Disclaimer: These exercises are not provided by the module organiser, and to the best of our knowledge are not present in any course materials. While the aim is to ensure these exercises are appropriate and relevant to each module's content, they may not reflect the difficulty or style of question present in upcoming exam papers - you should first exhaust the module resources provided.

Starter Questions

1. State whether the following CNFs are tautologies, contradictions, or neither.
   (a) $\langle\,[\,]\,\rangle$
   (b) $\langle [x, \neg y], [y, \neg z], [z, \neg x] \rangle$
   (c) $\langle [\neg x_3, x_2, \neg x_1, x_2, x_4, x_1], [x_1, \neg x_2, x_1, \neg x_4, x_3, \neg x_4, x_2, \neg x_4], [\top, \bot] \rangle$
2. Consider the boolean function $f(x_1, \ldots, x_5)$ that is true iff the following conditions hold:
   - At least one of x_2, x_3 or x_4 is true.
   - Either both x_1 and x_4 are false, or both x_5 and x_1 are true.
   - Exactly one of x_2 and x_5 is true.

   What is the conjunctive normal form of f?
3. How can it be formally shown that some CNF formula is a contradiction?

Satisfiability

This section looks scary, because most of the questions are trying to get you used to converting problems into a CNF formula. The rest of SAT isn't too bad, so you could save a lot of time attempting the SAT question in the exam.

1. Consider each of the following variants of boolean satisfiability. Determine whether each is known to be solvable in polynomial time. Justify your answers.
   - ANTI-HORN-SAT - Each clause has at most 1 negative literal.
   - HORNS-SAT - Each clause has at most 2 positive literals.
   - ALL-SAT - All n variables are present in each clause.
2. Under which assignments of $x_1, x_2, \ldots, x_n$ are the following CNF formulas satisfied?

   Hint: Think in terms of implication.

   (a)
   $$\left(\bigwedge_{i=1}^{n} \neg x_i \vee x_{i+1}\right) \wedge \left(\bigwedge_{i=1}^{n} x_i \vee \neg x_{i+1}\right)$$

   (b)
   $$\left(\bigvee_{i=1}^{n} x_i\right) \wedge \left(\bigwedge_{i=1}^{n} \bigwedge_{j=i}^{n} \neg x_i \vee \neg x_j\right)$$

   (c)
   $$\left(\bigwedge_{i=1}^{n} x_i \vee x_{f(i)}\right) \wedge \left(\bigwedge_{i=1}^{n} \neg x_i \vee \neg x_{f(i)}\right)$$

   where $f(i) = i (\text{mod } n) + 1$.
3. Given a graph $G = (V, E)$, define a CNF formula $\mathcal{F}(G)$ that is satisfiable iff G is bipartite.
4. You are given an array $A[1 \ldots n]$ consisting of integers from the finite set B. Using variables $x_{i,b}$ to denote the presence of integer b at position i, Define a CNF formula $\mathcal{F}(A, B)$ that is satisfiable iff all elements in A are unique.

   Hint: Your answer to 5(b) may be helpful.
5. A white knight on an $n \times n$ chessboard C is trying to travel from one square $s = (s_1, s_2)$ to another $t = (t_1, t_2)$ over a series of moves. However, other white pieces (which do not move) are already placed on specified squares of the chessboard - the white knight cannot move to these squares. Assuming standard knight movement in chess, define a CNF formula $\mathcal{F}(C, s, t)$ which is satisfiable iff the knight can eventually reach t from s on C.
6. A subset $S \subseteq C$ of a standard 52-card deck of playing cards is 'orderable' if the cards can be arranged from left to right, such that adjacent cards have either the same number or suit.

   (a) What is the size of the largest non-orderable subset S?

   (b) Define a CNF formula which is satisfiable if and only if a subset of cards is orderable.

First Order Logic

1. Determine whether the following statements are true or false, justifying your answer.

   (a) $P(x) \to P(x)$ is valid.

   (b) If $|D| < 3$, then any predicate $P(x, y, z)$ is false.

   (c) There exists a model under which $\forall x(P(x, y) \wedge \neg P(x, y))$ is true.

   (d) There exists a model with $|D| = 1$ in which $P(x, y)$ is satisfiable, but not true.

2. Let ϕ be the formula
$$\forall x(P(x, y) \to \exists z(P(x, z) \wedge P(z, y)))$$

   (a) Give a model with $|D| = \infty$ under which ϕ is true.

   (b) Give a model with $|D| = \infty$ under which ϕ is satisfiable, but not true.

   (c) Is there a model with $|D| = \infty$ under which ϕ is false? Justify your answer.

3. (a) Prove the following tautologies using tableau, resolution, and natural deduction.

   i. $\exists x \forall y P(x, y) \to \forall y \exists x P(x, y)$

   ii. $(\forall x(Q(x) \to R(x)) \wedge \exists x(P(x) \wedge Q(x))) \to \exists x(P(x) \wedge R(x))$

   iii. $\{\forall x \forall y(P(x, y) \to \neg\neg Q(y, x)), \forall y \forall z(\neg Q(y, z) \vee P(y, z))\} \vdash \forall x \forall z(P(x, z) \equiv Q(x, z))$

   (b) Which of the above tautologies are sentences?

Verification

1. Prove the following Hoare triples:

   (a)
$$[z > 1]\; y = 0;\; x = z;\; \text{while}\,(x > 0)\,\{x = x - 1;\; y = y + 2\}\; [y = 2z]$$

   (b)
$$[\top]\; \text{if}\,(y > z)\; \text{then}\,\{y = z\}\; \text{else}\,\{z = y\};\; \text{if}\,(x > y)\; \text{then}\,\{x = y\}\; \text{else}\,\{\}\; [(x \leq y) \wedge (x \leq z)]$$

   (c)
$$[(a \geq 0) \wedge (b > 0)]\; d = a;\; c = 0;\; \text{while}\,(d \geq b)\,\{d = d - b;\; c = c + 1\}\; \left[c = \left\lfloor \frac{a}{b} \right\rfloor\right]$$

   Which of the above programs does not have the weakest pre-condition for its post-condition?

Prolog

1. Consider the following Prolog program:

```
s(X, X):- X < 10.
s(X, Y):- A is X // 10, B is X mod 10, s(A, C), Y is B+C.

d(X, 0):- X < 10.
d(X, Y):- s(X, Z), d(Z, A), Y is A+1.
```

   (a) What is the first result of each of the following queries?

      i. `?- s(-2, Y)`

      ii. `?- s(13, X)`

      iii. `?- s(123456789, A)`

      iv. `?- d(13.4, Y)`

      v. `?- d(199, Z)`

   (b) What does the predicate `d/2` compute? The first argument is a positive integer we consider the input, and the second argument is an integer we consider the output.

2. Consider the following Prolog program:

```
fun([], [], 0).
fun([X|Xs], [], Y):- fun(Xs, [], Z), Y is X+Z.
fun([], [X|Xs], Y):- fun([], Xs, Z), Y is X+Z.
fun([X|Xs], [Y|Ys], Z):- fun(Xs, [Y|Ys], A), fun([X|Xs], Ys, B), Z is A+B.
```

   (a) Draw the Prolog search tree for running the query `fun([1, 2], [3, 4], X).`

   (b) When the query `fun([11, 5, 5], [7, 1, 14], Z).` is run, how many additional calls to `fun/3` are made in the process?

3. Write a Prolog predicate `standardise/2`, which takes a list of numbers as input, transforming them such that they collectively have a mean of 0 and standard deviation of 1 as output. In other words, replace each of the n elements x_i with $\hat{x}_i$, where:

$$\hat{x}_i = \frac{x_i - \mu}{\sigma}$$

$\mu = \frac{1}{n}\sum_{j=1}^{n} x_j$ is the mean of the data, and $\sigma = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(x_i - \mu)^2}$ the standard deviation. You may use standard arithmetic functions and `sqrt/2`, but no other built-in predicates.

Hint: Break the problem down into multiple predicates.