

# SLR1 PARSER

IMPLEMENTAZIONE DI UN PARSER SLR1 PER GRAMMATICHE EBNF

Materia: Linguaggi e traduttori

Prof.ssa Vincenza Carchiolo

Cristina Lombardo  
Lombardo.cristina84@gmail.com

Università degli Studi di Catania  
Aa 2014/15

# SOMMARIO

- ✓ **Requisiti**
- ✓ **Scelte progettuali**
- ✓ **Implementazione**
- ✓ **Appendice: Esecuzione del programma**
- ✓ **Appendice: Compilazione con Maven**

# REQUISITI

Data una grammatica non contestuale verificare che:

La grammatica non contestuale sia di tipo SLR(1)

Nel caso che la grammatica non sia SLR(1) salvare su un file ERRORE le condizioni di conflitto.

Nel caso la grammatica soddisfi le condizioni generare le tabelle GOTO ed ACTION e salvarle su un file OUTPUT.TXT. Il formato del file è libero.

Frase appartiene al linguaggio di ingresso.

File di input: Grammatica non contestuale Il file di input contiene la grammatica non contestuale nella forma BNF Estesa (<http://www.cs.cmu.edu/~pattis/misc/ebnf.pdf>) . Ogni produzione è etichettata con un'etichetta numerica seguita dal simbolo ":".

# SCELTE PROGETTUALI

## EBNF Input

### ✓ Pattern per non terminali

✓ `<[a-zA-Z]+[0-9]*>`

### ✓ Riga del tipo

✓ `<numero>: <EBNF-line>`

### ✓ # riga commentata

### ✓ \ carattere di escape

```
# Created by Cristina Lombardo
# Rules
#           "\"" Comments
#           \ :Escape char
#           <E>, <F>, <T>, <X>: Non Terminal
#           i, +, -, ^: Terminals
#           \\: Escaped / used as terminal
#           \(: Special Char ( escaped used as terminal
#           () [] {} |: Special chars

#####

# Grammar

1: <E> ::= <X> { (+|-) <X> }
2: <X> ::= a|b|c

#Example of input string a+b-a-b
```

# SCELTE PROGETTUALI

## Salvataggio ActionGoto

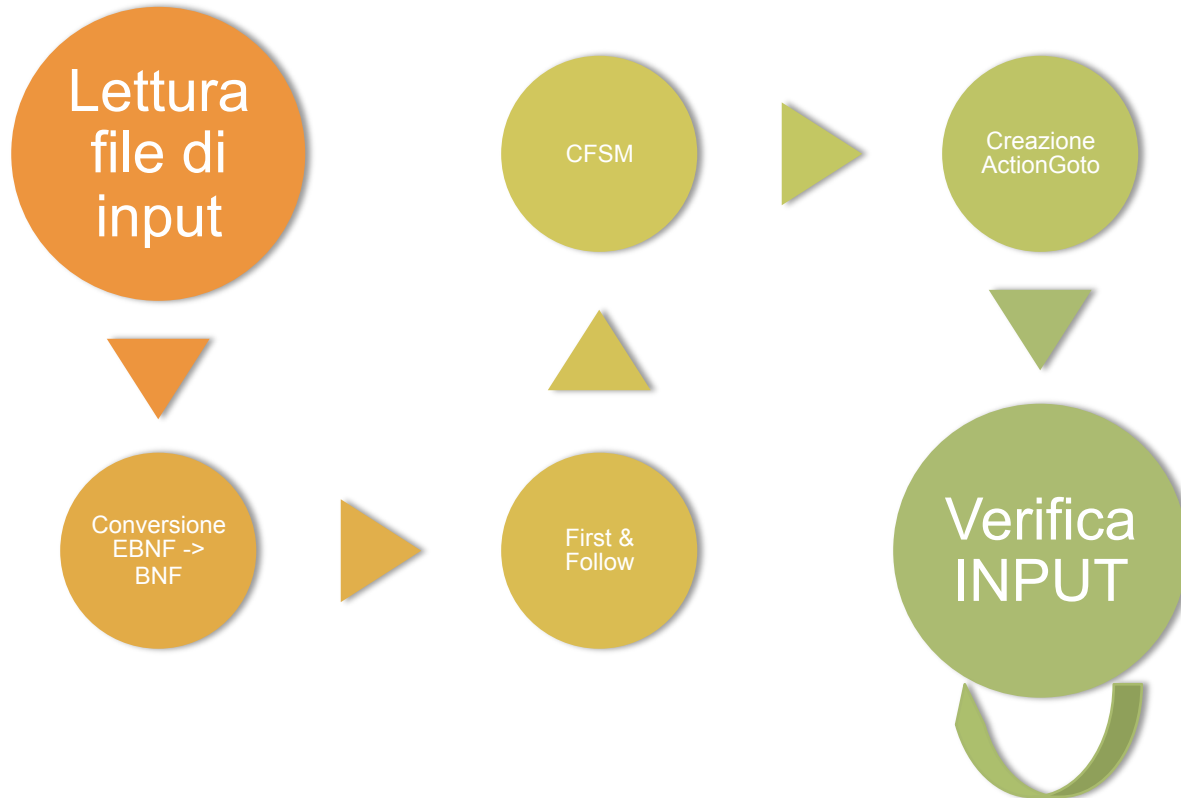
Nome file: output.txt - Formato dati CSV separati da “,”

```
state,a,b,c,+,-,$,E,X,f1,f2
S0,S1,S2,S3,,,4,6,,
S1,,,X->a ,X->a ,X->a ,,,,
S2,,,X->b ,X->b ,X->b ,,,,
S3,,,X->c ,X->c ,X->c ,,,,
S4,,,,,S5,,,
S5,Accept
S6,,,S7,S8,f1->>EPS< ,,,9,10
S7,f2->+ ,f2->+ ,f2->+ ,,,,,,
S8,f2->- ,f2->- ,f2->- ,,,,,,
S9,,,,,E->X f1 ,,,,
S10,S1,S2,S3,,,,,11,,
S11,,,S7,S8,f1->>EPS< ,,,12,10
S12,,,,,f1->f2 X f1 ,,,,
```



state	a	b	c	+	-	\$	E	X	f1	f2
S0	S1	S2	S3				4	6		
S1				X->a	X->a	X->a				
S2				X->b	X->b	X->b				
S3				X->c	X->c	X->c				
S4						S5				
S5	Accept									
S6				S7	S8	f1->>EPS<			9	10
S7	f2->+	f2->+	f2->+							
S8	f2->-	f2->-	f2->-							
S9						E->X f1				
S10	S1	S2	S3					1		
S11				S7	S8	f1->>EPS<			12	10
S12						f1->f2 X f1				

# IMPLEMENTAZIONE



# LETTURA FILE DI INPUT

- ✓ Eliminazione righe bianche
- ✓ Eliminazione righe commentate

```
# Created by Cristina Lombardo
# Rules
#         "#" Comments
#         \ :Escape char
#         <E>, <F>, <T>, <X>: Non Terminal
#         i, +, -, ^: Terminals
#         \\: Escaped / used as terminal
#         \(: Special Char ( escaped used as terminal
#         () [] {} |: Special chars

#####

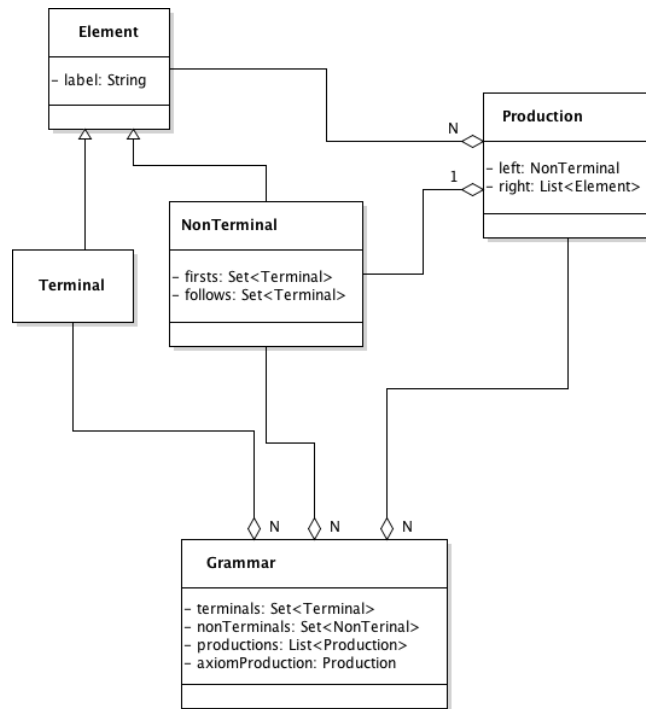
# Grammar

1: <E> ::= <X> { (+|-) <X> }
2: <X> ::= a|b|c

#Example of input string a+b-a-b
```

# CONVERSIONE EBNF -> BNF

- ✓ Grammatica Context Free  
o di Tipo2





# CONVERSIONE EBNF -> BNF

## TRASFORMAZIONE {}

**File di input**

1:  $\langle E \rangle ::= a\{b\}$

**Traduzione BNF**

$E \rightarrow a \text{ } \text{£1}$

$\text{£1} \rightarrow b \text{ } \text{£1}$

$\text{£1} \rightarrow \epsilon$

## TRASFORMAZIONE []

**File di input**

1:  $\langle E \rangle ::= a[b]$

**Traduzione BNF**

$E \rightarrow a \text{ } \text{£1}$

$\text{£1} \rightarrow b$

$\text{£1} \rightarrow \epsilon$

## TRASFORMAZIONE () |

**File di input**

1:  $\langle E \rangle ::= a(b|c)$

**Traduzione BNF**

$E \rightarrow a \text{ } \text{£1}$

$\text{£1} \rightarrow b$

$\text{£1} \rightarrow c$

# FIRST & FOLLOW

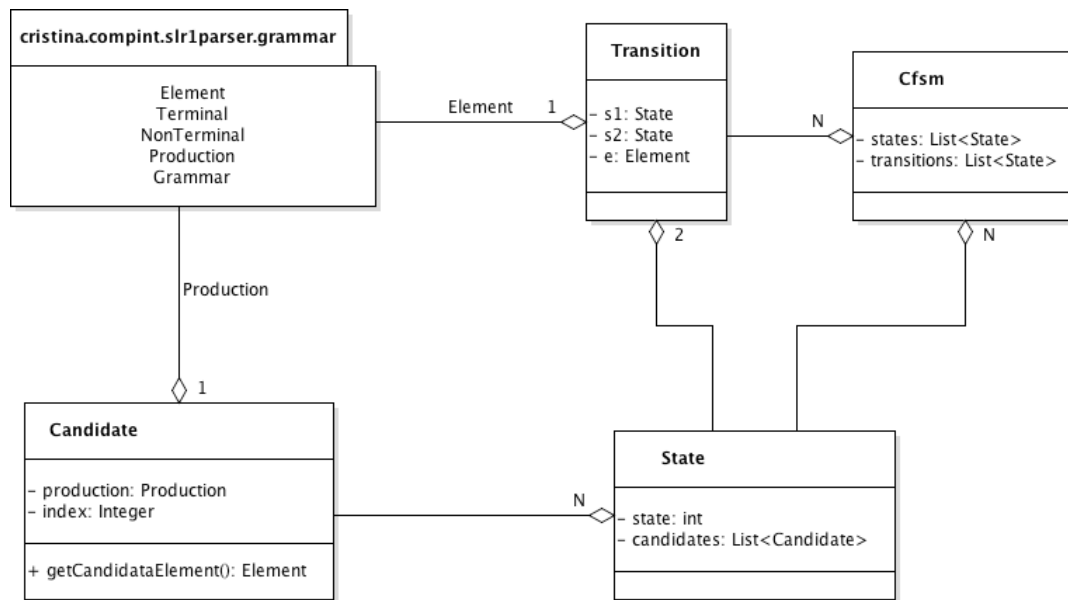
- ✓ **Set di terminali per ogni non terminale**
- ✓ **Funzioni ricorsive per il calcolo**

# CFSM

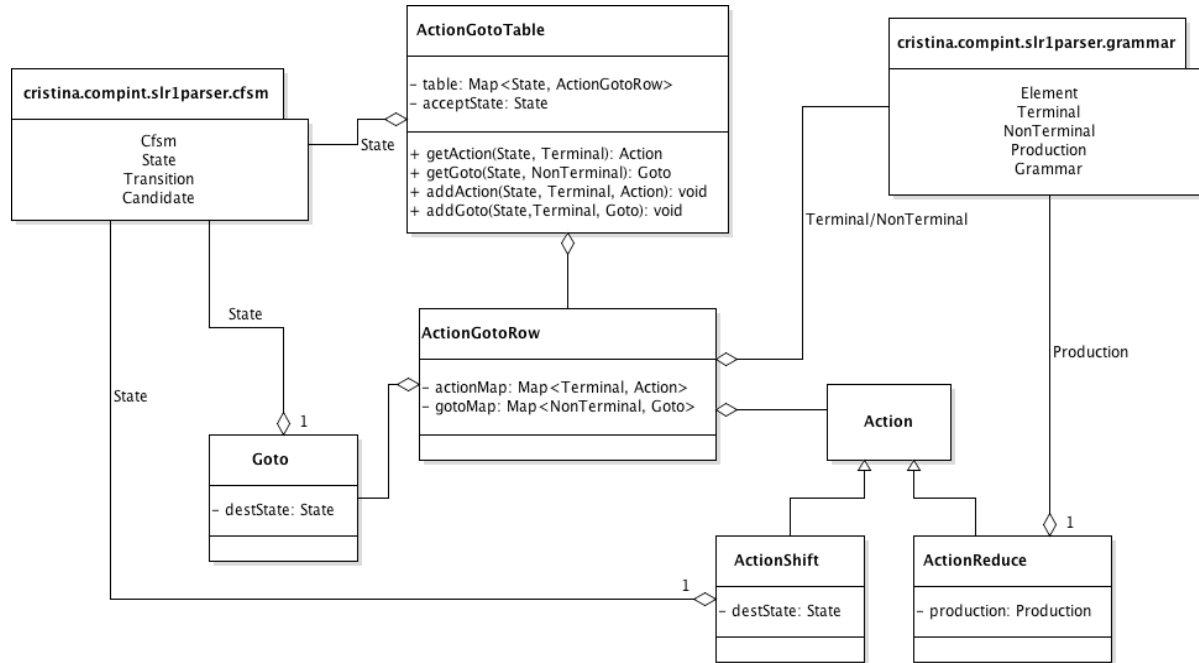
Durante la creazione della CFSM  
vengono verificate le ambiguità  
Shift-Reduce.

Nel caso uno stato presenti un  
ambiguità non eliminabile verrà  
lanciata un'eccezione di tipo

`cristina.compint.slr1parser.exception.C fsmException`



# ACTIONGOTO



# PARSING

state	a	b	c	+	-	\$	E	X	£1	£2
S0	S1	S2	S3				4	6		
S1				X->a	X->a	X->a				
S2				X->b	X->b	X->b				
S3				X->c	X->c	X->c				
S4						S5				
S5	Accept									
S6				S7	S8	£1->>EPS<			9	10
S7	£2->+	£2->+	£2->+							
S8	£2->-	£2->-	£2->-							
S9						E->X £1				
S10	S1	S2	S3					11		
S11				S7	S8	£1->>EPS<			12	10
S12						£1->£2 X £1				

Parsing string: a+b

STACK	INPUT	ACTION
>\$< 0	a+b>\$<	S1
>\$< 0, a 1	+b>\$<	X->a
>\$< 0, X 6	+b>\$<	S7
>\$< 0, X 6, + 7	b>\$<	£2->+
>\$< 0, X 6, £2 10	b>\$<	S2
>\$< 0, X 6, £2 10, b 2	>\$<	X->b
>\$< 0, X 6, £2 10, X 11	>\$<	£1->>EPS<
>\$< 0, X 6, £2 10, X 11, £1 12	>\$<	£1->£2 X £1
>\$< 0, X 6, £1 9	>\$<	E->X £1
>\$< 0, E 4	>\$<	S5