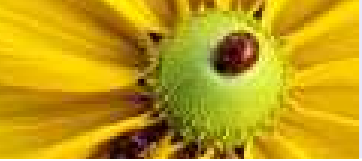


Linguaggi di Programmazione I – Lezione 3

Prof. Marcello Sette
<mailto://marcello.sette@gmail.com>
<http://sette.dnsalias.org>

16 marzo 2010



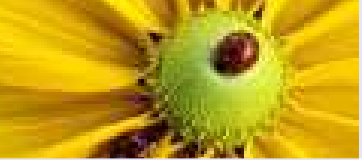
Panoramica della lezione

Procedure come astrazioni

Record di attivazione

Propagazione dei data object

Bibliografia



Procedure come astrazioni

Procedure
Astrazione
procedurale

Dichiarazione

Invocazione di . . .

Ambiente di . . .

Esempio

Record di attivazione

Propagazione dei
data object

Bibliografia

Procedure come astrazioni



Procedure

Procedure come
astrazioni

Procedure

Astrazione
procedurale

Dichiarazione

Invocazione di . . .

Ambiente di . . .

Esempio

Record di attivazione

Propagazione dei
data object

Bibliografia

Procedure sono astrazioni di parti di programma in unità di esecuzione più piccole, come enunciati o espressioni, in modo da nascondere i dettagli irrilevanti ai fini del loro (ri-)uso.



Procedure

Procedure come
astrazioni

Procedure

Astrazione
procedurale

Dichiarazione

Invocazione di . . .

Ambiente di . . .

Esempio

Record di attivazione

Propagazione dei
data object

Bibliografia

Procedure sono astrazioni di parti di programma in unità di esecuzione più piccole, come enunciati o espressioni, in modo da nascondere i dettagli irrilevanti ai fini del loro (ri-)uso.

Vantaggi:

- Programmi più semplici da scrivere, leggere o modificare; suddivisione dei compiti in ogni brano di programma; progettazione top-down.



Procedure

Procedure come
astrazioni

Procedure

Astrazione
procedurale

Dichiarazione

Invocazione di . . .

Ambiente di . . .

Esempio

Record di attivazione

Propagazione dei
data object

Bibliografia

Procedure sono astrazioni di parti di programma in unità di esecuzione più piccole, come enunciati o espressioni, in modo da nascondere i dettagli irrilevanti ai fini del loro (ri-)uso.

Vantaggi:

- Programmi più semplici da scrivere, leggere o modificare; suddivisione dei compiti in ogni brano di programma; progettazione top-down.
- Unità di programmi indipendenti o con dipendenze ben specificate a livello più alto.



Procedure

Procedure come astrazioni

Procedure

Astrazione procedurale

Dichiarazione

Invocazione di . . .

Ambiente di . . .

Esempio

Record di attivazione

Propagazione dei data object

Bibliografia

Procedure sono astrazioni di parti di programma in unità di esecuzione più piccole, come enunciati o espressioni, in modo da nascondere i dettagli irrilevanti ai fini del loro (ri-)uso.

Vantaggi:

- Programmi più semplici da scrivere, leggere o modificare; suddivisione dei compiti in ogni brano di programma; progettazione top-down.
- Unità di programmi indipendenti o con dipendenze ben specificate a livello più alto.
- Riutilizzabilità di brani di programmi; riduzione errori.



Astrazione procedurale

Procedure come
astrazioni

Procedure
Astrazione
procedurale

Dichiarazione

Invocazione di . . .

Ambiente di . . .

Esempio

Record di attivazione

Propagazione dei
data object

Bibliografia

- Se si distinguono come unità di esecuzione, in ordine crescente di complessità, *espressioni*, *enunciati*, *blocchi*, *programmi*, allora si definisce



Astrazione procedurale

Procedure come
astrazioni

Procedure
Astrazione
procedurale

Dichiarazione

Invocazione di . . .

Ambiente di . . .

Esempio

Record di attivazione

Propagazione dei
data object

Bibliografia

- Se si distinguono come unità di esecuzione, in ordine crescente di complessità, *espressioni*, *enunciati*, *blocchi*, *programmi*, allora si definisce
- **astrazione procedurale** la rappresentazione di una unità di esecuzione attraverso un'altra unità più semplice.



Astrazione procedurale

Procedure come
astrazioni

Procedure
Astrazione
procedurale

Dichiarazione

Invocazione di . . .

Ambiente di . . .

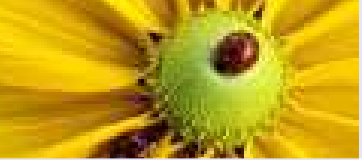
Esempio

Record di attivazione

Propagazione dei
data object

Bibliografia

- Se si distinguono come unità di esecuzione, in ordine crescente di complessità, *espressioni*, *enunciati*, *blocchi*, *programmi*, allora si definisce
- **astrazione procedurale** la rappresentazione di una unità di esecuzione attraverso un'altra unità più semplice.
- In pratica è la rappresentazione di un blocco attraverso un enunciato o una espressione.



Dichiarazione di procedura

Procedure come
astrazioni

Procedure
Astrazione
procedurale

Dichiarazione

Invocazione di . . .

Ambiente di . . .

Esempio

Record di attivazione

Propagazione dei
data object

Bibliografia

- Causa la generazione di un oggetto analogo al Type Object (anche se in questo corso non esaminato).

Dichiarazione di procedura

Procedure come astrazioni

Procedure
Astrazione
procedurale

Dichiarazione

Invocazione di . . .

Ambiente di . . .

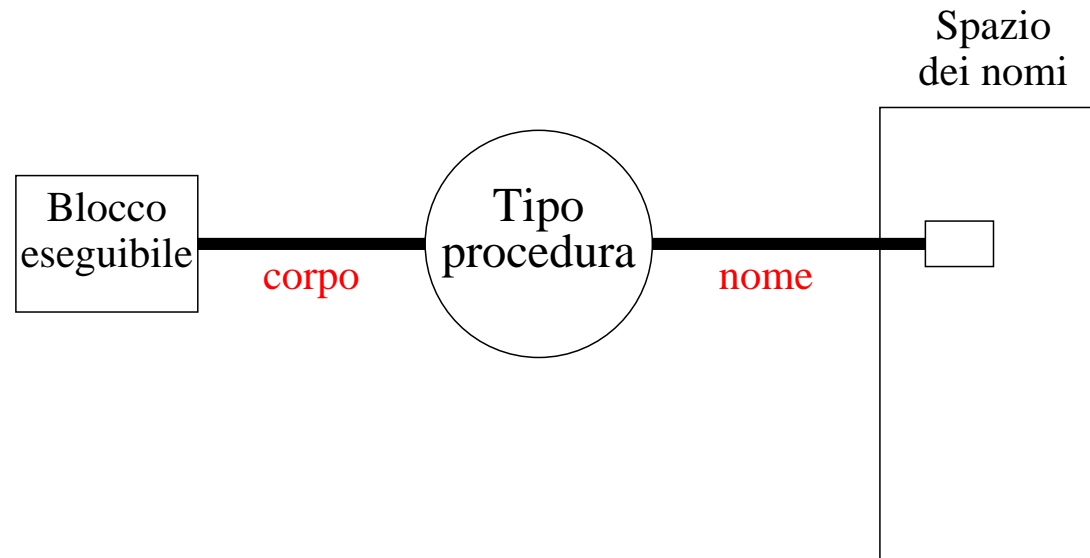
Esempio

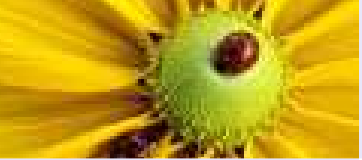
Record di attivazione

Propagazione dei
data object

Bibliografia

- Causa la generazione di un oggetto analogo al Type Object (anche se in questo corso non esaminato).
- Il processo avviene durante la compilazione.





Invocazione di una procedura

Procedure come
astrazioni

Procedure
Astrazione
procedurale

Dichiarazione

Invocazione di . . .

Ambiente di . . .

Esempio

Record di attivazione

Propagazione dei
data object

Bibliografia

- Causa la generazione di un oggetto analogo al Data Object.

Invocazione di una procedura

Procedure come astrazioni

Procedure
Astrazione
procedurale

Dichiarazione

Invocazione di . . .

Ambiente di . . .

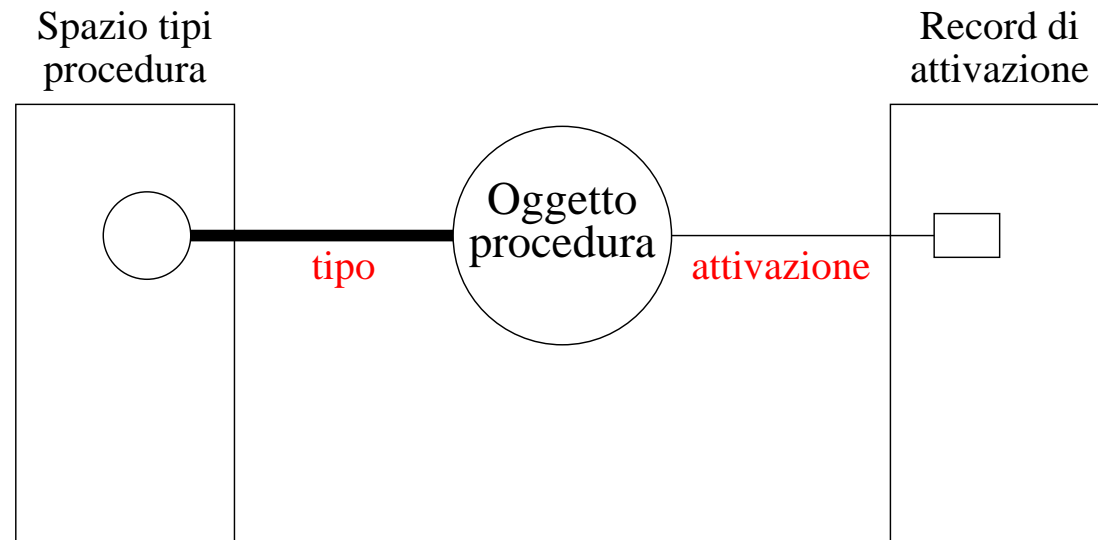
Esempio

Record di attivazione

Propagazione dei
data object

Bibliografia

- Causa la generazione di un oggetto analogo al Data Object.
- Il processo avviene durante l'esecuzione, nel momento in cui c'è l'invocazione della procedura.
- Ogni invocazione diversa della stessa procedura causa la generazione di un nuovo "oggetto procedura" con lo stesso legame di tipo, ma con diverso record di attivazione.





Ambiente di esecuzione

[Procedure come astrazioni](#)

[Procedure Astrazione procedurale](#)

[Dichiarazione](#)

[Invocazione di . . .](#)

[Ambiente di . . .](#)

[Esempio](#)

[Record di attivazione](#)

[Propagazione dei data object](#)

[Bibliografia](#)

Analogamente a quanto avveniva per un blocco (di cui la procedura è astrazione), il **record di attivazione** rappresenta l'intero ambiente di esecuzione di una procedura.



Ambiente di esecuzione

[Procedure come astrazioni](#)

[Procedure Astrazione procedurale](#)

[Dichiarazione](#)

[Invocazione di . . .](#)

[Ambiente di . . .](#)

[Esempio](#)

[Record di attivazione](#)

[Propagazione dei data object](#)

[Bibliografia](#)

Analogamente a quanto avveniva per un blocco (di cui la procedura è astrazione), il **record di attivazione** rappresenta l'intero ambiente di esecuzione di una procedura. Esso consiste di solito in:

1. ambiente locale (tutti i data object che sono definiti all'interno della procedura);



Ambiente di esecuzione

Procedure come
astrazioni

Procedure
Astrazione
procedurale

Dichiarazione

Invocazione di . . .

Ambiente di . . .

Esempio

Record di attivazione

Propagazione dei
data object

Bibliografia

Analogamente a quanto avveniva per un blocco (di cui la procedura è astrazione), il **record di attivazione** rappresenta l'intero ambiente di esecuzione di una procedura. Esso consiste di solito in:

1. ambiente locale (tutti i data object che sono definiti all'interno della procedura);
2. ambiente non locale (tutti i data object la cui definizione è propagata da altre procedure);



Ambiente di esecuzione

Procedure come
astrazioni

Procedure
Astrazione
procedurale

Dichiarazione

Invocazione di . . .

Ambiente di . . .

Esempio

Record di attivazione

Propagazione dei
data object

Bibliografia

Analogamente a quanto avveniva per un blocco (di cui la procedura è astrazione), il **record di attivazione** rappresenta l'intero ambiente di esecuzione di una procedura. Esso consiste di solito in:

1. ambiente locale (tutti i data object che sono definiti all'interno della procedura);
2. ambiente non locale (tutti i data object la cui definizione è propagata da altre procedure);
3. ambiente dei parametri (contiene informazioni sui dati che sono passati [d]alla procedura).



Ambiente di esecuzione

Procedure come
astrazioni

Procedure
Astrazione
procedurale

Dichiarazione

Invocazione di . . .

Ambiente di . . .

Esempio

Record di attivazione

Propagazione dei
data object

Bibliografia

Analogamente a quanto avveniva per un blocco (di cui la procedura è astrazione), il **record di attivazione** rappresenta l'intero ambiente di esecuzione di una procedura. Esso consiste di solito in:

1. ambiente locale (tutti i data object che sono definiti all'interno della procedura);
2. ambiente non locale (tutti i data object la cui definizione è propagata da altre procedure);
3. ambiente dei parametri (contiene informazioni sui dati che sono passati [d]alla procedura).

Ogni volta che una procedura viene invocata il suo record di attivazione viene aggiunto al cosiddetto **stack di esecuzione**. Sul top dello stack c'è sempre il record relativo alla procedura correntemente in esecuzione.

Alla terminazione della procedura, il record di attivazione viene rimosso dallo stack.



Procedure come
astrazioni

Procedure
Astrazione
procedurale

Dichiarazione

Invocazione di . . .

Ambiente di . . .

Esempio

Record di attivazione

Propagazione dei
data object

Bibliografia

Stack esecuzione

p: prima di chiamare s

p

```
program p;  
var i;  
  
  procedure q;  
  begin  
    ⋮  
  end;  
  
  procedure r;  
  begin  
    i := i - 1;  
    if i > 0 then  
      r  
    else  
      q  
    end;  
  
  procedure s;  
  begin  
    r  
    q  
  end;  
  
begin  
  ○ i := 2;  
  ● s  
end.
```

Procedure come
astrazioni

Procedure
Astrazione
procedurale

Dichiarazione

Invocazione di . . .

Ambiente di . . .

Esempio

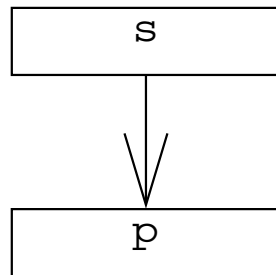
Record di attivazione

Propagazione dei
data object

Bibliografia

Stack esecuzione

s: prima di chiamare r



```
program p;  
var i;  
  
  procedure q;  
  begin  
    ⋮  
  end;  
  
  procedure r;  
  begin  
    i := i-1;  
    if i > 0 then  
      r  
    else  
      q  
    end;  
  end;  
  
  procedure s;  
  begin  
    r  
    q  
  end;  
  
begin  
  i := 2;  
  s  
end.
```

Esempio

Procedure come
astrazioni

Procedure
Astrazione
procedurale

Dichiarazione

Invocazione di . . .

Ambiente di . . .

Esempio

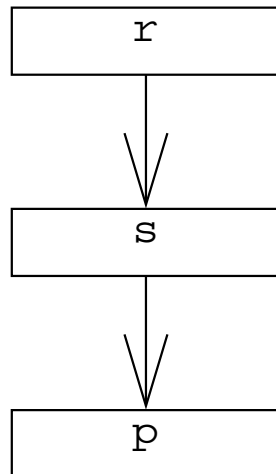
Record di attivazione

Propagazione dei
data object

Bibliografia

Stack esecuzione

r: prima di chiamare r



```
program p;  
var i;  
  
  procedure q;  
  begin  
    ⋮  
  end;  
  
  procedure r;  
  begin  
    i := i - 1;  
    if i > 0 then  
      r  
    else  
      q  
    end;  
  
  procedure s;  
  begin  
    r;  
    q  
  end;  
  
begin  
  i := 2;  
  
  s  
  
end.
```

Esempio

Procedure come
astrazioni

Procedure
Astrazione
procedurale

Dichiarazione

Invocazione di . . .

Ambiente di . . .

Esempio

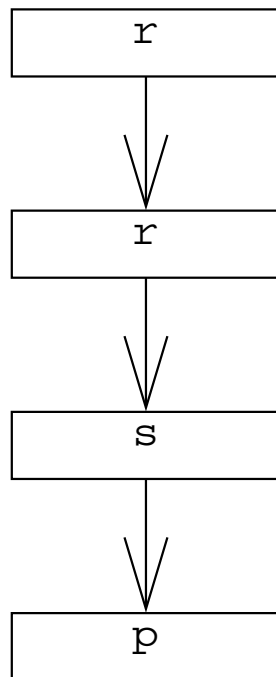
Record di attivazione

Propagazione dei
data object

Bibliografia

Stack esecuzione

r: prima di chiamare q



```
program p;  
var i;  
  
  procedure q;  
  begin  
    ⋮  
  end;  
  
  procedure r;  
  begin  
    i := i-1;  
    if i > 0 then  
      r  
    else  
      q  
    end;  
  
  procedure s;  
  begin  
    r  
    q  
  end;  
  
begin  
  i := 2;  
  
  s  
  
end.
```

Esempio

Procedure come
astrazioni

Procedure
Astrazione
procedurale

Dichiarazione

Invocazione di ...

Ambiente di ...

Esempio

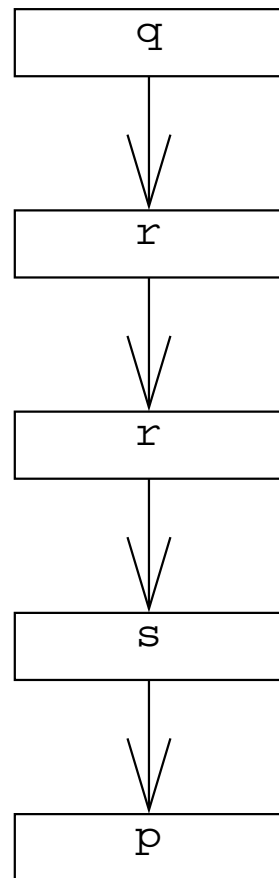
Record di attivazione

Propagazione dei
data object

Bibliografia

Stack esecuzione

q: prima di terminare



```
program p;  
var i;  
  
  procedure q;  
  begin  
    :::  
  end;  
  
  procedure r;  
  begin  
    i := i-1;  
    if i > 0 then  
      r  
    else  
      q  
    end;  
  end;  
  
  procedure s;  
  begin  
    r;  
    q;  
  end;  
  
begin  
  i := 2;  
  
  s  
  
end.
```


Esempio

Procedure come
astrazioni

Procedure
Astrazione
procedurale

Dichiarazione

Invocazione di . . .

Ambiente di . . .

Esempio

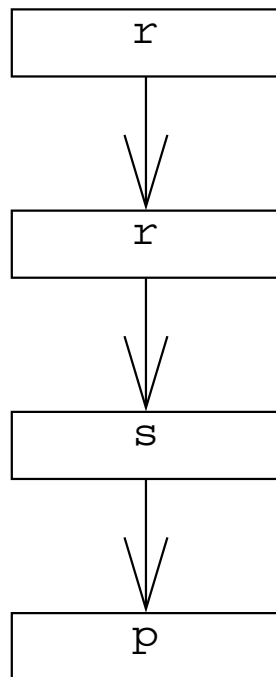
Record di attivazione

Propagazione dei
data object

Bibliografia

Stack esecuzione

r: prima di terminare



```
program p;  
var i;  
  
  procedure q;  
  begin  
    ⋮  
  end;  
  
  procedure r;  
  begin  
    i := i - 1;  
    if i > 0 then  
      r  
    else  
      q  
    end;  
  end;  
  
  procedure s;  
  begin  
    r;  
    q;  
  end;  
  
begin  
  i := 2;  
  s  
end.
```

Esempio

Procedure come
astrazioni

Procedure
Astrazione
procedurale

Dichiarazione

Invocazione di . . .

Ambiente di . . .

Esempio

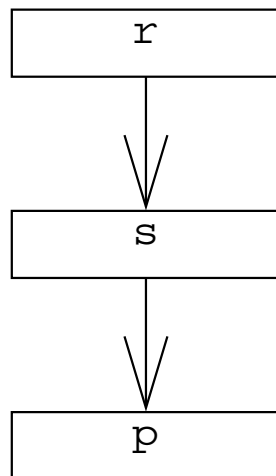
Record di attivazione

Propagazione dei
data object

Bibliografia

Stack esecuzione

r: prima di terminare



```
program p;  
var i;  
  
  procedure q;  
  begin  
    ⋮  
  end;  
  
  procedure r;  
  begin  
    i := i-1;  
    if i > 0 then  
      ○ r  
      ● q  
    end;  
  
  procedure s;  
  begin  
    r;  
    q;  
  end;  
  
begin  
  i := 2;  
  
  s  
  
end.
```

Esempio

Procedure come
astrazioni

Procedure
Astrazione
procedurale

Dichiarazione

Invocazione di . . .

Ambiente di . . .

Esempio

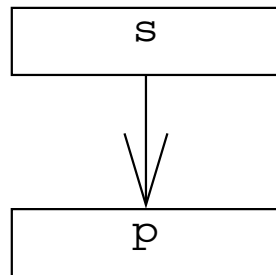
Record di attivazione

Propagazione dei
data object

Bibliografia

Stack esecuzione

s: prima di q



```
program p;  
var i;  
  
  procedure q;  
  begin  
    ⋮  
  end;  
  
  procedure r;  
  begin  
    i := i - 1;  
    if i > 0 then  
      r  
    else  
      q  
    end;  
  end;  
  
  procedure s;  
  begin  
    r  
    q  
  end;  
  
begin  
  i := 2;  
  
  s  
  
end.
```

Esempio

Procedure come
astrazioni

Procedure
Astrazione
procedurale

Dichiarazione

Invocazione di ...

Ambiente di ...

Esempio

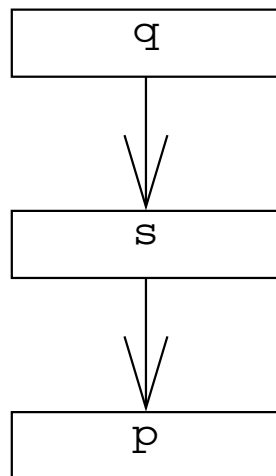
Record di attivazione

Propagazione dei
data object

Bibliografia

Stack esecuzione

q: prima di terminare



```
program p;  
var i;  
  
  procedure q;  
  begin  
    :::  
  end;  
  
  procedure r;  
  begin  
    i := i-1;  
    if i > 0 then  
      r  
    else  
      q  
    end;  
  end;  
  
  procedure s;  
  begin  
    r  
    q  
  end;  
  
begin  
  i := 2;  
  s  
end.
```

Esempio

Procedure come
astrazioni

Procedure
Astrazione
procedurale

Dichiarazione

Invocazione di . . .

Ambiente di . . .

Esempio

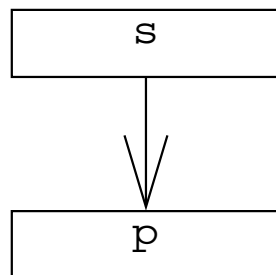
Record di attivazione

Propagazione dei
data object

Bibliografia

Stack esecuzione

s: prima di terminare



```
program p;  
var i;  
  
  procedure q;  
  begin  
    ⋮  
  end;  
  
  procedure r;  
  begin  
    i := i-1;  
    if i > 0 then  
      r  
    else  
      q  
    end;  
  end;  
  
  procedure s;  
  begin  
    r  
    q  
  end;  
  
begin  
  i := 2;  
  s  
end.
```

Esempio

Procedure come
astrazioni

Procedure
Astrazione
procedurale

Dichiarazione

Invocazione di . . .

Ambiente di . . .

Esempio

Record di attivazione

Propagazione dei
data object

Bibliografia

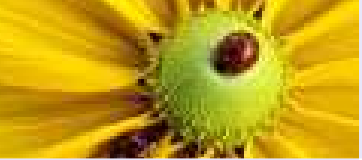
Stack esecuzione

p: prima di terminare

p

```
program p;  
var i;  
  
  procedure q;  
  begin  
    ⋮  
  end;  
  
  procedure r;  
  begin  
    i := i - 1;  
    if i > 0 then  
      r  
    else  
      q  
    end;  
  
  procedure s;  
  begin  
    r;  
    q  
  end;  
  
begin  
  i := 2;
```

○ s
● end.



[Procedure come
astrazioni](#)

Record di attivazione

[Ambiente locale](#)

[Esempio](#)

[Propagazione dei
data object](#)

[Bibliografia](#)

Record di attivazione



Ambiente locale

[Procedure come astrazioni](#)

[Record di attivazione](#)

[Ambiente locale](#)

[Esempio](#)

[Propagazione dei data object](#)

[Bibliografia](#)

Include:

1. Tutte le variabili dichiarate localmente.



Ambiente locale

[Procedure come astrazioni](#)

[Record di attivazione](#)

[Ambiente locale](#)

[Esempio](#)

[Propagazione dei data object](#)

[Bibliografia](#)

Include:

1. Tutte le variabili dichiarate localmente.
2. Puntatore alla prossima istruzione [IP] (permette di riprendere l'esecuzione quando il controllo viene restituito alla procedura chiamante).



Ambiente locale

[Procedure come astrazioni](#)

[Record di attivazione](#)

[Ambiente locale](#)

[Esempio](#)

[Propagazione dei data object](#)

[Bibliografia](#)

Include:

1. Tutte le variabili dichiarate localmente.
2. Puntatore alla prossima istruzione [IP] (permette di riprendere l'esecuzione quando il controllo viene restituito alla procedura chiamante).
3. Memoria temporanea necessaria alla valutazione delle espressioni contenute nella procedura (altamente dipendente dalla realizzazione).



Esempio

[Procedure come astrazioni](#)

[Record di attivazione](#)

[Ambiente locale](#)

Esempio

[Propagazione dei data object](#)

[Bibliografia](#)

Run time stack

```

program p;
  var i;

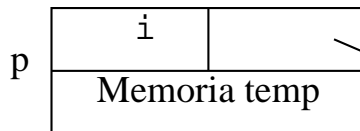
  procedure q;
    var vq;
    begin
      ...
    end;

  procedure r;
    var vr;
    begin
      i := i - 1;
      vr := i;
      if vr > 0 then
        r
      else
        q
      end;
    end;

  procedure s;
    var vs;
    begin
      r;
      s;
    end;

begin
  i := 2;
  s;
end.

```





Esempio

[Procedure come astrazioni](#)

[Record di attivazione](#)

[Ambiente locale](#)

Esempio

[Propagazione dei data object](#)

[Bibliografia](#)

Run time stack

```

program p;
  var i;

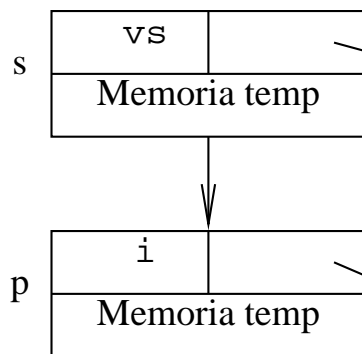
  procedure q;
    var vq;
    begin
      ...
    end;

  procedure r;
    var vr;
    begin
      i := i - 1;
      vr := i;
      if vr > 0 then
        r
      else
        q
      end;
    end;

  procedure s;
    var vs;
    begin
      r;
      s;
    end;

begin
  i := 2;
  s
end.

```





Esempio

[Procedure come astrazioni](#)

[Record di attivazione](#)

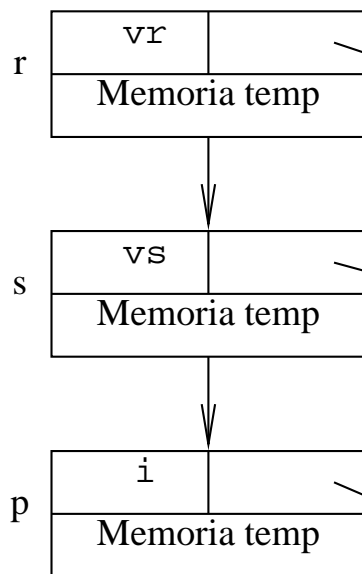
[Ambiente locale](#)

Esempio

[Propagazione dei data object](#)

[Bibliografia](#)

Run time stack



```

program p;
  var i;

  procedure q;
    var vq;
    begin
      ...
    end;

  procedure r;
    var vr;
    begin
      i := i - 1;
      vr := i;
      if vr > 0 then
        r
      else
        q
      end;
    end;

  procedure s;
    var vs;
    begin
      r;
      s;
    end;

begin
  i := 2;
  s;
end.

```



Esempio

[Procedure come astrazioni](#)

[Record di attivazione](#)

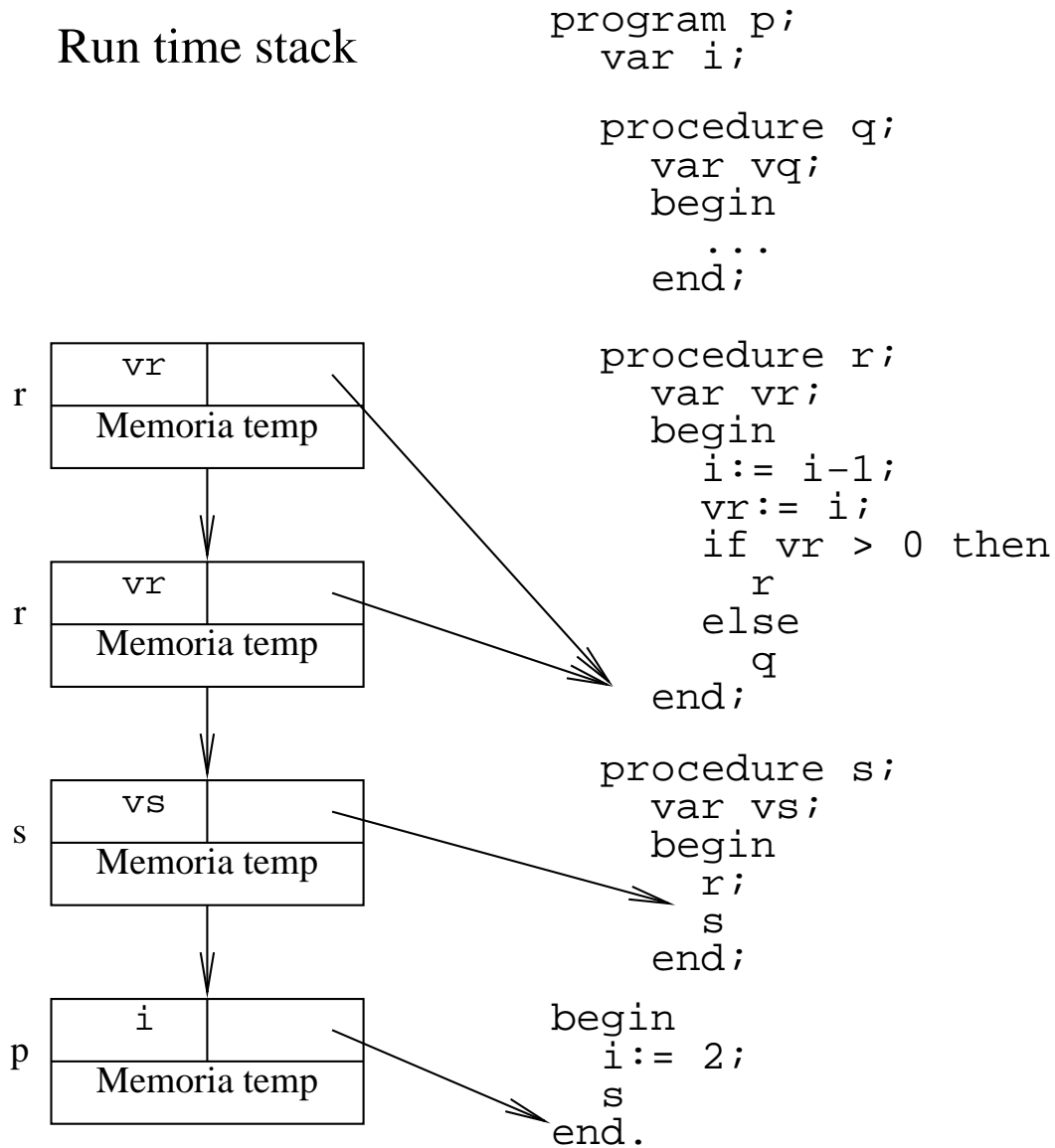
[Ambiente locale](#)

Esempio

[Propagazione dei data object](#)

[Bibliografia](#)

Run time stack





Esempio

[Procedure come astrazioni](#)

[Record di attivazione](#)

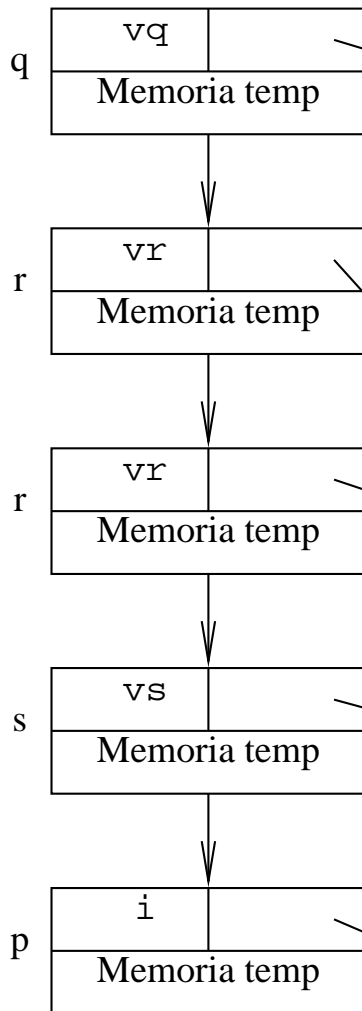
[Ambiente locale](#)

Esempio

[Propagazione dei data object](#)

[Bibliografia](#)

Run time stack



```

program p;
  var i;

```

```

procedure q;
  var vq;
  begin
    ...
  end;

```

```

procedure r;
  var vr;
  begin
    i := i - 1;
    vr := i;
    if vr > 0 then
      r
    else
      q
    end;
  end;

```

```

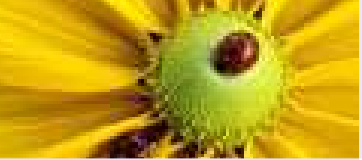
procedure s;
  var vs;
  begin
    r;
    s;
  end;

```

```

begin
  i := 2;
  s;
end.

```



Procedure come
astrazioni

Record di attivazione

**Propagazione dei
data object**

Realizzazione

Ambito statico

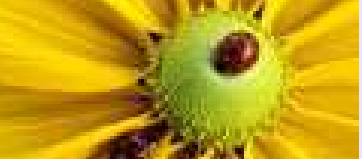
Ambito statico (2)

Ambito dinamico

Osservazioni

Bibliografia

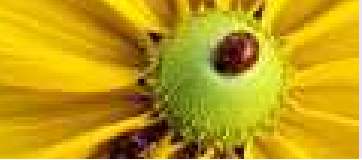
Propagazione dei data object



Realizzazione

- Viene realizzata aggiungendo al record di attivazione un puntatore al record di attivazione della procedura da cui vengono propagate le definizioni o i dati.

- Viene realizzata aggiungendo al record di attivazione un puntatore al record di attivazione della procedura da cui vengono propagate le definizioni o i dati.
- Se viene richiesto l'accesso ad un dato che non è definito localmente, esso viene ricercato in modo ricorsivo nei record di attivazione precedenti.



Realizzazione

- Viene realizzata aggiungendo al record di attivazione un puntatore al record di attivazione della procedura da cui vengono propagate le definizioni o i dati.
- Se viene richiesto l'accesso ad un dato che non è definito localmente, esso viene ricercato in modo ricorsivo nei record di attivazione precedenti.
- Tre tipologie di realizzazione della propagazione.

- Viene realizzata aggiungendo al record di attivazione un puntatore al record di attivazione della procedura da cui vengono propagate le definizioni o i dati.
- Se viene richiesto l'accesso ad un dato che non è definito localmente, esso viene ricercato in modo ricorsivo nei record di attivazione precedenti.
- Tre tipologie di realizzazione della propagazione.
 1. Propagazione in ambito statico. In questo caso l'ambiente non locale di una procedura è propagato dal programma che la contiene sintatticamente: propagazione di posizione.



Realizzazione

- Viene realizzata aggiungendo al record di attivazione un puntatore al record di attivazione della procedura da cui vengono propagate le definizioni o i dati.
- Se viene richiesto l'accesso ad un dato che non è definito localmente, esso viene ricercato in modo ricorsivo nei record di attivazione precedenti.
- Tre tipologie di realizzazione della propagazione.
 1. Propagazione in ambito statico. In questo caso l'ambiente non locale di una procedura è propagato dal programma che la contiene sintatticamente: propagazione di posizione.
 2. Propagazione in ambito dinamico. In questo caso l'ambiente non locale di una procedura è propagato dal programma chiamante.

- Viene realizzata aggiungendo al record di attivazione un puntatore al record di attivazione della procedura da cui vengono propagate le definizioni o i dati.
- Se viene richiesto l'accesso ad un dato che non è definito localmente, esso viene ricercato in modo ricorsivo nei record di attivazione precedenti.
- Tre tipologie di realizzazione della propagazione.
 1. Propagazione in ambito statico. In questo caso l'ambiente non locale di una procedura è propagato dal programma che la contiene sintatticamente: propagazione di posizione.
 2. Propagazione in ambito dinamico. In questo caso l'ambiente non locale di una procedura è propagato dal programma chiamante.
 3. Nessuna propagazione. L'uso di ambienti non locali è scoraggiato perché produce **effetti collaterali** non facilmente prevedibili.

Ambito statico

[Procedure come astrazioni](#)

[Record di attivazione](#)

[Propagazione dei data object](#)

[Realizzazione](#)

[Ambito statico](#)

[Ambito statico \(2\)](#)

[Ambito dinamico](#)

[Osservazioni](#)

[Bibliografia](#)

```
program p;
  var a, b, c: integer;

  procedure q;
    var a, c: integer;
    procedure r;
      var a: integer;
      begin {r}
        ...
      end; {r}
    begin {q}
      ...
    end; {q}

  procedure s;
    var b: integer;
    begin {s}
      ...
    end; {s}

begin {p}
  ...
end. {p}
```

{variabili: a da r; b da p; c da q;
procedure: q da p; r da q}

{variabili: a da q; b da p; c da q;
procedure: q ed s da p; r da q}

{variabili: a da p; b da s; c da p;
procedure: q ed s da p}

{variabili: a, b, c da p;
procedure: q, s da p}



Ambito statico (2)

Procedure come
astrazioni

Record di attivazione

Propagazione dei
data object

Realizzazione

Ambito statico

Ambito statico (2)

Ambito dinamico

Osservazioni

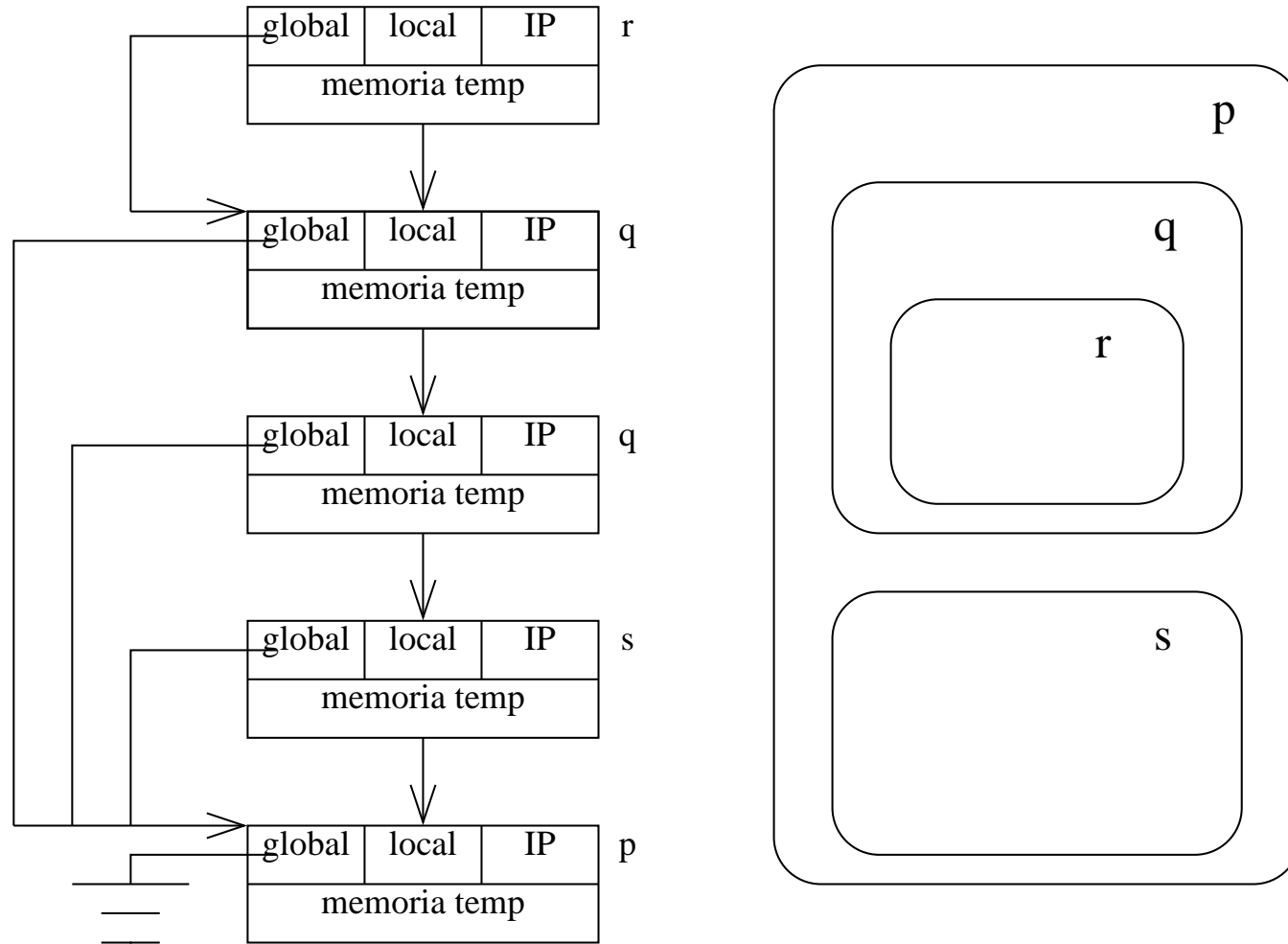
Bibliografia

Supponendo una sequenza di attivazione (p, s, q, q, r) , lo stack di esecuzione ha questa forma:



Ambito statico (2)

Supponendo una sequenza di attivazione (p, s, q, q, r), lo stack di esecuzione ha questa forma:



[Procedure come astrazioni](#)

[Record di attivazione](#)

[Propagazione dei data object](#)

[Realizzazione](#)

[Ambito statico](#)

Ambito statico (2)

[Ambito dinamico](#)

[Osservazioni](#)

[Bibliografia](#)



Ambito dinamico

Procedure come astrazioni

Record di attivazione

Propagazione dei data object

Realizzazione

Ambito statico

Ambito statico (2)

Ambito dinamico

Osservazioni

Bibliografia

La stessa sequenza di attivazione precedente (p, s, q, q, r), genera allora lo stack di esecuzione:



Ambito dinamico

- [Procedure come astrazioni](#)

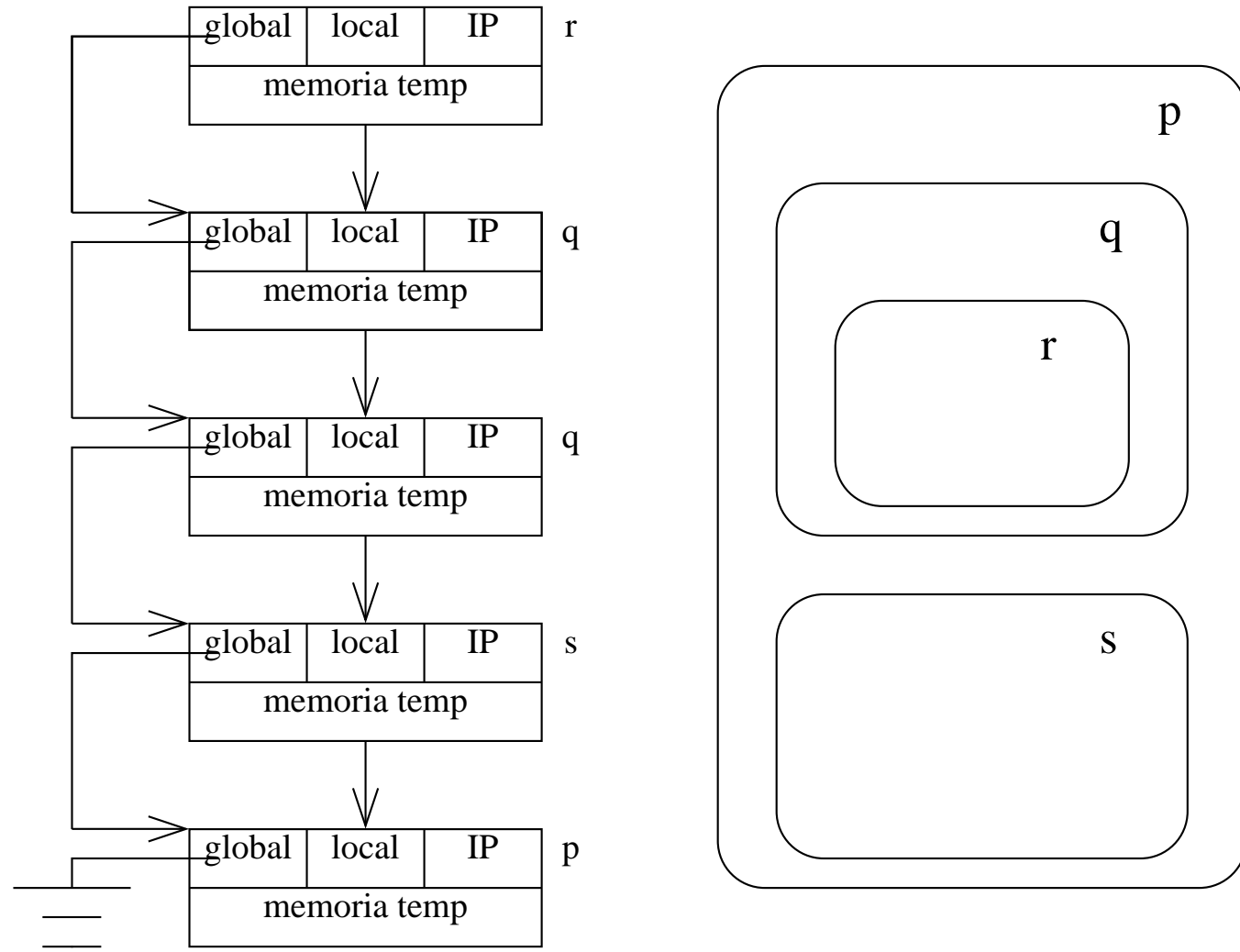
- [Record di attivazione](#)

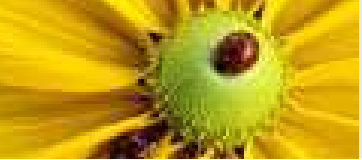
- [Propagazione dei data object](#)

- [Realizzazione](#)
- [Ambito statico](#)
- [Ambito statico \(2\)](#)
- [Ambito dinamico](#)
- [Osservazioni](#)

- [Bibliografia](#)

La stessa sequenza di attivazione precedente (p, s, q, q, r), genera allora lo stack di esecuzione:





Osservazioni

Nella propagazione in ambito dinamico:

- il puntatore all'ambiente non locale non è più necessario;

Nella propagazione in ambito dinamico:

- il puntatore all'ambiente non locale non è più necessario;
- è praticamente impossibile la determinazione dell'ambiente di esecuzione di una procedura durante la scrittura del codice sorgente.

Nella propagazione in ambito dinamico:

- il puntatore all'ambiente non locale non è più necessario;
- è praticamente impossibile la determinazione dell'ambiente di esecuzione di una procedura durante la scrittura del codice sorgente.

```
program p;  
  var a: integer;  
  procedure q;  
    begin                               {vars: a da p o da r; procs: q da p}  
    ...  
  end;  
  procedure r;  
    var  
      a: integer;  
    begin                               {vars: a da r; procs: q, r da p}  
    ...  
  end;  
begin                               {vars: a da p; procs: q, r, da p}  
  ...  
end.
```



Bibliografia

Procedure come
astrazioni

Record di attivazione

Propagazione dei
data object

Bibliografia

Bibliografia

- H. L. Dershem. M. J. Jipping. *Programming languages: structures and models*. Second edition. Cap. 5, par. 5.1 - 5.3.