Delta Logic : Internal Report III

Adithya Murali ; adithya5@illinois.edu

January 27, 2018

1 Separability Theorem

In this section, we show a key result: that for any quantifier-free FO+*lfp* formula, there exists an equisatisfiable delta-logic formula. This result substantiates and its elements illustrate our primary thesis: that delta-logic is the correct logic in which to write VCs.

First, let us consider a recursively defined predicate/function R, with the restricted in form as above **??**. Let the set of pointer fields be interpreted by unary functions in $P = \{p_i \mid 1 \leq i \leq k\}$ for some k. We argue that one can rewite this definition so as to be able to handle the changes that the program makes on a finite portion of the heap (which we interpret as the finite set of variables Δ) not by having entirely new recursively defined predicates/functions, but by writing the formula in a static logic that accounts for the change by defining the recursive predicates/-functions in context-logic that stays oblivious of the exact model on Δ, which when combined with the changes to the finite portion can be written into a delta-logic formula.

Observe, that on any given model of locations related by pointers in a set of pointer fields interpreted by P, the valuation on this *lfp* definition can be viewed as an iterative process that assigns the least element of the lattice to every location at the beginning, and then, iteratively imbibes the value of at a location from its decendendants on P. We relativise this process with respect to Δ by introducing a set of booleans and first-order variables, one for each element in Δ which parameterises the new definition on this set, as in:
If R is defined as $R :=_{lfP} \phi$, we define a set of variables $B_R^\Delta = \{b_R^d \mid d \in \Delta\} \subseteq range(R)$.
Then, we define the new recursively defined function/precdicate corresponding to R, namely $R^{B_R^\Delta}$ as follows:

$$\begin{aligned} R^{B_R^\Delta}(x) :=_{lfp}\ & b_R^d, x = d \text{ for some } d \in \Delta \\ & \phi[R^{B_R^\Delta}/R], x \notin \Delta \end{aligned}$$

It is easy to see that for a formula $R(x)$, writing it as $R^{B_R^\Delta}(x)$ would be a formula in context-logic since an model of it would not depend on a valuation over Δ.

However, the above is obviously not equivalent to R. To capture the semantics of the original *lfp* definition, we must ensure that the values of variables in Δ must be constrained to match the values that one would have obtained on a valuation on the original definition. This will yield definitions that are equivalent in FO+*lfp* under such constraints.

We do this by writing constraints that simulate the 'imbibing' of values from descendants that we described earlier. However, in doing so, we run into the problem of cycles. Simple implication constraints that produce a value on a location from values of its descendants will not work when, for example, we have a circular list. If the constraint were that the location pointed to being the head of a list sufficed for the location itself to be the head of a list, then one could easily provide a valuation on every element on the cycle as beingthe head of a/ pointing to a list. This, of course, is not true.
We handle this by introducing the notion of the 'rank' of a location w.r.t R, which intuitively captures an order among elements where elements that were provided with their correct valuation on a later iteration (in the iteration view of *lfp* valuation) would be higher in the order. In particular, for the example of a circular list before, if we recursively defined a 'rank' at a location as being 0 at the location *nil*, and being one higher than the rank of the location pointed to for every node that was the head of a list, there would be no way to provide a valuation of every element on the cycle as being the head of a list. However, since the rank will need to communicate through the elements outside Δ to maintain this order, it will also be a similarly relativised *lfp* definition with parameters for values on locations interpreted by variables in Δ. We choose to model the rank as a function to $\mathbb{N} \cup \{ bot \}$ ($\perp$ signifies undefined rank), as follows for a recursively defined predicate R:

$$
\begin{aligned}
Rank_R^{\Delta}(x) :=_{lfp}\ & \rho_R^d\,, x = d \text{ for some } d \in \Delta \\
& max\{Rank_R^{\Delta}(p_i(x)) \mid 1 \leq i \leq k\} + 1\,, x \notin \Delta \wedge R^{B_R^{\Delta}}(x) \\
& \perp\,, x \notin \Delta \wedge \neg R^{B_R^{\Delta}}(x)
\end{aligned}
$$

;such that $max(S) = -1$ for $S \subseteq \mathbb{N} \cup \{\perp\} \iff \perp \in S$
,and where $\rho_R^{\Delta} = \{\rho_R^d \mid d \in \Delta\} \subseteq \mathbb{N} \cup \{\perp\}$ is a set of parameters to provide the value of rank on variables in Δ.

Finally, we define:
$B_R^{p_i(\Delta)} = \{b_R^{p_i(d)} \mid d \in \Delta\} \subseteq range(R)$
and $\rho_R^{p_i(\Delta)} = \{\rho_R^{p_i(d)} \mid d \in \Delta\} \subseteq \mathbb{N} \cup \{\perp\}$
for every $1 \leq i \leq k$.

We then denote the substitution $\gamma[B/R]$ as replacing the term $R(p_i(x))$ with $b_R^{p_i(x)}$ for every $1 \leq i \leq k$, and $\gamma[\{\perp\}/R]$ as replacing with $\perp$. We shall also denote $\phi[d/x]$ as $\phi(d)$. With the above, we write the following constraint β_R^{Δ} for a recursively defined predicate R:

$$\bigwedge_{d\in\Delta}\Big[\Big(\phi(d)[\{\bot\}/R] \implies b_R^d \wedge \Big(\rho_R^d = 0\Big)\Big)$$
$$\wedge\Big((\phi(d)[B/R]\wedge\neg\phi(d)[\{\bot\}/R]) \implies \Big(b_R^d\wedge\Big(\rho_R^d = max(\{\rho_R^{p_i(d)}, 1\le i\le k\})+1\Big)\Big)\Big)$$
$$\wedge\Big((\neg\phi(d)[B/R]\wedge\neg\phi(d)[\{\bot\}/R]) \implies \Big(\neg b_R^d\wedge\Big(\rho_R^d=\bot\Big)\Big)\Big)$$
$$\bigwedge_{1\le i\le k}[\Big(p_i(d)\notin\Delta \implies b_R^{p_i(d)} \iff R^{B_R^\Delta}(p_i(d))\Big)$$
$$\wedge\Big(p_i(d)\notin\Delta \implies \rho_R^{p_i(d)} = Rank_R^\Delta(p_i(d))\Big)]\Big]$$

For a recursively defined function, we shall do the same, except for a few minor differences. The definition of rank and β_R^Δ for a recursively defined function $G :=_{lfp} \phi$ with the same restrictions of form are as follows:

$$Rank_G^\Delta(x) :=_{lfp} \rho_G^d, x = d \text{ for some } d\in\Delta$$
$$max\{Rank_G^\Delta(p_i(x))\,|\,1\le i\le k\}+1, x\notin\Delta\wedge G^{B_G^\Delta}(x)\neq\bot$$
$$\bot, x\notin\Delta\wedge G^{B_G^\Delta}(x)=\bot$$

,and similarly β_G^Δ :

$$\bigwedge_{d\in\Delta}\Big[\Big(\phi(d)[\{\bot\}/G]\neq\bot \implies b_G^d = \phi(d)[\{\bot\}/G]\wedge\Big(\rho_G^d=0\Big)\Big)$$
$$\wedge\Big((\phi(d)[B/G]\neq\bot\wedge\phi(d)[\{\bot\}/G]=\bot) \implies \Big(b_G^d=\phi(d)[B/G]$$
$$\wedge\Big(\rho_G^d = max(\{\rho_G^{p_i(d)}, 1\le i\le k\})+1\Big)\Big)\Big)$$
$$\wedge\Big((\phi(d)[B/G]=\bot\wedge\phi(d)[\{\bot\}/G]=\bot) \implies \Big(b_G^d=\bot\wedge\Big(\rho_G^d=\bot\Big)\Big)\Big)$$
$$\bigwedge_{1\le i\le k}[\Big(p_i(d)\notin\Delta \implies b_G^{p_i(d)} = G^{B_G^\Delta}(p_i(d))\Big)$$
$$\wedge\Big(p_i(d)\notin\Delta \implies \rho_G^{p_i(d)} = Rank_G^\Delta(p_i(d))\Big)]\Big]$$

We are now ready to state our main theorem. Let us call the total set of variables introduced $B_R^\Delta\cup\rho_R^\Delta\cup\bigcup_{1\le i\le k}\Big(B_R^{p_i(\Delta)}\cup\rho_R^{p_i(\Delta)}\Big)$ as the set of 'parameters' P_R^Δ.Then,

Theorem 1. *Fory any recursively defined function/predicate R,* $\big(\exists P_R^\Delta.\,\beta_R^\Delta\big)\wedge\Big(\forall P_R^\Delta.\,\Big(\beta_R^\Delta \implies R^{B_R^\Delta}=R\Big)\Big)$

We shall delay a proof of this until the end of this paper. Assuming this is true, we have, for any FO+*lfp* formula α, an equisatisfiable delta-logic formula. Let the set of recursive functions/predicates mentioned in α be $\mathscr{R}$, and Δ be fixed. Let $\mathscr{R}^\Delta = \{R^{B_R^\Delta}\,|\,R\in\mathscr{R}\}$. Then:

Corollary 1 (Separability)**.** *α is satisfiable iff $\alpha[\mathscr{R}^\Delta/\mathscr{R}] \wedge \left(\bigwedge_{R \in \mathscr{R}} \beta_R^\Delta \right)$ is satisfiable.*

Proof. To see why this is true, consider that α is satisfiable. From theorem 1, for every $R \in \mathscr{R}$, we can pick P_R^Δ such that β_R^Δ and, therefore, $R^{B_R^\Delta} = R$. Thus, we have that $\alpha[\mathscr{R}^\Delta/\mathscr{R}] \wedge \left(\bigwedge_{R \in \mathscr{R}} \beta_R^\Delta \right)$ is satisfiable.

Conversely, let $\alpha[\mathscr{R}^\Delta/\mathscr{R}] \wedge \left(\bigwedge_{R \in \mathscr{R}} \beta_R^\Delta \right)$ be satisfiable. Again, from theorem 1 we have that for every $R \in \mathscr{R}$, the valuation given by the model for P_R^Δ satisfies β_R^Δ, and therefore $R^{B_R^\Delta} = R$. Therefore, $\alpha[\mathscr{R}^\Delta/\mathscr{R}][\mathscr{R}/\mathscr{R}^\Delta] = \alpha$ is satisfiable. □