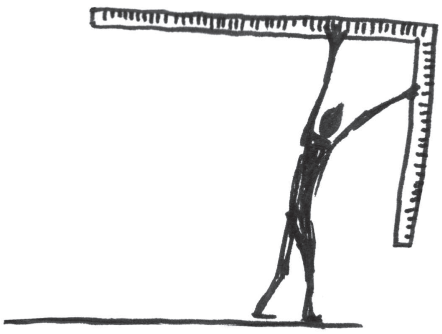
Miara informacji

Eryk KOPCZYŃSKI[*]

[*]Instytut Informatyki, Uniwersytet Warszawski



Jak wiadomo, komputery traktują wszystkie dane, na których działają, jako ciągi bitów, z których każdy może mieć dwie wartości (0 lub 1). Przykładowo, ten tekst jest zapisany w ten sposób, że każdej z liter i pozostałych znaków odpowiada ciąg 8 bitów. Daje to 2^8, czyli 256 możliwości, co w zupełności wystarcza do zapisania wszystkich potrzebnych znaków. Jednak w rzeczywistości różnych znaków występujących w tekście jest mniej. Problem jest więc taki: jak zapisać tekst tak, żeby na każdą jego literę przypadało jak najmniej bitów?

Załóżmy dla przykładu, że w naszym alfabecie są 3 litery: **A**, **B** i **C**. Najprostszy sposób kodowania jest następujący – każdą z liter kodujemy ciągiem dwóch bitów, powiedzmy **A** $= 01$, **B** $= 10$, **C** $= 11$. Ten sposób zapisu (nazwijmy go (1)) daje 2 bity na każdą literę.

Łatwo zauważyć, że powyższy kod da się poprawić: jeśli pierwszym bitem kodu litery jest 0, to już nie musimy pisać drugiego bitu 1. Jeśli założymy, że nasze litery występują tak samo często, to okaże się, że nasz nowy sposób zapisu (2) używa $5/3 = 1,666...$ bitów na każdą literę (co trzecia litera wymaga tylko jednego bitu). Są też możliwe bardziej skomplikowane kody. Czytelnikowi zostawiamy kodowanie (3) o „efektywności” $8/5 = 1,6$ bitów na literę (podpowiedź: kodujemy bloki 5 znaków).

Teraz zwróćmy uwagę na powyższe założenie: *litery występują tak samo często*. A co, jeśli, na przykład, litera **A** występuje z prawdopodobieństwem 1/2, a **B** i **C** z prawdopodobieństwem 1/4? W sposobach zapisu (1) i (3) nic się nie zmienia, ale sposób (2) jest teraz w stanie używać średnio tylko $\frac{1}{2}\cdot 1 + \frac{1}{4}\cdot 2 + \frac{1}{4}\cdot 2 = 3/2 = 1{,}5$ bitów na literę!

Jaka jest najmniejsza możliwa średnia liczba bitów potrzebnych do zapisania ciągu znaków, które występują (niezależnie) z podanym rozkładem prawdopodobieństwa? Odpowiedzi na to pytanie dostarcza teoria informacji Shannona: najmniejszą możliwą średnią liczbą bitów na jeden znak jest *entropia* rozkładu prawdopodobieństwa, zadana wzorem

$$H = -\sum_{i=1,\ldots,n} p_k \log_2 p_k,$$

gdzie $p_1, \ldots, p_n$ są prawdopodobieństwami kolejnych znaków.

Przykładowo, entropia naszego pierwszego rozkładu (3 litery z równym prawdopodobieństwem) wynosi $\log_2 3 = 1,58496....$ (Jeśli wymyślony przez Czytelnika sposób kodowania 5 znaków w 8 bitach jest taki sam, jak Autora, to łatwo go poprawiać, zbliżając się do granicy.) Entropia naszego drugiego rozkładu (**A** występuje 2 razy częściej) to $1,5$, czyli nasz sposób kodowania jest optymalny. Czytelnikowi zostawiamy obliczenie entropii rozkładu, w którym są dwie litery, **A** i **B**, przy czym **B** występuje bardzo rzadko – powiedzmy, raz na 2^{10} liter – i znalezienie odpowiedniego sposobu kodowania, który pozwoli nam się zbliżyć do otrzymanej entropii, np. wymagającego średnio 12 bitów na 2^{10} znaków.

Entropię można rozumieć jako pewnego rodzaju miarę informacji, w naszym przypadku informacji niesionej przez 1 znak w naszym alfabecie. A teoria informacji, poza omówioną wyżej kompresją, ma również zastosowanie w innych dziedzinach informatyki (kryptografia – jak zaszyfrować tekst, by osoba podglądająca nie dostała na jego temat żadnej informacji; generowanie liczb pseudolosowych) i nie tylko (np. fizyka, badanie języka naturalnego)...

Rozwiązanie zadania F 721.
Z założenia o całkowitym zwilżaniu szkła przez wodę wynika, że promień krzywizny warstwy wody między płytkami jest równy połowie grubości warstwy wody: $R = l/2$. Zatem nadwyżka ciśnienia zewnętrznego działającego na płytki wynosi

$$p = \frac{\alpha}{R} = \frac{2\alpha}{l}.$$

Wobec tego siła potrzebna do oderwania płytek wynosi

$$F = pS = \frac{2\alpha}{l},$$

gdzie powierzchnia płytek $S = \frac{m}{\rho l}$. Stąd $F = \dfrac{2\alpha m}{\rho l^2} \approx 800$ N.