

- Unter ***www.i1.informatik.uni-bonn.de*** werden Unterlagen und Informationen zu Vorlesung und Übung bereitgestellt.
- Bearbeitung und Abgabe der Übungsblätter ist in festen Gruppen von bis zu 3 Personen erlaubt.
- Die Abgabe muss auf dem ersten Blatt in der **ersten Zeile** deutlich die **Namen** der Studierenden und die **Nummer** der Übungsgruppe enthalten. Eine Abgabe aus mehreren Blättern ist zu heften!
- Die Lösungen können bis **Mittwoch 09 Uhr** in den Postkasten (**Nicht in der Vorlesung!**) im AVZ III eingeworfen werden.

Aufgabe 16: Primitive Rekursion (4 Punkte)

Zeigen Sie ohne Verwendung von Turingmaschinen, dass die Funktion der Fibonacci Zahlen $f : \mathbb{N}_0 \rightarrow \mathbb{N}_0$ mit

$$\text{fib}(n) = \begin{cases} 1, & \text{falls } n \in \{0, 1\} \\ \text{fib}(n-1) + \text{fib}(n-2), & \text{sonst.} \end{cases}$$

primitiv rekursiv ist.

Hinweis: Die Funktion encode ist primitiv rekursiv. Sie dürfen annehmen, dass die zugehörigen Umkehrfunktionen ebenfalls primitiv rekursiv sind.

Aufgabe 17: Turingmaschine entwerfen (4 Punkte)

Betrachten Sie zwei Wörter $u, v \in \{0, 1\}^*$. Es soll eine 1-Band Turingmaschine konstruiert werden, welche gestartet mit der Eingabe $\$u\#v$ in einem Zustand q_t hält, falls $u = v$ gilt. Gehen Sie dazu wie folgt vor:

- Geben Sie Zustandsübergangsfunktion ihrer Turingmaschine als Diagramm (wie bei Automaten) an. Beschriften Sie Übergänge im Diagramm in der Form $0 : 1, L$. Die Bedeutung der Notation ist: Beim

Lesen von 0 an der Kopfposition wird eine 1 geschrieben und der Kopf nach links bewegt.

- Erläutern Sie kurz die Idee Ihrer Konstruktion.
- Geben Sie zu jedem Zustand eine (kurze, informelle) Beschreibung an.
- Notieren Sie die Folge der Konfigurationen (als Zeichenketten), die Ihre Turingmaschine bei Abarbeitung der Eingabe $u = v = 10$ durchläuft.

Aufgabe 18: Turingmaschinen simulieren (4 Punkte)

Seien A und B zwei 1-Band Turingmaschinen. Während B der gewohnten Definition aus dem Skript entspricht, gibt es bei A eine Besonderheit: A hat ein zu beiden Seiten unbeschränktes Band, nicht nur ein zur rechten Seite unbeschränktes. In diesem Fall markiert das $\$$ -Zeichen den Mittelpunkt des Bandes, und nicht das linke Ende.

Ihre Aufgabe ist nun, die beidseitig unbeschränkte Turingmaschine A mit Hilfe der normalen Turingmaschine B zu simulieren. Wir sagen, dass die Turingmaschine B die andere Turingmaschine A simuliert, wenn B die gleiche grundsätzliche Funktionsweise an internen Zuständen aufweist wie A (zusätzliche, aufgrund der Simulation erforderliche Zustände dürfen in B natürlich existieren) und insbesondere auch für die gleichen Dateneingaben akzeptiert bzw. nicht akzeptiert.

Geben Sie auf formal nachvollziehbare Weise an, wie B anhand A konstruiert wird und wie dafür die Eingabe auf dem Band von B deponiert wird. Eine Aufgabe, die mit A gelöst wird, soll mit B maximal doppelt soviel Zeit benötigen.

Aufgabe 19: WHILE-Programme (Präsenzaufgabe)

In den Übungen haben wir bereits WHILE-Programme kennengelernt. Eine (intuitiv) berechenbare Funktion f heißt *WHILE-berechenbar*, falls ein WHILE-Programm formuliert werden kann, welches f berechnet. Mit $\mathcal{F}_{\text{WHILE}}$ bezeichnen wir die Klasse/Menge der WHILE-berechenbaren Funktionen.

Zeigen Sie, dass jede μ -rekursive Funktion auch *WHILE-berechenbar* ist, d.h. es gilt $\mathcal{F}_\mu \subseteq \mathcal{F}_{\text{WHILE}}$.