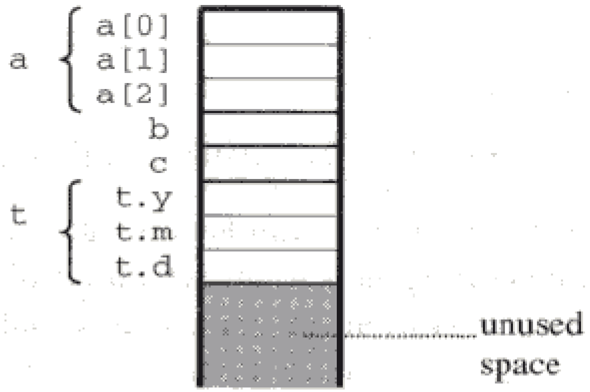
ALOCAÇÃO DE MEMÓRIA

Baseado no Capítulo 6 de Programming Language Processors in Java, de Watt & Brown

Alocação Estática

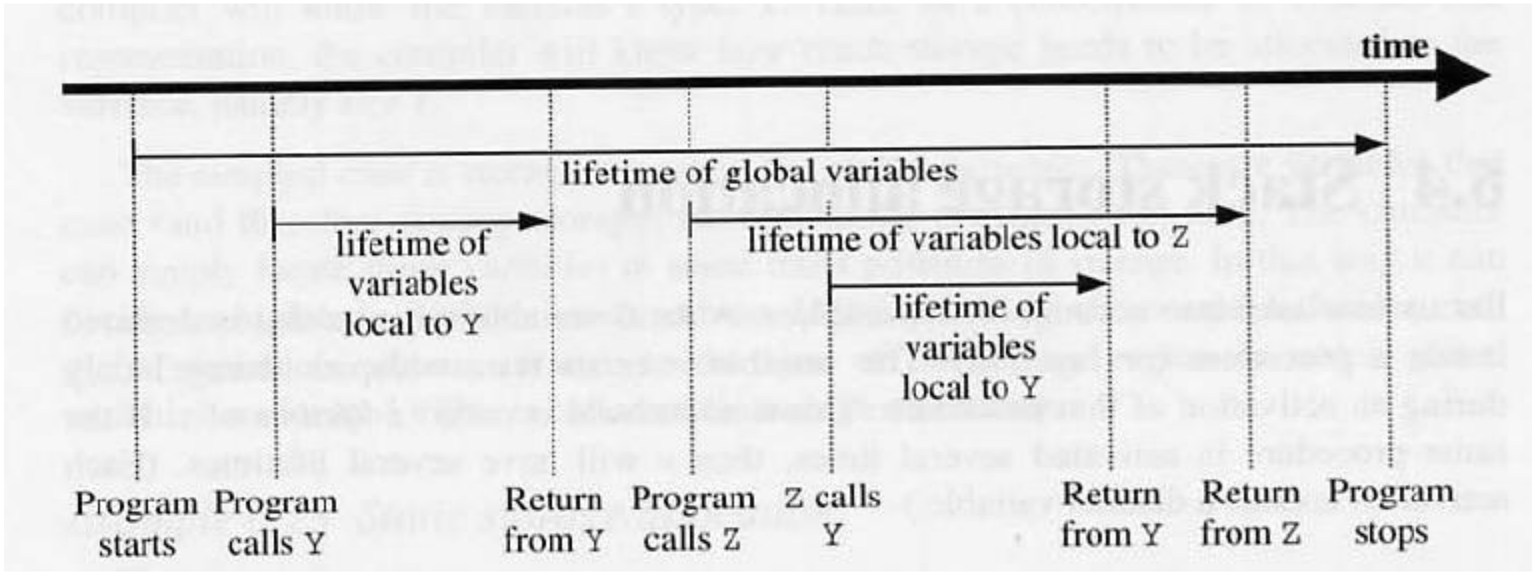
```
let
   type Date = record
                     y: Integer,
                     m: Integer,
                     d: Integer
                  end;
   var a: array 3 of Integer;
   var b: Boolean;
   var c: Char;
   var t: Date
in
   ...
```

$$\begin{aligned} \mathit{address}[\![\texttt{a}]\!] &= 0 \\ \mathit{address}[\![\texttt{b}]\!] &= 3 \\ \mathit{address}[\![\texttt{c}]\!] &= 4 \\ \mathit{address}[\![\texttt{t}]\!] &= 5 \end{aligned}$$

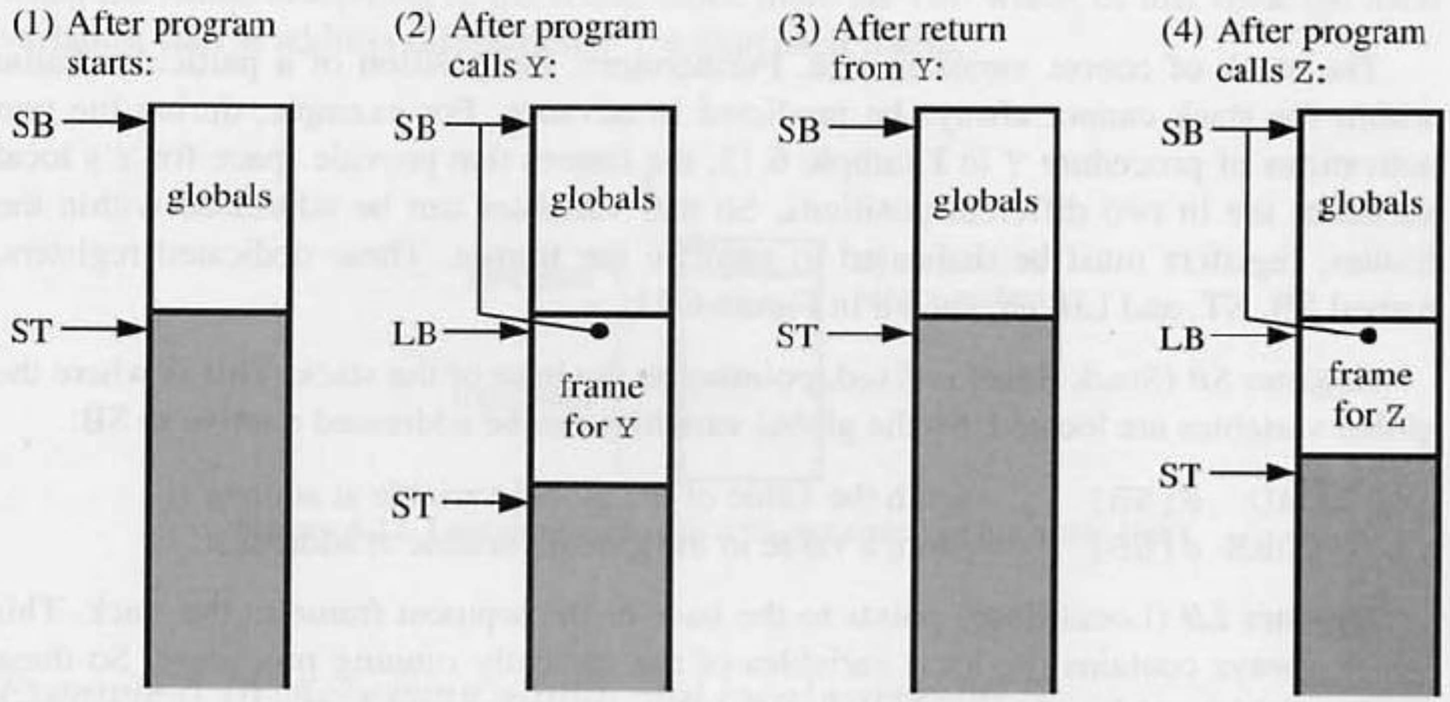


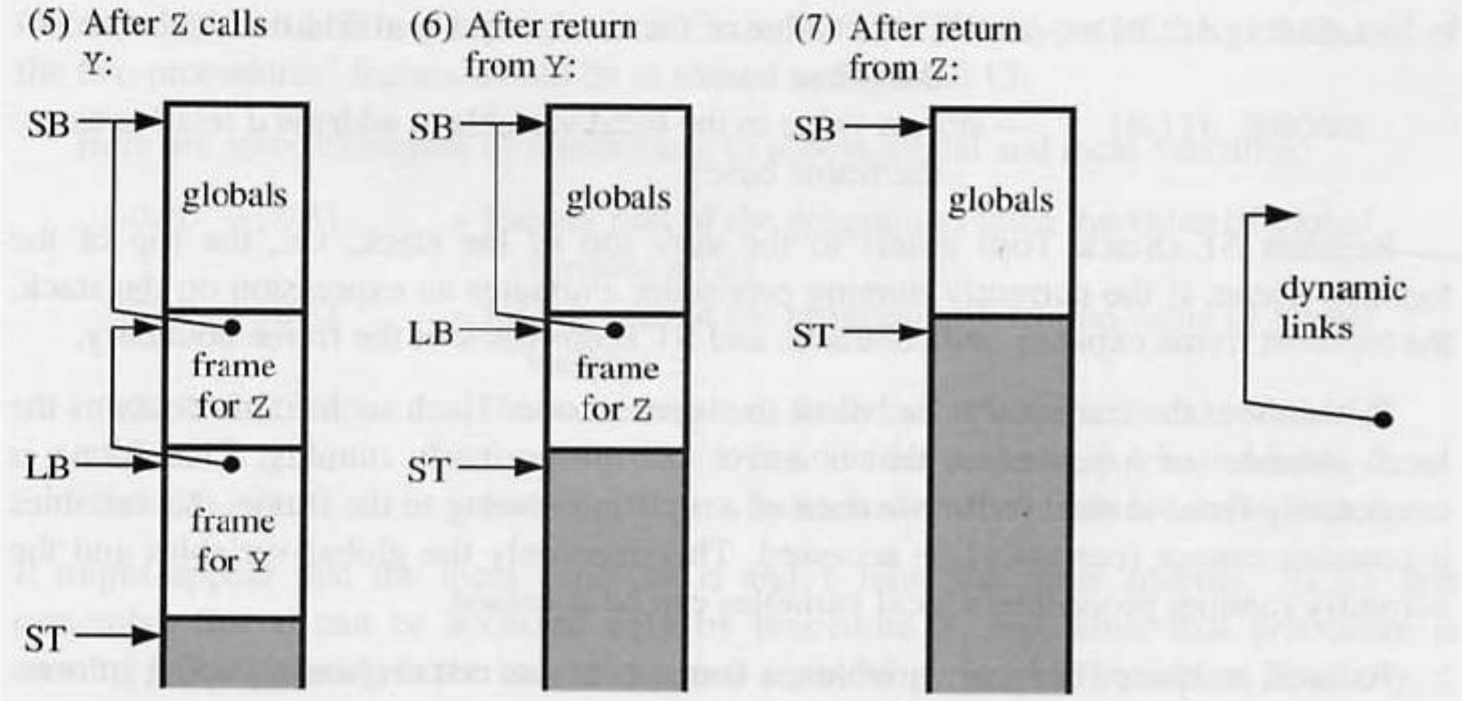
Alocação Automática (stack)

```
let
   var a: array 3 of Integer;
   var b: Boolean;
   var c: Char;
   proc Y () ~
      let
         var d: Integer;
         var e: record c: Char, n: Integer end
      in
         ...;
   proc Z () ~
      let
         var f: Integer
      in
         begin ...; Y(); ... end
in
   begin ...; Y(); ...; Z(); ... end
```


time
lifetime of global variables
lifetime of variables local to Y
lifetime of variables local to Z
lifetime of variables local to Y
Program starts
Program calls Y
Return from Y
Program calls Z
Z calls Y
Return from Y
Return from Z
Program stops

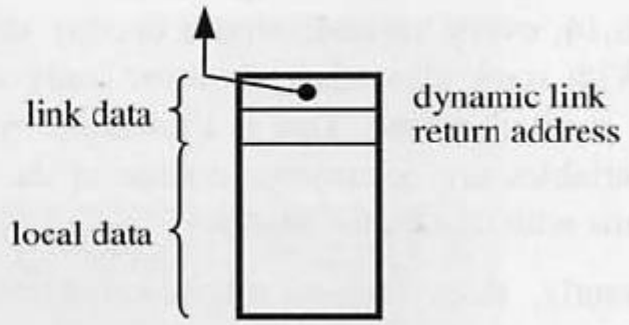
Variáveis globais
e
locais


(1) After program starts:
SB
globals
ST
(2) After program calls Y:
SB
globals
LB
frame for Y
ST
(3) After return from Y:
SB
globals
ST
(4) After program calls Z:
SB
globals
LB
frame for Z
ST

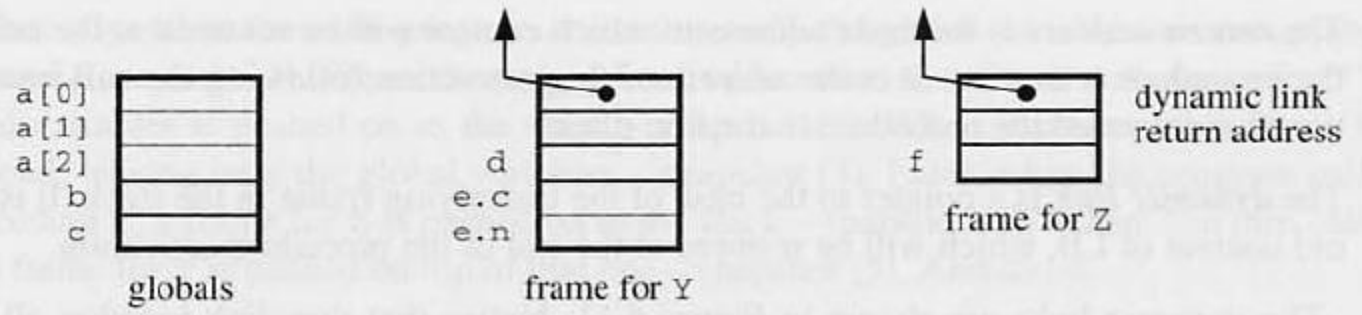

(5) After Z calls Y:
SB
globals
frame for Z
LB
frame for Y
ST
(6) After return from Y:
SB
globals
LB
frame for Z
ST
(7) After return from Z:
SB
globals
ST
dynamic links

```
LOAD  d[SB]     – fetch the value of the global variable at address d.
STORE d[SB]     – store a value in the global variable at address d.
```

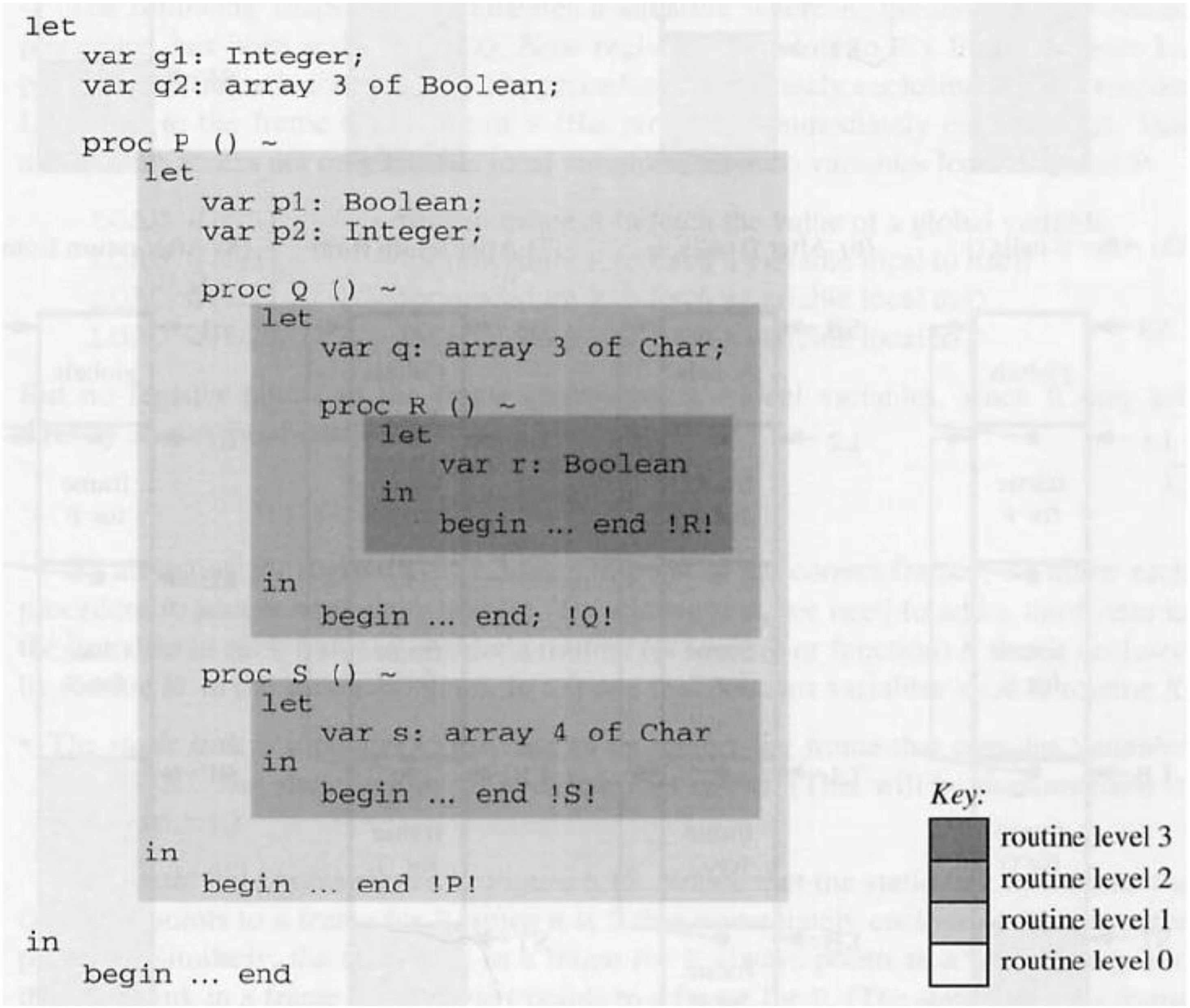
```
LOAD  d[LB]     – fetch the value of the local variable at address d relative to
                    the frame base.
STORE d[LB]     – store a value in the local variable at address d relative to
                    the frame base.
```



`LOAD 0[SB]`	– for any part of the program to fetch the value of global variable `a[0]`
`LOAD 4[SB]`	– for any part of the program to fetch the value of global variable `c`
`LOAD 2[LB]`	– for procedure `Y` to fetch the value of its local variable `d`
`LOAD 4[LB]`	– for procedure `Y` to fetch the value of its local variable `e.n`
`LOAD 2[LB]`	– for procedure `Z` to fetch the value of its local variable `f`



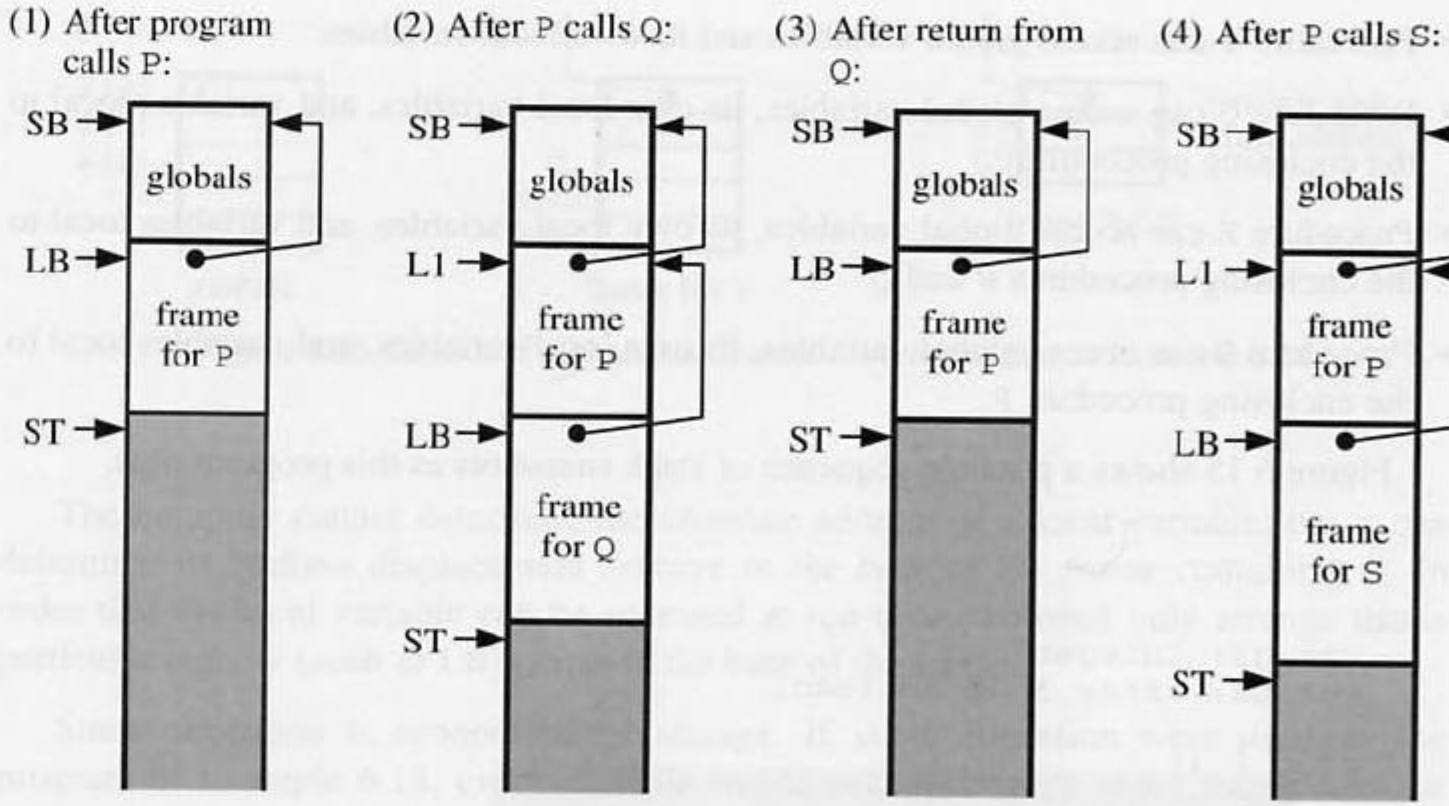
Variáveis não-locais

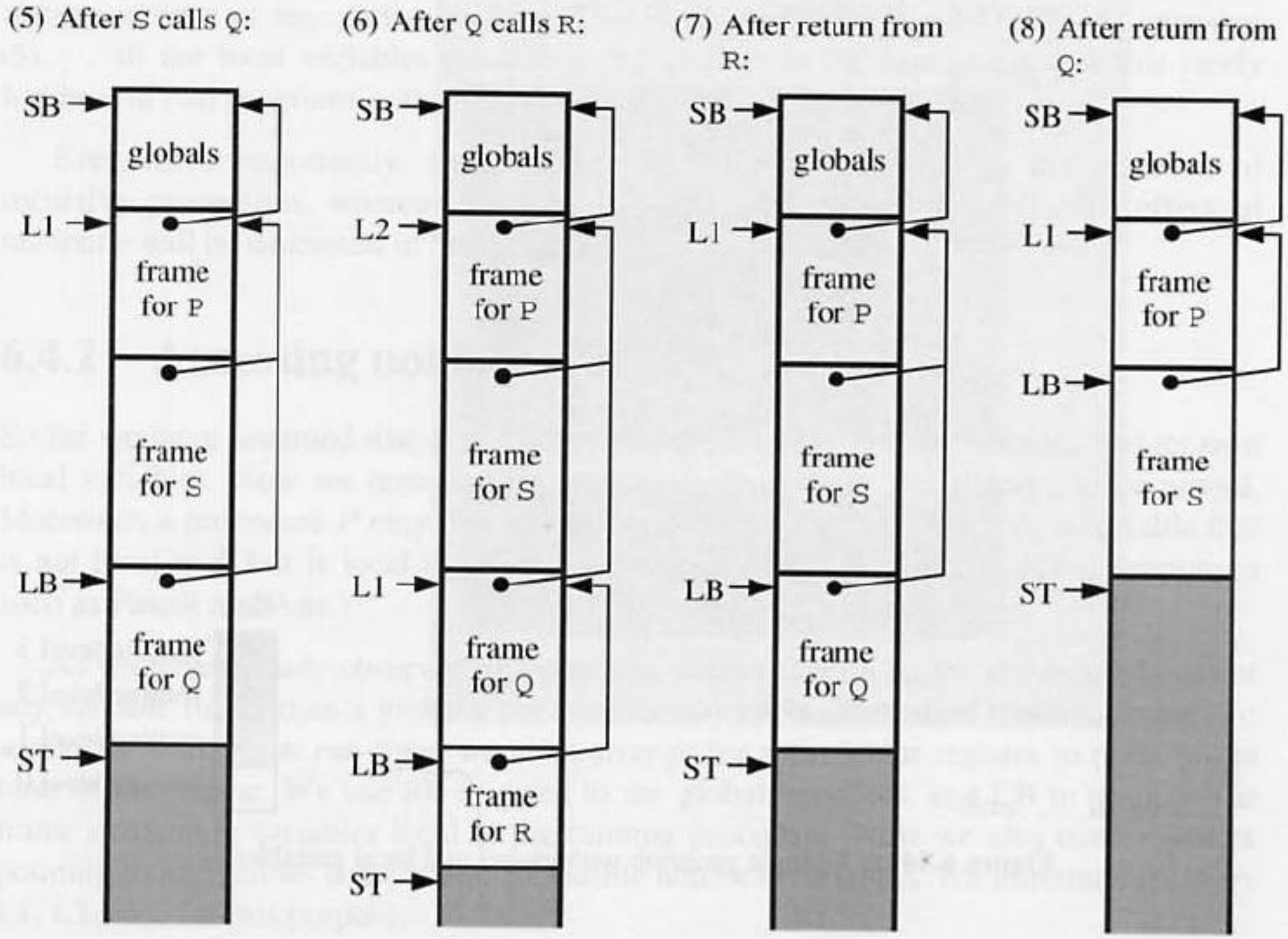


```
let
   var g1: Integer;
   var g2: array 3 of Boolean;

   proc P () ~
      let
         var p1: Boolean;
         var p2: Integer;

         proc Q () ~
            let
               var q: array 3 of Char;

               proc R () ~
                  let
                     var r: Boolean
                  in
                     begin ... end !R!

            in
               begin ... end; !Q!

         proc S () ~
            let
               var s: array 4 of Char
            in
               begin ... end !S!

      in
         begin ... end !P!

in
   begin ... end
```

Key:

■	routine level 3
▩	routine level 2
▨	routine level 1
□	routine level 0

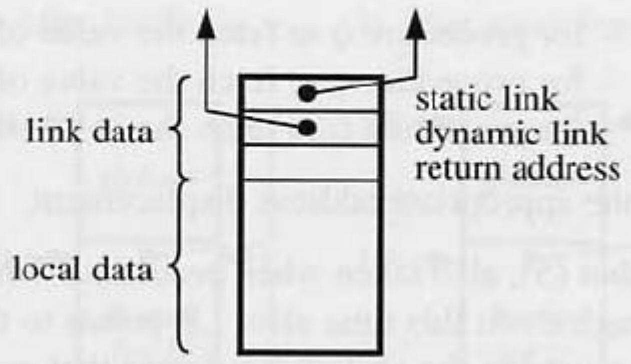

(1) After program calls P:
SB
globals
LB
frame for P
ST
(2) After P calls Q:
SB
globals
L1
frame for P
LB
frame for Q
ST
(3) After return from Q:
SB
globals
LB
frame for P
ST
(4) After P calls S:
SB
globals
L1
frame for P
LB
frame for S
ST

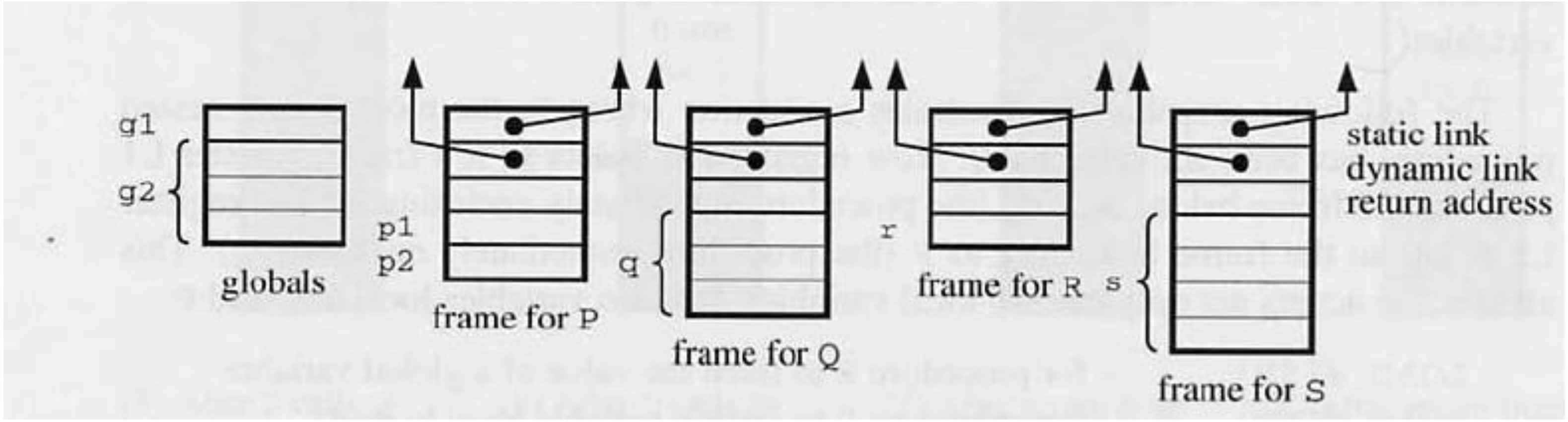


(5) After S calls Q:
SB
globals
L1
frame for P
frame for S
LB
frame for Q
ST
(6) After Q calls R:
SB
globals
L2
frame for P
frame for S
L1
frame for Q
LB
frame for R
ST
(7) After return from R:
SB
globals
L1
frame for P
frame for S
LB
frame for Q
ST
(8) After return from Q:
SB
globals
L1
frame for P
LB
frame for S
ST

`LOAD d[SB]`	– for procedure Q to fetch the value of a global variable
`LOAD d[LB]`	– for procedure Q to fetch the value of a variable local to itself
`LOAD d[L1]`	– for procedure Q to fetch the value of a variable local to P

`LOAD d[SB]`	– for procedure R to fetch the value of a global variable
`LOAD d[LB]`	– for procedure R to fetch a variable local to itself
`LOAD d[L1]`	– for procedure R to fetch a variable local to Q
`LOAD d[L2]`	– for procedure R to fetch a variable local to P





$$L1 = \mathit{content}(LB)$$
$$L2 = \mathit{content}(L1) = \mathit{content}(\mathit{content}(LB))$$
$$L3 = \mathit{content}(L2) = \mathit{content}(\mathit{content}(\mathit{content}(LB)))$$

If $l = 0$ (i.e., v is a global variable):

```
    LOAD d[SB]          – for any code to fetch the value of v
```

If $l > 0$ (i.e., v is a local variable):

```
    LOAD d[LB]          – for code at level l to fetch the value of v
    LOAD d[L1]          – for code at level l+1 to fetch the value of v
    LOAD d[L2]          – for code at level l+2 to fetch the value of v
```