Delta Logics: Logics for Change

ACM Reference Format:
. 2018. Delta Logics: Logics for Change. 1, 1 (April 2018), 2 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 AN EXPLICIT FORMULATION

We describe an explicit formulation that offers more insight into the working of delta-logics. Recall that the main idea is to separate the heap into two parts: one that changes and one that does not and, in correspondence, express the VC as a boolean combination of two kinds of formulae describing each one of those parts. This is a formulation in a static logic with one (copy of) parameterised recursive function. We then use frame reasoning on the unchanging context to relate the two forms of the parameterised recursive function, SMT reasoning on the bounded Δ to relate the two sets of parameters and solve the mutual constraints to check VC validity. In the general formulation the prescence of the two different sets of parameters does not necessarily make clear this intuition. However in some cases, and in particular the case of lists and measures described in Section **??**, the recursive function R satisfies the following condition for all sets of parameters P:

$$\forall x\, \exists y, y \in \Delta.\, R(x, P) = R(x, \bot_R) \oplus_R P(R, y)$$

where $P(R, x)$ is a slight abuse of notation referring to the parameter variable in P 'corresponding' to x for R, $\bot_R$ is a constant of the type of $P(R, x)$, and $\oplus_R$ is some binary operator of appropriate signature such that the above equation is valid. In fact, we can make it a little more general and make $\bot_R$ a function of x as well. The $\oplus_R$ operator may also require more than just the corresponding parameter, and may make use of the parameters corresponding to $Rank_R$ as well.

In this case, observe that we can explicitly apply frame reasoning to conclude that the valuation of the term $R(x, \bot_R)$ does not change by pointing out that the underlying heaplet of its definition is exclusively outside Δ, which does not change. This is true for any x (from our formulation of the definiton R above). The changed parameter $P'(R, x)$ can then be recombined through the $\oplus_R$ operator with this term (since the equation is valid for all sets of parameters) to yield a valuation for $R(x, P')$.

Example: Consider the example given in Section **??***:*

$$\{ls(x, nil) \wedge ls(w, nil) \wedge y \notin hls(w, nil)\}\ \texttt{y.next := w}\ \{ls(x, nil)\}$$

Even in the simple subcase where $w \notin hls(x, nil)$, we have to argue the validity of the postcondition in both cases, namely $y \in hls(x, nil)$ and $y \notin hls(x, nil)$. As we saw earlier, vanilla frame reasoning will not work for the former case. In the delta-logic formulation, let the set of parameters for the pre- and post- states be P, P' respectively (the individual variables are thought to be primed versions as well). The relativised definition of **ls** *with* **nil** *as the terminal point given in Section* **??**

Author's address:

Manuscript submitted to ACM

satsifies the following equation for any h:
$ls^{P}_{nil}(h) \iff ls^{true}_{nil}(h) \wedge LS^{f(h)}_{nil}$
where $\bot_{ls} = true$, $\oplus_{ls} \equiv \wedge$ *and we have applied the same abuse of notation in our description to the LS parameters and* $f(x)$ *is a Skolem function. Observe that in this case, where* y *and* w *are* Δ*, it is possible to easily conclude that* LS^{y}_{nil} *holds, and since for none of the locations* $d \in \Delta$ *the parameter change* LS^{d}_{nil} *to* $LS'_{nil}{}^{d}$ *is from* $true$ *to* $false$ *(in fact for* y *it may have been* $false$ *to* $true$*), and that* $ls^{P}_{nil}(x)$ *held at the beginning, we have* $ls^{P'}_{nil}$ *holds on the post-state, which is what we wanted.*

Observe here that the application of simple frame reasoning to our formulation yields a more fine-grained frame rule in that we can still conclude the truth of recursive predicates if the parameters they depend on become only 'more' true.

This condition is true of some of our list measures as well: with the $\bot_R$ and $\oplus_R$ being respectively: empty set and set-union for *hls*, similarly empty multiset and multiset-union for *mskeys*, and 0 and standard addition for *len*. This formulation closely illustrates the motivation and working of delta-logics: it is about enabling simpler reasoning by the use of close and fine-grained frame reasoning while being able to handle complex conditions on bounded worlds.