Canberra Computer Science Enrichment: Logic 1

Dirk Pattinson and Nicholas Miehlbradt

Feb 25, 2022

1 Basics

Logic talks about *propositions*, like 'It is raining' or 'Tomorrow is Monday'. Propositions can be true (written $\top$) or false (written $\bot$). At the heart of logic are *connectives*:

Symbol	Meaning	Example	English Meaning
$\wedge$	and	'It's raining' $\wedge$ 'The sun shines'	It is raining and the sun shines
$\vee$	or	'Olf is red' $\vee$ 'Olf is blue'	Olf is red or blue
$\rightarrow$	implies	'Luna is a Husky' $\rightarrow$ 'Luna likes to run'	If Luna is a Husky, then she likes to run
$\neg$	not	$\neg$ 'Luna is a Husky'	Luna is not a Husky

Typically, we use single letter abbreviations for propositions, for example:

$$H \equiv \text{'Luna is a Husky'} \quad P \equiv \text{'Luna is a Pointer'} \quad F \equiv \text{'Luna plays Fetch'} \quad R \equiv \text{'Luna chases rats'}$$

We can then write *formulae* using propositions and connectives, for example:

$$H \rightarrow \neg F$$

which means 'If Luna is a Husky then she does not play fetch'. Formulae can be true of false, depending on a situation. For example, the formula

$$F \wedge \neg F$$

is always false.

2 English to Logic

Try your hand at writing the following sentences in logic, using the abbreviations L, P, F and R above. The key is to think precisely about the meaning of conjunctions in English.

1. If Luna does not chase rats, she is a Husky.
2. Luna chases rats but she is a Husky.
3. Luna is not a Pointer unless she plays fetch
4. Luna is neither a Husky nor a Pointer.
5. Luna plays fetch even if she is a Husky.
6. Luna plays fetch only if she is a pointer.

Conjunction	Meaning
but	
unless	
neither nor	
even if	
only if	

3 Logic to English

Again, using the same propositions as above, try and translate the following two sentences into English:

1. $(F \wedge R) \to H$
2. $(F \to H) \vee (R \to H)$

4 Meaning of Formulae

It's not always easy to understand the meaning of formulas. Mathematics to the rescue! The meaning of a formula depends on a *situation*, that is, a choice of truth value in $\{\top, \bot\}$ for every proposition. For example:

Proposition	Truth Value
H	$\top$
P	$\top$
F	$\bot$
R	$\top$

Here, the intended meaning of H and P suggests that they cannot both be true at the same time. Maths doesn't know this, so that the above is a perfectly acceptable situation.

Once we know the truth values of the propositions, we can evaluate the truth values of formulae according to the following tables:

p	q	$p \wedge q$
$\top$	$\top$	$\top$
$\top$	$\bot$	$\bot$
$\bot$	$\top$	$\bot$
$\bot$	$\bot$	$\bot$

p	q	$p \vee q$
$\top$	$\top$	$\top$
$\top$	$\bot$	$\top$
$\bot$	$\top$	$\top$
$\bot$	$\bot$	$\bot$

p	q	$p \to q$
$\top$	$\top$	$\top$
$\top$	$\bot$	$\bot$
$\bot$	$\top$	$\top$
$\bot$	$\bot$	$\top$

p	$\neg p$
$\top$	$\bot$
$\bot$	$\top$

In the situation $H = P = R = \top$ and $F = \bot$, we can now determine the truth value of the two formulae above, like so:

$$\begin{aligned}
& (F \wedge R) \to H && \text{using the given values} \\
=& (\bot \wedge \top) \to \top && \text{using truth table for } \wedge \\
=& \bot \to \top && \text{using truth table for } \to \\
=& \top
\end{aligned}$$

Can you work out the truth tables for the formula $(F \to H) \vee (R \to H)$ using the same method? Can you work out the truth values for both in *all* situations?

5 Equivalent Formulae

Two formulae are equivalent if they have the same truth values in all situations. Given that they have the same truth values, we can always replace one with the other: they have the same mathematical *meaning*, just like $2x + 2$ and $2(x + 1)$ have the same meaning. Here are some examples:

1. $p \to q = \neg p \lor q$
2. $\neg(p \land q) = \neg p \lor \neg q$
3. $p = p \lor p = p \land p$

Can you show or convince yourself *why* these formulae are equivalent? And can you use these equivalences to re-write the formula $(F \to H) \lor (R \to H)$ to a more readable form?

6 Conjunctive Normal Form

Conjunctive Normal Form is an important input format for computer programs (sat solvers) that compute situations that make formulae true. A formula in CNF is of the form

$$C_1 \lor C_2 \lor \cdots \lor C_n$$

where each C_i is a *clause*, that is, it's of the form

$$C_i = l_{i1} \lor \cdots \lor l_{ik}$$

and the l_{ij} are *literals*, that is either variables p or negated variables $\neg p$. For example:

1. p and $\neg p$ are in CNF
2. $(p \lor q) \land (p \lor \neg q)$ is in CNF
3. $p \land q$ is in CNF
4. $(p \land q) \land \neg q$ is *not* in CNF

A set of m clauses that uses n variables is represented as

```
p cnf n m
1 2 0
-3 4 0
...
```

in the dimacs-format. Google 'dimacs cnf' for more.

7 Knights and Knaves

Each of A and B are either a knight or a knave. A knight always tells the truth; a knave always lies.

A says: "At least one of us is a knave."

Who is a knight and who is a knave? Try encoding the following problems to conjunctive normal form.

Once you believe you have a correct encoding (and have represented it as a DIMACS CNF text format), run it through MiniSAT online (Google 'cryptominisat web') to find the solution.