
Aufgaben zur Klausur **Compilerbau** im WS 2000/01 (II h769, WI h707, Master h100)

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.

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 die folgende kontextfreie Grammatik $G=(\mathbf{T}, \mathbf{N}, \mathbf{P}, \mathbf{S})$ mit

$$\mathbf{T} = \{ a, (,), [,], +, * \}$$

$$\mathbf{N} = \{ E \}$$

$$\mathbf{S} = E$$

und den Produktionen \mathbf{P} :

$$E ::= E + E$$

$$E ::= E * E$$

$$E ::= E [E]$$

$$E ::= (E)$$

$$E ::= a$$

Diese Grammatik ist mehrdeutig. Konstruieren Sie eine gleichwertige eindeutige kontextfreie Grammatik. Hierbei sollen folgende Prioritäten eingehalten werden: $*$ bindet stärker als $+$ und $[]$ bindet stärker als $*$. $+$ und $*$ sollen rechtsassoziative Operatoren sein.

Hinweis: Verwenden Sie die Nichtterminalmenge $\mathbf{N} = \{ E, \mathbf{T}, \mathbf{F}, \mathbf{V} \}$ und das Startsymbol E .

Die Produktionen:

- 1)
- 2)
- 3)
- 4)
- 5)
- 6)
- 7)
- 8)
- 9)
- 10)
- 11)
- 12)

Die resultierende Grammatik ist für die LL(1)-Analyse nicht geeignet. Transformieren Sie die Grammatik in eine gleichwertige, die die LL(1)-Eigenschaft besitzt.
Hinweis: Verwenden Sie die Nichtterminalmenge $\mathbf{N} = \{ E, E', T, T', F, F', V \}$ und das Startsymbol E .

Die Produktionen:

- 1)
- 2)
- 3)
- 4)
- 5)
- 6)
- 7)
- 8)
- 9)
- 10)
- 11)
- 12)
- 13)
- 14)

Aufgabe 2:

Transformieren sie den regulären Ausdruck $a * |ab+$ gemäß des Transformationsschemas aus der Vorlesung in einen nichtdeterministischen endlichen Automaten. Das zu Grunde liegende Alphabet sei dabei $I = \{a, b\}$. Hinweis: Der $*$ - und der $+$ -Operator binden stärker als der $|$ -Operator.

Der Automat als Zustandsübergangsdiagramm:

Aufgabe 3:

1. Warum sind reguläre Ausdrücke ungeeignet, die Syntax einer Programmiersprache vom Umfang von Pascal oder C zu definieren?

.....

.....

.....

2. Warum werden für die lexikalische Analyse reguläre Ausdrücke eingesetzt und nicht kontextfreie Grammatiken?

.....

.....

.....

3. Warum benötigt man zur Überprüfung, ob ein Programm compilierbar ist, neben der Syntaxanalyse noch die Phase der semantischen Analyse?

.....

.....

.....

4. Welche Fehlersituationen werden in der semantischen Analyse erkannt?

.....

.....

.....