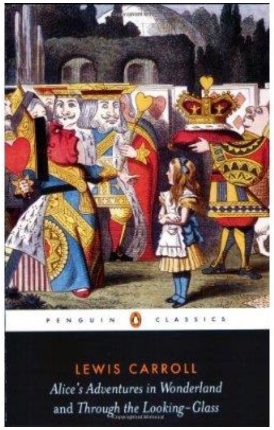
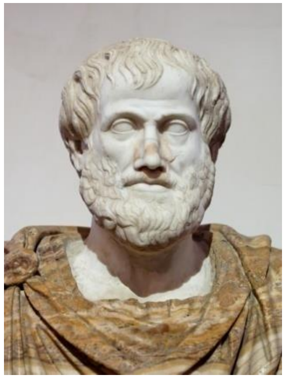


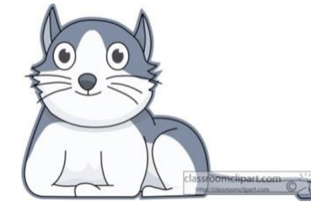


COMP 1002

Intro to Logic for Computer Scientists

Lecture 9


Meow-stery

- One evening there was a cat fight in a family consisting of a mother cat, a father cat, and their son and daughter kittens.
- One of these four cats attacked and bit another!
- One of the cats watched the fight.
- The other one hissed at the fighters.

- These are the things we know for sure:
  - 1. The watcher and the hisser were not of the same sex.
  - 2. The oldest cat and the watcher were not of the same sex.
  - 3. The youngest cat and the victim were not of the same sex.
  - 4. The hissing cat was older than the victim.
  - 5. The father was the oldest of the four.
  - 6. The attacker was not the youngest of the four.

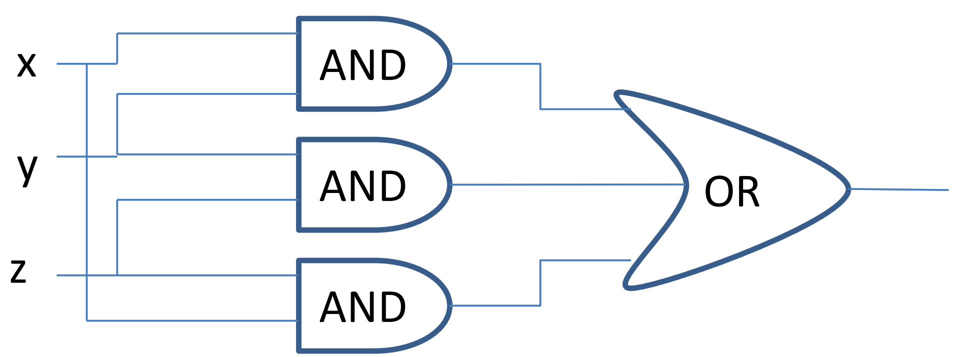
- Which nasty cat was the attacker?

Boolean functions and circuits

- What is the relation between propositional logic we studied and logic circuits?
  - View a formula as computing a function (called a Boolean function),
    - inputs are values of variables,
    - output is either *true (1)* or *false (0)*.
  - For example, $Majority(x, y, z) = true$ when at least two out of x, y, z are true, and false otherwise.
  - Such a function is fully described by a truth table of its formula (or its circuit: circuits have truth tables too).

Boolean functions and circuits

- What is the relation between propositional logic and logic circuits?
  – So both formulas and circuits "compute" Boolean functions – that is, truth tables.
  – In a circuit, can "reuse" a piece in several places, so a circuit can be smaller than a formula.
    - Still, most circuits are big!
  – $Majority(x, y, z)$ is $(x \wedge y) \vee (x \wedge z) \vee (y \wedge z)$



DNF formulas

- We talked about CNF formulas (Product of Sums), which are of the form AND of ORs of possibly negated variables (literals).
  - For example, $(x \vee \neg y) \wedge (\neg x \vee \neg z \vee w) \wedge (x)$

- A negation of a CNF is an OR of ANDs of literals. It is called a **DNF (disjunctive normal form),** or Sum of Products.
  - For example, $(\neg x \wedge y) \vee (x \wedge z \wedge \neg w) \vee (\neg x)$

Formulas for Boolean functions

- Suppose there is a Boolean function f such that $f(1,0,0) = 1$.
  - First, interpret 1 as True and 0 as False.
    - Say, A = True, B = False, C=False,
    - Then the formula F_f encoding function f above should be true on A=True, B=False, C=False.
  - We can now write a formula which is true only on this assignment:
    - $A \wedge \neg B \wedge \neg C$

Canonical DNF

- Now, to write a formula encoding the whole Boolean function f, write a formula encoding every assignment that makes f output 1.
  - Say $f(1{,}0{,}0) = 1$, $f(1{,}0{,}1) = 1$, and for any other input $f(x, y, z) = 0$
  - Then the corresponding assignments are $A \wedge \neg B \wedge \neg C$ and $A \wedge \neg B \wedge C$

- And finally take an OR of these formulas.
  - So the resulting formula would say "Either the formula is true because we are in the first scenario where f outputs 1, or the second, etc…"
  - $(A \wedge \neg B \wedge \neg C) \vee (A \wedge \neg B \wedge C)$

- This formula is called the **canonical DNF** of f
  - Every Boolean function f has a canonical DNF.

Canonical CNF

- Every truth table (Boolean function) can be also be written as a CNF:
  - Take every falsifying assignment $(f(x, y, z) = 0)$
    - Say, A = False, B = True, C=False.
  - Write it as a formula which is true only on this assignment:
    - $\neg A \wedge B \wedge \neg C$
  - To say that this assignment does not happen, write its negation:
    - $\neg(\neg A \wedge B \wedge \neg C) \equiv (A \vee \neg B \vee C)$
  - Take an AND of these for all falsifying assignments
    - It is equivalent to the original formula.
    - And it is a CNF! Called the **canonical CNF** of this formula.

Complete set of connectives

- CNFs only have ¬,∨,∧, yet any formula can be converted into a CNF
  - Any truth table can be coded as a CNF
- Call a set of connectives which can be used to express any formula **a complete set** of connectives.
  - In fact, ¬,∨ is already complete. So is ¬,∧ .
    - By DeMorgan, $(A \vee B) \equiv \neg(\neg A \wedge \neg B)$ No need for ∨!
  - But ∧,∨ is not: cannot do ¬ with just ∧,∨.
    - Because when both inputs have the same value, both ∧,∨ leave them unchanged.

Complete set of connectives

- How many connectives is enough?
  - Just one: NAND (NotAND), also called the Sheffer stroke, written as |
  - $\neg A \equiv A \mid A$
  - $A \vee B \equiv \neg(\neg A \wedge \neg B)$

    $\equiv (\neg A \mid \neg B)$

    $\equiv ((A \mid A) \mid (B \mid B))$
  - In practice, most often stick to $\wedge, \vee, \neg$

A	B	A | B
True	*True*	*False*
True	*False*	*True*
False	*True*	*True*
False	*False*	*True*

Puzzle 9

- Susan is 28 years old, single, outspoken, and very bright. She majored in philosophy. As a student she was deeply concerned with issues of discrimination and social justice and also participated in anti-nuke demonstrations.

  Please rank the following possibilities by how likely they are. List them from least likely to most likely. Susan is:

1. a kindergarden teacher
2. works in a bookstore and takes yoga classes
3. an active feminist
4. a psychiatric social worker
5. a member of an outdoors club
6. a bank teller
7. an insurance salesperson
8. a bank teller and an active feminist