The original code, given two sets containing some values, tore apart the sets into individual element values and then checked which all elements were common to both the sets. These common elements were then reassembled into a set called intersection. However, this code did not work as the set intersection, after being reassembled, was again getting disassembled as in the rule for disassemble the highlighted part below chose a set with any name to disassemble:

(defrule disassemble

(declare (salience 1))

?set <- (set ?name ?value $?elements)

=>

(retract ?set)

(assert (set ?name $?elements))

(assert (element ?name ?value)) )

The code is now changed as below which considers a set with any name except one called intersection. This code works properly(as can be seen in the dribble file):

(defrule disassemble

(declare (salience 1))

?set <- (set ?name&~intersection ?value $?elements)

=>

(retract ?set)

(assert (set ?name $?elements))

(assert (element ?name ?value)) )

If we remove salience from the given code, then also it works fine(can be seen in the dribble file) as it does not matter here if we reassemble after finding each common element or reassemble after finding all common elements between the two sets etc. Thus, salience is not required here.

The above corrected code, both with and without salience, has the elements 1 and 2 in the intersection set.