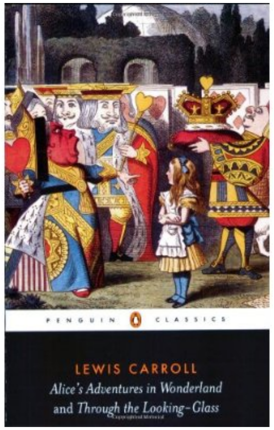
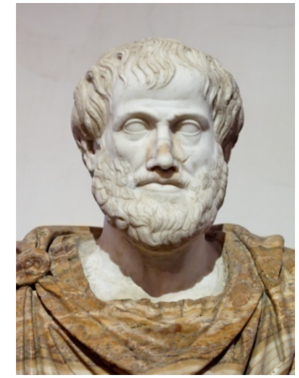
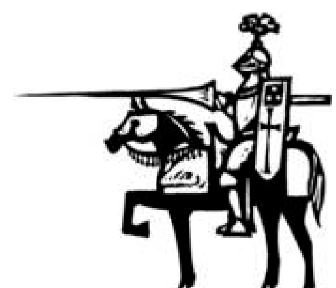




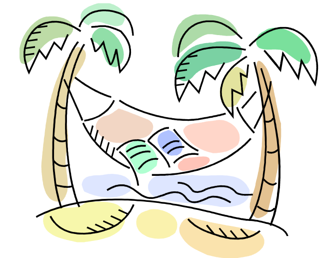
COMP 1002

Intro to Logic for Computer Scientists

Lecture 3




Knights and knaves



- On a mystical island, there are two kinds of people: knights and knaves. Knights always tell the truth. Knaves always lie.

- Puzzle 1: You meet two people on the island, Arnold and Bob. Arnold says “Either I am a knave, or Bob is a knight”. Is Arnold a knight or a knave? What about Bob?

Knights and knaves

- Puzzle 1: You meet two people on the island, Arnold and Bob. Arnold says "Either I am a knave, or Bob is a knight". Is Arnold a knight or a knave? What about Bob?
  - A: Arnold is a knight
  - B: Bob is a knight
  - Formula: $\neg A \vee B$: "Either Arnold is a knave, or Bob is a knight"
  - Want: scenarios where either both A is a knight and the formula is true, or A is a knave and the formula is false. Use "if and only if" notation: $(\neg A \vee B) \leftrightarrow A$. True if both formulas have same value

A	B	$\neg A$	$\neg A \vee B$	$(\neg A \vee B) \leftrightarrow A$
True	*True*	False	True	True
True	*False*	False	False	False
False	*True*	True	True	False
False	*False*	True	True	False

Special types of sentences

- A sentence that has a satisfying assignment is **satisfiable.**
  - Some row in the truth table ends with *True*.
  - Example: $B \rightarrow A$

A	B	$B \rightarrow A$
True	*True*	*True*
True	*False*	*True*
False	*True*	*False*
False	*False*	*True*

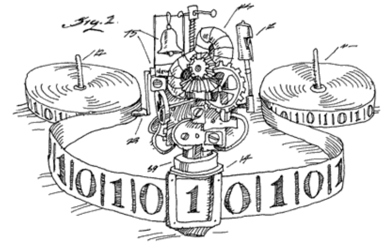
- Sentence is a **contradiction:**
  - All assignments are falsifying.
  - All rows end with *False.*
  - Example: $A \wedge \neg A$

A	$A \wedge \neg A$
True	*False*
False	*False*

- Sentence is a **tautology:**
  - All assignments are satisfying
  - All rows end with *True*.
  - Example: $B \rightarrow A \vee B$

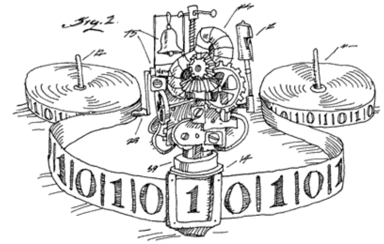
A	B	$A \vee B$	$B \rightarrow A \vee B$
True	*True*	*True*	*True*
True	*False*	*True*	*True*
False	*True*	*True*	*True*
False	*False*	*False*	*True*

Determining formula type

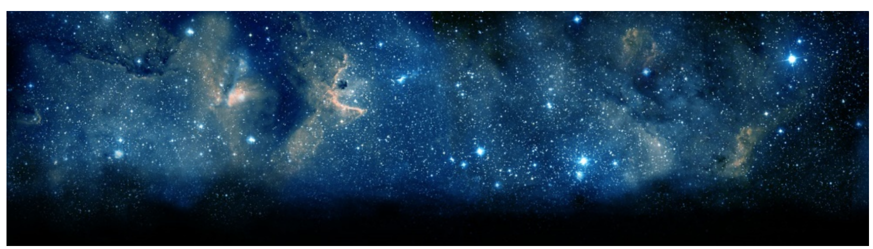


- How long does it take to check if a formula is satisfiable?
  - If somebody gives you a satisfying assignment, then in time roughly the size of the formula.
    - On a m-symbol formula, take time O(m) = constant * m, for some constant depending on the computer/software.
  - What if you don’t know a satisfying assignment? How hard it is to find it?
    - Using a truth table: in time $O(m * 2^n)$ on a length m n-variable formula.
    - Is it efficient?...

Complexity of computation

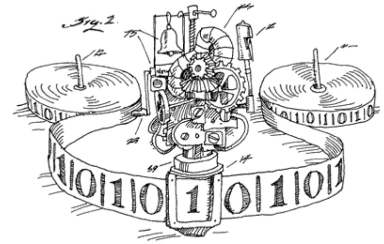


- Would you still consider a problem really solvable if it takes very long time?
  - Say 10^n steps on an n-symbol string?
  - At a billion (10^9) steps per second (~1GHz)?
  - To process a string of length 100…
  - will take $10^{100}/10^9$ seconds, or ~$3\text{x}10^{72}$ centuries.



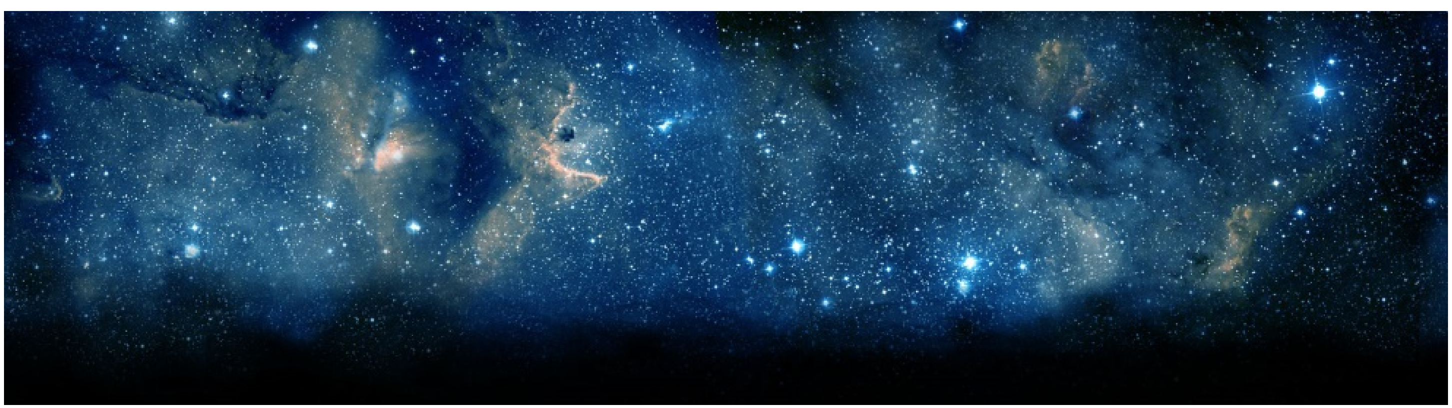
  - Age of the universe: about $1.38\text{x}10^{10}$ years.
  - Atoms in the observable universe: 10^{78}-10^{82}.

Complexity of computation

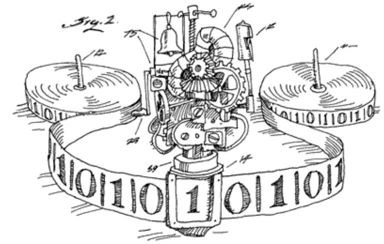


- What strings do we work with in real life?
  – A DNA string has 3.2×10^9 base pairs

  – A secure key in crypto: 128-256 bits
  – Number of Walmart transactions per day: 10^6.
  – URLs searched by Google in 2012: 3×10^{12}.



Determining formula type



- How long does it take to check if a formula is satisfiable?
  - Using a truth table: in time $O(m * 2^n)$ on a length m n-variable formula.

  - Is it efficient?
    - Not really!
    - Formula with 100 variables is already too big!
    - In software verification: millions of variables!
  - Can we do better?

A million-dollar question!

Formula simplification

- Equivalent formulas:
  - Have the same truth table.

- If two formulas F and G are equivalent, then can substitute F for G (and vice versa) in any formula H.
  - $A \land C \ \rightarrow (\neg B \lor C)$
  - We know: $(p \rightarrow q)$ is equivalent to $(\neg p \lor q)$

  - $A \land C \ \rightarrow (\neg B \lor C)$ is equivalent to: $\neg(A \land C) \lor (\neg B \lor C)$
    - But now it looks inconvenient, with that negation on the outside… Can we make it simpler?

Negation example

- Let A be “it’s sunny” and B “it’s cold”.
  - “It’s sunny and cold today”! -- No, it’s not!
  - That could mean
    - No, it’s not sunny.
    - No, it’s not cold.
    - No, it’s neither sunny nor cold.
  - In all of these scenarios, “It’s either not sunny or not cold” is true.

The law of excluded middle

- In classical logic, the law of excluded middle say that either a statement or its opposite must be true.
- But here by the opposite we really mean a negation

  - A: It is sunny.
    - $\neg$A: It is not sunny
  - A: Today is Tuesday.
    - $\neg$A : Today is not Tuesday
  - A: John votes for NDP.
    - $\neg$A : John does not vote for NDP
  - A: You are with us
    - $\neg$A : You are not with us.

De Morgan's Laws

- What is the negation of a longer logic statement?
  - Take a truth table column and flip all the values.

- Some useful simplifications: De Morgan's laws.
  - For AND: $\neg\,(A \wedge B)$ is equivalent to $(\neg A \vee \neg B)$
  - For OR: $\neg\,(A \vee B)$ is equivalent to $(\neg A \wedge \neg B)$

- Example:
  - $\neg\,(\neg A \vee B)$ is $\neg\neg\, A \wedge \neg B$, same as $A \wedge \neg B$
  - So, since $(A \rightarrow B)$ is equivalent to $(\neg A \vee B)$,

    $\neg(A \rightarrow B)$ is equivalent to $A \wedge \neg B$


Knights and knaves

- On a mystical island, there are two kinds of people: knights and knaves. Knights always tell the truth. Knaves always lie.

- Puzzle 2: You see three islanders talking to each other, Arnold, Bob and Charlie.
  - You ask Arnold “Are you a knight?”, but can’t hear what he answered.
  - Bob pitches in: “Arnold said that he is a knave!”
  - and Charlie interjects “Don’t believe Bob, he’s lying”.
  - Out of Bob and Charlie, who is a knight/knave?