Universität Leipzig
Institut für Informatik
Sommersemester 2025

Prof. Dr. Andreas Maletti,
Dr. habil. Karin Quaas,
Fabian Sauer

Aufgaben zur Lehrveranstaltung

Berechenbarkeit

Lösungen zu Serie 3

Übungsaufgabe 3.1 (Turingmaschinen Mächtigkeit)

Beweisen Sie die folgende Aussage.

Für jeden endlichen Automaten $A = (Q, \Sigma, q_0, \delta, F)$ existiert eine normierte Turingmachine M_A mit $L(M_A) = L(A)$.

In Ihrem Beweis geben Sie bitte für einen beliebigen Automaten A eine *direkte* Konstruktion von M_A an, das heisst, vermeiden Sie die Verwendung von bereits bekannten Zusammenhängen zwischen Automaten und Grammatiken.

LÖSUNG: Sei $A = (Q, \Sigma, q_0, \delta, F)$ ein endlicher Automat. Konstruiere $M_A = (Q', \Sigma, \Gamma, \Delta, \Box, q_0, q_+, q_-)$, wobei

- $Q' = Q \cup \{q_+, q_-, q_L\}$, wobei $q_+, q_-, q_L \notin Q$ nicht in Q vorkommen
- $\Gamma = \Sigma \cup \{\Box\}$,
- Δ enthält die folgenden Transitionen:
  - $(q, a) \to (q', a, \triangleright)$ für alle $(q, a, q') \in \delta$
  - $(q, \Box) \to (q_L, \Box, \triangleleft)$ für alle $q \in F$
  - $(q_L, \sigma) \to (q_L, \sigma, \triangleleft)$ für alle $\sigma \in \Sigma$, and $(q_L, \Box) \to (q_+, \Box, \triangleright)$ (Bewegung zum Bandanfang für Normiertheit)

M_A ist normiert, denn für alle $w \in \Sigma^*$, $u, v \in \Sigma^*$ mit $\varepsilon q_0 w \vdash^*_{M_A} u q_+ v$ gilt: $u \in \{\Box\}^*$ und $v \in \Gamma^*_{M_A} \cdot \{\Box\}^*$. Es gilt $L(A) = L(M_A)$ (ohne Beweis).

Zusatzfrage: Wie funktioniert der Beweis für die stärkere Aussage: M_A soll deterministisch sein? (Mehrere Möglichkeiten: (1) Für jede TM M gibt es deterministische TM M' mit $L(M') = L(M)$. (2) TM kann Potenzmengenkonstruktion für endliche Automaten direkt ausführen.)

Übungsaufgabe 3.2 (Turing-berechenbare Funktionen)

Sei Σ ein endliches Alphabet. Wir definieren die Präfixrelation $\leq_p$ über Σ^* durch $u \leq_p w$ falls $v \in \Sigma^*$ mit $w = u \cdot v$ existiert. Analog definieren wir die Suffixrelation $\leq_s$

über Σ^* durch $u \leq_p w$ falls $v \in \Sigma^*$ mit $w = v \cdot u$ existiert. Wir definieren die Funktion $f : \Sigma^* \to \mathcal{P}(\Sigma^+)$ durch

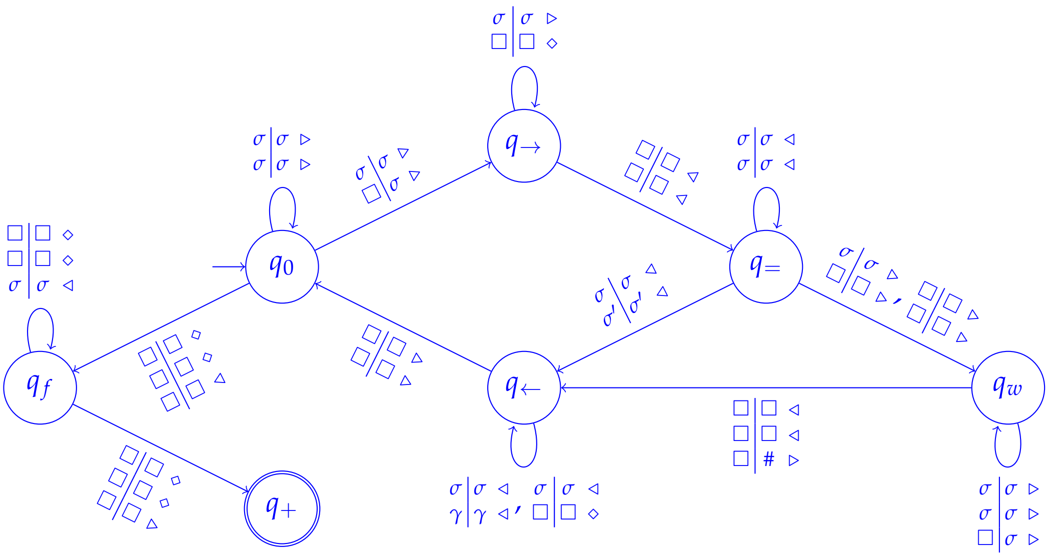
$$f(w) = \{u \in \Sigma^+ \mid u \leq_p w \text{ und } u \leq_s w\}$$

für alle $w \in \Sigma^*$. Beispielsweise ist $f(abaab) = \{ab, abaab\}$ und $f(aaaa) = \{a, aa, aaa, aaaa\}$.

Zeigen Sie, dass f Turing-berechenbar ist. Bitte geben Sie aussagekräftige Erläuterungen zur Verhaltensweise Ihrer Turingmaschine.

LÖSUNG: Wir definieren eine deterministische Turingmaschine M mit $T(M) = f$. Die Idee der Funktionsweise von M ist wie folgt: M besitzt 3 Bänder. Auf Band 1 befindet sich das Eingabewort w. Auf Band 2 speichert M nacheinander jeweils (Buchstabe für Buchstabe) alle möglichen Präfixe u von w und prüft, ob u auch Suffix von w ist. Falls ja, wird u auf Band 3 gespeichert (die einzelnen Wörter auf Band 3 trennen wir durch ein frisches Symbol #).

(In der folgenden Abbildung der Turingmaschine haben wir der Einfachheit halber an manchen Transitionen die Angaben für das 3. Band weggelassen.)



Übungsaufgabe 3.3 (LOOP-berechenbare Funktionen)

Zeigen Sie, dass die beidem im Folgenden definierten Funktionen LOOP-berechenbar sind.

(a) $g_1 \colon \mathbb{N} \to \mathbb{N}$ sei definiert durch

$$n \mapsto n!$$

für alle $n \in \mathbb{N}$.

(b) $g_2 \colon \mathbb{N}^2 \to \mathbb{N}$ sei definiert durch

$$g_2(n_1, n_2) = \begin{cases} 1 & \text{falls } n_1 = n_2 \\ 0 & \text{sonst} \end{cases}$$

für alle $n_1, n_2 \in \mathbb{N}$.

LÖSUNG: (a) Wir konstruieren ein LOOP-Program P_1 mit $|P_1| = g_1$.

```
IF (x1 = 0) {x1 = 1} ELSE {
   x2 = x1 − 1;
   x3 = x1 − 2;
   LOOP(x3){
    x1 = x1 · x2;
    x2 = x2 − 1
   }
}
```

Zuerst Behandlung des Sonderfalls $x_1 = 0$ (Ausgabe 1 denn $0! = 1$). Anderenfalls: In x_1 wird der Eingabewert gespeichert; auch die Ausgabe steht wieder in x_1. In x_2 werden die zu multiplizierenden Faktoren schrittweise verkleinert. x_3 legt fest, wie oft der LOOP ausgeführt wird, in unserem Fall 2-mal weniger als der Wert der Eingabe.

(b) Wir konstruieren ein LOOP-Program P_2 mit $|P_2| = g_2$.

```
x3 = x2 − x1;
x4 = x1 − x2;
x3 = x3 + x4:
x1 = 1;
LOOP(x3){x1 = 0}
```

Hausaufgabe 3.4 (Turingmaschinen Mächtigkeit) (6)

Sei $A = (Q, \Sigma, q_0, \delta, F)$ ein endlicher Automat[1] und sei $w \in \Sigma^*$. Wir nennen w einen *Zeugen für Mehrdeutigkeit von* A falls es mindestens zwei unterschiedliche akzeptierende Läufe von A auf w gibt.

Beweisen Sie die folgende Aussage.

Für jeden endlichen Automaten $A = (Q, \Sigma, q_0, \delta, F)$ *existiert eine Turingmachine* M_A *mit* $L(M_A) = \{w \in \Sigma^* \mid w \text{ ist Zeuge für Mehrdeutigkeit von } A\}$.

[1] $\delta \subseteq Q \times \Sigma \times Q$, d.h., es sind keine ε-Transitionen erlaubt.

LÖSUNG: Sei $A = (Q, \Sigma, q_0, \delta, F)$ ein endlicher Automat. Definiere die Turingmaschine $M_A = (Q', \Sigma, \Gamma, \Box, \Delta', q_0, q_+, q_-)$, wobei

- $Q' = \{q_+, q_-\} \cup Q \cup (Q \times Q)$, wobei $q_+, q_- \notin Q$;
- Δ enthält genau die folgenden Transitionen:
  - $(q, a) \to (q', a, \triangleright)$ für alle $(q, a, q') \in \delta$;
  - $(q, a) \to ((p, r), a, \triangleright)$ für alle $(q, a, p), (q, a, r) \in \delta$ mit $p \neq r$;
  - $((p, r), a) \to ((p', r'), a, \triangleright)$ für alle $(p, a, p'), (r, a, r') \in \delta$;
  - $((p, r), \Box) \to (q_+, \Box, \diamond)$ für alle $p, r \in F$.

Sei $w = a_1 \dots a_n$. Die Idee ist, den Beginn i von zwei unterschiedlichen Läufen von A auf w zu raten: simuliere zunächst auf $a_1 \dots a_{i-1}$ *ein* Verhalten von A, d.h. verfolge einen Lauf von A auf $a_1 \dots a_{i-1}$. Angenommen, dieser Lauf endet q. Falls es von q aus für den nächsten Buchstaben a_i zwei mögliche Fortführungen p und r des Laufes gibt, und es von p und r von A auf $a_{i+1} \dots a_n$ einen akzeptierenden Lauf gibt, so wird w akzeptiert von M.

Punktevergabe: ●1 ●2 für richtige Idee, ●3 ●4 ●5 für richtige Umsetzung ●6 Formal keinerlei Fehler

Hausaufgabe 3.5 (Turing-berechenbare Funktionen) (11)

Sei $\Sigma = \{a, b\}$. Wir definieren die Infixrelation $\leq_i$ über Σ^* durch $u \leq_i w$ falls $v_1, v_2 \in \Sigma^*$ mit $w = v_1 \cdot u \cdot v_2$ existieren. Wir definieren die partielle Funktion $f : \{a, b, \#\}^* \dashrightarrow \{0, 1\}$ durch

$$f(u\#w) = \begin{cases} 1 & \text{falls } u \leq_i w \\ 0 & \text{sonst} \end{cases}$$

für alle $u, w \in \{a, b\}^*$. Beispielsweise ist $f(ab\#aab) = 1$, $f(aa\#b) = 0$, und $f(aa) = \perp$ ist undefiniert.

Zeigen Sie, dass f Turing-berechenbar ist. Bitte geben Sie aussagekräftige Erläuterungen zur Verhaltensweise Ihrer Turingmaschine.

LÖSUNG: Wir definieren eine deterministische 2-Band-Turingmaschine M mit $T(M) = f$.

Sei $M = (Q, \Sigma \cup \{\#\}, \Gamma, \Box, \Delta, q_0, q_+, q_-)$, wobei

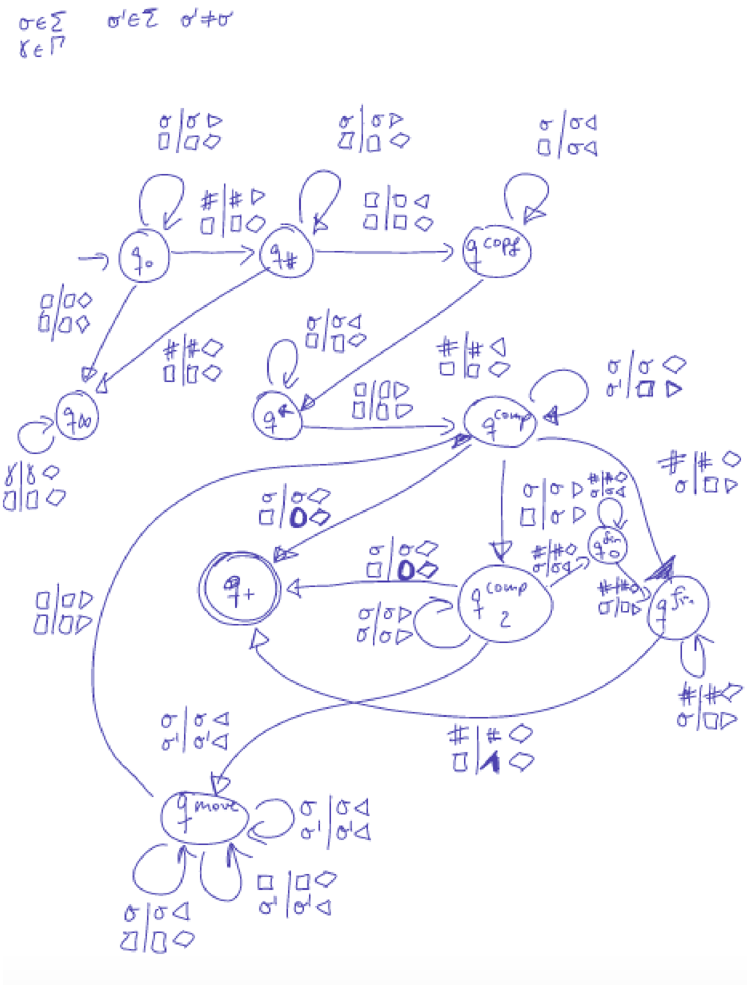
- $Q = \{q_0, q_\#, q^{copy}, q^{\leftarrow}, q^{comp}, q_\infty, q_+, q_-, q_2^{comp}, q_0^{fin}, q^{fin}, q^{move}\}$,
- Δ besteht aus folgenden Transitionen:
  - Zunächst prüft M ob das Eingabewort die Form $\Sigma^*\#\Sigma^*$ besitzt.
    - für alle $\gamma \in \Gamma$ definiere $(q_\infty, \langle \gamma, \Box \rangle) \to (q_\infty, \langle (\gamma, \diamond), (\Box, \diamond) \rangle)$ für Endlosschleife,

    * $(q_0, \langle\square,\square\rangle) \to (q_\infty, \langle(\square,\diamond),(\square,\diamond)\rangle)$ für Eingabe von der Form Σ^* ist f nicht definiert - gehe in q_∞;
    * Für alle $\sigma \in \Sigma$ definiere $(q_0, (\sigma,\square)) \to (q_0, \langle\sigma,\triangleright),(\square,\diamond)\rangle)$ Lesen von $\sigma \in \Sigma$ vor dem Trennsymbol.
    * $(q_0, \langle\#,\square\rangle) \to (q_\#, \langle(\#,\triangleright),(\square,\diamond)\rangle)$ Lesen von Trennsymbol #
    * $(q_\#, \langle\#,\square\rangle) \to (q_-, \langle(\#,\diamond),(\square,\diamond)\rangle)$ Falls ein weiteres Trennsymbol gelesen wird, hat Eingabewort nicht die richtige Form. Gehe in q_∞.
    * Für alle $\sigma \in \Sigma$, define $(q_\#, \langle\sigma,\square\rangle) \to (q_\#, \langle(\sigma,\triangleright),(\square,\diamond)\rangle)$ gehe bis zum rechten Ende der Eingabe
    * $(q_\#, \langle\square,\square\rangle) \to (q^{copy}, \langle(\square,\triangleleft),(\square,\diamond)\rangle)$ Eingabe ist korrekt; LSM steht auf Band 1 am Ende der Eingabe und M ist in Zustand q^{copy}.
  - Kopiere nun den Teil hinter dem Trennsymbol von Band 1 auf Band 2 (um leichter vergleichen zu können): für alle $\sigma \in \Sigma$, definiere $(q^{copy}, \langle\sigma,\square\rangle) \to (q^{copy}, \langle(\sigma,\triangleleft),(\sigma,\triangleleft)\rangle)$ und $(q^{copy}, \langle\#,\square\rangle) \to (q^{\leftarrow}, \langle(\#,\triangleleft),(\square,\diamond)\rangle)$
  - Laufe in $q^{\leftarrow}$ auf Band bis zum Anfang des Eingabewortes: für alle $\sigma \in \Sigma$, definiere $(q^{\leftarrow}, \langle\sigma,\square\rangle) \to (q^{\leftarrow}, \langle(\sigma,\triangleleft),(\square,\diamond)\rangle)$ und $(q^{\leftarrow}, \langle\square,\square\rangle) \to (q^{comp}, \langle(\square,\triangleright),(\square,\triangleright)\rangle)$. SLK steht nun auf Band 1 und Band 2 vor Beginn des jeweiligen Bandinhalts. Hier beginnt nun in q^{comp} der eigentliche Infix-Check.
  - In q^{comp} wird geprüft, ob Eingabe auf Band 1 vor dem # ein Infix des Wortes auf Band 2 ist. Die Idee dazu ist wie folgt. Das Wort auf Band 1 (vor #) ist Infix des Wortes auf Band 2 falls es eine Position i im Wort auf Band 2 gibt, von wo an die Wörter auf Band 1 und Band 2 übereinstimmen solange bis auf Band 1 das Trennsymbol # gelesen wird.

    Beginnend ab Position 1 auf Band 2 wird durch synchrone Bewegung von links nach rechts auf beiden Bändern geprüft, ob Symbol auf Band 1 gleich zu Symbol auf Band 2 ist, solange bis (a) Ungleichheit auftritt (Neubeginn nötig) (b) auf Band 2 $\square$ gelesen wird (Miserfolg) (c) auf Band 1 # gelesen wird (Erfolg). Das dabei erste Symbol auf Band 2 muss markiert (hier: durch $\square$ ersetzt) werden, damit wir dieses Symbol nur genau einmal als Start des Vergleichs verwenden.
  - Verschiedene Fälle:
    * Für alle $\sigma \in \Sigma$ definiere $(q^{comp}, \langle\sigma,\sigma\rangle) \to (q_2^{comp}, \langle(\sigma,\triangleright),(\square,\triangleright)\rangle)$.
      Das erste Zeichen auf Band 1 und 2 ist gleich, vergleiche weiter in Zustand q_2^{comp}, aber lösche das erste Zeichen von Band 2 damit ein eventuell neuer Beginn ab der nächsten Position beginnt.
    * Für alle $\sigma, \sigma' \in \Sigma$ mit $\sigma \neq \sigma'$ definiere $(q^{comp}, \langle\sigma,\sigma'\rangle) \to (q^{comp}, \langle(\sigma,\diamond),(\square,\triangleright)\rangle)$.
      Die jeweils ersten Zeichen sind nicht gleich: lösche auf Band 2 und beginne direkt neuen Vergleich ab der nächsten Position.
    * Sonderfall kein Symbol auf Band 2 mehr übrig (Miserfolg): gib 0 aus auf Band 2 und akzeptiere. Für alle $\sigma \in \Sigma$ definiere $(q^{comp}, \langle\sigma,\square\rangle) \to (q_+, \langle(\sigma,\diamond),(0,\diamond)\rangle)$

    * Sonderfall: auf Band 1 kommt direkt ein # (Erfolg) Dann kein Vergleich nötig. Ersetze Inhalt von Band 2 durch 1 und akzeptiere. Für alle $\sigma \in \Sigma$ definiere $(q^{comp}, \langle \#, \sigma \rangle) \rightarrow (q^{fin}, \langle (\#, \diamond), (\square, \triangleright) \rangle)$ und $(q^{fin}, \langle \#, \sigma \rangle) \rightarrow (q^{fin}, \langle (\#, \diamond), (\square, \triangleright) \rangle)$ und $(q^{fin}, \langle \#, \square \rangle) \rightarrow (q_+, \langle (\#, \diamond), (1, \diamond) \rangle)$

- In q_2^{comp} hat ähnliche Verhaltensweise wie q^{comp}:
    * Für alle $\sigma \in \Sigma$ definiere $(q_2^{comp}, \langle \sigma, \sigma \rangle) \rightarrow (q_2^{comp}, \langle (\sigma, \triangleright), (\sigma, \triangleright) \rangle)$.
      Das erste Zeichen auf Band 1 und 2 ist gleich, vergleiche weiter in Zustand q_2^{comp}.
    * Für alle $\sigma, \sigma' \in \Sigma$ mit $\sigma \neq \sigma'$ definiere $(q_2^{comp}, \langle \sigma, \sigma' \rangle) \rightarrow (q^{move}, \langle (\sigma, \triangleleft), (\sigma', \triangleleft) \rangle)$.
      Die jeweils ersten Zeichen sind nicht gleich. Es muss nun auf Band 1 dann auf Band 2 zum jeweils linken Ende gelaufen und ein neuer Vergleichsprozess in Gang gesetzt werden: $(q^{move}, \langle \sigma, \sigma' \rangle) \rightarrow (q^{move}, \langle (\sigma, \triangleleft), (\sigma', \triangleleft) \rangle)$, and $(q^{move}, \langle \square, \sigma' \rangle) \rightarrow (q^{move}, \langle (\square, \diamond), (\sigma', \triangleleft) \rangle)$, and $(q^{move}, \langle \sigma, \square \rangle) \rightarrow (q^{move}, \langle (\sigma, \triangleleft), (\square, \diamond) \rangle)$, and $(q^{move}, \langle \square, \square \rangle) \rightarrow (q^{comp}, \langle (\square, \triangleright), (\square, \triangleright) \rangle)$.
    * Sonderfall kein Symbol auf Band 2 mehr übrig (Miserfolg): gib 0 aus auf Band 2 und akzeptiere. Für alle $\sigma \in \Sigma$ definiere $(q_2^{comp}, \langle \sigma, \square \rangle) \rightarrow (q_+, \langle (\sigma, \diamond), (0, \diamond) \rangle)$
    * Sonderfall: auf Band 1 kommt direkt ein # (Erfolg) Dann kein Vergleich nötig. Ersetze Inhalt von Band 2 durch 1 und akzeptiere. Für alle $\sigma \in \Sigma$ definiere $(q_2^{comp}, \langle \#, \sigma \rangle) \rightarrow (q_0^{fin}, \langle (\#, \diamond), (\sigma, \triangleleft) \rangle)$, $(q_0^{fin}, \langle \#, \sigma \rangle) \rightarrow (q_0^{fin}, \langle (\#, \diamond), (\sigma, \triangleleft) \rangle)$, und $(q_0^{fin}, \langle \#, \square \rangle) \rightarrow (q^{fin}, \langle (\#, \diamond), (\square, \triangleright) \rangle)$.



Punktevergabe: ●7 ●8 ●9 Für die richtige Idee der eigentlichen Aufgabe. Diese sollte beinhalten: Check ob Eingabewort korrekt ist (Sonderfälle beachten), sowie die Prüfung des Infixchecks. ●10 ●11 Deterministische (!) Turingmaschine formal weitestgehend richtig aufgeschrieben ●12 ●13 Gute Erklärungen! ●14 ●15 ●16 korrekte Ausgabe 1, 0 bzw. Endlosschleife mit richtiger Position des SLK am Anfang von Ausgabeband (Band k bei k-Band TMs) ●17 Keinerlei formale Fehler

Hausaufgabe 3.6 (LOOP-berechenbare Funktionen) (5)

Definiere die Funktion $g : \{0,1\}^3 \to \{0,1\}$ durch

$$g(a_1, a_2, a_3) = (a_1 \wedge a_2) \vee \neg a_3$$

für alle $a_1, a_2, a_3 \in \{0,1\}$.

Zeigen Sie, dass g LOOP-berechenbar ist.

LÖSUNG: $x_4 = 1;$

```
x4 = 1;
LOOP(x3) {
    x4 = 0;
    LOOP(x1) {
        LOOP(x2) {
            x4 = 1;
        }
    }
}
x1 = x4;
```

●18 ●19 Richtige Idee ●20 ●21 Formal richtig umgesetzt (also die LOOP Syntax und Semantik einhaltend) und richtiges Ergebnis ●22 Alles absolut richtig