Grundwissen theoretische Informatik

Eine Wiederholung mit Lösungsansätzen *

3. März 2005

1. **Was sind formale Probleme?**
   Relationen zwischen Eingaben aus Σ^* und Ausgaben aus Σ^*, wobei (x,y) $\in$ R, falls y zulässige Ausgabe zur Eingabe x ist. Insbesondere sind also auch Funktionen und Sprachen formale Probleme.

2. **Was heißt berechenbar?**
   Eine Funktion f heißt berechenbar, wenn es eine TM gibt, die zu Eingabe x den Funktionswert f(x) ausgibt.

3. **Was ist eine RAM?**
   Eine Registermaschine (englisch: random access machine) besteht aus einem Befehlszähler, einem Akkumulator-Register und einem Speicher aus (unbeschränkt vielen) Registern. Sie arbeitet auf natürlichen Zahlen und kann mit einem speziellen Befehlssatz (unter anderem ADD-, LOAD-, STORE- und GOTO-Befehle) programmiert werden.

4. **Wie lautet die korrekte Definition der Turingmaschine (7-Tupel)?**
   (Q, Σ, Γ, B, q_0, δ, F) mit

   - Q eine endliche Menge von Zuständen
   - Σ ein endliches Eingabealphabet
   - $\Gamma \supseteq \Sigma$ ein endliches Ausgabealphabet
   - B $\in \Gamma \backslash \Sigma$ das Blanksymbol
   - $q_0 \in$ Q der Startzustand
   - $\delta : Q \times \Sigma \rightarrow Q \times \Gamma \times \{R, L, N\}$ die Überführungsfunktion
   - F $\subseteq$ Q die Menge akzeptierender Endzustände

5. **Wann heißt eine Sprache rekursiv/entscheidbar?**
   Eine Sprache L über einem Alphabet Σ ist rekursiv, wenn es eine Turingmaschine gibt, die auf allen Eingaben anhält und genau die Eingaben aus L akzeptiert.

6. **Wann heißt eine Sprache rekursiv aufzählbar?**
   Eine Sprache L über einem Alphabet Σ ist rekursiv aufzählbar, wenn es eine Turingmaschine gibt, die genau die Eingaben aus L akzeptiert.

7. **Was ist semi-entscheidbar?**
   rekursiv aufzählbar

8. **Wann ist eine semi-entscheidbare Sprache entscheidbar?**
   Genau dann, wenn auch ihr Komplement semi-entscheidbar ist.

*von A. Kimmig und M. Exner

9. **In welchen mengentheoretischen Eigenschaften unterscheiden sich rekursive Sprachen und rekursiv aufzählbare Sprachen?**
   Die rekursiven Sprachen sind unter Komplementbildung abgeschlossen, die rekursiv aufzählbaren nicht (vgl. Punkt 8). Jede der beiden Sprachklassen ist unter Schnitt und Vereinigung abgeschlossen.

10. **Was ist eine universelle Turingmaschine?**
    Eine TM, die den ersten Teil ihrer Eingabe als Codierung einer TM auffasst und diese codierte Maschine auf dem zweiten Teil ihrer Eingabe simuliert.

11. **Wie lautet die Churchsche These?**
    Die durch die formale Definition der Turing-Berechenbarkeit erfasste Klasse von Funktionen stimmt genau mit der Klasse der im intuitiven Sinn berechenbaren Funktionen überein.

12. **Sind RAMs und TMs äquivalent?**
    Ja, jede der beiden Maschinenarten kann durch die jeweils andere simuliert werden.

13. **Was versteht man unter Determinismus und Nicht-Determinismus?**
    Determinismus bedeutet, daß in jeder Situation genau festgelegt ist, was im nächsten Schritt passiert, während Nichtdeterminismus mehrere Möglichkeiten erlaubt. Beispiel: bei deterministischen TMs ordnet die Übergangsfunktion δ jedem Zustand-Zeichen-Paar ein Zustand-Zeichen-Bewegung-Tripel zu, bei einer nichtdeterministischen TM eine Menge solcher Tripel. (Anmerkung: die unter Punkt 4 definierte TM ist deterministisch!)

14. **Was sind Gödelnummern?**
    Die Gödelnummer $\langle M \rangle$ einer TM M ist eine Codierung der Maschine über dem Alphabet $\{0,1\}$.

15. **Wie lautet das Halteproblem? Warum ist es wichtig?**
    $H := \{\langle M \rangle w | M \text{ hält auf } w\}$
    Zum Einen ist die Frage, ob ein Programm terminiert, von praktischem Interesse, zum Anderen ermöglicht uns das Halteproblem Aussagen über die (Un-)Entscheidbarkeit anderer Probleme (vgl. Punkt 24).

16. **Welche Varianten des Halteproblems gibt es?**
    Das Halteproblem auf leerem Band ($H_\epsilon := \{\langle M \rangle | M \text{ hält auf leerem Band}\}$), das spezielle Halteproblem oder Selbstanwendungsproblem ($K := \{\langle M \rangle | M \text{ hält auf } \langle M \rangle\}$), $H_{all} := \{\langle M \rangle | M \text{ hält auf allen Eingaben}\}$, ...

17. **Ist das Halteproblem rekursiv aufzählbar? Ist es entscheidbar?**
    H ist rekursiv aufzählbar, aber nicht entscheidbar, denn wir können nach jedem Rechenschritt feststellen, ob M angehalten hat, aber falls M im aktuellen Schritt nicht angehalten hat, können wir nicht sagen, ob M in irgendeinem zukünftigen Schritt anhalten wird und müssen daher den nächsten Schritt betrachten...

18. **Was ist das Komplement des Halteproblems? Ist es rekursiv aufzählbar?**
    $\overline{H} := \{\langle M \rangle w | M \text{ hält nicht auf } w\}$ ist nicht rekursiv aufzählbar. (Warum? Weil sich andernfalls ein Widerspruch aus Punkt 8 und Punkt 17 ergeben würde!)

19. **Was ist die Diagonalsprache?**
    Die Diagonalsprache ist eine Sprache, von der wir direkt zeigen können, daß sie nicht rekursiv ist. Sie ist folgendermaßen definiert: Sei M_i die TM, deren

Gödelnummer in der kanonischen Ordnung aller Gödelnummern an Position i steht, w_i das i-te Wort in der kanonischen Ordnung auf $\{0{,}1\}^*$. Dann ist D die Menge aller Wörter w_i, so daß M_i w_i nicht akzeptiert.

20. **Hat Endlichkeit einer Sprache etwas mit Entscheidbarkeit zu tun?**
    Ja! Endliche Sprachen sind entscheidbar, denn wir können ein gegebenes Wort in kanonischer Reihenfolge mit allen Wörtern der endlichen Sprache vergleichen und damit in endlich vielen Schritten eine Entscheidung treffen.

21. **Wie lautet der Satz von Rice?**
    Sei R die Klasse aller turingberechenbaren Funktionen und S eine echte, nichtleere Teilmenge von R. Dann ist $L(S) := \{\langle M\rangle | M$ berechnet eine Funktion aus S$\}$ unentscheidbar.

22. **Was ist das PKP?**
    gegeben: eine endliche Folge von Wortpaaren $(x_1, y_1), (x_2, y_2), ..., (x_k, y_k)$ mit $x_i, y_i \in \Sigma^+$ (zu beliebigem Alphabet Σ)
    gefragt: Gibt es eine Folge von Indizes $i_1, i_2, ..., i_n \in \{1, ..., k\}$ mit $n \geq 1$ und $x_{i_1}x_{i_2}...x_{i_n} = y_{i_1}y_{i_2}...y_{i_n}$?

23. **Ist das PKP rekursiv?**
    Nein, es ist nur semi-entscheidbar.

24. **Welche Methoden gibt es, die Un-/Entscheidbarkeit eines Problems zu zeigen?**
    Entscheidbarkeit: Algorithmus angeben.
    Unentscheidbarkeit: Um zu zeigen, daß ein gegebenes Problem A unentscheidbar ist, wählen wir ein geeignetes Problem B, von dem wir bereits wissen, daß es unentscheidbar ist, und reduzieren B auf A (in Zeichen: B≤A). Wenn jetzt A entscheidbar wäre, so könnten wir auch B entscheiden, denn wir könnten die Eingabe zu B mithilfe der Reduktionsfunktion umbauen und den Algorithmus für A benutzen. Widerspruch!

25. **Wie ist die Klasse P definiert?**
    P = { A | es gibt eine (deterministische) TM M und ein Polynom p mit L(M)=A und $time_M(x) \leq p(|x|)\} = \{\bigcup_{pPolynom} TIME(p(n))\}$

26. **Wie ist die Klasse NP definiert? Wofür steht NP?**
    NP = $\{\bigcup_{pPolynom} NTIME(p(n))\}$
    NP steht für nichtdeterministisch polynomiell, d.h. es gibt eine nichtdeterministische TM für das betrachtete Problem, deren kürzeste akzeptierende Rechnung polynomiell in der Länge der Eingabe ist.

27. **Was ist NP-Vollständigkeit?**
    Ein Problem ist NP-vollständig genau dann, wenn es in NP liegt und NP-hart ist.

28. **Was bedeutet NP-hart?**
    intuitiv: mindestens so schwer wie alle Probleme in NP.
    formal: X ist NP-hart gdw. $\forall Y \in NP : Y \leq_p X$

29. **Welche Inklusionsbeziehung besteht trivialerweise zwischen den Klassen P und NP? Warum?**
    $P \subseteq NP$, da jede deterministische TM auch als eine (sehr spezielle) nichtdeterministische TM aufgefasst werden kann.

30. **Wo liegt der Unterschied zwischen Reduktion und polynomieller Reduktion?**
Bei der polynomiellen Reduktion ist vorgeschrieben, daß die „Umwandlung“ in polynomiell beschränkter Zeit erfolgen muß, bei der Reduktion fehlt diese Zeitbeschränkung, hier genügt es, daß die benutzte Funktion berechenbar ist.

31. **Warum ist der Satz von Cook zentral?**
Weil er uns mit SAT ein erstes NP-vollständiges Problem gibt, das wir für weitere NP-Vollständigkeitsbeweise benutzen können.

32. **Wie sind die Probleme CLIQUE, SAT, 3-SAT, TSP, RUCKSACK, SUBSET-SUM und HC definiert?**
CLIQUE: Gegeben G=(V,E) und eine natürliche Zahl k, gibt es eine Clique der Größe k in G, d.h. existiert $S \subseteq V$ mit $|S| = k$ und $\forall u, v \in S u, v \in E$?
SAT: Gegeben eine aussagenlogische Formel F, gibt es eine Belegung, die F erfüllt?
3-SAT: Gegeben eine aussagenlogische Formel F in KNF mit maximal 3 Literalen pro Klausel, gibt es eine Belegung, die F erfüllt?
TSP: Gegeben eine $n \times n$-Matrix von Entfernungen zwischen n Punkten und eine Zahl k, gibt es eine Permutation der n Punkte, so daß die Summe der Entfernungen zwischen benachbarten Punkten (inklusive der Entfernung vom letzten zum ersten Punkt) k nicht überschreitet?
RUCKSACK/SUBSET-SUM: Gegeben natürliche Zahlen b und a_i für i=1,...,k, gibt es eine Menge $I \subseteq \{1, ..., k\}$ mit $\sum_{i \in I} a_i = b$?
HC: Gegeben ein ungerichteter Graph G=(V,E), gibt es einen Hamiltonkreis in G, d.h. eine Permutation der Knoten, so daß aufeinanderfolgene Knoten (und der letzte und der erste Knoten) jeweils mit einer Kante aus E verbunden sind?

33. **Wie wird CLIQUE auf VC und SUBSET-SUM auf RUCKSACK reduziert?**
CLIQUE $\leq$ VC: gegeben G=(V,E) und k für CLIQUE, nutze den Komplementgraphen $\overline{G} = (V, \{\{u, v\} | u, v \in V, \{u, v\} \notin E\})$ und die Zahl $|V| - k$ für VC.
SUBSET-SUM $\leq$ RUCKSACK: identisches Problem!

34. **Was ist ein pseudopolynomieller Algorithmus?**
Ein Algorithmus, dessen Rechenzeit durch ein Polynom in L(I) (Länge der konkreten Eingabe I) und MAX(I) (Wert der größten in I vorkommenden Zahl) beschränkt ist.

35. **Was ist eine Orakel-Turingmaschine?**
Eine deterministische TM mit einem zusätzlichen Orakelband (für die Funtion $g : \Sigma^* \to \Sigma^*$ und zwei zusätzlichen Zuständen q_f (für Fragen an das Orakel) und q_a (Orakelantwortzustand). Damit kann zu einer Eingabe x auf dem Orakelband in einem Schritt g(x) berechnet werden.