Exercises in Predicate Logic:

1. Prove the following sequents using natural deduction for predicate logic:

   (a) $\exists x\, \exists y\, (H(x,y) \vee H(y,x)),\ \neg\exists x\, H(x,x)\ \vdash\ \exists x\, \exists y\, \neg(x = y)$.

   (b) $\forall x\, P(a,x,x),\ \forall x\, \forall y\, \forall z\, (P(x,y,z) \to\ P(f(x),y,f(z))\ \vdash\ \exists z\, P(f(a),z,f(f(a)))$.

   (c) $\forall x\, \exists y\, P(x,y), \forall x\, \forall y\, P(x,y) \to \neg P(y,x)\ \vdash\ \neg\exists z\, \forall u\, P(u,z)$.

   (d) $\forall x\, (P(x) \to\ (x = b)),\ \forall x\, (x = b) \to\ P(x)\ \vdash\ P(b) \wedge \forall x\, \forall y\, (P(x) \wedge P(y) \to\ (x = y))$.

   (e) $\forall x\, (P(x) \vee Q(f(x))),\ \forall x\, (R(f(x)) \to \neg P(x)),\ \exists x\, R(f(x))\ \vdash\ \exists x\, Q(f(x))$.

   (f) $\forall x\, \forall y\, P(x,y) \wedge P(y,x) \to\ (x = y),\ \forall x\, P(x,f(x)),\ \exists z\, \forall x\, P(x,z)\ \vdash\ \exists u\, (u = f(u))$.

2. Consider the predicate logic formula $\phi(y)\ =\ \forall x(P(x,y) \to \forall x P(x,y))$, with free variable y.

   (a) You are given a model $\mathcal{M}$, consisting of the universe $S = \{a,b,c\}$, and interpretation of the binary predicate given by $P\ =\ \{(a,c),\ (a,a),\ (b,c),\ (b,a),\ (c,c),\ (c,a)\}$. If l is an assignment (or environment) defined by $l(y) = c$, determine whether $\mathcal{M} \models_l \forall y\, \phi(y)$. marks.

   (b) We wish to construct a new structure $\mathcal{M}'$ consisting of the same S as in the previous subquestion, but with a possibly new P', such that $\mathcal{M}' \models_l \neg\phi(y)$. Give a suitable P' for this purpose. marks. *Note that you are required to keep both S and l the same as in the previous questions.*

3. Let Cat(x), Dog(x), Striped(x), Friends(x,y), Equal(x,y) and ShortTempered(x) be predicates to be evaluated over the set of all animals. The predicates have the following obvious interpretations. Cat(x) (Dog(x)) evaluates to True iff x is a cat (dog). Striped(x) evaluates to True iff the coat of x is striped. Friends(x,y) evaluates to True iff x and y are friends; obviously, Friends(x,y) also implies Friends(y,x). Equal(x,y) evaluates to True iff x and y are one and the same animal. ShortTempered(x) evaluates to True iff x is short-tempered.

   Express the following English language sentences as predicate logic sentences (formulae without free variables). You may assume that the domain for evaluating the truth of these sentences is always the set of all animals (which could include animals other than cats and dogs as well).

   (a) There is a short-tempered dog who is not friendly with any other dog, but is friendly with at least one striped cat.

   (b) Every cat that is not striped is friendly with at least one dog that is not a friend of any striped cat.

4. Consider the following predicate logic sentences (formulae with no free variables):

   $\phi\ =\ \forall x(\exists y(E(x,y) \wedge \neg E(y,x)))$
   $\psi\ =\ \forall x(\forall y(\forall z((E(x,y) \wedge E(y,z)) \to\ E(x,z))))$.

   We wish to evaluate these sentences (i.e., determine their truth value) over models obtained from directed graphs. Given a directed graph, $G = (V_G, E_G)$, a model $\mathcal{M}_G$ is obtained by letting the domain of variables be V_G, and by letting predicate $E(x,y)$ evaluate to true if and only if there is a directed edge from x to y in E_G. Since there are no function symbols and only one binary predicate symbol E in ϕ and ψ, every directed graph uniquely defines a model over which ϕ and ψ can be evaluated. For example, if G is the graph in Fig. 1, the corresponding model $\mathcal{M}_G$ has $\{v1, v2\}$ as the domain of variables, and $E(x,y) =$ True if and only if $x = v1, y = v2$ or $x = y = v2$.

   (a) Give two directed graphs G_1 and G_2 with no more than 5 vertices in each such that ϕ evaluates to True over $\mathcal{M}_{G_1}$, and $\phi \to\ \psi$ evaluates to False over $\mathcal{M}_{G_2}$.

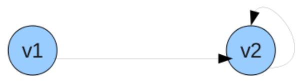

Figure 1:

(b) We now wish to find a directed graph G such that $\phi \wedge \psi$ evaluates to True over $\mathcal{M}_G$. Indicate with justification whether G can be a finite graph (i.e., graph with finite number of vertices) and yet cause $\phi \wedge \psi$ to evaluate to True over $\mathcal{M}_G$.

5. In our lecture, we studied about structures in the context of first-order logic.

All formulae in the following questions (except the second question below) are assumed to be free of free variables, i.e. all variables are bound. Such formulae are also sometimes called *closed formulae* or *sentences*, and play a very important role in the study of logic.

Recall further that a structure $\mathcal{M}$ is called a *model* of a sentence φ if $\mathcal{M} \models \varphi$. We write $\models \varphi$ to denote that φ is valid, i.e., evaluates to true in all Σ-structures, where Σ is the vocabulary of φ.

(a) Give an example of a first-order logic sentence φ such that every model of φ necessarily has an infinite universe. You are allowed to have the equality predicate, i.e "=", as part of your vocabulary. Note that every model $\mathcal{M}$ must interpret "=" as follows: For all $a, b \in U_{\mathcal{M}}$, $=^{\mathcal{M}} (a, b)$ is true iff a and b are the same elements. You must try to minimize the number of predicate and function symbols (other than "=") that you need in the vocabulary.

(b) Let $\Sigma = \{P, a, b\}$ be a signature, where P is a unary predicate, and a and b are nullary functions (or constants). Let $\varphi(x)$ be the first-order logic formula $P(x) \rightarrow (P(a) \wedge P(b))$ with signature Σ. Show that there is no term t that is free for x in φ such that $\models \varphi(t)$. Show also that $\models \exists x\, \varphi(x)$. Is there a contradiction in the above two statements? Explain why.

(c) Let Σ_{gp} be the vocabulary $\{+, 0, =\}$, where "+" is a binary function symbol, "=" is the binary equality predicate, and "0" is a nullary function symbol (or constant). You must resist the temptation of interpretting "+" as arithmetic addition, and "0" as the additive identity in arithmetic. In fact, different Σ_{gp}-structures may assign different interpretations to "+" and "0". However, the interpretation of "=" must stay the same across all structures. For convenience of notation and of reading, we will write $x + y$ instead of $+(x, y)$, and $x = y$ instead of $= (x, y)$ in the following.

Consider the following first-order logic sentences over the vocabulary Σ_{gp}.

- $\varphi_1 \equiv \forall x \forall y \forall z\, (x + (y + z) = (x + y) + z)$
- $\varphi_2 \equiv \forall x\, ((x + 0 = x) \wedge (0 + x = x))$
- $\varphi_3 \equiv \forall x\, (\exists y(x + y = 0) \wedge \exists z(z + x = 0))$

Let $\varphi \equiv \varphi_1 \wedge \varphi_2 \wedge \varphi_3$.

- Show that φ is satisfiable by giving a model.
- Show that φ is not valid by giving a Σ-structure $\mathcal{M}$ such that $\mathcal{M} \not\models \varphi$.
- Show that $\forall x \forall y\, ((x + y) = (y + x))$ is not logically implied by φ.
- Show that the conjunction of any two of φ_1, φ_2 and φ_3 does not logically imply φ.

6. Let Σ_G be the vocabulary $\{E, =\}$ where E is a binary predicate, and "=" is the usual equality predicate. Every Σ_G structure $\mathcal{M} = (U_{\mathcal{M}}, E^{\mathcal{M}})$ can be interpretted as a directed graph, where the elements of $U_{\mathcal{M}}$ are the nodes, and for all $a, b \in U_{\mathcal{M}}$, there exists an edge from a to b iff $E^{\mathcal{M}}(a, b)$ is true. Similarly, every directed graph can be interpretted as a Σ_G structure. Thus, every first-order logic sentence φ on the vocabulary Σ_G can be thought of as describing a set of directed graphs, i.e. the set of graphs that serve as models of φ.

(a) Give a formula φ_1 on the vocabulary Σ_G such that a graph $\mathcal{G}$ is a model of φ_1 iff G has no cliques of size > 5.

(b) Give a formula φ_2 on Σ_G such that there are arbitrarily large finite graphs that serve as models of φ_2, and any finite graph that is a model of φ_2 necessarily has an even number of nodes.

(c) Given any finite graph $\mathcal{G}$, show that there exists a first order logic sentence $\varphi_{\mathcal{G}}$ on the vocabulary Σ_G such thate every model of $\varphi_{\mathcal{G}}$ is necessarily isomorphic to $\mathcal{G}$. Can this result be extended to graphs with countably infinite nodes?

(d) Suppose you are told that the Compactness Theorem holds for first-order logic as well, i.e. given a set of first-order logic sentences that is unsatisfiable, there must exist a finite subset of the set that is unsatisfiable. Can you use this to show that there cannot exist any first-order logic formula φ on the signature Σ_G such that a graph satisfies φ iff it does not contain a cycle? This illustrates one of the many useful applications of Compactness Theorem, whereby we can conclude that something cannot be expressed in first-order logic.

7. Consider the following first-order logic sentences:
$\phi \equiv \forall x\, \exists y\, \forall z\ (P(x, y) \vee \neg P(z, y))$, and
$\psi \equiv \forall x\, ((\exists y\, P(x, y)) \vee (\forall y\, \neg P(y, x)))$.

Prove that $\psi \rightarrow \phi$ by demonstrating that every model $M = (U^M, P^M)$ that satisfies ψ also satisfies ϕ. In other words, we wish to reason about the models of ϕ and ψ in order to prove the semantic entailment of the two sentences.

[Hint: Partition the universe U^M of a model M into two parts, one containing all elements x that satisfy $\exists y\, P(x, y)$, and the other containing all other elements of U^M. Of course, you are welcome to use a different reasoning as well.]

8. In a sleepy little town, there is an old postman who *must* visit each morning all houses/offices that have letter(s) to be delivered that day. There are N houses/offices in the town, and each house/office has a distinct address that is a natural number in the range $[1, N]$. The post office itself has the address P.

The postman, being old, must figure out the "best" sequence of addresses for delivering letters each morning, so that the *total distance* travelled by him *from the post-office to the last address where a letter is delivered and back to the post-office* is minimized. It is assumed that if the postman is currently at address x_i and the next address in the sequence is x_j, then he always takes the shortest path from x_i to x_j. Note that the "best" sequence for delivering letters need not be unique, and there may be letter(s) addressed to the post-office itself.

Interestingly, this is no ordinary post-office! It has recently acquired a software that can analyze certain classes of predicate logic sentences (formulae without free variables) and can compute models of satisfiable sentences using natural numbers (i.e., the domain of elements is the set of natural numbers). Our old postman, having realized the fun of playing with predicate logic, now wishes to use this software for finding the best sequence of delivery addresses each morning. We must help him in this noble endeavour. In particular, we must formulate a satisfiable predicate-logic sentence ϕ (containing no free variables), such that the "best" sequence of addresses can be "extracted" from a model of ϕ using natural numbers. In constructing ϕ, we are allowed to use the following predefined predicates and functions:

- $\mathsf{L}(x_i)$: This predicate takes a natural number x_i as argument, and if x_i lies in the range $[1, N]$, then it evaluates to True iff there is a letter for address x_i. If x_i lies outside the range $[1, N]$, the predicate evaluates to False. Clearly, the definition of $\mathsf{L}(x_i)$ potentially changes every morning (depending on which addresses have letters to be delivered that day). Our postman, in his enthusiasm to use the software, has agreed to look at the pile of letters each morning and update the definition of this predicate on the post-office computer before running the software on ϕ.

- $\mathsf{D}(x_i, x_j)$: This function takes two natural numbers and returns a natural number giving the shortest distance between addresses x_i and x_j in the town, for all $x_i, x_j \in \{1 \dots N\}$. If either argument is outside the range $[1, N]$, the function returns (rather arbitrarily chosen) 13. The town is seismologically stable, and there are no new constructions or demolitions. So the definition of this function is assumed to be time invariant, and does not change from one day to another
- $+(x_i, x_j)$: This function takes two natural numbers and returns their sum.
- $*(x_i, x_j)$: This function takes two natural numbers and returns their product.
- $x_i = x_j$: This is the equality predicate for natural numbers.
- $x_i \geq x_j$: This is the greater-than-or-equal-to predicate for natural numbers.

In addition to the above predefined predicates and functions, you are allowed to use the function symbol Seq of arity 1 (from natural numbers to natural numbers) in formulating ϕ. No other function or predicate symbols may be used in ϕ.

Recall that our noble goal was to to help the old postman "extract" the "best" sequence of addresses every morning. So, we must use the function symbol Seq in ϕ in such a way that if the post-office's software returns a model $M = (\mathbf{N}, \mathsf{Seq}^M)$ for ϕ (where $\mathbf{N}$ is the set of natural numbers and $\mathsf{Seq}^M : \mathbf{N} \to \mathbf{N}$ is a concrete function definition), then Seq^M can be used to obtain a "best" sequence of addresses for delivering letters. In particular, if there are K addresses that have letters to be delivered on a particular day, then $\mathsf{Seq}^M(j)$ must give the j^{th} address in a "best" sequence for that day, for all $j \in [1, K]$. For $j \notin [1, K]$, $\mathsf{Seq}^M(j)$ must be 0. You may assume that there is at least one letter to be delivered each day, i.e. $1 \leq K \leq N$ for each day, although K may vary from day to day.

Note that ϕ must not involve K, since this may indeed differ from day to day, and our old postman cannot enter a new formula to the software each day. N is assumed to be fixed, so you may use N and functions of N in formulating ϕ, but once formulated, our postman must be able to use the same formula ϕ every day, simply by changing the definition of $\mathsf{L}(x_i)$ each morning as described above.

As an example, suppose there are 20 houses/offices in the town, and on a particular day, there are letters for addresses $1, 2$ and 12. Suppose also that the unique best sequence of addresses (for minimizing the total distance travelled) is $(12, 2, 1)$. To find this sequence, the postman goes and updates the definiton of L so that $\mathsf{L}(1) = \mathsf{L}(2) = \mathsf{L}(12) = \mathsf{True}$ and $\mathsf{L}(x_i) = \mathsf{False}$ for all other x_i. Our formula ϕ should be such that with the above definiton of L, a model $M = (\mathbf{N}, \mathsf{Seq}^M)$ of ϕ has $\mathsf{Seq}^M(1) = 12, \mathsf{Seq}^M(2) = 2, \mathsf{Seq}^M(3) = 1$ and $\mathsf{Seq}^M(i) = 0$ for all other i.

(a) Give a predicate logic sentence ϕ for the above purpose, and explain briefly the justification behind the construction of your formula. In your justification, you must argue why any model $(\mathbf{N}, \mathsf{Seq}^M)$ of ϕ must contain a function Seq^M that gives a "best" sequence of addresses.

Important: Your justification **must not** exceed 500 words, and your formula **must** have size linear in N. In other words, the total no. of symbols (variables, predicate-logic operators, function and predicate symbols, parentheses) in your formula must be $O(N)$.

(b) Suppose $N = 5$ in the above question. Suppose further that the distances between addresses are as given in the following symmetric matrix, where bold-faced entried indicate addresses and non-bold-faced entries indicate distances:

	1	**2**	**3**	**4**	**5**
1	0	3	2	10	5
2	3	0	6	1	10
3	2	6	0	3	4
4	10	1	3	0	7
5	5	10	4	7	0

Using the formula you have obtained above (or a suitable variant of it) and the SMT solver Z3 (http://research.microsoft.com/en-us/um/redmond/projects/z3/), find an optimal sequence of addresses for the postman on a day in which every address has a letter to be delivered. In the process, you are expected to read up the documentation of Z3 a bit and use the option that solves for quantified formulae. Your solution should contain the formula fed in to Z3 and the output showing the values of Seq for arguments 1, 2, 3, 4 and 5.

Note: Although this example shows that it is possible to find the shortest Hamiltonian circuit using a first-order logic solver, you will easily see that as N increases, this is not the most efficient way to find the shortest Hamiltonian circuit.

9. A Quantified Boolean Formula (QBF) is a formula of the form

$$\psi \equiv Q_1 x_1 Q_2 x_2 \dots Q_n x_n\, \varphi(y_1, y_2, \dots y_m, x_1, x_2, \dots x_n), \text{ where}$$

$y_1, y_2, \dots y_m, x_1, x_2, \dots x_n$ are propositional variables, $Q_1, Q_2, \dots Q_n$ are existential or universal quantifiers and $\varphi(y_1, y_2, \dots y_m, x_1, x_2, \dots x_n)$ is a CNF propositional formula on the variables $y_1, y_2, \dots y_m$, $x_1, x_2, \dots x_n$. Note that ψ does not strictly follow the syntax of first-order logic formulae. Specifically, variables cannot be combined with propositional connectives in first-order logic; instead they must appear as terms in the arguments of predicates before propositional connectives can be used.

The QBF formula ψ is said to be satisfiable iff the first-order formula

$$\psi' \equiv Q_1 x_1 Q_2 x_2 \dots Q_n x_n\, \varphi(P(y_1), P(y_2), \dots P(y_m), P(x_1), P(x_2), \dots P(x_n))$$

is satisfied by the model $M = (U^M, P^M)$, where $U^M = \{\mathsf{False}, \mathsf{True}\}$, $P^M(\mathsf{False}) = \mathsf{False}$ and $P^M(\mathsf{True}) = \mathsf{True}$. Note that in order for ψ' to be satisfied by M, an environment specifying the values of the free variables $y_1, y_2, \dots y_m$ must also be specified. A satisfiablity checker for the QBF formula ψ outputs the environment for which ψ' is satisfied by M. Since the universe of the model is finite, it is easy to see that satisfiability checking of QBF formulae is decidable.

In this question, we wish to use a QBF satisfiability solver (henceforth, called a QBF-SAT solver) to help us win a board game, whenever possible. The game is as described below.

We have a 3×3 board, divided into 9 squares as shown in the figure below. In this figure, squares on the board are numbered $(0,0)$ through $(2,2)$. The game is to be played between you and an opponent. You are required to insert As in the squares of the board, and your opponent must insert Bs, one at a time, and in alternation.

$(0,0)$	$(0,1)$	$(0,2)$
$(1,0)$	$(1,1)$	$(1,2)$
$(2,0)$	$(2,1)$	$(2,2)$

Figure 2:

Suppose you start the game by first inserting an A in one of the 9 squares. Your opponent then inserts a B in one of the remaining 8 squares. You are now allowed to insert an A only in those remaining squares that cannot be reached by a diagonal from the square in which a B has been placed by your opponent. If you cannot find such an empty square, you lose the game. Otherwise, you can insert an A in an empty square not reachable along a diagonal from the square containing B. Your opponent then inserts a B in one of the remaining 6 squares, and you are again required to insert an A in one of the remaining 5 squares that cannot be reached along a diagonal from any of the squares containing B. If you can find such a square and insert the third A in that square, you win the game. Otherwise, you lose the game. Thus, if you start the game, your goal at each step is to insert an A in one of the unfilled squares that is not reachable along a diagonal from any of the squares already containing a B. You win the game in this case if you can insert three As in this way.

If your opponent starts the game, however, by first filling in a B in one of the 9 squares, then your goal at each step is to insert an A in one of the unfilled squares such that your opponent eventually cant insert three Bs, none of which are reachable along a diagonal from previously inserted As. In this case, you win the game, if you can prevent the opponent from inserting three Bs.

We wish to solve the problem of determining your (winning) sequence of moves in the above board game using a QBF-SAT solver. For this purpose, we will use 27 propositions named $a_{i,j}, b_{i,j}$ and $m_{i,j}$, where $i, j \in \{0, 1, 2\}$. We wish to associate the following meanings with these propositions:

- $a_{i,j}$ is True iff an A has already been inserted in the i^{th} row and j^{th} column of the board in an earlier step.
- $b_{i,j}$ is True iff your opponent has already inserted a B in the i^{th} row and j^{th} column of the board in an earlier move.
- If $m_{i,j}$ is True, then in the current move, you can place an A in the i^{th} row and j^{th} column to eventually win the game.

(a) Give a *QBF* formula φ_1 using the above propositional variables and with $m_{i,j}$'s as free variables, such that if you start the game, and wish to determine a move that will eventually lead you to a win (for all possible moves of the opponent), you can use the following procedure.

- Simplify φ_1 by assigning True and False values for $a_{i,j}$ and $b_{i,j}$ depending on previous moves made by you and the opponent.
- Feed the simplified formula to a QBF-SAT solver.
- If the QBF-SAT solver returns an environment specifying values of the free variables for which the formula can be satisfied, you place an A in the $(i, j)^{th}$ square, if $m_{i,j}$ is set to True in the satisfying environment. Note that there could be several choices of (i, j) from the satisfying assignment, and you can pick one at random.
- If the QBF-SAT solver says that the simplified formula is unsatisfiable, then you give up and declare yourself to be a loser.

(b) Using only the above propositional variables, is it possible to give a propositional logic formula φ_2 that is to be used exactly as in the previous subquestion to determine a move that will eventually lead you to a win, but only if your opponent starts the game. Give justification if your answer is in the negative. Otherwise, describe how would obtain the required formula φ_2, without necessarily giving the formula.

(c) *[Optional]* There are quite a few interesting QBF-SAT solvers available in the public domain (check out http://www.qbflib.org). It might be instructive to try this problem using one of them.