Automatic Computation of Static Variable Permissions

John C. Reynolds
Carnegie Mellon University

MFPS, May 28, 2011

Research partially supported by National Science Foundation Grant
CCF-0916808
Introduction

We will describe a procedure for inferring permissions for the proof system described by Uday S. Reddy in “Syntactic Control of Interference for Concurrent Separation Logic” (presented earlier at this conference).

Given a purported proof in Reddy’s formalism in which the variable permissions have been erased, our goal is to determine if there is an assignment of permissions that will give a valid proof.
Contexts

Reddy uses contexts such as

\[
\sum_{x^p, y^p, z^p} | \Gamma \vdash r_1(x^{p_1}, y^{p_1}): R_1, r_2(x^{p_2}, z^{p_2}): R_2 \vdash \cdots
\]

(We assume that the contexts are in normal form.)

We introduce the set

\[
\text{Owners} = \text{Resources} \cup \{\text{self}\},
\]

and treat \( \Sigma \) as a list associated with \text{self}. We also break out the resource invariants as a separate part of the context:

\[
\text{self}(x^p, y^p, z^p), r_1(x^{p_1}, y^{p_1}), r_2(x^{p_2}, z^{p_2}) | \Gamma \\
\vdash r_1: R_1, r_2: R_2 \vdash \cdots
\]

Then we transpose the \((\text{Owner}, \text{Variable})\)-matrix to bring the variables to the outside:

\[
\Delta \vdash x(\text{self}^{p_1}, r_1^{p_1}, r_2^{p_2}), y(\text{self}^{p_2}, r_1^{p_1}), z(\text{self}^{p_2}, r_2^{p_2}) | \Gamma \\
\vdash r_1: R_1, r_2: R_2 \vdash \cdots
\]
Contexts (continued)

We limit our development to fractional permissions, which are real numbers in the set

$$\text{Perms} = \{ p \mid 0 < p \leq 1 \}.$$  

However, rather than regarding a context $\Delta$ as a partial function into permissions, we will extend it to a total function by filling in the missing permissions with the nonpermission 0. Thus a permission context is a function

$$\Delta: \text{Vars} \rightarrow \text{Owners} \rightarrow \text{Perms} \cup \{0\},$$

such that

$$\sum_{o \in \text{Owners}} \Delta v o \leq 1.$$  

We assume that Vars and Owners are finite sets.
Judgements

The judgements used by Reddy:

\[ \Sigma \vdash E \text{ Exp} \quad \Sigma \vdash P \text{ Assert} \]
\[ \Sigma \vdash x \text{ Var} \quad \Sigma|\Gamma \vdash \{P\} C \{Q\}. \]

become, with our altered view of contexts:

\[ \Delta|\Upsilon \vdash E \text{ Exp} \quad \Delta|\Upsilon \vdash P \text{ Assert} \quad \text{(passive)} \]
\[ \Delta|\Upsilon \vdash x \text{ Var} \quad \Delta|\Upsilon \vdash \{P\} C \{Q\}. \quad \text{(active)} \]

To deal with the Rule of Consequence, we will also need a passive judgement that an assertion is valid:

\[ \Delta|\Upsilon \vdash P \text{ Valid}. \quad \text{(passive)} \]

*Passive* judgements are those which only describe the reading of variables, while *active* judgements may describe writing as well. (Our presentation is simplified by using the same form of context for all judgements.)

A *pre-judgement* has the form

\[ \Upsilon \vdash E \text{ Exp} \quad \Upsilon \vdash P \text{ Assert} \quad \Upsilon \vdash P \text{ Valid} \quad \text{(passive)} \]
\[ \Upsilon \vdash x \text{ Var} \quad \Upsilon \vdash \{P\} C \{Q\}. \quad \text{(active)} \]
Rules

The *rules* of (our slight modification of) Reddy’s logic are schemas of the form

\[
\begin{array}{c}
P_1 \quad \cdots \quad P_k \\
\hline
C
\end{array}
\]

where the premisses \(P_i\) and the conclusion \(C\) are schematic judgements.

(An instance of) a *pre-rule* is obtained from (an instance of) a rule by deleting the permission contexts.
Trees and Proofs

A tree consists of a finite set \( \text{Nodes} \), a node \( \text{root} \in \text{Nodes} \), and a function \( \text{parents} \in \text{Nodes} \rightarrow (\text{Nodes}^*) \), satisfying conditions that insure reachability from root, and the absence of cycles and common ancestors.

A proof (pre-proof) of shape \( \langle \text{Nodes}, \text{root}, \text{parents} \rangle \) is a node-indexed family \( \langle J_n \rangle \) of judgements (pre-judgements) such that, for each node \( n \) with parents \( n_1, \ldots, n_k \),

\[
\begin{array}{c}
J_{n_1} \\
\vdots \\
J_{n_k} \\
\hline \\
J_n
\end{array}
\]

is an instance of a rule (pre-rule).

We say that a node \( n \) is passive (active) if \( J_n \) is passive (active).
Erasure and Extension

If a pre-judgement, pre-rule instance, or pre-proof $X^0$ is obtained from a judgement, rule instance, or proof $X$ by deleting all permission contexts, we say that $X_0$ is the erasure of $X$, or that $X$ erases to $X^0$.

If a pre-proof $P^0$ is obtained from a proof $P$ by deleting all permission contexts $\Delta_n$, we say that $P$ extends $P_0$ with the node-indexed family $\langle \Delta_n \rangle$ of contexts.
Passive Rules

Reddy’s rules with passive conclusions will be replaced by two premiss-free rules with side conditions:

\[ \Delta | \gamma \vdash E \text{ Exp} \quad \text{where } \forall v \in \text{FV}(E). \Delta v \text{ self} > 0 \]

\[ \Delta | \gamma \vdash P \text{ Assert} \quad \text{where } \forall v \in \text{FV}(P). \Delta v \text{ self} > 0 \]

\[ \Delta | \gamma \vdash P \text{ Valid} \quad \text{where } \forall v \in \text{FV}(P). \Delta v \text{ self} > 0 \]

and \( P \) is a valid assertion.

where \( \text{FV}(X) \) denotes the set of free variables of \( X \).

Write Proofs

A \emph{write-proof} is a proof in which the side conditions \( \forall v \in \text{FV}(X). \Delta v \text{ self} > 0 \) of the passive rules are ignored, so that the permissions needed for variable reading are not checked.
The Plot (Phase I)

Given a pre-proof $P^0$, our algorithm should produce a proof that extends $P^0$, if such a proof exists. We assume that the pre-proof $P^0$ and its underlying tree are fixed.

In its first phase, the algorithm traverses the pre-proof from leaves to root, and computes at each node permission restrictions that must be satisfied by any write-proof.
Permission Restrictions

A permission restriction is a partial function
\[ \Phi : \text{Vars} \rightarrow \mathcal{P}(\text{Owners}). \]

If a permission restriction is attached to a node \( n \) in a pre-proof, its domain will be the set of variables that may be assigned by the right side of the pre-judgement at \( n \). (If \( n \) is passive, the domain will be empty.)

We say that a context \( \Delta \) satisfies a permission restriction \( \Phi \) iff
\[
\forall v \in \text{dom } \Phi. \sum_{o \in \text{Owners}} \Delta v o = 1
\]
and
\[
\forall v \in \text{dom } \Phi, o \in \text{Owners}. o \notin \Phi v \text{ implies } \Delta v o = 0.
\]

As a consequence,
\[
\forall v \in \text{dom } \Phi. \sum_{o \in \Phi v} \Delta v o = 1.
\]

Note that, if there is any \( v \in \text{dom } \Phi \) such that \( \Phi v \) is empty, then no \( \Delta \) satisfies \( \Phi \).
The Permission Ordering

We impose the following preorder on permission contexts:

\[ \Delta \leq \Delta' \quad \text{iff} \quad \forall v \in \text{Vars}, o \in \text{Owners}. \]

\[ \Delta v o > 0 \ \text{implies} \ \Delta' v o > 0. \]

When \( \Delta \leq \Delta' \), we say that \( \Delta' \) is more permissive than \( \Delta \).
The Plot (Phase II)

If any permission restriction computed in the first phase is unsatisfiable, then there is no proof extending $P^0$.

Otherwise, in its second phase, the algorithm traverses the pre-proof from root to leaves, computing contexts that extend the pre-proof to a \textit{maximally permissive} write-proof.

During the second phase, the algorithm checks the side-conditions on the instances of passive rules. Since, if any write-proof is a proof, the maximally permissive write-proof will be a proof, this suffices to decide whether a proof (extending $P^0$) exists.
An Example (The Problematic Program)

\[ p: \text{self}: 1 \vdash \{ R_1 \ast R_2 \} \]

resource \( r_1 \) in resource \( r_2 \) in

\[(\text{with } r_1 \text{ do } ((\text{with } r_2 \text{ do } p := 0); [0] := 3)) \]

\[ \| (\text{with } r_2 \text{ do } ((\text{with } r_1 \text{ do } p := 1); [0] := 4)) \]

\[ \{ R_1 \ast R_2 \} \]

\[ \text{Vars} = \{ p \} \]

\[ \text{Owners} = \{ r_1, r_2, \text{self} \}. \]
The Resource Invariants

\[ R_1 = \text{if } p = 0 \text{ then } 0 \mapsto 3 \text{ else emp} \]
\[ R_2 = \text{if } p = 0 \text{ then emp else } 0 \mapsto 4. \]

Thus

\[ R_1 \ast R_2 \]
\[ \text{iff if } p = 0 \text{ then } 0 \mapsto 3 \ast \text{emp else } \text{emp} \ast 0 \mapsto 4 \]
\[ \text{iff if } p = 0 \text{ then } 0 \mapsto 3 \text{ else } 0 \mapsto 4 \]
\[ \text{implies } 0 \mapsto - \]

and

\[ p = 0 \land R_1 \text{ iff } p = 0 \land 0 \mapsto 3 \]
\[ p = 0 \land R_2 \text{ iff } p = 0 \land \text{emp} \]
\[ p \neq 0 \land R_1 \text{ iff } p \neq 0 \land \text{emp} \]
\[ p \neq 0 \land R_2 \text{ iff } p \neq 0 \land 0 \mapsto 4. \]
Using the Rule for (Assignable) Variables

\[ p: [\text{self}: 1] \vdash p \text{ Var} \quad \Phi = [p: \{\text{self}\}] \]

Using the Rule for Assignment

\[ \Delta \vdash p \text{ Var} \quad \Phi_1 = [p: \{\text{self}\}] \]
\[ \Delta \vdash 0 \text{ Exp} \quad \Phi_2 = [] \]
\[ \Delta \vdash 0 \leftrightarrow - \land p = 0 \text{ Assert} \quad \Phi_3 = [] \]
\[ \Delta \vdash \{0 \leftrightarrow - \land 0 = 0\} \]
\[ p := 0 \]
\[ \{0 \leftrightarrow - \land p = 0\} \quad \Phi = [p: \{\text{self}\}] \]
Using the Rule of Consequence

\[\Delta \vdash R_1 \ast R_2 \Rightarrow 0 \leftrightarrow - \wedge 0 = 0 \text{ Valid}\]

\[\Phi_1 = []\]

\[\Delta \vdash \{0 \leftrightarrow - \wedge 0 = 0\}\]
\[p := 0\]
\[\{0 \leftrightarrow - \wedge p = 0\}\]

\[\Phi_2 = [p:\{\text{self}\}]\]

\[\Delta \vdash 0 \leftrightarrow - \wedge p = 0 \Rightarrow R_2 \ast (0 \leftrightarrow - \wedge p = 0) \text{ Valid}\]

\[\Phi_3 = []\]

\[\Delta \vdash \{R_1 \ast R_2\}\]
\[p := 0\]
\[\{R_2 \ast (0 \leftrightarrow - \wedge p = 0)\}\]

\[\Phi = [p:\{\text{self}\}]\]
An (Unconditional) Critical Region

\[ \Delta | r_2 : R_2 \vdash R_1 \text{ Assert} \quad \Phi_1 = [] \]
\[ \Delta | r_2 : R_2 \vdash 0 \leftrightarrow - \land p = 0 \text{ Assert} \quad \Phi_2 = [] \]
\[ \Delta' \vdash \{ R_1 \ast R_2 \} \quad \Phi_3 = [p: \{\text{self}\}] \]
\[ \begin{array}{c}
    p := 0 \\
    \{ R_2 \ast (0 \leftrightarrow - \land p = 0) \}
\end{array} \]
\[ \Delta | r_2 : R_2 \vdash \{ R_1 \} \quad \Phi = [p: \{r_2, \text{self}\}] \]

where

\[ \Delta' po = \Delta po \text{ when } o \notin \{r_2, \text{self}\} \]
\[ \Delta' p \text{ self} = \Delta p \text{ self} + \Delta pr_2 \leq 1 \]
\[ \Delta' pr_2 = 0. \]

From \( \Phi_3 \), we have \( \Delta' po = \text{if } o = \text{self then } 1 \text{ else } 0 \), so that

\[ \Delta po = \text{if } o = \text{self then } \pi_1 \text{ else if } o = r_2 \text{ then } \pi_2 \text{ else } 0, \]

where \( \pi_1 + \pi_2 = 1 \). Thus \( \Phi = [p: \{r_2, \text{self}\}] \).
Mutation

\[
\Delta | r_2 : R_2 \vdash 0 \text{ Exp} \quad \Phi_1 = [] \\
\Delta | r_2 : R_2 \vdash 3 \text{ Exp} \quad \Phi_2 = [] \\
\Delta | r_2 : R_2 \vdash \{ 0 \mapsto - \land p = 0 \} \\
\quad \quad [0] := 3 \\
\quad \quad \{ 0 \mapsto 3 \land p = 0 \} \\
\Phi = []
\]

Sequential Composition

\[
\Delta | r_2 : R_2 \vdash \{ R_1 \} \\
\quad \text{with } r_2 \text{ do } p := 0 \\
\quad \quad \{ 0 \mapsto - \land p = 0 \} \\
\Phi_1 = [p : \{ r_2, \text{self} \}] \\
\Delta | r_2 : R_2 \vdash \{ 0 \mapsto - \land p = 0 \} \\
\quad \quad [0] := 3 \\
\quad \quad \{ 0 \mapsto 3 \land p = 0 \} \\
\Phi_2 = [] \\
\Delta | r_2 : R_2 \vdash \{ R_1 \} \\
\quad \text{with } r_2 \text{ do } p := 0 ; \\
\quad \quad [0] := 3 \\
\quad \quad \{ 0 \mapsto 3 \land p = 0 \} \\
\Phi = [p : \{ r_2, \text{self} \}]
\]
Consequence

\[ \Delta | r_2: R_2 \vdash \{ R_1 \} \]
\[ \text{with } r_2 \text{ do } p := 0 ; \]
\[ [0] := 3 \]
\[ \{ 0 \mapsto 3 \land p = 0 \} \]
\[ \Phi_1 = [p: \{ r_2, \text{self} \}] \]

\[ \Delta | r_2: R_2 \vdash 0 \mapsto 3 \land p = 0 \Rightarrow R_1 \text{ Valid} \]
\[ \Phi_2 = [\] \]

\[ \Delta | r_2: R_2 \vdash \{ R_1 \} \]
\[ \text{with } r_2 \text{ do } p := 0 ; \]
\[ [0] := 3 \]
\[ \{ R_1 \} \]
\[ \Phi = [p: \{ r_2, \text{self} \}] \]
Another Critical Region

\[ \Delta | r_1: R_1, r_2: R_2 \vdash \text{emp Assert} \quad \Phi_1 = [\] \\
\Delta | r_1: R_1, r_2: R_2 \vdash \text{emp Assert} \quad \Phi_2 = [\] \\
\Delta' | r_2: R_2 \vdash \{ R_1 \} \\
\text{with } r_2 \text{ do } p := 0 ; \\
[0] := 3 \\
\{ R_1 \} \\
\Phi_3 = [p: \{ r_2, \text{self} \}] \\
\]

\[ \Delta | r_1: R_1, r_2: R_2 \vdash \{ \text{emp} \} \\
\text{with } r_1 \text{ do (} \\
\text{with } r_2 \text{ do } p := 0 ; \\
[0] := 3) \\
\{ \text{emp} \} \\
\Phi = [p: \{ r_1, r_2, \text{self} \}] \\
\]

where

\[ \Delta' p o = \Delta p o \text{ when } o \notin \{ r_1, \text{self} \} \]
\[ \Delta' p \text{ self} = \Delta p \text{ self} + \Delta p r_1 \leq 1 \]
\[ \Delta' p r_1 = 0. \]

From \( \Delta' p r_1 = 0 \), we obtain \( \Phi_3 = [p: \{ r_2, \text{self} \}] \).
Similarly

$\Delta | r_1: R_1, r_2: R_2 \vdash \{\text{emp}\}$

with $r_2$ do (with $r_1$ do $p := 1$ ; [0] := 4)

{emp}

$\Phi = [p: \{r_1, r_2, \text{self}\}]$
Parallel Composition

\[ \Delta_1|_{r_1: R_1, r_2: R_2} \vdash \{\text{emp}\} \quad \Phi_1 = [p: \{r_1, r_2, \text{self}\}] \\
\text{with } r_1 \text{ do (with } r_2 \text{ do } p := 0 ; [0] := 3) \{\text{emp}\} \]

\[ \Delta_2|_{r_1: R_1, r_2: R_2} \vdash \{\text{emp}\} \quad \Phi_2 = [p: \{r_1, r_2, \text{self}\}] \\
\text{with } r_2 \text{ do (with } r_1 \text{ do } p := 1 ; [0] := 4) \{\text{emp}\} \]

\[ \Delta|_{r_1: R_1, r_2: R_2} \vdash \{\text{emp } \ast \text{emp}\} \quad \Phi = [p: \{r_1, r_2\}] \\
\text{with } r_1 \text{ do (with } r_2 \text{ do } p := 0 ; [0] := 3) \parallel \\
\text{with } r_2 \text{ do (with } r_1 \text{ do } p := 1 ; [0] := 4) \{\text{emp } \ast \text{emp}\} \]

where

\[ \Delta_1 p = [r_1: \pi_1, r_2: \pi_2, \text{self: } \pi_s] \]
\[ \Delta_2 p = [r_1: \pi_1, r_2: \pi_2, \text{self: } \pi'_s] \]
\[ \Delta p = [r_1: \pi_1, r_2: \pi_2, \text{self: } \pi_s + \pi'_s] , \]

so that

\[ \pi_1 + \pi_2 + \pi_s = 1 \]
\[ \pi_1 + \pi_2 + \pi'_s = 1 \]
\[ \pi_1 + \pi_2 + \pi_s + \pi'_s \leq 1 , \]

which implies that \( \pi_s = \pi'_s = \pi_s + \pi'_s = 0 \). Thus \( \Phi = [p: \{r_1, r_2\}] \).
Resource Declaration

\[ \Delta_1 | r_1 : R_1 \vdash R_2 \] Assert \hspace{1cm} \Phi_1 = []

\[ \Delta_2 | r_1 : R_1, r_2 : R_2 \vdash \{ \text{emp} \ast \text{emp} \} \] with \( r_1 \) do (with \( r_2 \) do \( p := 0 ; [0] := 3 \)) \hspace{1cm} \Phi_2 = [p : \{r_1, r_2\}]

\| \hspace{1cm} \| 

\[ \text{with } r_2 \text{ do (with } r_1 \text{ do } p := 1 ; [0] := 4) \] \hspace{1cm} \{ \text{emp} \ast \text{emp} \}

\[ \Delta | r_1 : R_1 \vdash \{ \text{emp} \ast \text{emp} \ast R_2 \} \] resource \( r_2 \) in
\hspace{1cm} \Phi = [p : \{r_1, \text{self}\}]

\[ \text{with } r_1 \text{ do (with } r_2 \text{ do } p := 0 ; [0] := 3 \) \| \hspace{1cm} \] with \( r_2 \) do (with \( r_1 \) do \( p := 1 ; [0] := 4 \)) \hspace{1cm} \{ \text{emp} \ast \text{emp} \ast R_2 \}

where

\[ \Delta_2 p = [r_1 : \pi_1, r_2 : \pi_2, \text{self} : \pi_s] \]
\[ \Delta_1 p = [\text{self} : \pi_2] \]
\[ \Delta p = [r_1 : \pi_1, r_2 : 0, \text{self} : \pi_s + \pi_2]. \]

Thus \( \Phi = [p : \{r_1, \text{self}\}] \).
Another Resource Declaration

\[ \Delta_1 \vdash R_1 \text{ Assert} \]
\[ \Phi_1 = [] \]
\[ \Delta_2 | r_1 : R_1 \vdash \{ \text{emp} \ast \text{emp} \ast R_2 \} \]
\[ \Phi_2 = [p:\{r_1, \text{self}\}] \]

resource \( r_2 \) in
\[ \text{with } r_1 \text{ do (with } r_2 \text{ do } p := 0 ; [0] := 3) \]
\[ \parallel \]
\[ \text{with } r_2 \text{ do (with } r_1 \text{ do } p := 1 ; [0] := 4) \]
\[ \{ \text{emp} \ast \text{emp} \ast R_2 \} \]

\[ \Delta \vdash \{ \text{emp} \ast \text{emp} \ast R_1 \ast R_2 \} \]
\[ \Phi = [p:\{\text{self}\}] \]

resource \( r_1 \) in resource \( r_2 \) in
\[ \text{with } r_1 \text{ do (with } r_2 \text{ do } p := 0 ; [0] := 3) \]
\[ \parallel \]
\[ \text{with } r_2 \text{ do (with } r_1 \text{ do } p := 1 ; [0] := 4) \]
\[ \{ \text{emp} \ast \text{emp} \ast R_1 \ast R_2 \} \]

where

\[ \Delta_2 p = [r_1: \pi_1, r_2: \pi_2, \text{self: } \pi_s] \]
\[ \Delta_1 p = [\text{self: } \pi_1] \]
\[ \Delta p = [r_1: 0, r_2: \pi_2, \text{self: } \pi_s + \pi_1]. \]

But by \( \Phi_2 \), \( \pi_2 = 0 \). Thus \( \Phi = [p:\{\text{self}\}] \).
At the Root

We take $\Delta_{\text{root}}^{\text{max}}$ to be the maximally permissive context that satisfies $\Phi_{\text{root}} = [p: \{\text{self}\}]$:

$$\Delta_{\text{root}}^{\text{max}} p = [\text{self} : 1].$$

Then we go backwards through our proof.

Passive judgement whose side conditions hold (either because $\Delta_{\text{root}}^{\text{max}} p \text{self} > 0$ or because $p$ is not a free variable) are marked with an asterisk.
Another Resource Declaration

\[ *\Delta_1 \vdash R_1 \text{ Assert} \]
\[ \Phi_1 = [] \]
\[ \Delta_2 | r_1: R_1 \vdash \{\text{emp} \ast \text{emp} \ast R_2\} \quad \Phi_2 = [p: \{r_1, \text{self}\}] \]

resource \( r_2 \) in

\[ \text{with } r_1 \text{ do (with } r_2 \text{ do } p := 0 ; [0] := 3) \]
\[ \| \]
\[ \text{with } r_2 \text{ do (with } r_1 \text{ do } p := 1 ; [0] := 4) \]
\[ \{\text{emp} \ast \text{emp} \ast R_2\} \]

\[ \Delta_1 p = [\text{self}: \pi_1] \quad \Delta_2 p = [r_1: \pi_1, r_2: \pi_2, \text{self}: \pi_s] \]
\[ \Delta p = [r_1: 0, r_2: \pi_2, \text{self}: \pi_s + \pi_1] . \]

From \( \Delta^{\max}_p = [\text{self} : 1] \), we get \( \pi_2 = 0 \) and \( \pi_s + \pi_1 = 1 \). Choosing \( \pi_s = \pi_1 = \frac{1}{2} \), we have

\[ \Delta^{\max}_1 p = [\text{self}: \frac{1}{2}] \quad \Delta^{\max}_2 p = [r_1: \frac{1}{2}, \text{self}: \frac{1}{2}] \]
Resource Declaration

\[\Delta_1 | r_1: R_1 \vdash R_2 \text{ Assert} \quad \Phi_1 = []\]

\[\Delta_2 | r_1: R_1, r_2: R_2 \vdash \{\text{emp} \ast \text{emp}\} \quad \Phi_2 = [p: \{r_1, r_2\}]\]

with \(r_1\) do (with \(r_2\) do \(p := 0 ; [0] := 3\))

\[\|\]

with \(r_2\) do (with \(r_1\) do \(p := 1 ; [0] := 4\))

\[\{\text{emp} \ast \text{emp}\}\]

\[\Delta | r_1: R_1 \vdash \{\text{emp} \ast \text{emp} \ast R_2\} \quad \Phi = [p: \{r_1, \text{self}\}]\]

resource \(r_2\) in

with \(r_1\) do (with \(r_2\) do \(p := 0 ; [0] := 3\))

\[\|\]

with \(r_2\) do (with \(r_1\) do \(p := 1 ; [0] := 4\))

\[\{\text{emp} \ast \text{emp} \ast R_2\}\]

where

\[\Delta_2 p = [r_1: \pi_1, r_2: \pi_2, \text{self}: \pi_s]\]

\[\Delta_1 p = [\text{self}: \pi_2]\]

\[\Delta p = [r_1: \pi_1, r_2: 0, \text{self}: \pi_s + \pi_2].\]

From \(\Delta_{\text{max}} p = [r_1: \frac{1}{2}, \text{self}: \frac{1}{2}]\), we get \(\pi_1 = \frac{1}{2}\) and \(\pi_s + \pi_2 = \frac{1}{2}\). But \(\Phi_2 p\) forces \(\pi_s = 0\), so that \(\pi_2 = \frac{1}{2}\) and

\[\Delta_{\text{max}}^1 p = [\text{self}: \frac{1}{2}] \quad \Delta_{\text{max}}^2 p = [r_1: \frac{1}{2}, r_2: \frac{1}{2}].\]
Parallel Composition

\[ \Delta_1| r_1: R_1, r_2: R_2 \vdash \{ \text{emp} \} \quad \Phi_1 = [p: \{ r_1, r_2, \text{self} \}] \]
with \( r_1 \) do (with \( r_2 \) do \( p := 0 ; [0] := 3 \)) \{ \text{emp} \}

\[ \Delta_2| r_1: R_1, r_2: R_2 \vdash \{ \text{emp} \} \quad \Phi_2 = [p: \{ r_1, r_2, \text{self} \}] \]
with \( r_2 \) do (with \( r_1 \) do \( p := 1 ; [0] := 4 \)) \{ \text{emp} \}

\[ \Delta| r_1: R_1, r_2: R_2 \vdash \{ \text{emp} \ast \text{emp} \} \quad \Phi = [p: \{ r_1, r_2 \}] \]
with \( r_1 \) do (with \( r_2 \) do \( p := 0 ; [0] := 3 \))
\parallel
with \( r_2 \) do (with \( r_1 \) do \( p := 1 ; [0] := 4 \)) \{ \text{emp} \ast \text{emp} \}

where

\[ \Delta_1 p = [r_1: \pi_1, r_2: \pi_2, \text{self}: \pi_s] \]
\[ \Delta_2 p = [r_1: \pi_1, r_2: \pi_2, \text{self}: \pi'_s] \]
\[ \Delta p = [r_1: \pi_1, r_2: \pi_2, \text{self}: \pi_s + \pi'_s] . \]

From \( \Delta_{\text{max}} p = [r_1: \frac{1}{2}, r_2: \frac{1}{2}] \), we find that \( \pi_s = 0 \) and \( \pi'_s = 0 \), and

\[ \Delta_{\text{max}}^1 p = \Delta_{\text{max}}^2 p = \Delta_{\text{max}} p . \]
Another Critical Region

\[ *\Delta | r_1: R_1, r_2: R_2 \vdash \text{emp Assert} \quad \Phi_1 = [\] \]

\[ *\Delta | r_1: R_1, r_2: R_2 \vdash \text{emp Assert} \quad \Phi_2 = [\] \]

\[ \Delta' | r_2: R_2 \vdash \{ R_1 \} \]
\[ \text{with } r_2 \text{ do } p := 0 ; \]
\[ [0] := 3 \]
\[ \{ R_1 \} \quad \Phi_3 = [p: \{ r_2, \text{self} \}] \]

\[ \Delta | r_1: R_1, r_2: R_2 \vdash \{ \text{emp} \} \]
\[ \text{with } r_1 \text{ do (} \]
\[ \text{with } r_2 \text{ do } p := 0 ; \]
\[ [0] := 3 \) \]
\[ \{ \text{emp} \} \]
\[ \Phi = [p: \{ r_1, r_2, \text{self} \}] \]

where

\[ \Delta' p o = \Delta p o \text{ when } o \notin \{ r_1, \text{self} \} \]
\[ \Delta' p \text{ self} = \Delta p \text{ self} + \Delta p r_1 \leq 1 \]
\[ \Delta' p r_1 = 0. \]

From \( \Delta_{\text{max}} p = [r_1: \frac{1}{2}, r_2: \frac{1}{2}] \) we get

\[ \Delta_{\text{max'}} p = [r_2: \frac{1}{2}, \text{self}: \frac{1}{2}] . \]
Consequence

\[ \Delta | r_2: R_2 \vdash \{ R_1 \} \]
\[ \text{with } r_2 \text{ do } p := 0 ; \]
\[ [0] := 3 \]
\[ \{0 \iff 3 \land p = 0\} \]

\[ \Phi_1 = [p: \{r_2, \text{self}\}] \]

\[ \Phi_2 = [\] \]

\[ \Phi = [p: \{r_2, \text{self}\}] \]

\[ \text{Obviously, } \Delta^{\text{max}} \text{ is preserved.} \]
Mutation

\[
\begin{align*}
*\Delta | r_2: R_2 &\vdash 0 \ Exp \\
*\Delta | r_2: R_2 &\vdash 3 \ Exp \\
\hline
\Delta | r_2: R_2 &\vdash \{0 \mapsto - \land \ p = 0\} \\
&[0] := 3 \\
&\{0 \mapsto 3 \land \ p = 0\}
\end{align*}
\]

Obviously, $\Delta^{\text{max}}$ is preserved.

Sequential Composition

\[
\begin{align*}
\Delta | r_2: R_2 &\vdash \{R_1\} \\
&\text{with } r_2 \ 	ext{do } \ p := 0 \\
&\{0 \mapsto - \land \ p = 0\}
\end{align*}
\]

\[
\begin{align*}
*\Delta | r_2: R_2 &\vdash \{0 \mapsto - \land \ p = 0\} \\
&[0] := 3 \\
&\{0 \mapsto 3 \land \ p = 0\}
\end{align*}
\]

\[
\begin{align*}
\Delta | r_2: R_2 &\vdash \{R_1\} \\
&\text{with } r_2 \ 	ext{do } \ p := 0 \ ; \\
&[0] := 3 \\
&\{0 \mapsto 3 \land \ p = 0\}
\end{align*}
\]

Obviously, $\Delta^{\text{max}}$ is preserved.
An (Unconditional) Critical Region

\[ *\Delta | r_2: R_2 \vdash R_1 \text{ Assert} \]
\[ \Delta | r_2: R_2 \vdash 0 \leftrightarrow - \wedge p = 0 \text{ Assert} \]
\[ \Delta' \vdash \{ R_1 \ast R_2 \} \]
\[ p := 0 \]
\[ \{ R_2 \ast (0 \leftrightarrow - \wedge p = 0) \} \]
\[ \Delta | r_2: R_2 \vdash \{ R_1 \} \]
with \( r_2 \) do \( p := 0 \)
\[ \{ 0 \leftrightarrow - \wedge p = 0 \} \]

where

\[ \Delta'_{p o} = \Delta_{p o} \text{ when } o \notin \{ r_2, \text{self} \} \]
\[ \Delta'_{p \text{self}} = \Delta_{p \text{self}} + \Delta_{p r_2} \leq 1 \]
\[ \Delta'_{p r_2} = 0. \]

From \( \Delta_{\text{max}} p = [ r_2: \frac{1}{2}, \text{self}: \frac{1}{2} ] \), we get

\[ \Delta'_{\text{max}} p = [ \text{self}: 1 ]. \]
Using the Rule of Consequence

\[ \Delta \vdash R_1 * R_2 \Rightarrow 0 \leftrightarrow - \land 0 = 0 \quad \text{Valid} \]

\[ \Delta \vdash \{0 \leftrightarrow - \land 0 = 0\} \]
\[ p := 0 \]
\[ \{0 \leftrightarrow - \land p = 0\} \]

\[ \Delta \vdash \{R_1 * R_2\} \]
\[ p := 0 \]
\[ \{R_2 * (0 \leftrightarrow - \land p = 0)\} \]

\[ \Phi_1 = [] \]
\[ \Phi_2 = [p: \{\text{self}\}] \]
\[ \Phi_3 = [] \]
\[ \Phi = [p: \{\text{self}\}] \]

Obviously, \( \Delta^{\text{max}} \) is preserved.
Using the Rule for (Assignable) Variables

\[
p : [\text{self: 1}] \vdash p \text{ Var} \quad \Phi = [p: \{\text{self}\}]
\]

which is satisfied by \( \Delta^{\text{max}} p = [\text{self: 1}] \).

Using the Rule for Assignment

\[
\Delta \vdash p \text{ Var} \\
\ast \Delta \vdash 0 \text{ Exp} \\
\ast \Delta \vdash 0 \iff - \wedge p = 0 \quad \text{Assert}
\]

\[
\Delta \vdash \{0 \iff - \wedge 0 = 0\} \\
p := 0 \\
\{0 \iff - \wedge p = 0\}
\]

\( \Phi_1 = [p: \{\text{self}\}] \)
\( \Phi_2 = [] \)
\( \Phi_3 = [] \)
\( \Phi = [p: \{\text{self}\}] \)

Obviously, \( \Delta^{\text{max}} \) is preserved.
At each Node during Phase I

Consider a node $n$ in $P^0$ whose parents are $n_1, \ldots, n_k$. The judgements at these nodes will form an instance of a pre-rule:

$$R^0: \frac{\gamma_{n_1} \vdash S_{n_1} \quad \cdots \quad \gamma_{n_k} \vdash S_{n_k}}{\gamma_n \vdash S_n}.$$ 

During Phase I, the algorithm will accept permission restrictions $\Phi_{n_1}, \ldots, \Phi_{n_k}$ and will produce a permission restriction $\Phi_n$ such that

(1) If

$$\frac{\Delta_{n_1} | \gamma_{n_1} \vdash S_{n_1} \quad \cdots \quad \Delta_{n_k} | \gamma_{n_k} \vdash S_{n_k}}{\Delta_n | \gamma_n \vdash S_n}$$

is a rule instance that erases to $R^0$, and if $\Delta_{n_i}$ satisfies $\Phi_{n_i}$ for $1 \leq i \leq k$, then $\Delta_n$ will satisfy $\Phi_n$. 
The Result of Phase I

In Phase I, the algorithm will produce a permission restriction $\Phi_n$ for each node $n$ in $P^0$.

By structural induction on $P^0$, using (1):

(2) If $P^w$ is a write-proof that extends $P^0$ with contexts $\langle \Delta_n \rangle$, then each $\Delta_n$ satisfies $\Phi_n$. 
At the Root

In Phase II, the algorithm will search for a proof whose root judgement contains the context $\Delta_{\text{root}}^{\text{max}}$, which must satisfy $\Phi_{\text{root}}$. There are two cases:

**Specified Root Context:** We take $\Delta_{\text{root}}^{\text{max}}$ to be the specified root context, providing it satisfies $\Phi_{\text{root}}$. Otherwise, by (2), there is no write-proof (and therefore no proof) that extends $P^0$ and has the specified root context.

**Arbitrary Root Context:** If, for every $v$ in $\text{dom } \Phi_{\text{root}}$, $\Phi_{\text{root}} v$ is nonempty, then we take $\Delta_{\text{root}}^{\text{max}}$ to be

$$
\Delta_{\text{root}}^{\text{max}} v o = \begin{cases} 
\text{if } v \in \text{dom } \Phi_{\text{root}} \text{ then} \\
\quad \text{if } o \in \Phi_{\text{root}} v \text{ then } 1/\#\Phi_{\text{root}} v \text{ else } 0 \\
\quad \text{else } 1/ (\#\text{Owners} + 1) 
\end{cases}
$$

(where $\#S$ is the size of $S$), which is (one of) the most permissive contexts satisfying $\Phi_{\text{root}}$.

On the other hand, if there is some variable $v$ such that $\Phi_{\text{root}} v$ is empty, then there is no root context satisfying $\Phi_{\text{root}}$, and by (2), no proof extends $P^0$. 
At each Node during Phase II

During Phase II, the algorithm will accept a context $\Delta_n$ that satisfies $\Phi_n$ and will produce contexts $\Delta_{n_1}^{\text{max}}, \ldots, \Delta_{n_k}^{\text{max}}$ such that

(3) Each $\Delta_{n_i}^{\text{max}}$ satisfies $\Phi_{n_i}$ and

$$\Delta_{n_1}^{\text{max}} | \gamma_{n_1} \vdash S_{n_1} \cdots \Delta_{n_k}^{\text{max}} | \gamma_{n_k} \vdash S_{n_k}$$

$$\Delta_n^{\text{max}} | \gamma_n \vdash S_n.$$ 

is a rule instance that erases to $R^0$. Moreover,

(4) If $\Delta_{n_1}, \ldots, \Delta_{n_k}$, and $\Delta_n$ satisfy $\Phi_{n_1}, \ldots, \Phi_{n_k}$, and $\Phi_n$ respectively,

$$\Delta_{n_1} | \gamma_{n_1} \vdash S_{n_1} \cdots \Delta_{n_k} | \gamma_{n_k} \vdash S_{n_k}$$

$$\Delta_n | \gamma_n \vdash S_n$$

is a rule instance that erases to $R^0$, and $\Delta_n \leq \Delta_n^{\text{max}}$, then $\Delta_{n_i} \leq \Delta_{n_i}^{\text{max}}$ for $1 \leq i \leq k$. 
The Result of Phase II

In Phase II, given a context $\Delta_{\text{root}}^\text{max}$ satisfying $\Phi_{\text{root}}$, the algorithm will produce a context $\Delta_n^\text{max}$ for each node $n$.

By induction on distance from the root, using (3):

(5) There is a write-proof that extends $P^0$ with $\langle \Delta_n^\text{max} \rangle$.

Moreover, using (2), and then induction on distance from the root, using (4):

(6) If there is a write-proof that extends $P^0$ with $\langle \Delta_n \rangle$, and $\Delta_{\text{root}} \leq \Delta_{\text{root}}^\text{max}$, then $\Delta_n \leq \Delta_n^\text{max}$ for each node $n$. 
In Phase II, while generating the $\Delta_{n}^{\text{max}}$, the algorithm can check whether, at all passive nodes, the side conditions of the rules

$\Delta|\gamma \vdash E \text{ Exp}$ where $\forall v \in \text{FV}(E)$. $\Delta v \text{ self} > 0$

$\Delta|\gamma \vdash P \text{ Assert}$ where $\forall v \in \text{FV}(P)$. $\Delta v \text{ self} > 0$

$\Delta|\gamma \vdash P \text{ Valid}$ where $\forall v \in \text{FV}(P)$. $\Delta v \text{ self} > 0$

are satisfied. If and only if these conditions are satisfied, the write-proof that extends $P^{0}$ with $\langle \Delta_{n}^{\text{max}} \rangle$ will be a proof.

Moreover, suppose there is some proof that extends $P^{0}$ with $\langle \Delta_{n} \rangle$ and that $\Delta_{\text{root}} \leq \Delta_{\text{root}}^{\text{max}}$. Then by (6), $\Delta_{n} \leq \Delta_{n}^{\text{max}}$ for all nodes $n$. It follows that, since the side conditions at passive $n$ are met by $\Delta_{n}$, they will be met by $\Delta_{n}^{\text{max}}$, so that the write-proof that extends $P^{0}$ with $\Delta_{n}^{\text{max}}$ will also be a proof.

It follows that either the algorithm will find a proof that extends $P^{0}$ with $\Delta_{\text{root}}^{\text{max}}$ at the root, or there is no proof that extends $P^{0}$ with any $\Delta_{\text{root}} \leq \Delta_{\text{root}}^{\text{max}}$. 
The Finale (continued)

It follows that either the algorithm will find a proof that extends $P^0$ with $\Delta_{\text{root}}^{\text{max}}$ at the root, or there is no proof that extends $P^0$ with any $\Delta_{\text{root}} \leq \Delta_{\text{root}}^{\text{max}}$.

Specified Root Context: If $\Delta_{\text{root}}^{\text{max}}$ is the specified root context, then either the algorithm will find a proof that extends $P^0$ with $\Delta_{\text{root}}^{\text{max}}$ at the root, or, since $\Delta_{\text{root}}^{\text{max}} \leq \Delta_{\text{root}}^{\text{max}}$, there is no proof that extends $P^0$ with $\Delta_{\text{root}}^{\text{max}}$ at the root.

Arbitrary Root Context: Here $\Delta_{\text{root}}^{\text{max}}$ is the most permissive context satisfying $\Phi_{\text{root}}$. Either the algorithm will find a proof that extends $P^0$, or there is no proof thatextends $P^0$ with any $\Delta_{\text{root}}$ that satisfies $\Phi_{\text{root}}$. But by (2), there is no proof that extends $P_0$ with any $\Delta_{\text{root}}$ that does not satisfies $\Phi_{\text{root}}$. 
The Passive Rules

\[ \Delta \vdash \gamma \vdash E \quad \text{Exp} \quad \text{where } \forall v \in \text{FV}(E). \Delta v \text{ self} > 0 \]

\[ \Delta \vdash \gamma \vdash P \quad \text{Assert} \quad \text{where } \forall v \in \text{FV}(P). \Delta v \text{ self} > 0 \]

\[ \Delta \vdash \gamma \vdash P \quad \text{Valid} \quad \text{where } \forall v \in \text{FV}(P). \Delta v \text{ self} > 0, \]

where \( \text{FV}(X) \) denotes the set of free variables of \( X \).

\( \Phi \) is the empty function.

Since there are no premisses, there are no \( \Delta^\text{max}_i \) to be computed. But the side conditions must be checked to determine if a write-proof is a proof.
The Rule for (Assignable) Variables

\[ \Delta|\gamma \vdash v \text{ Var}, \]

where

\[ \Delta v' o = 0 \text{ when } v' \neq v \]
\[ \Delta v o = \text{ if } o = \text{ self then } 1 \text{ else } 0. \]

\[ \text{dom } \Phi = \{v\} \quad \Phi v = \{\text{self}\}. \]

Since there are no premisses, there are no \( \Delta^\text{max}_i \) to be computed. Moreover, it is clear that \( \Delta^\text{max} \) will meet the side condition since \( \Delta^\text{max} \) will satisfy \( \Phi \).
Sequential Composition (Many rules are similar.)

\[
\frac{\Delta_1|\gamma \vdash \{P\} \ C \ \{Q\} \ \Delta_2|\gamma \vdash \{Q\} \ C' \ \{R\}}{
\Delta|\gamma \vdash \{P\} \ C ; C' \ \{R\},}
\]

where

\[
\Delta_1 = \Delta_2 = \Delta.
\]

\[
dom \Phi = dom \Phi_1 \cup dom \Phi_2.
\]

When \( v \in dom \Phi \):

\[
o \in \Phi \ v \iff \begin{cases} 
(v \in dom \Phi_1 \Rightarrow o \in \Phi_1 \ v) \wedge \\
(v \in dom \Phi_2 \Rightarrow o \in \Phi_2 \ v),
\end{cases}
\]

or equivalently

\[
o \notin \Phi \ v \iff \begin{cases} 
(v \in dom \Phi_1 \land o \notin \Phi_1 \ v) \lor \\
(v \in dom \Phi_2 \land o \notin \Phi_2 \ v).
\end{cases}
\]

\[
\Delta_1^{\text{max}} = \Delta_2^{\text{max}} = \Delta^{\text{max}}.
\]
Conditionals

\[ \frac{\Delta_1 | \gamma \vdash B \text{ Assert}}{\Delta_2 | \gamma \vdash \{P \land B\} \ C \ \{Q\} \ \Delta_3 | \gamma \vdash \{P \land \neg B\} \ C' \ \{Q\}} \]

\[ \Delta | \gamma \vdash \{P\} \ \text{if} \ B \ \text{then} \ C \ \text{else} \ C' \ \{Q\}, \]

where

\[ \Delta_1 = \Delta_2 = \Delta_3 = \Delta. \]

\[ \text{dom } \Phi = \text{dom } \Phi_1 \cup \text{dom } \Phi_2 \cup \text{dom } \Phi_3. \]

When \( v \in \text{dom } \Phi \): \[
\begin{align*}
\Phi_v (v) \iff \begin{cases} 
(v \in \text{dom } \Phi_1 \Rightarrow o \in \Phi_1 v) \\
\land \\
(v \in \text{dom } \Phi_2 \Rightarrow o \in \Phi_2 v) \\
\land \\
(v \in \text{dom } \Phi_3 \Rightarrow o \in \Phi_3 v).
\end{cases}
\end{align*}
\]

\[ \Delta_1^{\max} = \Delta_2^{\max} = \Delta_3^{\max} = \Delta^{\max}. \]

Note that \( \Phi_1 \) will be the empty function.
Parallel Composition (Frame is similar)

\[
\frac{\Delta_1|\gamma \vdash \{P\} C \{Q\} \quad \Delta_2|\gamma \vdash \{P'\} C' \{Q'\}}{\Delta|\gamma \vdash \{P * P'\} C || C' \{Q * Q'\}},
\]

where

\[
\Delta v o = \Delta_1 v o = \Delta_2 v o \text{ when } o \neq \text{self} \\
\Delta v \text{self} = \Delta_1 v \text{self} + \Delta_2 v \text{self} \leq 1. \tag{A}
\]

dom \Phi = \text{dom } \Phi_1 \cup \text{dom } \Phi_2.

When \( v \in \text{dom } \Phi \):

\[
o \in \Phi v \text{ iff } \begin{cases} (v \in \text{dom } \Phi_1 \Rightarrow o \in \Phi_1 v) \\
\wedge \\
(v \in \text{dom } \Phi_2 \Rightarrow o \in \Phi_2 v) \\
\wedge \\
(v \in \text{dom } \Phi_1 \cap \text{dom } \Phi_2 \Rightarrow o \neq \text{self}), \end{cases}
\]
or equivalently,

\[
o \notin \Phi v \text{ iff } \begin{cases} (v \in \text{dom } \Phi_1 \wedge o \notin \Phi_1 v) \\
\lor \\
(v \in \text{dom } \Phi_2 \wedge o \notin \Phi_2 v) \\
\lor \\
(v \in \text{dom } \Phi_1 \cap \text{dom } \Phi_2 \wedge o = \text{self}). \tag{B}
\]

Parallel Composition (continued)

\[
\begin{align*}
\Delta_{\text{max}}^1 v o & = \Delta_{\text{max}}^1 v o \\
\Delta_{\text{max}}^2 v o & = \Delta_{\text{max}}^2 v o \\
\end{align*}
\]  \quad \text{when } o \neq \text{self}

\[
\begin{align*}
\Delta_{\text{max}}^1 v \text{self} & = \Delta_{\text{max}}^1 v \text{self} \\
\Delta_{\text{max}}^2 v \text{self} & = 0 \\
\end{align*}
\]  \quad \text{when } \{ v \in \text{dom } \Phi_1 \land v \notin \text{dom } \Phi_2 \}

\[
\begin{align*}
\Delta_{\text{max}}^1 v \text{self} & = 0 \\
\Delta_{\text{max}}^2 v \text{self} & = \Delta_{\text{max}}^2 v \text{self} \\
\end{align*}
\]  \quad \text{when } \{ v \notin \text{dom } \Phi_1 \land v \in \text{dom } \Phi_2 \} \quad (C)

\[
\begin{align*}
\Delta_{\text{max}}^1 v \text{self} & = \frac{1}{2} \Delta_{\text{max}}^1 v \text{self} \\
\Delta_{\text{max}}^2 v \text{self} & = \frac{1}{2} \Delta_{\text{max}}^2 v \text{self} \\
\end{align*}
\]  \quad \text{when } \{ v \in \text{dom } \Phi_1 \land v \in \text{dom } \Phi_2 \}

\[
\begin{align*}
\Delta_{\text{max}}^1 v \text{self} & = \frac{1}{2} \Delta_{\text{max}}^1 v \text{self} \\
\Delta_{\text{max}}^2 v \text{self} & = \frac{1}{2} \Delta_{\text{max}}^2 v \text{self} \\
\end{align*}
\]  \quad \text{when } \{ v \notin \text{dom } \Phi_1 \land v \notin \text{dom } \Phi_2 \}
Parallel Composition — Proof of (1)

(1) If $\Delta_1$ satisfies $\Phi_1$, $\Delta_2$ satisfies $\Phi_2$, and (A) and (B), then $\Delta$ satisfies $\Phi$.

Proof Suppose, for $i \in \{1, 2\}$, $\Delta_i$ satisfies $\Phi_i$, so that

$$\forall v \in \text{dom } \Phi_i. \sum_{o \in \text{Owners}} \Delta_i v o = 1$$
$$\forall v \in \text{dom } \Phi_i, o \in \text{Owners}. o \notin \Phi_i v \text{ implies } \Delta_i v o = 0.$$ 

Now suppose $v \in \text{dom } \Phi = \text{dom } \Phi_1 \cup \text{dom } \Phi_2$. If $v \in \text{dom } \Phi_1$, then by (A):

$$\sum_{o \in \text{ Owners}} \Delta v o = (\sum_{o \in \text{ Owners}} \Delta_1 v o) + \Delta_2 v \text{ self}$$
$$= 1 + \Delta_2 v \text{ self}.$$ 

But $\sum_{o \in \text{ Owners}} \Delta v o \leq 1$, so

$$\sum_{o \in \text{ Owners}} \Delta v o = 1 \quad \text{and} \quad \Delta_2 v \text{ self} = 0.$$ 

Similarly, if $v \in \text{dom } \Phi_2$, then

$$\sum_{o \in \text{ Owners}} \Delta v o = 1 \quad \text{and} \quad \Delta_1 v \text{ self} = 0.$$
Suppose \( v \in \text{dom } \Phi \) and \( o \notin \Phi v \). Then, by (B), there are three possibilities, each of which implies \( \Delta vo = 0 \):

- \( v \in \text{dom } \Phi_1 \) and \( o \notin \Phi_1, v \), so that \( \Delta_1 vo = 0 \) and \( \Delta_2 v\text{self} = 0 \).
- \( v \in \text{dom } \Phi_2 \) and \( o \notin \Phi_2, v \), so that \( \Delta_2 vo = 0 \) and \( \Delta_1 v\text{self} = 0 \).
- \( v \in \text{dom } \Phi_1, v \in \text{dom } \Phi_2 \) and \( o = \text{self} \), so that \( \Delta_2 v\text{self} = 0, \Delta_1 v\text{self} = 0 \), and \( o = \text{self} \).

Thus we have

\[
\forall v \in \text{dom } \Phi. \sum_{o \in \text{Owners}} \Delta vo = 1
\]

\[
\forall v \in \text{dom } \Phi, o \in \text{Owners}. o \notin \Phi v \text{ implies } \Delta_i vo = 0,
\]

so that \( \Delta \) satisfies \( \Phi \).
Parallel Composition — Proof of (3)

(3) If $\Delta^{\text{max}}$ satisfies $\Phi$, then $\Delta^{\text{max}}_1$ and $\Delta^{\text{max}}_2$, as defined by (C), satisfy $\Phi_1$ and $\Phi_2$ respectively, and

\[
\Delta^{\text{max}} v_o = \Delta^{\text{max}}_1 v_o = \Delta^{\text{max}}_2 v_o \text{ when } o \neq \text{self} \\
\Delta^{\text{max}} v_{\text{self}} = \Delta^{\text{max}}_1 v_{\text{self}} + \Delta^{\text{max}}_2 v_{\text{self}} \leq 1. \tag{D}
\]

Proof  It is easily seen that (C) satisfies (D).

To show that $\forall v \in \text{dom } \Phi_1$. $\sum_{o \in \text{Owners}} \Delta^{\text{max}}_1 v_o$, assume $v \in \text{dom } \Phi_1$. From (D) we have

\[
\sum_{o \in \text{Owners}} \Delta^{\text{max}}_1 v_o = \left( \sum_{o \in \text{Owners}} \Delta^{\text{max}} v_o \right) - \Delta^{\text{max}}_2 v_{\text{self}}.
\]

When $v \notin \text{dom } \Phi_2$, (C) gives $\Delta^{\text{max}}_2 v_{\text{self}} = 0$ directly. When $v \in \text{dom } \Phi_2$, (B) gives $\text{self} \notin \Phi v$ and since $\Delta^{\text{max}}$ is assumed to satisfy $\Phi$, $\Delta^{\text{max}} v_{\text{self}} = 0$, so that the penultimate case of (C) gives $\Delta^{\text{max}}_2 v_{\text{self}} = 0$. Thus, in either case,

\[
\sum_{o \in \text{Owners}} \Delta^{\text{max}}_1 v_o = \sum_{o \in \text{Owners}} \Delta^{\text{max}} v_o = 1.
\]
To show that

\[ \forall v \in \text{dom } \Phi_1, o \in \text{Owners. } o \notin \Phi_1 v \text{ implies } \Delta^\text{max}_1 v o = 0, \]

assume \( v \in \text{dom } \Phi_1, o \in \text{Owners} \) and \( o \notin \Phi_1 v \). Then

(B) gives \( o \notin \Phi v \), and the the assumption that \( \Delta^\text{max} \) satisfies \( \Phi \) gives \( \Delta^\text{max} v o = 0 \). Finally, (D) gives \( \Delta^\text{max}_1 v o = 0 \).

Thus \( \Delta^\text{max}_1 \) satisfies \( \Phi_1 \). The argument for \( \Delta^\text{max}_2 \) satisfies \( \Phi_2 \) is symmetric.
Parallel Composition — Proof of (4)

If $\Delta_1$, $\Delta_2$, and $\Delta$ satisfy $\Phi_1$, $\Phi_2$, and $\Phi$ respectively, (A) holds, and $\Delta \leq \Delta^{\text{max}}$, then $\Delta_1 \leq \Delta_1^{\text{max}}$ and $\Delta_2 \leq \Delta_2^{\text{max}}$.

Proof  Assume the hypotheses of the lemma, and $\Delta_1 v_o > 0$. To show $\Delta_1^{\text{max}} v_o > 0$, we first note that (A) gives $\Delta_1 v_0 \leq \Delta v_0$, which, with $\Delta \leq \Delta^{\text{max}}$, gives

$$\Delta_1 v_0 > 0 \Rightarrow \Delta v_0 > 0 \Rightarrow \Delta^{\text{max}} v_0 > 0.$$  

So we need to show $\Delta^{\text{max}} v_0 > 0 \Rightarrow \Delta_1^{\text{max}} v_0 > 0$.

- When $o \neq \text{self}$, (D) gives $\Delta_1^{\text{max}} v_0 = \Delta^{\text{max}} v_0$.
- When $o = \text{self}$, (A) gives $\Delta_1 v_{\text{self}} \leq \Delta v_{\text{self}}$. When $v \notin \text{dom } \Phi_1$ and $v \in \text{dom } \Phi_2$, since $\Delta_2$ satisfies $\Phi_2$, (A) gives

$$\sum_{o \in \text{Owners}} \Delta v_o = (\sum_{o \in \text{Owners}} \Delta_2 v_o) + \Delta_1 v_{\text{self}}$$

$$= 1 + \Delta_1 v_{\text{self}}.$$  

so that $\sum_{o \in \text{Owners}} \Delta v_o \leq 1$ gives $\Delta_1 v_{\text{self}} = 0$, which contradicts $\Delta_1 v_0 > 0$.

Otherwise, (C) gives

$$\Delta^{\text{max}} v_{\text{self}} > 0 \Rightarrow \Delta_1^{\text{max}} v_{\text{self}} > 0.$$  

The argument for $\Delta_2 \leq \Delta_2^{\text{max}}$ is symmetric.
Resource Declaration

\[
\frac{\Delta_1|\gamma \vdash R \text{ Assert} \quad \Delta_2|\gamma, r: R \vdash \{P\} C \{Q\}}{\Delta|\gamma \vdash \{P \ast R\} \text{ resource } r \text{ in } C \{Q \ast R\},} \tag{R \text{ precise)}
\]

where

\[
\begin{align*}
\Delta vo &= \Delta_2 vo \text{ when } o \notin \{\text{self}, r\} \\
\Delta v self &= \Delta_2 v self + \Delta_2 vr \leq 1 \\
\Delta_1 vo &= 0 \text{ when } o \neq \text{ self} \\
\Delta_1 v self &= \Delta_2 vr.
\end{align*}
\]

\[
\text{dom } \Phi = \text{dom } \Phi_2.
\]

When \( v \in \text{dom } \Phi \):

\[
\begin{align*}
o &\in \Phi v \text{ iff } o \in \Phi_2 v \text{ when } o \notin \{\text{self}, r\} \\
r &\notin \Phi v \\
\text{self} &\in \Phi v \text{ iff } \text{self} \in \Phi_2 v \lor r \in \Phi_2 v.
\end{align*}
\]
\[ \Delta_2^{\max} v_o = \Delta_1^{\max} v_o \quad \text{when} \quad o \notin \{\text{self, } r\} \]

\[ \begin{align*}
\Delta_2^{\max} v_{\text{self}} &= \frac{1}{2} \Delta_1^{\max} v_{\text{self}} \\
\Delta_2^{\max} v_r &= \frac{1}{2} \Delta_1^{\max} v_{\text{self}} 
\end{align*} \quad \text{when} \quad v \notin \text{dom } \Phi_2 \]

\[ \begin{align*}
\Delta_2^{\max} v_{\text{self}} &= \Delta_1^{\max} v_{\text{self}} \\
\Delta_2^{\max} v_r &= 0 
\end{align*} \quad \text{when} \quad \begin{cases} 
v \in \text{dom } \Phi_2 \land \\
\text{self} \notin \Phi_2 v \land \\
r \notin \Phi_2 v 
\end{cases} \]

\[ \begin{align*}
\Delta_2^{\max} v_{\text{self}} &= 0 \\
\Delta_2^{\max} v_r &= \Delta_1^{\max} v_{\text{self}} 
\end{align*} \quad \text{when} \quad \begin{cases} 
v \in \text{dom } \Phi_2 \land \\
\text{self} \notin \Phi_2 v \land \\
r \in \Phi_2 v 
\end{cases} \]

\[ \begin{align*}
\Delta_2^{\max} v_{\text{self}} &= \frac{1}{2} \Delta_1^{\max} v_{\text{self}} \\
\Delta_2^{\max} v_r &= \frac{1}{2} \Delta_1^{\max} v_{\text{self}} 
\end{align*} \quad \text{when} \quad \begin{cases} 
v \in \text{dom } \Phi_2 \land \\
\text{self} \in \Phi_2 v \land \\
r \in \Phi_2 v 
\end{cases} \]

\[ \Delta_1^{\max} v_o = 0 \quad \text{when} \quad o \neq \text{self} \]

\[ \Delta_1^{\max} v_{\text{self}} = \Delta_2^{\max} v_r \]
Critical Regions

\[ \Delta_1|\gamma, r: R \vdash P \quad \text{Assert} \quad \Delta_2|\gamma, r: R \vdash Q \quad \text{Assert} \]
\[ \Delta_3|\gamma \vdash B \quad \text{Assert} \quad \Delta_4|\gamma \vdash \{(P \star R) \land B\} \quad C \quad \{Q \star R\} \]

\[ \Delta|\gamma, r: R \vdash \{P\} \text{ with } r \text{ when } B \text{ do } C \quad \{Q\}, \]

where

\[ \Delta_1 = \Delta_2 = \Delta \quad \Delta_3 = \Delta_4, \]
\[ \Delta_4|v o = \Delta|v o \text{ when } o \notin \{\text{self}, r\} \]
\[ \Delta_4|v \text{ self} = \Delta|v \text{ self} + \Delta|v r \leq 1 \]
\[ \Delta_4|v r = 0. \]

\[ \text{dom } \Phi = \text{dom } \Phi_{4}. \]

When \( v \in \text{dom } \Phi \):

\[ o \in \Phi|v \quad \text{iff} \quad o \in \Phi_{4}|v \text{ when } o \notin \{\text{self}, r\} \]
\[ \text{self} \in \Phi|v \quad \text{iff} \quad \text{self} \in \Phi_{4}|v \]
\[ r \in \Phi|v \quad \text{iff} \quad \text{self} \in \Phi_{4}|v. \]

\[ \Delta_{1}^{\max} = \Delta_{2}^{\max} = \Delta_{3}^{\max} = \Delta_{4}^{\max}, \]
\[ \Delta_{4}^{\max}|v o = \Delta_{4}^{\max}|v o \text{ when } o \notin \{\text{self}, r\} \]
\[ \Delta_{4}^{\max}|v \text{ self} = \Delta_{4}^{\max}|v \text{ self} + \Delta_{4}^{\max}|v r \]
\[ \Delta_{4}^{\max}|v r = 0. \]
Variable Declaration

\[ \Delta_1 \mid \gamma \vdash P \text{ Assert} \quad \Delta_2 \mid \gamma \vdash Q \text{ Assert} \]
\[ \Delta_3 \mid \gamma \vdash E \text{ Exp} \quad \Delta_4 \mid \gamma \vdash \{P\} C \{Q\} \]
\[ \Delta \mid \gamma \vdash \{P\} \text{ local } v := E \text{ in } C \{Q\}, \]

where

\[ \Delta_1 = \Delta_2 = \Delta_3 = \Delta, \]
\[ \Delta_4 v' o = \Delta v' o \text{ when } v' \neq v \]
\[ \Delta_4 v o = \text{ if } o = \text{ self then } 1 \text{ else } 0. \]

If \( v \in \text{dom } \Phi_4 \) and \( \text{self} \notin \Phi_4 v \) then there is no write-proof extending \( P^0 \). Otherwise,

\[ \text{dom } \Phi = (\text{dom } \Phi_4) - \{v\} \]

When \( v' \in \text{dom } \Phi \):

\[ \Phi v' = \Phi_4 v' \text{ when } v' \in \text{dom } \Phi. \]

\[ \Delta_1^{\text{max}} = \Delta_2^{\text{max}} = \Delta_3^{\text{max}} = \Delta_4^{\text{max}}, \]
\[ \Delta_4^{\text{max}} v' o = \Delta^{\text{max}} v' o \text{ when } v' \neq v \]
\[ \Delta_4^{\text{max}} v o = \text{ if } o = \text{ self then } 1 \text{ else } 0. \]