On access control, capabilities, their equivalence and confused deputy attacks (Technical appendix)

Vineet Rajani
MPI-SWS
vrajani@mpi-sws.org

Deepak Garg
MPI-SWS
dg@mpi-sws.org

Tamara Rezk
INRIA
tamara.rezk@inria.fr

May 19, 2016

1 Region calculus with computable references

1.1 Syntax

Locations are drawn from the set *Loc* and values are drawn from the set *Val* and principals are drawn from the set *Prin*

$e ::=$	Expression
$\mid v$	Value
$\mid\ !e$	Dereference
$\mid e \oplus e$	reference computation

$v ::=$	Value
$\mid n$	Integer
$\mid tt$	True
$\mid ff$	False
$\mid {}^{\mathbb{R}}r$	Read view of a location
$\mid {}^{\mathbb{W}}r$	Write view of a location

$c ::=$	Command
$\mid \text{if } e \text{ then } c \text{ else } c$	Conditional
$\mid \text{while } e \text{ do } c$	Loop
$\mid e := e$	Assignment
$\mid c; c$	Sequential composition
$\mid skip$	Skip

$P ::=$	Program
$\mid \rho\{c\}$	Region
$\mid P \circ P$	Region composition

$\rho ::=$	Principal
$\mid \mathbb{P}$	Normal principal
$\mid \overline{\mathbb{P}}$	Endorsed principal

1.2 Semantics

1.2.1 Access control semantics

Expressions:

$$\text{A-val}\ \frac{}{\langle H, v\rangle \Downarrow_A^{\rho} v} \qquad \text{A-deref}\ \frac{\langle H, e\rangle \Downarrow_A^{\rho}\ {}^{\mathbb{R}}r \qquad v = H(r)}{\langle H, !e\rangle \Downarrow_A^{\rho} v}$$

$$\text{A-refComp}\ \frac{\langle H, e_1\rangle \Downarrow_A^{\rho}\ {}^{\nu}r \qquad \langle H, e_2\rangle \Downarrow_A^{\rho} n \qquad (r \oplus n) \in dom(H)}{\langle H, e_1 \oplus e_2\rangle \Downarrow_A^{\rho}\ {}^{\nu}(r \oplus n)}$$

Commands:

$$\text{A-if}\ \frac{\langle H, e\rangle \Downarrow_A^{\rho} v \qquad v = tt}{\langle H, \text{if } e \text{ then } c_1 \text{ else } c_2\rangle \xrightarrow{\rho}_A \langle H, c_1\rangle}$$

$$\text{A-else}\ \frac{\langle H, e\rangle \Downarrow_A^{\rho} v \qquad v = ff}{\langle H, \text{if } e \text{ then } c_1 \text{ else } c_2\rangle \xrightarrow{\rho}_A \langle H, c_2\rangle}$$

$$\text{A-while 1}\ \frac{\langle H, e\rangle \Downarrow_A^{\rho} v \qquad v = tt}{\langle H, \text{while } e \text{ do } c\rangle \xrightarrow{\rho}_A \langle H, c; \text{while } e \text{ do } c\rangle}$$

$$\text{A-while 2}\ \frac{\langle H, e\rangle \Downarrow_A^{\rho} v \qquad v = ff}{\langle H, \text{while } e \text{ do } c\rangle \xrightarrow{\rho}_A \langle H, skip\rangle}$$

$$\text{A-assign}\ \frac{\langle H, e_1\rangle \Downarrow_A^{\rho}\ {}^{\mathbb{W}}r \qquad \langle H, e_2\rangle \Downarrow_A^{\rho} v \qquad \beta(\rho) \geq_{\mathbb{L}} \mathbb{O}(r)}{\langle H, e_1 := e_2\rangle \xrightarrow{\rho}_A \langle H[r \mapsto v], skip\rangle}$$

$$\text{A-seq 1}\ \frac{\langle H, c_1\rangle \xrightarrow{\rho}_A \langle H', c_1'\rangle}{\langle H, c_1; c_2\rangle \xrightarrow{\rho}_A \langle H', c_1'; c_2\rangle}$$

$$\text{A-seq 2}\ \frac{}{\langle H, skip; c_2\rangle \xrightarrow{\rho}_A \langle H, c_2\rangle}$$

Program:

$$\text{A-prg 1}\ \frac{\langle H, c\rangle \xrightarrow{\varrho}_A \langle H', c'\rangle}{\langle H, \rho\{c\}\rangle \rightarrow_A \langle H, \rho\{c'\}\rangle}$$

$$\text{A-comp 1}\ \frac{\langle H, P_1\rangle \rightarrow_A \langle H', P_1'\rangle}{\langle H, P_1 \circ P_2\rangle \rightarrow_A \langle H', P_1' \circ P_2\rangle}$$

$$\text{A-comp 2}\ \frac{}{\langle H, \rho\{\mathit{skip}\} \circ P\rangle \rightarrow_A \langle H, P\rangle}$$

1.2.2 Capability semantics

Expressions:

$$\text{C-val}\ \frac{v = {}^{\mathbb{W}}r' \implies \beta(\rho) \geq_{\mathbb{L}} \mathbb{O}(r')}{\langle H, v\rangle \ \overset{\rho}{\Downarrow}_C\ v} \qquad \text{C-deref}\ \frac{\begin{array}{c}\langle H, e\rangle \overset{\rho}{\Downarrow}_C\ {}^{\mathbb{R}}r \qquad v = H(r) \\ v = {}^{\mathbb{W}}r' \implies \beta(\rho) \geq_{\mathbb{L}} \mathbb{O}(r')\end{array}}{\langle H, !e\rangle \overset{\rho}{\Downarrow}_C\ v}$$

$$\text{C-refComp}\ \frac{\begin{array}{c}\langle H, e_1\rangle \overset{\rho}{\Downarrow}_C\ {}^{\nu}r \qquad \langle H, e_2\rangle \overset{\rho}{\Downarrow}_C\ n \\ (r \oplus n) \in \mathit{dom}(H) \qquad \nu = \mathbb{W} \implies \beta(\rho) \geq_{\mathbb{L}} \mathbb{O}(r \oplus n)\end{array}}{\langle H, e_1 \oplus e_2\rangle \overset{\rho}{\Downarrow}_C\ {}^{\nu}(r \oplus n)}$$

Commands:

$$\text{C-if}\ \frac{\langle H, e\rangle \overset{\rho}{\Downarrow}_C\ v \qquad v = \mathit{tt}}{\langle H, \text{if } e \text{ then } c_1 \text{ else } c_2\rangle \xrightarrow{\varrho}_C \langle H, c_1\rangle}$$

$$\text{C-else}\ \frac{\langle H, e\rangle \overset{\rho}{\Downarrow}_C\ v \qquad v = \mathit{ff}}{\langle H, \text{if } e \text{ then } c_1 \text{ else } c_2\rangle \xrightarrow{\varrho}_C \langle H, c_2\rangle}$$

$$\text{C-while 1}\ \frac{\langle H, e\rangle \overset{\rho}{\Downarrow}_C\ v \qquad v = \mathit{tt}}{\langle H, \text{while } e \text{ do } c\rangle \xrightarrow{\varrho}_C \langle H, c; \text{while } e \text{ do } c\rangle}$$

$$\text{C-while 2}\ \frac{\langle H, e\rangle \overset{\rho}{\Downarrow}_C\ v \qquad v = \mathit{ff}}{\langle H, \text{while } e \text{ do } c\rangle \xrightarrow{\varrho}_C \langle H, \mathit{skip}\rangle}$$

$$\text{C-assign}\ \frac{\langle H, e_1\rangle \Downarrow_C^{\rho}\ {}^{\mathbb{W}}r \qquad \langle H, e_2\rangle \Downarrow_C^{\rho}\ v}{\langle H, e_1 := e_2\rangle \xrightarrow{\rho}_C \langle H[r \mapsto v], \mathit{skip}\rangle}$$

$$\text{C-seq 1}\ \frac{\langle H, c_1\rangle \xrightarrow{\rho}_C \langle H', c_1'\rangle}{\langle H, c_1; c_2\rangle \xrightarrow{\rho}_C \langle H', c_1'; c_2\rangle}$$

$$\text{C-seq 2}\ \frac{}{\langle H, \mathit{skip}; c_2\rangle \xrightarrow{\rho}_C \langle H, c_2\rangle}$$

Program:

$$\text{C-prg 1}\ \frac{\langle H, c\rangle \xrightarrow{\rho}_C \langle H', c'\rangle}{\langle H, \rho\{c\}\rangle \rightarrow_C \langle H, \rho\{c'\}\rangle}$$

$$\text{C-comp 1}\ \frac{\langle H, P_1\rangle \rightarrow_C \langle H', P_1'\rangle}{\langle H, P_1 \circ P_2\rangle \rightarrow_C \langle H', P_1' \circ P_2\rangle}$$

$$\text{C-comp 2}\ \frac{}{\langle H, \rho\{\mathit{skip}\} \circ P\rangle \rightarrow_C \langle H, P\rangle}$$

Lemma 1 ($\Downarrow_C^{\rho} \implies \Downarrow_A^{\rho}$)**.** $\forall H, e, \rho.\ \langle H, e\rangle \Downarrow_C^{\rho}\ v \implies \langle H, e\rangle \Downarrow_A^{\rho}\ v$

Proof. Proof by induction on the $\Downarrow_C^{\rho}$

1. C-val:
   From A-val
2. C-deref:
   IH: $\langle H, e\rangle \Downarrow_A^{\rho}\ {}^{\mathbb{R}}r$
   From IH and A-deref.
3. C-refComp:
   From A-refComp

□

Lemma 2 ($\Downarrow_C^{\rho}$ cant evaluate to references higher than ρ)**.** $\forall H, e, \rho.\ \langle H, e\rangle \Downarrow_C^{\rho}\ v \wedge v = {}^{\mathbb{W}}r \implies \beta(\rho) \geq_{\mathbb{L}} \mathbb{O}(r)$

Proof. Proof by induction on the $\Downarrow_C^{\rho}$

1. C-val:
   Directly from the premise

2. C-deref:
   Directly from the premise

3. C-refComp:
   Directly from the premise

$\square$

Lemma 3 ($\xrightarrow{\rho}_C \implies \xrightarrow{\rho}_A$)**.** $\forall H, \rho, c.\ \ \langle H, c\rangle \xrightarrow{\rho}_C \langle H', c'\rangle \implies \langle H, c\rangle \xrightarrow{\rho}_A \langle H', c'\rangle$

Proof. Proof by induction on $\xrightarrow{\rho}_C$

1. C-if:
   From Lemma 1 we know that $\langle H, e\rangle \Downarrow^{\rho}_A v$ and $v = tt$
   Therefore, from A-if.

2. C-else:
   From Lemma 1 we know that $\langle H, e\rangle \Downarrow^{\rho}_A v$ and $v = \mathit{ff}$
   Therefore, from A-else.

3. C-while 1: From Lemma 1 we know that $\langle H, e\rangle \Downarrow^{\rho}_A v$ and $v = tt$
   Therefore, from A-while 1.

4. C-while 2: From Lemma 1 we know that $\langle H, e\rangle \Downarrow^{\rho}_A v$ and $v = \mathit{ff}$
   Therefore, from A-while 2.

5. C-assign:
   From Lemma 1 we know that $\langle H, e_1\rangle \Downarrow^{\rho}_A {}^{\mathbb{W}}r$ and $\langle H, e_2\rangle \Downarrow^{\rho}_A v$.
   From Lemma 2 we know that $\beta(\rho) \geq_{\mathbb{L}} \mathbb{O}(r)$
   Therefore, from A-assign

6. C-seq 1:
   IH: $\langle H, c_1\rangle \xrightarrow{\rho}_A \langle H', c_1'\rangle$
   Therefore, from A-seq 1.

7. C-seq 2:
   Therefore, from A-seq 2.

$\square$

Theorem 1 ($\rightarrow_C \implies \rightarrow_A$)**.** $\forall H, P.\ \ \langle H, P\rangle \rightarrow_C \langle H', P'\rangle \implies \langle H, P\rangle \rightarrow_A \langle H', P'\rangle$

Proof. Proof by induction on the $\rightarrow_C$

1. C-prg 1:
   From Lemma 3 and A-prg 1

2. C-comp 1:
   From A-comp 1

3. C-comp 2:
   From A-comp 1

$\square$

Lemma 4 (ACs: Write integrity for commands)**.** *If* $\langle H, c\rangle \xrightarrow{\rho}_A^* \langle H', \_\rangle$ *and* $H(r) \neq H'(r)$*, then* $\beta(\rho) \geq_{\mathbb{L}} \mathbb{O}(r)$*.*

Proof. Say reduction happens like this, $\langle H, \rho\{c\}\rangle \rightarrow_A \langle H_1, \_\rangle \ldots \rightarrow_A \langle H_n, \_\rangle \rightarrow_A \langle H', \_\rangle$
IH: $\langle H, c\rangle \xrightarrow{\rho}_A^* \langle H_n, \_\rangle$ and $H(r) \neq H_n(r) \implies \beta(\rho) \geq_{\mathbb{L}} \mathbb{O}(r)$
By induction on the last reduction

1. A-if, A-else, A-while 1, A-while 2, A-seq 2:
   $H_n = H'$.

2. A-assign:
   Directly from the premise we know that $\beta(\rho) \geq_{\mathbb{L}} \mathbb{O}(r)$

$\square$

Theorem 2 (ACs have write integrity)**.** *If* $\langle H, \rho\{c\}\rangle \rightarrow_A^* \langle H', \_\rangle$ *and* $H(r) \neq H'(r)$*, then* $\beta(\rho) \geq_{\mathbb{L}} \mathbb{O}(r)$*.*

Proof. From Lemma 4 $\square$

Theorem 3 (Cs have write integrity)**.** *If* $\langle H, \rho\{c\}\rangle \rightarrow_C^* \langle H', \_\rangle$ *and* $H(r) \neq H'(r)$*, then* $\beta(\rho) \geq_{\mathbb{L}} \mathbb{O}(r)$*.*

Proof. From theorem 2 and 1 $\square$

Lemma 5. $\forall H, e, \rho.$
$\langle H, e\rangle \overset{\rho}{\Downarrow}_C v \wedge v =^{\mathbb{W}} r' \implies \beta(\rho) \geq_{\mathbb{L}} \mathbb{O}(r')$

Proof. Proof by induction on $\overset{\rho}{\Downarrow}_C$:

1. C-val:
   Given

2. C-deref:
   The check $\beta(\rho) \geq_{\mathbb{L}} \mathbb{O}(r')$ takes care of it.

3. C-refComp:
   The check $\beta(\rho) \geq_{\mathbb{L}} \mathbb{O}(r')$ takes care of it.

$\square$

Theorem 4 (No illicit expansion of authority)**.** *If* $\langle H, \rho\{c\}\rangle \to_C{}^* \langle H', \rho\{c'\}\rangle$ *and* $H'(r') = {}^{\mathbb{W}}r$*, then either* $H(r') = {}^{\mathbb{W}}r$ *or* $\beta(\rho) \geq_{\mathbb{L}} \mathbb{O}(r)$*.*

Proof. Say the reduction happens as follows:
$\langle H, \rho\{c\}\rangle \to_C \langle H_1, \rho\{c_1\}\rangle \ldots \langle H_n, \rho\{c_n\}\rangle \to_C \langle H', \rho\{c'\}\rangle$
Induction on the reduction sequence:
IH1: $H_n(r') = {}^{\mathbb{W}}r \implies H_n(r') = {}^{\mathbb{W}}r$ or $\beta(\rho) \geq_{\mathbb{L}} \mathbb{O}(r)$.
Induction on the last derivation:

1. C-if, C-else, C-while 1, C-while 2, C-seq 2:
   $H' = H_n$, therefore from IH1

2. C-assign:
   From Lemma 5 we know that $\beta(\rho) \geq_{\mathbb{L}} \mathbb{O}(r)$. 2 cases arise

   - $v = {}^{\mathbb{W}}r''$:
     From Lemma 5 $\beta(\rho) \geq_{\mathbb{L}} \mathbb{O}(r'')$. Thus satisfying 2nd disjunct.

   - $v \neq {}^{\mathbb{W}}r''$:
     Vacuous

3. C-seq 1:
   From IH

□

A and C semantics don't provide any CDA freedom (see counter examples in the paper). But one can use C semantics to obtain CDA freedom on a language without reference computation with additional restrictions (as we will describe in the subsequent sections). Those restrictions are quite strict and don't admit many useful programs. So, we relax those restrictions at the cost of doing explicit-only provenance tracking (for the same restricted language). And finally we show that at the cost doing full blown provenance tracking one can remove all those restrictions (this can even be done for the full language with reference computation).

2 Region calculus without computable references

2.1 Syntax

Locations are drawn from the set *Loc* and values are drawn from the set *Val* and principals are drawn from the set *Prin*

$$
\begin{array}{lr}
e ::= & \text{Expression} \\
\quad \mid v & \text{Value} \\
\quad \mid \,!e & \text{Dereference}
\end{array}
$$

$$
\begin{array}{lr}
v ::= & \text{Value} \\
\quad \mid n & \text{Integer} \\
\quad \mid tt & \text{True} \\
\quad \mid \mathit{ff} & \text{False} \\
\quad \mid {}^{\mathbb{R}}r & \text{Read view of a location} \\
\quad \mid {}^{\mathbb{W}}r & \text{Write view of a location}
\end{array}
$$

$$
\begin{array}{lr}
c ::= & \text{Command} \\
\quad \mid \text{if } e \text{ then } c \text{ else } c & \text{Conditional} \\
\quad \mid \text{while } e \text{ do } c & \text{Loop} \\
\quad \mid e := e & \text{Assignment} \\
\quad \mid c; c & \text{Sequential composition} \\
\quad \mid \mathit{skip} & \text{Skip}
\end{array}
$$

$$
\begin{array}{lr}
P ::= & \text{Program} \\
\quad \mid \rho\{c\} & \text{Region} \\
\quad \mid P \circ P & \text{Region composition}
\end{array}
$$

$$
\begin{array}{lr}
\rho ::= & \text{Principal} \\
\quad \mid \mathbb{P} & \text{Normal principal} \\
\quad \mid \overline{\mathbb{P}} & \text{Endorsed principal}
\end{array}
$$

2.2 Semantics

Definition 1 (Heap)**.** *Heap (H) is defined as a mapping from location to value, formally: $H : Loc \rightarrow Val$*

Definition 2 (Ownership map)**.** *Ownership map ($\mathbb{O}$) is a mapping from location to the principal owning it, formally: $\mathbb{O} : Loc \rightarrow Prin$.*

Definition 3 (Get principal of a region)**.**

$$
\beta(\rho) \triangleq \begin{cases} \mathbb{P} & \rho = \mathbb{P} \\ \mathbb{P} & \rho = \overline{\mathbb{P}} \end{cases}
$$

2.2.1 Access control semantics

Expressions:

$$\text{A-val}\ \frac{}{\langle H, v\rangle \Downarrow^{\rho}_{A} v} \qquad \text{A-deref}\ \frac{\langle H, e\rangle \Downarrow^{\rho}_{A}{}^{\mathbb{R}} r \qquad v = H(r)}{\langle H, !e\rangle \Downarrow^{\rho}_{A} v}$$

Commands:

$$\text{A-if}\ \frac{\langle H, e\rangle \Downarrow^{\rho}_{A} v \qquad v = tt}{\langle H, \text{if } e \text{ then } c_1 \text{ else } c_2\rangle \xrightarrow{\rho}_{A} \langle H, c_1\rangle}$$

$$\text{A-else}\ \frac{\langle H, e\rangle \Downarrow^{\rho}_{A} v \qquad v = \mathit{ff}}{\langle H, \text{if } e \text{ then } c_1 \text{ else } c_2\rangle \xrightarrow{\rho}_{A} \langle H, c_2\rangle}$$

$$\text{A-while 1}\ \frac{\langle H, e\rangle \Downarrow^{\rho}_{A} v \qquad v = tt}{\langle H, \text{while } e \text{ do } c\rangle \xrightarrow{\rho}_{A} \langle H, c; \text{while } e \text{ do } c\rangle}$$

$$\text{A-while 2}\ \frac{\langle H, e\rangle \Downarrow^{\rho}_{A} v \qquad v = \mathit{ff}}{\langle H, \text{while } e \text{ do } c\rangle \xrightarrow{\rho}_{A} \langle H, \mathit{skip}\rangle}$$

$$\text{A-assign}\ \frac{\langle H, e_1\rangle \Downarrow^{\rho}_{A}{}^{\mathbb{W}} r \qquad \langle H, e_2\rangle \Downarrow^{\rho}_{A} v \\ \beta(\rho) \geq_{\mathbb{L}} \mathbb{O}(r)}{\langle H, e_1 := e_2\rangle \xrightarrow{\rho}_{A} \langle H[r \mapsto v], \mathit{skip}\rangle}$$

$$\text{A-seq 1}\ \frac{\langle H, c_1\rangle \xrightarrow{\rho}_{A} \langle H', c_1'\rangle}{\langle H, c_1; c_2\rangle \xrightarrow{\rho}_{A} \langle H', c_1'; c_2\rangle}$$

$$\text{A-seq 2}\ \frac{}{\langle H, \mathit{skip}; c_2\rangle \xrightarrow{\rho}_{A} \langle H, c_2\rangle}$$

Program:

$$\text{A-prg 1}\ \frac{\langle H, c\rangle \xrightarrow{\rho}_{A} \langle H', c'\rangle}{\langle H, \rho\{c\}\rangle \to_{A} \langle H, \rho\{c'\}\rangle}$$

$$\text{A-comp 1}\ \frac{\langle H, P_1\rangle \to_{A} \langle H', P_1'\rangle}{\langle H, P_1 \circ P_2\rangle \to_{A} \langle H', P_1' \circ P_2\rangle}$$

$$\text{A-comp 2}\ \frac{}{\langle H, \rho\{\mathit{skip}\} \circ P\rangle \to_{A} \langle H, P\rangle}$$

2.2.2 Capability semantics

Expressions:

$$\text{C-val}\ \frac{v = {}^{\mathbb{W}}r' \implies \beta(\rho) \geq_{\mathbb{L}} \mathbb{O}(r')}{\langle H, v\rangle \ \overset{\rho}{\Downarrow}_C \ v} \qquad \text{C-deref}\ \frac{\langle H, e\rangle \overset{\rho}{\Downarrow}_C \ {}^{\mathbb{R}}r \qquad v = H(r) \qquad v = {}^{\mathbb{W}}r' \implies \beta(\rho) \geq_{\mathbb{L}} \mathbb{O}(r')}{\langle H, !e\rangle \overset{\rho}{\Downarrow}_C \ v}$$

Commands:

$$\text{C-if}\ \frac{\langle H, e\rangle \overset{\rho}{\Downarrow}_C \ v \qquad v = tt}{\langle H, \text{if } e \text{ then } c_1 \text{ else } c_2\rangle \xrightarrow{\rho}_C \langle H, c_1\rangle}$$

$$\text{C-else}\ \frac{\langle H, e\rangle \overset{\rho}{\Downarrow}_C \ v \qquad v = \mathit{ff}}{\langle H, \text{if } e \text{ then } c_1 \text{ else } c_2\rangle \xrightarrow{\rho}_C \langle H, c_2\rangle}$$

$$\text{C-while 1}\ \frac{\langle H, e\rangle \overset{\rho}{\Downarrow}_C \ v \qquad v = tt}{\langle H, \text{while } e \text{ do } c\rangle \xrightarrow{\rho}_C \langle H, c; \text{while } e \text{ do } c\rangle}$$

$$\text{C-while 2}\ \frac{\langle H, e\rangle \overset{\rho}{\Downarrow}_C \ v \qquad v = \mathit{ff}}{\langle H, \text{while } e \text{ do } c\rangle \xrightarrow{\rho}_C \langle H, \mathit{skip}\rangle}$$

$$\text{C-assign}\ \frac{\langle H, e_1\rangle \overset{\rho}{\Downarrow}_C \ {}^{\mathbb{W}}r \qquad \langle H, e_2\rangle \overset{\rho}{\Downarrow}_C \ v}{\langle H, e_1 := e_2\rangle \xrightarrow{\rho}_C \langle H[r \mapsto v], \mathit{skip}\rangle}$$

$$\text{C-seq 1}\ \frac{\langle H, c_1\rangle \xrightarrow{\rho}_C \langle H', c_1'\rangle}{\langle H, c_1; c_2\rangle \xrightarrow{\rho}_C \langle H', c_1'; c_2\rangle}$$

$$\text{C-seq 2}\ \frac{}{\langle H, \mathit{skip}; c_2\rangle \xrightarrow{\rho}_C \langle H, c_2\rangle}$$

Program:

$$\text{C-prg 1}\ \frac{\langle H, c\rangle \xrightarrow{\rho}_C \langle H', c'\rangle}{\langle H, \rho\{c\}\rangle \rightarrow_C \langle H, \rho\{c'\}\rangle}$$

$$\text{C-comp 1}\ \frac{\langle H, P_1\rangle \rightarrow_C \langle H', P_1'\rangle}{\langle H, P_1 \circ P_2\rangle \rightarrow_C \langle H', P_1' \circ P_2\rangle}$$

$$\text{C-comp 2}\ \frac{}{\langle H, \rho\{\mathit{skip}\} \circ P\rangle \rightarrow_C \langle H, P\rangle}$$

Lemma 6 **($\overset{\rho}{\Downarrow}_C \Longrightarrow \overset{\rho}{\Downarrow}_A$).** $\forall H, e, \rho.\ \langle H, e\rangle \overset{\rho}{\Downarrow}_C v \implies \langle H, e\rangle \overset{\rho}{\Downarrow}_A v$

Proof. Proof by induction on the $\overset{\rho}{\Downarrow}_C$

1. C-val:
   From A-val

2. C-deref:
   IH: $\langle H, e\rangle \overset{\rho}{\Downarrow}_A {}^{\mathbb{R}}r$
   From IH and A-deref.

□

Lemma 7 **($\overset{\rho}{\Downarrow}_C$ cant evaluate to references higher than ρ).** $\forall H, e, \rho.\ \langle H, e\rangle \overset{\rho}{\Downarrow}_C v$
$\wedge\ v = {}^{\mathbb{W}}r \implies$
$\mathbb{P} \geq_{\mathbb{L}} \mathbb{O}(r)$ *where* $\mathbb{P} = \beta(\rho)$

Proof. Proof by induction on the $\overset{\rho}{\Downarrow}_C$

1. C-val:
   Directly from the premise

2. C-deref:
   Directly from the premise

□

Lemma 8 **($\xrightarrow{\rho}_C \Longrightarrow \xrightarrow{\rho}_A$).** $\forall H, c, \rho.\ \langle H, c\rangle \xrightarrow{\rho}_C \langle H', c'\rangle \implies \langle H, c\rangle \xrightarrow{\rho}_A$
$\langle H', c'\rangle$

Proof. Proof by induction on $\xrightarrow{\rho}_C$

1. C-if:
   From Lemma 6 we know that $\langle H, e\rangle \overset{\rho}{\Downarrow}_A v$ and $v = \mathit{tt}$
   Therefore, from A-if.

2. C-else:
   From Lemma 6 we know that $\langle H, e\rangle \overset{\rho}{\Downarrow}_A v$ and $v = \mathit{ff}$
   Therefore, from A-else.

3. C-while 1: From Lemma 6 we know that $\langle H, e\rangle \Downarrow_A^{\rho} v$ and $v = tt$
   Therefore, from A-while 1.

4. C-while 2: From Lemma 6 we know that $\langle H, e\rangle \Downarrow_A^{\rho} v$ and $v = \mathit{ff}$
   Therefore, from A-while 2.

5. C-assign:

   From Lemma 6 we know that $\langle H, e_1\rangle \Downarrow_A^{\rho} {}^{\mathbb{W}}r$ and $\langle H, e_2\rangle \Downarrow_A^{\rho} v$.
   From Lemma 7 we know that $\mathbb{P} \geq_{\mathbb{L}} \mathbb{O}(r)$ where $\mathbb{P} = \beta(\rho)$
   Therefore, from A-assign

6. C-seq 1:
   IH: $\langle H, c_1\rangle \xrightarrow{\rho}_A \langle H', c_1'\rangle$
   Therefore, from A-seq 1.

7. C-seq 2:
   Therefore, from A-seq 2.

□

Theorem 5 ($\rightarrow_C \implies \rightarrow_A$)**.** *$\forall H, P.\ \langle H, P\rangle \rightarrow_C \langle H', P'\rangle \implies \langle H, P\rangle \rightarrow_A \langle H', P'\rangle$*

Proof. Proof by induction on the $\rightarrow_C$

1. C-prg 1:
   From Lemma 8 and A-prg 1

2. C-comp 1:
   From A-comp 1

3. C-comp 2:
   From A-comp 2

□

Lemma 9 (ACs: Write integrity for commands)**.** *If $\langle H, c\rangle \xrightarrow{\rho}_A^{*} \langle H', \_\rangle$ and $H(r) \neq H'(r)$, then $\beta(\rho) \geq_{\mathbb{L}} \mathbb{O}(r)$.*

Proof. Say reduction happens like this, $\langle H, \rho\{c\}\rangle \rightarrow_A \langle H_1, \_\rangle \ldots \rightarrow_A \langle H_n, \_\rangle \rightarrow_A \langle H', \_\rangle$
IH: $\langle H, c\rangle \xrightarrow{\rho}_A^{*} \langle H_n, \_\rangle$ and $H(r) \neq H_n(r) \implies \beta(\rho) \geq_{\mathbb{L}} \mathbb{O}(r)$
By induction on the last reduction

1. A-if, A-else, A-while 1, A-while 2, A-seq 2:
   $H_n = H'$.

2. A-assign:
   Directly from the premise we know that $\beta(\rho) \geq_{\mathbb{L}} \mathbb{O}(r)$

□

Theorem 6 (ACs have write integrity)**.** *If* $\langle H, \rho\{c\}\rangle {\to_A}^* \langle H', \_\rangle$ *and* $H(r) \neq H'(r)$*, then* $\beta(\rho) \geq_{\mathbb{L}} \mathbb{O}(r)$*.*

Proof. From Lemma 9 □

Theorem 7 (Cs have write integrity)**.** *If* $\langle H, \rho\{c\}\rangle {\to_C}^* \langle H', \_\rangle$ *and* $H(r) \neq H'(r)$*, then* $\beta(\rho) \geq_{\mathbb{L}} \mathbb{O}(r)$*.*

Proof. From theorem 6 and 5 □

Lemma 10. $\forall H, e, \rho.$
$\langle H, e\rangle \overset{\rho}{\Downarrow}_C v \wedge v = {}^{\mathbb{W}}r' \implies \beta(\rho) \geq_{\mathbb{L}} \mathbb{O}(r')$

Proof. Proof by induction on $\overset{\rho}{\Downarrow}_C$:

1. C-val:
   Given

2. C-deref:
   The check $\beta(\rho) \geq_{\mathbb{L}} \mathbb{O}(r')$ takes care of it.

□

Theorem 8 (No illicit expansion of authority)**.** *If* $\langle H, \rho\{c\}\rangle {\to_C}^* \langle H', \rho\{c'\}\rangle$ *and* $H'(r') = {}^{\mathbb{W}}r$*, then either* $H(r') = {}^{\mathbb{W}}r$ *or* $\beta(\rho) \geq_{\mathbb{L}} \mathbb{O}(r)$*.*

Proof. Say the reduction happens as follows:
$\langle H, \rho\{c\}\rangle \to_C \langle H_1, \rho\{c_1\}\rangle \dots \langle H_n, \rho\{c_n\}\rangle \to_C \langle H', \rho\{c'\}\rangle$
Induction on the reduction sequence:
IH1: $H_n(r') = {}^{\mathbb{W}}r \implies H_n(r') = {}^{\mathbb{W}}r$ or $\beta(\rho) \geq_{\mathbb{L}} \mathbb{O}(r)$.
Induction on the last derivation:

1. C-if, C-else, C-while 1, C-while 2, C-seq 2:
   $H' = H_n$, therefore from IH1

2. C-assign:
   From Lemma 10 we know that $\beta(\rho) \geq_{\mathbb{L}} \mathbb{O}(r)$. 2 cases arise

   - $v = {}^{\mathbb{W}}r''$:
     From Lemma 10 $\beta(\rho) \geq_{\mathbb{L}} \mathbb{O}(r'')$. Thus satisfying 2nd disjunct.
   - $v \neq {}^{\mathbb{W}}r''$:
     Vacuous

3. C-seq 1:
   From IH

□

Definition 4 (Authority Context). *An authority context, $\mathbb{E}_{\rho_A}$, is a program with one hole of the form $\rho_A\{\bullet\}$. Formally, $\mathbb{E}_{\rho_A} ::= \rho_1\{c_1\} \circ \ldots \circ \rho_A\{\bullet\} \circ \ldots \circ \rho_n\{c_n\}$. We write $\mathbb{E}_{\rho_A}[c_A]$ for the program that replaces the hole $\bullet$ with the adversary's commands c_A, i.e., the program $\rho_1\{c_1\} \circ \ldots \circ \rho_A\{c_A\} \circ \ldots \circ \rho_n\{c_n\}$.*

Any program P (without a hole) can be trivially treated as an authority context $\mathbb{E}_{\rho_A} = P \circ \rho_A\{\bullet\}$.

Definition 5 (Attacker's Interest Set). *$AIS \triangleq$ the set of references that the attacker is interested in.*

Definition 6 (No Interesting High References in Program). *$nihrP(P, \rho_A) \triangleq$
Say $P = \rho_1\{c_1\} \circ \ldots \circ \rho_n\{c_n\}$,
$\forall 1 \leq i \leq n.\ \beta(\rho_A) \not\geq_{\mathbb{L}} \beta(\rho_i) \wedge \rho_i \neq \overline{\mathbb{P}} \implies nihrC(c_i, \rho_A)$*

Definition 7 (No Interesting High References in Command).

$$nihrC(c, \rho_A) \triangleq \begin{cases} nihrE(e, \rho_A) \wedge nihrC(c_1, \rho_A) \wedge nihrC(c_2, \rho_A) & c = (\mathit{if}\ e\ \mathit{then}\ c_1\ \mathit{else}\ c_2) \\ nihrE(e, \rho_A) \wedge nihrC(c', \rho_A) & c = (\mathit{while}\ e\ \mathit{do}\ c') \\ nihrE(e_1, \rho_A) \wedge nihrE(e_2, \rho_A) & c = (e_1 := e_2) \\ nihrC(c_1, \rho_A) \wedge nihrC(c_2, \rho_A) & c = (c_1; c_2) \\ true & otherwise \end{cases}$$

Definition 8 (No Interesting High References in Expression).

$$nihrE(e, \rho_A) \triangleq \begin{cases} false & e = {}^{w}r \wedge \beta(\rho_A) \not\geq_{\mathbb{L}} \mathbb{O}(r) \wedge r \in AIS \\ nihrE(e', \rho_A) & e = !e' \\ true & otherwise \end{cases}$$

Definition 9 (No Interesting High References in Heap). *$nihrH(H, \rho_A) \triangleq$
$\forall r \in dom(H).\ H(r) = {}^{w}r' \implies \beta(\rho_A) \not\geq_{\mathbb{L}} \mathbb{O}(r') \implies r' \notin AIS$*

Lemma 11 (High non-endorsed expressions cannot compute high reference from AIS). $\forall e, \rho, H, \rho_A$.
$\beta(\rho_A) \not\geq_{\mathbb{L}} \beta(\rho) \wedge \rho \neq \overline{\mathbb{P}} \wedge nihrE(e, \rho_A) \wedge nihrH(H, \rho_A) \wedge$
$v = {}^{-}r' \wedge \beta(\rho_A) \not\geq_{\mathbb{L}} \mathbb{O}(r') \wedge \langle H, e\rangle \overset{\rho}{\Downarrow}_C v \implies$
$r' \notin AIS$

Proof. Proof by induction on $\overset{\rho}{\Downarrow}_C$:

1. C-val:
   Since $e = v = {}^{w}r'$, $\beta(\rho_A) \not\geq_{\mathbb{L}} \mathbb{O}(r')$ and $nihrE(e, \rho_A)$
   therefore $r' \notin AIS$ (From Definition 8).

2. C-deref:
   IH: $r \notin AIS$.
   Since $\beta(\rho_A) \not\geq_{\mathbb{L}} \mathbb{O}(r')$ and $nihrH(H, \rho_A)$ therefore from Definition 9
   we know that $H(r) = {}^{w}r' \notin AIS$.

$\square$

Lemma 12 (*nihrC* and *nihrH* are invariants for high non-endorsed regions)**.**
$\forall c, \rho, H, \rho_A.$
$\beta(\rho_A) \not\geq_{\mathbb{L}} \beta(\rho) \wedge \rho \neq \overline{\mathbb{P}} \wedge$
$nihrC(c, \rho_A) \wedge nihrH(H, \rho_A) \wedge$
$\langle H, c\rangle \xrightarrow{\ell}_C \langle H', c'\rangle$
$\Longrightarrow$
$nihrC(c', \rho_A) \wedge nihrH(H', \rho_A)$

Proof. Proof by induction on $\xrightarrow{\ell}_C$

1. C-if: Since $H' = H$, therefore $nihrH(H', \rho_A)$.
   Since, $nihrC(c, \rho_A)$ therefore $nihrC(c_1, \rho_A)$.

2. C-else: Since $H' = H$, therefore $nihrH(H', \rho_A)$.
   Since, $nihrC(c, \rho_A)$ therefore $nihrC(c_2, \rho_A)$

3. C-while 1: $H' = H$, therefore $nihrH(H', \rho_A)$.
   Since, $nihrC(\text{while} e \text{do} c', \rho_A)$ therefore $nihrC(c', \rho_A)$

4. C-while 2: $H' = H$, therefore $nihrH(H', \rho_A)$.
   $nihrC(skip, \rho_A)$

5. C-assign:
   2 cases arise:

   (a) $H'(r) \neq r'$: $nihrH(H', \rho_A)$ holds vacuously.
   (b) $H'(r) = r'$: Again 2 cases arise:
      i. $\beta(\rho_A) \geq_{\mathbb{L}} \mathbb{O}(r')$: $nihrH(H', \rho_A)$ holds vacuously.
      ii. $\beta(\rho_A) \not\geq_{\mathbb{L}} \mathbb{O}(r')$: From Lemma 11 we get $r' \notin AIS$.
      And since $H'(r) = r'$, therefore $nihrH(H', \rho_A)$

   $nihrC(skip, \rho_A)$

6. C-seq 1: By IH $nihrH(H', \rho_A)$ and $nihrC(c_1', \rho_A)$. Therefore, $nihrH(H', \rho_A)$
   Since, $nihrC(c_1; c_2, \rho_A)$ and from IH, $nihrC(c_1'; c_2, \rho_A)$

7. C-seq 2: $H' = H$. Therefore, $nihrH(H', \rho_A)$.
   Since $nihrC(skip; c, \rho_A)$ therefore $nihrC(c, \rho_A)$

□

Lemma 13 (*nihrH* is invariant in low regions)**.** $\forall c, \rho, H, \rho_A.$
$\beta(\rho_A) \geq_{\mathbb{L}} \beta(\rho) \wedge$
$nihrH(H, \rho_A) \wedge$
$\langle H, c\rangle \xrightarrow{\ell}_C \langle H', c'\rangle$
$\Longrightarrow$
$nihrH(H', \rho_A)$

Proof. Proof by induction on $\xrightarrow{\ell}_C$

1. C-if, C-else, C-while 1, C-while 2, C-seq 2: Since $H' = H$, therefore $nihrH(H', \rho_A)$.

2. C-assign:
   2 cases arise:

   (a) $H'(r) \neq r'$: $nihrH(H', \rho_A)$ holds vacuously.

   (b) $H'(r) = r'$: Again 2 cases arise:

      i. $\beta(\rho_A) \geq_{\mathbb{L}} \mathbb{O}(r')$: $nihrH(H', \rho_A)$ holds vacuously.

      ii. $\beta(\rho_A) \not\geq_{\mathbb{L}} \mathbb{O}(r')$: From Lemma 14 $\beta(\rho_A) \geq_{\mathbb{L}} \mathbb{O}(r')$, therefore $nihrH(H', \rho_A)$

3. C-seq 1: By IH $nihrH(H', \rho_A)$ and $nihrC(c_1', \rho_A)$. Therefore, $nihrH(H', \rho_A)$

□

Lemma 14 (Low regions cannot compute high writable references)**.** $\forall e, \rho, H, \rho_A$.
$\beta(\rho_A) \geq_{\mathbb{L}} \beta(\rho) \wedge v = {}^w r' \wedge$
$\langle H, e \rangle \Downarrow_C^{\rho} v$
$\Longrightarrow$
$\beta(\rho_A) \geq_{\mathbb{L}} \mathbb{O}(r')$

Proof. Proof by case analysis on $\Downarrow_C^{\rho}$:

1. C-val:
   Since $v = {}^w r'$. Therefore, $\beta(\rho_A) \geq_{\mathbb{L}} \beta(\rho) \geq_{\mathbb{L}} \mathbb{O}(r')$ from C-val.

2. C-deref:
   Since $v = {}^w r'$. Therefore, $\beta(\rho_A) \geq_{\mathbb{L}} \beta(\rho) \geq_{\mathbb{L}} \mathbb{O}(r')$ from C-deref.

□

Lemma 15 ($nihrP$ and $nihrH$ are invariants on programs)**.** $\forall\ P, H, \rho_A$.
$nihrP(P, \rho_A) \wedge nihrH(H, \rho_A) \wedge$
$\langle H, P \rangle \rightarrow_C \langle H', P' \rangle$
$\Longrightarrow$
$nihrP(P', \rho_A) \wedge nihrH(H', \rho_A)$

Proof. Proof by inductipn on $\rightarrow_C$

1. C-prg 1:
   2 cases arise:

   (a) $\beta(\rho_A) \geq_{\mathbb{L}} \beta(\rho)$: Lemma 13 we know that $nihrH(H', \rho_A)$ Say, $P = \rho\{c\}$ and $P' = \rho\{c'\}$.
   Since $\beta(\rho_A) \geq_{\mathbb{L}} \beta(\rho')$ therefore $nihrP(P', \rho_A)$ holds vacuously.

   (b) $\beta(\rho_A) \not\geq_{\mathbb{L}} \beta(\rho)$: From Lemma 12 we know that $nihrH(H', \rho_A)$ and $nihrC(c', \rho_A)$.
   Say, $P = \rho\{c\}$ and $P' = \rho\{c'\}$.
   Therefore, $nihrP(P', \rho_A)$

2. C-comp 1: From IH

3. C-comp 2: $H' = H$ therefore $nihrH(H', \rho_A)$
   $nihrP(P, \rho_A)$

$\square$

Lemma 16 (High interesting references dont change value in high non-endorsed regions)**.** $\forall c, \rho, H, \rho_A.$
$\beta(\rho_A) \not\geq_{\mathbb{L}} \beta(\rho) \wedge \rho \neq \overline{\mathbb{P}} \wedge$
$nihrC(c, \rho_A) \wedge nihrH(H, \rho_A) \wedge$
$\langle H, c\rangle \xrightarrow{\ell}_C \langle H', c'\rangle$
$\Longrightarrow$
$\forall\ r \in AIS.\beta(\rho_A) \not\geq_{\mathbb{L}} \mathbb{O}(r) \implies H(r) = H'(r)$

Proof. Induction on $\xrightarrow{\ell}_C$:

1. C-if, C-else, C-while 1, C-while 2, C-seq 2:
   $H = H'$

2. C-assign:
   From Lemma 12 we know that $\beta(\rho_A) \not\geq_{\mathbb{L}} \mathbb{O}({}^w r) \implies r \notin AIS$. Therefore, $\forall\ r \in AIS.\beta(\rho_A) \not\geq_{\mathbb{L}} \mathbb{O}(r) \implies H(r) = H'(r)$

3. C-seq 1:
   By IH

$\square$

Lemma 17 (High interesting references dont change value in low regions)**.** $\forall c, \rho, H, \rho_A.$
$\beta(\rho_A) \geq_{\mathbb{L}} \beta(\rho) \wedge$
$\langle H, c\rangle \xrightarrow{\ell}_C \langle H', c'\rangle$
$\Longrightarrow$
$\forall\ r \in AIS.\beta(\rho_A) \not\geq_{\mathbb{L}} \mathbb{O}(r) \implies H(r) = H'(r)$

Proof. Induction on $\xrightarrow{\ell}_C$:

1. C-if, C-else, C-while 1, C-while 2, C-seq 2:
   $H = H'$

2. C-assign:
   From C-assign$\beta(\rho_A) \geq_{\mathbb{L}} \beta(\rho) \geq_{\mathbb{L}} \beta(\rho_w)$. Therefore, $\forall\ r \in AIS.\beta(\rho_A) \not\geq_{\mathbb{L}} \mathbb{O}(r) \implies H(r) = H'(r)$

3. C-seq 1:
   By IH

$\square$

Corollary 9. $\forall P, H_1, \rho_A.$
$nihrP(P, \rho_A) \wedge nihrH(H, \rho_A) \wedge$
$Say\ P\ = \rho_1\{c_1\} \circ \ldots \rho_n\{c_n\}$
$\forall 1 \leq i \leq n.\ \rho_i \neq \overline{\mathbb{P}} \wedge$
$\langle H_1, P_1\rangle \xrightarrow{*}_C \langle H_n, P_n\rangle$
$\Longrightarrow$
$\forall r \in AIS.\beta(\rho_A) \not\geq_{\mathbb{L}} \mathbb{O}(r) \implies H(r) = H'(r)$

Proof. Say the reduction happens as follows: $\langle H_1, P_!\rangle \rightarrow_C \langle H_2, P_2\rangle \ldots \langle H_{n-1}, P_{n-1}\rangle \rightarrow_C \langle H_n, P_n\rangle$
From Lemma 15 that $nihrP(P_{n-1}, \rho_A) \wedge nihrH(H_{n-1}, \rho_A)$
By induction on the length of the reduction
IH1: $\forall\ r \in AIS.\beta(\rho_A) \not\geq_{\mathbb{L}} \mathbb{O}(r) \implies H(r) = H_{n-1}(r)$
Induction on $\rightarrow_C$ on the last step:

1. C-prg 1:
   From IH1, Lemma 16 and Lemma 17

2. C-comp 1:
   By IH and IH1

3. C-comp 2:
   $H_{n-1} = H_n$ and from IH1.

□

Definition 10 (Freedom from Confused Deputy Attack)**.** $CDAF(\mathbb{E}_{\rho_A}, H, \rightarrow_{red}) \triangleq$
$\forall\ c_{\rho_A}, c'_{\rho_A}.$
$\langle H, \mathbb{E}_{\rho_A}[c_{\rho_A}]\rangle \rightarrow^*_{red} \langle H_1, P_1\rangle \implies$
$\forall r \in AIS.$
$\langle H, \mathbb{E}_{\rho_A}[c'_{\rho_A}]\rangle \rightarrow^*_{red} \langle H_2, P_2\rangle \implies H_1(r) = H_2(r)$
$\vee$
$\exists c''_{\rho_A}.\langle H, \rho_A[c''_{\rho_A}]\rangle \rightarrow^*_{red} \langle H_3, P_3\rangle \wedge H_1(r) = H_3(r)$

Definition 11 (CDAF with endorsement)**.** $CDAF\text{-}E(\mathbb{E}_{\rho_A}, H, \rightarrow_{red}) \triangleq$
$Say\ \mathbb{E}_{\rho_A} = \rho_1\{c_1\} \circ \ldots \rho_n\{c_n\}$
$1 \leq i, j \leq n.\ P_{ij}\ = \rho_i\{c_j\} \circ \ldots \rho_j\{c_j\}\ s.t$
$\forall\ i \leq k \leq j.P_k \neq \overline{\mathbb{P}} \implies CDAF(P_{ij}, H, \rightarrow_{red})$

Theorem 10 (CDA freedom with endorsement)**.** $\forall\ \mathbb{E}_{\rho_A}, H.$
$nihrP(\mathbb{E}_{\rho_A}, \rho_A) \implies$
$nihrH(H, \rho_A) \implies$
$CDAF\text{-}E(\mathbb{E}_{\rho_A}, H, \rightarrow_C)$

Proof. $\forall r \in AIS$ the following cases arise:

1. $\beta(\rho_A) \geq_{\mathbb{L}} \mathbb{O}(r)$: All these references can be written directly by the attacker. So, for every write on a such a reference can be simulated by the attacker code only. Thus, satisfying the second disjunct.

2. $\beta(\rho_A) \not\geq_{\mathbb{L}} \mathbb{O}(r)$: From Corollary 9, $H_1(r) = H_2(r) = H(r)$.

$\square$

3 Explicit-only provenance tracking

Expressions:

$$\text{EP-val}\frac{}{\langle H, v\rangle \overset{\rho}{\Downarrow}_{EP} v^{\top}} \qquad \text{EP-deref}\frac{\langle H, e\rangle \overset{\rho}{\Downarrow}_{A} {}^{\mathbb{R}}r^{\ell_r} \qquad \mathbb{P}_r = \mathbb{O}(r) \qquad v = H(r)}{\langle H, !e\rangle \overset{\rho}{\Downarrow}_{EP} v^{\ell_r \sqcap \mathbb{P}_r}}$$

Commands:

$$\text{EP-if}\ \frac{\langle H, e\rangle \overset{\rho}{\Downarrow}_{EP} v \qquad v = tt}{\langle H, \text{if } e \text{ then } c_1 \text{ else } c_2\rangle \xrightarrow{\rho}_{EP} \langle H, c_1\rangle}$$

$$\text{EP-else}\ \frac{\langle H, e\rangle \overset{\rho}{\Downarrow}_{EP} v \qquad v = f\!f}{\langle H, \text{if } e \text{ then } c_1 \text{ else } c_2\rangle \xrightarrow{\rho}_{EP} \langle H, c_2\rangle}$$

$$\text{EP-while 1}\frac{\langle H, e\rangle \overset{\rho}{\Downarrow}_{EP} v \qquad v = tt}{\langle H, \text{while } e \text{ do } c\rangle \xrightarrow{\rho}_{EP} \langle H, c; \text{while } e \text{ do } c\rangle}$$

$$\text{EP-while 2}\frac{\langle H, e\rangle \overset{\rho}{\Downarrow}_{EP} v \qquad v = f\!f}{\langle H, \text{while } e \text{ do } c\rangle \xrightarrow{\rho}_{EP} \langle H, \mathit{skip}\rangle}$$

$$\text{EP-assign}\ \frac{\begin{array}{c}\langle H, e_1\rangle \overset{\rho}{\Downarrow}_{EP} {}^{\mathbb{W}}r^{\ell_r} \qquad \mathbb{P} = \beta(\rho) \qquad \mathbb{P} \geq_{\mathbb{L}} \mathbb{O}(r) \qquad \langle H, e_2\rangle \overset{\rho}{\Downarrow}_{EP} v^{\ell_v} \\ \rho \neq \overline{\mathbb{P}} \implies \ell_r \sqcap \ell_v \geq_{\mathbb{L}} \mathbb{O}(r)\end{array}}{\langle H, e_1 := e_2\rangle \xrightarrow{\rho}_{EP} \langle H[r \mapsto v], \mathit{skip}\rangle}$$

$$\text{EP-seq 1}\frac{\langle H, c_1\rangle \xrightarrow{\rho}_{EP} \langle H', c_1'\rangle}{\langle H, c_1; c_2\rangle \xrightarrow{\rho}_{EP} \langle H', c_1'; c_2\rangle}$$

$$\text{EP-seq 2}\frac{}{\langle H, \mathit{skip}; c_2\rangle \xrightarrow{\rho}_{EP} \langle H, c_2\rangle}$$

Program:

$$\text{EP-prg 1}\ \frac{\langle H, c\rangle \xrightarrow{\rho}_{EP} \langle H', c'\rangle}{\langle H, \rho\{c\}\rangle \rightarrow_{EP} \langle H, \rho\{c'\}\rangle}$$

$$\text{EP-comp 1}\ \frac{\langle H, P_1\rangle \rightarrow_{EP} \langle H', P_1'\rangle}{\langle H, P_1 \circ P_2\rangle \rightarrow_{EP} \langle H', P_1' \circ P_2\rangle}$$

$$\text{EP-comp 2}\ \frac{}{\langle H, \rho\{skip\} \circ P\rangle \rightarrow_{EP} \langle H, P\rangle}$$

Definition 12 (No Interesting References in only High Heap)**.** $nihrHH(H, \rho_A) \triangleq \forall r \in dom(H).\ \beta(\rho_A) \not\geq_{\mathbb{L}} \mathbb{O}(r) \wedge H(r) =\ {}^{\mathbb{W}}r'^{\ell} \implies \beta(\rho_A) \not\geq_{\mathbb{L}} \mathbb{O}(r') \implies r' \notin AIS$

Lemma 18 (Any AIS reference computed in a high region must have attackers label)**.** $\forall e, \rho, H, \rho_A.$
$\beta(\rho_A) \not\geq_{\mathbb{L}} \beta(\rho) \wedge nihrE(e, \rho_A) \wedge nihrHH(H, \rho_A) \wedge v =\ {}^{-}r'^{\ell} \wedge \beta(\rho_A) \not\geq_{\mathbb{L}} \mathbb{O}(r')$
$\wedge$
$\langle H, e\rangle \overset{\rho}{\Downarrow}_{EP} v$
$\implies$
$r' \notin AIS \vee \beta(\rho_A) \geq_{\mathbb{L}} \ell$

Proof. Proof by induction on $\overset{\rho}{\Downarrow}_{EP}$:

1. EP-val:
   Since $e = v =\ {}^{\mathbb{W}}r'$, $\beta(\rho_A) \not\geq_{\mathbb{L}} \mathbb{O}(r')$ and $nihrE(e, \rho_A)$
   therefore $r' \notin AIS$ (From Definition 8).

2. EP-deref:
   IH: $r^{\ell_r} \notin AIS \vee \beta(\rho_A) \geq_{\mathbb{L}} \ell_r$.
   Case analysisng IH:

   (a) $r^{\ell_r} \notin AIS$:

      i. $\beta(\rho_A) \geq_{\mathbb{L}} \mathbb{O}(r_r)$: $\beta(\rho_A) \geq_{\mathbb{L}} (\ell_r \sqcap \mathbb{O}(r_r))$

      ii. $\beta(\rho_A) \not\geq_{\mathbb{L}} \mathbb{O}(r_r)$: From $nihrHH(H, \rho_A)$ we know that $r' \notin AIS$

   (b) $\beta(\rho_A) \geq_{\mathbb{L}} \ell_r$: In this case $\beta(\rho_A) \geq_{\mathbb{L}} (\ell = \ell_r \sqcap \mathbb{O}(r))$

$\square$

Lemma 19 ($NSRC$ and *nihrHH* are invariants in high non-endorsed regions)**.** $\forall c, \rho, H, \rho_A.$
$\beta(\rho_A) \not\geq_{\mathbb{L}} \beta(\rho) \wedge \rho \neq \overline{\mathbb{P}} \wedge$
$nihrC(c, \rho_A) \wedge nihrHH(H, \rho_A) \wedge$
$\langle H, c\rangle \xrightarrow{\rho}_{EP} \langle H', c'\rangle$
$\implies$
$nihrC(c', \rho_A) \wedge nihrHH(H', \rho_A)$

Proof. Proof by induction on $\xrightarrow{\rho}_{EP}$

1. EP-if: Since $H' = H$, therefore $\mathit{nihrHH}(H', \rho_A)$.
   Since, $\mathit{nihrC}(c, \rho_A)$ therefore $\mathit{nihrC}(c_1, \rho_A)$.

2. EP-else: Since $H' = H$, therefore $\mathit{nihrHH}(H', \rho_A)$.
   Since, $\mathit{nihrC}(c, \rho_A)$ therefore $\mathit{nihrC}(c_2, \rho_A)$

3. EP-while 1: $H' = H$, therefore $\mathit{nihrHH}(H', \rho_A)$.
   Since, $\mathit{nihrC}(\text{while}e\text{do}c', \rho_A)$ therefore $\mathit{nihrC}(c', \rho_A)$

4. EP-while 2: $H' = H$, therefore $\mathit{nihrHH}(H', \rho_A)$.
   $\mathit{nihrC}(\mathit{skip}, \rho_A)$

5. EP-assign:
   2 cases arise:

   (a) $\beta(\rho_A) \geq_{\mathbb{L}} \mathbb{O}(r)$: $\mathit{nihrHH}(H', \rho_A)$ holds vacuously.

   (b) $\beta(\rho_A) \not\geq_{\mathbb{L}} \mathbb{O}(r)$: Again 2 cases arise from Lemma 18:

      i. $r \notin \mathit{AIS}$: 2 cases arise:

         A. $r' \notin \mathit{AIS}$: $\mathit{nihrHH}(H', \rho_A)$

         B. $\beta(\rho_A) \geq_{\mathbb{L}} \ell'_r$: This case cannot arise as EP-assign requires $\ell'_r \geq_{\mathbb{L}} \beta(\rho_r)$

      ii. $\beta(\rho_A) \geq_{\mathbb{L}} \ell_r$: This case cannot arise as EP-assign requires $\ell_r \geq_{\mathbb{L}} \beta(\rho_r)$

   $\mathit{nihrC}(\mathit{skip}, \rho_A)$

6. EP-seq 1: By IH $\mathit{nihrHH}(H', \rho_A)$ and $\mathit{nihrC}(c'_1, \rho_A)$. Therefore, $\mathit{nihrHH}(H', \rho_A)$
   Since, $\mathit{nihrC}(c_1; c_2, \rho_A)$ and from IH, $\mathit{nihrC}(c'_1; c_2, \rho_A)$

7. EP-seq 2: $H' = H$. Therefore, $\mathit{nihrHH}(H', \rho_A)$.
   Since $\mathit{nihrC}(\mathit{skip}; c, \rho_A)$ therefore $\mathit{nihrC}(c, \rho_A)$

□

Lemma 20 (*nihrHH* is invariant in low regions)**.** $\forall c, \rho, H, \rho_A.$
$\beta(\rho_A) \geq_{\mathbb{L}} \beta(\rho) \wedge$
$\mathit{nihrHH}(H, \rho_A) \wedge$
$\langle H, c\rangle \xrightarrow{\rho}_{EP} \langle H', c'\rangle$
$\Longrightarrow$
$\mathit{nihrHH}(H', \rho_A)$

Proof. Proof by induction on $\xrightarrow{\rho}_{EP}$

1. EP-if: Since $H' = H$, therefore $\mathit{nihrHH}(H', \rho_A)$.

2. EP-else: Since $H' = H$, therefore $\mathit{nihrHH}(H', \rho_A)$.

3. EP-while 1: $H' = H$, therefore $nihrHH(H', \rho_A)$.

4. EP-while 2: $H' = H$, therefore $nihrHH(H', \rho_A)$.

5. EP-assign:
   2 cases arise:
   (a) $\beta(\rho_A) \geq_{\mathbb{L}} \mathbb{O}(r)$: $nihrHH(H', \rho_A)$ holds vacuously.
   (b) $\beta(\rho_A) \not\geq_{\mathbb{L}} \mathbb{O}(r)$: This case cannot arise as EP-assign requires $\rho \geq_{\mathbb{L}} \mathbb{O}(r)$.

6. EP-seq 1: By IH $nihrHH(H', \rho_A)$ and $nihrC(c'_1, \rho_A)$. Therefore, $nihrHH(H', \rho_A)$

7. EP-seq 2: $H' = H$. Therefore, $nihrHH(H', \rho_A)$.

□

Lemma 21 (High non-endorsed regions dont change high AIS references)**.**
$\forall c, \rho, H, \rho_A.$
$\beta(\rho_A) \not\geq_{\mathbb{L}} \beta(\rho) \wedge \rho \neq \overline{\mathbb{P}} \wedge$
$nihrC(c, \rho_A) \wedge nihrH(H, \rho_A) \wedge$
$\langle H, c\rangle \xrightarrow{\rho}_{EP} \langle H', c'\rangle$
$\Longrightarrow$
$\forall\ r \in dom(AIS).\beta(\rho_A) \not\geq_{\mathbb{L}} \mathbb{O}(r) \implies H(r) = H'(r)$

Proof. Induction on $\xrightarrow{\rho}_{EP}$:

1. EP-if, EP-else, EP-while 1, EP-while 2, EP-seq 2:
   $H = H'$

2. EP-assign:
   From Lemma 18 we know that $\beta(\rho_A) \not\geq_{\mathbb{L}} \mathbb{O}(^{\mathbb{W}}r) \implies r \notin AIS \vee \beta(\rho_A) \geq_{\mathbb{L}} \ell_r$. In both cases either the assignment happens to a non-high AIS or assignment cant happen at all. Case analysing:
   (a) $r \notin AIS$:
      i. $\beta(\rho_A) \geq_{\mathbb{L}} \ell_v$: Assignment cannot happen
      ii. $\beta(\rho_A) \not\geq_{\mathbb{L}} \ell_v$: Clearly $\forall\ r \in dom(AIS).\beta(\rho_A) \not\geq_{\mathbb{L}} \mathbb{O}(r) \implies H(r) = H'(r)$
   (b) $\beta(\rho_A) \geq_{\mathbb{L}} \ell_r$: Assignment cannot happen

3. EP-seq 1:
   By IH

□

Lemma 22 (Low regions cannot write to high references of interest)**.** $\forall c, \rho, H, \rho_A.$
$\beta(\rho_A) \geq_{\mathbb{L}} \beta(\rho) \wedge$
$nihrH(H, \rho_A) \wedge$
$\langle H, c\rangle \xrightarrow{\rho}_{EP} \langle H', c'\rangle$
$\implies$
$\forall\ r \in dom(AIS).\beta(\rho_A) \not\geq_{\mathbb{L}} \mathbb{O}(r) \implies H(r) = H'(r)$

Proof. Induction on $\xrightarrow{\rho}_{EP}$:

1. EP-if, EP-else, EP-while 1, EP-while 2, EP-seq 2:
   $H = H'$
2. EP-assign:
   From EP-assign$\beta(\rho_A) \geq_{\mathbb{L}} \beta(\rho) \geq_{\mathbb{L}} \beta(\rho_w)$. Therefore, $\forall\ r \in dom(AIS).\beta(\rho_A) \not\geq_{\mathbb{L}} \mathbb{O}(r) \implies H(r) = H'(r)$
3. EP-seq 1:
   By IH

□

Lemma 23. *nihrP and nihrHH are Invariants on programs]* $\forall c, \rho, H, \rho_A.$
$nihrP(P, \rho_A) \wedge nihrHH(H, \rho_A) \wedge$
$\langle H, P\rangle \rightarrow_{EP} \langle H', P'\rangle$
$\implies$
$nihrP(P', \rho_A) \wedge nihrHH(H', \rho_A)$

Proof. Proof by inductipn on $\rightarrow_{EP}$

1. EP-prg 1:
   2 cases arise:
   (a) $\beta(\rho_A) \geq_{\mathbb{L}} \beta(\rho)$: Lemma 20 we know that $nihrHH(H', \rho_A)$
   Say, $P = \rho\{c\} \circ P''$ and $P' = \rho\{c'\} \circ P''$.
   Since $\beta(\rho_A) \geq_{\mathbb{L}} \beta(\rho')$ therefore, $nihrP(P', \rho_A)$
   (b) $\beta(\rho_A) \not\geq_{\mathbb{L}} \beta(\rho)$: From Lemma 19 we know that $nihrHH(H', \rho_A)$ and $nihrC(c', \rho_A)$.
   Say, $P = \rho\{c\} \circ P''$ and $P' = \rho\{c'\} \circ P''$.
   Therefore, $nihrP(P', \rho_A)$
2. EP-comp 1: From IH
3. EP-comp 2: Since $H' = H$ therefore $nihrP(skip \circ P, \rho_A)$ and $nihrHH(H', \rho_A)$

□

Corollary 11. $\forall P_1, H_1, \rho_A.$
$nihrP(P, \rho_A) \wedge nihrH(H, \rho_A) \wedge$
$Say\ P\ = \rho_1\{c_1\} \circ \ldots \rho_n\{c_n\}$
$\forall 1 \leq i \leq n.\ \rho_i \neq \overline{\mathbb{P}} \wedge$

$\langle H_1, P_1\rangle \to^*_{EP} \langle H_n, P_n\rangle$
$\implies$
$\forall r \in \mathit{AIS}.\beta(\rho_A) \not\geq_{\mathbb{L}} \mathbb{O}(r) \implies H(r) = H'(r)$

Proof. Say the reduction happens as follows: $\langle H_1, P_!\rangle \to_{EP} \langle H_2, P_2\rangle \ldots \langle H_{n-1}, P_{n-1}\rangle \to_{EP} \langle H_n, P_n\rangle$
By induction on the length of the reduction
IH1: $\forall\ r \in \mathit{dom}(\mathit{AIS}).\beta(\rho_A) \not\geq_{\mathbb{L}} \mathbb{O}(r) \implies H(r) = H_{n-1}(r)$
From Lemma 23 that $\mathit{nihrP}(P_{n-1}, \rho_A) \wedge \mathit{nihrHH}(H_{n-1}, \rho_A)$
Induction on $\to_{EP}$ on the last step:

1. EP-prg 1:
   From Lemma 21 and Lemma 22

2. EP-comp 1:
   By IH

3. EP-comp 2:
   $H = H'$

□

Theorem 12. $\forall\ \mathbb{E}_{\rho_A}, H.$
$\mathit{nihrP}(\mathbb{E}_{\rho_A}, \rho_A) \implies$
$\mathit{nihrHH}(H, \rho_A) \implies$
$\mathit{CDAF\text{-}E}(\mathbb{E}_{\rho_A}, H, \to_{EP})$

Proof. $\forall r \in \mathit{dom}(H)$ the following cases arise:

1. $\beta(\rho_A) \geq_{\mathbb{L}} \mathbb{O}(r)$: All these references can be writen directly by the attacker. So, for every write on a such a reference can be simulated by the attacker code only. Thus, satisfying the second disjunct.

2. $\beta(\rho_A) \not\geq_{\mathbb{L}} \mathbb{O}(r)$: From Corollary 11, $H_1(r) = H_2(r) = H(r)$.

□

4 Full provenance semantics

Expressions:

$$\text{FP-val}\ \frac{}{\langle H, v\rangle \Downarrow^{\rho}_{FP} v^{\top}} \qquad \text{FP-deref}\ \frac{\langle H, e\rangle \Downarrow^{\rho}_{FP}\ {}^{\mathbb{R}}r^{\ell_r} \qquad \mathbb{P}_r = \mathbb{O}(r) \qquad v = H(r)}{\langle H, !e\rangle \Downarrow^{\rho}_{FP} v^{\ell_r \sqcap \mathbb{P}_r}}$$

$$\text{FP-refComp}\ \frac{\langle H, e_1\rangle \Downarrow^{\rho}_{FP}\ {}^{\nu}r^{\ell_r} \qquad \langle H, e_2\rangle \Downarrow^{\rho}_{FP} n^{\ell_n} \qquad (r \oplus n) \in \mathit{dom}(H)}{\langle H, e_1 \oplus e_2\rangle \Downarrow^{\rho}_{FP}\ {}^{\nu}(r \oplus n)^{\ell_r \sqcap \ell_n}}$$

Commands:

$$\text{FP-if}\ \frac{\langle H, e\rangle \Downarrow^{\rho}_{FP} v^{\ell} \qquad v = tt}{\langle H, pc :: PC, \text{if } e \text{ then } c_1 \text{ else } c_2\rangle \rightarrow^{\rho}_{FP} \langle H, (pc \sqcap \ell) :: pc :: PC, c_1; \text{endif}\rangle}$$

$$\text{FP-else}\ \frac{\langle H, e\rangle \Downarrow^{\rho}_{FP} v^{\ell} \qquad v = \mathit{ff}}{\langle H, pc :: PC, \text{if } e \text{ then } c_1 \text{ else } c_2\rangle \rightarrow^{\rho}_{FP} \langle H, (pc \sqcap \ell) :: pc :: PC, c_2; \text{endif}\rangle}$$

$$\text{FP-endif}\ \frac{}{\langle H, pc :: PC, \text{endif}\rangle \rightarrow^{\rho}_{FP} \langle H, PC, \mathit{skip}\rangle}$$

$$\text{FP-while 1}\ \frac{\langle H, e\rangle \Downarrow^{\rho}_{FP} v^{\ell} \qquad v = tt}{\langle H, pc :: PC, \text{while } e \text{ do } c\rangle \rightarrow^{\rho}_{FP} \langle H, (pc \sqcap \ell) :: pc :: PC, c; \text{while } e \text{ do } c; \text{endwhile}\rangle}$$

$$\text{FP-while 2}\ \frac{\langle H, e\rangle \Downarrow^{\rho}_{FP} v \qquad v = \mathit{ff}}{\langle H, PC, \text{while } e \text{ do } c\rangle \rightarrow^{\rho}_{FP} \langle H, PC, \mathit{skip}\rangle}$$

$$\text{FP-endwhile}\ \frac{}{\langle H, pc :: PC, \text{endwhile}\rangle \rightarrow^{\rho}_{FP} \langle H, PC, \mathit{skip}\rangle}$$

$$\text{FP-assign}\ \frac{\begin{array}{c}\langle H, e_1\rangle \Downarrow^{\rho}_{FP} {}^{\mathbb{W}}r^{\ell_r} \qquad \mathbb{P} = \beta(\rho) \qquad \mathbb{P} \geq_{\mathbb{L}} \mathbb{O}(r) \qquad \langle H, e_2\rangle \Downarrow^{\rho}_{FP} v^{\ell_v} \\ \rho \neq \overline{\mathbb{P}} \implies \ell_r \sqcap \ell_v \sqcap pc \geq_{\mathbb{L}} \mathbb{O}(r)\end{array}}{\langle H, pc :: PC, e_1 := e_2\rangle \rightarrow^{\rho}_{FP} \langle H[r \mapsto v], pc :: PC, \mathit{skip}\rangle}$$

$$\text{FP-seq 1}\ \frac{\langle H, PC, c_1\rangle \rightarrow^{\rho}_{FP} \langle H', PC' c_1'\rangle}{\langle H, PC, c_1; c_2\rangle \rightarrow^{\rho}_{FP} \langle H', PC', c_1'; c_2\rangle}$$

$$\text{FP-seq 2}\ \frac{}{\langle H, PC, \mathit{skip}; c_2\rangle \rightarrow^{\rho}_{FP} \langle H, PC, c_2\rangle}$$

Program:

$$\text{FP-prg 1}\ \frac{\langle H, PC, c\rangle \rightarrow^{\rho}_{FP} \langle H', PC', c'\rangle}{\langle H, PC, \rho\{c\}\rangle \rightarrow_{FP} \langle H', PC', \rho\{c'\}\rangle}$$

$$\text{FP-comp 1}\ \frac{\langle H, PC, P_1\rangle \rightarrow_{FP} \langle H', PC', P_1'\rangle}{\langle H, PC, P_1 \circ P_2\rangle \rightarrow_{FP} \langle H', PC', P_1' \circ P_2\rangle}$$

$$\text{FP-comp 2}\ \frac{}{\langle H, [\top], \rho\{\mathit{skip}\} \circ P\rangle \rightarrow_{FP} \langle H, [\top], P\rangle}$$

Definition 13. *Two labeled values* $v_1^{\ell_1}$ *and* $v_2^{\ell_2}$ *are* ρ_A*-equivalent, written* $v_1^{\ell_1} \underset{\rho_A}{\overset{\mathbb{L}}{\sim}} v_2^{\ell_2}$*, iff either:*

1. $\beta(\rho_A) \not\geq_{\mathbb{L}} (\ell_1 = \ell_2)$ *and* $v_1 = v_2$ *or*
2. $\beta(\rho_A) \geq_{\mathbb{L}} \ell_1$ *and* $\beta(\rho_A) \geq_{\mathbb{L}} \ell_2$

Definition 14 (PC Stack)**.** *PC stack* (PC) *is a stack of labels from the integrity lattice*

Definition 15 (PC stack equivalence)**.**

$$PC_1 \underset{\rho_A}{\overset{\mathbb{L}}{\sim}} PC_2 \triangleq \begin{cases} \beta(\rho_A) \geq_{\mathbb{L}} pc_1 \wedge \beta(\rho_A) \geq_{\mathbb{L}} pc_2 & \\ \beta(\rho_A) \geq_{\mathbb{L}} pc_1 \wedge \beta(\rho_A) \not\geq_{\mathbb{L}} pc_2 \implies \mathit{false} & PC_1 = PC'_1@[pc_1] \wedge PC_2 = PC'_2@[pc_2] \\ \beta(\rho_A) \not\geq_{\mathbb{L}} pc_1 \implies \beta(\rho_A) \not\geq_{\mathbb{L}} pc_2 \wedge PC'_1 \underset{\rho_A}{\overset{\mathbb{L}}{\sim}} PC'_2 & \\ \beta(\rho_A) \not\geq_{\mathbb{L}} pc_1 \implies \mathit{false} & PC_1 = PC'_1@[pc_1] \wedge PC_2 = \mathit{Nil} \\ \beta(\rho_A) \geq_{\mathbb{L}} pc_1 & \\ \beta(\rho_A) \not\geq_{\mathbb{L}} pc_2 \implies \mathit{false} & PC_1 = \mathit{Nil} \wedge PC_2 = PC'_2@[pc_2] \\ \beta(\rho_A) \geq_{\mathbb{L}} pc_2 & \\ \mathit{true} & \mathit{otherwise} \end{cases}$$

Definition 16 (Heap equivalence - CDA)**.** $\forall H_1, H_2, \rho_A.\ H_1 \underset{\rho_A}{\overset{\mathbb{L}}{\sim}} H_2 \triangleq$ $dom(H_1) = dom(H_2) \implies \forall r \in dom(H_1).\ \beta(\rho_A) \not\geq_{\mathbb{L}} (\mathbb{O}(r) \implies H_1(r) = H_2(r)$

Lemma 24. $\forall H_1, H_2, e, \rho, \rho_A.$
$H_1 \underset{\rho_A}{\overset{\mathbb{L}}{\sim}} H_2 \wedge \rho \neq \overline{\mathbb{P}} \wedge$
$\langle H_1, e\rangle \Downarrow^{\rho}_{FP} v_1^{\ell_1} \wedge$
$\langle H_2, e\rangle \Downarrow^{\rho}_{FP} v_2^{\ell_2}$
$\implies v_1 \underset{\rho_A}{\overset{\mathbb{L}}{\sim}} v_2$

Proof. Proof by induction on $\Downarrow^{\rho}_{FP}$

1. FP-val: Trivial
2. FP-deref:
   IH: $r_1^{\ell_{r1}} \underset{\rho_A}{\overset{\mathbb{L}}{\sim}} r_2^{\ell_{r2}}$ The following cases arise:
   (a) $\beta(\rho_A) \not\geq_{\mathbb{L}} (\ell_{r1} = \ell_{r2})$: In this case $r_1 = r_2 = r$. 2 cases:
      i. $\beta(\rho_A) \not\geq_{\mathbb{L}} \mathbb{O}(r)$: Since $H_1 \underset{\rho_A}{\overset{\mathbb{L}}{\sim}} H_2$, therefore $v_1 = v_2$
      ii. $\beta(\rho_A) \geq_{\mathbb{L}} \mathbb{O}(r)$: $v_1^{\ell_1} \underset{\rho_A}{\overset{\mathbb{L}}{\sim}} v_2^{\ell_2}$ since $\ell_1 = \ell_2 = \mathbb{O}(r) \sqcap \ell_{r1}$ and thus $\beta(\rho_A) \geq_{\mathbb{L}} \ell_1$

(b) $\beta(\rho_A) \geq_{\mathbb{L}} \ell_{r1} \wedge \beta(\rho_A) \geq_{\mathbb{L}} \ell_{r2}$: $v_1^{\ell_1} \underset{\rho_A}{\overset{\mathbb{L}}{\sim}} v_2^{\ell_2}$ since $\ell_1 = \mathbb{O}(r_{r1}) \sqcap \ell_{r1}$ and thus $\beta(\rho_A) \geq_{\mathbb{L}} \ell_1$.
Similary $\ell_2 = \mathbb{O}(r_{r2}) \sqcap \ell_{r1}$ and thus $\beta(\rho_A) \geq_{\mathbb{L}} \ell_2$

3. FP-refComp:
IH1: $r_1^{\ell_{r1}} \underset{\rho_A}{\overset{\mathbb{L}}{\sim}} r_2^{\ell_{r2}}$
IH2: $v_1^{\ell_{n1}} \underset{\rho_A}{\overset{\mathbb{L}}{\sim}} v_2^{\ell_{n2}}$
2 cases arise:
   (a) $\beta(\rho_A) \not\geq_{\mathbb{L}} (\ell_{r1} = \ell_{r2})$:
   $r_1 = r_2$
      i. $\beta(\rho_A) \not\geq_{\mathbb{L}} (\ell_{n1} = \ell_{n2})$:
      $v_1 = v_2$ therefore $(r_1 \oplus v_1) = (r_2 \oplus v_2)$
      ii. $\beta(\rho_A) \geq_{\mathbb{L}} \ell_{n1} \wedge \beta(\rho_A) \geq_{\mathbb{L}} \ell_{n2}$:
      $(r_1 \oplus v_1)^{\ell'} \underset{\rho_A}{\overset{\mathbb{L}}{\sim}} (r_2 \oplus v_2)^{prov''}$ as $\beta(\rho_A) \geq_{\mathbb{L}} \ell'$ and $\beta(\rho_A) \geq_{\mathbb{L}} \ell''$
   (b) $\beta(\rho_A) \geq_{\mathbb{L}} \ell_{r1} \wedge \beta(\rho_A) \geq_{\mathbb{L}} \ell_{r2}$:
      i. $\beta(\rho_A) \not\geq_{\mathbb{L}} (\ell_{n1} = \ell_{n2})$:
      $(r_1 \oplus v_1)^{\ell'} \underset{\rho_A}{\overset{\mathbb{L}}{\sim}} (r_2 \oplus v_2)^{prov''}$ as $\beta(\rho_A) \geq_{\mathbb{L}} \ell'$ and $\beta(\rho_A) \geq_{\mathbb{L}} \ell''$
      ii. $\beta(\rho_A) \geq_{\mathbb{L}} \ell_{n1} \wedge \beta(\rho_A) \geq_{\mathbb{L}} \ell_{n2}$:
      $(r_1 \oplus v_1)^{\ell'} \underset{\rho_A}{\overset{\mathbb{L}}{\sim}} (r_2 \oplus v_2)^{prov''}$ as $\beta(\rho_A) \geq_{\mathbb{L}} \ell'$ and $\beta(\rho_A) \geq_{\mathbb{L}} \ell''$

$\square$

Lemma 25 (CDA-Simulation)**.** $\forall H_1, H_2, c, pc_1, pc_2, PC_1, PC_2, \rho, \rho_A.$
$\rho \neq \overline{\mathbb{P}} \wedge H_1 \underset{\rho_A}{\overset{\mathbb{L}}{\sim}} H_2 \wedge (pc_1 :: PC_1) \underset{\rho_A}{\overset{\mathbb{L}}{\sim}} (pc_2 :: PC_2) \wedge$
$\langle H_1, pc_1 :: PC_1, c\rangle \rightarrow^{\rho}_{FP} \langle H_1', pc_1' :: PC_1', c'\rangle \wedge$
$\langle H_2, pc_2 :: PC_2, c\rangle \rightarrow^{\rho}_{FP} \langle H_2', pc_2' :: PC1_2', c'\rangle \wedge \beta(\rho_A) \not\geq_{\mathbb{L}} (pc_1 = pc_2) \wedge \beta(\rho_A) \not\geq_{\mathbb{L}} (pc_1' = pc_2')$
$\implies$
$H_1' \underset{\rho_A}{\overset{\mathbb{L}}{\sim}} H_2' \wedge pc_1' :: PC_1' \underset{\rho_A}{\overset{\mathbb{L}}{\sim}} pc_2' :: PC_2'$

Proof. Proof by induction on the $\rightarrow^{\rho}_{FP}$:

1. FP-endif:
Here, $c =$ endif.
$H_1' = H_1 \underset{\rho_A}{\overset{\mathbb{L}}{\sim}} H_2 = H_2'$
From definition 15, $pc_1' :: PC_1' = PC_1 \underset{\rho_A}{\overset{\mathbb{L}}{\sim}} PC_2 = pc_2' :: PC_2'$
$c' = skip$

2. FP-if:
Here, $c = \text{if } e \text{ then } c_1 \text{ else } c_2$, $H_1' = H_1 \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} H_2 = H_2'$.
From Lemma 24 we know that $v_1^{\ell_1} \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} v_2^{\ell_1}$
   (a) When $\beta(\rho_A) \not\geq_{\mathbb{L}} (\ell_1 = \ell_2 = \ell)$: Since, $\beta(\rho_A) \not\geq_{\mathbb{L}} (pc_1 = pc_2)$.
   Therefore, $PC_1' = pc_1 \sqcap \ell :: pc_1 :: PC_1 \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} pc_2 \sqcap \ell :: pc_2 :: PC_2 = PC_2'$.
   $c_1' = c_1; \text{endif} = c_2'$
   (b) When $\beta(\rho_A) \geq_{\mathbb{L}} \ell_1$ and $\beta(\rho_A) \geq_{\mathbb{L}} \ell_2$: Case cannot arise as $\beta(\rho_A) \not\geq_{\mathbb{L}} (pc_1' = pc_2')$

3. FP-else:
Here, $c = \text{if } e \text{ then } c_1 \text{ else } c_2$, $H_1' = H_1 \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} H_2 = H_2'$
From Lemma 24 we know that $v_1^{\ell_1} \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} v_2^{\ell_1}$
   (a) When $\beta(\rho_A) \not\geq_{\mathbb{L}} (\ell_1 = \ell_2 = \ell)$: Since, $\beta(\rho_A) \not\geq_{\mathbb{L}} (pc_1 = pc_2)$.
   Therefore, $PC_1' = pc_1 \sqcap \ell :: pc_1 :: PC_1 \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} pc_2 \sqcap \ell :: pc_2 :: PC_2 = PC_2'$.
   $c_1' = c_2; \text{endif} = c_2'$
   (b) When $\beta(\rho_A) \geq_{\mathbb{L}} \ell_1$ and $\beta(\rho_A) \geq_{\mathbb{L}} \ell_2$: Case cannot arise as $\beta(\rho_A) \not\geq_{\mathbb{L}} (pc_1' = pc_2')$

4. FP-while 1:
Here, $c = \text{while } e \text{ do } c_1$, $H_1' = H_1 \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} H_2 = H_2'$
From Lemma 24 we know that $v_1^{\ell_1} \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} v_2^{\ell_1}$
   (a) When $\beta(\rho_A) \not\geq_{\mathbb{L}} (\ell_1 = \ell_2 = \ell)$: Since, $\beta(\rho_A) \not\geq_{\mathbb{L}} (pc_1 = pc_2)$.
   Therefore, $PC_1' = pc_1 \sqcap \ell :: pc_1 :: PC_1 \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} pc_2 \sqcap \ell :: pc_2 :: PC_2 = PC_2'$.
   $c_1' = c_1; \text{endwhile} = c_2'$
   (b) When $\beta(\rho_A) \geq_{\mathbb{L}} \ell_1$ and $\beta(\rho_A) \geq_{\mathbb{L}} \ell_2$: Case cannot arise as $\beta(\rho_A) \not\geq_{\mathbb{L}} (pc_1' = pc_2')$

5. FP-while 2:
Here, $c = \text{while } e \text{ do } c_1$, $H_1' = H_1 \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} H_2 = H_2'$, $PC_1' = pc_1 :: PC_1 \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} pc_2 :: PC_2 = PC_2'$
and $c_1' = \mathit{skip} = c_2'$

6. FP-endwhile:
Here, $c = \text{endwhile}$
$H_1 \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} H_2 = H_2'$, $PC_1' = pc_1 :: PC_1 \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} pc_2 :: PC_2 = PC_2'$ and
$c_1' = \mathit{skip} = c_2'$

7. FP-assign:
   Here, $c = e_1 := e_2$
   From Lemma 24 we know that $r_1^{\ell_{r1}} \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} r_2^{\ell_{r2}}$ and $v_1^{\ell_{v1}} \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} v_2^{\ell_{v2}}$

   (a) $\beta(\rho_A) \not\geq_{\mathbb{L}} (\ell_{r1} = \ell_{r2})$ and $\beta(\rho_A) \not\geq_{\mathbb{L}} (\ell_{v1} = \ell_{v2})$:
   $r_1 = r_2$ and $v_1 = v_2$.
   Since assignment happens therefore $\rho \geq_{\mathbb{L}} \mathbb{O}(r_1)$ and $pc \sqcap \ell_{r1} \sqcap \ell_{v1} \geq_{\mathbb{L}} \mathbb{O}(r_1)$. And since, $H_1'(r_1) = v_1 = v_2 = H_2'(r_2)$. Therefore $H_1' \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} H_2'$ from definition 16.

   (b) $\beta(\rho_A) \not\geq_{\mathbb{L}} (\ell_{r1} = \ell_{r2})$ and $\beta(\rho_A) \geq_{\mathbb{L}} \ell_{v1}, \beta(\rho_A) \geq_{\mathbb{L}} \ell_{v2}$:
   $r_1 = r_2$. Since $\beta(\rho_A) \geq_{\mathbb{L}} \ell_{v1}, \beta(\rho_A) \geq_{\mathbb{L}} \ell_{v2}$, therefore $\beta(\rho_A) \geq_{\mathbb{L}} (pc_1 \sqcap \ell_{v1} \sqcap \ell_{r1})$ and $\beta(\rho_A) \geq_{\mathbb{L}} (pc_1 \sqcap \ell_{v2} \sqcap \ell_{r2})$.
   Since assignment happens therefore $\rho \geq_{\mathbb{L}} \mathbb{O}(r_1)$ and $\beta(\rho_A) \geq_{\mathbb{L}} (pc \sqcap \ell_{r1} \sqcap \ell_{v1}) \geq_{\mathbb{L}} \mathbb{O}(r_1)$.
   And since $\beta(\rho_A) \geq_{\mathbb{L}} \mathbb{O}(r_1)$, therefore $H_1' \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} H_2'$ from definition 16

   (c) $\beta(\rho_A) \geq_{\mathbb{L}} \ell_{r1}, \beta(\rho_A) \geq_{\mathbb{L}} \ell_{r2}$ and $\beta(\rho_A) \not\geq_{\mathbb{L}} (\ell_{v1} = \ell_{v2})$:
   $v_1 = v_2$. Since $\beta(\rho_A) \geq_{\mathbb{L}} \ell_{r1}, \beta(\rho_A) \geq_{\mathbb{L}} \ell_{r2}$, therefore $\beta(\rho_A) \geq_{\mathbb{L}} (pc_1 \sqcap \ell_{v1} \sqcap \ell_{r1})$ and $\beta(\rho_A) \geq_{\mathbb{L}} (pc_1 \sqcap \ell_{v2} \sqcap \ell_{r2})$.
   Since assignment happens therefore $\rho \geq_{\mathbb{L}} \mathbb{O}(r_1)$ and $\beta(\rho_A) \geq_{\mathbb{L}} (pc \sqcap \ell_{r1} \sqcap \ell_{v1}) \geq_{\mathbb{L}} \mathbb{O}(r_1)$.
   Similarly $\rho \geq_{\mathbb{L}} \mathbb{O}(r_2)$ and $\beta(\rho_A) \geq_{\mathbb{L}} (pc \sqcap \ell_{r1} \sqcap \ell_{v1}) \geq_{\mathbb{L}} \mathbb{O}(r_2)$. And since $\beta(\rho_A) \geq_{\mathbb{L}} \mathbb{O}(r_1)$ and $\beta(\rho_A) \geq_{\mathbb{L}} \mathbb{O}(r_2)$, therefore $H_1' \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} H_2'$ from definition 16

   (d) $\beta(\rho_A) \geq_{\mathbb{L}} \ell_{r1}, \beta(\rho_A) \geq_{\mathbb{L}} \ell_{r2}$ and $\beta(\rho_A) \geq_{\mathbb{L}} \ell_{v1}, \beta(\rho_A) \geq_{\mathbb{L}} \ell_{v2}$:
   $\beta(\rho_A) \geq_{\mathbb{L}} (pc_1 \sqcap \ell_{v1} \sqcap \ell_{r1})$ and $\beta(\rho_A) \geq_{\mathbb{L}} (pc_1 \sqcap \ell_{v2} \sqcap \ell_{r2})$.
   Since assignment happens therefore $\rho \geq_{\mathbb{L}} \mathbb{O}(r_1)$ and $\beta(\rho_A) \geq_{\mathbb{L}} (pc \sqcap \ell_{r1} \sqcap \ell_{v1}) \geq_{\mathbb{L}} \mathbb{O}(r_1)$.
   Similarly $\rho \geq_{\mathbb{L}} \mathbb{O}(r_2)$ and $\beta(\rho_A) \geq_{\mathbb{L}} (pc \sqcap \ell_{r1} \sqcap \ell_{v1}) \geq_{\mathbb{L}} \mathbb{O}(r_2)$. And since $\beta(\rho_A) \geq_{\mathbb{L}} \mathbb{O}(r_1)$ and $\beta(\rho_A) \geq_{\mathbb{L}} \mathbb{O}(r_2)$, therefore $H_1' \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} H_2'$ from definition 16

   $PC_1' = pc_1 :: PC_1 \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} pc_2 :: PC_2 = PC_2'$
   $c' = \mathit{skip}$

8. FP-seq 1:
   Here, $c = \ c_1; c_2$
   From IH $H_1' \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} H_2'$, $\rho \sqcap pc_1' :: PC_1' \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} pc_2' :: PC_2'$ and $\beta(\rho_A) \not\geq_{\mathbb{L}} (pc_1' = pc_2')$
   $\implies c_1' = c_2'$

9. FP-seq 2:
   Here, $c = \ \mathit{skip}; c_1$
   From IH $H_1' = H_1 \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} H_2 = H_2'$, $PC_1' = PC_1 \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} PC_2 = PC_2'$ and $c' = \mathit{skip}$

$\square$

Lemma 26 (CDA-High to Low transition)**.** $\forall H_1, H_2, c, PC_1, PC_2, \rho, \rho_A.$
$\rho \neq \overline{\mathbb{P}} \wedge H_1 \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} H_2 \wedge (pc_1 :: PC_1) \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} PC_2 \wedge$
$\langle H_1, pc_1 :: PC_1, c\rangle \rightarrow^{\rho}_{FP} \langle H_1', pc_1' :: PC_1', c'\rangle \wedge \beta(\rho_A) \not\geq_{\mathbb{L}} pc_1 \wedge \beta(\rho_A) \geq_{\mathbb{L}} pc_1'$
$\implies$
$H_1' \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} H_2 \wedge pc_1' : PC_1' \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} PC_2$

Proof. Proof by induction on the $\rightarrow^{\rho}_{FP}$ relation:

1. FP-endif, FP-endwhile:
   This case cannot arise as PC is popped in this case and we cannot go from a $\beta(\rho_A) \not\geq_{\mathbb{L}} \rho \sqcap pc_1$ to $\beta(\rho_A) \geq_{\mathbb{L}} \rho \sqcap pc_1'$

2. FP-if, FP-else, FP-while 1:
   $H_1' = H_1 \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} H_2$
   Since $\beta(\rho_A) \geq_{\mathbb{L}} pc_1'$.
   Therefore from definition 15 $PC_1' = pc_1 \sqcap \ell :: pc_1 :: PC_1'' \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} PC_2$

3. FP-while 2, FP-assign, FP-seq 2:
   Case cannot arise as $pc_1 :: PC_1' = pc_1' :: PC_1$

4. FP-seq 1:
   From IH, $H_1' \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} H_2$, $PC_1' \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} PC_2$

$\square$

Lemma 27 (CDA-Confinment)**.** $\forall H_1, H_2, c, PC_1, PC_2, \rho, \rho_A.$
$\rho \neq \overline{\mathbb{P}} \wedge H_1 \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} H_2 \wedge (PC_1 = (pc_1 :: PC_1'')) \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} PC_2 \wedge$
$\langle H_1, PC_1, c\rangle \rightarrow^{\rho}_{FP} \langle H_1', PC_1', c'\rangle \wedge \beta(\rho_A) \geq_{\mathbb{L}} pc_1 \implies$
$H_1' \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} H_2 \wedge PC_1' \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} PC_2$

Proof. Proof by induction on the $\rightarrow^{\rho}_{FP}$ relation:

1. FP-endif, FP-endwhile:
   $H_1' = H_1 \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} H_2$
   Since $pc_1 :: PC_1'' \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} PC_2$and $\beta(\rho_A) \geq_{\mathbb{L}} pc_1$ (given).
   Therefore, from definition 15 $(PC_1' = PC_1'') \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} PC_2$

2. FP-if, FP-else:
   $H_1' = H_1 \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} H_2$

Since $pc_1 :: PC_1'' \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} PC_2$ and $\beta(\rho_A) \geq_{\mathbb{L}} pc_1$ (given).

Therefore from definition 15 $PC_1' = pc_1 \sqcap \ell :: pc_1 :: PC_1'' \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} PC_2$

3. FP-while 1:
   $H_1' = H_1 \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} H_2$

   Since $pc_1 :: PC_1'' \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} PC_2$ and $\beta(\rho_A) \geq_{\mathbb{L}} pc_1$ (given).

   Therefore from definition 15 $PC_1' = pc_1 \sqcap \ell :: pc_1 :: PC_1'' \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} PC_2$

4. FP-while 2:
   $H_1' = H_1 \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} H_2$, $PC_1' = PC_1 \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} PC_2$

5. FP-assign:
   From FP-assign, since $\beta(\rho_A) \geq_{\mathbb{L}} pc_1$, therefore $\beta(\rho_A) \geq_{\mathbb{L}} (pc_1 \sqcap \ell_{r1} \sqcap \ell_{v1})$.
   Since the assignment happens therefore $\beta(\rho_A) \geq_{\mathbb{L}} (pc_1 \sqcap \ell_{r1} \sqcap \ell_{v1}) \geq_{\mathbb{L}} \rho_r$
   Therefore, $H_1' \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} H_2$ from definition 16 and $PC_1' = PC_1 \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} PC_2$

6. FP-seq 1:
   From IH, $H_1' \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} H_2$, $PC_1' \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} PC_2$

7. FP-seq 2:
   From IH, $H_1' = H_1 \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} H_2$, $PC_1' = PC_1 \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} PC_2$

□

Lemma 28 (CDA-Low to high transition)**.** $\forall H_1, H_2, c, PC_1, PC_2, \rho, \rho_A.$
$\rho \neq \overline{\mathbb{P}} \wedge H_{11} \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} H_{21} \wedge$
$(pc_{11} :: PC_{11}) \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} pc_{21} :: PC_{21} \wedge$
$\beta(\rho_A) \not\geq_{\mathbb{L}} pc_{11} \wedge \beta(\rho_A) \not\geq_{\mathbb{L}} pc_{21} \wedge$
$\beta(\rho_A) \geq_{\mathbb{L}} pc_{12} \wedge \beta(\rho_A) \geq_{\mathbb{L}} pc_{22} \wedge$
$\forall 2 \leq i \leq n.\ \beta(\rho_A) \geq_{\mathbb{L}} pc_{1i} \wedge$
$\forall 2 \leq j \leq m.\ \beta(\rho_A) \geq_{\mathbb{L}} pc_{2j} \wedge$
$\beta(\rho_A) \not\geq_{\mathbb{L}} pc_{1n} \wedge \beta(\rho_A) \not\geq_{\mathbb{L}} pc_{2m} \wedge$
$\langle H_{11}, pc_{11} :: PC_{11}, c\rangle \rightarrow^{\rho}_{FP} \langle H_{12}, pc_{12} :: PC_{12}, c_{12}\rangle \rightarrow^{\rho *}_{FP} \langle H_{1n}, pc_{1n} :: PC_{1n}, c_{1n}\rangle$
$\wedge$
$\langle H_{21}, pc_{21} :: PC_{21}, c\rangle \rightarrow^{\rho}_{FP} \langle H_{22}, pc_{22} :: PC_{22}, c_{22}\rangle \rightarrow^{\rho *}_{FP} \langle H_{2m}, pc_{2m} :: PC_{2m}, c_{2m}\rangle$
$\Longrightarrow$
$H_{1n} \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} H_{2m} \wedge pc_{1n} : PC_{1n} \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} pc_{2m} :: PC_{2m} \wedge c_{1n} = c_{2m}$

Proof. To prove $H_{1n} \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} H_{2m}$ and $pc_{1n} : PC_{1n} \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} pc_{2m} :: PC_{2m}$:

$H_{11} \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} H_{21}$ and $(pc_{11} :: PC_{11}) \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} pc_{21} :: PC_{21}$ (given)

From Lemma 26 we know that $H_{11} \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} H_{21}$ and $pc_{11} : PC_{11} \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} pc_{21} :: PC_{21}$
From Lemma 27 we know that $H_{1n-1} \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} H_{21}$ and $pc_{1n-1} : PC_{1n-1} \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} pc_{21} :: PC_{21}$
Similarly from Lemma 27 we also know that $H_{11} \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} H_{2m-1}$ and $pc_{11} : PC_{11} \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} pc_{2m-1} :: PC_{2m-1}$
Therefore, $H_{1n-1} \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} H_{2m-1}$ and $pc_{1n-1} : PC_{1n-1} \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} pc_{2m-1} :: PC_{2m-1}$

To prove $c_{1n} = c_{2m}$:
By induction on c:

1. $c = \text{if } e \text{ then } c_1 \text{ else } c_2$:
   From FP-if and FP-else we know that $c_{12} = c_1; \text{endif}$ and $c_{22} = c_2; \text{endif}$ or $c_{11} = c_2; \text{endif}$ and $c_{21} = c_1; \text{endif}$
   Now, since $\beta(\rho_A) \geq_{\mathbb{L}} pc_{1n-1}$ and $\beta(\rho_A) \geq_{\mathbb{L}} pc_{2m-1}$ therefore $c_{1n-1} = c_{2m-1} = \text{endif}$
   From T-endif $c_{1n} = c_{2m} = skip$

2. $c = \text{while } e \text{ do } c_1$:
   Since, $\beta(\rho_A) \geq_{\mathbb{L}} pc_{12}$ and $\beta(\rho_A) \geq_{\mathbb{L}} pc_{22}$ therefore from FP-while 1 $c_{12} = c_1; \text{while } e \text{ do } c_1; \text{endwhile}$ and $c_{22} = c_1; \text{while } e \text{ do } c_1; \text{endwhile}$
   Now, since $\beta(\rho_A) \geq_{\mathbb{L}} pc_{1n-1}$ and $\beta(\rho_A) \geq_{\mathbb{L}} pc_{2m-1}$ therefore $c_{1n-1} = c_{2m-1} = \text{endwhile}$
   From T-endWhile $c_{1n} = c_{2m} = skip$

3. $c = skip$, $c = e_1 := e_2$:
   This case cannot arise

4. $c = c_1; c_2$:
   From IH

□

Lemma 29 (CDA-Attacker-Confinment)**.** $\forall H_1, H_2, c, PC_1, PC_2, \rho, \rho_A.$
$\rho \neq \overline{\mathbb{P}} \wedge \rho = \rho_A \wedge H_1 \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} H_2 \wedge$
$\langle H_1, \top, c \rangle \overset{*}{\rightarrow}^{\rho}_{FP} \langle H_1', \top, c' \rangle \implies$
$H_1' \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} H_2$

Proof. Say the reduction happens in the following way:
$\langle H_1, \top, c \rangle \overset{*}{\rightarrow}^{\rho}_{FP} \langle H_{n-1}, PC_{n-1}, c_{n-1} \rangle \rightarrow^{\rho}_{FP} \langle H_1', \top, c' \rangle$
IH1: $H_{n-1} \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} H_2$
By induction on the last reduction:

1. FP-if, FP-else, FP-endif, FP-while 1, FP-while 2, FP-endWhile, FP-seq 2:
   $H_1' = H_{n-1}$. Therefore for IH1

2. FP-assign:
   From FP-assign we know that $\beta(\rho) \geq_{\mathbb{L}} \mathbb{O}(r)$ and since $\rho = \rho_A$ therefore $H_1' \underset{\rho_A}{\overset{\mathbb{L}}{\sim}} H_{n-1} \underset{\rho_A}{\overset{\mathbb{L}}{\sim}} H_2$ (from IH1 and Definition 16)

3. FP-seq 1:
   From IH and IH1

□

Definition 17 (State). $\mathbb{S}(\langle H, PC, \rho\{c\}\rangle \triangleq (H, PC)$

Definition 18 (Trace). $\mathbb{T}(\langle H_1, pc_1 :: PC_1, \rho\{c_1\}\rangle \rightarrow^*_{FP} \langle H_n, pc_n :: PC_n, \rho\{c_n\}\rangle) \triangleq \{\mathbb{S}(\langle H_i, pc_i :: PC_i, \rho\{c_i\}\rangle) | 1 \leq i \leq n \wedge \beta(\rho_A) \not\geq_{\mathbb{L}} pc_i\}$

Definition 19 (Trace equivalence). $\mathbb{T}_1 \underset{\rho_A}{\overset{\mathbb{L}}{\sim}} \mathbb{T}_2 \triangleq$

$|\mathbb{T}_1| = |\mathbb{T}_2| \wedge \forall 1 \leq i \leq |\mathbb{T}_1|.\ \mathbb{T}_1(i).H \underset{\rho_A}{\overset{\mathbb{L}}{\sim}} \mathbb{T}_2(i).H \wedge \mathbb{T}_1(i).PC \underset{\rho_A}{\overset{\mathbb{L}}{\sim}} \mathbb{T}_2(i).PC$

Lemma 30 (NI-for-a-region). $\forall H_1, H_2, c, pc_1, pc_2, PC_1, PC_2, \rho, \rho_A.$
$\rho \neq \overline{\mathbb{P}} \wedge \rho \neq \rho_A \wedge H_1 \underset{\rho_A}{\overset{\mathbb{L}}{\sim}} H_2 \wedge$
$\langle H_1, \top, \rho\{c\}\rangle \rightarrow^*_{FP} \langle H_1', \top, \rho\{skip\}\rangle \wedge$
$\langle H_2, \top, \rho\{c\}\rangle \rightarrow^*_{FP} \langle H_2', \top, \rho\{skip\}\rangle \implies$
$\mathbb{T}(\langle H_1, \top, \rho\{c\}\rangle \rightarrow^*_{FP} \langle H_1', \top, \rho\{skip\}\rangle) \underset{\rho_A}{\overset{\mathbb{L}}{\sim}} \mathbb{T}(\langle H_2, \top, \rho\{c\}\rangle \rightarrow^*_{FP} \langle H_2', \top, \rho\{skip\}\rangle)$

Proof. Both the configurations start in equivalent heaps ($H_1 \underset{\rho_A}{\overset{\mathbb{L}}{\sim}} H_2$) and same PC stacks ($\top$).
Say $TR_1 = \mathbb{T}(\langle H_1, \top, \rho\{c\}\rangle \rightarrow^*_{FP} \langle H_1', \top, \rho\{skip\}\rangle)$ and $TR_2 = \mathbb{T}(\langle H_2, \top, \rho\{c\}\rangle \rightarrow^*_{FP} \langle H_2', \top, \rho\{skip\}\rangle)$

Since both executions will take equal number of steps in the high integrity context, therefore $|TR_1| = |TR_2|$.
By induction on $|TR_1|$
IH: $\forall 1 \leq i < p = |TR_1|.\ TR_1(i).H \underset{\rho_A}{\overset{\mathbb{L}}{\sim}} TR_2(i).H \wedge TR_1(i).PC \underset{\rho_A}{\overset{\mathbb{L}}{\sim}} TR_2(i).PC$
To prove: $TR_1(p).H \underset{\rho_A}{\overset{\mathbb{L}}{\sim}} TR_2(p).H \wedge TR_1(p).PC \underset{\rho_A}{\overset{\mathbb{L}}{\sim}} TR_2(p).PC$
Say in first execution it takes q steps to go from $TR_1(p-1)$ to $TR_1(p)$ and in second execution it takes r steps to go from $TR_r(p-1)$ to $TR_2(p)$. The following cases arise:

1. $q = 1$ and $r = 1$: By Lemma 25

2. $q > 1$ and $r = 1$: By Lemma 27 and Lemma 26

3. $q = 1$ and $r > 1$: By Lemma 27 and Lemma 26

4. $q > 1$ and $r > 1$: By Lemma 28

□

Definition 20 (Freedom from Confused Deputy Attack). *CDA-freedom-FP*$(\mathbb{E}_{\rho_A}, H, \rightarrow_{red}$
$) \triangleq$
$\forall PC, c_{\rho_A}, c'_{\rho_A}.$
$\langle H, PC, \mathbb{E}_{\rho_A}[c_{\rho_A}]\rangle \rightarrow^*_{red} \langle H_1, PC_1, P_1\rangle \implies$
$\forall r \in AIS.$
$\langle H, PC, \mathbb{E}_{\rho_A}[c'_{\rho_A}]\rangle \rightarrow^*_{red} \langle H_2, PC_2, P_2\rangle \implies H_1(r) = H_2(r)$
$\vee$
$\exists c''_{\rho_A}.\langle H, PC, \mathbb{E}_{\rho_A}[c''_{\rho_A}]\rangle \rightarrow^*_{red} \langle H_3, PC_3, P_3\rangle \wedge H_1(r) = H_3(r)$

Definition 21 (CDAF with endorsement). *CDA-freedom-FP-E*$(\mathbb{E}_{\rho_A}, H, \rightarrow_{red})$
$\triangleq$
Say $\mathbb{E}_{\rho_A} = \rho_1\{c_1\} \circ \ldots \rho_n\{c_n\}$
$1 \leq i, j \leq n.\ P_{ij} = \rho_i\{c_i\} \circ \ldots \rho_j\{c_j\}\ s.t$
$\forall\ i \leq k \leq j.P_k \neq \overline{\mathbb{P}} \implies$ *CDA-freedom-FP*$(P_{ij}, H, \rightarrow_{red})$

Definition 22 (Non-interference for Active Adversaries). *NI-A*$(\mathbb{E}_{\rho_A}, \rightarrow_{red})$. $\triangleq$
$\forall H_1, H_2, PC_1, PC_2, c_A, c'_A.$
$H_1 \underset{\rho_A}{\overset{\mathbb{L}}{\sim}} H_2 \wedge PC_1 \underset{\rho_A}{\overset{\mathbb{L}}{\sim}} PC_2$
$\langle H_1, PC_1, \mathbb{E}_{\rho_A}[c_A]\rangle \rightarrow^*_{red} \langle H'_1, PC'_1, P'_1\rangle \wedge$
$\langle H_2, PC_2, \mathbb{E}_{\rho_A}[c'_A]\rangle \rightarrow^*_{red} \langle H'_2, PC'_2, P'_2\rangle$
$\implies$
$H'_1 \underset{\rho_A}{\overset{\mathbb{L}}{\sim}} H'_2 \wedge PC'_1 \underset{\rho_A}{\overset{\mathbb{L}}{\sim}} PC'_2$

Definition 23 (*NI-A* with endorsement). *NI-A-E*$(\mathbb{E}_{\rho_A}, \rightarrow_{red}) \triangleq$
Say $\mathbb{E}_{\rho_A} = \rho_1\{c_1\} \circ \ldots \rho_n\{c_n\}$
$1 \leq i, j \leq n.\ P_{ij} = \rho_i\{c_i\} \circ \ldots \rho_j\{c_j\}\ s.t$
$\forall\ i \leq k \leq j.P_k \neq \overline{\mathbb{P}} \implies$ *NI-A*$(P_{ij}, \rightarrow_{red})$

Theorem 13 (*NI-A* $\implies$ *CDA-freedom-FP*). $\forall\ \mathbb{E}_{\rho_A}, H, \rho_A.$
NI-A$(\mathbb{E}_{\rho_A}, \rightarrow_{FP}) \implies$
CDA-freedom-FP$(\mathbb{E}_{\rho_A}, H, \rightarrow_{FP})$

Proof. We choose H_1 and H_2 in the definition of *NI-A* as the H for which we want to prove *CDA-freedom-FP*.
From *NI-A*, we know $H'_1 \underset{\rho_A}{\overset{\mathbb{L}}{\sim}} H'_2$.
$\forall r \in dom(H'_1)$, 2 cases arise :

1. $\beta(\rho_A) \not\geq_{\mathbb{L}} \mathbb{O}(r)$: $H'_1(r) = H_2(r)'$ from Definition 16.
   This satisfies first disjunct of *CDA-freedom-FP*.

2. $\beta(\rho_A) \geq_{\mathbb{L}} \mathbb{O}(r)$: Second disjunct of *CDA-freedom-FP* can be trivially satisfied.

□

Theorem 14 (*NI-A-E* $\implies$ *CDA-freedom-FP-E*)**.** $\forall\ \mathbb{E}_{\rho_A}, H, \rho_A$.
NI-A-E$(\mathbb{E}_{\rho_A}, \rightarrow_{FP}) \implies$
CDA-freedom-FP-E$(\mathbb{E}_{\rho_A}, H, \rightarrow_{FP})$

Proof. Directly from Theorem 13 □

Theorem 15 ($\rightarrow_{FP}$ guarantees *NI-A*)**.** $\forall\ \mathbb{E}_{\rho_A}$.
NI-A-E$(\mathbb{E}_{\rho_A}, \rightarrow_{FP})$

Proof. Expanding the definition of *NI-A-E*$(\mathbb{E}_{\rho_A}, \rightarrow_{FP})$, we get
Say $\mathbb{E}_{\rho_A} = \rho_1\{c_1\} \circ \ldots \rho_n\{c_n\}$
$1 \leq i, j \leq n$. $P_{ij} = \rho_i\{c_i\} \circ \ldots \rho_j\{c_j\}$ s.t
$\forall\ i \leq k \leq j.P_k \neq \overline{\mathbb{P}} \implies$ *NI-A*$(P_{ij}, \rightarrow_{FP})$
Now, say P_{ij} is of the form $\rho_1^{ij}\{c_1^{ij}\} \circ \ldots \rho_m^{ij}\{c_m^{ij}\}$
$\exists I \in \{1...m\}$. $\rho_I^{ij} = \rho_A$ then from Lemma 30 we get $H'_{(I-1)1} \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} H'_{(I-1)2}$ and $PC'_{(I-1)1} \overset{\mathbb{L}}{\underset{\rho_A}{\sim}} PC'_{(I-1)2}$
where $H'_{(I-1)1}$ and $H'_{(I-1)2}$ are the final heaps obtained after the execution of command in (I-1)th region
Similarly $PC'_{(I-1)1}$ and $PC'_{(I-1)2}$ are the final PC stacks obtained after the execution of command in (I-1)th region
For the Ith (attacker region), we get equivalence from Lemma 29
And again for the remaning regions from Lemma 30 again. □

5 Capability safety

5.1 Abstract Definitions

Definition 24 (Valid authority map)**.** *$auth_{\rho,\mathbb{O}} : \mathbb{H} \times \mathbb{T} \rightarrow \mathbb{A}$ is valid if $\forall H, c$.
$\langle H, c\rangle \xrightarrow{\rho}_C \langle H', c'\rangle \implies$*

1. *$acc_{\rho,\mathbb{O}}(H, c) \subseteq auth_{\rho,\mathbb{O}}(H, c)$ and*
2. *$auth_{\rho,\mathbb{O}}(H', c') \subseteq auth_{\rho,\mathbb{O}}(\rho, H, c)$*

Definition 25 (Capability system)**.** *A capability is defined by the following tuple:*

- *A set of capabilities, $\mathbb{C}$*
- *A function from capability to resources, $desg : \mathbb{C} \rightarrow R$*
- *A function from capability to its privileges, $priv : \mathbb{C} \rightarrow 2^{\{\mathbb{R},\mathbb{W}\}}$*
- *A function from command to capabilities, $tCap : \mathbb{T}_c \rightarrow 2^{\mathbb{C}}$*
- *A function from heap to capabilities, $hCap : \mathbb{H} \rightarrow 2^{\mathbb{C}}$*

- *A function from heap and capability to the set actions,* $cAuth_{\rho,\mathbb{O}} : \mathbb{H} \times \mathbb{C} \to 2^{\mathbb{A}}$

This tuple must satisfy the following conditions in order to be termed as a valid capability system:

1. *Basic conditions:* $\forall H \in \mathbb{H}, {}^{\nu}r \in \mathbb{C}$
   (a) $\forall c_1, c_2 \in \mathbb{T}.\ c_1 \sqsubseteq c_2 \implies tCap(c_1) \subseteq tCap(c_2)$
   (b) $desg(\kappa) \in res(H) \iff \kappa \in hCap(H)$
   (c) $(\kappa \notin hCap(H) \vee priv(\kappa) = \emptyset) \implies cAuth_{\rho,\mathbb{O}}(H,\kappa) = \emptyset$
   (d) $(\kappa \in hCap(H) \wedge priv(\kappa) \neq \emptyset) \implies \{desg(\kappa)\} \times priv(\kappa) \subseteq cAuth_{\rho,\mathbb{O}}(H,\kappa) \subseteq act(H)$
   (e) $\forall c \in \mathbb{T}.\ Wf(H,c) \implies tCap(c) \subseteq hCap(H)$

2. *Topology-only bound for* $cAuth_{\rho,\mathbb{O}}$*:* $\forall H \in \mathbb{H}, \kappa \in \mathbb{C}$.
   (a) $\mathbb{R} \in priv(\kappa) \implies cAuth_{\rho,\mathbb{O}}(H,\kappa) \subseteq \{(desg(\kappa)\} \times priv(\kappa) \cup \bigcup_{\kappa' \in C} cAuth_{\rho,\mathbb{O}}(H,\kappa')$
   where $C = tCap(H(desg(\kappa))) \cup \bigcup_{n} \{(\kappa \oplus n) | \kappa \oplus n \text{ is defined } \}$
   (b) $\mathbb{R} \notin priv(\kappa) \implies cAuth_{\rho,\mathbb{O}}(H,\kappa) \subseteq \{desg(\kappa)\} \times priv(\kappa) \cup \bigcup_{\kappa' \in C} cAuth_{\rho,\mathbb{O}}(H,\kappa')$
   where $C = \bigcup_{n} \{\kappa \oplus n) | \kappa \oplus n \text{ is defined } \}$

Definition 26 (Capability Safety)**.** *A capability system (as defined in Definition 25) is capability safe if* $\forall\ H, c$ *the following conditions hold:*

1. $auth_{\rho,\mathbb{O}}(H,c) = \bigcup_{{}^{\nu}r \in tCap(c)} cAuth_{\rho,\mathbb{O}}(H, {}^{\nu}r)$ *is a valid authority map*

2. $\langle H, c\rangle \xrightarrow{\varrho}_{C} \langle H', c'\rangle$ *and* $\forall \kappa \in hCap(H')$.
   (a) $acc_{\rho,\mathbb{O}}(H,c) \not\triangleright cAuth_{\rho,\mathbb{O}}(H,\kappa) \implies cAuth_{\rho,\mathbb{O}}(H',\kappa) = cAuth_{\rho,\mathbb{O}}(H,\kappa)$
   (b) $acc_{\rho,\mathbb{O}}(H,c) \triangleright cAuth_{\rho,\mathbb{O}}(H,\kappa) \implies cAuth_{\rho,\mathbb{O}}(H',\kappa) \subseteq cAuth_{\rho,\mathbb{O}}(H,\kappa) \cup \bigcup_{\kappa' \in \in tCap(c)} cAuth_{\rho,\mathbb{O}}(H,\kappa')$

5.2 Instantiating the definitions for Capability semantics with reference computations

Definition 27 (*act* for our system)**.** $act(H) \triangleq \{r \mid r \in dom(H)\} \times \{\mathbb{R}, \mathbb{W}\}$

Definition 28 (*res* for our system)**.** $res(H) \triangleq \{r \mid r \in dom(H)\}$

Definition 29 (*desg* for our system)**.** $desg\ {}^{\nu}r \triangleq r$

Definition 30 (*priv* for our system)**.** $priv\ {}^{\nu}r \triangleq \nu$ *where* $\nu \in \{\mathbb{R}, \mathbb{W}\}$

Definition 31 ($acc_{\rho,\mathbb{O}}$ for our system).

$$acc_{\rho,\mathbb{O}}\ H\ c \triangleq \begin{cases} accE_{\rho,\mathbb{O}}\ H\ e & c = \mathit{if}\ e\ \mathit{then}\ c_1\ \mathit{else}\ c_2 \\ accE_{\rho,\mathbb{O}}\ H\ e & c = \mathit{while}\ e\ \mathit{do}\ c_1 \\ accE_{\rho,\mathbb{O}}\ H\ e_1 \cup accE_{\rho,\mathbb{O}}\ H\ e_2 \cup \{(r,\mathbb{W})\} & c = e_1 := e_2 \wedge \langle H, e_1\rangle \Downarrow^{\rho}_{C}\ {}^{\mathbb{W}}r \\ acc_{\rho,\mathbb{O}}\ H\ c_1 & c = c_1; c_2 \end{cases}$$

Definition 32 ($accE_{\rho,\mathbb{O}}$ for our system).

$$accE_{\rho,\mathbb{O}}\ H\ e \triangleq \begin{cases} \emptyset & e = v \\ accE_{\rho,\mathbb{O}}\ e' \cup \{(r,\mathbb{R})\} & e = !e' \wedge \langle H, e'\rangle \Downarrow^{\rho}_{C}\ {}^{\mathbb{R}}r \\ accE_{\rho,\mathbb{O}}\ e_1 \cup accE_{\rho,\mathbb{O}}\ e_2 & e = e_1 \oplus e_2 \end{cases}$$

Definition 33 ($tCap$ for our system).

$$tCap\ c \triangleq \begin{cases} tCapExpr\ e \cup tCap\ c_1 \cup tCap\ c_2 & c = \mathit{if}\ e\ \mathit{then}\ c_1\ \mathit{else}\ c_2 \\ tCapExpr\ e \cup tCap\ c_1 & c = \mathit{while}\ e\ \mathit{do}\ c_1 \\ tCapExpr\ e_1 \cup tCapExpr\ e_2 & c = e_1 := e_2 \\ tCap\ c_1 \cup tCap\ c_2 & c = c_1; c_2 \\ \emptyset & c = \mathit{skip} \end{cases}$$

Definition 34 ($tCapExpr$ for our system).

$$tCapExpr\ e \triangleq \begin{cases} \emptyset & e = \mathit{tt}\ \mathit{or}\ \mathit{ff} \\ \{{}^{\nu}r\} & e = {}^{\nu}r \\ tCapExpr\ e' & e = !e' \\ tCapExpr\ e_1 \cup tCapExpr\ e_2 & e = e_1 \oplus e_2 \end{cases}$$

Definition 35 ($hCap$ for our system). $hCap\ H \triangleq \{{}^{\nu}r | r \in dom(H) \wedge \nu \in \{\mathbb{R}, \mathbb{W}\}\}$

Definition 36 ($cAuth_{\rho,\mathbb{O}}$ for our system).

$$cAuth_{\rho,\mathbb{O}}\ H\ {}^{\nu}r \triangleq \begin{cases} \{(r,\mathbb{W})\} \cup \bigcup_n cAuth_{\rho,\mathbb{O}}\ H\ {}^{\nu}(r \oplus n) & \nu = \mathbb{W} \wedge {}^{\nu}r \in hCap(H) \wedge \beta(\rho) \geq_{\mathbb{L}} \mathbb{O}(r) \\ \{(r,\mathbb{R})\} \cup \bigcup_n cAuth_{\rho,\mathbb{O}}\ H\ {}^{\nu}(r \oplus n) & \nu = \mathbb{R} \wedge {}^{\nu}r \in hCap(H) \wedge H(r) = {}^{\nu'}r' \\ \cup\ cAuth_{\rho,\mathbb{O}}\ \mathbb{O}\ {}^{\nu'}r' & \\ \{(r,\mathbb{R})\} \cup \bigcup_n cAuth_{\rho,\mathbb{O}}\ H\ {}^{\nu}(r \oplus n) & \nu = \mathbb{R} \wedge {}^{\nu}r \in hCap(H) \wedge H(r) \neq {}^{\nu'}r' \\ \emptyset & \mathit{otherwise} \end{cases}$$

Definition 37 ($\subseteq_E$ for our system).

$$e' \sqsubseteq_E e \triangleq \begin{cases} e' = v & e = v \\ e' \sqsubseteq_E e_1 & e = !e_1 \\ e' \sqsubseteq_E e_1 \vee & \\ e' \sqsubseteq_E e_2 \vee & e = e_1 \oplus e_2 \\ e' = e'_1 \oplus e'_2 \wedge e'_1 \sqsubseteq_E e_1 \wedge e'_2 \sqsubseteq_E e_2 & \end{cases}$$

Definition 38 ($\sqsubseteq_C$ for our system)**.**

$$c' \sqsubseteq_C c \triangleq \begin{cases} c' \sqsubseteq_C c_1 \vee & c = \textit{ if } e \textit{ then } c_1 \textit{ else } c_2 \\ c' \sqsubseteq_C c_2 \vee & \\ c' = \textit{ if } e' \textit{ then } c_1' \textit{ else } c_2' \textit{ s.t } e' \sqsubseteq_E e \wedge c_1' \sqsubseteq_C c_1 \wedge c_2' \sqsubseteq_C c_2 & \\ c' \sqsubseteq_C c_1 \vee & c = \textit{ while } e \textit{ do } c_1 \\ c' = \textit{ while } e' \textit{ do } c_1' \textit{ s.t } e' \sqsubseteq_E e \wedge c_1' \subseteq_C c_1 & \\ c' = e_1' := e_2' \textit{ s.t } e_1' \subseteq_E e_1 \wedge e_2' \sqsubseteq_E e_2 & c = \ e := e \\ c' \sqsubseteq_C c_1 \vee & c = c_1; c_2 \\ c' \sqsubseteq_C c_2 \vee & \\ c' = c_1'; c_2 \wedge c_1' \sqsubseteq_C c_1 \wedge c_1' \sqsubseteq_C c_1 & \end{cases}$$

5.3 Results

Lemma 31 (Basic condition 1a)**.** $\forall e', e'' \in \mathbb{T}_e.\ e' \sqsubseteq_E e'' \implies \textit{tCap}(e') \subseteq_E \textit{tCap}(e'')$

Proof. Proof by induction on e'':

1. v:
   An $e' \sqsubseteq_E v$ must be v itself, from Definition 37
   From Definition 34 $\textit{tCap}(e') \subseteq_E \textit{tCap}(e'')$

2. e_1:
   An $e' \sqsubseteq_E !e_1$ must be a subset of e_1, from Definition 37
   From IH and Definition 34 $\textit{tCap}(e') \subseteq_E \textit{tCap}(e'')$

3. $e_1 \oplus e_2$:
   Since $e' \sqsubseteq_E e_1 \oplus e_2$, 3 cases arise:

   (a) $e' \sqsubseteq_E e_1$: From IH and Definition 34

   (b) $e' \sqsubseteq_E e_2$: From IH and Definition 34

   (c) $e' = e_1' \oplus e_2' \wedge e_1' \sqsubseteq_E e_1 \wedge e_2' \sqsubseteq_E e_2$:
   IH1: $\forall e_1', e_1 \in \mathbb{T}_e.\ e_1' \sqsubseteq_E e_1 \implies \textit{tCap}(e_1') \subseteq_E \textit{tCap}(e_1)$
   IH2: $\forall e_2', e_2 \in \mathbb{T}_e.\ e_2' \sqsubseteq_E e_2 \implies \textit{tCap}(e_2') \subseteq_E \textit{tCap}(e_2)$
   From Definition 34, $\textit{tCapExpr}(e_1') \cup \textit{tCapExpr}(e_2') = \textit{tCapExpr}(e')$ $\subseteq_E \textit{tCapExpr}(e) = \textit{tCapExpr}(e_1) \cup \textit{tCapExpr}(e_2)$

$\square$

Theorem 16 (Basic condition 1b)**.** $\forall c_1, c_2 \in \mathbb{T}_c.\ c' \sqsubseteq c'' \implies \textit{tCap}(c') \subseteq \textit{tCap}(c'')$

Proof. By induction on c''

1. if e then c_1 else c_2:
   Since $c' \sqsubseteq c''$, 3 cases arise :

   (a) $c' \sqsubseteq_C c_1$:
   From IH and Definition 33

(b) $c' \sqsubseteq_C c_2$:
From IH and Definition 33

(c) $c' =$ if e' then c_1' else c_2' s.t $e' \sqsubseteq_E e \wedge c_1' \sqsubseteq_C c_1 \wedge c_2' \sqsubseteq_C c_2$:
From IH1, IH2, Lemma 31 and Definition 33.

2. while e do c:
Since $c' \sqsubseteq_C c''$, 2 cases arise :

   (a) $c' \sqsubseteq_C c_1$:
   From IH and Definition 33

   (b) $c' =$ while e' do c_1' s.t $e' \sqsubseteq_E e \wedge c_1' \sqsubseteq_C c_1$:
   From IH, Lemma 31 and Definition 33.

3. $e_1 := e_2$:
Since $c' \subseteq c''$, therefore, $c' = e_1' := e_2'$ s.t $e_1' \sqsubseteq_E e_1 \wedge e_2' \sqsubseteq_E e_2$.
From Lemma 31 and Definition 33

4. $c_1; c_2$:
Since $c' \sqsubseteq c''$. 3 cases arise:

   (a) $c' \sqsubseteq_C c_1$:
   From IH and Definition 33

   (b) $c' \sqsubseteq_C c_2$:
   From IH and Definition 33

   (c) $c' = c_1'; c_2 \wedge c_1' \sqsubseteq_C c_1 \wedge c_1' \sqsubseteq_C c_1$:
   From IH1, IH2 and Definition 33

□

Theorem 17 (Basic condition 2)**.** $\forall H \in \mathbb{H}, {}^{\nu}r \in \mathbb{C}.\ desg({}^{\nu}r) \in res(H) \iff {}^{\nu}r \in hCap(H)$

Proof. To prove: $\forall H \in \mathbb{H}, {}^{\nu}r \in \mathbb{C}.\ desg({}^{\nu}r) \in res(H) \implies {}^{\nu}r \in hCap(H)$
From Definition 29 and Definition 35 we know that $r \in dom(H)$. And since $dom(\nu) = \{\mathbb{R}, \mathbb{W}\}$ therefore ${}^{\nu}r \in hCap(H)$ from Definition 35

To prove: $\forall H \in \mathbb{H}, {}^{\nu}r \in \mathbb{C}.\ desg({}^{\nu}r) \in res(H) \Longleftarrow {}^{\nu}r \in hCap(H)$
Since ${}^{\nu}r \in hCap(H)$, therefore from Definition 35 we know that $r \in dom(H)$ and $\nu \in \{\mathbb{R}, \mathbb{W}\}$. This means from Definition 29 and Definition 28 that $desg({}^{\nu}r) \in res(H)$ □

Theorem 18 (Basic condition 3)**.** $\forall H \in \mathbb{H}, {}^{\nu}r \in \mathbb{C}.\ ({}^{\nu}r \notin hCap(H) \vee priv({}^{\nu}r) = \emptyset) \implies cAuth_{\rho,\mathbb{O}}(H, {}^{\nu}r) = \emptyset$

Proof. Case analyzing the premise $({}^{\nu}r \notin hCap(H) \vee priv({}^{\nu}r) = \emptyset)$:

1. ${}^{\nu}r \notin hCap(H)$: From Definition 36, $cAuth_{\rho,\mathbb{O}}(H, {}^{\nu}r) = \emptyset$

2. $priv(^{\nu}r) = \emptyset$: Give that $^{\nu}r \in \mathbb{C}$ ($^{\nu}r$ is a valid capability) therefore $\nu \in \{\mathbb{R}, \mathbb{W}\}$. Therefore, from Definition 30 $priv(^{\nu}r) \neq \emptyset$. And therefore this case cannot arise

$\square$

Theorem 19 (Basic condition 4)**.** $\forall H \in \mathbb{H}, {}^{\nu}r \in \mathbb{C}.\ ({}^{\nu}r \in hCap(H) \wedge priv({}^{\nu}r) \neq \emptyset) \implies \{desg({}^{\nu}r)\} \times priv({}^{\nu}r) \subseteq cAuth_{\rho,\mathbb{O}}(H, {}^{\nu}r) \subseteq act(H)$

Proof. To prove $\{desg({}^{\nu}r)\} \times priv({}^{\nu}r) \subseteq cAuth_{\rho,\mathbb{O}}(H, {}^{\nu}r)$
We know that ${}^{\nu}r \in hCap(H)$. Case analyzing on the ν:

1. $\nu = \mathbb{W}$: From Definition 36 we know that $cAuth_{\rho,\mathbb{O}}(H, {}^{\nu}r) = \{(r, \mathbb{W})\} \cup$ Some set. And $\{desg({}^{\nu}r)\} \times priv({}^{\nu}r)$ in this case is $(r, \mathbb{W})$. So, clearly, $\{desg({}^{\nu}r)\} \times priv({}^{\nu}r) \subseteq cAuth_{\rho,\mathbb{O}}(H, {}^{\nu}r)$

2. $\nu = \mathbb{R}$: In this case $cAuth_{\rho,\mathbb{O}}(H, {}^{\nu}r) = \{(r', \mathbb{R}) | r' \in dom(H)\} \supseteq \{desg({}^{\nu}r)\} \times priv({}^{\nu}r) = (r, \mathbb{R})$

To prove $cAuth_{\rho,\mathbb{O}}(H, {}^{\nu}r) \subseteq act(H)$
Case analyzing the ν:

1. $\nu = \mathbb{W}$: The largest $cAuth_{\rho,\mathbb{O}}(H, {}^{\nu}r)$ s.t $\nu = \mathbb{W} \wedge {}^{\nu}r \in hCap(H)$ would occur when $\forall n.\beta(\rho) \geq_{\mathbb{L}} \mathbb{O}(r \oplus n)$. And such largest $cAuth_{\rho,\mathbb{O}}(H, {}^{\nu}r)$ would be exactly $\{r \mid r \in dom(H)\} \times \{\mathbb{R}, \mathbb{W}\}$

2. $\nu = \mathbb{R}$: This case can produce read actions only and $cAuth_{\rho,\mathbb{O}}(H, {}^{\nu}r) = \{(r', \mathbb{R}) | r' \in dom(H)\} \subseteq \{r \mid r \in dom(H)\} \times \{\mathbb{R}, \mathbb{W}\}$

$\square$

Lemma 32. $\forall H \in \mathbb{H}, {}^{\nu}r \in \mathbb{C}.\ \forall e.\ Wf(H, e) \implies tCapExpr(e) \subseteq hCap(H)$

Proof. By induction on e:

1. tt, ff:
   From Definition 34 $tCapExpr(e) = \emptyset \subseteq hCap(H)$

2. ${}^{\nu}r$:
   From Definition 34 $tCapExpr(e) = \{{}^{\nu}r\} \subseteq hCap(H)$ because $Wf(H, e)$

3. $!e'$:
   IH: $tCapExpr(e') \subseteq hCap(H)$
   From Definition 34 and IH

4. $e_1 \oplus e_2$:
   IH1: $tCapExpr(e_1) \subseteq hCap(H)$
   IH2: $tCapExpr(e_2) \subseteq hCap(H)$
   From Definition 34 and IH1 and IH2

$\square$

Theorem 20 (Basic condition 5). $\forall H \in \mathbb{H}, {}^{\nu}r \in \mathbb{C}.\ \forall c \in \mathbb{T}.\ \mathit{Wf}(H, c) \implies \mathit{tCap}(c) \subseteq \mathit{hCap}(H)$

Proof. By induction on c:

1. if e then c_1 else c_2:
   From Definition 33 $\mathit{tCap}(c) = \mathit{tCapExpr}(e) \cup \mathit{tCap}(c_1) \cup \mathit{tCap}(c_2)$
   From Lemma 32, $\mathit{tCapExpr}(e) \subseteq \mathit{hCap}(H)$
   IH1: $\mathit{tCap}(c_1) \subseteq \mathit{hCap}(H)$
   IH2: $\mathit{tCap}(c_2) \subseteq \mathit{hCap}(H)$
   From Definition 34 and IH1 and IH2

2. while e do c':
   From Definition 33 $\mathit{tCap}(c) = \mathit{tCapExpr}(e) \cup \mathit{tCap}(c_1)$
   From Lemma 32, $\mathit{tCapExpr}(e) \subseteq \mathit{hCap}(H)$
   IH: $\mathit{tCap}(c_1) \subseteq \mathit{hCap}(H)$
   From Definition 34 and IH

3. $e_1 := e_2$:
   From Definition 33 $\mathit{tCap}(c) = \mathit{tCapExpr}(e_1) \cup \mathit{tCapExpr}(e_1)$
   From Lemma 32, $\mathit{tCapExpr}(e_1) \subseteq \mathit{hCap}(H)$
   Also from Lemma 32, $\mathit{tCapExpr}(e_2) \subseteq \mathit{hCap}(H)$

4. $c_1; c_2$:
   IH1: $\mathit{tCap}(c_1) \subseteq \mathit{hCap}(H)$
   IH2: $\mathit{tCap}(c_2) \subseteq \mathit{hCap}(H)$
   From Definition 34 and IH1 and IH2

$\square$

Theorem 21 (Topology-only bound). $\forall H \in \mathbb{H}, \kappa \in \mathbb{C}.$

1. $\mathbb{R} \in \mathit{priv}(\kappa) \implies \mathit{cAuth}_{\rho,\mathbb{O}}(H, \kappa) \subseteq \{(\mathit{desg}(\kappa)\} \times \mathit{priv}(\kappa) \cup \bigcup_{\kappa' \in C} \mathit{cAuth}_{\rho,\mathbb{O}}(H, \kappa')$
   where $C = \mathit{tCap}(H(\mathit{desg}(\kappa)) \cup \bigcup_{n} \{(\kappa \oplus n) | \kappa \oplus n \text{ is defined } \}$

2. $\mathbb{R} \notin \mathit{priv}(\kappa) \implies \mathit{cAuth}_{\rho,\mathbb{O}}(H, \kappa) \subseteq \{\mathit{desg}(\kappa)\} \times \mathit{priv}(\kappa) \cup \bigcup_{\kappa' \in C} \mathit{cAuth}_{\rho,\mathbb{O}}(H, \kappa')$
   where $C = \bigcup_{n} \{\kappa \oplus n) | \kappa \oplus n \text{ is defined } \}$

Proof. $\kappa = {}^{\nu}r$ (For our capability system ${}^{\nu}r$ are the valid capabilities)

1. Proving 1):
   Case analyzing on ${}^{\nu}r \in \mathit{hCap}(H)$

   (a) ${}^{\nu}r \notin \mathit{hCap}(H)$:
   In this case $\mathit{cAuth}_{\rho,\mathbb{O}}(H, {}^{\nu}r) = \emptyset$ (from Definition 36)

   (b) ${}^{\nu}r \in \mathit{hCap}(H)$:
   Since $\nu = \mathbb{R}$, so $\mathit{cAuth}_{\rho,\mathbb{O}}(H, {}^{\nu}r)$ can be:

i. $\{(r,\mathbb{R})\} \cup \bigcup_n cAuth_{\rho,\mathbb{O}}\ H\ {}^{\nu}(r \oplus n)$:
Now, C would be $\bigcup_n \{{}^{\nu}(r \oplus n)\}$ as $H(r) \neq^{-} r'$ leading to $tCap(H(r)) = \emptyset$. Thus, proving the result.

ii. $\{(r,\mathbb{R})\} \cup \bigcup_n cAuth_{\rho,\mathbb{O}}\ H\ {}^{\nu}(r \oplus n) \cup cAuth_{\rho,\mathbb{O}}\ \mathbb{O}\ {}^{\nu'}r'$:
In this case C would be $\bigcup_n \{{}^{\nu}(r \oplus n)\} \cup T$ where $T = tCap(H(desg({}^{\nu}r))$
From Definition 29 and given, $H(desg({}^{\nu}r)) =^{\nu'} r$. And from Definition 33 $tCap({}^{\nu'}r) = \{{}^{\nu'}r\}$. Thus, proving the result.

2. Proving 2):

Since ${}^{\nu}r \in \mathbb{C}$ and if $\mathbb{R} \notin priv({}^{\nu}r)$ then from Definition 30 it must be the case $priv({}^{\nu}r) = \mathbb{W}$:
2 cases arise:

(a) $\beta(\rho) \geq_{\mathbb{L}} \mathbb{O}(r)$:
In this case from Definition 36, $cAuth_{\rho,\mathbb{O}}(H, {}^{\nu}r) = \{(r,\mathbb{W})\} \cup \bigcup_n cAuth_{\rho,\mathbb{O}}\ H\ {}^{\nu}(r \oplus n)$
Since ${}^{\nu}r \in hCap(H) \wedge priv({}^{\nu}r) = \mathbb{W}$ therefore $C = \bigcup_n \{{}^{\nu}(r \oplus n)\}$ at least. Thus, proving the result.

(b) $\beta(\rho) \ngeq_{\mathbb{L}} \mathbb{O}(r)$:
In this case from Definition 36, $cAuth_{\rho,\mathbb{O}}(H, {}^{\nu}r) = \emptyset$. Thus, proving the result.

□

Lemma 33. $\forall \rho, \mathbb{O}, H, e.$
$accE_{\rho,\mathbb{O}}(H, e) \subseteq \bigcup_{{}^{\nu}r \in tCapExpr(e)} cAuth_{\rho,\mathbb{O}}(H, {}^{\nu}r)$

Proof. Proof by induction on e:

1. $e = v$:
From Definition 32, $accE_{\rho,\mathbb{O}}(H, v) = \emptyset$
So, from Definition 36 the required is proved.

2. $e = !e'$:
IH: $accE_{\rho,\mathbb{O}}(H, e') \subseteq \bigcup_{{}^{\nu}r \in tCapExpr(e')} cAuth_{\rho,\mathbb{O}}(H, {}^{\nu}r)$
From Definition 32, $accE_{\rho,\mathbb{O}}(H, e) = accE_{\rho,\mathbb{O}}(H, e') \cup \{(r',\mathbb{R})\}$ where $\langle H, e' \rangle \Downarrow^{\rho}_{C}{}^{\mathbb{R}} r'$
$\{(r',\mathbb{R})\}$ is already included in $\bigcup_{{}^{\nu}r \in tCapExpr(e)} cAuth_{\rho,\mathbb{O}}(H, {}^{\nu}r)$, proof by induction on $\langle H, e' \rangle \Downarrow^{\rho}_{C}{}^{\mathbb{R}} r'$

(a) C-val:
Since e' evaluates to ${}^{\mathbb{R}}r'$. Therefore, $e' = {}^{\mathbb{R}}r'$
In this case ${}^{\mathbb{R}}r'$ is included in $tCapExpr(e)$ and hence in $\bigcup_{{}^{\nu}r \in tCapExpr(e)} cAuth_{\rho,\mathbb{O}}(H, {}^{\nu}r)$
by Definition 36.

(b) C-deref:
From 2nd case of Definition 36 again ${}^{\mathbb{R}}r'$ is included in $\bigcup_{{}^{\nu}r \in tCapExpr(e)} cAuth_{\rho,\mathbb{O}}(H, {}^{\nu}r)$

(c) C-refComp:
${}^{\mathbb{R}}r'$ is included in $\bigcup_{{}^{\nu}r \in tCapExpr(e)} cAuth_{\rho,\mathbb{O}}(H, {}^{\nu}r)$ from 2nd or 3rd case of Defintion 36

3. $e = e_1 \oplus e_2$:
IH1: $accE_{\rho,\mathbb{O}}(H, e_1) \subseteq \bigcup_{{}^{\nu}r \in tCapExpr(e_1)} cAuth_{\rho,\mathbb{O}}(H, {}^{\nu}r)$
IH2: $accE_{\rho,\mathbb{O}}(H, e_2) \subseteq \bigcup_{{}^{\nu}r \in tCapExpr(e_2)} cAuth_{\rho,\mathbb{O}}(H, {}^{\nu}r)$
Therefore from Definition 32 and Definition 36

□

Lemma 34. $\forall \rho, \mathbb{O}, H, e.$
$\langle H, e\rangle \Downarrow^{\rho}_{C} {}^{\nu}r \wedge$
$cAuth_{\rho,\mathbb{O}}(H, {}^{\nu}r) \subseteq \bigcup_{{}^{\nu'}r' \in tCapExpr(e)} cAuth_{\rho,\mathbb{O}}(H, {}^{\nu'}r')$

Proof. By induction on $\Downarrow^{\rho}_{C}$:

1. C-val:
   Directly from Definition 36

2. C-deref:
   Say $e = !e'$, $\langle H, e'\rangle \Downarrow^{\rho}_{C} {}^{\mathbb{R}}r_r$ and $H(r_r) = {}^{\nu}r$
   IH: $cAuth_{\rho,\mathbb{O}}(H, {}^{\mathbb{R}}r_r) \subseteq \bigcup_{{}^{\nu'}r' \in tCapExpr(e')} cAuth_{\rho,\mathbb{O}}(H, {}^{\nu'}r')$
   Since $H(r_r) = {}^{\nu}r$ therefore $cAuth_{\rho,\mathbb{O}}(H, {}^{\mathbb{R}}r_r)$ must already include (r, ν) from Definition 36.

3. C-refComp:
   Say $e = e_1 \oplus e_2$, $\langle H, e_1\rangle \Downarrow^{\rho}_{C} {}^{\nu}r_1$ and $\langle H, e_2\rangle \Downarrow^{\rho}_{C} n$ s.t ${}^{\nu}r_1 \oplus n =^{\nu} r_2$
   IH: $cAuth_{\rho,\mathbb{O}}(H, {}^{\nu}r_1) \subseteq \bigcup_{{}^{\nu'}r' \in tCapExpr(e_1)} cAuth_{\rho,\mathbb{O}}(H, {}^{\nu'}r')$
   Since $\langle H, e\rangle \Downarrow^{\rho}_{C} {}^{\nu}r_2$ therefore from C-refComp and Definition 36 $cAuth_{\rho,\mathbb{O}}(H, {}^{\mathbb{R}}r_1)$ must already include (r_2, ν)
   And hence, $cAuth_{\rho,\mathbb{O}}(H, {}^{\nu}r_2) \subseteq \bigcup_{{}^{\nu'}r' \in tCapExpr(e)} cAuth_{\rho,\mathbb{O}}(H, {}^{\nu'}r')$

$\square$

Theorem 22 (Capability safety)**.** *A capability system (as defined in Definition 25) is capability safe if* $\forall\ H, c$ *the following conditions hold:*

1. $auth_{\rho,\mathbb{O}}(H, c) = \bigcup_{^\nu r \in tCap(c)} cAuth_{\rho,\mathbb{O}}(H, {}^\nu r)$ *is a valid authority map*

2. $\langle H, c\rangle \xrightarrow{\rho}_C \langle H', c'\rangle$ *and* $\forall^\nu r \in hCap(H')$.

   (a) $acc_{\rho,\mathbb{O}}(H, c) \not\triangleright cAuth_{\rho,\mathbb{O}}(H, {}^\nu r) \implies cAuth_{\rho,\mathbb{O}}(H', {}^\nu r) = cAuth_{\rho,\mathbb{O}}(H, {}^\nu r)$

   (b) $acc_{\rho,\mathbb{O}}(H, c) \triangleright cAuth_{\rho,\mathbb{O}}(H, {}^\nu r) \implies cAuth_{\rho,\mathbb{O}}(H', {}^\nu r) \subseteq cAuth_{\rho,\mathbb{O}}(H, {}^\nu r)$ $\cup \bigcup_{^{\nu'} r' \in tCap(c)} cAuth_{\rho,\mathbb{O}}(H, {}^{\nu'} r')$

Proof. 1. To prove (1) we need to show 2 things:

(a) $acc_{\rho,\mathbb{O}}(H, c) \subseteq auth_{\rho,\mathbb{O}}(H, c)$: Since $auth_{\rho,\mathbb{O}}(H, c) = \bigcup_{^\nu r \in tCap(c)} cAuth_{\rho,\mathbb{O}}(H, {}^\nu r)$ therefore we need to show that $acc_{\rho,\mathbb{O}}(H, c) \subseteq \bigcup_{^\nu r \in tCap(c)} cAuth_{\rho,\mathbb{O}}(H, {}^\nu r)$

By induction on c

i. $c =$ if e then c_1 else c_2:
From Lemma 33 we know that $acc_{\rho,\mathbb{O}}(H, e) \subseteq \bigcup_{^\nu r \in tCap(e)} cAuth_{\rho,\mathbb{O}}(H, {}^\nu r)$
From Definition 33, $tCap(c) = tCapExpr(e) \cup tCap(c_1) \cup tCap(c_2)$
Therefore from Definition 31, $acc_{\rho,\mathbb{O}}(H, c) \subseteq \bigcup_{^\nu r \in tCap(c)} cAuth_{\rho,\mathbb{O}}(H, {}^\nu r)$

ii. $c =$ while e do c_1:
From Lemma 33 we know that $acc_{\rho,\mathbb{O}}(H, e) \subseteq \bigcup_{^\nu r \in tCap(e)} cAuth_{\rho,\mathbb{O}}(H, {}^\nu r)$
From Definition 33, $tCap(c) = tCapExpr(e) \cup tCap(c_1)$
Therefore from Definition 31, $acc_{\rho,\mathbb{O}}(H, c) \subseteq \bigcup_{^\nu r \in tCap(c)} cAuth_{\rho,\mathbb{O}}(H, {}^\nu r)$

iii. $c = e_1 := e_2$:
From Lemma 33 we know that $acc_{\rho,\mathbb{O}}(H, e_1) \subseteq \bigcup_{^\nu r \in tCap(e_1)} cAuth_{\rho,\mathbb{O}}(H, {}^\nu r)$
Simialrly from Lemma 33 we know that $acc_{\rho,\mathbb{O}}(H, e_2) \subseteq \bigcup_{^\nu r \in tCap(e_2)} cAuth_{\rho,\mathbb{O}}(H, {}^\nu r)$
From Definition 33, $tCap(c) = tCapExpr(e) \cup tCapExpr(e_1)$
Therefore from Definition 31, $acc_{\rho,\mathbb{O}}(H, c) \subseteq \bigcup_{^\nu r \in tCap(c)} cAuth_{\rho,\mathbb{O}}(H, {}^\nu r)$

iv. $c = c_1; c_2$:
IH1: $acc_{\rho,\mathbb{O}}(H, c_1) \subseteq \bigcup_{^\nu r \in tCap(c_1)} cAuth_{\rho,\mathbb{O}}(H, {}^\nu r)$
IH2: $acc_{\rho,\mathbb{O}}(H, c_2) \subseteq \bigcup_{^\nu r \in tCap(c_2)} cAuth_{\rho,\mathbb{O}}(H, {}^\nu r)$
From Definition 33, $tCap(c) = tCap(c_1) \cup tCap(c_2)$
Therefore from Definition 31, $acc_{\rho,\mathbb{O}}(H, c) \subseteq \bigcup_{^\nu r \in tCap(c)} cAuth_{\rho,\mathbb{O}}(H, {}^\nu r)$

(b) $auth_{\rho,\mathbb{O}}(H'', c'') \subseteq auth_{\rho,\mathbb{O}}(H, c)$
Induction on the execution step:

i. C-if:
$H'' = H$ and $c'' = c_1$
Since $tCap(c_1) \subseteq tCap(c)$ from Definition 33, therefore $\bigcup_{{}^{\nu}r \in tCap(c'')} cAuth_{\rho,\mathbb{O}}(H, {}^{\nu}r)$
$\subseteq \bigcup_{{}^{\nu}r \in tCap(c)} cAuth_{\rho,\mathbb{O}}(H, {}^{\nu}r)$

ii. C-else:
$H'' = H$ and $c'' = c_2$
Since $tCap(c') \subseteq tCap(c)$ from Definition 33, therefore $\bigcup_{{}^{\nu}r \in tCap(c'')} cAuth_{\rho,\mathbb{O}}(H, {}^{\nu}r)$
$\subseteq \bigcup_{{}^{\nu}r \in tCap(c)} cAuth_{\rho,\mathbb{O}}(H, {}^{\nu}r)$

iii. C-while 1:
$H'' = H$ and $c'' = c_1; \text{while } e \text{ do } c_1$
Since $tCap(c') \subseteq tCap(c)$ from Definition 33, therefore $\bigcup_{{}^{\nu}r \in tCap(c'')} cAuth_{\rho,\mathbb{O}}(H, {}^{\nu}r)$
$\subseteq \bigcup_{{}^{\nu}r \in tCap(c)} cAuth_{\rho,\mathbb{O}}(H, {}^{\nu}r)$

iv. C-while 2:
$H'' = H$ and $c'' = skip$
Since $tCap(c') \subseteq tCap(c)$ from Definition 33, therefore $\bigcup_{{}^{\nu}r \in tCap(c'')} cAuth_{\rho,\mathbb{O}}(H, {}^{\nu}r)$
$\subseteq \bigcup_{{}^{\nu}r \in tCap(c)} cAuth_{\rho,\mathbb{O}}(H, {}^{\nu}r)$

v. C-assign:
$c'' = skip$
Since $tCap(c'') = \emptyset$ from Definition 33, therefore $\bigcup_{{}^{\nu}r \in tCap(c'')} cAuth_{\rho,\mathbb{O}}(H'', {}^{\nu}r)$
$\subseteq \bigcup_{{}^{\nu}r \in tCap(c)} cAuth_{\rho,\mathbb{O}}(H, {}^{\nu}r)$

vi. C-seq 1:
$c'' = c_1'; c_2$
IH: $\bigcup_{{}^{\nu}r \in tCap(c_1')} cAuth_{\rho,\mathbb{O}}(H'', {}^{\nu}r) \subseteq \bigcup_{{}^{\nu}r \in tCap(c_1)} cAuth_{\rho,\mathbb{O}}(H, {}^{\nu}r)$
Therefore, $\bigcup_{{}^{\nu}r \in tCap(c_1' \cup c_2)} cAuth_{\rho,\mathbb{O}}(H'', {}^{\nu}r) \subseteq \bigcup_{{}^{\nu}r \in tCap(c_1 \cup c_2)} cAuth_{\rho,\mathbb{O}}(H, {}^{\nu}r)$

vii. C-seq 2:
$c = skip; c_1$ and $c'' = c_1$. Also, $H'' = H$
Since $tCap(c) = tCap(c')$ from Definition 33.
Therefore, $\bigcup_{{}^{\nu}r \in tCap(c'')} cAuth_{\rho,\mathbb{O}}(H, {}^{\nu}r) \subseteq \bigcup_{{}^{\nu}r \in tCap(c)} cAuth_{\rho,\mathbb{O}}(H, {}^{\nu}r)$

2. To prove (2):

We know that one of the 2 cases can arise:

(a) $acc_{\rho,\mathbb{O}}(H,c) \not\triangleright cAuth_{\rho,\mathbb{O}}(H, {}^{\nu}r)$:
In this case we need to prove $cAuth_{\rho,\mathbb{O}}(H', {}^{\nu}r) = cAuth_{\rho,\mathbb{O}}(H, {}^{\nu}r)$

Case $\nu = \mathbb{R}$
By induction on $\xrightarrow{\rho}_C$

i. C-if, C-else, C-while 1, C-while 2, C-seq 2:
$H' = H$
Therefore, $cAuth_{\rho,\mathbb{O}}(H', {}^{\nu}r) = cAuth_{\rho,\mathbb{O}}(H, {}^{\nu}r)$

ii. C-assign:
From Definition 31 we know that there exists a $(r', \mathbb{W}) \in acc_{\rho,\mathbb{O}}(H,c)$.
Given this, no matter which ${}^{\mathbb{R}}r$ we choose $cAuth_{\rho,\mathbb{O}}(H, {}^{\mathbb{R}}r)$ will include $r', \mathbb{R}$ in it (from Definition 36).
Hence $acc_{\rho,\mathbb{O}}(H,c) \triangleright cAuth_{\rho,\mathbb{O}}(H, {}^{\nu}r)$ and this case cannot arise.

iii. C-seq 1:
From IH

Case $\nu = \mathbb{W}$
By induction on $\xrightarrow{\rho}_C$

i. C-if, C-else, C-while 1, C-while 2, C-seq 2:
$H' = H$
Therefore, $cAuth_{\rho,\mathbb{O}}(H', {}^{\nu}r) = cAuth_{\rho,\mathbb{O}}(H, {}^{\nu}r)$

ii. C-assign:
In the $\mathbb{W}$ case $cAuth_{\rho,\mathbb{O}}(H, {}^{\nu}r)$ can have all the possible $\mathbb{W}$ actions for the principal ρ and under authority map $\mathbb{O}$.
According to Definition 36 the only dependence on heap is ${}^{\nu}r \in hCap(H)$, besides this the values of the heap doesnt matter. And since $dom(H) = dom(H')$ thus $cAuth_{\rho,\mathbb{O}}(H', {}^{\nu}r) = cAuth_{\rho,\mathbb{O}}(H, {}^{\nu}r)$

iii. C-seq 1:
From IH

(b) $acc_{\rho,\mathbb{O}}(H,c) \triangleright cAuth_{\rho,\mathbb{O}}(H, {}^{\nu}r)$:
In this case we need to prove $cAuth_{\rho,\mathbb{O}}(H', {}^{\nu}r) \subseteq cAuth_{\rho,\mathbb{O}}(H, {}^{\nu}r) \cup \bigcup_{{}^{\nu'}r' \in tCap(c)} cAuth_{\rho,\mathbb{O}}(H, {}^{\nu'}r')$

Since $acc_{\rho,\mathbb{O}}(H,c) \triangleright cAuth_{\rho,\mathbb{O}}(H, {}^{\nu}r)$ so ν must be $\mathbb{R}$ otherwise $acc_{\rho,\mathbb{O}}(H,c) \not\triangleright cAuth_{\rho,\mathbb{O}}(H, {}^{\nu}r)$ (defintion of $\triangleright$ and Definition 36).
By induction on $\xrightarrow{\rho}_C$:

i. C-if, C-else, C-while 1, C-while 2, C-seq 2:
$H' = H$. Therefore, $auth_{\rho,\mathbb{O}}(H, {}^{\nu}r) = auth_{\rho,\mathbb{O}}(H', {}^{\nu}r)$
Therefore, $auth_{\rho,\mathbb{O}}(H', {}^{\nu}r) \subseteq auth_{\rho,\mathbb{O}}(H, {}^{\nu}r) \cup \bigcup_{{}^{\nu'}r' \in tCap(c)} cAuth_{\rho,\mathbb{O}}(H, {}^{\nu'}r')$

ii. C-assign:
5 cases arise:

A. $H'(r) = H(r) = {}^{\nu_1}r_1$
$cAuth_{\rho,\mathbb{O}}(H', {}^{\nu}r) = cAuth_{\rho,\mathbb{O}}(H, {}^{\nu}r)$
Therefore, $auth_{\rho,\mathbb{O}}(H', {}^{\nu}r) \subseteq auth_{\rho,\mathbb{O}}(H, {}^{\nu}r) \cup \bigcup_{{}^{\nu'}r' \in tCap(c)} cAuth_{\rho,\mathbb{O}}(H, {}^{\nu'}r')$

B. $H'(r) = {}^{\nu_1}r_1$ and $H(r) = (v_2 \neq {}^{\nu_2}r_2)$
2 cases arise

- $\nu_1 = \mathbb{R}$:
$cAuth_{\rho,\mathbb{O}}(H', {}^{\nu}r) = cAuth_{\rho,\mathbb{O}}(H, {}^{\nu}r)$ by Definition 36.
Therefore, $auth_{\rho,\mathbb{O}}(H', {}^{\nu}r) \subseteq auth_{\rho,\mathbb{O}}(H, {}^{\nu}r) \cup \bigcup_{{}^{\nu'}r' \in tCap(c)} cAuth_{\rho,\mathbb{O}}(H, {}^{\nu'}r')$
- $\nu_1 = \mathbb{W}$:
$cAuth_{\rho,\mathbb{O}}(H', {}^{\nu}r)$ would also contain the authority to write over r_1 and everything that can be computed via r_1 under ρ and $\mathbb{O}$.
This is already upper bounded by $\bigcup_{{}^{\nu'}r' \in tCap(e_2)} cAuth_{\rho,\mathbb{O}}(H, {}^{\nu'}r')$ from Lemma 34 and hence by $\bigcup_{{}^{\nu'}r' \in tCap(c)} cAuth_{\rho,\mathbb{O}}(H, {}^{\nu'}r')$

C. $H'(r) = (v_2 \neq {}^{\nu_2}r_2)$ and $H(r) = {}^{\nu_1}r_1$
2 cases arise

- $\nu_1 = \mathbb{R}$:
$cAuth_{\rho,\mathbb{O}}(H', {}^{\nu}r) = cAuth_{\rho,\mathbb{O}}(H, {}^{\nu}r)$ by Definition 36.
Therefore, $auth_{\rho,\mathbb{O}}(H', {}^{\nu}r) \subseteq auth_{\rho,\mathbb{O}}(H, {}^{\nu}r) \cup \bigcup_{{}^{\nu'}r' \in tCap(c)} cAuth_{\rho,\mathbb{O}}(H, {}^{\nu'}r')$
- $\nu_1 = \mathbb{W}$:
$cAuth_{\rho,\mathbb{O}}(H', {}^{\nu}r)$ is already a subset of $cAuth_{\rho,\mathbb{O}}(H, {}^{\nu}r)$ from Definition 36.
Therefore, $auth_{\rho,\mathbb{O}}(H', {}^{\nu}r) \subseteq auth_{\rho,\mathbb{O}}(H, {}^{\nu}r) \cup \bigcup_{{}^{\nu'}r' \in tCap(c)} cAuth_{\rho,\mathbb{O}}(H, {}^{\nu'}r')$

D. $H'(r) = (v_2 \neq {}^{\nu_2}r_2)$ and $H(r) = (v_1 \neq {}^{\nu_1}r_1)$
$cAuth_{\rho,\mathbb{O}}(H', {}^{\nu}r) = cAuth_{\rho,\mathbb{O}}(H, {}^{\nu}r)$ by Definition 36.
Therefore, $auth_{\rho,\mathbb{O}}(H', {}^{\nu}r) \subseteq auth_{\rho,\mathbb{O}}(H, {}^{\nu}r) \cup \bigcup_{{}^{\nu'}r' \in tCap(c)} cAuth_{\rho,\mathbb{O}}(H, {}^{\nu'}r')$

E. $H'(r) = {}^{\nu_1}r_1$ and $H(r) = {}^{\nu_2}r_2$:
4 cases arise:

- $\nu_1 = \mathbb{R}$ and $\nu_2 = \mathbb{R}$:
$cAuth_{\rho,\mathbb{O}}(H', {}^{\nu}r) = cAuth_{\rho,\mathbb{O}}(H, {}^{\nu}r)$ by Definition 36.
Therefore, $auth_{\rho,\mathbb{O}}(H', {}^{\nu}r) \subseteq auth_{\rho,\mathbb{O}}(H, {}^{\nu}r) \cup \bigcup_{{}^{\nu'}r' \in tCap(c)} cAuth_{\rho,\mathbb{O}}(H, {}^{\nu'}r')$
- $\nu_1 = \mathbb{W}$ and $\nu_2 = \mathbb{R}$:
$cAuth_{\rho,\mathbb{O}}(H', {}^{\nu}r)$ would also contain the authority to write over r_1 and everything that can be computed via r_1 under ρ and $\mathbb{O}$.
This is already upper bounded by $\bigcup_{{}^{\nu'}r' \in tCap(e_2)} cAuth_{\rho,\mathbb{O}}(H, {}^{\nu'}r')$

from Lemma 34 and hence by
$\bigcup_{{}^{\nu'}r' \in tCap(c)} cAuth_{\rho,\mathbb{O}}(H, {}^{\nu'}r')$

- $\nu_1 = \mathbb{R}$ and $\nu_2 = \mathbb{W}$:
  $cAuth_{\rho,\mathbb{O}}(H', {}^{\nu}r)$ is already a subset of $cAuth_{\rho,\mathbb{O}}(H, {}^{\nu}r)$ from Definition 36.
  And hence from IH1, $auth_{\rho,\mathbb{O}}(H', {}^{\nu}r) \subseteq auth_{\rho,\mathbb{O}}(H, {}^{\nu}r) \cup \bigcup_{{}^{\nu'}r' \in tCap(c)} cAuth_{\rho,\mathbb{O}}(H, {}^{\nu'}r')$
- $\nu_1 = \mathbb{W}$ and $\nu_2 = \mathbb{W}$:
  Since $H'(r) = {}^{\mathbb{W}}r_1$. Therefore, from Lemma 2 $\beta(\rho) \geq_{\mathbb{L}} \mathbb{O}(r_1)$.
  And from Definition 36, we know that $(r_1, \mathbb{W})$ is in $cAuth_{\rho,\mathbb{O}}(H, {}^{\nu}r)$ for all $r_1 \in dom(H)$
  Therefore, $cAuth_{\rho,\mathbb{O}}(H', {}^{\nu}r) = cAuth_{\rho,\mathbb{O}}(H, {}^{\nu}r)$

iii. C-seq 1:
IH: $auth_{\rho,\mathbb{O}}(H', {}^{\nu}r) \subseteq auth_{\rho,\mathbb{O}}(H, {}^{\nu}r) \cup \bigcup_{{}^{\nu'}r' \in tCap(c)} cAuth_{\rho,\mathbb{O}}(H, {}^{\nu'}r')$
Therefore from IH we get
$auth_{\rho,\mathbb{O}}(H', {}^{\nu}r) \subseteq auth_{\rho,\mathbb{O}}(H, {}^{\nu}r) \cup \bigcup_{{}^{\nu'}r' \in tCap(c)} cAuth_{\rho,\mathbb{O}}(H, {}^{\nu'}r')$

□