
Compilerbau

Blatt 6

Abgabe: 28.11.2001

1. [4 Punkte]
 - (a) Wie löst der Parser aus der Struktur Lr die Mehrdeutigkeiten, die beim Parsen mit rekursivem Aufstieg auftreten können?
 - (b) Erweitern Sie den Parser aus der Struktur Lr so, daß er eine Warnung ausgibt, wenn er sich in einem Zustand befindet, in dem eine Mehrdeutigkeit aufgetreten ist.
2. [6 Punkte] Implementieren Sie Attributauswertung für den Parser mit rekursivem Aufstieg.
3. [10 Punkte] Implementieren Sie im Modul Camlparse einen Parser für Mini-Caml, der abstrakte Syntax aus dem Modul Camlsyn erzeugt.

Die Definition des Attributtyps für den Parser ist dabei problematisch: Warum ist es nicht möglich, den Typ Camlsyn.program aus der abstrakten Syntax direkt als Attributtyp zu verwenden? Warum ist es nicht möglich, den Attributtyp des Scanners zu verwenden? Verwenden Sie folgenden Summentyp als Attributtyp:

```
type prog_attr =  
  Prog of Camlsyn.program  
  | Defs of Camlsyn.definition list  
  | Bind of Camlsyn.binding  
  | Def of Camlsyn.definition  
  | Expr of Camlsyn.syntax
```

Benutzen Sie (vorzugsweise) Ihren eigenen Implementierungen oder den Scanner in der Struktur Camllex und die Grammatik aus Camlparse. Sollten Sie Aufgabe 2 nicht gelöst haben, wenden Sie sich an Martin Gasbichler.

4. [5 Punkte] Geben Sie einen "Un-Parser" für Mini-Caml an, der aus der abstrakten Syntax wieder ein konkretes Programm produziert. Testen Sie den Un-Parser an einigen repräsentativen Beispielen.
5. [5 Punkte] Geben Sie die formale Spezifikation eines Parsers mit rekursivem Abstieg an, der während des Parsens Attributauswertung für L-attributierte Grammatiken vornimmt.

Anleitung: Schreiben Sie zunächst eine Akzeptorspezifikation auf der Basis von Funktionen $[p]$, wobei p eine Position anstatt einer Folge von Grammatiksymbolen ist. Erweitern Sie diese um Attributauswertung.