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 Elimation der linksrekursion? Script Seite 174
- Wie testet man auf LL-Bedingung? siehe hier 3.1
- Wie geh ich mit Mehrdeutigkeit um im Shift Reduce Kontext?

  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

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

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

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

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

3.1 LL-Bedingungen

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

$$FF_1 = FIRST(TE'FOLLOW(E)) = i*$$
$$FF_2 = FIRST(\epsilon FOLLOW(E)) = \#$$

3.2 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. Transfomieren 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 "Programmï + i[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.

$$E' ::= +TE'$$
$$E'qt \longrightarrow E'T+$$

4 LR-Automaten

Beispiel 75 *(Ausdrucksgrammatik)*

```
Grammatik G = (N, T, P, E),
T = {+, *, (, ), i},
N = {E, T, F}
P = {1) E ::= T       2) E ::= E + T
     3) T ::= F       4) T ::= T * F
     5) F ::= i       6) F ::= ( E )}


LR-Automat A = (T, {q}, R, q, {q}, N ∪ T, ε)
R = {1)   Tq → Eq    2) E+Tq → Eq
     3)   Fq → Tq    4) T*Fq → Tq
     5)   iq → Fq    6) (E)q → Fq
     q+ → +q, q* → *q, q) → )q, q( → (q, qi → iq
     Eq# → q#
    }
```

Beispiel 76 *(Ausdrucksgrammatik, LR-Ableitung)*

Regel	*Keller*	*Eingabe*	*umgekehrte Rechtssableitung*
shift	q	1 + 2 * 3 #	
reduce 5	i q	+ 2 * 3 #	1 + 2 * 3
reduce 3	F q	+ 2 * 3 #	F + 2 * 3
reduce 1	T q	+ 2 * 3 #	T + 2 * 3
shift	E q	+ 2 * 3 #	
shift	E + q	2 * 3 #	
reduce 5	E + i q	* 3 #	E + 2 * 3
reduce 3	E + F q	* 3 #	E + F * 3
shift	E + T q	* 3 #	
shift	E + T * q	3 #	
reduce 5	E + T * i q	#	E + T * 3
reduce 4	E + T * F q	#	E + T * F
reduce 2	E + T q	#	E + T
reduce	E q	#	E
	q	#	

Mehrdeutigkeit wird unterbunden in dem bevorzugt gesschiftet wird. $\Longrightarrow$ Lösung Dangeling-Else Problem.

5 Struktur vom Compiler

1. Wozu kann ein Übersetzerbenutzt werden?

   Erzeugen von Maschinencode

   Programmanalyse und das Füllen einer Datenbank mit Informationen über das Programm

   Programmtransformation in eine andere Programmiersprache

2. Wie ist ein Übersetzer strukturiert? Beschreiben Sie **kurz** die Aufgaben und Ergebnisse der einzelen Phasen!

   1. Lexikalische Analyse, 2.syntaktische Analyse, 3. Transformation(nicht ausreichend)

3. Welche Rolle spielt die Grammatik in einer Sprache?

   a Eine Grammatik dient zur Spezifikation der Programmiersprachr für den Benutzer der Sprache, damit dieser weiß, wie ein korrektes Programm geschriebenwerden kann(ANleitung zum Generieren eines Satzes der Sprache).

   b Eine Grammatik dient zur Spezifikation des Übersetzters für diese Programmiersprache(Anleitung zur Konstruktion eines Akzeptors für diese Sprache).

6 Fragen zur Grammatik

1. Geben sie die Definition einer regulären Grammatik an.

   $G = (N, T, P, Z)$, Nichtterminale(z.B. E,T,D die Dinge mit denen man die Regeln macht), Terminale = Symbole(z.B. +,-,*),Produktion (Regel), Startsymbol

2. Geben sie die Definition des Begriffes der von einer Grammatik erzeugten Sprache an.

$$L = \{w \in T* | Z \Longrightarrow *w\}$$

   - Kein Plan was das heißen soll steht auch auf Seite 71 im Script

3. Geben Sie die Definition eines endlichen Automaten an.

4. Geben sie die Definition der von diesem Automaten akzeptierten Sprache an.(Welche Art von Sprachen wird vom Automat akzeptiert)

5. Was ist der Zusammenhang zwischen einem endlichen Automaten A, der die von G erzeugte Sprache akzeptiert.

   - Zu jeder regulären Grammatik G gibt es einen endlichen Automaten A, der die von G erzeugte Sprache akzeptiert(Gibt einen Beweis aber kein bock den hinzuschreiben oder zu lernen)

6. Was ist der Zusammenhang zwischen deterministischen und nichtdeterministischen endlichen Automaten?

   - Zu jedem nichtdeterministischen endlichen Automaten gibt es einen deterministischen endlichen Automaten, der die gleiche Sprache akzeptiert.

7. Es wird eine Sprache beschrieben an dieser sollen folgende Aufgaben durch geführt werden

   a Die rechts reguläre Grammatik soll für die Sprache angegeben werden.

   b Für die Grammatik soll der Endliche Automat angegeben werden. (eigentlich dann mit Shift und Reduce)

   c Stellen sie ihren Automaten graphisch dar. Kennzeichnen Sie Start- und Finalzustände.

   d Ist Ihr Automat deterministisch? Falls nein,kennzeichnen Sie nichtdeterministischen Übergänge.

8. Konstruieren Sie (mittels des aus der Vorlesung bekannten Verfahrens) den deterministischen endlichen Automaten, der die von Ihnen definierten C-Bezeichner akzeptiert. Zeichen Sie den resultierenden Automaten(Es ist die Teilmengen Konstruktion)