Section 2.1 – Logical Form and Logical Equivalence

Consider the compilation process, converting source code into machine code, ie. 0's and 1's.

Source Code $\xrightarrow{\textbf{\textit{compiler}}}$ Assembly/Object Code $\xrightarrow{\textbf{\textit{assembler}}}$ Machine Code

Source Code

```
int square(int num)
{
  return num * num;
}
```

→

Assembly Code

```
square(int):
push rbp
mov  rbp, rsp
mov  DWORD PTR [rbp-4], edi
mov  eax, DWORD PTR [rbp-4]
imul eax, eax
pop  rbp
ret
```

Machine Code

0010101110101101111100010101 1...

These 0's and 1's are natural to use to represent the state of a computer circuit, since digital circuits have just two states, ON and OFF.

And this leads us to logic statements, which are only TRUE or FALSE.

Consider $x = 7.4$ or $\theta = \frac{\pi}{2}$. What are they? ***Variables***.

We can do the same with statements:

$p =$ Juniata College is in Huntingdon, PA $\qquad$ $q =$ Discrete Structures is MA 300.

These are ***propositions***, they are either True (***T***) or False (***F***).

Are the following ***propositional*** statements? *He likes math* vs. *Joe likes math*

$x + y > 5$ vs. $10 + 14 > 5$

We can combine propositional statements with ***connectives***:

C++/Java	Python	"English"	Symbolic	Definition
`&&`	`and`	AND	$\wedge$	***conjunction***
`||`	`or`	OR	$\vee$	***disjunction***
`!`	`not`	NOT	$\sim$	***negation***

We use the connectives to create more complicated expressions, called compound statements.

statement form $\qquad p \wedge q \qquad\qquad p \vee q$

Using the above definitions: Juniata College is in Huntingdon, PA AND Discrete Structures is MA 300

Truth Table

p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

statement form $p \vee q$

Using the above definitions: Juniata College is in Huntingdon, PA OR Discrete Structures is MA 300

Truth Table

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

statement form $\sim p$

Using the above definitions: Juniata College is NOT in Huntingdon, PA

Truth Table

p	$\sim p$
T	F
F	T

A common theme in this class is translation. And how when using the strict mathematical definitions, which are not ambiguous, may disagree with the informal every day use.

Would you like French fries ***OR*** Onion rings?

The definition of the OR/Disjunction in ***inclusive***, both fries and onion rings are allowed mathematically. But the intent is that this is ***exclusive***, only one of fries or onion rings are being offered. Note the truth table for the ***Exclusive OR***, XOR.

p	q	$p\ XOR\ q$
T	T	***F***
T	F	***T***
F	T	***T***
F	F	***F***

Partner up

Using ONLY

- and/∧
- or/∨
- not/∼

Develop two different compound expressions which are equivalent to the XOR.

p	q	$p \vee q$	$p \wedge q$	$\sim(p \wedge q)$	$(p \vee q) \wedge \sim(p \wedge q)$
T	T	T	T	F	F
T	F	T	F	T	T
F	T	T	F	T	T
F	F	F	F	T	F

and

$(p \wedge \sim q) \vee (\sim p \wedge q)$

Note that we will show equivalence between two compound expressions by showing they have the same truth table.

AND IT IS OK TO USE SEVERAL INTERMEDIATE COLUMNS.

Construct a truth table for the statement form $(p \wedge q) \vee \sim r$.

Partner up

Note that there are three variables, p, q and r.
How many rows will be in the Truth Table?
How many different combinations of three T/F variables?
How many different ways to flip a coin three times in a row, when order matters?

==

p	q	r	$p \wedge q$	$\sim r$	$(p \wedge q) \vee \sim r$
T	T	T	T	F	T
T	T	F	T	T	T
T	F	T	F	F	F
T	F	F	F	T	T
F	T	T	F	F	F
F	T	F	F	T	T
F	F	T	F	F	F
F	F	F	F	T	T

For the statement "John is tall and Jim is redheaded" to be true, both components must be true. So for the statement to be false, one or both components must be false. Thus the negation can be written as "John is not tall or Jim is not redheaded." In general, the negation of the conjunction of two statements is logically equivalent to the disjunction of their negations. That is, statements of the forms $\sim(p \wedge q)$ and $\sim p \vee \sim q$ are logically equivalent. Check this using truth tables.

p	q	$\sim p$	$\sim q$	$p \wedge q$	$\sim(p \wedge q)$	$\sim p \vee \sim q$
T	T	F	F	T	F	F
T	F	F	T	F	T	T
F	T	T	F	F	T	T
F	F	T	T	F	T	T
$\sim(p \wedge q)$ and $\sim p \vee \sim q$ always have the same truth values, so they are logically equivalent						

Symbolically: $\sim(p \wedge q) \equiv \sim p \vee \sim q.$ $\sim(p \vee q) \equiv \sim p \wedge \sim q.$ $\sim(\sim p) \equiv p$

A couple of points. The "triple equal" ≡ means "equivalent to."

And we call these DeMorgan's Laws.

Finally,

A variety of English words translate into logic as ∧, ∨, or ~. For instance, the word *but* translates the same as *and* when it links two independent clauses, as in "Jim is tall but he is not heavy." Generally, the word *but* is used in place of *and* when the part of the sentence that follows is, in some way, unexpected. Another example involves the words *neither-nor*. When Shakespeare wrote, "Neither a borrower nor a lender be," he meant, "Do not be a borrower and do not be a lender." So if p and q are statements, then

p but q	means	p and q
neither p nor q	means	$\sim p$ and $\sim q$.

Example 2.1.2 **Translating from English to Symbols: *But* and *Neither-Nor***

Write each of the following sentences symbolically, letting h = "It is hot" and s = "It is sunny."

a. It is not hot but it is sunny.

b. It is neither hot nor sunny.

Solution

a. The given sentence is equivalent to "It is not hot and it is sunny," which can be written symbolically as $\sim h \wedge s$.

b. To say it is neither hot nor sunny means that it is not hot and it is not sunny. Therefore, the given sentence can be written symbolically as $\sim h \wedge \sim s$.

A **tautology** is a statement form that is always true regardless of the truth values of the individual statements substituted for its statement variables. A statement whose form is a tautology is a **tautological statement**.

A **contradiction** is a statement form that is always false regardless of the truth values of the individual statements substituted for its statement variables. A statement whose form is a contradiction is a **contradictory statement**.