1 Is Propagation Compositional?

The two solution sets are identical which is shown by the program. Does this mean propagation is compositional?

Is there any other aspect in which they compute differently? In which aspect?

Propagation is not compositional, propagations communicate through the variables values and assignments.

Yes they result in different search trees despite the fact that they are using the same search-heuristics (branching strategies):

First we have script $S1$ which models $A + B + C + B = X$ where $D_A = D_B = D_C = D_X = \{1, 2, 3, 4, 5\}$

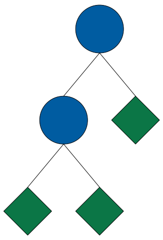

Figure 1: First-fail Minimum-value for Script S1

Second we have script $S2$ which models $A + B = U; U + B + C = X$ where $D_A = D_B = D_C = D_X = D_U = \{1, 2, 3, 4, 5\}$

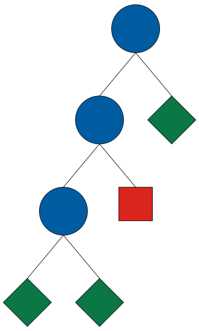

Figure 2: First-fail Minimum-value for Script S2

What is happening?

In S1 we only post one constraint that will create propagators for that single constraint while in S2 there are two constraints posted. In the search tree of S2 the propagator for $A + B = U$ is executed first which results in interleaving of constraint propagation and variable assignment until the constraint is satisfied, in particular the assignments $A = 1, B = 2, U = 3$ is tried which leads to a (red) failed node since that means that there are no satisfying assignments for C and X that gives $U + B + C = X$. This is because $U + B = 5$ and the maximum value of X is 5 and the minimum value of C is 1.

In S1 already before any assignment is made constraint propagation can infer that B must be 1. This is the main difference. In S2 the constraint propagation were not able to resolve this initially since just looking at the constraint $U + B + C = X$ B could be other things than 1. But if the constraint propagation were able to consider both constraint simultaneously it would find out that B could not be something else than 1, looking at each constraint in isolation B does not have to be only 1.