Theorie der Informatik (CS 206)

M. Helmert, G. Röger | Universität Basel
Frühjahrssemester 2014 | Fachbereich Informatik

Übungsblatt 11

Abgabe: 14. Mai

Hinweis: Für Abgaben, die ausschliesslich mit LaTeX erstellt wurden, gibt es einen Bonuspunkt. Bitte geben Sie nur die resultierende PDF-Datei bzw. einen Ausdruck davon ab.

Aufgabe 11.1 (Transitivität von Reduktionen, 2 Punkte)

Zeigen Sie, dass für beliebige Sprachen A, B und C gilt: Wenn $A \leq B$ und $B \leq C$, dann auch $A \leq C$.

Aufgabe 11.2 (Unentscheidbare Grammatik-Probleme, 2+1.5+1.5 Punkte)

Das *Leerheitsproblem*, *Äquivalenzproblem* und *Schnittproblem* für allgemeine Grammatiken (d.h. Typ-0-Grammatiken) sind wie folgt definiert:

- LEERHEIT: Gegeben eine allgemeine Grammatik G, ist $\mathcal{L}(G) = \emptyset$?
- ÄQUIVALENZ: Gegeben zwei allgemeine Grammatiken G_1 und G_2, ist $\mathcal{L}(G_1) = \mathcal{L}(G_2)$?
- SCHNITT: Gegeben zwei allgemeine Grammatiken G_1 und G_2, gilt $\mathcal{L}(G_1) \cap \mathcal{L}(G_2) = \emptyset$?

Beweisen Sie, dass LEERHEIT, ÄQUIVALENZ und SCHNITT unentscheidbar sind.

Hinweis: Sie dürfen ohne Beweis verwenden, dass man mit einer berechenbaren Funktion zu jeder gegebenen Typ-0-Grammatik G die Kodierung eine DTM $M(G)$ konstruieren kann, für die $\mathcal{L}(M(G)) = \mathcal{L}(G)$ gilt. Wenden Sie den Satz von Rice in geeigneter Weise an, um die Unentscheidbarkeit von LEERHEIT zu zeigen, und reduzieren Sie dann LEERHEIT auf ÄQUIVALENZ und LEERHEIT auf SCHNITT.

Aufgabe 11.3 (Deterministische und nichtdeterministische Algorithmen, 1.5+1.5 Punkte)

Das *Erfüllbarkeitsproblem in der Aussagenlogik* (SAT) ist wie folgt definiert: gegeben eine aussagenlogische Formel φ, ist φ erfüllbar?

(a) Geben Sie einen nichtdeterministischen Algorithmus für SAT an, dessen Laufzeit durch ein Polynom in der Länge von φ beschränkt ist. Begründen Sie, warum die Laufzeit des Algorithmus polynomiell ist.

(b) Geben Sie einen deterministischen Algorithmus für SAT an.

(c) *optionale Zusatzaufgabe für 1 Bonuspunkt:* Schätzen Sie die Laufzeit Ihres Algorithmus aus Teil (b) in O-Notation ab.

Sie dürfen bei Ihrer Antwort beliebige übliche Programmierkonzepte wie z.B. übliche Datenstrukturen oder rekursive Funktionsaufrufe verwenden. Es ist nicht nötig, dass Sie sich auf die eingeschränkte Syntax von WHILE-Programmen o.ä. beschränken.