

Datalogi som vetenskap

Är datalogi vetenskap?

Varför skulle datalogi inte vara det?

Skäl för att datalogi är vetenskap:

- Det ges akademiska kurser i ämnet.
- Det publiceras forskningsartiklar i ämnet.
- Det finns professorer i ämnet.
- Ämnet använder vetenskapliga metoder.
- Ämnet berör fundamental vetenskapliga frågeställningar.

Skäl mot

- Datalogi sysslar bara med saker som uppfunnits av människor. Ämnet studerar inte naturen (och är därför inte naturvetenskap).
- Datalogi har ingen samlad teori. Det är en samling knep för att lösa olika problem.
- Datalogi är en *teknik* istället för en vetenskap.
- Var är datalogins relativitetsteori? Var är dess evolutionslära? Var är dess storslagna insikter?

Vad är då datalogi?

Det finns ingen definition som är allmänt accepterad. (Några förslag presenteras snart.)

1969 argumenterade en forskare att datalogin styrs av en fundamental fråga som också är en central fråga i den mänskliga civilisationen.

Vad kan automatiseras?

Historisk utveckling av datalogi

- Från början var datalogi i huvudsak algoritmt teori och matematik.
- Sedan kom problem kring semantik hos programspråk.
- På senare tid har frågan om hur människor använder datorer blivit betydelsefull. Vi ser en övergång från datorcentrerad teori till humancentrerad teori.

En grov och tentativ uppdelning

- Det som vi kallar datalogi kan också kallas datavetenskap. Vi kan (i den här kursen åtminstone) göra följande uppdelning:
- Teoretisk datalogi handlar om algoritmteori, komplexitet, mer fundamentala semantiska frågor och liknande.
- Datalogi handlar dessutom om mer konkreta programmeringsfrågor.
- Datavetenskap handlar dessutom om frågor kring hur människor använder datorer.

Vi får då följande delfrågor:

- Är teoretisk datalogi en vetenskap?
- Är datalogi en vetenskap?
- Är datavetenskap en vetenskap?
- Åtminstone den första frågan besvaras nog alltid med ja.

Några forskningsproblem

Stanford Encyclopedia of Philosophy

- What kinds of things are programs? Are they abstract or concrete? (Moor 1978; Colburn 2004)
- What are the differences between programs and algorithms? (Rapaport 2005a)
- What is a specification? And what is being specified? (Smith 1985; Turner 2005)
- Are specifications fundamentally different from programs? (Smith 1985)

- What is an implementation? (Rapaport 2005b)
- What distinguishes hardware from software?
Do programs exist in both physical and symbolic forms? (Moor 1978; Colburn 2004)
- What kinds of things are digital objects? Do we need a new ontological category to house them? (Allison et al. 2005)
- What are the objectives of the various semantic theories of programming languages? (White 2004; Turner 2007)

- How do questions in the philosophy of programming languages relate to parallel ones in the philosophy of language? (White 2004)
- Does the principle of modularity (e.g., Dijkstra 1968) relate to the conceptual issues of full-abstraction and compositionality?
- What are the underlying conceptual differences between the following programming paradigms: structured, functional, logic, and object-oriented programming?
- What are the roles of types in Computer Science? (Barandregt 1992; Pierce 2002)

- What is the difference between operational and denotational semantics? (Turner 2007)
- What does it mean for a program to be correct? What is the epistemological status of correctness proofs? Are they fundamentally different from proofs in mathematics? (DeMillo et al. 1979; Smith 1985)
- What do correctness proofs establish? (Fetzer 1988; Fetzer 1999; Colburn 2004)
- What is abstraction in computer science? How is it related to abstraction in mathematics? (Colburn & Shute 2007; Fine 2008; Hale & Wright 2001)

- What are formal methods? What is formal about formal methods? What is the difference between a formal method and informal one? (Bowen & Hinchey 2005; Bowen & Hinchey 1995)
- What kind of discipline is computer science? What are the roles of mathematical modelling and experimentation? (Minsky 1970; Denning 1980; Denning 1981; Denning et al. 1989; Denning 1985; Denning 1980b; Hartmanis 1994; Hartmanis 1993; Hartmanis 1981; Colburn 2004; Eden 2007)

- Should programs be considered as scientific theories? (Rapaport 2005a)
- How is mathematics used in computer science? Are mathematical models used in a descriptive or normative way? (White 2004; Turner 2007)
- Does the Church-Turing thesis capture the mathematical notion of an *effective* or *mechanical* method in logic and mathematics? Does it capture the computations that can be performed by a human? Does its scope apply to physical machines? (Copeland 2004; Copeland 2007; Hodges 2006)
- Can the notion of *computational thinking* withstand philosophical scrutiny? (Wing 2006)

- What is the appropriate logic with which to reason about program correctness and termination? (Hoare 1969; Feferman 1992) How is the logic dependent upon the underlying programming language?
- What is information? (Floridi 2004; Floridi 2005) Does this notion throw light on some of the questions listed here?
- Why are there so many programming languages and programming paradigms? (Krishnamurthi 2003)

- Do programming languages (and paradigms) have the nature of scientific theories? What causes a programming paradigm shift? (Kuhn 1970)
- Does software engineering raise any philosophical issues? (Eden 2007)

Klassificering av de olika vetenskaperna

Vetenskap	Objekt	Metod
Logik/Matematik	Abstrakta objekt	Deduktion
Naturvetenskap	Fysiska objekt, fält, levande organismer	Hypotetisk-deduktiva metoden
Samhällsvetenskap	Sociala objekt, människor grupper , samhällen	Hypotetisk-deduktiva metoden. Hermeneutik
Humanvetenskaperna	Kulturella objekt, idéer språk	Hermeneutik

Beskrivning av vetenskaplig metod igen

1. Ställ frågan uttryckt mot bakgrund av tillgänglig kunskap.
2. Formulera en hypotes.
3. Härled konsekvenser och gör förutsägelser.
4. Testa hypotesen.
5. Om hypotesen klarar testen etableras den som en ny teori.
6. Jämför olika teorier och välj de mest framgångsrika.

Metod kontra filosofi

Vi kan ställa följande frågor:

- Vilka är datalogins/datavetenskapens vetenskapliga metoder?
- Vilka är datalogins/datavetenskapens centrala vetenskapsfilosofiska problem?
- För att besvara den första är det naturligt att fokusera på datavetenskap.
- För att besvara den andra är det kanske naturligt att fokusera på datalogi.

Skillnaden mellan vetenskap och teknik

	Vetenskap	Teknik
Objekt	Oföränderliga	Föränderliga
Mål	Förstå det generella	Förstå det speciella
Aktivitet	Mål i sig självt	Mål i något annat
Metod	Abstraktion	Modellering
Processer	Konceptualisering	Optimering
Nyhet	Upptäckt	Uppfinning
Resultat	Lagar	Regler
Tid	Lång	Kort

*Är datalogi/datavetenskap en
vetenskap eller en teknik?*

Olika definitioner av DV

1. The discipline of Computing is the systematic study of algorithmic processes that describe and transform information: their theory, analysis, design, efficiency, implementation and applications.

2. Computer Science is the study of phenomena related to computers, Newell, Perlis and Simon, (1967).
3. Computer Science is the study of information structures, Wegner, (1968)

4. Computer Science is the study and management of complexity, Dijkstra, (1969)
5. Computer Science is the mechanization of abstraction, Aho and Ullman (1992)

6. Computer Science is a field of study that is concerned with theoretical and applied disciplines in the development and use of computers for information storage and processing, mathematics, logic, science, and many other areas

Underområden av DV

1. Discrete Structures
2. Programming Fundamentals
3. Algorithms and Complexity
4. Programming Languages
5. Architecture and Organization
6. Operating Systems
7. Net-Centric Computing
8. Human-Computer Interaction

- 9. Graphics and Visual Computing
- 10. Intelligent Systems
- 11. Information Management
- 12. Software Engineering
- 13. Social and Professional Issues
- 14. Computational Science and Numerical
Methods

Vetenskapliga metoder inom datalogi/datavetenskap

När vi talar om vetenskapliga metoder kan vi välja mellan två olika perspektiv.

1. Det hårda (reduktionistiska) perspektivet. Vi studerar bara datalogi eller teoretisk datalogi. Den enda metoden är deduktion och specialvarianter av deduktion.
2. Det mjuka (holistiska) perspektivet. Vi studerar datavetenskap i sin helhet.

Vetenskapliga metoder inom DV

Det mjuka synsättet

Fyra viktiga metodområden:

1. Modellering
2. Teoretisk datalogi
3. Experimentell DV
4. Datorsimulering

Modellering

Modellering förekommer i stort sett inom alla vetenskaper. Vi ersätter något komplext med en enklare struktur uttryckt i ett språk som t.ex matematik.

Frågor som uppstår:

- Hur modellerar man? Vad tas med?
- Hur avgör man om en modell är relevant?
- Hur skiljer sig modellen från verkligheten?
- Hur översätts resultat från modellen till verkligheten?

Teoretisk datalogi

- De viktigaste begreppen är (förmodligen) algoritmer, datastrukturer och komplexitet.
- Teoretisk datalogi skapar metoder, logiker och semantiska modeller som hjälper oss att designa program, att resonera kring program, att visa korrekhet och effektivitet hos program och att designa nya programspråk.
- Det finns inte några "kritiska" experiment.
- Olika teorier bedöms efter de insikter de ger och deras praktiska tillämpning.

Experimentell DV

- Experimentell DV studerar informationsprocesser ungefär som fysik studerar mekaniska processer.
- Olika hypoteser testas mot mätdata.
- Experimentell DV används där deduktion kommer till korta.
- Experiment används t.ex. inom sökmetoder, automatisk teoremprovning, NP-fullständiga problem, spel och maskininlärning.

- Inom t.ex. Artificiella Neurala Nätverk har experimentella metoder gett bättre resultat än vid teoretiska metoder visat.
- Inom flera områden saknas dock experiment, t.ex. när det gäller att avgöra effektivitet hos funktionell programmering kontra imperativ programmering.
- Behövs det mer experiment?

Datorsimulering

- Datorsimulering är ett redskap inom många olika vetenskaper. Simuleringen berörs därför av de vetenskapliga problem och metoder som finns inom de vetenskaperna.
- Simuleringsmetoder sträcker ut det möjliga området för experiment (oändlig tid t.ex.).
- Några framgångsrika områden är: Kaos och komplexa system, virtuell verklighet, artificiellt liv och fysiska system med datoranimering.

Filosofiska problem inom datalogi

Om vi fokuserar mer på teoretisk datalogi och deduktiva metoder. Det finns fem områden värda att nämna:

- Existens och identitet
- Semantik
- Bevis och program
- Beräkningsbarhet
- Programspråk

Existens och identitet för program

- Det finns en dual natur hos program som dyker upp på olika sätt.
- Program som text kontra implementation
- Program kontra algoritm
- Program kontra specifikation (samband indata /utdata)

Semantik

Semantik avser att förklara meningen hos program och inte bara syntaxen. Vissa problem analyseras:

- Samband mellan operationell och denotationell semantik.
- Samband mellan denotationell semantik och implementation.
- Vad är identitet och ekvivalens mellan program? Hur uttrycks de med semantik?

Bevis och program

Vilken status har egentligen korrekthetsbevis för program? Olika typer av kritiska frågor finns:

- Är inte korrekthetsbevis ointressanta i sig? Jämför med matematiska bevis.
- Bevis är längre än programmen själva. Strider mot matematiska estetiska krav?
- Matematiska bevis använder nya konceptuella modeller. Det gör inte korrekthetsbevis?
- Korrekthetsbevis ger korrekthet på *textnivå*. Ger de korrekthet på implementationsnivå?

Beräkningsbarhet

- Church-Turings tes innebär att allt som kan beräknas kan beräknas av en Turing-maskin.
- Det finns olika tolkningar av vad detta innebär. Vad menar man med en beräkning?
- Man kan mena en mekanisk process som kan uttryckas matematiskt
- Eller man kan mena något som utförs av en fysisk maskin.
- Täcker CT-tesen verkligen in all typ av beräkning.
- Frågan verkar leda till sambandet mellan matematiska, abstrakta objekt och den fysiska "verkligheten".

Programspråk

Analys av programspråk leder till flera svårigheter som har samband med matematik och vanliga språk.

- Vad är egentligen *abstraktion*? Är det ett annat begrepp i DV än t.ex. i matematik. En åsikt är att det i matematik betyder att vi bortser från information medan vi i DV menar att vi gömmer information.
- Vi har typer i DV. Vad är sambandet med typer som i Freges och Russells filosofier?
- Liknande mycket svåra frågor finns.

*Finns det verkligen genuina
vetenskapliga problem inom
datavetenskap?*

*Finns det problem som inte finns
inom andra områden?*

Hur är det med AI och medvetandeproblemet?

Några viktiga händelser och personer i datavenskapsens historia

Före 1900-talet

- 1600-talet: Pascal och Leibniz skapar räknemaskiner.
- 1700-talet: Jacquard skapar en programmerbar vävstol.
- 1800-talet: Babbage konstruerar den första "datorn".
- Hollerith skapar hålkortet.

1900-1930

- Hilbert sätter upp en plan som går ut på att mekanisera bevis. Han vill hitta en mekanisk process som producerar bevis.
- Gödel visar att det inte finns någon generell sådan metod som alltid fungerar.
- Turing behandlar Hilberts avgörbarhetsproblem genom att definiera Turing-maskinen och visa att det finns problem den inte kan lösa.

1940-talet

- De första riktiga datorerna skapas under 2:a världskriget.
- von Neumann publicerar sin modell av dator. Skapar också algoritmer som Mergesort.
- Transistorn uppträffas 1947.

1950-talet

- De första kompilatorerna konstrueras.
- FORTRAN skapas 1957.
- LISP skapas 1958.
- Den integrerade kretsen skapas.
- Dijkstra presenterar sin algoritm 1956.

1960-talet

- Datavetenskap uppstår som akademiskt ämne 1962.
- Arpanet, en föregångare till internet, börjar konstrueras.
- BASIC utvecklas 1964.
- Forskning kring automatteori och formella språk. Chomsky presenterar sina lingvistiska teorier. (Kontextfri grammatik.)
- De första korrekthetsbevisen för program.
- Knuth skapar den rigorösa algoritmanalysen.

1970-talet

- Unix skapas.
- Den första superdatorn.
- Komplexitetsteorin grundas 1971 med de första resultaten om NP-fullständiga problemen.
- Datorer börjar säljas till privatpersoner.

1980-talet

- Den första PC:n 1981.
- Apple-datorer.
- Nätverk. LAN.
- Internet skapas.
- E-mail.

1990-talet

- Paralleldatorer.
- Teorin för kvantdatorer.
- Nanoteknologi.
- Användningsområdena för datorer utvecklas i och med att antalet användare ökar.