

Einführung in die Theoretische Informatik

Sommersemester 2020 – Übungsblatt 5

AUFGABE 5.1. (*Cocke-Younger-Kasami*)

0.5P

Lösen Sie die Aufgaben “H5.1 a” und “H5.1 b” im AutomataTutor unter der Kategorie “CYK Algorithm”.
Beachten Sie, dass Sie pro Aufgabe nur 2 Fehler machen dürfen.

AUFGABE 5.2.

1P

Gegeben sei eine Grammatik $G = (V, \Sigma, P, S)$ in Chomsky-Normalform, d.h. jede Produktion in P hat entweder die Form $X \rightarrow YZ$ mit $X, Y, Z \in V$ oder $X \rightarrow a \in \Sigma$. Geben Sie eine Grammatik G_s an, sodass $L(G_s)$ alle nicht-leeren Suffixe der Sprache $L(G)$ enthält. Genauer soll $L(G_s) = \{v \mid uv \in L(G) \wedge v \neq \varepsilon\}$ gelten.

AUFGABE 5.3. (*Dieses doofe Trennzeichen*)

2P

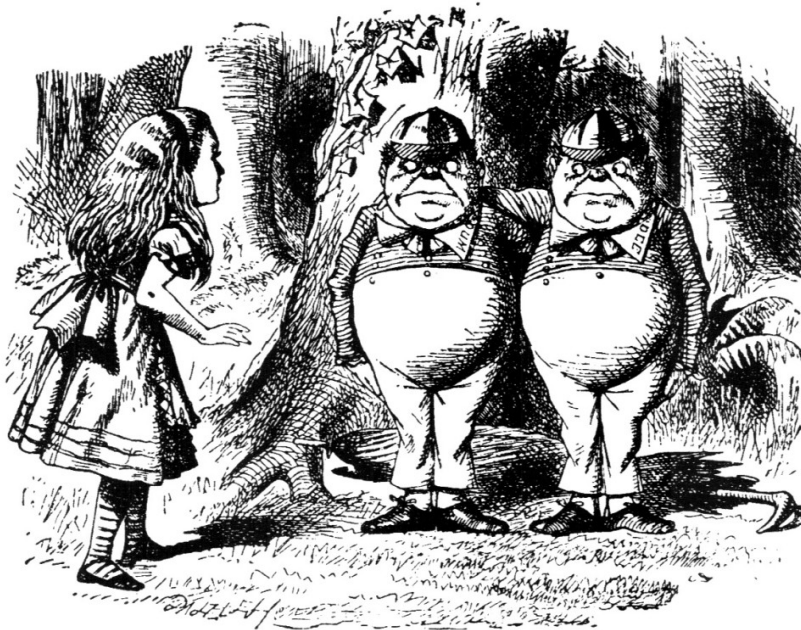
Sei $\Sigma := \{a, b, \nabla\}$. Zeigen oder widerlegen Sie: Die Sprache $L := \{u\nabla v \mid u, v \in \{a, b\}^*, |u| = |v|, u \neq v\}$ ist kontextfrei.

AUFGABE 5.4.

2P

Bei dieser Aufgabe handelt es sich um eine Programmieraufgabe auf TUMjudge(<https://judge.in.tum.de/theo/public/>). Zur Bearbeitung der Hausaufgabe gehen Sie wie folgt vor.

- Lesen Sie sich die PDF-Angabe zu der Aufgabe ‘CFGFinite’ durch (zu finden unter ‘course/problems’ auf TUMjudge).
- Laden Sie das Codegerüst für die Hausaufgabe auf Moodle herunter.
- Wir stellen ein Gerüst in Haskell und in Java bereit.
- Implementieren Sie alle mit `TODO` markierten Methoden in der Klasse `Finiteness`, bzw. die undefinierten Funktionen im Modul `Main` des Haskell Templates.
- Laden Sie dann für Problem A (CFGFinite) alle benötigten Dateien hoch – falls Sie eines unserer Templates benutzen, müssen Sie *alle* Dateien des Templates hochladen, nicht nur die, die Sie verändert haben.



“If it was so, it might be; and if it were so, it would be; but as it isn’t, it ain’t. That’s logic.”

— Lewis Carroll in *Alice hinter den Spiegeln* (*Through the Looking-Glass*)