A puzzle
Suppose we try to provide a linguistic description of each number. For example “1” might be described as “The number of moons orbiting the Earth”, “2” might be “The number of sides of a coin” etc. What is the first number not name-able in under ten words?

Introduction
Consider the argument:

1. All Englishmen are reserved.
2. John is an Englishman.
   Therefore
3. John is reserved.

We could transform this into propositional logic using the following mapping:
- A = All Englishmen are reserved
- B = John is an Englishman.
- C = John is reserved.

However this will result in the formal argument A, B : C.

The intuitive validity of the above is obvious (even if perhaps the soundness isn't). However there is no way in Propositional Logic to verify this argument. This is because in PL letters represent entire propositions (or sentences) and therefore represent nothing of the internal meaning of the above sentences. We clearly need a more representative and powerful logic. This logic is called predicate logic (or First Order Predicate Logic etc.).

Predicates
Sentences 2  3 are termed subject-predicate sentences. That is they attribute some property (or predicate) to some person, place or object (the subject). In addition since they attribute a property to only one subject, they are singular sentences.

We will use lower case letters to represent subjects and upper case to present predicates. So that:

- E = ... is an Englishman
- R = ... is reserved
- j = John.

Notice that E and R are incomplete symbols and have a slot which can be filled by an appropriate subject. We can therefore represent sentences 2 and 3 as

Ej
Rj

We still need to represent sentence 1. This sentence is a general sentence i.e. it makes a claim about all members of a certain class. To represent this we need three things:
1. **Variables.**
   We will use lower case italic variables (and typically from the end of the alphabet) such as \( x, y, z \).

2. **Quantifiers.**
   \( \forall x \) [ ... some expression...] This reads “For all X [....]”
   The square brackets are an integral part of the quantifier and are not optional. They represent
   the scope of the quantifier. (More technically the square brackets represent what the
   quantifier governs) This is essential to avoid ambiguity and allows for nested quantifiers.

3. **The domain.** The domain represents the set of things we're talking about. We can decide that
   the domain is “unrestricted” i.e. that the sentence applies to all things in the universe.
   However, it will be typically more intuitive to say the domain is restricted. For example, we
   can specify that the domain is “human beings”.

Therefore we can represent the argument as follows:
1. \( \forall x \ [Ex \rightarrow Rx] \)
2. \( Ej \)
3. : (therefore)
4. \( Rj \)

So far we have outlined how to represent just 2 types of sentence. In addition, We can add any
connective from propositional logic. Therefore the following are legal expressions:

\[
\forall x \ [Fx \land Ex] \text{ (for all x it is true that Fx and Ex)} \\
\forall x \ [Fx \lor \neg Ex] \text{ (for all x it is true that Fx or not Ex)} \\
\neg \forall x \ [Fx] \text{ (for not all x it is true that Fx)}
\]

**Extending the syntax of Propositional Logic to cover Quantifiers**

Please recall the syntax rules given for Propositional Logic in week 5. i.e.

1. \( F \rightarrow A \)
2. \( F \rightarrow F \land F \)
3. \( F \rightarrow U F \) etc.

plus we need two syntax rules for Quantifiers

4. \( F \rightarrow Q M \)
5. \( F \rightarrow UQ M \)

and rules for subject-predicate formulae

8. \( F \rightarrow S-P N \)
7. \( S-P \rightarrow A|B|C ... \)
8. \( N \rightarrow x|y|z ... \)

We can therefore construct formulae involving any connective in propositional logic i.e. \( \land, \rightarrow, \) etc.

**The Existential Quantification**

Consider the sentence

All unicorns have one horn

This can be translated as
∀x [Ux → Hx]

It's important to be clear what the expression says. It says “If something is a unicorn then it has one horn”. It does not imply that unicorns exist or indeed that there is anything in the world which could be described as having the property of “being a unicorn”.

Similarly the expression “not all men are brave” can be translated as

¬∀x [Mx → Bx]

but does not imply either that there are any men in existence OR that there is a man who is brave. This is because the Universal operator makes no claims about existence. Therefore we need to introduce an existential operator ∃ so that

if F= the property of being Father Christmas then

∃x [Fx] (There exists a being with Father Christmas properties)

Translating from Natural Language

Given an argument in Natural Language, our first step should be to translate into Predicate Logic. There followin

Universal Conditional
All As are Bs → ∀x [Ax → Bx]

Existentially qualified conjunction
Some As are Bs → ∃x [Ax ∧ Bx]

It is common to make the following mistakes:

All As are Bs → ∀x [Ax ∧ Bx] (“For all x, x are A and x are B”) (Wrong!)

Some As are Bs → ∃x [Ax → Bx] (“For all x, if x is an A then x is a B”) (Wrong!)

Interpretation and Domains

Recall when we transformed natural language into propositional logic, we provided a key for which sentence A, B, C etc. stood for. Since Predicate Logic is far more powerful, we need to provide something more: An Interpretation.

In PL we were able to assign truth values to individual parts of a complex expression (and in the case of proving with truth tables show a particular argument was valid or invalid by virtue of showing the result of every possible combination of truth value). We cannot do this with Predicate Logic since the truth value of any part of a complex expression is dependent on the rest of the expression.

To give an interpretation I we must:

1. Give the current choice of domain.
2. A description of exactly how the elements of the formal vocabulary involved e.g. Names Predicates are related to the elements of that domain.

Validity and Invalidity

In propositional logic, an argument (or sequent) is valid iff the premises are true then the conclusion must also be valid. Conversely an argument is invalid iff there is a possible instance where the
premises are true and the conclusion does not follow.

We need a more sophisticated notion of validity to deal with Predicate Logic. Consider the following domains

\[ \overline{D}: \text{(Mark Lee)} \]
\[ C: \text{likes cheese} \]

\[ \overline{D}: \text{(Human beings)} \]
\[ C: \text{likes cheese} \]

and the expressions: \( \forall x \ [Cx] \) and \( \exists x \ [Cx] \). In the first domain, both are most definitely true. In the second domain the latter expression is true but the former is obviously false. Therefore we need to extend the notion of validity:

Validity
A sequent of Predicate Logic is valid iff, for every domain, there is no possible interpretation under which all the premises are true and the conclusion is false.

Invalidity
A sequent of Predicate Logic is invalid iff, for some domain, there is a possible interpretation under which all the premises are true and the conclusion is false.

And for Formulae:

Validity
A formula of Predicate Logic is valid iff, for every domain that formula is true under every possible interpretation.

Negation
As in propositional logic, negation gives up a powerful representational tool. Consider:

There are no unicorns \( \rightarrow \neg \exists x \ [Ux] \) i.e. A negated existential.

OR

There are no unicorns \( \rightarrow \forall x \ [\neg Ux] \) i.e. An universal negation

Moreover with negation we can convert any universal quantifier into an existential. For example, if we say “All As are Bs” then we are also saying “There does not exist an A which is not a B”. More formally we can do the following transformations:

<table>
<thead>
<tr>
<th>Something is A</th>
<th>( \exists x \ [Ax] )</th>
<th>( \neg \forall x [\neg Ax] )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nothing is A</td>
<td>( \neg \exists x \ [Ax] )</td>
<td>( \forall x \ [\neg Ax] )</td>
</tr>
<tr>
<td>Everything is A</td>
<td>( \forall x \ [Ax] )</td>
<td>( \neg \exists x \ [\neg Ax] )</td>
</tr>
<tr>
<td>Some As are Bs</td>
<td>( \exists x \ [Ax \land Bx] )</td>
<td>( \neg \forall x \ [(Ax \land \neg Bx)] )</td>
</tr>
<tr>
<td>All As are Bs</td>
<td>( \forall x \ [Ax \rightarrow Bx] )</td>
<td>( \neg \exists x \ [Ax \land \neg Bx] )</td>
</tr>
<tr>
<td>No As are Bs</td>
<td>( \neg \exists x \ [Ax \land Bx] )</td>
<td>( \forall x \ [Ax \rightarrow \neg Bx] )</td>
</tr>
</tbody>
</table>

Relationships

John loves Mary \( \text{Ljm} \)

Everybody hates Chris

Somebody loves Chris

John loves everybody

John loves somebody