

1. Soundness

In this section we provide a proof that asserts that transactional evaluation in which isolate execution is defined via state partitioning (\Rightarrow) is equivalent to transactional evaluation in which isolate arguments are evaluated in an arbitrary serial order (\rightsquigarrow). We also provide a proof that asserts that transactional evaluation in which the left and right events of isolates are executed concurrently (\Longrightarrow) is equivalent to evaluation in which the arguments are evaluated in a serial order (\rightsquigarrow).

1.1 State Partitioning

We first prove transactional evaluation defined via state partitioning (\Rightarrow) is equivalent to transactional evaluation of isolates in an arbitrary serial order (\rightsquigarrow). We use transform function \mathcal{T} which we define below.

THEOREM 1.1. *State Partitioning Soundness*

If

$$\begin{aligned} & \oplus, \bar{K} \parallel (\mathbf{t}^\tau, M, \mathcal{F}[\text{isolatedEvt}(v_1, v_2)]) \Rightarrow \odot, \bar{K}_1 \Rightarrow \dots \Rightarrow \\ & \odot, \bar{K}_n \Rightarrow \oplus, \bar{K}' \parallel (\mathbf{t}^\tau, M, \mathcal{F}[\text{alwaysEvt}(v'_1, v'_2)]), \\ & \text{then} \\ & \mathcal{T}(\oplus, \bar{K} \parallel (\mathbf{t}^\tau, M, \mathcal{F}[\text{isolatedEvt}(v_1, v_2)])) \rightsquigarrow \bar{K}'_1 \rightsquigarrow \dots \rightsquigarrow \\ & \bar{K}'_n \rightsquigarrow \mathcal{T}(\oplus, \bar{K}' \parallel (\mathbf{t}^\tau, M, \mathcal{F}[\text{alwaysEvt}(v'_1, v'_2)])). \end{aligned}$$

Theorem 1.1 states that if transactional evaluation defined via relation \Rightarrow evaluates an isolate to an always event, then transactional evaluation defined via \rightsquigarrow can evaluate the transform of the initial state to the transform of the final state.

Intermediate states in an evaluation sequence that define transactional evaluation via relation \Rightarrow may contain states that evaluate isolates concurrently. An evaluation sequence that defines transactional evaluation via relation \rightsquigarrow , on the other hand, evaluates isolates in an arbitrary serial order. Our proof utilizes a transform function \mathcal{T} that maps *well-defined token states* in an evaluation sequence defined via relation \Rightarrow to intermediate states in an evaluation sequence defined via \rightsquigarrow .

DEFINITION 1. *State \odot, \bar{K} is said to be a well-defined token state if any of the following conditions hold:*

1. $(\mathbf{t}^\tau, M, \mathbf{t}_1, \mathbf{t}_2) \notin \bar{K}$
2. $\bar{K} = \bar{K}' \parallel (\mathbf{t}^\tau, M, \mathbf{t}_1, \mathbf{t}_2) \parallel (\mathbf{t}_1^\tau, \cdot, v_1) \parallel (\mathbf{t}_2^\tau, \cdot, v_2)$
3. $\bar{K} = \bar{K}' \parallel (\mathbf{t}^\tau, M, \mathbf{t}_1, \mathbf{t}_2) \parallel (\mathbf{t}_1^\tau, M_1, \mathcal{F}[e_1]) \parallel (\mathbf{t}_2^\tau, \cdot, v_2)$
4. $\bar{K} = \bar{K}' \parallel (\mathbf{t}^\tau, M, \mathbf{t}_1, \mathbf{t}_2) \parallel (\mathbf{t}_1^\tau, \cdot, v_1) \parallel (\mathbf{t}_2^\tau, M_2, \mathcal{F}[e_2])$
5. $\bar{K} = \bar{K}' \parallel (\mathbf{t}^\tau, M, \mathbf{t}_1, \mathbf{t}_2) \parallel (\mathbf{t}_1^\tau, \cdot, \text{alwaysEvt } v'_1) \parallel (\mathbf{t}_2^\tau, M_2, \mathcal{F}[e_2])$
6. $\bar{K} = \bar{K}' \parallel (\mathbf{t}^\tau, M, \mathbf{t}_1, \mathbf{t}_2) \parallel (\mathbf{t}_1^\tau, M_1, \mathcal{F}[e_1]) \parallel (\mathbf{t}_2^\tau, \cdot, \text{alwaysEvt } v'_2)$
7. $\bar{K} = \bar{K}' \parallel (\mathbf{t}^\tau, M, \mathbf{t}_1, \mathbf{t}_2) \parallel (\mathbf{t}_1^\tau, \cdot, \text{alwaysEvt } v'_1) \parallel (\mathbf{t}_2^\tau, \cdot, \text{alwaysEvt } v'_2)$

Thus, a state is well-defined if the left and right events of a concurrently evaluating isolate are not both partially evaluated. Transform function \mathcal{T} (see Figure 1) maps well-defined token states to states in an evaluation sequence defined via relation \rightsquigarrow by identifying the thread that initiated a parallel isolate along with the two threads evaluating each part and replacing them with a single thread that sequences the two events.

LEMMA 1.2.

If

$$\begin{aligned} & \oplus, \bar{K} \parallel (\mathbf{t}^\tau, M, \mathcal{F}[\text{isolatedEvt}(v_1, v_2)]) \Rightarrow \odot, \bar{K}_1 \Rightarrow \dots \Rightarrow \\ & \odot, \bar{K}_n \Rightarrow \oplus, \bar{K}' \parallel (\mathbf{t}^\tau, M, \mathcal{F}[\text{alwaysEvt}(v'_1, v'_2)]), \end{aligned}$$

then there exists a evaluation sequence

$$\begin{aligned} & \oplus, \bar{K} \parallel (\mathbf{t}^\tau, M, \mathcal{F}[\text{isolatedEvt}(v_1, v_2)]) \Rightarrow \odot, \bar{K}'_1 \Rightarrow \dots \Rightarrow \\ & \odot, \bar{K}'_n \Rightarrow \oplus, \bar{K}' \parallel (\mathbf{t}^\tau, M, \mathcal{F}[\text{alwaysEvt}(v'_1, v'_2)]), \end{aligned}$$

such that all \bar{K}'_i are well-defined token states.

The lemma states that any evaluation sequence defined via relation \Rightarrow starting at state $\oplus, \bar{K} \parallel (\mathbf{t}^\tau, M, \mathcal{F}[\text{isolatedEvt}(v_1, v_2)])$ and ending at state $\oplus, \bar{K}' \parallel (\mathbf{t}^\tau, M, \mathcal{F}[\text{alwaysEvt}(v'_1, v'_2)])$ has an equivalent evaluation sequence (also defined via \Rightarrow), with the same start and end state, such that all intermediate states are well-defined token states. The proof leverages the observation that due to the thread partitioning of the PARALLELFORK rule, a given thread witnesses only the effects of one side of the isolate until the other side completes and releases its partition (via rules RELEASELEFT/RIGHT). Thus the side that completes first and releases its partition could have completed all its communication actions before the other side began evaluating.

The proof of Lemma 1.2 is a proof by contradiction. Assume the following transactional evaluation sequence defined by \Rightarrow does not have an equivalent evaluation sequence where all intermediate states are well defined token states:

$$\begin{aligned} & \oplus, \bar{K} \parallel (\mathbf{t}^\tau, M, \mathcal{F}[\text{isolatedEvt}(v_1, v_2)]) \Rightarrow \ominus_{\frac{t'_1}{t'_1}} \bar{K}_1 \Rightarrow \dots \Rightarrow \\ & \ominus_{\frac{t'_n}{t'_n}} \bar{K}_n \Rightarrow \oplus, \bar{K}' \parallel (\mathbf{t}^\tau, M, \mathcal{F}[\text{alwaysEvt}(v'_1, v'_2)]). \end{aligned}$$

This implies that all communication actions from one side of the isolate whose arguments are being evaluated concurrently can not have occurred before the first communication actions from the other side. Therefore we know the following two statements hold:

- The left side cannot have completed before the first communication action from the right side.
- The right side cannot have completed before the first communication action from the left side.

Note that in the original evaluation sequence, it could not have been the case that the two sides of the isolate communicate with each other since the PARALLELFORK rule puts the two sides of the isolate in different partitions. Thus the evaluation sequence must contain intermediate states \odot, \bar{K}_i , \odot, \bar{K}_j and \odot, \bar{K}_k such that $i < j < k$ and:

- $\odot, \bar{K}_i = \ominus_{\frac{t'_i}{t'_i}} \bar{K} \parallel (\mathbf{t}^\tau, M, \mathbf{t}_1, \mathbf{t}_2) \parallel (\mathbf{t}_1^\tau, \cdot, v_1) \parallel (\mathbf{t}_2^\tau, \cdot, v_2)$
- $\odot, \bar{K}_j = \ominus_{\frac{t'_j}{t'_j}} \bar{K} \parallel (\mathbf{t}^\tau, M, \mathbf{t}_1, \mathbf{t}_2) \parallel (\mathbf{t}_1^\tau, M_1, e_1) \parallel (\mathbf{t}_2^\tau, M_2, e_2)$
- $\odot, \bar{K}_j = \ominus_{\frac{t'_j}{t'_j}} \bar{K} \parallel (\mathbf{t}^\tau, M, \mathbf{t}_1, \mathbf{t}_2) \parallel (\mathbf{t}_1^\tau, \cdot, \text{alwaysEvt } v'_1) \parallel (\mathbf{t}_2^\tau, \cdot, \text{alwaysEvt } v'_2)$

Furthermore, e_1 must perform a communication that transitively allows a communication that e_2 requires to complete, and e_2 must perform a communication that transitively allows a communication that e_1 requires to complete.

Since the evaluation sequence evaluates the right hand side of the isolate (i.e. e_2) to an always event, it must be the case that e_1 communicates with a thread \mathbf{t} , which may then communicate with other threads, $\bar{\mathbf{t}}$, such that either \mathbf{t} or a thread in $\bar{\mathbf{t}}$ subsequently communicates with e_2 allowing it to complete (since e_2 obviously completes in the original evaluation sequence). By the structure of the rules, the only way for thread \mathbf{t} and any thread in $\bar{\mathbf{t}}$, which have communicated (transitively) with the left side of the isolate (i.e. e_1), to communicate with the right side of the isolate (i.e. e_2), is for the left side of the isolate to complete and release those threads from its partition via rule RELEASELEFT. This contradicts the statement that e_2 performs a communication that transitively allows a com-

$$\begin{aligned}
\mathcal{T}(\oplus, \bar{K}) &= \bar{K} \\
\mathcal{T}(\ominus_{\tau}^{\bar{v}}, \bar{K} \parallel (\mathfrak{t}^{\tau}, \mathcal{F} : M, \mathfrak{t}_1, \mathfrak{t}_2) \parallel (\mathfrak{t}_1^{\tau}, \cdot, v_1) \parallel (\mathfrak{t}_2^{\tau}, \cdot, v_2)) &= \begin{cases} \bar{K} \parallel (\mathfrak{t}^{\tau}, M, \mathcal{F}[v_1;_L v_2]) \\ \bar{K} \parallel (\mathfrak{t}^{\tau}, M, \mathcal{F}[v_2;_R v_1]) \end{cases} \\
\mathcal{T}(\ominus_{\tau}^{\bar{v}}, \bar{K} \parallel (\mathfrak{t}^{\tau}, M, \mathfrak{t}_1, \mathfrak{t}_2) \parallel (\mathfrak{t}_1^{\tau}, M_1, \mathcal{F}_1[e_1]) \parallel (\mathfrak{t}_2^{\tau}, \cdot, v_2)) &= \bar{K} \parallel (\mathfrak{t}^{\tau}, M', \mathcal{F}_1[e_1];_L v_2) \quad \begin{array}{l} M_1 = \mathcal{F}'_1 : \dots : \mathcal{F}'_n \cdot \\ M' = \mathcal{F}'_1 : \dots : \mathcal{F}'_n \cdot \end{array} \\
\mathcal{T}(\ominus_{\tau}^{\bar{v}}, \bar{K} \parallel (\mathfrak{t}^{\tau}, M, \mathfrak{t}_1, \mathfrak{t}_2) \parallel (\mathfrak{t}_1^{\tau}, \cdot, v_1) \parallel (\mathfrak{t}_2^{\tau}, M_2, \mathcal{F}_2[e_2])) &= \bar{K} \parallel (\mathfrak{t}^{\tau}, M', \mathcal{F}_2[e_2];_R v_1) \quad \begin{array}{l} M_2 = \mathcal{F}'_1 : \dots : \mathcal{F}'_n \cdot \\ M' = \mathcal{F}'_1 : \dots : \mathcal{F}'_n \cdot \end{array} \\
\mathcal{T}(\ominus_{\tau}^{\bar{v}}, \bar{K} \parallel (\mathfrak{t}^{\tau}, M, \mathfrak{t}_1, \mathfrak{t}_2) \parallel (\mathfrak{t}_1^{\tau}, \cdot, \text{alwaysEvt } v'_1) \parallel (\mathfrak{t}_2^{\tau}, M_2, \mathcal{F}_2[e_2])) &= \bar{K} \parallel (\mathfrak{t}^{\tau}, M', \text{alwaysEvt } v'_1;_L \mathcal{F}_2[e_2]) \quad \begin{array}{l} M_2 = \mathcal{F}'_1 : \dots : \mathcal{F}'_n \cdot \\ M' = \mathcal{F}'_1 : \dots : \mathcal{F}'_n \cdot \end{array} \\
\mathcal{T}(\ominus_{\tau}^{\bar{v}}, \bar{K} \parallel (\mathfrak{t}^{\tau}, M, \mathfrak{t}_1, \mathfrak{t}_2) \parallel (\mathfrak{t}_1^{\tau}, M_1, \mathcal{F}_1[e_1]) \parallel (\mathfrak{t}_2^{\tau}, \cdot, \text{alwaysEvt } v'_2)) &= \bar{K} \parallel (\mathfrak{t}^{\tau}, M', \text{alwaysEvt } v'_2;_R \mathcal{F}_1[e_1]) \quad \begin{array}{l} M_1 = \mathcal{F}'_1 : \dots : \mathcal{F}'_n \cdot \\ M' = \mathcal{F}'_1 : \dots : \mathcal{F}'_n \cdot \end{array} \\
\mathcal{T}(\ominus_{\tau}^{\bar{v}}, \bar{K} \parallel (\mathfrak{t}^{\tau}, M, \mathfrak{t}_1, \mathfrak{t}_2) \parallel (\mathfrak{t}_1^{\tau}, \cdot, \text{alwaysEvt } v'_1) \parallel (\mathfrak{t}_2^{\tau}, \cdot, \text{alwaysEvt } v'_2)) &= \bar{K} \parallel (\mathfrak{t}^{\tau}, M, \text{alwaysEvt } v'_1;_L \text{alwaysEvt } v'_2)
\end{aligned}$$

Figure 1. Transform function from \Rightarrow to \rightsquigarrow

munication that e_1 requires to complete. Thus, by contradiction, we have demonstrated that there must exist an equivalent evaluation sequence defined by \Rightarrow , such that all intermediate states of that sequence are well defined states.

LEMMA 1.3.

If $\odot, \bar{K} \Rightarrow \odot, \bar{K}'$, and \odot, \bar{K} and \odot, \bar{K}' are well-defined token states, then $\mathcal{T}(\odot, \bar{K}) \rightsquigarrow \mathcal{T}(\odot, \bar{K}')$.

The lemma states that every \Rightarrow transition from a well-defined state to a new well-defined state is equivalent to a transition from the transform of the initial state to the transform of the new state.

The proof is a straightforward case analysis on evaluation derivations $\odot, \bar{K} \Rightarrow \odot, \bar{K}'$, where both \odot, \bar{K} and \odot, \bar{K}' are well-defined token states.

The proof of Theorem 1.1 is by induction on the length of the evaluation sequence. Given an evaluation sequence $S_1 \Rightarrow \dots \Rightarrow S_n$, the proof leverages Lemma 1.2 to reason about an evaluation sequence, $S'_1 \Rightarrow \dots \Rightarrow S'_n$, that is equivalent to $S_1 \Rightarrow \dots \Rightarrow S_n$, and contains intermediate states that are well-defined token states. Note that the proof only needs to consider sequences where the start state is \oplus , because if the start state is \ominus , then it must be the case that the sequence is bounded by a larger evaluation sequence that starts with a state with token \oplus .

The base case is evaluation sequences with one intermediate state (i.e. three states). Thus, the events being evaluated by the isolate are both always events. In this sequence, all three states are well-defined token states. Therefore, by Lemma 1.3 there exists an equivalent evaluation defined by \rightsquigarrow on the transforms of the states in the evaluation sequence. The inductive case also follows directly by Lemma 1.3.

1.2 Concurrent Isolates

Next, we prove that transactional evaluation in which all isolates execute their left and right events concurrently (\Longrightarrow) is equivalent to transactional evaluation via \rightsquigarrow . We use a transform function \mathcal{U} , which we define after we state the theorem, to map states that relation \Longrightarrow operates over to states that relation \rightsquigarrow operates over.

THEOREM 1.4. *Capability Soundness*

If $\bar{K} \parallel (\mathfrak{t}_{\kappa}^{\tau, \bar{l}}, M, \mathcal{F}[\text{isolatedEvt}(v_1, v_2)]) \Longrightarrow \bar{K}_1 \Longrightarrow \dots \Longrightarrow \bar{K}_n \Longrightarrow \bar{K}' \parallel (\mathfrak{t}_{\kappa}^{\tau, \bar{l}}, M, \mathcal{F}[\text{alwaysEvt}(v'_1, v'_2)])$, then $\mathcal{U}(\bar{K} \parallel (\mathfrak{t}_{\kappa}^{\tau, \bar{l}}, M, \mathcal{F}[\text{isolatedEvt}(v_1, v_2)])) \rightsquigarrow \bar{K}'_1 \rightsquigarrow \dots \rightsquigarrow \bar{K}'_n \rightsquigarrow \mathcal{U}(\bar{K}' \parallel (\mathfrak{t}_{\kappa}^{\tau, \bar{l}}, M, \mathcal{F}[\text{alwaysEvt}(v'_1, v'_2)]))$.

Theorem 1.4 states that if transactional evaluation defined via relation \Longrightarrow evaluates an isolate to an always event, then transactional evaluation defined via \rightsquigarrow can evaluate the transform of the initial state to the transform of the new state.

Intermediate states in an evaluation sequence that define transactional evaluation via relation \Longrightarrow may contain states that evaluate multiple isolates concurrently. An evaluation sequence that defines transactional evaluation via relation \rightsquigarrow , on the other hand, evaluates isolates sequentially in a single thread. Our proof utilizes a transform function that maps *well-defined label states* in an evaluation sequence defined via relation \Longrightarrow to intermediate states in an evaluation sequence defined via \rightsquigarrow .

DEFINITION 2. A state in the \Longrightarrow sequence is said to be a *well-defined label state* if one of the following conditions holds:

1. $(\mathfrak{t}_{\kappa}^{\tau, \bar{l}}, M, \mathfrak{t}_1, \mathfrak{t}_2) \notin \bar{K}$
2. $\bar{K} = \bar{K}' \parallel (\mathfrak{t}_{\kappa}^{\tau, \bar{l}}, M, \mathfrak{t}_1, \mathfrak{t}_2) \parallel (\mathfrak{t}_{1\kappa_L}^{\tau, \bar{l}}, \cdot, v_1) \parallel (\mathfrak{t}_{2\kappa_R}^{\tau, \bar{l}}, \cdot, v_2)$ and \bar{K}' is well-defined
3. $\bar{K} = \bar{K}' \parallel (\mathfrak{t}_{\kappa}^{\tau, \bar{l}}, M, \mathfrak{t}_1, \mathfrak{t}_2) \parallel (\mathfrak{t}_{1\kappa_L}^{\tau, \bar{l}}, M, \mathcal{F}[e_1]) \parallel (\mathfrak{t}_{2\kappa_R}^{\tau, \bar{l}}, \cdot, v_2)$ and \bar{K}' is well-defined
4. $\bar{K} = \bar{K}' \parallel (\mathfrak{t}_{\kappa}^{\tau, \bar{l}}, M, \mathfrak{t}_1, \mathfrak{t}_2) \parallel (\mathfrak{t}_{1\kappa_L}^{\tau, \bar{l}}, \cdot, v_1) \parallel (\mathfrak{t}_{2\kappa_R}^{\tau, \bar{l}}, M, \mathcal{F}[e_2])$ and \bar{K}' is well-defined
5. $\bar{K} = \bar{K}' \parallel (\mathfrak{t}_{\kappa}^{\tau, \bar{l}}, M, \mathfrak{t}_1, \mathfrak{t}_2) \parallel (\mathfrak{t}_{1\kappa_L}^{\tau, \bar{l}}, \cdot, \text{alwaysEvt } v'_1) \parallel (\mathfrak{t}_{2\kappa_R}^{\tau, \bar{l}}, M, \mathcal{F}[e_2])$ and \bar{K}' is well-defined
6. $\bar{K} = \bar{K}' \parallel (\mathfrak{t}_{\kappa}^{\tau, \bar{l}}, M, \mathfrak{t}_1, \mathfrak{t}_2) \parallel (\mathfrak{t}_{1\kappa_L}^{\tau, \bar{l}}, M, \mathcal{F}[e_1]) \parallel (\mathfrak{t}_{2\kappa_R}^{\tau, \bar{l}}, \cdot, \text{alwaysEvt } v'_2)$ and \bar{K}' is well-defined
7. $\bar{K} = \bar{K}' \parallel (\mathfrak{t}_{\kappa}^{\tau, \bar{l}}, M, \mathfrak{t}_1, \mathfrak{t}_2) \parallel (\mathfrak{t}_{1\kappa_L}^{\tau, \bar{l}}, \cdot, \text{alwaysEvt } v'_1) \parallel (\mathfrak{t}_{2\kappa_R}^{\tau, \bar{l}}, \cdot, \text{alwaysEvt } v'_2)$ and \bar{K}' is well-defined

A state is well-defined if it has not partially evaluated both the left and right event of a concurrently evaluating isolate. Transform

$$\begin{aligned}
\mathcal{U}(\overline{K}) &= \overline{K} \quad \text{if } (\mathfrak{t}_{\kappa}^{\tau, \bar{l}}, M, \mathfrak{t}_1, \mathfrak{t}_2) \notin \overline{K} \\
\mathcal{U}(\overline{K} \parallel (\mathfrak{t}_{\kappa}^{\tau, \bar{l}}, \mathcal{F} : M, \mathfrak{t}_1, \mathfrak{t}_2) \parallel (\mathfrak{t}_{\kappa_L}^{\tau, l, \bar{l}}, \cdot, v_1) \parallel (\mathfrak{t}_{\kappa_R}^{\tau, l, \bar{l}}, \cdot, v_2)) &= \begin{cases} \mathcal{U}(\overline{K}) \parallel (\mathfrak{t}^{\tau}, M, \mathcal{F}[v_1;_L v_2]) \\ \mathcal{U}(\overline{K}) \parallel (\mathfrak{t}^{\tau}, M, \mathcal{F}[v_2;_R v_1]) \end{cases} \\
\mathcal{U}(\overline{K} \parallel (\mathfrak{t}_{\kappa}^{\tau, \bar{l}}, M, \mathfrak{t}_1, \mathfrak{t}_2) \parallel (\mathfrak{t}_{\kappa_L}^{\tau, l, \bar{l}}, M_1, \mathcal{F}_1[e_1]) \parallel (\mathfrak{t}_{\kappa_R}^{\tau, l, \bar{l}}, \cdot, v_2)) &= \mathcal{U}(\overline{K}) \parallel (\mathfrak{t}^{\tau}, M', \mathcal{F}_1[e_1];_L v_2) & \begin{matrix} M_1 = \mathcal{F}'_1 : \dots : \mathcal{F}'_n : \cdot \\ M' = \mathcal{F}'_1 : \dots : \mathcal{F}'_n : M \end{matrix} \\
\mathcal{U}(\overline{K} \parallel (\mathfrak{t}_{\kappa}^{\tau, \bar{l}}, M, \mathfrak{t}_1, \mathfrak{t}_2) \parallel (\mathfrak{t}_{\kappa_L}^{\tau, l, \bar{l}}, \cdot, v_1) \parallel (\mathfrak{t}_{\kappa_R}^{\tau, l, \bar{l}}, M_2, \mathcal{F}_2[e_2])) &= \mathcal{U}(\overline{K}) \parallel (\mathfrak{t}^{\tau}, M', \mathcal{F}_2[e_2];_R v_1) & \begin{matrix} M_2 = \mathcal{F}'_1 : \dots : \mathcal{F}'_n : \cdot \\ M' = \mathcal{F}'_1 : \dots : \mathcal{F}'_n : M \end{matrix} \\
\mathcal{U}(\overline{K} \parallel (\mathfrak{t}_{\kappa}^{\tau, \bar{l}}, M, \mathfrak{t}_1, \mathfrak{t}_2) \parallel (\mathfrak{t}_{\kappa_L}^{\tau, l, \bar{l}}, M_1, \mathcal{F}_1[e_1]) \parallel (\mathfrak{t}_{\kappa_R}^{\tau, l, \bar{l}}, \cdot, \text{alwaysEvt } v_2)) &= \mathcal{U}(\overline{K}) \parallel (\mathfrak{t}^{\tau}, M', \text{alwaysEvt } v'_2;_R \mathcal{F}_1[e_1]) & \begin{matrix} M_1 = \mathcal{F}'_1 : \dots : \mathcal{F}'_n : \cdot \\ M' = \mathcal{F}'_1 : \dots : \mathcal{F}'_n : M \end{matrix} \\
\mathcal{U}(\overline{K} \parallel (\mathfrak{t}_{\kappa}^{\tau, \bar{l}}, M, \mathfrak{t}_1, \mathfrak{t}_2) \parallel (\mathfrak{t}_{\kappa_L}^{\tau, l, \bar{l}}, \cdot, \text{alwaysEvt } v'_1) \parallel (\mathfrak{t}_{\kappa_R}^{\tau, l, \bar{l}}, M_2, \mathcal{F}_2[e_2])) &= \mathcal{U}(\overline{K}) \parallel (\mathfrak{t}^{\tau}, M', \text{alwaysEvt } v'_1;_L \mathcal{F}_2[e_2]) & \begin{matrix} M_2 = \mathcal{F}'_1 : \dots : \mathcal{F}'_n : \cdot \\ M' = \mathcal{F}'_1 : \dots : \mathcal{F}'_n : M \end{matrix} \\
\mathcal{U}(\overline{K} \parallel (\mathfrak{t}_{\kappa}^{\tau, \bar{l}}, M, \mathfrak{t}_1, \mathfrak{t}_2) \parallel (\mathfrak{t}_{\kappa_L}^{\tau, l, \bar{l}}, \cdot, \text{alwaysEvt } v'_1) \parallel (\mathfrak{t}_{\kappa_R}^{\tau, l, \bar{l}}, \cdot, \text{alwaysEvt } v'_2)) &= \mathcal{U}(\overline{K}) \parallel (\mathfrak{t}^{\tau}, M, \text{alwaysEvt } v'_1;_L \text{alwaysEvt } v'_2)
\end{aligned}$$

Figure 2. Transform function from \Longrightarrow to \rightsquigarrow

function \mathcal{U} (see Figure 2) maps well-defined states by identifying the three threads in the state that represent each concurrently executing isolate, and replacing them with a single thread that sequences the two events.

LEMMA 1.5.

If $\overline{K} \parallel (\mathfrak{t}_{\kappa}^{\tau, \bar{l}}, M, \mathcal{F}[\text{isolatedEvt}(v_1, v_2)]) \Longrightarrow \overline{K}_1 \Longrightarrow \dots \Longrightarrow \overline{K}_n \Longrightarrow \overline{K}' \parallel (\mathfrak{t}_{\kappa}^{\tau, \bar{l}}, M, \mathcal{F}[\text{alwaysEvt}(v'_1, v'_2)])$, then there exists an evaluation sequence $\overline{K} \parallel (\mathfrak{t}_{\kappa}^{\tau, \bar{l}}, M, \mathcal{F}[\text{isolatedEvt}(v_1, v_2)]) \Longrightarrow \overline{K}'_1 \Longrightarrow \dots \Longrightarrow \overline{K}'_n \Longrightarrow \overline{K}' \parallel (\mathfrak{t}_{\kappa}^{\tau, \bar{l}}, M, \mathcal{F}[\text{alwaysEvt}(v'_1, v'_2)])$, such that all \overline{K}'_i are well-defined token states.

The lemma states that any evaluation sequence defined via \Longrightarrow starting at state $\overline{K} \parallel (\mathfrak{t}_{\kappa}^{\tau, \bar{l}}, M, \mathcal{F}[\text{isolatedEvt}(v_1, v_2)])$ and ending at state $\overline{K}' \parallel (\mathfrak{t}_{\kappa}^{\tau, \bar{l}}, M, \mathcal{F}[\text{alwaysEvt}(v'_1, v'_2)])$ has an equivalent evaluation sequence (also defined via \Longrightarrow), such that all intermediate states are well-defined label states. Intuitively, we are able to prove this lemma because the CAPABILITY ENRICHMENT rule guarantees that once thread \mathfrak{t} has witnessed the effects of both sides of an isolate, thus imposing a certain serial order on the events of the isolate, the thread is only allowed to communicate with the event that is ordered second. Until that point, communications with the left and right sides of the isolate are independent and can thus have happened in an order such that all communication with one side occurred before all communication with the other.

The proof is by induction on the length of the evaluation sequence. For the base case the evaluation sequence is of length 2, where the two transitions applied are PARALLELFORK and PARALLELJOIN, and the isolate being evaluated takes two always events as arguments, and there are originally no parallel isolates.

$\overline{K} \parallel (\mathfrak{t}_{\kappa}^{\tau, \bar{l}}, M, \mathcal{F}[\text{isolatedEvt}(v_1, v_2)]) \Longrightarrow \overline{K} \parallel (\mathfrak{t}_{\kappa_I}^{\tau, \ell, \bar{l}}, F : M, \mathfrak{t}_1, \mathfrak{t}_2) \parallel (\mathfrak{t}_{1_{\kappa[\ell \rightarrow L]}}^{\tau, \ell, \bar{l}}, \cdot, \text{alwaysEvt } v_1) \parallel (\mathfrak{t}_{2_{\kappa[\ell \rightarrow R]}}^{\tau, \ell, \bar{l}}, \cdot, \text{alwaysEvt } v_2) \Longrightarrow \overline{K} \parallel (\mathfrak{t}_{\kappa}^{\tau, \bar{l}}, M, \mathcal{F}[\text{alwaysEvt}(v'_1, v'_2)])$.

The proof trivially holds because all 3 states in the evaluation sequence are well-defined label states. The first and last state satisfy

condition 1 of Definition 2. The middle intermediate state satisfies condition 7.

For the inductive case, we consider the following two cases:

- Case 1: The evaluation sequence of length n imposes an ordering on the arguments of a given isolate (i.e. the last state in the sequence contains a thread that has capability $\ell \mapsto \text{LR}$ or $\ell \mapsto \text{RL}$ for isolate ℓ).
- Case 2: The evaluation sequence of length n imposes no ordering on the arguments of a given isolate (i.e. all threads in the last state in the sequence have capability $\ell \mapsto \text{L}$ or $\ell \mapsto \text{R}$ for isolate ℓ).

The inductive hypothesis states that for an evaluation sequence of length n , if case 1 holds, then evaluation of the left side of ℓ can occur before the evaluation of the right side of ℓ begins if $\ell \mapsto \text{LR}$ and *vice versa* if $\ell \mapsto \text{RL}$. If case 2 holds then the derivations can be permuted such that the left evaluates completely before the right or the right evaluates before the left.

Note from the structure of the rules that capability enrichment is completely independent of all other threads in the program state, and that capabilities can only be enriched to further constrain the communication and commit of transactional threads. For those rules that depend on thread having certain capabilities, we enumerate all possible capabilities and show the proof condition holds for all possibilities.

1. Case 1: Given evaluation sequence of length n , prove for evaluation sequence of length $n + 1$ resulting from applying derivation $\overline{K} \Longrightarrow \overline{K}'$, such that $\overline{K} = \overline{K}'' \parallel (\mathfrak{t}_{\kappa[\ell \rightarrow \text{LR}]}^{\tau, \ell}, M, e)$. Without loss of generality, we demonstrate the $\ell \mapsto \text{LR}$ case.

$$(a) \text{ RUNTHREAD: } \frac{\overline{K} \parallel (\mathfrak{t}^{\tau}, M, e) \rightsquigarrow \overline{K} \parallel (\mathfrak{t}^{\tau}, M, e')}{\overline{K} \parallel (\mathfrak{t}_{\kappa}^{\tau, \ell}, M, e) \Longrightarrow \overline{K} \parallel (\mathfrak{t}_{\kappa}^{\tau, \ell}, M, e')}$$

Thus, the subcases are the following:

$$i. \text{ STEP RUNTHREAD: } \frac{e \hookrightarrow e'}{\overline{K} \parallel (\mathfrak{t}^{\tau}, M, \mathcal{F}[e]) \rightsquigarrow \overline{K} \parallel (\mathfrak{t}^{\tau}, M, \mathcal{F}[e'])}$$

The inductive hypothesis states that for the evaluation sequence of length n , the left side of isolate ℓ could have been completely evaluated before any derivations ap-

plied to the right side of the isolate. The only reductions are application and channel, both of which do not effect any other thread and are not effected by other threads. Thus, if thread \mathfrak{t} is evaluating the left side of isolate ℓ it can be moved before all derivations that evaluate the right side and still be an equivalent sequence with well-defined label states. If \mathfrak{t} is evaluating any other thread, the proof follows straightforwardly from the inductive hypothesis.

ii. NESTEDSYNC:

$$\overline{K} \parallel (\mathfrak{t}^\tau, M, \mathcal{F}[\text{sync } v]) \rightsquigarrow \overline{K} \parallel (\mathfrak{t}^\tau, \mathcal{F} : M, v)$$

- the derivation only effects the evaluation of thread \mathfrak{t}
- the derivation does not depend on actions of any other threads

If \mathfrak{t} is evaluating the left part of isolate ℓ , then the derivation can be permuted such that it precedes all of the right event's evaluation. It follows from the inductive hypothesis that such an evaluation sequence has only well-defined label states and is equivalent to the original sequence. If \mathfrak{t} is evaluating any other thread, the proof follows straightforwardly from the inductive hypothesis.

iii. NESTEDSYNCCOMPLETE:

$$\overline{K} \parallel (\mathfrak{t}^\tau, \mathcal{F} : M, \text{alwaysEvt } v) \rightsquigarrow \overline{K} \parallel (\mathfrak{t}^\tau, M, \mathcal{F}[v])$$

- the derivation only effects the evaluation of thread \mathfrak{t}
- the derivation does not depend on actions of any other threads

If \mathfrak{t} is evaluating the left part of isolate ℓ , then the derivation can be permuted such that it precedes all of the right event's evaluation. It follows from the inductive hypothesis that such an evaluation sequence has only well-defined label states and is equivalent to the original sequence. If \mathfrak{t} is evaluating any other thread, the proof follows straightforwardly from the inductive hypothesis.

iv. THENALWAYS:

$$\overline{K} \parallel (\mathfrak{t}^\tau, M, \mathcal{F}[\text{thenEvt}(\text{alwaysEvt } (v_1), v_2)]) \rightsquigarrow \overline{K} \parallel (\mathfrak{t}^\tau, M, \mathcal{F}[v_2])$$

- the derivation only effects the evaluation of thread \mathfrak{t}
- the derivation does not depend on actions of any other threads

If \mathfrak{t} is evaluating the left part of isolate ℓ , then the derivation can be permuted such that it precedes all of the right event's evaluation. It follows from the inductive hypothesis that such an evaluation sequence has only well-defined label states and is equivalent to the original sequence. If \mathfrak{t} is evaluating any other thread, the proof follows straightforwardly from the inductive hypothesis.

v. JOINLEFT:

$$\overline{K} \parallel (\mathfrak{t}^\tau, M, \mathcal{F}[\text{alwaysEvt } v_1;_L \text{alwaysEvt } v_2]) \rightsquigarrow \overline{K} \parallel (\mathfrak{t}^\tau, M, \mathcal{F}[\text{alwaysEvt } (v_1, v_2)])$$

- the derivation only effects the evaluation of thread \mathfrak{t}
- the derivation does not depend on actions of any other threads

If \mathfrak{t} is evaluating the left part of isolate ℓ , then the derivation can be permuted such that it precedes all of the right event's evaluation. It follows from the inductive

hypothesis that such an evaluation sequence has only well-defined label states and is equivalent to the original sequence. If \mathfrak{t} is evaluating any other thread, the proof follows straightforwardly from the inductive hypothesis.

vi. JOINRIGHT:

$$\overline{K} \parallel (\mathfrak{t}^\tau, M, \mathcal{F}[\text{alwaysEvt } v_1;_R \text{alwaysEvt } v_2]) \rightsquigarrow \overline{K} \parallel (\mathfrak{t}^\tau, M, \mathcal{F}[\text{alwaysEvt } (v_2, v_1)])$$

- the derivation only effects the evaluation of thread \mathfrak{t}
- the derivation does not depend on actions of any other threads

If \mathfrak{t} is evaluating the left part of isolate ℓ , then the derivation can be permuted such that it precedes all of the right event's evaluation. It follows from the inductive hypothesis that such an evaluation sequence has only well-defined label states and is equivalent to the original sequence. If \mathfrak{t} is evaluating any other thread, the proof follows straightforwardly from the inductive hypothesis.

(b) PARALLELFORK:

$$\overline{K} \parallel (\mathfrak{t}^{\tau, \ell}, M, F[\text{isolatedEvt}(v_1, v_2)]) \Longrightarrow \overline{K} \parallel (\mathfrak{t}^{\tau, \ell}, F : M, \mathfrak{t}_1, \mathfrak{t}_2) \parallel (\mathfrak{t}_{1\kappa[\ell \rightarrow L]}^{\tau, \ell}, \cdot, v_1) \parallel (\mathfrak{t}_{2\kappa[\ell \rightarrow R]}^{\tau, \ell}, \cdot, v_2)$$

The derivation evaluates one thread in the program state (i.e. the one performing the isolate), and adds two new threads to evaluate the left and right events of the new isolate. If thread \mathfrak{t} is evaluating the left event of an isolate, the derivation can be permuted to precede the evaluation of the right event. By the inductive hypothesis, it holds that the permuted sequence would comprise of only well-defined label states and be equivalent to the original evaluation sequence. If \mathfrak{t} evaluates another thread, it follows from the inductive hypothesis that the new sequence has only well-defined label states, because in the new state both sides of the isolate have yet to be evaluated, and thus is a well-defined label state.

(c) PARALLELJOIN:

$$\overline{K} = (\mathfrak{t}_{1\kappa_1[\ell \rightarrow L]}^{\tau, \ell_1}, M_1, e_1) \parallel \dots \parallel (\mathfrak{t}_{n\kappa_n[\ell \rightarrow R]}^{\tau, \ell_n}, M_n, e_n) \parallel \overline{K}'$$

$$\overline{K}'' = (\mathfrak{t}_{1\kappa_1[\ell \rightarrow \cdot]}^{\tau, \ell_1}, M_1, e_1) \parallel \dots \parallel (\mathfrak{t}_{n\kappa_n[\ell \rightarrow \cdot]}^{\tau, \ell_n}, M_n, e_n) \parallel \overline{K}'$$

$$\overline{K} \parallel (\mathfrak{t}_{\kappa_I}^{\tau, \ell, \bar{\ell}}, F : M, \mathfrak{t}_1, \mathfrak{t}_2) \parallel (\mathfrak{t}_{1\kappa[\ell \rightarrow L]}^{\tau, \ell, \bar{\ell}}, M, \text{alwaysEvt } v_1) \parallel (\mathfrak{t}_{2\kappa[\ell \rightarrow R]}^{\tau, \ell, \bar{\ell}}, M, \text{alwaysEvt } v_2) \Longrightarrow \overline{K}'' \parallel (\mathfrak{t}_{\kappa[\ell \rightarrow \cdot]}^{\tau, \ell, \bar{\ell}}, M, F[\text{alwaysEvt } (v_1, v_2)])$$

The derivation combines three states in the program state that are evaluating a parallel isolate into a single isolate once both the left and right event of the isolate have completed. For the rule to apply, the rule requires all threads that have communicated with isolate ℓ to have the same capability for ℓ . In the new state the capability of ℓ is erased because the isolate has completed. If thread \mathfrak{t} is itself the left event of an isolate, then by the inductive hypothesis, all of its actions can be permuted to precede the right event's evaluation. As argued before, all the capability enrichment rules that have been applied to make the commit possible, can be permuted such that they precede all of the left event's evaluation. By the inductive hypothesis, the resulting evaluation sequence would be equivalent to the original sequence and contain only well-defined label states. If \mathfrak{t} is not evaluating the left event of isolate ℓ , the proof follows directly from the

inductive hypothesis because the new state is clearly a well-defined label state.

(d) SENDRECV:

$$\frac{(\mathfrak{t}_{1_{\kappa}}^{\tau}, M_1, F_1[\text{sendEvt}(c, v)]) \parallel (\mathfrak{t}_{2_{\kappa}}^{\tau}, M_2, F_2[\text{recvEvt}(c)]) \parallel \bar{K} \implies (\mathfrak{t}_{1_{\kappa}}^{\tau}, M_1, F_1[\text{alwaysEvt unit}]) \parallel (\mathfrak{t}_{2_{\kappa}}^{\tau}, M_2, F_2[\text{alwaysEvt } v]) \parallel \bar{K}}$$

First we prove by contradiction that in the resulting state, $(\mathfrak{t}_{1_{\kappa}}^{\tau}, M_1, F_1[\text{alwaysEvt unit}]) \parallel (\mathfrak{t}_{2_{\kappa}}^{\tau}, M_2, F_2[\text{alwaysEvt } v]) \parallel \bar{K}$, there are no threads such that $\ell \mapsto \text{RL}$. We know that $\bar{K} = \bar{K}'' \parallel (\mathfrak{t}_{\kappa[\ell \mapsto \text{LR}]}^{\tau, \bar{\ell}}, M, e)$. Assume there is a thread in \bar{K}'' such that $\ell \mapsto \text{RL}$. Thus: $\bar{K} = \bar{K}''' \parallel (\mathfrak{t}_{\kappa[\ell \mapsto \text{LR}]}^{\tau, \bar{\ell}}, M, e) \parallel (\mathfrak{t}'^{\tau', \bar{\ell}'}, M', e')$. For isolate ℓ to commit (via rule PARALLELJOIN) it must be the case that all threads that have capabilities for ℓ have the same capability.

- \mathfrak{t} has capability $\ell \mapsto \text{LR}$.
- \mathfrak{t}' has capability $\ell \mapsto \text{RL}$.
- The CAPABILITYENRICHMENT rule, which asserts that tags can be enriched, cannot enrich an LR capability to an RL capability, or *vice versa*.
- For isolate ℓ to commit (via rule PARALLELJOIN) it must be the case that all threads that have capabilities for ℓ have the same capability.

This implies the isolate ℓ never commits. This contradicts the original evaluation sequence which was able to join the isolate (i.e. commit). Thus, there does not exist a \mathfrak{t}' with capability $\ell \mapsto \text{RL}$ in the resulting state of the derivation.

- $\ell \mapsto \text{L}$: Threads \mathfrak{t}_1 and \mathfrak{t}_2 are not evaluating either side of an isolate. Threads \mathfrak{t}_1 and \mathfrak{t}_2 have capability $\ell \mapsto \text{L}$, and thus have not communicated with the right event of an isolate, or with any threads that have communicated with the right event. Thus, the communication can be permuted to occur before the evaluation of the right event, yielding an equivalent evaluation sequence with well-defined label states.
- $\ell \mapsto \text{R}$: Threads \mathfrak{t}_1 and \mathfrak{t}_2 have capability $\ell \mapsto \text{R}$. It holds from the inductive hypothesis that all intermediate states with capability $\ell \mapsto \text{L}$ can occur before this derivation, yielding an equivalent sequence with well-defined label states.
- $\ell \mapsto \text{LR}$: Thread \mathfrak{t}_1 and \mathfrak{t}_2 have capability $\ell \mapsto \text{LR}$. The capability of the thread must have been L at some point. By the inductive hypothesis, every evaluation step evaluating the left side can be permuted to occur before the transition to LR and all of the rights evaluation. Thus, the new sequence contains all well-defined label states.

(e) SENDRECVISOLATELEFT:

$$\frac{\iota \in \{\text{L}, \text{RL}\} \quad \forall \ell' \notin \bar{\ell} \quad \kappa(\ell') = \cdot}{(\mathfrak{t}_{1_{\kappa[\ell \mapsto \text{L}]}}^{\tau, \ell, \bar{\ell}}, M_1, F_1[\text{sendEvt}(c, v)]) \parallel (\mathfrak{t}_{2_{\kappa[\ell \mapsto \text{L}]}}, M_2, F_2[\text{recvEvt}(c)]) \parallel \bar{K} \implies (\mathfrak{t}_{1_{\kappa[\ell \mapsto \text{L}]}}^{\tau, \ell, \bar{\ell}}, M_1, F_1[\text{alwaysEvt unit}]) \parallel (\mathfrak{t}_{2_{\kappa[\ell \mapsto \text{L}]}}^{\tau}, M_2, F_2[\text{alwaysEvt } v]) \parallel \bar{K}}$$

From the structure of the derivation (i.e. condition $\iota \in \{\text{L}, \text{RL}\}$), we know $\iota = \text{L}$ or RL . We prove that $\iota \neq \text{RL}$. The proof by contradiction in case 1d applies.

- $\ell \mapsto \text{L}$: Thread \mathfrak{t}_1 is evaluating the left side of isolate ℓ . We proved that the new state cannot have a thread with capability $\ell \mapsto \text{RL}$. Thus, $\iota = \text{L}$, implying the communicating threads have not communicated with the right event of an isolate, or with any threads that have communicated with the right event. Thus, the communication can be permuted to occur before the evaluation of the right event, yielding an equivalent evaluation sequence with well-defined label states.

(f) RECVSENDISOLATELEFT:

$$\frac{\iota \in \{\text{L}, \text{RL}\} \quad \forall \ell' \notin \bar{\ell} \quad \kappa(\ell') = \cdot}{(\mathfrak{t}_{1_{\kappa[\ell \mapsto \text{L}]}}, M_1, F_1[\text{recvEvt}(c)]) \parallel (\mathfrak{t}_{2_{\kappa[\ell \mapsto \text{L}]}}, M_2, F_2[\text{sendEvt}(c, v)]) \parallel \bar{K} \implies (\mathfrak{t}_{1_{\kappa[\ell \mapsto \text{L}]}}, M_1, F_1[\text{alwaysEvt } v]) \parallel (\mathfrak{t}_{2_{\kappa[\ell \mapsto \text{L}]}}, M_2, F_2[\text{alwaysEvt unit}]) \parallel \bar{K}}$$

The proof for this derivation is the same as the previous case. The only difference is that the left isolate in this case is receiving instead of sending.

(g) SENDRECVISOLATERIGHT:

$$\frac{\iota \in \{\text{R}, \text{LR}\} \quad \forall \ell' \notin \bar{\ell} \quad \kappa(\ell') = \cdot}{(\mathfrak{t}_{1_{\kappa[\ell \mapsto \text{R}]}}^{\tau, \ell, \bar{\ell}}, M_1, F_1[\text{sendEvt}(c, v)]) \parallel (\mathfrak{t}_{2_{\kappa[\ell \mapsto \text{L}]}}, M_2, F_2[\text{recvEvt}(c)]) \parallel \bar{K} \implies (\mathfrak{t}_{1_{\kappa[\ell \mapsto \text{R}]}}^{\tau, \ell, \bar{\ell}}, M_1, F_1[\text{alwaysEvt unit}]) \parallel (\mathfrak{t}_{2_{\kappa[\ell \mapsto \text{L}]}}, M_2, F_2[\text{alwaysEvt } v]) \parallel \bar{K}}$$

From the structure of the rule (i.e. condition $\iota \in \{\text{R}, \text{LR}\}$), it follows that $\iota = \text{R}$ or LR .

- $\iota = \text{R}$: If thread \mathfrak{t}_2 has capability $\ell \mapsto \text{R}$, then \mathfrak{t}_2 is guaranteed to not have communicated with the left side of the isolate. By the inductive hypothesis, it follows that derivations that evaluate the left side of isolate ℓ can occur before this derivation, thus yielding an equivalent sequence with well-defined label states.
- $\iota = \text{LR}$: If thread \mathfrak{t}_2 has capability $\ell \mapsto \text{LR}$, then at some point \mathfrak{t}_2 had capability $\ell \mapsto \text{L}$. By the inductive hypothesis, it follows that every evaluation step evaluating the left side can be permuted to occur before the transition to LR and all of the rights evaluation; thus, resulting in an equivalent sequence with well-defined label states.

(h) RECVSENDISOLATERIGHT:

$$\frac{\iota \in \{\text{R}, \text{LR}\} \quad \forall \ell' \notin \bar{\ell} \quad \kappa(\ell') = \cdot}{(\mathfrak{t}_{1_{\kappa[\ell \mapsto \text{R}]}}^{\tau, \ell, \bar{\ell}}, M_1, F_1[\text{recvEvt}(c)]) \parallel (\mathfrak{t}_{2_{\kappa[\ell \mapsto \text{L}]}}, M_2, F_2[\text{sendEvt}(c, v)]) \parallel \bar{K} \implies (\mathfrak{t}_{1_{\kappa[\ell \mapsto \text{R}]}}^{\tau, \ell, \bar{\ell}}, M_1, F_1[\text{alwaysEvt } v]) \parallel (\mathfrak{t}_{2_{\kappa[\ell \mapsto \text{L}]}}, M_2, F_2[\text{alwaysEvt unit}]) \parallel \bar{K}}$$

The proof for this derivation is the same as the previous case. The only difference is that the right isolate in this case is receiving instead of sending.

(i) SENDRECVISOLATERIGHTISOLATELEFT:

$$\frac{\iota \in \{\text{R}, \text{LR}\} \quad \iota' \in \{\text{L}, \text{RL}\} \quad \forall \ell \notin \bar{\ell} \cup \bar{\ell}' \quad \kappa(\ell) = \cdot}{(\mathfrak{t}_{1_{\kappa[\ell \mapsto \text{R}, \ell' \mapsto \text{L}]}}, M_1, F_1[\text{sendEvt}(c, v)]) \parallel (\mathfrak{t}_{2_{\kappa[\ell \mapsto \text{L}, \ell' \mapsto \text{L}]}}, M_2, F_2[\text{recvEvt}(c)]) \parallel \bar{K} \implies (\mathfrak{t}_{1_{\kappa[\ell \mapsto \text{R}, \ell' \mapsto \text{L}]}}, M_1, F_1[\text{alwaysEvt unit}]) \parallel (\mathfrak{t}_{2_{\kappa[\ell \mapsto \text{L}, \ell' \mapsto \text{L}]}}, M_2, F_2[\text{alwaysEvt } v]) \parallel \bar{K}}$$

From structure of the derivation, $\iota = L$ or LR and $\iota' = R$ or RL . Two separate isolates ℓ and ℓ' are communicating. We prove that the left and right events of both ℓ and ℓ' can be serialized.

i. ($\iota = R, \iota' = L$): It holds from the inductive hypothesis that all intermediate states that communicate with the left of ℓ can occur before evaluation of the right side of ℓ , including this communication step. By the structure of the rules, threads can only communicate to one isolate at a time (not both ℓ and ℓ'). Therefore we can apply the inductive hypothesis to isolate ℓ' as well. Thus, this derivation can be permuted to precede the evaluation of the right side of the isolate. Permuting derivations as described yields an equivalent sequence with well-defined label states.

ii. ($\iota = R, \iota' = RL$): Same as case $\iota = R, \iota' = L$.

iii. ($\iota = LR, \iota' = L$): By the structure of the rules, the fact that $\ell \mapsto LR$ implies that at some point in the thread $\ell \mapsto L$. By the inductive hypothesis, it holds that all derivations evaluating the left of ℓ can be permuted to precede evaluation the transition to LR and all evaluation of the right. By the structure of the rules, threads can only communicate to one isolate at a time (not both ℓ and ℓ'). Therefore we can apply the inductive hypothesis to isolate ℓ' as well. Thus, this derivation can be permuted to precede the evaluation of the right side of the isolate. Permuting derivations as described yields an equivalent sequence with well-defined label states.

iv. ($\iota = LR, \iota' = RL$): Same as case $\iota = LR, \iota' = L$.

$$(j) \text{ RECVSENDISOLATERIGHTISOLATELEFT: } \frac{\iota \in \{R, LR\} \quad \iota' \in \{L, RL\} \quad \forall \ell \notin \bar{\ell} \cup \bar{\ell}' \quad \kappa(\ell) = \cdot}{\begin{array}{l} (\mathfrak{t}_{\kappa[\ell \mapsto R, \ell' \mapsto \iota']}^{\tau, \ell, \bar{\ell}}, M_1, F_1[\text{recvEvt}(c)]) \parallel \\ (\mathfrak{t}_{\kappa[\ell' \mapsto L, \ell \mapsto \iota]}^{\tau, \ell', \bar{\ell}'}, M_2, F_2[\text{sendEvt}(c, v)]) \parallel \bar{K} \implies \\ (\mathfrak{t}_{\kappa[\ell \mapsto R, \ell' \mapsto \iota']}^{\tau, \ell, \bar{\ell}}, M_1, F_1[\text{alwaysEvt } v]) \parallel \\ (\mathfrak{t}_{\kappa[\ell' \mapsto L, \ell \mapsto \iota]}^{\tau, \ell', \bar{\ell}'}, M_2, F_2[\text{alwaysEvt unit}]) \parallel \bar{K} \end{array}}$$

The proof for this derivation is the same as the previous case. The only difference is that the right isolate in this case is receiving and the left is sending, instead of the right sending and the left receiving.

2. Case 2: Given evaluation sequence of length n , prove for evaluation sequence of length $n + 1$ resulting from applying derivation $\bar{K} \implies \bar{K}'$, such that $(\mathfrak{t}_{\kappa[\ell \mapsto RL]}^{\tau, \bar{\ell}}, M, e) \notin \bar{K}$ and $(\mathfrak{t}_{\kappa[\ell \mapsto LR]}^{\tau, \bar{\ell}}, M, e) \notin \bar{K}$. Thus, threads have not witnessed the effects of both the left and right event of isolate ℓ .

$$(a) \text{ RUNTHREAD: } \frac{\bar{K} \parallel (\mathfrak{t}^{\tau}, M, e) \rightsquigarrow \bar{K} \parallel (\mathfrak{t}^{\tau}, M, e')}{\bar{K} \parallel (\mathfrak{t}_{\kappa}^{\tau, \bar{\ell}}, M, e) \implies \bar{K} \parallel (\mathfrak{t}_{\kappa}^{\tau, \bar{\ell}}, M, e')}$$

Thus, the subcases are the following:

i. STEPRUNTHREAD:

$$\frac{e \hookrightarrow e'}{\bar{K} \parallel (\mathfrak{t}^{\tau}, M, \mathcal{F}[e]) \rightsquigarrow \bar{K} \parallel (\mathfrak{t}^{\tau}, M, \mathcal{F}[e'])}$$

The inductive hypothesis states that for the evaluation sequence of length n , the left side of isolate ℓ could have been completely evaluated before any derivations applied to the right side of the isolate. The only reductions are application and channel, both of which do not effect any other thread and are not effected by other threads. Thus, if thread \mathfrak{t} is evaluating the left side of isolate ℓ

it can be moved before all derivations that evaluate the right side and still be an equivalent sequence with well-defined label states. If \mathfrak{t} is evaluating any other thread, the proof follows straightforwardly from the inductive hypothesis.

ii. NESTEDSYNC:

$$\frac{}{\bar{K} \parallel (\mathfrak{t}^{\tau}, M, \mathcal{F}[\text{sync } v]) \rightsquigarrow \bar{K} \parallel (\mathfrak{t}^{\tau}, \mathcal{F} : M, v)}$$

- the derivation only effects the evaluation of thread \mathfrak{t}
- the derivation does not depend on actions of any other threads

If \mathfrak{t} is evaluating the left part of isolate ℓ , then the derivation can be permuted such that it precedes all of the right event's evaluation. It follows from the inductive hypothesis that such an evaluation sequence has only well-defined label states and is equivalent to the original sequence. If \mathfrak{t} is evaluating any other thread, the proof follows straightforwardly from the inductive hypothesis.

iii. NESTEDSYNCCOMPLETE:

$$\frac{}{\bar{K} \parallel (\mathfrak{t}^{\tau}, \mathcal{F} : M, \text{alwaysEvt } v) \rightsquigarrow \bar{K} \parallel (\mathfrak{t}^{\tau}, M, \mathcal{F}[v])}$$

- the derivation only effects the evaluation of thread \mathfrak{t}
- the derivation does not depend on actions of any other threads

If \mathfrak{t} is evaluating the left part of isolate ℓ , then the derivation can be permuted such that it precedes all of the right event's evaluation. It follows from the inductive hypothesis that such an evaluation sequence has only well-defined label states and is equivalent to the original sequence. If \mathfrak{t} is evaluating any other thread, the proof follows straightforwardly from the inductive hypothesis.

iv. THENALWAYS:

$$\frac{}{\bar{K} \parallel (\mathfrak{t}^{\tau}, M, \mathcal{F}[\text{thenEvt}(\text{alwaysEvt}(v_1), v_2)]) \rightsquigarrow \bar{K} \parallel (\mathfrak{t}^{\tau}, M, \mathcal{F}[v_2 \ v_1])}$$

- the derivation only effects the evaluation of thread \mathfrak{t}
- the derivation does not depend on actions of any other threads

If \mathfrak{t} is evaluating the left part of isolate ℓ , then the derivation can be permuted such that it precedes all of the right event's evaluation. It follows from the inductive hypothesis that such an evaluation sequence has only well-defined label states and is equivalent to the original sequence. If \mathfrak{t} is evaluating any other thread, the proof follows straightforwardly from the inductive hypothesis.

v. JOINLEFT:

$$\frac{}{\bar{K} \parallel (\mathfrak{t}^{\tau}, M, \mathcal{F}[\text{alwaysEvt } v_{1;L} \ \text{alwaysEvt } v_2]) \rightsquigarrow \bar{K} \parallel (\mathfrak{t}^{\tau}, M, \mathcal{F}[\text{alwaysEvt}(v_1, v_2)])}$$

- the derivation only effects the evaluation of thread \mathfrak{t}
- the derivation does not depend on actions of any other threads

If \mathfrak{t} is evaluating the left part of isolate ℓ , then the derivation can be permuted such that it precedes all of the right event's evaluation. It follows from the inductive hypothesis that such an evaluation sequence has only well-defined label states and is equivalent to the original sequence. If \mathfrak{t} is evaluating any other thread, the proof follows straightforwardly from the inductive hypothesis.

vi. JOINRIGHT:

$$\frac{\overline{K} \parallel (\mathfrak{t}^\tau, M, \mathcal{F}[\text{alwaysEvt } v_1;_R \text{ alwaysEvt } v_2]) \rightsquigarrow \overline{K} \parallel (\mathfrak{t}^\tau, M, \mathcal{F}[\text{alwaysEvt } (v_2, v_1)])}{\text{the derivation only effects the evaluation of thread } \mathfrak{t}}$$

- the derivation only effects the evaluation of thread \mathfrak{t}
- the derivation does not depend on actions of any other threads

If \mathfrak{t} is evaluating the left part of isolate ℓ , then the derivation can be permuted such that it precedes all of the right event's evaluation. It follows from the inductive hypothesis that such an evaluation sequence has only well-defined label states and is equivalent to the original sequence. If \mathfrak{t} is evaluating any other thread, the proof follows straightforwardly from the inductive hypothesis.

(b) PARALLELFORK:

$$\frac{\ell \text{ fresh} \quad \overline{K} \parallel (\mathfrak{t}_{\kappa}^{\tau, \ell}, M, F[\text{isolatedEvt}(v_1, v_2)]) \Longrightarrow \overline{K} \parallel (\mathfrak{t}_{\kappa}^{\tau, \ell}, F : M, \mathfrak{t}_1, \mathfrak{t}_2) \parallel (\mathfrak{t}_{1_{\kappa[\ell \rightarrow L]}}^{\tau, \ell}, \cdot, v_1) \parallel (\mathfrak{t}_{2_{\kappa[\ell \rightarrow R]}}^{\tau, \ell}, \cdot, v_2)}{\text{The derivation evaluates one thread in the program state (i.e. the one performing the isolate), and adds two new threads to evaluate the left and right events of the new isolate. If thread } \mathfrak{t} \text{ is evaluating the left event of an isolate, the derivation can be permuted to precede the evaluation of the right event. By the inductive hypothesis it holds that the permuted sequence would comprise of only well-defined label states and be equivalent to the original evaluation sequence. If } \mathfrak{t} \text{ evaluates another thread, it follows from the inductive hypothesis that the new sequence has only well-defined label states, because in the new state both sides of the isolate have yet to be evaluated, and thus is a well-defined label state.}}$$

The derivation evaluates one thread in the program state (i.e. the one performing the isolate), and adds two new threads to evaluate the left and right events of the new isolate. If thread \mathfrak{t} is evaluating the left event of an isolate, the derivation can be permuted to precede the evaluation of the right event. By the inductive hypothesis it holds that the permuted sequence would comprise of only well-defined label states and be equivalent to the original evaluation sequence. If \mathfrak{t} evaluates another thread, it follows from the inductive hypothesis that the new sequence has only well-defined label states, because in the new state both sides of the isolate have yet to be evaluated, and thus is a well-defined label state.

(c) PARALLELJOIN:

$$\frac{\begin{aligned} \overline{K} &= (\mathfrak{t}_{1_{\kappa_1[\ell \rightarrow L]}}^{\tau, \ell_1}, M_1, e_1) \parallel \dots \parallel (\mathfrak{t}_{n_{\kappa_n[\ell \rightarrow R]}}^{\tau, \ell_n}, M_n, e_n) \parallel \overline{K}' \\ \overline{K}'' &= (\mathfrak{t}_{1_{\kappa_1[\ell \rightarrow \cdot]}}^{\tau, \ell_1}, M_1, e_1) \parallel \dots \parallel (\mathfrak{t}_{n_{\kappa_n[\ell \rightarrow \cdot]}}^{\tau, \ell_n}, M_n, e_n) \parallel \overline{K}' \end{aligned}}{\overline{K} \parallel (\mathfrak{t}_{\kappa_I}^{\tau, \ell, \ell}, F : M, \mathfrak{t}_1, \mathfrak{t}_2) \parallel (\mathfrak{t}_{1_{\kappa[\ell \rightarrow L]}}^{\tau, \ell, \ell}, M, \text{alwaysEvt } v_1) \parallel (\mathfrak{t}_{2_{\kappa[\ell \rightarrow R]}}^{\tau, \ell, \ell}, M, \text{alwaysEvt } v_2) \Longrightarrow \overline{K}'' \parallel (\mathfrak{t}_{\kappa[\ell \rightarrow \cdot]}^{\tau, \ell, \ell}, M, F[\text{alwaysEvt } (v_1, v_2)])}$$

The derivation combines three states in the program state that are evaluating a parallel isolate into a single isolate once both the left and right event of the isolate have completed. For the rule to apply, the rule requires all threads that have communicated with isolate ℓ to have the same capability for ℓ . In the new state the capability of ℓ is erased because the isolate has completed. If thread \mathfrak{t} is itself the left event of an isolate, then by the inductive hypothesis, all of its actions can be permuted to precede the right event's evaluation. As argued before, all the capability enrichment rules that have been applied to make the commit possible, can be permuted such that they precede all of the left event's evaluation. By the inductive hypothesis, the resulting evaluation sequence would be equivalent to the original sequence and contain only well-defined label states. If \mathfrak{t} is not evaluating the left event of isolate ℓ , the proof follows directly from the inductive hypothesis because the new state is clearly a well-defined label state.

(d) SENDRECV:

$$\frac{(\mathfrak{t}_{1_{\kappa}}^{\tau, \cdot}, M_1, F_1[\text{sendEvt}(c, v)]) \parallel (\mathfrak{t}_{2_{\kappa}}^{\tau, \cdot}, M_2, F_2[\text{recvEvt}(c)]) \parallel \overline{K} \Longrightarrow (\mathfrak{t}_{1_{\kappa}}^{\tau, \cdot}, M_1, F_1[\text{alwaysEvt unit}]) \parallel (\mathfrak{t}_{2_{\kappa}}^{\tau, \cdot}, M_2, F_2[\text{alwaysEvt } v]) \parallel \overline{K}}$$

i. $\ell \mapsto L$: From the inductive hypothesis, it follows that all threads in the new program state have seen the effects of neither the left or right of ℓ , or the effects of only one of them. Thus, the evaluation steps of the left and right of ℓ can be permuted such that the evaluation is serialized in either order (i.e. left first then right or right first and then left). Thus it follows that there exists a evaluation sequence equivalent to the original with all well-defined label states.

ii. $\ell \mapsto R$: Same as the case for $\ell \mapsto L$

iii. $\ell \mapsto LR$: If the capability for ℓ is LR it must be the case that at some point, it was enriched from L. By the inductive hypothesis, every evaluation step evaluating the left side can be permuted to occur before the transition to LR and all evaluation derivations of the right side, thus creating an evaluation sequence with well-defined label states that is equivalent to the original evaluation sequence.

iv. $\ell \mapsto RL$: If the capability for ℓ is RL it must be the case that at some point, it was enriched from R. By the inductive hypothesis, every evaluation step evaluating the right side can be permuted to occur before the transition to RL and all evaluation derivations of the left side, thus creating an evaluation sequence with well-defined label states that is equivalent to the original evaluation sequence.

(e) SENDRECVISOLATELEFT:

$$\frac{\iota \in \{L, RL\} \quad \forall \ell' \notin \ell \quad \kappa(\ell) \cdot \cdot}{(\mathfrak{t}_{1_{\kappa[\ell \rightarrow L]}}^{\tau, \ell, \ell}, M_1, F_1[\text{sendEvt}(c, v)]) \parallel (\mathfrak{t}_{2_{\kappa[\ell \rightarrow \iota]}}^{\tau, \cdot}, M_2, F_2[\text{recvEvt}(c)]) \parallel \overline{K} \Longrightarrow (\mathfrak{t}_{1_{\kappa[\ell \rightarrow L]}}^{\tau, \ell, \ell}, M_1, F_1[\text{alwaysEvt unit}]) \parallel (\mathfrak{t}_{2_{\kappa[\ell \rightarrow \iota]}}^{\tau, \cdot}, M_2, F_2[\text{alwaysEvt } v]) \parallel \overline{K}}$$

i. $\iota = L$: If thread \mathfrak{t}_2 has capability $\ell \mapsto L$, then it follows from the inductive hypothesis that all threads in the sequence have only seen the effects of zero or one side of the isolate. Therefore, the evaluation of either side could precede the evaluation of the other to create an equivalent evaluation sequence that has all well-defined label states.

ii. $\iota = RL$: If the capability for ℓ is RL it must be the case that at some point, it was enriched from R. By the inductive hypothesis, every evaluation step evaluating the right side can be permuted to occur before the transition to RL and all evaluation derivations of the left side, thus creating an evaluation sequence with well-defined label states that is equivalent to the original evaluation sequence.

$$(f) \text{ RECVSENDISOLATELEFT:}$$

$$\frac{\iota \in \{L, RL\} \quad \forall \ell' \notin \bar{\ell} \quad \kappa(\ell) = \cdot}{\begin{array}{l} (\mathfrak{t}_{1\kappa[\ell \mapsto L]}^{\tau, \ell, \bar{\ell}}, M_1, F_1[\text{recvEvt}(c)]) \parallel \\ (\mathfrak{t}_{2\kappa[\ell \mapsto \iota]}^{\tau, \cdot}, M_2, F_2[\text{sendEvt}(c, v)]) \parallel \bar{K} \implies \\ (\mathfrak{t}_{1\kappa[\ell \mapsto L]}^{\tau, \ell, \bar{\ell}}, M_1, F_1[\text{alwaysEvt } v]) \parallel \\ (\mathfrak{t}_{2\kappa[\ell \mapsto \iota]}^{\tau, \cdot}, M_2, F_2[\text{alwaysEvt unit}]) \parallel \bar{K} \end{array}}$$

The proof for this derivation is the same as the previous case. The only difference is that the left isolate in this case is receiving instead of sending.

$$(g) \text{ SENDRECVISOLATERIGHT:}$$

$$\frac{\iota \in \{R, LR\} \quad \forall \ell' \notin \bar{\ell} \quad \kappa(\ell) = \cdot}{\begin{array}{l} (\mathfrak{t}_{1\kappa[\ell \mapsto R]}^{\tau, \ell, \bar{\ell}}, M_1, F_1[\text{sendEvt}(c, v)]) \parallel \\ (\mathfrak{t}_{2\kappa[\ell \mapsto \iota]}^{\tau, \cdot}, M_2, F_2[\text{recvEvt}(c)]) \parallel \bar{K} \implies \\ (\mathfrak{t}_{1\kappa[\ell \mapsto R]}^{\tau, \ell, \bar{\ell}}, M_1, F_1[\text{alwaysEvt unit}]) \parallel \\ (\mathfrak{t}_{2\kappa[\ell \mapsto \iota]}^{\tau, \cdot}, M_2, F_2[\text{alwaysEvt } v]) \parallel \bar{K} \end{array}}$$

- i. $\iota = R$: If thread \mathfrak{t}_2 has capability $\ell \mapsto R$, then it follows from the inductive hypothesis that all threads in the sequence have only seen the effects of zero or one side of the isolate. Therefore, the evaluation of either side could precede the evaluation of the other to create an equivalent evaluation sequence that has all well-defined label states.
- ii. $\iota = LR$: If the capability for ℓ is LR it must be the case that at some point, it was enriched from L. By the inductive hypothesis, every evaluation step evaluating the left side can be permuted to occur before the transition to LR and all evaluation derivations of the right side, thus creating an evaluation sequence with well-defined label states that is equivalent to the original evaluation sequence.

$$(h) \text{ RECVSENDISOLATERIGHT:}$$

$$\frac{\iota \in \{R, LR\} \quad \forall \ell' \notin \bar{\ell} \quad \kappa(\ell) = \cdot}{\begin{array}{l} (\mathfrak{t}_{1\kappa[\ell \mapsto R]}^{\tau, \ell, \bar{\ell}}, M_1, F_1[\text{recvEvt}(c)]) \parallel \\ (\mathfrak{t}_{2\kappa[\ell \mapsto \iota]}^{\tau, \cdot}, M_2, F_2[\text{sendEvt}(c, v)]) \parallel \bar{K} \implies \\ (\mathfrak{t}_{1\kappa[\ell \mapsto R]}^{\tau, \ell, \bar{\ell}}, M_1, F_1[\text{alwaysEvt } v]) \parallel \\ (\mathfrak{t}_{2\kappa[\ell \mapsto \iota]}^{\tau, \cdot}, M_2, F_2[\text{alwaysEvt unit}]) \parallel \bar{K} \end{array}}$$

The proof for this derivation is the same as the previous case. The only difference is that the right isolate in this case is receiving instead of sending.

$$(i) \text{ SENDRECVISOLATERIGHTISOLATELEFT:}$$

$$\frac{\iota \in \{R, LR\} \quad \iota' \in \{L, RL\} \quad \forall \ell \notin \bar{\ell} \cup \bar{\ell}' \quad \kappa(\ell) = \cdot}{\begin{array}{l} (\mathfrak{t}_{1\kappa[\ell \mapsto R, \ell' \mapsto \iota']}^{\tau, \ell, \bar{\ell}}, M_1, F_1[\text{sendEvt}(c, v)]) \parallel \\ (\mathfrak{t}_{2\kappa[\ell' \mapsto L, \ell \mapsto \iota]}^{\tau, \cdot}, M_2, F_2[\text{recvEvt}(c)]) \parallel \bar{K} \implies \\ (\mathfrak{t}_{1\kappa[\ell \mapsto R, \ell' \mapsto \iota']}^{\tau, \ell, \bar{\ell}}, M_1, F_1[\text{alwaysEvt unit}]) \parallel \\ (\mathfrak{t}_{2\kappa[\ell' \mapsto L, \ell \mapsto \iota]}^{\tau, \ell', \bar{\ell}'}, M_2, F_2[\text{alwaysEvt } v]) \parallel \bar{K} \end{array}}$$

Two separate isolates ℓ and ℓ' are communicating. By the structure of the derivation rule, $\iota = R$ or LR and $\iota' = L$ or LR . For the possible cases for ι and ι' , we prove that the left and right events of both ℓ and ℓ' can be serialized.

- i. $(\iota = R, \iota' = L)$: By the structure of the rules, threads can only communicate to one isolate at a time (not both ℓ and ℓ'). Thus, it follows from the inductive hypothesis that all threads have witnessed the effects of at most one

side of each isolate (ℓ and ℓ') for the n length evaluation sequence. The right side of ℓ has not witnessed the effects of the right side of ℓ' and the left side of ℓ has not witnessed effects of the left side of ℓ' . This derivation maintains this invariant, and thus it follows that the left and right events of both ℓ and ℓ' can be permuted such that their sides are serialized in either order, resulting in an equivalent state with well-defined label states.

- ii. $(\iota = R, \iota' = RL)$: By the structure of the rules, threads can only communicate to one isolate at a time (not both ℓ and ℓ'). Thus, it follows from the inductive hypothesis that all threads have witnessed the effects of at most one side of each isolate (ℓ and ℓ') for the n length evaluation sequence. The right side of ℓ has chosen serialization RL for ℓ' and the left side of ℓ' has not witnessed the effects from the left side of ℓ . Since thread \mathfrak{t}_1 has capability $\ell' \mapsto RL$, it must have been the case that at some point \mathfrak{t}_1 had capability $\ell' \mapsto R$. It follows from the inductive hypothesis that all the right events of isolate ℓ' can be permuted to occur before the capability enrichment to RL and the evaluation of the left event (yielding an equivalent sequence). It also follows from the inductive hypothesis that the evaluation of the left and right events of isolate ℓ can be permuted such that the evaluation of either event precedes the evaluation of the other (also yielding an equivalent sequence). Thus, there exists an equivalent evaluation sequence with well-defined label states.
- iii. $(\iota = LR, \iota' = L)$: By the structure of the rules, threads can only communicate to one isolate at a time (not both ℓ and ℓ'). Thus, it follows from the inductive hypothesis that all threads have witnessed the effects of at most one side of each isolate (ℓ and ℓ') for the n length evaluation sequence. The left side of ℓ' has chosen serialization LR for ℓ and the right side of ℓ has not witnessed the effects from the right side of ℓ' . Since thread \mathfrak{t}_2 has capability $\ell \mapsto LR$, it must have been the case that at some point \mathfrak{t}_2 had capability $\ell \mapsto L$. It follows from the inductive hypothesis that all the left events of isolate ℓ can be permuted to occur before the capability enrichment to LR and the evaluation of the right event (yielding an equivalent sequence). It also follows from the inductive hypothesis that the evaluation of the left and right events of isolate ℓ' can be permuted such that the evaluation of either event precedes the evaluation of the other (also yielding an equivalent sequence). Thus, there exists an equivalent evaluation sequence with well-defined label states.
- iv. $(\iota = LR, \iota' = RL)$: By the structure of the rules, threads can only communicate to one isolate at a time (not both ℓ and ℓ'). Thus, it follows from the inductive hypothesis that all threads have witnessed the effects of at most one side of each isolate (ℓ and ℓ') for the n length evaluation sequence. The left side of ℓ' has chosen serialization LR for ℓ and the right side of ℓ has chosen serialization RL for ℓ' . Since thread \mathfrak{t}_2 has capability $\ell \mapsto LR$, it must have been the case that at some point \mathfrak{t}_2 had capability $\ell \mapsto L$. It follows from the inductive hypothesis that all the left events of isolate ℓ can be permuted to occur before the capability enrichment to LR and the evaluation of the right event (yielding an equivalent sequence). Since thread \mathfrak{t}_1 has capability $\ell' \mapsto RL$, it must have been the case that at some point \mathfrak{t}_1 had capability $\ell' \mapsto R$. It follows

from the inductive hypothesis that all the right events of isolate ℓ' can be permuted to occur before the capability enrichment to RL and the evaluation of the left event (yielding an equivalent sequence). Thus, there exists an equivalent evaluation sequence with well-defined label states.

(j) RECVSENDISOLATERIGHTISOLATELEFT:

$$\frac{\begin{array}{l} \iota \in \{\mathbf{R}, \mathbf{LR}\} \quad \iota' \in \{\mathbf{L}, \mathbf{RL}\} \\ \forall \ell \notin \bar{\ell} \cup \bar{\ell}' \quad \kappa(\ell) = \cdot \end{array}}{\begin{array}{l} (\mathbf{t}_{1\kappa[\ell \mapsto \mathbf{R}, \ell' \mapsto \iota']}^{\tau, \ell, \bar{\ell}} || M_1, F_1[\mathbf{recvEvt}(c)]) || \\ (\mathbf{t}_{2\kappa[\ell' \mapsto \mathbf{L}, \ell \mapsto \iota]}^{\tau, \ell', \bar{\ell}'} || M_2, F_2[\mathbf{sendEvt}(c, v)]) || \bar{K} \implies \\ (\mathbf{t}_{1\kappa[\ell \mapsto \mathbf{R}, \ell' \mapsto \iota']}^{\tau, \ell, \bar{\ell}} || M_1, F_1[\mathbf{alwaysEvt} v]) || \\ (\mathbf{t}_{2\kappa[\ell' \mapsto \mathbf{L}, \ell \mapsto \iota]}^{\tau, \ell', \bar{\ell}'} || M_2, F_2[\mathbf{alwaysEvt} \mathbf{unit}]) || \bar{K} \end{array}}$$

The proof for this derivation is the same as the previous case. The only difference is that the right isolate in this case is receiving and the left is sending, instead of the right sending and the left receiving.

LEMMA 1.6.

If $\bar{K} \implies \bar{K}'$, and \bar{K} and \bar{K}' are well-defined label states, then $T(\bar{K}) \rightsquigarrow T(\bar{K}')$.

The proof is a straightforward case analysis on evaluation derivations $\bar{K} \implies \bar{K}'$, where both \bar{K} and \bar{K}' are well-defined label states.

The proof of Theorem 1.4 is by induction on the length of evaluation sequence $S_1 \implies \dots \implies S_n$. The proof leverages Lemma 1.5 to reason about an evaluation sequence $S'_1 \implies \dots \implies S'_n$, which is equivalent to $S_1 \implies \dots \implies S_n$ and has the property that all intermediate states are well-defined label states. The induction follows directly by Lemma 1.6.