

Theoretische Informatik

D. Flumini, R. Büchi,

E. Bazzi, O. Stern

Übung 2

Kellerautomat Papier- und Programmieraufgabe

Abgabe als PDF-Datei auf Moodle. In der Abgabe sollten *alle* Team-Namen und ZHAW-Kürzel enthalten sein.

Team: **Andrin Fassbind (Fassband), Leonardo Brandenberger (Brandleo), Roman Schenk (Schrom01)**

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.

¹Umgekehrte 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, wenn der Ausdruck kein UPN ist.

Achtung: Der deterministische Kellerautomat kann den Ausdruck nicht berechnen.

- **(5P)** Entwerfen und zeichnen Sie einen deterministischen Kellerautomaten, der die Sprache der Wörter akzeptiert, die zur vereinfachten umgekehrten polnischen Notation gehören. Vereinfacht bedeutet, dass er alle Wörter akzeptieren muss, 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.

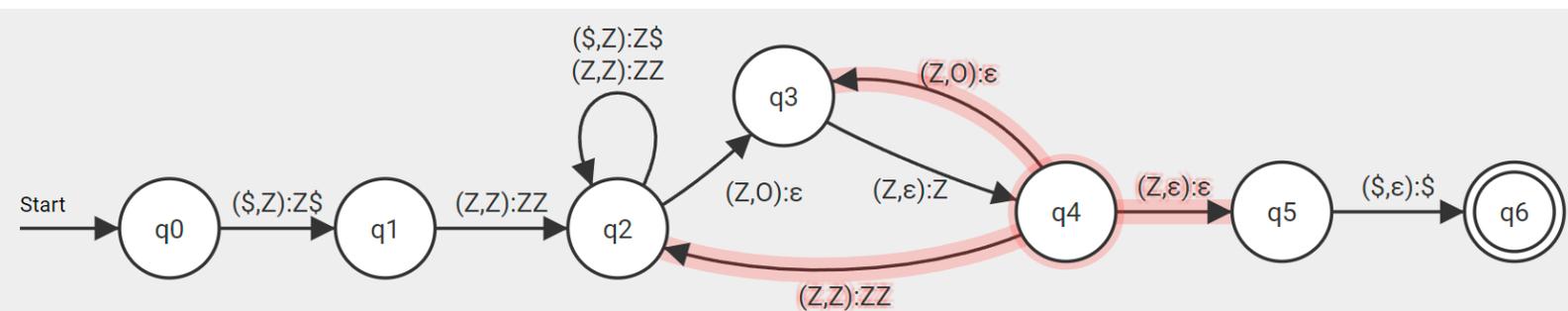
- Beispiele für Wörter, welche er akzeptiert: ZZO , $ZZOZZOO$ oder $ZZZOO$.
- Beispiele für Wörter, welche er **nicht** akzeptiert: $ZZOO$, $ZZOZZO$ oder ZOZ .

- **(3P)** Führen Sie mit dem deterministischen Kellerautomaten von Hand die Berechnung für folgende Wörter durch.

– ZZO $(q_0, ZZO, \$) \vdash (q_1, ZO, Z\$) \vdash (q_2, O, ZZ\$) \vdash (q_3, \varepsilon, Z\$) \vdash (q_4, \varepsilon, Z\$) \vdash (q_5, \varepsilon, \$) \vdash (q_6, \varepsilon, \$)$

– $ZZOZZOO$ $(q_0, ZZOZZOO, \$) \vdash (q_1, ZOZZOO, Z\$) \vdash (q_2, OZZOO, ZZ\$) \vdash (q_3, ZZOO, Z\$) \vdash (q_4, ZZOO, Z\$) \vdash (q_2, ZOO, ZZ\$) \vdash (q_2, OO, ZZZ\$) \vdash (q_3, O, ZZ\$) \vdash (q_4, O, ZZ\$) \vdash (q_3, \varepsilon, Z\$) \vdash (q_4, \varepsilon, Z\$) \vdash (q_5, \varepsilon, \$) \vdash (q_6, \varepsilon, \$)$

– $ZZOO$ $(q_0, ZZOO, \$) \vdash (q_1, ZOO, Z\$) \vdash (q_2, OO, ZZ\$) \vdash (q_3, O, Z\$) \vdash (q_4, O, Z\$) \vdash (q_3, \varepsilon, \$)$ wird verworfen.



Aufgabe 2. Programmieraufgabe.

Implementieren Sie den deterministischen Kellerautomaten von Aufgabe 1 nun mit einer Programmierumgebung Ihrer Wahl um. Erweitern Sie diesen so, dass Ausdrücke in vereinfachter umgekehrt polnischer Notation für die Grundrechenarten $+$ und $*$ berechnet werden können.

Zeigen Sie folgende Punkte:

- **(5P)** Implementierung des deterministischen Kellerautomaten aus Aufgabe 1, für einstellige Zahlen und die Operatoren $+$ und $*$.
- **(2P)** Zusätzliche Implementierung der Berechnung des Resultats der UPN, sofern das Eingabewort akzeptiert wird. Hinweis: Für die Berechnung der zwei Grundrechenarten ($+$ und $*$) dürfen die üblichen Operatoren der Programmiersprache verwendet werden.
- **(1P) Test a:** $3\ 4\ +\ 6\ 2\ +\ 8\ 9\ +\ 4\ 3\ +\ *\ * \ *$ (akzeptierend und Resultat)
- **(1P) Test b:** $3\ 1\ +\ 7\ 8\ +\ 9\ 8\ 7\ +\ 1\ 2\ 1\ 4\ +\ +\ 7\ +\ +\ +\ +\ +$ (akzeptierend und Resultat)
- **(1P) Test c:** $3\ 4\ +\ *$ (verwerfend) und $8\ +\ 9\ +\ 7\ * \ 2\ *$ (verwerfend)

Hinweis: Der Stack (Keller) darf nicht mit einer Bibliotheksklasse implementiert werden. Die Zahlen im Keller, die Zwischenergebnisse und das Endergebnis können mehrstellig sein.

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

10 Punkte