

Programmierung und Modellierung

Aufgabe 10-1

Lexer für SML-Typen

Schreiben Sie einen Lexer für die auf Folie 22 des Foliensatzes 11 definierte Grammatik für Typausdrücke. Die Token sind dabei wie folgt definiert:

```
datatype tok = BOOL | INT | ARROW | LPAR | RPAR;
```

Ihr Lexer soll Leerzeichen geeignet behandeln.

- Ergänzen Sie Ihre Lösung in der Datei `10-1.sml`, die Sie von der Vorlesungswebseite herunterladen können.
- Ergänzen Sie die Implementierung Ihres Lexers sowie des Syntaxcheckers in `10-1.sml` so, dass auch Typvariablen akzeptiert werden (`'a`, `'b` etc.). Beachten Sie dabei, dass `'8` etc. ausgeschlossen werden soll.

Aufgabe 10-2

Parser für Bäume

In dieser Aufgabe soll ein Parser für Bäume entwickelt werden. Bäume speichern dabei `int`-Werte in den Knoten und den Blättern, Knoten speichern zudem linke und rechte Teilbäume. Der Datentyp für diese Bäume ist

```
datatype tree = Node of tree * int * tree | Leaf of int;
```

Als Eingabe werden Bäume in geklammerter Infix-Notation gegeben, d.h. Worte der folgenden Grammatik:

```
tree ::= (tree,int,tree) | int
```

So sind z.B. `1`, `(1,2,3)` und `(1,2,(3,4,(5,6,7)))` syntaktisch korrekte Bäume.

- Definieren Sie geeignete Token und einen Lexer.
- Schreiben Sie einen Parser, der einen syntaktisch korrekten Baum in die oben angegebene Datenstruktur parst, und für syntaktisch inkorrekte Bäume einen Fehler ausgibt. Beachten Sie dabei, dass auch dann ein Fehler ausgegeben wird, wenn ein eindeutiges Parsen möglich wäre (z.B. für die Eingabe `(1,2,3)4`).

Abgabe: Montag, den 13.7.2009, 12 Uhr, per UniWorx.