Logic and Computation – CS 2800
Fall 2019

Lecture 10

Propositional logic

Stavros Tripakis

Northeastern University
Khoury College of
Computer Sciences

Outline

- Logic: a brief history
- Propositional logic
- Truth tables
- Satisfiability and validity

Logic

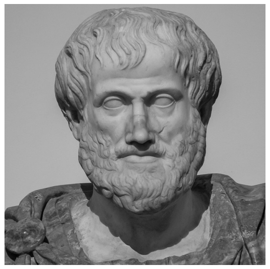
Logic

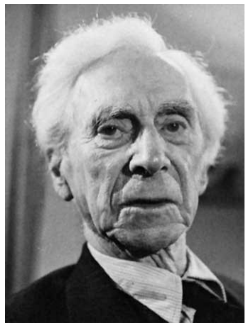
From Old French *logike*, from Latin *logica*, from Ancient Greek *λογική* (logikḗ, “logic”), from feminine of *λογικός* (logikós, “of or pertaining to speech or reason or reasoning, rational, reasonable”), from *λόγος* (lógos, “speech, reason”).

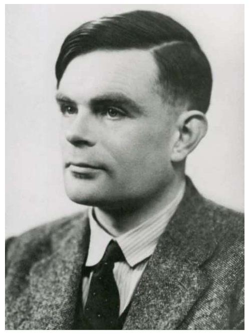
- What is logic?
  - A mathematical **language**: precise, unambiguous
  - A set of reasoning tools: deductions, proofs, …
- Why is logic important?
  - Foundation of mathematics
  - Foundation of computer science
  - Foundation of language? reason? intelligence?

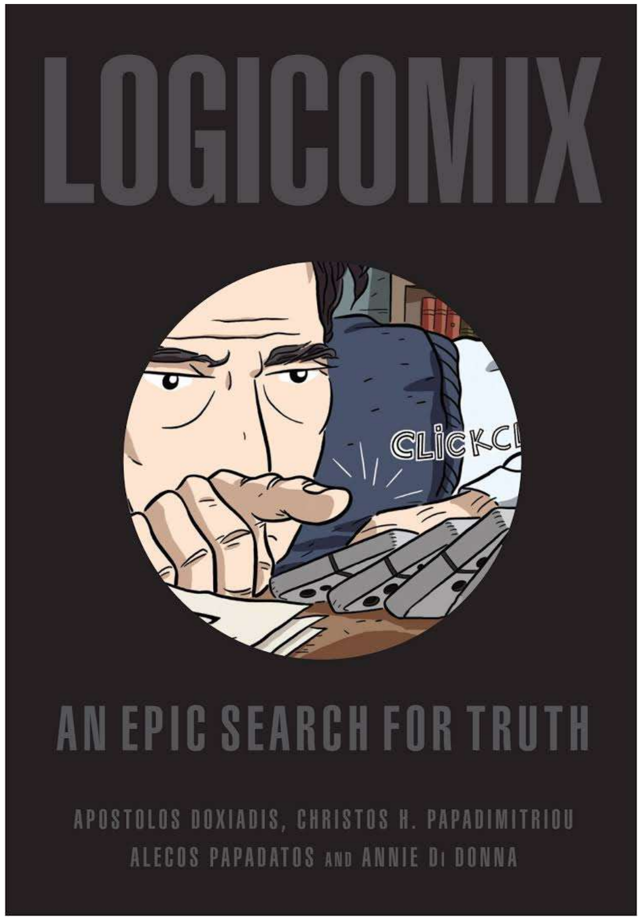
Logic – a brief history

- Very old:
  - Aristotle (384-322 BCE): syllogistic logic
  - His "Organon" books were the foundations of logic for 2000 years
  - Kant 1787: *"logic ..., since Aristotle, has been unable to advance a step and, thus, to all appearance has reached its completion."*
  - Logic historian Karl von Prantl (1820-1888) claimed that any logician who said anything new about logic was "confused, stupid or perverse."
- Very new:
  - Russell's paradox (1901)
  - Zermelo-Fraenkel set theory (1908-1920s)
  - Type theory (1908 – today)
  - Gödel's incompleteness theorem (1930)
  - Turing machines (1936)

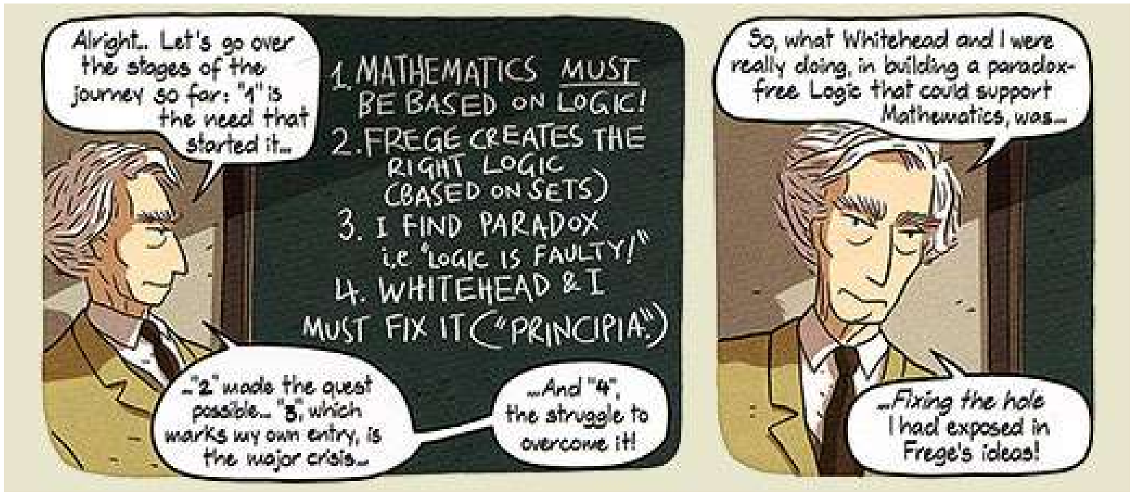






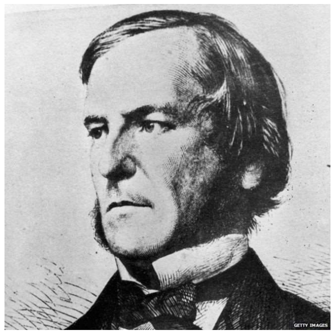

LOGICOMIX
CLICKCL
AN EPIC SEARCH FOR TRUTH
APOSTOLOS DOXIADIS, CHRISTOS H. PAPADIMITRIOU
ALECOS PAPADATOS AND ANNIE DI DONNA

ABOUT WHICH, WE CAN WELL ASK...
DOES IT CONT—
!?
WAIT A MINUTE!


Alright... Let's go over the stages of the journey so far: "1" is the need that started it...
1. MATHEMATICS MUST BE BASED ON LOGIC!
2. FREGE CREATES THE RIGHT LOGIC (BASED ON SETS)
3. I FIND PARADOX i.e "LOGIC IS FAULTY!"
4. WHITEHEAD & I MUST FIX IT ("PRINCIPIA")
..."2" made the quest possible... "3", which marks my own entry, is the major crisis...
...And "4", the struggle to overcome it!
So, what Whitehead and I were really doing, in building a paradox-free Logic that could support Mathematics, was...
...Fixing the hole I had exposed in Frege's ideas!

Propositional logic

Propositional logic = Boolean logic



- George Boole (1815-1864)
  - Boolean algebra (1847)
- What's the simplest possible arithmetic?
  - Empty set? Set of one element? They are trivial.
  - Booleans = set of two elements = {true, false} = {T,F} = {0,1} = {$\top$, $\bot$} = { t, nil} = …
  - Now our arithmetic becomes more interesting
  - There are many functions from Booleans to Booleans
- Rich domain, impressive list of applications
  - Logic, circuit design, SAT, verification, scheduling, AI, game theory, reliability, security, …

Boolean expressions

- The expressions of propositional logic
  - Recall: logic = language = set of expressions
  - To have a precise, unambiguous language, we must first define what are the syntactically valid expressions

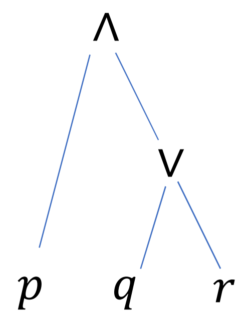
- Atomic expressions:
  - The constants *true* and *false* (or 0 and 1, or t and nil, …)
  - Propositional atoms or variables: p, q, r, …, or a, b, c, …, or x, y, z, …

- Composite expressions:
  - If ϕ and ψ are Boolean expressions, then we can combine them using the Boolean/propositional logic operators:

○ Negation:	$\neg\phi$	(“not ϕ”)	
○ Conjunction:	$\phi \land \psi$	(“ϕ and ψ”)	
○ Disjunction :	$\phi \lor \psi$	(“ϕ or ψ”)	
○ Implication :	$\phi \to \psi$	(“ϕ implies ψ”),	also written $\phi \Rightarrow \psi$
○ Equivalence :	$\phi \equiv \psi$	(“ϕ iff ψ”),	also written $\phi \leftrightarrow \psi$
○ Exclusive or (xor) :	$\phi \oplus \psi$	(“ϕ xor ψ”)	

Boolean expressions

- Boolean expressions are really **trees**:



- To avoid ambiguities, use parentheses
  - e.g., $p \wedge (q \vee r)$

Quiz

- Consider the expressions:
  1. $\land p \land q$
  2. $\land p \land q \land$
  3. $p \land q \land$
  4. $p \land q$

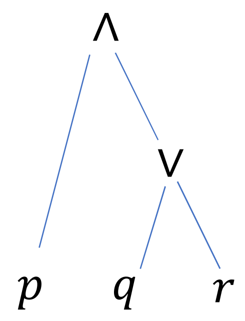
A. All of them are syntactically valid
B. None of them are syntactically valid
C. Two are valid, two are invalid
D. Only one is valid

Quiz

- Consider the expression: $\neg\neg p \wedge q$

  A. It is syntactically valid and unambiguous
  B. It is syntactically valid but ambiguous
  C. It is not syntactically valid

Parentheses



- Boolean expressions are really **trees**:

- To avoid ambiguities, use parentheses
  - e.g., $p \wedge (q \vee r)$

- When there is no ambiguity, don't use parentheses!
  - e.g., $p \wedge q \wedge r$ – why?

Precedence rules

- Sometimes we end up with too many parentheses
  - e.g., $\big((p \vee (\neg q)) \rightarrow r\big) \oplus ((\neg r) \rightarrow \big(q \wedge (\neg p)\big))$
- To avoid having to write too many parentheses, we establish precedence rules:
  - Negation $\neg$ binds strongest
  - Followed by conjunction and disjunction, $\wedge$,$\vee$
  - Followed by implication $\rightarrow$
  - Followed by iff and xor, $\equiv$,$\oplus$
- So we can rewrite the above equivalently as
  - $p \vee \neg q \rightarrow r \oplus \neg r \rightarrow q \wedge \neg p$

But: to me this is too confusing…
Often hard to remember the rules.

Use parentheses reasonably.
You will never be penalized for using "too many" parentheses
(unless we explicitly ask you to remove parentheses, as an exercise).

Quiz

- Precedence rules:
  - Negation $\neg$ binds strongest
  - Followed by conjunction and disjunction, $\wedge, \vee$
  - Followed by implication $\rightarrow$
  - Followed by iff and xor, $\equiv, \oplus$

- Is this ambiguous? $p \wedge q \vee r$
  A. Yeah
  B. Nope

Syntax and semantics

- We have seen the syntax of propositional logic: Boolean expressions

- But what do these expressions **mean**?

- Propositional logic semantics:
  - **Boolean functions**
  - Can be represented as **truth tables**

Truth tables of basic Boolean operators

p	$\neg p$
T	F
F	T

p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

p	q	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F

p	q	$p \Rightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T

p	q	$p \equiv q$
T	T	T
T	F	F
F	T	F
F	F	T

p	q	$p \oplus q$
T	T	F
T	F	T
F	T	T
F	F	F

Can you think of a new (different) truth table for a new (different) binary operator? unary operator?

Quiz

- Consider a Boolean operator over *n* propositional variables, i.e., an *n*-ary operator. How many rows does this operator have in its truth table?
  A. n
  B. $2n$
  C. n^2
  D. n^n
  E. none of the above

English usage

- In English “or” often means “exclusive or”
  - E.g., “you can have ice cream or a cookie” (implied: but you can’t have both)
  - In logic, or means at least one (see truth table!)
  - If you want to have “either-or” (exclusive or) use xor

English usage

- True or false?
  1. “if pigs can fly, then I am the president of the USA”
  2. “if I am the president of the USA, then pigs can fly”
  3. “if I am faculty at Northeastern, then pigs can fly”

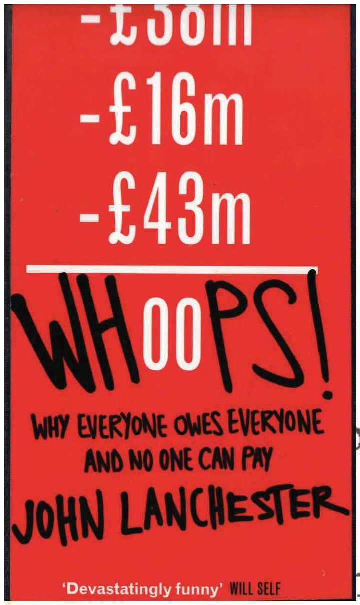
- Answers:
  1. True: pigs cannot fly
  2. True: I am not president of the USA
  3. False: I am faculty at Northeastern, and pigs can’t fly

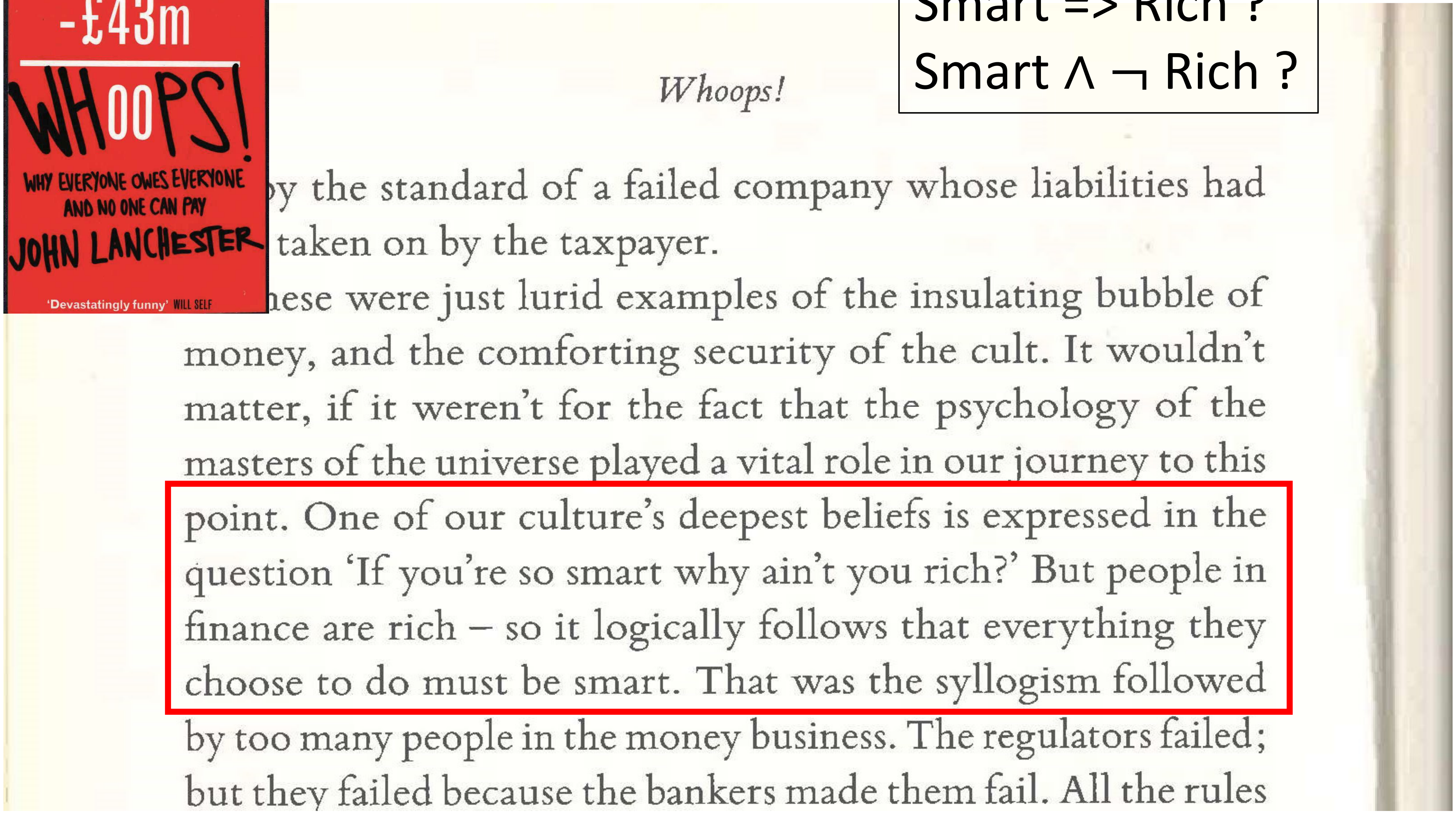
  — Truth table! (implication)
  — Some English speakers might say that the first two statements are false, since I am not the president / pigs can’t fly
  — Contrary to English, logic is precise => no debate

Rich => Smart ?

Smart => Rich ?

Smart ∧ ¬ Rich ?


-£16m
-£43m
WHOOPS!
WHY EVERYONE OWES EVERYONE AND NO ONE CAN PAY
JOHN LANCHESTER
'Devastatingly funny' WILL SELF



y the standard of a failed company whose liabilities had taken on by the taxpayer.

hese were just lurid examples of the insulating bubble of money, and the comforting security of the cult. It wouldn't matter, if it weren't for the fact that the psychology of the masters of the universe played a vital role in our journey to this point. One of our culture's deepest beliefs is expressed in the question 'If you're so smart why ain't you rich?' But people in finance are rich – so it logically follows that everything they choose to do must be smart. That was the syllogism followed by too many people in the money business. The regulators failed; but they failed because the bankers made them fail. All the rules

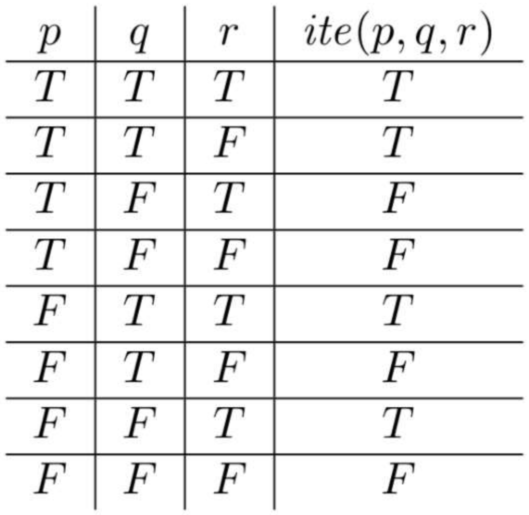
Implication

False => Anything!
False => False

- Cannot overemphasize its importance in logic
  - Make sure you understand its semantics
  - Truth table!

- True or false?
  - "if x is a natural number, then x>=0"
    - True
    - What about x=-1? Isn't it a counter-example since -1<0 ?
      - No, because -1 is NOT a natural number
    - The only way to make $p \rightarrow q$ false is to make p true and q false
    - Truth table!

A ternary operator: *ite*



p	q	r	$ite(p,q,r)$
T	T	T	T
T	T	F	T
T	F	T	F
T	F	F	F
F	T	T	T
F	T	F	F
F	F	T	T
F	F	F	F

Truth tables of Boolean formulas

- For every Boolean formula (= Boolean expression) we can construct its truth table
  - E.g., for the formula $\neg p \vee q \equiv p \Rightarrow q$ we get the truth table:

p	q	¬p	¬p ∨ q	p ⇒ q	¬p ∨ q ≡ p ⇒ q
T	T	F	T	T	T
T	F	F	F	F	T
F	T	T	T	T	T
F	F	T	T	T	T

Assignment: assigns truth values to the propositional variables of the formula

Subformula: a subexpression (subtree)

Satisfiability and validity

Satisfiability and validity

- A Boolean formula (=Boolean expression) is:
  - **Satisfiable**: when it is **sometimes true**
  - **Unsatisfiable**: when it is **never true**
  - **Valid**: when it is **always true**
  - **Falsifiable**: when it is **sometimes false**
- "Sometimes", "always", "never" refer to the truth table of the formula:
  - Sometimes: exists assignment to make the formula true
  - Never: no assignment makes the formula true
  - Always: all assignments make the formula true

Satisfiability and validity

- Every Boolean formula satisfies exactly two of the previous characterizations

- In particular, a Boolean formula is exactly one of the following:
  - Satisfiable and valid: always true
  - Satisfiable but not valid = satisfiable and falsifiable: sometimes true and sometimes false
  - Unsatisfiable and falsifiable: always false

- Think of it as a *lattice*:

$$\begin{tikzcd}
\text{valid} \arrow[d, dash] \\
\text{neither} \arrow[d, dash] \\
\text{unsatisfiable}
\end{tikzcd}$$

Examples

- Think of examples of the three categories

- Valid:

- Satisfiable and falsifiable:

- Unsatisfiable:

Next time

- Propositional logic continued
- Read Chapter 3, up to and including Section 3.6