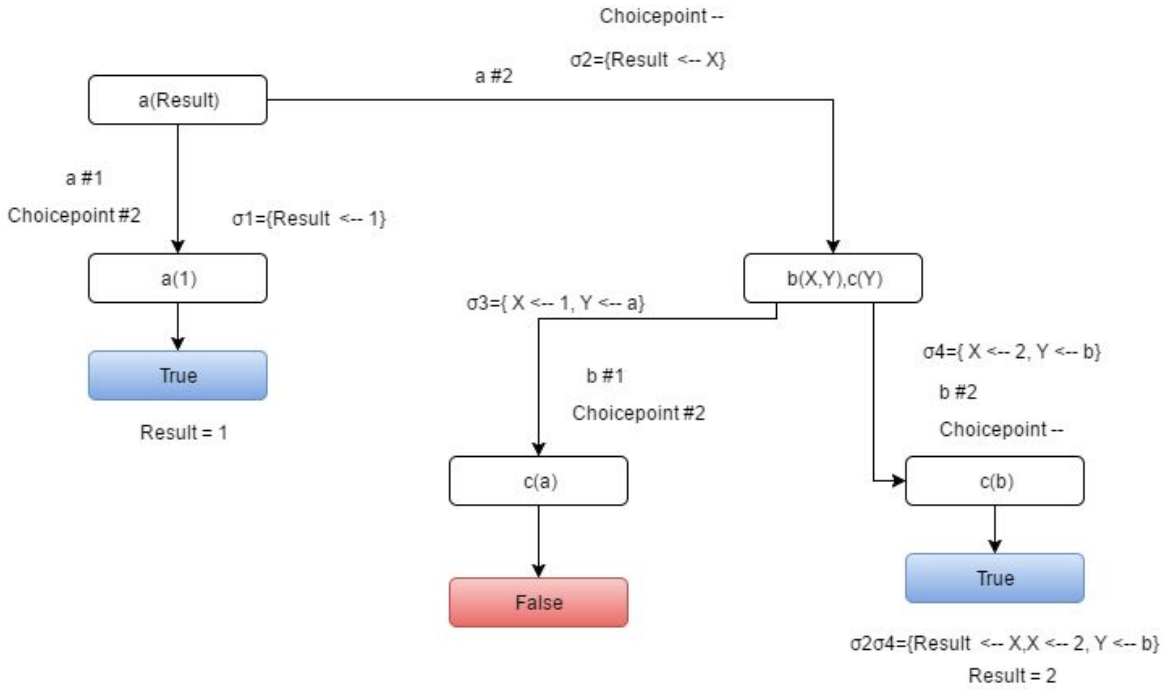
a) ?- a(Result).

a(1).
a(X) :- b(X,Y), c(Y).

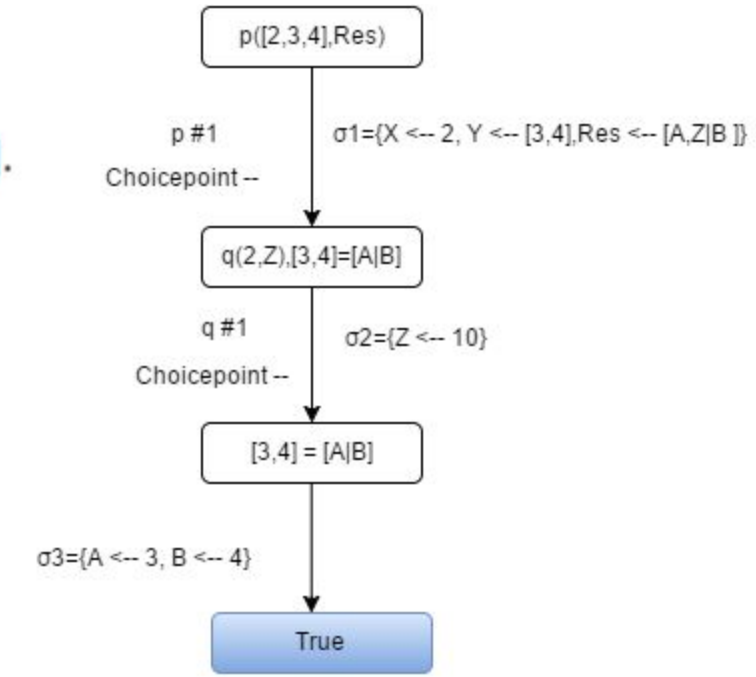
b(1, a). c(b).
b(2, b). c(c).



b) ?- p([2, 3, 4], Res).

p([X|Y], [A, Z|B]) :- q(X, Z), Y = [A|B].

q(2, 10).



σ2σ3={Z <-- 10, A <--3, B <-- 4}

σ1σ2σ3={X <--2,Y <--[3,4],Res <-- [3,10|4], Z <-- 10, A <--3, B <-- 4}

Res = [3,10,4]

Task 3:

The rules r/1 and s/1 are not the same in most cases, but are the same iff they are used like this:
% **r(+X)** this means that a value is passed to the X and thus the rule work as logical operator
% also the same for **s(+X)**
% and return true or false and the same result is done by the double negation rule
r(X):-p(a,X).
s(X) :- not(not(p(a,X))).
If not the two rules will behave differently because r/1 will then return the result from the knowledge base (the facts) and s/1 will always return true or false because it returns the result of the not predicate which is a logical predicate.
Conclusion: the predicates are not alike because the main reason of the r/1 predicate is to get all the possible cases of X defined by p/2, while the not predicate only produces true or false if the unification of negation succeeds.

PS. the test runs are in the respective file.