Homework #1

CS 474: Logic in Computer Science: Fall 2024

Due on Friday, September 20

Points: Problem 1: 20 points, Problem 2: 10+10+5+5=30 points; Total: 50 points

Problem 1. Consider the following syntax for propositional logic.
Fix a countable infinite set of propositions $P = \{p_0, p_i, \ldots\}$.

$$\alpha, \beta ::= p_i \ \mid \ (\alpha \vee \beta) \ \mid \ (\neg \alpha)$$

where $p_i \in P$.

The length of a formula is, roughly speaking, the number of symbols occurring in it. For example, $\mathit{len}(\ (\neg(p \vee q))\ ) = 8$ Here is a formal recursive definition of length:

$$\mathit{len}(p_i) = 1$$

$$\mathit{len}(\ (\alpha \vee \beta)\ ) = \mathit{len}(\alpha) + \mathit{len}(\beta) + 3$$

$$\mathit{len}(\ (\neg \alpha)\ ) = \mathit{len}(\alpha) + 3$$

The number of occurrences of Boolean operators ($\vee$ and $\neg$) in a formula is the natural one you'd think of. For example, $\mathit{len}(\ (\neg(p \vee q))\ ) = 2$. Formally, it is defined as:

$$\mathit{occ}(p_i) = 0$$

$$\mathit{occ}(\ (\alpha \vee \beta)\ ) = \mathit{occ}(\alpha) + \mathit{occ}(\beta) + 1$$

$$\mathit{occ}(\ (\neg \alpha)\ ) = \mathit{occ}(\alpha) + 1$$

We want to show that that the length of a formula is linear in the number of occurrences of Boolean operators. More precisely, we want to show that there are some constants m, n such that for every formula α,

$$\mathit{len}(\alpha) \leq m \cdot \mathit{occ}(\alpha) + n$$

Hypothesize (choose carefully) a particular m, n such that the above holds, and prove the above formally using induction on structure of formulas.

Problem 2. Assume there are a countably infinite number of people in this world $P =]p_1, p_2, \ldots, \}$ and a countable number of dogs $D = \{d_1, d_2, \ldots, \}$. We want each person to adopt *two* dogs as pets.

Each person $p \in P$ expresses their wish that the pets they adopt come from a (finite) set of dogs $D_p \subseteq D$, where D_p contains at least two dogs of course.

Unfortunately, not all dogs get along. Every dog d has a finite set of dogs $E_d \subseteq D$ that it doesn't get along with.

An adoption plan for a set of people $P' \subseteq P$ is a pairing of each person with at least two dogs such that (a) the dogs adopted by a person p are dogs the person wants to adopt (i.e., belong to D_p), (b) no dog is paired with two people, (c) any two dogs that are paired with a single person must get along (more precisely, if d and d' are allocated to a person, then $d' \notin E_d$ and $d \notin E_{d'}$).

We also happen to know that for any finite subset of people, there is an adoption plan for that set of people.

We want to prove the following claim:

Claim: There is an adoption plan for all people P.

Fix arbitrary sets D_p and E_d, for each $p \in P$ and $d \in D$. Asssume $|D_p| \geq 2$ for each D_p.

(a) Write down an infinite set of propositional logic formulae Γ such that Γ is satisfiable iff there is an adoption plan.

You may want to use propositions $q_{p,d}$ for each person $p \in P$ and dog $d \in D$, with the meaning that this proposition is true iff person p is assigned dog d in an adoption plan. You may also want to only use propositions $q_{p,d}$ where $d \in D_p$.

Make sure each formula in Γ is of finite length! Γ is an infinite set, of course. Also, ensure you capture all constraints that adoption plans impose: (1) each person is assigned at least two dogs, (2) no dog is paired with multiple people, (3) dogs are paired with only people who want them, and (4) the dogs paired with a person must get along,

(b) Using compactness theorem argue why the claim is true.

(c) Using your encoding, encode the following constraints as a set of propositional formulae, and check using a SAT solver whether the formula is satisfiable.

People $P = \{a, b, c\}$; $D = \{u, v, w, x, y, z\}$, and $D_a = \{u, v, w, x, y, z\}$, $D_b = \{y, z\}$, $D_c = \{w, x\}$ with $E_v = \{x\}$ and all other E_d sets are empty.

You expect your SAT solver to say your formula is satisfiable as there is an adoption plan.

(d) Do the same as above for the following:

Same P, D, and D_p sets as above but with $E_v = \{u\}$ and other E_d being empty.

Your formula should be unsatisfiable as there is no adoption plan.

You can install and use the SAT/SMT solver Z3. Include the Z3 file and the resul it gives.