

Theoretische Informatik: Blatt 7

Abgabe bis 13. November 2015
Assistent: Sacha Krug, CHN D 42

Linus Fessler, Markus Hauptner, Philipp Schimmelfennig

Aufgabe 19

Sei A eine k -Band-TM. Wir konstruieren nun nach und nach eine 1-Band-TM B . Die Vorgehensweise ist analog zum Beweis von *Lemma 4.2*

1. Zu jedem Band i von A bis auf das Eingabeband fügen wir ein zweites Band i' direkt darunter ein. Dieses soll nur die Symbole $\{\dot{\varsigma}, \sqcup, \uparrow\}$ beinhalten und \uparrow soll nur einmal auf diesem Band vorkommen. Der Pfeil indiziert die Position des Kopfes, der in A auf Band i zeigt. Die Anzahl an Arbeitsbändern verdoppelt sich.
2. Jetzt fassen wir sämtliche Arbeitsbänder i und i' , $i \in \{1, 2, \dots, k\}$ zu einem Band zusammen. Der j -te Eintrag auf den Arbeitsband von B muss also die Infos über dem j -ten Eintrag sämtlicher Arbeitsbänder i und i' beinhalten. Das Eingabeband von A lassen wir unangetastet. Dadurch vergrößert sich unser Arbeitsalphabet.

$$\Gamma_B = \underbrace{\Sigma_A \cup \{\dot{\varsigma}, \$\}}_{(1)} \underbrace{\cup}_{(2)} \left(\underbrace{\{\Sigma_A \cup \{\dot{\varsigma}, \sqcup\}\}}_{(3)} \times \underbrace{\{\sqcup, \uparrow\}}_{(4)} \right)^k$$

- (1) Alle Zeichen, die im Eingabealphabet stehen können.
 - (2) wir lesen entweder ein Zeichen auf dem Eingabeband oder auf dem Arbeitsband.
 - (3) Zeichen auf dem i -ten Arbeitsband von A .
 - (4) Zeichen auf dem i' -ten Arbeitsband (siehe oben).
3. (a) B liest einmal den Inhalt des Eingabebandes von links nach rechts, bis alle k Kopfpositionen von A gefunden wurden und speichert dabei in ihrem Zustand die k Symbole, die sie bei den k Köpfen von A gelesen hat. (Das sind die Symbole, die auf der Spur über ' \uparrow ' stehen.) Die Anzahl Zustände in B ist also grösser als die in A .
 - (b) Jetzt kennt B das ganze Argument (B hat den zu A passenden Zustand.) der Transitfunktion von A und kann entsprechend die Köpfe bewegen (= Symbole auf der $2i$ -ten Spur ersetzen). Diese Änderung kann B in einem Lauf von rechts nach links ausführen.

Aufgabe 20

- (a) $e(n) = 2^n$

Wir konstruieren eine 2-Band Turingmaschine M , wobei *Band 0* das Eingabeband und *Bänder 1–2* die Arbeitsbänder sind. M bekommt als Eingabe das Wort 0^n auf *Band 0*. Zu Beginn schreibt M eine 0 auf *Band 1*. Solange der Lesekopf des Eingabebandes nicht $\$$ liest:

1. Gehe auf *Band 1* nach links bis $\dot{\varsigma}$.
2. Gehe auf *Band 2* nach links bis $\dot{\varsigma}$
3. Lies Zeichen auf *Band 1*. Schreibe für jede gelesene 0 auf *Band 1* 00 auf *Band 2*. Für ein \sqcup schreibe ein \sqcup .
4. Gehe auf beiden Bändern nach links und kopiere Inhalt von *Band 2* auf *Band 1* einschließlich bis Zeichen \sqcup .
5. Rücke mit Lesekopf nach rechts.

Das Ergebnis steht dann auf *Band 2* bis zum ersten \sqcup .

Auf diese Art generieren wir 2^n Nullen. Für n Nullen der Eingabe lesen wir pro Schritt 2^i Nullen. Das Schreiben geschieht jeweils in $\mathcal{O}(1)$.

$$\sum_{i=1}^n 2^i = 2^{n+1} - 2 \in \mathcal{O}(2^n)$$

Folglich ist $e(n)$ zeitkonstruierbar.

(b) $f(n) = fib_n$

Wir konstruieren eine 3-Band Turingmaschine M , wobei *Band 0* das Eingabeband und *Bänder 1–3* die Arbeitsbänder sind. M bekommt als Eingabe das Wort $w = 0^n$ auf *Band 0*. Wir unterscheiden drei Eingaben w .

Fall 1: $w = \lambda$.

In diesem Fall ist $n = 0$. M schreibt $0^{fib_0} = \lambda$ auf

Fall 2: $w = 0$.

In diesem Fall ist $n = 1$. M schreibt $0^{fib_1} = 0$ auf *Band 1* und hält.

Fall 3: $|w| = n, n \geq 2$. Der Lesekopf auf *Band 0* liegt auf der zweiten 0 und M liest auf *Band 0* von links nach rechts. Für die erste gelesene 0 schreibt M eine 0 auf *Band 2*. Für jede weitere gelesene 0 führt M Schritte 1.–3. aus, bis \$ gelesen wird. Dann ist auf *Band 1* das Ergebnis 0^{fib_n} (und auf *Band 2* $0^{fib_{n-1}}$ und auf *Band 3* $0^{fib_{n-2}}$).

1. M überschreibt *Band 3* mit dem Inhalt von *Band 2* ($fib_{i-2} \leftarrow fib_{i-1}$).

2. M überschreibt *Band 2* mit dem Inhalt von *Band 1* ($fib_{i-1} \leftarrow fib_i$).

3. M schreibt den Inhalt von *Band 3* in *Band 1* (konkateniert also die Nullen auf *Band 1* mit den Nullen von *Band 3*) ($fib_i \leftarrow fib_{i-1} + fib_{i-2}$).

In den Fällen 1 und 2 ist die Laufzeit konstant, also ist $f(n) \in \mathcal{O}(1)$. Im Fall 3 schreiben wir bei der i -ten Ausführung, $i \leq n-2$ (weil wir auf der zweiten 0 starten), fib_{i-2} Nullen auf *Band 3*, fib_{i-1} Nullen auf *Band 2* und fib_{i-2} Nullen auf *Band 1*, also pro Schritt $2fib_{i-2} + fib_{i-1} \leq 3fib_{i-1}$ Nullen. Insgesamt ergibt das also

$$\begin{aligned} 3 \sum_{i=2}^n fib_{i-1} &= 3 \sum_{i=1}^{n-1} fib_i && \text{(Indexverschiebung)} \\ &= 3 \sum_{i=0}^{n-1} fib_i && (fib_0 = 0) \\ &= 3(fib_{n+1} - 1). && \left(\sum_{i=0}^n fib_i = fib_{n+2} - 1\right) \\ &= 3(fib_n + fib_{n-1} - 1) && \text{(Definition } fib_n) \\ &\leq 3(2fib_n - 1) \end{aligned}$$

$3(2fib_n - 1) = 6fib_n - 2 \in \mathcal{O}(fib_n)$. Damit ist $f(n)$ zeitkonstruierbar.

Aufgabe 21

Wir wissen: $f : \mathbb{N} \rightarrow \mathbb{N}$, $g : \mathbb{N} \rightarrow \mathbb{N}$ und f und g sind beide platzkonstruierbar.

\Rightarrow Es gibt 1-Band-Turingmaschinen F und G , so dass

$$\begin{aligned} \text{Space}_F(n_1) &\leq f(n_1) \\ \text{Space}_G(n_2) &\leq g(n_2) \end{aligned} \quad \forall n_1, n_2 \in \mathbb{N}$$

und für jede Eingabe 0^{n_1} generiert F das Wort $0^{f(n_1)}$ auf ihrem Arbeitsband und hält in Zustand q_{accept} .
 0^{n_2} generiert G das Wort $0^{g(n_2)}$

Wir konstruieren eine 5-Band-Turingmaschine H . H bekommt als Inpt 0^n auf sein Eingabeband. H kopiert die Eingabe auf *Band 2* und auf *Band 4* und simuliert F , dann G . Dabei sind *Band 2, 3* das Eingabe- und Arbeitsband von F und *Band 4, 5* Eingabe- und Arbeitsband von G .

Auf *Band 3* steht nun $0^{f(n)}$ und auf *Band 5* steht $0^{g(n)}$. H geht nach an den Anfang von *Band 3* und geht für jede gelesene 0 eins nach rechts und hängt den gesamten Inhalt von *Band 5* an *Band 1* an. Hat H alle 0en auf *Band 3* gelesen steht auf *Band 1* nun $0^{f(n) \cdot g(n)}$. H akzeptiert.

Die längste Konfiguration über alle Bänder und Schritte hat H am Ende, wenn das Ergebnis steht. Damit ist

$$\text{Space}_H = f(n) \cdot g(n) =: h(n) \quad \forall n \in \mathbb{N} \quad (\text{nach Def. 6.2})$$

und H hält immer in q_{accept} .

Nach *Lemma 6.1* gibt es eine äquivalente 1-Band-Turingmaschine H' mit $\text{Space}_{H'} \leq \text{Space}_H \leq h(n)$. Folglich ist $h(n)$ platzkonstruierbar.