Grundlagen der theoretischen Informatik, SS 2005 — 9. Übungsblatt

Abgabe bis Montag, 27. Juni 2005, 10:15 Uhr

64. (a) L_1 sei eine kontextfreie Sprache über dem Alphabet $\{0,1\}$, und

$$L_2 = \{\, w_1 0 w_2 0 \dots w_n 0 \mid w_1 w_2 \dots w_n \in L_1,\ n \geq 0 \,\}$$

sei die Sprache, wo nach jedem Buchstaben jedes Wortes eine Null eingefügt ist. Beweisen Sie, dass L_2 ebenfalls kontextfrei ist.

(b) Lösen Sie die gleiche Aufgabe für die Sprache

$$L_3 = \{\, w_1 0 w_2 1 w_3 0 w_4 1 w_5 0 \dots \mid w_1 w_2 \dots w_n \in L_1,\ n \geq 0 \,\},$$

wo nach jedem Buchstaben abwechselnd eine Null und eine Eins eingefügt wird.

65. Wenn man beim Beweis des Pumpinglemmas den *längsten* Pfad im Ableitungsbaum betrachtet, kann man die stärkere Aussage herleiten, dass es eine Zerlegung $w = xyzuv$ mit der *Längenbeschränkung* $|yzu| \leq n_0$ gibt, sodass all Wörter $xy^i zu^i v \in L$ sind.

66. (10 Punkte) Beweisen Sie, dass die folgenden Sprachen *über dem Alphabet* $\{a, b, c\}$ nicht kontextfrei sind.

(a) $L_1' = \{\, w \mid w$ enthält doppelt so viele a's wie b's und doppelt so viele b's wie c's $\}$

(b) $L_2 = \{\, a^i b^j c^k \mid i > j \geq k \geq 1 \,\}$

67. Schreiben Sie eine kontextfreie Grammatik für die regulären Ausdrücke über dem Alphabet $\{0,1\}$. Sie können annehmen, dass die Ausdrücke voll geklammert sind, entsprechend der ursprünglichen Definition regulärer Ausdrücke, zum Beispiel

$$(((0 + (1)^*)) \cdot ((0 + (0 \cdot 1)) + \varepsilon))$$

68. Kontextfreie Sprachen sind unter Umkehrung (Spiegelung) abgeschlossen.

69. Entfernen Sie überflüssige Variablen und Regeln aus dieser Grammatik:

$$\begin{array}{lll} S \to AB \mid CA & B \to BC \mid AB \mid BDE & D \to ADEba \mid ACA \\ A \to a & C \to aB \mid c & E \to DD \mid BA \end{array}$$

70. (0 Punkte) Übersetzen Sie den folgenden vereinfachten Ausschnitt aus der Syntax der Programmiersprache Java aus der EBNF (extended Backus-Naur-Form) in entsprechende kontextfreie Regeln:

TryBlock	::= **try** Block Catches | **try** Block [Catches] **finally** Block
Catches	::= CatchClause { CatchClause }
CatchClause	::= **catch** **(** Exception Identifier **)** Block
Exception	::= Identifier

Die Terminalsymbole sind hier fett gedruckt. Sie können die Variablen durch passende einbuchstabige Namen abkürzen. (Wozu dient eigentlich die Variable „Exception“, die man doch genausogut zusammen mit der Regel „Exception ::= Identifier“ eliminieren könnte?)

71. (a) Bleibt eine eindeutige Grammatik bei der Transformation in Chomsky-Normalform eindeutig?

(b) G sei eine Grammatik in Chomsky-Normalform, die eine endliche Sprache erzeugt. Finden Sie eine Schranke für das längste Wort in $L(G)$.

(c) Wie lang kann die Ableitung eines Wortes w bei einer Grammatik in Chomsky-Normalform höchstens sein?