```
proc  cria-folha (p, x);
begin
      aloque (p);
      r (p):= x;
      esq(p):= dir (p):= nil
end;
```

Algoritmo 7.15 – Cria folha para Árvore PATRICIA – Algoritmo auxiliar

```
proc insere-pat (ptr, x, a)
begin
    a:= 0;       { a = 0: operação bem sucedida }
    if  ptr = nil
    then cria-folha (ptr, x)      {árvore vazia}
    else  begin   { árvore não vazia }
          pt:= ptr;
          pesqpat (x, pt, a);
          if a<> 1 {insere se x não está na árvore}
          then begin
             {localiza uma chave daquele ramo para comparar com x}
             while  esq (pt)  <>  nil  do  pt:= esq (pt);
             x1:= r(pt);
             pt:= ptr; pta:= nil;
             l:= 0;
             {determina primeiro dígito onde x se diferencia da chave x1}
             {k= número de dígitos da chave x}
             {c= número de dígitos da chave x1 localizada na árvore}
             while  ( l < min (k, c)  and  d[l+1]=d'[l+1])
             do begin
                   l:=l+1;
                   if r(pt) = l
                   then  if d[l] = 0
                         then begin pta:= pt; pt:= esq(pt); ed:= V end
                         else  begin pta:= pt; pt:= dir(pt);   ed:= F  end
                end;
             if  l = min (k, c)
             then a:= 2    {erro: x é prefixo ou tem prefixo na árvore }
             else  begin
                      {inserção é possível, aloca e prepara novos nós }
                      {um (px) para a nova chave (x) e outro (ptv) para a bifurcação}
                      cria-folha (px, x);
                      aloque (ptv); r (ptv):= l + 1;

                      {cria a bifurcação para o novo nó}
                      if d [l+1] = 0
                      then  begin esq (ptv):= px; dir (ptv):= pt end
                      else  begin  esq (ptv):= pt; dir (ptv):= px end;

                      {teste falhou na raíz: nova raíz}
                      if pt = ptr then ptr:= ptv;

                      {teste falhou em um nível qualquer, diferente da raíz}
                      else  if ed
                            then esq (pta):= ptv
                            else dir (pta):= ptv
                   end
          end
   end;
end;
```

Algoritmo 7.16 – Inserção em Árvore PATRICIA

```
proc  removepat (x, pt, a)
{remove a chave x da árvore PATRICIA pt}
begin
   {pt aponta para uma folha?}
   if  esq (pt) = nil
   then     if  x = r (pt)   {a folha contém chave igual a x?}
            then  begin
                     a := 1;
                     desaloque (pt);
                     pt:= nil
                  end
            else   a := 2
   else  if  k < r (pt)       {foram esgotados os dígitos de x?}
         then  a := 2
         else      if d [r (pt)] = 0
                   {deriva para a esquerda ou a direita, dependendo do dígito indicado}
                   then  begin
                            {remove na subárvore esquerda}
                            removepat (x, esq (pt) , a);
                            if  a = 1
                            then begin
                                    pt1:= pt;
                                    pt:= dir (pt);
                                    desaloque (pt1);
                                    a:= 3
                                 end
                         end
                   else  begin
                            {remove na subárvore direita}
                            removepat (x, dir (pt) , a);
                            if  a = 1
                            then begin
                                    pt1:= pt;
                                    pt:= esq (pt);
                                    desaloque (pt1);
                                    a:= 3
                                 end
                         end
end;
```

Algoritmo 7.17 - Remove nó em árvore **PATRICIA**

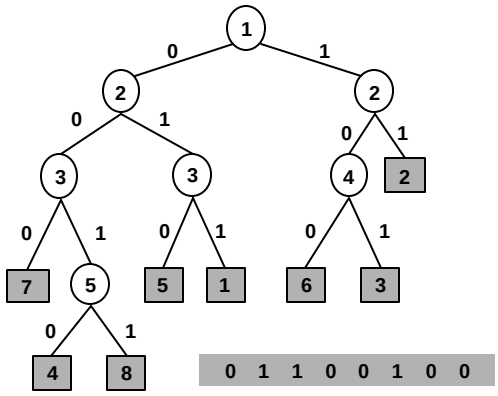

Figura 7.28 – Indexação de um texto com o emprego de uma árvore PATRICIA

A figura 7.28 mostra uma árvore PATRICIA utilizada para indexar um texto. A árvore identifica todos os strings (conhecidos como systrings) contidos no texto e iniciando em qualquer dígito do texto. Cada string é considerado a partir do dígito considerado até o final do texto. Para evitar que um string constitua um prefixo de outro também presente no texto, considera-se que ao final do texto há um caracter (conjunto de 8 bits, por exemplo) diferente de todos os demais.

Em cada folha da árvore é indicada a posição no texto onde inicia o string correspondente.

Na indexação de textos, deve definir a unidade de indexação, ou seja, se os strings serão tomados no início de cada caráter, ou de cada palavra, por exemplo.