Aufgaben zum Tut am 13.02.2006

Thomas Pajor

14. Februar 2006

Aufgabe 1) [Wiederholung]

Sei $L := \{a^j b^k c^l \mid j, k, l \geq 0 \text{ und } k = l \text{ falls } j = 1\}$ mit $\Sigma := \{a, b, c\}$.

(a) Zeigen Sie, dass L alle Bedingungen des Pumping Lemma erfüllt.

(b) Zeigen Sie, dass L trotzdem nicht regulär ist.

Lösung.

(a) Sei n die Pumping Lemma Zahl. Betrachte alle Wörter $w \in L$ mit $|w| > n$. Es gibt nun folgende drei Fälle zu unterscheiden:

- w enthält genau ein a. Das Wort hat also die Form ab^kc^k. Wähle als Zerlegung $w = uvx$ mit $u = \varepsilon$ und $v = a$. Es ist also $|uv| \leq n$ und $|v| > 0$ erfüllt. Damit ist aber auch für jedes $i \in \mathbb{N}_0$ das Wort $uv^ix = a^ib^kc^k \in L$.
- w enthält genau zwei as. Wähle als Zerlegung $w = uvx$ mit $u = \varepsilon$ und $v = aa$. Damit ist auch $uv^ix \in L$ für alle $i \in \mathbb{N}_0$.
- w enthält kein a oder mehr als 2 as. Wähle als Zerlegung $w = uvx$ wieder $u = \varepsilon$ und $v = w_1$ wobei w_1 der erste Buchstabe von w ist. Ist $w_1 = a$, so gilt $|w|_a > 2$ und es gilt für alle $i \in \mathbb{N}_0$ dass $|uv^ix|_a > 1$ und somit ist $uv^ix \in L$ für alle $i \in \mathbb{N}_0$. Ist $w_1 \neq a$, so ist $w_1 = b$ oder $w_1 = c$ und damit auf triviale Weise $uv^ix \in L$ für alle $i \in \mathbb{N}_0$.

L erfüllt also die Bedingungen des Pumping Lemmas.

(b) Wir zeigen $\operatorname{index}(R_L) = \infty$, wobei R_L die Nerode Relation von L bezüglich Σ^* ist. Dann folgt mit dem Satz von Nerode, dass L nicht regulär ist.

Betrachte für beliebiges $k \geq 0$ folgende Äquivalenzklassen $A_k := [ab^{n+k}c^n]_\sim$, also

$$[ab^nc^n]_\sim := \{a, abc, abbcc, abbbccc, \dots\}$$
$$[ab^{n+1}c^n]_\sim := \{ab, abbc, abbbcc, \dots\}$$
$$[ab^{n+2}c^n]_\sim := \{abb, abbbc, abbbbcc, abbbbbccc, \dots\}$$
$$\vdots$$

Jede dieser Klassen enthält nur äquivalente Wörter, denn es gilt für zwei Wörter $w_1, w_2 \in A_k$

$$w_1 z \in L \quad \Leftrightarrow \quad z = c^k \quad \Leftrightarrow \quad w_2 z \in L$$

Außerdem fallen keine zwei paarweise verschiedenen Klassen A_k und A_l zusammen, da für zwei beliebige Wörter $w_1 \in A_k$ und $w_2 \in A_l$ gilt

$$w_1 z \in L \quad \Leftrightarrow \quad z = c^k \quad \overset{l \neq k}{\Leftrightarrow} \quad z \neq c^l \quad \Leftrightarrow \quad w_2 z \notin L$$

$\Rightarrow$ Σ^* zerfällt über R_L in unendlich viele Äquivalenzklassen und damit folgt mit dem Satz von Nerode dass L nicht regulär ist.

Aufgabe 2) [Wiederholung]

Betrachten Sie das Problem SUBGRAPHENISOMORPHIE:

Gegeben: Zwei Graphen $G_1 := (V_1, E_1)$ und $G_2 := (V_2, E_2)$ mit $|V_2| < |V_1|$.

Frage: Gibt es eine Menge $U \subset V_1$ mit $|U| = |V_2|$ und einen Isomorphismus $\Phi : U \to V_2$, also eine bijektive Abbildung so, dass gilt

$$\{x, y\} \in E_1 \quad \Leftrightarrow \quad \{\Phi(x), \Phi(y)\} \in E_2 \qquad\qquad \forall x, y \in U$$

Also ist G_2 isomorph zu einem Teilgraph von G_1? Ein Beispiel ist in Abbildung (1) gegeben.

Zeigen Sie: SUBGRAPHENISOMORPHIE ist $\mathcal{NP}$-vollständig.

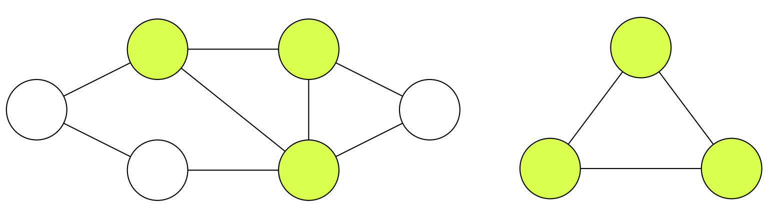

Abbildung 1: Beispiel für eine erfüllbare Instanz von SUBGRAPHENISOMORPHIE. Der Graph G_2 (rechts) ist isomorph zu der grünen Menge an Knoten in G_1.

Lösung.

Um zu zeigen dass SUBGRAPHENISOMORPHIE $\mathcal{NP}$-vollständig ist, zeigen wir zunächst, dass eine geratene Lösung in polynomieller Zeit daraufhin überprüft werden kann, ob sie eine gültige Lösung ist. Und im zweiten Schritt reduzieren wir ein bekanntes $\mathcal{NP}$-vollständiges Problem auf SUBGRAPHENISOMORPHIE. Wir werden dafür CLIQUE benutzen.

1. SUBGRAPHENISOMORPHIE $\in \mathcal{NP}$

Sei also $(U \subset V_1, \Phi)$ eine geratene Lösung. Folgender Algorithmus in Pseudocode überprüft ob U eine gültige Lösung ist: Da wir nur zwei Schleifen über U durchlaufen und bei jedem Durchlauf überprüfen

Algorithm 1 VERIFIZIERE(U, Φ)

for all $x \in U$ **do**
    for all $y \in U$ **do**
        if $\{x, y\} \in E_1$ and $\{\Phi(x), \Phi(y)\} \notin E_2$ **then**
            return „nein!“
        if $\{x, y\} \notin E_1$ and $\{\Phi(x), \Phi(y)\} \in E_1$ **then**
            return „nein!“
return „ja!“

müssen ob eine Kante in E_1 bzw. E_2 enthalten ist, ist der Aufwand in $\mathcal{O}(|V_1|^2|E_1||E_2|)$, und somit polynomiell zur Eingabelänge.

Damit kann eine geratene Lösung von SUBGRAPHENISOMORPHIE in polynomieller Zeit verifiziert werden und es folgt SUBGRAPHENISOMORPHIE $\in \mathcal{NP}$.

2. SUBGRAPHENISOMORPHIE ist $\mathcal{NP}$-schwer

Wir zeigen CLIQUE $\leq_p$ SUBGRAPHENISOMORPHIE. Wir müssen also eine Abbildung f angeben, die jede Instanz $I = (V, E, K)$ von CLIQUE in eine Instanz $I' = (G_1 := (V_1, E_1), G_2 := (V_2, V_2))$ von SUBGRAPHENISOMORPHIE transformiert. Für f muss gezeigt werden, dass f in polynomieller Zeit berechnet werden kann, und außerdem muss gezeigt werden dass gilt

$$I \text{ enthält eine Lösung} \quad \Longleftrightarrow \quad f(I) = I' \text{ enthält eine Lösung}$$

Sei also $I := (G := (V, E), K)$ eine beliebige Instanz von CLIQUE. Konstruiere dazu eine Instanz $I' := (G_1 := (V_1, E_1), G_2 := (V_2, V_2))$ von Subgraphenisomorphie wie folgt:

- $V_1 := V$
- $E_1 := E$
- $V_2 := \{1, \ldots, K\}$ (G_2 soll also genau K Knoten haben)

- $E_2 := \{\{a,b\} \mid a,b \in V_2, a \neq b\}$ (G_2 soll vollständig verbunden sein)

Die Konstruktion ist mit polynomiellem Zeitaufwand möglich, da mit $|V|_2 =: n$ gilt $|E_2| = \binom{n}{2} = \frac{n^2-n}{2} \in \mathcal{O}(n^2)$[1].

Nun bleibt noch zu zeigen, dass I genau dann eine Lösung enthält wenn auch $f(I) = I'$ eine Lösung enthält.

$\Rightarrow$: Sei I eine Instanz von CLIQUE, die eine Lösung enthält. Das heißt also in $G = (V, E)$ gibt es eine Clique der Größe K. Sei $U := \{u_1, \ldots, u_K\}$ die Menge der Knoten, die die Clique bilden. Dann ist $|U| = K$ und der durch U induzierte Teilgraph von G ein vollständiger Graph mit K Knoten. Da nach Konstruktion von I' der Graph G_2 ebenfalls ein vollständiger Graph mit K Knoten ist, ist G_2 isomorph zu dem durch U induzierten Teilgraph in G.

$\Rightarrow$ Die Instanz I' enthält ebenfalls eine Lösung.

$\Leftarrow$: Sei I' eine aus I (nach obigem Verfahren) hervorgegangene Instanz von SUBGRAPHENISOMORPHIE. Außerdem habe I' eine Lösung. Das heißt der Graph G_2 kommt als Teilgraph in G_1 vor. Sei U die Lösungsmenge der ausgewählten Knoten von G_1. Dann hatte aber auch I eine Lösung, da nach Konstruktion $G = G_1$ gilt, und U damit auch in G enthalten ist. Wegen $|U| = K$, enthält G eine Clique der Größe K.

$\Rightarrow$ SUBGRAPHENISOMORPHIE ist $\mathcal{NP}$-schwer.

Mit SUBGRAPHENISOMORPHIE $\in \mathcal{NP}$ folgt die $\mathcal{NP}$-Vollständigkeit. $\square$

Aufgabe 3) [Wiederholung]

Gegeben sei das Problem NEAR–TAUT das wie folgt definiert ist:

Gegeben: Variablenmenge U und eine Klauselmenge C über U.
Frage: Gibt es höchstens eine Variablenbelegung in U so, dass nicht alle Klauseln erfüllt sind?

Zeigen Sie: NEAR–TAUT $\in$ co–$\mathcal{NP}$.

Lösung.

Die Definition von co–$\mathcal{NP}$ lautet co–$\mathcal{NP} := \{L \mid L^c \in \mathcal{NP}\}$. Wir müssen also zeigen dass das zu NEAR–TAUT komplementäre Problem co–NEAR–TAUT in $\mathcal{NP}$ liegt. Das Problem lässt sich wie folgt formulieren:

Frage: Gibt es mindestens zwei Variablenbelegungen in U so, dass nicht alle Klauseln erfüllt sind?

[1] In einem vollständigen einfachen Graph $G = (V, E)$ gilt immer $|E| = \binom{|V|}{2}$, denn der Binomialkoeffizient $\binom{n}{k}$ beschreibt ja gerade die Anzahl k–elementiger Teilmengen aus einer n–elementigen Menge. In unserem Fall für $k = 2$ sind das gerade die Kanten zwischen den Knoten, da wir eine Kante durch eine zweielementige Teilmenge von V beschreiben.

Um einen geratenen Lösungsvorschlag zu überprüfen ob er eine gültige Lösung ist, braucht man nur zwei der geratenen Variablenbelegungen daraufhin überprüfen ob sie mindestens eine Klausel zu falsch auswerten. Dies ist mit Aufwand $\mathcal{O}(|C|)$ möglich.

Damit ist co–NEAR–TAUT $\in \mathcal{NP}$, also co–co–NEAR–TAUT $=$ NEAR–TAUT $\in$ co–$\mathcal{NP}$. $\square$