

Theoretische Informatik II Übungen

Aufgabe 21. Implementieren Sie Input und Output Funktionen für den Scheme Interpreter.

In der vorangehenden Übung wurden Funktionen implementiert mit denen sich Scheme Strukturen (Listen, Listen von Listen, Symbole, Zahlen, etc.) im simulierten Speicher anlegen lassen. Nun werden Funktionen benötigt, die diese Strukturen einlesen und ausgeben.

Wir umgehen die Konstruktion eines Scanners und Parsers, die man ansonsten mit Werkzeugen wie `lex` und `yacc` (diese gibt es auch für Scheme) erzeugen würde und benutzen statt dessen die Funktionen `read` und `display`, die in Scheme schon vorhanden sind.

- Input.

Die Scheme Funktion `read` liest vom Input bereits einen Scheme Ausdruck ein. Schreiben Sie eine Konversionsfunktion `expr->i-expr`, die einen Schemeausdruck (rekursiv) in eine i-scheme Struktur im simulierten Speicher verwandelt. Benutzen sie diese Funktion um mittels `read` die Funktion `i-read` zu schreiben.

- Output.

Die Scheme Funktion `display` schreibt bereits einen Scheme Ausdruck auf die Ausgabe. Schreiben Sie eine Konversionsfunktion `i-expr->expr`, die eine i-scheme Struktur im simulierten Speicher (rekursiv) in einen Scheme Ausdruck verwandelt. Benutzen sie diese Funktion um mittels `display` die Funktion `i-display` zu schreiben.