



3. Übungsblatt zur Vorlesung Theoretische Informatik

Aufgabe 1: Quotienten, Head und Tail

1+0.5+0.5 Punkte

Seien Σ ein Alphabet; U, W, L Sprachen. Der Rechtsquotient von U und W ist

$$U/W := \{v \in \Sigma^* \mid \exists w \in W. vw \in U\}.$$

Die Menge der Präfixe von L ist

$$\text{Head}(L) := \{v \in \Sigma^* \mid \exists w \in \Sigma^*. vw \in L\}.$$

Die Menge der Suffixe von L ist

$$\text{Tail}(L) := \{w \in \Sigma^* \mid \exists v \in \Sigma^*. vw \in L\}.$$

Geben Sie an, welche Bedingungen die Sprachen U und W bzw. L erfüllen müssen, damit Folgendes gilt:

- $\varepsilon \in U/W$.
- $\varepsilon \in \text{Head}(L)$.
- $\varepsilon \in \text{Tail}(L)$.

Aufgabe 2: Reflexive transitive Hülle

0,5+0,5+3+2 Punkte

Sei A eine Menge. Die Komposition zweier Relationen $R, R' \subseteq A \times A$ ist definiert als

$$R \circ R' := \{(a, c) \in A \times A \mid \exists b \in A. (a, b) \in R \wedge (b, c) \in R'\}$$

Sei $R \subseteq A \times A$ eine Relation auf A . Die reflexiv-transitive Hülle von R , R^* , wurde bereits in der Vorlesung wie folgt definiert:

$$\begin{aligned} R^0 &:= \{(a, a) \mid a \in A\} \\ R^{n+1} &:= R \circ R^n \\ R^* &:= \bigcup_{n \in \mathbb{N}} R^n \end{aligned}$$

Zeigen Sie, dass diese Definition die charakteristischen Eigenschaften der reflexiv-transitiven Hülle erfüllt, d.h. dass gilt:

a) $R \subseteq R^*$

b) R^* ist reflexiv:

$$\forall a \in A. (a, a) \in R^*$$

c) R^* ist transitiv:

$$\forall a, b, c \in A. (a, b) \in R^* \wedge (b, c) \in R^* \implies (a, c) \in R^*$$

d) R^* ist die kleinste Relation mit diesen Eigenschaften:

$$\forall S. R \subseteq S \wedge S \text{ reflexiv} \wedge S \text{ transitiv} \implies R^* \subseteq S$$

Hinweis: Zeigen Sie falls nötig separat, dass gilt:

$$R^n \circ R^m = R^{n+m}$$

Sie dürfen voraussetzen, dass die Komposition \circ assoziativ ist.

Aufgabe 3: Programmieren mit Registermaschinen

1+1,5+2,5+3 Punkte

Konstruieren Sie folgende Registermaschinen-Programme:

a) **reset** sodass

$$f_{\text{reset}(1)}(x) = (0)$$

Intuitiv setzt die Maschine den Wert von Register 1 auf 0.

b) **sub** sodass

$$\begin{aligned} f_{\text{sub}(1,2)}(n, m) &= (n_{\text{out}}, m_{\text{out}}) \\ n_{\text{out}} &= \begin{cases} n - m & | n \geq m \\ 0 & | \text{sonst} \end{cases} \\ m_{\text{out}} &= \begin{cases} 0 & | n \geq m \\ m - n & | \text{sonst} \end{cases} \end{aligned}$$

Intuitiv berechnet die Maschine die Differenz der Zahlen in den Registern 1 und 2. Eventuell bleibt ein Rest im Register 1 stehen.

c) **copy** sodass

$$f_{\text{copy}(n,m)}(x_1, \dots, x_l) = (x_1, \dots, x_{m-1}, x_n, x_{m+1}, \dots, x_l) \quad \text{mit } l \geq \max(n, m)$$

Intuitiv überschreibt die Maschine den Wert des Registers m durch den Wert des Registers n .

d) `div` sodass

$$f_{\text{div}}(n, m) = (n_{\text{out}}, m_{\text{out}}, \lfloor n/m \rfloor)$$

Dabei ist $\lfloor k \rfloor$ die größte ganze Zahl x mit $x \leq k$. Intuitiv berechnet die Maschine die ganzzahlige Division der Werte in den Registern 1 und 2 und gibt das Ergebnis im Register 3 aus. n_{out} und m_{out} sind beliebig.

Ihre Programme müssen für alle möglichen Eingaben terminieren. (Register, die nicht durch den Eingabevektor gesetzt werden, haben im Anfangszustand den Wert 0.) Geben Sie die Programme jeweils als Tabelle an.

Hinweis: Zur Vereinfachung dürfen Sie folgende Konventionen verwenden:

- Ein Programm kann andere Programme aufrufen. Zum Beispiel setzt folgende Maschine die Register 1 und 2 jeweils auf 1, kopiert sie dann in die Register 3 und 4 und inkrementiert diese jeweils wieder um 1.

```
0  Inc(1)
1  Inc(2)
2  copy(1, 3)
3  copy(2, 4)
4  Inc(3)
5  Inc(4)
```

Beachten Sie, dass `copy` den Wert der geteilten Register 1 bis 4 sowohl lesen als auch schreiben kann.

- Jedes Programm kann beliebig viele temporäre Register t_i ($i \in \mathbb{N}$) verwenden. Sie dürfen annehmen, dass diese verschieden von allen anderen Registern (insbesondere den Registern anderer Programme, die Sie aufrufen oder von denen Ihr Programm aufgerufen wird) sind. Diese Register haben anfangs den Wert 0 und können im Endzustand beliebige Werte annehmen.
- Teilaufgabe d) hat eine kompakte Lösung unter Verwendung der in a) bis c) definierten Subprogramme. Sie dürfen dabei Varianten von a) und b) benutzen, die auf beliebigen Registern arbeiten.