Tutorium Theoretische Informatik II 14. Juli 2010

Note Title 5/3/2010

- 11.4
- Platzkomplexität NP $\subseteq$ PSPACE
- 11.8 b
- 12.1 b

NP $\subseteq$ PSPACE

$NTIME(f) \subseteq SPACE(f) \subseteq TIME(2^f)$

↳ Wieviel Speicherplatz kann ich in Zeit $f(n)$ ausgeben

$NP \neq P$?

Aufgabe 11.4: Partition $= \{ (a_1, .. a_n) \mid \exists I \subseteq \{1..n\} . \sum_{i \in I} a_i = \sum_{i \notin I} a_i \}$

a) Partition $\in NP$

1) Rate nichtdeterministisch eine Menge $I \subseteq \{1..n\}$

2) Berechne $\sum_{i \in I} a_i$ und $\sum_{i \notin I} a_i$ und teste auf Gleichheit

3) Rechenzeit: Insgesamt n Additionen und ein Test bei m-stelligen Zahlen $\mathcal{O}(n \cdot \cancel{m})$ Operationen da Größe der Eingabe ebenfalls im Bereich $n \cdot \cancel{m}$ liegt ist Rechenzeit linear, also polynomiell

Damit kann ein Lösungsvorschlag in polynomieller Zeit verifiziert werden
also Partition in NP

b) Partition ist NP-hart

Hier sind einige Vorüberlegungen nötig, auch wenn auf den Folien in §6.3 der Hinweis $KP^* \leq_p$ PARTITION steht bei Partition muß eine Menge von Zahlen so auf zwei Stapel verteilt werden, daß jeder Stapel genau den Wert $\sum a_i / 2$ erreicht

Das einzige bisher bekannte NPC Problem, das hierzu ähnlich ist, ist das Rucksackproblem in seiner strikten Form KP*

Hier geht es darum eine Menge von Zahlen so aufzuteilen, daß die gewählte Zahlen genau eine Summe A ergeben.

Der Unterschied ist, daß bei Partition die nicht gewählte Zahlen ebenfalls diese Summe ergeben müssen, daß "A" also genau die Hälfte der Summe aller Zahlen sein muß.

Bei KP* wäre $\sum_{i \in I} a_i = A$ und $\sum_{i \notin I} a_i = \sum a_i - A$

wenn man für die Reduktion $KP^* \leq_p$ Partition noch zwei weitere Zahlen $\underline{a_{n+1} = A}$ und $\underline{a_{n+2} = \sum a_i - A}$ hinzufügt dann wäre für ein lösbares KP* Problem

$$\sum_{i \in I} a_i + a_{n+2} = A + \sum a_i - A = \sum a_i = \sum_{i \notin I} a_i + a_{n+1}$$

Zwei pathologische Situationen sind zu beachten

- wenn $\sum a_i = A$ ist, dann wäre $a_{n+2} = 0$
  das ist unschön, weil in einer realen Partitionierung
  die Summanden positiv sein sollten
  (viele Formalisierungen verlangen daher $a_i > 0$)
  dies umgeht man, indem man $a_{n+1} = A+1$ $\quad a_{n+2} = \sum a_i - A + 1$
  wählt

- wenn $\sum a_i < A$ ist, dann wäre $a_{n+2} < 0$, was nicht
  erlaubt ist. Allerdings ist in diesem Fall das Rucksackproblem
  für die a_i nicht lösbar und kann somit in ein nicht lösbares
  Partitionierungsproblem abgebildet werden. Hier bietet sich
  z.B. ein Problem mit <u>einem</u> Summanden $a_1 = 1$ an

Nach all diesen Vorüberlegungen nun die offizielle Lösung

4) Wir wählen das bereits als NP-vollständig bekannte
Problem UP* und zeigen UP* $\leq_p$ Partition

5) Sei $(a_1, \dots, a_n, A)$ Eingabe für ein UP* Problem

dann sei

$$f(a_1, \dots a_n, A) = \begin{cases} (a_1, \; a_n, \; A+1, \; \Sigma a_i - A + 1) & \text{falls } \Sigma a_i \geq A \\ (1) & \text{sonst} \end{cases}$$

dabei steht Σa_i kurz für $\sum_{i=1}^{n} a_i$

6) Wir zeigen $(a_1, \dots a_n, A) \in UT^* \iff f(a_1 \dots a_n, A) \in$ Partition

$\Rightarrow$ Sei $(a_1, \dots a_n, A) \in UT^*$

Dann gibt es ein $I \subseteq \{1 \dots n\}$ mit $\sum_{i \in I} a_i = A$

dann ist $f(a_1 \dots a_n, A) = (a_1, \dots a_n, \underbrace{A+1}_{a_{n+1}}, \underbrace{\Sigma a_i - A + 1}_{a_{n+2}})$

und es gilt

$$\sum_{i \in I} a_i + (\underbrace{\Sigma a_i - A + 1}_{a_{n+2}}) = \Sigma a_i + 1 = (\Sigma a_i - A) + (A+1) = \sum_{i \notin I} a_i + (\underbrace{A+1}_{a_{n+1}})$$

also gilt für $J = I \cup \{n+2\}$: $J \subseteq \{1 \dots n+2\}$ und

$$\sum_{i \in J} a_i = \sum_{i \notin J} a_i$$ d.h. $f(a_1 \dots a_n, A) \in$ Partition

$\Leftarrow$ Sei $f(a_1,..a_n,A) \in$ Partition

Dann ist $\sum a_i \geq A$, da (1) nicht Partitionierbar ist

und es gibt ein $J \subseteq \{1..n+2\}$ mit $\sum_{i \in J} a_i = \sum_{i \notin J} a_i = \sum a_i + 1$

Wegen $a_{n+1} + a_{n+2} = \sum a_i + 2$ muß

a_{n+1} in J und a_{n+2} in $\bar{J}$ liegen oder umgekehrt

Wir setzen
$$I = \begin{cases} J \setminus \{a_{n+2}\} & \text{wenn } a_{n+2} \in J \\ \bar{J} \setminus \{a_{n+2}\} & \text{sonst} \end{cases}$$

dann ist $\sum_{i \in I} a_i = \sum a_i + 1 - (\sum a_i - A + 1) = A$

also $(a_1..a_n,A) \in KP^*$

7) Die Funktion f kann in linearer Zeit durch einfaches Aufsummieren berechnet werden, hat also polynomielle Laufzeit

Insgesamt folgt also $KP^* \leq_p$ Partition und damit ist Partition NP hart

Die Lösungen für Aufgabe 11.3 bzw 12.2 verfahren nach dem gleichen Prinzip. Die Schwierigkeit liegt dabei in in der Reduktionsfunktion. Die Überprüfung, ob das Problem in NP liegt oder daß die Reduktionsfunktion in polynomieller Zeit arbeitet und korrekt ist, ist verhältnismäßig schematisch.

11.3 Clique $\leq_p$ Independent mit $f(G,k) = (G^c, k)$

12.2 Partition $\leq_p$ Binpack mit $f(a_1..a_n) = (a_1, ..a_n, \frac{\sum a_i}{2}, 2)$

11.5b: Wenn wir lösen

" $L = \{w \mid w \in \{0,1\}^* \wedge \#_1(w)$ gerade$\} \in NPC$?

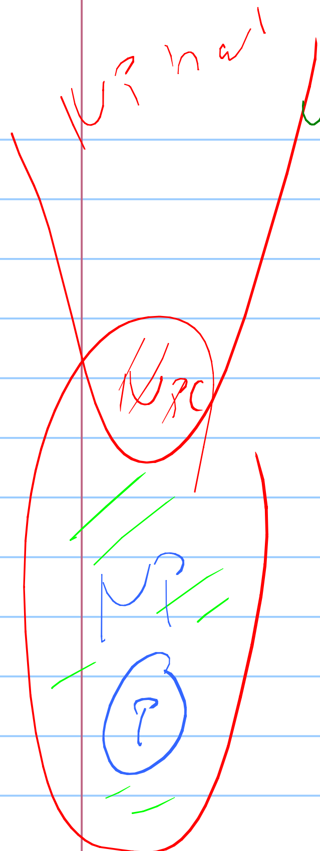
"

dann wissen wir ob $P = NP$ oder nicht

Wir wissen $L \in P$

wenn $L \in NPC \Rightarrow \forall L' \in NP: L' \leq_p L \in P$

$NP \subseteq P$ also $P = NP$



ungeklärt wenn $P = NP$ dann $L \in NPC$, weil

für jede Sprache $L' \in NP$ gilt $L' \in P$ also reduzieren auf L

mit $f(x) = \begin{cases} \text{'0'} \in L & \text{wenn } x \in L' \\ \text{'1'} \notin L & \text{sonst} \end{cases}$

d.h. $\forall L' \in NP.\ L' \leq_p L$ d.h. L ist NP hart

und $L \in P \subseteq NP \rightsquigarrow L \in NPC$

Beweis für 11.6a

$L \in NPC \Rightarrow P = NP$

$L \in NPC \Leftarrow P = NP \quad \hat{=} \quad L \notin NPC \Rightarrow P \neq NP$

folgt aus 11.5a

12.1.5

a) $C = co\text{-}C$ wenn C deterministische Zeit/Platzklasse

$$\Updownarrow$$

$$\{\overline{L} \mid L \in C\}$$

Begründung: nehme akzeptierende Maschine (det.!) und vertausche Endzustände mit Nicht-Endzustand

$$\chi_{\overline{L}}(x) := 1 - \chi_L(x)$$

poly poly

b) geht das auch wenn C nichtdeterministische Klasse

- NTM M akzeptiert, wenn es einen akzeptierenden Pfad gibt und Rechenzeit poly
- OTM akzeptiert, wenn die richtige Lösung in poly Zeit zu prüfen ist

die veränderte NTM M' wurde akzeptieren wenn es einen Pfad gibt, der $w \in L$ nicht akzeptiert

brauche: $w \in \bar{L}$, wenn M' sagt es gibt keinen Weg zu akzeptieren

ich muß alle Pfade durchgehen um das zu zeigen

All-Pfad NTM akzeptiert wenn jeder Abarbeitungspfad zur akzeptierenden Endzustand führt

↳ kann man nicht einmal im Ansatz bauen