

43,5/50

Progra Blatt 01

Tut. 16

Andre Oppenheimer

372217

Bastian Schymik

468244

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 Bewertung 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 Abgabepartner*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.

Hausaufgabe 3 (Syntax und Semantik): 18/18 (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) $p. p :- s. q. r :- q.$

i) Nicht korrekt, nach Produktionsregel müsste innerhalb des Ausdrucks noch ein Punkt sein.

$$\begin{pmatrix} S_2 \rightarrow A. \\ S_2 \rightarrow A.S_1 \end{pmatrix}$$



ii) Korrekt, Semantik: $\{r, s, p\}$ ✓

iii) Nicht korrekt, "t" ist nicht in der Sprache enthalten. ✓

iv) Korrekt, Semantik: $\{p, q, r\}$ ✓

8/8

- 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.

b) gleiche Syntax \Leftrightarrow gleiche Semantik

" \Rightarrow " gilt *nur in derselben Sprache*

" \Leftarrow " gilt nicht, bsp. $(x+1)^2 = x^2 + 2x + 1$, oder $x = x + 1;$
und $x++;$

haben eine unterschiedliche Syntax, aber die selbe Semantik. ✓
5/5

c) unterschiedliche Syntax \Rightarrow unterschiedliche Semantik.

Gilt nicht.

Gleiche Beispiele wie in b widerlegen die Implikation. ✓ 5/5

13/14

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:

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

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.

a) $G = (N, T, P, S)$ mit

$$N = \{ A \}$$

$$T = \{ (,) \}$$

$$P = A \rightarrow \{ A \} A$$

$$A \rightarrow \varepsilon$$

$$S = A$$

warum?

5/5



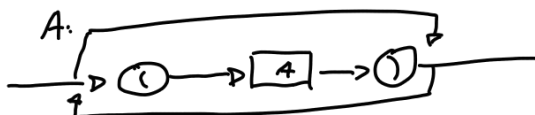
b) $A = \{ "(" A ")" \} [A]$

unnötig

A kommt zweimal vor

4/5

c)



4/4

12,5/18

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.

a) $0011001100 \rightarrow 204$

$1001001001 \rightarrow 0110110110 \rightarrow 0110110111 \rightarrow -439$

$1000000000 \rightarrow 0111111111 \rightarrow 1000000000 \rightarrow -512$

$1110101101 \rightarrow 0001010010 \rightarrow 0001010011 \rightarrow -83$

$0010110100 \rightarrow 180$

✓ 10/10

b)

Standarddatentyp für Zahlen in Java ist `int`, welche die Zahlen von -2^{31} bis $2^{31}-1$ darstellt.

1)

$$2^{31} = 2_147_483_648$$

Das Ergebnis ist:

$$\begin{array}{r} 2_000_000_000 \\ + 1_000_000_000 \\ + 1_294_967_296 \\ \hline 4_294_967_296 = 2 \cdot 2^{31} > 2^{31}, \text{ also kommt es zu einem Overflow.} \end{array}$$

Somit ist das Ergebnis die $2^{31} + 1$ te Zahl, was genau die 0 ist.

2,5/3,5

2)

Das Ergebnis ist:

$$\begin{array}{r} -2_000_000_000 \\ -147_000_000 \\ -483_000 \\ -648 \\ \hline -2_147_483_648 \end{array}$$

zunächst Addition in der Klammer,
dann negieren 0/4,5

$-2_147_483_648 = -2^{31}$, also genau die kleinste durch `int` darstellbare Zahl und somit < 0 .