

# Initial Value Computations

## 1 Graph representation of maps

Input variables of each map can be represented in unions and or intersections of relations's variables. We can model these dependencies with a directed graph. We construct a graph  $G = \langle V, E \rangle$  as follow.  $V$  is the union of all variables(including the bigsum variables) and the set of all maps. The set of edges consist of all directed edges of form  $(a, b)$  which shows that there is an directed edge between node  $a$  and  $b$ . Here  $a$  can be a variable and/or a map name, but  $b$  definitely is a map name because any variable is an independent entity which does not depend on any map or other variables. When there is  $(a, b)$  edge it means that the computation of map  $b$  needs the value of variable/map of  $a$ .

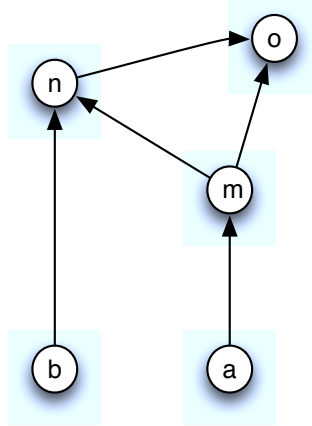


Figure 1: Sample dependencies,  $a, b$  are variables and  $m, n, o$  are map names

Obviously this graph does not have any directed cycle. For each variable node we can have a list of pertaining relations in which that variable has occurred in the schema. With this kind of data structure we know when to alter the domain of each maps. When an new item is added to a relation we know which variables depend on the relation and we can change the domain of each map in a breadth first search(BFS).

When we add/delete any value from the domain of any variable we need to propagate this modification through the dependency graph. We do this task by a layer by layer manner. Suppose we have altered variable  $a$ . First we change the domains of its immediate adjacent. If their domains changed we do this for the next layer and so on. When none of the adjacent of a node changed, we are not required to check the modifications to the variables' domain, since it won't change(i.e. in figure 1 if none of  $m_1, m_2$  changes we don't need to check the modification of  $m_3$ ).