

# 1 Domain evolution

Consider the following query:

$$q[] = R(a) \cdot S(a) \cdot (0 < ((T()) \leq a))$$

When computing the  $\Delta$  for this query, regarding to relation  $R$ , the result will be:

$$\Delta_R q(a') = S(a') \cdot (0 < ((T()) \leq a'))$$

$S(a')$  and  $(0 < (T()) \leq a)$  will become maps that will help maintain the delta query:  $S(a')$  will be replaced by map  $m_S[] [a]$  and  $((T()) < a)$  by map  $m[a][]$ :

$$\Delta_R \alpha(a') = m_S[] [a'] \cdot (0 < m[a'][])$$

We are wondering how the domain of variable  $a$  will change when an update will appear in relation  $S$ , taking into account that map  $m_S$  has the variable  $a$  as an output variable and the map  $m$  has the variable  $a$  as an input variable.

There will always be a data flow from a map to another, from the output variables of a map to the input variables of another map. Going back to the example, we can say that  $domain(a_m) \cup = domain(a_{m_S})$ , where  $a_m$  is an input variable in the map  $m$  and  $a_{m_S}$  is an output variable in the map  $m_S$ . The  $domain(a_m)$  will not increase if the following relation between the two domains will be satisfied:  $domain(a_{m_S}) \setminus domain(a_m) = \emptyset$ , otherwise the value that was added will be in the domain of  $S$ , however it will not be contained in the domain of the map for relation  $((T()) < a)$ , therefore an update of the maps should be performed.

Starting from the delta  $\Delta_R q(a') = m_S[] [a'] \cdot (0 < m[a'][])$  and the relation between input and output variables we can write the following pseudo-code for computing the domain of variable  $a$ :

$$[a \leftarrow domain(map_{a_{as\ output\ variable}})]; domain(map_{a_{as\ input\ variable}}) \cup = \{a\}$$

For the example presented, we are going to have:

$$\begin{aligned} & [m_S[] [x] : \{x \leftarrow dom(m_S[] [x])\}] \\ & [m[x][] : \{dom(m[x][]) \leftarrow x\}] \\ & [m_S[] [x] \cdot (0 < m[x][]) : \{x \leftarrow dom(m_S[] [x]); dom(m[x][]) \leftarrow x; \\ & \quad dom(m[x][]) \leftarrow dom(m_S[] [x])\}] \end{aligned}$$

However, considering all the expressions from AGCA, how will the domain of variables will change?

We will take all the expressions and try to come up with some rules which will help in the computation of the domain of each variable present in the definition of a map:  $q ::= q \cdot q \mid q + q \mid q\theta t \mid t <- q \mid var \mid const$ .

???What value does an extension of the map receive???

We will start by trying to give a definition to the notion of  $dom(map)$  and the operator  $\leftarrow$ . We will begin with an example:

$$[map[][\vec{x}] : \{x_i \leftarrow dom(x_i \in map[][\vec{x}]) \mid x_i \in \vec{x}\}]$$

This means that the  $map[]$  will be bound to the rule that for every  $x_i$ , an output variable, the domain for the variable will be given by the values offered by the relation. If the map has input variables then the domain of an input variable will be given by an outside relation. For example:

$$[map[\vec{x}][] : \{dom(x_i \in map[\vec{x}][]) \leftarrow x_i \mid x_i \in \vec{x}\}]]$$

Therefore  $dom(x_i \in map[][])$  represents the domain of variable  $x_i$ , whether the variable is an output variable or an input variable. For  $map[][\vec{x}]$ , where  $\vec{x} = [x_1, x_2, \dots, x_n]$  we will have  $x_1 \leftarrow dom(x_1 \in map[][\vec{x}])$ ,  $x_2 \leftarrow dom(x_2 \in map[][\vec{x}])$  and so on. The relation stand also for input variables, however the  $\leftarrow$  relation is inverted:  $dom(x_1 \in map[\vec{x}][]) \leftarrow x_1$ .

For  $q ::= q \cdot q$  we will have:

$$\frac{[q1 \cdot q2 : \{z \leftarrow x \mid (z \leftarrow y) \in \vec{a} \wedge (y \leftarrow x) \in \vec{b}\} \cup \vec{a} \cup \vec{b}]}{[q1 : \vec{a}] \wedge [q2 : \vec{b}]}$$

For  $q ::= q + q$  we will have the following:

$$\frac{[q1 + q2 : \{x \leftarrow s \mid property_s\}]}{[q1 : \vec{a}] \wedge [q2 : \vec{b}]}$$

where  $property_s$  is:

$$\begin{aligned} &((x \leftarrow s1) \in \vec{a} \wedge (x \leftarrow s2) \in \vec{b} \wedge (s = s1 \cup s2)) \vee \\ &((x \leftarrow s1) \in \vec{a} \wedge (x \leftarrow s2) \notin \vec{b} \wedge (s = s1)) \vee \\ &((x \leftarrow s1) \notin \vec{a} \wedge (x \leftarrow s2) \in \vec{b} \wedge (s = s2)) \end{aligned}$$