Huffmanův kód

Kódování

Naším cílem bude nalézt 'efektivní' zakódování symbolů z dané **abecedy** X, kde pro každý symbol je předem dána jeho **váha**, dejme tomu jeho relativní četnost v typickém textu v této abecedě. Kód každého symbolu bude jednoznačně daná konečná posloupnost nul a jedniček. Pod 'efektivním' zakódováním rozumíme zhruba řečeno takové, kde kódy jsou co nejkratší, a kde symbol s větší vahou nemá nikdy delší kód než symbol s menší vahou. Přesné měřítko efektivity kódování představuje pojem vážená hloubka definovaný níže.

Budeme navíc požadovat, aby kódování bylo **prefixové**, tj. aby celý kód jednoho symbolu nebyl počátečním úsekem ('prefixem') kódu jiného symbolu. Jinak by nebylo možné text jednoznačně dekódovat.

Binární stromy

Kořenový (zakořeněný) strom je dvojice (T, r), kde T je strom a r jeho vrchol (**kořen**). Pro každý vrchol w existuje jednoznačně určená cesta $P(w)$ z kořene r do w. Hloubka $h(w)$ vrcholu $w \in V(T)$ je délka cesty $P(w)$. Vrchol v je **rodičem** vrcholu w, pokud je sousedem vrcholu w na $P(w)$. V takovém případě je pak vrchol w označován jako **potomek** vrcholu w. Vrchol v je **následovníkem** vrcholu u, pokud u leží na cestě $P(v)$. Speciálně je tedy každý vrchol svým vlastním následníkem.

k-**ární** strom je kořenový strom, jehož každý vrchol má nejvýše k potomků. **Binární** strom je 2-ární strom, v němž každý vrchol kromě kořene je označen jako **levý** nebo **pravý potomek** svého rodiče. **Levý (pravý) podstrom** pod vrcholem v je indukovaný podgraf stromu T na množině všech následovníků levého (pravého) potomka vrcholu v.

Nechť (T, r) je binární strom s listy $\ell_1, \ldots, \ell_p$. kde každému listu je přiřazena **váha** $w(\ell_i)$. **Průměrná vážená hloubka** stromu T je definována jako

$$vh(T) = \sum_{i=1}^{p} w(\ell_i) \cdot h(\ell_i).$$

Souvislost mezi kódy a stromy

Uvažme libovolný binární strom T, jehož listům jsou vzájemně jednoznačně přiřazeny symboly z dané abecedy X. Označme list odpovídající symbolu $y \in X$ jako ℓ_y. Strom T určuje kódování pro abecedu X, a to následujícím způsobem.

Hledáme kód $a(x)$ symbolu $x \in X$. Nechť hloubka odpovídajícího listu ℓ_x je d a označme vrcholy na cestě $P(\ell_x)$ v pořadí od kořene jako $r = v_0, v_1, \ldots, v_h = \ell_x$.

Hledaný kód pak bude posloupnost $a(x) = (a_1, \ldots, a_h)$, kde h je hloubka listu ℓ_x, všechny a_i jsou nuly nebo jedničky a

$$a_i = 0 \iff \text{vrchol } v_i \text{ je levým potomkem vrcholu } v_{i-1}.$$

Huffmanův algoritmus

Vstupem Huffmanova algoritmu je abeceda X spolu s přiřazením vah $w(x)$ symbolům $x \in X$. Jeho výstupem je strom H, jehož listy jsou vzájemně jednoznačně přiřazeny symbolům v X a jeho průměrná vážená hloubka je mezi všemi takovými stromy **minimální**. (Při jejím výpočtu samozřejmě váhu každého listu definujeme jako váhu odpovídajícího symbolu.)

Kód určený stromem H se nazývá **Huffmanův kód** pro abecedu X a váhy w. Je ale třeba mít na paměti, že H (a tím pádem ani odpovídající kód) nejsou jednoznačně určeny.

Zvolme uspořádání $\{ x_1, \ldots, x_p \}$ symbolů abecedy X. Algoritmus pracuje s posloupností kořenových stromů s váženými listy, která se v každém kroku mění. Výchozí posloupnost $M_0 = (T_1, \ldots, T_p)$ je tvořena triviálními stromy T_i o jediném vrcholu x_i s vahou $w(x_i)$.

Algoritmus skončí, až když naše pracovní posloupnost bude obsahovat jediný strom, a tím bude hledaný Huffmanův strom H. Popišme k-tý krok algoritmu.

Nechť

$$Q_{k-1} = \{ (s,t) \mid s < t \text{ a součet } w(T_s) + w(T_t) \text{ je minimální možný} \},$$

kde $w(T_i)$ je součet vah všech listů stromu T_i. V množině uspořádaných dvojic Q_{k-1} najdeme nejmenší prvek (i,j) v lexikografickém uspořádání. Posloupnost M_k vznikne z posloupnosti M_{k-1} následovně:

(1) vypustíme strom T_j,

(2) nahradíme strom T_i stromem, jehož kořenem je nově přidaný vrchol v_k, levým podstromem pod kořenem je T_i a pravým podstromem pod kořenem je T_j (váhy na listech zůstávají beze změny).

Vidíme, že počet prvků pracovní posloupnosti stromů se provedením k-tého kroku skutečně snížil o 1. Algoritmus tedy skončí po $p-1$ krocích. Důkaz, že výsledný strom má požadovanou vlastnost, nebudeme provádět.

Příklad

Nechť X je abeceda tvořená písmeny A, B, C, D, E, H, I s následujícími průměrnými četnostmi:

A	E	G	I	L	R
0.2	0.4	0.06	0.1	0.1	0.14

Najdeme Huffmanův kód pro tuto abecedu a zakódujeme v něm slovo ALERGIE.

V následujícím schématickém znázornění průběhu Huffmanova algoritmu každý řádek zachycuje stav pracovní posloupnosti stromů v každém kroku. Písmena odpovídají stromům na jednom vrcholu. Strom S na více vrcholech je znázorněn symbolem $\boxed{S_1 S_2}$, kde S_1 (resp. S_2) je levý (resp. pravý) podstrom pod kořenem stromu S. Prvky posloupnosti jsou odděleny mezerami, čísla udávají celkovou váhu listů daného stromu, pro přehlednost násobenou stem.

$$
\begin{array}{c}
\mathrm{A}_{20} \quad \mathrm{E}_{40} \quad \mathrm{G}_{6} \quad \mathrm{I}_{10} \quad \mathrm{L}_{10} \quad \mathrm{R}_{14} \\
\mathrm{A}_{20} \quad \mathrm{E}_{40} \quad \boxed{\mathrm{G\ I}}_{16} \quad \mathrm{L}_{10} \quad \mathrm{R}_{14} \\
\mathrm{A}_{20} \quad \mathrm{E}_{40} \quad \boxed{\mathrm{G\ I}}_{16} \quad \boxed{\mathrm{L\ R}}_{24} \\
\boxed{\mathrm{A}\ \boxed{\mathrm{G\ I}}}_{26} \quad \mathrm{E}_{40} \quad \boxed{\mathrm{L\ R}}_{24} \\
\boxed{\boxed{\mathrm{A}\ \boxed{\mathrm{G\ I}}}\ \boxed{\mathrm{L\ R}}}_{60} \quad \mathrm{E}_{40} \\
\boxed{\boxed{\boxed{\mathrm{A}\ \boxed{\mathrm{G\ I}}}\ \boxed{\mathrm{L\ R}}}\ \mathrm{E}}_{100}
\end{array}
$$

Z výsledného stromu zjistíme, že Huffmanův kód pro abecedu X vypadá následovně:

A	000
E	1
G	0010
I	0011
L	010
R	011

Kód slova ALERGIE tedy bude 0000101011001000111. Ověřte, že toto slovo je možné jednoznačně dekódovat (díky tomu, že nalezený kód je prefixový).

Jaká je průměrná vážená hloubka $vh(T)$ nalezeného stromu? Vzhledem k tomu, že symbol E má hloubku 1, symboly A, L, R hloubku 3 a symboly G, I hloubku 4, bude

$$vh(T) = 1 \times 0.4 + 3 \times (0.2 + 0.1 + 0.14) + 4 \times (0.06 + 0.1) = 2.36.$$