Automaten und Formale Sprachen
Wintersemester 2001/2002

Universität Leipzig
Institut für Informatik

Erläuterungen zur Prüfungs-Klausur

Aufgabe 1

a) Typ 3 (= regulär = linkslinear = rechtslinear).

b) Ja.

Schreibfehler in Aufgabenstellung (nach „Untersuchen Sie“ fehlt ein Komma, statt „erzeugte Grammatik“ muß „erzeugte Sprache“ stehen)

c) $L(G) = \{a^x b^y \mid x, y \geq 1\}$. Häufig Syntaxfehler bei Mengen-Angabe, z. B. Mengenklammern vergessen.

Aufgabe 2

Man wählt etwa $x = a^n ba^n \in L$ und muß sicherstellen, daß das „aufgepumpte Teilwort“ aus lauter gleichen Buchstaben besteht. Das geht sowohl mit der komplizierten ($x = uv_1v_2v_3w$ und v_2 pumpen) als auch mit der einfachen ($x = uvw$ und v pumpen) Variante.

Fehler: Bedingungen an Länge der Wörter x und der aufzupumpenden Teile (v bzw. v_2) nicht beachtete, falsche (oder gar keine) Quantoren für die Zerlegungs-Bausteine, deswegen nicht *alle*, sondern nur eine Zerlegung betrachtet.

Aufgabe 3

Lösung: man zählt die a modulo 3, und ignoriert die b. Benötigt drei Zustände.

Häufigster Fehler: $l_a(w) \cong 1 \bmod 3$ fälschlicherweise aufgefaßt als $l_a(w) = (1 \bmod 3) = 1$. Die Kongruenz-Schreibweise bedeutet aber $(l_a(w) \cong 1) \bmod 3$, d. h. die Zahlen sind nicht gleich, sondern liegen nur in der gleichen Restklasse.

Aufgabe 4

Fehler in Aufgabenstellung: $\{a^m ba^m\}$ ist eine ein-elementige Sprache (die ist endlich, also auch regulär). (Das hat auch wenigstens ein Student erkannt.) Gemeint war $\{a^m ba^m \mid m \geq 0\}$.

a) Die Bemerkung („…rechtslineare Grammatik“) trug nichts zur Lösung bei.

Beweis nur möglich mit Pumping-Lemma, wie bei Aufgabe 2, deswegen auch im wesentlichen gleiche Bewertung, gleiche Fehler.

b) Regelmenge $S \to aSa \mid b$.

Häufiger Fehler: $S \to AbA, A \to aA \mid \epsilon$, das erzeugt aber $a^*ba^* \neq L$.

Aufgabe 5

Zu Grammatiken $G_i = (V_i, \Sigma_i, P_i, S_i)$ für L_i mit disjunkten V_i konstruiert man am besten eine Grammatik $G = (V_1 \cup V_2 \cup \{S\}, \Sigma_1 \cup \Sigma_2, P_1 \cup P_2 \cup \{S \to S_1, S \to S_2\}, S)$, mit neuem Startsymbol S.

Fehler bei Grammatik-Konstruktion: man darf nicht in G_i jedes S_i durch *ein* neues S ersetzen. Gegenbeispiel: $P_1 = \{S_1 \to aS_1, S_1 \to \epsilon\}$ erzeugt a^*, $P_2 = \{S_2 \to bS_2, S_2 \to \epsilon\}$ erzeugt b^*, aber $P = \{S \to aS, S \to bS, S \to \epsilon\}$ erzeugt $(a+b)^*$, und das ist *nicht* $a^* + b^*$.

Kürzeste Lösung geht direkt über reguläre Ausdrücke (L_i regulär $\Rightarrow$ es gibt reguläre Ausdrücke A_i für $L_i \Rightarrow \quad (A_1 + A_2)$ ist regulärer Ausdruck für $L_1 \cup L_2$),

andere Lösung benutzt Abschluß regulärer Sprachen gegen Komplement und Durchschnitt, und De-Morgansche Regel.

Aufgabe 6

Antwort: Nein, Beweis: wir nehmen irgendeine nicht kontextfreie Sprache L, zum Beispiel $\{a^n b^n c^n \mid n \geq 0\}$, und betrachten die ein-elementigen Sprachen $L_i = \{a^i b^i c^i\}$. Die sind alle endlich, also kontextfrei (sogar regulär), aber ihre Vereinigung ist L, also nicht kontextfrei.

Beliebte Fehler:

„die Grammatik müßte dann unendlich viele Regeln haben“: daraus folgt zunächst gar nichts, denn damit könnte man auch „zeigen“, daß es keine unendlichen kontextfreien Sprachen gibt, was nicht stimmt.

„man vereinigt (wie bei Aufgabe 5) der Reihe nach...“: das geht zwar beliebig, aber nur endlich oft. Es ist ein Unterscheid zwischen *beliebig oft* und *unendlich oft.*

Analogie: Es gibt natürliche Zahlen mit *beliebig* vielen Primfaktoren (d. h. tatsächlich: für jedes k existiert ein n_k, so daß n_k wenigstens k Primfaktoren besitzt), aber es gibt keine natürliche Zahl mit *unendlich* vielen Primfaktoren.

Große Unsicherheit bei den Begriffen *unendlich, aufzählbar, (über-)abzählbar.*