

# Theorie der Informatik

G. Röger  
Frühjahrssemester 2019

Universität Basel  
Fachbereich Informatik

## Übungsblatt 7 — Lösungen

**Aufgabe 7.1** (Komposition berechenbarer Funktionen ist berechenbar; 2 Punkte)

Seien  $f : \Sigma^* \rightarrow \Sigma^*$  und  $g : \Sigma^* \rightarrow \Sigma^*$  Turing-berechenbare partielle Funktionen für ein Alphabet  $\Sigma$ . Zeigen Sie, dass dann auch die *Komposition*  $(f \circ g) : \Sigma^* \rightarrow \Sigma^*$  Turing-berechenbar ist.

Die Komposition zweier Funktionen ist allgemein definiert als  $(f \circ g)(x) = f(g(x))$ . Der Funktionswert ist insbesondere auch undefiniert, wenn  $g(x)$  undefiniert ist.

**Lösung:**

*Wesentliche Ideen:*

Wenn  $f$  und  $g$  Turing-berechenbar sind, dann gibt es DTMs  $M_f$  und  $M_g$ , die auf Eingabe  $y$  das Wort  $f(y)$  bzw. auf Eingabe  $x$  das Wort  $g(x)$  berechnen.

Wenn wir die DTM  $M_g$  auf einer Eingabe  $x$  starten, für die  $g(x)$  definiert ist, dann sind nach Definition der von einer DTM berechneten Funktion Bandinhalt und Kopfposition beim Terminieren in genau der Form, in der sie auch für die Eingabe von  $M_f$  benötigt werden. Daher reicht es in diesem Fall aus,  $M_f$  und  $M_g$  zu einer gemeinsamen DTM zu kombinieren, indem alle Übergänge auf Endzustände von  $M_g$  stattdessen auf den Startzustand von  $M_f$  führen. Damit berechnet die Maschine zunächst  $g(x)$  auf der Eingabe  $x$ . Dies ist dann die Eingabe  $y$  für  $f$ , so dass insgesamt  $f(y) = f(g(x)) = (f \circ g)(x)$  berechnet wird.

*Technische Details:*

Wir gehen ohne Einschränkung davon aus, dass die Zustandsmengen von  $M_f$  und  $M_g$  disjunkt sind (sonst Zustände umbenennen) und ihre Bandalphabete identisch sind (sonst durch die Vereinigung der beiden Bandalphabete ersetzen; die dadurch neu zu definierenden Transitionen, z.B. wenn  $M_g$  ein Zeichen liest, das nur im Bandalphabet von  $M_f$  vorkommt, können beliebig gewählt werden, da sie nie benötigt werden).

Ein technisches Problem entsteht hierbei allerdings, wenn  $M_g$  in einer Konfiguration terminiert, die keine gültige Berechnung repräsentiert (weil der Kopf in der falschen Position steht oder unerlaubte Zeichen auf dem Band stehen). In diesem Fall sollte die kombinierte Maschine kein gültiges Ergebnis liefern, da  $(f \circ g)(x)$  undefiniert sein soll, aber ohne weitere Forderungen an  $M_f$  können wir das nicht garantieren.

Um dieses Problem zu beheben, ist es am einfachsten, die Maschine  $M_g$  so zu modifizieren, dass sie nie in einer ungültigen Konfiguration anhält, sondern stattdessen in eine Endlosschleife geht. Dies ist zum Beispiel zu erreichen, indem man am Ende der Berechnung von  $M_g$  den Bandinhalt auf korrekte Form prüft. Eine Schwierigkeit ist hierbei, festzustellen, wo der besuchte Teil des Bandes beginnt und endet. Dies kann man am einfachsten lösen, indem man während der Berechnung von  $M_g$  nie das Zeichen  $\square$  sondern stattdessen ein neues Zeichen  $\hat{\square}$  schreibt, das sich ansonsten wie  $\square$  verhält. Dann ist sicher gestellt, dass bei der Prüfung des Ergebnisses das Zeichen  $\square$  auf beiden Seiten das Ende des besuchten Bandes markiert. Bei der abschliessenden Prüfung kann dann  $\hat{\square}$  wieder durch  $\square$  ersetzt werden.

**Aufgabe 7.2** (Aufzählungsfunktionen; 1.5+1.5 Punkte)

Sei  $\Sigma = \{a, b, c\}$ . Geben Sie totale und berechenbare Funktionen  $f : \mathbb{N}_0 \rightarrow \Sigma^*$  an, die die folgenden Sprachen rekursiv aufzählen.

(a)  $L_1 = \{a^x b^y \mid x, y \in \mathbb{N}_0\}$

(b)  $L_2 = L_A \cup L_B$  wobei  $L_A$  und  $L_B$  Sprachen über  $\Sigma$  sind, die von den Funktionen  $f_A$  und  $f_B$  rekursiv aufgezählt werden.

**Lösung:**

$$(a) f_{L_1}(n) = \mathbf{a}^{\text{decode}_1(n)} \mathbf{b}^{\text{decode}_2(n)}$$

$$(b) f_{L_2}(n) = \begin{cases} f_A(\frac{n}{2}) & \text{falls } n \text{ gerade} \\ f_B(\frac{n-1}{2}) & \text{sonst} \end{cases}$$

**Aufgabe 7.3** (Entscheidbarkeit und Semi-Entscheidbarkeit; 0.5+0.5+1+1+1 Punkte)

Welche der folgenden Aussagen sind richtig, welche falsch? Geben Sie einen kurzen Beweis (1 Satz) oder ein Gegenbeispiel an. Sie dürfen alle Ergebnisse der Vorlesung verwenden.

- (a) Jede entscheidbare Sprache ist vom Typ 0.

**Lösung:**

Richtig. Aus der Vorlesung wissen wir: wenn  $A$  entscheidbar ist, sind  $A$  und  $\bar{A}$  semi-entscheidbar und damit vom Typ 0.

- (b) Wenn  $A$  entscheidbar ist, ist auch  $\bar{A}$  entscheidbar.

**Lösung:**

Richtig. Für die charakteristische Funktion von  $\bar{A}$  gilt:  $\chi_{\bar{A}}(w) = 1 - \chi_A(w)$ . Dies ist berechenbar, wenn  $\chi_A$  berechenbar ist.

- (c) Jede Sprache, die von einer Turingmaschine akzeptiert wird, ist entscheidbar.

**Lösung:**

Falsch. Wenn eine Sprache von einer Turingmaschine akzeptiert wird, ist sie semi-entscheidbar, aber nicht notwendigerweise auch entscheidbar. Wir haben in der Vorlesung mehrere Beispiele für Sprachen kennengelernt, die semi-entscheidbar aber nicht entscheidbar sind, z.B. das spezielle Halteproblem  $K$ .

- (d) Jede Sprache, die durch einen regulären Ausdruck beschrieben werden kann, ist entscheidbar.

**Lösung:**

Richtig. Zu jeder regulären Sprache gibt es einen DFA, der genau diese Sprache akzeptiert. Es ist einfach, zu einem gegebenen DFA ein Programm anzugeben, das genau das Verhalten des DFA auf einer gegebenen Eingabe simuliert und am Ende 1 oder 0 zurückliefert, je nach dem ob der Automat in einem Endzustand ist oder nicht.

- (e) Jede entscheidbare Sprache ist kontextfrei.

**Lösung:**

Falsch. Die Sprache  $\{\mathbf{a}^n \mathbf{b}^n \mathbf{c}^n \mid n \in \mathbb{N}\}$  ist zum Beispiel sicherlich entscheidbar (die Turingmaschine aus dem Beispiel in C7 war deterministisch und kann dazu leicht angepasst werden), aber wir wissen aus der Vorlesung (C6), dass sie nicht kontextfrei ist.

*Ergänzung zur Lösung:*

Es gelten die folgenden allgemeinen Beziehungen:

- Jede Typ-3-Sprache (reguläre Sprache) ist eine Typ-2-Sprache (kontextfreie Sprache), aber die Umkehrung gilt im Allgemeinen nicht.

- Jede Typ-2-Sprache (kontextfreie Sprache) ist eine Typ-1-Sprache (kontextsensitive Sprache), aber die Umkehrung gilt im Allgemeinen nicht.
- Jede Typ-1-Sprache ist entscheidbar, aber nicht jede entscheidbare Sprache ist vom Typ 1. (Im Gegensatz zu den anderen Aussagen wurde diese Aussage nicht in der Vorlesung diskutiert.)
- Jede entscheidbare Sprache ist semi-entscheidbar (eine Typ-0-Sprache), aber die Umkehrung gilt im Allgemeinen nicht.