

Infix- till postfixomvandling

Infix: *operand operator operand* exempel: a + b

Postfix: *operand operand operator* exempel: a b +

a + b - c	⇒	a b + c -
a + (b - c)	⇒	a b c - +
a * b + c	⇒	a b * c +
a * (b + c)	⇒	a b c + *
a / b + c * (d + e) - f	⇒	a b / c d e + * + f -

Varför postfix?

- Kan läsas och behandlas sekvensiellt.
 - ingen vetskap om vad som följer ("look-ahead") krävs.
 - läsning och bearbetning kan göras med t.ex. hjälp av stackteknik .
- Väl lämpad för maskinell bearbetning – t.ex. beräkning, kodgenerering, etc.

Infixuttryck med additiva och multiplikativa operatörer och parenteser

Operanderna är heltal eller enkla reella tal utan förtecken.

Operatorerna är +, −, * och /, samtliga *tvåställiga* (binära) och *vänsterassociativa*.

Parenteser kan förekomma för att avgränsa deluttryck och styra beräkningsordningen.

Prioritetsordning för operatorerna:

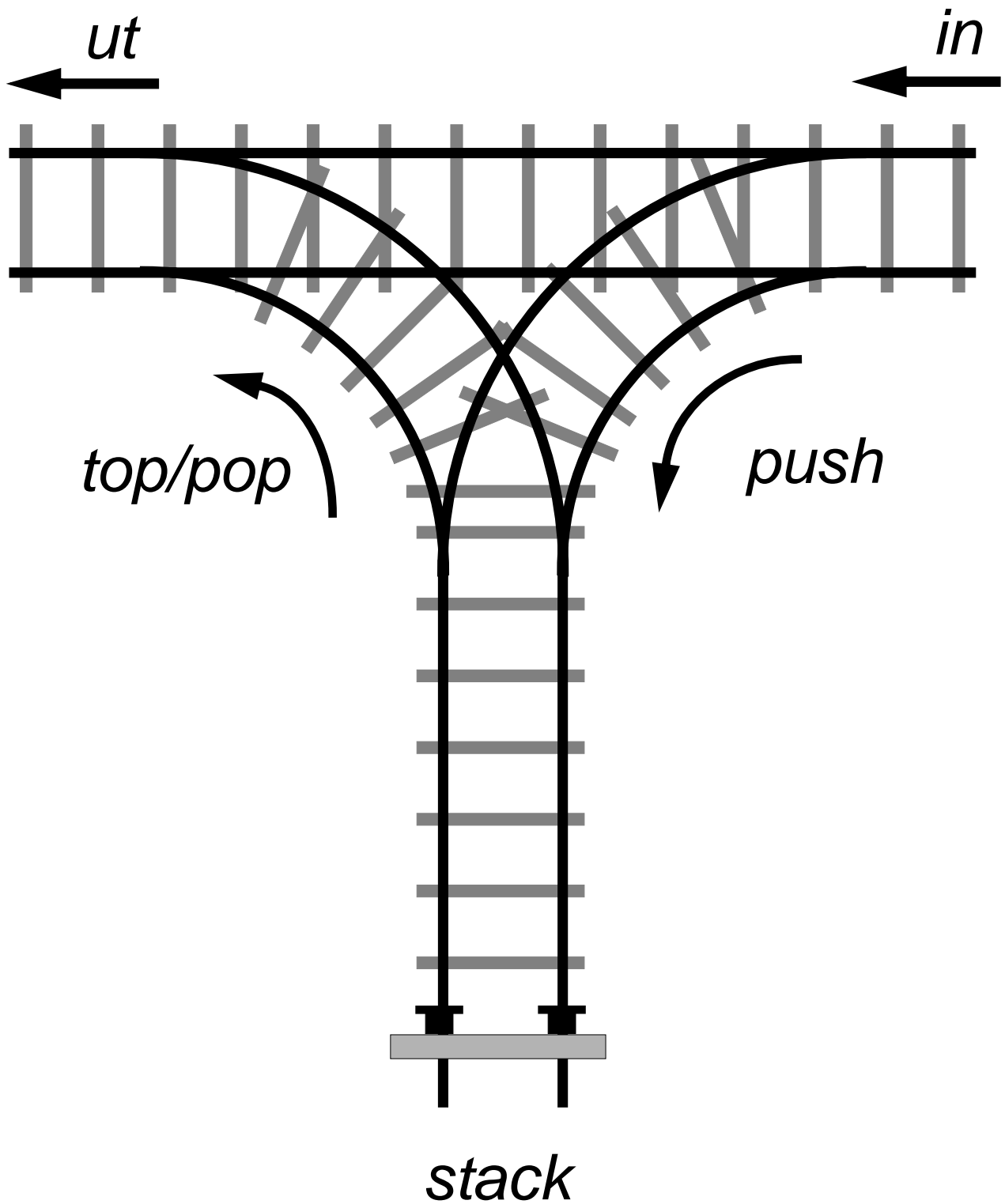
* /

+ −

(Även parenteserna kan ges prioritet)

"Dijkstras Järnvägsalgoritm"

Enkel "rangerbangård" realiserad med en stack.



Algoritm för infix-till-postfixomvandling

Inkommande	Åtgärd
* / + -	sålänge <i>en operator finns överst på stacken och prioriteten för operatorn på stacken \geq prioriteten för inkommande operator</i> { <i>ta bort och skriv ut operatorn på stacken</i> } <i>stacka den inkommande operatorn</i>
(<i>stacka '('</i>
)	sålänge <i>stacken ej innehåller '(' överst</i> { <i>ta bort och skriv ut operatorn överst på stacken</i> } <i>ta bort '(' från stacken (ska finnas om infixuttrycket så långt är korrekt)</i>
<i>eol</i>	<i>ta bort och skriv ut alla operatorer på stacken</i>
<i>operand</i>	<i>skriv ut operanden</i>

Felkontroller ej införda.

Exempel 1

 $a * b + c / d$

<i>Inkommande</i>	<i>Stack</i>	<i>Ut</i>
a		a
*	*	a
b	*	a b
+	+	a b *
c	+	a b * c
/	+ /	a b * c
d	+ /	a b * c d
<i>eol</i>		a b * c d / +

Exempel 2

$$(a + b) * c + (d - e)$$

<i>Inkommande</i>	<i>Stack</i>	<i>Ut</i>
((
a	(a
+	(+	a
b	(+	a b
)		a b +
*	*	a b +
c	*	a b + c
+	+	a b + c *
(+ (a b + c *
d	+ (a b + c * d
-	+ (-	a b + c * d
e	+ (-	a b + c * d e
)	+	a b + c * d e -
<i>eol</i>		a b + c * d e - +

Beräkning av postfixuttryck

Postfixuttrycket läses sekventiellt och beräknas med hjälp av en stack:

En *operand* som läses ska stackas.

För en *operator* ska följande utföras:

- De två översta värdena på stacken (ska finnas för korrekta uttryck) tas bort från stacken.
 - det översta värdet ska vara höger operand.
 - det näst översta värdet ska vara vänster operand.
- Operationen utförs på operanderna.
- Det erhållna värdet stackas.