# Teoretyczne podstawy informatyki
# zadanie 27

Jarosław Socha

7 kwietnia 2024

## 1 Treść zadania

**Zadanie 27**

***Niech $\chi_L : N \to \{0, 1\}$ będzie funkcją charakterystyczną języka $L \subset \Sigma^*$, która dla $n$-tego słowa w porządku leksykograficznym w zbiorze $\Sigma^*$ zwraca 0 jeśli słowo należy do języka a 1 w przeciwnym przypadku.***

Niech $L$ będzie językiem rozstrzygalnym na jednotaśmowej DTM w czasie $f(n)$. Pokaż, że istnieje program na maszynę RAM obliczający $\chi_L$ w czasie $O(f(n))$.

## 2 Rozwiązanie

Aby udowodnić, że $\chi_L$ da się obliczyć w czasie $O(f(n))$, wystarczy zasymulować program na maszynę DTM na maszynie RAM, nie zwiększając złożoności obliczeniowej.

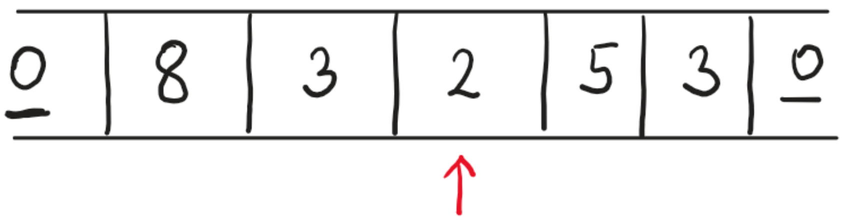
Niech $Q$ to liczba stanów maszyny Turinga, a $k$ to taka liczba, że $k = 2^l$ jest najmniejszą liczbą nie mniejszą od wielkości alfabetu taśmowego maszyny (jako że jest to potęga liczby 2, ułatwi to nam późniejszy dowód). Zauważmy, że dodanie do programu podprocedur o złożoności $O(k)$ lub $O(Q)$ nie zwiększy nam złożoności w notacji $O$, ponieważ obydwie te liczby są stałymi niezależnymi od wielkości wejścia.

### 2.1 Reprezentacja taśmy

Taśmę będziemy symulować za pomocą dwóch liczb, więc za pomocą dwóch rejestrów: $r_l$ (taśma po lewej stronie od głowicy) i $r_r$ (taśma po prawej stronie od głowicy). Każda cyfra w danym rejestrze w systemie numerycznym o bazie $k$ odpowiada literze na taśmie (poza największymi, jeśli $k$ jest większe od alfabetu taśmowego), odpowiednio po lewej i po prawej stronie głowicy.

### 2.2 Rejestry

Oprócz wyżej wspomnianych rejestrów $r_l$ i $r_r$, będziemy używać rejestru $r_q$ aby przechowywać aktualny stan, rejestru $r_d$ aby przechować kierunek na taśmie oraz rejestrów $i_l = k^{\lfloor \log_k r_l \rfloor}$ i $i_r = k^{\lfloor \log_k r_r \rfloor}$ (po-




Rysunek 1: Przykład symulacji taśmy dla alfabetu 9 - elementowego (zero to symbol *blank*). Głowica wskazuje na liczbę 9, więc $r_l = 38$ (przy głowicy jest najbardziej znacząca cyfra), $r_r = 253$, $i_l = 10$ i $i_r = 100$

trzebne do dodawania i usuwania elementów z taśmy). Dodatkowo użyjemy kilku rejestrów w ramach wczytywania i odkładania zmiennych podczas wykonywania podprocedur.

## 2.3 Podprocedury

Zdefiniujemy kilka podprocedur, które ułatwią nam późniejszy dowód.

### 2.3.1 SHL $r$

Wynikiem tej operacji będzie pomnożenie rejestru $r$ przez $k$

1. Pobierz wartość rejestru $r$, dodaj do niej wartość rejestru $r$ i odłóż do rejestru $r$ (mnożenie razy dwa)
2. powtórz powyższe $l$ razy

Nakład czasowy: $O(l)$.

### 2.3.2 SHR $r$

Podobnie do poprzedniej operacji, tym razem będziemy $l$ razy dzielić przez 2.
Powyższe procedury przesuwają liczbę o $k$, dzieląc liczbę na kawałki długości $k$, odpowiadające komórkom taśmy maszyny Turinga. Jako że kawałki te są nie mniejsze od alfabetu, to gdy zapisujemy nowe liczby na "taśmie" to nie interferują one ze sobą nawzajem.

### 2.3.3 DelHigh $r$, $r_x$

Operacja usuwa najbardziej znaczącą cyfrę $r$ i zapamiętuje ją w $r_x$ (jeżeli $r_x$ nie jest podany, to nie zapamiętujemy te cyfry). Po operacji najbardziej znaczącą cyfrą będzie druga cyfra liczby (przykładowo liczba 235 zmieni się w 35). Operacje będziemy wykonywać na rejestrach $r_l$ lub $r_r$, więc w tym przykładzie odpowiadający mu rejestr $i_l$ lub $i_r$ nazwiemy $i$.

1. Odejmij od rejestru $r$ wartość rejestru $i$, zwiększ $r_x$ o 1
2. Powtarzaj powyższe aż $r < 0$,
3. Dodaj jednokrotnie $i$ do $r$, zmniejsz $r_x$ o 1

Nakład czasowy: $O(k)$, ponieważ w najgorszym wypadku usuwaliśmy największą cyfrę odpowiadającą literze alfabetu.

## 2.4 Wczytywanie wejścia

Wejście będziemy wczytywać do rejestru $r_r$, ponieważ główny program zacznie pracę z początku taśmy.

1. Wczytaj symbol taśmy do rejestru $r_x$ (liczba od 1 do $2^k$)
2. **SHL** $r_r$
3. dodaj do $r_r$ wartość rejestru $r_x$

Po powtórzeniu procedury dla każdego symbolu taśmy, w $r_r$ mamy liczbę odpowiadającą danym wejściowym.

## 2.5 Operacje na taśmie

Stałą podczas rozważań jest to, że $r_l$ zawiera elementy po lewej od głowicy, $r_r$ zawiera elementy po prawej, $i_l = k^{\lfloor \log_k r_l \rfloor}$ oraz $i_r = k^{\lfloor \log_k r_r \rfloor}$. Jeżeli po kroku te wartości będą zgodne z nową pozycją głowicy, to osiągnęliśmy symulację pracy maszyny.

### 2.5.1 Sprawdzanie litery i zmiana stanu

Wykonamy podprocedurę **DelHigh** $r_r$, podczas której przy sprawdzaniu, czy wartość $r_r$ jest mniejsza od zera zmienimy kierunek przejścia $r_d$, a następnie stan $r_q$ adekwatnie do funkcji przejścia maszyny, litery w $r_x$ i stanu w $r_q$. Dodatkowo w $r_x$ zapamiętujemy nową wartość cyfry na taśmie. Złożoność takiego sprawdzania to $O(k \cdot Q)$.

### 2.5.2 Stanie w miejscu

Aby zostać w miejscu, wystarczy przywrócić usuniętą liczbę, a więc dodać $r_x$ razy $i_r$ do $r_r$, w co najwyżej $O(k)$ operacjach.

### 2.5.3 Przejście w prawo

1. **SHL** $i_l$
2. do $x_l$ dodaj $i_l$, $r_x$ razy
3. **SHR** $i_r$

Zwiększyliśmy rejestr $i_l$, zmniejszyliśmy $i_r$, a nowy symbol jest teraz najbardziej znaczącą cyfrą $r_l$. Złożoność to $O(k)$.

#### 2.5.4 Przejście w lewo

1. do $x_l$ dodaj $i_l$, $r_x$ razy
2. **DelHigh** $r_l$, $r_x$
3. **SHR** $i_l$
4. **SHL** $i_r$
5. do $x_r$ dodaj $i_r$, $r_x$ razy

Odłożyliśmy nowy symbol do rejestru $r_r$, po czym przenieśliśmy najbardziej znaczącą cyfrę $r_l$ do $r_r$. Złożoność to $O(k)$.

### 2.6 Podsumowanie złożoności

Jak można zauważyć, w każdym kroku dodawane złożoności do operacji taśmowych nie były uzależnione od wielkości danych, jedynie od stałych takich jak wielkość alfabetu czy liczba stanów. Oznacza to, że powyższe operacje nie zwiększą złożoności obliczeniowej w notacji $O$, więc program policzy $\chi_L$ w czasie $O(f(n))$.