Exam 1 Solutions

October 24, 2000

Problem 1 [30 points] **Predicates and Quantifiers**

Use the following propositions where the universe of discourse is the set of all integers:

$$\begin{aligned} P(x):&\quad x \text{ is a perfect square.}\\ Q(x):&\quad x \text{ is prime.}\\ R(x,y):&\quad x < y.\\ S(x,y):&\quad x \text{ divides } y.\\ T(x,y):&\quad x = y. \end{aligned}$$

Express each of the following statements using quantifiers, logical connectives, and the above propositions, along with variables and integers.

(a) **(5 points)** 4 is a perfect square.

Solution: $P(4)$

(b) **(5 points)** There is an integer which divides all integers.

Solution: $\exists x \forall y S(x,y)$

(c) **(5 points)** Two numbers are equal exactly when neither is less than the other.

Solution: $\forall x \forall y T(x,y) \leftrightarrow \neg R(x,y) \wedge \neg R(y,x)$

(d) **(5 points)** An integer p is prime exactly when the only positive number less than p that divides p is 1.

Solution: $\forall p Q(p) \leftrightarrow [\forall x R(0,x) \wedge R(x,p) \wedge S(x,p) \rightarrow T(x,1)]$

(e) (5 points) An integer r is a perfect square only if there is a positive number between 1 and r that divides r.

Solution: $\forall r P(r) \to [\exists x R(0,x) \land R(1,x) \land R(x,r) \land S(x,r)]$

(f) (5 points) Given ONLY the above statements, can you conclude that 4 is not prime? If not, list a true (but general) fact you need. If so, list which of the above statements are necessary.

Solution: We can conclude that 4 is not prime, using parts (a), (c), (d), and (e). Here is a structured proof:

1. $P(4) \to [\exists x R(0,x) \land R(1,x) \land R(x,4) \land S(x,4)]$ (UI, from part (e).)
2. $\exists x R(0,x) \land R(1,x) \land R(x,4) \land S(x,4)$ (Modus Ponens, from 1 and part (a).)
3. $R(0,x_0) \land R(1,x_0) \land R(x_0,4) \land S(x_0,4)$ (EI, from 2.)
4. $R(1,x_0)$ (Simplification, from 3.)
5. $T(1,x_0) \leftrightarrow \neg R(1,x_0) \land \neg R(x_0,1)$ (Double UI, from part (c).)
6. $\neg R(1,x_0) \land \neg R(x_0,1) \to T(1,x_0)$ (Simplification, from 5, by definition of $\leftrightarrow$.)
7. $\neg[R(1,x_0) \lor R(x_0,1)] \to T(1,x_0)$ (DeMorgan's law, from 6.)
8. $R(1,x_0) \lor R(x_0,1)$ (Addition, from 4.)
9. $\neg T(1,x_0)$ (Modus Tollens, from 7.)
10. $R(0,x_0) \land R(x_0,4) \land S(x_0,4)$ (Simplification, from 3.)
11. $(R(0,x_0) \land R(x_0,4) \land S(x_0,4)) \land \neg T(1,x_0)$ (Conjunction, from 10 and 9.)
12. $\neg[\neg(R(0,x_0) \land R(x_0,4) \land S(x_0,4)) \lor T(1,x_0)]$ (DeMorgan's Law, from 11.)
13. $\neg[(R(0,x_0) \land R(x_0,4) \land S(x_0,4)) \to T(1,x_0)]$ (Logical equivalence, from 12.)
14. $\exists x \neg[(R(0,x) \land R(x,4) \land S(x,4)) \to T(1,x)]$ (EG, from 13.)
15. $\neg \forall x[(R(0,x) \land R(x,4) \land S(x,4)) \to T(1,x)]$ (Negation rule for quantifiers)
16. $Q(4) \leftrightarrow [\forall x R(0,x) \land R(x,4) \land S(x,4) \to T(x,1)]$ (UI, from part (d).)
17. $\neg Q(4)$ (Modus Tollens, from 15 and 16.) QED.

Problem 2 [25 points] **Relations**

For each relation listed in parts (a) through (e) circle all terms listed which apply to it.

(a) **(5 points)** $A = \{a, b\}, R = \{(a, a), (a, b)\}$.

Reflexive	Symmetric
Antisymmetric	Transitive
None of these	

Solution: This relation is antisymmetric and transitive.

(b) **(5 points)** aRb whenever a and b are students and either $a = b$ or a has a GPA greater than that of b.

Partial Order	Total Order
Equivalence Relation	None of These

Solution: This relation is reflexive, transitive and antisymmetric, so it is a partial order. However, it is not a total order since two different students with the same GPA cannot be compared with this relation.

(c) **(5 points)** aRb whenever a and b are real numbers and $a = b^x$ for some rational number x.

Partial Order	Total Order
Equivalence Relation	None of These

Solution: This relation is not symmetric, since $1 = 4^0$ but 1^x is 1 for any rational number x. However, it is not antisymmetric since $2 = 4^{.5}$ and $4 = 2^2$. Thus is is none of these.

(d) **(5 points)** aRb whenever a and b are rational numbers and ab is an integer.

Partial Order	Total Order
Equivalence Relation	None of These

Solution: This relation is not reflexive (for instance, $(1/2) \not R (1/2)$.) Therefore, it is none of these.

(e) **(5 points)** aRb whenever a and b are integers such that there is a positive integer k such that $3^k|a$ and $3^k|b$ and $3^{k+1} \not| a$ and $3^{k+1} \not| b$.

Partial Order	Total Order
Equivalence Relation	None of These

Solution: This relation is essentially just saying that 3 divides a and b the same number of times. This is an equivalence relation, since the same number of times is reflexive, symmetric and transitive.

Problem 3 [15 points] **Directed Graphs**

Recall that a *tournament graph* is a directed graph in which between any two different nodes a and b there is an edge either from a to b or from b to a but not both, and there are no other edges in the graph.

For each of the following, answer true or false and give a brief justification of your answer.

(a) **(5 points)** There is no tournament graph on 4 nodes with an Euler circuit.

Solution: The claim is true.

In a tournament graph on 4 nodes, every node will have an odd number of edges, so the degree of every node is odd. By the theorem from class, it won't have an Euler circuit.

(b) **(5 points)** There is no tournament graph on 5 nodes with an Euler circuit.

Solution: The claim is false.

An example is the graph on three nodes a, b, c, d, e where there are edges are (a, b), (b, c), (c, d), (d, e), (e, a), (a, c), (c, e), (e, b), (b, d), (d, a).

(c) **(5 points)** There is no tournament graph on 225 nodes with an Euler circuit.

Solution: The claim is false.

The complete graph on 225 nodes has an Euler circuit, since every node has degree 224, which is even. Since an Euler circuit contains every edge, all we have to do to construct a 225-node tournament graph with an Euler circuit is take the Euler circuit on K_{225} and choose a direction, then include every edge in our 225-node tournament graph in the direction it appears in the Euler circuit. This will be a tournament graph, since every two nodes are connected and every edge is directed.

Problem 4 [40 points] **Proof Comprehension**

For this proof, fill in reasons for each step denoted with a '*' and say which statements they are based on.

(a) (8 points) Given: $\forall x \exists y(Q(x,y) \vee R(x,y))$,

Prove: $\forall x[\exists y Q(x,y)] \vee [\exists y R(x,y)]$

Proof:

1. Let x_0 be arbitrary.
2. $\exists y(Q(x_0,y) \vee R(x_0,y))$
3. $Q(x_0,y_0) \vee R(x_0,y_0)$

   $(\exists y Q(x_0,y) \vee (\exists y R(x_0,y))$:

   1. case i: Assume $Q(x_0,y_0)$.

      *1.1. $\exists y Q(x_0,y)$

      *1.2. $(\exists y Q(x_0,y) \vee (\exists y R(x_0,y))$

   2. case ii: Assume $R(x_0,y_0)$.

      2.1. $\exists y R(x_0,y)$

      2.2. $(\exists y Q(x_0,y) \vee (\exists y R(x_0,y))$

   *3. Therefore, $(\exists y Q(x_0,y) \vee (\exists y R(x_0,y))$

*4.$\forall x[\exists y Q(x,y)] \vee [\exists y R(x,y)]$

Solution: 4.1: Existential Generalization, from 4. 4.2: Addition, from 4.1. 6: Proof by cases from 3, 4.2, and 5.2. 7: Universal Generalization, from 1 and 6.

For each of the following proofs, either say it is valid, or say it is invalid and what about the proof is incorrect.

(b) (8 points) Theorem: $2 = 1$. Proof:

1. Assume $2 = 1$.
2. $x^2 + 2 = x^2 + 1$.
3. $2xdx = 2xdx$. (differentiating both sides)
4. $2 = 2$ (dividing both sides by xdx)
5. Therefore, $2 = 1$ (QED)

Solution: Step 5 is invalid. After step 4 we have proven that $(2 = 1) \rightarrow (2 = 2)$. This is an implication, and is true even if $2 \neq 1$. So we can't conclude that $2 = 1$.

(c) **(8 points)** Theorem: All rational numbers x greater than or equal to 1 are such that $x \leq 2^x$.

Proof: Let $P(n)$ be the statement "$n \leq 2^n$."

1. We know that $1 \leq 2^1 = 2$. Therefore, $P(1)$.
2. Assume $P(n)$ where $n \geq 1$.
3. Then, $n \leq 2^n$
4. Adding 1 to both sides, we get $n + 1 \leq 2^n + 1$
5. Substituting that $1 \leq n$, we get $n + 1 \leq 2^n + n$.
6. Substituting from step 3 into the RHS of step 5, we get $n + 1 \leq 2^n + 2^n$
7. Simplifying, we get $n + 1 \leq 2^{n+1}$.
8. Thus, $P(n + 1)$. Therefore, by induction, $\forall n P(n)$.

Solution: This proof is invalid. Step 7 is invalid, since $\forall n P(n)$ would be a statement about the rational numbers (since that was the statement of the theorem), but induction allows us to conclude only that $\forall n \in \mathbb{Z}^+ P(n)$.

(d) **(8 points)** Theorem: If G is a finite graph with positive integer-weighted edges that has a spanning tree, it has a minimal spanning tree (a spanning tree T such that the sum of the weights of the edges in T is as small as it could be for any spanning tree).

Proof: Every spanning tree of G has a total weight w which is non-negative integer. Therefore, the set of possible total weights of spanning trees is a subset of the non-negative integers. By the well-ordering property, since there is a spanning tree, there must be a least weight of a spanning tree, and therefore there must be a spanning tree that has that weight. Thus, there is a minimal spanning tree.

Solution: This proof is valid.

(e) **(8 points)** Theorem: There are infinitely many prime numbers. Proof:

Let S be the set of all prime numbers. Now let $k = 1 + \prod_{p \in S} p$. Note that k is not divisible by any p in S since the remainder when dividing by p would be 1. Furthermore, notice that if p is any element of S then $k = 1 + \prod_{q \in S} q \geq 1 + p > p$. But this is a contradiction. Therefore, there must be infinitely many prime numbers.

Solution: The proof is invalid.

This is basically just like the classical proof that there are infinitely many prime numbers with one significant thing missing: the proof never assumes there are finitely many prime numbers. This leads to lots of problems: for one thing, if S is infinite, k is not a number.

Problem 5 [30 points] **Hamiltonian Circuits**

In this problem you will prove that for any positive integer n, any $2n$ by $2n$ grid graph has a Hamiltonian circuit.

An n by m grid graph is made up of vertices described as pairs of integers (x, y) where $0 \le x < n$ and $0 \le y < m$. There is an edge between two vertices if their x-coordinates differ by 1 or if their y-coordinates differ by 1 but NOT if both differ by 1.

So, for example, there would be edges between the vertices $(2, 3)$ and $(3, 3)$ but not between $(2, 3)$ and $(3, 2)$.

(a) **(8 points)** Draw a 4 by 4 grid graph. Label vertices $(1, 3)$, $(2, 2)$, and $(2, 3)$.

(b) **(22 points)** Prove by induction that any $2n$ by $2n$ grid graph has a Hamiltonian circuit for $n \ge 1$. (Other proof methods will not get credit)

Solution: Let $P(n)$ be the statement "Any $2n$ by $2n$ grid graph has a Hamiltonian circuit."

$P(1)$: Any 2 by 2 grid graph is just a square. The square itself is a circuit, and it touches all the nodes. Thus, it is a Hamiltonian circuit.

Assume $P(n)$. Let G be a $2(n + 1)$ by $2(n + 1)$ grid graph. We construct a Hamiltonian circuit for G as follows. First, let the subgraph on all the vertices with $x < 2n$ and $y < 2n$ be called H. H is a $2n$ by $2n$ grid graph, so it has a Hamiltonian circuit.

Start at vertex $(0, 2n+1)$ and continue straight to $(2n+1, 2n+1)$ and then down to $(2n+1, 0)$, then to $(2n, 0)$. Now, since we know there's a Hamiltonian circuit in H, pick one. H must include the edge from $(2n - 1, 0)$ to $(2n - 1, 1)$ since in H, vertex $(2n - 1, 0)$ has degree 2.

Now we continue our path to vertex $(2n-1, 0)$ and through the circuit for H in the direction of $(2n - 2, 0)$. After visiting all the nodes in H, this will return us to $(2n - 1, 1)$. From there we continue to $(2n, 1)$ and up to $(2n, 2n)$ and back to $(0, 2n)$ and we close the circuit by going back to $(0, 2n + 1)$.

Thus, $P(n + 1)$. By induction, if n is an integer ≥ 1 then Any $2n$ by $2n$ grid graph has a Hamiltonian circuit.

Problem 6 [40 points] **State Machines and Invariants**

Consider the following algorithm, that, given a number, produces the prime factorization of that number. The prime factorization of a number is a list of all primes that divide that number, where primes may be listed multiple times if they divide the number multiple times. The product of all the numbers in the list should be equal to the original number.

Assume that Append $(List, i)$ takes a list and adds the element i to the beginning of it. Assume that () refers to the list with no elements.

FactorPrime (n):
1. $i := 2$.
2. $List := ()$.
3. $m := n$.
4. IF $m = 1$ THEN
   FactorPrime(n): $= List$.
5. IF $i|m$ THEN
   Append $(List, i)$.
   $m := m/i$.
   ELSE
   $i := i + 1$.
6. Go to step 4.

(a) **(10 points)** Represent this algorithm as a state machine. In what states will there be no transitions?

Solution:

The states of the machine can be described by

$$Q = \{(i, m, List)|i, m \in \mathbb{Z}\}$$

The starting state on input n is $(2, n, ())$. The transitions are described by:

$$\begin{aligned}\delta(i, m, List) &= \emptyset \text{ if } m = 1,\\ &= (i, m/i, \text{Append}(List, i)) \text{ if } i|m, \text{ or}\\ &= (i + 1, m, List) \text{ otherwise.}\end{aligned}$$

Only the states where $m = 1$ have no transitions.

(b) **(10 points)** Using an invariant, prove that if the algorithm ever gets to the point where $m = 1$, then the list is a factorization of its input n.

Solution: The following is the invariant we will use.

$P(i, m, List)$: "The product of all the numbers in $List$ and the number m equals n."

To prove that this is an invariant, we note that since the start state of the state machine is $(2, n, ())$, the product of all the numbers in $List$ and m is just m, which is n.

Now suppose that $P(i, m, List)$ is true. We wish to prove that $P(\delta(i, m, List))$ is true. We proceed by cases.

Case 1: $i|m$. Then, $\delta(i, m, List) = (i, m/i, Append(List, i))$.

Note that if the product of all numbers in list was k, the product of all numbers in Append$(List, i)$ would be ki. Therefore, since $km = n$ then $(ki)(m/i) = km = n$ also, so $P(\delta(i, m, List))$ is true.

Case 2: $i \not| m$. Then, $\delta(i, m, List) = (i + 1, m, List)$.

Since m and $List$ are unchanged, the invariant continues to hold.

Therefore, $P(i, m, List)$ is an invariant of our state machine. So, if $m = 1$, then the product of all the numbers in $List$ must equal n. Thus, $List$ is a factorization of n.

(c) **(10 points)** Prove that $List$ never contains any numbers that are not prime.

Solution: The following is the invariant we will use.

$R(i, m, List) =$ "$List$ contains no non-prime numbers, and no prime numbers (other than 1) less than i divide m."

In the start state, the list contains no numbers at all so it cannot contain any non-prime numbers. Finally, there are no prime numbers less than 2 other than 1, so $R(2, n, ())$ is true.

Now suppose we know that $R(i, m, List)$ is true. We will prove that $R(\delta(i, m, List))$ is true. There are two cases:

Case 1: $i|m$. Then, $\delta(i, m, List) = (i, m/i, Append(List, i))$.

If no prime numbers less than i divide m, then no prime numbers less than i divide m/i since m/i divides m.

We must also prove that Append($List, i$) contains no prime numbers. The only element in Append($List, i$) that is not in $List$ (which we know contains no prime numbers) is i. Thus, we are asked to prove that i is prime. Suppose i is not prime. Then i has a prime factor smaller than i other than 1: call it h. Since $h|i$ and $i|m$ we know that $h|m$. But, since $R(i, m, List)$ is assumed, there can be no prime numbers smaller than i other than 1 which divide m. Thus, by contradiction, we see that i is prime. Therefore, $R(\delta(i, m, List))$.

Case 2: $i \not| m$. Then, $\delta(i, m, List) = (i + 1, m, List)$.

Since $List$ did not change, it cannot contain any non-prime numbers, since $R(i, m, List)$ is true.

Also, we must verify that no primes less than $i + 1$ other than 1 divide m. The only number less than $i + 1$ which is not less than i is i itself. Since we know that i does not divide m, we see that i certainly cannot be a *prime* less than $i + 1$ which divides m.

Therefore, $R(\delta(i, m, List)$.

Since R is an invariant, it is clear that in particular, $List$ never contains any numbers with are not prime.

(d) **(10 points)** Prove that this algorithm always terminates. You may need to use your invariant from the previous parts.

Solution: The following is a termination function:

$f(q) = n + m - i$.

Note that if we increase i, this decreases $f(\delta q)$ by 1. If we do not increase i, we divide m by i. This decreases $f(\delta q)$ by a factor of $m(i-1)/i$.

By our invariant from part (b), we know that $m|n$, so therefore $m \leq n$. We now prove that $i \leq n$. Suppose not. Then, since $i > m$, by our invariant from part (c), we know that there is no prime less than i that divides m other than 1, so m must be 1. (Otherwise, m would have a prime factor which would be less than or equal to itself, and therefore less than i) But then since $m = 1$ the previous state must have been $(i, i, List)$ for some $List$. Thus, $i|n$ which implies that $i \leq n$.

Thus, $f(q)$ is always positive. It achieves its lowest value when m is as small as possible and i is as large as possible. Since $m \geq 1$ and $i \leq n$, this shows that $f(q) \geq 1$.

By the termination theorem, the algorithm terminates.

Problem 7 [20 points]

Consider the following description of a state machine A.

Q: The set of tuples consisting of:

  M, a multiset of integers

  request, either *insert*(x) or *get*

  response, either *null* or *OK* or x or *error*

Q_0: $M = \emptyset$, *request* $=$*null*,*response* $=$ *null*

L:

  Input transitions: $Insert(x)$, *Get*

  Internal transitions: *Perform-insert*, *Perform-get*

  Output transitions: $Respond(r)$

δ:

  Insert(x):

    Can occur if: *request* $=$ *null*

    Effect: *request* $:=$ $insert(x)$

  Get:

    Can occur if: *request* $=$ *null*

    Effect: *request* $:=$ *get*

  Perform-insert:

    Can occur if: *request* $=$ $insert(x)$

    Effect: add x to the multiset M, *response* $:=$*OK*

  Perform-get:

    Can occur if: *request* $=$ *get*

    Effect: if M is not empty

        remove any element x nondeterministically from M

        response $:= x$

      else

        response $:=$ *error*

  Respond(r):

    Can occur if: *response* $= r, r \neq$ *null*

    Effect: *response* $:=$ *null*, *request* $:=$ *null*

Now consider another machine B which differs from A only in its state and the *Perform* transitions.

Q: The set of tuples consisting of:
  S, a sequence of integers
  request, either *insert*(x) or *get*
  response, either *null* or *OK* or x or *error*
δ:
  Perform-insert:
    Can occur if: $request = insert(x)$
    Effect: $S := xS$ where xS denotes the concatenation of x and S
      $response := OK$
  Perform-get:
      Can occur if: $request = get$
      Effect: if M is not empty
    Can occur if: $request = get$
    Effect: if $S = xS'$
      $S := S'$
        $response := x$
      else
        $response := error$

Answer the following questions based on the above descriptions of A and B. If your answer is yes, give an abstraction relation. If no, give a sample trace to justify your answer.

(a) (10 points)

Does A implement B ?

Solution:

No. Consider the following trace of A's execution.

$Insert(1), Respond(OK), Insert(2), Respond(OK), Get, Respond(1)$.

There's no possible execution of B that can match this trace.

(b) (10 points)

Does B implement A ?

Solution:

Yes. The following is an abstraction relation from B to A.

$R = \{(q_a, q_b) | q_a.request = q_b.request, q_a.response = q_b.response, equal(q_a.M, q_b.S)\}$

where M snd S are *equal* if they have the same number of copies of each distinct umber.

This problem shows why a stack (B) can implement a bag (A) but not vice-versa.