

# Programmering

Seminarier i datavetenskap, datorteknik och informationsteknik

Niklas Broberg

[niklas.broberg@chalmers.se](mailto:niklas.broberg@chalmers.se)

2018-09-27

# Hur många från...

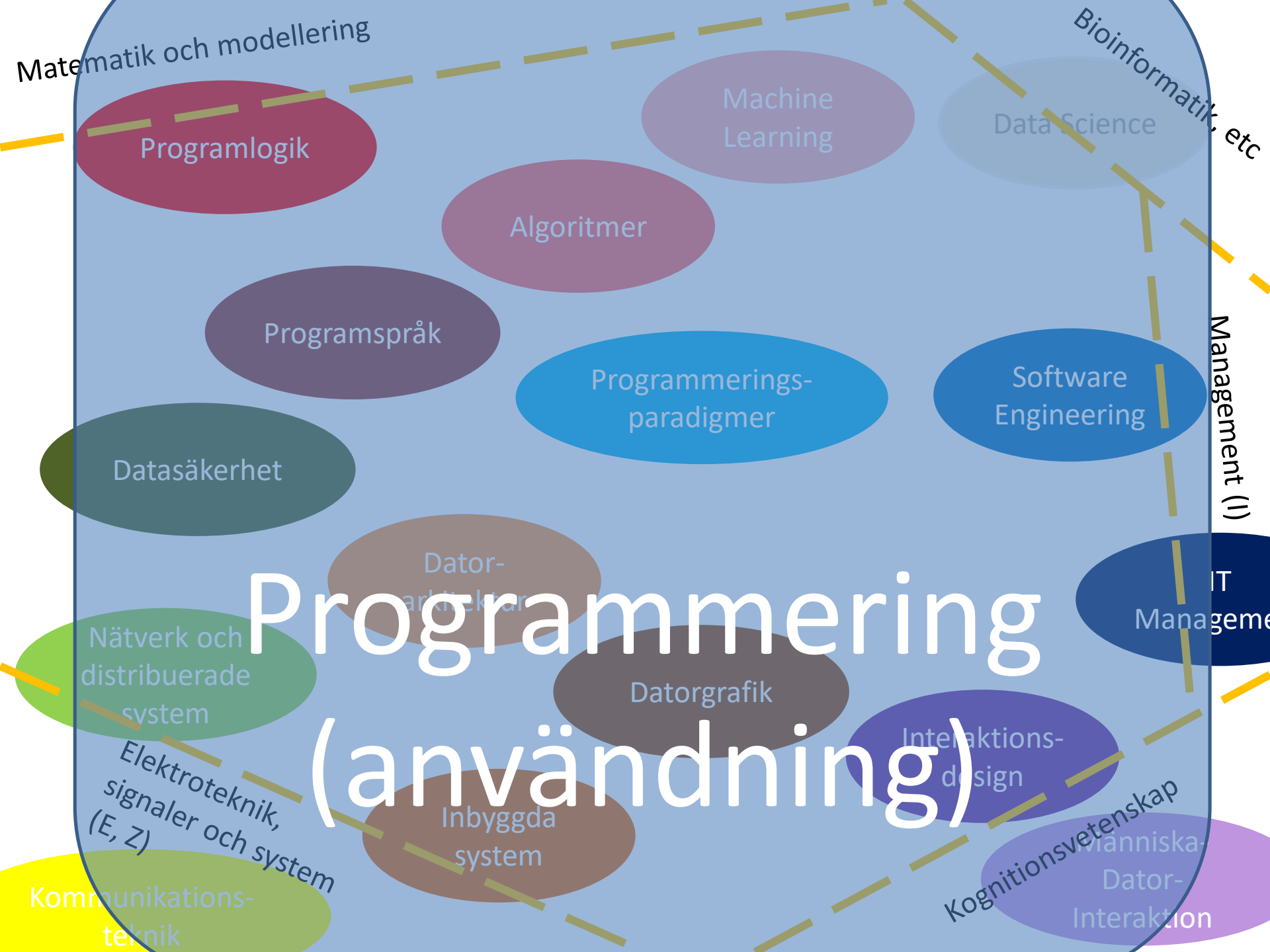
- Datavetenskap?
- Datateknik?
- Informationsteknik?

# Översikt

- Programmering som koncept
  - Vad är det? Vad omfattas rent generellt? Hur förhåller sig programmeringen till andra områden inom D&IT?
- Programmering i er utbildning
  - Vad kommer ni möta? Vad skiljer de tre studieprogrammen åt? Vilka möjligheter finns till fördjupning?

# Programmering – vad är det?

- Kommunikation mellan programmerare och dator
- Konkret användning av ett programspråk, bibliotek, ramverk, ...
- Modeller, paradigmer, mönster, algoritmer, ...
- (Möjligheter och begränsningar, beräkningsbarhet, ...)



# Programmering (användning)

Matematik och modellering

Programlogik

Machine Learning

Bioinformatik, etc

Data Science

Algoritmer

Programspråk

Programmeringsparadigmer

Software Engineering

Management (IT)

Datasäkerhet

Datorarkitektur

IT Management

Nätverk och distribuerade system

Datorgrafik

Interaktionsdesign

Inbyggda system

Kognitionsvetenskap

Elektroteknik, signaler och system (E, Z)

Människa-Dator-Interaktion

Kommunikationsteknik

# Abstraktion vs precision

- ”Hög-nivå vs låg-nivå”
  - Hur utvecklar vi program så effektivt som möjligt?
  - Hur utvecklar vi så effektiva program som möjligt?
- Abstraktion (”hög-nivå”) *kan* hjälpa till med båda
  - ... men ibland krävs precision för det senare (ex Datorgrafik)

# Paradigmer

- Sätt att tänka kring programstruktur
- Objekt-orienterad, imperativ, funktionell, reaktiv, parallell, ...
  - Inga tydliga gränser
- Olika programspråk har inbyggt stöd för vissa paradigmer
  - ... men de flesta paradigmer går att applicera i de flesta språk – bara med olika mycket stöd

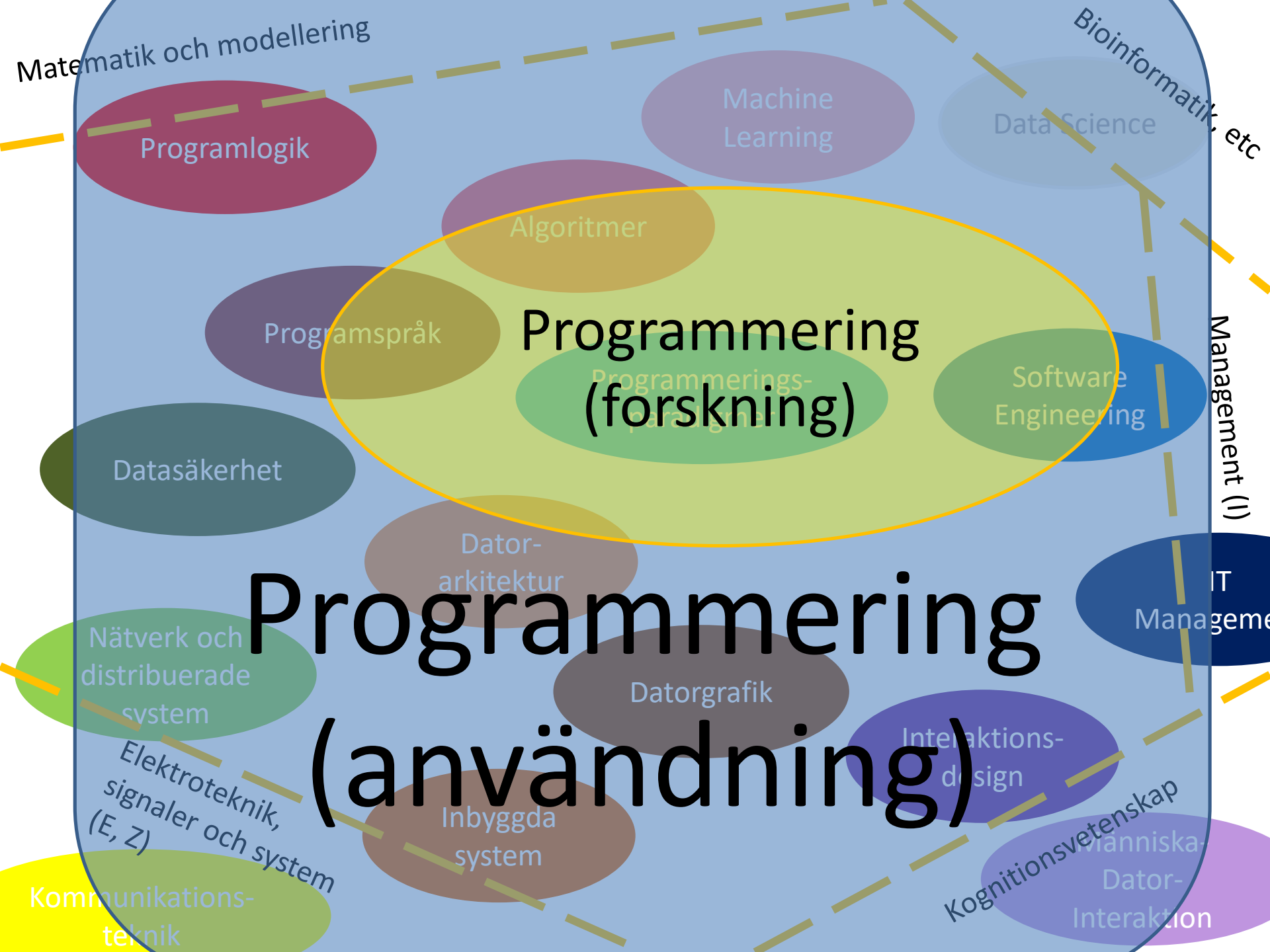
# Mönster

- Givet en paradigm, hur strukturerar jag mitt program?
  - Inbyggda verktyg ("vad"): Funktioner, metoder, objekt, algebraiska datatyper, typklasser, generiska metoder, ...
  - Användningsmönster ("hur"): Events, observers, stream transformers, semaphores, factories, facades, ...
- Ett mönster går oftast att uttrycka i vilket språk som helst – det är bara mer eller mindre komplicerat
  - Olika språk kan ha inbyggt stöd för olika mönster



# Forskning om programmering?

- Hur kan vi skriva mer effektiva program?
  - Algoritmer, programspråk, datorarkitektur
- Hur kan vi utveckla program mer effektivt?
  - Software engineering, programspråk
- Hur kan vi göra programmering lättare?
  - Programspråk, paradigmer



# Vi utbildar inte programmerare!

- Vi utbildar datavetare och ingenjörer!
- Mycket bredare roller än ”programmerare”

# Plats för alla!

- Alla behöver inte (bör inte!) bli programmerare!
- ... men alla behöver förstå programmering.

# Tre studieprogram

- **Datavetenskap**
  - Väldigt brett, (extra) flexibelt program
  - Spetsfokus mot modellering och korrekthet
- **Datateknik**
  - DATAvetenskap + datorTEKNIK
  - Flexibelt – men spetsfokus mot hårdvara
- **Informationsteknik**
  - Flexibelt – men spetsfokus mot mjukvara

# Programmering i utbildningen

- Vi tillhandahåller tre tydliga spår:
  - Imperativ/Objekt-orienterad programmering
  - Maskinnära programmering
  - Funktionell programmering

# Objektorientering (OO)

- Imperativ/OO programmering (introduktionskurs)
  - Grundläggande imperativ programmering, introduktion till objekt och OO-tänkande (Java)
- OO programmering och design (fortsättningskurs)
  - Stort fokus på OO-tänkande
- OO programmeringsprojekt
  - Applicera i praktiken
- (Modellering, processer, Software Engineering (Master))

# Maskinnära programmering

- Grundläggande datorteknik
  - Introduktion till lågnivå-programmering, assembler
- Maskinorienterad Programmering
  - Fokus på lågnivå-programmering (C)
- Datatekniskt projekt
  - Applicera i praktiken
- (Inbyggda system (Master), datorsystem, hårdvarukonstruktion ...)
- Mer på framtida seminarium om Digital konstruktion och inbyggda system



# Funktionell programmering (FP)

- (Introduktion till) FP
  - Grundläggande funktionell programmering, introduktion till FP-tänkande
- Avancerad Funktionell Programmering
  - Stort fokus på FP-tänkande
- (Parallell FP)
- Mer på framtida seminarium om Funktionell programmering

# Breda koncept

- Datastrukturer, Algoritmer
  - Applicerbart inom alla paradigmer (om än med viss specialisering)
  - Läses av alla programmen
  - Mer om detta på senare seminarium om Algoritmer
- Parallell programmering
  - Arbeta med trådar, processer och parallellism

# Bredd vs spets

- Bredd: Läs alla introduktionskurser!
  - Ger en större förståelse än summan av komponenterna
- Spets: Fokusera på ett område för framtida specialisering
  - Största skillnaden mellan programmen

# Datavetenskap

- LP1: Introduktion till Funktionell Programmering (DIT440)
- LP2: Imperativ programmering med grundläggande objektorientering (DIT012)
  - Första OO-kurs
- LP3: Objektorienterad programmering och design (DIT953)
  - Andra OO-kurs
- År 2/3: Parallell programmering (DIT391); massor av valbart!
  - Ex: Grundläggande datorteknik (DIT791); Maskinorienterad programmering (DIT152); Advanced FP (DIT260);

# Datateknik

- LP1: Introduktion till Funktionell Programmering (TDA555)
- LP2: Grundläggande Datorteknik (EDA452)
  - Första maskinnära kurs (assembler)
- LP3: Objektorienterad programmering (DAT043)
  - Första OO-kurs
- LP4: Maskinorienterad Programmering (EDA482)
  - Andra maskinnära kurs (C)
- År 2/3: Datatekniskt projekt (DAT290); valbart.
  - Ex: Parallell Programmering (TDA383); Advanced FP (TDA342)

# Informationsteknik

- LP1: Grundläggande programvaruutveckling (TDA548)
  - Första OO-kurs
- LP1: Grundläggande datorteknik (EDA433)
  - Första maskinnära kurs (assembler)
- LP2: Objektorienterad programmering och design (TDA552)
  - Andra OO-kurs
- LP3: Maskinorienterad programmering (DAT017)
  - Andra maskinnära kurs (C)
- År 2/3: Objektorienterat programmeringsprojekt (TDA367); valbart.
  - Ex: Parallell Programmering (TDA383); Funktionell Programmering (TDA452); Advanced FP (TDA260);

# Vidare möjligheter

- Mer i kommande seminarier om:
  - Inbyggda system (vt)
    - Fokus på maskinnära program
    - Precision, hårdvaruintegrering
  - Programspråksteknologi (vt)
    - Teorier och mekanismer bakom programspråk
  - Software Engineering (lp2)
    - Fokus på mjukvaruutveckling
    - Modeller, högre-nivå-abstraktion
    - Processer för utveckling

# Kursutvärdering

- För introprogrammeringen, nu utskickad!



Frågor?