Logic, First Course, Winter 2020. Week 4, Lecture 1. Back to course website

# Quantification

In this lecture, we motivate the idea of predicate logic and try to state the general idea of how it extends the propositional logic which we have been learning so far. Then we learn the basic idea of translating natural-language sentences into predicate logic, focusing on atomic sentences and quantifier expressions.



## Predicate logic and atomic statements

In propositional logic, we take as our basic unit the proposition, something that can be true or false. By taking it as the basic unit, we mean we do not subject it to any further analysis. For instance, while we translate "Christina is a student and Daisy is a student" as $p \wedge q$, we treat $p$ as a primitive and $q$ as primitives: they are basic. In predicate logic, the idea is to introduce another layer of analysis. In particular, the most basic idea is to further analyze "Christina is a student" as $Sc$ and analyze "Daisy is a student" as $Sd$. So we translate "Christina is a student and Daisy is a student as" as $Sc \wedge Sd$.

One reason is that "Christina is a student" and "Daisy is a student" seem to have a common structure. Someone could not understand the one without understanding the other. And there is a common way to evaluate them: look at the list of students! So translating as $Sc$ and $Sd$ reveals this common structure. Here are some more examples, where we use the key:

$a$ = "Alyssa"
$b$ = "Bryan"
$c$ = "Christina"
$d$ = "Daisy"
$L$ = "is a lawyer"
$M$ = "is a musician"

*H* = "likes hockey"
*S* = "likes soccer"

The name 'Predicate logic' comes from the idea that things like L,M,H,S are predicates or properties. When paired with objects, they form propositions. This is the way in which predicate logic extends propositional logic.

```
Alyssa is a lawyer and Bryan is a musician.
```

Alyssa is a lawyer and Bryan is a musician.

Below is a video of the first one-- remember that you have to "select all" and delete the entry in the first box, and then you have to press **return** at the end of the problem. If you need a refresher on how to type the propositional connectives on the keyboard, please consider reviewing Typing the connectives on the keyboard.

video controls width="700" src="https://logic-teaching.github.io/prop/vid/quantifiers-example1.mp4"/>

```
Alyssa is not a musician and Bryan is not a l
```

Alyssa is not a musician and Bryan is not a lawyer.

```
If Christina likes hockey then Daisy likes ho
```

If Christina likes hockey then Daisy likes hockey.

```
Christina likes hockey or Daisy likes soccer.
```

Christina likes hockey or Daisy likes soccer.

# Motivating quantifiers

Consider the following three sentences:

| Sentence |
| --- |
| Alyssa is nice. |
| Someone is nice. |
| Eveyrone is nice. |

There is an obvious sense in which all of these sentences are formed by plugging different words into the blank in "--- is nice." This is an important perspective. But our translations are intended to display the truth-conditions of the statements: how their truth is determined, usually on the basis of decomposing them into smaller parts. And the truth-conditions of these three sentences differ in obvious ways. When we are trying to figure out whether "Alyssa is nice" is true, we go look at Alyssa and see whether she is nice. To see whether "someone is nice" is true, we go searching for someone (perhaps Bryan?) who is nice. For "everyone is nice," we need to make sure that Alyssa is nice, Bryan is nice, etc. And note that in thinking about the first sentence, we had said how to do that: to see whether a person is nice, simply go look and see whether they are nice. Hence, the moral is: while recognizing the differences between the three sentences, we should recognize the truth of the last two is somehow based on the kinds of things that make sentences like the first sentence true.

# The quantifiers

Our solution is to translate these as follows, where we follow the key that $Nx$ = $x$ is nice, and $a$= Alyssa:

| Sentence | Translation | Typed | Pronunciation | Colloquial pronunciation |
| --- | --- | --- | --- | --- |
| Alyssa is nice. | $Na$ | `Na` | $Na$ | $a$ has $N$ |
| Someone is nice. | $\exists\, x\, Nx.$ | `ExNx` | There is $x$ such that $Nx$ | There is $x$ such that $x$ has $N$ |
| Everyone is | | | | |

| | | | | |
|---|---|---|---|---|
| nice. | ∀ *x Nx* | `AxNx` | For all *x* (pause), *Nx* | For all *x* (pause), *x* has *N* |

In this we use the following symbols:

| Symbol | Name | Handwritten | Typed |
|---|---|---|---|
| ∃ | existential quantifier |  | `E` |
| ∀ | universal quantifier |  | `A` |

Note that in our translation of the second sentence, the *x* appears twice. The instance of *x* in ∃ *x* is a signal: go look for *d* such that *Nd*, where you get *Nd* from *Nx* by plugging in *x* = *d*. More generally, the rule for the truth of ∃ *x Fx* is: it is true when there is at least one *d* such that *Fd*.

Similarly, in our translation of the third sentence, the *x* appears twice. The instance of *x* in ∀ *x* is a signal that one is making an "every" or "all" claim. And the rule for the truth of ∀ *x Fx* is: it is true when *Fd* holds for every individual *d*.

Here are some examples using the key *Nx* = *x* is nice, and *a*= Alyssa:

```
If someone is nice then everyone is nice.
```

If someone is nice then everyone is nice.

Here is a video of this one-- note that the second time we did in the video, we just cut and paste the quantifiers ∃ and ∀ from the text:

video controls width="700" src="https://logic-teaching.github.io/prop/vid/quantifiers-example2.mp4"/>

Note that the computer system is very picky about parentheses and quantifiers and will reject answers that puts a pair of parentheses around a statement whose main connective is the quantifier. This is illustrated in the below video where we try unsuccessfully to type the response `~(AxNx)` to the next problem and then fix it by removing the parentheses and typing the right answer `~AxNx` :

video controls width="700" src="https://logic-teaching.github.io/prop/vid/quantifiers-example3.mp4"/>

You can try it yourself now:

```
Not everyone is nice.
```

Not everyone is nice.

```
Everyone is not nice.
```

Everyone is not nice.

```
If Alyssa is nice then someone is nice.
```

If Alyssa is nice then someone is nice.

# Multiple variables

The variables act like the variables in algebra. For instance, $x = 3$ is a solution to $x + 2 = 5$, since if you plug in 3 for $x$ you get something true. And $x = 4$ and $y = 2$ is a solution to $x + y = 6$, since if you plug in 4 for $x$ and 2 for $y$ you get something true. The existential quantifier is a

way of saying that *there is* a solution which makes the corresponding "equation" true. Let's first practice with respect to the following familiar key:

*a* = "Alyssa"
*b* = "Bryan"
*c* = "Christina"
*d* = "Daisy"
*L* = "is a lawyer"
*M* = "is a musician"
*H* = "likes hockey"
*S* = "likes soccer"

`If Alyssa is a lawyer and Alyssa likes hockey`

If Alyssa is a lawyer and Alyssa likes hockey and Bryan likes soccer, then someone is both a lawyer and likes hockey.

`It is not the case that if Alyssa is a lawyer`

It is not the case that if Alyssa is a lawyer and Alyssa likes hockey and Bryan likes soccer, then someone is both a lawyer and likes soccer.

Note in this next example, the answer will involve a consequent whose main connective is a conjunction and whose first conjunct is a quantifier-phrase. Hence, the consequent in this case will look like `(Ex . . . ./\ . . . )`:

`If Alyssa is a lawyer and Alyssa likes hockey`

If Alyssa is a lawyer and Alyssa likes hockey and Bryan likes soccer, then someone is both a lawyer and likes hockey and someone likes soccer.

```
If someone is a lawyer, then Alyssa is a lawy
```

If someone is a lawyer, then Alyssa is a lawyer or Bryan is a lawyer.

```
If someone is a lawyer and someone is a music
```

If someone is a lawyer and someone is a musician, then Alyssa is a lawyer and Daisy is a musician.

```
If someone likes soccer and is a musician, th
```

If someone likes soccer and is a musician, then Christina likes soccer and Christina is a musician, or Daisy likes soccer and Daisy is a musician.

The universal quantifier is usually a way of stating generalizations or laws. It is a way of saying that everyone/everything has to have some property or feature. Let's translate with the following key:

*a* = "Alyssa"
*b* = "Bryan"
*c* = "Christina"
*d* = "Daisy"
*C* = "is drinking coffee"
*T* = "is drinking tea"
*W* = "is wide awake"

```
If everyone is drinking coffee or tea, then e
```

If everyone is drinking coffee or tea, then everyone is wide awake.

```
If everyone is drinking coffee or everyone is
```

If everyone is drinking coffee or everyone is drinking tea, then everyone is wide awake.

```
Everyone drinking coffee or everyone drinking
```

Everyone drinking coffee or everyone drinking tea is a sufficient condition for everyone drinking coffee or tea.

```
Everyone drinking coffee or everyone drinking
```

Everyone drinking coffee or everyone drinking tea is not a sufficient condition for everyone drinking coffee or for everyone drinking tea.

```
Daisy is not drinking coffee and Christina is
```

Daisy is not drinking coffee and Christina is not drinking tea, and it is not the case that everyone is drinking coffee or everyone is drinking tea.

These last examples hopefully give one a better sense of the truth-conditions of universal statements. For instance, the last couple of examples were pointing out that it can be true that every person taken one by one is either drinking coffee or tea, without it being true that everyone is in the coffee category or everyone is in the tea category.

Here are some examples with multiple universal quantifiers. In this first one, note that even though we use different variables for the two universal statements in the antecedent, they both have consequences for the individual Bryan under consideration.

```
If everyone is drinking coffee and everyone i
```

If everyone is drinking coffee and everyone is wide awake, then Bryan is drinking coffee and Bryan is wide awake.

```
If everyone is wide awake and not everyone is
```

If everyone is wide awake and not everyone is drinking tea, then Bryan is drinking coffee.

```
If Daisy is not wide awake and Daisy is not d
```

If Daisy is not wide awake and Daisy is not drinking tea, then not everyone is wide awake and not everyone is drinking tea.

These are lecture notes written for this course.[1]

---

1. It is run on the Carnap software, which is↩