Hans U. Simon
Malte Darnstädt

Bochum, den 16.06.2010
Abgabe bis zum 23.06. in der Übung

Übungen zur Vorlesung

Komplexitätstheorie

Sommer 2010

Übungsblatt 8

Aufgabe 8.1

Sei $f : \{0,1\}^n \to \{0,1\}$ eine beliebige Abbildung. Beweise:

a) (Die Universalität von $\wedge, \vee, \neg$) Jedes f kann als CNF-Formel mit Hilfe von $n2^n$ $\wedge, \vee$-Operationen ausgedrückt werden. Daher kann f auch von einem Schaltkreis der Größe $O(n2^n)$ berechnet werden.

b) Man kann für jedes f rekursiv einen Schaltkreis aufbauen, der f mit $O(2^n)$ Gattern berechnet.

c) Kombiniere die Konstruktion aus b) geschickt mit der Aussage von a) um zu zeigen: f kann von einem Schaltkreis mit $O(\frac{2^n}{n})$ Gattern berechnet werden.

Aufgabe 8.2

Sei uniform-NC^d die Klasse NC^d mit der Einschränkung, dass die Schaltkreisfamilien logspace-uniform sind.
Zeige, dass uniform-$\mathrm{NC}^1 \subseteq \mathcal{L}$. Folgere daraus PSPACE $\neq$ uniform-NC^1.

Aufgabe 8.3

Wir definieren eine Klasse von Schaltkreisfamilien, die *PH-Circuits*, über folgende Eigenschaften:

- die Gatter sind AND-, OR- oder NOT-Gatter
- die AND- und OR-Gatter haben unbeschränkt (exponentiell) viele Eingänge
- die NOT-Gatter befinden sich nur auf der Eingabeebene
- folgende Abbildungen sind in polynomieller Zeit berechenbar:
  - SIZE(n): Liefert die Anzahl (in Binärdarstellung) der Gatter von Schaltkreis C_n. Dabei spielen die ersten $n + 2$ Gatter die Rolle der Eingabevariablen $x_1, \ldots, x_n$ sowie der Konstanten 0 und 1.

  - TYPE(n,i): Liefert den Typ (AND, OR, NOT, None) von Gatter i in C_n
  - EDGE(n,i,j): Gibt 1 aus, genau dann wenn eine gerichtete Kante von Gatter i zu Gatter j in C_n existiert
- die Größe der Schaltkreise ist durch $2^{n^{O(1)}}$ beschränkt
- die Tiefe der Schaltkreise ist konstant

Zeige: In PH sind genau die Sprachen, die von PH-Circuits erkannt werden.

Aufgabe 8.4

Zeige, dass man einen effizienten Zufallszahlengenerator für den Zahlenbereich 1 bis N implementieren kann, wenn nur zufällige Münzwürfe zur Verfügung stehen. Das heißt, zeige dass für jedes $N \in \mathbb{N}$ und $\delta > 0$ ein probabilistischer Algorithmus A mit folgenden Eigenschaften exisitert:

- Die Laufzeit von A ist $\text{poly}(\log N \log(1/\delta))$
- A gibt ein Element aus der Menge $\{1, \ldots, N, ?\}$ aus
- Unter der Bedingung, dass A nicht ? ausgibt, ist die Ausgabe auf der Menge $\{1, \ldots, N\}$ gleichverteilt
- Die Wahrscheinlichkeit, dass ? ausgegeben wird, ist höchstens δ