Cap. 5 Pesquisa em Memória Primária

## Continuação do Programa 5.15

```
then begin writeln ('Chave nao esta na arvore'); Fim := true; end
procedure Antecessor (q: TipoApontador; var r: TipoApontador;
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              if not Fim then EsqCurto (Ap, Fim);
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              Antecessor (Aux, Aux^.Esq, Fim);
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           Ap := Aux^.Dir; dispose (Aux); if Ap <> nil then Fim := true;
                                                                                                                                                                                                                                                                                                                                                                                                                       if not Fim then EsqCurto (Ap, Fim);
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          if not Fim then DirCurto (Ap, Fim);
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     Ap := Aux^.Esq; dispose (Aux); if Ap <> nil then Fim := true;
                                                                                                                           if not Fim then DirCurto (r, Fim);
                      var Fim: boolean);
                                                                                                                                                                                                                                                                                                                                                                                                                                                              else if x.Chave > Ap^.Reg.Chave
                                                                                                                                                                                                                                                                                                                                                                                                    IRetira (x, Ap^.Esq, Fim);
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        IRetira (x, Ap^.Dir, Fim);
                                                                                                                                                                                                              dispose (q);
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    else begin { Encontrou chave }
                                                                                                                                                                                                               r := r^.Esq; dispose (q)

if r <> nil then Fim := true;
                                                                                                         Antecessor (q, r .. Dir, Fim);
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      Fim := false; Aux := Ap;
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             else if Aux^.Esq = nil
                                                                                                                                                                                                                                                                                                                                                            else if x.Chave < Ap^.Reg.Chave
                                                                                                                                                                                          q^.Reg := r^.Reg; q := r;
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          if Aux Dir = nil
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       then begin
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         else begin
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   then begin
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           IRetira (x, Ap, Fim)
                                                                                                                                                                                                                                                                             Antecessor }
                                                                   if r^.Dir <> nil
                                                                                                                                                                                                                                                                                                                                                                              then begin
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     then begin
                                                                                                                                                                                                                                                                                             begin { IRetira }
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   end; { IRetira
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                end; { Retira }
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     begin { Retira
                                                                                        then begin
                                                                                                                                                                         else begin
                                                                                                                                                                                                                                                           end;
                                                                                                                                                      end
```

A Figura 5.8 mostra o resultado obtido quando se retira uma sequência de chaves da árvore SBB: a árvore à esquerda é obtida após a retirada da chave 7 da árvore a direita na Figura 5.7; a árvore do meio é obtida após a retirada da chave 5 da árvore anterior; a árvore à direita é obtida após a retirada da chave 9 da árvore anterior.

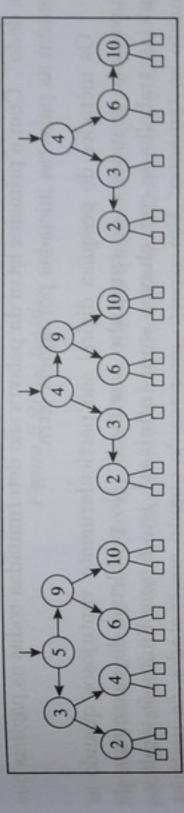


Figura 5.8 Decomposição de uma árvore SBB.

Análise Para as árvores SBB é necessário distinguir dois tipos de altura. Uma delas é a altura vertical h, necessária para manter a altura uniforme e obtida por meio da contagem do número de apontadores verticais em qualquer caminho entre a raiz e um nó externo. A outra é a altura k, que representa o número máximo de comparações de chaves obtidas mediante contagem do número total de apontadores no maior caminho entre a raiz e um nó externo. A altura k é maior que a altura h sempre que existirem apontadores horizontais na árvore. Para uma árvore SBB com n nós internos, temos:

$$h < k < 2h$$
.

De fato, Bayer (1972) mostrou que:

$$\log(n+1) \le k \le 2\log(n+2) - 2.$$

O custo para manter a propriedade SBB é exatamente o custo para percorrer o caminho de pesquisa para encontrar a chave, seja para inseri-la seja para retirála. Logo, esse custo é  $O(\log n)$ .

O número de comparações em uma pesquisa com sucesso na árvore SBB é

melhor caso : 
$$C(n) = O(1)$$
,  
pior caso :  $C(n) = O(\log n)$ ,  
caso médio :  $C(n) = O(\log n)$ .

Na prática, o caso médio para  $C_n$  é apenas cerca de 2% pior que o  $C_n$  para uma árvore completamente balanceada, conforme mostrado em Ziviani e Tompa (1982).