CS 228: Logic in CS Autumn 2024

CS 228 End semester Examination

Total Marks: 75 marks *16 November 2024*

Instructions.

- Write your ROLL NUMBER on your answer sheet
- Unless asked for explicitly, you may cite results/proofs covered in class without reproducing them.
- When asked to prove something, give a formal proof with suitable explanation. Similarly, if asked to disprove give a counter-example and argue why it is in-fact a counter-example.
- If you need to make any assumptions, state them clearly.
- Write your answers clearly and legibly. Make sure you write in proper English.
- Whenever you draw an automata make sure the transitions are clearly shown. Also, clearly indicate start states and accepting states. If your automata is clumsy and not clear, you might lose marks.
- **DO NOT COPY SOLUTIONS FROM OTHERS.**
- **Don't let the exam defeat you, defeat the exam. Fight till the end!**

Section A : Ease Your Way Up

1. **SAT Detective.**
   [**5 marks**] Consider a "black-box" SAT solving algorithm $\mathcal{A}$ that takes as input a propositional formula ϕ and returns whether or not ϕ is satisfiable.

   Suppose you have a propositional formula ψ with n variables that you know is satisfiable. Design an algorithm that uses $\mathcal{A}$ as a subroutine to obtain a satisfying assignment for ψ using at most n calls to $\mathcal{A}$.

   To get marks for this questions you should only make at most n calls to $\mathcal{A}$

2. **The Skolem Blueprint.**
   [**6 × 1 = 6 marks**] Skolem Arithmetic is the first-order theory of the natural numbers with multiplication. Formulae in Skolem Arithmetic (SA) use first-order variables $x, y, \dots$ that range over natural numbers, $\mathbb{N} = \{0, 1, 2, \dots\}$ and the ternary multiplication predicate $x * y = z$. This is the only predicate available to you, other than the usual equality predicate $(=)$. Moreover, you are free to use constants from $\mathbb{N}$. Here is a well formed sentence in this logic:

$$\forall x \forall y \exists z \, (x * z = y)$$

Construct formulae in SA which capture the following:

(a) $\mathsf{Divide}(a, b)$: a divides b.

(b) $\mathsf{GCD}(a, b, d)$: d is the GCD (greatest common divisor) of a and b.

(c) $\mathsf{LCM}(a, b, e)$: e is the LCM (least common multiple) of a and b.

(d) $\mathsf{Prime}(p)$: p is a prime number.

(e) $\mathsf{k\text{-}Product}(b)$: b is the product of k primes, where k is a fixed number.

(f) $\mathsf{Prime\text{-}Power}(b)$: b is a power of some prime.

3. **When Skolem meets Resolution.**
[**10 marks**] In this question, we will try to prove the validity of a FO sentence by invoking Herbrand's Theorem. Consider the formula $\phi = ((\phi_1 \wedge \phi_2 \wedge \phi_3) \rightarrow \phi_4)$, where:

$$\begin{aligned}
\phi_1 &= \exists x P(x, g(x)) \\
\phi_2 &= \forall y P(y, f(y)) \\
\phi_3 &= \forall u \forall v \forall w (P(u, v) \wedge P(v, w) \rightarrow P(u, w)) \\
\phi_4 &= \exists z P(z, f(g(z)))
\end{aligned}$$

(a) Give a FO formula ψ in Skolem Normal Form, such that ϕ is valid iff ψ is unsatisfiable. In order to score marks, ψ must be in Skolem Normal Form.

(b) Use Herbrand's Theorem and resolution to show that ψ is indeed unsatisfiable. In your solution, you must clearly show the following:

- Elements of the Herbrand Universe (ground terms) that are used in your proof.
- The finite set of ground clauses whose conjunction is unsatisfiable.
- Resolution proof for the propositional unsatisfiability of the above set of ground clauses.

Section B: At the Top of the World

4. **Be Consistent and you will be Satisfied.**
[**6+4+6 = 16 marks**] Each formula in propositional logic is categorized as either an α (conjunctive) or β (disjunctive) type formula. α formulae (resp. β) have sub-formulae commonly denoted as α_1 and α_2 (resp. β_1 and β_2). In simple terms, an α formula is one which can be written as the conjunction of two other formulae α_1 and α_2. Similarly, a β formula can be written as a disjunction of β_1 and β_2. See Table 1 and 2.

Formula	α_1	α_2
$A \wedge B$	A	B
$\neg(A \vee B)$	$\neg A$	$\neg B$
$\neg(A \rightarrow B)$	A	$\neg B$

Table 1: **Conjunctive (α) Formulas**

Formula	β_1	β_2
$A \vee B$	A	B
$A \rightarrow B$	$\neg A$	B
$\neg(A \wedge B)$	$\neg A$	$\neg B$

Table 2: **Disjunctive (β) Formulas**

We will use some definitions before solving questions.

Definition 1 (Consistency Property) *Let $\mathcal{C}$ be a (possibly infinite) collection of sets of formulae. $\mathcal{C}$ is a consistency property if each $S \in \mathcal{C}$ satisfies the following:*

(a) For each propositional variable p, S does not contain both p and $\neg p$.

(b) $\bot \notin S$ and $\neg\top \notin S$.

(c) If $\neg\neg\phi \in S$, then $S \cup \{\phi\} \in \mathcal{C}$.

(d) If $\alpha \in S$, then $S \cup \{\alpha_1, \alpha_2\} \in \mathcal{C}$ for any alpha formula α with sub-formulae α_1, α_2.

(e) If $\beta \in S$, then $S \cup \{\beta_1\} \in \mathcal{C}$ or $S \cup \{\beta_2\} \in \mathcal{C}$ for any beta formula β with sub-formulae β_1, β_2.

Notice that $\mathcal{C}$ is itself a set of sets and any $S \in \mathcal{C}$ can be possibly infinite.

Definition 2 *A propositional consistency property is called* ***subset closed*** *if it contains, with each member, all subsets of that member.*

(a) Show that every consistency property $\mathcal{C}$ can be extended to a consistency property that is subset closed, i.e, for any consistency property $\mathcal{C}$, there exists a superset $\mathcal{C}^+ \supseteq \mathcal{C}$ such that $\mathcal{C}^+$ is a subset-closed consistency property.

(b) **Definition 3** *A propositional consistency property, $\mathcal{C}$, has finite character if*

$S \in \mathcal{C}$ iff every finite subset of S is in $\mathcal{C}$.

Show that every propositional consistency property of finite character is subset closed.

(c) **Theorem 4 (Model Existence Theorem)** *If $\mathcal{C}$ is a propositional consistency property and $S \in \mathcal{C}$, then S is satisfiable.*

Theorem 5 (Propositional Compactness Theorem) *Let S be a set of propositional formulae. Then S is satisfiable if and only if every finite subset of S is satisfiable.*

Using Model Existence Theorem, prove Propositional Compactness Theorem.

You are allowed to use only ***Model Existence Theorem****. Any other proof will get zero marks.*

5. **Perfect Channel Magic**

In this semester, you have studied finite state machines which have a reasonable expressive power. In this question, let us try to reason about the expressive power of finite state machines with some enhancements.

(1) **[5 + 7 = 12 marks]** A *channel machine* $\mathcal{C}$ is a finite state automaton which is equipped with a channel ρ. It is formally defined as $\mathcal{C} = (Q, \Sigma, \Gamma, \delta, q_0, F)$ where Q is a finite set of states, Σ is a finite input alphabet, Γ is a finite channel alphabet, $\delta \subseteq Q \times \Sigma \cup \{\epsilon\} \times \mathsf{Op} \times Q$ is the transition relation over $\Sigma \cup \{\epsilon\}$ and the set $\mathsf{Op} = \{!a, ?a, nop \mid a \in \Gamma\}$ and $F \subseteq Q$ is the set of final states. Here, $!a$ denotes writing a to the tail of the channel ρ, $?a$ denotes reading a from the head of channel ρ, and *nop* represents no operation on the channel. To begin, $\mathcal{C}$ is in the initial configuration (q_0, ϵ) representing being in the initial state q_0 with an empty channel. The transitions can be seen as follows.

(1) $(q, x) \xrightarrow{a,!b} (q', xb)$ where $a \in \Sigma \cup \{\epsilon\}, b \in \Gamma, x \in \Gamma^*$ if $(q, a, !b, q') \in \delta$.

(2) $(q, bx) \xrightarrow{a,?b} (q', x)$ where $a \in \Sigma \cup \{\epsilon\}, b \in \Gamma, x \in \Gamma^*$ if $(q, a, ?b, q') \in \delta$.

(3) $(q, x) \xrightarrow{a,nop} (q', x)$ where $a \in \Sigma \cup \{\epsilon\}, x \in \Gamma^*$ if $(q, a, nop, q') \in \delta$.

A configuration is accepting if it has the form $(q, x) \in F \times \Gamma^*$. A run of $\mathcal{C}$ on a word $w \in \Sigma^*$ is a sequence of configurations starting from the initial configuration which reads w, and ends in some configuration. An accepting run is one which ends in an accepting configuration. The language accepted by $\mathcal{C}$ consists of all words $w \in \Sigma^*$ such that w has a run from the initial configuration to an accepting configuration. In the following, let $\Sigma = \{a, b, c\}$.

(a) Construct a channel machine accepting the language $\{a^m b^{2m} c^m \mid m \geq 1\}$.

(b) Construct a channel machine accepting the language $\{ww^R \mid w \in \Sigma^*\}$, where w^R denotes the reverse of w. For example, $abccba$ must be accepted.

(2) **[4 + 12 = 16 marks]** Let us now recall automata over infinite words. Let Q be a finite set of states and Σ be a finite alphabet. A run is an infinite sequence of states, $r = r_1 r_2 r_3 \cdots \in Q^\omega$. Let $inf(r) = \{q \in Q \mid \text{for infinitely many } i,\ r_i = q\}$. Note that $inf(r) \subseteq Q$ and is non-empty. A run r is accepting iff the set $inf(r)$ satisfies an acceptance condition α.

A set $S \subseteq Q$ satsifies a Büchi acceptance condition $\alpha \subseteq Q$ iff $S \cap \alpha \neq \emptyset$. Dually, a set $S \subseteq Q$ satsifies a co-Büchi acceptance condition $\alpha \subseteq Q$ iff $S \cap \alpha = \emptyset$. Deterministic Büchi Automata(DBA), Non-determinictic Büchi Automata (NBA) have Büchi acceptance while Deterministic co-Büchi Automata (DCA) and Non-deterministic co-Büchi Automata (NCA) have co-Büchi acceptance.

(a) Prove or disprove : a language L can be recognized by a DBA iff its complement can be recognized by a DCA.

(b) We define an infinite family of languages $L_1, L_2, \ldots$ over the alphabet $\Sigma = \{a, b\}$. For every $k \geq 1$, define

$$L_k = \{w \in \Sigma^\omega \mid \text{ both } a, b \text{ appear at least } k \text{ times in } w\}$$

Construct an NBA with $2k + 1$ states accepting L_k and an NCA with $3k + 1$ states accepting L_k.

6. **Simple Modalities, Intractable Problems**

[10 marks] We have studied LTL in class. Recall that LTL had modalities $\bigcirc, \mathsf{U}$ as well as derived ones $\Diamond, \Box$. Consider the fragments $LTL[\bigcirc], LTL[\Diamond]$ which restricts LTL to respectively use only the $\bigcirc, \Diamond$ modality. Indeed, you can use all the Boolean connectives $\wedge, \vee, \neg$ as before.

Thus, $p \wedge \bigcirc(q \Rightarrow \neg \bigcirc r) \in LTL[\bigcirc]$, but $p \wedge \bigcirc(q \mathsf{U} (r \wedge \neg \Box q)) \notin LTL[\bigcirc]$. Likewise, $\Diamond p \vee (a \mathsf{U} \bigcirc q) \notin LTL[\Diamond], LTL[\bigcirc]$, while $\Diamond[p \wedge \Diamond q] \in LTL[\Diamond]$.

In the last tutorial, we established a weak lower bound (co-NP hardness) for LTL model checking, by reducing the HPP problem to the complement of the model checking problem, that is, $TS \nvDash \psi$.

In this question, we explore $LTL[\bigcirc], LTL[\Diamond]$ model checking, that is, given a transition system TS and a formula $\psi \in \{LTL[\bigcirc], LTL[\Diamond]\}$, does $TS \models \psi$? Prove that the model checking problem for $LTL[\bigcirc]$ and $LTL[\Diamond]$ are co-NP hard.

To achieve this, start with an instance of a propositional logic formula φ in 3-DNF (that is, a formula in DNF such that each clause has exactly 3 literals), and construct a transition system TS as well as a formula $\psi \in LTL[\bigcirc]$ (respectively, $\psi \in LTL[\Diamond]$) and show that φ is valid iff $TS \models \psi$. DNF stands for disjunctive normal form.

You can give two independent reductions starting with φ, that is construct a transition system TS_1 and a formula $\psi_1 \in LTL[\bigcirc]$ such that $TS_1 \models \psi_1$ iff φ is valid, and yet another transition system TS_2 and a formula $\psi_2 \in LTL[\Diamond]$ such that $TS_2 \models \psi_2$ iff φ is valid.