

Langaggi Formali e Traduttori

5.6 Traduzione di comandi

- Sommario
- Grammatica dei comandi
- Assegnamento
- Comandi condizionali
- Comandi iterativi
- Composizione sequenziale
- Esempio: somma dei primi 10 numeri naturali
- Esercizi

È proibito condividere e divulgare in qualsiasi forma i materiali didattici caricati sulla piattaforma e le lezioni svolte in videoconferenza: ogni azione che viola questa norma sarà denunciata agli organi di Ateneo e perseguita a termini di legge.

Sommario

- In questa lezione presentiamo le SDD per la traduzione di comandi.

Riferimenti esterni

- [Java Language and Virtual Machine Specifications](#)
- [JVM Instruction set](#)

Grammatica dei comandi

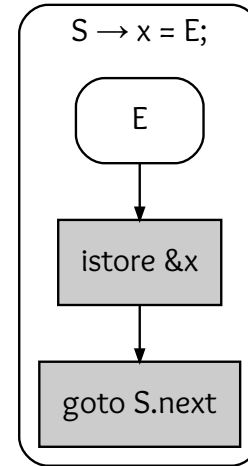
Produzioni	Descrizione
$S \rightarrow x = E;$	Assegna il valore di E ad x
$S \rightarrow \text{if } (B) S_1$	Esegue S_1 se B è vera, non fa nulla altrimenti
$S \rightarrow \text{if } (B) S_1 \text{ else } S_2$	Esegue S_1 se B è vera e S_2 altrimenti
$S \rightarrow \text{while } (B) S_1$	Esegue S_1 finché B è vera
$S \rightarrow \text{do } S_1 \text{ while } (B);$	Esegue S_1 finché B è vera e <u>almeno una volta</u>
$S \rightarrow \{S_{list}\}$	Esegue in sequenza i comandi in S_{list}
$S_{list} \rightarrow \epsilon$	Non fa nulla
$S_{list} \rightarrow S S_{list}$	Esegue S e poi i comandi in S_{list}

Attributi

- $S.next$ = etichetta a cui saltare dopo che S è stato eseguito (ereditato)
- $S.code$ = codice che esegue S e salta a $S.next$

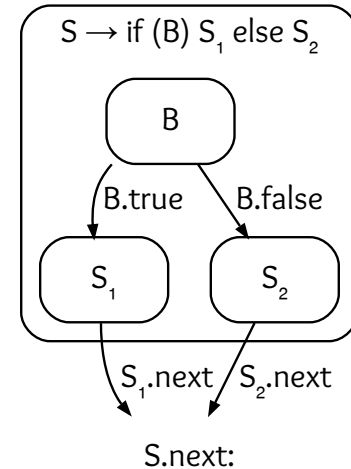
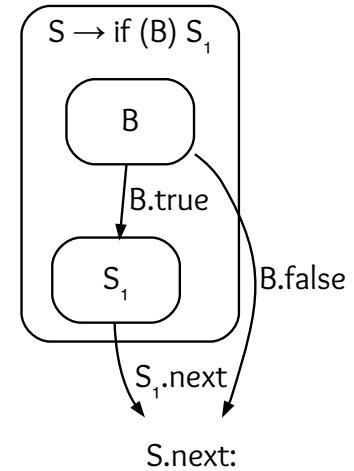
Assegnamento

Produzioni	Regole semantiche
$S \rightarrow x = E;$	$S.code = E.code$ istore &x goto S.next



Comandi condizionali

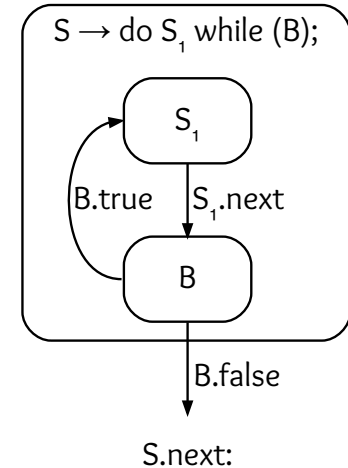
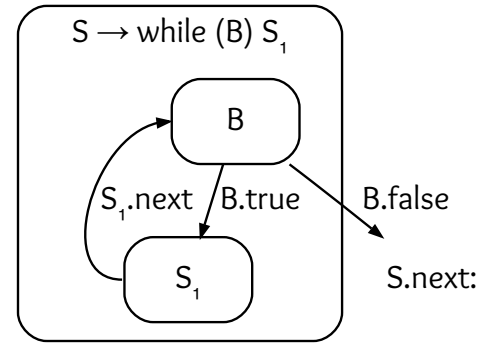
Produzioni	Regole semantiche
$S \rightarrow \text{if } (B) S_1$	$B.\text{true} = \text{newlabel}()$ $B.\text{false} = S.\text{next}$ $S_1.\text{next} = S.\text{next}$ $S.\text{code} = B.\text{code}$ $\parallel B.\text{true} : S_1.\text{code}$
$S \rightarrow \text{if } (B) S_1 \text{ else } S_2$	$B.\text{true} = \text{newlabel}()$ $B.\text{false} = \text{newlabel}()$ $S_1.\text{next} = S.\text{next}$ $S_2.\text{next} = S.\text{next}$ $S.\text{code} = B.\text{code}$ $\parallel B.\text{true} : S_1.\text{code}$ $\parallel B.\text{false} : S_2.\text{code}$



Comandi iterativi

Produzioni	Regole semantiche
$S \rightarrow \text{while } (B) S_1$	$B.\text{true} = \text{newlabel}()$ $B.\text{false} = S.\text{next}$ $S_1.\text{next} = \text{newlabel}()$ $S.\text{code} = S_1.\text{next} : B.\text{code}$ $\parallel B.\text{true} : S_1.\text{code}$
$S \rightarrow \text{do } S_1 \text{ while } (B);$	$S_1.\text{next} = \text{newlabel}()$ $B.\text{true} = \text{newlabel}()$ $B.\text{false} = S.\text{next}$ $S.\text{code} = B.\text{true} : S_1.\text{code}$ $\parallel S_1.\text{next} : B.\text{code}$

invert
ordine

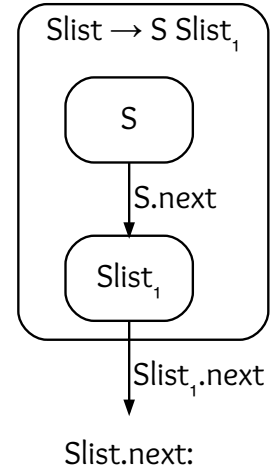
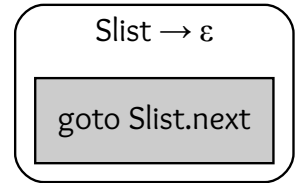


Composizione sequenziale

Produzioni	Regole semantiche
$S \rightarrow \{S_{list}\}$	$S_{list}.next = S.next$ $S.code = S_{list}.code$
$S_{list} \rightarrow \epsilon$	$S_{list}.code = \text{goto } S_{list}.next$
$S_{list} \rightarrow SS_{list1}$	$S.next = \text{newlabel}()$ $S_{list1}.next = S_{list}.next$ $S_{list}.code = S.code$ $\parallel S.next : S_{list1}.code$

Attributi

- $S_{list}.next$ = etichetta a cui saltare dopo che tutti i comandi in S_{list} sono stati eseguiti (ereditato)
- $S_{list}.code$ = codice che esegue tutti i comandi in S_{list}



Esempio: somma dei primi 10 numeri naturali

```
{  
    x = 10;  
    y = 0;  
    while (x > 0) {  
        y = x + y;  
        x = x - 1;  
    }  
}
```

```
        ldc 10  
        istore &x  
        goto L1  
L1:     ldc 0  
        istore &y  
        goto L2  
L2:       
L3:     iload &x  
        ldc 0  
        if_icmpgt L4  
        goto L0  
L4:     iload &x  
        iload &y  
        iadd  
        istore &y  
        goto L5  
L5:     iload &x  
        ldc 1  
        isub  
        istore &x  
        goto L3  
L0:     
```

Note

- L0 è l'etichetta presente nell'attributo ereditato ***S.next*** della variabile ***S*** da cui è stato generato il comando.
- Nel codice generato ci sono diversi goto ridondanti che possono essere eliminati nella fase di ottimizzazione del codice.

Esercizi

1. Scrivere le regole semantiche per la traduzione del doppio assegnamento simultaneo $S \rightarrow x, y = E_1, E_2;$, che può essere utile per scambiare il contenuto di due variabili senza usare una variabile di appoggio (es. $x, y = y, x;$). Cosa contiene la variabile m al termine dell'esecuzione del seguente programma?

```
{ m, n = 0, 1;  
  k = 10;  
  while (k > 0) {  
    m, n = n, m + n;  
    k = k - 1;  
  }  
}
```

2. Scrivere le regole semantiche per tradurre il ciclo for $S \rightarrow \mathbf{for} (S_1; B; S_2) S_3$