

Theoretische Informatik - Übungsblatt 3
Übungsgruppe 7 - Mi, 16-18 Uhr
Elena Kvassova

Jan Hendrik Dithmar (Matrikelnummer 2031259)
Dirk Heine (Matrikelnummer 2030941)

Aufgabe 1

- (a) Um zu entscheiden, ob der Zustand z erreicht wird, ist es erforderlich, die Ausführung zu simulieren. Wird Zustand z irgendwann erreicht, ist das Problem sofort entschieden. Wird ein (akzeptierender oder nicht-akzeptierender) Endzustand erreicht, ohne dass vorher der Zustand z erreicht wurde, ist das Problem ebenfalls sofort entschieden. Um zu entscheiden, ob die Turingmaschine M einen Endzustand erreicht, müsste das Halteproblem gelöst werden. Es ist bewiesen, dass dieses Problem nicht entscheidbar ist, somit ist auch das Problem aus der Aufgabenstellung nicht entscheidbar.
- (b) Dieses Problem ist ebenfalls nicht entscheidbar mit Begründung wie in Aufgabenteil (a). Jedoch gibt es keinen speziellen designierten Zustand z , sondern wir betrachten die Menge der Zustände, die überhaupt das Zeichen a schreiben können. Erreichen wir einen solchen Zustand und schreiben in Abhängigkeit des Bandinhalts an der Lesekopfposition tatsächlich das Zeichen a , dann ist das Problem sofort positiv entschieden. Der Negativfall kann wieder nur entschieden werden, wenn die Maschine auf jeder Eingabe stoppt. Da das Halteproblem nicht entscheidbar ist, ist auch dieses Problem nicht entscheidbar.
- (c) Es ist ein entscheidbares Problem, da die Menge der möglichen Konfigurationen *bevor* ein von B verschiedenes Zeichen geschrieben wird endlich ist (die Zustandsmenge ist nach Definition endlich und vor / nach dem Lesekopf stehen immer nur unendlich viele B). Für jede dieser Konfigurationen aus dieser endlichen Menge von Konfigurationen kann in endlicher Zeit überprüft werden, ob durch sie eine Konfiguration *außerhalb* dieser Menge erreicht wird (also ob ein von B verschiedenes Zeichen geschrieben wird). Tritt dieser Fall ein, ist das Problem sofort positiv entschieden, anderenfalls stoppen wir nach endlich vielen Schritten (wenn wir die endliche Konfigurationsmenge durchlaufen haben) und das Problem ist negativ entschieden.
- (d) Um dieses Problem zu entscheiden, müsste die Turingmaschine mit jeder Eingabe aus Σ^* ausgeführt werden (unendlich viele Eingaben), für die jeweils getestet werden muss, ob die Turingmaschine die Eingabe akzeptiert. Da wir nichts weiteres über die Sprache wissen, kann diese auch semi-entscheidbar sein, so dass wir für alle Elemente aus Σ^* , die nicht der Sprache angehören, potentiell eine unendliche Laufzeit haben. Somit ist das Problem nicht entscheidbar.

Aufgabe 2

Das Problem ist wegen dem Satz von Rice nicht entscheidbar.

Aufgabe 3

Um zu prüfen, ob ein Wort in der Sprache $L_1 \setminus L_2 = L_1 \cap \overline{L_2}$ enthalten ist, müssen wir prüfen ob das Wort in L_1 enthalten ist und ob das Wort *nicht* in L_2 enthalten ist. Letzteres Prüfung lässt sich nicht durchführen, da L_2 nur rekursiv aufzählbar ist und somit $\overline{L_2}$ nicht rekursiv aufzählbar ist. Siehe Korollar 2.7.5 im Wegener:

- (1) gilt nicht, da L_2 nach Aufgabenstellung nicht rekursiv ist
- (2) gilt nicht, da L_2 nach Aufgabenstellung rekursiv aufzählbar ist
- (3) muss gelten, da nach Korollar eine der genannten drei Eigenschaften gelten muss; somit ist genau eine der beiden Sprachen L_2 und $\overline{L_2}$ rekursiv aufzählbar (in diesem Fall L_2) und die andere nicht rekursiv aufzählbar (in diesem Fall $\overline{L_2}$)

Aufgabe 4

- (a) Wir gehen zunächst davon aus, dass wir eine Sprache $L \subseteq \Sigma^*$ haben, die rekursiv ist. Das heißt, wir haben eine Turingmaschine M' , welche auf allen Eingaben stoppt und die Eingabe w genau dann akzeptiert, wenn $w \in L$. Wir können trivialerweise die gewünschte Turingmaschine M bauen, indem wir alle möglichen Wörter w in kanonischer Reihenfolge produzieren und sofern M' das jeweilige Wort w akzeptiert (die Turingmaschine M' stoppt nach Voraussetzung für jede Eingabe), dieses auf das Ausgabeband getrennt durch das Sonderzeichen schreiben. $\square$

Wir gehen jetzt davon aus, dass wir die Turingmaschine M haben und somit ein (potentiell unendlich langes) Band, auf dem alle Wörter w stehen, die in der Sprache L enthalten sind. Wir bauen die Turingmaschine M' , indem wir auf dem Band mit der Ausgabe von M nach der Eingabe von M' suchen. Die Suche terminiert in jedem Fall, da eine konkrete Eingabe immer endlich ist und wir aufgrund der kanonischen Reihenfolge nur so lange zu suchen brauchen, bis wir die Eingabe gefunden haben oder ein Wort, das länger als die Eingabe ist. $\square$

- (b) Wir gehen zunächst davon aus, dass wir eine Sprache $L \subseteq \Sigma^*$ haben, die rekursiv aufzählbar ist...

Wir gehen jetzt davon aus, dass wir die Turingmaschine M haben und somit ein (potentiell unendlich langes) Band, auf dem alle Wörter w stehen, die in der Sprache L enthalten sind, jedoch in beliebiger Reihenfolge. Die Turingmaschine M' lässt sich trivialerweise analog zu Aufgabenteil (a) bauen, jedoch stoppen wir nur, wenn wir das Ende des Ausgabebandes erreicht haben oder die Eingabe auf dem Ausgabeband gefunden haben. Dies ist für rekursive Aufzählbarkeit jedoch keine Einschränkung. $\square$