

Dependent Types for the Verification of
Information Flow and Access Control Policies: Online Appendix

Version of May 18, 2012

Aleksandar Nanevski Anindya Banerjee
IMDEA Software Institute
{aleks.nanevski, anindya.banerjee}@imdea.org

Deepak Garg
Max Planck Institute for Software Systems
dg@mpi-sws.org

1 Right Reduction Rules

$$\begin{array}{c}
\text{VAL_RETV} \frac{r \ v \ i}{\text{verify } i \ (\text{return } @ \ v) \ r} \qquad \text{BND_RETV} \frac{\text{verify } i \ (e \ @ \ v) \ r}{\text{verify } i \ ((\text{return } @ \ v); e) \ r} \\
\\
\text{VAL_READV} \frac{r \ v \ (\ell \mapsto v \bullet i)}{\text{verify } (\ell \mapsto v \bullet i) \ (\text{read } @ \ l) \ r} \qquad \text{BND_READV} \frac{\text{verify } (\ell \mapsto v \bullet i) \ (e \ @ \ v) \ r}{\text{verify } (\ell \mapsto v \bullet i) \ ((\text{read } @ \ l); e) \ r} \\
\\
\text{VAL_WRITEV} \frac{r \ () \ (\ell \mapsto v \bullet i)}{\text{verify } (\ell \mapsto w \bullet i) \ (\text{write } @ \ (l, v)) \ r} \\
\\
\text{BND_WRITEV} \frac{\text{verify } (\ell \mapsto v \bullet i) \ (e \ @ \ ()) \ r}{\text{verify } (\ell \mapsto w \bullet i) \ ((\text{write } @ \ (l, v)); e) \ r} \qquad \text{VAL_LALLOCV} \frac{\forall \ell : \text{ptr}. r \ \ell \ (\ell \mapsto v \bullet i)}{\text{verify } i \ (\text{lalloc } @ \ v) \ r} \\
\\
\text{BND_LALLOCV} \frac{\forall \ell : \text{ptr}. \text{verify } (\ell \mapsto v \bullet i) \ (e \ @ \ \ell) \ r}{\text{verify } i \ ((\text{lalloc } @ \ v); e) \ r} \\
\\
\text{VAL_HALLOCV} \frac{\forall \ell : \text{ptr}. \text{even } \ell \Rightarrow r \ \ell \ (\ell \mapsto v \bullet i)}{\text{verify } i \ (\text{alloc } @ \ v) \ r} \\
\\
\text{BND_HALLOCV} \frac{\forall \ell : \text{ptr}. \text{verify } (\ell \mapsto v \bullet i) \ (e \ @ \ \ell) \ (\text{fun } y \ m. \ \text{even } \ell \Rightarrow r \ y \ m)}{\text{verify } i \ ((\text{alloc } @ \ v); e) \ r} \\
\\
\text{VAL_DEALLOCV} \frac{r \ () \ i}{\text{verify } (\ell \mapsto v \bullet i) \ (\text{dealloc } @ \ \ell) \ r} \\
\\
\text{BND_DEALLOCV} \frac{\text{verify } i \ (e \ @ \ ()) \ r}{\text{verify } (\ell \mapsto v \bullet i) \ ((\text{dealloc } @ \ l); e) \ r} \qquad \text{VAL_INSTV} \frac{\text{verify } i \ (e_1 \ @ \ (f a)) \ r}{\text{verify } i \ (e_1 \ @ \ f \ @ \ a) \ r} \\
\\
\text{BND_PUSHV} \frac{\text{verify } i \ ((e_1 \ @ \ a); (e_2 \ @_1 \ (\text{fun } y. \ (y, a)))) \ r}{\text{verify } i \ ((e_1; e_2) \ @ \ a) \ r} \qquad \text{BND_INSTV} \frac{\text{verify } i \ ((e_1 \ @ \ (f a)); e_2) \ r}{\text{verify } i \ ((e_1 \ @ \ f \ @ \ a); e_2) \ r}
\end{array}$$

Figure 1: Right reduction rules

2 Both Ways Reduction Rules

$$\begin{array}{c}
\text{VAL_RET} \frac{r(v_1, v_2)(i_1, i_2)}{\text{verify2 } i_1 \ i_2 \ (\text{return } @ \ v_1) \ (\text{return } @ \ v_2) \ r} \\
\\
\text{BND_RET} \frac{\text{verify2 } i_1 \ i_2 \ (e_1 \ @ \ v_1) \ (e_2 \ @ \ v_2) \ r}{\text{verify2 } i_1 \ i_2 \ ((\text{return } @ \ v_1); e_1) \ ((\text{return } @ \ v_2); e_2) \ r} \\
\\
\text{VAL_READ} \frac{r \ (v_1, v_2) \ (l_1 \mapsto v_1 \bullet i_1, l_2 \mapsto v_2 \bullet i_2)}{\text{verify2 } (l_1 \mapsto v_1 \bullet i_1) \ (l_2 \mapsto v_2 \bullet i_2) \ (\text{read } @ \ l_1) \ (\text{read } @ \ l_2) \ r} \\
\\
\text{BND_READ} \frac{\text{verify2 } (l_1 \mapsto v_1 \bullet i_1) \ (l_2 \mapsto v_2 \bullet i_2) \ (e_1 \ @ \ v_1) \ (e_2 \ @ \ v_2) \ r}{\text{verify2 } (l_1 \mapsto v_1 \bullet i_1) \ (l_2 \mapsto v_2 \bullet i_2) \ ((\text{read } @ \ l_1); e_1) \ ((\text{read } @ \ l_2); e_2) \ r} \\
\\
\text{VAL_WRITE} \frac{r \ ((), ()) \ (l_1 \mapsto v_1 \bullet i_1, l_2 \mapsto v_2 \bullet i_2)}{\text{verify2 } (l_1 \mapsto w_1 \bullet i_1) \ (l_2 \mapsto w_2 \bullet i_2) \ (\text{write } @ \ (l_1, v_1)) \ (\text{write } @ \ (l_2, v_2)) \ r} \\
\\
\text{BND_WRITE} \frac{\text{verify2 } (l_1 \mapsto v_1 \bullet i_1) \ (l_2 \mapsto v_2 \bullet i_2) \ (e_1 \ @ \ ()) \ (e_2 \ @ \ ()) \ r}{\text{verify2 } (l_1 \mapsto v_1 \bullet i_1) \ (l_2 \mapsto v_2 \bullet i_2) \ ((\text{write } @ \ (l_1, v_1)); e_1) \ ((\text{write } @ \ (l_2, v_2)); e_2) \ r} \\
\\
\text{VAL_LALLOC} \frac{\forall l_1, l_2 : ptr. i_1 \cong i_2 \Rightarrow (l_1 = l_2) \wedge (l_1 \mapsto v_1 \bullet i_1 \cong l_2 \mapsto v_2 \bullet i_2) \\ \forall l_1, l_2 : ptr. r \ (l_1, l_2) \ (l_1 \mapsto v_1 \bullet i_1, l_2 \mapsto v_2 \bullet i_2)}{\text{verify2 } i_1 \ i_2 \ (\text{lalloc } @ \ v_1) \ (\text{lalloc } @ \ v_2) \ r} \\
\\
\text{BND_LALLOC} \frac{\forall l_1, l_2 : ptr. \text{verify2 } (l_1 \mapsto v_1 \bullet i_1) \ (l_2 \mapsto v_2 \bullet i_2) \ (e_1 \ @ \ l_1) \ (e_2 \ @ \ l_2) \ r' \\ r' = \text{fun } y \ m. \ (i_1 \cong i_2 \wedge l_1 = l_2 \wedge l_1 \mapsto v_1 \bullet i_1 \cong l_2 \mapsto v_2 \bullet i_2) \Rightarrow r \ y \ m}{\text{verify2 } i_1 \ i_2 \ ((\text{lalloc } @ \ v_1); e_1) \ ((\text{lalloc } @ \ v_2); e_2) \ r}
\end{array}$$

Figure 2: Both ways reduction rules: 1 of 2

$$\begin{array}{c}
\text{VAL_HALLOC} \frac{\forall \ell_1, \ell_2 : ptr. \text{even } \ell_1 \wedge \text{even } \ell_2 \Rightarrow r(\ell_1, \ell_2) \ (\ell_1 \mapsto v_1 \bullet i_1, \ell_2 \mapsto v_2 \bullet i_2)}{\text{verify2 } i_1 i_2 (\text{alloc } @ v_1) (\text{alloc } @ v_2) r} \\
\\
\text{BND_HALLOC} \frac{\forall \ell_1, \ell_2 : ptr. \text{verify2 } (\ell_1 \mapsto v_1 \bullet i_1) (\ell_2 \mapsto v_2 \bullet i_2) (e_1 @ \ell_1) (e_2 @ \ell_2) r' \\ r' = \text{fun } yy \text{ mm}. (\text{even } \ell_1 \wedge \text{even } \ell_2) \Rightarrow r \ yy \text{ mm}}{\text{verify2 } i_1 i_2 ((\text{alloc } @ v_1); e_1) ((\text{alloc } @ v_2); e_2) r} \\
\\
\text{VAL_DEALLOC} \frac{r ((), ()) (i_1, i_2)}{\text{verify2 } (\ell_1 \mapsto v_1 \bullet i_1) (\ell_2 \mapsto v_2 \bullet i_2) (\text{dealloc } @ \ell_1) (\text{dealloc } @ \ell_2) r} \\
\\
\text{BND_DEALLOC} \frac{\text{verify2 } i_1 i_2 (e_1 @ ()) (e_2 @ ()) r}{\text{verify2 } (\ell_1 \mapsto v_1 \bullet i_1) (\ell_2 \mapsto v_2 \bullet i_2) ((\text{dealloc } @ \ell_1); e_1) ((\text{dealloc } @ \ell_2); e_2) r} \\
\\
\text{VAL_INSTR} \frac{\text{verify2 } i j e_2 (e_1 @ (f a)) r}{\text{verify2 } i j e_2 (e_1 @ f @ a) r} \\
\\
\text{BND_PUSHR} \frac{\text{verify2 } i j e_3 ((e_1 @ a); (e_2 @_1 (\text{fun } y. (y, a)))) r}{\text{verify2 } i j e_3 ((e_1; e_2) @ a) r} \\
\\
\text{BND_INSTR} \frac{\text{verify2 } i j e_3 ((e_1 @ (f a)); e_2) r}{\text{verify2 } i j e_3 ((e_1 @ f @ a); e_2) r}
\end{array}$$

Figure 3: Both ways reduction rules: 2 of 2