

## Linguaggi Formali e Traduttori

– grammatiche LL(1) e parsificazione top-down –

### Alcuni esercizi risolti

1. Eliminare la ricorsione sinistra dalle grammatiche con le produzioni seguenti:

$$\begin{array}{ll} P_1: S \rightarrow R \mid 0 & P_2: S \rightarrow Ss- \mid S\{S\} \mid D \mid \varepsilon \\ R \rightarrow RC \mid R0 \mid C & D \rightarrow TL-R \\ C \rightarrow 1 \mid 2 \mid 3 & R \rightarrow \varepsilon \mid D \quad \text{N.B. } \{s, -, \{, \}, ,, m, i, r\} \\ & L \rightarrow L;m \mid m \quad \text{è l'insieme dei simboli terminali} \\ & T \rightarrow i \mid r \end{array}$$

Produzioni senza ricorsioni sinistre di grammatiche equivalenti:

$$\begin{array}{ll} P_1': S \rightarrow R \mid 0 & P_2': S \rightarrow DS' \mid S' \quad S' \rightarrow s-S' \mid \{S\}S' \mid \varepsilon \\ R \rightarrow CR' & D \rightarrow TL-R \\ R' \rightarrow CR' \mid 0R' \mid \varepsilon & R \rightarrow \varepsilon \mid D \\ C \rightarrow 1 \mid 2 \mid 3 & L \rightarrow mL' \quad L' \rightarrow ;mL' \mid \varepsilon \\ & T \rightarrow i \mid r \end{array}$$

2. Per ognuna delle seguenti grammatiche, specificate dall'insieme delle produzioni, verificare che sia LL(1) e costruire l'analizzatore a discesa ricorsiva.

$P_1: N \rightarrow D K$                        $P_2: T \rightarrow S T \mid \varepsilon$   
 $K \rightarrow N \mid \varepsilon$                                $S \rightarrow id := E;$   
 $D \rightarrow 0 \mid 1$                                  $E \rightarrow id G$   
 $G \rightarrow + id G \mid \varepsilon$

Produzioni $P_1$	Insiemi guida	
$N \rightarrow D K$	$\{0, 1\}$	
$K \rightarrow N$	$\{0, 1\}$	
$K \rightarrow \varepsilon$	$\{\$ \}$	
$D \rightarrow 0$	$\{0\}$	
$D \rightarrow 1$	$\{1\}$	La grammatica è LL(1)

Analizzatore a discesa ricorsiva:

Program discesa\_ricorsiva( )

```

cc ← PROSS( )
N( )
if (cc = '$') "stringa accettata"
else ERRORE (...)
```

function N( )

```

if (cc ∈ {'0', '1'})
D( )
K( )
else ERRORE (...)
```

function K( )

```

if (cc ∈ {'0', '1'})
N( )
elseif (cc = '$') do nothing
else ERRORE (...)
```

function D( )

```

if (cc = '0')
cc ← PROSS( )
elseif (cc = '1')
cc ← PROSS( )
else ERRORE (...)
```

Produzioni $P_2$	Insiemi guida	
$T \rightarrow S T$	{id}	
$T \rightarrow \varepsilon$	{\\$}	
$S \rightarrow id := E;$	{id}	
$E \rightarrow id G$	{id}	
$G \rightarrow + id G$	{+}	
$G \rightarrow \varepsilon$	{;}	La grammatica è LL(1)

Analizzatore a discesa ricorsiva:

Program discesa\_ricorsiva( )

```

cc ← PROSS( )
T( )
if (cc = '$') "stringa accettata"
else ERRORE (...)
```

function T( )

```

if (cc = 'id')
    S( )
    T( )
elseif (cc = '$') do nothing
else ERRORE (...)
```

function S( )

```

if (cc = 'id') cc ← PROSS( )
    if (cc = ':=' ) cc ← PROSS( )
    else ERRORE (... )
    E( )
    if (cc = ';' ) cc ← PROSS( )
    else ERRORE (... )
else ERRORE (...)
```

function G( )

```

if (cc = '+' ) cc ← PROSS( )
    if (cc = 'id' ) cc ← PROSS( )
    else ERRORE (... )
    G( )
elseif (cc = ';' ) do nothing
else ERRORE (...)
```

function E( )

```

if (cc = 'id' )
    cc ← PROSS( )
    G( )
else ERRORE (...)
```

3. Per ognuna delle seguenti grammatiche:

$$G_1 = (\{S,A\}, \{a,b,c\}, \{S \rightarrow aAbS \mid cA, A \rightarrow aScA \mid \varepsilon\}, S)$$

$$G_2 = (\{S,A\}, \{a,b,c,d\}, \{S \rightarrow aA \mid \varepsilon, A \rightarrow bAd \mid S\}, S)$$

- Calcolare gli insiemi FIRST e FOLLOW delle variabili;
- Calcolare gli insiemi guida delle produzioni;
- Scrivere il parsificatore a discesa ricorsiva.

$$\text{Grammatica } G_1: F(S) = F(aAbS) \cup F(cA) = \{a, c\} \quad FW(S) = \{c, \$\}$$

$$F(A) = F(aScA) \cup F(\varepsilon) = \{a, \varepsilon\} \quad FW(A) = \{b\} \cup FW(S) = \{b, c, \$\}$$

Insieme guida

$S \rightarrow aAbS$	$\{a\}$
$S \rightarrow cA$	$\{c\}$
$A \rightarrow aScA$	$\{a\}$
$A \rightarrow \varepsilon$	$\{b, c, \$\}$

function S()

```

if (cc = 'a') cc ← PROSS()
    A()
    if (cc = 'b') cc ← PROSS()
    else ERRORE (...)
    S()
elseif (cc = 'c') cc ← PROSS()
    A (...)
else ERRORE (...)

```

function A()

```

if (cc = 'a') cc ← PROSS()
    S()
    if (cc = 'c') cc ← PROSS()
    else ERRORE (...)
    A()
elseif (cc ∈ {'b', 'c', '$'}) do nothing
else ERRORE (...)

```

**Remark:** come previsto dal linguaggio di disegno introdotto a lezione, l'istruzione di "return" è stata omessa in quanto le funzioni del parsificatore non restituiscono nessun valore.