J003 – Fundamental Concepts of Computer Science and Some Surprising Discoveries: 1. sada

Karel Kubíček

17. dubna 2015

Úloha 1.

V intervalu $I = [2^i, 2^{i+1} - 1]$ se nachází 2^i různých čísel. Pokud by všechna čísla v tomto intervalu byla náhodná, pak by podle definice kolmogorovské složitosti musela být programem zapsatelná kratší posloupností bitů, než je reprezentuje ve dvojkové soustavě.

Chtěli bychom tedy program, který všechna čísla v zadaném intervalu kóduje na slova délky maximálně $\lceil \log_2(2^i + 1) \rceil - 1 = i$. Slov délky maximálně i je $2^i - 1$. Předpokládejme, že program odpovídá injektivnímu zobrazení z I do množiny kódů maximálně délky i. Mohutnost první množiny je 2^i, druhé $2^i - 1$, proto takové zobrazení neexistuje. To je spor s předpokladem o existenci programu, který by mohl kódovat zadaná čísla na kratší. Tedy v zadaném intervalu existuje alespoň jedno číslo, které je náhodné.

Úloha 2.

Pro určení kolmogorovské složitosti musíme zvolit program, který k zadané hodnotě n určí slovo w_n. Program je závislý pouze na hodnotě n, která do něj musí být zakódována (binárně, délka je tedy $\lceil \log_2(n + 1) \rceil$).

Délka $|w_n| = 2^{n^2}$. Pokud z toho vyjádříme n, získáváme $n = \sqrt{\log_2 |w_n|}$. Zadané n do programu kódujeme binárně, složitost programu vzhledem k $|w_n|$ získáme dosažením n do rovnice předchozího odstavce.

$$K(w_n) \leq \left\lceil \log_2 \left(\sqrt{\log_2 |w_n|} + 1 \right) \right\rceil .$$

Úloha 3.

K rekurzivnímu jazyku existuje program, který v konečném čase skončí. Tento program rozhoduje, zdali slovo do jazyka přísluší. Program budeme používat jako podproceduru našeho programu, který generuje postupně všechna slova délky k (kterých je konečně mnoho). Pro každé vygenerované slovo pomocí podprocedury ověří, zdali slovo do jazyka patří. Tento program v konečném čase najde slovo w_k a kolmogorovská složitost je závislá jen na k, které můžeme kódovat binárně. Výsledná horní hranice kolmogorovské složitosti je tedy $\lceil \log_2(k + 1) \rceil$.

Úloha 7.

Mějme program, který iterativně generuje čísla a testuje, zdali jsou prvočíslem. Při tom si počítá počet prvočísel. Takový program pak podle zadané hranice k, kdy má zastavit nalezne k-té prvočíslo. Kolmogorovská složitost tohoto programu p je $K(p) = \lceil \log_2(k+1) \rceil$.

Kolmogorovská složitost k-tého prvočísla lze tedy pomocí zadaného programu ohraničit shora:

$$K(Prim(k)) \leq \lceil \log_2(k+1) \rceil.$$

Déle použijme Prime number theorem. Ten nám říká, jaká je hustota prvočísel vzhledem v limitě do nekonečna ($\pi(n)$ je počet prvočísel v intervalu od 1 po číslo n):

$$\lim_{n \to \infty} \frac{\pi(n)}{n / \ln n} = 1$$

Z toho můžeme vyjádřit v limitě, čemu je rovno zadané prvočíslo ($Prim(k)$ vyjadřuje k-té prvočíslo):

$$\lim_{k \to \infty} \frac{Prim(k)}{k \cdot \ln k} = 1$$

Pomožme si asymptotickou notací, která popisuje funkci vzhledem k asymptotě v nekonečnu. Pak $Prim(k) = \Theta(k \cdot \ln k)$.

Aby bylo zadané číslo náhodné, pak musí mít větší nebo rovnu kolmogorovskou složitost délce svého zápisu. Vyjádřeme složitost pomocí tvrzení z Prime number theorem a získáváme:

$$K(Prim(k)) = \Theta(\lceil \log_2(n \cdot \ln n + 1) \rceil)$$

Náš program však dokáže k-té prvočíslo generovat s kolmogorovskou složitostí $K(Prim(k)) \leq \lceil \log_2(k+1) \rceil$., což je pro dostatečně velká čísla menší hodnota, než program, který přímo do programu zakóduje dané prvočíslo. Výsledkem je, že existuje pouze konečně mnoho náhodných prvočísel.

Úloha 8.

(a) Důkaz provedeme sporem. Předpokládejme, že jazyk L_{kol} je rozhodnutelný. Potom existuje program, který rozhoduje, zdali zadané slovo patří do L_{kol}. Program bychom spouštěli postupně s rostoucími hodnotami $x \in \mathbb{N}$. Dokud by program vracel *false*, tak by platila nerovnost z podmínky jazyka $K(w) \leq Number(x)$. V momentě, kdy vrátí *true* víme, že kolmogorovská složitost byla právě hodnota, pro kterou vrátil program poprvé *true*. Tím jsme získali program, který určuje kolmogorovskou složitost zadaného slova. To je ale ve sporu s větou 5.64 z Theoretical Computer Science (J. Hromkovič).

(b) Zadaný jazyk je rekurzivně spočetný. Program pro slovo w generuje programy v kanonickém pořadí, které na zadané slovo pouští pomocí dovetailingu. Pokud existuje program, který k zadanému w kolmogorovskou složitost určí, pak se nachází v našem seřazeném seznamu programů a jeho výpočet skončí.