Zürcher Hochschule
für Angewandte Wissenschaften

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

Übung 2

Kellerautomat Papier- und Programmieraufgabe

Präsentation: nach Absprache mit Dozent

Umgekehrte polnische Notation

Damit zusammengesetzte Ausdrücke auch ohne Klammern geschrieben werden können, hat der polnische Mathematiker Jan Lukasiewicz eine Notation entworfen, welche die Operatoren nach den Zahlen und Variablen (und nicht dazwischen) aufführt. Die Notation wird *Umgekehrte polnische Notation*[1] genannt.

Anstelle von '$3 + 4$' schreibt man hier '$3\ 4\ +$'. Damit können auch zusammengesetzte Ausdrücke ohne Klammern geschrieben und eindeutig interpretiert werden. Ein weiteres Beispiel: Für '$(3 + 4) * (6 - 2)$' schreibt man '$3\ 4\ +\ 6\ 2\ -\ *$'. Es wird also zuerst '$3 + 4$' berechnet, danach '$6 - 2$' und am Schluss werden die beiden Faktoren miteinander multipliziert. Das Resultat der Berechnung beträgt 28.

Man darf sich vorstellen, dass die Zahlen der Reihe nach in den Keller, bzw. auf den Stack geschrieben werden. Wird ein Operator eingelesen (bzw. kommt als Eingabe), so werden die beiden obersten Zahlen vom Stack genommen, miteinander verrechnet und das Resultat wieder auf dem Stack gelegt.

[1] Umgekehrte polnische Notation – Wikipedia: https://de.wikipedia.org/wiki/Umgekehrte_polnische_Notation

Aufgabe 1. ***Papieraufgabe.***
Ein deterministischer Kellerautomat ist in der Lage, einen Ausdruck in umgekehrter polnischer Notation (UPN) zu **akzeptieren** bzw. **nicht zu akzeptieren**. Entwerfen und zeichnen Sie einen deterministischen Kellerautomaten, der die Sprache der Wörter akzeptiert, die zur vereinfachten UPN gehören.

Vereinfachte UPN bedeutet, dass der Ausdruck nicht berechnet wird. Er muss alle Wörter akzeptieren, welche einstellige Zahlen (als Symbol Z für Zahlen) 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.

Hinweis: Verifizieren Sie, dass Ihr entworfener deterministischer Kellerautomat, tatsächlich auch deterministisch ist.

- Für diese Aufgabe kann somit 3 4 + als ZZO geschrieben werden.
- Beispiele für Wörter, welche er akzeptiert: ZZO, $ZZOZZOO$ oder $ZZZOO$.
- Beispiele für Wörter, welche er **nicht** akzeptiert: $ZZOO$, $ZZOZZO$ oder ZOZ.
- Führen Sie mit dem deterministischen Kellerautomaten von Hand die Berechnung für folgende Wörter durch.
  (a) ZZO
  (b) $ZZOZZOO$
  (c) $ZZOO$

8 Punkte

Aufgabe 2. ***Programmieraufgabe.***
Implementieren Sie einen deterministischen Kellerautomaten, welcher eine UPN berechnet, mit einer Programmierumgebung Ihrer Wahl.

Hinweis: Diese Aufgabe kann unabhängig von Aufgabe 1 gelöst werden.

Folgende Implementationen werden erwartet:

- Einen deterministischen Kellerautomaten, für **einstellige Zahlen** und die Operatoren + und $*$. Die Zahlen im Keller, die Zwischenergebnisse und das Endergebnis können mehrstellig sein.
- Die Berechnung und Ausgabe des Resultats, sofern das Eingabewort akzeptiert wird. Für die Berechnung der zwei Grundrechenarten dürfen die üblichen Operatoren der Programmiersprache verwendet werden.
- Der Keller/Stack. Dieser darf nicht mit einer Bibliotheksklasse (wie java.util.Stack) implementiert werden.
- Ausgabe des ganzen Programms mit zwei Modi,
  - ein **Step-Modus**: Ausgabe von dem Inhalt des Kellers, danach etwa ≈ 1 Sekunde warten. Dies für jeden einzelnen Berechnungsschritt. Abschliessend das Resultat.
  - ein **Laufmodus**: einmalige Ausgabe des Resultats nach Abschluss der ganzen Berechnung

10 Punkte

Präsentation
In der Präsentation werden folgende Punkte erwartet:

Auf. 1:
a) Erklärung und grafische Darstellung des deterministischen Kellerautomaten
b) Berechnungsschritte der drei Wörter von Hand

Auf. 2:
c) Präsentation des Codes
d) Präsentation der Implementierung des Kellers/Stacks
e) Live Demo im Step- oder Laufmodus: Berechnung der Test-Wörter, eigene Wörter oder die vom Dozenten vorgegebenen Wörter

☆ *Optional:*
f) Animierte Ausgabe des Kellers/Stack, Grafisches Interface, usw. *Bitte nur angehen, wenn Sie wirklich Zeit dazu haben.*

Zur Überprüfung können Sie folgende Test-Wörter verwenden:

Ausdruck (Input)	**Finale Ausgabe (Output)**
3 4 + 6 2 + 8 9 + 4 3 + * * *	Akzeptierend und das korrekte Resultat
3 1 + 7 8 + 9 8 7 + 1 2 1 4 + + 7 + + + + + + +	Akzeptierend und das korrekte Resultat
3 4 + *	Verwerfend
8 + 9 + 7 * 2 *	Verwerfend