

Übungen zu „Grundlagen des Compilerbau“, Winter 2011/12

Nr. 8, Abgabe der Aufgaben: 10. Januar 2012 vor der Vorlesung



Frohe Weihnachten
 und alles Gute in 2012



Hinweis: Am Mittwoch den 21.12. und Donnerstag den 22.12. fallen die Vorlesung und das Tutorium aus.

Aufgaben

8.1 Eigenschaften von LR(0) Grammatiken

2 Punkte

Für eine LR(0)-Grammatik arbeitet der LR(0)-Analyseautomat bekanntlich ohne jeden Lookahead *deterministisch*.

Eine Sprache L heißt „präfixfrei“, falls gilt: $\forall w \in L, k < |w| : first_k(w) \notin L$.

Zeigen Sie mit Hilfe des LR(0)-Analyseautomaten:

Ist eine Sprache L in $\mathcal{L}(LR(0))$, so ist sie präfixfrei.

8.2 Konflikte in Happy

5 Punkte

Schreiben Sie zu den gegebenen Grammatiken Eingabedateien für den Parsergenerator HAPPY, so dass die erzeugten Parser bei korrekten Eingaben (Tokentyp **Char**) das Paar (Anzahl a, Anzahl b) zurückliefern (Typ **(Int, Int)**).

$$G_{\text{palindrom}} : S \rightarrow aXa \mid bXb \\ X \rightarrow S \mid \varepsilon$$

$$G_{\text{aUndB}} : S \rightarrow a \mid aS \mid aSb$$

Happy meldet für beide Grammatiken Konflikte (verwenden Sie die Option `-i`, um zusätzliche Informationen zu erhalten). Woran liegt das? Welches Verhalten zeigen die erzeugten Parser?

Geben Sie, falls möglich, jeweils eine Grammatik für die gleiche Sprache an, mit der HAPPY einen konfliktfreien Parser erzeugen kann.

8.3 Abstrakter Syntaxbaum mit Happy

5 Punkte

Gegeben sei die kontextfreie Grammatik G_S mit den Produktionen:

$$\begin{aligned} S &\rightarrow \text{id} = E \\ &\quad | \text{begin } S; S \text{ end} \\ &\quad | \text{if } B \text{ then } S \\ E &\rightarrow E + E \mid \text{id} \mid \text{num} \\ B &\rightarrow E < E \end{aligned}$$

- (a) Begründen Sie, warum $G_S \notin LALR(1)$. / 1
- (b) Mit den Typen s , E und B lässt sich ein konkreter Syntaxbaum für G_S repräsentieren. Definieren Sie Haskell-Datentypen s' , E' und B' , mit denen ein abstrakter Syntaxbaum der Grammatik repräsentiert werden kann. / 1,5
- (c) Definieren Sie eine *konfliktfreie* Happy-Datei¹ für G_S , so dass durch Happy ein abstrakter Syntaxbaum vom Typ S' erzeugt wird. Testen Sie den Parser für:
„if id < num then begin id = num + id + num; id = num end“ / 2,5

```
data Token = Id | Num | If | Then | Begin
          | End | Sem | Ass | Plus | Less deriving Show
data S = Assign Token Token E | Seq Token S Token S Token
      | Cond Token B Token deriving Show
data E = Add E Token E | Var Token | Val Token deriving Show
data B = Less E Token E deriving Show
```

¹Ein Template der Happy Datei mit Scanner-Funktion finden Sie auf der VL-Seite