

# Esercizio 1: ASL: Bottom-Up

Si consideri la grammatica  $G$  data per un linguaggio di espressioni con somma e prodotto associativi a sinistra, e prodotto con precedenza sulla somma:

$$E ::= E + F \mid F$$
$$F ::= F * T \mid T$$
$$T ::= \text{num} \mid \text{ide} \mid (E)$$

- a) Si dia uno schema di traduzione ascendente per una oblivious bottom-up che calcoli un attributo *ast* contenente l'albero di sintassi astratta della frase riconosciuta dal simbolo.
- b) Si utilizzi lo schema per il riconoscimento e il calcolo degli attributi della frase:  $x+5*y$ . Allo scopo si mostri la sequenza di stati dell'esecutore oblivious durante l'analisi della frase.

# Ex1- (a): La grammatica G

```
E1::= E2 + F {E1.ast=mkT("+", E1.ast, F.ast);}
E::= F {E.ast=F.ast;}
F1::= F2 + T {F1.ast=mkT("+", F1.ast, T.ast);}
F::= T {F.ast=T.ast;}
T::= num {T.ast=mkT("num");}
T::= ide {T.ast= mkT("ide");}
T::= (E) {T.ast=E.ast;}
```

# Ex1- (b): Esecuzione Oblivious

La stringa da riconoscere

**x+5\*y**

Tabella di Analisi

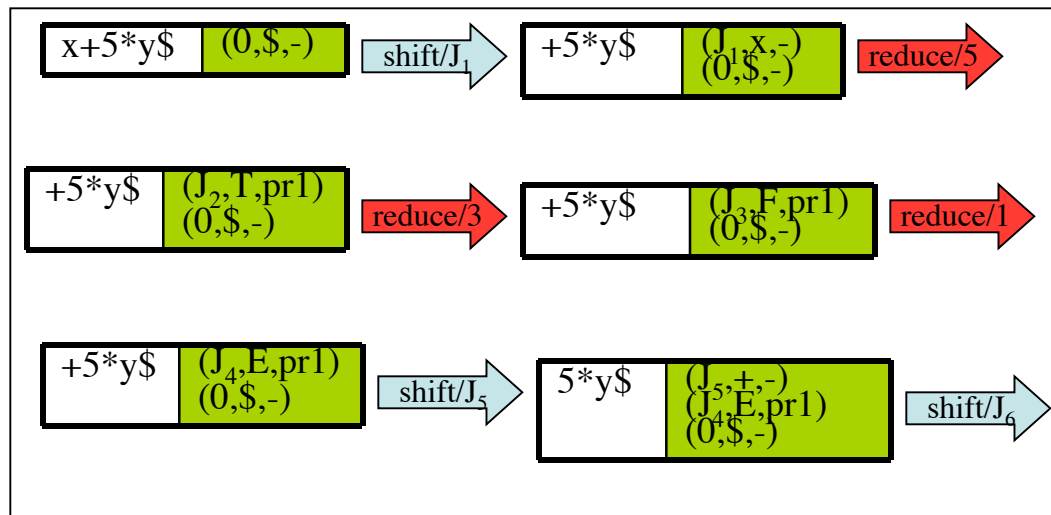
Non la calcoliamo ma ne useremo gli stati in accordo alle azioni shift e reduce che si renderanno necessarie nel corso del riconoscimento

Lo schema di traduzione

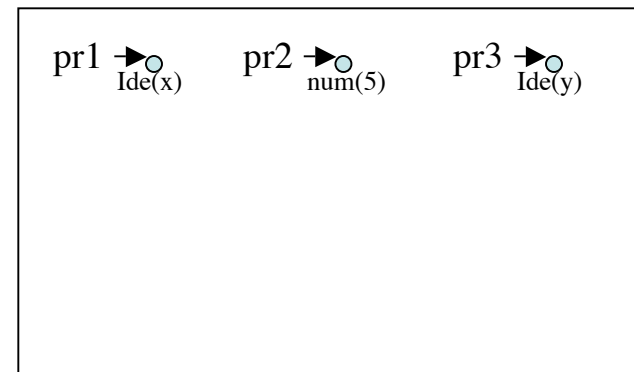
```

E1::= E2 + F {E1.ast=mkT("+", E1.ast, F.ast);}
E::= F {E.ast=F.ast;}
F1::= F2 * T {F1.ast=mkT("*", F1.ast, T.ast);}
F::= T {F.ast=T.ast;}
T::= num {T.ast=mkT("num");}
T::= ide {T.ast=mkT("ide");}
T::= (E) {T.ast=E.ast;}
    
```

Stati



Memoria Dinamica



# Ex1- (b): continua

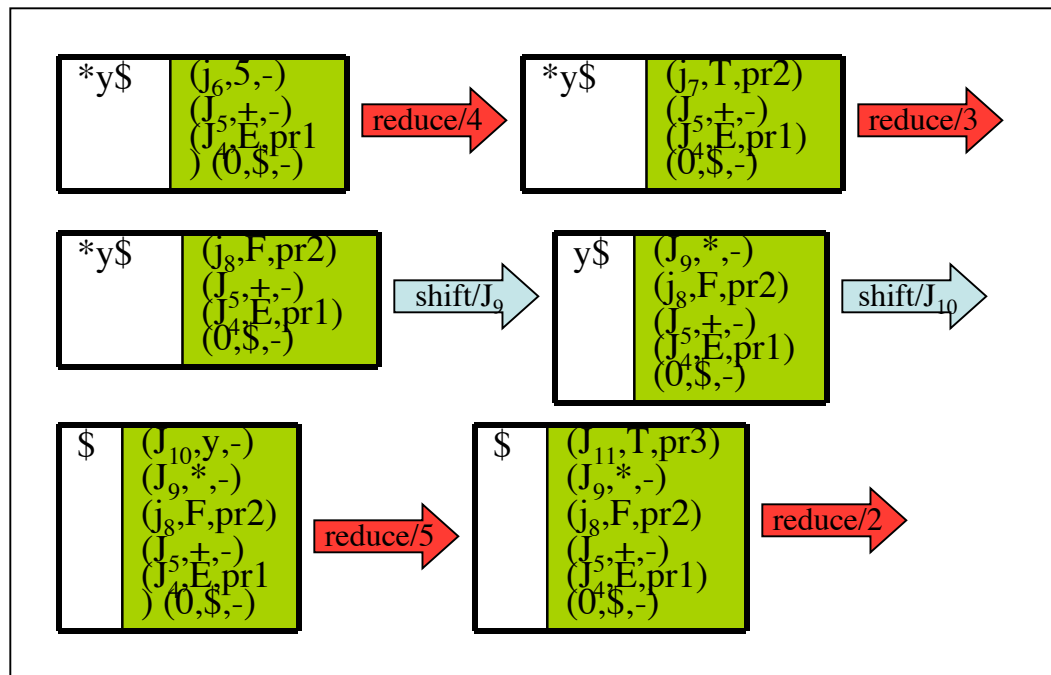
Lo schema di traduzione

```

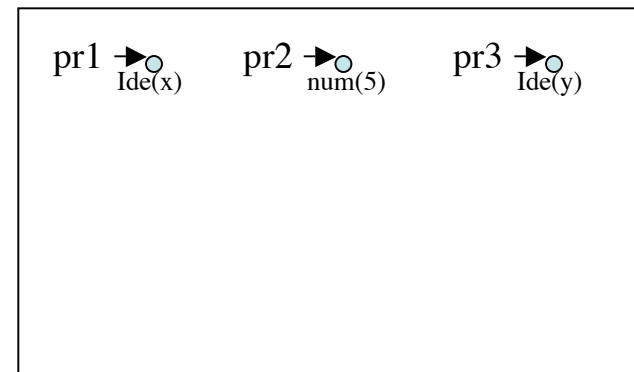
E1::= E2 + F {E1.ast=mkT("+", E1.ast, F.ast);}
E::= F {E.ast=F.ast;}
F1::= F2 * T {F1.ast=mkT("*", F1.ast, T.ast);}
F::= T {F.ast=T.ast;}
T::= num {T.ast=mkT("num");}
T::= ide {T.ast= mkT("ide");}
T::= (E) {T.ast=E.ast;}
    
```

+5*y\$	(J <sub>6</sub> ,x,-) (0,\$,-)
--------	-----------------------------------

## Stati



## Memoria Dinamica

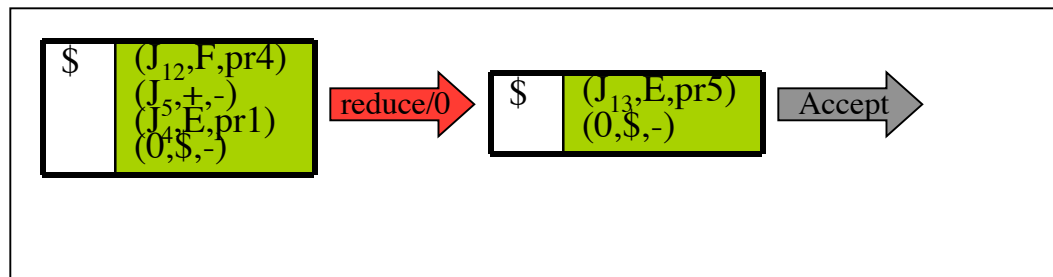


# Ex1- (b): continua-2

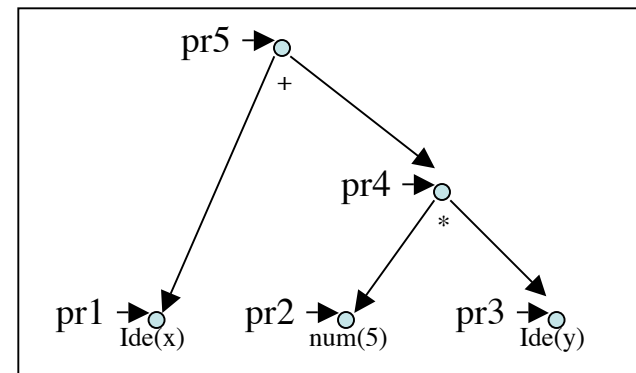
Lo schema di traduzione

```
E1::= E2 + F {E1.ast=mkT("+", E1.ast, F.ast);}
E::= F {E.ast=F.ast;}
F1::= F2 * T {F1.ast=mkT("*", F1.ast, T.ast);}
F::= T {F.ast=T.ast;}
T::= num {T.ast=mkT("num");}
T::= ide {T.ast= mkT("ide");}
T::= (E) {T.ast=E.ast;}
```

Stati



Memoria Dinamica



# Esercizio 2. ASL: Top-Down

- a) Si fornisca una grammatica  $G$  analizzabile LL(1) per un linguaggio di espressioni con somma e prodotto associativi a sinistra, e prodotto con precedenza sulla somma:
- b) Si dia uno schema di traduzione discendente per una oblivious top-down che calcoli un attributo *ast* contenente l'albero di sintassi astratta della frase riconosciuta dal simbolo.
- c) Si utilizzi lo schema per il riconoscimento e il calcolo degli attributi della frase:  $x+5*y$ . Allo scopo si mostri la sequenza di stati dell'esecutore oblivious durante l'analisi della frase.

# Esercizio 3. Source-to-Source

- a) Utilizzando la grammatica di Esercizio 1, Si dia uno schema di traduzione ascendente per una oblivious bottom-up che calcola un attributo *op* contenente il carattere  $\{+,*,u\}$  a seconda che il primo operatore della frase derivata dal simbolo sia +, ovvero sia \*, ovvero la frase non contenga operatori.
- b) Si utilizzi l'attributo *op* estendendo lo schema precedente in uno schema che risolva l'esercizio 5.4 [AHO], ovvero trasformi ogni espressione in input in una espressione equivalente nella quale sono rimosse tutte e sole le parentesi ridondanti. Ad esempio:  $((a*(b+c))*d)$  è trasformata nella espressione  $a*(b+c)*d$ .