Logic, First Course, Winter 2020. Week 4, Lecture 1, Handout.

Predicate logic and atomic statements

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 not a musician and Bryan is not a lawyer.

If Christina likes hockey then Daisy likes hockey.

Christina likes hockey or Daisy likes soccer.

Motivating quantifiers

Consider the following three sentences:

Sentence
Alyssa is nice.
Someone is nice.
Eveyrone is nice.

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.	∃ *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

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

If someone is nice then everyone is nice.

Not everyone is nice.

Everyone is not nice.

If Alyssa is nice then someone is nice.

Multiple variables

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

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.

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 lawyer or Bryan is a lawyer.

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, 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 everyone is wide awake.

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

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

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

Here are some examples with multiple universal quantifiers.

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 drinking tea, then Bryan is drinking coffee.

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

Translation schemas

Quantifiers are also very good at capturing the truth-conditions of statements about "some" and "all" which connect two properties. Here are some common translation schemas:

Schema	**Translation**	**Schema Negated**	**Translation of Negation**
Some *F* are *G*	$\exists x\, (Fx \wedge Gx)$	No *F* are *G*	$\neg(\exists x\, (Fx \wedge Gx))$
Some *F* are not *G*	$\exists x\, (Fx \wedge \neg Gx)$	No *F* are not *G*	$\neg(\exists x\, (Fx \wedge \neg Gx))$
All *F* are *G*	$\forall x\, (Fx \rightarrow Gx)$	Not all *F* are *G*	$\neg(\forall x\, (Fx \rightarrow Gx))$
All *F* are not *G*	$\forall x\, (Fx \rightarrow \neg Gx)$	Not all *F* are *G*	$\neg(\forall x\, (Fx \rightarrow Gx))$

Note that in these schemas the "some"-statements are translated with an existential quantifier together with a conjunction, while the "all"-statements are translated with a universal quantifier together with an arrow statement.

Let us practice with these translation schemas with respect to the following examples:

C = "is cheerful"
F = "is festive"
G = "is a geography major"
L = "is a law student"
R = "is relaxed"
T = "likes table-tennis"

Every law student is cheerful and some geography majors are cheerful.

In the following, remember that "but" would just be translated by "and":

Some geography majors are relaxed but no law students are relaxed.

Some geography majors are not festive, but no law students are not festive.

Not all law students like table-tennis.

If all law students are relaxed, then all geography majors are festive.

If all law students are cheerful, then some law students are festive.

These is a handout written for Logic, First Course, Winter 2020. It is run on the Carnap software, which is an: