Today, I'm going to show you a C++ script to help you visualize the logic of quantified statements.

We will start with an existential statement nested inside of a universal statement. We can see here the Set of X and Set of Y. Our quantified statement is: "For all n in X; there is an m in Y; where n times m is >= 15". To validate this statement, we iterate through each value of n and check the validity for each value of m. If at any point we find a valid value of m for a value of n, we exit the inner loop and continue to the next value of n. If no valid value of m is found for a value of n, we exit the outer loop and return that the statement is invalid. In this example, we find a valid value of m for each value of n, so we say this statement is valid.

Next, we will look at a universal statement nested within an existential statement. The Set of X and the Set of Y have remained the same. Our quantified statement says: “For some n in X; and for all m in Y; n divided by m is an integer.” To validate this statement, we iterate through each value of n and check the validity for each value of m. If at any point we find an invalid value of m for a value of n, we exit the inner loop and continue to the next value of n. If there is no value of n that is valid for all values of m, we exit the outer loop and return that the statement is invalid. In this example, we find a value of n that is valid for all values of m, so we say this statement is valid.

The next example is a universal statement nested within a universal statement. The Set of X and the Set of Y have remained the same. Our quantified statement says: “For all n in X; and for all m in Y; n times m is greater than or equal to 2”. To validate this statement, we must iterate through each value of n and check the validity for each value of m. If at any point we find an invalid value of m for a value of n, we exit the outer loop and return that the statement is invalid. In this example, we find that for each value of n, it is valid for all values of m, so we say this statement is valid.

Our final example is an existential statement nested within an existential statement. The Set of X and the Set of Y remained the same. Our quantified statement says: “For some n in X; there is some m in Y; where n times m is >= 10.” To validate this statement, we must iterate through each value of n and check the validity for each value of m. If at any point we find a valid value of m for a value of n, we exit the outer loop and return that the statement is valid. In this example, we find that for a value of n, there is a valid value of m, so we say this statement is valid.

If you would like to view the source code, the public repository can be found by navigating to this page. I encourage you to test your own quantified statements against the validation logic of this program. Thanks for watching.