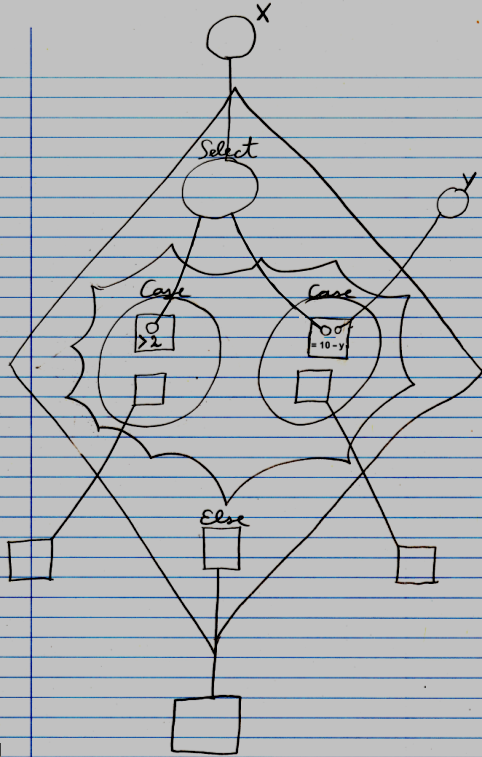
|  |
| --- |
| Circle Language Spec: Execution Control |

## Select Case (split formula) in a Diagram

The concept of the Select Case (split formula) statement is already covered by the article *Select Case (split formula)*. This article only explains its expression in a diagram.

There are two forms of this statement: the first half of the formula is a value or the first half of the formula is an operation with an operand missing. These two forms have a different definition and look different in the diagram

Below is an example of the diagrammatic expression of a Select Case statement for split formulas, of which the first half of the formula is a value.

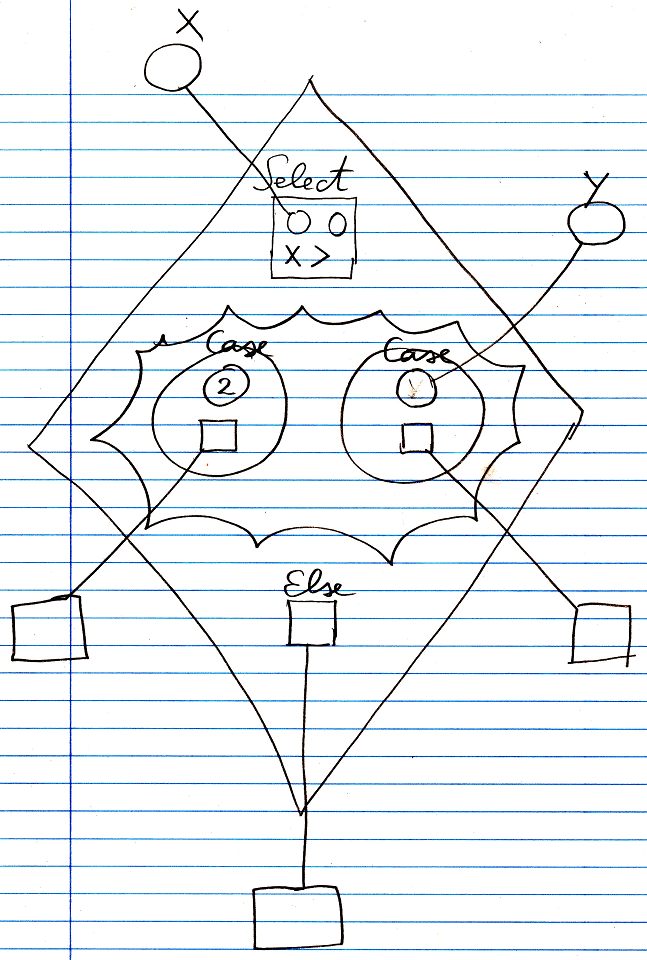


The diamond is a call to the Select Case command. The circle inside the diamond has the title Select. It points to an object outside the command call. This is called the variable. It is the value, that will be filled into the missing operands of the other halves of the formula. In the middle of the diamond there is a nonagon, that represents the cases: different operations into which the variable will be filled in. The nonagon can contain any number of cases. Each circle inside a nonagon is a Case. Each case defines a command for which the variable is filled in. Each case also defines the command to call when the result of the formula is True. In this example, there are two Cases. The other halves of the formula are not drawn out in full detail. That would obscure the picture in this demostration. The literals of the half formulas are shown. The command definitions of the half formulas are not pointed out, and the build-up of the formula’s is not fully graphically drawn out with objects connected with operations, because that would obscure the picture of this demonstration, but they do belong in the diagram, though.

One Case is the half formula > 2 . X is to be filled in as the first operand of this formula, which would make the resultant formula X > 2. The second Case is the half formula = 10 – Y. X is to be filled in as the first half of the formula, which would make the resultant formula X = 10 – Y.

Each Case has a command associated to it. Those command references are pointing to clauses defined outside the diamond. When a resultant formula of the case returns True, then the command associated with the case will be executed. At the bottom of the diamond there is also a command reference called Else. It points to a clause defined outside the diamond. The command pointed to will be called if *none* of the formulas results in the Boolean value True. If the Else clause is not used, it can be left out of the call and then it will not be shown in the diagram.

Below is an example of the diagrammatic expression of a Select Case statement for split formulas, of which the first half of the formula is half a formula, and the second halves of the formula are values to be filled in as the missing operand in the first half of the formula.

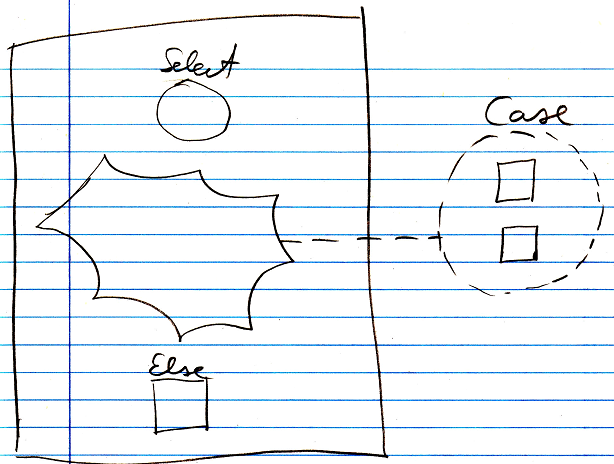


The diamond is a call to the Select Case command. The square at the top of the diamond is the first half of the formula. It is a *greater than* operation. The first operand of the formula is already filled in with the object X, by having the parameter point out of the diamond to the object called X. The second parameter of the operation is yet to be filled in by the Cases of the Select Case statement. In the middle of the diamond there is a nonagon, that represents the cases: different values to be filled in as the missing operand of the formula. The nonagon can contain any number of cases. Each circle inside a nonagon is a Case. Each case defines a value to be filled in as the missing operand of the formula. Each case also defines the command to call when the result of the formula is True. In this example, there are two Cases. One Case is the value 2. This value will be filled in as the missing operand of the formula. This makes the resultant formula X > 2. The second Case is not a fixed value, but points to the variable Y, which is defined outside the diamond. Y is filled in as the missing operand of the formula. This makes the resultant formula X > Y.

Each Case has a command associated to it. Those command references are pointing to clauses defined outside the diamond. Every command for which the resultant formula returns True is executed. At the bottom of the diamond there is also a command reference called Else. It points to a clause defined outside the diamond. The command pointed to will be called if none of the formulas results in the Boolean value True. If the Else clause is not used, it can be left out of the call and then it will not be shown in the diagram.

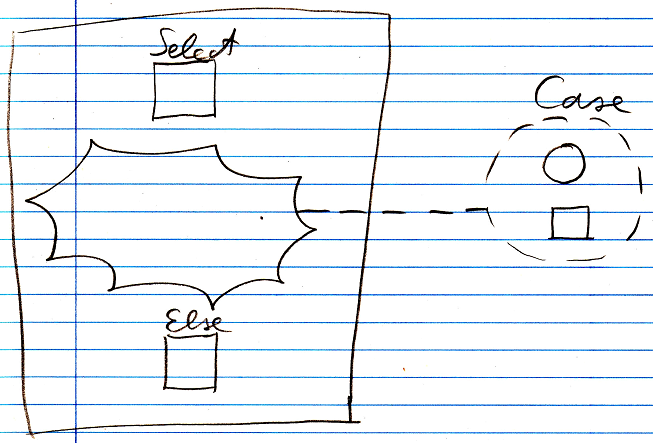
In the examples above, each value, that was litterly filled in, could also have been a pointer to something remote. Conversely, everything that was a pointer to something outside the diamond, could also have been defined directly inside the diamond.

The definition of the Select Case execution control commands is part of a system module of execution control commands. The public elements of the definition for a value as the first part of the formula looks like this:



Nothing is filled in yet as the Select or Else, and there are no Cases defined, but a *class* for a Case *is* defined. There is also a definition without an Else clause in it.

The public elements of the definition for half a formula as the first part of the formula looks like this:



Nothing is filled in yet as the Select or Else, and there are no Cases defined, but a *class* for a Case *is* defined.

There is another, separate definition of the Select Case command for formulas, that is the same as the other Select Case command definition, except that it does not have an Else clause in it.