Prof. T. Nipkow
M. Eberl, J. Rädle, S. Wimmer

Einführung in die theoretische Informatik

Sommersemester 2019 – Knobelblatt 2

Dieses Blatt enthält Knobelaufgaben. Diese Aufgaben werden nicht bepunktet und sind *besonders schwer*. Die Bearbeitung dieses Blattes ist komplett mit Mitteln der Vorlesung möglich, aber die Aufgaben sind in Ihrer Schwierigkeit so, dass ein Mitglied der Übungsleitung längere Zeit nachdenken muss, um Sie zu lösen (oder sogar irgendwann aufgibt). Wir haben aber für alle Aufgaben (hoffentlich richtige) Musterlösungen. Eine grobe Einschätzung der Schwierigkeit ist in der Anzahl der Totenköpfe rechts von der Aufgabe angegeben.

Keine dieser Aufgaben ist klausurrelevant! Bearbeiten Sie diese Aufgaben *nur*, wenn Sie zu viel Freizeit haben und die normalen Übungsaufgaben Sie nicht genug herausfordern!

Die Namen der ersten paar Einsender richtiger Lösungen zu je einer Aufgabe (u.U. auch zu Teilaufgaben wenn signifikant genug) falls gewünscht auf der Webseite veröffentlicht. Unter Umständen gibt es auch einen Sachpreis in Höhe von einer Tafel Schokolade solange der Vorrat reicht. Lösungen sind einzusenden an den “Puzzle-Meister” `manuel.eberl@tum.de`.

KNOBELAUFGABE 2.1. (*Unentscheidbarkeit und* NP-*schwer*)

(a) Sei $A \subseteq \{a\}^*$ eine Sprache auf einem einelementigen Alphabet. Zeigen Sie: Wenn A NP-schwer ist, ist $\mathsf{P} = \mathsf{NP}$.

(b) Folgern Sie: Wenn $\mathsf{P} \neq \mathsf{NP}$ dann gibt es unentscheidbare Probleme, die nicht NP-schwer sind.

KNOBELAUFGABE 2.2. (*WHILE-Programme mit nur einer Variable*)

Sei WHILE_1 die Menge der WHILE-Programme, die nur eine Variable verwenden. Wir verwenden o.B.d.A. für diese Variable den Namen x und bezeichnen die von solch einem Programm p berechnete Funktion mit $\varphi_p : \mathbb{N} \to \mathbb{N} \cup \{\bot\}$ (d.h. wenn vor der Ausführung von p $x = n$ gilt, gilt danach $x = \varphi_p(n)$ falls $\varphi_p(n) \neq \bot$; falls $\varphi_p(n) = \bot$ ist, terminiert das p nicht.

Wir nennen ein WHILE_1-Programm *einfach*, wenn es keine WHILE-Schleifen enthält.

Um Platz zu sparen, schreiben wir

$$\text{IF } x = 0 \text{ THEN } p \text{ ELSE } q \text{ END } =: \ p\,?\,q \qquad\qquad \text{WHILE } x \neq 0 \text{ DO } p \text{ END } =: \ p^* \ .$$

Sie dürfen o.B.d.A. davon ausgehen, dass als Zuweisungen nur $x := x + 1$ und $x := x - 1$ vorkommen und diese als INC und DEC notieren.

(a) Klassifizieren Sie die Funktionen $f : \mathbb{N} \to \mathbb{N} \cup \{\bot\}$, die sich mit einem einfachen WHILE_1-Programm berechnen lassen. Finden Sie hierzu eine möglichst einfache Beschreibung dieser Funktionen.

(b) Überlegen Sie sich eine kompakte Repräsentation für diese Funktionen und geben Sie ein Verfahren an, um für ein gegebenes einfaches WHILE_1-Programm p eine solche Repräsentation von φ_p zu berechnen.

(c) Klassifizieren Sie die Funktionen, die sich von einem WHILE_1-Programm der Form p^* berechnen lassen, wobei p einfach ist. Geben Sie wieder eine entsprechende Repräsentation von φ_{p^*} an und ein Berechnungsverfahren. Insbesondere soll damit $\{p\#n\#c \mid \varphi_{p^*}(n) = c\}$ für alle $n \in \mathbb{N}$ und $c \in \mathbb{N} \cup \{\bot\}$ entscheidbar sein.

(d) Klassifizieren Sie die Funktionen, die sich von einem allgemeinen WHILE_1-Programm p berechnen lassen. Geben Sie wieder ein Berechnungsverfahren an, um eine geeignete Repräsentation von φ_p zu berechnen. Insbesondere soll damit $\{p\#n\#c \mid \varphi_p(n) = c\}$ für alle $n \in \mathbb{N}$ und $c \in \mathbb{N} \cup \{\bot\}$ entscheidbar sein.

(e) Folgern Sie: Das Halteproblem auf WHILE_1-Programmen sowie das Äquivalenzproblem von WHILE_1-Programmen sind entscheidbar.