COMPSCI 250 Fall 2019 - First Midterm

Marius Minea 10 October 2019

Question 1 (25):

On their morning walks, Cardie and Duncan always visit Simple Gifts Farm, home of the busy farm dogs Danish (d) and Scout (s). In one recent four-day period, each of the farm dogs in the set $F = \{d, s\}$ was assigned zero or more tasks from the set T each day, so that the relation A is a subset of $F \times T \times D$ and $A(x, y, z)$ means "farm dog x was assigned task y on day z".

The set T consists of the tasks Deter Vermin (DV), Greet Cardie and Duncan (GCD), Supervise Chicken Feeding (SCF), and Supervise Squash Harvest (SSH). The set D consists of Monday, Tuesday, Wednesday, and Thursday.

a. (10p) Translate the following sentences to quantified statements in predicate logic.

1) There is a task that was assigned to both farm dogs every day.

$\exists y : \forall x : \forall z : A(x, y, z)$

2) Scout was never assigned SCF, and Danish was never assigned SSH.

$\neg(\exists z : A(s, \text{SCF}, z)) \wedge \neg\exists z : A(d, \text{SSH}, z)$, or $(\forall z : \neg A(s, \text{SCF}, z)) \wedge \forall z : \neg A(d, \text{SSH}, z)$.

3) SSH was assigned to a farm dog on exactly the days that were not Tuesday,
and SCF was assigned to a farm dog on exactly the days that were not Wednesday.

$(\forall z : (z \neq \text{Tue}) \leftrightarrow \exists x : A(x, \text{SSH}, z)) \wedge \forall z : (z \neq \text{Wed}) \leftrightarrow \exists x : A(x, \text{SCF}, z)$

It is essential to use equivalence. The implication $(z \neq \text{Tue}) \rightarrow \exists x : A(x, \text{SSH}, z)$ is still true if the task is also assigned on a Tuesday. The existential quantifier for dog is inside the universal quantifier for days: for every considered day there is a (some) dog doing the task, it need not be the same dog every day. This also means $\exists$ must be inside the equivalence, not outside. The incorrect form $\exists x : (z \neq \text{Tue}) \leftrightarrow A(x, \text{SSH}, z)$ fixes the duties of *one* dog x to do SSH every day except Tuesday, but other dogs may do as they please (including SSH on Tuesday, whenit is said it's not assigned). In the correct version $(z \neq \text{Tue}) \leftrightarrow \exists x : A(x, \text{SSH}, z)$ no dog does SSH on Tuesdays, and some (possibly different) dog does it on every other day. You can also see the difference expanding:

$$
\begin{aligned}
& (z \neq \text{Tue}) \leftrightarrow \exists x : A(x, \text{SSH}, z) \\
= \;& ((z \neq \text{Tue}) \rightarrow \exists x : A(x, \text{SSH}, z)) \wedge ((\exists x : A(x, \text{SSH}, z)) \rightarrow (z \neq \text{Tue})) \\
= \;& ((z = \text{Tue}) \vee \exists x : A(x, \text{SSH}, z)) \wedge (\neg(\exists x : A(x, \text{SSH}, z)) \vee (z \neq \text{Tue})) \\
= \;& ((z = \text{Tue}) \vee \exists x : A(x, \text{SSH}, z)) \wedge ((\forall x : \neg A(x, \text{SSH}, z)) \vee (z \neq \text{Tue}))
\end{aligned}
$$

We see this has a universal quantifier, whereas the wrong translation $\exists x : (z \neq \text{Tue}) \leftrightarrow A(x, \text{SSH}, z)$ doesn't. We should try to translate as close as possible to the original statement ("what is equivalent to it being a Tuesday? Some farm dog being assigned SSH"), and only start introducing quantifiers over the scope that requires them, not all in front (though in simple cases, the latter may work).

4) On any two different days, there exists a farm dog that was assigned GCD on one day but not the other.

$\forall z_1 : \forall z_2 : z_1 \neq z_2 \rightarrow \exists x : A(x, \text{GCD}, z_1) \oplus A(x, \text{GCD}, z_2)$

We need to use $\oplus$, rather than $\wedge\neg$, since we don't know which day has it assigned and which not.

5) Every farm dog has a day on which they were assigned exactly three tasks.

$\forall x$: $\exists z$: $\exists y_1$: $\exists y_2$: $\exists y_3$: $(y_1 \neq y_2) \wedge (y_2 \neq y_3) \wedge (y_1 \neq y_3) \wedge \forall y$: $A(x, y, z) \leftrightarrow ((y = y_1) \vee (y = y_2) \vee (y = y_3))$.

We use equivalence to state that a task is assigned iff it is only one of the three tasks.

A short elegant form, only equivalent as we have four days in total, adapts expressing uniqueness: one unassigned task means that the other three are: $\forall x : \exists z : \exists w : \forall y : (y \neq w) \leftrightarrow A(x, y, z)$. Here, $y = w$ makes the task assignment false, every other value makes it true.

b. (15p) Show that the four premises (1)-(4) together imply the goal (5).
You may use a combination of formulas and English, but make the use of quantifier rules clear.

First, we show the task at (1) cannot be GCD, SCF, or SSH.

By Instantiation, (1) gives $\forall x : \forall z : A(x, t, z)$ (6) for some task t (a constant). Specifying x to s and d gives us $\forall z : A(s, t, z)$ (6a) and $\forall z : A(d, t, z)$ (6b).

From (2) we get by separation $\neg\exists z : A(s, \text{SCF}, z)$ and $\neg\exists z : A(d, \text{SSH}, z)$. These can be rewritten as $\forall z : \neg A(s, \text{SCF}, z)$ (7a) and $\forall z : \neg A(d, \text{SSH}, z)$ (7b).

If we had $t = \text{SCF}$, specifying z to the same variable in (6a) and (7a) would give a contradiction. Likewise, for $t = \text{SSH}$, we'd get a contradiction from (6b) and (7b). Thus, $t \neq \text{SCF} \wedge t \neq \text{SSH}$.

In (4), we specify $z_1 = \text{Tue}$ and $z_2 = \text{Wed}$, and instantiate x and get $A(c, \text{GCD}, \text{Tue}) \oplus A(c, \text{GCD}, \text{Wed})$ for some dog c (whose identity we don't know). This implies $\neg(A(c, \text{GCD}, \text{Tue}) \wedge A(c, \text{GCD}, \text{Wed}))$. Specifying x to c and successively z to Tue and Wed in (6) gives us $A(c, t, \text{Tue}) \wedge A(c, t, \text{Wed})$. Having $t = \text{GCD}$ would lead to a contradiction, thus $t \neq \text{GCD}$. The task t is thus DV.

We can use separation on (3) and get two conjuncts (3a), and (3b). Since $A(d, \text{SSH}, z)$ is always false (2), in (3a) the only dog that can do SSH every day but Tuesday is Scott, and likewise the dog assigned SCF every day but Wednesday must be Danish.

Statement (4) implies there cannot be two distinct days a, b with the same assignment of GCD to the dogs. This would make $A(x, GCD, a) \oplus A(x, GDC, b)$ false for any x, contradicting the statement $\exists x : A(x, GCD, a) \oplus A(x, GDC, b)$ obtained from (4) by specification. Thus, each of the four possible assignments (GCD to both, one, the other, and none) happens on one day. This was the key step of the proof – a number of writeups stated without proof that every dog does GCD on two days, and some said that each day, GCD is covered by exactly one dog, which is false.

We already know that each dog has three of four days on which they do exactly two of DV, SCF and SSH. Since each dog does GCD on two days out of four, at least one of those must overlap one of the three days when they have two other tasks, completing the proof.

Question 2 (15):

Consider the following compound proposition:

$(d \to \neg(c \to a)) \wedge (\neg d \to (c \to b)) \wedge (a \oplus b) \wedge ((a \vee c) \to \neg b) \wedge (c \to (a \wedge \neg d))$

a. (8p) Convert it to conjunctive normal form (a conjunction of clauses, which are disjunctions of propositions or their negations).

$$
\begin{aligned}
& (d \to \neg(c \to a)) \wedge (\neg d \to (c \to b)) \wedge (a \oplus b) \wedge ((a \vee c) \to \neg b) \wedge (c \to (a \wedge \neg d)) \\
\leftrightarrow\ & (d \to (c \wedge \neg a)) \wedge (d \vee \neg c \vee b) \wedge (a \vee b) \wedge (\neg a \vee \neg b) \wedge (\neg(a \vee c) \vee \neg b) \wedge (\neg c \vee (a \wedge \neg d)) \\
\leftrightarrow\ & (\neg d \vee (c \wedge \neg a)) \wedge (d \vee \neg c \vee b) \wedge (a \vee b) \wedge (\neg a \vee \neg b) \wedge ((\neg a \wedge \neg c) \vee \neg b) \wedge (\neg c \vee a) \wedge (\neg c \vee \neg d) \\
\leftrightarrow\ & (\neg d \vee c) \wedge (\neg d \vee \neg a) \wedge (d \vee \neg c \vee b) \wedge (a \vee b) \wedge (\neg a \vee \neg b) \wedge (\neg c \vee \neg b) \wedge (\neg c \vee a) \wedge (\neg c \vee \neg d)
\end{aligned}
$$

b. (7p) Find an assignment to atomic propositions that makes it true, or show there is none.

From conjuncts 1 and 8 we get $(c \vee \neg d) \wedge (\neg c \vee \neg d) \leftrightarrow \neg d$. So $d = \text{F}$.

The formula simplifies to: $(\neg c \vee b) \wedge (a \vee b) \wedge (\neg a \vee \neg b) \wedge (\neg c \vee \neg b) \wedge (\neg c \vee a)$

From conjuncts 1 and 4 we get $(\neg c \vee b) \wedge (\neg c \vee \neg b) \leftrightarrow \neg c$, thus $c = \text{F}$.

We are left with: $(a \vee b) \wedge (\neg a \vee \neg b)$.

One satisfying assignment is $a = \text{F}, b = \text{T}, c = d = \text{F}$. The other is $a = \text{T}, b = c = d = \text{F}$.

Question 3 (20):

Let C be a set of finite languages over an alphabet Σ. Define a relation on C as follows: $P(X, Y)$ if $X = Y$ or every string in X is a proper substring of some string in Y.

a. (10p) Show that P is a partial order on C.

P is a partial order iff it is reflexive, antisymmetric, and transitive.

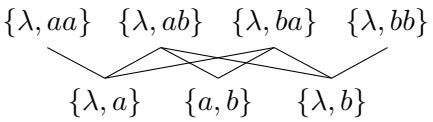
Substituting $Y = X$ in the definition of $P(X, Y)$ we get $X = X$, which is true, thus $P(X, X)$ for arbitrary X, and by Generalization $\forall X : P(X, X)$, thus P is reflexive.

Antisymmetry means $\forall X : \forall Y : P(X, Y) \wedge P(Y, X) \rightarrow X = Y$. Let $\ell_{\max}(X)$ be the maximum length of any string in X (it is essential for X to be finite for this to be well-defined). Take X, Y arbitrary, assume for proof by contradiction that $X \neq Y$. Then $P(X, Y)$ implies that every string in X, including the longest, is a proper substring of a string in Y (which must be longer), thus $\ell_{\max}(Y) > \ell_{\max}(X)$. In the same way, $P(Y, X) \rightarrow \ell_{\max}(X) > \ell_{max}(Y)$. We have a contradiction and thus, we have shown $P(X, Y) \wedge P(Y, X) \rightarrow X = Y$, which gives us antisymmetry by Generalization.

For transitivity, take X, Y and Z with $P(X, Y) \wedge P(Y, Z)$. If either $X = Y$ or $Y = Z$, $P(X, Z)$ follows by substitution. Otherwise, an arbitrary string $x \in X$ must be a proper substring of $y \in Y$, which in turn is a proper substring of $z \in Z$, thus x is a proper substring of z, and we have $P(X, Z)$. Transitivity follows by Generalization.

b. (10p) Take C to be the set of all languages over $\{a, b\}$ that have exactly two strings, and the sum of their lengths is at most 2. Draw a Hasse diagram for the the order P on C.

To satisfy the constraints, both lengths must be 1, or one string must be λ. The Hasse diagram is:



Question 4 (20):

a. (10p) Perform the extended Euclidean algorithm on numbers 104 and 41, and find the inverse of 104 modulo 41 and the inverse of 41 modulo 104, if they exist. Clearly state your answer.

We find successive remainders, and then linear combinations:

	$41 = 0 \cdot 104 + 1 \cdot 41$
$104 - 2 \cdot 41 = 22$	$22 = 1 \cdot 104 - 2 \cdot 41$
$41 - 1 \cdot 22 = 19$	$19 = -1 \cdot 104 + 3 \cdot 41$
$22 - 1 \cdot 19 = 3$	$3 = 2 \cdot 104 - 5 \cdot 41$
$19 - 6 \cdot 3 = 1$	$1 = -13 \cdot 104 + 33 \cdot 41$

Thus, 33 is the inverse of 41 modulo 104, and -13 (or 28) is the inverse of 104 modulo 41.

b. (10p) Solve the congruence system $x \equiv -12 \pmod{104}$, $x \equiv 13 \pmod{41}$.
Does it have a positive solution less than 1000? If so, which?

$$\begin{aligned}
\text{The solution is:} \quad & 13 \cdot (-13) \cdot 104 + (-12) \cdot 33 \cdot 41 \pmod{41 \cdot 104} \\
= \; & (-169 \bmod 41) \cdot 104 + (-4) \cdot (99 \bmod 104) \cdot 41 \pmod{41 \cdot 104} \\
= \; & -5 \cdot 104 + (-4) \cdot (-5) \cdot 41 \pmod{41 \cdot 104} \\
= \; & -520 + 820 \pmod{41 \cdot 104} \\
= \; & 300 \pmod{4264}
\end{aligned}$$

The set of solutions is the class of 300 (mod 4264). 300 is the only positive solution less than 1000. We can check that $300 = 7 \cdot 41 + 13$ and $300 = 3 \cdot 104 - 12$.

Question 5 (30):

The following are fifteen true/false questions, with no explanation needed or wanted, no partial credit for wrong answers, and no penalty for guessing.

a. $((p \oplus q) \oplus p) \leftrightarrow q$ is a tautology.
True. $\oplus$ is associative and commutative, and $p \oplus p = \mathrm{F}$, $q \oplus \mathrm{F} = q$

b. $(p \wedge (p \to q)) \leftrightarrow (p \wedge q)$ is a tautology.
True. $\to$: we get q from Modus Ponens; $\leftarrow$ is obvious.

c. The proposition $(p \to q) \vee (r \to s)$ has 9 true entries in the truth table.
False. It has fifteen – it's only false when both implications are false, $p = r = \mathrm{T}; q = s = \mathrm{F}$

d. If I assume P and prove Q; then separately I assume Q and prove R; then separately I assume $\neg R$ and prove $\neg P$, then P, Q and R are all equivalent.
False. The last is equivalent to $P \to R$. We could have $P = \mathrm{F}, R = \mathrm{T}$, and Q arbitrary.

e. The statement $(A \Delta B) \Delta (A \Delta C) = (B \Delta C) \setminus A$ is a set identity.
False. It should be $B \Delta C$. Δ is associative and $A \Delta A$ cancels out, like $\oplus$.

f. The statements $\exists x : P(x) \vee Q(x)$ and $(\exists x : P(x)) \vee (\exists x : Q(x))$ are equivalent.
True. This can be proved by cases, using Instantiation, Joining, and Existence.

g. If A is a set of strings, all of length k, B is a set of strings, all of length l, and $k \neq l$, then AB and BA have the same number of elements.
True. All strings in AB are distinct, the only way to get length $k + l$ is from k and l, likewise BA, so $|AB| = |BA| = |A| \cdot |B|$. See also P2.5.7 on HW2.

h. If a relation is total, symmetric and transitive, it must be reflexive.
True. For any x, we must have some y with $R(x, y)$, thus $R(y, x)$, and by transitivity $R(x, x)$.

i. If $f : A \to B$ is any function, then the sets $\{x \in A : f(x) = b\}$ for $b \in B$ define a partition of A.
True. Everys $x \in A$ is in the class of $f(x) \in B$. $f(x)$ is well-defined, so x can't be in two sets.

j. If $f : A \to B$ is one-to-one and $g : B \to C$ is bijective, then $g \circ f$ is bijective.
False. It is one-to-one, but we may have $|C| = |B| > |A|$

k. The relation $R(u, v)$ defined as "u is a prefix of v or v is a prefix of u" is an equivalence relation over any language of strings.
False. It may not be transitive. On $\{a, ab, ac\}$ we have $R(ab, a)$ and $R(a, ac)$, but not $R(ab, ac)$.

l. If in Question 3, we do not require languages to be finite, then the statement at 3a may be false.
True. Take any two different infinite languages over $\Sigma = \{a\}$, for instance, Σ^* and $\Sigma^* \setminus \{\lambda\}$.
For every string in one language we find a longer one in the other language, since it's infinite.

m. Let A be a set of natural numbers. Then the relation $R(x, y)$ defined as "x and y have the same number of divisors" is an equivalence relation but cannot be a partial order.
False. It is a partial order (the equality relation) precisely if A does not have two members with the same number of divisors.

n. If p is a prime, then the relation $R(x, y)$ on naturals, defined as "x and y both have inverses modulo p" is an equivalence relation.
False. $R(x, x)$ is false iff x doesn't have an inverse, so for multiples of p.

o. If m and n are distinct odd numbers with a common divisor greater than 1, then the system $x \equiv 3 \pmod{m}$ and $x \equiv 7 \pmod{n}$ always has no solution.
True. For the common divisor d we'd get $x \equiv 3 \pmod{d}$ and $x \equiv 7 \pmod{d}$, thus $3 \equiv 7 \pmod{d}$, so d divides $7 - 3 = 4$, but d is odd and > 1.