

Addendum a lezione 3

Implementazione efficiente dell'ambiente non locale con scoping statico

Piero Bonatti

Annidamento (nesting)

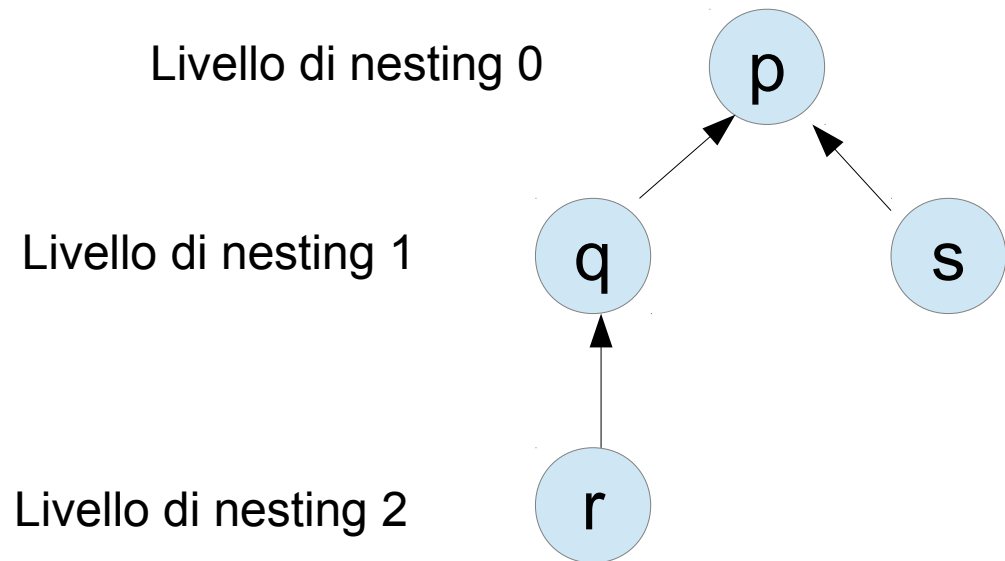
```
program p;  
  var a,b,c int;  
  
  procedure q;  
    var a,c int;  
  
    procedure r;  
      var a int;  
      ...  
    ...  
  
    procedure s;  
      var b int;  
      ...  
    ...  
  ...
```

- Una procedura q è **annidata** in un blocco b se è definita dentro b, ad esempio:
 - q e s sono annidate in p
 - r è annidata in q
- Il **livello di nesting di una procedura** è il numero di blocchi che la contengono
 - il livello di nesting di q e s è 1
 - quello di r è 2
- **L'ambiente statico non locale** di una procedura è dato dall'ambiente delle procedure in cui è innestata

Annidamento (nesting)

```
program p;  
  var a,b,c int;  
  
  procedure q;  
    var a,c int;  
  
    procedure r;  
      var a int;  
      ...  
    ...  
  
  procedure s;  
    var b int;  
    ...  
  ...
```

- L'annidamento si può rappresentare come un albero
- I livelli dell'albero di nesting corrispondono ai livelli di nesting



Le frecce indicano l'ambiente non locale

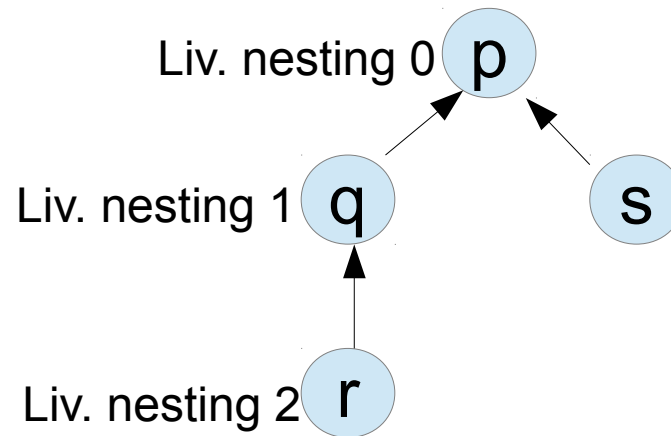
Rappresentazione variabili non locali

```
program p;
  var a,b,c int;
```

```
procedure q;
  var a,c int;
```

```
procedure r;
  var a int;
  ...
  ...
```

```
procedure s;
  var b int;
  ...
  ...
```

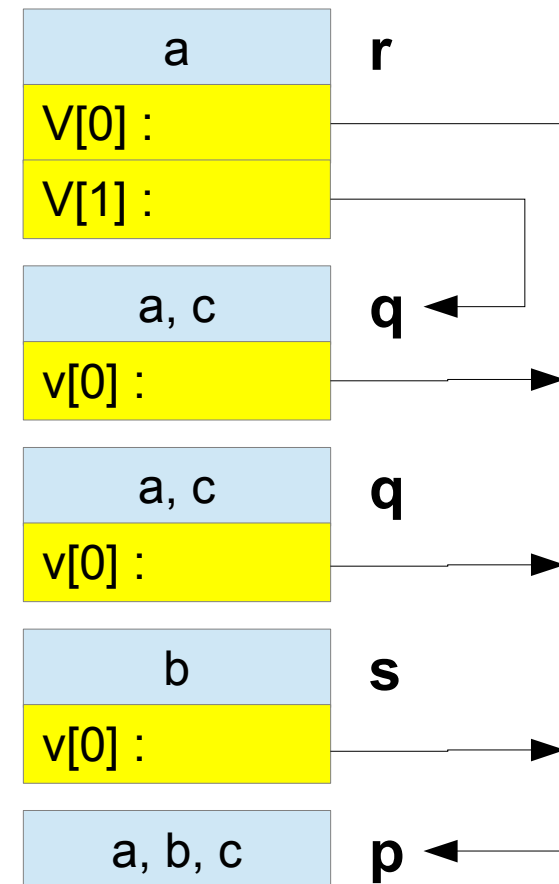


Rappresentazione variabili non locali:
(livello, offset)

Nella procedura r:
b è rappresentata da (0,2)
c è rappresentata da (1,2)

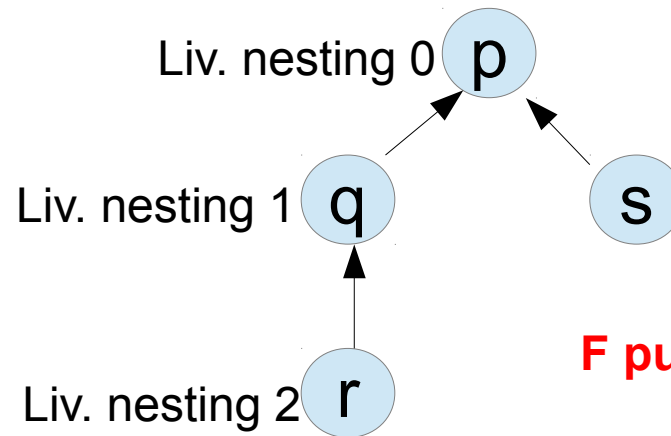
$$env(x) = v[livello] + offset$$

Stack di attivazione



Visibilità delle procedure

```
program p;  
  var a,b,c int;  
  
  procedure q;  
    var a,c int;  
  
    procedure r;  
      var a int;  
      ...  
    ...  
  
  procedure s;  
    var b int;  
    ...  
  ...
```



(Ambito statico)

F può chiamare G se:

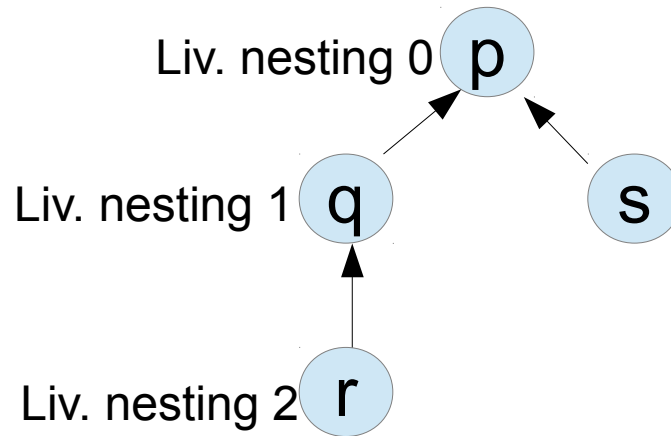
- G è definita in F
- G è definita in uno dei blocchi che contiene F

quindi:

F e G hanno sempre una parte di ambiente non locale in comune

Mantenimento del vettore degli ambienti non locali

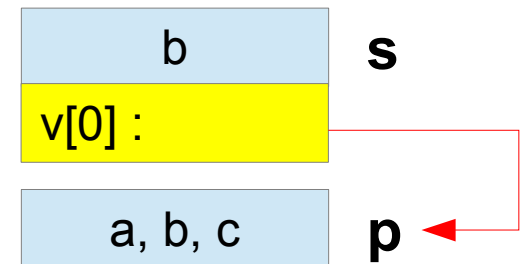
```
program p;  
  var a,b,c int;  
  
  procedure q;  
    var a,c int;  
  
    procedure r;  
      var a int;  
      ...  
    ...  
  
  procedure s;  
    var b int;  
    ...  
  ...
```



Chiamata a procedure definite localmente:

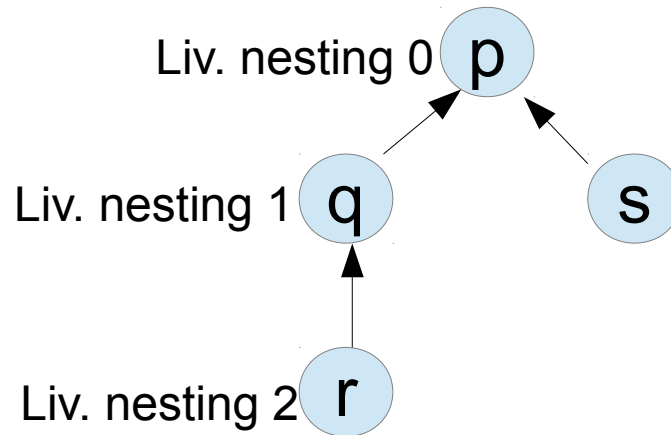
Esempio 1: p chiama s:

- Aumenta liv. di nesting
- Aggiungere elem. a v[]



Mantenimento del vettore degli ambienti non locali

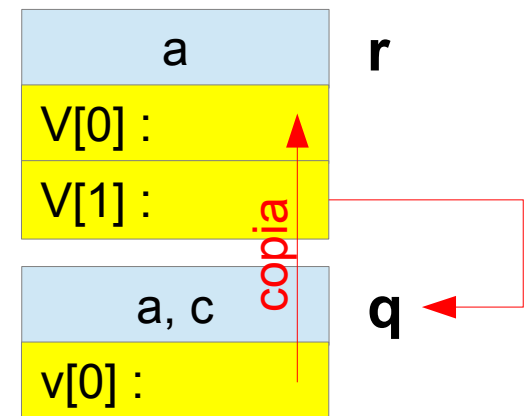
```
program p;  
  var a,b,c int;  
  
  procedure q;  
    var a,c int;  
  
    procedure r;  
      var a int;  
      ...  
    ...  
  
  procedure s;  
    var b int;  
    ...  
  ...
```



Chiamata a procedure definite localmente:

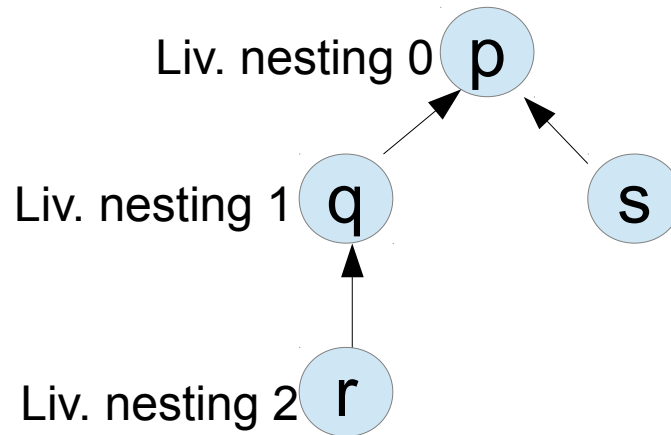
Esempio 2: q chiama r:

- Aumenta liv. di nesting
- Aggiungere elem. a v[]
- Copiare gli altri elementi



Mantenimento del vettore degli ambienti non locali

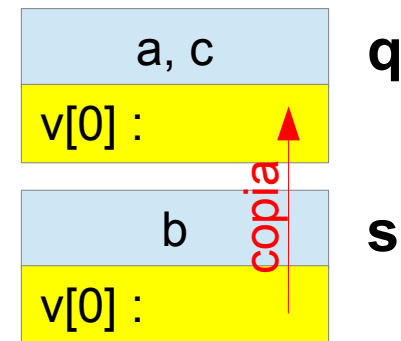
```
program p;  
  var a,b,c int;  
  
  procedure q;  
    var a,c int;  
  
    procedure r;  
      var a int;  
      ...  
    ...  
  
  procedure s;  
    var b int;  
    ...  
  ...
```



Chiamata a procedure definite **esternamente**:

Esempio 1: s chiama q:

- stesso liv. di nesting
- Copiare gli elementi di v[]



Mantenimento del vettore degli ambienti non locali

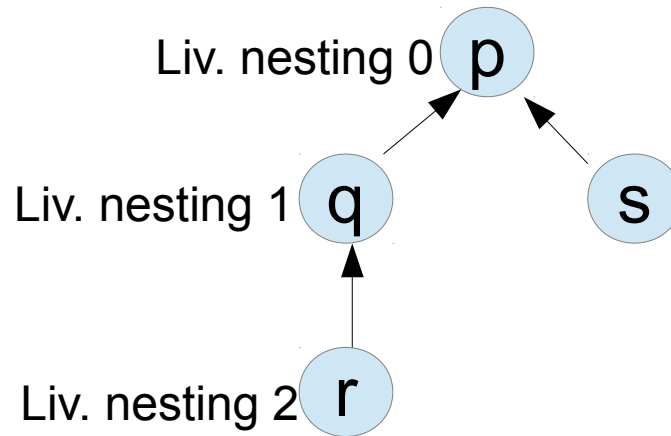
```

program p;
  var a,b,c int;

  procedure q;
  var a,c int;

    procedure r;
    var a int;
    ...
  ...

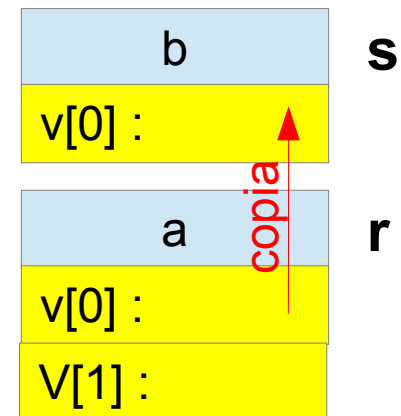
  procedure s;
  var b int;
  ...
  ...
  
```



Chiamata a procedure definite esternamente:

Esempio 2: r chiama s:

- il liv. di nesting **decre**sce
- Copiare solo gli elementi di livello *minore o uguale* a quello di s
- Struttura ad albero + scoping statico garantiscono sempre di trovare l'ambiente non locale della procedura chiamata nel record del chiamante



Proprietà di questa implementazione

- Accesso alle variabili in tempo costante
 - Accesso a vettore + somma del puntatore ivi contenuto e dell'offset
 - Indipendente dai livelli di nesting
 - Calcolo supportato direttamente dalle istruzioni macchina
- Creazione record di attivazione lineare nel livello di nesting (copia degli elementi di $v[]$)
 - Indipendente dall'esecuzione
 - Dipende solo dal testo del sorgente
- Tempo **costante**
- L'implementazione naïve richiederebbe a run time di percorrere le liste di puntatori all'ambiente non locale
 - Accesso alle variabili in tempo **lineare** nel livello di nesting
 - Creazione dei record di attivazione in tempo **lineare** nel livello di nesting