Strutturare il Controllo di Sequenza

Sommario: 21 marzo, 2019

- Controllo di Sequenza Nei Linguaggi Prescrittivii: Linguaggi Machine-Level, High-level e Valori modificabili
- Espressioni: Valutazione e Controllo di sequenza
- Comandi vs. Espressioni: Assegnamento
- Comandi: Controllo di sequenza
- Comandi Strutturati: goto e comandi per iterazione
- Ricorsione: Programmazione con induzione, Tail Recursion e Memoization
- Programmazione Strutturata: Antesignana delle moderne Metodologie
- Esercizi

Controllo di sequenza: Linguaggi Machine-Level Prescrittivi

- **Linguaggi Machine-Level**: La struttura di un programma è una sequenza di Statement Atomici che operano sullo Stato:

  Un frammento di programma in 2ACode (2 operandi atomici)

  ```
  Mov Lx0 , R0
  Mov Lx2 , R1
  Add R0, R1
  Mov R1, Lx1
  Mov Lx2 , R0
  Mul R0, R1
  Mov R1, Lx0
  ```

  - che usano Locazioni $\mathtt{L}_{\mathtt{x}_i}$ e registri $\mathtt{R}_i$ per Valori Modificabili
  - controllata da salti condizionati e/o incondizionati

  - La sequenza di stm atomici è:
    facile da eseguire per la MA ma difficile da scrivere.

  - Costrutti per un controllo di sequenza più astratto.

$$\mathtt{x}_0 = (\mathtt{x}_1 = \mathtt{x}_0 + \mathtt{x}_2) * \mathtt{x}_2$$

Controllo di sequenza: Linguaggi High-Level Prescrittivi

- Linguaggi Machine-Level:
- **Linguaggi High-Level**: La struttura di un programma è una sequenza di Statement Non-Atomici che:
  - operano sullo Stato ed
  - usano composizioni di statements per il controllo di sequenze correlate

    $$\texttt{Exp} ::= \texttt{Exp} * \texttt{Exp}$$
    $$\texttt{Cmd} ::= \texttt{while BExp do Cmd}$$

    Ogni termine `Exp`, `BExp`, `Cmd` definisce una sequenza (indipendente) che è controllata dal costrutto in cui occorre
- Semantica e Controllo di Sequenza:
  Controllo di Seq. definito da sem. del costrutto

Osservazione (Translation Semantics)

La Semantica ci dice:
Come stms Non Atomici sono decomposti in stm più semplici e in quale sequenza questi ultimi devono essere considerati e valutati

Controllo di sequenza: Espressioni

Espressioni sono presenti in tutti gli High-Level L.

- Semantica: In tutti i linguaggi, la sem. prevede che calcolino un **valore**, o risultino in una computazione **non-terminante**.
- Struttura:
  - Un operatore (principale) ed $\mathtt{n} \geq 0$ argomenti (a loro volta espressioni)
  - Un'invocazione di funzione applicata a $\mathtt{n} \geq 0$ argomenti (a loro volta espressioni)
  - Un termine atomico (variabile, valore atomico)
- Controllo di sequenza: segue l'ordine e il metodo di valutazione degli argomenti (stabilito dal L. di Programmazione)
  - **Ordine**: leftmost, rightmost, associatività dx o sin.
  - **Metodo**:
    - Operatori: stretto, circuito-corto (o lazy),
    - Invocazioni: trasmissione parametri

Espressioni: Ordine e Metodo di valutazione

Espressioni sono presenti in tutti gli High-Level L.

- **Ordine** di valutazione: *leftmost, rightmost, assoc. dx o sin.*
  - critico quando:
    - operatori su aritmetica finita (virgola mobile)
    - effetti laterali (modifica di variabili):
      Es. (x=3)+(y=x)
- **Metodo** di valutazione: *stretto, circuito-corto*
  - critico quando:
    - Argomento indefinito.
      Esempio. Sia:
      $$\textbf{int}\,\mathtt{p}(\,\textbf{int}\,\mathtt{x},\,\textbf{int}\,\mathtt{y})\{\textbf{if}(\mathtt{x}>0)\textbf{return}\,\mathtt{x};\textbf{else return}\,\mathtt{x}+\mathtt{y}\}$$
      - "e" valuta indefinito. Allora:
        $$\mathtt{p}(15,\mathtt{e})$$
        calcola 15 se non stretta sul primo argomento
      - "e_1" o "e_2" valuta indefinito e l'altra valuta **true**.
        $$e_1\ \textbf{Or}\ e_2$$
        calcola **true** se **Or** è valutata a circuito-corto

Assegnamento: Espressione o Comando

Assegnamento

- è un'espressione in alcuni linguaggi (C, Java).
- è un comando in altri (Pascal).
- Semantica:
  - In tutti i casi modica un valore modificabile
  - Linguaggi differenti possono usare differenti ordini di valutazione degli argomenti.
  - conducendo a sequenze di valutazione e computazioni completamente diverse

```
b = 0;
a[f(b)] = a[f(b)]+1
```

per f così definito (e non locale "b" quella dell'assegnamento sopra):

```
int f(int x){
  if (x==0){b=1; return 1;}
  else return 2;
}
```

  - usare ora ordine leftmost ora rightmost.

Assegnamento: l-value e r-value

Assegnamento:

- Gli argomenti di un assegnamento devono essere un valore modificabile (l-value) e un valore memorizzabile (r-value)
  - valore modificabile = Locazione di memoria
  - valore memorizzabile = valore assegnabile ad una locazione nella memoria
  - valore denotabile = valore associabile ad un identificatore nell'ambiente
  - valore esprimibile = valore che può essere calcolato con un'espressione (non singola variabile)
- Alcune espressioni possono essere valutate per ottenere un l-value oppure un r-value

```
a[f(b)] = a[f(b)] +1
```

Controllo di sequenza: Comandi

Comandi sono presenti in tutti i Linguaggi Prescrittivi (Macchina, Imperativi,...)

- Semantica: In tutti i linguaggi, la semantica prevede che modifichino la memoria e/o lo stato (effetti laterali).
- Struttura. Varie classi di comandi:
  - Controllo di Sequenza esplicito
  - Condizionali (o di selezione)
  - Iterativi

Comandi di Sequenza Esplicito

Introducono una sequenza di C, o alterano la sequenza corrente.

- Includono:
  - Comando Sequenza: $\texttt{C}_1\texttt{;}\,\texttt{C}_2$
  - Blocco in-Line
  - goto, break, continue, return.
- Comando Sequenza e Blocco in-Line sono presenti in tutti i Linguaggi con comandi ed hanno una semantica composizionale ottenuta dalla naturale composizione della semantica dei componenti.
- goto: Iterazione non strutturata
  - oscuro e poco espressivo
  - può sempre essere evitato a favore di iterazione strutturata
  - va evitato
- break, continue, return
  - Utilizzati in C per uscita (non strutturata??) da blocchi.

Comandi di Condizionali

Discriminano tra due o più sequenze alternative

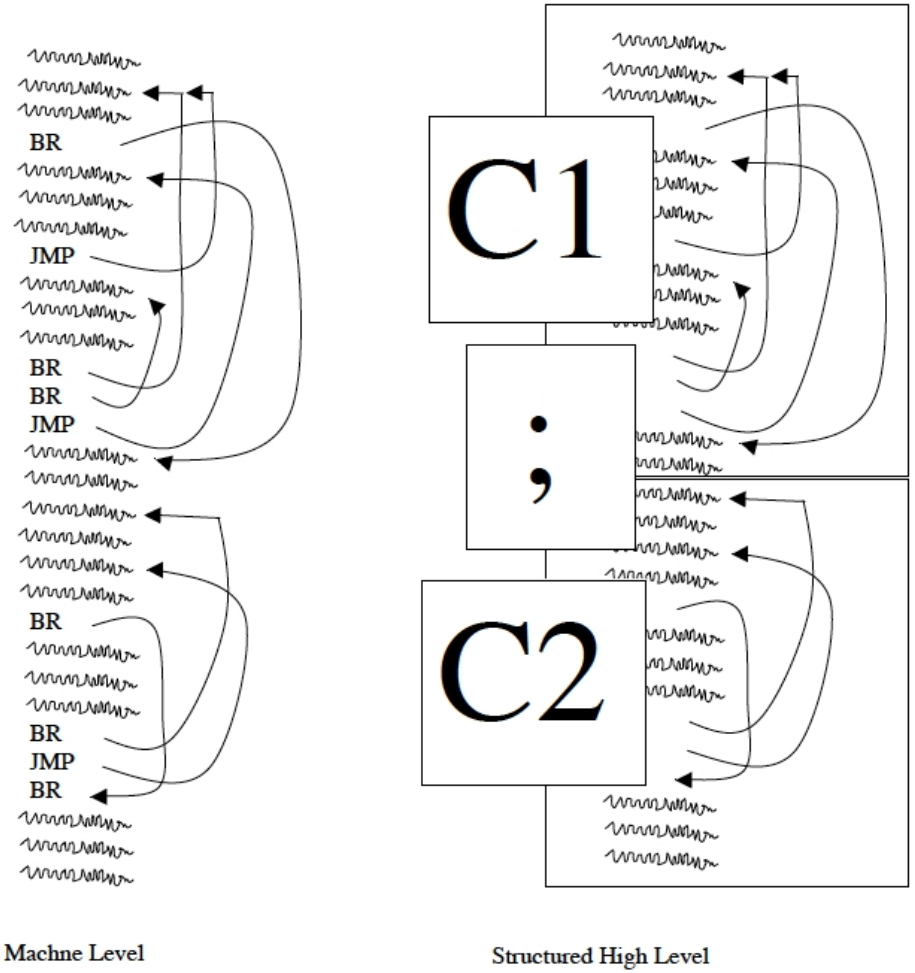
- Includono:
  - if-then-else e if-then
    - con struttura e comportamento analogo in tutti i linguaggi
  - case e switch
    - con struttura simile ma comportamento, a volte, diverso:
      $\texttt{case 2 of 1 : C}_1\texttt{; 2 : C}_2\texttt{; 3 : C}_3\texttt{; 4 : C}_4\texttt{; end; C;}$
      $\texttt{switch (2)\{case 1 : C}_1\texttt{; case 2 : C}_2\texttt{; case 3 : C}_3\texttt{; case 4 : C}_4\texttt{;\}C;}$
      conducono alla sequenza:
      $\texttt{C}_2\texttt{; C;}$
      $\texttt{\{C}_2\texttt{; C}_3\texttt{; C}_4\texttt{;\} C;}$
      rispettivamente.

Comandi Iterativi (strutturati)

Incapsulano ed evidenziano la sequenza da iterare

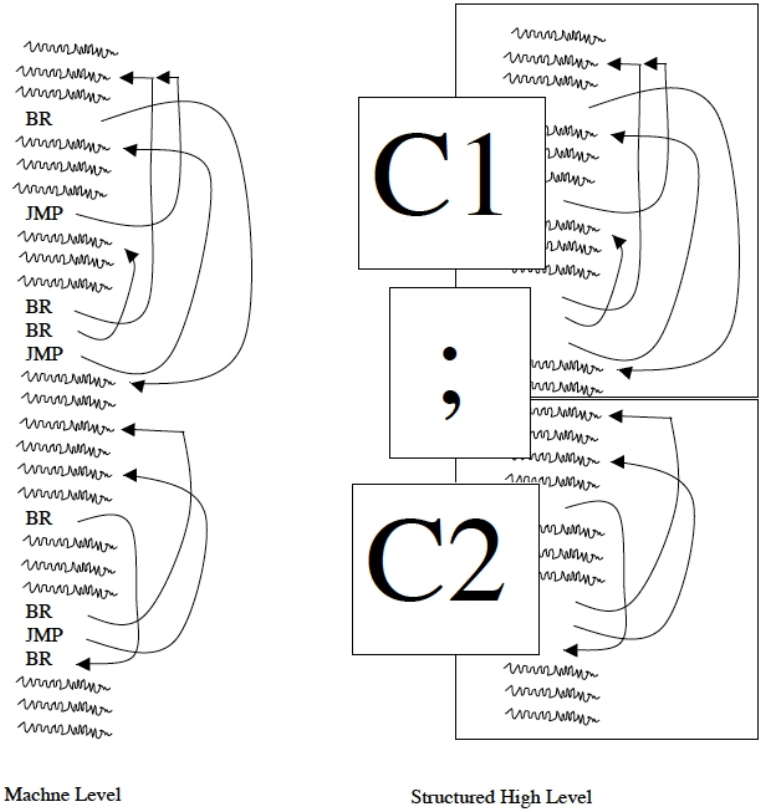
- Si differenziano in questo dall'iterazione non strutturata basata sul goto
- Si suddividono in:
  - iterazione indeterminata: while, repeat, for (del C)
  - iterazione determinata: for del Pascal
- Iterazione indeterminata: **while** `E` **do** `C`
  - il numero di iterazioni può non essere determinato utilizzando lo stato iniziale e l'espressione di controllo `E`
  - l'iterazione può anche non terminare
  - **while** `(x>1 and x<10)` **do** `x = 5;`
- Iterazione determinata: **for** `i` $=$ $\mathtt{E}_{\mathtt{inizio}}$ **to** $\mathtt{E}_{\mathtt{fine}}$ **do** `C`
  - il numero di iterazioni è: $\mathtt{n} = \mathtt{v}_{\mathtt{E}_{\mathtt{fine}}} - \mathtt{v}_{\mathtt{E}_{\mathtt{inizio}}}$

Costrutto Strutturato: Sequenze chiuse



- Possiamo sempre "estendere" una sequenza atomica con JMP e BR in una sequenza atomica equivalente formata di sole sezioni "chiuse"

Costrutto e Programma Strutturato



- **Costrutto Strutturato**: È un costrutto non atomico che definisce una sequenza atomica "chiusa", ovvero ha un unico punto di ingresso (il primo atomo) ed un unico punto di uscita (l'ultimo atomo).
- Linguaggio, **Programma Strutturato**: Le funzioni calcolabili possono essere espresse con programmi che utilizzano solo costrutti strutturati (Bohm-Jacopini, 1966)

Costrutti non strutturati

- **Costrutto Strutturato**: È un costrutto non atomico che definisce una sequenza atomica "chiusa", ovvero ha ha un unico punto di ingresso (il primo atomo) ed un unico punto di uscita (l'ultimo atomo).

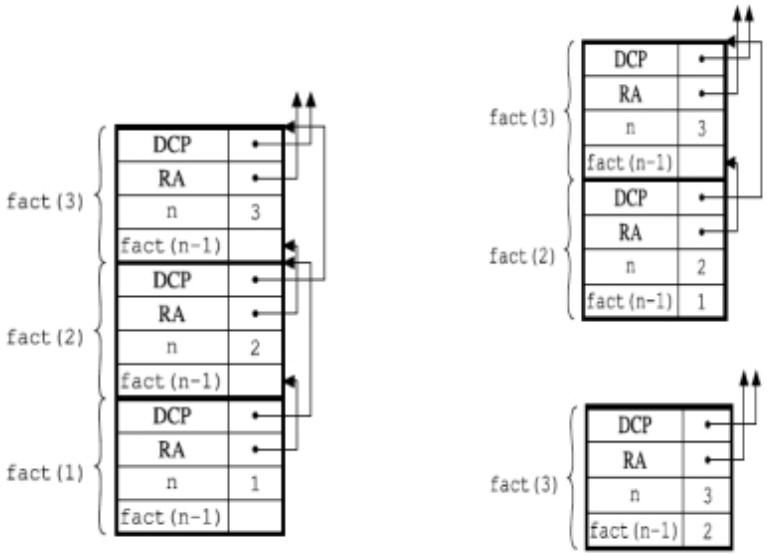
- **Linguaggio/Programma Strutturato**: Le funzioni calcolabili possono essere espresse con programmi che utilizzano solo costrutti strutturati (Bohm-Jacopini, 1966)

- Rispondiamo:
  - Quale dei seguenti costrutti potrebbe essere usato all'interno di un programma strutturato e perchè?
    (a) Goto; (b) break; (c) continue; (d) return
  - Il costrutto Switch del C è un costrutto strutturato?

Ricorsione

- Algoritmi definiti in modo induttivo
- Introdotta attraverso Procedure/Funzioni
- Possono richiedere l'allocazione di una grande quantità di AR

```
int fact(int n){
        if (n<0) return 0;
         else if (n==0) return 1;
             else return n * fact(n − 1);
 }
```



- Dire chi sono i campi: DCP, RA presenti negli AR e di quale campo fa parte l'entry "fact(n-1)..."