

Aufgabe A4.2

1. Beispielprogramme und Fehlerkategorien

Um die Sprache zu testen, wurden zuerst verschiedene Programme geschrieben. Dazu gehören einfache Sachen wie Zahlen, Strings oder kleine Operator-Ausdrücke, aber auch größere Beispiele mit `defn`, `let` und `if`.

Ein etwas komplexeres Beispiel war diese Funktion, die je nach Wert von `n` eine andere Ausgabe macht:

```
(defn hello (n)
  (if (< n 10)
      (print (str "small: " n))
      (print (str "big: " n))))  
  
(hello 5)
```

Dabei wurden auch gezielt Fehler ausprobiert, zum Beispiel:

- fehlende oder falsche Klammern,
- kaputte Strings,
- unvollständige `def` oder `defn` Formen,
- falsche Anzahl an Argumenten.

So kann man gut prüfen, ob die Grammatik ungültige Programme richtig ablehnt.

2. Grammatik

Auf Basis der Sprachbeschreibung wurde eine passende Grammatik erstellt. Sie beschreibt Atome, S-Expressions, Operatoren und die Sonderformen `def`, `defn`, `let`, `if` und `do`. Die Grammatik wurde in der Datei `Aufgabe2.g4` angelegt und dient später als Grundlage für Lexer und Parser.

Ein Auszug aus dem Parserteil der Grammatik:

```
program  : expr* EOF ;
expr     : atom | listExpr ;
atom    : INT | STRING | BOOL | IDENT ;
listExpr : LPAREN form RPAREN ;
form    : ifExpr | doExpr | defExpr | defnExpr | letExpr | appExpr ;
```

Damit lassen sich sowohl einfache als auch verschachtelte Ausdrücke der Sprache beschreiben.

Parsebaum-Beispiel

Das oben gezeigte Beispiel wurde im ANTLR-Preview getestet. Der entstandene Parsebaum zeigt, dass die Grammatik den Aufbau des Programms richtig erkennt und die Struktur korrekt einordnet.

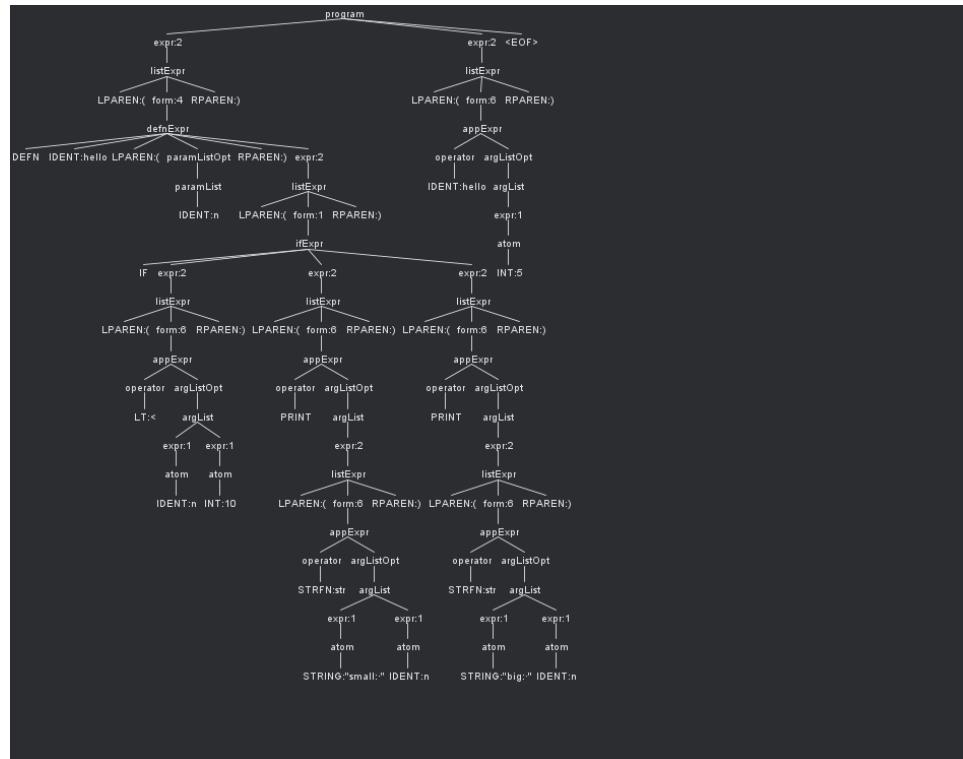


Abbildung 1: Parsebaum für das Beispiel oben.