Prof. Dr. R. Schrader
Birgit Engels [engels@zpr.uni-koeln.de]

WS 2006/2007

Theoretische Informatik

6. Übung, Abgabe Mittwoch, 29.11.2006

Aktuelle Informationen bezüglich der Vorlesung und der Übungen finden sich unter :
http://www.zaik.uni-koeln.de/AFS/teachings/courses/ThInf/index.html und
http://www.zaik.uni-koeln.de/AFS/teachings/courses/ThInf/uebungen.html

Aufgabe 23 (Probabilistische Algorithmen — Exkurs Mobilkommunikation — 6 Punkte):
Betrachten Sie folgendes Problem aus der Mobilkommunikation: WLAN-Netze ermöglichen nicht nur den Zugang zum Internet (über eine *feste* Netzwerktopologie, *Hotspots*), sondern können auch zur direkten Kommunikation zwischen einzelnen Rechnern genutzt werden. Diese sind mobil, können sich "spontan" zu Netzwerken zusammenfinden (*ad hoc* Netzwerke) und verfügen daher nicht über eine feste Netzwerktopologie oder vorab bestimmte previligierte "Master"-Rechner. Da gewisse Kommunikationsprotokolle oder andere Netzwerk-Dienste jedoch auf einen solchen angewiesen sind, ergibt sich (unter vielen anderen :o) folgendes Problem:

> **Leader Election:** Aus einer Reihe von Rechnern (Stationen) muss ein eindeutiger Leader ausgewählt werden, dessen Identität alle Stationen (inkl. ihm selbst) kennen. Da im Voraus unbekannt ist, welche Stationen ein Netz bilden, können keine Vorabannahmen gemacht werden und alle Rechner werden als identisch (gleichberechtigt) angenommen. Weiterhin wird kein Rechner "von aussen" bestimmt. Alle Stationen können Nachrichten senden und empfangen, wobei bei gleichzeitigem Senden von 2 oder mehr Stationen eine Kollision entsteht und keine Nachricht empfangen werden kann. Nachrichten (sowie Kollisionen) werden von allen Stationen empfangen (keine spezifische Addressierung, *Broadcast*).

Ein Algorithmus A zur Durchführung der **Leader Election** wird im Prinzip die Gleichberechtigung aller Stationen im Netz nachahmen müssen und für alle Rechner identisch sein. Weiterhin ist A darauf angewiesen durch Nachrichtenaustausch mit den anderen Stationen zu kommunizieren. Falls jedoch auf jeder von n Stationen dieser identische Algorithmus synchron abläuft, wird es zwangsläufig zu dauernden Kollisionen kommen. Analog zum Problem der "dinnierenden Philosophen" (*Dining Philosophers*) bietet sich hier eine Symmetriebrechung durch Einsatz eines randomisierten Algorithmus an.

Bitte wenden !

Betrachten Sie folgenden sehr einfachen Algorithmus A_i, der auf jeder Station i von insgesamt n Stationen simultan ausgeführt wird:

Werfe Münze (Randomisierung, vernachlässige Problem der Pseudo-Zufallszahlen)

- 1: Sende Nachricht `Ich (`i`) bin Leader.`
  - Keine Kollision: Station i ist Leader und alle wissen Bescheid.
  - Kollision: Werfe Münze (Randomisierung)
    - 1: Sende Nachricht `Ich (`i`) bin Leader.` (Weiter wie oben.)
    - 0: Scheide aus (Nur noch Nachrichtenempfang)
- 0: Scheide aus (Nur noch Nachrichtenempfang)

a) Mit welcher Wahrscheinlichkeit terminiert der obige Algorithmus A_i nach t Schritten (Der Algorithmus gilt auf Station i auch im Stand-By-Modus —i empfängt nur noch Nachrichten— als terminiert), ohne dass ein Leader gefunden wird ?

b) Wie gross ist die Wahrscheinlichkeit, dass mittels A_i nach t Schritten ein Leader bestimmt wurde?

c) Verändern Sie den obigen Algorithmus (minimal) so, dass er auf jeden Fall einen Leader findet (Es werden keine weiteren Anforderungen an den Algorithmus gestellt)!

Aufgabe 24 (Probabilistische Turingmaschinen — 2 Modelle — 2 Punkte):
In der Vorlesung wurde eine probabilistische Turingmaschine wie folgt eingeführt:

> Eine Probabilistische Turingmaschine (PTM) M ist eine NTM[1] mit Übergangsfunktion
> $$\delta : Q \times A \to (Q \times A \times \{L, R, N\})^2,$$
> deren beide mögliche Folgezustände jeweils mit Wahrscheinlichkeit $\frac{1}{2}$ gewählt werden (binärer Berechnungsbaum). Dabei kann M akzeptierend, verwerfend oder mit "weiß nicht" ("?") anhalten.

Ebenfalls kann die folgende Definition verwendet werden:

> Eine Probabilistische Turingmaschine (PTM) M' ist eine DTM[2] mit Übergangsfunktion
> $$\delta : Q \times A \times 0, 1 \to (Q \times A \times \{L, R, N\}).$$
> Dabei wird die zusätzliche Information "0" oder "1" von einem separaten "Zufallsband" gelesen, welches nur einmal gelesen und nicht beschrieben werden darf, jedoch (für eine Eingabe x) einen genügend langen zufälligen Binärstring enthält, so dass auch für den längsten von M' auf der Eingabe x durchführbaren Berechnungsweg genügend "Zufallsbits" zur Verfügung stehen. M' kann akzeptierend, verwerfend oder mit "weiß nicht" ("?") anhalten.

[1] Da δ hier keine injektive Übergangsunktion ist und damit die Folgegonfiguration K_{i+1}^M einer Konfiguration K_i^M nicht determiniert ist, kann man eine PTM als spezielle NTM sehen.

[2] Da δ selbst hier als injektive Übergangsunktion definiert werden kann ist und damit die Folgegonfiguration $K_{i+1}^{M'}$ einer Konfiguration $K_i^{M'}$ determiniert ist, kann man eine PTM als erweiterte DTM sehen.

Zeigen Sie die Äquivalenz beider Definitionen:

$$\exists M \texttt{mit} M(x) = f(x) \Leftrightarrow \exists M' \texttt{mit} M'(x) = f(x)$$

Aufgabe 25 (NP vs. coNP — 4 Punkte):
Wie in der Vorlesung definiert, ist die Sprachklasse $coNP$ die Klasse derjenigen Sprachen L (über Σ), deren Komplement $\overline{L} = \Sigma^* \setminus L$ in NP liegt.
Das Verhältnis beider Komplexitätsklassen ist bisher ungeklärt. Nach folgendem Satz

$$\text{“}NP = coNP \Leftrightarrow \exists L : L\ ist\ NP - vollständig, \overline{L} \in NP\text{“}$$

würde es jedoch ausreichen, *eine* NP-vollständige Sprache zu kennen, deren Komplement *ebenfalls* in NP liegt, um zu zeigen, dass $NP = coNP$ gilt. Beweisen Sie die oben angebene Äquivalenz.
Bemerkung: Obwohl keine der beiden Vermutungen bewiesen ist, wird eher $NP \neq coNP$ angenommen.

Aufgabe 26 (Primzahlen, Anwendung RSA-Verfahren — 8 Punkte):
In der Übermittlung privater Daten über ein öffentliches Netzwerk kommen teils sogenannte *Public Key* Kryptosysteme zum Einsatz, die formal als Fünftupel $S = (P, C, K, E, D)$ definiert werden können, wobei :

- P die Menge der möglichen Klartexte
- C die Menge der möglichen codierten Texte
- K die Menge der möglichen Schlüssel
- $E = \{E_k | k \in K\}$ die Menge der Encoder-Funktionen $E_k : P \rightarrow C$
- $D = \{D_k | k \in K\}$ die Menge der Decoder-Funktionen $D_k : C \rightarrow P$

darstellt. Die namensgebende Besonderheit dieser Systeme liegt darin, dass jeder Nutzer des Netzwerkes über ein Paar $(e, d) \in K \times K$ mit $D_d(E_e(x)) = x\ mod\ N$ für alle $x \in P$ verfügt,wobei e öffentlich ist (daher *public* key), d jedoch geheim und aus e nicht (einfach) ableitbar. Nun kann jeder andere Benutzer den öffentlichen Schlüssel e benutzen, um $c = E_e(x)$ zu erzeugen, also die Nachricht x zu verschlüsseln und abzusenden. Jedoch nur der Adressat, der über d verfügt kann die verschlüsselte Nachricht c wiederum als $x = D_d(c)$ entschlüsseln.
Betrachten wir als Beispiel das nach den Erfindern (R.L. **R**ivest, A. **S**hamir, L. **A**dleman) benannnte RSA-Verfahren zur Erzeugung eines Paares (e, d):

Bitte wenden!

- Wähle zufällig zwei große Primzahlen p, q, so daß:
  - $p \neq q$
  - Die Binärdarstellung von p und q jeweils mindestens $\frac{n}{2}$ Bits lang sind.
  - $2^{n-1} \leq pq < 2^n$
- Berechne $N = pq$ und $\|\Phi(N)\| = (p-1)(q-1)$
- Wähle zufällig $e \in \{2, 3, \ldots, \|\Phi(N)\| - 2\}$ mit $ggT(e, \|\Phi(N)\|) = 1$
- Berechne d, so dass $1 < d < \|\Phi(N)\|$ und $de = 1\ mod \|\Phi(N)\|$.

Wir betrachten dabei die Menge der Klartexte $x \in P$ als Bitstring-Darstellungen über $\{0,1\}^{n-1}$ bzw. assoziieren diese mit der entsprechenden Dezimalzahl. Dann erhalten wir mit (N, e) einen öffentlichen Schlüssel und mit (N, d) einen privaten Schlüssel, die wie folgt verwendet werden können:

- Verschlüsseln : $c = E_{(N,e)}(x) = x^e\ mod N$
- Entschlüsseln : $x* = D_{(N,d)}(c) = c^d\ mod N$

Beantworten Sie zum obigen Verfahren folgende Fragen:

a) Wie wird d aus $|\Phi(N)\|$ und e berechnet? Wie kann d ohne $|\Phi(N)\|$ und e nur bestimmt werden ?
Hinweis : Benutzen Sie den euklidischen ggT-Algorithmus und erläutern sie die anschliessende Berechnung von d.

b) Berechnen Sie d für $p = 11$, $q = 23$ und wählen Sie $e = 9$. Zeigen Sie auch, dass $ggT(e, \|\Phi(N)\|) = 1$ und damit eine gültige Wahl ist.

c) Zeigen Sie, dass $x* = x$ gilt,d. h. dass das RSA-Verfahren korrekt ist!
Hinweis : Benutzen Sie den kleinen Fermatschen Satz (Lemma aus der Vorlesung) bezüglich x und p bzw q und erweitern Sie die Gleichung geschickt, um $x* = x$ zu erhalten. Beachten Sie die Voraussetzungen, die e und d (und x) bezüglich N bzw. p und q erfüllen.