Informatik III - Tutorium IX & X (SR -107)

Tut Nr. 11 – Üb11, Komplexitätsklassen, Kryptographie

David Münch

Universität Karlsruhe (TH)
Institut für Informatik
IAKS Beth

28. Januar 2009

Universität Karlsruhe (TH)
Forschungsuniversität • gegründet 1825

Inhaltsverzeichnis

Inhaltsverzeichnis

Inhaltsverzeichnis

Inhaltsverzeichnis

Organisatorisches

Email: muenchdavid@gmail.com

https://www.stud.uni-karlsruhe.de/~uhbro/

Tutorium 09: Mittwochs 8:00 Uhr - Raum -107
Tutorium 10: Mittwochs 9:45 Uhr - Raum -107

Übungsblattabgabe Donnerstag.

Was wollen wir heute erreichen?

Was wollen wir heute erreichen?

- Übungsblatt 11 besprechen

Was wollen wir heute erreichen?

- Übungsblatt 11 besprechen
- Randomisierte Komplexitätsklassen $\mathcal{R}, co - \mathcal{R}, \mathcal{BPP}$ kennen lernen.

Was wollen wir heute erreichen?

- Übungsblatt 11 besprechen
- Randomisierte Komplexitätsklassen $\mathcal{R}, co - \mathcal{R}, \mathcal{BPP}$ kennen lernen.
- Einführung in die Kryptographie mit dem Rabin Kryptosystem

Aufgabe 1

Gegeben ist das folgende Problem:

CLIQUE

Gegeben: Ein ungerichteter Graph $G = (V, E)$ und eine Zahl k.

Gesucht: Besitzt G einen vollständigen Subgraphen mindestens der Grösse k (also einen Subgraph $G' = (V', E')$ mit $|V'| \geq k, V' \subseteq V, E' \subseteq E$ und $\forall v_i, v_j \in V' : (v_i, v_j) \in E'$)?

Aufgabe 1

Gegeben ist das folgende Problem:

CLIQUE

Gegeben: Ein ungerichteter Graph $G = (V, E)$ und eine Zahl k.

Gesucht: Besitzt G einen vollständigen Subgraphen mindestens der Grösse k (also einen Subgraph $G' = (V', E')$ mit $|V'| \geq k, V' \subseteq V, E' \subseteq E$ und $\forall v_i, v_j \in V' : (v_i, v_j) \in E'$)?

Zeigen Sie, dass CLIQUE NP-vollständig ist.

Tipp: Benutzen Sie, dass für SAT und Vertex Cover (eines genügt) bewiesen ist, dass diese Probleme NP-hart sind.

Aufgabe 1

1. z.z. CLIQUE $\in$ NP.
Als erstes sieht man, dass CLIQUE in NP liegt, da es in polynomialer Zeit möglich ist, zu überprüfen, ob eine Menge von Knoten einen vollständigen Teilgraph bildet. Um zu zeigen, dass CLIQUE NP-hart ist, reduziert man ein NP-hartes Problem auf CLIQUE.

2. z.z. VERTEX COVER $\leq_p$ CLIQUE

Sei ein Graph $G = (V, E)$ gegeben. Dann bilde einen Graphen $G' = (V', E')$ mit $V' = V$ und $E' = V \times V \setminus E$ (also $(v, w) \in E' \Leftrightarrow (v, w) \notin E$).

Anschaulich wird ein Komplementärgraph G' zum Graphen G gebildet, so dass bei G' genau die Kanten vorhanden sind, die bei G' nicht vorhanden sind.

Der Graph G hat genau dann ein Vertex Cover höchstens der Größe k wenn es einen vollständigen Subgraphen von G' mindestens der Größe $n = |V| - k$ gibt.

Aufgabe 2

Beweisen Sie folgende Aussagen:

1. Die Klasse NP ist unter Schnittbildung abgeschlossen.

Dabei nehmen wie an, dass alle Sprachen über dem binären Alphabet $A = \{0, 1\}$ definiert sind, also dass NP $\subseteq P(A^*)$.

Aufgabe 2

Beweisen Sie folgende Aussagen:

1. Die Klasse NP ist unter Schnittbildung abgeschlossen.
2. Die Klasse NP ist unter Vereinigung abgeschlossen.

Dabei nehmen wie an, dass alle Sprachen über dem binären Alphabet $A = \{0, 1\}$ definiert sind, also dass NP $\subseteq P(A^*)$.

Aufgabe 2

Beweisen Sie folgende Aussagen:

1. Die Klasse NP ist unter Schnittbildung abgeschlossen.
2. Die Klasse NP ist unter Vereinigung abgeschlossen.
3. Die Klasse NP ist unter dem kleeneschen Sternoperator * abgeschlossen.

Dabei nehmen wie an, dass alle Sprachen über dem binären Alphabet $A = \{0, 1\}$ definiert sind, also dass NP $\subseteq P(A^*)$.

Aufgabe 2

Seien R_{L_1}, R_{L_2} zu $\mathcal{NP}$-Sprachen L_1, L_2 gehörenden polynomiell entscheidbaren Zeugenrelationen. Dann ist

$$R_{L_1 \cap L_2} = \{(x, (w_1, w_2)) | (x, w_1) \in R_{L_1} \text{ und } (x, w_2) \in R_{L_2}\}$$

eine zu $L_1 \cap L_2$ gehörende polynomiell entscheidbare Zeugenrelation, denn ist

$$\begin{aligned} x \in L_1 \cap L_2 &\Leftrightarrow x \in w_1 \text{ und } x \in w_2 \\ &\Leftrightarrow \exists w_1, w_2 : (x, w_1) \in R_{L_1} \text{ und } (x, w_2) \in R_{L_2} \\ &\Leftrightarrow \exists (w_1, w_2) : (x, (w_1, w_2)) \in R_{L_1 \cap L_2}. \end{aligned}$$

Entsprechend ist

$$R_{L_1 \cup L_2} = \{(x, (w_1, w_2)) | (x, w_1) \in R_{L_1} \text{ oder } (x, w_2) \in R_{L_2}\}$$

eine zu $L_1 \cup L_2$ gehörende Zeugenrelation.

Aufgabe 2

Sei $L \in \mathcal{NP}$, R_L die zugehörige Zeugenrelation, so definieren wir

$$R_{L^*} = \{x, ((x_1, w_1), \ldots, (x_k, w_k)) | x = x_1 x_2 \ldots x_k \text{ und } \forall i = 1, \ldots, k : (x_i, w_i) \in R_L\}.$$

R_{L^*} ist polynomiell entscheidbar, denn die Länge der Zeugen $((x_1, w_1), \ldots, (x_k, w_k))$ ist polynomiell beschränkt in der Länge von x (höchsten $|x|$ Paare (x_i, w_i)). Es gilt

$$\begin{aligned} x \in L^* &\Leftrightarrow \exists x_1, \ldots, x_k \in L : x = x_1 \ldots x_k \\ &\Leftrightarrow \exists x_1, \ldots, x_k \exists w_1, \ldots, w_k : x = x_1 \ldots x_k \text{ und} \\ &\forall i = 1, \ldots, k : (x_i, w_i) \in R_L \\ &\Leftrightarrow \exists (x_1, w_i), \ldots, (x_k, w_k) : (x, ((x_1, w_i), \ldots, (x_k, w_k))) \in R_{L^*} \end{aligned}$$

Randomisierte Komplexitätsklassen $\mathcal{R}, co - \mathcal{R}, \mathcal{BPP}$.

Definition: stochastische Turingmaschine (RTM)

Eine RTM ist eine NTM, die indeterministisch Entscheidungen zufällig und gleichverteilt trifft.

Definition: stochastische Turingmaschine (RTM)

Eine RTM ist eine NTM, die indeterministisch Entscheidungen zufällig und gleichverteilt trifft.

Definition: random polynomial $\mathcal{R}$ bzw. $\mathcal{RP}$

Die Komlexitätsklasse $\mathcal{R}$ besteht aus den Entscheidungsproblemen, die von polynomzeit RTMs gelöst werden.
Akzeptiert die RTM, so ist $x \in L$ ohne Irrtumswahrscheinlichkeit.
Akzeptiert die RTM nicht, so ist ein Irrtum möglich.

Idee: Die Wahrscheinlichkeit eines Irrtums kann durch Wiederholung beliebig klein gemacht werden.

Es gilt: $\mathcal{P} \subseteq \mathcal{R} \subseteq \mathcal{NP}$

Vermutlich ist $\mathcal{R} \neq co - \mathcal{R}$
$\Rightarrow$ Zero-error Probabilistic Polynomial Time$(\mathcal{ZPP}) = \mathcal{R} \cap co - \mathcal{R}$

Idee: Die Wahrscheinlichkeit eines Irrtums kann durch Wiederholung beliebig klein gemacht werden.

Es gilt: $\mathcal{P} \subseteq \mathcal{R} \subseteq \mathcal{NP}$

Vermutlich ist $\mathcal{R} \neq co - \mathcal{R}$
$\Rightarrow$ Zero-error Probabilistic Polynomial Time$(\mathcal{ZPP}) = \mathcal{R} \cap co - \mathcal{R}$

$\mathcal{ZPP}$ gibt immer die richtige Antwort zurück (daher zero-error).
Die Laufzeit ist nicht begrenzt, jedoch existiert ein Polynom, durch das die mittlere Laufzeit begrenzt ist.
Der randomisierte Algorithmus ist also korrekt, kann aber mitunter eine sehr viel längere Laufzeit als im typischen Fall haben.
Man nennt solche Algorithmen Las Vegas Algorithmen.

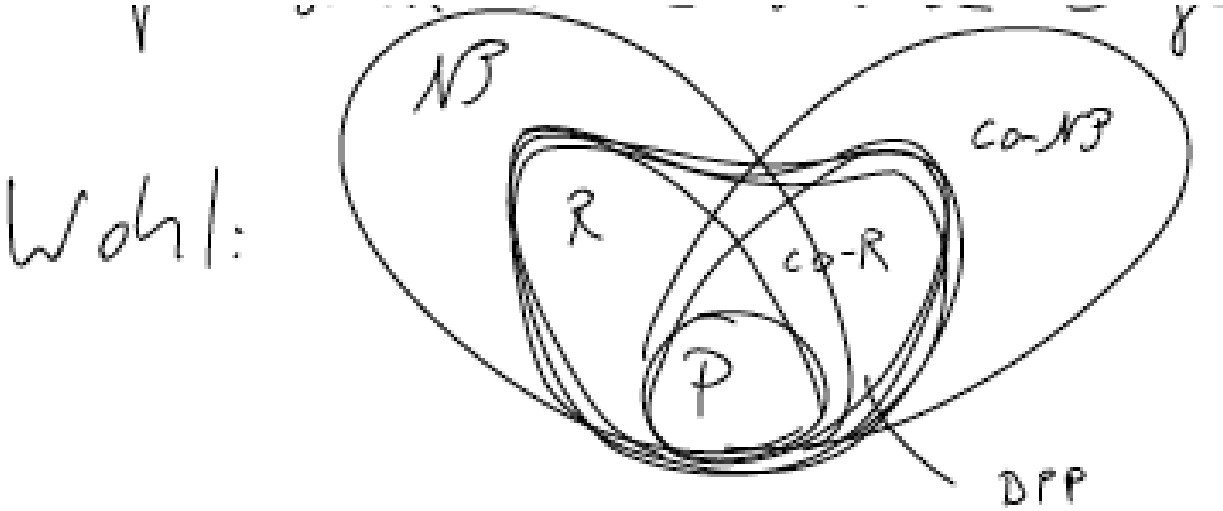
Definition: bounded error probability polynomial time ($\mathcal{BPP}$)

Die Klasse $\mathcal{BPP}$ besteht aus den Entscheidungsproblemen, die durch eine RTM lösbar sind, die mit Wahrscheinlichkeit $1/2 + \delta$ die richtige Antwort gibt.

Definition: bounded error probability polynomial time ($\mathcal{BPP}$)

Die Klasse $\mathcal{BPP}$ besteht aus den Entscheidungsproblemen, die durch eine RTM lösbar sind, die mit Wahrscheinlichkeit $1/2 + \delta$ die richtige Antwort gibt.

Vermutlich sind $\mathcal{BPP}$ und $\mathcal{NP}$ nicht vergleichbar: $\mathcal{BPP} \not\subseteq \mathcal{NP}$ und $\mathcal{BPP} \not\supseteq \mathcal{NP}$



Aufgabe

Sei $0 < \varepsilon_1 < \varepsilon_2 < 1$. Ein Monte-Carlo-Algorithmus liefere für jede Eingabe mit einer Wahrscheinlichkeit von mindestens $1 - \varepsilon_2$ eine korrekte Lösung. Wieviele unabhängige Ausführungen des Algorithmus benötigt man, um für jede Eingabe mit einer Wahrscheinlichkeit von mindestens $1 - \varepsilon_1$ eine korrekte Lösung zu erhalten?

Lösung:

Sei p die Wahrscheinlichkeit, dass die Lösung des Algorithmus korrekt ist. Es gilt also $p \geq 1 - \varepsilon_2$.
Führt man den Algorithmus n-mal unabhängig aus, dann erhält man mit einer Wahrscheinlichkeit $1 - (1-p)^n$ eine korrekte Lösung, d.h. das eine der n Ausführungen des Algorithmus eine korrekte Ausgabe liefert.
Es gilt dann:
$1 - (1-p)^n \geq 1 - (1 - (1 - \varepsilon_2))^n = 1 - \varepsilon_2^n \geq 1 - \varepsilon_1$
$\Leftrightarrow \varepsilon_2^n \leq \varepsilon_1$
$\Leftrightarrow n * log\ \varepsilon_2 \leq log\ \varepsilon_1$
$\Leftrightarrow n \geq \frac{log\ \varepsilon_1}{log\ \varepsilon_2}$ (Ungleichheitswechsel, weil $log(x) < 0$ für $x \in (0,1)$)
Um mit Wahrscheinlichkeit von mindestens $1 - \varepsilon_1$ eine korrekte Lösung zu erhalten, benötigt man also $\geq \frac{log\ \varepsilon_1}{log\ \varepsilon_2}$ Ausführungen.

Aufgabe

Zeigen Sie folgende Inklusionen der Komplexitätsklassen:

i) $P \subseteq R$,

Aufgabe

Zeigen Sie folgende Inklusionen der Komplexitätsklassen:

i) $P \subseteq R$,

ii) $R \subseteq BPP$,

Aufgabe

Zeigen Sie folgende Inklusionen der Komplexitätsklassen:

i) $P \subseteq R$,

ii) $R \subseteq BPP$,

iii) $R \subseteq NP$.

Lösung:

Sei $\mathcal{L} \in P$.
Dann gibt es eine deterministische Turingmaschine M, die $\mathcal{L}$ in polynomialer Zeit fehlerfrei akzeptiert, d.h.

$$Pr[M(\alpha) = 1] = \begin{cases} 1, & \text{falls } \alpha \in \mathcal{L} \\ 0, & \text{sonst} \end{cases}$$

Also ist auch $\mathcal{L} \in R$.

Lösung:

Sei $\mathcal{L} \in R$.
Dann gibt es eine stochastische Turingmaschine M, die $\mathcal{L}$ in polynomialer Zeit akzeptiert, wobei

$$Pr[M(\alpha) = 1] \begin{cases} \geq \frac{1}{2}, & \text{falls } \alpha \in \mathcal{L} \\ = 0, & \text{sonst} \end{cases}$$

Durch mehrfache Aufrufe dieser Turingmaschine kann man, wie in der Vorlesung gezeigt, die Schranke $\frac{1}{2}$ durch eine beliebige Schranke $c \in (0, 1)$ ersetzen.
D.h. es gibt eine stochastische Turingmaschine M', die $\mathcal{L}$ in polynomialer Zeit akzeptiert, wobei

$$Pr[M(\alpha) = 1] \begin{cases} \geq \frac{3}{4}, & \text{falls } \alpha \in \mathcal{L} \\ = 0, & \text{sonst} \end{cases}$$

Lösung:

Sei wieder $\mathcal{L} \in R$.
Dann gibt es eine stochastische Turingmaschine M, die $\mathcal{L}$ in polynomialer Zeit akzeptiert, wobei

$$Pr[M(\alpha) = 1] \begin{cases} \geq \frac{1}{2}, & \text{falls } \alpha \in \mathcal{L} \\ = 0, & \text{sonst} \end{cases}$$

D. h. falls $\alpha \notin \mathcal{L}$, dann gibt es keine Möglichkeit, dass die Turingmaschine M das Wort α akzeptiert, und falls $\alpha \in \mathcal{L}$, dann muss es eine geben. Lässt man die Turingmaschine nichtdeterministisch arbeiten, dann kann sie also in polynomialer Zeit entscheiden, ob $\alpha \in \mathcal{L}$. Also ist auch $\mathcal{L} \in$ NP.

Rabin Kryptosystem

Das Rabin-Kryptosystem ist innerhalb der Kryptologie ein asymmetrisches Kryptosystem, das auf dem Faktorisierungsproblem beruht und mit RSA verwandt ist. Es lässt sich prinzipiell auch zur Signatur verwenden. In der Praxis findet das Verfahren allerdings wegen bestimmter Angriffsmöglichkeiten kaum Anwendung.
Das Verfahren wurde im Januar 1979 von Michael O. Rabin veröffentlicht. Das Rabin-Kryptosystem war das erste asymmetrische Kryptosystem, dessen Sicherheit - das Faktorisierungsproblem als nicht effizient lösbar vorausgesetzt - auf mathematischem Weg bewiesen werden konnte. [1]

[1] wikipedia.de

Das (vereinfachte)[2] Rabin Kryptosystem ist ein asymmetrisches Verschlüssel-ungsverfahren, d. h. es gibt für jeden Teilnehmer einen öffentlichen und eine privaten Schlüssel. Mit dem Öffentlichen verschlüsselt man, mit dem Privaten kann man entschlüsseln. Es hat folgenden Aufbau:

privater Schlüssel:	Primzahlen p und q mit $p \equiv q \equiv 3 \ (mod\ 4)$
öffentlicher Schlüssel:	$n = p * q$
Verschlüsselung:	$f\colon \{0, \ldots, n-1\} \to \{0, \ldots, n-1\}$
	$m \mapsto c = m^2 \ mod\ n$

[2] die Bedingung $p \equiv q \equiv 3 \ mod\ 4$ vereinfacht die Rechnung, ist aber nicht notwendig.

Entschlüsselung: finde mit EEA[3] y_p, y_q mit $y_p * p + y_q * q \equiv 1 \ mod \ n$

$m_p := c^{\frac{p+1}{4}} \ mod \ p$

$m_q := c^{\frac{q+1}{4}} \ mod \ q$

Mit dem chinesischen Restesatz findet man:

$r = (y_p * p * m_q + y_q * q * m_p) \ mod \ n$

$-r \equiv n - r \ mod \ n$

$s = (y_p * p * m_q - y_q * q * m_p) \ mod \ n$

$-s \equiv n - s \ mod \ n$

Für den Klartext m gilt: $m \in \{r, -r, s, -s\}$

[3] EEA bezeichnet den erweiterten euklidischen Algorithmus.

Die Entschlüsselung liefert zusätzlich zum Klartext drei weitere Ergebnisse, das richtige Ergebnis muss daher erraten werden. Dies ist der große Nachteil des Rabin-Kryptosystems.

Der große Vorteil des Rabin-Kryptosystems ist, dass man es nur dann brechen kann, wenn man das beschriebene Faktorisierungsproblem effizient lösen kann.[4]

[4] wikipedia.de

Aufgabe

Gegeben sei der private Schlüssel p=11 und q=19.

1. Wie lautet der öffentliche Schlüssel?

Aufgabe

Gegeben sei der private Schlüssel p=11 und q=19.

1. Wie lautet der öffentliche Schlüssel?
2. Was ist 23 verschlüsselt?

Aufgabe

Gegeben sei der private Schlüssel p=11 und q=19.

1. Wie lautet der öffentliche Schlüssel?
2. Was ist 23 verschlüsselt?
3. Was könnte 100 entschlüsselt sein?

Lösung:

1. $n = p * q = 11 * 19 = \mathbf{209}$

Lösung:

1. $n = p * q = 11 * 19 = \mathbf{209}$
2. $c = m^2 \ mod \ n \Rightarrow c = 23^2 = 529 \equiv \mathbf{111} \ mod \ 209$

Lösung:

1. $n = p * q = 11 * 19 = \mathbf{209}$
2. $c = m^2 \ mod \ n \Rightarrow c = 23^2 = 529 \equiv \mathbf{111} \ mod \ 209$
3. EEA:

$$\begin{aligned}
19 &= 0 \quad * 11 + 1 \quad * 19 & & \\
11 &= 1 \quad * 11 + 0 \quad * 19 & &\mid * (-1) \\
8 &= (-1) * 11 + 1 \quad * 19 & &\mid * (-1) \\
3 &= 2 \quad * 11 + (-1) * 19 & &\mid * (-2) \\
2 &= (-5) * 11 + 3 \quad * 19 & &\mid * (-1) \\
1 &= 7 \quad * 11 + (-4) * 19 & &
\end{aligned}$$

$$\Rightarrow y_p = 7, y_q = -4$$

Lösung:

$$m_p := c^{\frac{p+1}{4}} \ mod \ p$$
$$\Rightarrow m_p = 100^{\frac{11+1}{4}} \equiv 1 \ mod \ 11$$
$$m_q := c^{\frac{q+1}{4}} \ mod \ q$$
$$\Rightarrow m_q = 100^{\frac{19+1}{4}} \equiv 9 \ mod \ 19$$
$$r = (y_p * p * m_q + y_q * q * m_p) \ mod \ n$$
$$\Rightarrow r = (7 * 11 * 9 + (-4) * 19 * 1) \equiv 199 \ mod \ 209$$
$$-r \equiv n - r \ mod \ n$$
$$\Rightarrow -r = 209 - 199 \equiv 10 \ mod \ 209$$
$$s = (y_p * p * m_q - y_q * q * m_p) \ mod \ n$$
$$\Rightarrow r = (7 * 11 * 9 - (-4) * 19 * 1) \equiv 142 \ mod \ 209$$
$$-s \equiv n - s \ mod \ n$$
$$\Rightarrow -s = 209 - 142 \equiv 67$$

Für den Klartext m gilt also:

$$m \in \{r, -r, s, -s\} = \{199, 10, 142, 67\}$$

LESEN!

Handbook of Applied Cryptography (Discrete Mathematics and Its Applications)
`http://www.cacr.math.uwaterloo.ca/hac/`
Bisher Kapitel 8.

Reflexion

Was haben wir heute gelernt?

Reflexion

Was haben wir heute gelernt?

- Randomisierte Komplexitätsklassen kennen gelernt.

Reflexion

Was haben wir heute gelernt?

- Randomisierte Komplexitätsklassen kennen gelernt.
- Einführung in die Kryptographie mit dem Rabin Kryptosystem.

Noch Fragen?

Vorschau

Vorschau

- One-Time-Pad

Vorschau

- One-Time-Pad
- Zero Knowledge

Vorschau

- One-Time-Pad
- Zero Knowledge
- Informationstheorie

Bis zum nächsten Mal