10. Übungsblatt zu Grundzüge der Theoretischen Informatik, WS 2013/14

Prof. Dr. Gert Smolka, Jonas Kaiser, M.Sc.
www.ps.uni-saarland.de/courses/ti-ws13/

Lesen Sie im Buch: Lecture 22 – Lecture 25 (wdh.), sowie Lecture 26 und Lecture 27

Grundlagen und Wiederholung

Aufgabe 10.1 Sei KK_2 die Sprache aller korrekt geklammerten Wörter mit zwei Sorten von Klammern, nämlich () sowie []. Es gilt beispielhaft $([]) \in KK_2$ und $([)] \notin KK_2$. Geben Sie eine kontextfreie Grammatik für KK_2 an.

Aufgabe 10.2 Entscheiden Sie für jede der folgenden Grammatiken, ob diese die Sprache $GA' = \{x \in \{a,b\}^* \mid \#a(x) = \#b(x) > 0\}$ beschreibt. Geben Sie ein Gegenbeispiel oder eine kurze Rechtfertigung an.

(a)

$$S \longrightarrow \epsilon \mid aSb \mid bSa \mid SS$$

(b)

$$S \longrightarrow aSb \mid bSa \mid SS$$

(c)

$$S \longrightarrow ab \mid ba \mid aSb \mid bSa$$

(d)

$$S \longrightarrow ab \mid ba \mid aSb \mid bSa \mid SS$$

Aufgabe 10.3 Wir betrachten Binärbäume mit beschrifteten Blättern. Die Tiefe eines solchen Baumes ist definiert als die Länge des längstens Pfades (also Anzahl der Kanten) von der Wurzel zu einem Blatt. Zeigen Sie, dass ein Binärbaum der Tiefe n höchstens 2^n Blätter besitzt.

Kontextfreie Sprachen und das Pumping Lemma

Aufgabe 10.4 Kontextfreie Sprachen lassen sich pumpen, ähnlich wie reguläre Sprachen.

(a) Nennen Sie kurz in Ihren eigenen Worten die grundlegende Idee der Pumpeigenschaft von kontextfreien Sprachen.
(b) Beschreiben Sie die Herleitung des Pumplemmas. Stellen Sie sicher, dass Sie alle auftretenden Randbedingungen rechtfertigen können.
(c) Geben Sie nun die formale Definition des Pumplemmas für kontextfreie Sprachen an.

Aufgabe 10.5 Entscheiden Sie für jede der folgenden Sprachen, ob diese kontextfrei ist, oder nicht. Geben Sie zu jeder kontextfreien Sprache eine Grammatik und einen Kellerautomaten ohne Zustände an. Hierfür müssen Sie Ihre Grammatik gegebenenfalls noch in die PDA-Normalform konvertieren. Andernfalls geben Sie einen Beweis, dass die Sprache nicht kontextfrei ist.

(a) $\{a^i b^j c^k d^l \mid i,j,k,l \geq 1, i = j, k = l\}$

(b) $\{a^i b^j c^k d^l \mid i,j,k,l \geq 1, i = k, j = l\}$
(c) $\{a^i b^j c^k d^l \mid i,j,k,l \geq 1, i = l, j = k\}$

Aufgabe 10.6 Sei $W = \{ww \mid w \in \{a,b\}^*\}$ die Wiederholungssprache. Zeigen Sie mithilfe des Pumplemmas für kontextfreie Sprachen, das W nicht kontextfrei ist.

Aufgabe 10.7 Im folgenden Betrachten wir die Sprache $\sim W = \Sigma^* - W$.

(a) Geben Sie eine formale Charakterisierung $P(z)$ der Sprache $\sim W$ an, so dass

$$z \in \sim W \iff P(z)$$

gilt.
(b) Beweisen Sie $z \in \sim W \iff P(z)$.
(c) Nutzen Sie Ihre Definition von $P(z)$ um eine kontextfreie Grammatik für $\sim W$ anzugeben.
(d) Geben Sie einen Kellerautomaten ohne Zustände an, der Ihrer Grammatik entspricht.

Parsing

Aufgabe 10.8 Sei folgende Grammatik für arithmetische Ausdrücke gegeben:

$$S \longrightarrow x \mid S + S \mid S - S \mid S * S \mid (S),$$

wobei wir folgende Präzedenz-Ordnung für die Binäroperatoren festlegen: $* > +, -$. Alle Binäroperatoren sind links-assoziativ.

(a) Geben Sie eine eindeutige Grammatik an.
(b) Geben Sie alle Reduktionsregeln (reduce) eines entsprechenden Shift-Reduce-Parsers an.

Aufgabe 10.9 Sei folgende Grammatik für boolsche Ausdrücke gegeben:

$$S \longrightarrow 0 \mid 1 \mid \neg S \mid S \wedge S \mid S \vee S \mid S \rightarrow S \mid (S),$$

wobei wir folgende Präzedenz-Ordnung für die Binäroperatoren festlegen: $\wedge > \vee > \rightarrow$. Konjunktion und Disjunktion sind links-assoziativ, Implikation ist rechts-assoziativ.

(a) Geben Sie eine eindeutige Grammatik an.
(b) Geben Sie alle Reduktionsregeln (reduce) eines entsprechenden Shift-Reduce-Parsers an.

Aufgabe 10.10 Implementieren Sie den Shift-Reduce-Parser aus Aufgabe 10.8.b in ML. Vervollständigen Sie dafür die Funktion *parse* : *token list* → *item list* → *exp* im folgenden Programfragment:

```
datatype Op = Mul | Add | Sub

datatype exp =
  Var of string
| app of Op * exp * exp

datatype token = VT of string | OT of Op | LP | RP

datatype item = E of exp | O of Op | LPar

fun pre Mul = 2
  | pre Add = 1
  | pre Sub = 1

exception Error


fun parse ...


val x = VT"x"
val y = VT"y"
val z = VT"z"
val min = OT Sub
val mul = OT Mul

val test1 = parse [x, min, y, mul, x, min, z] nil
val test2 = parse [LP, x, RP] nil
val test3 = parse [x, min, y, mul, LP, x, min, z, RP] nil
```

Aufgabe 10.11 (Challenge) Ergänzen Sie die Grammatik aus Aufgabe 10.8 mit einem unären Minus, welches das gleiche Symbol wie die Subtraktion verwendet. Geben Sie nun wieder eine eindeutige Grammatik und die Reduktionsregeln an und implementieren Sie den korrespondierenden Shift-Reduce-Parser. Ihr Parser sollte den Ausdruck $x - -y$ korrekt erkennen können.