Algoritmos Gulosos

Ricardo Dutra da Silva

Universidade Tecnológica Federal do Paraná

Codificação de mensagens

- Suponha uma mensagem com os caracteres $\Gamma = \{a, b, c, d\}$.
- Os caracteres aparecem com as seguintes frequências:

Γ	Freq.
a	60%
b	25%
c	10%
d	5%

Codificação de mensagens

- Podemos usar um código binário de 2 bits para codificar a mensagem.

Γ	Freq.	CF
a	60%	00
b	25%	01
c	10%	10
d	5%	11

Codificação de mensagens

- Podemos usar menos de 2 bits em média, e obter compressão da mensagem.
- Usamos um código variável.
- Menos bits para caracteres mais frequentes.

Γ	Freq.	CF	CV
a	60%	00	0
b	25%	01	1
c	10%	10	00
d	5%	11	11

Codificação de mensagens

- Problema: código ambíguo.
- O que significa a sequência: 01100?

Γ	Freq.	CF	CV
a	60%	00	0
b	25%	01	1
c	10%	10	00
d	5%	11	11

Codificação de mensagens

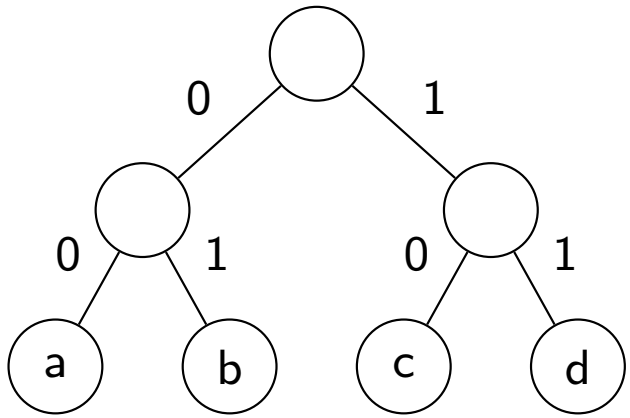
- Código livre de prefixo.

Γ	Freq.	CF	CV	LP
a	60%	00	0	0
b	25%	01	1	10
c	10%	10	00	110
d	5%	11	11	111

- Em LP, 01100 é *aca*.
- $0.6 \times 1 + 0.25 \times 2 + 0.1 \times 3 + 0.05 \times 3 = 1.55$ bits por caractere.

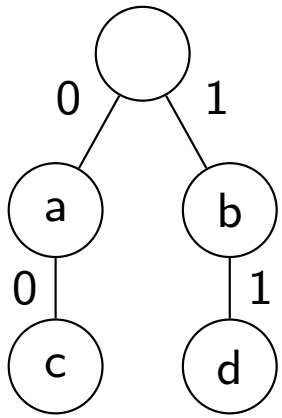
Árvore de Codificação

Γ	Freq.	CF	CV	LP
a	60%	00	0	0
b	25%	01	1	10
c	10%	10	00	110
d	5%	11	11	111



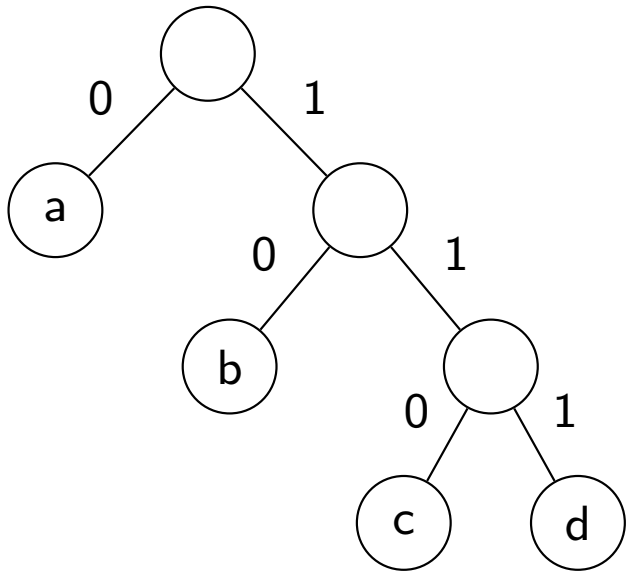
Árvore de Codificação

Γ	Freq.	CF	CV	LP
a	60%	00	0	0
b	25%	01	1	10
c	10%	10	00	110
d	5%	11	11	111



Árvore de Codificação

Γ	Freq.	CF	CV	LP
a	60%	00	0	0
b	25%	01	1	10
c	10%	10	00	110
d	5%	11	11	111



Árvore de Codificação Mínima

Entrada

Um alfabeto Γ e a frequência p_a para todo caractere $a \in \Gamma$.

Saída

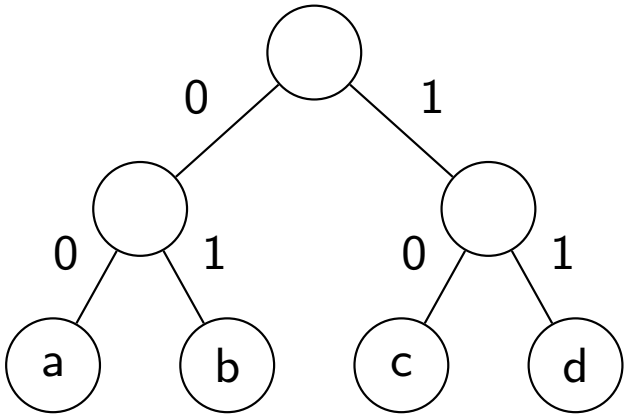
Árvore de codificação T que minimiza a quantidade média de bits para codificação

$$L(T) = \sum_{a \in \Gamma} p_a d_a$$

tal que d_a é a profundidade (número de bits) do caractere na árvore T.

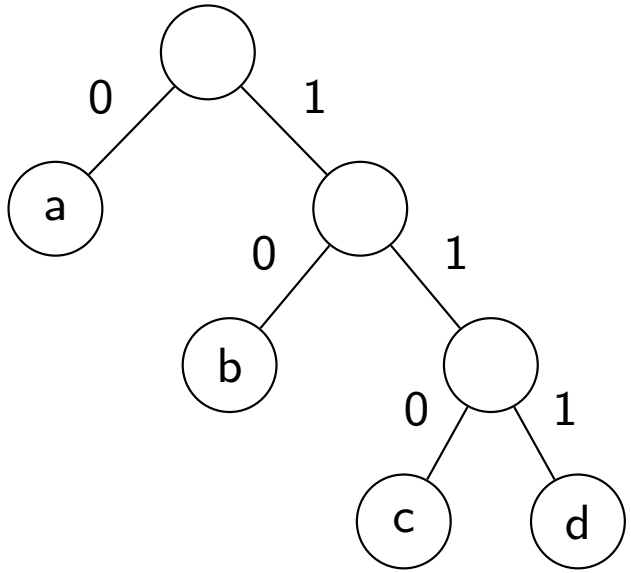
Árvore de Codificação

T_1 :



$L(T_1) = 2.0$

T_2 :



$L(T_2) = 1.55$

Código de Huffman

Solução gulosa:

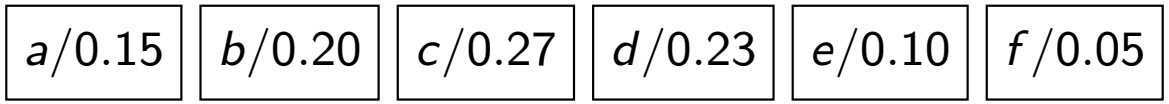
- escolha gulosa;
- subproblemas;
- subestrutura ótima.

Código de Huffman

- Escolha gulosa: combina caracteres $a, b \in \Gamma$ com menor frequência como irmãos em uma subárvore.
- Subproblema: $\Gamma' = \Gamma \setminus \{a, b\} \cup \{c\}$, com $p_c = p_a + p_b$.

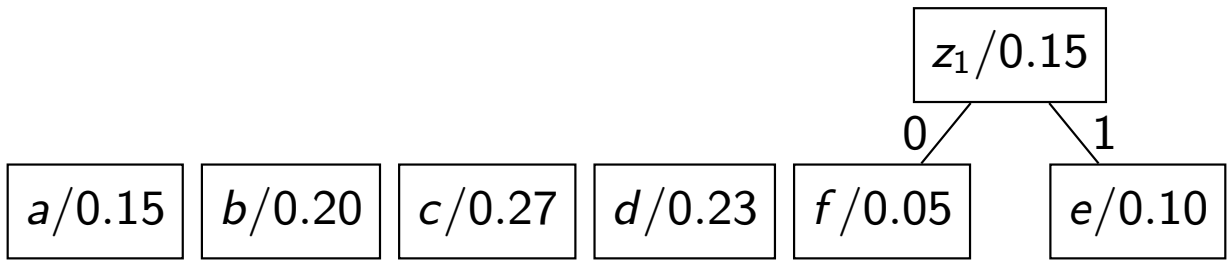
Código de Huffman

Γ	a	b	c	d	e	f
Freq.	0.15	0.20	0.27	0.23	0.10	0.05



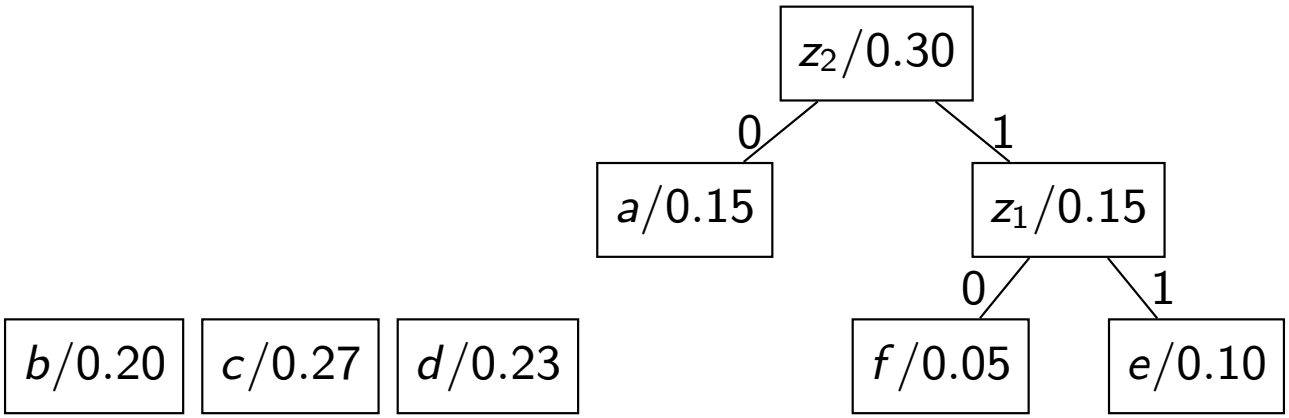
Código de Huffman

Γ	a	b	c	d	z_1
Freq.	0.15	0.20	0.27	0.23	0.15



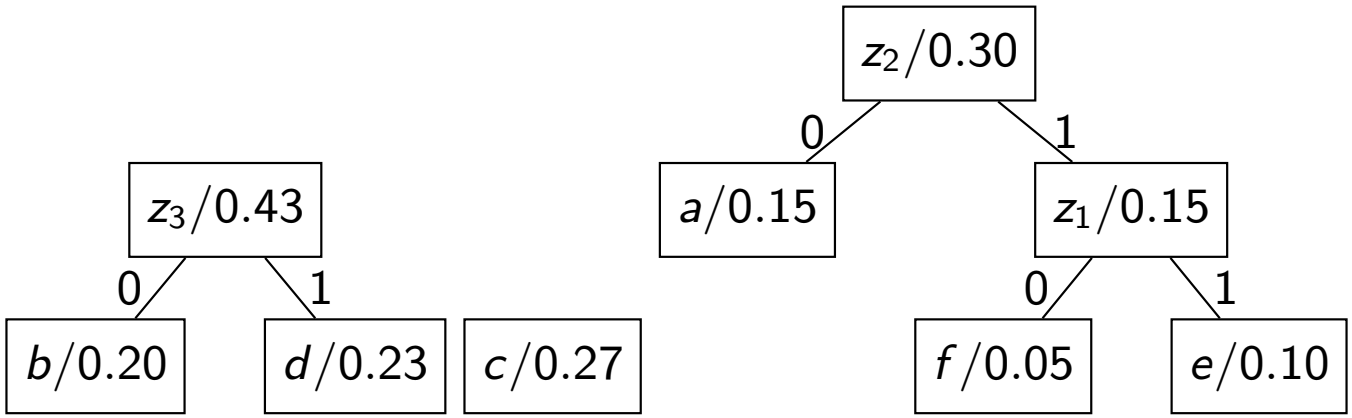
Código de Huffman

Γ	b	c	d	z_2
Freq.	0.20	0.27	0.23	0.30



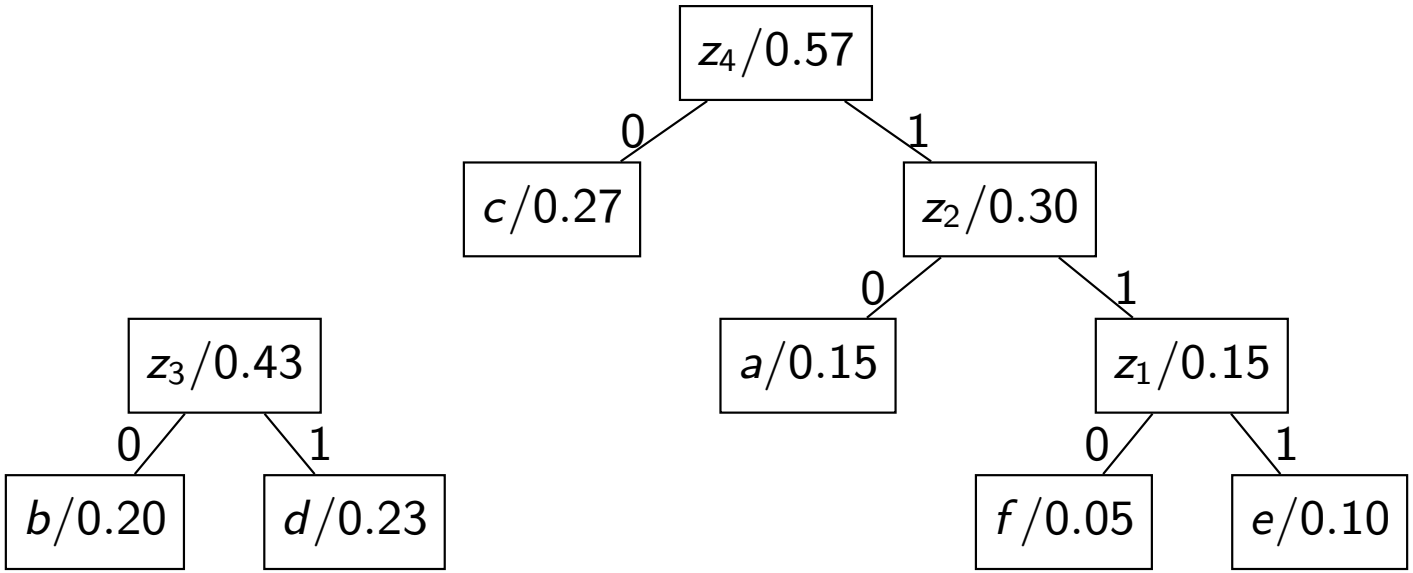
Código de Huffman

Γ	z_3	c	z_2
Freq.	0.43	0.27	0.30



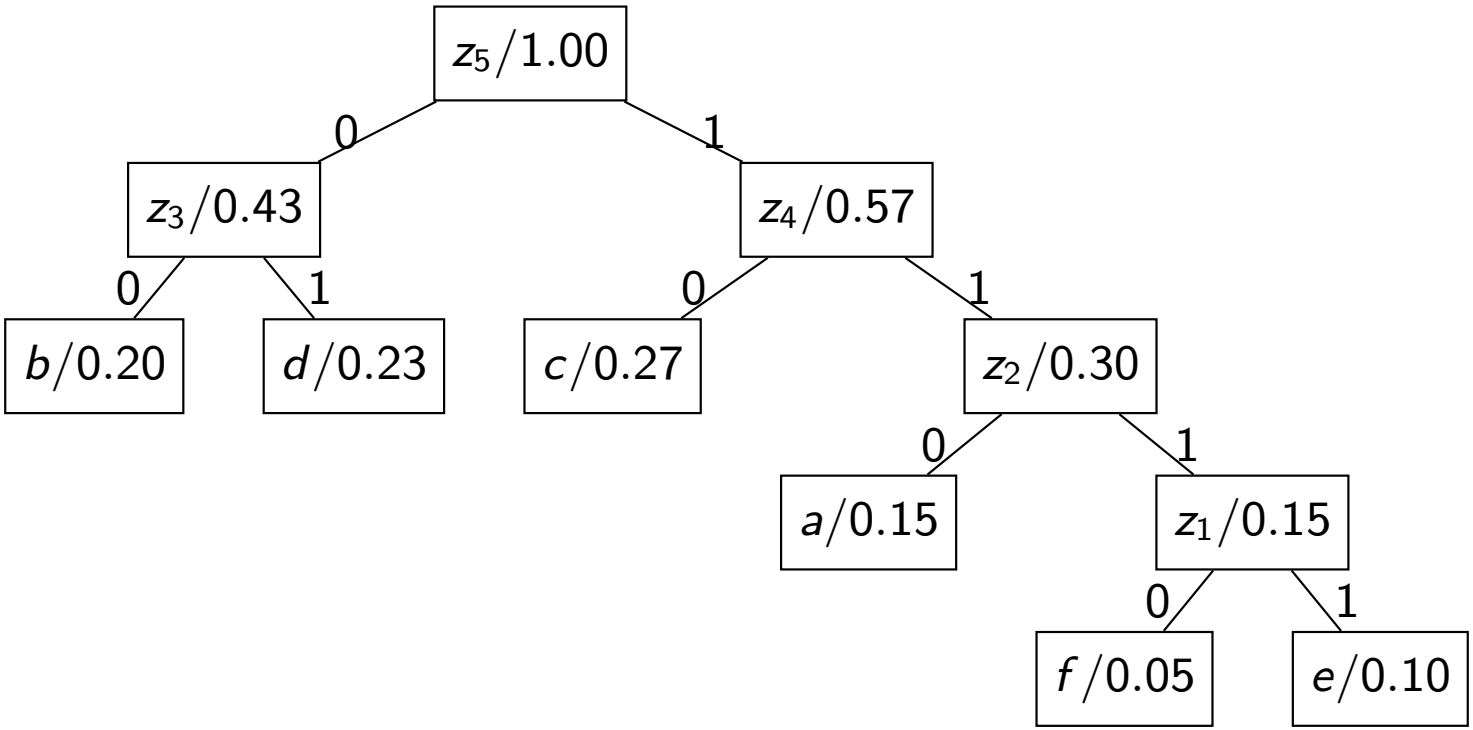
Código de Huffman

Γ	z_3	z_4
Freq.	0.43	0.57



Código de Huffman

Γ	z_5
Freq.	1.00



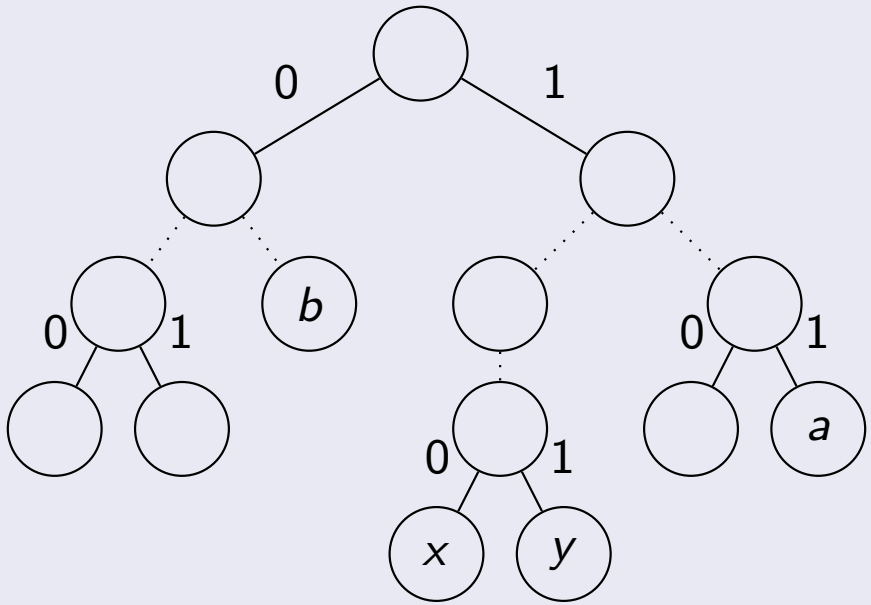
Código de Huffman

Teorema

A escolha gulosa de Huffman pertence a uma árvore de codificação ótima.

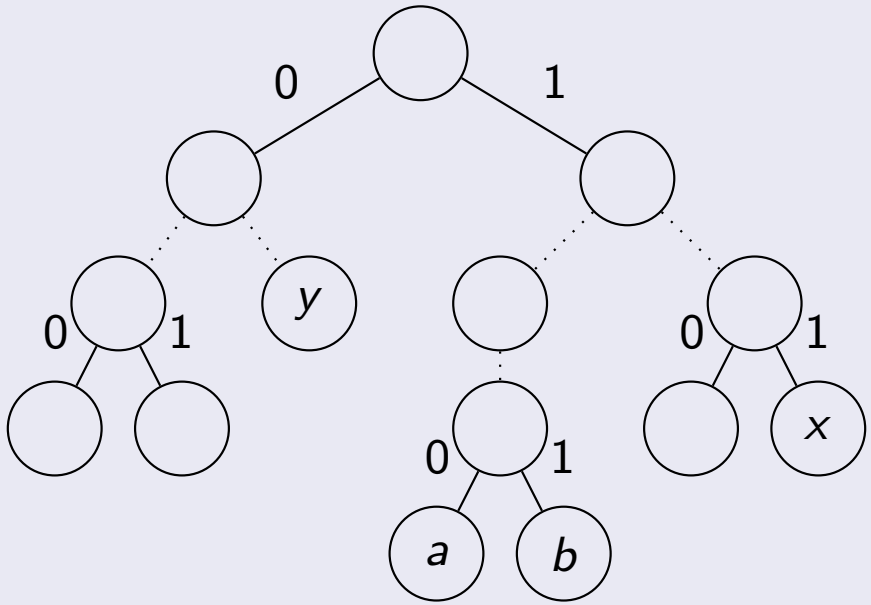
Demonstração.

Suponha uma árvore de codificação ótima qualquer T sem a a escolha gulosa de Huffman (nós de menor frequência a e b não estão no último nível da árvore).



Demonstração.

Construímos uma árvore T' com a e b sendo irmãos no último nível.



Demonstração.

Comparando os custos das árvores

$$
\begin{aligned}
L(T) - L(T') &= \sum_{a \in \Gamma} p_a d_a - \sum_{a \in \Gamma} p_a d'_a \\
&= p_a d_a + p_b d_b + p_x d_x + p_y d_y - p_a d'_a - p_b d'_b - p_x d'_x - p_y d'_y \\
&= p_a d_a + p_b d_b + p_x d_x + p_y d_y - p_a d_x - p_b d_y - p_x d_a - p_y d_b \\
&= (p_x - p_a)(d_x - d_a) + (p_y - p_b)(d_y - d_b).
\end{aligned}
$$

Como todos os fatores são positivos, temos

$$L(T) - L(T') \geq 0$$

e, portanto, T' é ótima. □

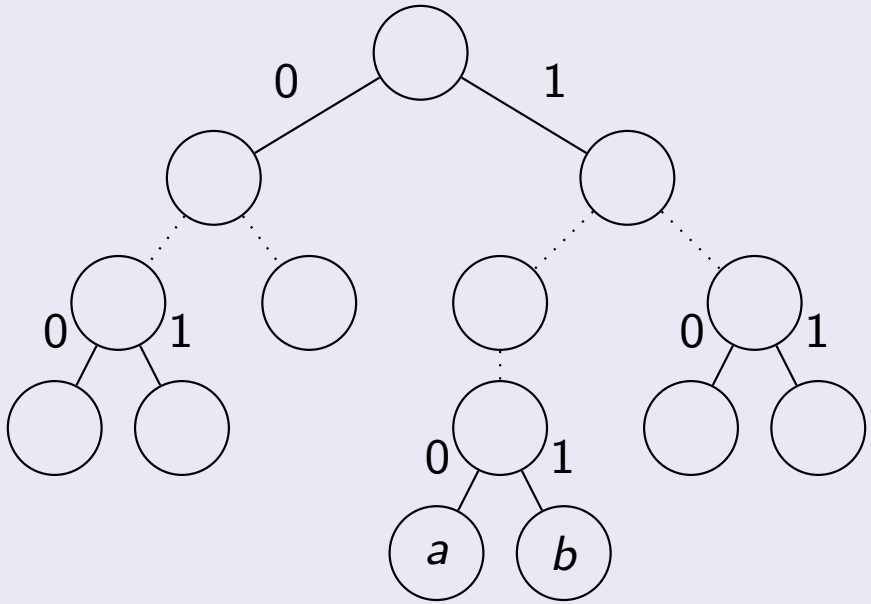
Código de Huffman

Teorema

Se T é uma árvore de Huffman ótima para Γ e $a, b \in \Gamma$ são os caracteres da escolha gulosa, então T' é uma árvore ótima para $\Gamma' = \Gamma \setminus \{a, b\} \cup \{z\}$, com $p_z = p_a + p_b$.

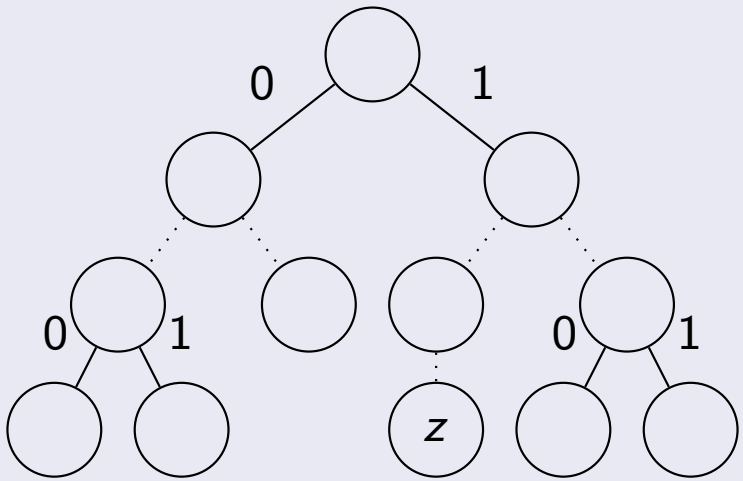
Demonstração.

Seja T uma árvore de Huffman qualquer.



Demonstração.

Seja a árvore T' correspondente.



Demonstração.

A diferença entre o tamanho médio dos códigos dessas duas árvores é dada por

$$
\begin{aligned}
L(T) - L(T') &= \sum_{a \in \Gamma} p_a d_a - \sum_{a \in \Gamma'} p_a d'_a \\
&= p_a d_a + p_b d_b - p_z d_z \\
&= p_a(d_z + 1) + p_b(d_z + 1) - (p_a + p_b) d_z \\
&= p_a d_z - p_a d_z + p_b d_z - p_b d_z + p_a + p_b \\
&= p_a + p_b.
\end{aligned}
$$

Então a diferença entre as árvores é $p_a + p_b$ e

$$L(T) = L(T') + p_a + p_b.$$

Demonstração.

Para argumento de contradição, suponha que T' não é ótima, então exite T'^* com codificação menor, $L(T'^*) < L(T')$.
No entanto, neste caso teríamos

$$\begin{aligned} L(T) &= L(T') + p_a + p_b \\ &> L(T'^*) + p_a + p_b \\ &= L(T*). \end{aligned}$$

Ou seja, teríamos uma árvore T^* melhor do que a árvore ótima T.
Portanto, por contradição, T' é ótima. □

Código de Huffman

Algoritmo: Huffman(Γ)

```
/* Heap mínimo auxiliar H.                                    */
1  para a ∈ Γ faça
2      cria nó n
3      n.p = p_a
4      n.esq = nulo
5      n.dir = nulo
6      Insere(H,n)
7  enquanto |H| > 1 faça
8      a ← Remove(H)
9      b ← Remove(H)
10     cria nó n
11     n.p = a.p + b.p
12     n.esq = a
13     n.dir = b
14     Insere(H,n)
```

Código de Huffman

- Laço da linha 1 é executado $\mathcal{O}(n)$ vezes.
- Linhas 2 a 5 tem operações de tempo constante, no total são executadas $\mathcal{O}(n)$ vezes.
- Linha 6 é uma operação de tempo $\mathcal{O}(\log n)$, no total toma tempo $\mathcal{O}(n \log n)$.
- Laço da linha 7 é executado $\mathcal{O}(n)$ vezes.
- Linhas 10 a 13 tem operações de tempo constante, no total leva tempo $\mathcal{O}(n)$.
- Linhas 8, 9 e 14 são operações de tempo $\mathcal{O}(\log n)$, no total tomam tempo $\mathcal{O}(n \log n)$.
- Portanto, Huffman toma tempo $\mathcal{O}(n \log n)$.