Deduction
by Daniel Bonevac

Chapter 5 Quantifiers

Limitations of sentential logic

- Sentential logic is useful, but there are many logical relationships that it simply can not represent.
- For example, sentential logic can not represent the following syllogistic argument as valid.

1. All stoolies are cowards.
2. <u>Frank is a stoolie.</u>
3. Frank is a coward.

- This is because sentential logic simply takes every sentence lacking a connective as atomic. Hence, a sentential logic representation of the above argument could only be:

1. p
2. <u>q</u>
3. r

- Which clearly is not a valid argument form.

Constants

- In order to develop a formalism capable of representing more subtle relationships we need to develop some tools for representing the internal structure of sentences.
- The first thing we need to be able to do is represent the relation of <u>subject</u> to <u>predicate</u>.
  - Subjects are typically <u>nouns</u> and <u>noun phrases.</u>
  - Predicates are typically <u>verbs</u> and <u>verb phrases</u>.
- The simplest kind of subject predicate relation is expressed by a sentence using a proper name like ‘Mike’ and a simple verb phrase like ‘is happy’.
  - Mike is happy.
- The standard way of representing a sentence like this in formal logic is to use a capital letter like ‘A’ to stand for the predicate and a small letter like ‘m’ to represent the subject. We call these letters <u>constants</u>.

Symbolizing simple subject/predicate statements.

- In quantificational or predicate logic it is conventional to write the predicate constant first and the subject constant second. So:
  - Mike is happy

gets read as:

  - Happy (is Mike)

and therefore represented as:

  - Hm.
- To represent a sentence like
  - Frank is a nincompoop.

we would write

  - Nf

In each case, the lower-case constant names an individual <u>object</u> or <u>thing</u> (like a person) and the upper case constant names some <u>property</u> that we ascribe to that thing.

Quantifiers

- Sometimes we assign properties to things using more general terms like:
  - Something is weird.
  - Everything is fine.
  - Some dogs howl
  - All fibbers rot in hell.
- The words “something,” “everything,” “some,” and “all” aren’t proper names, but pronouns.
- In quantificational logic (which we heceforth abbreviate as Q) we introduce new symbols, or quantifiers, which capture the function of words like “something” and “everything”. These are the <u>existential quantifier</u>, which we represent with $\exists$ and the <u>universal quantifier</u> which we represent with $\forall$.

The existential quantifier

- In Q, a sentence like:
  - Something is weird.

can be rewritten as

  - “There is something that is weird.”
  - The phrase “There exists” is captured with the existential quantifier ∃.
  - The pronoun “something” is captured with a <u>variable</u>. In Q variables are lower-case letters from the back of the alphabet, just like in algebra: u, v, w, x, y, z.
- In Q, then, the sentence above would be represented as:
  - $\exists xWx$ = There is an x that is weird.

Reading existentially quantified sentences.

- There are various acceptable ways to read a sentence in Q like $\exists xWx$. For example:
  - There is an x that is W.
  - There exists an x, such that x is W.
  - For some x, x has W.
  - Some x is such that it is W.
  - Some x exemplifies the property of being W.

The universal quantifier

- The universal quantifier ∀ is used to represent the function of words like: <u>all</u> and <u>every</u>. So, for example, the sentence:
  - Everything is swell.

would be expressed in quantificational English as

  - For every x, x is swell.

And written as

  - $\forall xSx$

Categorical sentence forms 1

- From classical syllogistic logic there are four basic categorical sentence forms. We call them categorical because they do not ascribe properties to specific objects, but to classes or categories of objects.
- These forms are
  - Universal affirmative: All F are G.
  - Particular affirmative: Some F are G.
  - Particular negative: Some F are not G.
  - Universal negative: No F are G.
- These sentence forms all have nice intuitively satisfying expression in sentential logic.

Universal affirmative

- A universal affirmative statement like:
  - All frogs are green.

can be written in quantificational English as:

- If x is a frog, then x is green.

and represented in Q as:

- $\forall x\ (Fx \rightarrow Gx)$

Particular affirmative

- A particular affirmative statement like:
  - Some frogs are green.

can be written in quantificational English as:

  - There is at least one x such that it is a frog, and x is green.

and it is represented in Q as:

  - $\exists x\ (Fx\ \&\ Gx)$

- Notice that we use the & here rather than the $\rightarrow$. The reason is that a statement like "Some frogs are green" does not tell us that we can <u>infer</u> from the fact that something is a frog that it is green. It just tells us that there are two properties, frogness and greenness, that sometimes occur together.
- Notice as well that for us 'some' means 'at least one.' This takes some getting used to because in English 'some' usually suggests <u>more</u> than one.

Particular negative

- A particular negative statement like:
  - Some frogs are not green.

can be written in quantificational English as:

- It is not the case that there exists an x such that x is a frog, and x is green.

- It is represented in Q simply as the denial of the particular affirmative.
  - $\neg\exists x\ (Fx\ \&\ Gx)$

Universal negative

- A universal negative statement like:
  - No frogs are green.

can be written in quantificational English as:
  - For all x, if x is a frog, then x is not green.

- It is represented in Q as.
  - $\forall x\ (Fx \rightarrow \neg Gx)$

Polyadic predicates

- Some predicates do not simply express properties of objects, but relations between objects. We call properties “monadic predicates” and relations “polyadic” predicates.
- The statement
  - Tom loves Jenny

  expresses the dyadic predicate or relation ‘loves’ as holding between Tom and Jenny.
- In Q we write this as:
  - Loves (tom, jenny)

or

  - Ltj
- Note that the order here is important. If we wanted to say “Jenny loves Tom” we would write:
  - Ljt

Polyadic predicates with quantifiers

- Polyadic predicates are subject to quantification as well. Here are some English sentences and their involving the dyadic predicate ‘love’, and their Q translations. (See Bonevac p.151.)

Someone loves Tom.	$\exists xLxt$
Jenny loves someone.	$\exists xLjx$
Jenny is loved by someone.	$\exists xLxj$
Everyone loves Tom.	$\forall xLxt$
Tom loves everyone.	$\forall xLtx$
Someone loves everyone.	$\exists x\forall yLxy$
Everyone loves someone.	$\forall x\exists yLxy$
Everyone loves everyone.	$\forall x\forall yLxy$
No one loves anyone.	$\forall x\forall y\neg Lxy$
There is someone who nobody loves.	$\exists x\forall y\neg Lyx$
Everyone loves him/herself	$\forall xLxx$