Theoretische Informatik
Lösung für das Übungsblatt 2 (2024W)

Aufgabe 2.1
Betrachten Sie die Turingmaschine $M = (\{q_0, q_1, q_2, q_3\}, \{0, 1\}, \{0, 1, B\}, \delta, q_0, B, \{q_3\})$,

wobei δ wie folgt spezifiziert ist:

Zustand	0	1	B
q_0	$(q_0, 0, R)$	$(q_0, 1, R)$	(q_1, B, L)
q_1	$(q_2, 1, L)$	$(q_2, 1, L)$	(q_2, B, L)
q_2	$(q_2, 0, L)$	$(q_2, 0, L)$	(q_3, B, R)

a) Wenden Sie M auf den Eingabestring 110, also $[6]_2$, an.
Genauer: Geben Sie eine Folge von Konfigurationen an, die mit der Anfangskonfiguration $q_0 110$ beginnt und einer Endkonfiguration endet. Wie lautet die Ausgabe als natürliche Zahl?

b) Beschreiben Sie die Arbeitsweise von M für beliebige (binäre) Eingabestrings informell.

c) Welche Funktion f_M vom Typ $\mathbb{N} \to \mathbb{N}$ wird von M berechnet?

Lösung 2.1

a) $q_0 110 \Rightarrow 1q_0 10 \Rightarrow 11q_0 0 \Rightarrow 110q_0[B] \Rightarrow 11q_1 0 \Rightarrow 1q_2 11 \Rightarrow q_2 101 \Rightarrow q_2 B001 \Rightarrow q_3 001$.
Da q_3 ein Endzustand ist hält M mit Bandinhalt 001 (und rechts und links nur Bs). Der Lese/Schreibkopf steht über dem am weitesten links stehenden Vorkommen von 0. Die Ausgabe ist 1, da $[001]_2 = 1$.

b) Im Anfangszustand q_0 geht M schrittweise (wiederholt) nach rechts ohne den Bandinhalt zu ändern bis ein Leersymbol (B) erreicht wird. Dieses Vorkommen von B bleibt unverändert, aber M geht nun in den Zustand q_1 und stellt den Lese/Schreibkopf eine Zelle weiter links, also über das letzte Symbol des binären Eingabestrings. Wenn dieses Symbol 0 lautet, so wird es durch 1 ersetzt. Wenn dieses Symbol 1 lautet, so bleibt es unverändert 1. In beiden Fällen ist M nun im Zustand q_2. In diesem Zustand geht die Maschine wiederholt nach links und ersetzt dabei jedes Binärsymbol, das links von der soeben auf 1 gesetzten Bandzelle steht, durch 0 bis der Lese/Schreibkopf über einem B steht. Dieses B bleibt unverändert, aber M geht nun einen Schritt nach rechts und in den Endzustand q_3. Der Lese/Schreibkopf steht somit über dem ersten (von links gelesen) Symbol des resultierenden Binärstrings.

c) M berechnet $f_M(n) = 1$, also die konstante Funktion, die jede Zahl auf 1 abbildet.

Aufgabe 2.2
Schreiben und analysieren Sie ein 1-RM Programm P, das die Funktion $f_P(n) = n \bmod 2$ berechnet. Genauer:

a) Geben Sie P an.

b) Geben Sie eine vollständige Konfigurationsfolge für die Eingabe 3 an.

c) Beschreiben Sie die Arbeitsweise von P für eine beliebige Eingabe n in Worten.

Lösung 2.2

a) $P = (0,\ \{(0, t_1, 5, 1),\ (1, S_1, 2),\ (2, t_1, 4, 3),\ (3, S_1, 0), (4, A_1, 5)\})$

b) 0:(3) $\Rightarrow$ 1:(3) $\Rightarrow$ 2:(2) $\Rightarrow$ 3:(2) $\Rightarrow$ 0:(1) $\Rightarrow$ 1:(1) $\Rightarrow$ 2:(0) $\Rightarrow$ 4:(0) $\Rightarrow$ 5:(1)
5 ist keine Kennmarke und daher eine Endmarke. Das Programm hält mit Ausgabe 1.

c) Die Eingabezahl n steht anfangs im einzigen Register, also $R(1) = n$. P beginnt mit einem Nulltest an der Kennmarke (= Startmarke) 0. Wenn $R(1)$ bereits auf 0 steht, dann gilt $n \bmod 2 = 0$ und das Programm terminiert mit $R(1) = 0$ als Ausgabe. Andernfalls geht P zur Kennmarke 1 und dekrementiert $R(1)$ im Übergang zur Kennmarke 2. Dort, an der Kennmarke 2, erfolgt ein weiterer Nulltest. Falls $R(1) = 0$, dann war die ursprüngliche Eingabezahl n ungerade. P stellt daher $R(1)$ auf 1 (Ausführung von A_1 im Übergang von Kennmarke 4 aus Endmarke 5) und terminiert ($n \bmod 2 = 1$). Falls der Nulltest an der Kennmarke 2 negativ ist, also $R(1) > 0$ gilt, wird an der Kennmarke 3 das Register dekrementiert und geht wieder zur Startmarke 0. In diesem Fall wurde in der Schleife zwei mal dekrementiert. Es wir nun mit $R(1) = n - 2$ fortgesetzt, was zum gewünschten Ergebnis führt da für $n \geq 2$ gilt: $n \bmod 2 = (n - 2) \bmod 2$.

Aufgabe 2.3

Nennen Sie alle Fehler in folgender (misslungener) Auswertung eines λ-Terms. Die Ziffern stehen für die auf Folie 39 definierten Church-Numerale und **mult** für den dort angegeben λ-Term für die Multiplikation.
Hinweis: Folgefehler sollen unberücksichtigt bleiben. Für jede Zeile, außer der Zeile (0), soll separat überprüft werden, ob sie das Ergebnis einer korrekt durchgeführten β- oder α-Reduktion ist. Es zählt auch als Fehler, wenn eine der Definitionen oder eine der (optionalen) Schreibkonventionen (Folie 30) nicht korrekt verwendet wurde, beziehungsweise wenn eine Unterstreichung nicht den verwendeten Redex bezeichnet. Letzteres gilt auch für Zeile (0).

$$\begin{array}{rlcl}
(0) & \textbf{mult}\,4\,2 & = & \underline{(\lambda mnf.m(nf))\,4\,2} \\
(1) & & \to_\beta & \underline{(\lambda nf.2(nf))\,4} \\
(2) & & \to_\beta & \underline{\lambda f.2(4f)} \\
(3) & & =_\alpha & \lambda f.2(\underline{\lambda f'y.f'(f'(f'y))f}) \\
(4) & & \to_\beta & \lambda f.2(\lambda y.f(f'(f'y))) \\
(5) & & =_\alpha & \lambda f.\underline{(\lambda f'z.f'(f'z))(\lambda y.f(f(fy)))} \\
(6) & & \to_\beta & \lambda f.(\lambda z.(\lambda y.f(f(f(y)))(\underline{\lambda y.f'(f'(f'y))z}))) \\
(7) & & \to_\beta & \lambda f.(\underline{\lambda z.(\lambda y.f(f(f(y)))(f'(f'(f'y))))}) \\
(8) & & \to_\beta & \lambda f.(\lambda z.f(f(f(f'(f'(f'y)))))) \\
(9) & & =_\alpha & \lambda fz.f(f(f(f(f(fz))))) \\
(10) & & =_\alpha & \lambda fx.f(f(f(f(f(fx))))) = 8
\end{array}$$

Lösung 2.3

Zeile (0): Es wurde der gesamte λ-Term unterstrichen. Dieser hat aber nicht die Form eines Redex.

Zeile (1): Der einzige Redex in Zeile (0) ist $\underline{\lambda mnf.m(nf)\,4}$. Wäre diese korrekt reduziert worden, so wäre das Resultat für den gesamten λ-Term $\underline{(\lambda nf.4(nf))\,2}$.

Zeile (2): Die β-Reduktion ist korrekt erfolgt. Allerdings ist der gesamte λ-Term unterstrichen, obwohl er kein Redex ist.

Zeile (3): Hier wurde nicht das Church-Numeral '4', sondern die Definition des Church-Numerals von '3' expandiert. Außerdem sollte man den expandierten λ-Term in Klammern setzen, da der unterstrichene Teil-Term sonst kein Redex gemäß der Definition auf Folie 33 ist. (Die

Umbenennung der Variablennamen — f' für f und y für x — entspricht einer zweifachen α-Reduktion. Das Zeichen '$=_\alpha$' wurde also korrekt verwendet.)

Zeile (4): Die β-Reduktion des in Zeile 3 (korrekt) bezeichneten Redex verlangt, dass alle drei (und nicht nur das erste) Vorkommen von f' im Body des Redex durch 'f' ersetzt werden.

Zeile (5): In dieser Zeile liegt kein Fehler vor: Die Definition von '2' wurde korrekt expandiert und die dabei erfolgte Umbenennung der Variablen von f in f' und x in z korrekt mit '$=_\alpha$' vermerkt. Der unterstrichene Teil-Term ist tatsächlich ein Redex.

Zeile (6): In der β-Reduktion müsste f' —zwei mal, im Body des Redex — durch $\lambda y.f(f(fy))$ und nicht einmal durch $\lambda y.f(f(fy))$ und einmal durch $\lambda y.f'(f'(f'y))$ ersetzt werden.

Zeile (7): Die β-Reduktion wäre nur dann korrekt, wenn das letzte Vorkommen von y durch z ersetzt würde. Außerdem ist der unterstrichene Teil-Term kein Redex. Der einzige Redex im gesamten λ-Term ist $\lambda y.f(f(f(y)))(f'(f'(f'y)))$.

Zeile (8): Es fehlt eine schließende Klammer. Mit dieser Korrektur wäre der λ-Term das korrekte Ergebnis einer β-Reduktion, wenn nicht der in (7) unterstrichene Teil-Term, sondern der Teil-Term $\lambda y.f(f(f(y)))(f'(f'(f'y)))$ als Redex verwendet wird.

Zeile (9): Man kann zwar, laut den Schreibkonventionen '$\lambda fz \ldots$' anstatt '$\lambda f.(\lambda z \ldots)$' schreiben, aber es liegt keine korrekte α-Reduktion vor: Die substituierte Variable darf nicht bereits gebunden vorkommen. Hier wurde aber y durch die bereits gebundene Variable z, sowie f' durch die bereits gebundene Variable f ersetzt.

Zeile (10): Der finale λ-Term ist das Church-Numeral für die Zahl 6, nicht das für 8.

Aufgabe 2.4

a) Präsentieren Sie einen **Pr**-Funktionsausdruck E für die Funktion $E(n) = n^2 + n + 2$. Benutzen Sie $\sum_{i=0}^{n} i = n(n+1)/2$. Sie dürfen dabei den bereits in der Vorlesung definierten Ausdruck *Add* für die Addition verwenden. Vermeiden Sie jedoch einen Teilausdruck für die Multiplikation zu verwenden.

b) Werten Sie Ihren Ausdruck E für die Eingabe 3 schrittweise aus. Behandeln Sie dabei *Add* wie eine Grundfunktion. (Übersetzen Sie also *Add* direkt in entsprechende Additionen.)

c) Analysieren Sie die Funktion $\overline{\mu}E$. Geben Sie zwei Eingabewerte n an, für die $\overline{\mu}E(n)$ definiert ist, sowie zwei Eingabewerte n, für die $\overline{\mu}E(n)$ undefiniert ist.

d) Welche Funktion wird durch $\mu_0 E$ dargestellt?

Lösung 2.4

a) Wir schreiben $Sum(n)$ für $\sum_{i=0}^{n} i$ und beobachten, dass diese Funktion durch $Sum(0) = 0$ und $Sum(n+1) = Sum(n) + n + 1$ induktiv definiert ist. Entsprechend erhalten wir den Funktionsausdruck $Sum = \mathbf{Pr}(C_0^0, S \circ Add)$, wobei Add der auf Folie 51 definierte Ausdruck für die Addition ist. (Beachten Sie, dass sich $Sum(n+1)$ auf das Wertepaar $(n, Sum(n))$ bezieht.)

Wegen $Sum(n) = n(n+1)/2 = (n^2+n)/2$ gilt $E(n) = 2 \cdot Sum(n) + 2$. Daher erhalten wir die gesuchte Definition $E = S \circ S \circ Add \circ (Sum, Sum)$, wobei $Sum = \mathbf{Pr}(C_0^0, S \circ Add)$. Alternativ könnten wir E, z.B., auch durch den Ausdruck $Add \circ (C_2^1, Add \circ (P_1^1, P_1^1) \circ Sum)$ definieren.

b) Wir werten zunächst $Sum(3) = \mathbf{Pr}(C_0^0, S \circ Add)(3)$ schrittweise aus:
$Sum(0) = \mathbf{Pr}(C_0^0, S \circ Add)(0) = C_0^0 = 0$
$Sum(1) = \mathbf{Pr}(C_0^0, S \circ Add)(1) = 1 + Add(0, Sum(0)) = 1 + (0+0) = 1$
$Sum(2) = \mathbf{Pr}(C_0^0, S \circ Add)(2) = 1 + Add(1, Sum(1)) = 1 + (1+1) = 3$

$Sum(3) = \mathbf{Pr}(C_0^0, S \circ Add)(3) = 1 + Add(2, Sum(2)) = 1 + (2 + 3) = 6$
Damit erhalten wir schließlich $E(3) = S \circ S \circ Add \circ (Sum, Sum)(3) = 1 + 1 + (6 + 6) = 14$.

c) Es gilt $\overline{\mu}E(n) = \min_k[E(k) = n]$. Wir suchen also kleinste ganzzahlige, nicht-negative Lösungen (für k) von $k^2 + k + 2 = n$.
Für $n = 2$ erhalten wir $\overline{\mu}E(2) = 0$, da $0^2 + 0 + 2 = 2$.
Für $n = 4$ erhalten wir $\overline{\mu}E(4) = 1$, da $1^2 + 1 + 2 = 4$.
Hingegen ist $E(0)$ und $E(1)$ (aber auch, z.B., $E(3)$) undefiniert.

d) Es gilt $\mu_0 E = \min_k[E(k) = 0]$. Es gibt aber keine ganzzahligen, nicht-negative Lösungen von $k^2+k+2 = 0$. Daher stellt $\mu_0 E$ eine nirgends definierte 0-stellige Funktion dar. (Beachten Sie, dass $\mu_0 f$ für jede 1-stellige Funktion f von keinem Eingabewert abhängt und daher immer 0-stellig ist. Z.B. gilt $\mu_0 P_1^1 = C_0^0$.)

Aufgabe 2.5
Erklären Sie für jedes der folgenden Entscheidungsprobleme, warum der Satz von Rice (in einer der beiden Varianten) anwendbar bzw. nicht anwendbar ist um zu folgern, dass das Problem unentscheidbar ist.

a) Berechnet eine vorgelegte TM die Identitätsfunktion $f(n) = n$?

b) Lässt eine vorgelegte TM das Band während der Berechnung unverändert?

c) Ist die von einer vorgelegten TM berechnete Funktion μ-rekursiv?

d) Ist die von einer vorgelegten TM berechnete Funktion primitiv-rekursiv?

e) Besteht eine (als entsprechende TM) vorgelegte Sprache aus nur einem Wort?

f) Ist eine (als entsprechende TM) vorgelegte Sprache rekursiv aufzählbar?

Lösung 2.5

a) Es gibt natürlich Turingmaschine, die die Identitätsfunktion berechnen. Anderseits ist das natürlich nicht die einzige partiell rekursive, also Turing-berechenbare Funktion. Daher handelt es sich hier um eine nicht-triviale Funktions-Eigenschaft. Der Satz von Rice für Funktionen ist also anwendbar. Entsprechend ist das Problem unentscheidbar.

b) Es gibt Paare von Turingmaschinen, die dieselbe Sprache akzeptieren (z.B. die leere Sprache) bzw. dieselbe Funktion berechnen (z.B. die nirgends definierte partielle Funktion vom Typ $\mathbb{N} \to \mathbb{N}$), von denen eine das Band während der Berechnung verändert, die andere aber nicht. Das bedeutet, dass die betreffende Eigenschaft nicht extensional, sondern intensional ist. Der Satz von Rice ist daher hier nicht anwendbar. (Das Problem ist aber dennoch unentscheidbar, wie man durch eine Reduktion des Halteproblems zeigen könnte.)

c) Der Satz von Rice ist nicht anwendbar, da die Funktions-Eigenschaft μ-rekursiv zu sein trivial ist: Alle Turing-berechenbaren Funktionen sind μ-rekursiv (siehe Folien 56 bzw. 54 von Teil 2). Die Antwort auf das Entscheidungsproblem lautet also 'ja' für jede vorgelegte TM und ist somit (trivial) entscheidbar.

d) Die (Funktions-)Eigenschaft primitiv-rekursiv zu sein ist nicht-trivial: Viele, aber nicht alle Turing-berechenbaren Funktionen sind primitiv-rekursiv. Der Satz von Rice ist somit anwendbar und es folgt, dass das Problem unentscheidbar ist.

e) Aus nur einem Element zu bestehen ist offensichtlich eine nicht-triviale (Sprach-)Eigenschaft. Es folgt aus dem Satz von Rice, dass das Problem unentscheidbar ist.

f) Dieses Problem ist trivial: Für jede als Turingmaschine vorgelegte Sprache lautet die Antwort *per definitionem* 'ja'. Der Satz von Rice ist also nicht anwendbar.

Aufgabe 2.6
Gegeben sei folgende (deterministische) Turingmaschine M (siehe Foliensatz 3, Folien 11 ff):

$$M = (\{q_0, q_1, q_2, q_3, q_f\}, \{a, b, c\}, \{A, Z_0, B\}, \delta, q_0, \{Z_0 Z_1, Z_2\}, B, \{q_f\})$$

wobei die Übergangsfunktion δ folgendermaßen definiert ist:

$1: \quad \delta(q_0, a, B) = (q_1, A, S, R)$
$2: \quad \delta(q_1, a, B) = (q_0, A, R, R)$
$3: \quad \delta(q_0, b, B) = (q_2, A, R, R)$
$4: \quad \delta(q_2, b, B) = (q_2, A, R, R)$
$5: \quad \delta(q_2, c, B) = (q_3, B, S, L)$
$6: \quad \delta(q_3, c, A) = (q_3, B, R, L)$
$7: \quad \delta(q_3, Z_2, Z_0) = (q_f, Z_0, S, R)$

a) Beschreiben Sie kurz die Arbeitsweise dieser Maschine.

b) Geben Sie $L = L(M)$ (also die Sprache, die von M akzeptiert wird) an.

c) Handelt es sich bei M um einen LBA, Kellerautomaten und/oder EA? Begründen Sie Ihre Antwort.

d) Wo in der Chomsky-Hierarchie ordnen Sie die Sprache L ein? Begründen Sie Ihre Antwort.

Lösung 2.6

a) $1, 2$: Für jedes eingelesene Symbol a werden zwei Symbole A auf das Arbeitsband geschrieben.

$3, 4$: nun wird für jedes eingelesene Symbol b ein Symbol A auf das Arbeitsband geschrieben.

$5, 6$: Für jedes eingelesene Symbol c wird ein Symbol A auf dem Arbeitsband gelöscht.

7 : wird gleichzeitig mit dem Ende der Eingabe (dem Erreichen von Z_2 auf dem Eingabeband) das linke Begrenzungssymbol Z_0 auf dem Arbeitsband erreicht, geht M in den (einzigen) Enzustand q_f über und akzeptiert somit die Eingabe.

b) $L_2 = \{a^n b^k c^{2n+k} \mid n \geq 0, k \geq 1\}$

c) M ist ein LBA, da auf dem Arbeitsband während der Analyse eines Wortes w weniger als $|w|$ Zellen verwendet werden. M erfüllt darüberhinaus die Kellerautomatenbedingung, da zu jedem Zeitpunkt immer nur das "oberste" Symbol auf dem Arbeitsband gelesen wird.

d) Die Sprache L ist jedenfalls kontextfrei, da sie von einem Kellerautomaten akzeptiert wird. Damit ist L entsprechend der Chomsky Hierarchie natürlich auch kontextsensitiv und rekursiv aufzählbar. L ist aber sicher nicht regulär, was mit dem Pumping Lemma gezeigt werden kann, z.B. mt dem Wort $w = a^m c^{2m}$ (Vergleichen Sie dazu auch Aufgabe 2.8 a))

Aufgabe 2.7
Sind folgende Aussagen korrekt? Begründen Sie jeweils Ihre Antwort.

a) Ist eine Sprache L nicht-regulär, so ist auch $\overline{L}$ (das Komplement von L) nicht-regulär.

b) Sind die beiden Sprachen A und B nicht regulär, so ist auch deren Vereinigung $A \cup B$ nicht regulär.

c) Ist die Sprache L kontextfrei, so gilt dies auch für jede Teilmenge von L.

d) Es gibt einen Homomorphismus, der die Sprache $\{0^n 1^{2n} \mid n \geq 0\}$ auf die Sprache $\{a^{123n} \mid n \geq 0\}$ abbildet.

e) Die von der Grammatik $G = (\{S, A\}, \{a\}, \{S \to A \mid a, A \to a\}, S)$ erzeugte Sprache ist (inhärent) mehrdeutig.

f) Es gibt Sprachen L, für die gilt: $L^* = L^+$.

Lösung 2.7

a) **Richtig**. Reguläre Sprachen sind unter Komplement abgeschlossen. Wenn L regulär wäre, so müsste auch das Komplement von L regulär sein.

b) **Falsch.** Hier ein Gegenbeispiel: Sei A eine nicht reguläre Sprache und $B = \overline{A}$, also das Komplement von A. Dann ist auch B nicht regulär (siehe obigen Unterpunkt a)). Dann gilt aber $A \cup B = \Sigma^*$, was sehr wohl regulär ist.

c) **Falsch**. Ein Gegenbeispiel: Sei $A = \{a, b, c\}^*$. A ist regulär und damit auch kontextfrei. Sei weiters $B = \{a^n b^n c^n \mid n \geq 0\}$, dann gilt $B \subseteq A$. B ist aber bekannterweise nicht kontextfrei (siehe z.B. Foliensatz 3, Folie 111 ff).

d) **Richtig**. Es existiert z.B. folgender Homomorphismus $h : \{0,1\}^* \longrightarrow \{a\}^*$ mit $h(0) = a^{123}, \quad h(1) = \varepsilon$.

e) **Falsch**. Zwar ist die Grammatik G mehrdeutig, da es für das Wort a zwei verschiedene Linksableitungen gibt: $S \Rightarrow A \Rightarrow a$ bzw. $S \Rightarrow a$.

Für die von der mehrdeutigen Grammatik G erzeugte Sprache $L(G) = \{a\}$ existiert aber z.B. folgende eindeutige Grammatik: $G = (\{S\}, \{a\}, \{S \to a\}, S)$

f) **Richtig**. Es gilt z.B. $\{\varepsilon\}^* = \{\varepsilon\}^+ = \{\varepsilon\}$.

Aufgabe 2.8

Die unten gegebenen Sprachen A und B sind nicht regulär, was mit dem Pumping Lemma für reguläre Sprachen gezeigt werden soll. Dabei haben sich jedoch Fehler in der Argumentation eingeschlichen. Beschreiben Sie diese Fehler, und verbessern Sie die Beweise entsprechend.

a) Sei $A = \{uc^n \mid u \in \{a, b\}^*, n = 2|u|_a + |u|_b\}$, wobei $\Sigma = \{a, b, c\}$. (*Hinweis*: $|u|_a$ bezeichnet die Anzahl der Symbole a in u.)

Wir beweisen mit Hilfe des Pumping Lemmas für reguläre Sprachen, dass L nicht regulär ist:

Beweis indirekt. Angenommen, A ist regulär. Sei dann m die Konstante aus dem Pumping Lemma und $w = a^{2m}c^m$. Dann gilt $w \in L$ und $|w| = 3m$, also $|w| > m$.

Wir teilen nun w in xyz so auf, dass $|xy| \leq m$ und $|y| > 0$. Nachdem $|xy| \leq m$ und $w = a^{2m}c^m$, kann xy nur aus (mindestens 1 bis höchstens m) Symbolen a am Wortanfang bestehen.

Nach dem Pumping Lemma muss nun $xy^iz \in L$ für alle $i \geq 0$ gelten.

Bei $i = 0$ müsste (wenn L regulär wäre) auch $xy^0z = a^{2m-|y|}c^m$ aus L sein. Dies ist hier aber nicht der Fall, da nun mindestens ein Symbol a aus der ersteh Worthälfte fehlt, und damit die Anzahl der Symbole a in w nicht mehr doppelt so gross ist wie die Anzahl der Symbole c. Dementsprechend ist das Wort xy^0z nicht in der Sprache A! Wir haben also einen Widerspruch gefunden, A kann somit keine reguläre Sprache sein.

b) Sei $B = \{uu^r \mid u \in \{a, b, c\}^*\}$. (Hinweis: u^r bezeichnet das Spiegelbild von u.)

Wir beweisen mit Hilfe des Pumping Lemmas für reguläre Sprachen, dass L nicht regulär ist:

Beweis indirekt. Angenommen, L ist regulär. Sei dann m die Konstante aus dem Pumping Lemma und $w = a^m a^m$. Dann gilt $w \in L$ und $|w| = 2m \geq m$.

w kann nun in xyz so aufgeteilt werden, dass $|xy| \leq m$ und $|y| > 0$. Nachdem $|xy| \leq m$ und $w = a^m a^m$, kann xy nur aus Symbolen a der ersten Worthälfte bestehen.

Nach dem Pumping Lemma muss aber $xy^i z \in L$ für alle $i \geq 0$ gelten.

Für $i = 2$ müsste also auch $xy^2 z = a^{m+|y|} a^m$ aus L sein, was aber nicht der Fall ist, da nun in der ersten Worthälfte mehr Symbole a sind als in der zweiten. Wir haben einen Widerspruch gefunden, B kann somit keine reguläre Sprache sein.

Lösung 2.8

a) Das Problem in diesem vermeintlichen Beweis ist die Wortwahl, denn das gewählte Wort w ist nicht in der Sprache A. Damit führt sich der gesamte Beweis *ad absurdum*.

   Eine mögliche korrekte Wortwahl ist z.B. $w = a^m c^{2m}$. Damit wäre dann die "aufgepumpte" Version von w also $xy^0 z = a^{m-|y|} c^{2m}$. Dies würde bedeuten, dass mindestens ein Symbol a am Wortanfang wegfällt, womit die Anzahl der Symbole c nicht mehr doppelt so groß ist wie jene der Symbole a, und dementsprechend nicht in A liegen kann.

b) Auch hier ist die Wortwahl problematisch. Zwar liegt das gewählte Wort $w = a^m a^m$ in der Sprache B, ist aber für den Beweis nicht zielführend. Denn es gibt sehr wohl Zerlegungen von w, für die $xy^2 z = a^{m+|y|} a^m$ wiederum in B liegt. Ist nämlich $|y|$ gerade, so resultiert $xy^2 z$ in einem Wort mit einer geraden Anzahl von Symbolen a, welches wiederum die Form uu^r hat, und somit in L liegt.

   Eine bessere Wortwahl wäre hier z.B. $w = a^m cca^m$. Dann ist $|w| = 2m + 2$ und $xy^2 z = a^{m+|y|} cca^m$, was auf den gewünschten Widerspruch führt.

Aufgabe 2.9

Beweisen Sie mit Hilfe des Pumping Lemmas für reguläre Sprachen, dass die folgenden Sprachen L nicht regulär sind. (Das Alphabet ist dabei jeweils $\Sigma = \{a, b, c\}$.)

Geben Sie darüberhinaus auch an, ob es sich bei L jeweils um eine kontextfreie Sprache handelt, und begründen Sie Ihre Antwort.

a) $L = \{uc^n \mid u \in \{a, b\}^*, |u| > n\}$

b) $L = \{vcxcv^r \mid v, x \in \{a, b\}^*, |v| = |x|\}$

Lösung 2.9

a) Beweis indirekt. Angenommen, L ist regulär. Sei dann m die Konstante aus dem Pumping Lemma und das gewählte Wort z.B.

$$w = b^{m+1} c^m.$$

   Dann gilt $w \in L$ und $|w| = 2m + 1 > m$.

   Wir teilen nun w in xyz so auf, dass $|xy| \leq m$ und $|y| > 0$. Nachdem $|xy| \leq m$ und $w = b^{m+1} c^m$, kann xy nur aus (mindestens einem bis höchstens m) Symbolen b des ersten Wortteils bestehen.

   Nach dem Pumping Lemma muss aber $xy^i z \in L$ für alle $i \geq 0$ gelten.

   Wählen wir nun $i = 0$, so müsste auch $xy^0 z = b^{m+1-|y|} c^m$ aus L sein, was aber nicht der Fall ist! Da y aus mindestens einem Symbol besteht, fällt mit $i = 0$ auch mindestens ein Symbol b am Wortanfang weg. Damit sind im resultierenden Wort aber nicht mehr strikt mehr Symbole b als c. Wir erhalten also ein Wort, welches nicht in L liegt. Widerspruch! L kann somit keine reguläre Sprache sein.

   L ist **kontextfrei**, was man z.B. in ähnlicher Weise wie in Aufgabe 2.11 b) zeigen kann.

b) Beweis indirekt. Angenommen, L ist regulär. Sei dann m die Konstante aus dem Pumping Lemma und das gewählte Wort z.B.

$$w = a^m ca^m ca^m.$$

Dann gilt $w \in L$ und $|w| = 3m + 2$, also jedenfalls $|w| \geq m$.

Nun kann w in xyz so aufgeteilt werden, dass $|xy| \leq m$ und $|y| > 0$. Nachdem $|xy| \leq m$ und $w = a^m ca^m ca^m$, kann xy nur aus (mindestens einem bis maximal m) Symbolen a am Wortanfang bestehen.

Nach dem Pumping Lemma muss dann $xy^iz \in L$ für alle $i \geq 0$ gelten.

Für z.B. $i = 2$ müsste nun aber auch $xy^iz = a^{m-|y|}ca^m ca^m$ ein Wort in L sein. Dies ist aber nicht der Fall, da nun vor dem ersten Symbol c mindestens ein Symbol a mehr vorkommt, als in den nachfolgenden a-Blöcken, und das resultierende Wort damit das nicht in L sein kann.

Widerspruch! L kann keine reguläre Sprache sein.

L ist **nicht kontextfrei**. Ein Kellerautomat könnte sehr wohl feststellen, ob in einem Wort $vcxcv^r$ mit $v, x \in \{a, b\}^*, |v| = |x|$ die Teilwörter v und x gleich lange sind. (z.B. könnte für jedes eingelesene Symbol des Teilwortes v ein Symbol auf das Arbeitsband (bzw. in den Keller) geschrieben werden, welches dann beim Einlesen eines Symbols des Wortteils x wieder gelöscht wird.) Ebenso könnte ein Kellerautomat feststellen, ob es sich bei einem Wort um ein Palindrom handelt. Um beides zu überprüfen reicht ein Stack aber nicht aus.

In der Tat ist L kontextsensitiv und könnte von einem linear beschränkten Automaten akzeptiert werden (ähnlich dem Beispiel auf Folie 14).

Aufgabe 2.10

Geben Sie jeweils eine kontextfreie Grammatik an, welche die folgenden Sprachen erzeugt, sowie eine Linksableitung und einen Ableitungsbaum für ein von Ihnen gewähltes Wort $w \in L$ mit $|w| \geq 5$.

a) $L = \{a^{2k}b^kc^nd^{3n} \mid k, n \geq 0\}$

b) $L = \{a^{4k}b^{2l}c^m \mid k, l, m \geq 0\} \cap \{a^{2m}b^{4k}c^m \mid m, k \geq 0\}$ (*Hinweis*: Bestimmen Sie zunächst L.)

Lösung 2.10

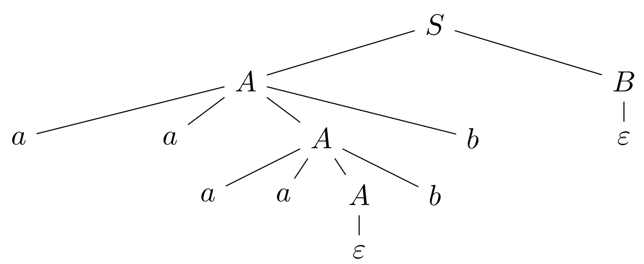
a) L wird z.B. von folgender kontextfreien Grammatik erzeugt:

$G = (\{S, A, B\}, \{a, b, c, d\}, \{S \rightarrow AB, \quad A \rightarrow a^2Ab \mid \varepsilon, \quad B \rightarrow cBd^3 \mid \varepsilon\}, S\})$

Linksableitung für $w = aaaabb = a^4b^2$:

$S \Rightarrow AB \Rightarrow aaAbB \Rightarrow aaaaAbbB \Rightarrow aaaabbB \Rightarrow a^4b^2$

Ableitungsbaum für $w = a^4b^2$:



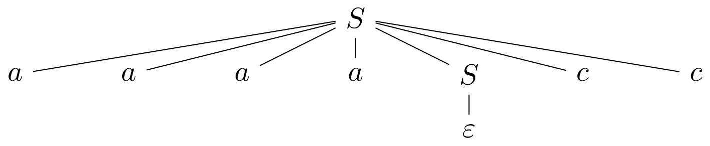
b) Wir überlegen zunächst, dass $L = \{a^{4n}b^{4k}c^{2n} \mid k, n \geq 0\}$.

Eine kontextfreie Grammatik für L ist z.B.:

$G = (\{S, B\}, \{a, b, c\}, \{S \to a^4Sc^2 \mid b^4B \mid \varepsilon, B \to b^4B \mid \varepsilon\}, S)$

Linksableitung für $w = aaaacc = a^4c^2$: $S \Rightarrow aaaaScc \Rightarrow aaaacc$

Ableitungsbaum für $w = a^4c^2$:



Aufgabe 2.11

Zeigen Sie die Kontextfreiheit der folgenden Sprachen L über $\Sigma = \{a, b, c\}$ mit Hilfe des Satzes von Chomsky-Schützenberger (indem Sie jeweils entsprechende Sprachen D_n und R sowie einen entsprechenden Homomorphismus h so angeben, dass gilt: $L = h(D_n \cap R)$).

a) $\{wc^n \mid w \in \{a, b\}^*, n = |w|_a + 2 * |w|_b\}$

b) $\{a^nb^mc^k \mid 0 \leq n + m \leq k\}$

Lösung 2.11

a) L ist kontextfrei, da $L = h(D_2 \cap R)$, wobei

$$R = \{(, [\}^*\{],)\}^*$$

und

$$h : \{(,), [,]\}^* \longrightarrow \{a, b, c\}^* \quad \text{mit} \quad h(() = a, \quad h([) = b, \quad h()) = c, \quad h(]) = c^2$$

b) L ist kontextfrei, da $L = h(D_3 \cap R)$, wobei

$$R = \{(\}^*\{[\}^*\{]\}^*\{)\}^*\{\langle, \rangle\}^*$$

und

$$h : \{(,), [,], \langle, \rangle\}^* \longrightarrow \{a, b, c\}^* \quad \text{mit} \quad h(() = a, \quad h([) = b, \quad h()) = h(]) = h(\langle) = c, \quad h(\rangle) = \varepsilon$$

Aufgabe 2.12

Geben Sie für jede der folgenden Grammatiken an, ob diese regulär, kontextfrei, monoton, und/oder kontextsensitiv ist. Geben Sie weiters an, welche Sprache von der jeweils angegebenen Grammatik erzeugt wird, und ob diese regulär, kontextfrei, kontextsensitiv und/oder rekursiv aufzählbar ist. Begründen Sie Ihre Antwort.

Beispiel:

Sei $G_{BSP} = (\{S\}, \{a, b\}, \{S \to Sxx \mid x \in \{a, b\}\} \cup \{S \to \varepsilon\}, S)$.

G_{BSP} ist kontextfrei. Wegen z.B. der Produktion $S \to Saa$ ist G_{BSP} nicht regulär, und wegen $S \to \varepsilon$ nicht monoton und auch nicht kontextsensitiv, nachdem S auch auf der rechten Seite von Produktionen vorkommt.

$L(G_{BSP}) = \{aa, bb\}^*$ ist aber regulär, und damit auch kontextfrei, kontextsensitiv und rekursiv aufzählbar.

a) $G = (\{S, A, B\}, \{a, b, c\}, \{S \to aA \mid \varepsilon, \quad A \to bB, \quad B \to cS\}, S)$

b) $G = (\{S, B\}, \{a, b\}, \{S \to aB \mid \varepsilon, B \to Sb\}, S)$

c) $G = (\{S\}, \{a, b, c\}, \{S \to bacS \mid cab\} \cup \{xy \to yx \mid x, y \in \{a, b, c\}\}, S)$

Lösung 2.12

a) G ist regulär und somit auch kontextfrei. Wegen $S \to \varepsilon$ ist G aber weder monoton noch kontextsensitiv, da S auch auf der rechten Seite einer Produktion ($B \to cS$) vorkommt.

   $L(G) = \{abc\}^*$ ist eine reguläre Sprache, und somit auch kontextfrei, kontextsensitiv und rekursiv aufzählbar.

b) G_4 ist kontextfrei, aber weder kontextsensitiv noch monoton, da S auch auf der rechten Seite einer Produktion ($B \to Sb$) vorkommt. Wegen $B \to Sb$ ist G auch nicht regulär. (Beachten Sie, dass hier auf der rechten Seite der Produktionen das Terminalsymbol links und rechts vom Nonterminalsymbol vorkommt, was in regulären Grammatiken nicht zulässig ist.)

   $L(G) = \{a^n b^n \mid n \geq 0\}$ ist ebenfalls kontextfrei, kontextsensitiv und auch rekursiv aufzählbar (aber sicher nicht regulär).

c) G ist wegen z.B. $ab \to ba$ weder regulär noch kontextfrei noch kontextsensitiv. G ist allerdings monoton, da für alle Produktionen $\alpha \to \beta$ in G gilt, dass $|\alpha| \leq |\beta|$.

   $L(G) = \{w \in \{a, b, c\}^+ \mid |w|_a = |w|_b = |w|_c\}$ Diese Sprache ist weder regulär noch kontextfrei. Nachdem G aber monoton ist, ist $L(G)$ entsprechend kontextsensitiv und damit, aufgrund der Chomsky Hierarchie, auch rekursiv aufzählbar.