



Sintassi e semantica operativa dei linguaggi di programmazione

Paolo Costa

paolo.costa@polimi.it

Parte II



SIMPLESEM-2

```
program pippo
var a,b,c:integer
  procedure p(procedure x)
    var a,b,d:integer;
    procedure q
      var b,e:integer;
      .....
    end q;
    if c=0 then
      begin
        c:=c+1;
        x(q);
      end
    else x;
  end p;
  .....
c:=0;
p(p);
end;
```

schizzare lo stato della macchina astratta, tracciando link statici e dinamici, fino a quando q viene chiamata per la prima volta.



Catena chiamate

- ✘ Pippo pone $c=0$ e chiama $p(p)$
- ✘ In $p(p)$ si entra nel ramo then dell'if ponendo quindi $c=1$ e chiamando $p(q)$
- ✘ In $p(q)$ si entra nel ramo else dell'if quindi si chiama q
- ✘ Quindi la sequenza delle chiamate e':
 $\text{pippo} \rightarrow p(p) \rightarrow p(q) \rightarrow q$



Annidamento statico

pippo



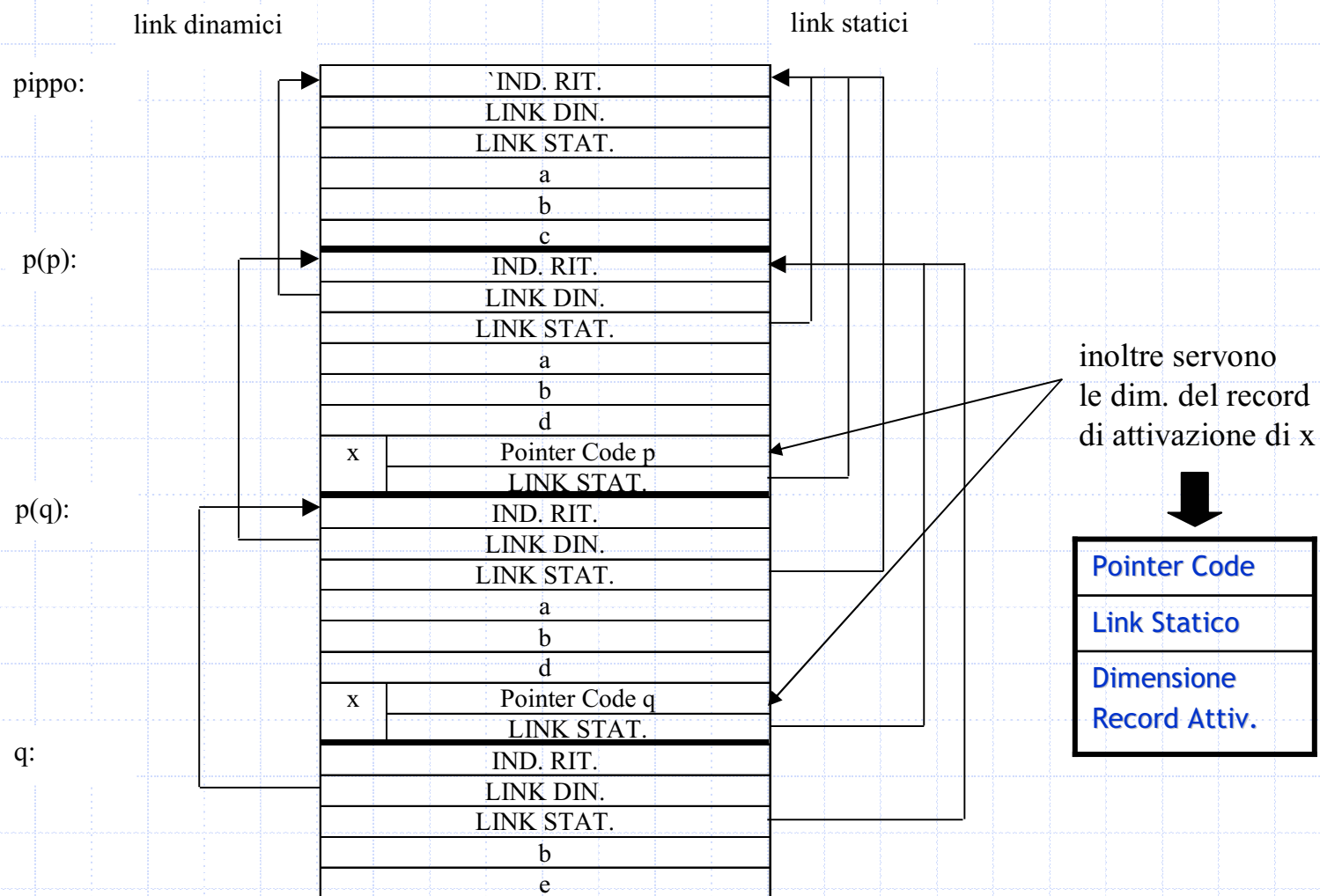
p



q



Soluzione





SIMPLESEM-3

```
program pippo
var l,m,n:real;
  procedure q(procedure w)
    procedure p
      var r,s:real;
      .....
    end
    if n>0 then w
    else begin
      n:=n+1.77;
      w(p);
    end
    .....
  n:= -0.5;
  q(q);
end
```

schizzare lo stato della macchina astratta, tracciando link statici e dinamici, fino alla chiamata di w effettuata nel ramo then della procedura q. Si scriva inoltre come viene tradotto l'accesso alla variabile n in termini della coppia (distanza, offset) [scope statico].



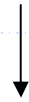
Sequenza chiamate

- ✘ Pippo pone $n = -0.5$ e chiama $q(q)$
- ✘ In $q(q)$ si entra nel ramo else dell'if ponendo quindi $n = -0.5 + 1.77 = 1.27$ e chiamando $q(p)$
- ✘ In $q(p)$ si entra nel ramo then dell'if quindi viene chiamata p
- ✘ Quindi la sequenza delle chiamate e':
 $\text{pippo} \rightarrow q(q) \rightarrow q(p) \rightarrow p$



Annidamento statico

pippo



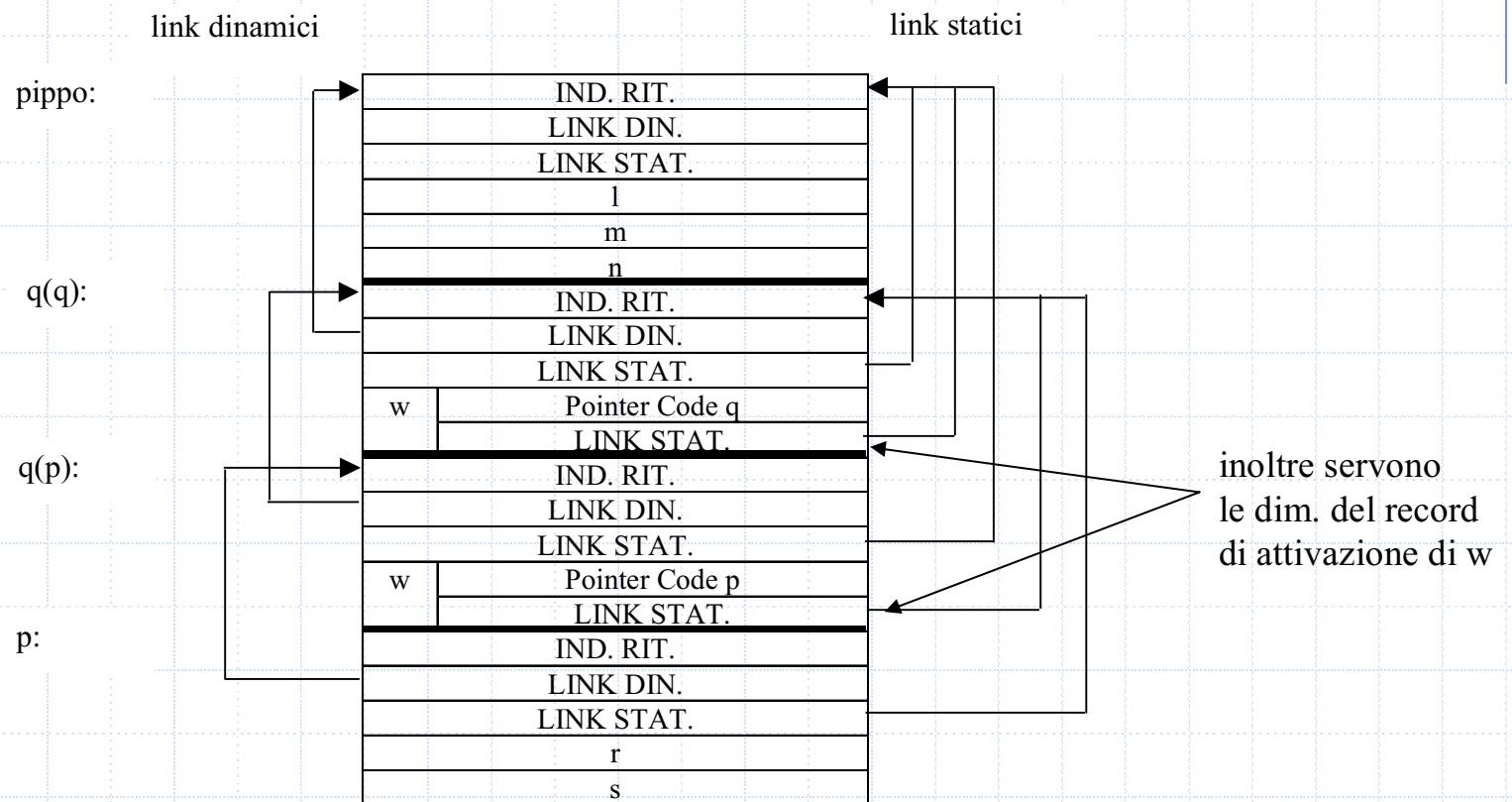
q



p



Soluzione





Accesso alla variabile n

- ✘ Accesso alla variabile n da pippo: $n = \langle 0, 5 \rangle$
- ✘ Accesso alla variabile n per la prima chiamata di q: $n = \langle 1, 5 \rangle$
- ✘ Accesso alla variabile n per la seconda chiamata di q: $n = \langle 1, 5 \rangle$ (valgono regole di scope statico)



Domanda...

Si dica, giustificando brevemente la risposta, se la seguente affermazione è vera o falsa:

Se un programma C non fa uso né di blocchi né di sottoprogrammi annidati (ossia se consiste solo del main e di una lista di sottoprogrammi dichiarati tutti allo stesso livello del main) la semantica determinata dalla catena statica equivale alla semantica della catena dinamica.



Risposta:

Falsa. Infatti se si adotta la regola della catena statica le uniche variabili non locali usabili dai vari sottoprogrammi sono quelle globali; si invece si adotta la regola della catena dinamica un generico sottoprogramma P "cercherà" una variabile non locale x nel record di attivazione del sottoprogramma Q che lo ha chiamato e via via più in basso nella pila.



Passaggio di parametri

```
1. program esempio;  
2.   var x:integer  
3.     procedure p(y:integer)  
4.     begin  
5.         x:=5;  
6.         y:=y+x;  
7.     end;  
8. begin  
9.     x:=10;  
10.    p(x);  
11.    write(x);  
12. end
```

Analizzare il valore stampato a seconda della tipologia di passaggio di parametri considerata



By reference

(parametri formali e attuali dividono la stessa area di memoria)

# Riga	x	y
9	10	
3	10	10
5	5	5
6	10	10
11	10	

By value

(par. attuale copiato in quello formale che sostituisce la var. locale nella procedura)

# Riga	x	y
9	10	
3	10	10
5	5	10
6	5	15
11	5	



By result

(par. formale copiato in quello attuale al termine della procedura)

# Riga	x	y
9	10	
3	10	0
5	5	0
6	5	5
11	5	

By value/result

# Riga	x	y
9	10	
3	10	10
5	5	10
6	5	15
11	15	



By name

(nel testo della procedura i par. formali sostituiscono quelli attuali)

# Riga	x	y
9	10	
3	10	10
5	5	5
6	10	10
11	10	