

Uebungsblatt 02 - Theoretische Informatik 2

Kügler, Lennart - Informatik - 5236372
`l.kuegler@tu-braunschweig.de`

Rausch, Moritz - Informatik - 5155947
`moritz.rausch@tu-braunschweig.de`

Schäfer, Maximilian - Elektrotechnik - 4763431
`maximilian.schaefer@tu-braunschweig.de`

Viradia, Yash - Informatik - 5275038
`y.viradia@tu-braunschweig.de`

09.05.2023

A2.1

a)

Zustand	Eingabeband	Ausgabeband
q ₀	1#01#01#101	⌊
q ₁	1#01#01#101	⌊
q ₁	1#01#01#101	⌊
q ₂	1#01#01#101	⌊⌊
q ₃	1#01#01#101	⌊1
q ₁	1#01#01#101	⌊1
q ₁	1#01#01#101	⌊1
q ₁	1#01#01#101	⌊1
q ₂	1#01#01#101	⌊1
q ₂	1#01#01#101	⌊0⌊
q ₃	1#01#01#101	⌊01
q ₃	1#01#01#101	⌊01⌊
q ₄	1#01#01#101	⌊01⌊
q ₄	1#01#01#101	⌊01⌊
q ₄	1#01#01#101	⌊01⌊
acc	1#01#01#101	⌊01⌊

- b) Die TM zählt die abgearbeiteten #, wenn diese Anzahl mit der aktuellen Eingabezahl übereinstimmt akzeptiert die TM.

q_0 : Startzustand

- bei \perp oder $\#$ ^{2.anny} akzeptiert werden
- bei $\#$, 0, 1 kommt man zu q_1

q_1 : Suche des nächsten #

- Zahlen werden "übersprungen"
- wenn # gefunden q_2

q_2 : Zählt die # und schreibt das auf das Arbeitsband

- wenn bereits eine 1 vorhanden ist, wird diese zur 0 und rechts wird eine weitere 1 hinzugefügt oder es wird nur eine 1 eingefügt, falls keine 1 vorhanden ist
- diese 1 wird bei Übergang in q_3 geschrieben

q_3 : geht zur ersten (linksten) Position des Arbeitsbandes

- von hier kommt man zum Vergleich (q_4) oder zurück nach q_1 um die Eingabezahl zu "überspringen"

q_4 : Hier wird die aktuell Eingabezahl mit der aktuellen Zahl der # verglichen, stimmen diese überein kommt man zu acc

acc: Akzeptierender Zustand

c) Da das Eingabeband 1x abgearbeitet wird
haben wir eine Laufzeit in $O(n)$,
Der Zeitverbrauch ist daher:

$$NTIME_2(O(n))$$

und der Platzverbrauch bei:

$$NSPACE_2(O(n)).$$

Das N steht für eine nicht-deterministische TM
und die tiefgestellte 2 für die 2 Bänder, die
sie verwendet.

2)

a)

Der Algorithmus überprüft für alle $m \in \mathbb{N}$, $1 < m < n$, ob n durch m teilbar ist. Gibt es ein solches m , ist n keine Primzahl. Gibt es kein solches m , ist n eine Primzahl.

b)

Eingabe: $n \in \mathbb{N}$

Ausgabe: true, wenn n prim
false, wenn n nicht prim

```
1.  if ( $n < 2$ )
2.      return false
3.  fi
4.  for  $i = 2$  to  $n-1$  do
5.      if ( $n \bmod i == 0$ )
6.          return false
7.      fi
8.  od
9.  return true
```

c)

Variablen: n, i

2 Variablen \rightarrow 2 Bänder