2 Applications

2.1 Natural Language

Replace each proposition sentence with a proposition.

In the case of a conditional all of the following are sometimes used:

- if $p$ then $q$
- if $p$, $q$
- $p$ is sufficient for $q$
- $q$ if $p$
- $q$ when $p$
- a necessary condition for $p$ is $q$
- $q$ unless $\neg p$
- $p$ implies $q$
- $p$ only if $q$
- a sufficient condition for $q$ is $p$
- $q$ whenever $p$
- $q$ is necessary for $p$
- $q$ follows from $p$
Example 1

Write the following statement using propositional logic:
*You can access the internet on campus only if you are a CS major or you are not a Freshman*

- $p=$ you can access the internet on campus
- $q=$ you are a CS major
- $r=$ you are a freshman

$p \rightarrow q \lor \neg r$

Check the following cases:

1. I am an English Major and a Freshman
2. I am a CS Major and a Freshman
3. I am a Biology Major and a Junior
Example 2

Write the following statement using propositional logic:
You cannot ride the roller coaster if you are less than 4 feet tall unless you are older than 16

- \( a = \) you can ride the roller coaster
- \( b = \) you are less than 4 feet tall
- \( c = \) you are older than 16

\[ a \rightarrow \neg(b \land \neg c) \]

Check the following cases:
1. 10 year old 3 feet tall.
2. 10 year old 4'6" tall
3. 17 year old 3' 11" tall
2.2 System Specifications

When designing a system we need all of the specifications to be consistent, i.e. no conflicting requirements.

The idea is to rewrite all of the specifications using propositional logic and then use a truth table to find out if there is at least one way in which all of them are satisfied.
**Example 1**

The following are the System Specifications:

1. The diagnostic message is stored or transmitted
2. The diagnostic message is not stored
3. If the diagnostic message is stored, then it is transmitted

We first assign propositions:

- $p =$ the diagnostic message is stored
- $q =$ the diagnostic message is transmitted

Now we translate the specifications into propositional logic expressions:

1. $p \lor q$
2. $\neg p$
3. $p \rightarrow q$

Then we use a truth table to compute all possible truth values of each one of the specifications:

<table>
<thead>
<tr>
<th>$p$</th>
<th>$q$</th>
<th>$p \lor q$</th>
<th>$\neg p$</th>
<th>$p \rightarrow q$</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>

Since there is assignment of truth values that makes all of the specifications true, then the specifications given are consistent.
Example 2

The following are the System Specifications:

1. If the user answers “yes” then do not execute process A
2. Process A is not executed and the user answers “yes”
3. The user answers “no”

We first assign propositions:

- \( a \) = the user answers “yes”
- \( b \) = process A is executed

Now we translate the specifications into propositional logic expressions:

1. \( a \rightarrow \neg b \)
2. \( \neg b \land a \)
3. \( \neg a \)

Then we use a truth table to compute all possible truth values of each one of the specifications:

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>( a )</td>
<td>( b )</td>
<td>( \neg a )</td>
<td>( \neg b )</td>
<td>( a \rightarrow \neg b )</td>
<td>( \neg b \land a )</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
</tr>
</tbody>
</table>

Notice that there is no assignment of the variables that can make all of the specifications true. Therefore the system is inconsistent.
Observation:

What would happen if we had a small real-life problem with 100 propositions?

Is this problem solvable in practice?

This problem is a version of the Satisfiability Problem (SAT) where given a boolean expression we must determine if there is an truth assignment of the variables so that the given proposition is true.

SAT was the first problem that was shown to be NP complete.
2.3 Boolean Searches on the Internet

Search engines such as “google” use boolean expressions to find web pages on the internet.

It is possible to directly use AND, OR, NOT operators when performing a search. However, there are several conventions used:

- $w_1 \land w_2$ is written as $w_1w_2$ there is no need to include the AND operator.
- If we want to search for a string made up of two words, use “$w_1w_2$” (between quotes).
- To exclude pages containing the word $w$ we use $-w$.
- The usual precedence of boolean operators is used.

**Example 1**
Find pages containing Universities in New Mexico.

**Example 2**
Find pages containing Universities in New Mexico or in Arizona.

**Example 3**
Find pages containing Universities in Mexico (not in New Mexico).