Lehrstuhl für Programmiersprachen und Übersetzerkonstruktion
CAU Kiel, Informatik und Prakt. Mathematik
Prof. Dr. M. Hanus, Dr. F. Huch

24118 Kiel
Christan-Albrechts-Platz 4
Tel.: 0431 / 880-7277

6. Übung „Übersetzerbau“

Abgabe am 30. Mai in der Vorlesung

Aufgabe 21 9 Punkte

In dieser Aufgabe sollen Sie einen shift-reduce-Parser implementieren. Gehen Sie dazu in folgenden Schritten vor:

a) Formalisieren Sie zunächst, wie und wann der shift-reduce-Parser einen shift- bzw. einen reduce-Schritt durchführt.
Repräsentieren Sie den Keller als ein Wort über Zuständen ($\mathbb{N}^*$). In der Vorlesung wurden auch noch Grammatiksymbole auf den Keller geschrieben. Diese sind für das Arbeiten des Parsers aber überflüssig und können wegfallen.
Um völlig unabhängig von der Grammatik zu sein erweitern Sie die reduce-Einträge um die Länge der rechten Seite der anzuwendenden Regel und das Nichtterminalsymbol, zu welchem reduziert wird.
Formalisieren Sie auch, in welcher Konfiguration der Parser erfolgreich terminiert.

b) Erweitern Sie den Parser um ein Ausgabeband, auf welchem die Nummer der Verwendeten Regel beim reduce-Schritt ausgegeben wird. Liefert Ihr Parser die Rechtsableitung als Ausgabe?

c) Definieren Sie eine polymorphe Datenstruktur `ParsingTabelle a b` zur Darstellung der Parsingtabelle. Die Parsingtabelle soll polymorph bezüglich der gewählten Alphabete sein. `a` repräsentiert die Terminal- und `b` die Nichtterminalsymbole. Da die Parsingtabelle in der Regel nur eine partielle Funktion repräsentiert, also Lücken aufweist, sollten Sie den Haskell-Datentyp `Maybe` verwenden.

d) Definieren Sie eine Funktion `parse :: ParsingTabelle a b -> [a] -> [Int]`. Als Eingabe erhält dieser Parser eine Parsingtabelle und eine Tokenfolge. Als Ausgabe soll der Parser im Erfolgsfall die Ausgabe des Automaten liefern.

Aufgabe 22 3 Punkte

Zeigen Sie, dass die Klasse REG aller regulären Sprachen schief zur Klasse $\mathfrak{L}(LR(0))$ aller von $LR(0)$–Grammatiken erzeugten Sprachen liegt. Da Sie etwas für Sprachen zeigen sollen, sollten Sie möglichst einfache Sprachen betrachten. Überlegen Sie zunächst, mit welchen Produktionen es bei LR(0) meistens Probleme gibt.

Aufgabe 23 3 Punkte

In der Vorlesung wurde eine Grammtik vorgestellt, welche nicht SLR(1) aber LR(1) und auch LALR(1) ist. Zeigen Sie, dass diese Grammatik die SLR(1)-Eigenschaft nicht besitzt.

Aufgabe 24 5 Punkte

a) Überlegen Sie sich ein Verfahren, wie Sie direkt die $LALR(1)$–Mengen berechnen können, ohne zuvor die $LR(1)$-Mengen zu berechnen (nicht formal, Idee reicht aus).

b) Wenden Sie Ihr Verfahren auf folgende Grammatik an:

$$\begin{array}{rlll} G: & S & \longrightarrow & aaSb \mid A \mid \varepsilon \\ & A & \longrightarrow & Aa \mid c \end{array}$$

Liegt G in $LR(0)$, $SLR(1)$ bzw. $LALR(1)$?
Konstruieren Sie ggf. die $LALR(1)$–Analysetabelle.