

12. Januar 2004

Übungen zu „Grundlagen des Compilerbaus“, WS 2003/04

Nr. 9 , Abgabe und Besprechung: 19. Januar 2004 in der Übung

Mündliche Aufgaben

9.1 Grammatik für hierarchischen Namensraum

Die folgende Grammatik beschreibt die Syntax von Zugriffen auf einen hierarchischen Namensraum, etwa auf Felder und (parameterlose) Methoden in Java.

$G : S \rightarrow A$

$A \rightarrow B \mid B()$

$B \rightarrow \text{identifizier} \mid B.A$

Beispiele:

`mypackage.MyClass.Subclass.staticMethod()`

`MyObject.selector().aField`

Zeigen Sie, dass G nicht in $SLR(1)$ ist, indem Sie bestehende Konflikte untersuchen.

9.2 Beweisen oder widerlegen Sie:

Eine Grammatik mit ε -Produktionen kann nicht in $LR(0)$ / in $SLR(1)$ sein.

Schriftliche Aufgaben

9.3 LR(0)-Test in Haskell

8 Punkte

- (a) Implementieren Sie in Haskell eine Funktion `gram2nfa :: CFG -> NFA Item`, die aus einer gegebenen kontextfreien Grammatik einen NFA für die $LR(0)$ -Analyse erstellt. Die Zustände `Item` dieses NFA sind die $LR(0)$ -Auskünfte, welche man (beispielsweise) beschreiben kann, indem man außer der rechten Regelseite den Teil hinter dem Punkt dazu nimmt.

```
type Item = (Nonterminal,Satzform,Satzform)
{- z.B: Regel ('S',"Ab"),
   Items: ('S',"Ab","Ab"), ('S',"Ab","b"), ('S',"Ab","") -}
```

Verwenden sie die Repräsentation `CFG` für Grammatiken aus Aufgabe 5.5 und den `NFA`-Typ aus Aufgabe 2.4.

- (b) Implementieren Sie mit Hilfe von a) einen $LR(0)$ -Konflikttest, welcher die widersprüchlichen $LR(0)$ -Informationen ausgibt.

Sie können ggf. die Existenz einer Funktion $\bar{\delta}$ für NFAs voraussetzen:

```
deltaquer :: DeltaNFA state -> [state] -> String -> [state]
```

9.4 SLR-Tabelle

4 Punkte

Konstruieren Sie die $SLR(1)$ -Analysetabelle zu der Grammatik G mit den Regeln:

$S' \rightarrow S$

$S \rightarrow A \mid b$

$A \rightarrow Aa \mid Sb$