Humboldt-Universität zu Berlin
Institut für Informatik
Lehrstuhl Logik in der Informatik
Prof. Dr. Nicole Schweikardt

21. April 2015

Logik und Komplexität

Sommersemester 2015

Übungsblatt 1

Zu bearbeiten bis 29. April 2015

Aufgabe 1: **(25 Punkte)**

Für eine FO[σ]-Formel $\varphi(\bar{x})$ mit k freien Variablen (für $k \geqslant 1$) sei $\varphi(\mathfrak{A}) := \{\bar{a} \in A^k : \mathfrak{A} \models \varphi(\bar{a})\}$, für jede σ-Struktur $\mathfrak{A}$. Zeigen Sie, dass es eine Signatur σ gibt, so dass das folgende Problem unentscheidbar ist.

> Query Containment Problem für FO[σ]
>
> *Eingabe:* Eine Zahl $k \geqslant 1$ und FO[σ]-Formeln φ und ψ mit je k freien Variablen.
>
> *Frage:* Gilt für alle endlichen σ-Strukturen $\mathfrak{A}$: $\varphi(\mathfrak{A}) \subseteq \psi(\mathfrak{A})$?

Aufgabe 2: **(25 Punkte)**

Eine Signatur σ nennen wir **binär**, falls jedes Symbol in σ ein Relationssymbol der Stelligkeit 2 ist.

Beweisen Sie folgende Verschärfung des Satzes von Trakhtenbrot:

Es gibt eine endliche, binäre Signatur $\hat{\sigma}$, so dass das endliche Erfüllbarkeitsproblem für FO[$\hat{\sigma}$] unentscheidbar ist.

Hinweis: Überlegen Sie sich eine geeignete Repräsentation beliebiger σ-Strukturen durch kantengefärbte Graphen, repräsentiert durch Strukturen über einer geeigneten binären Signatur $\hat{\sigma}$. Benutzen Sie die in der Vorlesung für $\sigma := \{<, \mathsf{succ}, 0, B, K, Z\}$ bewiesene Unentscheidbarkeit des endlichen Erfüllbarkeitsproblems für FO[σ]-Sätze, um die Unentscheidbarkeit des endlichen Erfüllbarkeitsproblems für FO[$\hat{\sigma}$]-Sätze zu beweisen.

Aufgabe 3: **($8 + 8 + 9 = 25$ Punkte)**

(a) Was drückt der folgende Satz in einem ungerichteten Graphen aus?

$$\forall X \Big(\big(\exists x X(x) \ \wedge \ \exists x \neg X(x) \big) \rightarrow \ \exists x \exists y \big(X(x) \wedge E(x,y) \wedge \neg X(y) \big) \Big)$$

(b) Sei $\sigma_{\text{Graph}} = \{E\}$, wobei E ein 2-stelliges Relationssymbol ist. Geben Sie einen MSO[σ_{Graph}]-Satz an, der in einem ungerichteten Graphen G ausdrückt, dass G ein Baum ist.

(c) Geben Sie einen ESO[σ_{Graph}]-Satz an, der in einem ungerichteten Graphen G ausdrückt, dass G eine gerade Anzahl an Zusammenhangskomponenten enthält.

— auf der nächsten Seite geht's weiter —

Aufgabe 4: **(13 + 12 = 25 Punkte)**

Die für diese Aufgabe nötigen Definitionen zu Baumautomaten und regulären Baumsprachen finden Sie unten.

(a) Sei $\Sigma = \{a, b\}$ und sei $L \subseteq T_\Sigma$ die Baumsprache, die aus allen Σ-Bäumen besteht, in denen jedes Blatt gerade Höhe hat. Hierbei sei die *Höhe* eines Blattes b definiert als die Anzahl der Kanten auf einem gerichteten Pfad von der Wurzel zu b. Zeigen Sie, dass L regulär ist.

(b) Sei Σ ein endliches Alphabet. Zeigen Sie, dass Folgendes gilt: Wenn $L_1 \subseteq T_\Sigma$ und $L_2 \subseteq T_\Sigma$ regulär sind, so ist auch $L_\cup := L_1 \cup L_2$ regulär.

Definitionen

Binärbäume. Ein *gewurzelter Baum* G ist ein endlicher gerichteter Graph, der als ungerichteter Graph ein Baum ist (d.h. der ungerichtete Graph, der aus dem gerichteten Graphen G entsteht, indem jede Kante durch eine ungerichtete Kante ersetzt wird, ist ein Baum) und der einen *Wurzelknoten* enthält, von dem aus jeder andere Knoten über einen gerichteten Pfad erreichbar ist. Wenn jeder Knoten von G entweder ein Blatt ist oder genau zwei Kinder besitzt, so heißt G *voll*. Ein voller Baum, dessen Kantenmenge E in $E_1 \dot\cup E_2$ partitioniert ist, so dass jeder Knoten, der kein Blatt ist, genau ein Kind in E_1 (sein *erstes Kind*) und ein Kind in E_2 (sein *zweites Kind*) besitzt, heißt *geordneter Binärbaum*.

Σ-Bäume. Sei Σ ein endliches (nicht-leeres) Alphabet. Ein Σ-*Baum* $t = (B, \lambda)$ besteht aus einem geordneten Binärbaum B und einer Abbildung λ, die jedem Knoten v von B eine *Beschriftung* $\lambda(v) \in \Sigma$ zuordnet. Die Menge aller Σ-Bäume bezeichnen wir mit T_Σ. Eine *Baumsprache* ist eine Teilmenge L von T_Σ.

Baumautomaten und reguläre Baumsprachen. Analog zu den endlichen Automaten aus der Theorie der formalen Sprachen, definieren wir Automaten, die Σ-Bäume verarbeiten. Ein (bottom-up) *Baumautomat* ist ein Tupel $\mathbb{A} := (Q, \Sigma, \Delta, F)$, wobei Q eine endliche Menge, Σ ein endliches Alphabet, $F \subseteq Q$ und $\Delta \subseteq (Q^2 \cup \{\bot\}) \times \Sigma \times Q$ eine Relation ist. Q heißt *Zustandsmenge*, F heißt *Menge der akzeptierenden Zustände* und Δ heißt *Überführungsrelation* von $\mathbb{A}$. Falls Δ der Graph einer Abbildung ist, die auf ganz $(Q^2 \cup \{\bot\}) \times \Sigma$ definiert ist, so heißt $\mathbb{A}$ *deterministisch*, ansonsten heißt $\mathbb{A}$ *nichtdeterministisch*. Intuitiv können wir uns vorstellen, dass $\mathbb{A}$ einen Σ-Baum t von den Blättern ausgehend verarbeitet und sich dabei auf die Wurzel zubewegt. Sei $V(t)$ die Knotenmenge des Binärbaums von t. Dabei baut $\mathbb{A}$ eine Funktion $q : V(t) \to Q$ auf, die als *Lauf von* $\mathbb{A}$ *auf* t bezeichnet wird. Wenn $\mathbb{A}$ deterministisch ist, ist q eindeutig bestimmt; ansonsten kann $\mathbb{A}$ viele Läufe auf t haben. Zunächst wird jedem Blatt v mit Beschriftung $a \in \Sigma$ ein Zustand $q(v) \in Q$ mit $(\bot, a, q(v)) \in \Delta$ zugewiesen. Nun bestimmt $\mathbb{A}$ rekursiv den Zustand $q(v)$ eines mit $a \in \Sigma$ gefärbten Knotens v, der kein Blatt ist, aus den Zuständen seines ersten und zweiten Kindes u_1 und u_2. Dazu weist $\mathbb{A}$ dem Knoten v einen Zustand $q(v) \in Q$ mit $(q(u_1), q(u_2), a, q(v)) \in \Delta$ zu. Wenn schließlich alle Knoten von t durch q mit Zuständen markiert sind, prüft $\mathbb{A}$, ob der Zustand $q(w)$ an der Wurzel w von t zur Menge F gehört. Falls ja, so ist q ein *akzeptierender Lauf*, ansonsten ist q ein *verwerfender Lauf*. Ein Σ-Baum t wird von $\mathbb{A}$ genau dann *akzeptiert*, wenn es (mindestens) einen akzeptierenden Lauf von $\mathbb{A}$ auf t gibt. Die vom Baumautomat $\mathbb{A}$ *erkannte* Baumsprache $L(\mathbb{A})$ ist die Menge aller Σ-Bäume t, die von $\mathbb{A}$ akzeptiert werden. Eine Baumsprache L heißt *regulär*, wenn es einen Baumautomaten $\mathbb{A}$ gibt, der L erkennt.