
Project: DRA FRONT END FILTER PROJECT

Title: Phase 3 Theory Listings

Ref: DS/FMU/FEF/045

Issue: Revision : 2.1

Date: 5 June 2016

Status: Approved

Type: Specification

Keywords:

Author:

<i>Name</i>	<i>Location</i>	<i>Signature</i>	<i>Date</i>
R. B. Jones	WIN01		

Authorisation for Issue:

<i>Name</i>	<i>Function</i>	<i>Signature</i>	<i>Date</i>
R.B. Jones	HAT Manager		

Abstract: A collection of listings of theories developed under Phase 3 of the FEF contract, together with an index. (for DRA Front End Filter project RSRE 1C/6130.)

Distribution: HAT FEF File
Simon Wiseman

0 DOCUMENT CONTROL

0.1 Contents List

0	DOCUMENT CONTROL	2
0.1	Contents List	2
0.2	Document Cross References	3
0.3	Changes History	3
0.4	Changes Forecast	3
1	GENERAL	4
1.1	Scope	4
2	THE THEORY wrk057	5
2.1	Parents	5
2.2	Children	5
2.3	Constants	5
2.4	Types	5
2.5	Definitions	5
2.6	Theorems	6
3	THE THEORY fef040	11
3.1	Parents	11
3.2	Children	11
3.3	Constants	11
3.4	Type Abbreviations	11
3.5	Fixity	11
3.6	Definitions	12
3.7	Theorems	13
4	THE THEORY fef042	15
4.1	Parents	15
4.2	Children	15
4.3	Constants	15
4.4	Types	16
4.5	Type Abbreviations	16
4.6	Definitions	16
4.7	Theorems	18
5	THE THEORY fef043	22
5.1	Parents	22
5.2	Constants	22
5.3	Types	23
5.4	Type Abbreviations	23
5.5	Fixity	23
5.6	Definitions	23
5.7	Theorems	26
6	INDEX	29

0.2 Document Cross References

- [1] DS/FMU/FEF/040. *Multi-level Formal Security Policy*. R.D. Arthan, ICL Secure Systems, WIN01.
- [2] DS/FMU/FEF/042. *Multi-level Architectural Model*. R.D. Arthan, ICL Secure Systems, WIN01.
- [3] DS/FMU/FEF/043. *The Labelling Property for SWORD*. R.B. Jones, ICL Secure Systems, WIN01.
- [4] DS/FMU/IED/wrk057. *Examples of HOL Type Definitions*. R.D. Arthan, Lemma 1 Ltd., <http://www.lemma-one.com>.

0.3 Changes History

Issue 1.1 (11 March 1994) First draft.

Issue Revision : 2.1 (5 June 2016) Final approved version.

Issue 2.2 Removed dependency on ICL logo font

0.4 Changes Forecast

None.

1 GENERAL

1.1 Scope

This document contains listings of the theories developed under Phase 3 of the FEF contract [1, 2, 3], together with an index. The theory *wrk057*[4] is also included since its listing is not available elsewhere.

2 THE THEORY wrk057

2.1 Parents

lib_thms

2.2 Children

fef042

2.3 Constants

FinitaryRecType

$$\begin{aligned}
& ('D \rightarrow 'T) \\
& \leftrightarrow (('D \rightarrow 'T \ \mathbb{P}) \times ('T \rightarrow \mathbb{N}) \times (('T \rightarrow 'y) \rightarrow 'D \rightarrow 'Y))
\end{aligned}$$

LocalFunctional

$$('T \rightarrow 'y) \rightarrow 'D \rightarrow 'Y \leftrightarrow ('D \rightarrow 'T \ \mathbb{P})$$

Tree

$$(\mathbb{N} \times 'a) \ \text{LIST} \ \mathbb{P}$$

Unparse

$$(\mathbb{N} \times 'a) \ \text{LIST} \rightarrow \mathbb{N} \rightarrow (\mathbb{N} \times 'a) \ \text{LIST}$$

MkTree

$$'a \times 'a \ \text{TREE} \ \text{LIST} \rightarrow 'a \ \text{TREE}$$

2.4 Types

'1 TREE

2.5 Definitions

FinitaryRecType

$$\begin{aligned}
& \vdash \text{FinitaryRecType} \\
& = \{(k, c, w, M) \\
& \quad | \text{OneOne } k \\
& \quad \wedge (\forall t \bullet \exists x \bullet c \ x \subseteq \{z | w \ z < w \ t\} \wedge t = k \ x) \\
& \quad \wedge (\forall i \ g_1 \ g_2 \\
& \quad \bullet (\forall y \bullet w \ y < i \Rightarrow g_1 \ y = g_2 \ y) \\
& \quad \Rightarrow (\forall x \\
& \quad \bullet c \ x \subseteq \{y | w \ y < i\} \Rightarrow M \ g_1 \ x = M \ g_2 \ x))\}
\end{aligned}$$

LocalFunctional

$$\begin{aligned}
& \vdash \text{LocalFunctional} \\
& = \{(M, c) \\
& \quad | \forall I \ g_1 \ g_2 \\
& \quad \bullet (\forall y \bullet y \in I \Rightarrow g_1 \ y = g_2 \ y) \\
& \quad \Rightarrow (\forall x \bullet c \ x \subseteq I \Rightarrow M \ g_1 \ x = M \ g_2 \ x)\}
\end{aligned}$$

Tree

$$\begin{aligned}
& \vdash \text{Tree} \\
& = \bigcap \\
& \quad \{A \\
& \quad | \forall \text{lab trees} \\
& \quad \bullet \text{Elems trees} \subseteq A
\end{aligned}$$

$\Rightarrow \text{Cons } (\# \text{ trees}, \text{ lab}) (\text{Flat trees}) \in A\}$

Unparse $\vdash \forall i \text{ nv more}$

- $\text{Unparse } [] \ i = []$
- $\wedge \text{Unparse } (\text{Cons } \text{nv more}) \ i$
- $= (\text{if } i = 0$
- $\text{then } []$
- else
- $\text{Cons } \text{nv } (\text{Unparse } \text{more } ((\text{Fst } \text{nv} + i) - 1))$

TREE**tree_def** $\vdash \exists f \bullet \text{TypeDefn } (\lambda t \bullet t \in \text{Tree}) \ f$ **MkTree** $\vdash \text{ConstSpec}$ $(\lambda \text{MkTree}'$

- $\text{OneOne } \text{MkTree}'$

 $\wedge (\exists w$

- $\forall t$

- $\exists x$

- $(\forall z \bullet z \in \text{Elems } (\text{Snd } x) \Rightarrow w \ z < w \ t)$

- $\wedge t = \text{MkTree}' \ x)$

 MkTree **2.6 Theorems** **\wedge_empty_thm** $\vdash \forall l1 \ l2 \bullet l1 \wedge l2 = [] \Leftrightarrow l1 = [] \wedge l2 = []$ **flat_empty_thm** $\vdash \forall ls \bullet \text{Flat } ls = [] \Leftrightarrow \text{Elems } ls \subseteq \{[]\}$ **append_assoc_thm** $\vdash \forall l1 \ l2 \ l3 \bullet (l1 \wedge l2) \wedge l3 = l1 \wedge l2 \wedge l3$ **flat_append_thm** $\vdash \forall ls1 \ ls2 \bullet \text{Flat } (ls1 \wedge ls2) = \text{Flat } ls1 \wedge \text{Flat } ls2$ **length_append_thm** $\vdash \forall ls1 \ ls2 \bullet \# (ls1 \wedge ls2) = \# ls1 + \# ls2$ **elems_append_thm** $\vdash \forall l1 \ l2 \bullet \text{Elems } (l1 \wedge l2) = \text{Elems } l1 \cup \text{Elems } l2$ **append_empty_thm** $\vdash \forall l \bullet l \wedge [] = l$ **append_cancel_thm** $\vdash \forall l1 \ l2 \ l3 \bullet l1 \wedge l2 = l1 \wedge l3 \Leftrightarrow l2 = l3$ **map_map_id_thm** $\vdash \forall f \ g \ l$

- $(\forall x \bullet x \in \text{Elems } l \Rightarrow f (g \ x) = x)$

 $\Rightarrow \text{Map } f (\text{Map } g \ l) = l$ **length_length_flat_thm** $\vdash \forall ll \bullet l \in \text{Elems } ll \Rightarrow \# l \leq \# (\text{Flat } ll)$ **elems_map_thm** $\vdash \forall f \ l$

- $\text{Elems } (\text{Map } f \ l) = \{y \mid \exists x \bullet x \in \text{Elems } l \wedge f \ x = y\}$

length_0_thm $\vdash \forall l \bullet \# l = 0 \Leftrightarrow l = []$

one_one_left_inv_thm

$$\vdash \forall f \bullet \text{OneOne } f \Rightarrow (\exists g \bullet \forall x \bullet g (f x) = x)$$
onto_right_inv_thm

$$\vdash \forall f \bullet \text{Onto } f \Rightarrow (\exists g \bullet \forall y \bullet f (g y) = y)$$
one_one_onto_inv_thm

$$\vdash \forall f$$

- $\text{OneOne } f \wedge \text{Onto } f$

$$\Rightarrow (\exists g \bullet (\forall x \bullet g (f x) = x) \wedge (\forall y \bullet f (g y) = y))$$
fin_rec_type_induction_thm

$$\vdash \forall k c w M$$

- $(k, c, w, M) \in \text{FinitaryRecType}$

$$\Rightarrow (\forall X \bullet (\forall x \bullet c x \subseteq X \Rightarrow k x \in X) \Rightarrow (\forall t \bullet t \in X))$$
fin_rec_type_induction_thm1

$$\vdash \forall k c w M$$

- $(k, c, w, M) \in \text{FinitaryRecType}$

$$\Rightarrow (\forall P$$

- $(\forall x \bullet (\forall t \bullet t \in c x \Rightarrow P t) \Rightarrow P (k x))$

$$\Rightarrow (\forall t \bullet P t))$$
fin_rec_type_prim_rec_lemma1

$$\vdash \forall k c w$$

- $(k, c, w, M) \in \text{FinitaryRecType}$

$$\Rightarrow (\exists \delta$$

- $(\forall x \bullet \delta (k x) = x)$

$$\wedge (\forall y \bullet k (\delta y) = y)$$

$$\wedge (\forall d$$

- $\exists H$

- $(\forall y \bullet H 0 y = d (\delta y, M \text{Arbitrary } (\delta y)))$

$$\wedge (\forall i y$$

- $H (i + 1) y$

$$= (\text{if } w y \leq i$$

$$\text{then } H i y$$

$$\text{else } d (\delta y, M (H i) (\delta y)))$$

$$\wedge (\forall j y \bullet H (w y + j) y = H (w y) y))$$
fin_rec_type_prim_rec_exists_thm

$$\vdash \forall k c w M$$

- $(k, c, w, M) \in \text{FinitaryRecType}$

$$\Rightarrow (\forall d \bullet \exists h \bullet \forall x \bullet h (k x) = d (x, M h x))$$
fin_rec_type_prim_rec_unique_thm

$$\vdash \forall k c w M$$

- $(k, c, w, M) \in \text{FinitaryRecType}$

$$\Rightarrow (\forall d h_1 h_2$$

- $(\forall x \bullet h_1 (k x) = d (x, M h_1 x))$

$$\wedge (\forall x \bullet h_2 (k x) = d (x, M h_2 x))$$

$$\Rightarrow h_1 = h_2)$$
fin_rec_type_prim_rec_thm

$$\vdash \forall k c w M$$

- $(k, c, w, M) \in \text{FinitaryRecType}$

$$\Rightarrow (\forall d \bullet \exists_1 h \bullet \forall x \bullet h (k x) = d (x, M h x))$$

local_functional_thm

$$\begin{aligned} &\vdash \forall c w M \\ &\bullet (M, c) \in \text{LocalFunctional} \\ &\Rightarrow (\forall i g_1 g_2 \\ &\bullet (\forall y \bullet w y < i \Rightarrow g_1 y = g_2 y) \\ &\Rightarrow (\forall x \\ &\bullet c x \subseteq \{y \mid w y < i\} \Rightarrow M g_1 x = M g_2 x)) \end{aligned}$$

i_local_thm $\vdash ((\lambda f \bullet f), (\lambda x \bullet \{x\})) \in \text{LocalFunctional}$

k_local_thm $\vdash ((\lambda f x \bullet \{ \}), (\lambda x \bullet \{x\})) \in \text{LocalFunctional}$

×_local_thm $\vdash \forall M_1 c_1 M_2 c_2$
 $\bullet (M_1, c_1) \in \text{LocalFunctional}$
 $\wedge (M_2, c_2) \in \text{LocalFunctional}$
 $\Rightarrow ((\lambda f (x, y) \bullet (M_1 f x, M_2 f y)),$
 $(\lambda (x, y) \bullet c_1 x \cup c_2 y))$
 $\in \text{LocalFunctional}$

sum_local_thm

$$\begin{aligned} &\vdash \forall M_1 c_1 M_2 c_2 \\ &\bullet (M_1, c_1) \in \text{LocalFunctional} \\ &\wedge (M_2, c_2) \in \text{LocalFunctional} \\ &\Rightarrow ((\lambda f x \\ &\bullet \text{if } \text{IsL } x \\ &\quad \text{then } M_1 f (\text{OutL } x) \\ &\quad \text{else } M_2 f (\text{OutR } x)), \\ &(\lambda x \\ &\bullet \text{if } \text{IsL } x \\ &\quad \text{then } c_1 (\text{OutL } x) \\ &\quad \text{else } c_2 (\text{OutR } x))) \\ &\in \text{LocalFunctional} \end{aligned}$$
list_local_thm

$$\begin{aligned} &\vdash \forall M c \\ &\bullet (M, c) \in \text{LocalFunctional} \\ &\Rightarrow ((\lambda f \bullet \text{Map } (M f)), (\lambda x \bullet \cup (\text{Elems } (\text{Map } c x)))) \\ &\in \text{LocalFunctional} \end{aligned}$$
list_local_thm1

$$\vdash (\text{Map}, \text{Elems}) \in \text{LocalFunctional}$$
o_snd_local_thm

$$\begin{aligned} &\vdash \forall M c \\ &\bullet (M, c) \in \text{LocalFunctional} \\ &\Rightarrow ((\lambda f \bullet M f \circ \text{Snd}), c \circ \text{Snd}) \in \text{LocalFunctional} \end{aligned}$$
o_fst_local_thm

$$\begin{aligned} &\vdash \forall M c \\ &\bullet (M, c) \in \text{LocalFunctional} \\ &\Rightarrow ((\lambda f \bullet M f \circ \text{Fst}), c \circ \text{Fst}) \in \text{LocalFunctional} \end{aligned}$$
tree_induction_lemma1

$$\begin{aligned} &\vdash \forall X \\ &\bullet (\forall x ts \\ &\bullet \text{Elems } ts \subseteq X \Rightarrow \text{Cons } (\# ts, x) (\text{Flat } ts) \in X) \\ &\Rightarrow \text{Tree} \subseteq X \end{aligned}$$

tree_induction_lemma2 $\vdash \forall x ts$

- $Elms\ ts \subseteq Tree \Rightarrow Cons\ (\#\ ts,\ x)\ (Flat\ ts) \in Tree$

tree_induction_lemma $\vdash \forall X$

- $(\forall x ts$
 - $Elms\ ts \subseteq Tree \cap X$
 - $\Rightarrow Cons\ (\#\ ts,\ x)\ (Flat\ ts) \in X$
 - $\Rightarrow Tree \subseteq X$

tree_induction_tac_lemma $\vdash \forall P$

- $(\forall x ts$
 - $(\forall t \bullet t \in Elms\ ts \Rightarrow t \in Tree \wedge P\ t)$
 - $\Rightarrow P\ (Cons\ (\#\ ts,\ x)\ (Flat\ ts))$
 - $\Rightarrow (\forall t \bullet t \in Tree \Rightarrow P\ t)$

tree_cases_lemma $\vdash \forall t$

- $t \in Tree$
- $\Rightarrow (\exists x ts$
 - $Elms\ ts \subseteq Tree \wedge t = Cons\ (\#\ ts,\ x)\ (Flat\ ts)$

 \neg _empty_list_tree_lemma $\vdash \neg [] \in Tree$ **unparse_thm** $\vdash \forall ts\ more$

- $Elms\ ts \subseteq Tree$
- $\Rightarrow Unparse\ (Flat\ ts \wedge more)\ (\#\ ts) = Flat\ ts$

unparse_thm1 $\vdash \forall t\ more \bullet t \in Tree \Rightarrow Unparse\ (t \wedge more)\ 1 = t$ **tree_cases_lemma1** $\vdash \forall t$

- $t \in Tree$
- $\Rightarrow (\exists_1\ (x,\ ts)$
 - $Elms\ ts \subseteq Tree \wedge t = Cons\ (\#\ ts,\ x)\ (Flat\ ts)$

leaf_is_a_tree_thm $\vdash \forall x \bullet [(0,\ x)] \in Tree$ **MkTree_consistent** $\vdash Consistent$ $(\lambda\ MkTree'$

- $OneOne\ MkTree'$
- $\wedge (\exists w$
 - $\forall t$
 - $\exists x$
 - $(\forall z \bullet z \in Elms\ (Snd\ x) \Rightarrow w\ z < w\ t)$
 - $\wedge t = MkTree'\ x)$

tree_local_thm $\vdash ((\lambda\ f \bullet Map\ f\ o\ Snd), Elms\ o\ Snd) \in LocalFunctional$ **tree_fin_rec_thm** $\vdash \exists w$

- $(MkTree,\ Elms\ o\ Snd,\ w,\ (\lambda\ f \bullet Map\ f\ o\ Snd))$
- $\in FinitaryRecType$

tree_induction_thm $\vdash \forall P$ $\bullet (\forall x$ $\bullet (\forall t \bullet t \in \text{Elems } (\text{Snd } x) \Rightarrow P t) \Rightarrow P (\text{MkTree } x))$ $\Rightarrow (\forall t \bullet P t)$ **tree_prim_rec_thm** $\vdash \forall d \bullet \exists_1 h \bullet \forall x \bullet h (\text{MkTree } x) = d x (\text{Map } h (\text{Snd } x))$

3 THE THEORY fef040

3.1 Parents

fef010 fef003

3.2 Children

fef042

3.3 Constants

$\$ \downarrow$	$Class \rightarrow Class \mathbb{P}$
$\$ \uparrow$	$Class \rightarrow Class \mathbb{P}$
$\$ \perp$	$Class \mathbb{P} \rightarrow Class \mathbb{P}$
Equivalence	$('a \leftrightarrow 'a) \mathbb{P}$
IndexedEquiv	$(Class \mathbb{P} \rightarrow 'a \leftrightarrow 'a) \mathbb{P}$
LiftRel	$(Class \rightarrow 'a \leftrightarrow 'a) \rightarrow Class \mathbb{P} \rightarrow 'a \leftrightarrow 'a$
Independent	$(Class \mathbb{P} \rightarrow 'a \leftrightarrow 'a) \mathbb{P}$
x_ml_secure	$(Class \mathbb{P} \rightarrow 'I \leftrightarrow 'I) \rightarrow (Class \mathbb{P} \rightarrow 'O \leftrightarrow 'O) \rightarrow ('I \rightarrow 'O) \mathbb{P}$
ml_secure	$(Class \rightarrow 'I \leftrightarrow 'I) \rightarrow (Class \rightarrow 'O \leftrightarrow 'O) \rightarrow ('I \rightarrow 'O) \mathbb{P}$
Influenced	$(Class \mathbb{P} \rightarrow 'a \leftrightarrow 'a) \rightarrow 'a \rightarrow ('a \rightarrow 'b) \rightarrow Class \mathbb{P}$
ObservedValue	$('a, 'b) OBSERVATION \rightarrow 'a \rightarrow Class \times 'b$
SameLabVal	$Class \rightarrow (Class \times 'b) \leftrightarrow (Class \times 'b)$
BoundedObs	$(Class \mathbb{P} \rightarrow 'a \leftrightarrow 'a)$ $\rightarrow Class \leftrightarrow (('a \rightarrow Class) \times ('a \rightarrow 'b))$

3.4 Type Abbreviations

$('a, 'b) OBSERVATION$
 $('a, 'b) OBSERVATION$

3.5 Fixity

Postfix 300: \downarrow \uparrow \perp

3.6 Definitions

\downarrow	$\vdash \forall c \bullet c \downarrow = \{d \mid c \text{ dominates } d\}$
\uparrow	$\vdash \forall c \bullet c \uparrow = \{d \mid d \text{ dominates } c\}$
\perp	$\vdash \forall A$ <ul style="list-style-type: none"> • A^\perp $= \{c$ $\mid \forall d \bullet d \in A \Rightarrow \neg c \text{ dominates } d \wedge \neg d \text{ dominates } c\}$
Equivalence	$\vdash \text{Equivalence} = \text{Reflexive} \cap \text{Symmetric} \cap \text{Transitive}$
IndexedEquiv	$\vdash \text{IndexedEquiv}$ $= \{s$ $\mid (\forall A \bullet s A \in \text{Equivalence})$ $\wedge (\forall A B \bullet A \subseteq B \Rightarrow s B \subseteq s A)\}$
LiftRel	$\vdash \forall R \bullet \text{LiftRel } R = (\lambda A \bullet \cap \{r \mid \exists c \bullet c \in A \wedge r = R c\})$
Independent	$\vdash \text{Independent}$ $= \{s$ $\mid \forall A x$ <ul style="list-style-type: none"> • $(\exists y \bullet \neg (x, y) \in s A)$ $\Rightarrow (\exists z \bullet \neg (x, z) \in s A \wedge (x, z) \in s (A^\perp))\}$
x_ml_secure	$\vdash \forall s_I s_O b$ <ul style="list-style-type: none"> • $b \in x_ml_secure s_I s_O$ $\Leftrightarrow s_I \in \text{IndexedEquiv}$ $\wedge s_O \in \text{IndexedEquiv}$ $\wedge (\forall c i_1 i_2$ <ul style="list-style-type: none"> • $(i_1, i_2) \in s_I (c \downarrow)$ $\Rightarrow (b i_1, b i_2) \in s_O (c \downarrow))$
ml_secure	$\vdash \forall r_I r_O$ <ul style="list-style-type: none"> • $ml_secure r_I r_O$ $= x_ml_secure (\text{LiftRel } r_I) (\text{LiftRel } r_O)$
Influenced	$\vdash \forall s x f$ <ul style="list-style-type: none"> • $\text{Influenced } s x f$ $= \{c$ $\mid \exists y \bullet (x, y) \in s (\sim (c \uparrow)) \wedge \neg f x = f y\}$
ObservedValue	$\vdash \forall c C V x \bullet \text{ObservedValue } (c, C, V) x = (C x, V x)$
SameLabVal	$\vdash \forall c$ <ul style="list-style-type: none"> • $\text{SameLabVal } c$ $= \{((c_1, v_1), c_2, v_2)$ $\mid c_1 = c_2 \wedge (c \text{ dominates } c_1 \Rightarrow v_1 = v_2)\}$
BoundedObs	$\vdash \forall s$ <ul style="list-style-type: none"> • $\text{BoundedObs } s$ $= \{(c, C, V)$ $\mid \forall x$ <ul style="list-style-type: none"> • $\text{Influenced } s x C \subseteq c \downarrow$ $\wedge \text{Influenced } s x V \subseteq C x \downarrow\}$

3.7 Theorems

$\downarrow\downarrow$ -thm $\vdash \forall l a b \bullet l \downarrow a \downarrow b = l \downarrow (a \cap b)$

not_lattice_top_thm

$\vdash \forall c$

$\bullet \neg c = \text{lattice_top} \Rightarrow \neg \text{lattice_top} = \text{lattice_bottom}$

up_down_thm1 $\vdash \text{lattice_bottom} \uparrow = \text{Universe}$

$\wedge \text{lattice_top} \downarrow = \text{Universe}$

up_down_thm2 $\vdash \forall c$

$\bullet c \in c \downarrow$

$\wedge c \in c \uparrow$

$\wedge \text{lattice_bottom} \in c \downarrow$

$\wedge \text{lattice_top} \in c \uparrow$

up_down_clauses

$\vdash \forall c$

$\bullet (\text{lattice_bottom} \uparrow = \text{Universe}$
 $\wedge \text{lattice_top} \downarrow = \text{Universe})$

$\wedge c \in c \downarrow$

$\wedge c \in c \uparrow$

$\wedge \text{lattice_bottom} \in c \downarrow$

$\wedge \text{lattice_top} \in c \uparrow$

up_down_thm3 $\vdash \forall c d$

$\bullet \neg d \text{ dominates } c \Rightarrow d \downarrow \subseteq \sim (c \uparrow)$

lift_rel_indexed_equiv_thm

$\vdash \forall R$

$\bullet (\forall c \bullet R c \in \text{Equivalence}) \Rightarrow \text{LiftRel } R \in \text{IndexedEquiv}$

same_ins_equiv_thm

$\vdash \forall c \bullet \text{same_ins } c \in \text{Equivalence}$

same_outs_equiv_thm

$\vdash \forall c \bullet \text{same_outs } c \in \text{Equivalence}$

same_ins_same_outs_indexed_equiv_thm

$\vdash \text{LiftRel same_ins} \in \text{IndexedEquiv}$

$\wedge \text{LiftRel same_outs} \in \text{IndexedEquiv}$

equiv_anti_mono_lift_rel_thm

$\vdash \forall R$

$\bullet (\forall c \bullet R c \in \text{Equivalence})$

$\wedge (\forall c d \bullet c \text{ dominates } d \Rightarrow R c \subseteq R d)$

$\Rightarrow (\forall c \bullet \text{LiftRel } R (c \downarrow) = R c)$

same_ins_anti_mono_thm

$\vdash \forall c d \bullet c \text{ dominates } d \Rightarrow \text{same_ins } c \subseteq \text{same_ins } d$

same_outs_anti_mono_thm

$\vdash \forall c d \bullet c \text{ dominates } d \Rightarrow \text{same_outs } c \subseteq \text{same_outs } d$

lift_rel_same_ins_same_outs_down_set_thm

$\vdash \forall c$

$\bullet \text{LiftRel same_ins } (c \downarrow) = \text{same_ins } c$

$\wedge \text{LiftRel same_outs } (c \downarrow) = \text{same_outs } c$

thm_040_1

$\vdash \forall bm \bullet bm \in \text{secure} \Leftrightarrow bm \in \text{ml_secure same_ins same_outs}$

same_lab_val_equiv_thm

$$\vdash \forall c \bullet \text{SameLabVal } c \in \text{Equivalence}$$

lift_rel_same_lab_val_equiv_thm

$$\vdash \text{LiftRel SameLabVal} \in \text{IndexedEquiv}$$

same_lab_val_anti_mono_thm

$$\vdash \forall c \ d \bullet c \text{ dominates } d \Rightarrow \text{SameLabVal } c \subseteq \text{SameLabVal } d$$

lift_rel_same_lab_val_down_set_thm

$$\vdash \forall c$$

- $\text{LiftRel SameLabVal } (c \downarrow) = \text{SameLabVal } c$

thm_040_2

$$\vdash \forall s$$

- $s \in \text{IndexedEquiv}$

$$\Rightarrow (\forall c \ C \ V$$

- $\text{ObservedValue } (c, C, V)$

$$\in x_ml_secure \ s \ (\text{LiftRel SameLabVal})$$

$$\Rightarrow (c, C, V) \in \text{BoundedObs } s)$$

down_∩_comp_up_thm

$$\vdash \forall c$$

- $c \downarrow$

$$= \bigcap$$

$$\{A \mid \exists d \bullet \neg c \text{ dominates } d \wedge A = \sim (d \uparrow)\}$$

4 THE THEORY fef042

4.1 Parents

fef036 wrk057 fef040

4.2 Children

fef043

4.3 Constants

objectRefers $Obj \rightarrow Obj\ LIST$
objectContains $Obj \rightarrow Text$
objectClass $Obj \rightarrow Class$
MkObj $Class \times Text \times Obj\ LIST \rightarrow Obj$
reqSql $Req \rightarrow Obj$
reqClearance $Req \rightarrow Class$
MkReq $Class \rightarrow Obj \rightarrow Req$
identicalObj $Class \rightarrow Obj \leftrightarrow Obj$
identicalObjs $Class \rightarrow Obj\ LIST \leftrightarrow Obj\ LIST$
VisibleReq $Class \rightarrow Req\ \mathbb{P}$
sameRequest $Class \rightarrow Req \leftrightarrow Req$
sameRequests $Class \rightarrow Req\ LIST \leftrightarrow Req\ LIST$
VisibleOutput $Class \rightarrow Obj\ \mathbb{P}$
sameOutputs $Class \rightarrow Obj\ LIST \leftrightarrow Obj\ LIST$
SWORD_ml_secure $ML_BEHAVIOUR\ \mathbb{P}$
Init $'State\ Machine \rightarrow 'State$
Output $'State\ Machine \rightarrow 'State \times Req \rightarrow Obj$
Next $'State\ Machine \rightarrow 'State \times Req \rightarrow 'State$
MkMachine $('State \times Req \rightarrow 'State)$
 $\rightarrow ('State \times Req \rightarrow Obj)$
 $\rightarrow 'State$
 $\rightarrow 'State\ Machine$
lift_machine $'State\ Machine \rightarrow 'State \times Req\ LIST \rightarrow Obj\ LIST \times 'State$
Behaviours $'State\ Machine \rightarrow ML_BEHAVIOUR$
FilterObj $Class \rightarrow Obj \rightarrow Obj$
SWORD_construction $'State\ Machine \rightarrow ML_BEHAVIOUR$
sameFilterInputs $Class \rightarrow Obj\ LIST \leftrightarrow Obj\ LIST$
FlowSecureMachine $'State\ Machine\ \mathbb{P}$

4.4 Types

Req

'1 Machine

4.5 Type Abbreviations

Text *Text***Obj** *Obj***ML_BEHAVIOUR** *ML_BEHAVIOUR*

4.6 Definitions

MkObj**objectClass****objectContains****objectRefers** \vdash *ConstSpec*

$$\begin{aligned}
& (\lambda \\
& \quad (MkObj', objectClass', objectContains', \\
& \quad \quad objectRefers') \\
& \bullet \forall c t os \\
& \bullet MkObj' (c, t, os) = MkTree ((c, t), os) \\
& \quad \wedge objectClass' (MkObj' (c, t, os)) = c \\
& \quad \wedge objectContains' (MkObj' (c, t, os)) = t \\
& \quad \wedge objectRefers' (MkObj' (c, t, os)) = os \\
& \quad (MkObj, objectClass, objectContains, objectRefers)
\end{aligned}$$
Req $\vdash \exists f \bullet TypeDefn (\lambda x \bullet true) f$ **MkReq****reqClearance****reqSsql**

$$\begin{aligned}
& \vdash \forall t x1 x2 \\
& \bullet reqClearance (MkReq x1 x2) = x1 \\
& \quad \wedge reqSsql (MkReq x1 x2) = x2 \\
& \quad \wedge MkReq (reqClearance t) (reqSsql t) = t
\end{aligned}$$
identicalObj

$$\begin{aligned}
& \vdash ConstSpec \\
& \quad (\lambda identicalObj' \\
& \bullet \forall c \\
& \bullet identicalObj' c \\
& \quad = \{(o_1, o_2) \\
& \quad \mid \exists c_1 t_1 os_1 c_2 t_2 os_2 \\
& \quad \bullet o_1 = MkObj (c_1, t_1, os_1) \\
& \quad \quad \wedge o_2 = MkObj (c_2, t_2, os_2) \\
& \quad \quad \wedge c_1 = c_2 \\
& \quad \quad \wedge (c \text{ dominates } c_1 \\
& \quad \quad \Rightarrow t_1 = t_2 \\
& \quad \quad \wedge \# os_1 = \# os_2 \\
& \quad \quad \wedge Elems (Combine os_1 os_2) \\
& \quad \quad \subseteq identicalObj' c)\}
\end{aligned}$$
identicalObj

identicalObjs

$$\vdash \forall c$$

- $identicalObjs\ c$
 $= \{(s_1, s_2) \mid \# s_1 = \# s_2 \wedge Elems\ (Combine\ s_1\ s_2) \subseteq identicalObj\ c\}$

VisibleReq

$$\vdash \forall c$$

- $VisibleReq\ c$
 $= \{r \mid c\ \text{dominates}\ objectClass\ (reqSql\ r)\}$

sameRequest

$$\vdash \forall c$$

- $sameRequest\ c$
 $= \{(r_1, r_2) \mid (reqSql\ r_1, reqSql\ r_2) \in identicalObj\ c\}$

sameRequests

$$\vdash \forall c$$

- $sameRequests\ c$
 $= \{(rs_1, rs_2) \mid let\ rs_3 = rs_1 \upharpoonright VisibleReq\ c$
 $in\ let\ rs_4 = rs_2 \upharpoonright VisibleReq\ c$
 $in\ \# rs_3 = \# rs_4 \wedge Elems\ (Combine\ rs_3\ rs_4) \subseteq sameRequest\ c\}$

VisibleOutput

$$\vdash \forall c$$

- $VisibleOutput\ c = \{ob \mid c\ \text{dominates}\ objectClass\ ob\}$

sameOutputs

$$\vdash \forall c$$

- $sameOutputs\ c$
 $= \{(os_1, os_2) \mid os_1 \upharpoonright VisibleOutput\ c = os_2 \upharpoonright VisibleOutput\ c\}$

SWORD_ml_secure

$$\vdash SWORD_ml_secure = ml_secure\ sameRequests\ sameOutputs$$
Machine

$$\vdash \exists f \bullet TypeDefn\ (\lambda x \bullet true)\ f$$
MkMachine**Next****Output****Init**

$$\vdash \forall t\ x1\ x2\ x3$$

- $Next\ (MkMachine\ x1\ x2\ x3) = x1$
 $\wedge Output\ (MkMachine\ x1\ x2\ x3) = x2$
 $\wedge Init\ (MkMachine\ x1\ x2\ x3) = x3$
 $\wedge MkMachine\ (Next\ t)\ (Output\ t)\ (Init\ t) = t$

lift_machine

$$\vdash ConstSpec$$

$$(\lambda lift_machine'$$

- $\forall mch\ s\ r\ rl$
 - $lift_machine'\ mch\ (s, []) = ([], s)$
 $\wedge lift_machine'\ mch\ (s, Cons\ r\ rl)$
 $= (let\ out = Output\ mch\ (s, r)$
 $and\ s' = Next\ mch\ (s, r)$
 $in\ let\ (outl, final_state)$
 $= lift_machine'\ mch\ (s', rl)$

in (Cons out outl, final_state)))

Behaviours $\vdash \forall mch\ rs$

- *Behaviours mch rs*
- = *Fst (lift_machine mch (Init mch, rs))*

FilterObj $\vdash ConstSpec$

($\lambda FilterObj'$

- $\forall c\ d\ t\ os$
- *FilterObj' c (MkObj (d, t, os))*
- = (*if c dominates d*
- then MkObj (d, t, Map (FilterObj' c) os)*
- else MkObj (d, Arbitrary, Arbitrary))*)

FilterObj

SWORD_construction

$\vdash \forall mch$

- *SWORD_construction mch*
- = (*let sec_output (st, r)*
- = FilterObj*
- (reqClearance r)*
- (Output mch (st, r))*
- in let sec_mch*
- = MkMachine*
- (Next mch)*
- sec_output*
- (Init mch) in Behaviours sec_mch*)

sameFilterInputs

$\vdash \forall c$

- *sameFilterInputs c*
- = $\{os_1, os_2\}$
- | *let os₃ = os₁ \uparrow VisibleOutput c*
- in let os₄ = os₂ \uparrow VisibleOutput c*
- in # os₃ = # os₄*
- \wedge *Elms (Combine os₃ os₄)*
- \subseteq *identicalObj c*}

FlowSecureMachine

$\vdash FlowSecureMachine$

= $\{mch$

$|Behaviours\ mch$

$\in ml_secure\ sameRequests\ sameFilterInputs\}$

4.7 Theorems

mk_tree_one_one_thm

$\vdash \forall x\ y \bullet MkTree\ x = MkTree\ y \Rightarrow x = y$

mk_tree_onto_thm

$\vdash \forall t \bullet \exists x \bullet t = MkTree\ x$

MkObj_consistent

objectClass_consistent

objectContains_consistent**objectRefers_consistent** $\vdash \text{Consistent}$ $(\lambda$ $(\text{MkObj}', \text{objectClass}', \text{objectContains}',$
 $\text{objectRefers}')$ • $\forall c t os$ • $\text{MkObj}'(c, t, os) = \text{MkTree}((c, t), os)$
 $\wedge \text{objectClass}'(\text{MkObj}'(c, t, os)) = c$
 $\wedge \text{objectContains}'(\text{MkObj}'(c, t, os)) = t$
 $\wedge \text{objectRefers}'(\text{MkObj}'(c, t, os)) = os)$ **obj_prim_rec_thm** $\vdash \forall d$ • $\exists_1 h$ • $\forall c t os \bullet h(\text{MkObj}(c, t, os)) = d \text{ c t } (\text{Map } h \text{ os})$ **object_clauses** $\vdash (\forall x$ • $\text{MkObj } x$ $= \text{MkTree}((\text{Fst } x, \text{Fst } (\text{Snd } x)), \text{Snd } (\text{Snd } x))$ $\wedge (\forall ctos$ • $\text{objectClass}(\text{MkTree } ctos) = \text{Fst}(\text{Fst } ctos)$
 $\wedge \text{objectContains}(\text{MkTree } ctos) = \text{Snd}(\text{Fst } ctos)$
 $\wedge \text{objectRefers}(\text{MkTree } ctos) = \text{Snd } ctos)$ **lift_machine_consistent** $\vdash \text{Consistent}$ $(\lambda \text{lift_machine}'$ • $\forall mch s r rl$ • $\text{lift_machine}' mch(s, []) = ([], s)$
 $\wedge \text{lift_machine}' mch(s, \text{Cons } r rl)$
 $= (\text{let } out = \text{Output } mch(s, r)$
 $\text{and } s' = \text{Next } mch(s, r)$
 $\text{in } \text{let } (outl, \text{final_state})$
 $= \text{lift_machine}' mch(s', rl)$
 $\text{in } (\text{Cons } out \text{ outl}, \text{final_state}))$ **identicalObj_consistent** $\vdash \text{Consistent}$ $(\lambda \text{identicalObj}'$ • $\forall c$ • $\text{identicalObj}' c$ $= \{(o_1, o_2)$ $|\exists c_1 t_1 os_1 c_2 t_2 os_2$ • $o_1 = \text{MkObj}(c_1, t_1, os_1)$
 $\wedge o_2 = \text{MkObj}(c_2, t_2, os_2)$
 $\wedge c_1 = c_2$
 $\wedge (c \text{ dominates } c_1$
 $\Rightarrow t_1 = t_2$
 $\wedge \# os_1 = \# os_2$
 $\wedge \text{Elems}(\text{Combine } os_1 \text{ } os_2)$

$$\subseteq \text{identicalObj}' c\}})$$
FilterObj_consistent

$$\begin{aligned} &\vdash \text{Consistent} \\ &\quad (\lambda \text{FilterObj}' \\ &\quad \bullet \forall c d t os \\ &\quad \bullet \text{FilterObj}' c (\text{MkObj } (d, t, os)) \\ &\quad \quad = (\text{if } c \text{ dominates } d \\ &\quad \quad \text{then } \text{MkObj } (d, t, \text{Map } (\text{FilterObj}' c) os) \\ &\quad \quad \text{else } \text{MkObj } (d, \text{Arbitrary}, \text{Arbitrary})) \end{aligned}$$
elems_combine_thm

$$\begin{aligned} &\vdash \forall l x y \\ &\quad \bullet (x, y) \in \text{Elems } (\text{Combine } l) \Leftrightarrow y \in \text{Elems } l \wedge x = y \end{aligned}$$
elems_combine_elems_thm

$$\begin{aligned} &\vdash \forall l1 l2 x y \\ &\quad \bullet \# l1 = \# l2 \wedge (x, y) \in \text{Elems } (\text{Combine } l1 l2) \\ &\quad \Rightarrow x \in \text{Elems } l1 \end{aligned}$$
elems_combine_swap_thm

$$\begin{aligned} &\vdash \forall l1 l2 x1 x2 \\ &\quad \bullet \# l1 = \# l2 \wedge (x1, x2) \in \text{Elems } (\text{Combine } l1 l2) \\ &\quad \Rightarrow (x2, x1) \in \text{Elems } (\text{Combine } l2 l1) \end{aligned}$$
elems_combine_map_thm

$$\begin{aligned} &\vdash \forall f l1 l2 x1 x2 \\ &\quad \bullet \# l1 = \# l2 \\ &\quad \quad \wedge (x1, x2) \in \text{Elems } (\text{Combine } l1 l2) \\ &\quad \quad \wedge \text{Map } f l1 = \text{Map } f l2 \\ &\quad \Rightarrow f x1 = f x2 \end{aligned}$$
elems_combine_map_thm1

$$\begin{aligned} &\vdash \forall f l1 l2 \\ &\quad \bullet \# l1 = \# l2 \\ &\quad \quad \wedge (\forall y1 y2 \\ &\quad \quad \bullet (y1, y2) \in \text{Elems } (\text{Combine } l1 l2) \\ &\quad \quad \Rightarrow f y1 = f y2) \\ &\quad \Rightarrow \text{Map } f l1 = \text{Map } f l2 \end{aligned}$$
obj_induction_thm

$$\begin{aligned} &\vdash \forall P \\ &\quad \bullet (\forall c t ts \\ &\quad \quad \bullet (\forall t \bullet t \in \text{Elems } ts \Rightarrow P t) \\ &\quad \quad \Rightarrow P (\text{MkObj } (c, t, ts))) \\ &\quad \Rightarrow (\forall t \bullet P t) \end{aligned}$$
identical_obj_filter_obj_thm1

$$\begin{aligned} &\vdash \forall c ob1 ob2 \\ &\quad \bullet (ob1, ob2) \in \text{identicalObj } c \\ &\quad \Rightarrow \text{FilterObj } c ob1 = \text{FilterObj } c ob2 \end{aligned}$$
identical_obj_filter_obj_thm2

$$\begin{aligned} &\vdash \forall c ob1 ob2 \\ &\quad \bullet \text{FilterObj } c ob1 = \text{FilterObj } c ob2 \\ &\quad \Rightarrow (ob1, ob2) \in \text{identicalObj } c \end{aligned}$$
identical_obj_filter_obj_thm

$$\vdash \forall c$$

- $identicalObj\ c$
 $= \{(ob1, ob2) \mid FilterObj\ c\ ob1 = FilterObj\ c\ ob2\}$

identical_obj_refl_thm

$$\vdash \forall c\ ob \bullet (ob, ob) \in identicalObj\ c$$
identical_obj_sym_thm

$$\vdash \forall c\ ob1\ ob2$$

- $(ob1, ob2) \in identicalObj\ c$
 $\Rightarrow (ob2, ob1) \in identicalObj\ c$

identical_obj_trans_thm

$$\vdash \forall c\ ob1\ ob2\ ob3$$

- $(ob1, ob2) \in identicalObj\ c$
 $\wedge (ob2, ob3) \in identicalObj\ c$
 $\Rightarrow (ob1, ob3) \in identicalObj\ c$

identical_obj_rft_thm

$$\vdash \forall c$$

- $identicalObj\ c \in Reflexive$
 $\wedge identicalObj\ c \in Symmetric$
 $\wedge identicalObj\ c \in Transitive$

identical_obj_equiv_thm

$$\vdash \forall c \bullet identicalObj\ c \in Equivalence$$
same_requests_filter_obj_thm

$$\vdash \forall c$$

- $sameRequests\ c$
 $= \{(rs1, rs2) \mid Map\ (FilterObj\ c\ o\ reqSsql)\ (rs1 \upharpoonright VisibleReq\ c)$
 $= Map$
 $(FilterObj\ c\ o\ reqSsql)$
 $(rs2 \upharpoonright VisibleReq\ c)\}$

same_requests_equiv_thm

$$\vdash \forall c \bullet sameRequests\ c \in Equivalence$$
same_outputs_equiv_thm

$$\vdash \forall c \bullet sameOutputs\ c \in Equivalence$$
same_requests_indexed_equiv_thm

$$\vdash LiftRel\ sameRequests \in IndexedEquiv$$
same_outputs_indexed_equiv_thm

$$\vdash LiftRel\ sameOutputs \in IndexedEquiv$$

5 THE THEORY fef043

5.1 Parents

fef042

5.2 Constants

\$%f *'State Factor → 'State Factor → 'State Factor*

same_to_level *Worth → Obj ↔ Obj*

factor_level *Worth → 'State Factor ℙ*

factor3 *'State Factorisation → 'State Factor*

factor2 *'State Factorisation → 'State Factor*

factor1 *'State Factorisation → 'State Factor*

factor0 *'State Factorisation → 'State Factor*

MkFactorisation

'State Factor

→ 'State Factor

→ 'State Factor

→ 'State Factor

→ 'State Factorisation

composite *'State Factorisation → 'State Factor*

factor_out *'State Factorisation → 'State × Req → Obj*

levelled_factorisation

'State Factorisation ℙ

factors *'State FactoredMachine → 'State Factorisation*

machine *'State FactoredMachine → 'State Machine*

MkFactoredMachine

'State Machine

→ 'State Factorisation

→ 'State FactoredMachine

well_factored

'State FactoredMachine ℙ

special_machine

'State Machine × 'State Factor × 'State Factor

→ Req LIST

→ Req

→ Req LIST

→ Obj

same_at_c_below_level

Worth → Class → Obj ↔ Obj

label_secure_to

Worth

→ 'State Machine ↔ ('State Factor × 'State Factor)

label_secure *'State FactoredMachine ℙ*

simplest_witness *'State FactoredMachine*

identity_witness*'State FactoredMachine***purge_above_level***Worth → Class → Obj → Obj*

5.3 Types

*'1 Factorisation**'1 FactoredMachine*

5.4 Type Abbreviations

*'State Factor**'State Factor*

5.5 Fixity

*Right Infix 250:**%f*

5.6 Definitions

%f

$$\vdash \forall f1 f2 c s obj$$

- $(f1 \text{ \%f } f2) c obj s = f2 c (f1 c obj s) s$

same_to_level

$$\vdash \forall n obj1 obj2$$

- $same_to_level\ 0$
 $= \{(obj1, obj2)$
 $\mid objectClass\ obj1 = objectClass\ obj2\}$
 $\wedge same_to_level\ (n + 1)$
 $= \{(obj1, obj2)$
 $\mid objectContains\ obj1 = objectContains\ obj2$
 $\wedge objectClass\ obj1 = objectClass\ obj2$
 $\wedge \#(objectRefers\ obj1)$
 $= \#(objectRefers\ obj2)$
 $\wedge Elems$
 $(Combine$
 $\quad (objectRefers\ obj1)$
 $\quad (objectRefers\ obj2))$
 $\subseteq same_to_level\ n\}$

factor_level

$$\vdash \forall n$$

- $factor_level\ n$
 $= \{factor$
 $\mid \forall c\ state\ obj$
 $\quad (obj, factor\ c\ obj\ state) \in same_to_level\ n\}$

Factorisation

$$\vdash \exists f \bullet TypeDefn\ (\lambda x \bullet true)\ f$$

MkFactorisation**factor0****factor1****factor2****factor3**

$$\vdash \forall t \ x1 \ x2 \ x3 \ x4$$

- $factor0 \ (MkFactorisation \ x1 \ x2 \ x3 \ x4) = x1$
- $\wedge \ factor1 \ (MkFactorisation \ x1 \ x2 \ x3 \ x4) = x2$
- $\wedge \ factor2 \ (MkFactorisation \ x1 \ x2 \ x3 \ x4) = x3$
- $\wedge \ factor3 \ (MkFactorisation \ x1 \ x2 \ x3 \ x4) = x4$
- $\wedge \ MkFactorisation$
 - $(factor0 \ t)$
 - $(factor1 \ t)$
 - $(factor2 \ t)$
 - $(factor3 \ t)$

$$= t$$
composite

$$\vdash \forall f$$

- $composite \ f$
- $= \ factor0 \ f$
- $\%_f \ factor1 \ f$
- $\%_f \ factor2 \ f$
- $\%_f \ factor3 \ f$

factor_out

$$\vdash \forall f \ s \ r$$

- $factor_out \ f \ (s, r)$
- $= \ composite \ f \ (reqClearance \ r) \ (reqSsql \ r) \ s$

levelled_factorisation

$$\vdash \forall \ facts$$

- $facts \in \ levelled_factorisation$
- $\Leftrightarrow \ factor0 \ facts \in \ factor_level \ 0$
- $\wedge \ factor1 \ facts \in \ factor_level \ 1$
- $\wedge \ factor2 \ facts \in \ factor_level \ 2$
- $\wedge \ factor3 \ facts \in \ factor_level \ 3$

FactoredMachine

$$\vdash \exists f \bullet \ TypeDefn \ (\lambda \ x \bullet \ true) \ f$$
MkFactoredMachine**machine****factors**

$$\vdash \forall t \ x1 \ x2$$

- $machine \ (MkFactoredMachine \ x1 \ x2) = x1$
- $\wedge \ factors \ (MkFactoredMachine \ x1 \ x2) = x2$
- $\wedge \ MkFactoredMachine \ (machine \ t) \ (factors \ t) = t$

well_factored

$$\vdash \forall m$$

- $m \in \ well_factored$
- $\Leftrightarrow \ factors \ m \in \ levelled_factorisation$
- $\wedge \ Output \ (machine \ m) = \ factor_out \ (factors \ m)$

special_machine

$$\vdash \forall (m, f1, f2) \ rl1 \ rl2 \ r$$

- $special_machine \ (m, f1, f2) \ rl1 \ r \ rl2$
- $= \ (let \ s1 = \ Snd \ (lift_machine \ m \ (Init \ m, \ rl1)))$

and s2 = Snd (lift_machine m (Init m, rl2))
in let pe_req
 = f1 (reqClearance r) (reqSsql r) s1
in f2 (reqClearance r) pe_req s2)

same_at_c_below_level

$\vdash \forall n c \text{ obj1 obj2}$
 • *same_at_c_below_level 0 c = identicalObj c*
 $\wedge \text{same_at_c_below_level } (n + 1) c$
 $= \{(obj1, obj2)\}$
 $|\# (\text{objectRefers obj1}) = \# (\text{objectRefers obj2})$
 $\wedge \text{Elems}$
 (Combine
 (*objectRefers obj1*)
 (*objectRefers obj2*))
 $\subseteq \text{same_at_c_below_level } n c\}$

label_secure_to

$\vdash \forall n$
 • *label_secure_to n*
 $= \{mff$
 $|\forall rl r$
 • (*let sm = special_machine mff rl r*
 in sm
 $\in ml_secure$
 sameRequests
 (*same_at_c_below_level n*))\}

label_secure

$\vdash \text{label_secure}$
 $= \{fm$
 $|\text{let } m = \text{machine } fm \text{ and } fs = \text{factors } fm$
 in let lf1 = factor0 fs
 and lf2 = factor0 fs %f factor1 fs
 and lf3
 $= \text{factor0 } fs$
 $\%f \text{ factor1 } fs$
 $\%f \text{ factor2 } fs$
 and rf1
 $= \text{factor1 } fs$
 $\%f \text{ factor2 } fs$
 $\%f \text{ factor3 } fs$
 and rf2 = factor2 fs %f factor3 fs
 and rf3 = factor3 fs
 in (m, lf1, rf1) ∈ label_secure_to 1
 $\wedge (m, lf2, rf2) \in \text{label_secure_to } 2$
 $\wedge (m, lf3, rf3) \in \text{label_secure_to } 3\}$

simplest_witness

$\vdash \text{simplest_witness}$
 $= (\text{let next } sr = \text{Arbitrary}$
 and output } sr = \text{Arbitrary}
 and init = Arbitrary

and f0 c ob s = Arbitrary
in let (f1, f2, f3) = (f0, f0, f0)
in let mach = MkMachine next output init
and facs = MkFactorisation f0 f1 f2 f3
in MkFactoredMachine mach facs)

identity_witness

\vdash *identity_witness*
 $=$ (*let next = Fst*
and output = reqSsql o Snd
and init = Arbitrary
and f0 c ob s = ob
in let (f1, f2, f3) = (f0, f0, f0)
in let mach = MkMachine next output init
and facs = MkFactorisation f0 f1 f2 f3
in MkFactoredMachine mach facs)

purge_above_level

\vdash *ConstSpec*
 (λ *purge_above_level'*
 • (\forall *c ob*
 • *purge_above_level' 0 c ob = FilterObj c ob*)
 \wedge (\forall *n c d t os*
 • *purge_above_level'*
 (*n + 1*)
 c
 (*MkObj (d, t, os)*)
 $=$ *MkObj*
 (*Arbitrary, Arbitrary,*
 Map (purge_above_level' n c) os)))
purge_above_level

5.7 Theorems**purge_above_level_consistent**

\vdash *Consistent*
 (λ *purge_above_level'*
 • (\forall *c ob*
 • *purge_above_level' 0 c ob = FilterObj c ob*)
 \wedge (\forall *n c d t os*
 • *purge_above_level'*
 (*n + 1*)
 c
 (*MkObj (d, t, os)*)
 $=$ *MkObj*
 (*Arbitrary, Arbitrary,*
 Map (purge_above_level' n c) os)))

purge_above_level_thm

\vdash \forall *n c ob*
 • *purge_above_level (n + 1) c ob*

$$\begin{aligned}
&= \text{MkTree} \\
&\quad ((\text{Arbitrary}, \text{Arbitrary}), \\
&\quad \quad \text{Map} \\
&\quad \quad (\text{purge_above_level } n \ c) \\
&\quad \quad (\text{objectRefers } ob))
\end{aligned}$$

same_at_c_below_level_purge_above_level_thm1

$$\begin{aligned}
&\vdash \forall n \ c \ ob1 \ ob2 \\
&\quad \bullet (ob1, ob2) \in \text{same_at_c_below_level } n \ c \\
&\quad \Rightarrow \text{purge_above_level } n \ c \ ob1 \\
&\quad \quad = \text{purge_above_level } n \ c \ ob2
\end{aligned}$$

same_at_c_below_level_purge_above_level_thm2

$$\begin{aligned}
&\vdash \forall n \ c \ ob1 \ ob2 \\
&\quad \bullet \text{purge_above_level } n \ c \ ob1 \\
&\quad \quad = \text{purge_above_level } n \ c \ ob2 \\
&\quad \Rightarrow (ob1, ob2) \in \text{same_at_c_below_level } n \ c
\end{aligned}$$

same_at_c_below_level_purge_above_level_thm

$$\begin{aligned}
&\vdash \forall n \ c \ ob1 \ ob2 \\
&\quad \bullet (ob1, ob2) \in \text{same_at_c_below_level } n \ c \\
&\quad \Leftrightarrow \text{purge_above_level } n \ c \ ob1 \\
&\quad \quad = \text{purge_above_level } n \ c \ ob2
\end{aligned}$$

simplest_witness_thm

$$\begin{aligned}
&\vdash \text{machine simplest_witness} \\
&\quad = \text{MkMachine} \\
&\quad \quad (\lambda \ sr \bullet \text{Arbitrary}) \\
&\quad \quad (\lambda \ sr \bullet \text{Arbitrary}) \\
&\quad \quad \text{Arbitrary} \\
&\quad \wedge \text{factors simplest_witness} \\
&\quad \quad = \text{MkFactorisation} \\
&\quad \quad (\lambda \ c \ ob \ s \bullet \text{Arbitrary}) \\
&\quad \quad (\lambda \ c \ ob \ s \bullet \text{Arbitrary}) \\
&\quad \quad (\lambda \ c \ ob \ s \bullet \text{Arbitrary}) \\
&\quad \quad (\lambda \ c \ ob \ s \bullet \text{Arbitrary})
\end{aligned}$$

identity_witness_thm

$$\begin{aligned}
&\vdash \text{machine identity_witness} \\
&\quad = \text{MkMachine Fst (reqSsql o Snd) Arbitrary} \\
&\quad \wedge \text{factors identity_witness} \\
&\quad \quad = \text{MkFactorisation} \\
&\quad \quad (\lambda \ c \ ob \ s \bullet ob) \\
&\quad \quad (\lambda \ c \ ob \ s \bullet ob) \\
&\quad \quad (\lambda \ c \ ob \ s \bullet ob) \\
&\quad \quad (\lambda \ c \ ob \ s \bullet ob)
\end{aligned}$$

same_at_c_below_level_equiv_thm

$$\vdash \forall n \ c \bullet \text{same_at_c_below_level } n \ c \in \text{Equivalence}$$

lift_rel_same_at_c_below_level_indexed_equiv_thm

$$\vdash \forall n \bullet \text{LiftRel (same_at_c_below_level } n) \in \text{IndexedEquiv}$$

lift_rel_same_at_c_below_level_refl_thm

$$\vdash \forall n \ C \ x \bullet (x, x) \in \text{LiftRel (same_at_c_below_level } n) \ C$$

simplest_witness_label_secure_thm

$\vdash \text{simplest_witness} \in \text{label_secure}$ **identity_witness_label_secure_thm** $\vdash \text{identity_witness} \in \text{label_secure}$

6 INDEX

<i>append_assoc_thm</i>	6	<i>FlowSecureMachine</i>	18
<i>append_cancel_thm</i>	6	<i>identicalObjs</i>	15
<i>append_empty_thm</i>	6	<i>identicalObjs</i>	17
<i>Behaviours</i>	15	<i>identicalObj_consistent</i>	19
<i>Behaviours</i>	18	<i>identicalObj</i>	15
<i>BoundedObs</i>	11	<i>identicalObj</i>	16
<i>BoundedObs</i>	12	<i>identical_obj_equiv_thm</i>	21
<i>composite</i>	22	<i>identical_obj_filter_obj_thm1</i>	20
<i>composite</i>	24	<i>identical_obj_filter_obj_thm2</i>	20
<i>down_∩_comp_up_thm</i>	14	<i>identical_obj_filter_obj_thm</i>	20
<i>elems_append_thm</i>	6	<i>identical_obj_refl_thm</i>	21
<i>elems_combine_elems_thm</i>	20	<i>identical_obj_rft_thm</i>	21
<i>elems_combine_map_thm1</i>	20	<i>identical_obj_sym_thm</i>	21
<i>elems_combine_map_thm</i>	20	<i>identical_obj_trans_thm</i>	21
<i>elems_combine_swap_thm</i>	20	<i>identity_witness_label_secure_thm</i>	28
<i>elems_combine_thm</i>	20	<i>identity_witness_thm</i>	27
<i>elems_map_thm</i>	6	<i>identity_witness</i>	23
<i>Equivalence</i>	11	<i>identity_witness</i>	26
<i>Equivalence</i>	12	<i>Independent</i>	11
<i>equiv_anti_mono_lift_rel_thm</i>	13	<i>Independent</i>	12
<i>factor0</i>	22	<i>IndexedEquiv</i>	11
<i>factor0</i>	24	<i>IndexedEquiv</i>	12
<i>factor1</i>	22	<i>Influenced</i>	11
<i>factor1</i>	24	<i>Influenced</i>	12
<i>factor2</i>	22	<i>Init</i>	15
<i>factor2</i>	24	<i>Init</i>	17
<i>factor3</i>	22	<i>i_local_thm</i>	8
<i>factor3</i>	24	<i>k_local_thm</i>	8
<i>FactoredMachine</i>	23	<i>label_secure_to</i>	22
<i>FactoredMachine</i>	24	<i>label_secure_to</i>	25
<i>Factorisation</i>	23	<i>label_secure</i>	22
<i>factors</i>	22	<i>label_secure</i>	25
<i>factors</i>	24	<i>leaf_is_a_tree_thm</i>	9
<i>factor_level</i>	22	<i>length_0_thm</i>	6
<i>factor_level</i>	23	<i>length_append_thm</i>	6
<i>factor_out</i>	22	<i>length_length_flat_thm</i>	6
<i>factor_out</i>	24	<i>levelled_factorisation</i>	22
<i>Factor</i>	23	<i>levelled_factorisation</i>	24
<i>FilterObj_consistent</i>	20	<i>LiftRel</i>	11
<i>FilterObj</i>	15	<i>LiftRel</i>	12
<i>FilterObj</i>	18	<i>lift_machine_consistent</i>	19
<i>FinitaryRecType</i>	5	<i>lift_machine</i>	15
<i>fin_rec_type_induction_thm1</i>	7	<i>lift_machine</i>	17
<i>fin_rec_type_induction_thm</i>	7	<i>lift_rel_indexed_equiv_thm</i>	13
<i>fin_rec_type_prim_rec_exists_thm</i>	7	<i>lift_rel_same_at_c_below_level_indexed_equiv_thm</i>	27
<i>fin_rec_type_prim_rec_lemma1</i>	7	<i>lift_rel_same_at_c_below_level_refl_thm</i>	27
<i>fin_rec_type_prim_rec_thm</i>	7	<i>lift_rel_same_ins_same_outs_down_set_thm</i>	13
<i>fin_rec_type_prim_rec_unique_thm</i>	7	<i>lift_rel_same_lab_val_down_set_thm</i>	14
<i>flat_append_thm</i>	6	<i>lift_rel_same_lab_val_equiv_thm</i>	14
<i>flat_empty_thm</i>	6	<i>list_local_thm1</i>	8
<i>FlowSecureMachine</i>	15	<i>list_local_thm</i>	8

<i>LocalFunctional</i>	5	<i>purge_above_level</i>	23
<i>local_functional_thm</i>	8	<i>purge_above_level</i>	26
<i>Machine</i>	16	<i>reqClearance</i>	15
<i>Machine</i>	17	<i>reqClearance</i>	16
<i>machine</i>	22	<i>reqSsql</i>	15
<i>machine</i>	24	<i>reqSsql</i>	16
<i>map_map_id_thm</i>	6	<i>Req</i>	16
<i>MkFactoredMachine</i>	22	<i>sameFilterInputs</i>	15
<i>MkFactoredMachine</i>	24	<i>sameFilterInputs</i>	18
<i>MkFactorisation</i>	22	<i>SameLabVal</i>	11
<i>MkFactorisation</i>	24	<i>SameLabVal</i>	12
<i>MkMachine</i>	15	<i>sameOutputs</i>	15
<i>MkMachine</i>	17	<i>sameOutputs</i>	17
<i>MkObj_consistent</i>	18	<i>sameRequests</i>	15
<i>MkObj</i>	15	<i>sameRequests</i>	17
<i>MkObj</i>	16	<i>sameRequest</i>	15
<i>MkReq</i>	15	<i>sameRequest</i>	17
<i>MkReq</i>	16	<i>same_at_c_below_level_equiv_thm</i>	27
<i>MkTree_consistent</i>	9	<i>same_at_c_below_level_purge_above_level_thm1</i>	27
<i>MkTree</i>	5	<i>same_at_c_below_level_purge_above_level_thm2</i>	27
<i>MkTree</i>	6	<i>same_at_c_below_level_purge_above_level_thm</i>	27
<i>mk_tree_one_one_thm</i>	18	<i>same_at_c_below_level</i>	22
<i>mk_tree_onto_thm</i>	18	<i>same_at_c_below_level</i>	25
<i>ML_BEHAVIOUR</i>	16	<i>same_ins_anti_mono_thm</i>	13
<i>ml_secure</i>	11	<i>same_ins_equiv_thm</i>	13
<i>ml_secure</i>	12	<i>same_ins_same_outs_indexed_equiv_thm</i>	13
<i>Next</i>	15	<i>same_lab_val_anti_mono_thm</i>	14
<i>Next</i>	17	<i>same_lab_val_equiv_thm</i>	13
<i>not_lattice_top_thm</i>	13	<i>same_outputs_equiv_thm</i>	21
<i>objectClass_consistent</i>	18	<i>same_outputs_indexed_equiv_thm</i>	21
<i>objectClass</i>	15	<i>same_outs_anti_mono_thm</i>	13
<i>objectClass</i>	16	<i>same_outs_equiv_thm</i>	13
<i>objectContains_consistent</i>	19	<i>same_requests_equiv_thm</i>	21
<i>objectContains</i>	15	<i>same_requests_filter_obj_thm</i>	21
<i>objectContains</i>	16	<i>same_requests_indexed_equiv_thm</i>	21
<i>objectRefers_consistent</i>	19	<i>same_to_level</i>	22
<i>objectRefers</i>	15	<i>same_to_level</i>	23
<i>objectRefers</i>	16	<i>simplest_witness_label_secure_thm</i>	27
<i>object_clauses</i>	19	<i>simplest_witness</i>	22
<i>obj_induction_thm</i>	20	<i>simplest_witness</i>	25
<i>obj_prim_rec_thm</i>	19	<i>simple_witness_thm</i>	27
<i>Obj</i>	16	<i>special_machine</i>	22
<i>OBSERVATION</i>	11	<i>special_machine</i>	24
<i>ObservedValue</i>	11	<i>sum_local_thm</i>	8
<i>ObservedValue</i>	12	<i>SWORD_construction</i>	15
<i>one_one_left_inv_thm</i>	7	<i>SWORD_construction</i>	18
<i>one_one_onto_inv_thm</i>	7	<i>SWORD_ml_secure</i>	15
<i>onto_right_inv_thm</i>	7	<i>SWORD_ml_secure</i>	17
<i>Output</i>	15	<i>Text</i>	16
<i>Output</i>	17	<i>thm_040_1</i>	13
<i>ofst_local_thm</i>	8	<i>thm_040_2</i>	14
<i>o_snd_local_thm</i>	8	<i>tree_cases_lemma1</i>	9
<i>purge_above_level_consistent</i>	26	<i>tree_cases_lemma</i>	9
<i>purge_above_level_thm</i>	26	<i>tree_def</i>	6

<i>tree_fin_rec_thm</i>	9
<i>tree_induction_lemma1</i>	8
<i>tree_induction_lemma2</i>	9
<i>tree_induction_lemma</i>	9
<i>tree_induction_tac_lemma</i>	9
<i>tree_induction_thm</i>	10
<i>tree_local_thm</i>	9
<i>tree_prim_rec_thm</i>	10
<i>TREE</i>	5
<i>Tree</i>	5
<i>TREE</i>	6
<i>unparse_thm1</i>	9
<i>unparse_thm</i>	9
<i>Unparse</i>	5
<i>Unparse</i>	6
<i>up_down_clauses</i>	13
<i>up_down_thm1</i>	13
<i>up_down_thm2</i>	13
<i>up_down_thm3</i>	13
<i>VisibleOutput</i>	15
<i>VisibleOutput</i>	17
<i>VisibleReq</i>	15
<i>VisibleReq</i>	17
<i>well_factored</i>	22
<i>well_factored</i>	24
<i>x_ml_secure</i>	11
<i>x_ml_secure</i>	12
\uparrow	11
\uparrow	12
\downarrow	11
\downarrow	12
$\neg_empty_list_tree_lemma$	9
<i>%f</i>	22
<i>%f</i>	23
$\hat{\ }_empty_thm$	6
$\uparrow_ \uparrow_thm$	13
\times_local_thm	8
\perp	11
\perp	12