

**Einführung in die theoretische Informatik**  
Sommersemester 2019 – Hausaufgabenblatt 8

**Templates in anderen Sprachen**

Für die Korrektur Ihrer Programmierabgaben wird TUMjudge(<https://judge.in.tum.de/theo/public/>) verwendet. Auf dem ersten Hausaufgabenblatt wurden die für Sie relevanten Funktionen des Webinterfaces erklärt. Weitere Details finden Sie auf den Folien des Praktikums Algorithms for Programming Contests von 2017(<https://www7.in.tum.de/um/courses/theo/ss2018/materials/judge.pdf>).

**Programmierabgabe**

Aus Zeitgründen können wir diese Woche keine Python- und Haskell-Templates bereitstellen.

**AUFGABE 8.1.** (*AutomataTutor*)

Wenn Sie im Laufe des Semesters Aufgaben im AutomataTutor gelöst haben sollten Sie eine Mail mit einem persönlichen Link zu einer kurzen Umfrage über Automata Tutor bekommen haben. Bitte nehmen Sie daran teil – Ihr Feedback hilft uns, das Tool und damit die Lehre an der TUM nachhaltig zu verbessern. Die Daten sind natürlich anonym und werden gemäß Datenschutzrichtlinien gespeichert.

**AUFGABE 8.2.** (*TM für  $ww^R$* )

In dieser Aufgabe sollen Sie eine deterministische Turingmaschine entwerfen, die für ein Eingabewort  $w \in \{0, 1\}^*$  das Wort  $ww^R$  ausgibt.

1 Punkt

- Laden Sie das Codegerüst und die Angabe für die Hausaufgabe auf [https://www21.in.tum.de/teaching/theo/SS19/materials/ha/08/templates\\_ha08.tar.gz](https://www21.in.tum.de/teaching/theo/SS19/materials/ha/08/templates_ha08.tar.gz) herunter.
- Lesen Sie sich die PDF-Angabe zu der Aufgabe ‘ReverseTM’ durch und folgen Sie den Anweisungen dort.
- Laden Sie dann für Problem A (ReverseTM) ein Programm in einer der verfügbaren Sprachen hoch, das die Turingmaschine in geeignetem Format auf Stdout ausgibt.

**AUFGABE 8.3.** (*TM für eine Klammersprache*)

In dieser Aufgabe sollen Sie eine deterministische Turingmaschine entwerfen, die die Sprache aller korrekten Klammerungen über  $()$  und  $[]$  erkennt.

1 Punkte

- Laden Sie das Codegerüst und die Angabe für die Hausaufgabe auf [https://www21.in.tum.de/teaching/theo/SS19/materials/ha/08/templates\\_ha08.tar.gz](https://www21.in.tum.de/teaching/theo/SS19/materials/ha/08/templates_ha08.tar.gz) herunter.
- Lesen Sie sich die PDF-Angabe zu der Aufgabe ‘BracketTM’ durch und folgen Sie den Anweisungen dort.
- Laden Sie dann für Problem B (BracketTM) ein Programm in einer der verfügbaren Sprachen hoch, das die Turingmaschine in geeignetem Format auf Stdout ausgibt.

**AUFGABE 8.4.** (*WHILE  $\rightarrow$  TM: Teil A*)

In dieser Aufgabe sollen Sie ein Fragment der Übersetzung von WHILE-Programmen in Turingmaschinen implementieren.

1 Punkte

- Laden Sie das Codegerüst und die Angabe für die Hausaufgabe auf [https://www21.in.tum.de/teaching/theo/SS19/materials/ha/08/templates\\_ha08.tar.gz](https://www21.in.tum.de/teaching/theo/SS19/materials/ha/08/templates_ha08.tar.gz) herunter.
- Lesen Sie sich die PDF-Angabe zu der Aufgabe ‘WhileToTMEasy’ durch und folgen Sie den Anweisungen dort.
- Laden Sie dann für Problem C (WhileToTMEasy) alle benötigten Dateien hoch – falls Sie eines unserer Templates benutzen, müssen Sie *alle* Dateien des Templates hochladen, nicht nur die, die Sie verändert haben.

**AUFGABE 8.5.** (*WHILE  $\rightarrow$  TM: Teil B*)

In dieser Aufgabe sollen Sie die Übersetzung von allgemeinen WHILE-Programmen in Turingmaschinen implementieren.

1 Punkte

- Laden Sie das Codegerüst und die Angabe für die Hausaufgabe auf [https://www21.in.tum.de/teaching/theo/SS19/materials/ha/08/templates\\_ha08.tar.gz](https://www21.in.tum.de/teaching/theo/SS19/materials/ha/08/templates_ha08.tar.gz) herunter.
- Lesen Sie sich die PDF-Angabe zu der Aufgabe ‘WhileToTM’ durch und folgen Sie den Anweisungen dort.
- Laden Sie dann für Problem D (WhileToTM) alle benötigten Dateien hoch – falls Sie eines unserer Templates benutzen, müssen Sie *alle* Dateien des Templates hochladen, nicht nur die, die Sie verändert haben.