

Prolog Parser

Franziska Corradi {corrfl@bfh.ch}
Simon Krenger {krens1@bfh.ch}

January 17, 2014

1 Einleitung

Das vorliegende Projekt verwendet einen deterministischen endlichen Automaten, um die Tokens des Prolog-Programms zu analysieren.

2 Übergangstabelle

In der Datei “doc/Transitionstabelle.xlsx” findet sich die vom Programm verwendete Transitionstabelle.

3 Programmbeschreibung

Das Kommandozeilenprogramm (JAR) soll mittels des folgenden Befehl gestartet werden:

```
$ java -jar PrologParser.jar inputfile.pl
```

Das Programm selbst verwendet die folgenden Klassen:

- PrologParser: Einstiegspunkt des Programms. Diese Klasse enthält die `main()` Methode und erstellt die Instanzen der anderen Klassen.
- PrologPreprocessor: Diese Klasse wird dazu verwendet, den Quellcode einzulesen und für den Scanner vorzubereiten. Im Preprocessor werden alle Kommentare entfernt, alle Spezialzeichen entfernt (Methode `preprocess()`).
- PrologScanner: Diese Klasse stellt den Hauptteil des Programms dar. Hier wird der Automat erstellt und die Transitionen definiert. Die `tokenize(String input)` Methode liefert dann die Tokens zurück.
- LexerState: Klasse für den Automaten, repräsentiert einen Zustand des Automaten.
- LexerStateTranslation: Klasse für den Automaten, repräsentiert eine Zustandsänderung des Automaten.

4 Deterministischer endlicher Automat

Folgender Graph zeigt den Automaten für den vereinfachten PrologParser.

