2.4 Huffmanův kód pro kompresi dat

Jsou dána data obsahující znaky z abecedy C a pro každý znak $c \in C$ je dána četnost $c.freq$ výskytu c v datech. Kódovat znaky můžeme buď slovy stejné délky; délka jednotlivého kódového slova je dána počtem znaků — je to nejmenší k takové, že $|C| \leq 2^k$. V takovém případě je délka komprimovaných dat rovna součinu počtu znaků a délky jednotlivého kódového slova.

Jinou možností je kódovat znaky slovy o nestejné délce. V případě kódových slov o nestejné délce je však třeba, aby žádné kódové slovo pro znak abecedy C nebylo prefixem jiného kódového slova (jinak be se ztížilo dokódování). V tomto případě je délka dat po kompresi rovna

$$\sum_{c \in C} c.freq \cdot |w(c)|,$$

kde $w(c)$ je kódové slovo znaku c a $|w(c)|$ je jeho délka.

Každý kód si můžeme představit jako binární strom T, kde listy jsou ohodnoceny znaky abecedy C, hrany symbolem 0 nebo 1 a to tak, že ohodnocení cesty od kořene stromu k listu c je kódové slovo znaku c. Délka dat po kompresi je pak dána výrazem

$$B(T) = \sum_{c \in C} c.freq \cdot d_T(c),$$

kde $d_T(c)$ je hloubka listu c ve stromě T.

Huffmanův kód je binární kód nestejné délky jehož binární strom T má nejmenší možnou hodnotu $B(T)$.

2.4.1 Konstrukce Huffmanova kódu

Vstup: Máme dánu abecedu C, $n = |C|$, a četnosti $c.freq$ jednotlivých znaků $c \in C$ v textu.

Výstup: Strom T optimálního binárního kódu.

1. Vytvoříme n jednoprvkových stromů T_c, každý kořen je označený $c; c.freq$; $Q := C$; $\mathcal{T} := \{T_c \mid c \in C\}$.

2. Dokud $|Q| \neq 1$, vybereme $x \in Q$ s nejmenší hodnotou $x.freq$ a $y \in Q$ s druhou nejmenší hodnotou $y.freq$;
do Q přidáme nový prvek z, položíme $z.freq := x.freq + y.freq$, a x, y odstraníme z Q;
vytvoříme strom T_z s kořenem z (označeným $z; z.freq$) takto: levý podstrom z je strom T_x, pravý podstrom je strom T_y;
z množiny $\mathcal{T}$ odebereme stromy T_x a T_y a přidáme strom T_z.

3. Pro $Q = \{q\}$ a $\mathcal{T} = \{T_q\}$ je T_q binární strom, který určuje binární kód takto: každou hranu do levého následníka označíme 0, do pravého následníka označíme 1. Položíme $T := T_q$.

2.4.2 Variant. Po každém průchodu bodem 2 má množina Q o jeden prvek méně (totéž platí pro množinu $\mathcal{T}$). Tedy po $n-1$ průchodech bodem 2 algoritmus skončí.

2.4.3 Invariant

Tvrzení. Nechť C je abeceda a $c.freq$, $c \in C$, jsou frekvence výskytů znaků v datech. Nechť x a y jsou dva znaky s nejmenšími frekvencemi. Vytvoříme $C' = (C \setminus \{x, y\}) \cup \{z\}$, kde $z.freq = x.freq + y.freq$ a označíme T' optimální strom (tj. strom s nejmenším $B(T')$) pro C'.

Pak strom T, který jsme dostali z T' nahrazením vrcholu z stromem s kořenem z, levým následníkem x a pravým následníkem y, je optimální strom pro C.

Myšlenka důkazu. Dá se dokázat, že kdykoli ze stromu T' pro abecedu C' vytvoříme strom T tak, že list z s $z.freq = x.freq + y.freq$ nahradíme výše popsaným stromem (kořen z, levý podstrom x, pravý podstrom y), tak

$$B(T) = B(T') + x.freq + y.freq.$$

Nyní k dokončení důkazu potřebujeme vědět, že je vždy možné najít optimální kód pro abecedu C, takový, že v něm znaky x a y mají stejnou délku a liší se pouze v posledním bitu. A to říká následující lemma.

2.4.4 Lemma. Máme dánu abecedu C s frekvencemi $c.freq$. Nechť x a y jsou dva znaky s nejmenšími frekvencemi. Pak existuje optimální kód nestejné délky, kde kódová slova pro x a y mají stejnou délku a liší se pouze v posledním bitu.

Myšlenka důkazu. Označíme T strom optimálního kódu a označíme a, b ty prvky abecedy C, které jsou v poslední hladině stromu T, mají společného bezprostředního předchůdce a $a.freq \leq b.freq$. Platí $x.freq \leq a.freq$ a $y.freq \leq b.freq$.

Jestliže $x.freq = b.freq$, pak všechny prvky x, y, a, b mají stejnou frekvenci a můžeme vyměnit x s a a y s b a dostaneme strom se stejnou hodnotou B.

Předpokládejme, že $x.freq \neq b.freq$. Vytvoříme nový strom T' tak, že vyměníme x s a a y s b. Dá se spočítat, že

$$B(T) - B(T') = (a.freq - x.freq)(d_T(a) - d_T(x)) + (b.freq - y.freq)(d_T(b) - d_T(y)).$$

Výraz na pravé straně je nezáporný. Kladný být nemůže (pak by strom T nebyl optimální — strom T' by měl menší hodnotu $B(T')$); proto je T' také optimální a lemma se dokázáno.

Kapitola 3

Turingovy stroje

Nejprve uvedeme klasický model, který předcházel moderní výpočetní techniku a velmi pomohl k jejímu rychlému vývoji. Jedná se o tzv. Turingův stroj, model zavedený ve 30. letech minulého století Alanem Turingem.

3.1 Deterministický Turingův stroj

3.1.1 Turingův stroj si můžeme představit takto: skládá se

- z řídící jednotky, která se může nacházet v jednom z konečně mnoha stavů,
- potenciálně nekonečné pásky (nekonečné na obě strany) rozdělené na jednotlivá pole a
- hlavy, která umožňuje číst obsah polí a přepisovat obsah polí pásky.

Na základě symbolu X, který čte hlava na pásce, a na základě stavu q, ve kterém se nachází řídící jednotka, se řídící jednotka Turingova stroje přesune do stavu p, hlava přepíše obsah čteného pole na Y a přesune se buď doprava nebo doleva (tato akce je popsána tzv. přechodovou funkcí).

3.1.2 Formální definice. Turingův stroj je $(Q, \Sigma, \Gamma, \delta, q_0, B, F)$, kde

- Q je konečná množina stavů,
- Σ je konečná množina vstupních symbolů,
- Γ je konečná množina páskových symbolů, přitom $\Sigma \subset \Gamma$,
- B je prázdný symbol (též nazývaný *blank*), jedná se o páskový symbol, který není vstupním symbolem, (tj. $B \in \Gamma \setminus \Sigma$),
- δ je přechodová funkce, tj. parciální zobrazení z množiny $(Q \setminus F) \times \Gamma$ do množiny $Q \times \Gamma \times \{L, R\}$, (zde L znamená pohyb hlavy o jedno pole doleva, R znamená pohyb hlavy o jedno pole doprava),
- $q_0 \in Q$ je počáteční stav a
- $F \subseteq Q$ je množina koncových stavů.

3.1.3 Situace TM. *Situace Turingova stroje* (též konfigurace TM, angličtině nazývaná instantaneous description (ID)), plně popisuje obsah pásky, pozice hlavy na pásce a stav, ve kterém se nachází řídící jednotka. Jedná se o

$$X_1\, X_2\, \dots\, X_{i-1}\, q\, X_i\, X_{i+1}\, \dots\, X_k,$$

kde symboly $X_1\, X_2\, \dots\, X_k$ jsou páskové symboly a kromě nich jsou na pásce pouze blanky B; řídící jsednotka je ve stavu q a hlava čte symbol X_i.

3.1.4 Počáteční situace. Na začátku práce se Turingův stroj nachází v počátečním stavu q_0, na pásce má na n polích vstupní slovo $a_1\, a_2\, \dots\, a_n$ ($a_i \in \Sigma$), ostatní pole obsahují blank B a hlava čte pole pásky se symbolem a_1. Tedy formálně počáteční situaci zapisujeme

$$q_0\, a_1\, \dots\, a_n.$$

3.1.5 Krok Turingova stroje. Předpokládejme, že se Turingův stroj nachází v situaci

$$X_1\, X_2\, \dots\, X_{i-1}\, q\, X_i\, \dots\, X_k.$$

Pak na základě přechodové funkce TM v jednom kroku přejde do následující situace a to takto:

Jestliže $\delta(q, X_i)$ není definováno, TM se zastaví.

Jestliže $\delta(q, X_i) = (p, Y, R)$, TM se přesune do stavu p, na pásku místo symbolu X_i napíše symbol Y a hlavu posune o jedno pole doprava. Formálně to zapisujeme takto

$$X_1\, X_2\, \dots\, X_{i-1}\, q\, X_i\, \dots\, X_k \vdash X_1\, X_2\, \dots\, X_{i-1}\, Y\, p\, X_{i+1}\, \dots\, X_k. \tag{3.1}$$

Jestliže $\delta(q, X_i) = (p, Y, L)$, TM se přesune do stavu p, na pásku místo symbolu X_i napíše symbol Y a hlavu posune o jedno pole doleva. Formálně to zapisujeme takto

$$X_1\, X_2\, \dots\, X_{i-1}\, q\, X_i\, \dots\, X_k \vdash X_1\, \dots\, X_{i-2}\, p\, X_{i-1}\, Y\, X_{i+1}\, \dots\, X_k. \tag{3.2}$$

(Jestliže v případě 3.2 je $i = 1$, pak $q\, X_1\, \dots\, X_k \vdash p\, B\, Y\, \dots\, X_k$.)

3.1.6 Výpočet Turingova stroje nad slovem $w = a_1 a_2 \dots a_k$, je konečná posloupnost jeho kroků, která začíná v počáteční situaci $q_0\, a_1\, \dots\, a_k$.

Formálně se jedná o reflexivní a tranzitivní uzávěr $\vdash^\star$ relace $\vdash$ z 3.1.5 (na množině všech situací daného Turingova stroje).

Jestliže během výpočtu Turingova stroje nad slovem w se Turingův stroj dostane do jednoho z koncových stavů $q' \in F$, říkáme, že se TM *úspěšně zastavil.* Obsah pásky při úspěšném zastavení je *výstupem* TM, nad vstupem $w = a_1\, a_2 \dots\, a_n$.

Jestliže se během výpočtu Turingův stroj zastaví ve stavu, který není koncový, říkáme, že se TM *neúspěšně zastavil.*

3.1.7 Definice — jazyk přijímaný TM. Vstupní slovo $w \in \Sigma^\star$ je *přijato* Turingovým strojem M, jestliže se Turingův stroj na slově w úspěšně zastaví.

Množina slov $w \in \Sigma^\star$, která Turingův stroj přijímá, se nazývá *jazyk přijímaný* M a značíme ji $L(M)$.

3.1.8 Definice — funkce realizovaná TM. Je dáno zobrazení $f\colon \Sigma^\star \to \Sigma^\star$. Řekneme, že TM M *realizuje* zobrazení f, jestliže pro každé $w \in \Sigma^\star$, pro které je $f(w)$ definováno, se M úspěšně zastaví s výstupem $f(w)$ (tj. $q_0 w \vdash^\star \alpha q_F \beta$, kde $\alpha\beta = f(w)$). Pro w, pro něž $f(w)$ není definováno, se M zastaví neúspěšně.

V případě, že f je funkce $f\colon \mathbb{N}^k \to \mathbb{N}$, tj. přiřazuje k-tici přirozených čísel $(n_1, n_2, \dots n_k)$ přirozené číslo $f(n_1, \dots, n_k)$, je vstupem TM slovo $w = 0^{n_1}10^{n_2}1\dots10^{n_k}$. TM realizuje funkci f, jestliže se úspěšně zastaví nad slovem w v situaci, kdy na pásce je slovo $0^{f(n_1,\dots,n_k)}$. Jestliže vstupní slovo není ve tvaru $0^{n_1}10^{n_2}1\dots10^{n_k}$ nebo funkce není definovaná, TM se zastaví neúspěšně.

Poznámka. Někdy se požaduje, aby při úspěšném zastavení hlava TM četla první symbol slova $f(w)$, resp. $0^{f(n_1,\dots,n_k)}$; my to nevyžadujeme; chceme pouze, aby na pásce zbylo slovo $f(w)$, resp. $0^{f(n_1,\dots,n_k)}$ na sousedních polích pásky.

3.1.9 Časová složitost Turingova stroje je parciální zobrazení $T(n)$ z množiny všech přirozených čísel do sebe definované:

Jestliže pro nějaký vstup délky n se Turingův stroj nezastaví, $T(n)$ není definováno. V opačném případě je $T(n)$ rovno maximálnímu počtu kroků, po nichž dojde k zastavení Turingova stroje, kde maximum se bere přes všechny vstupy délky n.

3.1.10 Paměťová složitost Turingova stroje $S(n)$. Jestliže pro nějaký vstup délky n Turingův stroj použije nekonečnou část pásky (pak se nemůže v konečném čase zastavit), $S(n)$ není definováno. V opačném případě je $S(n)$ rovno největšímu rozdílu pořadových čísel polí, které byly během výpočtu použity, kde maximum se bere přes všechny vstupy délky n.

3.1.11 Výše jsme definovali, co je to jazyk přijímaný TM. Jedná se o množinu $L(M)$ všech slov w na nichž se TM úspěšně zastaví (tj. při výpočtu se dostane do koncového stavu).

Jestliže w je slovo, které v jazyce $L(M)$ neleží, TM se při práci nad ním může neúspěšně zastavit nebo nezastavit vůbec.

3.1.12 Jazyk přijímaný/rozhodovaný Turingovým strojem. Definice. Řekneme, že jazyk L je *přijímán* nějakým Turingovým strojem, jestliže existuje TM M takový, že $L = L(M)$.

Řekneme, že Turingův stroj *rozhoduje* jazyk L, jestliže tento jazyk přijímá a navíc se na každém vstupu zastaví.

3.1.13 Poznámky.

- Každý jazyk, který je rozhodován Turingovým strojem, je také tímto Turingovým strojem přijímán. Naopak to ale neplatí. Uvidíme, že existují jazyky, které jsou přijímány nějakým Turingovým strojem, ale neexistuje Turingův stroj, který by je rozhodl.
- Základní model Turingova stroje, tak jak jsme ho uvedli v minulých odstavcích, není jediným modelem. Jiná varianta Turingova stroje pracuje s nekonečnou páskou s pevným levým okrajem. U tohoto modelu se TM neúspěšně zastaví i v případě, že hlava čte nejvíc levé pole pásky

a přechodová funkce nařizuje pohyb hlavy doleva. Počáteční situace TM s pevným levým krajem má vždy vstupní slovo napsané na začátku pásky (tj. od levého okraje).

Další varianty umožňují hlavě Turingova stroje aby se nepohnula. To znamená, že přechodová funkce δ je parciální zobrazení z $(Q \setminus F) \times \Gamma$ do $Q \times \Gamma \times \{R, L, S\}$, kde symbol S znamená, že hlava čte stejné pole.

Všechny tyto modely jsou ekvivalentní v tom smyslu, že pro každý Turingův stroj M_1 jednoho typu existuje Turingův stroj M_2 jiného typu tak, že oba stroje realizují stejné zobrazení / přijímají nebo rozhodují stejný jazyk.