Prof. Dr. Hans-Jörg Kreowski, Renate Klempien-Hinrichs
Studiengang Informatik
MZH 3260, Tel.: 2956, 3697 (Sekr.), Fax: 4322
E-Mail: {kreo, rena}@informatik.uni-bremen.de

23. Oktober 2000
Aufgaben 1

Theoretische Informatik 2

1. Übungsblatt

1. Entwirf eine Grammatik, die alle *while*-Programme in PASCALchen erzeugt.

   Dabei wird empfohlen, sich an der syntaktischen Beschreibung in 6.1.1 bis 6.1.3 des Skripts Theoretische Informatik 1 zu orientieren, jedoch als Variablen nur Zeichenketten der Form Xk mit $k \in \mathbb{N}$ zu betrachten (vgl. 6.2) und auf Makroanweisungen zu verzichten. Als nichtterminale Zeichen sollen das Startsymbol <*program*> sowie

   <*compound*>, <*assign*>, <*iteration*>, <*stmt*>, <*stmtlist*>,
   <*var*>, <*expr*>, <*nat*>, <*cipherseq*>, <*cipher*>

   verwendet werden. 40%

2. Betrachte die Grammatik G_{riddle}, die S als Startsymbol, a als das einzige terminale Zeichen sowie die folgenden Produktionen besitzt:

   $S ::= AaBD$, $aB ::= Baa$, $AB ::= AC$, $Ca ::= aC$, $CD ::= BD$, $A ::= \lambda$, $BD ::= \lambda$.

   Welche Sprache wird von G_{riddle} erzeugt? Die Antwort ist auf etwa einer halben Seite anhand des Ableitungsprozesses zu begründen. 30%

3. Betrachte das terminale Alphabet $\{A, T, C, G\}$, das die Anfangsbuchstaben der vier Nukleinbasen der DNS enthält. Ein Einzelstrang einer DNS besteht aus einem Wort über diesem Alphabet. Nach der berühmten Watson-Crick-Komplementarität lässt sich jeder Einzelstrang eindeutig zu einem Doppelstrang ergänzen, wobei folgende Gesetzmäßigkeit gilt:

   $\overline{A} = T$, $\overline{T} = A$, $\overline{C} = G$, $\overline{G} = C$, sowie $\overline{\lambda} = \lambda$ und $\overline{uv} = \overline{u}\,\overline{v}$ für alle $u, v \in \{A, T, C, G\}^*$.

   Dabei bezeichnet $\overline{w}$ den komplementären Einzelstrang zu einem gegebenen Einzelstrang w und beide zusammen bilden den Doppelstrang.

   Die folgenden beiden Sprachen beschreiben also die Menge aller DNS-Doppelstränge:

   (a) $L_{mirror} = \{w\, trans(\overline{w}) \mid w \in \{A, T, C, G\}^*\}$, 20%
   (b) $L_{twin} = \{w\,\overline{w} \mid w \in \{A, T, C, G\}^*\}$. 30%

   Dabei ist $trans(w)$ die *Transposition* des Wortes w (d.h. $trans(\lambda) = \lambda$ und $trans(xu) = trans(u)x$ für $x \in \{A, T, C, G\}$ und $u \in \{A, T, C, G\}^*$).

   Für L_{mirror} und L_{twin} sollen Grammatiken entworfen werden. Es wird empfohlen, erst L_{mirror} zu erzeugen. L_{twin} ist ziemlich tückisch. Aber es lässt sich zumindest erzeugen, indem man die Idee für L_{mirror} mit der “Technik” verknüpft, die in G_{riddle} verwendet ist.

Abgabe: bis zum 6. November 2000 in den Tutorien.