

---

Aufgaben zur Klausur **Compilerbau** im WS 2018/19 (BInf, BMinf, BTInf, BWInf)

Zeit: 75 Minuten

erlaubte Hilfsmittel: keine

Bitte tragen Sie Ihre Antworten und fertigen Lösungen ausschließlich an den gekennzeichneten Stellen in das Aufgabenblatt ein. Ist ihre Lösung wesentlich umfangreicher, so überprüfen Sie bitte nochmals Ihren Lösungsweg.

Nutzen Sie die Rückseiten der Klausur zur Entwicklung der Lösungen und übertragen die fertigen Lösungen in das Aufgabenblatt.

Sollten Unklarheiten oder Mehrdeutigkeiten bei der Aufgabenstellung auftreten, so notieren Sie bitte, wie Sie die Aufgabe interpretiert haben.

Viel Erfolg!

Diese Klausur besteht einschließlich dieses Deckblattes aus 6 Seiten.

---

**Aufgabe 1:**

Gegeben sei eine kontextfreie Grammatik  $G = (T, N, P, S)$

1. Definieren Sie  $nullable(X)$  für  $x \in (N \cup T)$ .

$nullable(X) = \dots\dots\dots$

2. Definieren Sie  $FIRST(w)$  mit  $w \in (N \cup T)^*$ .

$FIRST(w) = \dots\dots\dots$

3. Definieren Sie  $FOLLOW(X)$  mit  $x \in (N \cup T)$ .

$FOLLOW(X) = \dots\dots\dots$

$\dots\dots\dots$

4. In welche Zellen einer LL(1) Parsertabelle wird eine Regel  $X ::= w$  aus  $P$  auf Grund der  $FIRST$ -Mengen eingetragen?

$\dots\dots\dots$

5. In welche Zellen einer LL(1) Parsertabelle wird eine Regel  $X ::= w$  aus  $P$  auf Grund der  $FOLLOW$ -Mengen eingetragen?

$\dots\dots\dots$

$\dots\dots\dots$

6. Welche Eigenschaft muss eine LL-Parsertabelle besitzen, damit sie überhaupt in einem Parser genutzt werden kann?

$\dots\dots\dots$

7. Nennen Sie drei Grammatik-Eigenschaften, die die LL(1)-Eigenschaft einer kontextfreien Grammatik zerstören.

1)  $\dots\dots\dots$

2)  $\dots\dots\dots$

3)  $\dots\dots\dots$

## **Aufgabe 2:**

Konstruieren Sie einen deterministischen endlichen Automaten, der C-Kommentare akzeptiert. Es sind hier nur Kommentare der Form `/*...*/` gemeint.

**Das Zustands-Übergangs-Diagramm:**

### Aufgabe 3:

Skizzieren Sie, welcher Assembler-Code für einen von-Neumann-Rechner für die folgende Funktion sinnvollerweise erzeugt wird. Verwenden Sie für die Befehle die in der Vorlesung vorgestellte virtuelle Maschine und deren Instruktionssatz.

```
function mult(a : int; b : int) : int  
  if a > 0  
  then mult(a - 1, b) + b  
  else if a = 0  
  then 0  
  else 0 - mult(0 - a, b)
```

Verwenden Sie in den Instruktionen die Namen der Variablen als symbolische Adressen für die zugehörigen Speicheradressen (Lösung auf der nächsten Seite).

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

**Aufgabe 4:**

Definieren Sie die Ableitung  $\Delta$  einer regulären Menge  $r$  nach einem Zeichen  $a$ :

.....  
.....

Definieren Sie die Ableitung  $\Delta$  einer regulären Menge  $r$  nach einem Wort  $w$ :

.....  
.....  
.....

Berechnen Sie zu dem regulären Ausdruck  $r = (u|w?e)^*$  über dem Alphabet  $\{e, u, w\}$  die Ableitung  $\Delta_u(r)$ .

Der vereinfachte Ausdruck für die Ableitung:

.....  
.....

Berechnen Sie zu dem regulären Ausdruck  $r = (u|w?e)^*$  über dem Alphabet  $\{e, u, w\}$  die Ableitung  $\Delta_{uwe}(r)$ .

Der vereinfachte Ausdruck für die Ableitung:

.....  
.....