Answers to Some Predicate Logic Formalisation Exercises from the Predicate Logic Notes

Predicates to be used:

lecTheatre/1 same/2

office/1 phd/1

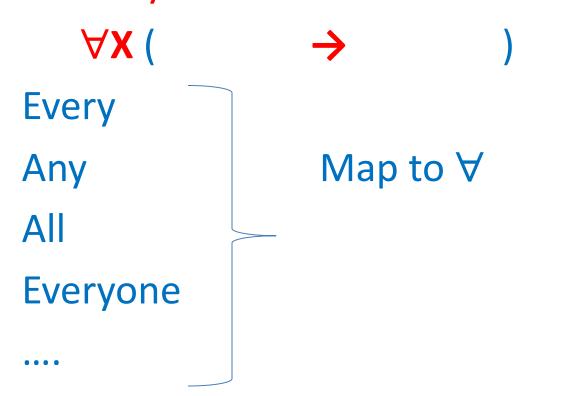
contains/2 supervises/2

lecturer/1 happy/1

has/2 completePhd/1

1.
311 is a lecture theatre and 447 is an office.
311 is a lecture theatre and 447 is an office
lecTheatre(311) ∧ office(447)

2. Every lecture theatre contains a projector. Every lecture theatre contains a projector.



Every lecture theatre contains a projector.

```
∀X (lecTheatre(X) → )
∀X (lecTheatre(X) → contains(X, projector))
Alternatively:
∀X (lecTheatre(X) →
∃Y (projector(Y) ∧ contains(X, Y)))
```

A universally quantified wff is usually like this:

$$\forall X (\rightarrow)$$

Or like this:

$$\forall X (\longleftrightarrow)$$

That is their principle connective is

$$\rightarrow$$
 or \leftrightarrow .

3. Every office contains a telephone and either a desktop or a laptop computer.

Every office contains a telephone and either a desktop or a laptop computer.

```
\forall X (office(X) \rightarrow .....)
```

... contains a telephone and either a desktop or a laptop computer.

```
\forall X \text{ (office(X) } \rightarrow \text{(....} \land ...))}
\forall X \text{ (office(X) } \rightarrow \text{(contains(X, telephone) } \land \text{(...)))}
```

... either a desktop or a laptop computer.

4. Every lecturer has at least one office. Every lecturer has at least one office.

$$\forall X (lecturer(X) \rightarrow ...)$$

... has at least one office.

$$\forall X (lecturer(X) \rightarrow \exists Y (office(Y) \land has(X, Y)))$$

At least one
Some map to 3
One
....

An existentially quantified wff is usually like this:

∃X (^)

Or like this:

∃**X** (∨)

That is their principle connective is

A or V.

- 5. No lecturer has more than one office.
- No lecturer has more than one office.
- Not any lecturer has more than one office.
- (There does not exist a lecturer who has more than one office.)
- $\neg \exists L$ (lecturer(L) $\land L$ has more than one office)
- $\neg \exists L$ (lecturer(L) $\land L$ has at least 2 offices that are not the same)

 $\neg \exists L \text{ (lecturer(L)} \land \text{ there are at least 2 offices that } L \text{ has and are not the same)}$

```
\neg\existsL (lecturer(L) \land \existsO1 \existsO2 (office(O1) \land office(O2) \land has(L, O1) \land has(L, O2) \land \negsame(O1,O2)))
```

There are other ways of doing 5. For example:

```
\forallL (lecturer(L) \rightarrow ¬ (\existsO1 \existsO2 (office(O1) \land office(O2) \land has(L, O1) \land has(L, O2) \land ¬same(O1,O2)))
```

```
\forallL \forallO1 \forallO2 (lecturer(L) \land office(O1) \land office(O2) \land has(L, O1) \land has(L, O2) \rightarrow same(O1,O2))
```

6. No lecturers share offices with anyone.

Try it

7. Some lecturers supervise PhD students and some do not.

$$(\exists L) \land (\exists L)$$
 or

Some lecturers supervise PhD students and some do not.

```
(∃L (lecturer(L) ∧ there is at least one PhD student that L supervises)) ∧
(∃L (lecturer(L) ∧ there is not at least one PhD student that L supervises))
(∃L (lecturer(L) ∧ ∃S (phd(S) ∧ supervises(L,S)))) ∧
(∃L (lecturer(L) ∧ ¬∃S (phd(S) ∧ supervises(L,S))))
```

```
(∃L (lecturer(L) \land ∃S (phd(S)\land supervises(L,S))))\land (∃L (lecturer(L) \land ¬∃S (phd(S) \land supervises(L,S)))) Drop some unneccesary brackets:
∃L (lecturer(L) \land ∃S (phd(S)\land supervises(L,S)))\land ∃L (lecturer(L) \land ¬∃S (phd(S) \land supervises(L,S)))
```

8. Each PhD student has an office, but all PhD students share their office with at least one other PhD student.

Try it

- A lecturer is happy if the PhD students he/she supervises successfully complete their PhD.
- ∀L (lecturer(L) ∧ his/her PhD students
 successfully complete → happy(L))
- ∀L (lecturer(L) ∧ all his/her PhD students
 successfully complete → happy(L))
- \forall L (lecturer(L) $\land \forall$ S if S is L's PhD student then S successfully completes \rightarrow happy(L))

```
∀L (lecturer(L) ∧
∀S if S is L's PhD student then S successfully
    completes→ happy(L))
∀L (lecturer(L) ∧
∀S (phd(S) ∧ supervises(L,S) → completePhD(S))
    → happy(L))
```

10.Not all PhD students complete their PhD.

Try it