Anmerkungen zur Infix / Postfix – Umwandlung von arithmetischen Ausdrücken
HS / 5.11.02

1. Korrektur zu Folie alp3-ADT-3- 6

Ich wurde nach der VL am Dienstag auf einen Fehler auf der Folie („Infix to Postfix") hingewiesen (Danke!). An der Tafel stand's richtig. Auf der Folie steht

(1) Infix-Darstellung : *5+3 * 5 / 10* hat die
(2) Postfix-Darstellung: *5 3 5 10 / * +*

Das gilt nicht, da außer bei rechtsassoziativen Operatoren (oder expliziten Klammern) nach Konvention von links nach rechts geklammert wird. (1) wird also wie *5 + ((3*5) / 10)* ausgewertet.
Dagegen würde die Auswertung des Postfix-Ausdrucks (2) so erfolgen: *5 + (3*(5/10))* .

2. Algorithmus Infix nach Postfix

Grundlegend ist die Frage, wann bei der späteren Auswertung des Postfix-Ausdrucks ein Operator angewendet wird, damit das Ergebnis dem entspricht, was man nach den Konventionen der Berechnung von Infix-Ausdrücken erwartet. Grundsätzlich gilt: Bevor ein Operator in die Ausgabe geschrieben wird, müssen seine Operanden bekannt sein, sei es explizit oder durch Berechnung eines Teilausdrucks, der in der Postfix-Darstellung links vom Operator steht.
Hat man zum Beispiel einen Ausdruck *....(......) *c.....* muss später der geklammerte Ausdruck als Operand von * vor der Multiplikation ...* c berechnet werden. Deshalb muss der Operatorstapel geleert werden, bis eine öffnende Klammer erscheint.
Das gleiche gilt für Operatorpräferenzen, die nicht durch Klammern ausgedrückt werden. Heißt der Ausdruck *... a * b - c...* und ist * „ top of Stack", so soll später * vor - ausgewertet werden. In der Ausgabe muss stehen: *....a b * c ...* und evtl. gleich anschließend - (Beispiel für den Fall, dass - nicht unmittelbar in der Ausgabe auf *c* folgt?)

Damit ergeben sich die folgende Regeln::

(i) Operanden werden direkt in die Ausgabe geschrieben, d.h. sie werden an die bisherige Ausgabe angehängt.
(ii) Operatoren in der Eingabe kommen auf den Stapel, ggf. nachdem Operatoren vom Stapel entfernt und in die Ausgabe geschrieben wurden.
Sei *x* der in der Eingabe gelesene Operator, *y* „top of Stack".
a) *x* hat höhere Präferenz als *y* : push(*x*)
b) *x* hat geringere Präferenz: alle Operatoren *y, y'*,... werden nacheinander aus dem Stack entfernt (pop) und in die Ausgabe geschrieben, bis der Stapel leer ist oder ein Operator *z* gefunden wird, der geringere Priorität hat oder der gleiche Priorität wie *x* hat und rechtsassoziativ ist. Danach: push(*x*)

Den zweiten Teil der Regel (ii) veranschaulicht das Beispiel
a + b ^ c ^ d mit der Bedeutung *a +(b ^ (c ^d)).* Nach Lesen der ersten 6 Zeichen hat man die folgende Situation: Ausgabe: *a b c* Stapel: + ^ , gelesenes Z eichen *:* ^
^ ist rechtsassoziativ, also wird das gelesene Zeichen auf den Stapel gelegt, ohne vorher Operatoren zu entfernen. Das gilt, da wegen der Rechtsklammerung noch nicht alle Operanden des aktuellen Operators ^bekannt sind. Man stelle sich z.B. vor, dass der Ausdruck *a + b ^ (c ^(d + e))* heißt. Wenn man bis zum zweiten ^ gelesen hat, weiß man noch nicht wie

es weiter geht. Die Postfix-Darstellung muss garantieren, dass *d+e* vor allen anderen Teiltermen ausgerechnet wird.
Damit ergibt sich für $a + b \wedge c \wedge (d + e)$ die Postfix-Form: $a\ b\ c\ d\ \ e + \wedge\ \wedge +$

3. Repräsentation von Präferenzen und Assoziativität

Mit einem kleinen technischen Trick sorgt man dafür, dass leicht entschieden werden kann, ob Operatoren aus dem Stapel geholt werden müssen, bevor der gelesene Operator auf dem Stapel abgelegt wird. Neben der eigentlichen Präferenz wird eine „Stapelpräferenz“ für jeden Operator vergeben, mit der sich Links-, Rechtsassoziativität ausdrücken lässt. Sie ist für den Fall „gelesenes Operatorsymbol und oberstes Stapelelement sind gleich“ nützlich. Ist der Operator linksassoziativ, d.h. zu berechnen ist *...(a op b) op c ...*, soll das oberste Stapelelement zunächst in die Ausgabe geschrieben werden. Um das mit den obigen Regeln zu erzwingen, ist die Präferenz des Operator *op* auf dem Stapel um 1 höher als die Präferenz desselben Operators in der Eingabe. Bei rechtsassoziativen Operatoren *op’* soll der gleiche Operator *op’* nicht abgeräumt werden (pop) , bevor der gelesene auf den Stapel gelegt wird. Also erhält *op’* eine um 1 kleinere „Stapelpräferenz“. Es wird also die Regel (ii) a) angewendet.