4. Aufgabenblatt zur Vorlesung

Grundlagen der Theoretischen Informatik, SoSe 2008

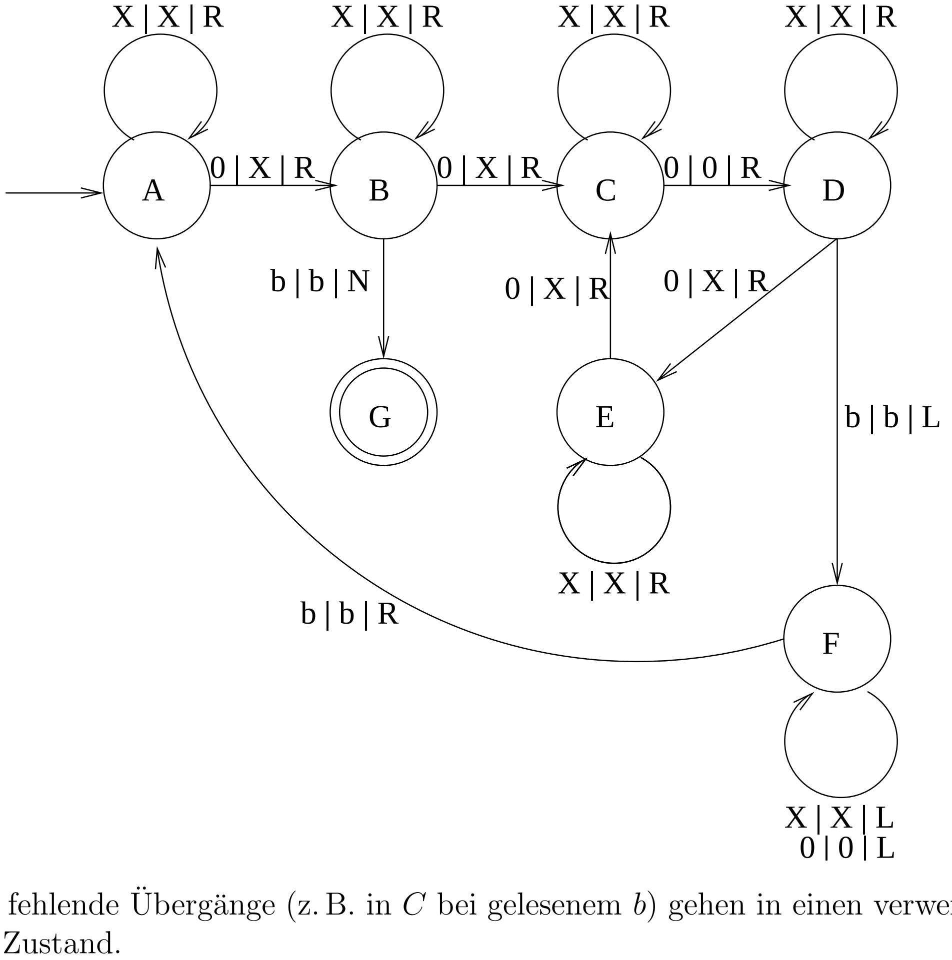
(Dr. Frank Hoffmann)

Lösung von David S. Karcher

Aufgabe 1 Turing–Maschinen (4+2+2+2 Punkte)

(a) Geben Sie explizit eine Turing–Maschine an, die die Sprache $L = \{0^{3^n} \mid n \geq 0\}$ über dem Alphabet $\{0\}$ entscheidet. Beschreiben Sie kurz, was die einzelnen Zustände tun.

Lösung:



Alle fehlende Übergänge (z. B. in C bei gelesenem b) gehen in einen verwerfenden Zustand.

Die Idee hinter dieser Turingmaschine ist, die Eingabe solange durch drei zu teilen, bis eins herauskommt (dann geht sie von B in den akzeptierenden Zustand G). Hierfür wird bei jeder Division in A angefangen, und es werden zwei Drittel der Nullen durch X überschrieben (jeweils die ersten beiden, und die dritte wird stehen gelassen, dies ist der Übergang von C nach D), dieser Vorgang wird solange iteriert (Schleife aus C,D und E), bis wir das Ende der Eingabe erreichen, falls dort ein Dreierblock endet, wird der Schreib-/Lesekopf wieder auf den Anfang der Eingabe gefahren (F), dann startet die Turingmaschine in A mit der nächsten Division. Ansonsten wird die Eingabe verworfen.

(b) Beweisen Sie, dass man jede Turing–Maschine mit beidseitig unbeschränktem Eingabe-/Arbeitsband (wie in der Vorlesung definiert) durch eine mit einseitig unbeschränktem Band simulieren kann.

Lösung:

Die Idee unserer Simulation beruht 2–Spur–Turing–Maschine, die auf dem oberen Band den positiven Bandteil speichert, und auf dem unteren den negativen. Die Eingabe steht ganz links auf dem oberen Band, und wird bei der Initalisierung um ein Zeichen nach rechts geschoben, und der linke Bandrand wird durch ein besonderes Zeichen markiert. Nun wird der Schreib-/Lesekopf auf den Anfang der Eingabe bewegt.

Die Berechnung läuft nun analog zu der, der Ursprungs Turing–Maschine, falls man das Sonderzeichen vom oberen Band erreicht, so wird auf dem unteren Band weitergearbeitet (natürlich sind die Kopfbewegungen invertiert), falls man man das Sonderzeichen vom unteren Band erreicht, so wird auf dem oberen Band weitergearbeitet (natürlich sind die Kopfbewegungen nicht invertiert).

(c) Beschreiben Sie die Simulation eines dfa durch eine Turing–Maschine. Beachten Sie insbesondere die verschiedenen Akzeptierungsmodi.

Lösung:

Sei $M = (Q, \Sigma, \delta, q_0, F)$ ein dfa, wir konstruieren nun eine Turingmaschine M', die $L(M)$ entscheidet. Es gilt $M' = (Q \cup \{q_{\text{ja}}, q_{\text{nein}}\}, \Sigma, \Sigma \cup \{\underline{b}\}, \delta', q_0, \underline{b}, \{q_{\text{ja}}, q_{\text{nein}}\})$, wobei q_{ja} ein akzeptierender und q_{nein} ein verwerfender Zustand ist. Die Überführungsfunktion δ' ist wie folgt definiert:

$$\delta'(q,a) = \begin{cases} (\delta(q,a), R), & \text{falls } a \neq \underline{b} \\ (q_{\text{ja}}, a, N), & \text{falls } a = \underline{b} \text{ und } q \in F \\ (q_{\text{nein}}, a, N), & \text{falls } a = \underline{b} \text{ und } q \notin F \end{cases}$$

Die Turingmaschine vollzieht also die gleichen Übergänge wie der dfa, und überprüft am Ende, ob sie akzeptieren oder verwerfen muss.

(d) Beschreiben Sie eine Turing–Maschine, die für unär kodierte natürliche Zahlen einen Primzahltest durchführt. Schätzen Sie die Anzahl der Konfigurationswechsel in der Länge der Eingabe ab.

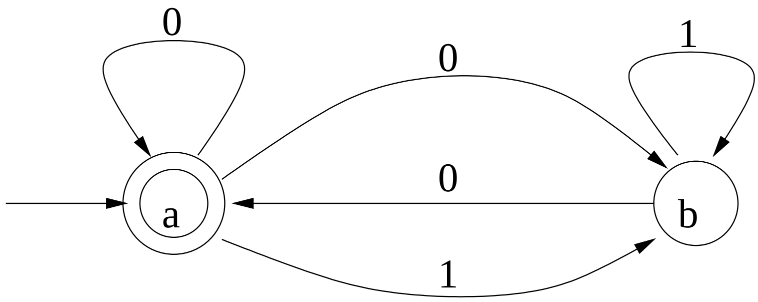
Lösung:

Wir kopieren die Eingabe einmal nach links, abgetrennt durch ein besonderes Symbol ($11111 \to 11111X11111$), dies geht in $\Theta(n^2)$ vielen Schritten, indem wir es Zeichen für Zeichen tun. Dann überprüfen wir ob die uspröngliche Einbgabe Länge eins hatte, wenn ja verwerfen wir. Nun werden links vom X zwei Einsen markiert (z. B. mit Z), und es wird geprüft, ob die Eingabe glatt durch zwei teilbar ist, indem wir sukzessive immer zwei Einsen der Eingabe mit Y markieren (, hierzu muss man jeweils eine der markierten Einsen anders markieren [z. B. mit W] und dann eine Eins der Eingabe markieren, dann die nächste markierte auf der linke Seite, und wieder eine rechts, nun werden werden die auf der linken Seite wieder so markiert, dass man weiter auf der rechten Seite Einsen markieren kann, also mit W). Wenn nun auf der rechten Seite alle Einsen markiert sind, aber links noch welche übrig sind (z.B $1111WZXYYYYYYY$), also im Beispiel das Z, so wissen wir das die Eingabe nicht durch die Zahl teilbar war, die wir gerade geprüft haben, nun wird der rechte Teil wieder mit Einsen überschrieben und links wird eine weitere markiert (in unserem Beispiel

also $111WWWX1111111$), und es wird analog verfahren, falls die Division irgendwann aufgeht, so können wir die Eingabe verwerfen, da sie ja dann keine Primzahl ist, ansonsten akzeptieren wir, wenn die gesammte linke Seite markiert ist (also wieder im Beispiel $WWWWWWWX1111111$, wird akzeptiert, da sieben ein Primzahl ist).
Für jeden potentielle Teiler haben wir, analog zum Kopieren oben, quadratischen Aufwand, da wir n Teiler testen müssen ergibt sich also insgesamt als Laufzeit $T(n) = \Theta(n^3)$.

Aufgabe 2 Nochmal nfa und dfa (4 Punkte)

Geben Sie für den unten dargestellten nfa einen äqivalenten und mininmalen dfa und den regulären Ausdruck an. Beschreiben Sie diese Sprache in Worten.



Aufgabe 3 Nichtregulär (6 Punkte)

(a) Zeigen Sie, dass für jedes feste k die Sprache $L = \{0^k 1^{n-k} 0^n \mid n \geq k\}$ nicht regulär ist.
Hinweis: Um einer unangenehmen Fallunterscheidung aus dem Weg zu gehen, erinnern Sie sich besser daran, dass reguläre Sprachen gegenüber einer ganzen Reihe von Operationen abgeschlossen sind.
Lösung:
Falls L regulär ist, so wäre die rückwärts gelesene Sprache $L^{\text{rev}} = \{0^n 1^{n-k} 0^k \mid n \geq k\}$ es ebenso. Wir zeigen nun das der Index der Neroderelation für L^{rev} unendlich ist, also kann sie nicht regulär sein, und somit ist auch L nicht regulär.
Sei $i, j \in \mathbb{N}$ und $i < j$, wir zeigen nun $[0^i] \neq [0^j]$.
Hierzu betrachten wir $x = 1^{i-k} 0^k$, wie man leicht sieht, folgt $0^i \circ x \in L^{\text{rev}}$ und $0^j \circ x \notin L^{\text{rev}}$. Also stehen 0^i und 0^j nicht in Relation. Und folglich ist L nicht regulär.

(b) Beweisen Sie die Nichtregularität der Sprache

$$\text{ADD} = \{x = y + z \mid x, y, z \text{ sind Binärzahlen und es gilt } x = y + z\}$$

Das Alphabet ist $\{0, 1, +, =\}$.
Lösung:
Wir geben wieder eine unendliche Familie von Äquivalenzklassen der Neroderelation an, und zeigen somit die Nichregularität.
Sei $i, j \in \mathbb{N}$ und $i < j$, wir zeigen nun $[1^i] \neq [1^j]$.
Hierzu betrachten wir $x = {=}1^i{+}0$, wie man leicht sieht, folgt $1^i \circ x \in \text{ADD}$ und $1^j \circ x \notin \text{ADD}$, da nur die erste Gleichung gilt. Also stehen 1^i und 1^j nicht in Relation. Und folglich ist ADD nicht regulär.

(c) Beweisen Sie die Nichtregularität von $L = \{w\overline{w} \mid w \in \Sigma^*\}$über $\Sigma = \{0, 1\}$. Dabei bezeichnet $\overline{w}$ das aus w durch Vertauschen der Nullen bzw. Einsen entstehende Wort.
Lösung:
Diese Aufgabe lösen wir zur Abwechslung mal mit dem Pumping-Lemma (es würde natürlich auch recht analog über den Index der Neroderelation funktionieren).
Zu $n \in \mathbb{N}$ wählen wir das Wort $x = 0^n 1^n$, für jede Zerlegung $x = u \circ v \circ w$ mit $|uv| \leq n$ und $|v| \geq 1$, ist klar, dass sich der uv Anteil irgendwo am Anfang des Nullerblocks befindet, und somit v nur aus Nullen besteht. Für alle $i \neq 1$ ist nun aber $u \circ v^i \circ w = 0^{n+(i-1)|v|} 1^n$ nicht in L, da dann $n + (i-1)|v| \neq n$. Also ist L nicht regulär.