Zürcher Hochschule
für Angewandte Wissenschaften

Theoretische Informatik
D. Flumini, R. Büchi, E. Bazzi, O. Stern

Übungsblatt 5b

Kellerautomat

Abgabe: 12. April

Abgabe als PDF-Datei auf Moodle mit dem Dateinamen: *thin-gruppe_#-uebungsblatt_#.pdf* . In der Abgabe sollten *alle* Team-Namen und ZHAW-Kürzel enthalten sein.

Umgekehrte polnische Notation, inklusive Präsentation in Gruppen

Damit zusammengesetzte Ausdrücke auch ohne Klammern geschrieben werden können, hat der polnische Mathematiker Jan Lukasiewicz eine Notation entworfen, welche die Operatoren neben den Zahlen und Variablen (und nicht dazwischen) aufführt. Die Notation schreibt die Operatoren nach den Zahlen und wird "Umgekehrte polnische Notation" genannt.

Anstelle von '$3+4$' schreibt man hier '$34+$'. Damit können auch zusammengesetzte Ausdrücke ohne Klammern geschrieben werden. Anstelle von '$(3+4)*(6-2)$' schreibt man nun '$34+62-*$'. Es wird also zuerst $3+4$ berechnet, danach $6-2$ und am Schluss werden die beiden Faktoren miteinander multipliziert.

Man darf sich vorstellen, dass die Zahlen der Reihe nach in den Keller, bzw. auf den Stack geschrieben werden. Kommt nun ein Operator, so werden die beiden obersten Zahlen vom Stack genommen, miteinander verrechnet und dann das Resultat wieder auf dem Stack gelegt.

Aufgabe 1.
Ein Kellerautomat ist in der Lage, einen Ausdruck in umgekehrter polnischer Notation zu erkennen (nicht aber, um ihn zu berechnen). Entwerfen Sie einen solchen Kellerautomaten. Dieser muss alle Wörter akzeptieren, welche einstellige Zahlen (als Symbol D für Digits) und die vier Grundrechenarten (als Symbol O für Operatoren) enthalten und zur Sprache der umgekehrten polnischen Notation (UPN) gehören. Alle anderen Wörter muss er verwerfen.

Aufgabe 2.
Setzen Sie den Kellerautomaten nun mit einer Programmierumgebung Ihrer Wahl um. Erweitern Sie diesen so, dass beliebige Ausdrücke in umgekehrt polnischer Notation berechnet werden.

Zeigen Sie schriftlich, bzw. per Programmausdruck und Screenshot der Ausgabe, folgende Punkte:

- Entwurf des Kellerautomaten (mit Kommentare im Code), Berechnung des Testworts $DDODDOO$ **(10P)**

- Implementierung der UPN (mit Kommentare im Code) für mindestens einstellige Zahlen und mindestens zwei Operatoren in einer Programmiersprache Ihrer Wahl **(10P)**
- Test a: $34 + 62 + *$ (akzeptierend und Resultat) **(5P)**
- Test b: $34 + *$ (**verwerfend**) **(5P)**
- Test c: ein eigenes gewähltes Wort (akzeptierend oder verwerfend) **(5P)**

Der Stack sollte als **Array** implementiert werden (und keine Bibliotheksklasse)
In der Musterlösung ist der Stack mit ['$', 0, 0, 0, 0, 0, 0, 0, 0, 0] initialisiert

Abgabe im *PDF*-Format mit den Screenshots inklusive Source Code

35 Punkte