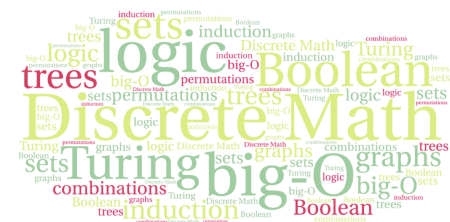# CS 220: Discrete Structures and their Applications

Propositional Logic

Sections 1.1-1.2 in zybooks

Logic in computer science

Used in many areas of computer science:

- ✓ Booleans and Boolean expressions in programs

- ✓ Reasoning about programs, proving program correctness

- ✓ Artificial intelligence (AI)

- ✓ Hardware design

Propositional Logic

A *proposition* is a statement that is either true or false.

Examples of propositions:

- Two plus two is four.
- Toronto is the capital of Canada.
- There is an infinite number of primes.

Not propositions:

- What time is it?
- Have a nice day!

A proposition's *truth value* is a value indicating whether the proposition is true or false.

Propositional variables

We can use *propositional variables* to represent propositions.

Examples:

❖ p: January has 31 days.

❖ q: February has 33 days.

Conjunction

Example:

p: January has 31 days.

q: February has 33 days.

p ∧ q: January has 31 days and February has 33 days.

There are many ways of saying that in English!

Define the propositional variables p and h as:

p: Sam is poor.

h: Sam is happy.

There are many ways to express the proposition $p \wedge h$ in English:

Sam is poor and he is happy.

Sam is poor, but he is happy.

Despite the fact that he is poor, Sam is happy.

Although Sam is poor, he is happy.

Compound propositions: the conjunction

In Java and Python propositions are logical expressions, that can form *compound propositions* using logical operators such as and, or, and not.

For example in Python you can write something like:

>>> (5 >= 0) and (5 <= 10)

We do the same in propositional logic, except we have different notations for the operators. **Don't be intimidated by notation!**

Let p and q be propositions. The *conjunction* of p and q is denoted by p ∧ q.

p ∧ q is true if both p and q are true.

In English, this is the statement "p and q"

Formal definition of the conjunction operator

We can define an operator using its *truth table.*

A truth table shows the truth value of a compound proposition for every possible combination of truth values for the variables contained in it.

p	q	p ∧ q
T	T	T
T	F	F
F	T	F
F	F	F

The disjunction operator

The *disjunction* operation is denoted by $\vee$.
The proposition $p \vee q$ is read "p or q".

$p \vee q$ is true if either one of p or q is true, or if both are true, and is false otherwise.

Notice that $T \vee T = T$. We call $\vee$ the "**inclusive or**" operator.

p	q	p $\vee$ q
T	T	T
T	F	T
F	T	T
F	F	F

Exclusive OR (XOR)

When a parent tells their child – "you can have chocolate or a lollypop", they likely mean that the child can have one of the two but NOT both. This corresponds to the exclusive or logical operator.

The exclusive or of logical statements p and q is denoted by $p \oplus q$

p	q	$p \oplus q$
T	T	F
T	F	T
F	T	T
F	F	F

Negation

Let p be a proposition.

The **negation** of p is denoted by $\neg p$

In English, this is the statement "not p"

p	¬p
T	F
F	T

Summary of operator truth tables

p	q	p ∧ q	p ∨ q	p ⊕ q	¬p
T	T	T	T	F	F
T	F	F	T	T	F
F	T	F	T	T	T
F	F	F	F	F	T

Truth Tables in Python

a	b	a and b	a or b	a^b	not a
True	True	True	True	False	False
True	False	False	True	True	False
False	True	False	True	True	True
False	False	False	False	False	True

Notice the different notation for the operators.
Here a and b are Boolean (`bool`) variables.

Compound propositions

Let's evaluate a proposition composed of multiple operations such as:

$(p \lor (\neg q)) \land r$

where the variables have the values: p:F, q:F, r:F

$(F \lor (\neg F)) \land F$

$(F \lor T) \land F$

$(T) \land F$

F

Compound propositions

In the absence of parentheses how do we evaluate this proposition?

$$p \lor \neg q \land r$$

Possible choices:

- ❖ $(p \lor \neg q) \land r$
- ❖ $p \lor (\neg q \land r)$
- ❖ $p \lor \neg (q \land r)$

Compound propositions

In the absence of parentheses how do we evaluate this proposition?

$$p \vee \neg q \wedge r$$

Possible choices:

- ❖ $(p \vee \neg q) \wedge r$
- ❖ **$p \vee (\neg q \wedge r)$**
- ❖ $p \vee \neg(q \wedge r)$

Like in arithmetic, operators have precedence:

1. Negation ($\neg$)
2. Conjunction ($\wedge$)
3. Disjunction ($\vee$)

Exclusive or

Question: What does $a \oplus b \oplus c$ mean?

1. $(a \oplus b) \oplus c$

2. $a \oplus (b \oplus c)$

3. 1 and 2 are the same

Arithmetic and Logic

The operator precedence in logic mirrors that of arithmetic:

1. Negation (¬)   Minus sign (-)
2. Conjunction (∧)   Multiplication (*)
3. Disjunction (∨)   Addition (+)

In fact, when you do arithmetic over the bits 0 and 1, there is a **correspondence** between conjunction and multiplication, and disjunction and addition (but 1+1=1).

Boolean ·	Logical ∧
0 · 0 = 0	F ∧ F = F
0 · 1 = 0	F ∧ T = F
1 · 0 = 0	T ∧ F = F
1 · 1 = 1	T ∧ T = T

Boolean +	Logical ∨
0 + 0 = 0	F ∨ F = F
0 + 1 = 1	F ∨ T = T
1 + 0 = 1	T ∨ F = T
1 + 1 = 1	T ∨ T = T

Compound propositions

When constructing the truth table for a compound proposition such as ¬q ∧ (p ∨ r) it is helpful to use intermediate columns.

Do it:

p	q	r	¬q	(p ∨ r)	¬q ∧ (p ∨ r)
T	T	T			
T	T	F			
T	F	T			
T	F	F			
F	T	T			
F	T	F			
F	F	T			
F	F	F			

Compound propositions

When constructing the truth table for a compound proposition such as ¬q ∧ (p ∨ r) it is helpful to use intermediate columns:

p	q	r	¬q	(p ∨ r)	¬q ∧ (p ∨ r)
T	T	T	F	T	F
T	T	F	F	T	F
T	F	T	T	T	T
T	F	F	T	T	T
F	T	T	F	T	F
F	T	F	F	F	F
F	F	T	T	T	T
F	F	F	T	F	F

Compound propositions

p	q	¬q	p ∨¬q	p ∧ q	(p ∨¬q) ∨ (p ∧ q)
T	T				
T	F				
F	T				
F	F				

Compound propositions

p	q	¬q	p ∨¬q	p ∧ q	(p ∨¬q) ∨ (p ∧ q)
T	T	F	T	T	T
T	F	T	T	F	T
F	T	F	F	F	F
F	F	T	T	F	T

From English to propositional logic

Consider the following pieces of identification a person might have in order to apply for a credit card:

B: Applicant presents a birth certificate.

D: Applicant presents a driver's license.

M: Applicant presents a marriage license.

Write a logical expression for the requirements under the following conditions:

The applicant must present either a birth certificate, or a driver's license, or a marriage license.

From English to propositional logic

Consider the following pieces of identification a person might have in order to apply for a credit card:

B: Applicant presents a birth certificate.

D: Applicant presents a driver's license.

M: Applicant presents a marriage license.

Write a logical expression for the requirements under the following conditions:

The applicant must present at least two of the following forms of identification: birth certificate, driver's license, marriage license.

From English to propositional logic

Consider the following pieces of identification a person might have in order to apply for a credit card:

B: Applicant presents a birth certificate.

D: Applicant presents a driver's license.

M: Applicant presents a marriage license.

Write a logical expression for the requirements under the following conditions:

Applicant must present either a birth certificate or both a driver's license and a marriage license.

How many rows in that truth table?

p	q	r	(p ∨ r) ∧ ¬q
T	T	T	F
T	T	F	F
T	F	T	T
T	F	F	T
F	T	T	F
F	T	F	F
F	F	T	T
F	F	F	F

How many rows are there in a truth table with 3 variables?
With 4 variables, 5? n?

Exercises

- ✓ Describe in words when the expression $p \lor q \lor r \lor s \lor t$ is true and when it is false.

- ✓ Describe in words when the expression $p \land q \land r \land s \land t$ is true and when it is false.

A puzzle

Knights and knaves.

An island has two types of inhabitants: knights, who always tell the truth, and knaves, who always lie.

You encounter two people, A and B.

A says: "B is a knight"

B says: "The two of us are opposite types"

Can you figure out what A and B are?

A puzzle

Knights and knaves.

An island has two types of inhabitants: knights, who always tell the truth, and knaves, who always lie.

You encounter two people, A and B.

A says: "We are both knaves"

Can you figure out what A and B are?

What "proof rule" did you (implicitly?) use?
  we'll come back to proof rules