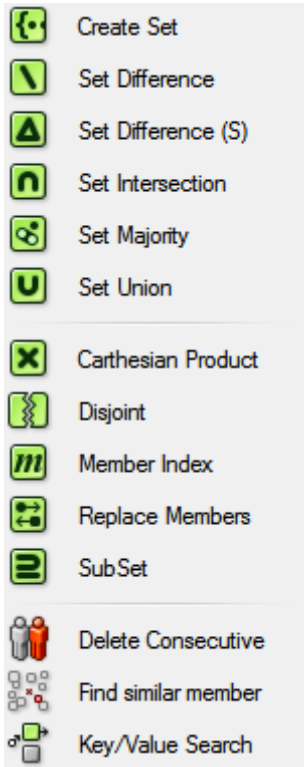
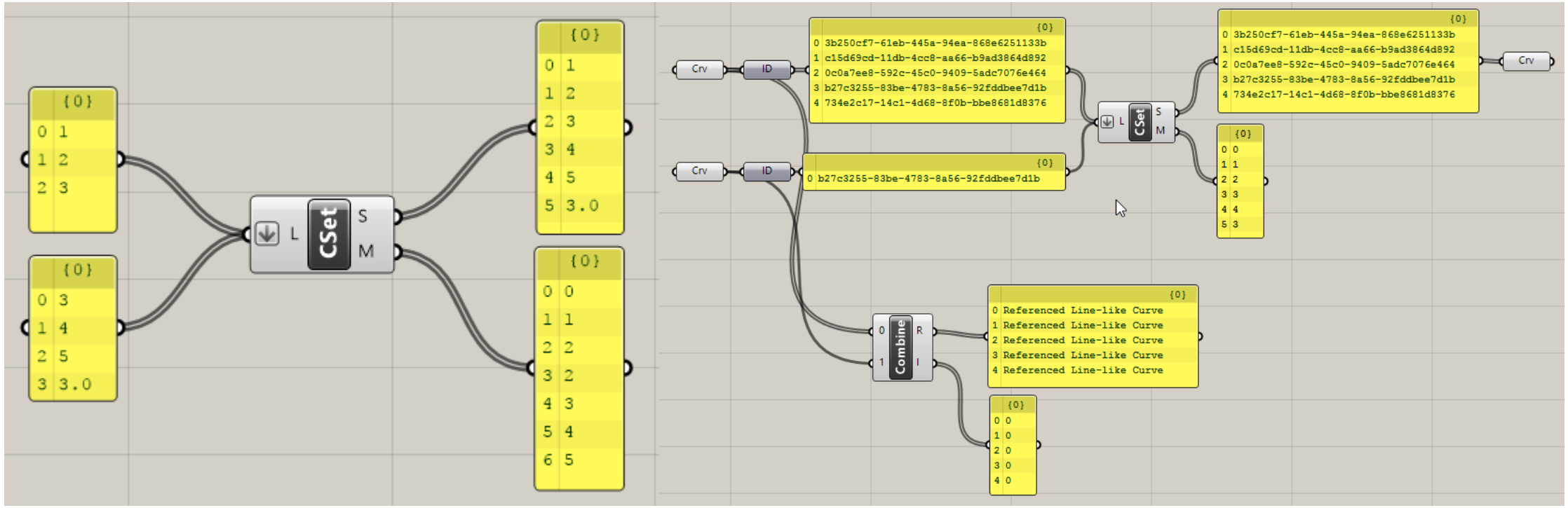
Working with grasshopper sets


Create Set
Set Difference
Set Difference (S)
Set Intersection
Set Majority
Set Union
Carthesian Product
Disjoint
Member Index
Replace Members
SubSet
Delete Consecutive
Find similar member
Key/Value Search

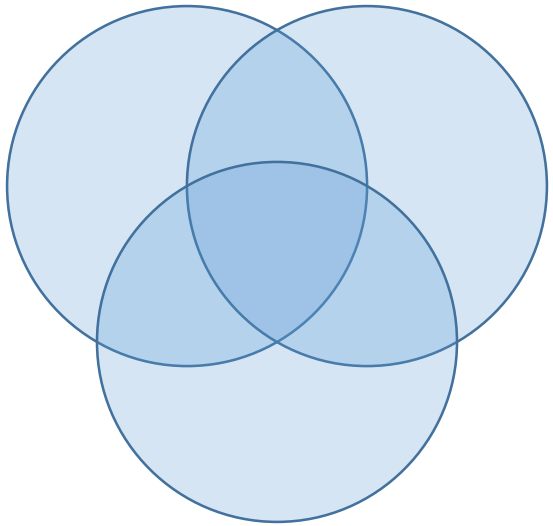
Creating Sets, using them to find unique objects

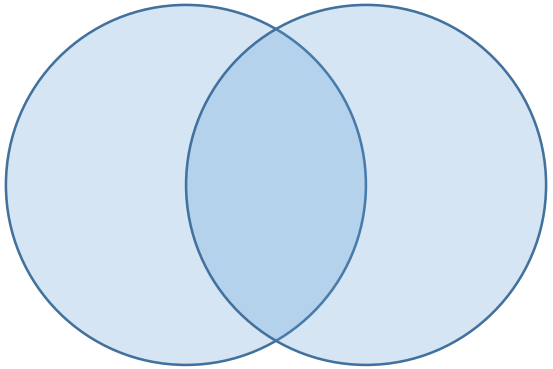
Sets are much like lists, except they contain only unique objects (no repeats where the test of repetition is equality, n.b. equality is by both type and value 3(integer)!=3.0(float)!=3.0(double))

Therefore sets are more limited in the types of data they can carry, only those that can be subjected to a test for equality (no curves, breps etc). However, these objects have a Geometry ID Number in Grasshopper, and these strings can be used in a set if this is helpful. Input the Geometry IDs back into a geometry container afterwards and they reconstituted to the referenced objects rather than the Geometry ID strings.



Basic Set Operations

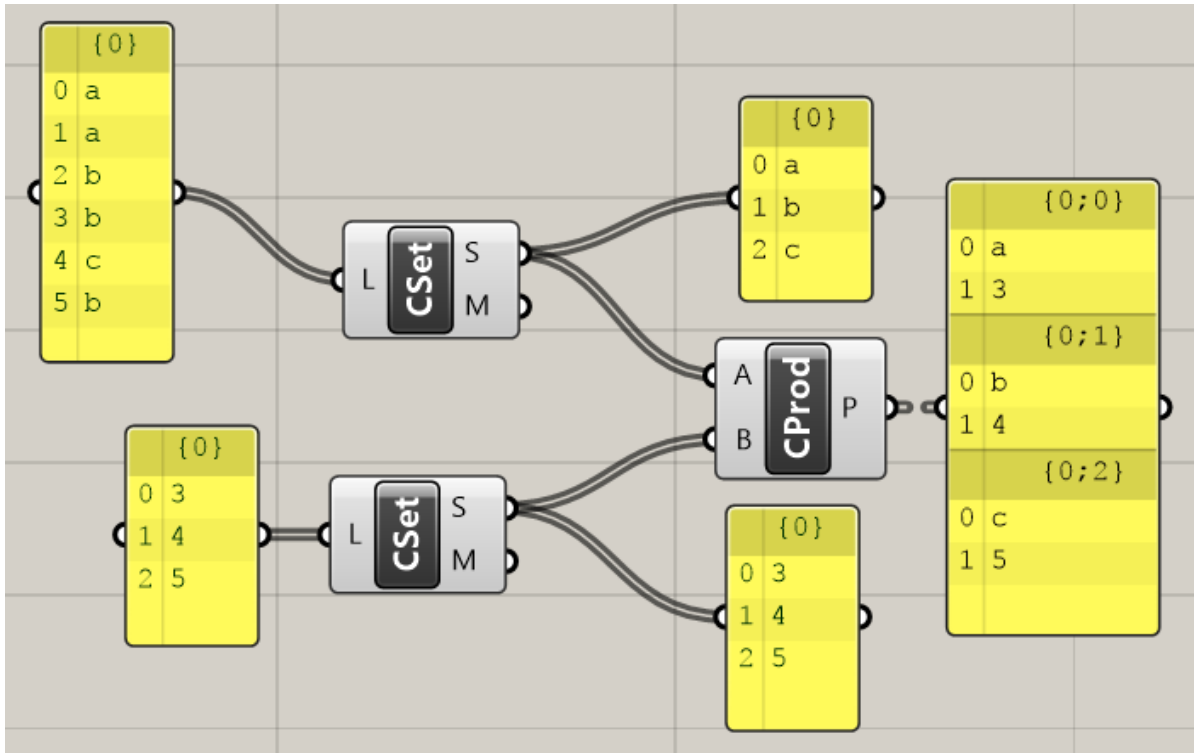




Set difference	Not in Grasshopper	A–B
Set difference (symmetrical)	Not in Grasshopper	A∪B–A∩B
Set intersection	(A∩B)∪(A∩C)∪(B∩C)	A∩B
Set majority	A∪B∪C	N/A
Set union	A∪B∪C	A∪B
Disjoint	Not in Grasshopper	A∩B=∅?
Subset	N/A	B⊆A

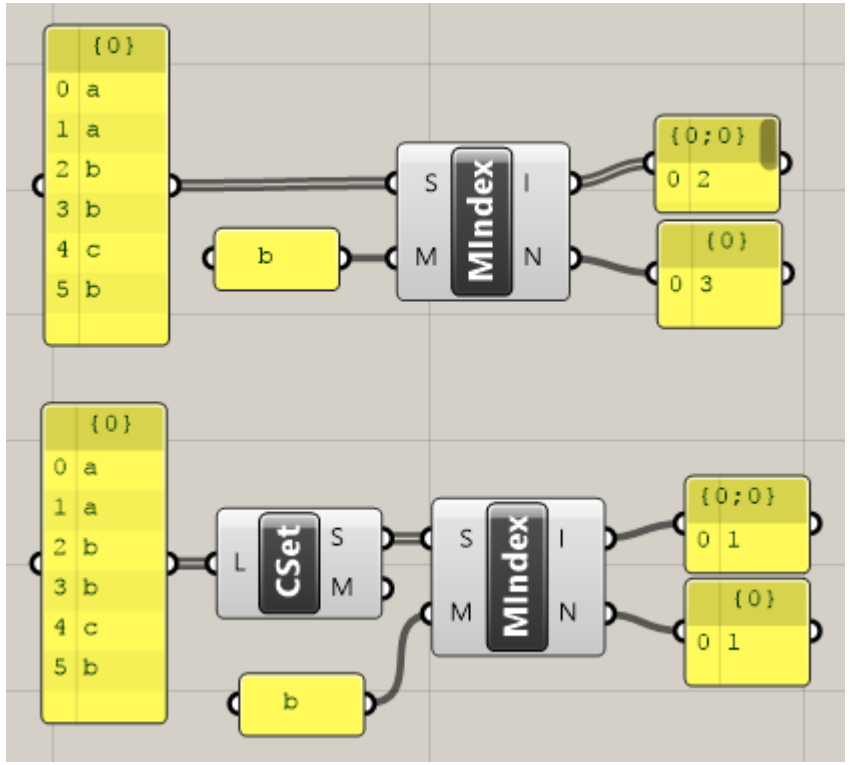
Cartesian Product of Sets (so-called)

Actually just acts as a zip of lists, giving out a series of lists.
A real cartesian product component would look more like the holistic cross reference component for lists.

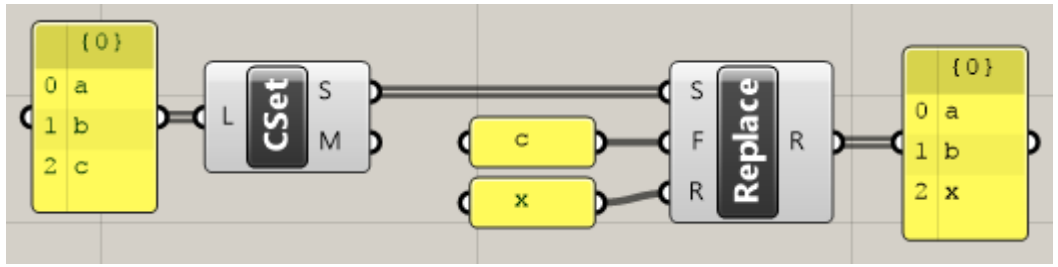


Member index

Returns the index number, and the number of occurrences (in a set function, really?) of an object in a list/set.
Shown once on a list, and once on the set generated from this list.

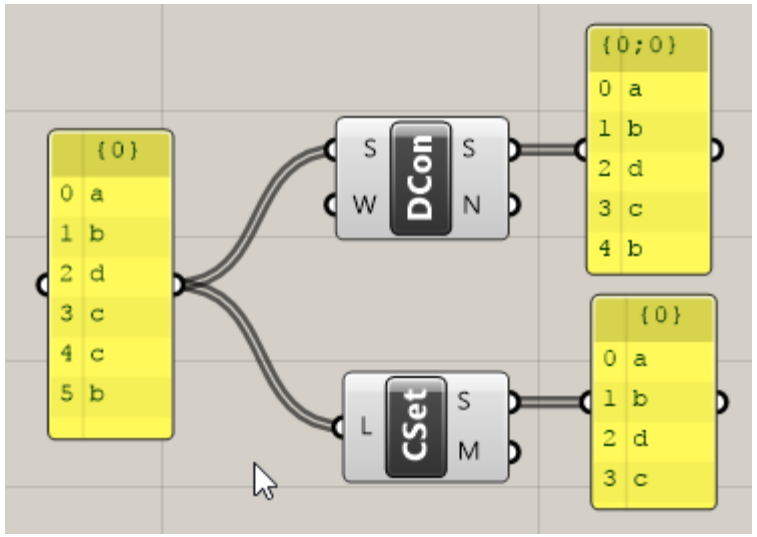


Replace member



Delete consecutive

Removes list elements that are equal and next to one another in a list.



Find similar member

Matches numbers to the nearest match in a list
N.B. Also matches text, so make sure your ‘numbers’ aren’t actually text. Here the “1.5” matches 3 of the 4 characters in “10.5”, which is why the behaviour can seem bizarre at times.