Prüfung zur Vorlesung „Compilerbau“

SS 2022 / 08. August 2022

Vorname: ______________________________

Nachname: ______________________________

Matrikelnummer: ______________________________

Aufgabe	**Punktzahl**	**Erreicht**
1	10	
2	13	
3	7	
4	12	
5	8	
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. (10 Punkte) Beantworten Sie die folgenden Fragen. Jede Teilaufgabe bringt Ihnen bei vollständiger und korrekter Beantwortung zwei Punkte.

(1) Was ist die Aufgabe eines Compilers? Nutzen alle heute gängigen Programmiersprachen einen Compiler (Begründung)?

Lösung: Der Compiler generiert aus einem Programmtext (Menschensprache) einen Code (Maschinenprogramm). Das Maschinenprogramm wiederum kann auf eine Eingabe (effizient) angewendet werden.

Es gibt auch Interpretersprachen wie Python, die keinen Compiler nutzen.

(2) Welche Arten der Laufzeitverbesserung oder Speicherplatzoptimierung bei der Implementierung von DFAs kenenn Sie? Nennen Sie mindestens 2.

Lösung:
DFA Minimierung, Displacementverfahren

(3) Bei dem Algorithmus, der eine kontextfreie Grammatik G in eine reduzierte kontextfreie Grammatik G' überführt, werden erst nicht-produktive und dann nicht-erreichbare Nichtterminale entfernt. Was passiert, wenn Sie die Reihenfolge tauschen?

Lösung: Der Algorithmus arbeitet nicht mehr korrekt. Beispielsweise:
$G = (\{S, A, B\}, \{a, b\}, P, S)$, $P = \{S \to AS \mid b, A \to BA, B \to a\}$
Entfernt man zunächst nicht-erreichbare Nichtterminale, so ändert sich nichts. Entfernt man anschließend auch nicht-produktive Nichtterminale, so ist B nicht mehr erreichbar. Dies darf jedoch nicht passieren.

(4) Sei $G = (\{S, A, B, C\}, \{a, b, c\}, P, S)$ eine kontextfreie Grammatik, wobei P gegeben ist durch $\{S \to ABC, A \to a \mid aA, B \to b \mid bB, C \to BAc\}$. Bestimmen Sie die Kardinalität der Übergangsrelation δ des Item-Kellerautomaten $M_G^{(2)}$.

Lösung:
Expansionen: $1 \cdot 1 + 3 \cdot 2 + 3 \cdot 2 + 1 \cdot 1 = 14$
Shift-Übergänge: 5
Reduce-Übergänge: 14 (wie bei Expansionen)
Ingesamt: $14 + 5 + 14 = 33$

(5) Sei M die Menge bestehend aus allen Grammatiken G, bei der jedes Nichtterminal höchstens 2 Produktionen hat. Gibt es ein k, so dass jedes G aus M eine LL(k)-Grammatik ist? Begründen Sie!

Lösung: Nein. Gegenbeispiel: Sei $k \in \mathbb{N}$ und
$G = (\{S\}, \{a\}, \{S \to a^k \mid a^{k+1}\}, S)$.
G ist nicht in LL(k).

Aufgabe 2. (13 Punkte) Sei $r \in \mathcal{E}_{\{a,b,c\}}$ mit $r = (a^? b)^* \mid ca$. 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:

$$r' := \mathrm{num}(r) = ([1,a]^?[2,b])^* \mid [3,c][4,a].$$

$\mathrm{empty}(r') = \mathrm{empty}(([1,a]^?[2,b])^*) \vee \mathrm{empty}([3,c][4,a]) = t$

- $\mathrm{empty}(([1,a]^?[2,b])^*) = t$
  - $\mathrm{empty}([1,a]^?[2,b]) = \mathrm{empty}([1,a]^?) \wedge \mathrm{empty}([2,b]) = f$
    - $\mathrm{empty}([1,a]^?) = t$
      - $\mathrm{empty}([1,a]) = f$
    - $\mathrm{empty}([2,b]) = f$
- $\mathrm{empty}([3,c][4,a]) = \mathrm{empty}([3,c])) \wedge \mathrm{empty}([4,a])) = f$
  - $\mathrm{empty}([3,c])) = f$
  - $\mathrm{empty}([4,a])) = f$

$\mathrm{first}(r') = \mathrm{first}(([1,a]^?[2,b])^*) \cup \mathrm{first}([3,c][4,a]) = \{[1,a],[2,b],[3,c]\}$

- $\mathrm{first}(([1,a]^?[2,b])^*) = \mathrm{first}([1,a]^?[2,b]) = \mathrm{first}([1,a]^?) \cup \mathrm{first}([2,b]) = \{[1,a],[2,b]\}$
  - $\mathrm{first}([1,a]^?) = \mathrm{first}([1,a]) = \{[1,a]\}$
  - $\mathrm{first}([2,b]) = \{[2,b]\}$
- $\mathrm{first}([3,c][4,a]) = \mathrm{first}([3,c]) = \{[3,c]\}$
  - $\mathrm{first}([4,a])) = \{[4,a]\}$

Lösung:

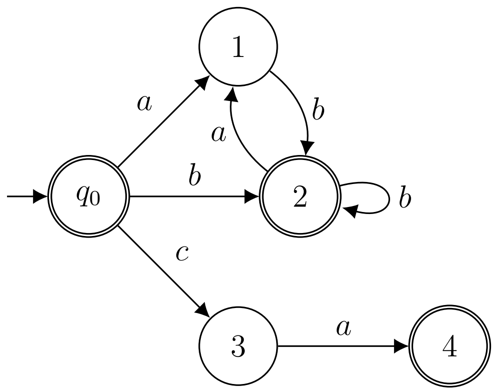
$\mathrm{last}(r') = \mathrm{last}(([1,a]^?[2,b])^*) \cup \mathrm{last}([3,c][4,a]) = \{[2,b],[4,a]\}$

- $\mathrm{last}(([1,a]^?[2,b])^*) = \mathrm{last}([1,a]^?[2,b]) = \mathrm{last}([2,b]) = \{[2,b]\}$
  - $\mathrm{last}([1,a]^?) = \mathrm{last}([1,a]) = \{[1,a]\}$
- $\mathrm{last}([3,c][4,a]) = \mathrm{last}([4,a]) = \{[4,a]\}$
  - $\mathrm{last}([3,c])) = \{[3,c]\}$

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

- $\mathrm{next}(([1,a]^?[2,b])^*) = \mathrm{next}(r') = \emptyset$
  - $\mathrm{next}([1,a]^?[2,b]) = \mathrm{first}([1,a]^?[2,b]) \cup \mathrm{next}(([1,a]^?[2,b])^*) = \{[1,a],[2,b]\}$
    - $\mathrm{next}([1,a]^?) = \mathrm{first}([2,b]) = \{[2,b]\}$
      - $\mathrm{next}([1,a]) = \mathrm{next}([1,a]^?) = \{[2,b]\}$
    - $\mathrm{next}([2,b]) = \mathrm{next}([1,a]^?[2,b]) = \{[1,a],[2,b]\}\}$
- $\mathrm{next}([3,c][4,a]) = \mathrm{next}(r') = \emptyset$
  - $\mathrm{next}([3,c]) = \mathrm{first}([4,a]) = \{[4,a]\}$
  - $\mathrm{next}([4,a]) = \mathrm{next}([3,c][4,a]) = \emptyset$

Damit ergibt sich der folgende NFA:



Aufgabe 3. (7 Punkte) Gegeben sei folgende Übergangsfunktion eines DFAs:

	0	1	2	3	4	5	6	7
a	1	4	1	6	1	5	2	1
b	3	1	0	1	1	4	1	1
c	1	7	1	1	1	0	1	1

Wenden Sie das Displacement-Verfahren an, um eine Übergangstabelle mit nur einer Zeile zu erhalten. Geben Sie die displacement-Funktion sowie die resultierende Tabelle inklusive der valid-Zeile an. Wählen Sie den Default-Wert sinnvoll.

Lösung: Wir wählen Default = 1 und erhalten

	0	1	2	3	4	5	6	7
a		4		6		5	2	
b	3		0			4		
c		7				0		

Dann verschieben wir die Zeile für a um 3 nach rechts und c um 2 nach rechts und erhalten

	0	1	2	3	4	5	6	7	8	9
a					4		6		5	2
b	3		0			4				
c				7				0		

Damit ergibt sich displacement$(a) = 3$, displacement$(b) = 0$, displacement$(c) = 2$ und

	0	1	2	3	4	5	6	7	8	9
A	3		0	7	4	4	6	0	5	2
valid	b		b	c	a	b	a	c	a	a

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

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

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

Lösung:

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

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

Lösung:

$$\begin{aligned} \text{Follow}_1(S) &= \{\varepsilon, a, *, \rangle, +\} \\ \text{Follow}_1(A) &= \{+, *\} \\ \text{Follow}_1(B) &= \{\rangle, +, *\} \end{aligned}$$

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

Lösung:

$$\begin{aligned} S \to \langle A{+}B\rangle &: \text{First}_1(\langle A{+}B\rangle) \odot_1 \text{Follow}_1(S) = \{\langle\} \\ S \to aSa &: \text{First}_1(aSa) \odot_1 \text{Follow}_1(S) = \{a\} \\ A \to +A{*}B &: \text{First}_1(+A{*}B) \odot_1 \text{Follow}_1(A) = \{+\} \\ A \to aB &: \text{First}_1(aB) \odot_1 \text{Follow}_1(A) = \{a\} \\ B \to S{*}S &: \text{First}_1(S{*}S) \odot_1 \text{Follow}_1(B) = \{\langle, a\} \\ B \to \varepsilon &: \text{First}_1(\varepsilon) \odot_1 \text{Follow}_1(B) = \{\rangle, +, *\} \end{aligned}$$

	$\langle$	$\rangle$	$+$	$*$	a	ε
S	$S \to \langle A{+}B\rangle$				$S \to aSa$	
A			$A \to +A{*}B$		$A \to aB$	
B	$B \to S{*}S$	$B \to \varepsilon$	$B \to \varepsilon$	$B \to \varepsilon$	$B \to S{*}S$	

Aufgabe 5. (8 Punkte) Sei $G = (\{S, A, B, C\}, \{a, b, *, \langle, \rangle\}, P, S)$, wobei P gegeben ist durch:

$$\begin{aligned} S &\to \langle A\rangle \mid \langle B\rangle \\ A &\to aAa \mid aa* \\ B &\to bbC \mid \varepsilon \\ C &\to *S* \end{aligned}$$

(a) Geben Sie $k \in \mathbb{N}$ an mit $G \notin \mathrm{LL}(k)$ und $G \in \mathrm{LL}(k+1)$. Es genügt eine kurze Begründung, Sie brauchen keinen formalen Beweis anzugeben.

Lösung: $k = 2$
Wir benötigen als Lookahead mindestens 3 Zeichen, um zu wissen, welche von beiden Produktionen wir bei A nehmen. Für S benötigen wir mindestens 2 Zeichen, aber es gilt ohnehin $3 > 2$.
Somit ist $G \in \mathrm{LL}(3)$, aber $G \notin \mathrm{LL}(2)$.

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

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

$$\begin{aligned} S &\to \langle D\rangle \\ D &\to A \mid B \\ A &\to aaE \\ E &\to aEa \mid * \\ B &\to bbC \mid \varepsilon \\ C &\to *S* \end{aligned}$$

Erklärung zu E: In der Grammatik G erzeugt A alle Wörter der Form $a^n(aa*)a^n (n \geq 0)$. Wir können die Wörter umschreiben zu $aa(a^n{*}a^n)$.

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

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

Berechnen Sie $\mathrm{First}_1(X)$ 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 5 Schritten.

Lösung: Wir haben das Ungleichungssystem:

$$\begin{aligned} \mathrm{First}_1(S) &\supseteq \mathrm{First}_1(S{+}S) \cup \mathrm{First}_1(A{*}B) \\ \mathrm{First}_1(A) &\supseteq \mathrm{First}_1(\langle B \rangle) \cup \mathrm{First}_1(\varepsilon) \\ \mathrm{First}_1(B) &\supseteq \mathrm{First}_1(aB) \cup \mathrm{First}_1({+}A{+}) \end{aligned}$$

Wir formen das System um und erhalten:

$$\begin{aligned} \mathrm{First}_1(S) &\supseteq (\mathrm{First}_1(S) \odot_1 \{+\} \odot_1 \mathrm{First}_1(S)) \cup (\mathrm{First}_1(A) \odot_1 \{*\} \odot_1 \mathrm{First}_1(B)) \\ \mathrm{First}_1(A) &\supseteq (\{\langle\} \odot_1 \mathrm{First}_1(B) \odot_1 \{\rangle\}) \cup \{\varepsilon\} \\ \mathrm{First}_1(B) &\supseteq (\{a\} \odot_1 \mathrm{First}_1(B)) \cup (\{+\} \odot_1 \mathrm{First}_1(A) \odot_1 \{+\}) \end{aligned}$$

Damit ergibt sich folgende Tabelle:

Schritt	0	1	2	3	4	5
$\mathrm{First}_1(S)$	$\emptyset$	$\emptyset$	$\emptyset$	$\{*\}$	$\{*, \langle\}$	$\{*, \langle\}$
$\mathrm{First}_1(A)$	$\emptyset$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\{\varepsilon, \langle\}$	$\{\varepsilon, \langle\}$	$\{\varepsilon, \langle\}$
$\mathrm{First}_1(B)$	$\emptyset$	$\emptyset$	$\{+\}$	$\{+, a\}$	$\{+, a\}$	$\{+, a\}$

Da sich von Schritt 4 auf 5 nichts verändert hat, terminiert der Algorithmus und es gilt $\mathrm{First}_1(S) = \{*, \langle\}$, $\mathrm{First}_1(A) = \{\varepsilon, \langle\}$ und $\mathrm{First}_1(B) = \{+, a\}$.