

Übungsblatt 3 (Lösungen hochladen bis Montag, 3.11.2014, 16:00 Uhr) Besprechungstermin: 3./5.11.2014

Aufgabe 1 (3 * 4 Punkte)

Betrachten Sie folgenden Grammatiken. Wenn es einen Prädiktiven Parser dafür gibt, schreiben Sie ihn auf, wenn nicht, transformieren Sie die Grammatik in eine äquivalente, zu der Sie dann einen Prädiktiven Parser entwickeln.

- a) $S \rightarrow +S S \mid - S S \mid a$
- b) $S \rightarrow S (S) S \mid \epsilon$
- c) $S \rightarrow 0 S 1 \mid 01$

Aufgabe 2 (6 Punkte)

Verwenden Sie die Grammatik $S \rightarrow S) \mid S (\mid \epsilon$ zur Erzeugung beliebiger Klammernfolgen und entwickeln dazu eine syntaxgerichtete Definition zur Überprüfung, ob ein Wort eine korrekte Klammerung darstellt.

Aufgabe 3 (4 + 2 Punkte)

a) Entwerfen Sie eine kontextfreie Grammatik für römische Zahlen.

I	1	X	10	C	100	M	1000	XVIII	18
II	2	XX	20	CC	200	MM	2000	XXIX	29
III	3	XXX	30	CCC	300	MMM	3000	XXXVI	36
IV	4	XL	40	CD	400	MMMM	4000	XLVIII	48
V	5	L	50	D	500	MMMMM	5000	LV	55
VI	6	LX	60	DC	600	MMMMMM	6000	LXIV	64
VII	7	LXX	70	DCC	700	MMMMMMM	7000	LXXI	71
VIII	8	LXXX	80	DCCC	800	MMMMMMMM	8000	LXXXIX	89
IX	9	XC	90	CM	900	MMMMMMMMM	9000	XCIX	99

b) Prüfen Sie, ob Sie für Ihre Grammatik einen prädiktiven Parser erstellen können. Begründen Sie Ihre Antwort.

Aufgabe 4 (2 * 4 Punkte)

- a) Entwickeln Sie ein Übersetzungsschema, das natürliche Zahlen in römische Zahlen übersetzt.
- b) Entwickeln Sie ein Übersetzungsschema, das römische Zahlen in natürliche Zahlen übersetzt.

bitte wenden

Aufgabe 5 (8 Punkte)

Betrachten Sie folgenden Auszug aus Java Bytecode

Befehl	Opcode	Beschreibung
iconst_m1	02	push -1 auf Stack
iconst_0	03	push 0 auf Stack
iconst_1	04	push 1 auf Stack
iconst_2	05	push 2 auf Stack
iconst_3	06	push 3 auf Stack
iconst_4	07	push 4 auf Stack
iconst_5	08	push 5 auf Stack
iadd	60	Addition auf Stack
isub	64	Subtraktion auf Stack
imul	68	Multiplikation auf Stack
idiv	6c	Division auf Stack
ineg	74	Negation auf Stack

Schreiben Sie für arithmetische Ausdrücke über den Ziffern 0 bis 5 und den Operationen $+$, $-$, $*$, uminus und $/$ einen prädiktiven Übersetzer nach Java Bytecode.