Computability and Logic
dBerLog

Q1 2007

Bachelor Information Meeting

- Department of Computer Science offers an meeting for students briefing on the bachelor year and the master's studies
- Time: 15-17, November 1, 2007
- Place: Store Auditorium

dBerLog - final lecture!

- Summary
  - Universality
  - Duality
  - Self-reference
  - Program Verification
- Life stories
- About the exam

Universal Computational Models

Turing: Turing machines (1930s)
Goedel: recursive functions (1930s)
Church: λ-calculus (1930s)
Chomsky: Language grammars (1950s)

All these have been shown to be "equivalent" wrt expressiveness!

Church's λ-calculus

- $(\lambda x.\ E)\ (F) \xrightarrow{\beta} E\ [F/x]$
- $(\lambda x.\ E) \xrightarrow{\alpha} \lambda y.\ (E\ [y/x])$
- **Theorem**
  A (partial) function is computable in the λ-calculus iff it is computable by a Turing machine!

Chomsky type 0 grammars

- Context dependent rules:
  $\alpha\ A\ \beta \rightarrow \alpha\ \gamma\ \beta$ where $A \in V$ and $\alpha, \beta, \gamma \in (V \cup T)^*$
- **Theorem**
  The class of languages generated by Chomsky type 0 grammars is exactly the class of (Turing) recursively enumerable languages!

Gödels μ-recursive functions $N^k \rightarrow N$

- Successor, zero-test, projections, function composition, and primitive recursion:
  $f(0, x) = h(x)$
  $f(n+1, x) = g(n, x, f(n, x))$
  unbounded minimization:
  $f(n) = \mathit{min}\ y.\ (g(y,n) = 0)$
- **Theorem**
  A (partial) function is definable as a μ-recusive function iff it is computable by a Turing machine.

dBerLog - final lecture!

- Summary
  - Universality
  - Duality
  - Self-reference
  - Program Verification
- Life stories
- About the exam

Hækleopskrift: en dug...

Opskrift/program:

Hækl 21 lm

1. række: * 1 stm, 1 lm, spring over næste lm *; Gentag fra * til * 9 gange til; 1 stm i sidste lm (10 mlmrum), vend

2.-10. række: * 1 stm på stm, 1 lm, spring over mlmrum * ; Gentag fra * til * 9 gange til;

.....

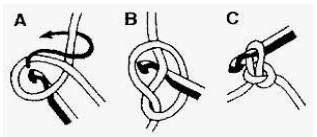
Materialer/data:

Perlebomuld # 5

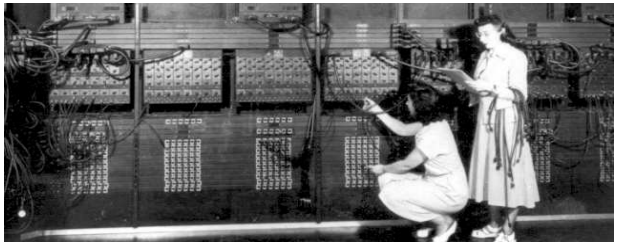
Hæklenål 1,9 mm

...og en hækler

En luftmaske



Programming ENIAC 1943-45



Duality between programs and data

John von Neumann
1903-1957

First draft on a report on EDVAC 1945

Duality between programs and data

Alan Turing
1912-1954

On Computable Numbers with an application to the Entscheidungsproblem
1936

Bootstrapping

Duality

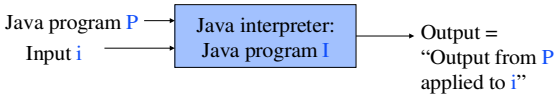
Rene Magritte (1966)

dBerLog - final lecture!

- Summary
  - Universality
  - Duality
  - Self-reference
  - Program Verification
- Life stories
- About the exam

Programs applied to programs

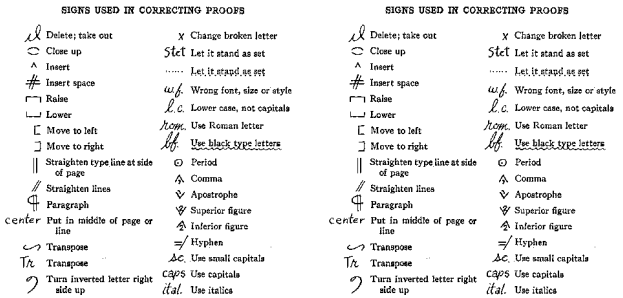
Happens ALL the time! Example: a Java interpreter



Text/program manipulation - examples

- Encryption programs
- Program optimizers
- Indentation programs

Proof reading


SIGNS USED IN CORRECTING PROOFS

Delete; take out	Change broken letter
Close up	Let it stand as set
Insert	Let it stand as set
Insert space	Wrong font, size or style
Raise	Lower case, not capitals
Lower	Use Roman letter
Move to left	Use black type letters
Move to right	Period
Straighten type line at side of page	Comma
Straighten lines	Apostrophe
Paragraph	Superior figure
Put in middle of page or line	Inferior figure
Transpose	Hyphen
Transpose	Use small capitals
Turn inverted letter right side up	Use capitals
	Use italics

SIGNS USED IN CORRECTING PROOFS

Delete; take out	Change broken letter
Close up	Let it stand as set
Insert	Let it stand as set
Insert space	Wrong font, size or style
Raise	Lower case, not capitals
Lower	Use Roman letter
Move to left	Use black type letters
Move to right	Period
Straighten type line at side of page	Comma
Straighten lines	Apostrophe
Paragraph	Superior figure
Put in middle of page or line	Inferior figure
Transpose	Hyphen
Transpose	Use small capitals
Turn inverted letter right side up	Use capitals
	Use italics

Self-reference

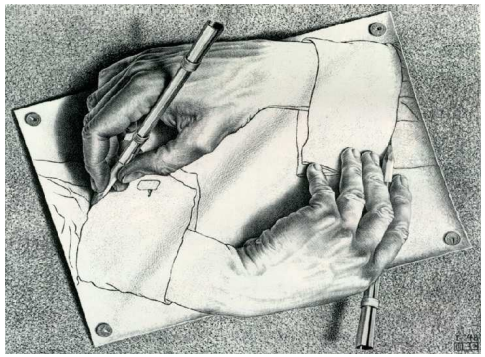
All books in our library have a list of references

Some books reference themselves ("see chapter..")
Let us call such a book self-referencing

Task:

write a book (for the library) containing a list of all the books (in the library), which are not self-referencing!

Selfreference - Escher 1898-1972



Selfrerence - dBerLog

dBerLog - final lecture!

- Summary
  - Universality
  - Duality
  - Self-reference
  - Program Verification
- Life stories
- About the exam

Program correctness

Algoritme:	Euklid (m, n)
Inputbetingelse:	$m, n \geq 1$
Outputkrav:	$r = sfd(m, n)$
Metode:	$\{m, n \geq 1\}$
	$p \leftarrow m;\ q \leftarrow n;$
	$I = \{sfd(p, q) = sfd(m, n)\}$ while $p \neq q$ do
	if $p > q$ then $p \leftarrow p - q$
	else $q \leftarrow q - p;$
	$r \leftarrow p$
	$\{r = sfd(m, n)\}$

Program incorrectness

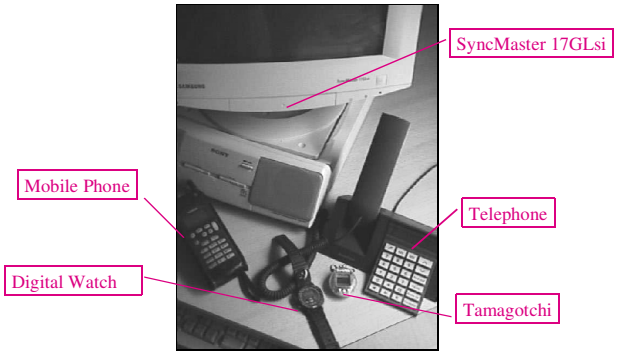
Algoritme: Euklid (m, n)
Inputbetingelse: $m, n \geq 1$
Outputkrav: $r = sfd(m, n)$
Metode: $\{m, n \geq 1\}$

$p \leftarrow m;\ q \leftarrow n;$
$I = \{sfd(p, q) = sfd(m, n)\}$ while $p \neq q$ do
    if $p > q$ then $p \leftarrow p - q$
        else $q \leftarrow q - p;$
$r \leftarrow q$
$\{r = sfd(m, n)\}$

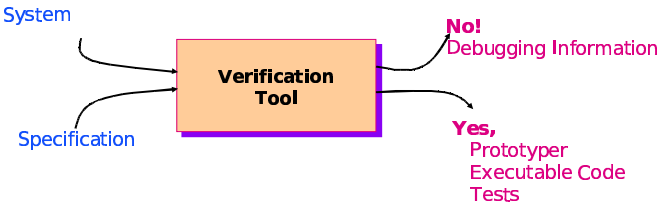
Program termination

$\{x > 1\}$
while $x \neq 1$ do
    if even (x) then x := x div 2 else x:= $3 \times x + 1$
$\{true\}$

Embedded Systems



Automated Verification

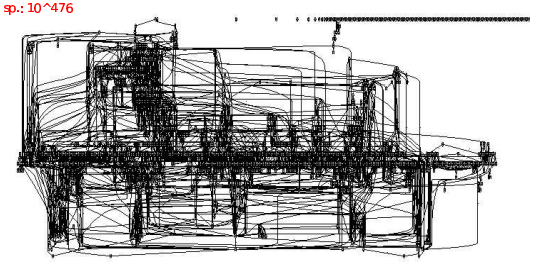


A REAL system

Train Simulator

1421 machines
11102 transitions
2981 inputs
2667 outputs
3204 local states
Declare state sp.: 10^476

BUGS ?

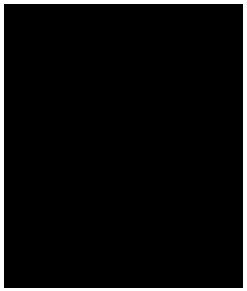


dBerLog - final lecture!

- Summary
  - Universality
  - Duality
  - Self-reference
  - Program Verification
- Life stories
- About the exam

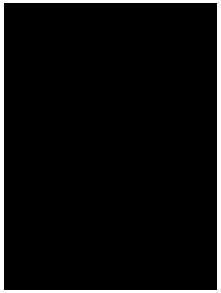
Life stories - A. Turing 1912-1956

Life stories - K. Gödel 1906-1978

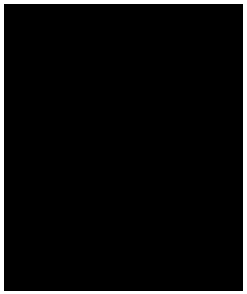


Ud og se med DSB - september 2007

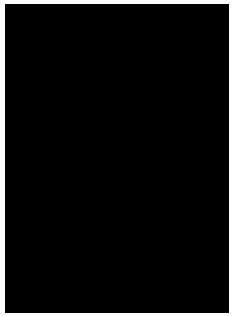
Life stories - E. Post 1897 - 1958



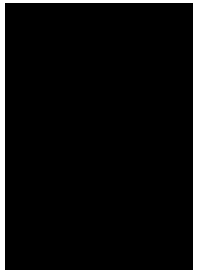
Life stories - A. Church 1903 - 1995



Life stories - N. Chomsky 1928 -



Life stories - T. Hoare 1934 -



Life stories - J. C. Martin ?? -

Life stories - J. Kelly ?? - 1995

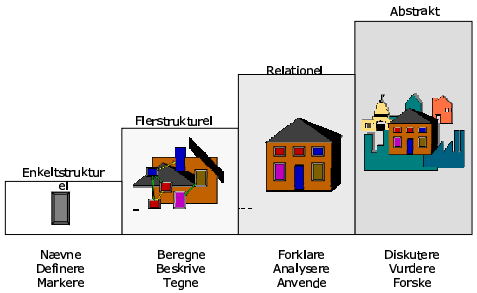
dBerLog - final lecture!

- Summary
  - Universality
  - Diagonalization
  - Self-reference
  - Program Verification
- Life stories
- About the exam

dBerLog - Exam

- **Oral exam**
  - Friday October 19 - Wednesday October 24 2007
  - Internal examiners:
    - Anders Møller, Michael Schwartzbach, Ole Østerby
  - Two questions:
    - Computability, Logic
  - 20 minutes each - no preparation time
- **Compulsory home works**
  - 2 compulsory written home work assignments must all be handed in and accepted by the tutor
  - OBS STRICT DEADLINE TOMRROW!!!!!!

SOLO taxonomi



dBerLog - Goals

- The goals of this course are to give the student the following capabilities
  - to be *familiar* with the basic *terminology* for computability and logic
  - to *describe* basic computability classes and fundamental logics
  - to *describe* basic *properties* of computability classes and logics
  - to *explain* constructive/algorithmic approaches to computability classes and logics
  - to *analyse* and to *prove* properties of computability classes and logics

dBerLog - Goals Computability

- The goals of this course are to give the student the following capabilities
  - to *be familiar with* the basic *terminology* for computability
    - problems as formal languages and operations on these, decidability, Turing machines
  - to *describe* basic computability classes and their properties
    - recursive and recursively enumerable languages, closure and decidability properties, from intuition and examples to formal notation and definitions
  - to *explain* algorithmic approaches to properties of computability classes
    - constructive arguments for closure and decidability properties, problem reductions
  - to *analyse* and to *prove* properties properties of computability classes
    - diagonalization, reduction

dBerLog - Goals Logic

- The goals of this course are to give the student the following capabilities
  - to *be familiar with* the basic *terminology* for logic
    - truth, satisfaction, validity, syntax, semantics
  - to *describe* fundamental logics and their properties
    - propositional logic, truth tables, predicate logic, interpretations and valuations, program logics, proof systems
  - to *explain* algorithmic approaches to properties of logics
    - decidability, normal forms, proofsystems and their proofs, from examples to formal definitions
  - to *analyse* and to prove properties properties of logics
    - soundness and completeness, existence and non-existence of proof systems, Gödel's theorems

Plans for the 7 weeks

- Model of Computation: Turing Machines
- Computability
- Non computable problems

- Propositional Logic
- Predicate Logic
- Program Logic - Gödel's theorems

- Summary - Exam

dBerLog Curriculum

Martin:

chapter 9, chapters 10.1-10.2, 10.3, 10.5 (excl. proofs of Thms 10.8 and 10.9), chapter 11 (excl. proof of Thms 11.14 and 11.15)

Kelly:

chapter 1, chapter 4, chapter 6.1-6.7.4, 6.9-6.10, chapter 7

Nielsen:

Limitations of Program Verification, 2004

Turing machines

- *Definition and operations of TMs* - examples of TMs solving problems, computing functions
- *Variations of TMs*
- *Churh-Turing thesis*
- *The universal TM*

Recursive and recursively enumerable languages

- *Definitions of RE and R*
- *Closure properties* of RE and R
- *Characterizations of RE* - enumerating a language, Chomsky grammars,…
- *Countability arguments* for the existence of non-RE languages

Unsolvable problems

- Definition of *solvable problem*
- *The languages NSA and SA* - and their membership wrt. RE and R
- *The reduction technique*
- *Unsolvable problems for TMs*
- *Rice's theorem*
- *Other unsolvable problems*, - PCP and CFG problems

Logic - semantics

- Clear understanding of logical *syntax* and *semantics*, logical *truth*, *satisfaction*, *validity*, logical *connectives*, logical *consequence*, *models*
- Propositional logic: *truth tables*
- Predicate logic: *variables, objects, predicates, functions, quantifiers, scope, binding, substitution, - interpretations, valuations*

Logic proof systems

- General definition of *axiomatic proof system* and its theorems
- Proof systems *AL* for propositional logic and *FOPL* for predicate logic in particular
- *Soundness* and *completeness* of axiomatic proof systems
- *Proofs* of soundness and completeness of AL, *decidability* for propositional logic
- *Awareness* of soundness, completeness, decidability results for FOPL/predicate logic

Limitations of program verification

- *Hoare triples, partial and total correctness*
- *Hoare proof system,* - and its relation to proof system for the model of natural numbers
- *Incompleteness theorem for Hoare triples* - and its proof: provability rec. enum. - truth not rec. enum.
- *Goedels incompleteness theorem* - and its proof (see above)

dBerLog exam

- Please make sure all formalities are in place - registration, compulsory exercises, etc!
- Please show up for your exam well in advance of your scheduled time !
- And remember to enjoy the exam.....