

Matematikens domänspecifika språk DAT326 / DIT982

(Pres. for TM3, Ti 2020-11-10 av Patrik Jansson.)

- ▶ Patrik Jansson, forskare & lärare i funktionell programmering
 - ▶ i gränsområdet mellan matematik och datavetenskap

Presentera klassiska matematiska ämnen från ett datavetenskapligt perspektiv:

- ▶ att tydligt beskriva de begrepp som introduceras,
- ▶ vara uppmärksam på syntax och typer,
- ▶ att bygga domänspecifika språk för: algebra, funktioner, derivator, polynom och potensserier, transformer
- ▶ att implementera (Haskell-)program för dessa områden (och därigenom nå en djupare förståelse)

Kursen är tänkt att ge en fördjupad matematisk förståelse för datastudenter och en fördjupad datavetenskaplig förståelse för matematikstudenter. (11 TM-studenter har gått kursen)

Student-citat

- ▶ “I feel like I have gotten a better understanding of mathematics, and how to make use of domain-specific languages.”
- ▶ “The teachers (and assistants) were very engaging.”
- ▶ “I found the ‘alternative’ perspective on common mathematical subjects in general to be very good. The part about logic and the part about linear algebra were my personal favorites.”
- ▶ “Patrik is an excellent lecturer for the course.”

"Jag läste DSLofMath för ett par år sedan och hade lite svårt att förstå syftet med kursen. Jag tyckte att den var lite luddig.

Nu har jag läst ... “Advanced Functional Programming”, “Programming Language Technology”, och “Types for Programs and Proofs”, och nu förstår jag.

Kursen har verkligen varit till hjälp, även om jag inte var mogen att förstå det när det begav sig."

Funktionell programmering (FP)

- ▶ Rena funktioner som bas: från indata till utdata
- ▶ Typer (`Int`, `String`, `[Int]`, `Int -> Int`, `[Int->Int]`, ...)
- ▶ Historik: matematiska bevisverktyg och algoritmer
- ▶ Nutid: Konkurrensfördel, FP-experter eftertraktade!
- ▶ Vackert möte mellan matematik och maskin.

```
kvadrat x = x^2
```

```
twice f = \x -> f(f(x))
```

```
upphöjtFyra = twice kvadrat
```

```
test = upphöjtFyra 3
```

```
list = [(1+), (2*), kvadrat, upphöjtFyra]
```

Funktionell programmering (FP) och *typer*

- ▶ Typer (Int, String, [Int], Int -> Int, [Int->Int], ...)
- ▶ Rena funktioner som bas: från indata till utdata
- ▶ Historik: matematiska bevisverktyg och algoritmer
- ▶ Nutid: Konkurrensfördel, FP-expert eftertraktade!
- ▶ Vackert möte mellan matematik och maskin.

```
kvadrat      :: Num a => a -> a
kvadrat x    = x^2
twice        :: (a -> a) -> (a -> a)
twice f      = \x -> f(f(x))
upphöjtFyra  :: Num a => a -> a
upphöjtFyra  = twice kvadrat
test         :: Double
test         = upphöjtFyra 3
list         :: Num a => [a -> a]
list         = [(1+), (2*), kvadrat, upphöjtFyra]
```

Matematikens domänspecifika språk

- ▶ finslipad förmåga att formalisera
- ▶ Funktionell programmering ger nya verktyg
- ▶ Mötet mellan matematiken och datavetenskapen
 - ▶ det är där det händer riktigt spännande saker ...

Välkomna! /Patrik

$$f(x) = 3 * x^2$$

$$g(x) = \int_x^{2x} f(x) dx \qquad = \int_x^{2x} f(t) dt$$

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = 0$$

Förkunskaper

Studenten ska ha klarat (+TM-exempel):

- ▶ en kurs i diskret matematik
 - ▶ TM1: en kombination av Mat.prog., Lin.alg&geometri, Mat.orientering, Sannolikhet, ...
- ▶ två andra kurser i matematik
 - ▶ TM1: exempelvis Linjär algebra, Analys I och II
- ▶ två kurser i datateknik
 - ▶ TM1: Programmeringsteknik + TM2: Datastrukturer
- ▶ ytterligare tre kurser (22.5hp) inom matematik, data eller IT
 - ▶ TM1: LinAlg&NumAn, Flervariabel + TM2: Komplex, ...

Rekommenderat (men ej krav): Funktionell programmering #
Matematikens domänspecifika språk (DSLsofMath) för andra ämnen

Inspirerat av den här kursen har flera studentgrupper genomfört kandidatarbetsprojekt under de senaste åren med följande resultat:
(*TM-student*)

- ▶ 2016: Programmering som undervisningsverktyg för Transformer, signaler och system - Utvecklingen av läromaterialet TSS med DSL

Mer om kursen: Aktiva studier

- ▶ föreläsningar växlat med övningar + grupparbete.
- ▶ Aktiva studier: det räcker inte att bara "läsa (eller lyssna) och hålla med"
- ▶ TM är en bra bakgrund: mat. problemlösning & logik \leadsto programmering och datavetenskap
- ▶ plocka isär & reda ut begrepp \rightarrow källkod (funktioner och *typer*)
- ▶ datorn (kompilatorn) ger direkt återkoppling när något inte stämmer.

Bakgrund: vad är ett “domänspecifikt språk” (DSL)?

Exempel:

- ▶ datum:
 - ▶ Syntax: “2015-11-18”, “tredje onsdagen i oktober”, “nästa lördag”
 - ▶ Semantik: Date, eller kanske Date -> Date
- ▶ excel-formler:
 - ▶ Syntax: “SUM(A1:A9)”, “RIGHT(LEFT(C7,4),2)”, ...
 - ▶ Semantik: [[Cell]] -> Value
- ▶ integraler: [Behöver ritas!]
 - ▶ (notera var variablerna binds)