

## Allgemeine Hinweise:

- Die **Deadline** zur **Abgabe** der Hausaufgaben ist am **Donnerstag, den 23.10.2025, um 14 Uhr**.
- Der **Workflow** sieht wie folgt aus. Die Abgabe der Hausaufgaben erfolgt **im Moodle-Lernraum** und kann nur in **Zweiergruppen** stattfinden. Dabei müssen die Abgabepartner\*innen **dasselbe Tutorium** besuchen. Nutzen Sie ggf. das entsprechende **Forum** im Moodle-Lernraum, um eine\*n Abgabepartner\*in zu finden. Es darf **nur ein\*e** Abgabepartner\*in die Abgabe hochladen. Diese\*r muss sowohl die **Lösung** als auch den **Quellcode** der Programmieraufgaben hochladen. Die Be-punktung wird dann von uns für **beide** Abgabepartner\*innen **separat** im Lernraum eingetragen. Die Feedbackdatei ist jedoch nur dort sichtbar, wo die Abgabe hochgeladen wurde und muss innerhalb des Abgabepaars **weitergeleitet** werden.
- Die **Lösung** muss als PDF-Datei hochgeladen werden. Damit die Punkte beiden Abgabepart-ner\*innen zugeordnet werden können, müssen **oben** auf der **ersten Seite** Ihrer Lösung die **Namen**, die **Matrikelnummern** sowie die **Nummer des Tutoriums** von **beiden** Abgabepartner\*innen angegeben sein.
- Einige Hausaufgaben müssen im Spiel **Codescape** gelöst werden. Klicken Sie dazu im Lernraum rechts in der ausklappbaren Seitenleiste im Block „Codescape“ auf den angegebenen Link. Diese Aufgaben werden getrennt von den anderen Hausaufgaben gewertet.

## Übungsaufgabe 1 (Überblickswissen):

- Was passiert beim Kompilieren und Ausführen eines Java-Programms?
- Was ist der Unterschied zwischen der Variable `x`, dem `char`-Wert `'x'` und dem `String`-Wert `"x"`?
- Woran erkennt man bei einer im Zweierkomplement dargestellten Zahl, ob diese positiv oder negativ ist?

## Übungsaufgabe 2 (Syntax und Semantik):

- Die Menge der syntaktisch korrekten einfachen arithmetischen Ausdrücke (**EAA**) wird durch die Gram-matik  $G_1 = (\{S_1\}, \{(\cdot, \cdot), \cdot, \cdot, \text{plus}, s, \mathcal{O}\}, P_1, S_1)$  definiert, wobei  $P_1$  genau die folgenden Produktionsregeln enthält:

$$\begin{aligned} S_1 &\rightarrow \mathcal{O} \\ S_1 &\rightarrow s(S_1) \\ S_1 &\rightarrow \text{plus}(S_1; S_1) \end{aligned}$$

Die Semantik  $\mathcal{W}(\mathcal{A})$  eines syntaktisch korrekten **EAA**s  $\mathcal{A}$  ist wie folgt definiert, wobei  $x$  und  $y$  ebenfalls syntaktisch korrekte **EAA**s sind:

$$\begin{aligned} \mathcal{W}(\mathcal{O}) &= 0 \\ \mathcal{W}(s(x)) &= \mathcal{W}(x) + 1 \\ \mathcal{W}(\text{plus}(x; y)) &= \mathcal{W}(x) + \mathcal{W}(y) \end{aligned}$$

Für alle **EAA**s  $\mathcal{A}$  gilt also  $\mathcal{W}(\mathcal{A}) \in \mathbb{N}$ .

Geben Sie für die folgenden drei Ausdrücke an, ob es sich um einen syntaktisch korrekten **EAA** handelt und welche Semantik er hat.

- $\text{plus}(\mathcal{O}; s(\mathcal{O}); s(s(\mathcal{O})))$
- $\text{plus}(s(s(\mathcal{O})); s(\mathcal{O}))$
- $\text{plus}(s(\text{plus}(s(\mathcal{O}); s(\mathcal{O}))); s(\text{plus}(\mathcal{O}; \mathcal{O})))$

- b) Begründen oder widerlegen Sie: Zwei Ausdrücke einer Sprache mit unterschiedlicher Semantik haben auch immer eine unterschiedliche Syntax.
- c) Begründen oder widerlegen Sie: Ein syntaktisch nicht korrektes Programm kann trotzdem semantisch korrekt sein.

### Hausaufgabe 3 (Syntax und Semantik):

(8 + 5 + 5 = 18 Punkte)

- a) Die Menge der syntaktisch korrekten **SASP** Programme wird durch die Grammatik  $G_2 = (\{A, B, S_2\}, \{., :-, p, q, r, s\}, P_2, S_2)$  definiert, wobei  $P_2$  genau die folgenden Produktionsregeln enthält:

$$\begin{aligned} S_2 &\rightarrow A. \\ S_2 &\rightarrow A. S_2 \\ A &\rightarrow B \\ A &\rightarrow B :- B \\ B &\rightarrow p \\ B &\rightarrow q \\ B &\rightarrow r \\ B &\rightarrow s \end{aligned}$$

Die Semantik  $\mathcal{W}(\mathcal{P})$  eines syntaktisch korrekten **SASP** Programms  $\mathcal{P}$  ist wie folgt definiert, wobei  $\mathcal{P}'$  ebenfalls ein syntaktisch korrektes **SASP** Programm ist und  $x, y \in \{p, q, r, s\}$ :

$$\begin{aligned} \mathcal{W}(x.) &= \{x\} \\ \mathcal{W}(x :- y.) &= \emptyset \\ \mathcal{W}(\mathcal{P}' x.) &= \mathcal{W}(\mathcal{P}') \cup \{x\} \\ \mathcal{W}(\mathcal{P}' x :- y.) &= \begin{cases} \mathcal{W}(\mathcal{P}') \cup \{x\} & \text{falls } y \in \mathcal{W}(\mathcal{P}') \\ \mathcal{W}(\mathcal{P}') & \text{sonst} \end{cases} \end{aligned}$$

Für alle **SASP** Programme  $\mathcal{P}$  gilt also  $\mathcal{W}(\mathcal{P}) \subseteq \{p, q, r, s\}$ .

Geben Sie für die folgenden Ausdrücke an, ob es sich um ein syntaktisch korrektes **SASP** Programm handelt und welche Semantik es hat.

- i)  $r :- s :- p.$       ii)  $q :- r. r. s. p :- s.$       iii)  $t.$       iv)  $p. p :- s. q. r :- q.$

- b) Begründen oder widerlegen Sie: Zwei Ausdrücke (aus möglicherweise verschiedenen Sprachen) haben genau dann die gleiche Syntax, wenn Sie die gleiche Semantik haben.
- c) Begründen oder widerlegen Sie: Zwei Ausdrücke einer Sprache mit unterschiedlicher Syntax haben auch immer eine unterschiedliche Semantik.

### Übungsaufgabe 4 (Formale Sprachen und Grammatiken):

Gegeben sei die folgende Sprache:

$L = \{w \in \{a, b\}^* \mid \text{auf jedes } b \text{ folgen direkt mindestens drei } a \text{ oder auf jedes } a \text{ folgt direkt mindestens ein } b\}$

Die folgenden Wörter sind beispielsweise in der Sprache enthalten:

$b$        $abbbbabb$        $aaaabaaabaaa$        $\varepsilon$

Folgende Wörter sind nicht Bestandteil der Sprache:

*aba      bbbbbbbba      bababa      abbaba*

- Geben Sie eine kontextfreie Grammatik an, welche die Sprache  $L$  erzeugt.
- Geben Sie eine Grammatik in EBNF an, die  $L$  definiert. Ihre Grammatik darf nur aus einer Regel bestehen und diese Regel darf nicht rekursiv sein (d. h. das Nichtterminalsymbol auf der linken Seite darf rechts nicht auftreten).

Um die Lesbarkeit zu erhöhen, dürfen Sie Anführungszeichen um Terminalsymbole weglassen.

- Geben Sie ein Syntaxdiagramm ohne Nichtterminalsymbole an, das die Sprache  $L$  definiert.

### Hausaufgabe 5 (Formale Sprachen und Grammatiken): (5 + 5 + 4 = 14 Punkte)

Gegeben sei die folgende Sprache  $D$  aller korrekt geklammerter Ausdrücke über dem Alphabet  $\Sigma = \{ (, ) \}$ :

$$D = \{ w \in \Sigma^* \mid |w|_{(} = |w|_{)} \text{ und für alle Präfixe } u \text{ von } w \text{ gilt } |u|_{(} \leq |u|_{)} \}$$

Für ein Wort  $w \in \Sigma^*$  und ein Symbol  $s \in \Sigma$  entspricht  $|w|_s$  der Anzahl an Vorkommen von  $s$  in  $w$ .

Die folgenden Wörter sind beispielsweise in der Sprache  $D$  enthalten:

$\varepsilon$        $()()$        $(( ))$        $()(( ))$

Es ist  $w_1 = ( \notin D$  und  $w_2 = ) \notin D$ , da hier  $|w_1|_{(} = 1 \neq 0 = |w_1|_{)}$  bzw.  $|w_2|_{(} = 0 \neq 1 = |w_2|_{)}$  gilt. Weiterhin ist  $w_1 = ()() \notin D$ , da  $|u|_{(} = 2 \not\leq 1 = |u|_{)}$  für den Präfix  $u = ()$  von  $w_1$  gilt.

- Geben Sie eine kontextfreie Grammatik an, welche die Sprache  $D$  erzeugt.
- Geben Sie eine Grammatik in EBNF an, die  $D$  definiert. Ihre Grammatik darf nur aus einer Regel bestehen und in dieser Regel darf das Nichtterminalsymbol der linken Seite höchstens einmal auf der rechten Seite auftreten.
- Geben Sie ein (rekursives) Syntaxdiagramm an, das die Sprache  $D$  definiert.

### Übungsaufgabe 6 (Zweierkomplement):

- Sei  $x$  eine ganze Zahl. Wie unterscheiden sich die Zweierkomplement-Darstellungen von  $x$  und  $-x$ ?
- Erklären Sie im Detail, wie die beiden Ausgaben des folgenden Programms berechnet werden.

```
void main() {
    int zahl = -2147483648;

    IO.println(zahl + 1);
    IO.println(zahl - 1);
}
```

*Hinweis:*  $-2^{31} = -2147483648$

- Welche Zahlen repräsentieren die folgenden Bitfolgen im 5-Bit Zweierkomplement?

00010   10111   11011   01101   10000

### Hausaufgabe 7 (Zweierkomplement):

(10 + 8 = 18 Punkte)

- a) Welche Zahlen repräsentieren die folgenden Bitfolgen im 10-Bit Zweierkomplement?

0011001100 1001001001 1000000000 1110101101 0010110100

- b) Die zwei folgenden Java-Ausdrücke werten jeweils zu `true` aus. Geben Sie dafür jeweils eine Begründung.

1) `2_000_000_000 + 1_000_000_000 + 1_294_967_296 == 0`

2) `-(2_000_000_000 + 147_000_000 + 483_000 + 648) < 0`

#### Hinweise:

- In Java dürfen in einer Zahl beliebig viele `_` zwischen den Ziffern auftreten. Dies verändert den Wert der Zahl nicht, sondern es wird nur verwendet, um die Lesbarkeit der Zahl zu erhöhen. `2_000_000_000` ist also nur eine andere Schreibweise für die Zahl 2000000000.

### Hausaufgabe 8 (Intro, Deck 0 und Deck 1):

(Codescape)

Schließen Sie das Intro und das Tutorial zum Spiel Codescape ab und lösen Sie die Missionen von Deck 0 und Deck 1.

Ihre Lösung für die Codescape Missionen wird nur dann für die Zulassung gezählt, wenn Sie Ihre Lösung vor der einheitlichen Codescape Deadline am Freitag, den 30.01.2026, um 23:59 Uhr abschicken.