

Berechnungen und Logik

Hausaufgabenreihe 8

Henri Heyden, Nike Pulow

stu240825, stu239549

A1

- a) Falsch
- b) Wahr
- c) Falsch
- d) Falsch
- e) Falsch
- f) Wahr
- g) Wahr
- h) Falsch
- i) Falsch
- j) Falsch

A2

Vor.: L_1, L_2 Turing erkennbar, sodass $L_1 \cup L_2$ und $L_1 \cap L_2$ Turing entscheidbar sind. Sei $\alpha \in L_1$

Beh.: L_1 ist Turing entscheidbar.

Bew.: Sei $f : A^* \rightarrow A^*$ mit $f(\overline{L_1 \cup L_2}) = \overline{L_1 \cup L_2}$ und $f(L_1 \cup L_2) = \{\alpha\}$.

Dann existiert eine Turingmaschine, die sich wie folgt verhält, für $w \in A^*$ Eingabewort:

Ist $w \in \overline{L_1 \cup L_2}$, dann tue nichts und akzeptiere.¹

Sonst leere die Eingabe und schreibe α hin und akzeptiere.

Dann gilt für alle Eingaben $w \in A^*$. Berechne w , dann ist die akzeptierende Konfiguration $f(w)$. Dies gilt nach den eben beschriebenen Definitionen.

Des Weiteren gilt: $\forall w \in A^* : w \in L_1 \Leftrightarrow f(w) \in L_1 \cup L_2$ nach Definition f .

Somit gilt also $L_1 \leq (L_1 \cup L_2)$. Beobachte, dass $L_1 \cup L_2$ Turing entscheidbar ist. Dann gilt nach Satz „Eigenschaften der Reduktion 1“, dass L_1 Turing entscheidbar ist, – was zu zeigen war. \square

¹Sei hier zu beachten, dass $\overline{L_1 \cup L_2}$ entscheidbar ist.

A3

Beh.: Eine Turingmaschine mit einem beidseitig unendlichem Band entscheidet genau die gleichen Sprachen, wie eine einseitig unendliche Turingmaschine.

Bew.: Wir teilen den Beweis in zwei Richtungen auf:

1. „ \Rightarrow “

Es wurde in der Vorlesung bereits gezeigt, dass k-bändige Turingmaschine gleich mächtig sind, wie einbändige, deswegen zeigen wir, dass für jede beidseitige Turingmaschine eine beidbandige Turingmaschine existiert.

Betrachten wir eine beidseitige Turingmaschine.

Dann existiert folgende beidbandige Turingmaschine: Wir nennen Band 1 „rechtes Band“ und Band 2 „linkes Band“. Für jegliches beschriftetes Band, dass verwendet wird, werden das linke Band und das rechte Band so aufgeteilt, dass das linke Band die gespiegelte linke Hälfte des originalen Bandes ist und das rechte Band die rechte Hälfte des originalen Bandes ist. Die Mitte ist dabei der Start des rechten Bandes.

Transformiere nun beide Bänder nochmal, so, dass ein Buchstabe, der nicht im Bandalphabet rechts bzw. links vom linken bzw. rechtem Band angefügt wird. Dieser Buchstabe repräsentiert die Mitte des originalen Bandes.

Nun adjustieren wir die Transitionsfunktion so, dass folgendes gilt:

Starte auf dem zweiten Buchstaben des rechten Bandes¹.

Wir rechnen so lange genau so, wie in der ursprünglichen Turingmaschine, bis wir (falls überhaupt) die Markierung erreichen.

Dann rechne mit invertierten Bewegungsaktionen bei der zweiten Stelle des linken Bandes weiter (also nicht auf der Markierung, sonst würden wir ja wieder zurückspringen).

Wenn wieder die Markierung erreicht wird, fahre wieder im „originalen“ Modus rechts weiter auf der zweiten Stelle.

¹wir betrachten hier eine zweibandige Turingmaschine, wessen Bänder nicht synchronisiert sind, also wo wir nicht sozusagen Tupel betrachten, dass diese Turingmaschinen mit den synchronen k-bandigen äquivalent sind in der Stärke, ließe sich leicht zeigen, aber wir werden nicht uns nicht darauf fokussieren in diesem Beweis, um ihn nicht unnötig in die Länge zu ziehen, – wie sonst.

Wir ersetzen sozusagen die Linke unendliche Hälfte der ursprünglichen Turingmaschine mit einem einzelnen Band.

Eine andere Möglichkeit mit einer Turingmaschine mit einem Band das Problem zu lösen ist es eine zweiseitige Turingmaschine auf der einbandigen zu simulieren, wenn man annimmt, dass sich jede mathematische Struktur über eine Turingmaschine simulieren lässt.

2. „ \Leftarrow “

Betrachten wir eine einseitige Turingmaschine, dann existiert eine Turingmaschine mit zweiseitigem Band, welche genau die gleichen Transitionsfunktionen besitzt.

Es ist uns schließlich egal, wie viel „Platz“ links von dem Ursprung ist, da wir nie diesen überschreiten werden.

Somit sind beide Richtungen gezeigt. □

A4

a)

Um die Nicht-Gültigkeit dieser Reduktionsfunktion zu zeigen, bedienen wir uns einem der Sätze, der in der verlinkten Ressource in Aufgabe 3 genannt wird:

Ist L_2 Turing-Entscheidbar, und gilt $L_1 \leq L_2$, so ist auch L_1 Turing-Entscheidbar.

Wir definieren $L_2 = L$ und $L_1 = HALT_{TM}^\epsilon$.

L ist offensichtlich entscheidbar, eine TM, welche diese Sprache entscheidet, ließe sich im Handumdrehen konstruieren. Wir wissen aber aus der genannten Ressource auch, dass $HALT_{TM}^\epsilon$ nicht entscheidbar ist. Es kann also $HALT_{TM}^\epsilon \leq L$ nicht gelten.

b)

Die Reduktion hier ist unvollständig formuliert, da der Schritt 2. Prüfe ob $|w| \equiv_2 0$ abhängig von der Existenz und Funktionalität einer TM ist, welche hier aber nicht weiter definiert ist. Wir wissen also nicht, ob $\langle M_w \rangle$ tatsächlich

das berechnet, was zu berechnen ist.

c)

Die Reduktion funktioniert nicht, da angenommen ist, dass $M_{\langle M \rangle}$ wirklich TM ist.

Jedoch ist $M_{\langle M \rangle}$ abhängig von der Turingmaschine, die das Halteproblem entscheidet, welche nicht existiert.

Somit existiert also auch $M_{\langle M \rangle}$ nicht als TM und die Reduktionsfunktion f ist ungültig.

Man sieht dies auch an einem anderen Punkt unserer Definition: $M_{\langle M \rangle}$ müsse nach unserer Definition immer halten, tut es aber nicht.