A_1	A_2	$A_1 \dot{\vee} A_2$	$A_1 \wedge A_2$	$\neg A_1$
W	W	F	W	F
W	F	W	F	F
F	W	W	F	W
F	F	F	F	W

Fig. 1

Lösungen zu ausgewählten Aufgaben

(3.2)

1. Alpha Strich
2. Alpha Dach ist Element von Gamma
3. A vereinigt B ist die leere Menge
4. Omega zwei Strich ihh ist wesentlich größer als Theta quer

Denkt euch mehr Ausdrücke und ihre verbalen Entsprechungen aus und fragt eure Mitstudis, ob sie das auch so gesagt hätten.

(4.1)

(cf. Fig. 1)

(4.2)

(a)

1. $\neg R$
2. $R \wedge N$
3. $R \vee F$
4. $R \to \neg F$
5. $F \leftrightarrow \neg R$
6. $\neg R \to \neg N$

(b) Die Anworten müsst ihr wohl oder übel aus den Wahrheitstafeln ablesen, nachdem ihr euch darüber klar geworden seid, wie es denn mit den Wahrheitswerten von F, N und R so aussieht. Wenn für euch F wahr ist, solltet ihr jetzt anhalten, bevor ihr irgendwen übermangelt.

Interessant ist allerdings, dass die Aussage (3) wahr ist, egal, wie ihr R und N belegt. Aussagen dieser Art werdet ihr in der Logik als Tautologie kennen lernen – sie spielen, wie man sich vielleicht vorstellen kann, eine große Rolle, wenn man sich Gedanken über Schlussprozesse macht.

Wenn ihr meint, dass auch Aussage (1) immer wahr ist, liegt ihr vielleicht real richtig (obwohl ihr natürlich unter Dach Rad fahren könntet), aus Sicht der (Aussagen-) Logik aber falsch. Eine Tautologie wird eine Aussage erst, wenn sie für *alle* Belegungen wahr bleibt, nicht nur für irgendwelche, die ihr gerade für plausibel haltet.

(4.3)

A	B	C	$A \vee (B \wedge C)$	$(A \vee B) \wedge (A \vee C)$
W	W	W	W	W
W	W	F	W	W
W	F	W	W	W
W	F	F	W	W
F	W	W	W	W
F	W	F	F	F
F	F	W	F	F
F	F	F	F	F

Diese Wahrheitstafel bedeutet nichts anderes, als dass

$$A \vee (B \wedge C) \leftrightarrow (A \vee B) \wedge (A \vee C)$$

eine Tautologie ist, die beiden Aussagen also äquivalent sind. Das ist so eine Art Distributivitätsgesetz für logische Operatoren.

(5.1)

Im Sommersemester 2004 könnte das so aussehen:

$$\{\textit{Kstr2 SR004}, \textit{KStr2 R107}, \textit{NUni HS5}\}.$$

(5.2)

Aus der Mitgliedschaft von x folgt die von axb, aus dessen Mitgliedschaft wieder die von $aaxbb$ und so fort. Wir haben also die Menge $\{x, axb, aaxbb, \ldots\}$, kurz als $\{a^i x b^i \mid i \in \mathbb{N}\}$ zu schreiben.

(5.3)

x ist eine gerade nichtnegative Zahl, also 0, 2, 4, 6 und so fort. In der Menge sind gerade immer diese Werte halbiert, also 0, 1, 2 und so fort. Mithin ist die gesuchte Menge einfach die der Natürlichen Zahlen.

(6.1)

$C = \{1, 2, 3, 4, a, b\}$ – a ist natürlich nicht „zweimal" in C enthalten. Mengen enthalten ein Element oder sie enthalten es nicht. Ihr hättet natürlich auch $C = \{1, 2, 3, a, b, 4\}$ oder $C = \{2, 4, a, 1, 4, b\}$ schreiben dürfen: Mengenelemente sind nicht angeordnet.

$A \cap B = \{a\}$

$\mathcal{P}(B) = \{\emptyset, \{4\}, \{a\}, \{4, a\}\}$ – nachzählen: Das Ding hat vier Elemente, und das war gemäß unserer $2^{|B|}$-Formel auch zu erwarten. In einem ruhigen Moment könnt ihr ja mal die $2^5 = 32$ Elemente von $\mathcal{P}(A)$ ausrechnen.

$A \setminus B = \{1, 2, 3, b\}$

$B \setminus A = \{4\}$ – beachtet, dass $\setminus$ damit ein nichtkommutativer Operator ist; bei der Addition ist ja $a + b = b + a$ („Die Addition kommutiert"), und für normale Zahlen ist auch $a \cdot b = b \cdot a$. Die Multiplikation ist allerdings schon bei den Matrizen nicht mehr kommutativ. Glaubt es oder nicht, aber dieses einfache Fakt hat so profunde Konsequenzen wie die Heisenberg'sche Unschärferelation.

(6.2)

Aber natürlich. Einmal hatten wir ja schon gezeigt, dass die leere Menge Teilmenge *jeder* Menge ist, also insbesondere auch der leeren Menge selbst. Zweimal gilt ohnehin für jede Menge $A \subseteq A$ (könnt ihr das beweisen?). Schließlich muss die leere Menge auch mindestens eine Teilmenge haben, denn es gilt $|\mathcal{P}(\emptyset)| = 2^{|\emptyset|} = 2^0 = 1$.

(6.3)

Die Behauptung ist $\emptyset \subset \emptyset$. Wenn wir die Definition der echten Teilmenge einsetzen, ist das äquivalent zu

$$(x \in \emptyset \Rightarrow x \in \emptyset) \wedge (\exists x : x \in \emptyset \wedge x \notin \emptyset).$$

Dabei haben wir mit dem ersten Teil der Konjunktion (des $\wedge$-Ausdrucks) keine Probleme, dass der wahr ist, haben wir schon gesehen. Allein der zweite macht Schwierigkeiten. Hier wird die Existenz eines Elements in der leeren Menge gefordert, und wir wissen schon, dass es das nicht gibt. Selbst wenn wir hier aber über eine nichtleere Menge reden würden, wäre der zweite Teil immer noch falsch, denn das Prädikat, das das x im zweiten Teil des Ausdrucks erfüllen soll, ist logisch Äquivalent zu $A \wedge \neg A$. Wenn ihr die Wahrheitstafel für diesen Ausdruck hinschreibt, seht ihr, dass der Ausdruck immer falsch ist. Dies ist ein Beispiel für ein Oxymoron, quasi das Gegenstück zu einer Tautologie.

Es gibt also keine Menge, die echte Teilmenge ihrer selbst ist.

(6.4)

Wir gehen Schritt für Schritt vor:

$$\begin{aligned}
A \cup (B \cap C) &= \{x \mid x \in A \vee x \in B \cap C\} && \text{Def. Vereinigung}\\
&= \{x \mid x \in A \vee (x \in B \wedge x \in C)\} && \text{Def. Schnitt}\\
&= \{x \mid (x \in A \vee x \in B) \wedge (x \in A \vee x \in C)\} && \text{„Distributivgesetz“}\\
&= \{x \mid x \in A \cup B \wedge x \in A \cup C\} && \text{Def. Vereinigung}\\
&= (A \cup B) \cap (A \cup C) && \text{Def. Schnitt}
\end{aligned}$$

(7.1)

$A \times B = \{\langle 1,4\rangle, \langle 2,4\rangle, \langle 3,4\rangle, \langle a,4\rangle, \langle b,4\rangle, \langle 1,a\rangle, \langle 2,a\rangle, \langle 3,a\rangle, \langle a,a\rangle, \langle b,a\rangle\}$

$C \setminus (A \cup B) = C \setminus \{a\} = \{1,2,3,4,b\}$

$(C \setminus A) \cup (C \setminus B) = \{4\} \cup \{1,2,3,b\} = \{1,2,3,4,b\}$ – es ist natürlich kein Zufall, dass das gleich dem letzten Ergebnis ist: Man kann sich hier viel Arbeit sparen, wenn man sieht, dass man mit DeMorgan argumentieren kann. Seht ihr, warum?

(8.2)

Für $n = 1$ ist $\prod_{i=1}^{n}(-1)$ einfach -1, für $n = 2$ ist es $(-1) * (-1) = 1$, beim nächsten Mal kommt wieder ein -1 dazu, es ist also wieder -1 und so fort. Wir haben also

$$\prod_{i=1}^{n}(-1) = (-1)^n = \begin{cases} -1 & n \text{ ungerade} \\ 1 & n \text{ gerade} \end{cases}$$

Eine strenge Art, diese Behauptung nachzuweisen, lernt ihr auf der nächsten Folie kennen.

Bei der Summe hat man für $n = 1$ einfach $(-1)^1 = -1$, für $n = 2$ dann $(-1) + 1 = 0$, für $n = 3$ dann $0 + (-1) = -1$ und so fort. Wir haben also

$$\sum_{i=1}^{n}(-1)^i = \begin{cases} -1 & n \text{ ungerade} \\ 0 & n \text{ gerade} \end{cases}$$

Auch das lässt sich mit den Instrumenten auf der nächsten Folie streng zeigen.

(8.3)

$$\begin{aligned}
\sum_{i=1}^{n}(a_i - a_{i-1}) &= \sum_{i=1}^{n} a_i - \sum_{i=1}^{n} a_{i-1}\\
&= \sum_{i=1}^{n} a_i - \sum i = 0^{n-1} a_i\\
&= \sum_{i=1}^{n-1} a_i + a_n - a_0 - \sum_{i=1}^{n-1} a_i\\
&= a_n - a_0
\end{aligned}$$

– analog reduziert man das Produkt auf a_n / a_0. Summen dieses Typs bezeichnet man gerne als Teleskopsumme, weil sie so schon zusammenschnurren.

(8.4)

$$\bigcap_{i=1}^{3} M_i = \{1\} \cap \{1,2\} \cap \{1,2,3\} = \{1\}$$

$$\bigcup_{i=1}^{3} M_i = \{1\} \cup \{1,2\} \cup \{1,2,3\} = \{1,2,3\}$$

(9.1)

$$\succ = \{ \langle a,b\rangle, \langle a,c\rangle, \langle a,d\rangle, \langle a,e\rangle, \langle a,f\rangle, \langle a,g\rangle, \langle b,c\rangle, \\ \langle b,d\rangle, \langle b,e\rangle, \langle b,f\rangle, \langle b,g\rangle, \langle d,e\rangle, \langle d,f\rangle, \langle d,g\rangle, \langle e,f\rangle\}$$

Wie ihr seht, macht die Relation bei weitem nicht für alle Elemente der Menge eine Aussage über ihr Verhältnis. So gilt etwa weder $e \succ c$ noch $c \succ e$. Dies – dass nämlich eine Relation keine Totalordnung vorgibt – ist der typische Fall für Relationen aus der realen Welt, was nicht nur Uni- und sonstige Ranker nur zu gerne vergessen.

(9.2)

Reflexiv ist die Relation schon mal nicht, es gibt kein einziges $\langle x,x\rangle$ in der Relation, geschweige denn eins für jedes Element der Grundmenge. Das ist vernünftig: Ein Begriff wird in der Regel nicht sein eigener Oberbegriff sein.

Symmetrisch ist die Relation auch nicht, weil z.B. schon für $\langle a,b\rangle$ das zugehörige $\langle b,a\rangle$ fehlt. Auch dies ist vernünftig, denn wenn a der Oberbegriff zu b ist, ist b sicher nicht allgemeiner als a und kommt also nicht als Oberbegriff in Frage.

Transitiv ist die Relation. Um das zu prüfen, geht man alle Elemente durch – wir fangen mit $\langle a,b\rangle$ an und prüfen für alle Tupel $\langle b,x\rangle$, ob $\langle a,x\rangle$ auch in der Relation ist. Wenn uns kein Tupel fehlt, ist die Relation transitiv. Auch hier ist die Transtivität vernünftig: Wenn a Oberbegriff von b ist, und b Oberbegriff von c ist natürlich erst recht a Oberbegriff von c.

(10.1)

Diese Aufgabe mag komplizierter geklungen haben, als sie ist.

Reflexivität: Zu zeigen: Für alle $x \in \mathbb{N}$ gilt $x = x \bmod n$. Das ist trivial – wenn schon die Zahlen selbst gleich sind, wird sich beim Rest bei der Division durch n auch kein Unterschied auftun.

Symmetrie: Zu zeigen: Wenn $x = y \bmod n$, ist auch $y = x \bmod n$. Also: $x = y \bmod n$ bedeutet, dass es ein a gibt, so dass $x = an + y$, was sich sofort zu $y = (-a)n + x$ und mithin zu $y = x \bmod n$ umformen lässt.

Transitivität: Zu zeigen: Wenn $x = y \bmod n$ und $y = z \bmod n$, dann ist auch $x = z \bmod n$. Aus den beiden Voraussetzungen folgt $x = a_1 n + y$ und $y = a_2 n + z$, was durch Substitution $x = a_1 n + a_2 n + z$ ergibt. Das bedeutet aber durch Ausklammern $x = bn + z$ und also wie gewünscht $x = z \bmod n$ mit $b = a_1 + a_2$.

(11.1)

1. f_1 ist surjektiv, injektiv und damit bijektiv. Das lässt sich direkt durch Einsetzen der Form in die Definitionsgleichungen sehen.
2. f_2 ist nicht surjektiv. So ist z.B. $n = -1 \in \mathbb{R}$, aber es gibt kein $m \in \mathbb{R}$, so dass $m^2 = -1$. Es ist auch nicht injektiv, weil z.B. $m_1 = 2$ und $m_2 = -2$ beide in $\mathbb{R}$ sind. Es ist also $f(m_1) = f(m_2)$, aber $m_1 \neq m_2$, was bei injektiven Funktionen verboten ist.
3. f_3 ist immerhin surjektiv, weil ich zu jeder nichtnegativen Zahl eine Wurzel finde, aber mit dem gleichen Argument wie eben nicht injektiv.
4. f_4 ist weiter surjektiv, diesmal aber auch injektiv, weil in $\mathbb{R}^+$ $x^2 = y^2 \Rightarrow x = y$ gilt und damit bijektiv.
5. f_5 ist ebenfalls bijektiv, was sich analog zu f_4 aus den Eigenschaften der Exponentialfunktion sehen lässt.

(11.2)

Um zu zeigen, dass f injektiv ist, müssen wir aus $f(x) = f(y)$ schlicht $x = y$ schließen.

Es muss einer der drei im Tipp genannten Fälle gelten. Kann $x < y$ sein? Nein, denn dann müsste im Gegensatz zur Annahme auch $f(x) < f(y)$ sein. Kann $x > y$ sein? Nein, denn durch Vertauschen der Rollen von x und y in der Definition wäre dann auch $f(x) > f(y)$. Im Fall $x = y$ schließlich ist trivial $f(x) = f(y)$; da keiner der anderen Fälle möglich ist, haben wir $x = y$ nachgewiesen, streng monotone Funktionen sind also injektiv.

Was passiert bei monotonen Funktionen, für die nur $x < y \Rightarrow f(x) \leq f(y)$ gilt?

(12.1)

Die Mengen haben jeweils gleiche Kardinalität. Die Bijektionen sehen so aus:

$$f_1\colon x \mapsto -x$$
$$f_2\colon x \mapsto 2x$$
$$f_3\colon x \mapsto 2x$$

Es ist zugegebenermaßen etwas eigenartig, dass es genauso viele gerade Zahlen wie natürliche Zahlen gibt, denn immerhin wurde ja bei den geraden Zahlen jede zweite Zahl rausgelassen. Aber f_3 ist nun mal bijektiv. Hintergrund ist, dass „Unendlich" ein komischer Begriff ist, $\aleph_0$ eben keine richtige, sondern eine „transfinite" Zahl ist. Die Arithmetik auf diesen transfiniten Zahlen läuft etwas ungewöhnlich, man könnte mit gutem Recht sagen, dass $\aleph_0/2 = \aleph_0$ ist.

Übrigens haben nicht alle unendlichen Mengen die Kardinalität $\aleph_0$ – die reellen Zahlen beispielsweise, die z.B. als beliebige (unendliche) Folgen von Ziffern zwischen 0 und 9 dargestellt werden können, haben eine höhere Kardinalität, es gibt keine Bijektion zwischen den natürlichen und den reellen Zahlen. Der Beweis dafür ist allerdings etwas aufwändig.

(12.2)

Fall 1: n gerade

Induktionsanfang ist die kleinste gerade Zahl, die für uns in Frage kommt, nämlich $n = 2$. Wir haben $(-1) \cdot (-1) = 1$, die Behauptung ist also für $n = 2$ richtig.

Induktionsannahme: Wenn die Behauptung für ein gerades n wahr ist, ist sie das auch für die nächste gerade Zahl, nämlich $n + 2$.

Induktionsschluss:

$$\begin{aligned}\prod_{i=0}^{n+2}(-1) &= (-1) \cdot (-1) \cdot \prod_{i=0}^{n}(-1) \\ &= 1 \cdot 1\end{aligned}$$

– beim letzten Gleichheitszeichen haben wir die Induktionsannahme werwendet.

Fall 2: n ungerade

Ach, das kriegt ihr selbst hin, oder?

(14.1)

Zu definieren ist, ob sich die Potenz auf ab oder nur auf b bezieht. Dies ist eine Frage der so genannten Präzedenz, also der gegenseitigen Bindungsstärken von Operatoren. Für Addition und Multiplikation haben wir „Punkt vor Strich", und die Potenz bindet wiederum stärker als „Punkt". Wenn wir die Verkettung als Addition auffassen und die Potenzbildung als Potenz lassen, würde sich also ergeben:

$$ab^2 = a \bullet b^2 = abb.$$

So wollen wir das hier auch halten und die Potenzierung eines ganzen Wortes durch Klammern markieren:

$$(ab)^2 = abab.$$

(14.2)

Es ist

$$A^* = \{\epsilon, 1, 2, 3, 11, 12, 13, 21, 22, 23, 31, 32, 33, \ldots\}$$

– die drei Pünktchen sind wichtig, die Sternmenge ist natürlich unendlich, und wir haben nur ihre kürzesten Element aufgezählt.

Demgegenüber ist

$$\mathcal{P}(A) = \{\emptyset, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}\}$$

– hier gibts keine Pünktchen, Potenzmengen endlicher Mengen sind zwar typischerweise groß, aber endlich. Wenn ihr aus der Hüfte sagen könnt, warum hier kein $\{3, 1\}$ oder $\{3, 3\}$ steht, seid ihr dem Grundverständnis von Mengen schon erheblich näher gekommen.

Beachtet, dass $\mathcal{P}(A)$ an sich überhaupt nichts mit A^* zu tun hat – diese Aufgabe entsprang der Beobachtung, dass viele Studierende Probleme hatten, die beiden Begriff auseinanderzuhalten.

Hier ist noch ein (nicht allzu effizientes) Programm zur Berechnung von Sternmengen. Von der Kommandozeile aus aufgerufen nimmt es im ersten Argument die Maximallänge der zu erzeugenden Wörter, alle folgenden Argumente definieren die Menge, auf die der Stern angewandt werden soll.

```
def computeKleeneStar(aSet, maxLen):
  """returns the star set for the elements of the iterable aSet with
  all words up to maxLen (exclusively).  The elements obviously
  have to support concatenation.  The strategy is a straight
  implementation of the recursive definition of the Kleene star.
  """
  if maxLen==0:
    return [""]
  else:
    shorterSet = computeKleeneStar(aSet, maxLen-1)
    return [a+w
      for a in aSet
        for w in shorterSet if len(w)==maxLen-1]+shorterSet

if __name__=="__main__":
  import sys
  starSet = computeKleeneStar(sys.argv[2:], int(sys.argv[1]))
  starSet.sort(lambda a,b: cmp(len(a), len(b)) or cmp(a, b))
  print starSet
```

(14.3)

Man kann sich das elementar überlegen. Für das erste Zeichen haben wir N Möglichkeiten, für das zweite wieder; es gibt also $N \cdot N$ Wörter der Länge 2. An der dritten Stelle können wieder N verschiedene Zeichen stehen, so dass wir dann N^3 Wörter haben usf.

Es gibt also N^n Wörter der Länge n. Menschen, die schon mal was mit Kombinatorik zu tun hatten, erkennen hier unschwer das Urnenmodell „Ziehen mit Zurücklegen mit Anordnung“.

Reality Check: Es gibt $N^0 = 1$ Wort der Länge Null, nämlich gerade unser ϵ.

(14.4)

Wir wissen, dass es N^k Wörter der Länge k gibt. Alles, was wir tun müssen, ist, die Beiträge aller k zwischen 0 und n zusammenzusummieren, $\sum_{k=0}^{n} N^k$. Summen dieses Typs lassen sich mit einem Trick berechnen, der so aussieht:

$$\begin{aligned}\sum_{k=0}^{n} N^k &= N \sum_{k=0}^{n} N^{k-1} = N \sum_{k=-1}^{n-1} N^k \\ &= N\left(\frac{1}{N} + \sum_{k=0}^{n} N^k - N^n\right) = 1 + N \sum_{k=0}^{n} N^k - N^{n+1}\end{aligned}$$

Diese Gleichung kann man jetzt nach der Summe auflösen und kommt auf

$$\sum_{k=0}^{n} = \frac{1 - N^{n+1}}{1 - N}.$$

Mithin hat die Sternmenge einer N-elementigen Menge $(1 - N^{n+1})/(1 - N)$ Elemente, die kürzer als n sind.

Prüft diese Formel mit eurem Ergebnis von oben. Beachtet, dass dies eine stark wachsende Funktion ist. In der Aufgabe oben ($N = 3$) wärt ihr, selbst wenn ihr ein Wort pro Sekunde schreiben könntet, über 24 Stunden beschäftigt gewesen, hättet ihr die Wörter bis zur Länge 11 schreiben wollen (das sind nämlich 88573 Stück), wolltet ihr bis Länge 16 kommen, wärt ihr fast ein Jahr beschäftigt gewesen, bis ihr die 21523360 Wörter hingeschrieben hättet.

Hat man ein Alphabet von 26 Zeichen, wird das alles noch viel schlimmer – hier gibts schon bei 321272407 Wörter bis zur Länge 7, für die ihr dann wohl gegen zehn Jahre brauchen würdet.

In diesen rapide wachsenden Zahlen liegt dann auch der Grund, warum Passwörter, die einfach aus Zeichen zusammengewürfelt werden, relativ sicher sind.

(17.1)

Also, zumindest fangen mal alle Wörter mit einem Backslash an. Für den Rest braucht man eine Fallunterscheidung – entweder, man hat ein Sonderzeichen, dann ist die Länge dieses Rests genau eins. Alternativ ist das Zeichen hinter dem Backslash ein „normaler“ Buchstabe; in diesem Fall kann die Kontrollsequenz beliebig lang sein.

(19.1)

$$S \overset{<}{\to} O \overset{b}{\to} T \overset{r}{\to} T \overset{>}{\to} C \overset{<}{\to} \text{DEA steht}$$

Der letzte Zustand des DEA war C, und das ist ja eigentlich ein Endzustand. Trotzdem akzeptiert der Automat *nicht*. Als Bedingung für die Annahme eines Wortes haben wir Endzustand *und* leere Eingabe gefordert. In unserem Fall sind wir, wenn die Eingabe leer ist, allerdings im „ungültigen“ Zustand „DEA steht“ und nicht mehr im Zustand C.

Wenn euch das komisch vorkommt, ergänzt den Automaten einfach um einen Zustand E (wie Error) und malt Pfeile von allen Zuständen zu diesem E, die mit allen Elementen von Σ, die nicht auf anderen Übergängen von den jeweiligen Zuständen weg sitzen. E selbst hat nur einen Übergang, nämlich auf E selbst, der für alle Elemente von Σ genommen wird; E ist natürlich kein Endzustand. Der resultierende Automat bleibt nie stehen, solange noch etwas in der Eingabe ist, verhält sich aber bezüglich Akzeptanz exakt wie unsere Automaten ohne Pannensicherung.

(20.1)

Die Skizze des Automaten spare ich mir, und auch die Umwandlung ist trivial, wenn man dem Rezept am Anfang dieses Abschnittes folgt: Ist nämlich $\delta($ *S*, long$) =$ *L*, so hat man einfach eine Regel *S* $\to$ long *L*. Das mag euch höchst langweilig vorkommen, aber gemacht haben muss mensch sowas eben doch mal.

Wenn ihr zu faul wart, um ein paar Wörter zu erzeugen, probiert es doch noch mal mit `unsigned long int a,b;` oder `int a;`. Macht diese Wörter irgendwie kaputt und seht zu, wie sie vom Automaten zurückgewiesen werden.

C-KennerInnen werden einwenden, dass das int auch unterbleiben kann, wenn eines der Zeichen long, short oder unsigned dabei stehen. Vielleicht wollt ihr ja den Automaten entsprechend ändern?

(22.1)

Dies ist eine einfache Anwendung des Pumping Lemma. Wir wissen aus dem Beweis des PL, dass die im PL erwähnte Schranke n einfach die Zahl der Zustände eines Automaten ist, der die fragliche Sprache erzeugt. Diese Schranke ist im Beispiel drei, wir haben aber ein längeres Wort, mithin muss es eine Zerlegung xyz von amtmta geben, so dass auch xy^iz noch in der Sprache ist. Damit ist sie aber unendlich.

(23.1)

Eine mögliche Ableitung ist

$$B \to SB \to QSB \to QQQB$$

– wenn ihr euch den Baum dazu malt, seht ihr, dass wir einen Fahrer für Kraftwagen für Lasten haben (im Gegensatz zu einem für Personen, paradigmatisch gesehen). Gut, ich habe etwas betrogen, weil ich Kraftwagen nicht zerlegt habe, und Wagen für Kraft geht natürlich nicht – es muss in diesem Fall eben „durch Kraft betriebener Wagen“ heißen.

Alternativ wäre etwa denkbar

$$B \to SB \to SQB \to QQQB$$

– dabei hätten wir einen Fahrer für Wagen für Lastkraft. Das klappt offenbar weit schlechter.

Wollte man Komposita mit regulären Grammatiken erzeugen, müssten alle Regeln einseitig linear sein. Wir hätten im Beispiel also entweder einen Fahrer für Wagen für Kraft für Last oder eine Last für eine Kraft für einen Wagen für einen Fahrer. Beides ist natürlich Unsinn.