INFORMATIKA

Teorie informace

EDUARD BARTL

Přírodovědecká fakulta UP, Olomouc

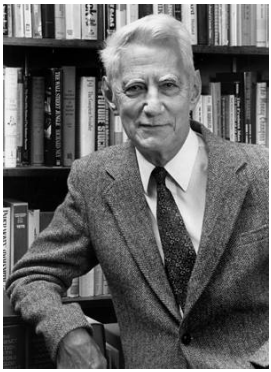
Informatika hraje v naší společnosti zásadní roli, proto je neoddělitelnou součástí výuky na základních a středních školách již po mnoho let; osobně pamatuji první nesmělé krůčky v druhé polovině 80. let na základní škole, kterou jsem navštěvoval. Studenti se ve škole (a samozřejmě také doma) naučí spoustu věcí, které s informatikou souvisejí. Naučí se například používat elektronickou poštu, dozví se, jak jsou na paměťovém zařízení organizovány soubory, jak tyto soubory komprimovat, někteří se naučí základům programování. Za tím vším však stojí složité principy, od kterých je běžný uživatel do značné míry odstíněn. Otázkou je, jestli jsme si při využívání těchto principů vědomi významu alespoň základních pojmů, na kterých jsou tyto principy vystavěny.

Jedním z těchto klíčových pojmů je *informace*. Je známo, že množství informace se udává v *bitech*: programátor ukládá informaci o určitém 32bitovém čísle například do proměnné typu uint32, grafik manipuluje s obrazovou informací prostřednictvím obrázků s 24bitovou barevnou hloubkou, správce sítě pracuje s 128bitovou IP adresou apod. S pojmem bit se však v současné době dostanou do kontaktu i lidé, kteří se jinak o informatiku příliš nezajímají. Většina z nich totiž pravděpodobně slyšela o tom, že rychlost stahování dat (nebo bychom také mohli říci informací) z internetu je například 15 megabitů za sekundu, nebo slyšela o tom, že kapacita nějakého paměťového média (třeba paměťové karty v jejich fotoaparátu)

se udává v bajtech, přičemž 1 bajt je tvořen 8 bity. Jsme však schopni říct, co přesně znamená 1 bit?

Tento článek se snaží jednoduchým způsobem vyložit základní poznatky teorie informace, která na zmíněné otázky odpovídá. Podrobný výklad je možné najít v [1, 3, 5, 6].

Jedním z tvůrců teorie informace je vynikající americký matematik Claude Elwood Shannon (1916–2001), viz obr. 1. Svoji výjimečnost dokázal už tím, že studium na prestižní americké univerzitě Massachusetts Institute of Technology (MIT) dokončil ve svých 21 letech. Shannon ovlivnil svojí prací mnoho oblastí informatiky, za nejvýznamnější lze považovat jeho přínos právě v teorii informace.

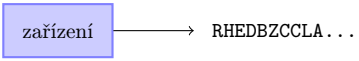


Obr. 1 Claude E. Shannon; zdroj: Wikipedia

Co tedy informace (alespoň z pohledu informatiky) znamená? Nejprve je důležité si uvědomit, že informace je vždy úzce spojena se zdrojem a příjemcem této informace a s komunikačním kanálem, kterým se tato informace přenáší od zdroje k příjemci. Uvedený model je velmi obecný, zahrnuje například přenos informace od mluvčího k posluchači pomocí řeči, vizuální přenos od obrazu vystaveného v galerii k jeho pozorovateli apod. V svém průlomovém článku [4] z roku 1948 však Shannon uvažoval ryze technickou podobu uvedeného modelu (např. elektrický telegraf) a zabýval se problémem, jak definovat informační obsah zprávy generované zdrojem.

Pokusíme se vysvětlit hlavní Shannonovy výsledky na jednoduchém příkladu. Uvažujme situaci, kdy zdrojem je zařízení produkující v určitých časových intervalech znaky nějaké abecedy, například znaky `A`, `B`, `C`, atd., přičemž dohromady je těchto znaků n (tím však nechceme říct, že zaří-

zení vygeneruje celkem n znaků, ale že v daném okamžiku se na výstupu zařízení objeví jeden z n znaků). Schematicky je tato situace znázorněna na obr. 2. Budeme se snažit definovat průměrnou informaci, kterou získá příjemce po přečtení vygenerované zprávy (případně průměrnou informaci přepočtenou na jeden znak zprávy).



Obr. 2 Zařízení produkující znaky určité abecedy

Předpokládejme, že zařízení se chová nepředvídatelně, nevíme tedy, jaký symbol se v daném okamžiku na výstupu objeví. Pokud by se zařízení chovalo předvídatelně, pak by byl informační obsah zjevně nulový. Budeme-li totiž s jistotou vědět, že se na výstupu zařízení v následujícím okamžiku objeví kupříkladu znak `A`, pak skutečnost, že se tak opravdu stane, nám nepřinese žádnou informaci. Co však víme, je pravděpodobnost objevení se daného symbolu. Označme si pravděpodobnost, že se na výstupu zařízení objeví i-tý symbol jako P_i, kde $i \in \{1, 2, \ldots, n\}$. Generování znaků navíc probíhá *nezávisle*, tedy pravděpodobnost objevení se znaku v daném okamžiku není nijak ovlivněna tím, jaký znak se objevil v předchozím okamžiku (někdy se proto mluví o zařízení *bez paměti*).

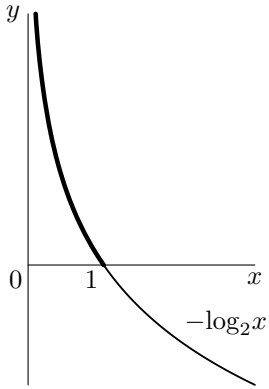
Množství informace bude zřejmě nějakým způsobem souviset s tím, jak moc bude příjemce překvapen, že se na výstupu objeví právě i-tý symbol – čím více bude příjemce překvapen, tím větší bude získaná informace. Tuto tzv. *míru překvapení*,[1] že se na výstupu zařízení objeví i-tý symbol, je možné definovat například jako

$$-\log_2 P_i. \tag{1}$$

Způsob, jakým definujeme tuto hodnotu, má svoje opodstatnění. Grafem funkce $f(x) = -\log_2 x$ je logaritma symetricky převrácená kolem osy x, jak můžeme vidět na obr. 3. Protože do funkce $f(x) = -\log_2 x$ dosazujeme hodnoty pravděpodobností, tzn. čísla mezi 0 a 1, zajímá nás ve skutečnosti pouze část grafu, která je na obrázku tučně zvýrazněna. Můžeme tedy snadno vidět, že se zvyšující se pravděpodobností objevení se nějakého symbolu, míra našeho překvapení klesá. Extrémním případem je

[1] V anglické literatuře se objevuje termín *suprise measure* [1] nebo *self-information* [3]. Prvně uvedený termín lépe vystihuje podstatu věci.

pak situace, kdy se určitý symbol objeví na výstupu zařízení s pravděpodobností 1 (tzn. vždy), pak míra překvapení bude nulová. Naopak, jestliže se bude pravděpodobnost objevení se znaku blížit hodnotě 0, pak se bude míra překvapení blížit nekonečnu. To přesně odpovídá naší představě, jak by měla míra překvapení pracovat.



Obr. 3 Graf logaritmy převrácené kolem osy x

Za základ logaritmu ve vztahu (1) nemusíme nutně volit číslo 2. Při volbě jiného základu budou samozřejmě výše uvedené vlastnosti míry překvapení zachovány. Volba základu logaritmu ovlivní pouze jednotku, ve které budeme míru překvapení počítat. Pokud uvažujeme logaritmus při základu 2, pak je jednotkou *bit* (z angličtiny *binary digit*, česky tedy *dvojková číslice*; proč zrovna dvojková číslice se dozvíme vzápětí). Při volbě přirozeného logaritmu se občas používá jednotka *nat* (z angličtiny *natural unit*), pro dekadický logaritmus pak jednotka *dit* (z angličtiny *decimal digit*) apod.

Průměrnou informaci E na jeden znak zprávy[2] pak budeme definovat jako vážený průměr jednotlivých měr překvapení, přičemž vahami jsou pravděpodobnosti výskytu příslušných znaků. Protože pro součet všech pravděpodobností platí $P_1 + \cdots + P_n = 1$, můžeme psát

$$E = \frac{P_1(-\log_2 P_1) + \cdots + P_n(-\log_2 P_n)}{P_1 + \cdots + P_n} = -\sum_{i=1}^{n} P_i \log_2 P_i. \qquad (2)$$

[2]Pro označení používáme velké písmeno E, protože Shannon průměrnou informaci nazývá *entropií* (česky *neurčitostí*). Při zavedení tohoto názvu se nechal inspirovat pojmem entropie, který známe z fyziky.

Jako konkrétní případ uvažujme zařízení produkující pouze čtyři symboly `A`, `B`, `C` a `D`, všechny se stejnou pravděpodobností

$$P_{\mathtt{A}} = P_{\mathtt{B}} = P_{\mathtt{C}} = P_{\mathtt{D}} = \tfrac{1}{4}.$$

Průměrná informace na jeden znak zprávy je pak rovna:

$$\begin{aligned} E &= -\big(\tfrac{1}{4}\log_2\tfrac{1}{4} + \tfrac{1}{4}\log_2\tfrac{1}{4} + \tfrac{1}{4}\log_2\tfrac{1}{4} + \tfrac{1}{4}\log_2\tfrac{1}{4}\big) = \\ &= -\log_2\tfrac{1}{4} = \log_2 4 = 2 \text{ bity}. \end{aligned}$$

Pokud ale nebude pravděpodobnost objevení se symbolů `A`, `B`, `C`, `D` stejná, průměrná informace musí být menší. Například, platí-li $P_{\mathtt{A}} = \frac{1}{2}$, $P_{\mathtt{B}} = P_{\mathtt{C}} = \frac{1}{8}$ a $P_{\mathtt{D}} = \frac{1}{4}$, pak je průměrná informace rovna 1,75 bitu, jak se můžeme snadno přesvědčit dosazením hodnot pravděpodobností do vztahu (2):

$$\begin{aligned} E &= -\big(\tfrac{1}{2}\log_2\tfrac{1}{2} + \tfrac{1}{8}\log_2\tfrac{1}{8} + \tfrac{1}{8}\log_2\tfrac{1}{8} + \tfrac{1}{4}\log_2\tfrac{1}{4}\big) = \\ &= -\big(-\tfrac{1}{2}\log_2 2 - \tfrac{1}{8}\log_2 8 - \tfrac{1}{8}\log_2 8 - \tfrac{1}{4}\log_2 4\big) = \\ &= \tfrac{1}{2} + \tfrac{1}{8}\cdot 3 + \tfrac{1}{8}\cdot 3 + \tfrac{1}{4}\cdot 2 = \tfrac{14}{8} = 1{,}75 \text{ bitu}. \end{aligned}$$

Zvyšování pravděpodobnosti výskytu jednoho nebo více znaků na úkor ostatních proto snižuje informační obsah zprávy.

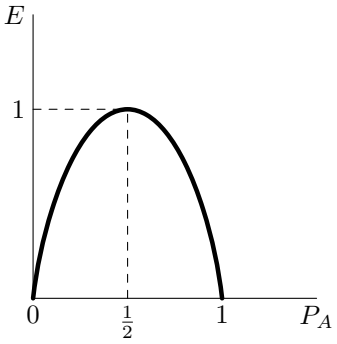
Zajímavá je situace, kdy zařízení produkuje pouze dva symboly `A` a `B` s pravděpodobnostmi $P_{\mathtt{A}} = P_{\mathtt{B}} = \frac{1}{2}$. V tomto případě bychom mohli místo zařízení generujícího symboly uvažovat hod „spravedlivou“ mincí (spravedlivou proto, že hlava i orel padá se stejnou pravděpodobností). Průměrná informace je pak rovna přesně 1 bitu, jak můžeme vidět dosazením pravděpodobností $P_{\mathtt{A}} = P_{\mathtt{B}} = \frac{1}{2}$ do vzorce (2):

$$E = -\big(\tfrac{1}{2}\log_2\tfrac{1}{2} + \tfrac{1}{2}\log_2\tfrac{1}{2}\big) = -\log_2\tfrac{1}{2} = \log_2 2 = 1 \text{ bit}.$$

Můžeme také říci, že 1 bit je množství informace, kterou získáme odpovědí na otázku typu ano/ne, tedy na otázku, která připouští pouze dvě (stejně pravděpodobné) odpovědi. Tyto odpovědi se dají reprezentovat číslicemi 0 a 1, tedy číslem ve dvojkové soustavě. Tím odpovídáme na otázku, proč při použití logaritmu o základu 2 zavádíme jednotku jednoho bitu (zopakujme, že slovo *bit* vzniklo z anglického *binary digit*).

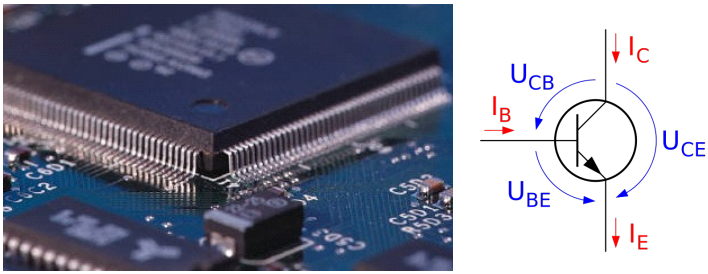
Pokud však použijeme falešnou minci, kdy hlava bude padat častěji než orel, průměrná informace bude menší než 1 bit. Krajním případem je

pak situace, kdy má mince na obou stranách vyraženého orla, hlava tedy nepadne nikdy. Průměrná informace je pak nulová. Celá situace je obecně vyjádřena prostřednictvím grafu na obr. 4.



Obr. 4 Průměrná informace zprávy o dvou znacích (hod mincí)

Poznamenejme, že integrované obvody (obr. 5), ze kterých jsou vyrobeny komponenty počítače, obsahují polovodičové součástky zvané tranzistory. Tyto součástky pracují ve dvou stavech (zapnuto/vypnuto), jsou tedy schopny uchovat pouze informaci jednoho bitu. Důsledkem toho je, že integrované obvody a tedy i samotné počítače interně pracují s daty vyjádřenými pouze pomocí dvou hodnot. Často se však uvažuje osminásobek jednoho bitu nazývaný bajt,[3] řídce se také používá označení oktet. Prostřednictvím jednoho bajtu můžeme reprezentovat $2^8 = 256$ různých hodnot.



Obr. 5 Integrovaný obvod (nalevo) a schematická značka tranzistoru s vyznačenými proudy a napětími mezi jednotlivými konektory (napravo); zdroj: Wikipedia

[3] Fonetický přepis anglického slova *byte*.

Na závěr si ukážeme, jak souvisí Shannonova definice průměrné informace s kódováním (a potažmo kompresí) zprávy. Uvažujme zprávu složenou ze znaků `A`, `B`, `C` a `D`. Znak `A` je ve zprávě obsažen 40krát, znak `B` 20krát, znak `C` 34krát a znak `D` pouze 6krát. Celková délka zprávy je tedy 100 znaků. Přímočarý způsob, jak binárně kódovat znaky zprávy, je takový, že znak `A` budeme reprezentovat řetězcem `00`, znak `B` řetězcem `01`, znak `C` řetězcem `10` a znak `D` řetězcem `11` (zmíněným řetězcům budeme říkat *kódová slova*). Toto kódování je přehledně zobrazeno v tabulce 1. Každý znak je tedy převeden na kódové slovo o dvou bitech, celkově má proto zakódovaná zpráva délku 200 bitů, tzn. 25 bajtů.

znak	kódové slovo
`A`	`00`
`B`	`01`
`C`	`10`
`D`	`00`

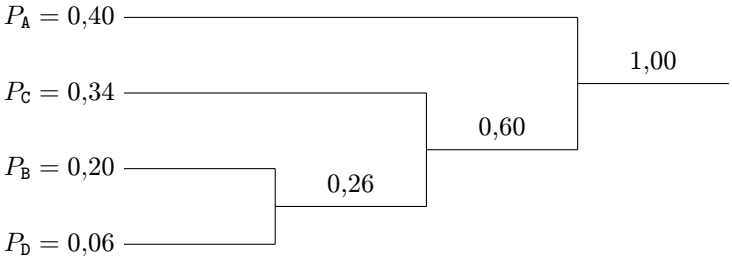
Tab. 1 Jednoduché přiřazení kódových slov

Na první pohled je ale zřejmé, že uvedený způsob není optimální – znak `A` se ve zprávě vyskytuje poměrně často, naopak znak `D` velmi zřídka, oba jsou však kódovány stejným počtem bitů. Bylo by jistě výhodnější znak `A` kódovat menším počtem bitů než znak `D`. Této úvahy využívá *Huffmanovo kódování*,[4] které patří do skupiny tzv. *prefixových kódů*. Vztah mezi průměrnou informací a prefixovými kódy je takový, že hodnota E definovaná vztahem (2) udává teoretickou mez průměrné délky kódového slova, pod kterou se není možné žádným prefixovým kódováním dostat. Bylo dokázáno, že Huffmanovo kódování je optimální, to znamená, že se této mezi přibližuje ze všech prefixových kódů nejvíce. Jinak řečeno, každý prefixový kód má průměrnou délku kódového slova aspoň rovnou hodnotě E a Huffmanův kód se této spodní mezi nejvíce přibližuje.

Jak tedy Huffmanovo kódovaní funguje? Pro velkou délku zprávy můžeme považovat relativní četnost výskytu daného znaku za pravděpodobnost jeho výskytu, můžeme tedy psát $P_{\texttt{A}} = 0,4$, $P_{\texttt{B}} = 0,2$, $P_{\texttt{C}} = 0,34$ a $P_{\texttt{D}} = 0,06$. Konstrukce Huffmanova kódu se provádí tak, že nejprve seřadíme pravděpodobnosti výskytu podle velikosti. Pak sdružíme dvě

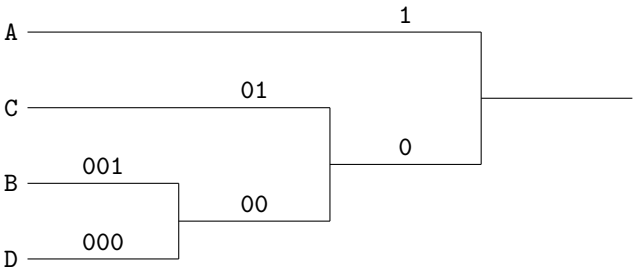
[4] Autorem kódování je David Albert Huffman (1925–1999). Svůj kód publikoval v [2] v době, kdy studoval na MIT.

nejmenší pravděpodobnosti do jedné a výsledku přiřadíme součet těchto pravděpodobností. Takto pokračujeme dokud nedostaneme součet všech pravděpodobností, tedy číslo 1. Postup je přehledně znázorněn na obr. 6.



Obr. 6 Konstrukce Huffmanova kódu, součet pravděpodobností

Nyní zbývá vhodným způsobem určit kódové slovo daného znaku tak, aby platilo, že čím méně pravděpodobný bude tento znak, tím delší bude jeho kódové slovo. Uděláme to tedy tak, že bitem `1` a `0` rozlišíme, jestli se jedná o daný znak nebo o skupinu znaků, jejichž součet pravděpodobností je menší než pravděpodobnost tohoto znaku. Způsob přiřazení kódových slov je znázorněn na obr. 7.



Obr. 7 Přiřazení kódových slov Huffmanovým kódováním

Z tohoto obrázku je patrné, že znak `A` bude kódován slovem `1`, všechny ostatní znaky slovem, které bude začínat `0`. Znak `C` bude kódován slovem `01`, zbývají znaky pak kódem začínajícím `00`. Znak `B` pak bude kódován slovem `001` a poslední zbývají znak, znak `D`, bude kódován slovem `000`.

Přehledně je přiřazení kódových slov uvedeno v tabulce 2.

znak	kódové slovo
A	1
B	001
C	01
D	000

Tab. 2 Přiřazení kódových slov Huffmanovým kódováním

Pokud tedy zpráva začíná například znaky CABAD, její kód bude začínat bity 0110011000. Díky tomu, že je Huffmanův kód prefixový, není potřeba uchovávat informaci o tom, kde končí jedno kódové slovo a začíná druhé. Při dekódování totiž postupujeme tak, že čteme jednotlivé bity zleva doprava a na základě toho, co jsme si řekli v předchozím odstavci, se rozhodujeme, zdali jsme přečetli celé kódové slovo, nebo jestli je potřeba přečíst další bit. Konkrétně tedy v zakódované zprávě začínající posloupností bitů 0110011000 přečteme (zleva) první bit, který říká, že se jistě nejedná o znak A, musí to tedy být jeden ze znaků B, C, D. Po přečtení druhého bitu však okamžitě zjistíme, že se jedná o znak C. Poté pokračujeme stejným způsobem se zbývajícími bity zakódované zprávy.

Vraťme se ještě na chvíli k průměrné informaci. Uvedli jsme, že hodnota E (tedy průměrná informace na jeden znak zprávy) udává teoretickou mez průměrné délky kódového slova libovolného prefixového kódování. Dále jsem také řekli, že se Huffmanovo kódování k této mezi přibližuje nejvíce.

Kódová slova znázorněná v tabulce 1 mají délku 2 bity. Průměrná délka kódového slova na znak (zprávy o 100 znacích, o které jsme mluvili na začátku) je tedy také 2 bity. Průměrná délka kódového slova Huffmanova kódovaní je však menší. Pokud symbolem $d_{\mathtt{A}}$ označíme délku kódového slova znaku A (a podobně pro další znaky), pak můžeme průměrnou délku kódového slova vypočítat jako

$$\frac{P_{\mathtt{A}} \cdot d_{\mathtt{A}} + P_{\mathtt{B}} \cdot d_{\mathtt{B}} + P_{\mathtt{C}} \cdot d_{\mathtt{C}} + P_{\mathtt{D}} \cdot d_{\mathtt{D}}}{P_{\mathtt{A}} + P_{\mathtt{B}} + P_{\mathtt{C}} + P_{\mathtt{D}}} =$$

$$= 0,4 \cdot 1 + 0,2 \cdot 3 + 0,34 \cdot 2 + 0,06 \cdot 3 = 1,86 \text{ bitu.}$$

Průměrná informace na jeden znak je rovna přibližně $E = 1{,}77$ bitu, Huffmanovo kódování se tedy k této hodnotě blíží více než jednoduché kódování uvedené v tabulce 1. Zakódováním zprávy o 100 znacích pomocí Huffmanova kódování tak obdržíme řetězec o délce pouze 186 bitů (oproti zmíněným 200 bitů pomocí jednoduchého kódování). Huffmanovo kódování je tedy skutečně efektivnější.

Huffmanovo kódování má v informatice široké uplatnění. Používá se například v závěrečné fázi algoritmu JPEG, který slouží pro ztrátovou kompresi obrázků (zejména fotografií). Dále je Huffmanovo kódování použito v bezeztrátovém kompresním algoritmu Deflate,[5] který je základem známého obrazového formátu PNG a také souborového formátu ZIP.

Jak jsme mohli v předchozím textu vidět, základní pojmy teorie informace jsou založeny na jednoduchých úvahách. Pro jejich pochopení není třeba složité matematiky, vystačíme si pouze se základy počtu pravděpodobnosti. Navzdory tomu teorie informace zasahuje do mnoha oblastí informatiky (komprese dat, šifrování, teorie složitosti, zpracování přirozeného jazyka, atd.), aplikované matematiky, fyziky a elektrotechniky.

Literatura

[1] *Cover, T. M. – Thomas, J. A.*: Elements of Information Theory. Wiley-Interscience, New York, 1991.

[2] *Huffman, D.*: A Method for the Construction of Minimum-Redundancy Codes. In: Proceedings of the IRE 40 (1952), č. 9, s. 1098–1101.

[3] *Reza, F. M.*: An Introduction to Information Theory. Dover Publications, New York, 2010.

[4] *Shannon, C. E.* : A Mathematical Theory of Communication. Bell System Technical Journal 27 (1948), č. 3, s. 379–423.

[5] *Vajda, I.*: Teorie informace. Nakl. ČVUT, Praha, 2004.

[6] *Wiener, N.*: Kybernetika a společnost. Nakl. ČSAV, Praha, 1963.

(Autorkou úvodní ilustrace je Mgr. Jaroslava Palzerová.)

[5] Algoritmus Deflate, který je kvůli své univerzálnosti někdy nazýván „švýcarským nožem komprese“, kombinuje Huffamnovo kódování a kompresní metodu LZ77. Autorem tohoto algoritmu je Phillip Walter Katz (1962–2000), poněkud tragická postava geniálního, alkoholem však zruinovaného programátora. Více informací o algoritmu Deflate je možné najít na stránkách IEEE Global History Network http://ieeeghn.org/wiki/index.php/History_of_Lossless_Data_Compression_Algorithms.