Koding av syntaks

La oss med et eksempel vise hvordan syntaks kan kodes. Vi ser på utsagnskalkyle. Den er bygd opp av utsagnsvariable og utsagn. De kan skrives slik på BNF-form

$\mathcal{V}$ ::= x | $\mathcal{V}$*

$\mathcal{U}$ ::= $\mathcal{V}$ | $\neg\mathcal{U}$ | ($\mathcal{U}\wedge\mathcal{U}$) | ($\mathcal{U}\vee\mathcal{U}$)

Vi har altså kategoriale symboler: $\mathcal{U}$ $\mathcal{V}$ og terminale symboler : x * $\neg$ $\wedge$ $\vee$ ( ) . Vi har valgt å la følgende være utsagnsvariable x x* x** x*** x**** …. . Det ser ut til å være et OK valg.

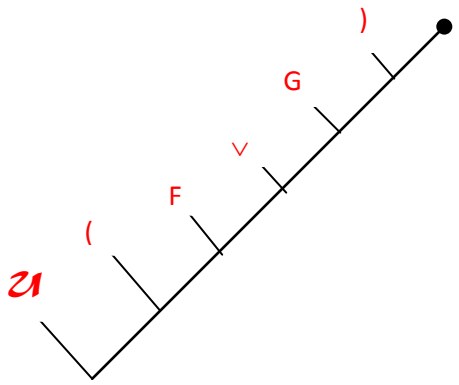
Først bestemmer vi oss for hvordan de grunnleggende symbolene skal representeres som par. Her det helt opp til en selv. For et uttrykk F skriver jeg koden til F i rød skrift som F . Og så bruker jeg – for å skrive litt mindre – følgende koding av tall som er brukt i monografien

0 = nil

n+1 = (nil . n)

Så til kodingen av de 9 grunnleggende symbolene $\mathcal{U}$ $\mathcal{V}$ x * $\neg$ $\wedge$ $\vee$ ( ) . Jeg koder dem som parene 0 1 2 3 4 5 6 7 8 . Det kunne være en hvilken som helst samling av 9 forskjellige par.

Nå må vi forklare hvordan konstruktorene er med på å bygge opp kode for sammensatte uttrykk. En rett fram idé er å kode dette inn som en sekvens der første element viser om det er en variabel eller et utsagn og de resterende elementene gir kodingen av elementene i stringen. Da koder vi utsagnet (F $\vee$ G) inn ved



Her er F og G igjen par-uttrykk som vi må sette inn igjen.

Når vi har satt inn alt, så står vi tilbake med et stort par-uttrykk.

Vi legger merke til at om et par-uttrykk er kodingen til et velformet utsagn er noe som kan uttrykkes som en Δ_0-setning – vi trenger bare kvantorer som er begrenset.