

Árvores Patrícias

UFRGS INF01124

INF01124

Métodos de busca até agora ...

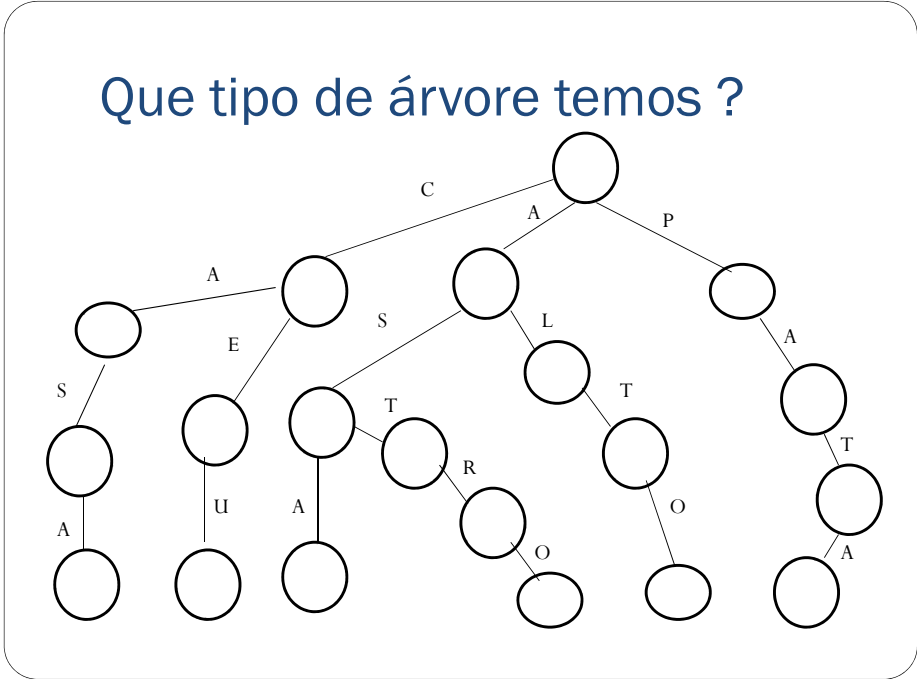
- Chaves indivisíveis
- Chaves de mesmo tamanho
- Eficiência depende especialmente da comparação
- Chaves parciais: possíveis mais com perda de eficiência
- Árvores Patricia: **P**ractical **A**lgorithm **T**o **R**etrieve **I**nformation **C**oded In **A**lphanumeric.

- # Métodos de busca até agora ...
- Chaves indivisíveis
 - Chaves de mesmo tamanho
 - Eficiência depende especialmente da comparação
 - Chaves parciais: possíveis mais com perda de eficiência
 - Árvores Patricia: **P**ractical **A**lgorithm **T**o **R**etrieve **I**nformation **C**oded In **A**lphanumeric.

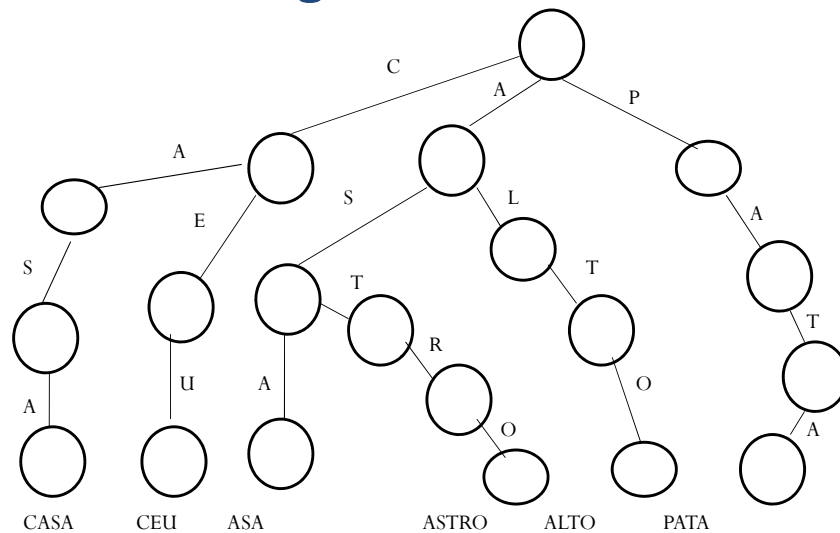
Problemas de decomposição de Chaves

- Busca de frases em texto
- Busca de palavras
- Busca de palavras parciais
- Todas têm tamanho variável

- ## Problemas de decomposição de Chaves
- Busca de frases em texto
 - Busca de palavras
 - Busca de palavras parciais
 - Todas têm tamanho variável

[illegible]

Que tipo de árvore temos ? Árvore Digital N-ária



Árvore Digital Binária

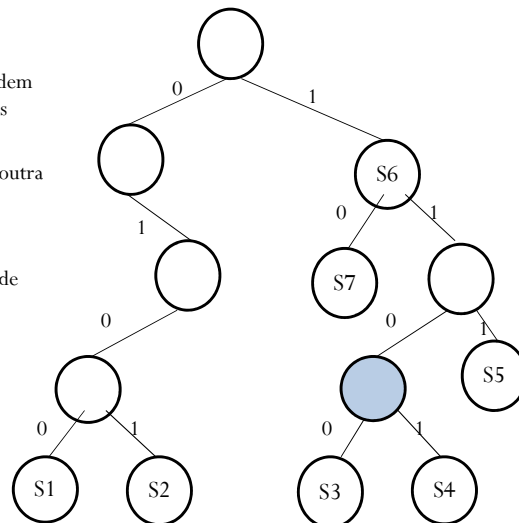
S1 = 00
S2 = 0100
S3 = 0101
S4 = 100
S5 = 1010
S6 = 110
S7 = 01
S8 = 101
S9 = 1

6

Instituto de Informática - UFRGS

Árvore Digital Binária

- Seu formato não depende da ordem em que as chaves são inseridas como a árvore binária
- Uma chave pode ser prefixo de outra
- A chave é dividida em partes
- Desvantagem: Caminhos longo de uma só direção.



Árvore Digital Binária de Prefixo

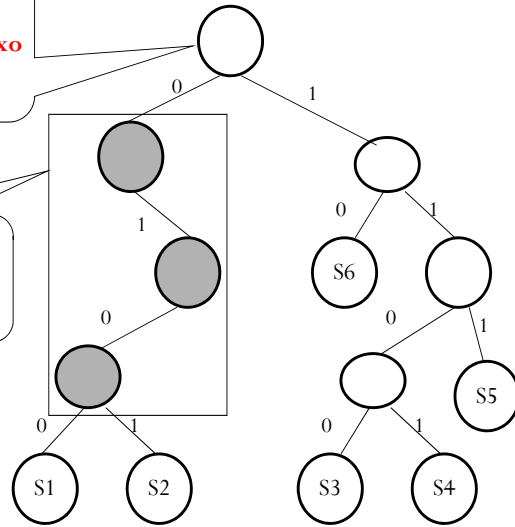
S1 = 0100
S2 = 0101
S3 = 1100
S4 = 1101
S5 = 111

8

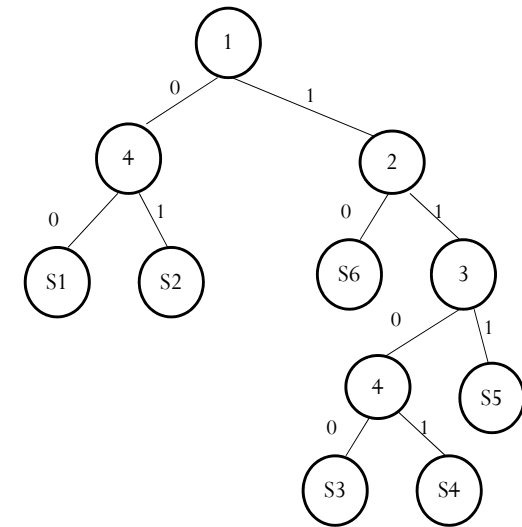
Instituto de Informática - UFRGS

Árvore Digital Binária **de Prefixo**

Um dos problemas mais sérios é a ocorrência de ziguezaques



Árvore Patrícia



Árvore Patrícia

- Árvore digital binária de prefixo obtida pela eliminação dos ziguezagues
- Rótulos nos respectivos nós em que iniciam os ziguezagues eliminados

Pesquisa árvore Patrícia

```

Proc pesqpat ( x,p,a );
    { a = 1 : x é uma chave presente em p }
    { a = 2 : x não é uma chave presente em p }
begin
    if esq(p) = nil
    then if x = r(p)
        then a:= 1
        else a:= 2
    else if kbits(x) < r(p)
        { kbits(x)=nº bits de x }
        then a:=2
        else if d[r(p)] = 0
            { dígito da posição r(p) = 0 }
        then begin
            p:= esq (p);
            pesqpat ( x, p, a);
        end
    else begin
        p:= dir(p);
        pesqpat(x, p, a);
    end
end;

```

```

Proc pesqpat ( x,p,a );
{ a = 1 : x é uma chave presente em p }
{ a = 2 : x não é uma chave presente em p }
begin
    if esq(p) = nil
    then if x = r(p)
        then a:= 1
        else a:= 2
    else if bit(x,r(p)) = 1
        then begin
            p:= dir(p);
            pesqpat(x, p, a);
        end
    else begin
        p:= esq (p);
        pesqpat (x, p, a);
    end
end;

```