

Corso: Fondamenti, Linguaggi e Traduttori 2

Paola Giannini

Trasformazioni grammatica **ac** per analisi top-down

```
int tempa;  
tempa = 5;  
float tempb = tempa + 3.2;  
tempb = tempb * 7;  
print tempb;
```

dichiarazione variabile
istruzione di assegnamento
dichiarazione variabile con inizializzazione
istruzione di assegnamento di espressione a variabile
istruzione di stampa variabile

Un programma è una sequenza di istruzioni e dichiarazioni.

Il lessico e la grammatica di partenza

Il lessico

Token	Simboli usati nella grammatica ^a
INT	intVal
FLOAT	floatVal
ID	id
TYINT	int
TYFLOAT	float
ASSIGN	=
PRINT	print
PLUS	+
MINUS	-
TIMES	*
DIVIDE	/
SEMICOLON	;
EOF	\$

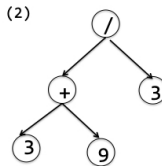
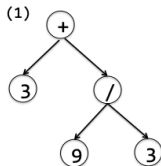
^aper non confondere i token con i simboli nonterminali che iniziano per maiuscola

Una grammatica iniziale

0. $Prg \rightarrow DSs \$$
1. $DSs \rightarrow Dcl DSs$
2. $DSs \rightarrow Stm DSs$
3. $DSs \rightarrow \epsilon$
4. $Dcl \rightarrow Ty id;$
5. $Dcl \rightarrow Ty id = Exp;$
6. $Stm \rightarrow id = Exp;$
7. $Stm \rightarrow print id;$
8. $Exp \rightarrow Exp + Exp$
9. $Exp \rightarrow Exp - Exp$
10. $Exp \rightarrow Exp * Exp$
11. $Exp \rightarrow Exp / Exp$
12. $Exp \rightarrow Val$
13. $Ty \rightarrow float$
14. $Ty \rightarrow int$
- 1X. $Val \rightarrow intVal \mid floatVal \mid id$

Trasformazioni della Grammatica: specifica precedenze (1)

- Il primo passo è decidere come vogliamo rendere **deterministica** la grammatica. Cioè data una stringa derivata da Exp , ad esempio $3+9/3$ quale derivazione scegliamo:
 - $Exp \xrightarrow{8} Exp + Exp \xrightarrow{12} Val + Exp \xrightarrow{1X} intVal + Exp \xrightarrow{11} intVal + Exp / Exp \xrightarrow{12} intVal + Val / Exp \xrightarrow{1X} intVal + intVal / Exp \xrightarrow{12} intVal + intVal / Val \xrightarrow{1X} intVal + intVal / intVal$
 - $Exp \xrightarrow{11} Exp / Exp \xrightarrow{8} Exp + Exp / Exp \xrightarrow{12} Val + Exp / Exp \xrightarrow{1X} intVal + Exp / Exp \xrightarrow{12} intVal + Val / Exp \xrightarrow{1X} intVal + intVal / Exp \xrightarrow{12} intVal + intVal / Val \xrightarrow{1X} intVal + intVal / intVal$
- quindi a due AST (Abstract Syntax Tree) diversi.



- Il codice oggetto, nel nostro caso **notazione polacca inversa** viene generato dal AST

(1) 3 9 3 / +

(2) 3 9 + 3 /

Quindi **diverso risultato**.

Trasformazioni della Grammatica: specifica precedenze (2)

- La convenzione è che $*$ e $/$ abbiano la **precedenza** su $+$ e $-$.
- Aggiungiamo un nuovo simbolo terminale Tr , da Exp facciamo generare le somme e sottrazioni di espressioni, mentre da Tr generiamo le moltiplicazioni e divisioni.
- Abbiamo due modi di scrivere la grammatica risultante, vero?

(1)
 $Exp \rightarrow Tr + Exp$
 $Exp \rightarrow Tr - Exp$
 $Exp \rightarrow Tr$
 $Tr \rightarrow Val * Tr$
 $Tr \rightarrow Val / Tr$
 $Tr \rightarrow Val$
 $Val \rightarrow intVal \mid floatVal \mid id$

(2)
 $Exp \rightarrow Exp + Tr$
 $Exp \rightarrow Exp - Tr$
 $Exp \rightarrow Tr$
 $Tr \rightarrow Tr * Val$
 $Tr \rightarrow Tr / Val$
 $Tr \rightarrow Val$
 $Val \rightarrow intVal \mid floatVal \mid id$

- Quando abbiamo operazioni con la stessa precedenza dobbiamo decidere la loro **associatività**.
- Quale associatività abbiamo specificato in (1) e in (2) per $+$ e $-$? Cioè se abbiamo $3+5-9$ cosa intendiamo

$(3+5)-9$ left-to-right oppure $3+(5-9)$ right-to-left

- e per $*$ e $/$?

Specifica precedenza e associatività

La grammatica iniziale

0. $Prg \rightarrow DSs \$$
1. $DSs \rightarrow Dcl DSs$
2. $DSs \rightarrow Stm DSs$
3. $DSs \rightarrow \epsilon$
4. $Dcl \rightarrow Ty id;$
5. $Dcl \rightarrow Ty id = Exp;$
6. $Stm \rightarrow id = Exp;$
7. $Stm \rightarrow print id;$
8. $Exp \rightarrow Exp + Exp$
9. $Exp \rightarrow Exp - Exp$
10. $Exp \rightarrow Exp * Exp$
11. $Exp \rightarrow Exp / Exp$
12. $Exp \rightarrow Val$
13. $Ty \rightarrow float$
14. $Ty \rightarrow int$
- 1X. $Val \rightarrow intVal \mid floatVal \mid id$

La grammatica non ambigua

0. $Prg \rightarrow DSs \$$
1. $DSs \rightarrow Dcl DSs$
2. $DSs \rightarrow Stm DSs$
3. $DSs \rightarrow \epsilon$
4. $Dcl \rightarrow Ty id;$
5. $Dcl \rightarrow Ty id = Exp;$
6. $Stm \rightarrow id = Exp;$
7. $Stm \rightarrow print id;$
8. $Exp \rightarrow Exp + Tr$
9. $Exp \rightarrow Exp - Tr$
10. $Exp \rightarrow Tr$
11. $Tr \rightarrow Tr * Val$
12. $Tr \rightarrow Tr / Val$
13. $Tr \rightarrow Val$
14. $Ty \rightarrow float$
15. $Ty \rightarrow int$
- 1X. $Val \rightarrow intVal \mid floatVal \mid id$

Rimozione ricorsione sinistra

La grammatica con ricorsione sinistra

0. $Prg \rightarrow DSs \$$
1. $DSs \rightarrow Dcl DSs$
2. $DSs \rightarrow Stm DSs$
3. $DSs \rightarrow \epsilon$
4. $Dcl \rightarrow Ty \text{ id};$
5. $Dcl \rightarrow Ty \text{ id} = Exp;$
6. $Stm \rightarrow \text{id} = Exp;$
7. $Stm \rightarrow \text{print id};$
8. $Exp \rightarrow Exp + Tr$
9. $Exp \rightarrow Exp - Tr$
10. $Exp \rightarrow Tr$
11. $Tr \rightarrow Tr * Val$
12. $Tr \rightarrow Tr / Val$
13. $Tr \rightarrow Val$
14. $Ty \rightarrow \text{float}$
15. $Ty \rightarrow \text{int}$
- 1X. $Val \rightarrow \text{intVal} \mid \text{floatVal} \mid \text{id}$

Non ricorsiva sinistra

0. $Prg \rightarrow DSs \$$
1. $DSs \rightarrow Dcl DSs$
2. $DSs \rightarrow Stm DSs$
3. $DSs \rightarrow \epsilon$
4. $Dcl \rightarrow Ty \text{ id};$
5. $Dcl \rightarrow Ty \text{ id} = Exp;$
6. $Stm \rightarrow \text{id} = Exp;$
7. $Stm \rightarrow \text{print id};$
8. $Exp \rightarrow Tr ExpP$
9. $ExpP \rightarrow -Tr ExpP$
10. $ExpP \rightarrow +Tr ExpP$
11. $ExpP \rightarrow \epsilon$
12. $Tr \rightarrow Val TrP$
13. $TrP \rightarrow /Val TrP$
14. $TrP \rightarrow *Val TrP$
15. $TrP \rightarrow \epsilon$
16. $Ty \rightarrow \text{float}$
17. $Ty \rightarrow \text{int}$
- 1X. $Val \rightarrow \text{intVal} \mid \text{floatVal} \mid \text{id}$

Fattorizzazione (rimozione prefissi comuni)

La grammatica con prefissi comuni *Dcl*

0. $Prg \rightarrow DSs \$$
1. $DSs \rightarrow Dcl DSs$
2. $DSs \rightarrow Stm DSs$
3. $DSs \rightarrow \epsilon$
4. $Dcl \rightarrow Ty \text{ id};$
5. $Dcl \rightarrow Ty \text{ id} = Exp;$
6. $Stm \rightarrow \text{id} = Exp;$
7. $Stm \rightarrow \text{print id};$
8. $Exp \rightarrow Tr ExpP$
9. $ExpP \rightarrow -Tr ExpP$
10. $ExpP \rightarrow +Tr ExpP$
11. $ExpP \rightarrow \epsilon$
12. $Tr \rightarrow Val TrP$
13. $TrP \rightarrow /Val TrP$
14. $TrP \rightarrow *Val TrP$
15. $TrP \rightarrow \epsilon$
16. $Ty \rightarrow \text{float}$
17. $Ty \rightarrow \text{int}$
- 1X. $Val \rightarrow \text{intVal} \mid \text{floatVal} \mid \text{id}$

La grammatica FINALE

0. $Prg \rightarrow DSs \$$
1. $DSs \rightarrow Dcl DSs$
2. $DSs \rightarrow Stm DSs$
3. $DSs \rightarrow \epsilon$
4. $Dcl \rightarrow Ty \text{ id } Dcl'$
5. $Dcl' \rightarrow ;$
6. $Dcl' \rightarrow = Exp;$
7. $Stm \rightarrow \text{id} = Exp;$
8. $Stm \rightarrow \text{print id};$
9. $Exp \rightarrow Tr ExpP$
10. $ExpP \rightarrow -Tr ExpP$
11. $ExpP \rightarrow +Tr ExpP$
12. $ExpP \rightarrow \epsilon$
13. $Tr \rightarrow Val TrP$
14. $TrP \rightarrow /Val TrP$
15. $TrP \rightarrow *Val TrP$
16. $TrP \rightarrow \epsilon$
17. $Ty \rightarrow \text{float}$
18. $Ty \rightarrow \text{int}$
- 1X. $Val \rightarrow \text{intVal} \mid \text{floatVal} \mid \text{id}$

Calcolare la tabella dei Predict delle produzioni di ac

Num.	LHS	RHS	Predict
0.	<i>Prg</i>	<i>DSs \$</i>	
1.	<i>DSs</i>	<i>Dcl DSs</i>	
2.	<i>DSs</i>	<i>Stm DSs</i>	
3.	<i>DSs</i>	ϵ	
4.	<i>Dcl</i>	<i>Ty idDcl'</i>	
5.	<i>Dcl'</i>	$;$	
6.	<i>Dcl'</i>	$= Exp;$	
7.	<i>Stm</i>	$id = Exp;$	
8.	<i>Stm</i>	$print id;$	
9.	<i>Exp</i>	<i>Tr ExpP</i>	
10.	<i>ExpP</i>	$+ Tr ExpP$	
11.	<i>ExpP</i>	$- Tr ExpP$	
12.	<i>ExpP</i>	ϵ	
13.	<i>Tr</i>	<i>Val TrP</i>	
14.	<i>TrP</i>	$* Val TrP$	
15.	<i>TrP</i>	$/ Val TrP$	
16.	<i>TrP</i>	ϵ	
17.	<i>Ty</i>	$float$	
18.	<i>Ty</i>	int	
19.	<i>Val</i>	$intVal$	
20.	<i>Val</i>	$floatVal$	
21.	<i>Val</i>	id	

Calcolare la tabella dei Predict delle produzioni di ac

Num.	LHS	RHS	First	Follow	Predict
0.	<i>Prg</i>	<i>DSs</i> \$			
1.	<i>DSs</i>	<i>Dcl DSs</i>			
2.	<i>DSs</i>	<i>Stm DSs</i>			
3.	<i>DSs</i>	ϵ			
4.	<i>Dcl</i>	<i>Ty idDcl'</i>			
5.	<i>Dcl'</i>	$;$			
6.	<i>Dcl'</i>	$= Exp;$			
7.	<i>Stm</i>	<i>id = Exp;</i>			
8.	<i>Stm</i>	<i>print id;</i>			
9.	<i>Exp</i>	<i>Tr ExpP</i>			
10.	<i>ExpP</i>	$+ Tr ExpP$			
11.	<i>ExpP</i>	$- Tr ExpP$			
12.	<i>ExpP</i>	ϵ			
13.	<i>Tr</i>	<i>Val TrP</i>			
14.	<i>TrP</i>	$* Val TrP$			
15.	<i>TrP</i>	$/ Val TrP$			
16.	<i>TrP</i>	ϵ			
17.	<i>Ty</i>	<i>float</i>			
18.	<i>Ty</i>	<i>int</i>			
19.	<i>Val</i>	<i>intVal</i>			
20.	<i>Val</i>	<i>floatVal</i>			
21.	<i>Val</i>	<i>id</i>			