Formal Methods in Software Engineering

LECTURE # 3,4

Recap

Formal Methods

Rigorous mathematically-based techniques and tools for the specification, development, and verification of software and hardware systems.

Software Development

> The process by which user needs are translated into a software product. This involves translating user needs into software requirements, transforming the software requirements into design, implementing the design in code, and testing the code[1].

Propositional Formula

- In propositional logic, a propositional formula is a type of syntactic formula which is well formed and has a truth value. If the values of all variables in a propositional formula are given, it determines a unique truth value. A propositional formula may also be called a propositional expression, a sentence, or a sentential formula.

Propositional Formula

- A propositional formula is constructed from simple propositions, such as "five is greater than three" or propositional variables such as P and Q, using connectives such as NOT, AND, OR, or IMPLIES; for example:
  - (P AND NOT Q) IMPLIES (P OR Q).

Formalize Natural Language

- It is the case that P:
  P
- It is not the case that P:
  $\neg P$
- P and Q. P but Q. Although P, Q:
  $(P \wedge Q)$
- P or Q. Either P or Q:
  $(P \vee Q)$
- P if and only if Q:
  $(P \leftrightarrow Q)$
- If P, then Q:
  $(P \rightarrow Q)$
- P if Q:
  $(Q \rightarrow P)$
- Q only if P:
  $(Q \rightarrow P)$
- P just in case Q:
  $(P \leftrightarrow Q)$
- Neither P nor Q:
  $\neg(P \vee Q)$
- It is not the case that both P and Q:
  $\neg(P \wedge Q)$
- Both not P and not Q:
  $(\neg P \wedge \neg Q)$
- P is necessary for Q:
  $(Q \rightarrow P)$
- P is sufficient for Q:
  $(P \rightarrow Q)$

Formalize Natural Language

- P is both necessary and sufficient for Q:
  $(P \leftrightarrow Q)$
- P unless Q:
  $(P \vee Q)$
- Among P and Q, only P:
  $(P \wedge \neg Q)$
- Among P and Q, not P:
  $(\neg P \wedge Q)$
- Only one among P and Q:
  $(\neg P \wedge Q) \vee (P \wedge \neg Q)$
- At most one among P and Q:
  $\neg(P \wedge Q)$
- At least one among P and Q:
  $(P \vee Q)$

Formalize Natural Language

Let's consider a propositional language where *p* means *"Paola is happy"*, *q* means *"Paola paints a picture"*, and *r* means *"Renzo is happy"*. Formalize the following sentences:

- "if Paola is happy and paints a picture then Renzo isn't happy"

  $(p \wedge q) \rightarrow \neg r$

- "if Paola is happy, then she paints a picture"

  $p \rightarrow q$

- "Paola is happy only if she paints a picture"

  $\neg(p \wedge \neg q)$ which is equivalent to $p \rightarrow q$!!!

Formalize Natural Language

Exercise

Let A ="Angelo comes to the party", B ="Bruno comes to the party", C ="Carlo comes to the party", and D ="Davide comes to the party".
Formalize the following sentences:

1. *"If Davide comes to the party then Bruno and Carlo come too"*
2. *"Carlo comes to the party only if Angelo and Bruno do not come"*
3. *"If Davide comes to the party, then, if Carlo doesn't come then Angelo comes"*
4. *"Carlo comes to the party provided that Davide doesn't come, but, if Davide comes, then Bruno doesn't come"*
5. *"A necessary condition for Angelo coming to the party, is that, if Bruno and Carlo aren't coming, Davide comes"*
6. *"Angelo, Bruno and Carlo come to the party if and only if Davide doesn't come, but, if neither Angelo nor Bruno come, then Davide comes only if Carlo comes"*

Formalize Natural Language

"If Davide comes to the party then Bruno and Carlo come too"

$$D \rightarrow (B \wedge C)$$

"Carlo comes to the party only if Angelo and Bruno do not come"

$$C \rightarrow (\neg A \wedge \neg B)$$

"If Davide comes to the party, then, if Carlo doesn't come then Angelo comes"

$$D \rightarrow (\neg C \rightarrow A)$$

"Carlo comes to the party provided that Davide doesn't come, but, if Davide comes, then Bruno doesn't come"

$$(C \rightarrow \neg D) \wedge (D \rightarrow \neg B)$$

"A necessary condition for Angelo coming to the party, is that, if Bruno and Carlo aren't coming, Davide comes"

$$A \rightarrow (\neg B \wedge \neg C \rightarrow D)$$

"Angelo, Bruno and Carlo come to the party if and only if Davide doesn't come, but, if neither Angelo nor Bruno come, then Davide comes only if Carlo comes"

$$(A \wedge B \wedge C \leftrightarrow \neg D) \wedge (\neg A \wedge \neg B \rightarrow (D \rightarrow C))$$

Tautology (1)

- A **tautology** is a statement that is always true, no matter what.
- If you construct a truth table for a statement and all of the column values for the statement are true (T), then the statement is a tautology because it's always true!

Tautology (2)

- If you are given a statement and want to determine if it is a tautology, then all you need to do is construct a truth table for the statement and look at the truth values in the final column. If all of the values are T (for true), then the statement is a tautology.

Tautology (3)

- 'I will either get paid or not get paid‘
- we can use *p* to represent the statement 'I will get paid' and
- ¬*p* to represent 'I will not get paid.‘ or it is not the case that I will get paid
- It concludes to following form
- *p* V ¬*p*

Tautology (4)

- Truth table f

Truth Table for p∨ ¬ p		
p	¬ p	p∨ ¬ p
T	F	T
F	T	T

Contradiction (1)

- A compound statement is a contradiction if it is false regardless of the truth values assigned to it

Contingency (1)

- A proposition that is neither a tautology nor a contradiction is called a **contingency**
- A relation that have both T & F values

Problem

Formalize the following arguments and verify whether they are correct: *"If you play and you study you'll pass the exams, while if you play and don't study you won't pass. Thus, if you play, either you study and you'll pass the exams, or you don't study and you won't pass."*

P = "you play" , S = "you study" ; E = "you pass the exam"

1.$(P \wedge S) \rightarrow E$
2.$(P \wedge \neg S) \rightarrow \neg E$
3.$P \rightarrow (S \wedge E) \vee (\neg S \wedge \neg E)$

We need to prove that $1 \wedge 2 \models 3$
Use truth tables

Revision

What is Boolean Algebra?

Boolean Algebra is a branch of algebra that involves bools, or true and false values. They're typically denoted as ***T or 1 for true*** and ***F or 0 for false***. Using this simple system we can boil down complex statements into digestible logical formulas.

Revision

Negation

The **negation operator** is commonly represented by a tilde (~) or ¬ symbol. It negates, or switches, something's truth value.

We can show this relationship in a truth table. A **truth table** is a way of organizing information to list out all possible scenarios.

We title the first column p for proposition. In the second column we apply the operator to p, in this case it's ~p (read: not p). So as you can see if our premise begins as True and we negate it, we obtain False, and vice versa.

p	~p
T	F
F	T

or

p	~p
1	0
0	1

Revision

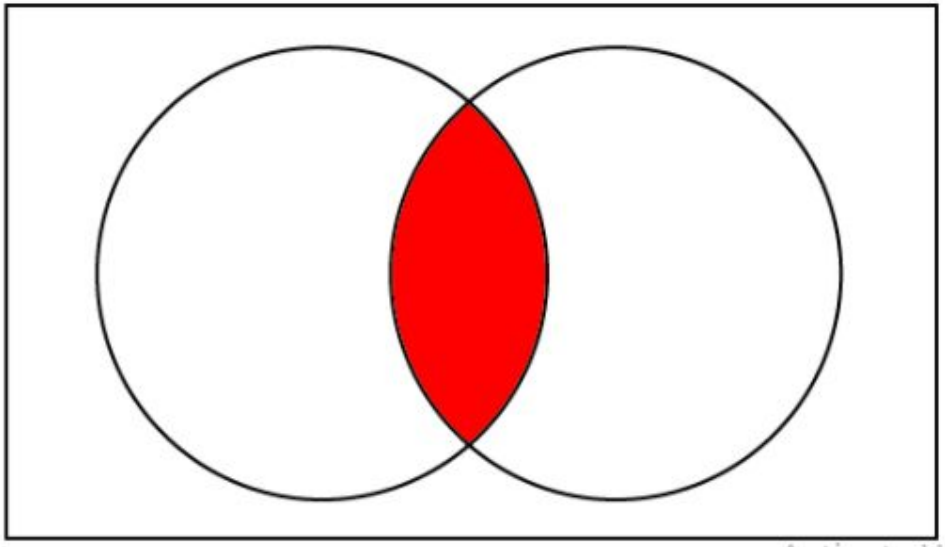
Binary Operators

Binary operators require two propositions. We'll use p and q as our sample propositions.

AND

The **AND operator** (symbolically: ∧) also known as **logical conjunction** *requires* both p and q to be True for the result to be True. All other cases result in False. This is logically the same as the intersection of two sets in a Venn Diagram.

p	q	p∧q
T	T	T
T	F	F
F	T	F
F	F	F

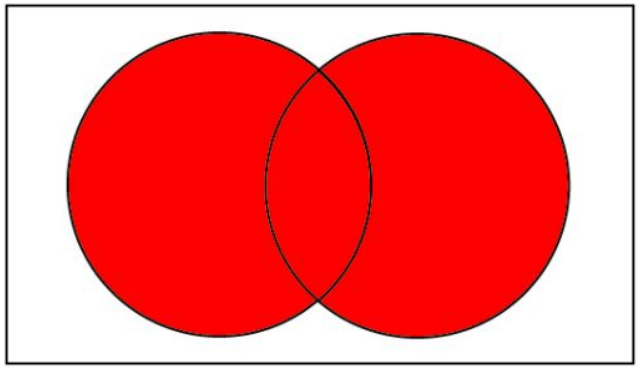


Revision

OR

The **OR operator** (symbolically: ∨) requires only one premise to be True for the result to be True. This is equivalent to the union of two sets in a Venn Diagram.

p	q	p∨q
T	T	T
T	F	T
F	T	T
F	F	F



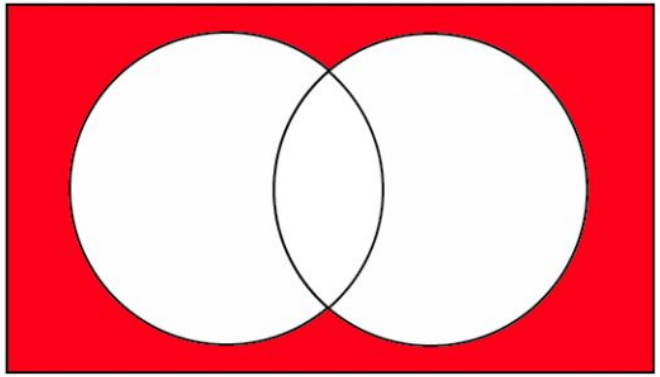
Truth Table for "OR" can be thought of as a Venn Diagram Union

Revision

NOR

Logical **NOR** (symbolically: ↓) is the exact opposite of OR. It requires both p and q to be False to result in True.

p	q	p ↓ q
T	T	F
T	F	F
F	T	F
F	F	T



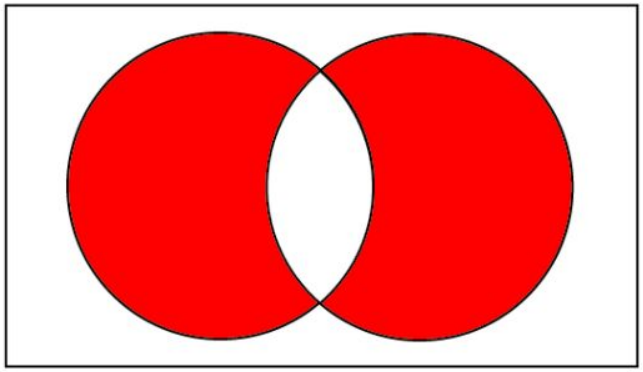
Truth Table for "NOR" can be thought of as the opposite of a Venn Diagram Union

XOR

XOR

Exclusive Or, or **XOR** for short, (symbolically: $\veebar$) requires exactly one True and one False value in order to result in True.

p	q	$p \veebar q$
T	T	F
T	F	T
F	T	T
F	F	F



Implication Truthtable

p	q	$p \rightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T

Equivalence Truthtable

p	q	$p \leftrightarrow q$
T	T	T
T	F	F
F	T	F
F	F	T

Class Activity

1. $p \leftrightarrow p$
2. $p \leftrightarrow \sim(\sim p)$
3. $[\sim(p \vee q)] \leftrightarrow [(\sim p) \wedge (\sim q)]$
4. $[\sim(p \wedge q)] \leftrightarrow [(\sim p) \vee (\sim q)]$
5. $[\sim(p \rightarrow q)] \leftrightarrow [p \wedge (\sim q)]$

1. $(p \rightarrow q) \wedge (p \wedge \sim q)$
2. $[(p \vee q) \wedge \sim p] \wedge \sim q$
3. $(p \wedge q) \wedge \sim p$