



Übung 2

Aufgabe 1 :

Geben Sie einen Algorithmus in Form eines Aktivitätsdiagramms an, nach dem zwei beliebige Brüche $p = \frac{z_1}{n_1}$ und $q = \frac{z_2}{n_2}$, $p, q \in \mathbb{Q}$ addiert werden zu einem Ergebnisbruch $r = \frac{z_3}{n_3}$. Die beiden Brüche p, q sollen dem Algorithmus als Eingabe übergeben werden. Das Resultat und die Ausgabe ist der Bruch r .

Zur Erinnerung: Bei einem Bruch $\frac{a}{b}$ nennt man a den Zähler und b den Nenner. Beachten Sie, dass auch ungleichnamige p, q Brüche mit $n_1 \neq n_2$ möglich sind.

Aufgabe 2 :

Schauen Sie sich an, welche Arten von Informationen Sie zu Java auf der Webseite finden:

<https://docs.oracle.com/javase/13/>

Aufgabe 3 :

1. Schauen Sie sich die Grammatikregeln zu Hexadezimalkonstanten in Java an (Nichtterminalsymbol $\langle HexNumeral \rangle$). Sie finden die Grammatikregeln dazu in Kapitel 3.10.1 des Javasprachstandards (siehe Link auf unserer Web-Seite zu Java). Hexadezimalkonstanten sind eine andere Darstellungsmöglichkeit für Zahlen, die wir später noch intensiver betrachten werden. Geben Sie mit eigenen Worten an, was diese Definitionen beschreiben.
2. Geben Sie Ableitungen an von $\langle HexNumeral \rangle$
 - (a) zu einem möglichst kurzen Wort nur aus Terminalsymbolen
 - (b) zu einem Wort nur aus Terminalsymbolen, in dessen Ableitung alle Regeln angewandt werden

Aufgabe 4 :

1. In der ersten Vorlesungswoche wurden algorithmische Grundbausteine (Einzelanweisung, Sequenz, Selektion, Mehrfachselektion, Iteration) vorgestellt, die alle auf dem Begriff einer Anweisung aufbauen. Geben Sie eine (vollständige) Grammatik in BNF für Anweisungen an, die alle vorgestellten Grundbausteine abdeckt. Das Startsymbol der Grammatik ist $\langle Anweisung \rangle$ und daraus müssen sich alle Bausteine erzeugen lassen (und nur die). Lassen Sie die rechte Seite der Grammatikregel zu einer Einzelanweisung frei, das wollen wir hier nicht weiter spezifizieren. In einer Selektion benötigen Sie zum Beispiel auch eine Bedingung. Lassen Sie in so einem Fall ebenso die rechte Seite einer Regel für Bedingung frei, also nicht weiter spezifiziert.
Überlegen Sie, was Terminalsymbole dieser Grammatik sind und welche Nichtterminalsymbole Sie benötigen (dies ergibt sich über ihre Regeln).
2. Geben Sie zu Ihren Grammatikregeln eine Ableitung an für das Programm auf Folie 25, wo ja bereits die Struktur des Programms farblich gekennzeichnet ist.

Aufgabe 5 :

1. Geben Sie in EBNF eine kontextfreie Grammatik an für arithmetische Ausdrücke über natürliche Dezimalzahlen, wie aus der Mathematik bekannt. Dabei seien als Operationen nur Additionen, Subtraktionen, Multiplikationen und Divisionen zu berücksichtigen, sowie die Klammerung von (Teil-)Ausdrücken.

Beispiele:

- $(4 + 19 - 6) * (3 * 3 - 5) / 2$
- $((((3 * 2 + 1) * 2 + 1) * 2 + 1) * 2 + 1$
- (617)

Hinweis: Verwenden Sie das aus der Vorlesung bekannte Nichtterminal **<DecimalNumeral>** für Zahlen.

2. Falls Ihre Grammatik aus dem vorangehenden Aufgabenteil dies nicht bereits gewährleistet, passen Sie diese an, so dass sie “Punktrechnung vor Strichrechnung” befolgt. Ein Ausdruck wie

$$5 * 4 + 3$$

soll also in derselben Reihenfolge aufgelöst werden wie

$$(5 * 4) + 3$$

und nicht wie

$$5 * (4 + 3)$$

Hinweis: Führen Sie Nichtterminale **<PunktAusdruck>**, **<StrichAusdruck>** und **<EinzelAusdruck>** für entsprechende (Teil-)Ausdrücke ein.

3. *Dieser Aufgabenteil ist etwas kniffliger!*

Falls Ihre Grammatik aus dem vorangehenden Aufgabenteil dies nicht bereits gewährleistet, passen Sie diese an, so dass sie Auswertung “von links nach rechts” befolgt. Ein Ausdruck wie

$$5 - 4 + 3$$

soll also in derselben Reihenfolge aufgelöst werden wie

$$(5 - 4) + 3$$

und nicht wie

$$5 - (4 + 3)$$