b) There are two counterexamples: \( x = \sqrt{2} \) and \( x = -\sqrt{2} \).
c) There is one counterexample: \( x = 0 \).

38. a) Some system is open.  
   b) Every system is either malfunctioning or in a diagnostic state. 
   c) Some system is open, or some system is in a diagnostic state.  
   d) Some system is unavailable. 
   e) No system is working. (We could also say “Every system is not working,” as long as we understood that this is different from “Not every system is working.”)

40. There are many ways to write these, depending on what we use for predicates.
   
a) Let \( F(x) \) be “There is less than \( x \) megabytes free on the hard disk,” with the domain of discourse being positive numbers, and let \( W(x) \) be “User \( x \) is sent a warning message.” Then we have \( F(30) \rightarrow \forall x W(x) \).
   
b) Let \( O(x) \) be “Directory \( x \) can be opened,” let \( C(x) \) be “File \( x \) can be closed,” and let \( E \) be the proposition “System errors have been detected.” Then we have \( E \rightarrow ((\forall x \neg O(x)) \land (\forall x \neg C(x))) \).
   
c) Let \( B \) be the proposition “The file system can be backed up,” and let \( L(x) \) be “User \( x \) is currently logged on.” Then we have \( (\exists x L(x)) \rightarrow \neg B \).
   
d) Let \( D(x) \) be “Product \( x \) can be delivered,” and let \( M(x) \) be “There are at least \( x \) megabytes of memory available” and \( S(x) \) be “The connection speed is at least \( x \) kilobits per second,” where the domain of discourse for the last two propositional functions are positive numbers. Then we have \( (M(8) \land S(56)) \rightarrow D(\text{video on demand}) \).

42. There are many ways to write these, depending on what we use for predicates.
   
a) Let \( A(x) \) be “User \( x \) has access to an electronic mailbox.” Then we have \( \forall x A(x) \).
   
b) Let \( A(x,y) \) be “Group member \( x \) can access resource \( y \),” and let \( S(x,y) \) be “System \( x \) is in state \( y \).” Then we have \( S(\text{file system}, \text{locked}) \rightarrow \forall x A(x, \text{system mailbox}) \).
   
c) Let \( S(x,y) \) be “System \( x \) is in state \( y \).” Recalling that “only if” indicates a necessary condition, we have \( S(\text{firewall, diagnostic}) \rightarrow S(\text{proxy server, diagnostic}) \).
   
d) Let \( T(x) \) be “The throughput is at least \( x \) kbps,” where the domain of discourse is positive numbers, let \( M(x,y) \) be “Resource \( x \) is in mode \( y \),” and let \( S(x,y) \) be “Router \( x \) is in state \( y \).” Then we have \( (T(100) \land \neg T(500) \land \neg M(\text{proxy server, diagnostic})) \rightarrow \exists x S(x, \text{normal}) \).

44. We want propositional functions \( P \) and \( Q \) that are sometimes, but not always, true (so that the second biconditional is \( P \iff F \) and hence true), but such that there is an \( x \) making one true and the other false. For example, we can take \( P(x) \) to mean that \( x \) is an even number (a multiple of 2) and \( Q(x) \) to mean that \( x \) is a multiple of 3. Then an example like \( x = 4 \) or \( x = 9 \) shows that \( \forall x (P(x) \iff Q(x)) \) is false.

46. a) There are two cases. If \( A \) is true, then \( (\forall x P(x)) \lor \exists x P(x) \lor \forall A(x) \) is also true. Thus both sides of the logical equivalence are true (hence equivalent). Now suppose that \( A \) is false. If \( P(x) \) is true for all \( x \), then the left-hand side is true. Furthermore, the right-hand side is also true (since \( P(x) \lor A(x) \) is true for all \( x \)). On the other hand, if \( P(x) \) is false for some \( x \), then both sides are false. Therefore again the two sides are logically equivalent.

   b) There are two cases. If \( A \) is true, then \( (\exists x P(x)) \lor \forall A(x) \) is true, and since \( P(x) \lor A \) is true for some (really all) \( x \), \( \exists x (P(x) \lor A(x)) \) is also true. Thus both sides of the logical equivalence are true (hence equivalent). Now suppose that \( A \) is false. If \( P(x) \) is true for at least one \( x \), then the left-hand side is true. Furthermore, the right-hand side is also true (since \( P(x) \lor A(x) \) is true for that \( x \)). On the other hand, if \( P(x) \) is false for all \( x \), then both sides are false. Therefore again the two sides are logically equivalent.

48. a) There are two cases. If \( A \) is false, then both sides of the equivalence are true, because a conditional statement with a false hypothesis is true. If \( A \) is true, then \( A 
\rightarrow P(x) \) is equivalent to \( P(x) \) for each \( x \), so the left-hand side is equivalent to \( \forall x P(x) \), which is equivalent to the right-hand side.
b) There are two cases. If \( A \) is false, then both sides of the equivalence are true, because a conditional statement with a false hypothesis is true (and we are assuming that the domain is nonempty). If \( A \) is true, then \( A \to P(x) \) is equivalent to \( P(x) \) for each \( x \), so the left-hand side is equivalent to \( \exists x P(x) \), which is equivalent to the right-hand side.

50. It is enough to find a counterexample. It is intuitively clear that the first proposition is asserting much more than the second. It is saying that one of the two predicates, \( P \) or \( Q \), is universally true; whereas the second proposition is simply saying that for every \( x \) either \( P(x) \) or \( Q(x) \) holds, but which it is may well depend on \( x \). As a simple counterexample, let \( P(x) \) be the statement that \( x \) is odd, and let \( Q(x) \) be the statement that \( x \) is even. Let the domain of discourse be the positive integers. The second proposition is true, since every positive integer is either odd or even. But the first proposition is false, since it is neither the case that all positive integers are odd nor the case that all of them are even.

52. a) This is false, since there are many values of \( x \) that make \( x > 1 \) true.
b) This is false, since there are two values of \( x \) that make \( x^2 = 1 \) true.
c) This is true, since by algebra we see that the unique solution to the equation is \( x = 3 \).
d) This is only false, since there are no values of \( x \) that make \( x = x + 1 \) true.

54. There are only three cases in which \( \exists x P(x) \) is true, so we form the disjunction of these three cases. The answer is thus \( (P(1) \land \neg P(2) \land \neg P(3)) \lor ( \neg P(1) \land P(2) \land \neg P(3)) \lor ( \neg P(1) \land \neg P(2) \land P(3)) \).

56. A Prolog query returns a yes/no answer if there are no variables in the query, and it returns the values that make the query true if there are.
a) None of the facts was that Kevin was enrolled in EE 222. So the response is no.
b) One of the facts was that Kiko was enrolled in Math 273. So the response is yes.
c) Prolog returns the names of the courses for which Grossman is the instructor, namely just cs301.
d) Prolog returns the names of the instructor for CS 301, namely grossman.
e) Prolog returns the names of the instructors teaching any course that Kevin is enrolled in, namely chan, since Chan is the instructor in Math 273, the only course Kevin is enrolled in.

58. Following the idea and syntax of Example 28, we have the following rule:

\[
\text{grandfather}(X, Y) :- \text{father}(X, Z), \text{father}(Z, Y); \text{father}(X, Z), \text{mother}(Z, Y).
\]

Note that we used the comma to mean “and” and the semicolon to mean “or.” For \( X \) to be the grandfather of \( Y \), \( X \) must be either \( Y \)’s father’s father or \( Y \)’s mother’s father.

60. a) \( \forall x (P(x) \to Q(x)) \)  b) \( \exists x (R(x) \land \neg Q(x)) \)  c) \( \exists x (R(x) \land \neg P(x)) \)
d) Yes. The unsatisfactory excuse guaranteed by part (b) cannot be a clear explanation by part (a).

62. a) \( \forall x (P(x) \to \neg S(x)) \)  b) \( \forall x (R(x) \to S(x)) \)  c) \( \forall x (Q(x) \to P(x)) \)  d) \( \forall x (Q(x) \to \neg R(x)) \)
e) Yes. If \( x \) is one of my poultry, then he is a duck (by part (c)), hence not willing to waltz (part (a)). Since officers are always willing to waltz (part (b)), \( x \) is not an officer.