Übung zur Vorlesung

Theoretische Informatik II

Prof. Christoph Kreitz, Dipl. Math. Eva Richter

Universität Potsdam, Theoretische Informatik — Wintersemester 2003/04

Blatt 8 — Abgabetermin: 02.02.2004

Dieses Übungsblatt ist soll dazu dienen, verschiedene Themen aus der Vorlesung miteinander zu kombinieren und sicherer in der Anwendung verschiedener Beweistechniken zu werden. Da der Arbeitsaufwand etwas größer ist als bei den übrigen Blättern, sollten Sie mehr Zeit für die Bearbeitung einplanen. Für ein vollständig richtig bearbeitetes Blatt kann man maximal 10 Punkte erhalten.

Ausgangspunkt unserer Überlegungen ist der folgende sogenannte *Right-Left-Binary-Algorithmus* zur Berechnung von Potenzen natürlicher Zahlen[1]. Für ein Paar von natürliche Zahlen m und n soll der Wert von m^n berechnet werden.

Die naive Methode würde $n-1$ Multiplikationen erfordern. Das läßt sich jedoch verbessern, wenn man sich die folgende Idee zunutze macht. Sei

$$n = \sum_i \epsilon_i \cdot 2^i$$

die Darstellung von n zur Basis 2, wobei $\epsilon_i \in \{0,1\}$ für alle i, dann gilt:

$$m^n = \prod_{\epsilon_i=1} (m^{2^i}).$$

Benutzt man eine Hilfsvariable für die Werte m^{2^i}, die durch fortgesetztes quadrieren berechnet werden, so ergibt sich daraus der folgende Algorithmus:

Algorithmus Right-Left-Binary

1. Initialisieren
   Setze $y \leftarrow 1$.
   Falls $n = 0$ gebe y aus und halte an, andernfalls setze $N \leftarrow n$ und $z \leftarrow m$.

2. Multiplizieren?
   Falls N ungerade ist, setze $y \leftarrow z \cdot y$.

3. Halbieren von N
   Setze $N \leftarrow \lfloor N/2 \rfloor$. Falls $N = 0$ gib y als Ergebnis aus und halte an. Im anderen Fall setze $z \leftarrow z \cdot z$ und fahre fort mit 2.

Aufgabe 8.1 (2 Punkte)

Zeigen Sie die Korrektheit des Algorithmus.

Aufgabe 8.2 (4 Punkte)

Konstruieren Sie für $n = 5$ eine Einband-Turingmaschine, die den Algorithmus ausführt, d.h. die bei Eingabe von m und n den Wert m^n zurückliefert. Dabei dürfen zwei Turingmaschinen

[1] entnommen aus H. Cohen, A Course in Algebraic Number Theory, Springer 1993.

τ', und τ'' die Multiplikation bzw. Quadrieren ausführen benutzt als Unterprogramme[2] benutzt werden.

Hinweis: Da eine Turingmaschine keinen Mechanismus besitzt, um eine „Rücksprungsadresse“zu speichern- also einen Zustand in den nach Beendigung gewechselt wird, falls unser Entwurf einer TM vorsieht, daß ein Unterprogramm aus mehreren Zuständen aufgerufen wird-, können wir das Unterprogramm kopieren und für jede Kopie eine neue Menge von Zuständen verwenden.

Aufgabe 8.3 (4 Punkte)

Geben Sie eine Abschätzung der Komplexität in Abhängigkeit von der Größe von n und m an. Es dürfen dabei aus der Vorlesung bekannte Ergebnisse benutzt werden.

[2]Ein Unterprogramm einer Turingmaschine besteht aus einer Menge von Zuständen, die einen nützlichen Prozess ausführen. diese Menge von Zuständen umfasst einen Startzustand, sowie einen weiteren Zustand der temporär keine Bewegung ausführt, und als Rückgabezustand dient. Der Aufruf eines Unterprogramms findet statt, wenn ein Zustandsübergang zu seinem Startzustand erfolgt. siehe: Hopcroft/Motwani/Ullman „Einführung in die Automatentheorie... “Abschnitt 8.3.3