WEEK2

# DISCRETE STRCUTURES

# Agenda

- Logic
- Propositions
- Compound propositions
- Logical Connectives

# Logic

We intuitively know that Truth and Falsehood are opposites. That statements describe the world and can be true/false. That the world is made up of objects and that objects can be organized to form collections.

The foundations of logic mimic our intuitions by setting down constructs that behave analogously.

# Logic

Axiomatic concepts in mathematics:

- Equals
- Opposite
- Truth and falsehood
- Statement
- Objects
- Collections

# False, True, Statements

Axiom: ***False*** is the opposite to ***Truth***.

A ***statement*** is a description of something.

Examples of statements:

- I'm 39 years old.
- I have 17 children.
- I always tell the truth.
- I'm lying to you.

Q's: Which statements are True? False? Both? Neither?

# False, True, Statements

True:  I’m 37 years old.

False: I have 17 children.

I always tell the truth.

Both: IMPOSSIBLE, by our Axiom.

# False, True, Statements

Neither: I'm lying to you. (*If viewed on its own*)

HUH? Well suppose that

$$S = \text{“I'm lying to you.”}$$

were **true**. In particular, I am actually lying, so *S* is false. So it's both **true and false**, impossible by the Axiom.

Okay, so I guess *S* must be **false**. But then I must not be lying to you. So the statement is true. Again it's both **true and false**.

In both cases we get the opposite of our assumption, so *S* is neither true nor false.

# Propositions

To avoid painful head-aches, we ban such silly non-sense and avoid the most general type of statements limiting ourselves to statements with valid truth-values instead:

DEF: A ***proposition*** is a statement that is true or false.

# Propositions

Propositional Logic is a *static* discipline of statements which lack *semantic content*.

E.G. $p$ = “Clinton was the president.”

$q$ = “The list of U.S. presidents includes Clinton.”

$r$ = “Lions like to sleep.”

All $p$ and $q$ are no more closely related than $q$ and $r$ are, in propositional calculus. They are both equally related as all three statements are true. Semantically, however, $p$ and $q$ are the same!

# Propositions

So why waste time on such matters?

Propositional logic is the study of how simple propositions can come together to make more complicated propositions. If the simple propositions were endowed with some meaning – *and they will be very soon*– then the complicated proposition would have meaning as well, and then finding out the truth value is actually important!

# Compound Propositions

In Propositional Logic, we assume a collection of *atomic* propositions are given: *p, q, r, s, t,* ….

Then we form compound propositions by using ***logical connectives*** (***logical operators***) to form propositional "molecules".

# Logical Connectives

| Operator | Symbol | Usage | Java |
|---|---|---|---|
| Negation | $\neg$ | not | `!` |
| Conjunction | $\wedge$ | and | `&&` |
| Disjunction | $\vee$ | or | `\|\|` |
| Exclusive or | $\oplus$ | xor | `(p\|\|q)&&(!p\|\|!q)` |
| Conditional | $\rightarrow$ | if,then | `p?q:true` |
| Biconditional | $\leftrightarrow$ | iff | `(p&&q)\|\|(!p&&!q)` |

# Compound Propositions: Examples

$p$ = "Cruise ships only go on big rivers."

$q$ = "Cruise ships go on the Hudson."

$r$ = "The Hudson is a big river."

$\neg r$ = "The Hudson is not a big river."

$p \wedge q$ = "Cruise ships only go on big rivers and go on the Hudson."

$p \wedge q \rightarrow r$ = "If cruise ships only go on big rivers and go on the Hudson, then the Hudson is a big river."

# Negation

This just turns a false proposition to true and the opposite for a true proposition.

EG: $p$ = "23 = 15 +7"

$p$ happens to be false, so $\neg p$ is true.

In Java, "!" plays the same role:

```
!(23 == 15+7)
```

has the `boolean` value `true` whenever evaluated.

# Negation – truth table

Logical operators are defined by **truth tables** – tables which give the output of the operator in the right-most column.

Here is the truth table for negation:

| $p$ | $\neg p$ |
|---|---|
| F | T |
| T | F |

# Conjunction

Conjunction is a ***binary*** operator in that it operates on two propositions when creating compound proposition. On the other hand, negation is a ***unary*** operator (the only non-trivial one possible).

# Conjunction

Conjunction is supposed to encapsulate what happens when we use the word "and" in English. I.e., for "$p$ and $q$" to be true, it must be the case that **BOTH** $p$ is true, as well as $q$. If one of these is false, than the compound statement is false as well.

# Conjunction

EG. $p$ = “Clinton was the president.”

$q$ = “Monica was the president.”

r = “The meaning of *is* is important.”

Assuming $p$ and $r$ are true, while $q$ false.

Out of $p \wedge q$, $p \wedge r$, $q \wedge r$

only $p \wedge r$ is **true**.

Java: `x==3 && x!=3`

Evaluates to `false` for *any possible value* of `x`.

# Conjunction – truth table

| $p$ | $q$ | $p \wedge q$ |
|---|---|---|
| T | T | T |
| T | F | F |
| F | T | F |
| F | F | F |

# Disjunction – truth table

Conversely, disjunction is true when at least one of the components is true:

| $p$ | $q$ | $p \vee q$ |
|---|---|---|
| T | T | T |
| T | F | T |
| F | T | T |
| F | F | F |

# Disjunction – caveat

Note: English version of disjunction "or" does not always satisfy the assumption that one of *p*/*q* being true implies that "*p* or *q* " is true.

Q: Can someone come up with an example?

# Disjunction – caveat

A: The entrée is served with

soup **or** salad.

Most restaurants definitely don't allow you to get *both* soup *and* salad so that the statement is false when both soup and salad is served. To address this situation, exclusive-or is introduced next.

# Exclusive-Or – truth table

| $p$ | $q$ | $p \oplus q$ |
| --- | --- | --- |
| T | T | F |
| T | F | T |
| F | T | T |
| F | F | F |

Note: in this course any usage of "or" will connote the logical operator $\vee$ as opposed to the exclusive-or.

# Conditional (Implication)

This one is probably the least intuitive. It's only partly akin to the English usage of "if,then" or "implies".

DEF: $p \rightarrow q$ is true if $q$ is true, or if $p$ is false. In the final case ($p$ is true while $q$ is false) $p \rightarrow q$ is false.

Semantics: "$p$ implies $q$" is true if one can mathematically derive $q$ from $p$.

# Conditional -- truth table

| $p$ | $q$ | $p \rightarrow q$ |
|---|---|---|
| T | T | T |
| T | F | F |
| F | T | T |
| F | F | T |

# Conditional

Q: Does this makes sense? Let's try examples for each row of truth table:

1. If pigs like mud then pigs like mud.
2. If pigs like mud then pigs can fly.
3. If pigs can fly then pigs like mud.
4. If pigs can fly then pigs can fly.

# Conditional

A:

1. If pigs like mud then pigs like mud.

   True: nothing about this statement is false.

2. If pigs like mud then pigs can fly.

   False: seems to assert falsehood

3. If pigs can fly then pigs like mud.

   True: argument for –only care about end-result.

   Argument against –counters common English hyperbole.

4. If pigs can fly then pigs can fly.

   True. WAIT! By first reasoning in 3, when "if" part is false, should only care about "then" part!!!!!

   On other hand, standard English hyperbole.

# Conditional: why F$\rightarrow$F is True

Remember, all of these are mathematical constructs, not attempts to mimic English. Mathematically, *p* should imply *q* whenever it is possible to derive *q* by from *p* by using valid arguments. For example consider the mathematical analog of no. 4:

$$\text{If } 0 = 1 \text{ then } 3 = 9.$$

Q: Is this true mathematically?

# Conditional: why $F \rightarrow F$ is True

A: YES mathematically and YES by the truth table.

Here's a mathematical proof:

1. 0 = 1 (assumption)

# Conditional: why $F \rightarrow F$ is True

A: YES mathematically and YES by the truth table.

Here's a mathematical proof:

1. $0 = 1$ (assumption)
2. $1 = 2$ (added 1 to both sides)

# Conditional: why F$\rightarrow$F is True

A: YES mathematically and YES by the truth table.

Here's a mathematical proof:

1. 0 = 1 (assumption)
2. 1 = 2 (added 1 to both sides)
3. 3 = 6 (multiplied both sides by 3)

# Conditional: why F→F is True

A: YES mathematically and YES by the truth table.

Here's a mathematical proof:

1. 0 = 1 (assumption)
2. 1 = 2 (added 1 to both sides)
3. 3 = 6 (multiplied both sides by 3)
4. 0 = 3 (multiplied no. 1 by 3)

# Conditional: why F→F is True

A: YES mathematically and YES by the truth table.

Here's a mathematical proof:

1. 0 = 1 (assumption)
2. 1 = 2 (added 1 to both sides)
3. 3 = 6 (multiplied both sides by 3)
4. 0 = 3 (multiplied no. 1 by 3)
5. 3 = 9 (added no. 3 and no. 4) *QED*

# Conditional: why F$\rightarrow$F is True

As we want the conditional to make sense in the semantic context of mathematics, we better define it as we have!

Other questionable rows of the truth table can also be justified in a similar manner.

# Conditional: synonyms

There are many ways to express the conditional statement $p \rightarrow q$ :

If *p* then *q.* *p* implies *q.* If *p, q.*

*p* only if *q.* *p* is sufficient for *q*.

Some of the ways **reverse** the order of *p* and *q* but have the same connotation:

*q* if *p.* *q* whenever *p.* *q* is necessary for *p.*

To aid in remembering these, I suggest inserting "is true" after every variable:

EG: "*p* is true only if *q* is true"

# Bi-Conditional -- truth table

For $p \leftrightarrow q$ to be true, $p$ and $q$ must have the same truth value. Else, $p \leftrightarrow q$ *is false:*

| $p$ | $q$ | $p \leftrightarrow q$ |
|---|---|---|
| T | T | T |
| T | F | F |
| F | T | F |
| F | F | T |

Q : Which operator is $\leftrightarrow$ the opposite of?

# Bi-Conditional

A : $\leftrightarrow$ has exactly the opposite truth table as $\oplus$.

This means that we could have defined the bi-conditional in terms of other previously defined symbols, so it is redundant. In fact, only really need negation and disjunction to define everything else.

Extra operators are for convenience.

Q: Could we define all other logical operations using only negation and exclusive or?

# Bi-Conditional

A: No. Notice that negation and exclusive-or each maintain parity between truth and false: No matter what combination of these symbols, impossible to get a truth table with four output rows consisting of 3 T's and 1 F (such as implication and disjuNction).

# Bit Strings

Electronic computers achieve their calculations inside semiconducting materials. For reliability, only two stable voltage states are used and so the most fundamental operations are carried out by switching voltages between these two stable states.

In logic, only two truth values are allowed. Thus propositional logic is ideal for modeling computers. High voltage values are modeled by True, which for brevity we call the number 1, while low voltage values are modeled by False or 0.

# Bit Strings

Thus voltage memory stored in a computer can be represented by a sequence of 0’s and 1’s such as

01 1011 0010 1001

Another portion of the memory might look like

10 0010 1111 1001

Each of the number in the sequence is called a ***bit***, and the whole sequence of bits is called a ***bit string***.

# Bit Strings

It turns out that the analogs of the logical operations can be carried out quite easily inside the computer, one bit at a time. This can then be transferred to whole bit strings. For example, the exclusive-or of the previous bit strings is:

$$
\begin{array}{rr}
 & 01\ 1011\ 0010\ 1001 \\
\oplus & \underline{10\ 0010\ 1111\ 1001} \\
 & 11\ 1001\ 1101\ 0000
\end{array}
$$

# Exercises

1. $q$ = “You miss the final exam.”
   $r$ = “You pass the course.”
   Express $q \rightarrow \neg r$ in English.

1. Construct a truth table for $\neg p \oplus \neg q$.
2. Can one determine relative salaries of B (Bello), R (Raheem) and A (Aisha) from the following?
   a. If B is not highest paid, then A is.
   b. If A is not lowest paid, then R is highest paid.