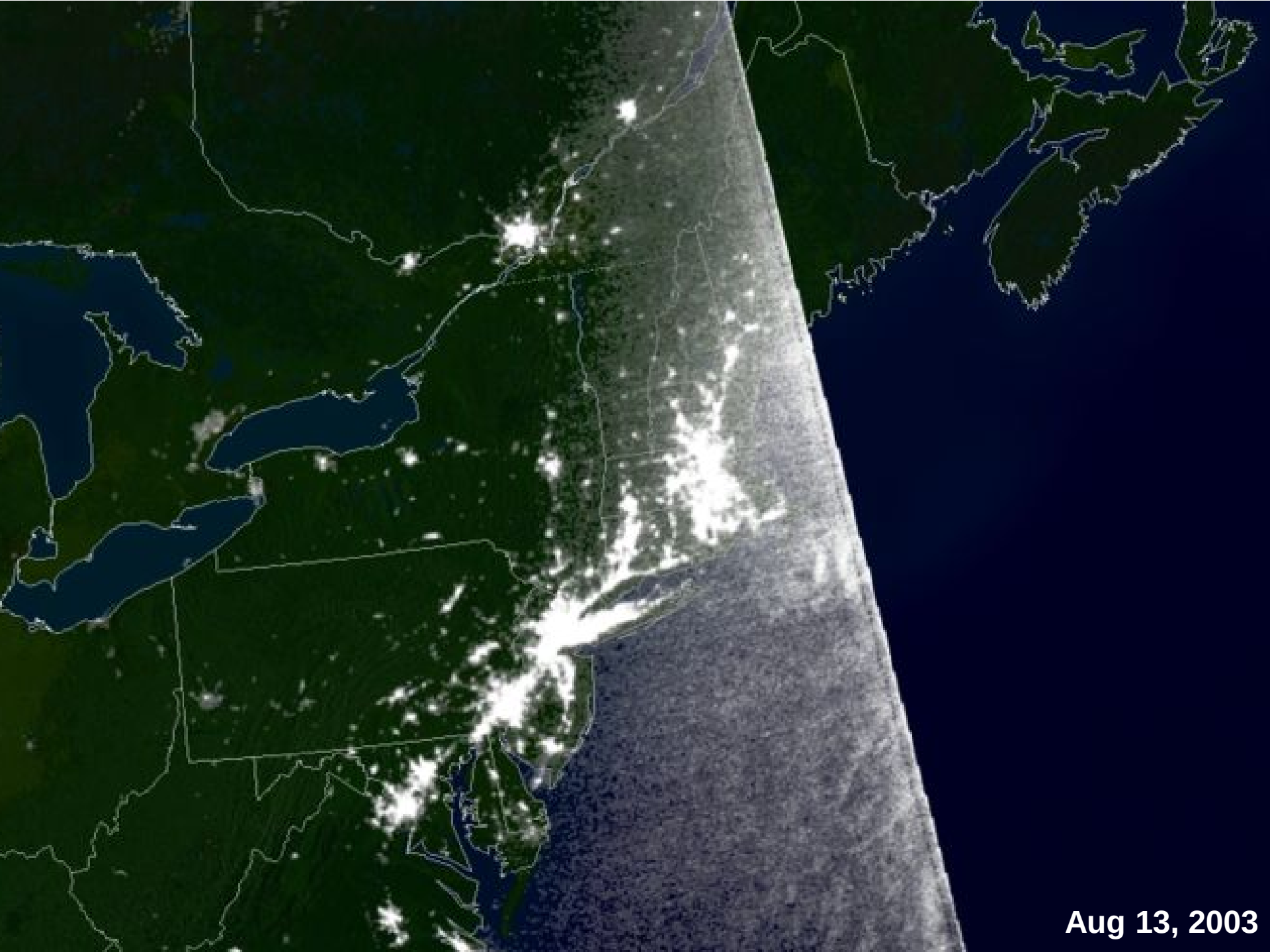
"Contrariwise," continued Tweedledee, "if it was so, it might be; and if it were so, it would be; but as it isn't, it ain't. *That's logic.*"

Lewis Carroll
Charles Lutwidge Dodgson, Oxford


Aug 13, 2003


Aug 14, 2003

Northeast Blackout of 2003

50 million people
60 billion US$

… because of a computer software bug in General Electric Energy's Unix-based XA/21 energy management system that prevented alarms from showing on their control system. This alarm system stalled because of a race condition bug.

Ontario
Aug 14, 2003

Limitations of Human Reasoning

Software design is teamwork over decades

Driven by business imperatives

First point of failure: Requirement Engineering

Writing the wrong program

Limitations of Human Reasoning

Software design is teamwork over decades

Driven by business imperatives

First point of failure: Requirement Engineering

Writing the wrong program

Second point of failure: Software Errors

Writing the program wrongly

Reason: Mathematical Complexity

Limitations of Human Reasoning

Program Sizes	6.000	Water Pump
	500.000	Pace Maker
	2.000.000	BMW 745i
	7.000.000	Boeing 777
	50.000.000	MS Vista / Linux

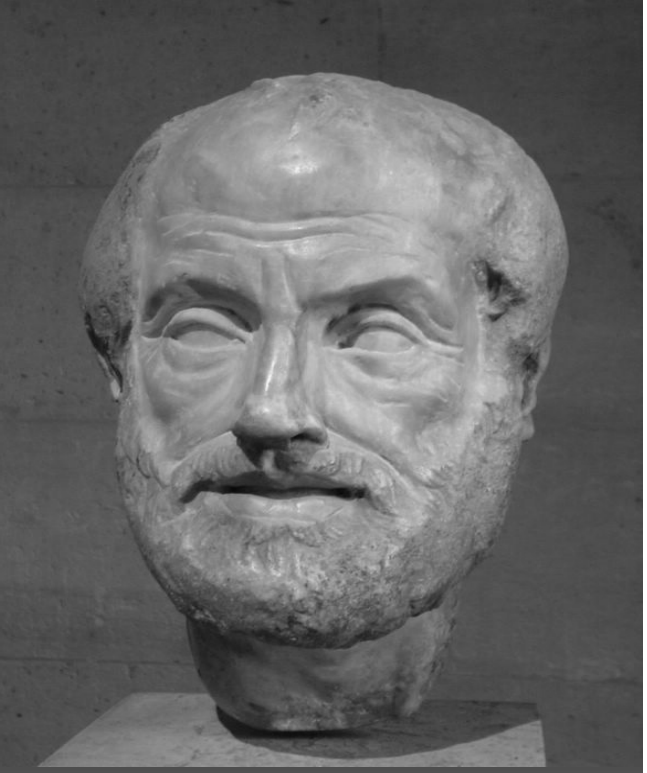
Essentially long mathematical formulas

Information content approaching human DNA

Errors per 10.000 Lines	250 Errors	(typical software)
	20 Errors	(good software)
	1 Error	(Space Shuttle)

Managing Complexity

- Structured Thinking
  software engineering, programming languages
- Fault Tolerance
- Manpower
  Aviation: 40 program lines per week
- Testing
  Empirical evidence
- Mathematical Proof of Correctness
- Ultimate Goal:
  Computer-generated Mathematical Proof of Correctness



Program and System Verification

Logic Revisited

Helmut Veith

SCHÜLER An ihrem Hals will ich mit Freuden hangen;
Doch sagt mir nur, wie kann ich hingelangen?

MEPHISTOPHELES Erklärt Euch, eh Ihr weiter geht,
Was wählt Ihr für eine Fakultät?

SCHÜLER Ich wünschte recht gelehrt zu werden,
Und möchte gern, was auf der Erden
Und in dem Himmel ist, erfassen,
Die Wissenschaft und die Natur.

MEPHISTOPHELES Da seid Ihr auf der rechten Spur;
Doch müßt Ihr Euch nicht zerstreuen lassen.

SCHÜLER Ich bin dabei mit Seel und Leib;
Doch freilich würde mir behagen
Ein wenig Freiheit und Zeitvertreib
An schönen Sommerfeiertagen.

MEPHISTOPHELES Gebraucht der Zeit, sie geht so schnell von hinnen,
Doch Ordnung lehrt Euch Zeit gewinnen.
Mein teurer Freund, ich rat Euch drum
Zuerst Collegium Logicum.
Da wird der Geist Euch wohl dressiert,
In spanische Stiefeln eingeschnürt,
Daß er bedächtiger so fortan
Hinschleiche die Gedankenbahn,
Und nicht etwa, die Kreuz und Quer,
Irrlichteliere hin und her.

What is Logic ?

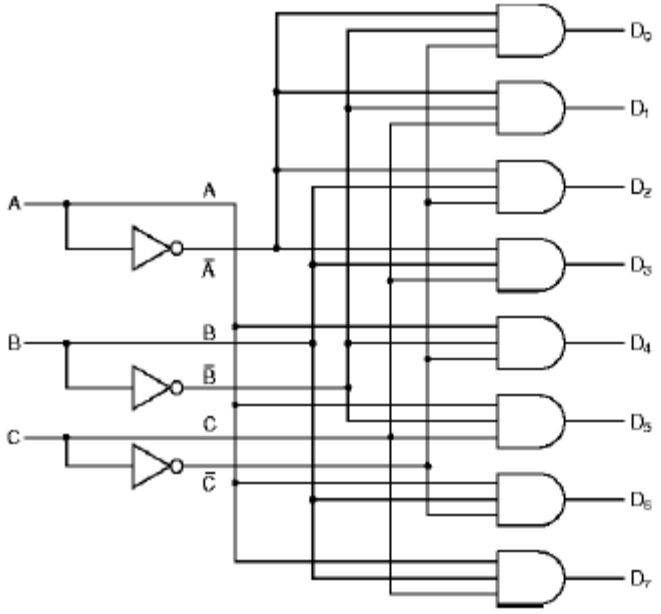
Logic differs from rhetoric in that logic handles reason exact and in truth, and rhetoric handles it as it is planted in popular opinions and matters.

Francis Bacon

Wissenschaft von den notwendigen Gesetzen des Verstandes und der Vernunft überhaupt oder – welches einerlei ist – von der blossen Form des Denkens überhaupt.
… Wissenschaft der Verstandesregeln überhaupt

Immanuel Kant

What is Logic ?



n data inputs

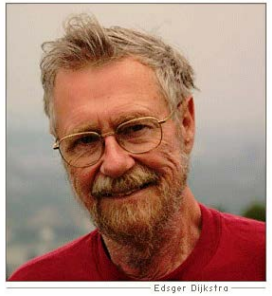
2^n data outputs

3-to-8 decoder

Logic and Computer Science

Alan Turing

> “I expect that digital computing machines will eventually stimulate a considerable interest in symbolic logic … The language in which one communicates with theses machines … forms a sort of symbolic logic.”



Dijskstra: *Computer Science* =

VLSAL (Very Large Scale Applications of Logic)

The unusual effectiveness of logic in computer science

1960 E.P. Wigner

On the unreasonable effectiveness of mathematics in the natural sciences.

Halpern et al 2001:

"As a matter of fact, logic has turned out to be significantly more effective in computer science than it has been in mathematics."

Logic = Calculus of Computer Science

- Verification
- Data bases
- Complexity
- Artificial Intelligence
- Programing Languages
- Software Engineering
- Computer Security

Boolean Functions

Helmut Veith

Boolean Functions

Truth	1
Falsity	0
Boolean domain	B = {0,1}
Boolean function	$f:\{0,1\}^n \rightarrow \{0,1\}$

Two Boolean functions f, g are identical if

$\forall\ x_1 \ldots x_n\ .\ f(x_1 \ldots x_n) = g(x_1 \ldots x_n)$

Number of Boolean Functions in n variables: 2^{2^n}

The 16 Boolean Functions in C

1. 0
2. x && y
3. y > x
4. y
5. x > y
6. x
7. x!=y
8. x || y
9. !(x||y)
10. x==y
11. !x
12. x <= y
13. !y
14. y<= x
15. !(x && y)
16. 1

The 16 Boolean Functions in C

1.	0		9.	!(x||y)	NOR
2.	x && y	$x \wedge y$, $x \times y$, max(x,y)	10.	x==y	$x \leftrightarrow y$, $x \equiv y$
3.	y > x		11.	!x	$\neg x$
4.	y		12.	x <= y	$x \rightarrow y$
5.	x > y		13.	!y	$\neg y$
6.	x		14.	y<= x	$y \rightarrow x$
7.	x!=y	$x \oplus y$	15.	!(x && y)	NAND
8.	x || y	$x \vee y$, max(x,y)	16.	1	

Some properties of Boolean operators

Double Negation

$\neg\neg\phi \quad \equiv \quad \phi$

De Morgan Laws

$\neg(\phi \wedge \psi) \equiv (\neg\, \phi \vee \neg\, \psi)$

Distributivity

$\psi \wedge (\phi \vee \delta) \equiv (\psi \wedge \phi) \vee (\psi \wedge \delta)$

$\psi \vee (\phi \wedge \delta) \equiv (\psi \vee \phi) \wedge (\psi \vee \delta)$

Contraposition

$\phi \rightarrow \psi \;\equiv\; \neg\, \psi \rightarrow \neg\, \phi$

Commutativity

$\phi \wedge \psi \equiv \psi \wedge \phi \quad \phi \vee \psi \equiv \psi \vee \phi$

Associativity

$(\phi \wedge \psi) \wedge \delta \equiv \psi \wedge (\phi \wedge \delta)$

$(\phi \vee \psi) \vee \delta \equiv \psi \vee (\phi \vee \delta)$

Commutativity and Associativitvy enable us to write

$\wedge_{1 \leq i \leq n} \phi_i \qquad \vee_{1 \leq i \leq n} \phi_i$

Properties of Boolean Functions

Satisfiable

$\exists\, x_1 \ldots x_n.\ f(x_1, \ldots, x_n) = 1$

Tautology

$\forall\, x_1 \ldots x_n.\ f(x_1, \ldots, x_n) = 1$

In this case we write $\models f$

Contradictory / Inconsistent

$\forall\, x_1 \ldots x_n.\ f(x_1, \ldots, x_n) = 0$

Monotone in x_i

$f(\ldots 0 \ldots) \leq f(\ldots 1 \ldots)$

Dependent on x_i

$f(\ldots 1 \ldots) \neq f(\ldots 0 \ldots)$

Var(f) = dependent variables of f

The 16 Boolean Functions in C

1.	0		9.	!(x||y)	NOR
2.	x && y	$x \wedge y$, $x \times y$, max(x,y)	10.	x==y	$x \leftrightarrow y$, $x \equiv y$
3.	y > x		11.	!x	$\neg x$
4.	y		12.	x <= y	$x \rightarrow y$
5.	x > y		13.	!y	$\neg y$
6.	x		14.	y<= x	$y \rightarrow x$
7.	x!=y	$x \oplus y$	15.	!(x && y)	NAND
8.	x || y	$x \vee y$, max(x,y)	16.	1	

Note: $\models f \equiv g$ IFF $f = g$

Similar for other operators

Data Structures for Boolean Functions

- Truth Tables
- Propositional Logic
- CNF, DNF, …
- Polynomials in GF(2)
- Boolean Circuits
- Quantified Boolean Formulas
- Binary Decision Trees
- Binary Decision Diagrams (BDDs)

Data Structures differ in

… their algorithms

… complexity

… tool support

… expressive power

QBF
Quantified Boolean Formulas

Extension of propositional logic by
propositional quantifiers $\exists$ p and $\forall$ p

Macro Expansion:

$\forall$ p. (q $\vee$ p $\rightarrow$ r)

is short for

(q $\vee$ 0 $\rightarrow$ r) $\wedge$ (q $\vee$ 1 $\rightarrow$ r)

Analogous
$\exists$ p. ϕ(p) is short for ϕ(0) $\vee$ ϕ(1)

- $\exists$ computes the maximum
- $\forall$ computes the minimum

Functional Completeness

A boolean data structure is functionally complete, if all Boolean functions can be expressed in this data structure.

All data structures mentioned above are functionally complete.

Proof Idea: express if-then-else and constants

Which Boolean operators result in a functionally complete logic?

Canonicity of Boolean Data Structures

Canonicity:

Data structures for f and g are identical

iff

f and g are identical

Examples: Truth tables, binary decision diagrams, CNF…

Size of Boolean Data Structures

Strings of size n: $|\Sigma|^n$

Number of Boolean functions: 2^{2^n}

→ There is no data structure with good average compression

Compression in practice:

QBF (good compression)
→ Circuits → prop. Logic → BDDs → Binary Decision Trees →
Truth Tables (bad compression)

Complexity of Boolean Data Structures

	Satisfiability	Validity	Compression
QBF	PSPACE	PSPACE	good
Boolean Circuits	NP	coNP	medium
Propositional Logic	NP	coNP	medium
Binary Decision Diagrams	Constant	constant	medium
Truth Tables	Linear	Linear	bad

Craig Interpolation

Important consequence of functional completeness

For all functions ϕ, ψ where $\models \phi \rightarrow \psi$

there existis an *interpolant* δ such that

1. $\models \phi \rightarrow \delta$
2. $\models \delta \rightarrow \psi$
3. $var(\delta) \subseteq var(\phi) \cap var(\psi)$

Proof of Interpolation

Write ϕ and ψ as Boolean functions
$\phi(u_1, \ldots, u_n, d_1, \ldots, d_m)$ und $\psi(d_1, \ldots, d_m, v_1, \ldots, v_l)$
Where the d_i are the joint variables.

Then $\models \phi \rightarrow \psi$ means that
$\phi(u_1, \ldots, u_n, d_1, \ldots, d_m) \leq \psi(d_1, \ldots, d_m, v_1, \ldots, v_l)$
for all values of u_i, v_i, d_i

The new Boolean function
$\phi'(d_1, \ldots, d_m) := \max\{ \phi(const_1, \ldots, const_n, d_1, \ldots, d_m) \mid const_1, \ldots, const_n \in \{0,1\} \}$
Is there an interpolant

By Boolean completeness, ϕ' is expressible by the data data structure

ϕ' is called left interpolant

equivalent construction of right interpolants using min instead of max

Localisation of Inconsistencies by Interpolation

If ϕ, ψ are inconsistent then $\models \phi \rightarrow \neg\, \psi$

By interpolation we obtain a function δ, such that

$\models \phi \rightarrow \delta$ and

$\models \delta \rightarrow \neg\, \psi$

Since δ depends only on joint variables, it explains the reason for the inconsistency

Interpolation and QBF

Let $\models \phi \rightarrow \psi$

where

$\phi(u_1, \ldots, u_n, d_1, \ldots, d_m)$ und $\psi(d_1, \ldots, d_m, v_1, \ldots, v_l)$

Left interpolant

$\lambda = \exists\, u_1 \ldots \exists\, u_n\, \phi$

Right interpolant

$\rho = \forall\, v_1 \ldots \forall\, v_\lambda$

Then we have

$$\phi \rightarrow \lambda \rightarrow \rho \rightarrow \psi$$