An Introduction to Logic

By
Chuck Cusack

Logic: Basic Definitions

- **Definition:** A proposition is a statement that is either *true* or *false*, but not both.
- **Defintion:** The value of a proposition is called its truth value. Denoted by T if it is *true*, F if it is *false*

Example 1: The statement "John Cusack is the president of the U.S.A." is a proposition with truth value *false*.

Example 2: The statement "Do your homework" is not a proposition because it is not a statement that can be *true* or *false*.

Logical Connectives

- **Connectives** are used to create a proposition from several other propositions.
- Such propositions are called **compound propositions**
- The most common connectives are:
  - NEGATION ($\neg$ or !)
  - AND ($\wedge$)
  - OR ($\vee$)
  - XOR ($\oplus$)
  - IMPLICATION ($\rightarrow$)
  - BICONDITIONAL or IF AND ONLY IF ($\leftrightarrow$)

Connective Examples

- Let p be the proposition "The sky is clear."
- Let q be the proposition "It is raining."
- Some examples that combine these are:
  - The sky is clear and it is raining. ($p \wedge q$)
  - The sky is clear and it is not raining. ($p \wedge \neg q$)
  - It is raining if and only if the sky is not clear. ($q \leftrightarrow \neg p$)

Truth Tables

- **Truth Tables** are used to show the relationship between the truth values of individual propositions and the compound propositions based on them.
- **Example:**

p	q	p ∧q
T	T	T
T	F	F
F	T	F
F	F	F

NEGATION

- If *p* is a proposition, the **negation** of *p*, denoted $\neg p$, is "it is not the case that *p*."
- **Example:** Let *p* be the statement "this class has 30 students." Then $\neg p$ is the statement "this class does not have 30 students."
- It should be obvious that the negation of a proposition has the opposite truth value. In other words, if *p* is true, then $\neg p$ is false.
- The truth table for $\neg p$ is

p	¬p
T	F
F	T

AND

- Let p and q be propositions. The proposition "p and q," denoted by $p \wedge q$, is true if and only if both p and q are true.
- $p \wedge q$ is called the **conjunction** of p and q.
- The truth table for $p \wedge q$ is

p	q	p ∧q
T	T	T
T	F	F
F	T	F
F	F	F

7

OR

- Let p and q be propositions. The proposition "p or q," denoted by $p \vee q$, is false if and only if both p and q are false. In other words, it is true if either p or q is true, and false otherwise.
- $p \vee q$ is called the **disjunction** of p and q.
- The truth table for $p \vee q$ is

p	q	p∨q
T	T	T
T	F	T
F	T	T
F	F	F

8

XOR

- Let p and q be propositions. The proposition "p exclusive or q," denoted by $p \oplus q$, is true if and only if either p or q is true, but not both.
- When the term **OR** is used in conversation, often the correct interpretation is **XOR.**
- The truth table for $p \oplus q$ is

p	q	p⊕q
T	T	F
T	F	T
F	T	T
F	F	F

9

IMPLICATION

- Let p and q be propositions. The proposition "p implies q," denoted by $p \rightarrow q$, is false if and only if p is true and q is false.
- $p \rightarrow q$ is called an **implication**.
- The truth table for $p \rightarrow q$ is

p	q	p→q
T	T	T
T	F	F
F	T	T
F	F	T

10

BICONDITIONAL

- Let p and q be propositions. The proposition "p if and only if q," denoted by $p \leftrightarrow q$, is true if and only if p and q have the same truth value.
- $p \leftrightarrow q$ is called a **biconditional**.
- The truth table for $p \leftrightarrow q$ is

p	q	p↔q
T	T	T
T	F	F
F	T	F
F	F	T

11

Constructing Truth Tables

- Construct the truth table for the proposition $((p \wedge q) \vee \neg q)$
- We do this step by step as follows:

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

12

Everyday Logic

- Logic is used in many places:
  - Writing
  - Speaking
  - Search engines
  - Mathematics
  - Computer Programs
- A proper understanding of logic is useful, as the following examples will demonstrate.

13

Logic in Searching I

- **Situation:** You want to find out all you can about disc golf.
- **Problem:** When you search for “disc golf,” you get many hits about golf and some about discs, but can’t find those about “disc golf.”
- **Solution:** You need to find sites which mention both disc and golf, not either word. Search for

  disc AND golf

14

Logic in Searching II

- **Situation:** You just bought some fresh corn, and you need a cornhusker to husk it, so you search for “cornhusker” on the Internet
- **Problem:** Most of the results you get are about UNL’s football team.
- **Solution:** You need to find sites which mention cornhusker, but not UNL or football. Search for

  cornhusker AND NOT (UNL OR football)

15

Logic at Home

- **Situation:** Your mom said “If you are good, you can have some ice cream **or** some cake.”
- **Problem:** You were good, so you ate some ice cream **and** some cake. Your mom got mad because you had both.
- **Solution:** A simple miscommunication. By having ice cream **and** cake, you had ice cream **or** cake. But as is often the case in conversation, she really meant **XOR**, not **OR**.

16

Logic in School

- **Situation:** You have 3 tests for a class. If you get an **A** on any two of them, or get an **A** on at least one but do not fail any of them, you will get an **A** for the course.
- **Problem:** You are lazy, but want an **A.**
- **Solution:** Because of the OR condition, the minimal you can do is get an **A** on two exams and fail the third, or get an **A** on one exam and **D**s on the other two. I’ll pick one **A** and 2 **D**s.

17

Logic in Programming I

- **Situation:** If *x* is greater than 0 and is less than or equal to 10, you need to increment it.
- **Problem:** You tried the following, but it seems too complicated, and doesn’t compile.

```
if(0<x<10 OR x=10) x++;
```

- **Solution:** Try:

```
if(x>0 AND x<=10) x++;
```

18

Logic and e-Mail Filtering I

- **Situation:** You are tired of getting *spam* about losing weight and making money on eBay.
- **Problem:** You tried the following filter:

```
If(subject contains weight and subject
    contains eBay) Delete message
```

- **Solution:** You meant:

```
If(subject contains weight or subject
    contains eBay) Delete message
```

19

Logic in Programming II

- **Situation:** Consider the following loop:

```
while(NOT(A[i]!= 0 AND NOT(A[i]>= 10)))
```

- **Problem:** You are convinced this is way too complicated.
- **Solution:** Well, it's hard to say at this point…

20

Logic in Programming III

- **Situation:** Consider the following loop:

```
while( (i<size AND A[i]>10) OR (i<size
    AND A[i]<0) OR NOT (A[i]!= 0 AND
    NOT (A[i]>= 10) ) )
```

- **Problem:** You are convinced this is way too complicated.
- **Solution:** Yet another example we can't solve.

21

Logic and e-mail Filtering II

- **Situation:** You get so much *spam*, you decide to delete any message not sent (or copied) to you (blah), or a group to which you belong (foo). You try:

```
if( TO or CC does not contain blah OR
    TO or CC does not contain foo)
    Delete message
```

- **Problem:** It seems like all of your e-mail is being deleted.
- **Solution:** This one is confusing, and we will solve it later.

22

Logic and Medication

- **Situation:** Your new medication has the following label:

  Take 1 or 2 pills every 4-6 hours until condition improves. Do not exceed 6 pills per day, or take for longer than 7 days unless directed by a doctor. Do not take this drug during the last 3 months of pregnancy, unless directed by a doctor. If you are taking XYZs and PDQs or have taken either an XYZ or a PDQ within the last 90 days, and the other within the last 30 days, or are taking an ABC or have taken an ABC within the last 60 days, you should not take this drug. Do not drink alcohol or smoke while on this drug. If you have a heart condition, asthma, diabetes, or have an IQ below 25, do not take this drug. Do not take this drug if you have a high fever, cold sweats, runny nose, headache, or sore throat, and discontinue use if you develop any of these symptoms, and dizziness, nervousness, or sleeplessness occur.
- **Problem:** Under what conditions *can* you take it?
- **Solution:** Well, this one may take more than simple logic.

23

Why Logic?

- Hopefully the last several examples have convinced you that knowing more logic is important for computer science, and life in general.
- If you are still not convinced, tough. You are going to learn it anyway.

24

Propositional Equivalences

- Many problems, included the last several examples, can be solved by understanding the concept of propositional equivalences.
- **Example:** The statement “I am not a student and I am not living in the dorm” is equivalent to “I am not a student or living in the dorm.”
- **Example:** “You will pass this class or you will not be in the J.D. Edwards program next year” is equivalent to “If you are to be in the J.D. Edwards program next year, then you must pass this class.”

25

Some Terminology

- **Definition:** A tautology is a proposition that is always *true*.
- **Definition:** A contradiction is a proposition that is always *false*.
- **Definition:** A proposition that is not a tautology or a contradiction is a contingency.

26

Propositional Equivalence

- **Definition #1:** Propositions p and q are called logically equivalent if $p \leftrightarrow q$ is a tautology.
- **Definition #2:** Propositions p and q are logically equivalent if and only if they have the same truth table.
- **Notation:** If p and q are equivalent, we write $p \Leftrightarrow q$
- **Example:** The propositions $\neg p \vee q$ and $p \rightarrow q$ are logically equivalent. We can see this by constructing the truth tables.

p	q	p→q	¬p	¬p∨q
T	T	T	F	T
T	F	F	F	F
F	T	T	T	T
F	F	T	T	T

27

Another Example

- We show that $p \vee (q \wedge r)$ and $(p \vee q) \wedge (p \vee r)$ are logically equivalent.

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

28

Simple Logical Equivalences

Equivalences involving one proposition	
Equivalence	**Name**
p∨T ⇔ T p∧F ⇔ F	Domination laws
p∧T ⇔ p p∨F ⇔ p	Identity laws
p∨p ⇔ p p∧p ⇔p	Idempotent laws
¬(¬p) ⇔p	Double negation law
p∨¬p ⇔ T p∧¬p ⇔ F	Cancellation laws (Not an offical name)

29

Logical Equivalences

Equivalences involving multiple propositions	
Equivalence	**Name**
p∨q ⇔ q∨p p∧q ⇔ q∧p	Commutative laws
(p∨q)∨r ⇔ p∨(q∨r) (p∧q)∧r ⇔ p∧(q∧r)	Associative laws
p∨(q∧r) ⇔ (p∨q)∧(p∨r) p∧(q∨r) ⇔ (p∧q)∨(p∧r)	Distributive laws
¬(p∧q) ⇔ ¬p∨¬q ¬(p∨q) ⇔ ¬p∧¬q	De Morgan’s laws
(p→q) ⇔ (¬p∨q)	Implication law

30

Using Logical Equivalences I

Example 1:

- Show that $(p \wedge q) \to q$ is a tautology using logical equivalences.

$$
\begin{aligned}
((p \wedge q) \to q) &\Leftrightarrow \neg(p \wedge q) \vee q && \leftarrow \text{Implication law} \\
&\Leftrightarrow (\neg p \vee \neg q) \vee q && \leftarrow \text{De Morgan's Law} \\
&\Leftrightarrow \neg p \vee (\neg q \vee q) && \leftarrow \text{Associative law} \\
&\Leftrightarrow \neg p \vee \mathrm{T} && \leftarrow \text{Cancellation Law} \\
&\Leftrightarrow \mathrm{T} && \leftarrow \text{Domination Law}
\end{aligned}
$$

31

Using Logical Equivalences II

Example 2:

Show that $(\neg(q \to p)) \vee (p \wedge q)$ is logically equivalent to q

$$
\begin{aligned}
& && \downarrow \text{Implication law} \\
(\neg(q \to p)) \vee (p \wedge q) &\Leftrightarrow (\neg(\neg q \vee p)) \vee (p \wedge q) && \\
&\Leftrightarrow (q \wedge \neg p) \vee (p \wedge q) && \leftarrow \text{De Morgan's and double negation} \\
&\Leftrightarrow (q \wedge \neg p) \vee (q \wedge p) && \leftarrow \text{Commutative law} \\
&\Leftrightarrow q \wedge (\neg p \vee p) && \leftarrow \text{Distributive law} \\
&\Leftrightarrow q \wedge \mathrm{T} && \leftarrow \text{Cancellation law} \\
&\Leftrightarrow q && \leftarrow \text{Identity law}
\end{aligned}
$$

32

Logic in Programming II

- **Situation:** Consider the following loop:
```
while(NOT(A[i]!=0 AND NOT(A[i]>= 10)))
```
- **Problem:** You are *still* convinced this is way too complicated, and now you think you can simply it.
- **Solution:** We can use De Morgan's law and the double negation law to obtain
```
while( A[i]==0 OR A[i]>= 10)
```

33

Logic in Programming III

- **Situation:** Consider the following loop:
```
while( (i<size AND A[i]>10) OR (i<size  AND
A[i]<0) OR NOT (A[i]!= 0 AND NOT (A[i]>= 10)))
```
- **Problem:** You are convinced this is way too complicated, and with some work, you can simplify it.
- **Solution:** Start by simplifying the last part is in the last example:
```
while( (i<size AND A[i]>10) OR (i<size  AND
A[i]<0) OR (A[i]==0 OR A[i]>= 10) )
```
- Then, use the distributive law:
```
while( (i<size AND ( A[i]>10 OR A[i]<0) ) OR
(A[i]==0 OR A[i]>= 10) )
```

34

An Important Note

- In many programming languages, including Java, C++, and C, applying the commutative law to a proposition may or may not be a good idea.
- The reason for this is that these languages use a technique sometimes call "short circuiting."
- For instance, if A is an array of n elements, the statements
```
if(i<n AND A[i]==0)
```
and
```
if(A[i]==0 AND i<n)
```
are NOT equivalent. Why?

35

Logic and e-Mail Filtering II

- **Situation:** You get so much *spam*, you decide to delete any message not sent (or copied) to you (blah), or a group to which you belong (foo). You try:
```
if( TO or CC does not contain blah OR TO or CC
does not contain foo) Delete message
```
- **Problem:** It seems like all of your e-mail is being deleted.
- **Solution:** This one is a little more complicated. We start by applying De Morgan's law:
```
if(NOT (TO or CC contains blah AND TO or CC
contains foo) ) Delete message
```
- Let p="TO or CC contains blah" and q="TO or CC contains foo".

36

Logic and e-Mail Filtering II

- The statement becomes
  `if(NOT (p AND q)) Delete message`
- The truth table for `NOT(p AND q)` is:
- So e-mail is deleted unless TO or CC contain both blah and foo, which is clearly not what we wanted.
- If TO or CC contains either blah or foo, we do not want to delete. The truth table we want is…
- Applying negation, we now recognize this as:

p	q	$p \wedge q$	$\neg(p \wedge q)$	?	$p \vee q$
T	T	T	F	F	T
T	F	F	T	F	T
F	T	F	T	F	T
F	F	F	T	T	F

37

Logic and e-Mail Filtering II

- Then what we really want is:
  `if(NOT(p OR q)) Delete message`
- Using DeMorgan's Law, it becomes:
  `if(NOT p AND NOT q)) Delete message`
- Retranslating, we seem to have wanted
  `if( TO or CC does not contain blah AND TO or CC does not contain foo) Delete message`
- Great. We did it.
- Or did we?

p	q	$\neg(p \vee q)$
T	T	F
T	F	F
F	T	F
F	F	T

38

Logic and e-Mail Filtering II

- We now think that the following filter should work:
  `if( TO or CC does not contain blah AND TO or CC does not contain foo) Delete message`
- Unfortunately, all of your e-mail is still being deleted.
- Let's keep trying. Let
  p = "TO contains blah" q = "CC contains blah"
  r = "TO contains foo" s = "CC contains foo"
- The filter is
  `if( (NOT p OR NOT q ) AND (NOT r OR NOT s) ) Delete message`
- This is where we went wrong earlier…

39

Logic and e-Mail Filtering II

- Continuing, we had
  `if( (NOT p OR NOT q ) AND (NOT r OR NOT s) ) Delete message`
- Applying De Morgan's Law, we have
  `if( NOT (p AND q ) AND NOT( r AND s) ) Delete message`
- Applying De Morgan's Law again, we have
  `if( NOT( (p AND q ) OR ( r AND s) ) ) Delete message`
- In English, the condition is loosely translated
  "If it is not the case that either
  both TO AND CC contains blah OR
  both TO AND CC contains foo"

40

Logic and e-Mail Filtering II

- In other words, unless the person sending the message put your address in both the TO and CC fields, it will be deleted.
- The problem is that the filter parses "if TO OR CC (X)" as "if TO (X) OR CC (X)."
- This means that "if TO OR CC (NOT X)" is parsed as
  "if TO (NOT X) OR CC (NOT X),"
  not as
  "if NOT (TO (X) OR CC (X)),"
  which is what we did earlier in the example.
- Because of this, you should not use the "if TO OR CC…" filter with the "does not contain" condition, since it is most likely not what you intended.

41

Logic and e-Mail Filtering II

- Now that we know the problem, we can fix it.
- What we want is really:

```
If(TO does not contain blah AND
   CC does not contain blah AND
   TO does not contain foo AND
   CC does not contain foo) Delete message
```

- Recall the original filter was:

```
if(TO or CC does not contain blah OR
   TO or CC does not contain foo) Delete message
```

42

Logic and e-Mail Filtering II

- This example should help illustrate the following
  1. Sometimes we can think a statement means one thing when it actually means another.
  2. Sometimes, we simply can't figure out what a statement means at all.
- In these cases, we can use logic to assist us in determining the true meaning of statements.

Logic and Medication

- **Situation:** Your new medication has the following label:
  Take 1 or 2 pills every 4-6 hours until condition improves. Do not exceed 6 pills per day, or take for longer than 7 days unless directed by a doctor. Do not take this drug during the last 3 months of pregnancy, unless directed by a doctor. If you are taking XYZs and PDQs or have taken either an XYZ or a PDQ within the last 90 days, and the other within the last 30 days, or are taking an ABC or have taken an ABC within the last 60 days, you should not take this drug. Do not drink alcohol or smoke while on this drug. If you have a heart condition, asthma, diabetes, or have an IQ below 25, do not take this drug. Do not take this drug if you have a high fever, cold sweats, runny nose, headache, or sore throat, and discontinue use if you develop any of these symptoms, and dizziness, nervousness, or sleeplessness occur.
- **Problem:** Under what conditions *can* you take it?
- **Solution:** O.K., I give up. But seriously, there may be a time when you really need to solve a similar problem.

Some Exercises

1. Construct the truth table for the following propositions
   a) $(p\rightarrow q)\wedge p$
   b) $(\neg p\leftrightarrow\neg q)\leftrightarrow(q\leftrightarrow r)$
   c) $p\wedge(q\rightarrow\neg(r \vee q))$
2. I do not want any e-mail that contains the words puke, ralph, or hurl, unless it was specifically sent to me (blah). How do I do it?
3. Is $\neg(p\rightarrow q)\rightarrow\neg q$ a tautology? Give two different proofs.
4. Show that $p\leftrightarrow q$ and $(p\wedge q)\vee(\neg p\wedge\neg q)$ are logically equivalent.
5. Show that $[(p\vee q)\wedge(p\rightarrow r)\wedge(q\rightarrow r)]\rightarrow r$ is a tautology.
   Give a proof using equivalences and a truth table.

6. Can you enter this contest? Can I? Can your friends?
   Rules: Must be at least 18 years old to enter. Must be enrolled in a computer science course at NU, but cannot be enrolled in a course in the J.D. Edwards Program. Full-time employees of UNL, UNO, and UNK are not allowed to enter, unless they are faculty of the computer science department, or work in information systems support.
7. Your mom says "If you don't eat your meat, you can't have any pudding." What should you do? Why?
8. Your friend offers "Heads I win, tails you lose." Do you take the bet?
9. If you can't understand this problem, then you didn't learn the material from these notes. Did you learn the material from these notes?
10. If you don't want to not learn the material from this class, you should not fail to not skip doing problems. Or should you?
11. In an episode of *The Simpsons* Bart said something similar to "All I know is that I'm getting straight A's, and that ain't not bad." Does he deserve all A's?