

A Figura 5.7 mostra o resultado obtido quando se insere uma sequência de chaves em uma árvore SBB inicialmente vazia: a árvore à esquerda é obtida após a inserção das chaves 7, 10, 5; a árvore do meio é obtida após a inserção das chaves 2, 4 na árvore anterior; a árvore à direita é obtida após a inserção das chaves 9, 3, 6 na árvore anterior. A árvore de pesquisa mostrada na Figura 5.5 pode ser obtida quando as chaves 1, 8 são inseridas na árvore à direita na Figura 5.7.

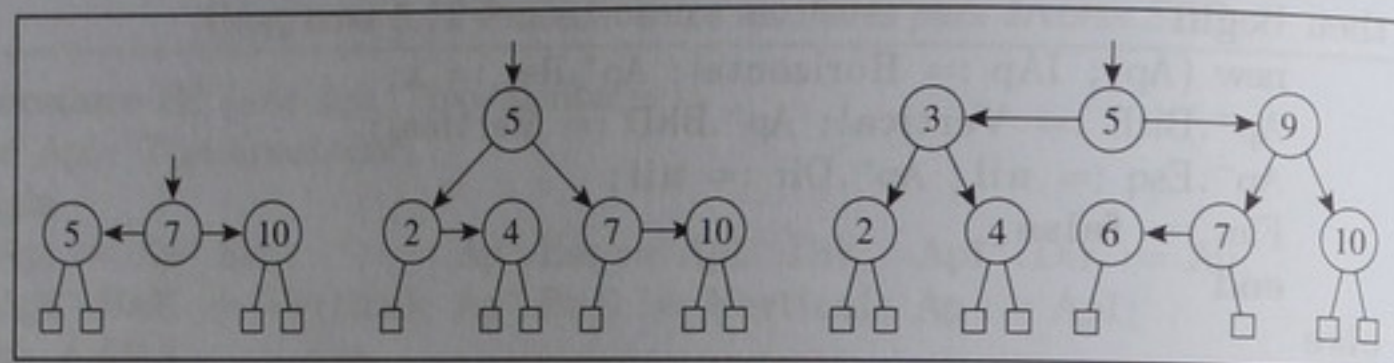


Figura 5.7 Crescimento de uma árvore SBB.

O procedimento Inicializa é extremamente simples, conforme ilustra o Programa 5.14.

Programa 5.14 Procedimento para inicializar a árvore SBB

```
procedure Inicializa (var Dicionario: TipoDicionario);
begin
  Dicionario := nil;
end; { Inicializa }
```

O procedimento **Retira** pode ser visto no Programa 5.15. Assim como o procedimento **Inserir** mostrado anteriormente, o procedimento **Retira** contém outro procedimento interno de nome **IRetira**, cuja interface contém um parâmetro a mais que o procedimento **Retira**, a saber: o parâmetro **Fim** toma o valor **true** quando a propriedade SBB é restabelecida e nada mais é necessário fazer.

Por sua vez, o procedimento **IRetira** utiliza três procedimentos internos, a saber:

- ❑ **EsqCurto** (**DirCurto**) é chamado quando um nó folha (que é referenciado por um apontador vertical) é retirado da subárvore à esquerda (direita), tornando-a menor na altura após a retirada;
- ❑ Quando o nó a ser retirado possui dois descendentes, o procedimento **Antecessor** localiza o nó antecessor para ser trocado com o nó a ser retirado.

Programa 5.15 Procedimento para retirar da árvore SBB

```
procedure Retira (x: TipoRegistro; var Ap: TipoApontador);
var Fim: boolean;
procedure IRetira (x: TipoRegistro; var Ap: TipoApontador; var Fim: boolean);
var Aux: TipoApontador;
procedure EsqCurto (var Ap: TipoApontador; var Fim: boolean);
var Apl: TipoApontador;
begin { Folha esquerda retirada => arvore curta na altura esquerda }
  if Ap.BitE = Horizontal
  then begin Ap.BitE := Vertical; Fim := true; end
  else if Ap.BitD = Horizontal
  then begin
    Apl := Ap.Dir; Ap.Dir := Apl.Esq; Apl.Esq := Ap; Ap := Apl;
    if Ap.Esq.Dir.BitE = Horizontal
    then begin DE (Ap.Esq); Ap.BitE := Horizontal; end
    else if Ap.Esq.Dir.BitD = Horizontal
    then begin DD (Ap.Esq); Ap.BitE := Horizontal; end;
    Fim := true;
  end
  else begin
    Ap.BitD := Horizontal;
    if Ap.Dir.BitE = Horizontal
    then begin DE (Ap); Fim := true; end
    else if Ap.Dir.BitD = Horizontal
    then begin DD (Ap); Fim := true; end;
  end;
end; { EsqCurto }
procedure DirCurto (var Ap: TipoApontador; var Fim: boolean);
var Apl: TipoApontador;
begin { Folha direita retirada => arvore curta na altura direita }
  if Ap.BitD = Horizontal
  then begin Ap.BitD := Vertical; Fim := true; end
  else if Ap.BitE = Horizontal
  then begin
    Apl := Ap.Esq; Ap.Esq := Apl.Dir; Apl.Dir := Ap; Ap := Apl;
    if Ap.Dir.Esq.BitD = Horizontal
    then begin ED (Ap.Dir); Ap.BitD := Horizontal; end
    else if Ap.Dir.Esq.BitE = Horizontal
    then begin EE (Ap.Dir); Ap.BitD := Horizontal; end;
    Fim := true;
  end
  else begin
    Ap.BitE := Horizontal;
    if Ap.Esq.BitD = Horizontal
    then begin ED (Ap); Fim := true; end
    else if Ap.Esq.BitE = Horizontal
    then begin EE (Ap); Fim := true; end;
  end;
end; { DirCurto }
```