Niestandardowe modele obliczeń

Zadania kwalifikacyjne

Adam Michalik

11 czerwca 2014

1 Uwagi ogólne

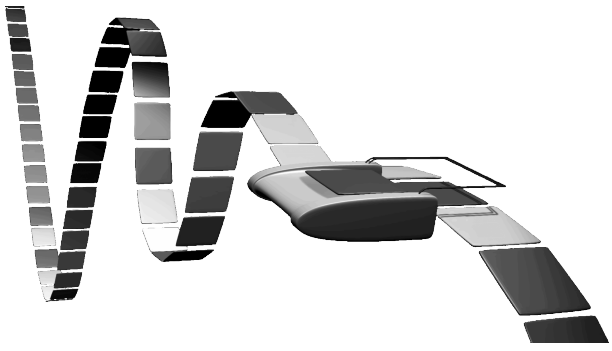
Do kwalifikacji należy rozwiązać wszystkie zadania o maszynach Turinga, oraz kilka zadań matematycznych – niekoniecznie wszystkie, chociaż oczywiście im więcej, tym lepiej. W razie trudności w zmaganiach z zadaniami, proszę napisać o nich do mnie – będę dawał wskazówki i nie będzie wpływać to na kwalifikację, o ile zadanie ostatecznie zostanie rozwiązane. Nie trzeba wysyłać wszystkich rozwiązań naraz, można stopniowo. Jeżeli nie jesteście pewni, czy zadanie jest rozwiązane poprawnie, albo czy wasze rozwiązanie jest wystarczająco przekonujące, to pamiętajcie, że zadania można poprawiać aż do ostatecznego terminu, oraz im wcześniej mi je wyślecie, tym szybciej mogę wam wysłać na ich temat opinię i tym więcej czasu macie na ewentualne poprawki.

2 Maszyny Turinga

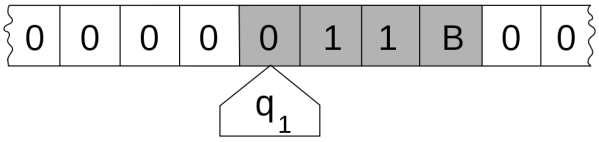
2.1 Wprowadzenie

Podstawowym modelem obliczeń używanym w teorii obliczeń jest maszyna Turinga. Choć początkowo może wydawać się być skomplikowany, w gruncie rzeczy jest to w miarę prosty i intuicyjny model, do czego (mam nadzieję) uda mi się was dość szybko przekonać.

O maszynie Turinga należy mysleć jako o głowicy, która czyta symbol z przypiętej do niej (obustronnie nieskończonej) taśmy, po czym, na podstawie przeczytanego symbolu oraz wewnętrznego stanu, zgodnie z zaprogramowanymi regułami, zapisuje na taśmie nowy symbol, zmienia wewnętrzny stan i przesuwa się o jedną pozycję w lewo lub w prawo.



Rysunek 1: Artystyczna wizja maszyny Turinga



Rysunek 2: Praktyczna wizja maszyny Turinga. Na taśmie, oprócz blanków (czyli startowych symboli, oznaczanych tu jako 0), znajdują się jeszcze symbole 1, 1, oraz B. Maszyna jest w stanie q_1, a głowica znajduje się nad pozycją na lewo od pierwszej jedynki.

Bardziej formalnie, (deterministyczna) maszyna Turinga składa się z:

1. skończonego *zbioru stanów* $Q = \{q_1, \ldots, q_n\}$
2. skończonego *alfabetu* symboli taśmowych Γ, który zawiera wyróżniony symbol B oznaczający blanka
3. zbioru symboli wejściowych $\Sigma \subset \Gamma - \{B\}$ – dane wejściowe maszyny Turinga to ciąg symboli wejściowych zapisany w kolejnych komórkach taśmy (we wszystkich pozostałych komórkach na starcie są blanki)
4. początkowego stanu $q_0 \in Q$
5. stanu akceptującego $f \in Q$
6. funkcji przejścia δ, która opisuje, jak na podstawie aktualnego stanu wewnętrznego i aktualnie wczytanego symbolu decyduje, do jakiego stanu wewnętrznego przejść, jaki symbol zapisać na taśmie, oraz gdzie się przesunąć (w lewo, w prawo, albo w ogóle). Innymi słowy, δ to funkcja ze zbioru par (Stan wewnętrzny maszyny, wczytany symbol) do zbioru trójek (Nowy stan wewnętrzny, symbol do zapisania, kierunek przesunięcia).

Działanie maszyny jest proste: startujemy z głowicą w zerowej komórce taśmi i maszyną w początkowym stanie q_0, po czym w pętli wczytujemy symbol z taśmy i na podstawie aktualnego stanu i wczytanego symbolu, głowica zapisuje nowy symbol, zmienia stan i przesuwa się. Jeżeli w którymś momencie maszyna przejdzie do stanu akceptującego f, to kończy bieg (zatrzymuje się).

2.2 Przykładowa maszyna

Dla przykładu, opiszemy maszynę Turinga REV, która, dla danego na taśmie wejściowego słowa, w wyniku da to samo słowo, ale zapisane w odwrotnej kolejności – czyli, jeżeli damy jej na wejściu słowo 0100110, na wyjściu dostaniemy słowo 0110010.

Alfabet symboli taśmowych to $\Gamma = \{B, P, 0, 1\}$. Alfabet symboli wejściowych to $\Sigma = \{0, 1\}$

Zbiór stanów wewnętrznych maszyny REV to

$$\begin{aligned} Q = \{&Start, OznaczPoczatek, IdzNaKoniecSlowaWejsciowego, WrocDoOstatniegoZnaku \\ &SkopiujPierwsze0, SkopiujPierwsza1, ZapiszPierwsze0, ZapiszPierwsza1, \\ &IdzDoSlowaWejsciowego, Zanies0DoOdwr, Zanies1DoOdwr, \\ &ZnajdzMiejsceDla0WOdwr, ZnajdzMiejsceDla1WOdwr, IdzNaPoczatekOdwr, \\ &UsunPoczatek, Koniec\} \end{aligned} \tag{1}$$

Stan początkowy to oczywiście $Start$, stan końcowy to $Koniec$. Działanie maszyny jest następujące: na początek idziemy jedną pozycję na lewo od słowa wejściowego i zapisujemy w niej symbol P – symbolizuje on dla nas początek taśmy. Następnie, idziemy na koniec słowa wejściowego, bierzemy ostatni znak ze słowa wejściowego i go usuwamy z taśmy, idziemy dwie pozycje w prawo, zapisujemy ostatni znak ze słowa wejściowego – będzie on pierwszym znakiem słowa wyjściowego. Następnie, wracamy do słowa wejściowego (już krótszego o jeden znak), bierzemy ostatni znak, idziemy do słowa wyjściowego, zapisujemy go na końcu słowa wyjściowego i powtarzamy tę procedurę, aż trafimy na znak początek. Wtedy, usuwamy go, po czym kończymy działanie.

Bardziej formalnie, funkcja przejścia δ jest zdefiniowana na następnej stronie. Uważny czytelnik zauważy, że nie dla wszystkich możliwych kombinacji $(Stan, Symbolztasmy)$ wartość funkcji δ jest określona, ale jak łatwo się przekonać, pozostałe kombinacje nigdy nie wystąpią, zatem można je zdefiniować dowolnie.

$$\delta(Start, 0) = (OznaczPoczatek, 0, Lewo) \quad (2)$$
$$\delta(Start, 1) = (OznaczPoczatek, 1, Lewo) \quad (3)$$
$$\delta(OznaczPoczatek, B) = (IdzNaKoniecSlowaWejsciowego, P, Prawo) \quad (4)$$
$$\delta(IdzNaKoniecSlowaWejsciowego, 0) = (IdzNaKoniecSlowaWejsciowego, 0, Prawo) \quad (5)$$
$$\delta(IdzNaKoniecSlowaWejsciowego, 1) = (IdzNaKoniecSlowaWejsciowego, 1, Prawo) \quad (6)$$
$$\delta(IdzNaKoniecSlowaWejsciowego, B) = (WrocDoOstatniegoZnaku, B, Lewo) \quad (7)$$
$$\delta(WrocDoOstatniegoZnaku, 0) = (SkopiujPierwsze0, B, Prawo) \quad (8)$$
$$\delta(WrocDoOstatniegoZnaku, 1) = (SkopiujPierwsze1, B, Prawo) \quad (9)$$
$$\delta(SkopiujPierwsze0, B) = (ZapiszPierwsze0, B, Prawo) \quad (10)$$
$$\delta(SkopiujPierwsze1, B) = (ZapiszPierwsze1, B, Prawo) \quad (11)$$
$$\delta(ZapiszPierwsze0, B) = (IdzDoSlowaWejsciowego, 0, Lewo) \quad (12)$$
$$\delta(ZapiszPierwsze1, B) = (IdzDoSlowaWejsciowego, 1, Lewo) \quad (13)$$
$$\delta(IdzDoSlowaWejsciowego, B) = (IdzDoSlowaWejsciowego, B, Lewo) \quad (14)$$
$$\delta(IdzDoSlowaWejsciowego, 0) = (Zanies0DoOdwr, B, Prawo) \quad (15)$$
$$\delta(IdzDoSlowaWejsciowego, 1) = (Zanies1DoOdwr, B, Prawo) \quad (16)$$
$$\delta(IdzDoSlowaWejsciowego, P) = (UsunPoczatek, B, Zostanwmiejscu) \quad (17)$$
$$\delta(UsunPoczatek, B) = (Koniec, B, Zostanwmiejscu) \quad (18)$$
$$\delta(Zanies0DoOdwr, B) = (Zanies0DoOdwr, B, Prawo) \quad (19)$$
$$\delta(Zanies1DoOdwr, B) = (Zanies1DoOdwr, B, Prawo) \quad (20)$$
$$\delta(Zanies0DoOdwr, 0) = (ZnajdzMiejsceDla0WOdwr, 0, Prawo) \quad (21)$$
$$\delta(Zanies0DoOdwr, 1) = (ZnajdzMiejsceDla0WOdwr, 1, Prawo) \quad (22)$$
$$\delta(Zanies1DoOdwr, 0) = (ZnajdzMiejsceDla1WOdwr, 0, Prawo) \quad (23)$$
$$\delta(Zanies1DoOdwr, 1) = (ZnajdzMiejsceDla1WOdwr, 1, Prawo) \quad (24)$$
$$\delta(ZnajdzMiejsceDla0WOdwr, 0) = (ZnajdzMiejsceDla0WOdwr, 0, Prawo) \quad (25)$$
$$\delta(ZnajdzMiejsceDla0WOdwr, 1) = (ZnajdzMiejsceDla0WOdwr, 1, Prawo) \quad (26)$$
$$\delta(ZnajdzMiejsceDla1WOdwr, 0) = (ZnajdzMiejsceDla1WOdwr, 0, Prawo) \quad (27)$$
$$\delta(ZnajdzMiejsceDla1WOdwr, 1) = (ZnajdzMiejsceDla1WOdwr, 1, Prawo) \quad (28)$$
$$\delta(ZnajdzMiejsceDla0WOdwr, B) = (IdzNaPoczatekOdwr, 0, Lewo) \quad (29)$$
$$\delta(ZnajdzMiejsceDla1WOdwr, B) = (IdzNaPoczatekOdwr, 1, Lewo) \quad (30)$$
$$\delta(IdzNaPoczatekOdwr, 0) = (IdzNaPoczatekOdwr, 0, Lewo) \quad (31)$$
$$\delta(IdzNaPoczatekOdwr, 1) = (IdzNaPoczatekOdwr, 1, Lewo) \quad (32)$$
$$\delta(IdzNaPoczatekOdwr, B) = (IdzDoSlowaWejsciowego, B, Lewo) \quad (33)$$

2.3 Zadanie

Po tym niezwykle męczącym opisie, czas na faktyczne zadanie. Polega ono na napisaniu w dowolnym języku programowania symulatora maszyny Turinga. Program ma wczytywać opis maszyny wraz z wejściem, symulować jej działanie na podanym wejściu, a gdy osiągnie stan końcowy, wypisać wyjściowy stan taśmy.

2.3.1 Format danych wejściowych

Dane wejściowe rozpoczynają się od linii zawierającej liczbę $2 \leqslant Q \leqslant 1000$. Oznacza ona liczbę stanów wewnętrznych maszyny.

Następne Q linii zawiera nazwy poszczególnych stanów. Nazwa stanu składa się z co najwyżej 255 znaków i może zawierać małe lub duże litery alfabetu łacińskiego oraz cyfry.

Następna linia zawiera nazwę stanu początkowego (który jest jednym z wcześniej podanych Q stanów).

Następna linia zawiera nazwę stanu końcowego (uwaga jak wyżej).

Następna linia zawiera jedną liczbę $1 \leqslant G \leqslant 100$, oznacza liczbę symboli taśmowych.

Następne G linii zawiera nazwy symboli alfabetu taśmowego. Symbol alfabetu taśmowego to ciąg co najwyżej 10 znaków i może zawierać małe lub duże litery alfabetu łacińskiego, cyfry, oraz znaki „#$_-+=”.

Następna linia zawiera nazwę symbolu blanka.

Następna linia zawiera jedną liczbę $0 \leqslant N \leqslant 100000$, która jest liczbą podanych relacji przejścia.

Następne N linii zawiera opis funkcji przejścia. Każda linia składa się z pięciu oddzielonych spacjami słów $q\ \alpha\ q'\ \beta\ k$. Oznacza ona, że maszyna znajdująca się w stanie q, po wczytaniu z taśmy symbolu α, ma przejść do stanu q', zapisać na taśmie symbol β, oraz zmienić pozycję zgodnie z nakazem słowa k ($k \in \{Lewo, Prawo, Zostan\}$).

Następna linia zawiera liczbę $0 \leqslant K \leqslant 1000$, oznaczającą liczbę znaków w słowie wejściowym.

Następne K linii zawiera kolejne symbole słowa wejściowego.

2.3.2 Przykładowe dane wejściowe

Przykładowy opis maszyny Turinga (maszyna przechodzi na koniec słowa wejściowego, dopisuje znak X i się zatrzymuje):

```
4
Start
IdzNaKoniec
DopiszX
Koniec
Start
Koniec
3
B
X
Y
B
7
Start X IdzNaKoniec X Zostan
Start Y IdzNaKoniec Y Zostan
Start B IdzNaKoniec B Zostan
IdzNaKoniec X IdzNaKoniec X Prawo
IdzNaKoniec Y IdzNaKoniec Y Prawo
IdzNaKoniec B DopiszX X Zostan
DopiszX X Koniec X Zostan
3
Y
X
Y
```

2.3.3 Symulacja maszyny

Działanie maszyny symulujemy wedle wyżej podanego opisu. Dla uproszczenia, zakładamy, że taśma nie jest obustronnie nieskończona, ale ma 100000 komórek numerowanych od 0, a słowo wejściowe zapisujemy rozpoczynając od komórki o numerze 0. Na początku głowica znajduje się na pozycji 0. Gdyby relacja przejścia w pewnym momencie nakazała wyjście poza zakres $0 - 99999$, należy zwrócić błąd i zakończyć działanie. Podobnie, gdyby w pewnym momencie maszyna znalazła się w stanie oraz wczytała symbol, ale nie istniałaby odpowiednia relacja przejścia dla danej pary, należy zwrócić błąd i zakończyć działanie.

2.3.4 Dane wyjściowe

Na wyjściu program ma wypisać w czytelny sposób zawartość taśmy, od pierwszej pozycji, która nie jest blankiem, do ostatniej, która nie jest blankiem. Innymi słowy, pomijamy początkowe i końcowe blanki.

Należy również wypisać liczbę kroków, które wykonała maszyna w trakcie działania.

2.3.5 Przykładowe dane

Na stronie warsztatów można znaleźć plik z powyższym opisem maszyny dopisującej X, jak również z opisem maszyny odwracającej wejście.

2.4 Drugie zadanie

Drugim zadanie polega na napisaniu pewnych maszyn Turinga

2.4.1 Maszyna kopiująca wejście

Należy opisać maszynę, która dla wejścia będącego ciągiem 0 i 1, skopiuje je obok po znaku dolara. Przykładowo, dla wejścia `001011`, wynikiem działania ma być `001011$001011`. Opis maszyny ma polegać na podaniu alfabetu taśmowego (który oczywiście może być większy niż alfabet wejściowy), wypisaniu stanów i dokładnym opisie funkcji przejścia – to znaczy, w opisanej powyżej postaci wejściowej do naszej implementacji symulatora maszyn Turinga.

2.4.2 Maszyna mnożąca liczby

Druga maszyna, o implementacji której należy pomyśleć, to maszyna mnożąca liczby. Bardziej formalnie, alfabet wejściowy maszyny ma się składać ze znaków $1, x, =$, a maszyna, dla wejścia w postaci $\underbrace{11\cdots11}_{\text{n jedynek}} \times \overbrace{11\cdots11}^{\text{k jedynek}} =$ ma w stanie akceptującym zostawić na taśmie dokładnie $n \cdot k$ kolejno ułożonych jedynek, czyli przykładowo wynikiem działania maszyny na wejściu `111x1111=` ma być `111111111111111`.

Tym razem nie trzeba (choć można) podawać dokładnego opisu maszyny w postaci wejściowej do naszego symulatora – wystarczy podać dokładny alfabet taśmowy, z grubsza opisać stany, oraz to, jak i kiedy maszyna zmienia stany, oraz co robi w każdym stanie. Akceptowalny będzie mniej więcej taki opis, jak powyżej w opisie maszyny odwracającej wejście (chodzi tutaj o opis zaczynający się od słow „działanie maszyny jest następujące").

2.5 Palindromy

Opisać maszynę Turinga, która sprawdza, czy wejściowe słowo jest palindromem parzystej długości. Alfabet wejściowy składa się ze znaków a, b. Maszyna ma sprawdzić, czy zapisane na taśmie na wejściu słowo jest parzystej długości, oraz czytane od tyłu daje to samo słowo. Jeżeli tak jest, to maszyna ma zakończyć bieg z taśmą zawierającą dokładnie jeden znak a, a w przeciwnym wypadku dokładnie jeden znak b. Dokładność opisu może być taka, jak w poprzednim zadaniu.

2.6 Zbiór wszystkich słów jest rekurencyjnie przeliczalny

Napisz w dowolnym języku programowania funkcję, która na wejściu przyjmuje zbiór (albo tablicę, listę, lub wektor, można wybrać, które wygodniej) znaków, a której działanie polega na wypisaniu wszystkich możliwych słów (czyli dowolnych skończonych ciągów), które można ułożyć z tych znaków, każde z nich dokładnie raz. Ponieważ wszystkich słów jest nieskończenie wiele, taka funkcja oczywiście nigdy nie skończy działania, ale nie należy się tym przejmować – wystarczy że będzie wypisywać tak długo, jak tylko jest w stanie (tzn. dopóki nie przekręci się typ liczbowy, albo nie zabraknie pamięci), czyli ma działać przy założeniu nieskończonej ilości pamięci, nieskończonej długości typu liczbowego itd.

3 Trochę matematyki

Z każdego z poniższych działów wypadałoby rozwiązać przynajmniej po dwa zadania.

3.1 Trochę teorii zbiorów

1) Niech X, Y będą dowolnymi zbiorami. Pokazać, że injekcja prowadząca ze zbioru X w zbiór Y istnieje wtedy i tylko wtedy, gdy istnieje surjekcja ze zbioru Y na zbiór X.

2) Dla pewnego zbioru A, niech $P(A)$ oznacza zbiór jego wszystkich podzbiorów. Powiemy, że funkcja $f : P(A) \to P(A)$ jest addytywna, jeżeli $f(X \cup Y) = f(X) \cup f(Y)$ dla dowolnych zbiorów $X, Y \subseteq A$. Czy każda funkcja addytywna ma własność $f(X) = \bigcup_{x \in X} f(\{x\})$?

3) Niech $f : X \to Y$ będzie dowolną funkcją. Przypominamy, że dla dowolnego podzbioru $A \subseteq X$, $f[A]$ oznacza obraz zbioru A przy funkcji f, czyli zbiór:

$$f[A] = \{f(x) | x \in A\} \tag{34}$$

, a dla dowolnego podzbioru $B \subseteq Y$, $f^{-1}[B]$ oznacza przeciwobraz zbioru B przy funkcji f, czyli zbiór:

$$f^{-1}[B] = \{x \in X : f(x) \in B\} \tag{35}$$

Niech $\mathbb{N}^{\mathbb{N}}$ oznacza zbiór wszystkich funkcji $\mathbb{N} \to \mathbb{N}$. Określmy funkcję $F : \mathbb{N}^{\mathbb{N}} \to P(\mathbb{N})$ następująco:

$$F(f) = f^{-1}[\{1\}] \tag{36}$$

Czy F jest injekcją? Czy jest surjekcją? Znaleźć obraz zbioru funkcji stałych, oraz przeciwobraz zbioru $\{\{10\}\}$ przy działaniu funkcji F.

3.2 Trochę indukcji

1) Pokazać przez indukcję, że n-ta liczba Fibonacciego F_n (które są zdefiniowane jako $F_0 = 0, F_1 = 1, F_{n+2} = F_{n+1} + F_n$) spełnia $F_n < 2^n$ dla każdego $n \geqslant 0$.

2) Na kopertę należy nakleić znaczki o łącznej wartości dokładnie n groszy. Mamy nieograniczony zapas znaczków o wartości 5 i 12 groszy. Pokazać, że jeżeli $n \geqslant 44$, to na pewno znaczki da się tak nakleić.

3) Podzielono płaszczyznę n prostymi na pewną liczbę regionów. Pokazać, że można pomalować wszystkie regiony dwoma kolorami tak, żeby żadne dwa regiony, które graniczą ze sobą wspólnym odcinkiem, nie miały tego samego koloru.