1 Propositional logic, SAT, BDDs

1. Let P be a set of propositions containing a special symbol $\perp$ for false. Show that any propositional formula over P can be written using only $\rightarrow$ (implication). (Hint: Use structural recursion.)

2. Recall that we can get an equisatisfiable CNF formula (using Tseitin's encoding) that is linear in size wrt the original formula. Show that we cannot have an equivalent/equisatisfiable DNF that is linear (or even polynomial) in the original formula unless P=NP.

3. Prove by resolution

   - Consider the sets $A = \{\neg p \vee q, p, r\}$ and $B = \{\neg r \vee s, \neg q \vee \neg s\}$ of clauses. Do a proof by resolution of the unsatisfiability of the conjunction of clauses from $A \cup B$.
   - Prove by resolution that $((q \rightarrow r) \rightarrow p) \rightarrow (r \rightarrow p)$ is a valid formula.
   - Prove by resolution that $((p \rightarrow q) \wedge (q \rightarrow p)) \rightarrow ((p \wedge q) \vee (\neg p \wedge \neg q))$ is a valid formula.

4. Consider the following composite sentences:
   S1: If I solve this problem then I will be happy.
   S2: If I solve this problem then I will get a good grade.
   S3: If I solve this problem then I will be happy and get a good grade.

   Prove by natural deduction that S1 and S2 are *equivalent* to S3. Use propositional symbols to encode basic sentences like "I solve this problem" and indicate which propositional symbols encode which of the basic sentences before you do the proof.

5. Suppose you are given a propositional logic formula φ over propositional connectives $\wedge$, $\vee$, $\rightarrow$, and $\neg$. Let ψ be the negation normal form (NNF) of φ. Given φ, you need to identify the set of literals $\mathcal{L}$ in the correponding NNF ψ of φ.
   (a) Give an algorithm to identify the set of literals $\mathcal{L}$ *without* converting φ into its NNF. Explain the steps of the algorithm clearly. What is the complexity of your algorithm?
   (b) Illustrate an application of your algorithm to the formula $\varphi = (x_1 \vee \neg(x_1 \rightarrow x_2)) \rightarrow \neg x_1$.

6. Let $\varphi = p \wedge q \leftrightarrow r$ and $\psi = (q \rightarrow \neg r) \vee p$
   (a) Write a logically equivalent CNF of φ.
   (b) Write an equi-satisfiable CNF of ψ using Tseitin's encoding.
   (c) Draw the binary decision diagrams of φ, ψ, and $\neg\varphi \wedge \psi$ with the variable ordering $p < q < r$ from top-to-bottom.

2 Equality logic and linear arithmetic

1. Give an algorithm to construct a satisfying assignment to a formula in equality logic given a satisfying assignment to its propositional encoding.

2. Consider the following two programs P_1 and P_2:

Program P_1	Program P_2
$x := a + b;$	$p := (a + b)/2;$
$y := x/2 - c/2;$	$q := p - (c/2);$
return $y;$	return $q;$

(a) Specify the equivalence of P_1 and P_2 using equality logic with uninterpreted functions. Use uninterpreted function symbols F, G, and H to denote $+$, $/$, and $-$ respectively.

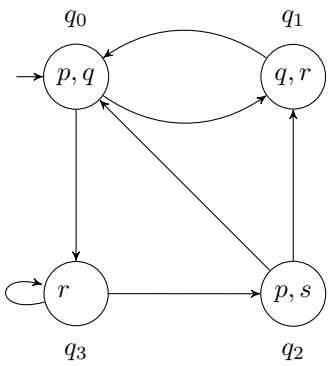
(b) Prove that the formula obtained above is valid using congruence closure. Clearly indicate every step of the closure algorithm.

3. Compute a satisfying assignment to the following system of inequalities using the general simplex method. Clearly indicate pivoting operation, tableau, and assignment at every step.

$$\begin{aligned} 2x_1 + 2x_2 + 2x_3 + 2x_4 &\leq 2 \\ 4x_1 + x_2 + x_3 - 4x_4 &\leq -2 \\ x_1 + 2x_2 + 4x_3 + 2x_4 &= 4 \end{aligned}$$

3 Temporal logic and model checking

1. Consider the transition system TS over atomic propositions $AP = \{p, q, r, s\}$:



Use the CTL model-checking algorithm to compute the states of TS where $\forall\Box(\exists\Diamond(p \vee r))$ holds.

2. Consider the transition system TS in Question 1. Encode the states of TS using boolean variables v_1 and v_2 as follows: $encode(q_0) = \neg v_1 \wedge \neg v_2, encode(q_1) = \neg v_1 \wedge v_2, encode(q_2) = v_1 \wedge \neg v_2, encode(q_3) = v_1 \wedge v_2$. Use the variable ordering $v_1 < v_2$ from top-to-bottom to construct BDDs in the following questions.

(a) For each of the atomic propositions in $AP = \{p, q, r, s\}$ give a BDD representing the states where the proposition holds.

(b) Give BDD for the transition relation $T(v, v')$ of TS. Use the variable ordering $v_1 < v_2 < v_1' < v_2'$ from top-to-bottom to construct the BDD.

(c) Express $\forall\Box(\forall \bigcirc (p \vee q))$ as a fixpoint of an equation. Clearly state whether it is the least or the greatest fixpoint of the equation.

(d) Express $\varphi \equiv \forall \bigcirc p$ as a boolean quantified formula ψ. (Hint: Write the formula φ as a function $\varphi(v_1, v_2)$ and then quantify over successors.)

(e) Let f be the BDD for p. Express ψ in terms of operations over f.

3. We usually assume that every state has some successor in a transition system. Suppose there are *terminal* states in a transition system, that is, states with no successors. We now define a new operator $\exists \bigotimes$. A state q_i satisfies $\exists \bigotimes \psi$ if:

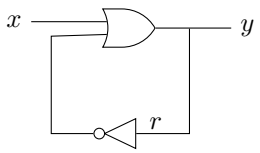
   1. There exists a cycle containing q_i whose all states satisfy ψ or
   2. There exists a path $q_i, \ldots, q_n$ such that q_j, for $i \leq j \leq n$, satisfies ψ and q_n is a terminal state.

(a) Define $\exists \bigotimes \psi$ in terms of the usual CTL operators.

(b) Consider the transition system TS' obtained from TS by deleting the transition (q_0, q_3) making q_0 a terminal state. Model check $\phi \equiv \exists \bigotimes q$ on both TS and TS'.

4. Consider a program f. In program verification, correctness of a program is often defined in term of a precondition and postcondition pair. Intuitively, precondition is a property of initial states of the program and postcondition is a property of final states of the program. Let us model operational semantics of f by a state transition system TS. Consider a set of atomic propositions $P = \{init, final, pre, post\}$ where *init* and $final$ hold only at initial and final states respectively, and *pre* and *post* respectively indicate that the precondition and the postcondition holds. We describe some correctness notions below.
   (a) The program f is *totally correct* if the initial states satisfy the precondition and the program *terminates* with final states that satisfy the postcondition. Give an LTL formula using the set of atomic propositions P that captures total correctness.
   (b) The program f is *conditionally partially correct* if whenever the initial states satisfy the precondition, the final states satisfy the postcondition. However, the program may or may not terminate. Give an LTL formula using the set of atomic propositions P that captures this correctness and a counter-example (a transition system TS and a path π) that violates the formula.

5. Consider the following simple circuit C. Let $P = \{x, r\}$ be the set of atomic propositions denoting boolean values of wires labelled x and r respectively. At the initial state both x and r are false.



   (a) Draw the state transition system TS for the circuit C.
   (b) Write an LTL formula to mean "once x becomes true, before it becomes false, r holds continuously". Make sure that the formula holds for the state transition system TS.
   (c) Write a CTL formula for "there exists a path along which eventually globally x or r holds but not both". Does this formula holds for the transition system TS?

6. Let $\psi \equiv \Box(a \leftrightarrow \bigcirc\neg a)$ and AP $= \{a\}$.
   (a) Show that ψ can be transformed into the following equivalent basic LTL formula

$$\varphi \equiv \neg[true\ U\ (\neg(a \wedge \bigcirc\neg a) \wedge \neg(\neg a \wedge \neg \bigcirc \neg a))]$$

   The syntax of the basic LTL formulae is given by the following context-free grammar:

$$\varphi ::= true \mid a \mid \varphi_1 \wedge \varphi_2 \mid \neg\varphi \mid \bigcirc \varphi \mid \varphi_1 U \varphi_2$$

   (b) Compute all elementary sets with respect to $closure(\varphi)$.
   (c) Construct the GNBA $\mathcal{G}_\varphi$ with $\mathcal{L}_\omega(\mathcal{G}_\varphi) = Words(\varphi)$. To that end:
      1. Define its set of initial states and its acceptance component.
      2. For each elementary set B, define $\delta(B, B \cap AP)$.

7. Prove or disprove the following CTL equivalence: $\forall(p\ W\ q) \equiv \neg\exists[\neg q\ U\ \neg(p \vee q)]$.