

Grundlagen der Theoretischen Informatik, WS11/12

Übungsblatt 9, Abgabe bis zum **Mi. 21. Dezember**¹

- In jeder Aufgabe können 5 Punkte erreicht werden.
- Es sind Doppelabgaben erlaubt. Name und Matr.Nr. sollten gut leserlich auf der Abgabe stehen.
- Bitte geben Sie Ihre Lösungen in gut leserlicher und sauberer Form ab.
- Begründen Sie Ihre Antworten und argumentieren Sie *nachvollziehbar*.

Aufgabe 1. Mehrere Programmiersprachen enthalten auch Schleifen der Form

`repeat C until E`

wobei C eine Anweisung und E ein Ausdruck ist. Die Semantik hierbei ist, dass C solange ausgeführt wird, bis E erfüllt ist, mindestens jedoch einmal.

- (a) Geben Sie an, wie solche Konstrukte in der WHILE-Sprache mit deren beschränktem Sprachumfang simuliert werden könnten.
- (b) Wie müsste man die Übergangsrelation $\cdot \vdash \cdot \rightarrow \cdot$ (Definition 3.2.12) erweitern, wenn man `repeat ... until` Konstrukte zum Sprachumfang von WHILE hinzufügt?

Aufgabe 2. Wir wollen die Sprache WHILE nun noch um das Konstrukt

`for I = E1 to E2 do C ;`

erweitern.

¹ Abgabe am Besten in der Vorlesung. Alternativ können Lösungen auch *persönlich* bei Christian Uhrhan (EN-B 0125) oder im Sekretariat der theoretischen Informatik (EN-B 0121) abgegeben werden. Verwenden Sie auf keinen Fall den Briefkasten des Lehrstuhls.

- (a) Erweitern Sie die Semantik von Ausdrücken (Definition 3.2.10) um den Ausdruck

$$(!= E F)$$

welcher testet, ob E und F übereinstimmen.

- (b) Geben Sie an, wie der Ungleichstest in WHILE simuliert werden könnte.
- (c) Geben Sie an, wie das `for ... to ... do ...` Konstrukt in WHILE simuliert werden könnte.
- (d) Wie muss man die Übergangsrelation $\cdot \vdash \cdot \rightarrow \cdot$ erweitern, damit `for ... to ... do ...` Konstrukte zum Sprachumfang von WHILE gehören?
-

Aufgabe 3. Konstruieren Sie eine Turingmaschine, die Wörter aus $\{a, b\}^*$ erkennt, in denen die gleiche Anzahl von *as* wie *bs* enthalten sind.

Aufgabe 4. Sei $A = (\Sigma, Q, s, F, \delta)$ ein DEA. Konstruieren Sie eine Turingmaschine M , die genau dann hält, wenn das Eingabewort aus $L(M)$ ist. (Korrektheitsbeweis ist nicht notwendig).

ENDE