Tall og telling

Hvordan bruke tall en ikke kan telle til

HRJ – LOGID – 07.03.2013

Vi teller overalt

- Compute – regning ved skjæring
- Calculate – regning med småstein
- Kontor – computorium – sted vi regner
- Algoritme – Al Khowarizmi (mannen fra Khiva) viste i Mesopotamia hvordan en kunne regne
- Teller opp til 10000 (myriade, maurtue) – det vi kan se som en mengde av enkelt ting

Hvor langt kan vi telle

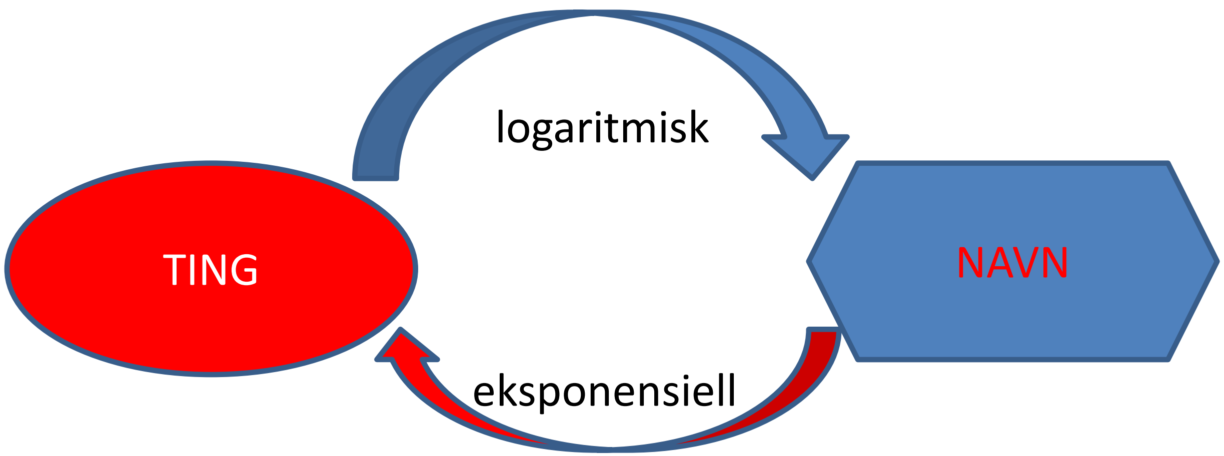
- 30×10^6 sekunder i et år
- 13×10^9 år - universets alder
- 2×10^{44} planck-tid i et sekund
- Med PC-kommer vi lenger
- *Main> product[1 .. 100]
- 93326215443944152681699238856266700490715968
  26438162146859296389521759999322991560894146
  39761565182862536979208272237582511852109168
  64000000000000000000000000
- Et tall med 158 sifre

Telling med kroppsdeler

- En – to – mange   (tre som trans, tres)
- Ti – taihun (?to hender?)
- Elve – tolv (en – to til overs)
- Hundre - taihun – taihun (to hender av to hender)
- Tusen - tus-hundre
- Million - (banksystemet i Italia 1600)
- Milliard – (fransk-tyske krig 1870-71)
- Arkimedes – Sandregneren - 10^{63}

Unære – binære tall

- Unært : IIIIIIIIIIIIIIIIIIII
- Binært : 10100111
- 10-tallsystemet : Fra India / Mesopotamia



Posisjonssystemet

- Binært (desimalt) system
- Brukes ved indekssering
- Leksikografisk ordning
- Posisjonene er unære
- Vitenskaplig notasjon 3.16×10^7
  - Posisjonene er binære
  - To eksponenter opp - unært

Eget system ∈

- Unære system bygger opp alt rundt 0 og s
  - 0 s0 ss0 sss0 ssss0 sssss0
  - 0 – start s – suksessor, etterfølger, konstruktor
- Eget system bygger opp alt rundt 0 og e
  - exy = $2^x + y$
  - e00 = 1
  - ee000 = e10 = 2
  - eeeeee0000000 = 2^{256} – atomer i universet

Aksiomer for E - Ax

1. $0 : \mathcal{N}$
2. $x : \mathcal{N} \quad \Longrightarrow \quad sx : \mathcal{N}$
3. $e0y = sy$ (siden $2^0+y = y+1$)
4. $esxy = exexy$ (siden $2^{x+1}+y = 2^x+2^x+y$)

Disse aksiomene kan behandles mekanisk.

1,2 er aksiomene for datastruktur

3,4 er likhetsaksiomer

Test av Ax

? eeeeee0000000 : $\mathcal{N}$?

Mekanisk – trenger mer enn 2^{256} trinn

Ved å bruke hjelpesetninger < 100 trinn

Problem – mekanisere hjelpesetninger

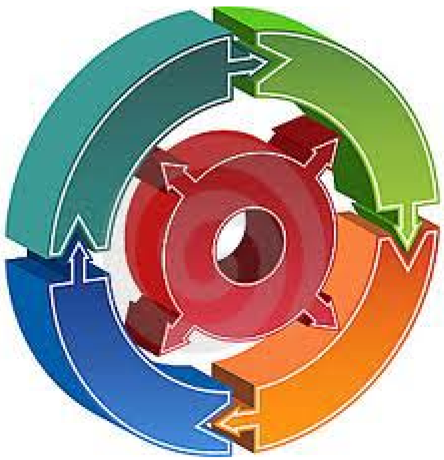
Hjelpestørrelser

- $\mathcal{N}_0 = \mathcal{N}$
- $x{:}\mathcal{N}_{i+1} \quad \longleftrightarrow \quad \forall y{:}\mathcal{N}_i \,.\, exy : \mathcal{N}_i$

Viser: $0 : \mathcal{N}_i$ for alle i (kort bevis)

- $0 : \mathcal{N}_6$ og $0{:}\mathcal{N}_5$ gir $e00{:}\mathcal{N}_5$
- $0{:}\mathcal{N}_4$ gir $ee000 : \mathcal{N}_4$
- $0{:}\mathcal{N}_3$ gir $eee0000 : \mathcal{N}_3$
- $0{:}\mathcal{N}_2$ gir $eeee00000 : \mathcal{N}_2$
- $0{:}\mathcal{N}_1$ gir $eeeee000000 : \mathcal{N}_1$
- $0{:}\mathcal{N}_0$ gir $eeeeee0000000 : \mathcal{N}_0 = \mathcal{N}$

Prosesser



FOR : Vet hvor mange ganger en går rundt i løkka ved start

WHILE : Tester om en er ferdig ved utgang av løkka

Knuths system

- $m \times n = m + m + \ldots + m$
- $m \uparrow n = m \times m \times \ldots \times m$
- $m \uparrow\uparrow n = m \uparrow m \uparrow \ldots \uparrow m$
- $m \uparrow\uparrow\uparrow n = m \uparrow\uparrow m \uparrow\uparrow m \ldots \uparrow\uparrow m$
- …….
- Gjentar n ganger
- Parenteser den vonde veien
- $\uparrow\uparrow$ gir tårn av eksponenter

Ackermann funksjonen

- $1 \uparrow 1$ - $2 \uparrow\uparrow 2$ - $3 \uparrow\uparrow\uparrow 3$ - $4 \uparrow\uparrow\uparrow\uparrow 4$ - ……
- Vokser raskere enn enhver FOR-prosess.
- Trenger måter å måle fortgang i prosesser

Telle nedover

- Five – four – three – two – one – FIRE

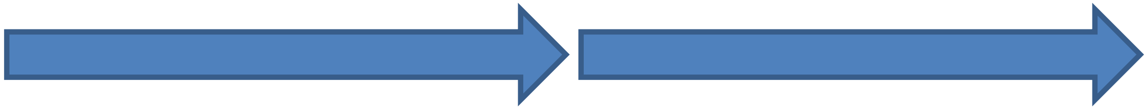
Ordinaltall

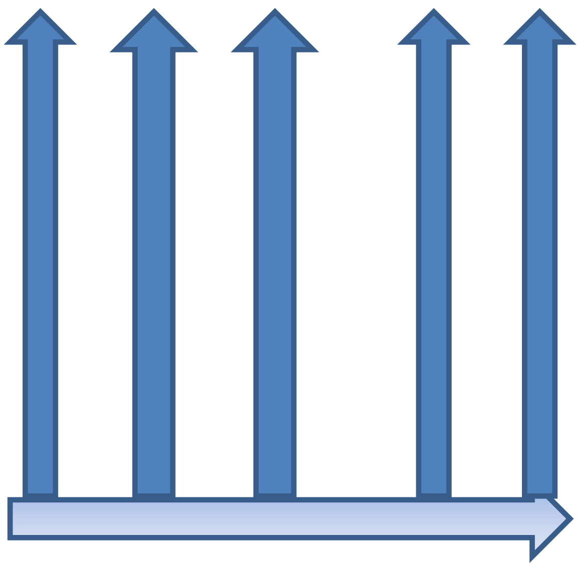
- Utvidelse av naturlige tall til det uendelige

- To hovedideer
  - brukes til å telle ned
  - brukes til å beskrive utsatte beslutninger

- Endelige ordinaltall
- Omega = ω = Minste uendelige ordinaltall

Noen ordinaltall

- ω = 
- $\omega + \omega$ = 
- $\omega \times \omega$ =



Leksikografisk ordning av par

Eksempler

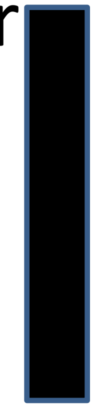
- $1 + \omega = \omega$

- $\omega + 1$

- Teller nedover på forskjellig måte

Kulespill

- Bokser som inneholder et endelig antall kuler



- Kan ta ut 1 kule fra en boks og plassere et endelig antall kuler i boksene til høyre
- Prosessen vil terminere
- Bruker ordinaltall til å beskrive fortgang

- $\omega^5 7 + \omega^4 2 + \omega^2 8 + \omega^1 3 + 1$

Utsatte beslutninger

- Kan beskrive prosessen selv om beslutninger er utsatt
- Vi vet ikke hvor mange trinn prosessen vil ta
- Men vi vet når vi vil kunne få vite
- Trenger uendelige ordinaltall
- Kulespill kan beskrives med ordinaltall mindre enn ω^ω
- Leksikografisk ordning av endelige sekvenser av tall

FOR-prosesser

- Kan beskrives på samme måte som kulespillet
- Bruker ordinaltall $< \omega^\omega$
- Løkkestrukturen i FOR-program svarer til boksene i kulespillet

- Og slik kan vi gå videre og videre

- Men