

**Prüfung zur Vorlesung
„Compilerbau“
WS 2022/23 / 13. Februar 2023**

Vorname: _____

Nachname: _____

Matrikelnummer: _____

Aufgabe	Punktzahl	Erreicht
1	8	
2	12	
3	12	
4	12	
5	6	
6	0	
Σ	50	

Generelle Hinweise:

- Prüfungsdauer: **60 Minuten**.
- Wenn Sie in der Klausur **25 Punkte** erreichen, haben Sie mit Sicherheit bestanden.
- Hilfsmittel: Ein beidseitig beschriebenes DIN-A4-Blatt.
- Schreiben Sie bitte deutlich. Unleserliche Lösungen sind ungültig.
- Benutzen Sie ein **dokumentenechtes Schreibgerät**.
- Überprüfen Sie die Ihnen ausgehändigte Klausur auf Vollständigkeit (**6 Aufgaben** auf 7 Seiten).
- Notieren Sie bitte **auf jedem Blatt**, das Sie verwenden, Ihren **Namen** und Ihre **Matrikelnummer**.
- Schreiben Sie Ihre Lösungen in die dafür vorgesehenen Felder. Reicht der Platz in einem Feld nicht aus, so benutzen Sie die Rückseite des entsprechenden Blattes und vermerken Sie dies auf der Vorderseite. Reicht der Platz dennoch nicht aus, können Sie die Aufsicht nach zusätzlichen Blättern fragen.
- Ein **Täuschungsversuch** führt umgehend zum Ausschluss und **Nichtbestehen**. Es erfolgt keine Vorwarnung.
- Alle mitgeführten **elektronischen Geräte** sind vor der Klausur bzw. spätestens jetzt auszuschalten.

Zur Erinnerung:

Definition. Sei $k \in \mathbb{N}$. Eine Grammatik $G = (N, \Sigma, P, S)$ heißt LL(k)-Grammatik, wenn für jede Linksableitung $S \rightarrow^* wA\beta$, wobei $w \in \Sigma^*$, $A \in N$ und $\beta \in (\Sigma \cup N)^*$ gilt: Für jedes Paar von Produktionen $A \rightarrow \alpha_1, A \rightarrow \alpha_2 \in P$ mit $\alpha_1 \neq \alpha_2$ gilt, dass $\text{First}_k(\alpha_1\beta) \cap \text{First}_k(\alpha_2\beta) = \emptyset$.

Name:

Matrikelnummer:

Aufgabe 1. (8 Punkte) Beantworten Sie die folgenden Fragen. Jede Teilaufgabe bringt Ihnen bei vollständiger und korrekter Beantwortung zwei Punkte.

- (1) Beschreiben Sie in kurzen und einfachen Worten, was der Tokenizing-Algorithmus macht und geben Sie die Laufzeit dieses Algorithmus', sowie die Laufzeit von Reps' Maximal-Munch-Algorithmus an (bei Eingabewort w und Zustandsmenge Q des Eingabe-DFAs).

Lösung: Der Tokenizing-Algorithmus wandelt den Programmcode (String) in einen Tokenstrom um. Es werden maximale Präfixe innerhalb des Programmcodes ermittelt und entsprechende Aktionen des Scanners ausgeführt.

Laufzeit vom Tokenizing-Algorithmus: $\mathcal{O}(|w|^2)$

Laufzeit von Reps' Maximal-Munch-Algorithmus: $\mathcal{O}(|w| \cdot |Q|)$

- (2) Gibt es für jede kontextfreie Sprache nur endlich viele verschiedene reduzierte Grammatiken? Begründen Sie Ihre Antwort. Die Umbenennung der Symbole zählt nicht.

Lösung: Nein, es gibt sogar unendlich viele reduzierte Grammatiken für jede kontextfreie Sprache. Reduziert bedeutet, jedes Nichtterminal innerhalb der Grammatik ist produktiv und erreichbar. Das allein hindert einen aber nicht daran beliebig viele „überflüssige“ Nichtterminale einzubauen: Sei G eine kontextfreie Grammatik mit Startsymbol S . Wir führen n viele neue Nichtterminale A_1, \dots, A_n ein mit Produktionen $A_i \rightarrow A_{i+1}$ ($1 \leq i \leq n-1$) und $A_n \rightarrow S$, sowie neuem Startsymbol A_1 . Dies definiert uns eine (reduzierte) Grammatik G' mit $L(G') = L(G)$.

Die Idee ist ausreichend für die volle Punktzahl. Es muss kein vollständiger Beweis geliefert werden.

- (3) Sei Σ ein Alphabet. Geben Sie eine Sprache $L \subseteq \Sigma^*$ an mit

$$\text{First}_{10}(L) \cap \text{First}_{11}(L) = \emptyset.$$

Lösung: Da Alphabete nicht-leer sein müssen, ist o.B.d.A. $a \in \Sigma$ und damit ist $L = \{a^{11}\}$ ein passendes Beispiel.

- (4) Sortieren Sie die folgenden Grammatik-Klassen so, dass die erstgenannte Klasse in der nächstgenannten enthalten ist: LL(1)-Grammatiken, kontextfreie Grammatiken, reguläre Grammatiken, stark-LL(5)-Grammatiken, deterministisch-kontextfreie Grammatiken¹.

Lösung: reguläre Gr., LL(1)-Gr., stark-LL(5)-Gr., deterministisch-kontextfreie Gr., kontextfreie Gr.

¹Eine Grammatik G heißt deterministisch-kontextfrei, wenn ein deterministischer Kellerautomat K existiert mit $L(G) = L(K)$ und G kann aus K abgeleitet werden.

Name:

Matrikelnummer:

Aufgabe 2. (12 Punkte) Sei $r \in \mathcal{E}_{\{a,b,c\}}$ mit $r = c^* | ([2, a][3, b]^+)?$. Konstruieren Sie den Berry-Sethi-Automaten zu r . Geben Sie für jeden Teilausdruck die Werte der Funktionen `empty`, `first`, `last` und `next` an. Sie dürfen die Werte auch in einen gut beschrifteten Syntaxbaum schreiben.

Lösung: Durchnummerieren der Terminalzeichen:

$$r' := \text{num}(r) = [1, c]^* | ([2, a][3, b]^+)?$$

$$\text{empty}(r') = \text{empty}([1, c]^*) \vee \text{empty}([2, a][3, b]^+)? = t$$

$$\begin{array}{l} \text{└─ empty}([1, c]^*) = t \\ \quad \text{└─ empty}([1, c]) = f \\ \text{└─ empty}([2, a][3, b]^+)? = t \\ \quad \text{└─ empty}([2, a][3, b]^+) = \text{empty}([2, a]) \wedge \text{empty}([3, b]^+) = f \\ \quad \quad \text{└─ empty}([2, a]) = f \\ \quad \quad \text{└─ empty}([3, b]^+) = \text{empty}([3, b]) = f \end{array}$$

$$\text{first}(r') = \text{first}([1, c]^*) \cup \text{first}([2, a][3, b]^+)? = \{[1, c], [2, a]\}$$

$$\begin{array}{l} \text{└─ first}([1, c]^*) = \text{first}([1, c]) = \{[1, c]\} \\ \text{└─ first}([2, a][3, b]^+)? = \text{first}([2, a][3, b]^+) = \text{first}([2, a]) = \{[2, a]\} \\ \quad \text{└─ first}([3, b]^+) = \text{first}([3, b]) = \{[3, b]\} \end{array}$$

$$\text{last}(r') = \text{last}([1, c]^*) \cup \text{last}([2, a][3, b]^+)? = \{[1, c], [3, b]\}$$

$$\begin{array}{l} \text{└─ last}([1, c]^*) = \text{last}([1, c]) = \{[1, c]\} \\ \text{└─ last}([2, a][3, b]^+)? = \text{last}([2, a][3, b]^+) = \text{last}([3, b]^+) = \text{last}([3, b]) = \{[3, b]\} \\ \quad \text{└─ last}([2, a]) = \{[2, a]\} \end{array}$$

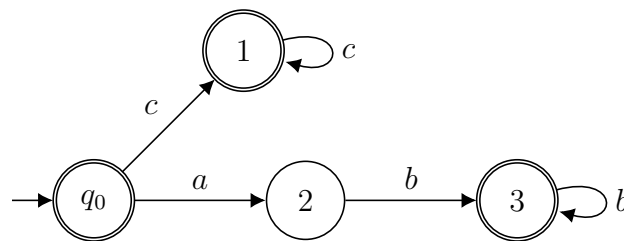
Lösung:

$$\text{next}(r') = \emptyset$$

$$\left\{ \begin{array}{l} \text{next}([1, c]^*) = \text{next}(r') = \emptyset \\ \quad \left\{ \begin{array}{l} \text{next}([1, c]) = \text{first}([1, c]) = \{[1, c]\} \end{array} \right. \end{array} \right.$$

$$\left\{ \begin{array}{l} \text{next}([2, a][3, b]^+) = \text{next}(r') = \emptyset \\ \quad \left\{ \begin{array}{l} \text{next}([2, a][3, b]^+) = \text{next}([2, a][3, b]^+) = \emptyset \\ \quad \left\{ \begin{array}{l} \text{next}([2, a]) = \text{first}([3, b]^+) = \{[3, b]\} \\ \text{next}([3, b]^+) = \text{next}([2, a][3, b]^+) = \emptyset \\ \quad \left\{ \begin{array}{l} \text{next}([3, b]) = \text{first}([3, b]) = \{[3, b]\} \end{array} \right. \end{array} \right. \end{array} \right.$$

Damit ergibt sich der folgende NFA:



Name:

Matrikelnummer:

Aufgabe 3. (12 Punkte) Sei $G = (\{A, B, C, D, E\}, \{a\}, P, A)$, wobei P gegeben ist durch

$$A \rightarrow Da \mid DE$$

$$B \rightarrow aC$$

$$D \rightarrow aa$$

$$E \rightarrow ABC \mid AD$$

Reduzieren Sie die Grammatik, indem Sie den Algorithmus aus der Vorlesung verwenden. Geben Sie sowohl die Grammatik nach dem Entfernen nicht erreichbarer, als auch die Grammatik nach dem Entfernen nicht produktiver Nichtterminale an.

Lösung:

	$A \rightarrow Da$	$A \rightarrow DE$	$B \rightarrow aC$	$D \rightarrow aa$	$E \rightarrow ABC$	$E \rightarrow AD$	W	R
	1	2	1	0	3	2	$\{D \rightarrow aa\}$	\emptyset
$D \rightarrow aa$	0	1	1	0	3	1	$\{A \rightarrow Da\}$	$\{D\}$
$A \rightarrow Da$	0	1	1	0	2	0	$\{E \rightarrow AD\}$	$\{A, D\}$
$E \rightarrow AD$	0	0	1	0	2	0	$\{A \rightarrow DE\}$	$\{A, D, E\}$
$A \rightarrow DE$	0	0	1	0	2	0	\emptyset	$\{A, D, E\}$

Grammatik nach dem Entfernen nicht produktiver Nichtterminale:

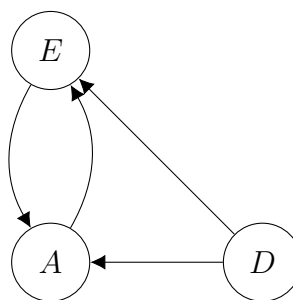
$G' = (\{A, D, E\}, \{a\}, P', A)$, wobei P' gegeben ist durch:

$$A \rightarrow Da \mid DE$$

$$D \rightarrow aa$$

$$E \rightarrow AD$$

Erreichbarkeitsgraph:



Alle Nichtterminale sind erreichbar, also ist die Grammatik nach dem Entfernen nicht erreichbarer Nichtterminale gleich G' .

Name:

Matrikelnummer:

Aufgabe 4. (12 Punkte) Sei $G = (N, \{+, \langle, \rangle, a, b\}, P, S)$, wobei $N = \{S, A, B\}$ und P gegeben ist durch

$$S \rightarrow \langle A \rangle \mid \varepsilon$$

$$A \rightarrow S+B \mid ab$$

$$B \rightarrow aB+ \mid bS$$

(a) Geben Sie $\text{First}_1(X)$ für jedes $X \in N$ an.

Lösung:

$$\text{First}_1(S) = \{\varepsilon, \langle\}$$

$$\text{First}_1(A) = \{\langle, +, a\}$$

$$\text{First}_1(B) = \{a, b\}$$

(b) Geben Sie $\text{Follow}_1(X)$ für jedes $X \in N$ an.

Lösung:

$$\text{Follow}_1(S) = \{\varepsilon, +, \rangle\}$$

$$\text{Follow}_1(A) = \{\rangle\}$$

$$\text{Follow}_1(B) = \{\rangle, +\}$$

(c) Geben Sie die Vorausschautabelle für stark LL(1) an.

Lösung:

$$S \rightarrow \langle A \rangle: \text{First}_1(\langle A \rangle) \odot_1 \text{Follow}_1(S) = \{\langle\}$$

$$S \rightarrow \varepsilon: \text{First}_1(\varepsilon) \odot_1 \text{Follow}_1(S) = \{\varepsilon, +, \rangle\}$$

$$A \rightarrow S+B: \text{First}_1(S+B) \odot_1 \text{Follow}_1(A) = \{+, \langle\}$$

$$A \rightarrow ab: \text{First}_1(ab) \odot_1 \text{Follow}_1(A) = \{a\}$$

$$B \rightarrow aB+: \text{First}_1(aB+) \odot_1 \text{Follow}_1(B) = \{a\}$$

$$B \rightarrow bS: \text{First}_1(bS) \odot_1 \text{Follow}_1(B) = \{b\}$$

	+	⟨	⟩	a	b	ε
S	$S \rightarrow \varepsilon$	$S \rightarrow \langle A \rangle$	$S \rightarrow \varepsilon$			$S \rightarrow \varepsilon$
A	$A \rightarrow S+B$	$A \rightarrow S+B$		$A \rightarrow ab$		
B				$B \rightarrow aB+$	$B \rightarrow bS$	

Name:

Matrikelnummer:

Aufgabe 5. (6 Punkte) Sei $G = (\{S, A, B, C, D\}, \{a, b, *\}, P, S)$, wobei P gegeben ist durch:

$$\begin{aligned} S &\rightarrow aA \mid aB \\ A &\rightarrow aCC \\ B &\rightarrow CD \\ C &\rightarrow Cb \mid Cbb \mid \varepsilon \\ D &\rightarrow *C \end{aligned}$$

Geben Sie eine LL(1)-Grammatik G' an mit $L(G') = L(G)$.

Lösung: $G' = (\{S, C, D, E\}, \{a, b, *\}, P', S)$, wobei P' gegeben ist durch:

$$\begin{aligned} S &\rightarrow aE \\ E &\rightarrow aC \mid CD \\ C &\rightarrow bC \mid \varepsilon \\ D &\rightarrow *C \end{aligned}$$

Erklärungen: Das Nichtterminal C erzeugt b^* , somit vereinfachen sich die Produktionen bei C und das zweite C bei der Produktion von A kann man weglassen. Beide Grammatiken erzeugen $\{aab^n \mid n \geq 0\} \cup \{ab^n * b^m \mid n, m \geq 0\}$ (muss nicht angegeben werden). Außerdem ist G' in der Tat LL(1).

Name:

Matrikelnummer:

Aufgabe 6. (8 Punkte (Bonus)) Sei $G = (N, \{a, \langle, \rangle\}, P, S)$, wobei $N = \{S, A, B\}$ und P gegeben ist durch:

$$\begin{aligned} S &\rightarrow AB \mid \varepsilon \\ A &\rightarrow aS \mid BS \\ B &\rightarrow \langle A \rangle \end{aligned}$$

Berechnen Sie $\text{Follow}_1(X)$ für alle $X \in N$ mit dem Algorithmus aus der Vorlesung. Stellen Sie dazu das Ungleichungssystem auf und vereinfachen Sie dieses so weit wie möglich. Sie dürfen für Ihre Rechnungen die Tabellenform benutzen.

Hinweis: Der Algorithmus terminiert nach 4 Schritten. Außerdem können Sie die folgenden Informationen verwenden:

$$\begin{aligned} \text{First}_1(S) &= \{\varepsilon, a, \langle\} \\ \text{First}_1(A) &= \{a, \langle\} \\ \text{First}_1(B) &= \{\langle\} \end{aligned}$$

Lösung: Wir haben das Ungleichungssystem:

$$\begin{aligned} \text{Follow}_1(S) &\supseteq \{\varepsilon\} \cup \text{First}_1(\varepsilon) \odot_1 \text{Follow}_1(A) \\ &\quad \cup \text{First}_1(\varepsilon) \odot_1 \text{Follow}_1(A) \\ \text{Follow}_1(A) &\supseteq \text{First}_1(B) \odot_1 \text{Follow}_1(S) \\ &\quad \cup \text{First}_1(\langle) \odot_1 \text{Follow}_1(B) \\ \text{Follow}_1(B) &\supseteq \text{First}_1(\varepsilon) \odot_1 \text{Follow}_1(S) \\ &\quad \cup \text{First}_1(S) \odot_1 \text{Follow}_1(A) \end{aligned}$$

Wir formen das System um und erhalten:

$$\begin{aligned} \text{Follow}_1(S) &\supseteq \{\varepsilon\} \cup \text{Follow}_1(A) \\ \text{Follow}_1(A) &\supseteq \{\langle\} \odot_1 \text{Follow}_1(S) \cup \{\langle\} \odot_1 \text{Follow}_1(B) \\ \text{Follow}_1(B) &\supseteq \text{Follow}_1(S) \cup \{\varepsilon, a, \langle\} \odot_1 \text{Follow}_1(A) \end{aligned}$$

Damit ergibt sich folgende Tabelle:

Schritt	0	1	2	3	4
$\text{Follow}_1(S)$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\{\varepsilon, \langle\}$	$\{\varepsilon, \langle, \rangle\}$	$\{\varepsilon, \langle, \rangle\}$
$\text{Follow}_1(A)$	\emptyset	$\{\langle\}$	$\{\langle, \rangle\}$	$\{\langle, \rangle\}$	$\{\langle, \rangle\}$
$\text{Follow}_1(B)$	\emptyset	$\{\varepsilon\}$	$\{\varepsilon, a, \langle\}$	$\{\varepsilon, a, \langle, \rangle\}$	$\{\varepsilon, a, \langle, \rangle\}$

Da sich von Schritt 3 auf 4 nichts verändert hat, terminiert der Algorithmus und es gilt $\text{Follow}_1(S) = \{\varepsilon, \langle, \rangle\}$, $\text{Follow}_1(A) = \{\langle, \rangle\}$ und $\text{Follow}_1(B) = \{\varepsilon, a, \langle, \rangle\}$.