IVC

July 25, 2011

1 Definitions

In this section we explain the domain which is represented here by dom. The $\mathsf{dom}$of a variable is a set of values that the variable can take. The $\mathsf{dom}$of a map is defined by the all values of its input variables(which is a vector).

$$\mathsf{dom}_i = \left\{ x \in map_i[\vec{x}][\vec{y}] | (x \in \vec{x}) \vee (x \in \vec{y}) \right\}$$

We need to compute the dependences between the domains of maps input variables. Without loose of generality we always need to work with two maps when we want to compute their doms. For this we define function F recursively as follow:

$$F = \mathsf{dom} | F \cap F | F \cup F$$

So we can compute the dependencies between the doms like this:

$$\mathsf{dom}(x \in m_i[\vec{x}][\vec{y}]) \leftarrow F(\mathsf{dom}(x' \leftarrow m_j[\vec{x'}][\vec{y'}]))$$

Where j is all the maps which occurs on the left hand of phrase for computing m_i as we have in **??** since the information can only be passed from left to right.

We call the arity of a domas the number of its input variables. We need to maintain the maps as long as their arities are greater that zero. If the arity of any domain drops to zero it means that we don't need to store it as a map since it doesn't contain any elements. The arity of a domain can be increased of decreased. It may be increased by adding more elements into some relation, also it may be decreased by deletions from some relations too.

We need to store the arities inside each map. We need a way to compute the arity of an AGCA expression. Since we substitute the subexpressions with maps, we can easily consider the maps without input variables as some relations. Also a map with input variables can be seen as a relation with a group-by clause. Input variable of a map bind some variables. Thus if we compute the map values by all different combinations of these variables, we will look up into these values and return the appropriate value according to the input variables.

2 Computing the arities of the AGCA expression

We can compute the arities recursively. Suppose we have an expression which is consist of two parts joining together as $e = e_1 \, op \, e_2$ which $op \in \{+, *\}$. Let I_i and O_1 to be the set of all input and output variables to e_i respectively. The arity of e is $|I_1 \cup I_2 - O_1|$, according to the information flow rule.