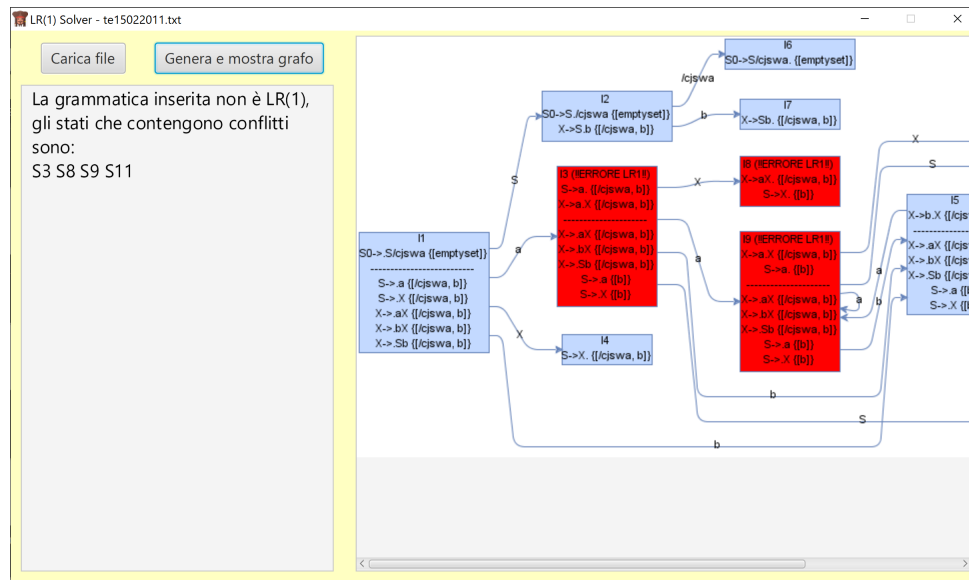


Figura 10: Grafo di una grammatica non LR(1)