Technische Universität Chemnitz Sommersemester 2002
Professur Theoretische Informatik
Prof. Dr. A. Goerdt, Dipl.-Inf. F. Schädlich, Dipl.-Inf. A. Lanka

12. Übung „Theoretische Informatik II“
Lösungsvorschläge

Aufgabe 1

a) (1) $\{(aa, ab,),\ (a, bb),\ (bba, a),\ (a, ab),\ (b, a)\}$

$$\begin{array}{ccllll} x & = & a & b & a & bba \\ y & = & ab & a & bb & a \\ & & 4, & 5, & 2, & 3 \end{array}$$

$\Rightarrow x = y \Rightarrow$ eine Lösung ist $(4, 5, 2, 3)$

(2) $\{(ab, b,),\ (aba, ba),\ (ab, ba),\ (a, abb)\}$

Eine mögliche Lösung muß offensichtlich mit dem Paar (a, abb) beginnen:
Man kann dann aber kein weiteres Wortpaar finden, um zu einer Lösung zu gelangen.

(3) $\{((01, 1),\ (10, 01),\ (10, 101),\ (110, 100)\}$

$$\begin{array}{cclll} x & = & 10 & 110 & 01 \\ y & = & 101 & 100 & 1 \\ & & 3, & 4, & 1 \end{array}$$

$\Rightarrow x = y \Rightarrow$ eine Lösung ist $(3, 4, 1)$

b) Es gibt unendlich viele Lösungen, da eine gefundene Lösung beliebig oft hintereinander geschrieben werden kann.

c) (1) Bei beiden Grammatiken ist $\Sigma = \{a, b, \$, a_1, \ldots, a_5\}$.

Regeln für G_1:

$$\begin{aligned}
S &\to A\$B \\
A &\to a_1Aaa \mid a_1aa \mid \\
&\quad\; a_2Aa \mid a_2a \mid \\
&\quad\; a_3Abba \mid a_3bba \mid \\
&\quad\; a_4Aa \mid a_4a \mid \\
&\quad\; a_5Ab \mid a_5b \\
B &\to baBa_1 \mid baa_1 \mid \\
&\quad\; bbBa_2 \mid bba_2 \mid \\
&\quad\; aBa_3 \mid aa_3 \mid \\
&\quad\; baBa_4 \mid baa_4 \mid \\
&\quad\; aBa_5 \mid aa_5
\end{aligned}$$

Regeln für G_2:

$$
\begin{aligned}
S &\to a_1Sa_1 \mid a_2Sa_2 \mid a_3Sa_3 \mid a_4Sa_4 \mid a_5Sa_5 \mid T \\
T &\to aTa \mid bTb \mid \$
\end{aligned}
$$

(2) Bei beiden Grammatiken ist $\Sigma = \{a, b, \$, a_1, \ldots, a_4\}$.

Regeln für G_1:

$$
\begin{aligned}
S &\to A\$B \\
A &\to a_1Aab \mid a_1ab \mid \\
&\quad\; a_2Aaba \mid a_2aba \mid \\
&\quad\; a_3Aab \mid a_3ab \mid \\
&\quad\; a_4Aa \mid a_4a \mid \\
B &\to bBa_1 \mid ba_1 \mid \\
&\quad\; abBa_2 \mid aba_2 \mid \\
&\quad\; abBa_3 \mid aba_3 \mid \\
&\quad\; bbaBa_4 \mid bbaa_4 \mid
\end{aligned}
$$

Regeln für G_2:

$$
\begin{aligned}
S &\to a_1Sa_1 \mid a_2Sa_2 \mid a_3Sa_3 \mid a_4Sa_4 \mid a_5Sa_5 \mid T \\
T &\to aTa \mid bTb \mid \$
\end{aligned}
$$

Bemerkungen:

1. Die beiden Sprachen $L_1 = L(G_1)$ und $L_2 = L(G_2)$ sind auf Grund der Konstruktion immer deterministisch kontextfrei.
2. Mittels des Pumpings-Lemmas für kontextfreie Sprachen kann leicht gezeigt werden, daß $L_1 \cap L_2$ nicht kontextfrei ist (sofern im Schnitt Wörter enthalten sind).

Diese beiden Aussagen werden für (d) benötigt.

d) Es seien G_1 und G_2 wieder die beiden deterministisch kontextfreinen Grammatiken, die aus einem PCP konstruiert wurden. Ferner sei $L_i = L(G_i)$ und $\overline{L_i} = \Sigma^* \backslash L_i$ das Komplement von L_i. Die deterministisch kontextfreien Sprachen sind bezüglich des Komplements abgeschlossen, es sind also auch $\overline{L_1}$ und $\overline{L_2}$ (deterministisch) kontextfrei. Wie bereits gezeigt, ist das Problem „$L_1 \cap L_2 = \emptyset$?“ nicht entscheidbar. Dieses Problem werden wir weiter auf andere reduzieren, um deren Nichtentscheidbarkeit zu zeigen.

(1) Ein PCP hat entweder keine oder unendlich viele Lösungen – siehe (b) $\Rightarrow L_1 \cap L_2 \neq \emptyset \Leftrightarrow |L_1 \cap L_2| = \infty$. $\Rightarrow PCP \leq_p$ „$|L(G) \cap L(H)| = \infty$?“

(2) Ist $L_1 \cap L_2$ leer, so liegt eine kontextfreie Sprache vor. Ist der Schnitt nicht leer, so ist $L_1 \cap L_2$ nicht kontextfrei (Beweis über Pumping-Lemma). $\Rightarrow L_1 \cap L_2 = \emptyset \Leftrightarrow L_1 \cap L_2$ ist kontextfrei $\Rightarrow PCP \leq_p$ „$L(G) \cap L(H)$ kontextfrei?“.

(3) $L_1 \cap L_2 = \emptyset \Leftrightarrow L_1 \subseteq \overline{L_2}$. $\overline{L_2}$ ist kontextfrei $\Rightarrow PCP \leq_p$ „$L(G) \subseteq L(H)$?“.

(4) $L_1 \cap L_2 = \emptyset \Leftrightarrow L_1 \subseteq \overline{L_2} \Leftrightarrow L_1 \cup \overline{L_2} = \overline{L_2}$. $\overline{L_2}$ ist kontextfrei, damit ist aber auch $L_1 \cup \overline{L_2}$ kontextfrei. $\Rightarrow PCP \leq_p$ „$L(G) = L(H)$?“.

Aufgabe 2

Sei folgendes PCP über dem einelementigen Alphabet $\Sigma = \{1\}$ gegeben:
$\{(1^{a_1}, 1^{b_1}),\ (1^{a_2}, 1^{b_2}),\ \ldots,\ (1^{a_n}, 1^{b_n})\}$

- Sei $a_i > b_i$ bzw. $a_i < b_i$ für alle i, dann gibt es keine Lösung.
- Sei $a_i = b_i$ für ein i, dann gibt es die Lösung (i).
- Seien $a_i > b_i$ und $a_j < b_j$ für gewisse i und j. Dann gibt es eine Lösung.
  Beweis:
  Seien ohne Beschränkung der Allgemeinheit $(1^{a_i}, 1^{b_i}),\ (1^{a_j}, 1^{b_j}) \in PCP$ mit $a_i > b_i$ und $a_j < b_j$.
  $a_i \cdot c + a_j \cdot d = b_i \cdot c + b_j \cdot d$
  $(a_i - b_i) \cdot c = (b_j - a_j) \cdot d$
  Nun setzten wir $c := (b_j - a_j) \ > 0$ und $d := (a_i - b_i) \ > 0$ und haben somit eine Lösung gefunden.

Aufgabe 3

Sei $X = \{x\ :\ x$ lässt sich schreiben als $x_{i_1} x_{i_2} \ldots x_{i_n}\} = \{x_1, x_2, \ldots, x_n\}^+$.
Sei $Y = \{y\ :\ y$ lässt sich schreiben als $y_{i_1} y_{i_2} \ldots y_{i_m}\} = \{y_1, y_2, \ldots, y_m\}^+$.
Problem: Ist $X \cap Y = \emptyset$?
X, Y sind reguläre Sprachen da $\{\ldots\}^+$ regulär ist.
$\Rightarrow Z = X \cap Y$ ist regulär.
$\Rightarrow Z = \emptyset$? ist für reguläre Sprachen, und somit für Z, entscheidbar.
Beweisidee:
Man betrachtet den DFA zu Z.
Es gibt einen Weg von einem Startzustand zu einem Endzustand $\Leftrightarrow$ Es gibt ein Wort $z \in Z$.

Aufgabe 4

Sei $x = 01001$ das Wort das auf dem Band einer Turing-Maschine steht. Dann ist $01z_i001$ die Konfiguration der Turing-Maschine im Zustand z_i, wenn der Schreib-Lese-Kopf der Maschine auf der zweiten 0 steht..

$x = 000$

Der Konfigurationsverlauf der Turing-Maschine wird nun auf ein MPCP abgebildet.

Start:

$$\left\{ \begin{pmatrix} \# \\ \#z_0000\# \end{pmatrix}, \right.$$

Kopieren:

$$\begin{pmatrix} \# \\ \# \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} \square \\ \square \end{pmatrix},$$

δ:

$$\begin{pmatrix} z_00 \\ 0z_E \end{pmatrix}, \begin{pmatrix} z_01 \\ \square z_1 \end{pmatrix}, \begin{pmatrix} z_0\# \\ \square z_1\# \end{pmatrix}, \begin{pmatrix} z_0\square \\ \square z_1 \end{pmatrix}, \begin{pmatrix} z_10 \\ 0z_E \end{pmatrix}, \begin{pmatrix} z_11 \\ \square z_1 \end{pmatrix}, \begin{pmatrix} z_1\# \\ \square z_1\# \end{pmatrix}, \begin{pmatrix} z_1\square \\ \square z_1 \end{pmatrix},$$

Löschen:

$$\begin{pmatrix} z_E0 \\ z_E \end{pmatrix}, \begin{pmatrix} z_E1 \\ z_E \end{pmatrix}, \begin{pmatrix} z_E\square \\ z_E \end{pmatrix}, \begin{pmatrix} 0z_E \\ z_E \end{pmatrix}, \begin{pmatrix} 1z_E \\ z_E \end{pmatrix}, \begin{pmatrix} \square z_E \\ z_E \end{pmatrix},$$

Ende:

$$\left.\begin{pmatrix} z_E\#\# \\ \# \end{pmatrix}\right\}$$

$$\#|z_00|0|0|\#|0z_E00\#|z_E0|0\#z_E0|\#z_E|\#\#$$

$$\#z_0000\#0|z_E|0|0|\#|z_E00\#|z_E|0\#z_E|\#|\#$$

$x = 111$

Der Konfigurationsverlauf der Turing-Maschine wird nun auf ein MPCP abgebildet.

Start:

$$\left\{\begin{pmatrix} \# \\ \#z_0111\# \end{pmatrix},\right.$$

Kopieren:

$$\begin{pmatrix} \# \\ \# \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} \square \\ \square \end{pmatrix},$$

δ:

$$\begin{pmatrix} z_00 \\ 0z_E \end{pmatrix}, \begin{pmatrix} z_01 \\ \square z_1 \end{pmatrix}, \begin{pmatrix} z_0\# \\ \square z_1\# \end{pmatrix}, \begin{pmatrix} z_0\square \\ \square z_1 \end{pmatrix}, \begin{pmatrix} z_10 \\ 0z_E \end{pmatrix}, \begin{pmatrix} z_11 \\ \square z_1 \end{pmatrix}, \begin{pmatrix} z_1\# \\ \square z_1\# \end{pmatrix}, \begin{pmatrix} z_1\square \\ \square z_1 \end{pmatrix},$$

Löschen:

$$\begin{pmatrix} z_E0 \\ z_E \end{pmatrix}, \begin{pmatrix} z_E1 \\ z_E \end{pmatrix}, \begin{pmatrix} z_E\square \\ z_E \end{pmatrix}, \begin{pmatrix} 0z_E \\ z_E \end{pmatrix}, \begin{pmatrix} 1z_E \\ z_E \end{pmatrix}, \begin{pmatrix} \square z_E \\ z_E \end{pmatrix},$$

Ende:

$$\left.\begin{pmatrix} z_E\#\# \\ \# \end{pmatrix}\right\}$$

$$\#|z_01|11\#\square|z_11|1\#\square\square|z_11|\#\square\square\square|z_1\#|\square\square\square|\square z_1\# \quad \ldots$$

$$\#z_0111\#|\square z_1|11\#\square|\square z_1|1\#\square\square|\square z_1|\#\square\square\square|\square z_1|\#\square\square\square\square|\square z_1\# \quad \ldots$$