

# Views on Edits to Variational Software – Appendix

ANONYMOUS AUTHOR(S)

## ACM Reference Format:

Anonymous Author(s). 2018. Views on Edits to Variational Software – Appendix. In . ACM, New York, NY, USA, 2 pages. <https://doi.org/XXXXXXXX.XXXXXXXX>

## 1 PROOF OF CORRECTNESS OF $view_{smart}$ (THEOREM 5.8)

LEMMA 1.1. For a variation  $diff D = diff(T_b, T_a) = (V, E, r, \tau, l, \Delta)$  that is the difference of two variation trees  $T_i = (V_i, E_i, r_i, \tau_i, l_i)$ ,  $i \in \{b, a\}$ , the root, typing, and labels of  $D$  and  $T_t$  are equal for all times  $t \in \{b, a\}$ , i.e.,  $r = r_t$  and  $\forall v \in V_t : l(v) = l_t(v)$  and  $\tau(v) = \tau_t(v)$ .

PROOF OF LEMMA 1.1. Let  $t \in \{b, a\}$ . As of  $diff(T_b, T_a) = (V, E, r, \tau, l, \Delta)$ , we also have  $project(diff(T_b, T_a), t) = project((V, E, r, \tau, l, \Delta), t)$ . The left side simplifies by Definition 5.2 to  $T_t$ . The right side, by definition of  $project$  (Equation 1), simplifies to  $(V', E', r, \tau, l)$  with  $V' = \{v \in V_t \mid t \in \Delta(v)\}$  and  $E' = \{e \in E_t \mid e \in \Delta(e)\}$ . Thus,  $T_t = (V_t, E_t, r_t, \tau_t, l_t) = (V', E', r, \tau, l)$  from which we conclude that  $r = r_t$  and  $\forall v \in V_t : l(v) = l_t(v)$  and  $\tau(v) = \tau_t(v)$ .  $\square$

PROOF OF CORRECTNESS OF  $VIEW_{SMART}$  (THEOREM 5.8). Let  $T_i = (V_i, E_i, r_i, \tau_i, l_i)$ ,  $i \in \{b, a\}$  be two variation trees and  $\rho$  a relevance. Let  $D = diff(T_b, T_a) = (V_D, E_D, r_D, \tau_D, l_D, \Delta_D)$ . Our goal is to prove

$$view_{smart}(D, \rho) \equiv diff(view_{tree}(T_b, \rho), view_{tree}(T_a, \rho))$$

which by definition of semantic equivalence  $\equiv$  (Definition 5.3) means that for all times  $t \in \{b, a\}$ , the following has to hold:

$$project(view_{smart}(D, \rho), t) = project(diff(view_{tree}(T_b, \rho), view_{tree}(T_a, \rho)), t)$$

The proof works by case analysis on the time  $t$  and showing that both sides of the equation simplify to the same term.

### Case $t = b$ :

By definition of  $project$  (Equation 1), the left side of our goal simplifies to

$$project(view_{smart}(D, \rho), b) = (\{v \in V_{smart} \mid b \in \Delta_{smart}(v)\}, \{e \in E_{smart} \mid b \in \Delta_{smart}(e)\}, r_D, \tau_D, l_D)$$

with  $(V_{smart}, E_{smart}, r_D, \tau_D, l_D, \Delta_{smart}) = view_{smart}(D, \rho)$ . The right side of our goal simplifies to

$$\begin{aligned} & project(diff(view_{tree}(T_b, \rho), view_{tree}(T_a, \rho)), b) && \left. \begin{array}{l} \text{apply Def. 5.2 by substituting} \\ T_b \text{ with } view_{tree}(T_b, \rho) \\ \text{Eq. 4} \\ \text{Lem. 1.1} \end{array} \right\} \\ & = view_{tree}(T_b, \rho) \\ & = (V_{view_{tree}}, E_{view_{tree}}, r_b, \tau_b, l_b) \\ & = (V_{view_{tree}}, E_{view_{tree}}, r_D, \tau_D, l_D) \end{aligned}$$

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

© 2018 Association for Computing Machinery.

Manuscript submitted to ACM

where  $V_{viewtree} = viewnodes(T_b, \rho)$  and  $E_{viewtree} = E_b \cap (V_{viewtree} \times V_{viewtree})$ . After simplifying both sides, thus our goal is to prove

$$(\{v \in V_{smart} \mid \mathbf{b} \in \Delta_{smart}(v)\}, \{e \in E_{smart} \mid \mathbf{b} \in \Delta_{smart}(e)\}, r_D, \tau_D, l_D) = (V_{viewtree}, E_{viewtree}, r_D, \tau_D, l_D).$$

To show that both simplified sides are equal, it thus remains to prove that

$$(I) \quad \{v \in V_{smart} \mid \mathbf{b} \in \Delta_{smart}(v)\} = V_{viewtree} \text{ and}$$

$$(II) \quad \{e \in E_{smart} \mid \mathbf{b} \in \Delta_{smart}(e)\} = E_{viewtree}.$$

(I)

$$\begin{aligned} & \{v \in V_{smart} \mid \mathbf{b} \in \Delta_{smart}(v)\} \\ &= \{v \in V_{smart} \mid \mathbf{b} \in tor(D, v, \rho)\} && \left. \begin{array}{l} \text{Eq. 10} \\ \text{def } V_{smart} \text{ via Eq. 10} \end{array} \right\} \\ &= \{v \in V_D \mid tor(D, v, \rho) \neq \emptyset, \mathbf{b} \in tor(D, v, \rho)\} && \left. \begin{array}{l} \text{simplify} \\ \text{Eq. 9} \end{array} \right\} \\ &= \{v \in V_D \mid \mathbf{b} \in \Delta_D(v), v \in viewnodes(project(D, \mathbf{b}), \rho)\} && \left. \begin{array}{l} \text{def } D \\ \text{Def. 5.2} \end{array} \right\} \\ &= \{v \in V_D \mid \mathbf{b} \in \Delta_D(v), v \in viewnodes(T_b, \rho)\} && \left. \begin{array}{l} \text{simplify} \\ \text{def } V_{viewtree} \end{array} \right\} \\ &= \{v \in V_b \mid v \in viewnodes(T_b, \rho)\} \\ &= viewnodes(T_b, \rho) \\ &= V_{viewtree} \end{aligned}$$

(II)

$$\begin{aligned} & \{e \in E_{smart} \mid \mathbf{b} \in \Delta_{smart}(e)\} \\ &= \{(v, w) \in E_{smart} \mid \mathbf{b} \in \Delta_{smart}(v), \mathbf{b} \in \Delta_{smart}(w)\} && \left. \begin{array}{l} \text{Eq. 10} \\ \text{def } E_{smart} \end{array} \right\} \\ &= \{(v, w) \in (E_b \cap (V_{smart} \times V_{smart})) \mid \mathbf{b} \in \Delta_{smart}(v), \mathbf{b} \in \Delta_{smart}(w)\} \\ &= \{(v, w) \in E_b \mid (v, w) \in V_{smart} \times V_{smart}, \mathbf{b} \in \Delta_{smart}(v), \mathbf{b} \in \Delta_{smart}(w)\} \\ &= \{(v, w) \in E_b \mid v \in V_{smart}, w \in V_{smart}, \mathbf{b} \in \Delta_{smart}(v), \mathbf{b} \in \Delta_{smart}(w)\} \\ &= \{(v, w) \in E_b \mid v \in \{v \in V_{smart} \mid \mathbf{b} \in \Delta_{smart}(v)\}, w \in \{v \in V_{smart} \mid \mathbf{b} \in \Delta_{smart}(v)\}\} && \left. \begin{array}{l} (I) \\ \text{def } E_{viewtree} \end{array} \right\} \\ &= \{(v, w) \in E_b \mid v \in V_{viewtree}, w \in V_{viewtree}\} \\ &= \{(v, w) \in E_b \mid (v, w) \in V_{viewtree} \times V_{viewtree}\} \\ &= E_b \cap (V_{viewtree} \times V_{viewtree}) \\ &= E_{viewtree} \end{aligned}$$

**Case  $t = a$ :**

Analogous to the first case  $t = b$ .

□