

Compiler-Abau

Offene Fragen und Aufgaben

Test

Matrikelnummer: **8809469**

Kurs: TINF21B3

Abgabedatum 16.12.2022

Inhaltsverzeichnis

1	Fragen	2
2	Aufgaben	2
3	LL-Eigenschaften	3
4	LL-Automaten	4

1 Fragen

- Lösung des Shift/Reduce Konflikt. Was steht auf dem Keller? Was ist das nächste Zeichen?
- Wie funktioniert der Keller im Bezug auf die Elimination der linksrekursion? Script Seite 174
- Wie testet man auf LL-Bedingung?

Ich antworte ihr auf die Frage

2 Aufgaben

Auf Seite 7 im Script sind die Übungsaufgaben verzeichnet
Laut den Alt-Klausuren

- Strukturierung von einem Übersetzer
- Fragen zur Grammatik
- Chomsky-Hierarchie
- Lark+Ast oder Rex
- Top-Down-Parser/Rekursiver Abstiegs-Parser
- Abstrakter Syntaxbaum
 - Grammatik
 - Automat
 - Ableitung

Laut Vollmer

- Scanner
- Parser (ist ein LR-Automat)
- Baum
- rekursiver Abstiegsparser (ist ein LL-Automat)

Andere Aufgaben

- Quiz File

3 LL-Eigenschaften

Seite 166 im Script stehen die Eigenschaften

Wie andere Grammatik transformiert findet man im Script auf Seite 169 Eine Grammatik kann nicht die LL-Eigenschaften erfüllen wenn sie linksrekursion bzw. linksgleiche Produktionen enthält (Was sind Produktionen?) Allgemeine Elimination von linksrekursion auf Seite 171 im Script

$$\begin{aligned} A &::= A\alpha \\ &\implies \\ A &::= \beta A' \\ A' &::= \alpha A' | \epsilon \end{aligned}$$

Definition (Linksfaktorisierung). Problem $FIRST(..FOLLOW(..))$ nicht disjunkt:

$$\begin{aligned} A &::= \alpha\beta_1 | \alpha\beta_2 \\ &\implies \\ A &::= \alpha A' \\ A' &::= \beta_1 | \beta_2 \end{aligned}$$

$FF_1 = FIRST(TE'FOLLOW(E)) = i*$ das in den geschweiften Klammern ist das $First(T)$ wenn ϵ nicht in der First-Menge ist. Falls doch ist es das $First$ vom nächsten nicht Terminal. Falls alle ein ϵ in ihrer First_Menge haben ist es das $Follow(E')$.

LL-Bedingungen

Die Grammatik erfüllt die LL-Bedingungen wenn die gleichen Follows in den First-Follow-Mengen einen unterschiedlichen Inhalt haben.

$$\begin{aligned} FF_1 &= FIRST(TE'FOLLOW(E)) = i* \\ FF_2 &= FIRST(\epsilon FOLLOW(E)) = \# \end{aligned}$$

4 LL-Automaten

Seite 177 findet man die LL-Automaten

- Der LL-Automat erzeugt eine Linksableitung des Eingabewortes
 - Erfüllt G die LL-Bedingungen, dann kann ein deterministischer Automat konstruiert werden.
1. Transformieren Sie die Grammatik, so dass die Grammatik die LL(1) Bedingung erfüllt
 2. Erstellen Sie den nichtdeterministischen LL(1)-Automaten für diese Grammatik
 3. Erstellen Sie hieraus den deterministischen LL(1)-Automaten (nun ja er ist nicht ganz deterministisch, da die Produktionen eines Nichtterminals die LL(1) Bedingung nicht erfüllt, erstellen Sie den Automaten trotzdem!)
Markieren Sie die nichtdeterministischen Automatenregeln.
 4. Akzeptieren Sie mit diesem Automaten das "Programm $i + i$ "

Beispiel Aufgabe auf Seite 179 im Script

- Grammatik linksrekursion rausbekommen
- First-Follow-Menge berechnen
- LL1 Eigenschaften herausfinden
- Automat
- Automat mit First-Follow

Die Regel schreib man einfach umgekehrt zur Grammatik.

$$\begin{aligned} E' &::= +TE' \\ E'qt &\longrightarrow E'T+ \end{aligned}$$