Automatentheorie und Formale Sprachen
Prof. Dr. Michael Zohner
Sommersemester 2021

Hochschule Fulda
University of Applied Sciences

Testat 2: Parserentwicklung mit JavaCC v1.0*

Abgabe: 31. August 2021, 23:59 Uhr

Aufgabenstellung Die esoterische Programmiersprache Brainfuck zeichnet sich durch ein einfaches Sprachkonzept mit wenigen Befehlen aber auch durch kaum verständliche Programme aus. Sie funktioniert ähnlich zu einer Turingmaschine:

1. Es werden Zeichen von der Eingabe gelesen
2. Zeichen werden durch einen Interpreter mit einer einfachen Arithmetischen-Logischen Einheit (In-/Dekrementierung und Schleifen) verarbeitet
3. Verarbeitete Zeichen werden auf der Ausgabe ausgegeben.

Brainfuck speichert den internen Zustand als Zahlenwert in Speicherzellen und hält sich einen Zeiger auf die aktuelle Speicherzelle. Insgesamt gibt es 8 Befehle in Brainfuck, die in Tabelle 1 dargestellt sind:

Befehl	**Semantik**
>	Inkrementiere den Zeiger auf die nächsthöhere Speicherzelle
<	Dekrementiere den Zeiger auf die nächstniedrigere Speicherzelle
+	Inkrementiere den Wert der aktuellen Speicherzelle um 1
-	Dekrementiere den Wert der aktuellen Speicherzelle um 1
.	Gib den Wert der aktuellen Speicherzelle im ASCII Format auf der Standardausgabe aus
,	Lese ein Zeichen von der Standardeingabe und speichere den ASCII Zahlenwert in der aktuellen Speicherzelle
[	Springe vor hinter den passenden "]"-Befehl, wenn der Wert der aktuellen Speicherzelle gleich 0 ist
]	Springe zurück hinter den passenden "["-Befehl, wenn der Wert der aktuellen Speicherzelle nicht 0 ist

Table 1: Befehlssatz der Programmiersprache Brainfuck (von `https://de.wikipedia.org/wiki/Brainfuck`).

Aufgabe Nutzen Sie den Java Compiler Compiler (JavaCC: `https://javacc.github.io/javacc/`) um einen Parser und eingeschränkten Interpreter für Brainfuck Programme zu implementieren. JavaCC übersetzt eine Grammatikspezifikation, die in einem proprietären Format spezifiziert wurde (*.jj Dateien), in ein Java Programm, das Eingabedateien scannt und parst. Zusätzlich erlaubt es JavaCC während des Parsens Befehle eingeschränkt zu interpretieren, was das gleichzeitige Parsen und Auswerten ermöglicht.

1. Der *Parser* soll Brainfuck Programme als Eingabeparameter einlesen und ihre Syntax verifizieren. Für syntaktisch fehlerhafte Programme soll eine Fehlermeldung ausgegeben werden. Beispiele für syntaktische Fehler sind unerlaubte Symbole oder unausgeglichene Schleifenbefehle ("[" und "]"). Zusätzlich zu den Brainfuckbefehlen sind in er Eingabedatei alle Whitespace Zeichen erlaubt (Leerzeichen, "\n", "\r", "\t", "\\").
2. Der *Interpreter* soll Brainfuck Programme auswerten und alle Befehle bis auf die Schleifenbefehle "[" und "]" interpretieren. Die Schleifenbefehle sollen ignoriert werden. Die Ein- und Ausgabe von Zeichen mittels der "." und "," Befehle soll auf der Kommandozeile erfolgen. Der Interpreter soll den Zustand in einem Ringpuffer von 256 Speicherzellen speichern, die Werte im Bereich von 0 - 255 enthalten dürfen.

Abgabe Laden Sie die Datei mit der Grammatikdefinition des JavaCC Parsers (*.jj Datei) unter dem Namen *Vorname_Nachname.jj* in Moodle hoch: `https://elearning.hs-fulda.de/ai/mod/assign/view.php?id=50827`. Falls Ihre Lösung weitere Dateien benötigt, laden Sie diese bitte zusätzlich hoch und beschrieben Sie den Kompilierprozess in einer beiliegenden Datei "Readme.txt". Tools die nicht ohne Änderung im Quellcode kompiliert werden können werden als nicht abgegebener Versuch gezählt. Die Abgabe schließt automatisch am 31 August 2021 um 23:59 Uhr. Spätere Einreichungen werden nicht berücksichtigt.

Bestehenskriterien Die Bewertung des Testates erfolgt anhand Ihres eingereichten Quellcodes unter der Eingabe von Brainfuck Beispielprogrammen. Ein solches Beispielprogramm ist hier zu finden: `https://elearning.hs-fulda.de/ai/pluginfile.php/85555/mod_assign/introattachment/0/ex_brainfuck_prog.bf?forcedownload=1`. Das Programm liest 5 einzelne Zeichen ein, erhöht bzw. verringert den ASCII Wert um 13 und gibt die resultierenden Werte aus. Zum Beispiel liefert das Programm bei Eingabe "F", "u", "l", "d" und "a" die Ausgabe "ShyWn".
Das Testat gilt als bestanden, wenn die Beispielprogramme korrekt ausgewertet und Programme mit fehlerhafter Syntax erkannt werden.

***Versionshistorie** 1.0: Initiale Veröffentlichung