Anmerkungen und Ergänzungen zur Vorlesung: Theoretische Grundlagen der Programmierung

Michael Köhler

28. September 2000

1 Eine Sprachhierarchie

$A \in V_N$, $\alpha, \beta \in V^*$ und $w \in V^+$

Chomsky	Maschine	Sprachfamilie	Bsp.
		überabzählbare	$\mathbb{R}$
		abzählbare	Diagonalsprache L_d
Typ-0: $\alpha A \beta \to \gamma$	TM	$\mathcal{RE}$	Haltesprache
		$\mathcal{REC}$	
Typ-1: $\alpha A \beta \to \alpha w \beta$	LBA	$\mathcal{CS} = \mathcal{MON}$	
Typ-2: $A \to \alpha$	PDA	$\mathcal{CF}$	$\{ww^{rev}\ w \in X\}$
	PDA	$\mathcal{DCF}$	$\{a^n b^n\ n \in \mathbb{N}\}$
Typ-3: $A \to \alpha B$	FA	$\mathcal{REG}$	a^*

1. Gibt es eine Sprache, die abzählbar aber nicht aufzählbar ist?

2. Gibt es eine Sprache, die aufzählbar aber nicht entscheidbar ist?

3. Definiere die Menge der rekursiv aufzählbaren, der semi-entscheidbaren, der rekursiven sowie der entscheidbaren Sprachen!

4. Welcher Zusammenhang besteht zwischen den Konzepten: rekursive Funktionen, Turingmaschinen, Lambdakalkül?

5. Welcher Zusammenhang besteht zwischen rekursiv aufzählbar und semi-entscheidbar? Welcher zwischen rekursiv und entscheidbar? Woher stammen die Begriffe?

6. Wieso bilden die entscheidbaren Mengen eine boolsche Algebra?

7. Schreibe die (formale) Definition auf: TM, UTM, RAM, Bit-RAM, arithmetische RAM, RAM+ und RASP!

8. Definieren Sie uniformes und logarithmisches Zeitmaß und erläutern sie die Beziehung und Bedeutung?

9. Geben Sie einige Maschinenbefehle der RAM an?

10. Warum ist uniformes Platzmaß für RAMs ungeeignet?

11. Was ist der Unterschied zwischen den Registern der RAM und den Bandzellen der TM?

12. Welche Platzbeschränkung besitzt eine $t(n)$-zeitbeschränkte TM?

    Theorem 1 (1. Maschinenklasse) *Es existiert eine Klasse von Maschinen, die sich gegenseitig in Polynomzeit und konstanten Platz simulieren können, z.B. TM, RAMs und RASPs in logarithmischen Maß, RAM+ in uniformen Maß.*

13. TM und RAM können sich wechselseitig simulieren. Sind sie daher gleichmächtig? Mit welchem Aufwand geschieht die Simulation und was bestimmt in maßgeblich?

14. Wieso ist das Halteproblem nicht entscheidbar?

15. Was beschreibt die „number of wisdom“?

16. Ist es entschiedbar, ob eine zeitbeschränkte (platzbeschränkte) TM hält? Warum?

17. Was ist eine nicht-triviale Eigenschaft? Was sagt der Satz von Rice über sie aus?

18. Welche Fragestellung beschreibt das 10. Hilbertsche Problem? Welche Zusammenhang besteht zur diophantischen Relationen?

19. Welche Mächtigkeit besitzt ein Zwei-Zählerautomat? Welche ein Ein-Zählerautomat? Ist ein Zwei-Kellerautomat mächtiger als ein Zwei-Zählerautomat?

20. Was beschreibt die Kolmogorov-Komplexität?

21. Wie ist eine primitiv-rekursiven Funktionen definiert? Kann jede rekursive Funktion im Schema der primitiv-rekursiven Funktionen definiert werden?

22. Welche Bedeutung hat die Ackermann-Funktion?

23. Wer oder was ist eigentlich Gödel? Was ist an ihm so besonders?

2 Weitere Automatenklassen

1. Definiere: ω-Wort, ω-reguläre Ausdrücke, Büchi-Automat, Muller-Automat!

2. Definiere: Probabilistischer Automat $(0; 1/2; 1)$, Zweiweg-Automat, Mehrkopf-Automat!

3. Kann der Zweiwegautomat mehr als der endliche? Und der Mehrkopfautomat?

4. Definiere das Postsche Korrspondenzproblem (PCP)!

5. Erkläre: Kongruenzrelation, Transitionsmonoid, Satz von Myhill!

6. Wieso folgt aus dem folgenden Theorem die besondere Bedeutung der Dyck-Sprachen D_k (Dyck-Sprachen sind die Mengen aller korrekt geklammerter Wörter über k Klammersymbolen)?

   Theorem 2 (Chomsky-Schürzenberger) *Zu jeder kontextfreien Sprache $L \in \mathcal{CF}$ existiert ein k, eine reguläre Menge R_L und ein Homomorphismus h_L, so daß gilt:*

   $$L = h_L(D_k \cap R_L)$$

7. Welche Sprachklassen der Chomskyhierarchie sind abgeschlossen gegen Homorphismus, Komplement, Schnitt und/oder Vereinigung? Geben Sie ein paar Beispiele an!

Definition 1 *Betrachte den Automaten $A = (Z, X, Y, K, q_0, F)$.*
Ist K von der Form $K \subseteq Z \times X^* \times Y^* \times Z$, so ist A ein a-transducer.
Ist K von der Form $K \subseteq Z \times X \times Y^* \times Z$ und ergibt K eingeschränkt auf $Z \times X \times Z$ einen vollständigen DFA, so ist A eine general-state-machine.
Ist K von der Form $K \subseteq Z \times X \times Y \times Z$, so ist A ein Mealy-Automat.

3 Parallele Programme

1. Erkläre: PRAM, EREW, CRCW(com, arb, pri), zellularer Automat, Schaltkreisfamilien!
2. Nennen Sie Beziehungen zwischen den Modellen EREW, CRCW(com, arb, pri)!
3. In welcher Zeit kann eine CRCW^{com} eine Oder-Funktion über n Argumenten berechnen?
4. Was ist die Prozessorkomplexität einer PRAM?
5. In welcher Beziehung stehen der zellulare Automat und die TM? Für welche Rechner und welche Probleme eignet sich ein ZA?
6. Wann heißt eine Schaltkreisfamilie uniform?
7. Uniforme Schaltkreisfamilien entsprechen ungefähr einer PRAM. Warum?

Theorem 3 (Brent's scheduling principle) *Benötigen m Prozessoren die Zeit t, so benötigen $p \leq m$ Prozessoren die Zeit $t + \lfloor \frac{x}{p} \rfloor$.*

Beweis:

$$\text{Mit } \lceil \frac{x_i}{p} \rceil \leq \frac{x_i}{p} + 1 \text{ gilt } \sum_1^t \lceil \frac{x_i}{p} \rceil = \lfloor \sum_1^t \lceil \frac{x_i}{p} \rceil \rfloor \leq \lfloor \sum_1^t (\frac{x_i}{p} + 1) \rfloor = t + \lfloor \frac{x}{p} \rfloor$$

qed.

Theorem 4 (Optimale Beschleunigung) *Benötigt für eine Problem* 1 *Prozessor $O(n)$ Zeit und n Prozessoren $O(\log n)$, so kann es auch von $O(\frac{n}{\log n})$ Prozessoren in $O(\log n)$ Zeit. gelöst werden*

Beweis: Dies folgt direkt aus Brent's scheduling principle:

$$t + \lfloor \frac{x}{p} \rfloor = O(\log n) + \lfloor \frac{O(n)}{O(\frac{n}{\log n})} \rfloor = O(\log n) + \lfloor O(\log n) \rfloor = O(\log n)$$

qed.

1. Nennen Sie Probleme, die von einem Prozessor in $O(n)$ Zeit und von n Prozessoren in $O(logn)$ Zeit gelöst werden können!
2. Beschreiben Sie, wie schnell die Maximumssuche mit einem, mit n oder mit n^2 Prozessoren möglich ist!
3. Welche Produktkomplexitäten ergeben sich aus Theorem 4? Wieso heißt es optimale Beschleunigung?

Theorem 5 *Seinen T, P beliebige Fkt. und k eine Fkt., die auf einer RAM+ in der Zeit T berechenbar ist.*

$$MTimeProc(T, P) \subset MTimeProc(kO(T), \lceil \frac{P}{k} \rceil)$$

1. Zeigen Sie, daß das folgende Theorem eine direkte Folgerung aus Theorem 5 ist!

   Theorem 6
   $$CRCW\text{-}TimeProc(Pol, Pol) = \mathcal{P}$$

   In welcher Beziehung stehen die beiden folgenden Theoreme?

Theorem 7
$$CRCW\text{-}Time(Pol) = \mathcal{PSPACE}$$

Theorem 8 (2. Maschinenklasse) *Es existiert eine Klasse von (parallelen) Maschinen, für die Polynomzeit genauso mächtig ist, wie Polynomplatz für Maschinen der 1. Klasse.*

4 Kommunikations-Netzwerke

1. Definiere die Begriffe Abstand, Durchmesser $D(G)$, Bisektionsweite $B(G)$, Layout area $A(G)$.
2. Beschreibe die Topologien Ring, Array, Toroid, Tree, Mesh of trees! Beschreibe Shuffle exchange SE_n, De Bruijn / Perfekt-schuffle DB_{n_1}

3. Beachte, daß jeder Shuffle exchange Graph SE_n ein De Bruijn / Perfektschuffle DB_{n_1} einer niedrigen Stufe ist! Warum?

4. Beschreibe Butterfly, cube connected cycle (CCC), Hypercube! Wieso ist ein Butterfly so gut für „Teile-und-Herrsche“ Algorithmen geeignet?

5. Was ist ein normaler Butterfly Algorithmus. Wieso läßt er sich so gut auf Hypercubes implementieren? Und warum ist das von Interesse?

6. Erkläre: Permutationsnetzwerk, Benes-Netzwerk.

7. Wieviel Schalter hat ein optimales Permutationsnetzwerk?

8. Wie bettet man Bäume auf dem Hypercube ein? Warum will man das?

9. Welche Bedeutung hat die Aussage $B(SE_m) = \Theta(\frac{2^m}{m})$ für die Performance?

4.1 Implementation der PRAM auf dem Hypercube

Hypercubes sind technisch zu realisieren, PRAMS einfach zu analysieren. Die Frage liegt daher nahe, ob man einen fundierten CREW-PRAM-Algorithmus effizient auf dem Hypercube optimieren kann. Mann kann: ein PRAM-Schritt von $\log N$ Prozessoren wird von einem Hypercube in $\log N$ Schritten bearbeitet, so daß die Prozessorzeitkomplexität gleich ist.

Theorem 9 *Ein Algorithmus, der in $T \log N$ Zeit auf $N \log N$ Prozessoren auf der PRAM läuft, arbeitet in $O(T)$ Zeit mit N Prozessoren auf dem Hypercube.*

Wichtige Teilprobleme bei der Simulation ist das gleichzeitige Lesen. Notwendig ist es das probabilistische Routing auf Hypercubes sowie die Datendistribution in logarithmischer Zeit zu schaffen, d.h in einem simulierten PRAM-Schritt..

5 Analyse von Algorithmen

5.1 Die O-, Ω- und Θ-Notation

Definitionsgemäß existieren zu einer Funktion f die Mengen $O(f)$, $\Omega(f)$ sowie $\Theta(f)$:

$$\begin{aligned}
f \in O(g) \quad &:\Longleftrightarrow \quad \exists c : |f(n)| \leq c|g(n)| \\
f \in \Omega(g) \quad &:\Longleftrightarrow \quad \exists c : c|g(n)| \leq |f(n)| \\
f \in \Theta(g) \quad &:\Longleftrightarrow \quad \exists c_1, c_2 : c_1|g(n)| \leq |f(n)| \leq c_2|g(n)| \\
&\Longleftrightarrow \quad f \in O(g) \wedge f \in \Omega(g) \\
f \in o(g) \quad &:\Longleftrightarrow \quad \forall \epsilon \exists n_0 \forall n > n_0 : |f(n)/g(n)| \leq \epsilon
\end{aligned}$$

Alternativ existieren folgende Relationen, die eine ähnliche Intention besitzen:

$$\begin{aligned}
f \prec g \quad &:\Longleftrightarrow \quad \lim_{n\to\infty} \frac{f(n)}{g(n)} = 0 \\
f \sim g \quad &:\Longleftrightarrow \quad \lim_{n\to\infty} \frac{f(n)}{g(n)} = 1 \\
f \succ g \quad &:\Longleftrightarrow \quad \lim_{n\to\infty} \frac{g(n)}{f(n)} = 0 \\
f \asymp g \quad &:\Longleftrightarrow \quad \exists c : |f(n)| \leq c|g(n)| \wedge |g(n)| \leq c|f(n)|
\end{aligned}$$

1. Beschreiben Sie die Bedeutung der O-, Ω- und Θ-Notation!
2. Gilt $\log x \leq x^n \leq n^x$ für alle x und n?
3. Wieso folgen aus den Definitionen unmittelbar folgende Beziehungen?

$$\begin{aligned}
f \in O(g) \quad &\Longleftrightarrow \quad g \in \Omega(f) \\
f \in \Theta(g) \quad &\Longleftrightarrow \quad g \in \Theta(f)
\end{aligned}$$

4. Zeige, daß beide Notationen einander entsprechen!

$$\begin{aligned}
f \in O(g) \quad &\Longleftrightarrow \quad g \in \Omega(f) \\
f \in \Theta(g) \quad &\Longleftrightarrow \quad g \in \Theta(f) \\
f \asymp g \quad &\Longleftrightarrow \quad f \in \Theta(g) \\
f \prec g \quad &\Longleftrightarrow \quad f \in o(g)
\end{aligned}$$

Leider bildet das Tripel $(\prec, \sim, \succ)$ keine vollständige Ordnung: Betrachte z.B. $f(n) = 2n$ und $g(n) = n$, so gilt $\lim f(n) = g(n) = 2$. Ersetzen wir die Relation $\sim$ durch $\asymp$, so gilt jedoch das folgende Theorem:

Theorem 10 *Für Funktionen f, g aus der Menge der* logarithmisch-exponentiellen Funktionen *gilt genau eine der Relationen:*

$$f \prec g, \quad f \asymp g, \quad f \succ g$$

Exkurs Folgendes wird nur für Neugierige erwähnt.

Theorem 11 *Falls $f(n), g(n) > 0$ für alle $n > 0$ gilt, dann ist*

$$\begin{aligned}
\lim_{n\to\infty} \frac{f(n)}{g(n)} = a \neq 0 \quad &\Longrightarrow \quad f \in \Theta(g) \\
\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0 \quad &\Longrightarrow \quad f \in O(g) \\
&\Longrightarrow \quad f \in o(g) \\
&\Longrightarrow \quad f \notin \Theta(g) \\
\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty \quad &\Longrightarrow \quad g \in O(f) \\
&\Longrightarrow \quad g \notin \Theta(f)
\end{aligned}$$

Insbesondere folgt aus der ersten Gleichung, daß:

$$f \sim g \Longrightarrow f \in \Theta(g)$$

Beweis:

$$\begin{aligned} \forall \epsilon \exists n_0 \forall n > n_0 : \left| \frac{f(n)}{g(n)} - a \right| \le \epsilon \quad &\Longrightarrow \quad (a-\epsilon)g(n) \le f(n) \le (a+\epsilon)g(n) \\ &\Longrightarrow \quad f \in \Theta(g) \end{aligned}$$

$$\begin{aligned} \forall \epsilon \exists n_0 \forall n > n_0 : f(n) \le \epsilon g(n) \quad &\Longrightarrow \quad f \in O(g), f \in o(g) \\ &\Longrightarrow \quad \nexists c : f(n) \ge cg(n) \Longrightarrow f \notin \Theta(g) \end{aligned}$$

$$\begin{aligned} \forall C \exists n_0 \forall n > n_0 : f(n) \ge Cg(n) \quad &\Longrightarrow \quad f \in O(g), f \in o(g) \\ &\Longrightarrow \quad \nexists c : f(n) \ge cg(n) \Longrightarrow f \notin \Theta(g) \end{aligned}$$

Theorem 12

$$f \sim g \iff f = g + o(g)$$

Beweis:

- $f \sim g \Longrightarrow f = g + o(g)$:

$$\begin{aligned} f \sim g \quad &\iff \quad \lim_{n\to\infty} \frac{f(n)}{g(n)} = 1 \\ &\iff \quad \forall \epsilon \exists n_0 : \forall n > n_0 \left| \frac{f(n)}{g(n)} \right| \le \epsilon \\ &\iff \quad |f(n) - g(n)| \le \epsilon g(n) \end{aligned}$$

Sei $h \in o(g)$, dann ist $|h(n)| \le \epsilon |g(n)|$, also $|h(n)| = |f(n) - g(n)|$.

- $f \sim g \Leftarrow f = g + o(g)$ Sei $h \in o(g)$:

$$\frac{f(n)}{g(n)} = \frac{g(n) + o(g)}{g(n)} = 1 + \frac{o(g)}{g(n)} = 1 + \frac{h(n)}{g(n)}$$

$$\forall \epsilon \exists n0 \forall n > n_0 : 1 - \epsilon \le 1 - \left| \frac{h(n)}{g(n)} \right| \le 1 - \frac{|h(n)|}{|g(n)|} \le 1 + \epsilon$$

Hiermit gilt folgende Abschätzung:

$$\begin{aligned} &\forall \epsilon \exists n0 \forall n > n_0 : \left| \frac{f(n)}{g(n)} - 1 \right| \le \epsilon \\ \iff \quad &\lim_{n\to\infty} \frac{f(n)}{g(n)} = 1 \iff f \sim g \end{aligned}$$

qed.

5.2 Rekurrenzgleichungen

1. Welche Bedeutung haben die erzeugenden Funktionen bei der Lösung von Rekurrenzgleichungen? Warum führt man soviel Mathematik ein? Macht man sich das Leben nicht nur unnötig kompliziert?

2. Was sind Stirling-Zahlen und was Binomialkoeffizienten? Wie ist das Wachstum der harmonischen Zahl? Wie ist das Wachstum der Fakultätsfunktion $n!$?

3. Wann ist eine Rekurrenzgleichung linear?

4. Was ist ein Spiegelpolynom q^R?

5. Gebe AB für $A = \sum_{n\geq 0} a_n z^n$ und $B = \sum_{n\geq 0} b_n z^n$ an! Welche Bedeutung haben die Diagonalsummen dabei?

6. Geben Sie die erzeugenden Funktionen und ihre geschlossene Form für folgende Sequenzen an!

$$f(n) = 1,\ f(n) = \binom{c}{k},\ f(n) = 1/n! \text{ und } f(n) = 1/n$$

Sind die α_i die Nullstellen von q^R, so kann man $q(z)$ umschreiben:

$$\begin{aligned} q(z) &= z^d q^R(\frac{1}{z}) = z^d(\frac{1}{z} - \alpha_1)^{d_1} \cdots (\frac{1}{z} - \alpha_k)^{d_k} \\ &= (1 - \alpha_1 z)^{d_1} \cdots (1 - \alpha_k z)^{d_k} \end{aligned}$$

Rekurrenzgleichungen und erzeugende Funktionen hängen wie folgt zusammen:

Theorem 13 *Sei $q_1, \ldots, q_d$ eine feste folge komplexer Zahlen ($d \geq 1$ und $q_d \neq 0$) und $q(z)$ das Polynom*

$$q(z) = 1 + q_1 z + \cdots + q_d z^d = (1 - \alpha_1 z)^{d_1} \cdots (1 - \alpha_k z)^{d_k}.$$

Dabei sind die α_i die Nullstellen des Spiegelpolynoms $q^R(z)$ in der Vielfachheit D_i. Dann sind folgende Beschreibungen äquivalent:

1. *Rekursion der Länge d.*

$$f(n+d) + q_1 f(n+d-1) + \cdots + q_d f(n) = 0$$

2. *Erzeugende Funktion.*

$$F(z) = \sum_{n\geq 0} f(n) z^n = \frac{p(z)}{q(z)},$$

wobei $p(z)$ ein Polynom vom Grad $< d$ ist.

3. Partialbruchzerlegung.

$$f(z) = \sum_{n \geq 0} f(n) z^n = \sum_{i=1}^{k} \frac{g_i(z)}{(1 - \alpha_i z)^{d_i}}$$

für Polynome $g_i(z)$ vom Grad $< d_i$

4. Explizite Darstellung.

$$f(n) = \sum_{i=1}^{k} p_i(n) \alpha_i^n$$

für Polynome $p_i(z)$ vom Grad $< d_i$

Beispiel Vorgehen zum Lösen einer Rekurrenz, die als Rekurrenzgleichung (1. Form) vorliegt, z.B. die Fibounacci-Zahlen:

$$f(n+2) = f(n+1) + f(n), f(1) = 1, f(0) = 0$$

Dies ist eine Rekursion der Länge $d = 2$:

$$f(n+2) - f(n+1) - f(n) = 0$$

Interessiert sind wir an der expliziten Darstellung (3.Form) $f(n) = \sum_{i=1}^{k} p_i(n) \alpha_i^n$, wir benötigen also zunächst die Nullstellen des Spiegelpolynoms, wobei gilt:

$$q(z) = 1 - 1z - 1z^2 \qquad q^R(z) = 1z^2 - 1z - 1$$

Das Spiegelpolynom besitzt die Nullstellen $\alpha_{1,2} = \frac{1 \pm \sqrt{5}}{2}$ – daraus folgt durch Einsetzen

$$f(n) = \sum_{i=1}^{k} p_i(n) \alpha_i^n = p_1(n) \alpha_1^n + p_2(n) \alpha_2^n$$

Die p_i sind Polynome vom Grad 1. Sie berechnen sich mit Hilfe des folgenden Theorems:

Theorem 14 *Sei $q(z)$ ein Polynom mit nur einfachen Nullstellen $q(z) = (1 - \alpha_1 z) \cdots (1 - \alpha_d z)$, dann besitzen die Koeffizienten a_i der explizten Darstellung den Wert:*

$$a_i = \frac{-\alpha_i p(1/\alpha_i)}{q'(1/\alpha_i)}$$

Wie ermittelt man nun das Polynom p? Es ist in der zweiten Form gegeben und berechnet sich nach folgendem Schema:

$$\begin{aligned}
f(n) &= f(n-1)+f(n-2)+[n=1]\\
z^n f(n) &= z^n f(n-1)+z^n f(n-2)+z^n[n=1]\\
\sum_{n\geq 0} z^n f(n) &= \sum_{n\geq 0} z^n f(n-1)+\sum_{n\geq 0} z^n f(n-2)+\sum_{n\geq 0} z^n[n=1]\\
\sum_{n\geq 0} z^n f(n) &= z\sum_{n\geq 0} z^{n-1} f(n-1)+z^2\sum_{n\geq 0} z^{n-2} f(n-2)+z\\
F(n) &= zF(n)+z^2F(n)+z\\
(1-z-z^2)F(n) &= z\\
F(n) &= \frac{z}{1-z-z^2}
\end{aligned}$$

Also ist $p(z)=z$. Außerdem ist

$$q(z)=1-1z-1z^2 \qquad q'(z)=-1-2z,$$

was eingesetzt ergibt:

$$a_i=\frac{-\alpha_i p(1/\alpha_i)}{q'(1/\alpha_i)}=\frac{-\alpha_i\frac{1}{\alpha_i}}{1-(2/\alpha_i)}=\frac{\alpha_i}{\alpha_i-2}$$

$$a_{1,2}=\frac{\frac{1\pm\sqrt{5}}{2}}{\frac{1\pm\sqrt{5}}{2}+2}=\frac{1\pm\sqrt{5}}{5\pm\sqrt{5}}=\frac{\pm4\sqrt{5}}{20}=\frac{\pm1}{\sqrt{5}}$$

Also noch einmal Einsetzen – fertig!

$$f(n)=p_1(n)\alpha_1^n+p_2(n)\alpha_2^n=\frac{(\frac{1+\sqrt{5}}{2})^n}{\sqrt{5}}+\frac{-(\frac{1-\sqrt{5}}{2})^n}{\sqrt{5}}$$

1. Wovon hängt die Laufzeit von „Teile ind herrsche“ Algorithmen ab? Welche asymptotischen Laufzeiten ergeben sich?

6 Komplexitätstheorie

Wieviel mehr Platz (oder alternativ: Zeit) muß man investieren, um eine neue Sprache erkennen zu können?

Theorem 15 *Sind $s_1, s_2 \geq \log$ und beide band-konstruierbar und gilt*

$$\inf_{n\to\infty}\frac{s_1(n)}{s_2(n)}=0,$$

dann existiert eine Sprache L mit $L\notin DBAND(s_1)$ aber $L\in DBAND(s_2)$.

Theorem 16 *Sind t_1, t_2 zeit-konstruierbar und gilt*

$$\inf_{n\to\infty}\frac{t_1(n)\log t_1(n)}{t_2(n)}=0,$$

dann existiert eine Sprache L mit $L\notin DZEIT(t_1)$ aber $L\in DZEIT(t_2)$.

1. Sind zeit- oder platzbeschränkte Probleme entscheidbar? Gilt $\mathcal{P} \subseteq \mathcal{REC}$? Gilt $\mathcal{NP} \subseteq \mathcal{REC}$?

2. Wann ist ein Problem polynomzeit-, logspace-reduzierbar?

3. Unter welchen Bedingungen ist ein Problem schwer oder vollständig für die Mengen $\mathcal{P}$, $\mathcal{NP}$ und $\mathcal{PPACE}$? Und bzgl. welcher Reduktion gilt dies?

4. Gilt $\mathcal{CF} \subseteq \mathcal{P}$? Warum? Wo liegen $\mathcal{P}$, $\mathcal{NP}$ und $\mathcal{PSPACE}$ in der Chomsky Hierarchie?

5. Was ist ein Orakel?

6. Erläutern Sie die folgenden Beispiele für $\mathcal{P}$-vollständige Probleme: $L(CFG) = \emptyset$?, 2-SAT, RLP, Euler-cycle!

7. Erläutern Sie die folgenden Beispiele für$\mathcal{NP}$-vollständige Probleme: SAT, KNF, 3-KNF, Knapsack, GLP, Hamilton-cycle, das Beschränktheitproblem für EFC-Netze, das (strukturelle) Liveness-Problem für EFC-Netze sowie das Erreichbarkeitsproblem für EFC-Netze, zyklenfreie Netze und BPP-Netze ($|^{\circ}t| = 1$)!

8. Erläutern Sie die folgenden Beispiele für $\mathcal{PSPACE}$-vollständige Probleme: $L(NFA_1) = L(NFA_1)$, $w \in L(NLBA)$ sowie das Erreichbarkeitsproblem für 1-konservative Netze und 1-sichere Netze!

9. Was bedeutet die Vermutung von Hartmanis?

   Theorem 17 (Vermutung von Hartmanis) *Alle $\mathcal{NP}$-vollständigen Probleme sind polynomzeit-isomorph.*

10. Warum impliziert der Satz von Savitch $\mathcal{PSPACE} = \mathcal{NPSPACE}$?

    Theorem 18 (Satz von Savitch)

    $NSpace(s) \subseteq DSpace(s^2)$ für eine platzkonstruierbare Fkt. $s(n) \geq \log n$

6.1 Komplexitätshierarchie

Es gelten folgende Inklusionen:

$$\mathcal{L} \subseteq \mathcal{NL} \subseteq \mathcal{NC} \subseteq \mathcal{P} \subseteq \mathcal{R} \subseteq \mathcal{NP} \subseteq \mathcal{PP} \subseteq \mathcal{PSPACE},$$

wobei

$\mathcal{PP}$: Menge aller auf probabilistischen TM in Polynomzeit erkannten Probleme

$\mathcal{R}$: Menge aller auf probabilistischen TM in Polynomzeit erkannten Probleme, wobei $P(\text{Fehler}) = \begin{cases} 0, & w \notin L \\ \leq \frac{1}{2}, & w \in L \end{cases}$

$$\begin{aligned} NC &= PRAM - Time(\log^{O(1)}(n); n^{O(1)}) \\ &= Circuit - Depth - Time(log^{O(1)}(n); n^{O(1)}) \end{aligned}$$

1. Wieso gilt $\mathcal{NC} \subseteq \mathcal{P}$?

2. Kann ein Petrinetz einen Zähler simulieren?

3. Das Erreichbarkeitsproblem benötigt mindestens $O(2^{O(n)})$ viel Platz für S/T-Netze. Wieso folgt daraus, daß TM mächtiger sind Petrinetze?

7 Kryptographie

1. Erkläre den Unterschied zwischen einem symmetrischen und einem asymmetrische Verfahren!

2. Erklären Sie das Prinzip der Public-key-Verschlüsselung!

3. Erklären Sie das Prinzip des Cesar-Cipher, von RSA und der Knapsack-Verschlüsselung!

4. Bei RSA ist der Klartext aus dem Verschlüsselungstext algorithmisch bestimmbar. Wieso gilt dann RSA als sicher? Wann ist eine System sicher?

5. Welche Beziehungen haben One-way-Funktionen zur Frage, ob $\mathcal{P} = \mathcal{NP}$ gilt?

6. Ist das Knapsack-Verfahren sicher?

7. Beruht die Knapsack-Verschlüsselung auf einem $\mathcal{NP}$-vollständigen Problem? Und RSA?

8. Welche Auswirkungen wird es auf die public-key-Verschlüsselung haben, wenn eine $\mathcal{NP}$-vollständige Sprache $L \in \mathcal{P}$ gefunden wird?