Lehrstuhl für Informatik 1
Prof. Dr. Berthold Vöcking
Alexander Skopalik

WS 2007/08
29.10.2007

Übung zur Vorlesung
Berechenbarkeit und Komplexität

Lösung Blatt 2

Aufgabe 2.1: **(2+4+4)**

(a) **Aufgabe:** Für eine Eingabe $w\#^k$ verschiebe w um k Positionen nach rechts.
Idee: Wir verschieben sukzessive immer das letzte, bislang nicht verschobene Symbol von w nach rechts.

1) Markiere die Position links der Eingabe $w\#^k$ mit einem Symbol $\$ \notin \Sigma$. (Zeitbedarf: $O(1)$)
2) Laufe nach rechts bis zum letzten Symbol von w. ($O(|w|)$)
3) Speichere das aktuelle Symbol im Zustand und überschreibe es durch ein $\#$. ($O(1)$)
4) Laufe bis zum letzten Symbol von $\#^k$ nach rechts (dahinter folgt entweder ein Symbol aus $\{0,1\}$ oder ein Blank). ($O(k)$)
5) Überschreibe die aktuelle Bandposition mit dem im Zustand gespeicherten Symbol. ($O(1)$)
6) Laufe über die $\#$ nach links bis zum ersten Nicht-$\#$-Symbol. ($O(k)$)
7) Falls dies das $\$$ ist, so überschreibe $\$$ und die rechts davon stehenden $\#$ bis zum nächsten Element aus $\{0,1\}$ durch Blanks und halte. ($O(k)$)
8) Falls dies ein Symbol aus $\{0,1\}$ ist, gehe zu Schritt 3.

Bei jedem Durchlauf der Schritte 3-8 wird ein Symbol von w verschoben, die Schleife wird also $O(|w|)$-Mal ausgeführt. Damit ergibt sich insgesamt ein Zeitbedarf von

$$O(|w|) + O(|w|) \cdot O(k) = O(k \cdot |w|).$$

(b) **Aufgabe:** Bestimme für eine Eingabe 1^n die binäre Kodierung von n.
Idee: Verwende zwei Spuren und erhöhe für jede 1 auf der ersten Spur (Eingabe) einen Binärzähler auf der zweiten Spur um 1. Den Zähler verschieben wir jeweils so, dass er rechbündig zur aktuell betrachteten Position auf der ersten Spur ist (ansonsten würde die TM „zu viel“ Zeit brauchen, um auf der zweiten Spur immer wieder zum Zähler zurückzukehren).

1) Durchlaufe die Eingabe von links nach rechts und überschreibe dabei jede 1 durch $\binom{1}{B}$. Kehre wieder zur Ausgangsposition zurück. ($O(n)$)
2) Falls das aktuelle Symbol B ist (die Eingabe ist leer), so schreibe 0 auf das Band und stoppe. ($O(1)$)
3) Falls das aktuelle Symbol $\binom{1}{x}$ ist, für $x \in \{0,1,B\}$, so erhöhe den Zähler auf der zweiten Spur um 1. Durchlaufe dazu den Zähler auf der zweiten Spur von rechts nach links. Wenn dabei eine 1 (auf der zweiten Spur) gelesen wird, so ersetze sie durch eine 0 und gehe weiter nach links. Wenn eine 0 oder ein B (auf der zweiten Spur) gelesen wird, so ersetze sie durch eine 1 und gehe zurück zur rechtesten Position des Zählers. (Da der Binärzähler höchstens Länge $O(\log n)$ besitzen kann, ist der Zeitbedarf in $O(\log n)$.)
4) Verschiebe den Zähler auf der zweiten Spur um eine Position nach rechts und kehre zur aktuellen Position zurück. (Entsprechend 2.1 (a): $O(\log n)$)
5) Gehe eine Position nach rechts und lese auf der ersten Spur das nächste Symbol der Eingabe. Falls dies eine 1 ist, gehe zu Schritt 3.

6) Falls dies ein B ist, so schreibe den Inhalt der zweiten Spur auf das Band und überschreibe alle übrigen Symbole durch B. ($O(n)$)

Da in Schritt 5 nur n-Mal eine 1 gelesen wird, ergibt sich insgesamt ein Zeitbedarf von

$$O(n) \cdot O(\log n) + O(n) = O(n \log n).$$

(c) **Aufgabe:** Akzeptiere $L = \{0^n 1^n \mid n \in \mathbb{N}\}$ in Zeit $O(m \log m)$.
Idee 1: Teile iterativ die Anzahl der Einsen und Nullen durch zwei und überprüfe jeweils, ob der Rest übereinstimmt.

Der Wert m bezeichnet die Länge der Eingabe.

1) Teste in einem Durchlauf, ob die Eingabe die Form 0^*1^* besitzt. ($O(m)$)
2) Markiere die Positionen rechts und links von w mit \$. ($O(m)$)
3) Durchlaufe die Eingabe von links nach rechts und ersetze jede zweite Eins durch #. Zähle dabei modulo zwei. ($O(m)$)
4) Überschreibe anschließend jede zweite Null durch #. Zähle dabei modulo zwei. Wähle die Zustände so, dass das Ergebnis von Schritt 3 nicht verloren geht. ($O(m)$)
5) Vergleich die beiden Ergebnisse. Bei Gleichheit gehe zu Schritt 3, andernfalls verwirf die Eingabe.

Da die Schleife höchstens $O(\log(m))$ mal durchlaufen wird, ergibt sich ein Zeitbedarf von

$$O(m) + O(m \log m) = O(m \log m).$$

Idee 2: Zähle die Anzahl der Einsen und die Anzahl der Nullen und vergleiche die Zähler. Wir verwenden hier drei Spuren: Auf der ersten steht die Eingabe, auf der zweiten der Zähler für die Nullen und auf der dritten der Zähler für die Einsen.

Der Wert m bezeichnet die Länge der Eingabe.

1) Teste in einem Durchlauf, ob die Eingabe die Form 0^*1^* besitzt. ($O(m)$)
2) Erzeuge in einem Durchlauf drei Spuren, ersetze also jedes Symbol $x \in \{0, 1\}$ der Eingabe durch $\begin{pmatrix} x \\ B \\ B \end{pmatrix}$. ($O(m)$)
3) Durchlaufe die Eingabe von links nach rechts bis zur ersten Eins und zähle dabei binär die Anzahl der Nullen auf der zweiten Spur. (Entsprechend 2.1 (b): $O(m \log m)$)
4) Durchlaufe die Eingabe weiter von links nach rechts und zähle dabei binär die Anzahl der Einsen auf der dritten Spur. (Entsprechend 2.1 (b): $O(m \log m)$)
5) Der Zähler auf der zweiten Spur endet auf der Position der letzten Null auf der ersten Spur. Der Zähler auf der dritten Spur endet auf der Position der letzten Eins auf der ersten Spur. Die Zähler sind also maximal m Positionen voneinander entfernt. Verschiebe den Zähler auf der zweiten Spur soweit nach rechts, dass nun beide Zähler rechtsbündig übereinander stehen. (Entsprechend 2.1 (a): $O(m \log m)$)
6) Vergleiche die beiden Zähler und akzeptiere, wenn sie gleich sind, ansonsten verwirf die Eingabe. ($O(\log m)$)

Insgesamt ergibt sich also ein Zeitbedarf von

$$O(m) + O(m \log m) + O(\log m) = O(m \log m).$$

Alternativ könnte man in Schritt 4 auch den Zähler auf der zweiten Spur sukzessive herunterzählen. Man würde somit mit 2 Spuren auskommen und das Verschieben des Zählers in Schritt 5 wäre überflüssig. In Schritt 6 bliebe dann noch zu testen, ob der Zähler den Wert 0 besitzt. (Sollte der Zähler zwischenzeitlich schon negativ werden, kann auch schon in Schritt 4 abgebrochen werden.)

Aufgabe 2.2: **(5+5)**

(a) Es wird jeweils das am linken Rand der Eingabe mit dem am rechten Rand der Eingabe stehenden Symbol verglichen.

Wenn nur ein Band zur Verfügung steht, muss die Turingmaschine für diesen Vergleich einmal über die komplette Eingabe laufen.

1. Falls Eingabe leer, terminiere mit JA.
2. Speichere das aktuell gelesene Symbol im Zustand, und ersetze es durch ein B.
3. Laufe zum rechten Ende der Eingabe, und vergleiche das dort gespeicherte mit dem im Zustand gespeicherten Symbol. Falls ungleich, terminiere mit NEIN. Ansonsten lösche es auf dem Band.
4. Laufe ganz nach links, und gehe zu Schritt 1.

Daraus folgt eine Laufzeit von $\theta(n^2)$ für obigen Algorithmus. Gleiches gilt für die Optimierung, in der beim Zurücklaufen in Schritt 4 gleichzeitig auch ein Symbol vom rechten Rand im Zustand mitgenommen wird, um so bei jedem Lauf über die Eingabe zwei Zeichen miteinander vergleichen zu können.

(b) Die Idee besteht darin, die Eingabe zunächst in umgekehrter Reihenfolge auf das zweite Band zu kopieren. Dafür sind $\theta(n)$ viele Schritte nötig. In einer zweiten Phase werden beide Zeichenketten miteinander verglichen. Durch die Verwendung von 2 Bändern (mit 2 Schreib-/Lese-Köpfen) ist Phase 2 ebenfalls in Zeit $\theta(n)$ möglich.

1. Lese Zeichen auf Band 1, schreibe es im selben Schritt auf Band 2. Bewege Kopf 1 nach links, Kopf 2 nach rechts.
2. Falls rechtes Ende von w noch nicht erreicht, gehe zu Schritt 1.
3. Bewege Kopf 1 wieder an das linke Ende der Eingabe.
4. Lese Band 1 und Band 2 gleichzeitig, und vergleiche beide Zeichen miteinander. Falls ungleich, terminiere mit NEIN. Bewege beide Köpfe nach rechts.
5. Falls rechtes Ende von w noch nicht erreicht, gehe zu Schritt 4. Ansonsten terminiere mit JA.

Da diese TM auch Palindrome ungerader Länge akzeptiert, sollten vorher Eingaben, deren Länge ungerade ist, abgefangen werden.

Aufgabe 2.3: **(5)**

Wir simulieren eine 1-Band-TM mit beidseitig unendlichem Band auf einer 1-Band-TM mit einseitig unendlichem Band. Dazu benutzen wir eine zweite Spur. Zugriffe auf Bandpositionen $p \leq 0$ werden auf Position $1-p$ umgelenkt. An Position 0 steht ein Sonderzeichen, das sonst nicht verwendet wird, z.B. ein \$. Die Beschreibung der Zustandsübergangsfunktion wird dabei doppelt so groß, sie enthält nämlich einmal das ursprüngliche und unveränderte δ, und einmal ein gespiegeltes δ'.

Falls der Kopf auf einer Position $p > 0$ steht, wird Spur 2 ignoriert. Falls Position 0 erreicht ist, was am \$ zu erkennen ist, findet ein Übergang statt, und es wird zum gespiegelten δ' gewechselt. δ' benutzt Spur 2, und ignoriert Spur 1. Wo δ einen Schritt nach links macht, geht δ' nach rechts, und umgekehrt. Ansonsten bleibt die Berechnung wie bei der TM mit beidseitig unendlichem Band.

Diese Simulation verändert die Laufzeit nicht.