Hans U. Simon
Christian Brandl

Bochum, den 20.03.2006

Klausur

Theoretische Informatik

WS 05/06

Bearbeitungszeit: 3 Stunden

Name:
Matrikelnummer:
Studiengang:

I Automatentheorie und Formale Sprachen

Aufgabe 1 (23 Punkte)

a) Gib eine reguläre Grammatik G an mit

$$L = \{w \in \{0,1\}^* : \ w \text{ enthält höchstens eine Doppelnull}\}.$$

(Beachte: Das Wort 000 enthält 00 zweimal.)

b) Konstruiere einen äquivalenten NFA nach dem Verfahren aus der Vorlesung.

Aufgabe 2 (16 Punkte)

Weise die Nichtkontextfreiheit nach von

$$L = \{0^r 1^s 2^t : \ r, s, t \in \mathbb{N}, r * s = t\}.$$

Aufgabe 3 (15 Punkte)

Bringe nach dem Schema der Vorlesung folgende Grammatik G in Chomsky Normalform:

$$G = \big(\{S, A, B\}\ ,\ \{a, b\}\ , P\ ,\ S\big)$$

mit P definiert durch

$$S \to ASB$$

$$A \to aAS \mid a \mid \epsilon$$

$$B \to SBS \mid A \mid bb$$

II Berechenbarkeitstheorie

Aufgabe 4 (15 Punkte)

Schreibe ein LOOP-Programm für die Funktion

$$f : \mathbb{N}^2 \to \mathbb{N}, f(x_1, x_2) = \sum_{i=0}^{x_1} \frac{(x_1 + x_2)!}{i!}$$

Dabei dürfen die LOOP-simulierbaren Konstrukte aus der Tabelle auf Folie 35 des Abschnitts „Berechenbarkeitstheorie und Church'sche These“verwendet werden (inklusive der DIV-Operation).

Aufgabe 5 (15 Punkte)

Stelle die Turingtafel einer Einband-DTM M auf, so dass gilt

$$T(M) = \{w \in \{0,1\}^* : \; |w|_0 = |w|_1\}.$$

(Die Transitionsfunktion δ muss dabei nicht notwendig an allen Stellen definiert werden.)
Beschreibe kurz die Arbeitsweise deiner Maschine.

III Komplexitätstheorie

Aufgabe 6 (16 Punkte)

Das Problem *Double Digest* ist wie folgt definiert.

Eingabe: Drei Listen natürlicher Zahlen

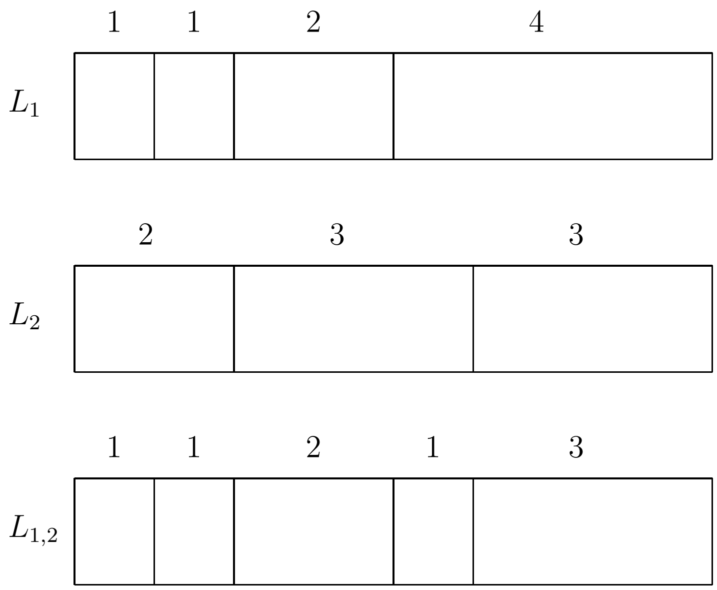
$$L_1 = (w_1, \ldots, w_r), L_2 = (u_1, \ldots, u_s), L_3 = (v_1, \ldots, v_t)$$

mit

$$\sum_{i=1}^{r} w_i = \sum_{i=1}^{s} u_i = \sum_{i=1}^{t} v_i$$

Schnitt von Listen: An einem Beispiel wird im folgenden demonstriert wie der mit $L_{1,2}$ bezeichnete <u>Schnitt</u> der Listen L_1 und L_2 zu verstehen ist.

$$L_1 = (1, 1, 2, 4), L_2 = (2, 3, 3)$$



Frage: Kann man die Einträge innerhalb der Listen L_1, L_2 und L_3 jeweils so permutieren, dass gilt $L_{1,2} = L_3$?

Weise durch eine polynomielle Reduktion von *Partition* auf *Double Digest* die NP-Härte von *Double Digest* nach.