

1 **Delta Logics: Logics for Change**

2

3 ANONYMOUS AUTHOR(S)

4

5 **ACM Reference Format:**

6 Anonymous Author(s). 2018. Delta Logics: Logics for Change. 1, 1 (April 2018), 2 pages. [https://doi.org/10.](https://doi.org/10.1145/nnnnnnnn.nnnnnnn)
7 [1145/nnnnnnnn.nnnnnnn](https://doi.org/10.1145/nnnnnnnn.nnnnnnn)



42 Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee
43 provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and
44 the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored.
45 Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires
46 prior specific permission and/or a fee. Request permissions from permissions@acm.org.

47 © 2018 Association for Computing Machinery.

48 XXXX-XXXX/2018/4-ART \$15.00

49 <https://doi.org/10.1145/nnnnnnnn.nnnnnnn>

APPENDIX

We give here the detailed definitions of the other measures that were omitted in Section 4.

- We have recursive definitions that capture the multiset of data elements (through a data-field *key*) stored in list segments, where again the multiset of data of list-segments from an element v in Δ to z (where $z \in \Delta \cup \{\text{nil}\}$) is imbibed from the set variable $MSKeys_z^v$:

$$\begin{aligned}
 mskeys_z^P(x) &:=_{lfp} && \emptyset \text{ if } \llbracket x \rrbracket = \llbracket z \rrbracket \\
 &&& \{key(x)\} \cup_m mskeys_z^P(n(x)) \text{ if } \llbracket x \rrbracket \neq \llbracket z \rrbracket \wedge \llbracket x \rrbracket \neq \llbracket \text{nil} \rrbracket \wedge \llbracket x \rrbracket \notin \llbracket \Delta \rrbracket \\
 &&& MSKeys_z^x \text{ if } \llbracket x \rrbracket \neq \llbracket z \rrbracket \wedge \llbracket x \rrbracket = \llbracket v \rrbracket \wedge v \in \Delta
 \end{aligned}$$

- We have recursive definitions that capture the maximum/minimum element of data elements stored in list segments, where again the maximum/minimum element of list-segments from an element v in Δ to z where $z \in \Delta \cup \{\text{nil}\}$ is imbibed from the data variable Max_z^v (or Min_z^v). We assume the data-domain has a linear-order \leq , and that there are special constants $-\infty$ and $+\infty$ that are the minimum and maximum elements of this order. Let $max(r_1, r_2) \equiv ite(r_1 \leq r_2, r_2, r_1)$.

$$\begin{aligned}
 Max_z^P(x) &:=_{lfp} && -\infty \text{ if } \llbracket x \rrbracket = \llbracket z \rrbracket \\
 &&& max(key(x), Max_z^P(n(x))) \text{ if } \llbracket x \rrbracket \neq \llbracket z \rrbracket \wedge \llbracket x \rrbracket \neq \llbracket \text{nil} \rrbracket \wedge \llbracket x \rrbracket \notin \llbracket \Delta \rrbracket \\
 &&& MSKeys_z^v \text{ if } \llbracket x \rrbracket \neq \llbracket z \rrbracket \wedge \llbracket x \rrbracket = \llbracket v \rrbracket \wedge v \in \Delta
 \end{aligned}$$

The function Min_z^P is similarly defined.

- We have a recursive definition that captures sortedness, using the minimum measure.

$$\begin{aligned}
 Sorted_z^P(x) &:=_{lfp} \Big(x = z \vee \\
 &\quad \left(x \neq z \wedge x \neq \text{nil} \wedge x \notin \Delta \wedge min_z^P(x) \neq \perp \wedge key(x) \leq min_z^P(x) \wedge Sorted_z^P(n(x)) \right) \vee \\
 &\quad \left(x \neq z \wedge x \in \Delta \wedge min_z^P(x) \neq \perp \wedge key(x) \leq min_z^P(x) \wedge \bigwedge_{v \in \Delta} (x = v \Rightarrow SORTED_z^v) \right) \Big)
 \end{aligned}$$