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	
$\boldsymbol{\Sigma}$	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 $S \to^* wA\beta$, wobei $w \in \Sigma^*$, $A \in N$ und $\beta \in (\Sigma \cup N)^*$ gilt: Für jedes Paar von Produktionen $A \to \alpha_1, A \to \alpha_2 \in P$ mit $\alpha_1 \neq \alpha_2$ gilt, dass* $\mathrm{First}_k(\alpha_1\beta) \cap \mathrm{First}_k(\alpha_2\beta) = \emptyset$.

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).

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

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

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

(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 Grammatiken[1].

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

Aufgabe 2. (12 Punkte) Sei $r \in \mathcal{E}_{\{a,b,c\}}$ mit $r = c^* \,|\, (ab^+)?$. 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.

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

$$\begin{aligned} A &\rightarrow Da \mid DE \\ B &\rightarrow aC \\ D &\rightarrow aa \\ E &\rightarrow ABC \mid AD \end{aligned}$$

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.

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

$$\begin{aligned} S &\to \langle A \rangle \mid \varepsilon \\ A &\to S{+}B \mid ab \\ B &\to aB{+} \mid bS \end{aligned}$$

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

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

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

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

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

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

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

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

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:

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