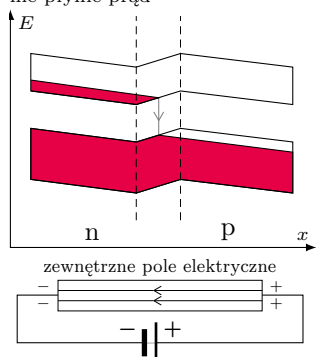
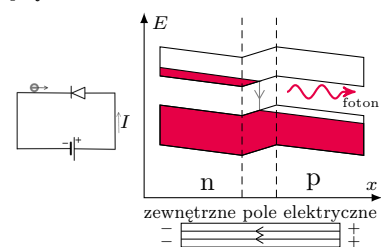


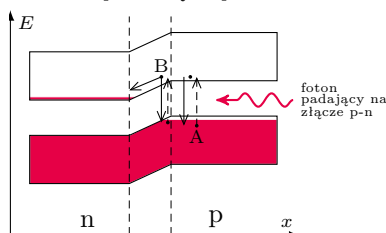
Rys. 5. Złącze p-n w sytuacji, w której nie płynie prąd



Rys. 6. Złącze p-n w sytuacji przepływu prądu



Rys. 7. Emisja fotonu podczas przejścia elektronu przez złącze p-n



Rys. 8. Pochłonięcie energii fotonu przez elektron w złącze p-n

Przykładamy teraz napięcie w drugą stronę (rys. 6).

Widać, że elektrony przemieszczają się tak, że mogą spadać do niższego pasma w warstwie typu p. **Dopóki złącze jest przyłączone do źródła napięcia, elektrony przez nie przepływają.** W ten sposób przez diodę **plynie** prąd. Mamy zatem w miarę proste, a zarazem wystarczająco precyzyjne wyjaśnienie, dlaczego dioda przewodzi prąd tylko w jedną stronę. Możemy również wyjaśnić, dlaczego prąd powstaje tylko po przekroczeniu pewnego napięcia progowego. Na rysunku 4 widać, że obszar przejściowy między obu półprzewodnikami jest pozbawiony elektronów swobodnych. Dopiero odpowiednio silne pole może je przesunąć do krawędzi tego obszaru, skąd już mogą spadać i poruszać się dalej.

Dlaczego dioda może świecić? Podobnie jak w atomie, elektrony spadając na niższy poziom, muszą pozbyć się nadmiaru energii. W niektórych półprzewodnikach emitują wtedy foton (rys. 7). W innych może to być fonon. Jak widać, warstwa typu p musi być w diodzie świecącej tak cienka, by nie pochłaniała powstałych fotonów. Złącze p-n jest również podstawą działania ogniw fotowoltaicznych i to także wyjaśnia nasz prosty model. W ogniwach zachodzi proces odwrotny do świecenia diody – padające fotony są pochłaniane przez elektrony. Przy wystarczającej energii fotonów elektrony w warstwie p przechodzą do wyższego pasma (rys. 8).

Te oddalone od strefy przejściowej prawie natychmiast spadną znowu do niższego pasma (elektron A z rys. 8). To samo dotyczy elektronów pochłaniających fotony w warstwie typu n. Jednak elektron na granicy strefy przejściowej (B na rys. 8) ma dwie możliwości. Może spaść jak jego kolega A. Może też „stoczyć się” w lewo i wtedy już nie może spaść do niższego pasma. Właśnie te elektrony powodują, że półprzewodnik typu n staje się zaciskiem ujemnym ogniwa, a typu p – dodatnim. Widać, jak ważne dla działania ogniw fotowoltaicznych jest nachylenie poziomów w strefie przejściowej.

Przedstawiony model jest dużym uproszczeniem złącza p-n, nie może zatem wyjaśnić wielu efektów w nim występujących. Jednak naszym zdaniem wyjaśnia zadowalająco wszystkie najbardziej istotne cechy diody, będąc jednocześnie modelem dość prostym. Zauważmy, że nie tylko nie ma w nim dziur – ani w sensie dosłownym, ani w przenośnym, ale również nie posługujemy się w nim pojęciem dyfuzji, które występuje w profesjonalnym opisie. Jego użycie jest kolejnym dużym utrudnieniem dla laików. Wiedzą oni bowiem ze szkolnej termodynamiki, że dyfuzja jest procesem powolnym. Trudno zatem zrozumieć, że w przypadku złącza p-n działa praktycznie natychmiastowo.

Naszym zdaniem przedstawiony model działania diody może być z pożytkiem wprowadzony do nauczania w szkołach ponadpodstawowych. Pojęcie diody i tranzystora występuje, i słusznie, w obowiązującej podstawie programowej. Można też w podobny sposób wyjaśnić działanie tranzystora polowego, ale to już byłby temat na osobny artykuł.

Co można napisać na odwróconym kalkulatorze?

Tomasz JANISZEWSKI

W książce pt. „Jak UNIX tworzył historię” Brian Kernighan opisuje sytuację z 1972 roku, kiedy to jeden z jego kolegów z Bell Labs zadzwonił do niego i zapytał, jakie słowa może wyświetlić na kalkulatorze trzymany do góry nogami. Na pytanie, jakie litery są dostępne, usłyszał: „BEhILOSZ”. Po chwili kolega otrzymał listę wszystkich słów.

Artykuł na Wikipedii zamieszczony pod hasłem „Calculator spelling” wspomina większy zbiór dostępnych liter, a mianowicie „BEghILOSZ”. Zwróciłem się z tym do Briana Kernighana, który stwierdził, że nie przypomina sobie liter „g” (6) oraz „Z” (2) i pozostawi swoją historię bez zmian.





Rozwiązanie zadania M 1714.

Niech \mathcal{A} oraz \mathcal{B} będą zbiorami możliwych ustawień pewnej liczby gońców na, odpowiednio, białych i czarnych polach, tak aby żadne dwa nie zagrażały sobie nawzajem. Zauważmy, że obrót o 90° zadaje bijekcję między zbiorami \mathcal{A} i \mathcal{B} . Ponieważ gońce z białych pól nie atakują gońców z czarnych pól, zbiór wszystkich dopuszczalnych konfiguracji odpowiada iloczynowi kartezjańskiemu $\mathcal{A} \times \mathcal{B}$, wobec czego jego liczebność to $|\mathcal{A}|^2$.

POSIX (*Portable Operating System Interface for UNIX*) to standard opisujący działanie systemu operacyjnego, mający ułatwić uruchamianie programów na różnych systemach operacyjnych z rodziny UNIX.

POSIX-Extended Regular Expressions:

.	dowolny znak z alfabetu
AB	konkatenacja wyrażeń AB
\b	początek lub koniec słowa
a-z	dowolny znak z zakresu (a-z)
[...]	dowolny znak spośród wymienionych (np. [abc] jeden z a, b, c)
[^...]	żaden spośród wymienionych (np. [^abc] – żaden z a, b, c)
	lub (a b \equiv [ab])
()	grupowanie wyrażenia (np. (abc)* oznacza powtórzenie całego abc)
*	powtórzenie poprzedzającego wyrażenia 0 lub więcej razy
+	powtórzenie poprzedzającego wyrażenia 1 lub więcej razy (a+ \equiv aa*)
{m}	powtórzenie poprzedzającego wyrażenia dokładnie m razy (a{2} \equiv aa)
{m,}	powtórzenie poprzedzającego wyrażenia co najmniej m razy (a{2,} \equiv aaa*)
{m,n}	powtórzenie poprzedzającego wyrażenia co najmniej m, ale co najwyżej n razy (a{2,3} \equiv aa aaa)

Dodatkowo format ten wprowadza predefiniowane grupy znaków, np. [:digit:] \equiv [0-9].

Zakładamy, że na kalkulatorze można wpisywać zera wiodące, np. dopuszczamy słowo „solo”, odpowiadające liczbie 0705. Jeżeli chcemy wykluczyć takie słowa, poprawnym wyrażeniem regularnym będzie [behilos]*[behils].

Algorytmy oparte na przeszukiwaniu z nawrotami zasadniczo sprawdzają wszystkie możliwe dopasowania wzorca do tekstu. Tych może być jednak wykładnicza liczba, np. wyrażenie (a|aa)*b w tekście „aa...ab”. Warto zaznaczyć, że istnieją szybsze algorytmy, które działają w czasie liniowym względem długości przeszukiwanego tekstu, oparte na automatach skończonych (Δ_{18}^9).

W jaki sposób tak szybko udało się znaleźć wszystkie słowa? Było to możliwe dzięki użyciu wyrażenia regularnego do automatycznego przeszukania słownika. W tym artykule postaram się przybliżyć tę tematykę.

Wyrażenie regularne to sposób opisu języka wymyślony przez Stephena Kleenego w latach pięćdziesiątych ubiegłego wieku. Oczywiście języka formalnego, więc przez alfabet (Σ) będziemy tu rozumieli ustalony zbiór znaków. Słowo będzie oznaczać dowolny ciąg znaków z alfabetu ($w = a_1a_2 \dots a_n, a_i \in \Sigma$), a sam język będzie dla nas zbiorem słów. Na przykład komputery operują językiem binarnym, czyli zbiorem słów złożonych z 0 i 1. Innym przykładem są kody pocztowe. Każdy kod to jedno słowo, które składa się z następujących po sobie dwóch cyfr, myślnika i trójki cyfr. W tym przypadku alfabetem są cyfry i myślnik $\Sigma = \{-, 0, 1, \dots, 9\}$.

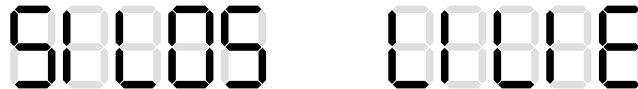
Wyrażenie regularne powstaje poprzez dowolne (w tym wielokrotne) złożenie kilku prostych operacji (| – alternatywy, \cdot – konkatenacji, czyli łączenia słów, * – powtórzenia). Definicję kodu pocztowego można streścić w jednej linii w formie następującego wyrażenia regularnego:

$(0|1|\dots|9) \cdot (0|1|\dots|9) \cdot - \cdot (0|1|\dots|9) \cdot (0|1|\dots|9) \cdot (0|1|\dots|9)$.

W praktycznych implementacjach dodano kilka usprawnień do definicji wyrażeń regularnych, tak aby były jeszcze łatwiejsze do zapisania i czytania. Niestety nie ma jednego wspólnego formatu definiowania wyrażeń regularnych. Chyba najpopularniejszym jest POSIX-Extended Regular Expressions wywodzący się z systemu UNIX: to tam Ken Thompson stworzył pierwszą implementację wyrażeń regularnych w edytorze ed, które następnie były przenoszone do kolejnych narzędzi, takich jak grep, sed czy awk.

Wycinek pełnego standardu znajduje się na marginesie. Korzystając z tej notacji, można by zapisać definicję kodu pocztowego w następującej postaci: $[0-9]\{2\}-[0-9]\{3\}$.

Wracając do historii z początku artykułu – żeby znaleźć wszystkie słowa, które można wyświetlić na odwróconym kalkulatorze, należy sprawdzić, czy słowa z danego języka (np. polskiego) są akceptowane przez następujące wyrażenie regularne: [behilos]+. W słowniku sjp.pl jest 239 takich słów. Przykłady znajdziemy, wpisując na kalkulatorze 50715 lub 31717:



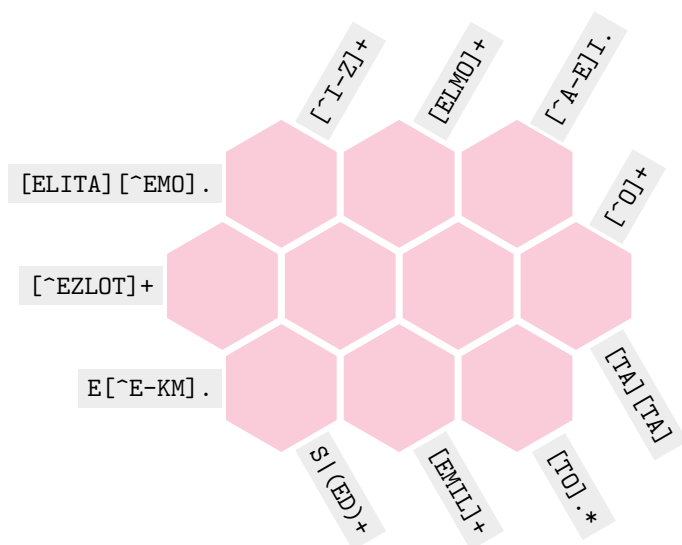
Jeśli w systemie zgodnym z POSIX (np. macOS lub GNU/Linux) zapiszemy wszystkie słowa w pliku slova.txt, to te możliwe do wyświetlenia na kalkulatorze znajdziemy poniższą komendą (ponieważ różne systemy w różny sposób traktują znak nowej linii, ograniczamy słowo przez znak końca słowa (\b) na początku i końcu):

```
grep "\b[behilos]+\b" slova.txt
```

O ile wyrażenia regularne wydają się idealnym narzędziem do przeszukiwania tekstu czy walidacji danych, to jednak należy ich używać ostrożnie. Większość silników stara się znaleźć najdłuższe dopasowanie, a to może wymagać sprawdzania wszystkich spójnych podciągów badanego tekstu. W dniu 20 lipca 2016 roku strona StackOverflow przestała działać właśnie przez takie zachowanie silnika wyrażeń regularnych dla wyrażenia, które miało znaleźć białe znaki na końcu komentarza. Feralny wpis miał 20 000 spacji zakończonych tekstem. Wyrażenie sprawdzało od początku wszystkie spacje, a kiedy napotkało tekst, zaczynało od kolejnej spacji, i tak

$$20\,000 + 19\,999 + 19\,998 + \dots + 3 + 2 + 1 = 200\,010\,000$$

razy – co spowodowało zawieszenie serwera. Co ciekawe, nie jest to najgorsze, co może nas spotkać przy korzystaniu z wyrażeń regularnych. Pesymistyczna złożoność używanych w praktyce algorytmów opartych na przeszukiwaniu z nawrotami (*backtracking*) może dojść do wykładniczej względem rozmiaru przeszukiwanego tekstu.



O ile wyrażenia regularne są potężnym narzędziem do szukania wzorca, to należy jednak pamiętać, że nie pozwalają zdefiniować dowolnego języka. Na przykład za pomocą wyrażenia regularnego nie sprawdzimy, czy wszystkie nawiasy są poprawnie zamknięte.

O językach regularnych pisaliśmy w Δ_{10}^{11} oraz Δ_{14}^6 .

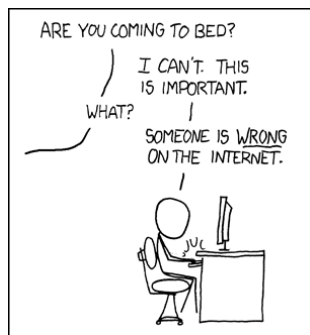
Rozrywkowym zastosowaniem wyrażeń regularnych są krzyżówki. W krzyżówce poniżej zadanie polega na dobraniu liter w taki sposób, aby wszystkie wyrażenia regularne były spełnione.



Ktoś w Internecie się myli! czyli o twierdzeniu Clairaut

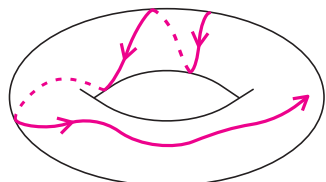
Michał MIŚKIEWICZ* Na jednym z kanałów edukacyjnych na platformie YouTube widziałem ostatnio dość popularny (ponad milion wyświetleń) 10-minutowy film. A w nim dwa błędy, o których chciałbym napisać.

*Wydział Matematyki, Informatyki i Mechaniki, Uniwersytet Warszawski
Instytut Matematyczny Polskiej Akademii Nauk



Duty calls, Randall Munroe (xkcd.com/386)

Film Eugene Khutoryansky'ego *Beauty of Geodesics* można obejrzeć: youtu.be/NfqrCdAjiks



Tak nie wyglądają geodezyjne na torusie

Nie można ślepo wierzyć w treści znalezione w Internecie, ale to już wszyscy wiemy. Zresztą film zawiera dużo pouczających animacji, więc nie zamierzam przestrzegać przed jego oglądaniem. Chciałbym zwrócić uwagę na inny aspekt sprawy – oba błędy wynikają z przesadnego zaufania do symulacji numerycznych. Symulacja to cudowna rzecz, zwłaszcza jeśli pozwala zwizualizować niełatwy fragment matematyki, jakim są krzywe geodezyjne. Ma jednak swoje ograniczenia, i czasami nie uciekniemy przed dobrą staromodną matematyką (taką uprawianą na papierze). Mam nadzieję, że ta teza znajdzie oparcie w poniższych przykładach.

Co to były za błędy?

Będzie tu mowa o *krzywych geodezyjnych*; można było o nich przeczytać m.in. w *Kartografii Płaszczyków* z Δ_{21}^7 . Krzywe geodezyjne na danej powierzchni (np. sferze, walcu albo torusie) odpowiadają prostym na płaszczyźnie i opisują ruch *prosto przed siebie*; w każdym razie na tyle prosto, na ile się da bez odrywania od powierzchni. W dobrym przybliżeniu geodezyjną uzyskamy, jeśli wzdłuż powierzchni poprowadzimy długi a wąski pasek papieru, nie zginając go przy tym.

Około 6. minuty filmu pojawiają się dwie geodezyjne na torusie. Jedna z nich początkowo ciasno nawija się wokół torusa (jak drut na cewce), by po pewnym czasie przenieść się na zewnętrzną stronę i tam już pozostać, okrążając torus po dużym obwodzie. Wbrew obserwacji poczynionej przez narratorkę jest to niemożliwe – geodezyjna, która nawinie się raz, musi to robić w nieskończoność, w równych odstępach czasu przechodząc przez punkt najbliższy środka.

Potem widzimy geodezyjne na nieskończonym lejku, który ku dołowi robi się dowolnie wąski, ale nigdy nie kończy. Spośród kilku geodezyjnych, które początkowo kierują się ku środkowi lejka, niektóre wydostają się z powrotem na górę, ale pozostałe wchodzi głębiej i głębiej. Narratorka ostrzega, że żegnamy się z nimi na zawsze – ale to również nie jest prawdą. Pojedyncza geodezyjna wysłana wprost ku środkowi lejka rzeczywiście jest skazana na wieczny spadek,