Technische Universität München
Fakultät für Informatik
Lehrstuhl für Effiziente Algorithmen
Christian Ivicevic

Sommersemester 2015
Studentisches Tutorium
Zusatzmaterial
13. Februar 2015

Theoretische Informatik

– Sammlung an knappen Fragen zur Klausurvorbereitung –

Beantworten Sie die folgenden Fragen bzw. geben Sie an, ob die darin beschriebenen Aussagen wahr oder falsch sind. Begründen Sie Ihre Antworten möglichst knapp! Gegebenenfalls können Sie ein Gegenbeispiel angeben.

1. Wenn L regulär ist und $L' \subset L$ gilt, ist dann auch L' regulär?

   Lösung. Falsch, denn $L = L(a^*b^*)$ ist regulär, nicht aber $L' = L(a^n b^n)$.

2. Ist die Sprache $L = \{a^m b^n \mid (m+n) \mod 2 = 0\}$ regulär? Lässt sich dies mit dem Pumping-Lemma zeigen?

   Lösung. Die Sprache ist regulär, denn $L = L\left((aa)^*(\epsilon|aa|ab|bb)(bb)^*\right)$. Mit dem Pumping-Lemma lässt sich jedoch nicht zeigen, dass eine Sprache regulär ist, sondern nur, dass sie <u>nicht</u> regulär ist.

3. Gibt es endliche, nicht kontextfreie Sprachen?

   Lösung. Nein, denn jede endliche Sprache ist regulär und damit kontextfrei.

4. Ist die Sprache $L = \{a^m b^n \mid m < n\}$ kontextfrei?

   Lösung. Ja, sie wird von der durch $S \to aSb \mid Sb \mid b$ implizierten Grammatik erzeugt und es gilt somit $L = L(S)$.

5. Welche <u>endliche</u> Sprache beschreibt die Grammatik mit den beiden Produktionen $S \to aS \mid bB$ und $B \to bBb$?

   Lösung. Die leere Sprache, denn jede endliche Ableitung endet mit einem Wort, das ein Nichtterminal enthält.

6. Ist das Wortproblem für ein Wort der Länge n und eine CFG in der Zeit $\mathcal{O}(n^4)$ entscheidbar?

   Lösung. Laut Vorlesung entscheidet der CYK-Algorithmus das Wortproblem in Zeit $\mathcal{O}(n^3)$ und damit insbesondere auch in $\mathcal{O}(n^4)$.

7. Gibt es eine Turingmaschine, die sich nie mehr als vier Schritte vom Startzustand entfernt und eine unendliche Sprache akzeptiert? Begründung!

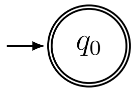
   Lösung. Ja, die Turingmaschine kann einfach unabhängig von der Eingabe direkt in den akzeptierenden Zustand wechseln.

8. Welche Sprachen lassen sich mit Turingmaschinen, die ihren Kopf immer nur nach rechts bewegen, erkennen?

   Lösung. Die regulären Sprachen. Da der Kopf sich nur nach rechts bewegt, liest eine solche Turingmaschine nur einmal die Eingabe, gegebenenfalls gefolgt von (unendlich) Blanks. Nachdem die Eingabe gelesen wurde, hängt es nur noch von dem aktuellen Zustand der Turingmaschine ab, ob irgendwann ein Endzustand erreicht werden wird, nicht mehr vom Band.

9. Es gibt einen NFA mit nur einem Zustand, der die Sprache $\{\epsilon\}$ akzeptiert.

   Lösung. Wahr. Beweis durch Beispiel:



10. Der Schnitt einer kontextfreien Sprache mit einer regulären Sprache ist regulär.

    Lösung. Falsch. Man wähle $L = \{a^n b^n \mid n \in \mathbb{N}\}$ als kontextfreie aber nicht reguläre Sprache und schneide mit Σ^*. $L \cap \Sigma^* = L$ ist nicht regulär und somit ein Gegenbeispiel zur Aussage.

11. Es gibt keine reguläre inhärent mehrdeutige Sprache.

    Lösung. Wahr. Sei L eine reguläre Sprache, dann existiert bekanntlich ein DFA A für L, aus dem eine rechts-lineare Grammatik herleitbar ist, die laut Vorlesung <u>niemals</u> mehrdeutig sein kann.

12. Jede unentscheidbare Menge enthält eine entscheidbare Teilmenge.

    Lösung. Wahr. Für jede unentscheidbare Sprache $L \subset \Sigma^*$ gilt $\emptyset \subset L$ und $\emptyset$ ist entscheidbar.

13. Jede Teilmenge einer entscheidbaren Sprache ist entscheidbar.

    Lösung. Falsch. Zwar ist Σ^* entscheidbar, das Halteproblem $K \subset \Sigma^*$ ist aber nicht entscheidbar.

14. Sind A und $A \cap B$ entscheidbar, dann muss B ebenfalls entscheidbar sein.

    Lösung. Falsch. Als Gegenbeispiel betrachten wir die entscheidbaren Megen $\emptyset$ und $\emptyset \cap H$, jedoch ist H nicht entscheidbar.

15. Jede totale Funktion $f : \mathbb{N} \to \mathbb{N}$ ist berechenbar.

    Lösung. Falsch. Man betrachte χ_K des Halteproblems K.

16. Das allgemeine Halteproblem ist eine Typ-0-Sprache.

    Lösung. Wahr. Das allgemeine Halteproblem H ist semi-entscheidbar. Es gibt also eine Turingmaschine, die H akzeptiert, deshalb ist H eine Typ-0-Sprache.

17. Ist $ntime_M$ für jede deterministische Turingmaschine M berechenbar? Begründung!

    Lösung. Nein. Eine deterministische Turingmaschine ist ein Spezialfall der nichtdeterministischen Turingmaschinen. Deshalb ist für M auch $ntime_M$ definiert. Im Unterschied zu $time_M$ ist allerdings $ntime_M$ eine totale Funktion auf Σ^*, der Menge der Gödelisierungen aller Turingmaschinen.

    Sei nun A eine nicht entscheidbare Menge, die von einer DTM M akzeptiert wird. Wäre $ntime_M$ berechenbar, dann gäbe es eine DTM N, die $ntime_M$ berechnet, d.h. bei jeder Eingabe stoppt, also auch im Fall $ntime_M(w) = 0$. Dies würde aber bedeuten, dass man entscheiden könnte, wann M nicht hält, und damit wäre A entscheidbar im Widerspruch zur Annahme!

18. Sei $f : \mathbb{N} \to \mathbb{N}$. Falls $NTIME(f(n))$ eine nichtentscheidbare Sprache enthält, dann ist f nicht berechenbar. Beweis!

    Lösung. Sei A eine nicht entscheidbare Sprache, die von einer NTM M akzeptiert wird, so dass $ntime_M(w) \leq f(|w|)$ für alle w gilt. Wir wissen, dass $ntime_M$ nicht berechenbar ist. Nun zeigen wir, dass es keine, auch noch so große, berechenbare Funktion f gibt, mit der man $ntime_M$ abschätzen kann.

    Nehmen wir an, es gäbe eine berechenbare Funktion f mit obiger Eigenschaft. Dann könnten wir für ein beliebiges Wort w den Wert $f(|w|)$ berechnen und die Turingmaschine M höchstens $f(|w|)$ Schritte rechnen lassen, um die Entscheidung über $w \in A$ zu erhalten, was ein Widerspruch zur Voraussetzung ist.

19. Es gibt eine Funktion $f : \mathbb{N} \to \mathbb{N}$, die primitiv-rekursiv ist, aber deren Definitionsbereich ($\{n \in \mathbb{N} \mid f(n) \neq \perp\}$) endlich ist.

    Lösung. Falsch. Jede primitiv-rekursive Funktion ist total und hat somit Definitionsbereich $\mathbb{N}$.

20. Es gibt eine Funktion $f : \Sigma^* \to \Sigma^*$, die total ist und für die $\{w \in \Sigma^* \mid \varphi_w = f\}$ entscheidbar ist.

    Lösung. Wahr. Man wähle eine beliebige nicht-berechenbare und totale Funktion (siehe Aufgabe 15). Dann ist die Menge leer und somit entscheidbar.

21. Es gibt eine Menge $A \subset \Sigma^*$, die nicht rekursiv-aufzählbar ist und deren Komplement ebenfalls nicht rekursiv aufzählbar ist.

    Lösung. Wahr. Die Menge $T = \{w \in \Sigma^* \mid \forall\ v : \varphi_w(v) \neq \perp\}$ aller terminierenden Turingmaschinen und $\overline{T}$ sind nach Rice-Shapiro nicht semi-entscheidbar.

22. Die Instanz

    $$\left\{ \begin{pmatrix} 01 \\ 0 \end{pmatrix}, \begin{pmatrix} 10 \\ 01 \end{pmatrix}, \begin{pmatrix} 0 \\ 01 \end{pmatrix} \right\}$$

    des PCP besitzt eine Lösung.

    Lösung. Falsch. Keine Karte kann die letzte sein.

23. Wenn f berechenbar ist, dann ist $A_f := \{w \in \Sigma^* \mid f(w) \neq \bot\}$ semi-entscheidbar.

    Lösung. Wahr. Die charakteristische Funktion

$$\chi_{A_f}(x) = \begin{cases} 1, & \text{falls } f(x) \neq \bot, \\ \bot, & \text{sonst.} \end{cases}$$

    ist berechenbar, da f berechenbar ist. Somit ist A semi-entscheidbar.

24. Für das spezielle Halteproblem K und eine beliebige Sprache A gilt: Wenn $K \cap A$ entscheidbar ist, dann ist A endlich.

    Lösung. Falsch. Mit $A = \overline{K}$ ist $K \cap \overline{K} = \emptyset$ entscheidbar, A jedoch offensichtlich nicht endlich.

25. Jedes Problem ist entweder in $\mathcal{P}$ oder $\mathcal{NP}$.

    Lösung. Falsch. 2COL ist in beiden. Alternativ: unentscheidbare Mengen liegen weder in $\mathcal{P}$ noch in $\mathcal{NP}$.

26. Für jede Turingmaschine ist die Funktion

$$\varphi(x) = \begin{cases} 1, & \text{falls } \mathcal{P} = \mathcal{NP}, \\ 0, & \text{falls } \mathcal{P} \neq \mathcal{NP}. \end{cases}$$

    berechenbar.

    Lösung. Wahr. φ ist konstant und somit unabhängig berechenbar von der Turingmaschine.

27. Wenn A $\mathcal{NP}$-vollständig ist, dann ist χ_A μ-rekursiv.

    Lösung. Wahr. $A \in \mathcal{NP} \implies A$ entscheidbar $\implies \chi_A$ berechenbar $\implies \chi_A\ \mu R$.

28. Wenn L_1 und L_2 rekursiv-aufzählbar sind, dann ist auch $L_1 \setminus L_2$ rekursiv-aufzählbar.

    Lösung. Falsch. $\Sigma^* \setminus K = \overline{K}$ wäre somit entscheidbar, was nicht zutrifft.

29. Die Menge $A = \{w \in \Sigma^* \mid \forall v : \varphi_w(v) = vv\}$ ist entscheidbar.

    Lösung. Falsch. Nach Rice-Shapiro gilt $A = \{w \mid \varphi_w \in C_F\}$ mit der Funktionenmenge $C_F = \{f \mid \forall v : f(v) = vv\}$. Da es offensichtlich Turingmaschinen gibt, die ihre Eingabe verdoppeln, aber auch welche, die es nicht tun, ist A unentscheidbar.[1]

30. Es gibt eine totale Funktion $f : \Sigma^* \to \Sigma^*$, für die gilt $|\{w \in \Sigma^* \mid \varphi_w = f\}| = 13$.

    Lösung. Falsch. Eine solche Funktion existiert nicht, da es unendlich viele Turingmaschinen gibt, die dieselbe berechenbare Funktion berechnen.

[1] Für die Klausur müsste man an dieser Stelle explizit hervorheben, dass wir in der Übung tatsächlich eine solche TM entwickelt haben, die Eingaben verdoppelt und daher existiert. Analog muss man als Gegenbeispiel eine Maschine angeben, die dies nicht tut und beweisen, dass sie auch das berechnen kann, was man beschrieben hat. In unserem Fall kann man trivialerweise eine TM hernehmen, die beispielweise 1+1 berechnet, da diese Funktion primitiv-rekursiv ist.

31. Seien L, K formale Sprachen. Aus $L^* = K^*$ folgt $L = K$.

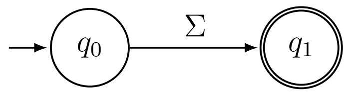
    Lösung. Falsch. $L = \{a\}$ und $K = \{a, aa\}$ sind ein Gegenbeispiel.

32. Jede Sprache ist entscheidbar oder semi-entscheidbar.

    Lösung. Falsch. $\overline{K}$ ist weder noch.

33. Jeder NFA mit genau zwei Zuständen, der mehr als drei unterschiedliche Wörter akzeptiert, akzeptiert eine unendliche Sprache.

    Lösung. Falsch. Für $|\Sigma| > 3$ ist der folgende NFA ein Gegenbeispiel:



34. Jede CFL liegt in $\mathcal{P}$.

    Lösung. Wahr. CYK entscheidet jede Sprache in $\mathcal{O}(n^3) \in \mathcal{P}$.

35. Eine Turingmaschine über Σ, die alle Wörter $w \in \Sigma^*$ mit $|w| > 13$ akzeptiert, beschreibt eine reguläre Sprache.

    Lösung. Wahr. Das Komplement der erkannten Sprache ist endlich und damit regulär. Da reguläre Sprachen abgeschlossen sind unter Komplement, ist also auch die erkannte Sprache regulär.

36. Ist f nicht LOOP-berechenbar, dann ist f auch nicht total.

    Lösung. Falsch. Ackermann ist nicht LOOP-berechenbar, aber total.

37. Jede totale Funktion wird von einem WHILE-Programm berechnet.

    Lösung. Falsch. Nicht jede totale Funktion ist berechenbar. Beispielsweise χ_K.

38. Eine Instanz des PCP hat entweder gar keine oder unendlich viele Lösungen.

    Lösung. Richtig. Wenn $i_1 \dots i_k$ ein Lösung ist, dann ist auch $(i_1 \dots i_k)^n$ für $n > 0$ eine Lösung.

39. Es gibt eine Turingmaschine M, so dass φ_M nicht berechenbar ist.

    Lösung. Falsch. Eine Funktion ist genau dann berechenbar, wenn es eine Turingmaschine gibt, die diese Funktion berechnet.

40. Jede Teilmenge einer kontextfreien Sprache ist wieder kontextfrei.

    Lösung. Falsch. $\{ww \mid w \in \Sigma^*\} \subset \Sigma^*$ ist nicht kontextfrei, Σ^* jedoch schon.

41. Seien A und B Sprachen mit $A = AB$. Dann gilt $A = B^*$.

    Lösung. Falsch. Sei $A = \emptyset$ und $B = \{\epsilon\}$. Dann ist $A = \emptyset = AB$, aber jedoch $A = \emptyset \neq \{\epsilon\} = B^*$.

42. $A \equiv a^*$ ist eine Lösung der Gleichung $A \equiv aA$.

    Lösung. Falsch. Durch Einsetzen von $A \equiv a^*$ in $A \equiv aA$ erhalten wir $a^* \equiv aa^*$. Dies gilt aber nicht, denn $\epsilon \in L(a^*)$ aber $\epsilon \notin L(aa^*)$.

43. Jedes erzeugende Symbol in einer Grammatik ist nützlich.

    Lösung. Falsch. Es könnte zum Beispiel nicht erreichbar sein.

44. Wenn $A \leq_p B$ und $B \leq_p A$, dann gilt $A = B$.

    Lösung. Falsch. Gegenbeispiel: $SAT \leq_p 3COL$ und $3COL \leq_p SAT$, aber es gilt definitiv $SAT \neq 3COL$.

45. Für alle $w \in \Sigma^*$ und $k \in \mathbb{N}$ gilt $|w^k| = |w|^k$.

    Lösung. Falsch. $|w^k| = k|w|$.