Manifest Deadlock-Freedom for Shared Session Types*

Stephanie Balzer[1], Bernardo Toninho[2], and Frank Pfenning[1]

[1] Carnegie Mellon University, USA
[2] NOVA LINCS, Universidade Nova de Lisboa, Portugal

Abstract. Shared session types generalize the Curry-Howard correspondence between intuitionistic linear logic and the session-typed π-calculus with adjoint modalities that mediate between linear and shared session types, giving rise to a programming model where shared channels must be used according to a lock discipline of acquire-release. While this generalization greatly increases the range of programs that can be written, the gain in expressiveness comes at the cost of deadlock-freedom, a property which holds for many linear session type systems. In this paper, we develop a type system for logically-shared sessions in which types capture not only the interactive behavior of processes but also constrain the order of resources (i.e., shared processes) they may acquire. This type-level information is then used to rule out cyclic dependencies among acquires and synchronization points, resulting in a system that ensures *deadlock-free communication* for well-typed processes in the presence of shared sessions, higher-order channel passing and recursive processes. We illustrate our approach on a series of examples, showing that it rules out deadlocks in circular networks of both shared and linear recursive processes, while still being permissive enough to type concurrent implementations of shared imperative data structures as processes.

Keywords: Linear and Shared Session Types · Deadlock-Freedom

1 Introduction

Session types [22,23,24] naturally describe the interaction protocols that arise amongst concurrent processes that communicate via message-passing. Session types have also been integrated into the Curry-Howard isomorphism through a deep correspondence between *linear logic* and the *session-typed π-calculus* [7,8,46,42]. Session-typed languages based on this logical correspondence [43,46,21] not only guarantee *session fidelity* (i.e. type preservation) but also a strong form of *deadlock-freedom* (i.e. global progress), even in the presence of interleaved sessions, which are often excluded from the deadlock-free fragments of traditional session-typed frameworks [44,23,24,17]. However, these logical session types exclude programming scenarios that demand *sharing* of mutable resources (e.g., shared databases or shared output devices) instead of functional resource replication.

* Supported by NSF Grant No. CCF-1718267: "Enriching Session Types for Practical Concurrent Programming" and NOVA LINCS (Ref. UID/CEC/04516/2013).

To increase their practicality, logical session types have been extended with *manifest sharing* [2]. In the resulting language, linear and shared sessions coexist, but the type system enforces that clients of shared sessions run in mutual exclusion of each other. This separation is achieved by enforcing an *acquire-release* policy, where a client of a shared session must first acquire the session before it can participate in it along a private linear channel. Conversely, when a client releases a session, it gives up its linear channel and only retains a shared reference to the session. Thus, sessions in the presence of manifest sharing can change, or *shift*, between shared and linear execution modes. At the type-level, the acquire-release policy manifests in a stratification of session types into linear and shared with adjoint modalities [4,41,40], connecting the two strata. Operationally, the modality shifting *up* from the linear to the shared layer translates into an *acquire* and the one shifting *down* from shared to linear into a *release*.

Manifest sharing greatly increases the range of programs that can be written because it recovers the expressiveness of the untyped asynchronous π-calculus [3], while maintaining session fidelity. As in the π-calculus, however, the gain in expressiveness comes at the cost of *deadlock-freedom*. An illustrative example is an implementation of the classical dining philosophers problem, shown in Figure 1, using the language $\mathsf{SILL_S}$ [2] that supports manifest sharing (in this setting we often equate a process with the session it offers along a distinguished channel). The code shows the process *fork_proc*, implementing a session of type sfork, and the processes *thinking* and *eating*, implementing sessions of type philosopher. We defer the details of the typing and the definition of the session types sfork and philosopher to Section 2 and focus on the programmatic workings of the processes for now. For ease of reading, we typeset shared session types and variables denoting shared channel references in red.

A *fork_proc* process represents a fork that can be perpetually acquired and released. The actions accept and detach are the duals of acquire and release, respectively, allowing a process to accept an acquire request by a client and to initiate a release by a client, respectively. Process thinking has two shared channel references as arguments, for the fork to the left and right of the philosopher, which the process tries to acquire. If the acquire succeeds, the process recurs as an eating philosopher with two (now) linear channel references of type lfork. Once a philosopher is done eating, it releases both forks and recurs as a thinking philosopher. Let's set a table for three philosopher that share three forks, all

```
fork_proc : {sfork}
c ← fork_proc =
  c' ← accept c ;
  c ← detach c' ;
  c ← fork_proc
```

```
thinking : {phil ← sfork, sfork}
c ← thinking ← left, right =
  left' ← acquire left ;
  right' ← acquire right ;
  c ← eating ← left', right' ;
```

```
eating : {phil ← lfork, lfork}
c ← eating ← left', right' =
  right ← release right' ;
  left ← release left' ;
  c ← thinking ← left, right
```

Fig. 1: Dining Philosophers in $\mathsf{SILL_S}$ [2].

spawned as processes executing in parallel:

f_0:sfork $\leftarrow$ *fork_proc* ; f_1:sfork $\leftarrow$ *fork_proc* ; f_2:sfork $\leftarrow$ *fork_proc* ;
p_0:phil $\leftarrow$ *thinking* $\leftarrow f_0, f_1$; p_1:phil $\leftarrow$ *thinking* $\leftarrow f_1, f_2$;
p_2:phil $\leftarrow$ *thinking* $\leftarrow f_2, f_0$;

Infamously, this configuration may deadlock because of the *circular* dependency between the acquires. We can break this cycle by changing the last line to p_2:phil $\leftarrow$ *thinking* $\leftarrow f_0, f_2$, ensuring forks are acquired in increasing order.

Perhaps unsurprisingly, cyclic dependencies between acquire requests are not the only possible source of deadlocks. Figure 2 gives an example, defining the processes *owner* and *contester*, which both have a shared channel reference to a shared resource that can be perpetually acquired and released. Both processes acquire the shared resource, but additionally also exchange the message ping. More precisely, process *owner* spawns the process *contester*, acquires the shared resource, and only releases the resource after having received the message ping from the *contester*. Process *contester*, on the other hand, first attempts to acquire the resource and then sends the message *ping* to the owner. The program deadlocks if process *owner* acquires the resource first. In that case, it waits for process *contester* to send the message ping while process *contester* waits to acquire the resource held by process *owner*. We note that this deadlock arises in both synchronous and asynchronous semantics.

```
owner : {&{ping : 1} ← sres}
o ← owner ← sr =
  c ← contester ← sr ;
  lr ← acquire sr ;
  case c of
  | ping → wait c ;
           sr ← release lr ; close o
```

```
contester : {1 ← sres}
c ← owner ← sr =
  lr ← acquire sr ;
  o.ping ;
  wait o ;
  close c
```

Fig. 2: Circular dependencies among acquire and synchronization actions.

In this paper, we develop a type system for manifest sharing that rules out cycles among acquire requests and interdependencies between acquire requests and synchronization actions, detecting the two kinds of deadlocks explained above. In our type system, session types not only prescribe *when* resources must be acquired and released, but also the *range* of resources that may be acquired. To this end, we equip the type system with the notion of a *world*, an abstract value at which a process abstractly resides, and type processes relative to an acyclic *ordering* on worlds, akin to the partial-order based approaches of [28,31]. The contributions of this paper are:

– a characterization of the possible forms of deadlocks that can arise in shared session types;

- the introduction of manifest deadlock-freedom, where resource dependencies are manifest in the type structure via world modalities;
- its elaboration in the programming language $\mathsf{SILL}_{\mathsf{S}^+}$, resulting in a type system, a synchronous operational semantics, and proofs of session fidelity (preservation) and a strong form of progress that excludes all deadlocks;
- the novel abstraction of green and red arrows to reason about the interdependencies between processes;
- an illustration of the concepts on various examples, including an extensive comparison with related work.

This paper is structured as follows: Section 2 provides a short introduction to manifest sharing. Section 3 develops the type system and dynamics of the language $\mathsf{SILL}_{\mathsf{S}^+}$. Section 4 illustrates the introduced concepts on an extended example. Section 5 discusses the meta-theoretical properties of $\mathsf{SILL}_{\mathsf{S}^+}$, emphasing progress. Section 6 compares with examples of related work and identifies future work. Section 7 discusses related work, and Section 8 concludes this paper.

2 Manifest Sharing

In the previous section, we have already explored the programmatic workings of *manifest sharing* [2], that enforces an *acquire-release* policy on shared channel references. In this section, we clarify the typing of shared processes.

A key contribution of manifest sharing was to not only support acquire-release as a programming primitive but also to make it *manifest* in the type system. Generalizing the idea of type *stratification* [40,4,41], session types are stratified into a linear and shared layer with two *adjoint modalities* going back and forth between them:

$$\begin{array}{lcl} A_{\mathsf{S}} & \triangleq & \uparrow^{\mathsf{S}}_{\mathsf{L}} A_{\mathsf{L}} \\ A_{\mathsf{L}}, B_{\mathsf{L}} & \triangleq & A_{\mathsf{L}} \otimes B_{\mathsf{L}} \mid \oplus\{\overline{l : A_{\mathsf{L}}}\} \mid \&\{\overline{l : A_{\mathsf{L}}}\} \mid A_{\mathsf{L}} \multimap B_{\mathsf{L}} \mid \exists x{:}A_{\mathsf{S}}.B_{\mathsf{L}} \mid \Pi x{:}A_{\mathsf{S}}.B_{\mathsf{L}} \mid \mathbf{1} \mid \downarrow^{\mathsf{S}}_{\mathsf{L}} A_{\mathsf{S}} \end{array}$$

In the linear layer, we get the standard connectives of intuitionistic linear logic ($A_{\mathsf{L}} \otimes B_{\mathsf{L}}$, $A_{\mathsf{L}} \multimap B_{\mathsf{L}}$, $\oplus\{\overline{l : A_{\mathsf{L}}}\}$, $\&\{\overline{l : A_{\mathsf{L}}}\}$, and $\mathbf{1}$). These connectives are extended with the modal operator $\downarrow^{\mathsf{S}}_{\mathsf{L}} A_{\mathsf{S}}$ shifting *down* from the shared to the linear layer. Similarly, in the shared layer we have the operator $\uparrow^{\mathsf{S}}_{\mathsf{L}} A_{\mathsf{L}}$ shifting *up* from the linear to the shared layer. The former translates into a *release* (and, dually, detach), the latter into an *acquire* (and, dually, accept). As a result, we obtain a system in which session types prescribe all forms of communication, including the acquisition and release of shared processes.

Table 1 provides an overview of $\mathsf{SILL}_{\mathsf{S}}$'s session types and their operational reading. Since $\mathsf{SILL}_{\mathsf{S}}$ is based on an intuitionistic interpretation of linear logic session types [7], types are expressed from the point of view of the *providing process*, with the channel along which the process provides the session behavior being characterized by its session type. This choice avoids the explicit duality operation present in original presentations of session types [22,23] and in those based on classical linear logic [46]. Table 1 lists the points of view of the *provider*

Table 1: Session types in $\mathsf{SILL_S}$ and their operational meaning.

Session type		**Process term**		
current	**cont**	**current**	**cont**	**Description**
$c_\mathsf{L} : \oplus\{\overline{l : A_\mathsf{L}}\}$	$c_\mathsf{L} : A_{\mathsf{L}_h}$	$c_\mathsf{L}.l_h \,;P$	P	sends label l_h along c_L
		$\mathsf{case}\ c_\mathsf{L}\ \mathsf{of}\ \overline{l \Rightarrow Q}$	Q_h	receives label l_h along c_L
$c_\mathsf{L} : \&\{\overline{l : A_\mathsf{L}}\}$	$c_\mathsf{L} : A_{\mathsf{L}_h}$	$\mathsf{case}\ c_\mathsf{L}\ \mathsf{of}\ \overline{l \Rightarrow P}$	P_h	receives label l_h along c
		$c_\mathsf{L}.l_h \,;Q$	Q	sends label l_h along c_L
$c_\mathsf{L} : A_\mathsf{L} \otimes B_\mathsf{L}$	$c_\mathsf{L} : B_\mathsf{L}$	$\mathsf{send}\ c_\mathsf{L}\ d_\mathsf{L} \,;P$	P	sends channel $d_\mathsf{L} : A_\mathsf{L}$ along c_L
		$y_\mathsf{L} \leftarrow \mathsf{recv}\ c_\mathsf{L} \,;Q_{y_\mathsf{L}}$	$[d_\mathsf{L}/y_\mathsf{L}]\,Q_{y_\mathsf{L}}$	receives channel $d_\mathsf{L} : A_\mathsf{L}$ along c_L
$c_\mathsf{L} : A_\mathsf{L} \multimap B_\mathsf{L}$	$c_\mathsf{L} : B_\mathsf{L}$	$y_\mathsf{L} \leftarrow \mathsf{recv}\ c_\mathsf{L} \,;P_{y_\mathsf{L}}$	$[d_\mathsf{L}/y_\mathsf{L}]\,P_{y_\mathsf{L}}$	receives channel $d_\mathsf{L} : A_\mathsf{L}$ along c_L
		$\mathsf{send}\ c_\mathsf{L}\ d_\mathsf{L} \,;Q$	Q	sends channel $d_\mathsf{L} : A_\mathsf{L}$ along c_L
$c_\mathsf{L} : \Pi x{:}A_\mathsf{S}.B_\mathsf{L}$	$c_\mathsf{L} : B_\mathsf{L}$	$\mathsf{send}\ c_\mathsf{L}\ d_\mathsf{S} \,;P$	P	sends channel $d_\mathsf{S} : A_\mathsf{S}$ along c_L
		$y_\mathsf{S} \leftarrow \mathsf{recv}\ c_\mathsf{L} \,;Q_{y_\mathsf{S}}$	$[d_\mathsf{S}/y_\mathsf{S}]\,Q_{y_\mathsf{S}}$	receives channel $d_\mathsf{S} : A_\mathsf{S}$ along c_L
$c_\mathsf{L} : \exists x{:}A_\mathsf{S}.B_\mathsf{L}$	$c_\mathsf{L} : B_\mathsf{L}$	$y_\mathsf{S} \leftarrow \mathsf{recv}\ c_\mathsf{L} \,;P_{y_\mathsf{S}}$	$[d_\mathsf{S}/y_\mathsf{S}]\,P_{y_\mathsf{S}}$	receives channel $d_\mathsf{S} : A_\mathsf{S}$ along c_L
		$\mathsf{send}\ c_\mathsf{L}\ d_\mathsf{S} \,;Q$	Q	sends channel $d_\mathsf{S} : A_\mathsf{S}$ along c_L
$c_\mathsf{L} : \mathbf{1}$	-	$\mathsf{close}\ c_\mathsf{L}$	-	sends "end" along c_L
		$\mathsf{wait}\ c_\mathsf{L} \,;Q$	Q	receives "end" along c_L
$c_\mathsf{L} : \downarrow^\mathsf{S}_\mathsf{L} A_\mathsf{S}$	$c_\mathsf{S} : A_\mathsf{S}$	$c_\mathsf{S} \leftarrow \mathsf{detach}\ c_\mathsf{L} \,;P_{x_\mathsf{S}}$	$[c_\mathsf{S}/x_\mathsf{S}]\,P_{x_\mathsf{S}}$	sends "$\mathsf{detach}\ c_\mathsf{S}$" along c_L
		$x_\mathsf{S} \leftarrow \mathsf{release}\ c_\mathsf{L} \,;Q_{x_\mathsf{S}}$	$[c_\mathsf{S}/x_\mathsf{S}]\,Q_{x_\mathsf{S}}$	receives "$\mathsf{detach}\ c_\mathsf{S}$" along c_L
$c_\mathsf{S} : \uparrow^\mathsf{S}_\mathsf{L} A_\mathsf{L}$	$c_\mathsf{L} : A_\mathsf{L}$	$c_\mathsf{L} \leftarrow \mathsf{acquire}\ c_\mathsf{S} \,;Q_{x_\mathsf{L}}$	$[c_\mathsf{L}/x_\mathsf{L}]\,Q_{x_\mathsf{L}}$	sends "$\mathsf{acquire}\ c_\mathsf{L}$" along c_S
		$x_\mathsf{L} \leftarrow \mathsf{accept}\ c_\mathsf{S} \,;P_{x_\mathsf{L}}$	$[c_\mathsf{L}/x_\mathsf{L}]\,P_{x_\mathsf{L}}$	receives "$\mathsf{acquire}\ c_\mathsf{L}$" along c_S

and *client* of a given connective in the first and second lines, respectively. Moreover, Table 1 gives for each connective its session type before (**Session type current**) and after the message exchange (**Session type cont(inuation)**), along with their respective process terms (**Process term current** and **Process term continuation**, respectively). We can see that the process terms of a provider and a client for a given connective come in matching pairs, indicating that the participants' views of the session change consistently.

We are now able to give the session types of the processes *fork_proc*, *thinking*, and *eating* defined in the previous section: $\mathsf{lfork} = \downarrow^\mathsf{S}_\mathsf{L} \mathsf{sfork}$, $\mathsf{sfork} = \uparrow^\mathsf{S}_\mathsf{L} \mathsf{lfork}$ and $\mathsf{phil} = \mathbf{1}$. We highlight the mutually recursive session types lfork and sfork, representing a fork that can perpetually be acquired and released. We adopt an *equi-recursive* [12] interpretation for recursive session types, silently equating a recursive type with its unfolding and requiring types to be *contractive* [16].

We briefly discuss the typing and the dynamics of acquire-release. The typing and the dynamics of the residual linear connectives are standard, and we detail them in the context of $\mathsf{SILL_{S^+}}$ (see Section 3). As is usual for an intuitionistic interpretation, each connective gives rise to a left and a right rule, denoting the use and provision, respectively, of a session of the given type:

$$\frac{\Gamma;\cdot \vdash P_{x_\mathsf{L}} :: (x_\mathsf{L} : A_\mathsf{L})}{\Gamma \vdash x_\mathsf{L} \leftarrow \mathsf{accept}\, x_\mathsf{S}; P_{x_\mathsf{L}} :: (x_\mathsf{S} : \uparrow^\mathsf{S}_\mathsf{L} A_\mathsf{L})}\ (\text{T-}\uparrow^\mathsf{S}_\mathsf{LR}) \qquad \frac{\Gamma, x_\mathsf{S} : \uparrow^\mathsf{S}_\mathsf{L} A_\mathsf{L}; \Delta, x_\mathsf{L} : A_\mathsf{L} \vdash Q_{x_\mathsf{L}} :: (z_\mathsf{L} : C_\mathsf{L})}{\Gamma, x_\mathsf{S} : \uparrow^\mathsf{S}_\mathsf{L} A_\mathsf{L}; \Delta \vdash x_\mathsf{L} \leftarrow \mathsf{acquire}\, x_\mathsf{S}; Q_{x_\mathsf{L}} :: (z_\mathsf{L} : C_\mathsf{L})}\ (\text{T-}\uparrow^\mathsf{S}_\mathsf{LL})$$

$$\frac{\Gamma \vdash P_{x_\mathsf{S}} :: (x_\mathsf{S} : A_\mathsf{S})}{\Gamma;\cdot \vdash x_\mathsf{S} \leftarrow \mathsf{detach}\, x_\mathsf{L}; P_{x_\mathsf{S}} :: (x_\mathsf{L} : \downarrow^\mathsf{S}_\mathsf{L} A_\mathsf{S})}\ (\text{T-}\downarrow^\mathsf{S}_\mathsf{LR}) \qquad \frac{\Gamma, x_\mathsf{S} : A_\mathsf{S}; \Delta \vdash Q_{x_\mathsf{S}} :: (z_\mathsf{L} : C_\mathsf{L})}{\Gamma; \Delta, x_\mathsf{L} : \downarrow^\mathsf{S}_\mathsf{L} A_\mathsf{S} \vdash x_\mathsf{S} \leftarrow \mathsf{release}\, x_\mathsf{L}; Q_{x_\mathsf{S}} :: (z_\mathsf{L} : C_\mathsf{L})}\ (\text{T-}\downarrow^\mathsf{S}_\mathsf{LL})$$

The typing judgments $\Gamma \vdash P :: (x_\mathsf{S} : A_\mathsf{S})$ and $\Gamma; \Delta \vdash P :: (x_\mathsf{L} : A_\mathsf{L})$ indicate that process P provides a session of type A along channel x, given the typing of the channels specified in typing contexts Γ (and Δ). Γ and Δ consist of hypotheses on the typing of shared and linear channels, respectively, where Γ is a structural and Δ a linear context. To allow for recursive process definitions, the typing judgment depends on a signature Σ that is populated with all process definitions prior to type-checking. The adjoint formulation precludes shared processes from depending on linear channel references [2,40], a restriction motivated from logic referred to as the independence principle [40]. Thus, when a shared session accepts an acquire and shifts to linear, it starts with an empty linear context.

Operationally, the dynamics of SILL_S is captured by *multiset rewriting rules* [10], which denote computation in terms of state transitions between configurations of processes. Multiset rewriting rules are local in that they only mention the parts of a configuration they rewrite. For acquire-release we have the following:

$$(\text{D-}\uparrow^\mathsf{S}_\mathsf{L})\quad \mathsf{proc}(a_\mathsf{S}, x_\mathsf{L} \leftarrow \mathsf{accept}\, a_\mathsf{S}\, ; P_{x_\mathsf{L}}),\ \mathsf{proc}(c_\mathsf{L}, x_\mathsf{L} \leftarrow \mathsf{acquire}\, a_\mathsf{S}\, ; Q_{x_\mathsf{L}}) \longrightarrow \mathsf{proc}(a_\mathsf{L}, [a_\mathsf{L}/x_\mathsf{L}]\, P_{x_\mathsf{L}}),\ \mathsf{proc}(c_\mathsf{L}, [a_\mathsf{L}/x_\mathsf{L}]\, Q_{x_\mathsf{L}}),\ \mathsf{unavail}(a_\mathsf{S})$$

$$(\text{D-}\downarrow^\mathsf{S}_\mathsf{L})\quad \mathsf{proc}(a_\mathsf{L}, x_\mathsf{S} \leftarrow \mathsf{detach}\, a_\mathsf{L}\, ; P_{x_\mathsf{S}}),\ \mathsf{proc}(c_\mathsf{L}, x_\mathsf{S} \leftarrow \mathsf{release}\, a_\mathsf{L}\, ; Q_{x_\mathsf{S}}),\ \mathsf{unavail}(a_\mathsf{S}) \longrightarrow \mathsf{proc}(a_\mathsf{S}, [a_\mathsf{S}/x_\mathsf{S}]\, P_{x_\mathsf{S}}),\ \mathsf{proc}(c_\mathsf{L}, [a_\mathsf{S}/x_\mathsf{S}]\, Q_{x_\mathsf{S}})$$

Configuration states are defined by the predicates $\mathsf{proc}(c_m, P)$ and $\mathsf{unavail}(a_\mathsf{S})$. The former denotes a running process with process term P providing along channel c_m, the latter acts as a placeholder for a shared process providing along channel a_S that is currently not available. The above rule exploits the invariant that a process' providing channel a can appear at one of two modes, a linear one, a_L, and a shared one, a_S. While the process (i.e. the session) is linear, it provides along a_L, while it is shared, along a_S. When a process shifts between modes, it switches between the two modes of its offering channel. The channel at the appropriate mode is substituted for the variables occurring in process terms.

3 Manifest Deadlock-Freedom

In this section, we introduce our language $\mathsf{SILL}_{\mathsf{S}^+}$, a session-typed language that supports sharing without deadlock. We focus on $\mathsf{SILL}_{\mathsf{S}^+}$'s type system and dynamics in this section and discuss its meta-theoretical properties in Section 5.

3.1 Competition and Collaboration

The introduction of acquire-release, to ensure that the multiple clients of a shared process interact with the process in mutual exclusion from each other, gives rise to an obvious source of deadlocks, as acquire-release effectively amounts to a locking discipline. The typical approach to prevent deadlocks in that case is to impose a partial order on the resources and to *"lock-up"*, i.e., to lock the resources in ascending order. We adopted this strategy in Section 1 (Figure 1) to break the cyclic dependencies among the acquires in the dining philosophers.

In Section 1, however, we also considered another example (Figure 2) and discovered that *cyclic acquisitions* are not the only source of deadlocks, but deadlocks can also arises from *interdependent acquisitions and synchronizations*. In that example, we can prevent the deadlock by moving the acquire past the synchronization, in either of the two processes. Whereas in a purely linear session-typed system the synchronization points of message exchange denote the only interdependence between processes, the relative sequencing of actions in between two synchronization points can now become relevant in a shared session-typed system. In particular, an acquire executed by one process may affect all the other processes that may attempt to acquire the same resource.

Based on this observation, we can divide the processes in a shared-session discipline into *competitors* and *collaborators*. The former compete for a set of resources, whereas the latter do not overlap in the set of resources they acquire. For example, in dining philosophers (Figure 1), the philosophers p_0, p_1, and p_2 compete with each other on the set of forks f_0, f_1, and f_2, whereas the process that spawns the philosophers and the forks collaborates with either of them.

Transferring this idea to the process graph that emerges at run-time, we note that competitors are siblings whereas collaborators stand in a parent-descendant relationship. We illustrate this outcome on Figure 3 that shows a possible run-time process graph for the dining philosophers. Linear processes are depicted as solid black circles with a white identifier and shared processes are depicted as dotted filled violet circles with a black identifier. Linear channels are depicted as black lines, shared channel references as dotted violet lines with the arrow head pointing to the shared process being acquired[3]. The identifiers P_0, P_1, and P_2 stand for the three philosophers, F_0, F_1, and F_2 for the three forks, and T for the process that sets the table. The current run-time graph depicts the scenario in which P_1 is eating, while the other two philosophers are still thinking.

Embedded in the graph is a *tree* that arises from the linear processes and the linear channels connecting them. For any two nodes in this tree, the *parent* node denotes the *client* process and the *child* node the *providing* process. We note that the *independence principle* (see Section 2), which precludes shared processes from depending on linear channel references, guarantees that there exists exactly one tree in the process graph, with the linear main process as its root. The shape of the tree changes when new processes are spawned, when

[3] We have made sure to make the different concepts sufficiently distinguishable even in greyscale mode.

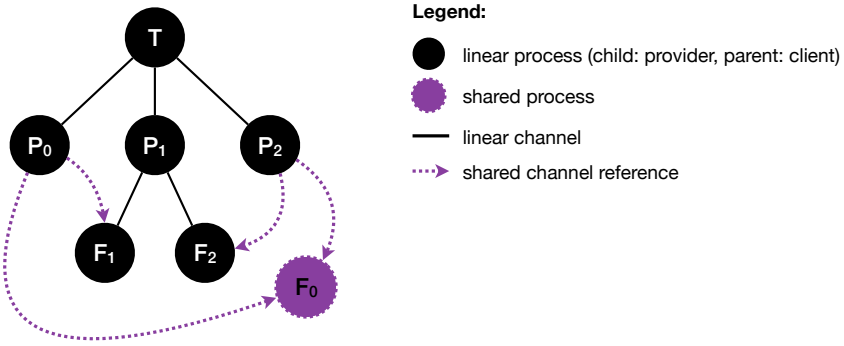


Fig. 3: Run-time process graph for dining philosopher (see Figure 1).

linear channels are exchanged (through $\otimes$ and $\multimap$), or when shared processes are acquired. For example, process P_2 could acquire the shared fork F_0, which then becomes a linear child process of P_2 should the acquire succeed. As indicated by the shared channel references, the sibling nodes P_0, P_1, and P_2 compete with each other for the nodes F_0, F_1, and F_2, whereas the node T does not compete for any of the resources acquired by its *descendants* (including F_1 and F_2). Our type system enforces this paradigm, as we discuss in the next section.

3.2 Type System

Invariants Having identified the notions of *collaborators* and *competitors*, our type system must guarantee: *(i)* that collaborators acquire mutually disjoint sets of resources; *(ii)* that competitors employ a locking-up strategy for the resources they share; and, *(iii)* that competitors have released all acquired resources when synchronizing with other competitors. Invariant *(ii)* rules out cyclic acquisitions and invariants *(i)* and *(iii)* combined rule out interdependent acquisitions and synchronizations.

To express the high-level invariants above in our type system, we introduce the notion of a *world* – an abstract value that is equipped with a partial order – and associate such a world with every process. Programmers can *create* worlds, indicate the world at which a process resides at spawn time, and define an *order* on worlds. Moreover, we associate with each process a *range of worlds* that indicates the worlds of resources that the process may acquire. As a result, we obtain the following typing judgments:

$$\Psi;\, \Gamma \vdash P :: (x_{\mathsf{s}} : A_{\mathsf{s}}[\omega_k \updownarrow_{\omega_l}^{\omega_n}]) \quad (\text{where } \Psi^+ \text{ irreflexive})$$

$$\Psi;\, \Gamma;\, \Phi;\, \Delta \vdash P :: (x_{\mathsf{L}} : A_{\mathsf{L}}[\omega_k \updownarrow_{\omega_l}^{\omega_n}]) \quad (\text{where } \Psi^+ \text{ irreflexive})$$

The typing judgments reveal that we impose worlds at the *judgmental level*, resulting in a *hybrid system*, in which the adjoint modalities for acquire-release

are complemented with world modalities that occur as *syntactic objects* in propositions [6]. We use the notation $x_m : A_m[\omega_k \updownarrow_{\omega_l}^{\omega_n}]$ (where m stands for S or L) to associate worlds ω_k, ω_l, and ω_n with a process that offers a session of type A_m along channel x. World ω_k denotes the world at which the process resides. We refer to this world as the *self* world. Worlds ω_l and ω_n indicate the range of worlds of resources that the process may acquire, with ω_l denoting the *minimal (min)* world in this range and ω_n the *maximal (max)* one.

Process terms are typed relative to the order specified in Ψ and the contexts Γ, Φ, and Δ. As in Section 2, Γ is a structural context consisting of hypotheses on the typing of variables bound to shared channel references, augmented with world annotations. We find it necessary, however, to split the linear context "Δ" from Section 2 into the two disjoint contexts Φ and Δ, allowing us to separate processes that are possibly aliased (due to sharing) from those that are not, respectively. Both Φ and Δ consist of hypotheses on the typing of variables that are bound to linear channels, augmented with world annotations. Ψ is presupposed to be *acyclic* and defined as: $\Psi \triangleq \cdot \mid \Psi',\ \omega_k < \omega_l \mid \Psi',\ \omega_o$, where ω stands for a concrete *world* w or a *world variable* δ. We allow Ψ to contain single worlds, in case Ψ is a singleton, as well as to accommodate world creation prior to order declarations. We define the transitive closure Ψ^+, yielding a *strict partial order*, and the reflexive transitive closure Ψ^*, yielding a *partial order*.

The high-level invariants *(i)*, *(ii)*, and *(iii)* identified earlier naturally transcribe into the following invariants, which we impose on the typing judgments above. We use the notation $\_\langle x_m \rangle; P$ to denote a process term that currently executes an action along channel x_m.

1. $\mathsf{min}(\mathsf{parent}) \leq \mathsf{self}(\mathsf{acquired\_child}) \leq \mathsf{max}(\mathsf{parent})$:
   $\forall y_{\mathsf{L}} : B_{\mathsf{L}}[\omega_o \updownarrow_{\omega_p}^{\omega_r}] \in \Phi : \Psi^* \vdash \omega_l \leq \omega_o \leq \omega_n$
2. $\mathsf{max}(\mathsf{parent}) < \mathsf{min}(\mathsf{child})$:
   $\forall y_{\mathsf{L}} : B_{\mathsf{L}}[\omega_o \updownarrow_{\omega_p}^{\omega_r}] \in \Delta \cup \Phi : \Psi^+ \vdash \omega_n < \omega_p$
3. If $\Psi;\ \Gamma, x_{\mathsf{S}} : \hat{A}[\omega_t \updownarrow_{\omega_u}^{\omega_v}];\ \Phi;\ \Delta \vdash x_{\mathsf{L}} \leftarrow \mathsf{acquire}\ x_{\mathsf{S}};\ Q_{x_{\mathsf{S}}} :: (z_{\mathsf{L}} : C_{\mathsf{L}}[\omega_k \updownarrow_{\omega_l}^{\omega_n}])$, then
   $\forall y_{\mathsf{L}} : B_{\mathsf{L}}[\omega_o \updownarrow_{\omega_p}^{\omega_r}] \in \Phi : \Psi^+ \vdash \omega_o < \omega_t$.
4. If $\Psi;\ \Gamma;\ \Phi;\ \Delta \vdash \_\langle x_m \rangle; P :: (x_{\mathsf{L}} : A_{\mathsf{L}}[\omega_k \updownarrow_{\omega_l}^{\omega_n}])$, then $\Phi = (\cdot)$.

Invariants 1 and 2 ensure that, for any node in the tree, the acquired resources reside at smaller worlds than those acquired by any descendant. As a result, the two invariants guarantee high-level invariant *(i)*. Invariant 3, on the other hand, imposes a lock-up strategy on acquires and guarantees the high-level invariant *(ii)*. To guarantee high-level invariant *(iii)*, we impose Invariant 4, which forces a process to release any acquired resources before communicating along its offering channel. Since sibling nodes cannot be directly connected by a linear channel, the only way for them to synchronize is through a common parent. Finally, to guarantee that world annotations are internally consistent, we require for each annotation that the minimal world is less than the maximal world.

Rules We now present select process typing rules (a complete listing is provided in Appendix B for ease of reference). The only new rules with respect to the

language SILL_S [2] are those pertaining to world creation and order determination. These are extra-logical judgmental rules. We allow both linear and shared processes to create and relate worlds. Rules (T-New$_\mathsf{L}$) and (T-New$_\mathsf{S}$) create a new world w and make it available to the continuation Q_w. Rules (T-Ord$_\mathsf{L}$) and (T-Ord$_\mathsf{S}$) relate two existing worlds, while preserving acyclicity of the order.

$$\frac{\Psi, \mathsf{w};\ \Gamma;\ \Phi;\ \Delta \vdash Q_\mathsf{w} :: (x_\mathsf{L} : A_\mathsf{L}[\omega_m \updownarrow^{\omega_v}_{\omega_u}])}{\Psi;\ \Gamma;\ \Phi;\ \Delta \vdash \mathsf{w} \leftarrow \mathsf{new\_world};\ Q_\mathsf{w} :: (x_\mathsf{L} : A_\mathsf{L}[\omega_m \updownarrow^{\omega_v}_{\omega_u}])}\ (\text{T-New}_\mathsf{L})$$

$$\frac{\Psi, \mathsf{w};\ \Gamma \vdash Q_\mathsf{w} :: (x_\mathsf{S} : A_\mathsf{S}[\omega_m \updownarrow^{\omega_v}_{\omega_u}])}{\Psi;\ \Gamma \vdash \mathsf{w} \leftarrow \mathsf{new\_world};\ Q_\mathsf{w} :: (x_\mathsf{S} : A_\mathsf{S}[\omega_m \updownarrow^{\omega_v}_{\omega_u}])}\ (\text{T-New}_\mathsf{S})$$

$$\frac{\begin{array}{c}\omega_p, \omega_r \in \Psi \qquad (\Psi, \omega_p < \omega_r)^+ \text{ irreflexive} \\ \Psi, \omega_p < \omega_r;\ \Gamma;\ \Phi;\ \Delta \vdash Q :: (x_\mathsf{L} : A_\mathsf{L}[\omega_m \updownarrow^{\omega_v}_{\omega_u}])\end{array}}{\Psi;\ \Gamma;\ \Phi;\ \Delta \vdash \omega_p < \omega_r;\ Q :: (x_\mathsf{L} : A_\mathsf{L}[\omega_m \updownarrow^{\omega_v}_{\omega_u}])}\ (\text{T-Ord}_\mathsf{L})$$

$$\frac{\begin{array}{c}\omega_p, \omega_r \in \Psi \qquad (\Psi, \omega_p < \omega_r)^+ \text{ irreflexive} \\ \Psi, \omega_p < \omega_r;\ \Gamma \vdash Q :: (x_\mathsf{S} : A_\mathsf{S}[\omega_m \updownarrow^{\omega_v}_{\omega_u}])\end{array}}{\Psi;\ \Gamma \vdash \omega_p < \omega_r;\ Q :: (x_\mathsf{S} : A_\mathsf{S}[\omega_m \updownarrow^{\omega_v}_{\omega_u}])}\ (\text{T-Ord}_\mathsf{S})$$

We now consider the typing rule for acquire, which must explicitly enforce the various low-level invariants above. Since an acquire results in the addition of a new child node to the executing process, the rule can interfere with invariants 1 and 2. The first two premises of the rule ensure that the two invariants are preserved. Moreover, the rule has to ensure that the acquiring process is locking-up, which is achieved by the third premise.

$$\frac{\begin{array}{c}\Psi^* \vdash \omega_k \leq \omega_m \leq \omega_n \qquad \Psi^+ \vdash \omega_n < \omega_u \qquad \forall y_\mathsf{L} : B_\mathsf{L}[\omega_l \updownarrow^{\omega_r}_{\omega_p}] \in \Phi : \omega_l < \omega_m \\ \Psi;\ \Gamma, x_\mathsf{S} : \uparrow^\mathsf{S}_\mathsf{L} A_\mathsf{L}[\omega_m \updownarrow^{\omega_v}_{\omega_u}];\ \Phi, x_\mathsf{L} : A_\mathsf{L}[\omega_m \updownarrow^{\omega_v}_{\omega_u}];\ \Delta \vdash Q_{x_\mathsf{L}} :: (z_\mathsf{L} : C_\mathsf{L}[\omega_j \updownarrow^{\omega_n}_{\omega_k}])\end{array}}{\Psi;\ \Gamma, x_\mathsf{S} : \uparrow^\mathsf{S}_\mathsf{L} A_\mathsf{L}[\omega_m \updownarrow^{\omega_v}_{\omega_u}];\ \Phi;\ \Delta \vdash x_\mathsf{L} \leftarrow \mathsf{acquire}\, x_\mathsf{S}\, ;\, Q_{x_\mathsf{L}} :: (z_\mathsf{L} : C_\mathsf{L}[\omega_j \updownarrow^{\omega_n}_{\omega_k}])}\ (\text{T-}\uparrow^\mathsf{S}_\mathsf{L}\text{L})$$

The remaining shift rules are actually *unchanged* with respect to SILL_S, modulo the world annotations. In particular, low-level invariant 4 is already satisfied because the conclusion of rule (T-$\uparrow^\mathsf{S}_\mathsf{L}$R) does not have a context Φ and because the independence principle forces Φ to be empty in rule (T-$\downarrow^\mathsf{S}_\mathsf{L}$R).

$$\frac{\Psi;\ \Gamma;\ \cdot;\ \cdot \vdash P_{x_\mathsf{L}} :: (x_\mathsf{L} : A_\mathsf{L}[\omega_m \updownarrow^{\omega_v}_{\omega_u}])}{\Psi;\ \Gamma \vdash x_\mathsf{L} \leftarrow \mathsf{accept}\, x_\mathsf{S}\, ;\, P_{x_\mathsf{L}} :: (x_\mathsf{S} : \uparrow^\mathsf{S}_\mathsf{L} A_\mathsf{L}[\omega_m \updownarrow^{\omega_v}_{\omega_u}])}\ (\text{T-}\uparrow^\mathsf{S}_\mathsf{L}\text{R})$$

$$\frac{\Psi;\ \Gamma, x_\mathsf{S} : A_\mathsf{S}[\omega_m \updownarrow^{\omega_v}_{\omega_u}];\ \Phi;\ \Delta \vdash Q_{x_\mathsf{S}} :: (z_\mathsf{L} : C_\mathsf{L}[\omega_j \updownarrow^{\omega_n}_{\omega_k}])}{\Psi;\ \Gamma;\ \Phi, x_\mathsf{L} : \downarrow^\mathsf{S}_\mathsf{L} A_\mathsf{S}[\omega_m \updownarrow^{\omega_v}_{\omega_u}];\ \Delta \vdash x_\mathsf{S} \leftarrow \mathsf{release}\, x_\mathsf{L}\, ;\, Q_{x_\mathsf{S}} :: (z_\mathsf{L} : C_\mathsf{L}[\omega_j \updownarrow^{\omega_n}_{\omega_k}])}\ (\text{T-}\downarrow^\mathsf{S}_\mathsf{L}\text{L})$$

$$\frac{\Psi;\ \Gamma \vdash P_{x_\mathsf{S}} :: (x_\mathsf{S} : A_\mathsf{S}[\omega_m \updownarrow^{\omega_v}_{\omega_u}])}{\Psi;\ \Gamma;\ \cdot;\ \cdot \vdash x_\mathsf{S} \leftarrow \mathsf{detach}\, x_\mathsf{L}\, ;\, P_{x_\mathsf{S}} :: (x_\mathsf{L} : \downarrow^\mathsf{S}_\mathsf{L} A_\mathsf{S}[\omega_m \updownarrow^{\omega_v}_{\omega_u}])}\ (\text{T-}\downarrow^\mathsf{S}_\mathsf{L}\text{R})$$

We now consider the linear connectives, starting with $\mathbf{1}$. Rule (T-$\mathbf{1}_\mathrm{L}$) reveals that only processes that have never been acquired may be terminated. This restriction is important to guarantee progress because existing clients of a shared process may wait indefinitely otherwise. We impose this restriction as a well-formedness condition on a session type, giving rise to a *strictly equi-synchronizing* session type. The notion of an *equi-synchronizing* session type [2]

has been defined for SILL_S and guarantees that a process that has been acquired at a type A_s must be released back to the type A_s. A strictly equi-synchronizing session type additionally requires that an acquired resource must be released. The corresponding rules are given in Appendix B.4, Figure 13. Linearity enforces invariant 4, making sure that no linear channels remain in rule (T-$\mathbf{1}_\mathrm{R}$).

$$\frac{\Psi;\ \Gamma;\ \Phi;\ \Delta \vdash Q :: (z_\mathsf{L} : C_\mathsf{L}[\omega_j \updownarrow_{\omega_k}^{\omega_n}])}{\Psi;\ \Gamma;\ \Phi;\ \Delta, x_\mathsf{L} : \mathbf{1}[\omega_m \updownarrow_{\omega_u}^{\omega_v}] \vdash \mathsf{wait}\, x_\mathsf{L}\,;Q :: (z_\mathsf{L} : C_\mathsf{L}[\omega_j \updownarrow_{\omega_k}^{\omega_n}])}\ (\text{T-}\mathbf{1}_\mathrm{L})$$

$$\frac{}{\Psi;\ \Gamma;\ \cdot\,;\ \cdot \vdash \mathsf{close}\, x_\mathsf{L} :: (x_\mathsf{L} : \mathbf{1}[\omega_m \updownarrow_{\omega_u}^{\omega_v}])}\ (\text{T-}\mathbf{1}_\mathrm{R})$$

Next, we consider internal and external choice. Since internal and external choice cannot alter the linear process tree of a process graph, the rules are very similar to the ones in SILL_S. The only differences are that we get two left rules for each connective and that the Φ-context of each right rule must be empty to satisfy invariant 4. The former is merely due to the tracking of possibly aliased sessions in the Φ context (we list only rules for internal choice, those for external choice are dual and can be found in Appendix B.2).

$$\frac{(\forall i)\ \ \Psi;\ \Gamma;\ \Phi;\ \Delta, x_\mathsf{L} : A_{\mathsf{L}_i}[\omega_m \updownarrow_{\omega_u}^{\omega_v}] \vdash Q_i :: (z_\mathsf{L} : C_\mathsf{L}[\omega_j \updownarrow_{\omega_k}^{\omega_n}])}{\Psi;\ \Gamma;\ \Phi;\ \Delta, x_\mathsf{L} : \oplus\{\overline{l : A_\mathsf{L}}\}[\omega_m \updownarrow_{\omega_u}^{\omega_v}] \vdash \mathsf{case}\ x_\mathsf{L}\ \mathsf{of}\ \overline{l \Rightarrow Q} :: (z_\mathsf{L} : C_\mathsf{L}[\omega_j \updownarrow_{\omega_k}^{\omega_n}])}\ (\text{T-}\oplus_{\mathrm{L}_1})$$

$$\frac{(\forall i)\ \ \Psi;\ \Gamma;\ \Phi, x_\mathsf{L} : A_{\mathsf{L}_i}[\omega_m \updownarrow_{\omega_u}^{\omega_v}];\ \Delta \vdash Q_i :: (z_\mathsf{L} : C_\mathsf{L}[\omega_j \updownarrow_{\omega_k}^{\omega_n}])}{\Psi;\ \Gamma;\ \Phi, x_\mathsf{L} : \oplus\{\overline{l : A_\mathsf{L}}\}[\omega_m \updownarrow_{\omega_u}^{\omega_v}];\ \Delta \vdash \mathsf{case}\ x_\mathsf{L}\ \mathsf{of}\ \overline{l \Rightarrow Q} :: (z_\mathsf{L} : C_\mathsf{L}[\omega_j \updownarrow_{\omega_k}^{\omega_n}])}\ (\text{T-}\oplus_{\mathrm{L}_2})$$

$$\frac{\Psi;\ \Gamma;\ \cdot\,;\ \Delta \vdash P :: (x_\mathsf{L} : A_{\mathsf{L}\,h}[\omega_m \updownarrow_{\omega_u}^{\omega_v}])}{\Psi;\ \Gamma;\ \cdot\,;\ \Delta \vdash x_\mathsf{L}.l_h\,;P :: (x_\mathsf{L} : \oplus\{\overline{l : A_\mathsf{L}}\}[\omega_m \updownarrow_{\omega_u}^{\omega_v}])}\ (\text{T-}\oplus_\mathrm{R})$$

More interesting are linear channel output and input, since these alter the linear process tree of a process graph. Moreover, additional world annotations are needed to indicate the worlds of the channel that is exchanged. For the latter we use the notation $@\omega_l \updownarrow_{\omega_p}^{\omega_r}$, indicating that the exchanged channel has the worlds ω_l, ω_p, and ω_r for self, min, and max, respectively. To account for induced changes in the process graph, the rules that type an input of a linear channel must guard against any disturbance of invariants 1 and 2. Because the two invariants guarantee that parents do not overlap with their descendants in terms of acquired resources, they prevent any exchange of acquired channels. We thus restrict $\otimes$ and $\multimap$ to the exchange of channels that have not yet been acquired. This is not a limitation since, as we will see below, shared channel output and input are unrestricted.

Even with the above restriction in place, we still have to make sure that a received channel satisfies invariant 2. If we were to state a corresponding premise on the receiving rules, invertibility of the rules would be disturbed. To uphold invertibility, we impose a well-formedness condition on session types that ensures for a session of type $A_\mathsf{L}@\omega_l \updownarrow_{\omega_p}^{\omega_r} \otimes B_\mathsf{L}[\omega_m \updownarrow_{\omega_u}^{\omega_v}]$ that $\omega_v < \omega_p$ and, analogously, for a session of type $A_\mathsf{L}@\omega_l \updownarrow_{\omega_p}^{\omega_r} \multimap B_\mathsf{L}[\omega_m \updownarrow_{\omega_u}^{\omega_v}]$ that $\omega_v < \omega_p$. The remaining type well-formedness conditions are standard and given in Appendix B.4. Session types are checked to be well-formed upon process definition. Given type well-formedness,

we obtain the following rules for $\multimap$, noting that the right rule enforces invariant 4 by requiring an empty Φ-context. The rules for $\otimes$ are dual (Appendix B.2).

$$\frac{\Psi;\ \Gamma;\ \Phi;\ \Delta, x_{\mathsf{L}} : B_{\mathsf{L}}[\omega_m \updownarrow_{\omega_u}^{\omega_v}] \vdash Q :: (z_{\mathsf{L}} : C_{\mathsf{L}}[\omega_j \updownarrow_{\omega_k}^{\omega_n}])}{\Psi;\ \Gamma;\ \Phi;\ \Delta, x_{\mathsf{L}} : A_{\mathsf{L}}@\omega_l \updownarrow_{\omega_p}^{\omega_r} \multimap B_{\mathsf{L}}[\omega_m \updownarrow_{\omega_u}^{\omega_v}], y_{\mathsf{L}} : A_{\mathsf{L}}[\omega_l \updownarrow_{\omega_p}^{\omega_r}] \vdash \mathsf{send}\ x_{\mathsf{L}}\ y_{\mathsf{L}}\ ;Q :: (z_{\mathsf{L}} : C_{\mathsf{L}}[\omega_j \updownarrow_{\omega_k}^{\omega_n}])}\ (\text{T-}\multimap_{\text{L}_1})$$

$$\frac{\Psi;\ \Gamma;\ \Phi, x_{\mathsf{L}} : B_{\mathsf{L}}[\omega_m \updownarrow_{\omega_u}^{\omega_v}];\ \Delta \vdash Q :: (z_{\mathsf{L}} : C_{\mathsf{L}}[\omega_j \updownarrow_{\omega_k}^{\omega_n}])}{\Psi;\ \Gamma;\ \Phi, x_{\mathsf{L}} : A_{\mathsf{L}}@\omega_l \updownarrow_{\omega_p}^{\omega_r} \multimap B_{\mathsf{L}}[\omega_m \updownarrow_{\omega_u}^{\omega_v}];\ \Delta, y_{\mathsf{L}} : A_{\mathsf{L}}[\omega_l \updownarrow_{\omega_p}^{\omega_r}] \vdash \mathsf{send}\ x_{\mathsf{L}}\ y_{\mathsf{L}}\ ;Q :: (z_{\mathsf{L}} : C_{\mathsf{L}}[\omega_j \updownarrow_{\omega_k}^{\omega_n}])}\ (\text{T-}\multimap_{\text{L}_2})$$

$$\frac{\Psi;\ \Gamma;\ \cdot\ ;\ \Delta, y_{\mathsf{L}} : A_{\mathsf{L}}[\omega_l \updownarrow_{\omega_p}^{\omega_r}] \vdash P_{y_{\mathsf{L}}} :: (x_{\mathsf{L}} : B_{\mathsf{L}}[\omega_m \updownarrow_{\omega_u}^{\omega_v}])}{\Psi;\ \Gamma;\ \cdot\ ;\ \Delta \vdash y_{\mathsf{L}} \leftarrow \mathsf{recv}\ x_{\mathsf{L}}\ ;P_{y_{\mathsf{L}}} :: (x_{\mathsf{L}} : A_{\mathsf{L}}@\omega_l \updownarrow_{\omega_p}^{\omega_r} \multimap B_{\mathsf{L}}[\omega_m \updownarrow_{\omega_u}^{\omega_v}])}\ (\text{T-}\multimap_{\text{R}})$$

Since there are no invariants imposed on the shared context Γ, the rules for shared channel output and input are identical to those in $\mathsf{SILL}_{\mathsf{S}}$. The only differences are that we have two left rules and that the Φ-context of the right rule must be empty to satisfy invariant 4. The former is merely due to the tracking of possibly aliased sessions in the Φ context.

$$\frac{\Psi;\ \Gamma, y_{\mathsf{S}} : A_{\mathsf{S}}[\omega_l \updownarrow_{\omega_p}^{\omega_r}]; \Phi;\ \Delta, x_{\mathsf{L}} : B_{\mathsf{L}}[\omega_m \updownarrow_{\omega_u}^{\omega_v}] \vdash Q_{y_{\mathsf{S}}} :: (z_{\mathsf{L}} : C_{\mathsf{L}}[\omega_j \updownarrow_{\omega_k}^{\omega_n}])}{\Psi;\ \Gamma; \Phi;\ \Delta, x_{\mathsf{L}} : \exists x{:}A_{\mathsf{S}}@\omega_l \updownarrow_{\omega_p}^{\omega_r} .\, B_{\mathsf{L}}[\omega_m \updownarrow_{\omega_u}^{\omega_v}] \vdash y_{\mathsf{S}} \leftarrow \mathsf{recv}\ x_{\mathsf{L}}\ ;Q_{y_{\mathsf{S}}} :: (z_{\mathsf{L}} : C_{\mathsf{L}}[\omega_j \updownarrow_{\omega_k}^{\omega_n}])}\ (\text{T-}\exists_{\text{L}_1})$$

$$\frac{\Psi;\ \Gamma, y_{\mathsf{S}} : A_{\mathsf{S}}[\omega_l \updownarrow_{\omega_p}^{\omega_r}]; \Phi, x_{\mathsf{L}} : B_{\mathsf{L}}[\omega_m \updownarrow_{\omega_u}^{\omega_v}];\ \Delta \vdash Q_{y_{\mathsf{S}}} :: (z_{\mathsf{L}} : C_{\mathsf{L}}[\omega_j \updownarrow_{\omega_k}^{\omega_n}])}{\Psi;\ \Gamma; \Phi, x_{\mathsf{L}} : \exists x{:}A_{\mathsf{S}}@\omega_l \updownarrow_{\omega_p}^{\omega_r} .\, B_{\mathsf{L}}[\omega_m \updownarrow_{\omega_u}^{\omega_v}];\ \Delta \vdash y_{\mathsf{S}} \leftarrow \mathsf{recv}\ x_{\mathsf{L}}\ ;Q_{y_{\mathsf{S}}} :: (z_{\mathsf{L}} : C_{\mathsf{L}}[\omega_j \updownarrow_{\omega_k}^{\omega_n}])}\ (\text{T-}\exists_{\text{L}_2})$$

$$\frac{\Psi;\ \Gamma, y_{\mathsf{S}} : A_{\mathsf{S}}[\omega_l \updownarrow_{\omega_p}^{\omega_r}]; \cdot\ ;\ \Delta \vdash P :: (x_{\mathsf{L}} : B_{\mathsf{L}}[\omega_m \updownarrow_{\omega_u}^{\omega_v}])}{\Psi;\ \Gamma, y_{\mathsf{S}} : A_{\mathsf{S}}[\omega_l \updownarrow_{\omega_p}^{\omega_r}]; \cdot\ ;\ \Delta \vdash \mathsf{send}\ x_{\mathsf{L}}\ y_{\mathsf{S}}\ ;P :: (x_{\mathsf{L}} : \exists x{:}A_{\mathsf{S}}@\omega_l \updownarrow_{\omega_p}^{\omega_r} .\, B_{\mathsf{L}}[\omega_m \updownarrow_{\omega_u}^{\omega_v}])}\ (\text{T-}\exists_{\text{R}})$$

We finally consider the rules for forwarding and spawning. We allow a shared forward between processes that offer the same session at *invariant worlds*. Linear forwards, on the other hand, must be forbidden as they can alter the linear tree and thus can endanger invariants 1 and 2, even if we were to restrict linear forwarding to be world invariant. We discuss linear forwarding in Section 6.

$$\frac{}{\Psi;\ \Gamma, y_{\mathsf{S}} : A_{\mathsf{S}}[\omega_j \updownarrow_{\omega_k}^{\omega_n}] \vdash \mathsf{fwd}\ x_{\mathsf{S}}\ y_{\mathsf{S}} :: (x_{\mathsf{S}} : A_{\mathsf{S}}[\omega_j \updownarrow_{\omega_k}^{\omega_n}])}\ (\text{T-Id}_{\mathsf{S}})$$

The rules for spawning depend on the possible modes of the spawning and spawned processes: (T-Spawn$_{\mathsf{LL}}$) specifies how a linear process can spawn another linear process; (T-Spawn$_{\mathsf{SS}}$) defines how shared processes can spawn other shared processes (Appendix B.2 details the spawning of a shared process by a linear process). The rules are checked relative to a process definition found in the signature Σ and to a world substitution mapping $\gamma : |\Psi| \to |\Psi'|$, such that for each $\delta \in \Psi'$ we have $\Psi \vdash \gamma(\delta)$, where $|\Psi|$ denotes the *field* of Ψ (i.e., the union of its domain and range). As usual, we lift substitution to types $\hat{\gamma}(A_m)$, contexts $\hat{\gamma}(\Gamma)$, and orders $\hat{\gamma}(\Psi)$. Both rules ensure that, given the mapping γ, the order Ψ of the spawning process entails that of the process definition ($\Psi \vdash \hat{\gamma}(\Psi')$). The linear spawn rule (T-Spawn$_{\mathsf{LL}}$) further enforces invariant 2 for the spawned child. We note that the spawned child enters the linear context Δ in the spawning process' continuation since no aliases to such a process can exist at this point.

Process definitions, on the other hand, are checked such that the world order is acyclic, that all types are well-formed, and that the process satisfies invariants 1 and 2. The interested reader can find the corresponding rules in Appendix B.4. We note that we provide a variant of rule (T-SPAWN$_{\mathsf{LL}}$) to accommodate a linear recursive tail call. Since we disallow linear forwarding, a linear tail call can no longer be implicitly "de-sugared" into a spawn and a linear forward [43,2,19], but can be accounted for explicitly. The rule is given in Appendix B.2.

$$\frac{\begin{array}{c}\Delta_1 = \overline{y_{\mathsf{L}} : B_{\mathsf{L}}[\omega_m \updownarrow_{\omega_u}^{\omega_v}]} \qquad \Phi_1 = \overline{\tilde{y}_{\mathsf{L}} : \tilde{B}_{\mathsf{L}}[\tilde{\omega}_m \updownarrow_{\tilde{\omega}_u}^{\tilde{\omega}_v}]} \qquad \Gamma_1 = \overline{z_{\mathsf{S}} : C_{\mathsf{S}}[\omega_l \updownarrow_{\omega_p}^{\omega_r}]} \\ (\Psi' \vdash x'_{\mathsf{L}} : A'_{\mathsf{L}}[\delta_j \updownarrow_{\delta_k}^{\delta_n}] \leftarrow X_{\mathsf{L}} \leftarrow \Delta', \Phi', \Gamma' = P_{x'_{\mathsf{L}}, \mathsf{dom}(\Delta'), \mathsf{dom}(\Phi'), \mathsf{dom}(\Gamma'), \Psi''}) \in \Sigma \\ \hat{\gamma}(A'_{\mathsf{L}}[\delta_j \updownarrow_{\delta_k}^{\delta_n}]) = A_{\mathsf{L}}[\omega_j \updownarrow_{\omega_k}^{\omega_n}] \qquad \hat{\gamma}(\Delta') = \Delta_1 \qquad \hat{\gamma}(\Phi') = \Phi_1 \qquad \hat{\gamma}(\Gamma') = \Gamma_1 \qquad \Psi \vdash \hat{\gamma}(\Psi') \\ \Psi^+ \vdash \omega_t < \omega_k \\ \Psi;\ \Gamma_1, \Gamma_2;\ \Phi_2;\ \Delta_2, x_{\mathsf{L}} : A_{\mathsf{L}}[\omega_j \updownarrow_{\omega_k}^{\omega_n}] \vdash Q_{x_{\mathsf{L}}} :: (z''_{\mathsf{L}} : D_{\mathsf{L}}[\omega_i \updownarrow_{\omega_q}^{\omega_t}])\end{array}}{\Psi;\ \Gamma_1, \Gamma_2;\ \Phi_1, \Phi_2;\ \Delta_1, \Delta_2 \vdash x_{\mathsf{L}} : A_{\mathsf{L}}[\omega_j \updownarrow_{\omega_k}^{\omega_n}] \leftarrow X_{\mathsf{L}} \leftarrow \overline{y_{\mathsf{L}}}, \overline{\tilde{y}_{\mathsf{L}}}, \overline{z_{\mathsf{S}}};\ Q_{x_{\mathsf{L}}} :: (z''_{\mathsf{L}} : D_{\mathsf{L}}[\omega_i \updownarrow_{\omega_q}^{\omega_t}])} \text{ (T-SPAWN}_{\mathsf{LL}}\text{)}$$

$$\frac{\begin{array}{c}\Gamma_1 = \overline{z_{\mathsf{S}} : C_{\mathsf{S}}[\omega_l \updownarrow_{\omega_p}^{\omega_r}]} \qquad (\Psi' \vdash x'_{\mathsf{S}} : A'_{\mathsf{S}}[\delta_j \updownarrow_{\delta_k}^{\delta_n}] \leftarrow X_{\mathsf{S}} \leftarrow \Gamma' = P_{x'_{\mathsf{S}}, \mathsf{dom}(\Gamma'), \Psi''}) \in \Sigma \\ \hat{\gamma}(A'_{\mathsf{S}}[\delta_j \updownarrow_{\delta_k}^{\delta_n}]) = A_{\mathsf{S}}[\omega_j \updownarrow_{\omega_k}^{\omega_n}] \qquad \hat{\gamma}(\Gamma') = \Gamma_1 \qquad \Psi \vdash \hat{\gamma}(\Psi') \\ \Psi;\ \Gamma_1, \Gamma_2, x_{\mathsf{S}} : A_{\mathsf{S}}[\omega_j \updownarrow_{\omega_k}^{\omega_n}] \vdash Q_{x_{\mathsf{S}}} :: (z''_{\mathsf{S}} : D_{\mathsf{S}}[\omega_i \updownarrow_{\omega_q}^{\omega_t}])\end{array}}{\Psi;\ \Gamma_1, \Gamma_2 \vdash x_{\mathsf{S}} : A_{\mathsf{S}}[\omega_j \updownarrow_{\omega_k}^{\omega_n}] \leftarrow X_{\mathsf{S}} \leftarrow \overline{z_{\mathsf{S}}};\ Q_{x_{\mathsf{S}}} :: (z''_{\mathsf{S}} : D_{\mathsf{S}}[\omega_i \updownarrow_{\omega_q}^{\omega_t}])} \text{ (T-SPAWN}_{\mathsf{SS}}\text{)}$$

3.3 Dining Philosophers in SILL$_{\mathsf{S}^+}$

Having introduced our type system, we revisit the dining philosophers from Section 1 and show how to program the example in SILL$_{\mathsf{S}^+}$, ensuring that the program will run without deadlocks. The code is given in Figure 4. We note the world annotations in the signature of the process definitions. For instance,

$$\mathit{thinking} : \{\delta_0 < \delta_1, \delta_1 < \delta_2, \delta_2 < \delta_3 \vdash \mathsf{phil}[\delta_0 \updownarrow_{\delta_1}^{\delta_2}] \leftarrow \mathsf{sfork}[\delta_1 \updownarrow_{\delta_3}^{\delta_3}], \mathsf{sfork}[\delta_2 \updownarrow_{\delta_3}^{\delta_3}]; \cdot; \cdot\}$$

indicates that, given the order $\delta_0 < \delta_1 < \delta_2 < \delta_3$, process *thinking* provides a session of type $\mathsf{phil}[\delta_0 \updownarrow_{\delta_1}^{\delta_2}]$ using two shared channel references of type $\mathsf{sfork}[\delta_1 \updownarrow_{\delta_3}^{\delta_3}]$ and $\mathsf{sfork}[\delta_2 \updownarrow_{\delta_3}^{\delta_3}]$, respectively. The two $\cdot$ indicate that neither acquired nor linear channel references are given as arguments. The signature indicates that the two shared fork references reside at different worlds, such that the world of the first one is smaller than that of the second. Let's briefly convince ourselves that the two acquires in process *thinking* are type-correct. For each acquire we have to show that: the world of the resource to be acquired is within the acquiring process' range; the max of the acquiring process is smaller than the min of the acquired resource; and, that the self of the acquired resource is larger than those of all already acquired resources. We can convince ourselves that all those conditions are readily met. We note, however, that if we were to swap the two acquires, the program would not type-check.

$$
\begin{array}{l}
\mathit{thinking} : \{\delta_0 < \delta_1, \delta_1 < \delta_2, \delta_2 < \delta_3 \vdash \\
\quad \mathsf{phil}[\delta_0 \updownarrow_{\delta_1}^{\delta_2}] \leftarrow \mathsf{sfork}[\delta_1 \updownarrow_{\delta_3}^{\delta_3}], \mathsf{sfork}[\delta_2 \updownarrow_{\delta_3}^{\delta_3}]; \cdot; \cdot\} \\
c[\delta_0 \updownarrow_{\delta_1}^{\delta_2}] \leftarrow \mathit{thinking} \leftarrow \mathit{left}[\delta_1 \updownarrow_{\delta_3}^{\delta_3}], \mathit{right}[\delta_2 \updownarrow_{\delta_3}^{\delta_3}] = \\
\quad \mathit{left}' \leftarrow \mathsf{acquire}\ \mathit{left}\ ; \\
\quad \mathit{right}' \leftarrow \mathsf{acquire}\ \mathit{right}\ ; \\
\quad c \leftarrow \mathit{eating} \leftarrow \mathit{left}', \mathit{right}'\ ;
\end{array}
$$

$$
\begin{array}{l}
\mathsf{lfork} = \downarrow_{\mathsf{L}}^{\mathsf{S}} \mathsf{sfork} \\
\mathsf{sfork} = \uparrow_{\mathsf{L}}^{\mathsf{S}} \mathsf{lfork} \\
\mathsf{phil} = \mathbf{1}
\end{array}
$$

$$
\begin{array}{l}
\mathit{eating} : \{\delta_0 < \delta_1, \delta_1 < \delta_2, \delta_2 < \delta_3 \vdash \\
\quad \mathsf{phil}[\delta_0 \updownarrow_{\delta_1}^{\delta_2}] \leftarrow \cdot; \mathsf{lfork}[\delta_1 \updownarrow_{\delta_3}^{\delta_3}], \mathsf{lfork}[\delta_2 \updownarrow_{\delta_3}^{\delta_3}]; \cdot\} \\
c[\delta_0 \updownarrow_{\delta_1}^{\delta_2}] \leftarrow \mathit{eating} \leftarrow \mathit{left}'[\delta_1 \updownarrow_{\delta_3}^{\delta_3}], \mathit{right}'[\delta_2 \updownarrow_{\delta_3}^{\delta_3}] = \\
\quad \mathit{right} \leftarrow \mathsf{release}\ \mathit{right}'\ ; \\
\quad \mathit{left} \leftarrow \mathsf{release}\ \mathit{left}'\ ; \\
\quad c \leftarrow \mathit{thinking} \leftarrow \mathit{left}, \mathit{right}
\end{array}
$$

$$
\begin{array}{l}
\mathit{fork\_proc} : \{\delta_0 < \delta_1 \vdash \mathsf{sfork}[\delta_0 \updownarrow_{\delta_1}^{\delta_1}]\} \\
c[\delta_0 \updownarrow_{\delta_1}^{\delta_1}] \leftarrow \mathit{fork\_proc} = \\
\quad c' \leftarrow \mathsf{accept}\ c\ ; \\
\quad c \leftarrow \mathsf{detach}\ c'\ ; \\
\quad c'' : \mathsf{sfork}[\delta_0 \updownarrow_{\delta_1}^{\delta_1}] \leftarrow \mathit{fork\_proc}\ ; \\
\quad \mathsf{fwd}\ c\ c''
\end{array}
$$

Fig. 4: Deadlock-free version of dining philosophers in $\mathsf{SILL}_{\mathsf{S}^+}$.

Let us once more set the table for three philosophers and three forks. We execute this code in a process with world annotations $[\delta_a \updownarrow_{\delta_b}^{\delta_b}]$ such that $\delta_a < \delta_b$. We first create new worlds and define their order:

$$
\mathsf{w}_1 \leftarrow \mathsf{new\_world}\ ;\ \mathsf{w}_2 \leftarrow \mathsf{new\_world}\ ;\ \mathsf{w}_3 \leftarrow \mathsf{new\_world}\ ;\ \mathsf{w}_4 \leftarrow \mathsf{new\_world}\ ; \\
\delta_b < \mathsf{w}_1\ ;\ \mathsf{w}_1 < \mathsf{w}_2\ ;\ \mathsf{w}_2 < \mathsf{w}_3\ ;\ \mathsf{w}_3 < \mathsf{w}_4\ ;
$$

We then spawn the forks, each residing at a different world, such that the max world of a fork is higher than the self of the highest fork, ensuring invariant 2 for the philosopher processes that we spawn afterwards:

$$
f_1 : \mathsf{sfork}[\mathsf{w}_1 \updownarrow_{\mathsf{w}_4}^{\mathsf{w}_4}] \leftarrow \mathit{fork\_proc}\ ;\ f_2 : \mathsf{sfork}[\mathsf{w}_2 \updownarrow_{\mathsf{w}_4}^{\mathsf{w}_4}] \leftarrow \mathit{fork\_proc}\ ; \\
f_3 : \mathsf{sfork}[\mathsf{w}_3 \updownarrow_{\mathsf{w}_4}^{\mathsf{w}_4}] \leftarrow \mathit{fork\_proc}\ ;
$$

When we spawn the philosophers, we ensure that P_0 is going to pick up fork F_1 and then F_2, P_1 is going to pick up F_2 and then F_3, and P_2 is going to pick up F_1 and then F_3.

$$
p_0 : \mathsf{phil}[\delta_a \updownarrow_{\mathsf{w}_1}^{\mathsf{w}_2}] \leftarrow \mathit{thinking} \leftarrow \cdot; \cdot; f_1, f_2\ ;\ p_1 : \mathsf{phil}[\delta_a \updownarrow_{\mathsf{w}_2}^{\mathsf{w}_3}] \leftarrow \mathit{thinking} \leftarrow \cdot; \cdot; f_2, f_3\ ; \\
p_2 : \mathsf{phil}[\delta_a \updownarrow_{\mathsf{w}_1}^{\mathsf{w}_3}] \leftarrow \mathit{thinking} \leftarrow \cdot; \cdot; f_1, f_3\ ;
$$

We note that the deadlocking spawn

$$
p_2 : \mathsf{phil}[\delta_a \updownarrow_{\mathsf{w}_1}^{\mathsf{w}_3}] \leftarrow \mathit{thinking} \leftarrow \cdot; \cdot; f_3, f_1\ ;
$$

is type-incorrect since we would substitute both w_1 and w_3 for δ_1 and w_3 and w_1 for δ_2, which violates the ordering constraints put in place by typing.

3.4 Dynamics

We now give the *dynamics* of $\mathsf{SILL}_{\mathsf{S}^+}$. Our current system is based on a *synchronous* dynamics. While this choice is more conservative, it allows us to narrow the complexity of the problem at hand.

As in $\mathsf{SILL}_{\mathsf{S}}$, we use *multiset rewriting rules* [10] to capture the dynamics of $\mathsf{SILL}_{\mathsf{S}^+}$ (see Section 2). Multiset rewriting rules represent computation in terms of local state transitions between configurations of processes, only mentioning the parts of a configuration they rewrite. We use the predicates $\mathsf{proc}(a_m, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P_{a_m})$ and $\mathsf{unavail}(a_{\mathsf{S}}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}})$ to define the states of a configuration (see Section 5.1). The former denotes a process executing term P that provides along channel a_m at mode m with worlds $\mathsf{w}_{\mathsf{a}_1}$, $\mathsf{w}_{\mathsf{a}_2}$, and $\mathsf{w}_{\mathsf{a}_3}$ for self, min, and max, respectively. The latter acts as a placeholder for a shared process providing along channel a_{S} with worlds $\mathsf{w}_{\mathsf{a}_1}$, $\mathsf{w}_{\mathsf{a}_2}$, and $\mathsf{w}_{\mathsf{a}_3}$ for self, min, and max, respectively, that is currently unavailable. We note that since worlds are also run-time artifacts, they must occur as part of the state-defining predicates.

Figure 5 lists selected rules of the dynamics. Since the rules remain largely the same as those of $\mathsf{SILL}_{\mathsf{S}}$, apart from the world annotations that are "threaded through" unchanged, we only discuss those that actually differ from the $\mathsf{SILL}_{\mathsf{S}}$ rules. The interested reader can find the remaining rules in Appendix C.

(D-Spawn$_{\mathsf{LL}}$)
$\mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, x_{\mathsf{L}} : A_{\mathsf{L}}[\mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}] \leftarrow X_{\mathsf{L}} \leftarrow \overline{c_{\mathsf{L}}}, \overline{\tilde{c}_{\mathsf{L}}}, \overline{d_{\mathsf{S}}};\ Q_{x_{\mathsf{L}}}),$
$\mathsf{!def}(\Psi' \vdash x'_{\mathsf{L}} : A'_{\mathsf{L}}[\delta_j\updownarrow^{\delta_n}_{\delta_k}] \leftarrow X_{\mathsf{L}} \leftarrow \Delta', \Phi', \Gamma' = P_{x'_{\mathsf{L}},\mathsf{dom}(\Delta'),\mathsf{dom}(\Phi'),\mathsf{dom}(\Gamma'),\Psi''})$
$\longrightarrow \mathsf{proc}(b_{\mathsf{L}}, \mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}, [b_{\mathsf{L}}/x'_{\mathsf{L}}, \overline{c_{\mathsf{L}}}/\mathsf{dom}(\Delta'), \overline{\tilde{c}_{\mathsf{L}}}/\mathsf{dom}(\Phi'), \overline{d_{\mathsf{S}}}/\mathsf{dom}(\Gamma')]\hat{\gamma}(P_{x'_{\mathsf{L}},\mathsf{dom}(\Delta'),\mathsf{dom}(\Phi'),\mathsf{dom}(\Gamma'),\Psi''})),$
$\quad \mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, [b_{\mathsf{L}}/x_{\mathsf{L}}]Q_{x_{\mathsf{L}}}),$
$\quad \mathsf{unavail}(b_{\mathsf{S}}, \mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}) \quad (b\ \textit{fresh})$

(D-New)
$\mathsf{proc}(a, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, \mathsf{w} \leftarrow \mathsf{new\_world};\ Q_{\mathsf{w}'}) \longrightarrow \mathsf{proc}(a, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, Q_{\mathsf{w}'}) \quad (\mathsf{w}\ \textit{fresh})$

(D-Ord)
$\mathsf{proc}(a, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, \mathsf{w} < \mathsf{w}';\ Q) \longrightarrow \mathsf{proc}(a, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, Q)$

Fig. 5: Selected multiset rewriting rules of $\mathsf{SILL}_{\mathsf{S}^+}$.

Noteworthy are the rules D-New and D-Ord for the constructs for creating and relating worlds, respectively. Rule D-New creates a fresh world, which will be globally available in the configuration. Rule D-Ord, on the other hand, updates the configuration's order with the pair $\mathsf{w} < \mathsf{w}'$. Rule D-New, lastly, substitutes actual worlds for world variables in the body of the spawned process, using the substitution mapping γ defined earlier. It relies on the existence of a corresponding definition predicate for each process defintion contained in the signature Σ. We note that the substitution γ in rule D-Spawn$_{\mathsf{LL}}$ instantiates the appropriate world variables in the spawned process P.

4 Extended Example: An Imperative Shared Queue

We now develop a shared implementation of an imperative queue that uses an auxiliary shared list process to store the queue's elements and has shared references to the front and the back of the list to dequeue and enqueue elements, respectively. The session types for the queue and the list are[4]:

$$\mathsf{queue}\ A_\mathsf{s} = \uparrow^\mathsf{S}_\mathsf{L} \& \{\mathsf{enq} : \Pi x{:}A_\mathsf{s}.\downarrow^\mathsf{S}_\mathsf{L}\mathsf{queue}\ A_\mathsf{s},\ \mathsf{deq} : \oplus\{\mathsf{none} : \downarrow^\mathsf{S}_\mathsf{L}\mathsf{queue}\ A_\mathsf{s},\ \mathsf{some} : \exists x{:}A_\mathsf{s}.\downarrow^\mathsf{S}_\mathsf{L}\mathsf{queue}\ A_\mathsf{s}\}\}$$

$$\mathsf{list}\ A_\mathsf{s} = \uparrow^\mathsf{S}_\mathsf{L} \& \{\mathsf{ins} : \Pi x{:}A_\mathsf{s}.\exists y{:}\mathsf{list}\ A_\mathsf{s}.\downarrow^\mathsf{S}_\mathsf{L}\mathsf{list}\ A_\mathsf{s}\ \mathsf{del} : \oplus\{\mathsf{none} : \downarrow^\mathsf{S}_\mathsf{L}\mathsf{list}\ A_\mathsf{s},\ \mathsf{some} : \exists x{:}A_\mathsf{s}.\downarrow^\mathsf{S}_\mathsf{L}\mathsf{list}\ A_\mathsf{s}\}$$

The list is implemented in terms of processes *empty* and *elem*, denoting the empty list and a cons cell, respectively. We show the more interesting case of a cons cell (Figure 6) and provide the implementation of an empty list in Appendix A. The queue is defined by processes *head* (Figure 7) and *queue_proc*, the latter being the queue's interface to its clients.

$$elem : \{\delta_1 < \delta_2, \delta_2 < \delta_3, \delta_3 < \delta_4 \vdash \mathsf{list}[\delta_1\updownarrow^{\delta_2}_{\delta_2}]A_\mathsf{s}[\delta_3\updownarrow^{\delta_4}_{\delta_4}] \leftarrow A_\mathsf{s}[\delta_3\updownarrow^{\delta_4}_{\delta_4}], \mathsf{list}[\delta_1\updownarrow^{\delta_2}_{\delta_2}]A_\mathsf{s}[\delta_3\updownarrow^{\delta_4}_{\delta_4}]\}$$

$$c[\delta_1\updownarrow^{\delta_2}_{\delta_2}][\delta_3\updownarrow^{\delta_4}_{\delta_4}] \leftarrow elem \leftarrow x[\delta_3\updownarrow^{\delta_4}_{\delta_4}], next[\delta_1\updownarrow^{\delta_2}_{\delta_2}][\delta_3\updownarrow^{\delta_4}_{\delta_4}] =$$

$c' \leftarrow \mathsf{accept}\ c\ ;$
$\mathsf{case}\ c'\ \mathsf{of}$
$\mid \mathsf{ins} \rightarrow y \leftarrow \mathsf{recv}\ c'\ ;$
$\quad n \leftarrow elem \leftarrow y,\ next\ ;$
$\quad \mathsf{send}\ c'\ n\ ;$
$\quad c \leftarrow \mathsf{detach}\ c'\ ;$
$\quad c'' : \mathsf{list}[\delta_1\updownarrow^{\delta_2}_{\delta_2}]A_\mathsf{s}[\delta_3\updownarrow^{\delta_4}_{\delta_4}] \leftarrow elem \leftarrow x,\ n\ ;\ \mathsf{fwd}\ c\ c''$
$\mid \mathsf{del} \rightarrow c'.\mathsf{some}\ ;$
$\quad \mathsf{send}\ c'\ x\ ;$
$\quad c \leftarrow \mathsf{detach}\ c'\ ;\ \mathsf{fwd}\ c\ next$

Fig. 6: Shared Queue Implementation – *elem* process.

We can now define a client (Figure 9) for the queue, assuming existence of a corresponding shared session type item and process *item_proc* offering a session of type $\mathsf{item}[\delta_3\updownarrow^{\delta_4}_{\delta_4}]$. The client instantiates the queue at world δ_b, allowing it to acquire resources at world w_1, which is exactly the world at which process *queue_proc* instantiates the list. Given that the client itself resides at world δ_a, which is smaller than the queue's world δ_b, the client is allowed to acquire the queue, which in turn will acquire the list to satisfy any requests by the

[4] Polymorphism is orthogonal to the development in this paper, so we adopt it for the example without formal treatment. Polymorphic session types have been developed in [39,20], and we expect the results to be applicable in our context.

$head : \{\delta_0 < \delta_1, \delta_1 < \delta_2, \delta_2 < \delta_3, \delta_3 < \delta_4 \vdash \mathsf{queue}[\delta_0 \updownarrow^{\delta_1}_{\delta_1}] A_{\mathsf{S}}[\delta_3 \updownarrow^{\delta_4}_{\delta_4}] \leftarrow \mathsf{list}[\delta_1 \updownarrow^{\delta_2}_{\delta_2}] A_{\mathsf{S}}[\delta_3 \updownarrow^{\delta_4}_{\delta_4}], \mathsf{list}[\delta_1 \updownarrow^{\delta_2}_{\delta_2}] A_{\mathsf{S}}[\delta_3 \updownarrow^{\delta_4}_{\delta_4}]\}$

$c[\delta_0 \updownarrow^{\delta_1}_{\delta_1}][\delta_3 \updownarrow^{\delta_4}_{\delta_4}] \leftarrow head \leftarrow front[\delta_1 \updownarrow^{\delta_2}_{\delta_2}][\delta_3 \updownarrow^{\delta_4}_{\delta_4}],\ back[\delta_1 \updownarrow^{\delta_2}_{\delta_2}][\delta_3 \updownarrow^{\delta_4}_{\delta_4}] = c' \leftarrow \mathsf{accept}\ c\ ;$
$\mathsf{case}\ c'\ \mathsf{of}$
$\mid \mathsf{enq} \rightarrow x \leftarrow \mathsf{recv}\ c'\ ;$
$\quad back' \leftarrow \mathsf{acquire}\ back\ ;$
$\quad back'.\mathsf{ins}\ ;\ \mathsf{send}\ back'\ x\ ;$
$\quad e \leftarrow \mathsf{recv}\ back'\ ;$
$\quad back \leftarrow \mathsf{release}\ back'\ ;$
$\quad c \leftarrow \mathsf{detach}\ c'\ ;$
$\quad c'' : \mathsf{queue}[\delta_0 \updownarrow^{\delta_1}_{\delta_1}] A_{\mathsf{S}}[\delta_3 \updownarrow^{\delta_4}_{\delta_4}] \leftarrow head \leftarrow front,\ e\ ;\ \mathsf{fwd}\ c\ c''$
$\mid \mathsf{deq} \rightarrow front' \leftarrow \mathsf{acquire}\ front\ ;$
$\quad front'.\mathsf{del}\ ;$
$\quad (\mathsf{case}\ front'\ \mathsf{of}$
$\quad \mid \mathsf{none} \rightarrow front \leftarrow \mathsf{release}\ front'\ ;$
$\quad\quad c'.\mathsf{none}\ ;$
$\quad\quad c \leftarrow \mathsf{detach}\ c'\ ;$
$\quad\quad c'' : \mathsf{queue}[\delta_0 \updownarrow^{\delta_1}_{\delta_1}] A_{\mathsf{S}}[\delta_3 \updownarrow^{\delta_4}_{\delta_4}] \leftarrow head \leftarrow front,\ back\ ;\ \mathsf{fwd}\ c\ c''$
$\quad \mid \mathsf{some} \rightarrow x \leftarrow \mathsf{recv}\ front'\ ;$
$\quad\quad front \leftarrow \mathsf{release}\ front'\ ;$
$\quad\quad c'.\mathsf{some}\ ;\ \mathsf{send}\ c'\ x\ ;$
$\quad\quad c \leftarrow \mathsf{detach}\ c'\ ;$
$\quad\quad c'' : \mathsf{queue}[\delta_0 \updownarrow^{\delta_1}_{\delta_1}] A_{\mathsf{S}}[\delta_3 \updownarrow^{\delta_4}_{\delta_4}] \leftarrow head \leftarrow front,\ back\ ;\ \mathsf{fwd}\ c\ c'')$

Fig. 7: Shared Queue Implementation – *head* process.

$queue\_proc : \{\delta_0 < \delta_1, \delta_1 < \delta_3, \delta_3 < \delta_4 \vdash \mathsf{queue}[\delta_0 \updownarrow^{\delta_1}_{\delta_1}] A_{\mathsf{S}}[\delta_3 \updownarrow^{\delta_4}_{\delta_4}]\}$
$c[\delta_0 \updownarrow^{\delta_1}_{\delta_1}][\delta_3 \updownarrow^{\delta_4}_{\delta_4}] \leftarrow queue\_proc =$
$\mathsf{w}_2 \leftarrow \mathsf{new\_world}\ ;$
$\delta_1 < \mathsf{w}_2\ ;\ \mathsf{w}_2 < \delta_3\ ;$
$front : \mathsf{list}[\delta_1 \updownarrow^{\mathsf{w}_2}_{\mathsf{w}_2}] A_{\mathsf{S}}[\delta_3 \updownarrow^{\delta_4}_{\delta_4}] \leftarrow empty\ ;$
$back : \mathsf{list}[\delta_1 \updownarrow^{\mathsf{w}_2}_{\mathsf{w}_2}] A_{\mathsf{S}}[\delta_3 \updownarrow^{\delta_4}_{\delta_4}] \leftarrow empty\ ;$
$c'' : \mathsf{queue}[\delta_0 \updownarrow^{\delta_1}_{\delta_1}] A_{\mathsf{S}}[\delta_3 \updownarrow^{\delta_4}_{\delta_4}] \leftarrow head \leftarrow front, back\ ;\ \mathsf{fwd}\ c\ c''$

Fig. 8: Shared Queue Implementation – *queue_proc* process.

client. The example showcases a paradigmatic use of several collaborators, where collaborators can hold resources while they "talk down" in the tree.

5 Semantics

In this section, we discuss the meta-theoretical properties of $\mathsf{SILL}_{\mathsf{S}^+}$. Our main focus is $\mathsf{SILL}_{\mathsf{S}^+}$' strong progress guarantee.

$client : \{\delta_a < \delta_b \vdash \mathbf{1}[\delta_a \updownarrow^{\delta_b}_{\delta_b}]\}$

$c[\delta_a \updownarrow^{\delta_b}_{\delta_b}] \leftarrow client =$
$\mathsf{w}_1 \leftarrow \mathsf{new\_world}$; $\mathsf{w}_3 \leftarrow \mathsf{new\_world}$; $\mathsf{w}_4 \leftarrow \mathsf{new\_world}$;
$\delta_b < \mathsf{w}_1$; $\mathsf{w}_1 < \mathsf{w}_3$; $\mathsf{w}_3 < \mathsf{w}_4$;
$i_0 : \mathsf{item}[\mathsf{w}_3 \updownarrow^{\mathsf{w}_4}_{\mathsf{w}_4}] \leftarrow item\_proc$;
$q : \mathsf{queue}[\delta_b \updownarrow^{\mathsf{w}_1}_{\mathsf{w}_1}] A_{\mathsf{S}}[\mathsf{w}_3 \updownarrow^{\mathsf{w}_4}_{\mathsf{w}_4}] \leftarrow queue\_proc$;
$q' \leftarrow \mathsf{acquire}\ q$;
$q'.\mathsf{enq}$;
$\mathsf{send}\ q'\ i_0$;
$q \leftarrow \mathsf{release}\ q'$;
$\mathsf{close}\ c$

Fig. 9: Shared Queue Implementation – Client.

5.1 Configuration Typing and Preservation

Given the hierarchy between mode S and L and the fact that shared processes cannot depend on linear processes, we divide a configuration into a *shared* part Λ and a linear part Θ. We use the typing judgment $\Psi; \Gamma \vDash \Lambda; \Theta :: \Gamma; \Phi, \Delta$ to type configurations. The judgment expresses that a well-formed configuration $\Lambda; \Theta$ provides the shared channels in Γ and the linear channels in Φ and Δ. A configuration is type-checked relative to all shared channel references and a global order Ψ. While type-checking is compositional insofar as each process definition can be type-checked separately, solely relying on the process' local Ψ (and Γ), at run-time, the entire order that a configuration relies upon is considered. We give the configuration typing rules in Figure 10.

Our progress theorem crucially depends on the guarantee that the invariants 1 and 2 from Section 3 hold for every linear process in a configuration's tree. This is expressed by the premises $\mathsf{Inv}_3(\mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P_{a_{\mathsf{L}}}))$ and $\mathsf{Inv}_4(\mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P_{a_{\mathsf{L}}}))$ in rule (T-Θ_2), based on the definitions 3 and 4 below that restate invariants 1 and 2 for an entire configuration. We note that Invariant 4 is based on the set of all transitive children (i.e., *descendants*) of a process. We formally define the notion of a descendant inductively over a well-typed linear configuration. The interested reader can find the definition in Appendix B.3.

Invariant 1 $(\mathsf{min}(\mathsf{parent}) \leq \mathsf{self}(\mathsf{acquired\_child}) \leq \mathsf{max}(\mathsf{parent}))$**.** *If* $\Psi; \Gamma \vDash \Theta :: \Phi, \Delta$ *and for any* $\mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P_{a_{\mathsf{L}}}) \in \Theta$ *such that* $\Psi; \Gamma; \Phi_1; \Delta_1 \vdash P_{a_{\mathsf{L}}} :: (a_{\mathsf{L}} : A_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])$, $\mathsf{Inv}_3(\mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P_{a_{\mathsf{L}}}))$ *holds if an only if for every acquired resource* $b_{\mathsf{L}} : B_{\mathsf{L}}[\mathsf{w}_{\mathsf{b}_1} \updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}] \in \Phi_1$ *it holds that* $\Psi^* \vdash \mathsf{w}_{\mathsf{a}_2} \leq \mathsf{w}_{\mathsf{b}_1} \leq \mathsf{w}_{\mathsf{a}_3}$*. Moreover, if* $P_{a_{\mathsf{L}}} = x_{\mathsf{L}} \leftarrow \mathsf{acquire}\, c_{\mathsf{S}}\, ; Q_{x_{\mathsf{L}}}$*, for a* $(c_{\mathsf{S}} : \uparrow^{\mathsf{S}}_{\mathsf{L}} C_{\mathsf{L}}[\mathsf{w}_{\mathsf{c}_1} \updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}]) \in \Gamma$*, then, for every acquired resource* $b_{\mathsf{L}} : B_{\mathsf{L}}[\mathsf{w}_{\mathsf{b}_1} \updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}] \in \Phi_1$*, it holds that* $\Psi^+ \vdash \mathsf{w}_{\mathsf{b}_1} < \mathsf{w}_{\mathsf{c}_1}$ *and that* $\Psi^* \vdash \mathsf{w}_{\mathsf{a}_2} \leq \mathsf{w}_{\mathsf{c}_1} \leq \mathsf{w}_{\mathsf{a}_3}$*.*

Invariant 2 $(\mathbf{max(parent) < minima(descendants)})$**.** *If* $\Psi; \Gamma \vDash \Theta :: \Phi, \Delta$ *and for any* $\mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P_{a_{\mathsf{L}}}) \in \Theta$ *and that process' descendants* $(\Psi; \Gamma \vDash$

$$\frac{}{\Psi; \Gamma \vDash (\cdot) :: (\cdot)}\ (\text{T-}\Theta_1)$$

$$\frac{\begin{array}{c}(a_{\mathsf{S}} : \hat{B}[\mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}]) \in \Gamma \qquad \vdash (A_{\mathsf{L}}, \hat{B})\ \mathsf{sesync} \qquad \Psi \vdash A_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}]\ \mathsf{type} \\ \Psi^* \vdash \mathsf{w}_{\mathsf{a}_2} \leq \mathsf{w}_{\mathsf{a}_3} \qquad \mathsf{Inv}_3(\mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P_{a_{\mathsf{L}}})) \qquad \mathsf{Inv}_4(\mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P_{a_{\mathsf{L}}})) \\ \Psi; \Gamma \vDash \Theta :: \Phi, \Phi_1, \Delta, \Delta_1 \qquad \Psi; \Gamma; \Phi_1; \Delta_1 \vdash P_{a_{\mathsf{L}}} :: (a_{\mathsf{L}} : A_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])\end{array}}{\Psi; \Gamma \vDash \Theta, \mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P_{a_{\mathsf{L}}}) :: (\Phi, \Delta, a_{\mathsf{L}} : A_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])}\ (\text{T-}\Theta_2)$$

$$\frac{}{\Psi; \Gamma \vDash (\cdot) :: (\cdot)}\ (\text{T-}\Lambda_1) \qquad \frac{\begin{array}{c}\vdash (\uparrow^{\mathsf{S}}_{\mathsf{L}} A_{\mathsf{L}}, \top)\ \mathsf{sesync} \qquad \Psi \vdash \uparrow^{\mathsf{S}}_{\mathsf{L}} A_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}]\ \mathsf{type} \\ \Psi^* \vdash \mathsf{w}_{\mathsf{a}_2} \leq \mathsf{w}_{\mathsf{a}_3} \qquad \Psi; \Gamma \vdash P_{a_{\mathsf{S}}} :: (a_{\mathsf{S}} : \uparrow^{\mathsf{S}}_{\mathsf{L}} A_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])\end{array}}{\Psi; \Gamma \vDash \mathsf{proc}(a_{\mathsf{S}}, \mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P_{a_{\mathsf{S}}}) :: (a_{\mathsf{S}} : \uparrow^{\mathsf{S}}_{\mathsf{L}} A_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])}\ (\text{T-}\Lambda_2)$$

$$\frac{}{\Psi; \Gamma \vDash \mathsf{unavail}(a_{\mathsf{S}}, \mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}) :: (a_{\mathsf{S}} : \hat{A}[\mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])}\ (\text{T-}\Lambda_3) \qquad \frac{\Psi; \Gamma \vDash \Lambda :: \Gamma_1 \qquad \Psi; \Gamma \vDash \Lambda' :: \Gamma_2}{\Psi; \Gamma \vDash \Lambda, \Lambda' :: \Gamma_1, \Gamma_2}\ (\text{T-}\Lambda_4)$$

$$\frac{\Psi; \Gamma \vDash \Lambda :: \Gamma \qquad \Psi; \Gamma \vDash \Theta :: \Phi, \Delta}{\Psi; \Gamma \vDash \Lambda; \Theta :: \Gamma; \Phi, \Delta}\ (\text{T-}\Omega)$$

Fig. 10: Configuration Typing

$\Theta :: \Phi, \Delta) \rhd a_{\mathsf{L}} = (\Phi', \Delta')$, $\mathsf{Inv}_4(\mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P_{a_{\mathsf{L}}}))$ *holds iff for every descendant* $b_{\mathsf{L}} : B_{\mathsf{L}}[\mathsf{w}_{\mathsf{b}_1} \updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}] \in (\Phi', \Delta')$ *it holds that* $\Psi^+ \vdash \mathsf{w}_{\mathsf{a}_3} < \mathsf{w}_{\mathsf{b}_2}$.

On the other hand, our preservation theorem states that Invariant 3 and Invariant 4 are preserved for every linear process in the configuration along transitions. Moreover, the theorem expresses that the types of the providing linear channels Φ and Δ are maintained along transitions and that new shared channels and worlds may be allocated. The proof relies, in particular, on session types being strictly equi-synchronizing, on a process' type well-formedness and assurance that the process' min world is less than or equal to its max world.

Theorem 5.1 (Preservation). *If* $\Psi; \Gamma \vDash \Lambda; \Theta :: \Gamma; \Phi, \Delta$ *and* $\Lambda; \Theta \longrightarrow \Lambda'; \Theta'$, *then* $\Psi'; \Gamma' \vDash \Lambda'; \Theta' :: \Gamma'; \Phi, \Delta$, *for some* Λ', Θ', Ψ', *and* Γ'.

5.2 Progress

In our development so far we have distilled the two scenarios of interdependencies between processes that can lead to deadlocks: *cyclic acquisitions* and *interdependent acquisitions and synchronizations*. This has lead to the development of a type system that ingrains the notions of *competitors* and *collaborators*, such that the former compete for a set of resources whereas the latter do not overlap in the set of resources they acquire. Our type system then ties these notions to a configuration's linear process tree such that collaborators stand in a parent-descendant relationship to each other and competitors in a sibling/cousin relationship. In this section, we prove that this orchestration is sufficient to rule out any of the aforementioned interdependencies.

To this, end we introduce the notions of *red* and *green arrows* that allow us to reason about process interdependencies in a configuration's tree. A red arrow points from a linear $\mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, Q)$ to a linear $\mathsf{proc}(b_{\mathsf{L}}, \mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}, P)$, if the former is attempting to acquire a resource held by the latter and, consequently, is waiting for the latter to release that resource. A green arrow points from a linear $\mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, Q)$ to a linear $\mathsf{proc}(b_{\mathsf{L}}, \mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}, P)$, if the former is waiting to synchronize with the latter. We define these arrows formally as follows:

Definition 5.2 (Acquire Dependency — "Red Arrow"). *Given a well-formed and well-typed configuration $\Psi; \Gamma \vDash \Lambda; \Theta :: \Gamma; \Phi, \Delta$, there exists a waiting-due-to-acquire relation $\mathcal{A}(\Theta)$ among linear processes in Θ at run-time such that*

$$\mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, x_{\mathsf{L}} \leftarrow \mathsf{acquire}\ c_{\mathsf{S}};\ Q_{x_{\mathsf{L}}}) <_{\mathcal{A}} \mathsf{proc}(b_{\mathsf{L}}, \mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}, P\langle c_{\mathsf{L}}\rangle)$$

where $P\langle c_{\mathsf{L}}\rangle$ denotes a process term with an occurrence of channel c_{L}.

Definition 5.3 (Synchronization Dependency — "Green Arrow"). *Given a well-formed and well-typed configuration $\Psi; \Gamma \vDash \Lambda; \Theta :: \Gamma; \Phi, \Delta$, there exists a waiting-due-to-synchronization relation $\mathcal{S}(\Theta)$ among linear processes in Θ at run-time such that*

$$\mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, \_\langle b_{\mathsf{L}}\rangle; Q) <_{\mathcal{S}} \mathsf{proc}(b_{\mathsf{L}}, \mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}, \_\langle \neg b_{\mathsf{L}}\rangle; P)$$
$$\mathsf{proc}(b_{\mathsf{L}}, \mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}, \_\langle b_{\mathsf{L}}\rangle; P) <_{\mathcal{S}} \mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, \_\langle \neg b_{\mathsf{L}}\rangle; Q\langle b_{\mathsf{L}}\rangle)$$

where $P\langle a_{\mathsf{L}}\rangle$ denotes a process term with an occurrence of channel b_{L}, $\_\langle a\rangle; P$ a process term that currently executes an action along channel a, and $\_\langle \neg a\rangle; P$ a process term whose currently executing action does not involve the channel a.

It may be helpful to consult Figure 3 at this point again and note the semantic difference between the violet arrows in that figure and the red arrows discussed here. Whereas violet arrows point from the acquiring process to the resource being acquired, red arrows point from the acquiring process to the process that is holding the resource. Thus, violet arrows can go out of the tree, while red arrows stay within. Given the definitions of red and green arrows, we can define the relation $\mathcal{W}(\Theta)$ on the configuration's tree, which contains all process pairs that are in some way waiting for each other:

Definition 5.4 (Waiting Dependency). *Given a well-formed and well-typed configuration $\Psi; \Gamma \vDash \Lambda; \Theta :: \Gamma; \Phi, \Delta$, there exists a waiting relation $\mathcal{W}(\Theta)$ among processes in Θ at run-time such that $\mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P) <_{\mathcal{W}} \mathsf{proc}(b_{\mathsf{L}}, \mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}, Q)$,*

- *if $\mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P) <_{\mathcal{A}} \mathsf{proc}(b_{\mathsf{L}}, \mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}, Q)$, or*
- *if $\mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P) <_{\mathcal{S}} \mathsf{proc}(b_{\mathsf{L}}, \mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}, Q)$.*

Having defined the relation $\mathcal{W}(\Theta)$, we can now state the key lemma underlying our progress theorem, indicating that $\mathcal{W}(\Theta)$ is acyclic in a well-formed and well-typed configuration:

Lemma 5.5 (Acyclicity of $\mathcal{W}(\Theta)$). *If $\Psi; \Gamma \vDash \Lambda; \Theta :: \Gamma; \Phi, \Delta$, then $\mathcal{W}(\Theta)$ is acyclic.*

The interested reader can find the proof of the lemma in Appendix D, we focus on explaining the main idea of the proof here. The proof proceeds by induction on $\Psi; \Gamma \vDash \Theta :: \Phi, \Delta$, assuming for the non-empty case $\Psi; \Gamma \vDash \Theta, \mathsf{proc}(a_{\text{L}}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P_{a_{\text{L}}}) :: (\Phi, \Delta, a_{\text{L}} : A_{\text{L}}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])$ that $\mathcal{W}(\Theta)$ is acyclic, by the inductive hypothesis. We then know that there cannot exist any paths of green and red arrows in Θ that form a cycle, and we have to show that there is no way of introducing such a cyclic path by adding node $\mathsf{proc}(a_{\text{L}}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P_{a_{\text{L}}})$ to the configuration Θ. In particular, the proof considers all possible new arrows that may be introduced by adding the node and that are necessary for creating a cycle, showing that such arrows cannot come about in a well-typed configuration.

We illustrate the reasoning for the two selected cases shown in Figure 11. Case **(a)** represents a case in which process $P_{a_{\text{L}}}$ is waiting to synchronize with its child $P_{b_{\text{L}}}$ while holding a resource a descendant of $P_{b_{\text{L}}}$ or $P_{b_{\text{L}}}$ itself wants to acquire. However, this scenario cannot come about in a well-typed configuration because $P_{a_{\text{L}}}$ and $P_{b_{\text{L}}}$ are collaborators and thus cannot overlap in resources they acquire. Case **(b)** represents a case in which process $P_{a_{\text{L}}}$ is waiting to synchronize with its child $P_{b_{\text{L}}}$ while another child, process $P_{c_{\text{L}}}$, is waiting to synchronize with $P_{a_{\text{L}}}$. Given acyclicity of $\mathcal{W}(\Theta)$, a necessary condition for a cycle to form is that there already must exist a red arrow **C** in the configuration that connects the subtrees in which the siblings $P_{b_{\text{L}}}$ and $P_{c_{\text{L}}}$ reside. However, this scenario cannot come about in a well-typed configuration because $P_{b_{\text{L}}}$ and $P_{c_{\text{L}}}$ are competitors, forcing $P_{c_{\text{L}}}$ or any of its descendant to release a resource before synchronizing with $P_{a_{\text{L}}}$. These arguments are made precise in various lemmas in Appendix D.

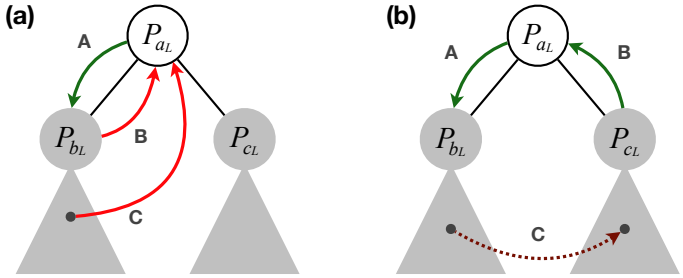


Fig. 11: Two prototypical cases in proof of acyclicty of $\mathcal{W}(\Theta)$.

Given acyclicity of $\mathcal{W}(\Theta)$, we can state and prove the following strong progress theorem. The theorem relies on the notion of a *poised* process, a process currently executing an action along its offering channel, and distinguishes a configuration only consisting of the top-level, linear "main" process from one that consists of several linear processes. We use $|\Theta|$ to denote the cardinality of Θ:

Theorem 5.6 (Progress). *If $\Psi; \Gamma \vDash \Lambda; \Theta :: (\Gamma; c_{\text{L}} : \mathbf{1}[\mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}])$, then either*

- $\Lambda \longrightarrow \Lambda'$, *for some* Λ', *or*
- Λ *is poised and*
  - *if* $|\Theta| = 1$, *then either* $\Lambda;\Theta \longrightarrow \Lambda';\Theta'$, *for some* Λ' *and* Θ', *or* Θ *is poised, or*
  - *if* $|\Theta| > 1$, *then* $\Lambda;\Theta \longrightarrow \Lambda';\Theta'$, *for some* Λ' *and* Θ'.

The theorem indicates that, as long as there exist at least two linear processes in the configuration, the configuration can always step. If the configuration only consists of the main process, then this process will become poised (i.e., ready to close), once all sub-computations are finished. The proof of the theorem is given in Appendix D and relies on the acyclicity of $\mathcal{W}(\Theta)$ and the fact that all sessions must be strictly equi-synchronizing.

6 Additional Discussion

Linear Forwarding Our current formalization does not include linear forwarding because a forward changes the process tree and thus endangers the invariants imposed on it. This means that certain programs from the purely linear fragment may not type-check in our system. However, the correspondingly η-expanded versions of these programs should be expressible and type-checkable in $\mathsf{SILL}_{\mathsf{S}^+}$. As part of future work, we want to explore the addition of the linear forward

$$\frac{\Psi^+ \vdash \omega_n < \omega_u}{\Psi;\, \Gamma;\, \cdot;\, y_{\mathsf{L}} : A_{\mathsf{L}}[\omega_m \updownarrow_{\omega_u}^{\omega_v}] \vdash \mathsf{fwd}\ x_{\mathsf{L}}\ y_{\mathsf{L}} :: (x_{\mathsf{L}} : A_{\mathsf{L}}[\omega_j \updownarrow_{\omega_k}^{\omega_n}])} \ (\text{T-Id}_{\mathsf{L}})$$

which allows forwarding to processes that are known to not yet be aliased and whose world annotations meet the premise $\Psi^+ \vdash \omega_n < \omega_u$. Restricting to processes in Δ should uphold Invariant 3, while the premise of the rule should uphold Invariant 4. However, this change will affect the inner working of the proofs, the use of inversion in particular, which might have far-reaching consequences that need to be carefully explored.

Unbounded Process Networks and World Polymorphism The typing discipline presented in the previous sections, while rich enough to account for a wide range of interesting programs, cannot type programs that spawn a statically undetermined number of shared sessions that are then to be used. For instance, while we can easily type a configuration of any given number of dining philosophers (Section 3.3), we cannot type a recursive process in which the number of philosophers (and forks) is potentially unbounded, due to the way worlds are created and propagated across processes.

The general issue lies in implementing a statically unbounded network of processes that interact with each other. These interactions require the processes to be spawned at different worlds which must be generated dynamically as needed. To interact with such a statically unknown number of processes uniformly, their offering channels must be stored in a list-like structure for later use. However,

in our system, recursive types have to be invariant with respect to worlds. For instance, in a recursive type such as $T = A_{\mathsf{L}}@\omega_l \updownarrow_{\omega_p}^{\omega_r} \otimes T$, the worlds ω_l, ω_p, ω_r are fixed in the unfoldings of T. Thus, we cannot type a world-heterogeneous list and cannot form such process networks.

Given that the issues preventing us from typing such unbounded networks lie in problems of world invariance, the natural solution is to explore some form of *world polymorphism*, where types can be parameterized by worlds which are instantiated at a later stage. Such techniques have been studied in the context of hybrid logical processes in [6] by considering session types of the form $\forall\delta.A$ and $\exists\delta.A$, sessions that are parametric in the world variable δ, that is instantiated by a concrete reachable world at runtime. While their development cannot be mapped directly to our setting, it is a promising avenue of future work.

7 Related Work

Behavioral Type Analysis of Deadlocks The addition of channel usage information to types in a concurrent, message-passing setting was pioneered by Kobayashi and Igarashi [28,25], who applied the idea to deadlock prevention in the π-calculus and later to more general properties [26,27], giving rise to a generic system that can be instantiated to produce a variety of concrete typing disciplines for the π-calculus (e.g. race detection, deadlock detection, etc.).

This line of work types π-calculus processes with a simplified form of *process* (akin to CCS [36] terms without name restriction) that characterizes the input/output behavior of processes. These types are augmented with abstract data that pertain to the relative ordering of channel actions, with the type system ensuring that the transitive closure of such orderings forms a strict partial order, ensuring deadlock-freedom (i.e., communication succeeds unless a process diverges). Building on this, Kobayashi et al. proposed type systems that ensure a stronger property dubbed lock-freedom [29] (i.e., communication always succeeds), and variants that are amenable to type inference [30,33]. Kobayashi [31] extended this latter system to more accurately account for recursive processes while preserving the existence of a type inference algorithm.

Our system draws significant inspiration from this line of work, insofar as we also equip types with abstract ordering data on certain communication actions, which is then statically enforced to form a strict partial order. We note that our $\mathsf{SILL}_{\mathsf{S}^+}$ language differs sufficiently from the pure π-calculus in terms of its constructs and semantics to make the formulation of a direct comparison or an immediate application of their work unclear (e.g., [31] uses replication to encode recursive processes). Moreover, we integrate this style of order-based reasoning with both linear and shared session typing, which interact in non-trivial ways (especially in the presence of recursive types and recursive process definitions).

In terms of typability, enforcing session fidelity can be a double-edged sword: some examples of the works above can be transposed to $\mathsf{SILL}_{\mathsf{S}^+}$ with mostly cosmetic changes and without making use of shared sessions (e.g., a parallel implementation of factorial that recurses via replication but always answers on a

private channel); others are incompatible with linear sessions and require the use of shared sessions via the acquire-release discipline, which entails a more indirect but still arguably faithful modelling of the original π-calculus behavior; some examples, however, cannot be easily adapted to the shared session discipline (e.g., $*c?(x,y).x?(z).y?(z) \mid *c?(x,y).y?(z).x?(z)$ is typable in [31], where $x?(z)$ denotes input on x and $*c?(x,y)$ denotes replicated input) and their transcription, while possible, would be too far removed from the original term to be deemed a faithful representation. Recursive processes are known to be produce patterns that can be challenging to analyze using such order-based techniques. The work of [18,32] specializes Kobayashi's system to account for potentially unbounded process networks with non-trivial forms of sharing. Such systems are not typable in our work (see Section 6 for additional discussion on this topic).

The work of Padovani [37] develops techniques inspired by [29,31] to develop a typing system for deadlock (and lock) freedom for the linear π-calculus where (linear) channels must be used exactly once. By enforcing this form of linearity, the resulting system uses only one piece of ordering data per channel usage and can easily integrate a form of channel polymorphism that accounts for intricate cyclic interleavings of recursive processes. The combination of manifest sharing and linear session typing does not seem possible without the use of additional ordering data, and the lack of single-use linear channels make the robust channel polymorphism of [37] not feasible in our setting.

Dardha and Gay [13] recently integrated a system of Kobayashi-style orderings in a logical session π-calculus based on classical linear logic, extended with the ability to form *cyclic dependencies* of actions on *linear* session channels (Atkey et al. [1] study similar cycles but do not consider deadlock-freedom), without the need for new process constructs or an acquire-release discipline. Their work considers only a restricted form of replication common in linear logic-based works, not including recursive types nor recursive process definitions. This reduces the complexity of their system, at the cost of expressiveness. We also note that the cycles enabled by their system are produced by processes sharing multiple *linear* names. Since linearity is still enforced, they cannot represent the more general form of cycles that exploit shared channels, as we do.

A comparative study of session typing and Kobayashi-style systems in terms of sharing was developed by Dardha and Pérez [14], showing that such order-based techniques can account for sharing in ways that are out of reach of both classical session typing and pure logic-based session typing. Our system (and that of [13]) aims to combine the heightened power of Kobayashi-style systems with the benefits of session typing.

Progress and Session Typing To address limitations of classical binary session types, Honda et al. [24] introduced *multiparty* session types, where sessions are described by so-called global types that capture the interactions between an arbitrary number of session participants. Under some well-formedness constraints, global types can be used to ensure that a collection of processes correctly implements the global behavior in a deadlock-free way. However, these global type-based approaches do not ensure deadlock freedom in the presence of

higher-order channel passing or interleaved multiparty sessions. Coppo et al. [11] and Bettini et al. [5] develop systems that track usage orders among interleaved multiparty sessions, ruling out cyclic dependencies that can lead to deadlocks. The resulting system is quite intricate, since it combines the full multiparty session theory with the order tracking mechanism, interacts negatively with recursion (essentially disallowing interleaving with recursion) and, by tracking order at the multiparty session-level, ends up rejecting various benign configurations that can be accounted for by our more fine-grained analysis. We also highlight the analyses of Vieira and Vasconcelos [45] and Padovani et al. [38] that are more powerful than the approaches above, at the cost of a more complex analysis based on conversation types [9] (themselves a partial-order based technique).

Static Analysis of Concurrent Programs Lange et al. [34,35] develop a deadlock detection framework applied to the Go programming language. Their work distills CCS processes from programs which are then checked for deadlocks by a form of symbolic execution [34] and *model-checked* against modal μ-calculus formulae [35] which encode deadlock-freedom of the abstracted process (among other properties of interest). Their abstraction introduces some distance between the original program and the analysed process and so the analysis is sound only for certain restricted program fragments, excluding any combination of recursion and process spawning. Our direct approach does not suffer from this limitation.

de'Liguoro and Padovani [15] develop a typing discipline for deadlock-freedom in a setting where processes exchange messages via unordered mailboxes. Their calculus subsumes the actor model and their analysis combines both so-called mailbox types and specialized dependency graphs to track potential cycles between mailboxes in actor-based systems. The unordered nature of actor-based communication introduces significant differences wrt our work, which crucially exploits the ordering of exchanged messages.

8 Concluding Remarks

In this paper we have developed the concept of manifest deadlock-freedom in the context of the language $\mathsf{SILL}_{\mathsf{S}^+}$, a shared session-typed language, showcasing both the programming methodology and the expressiveness of our framework with a series of examples. Deadlock-freedom of well-typed programs is established by a novel abstraction of so-called green and red arrows to reason about the interdependencies between processes in terms of linear and shared channel references.

In future work, we plan to address some of the limitations of the interactions of deadlock-free shared sessions with recursion, by considering promising notions of world polymorphism and world communication. We also plan to study the problem of world inference and the inclusion of a linear forwarding construct.

References

1. Atkey, R., Lindley, S., Morris, J.G.: Conflation confers concurrency. In: et al., S.L. (ed.) Wadler Festschrift. pp. 32–55. Springer LNCS 9600 (2016)
2. Balzer, S., Pfenning, F.: Manifest sharing with session types. Proceedings of the ACM on Programming Languages (PACMPL) **1**(ICFP), 37:1–37:29 (2017)
3. Balzer, S., Pfenning, F., Toninho, B.: A universal session type for untyped asynchronous communication. In: 29th International Conference on Concurrency Theory (CONCUR). pp. 30:1–30:18. LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2018)
4. Benton, P.N.: A mixed linear and non-linear logic: Proofs, terms and models. In: 8th International Workshop on Computer Science Logic (CSL). Lecture Notes in Computer Science, vol. 933, pp. 121–135. Springer (1994), an extended version appeared as Technical Report UCAM-CL-TR-352, University of Cambridge
5. Bettini, L., Coppo, M., D'Antoni, L., Luca, M.D., Dezani-Ciancaglini, M., Yoshida, N.: Global progress in dynamically interleaved multiparty sessions. In: CONCUR 2008 - Concurrency Theory, 19th International Conference,. pp. 418–433 (2008)
6. Caires, L., Pérez, J.A., Pfenning, F., Toninho, B.: Logic-based domain-aware session types, unpublished draft
7. Caires, L., Pfenning, F.: Session types as intuitionistic linear propositions. In: 21st International Conference on Concurrency Theory (CONCUR). pp. 222–236. Springer (2010)
8. Caires, L., Pfenning, F., Toninho, B.: Linear logic propositions as session types. Mathematical Structures in Computer Science **26**(3), 367–423 (2016)
9. Caires, L., Vieira, H.T.: Conversation types. Theor. Comput. Sci. **411**(51-52), 4399–4440 (2010)
10. Cervesato, I., Scedrov, A.: Relating state-based and process-based concurrency through linear logic. Information and Computation **207**(10), 1044–1077 (2009)
11. Coppo, M., Dezani-Ciancaglini, M., Yoshida, N., Padovani, L.: Global progress for dynamically interleaved multiparty sessions. Mathematical Structures in Computer Science **26**(2), 238–302 (2016)
12. Crary, K., Harper, R., Puri, S.: What is a recursive module? In: ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI). pp. 50–63 (1999)
13. Dardha, O., Gay, S.J.: A new linear logic for deadlock-free session-typed processes. In: Foundations of Software Science and Computation Structures (FoSSaCS). pp. 91–109 (2018)
14. Dardha, O., Pérez, J.A.: Comparing deadlock-free session typed processes. In: EXPRESS/SOS. pp. 1–15 (2015)
15. de'Liguoro, U., Padovani, L.: Mailbox types for unordered interactions. In: 32nd European Conference on Object-Oriented Programming, ECOOP 2018. pp. 15:1–15:28 (2018)
16. Gay, S.J., Hole, M.: Subtyping for session types in the π-calculus. Acta Informatica **42**(2–3), 191–225 (2005)
17. Gay, S.J., Vasconcelos, V.T., Ravara, A., Gesbert, N., Caldeira, A.Z.: Modular session types for distributed object-orientedriented programming. In: 37th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL). pp. 299–312 (2010)

18. Giachino, E., Kobayashi, N., Laneve, C.: Deadlock analysis of unbounded process networks. In: CONCUR 2014 - Concurrency Theory - 25th International Conference, CONCUR 2014, Rome, Italy, September 2-5, 2014. Proceedings. pp. 63–77 (2014)
19. Gommerstadt, H., Jia, L., Pfenning, F.: Session-typed concurrent contracts. In: Ahmed, A. (ed.) European Symposium on Programming (ESOP'18). pp. 771–798. Springer LNCS 10801, Thessaloniki, Greece (Apr 2018)
20. Griffith, D.: Polarized Substructural Session Types. Ph.D. thesis, University of Illinois at Urbana-Champaign (2016)
21. Griffith, D., Pfenning, F.: SILL. `https://github.com/ISANobody/sill` (2015)
22. Honda, K.: Types for dyadic interaction. In: 4th International Conference on Concurrency Theory (CONCUR). pp. 509–523. Springer (1993)
23. Honda, K., Vasconcelos, V.T., Kubo, M.: Language primitives and type discipline for structured communication-based programming. In: 7th European Symposium on Programming (ESOP). pp. 122–138. Springer (1998)
24. Honda, K., Yoshida, N., Carbone, M.: Multiparty asynchronous session types. In: 35th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL). pp. 273–284. ACM (2008)
25. Igarashi, A., Kobayashi, N.: Type-based analysis of communication for concurrent programming languages. In: Static Analysis, 4th International Symposium, SAS '97. pp. 187–201 (1997)
26. Igarashi, A., Kobayashi, N.: A generic type system for the pi-calculus. In: Conference Record of POPL 2001: The 28th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages. pp. 128–141 (2001)
27. Igarashi, A., Kobayashi, N.: A generic type system for the pi-calculus. Theor. Comput. Sci. **311**(1-3), 121–163 (2004)
28. Kobayashi, N.: A partially deadlock-free typed process calculus. In: Proceedings, 12th Annual IEEE Symposium on Logic in Computer Science. pp. 128–139 (1997)
29. Kobayashi, N.: A type system for lock-free processes. Inf. Comput. **177**(2), 122–159 (2002)
30. Kobayashi, N.: Type-based information flow analysis for the pi-calculus. Acta Inf. **42**(4-5), 291–347 (2005)
31. Kobayashi, N.: A new type system for deadlock-free processes. In: International Conference on Concurrency Theory (CONCUR). pp. 233–247 (2006)
32. Kobayashi, N., Laneve, C.: Deadlock analysis of unbounded process networks. Information and Computation **252**, 48–70 (2017)
33. Kobayashi, N., Saito, S., Sumii, E.: An implicitly-typed deadlock-free process calculus. In: CONCUR 2000 - Concurrency Theory, 11th International Conference, University Park, PA, USA, August 22-25, 2000, Proceedings. pp. 489–503 (2000)
34. Lange, J., Ng, N., Toninho, B., Yoshida, N.: Fencing off go: Liveness and safety for channel-based programming. In: 44th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL). pp. 748–761. ACM (2017)
35. Lange, J., Ng, N., Toninho, B., Yoshida, N.: A static verification framework for message passing in go using behavioural types. In: Proceedings of the 40th International Conference on Software Engineering, ICSE 2018, Gothenburg, Sweden, May 27 - June 03, 2018. pp. 1137–1148 (2018)
36. Milner, R.: A Calculus of Communicating Systems, Lecture Notes in Computer Science, vol. 92. Springer (1980)
37. Padovani, L.: Deadlock and lock freedom in the linear π-calculus. In: Computer Science Logic – Logic in Computer Science (CSL-LICS). pp. 72:1–72:10 (2014)

38. Padovani, L., Vasconcelos, V.T., Vieira, H.T.: Typing liveness in multiparty communicating systems. In: Coordination Models and Languages - 16th IFIP WG 6.1 International Conference, COORDINATION 2014. pp. 147–162 (2014)
39. Pérez, J.A., Caires, L., Pfenning, F., Toninho, B.: Linear logical relations and observational equivalences for session-based concurrency. Information and Computation **239**, 254–302 (2014)
40. Pfenning, F., Griffith, D.: Polarized substructural session types. In: 18th International Conference on Foundations of Software Science and Computation Structures (FoSSaCS). pp. 3–22. Springer (2015)
41. Reed, J.: A judgmental deconstruction of modal logic (January 2009), `http://www.cs.cmu.edu/~jcreed/papers/jdml.pdf`, unpublished manuscript
42. Toninho, B.: A Logical Foundation for Session-based Concurrent Computation. Ph.D. thesis, Carnegie Mellon University and New University of Lisbon (2015)
43. Toninho, B., Caires, L., Pfenning, F.: Higher-order processes, functions, and sessions: a monadic integration. In: 22nd European Symposium on Programming (ESOP). pp. 350–369. Springer (2013)
44. Vasconcelos, V.T.: Fundamentals of session types. Inf. Comput. **217**, 52–70 (2012)
45. Vieira, H.T., Vasconcelos, V.T.: Typing progress in communication-centred systems. In: Coordination Models and Languages, 15th International Conference, COORDINATION 2013, Held as Part of the 8th International Federated Conference on Distributed Computing Techniques, DisCoTec 2013, Florence, Italy, June 3-5, 2013. Proceedings. pp. 236–250 (2013)
46. Wadler, P.: Propositions as sessions. In: 17th ACM SIGPLAN International Conference on Functional Programming (ICFP). pp. 273–286. ACM (2012)

A Shared Queue Implementation: *empty* process

$$
\begin{array}{l}
\mathit{empty} : \{\delta_1 < \delta_2, \delta_2 < \delta_3, \delta_3 < \delta_4 \vdash \mathsf{list}[\delta_1 \updownarrow_{\delta_2}^{\delta_2}] A_{\mathsf{S}}[\delta_3 \updownarrow_{\delta_4}^{\delta_4}]\} \\
c[\delta_1 \updownarrow_{\delta_2}^{\delta_2}][\delta_3 \updownarrow_{\delta_4}^{\delta_4}] \leftarrow \mathit{empty} = \\
c' \leftarrow \mathsf{accept}\ c\ ; \\
\mathsf{case}\ c'\ \mathsf{of} \\
\mid \mathsf{ins} \rightarrow x \leftarrow \mathsf{recv}\ c'\ ; \\
\qquad e : \mathsf{list}[\delta_1 \updownarrow_{\delta_2}^{\delta_2}] A_{\mathsf{S}}[\delta_3 \updownarrow_{\delta_4}^{\delta_4}] \leftarrow \mathsf{empty}\ ; \\
\qquad \mathsf{send}\ c'\ e\ ; \\
\qquad c \leftarrow \mathsf{detach}\ c'\ ; \\
\qquad c'' : \mathsf{list}[\delta_1 \updownarrow_{\delta_2}^{\delta_2}] A_{\mathsf{S}}[\delta_3 \updownarrow_{\delta_4}^{\delta_4}] \leftarrow \mathit{elem} \leftarrow x,\ e\ ;\ \mathsf{fwd}\ c\ c'' \\
\mid \mathsf{del} \rightarrow c'.\mathsf{none}\ ; \\
\qquad c \leftarrow \mathsf{detach}\ c'\ ; \\
\qquad c'' : \mathsf{list}[\delta_1 \updownarrow_{\delta_2}^{\delta_2}] A_{\mathsf{S}}[\delta_3 \updownarrow_{\delta_4}^{\delta_4}] \leftarrow \mathit{empty}\ ;\ \mathsf{fwd}\ c\ c''
\end{array}
$$

Fig. 12: Shared Queue Implementation – *empty* process.

B Statics

B.1 Abstract Syntax

Session Types		Abstract Form	Remarks
A_m, B_m, C_m, D_m	$\triangleq$	$\oplus\{\overline{l : A_{\mathsf{L}}}\}$	internal choice, at least one label
		$\&\{\overline{l : A_{\mathsf{L}}}\}$	external choice, at least one label
		$A_{\mathsf{L}} @ \omega_o \updownarrow_{\omega_t}^{\omega_v} \otimes B_{\mathsf{L}}$	linear channel output
		$A_{\mathsf{L}} @ \omega_o \updownarrow_{\omega_t}^{\omega_v} \multimap B_{\mathsf{L}}$	linear channel input
		$\exists x{:}A_{\mathsf{S}} @ \omega_o \updownarrow_{\omega_t}^{\omega_v} . B_{\mathsf{L}}$	shared channel output
		$\Pi x{:}A_{\mathsf{S}} @ \omega_o \updownarrow_{\omega_t}^{\omega_v} . B_{\mathsf{L}}$	shared channel input
		$\mathbf{1}_{\mathsf{L}}$	termination
		$\downarrow_{\mathsf{L}}^{\mathsf{S}} A_{\mathsf{S}}$	downshift
		$\uparrow_{\mathsf{L}}^{\mathsf{S}} A_{\mathsf{L}}$	upshift
		Y_m	type variable

B.2 Process Typing

$$
\frac{\begin{array}{c}
\Delta_1 = \overline{y_{\mathsf{L}} : B_{\mathsf{L}}[\omega_m \updownarrow_{\omega_u}^{\omega_v}]} \qquad \Phi_1 = \overline{\tilde{y}_{\mathsf{L}} : \tilde{B}_{\mathsf{L}}[\tilde{\omega}_m \updownarrow_{\tilde{\omega}_u}^{\tilde{\omega}_v}]} \qquad \Gamma_1 = \overline{z_{\mathsf{S}} : C_{\mathsf{S}}[\omega_l \updownarrow_{\omega_p}^{\omega_r}]} \\
(\Psi' \vdash x'_{\mathsf{L}} : A'_{\mathsf{L}}[\delta_j \updownarrow_{\delta_k}^{\delta_n}] \leftarrow X_{\mathsf{L}} \leftarrow \Delta', \Phi', \Gamma' = P_{x'_{\mathsf{L}}, \mathsf{dom}(\Delta'), \mathsf{dom}(\Phi'), \mathsf{dom}(\Gamma'), \Psi''}) \in \Sigma \\
\hat{\gamma}(A'_{\mathsf{L}}[\delta_j \updownarrow_{\delta_k}^{\delta_n}]) = A_{\mathsf{L}}[\omega_j \updownarrow_{\omega_k}^{\omega_n}] \qquad \hat{\gamma}(\Delta') = \Delta_1 \qquad \hat{\gamma}(\Phi') = \Phi_1 \qquad \hat{\gamma}(\Gamma') = \Gamma_1 \qquad \Psi \vdash \hat{\gamma}(\Psi') \\
\Psi^+ \vdash \omega_t < \omega_k \\
\Psi;\ \Gamma_1, \Gamma_2;\ \Phi_2;\ \Delta_2, x_{\mathsf{L}} : A_{\mathsf{L}}[\omega_j \updownarrow_{\omega_k}^{\omega_n}] \vdash Q_{x_{\mathsf{L}}} :: (z''_{\mathsf{L}} : D_{\mathsf{L}}[\omega_i \updownarrow_{\omega_q}^{\omega_t}])
\end{array}}{\Psi;\ \Gamma_1, \Gamma_2;\ \Phi_1, \Phi_2;\ \Delta_1, \Delta_2 \vdash x_{\mathsf{L}} : A_{\mathsf{L}}[\omega_j \updownarrow_{\omega_k}^{\omega_n}] \leftarrow X_{\mathsf{L}} \leftarrow \overline{y_{\mathsf{L}}}, \overline{\tilde{y}_{\mathsf{L}}}, \overline{z_{\mathsf{S}}}\ ;\ Q_{x_{\mathsf{L}}} :: (z''_{\mathsf{L}} : D_{\mathsf{L}}[\omega_i \updownarrow_{\omega_q}^{\omega_t}])}\ (\text{T-SPAWN}_{\mathsf{LL}})
$$

$$\frac{\Gamma_1 = \overline{z_\mathsf{S} : C_\mathsf{S}[\omega_l \updownarrow^{\omega_r}_{\omega_p}]} \qquad (\Psi' \vdash x'_\mathsf{S} : A'_\mathsf{S}[\delta_j \updownarrow^{\delta_n}_{\delta_k}] \leftarrow X_\mathsf{S} \leftarrow \Gamma' = P_{x'_\mathsf{S},\mathsf{dom}(\Gamma'),\Psi''}) \in \Sigma \quad \hat{\gamma}(A'_\mathsf{S}[\delta_j \updownarrow^{\delta_n}_{\delta_k}]) = A_\mathsf{S}[\omega_j \updownarrow^{\omega_n}_{\omega_k}] \qquad \hat{\gamma}(\Gamma') = \Gamma_1 \qquad \Psi \vdash \hat{\gamma}(\Psi') \quad \Psi;\, \Gamma_1, \Gamma_2, x_\mathsf{S} : A_\mathsf{S}[\omega_j \updownarrow^{\omega_n}_{\omega_k}];\, \Phi;\, \Delta \vdash Q_{x_\mathsf{S}} :: (z''_\mathsf{L} : D_\mathsf{L}[\omega_i \updownarrow^{\omega_t}_{\omega_q}])}{\Psi;\, \Gamma_1, \Gamma_2;\, \Phi;\, \Delta \vdash x_\mathsf{S} : A_\mathsf{S}[\omega_j \updownarrow^{\omega_n}_{\omega_k}] \leftarrow X_\mathsf{S} \leftarrow \overline{z_\mathsf{S}};\, Q_{x_\mathsf{S}} :: (z''_\mathsf{L} : D_\mathsf{L}[\omega_i \updownarrow^{\omega_t}_{\omega_q}])} \text{ (T-SpawnLS)}$$

$$\frac{\Gamma_1 = \overline{z_\mathsf{S} : C_\mathsf{S}[\omega_l \updownarrow^{\omega_r}_{\omega_p}]} \qquad (\Psi' \vdash x'_\mathsf{S} : A'_\mathsf{S}[\delta_j \updownarrow^{\delta_n}_{\delta_k}] \leftarrow X_\mathsf{S} \leftarrow \Gamma' = P_{x'_\mathsf{S},\mathsf{dom}(\Gamma'),\Psi''}) \in \Sigma \quad \hat{\gamma}(A'_\mathsf{S}[\delta_j \updownarrow^{\delta_n}_{\delta_k}]) = A_\mathsf{S}[\omega_j \updownarrow^{\omega_n}_{\omega_k}] \qquad \hat{\gamma}(\Gamma') = \Gamma_1 \qquad \Psi \vdash \hat{\gamma}(\Psi') \quad \Psi;\, \Gamma_1, \Gamma_2, x_\mathsf{S} : A_\mathsf{S}[\omega_j \updownarrow^{\omega_n}_{\omega_k}] \vdash Q_{x_\mathsf{S}} :: (z''_\mathsf{S} : D_\mathsf{S}[\omega_i \updownarrow^{\omega_t}_{\omega_q}])}{\Psi;\, \Gamma_1, \Gamma_2 \vdash x_\mathsf{S} : A_\mathsf{S}[\omega_j \updownarrow^{\omega_n}_{\omega_k}] \leftarrow X_\mathsf{S} \leftarrow \overline{z_\mathsf{S}};\, Q_{x_\mathsf{S}} :: (z''_\mathsf{S} : D_\mathsf{S}[\omega_i \updownarrow^{\omega_t}_{\omega_q}])} \text{ (T-SpawnSS)}$$

$$\frac{\Delta = \overline{y_\mathsf{L} : B_\mathsf{L}[\omega_m \updownarrow^{\omega_v}_{\omega_u}]} \qquad \Phi = \overline{\tilde{y}_\mathsf{L} : \tilde{B}_\mathsf{L}[\tilde{\omega}_m \updownarrow^{\tilde{\omega}_v}_{\tilde{\omega}_u}]} \qquad \Gamma_1 = \overline{z_\mathsf{S} : C_\mathsf{S}[\omega_l \updownarrow^{\omega_r}_{\omega_p}]} \quad (\Psi' \vdash x'_\mathsf{L} : A'_\mathsf{L}[\delta_j \updownarrow^{\delta_n}_{\delta_k}] \leftarrow X_\mathsf{L} \leftarrow \Delta', \Phi', \Gamma' = P_{x'_\mathsf{L},\mathsf{dom}(\Delta'),\mathsf{dom}(\Phi'),\mathsf{dom}(\Gamma'),\Psi''}) \in \Sigma \quad \hat{\gamma}(A'_\mathsf{L}[\delta_j \updownarrow^{\delta_n}_{\delta_k}]) = A_\mathsf{L}[\omega_j \updownarrow^{\omega_n}_{\omega_k}] \qquad \hat{\gamma}(\Delta') = \Delta \qquad \hat{\gamma}(\Phi') = \Phi \qquad \hat{\gamma}(\Gamma') = \Gamma_1 \qquad \Psi \vdash \hat{\gamma}(\Psi')}{\Psi;\, \Gamma_1, \Gamma_2;\, \Phi;\, \Delta \vdash x_\mathsf{L} \leftarrow X_\mathsf{L} \leftarrow \overline{y_\mathsf{L}}, \overline{\tilde{y}_\mathsf{L}}, \overline{z_\mathsf{S}} :: (x_\mathsf{L} : A_\mathsf{L}[\omega_j \updownarrow^{\omega_n}_{\omega_k}])} \text{ (T-TailL)}$$

$$\frac{}{\Psi;\, \Gamma, y_\mathsf{S} : A_\mathsf{S}[\omega_j \updownarrow^{\omega_n}_{\omega_k}] \vdash \mathsf{fwd}\ x_\mathsf{S}\ y_\mathsf{S} :: (x_\mathsf{S} : A_\mathsf{S}[\omega_j \updownarrow^{\omega_n}_{\omega_k}])} \text{ (T-IdS)}$$

$$\frac{\Psi, \mathsf{w};\, \Gamma;\, \Phi;\, \Delta \vdash Q_\mathsf{w} :: (x_\mathsf{L} : A_\mathsf{L}[\omega_m \updownarrow^{\omega_v}_{\omega_u}])}{\Psi;\, \Gamma;\, \Phi;\, \Delta \vdash \mathsf{w} \leftarrow \mathsf{new\_world};\, Q_\mathsf{w} :: (x_\mathsf{L} : A_\mathsf{L}[\omega_m \updownarrow^{\omega_v}_{\omega_u}])} \text{ (T-NewL)}$$

$$\frac{\Psi, \mathsf{w};\, \Gamma \vdash Q_\mathsf{w} :: (x_\mathsf{S} : A_\mathsf{S}[\omega_m \updownarrow^{\omega_v}_{\omega_u}])}{\Psi;\, \Gamma \vdash \mathsf{w} \leftarrow \mathsf{new\_world};\, Q_\mathsf{w} :: (x_\mathsf{S} : A_\mathsf{S}[\omega_m \updownarrow^{\omega_v}_{\omega_u}])} \text{ (T-NewS)}$$

$$\frac{\omega_p, \omega_r \in \Psi \qquad (\Psi, \omega_p < \omega_r)^+ \text{ irreflexive} \qquad \Psi, \omega_p < \omega_r;\, \Gamma;\, \Phi;\, \Delta \vdash Q :: (x_\mathsf{L} : A_\mathsf{L}[\omega_m \updownarrow^{\omega_v}_{\omega_u}])}{\Psi;\, \Gamma;\, \Phi;\, \Delta \vdash \omega_p < \omega_r;\, Q :: (x_\mathsf{L} : A_\mathsf{L}[\omega_m \updownarrow^{\omega_v}_{\omega_u}])} \text{ (T-OrdL)}$$

$$\frac{\omega_p, \omega_r \in \Psi \qquad (\Psi, \omega_p < \omega_r)^+ \text{ irreflexive} \qquad \Psi, \omega_p < \omega_r;\, \Gamma \vdash Q :: (x_\mathsf{S} : A_\mathsf{S}[\omega_m \updownarrow^{\omega_v}_{\omega_u}])}{\Psi;\, \Gamma \vdash \omega_p < \omega_r;\, Q :: (x_\mathsf{S} : A_\mathsf{S}[\omega_m \updownarrow^{\omega_v}_{\omega_u}])} \text{ (T-OrdS)}$$

$$\frac{\Psi^* \vdash \omega_k \leq \omega_m \leq \omega_n \qquad \Psi^+ \vdash \omega_n < \omega_u \qquad \forall y_\mathsf{L} : B_\mathsf{L}[\omega_l \updownarrow^{\omega_r}_{\omega_p}] \in \Phi : \omega_l < \omega_m \quad \Psi;\, \Gamma, x_\mathsf{S} : \uparrow^\mathsf{S}_\mathsf{L} A_\mathsf{L}[\omega_m \updownarrow^{\omega_v}_{\omega_u}];\, \Phi, x_\mathsf{L} : A_\mathsf{L}[\omega_m \updownarrow^{\omega_v}_{\omega_u}];\, \Delta \vdash Q_{x_\mathsf{L}} :: (z_\mathsf{L} : C_\mathsf{L}[\omega_j \updownarrow^{\omega_n}_{\omega_k}])}{\Psi;\, \Gamma, x_\mathsf{S} : \uparrow^\mathsf{S}_\mathsf{L} A_\mathsf{L}[\omega_m \updownarrow^{\omega_v}_{\omega_u}];\, \Phi;\, \Delta \vdash x_\mathsf{L} \leftarrow \mathsf{acquire}\, x_\mathsf{S}\, ;\, Q_{x_\mathsf{L}} :: (z_\mathsf{L} : C_\mathsf{L}[\omega_j \updownarrow^{\omega_n}_{\omega_k}])} \text{ (T-}\uparrow^\mathsf{S}_\mathsf{L}\text{L)}$$

$$\frac{\Psi;\, \Gamma;\, \cdot\, ;\, \cdot \vdash P_{x_\mathsf{L}} :: (x_\mathsf{L} : A_\mathsf{L}[\omega_m \updownarrow^{\omega_v}_{\omega_u}])}{\Psi;\, \Gamma \vdash x_\mathsf{L} \leftarrow \mathsf{accept}\, x_\mathsf{S}\, ;\, P_{x_\mathsf{L}} :: (x_\mathsf{S} : \uparrow^\mathsf{S}_\mathsf{L} A_\mathsf{L}[\omega_m \updownarrow^{\omega_v}_{\omega_u}])} \text{ (T-}\uparrow^\mathsf{S}_\mathsf{L}\text{R)}$$

$$\frac{\Psi;\ \Gamma, x_{\mathsf{S}} : A_{\mathsf{S}}[\omega_m \updownarrow_{\omega_u}^{\omega_v}];\ \Phi;\ \Delta \vdash Q_{x_{\mathsf{S}}} :: (z_{\mathsf{L}} : C_{\mathsf{L}}[\omega_j \updownarrow_{\omega_k}^{\omega_n}])}{\Psi;\ \Gamma;\ \Phi, x_{\mathsf{L}} : \downarrow_{\mathsf{L}}^{\mathsf{S}} A_{\mathsf{S}}[\omega_m \updownarrow_{\omega_u}^{\omega_v}];\ \Delta \vdash x_{\mathsf{S}} \leftarrow \mathsf{release}\, x_{\mathsf{L}}\, ;\, Q_{x_{\mathsf{S}}} :: (z_{\mathsf{L}} : C_{\mathsf{L}}[\omega_j \updownarrow_{\omega_k}^{\omega_n}])}\ (\text{T-}{\downarrow_{\mathrm{L}}^{\mathsf{S}}}\mathrm{L})$$

$$\frac{\Psi;\ \Gamma \vdash P_{x_{\mathsf{S}}} :: (x_{\mathsf{S}} : A_{\mathsf{S}}[\omega_m \updownarrow_{\omega_u}^{\omega_v}])}{\Psi;\ \Gamma;\ \cdot\,;\ \cdot \vdash x_{\mathsf{S}} \leftarrow \mathsf{detach}\, x_{\mathsf{L}}\, ;\, P_{x_{\mathsf{S}}} :: (x_{\mathsf{L}} : \downarrow_{\mathsf{L}}^{\mathsf{S}} A_{\mathsf{S}}[\omega_m \updownarrow_{\omega_u}^{\omega_v}])}\ (\text{T-}{\downarrow_{\mathrm{L}}^{\mathsf{S}}}\mathrm{R})$$

$$\frac{\Psi;\ \Gamma;\ \Phi;\ \Delta \vdash Q :: (z_{\mathsf{L}} : C_{\mathsf{L}}[\omega_j \updownarrow_{\omega_k}^{\omega_n}])}{\Psi;\ \Gamma;\ \Phi;\ \Delta, x_{\mathsf{L}} : \mathbf{1}[\omega_m \updownarrow_{\omega_u}^{\omega_v}] \vdash \mathsf{wait}\, x_{\mathsf{L}}\, ;\, Q :: (z_{\mathsf{L}} : C_{\mathsf{L}}[\omega_j \updownarrow_{\omega_k}^{\omega_n}])}\ (\text{T-}\mathbf{1}_{\mathrm{L}})$$

$$\frac{}{\Psi;\ \Gamma;\ \cdot\,;\ \cdot \vdash \mathsf{close}\, x_{\mathsf{L}} :: (x_{\mathsf{L}} : \mathbf{1}[\omega_m \updownarrow_{\omega_u}^{\omega_v}])}\ (\text{T-}\mathbf{1}_{\mathrm{R}})$$

$$\frac{\Psi;\ \Gamma;\Phi;\ \Delta, x_{\mathsf{L}} : B_{\mathsf{L}}[\omega_m \updownarrow_{\omega_u}^{\omega_v}], y_{\mathsf{L}} : A_{\mathsf{L}}[\omega_l \updownarrow_{\omega_p}^{\omega_r}] \vdash Q_{y_{\mathsf{L}}} :: (z_{\mathsf{L}} : C_{\mathsf{L}}[\omega_j \updownarrow_{\omega_k}^{\omega_n}])}{\Psi;\ \Gamma;\Phi;\ \Delta, x_{\mathsf{L}} : A_{\mathsf{L}}@\omega_l \updownarrow_{\omega_p}^{\omega_r} \otimes B_{\mathsf{L}}[\omega_m \updownarrow_{\omega_u}^{\omega_v}] \vdash y_{\mathsf{L}} \leftarrow \mathsf{recv}\, x_{\mathsf{L}}\, ;\, Q_{y_{\mathsf{L}}} :: (z_{\mathsf{L}} : C_{\mathsf{L}}[\omega_j \updownarrow_{\omega_k}^{\omega_n}])}\ (\text{T-}{\otimes_{\mathrm{L}_1}})$$

$$\frac{\Psi;\ \Gamma;\Phi, x_{\mathsf{L}} : B_{\mathsf{L}}[\omega_m \updownarrow_{\omega_u}^{\omega_v}];\ \Delta, y_{\mathsf{L}} : A_{\mathsf{L}}[\omega_l \updownarrow_{\omega_p}^{\omega_r}] \vdash Q_{y_{\mathsf{L}}} :: (z_{\mathsf{L}} : C_{\mathsf{L}}[\omega_j \updownarrow_{\omega_k}^{\omega_n}])}{\Psi;\ \Gamma;\Phi, x_{\mathsf{L}} : A_{\mathsf{L}}@\omega_l \updownarrow_{\omega_p}^{\omega_r} \otimes B_{\mathsf{L}}[\omega_m \updownarrow_{\omega_u}^{\omega_v}];\ \Delta \vdash y_{\mathsf{L}} \leftarrow \mathsf{recv}\, x_{\mathsf{L}}\, ;\, Q_{y_{\mathsf{L}}} :: (z_{\mathsf{L}} : C_{\mathsf{L}}[\omega_j \updownarrow_{\omega_k}^{\omega_n}])}\ (\text{T-}{\otimes_{\mathrm{L}_2}})$$

$$\frac{\Psi;\ \Gamma;\cdot\,;\ \Delta \vdash P :: (x_{\mathsf{L}} : B_{\mathsf{L}}[\omega_m \updownarrow_{\omega_u}^{\omega_v}])}{\Psi;\ \Gamma;\cdot\,;\ \Delta, y_{\mathsf{L}} : A_{\mathsf{L}}[\omega_l \updownarrow_{\omega_p}^{\omega_r}] \vdash \mathsf{send}\, x_{\mathsf{L}}\, y_{\mathsf{L}}\, ;\, P :: (x_{\mathsf{L}} : A_{\mathsf{L}}@\omega_l \updownarrow_{\omega_p}^{\omega_r} \otimes B_{\mathsf{L}}[\omega_m \updownarrow_{\omega_u}^{\omega_v}])}\ (\text{T-}{\otimes_{\mathrm{R}}})$$

$$\frac{\Psi;\ \Gamma;\ \Phi;\ \Delta, x_{\mathsf{L}} : B_{\mathsf{L}}[\omega_m \updownarrow_{\omega_u}^{\omega_v}] \vdash Q :: (z_{\mathsf{L}} : C_{\mathsf{L}}[\omega_j \updownarrow_{\omega_k}^{\omega_n}])}{\Psi;\ \Gamma;\ \Phi;\ \Delta, x_{\mathsf{L}} : A_{\mathsf{L}}@\omega_l \updownarrow_{\omega_p}^{\omega_r} \multimap B_{\mathsf{L}}[\omega_m \updownarrow_{\omega_u}^{\omega_v}], y_{\mathsf{L}} : A_{\mathsf{L}}[\omega_l \updownarrow_{\omega_p}^{\omega_r}] \vdash \mathsf{send}\, x_{\mathsf{L}}\, y_{\mathsf{L}}\, ;\, Q :: (z_{\mathsf{L}} : C_{\mathsf{L}}[\omega_j \updownarrow_{\omega_k}^{\omega_n}])}\ (\text{T-}{\multimap_{\mathrm{L}_1}})$$

$$\frac{\Psi;\ \Gamma;\ \Phi, x_{\mathsf{L}} : B_{\mathsf{L}}[\omega_m \updownarrow_{\omega_u}^{\omega_v}];\ \Delta \vdash Q :: (z_{\mathsf{L}} : C_{\mathsf{L}}[\omega_j \updownarrow_{\omega_k}^{\omega_n}])}{\Psi;\ \Gamma;\ \Phi, x_{\mathsf{L}} : A_{\mathsf{L}}@\omega_l \updownarrow_{\omega_p}^{\omega_r} \multimap B_{\mathsf{L}}[\omega_m \updownarrow_{\omega_u}^{\omega_v}];\ \Delta, y_{\mathsf{L}} : A_{\mathsf{L}}[\omega_l \updownarrow_{\omega_p}^{\omega_r}] \vdash \mathsf{send}\, x_{\mathsf{L}}\, y_{\mathsf{L}}\, ;\, Q :: (z_{\mathsf{L}} : C_{\mathsf{L}}[\omega_j \updownarrow_{\omega_k}^{\omega_n}])}\ (\text{T-}{\multimap_{\mathrm{L}_2}})$$

$$\frac{\Psi;\ \Gamma;\ \cdot\,;\ \Delta, y_{\mathsf{L}} : A_{\mathsf{L}}[\omega_l \updownarrow_{\omega_p}^{\omega_r}] \vdash P_{y_{\mathsf{L}}} :: (x_{\mathsf{L}} : B_{\mathsf{L}}[\omega_m \updownarrow_{\omega_u}^{\omega_v}])}{\Psi;\ \Gamma;\ \cdot\,;\ \Delta \vdash y_{\mathsf{L}} \leftarrow \mathsf{recv}\, x_{\mathsf{L}}\, ;\, P_{y_{\mathsf{L}}} :: (x_{\mathsf{L}} : A_{\mathsf{L}}@\omega_l \updownarrow_{\omega_p}^{\omega_r} \multimap B_{\mathsf{L}}[\omega_m \updownarrow_{\omega_u}^{\omega_v}])}\ (\text{T-}{\multimap_{\mathrm{R}}})$$

$$\frac{\Psi;\ \Gamma, y_{\mathsf{S}} : A_{\mathsf{S}}[\omega_l \updownarrow_{\omega_p}^{\omega_r}];\Phi;\ \Delta, x_{\mathsf{L}} : B_{\mathsf{L}}[\omega_m \updownarrow_{\omega_u}^{\omega_v}] \vdash Q_{y_{\mathsf{S}}} :: (z_{\mathsf{L}} : C_{\mathsf{L}}[\omega_j \updownarrow_{\omega_k}^{\omega_n}])}{\Psi;\ \Gamma;\Phi;\ \Delta, x_{\mathsf{L}} : \exists x{:}A_{\mathsf{S}}@\omega_l \updownarrow_{\omega_p}^{\omega_r}\, .\, B_{\mathsf{L}}[\omega_m \updownarrow_{\omega_u}^{\omega_v}] \vdash y_{\mathsf{S}} \leftarrow \mathsf{recv}\, x_{\mathsf{L}}\, ;\, Q_{y_{\mathsf{S}}} :: (z_{\mathsf{L}} : C_{\mathsf{L}}[\omega_j \updownarrow_{\omega_k}^{\omega_n}])}\ (\text{T-}{\exists_{\mathrm{L}_1}})$$

$$\frac{\Psi;\ \Gamma, y_{\mathsf{S}} : A_{\mathsf{S}}[\omega_l \updownarrow_{\omega_p}^{\omega_r}];\Phi, x_{\mathsf{L}} : B_{\mathsf{L}}[\omega_m \updownarrow_{\omega_u}^{\omega_v}];\ \Delta \vdash Q_{y_{\mathsf{S}}} :: (z_{\mathsf{L}} : C_{\mathsf{L}}[\omega_j \updownarrow_{\omega_k}^{\omega_n}])}{\Psi;\ \Gamma;\Phi, x_{\mathsf{L}} : \exists x{:}A_{\mathsf{S}}@\omega_l \updownarrow_{\omega_p}^{\omega_r}\, .\, B_{\mathsf{L}}[\omega_m \updownarrow_{\omega_u}^{\omega_v}];\ \Delta \vdash y_{\mathsf{S}} \leftarrow \mathsf{recv}\, x_{\mathsf{L}}\, ;\, Q_{y_{\mathsf{S}}} :: (z_{\mathsf{L}} : C_{\mathsf{L}}[\omega_j \updownarrow_{\omega_k}^{\omega_n}])}\ (\text{T-}{\exists_{\mathrm{L}_2}})$$

$$\frac{\Psi;\ \Gamma, y_{\mathsf{S}} : A_{\mathsf{S}}[\omega_l \updownarrow_{\omega_p}^{\omega_r}];\cdot\,;\ \Delta \vdash P :: (x_{\mathsf{L}} : B_{\mathsf{L}}[\omega_m \updownarrow_{\omega_u}^{\omega_v}])}{\Psi;\ \Gamma, y_{\mathsf{S}} : A_{\mathsf{S}}[\omega_l \updownarrow_{\omega_p}^{\omega_r}];\cdot\,;\ \Delta \vdash \mathsf{send}\, x_{\mathsf{L}}\, y_{\mathsf{S}}\, ;\, P :: (x_{\mathsf{L}} : \exists x{:}A_{\mathsf{S}}@\omega_l \updownarrow_{\omega_p}^{\omega_r}\, .\, B_{\mathsf{L}}[\omega_m \updownarrow_{\omega_u}^{\omega_v}])}\ (\text{T-}{\exists_{\mathrm{R}}})$$

$$\frac{\Psi;\ \Gamma, y_{\mathsf{S}} : A_{\mathsf{S}}[\omega_l \updownarrow_{\omega_p}^{\omega_r}];\ \Phi;\ \Delta, x_{\mathsf{L}} : B_{\mathsf{L}}[\omega_m \updownarrow_{\omega_u}^{\omega_v}] \vdash Q :: (z_{\mathsf{L}} : C_{\mathsf{L}}[\omega_j \updownarrow_{\omega_k}^{\omega_n}])}{\Psi;\ \Gamma, y_{\mathsf{S}} : A_{\mathsf{S}}[\omega_l \updownarrow_{\omega_p}^{\omega_r}];\ \Phi;\ \Delta, x_{\mathsf{L}} : \Pi x{:}A_{\mathsf{S}}@\omega_l \updownarrow_{\omega_p}^{\omega_r} . B_{\mathsf{L}}[\omega_m \updownarrow_{\omega_u}^{\omega_v}] \vdash \mathsf{send}\ x_{\mathsf{L}}\ y_{\mathsf{S}}\,; Q :: (z_{\mathsf{L}} : C_{\mathsf{L}}[\omega_j \updownarrow_{\omega_k}^{\omega_n}])}\ (\text{T-}\Pi_{\mathrm{L}_1})$$

$$\frac{\Psi;\ \Gamma, y_{\mathsf{S}} : A_{\mathsf{S}}[\omega_l \updownarrow_{\omega_p}^{\omega_r}];\ \Phi, x_{\mathsf{L}} : B_{\mathsf{L}}[\omega_m \updownarrow_{\omega_u}^{\omega_v}];\ \Delta \vdash Q :: (z_{\mathsf{L}} : C_{\mathsf{L}}[\omega_j \updownarrow_{\omega_k}^{\omega_n}])}{\Psi;\ \Gamma, y_{\mathsf{S}} : A_{\mathsf{S}}[\omega_l \updownarrow_{\omega_p}^{\omega_r}];\ \Phi, x_{\mathsf{L}} : \Pi x{:}A_{\mathsf{S}}@\omega_l \updownarrow_{\omega_p}^{\omega_r} . B_{\mathsf{L}}[\omega_m \updownarrow_{\omega_u}^{\omega_v}];\ \Delta \vdash \mathsf{send}\ x_{\mathsf{L}}\ y_{\mathsf{S}}\,; Q :: (z_{\mathsf{L}} : C_{\mathsf{L}}[\omega_j \updownarrow_{\omega_k}^{\omega_n}])}\ (\text{T-}\Pi_{\mathrm{L}_2})$$

$$\frac{\Psi;\ \Gamma, y_{\mathsf{S}} : A_{\mathsf{S}}[\omega_l \updownarrow_{\omega_p}^{\omega_r}];\ \cdot\,;\ \Delta \vdash P_{y_{\mathsf{S}}} :: (x_{\mathsf{L}} : B_{\mathsf{L}}[\omega_m \updownarrow_{\omega_u}^{\omega_v}])}{\Psi;\ \Gamma;\ \cdot\,;\ \Delta \vdash y_{\mathsf{S}} \leftarrow \mathsf{recv}\ x_{\mathsf{L}}\,; P_{y_{\mathsf{S}}} :: (x_{\mathsf{L}} : \Pi x{:}A_{\mathsf{S}}@\omega_l \updownarrow_{\omega_p}^{\omega_r} . B_{\mathsf{L}}[\omega_m \updownarrow_{\omega_u}^{\omega_v}])}\ (\text{T-}\Pi_{\mathrm{R}})$$

$$\frac{(\forall i)\ \ \Psi;\ \Gamma;\ \Phi;\ \Delta, x_{\mathsf{L}} : A_{\mathsf{L}_i}[\omega_m \updownarrow_{\omega_u}^{\omega_v}] \vdash Q_i :: (z_{\mathsf{L}} : C_{\mathsf{L}}[\omega_j \updownarrow_{\omega_k}^{\omega_n}])}{\Psi;\ \Gamma;\ \Phi;\ \Delta, x_{\mathsf{L}} : \oplus\{\overline{l : A_{\mathsf{L}}}\}[\omega_m \updownarrow_{\omega_u}^{\omega_v}] \vdash \mathsf{case}\ x_{\mathsf{L}}\ \mathsf{of}\ \overline{l \Rightarrow Q} :: (z_{\mathsf{L}} : C_{\mathsf{L}}[\omega_j \updownarrow_{\omega_k}^{\omega_n}])}\ (\text{T-}\oplus_{\mathrm{L}_1})$$

$$\frac{(\forall i)\ \ \Psi;\ \Gamma;\ \Phi, x_{\mathsf{L}} : A_{\mathsf{L}_i}[\omega_m \updownarrow_{\omega_u}^{\omega_v}];\ \Delta \vdash Q_i :: (z_{\mathsf{L}} : C_{\mathsf{L}}[\omega_j \updownarrow_{\omega_k}^{\omega_n}])}{\Psi;\ \Gamma;\ \Phi, x_{\mathsf{L}} : \oplus\{\overline{l : A_{\mathsf{L}}}\}[\omega_m \updownarrow_{\omega_u}^{\omega_v}];\ \Delta \vdash \mathsf{case}\ x_{\mathsf{L}}\ \mathsf{of}\ \overline{l \Rightarrow Q} :: (z_{\mathsf{L}} : C_{\mathsf{L}}[\omega_j \updownarrow_{\omega_k}^{\omega_n}])}\ (\text{T-}\oplus_{\mathrm{L}_2})$$

$$\frac{\Psi;\ \Gamma;\ \cdot\,;\ \Delta \vdash P :: (x_{\mathsf{L}} : A_{\mathsf{L}\,h}[\omega_m \updownarrow_{\omega_u}^{\omega_v}])}{\Psi;\ \Gamma;\ \cdot\,;\ \Delta \vdash x_{\mathsf{L}}.l_h\,; P :: (x_{\mathsf{L}} : \oplus\{\overline{l : A_{\mathsf{L}}}\}[\omega_m \updownarrow_{\omega_u}^{\omega_v}])}\ (\text{T-}\oplus_{\mathrm{R}})$$

$$\frac{\Psi;\ \Gamma;\ \Phi;\ \Delta, x_{\mathsf{L}} : A_{\mathsf{L}\,h}[\omega_m \updownarrow_{\omega_u}^{\omega_v}] \vdash Q :: (z_{\mathsf{L}} : C_{\mathsf{L}}[\omega_j \updownarrow_{\omega_k}^{\omega_n}])}{\Psi;\ \Gamma;\ \Phi;\ \Delta, x_{\mathsf{L}} : \&\{\overline{l : A_{\mathsf{L}}}\}[\omega_m \updownarrow_{\omega_u}^{\omega_v}] \vdash x_{\mathsf{L}}.l_h\,; Q :: (z_{\mathsf{L}} : C_{\mathsf{L}}[\omega_j \updownarrow_{\omega_k}^{\omega_n}])}\ (\text{T-}\&_{\mathrm{L}_1})$$

$$\frac{\Psi;\ \Gamma;\ \Phi, x_{\mathsf{L}} : A_{\mathsf{L}\,h}[\omega_m \updownarrow_{\omega_u}^{\omega_v}];\ \Delta \vdash Q :: (z_{\mathsf{L}} : C_{\mathsf{L}}[\omega_j \updownarrow_{\omega_k}^{\omega_n}])}{\Psi;\ \Gamma;\ \Phi, x_{\mathsf{L}} : \&\{\overline{l : A_{\mathsf{L}}}\}[\omega_m \updownarrow_{\omega_u}^{\omega_v}];\ \Delta \vdash x_{\mathsf{L}}.l_h\,; Q :: (z_{\mathsf{L}} : C_{\mathsf{L}}[\omega_j \updownarrow_{\omega_k}^{\omega_n}])}\ (\text{T-}\&_{\mathrm{L}_2})$$

$$\frac{(\forall i)\ \ \Psi;\ \Gamma;\cdot\,;\ \Delta \vdash P_i :: (x_{\mathsf{L}} : A_{\mathsf{L}_i}[\omega_m \updownarrow_{\omega_u}^{\omega_v}])}{\Psi;\ \Gamma;\cdot\,;\ \Delta \vdash \mathsf{case}\ x_{\mathsf{L}}\ \mathsf{of}\ \overline{l \Rightarrow P} :: (x_{\mathsf{L}} : \&\{\overline{l : A_{\mathsf{L}}}\}[\omega_m \updownarrow_{\omega_u}^{\omega_v}])}\ (\text{T-}\&_{\mathrm{R}})$$

B.3 Configuration Typing

Given the hierarchy between mode S and L and the fact that shared processes cannot depend on linear processes, we devide a configuration into a *shared* part Λ and a linear part Θ, subject to the following well-formedness conditions:

$$\begin{array}{lll}
\Omega \triangleq \cdot \mid \Lambda;\ \Theta & & \\
\Lambda \triangleq \cdot \mid \mathsf{proc}(a_{\mathsf{S}}, _, P_{a_{\mathsf{S}}}),\ \Lambda' \mid \mathsf{unavail}(a_{\mathsf{S}}, _),\ \Lambda' & & (\mathsf{proc}(a_{\mathsf{S}}, _, _),\ \mathsf{unavail}(a_{\mathsf{S}}, _) \text{ not in } \Lambda') \\
\Theta \triangleq \cdot \mid \mathsf{proc}(a_{\mathsf{L}}, _, P_{a_{\mathsf{L}}}),\ \Theta' & & (\mathsf{proc}(a_{\mathsf{L}}, _, _) \text{ not in } \Theta')
\end{array}$$

In addition, we have the global well-formedness conditions, where $P\langle c_{\mathsf{S}}\rangle$ stands for a process term P with an occurrence of a channel a_{S}:

$$\forall a.\mathsf{proc}(a_{\mathsf{L}},\, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}},\, P_{a_{\mathsf{L}}}) \in \Theta : \mathsf{unavail}(a_{\mathsf{S}},\, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}) \in \Lambda$$
$$\forall a.\mathsf{proc}(a_{\mathsf{L}},\, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}},\, P_{a_{\mathsf{L}}}\langle c_{\mathsf{S}}\rangle) \in \Theta : (\mathsf{unavail}(c_{\mathsf{S}},\, \mathsf{w}_{\mathsf{c}1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}) \in \Lambda \vee \mathsf{proc}(c_{\mathsf{S}},\, \mathsf{w}_{\mathsf{c}1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}},\, Q_{c_{\mathsf{S}}}) \in \Lambda)$$

The configuration typing relies on the below two invariants that a well-formed and well-typed linear configuration must satisfy. These invariants must be preserved along transitions and are crucial to ensure acyclicity of relation $\mathcal{W}(\Theta)$. Invariant 4 relies on Definition B.1, yielding the transitive linear children of a process.

Definition B.1 (Descendants). *If* $\Psi;\, \Gamma \vDash \Theta :: \Phi, \Delta$*, then, for any* $\mathsf{proc}(a_{\mathsf{L}},\, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}},\, P_{a_{\mathsf{L}}}) \in \Theta$ *its descendants* Φ_1, Δ_1 *are defined by the following inductive definition* $(\Psi;\, \Gamma \vDash \Theta :: \Phi, \Delta) \rhd a_{\mathsf{L}} = (\Phi_1, \Delta_1)$*:*

$$\frac{}{(\Psi;\, \Gamma \vDash (\cdot) :: (\cdot)) \rhd a_{\mathsf{L}} = (\cdot)}\ (\text{T-DESC}_1)$$

$$\frac{(\forall b_{\mathsf{L}_i} : B_{\mathsf{L}_i}[\mathsf{w}_{\mathsf{b}_{1_i}}\updownarrow^{\mathsf{w}_{\mathsf{b}_{3_i}}}_{\mathsf{w}_{\mathsf{b}_{2_i}}}] \in \Phi', \Delta')\ \ (\Psi;\, \Gamma \vDash \Theta :: \Phi, \Phi', \Delta, \Delta') \rhd b_{\mathsf{L}_i} = (\Phi_i'', \Delta_i'') \qquad \Psi;\, \Gamma;\, \Phi';\, \Delta' \vdash P_{a_{\mathsf{L}}} :: (a_{\mathsf{L}} : A_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])}{(\Psi;\, \Gamma \vDash \Theta,\, \mathsf{proc}(a_{\mathsf{L}},\, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}},\, P_{a_{\mathsf{L}}}) :: (\Phi, \Delta, a_{\mathsf{L}} : A_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])) \rhd a_{\mathsf{L}} = (\Phi', \overline{\Phi_i''}, \Delta', \overline{\Delta_i''})}\ (\text{T-DESC}_2)$$

$$\frac{a_{\mathsf{L}} \neq b_{\mathsf{L}} \qquad (\Psi;\, \Gamma \vDash \Theta :: \Phi, \Phi', \Delta, \Delta') \rhd a_{\mathsf{L}} = (\Phi'', \Delta'') \qquad \Psi;\, \Gamma;\, \Phi';\, \Delta' \vdash P_{a_{\mathsf{L}}} :: (a_{\mathsf{L}} : A_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])}{(\Psi;\, \Gamma \vDash \Theta,\, \mathsf{proc}(a_{\mathsf{L}},\, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}},\, P_{a_{\mathsf{L}}}) :: (\Phi, \Delta, a_{\mathsf{L}} : A_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])) \rhd b_{\mathsf{L}} = (\Phi'', \Delta'')}\ (\text{T-DESC}_3)$$

We note that Definition B.1 tightly relies on the typing of a linear configuration $\Psi;\, \Gamma \vDash \Theta,\, \mathsf{proc}(a_{\mathsf{L}},\, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}},\, P_{a_{\mathsf{L}}}) :: (\Phi, \Delta, a_{\mathsf{L}} : A_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])$ such that there exists a derivation of $(\Psi;\, \Gamma \vDash \Theta,\, \mathsf{proc}(a_{\mathsf{L}},\, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}},\, P_{a_{\mathsf{L}}}) :: (\Phi, \Delta, a_{\mathsf{L}} : A_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])) \rhd a_{\mathsf{L}} = (\Phi', \Delta')$ only if there exists one of $\Psi;\, \Gamma \vDash \Theta,\, \mathsf{proc}(a_{\mathsf{L}},\, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}},\, P_{a_{\mathsf{L}}}) :: (\Phi, \Delta, a_{\mathsf{L}} : A_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])$ such that the premises of each derivation pairwise concur. We will make use of Definition B.1 in contexts of a well-typed configuration, guaranteeing that this correspondence is given.

Invariant 3 $(\mathsf{min}(\mathsf{parent}) \leq \mathsf{self}(\mathsf{acquired\_child}) \leq \mathsf{max}(\mathsf{parent}))$**.** *For any* $\mathsf{proc}(a_{\mathsf{L}},\, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}},\, P_{a_{\mathsf{L}}})$ *contained in* Θ *of a well-formed and well-typed linear configuration* $\Psi;\, \Gamma \vDash \Theta :: \Phi, \Delta$ *such that* $\Psi;\, \Gamma;\, \Phi_1;\, \Delta_1 \vdash P_{a_{\mathsf{L}}} :: (a_{\mathsf{L}} : A_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}]), \mathsf{Inv}_3(\mathsf{proc}(a_{\mathsf{L}},\, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}},\, P_{a_{\mathsf{L}}}))$ *holds if an only if for every acquired resource* $b_{\mathsf{L}} : B_{\mathsf{L}}[\mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}] \in \Phi_1$ *it holds that* $\Psi^* \vdash \mathsf{w}_{\mathsf{a}_2} \leq \mathsf{w}_{\mathsf{b}_1} \leq \mathsf{w}_{\mathsf{a}_3}$*. Moreover, if* $P_{a_{\mathsf{L}}} = x_{\mathsf{L}} \leftarrow \mathsf{acquire}\, c_{\mathsf{S}}\,; Q_{x_{\mathsf{L}}}$*, for a* $(c_{\mathsf{S}} : \uparrow^{\mathsf{S}}_{\mathsf{L}} C_{\mathsf{L}}[\mathsf{w}_{\mathsf{c}_1}\updownarrow^{\tilde{\mathsf{w}}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}]) \in \Gamma$*, then, for every acquired resource* $b_{\mathsf{L}} : B_{\mathsf{L}}[\mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}] \in \Phi_1$*, it holds that* $\Psi^+ \vdash \mathsf{w}_{\mathsf{b}_1} < \mathsf{w}_{\mathsf{c}_1}$ *and that* $\Psi^* \vdash \mathsf{w}_{\mathsf{a}_2} \leq \mathsf{w}_{\mathsf{c}_1} \leq \mathsf{w}_{\mathsf{a}_3}$*.*

Invariant 4 (max(parent) < minima(descendants)). *For any* $\mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P_{a_{\mathsf{L}}})$ *contained in* Θ *of a well-formed and well-typed linear configuration* $\Psi; \Gamma \vDash \Theta :: \Phi, \Delta$ *and for the process' descendants* $(\Psi; \Gamma \vDash \Theta :: \Phi, \Delta) \rhd a_{\mathsf{L}} = (\Phi', \Delta')$, $\mathsf{Inv}_4(\mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P_{a_{\mathsf{L}}}))$ *holds if an only if for every descendant* $b_{\mathsf{L}} : B_{\mathsf{L}}[\mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}] \in (\Phi', \Delta')$ *it holds that* $\Psi^+ \vdash \mathsf{w}_{\mathsf{a}_3} < \mathsf{w}_{\mathsf{b}_2}$.

We have the below typing rules. The rules presuppose that Ψ^+ irreflexive, a condition that holds for the premise of a rule, if assumed for its conclusion. The configuration typing is relative to a global order, consisting of all the local orders that are α-varied to guarantee freshness.

$$\frac{}{\Psi; \Gamma \vDash (\cdot) :: (\cdot)}\ (\text{T-}\Theta_1)$$

$$\frac{\begin{array}{c}(a_{\mathsf{S}} : \hat{B}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}]) \in \Gamma \qquad \vdash (A_{\mathsf{L}}, \hat{B})\ \mathsf{sesync} \qquad \Psi \vdash A_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}]\ \mathsf{type} \\ \Psi^* \vdash \mathsf{w}_{\mathsf{a}_2} \leq \mathsf{w}_{\mathsf{a}_3} \qquad \mathsf{Inv}_3(\mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P_{a_{\mathsf{L}}})) \qquad \mathsf{Inv}_4(\mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P_{a_{\mathsf{L}}})) \\ \Psi; \Gamma \vDash \Theta :: \Phi, \Phi_1, \Delta, \Delta_1 \qquad \Psi; \Gamma; \Phi_1; \Delta_1 \vdash P_{a_{\mathsf{L}}} :: (a_{\mathsf{L}} : A_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])\end{array}}{\Psi; \Gamma \vDash \Theta, \mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P_{a_{\mathsf{L}}}) :: (\Phi, \Delta, a_{\mathsf{L}} : A_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])}\ (\text{T-}\Theta_2)$$

$$\frac{}{\Psi; \Gamma \vDash (\cdot) :: (\cdot)}\ (\text{T-}\Lambda_1)$$

$$\frac{\vdash (\uparrow^{\mathsf{S}}_{\mathsf{L}} A_{\mathsf{L}}, \top)\ \mathsf{sesync} \qquad \Psi \vdash \uparrow^{\mathsf{S}}_{\mathsf{L}} A_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}]\ \mathsf{type} \qquad \Psi^* \vdash \mathsf{w}_{\mathsf{a}_2} \leq \mathsf{w}_{\mathsf{a}_3} \qquad \Psi; \Gamma \vdash P_{a_{\mathsf{S}}} :: (a_{\mathsf{S}} : \uparrow^{\mathsf{S}}_{\mathsf{L}} A_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])}{\Psi; \Gamma \vDash \mathsf{proc}(a_{\mathsf{S}}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P_{a_{\mathsf{S}}}) :: (a_{\mathsf{S}} : \uparrow^{\mathsf{S}}_{\mathsf{L}} A_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])}\ (\text{T-}\Lambda_2)$$

$$\frac{}{\Psi; \Gamma \vDash \mathsf{unavail}(a_{\mathsf{S}}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}) :: (a_{\mathsf{S}} : \hat{A}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])}\ (\text{T-}\Lambda_3)$$

$$\frac{\Psi; \Gamma \vDash \Lambda :: \Gamma_1 \qquad \Psi; \Gamma \vDash \Lambda' :: \Gamma_2}{\Psi; \Gamma \vDash \Lambda, \Lambda' :: \Gamma_1, \Gamma_2}\ (\text{T-}\Lambda_4)$$

$$\frac{\Psi; \Gamma \vDash \Lambda :: \Gamma \qquad \Psi; \Gamma \vDash \Theta :: \Phi, \Delta}{\Psi; \Gamma \vDash \Lambda; \Theta :: \Gamma; \Phi, \Delta}\ (\text{T-}\Omega)$$

When concatenating a context Γ with a context Γ', variables or channels $x_{\mathsf{s}} : \hat{A}$ that are shared between Γ and Γ' are contracted to the more concrete type.

B.4 Signature Checking

A well-formed *signature* Σ consists of a finite set of process definitions and recursive session type definitions. We assume that the signature Σ is populated prior to type-checking. The rules for checking the signature are given below. For recursive session types, we adopt the *equi-recursive* [12] interpretation developed in [16], requiring recursive session types to be *contractive*. This requirement is

stipulated in rule (T-Σ_3). Moreover, we require session types to be *strictly equi-synchronizing*, a refinement of the notion of an *equi-synchronizing* session type [2] that guarantees that a process that has been acquired at a type A_S must be released back to the type A_S. The notion is stricter because it requires an acquired resource to be released, ruling out blocking acquires due to termination of the resource. Lastly, we also require session types to be well-formed, a condition imposed on higher-order channels to make sure that they satisfy invariant 2 defined in Section 3. We give either set of rules in Figure 13 and Figure 14, respectively.

$$\frac{}{\vdash (\cdot)\ \mathsf{sig}}\ (\text{T-}\Sigma_1)$$

$$\frac{\begin{array}{c}\vdash (A_\mathsf{L}, \top)\ \mathsf{sesync} \qquad \vdash \Sigma'\ \mathsf{sig} \\ \Psi' = \mathsf{wrl}(A_\mathsf{L}[\delta_j \updownarrow_{\delta_k}^{\delta_n}]), \mathsf{wrl}(\Delta), \mathsf{wrl}(\Phi), \mathsf{wrl}(\Gamma) \qquad \Psi =< (\Psi') \qquad \Psi^+ \text{ irreflexive} \\ \Psi \vdash A_\mathsf{L}[\delta_j \updownarrow_{\delta_k}^{\delta_n}]\ \mathsf{type} \qquad \forall y_{\mathsf{L}_i} : B_{\mathsf{L}_i}[\delta_{m_i} \updownarrow_{\delta_{u_i}}^{\delta_{v_i}}] \in \Delta \cup \Phi : \Psi \vdash B_{\mathsf{L}_i}[\delta_{m_i} \updownarrow_{\delta_{u_i}}^{\delta_{v_i}}]\ \mathsf{type} \\ \forall z_{\mathsf{S}_i} : C_{\mathsf{S}_i}[\delta_{l_i} \updownarrow_{\delta_{p_i}}^{\delta_{r_i}}] \in \Gamma : \Psi \vdash C_{\mathsf{S}_i}[\delta_{l_i} \updownarrow_{\delta_{p_i}}^{\delta_{r_i}}]\ \mathsf{type} \\ \Psi^* \vdash \delta_k \leq \delta_n \\ \forall y_{\mathsf{L}_i} : B_{\mathsf{L}_i}[\delta_{m_i} \updownarrow_{\delta_{u_i}}^{\delta_{v_i}}] \in \Phi : \Psi^* \vdash \delta_k \leq \delta_{m_i} \leq \delta_n \qquad \forall y_{\mathsf{L}_i} : B_{\mathsf{L}_i}[\delta_{m_i} \updownarrow_{\delta_{u_i}}^{\delta_{v_i}}] \in \Delta \cup \Phi : \Psi^+ \vdash \delta_n < \delta_{u_i} \\ \Psi;\ \Gamma;\ \Phi;\ \Delta \vdash P_{x_\mathsf{L}, \mathsf{dom}(\Delta), \mathsf{dom}(\Phi), \mathsf{dom}(\Gamma), \Psi'} :: (x_\mathsf{L} : A_\mathsf{L}[\delta_j \updownarrow_{\delta_k}^{\delta_n}])\end{array}}{\vdash (\Psi \vdash x_\mathsf{L} : A_\mathsf{L}[\delta_j \updownarrow_{\delta_k}^{\delta_n}] \leftarrow X_\mathsf{L} \leftarrow \Delta, \Phi, \Gamma = P_{x_\mathsf{L}, \mathsf{dom}(\Delta), \mathsf{dom}(\Phi), \mathsf{dom}(\Gamma), \Psi'}),\ \Sigma'\ \mathsf{sig}}\ (\text{T-}\Sigma_2)$$

$$\frac{\begin{array}{c}\vdash (A_\mathsf{S}, \top)\ \mathsf{sesync} \qquad \vdash \Sigma'\ \mathsf{sig} \\ \Psi' = \mathsf{wrl}(A_\mathsf{S}[\delta_j \updownarrow_{\delta_k}^{\delta_n}]), \mathsf{wrl}(\Gamma) \qquad \Psi =< (\Psi') \qquad \Psi^+ \text{ irreflexive} \\ \Psi \vdash A_\mathsf{S}[\delta_j \updownarrow_{\delta_k}^{\delta_n}]\ \mathsf{type} \qquad \forall z_{\mathsf{S}_i} : C_{\mathsf{S}_i}[\delta_{l_i} \updownarrow_{\delta_{p_i}}^{\delta_{r_i}}] \in \Gamma : \Psi \vdash C_{\mathsf{S}_i}[\delta_{l_i} \updownarrow_{\delta_{p_i}}^{\delta_{r_i}}]\ \mathsf{type} \\ \Psi^* \vdash \delta_k \leq \delta_n \\ \Psi;\ \Gamma;\ \Phi;\ \Delta \vdash P_{x_\mathsf{S}, \mathsf{dom}(\Gamma), \Psi'} :: (x_\mathsf{S} : A_\mathsf{S}[\delta_j \updownarrow_{\delta_k}^{\delta_n}])\end{array}}{\vdash (\Psi \vdash x_\mathsf{S} : A_\mathsf{S}[\delta_j \updownarrow_{\delta_k}^{\delta_n}] \leftarrow X_\mathsf{S} \leftarrow \Gamma = P_{x_\mathsf{S}, \mathsf{dom}(\Gamma), \Psi'}),\ \Sigma'\ \mathsf{sig}}\ (\text{T-}\Sigma_3)$$

$$\frac{\vdash A_m\ \mathsf{contr} \qquad \vdash (A_m, \top)\ \mathsf{sesync} \qquad \vdash \Sigma'\ \mathsf{sig}}{\vdash Y_m = A_m,\ \Sigma'\ \mathsf{sig}}\ (\text{T-}\Sigma_4)$$

C Dynamics

The below rules give the synchronous dynamics of SILL_S in terms of multiset rewriting rules [10].

(D-Id$_\mathsf{S}$)
$\mathsf{proc}(a_\mathsf{S}, \mathsf{w}_{\mathsf{a}_1} \updownarrow_{\mathsf{w}_{\mathsf{a}_2}}^{\mathsf{w}_{\mathsf{a}_3}}, \mathsf{fwd}\ a_\mathsf{S}\ b_\mathsf{S}) \longrightarrow \mathsf{unavail}(a_\mathsf{S}, \mathsf{w}_{\mathsf{a}_1} \updownarrow_{\mathsf{w}_{\mathsf{a}_2}}^{\mathsf{w}_{\mathsf{a}_3}}),\ a_\mathsf{S} = b_\mathsf{S}$

$$\frac{(\forall i)\ \vdash (A_{\mathsf{L}_i}, \hat{D})\ \mathsf{sesync}}{\vdash (\oplus\{\overline{l : A_{\mathsf{L}}}\}, \hat{D})\ \mathsf{sesync}}\ (\text{T-SESYNC}_{\oplus}) \qquad \frac{(\forall i)\ \vdash (A_{\mathsf{L}_i}, \hat{D})\ \mathsf{sesync}}{\vdash (\&\{\overline{l : A_{\mathsf{L}}}\}, \hat{D})\ \mathsf{sesync}}\ (\text{T-SESYNC}_{\&})$$

$$\frac{\vdash (B_{\mathsf{L}}, \hat{D})\ \mathsf{sesync}}{\vdash (A_{\mathsf{L}} \otimes B_{\mathsf{L}}, \hat{D})\ \mathsf{sesync}}\ (\text{T-SESYNC}_{\otimes}) \qquad \frac{\vdash (B_{\mathsf{L}}, \hat{D})\ \mathsf{sesync}}{\vdash (A_{\mathsf{L}} \multimap B_{\mathsf{L}}, \hat{D})\ \mathsf{sesync}}\ (\text{T-SESYNC}_{\multimap})$$

$$\frac{\vdash (B_{\mathsf{L}}, \hat{D})\ \mathsf{sesync}}{\vdash (\exists x{:}A_{\mathsf{S}}.\, B_{\mathsf{L}}, \hat{D})\ \mathsf{sesync}}\ (\text{T-SESYNC}_{\exists}) \qquad \frac{\vdash (B_{\mathsf{L}}, \hat{D})\ \mathsf{sesync}}{\vdash (\Pi x{:}A_{\mathsf{S}}.\, B_{\mathsf{L}}, \hat{D})\ \mathsf{sesync}}\ (\text{T-SESYNC}_{\Pi})$$

$$\frac{}{\vdash (\mathbf{1}, \top)\ \mathsf{sesync}}\ (\text{T-SESYNC}_{\mathbf{1}})$$

$$\frac{\vdash (A_{\mathsf{L}}, \uparrow^{\mathsf{S}}_{\mathsf{L}} A_{\mathsf{L}})\ \mathsf{sesync}}{\vdash (\uparrow^{\mathsf{S}}_{\mathsf{L}} A_{\mathsf{L}}, \uparrow^{\mathsf{S}}_{\mathsf{L}} A_{\mathsf{L}})\ \mathsf{sesync}}\ (\text{T-SESYNC}_{\uparrow^{\mathsf{S}}_{\mathsf{L}_1}}) \qquad \frac{\vdash (A_{\mathsf{L}}, \uparrow^{\mathsf{S}}_{\mathsf{L}} A_{\mathsf{L}})\ \mathsf{sesync}}{\vdash (\uparrow^{\mathsf{S}}_{\mathsf{L}} A_{\mathsf{L}}, \top)\ \mathsf{sesync}}\ (\text{T-SESYNC}_{\uparrow^{\mathsf{S}}_{\mathsf{L}_2}})$$

$$\frac{\vdash (\uparrow^{\mathsf{S}}_{\mathsf{L}} A_{\mathsf{L}}, \uparrow^{\mathsf{S}}_{\mathsf{L}} A_{\mathsf{L}})\ \mathsf{sesync}}{\vdash (\downarrow^{\mathsf{S}}_{\mathsf{L}} \uparrow^{\mathsf{S}}_{\mathsf{L}} A_{\mathsf{L}}, \uparrow^{\mathsf{S}}_{\mathsf{L}} A_{\mathsf{L}})\ \mathsf{sesync}}\ (\text{T-SESYNC}_{\downarrow^{\mathsf{S}}_{\mathsf{L}_1}}) \qquad \frac{\vdash (\uparrow^{\mathsf{S}}_{\mathsf{L}} A_{\mathsf{L}}, \uparrow^{\mathsf{S}}_{\mathsf{L}} A_{\mathsf{L}})\ \mathsf{sesync}}{\vdash (\downarrow^{\mathsf{S}}_{\mathsf{L}} \uparrow^{\mathsf{S}}_{\mathsf{L}} A_{\mathsf{L}}, \top)\ \mathsf{sesync}}\ (\text{T-SESYNC}_{\downarrow^{\mathsf{S}}_{\mathsf{L}_2}})$$

Fig. 13: Strictly equi-synchronizing session type, coinductively defined.

$$\frac{(\forall i)\ \Psi \vdash A_{\mathsf{L}_i}[\omega_m \updownarrow^{\omega_v}_{\omega_u}]\ \mathsf{type}}{\Psi \vdash \oplus\{\overline{l : A_{\mathsf{L}}}\}[\omega_m \updownarrow^{\omega_v}_{\omega_u}]\ \mathsf{type}}\ (\text{T-type}_{\oplus}) \qquad \frac{(\forall i)\ \Psi \vdash A_{\mathsf{L}_i}[\omega_m \updownarrow^{\omega_v}_{\omega_u}]\ \mathsf{type}}{\Psi \vdash \&\{\overline{l : A_{\mathsf{L}}}\}[\omega_m \updownarrow^{\omega_v}_{\omega_u}]\ \mathsf{type}}\ (\text{T-type}_{\&})$$

$$\frac{\Psi^{+} \vdash \omega_v < \omega_p \qquad \Psi \vdash B_{\mathsf{L}}[\omega_m \updownarrow^{\omega_v}_{\omega_u}]\ \mathsf{type}}{\Psi \vdash A_{\mathsf{L}}@\omega_l \updownarrow^{\omega_r}_{\omega_p} \otimes B_{\mathsf{L}}[\omega_m \updownarrow^{\omega_v}_{\omega_u}]\ \mathsf{type}}\ (\text{T-type}_{\otimes})$$

$$\frac{\Psi^{+} \vdash \omega_v < \omega_p \qquad \Psi \vdash B_{\mathsf{L}}[\omega_m \updownarrow^{\omega_v}_{\omega_u}]\ \mathsf{type}}{\Psi \vdash A_{\mathsf{L}}@\omega_l \updownarrow^{\omega_r}_{\omega_p} \multimap B_{\mathsf{L}}[\omega_m \updownarrow^{\omega_v}_{\omega_u}]\ \mathsf{type}}\ (\text{T-type}_{\multimap})$$

$$\frac{\Psi \vdash B_{\mathsf{L}}[\omega_m \updownarrow^{\omega_v}_{\omega_u}]\ \mathsf{type}}{\Psi \vdash \exists x{:}A_{\mathsf{S}}@\omega_l \updownarrow^{\omega_r}_{\omega_p}.\, B_{\mathsf{L}}[\omega_m \updownarrow^{\omega_v}_{\omega_u}]\ \mathsf{type}}\ (\text{T-type}_{\exists})$$

$$\frac{\Psi \vdash B_{\mathsf{L}}[\omega_m \updownarrow^{\omega_v}_{\omega_u}]\ \mathsf{type}}{\Psi \vdash \Pi x{:}A_{\mathsf{S}}@\omega_l \updownarrow^{\omega_r}_{\omega_p}.\, B_{\mathsf{L}}[\omega_m \updownarrow^{\omega_v}_{\omega_u}]\ \mathsf{type}}\ (\text{T-type}_{\Pi})$$

$$\frac{}{\Psi \vdash \mathbf{1}[\omega_m \updownarrow^{\omega_v}_{\omega_u}]\ \mathsf{type}}\ (\text{T-type}_{\mathbf{1}})$$

$$\frac{\Psi \vdash A_{\mathsf{L}}[\omega_m \updownarrow^{\omega_v}_{\omega_u}]\ \mathsf{type}}{\Psi \vdash \uparrow^{\mathsf{S}}_{\mathsf{L}} A_{\mathsf{L}}[\omega_m \updownarrow^{\omega_v}_{\omega_u}]\ \mathsf{type}}\ (\text{T-type}_{\uparrow^{\mathsf{S}}_{\mathsf{L}}}) \qquad \frac{\Psi \vdash A_{\mathsf{S}}[\omega_m \updownarrow^{\omega_v}_{\omega_u}]\ \mathsf{type}}{\Psi \vdash \downarrow^{\mathsf{S}}_{\mathsf{L}} A_{\mathsf{S}}[\omega_m \updownarrow^{\omega_v}_{\omega_u}]\ \mathsf{type}}\ (\text{T-type}_{\downarrow^{\mathsf{S}}_{\mathsf{L}}})$$

Fig. 14: Well-formed session type, coinductively defined.

$$\begin{aligned}
&(\text{D-SPAWN}_{\mathsf{LL}})\\
&\mathsf{proc}(a_{\mathsf{L}},\ \mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}},\ x_{\mathsf{L}} : A_{\mathsf{L}}[\mathsf{w}_{\mathsf{b}_1} \updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}] \leftarrow X_{\mathsf{L}} \leftarrow \overline{c_{\mathsf{L}}}, \overline{\tilde{c}_{\mathsf{L}}}, \overline{d_{\mathsf{S}}};\ Q_{x_{\mathsf{L}}}),\\
&!\mathsf{def}(\,\Psi' \vdash x'_{\mathsf{L}} : A'_{\mathsf{L}}[\delta_j \updownarrow^{\delta_n}_{\delta_k}] \leftarrow X_{\mathsf{L}} \leftarrow \Delta',\, \Phi',\, \Gamma' = P_{x'_{\mathsf{L}}, \mathsf{dom}(\Delta'), \mathsf{dom}(\Phi'), \mathsf{dom}(\Gamma'),\, \Psi''})\\
&\longrightarrow \mathsf{proc}(b_{\mathsf{L}},\ \mathsf{w}_{\mathsf{b}_1} \updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}},\ [b_{\mathsf{L}}/x'_{\mathsf{L}}, \overline{c_{\mathsf{L}}}/\mathsf{dom}(\Delta'), \overline{\tilde{c}_{\mathsf{L}}}/\mathsf{dom}(\Phi'), \overline{d_{\mathsf{S}}}/\mathsf{dom}(\Gamma')]\hat{\gamma}(P_{x'_{\mathsf{L}}, \mathsf{dom}(\Delta'), \mathsf{dom}(\Phi'), \mathsf{dom}(\Gamma'),\, \Psi''})),\\
&\quad\ \mathsf{proc}(a_{\mathsf{L}},\ \mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}},\ [b_{\mathsf{L}}/x_{\mathsf{L}}]\, Q_{x_{\mathsf{L}}}),
\end{aligned}$$

$\mathsf{unavail}(b_{\mathsf{S}},\, \mathsf{w}_{\mathsf{b}_1} \updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}})$ $\quad (b\ \mathit{fresh})$

(D-Tail$_{\mathsf{L}}$)
$\mathsf{proc}(a_{\mathsf{L}},\, \mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}},\, x_{\mathsf{L}} \leftarrow X_{\mathsf{L}} \leftarrow \overline{c_{\mathsf{L}}}, \overline{\tilde{c}_{\mathsf{L}}}, \overline{d_{\mathsf{S}}}),$
$!\mathsf{def}(\, \Psi' \vdash x'_{\mathsf{L}} : A'_{\mathsf{L}}[\delta_j \updownarrow^{\delta_n}_{\delta_k}] \leftarrow X_{\mathsf{L}} \leftarrow \Delta',\, \Phi',\, \Gamma' = P_{x'_{\mathsf{L}}, \mathsf{dom}(\Delta'), \mathsf{dom}(\Phi'), \mathsf{dom}(\Gamma'),\, \Psi''})$
$\longrightarrow\ \mathsf{proc}(a_{\mathsf{L}},\, \mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}},\, [a_{\mathsf{L}}/x'_{\mathsf{L}}, \overline{c_{\mathsf{L}}}/\mathsf{dom}(\Delta'), \overline{\tilde{c}_{\mathsf{L}}}/\mathsf{dom}(\Phi'), \overline{d_{\mathsf{S}}}/\mathsf{dom}(\Gamma')]\hat{\gamma}(P_{x'_{\mathsf{L}}, \mathsf{dom}(\Delta'), \mathsf{dom}(\Phi'), \mathsf{dom}(\Gamma'),\, \Psi''}))$

(D-Spawn$_{\mathsf{LS}}$)
$\mathsf{proc}(a_{\mathsf{L}},\, \mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}},\, x_{\mathsf{S}} : A_{\mathsf{S}}[\mathsf{w}_{\mathsf{b}_1} \updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}] \leftarrow X_{\mathsf{S}} \leftarrow \overline{d_{\mathsf{S}}};\ Q_{x_{\mathsf{S}}}),$
$!\mathsf{def}(\, \Psi' \vdash x'_{\mathsf{S}} : A'_{\mathsf{S}}[\delta_j \updownarrow^{\delta_n}_{\delta_k}] \leftarrow X_{\mathsf{S}} \leftarrow \Gamma' = P_{x'_{\mathsf{S}}, \mathsf{dom}(\Gamma'),\, \Psi''})$
$\longrightarrow\ \mathsf{proc}(b_{\mathsf{S}},\, \mathsf{w}_{\mathsf{b}_1} \updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}},\, [b_{\mathsf{S}}/x'_{\mathsf{S}}, \overline{d_{\mathsf{S}}}/\mathsf{dom}(\Gamma')]\hat{\gamma}(P_{x'_{\mathsf{L}}, \mathsf{dom}(\Gamma'),\, \Psi''})),\, \mathsf{proc}(a_{\mathsf{L}},\, \mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}},\, [b_{\mathsf{S}}/x_{\mathsf{S}}]Q_{x_{\mathsf{S}}})$ $\quad (b\ \mathit{fresh})$

(D-Spawn$_{\mathsf{SS}}$)
$\mathsf{proc}(a_{\mathsf{S}},\, \mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}},\, x_{\mathsf{S}} : A_{\mathsf{S}}[\mathsf{w}_{\mathsf{b}_1} \updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}] \leftarrow X_{\mathsf{S}} \leftarrow \overline{d_{\mathsf{S}}};\ Q_{x_{\mathsf{S}}}),$
$!\mathsf{def}(\, \Psi' \vdash x'_{\mathsf{S}} : A'_{\mathsf{S}}[\delta_j \updownarrow^{\delta_n}_{\delta_k}] \leftarrow X_{\mathsf{S}} \leftarrow \Gamma' = P_{x'_{\mathsf{S}}, \mathsf{dom}(\Gamma'),\, \Psi''})$
$\longrightarrow\ \mathsf{proc}(b_{\mathsf{S}},\, \mathsf{w}_{\mathsf{b}_1} \updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}},\, [b_{\mathsf{S}}/x'_{\mathsf{S}}, \overline{d_{\mathsf{S}}}/\mathsf{dom}(\Gamma')]\hat{\gamma}(P_{x'_{\mathsf{L}}, \mathsf{dom}(\Gamma'),\, \Psi''})),\, \mathsf{proc}(a_{\mathsf{S}},\, \mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}},\, [b_{\mathsf{S}}/x_{\mathsf{S}}]Q_{x_{\mathsf{S}}})$ $\quad (b\ \mathit{fresh})$

(D-New)
$\mathsf{proc}(a,\, \mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}},\, \mathsf{w} \leftarrow \mathsf{new\_world};\ Q_{\mathsf{w}'}) \longrightarrow \mathsf{proc}(a,\, \mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}},\, Q_{\mathsf{w}'})$ $\quad (\mathsf{w}\ \mathit{fresh})$

(D-Ord) $\mathsf{proc}(a,\, \mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}},\, \mathsf{w} < \mathsf{w}';\ Q) \longrightarrow \mathsf{proc}(a,\, \mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}},\, Q)$

(D-$\uparrow^{\mathsf{S}}_{\mathsf{L}}$)
$\mathsf{proc}(a_{\mathsf{S}},\, \mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}},\, x_{\mathsf{L}} \leftarrow \mathsf{accept}\ a_{\mathsf{S}}\, ;\, P_{x_{\mathsf{L}}}),\, \mathsf{proc}(c_{\mathsf{L}},\, \mathsf{w}_{\mathsf{c}_1} \updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}},\, x_{\mathsf{L}} \leftarrow \mathsf{acquire}\ a_{\mathsf{S}}\, ;\, Q_{x_{\mathsf{L}}})$
$\longrightarrow\ \mathsf{proc}(a_{\mathsf{L}},\, \mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}},\, [a_{\mathsf{L}}/x_{\mathsf{L}}]\, P_{x_{\mathsf{L}}}),\, \mathsf{proc}(c_{\mathsf{L}},\, \mathsf{w}_{\mathsf{c}_1} \updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}},\, [a_{\mathsf{L}}/x_{\mathsf{L}}]\, Q_{x_{\mathsf{L}}}),\, \mathsf{unavail}(a_{\mathsf{S}},\, \mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}})$

(D-$\downarrow^{\mathsf{S}}_{\mathsf{L}}$)
$\mathsf{proc}(a_{\mathsf{L}},\, \mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}},\, x_{\mathsf{S}} \leftarrow \mathsf{detach}\ a_{\mathsf{L}}\, ;\, P_{x_{\mathsf{S}}}),\, \mathsf{proc}(c_{\mathsf{L}},\, \mathsf{w}_{\mathsf{c}_1} \updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}},\, x_{\mathsf{S}} \leftarrow \mathsf{release}\ a_{\mathsf{L}}\, ;\, Q_{x_{\mathsf{S}}}),\, \mathsf{unavail}(a_{\mathsf{S}},\, \mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}})$
$\longrightarrow\ \mathsf{proc}(a_{\mathsf{S}},\, \mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}},\, [a_{\mathsf{S}}/x_{\mathsf{S}}]\, P_{x_{\mathsf{S}}}),\, \mathsf{proc}(c_{\mathsf{L}},\, \mathsf{w}_{\mathsf{c}_1} \updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}},\, [a_{\mathsf{S}}/x_{\mathsf{S}}]\, Q_{x_{\mathsf{S}}})$

(D-$\mathbf{1}$)
$\mathsf{proc}(a_{\mathsf{L}},\, \mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}},\, \mathsf{close}\ a_{\mathsf{L}}),\, \mathsf{proc}(c_{\mathsf{L}},\, \mathsf{w}_{\mathsf{c}_1} \updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}},\, \mathsf{wait}\ a_{\mathsf{L}}\, ;\, Q)$
$\longrightarrow\ \mathsf{proc}(c_{\mathsf{L}},\, \mathsf{w}_{\mathsf{c}_1} \updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}},\, Q)$

(D-$\otimes/\exists$)
$\mathsf{proc}(a_{\mathsf{L}},\, \mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}},\, \mathsf{send}\ a_{\mathsf{L}}\ b\, ;\, P),\, \mathsf{proc}(c_{\mathsf{L}},\, \mathsf{w}_{\mathsf{c}_1} \updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}},\, y \leftarrow \mathsf{recv}\ a_{\mathsf{L}}\, ;\, Q_y)$
$\longrightarrow\ \mathsf{proc}(a_{\mathsf{L}},\, \mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}},\, P),\, \mathsf{proc}(c_{\mathsf{L}},\, \mathsf{w}_{\mathsf{c}_1} \updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}},\, [b/y]\, Q_y)$

(D-$\multimap/\Pi$)
$\mathsf{proc}(a_{\mathsf{L}},\, \mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}},\, y \leftarrow \mathsf{recv}\ a_{\mathsf{L}}\, ;\, P_y),\, \mathsf{proc}(c_{\mathsf{L}},\, \mathsf{w}_{\mathsf{c}_1} \updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}},\, \mathsf{send}\ a_{\mathsf{L}}\ b\, ;\, Q)$
$\longrightarrow\ \mathsf{proc}(a_{\mathsf{L}},\, \mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}},\, [b/y]P_y),\, \mathsf{proc}(c_{\mathsf{L}},\, \mathsf{w}_{\mathsf{c}_1} \updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}},\, Q)$

(D-$\oplus$)
$\mathsf{proc}(a_{\mathsf{L}},\, \mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}},\, a_{\mathsf{L}}.l_h\, ;\, P),\, \mathsf{proc}(c_{\mathsf{L}},\, \mathsf{w}_{\mathsf{c}_1} \updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}},\, \mathsf{case}\ a_{\mathsf{L}}\ \mathsf{of}\ \overline{l \Rightarrow Q})$
$\longrightarrow\ \mathsf{proc}(a_{\mathsf{L}},\, \mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}},\, P),\, \mathsf{proc}(c_{\mathsf{L}},\, \mathsf{w}_{\mathsf{c}_1} \updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}},\, Q_h)$

(D-$\&$)
$\mathsf{proc}(a_{\mathsf{L}},\, \mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}},\, \mathsf{case}\ a_{\mathsf{L}}\ \mathsf{of}\ \overline{l \Rightarrow P}),\, \mathsf{proc}(c_{\mathsf{L}},\, \mathsf{w}_{\mathsf{c}_1} \updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}},\, a_{\mathsf{L}}.l_h\, ;\, Q)$
$\longrightarrow\ \mathsf{proc}(a_{\mathsf{L}},\, \mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}},\, P_h),\, \mathsf{proc}(c_{\mathsf{L}},\, \mathsf{w}_{\mathsf{c}_1} \updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}},\, Q)$

D Progress

D.1 Definitions

Definition D.1 (Poised Process and Configuration). *A* $\mathsf{proc}(a, \mathsf{w}, P_a)$ *is poised if it is communicating along its providing channel. The poised processes in* $\mathsf{SILL_S}$ *are:*

Receiving	*Sending*
$\mathsf{proc}(a_\mathsf{L}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, y \leftarrow \mathsf{recv}\ a_\mathsf{L}\,;P_y)$	$\mathsf{proc}(a_\mathsf{L}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, \mathsf{send}\ a_\mathsf{L}\ b\,;P)$
	$\mathsf{proc}(a_\mathsf{L}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, \mathsf{close}\ a_\mathsf{L})$
$\mathsf{proc}(a_\mathsf{L}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, \mathsf{case}\ a_\mathsf{L}\ \mathsf{of}\ \overline{l \Rightarrow P})$	$\mathsf{proc}(a_\mathsf{L}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, a_\mathsf{L}.l_h\,;P)$
$\mathsf{proc}(a_\mathsf{S}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, x_\mathsf{L} \leftarrow \mathsf{accept}\ a_\mathsf{S}\,;P_{x_\mathsf{L}})$	$\mathsf{proc}(a_\mathsf{L}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, x_\mathsf{S} \leftarrow \mathsf{detach}\ a_\mathsf{L}\,;P_{x_\mathsf{S}})$

Moreover, we consider $\mathsf{unavail}(a_\mathsf{L}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}})$ *to be poised. A configuration* Θ *is poised if and only if all* $\mathsf{proc}(a_\mathsf{L}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P_{a_\mathsf{L}}) \in \Theta$ *are poised. A configuration* Λ *is poised if and only if all* $\mathsf{proc}(a_\mathsf{S}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P_{a_\mathsf{S}}) \in \Lambda$ *are poised.*

Definition D.2 (Acquire Dependency — "Red Arrows"). *Given a well-formed and well-typed configuration* $\Psi; \Gamma \vDash \Lambda; \Theta :: \Gamma; \Phi, \Delta$*, there exists a waiting-due-to-acquire relation* $\mathcal{A}(\Theta)$ *among linear processes in* Θ *at run-time such that*

$$\mathsf{proc}(a_\mathsf{L}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, x_\mathsf{L} \leftarrow \mathsf{acquire}\ c_\mathsf{S};\ Q_{x_\mathsf{L}}) <_\mathcal{A} \mathsf{proc}(b_\mathsf{L}, \mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}, P\langle c_\mathsf{L}\rangle)$$

where $P\langle c_\mathsf{L}\rangle$ *denotes a process term with an occurrence of channel* c_L*.*

Definition D.3 (Synchronization Dependency — "Green Arrows"). *Given a well-formed and well-typed configuration* $\Psi; \Gamma \vDash \Lambda; \Theta :: \Gamma; \Phi, \Delta$*, there exists a waiting-due-to-synchronization relation* $\mathcal{S}(\Theta)$ *among linear processes in* Θ *at run-time such that*

$$\mathsf{proc}(a_\mathsf{L}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, \_\langle b_\mathsf{L}\rangle; Q) <_\mathcal{S} \mathsf{proc}(b_\mathsf{L}, \mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}, \_\langle\neg b_\mathsf{L}\rangle; P)$$
$$\mathsf{proc}(b_\mathsf{L}, \mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}, \_\langle b_\mathsf{L}\rangle; P) <_\mathcal{S} \mathsf{proc}(a_\mathsf{L}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, \_\langle\neg b_\mathsf{L}\rangle; Q\langle b_\mathsf{L}\rangle)$$

where $P\langle a_\mathsf{L}\rangle$ *denotes a process term with an occurrence of channel* b_L*,* $\_\langle a\rangle; P$ *a process term that currently executes an action along channel* a*, and* $\_\langle\neg a\rangle; P$ *a process term whose currently executing action does not involve the channel* a*.*

Definition D.4 (Waiting Dependency). *Given a well-formed and well-typed configuration* $\Psi; \Gamma \vDash \Lambda; \Theta :: \Gamma; \Phi, \Delta$*, there exists a waiting relation* $\mathcal{W}(\Theta)$ *among processes in* Θ *at run-time such that* $\mathsf{proc}(a_\mathsf{L}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P) <_\mathcal{W} \mathsf{proc}(b_\mathsf{L}, \mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}, Q)$*,*

- *if* $\mathsf{proc}(a_\mathsf{L}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P) <_\mathcal{A} \mathsf{proc}(b_\mathsf{L}, \mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}, Q)$*, or*
- *if* $\mathsf{proc}(a_\mathsf{L}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P) <_\mathcal{S} \mathsf{proc}(b_\mathsf{L}, \mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}, Q)$*.*

D.2 Lemmas and Corollaries

Corollary D.5 (At most one out-going arrow). *For a well-formed and well-typed configuration* $\Psi; \Gamma \vDash \Lambda; \Theta :: \Gamma; \Phi, \Delta$ *and for any* $\mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P_{a_{\mathsf{L}}}) \in \Theta$*, there exists at most one occurrence of* $\mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P_{a_{\mathsf{L}}})$ *in* $\mathsf{dom}(\mathcal{W}(\Theta))$.

Lemma D.6 (Empty Φ when synchronizing along offering channel). *In a well-formed and well-typed configuration* $\Psi; \Gamma \vDash \Lambda; \Theta :: \Gamma; \Phi, \Delta$ *and for any* $\mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P_{a_{\mathsf{L}}}) \in \Theta$ *such that* $\Psi; \Gamma; \Phi_1; \Delta_1 \vdash P_{a_{\mathsf{L}}} :: (a_{\mathsf{L}} : A_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])$*, if* $P_{a_{\mathsf{L}}}$ *executes a synchronization action along channel* a_{L}*, then* Φ_1 *is empty.*

Lemma D.7 (No red arrows between parent and descendants). *In a well-formed and well-typed linear configuration* $\Psi; \Gamma \vDash \Theta :: \Phi, \Delta$ *and for any* $\mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P_{a_{\mathsf{L}}}) \in \Theta$ *and its descendants* $(\Psi; \Gamma \vDash \Theta :: \Phi, \Delta) \rhd a_{\mathsf{L}} = (\Phi', \Delta')$*, it holds for every descendant* $b_{\mathsf{L}} : B_{\mathsf{L}}[\mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}] \in (\Phi', \Delta')$ *that* $\mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P_{a_{\mathsf{L}}}) \not<_{\mathcal{A}} \mathsf{proc}(b_{\mathsf{L}}, \mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}, Q_{b_{\mathsf{L}}})$ *and* $\mathsf{proc}(b_{\mathsf{L}}, \mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}, Q_{b_{\mathsf{L}}}) \not<_{\mathcal{A}} \mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P_{a_{\mathsf{L}}})$.

Lemma D.8 (No outgoing green arrow to parent with ingoing red arrow). *In a well-formed and well-typed configuration* $\Psi; \Gamma \vDash \Lambda; \Theta :: \Gamma; \Phi, \Delta$ *and for any* $\mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P_{a_{\mathsf{L}}}) \in \Theta$*, such that* $\Psi; \Gamma; \Phi_1; \Delta_1 \vdash P_{a_{\mathsf{L}}} :: (a_{\mathsf{L}} : A_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])$*, and for any child* $b_{\mathsf{L}} : B_{\mathsf{L}}[\mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}] \in (\Phi_1, \Delta_1)$*, it holds that, if* $\mathsf{proc}(b_{\mathsf{L}}, \mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}, Q_{b_{\mathsf{L}}}) <_{\mathcal{S}} \mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P_{a_{\mathsf{L}}})$*, then there cannot exist a* $\mathsf{proc}(c_{\mathsf{L}}, \mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}, P'_{c_{\mathsf{L}}}) \in \Theta$ *such that* $\mathsf{proc}(c_{\mathsf{L}}, \mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}, P'_{c_{\mathsf{L}}}) <_{\mathcal{A}} \mathsf{proc}(b_{\mathsf{L}}, \mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}, Q_{b_{\mathsf{L}}})$.

Lemma D.9 (Acyclicity of $\mathcal{W}$). *If* $\Psi; \Gamma \vDash \Lambda; \Theta :: \Gamma; \Phi, \Delta$*, then* $\mathcal{W}(\Theta)$ *is acyclic.*

Proof.

$\Psi; \Gamma \vDash \Lambda; \Theta :: \Gamma; \Phi, \Delta$	(by assumption)
$\Psi; \Gamma \vDash \Theta :: \Phi, \Delta$	(by inversion on (T-Ω))
Ψ^+ irreflexive	(by inversion on (T-Ω))

We proceed by induction on $\Psi; \Gamma \vDash \Theta :: \Phi, \Delta$:

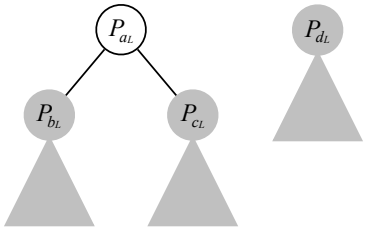
1. $\Psi; \Gamma \vDash (\cdot) :: (\cdot)$

   $\mathcal{W}(\cdot)$ is vacuously acyclic.

2. $\Psi; \Gamma \vDash \Theta_1, \mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P_{a_{\mathsf{L}}}) :: (\Phi_1, \Delta_1, a_{\mathsf{L}} : A_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])$

$\Psi; \Gamma \vDash \Theta_1 :: \Phi_1, \Phi'_1, \Delta_1, \Delta'_1$	(by inversion on (T-Θ_2))
$\Psi; \Gamma; \Phi'_1; \Delta'_1 \vdash P_{a_{\mathsf{L}}} :: (a_{\mathsf{L}} : A_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])$	(by inversion on (T-Θ_2))
$\mathsf{Inv}_3(\Theta_1)$ and $\mathsf{Inv}_3(\mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P_{a_{\mathsf{L}}}))$	(by inversion on (T-Θ_2))
$\mathsf{Inv}_4(\Theta_1)$ and $\mathsf{Inv}_4(\mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P_{a_{\mathsf{L}}}))$	(by inversion on (T-Θ_2))
$\mathcal{W}(\Theta_1)$ is acyclic	(by I.H.)

We assume that $\mathcal{W}(\Theta_1)$ is acyclic and proceed by analyzing what new arrows may arrise in the new configuration $\mathcal{W}(\Theta_1, \mathsf{proc}(a_{\text{L}}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P_{a_{\text{L}}}))$. Among these possible new arrows, we consider the ones that are necessary to introduce a cycle and show that those arrows cannot exists in a well-typed configuration.



We guide our proof by the schematic drawing of a configuration depicted above. In this drawing, the existing sub-configuration Θ_1 is shaded in gray and $\mathsf{proc}(a_{\text{L}}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P_{a_{\text{L}}})$ is represented by a black-rimmed white circle. For convenience, we refer to the latter as node $P_{a_{\text{L}}}$. We abstract subtrees by triangles, with the root node of the subtree depicted by a circle. The linear channels in $\Phi_1'; \Delta_1'$, by which the process is connected to the existing sub-configuration, are depicted as black edges. Without loss of generality, we focus on two children nodes, offering along the channels b_{L} and c_{L}. Similarly, without loss of generality, we abstract the part of the existing configuration that is not yet connected to node $P_{a_{\text{L}}}$ by a single tree, rooted at node $P_{d_{\text{L}}}$. Moreover, without loss of generality, we pictorially distinguish arrows that connect arbitrary nodes (nodes that are not directly connected by a linear channel) from arrows that connect a parent with a child node. The former, we depict as running between subtrees, whereas we depict the latter as running within a subtree. Our case analysis accounts for this distinction.

Since green arrows denote synchronization dependencies along linear channels, green arrows can only connect nodes that are directly connected by a linear channel. This means in terms of the schematic drawing above that *green* arrows can only go *along edges*, explicit ones or implicit ones within a subtree. Red arrows, on the other hand, denote acquire dependencies along shared channels, and as such *red* arrows can connect *any* two nodes in the configuration, whether or not there exists a linear path between them.

When we add node $P_{a_{\text{L}}}$ along with its linear channels b_{L} and c_{L} to the existing sub-configuration Θ_1, the following arrows can generally come about as a result:

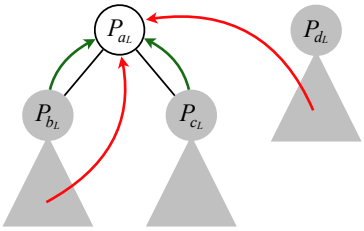
(a) *green* arrows along the linear channels b_{L} and c_{L};
(b) *red* arrows pointing *to* node $P_{a_{\text{L}}}$ from anywhere within the existing sub-configuration Θ_1, including its children nodes offering along b_{L} and c_{L};
(c) one *red* arrow pointing *from* node $P_{a_{\text{L}}}$ to anywhere in the existing sub-configuration Θ_1, if $P_{a_{\text{L}}}$ is an acquire and the to-be-acquired resource is currently held by the node pointed to.

We note that in the last case 2c, no red arrow arises if the to-be-acquired resource is available. We also remind ourselves of Corollary D.5, which says that there can exist at most one outgoing arrow per node.
Given that $\mathcal{W}(\Theta_1)$ is acyclic, the new arrows can only form a cycle with the existing arrows in $\mathcal{W}(\Theta_1)$, if there exist at least two new arrows that *pivot* around node P_{a_L}, such that one arrow leads *into* node P_{a_L} and one arrow goes *out* of node P_{a_L}. Given the kinds of possible new arrows 2a, 2c, and 2b, a cycle thus can only be created by any of the following *pivot pairs*:

- a green arrow going into node P_{a_L} and a green arrow going out of node P_{a_L},
- a red arrow going into node P_{a_L} and a green arrow going out of node P_{a_L},
- a green arrow going into node P_{a_L} and a red arrow going out of node P_{a_L},
- a red arrow going into node P_{a_L} and a red arrow going out of node P_{a_L}.

Next, we consider the circumstances under which the above pivot pairs can come about for node P_{a_L}. The determining factors are the direction of linear communication of node P_{a_L} and whether process term P_{a_L} is an acquire or not. These considerations give rise to the following three cases:

(a) $\Psi;\, \Gamma;\, \Phi_1';\, \Delta_1' \vdash P_{a_\text{L}} :: (a_\text{L} : A_\text{L}[\mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])$ synchronizes along its offering channel a_L

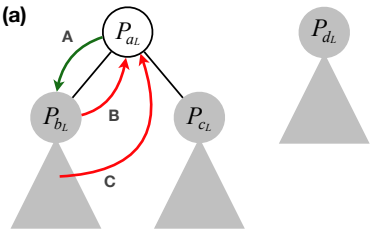


If node P_{a_L} synchronizes along its offering channel a_L, then any green arrows arising along the linear channels b_L and c_L must be going into node P_{a_L} because the processes P_{b_L} and P_{c_L} may be ready to synchronize along their offering channels b_L and c_L, respectively. Moreover, process term P_{a_L} cannot be an acquire, ruling out any red arrows going out of node P_{a_L}. The only new red arrows that can come about, are red arrows going into node P_{a_L}. As summarized by the schematic drawing above, no pivot pair of arrows can come about in this case, leaving the resulting configuration $\mathcal{W}(\Theta_1, \mathsf{proc}(a_\text{L}, \mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P_{a_\text{L}}))$ acyclic.

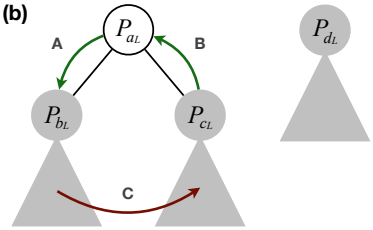
(b) $\Psi;\, \Gamma;\, \Phi_1';\, \Delta_1' \vdash P_{a_\text{L}} :: (a_\text{L} : A_\text{L}[\mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])$ synchronizes along any of its linear channel in $\Phi_1';\, \Delta_1'$

If node P_{a_L} synchronizes along any of its linear channels in $\Phi_1';\, \Delta_1'$, then a green arrow going out of node P_{a_L} can arise along that channel because the offering process may not be ready to synchronize. Without loss of generality, we assume that node P_{a_L} synchronizes along channel b_L. By

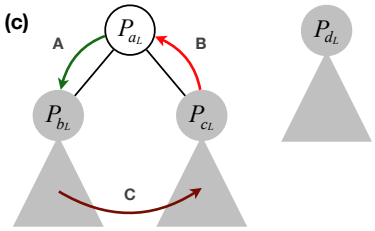
Corollary D.5 and by the fact that proces term P_{a_L} cannot be an acquire, we know furthermore that any remaining arrows connecting node P_{a_L} must be ingoing arrows. As a result, we get the pivot pairs depicted in drawings **(a)**, **(b)**, **(c)**, and **(d)**. We first show the pairs that are necessary to create a cycle within the same branch, then the ones that are necessary to create a cycle between two children subtrees, and lastly the ones that are necessary to create a cycle involving the part of the configuration that is not yet connected to node P_{a_L}. For each case, we prove that it cannot come about.



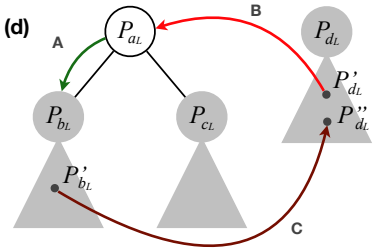
In case **(a)** we consider red arrows from node P_{b_L} or any of its direct or transitive children going into node P_{a_L}. The existence of any such red arrow, however, is ruled out by Lemma D.7. Consequently, the resulting configuration $\mathcal{W}(\Theta_1, \mathsf{proc}(a_\mathsf{L}, \mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P_{a_\mathsf{L}}))$ remains acyclic.



In case **(b)** we consider the possibility of a green ingoing arrow from any child of node P_{a_L} other than node P_{b_L}. Such an arrow, **B** in the drawing, comes about if that child node is ready to synchronize along its offering channel and thus waiting for node P_{a_L}. Although the arrows **A** and **B** form a pivot pair, there has to exist a red arrow **C** connecting the disjoint subtrees to form a cycle. Such an arrow **C** must already exist in the configuration Θ_1, hence the darker shade of red, and must either directly lead into node P_{c_L} or into a descendant of node P_{c_L}, such that there exists a path of arrows going upwards, one hop at a time, from one child node to its parent node until node P_{c_L} is reached. In the former case, we know by Lemma D.8 that such a red arrow **C** cannot exist. In the latter case, we know by Lemma D.7 that such a path can only exist of green arrows, ruling out the existence of the red arrow **C** by Lemma D.8. Consequently, the resulting configuration $\mathcal{W}(\Theta_1, \mathsf{proc}(a_\mathsf{L}, \mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P_{a_\mathsf{L}}))$ remains acyclic.



In case **(c)** we consider the possibility of a red ingoing arrow from any child of node P_{a_L} other than node P_{b_L}. The existence of such a red arrow, however, is ruled out by Lemma D.7. Consequently, the resulting configuration $\mathcal{W}(\Theta_1, \mathsf{proc}(a_L, \mathsf{w}_{a_1} \updownarrow^{\mathsf{w}_{a_3}}_{\mathsf{w}_{a_2}}, P_{a_L}))$ remains acyclic.

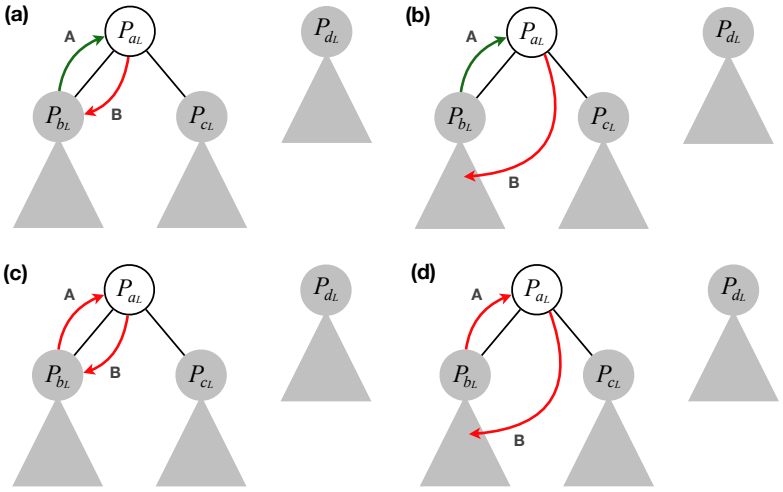


In case **(d)** we consider the possibility of a red ingoing arrow from any subtree rooted at a node P_{d_L}, which is not yet connected to the subtree roted at node P_{a_L}. Such an arrow, **B** in the drawing, comes about if node P_{a_L} holds a resource that the descendant P'_{d_L} of node P_{d_L} is acquiring. Although the arrows **A** and **B** form a pivot pair, there has to exist a red arrow **C** connecting the disjoint subtrees to form a cycle. Such an arrow **C** must already exist in the configuration Θ_1, hence the darker shade of red, and must either directly lead into node P'_{d_L} or into a descendant P''_{d_L} of node P'_{d_L}, such that there exists a path of arrows from P''_{d_L} to P'_{d_L}. We note that there could alternatively be a sequence of arrows leading from node P'_{b_L} via siblings of P''_{d_L} to P''_{d_L} itself, and we consider, without loss of generality, the last of any such red arrows.In the former case ($P'_{d_L} = P''_{d_L}$), we know by Definition D.2 that node P'_{d_L} holds a resource that a descendant P'_{b_L} of node P_{a_L} is acquiring and that node P'_{d_L} is acquiring a resource that node P_{a_L} holds. By $\mathsf{Inv}_3(\Theta_1, \mathsf{proc}(a_L, \mathsf{w}_{a_1} \updownarrow^{\mathsf{w}_{a_3}}_{\mathsf{w}_{a_2}}, P_{a_L}))$ we know moreover that the resource that node P'_{d_L} is acquiring must be at a higher world than the resource it already holds, which implies that the resource held by node P_{a_L} is at a higher world than the one its descendant P'_{b_L} is acquiring. However, this is impossible by $\mathsf{Inv}_3(\Theta_1, \mathsf{proc}(a_L, \mathsf{w}_{a_1} \updownarrow^{\mathsf{w}_{a_3}}_{\mathsf{w}_{a_2}}, P_{a_L}))$ and $\mathsf{Inv}_4(\Theta_1, \mathsf{proc}(a_L, \mathsf{w}_{a_1} \updownarrow^{\mathsf{w}_{a_3}}_{\mathsf{w}_{a_2}}, P_{a_L}))$, ruling out the simultaneous existence of the two arrows **B** and **C**, given the acyclicity of Ψ. In the latter case ($P'_{d_L} \neq P''_{d_L}$), we know by Lemma D.7 that such a path can only exist of green arrows, ruling out the existence of the red arrow **C** by Lemma D.8. Consequently, the resulting configuration $\mathcal{W}(\Theta_1, \mathsf{proc}(a_L, \mathsf{w}_{a_1} \updownarrow^{\mathsf{w}_{a_3}}_{\mathsf{w}_{a_2}}, P_{a_L}))$ remains acyclic.

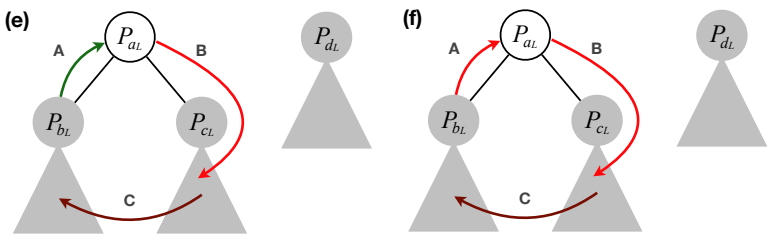
(c) $\Psi;\ \Gamma_1, b_{\mathsf{S}} : \uparrow^{\mathsf{S}}_{\mathsf{L}} B_{\mathsf{L}}[\mathsf{w}_{\mathsf{b}_1} \updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}];\ \Phi'_1;\ \Delta'_1 \vdash x_{\mathsf{L}} \leftarrow \mathsf{acquire}\, b_{\mathsf{S}}\, ;\, Q_{x_{\mathsf{L}}} :: (a_{\mathsf{L}} : A_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])$ executes an acquire

If node $P_{a_{\mathsf{L}}}$ executes an acquire, then a red arrow going out of node $P_{a_{\mathsf{L}}}$ to anywhere in the existing sub-configuration Θ_1 can result. Moreover, green arrows from any of node $P_{a_{\mathsf{L}}}$'s descendants going into node $P_{a_{\mathsf{L}}}$ can arise because those descendants might be ready to synchronize along their offering channels. And, lastly, as before, there may be ingoing red arrows into node $P_{a_{\mathsf{L}}}$ from anywhere within the existing sub-configuration Θ_1. As a result, we get the pivot pairs depicted in drawings **(a)**, **(b)**, **(c)**, **(d)**, **(e)**, **(f)**, **(g)**, **(h)**, **(i)**, **(j)**, **(k)**, **(l)**, and **(m)**. For each case, we prove that it cannot come about.

First, we consider the pivot pairs that are necessary to create a cycle within the same branch.

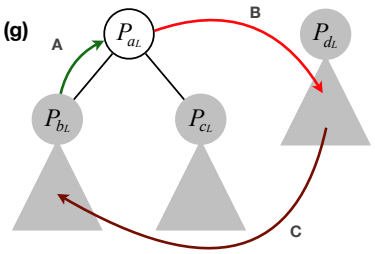


In cases **(a)**, **(b)**, **(c)**, and **(d)**, the existence of the red arrow **B** is ruled out by Lemma D.7. Moreover, in cases **(c)** and **(d)** also the existence of the red arrow **A** is ruled out by Lemma D.7. Consequently, the resulting configuration $\mathcal{W}(\Theta_1, \mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P_{a_{\mathsf{L}}}))$ remains acyclic.
Next, we consider the pivot pairs that are necessary to create a cycle between two children subtrees.

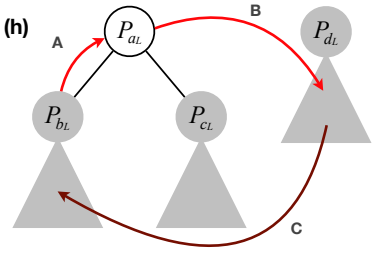


In cases **(e)** and **(f)**, the existence of the red arrow **B** is ruled out by Lemma D.7. Moreover, in case **(f)** also the existence of the red arrow **A** is ruled out by Lemma D.7. Consequently, the resulting configuration $\mathcal{W}(\Theta_1, \mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P_{a_{\mathsf{L}}}))$ remains acyclic.

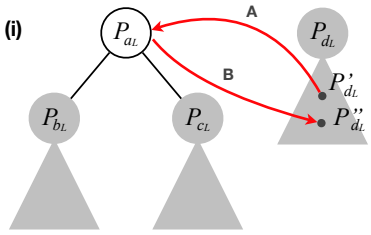
Next, we consider the pivot pairs that are necessary to create a cycle involving the part of the configuration that is not yet connected to node $P_{a_{\mathsf{L}}}$. Here, we consider cases where the cycle involves only one branch of that not yet connected sub-configuration. We extend to cycles that involve siblings afterwards.



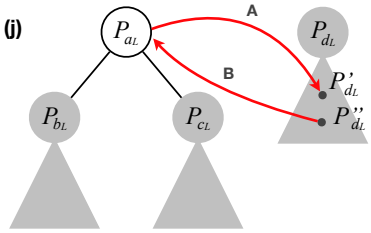
Although the arrows **A** and **B** form a pivot pair in case **(g)**, there has to exist a red arrow **C** connecting the disjoint subtrees to form a cycle. Such an arrow **C** must already exist in the configuration Θ_1 and must either directly lead into node $P_{b_{\mathsf{L}}}$ or into a descendant of node $P_{b_{\mathsf{L}}}$, such that there exists a path of arrows going upwards, one hop at a time, from one child node to its parent node until node $P_{b_{\mathsf{L}}}$ is reached. In the former case, we know by Lemma D.8 that such a red arrow **C** cannot exist. In the latter case, we know by Lemma D.7 that such a path can only exist of green arrows, ruling out the existence of the red arrow **C** by Lemma D.8. Consequently, the resulting configuration $\mathcal{W}(\Theta_1, \mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P_{a_{\mathsf{L}}}))$ remains acyclic.



In case **(h)**, the existence of the red arrow **A** is ruled out by Lemma D.7. Consequently, the resulting configuration $\mathcal{W}(\Theta_1, \mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P_{a_{\mathsf{L}}}))$ remains acyclic.



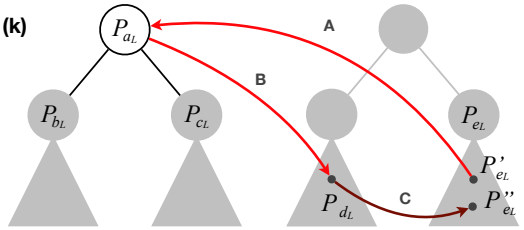
In case **(i)** we consider a pivot pair consisting of the red arrows **A** and **B**, indicating by Definition D.2 that node P'_{d_L} is acquiring a resource held by node P_{a_L} and that node P_{a_L} is acquiring a resource held by node P''_{d_L}. If $P'_{d_L} \neq P''_{d_L}$, then there exists a path of arrows from P''_{d_L} to P'_{d_L}, which, by Lemma D.7, can only exist of green arrows, ruling out the existence of the red arrow **B** by Lemma D.8. If $P'_{d_L} = P''_{d_L}$, we know by $\mathsf{Inv}_3(\Theta_1, \mathsf{proc}(a_L, \mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P_{a_L}))$ that the resource that node P'_{d_L} is acquiring must be at a higher world than the resource it already holds, which implies that the resource held by node P_{a_L} is at a higher world than the one it is acquiring. However, this implies that node P_{a_L} acquires the smaller resource after having acquired the larger one, which is impossible by $\mathsf{Inv}_3(\Theta_1, \mathsf{proc}(a_L, \mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P_{a_L}))$, ruling out the simultaneous existence of the two arrows **A** and **B**, given the acyclicity of Ψ. Consequently, the resulting configuration $\mathcal{W}(\Theta_1, \mathsf{proc}(a_L, \mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P_{a_L}))$ remains acyclic.



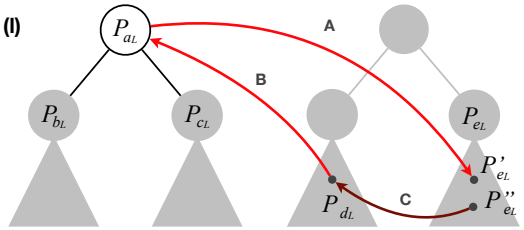
In case **(j)** we consider a pivot pair consisting of the red arrows **A** and **B**, indicating by Definition D.2 that node P_{a_L} is acquiring a resource held by node P'_{d_L} and that node P''_{d_L} is acquiring a resource held by node P_{a_L}. By $\mathsf{Inv}_3(\Theta_1, \mathsf{proc}(a_L, \mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P_{a_L}))$ we know moreover that the resource that node P_{a_L} is acquiring must be at a higher world than the resource it already holds, which implies that the resource held by node P'_{d_L} is at a higher world than the one node P''_{d_L} is acquiring. If $P'_{d_L} \neq P''_{d_L}$, then $\mathsf{Inv}_3(\Theta_1, \mathsf{proc}(a_L, \mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P_{a_L}))$ and $\mathsf{Inv}_4(\Theta_1, \mathsf{proc}(a_L, \mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P_{a_L}))$ make it impossible for a resource held by a parent to be at a higher world than the one being acquired by a descendant, ruling out the simultaneous existence of the two arrows **A** and **B**, given the acyclicity of Ψ. If $P'_{d_L} = P''_{d_L}$, node P'_{d_L} would be acquiring the smaller resource after having acquired the larger one, which is impossible by $\mathsf{Inv}_3(\Theta_1, \mathsf{proc}(a_L, \mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P_{a_L}))$, ruling out the simultaneous existence of the two arrows **A** and **B**, given

the acyclicity of Ψ. Consequently, the resulting configuration $\mathcal{W}(\Theta_1, \mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P_{a_{\mathsf{L}}}))$ remains acyclic.

Next, we consider the pivot pairs that are necessary to create a cycle involving the part of the configuration that is not yet connected to node $P_{a_{\mathsf{L}}}$. As opposed to the previous two cases, where the cycle involved only one branch of that not yet connected sub-configuration, here we investigate cases where the cycle involves siblings of that not yet connected sub-configuration.

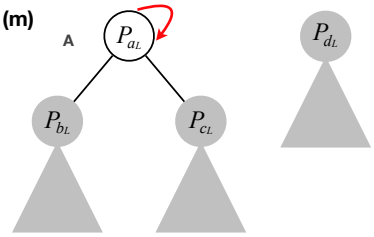


In case **(k)** we consider a pivot pair consisting of the red arrows **A** and **B**, connecting node $P_{a_{\mathsf{L}}}$ with nodes in a previously disjoint subtree. As opposed to case **(i)**, where the arrows **A** and **B** involve the same branch, here arrows **A** and **B** target sibling subtrees. As a result, there must exist a connecting arrow **C**, already existing in the configuration. By Definition D.2 we know that node $P_{a_{\mathsf{L}}}$ is acquiring a resource held by node $P_{d_{\mathsf{L}}}$, node $P_{d_{\mathsf{L}}}$ is acquiring a resource held by node $P''_{e_{\mathsf{L}}}$, and node node $P''_{e_{\mathsf{L}}}$ is acquiring a resource held by node $P_{a_{\mathsf{L}}}$. If $P'_{e_{\mathsf{L}}} \neq P''_{e_{\mathsf{L}}}$, then there exists a path of arrows from $P''_{e_{\mathsf{L}}}$ to $P'_{e_{\mathsf{L}}}$, which, by Lemma D.7, can only exist of green arrows, ruling out the existence of the red arrow **C** by Lemma D.8. If $P'_{e_{\mathsf{L}}} = P''_{e_{\mathsf{L}}}$, we know by $\mathsf{Inv}_3(\Theta_1, \mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P_{a_{\mathsf{L}}}))$ that the resource r_a that node $P'_{e_{\mathsf{L}}}$ is acquiring must be at a higher world than the resource r_e it already holds, that the resource r_d that node $P_{a_{\mathsf{L}}}$ is acquiring must be at a higher world than the resource r_a it already holds, and that the resource r_e that node $P_{d_{\mathsf{L}}}$ is acquiring must be at a higher world than the resource r_d it already holds, yielding the relation $r_e < r_a < r_d < r_e$. However, this contradicts acyclicity of Ψ, ruling out the simultaneous existence of the arrows **A**, **B**, and **C**. Consequently, the resulting configuration $\mathcal{W}(\Theta_1, \mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P_{a_{\mathsf{L}}}))$ remains acyclic.



In case **(l)** we consider a pivot pair consisting of the red arrows **A** and **B**, connecting node $P_{a_{\mathsf{L}}}$ with nodes in a previously disjoint subtree. As

opposed to case **(j)**, where the arrows **A** and **B** involve the same branch, here arrows **A** and **B** target sibling subtrees. As a result, there must exist a connecting arrow **C**, already existing in the configuration. By Definition D.2 we know that node P_{a_L} is acquiring a resource held by node P'_{e_L}, node P''_{e_L} is acquiring a resource held by node P_{d_L}, and node P_{d_L} is acquiring a resource held by node P_{a_L}. If $P'_{e_\mathsf{L}} \neq P''_{e_\mathsf{L}}$, then there exists a path of arrows from P''_{e_L} to P'_{e_L}, which, by Lemma D.7, can only exist of green arrows. Moreover, we know by $\mathsf{Inv}_3(\Theta_1, \mathsf{proc}(a_\mathsf{L}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P_{a_\mathsf{L}}))$ that the resource r_e that node P_{a_L} is acquiring must be at a higher world than the resource r_a it already holds and that the resource r_a that node P_{d_L} is acquiring must be at a higher world than the resource r_d it already holds, yielding the relation $r_d < r_a < r_e$. However, this would imply that the resource r_e held by node P'_{e_L} is greater than the resource r_d being acquired by node P'_{e_L}, which is impossible by $\mathsf{Inv}_3(\Theta_1, \mathsf{proc}(a_\mathsf{L}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P_{a_\mathsf{L}}))$ and $\mathsf{Inv}_4(\Theta_1, \mathsf{proc}(a_\mathsf{L}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P_{a_\mathsf{L}}))$. Thus the arrows **A**, **B**, and **C** cannot exist simultaneously. If $P'_{e_\mathsf{L}} = P''_{e_\mathsf{L}}$, we know by $\mathsf{Inv}_3(\Theta_1, \mathsf{proc}(a_\mathsf{L}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P_{a_\mathsf{L}}))$ that the resource r_e that node P_{a_L} is acquiring must be at a higher world than the resource r_a it already holds, that the resource r_d that node P'_{e_L} is acquiring must be at a higher world than the resource r_e it already holds, and that the resource r_a that node P_{d_L} is acquiring must be at a higher world than the resource r_d it already holds, yielding the relation $r_a < r_e < r_d < r_a$. However, this contradicts acyclicity of Ψ, ruling out the simultaneous existence of the arrows **A**, **B**, and **C**. Consequently, the resulting configuration $\mathcal{W}(\Theta_1, \mathsf{proc}(a_\mathsf{L}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P_{a_\mathsf{L}}))$ remains acyclic. Lastly, we consider a corner-case pivot pair that arises from a self-loop.



In case **(m)** we consider a pivot pair consisting of a red self-loop arrow, indicating that node P_{a_L} is acquiring a resources that it already holds. However, this is impossible by $\mathsf{Inv}_3(\Theta_1, \mathsf{proc}(a_\mathsf{L}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P_{a_\mathsf{L}}))$, ruling out the existence of arrow **A**, given the acyclicity of Ψ. Consequently, the resulting configuration $\mathcal{W}(\Theta_1, \mathsf{proc}(a_\mathsf{L}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P_{a_\mathsf{L}}))$ remains acyclic.

□

Lemma D.10 (Availability of shared provider). *Writing $P\langle a_\mathsf{S}\rangle$ for a process term P with an occurrence of a channel a_S, the following holds: For a well-formed top-level configuration $\Psi; \Gamma \vDash \Lambda; \Theta :: (\Gamma; c_\mathsf{L} : \mathbf{1}[\mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}])$ and for any $\mathsf{proc}(a_\mathsf{L}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, Q_{x_\mathsf{L}}) \in \Theta$ such that $Q_{x_\mathsf{L}} = x_\mathsf{L} \leftarrow \mathsf{acquire}\ d_\mathsf{S}; Q'_{x_\mathsf{L}}$, if there does*

not exist $\mathsf{proc}(b_{\mathsf{L}}, \mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}, P'\langle d_{\mathsf{L}}\rangle)$ *in* Θ*, then there exists a* $\mathsf{proc}(d_{\mathsf{S}}, \mathsf{w}_{\mathsf{d}_1}\updownarrow^{\mathsf{w}_{\mathsf{d}_3}}_{\mathsf{w}_{\mathsf{d}_2}}, P_{a_{\mathsf{S}}})$ *in* Λ*.*

Lemma D.11 (Existence of client of linear provider). *Writing* $P\langle a_{\mathsf{L}}\rangle$ *for a process term* P *with an occurrence of a channel* a_{L}*, the following holds: For a well-formed top-level configuration* $\Psi; \Gamma \vDash \Lambda; \Theta :: (\Gamma; c_{\mathsf{L}} : \mathbf{1}[\mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}])$ *and for any* $\mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, Q) \in \Theta$ *such that* $a_{\mathsf{L}} \neq c_{\mathsf{L}}$*, there exists a client* $\mathsf{proc}(b_{\mathsf{L}}, \mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}, P\langle a_{\mathsf{L}}\rangle)$ *in* Θ*.*

Lemma D.12 (Existence of linear provider). *Writing* $Q\langle a_{\mathsf{L}}\rangle$ *for a process term* Q *with an occurrence of a channel* a_{L}*, the following holds: In a well-formed and well-typed configuration* $\Psi;\ \Gamma \vDash \Theta_1, \mathsf{proc}(c_{\mathsf{L}}, \mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}, Q\langle a_{\mathsf{L}}\rangle) :: \Phi, \Delta$*, there exists exactly one* $\mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P_{a_{\mathsf{L}}})$*, for some* $P_{a_{\mathsf{L}}}$*, in* Θ_1*.*

D.3 Theorem

Theorem D.13 (Progress). *If* $\Psi; \Gamma \vDash \Lambda; \Theta :: (\Gamma; c_{\mathsf{L}} : \mathbf{1}[\mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}])$*, then either*

- $\Lambda \longrightarrow \Lambda'$*, for some* Λ'*, or*
- Λ *is poised and*
  - *if* $|\Theta| = 1$*, then either* $\Lambda; \Theta \longrightarrow \Lambda'; \Theta'$*, for some* Λ' *and* Θ'*, or* Θ *is poised, or*
  - *if* $|\Theta| > 1$*, then* $\Lambda; \Theta \longrightarrow \Lambda'; \Theta'$*, for some* Λ' *and* Θ'*.*

Proof.

$\Psi; \Gamma \vDash \Lambda; \Theta :: (\Gamma; c_{\mathsf{L}} : \mathbf{1}[\mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}])$ (by assumption)

$\Psi; \Gamma \vDash \Lambda :: \Gamma$ and $\Psi; \Gamma \vDash \Theta :: (c_{\mathsf{L}} : \mathbf{1}[\mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}])$ (by inversion on (T-Ω))

$\mathcal{W}(\Theta)$ is acyclic (by Lemma D.9)

We first show that either $\Lambda \longrightarrow \Lambda'$, for some Λ', or that Λ is poised. We proceed by induction on $\Psi; \Gamma \vDash \Lambda :: \Gamma_1$, where $\Gamma_1 \subseteq \Gamma$:

1. $\Psi; \Gamma \vDash (\cdot) :: (\cdot)$

   $(\cdot)$ is poised (by Definition D.1)

2. $\Psi; \Gamma \vDash \mathsf{proc}(a_{\mathsf{S}}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P_{a_{\mathsf{S}}}) :: (a_{\mathsf{S}} : \uparrow^{\mathsf{S}}_{\mathsf{L}} A_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])$*, for some* a_{S}*,* $\mathsf{w}_{\mathsf{a}_1}$*,* $\mathsf{w}_{\mathsf{a}_2}$*,* $\mathsf{w}_{\mathsf{a}_3}$*,* $P_{a_{\mathsf{S}}}$*, and* A_{L}

   $\Psi; \Gamma \vdash P_{a_{\mathsf{S}}} :: (a_{\mathsf{S}} : \uparrow^{\mathsf{S}}_{\mathsf{L}} A_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])$ (by inversion on (T-Λ_2))

   We proceed by case analysis of $\Psi; \Gamma \vdash P_{a_{\mathsf{S}}} :: (a_{\mathsf{S}} : \uparrow^{\mathsf{S}}_{\mathsf{L}} A_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])$:

   (a) $\Psi; \Gamma \vdash x_{\mathsf{L}} \leftarrow \mathsf{accept}\, a_{\mathsf{S}}\,; P_{x_{\mathsf{L}}} :: (a_{\mathsf{S}} : \uparrow^{\mathsf{S}}_{\mathsf{L}} A_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])$
   $\mathsf{proc}(a_{\mathsf{S}}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, x_{\mathsf{L}} \leftarrow \mathsf{accept}\, a_{\mathsf{S}}\,; P_{x_{\mathsf{L}}})$ is poised (by Definition D.1)

   (b) $\Psi; \Gamma \vdash \mathsf{fwd}\ a_{\mathsf{S}}\ b_{\mathsf{S}} :: (a_{\mathsf{S}} : \uparrow^{\mathsf{S}}_{\mathsf{L}} A_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])$*, for some* $(b_{\mathsf{S}} : \uparrow^{\mathsf{S}}_{\mathsf{L}} A_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}]) \in \Gamma$

$\mathsf{proc}(a_\mathsf{S}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, \mathsf{fwd}\ a_\mathsf{S}\ b_\mathsf{S}) \longrightarrow \mathsf{unavail}(a_\mathsf{S}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}), a_\mathsf{S} = b_\mathsf{S}$ (by D-Id$_\mathsf{S}$)

(c) $\Psi; \Gamma \vdash \mathsf{w} \leftarrow \mathsf{new\_world}; Q_\mathsf{w} :: (a_\mathsf{S} : \uparrow^\mathsf{S}_\mathsf{L} A_\mathsf{L}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])$
$\mathsf{proc}(a_\mathsf{S}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, \mathsf{w} \leftarrow \mathsf{new\_world}; Q_\mathsf{w}) \longrightarrow \mathsf{proc}(a_\mathsf{S}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, Q_\mathsf{w})$ (w *fresh*) (by D-New)

(d) $\Psi; \Gamma \vdash \mathsf{w} < \mathsf{w}'; Q :: (a_\mathsf{S} : \uparrow^\mathsf{S}_\mathsf{L} A_\mathsf{L}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])$
$\mathsf{proc}(a_\mathsf{S}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, \mathsf{w} < \mathsf{w}'; Q) \longrightarrow \mathsf{proc}(a_\mathsf{S}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, Q)$ (by D-Ord)

(e) $\Psi; \Gamma \vdash x_\mathsf{S} : A_\mathsf{S}[\mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}] \leftarrow X_\mathsf{S} \leftarrow \overline{d_\mathsf{S}}; Q_{x_\mathsf{S}} :: (a_\mathsf{S} : \uparrow^\mathsf{S}_\mathsf{L} A_\mathsf{L}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])$
$\mathsf{proc}(a_\mathsf{S}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, x_\mathsf{S} : A_\mathsf{S}[\mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}] \leftarrow X_\mathsf{S} \leftarrow \overline{d_\mathsf{S}}; Q_{x_\mathsf{S}}),$ (by D-Spawn$_\mathsf{SS}$)
$\mathsf{!def}(\Psi' \vdash x'_\mathsf{S} : A'_\mathsf{S}[\delta_j\updownarrow^{\delta_n}_{\delta_k}] \leftarrow X_\mathsf{S} \leftarrow \Gamma' = P_{x'_\mathsf{S}, \mathsf{dom}(\Gamma'), \Psi''})$
$\longrightarrow \mathsf{proc}(a_\mathsf{S}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, [b_\mathsf{S}/x_\mathsf{S}]Q_{x_\mathsf{S}}), \mathsf{proc}(b_\mathsf{S}, \mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}, [b_\mathsf{S}/x'_\mathsf{S}, \overline{d_\mathsf{S}}/\mathsf{dom}(\Gamma')]\hat{\gamma}(P_{x'_\mathsf{L}, \mathsf{dom}(\Gamma'), \Psi''}))$
(b *fresh*)

3. $\Psi; \Gamma \vDash \mathsf{unavail}(a_\mathsf{S}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}) :: (a_\mathsf{S} : \hat{A}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])$, *for some* a_S, $\mathsf{w}_{\mathsf{a}_1}$, $\mathsf{w}_{\mathsf{a}_2}$, $\mathsf{w}_{\mathsf{a}_3}$, *and* $\hat{A}$

$\mathsf{unavail}(a_\mathsf{S}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}})$ is poised (by Definition D.1)

4. $\Psi; \Gamma \vDash \Lambda_1, \Lambda_2 :: \Gamma_1, \Gamma_2$, *for some* Λ_1, Λ_2, Γ_1, *and* Γ_2

Either $\Lambda_1 \longrightarrow \Lambda'_1$, for some Λ'_1, or Λ_1 is poised, or $\Lambda_2 \longrightarrow \Lambda'_2$, for some Λ'_2, or Λ_2 is poised. (by I.H.)

(a) $\Lambda_1 \longrightarrow \Lambda'_1$, *for some* Λ'_1, and Λ_2 is poised
$\Lambda_1, \Lambda_2 \longrightarrow \Lambda'_1, \Lambda_2$

(b) $\Lambda_1 \longrightarrow \Lambda'_1$, *for some* Λ'_1, and $\Lambda_2 \longrightarrow \Lambda'_2$, *for some* Λ'_2
$\Lambda_1, \Lambda_2 \longrightarrow \Lambda'_1, \Lambda'_2$

(c) Λ_1 is poised and $\Lambda_2 \longrightarrow \Lambda'_2$, *for some* Λ'_2
$\Lambda_1, \Lambda_2 \longrightarrow \Lambda_1, \Lambda'_2$

(d) Λ_1 is poised and Λ_2 is poised
Λ_1, Λ_2 is poised

Having proved that either $\Lambda \longrightarrow \Lambda'$, for some Λ', or that Λ is poised, we assume that Λ is poised and show that either $\Lambda; \Theta \longrightarrow \Lambda'; \Theta'$, for some Λ' and Θ', or Θ is poised, if $|\Theta| = 1$, or that $\Lambda; \Theta \longrightarrow \Lambda'; \Theta'$, for some Λ' and Θ', if $|\Theta| > 1$.

1. $|\Theta| = 1$
We proceed by case analysis of $\Psi; \Gamma \vDash \Theta :: (c_\mathsf{L} : \mathbf{1}[\mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}])$:

(a) $\Psi; \Gamma \vDash \Theta_1, \mathsf{proc}(c_\mathsf{L}, \mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}, P_{c_\mathsf{L}}) :: (\Phi_1, \Delta_1, c_\mathsf{L} : \mathbf{1}[\mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}])$, *for some* P_{c_L} *and such that* $\Theta_1 = (\cdot)$, $\Phi_1 = (\cdot)$, *and* $\Delta_1 = (\cdot)$
$\Psi; \Gamma; \cdot; \cdot \vdash P_{c_\mathsf{L}} :: (c_\mathsf{L} : \mathbf{1}[\mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}])$ (by inversion on (T-Θ_2))
We proceed by case analysis of $\Psi; \Gamma; \cdot; \cdot \vdash P_{c_\mathsf{L}} :: (c_\mathsf{L} : \mathbf{1}[\mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}])$:

i. $\Psi; \Gamma; \cdot; \cdot \vdash x_\mathsf{L} \leftarrow \mathsf{acquire}\ a_\mathsf{S} :: (c_\mathsf{L} : \mathbf{1}[\mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}])$, *for some* $(a_\mathsf{S} : \uparrow^\mathsf{S}_\mathsf{L} A_\mathsf{L}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}]) \in \Gamma$

$\mathsf{proc}(a_\mathsf{S}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, Q_{a_\mathsf{S}})$ in Λ, for some Q_{a_S} (by Lemma D.10)
$Q_{a_\mathsf{S}} = x_\mathsf{L} \leftarrow \mathsf{accept}\ a_\mathsf{S}; Q'_{x_\mathsf{L}}$ (since Λ is poised and by well-formedness of Λ, Θ)
$\Lambda_1, \mathsf{proc}(a_\mathsf{S}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, x_\mathsf{L} \leftarrow \mathsf{accept}\ a_\mathsf{S}\,; Q'_{x_\mathsf{L}});\ \Theta_1, \mathsf{proc}(c_\mathsf{L}, \mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}, x_\mathsf{L} \leftarrow \mathsf{acquire}\ a_\mathsf{S}\,; P_{x_\mathsf{L}})$ (by D-$\uparrow^\mathsf{S}_\mathsf{L}$)
$\longrightarrow \Lambda_1, \mathsf{unavail}(a_\mathsf{S}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}); \Theta_1, \mathsf{proc}(a_\mathsf{L}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, [a_\mathsf{L}/x_\mathsf{L}]\, Q'_{x_\mathsf{L}}), \mathsf{proc}(c_\mathsf{L}, \mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}, [a_\mathsf{L}/x_\mathsf{L}]\, P_{x_\mathsf{L}})$
where $\Lambda = \Lambda_1, \mathsf{proc}(a_\mathsf{S}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, x_\mathsf{L} \leftarrow \mathsf{accept}\ a_\mathsf{S}\,; Q'_{x_\mathsf{L}})$

ii. $\Psi; \Gamma; \cdot; \cdot \vdash \mathsf{close}\ c_\mathsf{L} :: (c_\mathsf{L} : \mathbf{1}[\mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}])$
$\Theta_1, \mathsf{proc}(c_\mathsf{L}, \mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}, \mathsf{close}\ c_\mathsf{L})$ is poised (by Definition D.1)

iii. $\Psi; \Gamma; \cdot; \cdot \vdash \mathsf{w} \leftarrow \mathsf{new\_world}; P'_\mathsf{w} :: (c_\mathsf{L} : \mathbf{1}[\mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}])$
$\Lambda; \Theta_1, \mathsf{proc}(c, \mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}, \mathsf{w} \leftarrow \mathsf{new\_world}; P'_{\mathsf{w}'})$ (by D-New)
$\longrightarrow \Lambda; \Theta_1, \mathsf{proc}(c, \mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}, P'_{\mathsf{w}'})$ (w *fresh*)

iv. $\Psi; \Gamma; \cdot; \cdot \vdash \mathsf{w} < \mathsf{w}'; P'_\mathsf{w} :: (c_\mathsf{L} : \mathbf{1}[\mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}])$
$\Lambda; \Theta_1, \mathsf{proc}(c, \mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}, \mathsf{w} < \mathsf{w}'; P') \longrightarrow \Lambda; \Theta_1, \mathsf{proc}(c, \mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}, P')$ (by D-Ord)

v. $\Psi; \Gamma; \cdot; \cdot \vdash x_\mathsf{L} : A_\mathsf{L}[\mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}] \leftarrow X_\mathsf{L} \leftarrow \overline{c'_\mathsf{L}}, \overline{\tilde{c'}_\mathsf{L}}, \overline{d_\mathsf{S}};\ P'_{x_\mathsf{L}} :: (c_\mathsf{L} : \mathbf{1}[\mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}])$
$\Lambda; \Theta_1, \mathsf{proc}(c_\mathsf{L}, \mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}, x_\mathsf{L} : A_\mathsf{L}[\mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}] \leftarrow X_\mathsf{L} \leftarrow \overline{c'_\mathsf{L}}, \overline{\tilde{c'}_\mathsf{L}}, \overline{d_\mathsf{S}};\ P'_{x_\mathsf{L}})$ (by D-SpawnLL)
$\longrightarrow \Lambda, \mathsf{unavail}(b_\mathsf{S}, \mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}); \Theta_1, \mathsf{proc}(c_\mathsf{L}, \mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}, [b_\mathsf{L}/x_\mathsf{L}]P'_{x_\mathsf{L}}),$
$\mathsf{proc}(b_\mathsf{L}, \mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}, [b_\mathsf{L}/x'_\mathsf{L}, \overline{c'_\mathsf{L}}/\mathsf{dom}(\Delta'), \overline{\tilde{c'}_\mathsf{L}}/\mathsf{dom}(\Phi'), \overline{d_\mathsf{S}}/\mathsf{dom}(\Gamma')]\hat{\gamma}(Q_{x'_\mathsf{L},\mathsf{dom}(\Delta'),\mathsf{dom}(\Phi'),\mathsf{dom}(\Gamma'),\Psi''}))$
where b *fresh and* $!\mathsf{def}(\Psi' \vdash x'_\mathsf{L} : A'_\mathsf{L}[\delta_j\updownarrow^{\delta_n}_{\delta_k}] \leftarrow X_\mathsf{L} \leftarrow \Delta', \Phi', \Gamma' = Q_{x'_\mathsf{L},\mathsf{dom}(\Delta'),\mathsf{dom}(\Phi'),\mathsf{dom}(\Gamma'),\Psi''})$

vi. $\Psi; \Gamma; \cdot; \cdot \vdash x_\mathsf{L} \leftarrow X_\mathsf{L} \leftarrow \overline{b_\mathsf{L}}, \overline{\tilde{b}_\mathsf{L}}, \overline{d_\mathsf{S}} :: (c_\mathsf{L} : \mathbf{1}[\mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}])$
$\Lambda; \Theta_1, \mathsf{proc}(c_\mathsf{L}, \mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}, x_\mathsf{L} \leftarrow X_\mathsf{L} \leftarrow \overline{b_\mathsf{L}}, \overline{\tilde{b}_\mathsf{L}}, \overline{d_\mathsf{S}})$ (by D-TailL)
$\longrightarrow \Lambda; \Theta_1, \mathsf{proc}(c_\mathsf{L}, \mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}, [c_\mathsf{L}/x'_\mathsf{L}, \overline{b_\mathsf{L}}/\mathsf{dom}(\Delta'), \overline{\tilde{b}_\mathsf{L}}/\mathsf{dom}(\Phi'), \overline{d_\mathsf{S}}/\mathsf{dom}(\Gamma')]\hat{\gamma}(Q_{x'_\mathsf{L},\mathsf{dom}(\Delta'),\mathsf{dom}(\Phi'),\mathsf{dom}(\Gamma'}$
where $!\mathsf{def}(\Psi' \vdash x'_\mathsf{L} : A'_\mathsf{L}[\delta_j\updownarrow^{\delta_n}_{\delta_k}] \leftarrow X_\mathsf{L} \leftarrow \Delta', \Phi', \Gamma' = Q_{x'_\mathsf{L},\mathsf{dom}(\Delta'),\mathsf{dom}(\Phi'),\mathsf{dom}(\Gamma'),\Psi''})$

vii. $\Psi; \Gamma; \cdot; \cdot \vdash x_\mathsf{S} : A_\mathsf{S}[\mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}] \leftarrow X_\mathsf{S} \leftarrow \overline{d_\mathsf{S}};\ P'_{x_\mathsf{S}} :: (c_\mathsf{L} : \mathbf{1}[\mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}])$
$\Lambda; \Theta_1, \mathsf{proc}(c_\mathsf{L}, \mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}, x_\mathsf{S} : A_\mathsf{S}[\mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}] \leftarrow X_\mathsf{S} \leftarrow \overline{d_\mathsf{S}};\ P'_{x_\mathsf{S}})$ (by D-SpawnLS)
$\longrightarrow \Lambda, \mathsf{proc}(b_\mathsf{S}, \mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}, [b_\mathsf{S}/x'_\mathsf{S}, \overline{d_\mathsf{S}}/\mathsf{dom}(\Gamma')]\hat{\gamma}(Q_{x'_\mathsf{S},\mathsf{dom}(\Gamma'),\Psi''})); \Theta_1, \mathsf{proc}(c_\mathsf{L}, \mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}, [b_\mathsf{S}/x_\mathsf{S}]P'_{x_\mathsf{S}})$
where b *fresh and* $!\mathsf{def}(\Psi' \vdash x'_\mathsf{S} : A'_\mathsf{S}[\delta_j\updownarrow^{\delta_n}_{\delta_k}] \leftarrow X_\mathsf{S} \leftarrow \Gamma' = Q_{x'_\mathsf{S},\mathsf{dom}(\Gamma'),\Psi''})$

2. $|\Theta| > 1$

Since $\mathcal{W}(\Theta)$ is acyclic, there must exist at least one $\mathsf{proc}(a_\mathsf{L}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P_{a_\mathsf{L}}) \notin \mathsf{dom}(\mathcal{W}(\Theta))$. If $\mathsf{proc}(a_\mathsf{L}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P_{a_\mathsf{L}})$ communicates along its offering channel, it cannot be the main process, because otherwise we had $|\Theta| = 1$, and thus we know by Lemma D.11 that there exists a client process. By Definition D.4 we know furthermore that this client must be ready to synchronize, because otherwise $\mathsf{proc}(a_\mathsf{L}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P_{a_\mathsf{L}}) \in \mathsf{dom}(\mathcal{W}(\Theta))$. As a result, we know by the dynamics that $\Lambda, \Theta \longrightarrow \Lambda, \Theta'$. If $\mathsf{proc}(a_\mathsf{L}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P_{a_\mathsf{L}})$ communicates along any linear channel that it uses, we know by Lemma D.12 that a corresponding providing process exists and by Definition D.4 that this provider must be ready to synchronize, because otherwise $\mathsf{proc}(a_\mathsf{L}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P_{a_\mathsf{L}}) \in \mathsf{dom}(\mathcal{W}(\Theta))$. As a result, we know by the dynamics that $\Lambda, \Theta \longrightarrow \Lambda, \Theta'$. If $\mathsf{proc}(a_\mathsf{L}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P_{a_\mathsf{L}})$ executes an internal action (i.e., forward, spawn, tail-call, and world or order creation), then we know by rules D-Id$_\mathsf{L}$, D-SpawnLL, D-TailL, D-SpawnLS, D-New, and D-Ord that $\Lambda, \Theta \longrightarrow \Lambda', \Theta'$. Lastly,

if $\mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P_{a_{\mathsf{L}}})$ executes an acquire, we know by Definition D.4 that no other linear process currently holds that resource, because otherwise $\mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P_{a_{\mathsf{L}}}) \in \mathsf{dom}(\mathcal{W}(\Theta))$. By Lemma D.10 and by the fact that Λ is poised and well-formed we know furthermore that there exists an accepting provider process in Λ. As a result, we know by the D-$\uparrow^{\mathsf{S}}_{\mathsf{L}}$ that $\Lambda, \Theta \longrightarrow \Lambda', \Theta'$.

□

E Preservation: Critical Cases

E.1 Definitions

Definition E.1 (Concretization of Γ). *A concretization of Γ is a partial order on structural contexts Γ and is inductively defined by the following rules, relying on the partial order $\uparrow^{\mathsf{S}}_{\mathsf{L}} C_{\mathsf{L}} \leq \top$, for any C_{L}:*

$$\frac{}{\Gamma \lhd (\cdot)} \lhd_{\Gamma_1} \qquad \frac{\hat{A} \leq \hat{B} \qquad \Gamma' \lhd \Gamma}{\Gamma', a_{\mathsf{S}} : \hat{A}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}] \lhd \Gamma, a_{\mathsf{S}} : \hat{B}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}]} \lhd_{\Gamma_2}$$

E.2 Lemmas and Corollaries

Lemma E.2 (Process term substitution). *Given the partial order $\uparrow^{\mathsf{S}}_{\mathsf{L}} C_{\mathsf{L}} \leq \top$, for any C_{L}, and using $|\Psi|$ to denote the field of Ψ, i.e., the union of its domain and range, the following substitutions are type-preserving and thus admissible:*

1. *If $\Psi; \Gamma; \Phi, \Delta \vdash P :: (x_{\mathsf{L}} : A_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])$, then, for any fresh $a_{\mathsf{L}} : A_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}]$, $\Psi; \Gamma; \Phi; \Delta \vdash [a_{\mathsf{L}}/x_{\mathsf{L}}]\, P :: (a_{\mathsf{L}} : A_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])$.*
2. *If $\Psi; \Gamma; \Phi, \Delta \vdash P :: (x_{\mathsf{L}} : A_{\mathsf{L}}[\delta_{a_1}\updownarrow^{\delta_{a_3}}_{\delta_{a_2}}])$, and, for any fresh $a_{\mathsf{L}} : A'_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}]$ and a Ψ' and γ such that $\mathsf{w}_{\mathsf{a}_1}, \mathsf{w}_{\mathsf{a}_2}, \mathsf{w}_{\mathsf{a}_3} \in |\Psi'|$ and $\Psi'^{+} \vdash \hat{\gamma}(\Psi)$ and $A'_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}] = \hat{\gamma}(A_{\mathsf{L}}[\delta_{a_1}\updownarrow^{\delta_{a_3}}_{\delta_{a_2}}])$ and and $\Psi \vdash A_{\mathsf{L}}[\delta_{a_1}\updownarrow^{\delta_{a_3}}_{\delta_{a_2}}]\ \mathsf{type}$, then $\hat{\gamma}(\Psi); \hat{\gamma}(\Gamma); \hat{\gamma}(\Phi); \hat{\gamma}(\Delta) \vdash [a_{\mathsf{L}}/x_{\mathsf{L}}]\, \hat{\gamma}(P) :: (a_{\mathsf{L}} : A'_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])$.*
3. *If $\Psi; \Gamma; \Phi, y_{\mathsf{L}} : B_{\mathsf{L}}[\mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}]; \Delta \vdash P :: (a_{\mathsf{L}} : A_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])$, then, for any fresh $b_{\mathsf{L}} : B_{\mathsf{L}}[\mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}]$, $\Psi; \Gamma; \Phi, b_{\mathsf{L}} : B_{\mathsf{L}}[\mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}]; \Delta \vdash [b_{\mathsf{L}}/y_{\mathsf{L}}]\, P :: (a_{\mathsf{L}} : A_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])$.*
4. *If $\Psi; \Gamma; \Phi, y_{\mathsf{L}} : B_{\mathsf{L}}[\delta_{b_1}\updownarrow^{\delta_{b_3}}_{\delta_{b_2}}]; \Delta \vdash P :: (a_{\mathsf{L}} : A_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])$, and, for any fresh $b_{\mathsf{L}} : B'_{\mathsf{L}}[\mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}]$ and a Ψ' and γ such that $\mathsf{w}_{\mathsf{b}_1}, \mathsf{w}_{\mathsf{b}_2}, \mathsf{w}_{\mathsf{b}_3} \in |\Psi'|$ and $\Psi'^{+} \vdash \hat{\gamma}(\Psi)$ and $B'_{\mathsf{L}}[\mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}] = \hat{\gamma}(B_{\mathsf{L}}[\delta_{b_1}\updownarrow^{\delta_{b_3}}_{\delta_{b_2}}])$ and $\Psi \vdash B_{\mathsf{L}}[\delta_{b_1}\updownarrow^{\delta_{b_3}}_{\delta_{b_2}}]\ \mathsf{type}$, then $\hat{\gamma}(\Psi); \hat{\gamma}(\Gamma); \hat{\gamma}(\Phi), b_{\mathsf{L}} : B'_{\mathsf{L}}[\mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}]; \hat{\gamma}(\Delta) \vdash [b_{\mathsf{L}}/y_{\mathsf{L}}]\, \hat{\gamma}(P) :: (a_{\mathsf{L}} : A_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])$.*
5. *If $\Psi; \Gamma; \Phi; \Delta, y_{\mathsf{L}} : B_{\mathsf{L}}[\mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}] \vdash P :: (a_{\mathsf{L}} : A_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])$, then, for any fresh $b_{\mathsf{L}} : B_{\mathsf{L}}[\mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}]$, $\Psi; \Gamma; \Phi; \Delta, b_{\mathsf{L}} : B_{\mathsf{L}}[\mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}] \vdash [b_{\mathsf{L}}/y_{\mathsf{L}}]\, P :: (a_{\mathsf{L}} : A_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])$.*
6. *If $\Psi; \Gamma; \Phi; \Delta, y_{\mathsf{L}} : B_{\mathsf{L}}[\delta_{b_1}\updownarrow^{\delta_{b_3}}_{\delta_{b_2}}] \vdash P :: (a_{\mathsf{L}} : A_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])$, and, for any fresh $b_{\mathsf{L}} : B'_{\mathsf{L}}[\mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}]$ and a Ψ' and γ such that $\mathsf{w}_{\mathsf{b}_1}, \mathsf{w}_{\mathsf{b}_2}, \mathsf{w}_{\mathsf{b}_3} \in |\Psi'|$ and $\Psi'^{+} \vdash \hat{\gamma}(\Psi)$ and $B'_{\mathsf{L}}[\mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}] = \hat{\gamma}(B_{\mathsf{L}}[\delta_{b_1}\updownarrow^{\delta_{b_3}}_{\delta_{b_2}}])$ and $\Psi \vdash B_{\mathsf{L}}[\delta_{b_1}\updownarrow^{\delta_{b_3}}_{\delta_{b_2}}]\ \mathsf{type}$, then $\hat{\gamma}(\Psi); \hat{\gamma}(\Gamma); \hat{\gamma}(\Phi); \hat{\gamma}(\Delta), b_{\mathsf{L}} : B'_{\mathsf{L}}[\mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}] \vdash [b_{\mathsf{L}}/y_{\mathsf{L}}]\, \hat{\gamma}(P) :: (a_{\mathsf{L}} : A_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])$.*

7. *If* $\Gamma, y_{\mathsf{s}} : \hat{B}[\mathsf{w}_{\mathsf{b}_1} \updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}]; \Delta \vdash P :: (a_{\mathsf{L}} : A_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])$*, then, for any fresh* $y'_{\mathsf{s}} : \hat{C}[\mathsf{w}_{\mathsf{b}_1} \updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}]$ *such that* $\hat{C} \leq \hat{B}$*,* $\Gamma, y'_{\mathsf{s}} : \hat{C}[\mathsf{w}_{\mathsf{b}_1} \updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}]; \Delta \vdash [y'_{\mathsf{s}}/y_{\mathsf{s}}]\, P :: (a_{\mathsf{L}} : A_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])$*.*
8. *If* $\Gamma, y_{\mathsf{s}} : \hat{B}[\delta_{b_1} \updownarrow^{\delta_{b_3}}_{\delta_{b_2}}]; \Delta \vdash P :: (a_{\mathsf{L}} : A_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])$*, and, for any fresh* $y'_{\mathsf{s}} : \hat{C}[\mathsf{w}_{\mathsf{b}_1} \updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}]$ *and a* Ψ' *and* γ *such that* $\hat{C} \leq \hat{B}$ *and* $\mathsf{w}_{\mathsf{b}_1}, \mathsf{w}_{\mathsf{b}_2}, \mathsf{w}_{\mathsf{b}_3} \in |\Psi'|$ *and* $\Psi'^{+} \vdash \hat{\gamma}(\Psi)$ *and* $\hat{C}[\mathsf{w}_{\mathsf{b}_1} \updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}] = \hat{\gamma}(\hat{B}[\delta_{b_1} \updownarrow^{\delta_{b_3}}_{\delta_{b_2}}])$ *and* $\Psi \vdash \hat{B}[\delta_{b_1} \updownarrow^{\delta_{b_3}}_{\delta_{b_2}}]\ \mathsf{type}$*, then* $\hat{\gamma}(\Gamma), y'_{\mathsf{s}} : \hat{C}[\mathsf{w}_{\mathsf{b}_1} \updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}]; \hat{\gamma}(\Delta) \vdash [y'_{\mathsf{s}}/y_{\mathsf{s}}]\, \hat{\gamma}(P) :: (a_{\mathsf{L}} : A_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])$*.*
9. *If* $\Gamma, c_{\mathsf{s}} : \hat{C}[\mathsf{w}_{\mathsf{c}_1} \updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}], z'_{\mathsf{s}} : \hat{C}[\mathsf{w}_{\mathsf{c}_1} \updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}]; \Delta \vdash P :: (a_{\mathsf{L}} : A_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])$*, then* $\Gamma, c_{\mathsf{s}} : \hat{C}[\mathsf{w}_{\mathsf{c}_1} \updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}]; \Delta \vdash [c_{\mathsf{s}}/z'_{\mathsf{s}}]\, P :: (a_{\mathsf{L}} : A_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])$*.*
10. *If* $\Gamma \vdash P :: (x_{\mathsf{s}} : A_{\mathsf{s}}[\mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])$*, then, for any fresh* $x'_{\mathsf{s}} : A_{\mathsf{s}}[\mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}]$*,* $\Gamma, x'_{\mathsf{s}} : A_{\mathsf{s}}[\mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}] \vdash [x'_{\mathsf{s}}/x_{\mathsf{s}}]\, P :: (x'_{\mathsf{s}} : A_{\mathsf{s}}[\mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])$*.*
11. *If* $\Gamma \vdash P :: (x_{\mathsf{s}} : A_{\mathsf{s}}[\delta_{a_1} \updownarrow^{\delta_{a_3}}_{\delta_{a_2}}])$*, and, for any fresh* $x'_{\mathsf{s}} : A'_{\mathsf{s}}[\mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}]$ *and a* Ψ' *and* γ *such that* $\mathsf{w}_{\mathsf{a}_1}, \mathsf{w}_{\mathsf{a}_2}, \mathsf{w}_{\mathsf{a}_3} \in |\Psi'|$ *and* $\Psi'^{+} \vdash \hat{\gamma}(\Psi)$ *and* $A'_{\mathsf{s}}[\mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}] = \hat{\gamma}(A_{\mathsf{s}}[\delta_{a_1} \updownarrow^{\delta_{a_3}}_{\delta_{a_2}}])$ *and and* $\Psi \vdash A_{\mathsf{s}}[\delta_{a_1} \updownarrow^{\delta_{a_3}}_{\delta_{a_2}}]\ \mathsf{type}$*, then* $\hat{\gamma}(\Gamma), x'_{\mathsf{s}} : A'_{\mathsf{s}}[\mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}] \vdash [x'_{\mathsf{s}}/x_{\mathsf{s}}]\, \hat{\gamma}(P) :: (x'_{\mathsf{s}} : A'_{\mathsf{s}}[\mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])$*.*
12. *If* $\Gamma, a_{\mathsf{s}} : A_{\mathsf{s}}[\mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}], x_{\mathsf{s}} : A_{\mathsf{s}}[\mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}] \vdash P :: (x_{\mathsf{s}} : A_{\mathsf{s}}[\mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])$*, then* $\Gamma, a_{\mathsf{s}} : A_{\mathsf{s}}[\mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}] \vdash [a_{\mathsf{s}}/x_{\mathsf{s}}]\, P :: (a_{\mathsf{s}} : A_{\mathsf{s}}[\mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])$*.*
13. *If* $\Gamma, y_{\mathsf{s}} : \hat{B}[\mathsf{w}_{\mathsf{b}_1} \updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}] \vdash P :: (a_{\mathsf{s}} : A_{\mathsf{s}}[\mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])$*, then, for any fresh* $y'_{\mathsf{s}} : \hat{C}[\mathsf{w}_{\mathsf{b}_1} \updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}]$ *such that* $\hat{C} \leq \hat{B}$*,* $\Gamma, y'_{\mathsf{s}} : \hat{C}[\mathsf{w}_{\mathsf{b}_1} \updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}] \vdash [y'_{\mathsf{s}}/y_{\mathsf{s}}]\, P :: (a_{\mathsf{s}} : A_{\mathsf{s}}[\mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])$*.*
14. *If* $\Gamma, y_{\mathsf{s}} : \hat{B}[\delta_{b_1} \updownarrow^{\delta_{b_3}}_{\delta_{b_2}}] \vdash P :: (a_{\mathsf{s}} : A_{\mathsf{s}}[\mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])$*, and, for any fresh* $y'_{\mathsf{s}} : \hat{C}[\mathsf{w}_{\mathsf{b}_1} \updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}]$ *and a* Ψ' *and* γ *such that* $\hat{C} \leq \hat{B}$ *and* $\mathsf{w}_{\mathsf{b}_1}, \mathsf{w}_{\mathsf{b}_2}, \mathsf{w}_{\mathsf{b}_3} \in |\Psi'|$ *and* $\Psi'^{+} \vdash \hat{\gamma}(\Psi)$ *and* $\hat{C}[\mathsf{w}_{\mathsf{b}_1} \updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}] = \hat{\gamma}(\hat{B}[\delta_{b_1} \updownarrow^{\delta_{b_3}}_{\delta_{b_2}}])$ *and* $\Psi \vdash \hat{B}[\delta_{b_1} \updownarrow^{\delta_{b_3}}_{\delta_{b_2}}]\ \mathsf{type}$*, then* $\hat{\gamma}(\Gamma), y'_{\mathsf{s}} : \hat{C}[\mathsf{w}_{\mathsf{b}_1} \updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}] \vdash [y'_{\mathsf{s}}/y_{\mathsf{s}}]\, \hat{\gamma}(P) :: (a_{\mathsf{s}} : A_{\mathsf{s}}[\mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])$*.*
15. *If* $\Gamma, c_{\mathsf{s}} : \hat{C}[\mathsf{w}_{\mathsf{c}_1} \updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}], z_{\mathsf{s}} : \hat{C}[\mathsf{w}_{\mathsf{c}_1} \updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}] \vdash P :: (a_{\mathsf{s}} : A_{\mathsf{s}}[\mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])$*, then* $\Gamma, c_{\mathsf{s}} : \hat{C}[\mathsf{w}_{\mathsf{c}_1} \updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}] \vdash [c_{\mathsf{s}}/z_{\mathsf{s}}]\, P :: (a_{\mathsf{s}} : A_{\mathsf{s}}[\mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])$*.*

Corollary E.3 (Balance between Λ and Θ). *If the configuration* $\Lambda; \Theta$ *is well-formed, then, for any* $\mathsf{proc}(a_{\mathsf{s}}, \_, \_)$ *in* Λ*, there does not exist a* $\mathsf{proc}(a_{\mathsf{L}}, \_, \_)$ *in* Θ*.*

Lemma E.4 (Permutation of Θ). *Writing* $Q\langle a_{\mathsf{L}}\rangle$ *for a process term* Q *with an occurrence of a linear channel* a_{L}*, it holds for a well-formed and well-typed configuration*
$\Psi;\ \Gamma \vDash \Lambda;\ \Theta_1, \mathsf{proc}(a_{\mathsf{L}},\ \mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}},\ P_{a_{\mathsf{L}}}), \Theta_2, \mathsf{proc}(c_{\mathsf{L}},\ \mathsf{w}_{\mathsf{c}_1} \updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}},\ Q\langle a_{\mathsf{L}}\rangle), \Theta_3 :: \Gamma;\ \Phi, \Delta$
that the permutation
$\Psi;\ \Gamma \vDash \Lambda;\ \Theta_1, \Theta_2, \mathsf{proc}(a_{\mathsf{L}},\ \mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}},\ P_{a_{\mathsf{L}}}), \mathsf{proc}(c_{\mathsf{L}},\ \mathsf{w}_{\mathsf{c}_1} \updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}},\ Q\langle a_{\mathsf{L}}\rangle), \Theta_3 :: \Gamma;\ \Phi, \Delta$
remains well-formed and well-typed.

Proof. In the given typing derivation, $\mathsf{proc}(a_{\mathsf{L}},\ \mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}},\ P_{a_{\mathsf{L}}})$ provides $a_{\mathsf{L}} : A_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}]$ for some $A_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1} \updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}]$ to Θ_2, $\mathsf{proc}(c_{\mathsf{L}},\ \mathsf{w}_{\mathsf{c}_1} \updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}},\ Q\langle a_{\mathsf{L}}\rangle)$, Θ_3. Well-formedness of a

configuration (see Section B.3) guarantees that a_{L} is unique within a configuration, and hence there can be no other process in Θ_2 that provides a service along a_{L}. Moreover, channel a_{L} cannot be consumed by Θ_2 because channels are linear and a_{L} occurs in $\mathsf{proc}(c_{\mathsf{L}}, \mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}, Q\langle a_{\mathsf{L}}\rangle)$, leaving the set of descendants of any process in Θ_2 unaffected by the move, as witnessed by rule (T-DESC$_3$). Lastly, any channels used by $\mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P_{a_{\mathsf{L}}})$ will continue to be available if we move the process to the right, guaranteeing that the set of descendants of a_{L} remains unaffected by the move. Since the set of descendants of the involved processes remain invariant throughout the permutation, the Invariant 3 and Invariant 4 will continue to hold for the permuted configuration. □

Lemma E.5 (Truncate Θ). *For a well-formed and well-typed linear configuration*
$\Psi; \Gamma \vDash \Theta_1, \mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P_{a_{\mathsf{L}}}), \Theta_2 :: \Phi, \Delta$
there exist A_{L} *and* Φ'*, and* Δ' *such that*
$\Psi; \Gamma \vDash \Theta_1, \mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P_{a_{\mathsf{L}}}) :: (\Phi', \Delta', a_{\mathsf{L}} : A_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])$
is well-formed and well-typed.

Lemma E.6 (Invariant Sub-Configuration in Θ). *For well-formed and well-typed linear configurations*
$\Psi; \Gamma \vDash \Theta_1, \Theta_2 :: \Phi, \Delta$ *and*
$\Psi; \Gamma \vDash \Theta_1 :: \Phi', \Delta'$
it holds that
$\Psi'; \Gamma' \vDash \Theta'_1, \Theta_2 :: \Phi, \Delta$,
if there exist Ψ'*,* Γ' *and* Θ'_1 *such that* $\Psi'; \Gamma' \vDash \Theta'_1 :: \Phi', \Delta'$ *and* $\Gamma' \lhd \Gamma$ *and* $\Psi' \lhd \Psi$*.*

E.3 Theorem

The preservation theorem expresses that the types of the providing linear channels are maintained along transitions and that new shared channels may be allocated and the types of existing shared channels concretized.

Theorem E.7 (Preservation). *If* $\Psi; \Gamma \vDash \Lambda; \Theta :: \Gamma; \Phi, \Delta$ *and* $\Lambda; \Theta \longrightarrow \Lambda'; \Theta'$*, then* $\Psi'; \Gamma' \vDash \Lambda'; \Theta' :: \Gamma'; \Phi, \Delta$*, for some* Λ'*,* Θ'*,* Ψ'*, and* Γ' *such that* $\Psi' \lhd \Psi$ *and* $\Gamma' \lhd \Gamma$*.*

Proof. We prove preservation by induction on the dynamics, constructing a derivation of a well-formed and well-typed configuration $\Psi; \Gamma' \vDash \Lambda''; \Theta'' :: \Gamma'; \Phi, \Delta$, where Λ'' and Θ'' are permutations of Λ' and Θ', respectively:

1. (D-SPAWN$_{\mathsf{LL}}$) $\mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, x_{\mathsf{L}} : A_{\mathsf{L}}[\mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}] \leftarrow X_{\mathsf{L}} \leftarrow \overline{c_{\mathsf{L}}}, \overline{\tilde{c}_{\mathsf{L}}}, \overline{d_{\mathsf{S}}};\ Q_{x_{\mathsf{L}}}),$
$\mathsf{!def}(\Psi' \vdash x'_{\mathsf{L}} : A'_{\mathsf{L}}[\delta_j\updownarrow^{\delta_n}_{\delta_k}] \leftarrow X_{\mathsf{L}} \leftarrow \Delta', \Phi', \Gamma' = P_{x'_{\mathsf{L}},\mathsf{dom}(\Delta'),\mathsf{dom}(\Phi'),\mathsf{dom}(\Gamma'),\Psi''})$
$\longrightarrow \mathsf{proc}(b_{\mathsf{L}}, \mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}, [b_{\mathsf{L}}/x'_{\mathsf{L}}, \overline{c_{\mathsf{L}}}/\mathsf{dom}(\Delta'), \overline{\tilde{c}_{\mathsf{L}}}/\mathsf{dom}(\Phi'), \overline{d_{\mathsf{S}}}/\mathsf{dom}(\Gamma')]\hat{\gamma}(P_{x'_{\mathsf{L}},\mathsf{dom}(\Delta'),\mathsf{dom}(\Phi'),\mathsf{dom}(\Gamma}$
$\mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, [b_{\mathsf{L}}/x_{\mathsf{L}}]Q_{x_{\mathsf{L}}}),$
$\mathsf{unavail}(b_{\mathsf{S}}, \mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}})$ (b *fresh*)

$\Psi;\, \Gamma \vDash \Lambda;\, \Theta_1, \mathsf{proc}(a_{\mathsf{L}},\, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}},\, x_{\mathsf{L}} : A_{\mathsf{L}}[\mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}] \leftarrow X_{\mathsf{L}} \leftarrow \overline{c_{\mathsf{L}}}, \overline{\tilde{c}_{\mathsf{L}}}, \overline{d_{\mathsf{S}}};\; Q_{x_{\mathsf{L}}}), \Theta_2 :: \Gamma;\, \Phi, \Delta$ (by assumption)
for some Θ_1, *and* Θ_2 *and where* $\mathsf{!def}(\, \Psi' \vdash x'_{\mathsf{L}} : A'_{\mathsf{L}}[\delta_j\updownarrow^{\delta_n}_{\delta_k}] \leftarrow X_{\mathsf{L}} \leftarrow \Delta', \Phi', \Gamma' = P_{x'_{\mathsf{L}},\mathsf{dom}(\Delta'),\mathsf{dom}(\Phi'),\mathsf{dom}(\Gamma')}$

$\Lambda;\, \Theta_1, \mathsf{proc}(a_{\mathsf{L}},\, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}},\, x_{\mathsf{L}} : A_{\mathsf{L}}[\mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}] \leftarrow X_{\mathsf{L}} \leftarrow \overline{c_{\mathsf{L}}}, \overline{\tilde{c}_{\mathsf{L}}}, \overline{d_{\mathsf{S}}};\; Q_{x_{\mathsf{L}}}), \Theta_2$ (this case)
$\longrightarrow \Lambda, \mathsf{unavail}(b_{\mathsf{S}},\, \mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}});$
$\Theta_1, \mathsf{proc}(b_{\mathsf{L}},\, \mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}},\, [b_{\mathsf{L}}/x'_{\mathsf{L}}, \overline{c_{\mathsf{L}}}/\mathsf{dom}(\Delta'), \overline{\tilde{c}_{\mathsf{L}}}/\mathsf{dom}(\Phi'), \overline{d_{\mathsf{S}}}/\mathsf{dom}(\Gamma')]\hat{\gamma}(P_{x'_{\mathsf{L}},\mathsf{dom}(\Delta'),\mathsf{dom}(\Phi'),\mathsf{dom}(\Gamma'),\Psi''})),$
$\mathsf{proc}(a_{\mathsf{L}},\, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}},\, [b_{\mathsf{L}}/x_{\mathsf{L}}]Q_{x_{\mathsf{L}}}),\, \Theta_2 \quad (b\ \textit{fresh})$

$\Psi;\, \Gamma \vDash \Lambda :: \Gamma$ (by inversion on (T-Ω))

$\Psi;\, \Gamma \vDash \Theta_1, \mathsf{proc}(a_{\mathsf{L}},\, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}},\, x_{\mathsf{L}} : A_{\mathsf{L}}[\mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}] \leftarrow X_{\mathsf{L}} \leftarrow \overline{c_{\mathsf{L}}}, \overline{\tilde{c}_{\mathsf{L}}}, \overline{d_{\mathsf{S}}};\; Q_{x_{\mathsf{L}}}), \Theta_2 :: \Phi, \Delta$ (by inversion on (T-Ω))

$\Psi;\, \Gamma \vDash \Theta_1, \mathsf{proc}(a_{\mathsf{L}},\, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}},\, x_{\mathsf{L}} : A_{\mathsf{L}}[\mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}] \leftarrow X_{\mathsf{L}} \leftarrow \overline{c_{\mathsf{L}}}, \overline{\tilde{c}_{\mathsf{L}}}, \overline{d_{\mathsf{S}}};\; Q_{x_{\mathsf{L}}}) :: (\Phi_1, \Delta_1, a_{\mathsf{L}} : D_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])$
for some D_{L}, Φ_1, *and* Δ_1 (by Lemma E.5)

$\Psi;\, \Gamma \vDash \Theta_1 :: \Phi_1, \Phi_2, \Delta_1, \Delta_2$ (by inversion on (T-Θ_2))
for some Φ_2 *and* Δ_2

$\Psi;\, \Gamma;\, \Phi_2;\, \Delta_2 \vdash x_{\mathsf{L}} : A_{\mathsf{L}}[\mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}] \leftarrow X_{\mathsf{L}} \leftarrow \overline{c_{\mathsf{L}}}, \overline{\tilde{c}_{\mathsf{L}}}, \overline{d_{\mathsf{S}}};\; Q_{x_{\mathsf{L}}} :: (a_{\mathsf{L}} : D_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])$(by inversion on (T-$\Theta_2$))

$\vdash (D_{\mathsf{L}}, \hat{D})\ \mathsf{sesync}$ *and* $\Psi \vdash D_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}]\ \mathsf{type}$ *and* $\Psi^* \vdash \mathsf{w}_{\mathsf{a}_2} \leq \mathsf{w}_{\mathsf{a}_3}$ (by inversion on (T-Θ_2))
for some $(a_{\mathsf{S}} : \hat{D}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}]) \in \Gamma$

$\mathsf{Inv}_3(\mathsf{proc}(a_{\mathsf{L}},\, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}},\, x_{\mathsf{L}} : A_{\mathsf{L}}[\mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}] \leftarrow X_{\mathsf{L}} \leftarrow \overline{c_{\mathsf{L}}}, \overline{\tilde{c}_{\mathsf{L}}}, \overline{d_{\mathsf{S}}};\; Q_{x_{\mathsf{L}}}))$ (by inversion on (T-Θ_2))

$\mathsf{Inv}_4(\mathsf{proc}(a_{\mathsf{L}},\, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}},\, x_{\mathsf{L}} : A_{\mathsf{L}}[\mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}] \leftarrow X_{\mathsf{L}} \leftarrow \overline{c_{\mathsf{L}}}, \overline{\tilde{c}_{\mathsf{L}}}, \overline{d_{\mathsf{S}}};\; Q_{x_{\mathsf{L}}}))$ (by inversion on (T-Θ_2))

$\Psi;\, \Gamma;\, \Phi_2;\, \Delta_2 \vdash x_{\mathsf{L}} : A_{\mathsf{L}}[\mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}] \leftarrow X_{\mathsf{L}} \leftarrow \overline{c_{\mathsf{L}}}, \overline{\tilde{c}_{\mathsf{L}}}, \overline{d_{\mathsf{S}}};\; Q_{x_{\mathsf{L}}} :: (a_{\mathsf{L}} : D_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])$(by inversion on (T-SPAWN$_{\mathsf{LL}}$))
for some Γ_3, Φ_3, Δ_3, $\overline{C_{\mathsf{S}}[\mathsf{w}_{\mathsf{d}_1}\updownarrow^{\mathsf{w}_{\mathsf{d}_3}}_{\mathsf{w}_{\mathsf{d}_2}}]}$, $\overline{\tilde{B}_{\mathsf{L}}[\tilde{\mathsf{w}}_{\mathsf{c}_1}\updownarrow^{\tilde{\mathsf{w}}_{\mathsf{c}_3}}_{\tilde{\mathsf{w}}_{\mathsf{c}_2}}]}$, and $\overline{B_{\mathsf{L}}[\mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}]}$
such that $\Gamma = \Gamma_3, \overline{d_{\mathsf{S}} : C_{\mathsf{S}}[\mathsf{w}_{\mathsf{d}_1}\updownarrow^{\mathsf{w}_{\mathsf{d}_3}}_{\mathsf{w}_{\mathsf{d}_2}}]}$, $\Phi_2 = \Phi_3, \overline{\tilde{c}_{\mathsf{L}} : \tilde{B}_{\mathsf{L}}[\tilde{\mathsf{w}}_{\mathsf{c}_1}\updownarrow^{\tilde{\mathsf{w}}_{\mathsf{c}_3}}_{\tilde{\mathsf{w}}_{\mathsf{c}_2}}]}$, *and* $\Delta_2 = \Delta_3, \overline{c_{\mathsf{L}} : B_{\mathsf{L}}[\mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}]}$

$\Psi;\, \Gamma;\, \Phi_3;\, \Delta_3, x_{\mathsf{L}} : A_{\mathsf{L}}[\mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}] \vdash Q_{x_{\mathsf{L}}} :: (a_{\mathsf{L}} : D_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])$ (by inversion on (T-SPAWN$_{\mathsf{LL}}$))

$\Psi^+ \vdash \mathsf{w}_{\mathsf{a}_3} < \mathsf{w}_{\mathsf{b}_2}$ (by inversion on (T-SPAWN$_{\mathsf{LL}}$))

$(\Psi' \vdash x'_{\mathsf{L}} : A'_{\mathsf{L}}[\delta_j\updownarrow^{\delta_n}_{\delta_k}] \leftarrow X_{\mathsf{L}} \leftarrow \Delta', \Phi', \Gamma' = P_{x'_{\mathsf{L}},\mathsf{dom}(\Delta'),\mathsf{dom}(\Phi'),\mathsf{dom}(\Gamma'),\Psi''}) \in \Sigma$ (by inversion on (T-SPAWN$_{\mathsf{LL}}$)
(and since $\mathsf{!def}(\, \Psi' \vdash x'_{\mathsf{L}} : A'_{\mathsf{L}}[\delta_j\updownarrow^{\delta_n}_{\delta_k}] \leftarrow X_{\mathsf{L}} \leftarrow \Delta', \Phi', \Gamma' = P_{x'_{\mathsf{L}},\mathsf{dom}(\Delta'),\mathsf{dom}(\Phi'),\mathsf{dom}(\Gamma'),\Psi''})$)

$\hat{\gamma}(A'_{\mathsf{L}}[\delta_j\updownarrow^{\delta_n}_{\delta_k}]) = A_{\mathsf{L}}[\mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}]$ (by inversion on (T-SPAWN$_{\mathsf{LL}}$))

$\hat{\gamma}(\Delta') = \overline{c_{\mathsf{L}} : B_{\mathsf{L}}[\mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}]}$ (by inversion on (T-SPAWN$_{\mathsf{LL}}$))

$\hat{\gamma}(\Phi') = \overline{\tilde{c}_{\mathsf{L}} : \tilde{B}_{\mathsf{L}}[\tilde{\mathsf{w}}_{\mathsf{c}_1}\updownarrow^{\tilde{\mathsf{w}}_{\mathsf{c}_3}}_{\tilde{\mathsf{w}}_{\mathsf{c}_2}}]}$ (by inversion on (T-SPAWN$_{\mathsf{LL}}$))

$\hat{\gamma}(\Gamma') = \overline{d_{\mathsf{S}} : C_{\mathsf{S}}[\mathsf{w}_{\mathsf{d}_1}\updownarrow^{\mathsf{w}_{\mathsf{d}_3}}_{\mathsf{w}_{\mathsf{d}_2}}]}$ (by inversion on (T-SPAWN$_{\mathsf{LL}}$))

$\Psi \vdash \hat{\gamma}(\Psi')$ (by inversion on (T-SPAWN$_{\mathsf{LL}}$))

$\Psi';\, \Gamma';\, \Phi';\, \Delta' \vdash P_{x'_{\mathsf{L}},\mathsf{dom}(\Delta'),\mathsf{dom}(\Phi'),\mathsf{dom}(\Gamma'),\Psi''} :: (x'_{\mathsf{L}} : A'_{\mathsf{L}}[\delta_j\updownarrow^{\delta_n}_{\delta_k}])$ (by inversion on (T-Σ_2))
where $\Psi'' = \mathsf{wrl}(A'_{\mathsf{L}}[\delta_j\updownarrow^{\delta_n}_{\delta_k}]), \mathsf{wrl}(\Delta'), \mathsf{wrl}(\Phi'), \mathsf{wrl}(\Gamma')$ *and* $\Psi' =< (\Psi'')$ *and* Ψ'^+ *irreflexive*

$\vdash (A'_{\mathsf{L}}, \top)\ \mathsf{sesync}$ (by inversion on (T-Σ_2))

$\Psi' \vdash A'_{\mathsf{L}}[\delta_j \updownarrow_{\delta_k}^{\delta_n}]\ \mathsf{type}$ (by inversion on (T-Σ_2))

$\forall y_{\mathsf{L}_i} : B_{\mathsf{L}_i}[\delta_{m_i} \updownarrow_{\delta_{u_i}}^{\delta_{v_i}}] \in \Delta' \cup \Phi' : \Psi' \vdash B_{\mathsf{L}_i}[\delta_{m_i} \updownarrow_{\delta_{u_i}}^{\delta_{v_i}}]\ \mathsf{type}$ (by inversion on (T-Σ_2))

$\forall z_{\mathsf{S}_i} : C_{\mathsf{S}_i}[\delta_{l_i} \updownarrow_{\delta_{p_i}}^{\delta_{r_i}}] \in \Gamma' : \Psi' \vdash C_{\mathsf{S}_i}[\delta_{l_i} \updownarrow_{\delta_{p_i}}^{\delta_{r_i}}]\ \mathsf{type}$ (by inversion on (T-Σ_2))

$\Psi'^{*} \vdash \delta_k \leq \delta_n$ (by inversion on (T-Σ_2))

$\forall y_{\mathsf{L}_i} : B_{\mathsf{L}_i}[\delta_{m_i} \updownarrow_{\delta_{u_i}}^{\delta_{v_i}}] \in \Phi' : \Psi'^{*} \vdash \delta_k \leq \delta_{m_i} \leq \delta_n$ (by inversion on (T-Σ_2))

$\forall y_{\mathsf{L}_i} : B_{\mathsf{L}_i}[\delta_{m_i} \updownarrow_{\delta_{u_i}}^{\delta_{v_i}}] \in \Delta' \cup \Phi : \Psi'^{+} \vdash \delta_n < \delta_{u_i}$ (by inversion on (T-Σ_2))

$\hat{\gamma}(\Psi'); \Gamma''; \hat{\gamma}(\Phi'); \hat{\gamma}(\Delta') \vdash [b_{\mathsf{L}}/x'_{\mathsf{L}}, \overline{c_{\mathsf{L}}}/\mathsf{dom}(\Delta'), \overline{\tilde{c}_{\mathsf{L}}}/\mathsf{dom}(\Phi'), \overline{z'_{\mathsf{S}}}/\mathsf{dom}(\Gamma')]\hat{\gamma}(P)_{x'_{\mathsf{L}},\mathsf{dom}(\Delta'),\mathsf{dom}(\Phi'),\mathsf{dom}(\Gamma'),\Psi''} :: (b_{\mathsf{L}}$
for $\Gamma'' = \overline{z'_{\mathsf{S}} : C_{\mathsf{S}}[\mathsf{w}_{\mathsf{d}_1} \updownarrow_{\mathsf{w}_{\mathsf{d}_2}}^{\mathsf{w}_{\mathsf{d}_3}}]}$ (by Lemma E.2–2, E.2–4, E.2–6, E.2–8)
and where $\Psi \vdash \hat{\gamma}(\Psi')$ *and* $\hat{\gamma}(\Gamma') = \overline{d_{\mathsf{S}} : C_{\mathsf{S}}[\mathsf{w}_{\mathsf{d}_1} \updownarrow_{\mathsf{w}_{\mathsf{d}_2}}^{\mathsf{w}_{\mathsf{d}_3}}]}$ (since b, $\overline{z'_{\mathsf{S}}}$, $\overline{\tilde{c}_{\mathsf{L}}}$, and $\overline{c_{\mathsf{L}}}$ fresh)
and $\hat{\gamma}(\Phi') = \overline{\tilde{c}_{\mathsf{L}} : \tilde{B}_{\mathsf{L}}[\tilde{\mathsf{w}}_{\mathsf{c}_1} \updownarrow_{\tilde{\mathsf{w}}_{\mathsf{c}_2}}^{\tilde{\mathsf{w}}_{\mathsf{c}_3}}]}$ *and* $\hat{\gamma}(\Delta') = \overline{c_{\mathsf{L}} : B_{\mathsf{L}}[\mathsf{w}_{\mathsf{c}_1} \updownarrow_{\mathsf{w}_{\mathsf{c}_2}}^{\mathsf{w}_{\mathsf{c}_3}}]}$

$\hat{\gamma}(\Psi'); \Gamma; \hat{\gamma}(\Phi'); \hat{\gamma}(\Delta') \vdash P' :: (b_{\mathsf{L}} : A_{\mathsf{L}}[\mathsf{w}_{\mathsf{b}_1} \updownarrow_{\mathsf{w}_{\mathsf{b}_2}}^{\mathsf{w}_{\mathsf{b}_3}}])$ (by Lemma E.2–9 and weakening and since $\overline{d_{\mathsf{S}} : C_{\mathsf{S}}[\mathsf{w}_{\mathsf{d}_1} \updownarrow_{\mathsf{w}_{\mathsf{d}_2}}^{\mathsf{w}_{\mathsf{d}_3}}]}$
where $P' = [b_{\mathsf{L}}/x'_{\mathsf{L}}, \overline{c_{\mathsf{L}}}/\mathsf{dom}(\Delta'), \overline{\tilde{c}_{\mathsf{L}}}/\mathsf{dom}(\Phi'), \overline{d_{\mathsf{S}}}/\overline{z'_{\mathsf{S}}}, \overline{z'_{\mathsf{S}}}/\mathsf{dom}(\Gamma')]\hat{\gamma}(P)_{x'_{\mathsf{L}},\mathsf{dom}(\Delta'),\mathsf{dom}(\Phi'),\mathsf{dom}(\Gamma'),\Psi''}$

$\Psi \vdash A_{\mathsf{L}}[\mathsf{w}_{\mathsf{b}_1} \updownarrow_{\mathsf{w}_{\mathsf{b}_2}}^{\mathsf{w}_{\mathsf{b}_3}}]\ \mathsf{type}$ (since $\hat{\gamma}(A'_{\mathsf{L}}[\delta_j \updownarrow_{\delta_k}^{\delta_n}]) = A_{\mathsf{L}}[\mathsf{w}_{\mathsf{b}_1} \updownarrow_{\mathsf{w}_{\mathsf{b}_2}}^{\mathsf{w}_{\mathsf{b}_3}}]$ and $\Psi \vdash \hat{\gamma}(\Psi')$)

$\Psi^{*} \vdash \mathsf{w}_{\mathsf{b}_2} \leq \mathsf{w}_{\mathsf{b}_3}$ (since $\hat{\gamma}(A'_{\mathsf{L}}[\delta_j \updownarrow_{\delta_k}^{\delta_n}]) = A_{\mathsf{L}}[\mathsf{w}_{\mathsf{b}_1} \updownarrow_{\mathsf{w}_{\mathsf{b}_2}}^{\mathsf{w}_{\mathsf{b}_3}}]$ and $\Psi \vdash \hat{\gamma}(\Psi')$)

$\mathsf{Inv}_3(\mathsf{proc}(b_{\mathsf{L}}, \mathsf{w}_{\mathsf{b}_1} \updownarrow_{\mathsf{w}_{\mathsf{b}_2}}^{\mathsf{w}_{\mathsf{b}_3}}, P'))$ (since $\forall y_{\mathsf{L}_i} : B_{\mathsf{L}_i}[\delta_{m_i} \updownarrow_{\delta_{u_i}}^{\delta_{v_i}}] \in \Phi' : \Psi'^{*} \vdash \delta_k \leq \delta_{m_i} \leq \delta_n$ and $\hat{\gamma}(\Phi') = \overline{\tilde{c}_{\mathsf{L}} : \tilde{B}_{\mathsf{L}}[\tilde{\mathsf{w}}_{\mathsf{c}_1} \updownarrow_{\tilde{\mathsf{w}}_{\mathsf{c}_2}}^{\tilde{\mathsf{w}}_{\mathsf{c}_3}}]}$

$\mathsf{Inv}_4(\mathsf{proc}(b_{\mathsf{L}}, \mathsf{w}_{\mathsf{b}_1} \updownarrow_{\mathsf{w}_{\mathsf{b}_2}}^{\mathsf{w}_{\mathsf{b}_3}}, P'))$ (since $\forall y_{\mathsf{L}_i} : B_{\mathsf{L}_i}[\delta_{m_i} \updownarrow_{\delta_{u_i}}^{\delta_{v_i}}] \in \Delta' \cup \Phi : \Psi'^{+} \vdash \delta_n < \delta_{u_i}$ and $\hat{\gamma}(\Delta') = \overline{c_{\mathsf{L}} : B_{\mathsf{L}}[\mathsf{w}_{\mathsf{c}_1} \updownarrow_{\mathsf{w}_{\mathsf{c}_2}}^{\mathsf{w}_{\mathsf{c}_3}}]}$
(and $\hat{\gamma}(\Phi') = \overline{\tilde{c}_{\mathsf{L}} : \tilde{B}_{\mathsf{L}}[\tilde{\mathsf{w}}_{\mathsf{c}_1} \updownarrow_{\tilde{\mathsf{w}}_{\mathsf{c}_2}}^{\tilde{\mathsf{w}}_{\mathsf{c}_3}}]}$ and $\Delta' \cup \Phi$ are only descendants)

$\Psi; \Gamma \vDash \Theta_1, \mathsf{proc}(b_{\mathsf{L}}, \mathsf{w}_{\mathsf{b}_1} \updownarrow_{\mathsf{w}_{\mathsf{b}_2}}^{\mathsf{w}_{\mathsf{b}_3}}, P') :: (\Phi_1, \Phi_3, \Delta_1, \Delta_3, b_{\mathsf{L}} : A_{\mathsf{L}}[\mathsf{w}_{\mathsf{b}_1} \updownarrow_{\mathsf{w}_{\mathsf{b}_2}}^{\mathsf{w}_{\mathsf{b}_3}}])$ (by (T-Θ_2))

$\Psi; \Gamma; \Phi_3; \Delta_3, b_{\mathsf{L}} : A_{\mathsf{L}}[\mathsf{w}_{\mathsf{b}_1} \updownarrow_{\mathsf{w}_{\mathsf{b}_2}}^{\mathsf{w}_{\mathsf{b}_3}}] \vdash [b_{\mathsf{L}}/x_{\mathsf{L}}]\, Q_{x_{\mathsf{L}}} :: (a_{\mathsf{L}} : D_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1} \updownarrow_{\mathsf{w}_{\mathsf{a}_2}}^{\mathsf{w}_{\mathsf{a}_3}}])$ (by Lemma E.2–3)

$\mathsf{Inv}_3(\mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1} \updownarrow_{\mathsf{w}_{\mathsf{a}_2}}^{\mathsf{w}_{\mathsf{a}_3}}, [b_{\mathsf{L}}/x_{\mathsf{L}}]\, Q_{x_{\mathsf{L}}}))$ (since $\mathsf{dom}(\Phi_3) \subseteq \mathsf{dom}(\Phi_2)$)

$\mathsf{Inv}_4(\mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1} \updownarrow_{\mathsf{w}_{\mathsf{a}_2}}^{\mathsf{w}_{\mathsf{a}_3}}, [b_{\mathsf{L}}/x_{\mathsf{L}}]\, Q_{x_{\mathsf{L}}}))$ (since $\mathsf{Inv}_4(\mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1} \updownarrow_{\mathsf{w}_{\mathsf{a}_2}}^{\mathsf{w}_{\mathsf{a}_3}}, x_{\mathsf{L}} : A_{\mathsf{L}}[\mathsf{w}_{\mathsf{b}_1} \updownarrow_{\mathsf{w}_{\mathsf{b}_2}}^{\mathsf{w}_{\mathsf{b}_3}}] \leftarrow X_{\mathsf{L}} \leftarrow \overline{c_{\mathsf{L}}}, \overline{\tilde{c}_{\mathsf{L}}}, \overline{d_{\mathsf{S}}};\ Q_{x_{\mathsf{L}}})))$
(and $\Psi^{+} \vdash \mathsf{w}_{\mathsf{a}_3} < \mathsf{w}_{\mathsf{b}_2}$)

$\Psi; \Gamma \vDash \Theta_1, \mathsf{proc}(b_{\mathsf{L}}, \mathsf{w}_{\mathsf{b}_1} \updownarrow_{\mathsf{w}_{\mathsf{b}_2}}^{\mathsf{w}_{\mathsf{b}_3}}, P'), \mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1} \updownarrow_{\mathsf{w}_{\mathsf{a}_2}}^{\mathsf{w}_{\mathsf{a}_3}}, [b_{\mathsf{L}}/x_{\mathsf{L}}]\, Q_{x_{\mathsf{L}}}) :: (\Phi_1, \Delta_1, a_{\mathsf{L}} : D_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1} \updownarrow_{\mathsf{w}_{\mathsf{a}_2}}^{\mathsf{w}_{\mathsf{a}_3}}])$ (by (T-Θ_2))

$\Psi'; \Gamma' \vDash \Theta_1, \mathsf{proc}(b_{\mathsf{L}}, \mathsf{w}_{\mathsf{b}_1} \updownarrow_{\mathsf{w}_{\mathsf{b}_2}}^{\mathsf{w}_{\mathsf{b}_3}}, P'), \mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1} \updownarrow_{\mathsf{w}_{\mathsf{a}_2}}^{\mathsf{w}_{\mathsf{a}_3}}, [b_{\mathsf{L}}/x_{\mathsf{L}}]\, Q_{x_{\mathsf{L}}}) :: (\Phi_1, \Delta_1, a_{\mathsf{L}} : D_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1} \updownarrow_{\mathsf{w}_{\mathsf{a}_2}}^{\mathsf{w}_{\mathsf{a}_3}}])$ (by weakening)
where $\Gamma' = \Gamma, b_{\mathsf{S}} : \top$

$\Gamma' \lhd \Gamma$ (by Definition E.1)

$\Psi'; \Gamma' \vDash \Theta_1, \mathsf{proc}(b_{\mathsf{L}}, \mathsf{w}_{\mathsf{b}_1} \updownarrow_{\mathsf{w}_{\mathsf{b}_2}}^{\mathsf{w}_{\mathsf{b}_3}}, P'), \mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1} \updownarrow_{\mathsf{w}_{\mathsf{a}_2}}^{\mathsf{w}_{\mathsf{a}_3}}, [b_{\mathsf{L}}/x_{\mathsf{L}}]\, Q_{x_{\mathsf{L}}}), \Theta_2 :: \Phi, \Delta$ (by Lemma E.6 and since $\Gamma' \lhd$

$\Psi; \Gamma' \vDash \mathsf{unavail}(b_{\mathsf{S}}, \mathsf{w}_{\mathsf{b}_1} \updownarrow_{\mathsf{w}_{\mathsf{b}_2}}^{\mathsf{w}_{\mathsf{b}_3}}) :: (b_{\mathsf{S}} : \top[\mathsf{w}_{\mathsf{b}_1} \updownarrow_{\mathsf{w}_{\mathsf{b}_2}}^{\mathsf{w}_{\mathsf{b}_3}}])$ (by (T-Λ_3))

$\Psi; \Gamma' \vDash \Lambda, \mathsf{unavail}(b_{\mathsf{S}}, \mathsf{w}_{\mathsf{b}_1} \updownarrow_{\mathsf{w}_{\mathsf{b}_2}}^{\mathsf{w}_{\mathsf{b}_3}}) :: \Gamma'$ (by (T-Λ_4) and weakening)

$\Psi; \Gamma' \vDash \mathsf{unavail}(b_{\mathsf{S}}, \mathsf{w}_{\mathsf{b}_1} \updownarrow_{\mathsf{w}_{\mathsf{b}_2}}^{\mathsf{w}_{\mathsf{b}_3}}); \Theta_1, \mathsf{proc}(b_{\mathsf{L}}, \mathsf{w}_{\mathsf{b}_1} \updownarrow_{\mathsf{w}_{\mathsf{b}_2}}^{\mathsf{w}_{\mathsf{b}_3}}, P'), \mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1} \updownarrow_{\mathsf{w}_{\mathsf{a}_2}}^{\mathsf{w}_{\mathsf{a}_3}}, [b_{\mathsf{L}}/x_{\mathsf{L}}]\, Q_{x_{\mathsf{L}}}), \Theta_2 :: \Gamma'; \Phi, \Delta$

(by (T-Ω) and well-formedness preserved)

2. (D-$\uparrow^{\mathsf{S}}_{\mathsf{L}}$) $\mathsf{proc}(a_{\mathsf{S}}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, x_{\mathsf{L}} \leftarrow \mathsf{accept}\ a_{\mathsf{S}}\,;P_{x_{\mathsf{L}}}), \mathsf{proc}(c_{\mathsf{L}}, \mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}, x_{\mathsf{L}} \leftarrow \mathsf{acquire}\ a_{\mathsf{S}}\,;Q_{x_{\mathsf{L}}})$
$\longrightarrow \mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, [a_{\mathsf{L}}/x_{\mathsf{L}}]\,P_{x_{\mathsf{L}}}), \mathsf{proc}(c_{\mathsf{L}}, \mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}, [a_{\mathsf{L}}/x_{\mathsf{L}}]\,Q_{x_{\mathsf{L}}}), \mathsf{unavail}(a_{\mathsf{S}}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}})$

$\Psi;\, \Gamma \vDash \Lambda_1, \mathsf{proc}(a_{\mathsf{S}}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, x_{\mathsf{L}} \leftarrow \mathsf{accept}\ a_{\mathsf{S}}\,;P_{x_{\mathsf{L}}}); \Theta_1, \mathsf{proc}(c_{\mathsf{L}}, \mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}, x_{\mathsf{L}} \leftarrow \mathsf{acquire}\ a_{\mathsf{S}}\,;Q_{x_{\mathsf{L}}}), \Theta_2 :: \Gamma;\, \Phi, \Delta$
for some Λ_1, Θ_1, *and* Θ_2 (by assumption)

$\Lambda_1, \mathsf{proc}(a_{\mathsf{S}}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, x_{\mathsf{L}} \leftarrow \mathsf{accept}\ a_{\mathsf{S}}\,;P_{x_{\mathsf{L}}}); \Theta_1, \mathsf{proc}(c_{\mathsf{L}}, \mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}, x_{\mathsf{L}} \leftarrow \mathsf{acquire}\ a_{\mathsf{S}}\,;Q_{x_{\mathsf{L}}}), \Theta_2$ (this case)
$\longrightarrow \Lambda_1, \mathsf{unavail}(a_{\mathsf{S}}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}); \Theta_1, \mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, [a_{\mathsf{L}}/x_{\mathsf{L}}]\,P_{x_{\mathsf{L}}}), \mathsf{proc}(c_{\mathsf{L}}, \mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}, [a_{\mathsf{L}}/x_{\mathsf{L}}]\,Q_{x_{\mathsf{L}}}), \Theta_2$

$\Psi;\, \Gamma \vDash \Lambda_1, \mathsf{proc}(a_{\mathsf{S}}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, x_{\mathsf{L}} \leftarrow \mathsf{accept}\ a_{\mathsf{S}}\,;P_{x_{\mathsf{L}}}) :: \Gamma$ (by inversion on (T-Ω))

$\Psi;\, \Gamma \vDash \Lambda_1 :: \Gamma_1$ *and* $\Psi;\, \Gamma \vDash \mathsf{proc}(a_{\mathsf{S}}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, x_{\mathsf{L}} \leftarrow \mathsf{accept}\ a_{\mathsf{S}}\,;P_{x_{\mathsf{L}}}) :: \Gamma_2$ (by inversion on (T-Λ_4))
for some Γ_1 *and* Γ_2 *such that* $\Gamma = \Gamma_1, \Gamma_2$

$\Gamma_2 = a_{\mathsf{S}} : \uparrow^{\mathsf{S}}_{\mathsf{L}} A_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}]$ *and* $\Psi;\, \Gamma \vdash x_{\mathsf{L}} \leftarrow \mathsf{accept}\, a_{\mathsf{S}}\,;P_{x_{\mathsf{L}}} :: (a_{\mathsf{S}} : \uparrow^{\mathsf{S}}_{\mathsf{L}} A_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])$ (by inversion on (T-Λ_2))
for some A_{L}

$\Psi^* \vdash \mathsf{w}_{\mathsf{a}_2} \leq \mathsf{w}_{\mathsf{a}_3}$ *and* $\vdash (\uparrow^{\mathsf{S}}_{\mathsf{L}} A_{\mathsf{L}}, \top)\ \mathsf{sesync}$ *and* $\Psi \vdash \uparrow^{\mathsf{S}}_{\mathsf{L}} A_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}]\ \mathsf{type}$ (by inversion on (T-Λ_2))

$\Psi;\, \Gamma;\, \cdot\,;\, \cdot \vdash P_{x_{\mathsf{L}}} :: (x_{\mathsf{L}} : A_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])$ (by inversion on (T-$\uparrow^{\mathsf{S}}_{\mathsf{L}}$R))

$\Psi;\, \Gamma \vDash \mathsf{unavail}(a_{\mathsf{S}}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}) :: (a_{\mathsf{S}} : \uparrow^{\mathsf{S}}_{\mathsf{L}} A_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])$ (by (T-Λ_3))

$\Psi;\, \Gamma \vDash \Lambda_1, \mathsf{unavail}(a_{\mathsf{S}}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}) :: \Gamma$ (by (T-Λ_4) and since $a_{\mathsf{S}} \notin \Lambda_1$ by well-formedness of Λ)

$\Psi;\, \Gamma \vDash \Theta_1, \mathsf{proc}(c_{\mathsf{L}}, \mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}, x_{\mathsf{L}} \leftarrow \mathsf{acquire}\ a_{\mathsf{S}}\,;Q_{x_{\mathsf{L}}}), \Theta_2 :: \Phi, \Delta$ (by inversion on (T-Ω))

$\Psi;\, \Gamma \vDash \Theta_1, \mathsf{proc}(c_{\mathsf{L}}, \mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}, x_{\mathsf{L}} \leftarrow \mathsf{acquire}\ a_{\mathsf{S}}\,;Q_{x_{\mathsf{L}}}) :: (\Phi_1, \Delta_1, c_{\mathsf{L}} : C_{\mathsf{L}}[\mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}])$ (by Lemma E.5)
for some C_{L}, Φ_1, *and* Δ_1

$\Psi;\, \Gamma \vDash \Theta_1 :: (\Phi_1, \Phi_2, \Delta_1, \Delta_2)$ (by inversion on (T-Θ_2))
for some Φ_2 *and* Δ_2

$\Psi;\, \Gamma;\, \Phi_2;\, \Delta_2 \vdash x_{\mathsf{L}} \leftarrow \mathsf{acquire}\, a_{\mathsf{S}}\,;Q_{x_{\mathsf{L}}} :: (c_{\mathsf{L}} : C_{\mathsf{L}}[\mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}])$ (by inversion on (T-Θ_2))

$\vdash (C_{\mathsf{L}}, \hat{D})\ \mathsf{sesync}$ *and* $(c_{\mathsf{S}} : \hat{D}[\mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}]) \in \Gamma$ and $\Psi \vdash C_{\mathsf{L}}[\mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}]\ \mathsf{type}$ (by inversion on (T-Θ_2))

$\Psi^* \vdash \mathsf{w}_{\mathsf{c}_2} \leq \mathsf{w}_{\mathsf{c}_3}$ *and* $\mathsf{Inv}_3(\mathsf{proc}(c_{\mathsf{L}}, \mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}, x_{\mathsf{L}} \leftarrow \mathsf{acquire}\ a_{\mathsf{S}}\,;Q_{x_{\mathsf{L}}}))$ *and*
$\mathsf{Inv}_4(\mathsf{proc}(c_{\mathsf{L}}, \mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}, x_{\mathsf{L}} \leftarrow \mathsf{acquire}\ a_{\mathsf{S}}\,;Q_{x_{\mathsf{L}}}))$ (by inversion on (T-Θ_2))

$\Psi;\, \Gamma;\, \Phi_2, x_{\mathsf{L}} : A_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}];\, \Delta_2 \vdash Q_{x_{\mathsf{L}}} :: (c_{\mathsf{L}} : C_{\mathsf{L}}[\mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}])$ (by inversion on (T-$\uparrow^{\mathsf{S}}_{\mathsf{L}}$L) since $a_{\mathsf{S}} : \uparrow^{\mathsf{S}}_{\mathsf{L}} A_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}] \in$

$\Psi^* \vdash \mathsf{w}_{\mathsf{c}_2} \leq \mathsf{w}_{\mathsf{a}_1} \leq \mathsf{w}_{\mathsf{c}_3}$ *and* $\Psi^+ \vdash \mathsf{w}_{\mathsf{c}_3} < \mathsf{w}_{\mathsf{a}_2}$ *and* $\forall b_{\mathsf{L}} : B_{\mathsf{L}}[\mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}] \in \Phi_2 : \mathsf{w}_{\mathsf{b}_1} < \mathsf{w}_{\mathsf{a}_1}$ (by inversion on (T-$\uparrow^{\mathsf{S}}_{\mathsf{L}}$L

$\Psi;\, \Gamma;\, \cdot\,;\, \cdot \vdash [a_{\mathsf{L}}/x_{\mathsf{L}}]\,P_{x_{\mathsf{L}}} :: (a_{\mathsf{L}} : A_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])$ (by Lemma E.2–1 and Corollary E.3)

$\vdash (A_{\mathsf{L}}, \uparrow^{\mathsf{S}}_{\mathsf{L}} A_{\mathsf{L}})\ \mathsf{sesync}$ (by inversion on (T-SESYNC$_{\uparrow^{\mathsf{S}}_{\mathsf{L}2}}$) since $\vdash (\uparrow^{\mathsf{S}}_{\mathsf{L}} A_{\mathsf{L}}, \top)\ \mathsf{sesync}$)

$\Psi \vdash A_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}]\ \mathsf{type}$ (by inversion on (T-$\mathsf{type}_{\uparrow^{\mathsf{S}}_{\mathsf{L}}}$))

$\mathsf{Inv}_3(\mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, [a_{\mathsf{L}}/x_{\mathsf{L}}]\,P_{x_{\mathsf{L}}}))$ *and* $\mathsf{Inv}_4(\mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, [a_{\mathsf{L}}/x_{\mathsf{L}}]\,P_{x_{\mathsf{L}}}))$
(since a_{L} does not have any descendants)

$\Psi;\, \Gamma \vDash \Theta_1, \mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, [a_{\mathsf{L}}/x_{\mathsf{L}}]\,P_{x_{\mathsf{L}}}) :: (\Phi_1, \Phi_2, \Delta_1, \Delta_2, a_{\mathsf{L}} : A_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])$ (by (T-Θ_2) and Corollary E.3)

$\Psi;\, \Gamma;\, \Phi_2, a_{\mathsf{L}} : A_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1} \updownarrow_{\mathsf{w}_{\mathsf{a}_2}}^{\mathsf{w}_{\mathsf{a}_3}}];\, \Delta_2 \vdash [a_{\mathsf{L}}/x_{\mathsf{L}}]\, Q_{x_{\mathsf{L}}} :: (c_{\mathsf{L}} : C_{\mathsf{L}}[\mathsf{w}_{\mathsf{c}_1} \updownarrow_{\mathsf{w}_{\mathsf{c}_2}}^{\mathsf{w}_{\mathsf{c}_3}}])$ (by Lemma E.2–3 and Corollary E.3)

Descendants of c_{L} increase by acquired channel $a_{\mathsf{L}} : A_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1} \updownarrow_{\mathsf{w}_{\mathsf{a}_2}}^{\mathsf{w}_{\mathsf{a}_3}}]$ (by previous)

$\mathsf{Inv}_3(\mathsf{proc}(a_{\mathsf{L}},\, \mathsf{w}_{\mathsf{a}_1} \updownarrow_{\mathsf{w}_{\mathsf{a}_2}}^{\mathsf{w}_{\mathsf{a}_3}},\, [a_{\mathsf{L}}/x_{\mathsf{L}}]\, Q_{x_{\mathsf{L}}}))$
(since $\Psi^* \vdash \mathsf{w}_{\mathsf{c}_2} \leq \mathsf{w}_{\mathsf{a}_1} \leq \mathsf{w}_{\mathsf{c}_3}$ and $\mathsf{Inv}_3(\mathsf{proc}(c_{\mathsf{L}},\, \mathsf{w}_{\mathsf{c}_1} \updownarrow_{\mathsf{w}_{\mathsf{c}_2}}^{\mathsf{w}_{\mathsf{c}_3}},\, x_{\mathsf{L}} \leftarrow \mathsf{acquire}\, a_{\mathsf{S}}\, ;\, Q_{x_{\mathsf{L}}}))$)

$\mathsf{Inv}_4(\mathsf{proc}(a_{\mathsf{L}},\, \mathsf{w}_{\mathsf{a}_1} \updownarrow_{\mathsf{w}_{\mathsf{a}_2}}^{\mathsf{w}_{\mathsf{a}_3}},\, [a_{\mathsf{L}}/x_{\mathsf{L}}]\, Q_{x_{\mathsf{L}}}))$
(since $\Psi^+ \vdash \mathsf{w}_{\mathsf{c}_3} < \mathsf{w}_{\mathsf{a}_2}$ and $\mathsf{Inv}_4(\mathsf{proc}(c_{\mathsf{L}},\, \mathsf{w}_{\mathsf{c}_1} \updownarrow_{\mathsf{w}_{\mathsf{c}_2}}^{\mathsf{w}_{\mathsf{c}_3}},\, x_{\mathsf{L}} \leftarrow \mathsf{acquire}\, a_{\mathsf{S}}\, ;\, Q_{x_{\mathsf{L}}}))$)

$\Psi;\, \Gamma \vDash \Theta_1, \mathsf{proc}(a_{\mathsf{L}},\, \mathsf{w}_{\mathsf{a}_1} \updownarrow_{\mathsf{w}_{\mathsf{a}_2}}^{\mathsf{w}_{\mathsf{a}_3}},\, [a_{\mathsf{L}}/x_{\mathsf{L}}]\, P_{x_{\mathsf{L}}}), \mathsf{proc}(c_{\mathsf{L}},\, \mathsf{w}_{\mathsf{c}_1} \updownarrow_{\mathsf{w}_{\mathsf{c}_2}}^{\mathsf{w}_{\mathsf{c}_3}},\, [a_{\mathsf{L}}/x_{\mathsf{L}}]\, Q_{x_{\mathsf{L}}}) :: (\Phi_1, \Delta_1, c_{\mathsf{L}} : C_{\mathsf{L}}[\mathsf{w}_{\mathsf{c}_1} \updownarrow_{\mathsf{w}_{\mathsf{c}_2}}^{\mathsf{w}_{\mathsf{c}_3}}])$ (by (T-Θ_2)

$\Psi;\, \Gamma \vDash \Theta_1, \mathsf{proc}(a_{\mathsf{L}},\, \mathsf{w}_{\mathsf{a}_1} \updownarrow_{\mathsf{w}_{\mathsf{a}_2}}^{\mathsf{w}_{\mathsf{a}_3}},\, [a_{\mathsf{L}}/x_{\mathsf{L}}]\, P_{x_{\mathsf{L}}}), \mathsf{proc}(c_{\mathsf{L}},\, \mathsf{w}_{\mathsf{c}_1} \updownarrow_{\mathsf{w}_{\mathsf{c}_2}}^{\mathsf{w}_{\mathsf{c}_3}},\, [a_{\mathsf{L}}/x_{\mathsf{L}}]\, Q_{x_{\mathsf{L}}}), \Theta_2 :: \Phi, \Delta$ (by Lemma E.6)

$\Psi;\, \Gamma \vDash \Lambda_1, \mathsf{unavail}(a_{\mathsf{S}},\, \mathsf{w}_{\mathsf{a}_1} \updownarrow_{\mathsf{w}_{\mathsf{a}_2}}^{\mathsf{w}_{\mathsf{a}_3}}); \Theta_1, \mathsf{proc}(a_{\mathsf{L}},\, \mathsf{w}_{\mathsf{a}_1} \updownarrow_{\mathsf{w}_{\mathsf{a}_2}}^{\mathsf{w}_{\mathsf{a}_3}},\, [a_{\mathsf{L}}/x_{\mathsf{L}}]\, P_{x_{\mathsf{L}}}), \mathsf{proc}(c_{\mathsf{L}},\, \mathsf{w}_{\mathsf{c}_1} \updownarrow_{\mathsf{w}_{\mathsf{c}_2}}^{\mathsf{w}_{\mathsf{c}_3}},\, [a_{\mathsf{L}}/x_{\mathsf{L}}]\, Q_{x_{\mathsf{L}}}), \Theta_2 :: \Gamma; \Phi, \Delta$
(by (T-Ω) and well-formedness preserved)

3. (D-$\mathbf{1}$) : $\mathsf{proc}(a_{\mathsf{L}},\, \mathsf{w}_{\mathsf{a}_1} \updownarrow_{\mathsf{w}_{\mathsf{a}_2}}^{\mathsf{w}_{\mathsf{a}_3}},\, \mathsf{close}\, a_{\mathsf{L}}), \mathsf{proc}(c_{\mathsf{L}},\, \mathsf{w}_{\mathsf{c}_1} \updownarrow_{\mathsf{w}_{\mathsf{c}_2}}^{\mathsf{w}_{\mathsf{c}_3}},\, \mathsf{wait}\, a_{\mathsf{L}}\, ;\, Q) \longrightarrow \mathsf{proc}(c_{\mathsf{L}},\, \mathsf{w}_{\mathsf{c}_1} \updownarrow_{\mathsf{w}_{\mathsf{c}_2}}^{\mathsf{w}_{\mathsf{c}_3}},\, Q)$

$\Psi;\, \Gamma \vDash \Lambda;\, \Theta_1, \mathsf{proc}(a_{\mathsf{L}},\, \mathsf{w}_{\mathsf{a}_1} \updownarrow_{\mathsf{w}_{\mathsf{a}_2}}^{\mathsf{w}_{\mathsf{a}_3}},\, \mathsf{close}\, a_{\mathsf{L}}), \Theta_2, \mathsf{proc}(c_{\mathsf{L}},\, \mathsf{w}_{\mathsf{c}_1} \updownarrow_{\mathsf{w}_{\mathsf{c}_2}}^{\mathsf{w}_{\mathsf{c}_3}},\, \mathsf{wait}\, a_{\mathsf{L}}\, ;\, Q), \Theta_3 :: \Gamma;\, \Phi, \Delta$ (by assumption)
for some, Θ_1, Θ_2, *and* Θ_3

$\Psi;\, \Gamma \vDash \Lambda;\, \Theta_1, \Theta_2, \mathsf{proc}(a_{\mathsf{L}},\, \mathsf{w}_{\mathsf{a}_1} \updownarrow_{\mathsf{w}_{\mathsf{a}_2}}^{\mathsf{w}_{\mathsf{a}_3}},\, \mathsf{close}\, a_{\mathsf{L}}), \mathsf{proc}(c_{\mathsf{L}},\, \mathsf{w}_{\mathsf{c}_1} \updownarrow_{\mathsf{w}_{\mathsf{c}_2}}^{\mathsf{w}_{\mathsf{c}_3}},\, \mathsf{wait}\, a_{\mathsf{L}}\, ;\, Q), \Theta_3 :: \Gamma;\, \Phi, \Delta$ (by Lemma E.4)

$\Lambda;\, \Theta_1, \Theta_2, \mathsf{proc}(a_{\mathsf{L}},\, \mathsf{w}_{\mathsf{a}_1} \updownarrow_{\mathsf{w}_{\mathsf{a}_2}}^{\mathsf{w}_{\mathsf{a}_3}},\, \mathsf{close}\, a_{\mathsf{L}}), \mathsf{proc}(c_{\mathsf{L}},\, \mathsf{w}_{\mathsf{c}_1} \updownarrow_{\mathsf{w}_{\mathsf{c}_2}}^{\mathsf{w}_{\mathsf{c}_3}},\, \mathsf{wait}\, a_{\mathsf{L}}\, ;\, Q), \Theta_3$ (this case)
$\longrightarrow \Lambda;\, \Theta_1, \Theta_2, \mathsf{proc}(c_{\mathsf{L}},\, \mathsf{w}_{\mathsf{c}_1} \updownarrow_{\mathsf{w}_{\mathsf{c}_2}}^{\mathsf{w}_{\mathsf{c}_3}},\, Q), \Theta_3$

$\Psi;\, \Gamma \vDash \Lambda :: \Gamma$ (by inversion (T-Ω))

$\Psi;\, \Gamma \vDash \Theta_1, \Theta_2, \mathsf{proc}(a_{\mathsf{L}},\, \mathsf{w}_{\mathsf{a}_1} \updownarrow_{\mathsf{w}_{\mathsf{a}_2}}^{\mathsf{w}_{\mathsf{a}_3}},\, \mathsf{close}\, a_{\mathsf{L}}), \mathsf{proc}(c_{\mathsf{L}},\, \mathsf{w}_{\mathsf{c}_1} \updownarrow_{\mathsf{w}_{\mathsf{c}_2}}^{\mathsf{w}_{\mathsf{c}_3}},\, \mathsf{wait}\, a_{\mathsf{L}}\, ;\, Q), \Theta_3 :: \Phi, \Delta$ (by inversion (T-Ω))

$\Psi;\, \Gamma \vDash \Theta_1, \Theta_2, \mathsf{proc}(a_{\mathsf{L}},\, \mathsf{w}_{\mathsf{a}_1} \updownarrow_{\mathsf{w}_{\mathsf{a}_2}}^{\mathsf{w}_{\mathsf{a}_3}},\, \mathsf{close}\, a_{\mathsf{L}}), \mathsf{proc}(c_{\mathsf{L}},\, \mathsf{w}_{\mathsf{c}_1} \updownarrow_{\mathsf{w}_{\mathsf{c}_2}}^{\mathsf{w}_{\mathsf{c}_3}},\, \mathsf{wait}\, a_{\mathsf{L}}\, ;\, Q) :: (\Phi_1, \Delta_1, c_{\mathsf{L}} : C_{\mathsf{L}}[\mathsf{w}_{\mathsf{c}_1} \updownarrow_{\mathsf{w}_{\mathsf{c}_2}}^{\mathsf{w}_{\mathsf{c}_3}}])$ (by Lemm
for some C_{L}, Φ_1, *and* Δ_1

$\Psi;\, \Gamma \vDash \Theta_1, \Theta_2, \mathsf{proc}(a_{\mathsf{L}},\, \mathsf{w}_{\mathsf{a}_1} \updownarrow_{\mathsf{w}_{\mathsf{a}_2}}^{\mathsf{w}_{\mathsf{a}_3}},\, \mathsf{close}\, a_{\mathsf{L}}) :: (\Phi_1, \Phi_2, \Delta_1, \Delta_2)$ (by inversion on (T-Θ_2))
for some Φ_2, *and* Δ_2

$\Psi;\, \Gamma;\, \Phi_2;\, \Delta_2 \vdash \mathsf{wait}\, a_{\mathsf{L}}\, ; Q :: (c_{\mathsf{L}} : C_{\mathsf{L}}[\mathsf{w}_{\mathsf{c}_1} \updownarrow_{\mathsf{w}_{\mathsf{c}_2}}^{\mathsf{w}_{\mathsf{c}_3}}])$ (by inversion on (T-Θ_2))

$\vdash (C_{\mathsf{L}}, \hat{D})\ \mathsf{sesync}$ *and* $(c_{\mathsf{S}} : \hat{D}[\mathsf{w}_{\mathsf{c}_1} \updownarrow_{\mathsf{w}_{\mathsf{c}_2}}^{\mathsf{w}_{\mathsf{c}_3}}]) \in \Gamma$ *and* $\Psi \vdash C_{\mathsf{L}}[\mathsf{w}_{\mathsf{c}_1} \updownarrow_{\mathsf{w}_{\mathsf{c}_2}}^{\mathsf{w}_{\mathsf{c}_3}}]\ \mathsf{type}$ (by inversion on (T-Θ_2))

$\Psi^* \vdash \mathsf{w}_{\mathsf{c}_2} \leq \mathsf{w}_{\mathsf{c}_3}$ *and* $\mathsf{Inv}_3(\mathsf{proc}(c_{\mathsf{L}},\, \mathsf{w}_{\mathsf{c}_1} \updownarrow_{\mathsf{w}_{\mathsf{c}_2}}^{\mathsf{w}_{\mathsf{c}_3}},\, \mathsf{wait}\, a_{\mathsf{L}}\, ;\, Q))$ *and* $\mathsf{Inv}_4(\mathsf{proc}(c_{\mathsf{L}},\, \mathsf{w}_{\mathsf{c}_1} \updownarrow_{\mathsf{w}_{\mathsf{c}_2}}^{\mathsf{w}_{\mathsf{c}_3}},\, \mathsf{wait}\, a_{\mathsf{L}}\, ;\, Q))$
(by inversion on (T-Θ_2))

$\Psi;\, \Gamma;\, \Phi_2;\, \Delta_3 \vdash Q :: (c_{\mathsf{L}} : C_{\mathsf{L}}[\mathsf{w}_{\mathsf{c}_1} \updownarrow_{\mathsf{w}_{\mathsf{c}_2}}^{\mathsf{w}_{\mathsf{c}_3}}])$ (by inversion on (T-$\mathbf{1}_{\mathrm{L}}$))
for some Δ_3 *such that* $\Delta_2 = \Delta_3, a_{\mathsf{L}} : \mathbf{1}[\mathsf{w}_{\mathsf{a}_1} \updownarrow_{\mathsf{w}_{\mathsf{a}_2}}^{\mathsf{w}_{\mathsf{a}_3}}]$

$\Psi;\, \Gamma \vDash \Theta_1, \Theta_2 :: (\Phi_1, \Phi_2, \Phi_3, \Delta_1, \Delta_3, \Delta_4)$ (by inversion on (T-Θ_2))
for some Φ_3 *and* Δ_4

$\Psi;\, \Gamma;\, \Phi_3\, ;\, \Delta_4 \vdash \mathsf{close}\, a_{\mathsf{L}} :: (a_{\mathsf{L}} : \mathbf{1}[\mathsf{w}_{\mathsf{a}_1} \updownarrow_{\mathsf{w}_{\mathsf{a}_2}}^{\mathsf{w}_{\mathsf{a}_3}}])$ (by inversion on (T-Θ_2))

$\vdash (\mathbf{1}, \hat{B})\ \mathsf{sesync}$ *and* $(a_{\mathsf{S}} : \hat{B}[\mathsf{w}_{\mathsf{a}_1} \updownarrow_{\mathsf{w}_{\mathsf{a}_2}}^{\mathsf{w}_{\mathsf{a}_3}}]) \in \Gamma$ *and* $\Psi \vdash \mathbf{1}[\mathsf{w}_{\mathsf{a}_1} \updownarrow_{\mathsf{w}_{\mathsf{a}_2}}^{\mathsf{w}_{\mathsf{a}_3}}]\ \mathsf{type}$ (by inversion on (T-Θ_2))

$\mathsf{Inv}_3(\mathsf{proc}(a_{\mathsf{L}},\, \mathsf{w}_{\mathsf{a}_1} \updownarrow_{\mathsf{w}_{\mathsf{a}_2}}^{\mathsf{w}_{\mathsf{a}_3}},\, \mathsf{close}\, a_{\mathsf{L}}))$ *and* $\mathsf{Inv}_4(\mathsf{proc}(a_{\mathsf{L}},\, \mathsf{w}_{\mathsf{a}_1} \updownarrow_{\mathsf{w}_{\mathsf{a}_2}}^{\mathsf{w}_{\mathsf{a}_3}},\, \mathsf{close}\, a_{\mathsf{L}}))$ (by inversion on (T-Θ_2))

$\Phi_3 = (\cdot)$ *and* $\Delta_4 = (\cdot)$ (by inversion on (T-$\mathbf{1}_{\mathrm{R}}$))

$(\Psi;\, \Gamma \vDash \Theta_1, \Theta_2 :: (\Phi_1, \Phi_2, \Phi_3, \Delta_1, \Delta_3, \Delta_4)) \rhd c_{\mathsf{L}} \subset$ (by Definition B.1)
$(\Psi;\, \Gamma \vDash \Theta_1, \Theta_2, \mathsf{proc}(a_{\mathsf{L}},\, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}},\, \mathsf{close}\ a_{\mathsf{L}}) :: (\Phi_1, \Phi_2, \Delta_1, \Delta_2)) \rhd c_{\mathsf{L}}$

$\mathsf{dom}(\Phi_2; \Delta_3) \subset \mathsf{dom}(\Phi_2; \Delta_2)$ (by previous)

$\mathsf{Inv}_3(\mathsf{proc}(c_{\mathsf{L}},\, \mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}},\, Q))$ *and* $\mathsf{Inv}_4(\mathsf{proc}(c_{\mathsf{L}},\, \mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}},\, Q))$ (since set of descendants of c_{L} decreased and)
$(\mathsf{Inv}_3(\mathsf{proc}(c_{\mathsf{L}},\, \mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}},\, \mathsf{wait}\ a_{\mathsf{L}}\,;\, Q))$ and $\mathsf{Inv}_4(\mathsf{proc}(c_{\mathsf{L}},\, \mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}},\, \mathsf{wait}\ a_{\mathsf{L}}\,;\, Q)))$

$\Psi;\, \Gamma \vDash \Theta_1, \Theta_2, \mathsf{proc}(c_{\mathsf{L}},\, \mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}},\, Q) :: (\Phi_1, \Delta_1, c_{\mathsf{L}} : C_{\mathsf{L}}[\mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}])$ (by (T-Θ_2))

$\Psi;\, \Gamma \vDash \Theta_1, \Theta_2, \mathsf{proc}(c_{\mathsf{L}},\, \mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}},\, Q), \Theta_3 :: \Phi, \Delta$ (by Lemma E.6)

$\Psi;\, \Gamma \vDash \Lambda;\, \Theta_1, \Theta_2, \mathsf{proc}(c_{\mathsf{L}},\, \mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}},\, Q), \Theta_3 :: \Gamma; \Phi, \Delta$ (by (T-Ω) and well-formedness preserved)

4. (D-$\otimes$) : $\mathsf{proc}(a_{\mathsf{L}},\, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}},\, \mathsf{send}\ a_{\mathsf{L}}\ b_{\mathsf{L}}\,;\, P),\, \mathsf{proc}(c_{\mathsf{L}},\, \mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}},\, y_{\mathsf{L}} \leftarrow \mathsf{recv}\ a_{\mathsf{L}}\,;\, Q_{y_{\mathsf{L}}})$
$\longrightarrow\ \mathsf{proc}(a_{\mathsf{L}},\, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}},\, P),\, \mathsf{proc}(c_{\mathsf{L}},\, \mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}},\, [b_{\mathsf{L}}/y_{\mathsf{L}}]\, Q_{y_{\mathsf{L}}})$

$\Psi;\, \Gamma \vDash \Lambda;\, \Theta_1, \mathsf{proc}(a_{\mathsf{L}},\, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}},\, \mathsf{send}\ a_{\mathsf{L}}\ b_{\mathsf{L}}\,;\, P), \Theta_2, \mathsf{proc}(c_{\mathsf{L}},\, \mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}},\, y_{\mathsf{L}} \leftarrow \mathsf{recv}\ a_{\mathsf{L}}\,;\, Q_{y_{\mathsf{L}}}), \Theta_3 :: \Gamma;\, \Phi, \Delta$
for some, Θ_1*,* Θ_2*, and* Θ_3 (by assumption)

$\Psi;\, \Gamma \vDash \Lambda;\, \Theta_1, \Theta_2, \mathsf{proc}(a_{\mathsf{L}},\, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}},\, \mathsf{send}\ a_{\mathsf{L}}\ b_{\mathsf{L}}\,;\, P), \mathsf{proc}(c_{\mathsf{L}},\, \mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}},\, y_{\mathsf{L}} \leftarrow \mathsf{recv}\ a_{\mathsf{L}}\,;\, Q_{y_{\mathsf{L}}}), \Theta_3 :: \Gamma;\, \Phi, \Delta$
(by Lemma E.4)

$\Lambda;\, \Theta_1, \Theta_2, \mathsf{proc}(a_{\mathsf{L}},\, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}},\, \mathsf{send}\ a_{\mathsf{L}}\ b_{\mathsf{L}}\,;\, P), \mathsf{proc}(c_{\mathsf{L}},\, \mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}},\, y_{\mathsf{L}} \leftarrow \mathsf{recv}\ a_{\mathsf{L}}\,;\, Q_{y_{\mathsf{L}}}), \Theta_3$ (this case)
$\longrightarrow\ \Lambda;\, \Theta_1, \Theta_2, \mathsf{proc}(a_{\mathsf{L}},\, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}},\, P), \mathsf{proc}(c_{\mathsf{L}},\, \mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}},\, [b_{\mathsf{L}}/y_{\mathsf{L}}]\, Q_{y_{\mathsf{L}}}), \Theta_3$

$\Psi;\, \Gamma \vDash \Lambda :: \Gamma$ (by inversion (T-Ω))

$\Psi;\, \Gamma \vDash \Theta_1, \Theta_2, \mathsf{proc}(a_{\mathsf{L}},\, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}},\, \mathsf{send}\ a_{\mathsf{L}}\ b_{\mathsf{L}}\,;\, P), \mathsf{proc}(c_{\mathsf{L}},\, \mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}},\, y_{\mathsf{L}} \leftarrow \mathsf{recv}\ a_{\mathsf{L}}\,;\, Q_{y_{\mathsf{L}}}), \Theta_3 :: \Phi, \Delta$
(by inversion (T-Ω))

$\Psi;\, \Gamma \vDash \Theta_1, \Theta_2, \mathsf{proc}(a_{\mathsf{L}},\, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}},\, \mathsf{send}\ a_{\mathsf{L}}\ b_{\mathsf{L}}\,;\, P), \mathsf{proc}(c_{\mathsf{L}},\, \mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}},\, y_{\mathsf{L}} \leftarrow \mathsf{recv}\ a_{\mathsf{L}}\,;\, Q_{y_{\mathsf{L}}}) :: (\Phi_1, \Delta_1, c_{\mathsf{L}} : C_{\mathsf{L}}[\mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}]$
for some C_{L}*,* Φ_1*, and* Δ_1 (by Lemma E.5)

$\Psi;\, \Gamma \vDash \Theta_1, \Theta_2, \mathsf{proc}(a_{\mathsf{L}},\, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}},\, \mathsf{send}\ a_{\mathsf{L}}\ b_{\mathsf{L}}\,;\, P) :: (\Phi_1, \Phi_2, \Delta_1, \Delta_2)$ (by inversion on (T-Θ_2))
for some Φ_2 *and* Δ_2

$\Psi;\, \Gamma;\, \Phi_2;\, \Delta_2 \vdash y_{\mathsf{L}} \leftarrow \mathsf{recv}\ a_{\mathsf{L}}\,;\, Q_{y_{\mathsf{L}}} :: (c_{\mathsf{L}} : C_{\mathsf{L}}[\mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}])$ (by inversion on (T-Θ_2))

$\vdash (C_{\mathsf{L}}, \hat{D})\ \mathsf{sesync}$ *and* $\Psi \vdash C_{\mathsf{L}}[\mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}]\ \mathsf{type}$ (by inversion on (T-Θ_2))
for some $(c_{\mathsf{s}} : \hat{D}[\mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}]) \in \Gamma$

$\Psi^* \vdash \mathsf{w}_{\mathsf{c}_2} \leq \mathsf{w}_{\mathsf{c}_3}$ *and* $\mathsf{Inv}_3(\mathsf{proc}(c_{\mathsf{L}},\, \mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}},\, y_{\mathsf{L}} \leftarrow \mathsf{recv}\ a_{\mathsf{L}}\,;\, Q_{y_{\mathsf{L}}}))$ *and* $\mathsf{Inv}_4(\mathsf{proc}(c_{\mathsf{L}},\, \mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}},\, y_{\mathsf{L}} \leftarrow \mathsf{recv}\ a_{\mathsf{L}}\,;\, Q_{y_{\mathsf{L}}}))$
(by inversion on (T-Θ_2))

$\Psi;\, \Gamma \vDash \Theta_1, \Theta_2 :: (\Phi_1, \Phi_2, \Phi_3, \Delta_1, \Delta_2, \Delta_3)$ (by inversion on (T-Θ_2))
for some Φ_3 *and* Δ_3

$\Psi;\, \Gamma;\, \Phi_3;\, \Delta_3 \vdash \mathsf{send}\ a_{\mathsf{L}}\ b_{\mathsf{L}}\,;\, P :: (a_{\mathsf{L}} : A_{\mathsf{L}}@\omega_{b_1}\updownarrow^{\omega_{b_3}}_{\omega_{b_2}} \otimes B_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])$(by inversion on (T-$\Theta_2$))

$\Phi_3 = (\cdot)$ *and* $\Delta_3 = \Delta_4, b_{\mathsf{L}} : A_{\mathsf{L}}[\mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}]$ *and* $\Psi;\, \Gamma;\, \Phi_3;\, \Delta_4 \vdash P :: (a_{\mathsf{L}} : B_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])$ (by inversion on (T-$\otimes_{\mathrm{R}}$))
for some Δ_4*,* $\mathsf{w}_{\mathsf{b}_1}$*,* $\mathsf{w}_{\mathsf{b}_2}$*, and* $\mathsf{w}_{\mathsf{b}_3}$

$a_{\mathsf{L}} : A_{\mathsf{L}}@\mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}} \otimes B_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}]$ (by inversion on (T-$\otimes_{\mathrm{R}}$))

$\Psi;\, \Gamma;\, \Phi_2;\, \Delta_2 \vdash y_{\mathsf{L}} \leftarrow \mathsf{recv}\, a_{\mathsf{L}}\,;Q_{y_{\mathsf{L}}} :: (c_{\mathsf{L}} : C_{\mathsf{L}}[\mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}])$ (by previous)
for some Φ_5, Φ'_5, Δ_5, *and* Δ'_5 *such that either*
$\Phi_2 = \Phi_5$ *and* $\Phi'_5 = \Phi_5$ *and* $\Delta_2 = \Delta_5, a_{\mathsf{L}} : A_{\mathsf{L}}@\mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}} \otimes B_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}]$ *and* $\Delta'_5 = \Delta_5, a_{\mathsf{L}} : B_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}]$ *or*
$\Phi_2 = \Phi_5, a_{\mathsf{L}} : A_{\mathsf{L}}@\mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}} \otimes B_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}]$ *and* $\Phi'_5 = \Phi_5, a_{\mathsf{L}} : B_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}]$ *and* $\Delta_2 = \Delta_5$ *and* $\Delta'_5 = \Delta_5$

$\Psi;\, \Gamma;\, \Phi'_5;\, \Delta'_5, y_{\mathsf{L}} : A_{\mathsf{L}}[\mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}] \vdash Q_{y_{\mathsf{L}}} :: (c_{\mathsf{L}} : C_{\mathsf{L}}[\mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}])$ (by inversion on (T-$\otimes_{\mathrm{L}_1}$) and (T-$\otimes_{\mathrm{L}_2}$), resp.)

$\vdash (A_{\mathsf{L}} \otimes B_{\mathsf{L}}, \hat{B})\ \mathsf{sesync}$ *and* $\Psi \vdash A_{\mathsf{L}}@\mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}} \otimes B_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}]\ \mathsf{type}$
for some $(a_{\mathsf{S}} : \hat{B}[\mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}]) \in \Gamma$ (by inversion on (T-Θ_2))

$\Psi^* \vdash \mathsf{w}_{\mathsf{a}_2} \leq \mathsf{w}_{\mathsf{a}_3}$ *and* $\mathsf{Inv}_3(\mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, \mathsf{send}\ a_{\mathsf{L}}\ b_{\mathsf{L}}\,;P))$ *and* $\mathsf{Inv}_4(\mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, \mathsf{send}\ a_{\mathsf{L}}\ b_{\mathsf{L}}\,;P))$ (by inversion on (T-Θ_2))

$\mathsf{dom}(\Phi_3;\Delta_4) \subset \mathsf{dom}(\Phi_3;\Delta_3)$ (by previous)

$\mathsf{Inv}_3(\mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P))$ *and* $\mathsf{Inv}_4(\mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P))$ (since set of descendants of a_{L} decreased and)
($\mathsf{Inv}_3(\mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, \mathsf{send}\ a_{\mathsf{L}}\ b_{\mathsf{L}}\,;P))$ and $\mathsf{Inv}_4(\mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, \mathsf{send}\ a_{\mathsf{L}}\ b_{\mathsf{L}}\,;P))$)

$\vdash (B_{\mathsf{L}}, \hat{B})\ \mathsf{sesync}$ (by inversion on (T-SESYNC$_{\otimes}$))

$\Psi \vdash B_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}]\ \mathsf{type}$ *and* $\Psi^+ \vdash \mathsf{w}_{\mathsf{a}_3} < \mathsf{w}_{\mathsf{b}_2}$ (by inversion on (T-$\mathsf{type}_{\otimes}$))

$\Psi;\, \Gamma \vDash \Theta_1, \Theta_2, \mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P) :: (\Phi_1, \Phi'_5, \Delta_1, \Delta'_5, b_{\mathsf{L}} : A_{\mathsf{L}}[\mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}], a_{\mathsf{L}} : B_{\mathsf{L}}[\mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}])$ (by (T-Θ_2))

$\mathsf{dom}(\Phi'_5;\Delta'_5, y_{\mathsf{L}} : A_{\mathsf{L}}[\mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}]) = \mathsf{dom}(\Phi_2;\Delta_2) \cup \mathsf{dom}(y_{\mathsf{L}} : A_{\mathsf{L}}[\mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}])$ (by previous)

$\mathsf{Inv}_3(\mathsf{proc}(c_{\mathsf{L}}, \mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}, Q_{y_{\mathsf{L}}}))$ (since $y_{\mathsf{L}} \notin \Phi'_5$ and since $\mathsf{Inv}_3(\mathsf{proc}(c_{\mathsf{L}}, \mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}, y_{\mathsf{L}} \leftarrow \mathsf{recv}\ a_{\mathsf{L}}\,;Q_{y_{\mathsf{L}}}))$)

$\Psi^+ \vdash \mathsf{w}_{\mathsf{c}_3} < \mathsf{w}_{\mathsf{a}_2}$ (since $\mathsf{Inv}_4(\mathsf{proc}(c_{\mathsf{L}}, \mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}, y_{\mathsf{L}} \leftarrow \mathsf{recv}\ a_{\mathsf{L}}\,;Q_{y_{\mathsf{L}}}))$)

$\Psi^+ \vdash \mathsf{w}_{\mathsf{c}_3} < \mathsf{w}_{\mathsf{b}_2}$ (since $\Psi^+ \vdash \mathsf{w}_{\mathsf{c}_3} < \mathsf{w}_{\mathsf{a}_2}$, $\Psi^* \vdash \mathsf{w}_{\mathsf{a}_2} \leq \mathsf{w}_{\mathsf{a}_3}$, $\Psi^+ \vdash \mathsf{w}_{\mathsf{a}_3} < \mathsf{w}_{\mathsf{b}_2}$)

$\mathsf{Inv}_4(\mathsf{proc}(c_{\mathsf{L}}, \mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}, Q_{y_{\mathsf{L}}}))$ (since $\Psi^+ \vdash \mathsf{w}_{\mathsf{c}_3} < \mathsf{w}_{\mathsf{b}_2}$ and since $\mathsf{Inv}_4(\mathsf{proc}(c_{\mathsf{L}}, \mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}, y_{\mathsf{L}} \leftarrow \mathsf{recv}\ a_{\mathsf{L}}\,;Q_{y_{\mathsf{L}}}))$)

$\Psi;\, \Gamma;\, \Phi'_5;\, \Delta'_5, b_{\mathsf{L}} : A_{\mathsf{L}}[\mathsf{w}_{\mathsf{b}_1}\updownarrow^{\mathsf{w}_{\mathsf{b}_3}}_{\mathsf{w}_{\mathsf{b}_2}}] \vdash [b_{\mathsf{L}}/y_{\mathsf{L}}]\, Q_{y_{\mathsf{L}}} :: (c_{\mathsf{L}} : C_{\mathsf{L}}[\mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}])$ (by Lemma E.2–5)

$\Psi;\, \Gamma \vDash \Theta_1, \Theta_2, \mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P), \mathsf{proc}(c_{\mathsf{L}}, \mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}, [b_{\mathsf{L}}/y_{\mathsf{L}}]\, Q_{y_{\mathsf{L}}}) :: (\Phi_1, \Delta_1, c_{\mathsf{L}} : C_{\mathsf{L}}[\mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}])$ (by (T-Θ_2))

$\Psi;\, \Gamma \vDash \Theta_1, \Theta_2, \mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P), \mathsf{proc}(c_{\mathsf{L}}, \mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}, [b_{\mathsf{L}}/y_{\mathsf{L}}]\, Q_{y_{\mathsf{L}}}), \Theta_3 :: (\Phi, \Delta)$ (by Lemma E.6)

$\Psi;\, \Gamma \vDash \Lambda;\, \Theta_1, \Theta_2, \mathsf{proc}(a_{\mathsf{L}}, \mathsf{w}_{\mathsf{a}_1}\updownarrow^{\mathsf{w}_{\mathsf{a}_3}}_{\mathsf{w}_{\mathsf{a}_2}}, P), \mathsf{proc}(c_{\mathsf{L}}, \mathsf{w}_{\mathsf{c}_1}\updownarrow^{\mathsf{w}_{\mathsf{c}_3}}_{\mathsf{w}_{\mathsf{c}_2}}, [b_{\mathsf{L}}/y_{\mathsf{L}}]\, Q_{y_{\mathsf{L}}}), \Theta_3 :: \Gamma;\Phi, \Delta$
(by (T-Ω) and well-formedness preserved)

□