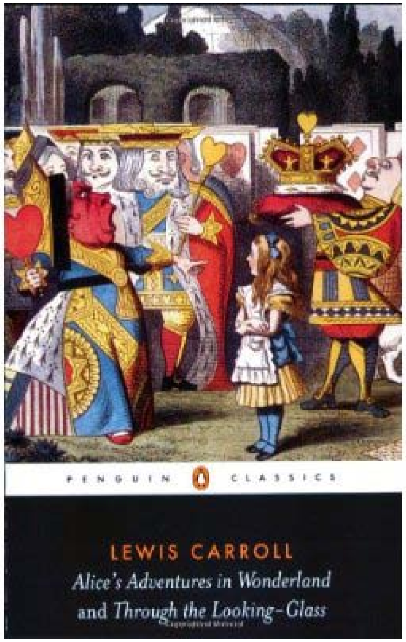
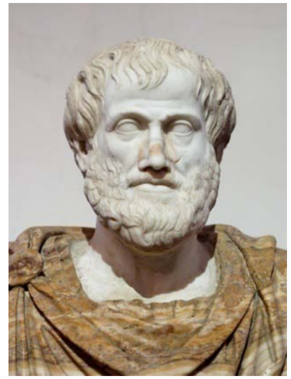
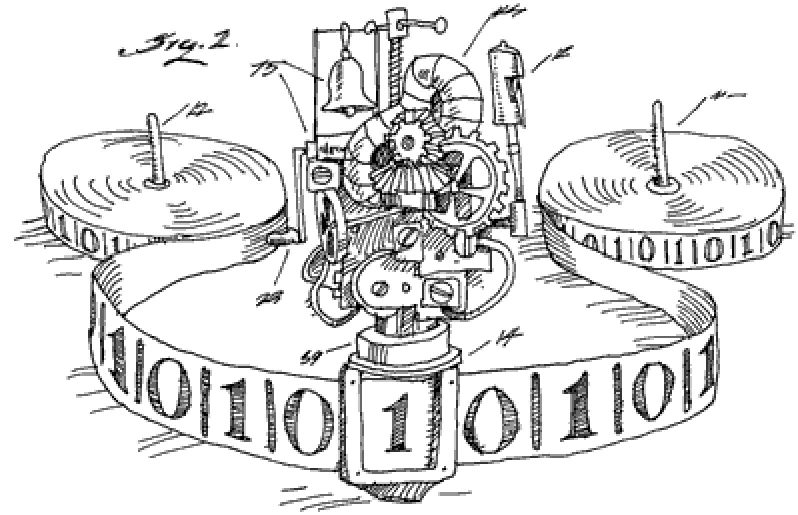
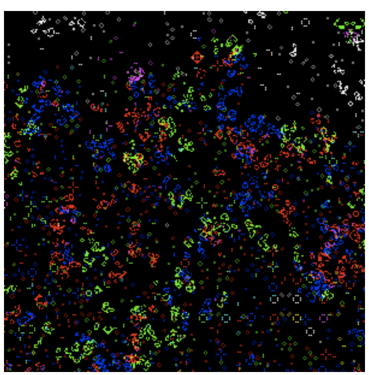


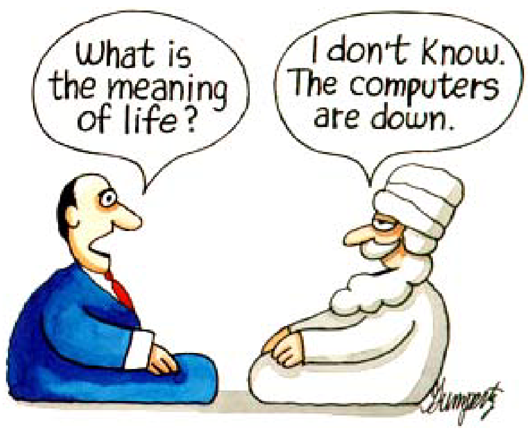


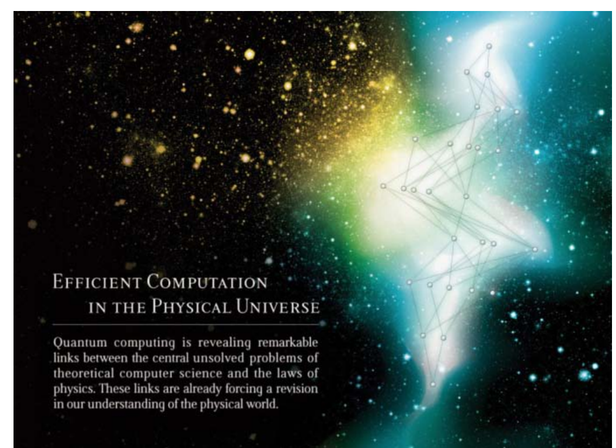


COMP2000

Computation, logic and meaning

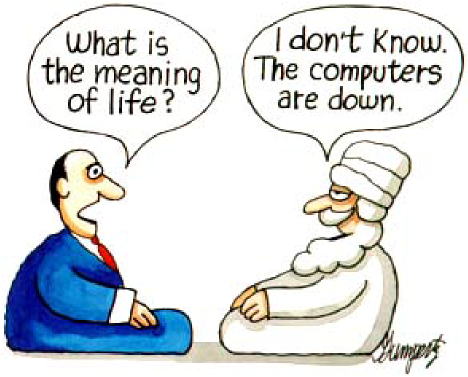






- What is computation?

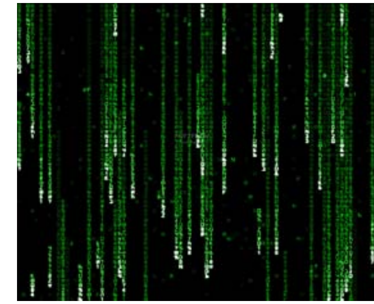
- What is information?



- What is meaning?

- What is (logical) thinking?

What is information?

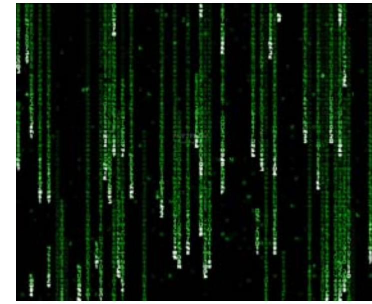


- What do you mean when you say that a certain lecture, conversation or a TV program was "informative"?

- Does it have something to do with learning something you have not known before?

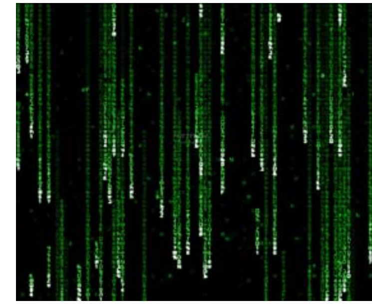
- Exact definition of information is related to entropy: see the textbook, chapter 3, for more detail.

What is information?



- Does string 11111111111 contain more information than the string 10010110100?
- How much information does a coin toss give you for a fair coin? For a coin with two heads?
- Do you learn more from a coin toss of a fair coin or a coin toss of a coin with two heads? How about a roll of dice?

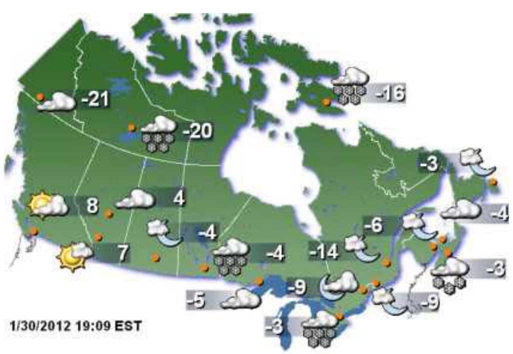
What is information?

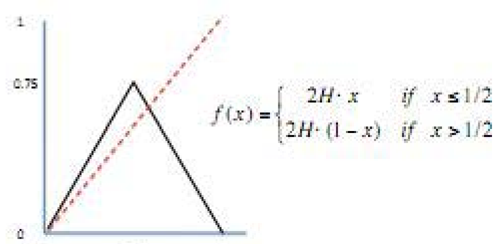


The less you can predict an outcome

The more you learn from it:

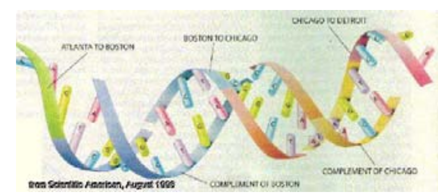
The more information you get.





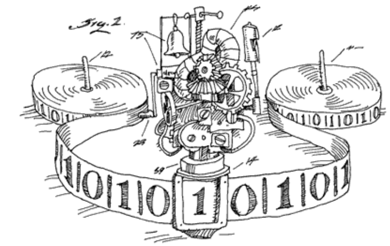
The science of information

- In many languages the word for “Computer Science” is derived from the word for information
  - French: Informatique
  - German: Informatik
  - Russian: Информатика



- The information comes in and we process it.
- So do computers. So do living cells, etc, etc.

What is computation?

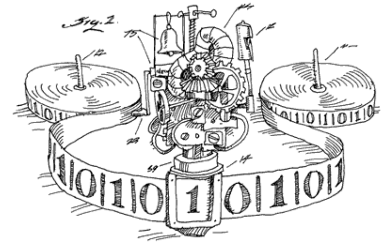


- We process information by doing a “computation on it”. Changing it from one representation to another.

- But what is computation?
  - What does your smartphone compute when you are playing Angry Birds?
  - How does DNA “compute”?

- Is there a limit to what can be computed?

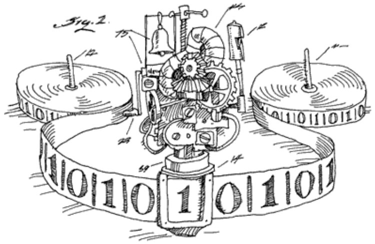
What is computation?



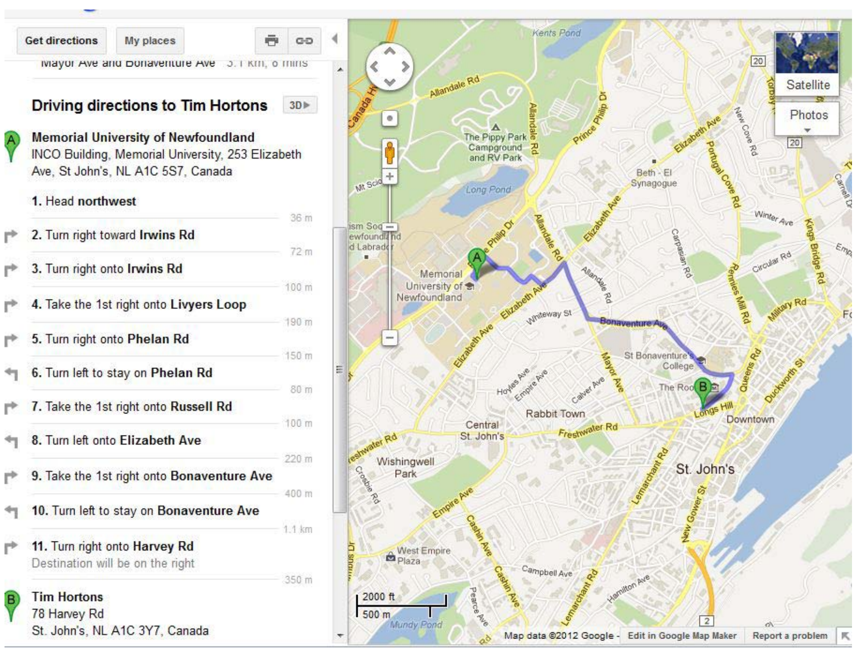
- During the World War II, hundreds of people were employed as "computers" to calculate ballistic trajectories.

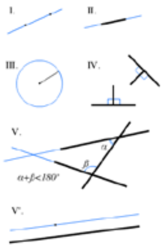
- This is the same kind of calculation as in the "Angry Birds".

What is computation?

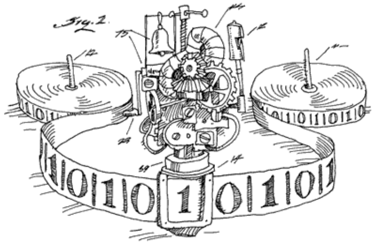


- Computation as executing a list of instructions:
  – Drive straight until you see the Basilica
  – Then turn right, and drive till the next light.
  – Then turn right, and enter Tim Hortons parking lot.



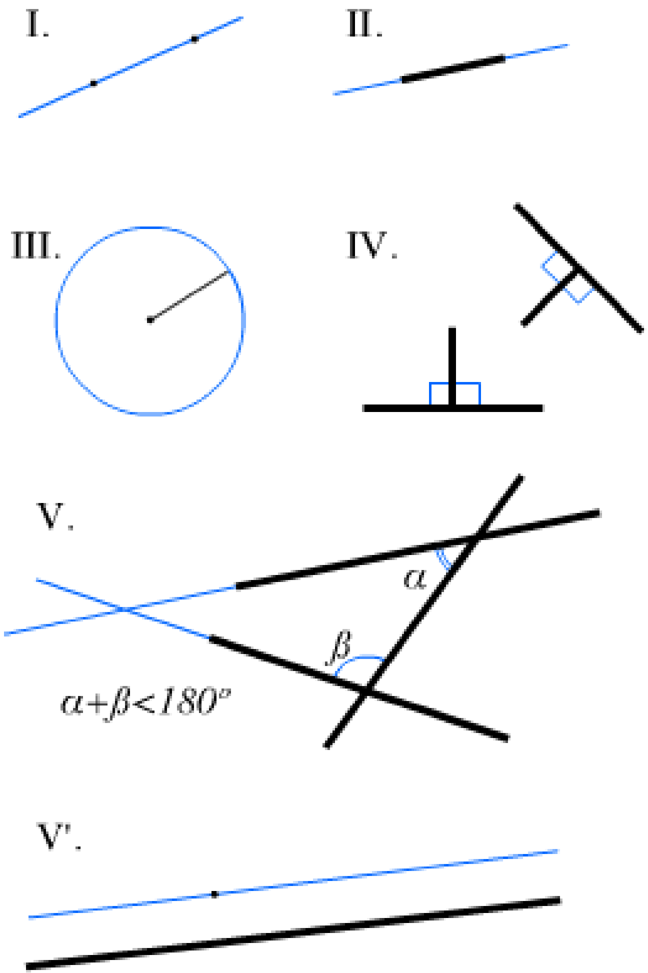


Limits of computation

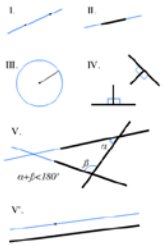


- In 1900, at the International Congress of Mathematicians in Paris, David Hilbert posed a list of 23 problems. Problem 2 asked to prove that mathematics contains no self-contradictions.
- In 1920, Hilbert extended it to what is now known as “Hilbert’s program”

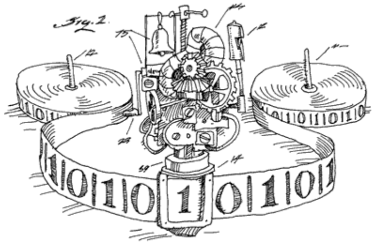
Axioms example: Euclid's postulates



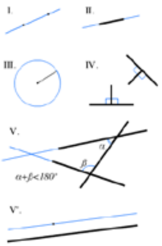
I. Through 2 points a line segment can be drawn
II. A line segment can be extended to a straight line indefinitely
III. Given a line segment, a circle can be drawn with it as a radius and one endpoint as a centre
IV. All right angles are congruent
V. Parallel postulate



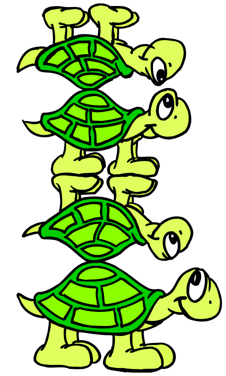
Hilbert's program



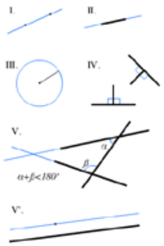
- Express all mathematics in a precise way
- Allowing a formal proof of all true statements
- With a proof, inside mathematics, that there is no self-contradiction
- And a procedure (an algorithm) for deciding, for any given mathematical statement, whether it is true or false.



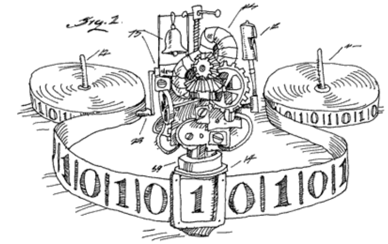
Gödel Incompleteness Theorem



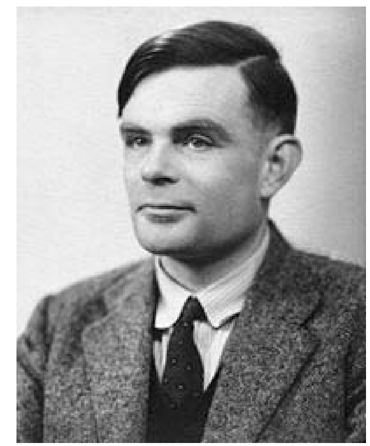
- If mathematics is not self-contradictory...
- Then there are statements that can’t be proven!
- Such as “I am not provable”
- Like with dynamical systems, self-reference leads to something strange, a paradox!



Church and Turing:



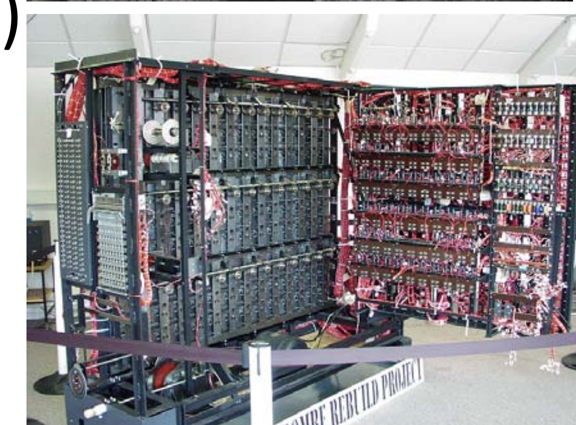
- Moreover,
- there is no procedure
- to decide if a given statement is true or false!
- And to decide many other things...

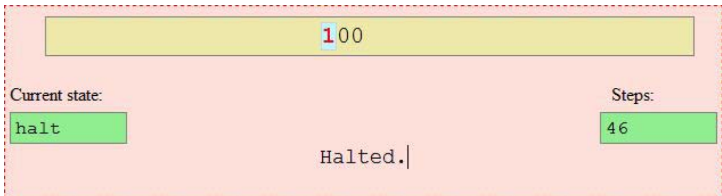


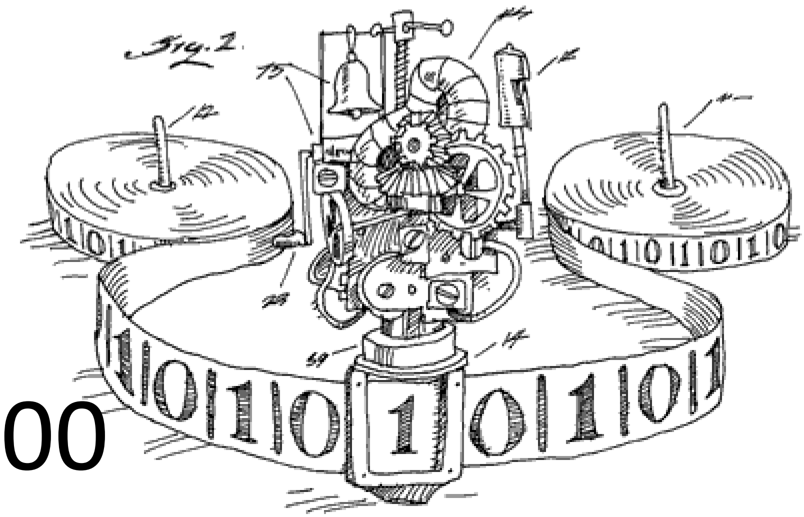
- But what do we mean by a “procedure”?

Digression: a bit about Alan Turing

- This year marks 100 years from Alan Turing's birth
- He is known for
  - The Turing machine
  - Breaking German's codes (Enigma machine) during the World War II
  - The Turing test (Artificial intelligence)
- He was prosecuted for homosexuality and died from suicide in 1954...

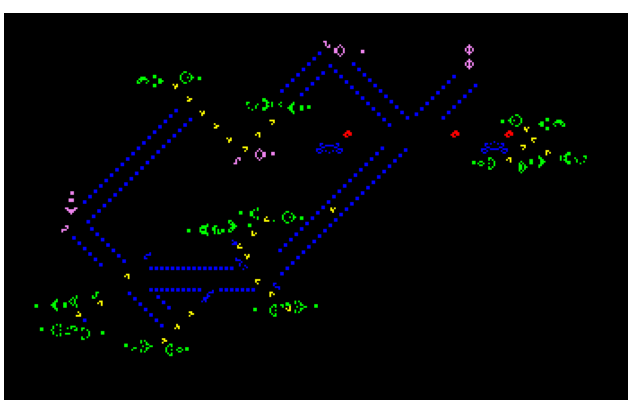


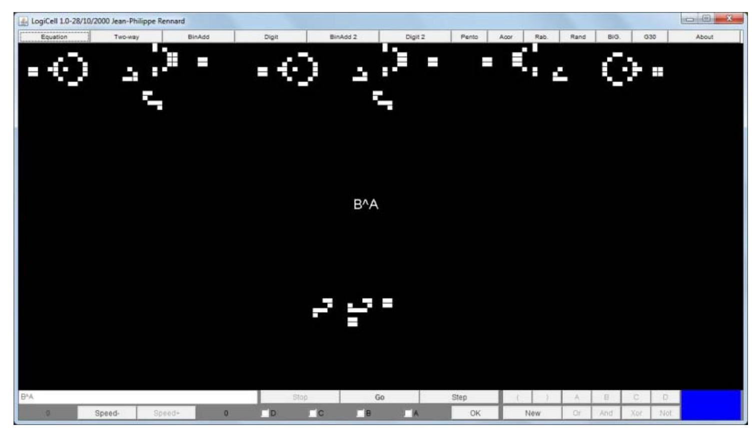




COMP2000
Logic and Computation
Lecture 2

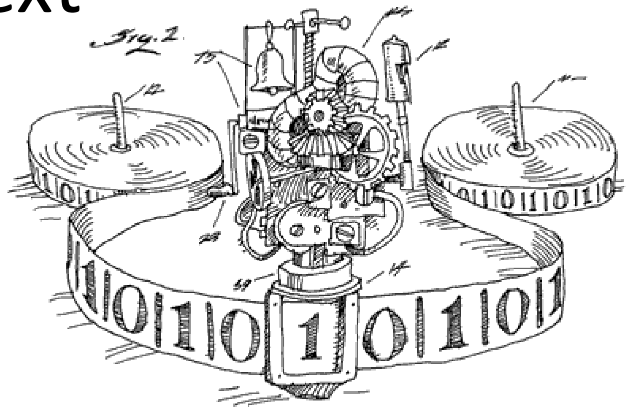
Life vs. machine



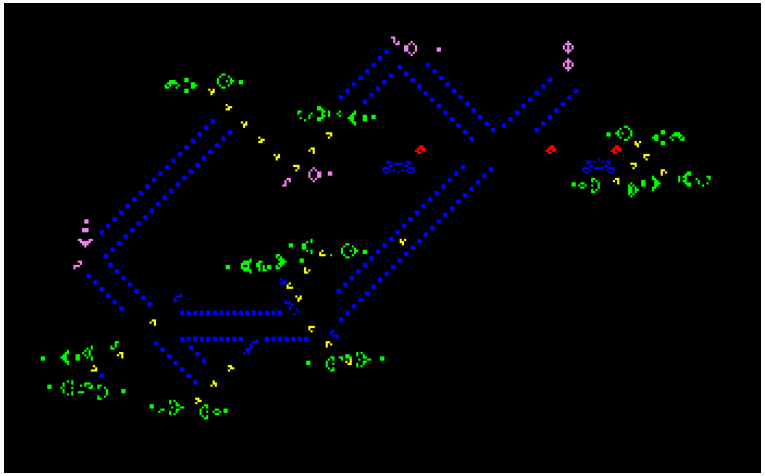


Administrative stuff

- Lab is February 15th (next Wednesday).

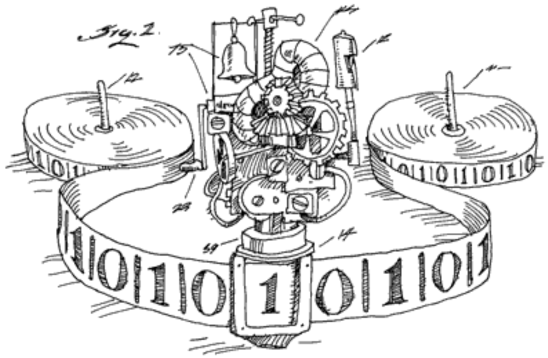


- Readings so far:
  - Chapter 1: Information
  - Chapter 2: Computation
  - Chapter 10: Cellular automata

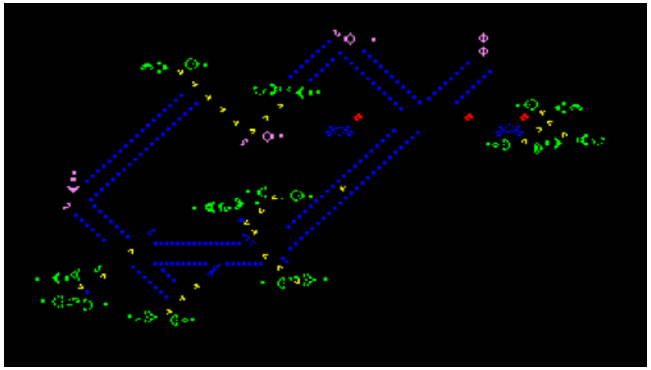


Models of computation

- In this lecture, we will talk about two (surprisingly, equivalent) models of computation
- The first one is the Turing machine
  - Our modern-day computers are based on this model

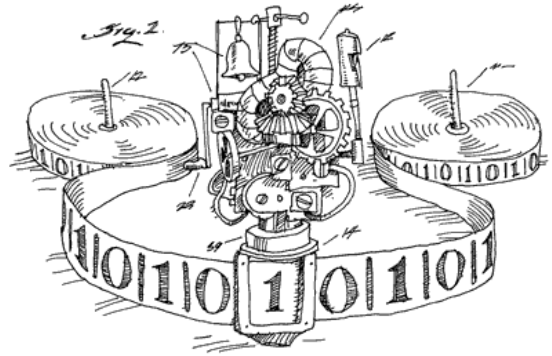


- The second is the Game of Life
  - Looks nothing like a computer, and yet has the same power.

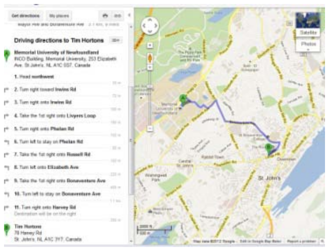


Turing machine

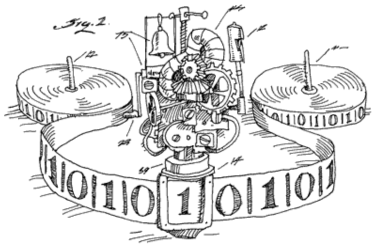
- A Turing machine has an (unlimited) memory, visualized as a tape



- Or a stack of paper
- And takes very simple instructions:
  – Read a symbol
  – Write a symbol
  – Move one step left or right on the tape
  – Change internal state.



Executing instructions



- Drive straight until you see the Basilica

  Internal state: looking for Basilica

  Go straight. Check for Basilica. Repeat.

- Then turn right, and drive till the next light.

  Turn right.

  Change state to "Look for traffic light"

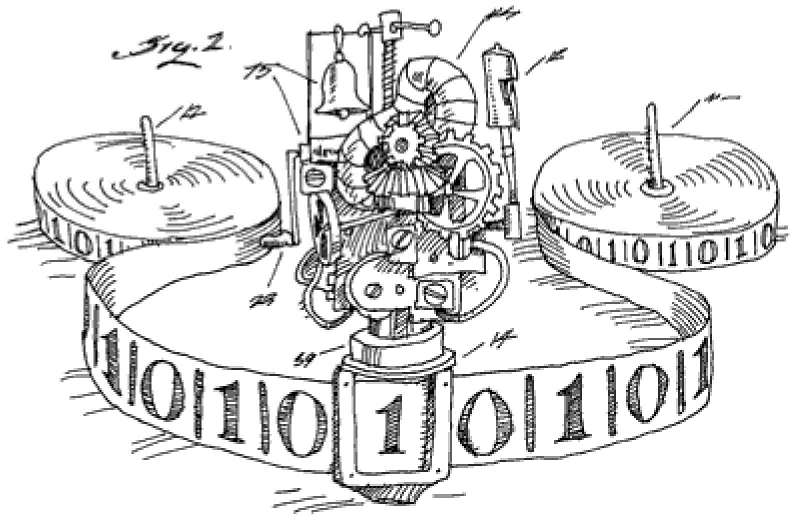
  Go straight. Check for traffic light. Repeat.

- Then turn right, and enter Tim Hortons parking lot.

  Change state to "Look for Tim Hortons"

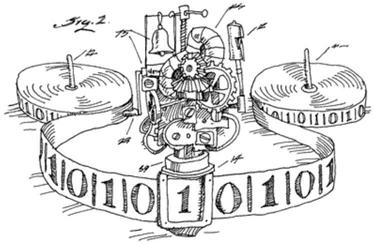
  When see Tim Hortons, turn right into the parking lot

Church-Turing thesis

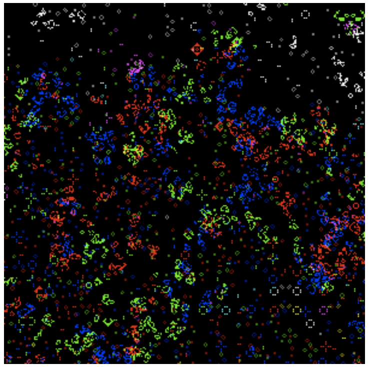


- Everything we can call “computable” in any sense of this word is computable by a Turing machine.

Church-Turing thesis

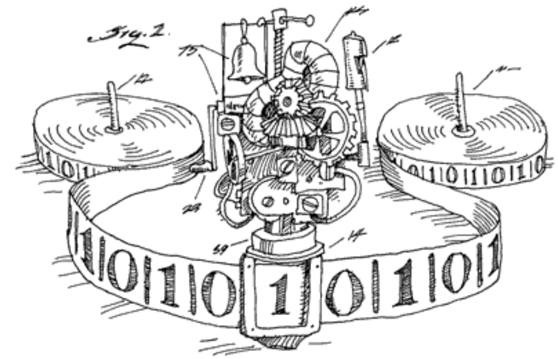


***Everything we can call "computable" in any sense of this word is computable by a Turing machine*.**

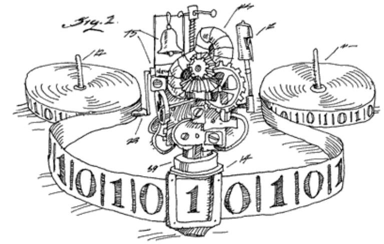


Turing machine

- A Turing machine computation starts with the tape blank except for the input
- It starts in the special start state looking at the start of the input
- Then keeps reading, writing and changing states according to the rules
- It may never stop
- If it stop, what is written on the tape is its output.



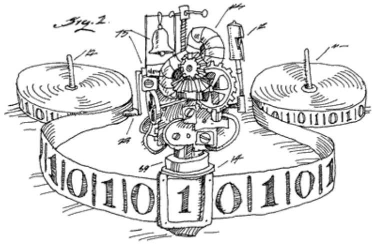
Turing machine example



- Check if the tape is empty:
  – At the start, read the first symbol
  – If it is blank, say "yes"
  – Otherwise, say "no"
- Instructions:

Current state	Reads	Writes	Moves	New state
Start_state	_	Y	Right	Halt
Start_state	0	N	Right	Halt
Start_state	1	N	Right	Halt

Turing machine example



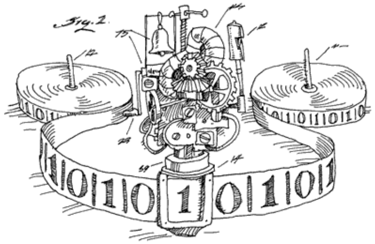
- "Check if the tape is empty" instructions:

Current state	Reads	Writes	Moves	New state
Start_state	_	Y	Right	Halt
Start_state	0	N	Right	Halt
Start_state	1	N	Right	Halt

- Simulator program:

```
0             *   *   *   start_state      ; rename start state

start_state   _   Y   r   halt    ; if empty, write Y and stop
start_state   0   N   r   halt    ; if tape has 0, write N and stop
start_state   1   N   r   halt    ; if tape has 1, write N and stop
```

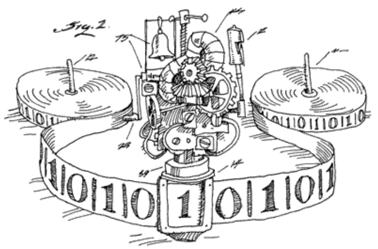
Turing machine example



- Check if the tape contains a 1:
  - At the start, read the first symbol
  - If it is 1, say “yes”
  - Otherwise, move right and repeat (keep looking)
  - Seeing a blank, say “no”
- Instructions:

Current state	Reads	Writes	Moves	New state
Start_state	_	N	Right	Halt
Start_state	0	0	Right	Start_state
Start_state	1	Y	Right	Halt

Turing machine example



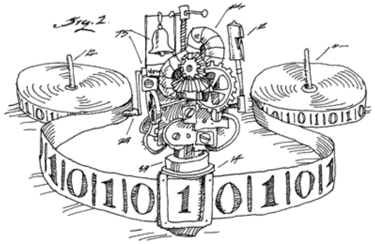
- “Check if the tape contains a 1 ” instructions:

Current state	Reads	Writes	Moves	New state
Start_state	_	N	Right	Halt
Start_state	0	0	Right	Start_state
Start_state	1	Y	Right	Halt

- Simulator program:

```
0              *   *   *   start_state    ; rename start state

start_state    _   N   r   halt           ; if reached blank, write N and stop
start_state    0   0   r   start_state    ; if still on input and no 1, repeat
start_state    1   Y   r   halt           ; if seeing a 1, write Y and stop
```

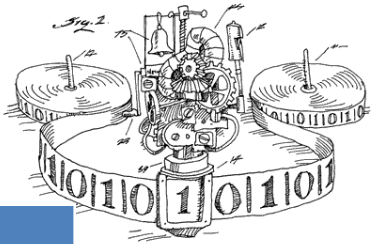
Turing machine example



- Check if the tape is empty:
  - At the start, read the first symbol
  - If it is blank, say “yes”
  - Otherwise, erase the tape and say “no”
- Instructions:

Current state	Reads	Writes	Moves	New state
Start_state	_	Y	Right	Halt
Start_state	0	_	Right	no_state
Start_state	1	_	Right	no_state
no_state	_	N	Right	Halt
no_state	0	_	Right	no_state
no_state	1	_	Right	no_state

Check if the tape is empty

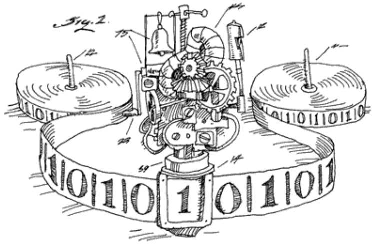


Current state	Reads	Writes	Moves	New state
Start_state	_	Y	Right	Halt
Start_state	0	_	Right	no_state
Start_state	1	_	Right	no_state
no_state	_	N	Right	Halt
no_state	0	_	Right	no_state
no_state	1	_	Right	no_state

- Simulator program:

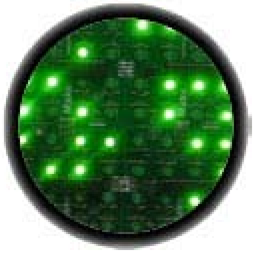
```
0            *   *   *   start_state      ; rename start state

start_state  _   Y   r   halt             ; if empty, write Y and stop
start_state  0   _   r   no_state         ; if tape has 0 or 1,  start erasing
start_state  1   _   r   no_state         ; while erasing, remember “no”

no_state     _   N   r   halt             ; tape is empty, write N and stop
no_state     0   _   r   no_state         ; keep erasing remembering “no”
no_state     1   _   r   no_state         ;keep erasing remembering “no”
```

Turing machine



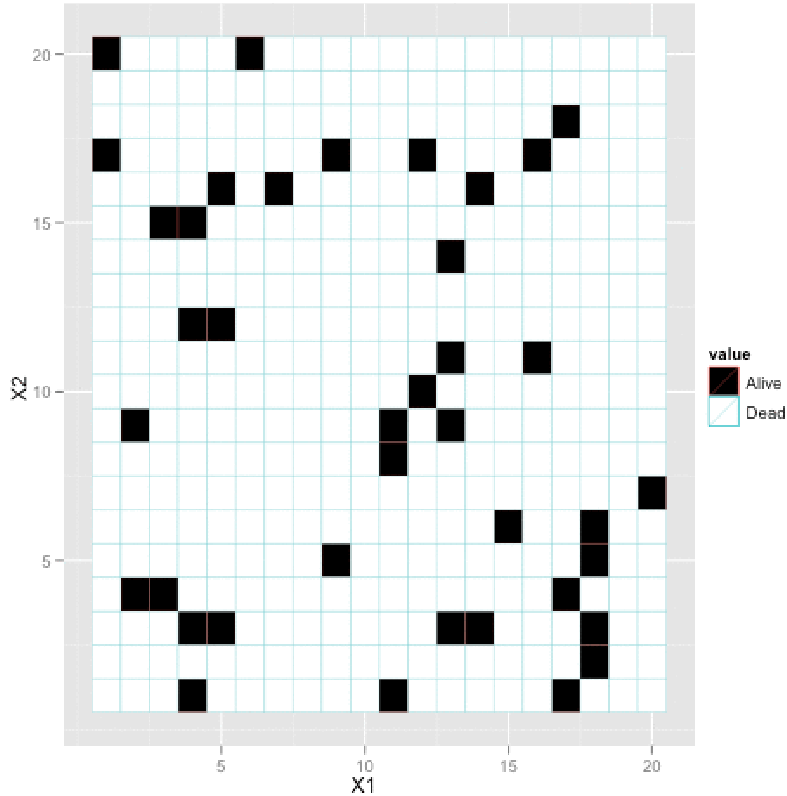
- Can do arithmetic (in binary)
  - see example of add 1
- Can do logic
  - Topic of the next class
- Can simulate any model of computation so far
  - Church-Turing thesis.
- Can have self-replicating programs
- Cannot solve some problems
  - "Am I lying"? "Is this true?"
  - "Will this computation ever stop?"

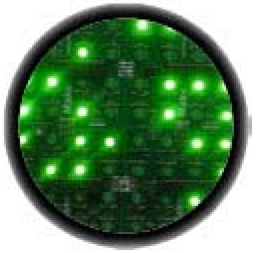
Does it mean nobody can solve them?



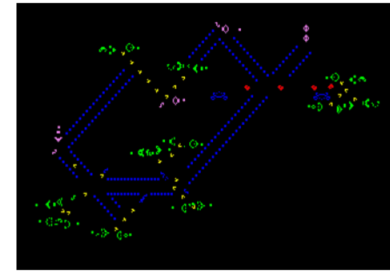
Conway's game of life

- Rules of the Game of Life:
- Start with a board with a square grid
- Mark some grid cells as "live"
- At every step of the game:
  - Every live cell with less than 2 neighbours dies
  - Every live cell with more than 3 neighbours dies
  - A cell with exactly 3 neighbours becomes alive (is "born").



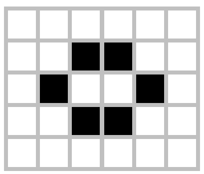


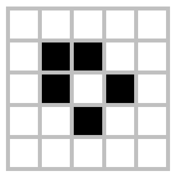
Conway's game of life: what can it do?



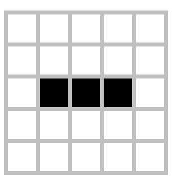
- Rules of the Game of Life:
- At every step of the game:
  - Every live cell with less than 2 neighbours dies
  - Every live cell with more than 3 neighbours dies
  - A cell with exactly 3 neighbours becomes alive (is "born").

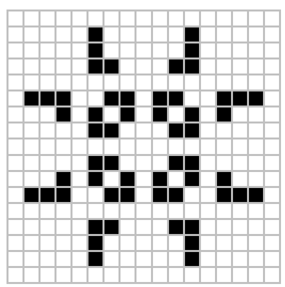
- Converge to a still pattern



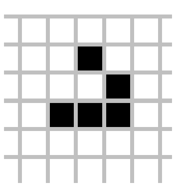


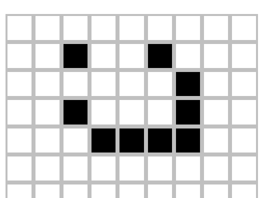
- Oscillate



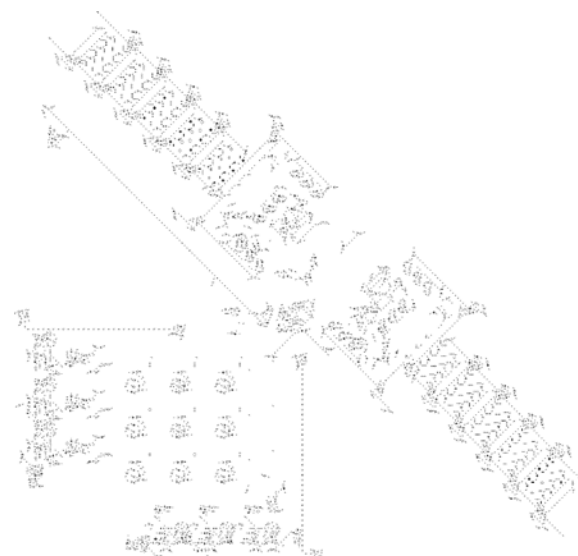


- Create a moving pattern



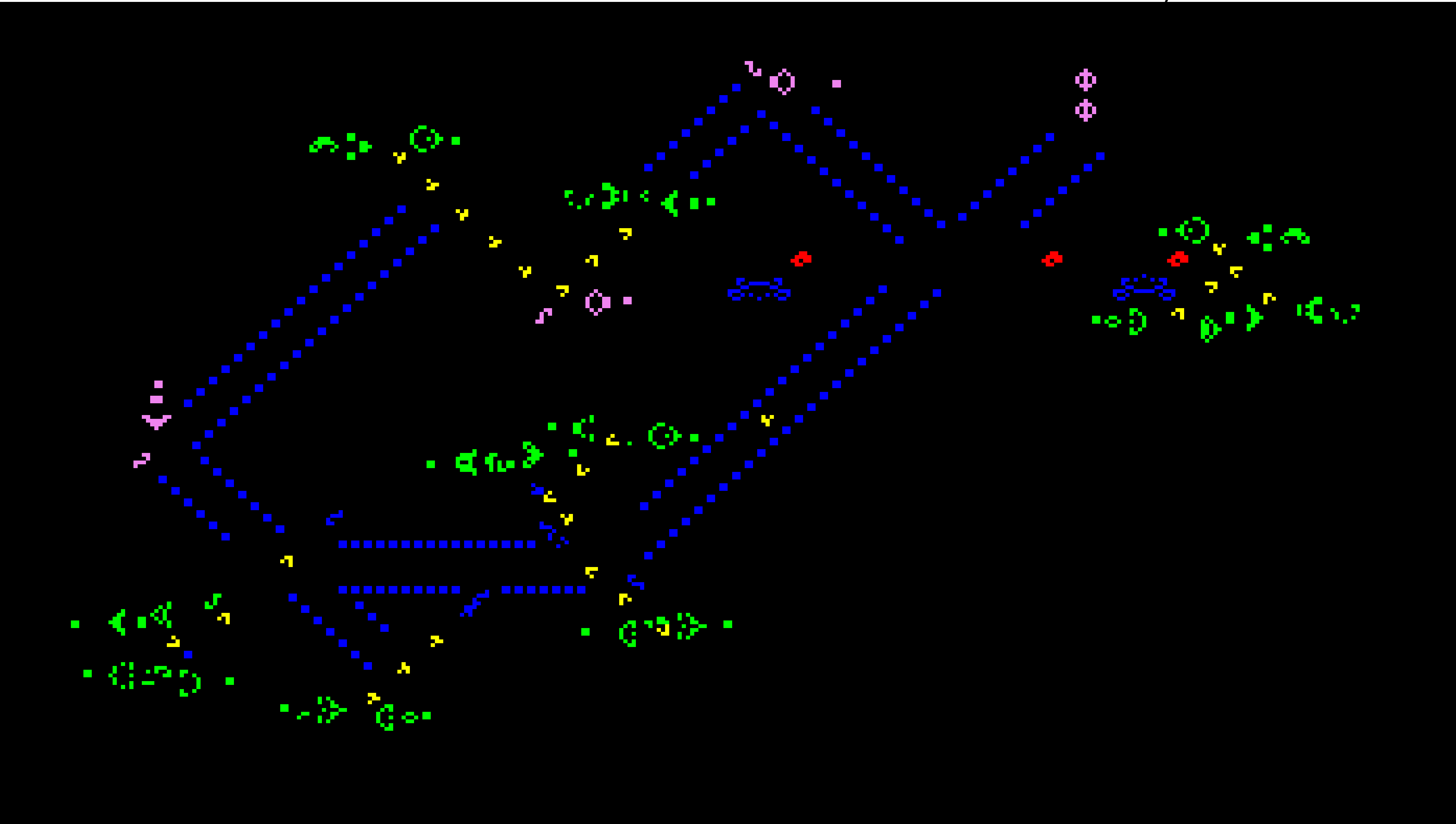


- Simulate a Turing machine



Conway's game of life: what can it do?

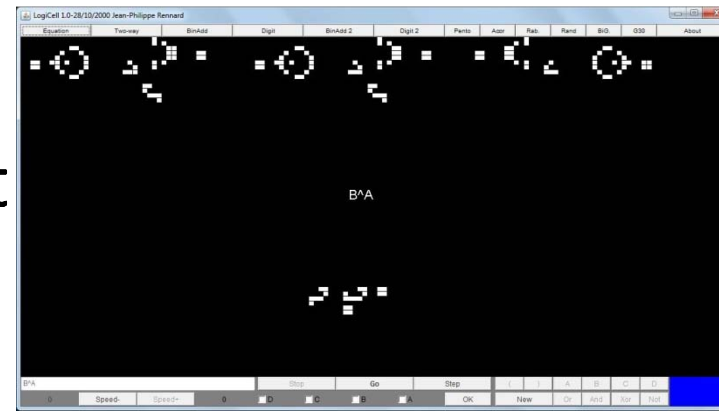
- At every step of the game:
  - Every live cell with less than 2 neighbours dies
  - Every live cell with more than 3 neighbours dies
  - A cell with exactly 3 neighbours becomes alive (is "born").

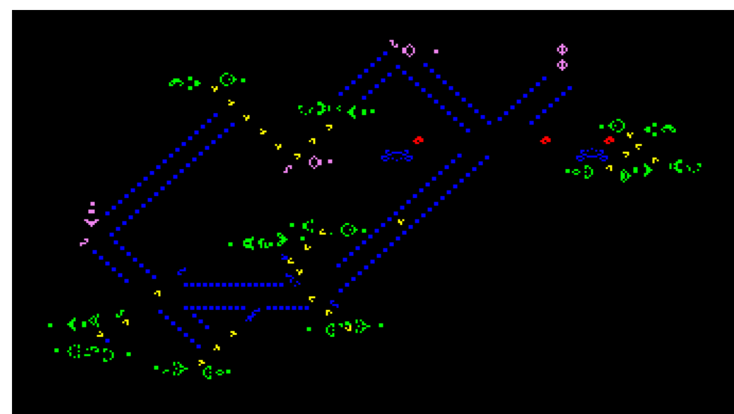


Conway's game of life: what does it mean to compute?

- Rules of the Game of Life:
- At every step of the game:
  - Every live cell with less than 2 neighbours dies
  - Every live cell with more than 3 neighbours dies
  - A cell with exactly 3 neighbours becomes alive (is "born").

- Start with a few cells lit up
- See if cells somewhere else light up
- Make it so they only light up if some condition holds
- Just like a Turing machine writing "Y" on the tape if some condition holds about its input

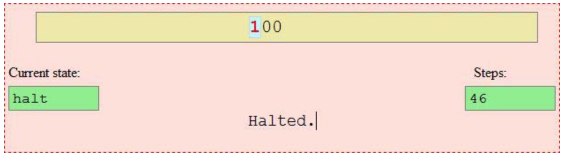




Simulators

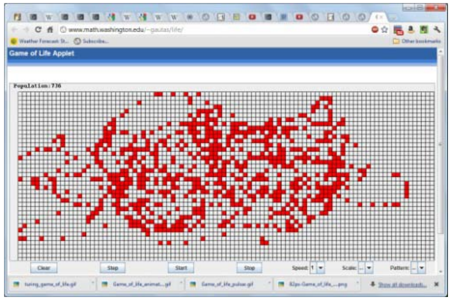
- Turing machine
  - Use this as the first line to give name to the start state

    0 * * * start_state

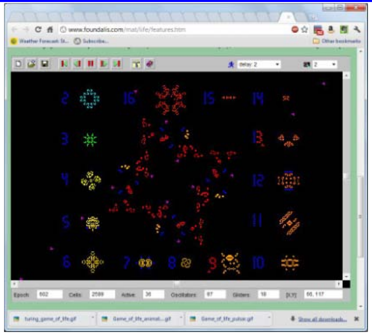


http://morphett.info/turing/turing.html

- Game of Life:
  - http://www.math.washington.edu/~gautas/life/

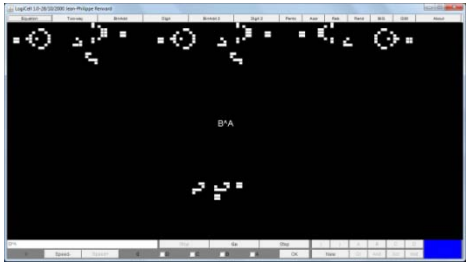


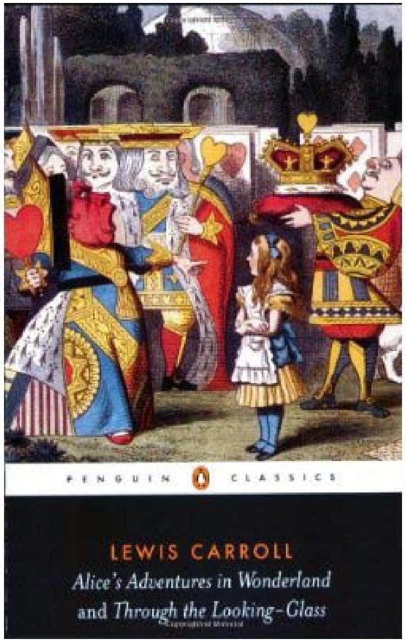
  - (with colours)

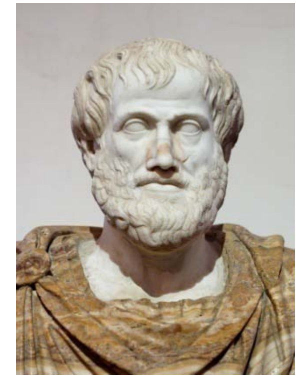


http://www.foundalis.com/mat/life/features.htm

  - LogiCell http://www.rennard.org/alife/english/logicellgb.html





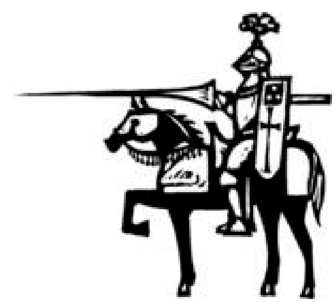


COMP2000

Lecture 3

Logic: puzzles, truth and human fallacies

B 5 2 J



Do we think logically?

- You see the following cards. Each has a letter on one side and a number on the other.

- Which cards do you need to turn to check that if a card has a J on it then it has a 5 on the other side?

Do we think logically?

- You see the following cards. Each has a letter on one side and a number on the other.

- Which cards do you need to turn to check that if a card has a J on it then it has a 5 on the other side?

Do we think logically?

- You see the following cards. Each has a letter on one side and a number on the other.

- Which cards do you need to turn to check that if a card has a J on it then it has a 5 on the other side?

Do we think logically?

- You see the following cards. Each has a letter on one side and a number on the other.

B 5 2 5

- Which cards do you need to turn to check that if a card has a J on it then it has a 5 on the other side?

Do we think logically?

- You see the following cards. Each has a letter on one side and a number on the other.

- Which cards do you need to turn to check that if a card has a J on it then it has a 5 on the other side?

Do we think logically?

- You see the following cards. Each has a letter on one side and a number on the other.

- Which cards do you need to turn to check that if a card has a J on it then it has a 5 on the other side?

Do we think logically?

- You see the following cards. Each has a letter on one side and a number on the other.

- Which cards do you need to turn to check that if a card has a J on it then it has a 5 on the other side?

Another puzzle

- You are one of the organizers at a mixer; many people are drinking, some are not.
- You need to make sure that nobody underage is drinking: that is, if somebody is drinking, then they are over 19 years old.

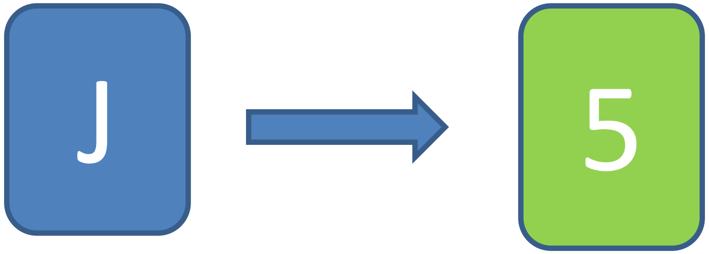
   

- Which category of people will you need to check?

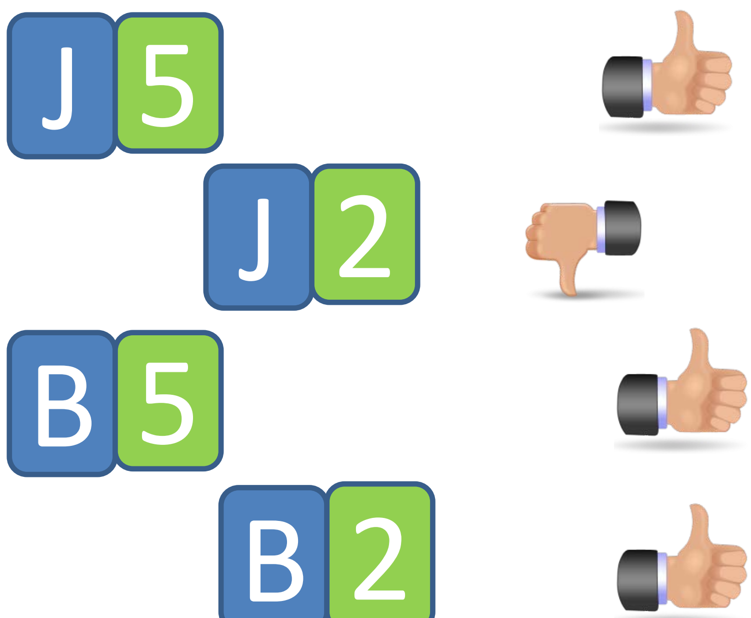
“if … then” in logic

- Both puzzles have the same structure:

“if A then B”



- What circumstances make this true?
  - A is true and B is true
  - A is true and B is false
  - A is false and B is true
  - A is false and B is false



$A \rightarrow B$

- We make logical conclusions all the time
- But do we always make them “logically”?
- Sometimes people think that “if ... then” goes both ways...
  - If you live in NL, you must pay HST. John lives in BC. Does he pay HST?
  - If today it Tuesday, then there is a COMP2000 lecture. Today is Thursday. Is there a lecture?

Natural vs. Logic language

- Natural languages are ambiguous.
- For example, the word “any” can have different meanings depending on the context:

- Any = some
  - She will be happy if she can solve any question.
  - She will be happy if she can solve every question.

- Any = all
  - Any student knows this.
  - Every student knows this.

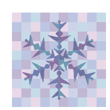
Language of logic

Pronunciation	Notation	Meaning
A and B	A /\ B	True if both A and B are true
A or B	A \/ B	True if either A or B are true (or both)
If A then B	A -> B	True whenever if A is true, then B is also true
Not A	~A	Opposite of A is true, so true if A is false

- Let A be "It is sunny" and B be "it is cold"
  - A /\ B: It is sunny and cold
  - A \/ B: It is either sunny or cold
  - A -> B: If it is sunny, then it is cold
  - ~A: It is not sunny

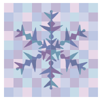
Language of logic

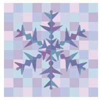
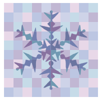
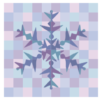
Pronunciation	Notation	True when
A and B	A /\ B	Both A and B must be true
A or B	A \/ B	Either A or B must be true (or both)
If A then B	A -> B	if A is true, then B is also true
Not A	~A	Opposite of A is true

- Now we can combine these operations to make longer formulas in the language of logic
- Let
  - A be "It is sunny", 
  - B be "it is cold", 
  - C be "It's snowing" 
- Let's make some sentences out of A, B, C

 Language of logic 

Pronunciation	Notation	True when
A and B	A /\ B	Both A and B must be true
A or B	A \/ B	Either A or B must be true (or both)
If A then B	A -> B	if A is true, then B is also true
Not A	~A	Opposite of A is true

- Let
  - A be “It is sunny”, 
  - B be “it is cold”, 
  - C be “It’s snowing” 

- What are the translations of:
  - B /\ C -> ~A IF ( AND ) THEN NOT 
    - If it is cold and snowing, then it is not sunny
  - B -> (C \/ A) IF  THEN ( OR )
    - If it is cold, then it is either snowing or sunny
  - ~A /\ A -> C IF (NOT  AND ) THEN 
    - If it is sunny and not sunny, then it is snowing.

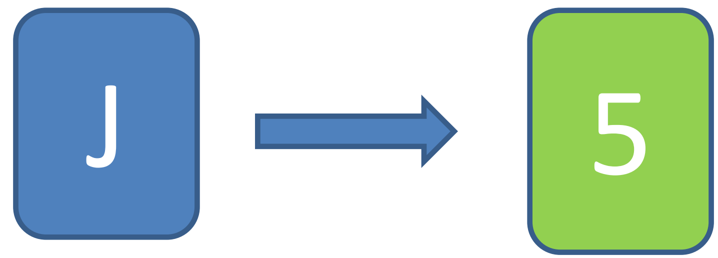
The truth

- We talk about a sentence being true or false when the values of the variables are known.
- If we didn't know whether it is sunny, we would not know whether $A \wedge B \rightarrow C$ is true or false.
- If a sentence is true for every possible combinations of variable truth assignments, we call it a "tautology"

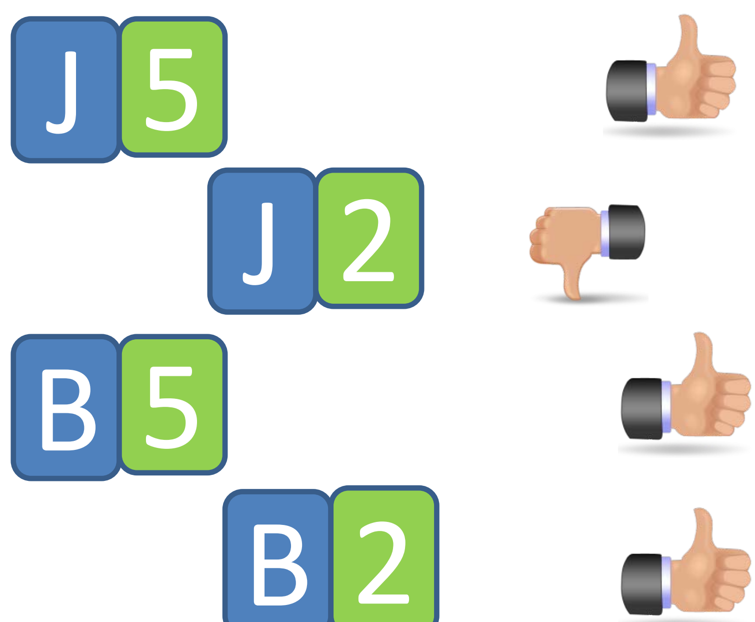
“if … then” in logic

- Both puzzles have the same structure:

“if A then B”



- What circumstances make this true?
  - A is true and B is true
  - A is true and B is false
  - A is false and B is true
  - A is false and B is false



Truth tables

A	B	not A	A and B	A or B	if A then B
True	*True*	False	True	True	True
True	*False*	False	False	True	False
False	*True*	True	False	True	True
False	*False*	True	False	False	True

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

- Let
  - A be “It is sunny”
  - B be “it is cold”
- It is sunny and cold.
- It is sunny and not cold
- It is not sunny and cold
- It is neither sunny nor cold

Truth tables

- Let
  - A be “It is sunny”
  - B be “it is cold”
- It is sunny and cold.
- It is sunny and not cold
- It is not sunny and cold
- It is neither sunny nor cold
- Now, ~ A $\bigvee$ B is:

A	B	not A	A and B	A or B	if A then B
True	*True*	False	True	True	True
True	*False*	False	False	True	False
False	*True*	True	False	True	True
False	*False*	True	False	False	True

A	B	(Not A) or B
True	*True*	*True*
True	*False*	*False*
False	*True*	*True*
False	*False*	*True*

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

Negating composite statements

- What is the negation (opposite) of a longer logic statement? Take a truth table column and flip all the values.
  - The negation of "A and B", not(A and B), is true whenever either A or B is false (check the truth table. That is not(A and B) is the same as (not A or not B).
  - The negation of "A or B" is true whenever both A and B are false: not(A or B) is the same as (not A and not B).
  - Since "if A then B" is the same as "not A or B", its negation is "A and not B". Remember that "if A then B" is only false when A is true, and B is false; check that "A and not B" is true in exactly the same scenario.

NOT'ing longer sentences

- For a longer combination, start with the connective applied last when computing a truth table:
  - "not (if (A or not B) then (A and C))" becomes
  - (A or not B) and not(A and C) by negating "if... Then.."
  - (A or not B) and (not A or not C) by negating the last "and"
- 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.

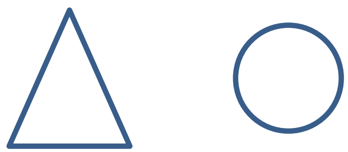
 -> 

More on “not if.. then”

- Remember that for “if A then B” there is only one scenario when it is false: it is when A is true, and B is false.
- Let A be “it’s raining” and B “it is cloudy”. Then “if A then B” means if it is raining it must be cloudy.
- “not (if A then B)” means “it’s not true that when it is raining it must be cloudy”. Or, equivalently, “it’s raining, and it is not cloudy” (there is probably a rainbow then, too) .
- “not (if A then B) is definitely not a negation of “if B then A”!

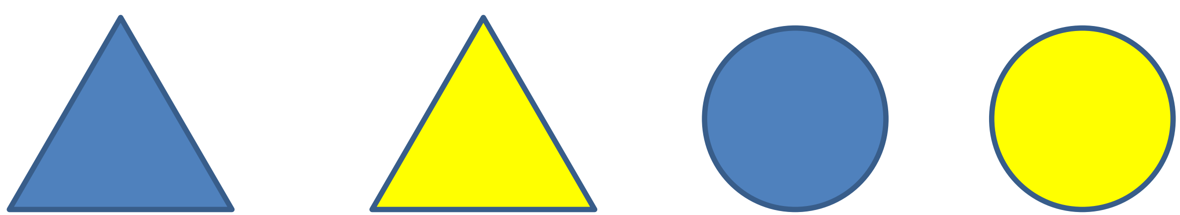
Or: elusive, not exclusive.

- I like one of the shapes.



- I like one of the colours.

- I like a figure if it has either my favourite shape or my favourite colour.



- I like the blue triangle. What can you say about the rest?

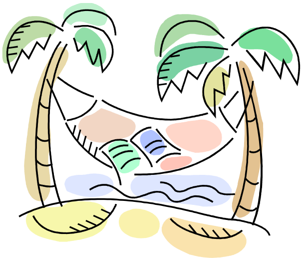
Proof vs. disproof

- To prove that something is (always) true:
  - Make sure it holds in every case.
    - I have classes on every day that starts with T. I have classes on Tuesday and Thursday (and Friday, but that's irrelevant).
  - Or assume it does not hold, and then get something strange as a consequence
    - Suppose there are finitely many prime numbers. What divides the number that's a product of all primes +1?
- To disprove that something is always true:
  - Give just one example where it breaks down.
    - I have classes every day! – No, you don't have classes on Saturday!
    - All girls hate math! – No, I love math and I am a girl :)

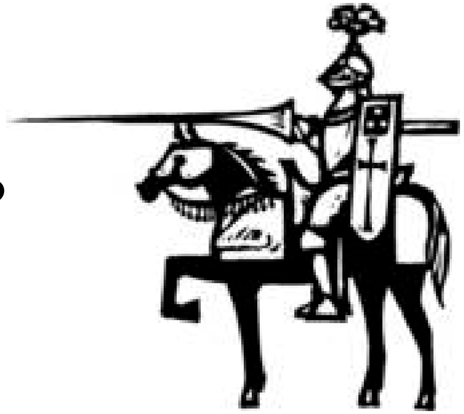
Proof vs. disproof

- To prove that something is (always) true:
  - Make sure it holds in every case.
    - I have classes on every day that starts with T. I have classes on Tuesday and Thursday (and Friday, but that's irrelevant).
  - Or assume it does not hold, and then get something strange as a consequence
    - Suppose there are finitely many prime numbers. What divides the number that's a product of all primes +1?
- To disprove that something is always true:
  - Give just one example where it breaks down.
    - I have classes every day! – No, you don't have classes on Saturday!
    - All girls hate math! – No, I love math and I am a girl :)

Knights and knaves



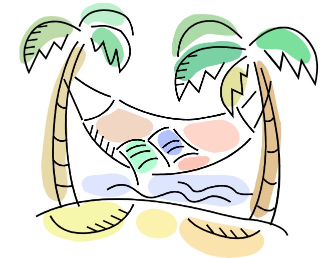
- On a mystical island, there are two kinds of people: knights and knaves.

-  Knights always say the truth.

- Knaves always lie.

Knights and knaves



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

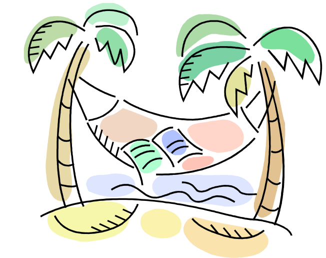
- Puzzle 1: You meet two people on the island, Arnold and Bob. Arnold says “at least one of us is a knave”. Is Arnold a knight or a knave? What about Bob?

Knights and knaves

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

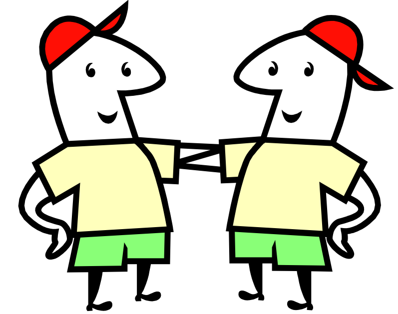
- Puzzle 2: You meet two people on the island, Arnold and Bob. Arnold says “Either I am NOT a knight, or Bob is a knave” Is Arnold a knight or a knave? What about Bob?


Knights and knaves

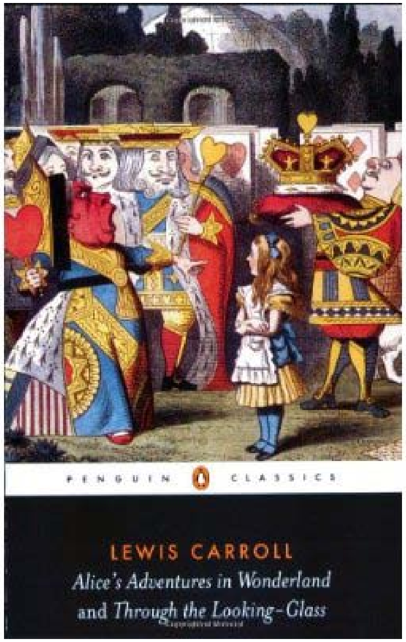


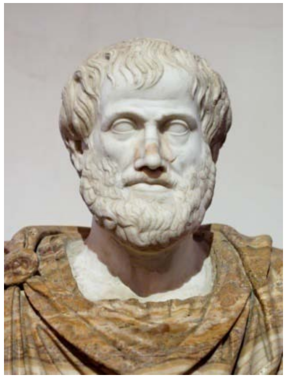
- On a mystical island, there are two kinds of people: knights and knaves. Knights always tell the truth. Knaves always lie.
- Puzzle 3: You see three islanders talking to each other, Arnold, Bob and Charlie. You ask Arnold "Are you a knight?", but can't hear what he answered. Bob pitches in: "Arnold 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 a who is a knave?

Twins puzzle



- There are two identical twin brothers, Dave and Jim.
- One of them always lies; another always tells the truth (like knights and knaves).
- Suppose you see one of them and you want to find out his name.
- How can you learn if you met Dave or Jim by asking just one short yes-no question? You don’t know which one of them is the liar.

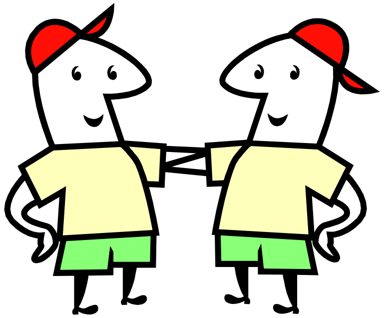




COMP2000

Lecture 4

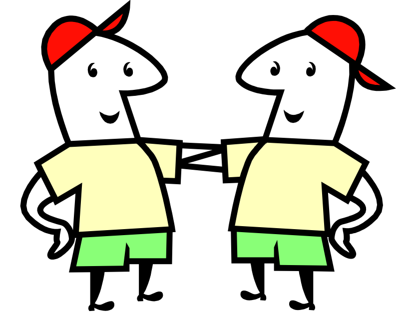
Logic: treasures, mysteries, All and NOT



Lab tomorrow!

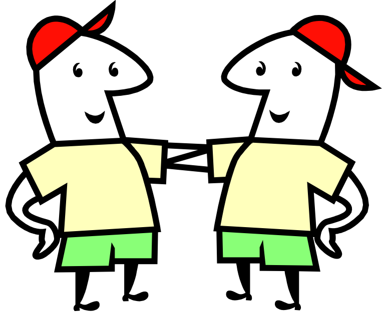
- The lab is tomorrow (but you can do it any time)
- It is posted from the main page (and mine, too)
- Let's go over the lab before we start the lecture

Twins puzzle



- There are two identical twin brothers, Dave and Jim.
- One of them always lies; another always tells the truth (like knights and knaves).
- Suppose you see one of them and you want to find out his name.
- How can you learn if you met Dave or Jim by asking just one short yes-no question? You don’t know which one of them is the liar.

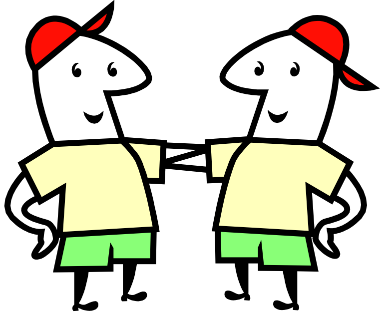

Twins puzzle



- There are two identical twin brothers, Dave and Jim.
- One of them always lies; another always tells the truth (like knights and knaves).
- Suppose you see one of them and you want to find out his name.
- How can you learn if you met Dave or Jim by asking just one short yes-no question? You don't know which one of them is the liar.

This is Jim	Jim is a liar				
Yes	Yes				
Yes	No				
No	Yes				
No	No				

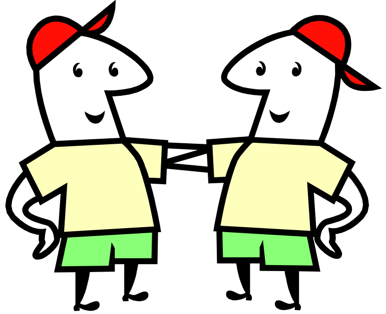
Twins puzzle



- There are two identical twin brothers, Dave and Jim.
- One of them always lies; another always tells the truth (like knights and knaves).
- Suppose you see one of them and you want to find out his name.
- How can you learn if you met Dave or Jim by asking just one short yes-no question? You don't know which one of them is the liar.

This is Jim	Jim is a liar	This is a liar			
Yes	Yes	Yes			
Yes	No	No			
No	Yes	No			
No	No	Yes			

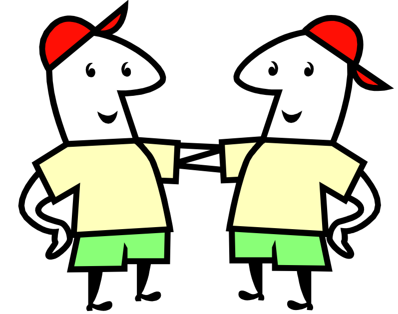
Twins puzzle



- There are two identical twin brothers, Dave and Jim.
- One of them always lies; another always tells the truth (like knights and knaves).
- Suppose you see one of them and you want to find out his name.
- How can you learn if you met Dave or Jim by asking just one short yes-no question? You don't know which one of them is the liar.

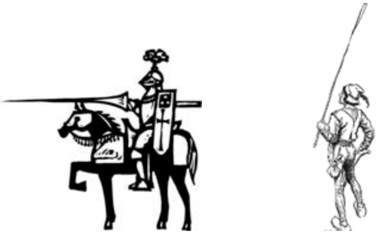
This is Jim	Jim is a liar	This is a liar	Are you Jim?		
Yes	Yes	Yes	No		
Yes	No	No	Yes		
No	Yes	No	No		
No	No	Yes	Yes		


Twins puzzle

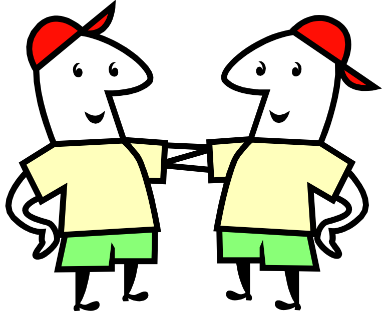


- There are two identical twin brothers, Dave and Jim.
- One of them always lies; another always tells the truth (like knights and knaves).
- Suppose you see one of them and you want to find out his name.
- How can you learn if you met Dave or Jim by asking just one short yes-no question? You don't know which one of them is the liar.

This is Jim	Jim is a liar	This is a liar	Are you Jim?	Is 2+2=4?	
Yes	Yes	Yes	No	No	
Yes	No	No	Yes	Yes	
No	Yes	No	No	Yes	
No	No	Yes	Yes	No	



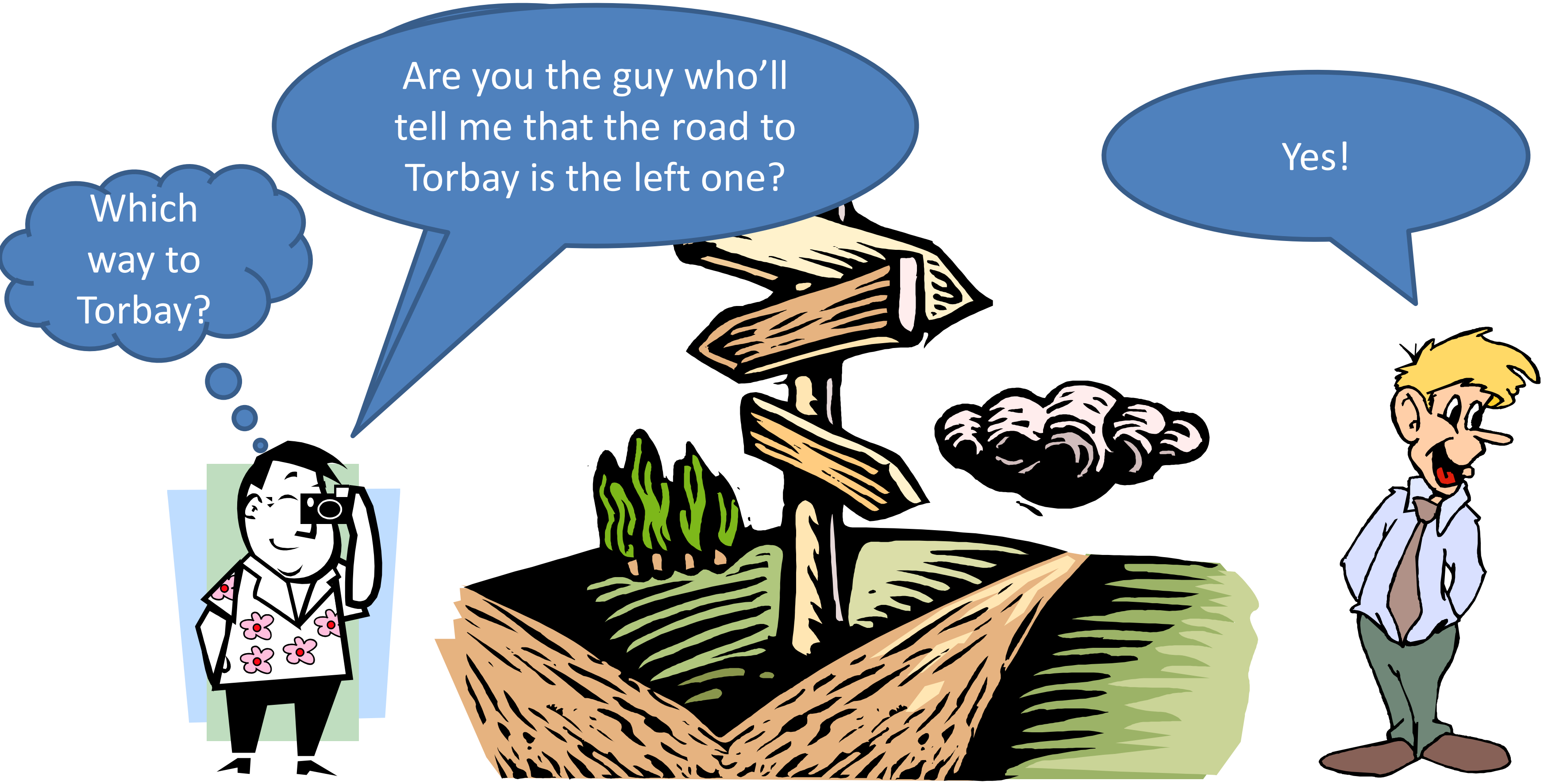
Twins puzzle



- There are two identical twin brothers, Dave and Jim.
- One of them always lies; another always tells the truth (like knights and knaves).
- Suppose you see one of them and you want to find out his name.
- How can you learn if you met Dave or Jim by asking just one short yes-no question? You don't know which one of them is the liar.

This is Jim	Jim is a liar	This is a liar	Are you Jim?	Is 2+2=4?	Is Dave a liar?
Yes	Yes	Yes	No	No	Yes
Yes	No	No	Yes	Yes	Yes
No	Yes	No	No	Yes	No
No	No	Yes	Yes	No	No

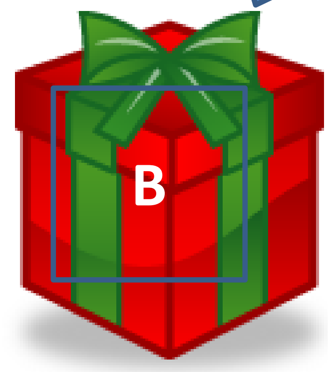
Liars on the crossroads



Treasure hunts

The label on box B is true and the gift is inbox A

The label on box B is not true and the gift is inbox A



Treasure hunt

- In the back of an old cupboard you discover a note signed by a pirate famous for his bizarre sense of humor and love of logical puzzles. In the note he wrote that he had hidden a treasure somewhere on the property. He listed 5 true statements and challenged the reader to use them to figure out the location of the treasure

Treasure hunt

1. If this house is next to a lake, then a treasure is not in the kitchen
2. If the tree in the font yard is an elm, then the treasure is in the kitchen
3. This house is next to a lake
4. The tree in the front yard is an elm, or the treasure is buried under the flagpole
5. If the tree in the back yard is an oak, then the treasure is in the garage.

Treasure hunt

1. If this house is next to a lake, then a treasure is not in the kitchen
2. If the tree in the font yard is an elm, then the treasure is in the kitchen.
3. This house is next to a lake
4. The tree in the front yard is an elm, or the treasure is buried under the flagpole
5. If the tree in the back yard is an oak, then the treasure is in the garage.

Too many variables for a nice truth table...

- A: this house is next to a lake.
- B: the treasure is in the kitchen
- C: The tree in front is elm
- D: the treasure is under the flagpole.
- E: The tree in the back is oak
- F: The treasure is in the garage

1. If A then not B
2. If C then B
3. A
4. C or D
5. If E then F

1. $A \rightarrow \sim B$
2. $C \rightarrow B$
3. A
4. $C \vee D$
5. $E \rightarrow F$

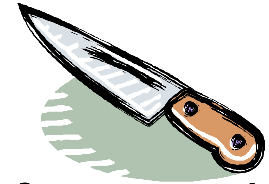
Modus ponens

- A: this house is next to a lake.
- B: the treasure is in the kitchen
- C: The tree in front is elm
- D: the treasure is under the flagpole.
- E: The tree in the back is oak
- F: The treasure is in the garage

1. If A then not B
2. If C then B
3. A
4. C or D
5. If E then F
6. Not B
7. Not C
8. D

- If house is next to the lake then the treasure is not in the kitchen
- The house is next to the lake
- Therefore, the treasure is not in the kitchen.

Murder mystery



- One evening there was a murder in the home of married couple, their son and daughter. One of these four people murdered one of the others.
- One of the members of the family witnessed the crime.
- The other one helped the murderer.

- These are the things we know for sure:
  - 1. The witness and the one who helped the murderer were not of the same sex.
  - 2. The oldest person and the witness were not of the same sex.
  - 3. The youngest person and the victim were not of the same sex.
  - 4. The one who helped the murderer was older than the victim.
  - 5. The father was the oldest member of the family.
  - 6. The murderer was not the youngest member of the family.

- Who was the murderer?

Four doors of Xanth




Four doors of Xanth

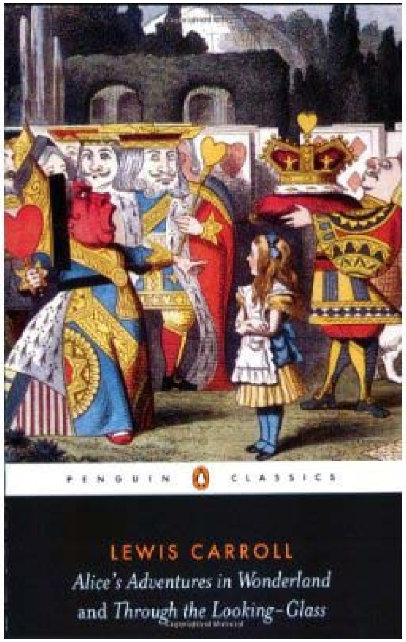
- Prince Questor is exploring the caves of Xanth. At the end of a tunnel, he finds four doors, he finds a scroll. Here is the message from the scroll. Each door conceals one item. The items are a treasure, a rope, a key, and a lantern. You must find all four items in a particular order to keep the treasure.
- As Questor is reading the scroll, three bats fly in. The first bat says, "You must find the key before you find the rope." The second bat says, "If you find the lantern before you find the rope, the treasure will disappear." The third bat says, "You must find the treasure last."
- As Questor is puzzling over these remark,s three ogres appear. The first one says, "The rope is not behind the 1st or 2nd door." The second ogre says, "The treasure is in the room just to the right of the lantern." The third ogre says, "The key is behind the fourth door." In what order should Questor open the doors to keep the treasure?

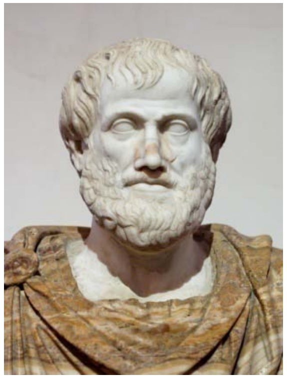
Natural deduction vs. Truth tables

- In this puzzle, it was faster to solve it using modus ponens (natural deduction method) than writing a truth table.
- But is it always better?
- The answer is...

Nobody knows!

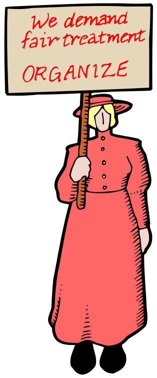
- And there is a [million dollar prize] for finding out!
- And solving this problem will solve many other.





COMP2000

Lecture 5 (last)

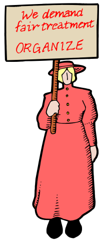


Logic: All, None, Some, and NOT

Admin stuff

- Quiz next Thursday, study guide to be posted by Monday.
- No lecture Tuesday: midterm break
- Lab questions?

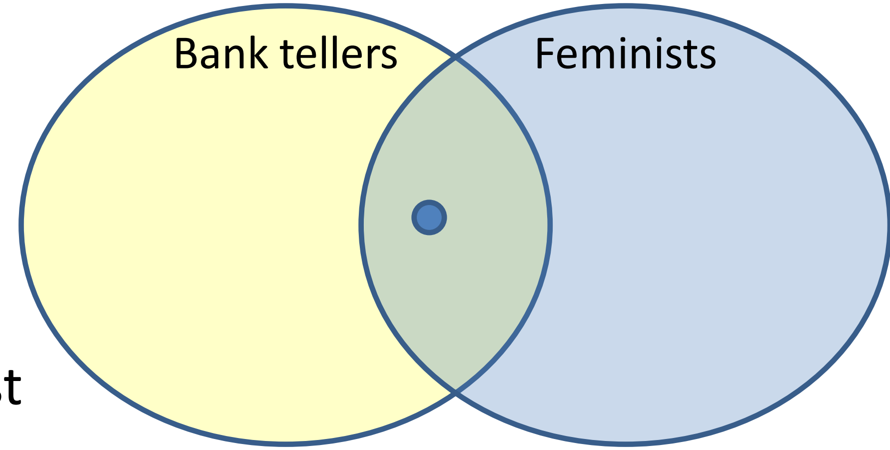
Probabilities and logic



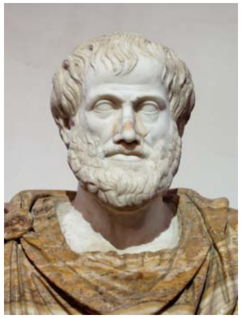
- Susan is 28 years old, single, outspoken, and very bright. She majored in philosophy. As a student she was deeply concerned with issues of discrimination and social justice and also participated in anti-nuke demonstrations.

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

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 the Sierra club
6. a bank teller
7. an insurance salesperson
8. a bank teller and an active feminist



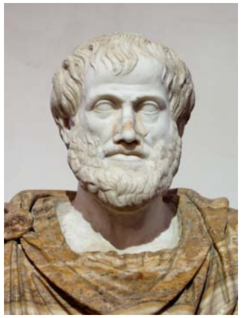
Universal Modus Ponens



- All men are mortal
- Socrates is a man
- Therefore, Socrates is mortal

- All cats like fish
- Molly likes fish
- Therefore, Molly is a cat

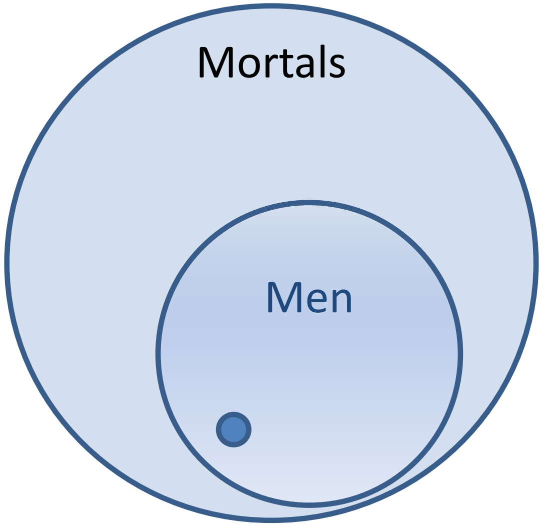
Universal Modus Ponens



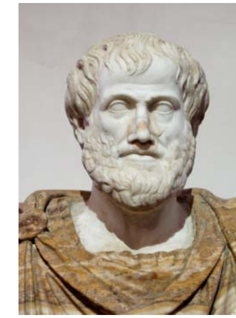
- All men are mortal
- Socrates is a man
- Therefore, Socrates is mortal

- All numbers are either odd or even
- 2 is a number
- Therefore, 2 is either odd or even.

- All trees drop leaves
- Pine does not drop leaves
- Therefore, pine is not a tree



Negating the universal

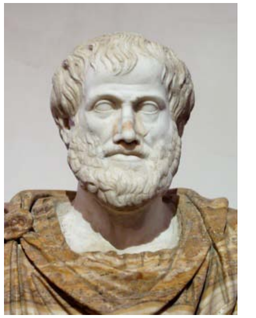


- What is the opposite of “All”?
  - All girls hate math.
  - No!
    - All girls love math?
    - Some girls do not hate math!

  - Everybody in O’Brian family is tall
    - No, Jenny is O’Brian and she is quite short.

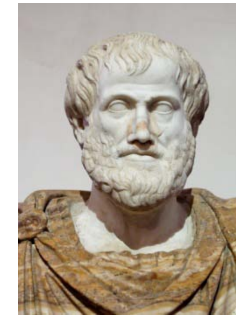
  - It is foggy all the time, every day in St. John’s
    - No, sometimes it is not foggy (like today).

Negating the universal

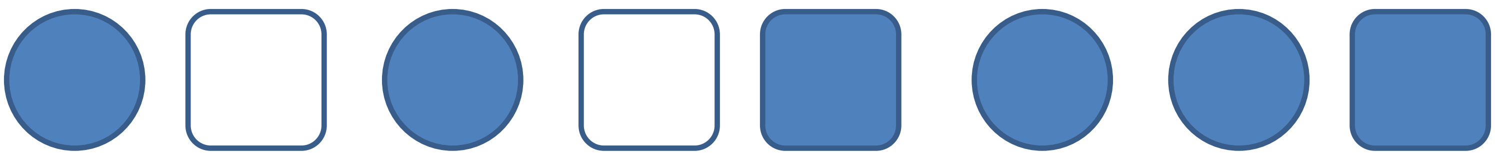


- What is the opposite of “All”? $\forall$
- The negation of “All” is “Some” $\exists$
- “Some” may include all, but not necessarily.
- “Nobody” is not a negation of “All”
  - You wouldn’t say “it is never foggy in St. John’s” when somebody tells you it is always foggy. Sometimes it is, sometimes it isn’t.
  - $\forall$ x (if x is a number, then x is even or odd.)
  - $\exists$ x (x is a number and x is prime)
- Negation of $\forall$ is $\exists$, and negation of $\exists$ is $\forall$
  - Also, negation of -> is /\

Tricky universals

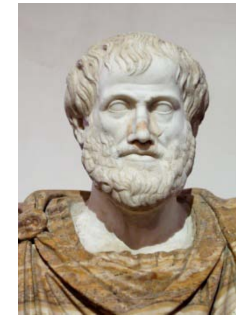


- Which statements are true?
  - All squares are white. All white shapes are squares
  - All circles are blue. All blue shapes are circles.



  - All lemurs live in the trees. All animals living in the trees are lemurs.

“NOT” makes life harder



- It is easy to visualize a tree, or a person
- It is harder to visualize a “not tree” or “not person”
- So “NOT (ALL trees have leaves)” is harder to understand than “some trees have something other than leaves (e.g., needles).
- Multiple negatives make it even harder
  - I vote against repealing the ban on smoking in public.
    - Do I like smoking in public?

Mixing quantifiers

- We can make statements of predicate logic mixing existential and universal quantifiers.
  - Predicate: X loves Y
  - Everybody loves somebody: $\forall$ X $\exists$ Y (X loves Y)
    - Normal people
  - Somebody loves everybody: $\exists$ X $\forall$ Y (X loves Y)
    - Mother Teresa
  - Everybody is loved by somebody $\forall$ X $\exists$ Y (Y loves X)
    - Their mother
  - Somebody is loved by everybody $\exists$ X $\forall$ Y (Y loves X)
    - Elvis Presley

Negating mixed quantifiers

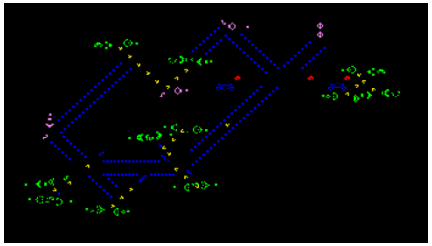
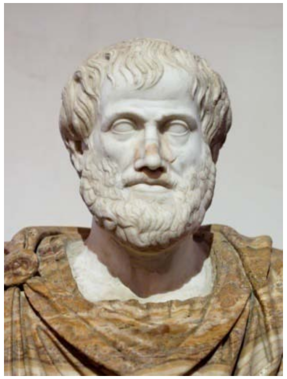
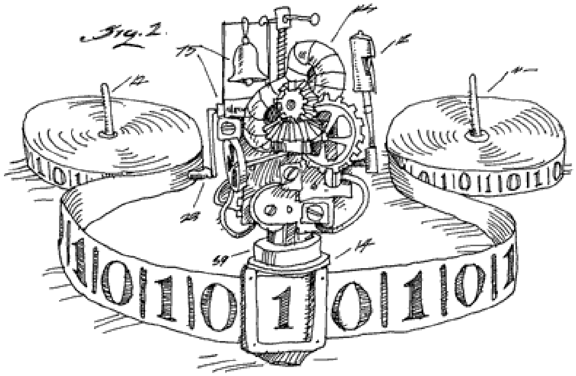
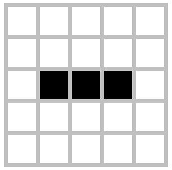
- Now, a “not” in front of such a sentence means all $\forall$ and $\exists$ are interchanged, and the inner part becomes negated.
  - Predicate: X loves Y
  - Everybody loves somebody: $\forall X \exists Y$ (X loves Y)
    - Somebody does not love anybody $\exists X \forall Y$ NOT(X loves Y)
  - Somebody loves everybody: $\exists X \forall Y$ (X loves Y)
    - Everyone doesn’t like somebody $\forall X \exists Y$ NOT(X loves Y)
  - Everybody is loved by somebody $\forall X \exists Y$ (Y loves X)
    - Somebody is not loved by anybody $\exists X \forall Y$ NOT(Y loves X)
  - Somebody is loved by everybody $\exists X \forall Y$ (Y loves X)
    - Everyone is not loved by somebody $\forall X \exists Y$ NOT(Y loves X)

Logic summary

- Propositional logic: A, B are either true or false
  - A and B: true if both are true
  - A or B: true if at least one is true
  - Not A: true if A is false
  - If A then B: true if either A is false, or both are true (equivalent to (not A or B).
- Predicate logic: Predicates are propositions with parameters: e.g., Odd(x) means number x is odd.
  - Universal quantifier: $\forall$ x Odd(x) – all x are odd.
  - Existential quantifier: $\exists$ x Odd(x) – some x is odd.

How do computers reason?

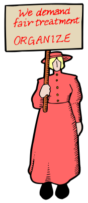
- They use logic (like one that we just studied)
  - For example, for relationships between concepts: if it is an animal, then it is alive.
- They use probabilities (which scenarios are more likely than other?)
- Fancy machine learning algorithms
- Example: IBM's "Watson" winning Jeopardy game.
- More to come...

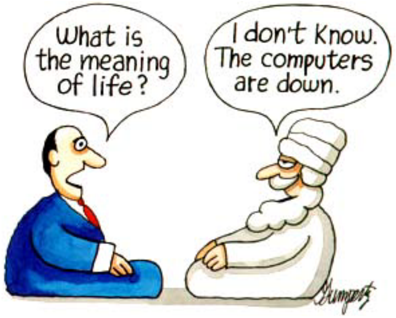






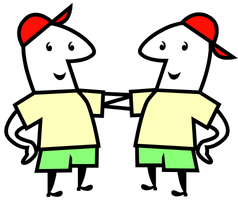
COMP2000 Quiz Notes

Computation and logic





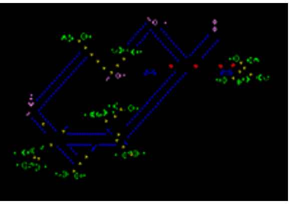





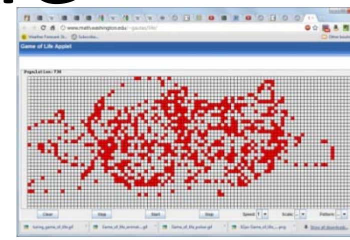
Computation: Turing machine

Current state	Reads	Writes	Moves	New state
Start_state	_	Y	Right	Halt
Start_state	0	_	Right	Start_state
Start_state	1	_	Right	Start_state
Start_state	_	N	Right	Halt

- Turing machine:
  - Can compute anything we know to be computable by other means (Church-Turing thesis)
  - Has an infinite tape with a read/write head, which starts with an input written on it. Has finitely many states in its description.
  - Simple rules: in a state, read a symbol then change a state (maybe), overwrite the symbol with another one (maybe) and move Left or Right
  - Invented by Alan Turing to show that some problems are not computable.

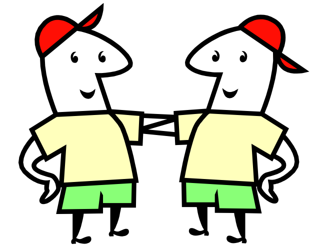


Computation: Game of Life



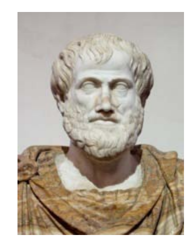
- Game of Life:
  - As powerful as a Turing machine: a Turing machine can simulate it moves, and it can simulate a Turing machine (non-trivial!)
  - Start with a board; some of the cells on the board are marked “live”. Every cell has 8 neighbours (above, below, sides and diagonal).
  - Rules:
    - If a live cell has fewer than 2, or more than 3 neighbours, it dies.
    - If a live cell has 2 or 3 neighbours, it stays alive.
    - If a dead cell has exactly 3 neighbours, it becomes live.
  - Many patterns: static, oscillating, moving


Logic: propositions

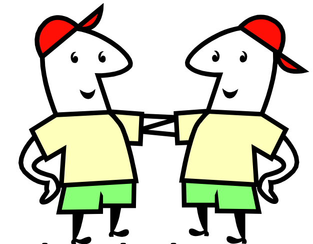


- *Propositions*: sentences that can be true or false
  - "It is Tuesday today" is true on a Tuesday and false on all other days.
  - "5 is a prime number" is true, "6 is prime" is false
  - Non-example: " x is a prime number". This is a predicate, it would become a proposition if x is set to a specific number. "Come here" is not a proposition either.
- *Logic connectives*: connect several propositions into a composite sentence that can be true or false, depending on the propositions. A sentence that is always true is called a *"tautology*", always false *"contradiction*".
- *Truth table*: a table listing all possible combinations of truth values of propositions, together with columns for composite statements. Can be used to find out when a composite statement is true and when it is false.
- Below is a truth table for the logic connectives we used:

A	B	not A	A and B	A or B	if A then B
True	*True*	False	True	True	True
True	*False*	False	False	True	False
False	*True*	True	False	True	True
False	*False*	True	False	False	True

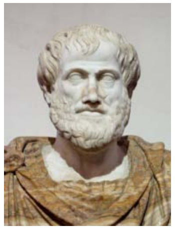


Logic: quantifiers

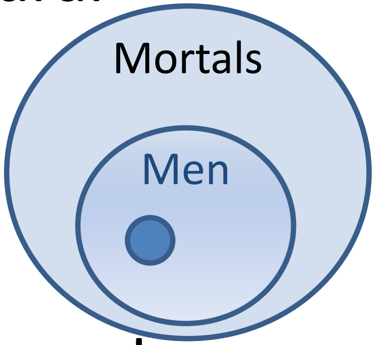


- A *predicate* is a "proposition with parameters", which becomes a proposition if the parameters are set to a specific value.
  - "x is a prime number". Prime(x)
  - "X loves Y"
- One can also make composite statements with predicates stating that for *every* or for *some* element of the kind the predicate is talking about something is true. The "every" (∀) and "some" or "exists", (∃) are called quantifiers, universal and existential.
  - Every man is mortal
  - There exists a number that is prime (some numbers are prime)
  - It is foggy every day in St. John's.
- Now, both propositional logic connectives and quantifiers can be combined to make composite statements of first-order logic.
  - Every man is an animal, and some animals have fur
  - Everybody loves somebody, and that somebody loves Elvis Presley.
    - ∀ X ∃ Y (X loves Y) and (Y loves Elvis)

Modus ponens

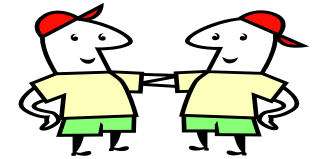


- Modus ponens is a rule used in proofs in natural deduction. Here is the general form:
  - If A then B
  - A holds
  - Therefore, B holds



- With universal quantifier, becomes universal modus ponens; second line substitutes specific value.
  - All man are mortal
  - Socrates is a man
  - Therefore, Socrates is mortal.
- Modus ponens allows solving some logic problems and puzzles faster than truth tables. But nobody knows if it is always better (there is a million dollar prize for that).


Negations



- To disprove a universal (“every”) statement, give a counterexample (“exists” statement).
  - It is foggy **every** day in St. John’s
    - No, on Monday it was sunny.
  - So “**not** (**every** day is foggy)” is **“exists** a day that was **not** foggy”.
  - But it’s definitely does not mean that every day is not foggy
  - “**not** (**exists** a day that is foggy)” is “**every** day is **not** foggy”.
- Similarly, negating an existential get universal of the “not”s:
  - There **exists** a natural number less than 1
    - No, **every** natural number is **not** less than 1.
- For logic connectives, notice that “if.. Then” is similar to “every”, and “and” is similar to “exists”:
  - **If** Socrates is a man, **then** Socrates is mortal
    - No, Socrates is a man, **and** he is immortal
  - Number 0.5 is a natural number **and** it is less than than 1
    - No, either 0.5 is **not** a natural number, **or** it is **not** less than 1.
  - Either today is Tuesday **or** today is Thursday
    - No, today is **not** Tuesday **and** today is **not** Thursday.

For the quiz:

- Be able to say what a transition of a Turing machine such as “in state A on reading 0 go to state B, write 1 and move right” does to the tape content.
- Be able to do a few generations of a game of life starting with a given pattern.
- There will be a few yes/no questions (e.g., is it true that everything we know to be computable is computable by a Turing machine?)

For the quiz

- Practice solving knight and knave puzzles and (simple) treasure hunts. Use truth tables and modus ponens, where appropriate.
- Make sure that you can recognize those fallacies we played with (which card to turn over? If I like blue triangle, what about yellow circle? Is Susan more likely to be a banker or a banker-activist?)
- Practice translating between logic and English, and identifying propositions and quantifiers in English sentences.
- Practice doing negations. See some new slides in Lecture 3, and last lecture.
- Email me if you have any questions! I will be around on Wednesday afternoon, if you'd like to stop by.