School of Computing and Information Systems
COMP30026 Models of Computation
Week 5: Translation — Predicate Logic

Homework problems

P5.1 Translate the following formulas into predicate logic, using $D(x)$ for "x is a duck", $M(x)$ for "x is muddy", $F(x, y)$ for "x follows y" and $R(x, y)$ for "x is muddier than y". Use the constant a as a name for Jemima, and b for Louise.

(i) Every duck that follows a muddy duck is muddy.
(ii) Any muddy duck is muddier than any duck that isn't muddy.
(iii) Jemima is muddier than any duck that is muddier than Louise.
(iv) There is a duck that is muddier than Louise, but not as muddy as Jemima.
(v) Given any two ducks, one is muddier than the other.
(vi) Given any three ducks, if the first is muddier than the second, and the second is muddier than the third, then the first is muddier than the third.

Solution:

(i) $\forall x((D(x) \land \exists y(D(y) \land M(y) \land F(x, y))) \to M(x))$
(ii) $\forall x((D(x) \land M(x)) \to \forall y((D(y) \land \neg M(y)) \to R(x, y)))$
(iii) $\forall x((D(x) \land R(x, b)) \to R(a, x))$
(iv) $\exists x(D(x) \land R(x, b) \land \neg R(x, a))$
(v) $\forall x \forall y((D(x) \land D(y)) \to (R(x, y) \lor R(y, x)))$
(vi) $\forall x \forall y \forall z((D(x) \land D(y) \land D(z)) \to ((R(x, y) \land R(y, z)) \to R(x, z)))$

P5.2 Translate the following formulas into predicate logic, using $D(x)$ for "x is a duck", $M(x)$ for "x is muddy", $E(x)$ for "x is an egg", $L(x, y)$ for "x lays y", and $R(x, y)$ for "x is muddier y". Use the constant a as a name for Jemima, and b for Louise.

(i) Not every duck lays an egg.
(ii) Every duck doesn't lay an egg.
(iii) There is a duck that doesn't lay an egg.
(iv) There isn't a duck who lays every egg.
(v) Louise is not muddier than every duck.
(vi) Louise is not muddier than any duck.
(vii) Every egg has a duck who laid it.
(viii) Not every egg is laid by a muddy duck.
(ix) If every duck is muddy, then no duck lays an egg.

Solution:

(i) $\neg\forall x(D(x) \to \exists y(E(y) \land L(x, y)))$
(ii) $\forall x(D(x) \to \neg\exists y(E(y) \land L(x, y)))$
(iii) $\exists x(D(x) \land \neg\exists y(E(y) \land L(x, y)))$
(iv) $\neg\exists x(D(x) \land \forall y(E(y) \to L(x, y)))$
(v) $\neg\forall x(D(x) \to R(b, x))$

(vi) $\forall x(D(x) \to \neg R(b, x))$

(vii) $\forall y(E(y) \to \exists x(D(x) \wedge L(x, y)))$

(viii) $\neg \forall y(E(y) \to \exists x(M(x) \wedge D(x) \wedge L(x, y)))$

(ix) $(\forall x(D(x) \to M(x))) \to (\forall x(D(x) \to \neg \exists y(L(x, y) \wedge E(y))))$

P5.3 For any formula G and variable x, $\neg \forall x\, G \equiv \exists x\, \neg G$, and $\neg \exists x\, G \equiv \forall x\, \neg G$. Interpret the formula $\neg \forall x(D(x) \to \exists y(E(y) \wedge L(x, y)))$ in natural language, then use these equivalences to "push the negation" through each of the quantifiers and connectives, and re-interpret the result in natural language. Reflect on why these are saying the same thing.

Solution: For any formula G and variable x, $\neg\forall xG \equiv \exists x\neg G$, and $\neg\exists xG \equiv \forall x\neg G$. Interpret the formula $\neg\forall x(D(x) \to \exists y(E(y) \wedge L(x, y)))$ in natural language, then use these equivalences to "push the negation" through each of the quantifiers and connectives, and re-interpret the result in natural language. Reflect on why these are saying the same thing. $\neg\forall x(D(x) \to \exists y(E(y) \wedge L(x, y)))$ says "It's not the case that every duck lays an egg". If we push negation all the way in, the resulting equivalent formula is $\exists x(D(x) \wedge \forall y(E(y) \to \neg L(x, y)))$. This says that there is a duck which doesn't lay any egg at all, i.e. taking any particular egg, we claim the duck doesn't lay that egg.

P5.4 In the following formulas, identify which variables are bound to which quantifiers, and which variables are free.

(i) $\forall y(D(x) \wedge \exists x(E(y) \leftrightarrow L(x, y)))$

(ii) $\exists z(E(z) \wedge M(y)) \to \forall y(E(z) \wedge M(y))$

(iii) $\exists x((E(x) \wedge M(y)) \to \forall y(E(x) \wedge M(y)))$

(iv) $\forall z((\exists z\, D(z)) \to D(z))$

(v) $\exists u((D(z) \wedge \forall x(M(x) \to D(x))) \to \forall z(L(x, z)))$

(vi) $\forall x(\forall y(M(x) \to D(x)) \wedge \exists y(D(y) \wedge \forall y(L(y, x))))$

Solution: In the following formulas, identify which variables are bound to which quantifiers, and which variables are free.

(i) In $\forall y(D(x) \wedge \exists x(E(y) \leftrightarrow L(x, y)))$ the first occurrence of x is free and the second is bound to the existential quantifier. Both y's are bound to the universal quantifier

(ii) In $\exists z(E(z) \wedge M(y)) \to \forall y(E(z) \wedge M(y))$, the first occurrence of z is bound to the existential quantifier, and the second is free. The first occurrence of y is free and the second is bound to the universal quantifier.

(iii) In $\exists x((E(x) \wedge M(y)) \to \forall y(E(x) \wedge M(y)))$, both occurences of x are bound to the existential quantifier. The first occurrence of y is free and the second is bound to the universal quantifier.

(iv) In $\forall z(\exists z(D(z)) \to D(z))$, the first occurrence of z is bound to the existential quantifier, and the second is bound to the universal quantifier.

(v) In $\exists u((D(z) \wedge \forall x(M(x) \to D(x))) \to \forall z(L(x, z)))$, the first occurrence of z is free, and the second is bound to the $\forall z$. The first two occurrences of x are bound to the $\forall x$, and the second is free.

(vi) In $\forall x(\forall y(M(x) \to D(x)) \wedge \exists y(D(y) \wedge \forall y(L(y, x))))$, all occurrences of x are bound to the $\forall x$. The first occurrence of y is bound to the $\exists y$, and the second is bound to the last $\forall y$.

P5.5 On the Island of Knights and Knaves, everyone is a knight or knave. Knights always tell the truth. Knaves always lie. You meet three people, let us call them A, B and C. A says to you: "If I am a knight then my two friends here are knaves." Use **propositional** logic to determine whether this statement gives you any real information. What, if anything, can be deduced about A, B, and C?

Solution: Let the propositional variable A stand for "A is a knight" and similarly for B and C. Note that if A makes statement S then we know that $A \leftrightarrow S$ holds. Or we can consider the two possible cases for A separately: Either A is a knight, and A's statement can be taken face value; or A is a knave, in which case the negation of the statement holds, that is,

$$\Big(A \land \big(A \to (\neg B \land \neg C)\big)\Big) \lor \Big(\neg A \land \neg\big(A \to (\neg B \land \neg C)\big)\Big)$$

We can rewrite the implications:

$$\Big(A \land \big(\neg A \lor (\neg B \land \neg C)\big)\Big) \lor \Big(\neg A \land \neg\big(\neg A \lor (\neg B \land \neg C)\big)\Big)$$

Then, pushing negation in and using the distributive laws, we get

$$\Big(A \land (\neg B \land \neg C)\Big) \lor \Big(\neg A \land A \land (B \lor C)\Big)$$

The second disjunct is false, so the formula is equivalent to $A \land \neg B \land \neg C$. So A must be a knight, and B and C are knaves.

P5.6 A graph colouring is an assignment of colours to nodes so that no edge in the graph connects two nodes of the same colour. The graph colouring problem asks whether a graph can be coloured using some fixed number of colours. The question is of great interest, because many scheduling problems are graph colouring problems in disguise. The case of three colours is known to be hard (NP-complete).

How can we encode the three-colouring problem in propositional logic—more precisely, in CNF? (One reason we might want to do so is that we can then make use of a SAT solver to determine colourability.) Using propositional variables

- B_i to mean node i is blue,
- G_i to mean node i is green,
- R_i to mean node i is red;
- E_{ij} to mean i and j are different but connected by an edge,

write formulas in CNF for these statements:

(a) Every node (0 to n inclusive) has a colour.
(b) Every node has at most one colour.
(c) No two connected nodes have the same colour.

For a graph with $n + 1$ nodes, what is the total size of these CNF formulas? (The size of a CNF formula is the number of literals it contains.)

Solution: These are the clauses generated:

(a) For each node i generate the clause $B_i \lor G_i \lor R_i$. That's $n + 1$ clauses of size 3 each.
(b) For each node i generate three clauses: $(\neg B_i \lor \neg G_i) \land (\neg B_i \lor \neg R_i) \land (\neg G_i \lor \neg R_i)$. That comes to $3n + 3$ clauses of size 2 each.
(c) For each pair (i, j) of nodes with $i < j$ we want to express $E_{ij} \to (\neg(B_i \land B_j) \land \neg(G_i \land G_j) \land \neg(R_i \land R_j)$. This means for each pair (i, j) we generate three clauses: $(\neg E_{ij} \lor \neg B_i \lor \neg B_j) \land (\neg E_{ij} \lor \neg G_i \lor \neg G_j) \land (\neg E_{ij} \lor \neg R_i \lor \neg R_j)$. There are $n(n + 1)/2$ pairs, so we generate $3n(n + 1)/2$ clauses, each of size 3.

Altogether we generate $3n + 3 + 6n + 6 + 9n(n + 1)/2$ literals, that is, $9(n + 1)(n/2 + 1)$.