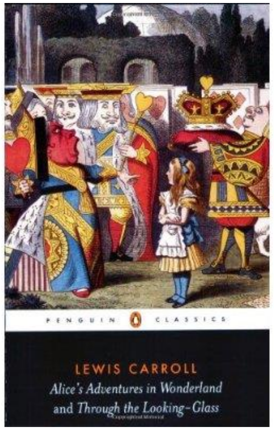
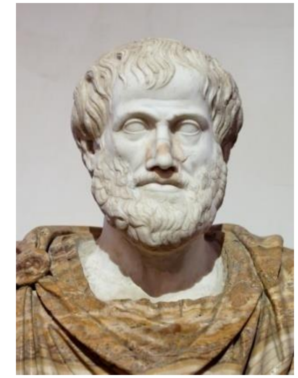




COMP 1002

Intro to Logic for Computer Scientists

Lecture 2

B 5 2 J

Language of logic: building blocks

- **Proposition**: A sentence that can be *true* or *false.*
  - A: “It is raining in St. John’s right now”.
  - B: “2+2=7”
  - But not “Hi!” or “what is x?”
- **Propositional variables:**
  - A, B, C (or p, q, r)
  - It is a shorthand to denote propositions:
    - “B is true”, for the B above, means “2+2=7” is true.

Language of logic: connectives

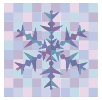
Pronunciation	Notation	Meaning
Not A (negation)	$\neg$ A	Opposite of A is true, $\neg$A is true when A is false
A and B (conjunction)	A $\wedge$ B	True if both A and B are true
A or B (disjunction)	A $\vee$ B	True if either A or B are true (or both)
If A then B (implication)	A $\rightarrow$ B	True whenever if A is true, then B is also true

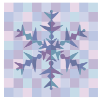
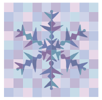
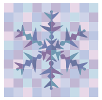
- Let A be "It is sunny" and B be "it is cold"
  - $\neg$ A: It is not sunny
  - A $\wedge$ B: It is sunny and cold
  - A $\vee$ B: It is either sunny or cold
  - A $\rightarrow$ B: If it is sunny, then it is cold

- Can build longer formulas by combining smaller formulas using connectives.
  - And parentheses: will see later when can remove them.

 Longer formulas 

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:
  - $(B \land C) \rightarrow (\neg A)$ IF ( AND ) THEN NOT 
    - If it is cold and snowing, then it is not sunny
  - $B \rightarrow (C \lor A)$ IF  THEN ( OR )
    - If it is cold, then it is either snowing or sunny
  - $((\neg A) \land A) \rightarrow C$ 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, or whether it is cold, then we would not know whether A ∧ B is true or false.

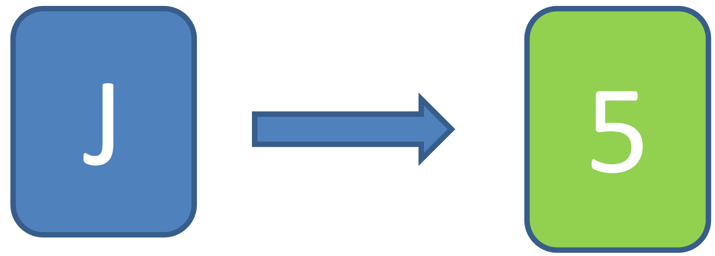
- **Truth assignment:** setting values of all propositional variables to true or false.
  - It is sunny, it is not cold, it is snowing
  - A=true, B=false, C= true

- **Satisfying assignment** for a sentence: assignment that makes it true.
  - (Otherwise, **falsifying** assignment).
  - A=true, B=true satisfies A ∧ B
  - A=true, B=false falsifies A ∧ B

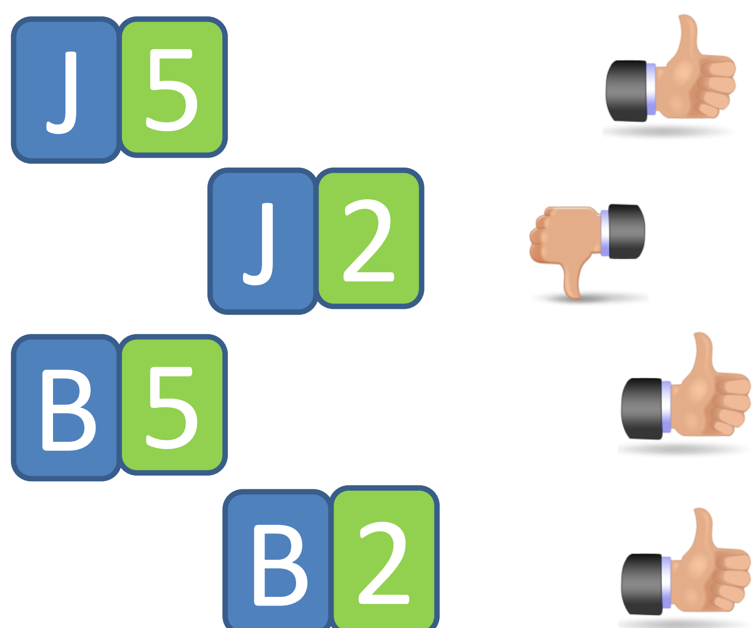
"if ... then" in logic

- Last class' puzzle has a logical 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

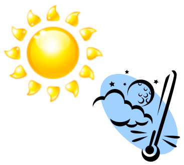


Truth tables

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"



- When it is true that it is sunny and true that it is cold, it is true that it is either sunny or cold.
  - When A is true, B is true, then $A \lor B$ is true
- When it is true that it is sunny, but false that it is cold, then "it is both sunny and cold" is false.
  - When A is true, B is false, then $A \land B$ is false.

Order of precedence

Pronunciation	Notation	True when
Not A	¬ A	Opposite of A is true
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

- Combine to make longer formulas
- In arithmetic:
  - 6 + -5 * 7 + 8 = (6+ ((-5)* x)) + 8
    - First negate 5, then multiply -5 and x, then add this to 6, add result to 8.
    - Order: unary -, then *, then +.
- In logic formulas
  1. Negation (¬) first: like unary minus
  2. then AND (∧): like times (*)
  3. then OR (∨): like plus (+)
  4. Only then "if … then" (→)
  5. Parentheses as in arithmetic formulas

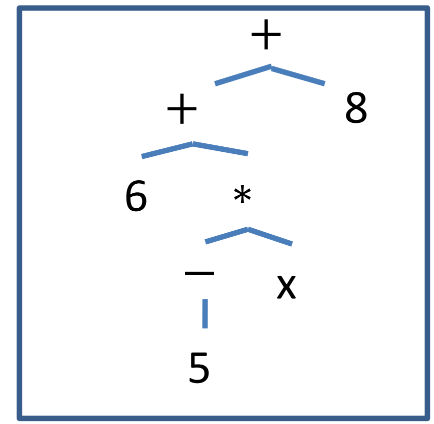
  $A \wedge \neg B \quad \vee \quad \neg(C \to A) \quad \to A$ is $\big((A \wedge (\neg B)) \vee (\neg(C \to A))\big) \to A$

- Make sure to keep track of parentheses!!!
  - $A \vee B \wedge C$ is not the same as $(A \vee B) \wedge C$, just like $2 + 3 * 4 \neq (2 + 3) * 4$
    - Check the scenario when A is true, but both B and C are false.

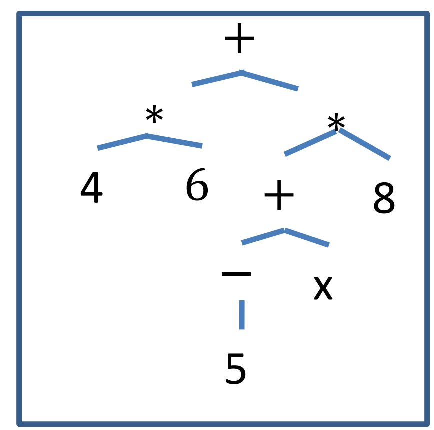
Parse trees

- Order of precedence in arithmetic:
  – First unary minus (-), then multiplication (*), then addition (+)

- Parse tree: a way to visualize the order of precedence
  – The last operation on top
  – The numbers/variables on the bottom.
  – Branch points (nodes) marked by arithmetic operations.
  – To compute a value at a branch point (node), compute all values below it.



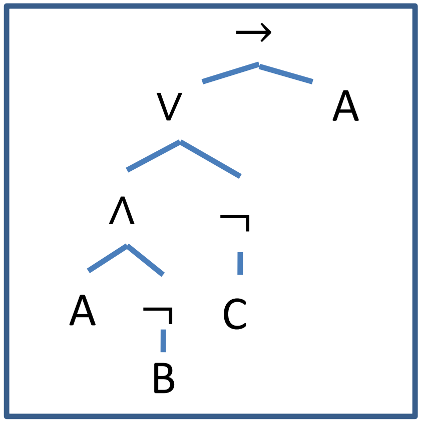
6 + -5 * x + 8 = (6+ ((-5)* x)) + 8



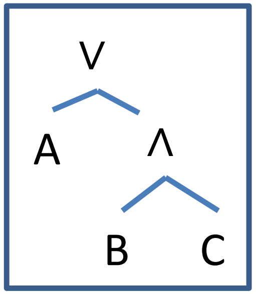
4* 6 + (-5 + x) * 8 = (4*6)+ (((-5)+ x)* 8)

Parse trees in logic formulas

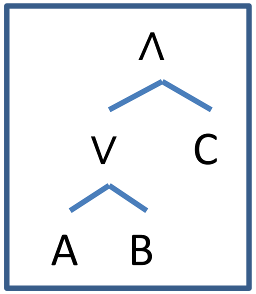
- Precedence:
  - $\neg$ first, then $\wedge$, then $\vee$, $\rightarrow$ last
  - $\neg$ is like a unary minus, $\wedge$ like * and $\vee$ like +
- Parse tree: a way to visualize the order
  - The last operation on top
  - Variables on the bottom.
  - Branch points marked by connectives.
- $A \wedge \neg B \vee \neg C \rightarrow A$
  - Which is $\big((A \wedge (\neg B)) \vee (\neg C)\big) \rightarrow A$
- $A \vee B \wedge C$ is $A \vee (B \wedge C)$
  - which is different from $(A \vee B) \wedge C$



$A \wedge \neg B \vee \neg C \rightarrow A$



$A \vee (B \wedge C)$

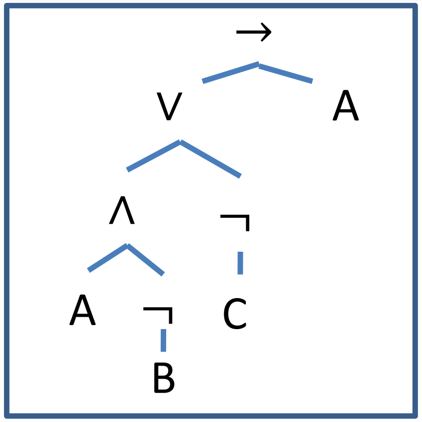


$(A \vee B) \wedge C$

Evaluating logic formulas

Pronunciation	Notation	True when
Not A	¬ A	Opposite of A is true
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

- Precedence:
  - ¬ first, then ∧, then ∨, → last
  - ¬ is like a unary minus, ∧ like * and ∨ like +



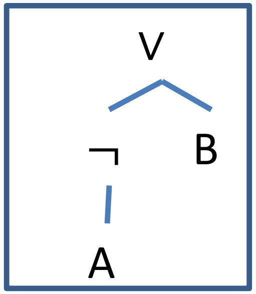
- Like in arithmetic:
  - evaluate bottom-up,
  - Start with variables, then go up the tree computing the value of each node.

- Example:
  - Let A be false, B be true and C be false.
  - Evaluate $A \land \neg B \lor \neg C \to A$
    - (which is $\left(\left(A \land (\neg B)\right) \lor (\neg C)\right) \to A)$

1. $\neg B$ is false
   - Negation of true is false
2. $A \land (\neg B)$ is false
   - False and true is false
3. $\neg C$ is true
4. $(A \land (\neg B)) \lor (\neg C)$ is true
   - "False or true" is true
5. $\left(A \land (\neg B)\right) \lor (\neg C) \to A$ is false
   - "If G then H" is false when G is true and H is false

Truth tables for longer formulas

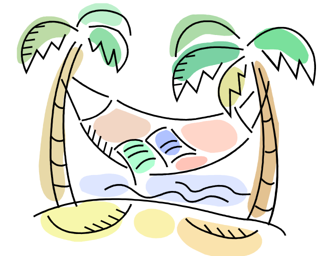
- Longer formulas
  - List all possible combinations of assigning TRUE/FALSE to all variables.
  - For each of them, find the value of the whole formula by evaluating bottom-up like in arithmetic
    - Use the parse tree.



- Example: $(\neg A) \vee B$
  - Notice: its truth table's last column is the same as for $\mathrm{A} \rightarrow \mathrm{B}$
  - So $(\neg A) \vee B$ and $\mathrm{A} \rightarrow \mathrm{B}$
  - are **equivalent.**

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

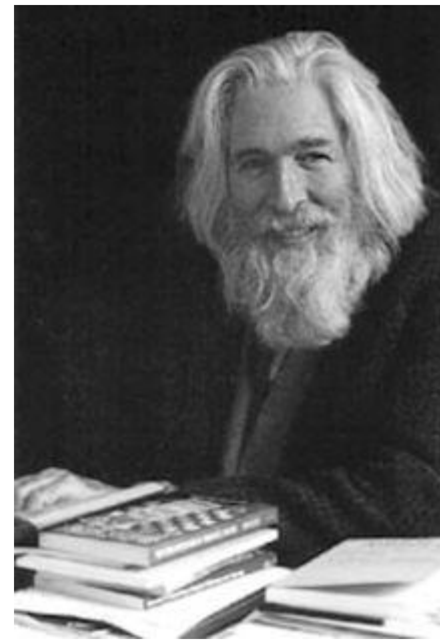
Knights and knaves



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

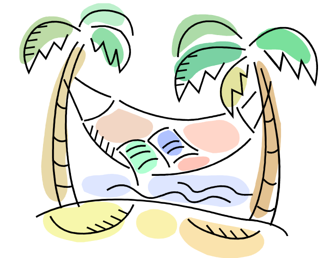
- Knights always tell the truth.

- Knaves always lie.



Raymond Smullyan


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?