Übungen zur Algorithmischen Bioinformatik I

Blatt 7

Xiheng He

Mai 2021

4. Aufgabe: NP-vollständigkeit von Erfüllbarkeits-Probleme (10 Punkte)

In der Vorlesung wurde bewiesen, dass CNF-SAT (also boolsche Formeln in CNF ('conjunctive normal form') und 3-SAT (CNF mit genau 3 Literalen pro Klausel) NP-vollständig sind.

(a) Beweisen oder widerlegen Sie (natürlich mit genauer Begründung = Beweis!)

(a) BOOL-SAT (beliebige boolsche Formeln) ist NP-vollständig.

Aus Therorem 5.18 (Satz von Cook und Levin) auf Skript Seite 255: BOOL-SAT ist $\mathcal{NP}-$ *vollständig.*

Beweis: Folien Seite 123-135

(b) 6-SAT (boolsche Formeln in CNV mit genau 6 Literalen pro Klausel) ist NP-vollständig.

Zu zeigen ist, dass 6-SAT ist NP-vollständig.

Zunächst zeigen wir, dass 6-SAT $\in \mathcal{NP}$.

- Angenommen, dass 6-SAT n boolsche Variablen besitzen, dann für jede Variable genau zwei boolsche Werte gibt damit beträgt die Laufzeit mindestens in $\Omega(2^n)$ um die 6-SAT Formel zu erfüllen. Deshalb gibt es keinen Algorithmus in polynomieller Laufzeit um das Problem zu lösen.
- Einer Algorithmus zur Validierung der beliebigen Erfüllbarkeit für 6-SAT kann aber in polynomieller Laufzeit durchführen da er nur alle Variable die entsprechende Werte zuweisen und die Erfüllbarkeit überprüfen muss. D.h. Es gibt aber eine polynomiell zeitbeschränkte DTM M und ein Polynom q, so dass M in nach höchstens $p(|x|)$ Schritten die Eingabe verfizieren (akzeptieren oder ablehnen) kann.

Daraus folgt: 6-SAT $\in \mathcal{NP}$

Anschließend zeigen wir: $\forall x \in \textbf{3-SAT} \Leftrightarrow f(x) \in \textbf{6-SAT}, T(f(x)) \in O(n^k)$

sei ein 3-SAT $:= (U, C)$ und ein 6-SAT $:= (U', C')$,

$C := \{c_1, c_2, \dots c_n\}, C_i = \{x_i, y_i, z_i\}$

$C' := \cup_{i=1}^{n} C'_i, U'_i = \{a_i, b_i, c_i\} \cup C_i$

$C'_i = \{\{x_i, y_i, z_i, a_i, b_i, c_i\}, \{x_i, y_i, z_i, \overline{a_i}, b_i, c_i\}, \{x_i, y_i, z_i, a_i, \overline{b_i}, c_i\}, \{x_i, y_i, z_i, a_i, b_i, \overline{c_i}\},$

$\{, \overline{a_i}, \overline{b_i}, c_i\}, \{x_i, y_i, z_i, a_i, \overline{b_i}, \overline{c_i}\}, \{x_i, y_i, z_i, \overline{a_i}, b_i, \overline{c_i}\}, \{x_i, y_i, z_i, \overline{a_i}, \overline{b_i}, \overline{c_i}\}\}$

Zu zeigen ist, dass die Reduktion f polynomielle Laufzeit besitzt.

$\forall C'_i \in C'$, eine Umformung von 3-SAT in 6-SAT kann in konstanter Zeit abgeschlossen werden, damit liegt die Laufzeit $T(f(x))$ offensichtlich in polynomieller Zeit ($T(f(x)) \in O(n)$ wobei n die Eingabegröße der Variablen in 3-SAT).

Zu zeigen ist, dass $\forall x \in$ **3-SAT** $\Leftrightarrow f(x) \in$ **6-SAT**

- falls 3-SAT erfüllbar ist, dann jede Klausel C_i als *true* ausgewertet werden kann, d.h $\forall x_i, y_i, z_i \in C_i, x_i \vee y_i \vee z_i$ ist wahr, daher ist jede entsprechende C'_i auch wahr, da jede Klausel in 6-SAT durch drei zusätzliche Literale a_i, b_i, c_i und ihre Komplemente umgeformt werden kann und somit macht keinen Unterschied. Deswegen kann 6-SAT erfüllt werden wenn 3-SAT erfüllbar ist.
- falls 6-SAT erfüllbar ist, dann ist jede Klausel C'_i wahr damit kann man herleiten, dass C_i auch wahr ist. D.h. 3-SAT kann erfüllt werden wenn 6-SAT erfüllt wurde.

Es gilt: **6-SAT** $\in \mathcal{NP} \wedge$ **3-SAT** $\leq_p$ **6-SAT** $\Longrightarrow$ 6-SAT ist NP-vollständig.

(c) 2-SAT (boolsche Formeln in CNV mit genau 2 Literalen pro Klausel) ist NP-vollständig.

2-SAT ist nicht NP-vollständig (**Lemma 5.24** Skript.Seite 256: 2-SAT $\in \mathcal{P}$)

Zu zeigen ist, dass 2-SAT $\in \mathcal{P}$.

Zunächst zeigen wir, dass es einen Algorithmus in polynomieller Zeit für 2-SAT gibt.

(1) Für eine beliebige Variable x_i in i-ter Klausel, setzen z.B $x_i := true$.

(2) Dann entfernen alle Klausel, die x_i enthält.

(3) überprüfen welche Klausel $\overline{x_i}$ enthält, setzen die andere Variable $y_i := true$ in der Klausel z.B $\{\overline{x_i}, y_i\}$ damit kann 2-SAT erfüllt werden. Falls Widerspruch entsteht, setzen $x_i := false$. Nach der überprüfung alle Klausel können diese solche Klausel gestrichen werden.

(4) Wiederholen (1), bis alle Variablen boolschen Wert zugeteilt worden haben, so dass 2-SAT erfüllt wurde.

Es ist leicht zu zeigen, dass für 2-SAT mit n Variablen genau $n/2$ Klausel hat und für jede unbestimmte Variable müssen maximal $n/2$ Klausel überprüft werden. Damit liegt die Laufzeit des obigen Algorithmus in $O(n^2)$ und 2-SAT kann in polynomieller Zeit gelöst werden.

Daraus folgt: 2-SAT ist nicht NP-vollständig.

(b) Was ist der Unterschied zwischen der *Äquivalenz* zweier Formeln und ihrer *Erfüllbarkeitsäquivalenz*?

Erfüllbarkeitsäquivalenz bedeutet, dass die zweier Formeln entweder erfüllt werden können oder nicht erfüllbar sind wobei die zweier Formeln nicht übereinstimmen dürfen.

Äquivalenz unterscheidet sich von *Erfüllbarkeitsäquivalenz*, da zwei äquivalente Formeln immer gleiche Modelle haben.