Deductive Arguments: Converting Statements to Symbols

ID1050– Quantitative & Qualitative Reasoning

Motivation

- Replacing statements with symbols…
  - …removes distraction and emotion
  - …simplifies the statement
  - …lets us see the logical structure more clearly
- We replace simple statements with a single letter.
  - It is important to clearly define what the letter represents.
  - Example: r="It will rain today."
  - The concept is what is important, not the exact wording: "Rain will fall today.", "It's going to rain today.", etc.
- Compound statements are formed from simple statements connected together with logical operators.

Unary NOT operator

- The NOT operator acts on a single variable.
- NOT is like the negative sign in arithmetic. It changes False to True and True to False.
- The symbol for NOT is ~, and is placed before the variable it acts on.
  - Example: the variable *r* represents the logical statement ‘it will rain today’
  - The logical statement “It will not rain today.” would be written as $\sim r$
  - Other phrasings are “It can’t rain today.”, “It is not the case that it will rain today.”, and others.

Binary Logical Operators

- Just like in arithmetic, binary operators take two values and result in another value. In binary logic, the values can only be True or False.
- We will concentrate on the binary operators AND, OR, and IF..THEN.
  - We would say something like "(simple statement 1) AND (simple statement 2)".
  - Logic texts often use the variables *p* and *q* to represent the two simple statements. The letters themselves aren't important; they are just placeholders.
- We often write symbols in place of AND, OR, and IF..THEN…
  - The symbol for AND is ∩. Symbolically, it is used as p ∩ q.
  - The symbol for OR is ∪. Symbolically, it is used as p ∪ q.
  - The symbol for IF..THEN is the → symbol, and would appear as p → q.

Binary Logical Operators: Details

- AND or ∩
  - AND is True only if both simple statements are True.
  - The word 'but' is an alternative version of AND. "p but NOT q" is logically equivalent to "p AND NOT q".
- OR or ∪: There are two types of *or*:
  - '*Inclusive or*' means "one or the other or both". This is the meaning we will assign to OR and ∪.
  - '*Exclusive or*' mean "one or the other, but not both".
    - When the flight attendant asks "Would you like the chicken or the fish?", you are expected to choose only one.
    - We won't be using this type of *or*, but its symbol is XOR.
- IF..THEN.. or →.
  - Alternative phrasing are *..implies..*, *..leads to..*, *..causes...*, and some others. We might equivalently say "if p then q" or "p implies q" or "p leads to q".
  - These are all written symbolically as $p \rightarrow q$.

Simple Examples

- With simple logical connectives, you can replace the logical operator with its symbol, and replace the phrases it connects with their variable.
- It is useful to write down exactly what concept a variable stands for. Here, r=‘it will rain’ and s=‘it will snow’.

Examples:

- “It will rain and it will snow.” $r \cap s$

  $r \quad \cup \quad s$
- “It will rain or snow.” $r \cup s$

  $r \quad \rightarrow \quad s$
- “If it rains, then it will snow.” $r \rightarrow s$

  $r \quad \rightarrow \quad s$
- “Rain implies snow.” $r \rightarrow s$

  $r \quad \rightarrow \quad s$
- “It will rain only if it snows.” $r \rightarrow s$

...ONLY IF...

- The *only if* logical connective is the same as *implies*, although the rain and snow example doesn't make this clear.
- Here is a better example, where r = 'the engine runs', f = 'the engine has fuel'
  - "The engine runs *only if* it has fuel." $r \rightarrow f$
  - This is equivalent to the more standard phrasing "If the engine runs, then it has fuel."
  - The given statement shouldn't be written f → r, because this would be saying "If the engine has fuel, then it will run." If you've ever had engine trouble, you know this statement isn't correct.

Complex Examples

Let's try some more complex examples:

- $\sim r \quad \cap \quad s$
  "It won't rain but it will snow." $\sim r \cap s$ or $(\sim r) \cap s$
- $r \quad \cup \quad \sim s$
  "It will rain or it won't snow." $r \cup \sim s$ or $r \cup (\sim s)$
- $\sim r \quad \cap \quad \sim s$
  "It won't rain and it won't snow." $\sim r \cap \sim s$ or $(\sim r) \cap (\sim s)$

Complex Examples

- $s \quad \rightarrow \quad r$
  "If it snows, then it will rain." $s \rightarrow r$
  - Order is important for the IF..THEN.. statement!
- $r \quad \rightarrow \quad s$
  "Rain implies snow." $r \rightarrow s$
- $r \quad \rightarrow \quad {\sim}s$
  "Rain will lead to no snow." $r \rightarrow {\sim}s$ or $r \rightarrow ({\sim}s)$

Complex Examples

Sometimes the concept of negation applies to the entire compound statement following it, not just the first simple statement in it.

- Saying "It is not the case that…" is implying that we need to negate everything else that follows it.
- Parentheses are used to make this concept clear. Writing ~(…) means that the negation applies to the result of whatever is in the parentheses.

$$\sim \quad (\quad r \cap s \quad)$$

- Example: "It is not the case that it will rain and snow" $\sim(r \cap s)$
  - Note: This is logically distinct from $\sim r \cap s$ and $\sim r \cap \sim s$. These statements have different results given the same possibilities of *r* and *s* being True or False.
- Other similar phrasings are "It can't be that …", "it can't both…", and others.

Challenge Questions

Try these more difficult examples:

$(\quad s \quad \cap \; r \quad) \rightarrow \qquad w$

- "If it both snows and rains, then the streets will be wet." $(s \cap r) \rightarrow w$

$(\quad a \cap b \qquad) \cup (\quad c \cap d \qquad)$

- "Either Ann and Bob will dance, or Carla and Dan will." $(a \cap b) \cup (c \cap d)$

$q \quad \rightarrow \qquad \sim (\qquad p \cup r \qquad)$

- If I have quiche, then I won't have either the pizza or the ravioli." $q \rightarrow \sim (p \cup r)$

Conclusion

- Logic symbols can be used to simplify a logical statement for analysis.
- Simple statements can be logically connected using unary and binary logical operators.
- We are concerned with the unary NOT and the binary AND, OR, and IF..THEN.. logical operators.
  - There are some alternative phrasings that mean the same thing as these literal phrasings.
  - We will use the *inclusive or*, not the *exclusive or*.
- Use parentheses to make the meaning of a logical statement clear.