

Minimal Session Types for the π -calculus (Extended Version)

Alen Arslanagić
University of Groningen
The Netherlands

Anda-Amelia Palamariuc
University of Groningen
The Netherlands

Jorge A. Pérez
University of Groningen and CWI
The Netherlands

ABSTRACT

Session types enable the static verification of message-passing programs. A session type specifies a channel’s protocol as *sequences* of messages. Prior work established a *minimality result*: every process typable with standard session types can be compiled down to a process typable using *minimal session types*: session types without the sequencing construct. This result justifies session types in terms of themselves; it holds for a higher-order session π -calculus, where values are abstractions (functions from names to processes).

This paper establishes a new minimality result but now for the session π -calculus, the language in which values are names and for which session types have been more widely studied. Remarkably, this new minimality result can be obtained by composing known results. We develop optimizations of our new minimality result, and establish its static and dynamic correctness.

CCS CONCEPTS

• **Theory of computation** → **Process calculi; Type structures.**

KEYWORDS

Concurrency, Session Types, Process Calculi, Expressiveness

1 INTRODUCTION

Session types are a type-based approach to statically ensure correct message-passing programs [5, 6]. A session type stipulates the sequence and payload of the messages exchanged along a channel. In this paper, we investigate a *minimal formulation* of session types for the π -calculus, the paradigmatic calculus of concurrency. This elementary formulation is called *minimal session types* (MSTs).

The gap between standard and minimal session types concerns *sequentiality* in types. Sequentiality, denoted by the action prefix ‘;’, is arguably the most distinguishing construct of session types. For instance, in the session type $S = ?(\text{Int}); ?(\text{Int}); !(\text{Bool}); \text{end}$, this construct conveniently specifies a channel protocol that *first* receives (?) two integers, *then* sends (!) a Boolean, and *finally* ends.

Because sequentiality is so useful for protocol specification and verification, a natural question is whether it could be recovered by other means. To this end, Arslanagić et al. [1] defined MSTs as the sub-class of session types without the ‘;’ construct. They established a *minimality result*: every well-typed session process can be *decomposed* into a process typable with MSTs. Their result is inspired by Parrow’s work on *trios processes* for the π -calculus [11]. The minimality result justifies session types in terms of themselves, and shows that sequentiality in types is useful but not indispensable, because it can be precisely mimicked by the process decomposition.

The minimality result in [1] holds for a *higher-order* process calculus called HO, in which values are only abstractions (functions from names to processes). HO does not include name passing nor process recursion, but it can encode them precisely [8, 10]. An

important question left open in [1] is whether the minimality result holds for the session π -calculus (dubbed π), the language in which values are names and for which session types have been more widely studied from foundational and practical perspectives.

In this paper, we positively answer this question. Our approach is simple, perhaps even deceptively so. In order to establish the minimality result for π , we compose the decomposition in [1] with the mutual encodings between HO and π given in [8, 10].

Let $\mu\pi$ and μHO denote the process languages π and HO with MSTs (rather than with standard session types). Also, let $\mathcal{D}(\cdot)$ denote the decomposition function from HO to μHO defined in [1]. Kouzapas et al. [8, 10] gave typed encodings from π to HO (denoted $\llbracket \cdot \rrbracket_g^1$) and from HO to π (denoted $\llbracket \cdot \rrbracket^2$). Therefore, given $\mathcal{D}(\cdot)$, $\llbracket \cdot \rrbracket_g^1$,

and $\llbracket \cdot \rrbracket^2$, to define a decomposition function $\mathcal{F}(\cdot) : \pi \rightarrow \mu\pi$, it suffices to follow Figure 1. This is sound for our purposes, because $\llbracket \cdot \rrbracket_g^1$ and $\llbracket \cdot \rrbracket^2$ preserve sequentiality in processes and their types.

The *first contribution* of this paper is the decomposition function $\mathcal{F}(\cdot)$, whose correctness follows from that of its constituent functions. $\mathcal{F}(\cdot)$ is significant, as it provides an elegant, positive answer to the question of whether the minimality result in [1] holds for π . Indeed, it proves that the values exchanged do not influence sequentiality in session types: the minimality result of [1] is not specific to the abstraction-passing language HO.

However, $\mathcal{F}(\cdot)$ is not entirely satisfactory. A side effect of composing $\mathcal{D}(\cdot)$, $\llbracket \cdot \rrbracket_g^1$, and $\llbracket \cdot \rrbracket^2$ is that the resulting decomposition of π into $\mu\pi$ is inefficient, as it includes redundant synchronizations. These shortcomings are particularly noticeable in the treatment of recursive processes. The *second contribution* of this paper is an optimized variant of $\mathcal{F}(\cdot)$, dubbed $\mathcal{F}^*(\cdot)$, in which we remove redundant synchronizations and target recursive processes and variables directly, exploiting the fact that π supports recursion natively.

Contributions. The main contributions of this paper are:

- (1) A minimality result for π , based on the function $\mathcal{F}(\cdot)$.
- (2) $\mathcal{F}^*(\cdot)$, an optimized variant of $\mathcal{F}(\cdot)$, without redundant communications and with native support for recursion.
- (3) Examples for $\mathcal{F}(\cdot)$ and $\mathcal{F}^*(\cdot)$.

Due to space limits, omitted material (definitions, correctness proofs for $\mathcal{F}(\cdot)$ and $\mathcal{F}^*(\cdot)$, additional examples) can be found in the appendices. Throughout the paper, we use **red** and **blue** colors to distinguish elements of the first and second decompositions, respectively.

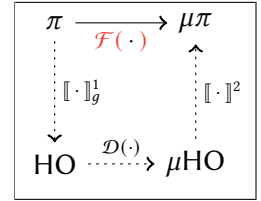


Figure 1: Decomposition by composition.

$n ::= a, b \mid s, \bar{s}$
$u, w ::= n \mid x, y, z$
$V, W ::= \mathbf{u} \mid \boxed{\lambda x. P} \mid \boxed{x, y, z}$
$P, Q ::= u!\langle V \rangle.P \mid u?(x).P$
$\mid \boxed{Vu} \mid P \mid Q \mid (v n)P \mid \mathbf{0} \mid X \mid \mu X.P$

Figure 2: Syntax of HO π . The sub-language HO lacks shaded constructs, while π lacks boxed constructs.

2 PRELIMINARIES

HO and π are actually sub-languages of HO π [9, 10], for which we recall its syntax, semantics, and session type system. We also recall the mutual encodings between HO and π [8, 10]. Finally, we briefly discuss MSTs for HO, and the minimality result in [1].

2.1 HO π (and its Sub-languages HO and π)

Fig. 2 gives the **syntax** of processes P, Q, \dots , values V, W, \dots , and conventions for names. Identifiers a, b, c, \dots denote *shared names*, while \bar{s}, \bar{s}, \dots are used for *session names*. Duality is defined only on session names, thus $\bar{\bar{s}} = s$, but $\bar{\bar{a}} = a$. Names (shared or sessions) are denoted by m, n, \dots , and x, y, z, \dots range over variables. We write \bar{x} to denote a tuple (x_1, \dots, x_n) , and use ϵ to denote the empty tuple.

An abstraction $\lambda x. P$ binds x to its body P . In processes, sequencing is specified via prefixes. The *output* prefix, $u!\langle V \rangle.P$, sends value V on name u , then continues as P . Its dual is the *input* prefix, $u?(x).P$, in which variable x is bound. Parallel composition, $P \mid Q$, reflects the combined behaviour of two processes running simultaneously. Restriction $(v n)P$ binds the endpoints n, \bar{n} in process P . Process $\mathbf{0}$ denotes inaction. Recursive variables and recursive processes are denoted X and $\mu X.P$, respectively. Replication is denoted by the shorthand notation $*P$, which stands for $\mu X.(P \mid X)$.

The sets of free variables, sessions, and names of a process are denoted $\text{fv}(P)$, $\text{fs}(P)$, and $\text{fn}(P)$. A process P is *closed* if $\text{fv}(P) = \emptyset$. We write $u!\langle \rangle.P$ and $u?() .P$ when the communication objects are not relevant. Also, we omit trailing occurrences of $\mathbf{0}$.

As Fig. 2 details, the **sub-languages** π and HO of HO π differ as follows: application Vu is only present in HO; constructs for recursion $\mu X.P$ are present in π but not in HO.

The **operational semantics** of HO π , enclosed in Figure 3, is expressed through a *reduction relation*, denoted \longrightarrow . Reduction is closed under *structural congruence*, \equiv , which identifies equivalent processes from a structural perspective. We write $P\{V/x\}$ to denote the capture-avoiding substitution of variable x with value V in P . We write $\{\}$ to denote the empty substitution. In Figure 3, Rule [App] denotes application, which only concerns names. Rule [Pass] defines a shared or session interaction on channel n 's endpoints. The remaining rules are standard.

2.2 Session Types for HO π

Fig. 4 (top) gives the syntax of types. Value types U include the first-order types C and the higher-order types L . Session types are denoted with S and shared types with $\langle S \rangle$ and $\langle L \rangle$. We write \diamond to

$(\lambda x. P) u \longrightarrow P\{u/x\}$	[App]
$n!\langle V \rangle.P \mid \bar{n}?(x).Q \longrightarrow P \mid Q\{V/x\}$	[Pass]
$P \longrightarrow P' \Rightarrow (v n)P \longrightarrow (v n)P'$	[Res]
$P \longrightarrow P' \Rightarrow P \mid Q \longrightarrow P' \mid Q$	[Par]
$P \equiv Q \longrightarrow Q' \equiv P' \Rightarrow P \longrightarrow P'$	[Cong]
$P_1 \mid P_2 \equiv P_2 \mid P_1 \quad P_1 \mid (P_2 \mid P_3) \equiv (P_1 \mid P_2) \mid P_3$	
$P \mid \mathbf{0} \equiv P \quad P \mid (v n)Q \equiv (v n)(P \mid Q) \quad (n \notin \text{fn}(P))$	
$(v n)\mathbf{0} \equiv \mathbf{0} \quad \mu X.P \equiv P\{\mu X.P/X\} \quad P \equiv Q \text{ if } P \equiv_{\alpha} Q$	

Figure 3: Operational Semantics of HO π .

$U ::= C \mid L$	$C ::= S \mid \langle S \rangle \mid \langle L \rangle$
$L ::= U \rightarrow \diamond \mid U \rightarrow \diamond$	$S ::= !\langle U \rangle; S \mid ?(U); S$
	$\mid \mu t.S \mid \text{end}$
.....	
$U ::= \bar{C} \rightarrow \diamond \mid \bar{C} \rightarrow \diamond$	$C ::= M \mid \langle U \rangle$
$\gamma ::= \text{end} \mid t$	$M ::= \gamma \mid !\langle \bar{U} \rangle; \gamma \mid ?(\bar{U}); \gamma \mid \mu t.M$

Figure 4: STs for HO π (top) and MSTs for HO (bottom).

denote the *process type*. The functional types $U \rightarrow \diamond$ and $U \rightarrow \diamond$ denote *shared* and *linear* higher-order types, respectively. The *output type* $!\langle U \rangle; S$ is assigned to a name that first sends a value of type U and then follows the type described by S . Dually, $?(U); S$ denotes an *input type*. Type end is the termination type. We assume the *recursive type* $\mu t.S$ is guarded, i.e., the type variable t only appears under prefixes. This way, e.g., the type $\mu t.t$ is not allowed. The sets of free/bound variables of a type S are defined as usual; the sole binder is $\mu t.S$. Closed session types do not have free type variables.

Session types for HO exclude C from value types U ; session types for π exclude L from value types U and $\langle L \rangle$ from C .

We write S_1 dual S_2 if S_1 is the *dual* of S_2 . Intuitively, duality converts $!$ into $?$ (and vice-versa). This intuitive definition is enough for our purposes; the formal definition is co-inductive, see [9, 10]. Typing *environments* are defined below:

$$\begin{aligned} \Gamma &::= \emptyset \mid \Gamma, x : U \rightarrow \diamond \mid \Gamma, u : \langle S \rangle \mid \Gamma, u : \langle L \rangle \mid \Gamma, X : \Delta \\ \Lambda &::= \emptyset \mid \Lambda, x : U \rightarrow \diamond \quad \Delta ::= \emptyset \mid \Delta, u : S \end{aligned}$$

Γ , Λ , and Δ satisfy different structural principles. Γ maps variables and shared names to value types, and recursive variables to session environments; it admits weakening, contraction, and exchange principles. While Λ maps variables to linear higher-order types, Δ maps session names to session types. Both Λ and Δ are only subject to exchange. The domains of Γ , Λ and Δ (denoted $\text{dom}(\Gamma)$, $\text{dom}(\Lambda)$, and $\text{dom}(\Delta)$) are assumed pairwise distinct.

Given Γ , we write $\Gamma \setminus x$ to denote the environment obtained from Γ by removing the assignment $x : U \rightarrow \diamond$, for some U . This notation applies similarly to Δ and Λ ; we write $\Delta \setminus \Delta'$ (and $\Lambda \setminus \Lambda'$) with the expected meaning. Notation $\Delta_1 \cdot \Delta_2$ means the disjoint union of Δ_1

and Δ_2 . We define *typing judgements* for values V and processes P :

$$\Gamma; \Lambda; \Delta \vdash V \triangleright U \qquad \Gamma; \Lambda; \Delta \vdash P \triangleright \diamond$$

The judgement on the left says that under environments Γ , Λ , and Δ value V has type U ; the judgement on the right says that under environments Γ , Λ , and Δ process P has the process type \diamond . The typing rules are presented in Appendix B.1.

Type soundness for $\text{HO}\pi$ relies on two auxiliary notions:

Definition 2.1 (Session Environments: Balanced/Reduction). Let Δ be a session environment.

- Δ is *balanced* if whenever $s : S_1, \bar{s} : S_2 \in \Delta$ then S_1 dual S_2 .
- We define reduction \longrightarrow on session environments as:

$$\Delta, s : \langle U \rangle; S_1, \bar{s} : \langle U \rangle; S_2 \longrightarrow \Delta, s : S_1, \bar{s} : S_2$$

THEOREM 2.2 (TYPE SOUNDNESS [9]). *Suppose $\Gamma; \emptyset; \Delta \vdash P \triangleright \diamond$ with Δ balanced. Then $P \longrightarrow P'$ implies $\Gamma; \emptyset; \Delta' \vdash P' \triangleright \diamond$ and $\Delta = \Delta' \text{ or } \Delta \longrightarrow \Delta'$ with Δ' balanced.*

2.3 Mutual Encodings between π and HO

The encodings $\llbracket \cdot \rrbracket_g^1 : \pi \rightarrow \text{HO}$ and $\llbracket \cdot \rrbracket^2 : \text{HO} \rightarrow \pi$ are *typed*: each consists of a translation on processes and a translation on types. This way, $\langle \cdot \rangle^1$ translates types for first-order processes into types for higher-order processes, while $\langle \cdot \rangle^2$ operates in the opposite direction—see Figures 5 and 6, respectively. Remarkably, these translations on processes and types do not alter their sequentiality.

From π to HO. To mimic the sending of name w , the encoding $\llbracket \cdot \rrbracket_g^1$ encloses w within the body of an input-guarded abstraction. The corresponding input process receives this higher-order value, applies it on a restricted session, and sends the encoded continuation through the session's dual.

Several auxiliary notions are used to encode recursive processes; we describe them intuitively (see [10] for full details). The key idea is to encode recursive processes in π using a “duplicator” process in HO, circumventing linearity by replacing free names with variables. The parameter g is a map from process variables to sequences of name variables. Also, $\llbracket \cdot \rrbracket$ maps sequences of session names into sequences of variables, and $\llbracket \cdot \rrbracket_\emptyset$ maps processes with free names to processes without free names (but with free variables instead).

The encoding $\langle \cdot \rangle^1$ depends on the auxiliary function $\llbracket \cdot \rrbracket^1$, defined on value types. Following the encoding on processes, this mapping on values takes a first-order value type and encodes it into a linear higher-order value type, which encloses an input type that expects to receive another higher-order type. Notice how the innermost higher-order value type is either shared or linear, following the nature of the given type.

From HO to π . The encoding $\llbracket \cdot \rrbracket^2$ simulates higher-order communication using first-order constructs, following Sangiorgi [12]. The idea is to use *trigger names*, which point towards copies of input-guarded server processes that should be activated. The encoding of abstraction sending distinguishes two cases: if the abstraction body does not contain any free session names (which are linear), then the server can be replicated. Otherwise, if the value contains session names then its corresponding server name must be used exactly once. The encoding of abstraction receiving proceeds inductively, noticing that the variable is now a placeholder for a first-order

Terms:

$$\begin{aligned} \llbracket u! \langle w \rangle . P \rrbracket_g^1 &\stackrel{\text{def}}{=} u! \langle \lambda z. z?(x).(xw) \rangle . \llbracket P \rrbracket_g^1 \\ \llbracket u?(x:C).Q \rrbracket_g^1 &\stackrel{\text{def}}{=} u?(y).(v s)(y s | \bar{s}! \langle \lambda x. \llbracket Q \rrbracket_g^1 \rangle . 0) \\ \llbracket P | Q \rrbracket_g^1 &\stackrel{\text{def}}{=} \llbracket P \rrbracket_g^1 | \llbracket Q \rrbracket_g^1 \\ \llbracket (v n)P \rrbracket_g^1 &\stackrel{\text{def}}{=} (v n)\llbracket P \rrbracket_g^1 \\ \llbracket 0 \rrbracket_g^1 &\stackrel{\text{def}}{=} 0 \\ \llbracket \mu X.P \rrbracket_g^1 &\stackrel{\text{def}}{=} (v s)(\bar{s}! \langle V \rangle . 0 | s?(z_X) . \llbracket P \rrbracket_{g, \{X \rightarrow \bar{n}\}}^1) \\ &\quad \text{where } (\bar{n} = \text{fn}(P)) \\ &\quad V = \lambda(\llbracket \bar{n} \rrbracket, y). y?(z_X) . \llbracket \llbracket P \rrbracket_{g, \{X \rightarrow \bar{n}\}}^1 \rrbracket_\emptyset \\ \llbracket X \rrbracket_g^1 &\stackrel{\text{def}}{=} (v s)(z_X(\bar{n}, s) | \bar{s}! \langle z_X \rangle . 0) \quad (\bar{n} = g(X)) \end{aligned}$$

Types:

$$\begin{aligned} \llbracket S \rrbracket^1 &\stackrel{\text{def}}{=} (?(\langle S \rangle^1 \rightarrow \diamond); \text{end}) \rightarrow \diamond \\ \llbracket \langle S \rangle \rrbracket^1 &\stackrel{\text{def}}{=} (?(\langle \langle S \rangle^1 \rangle \rightarrow \diamond); \text{end}) \rightarrow \diamond \\ \langle \langle U \rangle; S \rangle^1 &\stackrel{\text{def}}{=} ! \langle \llbracket U \rrbracket^1 \rangle; \langle S \rangle^1 \\ \langle ?(U); S \rangle^1 &\stackrel{\text{def}}{=} ?(\llbracket U \rrbracket^1); \langle S \rangle^1 \\ \langle \langle S \rangle \rangle^1 &\stackrel{\text{def}}{=} \langle \langle S \rangle^1 \rangle \quad \langle \mu t.S \rangle^1 \stackrel{\text{def}}{=} \mu t. \langle S \rangle^1 \\ \langle \text{end} \rangle^1 &\stackrel{\text{def}}{=} \text{end} \quad \langle t \rangle^1 \stackrel{\text{def}}{=} t \end{aligned}$$

Figure 5: Typed encoding of π into HO, selection from [10]. Above, $\text{fn}(P)$ is a lexicographically ordered sequence of free names in P . Maps $\llbracket \cdot \rrbracket$ and $\llbracket \cdot \rrbracket_\emptyset$ are in Def. B.2 and Fig. 17.

Terms:

$$\begin{aligned} \llbracket u! \langle \lambda x. Q \rangle . P \rrbracket^2 &\stackrel{\text{def}}{=} \\ &\begin{cases} (v a)(u! \langle a \rangle . (\llbracket P \rrbracket^2 | * a?(y).y?(x). \llbracket Q \rrbracket^2)) & \text{if } \text{fs}(Q) = \emptyset \\ (v a)(u! \langle a \rangle . (\llbracket P \rrbracket^2 | a?(y).y?(x). \llbracket Q \rrbracket^2)) & \text{otherwise} \end{cases} \\ \llbracket u?(x).P \rrbracket^2 &\stackrel{\text{def}}{=} u?(x). \llbracket P \rrbracket^2 \\ \llbracket x u \rrbracket^2 &\stackrel{\text{def}}{=} (v s)(x! \langle s \rangle . \bar{s}! \langle u \rangle . 0) \\ \llbracket (\lambda x. P) u \rrbracket^2 &\stackrel{\text{def}}{=} (v s)(s?(x). \llbracket P \rrbracket^2 | \bar{s}! \langle u \rangle . 0) \end{aligned}$$

Types:

$$\begin{aligned} \langle \langle S \rightarrow \diamond \rangle; S_1 \rangle^2 &\stackrel{\text{def}}{=} ! \langle \langle ?(\langle S \rangle^2); \text{end} \rangle \rangle; \langle S_1 \rangle^2 \\ \langle ?(S \rightarrow \diamond); S_1 \rangle^2 &\stackrel{\text{def}}{=} ?(\langle ?(\langle S \rangle^2); \text{end} \rangle); \langle S_1 \rangle^2 \end{aligned}$$

Figure 6: Typed encoding of HO into π [10].

name. The encoding of application is also in two cases; both of them depend on the creation of a fresh session, which is used to pass around the applied name.

2.4 Minimal Session Types for HO

The syntax of MSTs for HO is in Fig. 4 (bottom). We write μHO to denote HO with MSTs. The decomposition $\mathcal{D}(\cdot)$ in [1] relies crucially on the ability of communicating tuples of values. Hence, value types are of the form $\bar{C} \rightarrow \diamond$ and $\bar{C} \multimap \diamond$. Similarly, minimal session types for output and input are of the form $!\langle \bar{U} \rangle; \text{end}$ and $?\langle \bar{U} \rangle; \text{end}$; they communicate tuples of values but lack a continuation.

Following Parrow [11], $\mathcal{D}(\cdot)$ is defined in terms of a *breakdown function* $\mathcal{B}_{\bar{x}}^k(\cdot)$, which translates a process into a composition of *trios processes* (or simply *trios*). A trio is a process with exactly three nested prefixes. If P is a sequential process with k nested actions, then $\mathcal{D}(P)$ will contain k trios running in parallel: each trio in $\mathcal{D}(P)$ will enact exactly one prefix from P ; the breakdown function must be carefully designed to ensure that trios trigger each other in such a way that $\mathcal{D}(P)$ preserves the prefix sequencing in P . While trios decompositions elegantly induce processes typable with MSTs, they are not goal in themselves; rather, they offer one possible path to better understand sequentiality in session types.

We use some useful terminology for trios [11]. The *context* of a trio is a tuple of variables \bar{x} , possibly empty, which makes variable bindings explicit. We use a reserved set of *propagator names* (or simply *propagators*), denoted by c_k, c_{k+1}, \dots , to carry contexts and trigger the subsequent trio. Propagators with recursive types are denoted by c_k^r, c_{k+1}^r, \dots . We say that a *leading trio* is the one that receives a context, performs an action, and triggers the next trio; a *control trio* only activates other trios.

The breakdown function works on both processes and values. The breakdown of process P is denoted by $\mathcal{B}_{\bar{x}}^k(P)$, where k is the index for the propagators c_k , and \bar{x} is the context to be received by the previous trio. Similarly, the breakdown of a value V is denoted by $\mathcal{V}_{\bar{x}}^k(V)$. Table 1 gives the breakdown function defined in [1] for the sub-language of HO without recursion—this is the so-called *core fragment*. In the figure, we include side conditions that use the one-line conditional $x = (c) ? s_1 : s_2$ to express that $x = s_1$ if condition c is true, and $x = s_2$ otherwise. Notice that for session types we have either $C = S$ or $C = \langle S \rangle$.

To formally define $\mathcal{D}(\cdot)$ in terms of $\mathcal{B}_{\bar{x}}^k(\cdot)$, we need some notation. Let $\bar{u} = (a, b, s, s', \dots)$ be a finite tuple of names. We shall write $\text{init}(\bar{u})$ to denote the tuple $(a_1, b_1, s_1, s'_1, \dots)$. We say that a process has been *initialized* if all of its names have some index.

Definition 2.3 (Decomposing Processes [1]). Let P be a closed HO process such that $\bar{u} = \text{fn}(P)$. The *decomposition* of P , denoted $\mathcal{D}(P)$, is defined as:

$$\mathcal{D}(P) = (\nu \bar{c})(\bar{c}_k! \langle \cdot \rangle) \cdot \mathcal{B}_{\bar{c}}^k(P\sigma)$$

where: $k > 0$; $\bar{c} = (c_k, \dots, c_{k+|P|-1})$; $\sigma = \{\text{init}(\bar{u})/\bar{u}\}$; and the breakdown function $\mathcal{B}_{\bar{x}}^k(\cdot)$, is as defined in Table 1.

Definition 2.4 (Decomposing Session Types). The *decomposition function* on the types of Fig. 4, denoted $\mathcal{G}(\cdot)$, is defined in Fig. 7.

As already mentioned, the *minimality result* in [1] is that if P is a well-typed HO process then $\mathcal{D}(P)$ is a well-typed μHO process. It attests that the sequentiality in the session types for P is appropriately accommodated by the decomposition $\mathcal{D}(P)$. Its proof relies on an auxiliary result establishing the typability of $\mathcal{B}_{\bar{x}}^k(P)$.

$$\begin{aligned} \mathcal{G}(!\langle U \rangle; S) &= \begin{cases} !\langle \mathcal{G}(U) \rangle; \text{end} & \text{if } S = \text{end} \\ !\langle \mathcal{G}(U) \rangle; \text{end}, \mathcal{G}(S) & \text{otherwise} \end{cases} \\ \mathcal{G}(? \langle U \rangle; S) &= \begin{cases} ? \langle \mathcal{G}(U) \rangle; \text{end} & \text{if } S = \text{end} \\ ? \langle \mathcal{G}(U) \rangle; \text{end}, \mathcal{G}(S) & \text{otherwise} \end{cases} \\ \mathcal{G}(C \multimap \diamond) &= \mathcal{G}(C) \multimap \diamond & \mathcal{G}(\text{end}) &= \text{end} \\ \mathcal{G}(C \rightarrow \diamond) &= \mathcal{G}(C) \rightarrow \diamond & \mathcal{G}(\langle U \rangle) &= \langle \mathcal{G}(U) \rangle \\ \mathcal{G}(S_1, \dots, S_n) &= \mathcal{G}(S_1), \dots, \mathcal{G}(S_n) \end{aligned}$$

Figure 7: Decomposition of types (cf. Def. 2.4)

THEOREM 2.5 (MINIMALITY RESULT [1]). Let P be a closed HO process with $\bar{u} = \text{fn}(P)$ and $\sigma = \{\text{init}(\bar{u})/\bar{u}\}$. If $\Gamma; \emptyset; \Delta \vdash P \triangleright \diamond$ then

$$\mathcal{G}(\Gamma\sigma); \emptyset; \mathcal{G}(\Delta\sigma) \vdash \mathcal{D}(P) \triangleright \diamond$$

Having summarized the results on which our developments stand, we now move on to establish the minimality result but for π .

3 DECOMPOSE BY COMPOSITION

We define a decomposition function $\mathcal{F}(\cdot) : \pi \rightarrow \mu\pi$, given in terms of a breakdown function denoted $\mathcal{A}_{\bar{x}}^k(\cdot)_g$ (cf. Tab. 2). Following Figure 1, this breakdown function will result from the composition of $\llbracket \cdot \rrbracket_g^1$, $\mathcal{B}_{\bar{x}}^k(\cdot)$, and $\llbracket \cdot \rrbracket^2$, i.e., $\mathcal{A}_{\bar{x}}^k(\cdot)_g = \llbracket \mathcal{B}_{\bar{x}}^k(\llbracket \cdot \rrbracket_g^1) \rrbracket^2$. Using $\mathcal{F}(\cdot)$, we obtain a minimality result for π , given by Theorem 2.

3.1 Key Idea

Conceptually, $\mathcal{F}(\cdot)$ can be obtained in two steps: first, the composition $\mathcal{B}_{\bar{x}}^k(\llbracket \cdot \rrbracket_g^1)$, which returns a process in μHO ; second, a step that transforms that μHO process into a $\mu\pi$ process using $\llbracket \cdot \rrbracket^2$. We illustrate these two steps for output and input processes.

Output Let us write $\mathcal{A}_{\bar{x}}^k(\cdot)_g$ to denote the (partial) composition involved in the first step. Given $P = u_i! \langle w_j \rangle . Q$, we first obtain:

$$\mathcal{A}_{\bar{x}}^k(u_i! \langle w_j \rangle . Q)_g = c_k?(\bar{x}). u_i! \langle W \rangle . \bar{c}_{k+3}! \langle \bar{x} \rangle \mid \mathcal{A}_{\bar{x}}^{k+3}(Q\sigma)_g$$

where $\sigma = (u_i : S) ? \{u_{i+1}/u_i\} : \{ \}$ and

$$W = \lambda z_1. (\bar{c}_{k+1}! \langle \cdot \rangle \mid c_{k+1}?(\cdot). z_1?(x). \bar{c}_{k+2}! \langle x \rangle \mid c_{k+2}?(x). (x \bar{w}))$$

We have that $\mathcal{A}_{\bar{x}}^k(u_i! \langle w_j \rangle . Q)_g$ is a process in μHO . The second step uses $\llbracket \cdot \rrbracket^2$ to convert it into the following $\mu\pi$ process:

$$\begin{aligned} c_k?(\bar{x}). (\nu a)(u_i! \langle a \rangle . (\bar{c}_{k+3}! \langle \bar{x} \rangle \mid \mathcal{A}_{\bar{x}}^{k+3}(Q\sigma)_g \mid \\ a?(y). y?(z_1). \bar{c}_{k+1}! \langle z_1 \rangle \mid c_{k+1}?(z_1). z_1?(x). \bar{c}_{k+2}! \langle x \rangle \mid \\ c_{k+2}?(x). (\nu s)(x! \langle s \rangle . \bar{s}! \langle \bar{w} \rangle))) \end{aligned}$$

The subprocess mimicking the output action on u_i is guarded by an input on c_k . Then, the output of w on u_i action is delegated to a different channel through several redirections: first a private name a is sent, then along a name for z_1 is received and so on; until, finally, the breakdown of w is sent on name s_1 . These names are propagated through local trios. We can see that upon action on u_i unmodified context \bar{x} is sent to breakdown of continuation Q .

We say names typed with tail-recursive type are *recursive names*. Another form of output is when both u_i and w_j are recursive. This case is similar to the one just discussed, and omitted from Tab. 2.

P	$\mathcal{B}_{\tilde{x}}^k(P)$	
$u_i!(V).Q$	$c_k?(\tilde{x}).u_i!(\tilde{V}).\overline{c_{k+1}}!\langle\tilde{z}\rangle \mid \mathcal{B}_{\tilde{z}}^{k+1}(Q\sigma)$	$\tilde{V} = \mathcal{V}_{\tilde{y}}^{k+1}(V\sigma)$ $\tilde{y} = \text{fv}(V)$ $\tilde{z} = \text{fv}(Q)$ $l = V $ $\sigma = \text{next}(u_i)$
$u_i?(y).Q$	$c_k?(\tilde{x}).u_i?(y).\overline{c_{k+1}}!\langle\tilde{x}'\rangle \mid \mathcal{B}_{\tilde{x}'}^{k+1}(Q\sigma)$	$\tilde{x}' = \text{fv}(Q)$ $\sigma = \text{next}(u_i)$
$V u_i$	$c_k?(\tilde{x}).\mathcal{V}_{\tilde{x}}^{k+1}(V) \tilde{m}$	$u_i : C$ $\tilde{m} = (u_i, \dots, u_{i+ \mathcal{G}(C) -1})$ $\tilde{x} = \text{fv}(V)$
$(\nu s : C)P'$	$(\nu \tilde{s} : \mathcal{G}(C)) \mathcal{B}_{\tilde{x}}^k(P'\sigma)$	$\tilde{s} = (s_1, \dots, s_{ \mathcal{G}(C) })$ $\sigma = (C = S) ? \{s_1 s_1 / s \tilde{s}\} : \{s_1 / s\}$ $\tilde{x} = \text{fv}(P')$
$Q \mid R$	$c_k?(\tilde{x}).\overline{c_{k+1}}!\langle\tilde{y}\rangle.\overline{c_{k+1}}!\langle\tilde{z}\rangle \mid \mathcal{B}_{\tilde{y}}^{k+1}(Q) \mid \mathcal{B}_{\tilde{z}}^{k+1}(R)$	$\tilde{y} = \text{fv}(Q)$ $\tilde{z} = \text{fv}(R)$ $l = Q $
$\mathbf{0}$	$c_k?().\mathbf{0}$	
V	$\mathcal{V}_{\tilde{x}}^k(V)$	
y	y	
$\lambda u : C \rightsquigarrow . P$	$\lambda \tilde{y}. (\nu \tilde{c}). (\overline{c_{k+1}}!\langle\tilde{x}\rangle \mid \mathcal{B}_{\tilde{x}}^k(P\{y_1/y\}))$	$\tilde{x} = \text{fv}(V)$ $\tilde{y} = (y_1, \dots, y_{ \mathcal{G}(C) })$ $\tilde{c} = (\rightsquigarrow = \rightarrow) ? (c_k, \dots, c_{k+ P -1}) : \epsilon$

Table 1: The breakdown function for HO processes and values (core fragment from [1]).

$C ::= M \mid \langle M \rangle$
$\gamma ::= \text{end} \mid t$
$M ::= \gamma \mid !\langle \tilde{C} \rangle ; \gamma \mid ?\langle \tilde{C} \rangle ; \gamma \mid \mu t.M$

Figure 8: Minimal Session Types for π (cf. Definition 3.1)

Input The breakdown of $u_i?(w).Q$ as follows:

$$\begin{aligned}
 & c_k?(\tilde{x}).u_i?(y).\overline{c_{k+1}}!\langle\tilde{x}, y\rangle \mid \\
 & (\nu s_1)(c_{k+1}?(\tilde{x}, y).\overline{c_{k+2}}!\langle y, \overline{c_{k+3}}!\langle\tilde{x}\rangle \mid \\
 & c_{k+2}?(y).(\nu s)(y!\langle s, \tilde{s}\rangle!\langle s_1 \rangle) \mid \\
 & c_{k+3}?(\tilde{x}).(\nu a)(\tilde{s}_1!\langle a, \overline{c_{k+4}}!\langle \rangle \mid c_{k+4}?(?).\mathbf{0} \mid \\
 & a?(y').y'?(w).\overline{c_{k+4}}!\langle\tilde{x}\rangle \mid \mathcal{A}_{\tilde{x}}^{k+4}(Q\{w_1/w\}\sigma)_g))
 \end{aligned}$$

The activation on c_k enables the input on u_i . After several redirections, the actual input of variables \tilde{w} is on a name received for y' , which binds them in the decomposition of Q . Hence, context \tilde{x} does not get extended for an inductive call: it only gets extended locally (propagated by c_{k+1}). Indeed, in the core fragment, the context is always empty and propagators only enable subsequent actions. The context does play a role in breaking down recursion: variables z_X (generated to encode recursion) get propagated as context.

3.2 Formal Definition

Definition 3.1 (Minimal Session Types, MSTs). Minimal session types for π are defined in Figure 8.

The breakdown function $\mathcal{A}_{\tilde{x}}^k(\cdot)_g$ for all constructs of π is given in Table 2, using the following definitions.

Definition 3.2 (Degree of a Process). The degree of a π process P , denoted $\lfloor P \rfloor$, is defined as:

$$\begin{aligned}
 \lfloor u_i!(w_j).Q \rfloor &= \lfloor Q \rfloor + 3 & \lfloor (\nu s : S)Q \rfloor &= \lfloor Q \rfloor & \lfloor \mathbf{0} \rfloor &= 1 \\
 \lfloor u_i?(x : C).Q \rfloor &= \lfloor Q \rfloor + 5 & \lfloor Q \mid R \rfloor &= \lfloor Q \rfloor + \lfloor R \rfloor + 1 \\
 \lfloor X \rfloor &= 4 & \lfloor \mu X.Q \rfloor &= \lfloor Q \rfloor + 4
 \end{aligned}$$

Definition 3.3 (Predicates on Types and Names). Let C be a session type. We write $\text{tr}(C)$ to indicate that C is a tail-recursive session type. Also, given $u : C$, we write $\text{lin}(u)$ if $C = S$ and $\neg \text{tr}(S)$.

Definition 3.4 (Subsequent index substitution). Let n_i be an indexed name. We define $\text{next}(n_i) = (\text{lin}(n_i)) ? \{n_{i+1}/n_i\} : \{\}$.

We define how to obtain MSTs for π from standard session types:

Definition 3.5 (Decomposing First-Order Types). The decomposition function $\mathcal{H}(\cdot)$ on finite types, obtained by combining the mappings $\langle \cdot \rangle^1$, $\mathcal{G}(\cdot)$, and $\langle \cdot \rangle^2$, is defined in Fig. 9 (top, where omitted cases are defined homomorphically). It is extended to account for recursive session types in Fig. 9 (center).

The auxiliary function $\mathcal{R}'^*(\cdot)$, given in Fig. 9 (bottom), is used in Tab. 2 to decompose *guarded* tail-recursive types: it skips session prefixes until a type of form $\mu t.S$ is encountered; when that occurs, the recursive type is decomposed using $\mathcal{R}'(\cdot)$.

We are finally ready to define the decomposition function $\mathcal{F}(\cdot)$, the analog of Definition 2.3 but for processes in π :

Definition 3.6 (Process Decomposition). Let P be a closed π process with $\tilde{u} = \text{fn}(P)$ and $\tilde{v} = \text{rn}(P)$. Given the breakdown function $\mathcal{A}_{\tilde{x}}^k(\cdot)_g$ in Table 2, the decomposition $\mathcal{F}(P)$ is defined as:

$$\mathcal{F}(P) = (\nu \tilde{c}). (\nu \tilde{c}_r) \left(\prod_{r \in \tilde{v}} P^r \mid \overline{c_k}!\langle \rangle.\mathbf{0} \mid \mathcal{A}_{\tilde{e}}^k(P\sigma)_g \right)$$

P	$\mathcal{A}_{\bar{x}}^k(P)_g$	
$u_i! \langle w_j \rangle . Q$	$c_k?(\bar{x}).(v a)(u_i! \langle a \rangle . (\bar{c}_{k+3}! \langle \bar{x} \rangle \mathcal{A}_{\bar{x}}^{k+3}(Q\sigma)_g $ $a?(y).y?(z_1). \bar{c}_{k+1}! \langle z_1 \rangle c_{k+1}?(z_1).z_1?(x). \bar{c}_{k+2}! \langle x \rangle $ $c_{k+2}?(x).(v s)(x! \langle s \rangle . \bar{s}! \langle \bar{w} \rangle)))$	$w_j : C$ $\bar{w} = (w_j, \dots, w_{j+ \mathcal{H}(C) -1})$ $\sigma = \text{next}(u_i)$
$u_i?(w).Q$	$c_k?(\bar{x}).u_i?(y). \bar{c}_{k+1}! \langle \bar{x}, y \rangle $ $(v s_1)(c_{k+1}?(\bar{x}, y). \bar{c}_{k+2}! \langle y \rangle . \bar{c}_{k+3}! \langle \bar{x} \rangle $ $c_{k+2}?(y).(v s)(y! \langle s \rangle . \bar{s}! \langle s_1 \rangle) $ $c_{k+3}?(\bar{x}).(v a)(\bar{s}_1! \langle a \rangle . (\bar{c}_{k+l+4}! \langle \rangle c_{k+l+4}?(.) . \mathbf{0} \bar{Q}_{\bar{x}})))$ where: $\bar{Q}_{\bar{x}} = a?(y').y'?(\bar{w}).(\bar{c}_{k+4}! \langle \bar{x} \rangle \mathcal{A}_{\bar{x}}^{k+4}(Q\{w_l/w\}\sigma)_g)$	$w : C$ $\bar{w} = (w_1, \dots, w_{ \mathcal{H}(C) })$ $l = \lfloor Q \rfloor$ $\sigma = \text{next}(u_i)$
$r_i! \langle w_j \rangle . P$	$c_k?(\bar{x}).(v a_1)c^r! \langle a_1 \rangle . (\mathcal{A}_{\bar{x}}^{k+3}(P)_g a_1?(y_1).y_1?(\bar{z}).W)$ where: $W = (v a_2)(z_f(s)! \langle a_2 \rangle . (\bar{c}_{k+3}! \langle \bar{x} \rangle . c^r?(b).(v s)(b! \langle s \rangle . \bar{s}! \langle \bar{z} \rangle) $ $a_2?(y_2).y_2?(z'_1). (\bar{c}_{k+1}! \langle \rangle $ $c_{k+1}?(.) . z'_1?(x). \bar{c}_{k+2}! \langle x \rangle c_{k+2}?(x).(v s')(x! \langle s' \rangle . \bar{s}'! \langle \bar{w} \rangle)))$	$r : S \wedge \text{tr}(S)$ $\bar{z} = (z_1, \dots, z_{ \mathcal{R}^*(S) })$ $\bar{c} = (c_{k+1}, c_{k+2})$ $w : C \wedge \bar{w} = (w_j, \dots, w_{j+ \mathcal{H}(C) -1})$
$r_i?(w).P$	$c_k?(\bar{x}).(v a_1)(c^r! \langle a_1 \rangle . ((v s_1)(c_{k+1}?(y). \bar{c}_{k+2}! \langle y \rangle . \bar{c}_{k+3}! \langle \rangle $ $c_{k+2}?(y).(v s)(y! \langle s \rangle . \bar{s}! \langle s_1 \rangle) $ $c_{k+3}?(.) . (v a_2)(s_1! \langle a_2 \rangle . (\bar{c}_{k+l+4}! \langle \rangle c_{k+l+4}?(.) . \mathbf{0} $ $a_2?(y_2).y_2?(\bar{w}).(\bar{c}_{k+4}! \langle \bar{x} \rangle \mathcal{A}_{\bar{x}}^{k+4}(P\{w_l/w\}_g))) $ $a_1?(y_1).y_1?(\bar{z}).z_f(s)?(y).$ $\bar{c}_{k+1}! \langle y \rangle . c^r?(b).(v s')(b! \langle s' \rangle . \bar{s}'! \langle \bar{z} \rangle)))$	$r : S \wedge \text{tr}(S)$ $\bar{z} = (z_1, \dots, z_{ \mathcal{R}^*(S) })$ $l = \lfloor P \rfloor$ $w : C \wedge \bar{w} = (w_1, \dots, w_{ \mathcal{H}(C) })$
$(v s : C)P'$	$(v \bar{s} : \mathcal{H}(C)) \mathcal{A}_{\bar{x}}^k(P'\sigma)_g$	$\bar{s} = (s_1, \dots, s_{ \mathcal{H}(C) }) \quad \sigma = \{s_1 \bar{s}_1 / s \bar{s}\}$
$(v r : \mu t.S)P'$	$(v \bar{r} : \mathcal{R}'(S)) c^r?(b).(v s')(b! \langle s' \rangle . \bar{s}'! \langle \bar{r} \rangle) c^{\bar{r}}?(b).(v s')(b! \langle s' \rangle . \bar{s}'! \langle \bar{r} \rangle) \mathcal{A}_{\bar{x}}^k(P'\sigma)_g$	$\text{tr}(\mu t.S) \quad \sigma = \{r_1 \bar{r}_1 / r \bar{r}\}$ $\bar{r} = (r_1, \dots, r_{ \mathcal{R}'(S) })$ $\bar{\bar{r}} = (\bar{r}_1, \dots, \bar{r}_{ \mathcal{R}'(S) })$
$Q_1 Q_2$	$c_k?(\bar{x}). \bar{c}_{k+1}! \langle \bar{y} \rangle . \bar{c}_{k+l+1}! \langle \bar{z} \rangle \mathcal{A}_{\bar{y}}^{k+1}(Q_1)_g \mathcal{A}_{\bar{z}}^{k+l+1}(Q_2)_g$	$\bar{y} = \text{fv}(Q_1) \quad \bar{z} = \text{fv}(Q_2) \quad l = \lfloor Q \rfloor$
$\mathbf{0}$	$c_k?(.) . \mathbf{0}$	
$\mu X.P$	$(v s_1)(c_k?(\bar{x}). \bar{c}_{k+1}! \langle \bar{x} \rangle . \bar{c}_{k+3}! \langle \bar{x} \rangle $ $c_{k+1}?(\bar{x}).(v a_1)(s_1! \langle a_1 \rangle . (\bar{c}_{k+2}! \langle \rangle c_{k+2}?(.) . \mathbf{0} $ $c_{k+3}?(\bar{x}).s_1?(z_x). \bar{c}_{k+4}! \langle \bar{x}, z_x \rangle $ $\mathcal{A}_{\bar{x}, z_x}^{k+4}(P)_g, \{X \rightarrow \bar{n}\} $ $* a_1?(y'_1).y'_1?((\ \bar{n}^1\ , \dots, \ \bar{n}^m\ , y_1). \hat{P})))$ where: $\hat{P} = (v \bar{c})(\prod_{0 < i \leq m} c^{n_i}?(b).(v s')(b! \langle s' \rangle . \bar{s}'! \langle \ \bar{n}^i\ \rangle) \bar{c}_{k+2}! \langle \bar{x} \rangle$ $c_{k+2}?(\bar{x}).y_1?(z_x). \bar{c}_{k+3}! \langle \bar{x}, z_x \rangle \llbracket \mathcal{A}_{\bar{x}, z_x}^{k+3}(P)_g, \{X \rightarrow \bar{n}\} \rrbracket_{\bar{c}, \bar{c}_r})$	$\bar{n} = \text{fn}(P)$ $m = \bar{n} $ $\ \bar{n}\ = (\ n_1\ , \dots, \ n_m\)$ $i \in \{1, \dots, m\}$. $\ n_i\ : S_i$ $\ \bar{n}^i\ = (\ n_1^i\ , \dots, \ n_{ \mathcal{H}(S_i) }^i\)$ $\bar{c} = (c_{k+2}, \dots, c_{k+l} \ P\ _{g, \{X \rightarrow \bar{n}\}}^{l+1})$ $\bar{c}_r = \bigcup_{v \in \bar{n}} c^v$
X	$(v s_1)(c_k?(z_x). \bar{c}_{k+1}! \langle z_x \rangle . \bar{c}_{k+2}! \langle z_x \rangle c_{k+2}?(z_x). \bar{s}_1! \langle z_x \rangle . \bar{c}_{k+3}! \langle \rangle c_{k+3}?(.) . \mathbf{0}$ $c_{k+1}?(z_x).(v a_1)(c^{n_1}! \langle a_1 \rangle . (a_1?(y_1).y_1?(\bar{z}_1) . \dots (v a_j)Q)))$ where: $Q = (c^{n_j}! \langle a_j \rangle . (a_j?(y_j).y_j?(\bar{z}_j).(v s')(z_x! \langle s' \rangle . \bar{s}'! \langle \bar{z}_1, \dots, \bar{z}_j, s_1 \rangle)))$	$\bar{n} = g(X)$ $ \bar{n} = j$ $i \in \{1, \dots, j\}$ $n_i : S \wedge \text{tr}(S_i)$ $\bar{z}_i = (z_1^i, \dots, z_{ \mathcal{R}^*(S_i) }^i)$

Table 2: Decompose by composition: Breakdown function $\mathcal{A}_{\bar{x}}^k(\cdot)_g$ for π processes (cf. Definition 3.6).

where: $k > 0$, $\bar{c} = (c_k, \dots, c_{k+|\mathcal{H}(C)|-1})$; $\bar{c}_r = \bigcup_{r \in \bar{c}} c^r$; $\sigma = \{\text{init}(\bar{u})/\bar{u}\}$;
 $P^r = c^r?(b).(v s)(b! \langle s \rangle . \bar{s}! \langle \bar{r} \rangle)$ with $r : S$ and $\bar{r} = r_1, \dots, r_{|\mathcal{G}(S)|}$.

3.3 Examples

Example 3.7 (A Process with Delegation). Consider a process P that implements channels \bar{w} (with type $T = ?(\text{Int}); !(\text{Bool}); \text{end}$) and

u (with type $S = !(\bar{T}); \text{end}$):

$$P = (v u : S) \underbrace{(u! \langle w \rangle . \bar{w}?(t). \bar{w}! \langle \text{odd}(t) \rangle . \mathbf{0} | \bar{u}?(x).x! \langle 5 \rangle . x?(b). \mathbf{0})}_A \underbrace{)}_B$$

By Def. 3.2, $\lfloor P \rfloor = 25$. Then, the decomposition of P into a collection of first-order processes typed with minimal session types is:

$$\mathcal{F}(P) = (v c_1, \dots, c_{25})(\bar{c}_1! \langle \rangle . \mathbf{0} | (v u_1) \mathcal{A}_{\bar{c}}^1((A | B)\sigma')),$$

$$\begin{aligned}
\mathcal{H}(\langle S \rangle) &= \langle \mathcal{H}(S) \rangle \\
\mathcal{H}(!\langle S \rangle; S') &= \begin{cases} M & \text{if } S' = \text{end} \\ M, \mathcal{H}(S') & \text{otherwise} \end{cases} \\
&\text{where } M = !\langle \langle ?(\langle ?(\mathcal{H}(S)); \text{end}); \text{end}); \text{end} \rangle \rangle; \text{end} \\
\mathcal{H}(\langle S \rangle; S') &= \begin{cases} M & \text{if } S' = \text{end} \\ M, \mathcal{H}(S') & \text{otherwise} \end{cases} \\
&\text{where } M = ?(\langle ?(\langle ?(\mathcal{H}(S)); \text{end}); \text{end}); \text{end} \rangle \rangle; \text{end} \\
\mathcal{H}(\text{end}) &= \text{end} \\
\mathcal{H}(S_1, \dots, S_n) &= \mathcal{H}(S_1), \dots, \mathcal{H}(S_n) \\
\hline
\mathcal{H}(\mu t.S) &= \begin{cases} \mathcal{R}'(S) & \text{if } \mu t.S \text{ is tail-recursive} \\ \mu t.\mathcal{H}(S) & \text{otherwise} \end{cases} \\
\mathcal{R}'(!\langle S \rangle; S') &= \mu t.! \langle \langle ?(\langle ?(\mathcal{H}(S)); \text{end}); \text{end}); \text{end} \rangle \rangle; t, \mathcal{R}'(S') \\
\mathcal{R}'(\langle S \rangle; S') &= \mu t. \langle \langle ?(\langle ?(\mathcal{H}(S)); \text{end}); \text{end}); \text{end} \rangle \rangle; t, \mathcal{R}'(S') \\
\mathcal{H}(t) &= t \quad \mathcal{R}'(t) = \epsilon \\
\hline
\mathcal{R}^*(\langle S \rangle; S') &= \mathcal{R}^*(S') \quad \mathcal{R}^*(!\langle S \rangle; S') = \mathcal{R}^*(S') \\
\mathcal{R}^*(\mu t.S) &= \mathcal{R}^*(S)
\end{aligned}$$

Figure 9: Decomposition of types $\mathcal{H}(\cdot)$ (cf. Def. 3.5)

$$\begin{aligned}
\mathcal{A}_\epsilon^2(A) &= c_2?(().(v a_1)(u_1!\langle a_1 \rangle.(\bar{c}_5!\langle \rangle | \mathcal{A}_\epsilon^5(A') | \\
&\quad a_1?(y_1).y_1?(z_1).\bar{c}_3!\langle z_1 \rangle | c_3?(z_1).z_1?(x).\bar{c}_4!\langle x \rangle | \\
&\quad c_4?(x).(v s)(x!\langle s \rangle.\bar{s}!\langle w_1, w_2 \rangle))) \\
\mathcal{A}_\epsilon^5(A') &= c_5?(().\bar{w}_1?(y_2).\bar{c}_6!\langle y_2 \rangle | (v s_1)(c_6?(y_2).\bar{c}_7!\langle y_2 \rangle.\bar{c}_8!\langle \rangle | \\
&\quad c_7?(y_2).(v s')(y_2!\langle s' \rangle.\bar{s}'!\langle s_1 \rangle.\mathbf{0}) | \\
&\quad c_8?(().(v a_2)(\bar{s}_1!\langle a_2 \rangle.(\bar{c}_{10}\langle \rangle | c_{10}?(().\mathbf{0} | \\
&\quad a_2?(y_3).y_3?(t_1).(\bar{c}_9!\langle \rangle | \mathcal{A}_\epsilon^9(A'')))) \\
\mathcal{A}_\epsilon^9(A'') &= c_9?(().(v a)(\bar{w}_2!\langle a \rangle.(\bar{c}_{12}\langle \rangle | c_{12}?(().\mathbf{0} | \\
&\quad a?(y).y?(z_1).\bar{c}_{11}\langle z_1 \rangle | c_{10}?(z_1).z_1?(x).\bar{c}_{11}\langle x \rangle | \\
&\quad c_{11}?(x).(v s)(x!\langle s \rangle.\bar{s}!\langle \text{odd}(t) \rangle)))
\end{aligned}$$

Figure 10: Breakdown for process A in Exam. 3.7.

where $\sigma = \text{init}(\text{fn}(P))$ and $\sigma' = \sigma \cdot \{u_1\bar{u}_1/u\bar{u}\}$. We omit parameter g as it is empty. We have:

$$\mathcal{A}_\epsilon^1((A | B)\sigma') = c_1?(().\bar{c}_2!\langle \rangle.\bar{c}_{13}\langle \rangle | \mathcal{A}_\epsilon^2(A\sigma') | \mathcal{A}_\epsilon^{13}(B\sigma')$$

We use the following abbreviations for subprocesses of A and B : $A' = \bar{w}_1?(t).A''$, $A'' = \bar{w}_1!\langle \text{odd}(t) \rangle.\mathbf{0}$, and $B' = x_1!\langle 5 \rangle.x_1?(b).\mathbf{0}$. The breakdown of A is in Fig. 10; the breakdown of B follows:

$$\begin{aligned}
\mathcal{A}_\epsilon^{13}(B) &= c_{13}?(().\bar{u}_1?(y_4).\bar{c}_{14}\langle y_4 \rangle | (v s_1)(c_{14}?(y).\bar{c}_{15}\langle y \rangle.\bar{c}_{16}\langle \rangle | \\
&\quad c_{15}?(y_4).(v s'')(y_4!\langle s'' \rangle.\bar{s}''!\langle s_1 \rangle.\mathbf{0}) | c_{16}?(().(v a_3)(s_1!\langle a_3 \rangle. \\
&\quad (\bar{c}_{21}\langle \rangle | c_{21}?(().\mathbf{0} | a_3?(y_5).y_5?(x_1, x_2).(\bar{c}_{17}\langle \rangle | \mathcal{A}_\epsilon^{17}(B')))))
\end{aligned}$$

The breakdown of B' is similar and given in App. A.1. Type S is broken down into MSTs M_1 and M_2 , as follows:

$$\begin{aligned}
M_1 &= ?(\langle ?(\langle ?(\langle \text{Int} \rangle; \text{end}); \text{end}); \text{end} \rangle \rangle; \text{end} \\
M_2 &= !\langle \langle ?(\langle ?(\langle \text{Bool} \rangle; \text{end}); \text{end}); \text{end} \rangle \rangle; \text{end}
\end{aligned}$$

Names \bar{w}_1 and \bar{w}_2 are typed with M_1 and M_2 , respectively. Then, name u_1 is typed with M , given by:

$$M = !\langle \langle ?(\langle ?(\langle \bar{M}_1, \bar{M}_2 \rangle; \text{end}); \text{end}); \text{end} \rangle \rangle; \text{end}$$

Consider the reductions of $\mathcal{F}(P)$ that mimic the exchange of w along u in P . We first have three synchronizations on c_1, c_2, c_{13} :

$$\begin{aligned}
\mathcal{F}(P) &\longrightarrow^3 (v \bar{c})(v a_1)(u_1!\langle a_1 \rangle.(\bar{c}_5!\langle \rangle | \mathcal{A}_\epsilon^5(A') | \\
&\quad a_1?(y_1).y_1?(z_1).\bar{c}_3!\langle z_1 \rangle | c_3?(z_1).z_1?(x).\bar{c}_4!\langle x \rangle | \\
&\quad c_4?(x).(v s)(x!\langle s \rangle.\bar{s}!\langle w_1, w_2 \rangle))) | \bar{u}_1?(y_4).c_{14}\langle y_4 \rangle | \\
&\quad (v s_1)(c_{14}?(y).\bar{c}_{15}\langle y \rangle.\bar{c}_{16}\langle \rangle | \\
&\quad c_{15}?(y_4).(v s'')(y_4!\langle s'' \rangle.\bar{s}''!\langle s_1 \rangle.\mathbf{0}) | \\
&\quad c_{16}?(().(v a_3)(s_1!\langle a_3 \rangle.(\bar{c}_{21}\langle \rangle | \\
&\quad c_{21}?(().\mathbf{0} | a_3?(y_5).y_5?(x_1, x_2).(\bar{c}_{17}\langle \rangle | \mathcal{A}_\epsilon^{17}(B')))))
\end{aligned}$$

where $\bar{c} = (c_3, \dots, c_{12}, c_{14}, \dots, c_{25})$. Then, a synchronization on u_1 sends name a_1 (highlighted above). Name a_1 is further propagated along c_{14} and c_{15} . Another synchronization occurs on c_{16} .

$$\begin{aligned}
\mathcal{F}(P) &\longrightarrow^7 (v \bar{c}_*)(v a_1)(\bar{c}_5!\langle \rangle | \mathcal{A}_\epsilon^5(A') | a_1!(y_1).y_1?(z_1).\bar{c}_3!\langle z_1 \rangle | \\
&\quad c_3?(z_1).z_1?(x).\bar{c}_4!\langle x \rangle | c_4?(x).(v s)(x!\langle s \rangle.\bar{s}!\langle w_1, w_2 \rangle) | \\
&\quad (v s_1)((v s'')(a_1!\langle s'' \rangle.\bar{s}''!\langle s_1 \rangle.\mathbf{0}) | (v a_3)(s_1!\langle a_3 \rangle.(\bar{c}_{21}\langle \rangle | \\
&\quad c_{21}?(().\mathbf{0} | a_3?(y_5).y_5?(x_1, x_2).(\bar{c}_{17}\langle \rangle | \mathcal{A}_\epsilon^{17}(B'))))) \\
&\text{where } \bar{c}_* = (c_3, \dots, c_{12}, c_{17}, \dots, c_{25})
\end{aligned}$$

The next reduction communicates session name s'' along a_1 :

$$\begin{aligned}
\mathcal{F}(P) &\longrightarrow^8 (v \bar{c}_*)(v s'')(v s'')(\bar{c}_5!\langle \rangle | \mathcal{A}_\epsilon^5(A') | \\
&\quad s''?(z_1).\bar{c}_3!\langle z_1 \rangle | c_3?(z_1).z_1?(x).\bar{c}_4!\langle x \rangle | \\
&\quad c_4?(x).(v s)(x!\langle s \rangle.\bar{s}!\langle w_1, w_2 \rangle) | \\
&\quad (v s_1)(\bar{s}''!\langle s_1 \rangle.\mathbf{0} | (v a_3)(s_1!\langle a_3 \rangle.(\bar{c}_{21}\langle \rangle | c_{21}?(().\mathbf{0} | \\
&\quad a_3?(y_5).y_5?(x_1, x_2).(\bar{c}_{17}\langle \rangle | \mathcal{A}_\epsilon^{17}(B')))))
\end{aligned}$$

After the synchronization on channel s'' , name z_1 is further sent to the next parallel process through the propagator c_3 :

$$\begin{aligned}
\mathcal{F}(P) &\longrightarrow^{10} (v \bar{c}_{**})(v s_1)(\bar{c}_5!\langle \rangle | \mathcal{A}_\epsilon^5(A') | \\
&\quad s_1?(x).\bar{c}_4!\langle x \rangle | c_4?(x).(v s)(x!\langle s \rangle.\bar{s}!\langle w_1, w_2 \rangle) | \\
&\quad (v a_3)(s_1!\langle a_3 \rangle.(\bar{c}_{21}\langle \rangle | c_{21}?(().\mathbf{0} | \\
&\quad a_3?(y_5).y_5?(x_1, x_2).(\bar{c}_{17}\langle \rangle | \mathcal{A}_\epsilon^{17}(B'))))) \\
&\text{where } \bar{c}_{**} = (c_4, \dots, c_{12}, c_{17}, \dots, c_{25})
\end{aligned}$$

Communication on s_1 leads to variable x being substituted by name a_3 , which is then passed on c_4 to the next process. In addition,

inaction is simulated by a synchronization on c_{21} .

$$\begin{aligned} \mathcal{F}(P) &\longrightarrow^{13} \\ &(v \tilde{c}_\bullet)(v a_3)(\overline{c_5!}\langle \rangle \mid \mathcal{A}_\epsilon^5(A') \mid (v s)(a_3! \langle s \rangle. \overline{s!}\langle w_1, w_2 \rangle) \mid \\ &\quad a_3?(y_5). y_5?(x_1, x_2). (\overline{c_{17}!}\langle \rangle \mid \mathcal{A}_\epsilon^{17}(B'))) \\ &\quad \text{where } \tilde{c}_\bullet = (c_5, \dots, c_{12}, c_{17}, \dots, c_{25}) \end{aligned}$$

Now, the passing of the decomposition of w is finally simulated by two reductions: first, a synchronization on a_3 sends the endpoint of session s , which replaces variable y_5 ; then, the dual endpoint is used to send w_1, w_2 , substituting variables x_1, x_2 in $\mathcal{A}_\epsilon^{17}(B')$.

$$\begin{aligned} \mathcal{F}(P) &\longrightarrow^{14} (v \tilde{c}_{\bullet\bullet})(v s)(\overline{c_5!}\langle \rangle \mid \mathcal{A}_\epsilon^5(A') \mid \\ &\quad \overline{s!}\langle w_1, w_2 \rangle \mid s?(x_1, x_2). (\overline{c_{17}!}\langle \rangle \mid \mathcal{A}_\epsilon^{17}(B'))) \\ \mathcal{F}(P) &\longrightarrow^{15} (v \tilde{c}_{\bullet\bullet})(\overline{c_5!}\langle \rangle \mid \mathcal{A}_\epsilon^5(A') \mid \\ &\quad \overline{c_{17}!}\langle \rangle \mid \mathcal{A}_\epsilon^{17}(B')\{w_1 w_2/x_1 x_2\}) = Q \end{aligned}$$

Above, $\tilde{c}_{\bullet\bullet} = (c_5, \dots, c_{12}, c_{17}, \dots, c_{25})$. This is how $\mathcal{F}(P)$ simulates the first action of P . Notice that in Q names w_1, w_2 substitute x_1, x_2 and the first synchronization on w can be simulated on name w_1 .

Example 3.8 (A Recursive Process). Let $P = \mu X.P'$ be a process implementing a channel r with the tail-recursive session type $S = \mu t.?(Int);!(Int);t$, with $P' = r?(w).r!(-w).X$. We decompose r using S and obtain two channels typed with MSTs as in Fig. 9:

$$\begin{aligned} r_1 &: \mu t.?(?(?(?(Int); end); end); end); t \\ r_2 &: \mu t.!(?(?(?(Int); end); end); end); t \end{aligned}$$

Then, process $\mathcal{F}(P)$ is

$$(v \tilde{c})(v c^r)(c^r?(b).(v s)(b! \langle s \rangle. \overline{s!}\langle r_1, r_2 \rangle) \mid \overline{c_1!}\langle \rangle \mid \mathcal{A}_\epsilon^1(P\{r_1/r\})_\emptyset)$$

where $\tilde{c} = (c_1, \dots, c_{|P|})$ and $\mathcal{A}_\epsilon^1(P\{r_1/r\})_\emptyset$ is in Fig. 11.

In Fig. 11, $\mathcal{A}_\epsilon^1(P\{r_1/r\})$ simulates recursion in P using replication. Given some index k , process R^k mimics actions of the recursive body. It first gets a decomposition of r by interacting with the process providing recursive names on c^r (for the first instance, this is a top-level process in $\mathcal{F}(P)$). Then, it mimics the first input action on the channel received for z_1 (that is, r_1): the input of actual names for w_1 is delegated through channel redirections to name y_2 (both prefixes are highlighted in Fig. 11). Once the recursive name is used, the decomposition of recursive name is made available for the breakdown of the continuation by a communication on c^r . Similarly, in the continuation, the second action on r , output, is mimicked by r_2 (received for z_2), with the output of actual name w_1 delegated to $\overline{s'}$ (both prefixes are highlighted in Fig. 11).

Subprocess R^5 is a breakdown of the first instance of the recursive body. The replication guarded by a_1 produces a next instance, i.e., process $R^2\{x_{r_1}, x_{r_2}/r_1, r_2\}$ in \widehat{P} . By communication on a_1 and a few reductions on propagators, it gets activated: along a_1 it first receives a name for y'_1 along which it also receives: (i) recursive names r_1, r_2 for variables x_{r_1}, x_{r_2} , and (ii) a name for y_1 along which it will receive a_1 again, for future instances, as it can be seen in $\mathcal{A}_{z_x}^{k+7}(X)_g$.

$$\begin{aligned} \mathcal{A}_\epsilon^1(P\{r_1/r\})_\emptyset &= (v s_1)(c_1?(.)\overline{c_2!}\langle \rangle.\overline{c_4!}\langle \rangle \mid \\ &\quad c_2?(.)\overline{(v a_1)(s_1! \langle a_1 \rangle. (\overline{c_3!}\langle \rangle \mid c_3?(.)\mathbf{0} \mid \\ &\quad c_4?(.)s_1?(z_x).\overline{c_5!}\langle z_x \rangle} \mid \\ &\quad R^5 \mid * a_1?(y'_1).y'_1?(x_{r_1}, x_{r_2}, y_1).\widehat{P})) \end{aligned}$$

where:

$$\begin{aligned} \widehat{P} &= (v \tilde{c})(c^r?(b).(v s')(b! \langle s' \rangle. \overline{s'!}\langle x_{r_1}, x_{r_2} \rangle) \mid \overline{c_1!}\langle \rangle \mid \\ &\quad c_1?(.)y_1?(z_x).\overline{c_2!}\langle z_x \rangle \mid R^2\{x_{r_1}, x_{r_2}/r_1, r_2\}) \end{aligned}$$

$$R^k = c_k?(z_x).$$

$$(v a_1)(c^r! \langle a_1 \rangle. ((v s_1)(c_{k+1}?(y).\overline{c_{k+2}!}\langle y \rangle.\overline{c_{k+3}!}\langle \rangle \mid$$

$$c_{k+2}?(y).(v s)(y! \langle s \rangle. \overline{s!}\langle s_1 \rangle) \mid$$

$$c_{k+3}?(.)\overline{(v a_2)(s_1! \langle a_2 \rangle. (\overline{c_{k+4}!}\langle \rangle \mid$$

$$c_{k+l+4}?(.)\mathbf{0} \mid a_2?(y_2). y_2?(w_1).$$

$$(v \tilde{c})(\overline{c_{k+4}!}\langle z_x \rangle \mid \mathcal{A}_{z_x}^{k+4}(r_2! \langle -w_1 \rangle. X)_g))) \mid$$

$$a_1?(y_1).y_1?(z_1, z_2). z_1?(y).$$

$$\overline{c_{k+1}!}\langle y \rangle. c^r?(b).(v s')(b! \langle s' \rangle. \overline{s'!}\langle z_1, z_2 \rangle))))))$$

$$\mathcal{A}_{z_x}^{k+4}(r_2! \langle -w_1 \rangle. X)_g = c_k?(z_x).(v a_1)c^r! \langle a_1 \rangle.$$

$$(\mathcal{A}_{z_x}^{k+7}(X)_g \mid a_1?(y_1).y_1?(z_1, z_2). W)$$

$$W = (v a_2)(z_2! \langle a_2 \rangle. (\overline{c_{k+7}!}\langle z_x \rangle. c^r?(b).(v s)(b! \langle s \rangle. \overline{s!}\langle z_2 \rangle) \mid$$

$$a_2?(y_2).y_2?(z'_1).(v \tilde{c})(\overline{c_{k+5}!}\langle \rangle \mid$$

$$c_{k+5}?(.)z'_1?(x).\overline{c_{k+6}!}\langle x \rangle \mid$$

$$c_{k+6}?(x).(v s')(x! \langle s' \rangle. \overline{s'!}\langle -w_1 \rangle))))))$$

$$\mathcal{A}_{z_x}^{k+7}(X)_g = (v s_1)(c_{k+7}?(z_x).\overline{c_{k+8}!}\langle z_x \rangle.\overline{c_{k+9}!}\langle z_x \rangle \mid$$

$$c_{k+8}?(z_x).(v a_1)(c^r! \langle a_1 \rangle. (c_{k+9}?(z_x).\overline{s_1!}\langle z_x \rangle. \mid$$

$$\overline{c_{k+10}!}\langle \rangle c_{k+10}?(.)\mathbf{0} \mid$$

$$a_1?(y_1).y_1?(r_1, r_2).$$

$$(v s')(z_x! \langle s' \rangle. \overline{s'!}\langle r_1, r_2, s_1 \rangle))))))$$

with $g = \{X \mapsto r_1, r_2\}$

Figure 11: Breakdown of recursive process (Exam. 3.8)

3.4 Results

We establish the minimality result for π using the typability of $\mathcal{F}(\cdot)$. We need some auxiliary definitions to characterize the propagators required to decompose recursive processes.

Theorems 3.10 and 3.27 in [1] state typability results by introducing two typing environments, denoted Θ and Φ . While environment Θ is used to type linear propagators (e.g., c_k, c_{k+1}, \dots) generated by the breakdown function $\mathcal{B}_\ominus(\cdot)$, environment Φ types shared propagators used in trios that propagate breakdown of recursive names (e.g., c^r, c^v, \dots where r and v are recursive names).

Definition 3.9 (Session environment for propagators). Let Θ be the session environment and Φ be the recursive propagator environment defined in Theorem 3.10 and Theorem 3.27 [1], respectively.

Then, by applying the encoding $\llbracket \cdot \rrbracket_g^2$, we define Θ' and Φ' as follows: $\Theta' = \llbracket \Theta \rrbracket_g^2$, $\Phi' = \llbracket \Phi \rrbracket_g^2$.

We can use $\Theta' = \llbracket \Theta \rrbracket_g^2$ in the following statement, where we state the typability result for the breakdown function.

THEOREM 1 (TYPABILITY OF BREAKDOWN). *Let P be an initialized π process. If $\Gamma; \Delta, \Delta_\mu \vdash P \triangleright \diamond$, then $\mathcal{H}(\Gamma'), \Phi'; \mathcal{H}(\Delta), \Theta' \vdash \mathcal{A}_c^k(P)_g \triangleright \diamond$, where $k > 0$; $\bar{r} = \text{dom}(\Delta_\mu)$; $\Phi' = \prod_{r \in \bar{r}} c^r : \langle \langle ?(\mathcal{R}^*(\Delta_\mu(r))) \rangle \rangle; \text{end} \rangle$; and $\text{balanced}(\Theta')$ with*

$$\text{dom}(\Theta') = \{c_k, c_{k+1}, \dots, c_{k+[P]-1}\} \cup \{\overline{c_{k+1}}, \dots, \overline{c_{k+[P]-1}}\}$$

such that $\Theta'(c_k) = ?(\cdot)$; end .

PROOF. Directly by using Theorem 5.1 [10], Theorem 3.27 [1], and Theorem 5.2 [10]. See App. C.2 for details. \square

We now consider typability for the decomposition function, using $\Phi' = \llbracket \Phi \rrbracket_g^2$ as in Def. 3.9. The proof follows from Thm. 1; see App. C.2.

THEOREM 2 (MINIMALITY RESULT FOR π). *Let P be a closed π process, with $\bar{u} = \text{fn}(P)$ and $\bar{v} = \text{rn}(P)$. If $\Gamma; \Delta, \Delta_\mu \vdash P \triangleright \diamond$, where Δ_μ only involves recursive session types, then $\mathcal{H}(\Gamma\sigma); \mathcal{H}(\Delta\sigma), \mathcal{H}(\Delta_\mu\sigma) \vdash \mathcal{F}(P) \triangleright \diamond$, where $\sigma = \{\text{init}(\bar{u})/\bar{u}\}$.*

4 OPTIMIZATIONS

Although conceptually simple, the composition approach to decomposition induces redundancies. Here we propose $\mathcal{F}^*(\cdot)$, an optimization of the decomposition $\mathcal{F}(\cdot)$, and establish its static and dynamic correctness, in terms of the minimality result (cf. Thm. 4) but also *operational correspondence* (cf. Thm. 5), respectively.

4.1 Motivation

To motivate our insights, consider the process $\mathcal{A}_x^k(u_i?(w).Q)_g$ as presented in § 3.1 and Tab. 2. We identify some suboptimal features of this decomposition: (i) channel redirections; (ii) redundant synchronizations on propagators; (iii) the structure of trios is lost.

While an original process P receives a name for variable w along u_i , its breakdown does not input a breakdown of w directly, but through a series of channel redirections: u_i receives a name along which it sends restricted name s , along which it sends the restricted name s_1 and so on. Finally, the name received for y' receives \bar{w} , the breakdown of w . This redundancy is perhaps more evident in Def. 3.5, which gives the translation of types by composition: the mimicked input action is five-level nested for the original name. This is due to the composition of $\llbracket \cdot \rrbracket_g^1$ and $\llbracket \cdot \rrbracket_g^2$.

Also, $\mathcal{A}_x^k(u_i?(w).Q)_g$ features redundant communications on propagators. For example, the bound name y is locally propagated by c_{k+1} and c_{k+2} . This is the result of breaking down sequential prefixes induced by $\llbracket \cdot \rrbracket_g^1$ (not present in the original process). Last but not least, the trio structure is lost as subprocess \widehat{Q}_x is guarded and nested, and it inductively invokes the function on continuation Q . This results in an arbitrary level of process nesting, which is induced by the final application of encoding $\llbracket \cdot \rrbracket_g^2$ in the composition.

The non-optimality of $\mathcal{A}_x^k(\cdot)_g$ is more prominent in the treatment of recursive processes and recursive names. As HO does not feature recursion constructs, $\llbracket \cdot \rrbracket_g^1$ encodes recursive behaviors

by relying on abstraction passing and shared abstractions. Then, going back to π via $\llbracket \cdot \rrbracket_g^2$, this is translated to a process involving a replicated subprocess. But going through this path, the encoding of recursive process becomes convoluted. On top of that, all non-optimal features of the core fragment (as discussed for the case of input) are also present in the decomposition of recursion.

Here we develop an optimized decomposition function, denoted $\mathcal{F}^*(\cdot)$ (Def. 4.8), that avoids the redundancies described above. The optimized decomposition produces a composition of *trios processes*, with a fixed maximum number of nested prefixes. The decomposed process does not redirect channels and only introduces propagators that codify the sequentiality of the original process.

4.2 Preliminaries

We decompose a session type into a *list* of minimal session types:

Definition 4.1 (Decomposing Types). Let S and C be a session and a channel type, resp. (cf. Fig. 4). The *type decomposition function* $\mathcal{H}^*(\cdot)$ is defined in Figure 12.

Example 4.2 (Decomposing a Recursive Type). Let $S = \mu t.S'$ be a recursive session type, with $S' = ?(\text{Int}); ?(\text{Bool}); !(\text{Bool}); t$. By Fig. 12, since S is tail-recursive, $\mathcal{H}^*(S) = \mathcal{R}(S')$. Further,

$$\mathcal{R}(S') = \mu t.?(\mathcal{H}^*(\text{Int})); t, \mathcal{R}(?(\text{Bool})); !(\text{Bool}); t$$

By definition of $\mathcal{R}(\cdot)$, we obtain

$$\mathcal{H}^*(S) = \mu t.?(\text{Int}); t, \mu t.?(\text{Bool}); t, \mu t.!(\text{Bool}); t, \mathcal{R}(t)$$

(using $\mathcal{H}^*(\text{Int}) = \text{Int}$ and $\mathcal{H}^*(\text{Bool}) = \text{Bool}$). Since $\mathcal{R}(t) = \epsilon$, we have

$$\mathcal{H}^*(S) = \mu t.?(\text{Int}); t, \mu t.?(\text{Bool}); t, \mu t.!(\text{Bool}); t$$

Example 4.3 (Decomposing an Unfolded Recursive Type). Let $T = ?(\text{Bool}); !(\text{Bool}); S$ be a derived unfolding of S from Exam. 4.2. Then, by Fig. 12, $\mathcal{R}^*(T)$ is the list of minimal recursive types obtained as follows: first, $\mathcal{R}^*(T) = \mathcal{R}^*(!(\text{Bool}); \mu t.S')$ and after one more step, $\mathcal{R}^*(!(\text{Bool}); \mu t.S') = \mathcal{R}^*(\mu t.S')$. Finally, we have $\mathcal{R}^*(\mu t.S') = \mathcal{R}(S')$. We get the same list of minimal types as in Exam. 4.2:

$$\mathcal{R}^*(T) = \mu t.?(\text{Int}); t, \mu t.?(\text{Bool}); t, \mu t.!(\text{Bool}); t$$

Definition 4.4 (Decomposing Environments). Given environments Γ and Δ , we define $\mathcal{H}^*(\Gamma)$ and $\mathcal{H}^*(\Delta)$ inductively as $\mathcal{H}^*(\emptyset) = \emptyset$ and

$$\begin{aligned} \mathcal{H}^*(\Delta, u_i : S) &= \mathcal{H}^*(\Delta), (u_i, \dots, u_{i+|\mathcal{H}^*(S)|-1}) : \mathcal{H}^*(S) \\ \mathcal{H}^*(\Gamma, u_i : \langle S \rangle) &= \mathcal{H}^*(\Gamma), u_i : \mathcal{H}^*(\langle S \rangle) \end{aligned}$$

Definition 4.5 (Degree of a Process). The *optimized degree* of a process P , denoted $[P]^*$, is inductively defined as follows:

$$\left[\begin{array}{ll} [Q]^* + 1 & \text{if } P = u_i!(y).Q \text{ or } P = u_i?(y).Q \\ [Q]^* & \text{if } P = (v s : S)Q \\ [Q]^* + 1 & \text{if } P = (v r : S)Q \text{ and } \text{tr}(S) \\ [Q]^* + [R]^* + 1 & \text{if } P = Q \mid R \\ 1 & \text{if } P = \mathbf{0} \text{ or } P = X \\ [Q]^* + 1 & \text{if } P = \mu X.Q \end{array} \right.$$

$$\begin{aligned}
 \mathcal{H}^*(\text{end}) &= \text{end} \\
 \mathcal{H}^*(\langle S \rangle) &= \langle \mathcal{H}^*(S) \rangle \\
 \mathcal{H}^*(S_1, \dots, S_n) &= \mathcal{H}^*(S_1), \dots, \mathcal{H}^*(S_n) \\
 \mathcal{H}^*(! \langle C \rangle; S) &= \begin{cases} ! \langle \mathcal{H}^*(C) \rangle; \text{end} & \text{if } S = \text{end} \\ ! \langle \mathcal{H}^*(C) \rangle; \text{end}, \mathcal{H}^*(S) & \text{otherwise} \end{cases} \\
 \mathcal{H}^*(? \langle C \rangle; S) &= \begin{cases} ? \langle \mathcal{H}^*(C) \rangle; \text{end} & \text{if } S = \text{end} \\ ? \langle \mathcal{H}^*(C) \rangle; \text{end}, \mathcal{H}^*(S) & \text{otherwise} \end{cases} \\
 \mathcal{H}^*(\mu t. S') &= \mathcal{R}(S') \\
 \mathcal{H}^*(S) &= \mathcal{R}^*(S) \quad \text{where } S \neq \mu t. S' \\
 \mathcal{R}(t) &= \epsilon \\
 \mathcal{R}(! \langle C \rangle; S) &= \mu t. ! \langle \mathcal{H}^*(C) \rangle; t, \mathcal{R}(S) \\
 \mathcal{R}(? \langle C \rangle; S) &= \mu t. ? \langle \mathcal{H}^*(C) \rangle; t, \mathcal{R}(S) \\
 \mathcal{R}^*(? \langle C \rangle; S) &= \mathcal{R}^*(! \langle C \rangle; S) = \mathcal{R}^*(S) \\
 \mathcal{R}^*(\mu t. S) &= \mathcal{R}(S)
 \end{aligned}$$

Figure 12: Decomposition of types $\mathcal{H}^*(\cdot)$ (cf. Def. 4.1)

As before, given a finite tuple of names $\tilde{u} = (a, b, s, s', \dots)$, we write $\text{init}(\tilde{u})$ to denote the tuple $(a_1, b_1, s_1, s'_1, \dots)$; also, we say that a process is initialized if all of its names have some index.

Given a tuple of initialized names \tilde{u} and a tuple of indexed names \tilde{x} , it is useful to collect those names in \tilde{x} that appear in \tilde{u} .

Definition 4.6 (Free indexed names). Let \tilde{u} and \tilde{x} be two tuples of names. We define the set $\text{fnb}(\tilde{u}, \tilde{x})$ as $\{z_k : z_i \in \tilde{u} \wedge z_k \in \tilde{x}\}$.

As usual, we treat sets of names as tuples (and vice-versa). By abusing notation, given a process P , we shall write $\text{fnb}(P, \tilde{y})$ to stand for $\text{fnb}(\text{fn}(P), \tilde{y})$. Then, we have that $\text{fnb}(P, \tilde{x}) \subseteq \tilde{x}$. In the definition of the breakdown function, this notion allows us to conveniently determine a *context* for a subsequent trio.

Remark 1. Whenever $c_k?(y)$ (resp. $\overline{c_k}!\langle y \rangle$) with $\tilde{y} = \epsilon$, we shall write $c_k?()$ (resp. $\overline{c_k}!\langle \rangle$) to stand for $c_k?(y)$ (resp. $\overline{c_k}!\langle y \rangle$) such that $c_k : ?(\langle \text{end} \rangle); \text{end}$ (resp. $\overline{c_k} : !(\langle \text{end} \rangle); \text{end}$).

Definition 4.7 (Index function). Let S be an (unfolded) recursive session type. The function $f(S)$ is defined as follows:

$$f(S) = \begin{cases} f'_0(S' \{S/t\}) & \text{if } S = \mu t. S' \\ f'_0(S) & \text{otherwise} \end{cases}$$

where: $f'_l(! \langle U \rangle; S) = f'_{l+1}(S)$, $f'_l(? \langle U \rangle; S) = f'_{l+1}(S)$, and $f'_l(\mu t. S) = |\mathcal{R}(S)| - l + 1$.

Given a process P , we write $\text{frv}(P)$ to denote that P has a free recursive variable.

4.3 The Optimized Decomposition

We define the optimized decomposition $\mathcal{F}^*(\cdot)$ by relying on the revised breakdown function $\mathbb{A}_{\text{rec } \tilde{x}}^k(\cdot)$ (cf. § 4.3.1). Given a context \tilde{x} and a $k > 0$, $\mathbb{A}_{\text{rec } \tilde{x}}^k(\cdot)$ is defined on initialized processes. Table 3 gives the definition: we use an auxiliary function for recursive processes,

denoted $\mathbb{A}_{\text{rec } \tilde{x}}^k(\cdot)_g$ (cf. § 4.3.2), where parameter g is a mapping from recursive variables to a list of name variables.

In the following, to keep presentation simple, we assume processes $\mu X.P$ in which P does not contain a subprocess of shape $\mu Y.P'$. The generalization of our decomposition without this assumption is not difficult, but is notationally heavy.

4.3.1 The Breakdown Function. We describe entries 1-7 in Table 3.

1. Input Process $\mathbb{A}_{\text{rec } \tilde{x}}^k(u_i?(y).Q)$ consists of a leading trio that mimics the input and runs in parallel with the breakdown of Q . In the trio, a context \tilde{x} is expected along c_k . Then, an input on u_l mimics the input action: it expects the *decomposition* of name y , denoted \tilde{y} . To decompose y we use its type: if $y : S$ then $\tilde{y} = (y_1, \dots, y_l | \mathcal{H}^*(S)_l)$. The index of u_l depends on the type of u_i . Intuitively, if u_i is tail-recursive then $l = f(S)$ (Def. 4.7) as index and we do not increment it, as the same decomposition of u_i should be used to mimic a new instance in the continuation. Otherwise, if u_i is linear then we use the substitution $\sigma = \{u_{i+1}/u_i\}$ to increment it in Q . Next, the context $\tilde{z} = \text{fnb}(Q, \tilde{x}\tilde{y} \setminus \tilde{w})$ is propagated, where $\tilde{w} = (u_i)$ or $\tilde{w} = \epsilon$.

2. Output Process $\mathbb{A}_{\text{rec } \tilde{x}}^k(u_i!(y_j).Q)$ sends the decomposition of y on u_l , with l as in the input case. We decompose name y_j based on its type S : $\tilde{y} = (y_j, \dots, y_{j+l} | \mathcal{H}^*(S)_{j+l-1})$. The context to be propagated is $\tilde{z} = \text{fnb}(P, \tilde{x} \setminus \tilde{w})$, where \tilde{w} and σ are as in the input case.

3. Restriction (Non-recursive name) The breakdown of process $(\nu s : C)Q$ is $(\nu \tilde{s} : \mathcal{H}^*(C)) \mathcal{B}_{\tilde{x}}^k(Q\sigma)$, where s is decomposed using C : $\tilde{s} = (s_1, \dots, s_l | \mathcal{H}^*(C)_l)$. Since (νs) binds s and its dual \tilde{s} (or only s if C is a shared type) the substitution σ is simply $\{s_1\tilde{s}_1/s\}$ and initializes indexes in Q .

4. Restriction (Recursive name) As in the previous case, in the breakdown of $(\nu s : \mu t.S)Q$ the name s is decomposed into \tilde{s} by relying on $\mu t.S$. Here the breakdown consists of the breakdown of Q running in parallel with a control trio, which appends restricted (recursive) names \tilde{s} and \tilde{s} to the context, i.e., $\tilde{z} = \tilde{x}, \tilde{s}, \tilde{s}$.

5. Composition The breakdown of process $Q_1 | Q_2$ uses a control trio to trigger the breakdowns of Q_1 and Q_2 , similarly as before.

6. Inaction The breakdown of $\mathbf{0}$ is simply an input prefix that receives an empty context (i.e., $\tilde{x} = \epsilon$).

7. Recursion The breakdown of $\mu X.P$ is as follows:

$$(\nu c_X^r)(c_k?(\tilde{x}).\overline{c_{k+1}^r}!\langle \tilde{z} \rangle. \mu X. c_X^r?(\tilde{y}).\overline{c_{k+1}^r}!\langle \tilde{y} \rangle. X | \mathbb{A}_{\text{rec } \tilde{x}}^{k+1}(P)_g)$$

We have a control trio and the breakdown of P , obtained using $\mathbb{A}_{\text{rec } \tilde{x}}^k(\cdot)_g$ (§ 4.3.2). The trio receives the context \tilde{x} on c_k and propagates it further. To ensure typability, we bind all session free names of P using the context \tilde{z} , which contains the decomposition of those free names. This context is needed to break down P , and so we record it as $g = \{X \mapsto \tilde{z}\}$ in the definition of $\mathbb{A}_{\text{rec } \tilde{x}}^{k+1}(P)_g$. This way, \tilde{z} will be propagated all the way until reaching X .

Next, the recursive trio is enabled, and receives \tilde{y} along c_X^r , with $|\tilde{z}| = |\tilde{y}|$ and $l = |P|$. The tuple \tilde{y} is propagated to the first trio of $\mathbb{A}_{\text{rec } \tilde{x}}^{k+1}(P)_g$. By definition of $\mathbb{A}_{\text{rec } \tilde{x}}^{k+1}(P)_g$, its propagator c_X^r will send the same context as received by the first trio. Hence, the recursive part of the control trio keeps sending this context to the next instances of recursive trios of $\mathbb{A}_{\text{rec } \tilde{x}}^{k+1}(P)_g$.

	P	$A_{\tilde{x}}^k(P)$	
1	$u_i?(y).Q$	$c_k?(\tilde{x}).u_i?(\tilde{y}).\overline{c_{k+1}}! \langle \tilde{z} \rangle \mid A_{\tilde{z}}^{k+1}(Q\sigma)$	$y_j : S \wedge \tilde{y} = (y_1, \dots, y_{ S })$ $\tilde{w} = (\text{lin}(u_i)) ? \{u_i\} : \epsilon$ $\tilde{z} = \text{fnb}(Q, \tilde{x}\tilde{y} \setminus \tilde{w})$ $l = (\text{tr}(u_i)) ? f(S) : i$ $\sigma = \text{next}(u_i) \cdot \{y_i/y\}$
2	$u_i!(y_j).Q$	$c_k?(\tilde{x}).u_i!\langle \tilde{y} \rangle.\overline{c_{k+1}}! \langle \tilde{z} \rangle \mid A_{\tilde{z}}^{k+1}(Q\sigma)$	$y_j : S \wedge \tilde{y} = (y_j, \dots, y_{j+ \mathcal{H}^*(S) -1})$ $\tilde{w} = (\text{lin}(u_i)) ? \{u_i\} : \epsilon$ $\tilde{z} = \text{fnb}(Q, \tilde{x} \setminus \tilde{w})$ $l = (\text{tr}(u_i)) ? f(S) : i$ $\sigma = \text{next}(u_i)$
3	$(v s : C)Q$	$(v \tilde{s} : \mathcal{H}^*(C)) A_{\tilde{x}}^k(Q\sigma)$	$\tilde{s} = (s_1, \dots, s_{ \mathcal{H}^*(C) })$ $\sigma = \{s_i \tilde{s}_i / s \tilde{s}\}$
4	$(v s : \mu t.S)Q$	$(v \tilde{s} : \mathcal{R}(S)) (c_k?(\tilde{x}).\overline{c_{k+1}}! \langle \tilde{z} \rangle . 0 \mid A_{\tilde{z}}^{k+1}(Q))$	$\text{tr}(\mu t.S) \quad \tilde{s} = (s_1, \dots, s_{ \mathcal{R}(S) })$ $\tilde{z} = \tilde{x}, \tilde{s}, \tilde{s} \quad \tilde{s} = (\tilde{s}_1, \dots, \tilde{s}_{ \mathcal{R}(S) })$
5	$Q_1 \mid Q_2$	$c_k?(\tilde{x}).\overline{c_{k+1}}! \langle \tilde{y} \rangle . \overline{c_{k+1}}! \langle \tilde{z} \rangle \mid A_{\tilde{y}}^{k+1}(Q_1) \mid A_{\tilde{z}}^{k+1}(Q_2)$	$\tilde{y} = \text{fnb}(Q_1, \tilde{x}) \quad \tilde{z} = \text{fnb}(Q_2, \tilde{x})$ $l = Q_1 $
6	0	$c_k?().0$	
7	$\mu X.P$	$(v c_X^r)(c_k?(\tilde{x}).\overline{c_{k+1}}! \langle \tilde{z} \rangle . \mu X.c_X^r?(\tilde{y}).\overline{c_{k+1}}! \langle \tilde{y} \rangle . X \mid A_{\text{rec } \tilde{z}}^{k+1}(P)g)$	$\tilde{n} = \text{fs}(P) \quad \tilde{n} : \tilde{C} \wedge \tilde{z} = \text{bn}(\tilde{n} : \tilde{C})$ $ \tilde{z} = \tilde{y} \quad g = \{X \mapsto \tilde{z}\}$
	P	$A_{\text{rec } \tilde{x}}^k(P)g$	
8	$u_i!(y_j).Q$	$\mu X.c_k^r?(\tilde{x}).u_i!\langle \tilde{y} \rangle . \overline{c_{k+1}}! \langle \tilde{z} \rangle . X \mid A_{\text{rec } \tilde{z}}^{k+1}(Q\sigma)g \quad (\text{if } g \neq \emptyset)$ $\mu X.c_k^r?(\tilde{x}).(u_i!\langle \tilde{y} \rangle . \overline{c_{k+1}}! \langle \tilde{z} \rangle \mid X) \mid A_{\text{rec } \tilde{z}}^{k+1}(Q\sigma)g \quad (\text{if } g = \emptyset)$	$y : T \wedge (y_j, \dots, y_{j+ \mathcal{H}^*(T) -1})$ $\tilde{w} = (\text{lin}(u_i)) ? \{u_i\} : \epsilon$ $\tilde{z} = g(X) \cup \text{fnb}(Q, \tilde{x} \setminus \tilde{w})$ $l = (\text{tr}(u_i)) ? f(S) : i$ $\sigma = \text{next}(u_i)$
9	$u_i?(y).Q$	$\mu X.c_k^r?(\tilde{x}).u_i?(\tilde{y}).\overline{c_{k+1}}! \langle \tilde{z} \rangle . X \mid A_{\text{rec } \tilde{z}}^{k+1}(Q\sigma)g \quad (\text{if } g \neq \emptyset)$ $\mu X.c_k^r?(\tilde{x}).(u_i?(\tilde{y}).\overline{c_{k+1}}! \langle \tilde{z} \rangle \mid X) \mid A_{\text{rec } \tilde{z}}^{k+1}(Q\sigma)g \quad (\text{if } g = \emptyset)$	$\tilde{w} = (\text{lin}(u_i)) ? \{u_i\} : \epsilon$ $\tilde{z} = g(X) \cup \text{fnb}(Q, \tilde{x}\tilde{y} \setminus \tilde{w})$ $l = (\text{tr}(u_i)) ? f(S) : i$ $\sigma = \text{next}(u_i) \cdot \{y_i/y\}$
10	$Q_1 \mid Q_2$	$\mu X.c_k^r?(\tilde{x}).(\overline{c_{k+1}}! \langle \tilde{y}_1 \rangle . X \mid \overline{c_{k+1}}! \langle \tilde{y}_2 \rangle) \mid A_{\text{rec } \tilde{y}_1}^{k+1}(Q_1)g \mid A_{\text{rec } \tilde{y}_2}^{k+1}(Q_2)\emptyset \quad (\text{if } g \neq \emptyset)$ $\mu X.c_k^r?(\tilde{x}).(\overline{c_{k+1}}! \langle \tilde{y}_1 \rangle \mid \overline{c_{k+1}}! \langle \tilde{y}_2 \rangle \mid X) \mid A_{\text{rec } \tilde{y}_1}^{k+1}(Q_1)\emptyset \mid A_{\text{rec } \tilde{y}_2}^{k+1}(Q_2)\emptyset \quad (\text{if } g = \emptyset)$	$\text{frv}(Q_1)$ $\tilde{y}_1 = g(X) \cup \text{fnb}(Q_1, \tilde{x})$ $\tilde{y}_2 = \text{fnb}(Q_2, \tilde{x})$ $l = Q_1 $
11	$(v s : C)Q$	$\mu X.(v \tilde{s} : \mathcal{H}^*(C))c_k^r?(\tilde{x}).\overline{c_{k+1}}! \langle \tilde{z} \rangle . X \mid A_{\text{rec } \tilde{z}}^{k+1}(Q\sigma)g \quad (\text{if } g \neq \emptyset)$ $\mu X.(v \tilde{s} : \mathcal{H}^*(C))c_k^r?(\tilde{x}).(\overline{c_{k+1}}! \langle \tilde{z} \rangle \mid X) \mid A_{\text{rec } \tilde{z}}^{k+1}(Q\sigma)g \quad (\text{if } g = \emptyset)$	$\tilde{s} = (s_1, \dots, s_{ \mathcal{H}^*(S) })$ $\tilde{s} = (\text{lin}(S)) ? (\tilde{s}_1, \dots, \tilde{s}_{ \mathcal{H}^*(S) }) : \epsilon$ $\tilde{z} = \tilde{x}, \tilde{s}, \tilde{s} \quad \sigma = \{s_i \tilde{s}_i / s \tilde{s}\}$
12	X	$\mu X.c_k^r?(\tilde{x}).c_X^r! \langle \tilde{x} \rangle X \quad (\text{if } g \neq \emptyset)$ $\mu X.c_k^r?().(c_X^r! \langle \rangle \mid X) \quad (\text{if } g = \emptyset)$	
13	0	$c_k^r?().0$	

Table 3: Optimized breakdown function $A_{\tilde{x}}^k(\cdot)$ for processes, and auxiliary function for recursive processes $A_{\text{rec } \tilde{x}}^k(\cdot)g$.

Notice that the leading trio actually has four prefixes. This simplifies our presentation: this trio can be broken down into two trios by introducing an extra propagator c_{k+1} to send over c_{k+2}^r .

4.3.2 Handling P in $\mu X.P$. As already mentioned, we use the auxiliary function $A_{\text{rec } \tilde{x}}^k(\cdot)g$ to generate *recursive trios*.

We discuss entries 8-11 in Tab. 3 (other entries are similar as before). A key observation is that parameter g can be empty. To see this, consider a process like $P = \mu X.(Q_1 \mid Q_2)$ where X occurs free

in Q_1 but not in Q_2 . If X occurs free in Q_1 then its decomposition will have a non-empty g , whereas the g for Q_2 will be empty. In the recursive trios of Tab. 3, the difference between $g \neq \emptyset$ and $g = \emptyset$ is subtle: in the former case, X appears guarded by a propagator; in the latter case, it appears unguarded in a parallel composition. This way, trios in the breakdown of Q_2 replicate themselves on a trigger from the breakdown of Q_1 .

Given this difference, we only describe the cases when $g \neq \emptyset$:

8 / 9. Output and Input The breakdown of $u_i!\langle y_j \rangle.Q$ consists of the breakdown of Q in parallel with a leading trio, a recursive process whose body is defined as in $\mathcal{B}(\cdot)$. As names $g(X)$ may not appear free in Q , we must ensure that a context \bar{z} for the recursive body is propagated. The breakdown of $r?(y).Q$ is defined similarly.

10. Parallel Composition We discuss the breakdown of $Q_1 \mid Q_2$ assuming $\text{frv}(Q_1)$. We take $\bar{y}_1 = g(X) \cup \text{fnb}(Q_1, \bar{x})$ to ensure that $g(X)$ is propagated to the breakdown of X . The role of c_{k+l+1}^r is to enact a new instance of the breakdown of Q_2 ; it has a shared type to enable replication. In a running process, the number of these triggers in parallel denotes the number of available instances of Q_2 .

11. Recursive Variable In this case, the breakdown is a control trio that receives the context \bar{x} from a preceding trio and propagates it again to the first control trio of the breakdown of a recursive process along c_X^r . Notice that by construction we have $\bar{x} = g(X)$.

We may now define the optimized process decomposition:

Definition 4.8 (Decomposing Processes, Optimized). Let P be a π process with $\bar{u} = \text{fn}(P)$ and $\bar{v} = \text{rn}(P)$. Given the breakdown function $\mathbb{A}_{\bar{x}}^k(\cdot)$ in Table 3, the *optimized decomposition* of P , denoted $\mathcal{F}^*(P)$, is defined as

$$\mathcal{F}^*(P) = (\nu \bar{c})(\bar{c}_k!\langle \bar{r} \rangle. \mathbf{0} \mid \mathbb{A}_{\bar{r}}^k(P\sigma))$$

where: $k > 0$; $\bar{c} = (c_k, \dots, c_{k+\lfloor P \rfloor - 1})$; \bar{r} such that for $v \in \bar{v}$ and $v : S$ ($v_1, \dots, v_{|\mathcal{R}(S)|} \subseteq \bar{r}$; and $\sigma = \{\text{init}(\bar{u})/\bar{u}\}$.

4.4 Examples

We now illustrate $\mathcal{F}^*(\cdot)$, $\mathbb{A}_{\bar{x}}^k(\cdot)$, $\mathbb{A}_{\text{rec } \bar{x}}^k(\cdot)_g$, and $\mathcal{H}^*(\cdot)$.

Example 4.9 (Exam. 3.7, Revisited). Consider again the process $P = (\nu u)(A \mid B)$ as in Exam. 3.7. Recall that P implements session types $S = !\langle \bar{T} \rangle$; end and $T = ?(\text{Int}); !(\text{Bool})$; end.

By Def. 4.5, $\lfloor P \rfloor^* = 9$. The optimized decomposition of P is:

$$\mathcal{F}^*(P) = (\nu \bar{c})(\bar{c}_1!\langle \rangle \mid (\nu u_1)\mathbb{A}_{\bar{c}}^1((A \mid B)\sigma'))$$

where $\sigma' = \text{init}(\text{fn}(P)) \cdot \{u_1\bar{u}_1/\bar{u}\}$ and $\bar{c} = (c_1, \dots, c_9)$. We have:

$$\mathbb{A}_{\bar{c}}^1((A \mid B)\sigma') = c_1?().\bar{c}_2!\langle \rangle.\bar{c}_6!\langle \rangle \mid \mathbb{A}_{\bar{c}}^2(A\sigma') \mid \mathbb{A}_{\bar{c}}^6(B\sigma')$$

The breakdowns of sub-processes A and B are as follows:

$$\mathbb{A}_{\bar{c}}^2(A\sigma') = c_2?().u_1!\langle w_1, w_2 \rangle.\bar{c}_3!\langle \rangle \mid c_3?().\bar{w}_1?(t).\bar{c}_4!\langle \rangle \mid c_4?().\bar{w}_2!\langle \text{odd}(t) \rangle.\bar{c}_5!\langle \rangle \mid c_5?().\mathbf{0}$$

$$\mathbb{A}_{\bar{c}}^6(B\sigma') = c_6?().\bar{u}_1?(x_1, x_2).\bar{c}_7!\langle x_1, x_2 \rangle \mid c_7?(x_1, x_2).x_1!\langle 5 \rangle.\bar{c}_8!\langle x_2 \rangle \mid c_8?(x_2).x_2?(b_1).\bar{c}_9!\langle \rangle \mid c_9?().\mathbf{0}$$

Name w is decomposed as indexed names \bar{w}_1, \bar{w}_2 ; by using $\mathcal{H}^*(\cdot)$ (Def. 4.1) on T , their MSTs are $M_1 = !(\text{Int})$; end and $M_2 = ?(\text{Bool})$; end, respectively. Name u_1 is the decomposition of name u and it is typed

with $!(\bar{M}_1, \bar{M}_2)$; end. After a few administrative reductions on c_1 , c_2 , and c_6 , $\mathcal{F}^*(P)$ mimics the first source communication:

$$\begin{aligned} \mathcal{F}^*(P) &\longrightarrow^3 (\nu \bar{c}_*) (\bar{c}_3!\langle \rangle \mid \mathbb{A}_{\bar{c}}^3(\bar{w}?(t).\bar{w}!\langle \text{odd}(t) \rangle.\mathbf{0}) \mid \\ &\quad \bar{u}_1?(x_1, x_2).\bar{c}_7!\langle x_1, x_2 \rangle \mid \mathbb{A}_{x_1, x_2}^7(x!\langle 5 \rangle.x?(b).\mathbf{0})) \\ &\longrightarrow (\nu \bar{c}_*) (\bar{c}_3!\langle \rangle \mid \mathbb{A}_{\bar{c}}^2(\bar{w}?(t).\bar{w}!\langle \text{odd}(t) \rangle.\mathbf{0}) \mid \bar{c}_7!\langle w_1, w_2 \rangle \mid \\ &\quad \mathbb{A}_{x_1, x_2}^7(x!\langle 5 \rangle.x?(b).\mathbf{0})) \end{aligned}$$

Above, $\bar{c}_* = (c_3, c_4, c_5, c_7, c_8, c_9)$. After reductions on c_3 and c_7 , name w_1 substitutes x_1 and the communication along w_1 can be mimicked:

$$\begin{aligned} \mathcal{F}^*(P) &\longrightarrow^6 (\nu \bar{c}_{**}) (\bar{w}_1?(t).\bar{c}_4!\langle \rangle \mid c_4?().\bar{w}_2!\langle \text{odd}(t) \rangle.\bar{c}_5!\langle \rangle \mid \\ &\quad c_5?().\mathbf{0} \mid w_1!\langle 5 \rangle.\bar{c}_8!\langle w_2 \rangle \mid \\ &\quad c_8?(x_2).x_2?(b_1).\bar{c}_9!\langle \rangle \mid c_9?().\mathbf{0}) \\ &\longrightarrow (\nu \bar{c}_{**}) (\bar{c}_4!\langle 5 \rangle \mid c_4?(t).\bar{w}_2!\langle \text{odd}(t) \rangle.\bar{c}_5!\langle \rangle \mid c_5?().\mathbf{0} \mid \\ &\quad \bar{c}_8!\langle w_2 \rangle \mid c_8?(x_2).x_2?(b_1).\bar{c}_9!\langle \rangle \mid c_9?().\mathbf{0}) \end{aligned}$$

Above, $\bar{c}_{**} = (c_4, c_5, c_8, c_9)$. Further reductions follow similarly.

Example 4.10 (Example 3.8, Revisited). Consider again the tail-recursive session type $S = \mu t.?(Int); !(\text{Int}); t$. Also, let R be a process implementing a channel r with type with S as follows:

$$R = \mu X.R' \quad R' = r?(z).r!\langle -z \rangle.X$$

We decompose name r using S and obtain two channels typed with MSTs as in Fig. 12. We have: $r_1 : \mu t.?(Int); t$ and $r_2 : \mu t.!\langle Int \rangle; t$.

The trios produced by $\mathbb{A}_{\bar{c}}^k(R)$ satisfy two properties: they (1) mimic the recursive behavior of R and (2) use the same decomposition of channel r (i.e., r_1, r_2) in every instance.

To accomplish (1), each trio of the breakdown of the recursion body is a recursive trio. For (2), we need two things. First, we expect to receive all recursive names in the context \bar{x} when entering the decomposition of the recursion body; further, each trio should use one recursive name from the names received and propagate *all* of them to subsequent trio. Second, we need an extra control trio when breaking down prefix μX : this trio (i) receives recursive names from the last trio in the breakdown of the recursion body and (ii) activates another instance with these recursive names.

Using these ideas, we have the decomposed process $\mathbb{A}_{r_1, r_2}^1(R)$:

$$c_1?(r_1, r_2).\bar{c}_2!\langle r_1, r_2 \rangle.\mu X.c_X^r?(y_1, y_2).\bar{c}_2!\langle y_1, y_2 \rangle.X \mid \mathbb{A}_{\text{rec } r_1, r_2}^2(R')$$

where $\mathbb{A}_{\text{rec } r_1, r_2}^2(R')$ is the composition of three recursive trios:

$$\begin{aligned} &\mu X.c_2^r?(y_1, y_2).r_1?(z_1).\bar{c}_3!\langle y_1, y_2, z_1 \rangle.X \mid \\ &\mu X.c_3^r?(y_1, y_2, z_1).r_2?(-z_1).\bar{c}_4!\langle y_1, y_2 \rangle.X \mid \\ &\mu X.c_4^r?(y_1, y_2).\bar{c}_X^r!\langle y_1, y_2 \rangle.X \end{aligned}$$

c_2^r will first activate the recursive trios with context (r_1, r_2) . Next, each trio uses one of r_1, r_2 and propagate them both mimicking the recursion body. The last recursive trio sends r_1, r_2 to the top-level control trio, so it can enact another instance of the decomposition of the recursion body by activating the first recursive trio.

4.5 Measuring the Optimization

Here we measure the improvements of $\mathcal{F}^*(\cdot)$ over $\mathcal{F}(\cdot)$. A key metric for comparison is the number of prefixes/synchronizations induced by each decomposition. This includes (1) the number of prefixes involved in *channel redirections* and (2) the *number of propagators*; both can be counted by already defined notions:

- (1) Channel redirections can be counted by the levels of nesting in the decompositions of types (cf. Fig. 9 and Fig. 12)
- (2) The number of propagators is determined by the degree of a process (cf. Def. 3.2 and Def. 4.5)

These two metrics are related; let us discuss them in detail.

Channel redirections. The decompositions of types for $\mathcal{F}(\cdot)$ and $\mathcal{F}^*(\cdot)$ abstractly describe the respective channel redirections. The type decomposition for $\mathcal{F}(\cdot)$ (Fig. 9) defines 5 levels of nesting for the translation of input/output types. Then, at the level of (decomposed) processes, channels with these types implement redirections: the nesting levels correspond to 5 additional prefixes in the decomposed process that mimic a source input/output action. In contrast, the type decomposition for $\mathcal{F}^*(\cdot)$ (Fig. 12) induces no nesting, and so at the level of processes there are no additional prefixes.

Number of propagators. We define auxiliary functions to count the number of propagators induced by $\mathcal{F}(\cdot)$ and $\mathcal{F}^*(\cdot)$. These functions, denoted $\#(\cdot)$ and $\#^*(\cdot)$, respectively, are defined using the degree functions $\lfloor \cdot \rfloor$ and $\lfloor \cdot \rfloor^*$ given by Def. 3.2 and Def. 4.5.

Remarkably, $\lfloor \cdot \rfloor$ and $\#(\cdot)$ are not equal. The difference lies in the number of tail-recursive names in a process. In $\mathcal{F}(\cdot)$ there are propagators c_k but also c^r , used for recursive names. Def. 3.2, however, only counts propagators of form c_k . For any P , the number of propagators c^r in $\mathcal{F}(P)$ is the number of free and bound tail-recursive names in P . We remark that, by definition, there may be more than one occurrence of a propagator c^r in $\mathcal{F}(P)$: there is at least one prefix with subject c^r ; further occurrences depend on the sequencing structure of the (recursive) type assigned to r . On the other hand, in $\mathcal{F}^*(P)$ there are propagators c_k and propagators c_X^r , whose number corresponds to the number of recursive variables in the process. To define $\#(\cdot)$ and $\#^*(\cdot)$, we write $\text{brn}(P)$ to denote bound occurrences of recursive names and $\#_X(P)$ to denote the number of occurrences of recursive variables.

Definition 4.11 (Propagators in $\mathcal{F}(P)$ and $\mathcal{F}^(P)$).* Given a process P , the number of propagators in each decomposition is given by

$$\#(P) = \lfloor P \rfloor + 2 \cdot |\text{brn}(P)| + |\text{rn}(P)| \quad \#^*(P) = \lfloor P \rfloor^* + \#_X(P)$$

Notice that $\#^*(P)$ gives the exact number of actions induced by propagators in $\mathcal{F}^*(P)$; in contrast, due to propagators c^r , $\#(P)$ gives the *least* number of such actions in $\mathcal{F}(P)$.

In general, we have $\#(P) \geq \#^*(P)$, but we can be more precise for a broad class of processes. We say that a π process $P \neq \mathbf{0}$ is in *normal form* if $P = (v\tilde{n})(Q_1 \mid \dots \mid Q_n)$, where each Q_i (with $i \in \{1, \dots, n\}$) is not $\mathbf{0}$ and does not contain restriction nor parallel composition at top-level. We have the following result; see App. C.3.

Proposition 4.1. *If P is in normal form then $\#(P) \geq \frac{5}{3} \cdot \#^*(P)$.*

This result implies that the number of (extra) synchronizations induced by propagators in $\mathcal{F}(P)$ is larger than in $\mathcal{F}^*(P)$.

4.6 Results

4.6.1 Static Correctness. We first state Thm. 3, which ensures the typability of $\mathbb{A}_x^k(\cdot)$ under MSTs. We rely on an auxiliary predicate:

Definition 4.12 (Indexed Names). Suppose some typing environments Γ, Δ . Let \tilde{x}, \tilde{y} be two tuples of indexed names. We write $\text{indexed}_{\Gamma, \Delta}(\tilde{y}, \tilde{x})$ for the predicate

$$\forall z_i. (z_i \in \tilde{x} \Leftrightarrow ((z_i, \dots, z_{i+m-1}) \subseteq \tilde{y}) \wedge m = |\mathcal{H}^*(\Gamma, \Delta)(z_i)|)$$

THEOREM 3 (TYPABILITY OF BREAKDOWN). *Let P be an initialized process. If $\Gamma; \Delta \vdash P \triangleright \diamond$ then*

$$\mathcal{H}^*(\Gamma \setminus \tilde{x}); \mathcal{H}^*(\Delta \setminus \tilde{x}), \Theta \vdash \mathbb{A}_{\tilde{y}}^k(P) \triangleright \diamond \quad (k > 0)$$

where $\tilde{x} \subseteq \text{fn}(P)$ and \tilde{y} such that $\text{indexed}_{\Gamma, \Delta}(\tilde{y}, \tilde{x})$. Also, $\text{balanced}(\Theta)$ with

$$\text{dom}(\Theta) = \{c_k, c_{k+1}, \dots, c_{k+|P|-1}\} \cup \{\overline{c_{k+1}}, \dots, \overline{c_{k+|P|-1}}\}$$

and $\Theta(c_k) = ?(\tilde{M}); \text{end}$, where $\tilde{M} = (\mathcal{H}^*(\Gamma), \mathcal{H}^*(\Delta))(\tilde{y})$.

PROOF. By induction of the structure of P ; see § C.4 for details. \square

We now (re)state the minimality result, now based on the decomposition $\mathcal{F}^*(\cdot)$. The proof follows from Thm. 3; see App. C.5.

THEOREM 4 (MINIMALITY RESULT FOR π , OPTIMIZED). *Let P be a π process with $\tilde{u} = \text{fn}(P)$. If $\Gamma; \Delta \vdash P \triangleright \diamond$ then $\mathcal{H}^*(\Gamma\sigma); \mathcal{H}^*(\Delta\sigma) \vdash \mathcal{F}^*(P) \triangleright \diamond$, where $\sigma = \{\text{init}(\tilde{u})/\tilde{u}\}$.*

4.6.2 Dynamic Correctness. As a complement to the minimality result, we have established that P and $\mathcal{F}^*(P)$ are *behaviorally equivalent* (Thm. 5). We overview this result and its required notions.

Thm. 5 relies on *MST-bisimilarity* (cf. Def. 4.17, \approx^M), a variant of the *characteristic bisimilarity* in [9]. We discuss key differences between the two notions. First, we let an action along name n to be mimicked by an action on a possibly indexed name n_i , for some i .

Definition 4.13 (Indexed name). Given a name n , we write \tilde{n} to either denote n or any indexed name n_i , with $i > 0$.

Suppose we wish to relate P and Q using \approx^M , and that P performs an output action involving name v . In our setting, Q should send a *tuple* of names: the decomposition of v . The second difference is that output objects should be related by the relation \triangleright_c :

Definition 4.14 (Relating names). Let ϵ denote the empty list. We define \triangleright_c as the relation on names defined as

$$\frac{}{\epsilon \triangleright_c \epsilon} \quad \frac{\Gamma; \Delta \vdash n_i \triangleright C}{n_i \triangleright_c (n_i, \dots, n_{i+|\mathcal{H}^*(C)|-1})} \quad \frac{\tilde{n} \triangleright_c \tilde{m}_1 \quad n_i \triangleright_c \tilde{m}_2}{\tilde{n}, n_i \triangleright_c \tilde{m}_1, \tilde{m}_2}$$

Characteristic bisimilarity equates typed processes by relying on *characteristic processes* and *trigger processes*. These notions, which we recall below, need to be adjusted for them to work with MSTs.

Definition 4.15 (Characteristic trigger process [9]). The characteristic trigger process for type C is

$$t \leftarrow_c v : C \stackrel{\text{def}}{=} t?(x).(v s)(s?(y).\llbracket C \rrbracket^y \mid \bar{s}!(v).\mathbf{0})$$

where $\llbracket C \rrbracket^y$ is the characteristic process for C on name y [9].

Our variant of trigger processes is defined as follows:

Definition 4.16 (Minimal characteristic trigger process). Given a type C , the trigger process is

$$t \leftarrow_m v_i : C \stackrel{\text{def}}{=} t_1?(x).(v s_1).(s_1?(\bar{y}).\langle C \rangle_i^y \mid \bar{s}_1! \langle \bar{v} \rangle.0)$$

where $v_i \succ_c \bar{v}$, $y_i \succ_c \bar{y}$, and $\langle C \rangle_i^y$ is a minimal characteristic process for type C on name y (see App. C.2 for a definition).

We are now ready to define MST-bisimilarity:

Definition 4.17 (MST-Bisimilarity). A typed relation \mathfrak{R} is an *MST bisimulation* if for all $\Gamma_1; \Delta_1 \vdash P_1 \mathfrak{R} \Gamma_2; \Delta_2 \vdash Q_1$,

- (1) Whenever $\Gamma_1; \Delta_1 \vdash P_1 \xrightarrow{(v \bar{m}_1)n!(\bar{v}:C_1)} \Delta'_1; \Lambda'_1 \vdash P_2$ then there exist Q_2, Δ'_2 , and σ_v such that $\Gamma_2; \Delta_2 \vdash Q_1 \xrightarrow{(v \bar{m}_2)\bar{n}!(\bar{v}:\mathcal{H}^*(C))} \Delta'_2 \vdash Q_2$ where $v\sigma_v \succ_c \bar{v}$ and, for a fresh t ,

$$\Gamma; \Delta'_1 \vdash (v \bar{m}_1)(P_2 \mid t \leftarrow_c v : C_1) \mathfrak{R} \Delta'_2 \vdash (v \bar{m}_2)(Q_2 \mid t \leftarrow_m v\sigma : C_1)$$

- (2) Whenever $\Gamma_1; \Delta_1 \vdash P_1 \xrightarrow{n!(\bar{v})} \Delta'_1 \vdash P_2$ then there exist Q_2, Δ'_2 , and σ_v such that $\Gamma_2; \Delta_2 \vdash Q_1 \xrightarrow{\bar{n}!(\bar{v})} \Delta'_2 \vdash Q_2$ where $v\sigma_v \succ_c \bar{v}$ and $\Gamma_1; \Delta'_1 \vdash P_2 \mathfrak{R} \Gamma_2; \Delta'_2 \vdash Q_2$,

- (3) Whenever $\Gamma_1; \Delta_1 \vdash P_1 \xrightarrow{\ell} \Delta'_1 \vdash P_2$, with ℓ not an output or input, then there exist Q_2 and Δ'_2 such that $\Gamma_2; \Delta_2 \vdash Q_1 \xrightarrow{\hat{\ell}} \Delta'_2 \vdash Q_2$ and $\Gamma_1; \Delta'_1 \vdash P_2 \mathfrak{R} \Gamma_2; \Delta'_2 \vdash Q_2$ and $\text{sub}(\ell) = n$ implies $\text{sub}(\hat{\ell}) = \bar{n}$.

- (4) The symmetric cases of 1, 2, and 3.

The largest such bisimulation is called *MST bisimilarity* (\approx^M).

We can now state our dynamic correctness result:

THEOREM 5 (OPERATIONAL CORRESPONDENCE). *Let P be a π process such that $\Gamma_1; \Delta_1 \vdash P_1$. We have*

$$\Gamma; \Delta \vdash P \approx^M \mathcal{H}^*(\Gamma); \mathcal{H}^*(\Delta) \vdash \mathcal{F}^*(P)$$

PROOF. By coinduction: we exhibit a binary relation \mathcal{S} that contains $(P, \mathcal{F}^*(P))$ and prove that it is an MST bisimulation. The proof that \mathcal{S} is an MST bisimilarity is given by Lem. C.7 and Lem. C.8 (see App. C.7 for details). \square

5 CONCLUDING REMARKS

Concluding Remarks. We showed a minimality result for π , a session-typed π -calculus. This result says that sequentiality in session types is a convenient but not indispensable feature. Following [1], we introduced minimal session types (MSTs) for π and defined two decompositions, which transform processes typable with standard session types into processes typable with MSTs. The first decomposition composes existing encodability results and the minimality result for HO; the second decomposition optimizes the first one by (i) removing redundant synchronizations and (ii) using the native support of recursion in π . For this optimized decomposition, we proved also an operational correspondence result. This way, our work shows that the minimality result is independent from the kind of communicated objects (names or abstractions).

Sequentiality is the key distinguishing feature in the specification of message-passing programs using session types. We remark that by our minimality results do not mean that sequentiality in session

types is redundant in *modeling and specifying* processes; rather, we claim that it is not an indispensable notion to *type-checking* them. Because we can type-check *session typed* processes using type systems that do not directly support sequencing in types, our decomposition defines a technique for implementing session types into languages whose type systems do not support sequentiality.

For the sake of space, we have not considered choice constructs (selection and branching). There is no fundamental obstacle in treating them, apart from a very minor caveat: the decomposition in [1] assumes typed processes in which every selection construct comes with a corresponding branching (see App. A.2 for an example).

All in all, besides settling a question left open in [1], our work deepens our understanding about session-based concurrency and the connection between the first-order and higher-order paradigms.

Related Work. We use the trios decomposition by Parrow [11], which he studied for an untyped π -calculus with replication; in contrast, π processes feature recursion. We stress that our goal is to clarify the role of sequentiality in session types by using processes with MSTs, which lack sequentiality. While Parrow's approach elegantly induces processes typable with MSTs, defining trios decompositions for π is just one path towards our goal.

Our work differs significantly with respect to [1]. The source language in [1] is HO, based on abstraction-passing, whereas here we focus on the name-passing calculus π . While in HO propagators carry abstractions, in our case propagators are binding and carry names. Also, names must be decomposed and propagating them requires care. Further novelties appear when decomposing processes with recursion, which require a dedicated collection of *recursive trios* (not supported in HO).

Prior works have related session types with *different* type systems [2–4]. Loosely related is the work by Dardha et al. [2]. They compile a session π -calculus down into a π -calculus with the linear type system of [7] extended with variant types. They represent sequentiality using a continuation-passing style: a session type is interpreted as a linear type carrying a pair consisting of the original payload type and a new linear channel type, to be used for ensuing interactions. The differences are also technical: the approach in [2] thus involves translations connecting *two* different π -calculi and *two* different type systems. In contrast, our approach based on MSTs justifies sequentiality using a single typed process framework.

ACKNOWLEDGMENTS

This work has been partially supported by the Dutch Research Council (NWO) under project No. 016.Vidi.189.046 (Unifying Correctness for Communicating Software).

We are grateful to the anonymous reviewers for their careful reading and constructive remarks.

REFERENCES

- [1] Alen Arslanagić, Jorge A. Pérez, and Erik Voogd. 2019. Minimal Session Types (Pearl). In *33rd European Conference on Object-Oriented Programming, ECOOP 2019, July 15–19, 2019, London, United Kingdom (LIPICs)*, Alastair F. Donaldson (Ed.), Vol. 134. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 23:1–23:28. <https://doi.org/10.4230/LIPICs.ECOOP.2019.23>
- [2] Ornela Dardha, Elena Giachino, and Davide Sangiorgi. 2012. Session types revisited. In *Proc. of PPDP 2012*. ACM, 139–150. <https://doi.org/10.1145/2370776.2370794>

- [3] Romain Demangeon and Kohei Honda. 2011. Full Abstraction in a Subtyped π -Calculus with Linear Types. In *Proc. of CONCUR 2011 (LNCS)*, Vol. 6901. Springer, 280–296. https://doi.org/10.1007/978-3-642-23217-6_19
- [4] Simon J. Gay, Nils Gesbert, and António Ravara. 2014. Session Types as Generic Process Types. In *Proceedings Combined 21st International Workshop on Expressiveness in Concurrency and 11th Workshop on Structural Operational Semantics, EXPRESS 2014, and 11th Workshop on Structural Operational Semantics, SOS 2014, Rome, Italy, 1st September 2014. (EPTCS)*, Johannes Borgström and Silvia Crafa (Eds.), Vol. 160. 94–110. <https://doi.org/10.4204/EPTCS.160.9>
- [5] Kohei Honda. 1993. Types for Dyadic Interaction. In *CONCUR'93 (LNCS)*, Eike Best (Ed.), Vol. 715. Springer-Verlag, 509–523.
- [6] Kohei Honda, Vasco T. Vasconcelos, and Makoto Kubo. 1998. Language Primitives and Type Disciplines for Structured Communication-based Programming. In *ESOP'98 (LNCS)*, Vol. 1381. Springer, 22–138.
- [7] Naoki Kobayashi, Benjamin C. Pierce, and David N. Turner. 1999. Linearity and the Pi-Calculus. *TOPLAS* 21, 5 (Sept. 1999), 914–947.
- [8] Dimitrios Kouzapas, Jorge A. Pérez, and Nobuko Yoshida. 2016. On the Relative Expressiveness of Higher-Order Session Processes. In *Programming Languages and Systems - 25th European Symposium on Programming, ESOP 2016, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2016, Eindhoven, The Netherlands, April 2-8, 2016, Proceedings (Lecture Notes in Computer Science)*, Peter Thiemann (Ed.), Vol. 9632. Springer, 446–475. https://doi.org/10.1007/978-3-662-49498-1_18
- [9] Dimitrios Kouzapas, Jorge A. Pérez, and Nobuko Yoshida. 2017. Characteristic bisimulation for higher-order session processes. *Acta Inf.* 54, 3 (2017), 271–341. <https://doi.org/10.1007/s00236-016-0289-7>
- [10] Dimitrios Kouzapas, Jorge A. Pérez, and Nobuko Yoshida. 2019. On the relative expressiveness of higher-order session processes. *Inf. Comput.* 268 (2019). <https://doi.org/10.1016/j.ic.2019.06.002>
- [11] Joachim Parrow. 2000. Trios in concert. In *Proof, Language, and Interaction, Essays in Honour of Robin Milner*, Gordon D. Plotkin, Colin Stirling, and Mads Tofte (Eds.), The MIT Press, 623–638. Online version, dated July 22, 1996, available at <http://user.it.uu.se/~joachim/trios.pdf>.
- [12] Davide Sangiorgi. 1992. *Expressing Mobility in Process Algebras: First-Order and Higher Order Paradigms*. Ph.D. Dissertation. University of Edinburgh.

CONTENTS

Abstract	1
1 Introduction	1
2 Preliminaries	2
2.1 $\text{HO}\pi$ (and its Sub-languages HO and π)	2
2.2 Session Types for $\text{HO}\pi$	2
2.3 Mutual Encodings between π and HO	3
2.4 Minimal Session Types for HO	4
3 Decompose by Composition	4
3.1 Key Idea	4
3.2 Formal Definition	5
3.3 Examples	6
3.4 Results	8
4 Optimizations	9
4.1 Motivation	9
4.2 Preliminaries	9
4.3 The Optimized Decomposition	10
4.3.1 The Breakdown Function	10
4.3.2 Handling P in $\mu X.P$	11
4.4 Examples	12
4.5 Measuring the Optimization	13
4.6 Results	13
4.6.1 Static Correctness	13
4.6.2 Dynamic Correctness	13
5 Concluding Remarks	14
Acknowledgments	14
References	14
Contents	16
A Additional Examples	17
A.1 First Decomposition: Core Fragment	17
A.2 First Decomposition: Labelled Choice	19
B Relevant notions from [10]	21
B.1 Session Types for $\text{HO}\pi$	21
B.2 Labelled Transition System for $\text{HO}\pi$	21
B.3 Encoding of $\text{HO}\pi$ into HO	22
C Proofs	23
C.1 Auxiliary Results	23
C.2 Decompose by Composition	23
C.3 Measuring the Optimization	24
C.4 Typability of the Optimized Breakdown Function	25
C.5 Proof of Thm. 4 (Minimality Result, Optimized)	40
C.6 MST Bisimilarity	40
C.7 Operational correspondence	43

A ADDITIONAL EXAMPLES

A.1 First Decomposition: Core Fragment

Example A.1 (Core Fragment). Let P be a π process which incorporates name-passing and implements channels u and \bar{w} with types $S = !\langle \bar{T} \rangle$; end and $T = ?(\text{Int}); !\langle \text{Bool} \rangle$; end, respectively:

$$P = (v u : S)(u!\langle w \rangle.\bar{w}?(t).\bar{w}!\langle \text{odd}(t) \rangle.\mathbf{0} \mid \bar{u}(x).x!\langle 5 \rangle.x?(b).\mathbf{0}) = (v u)(A \mid B)$$

The degree of P is $\lfloor P \rfloor = 25$. Then, the decomposition of P into a collection of first-order processes typed with minimal session types is:

$$\begin{aligned} \mathcal{F}(P) &= (v c_1, \dots, c_{25})(\bar{c}_1!\langle \rangle.\mathbf{0} \mid \mathcal{A}_\epsilon^1(P\sigma)) \\ &= (v c_1, \dots, c_{25})(\bar{c}_1!\langle \rangle.\mathbf{0} \mid \mathcal{A}_\epsilon^1((v u)(A \mid B)\sigma)) \\ &= (v c_1, \dots, c_{25})(\bar{c}_1!\langle \rangle.\mathbf{0} \mid (v u_1)\mathcal{A}_\epsilon^1((A \mid B)\sigma')), \end{aligned}$$

where $\sigma = \text{init}(\text{fn}(P))$, $\sigma' = \sigma \cdot \{u_1\bar{u}_1/u\bar{u}\}$

We have:

$$\mathcal{A}_\epsilon^1((A \mid B)\sigma') = c_1?().\bar{c}_2!\langle \rangle.\bar{c}_{13}!\langle \rangle \mid \mathcal{A}_\epsilon^2(A\sigma') \mid \mathcal{A}_\epsilon^{13}(B\sigma')$$

We use some abbreviations for subprocesses of A and B :

$$\begin{aligned} A' &= \bar{w}_1?(t).A'' & A'' &= \bar{w}_1!\langle \text{odd}(t) \rangle.\mathbf{0} \\ B' &= x_1!\langle 5 \rangle.B'' & B'' &= x_1?(b).\mathbf{0} \end{aligned}$$

The breakdown of A is:

$$\begin{aligned} \mathcal{A}_\epsilon^2(A) &= c_2?().(v a_1)(u_1!\langle a_1 \rangle.\bar{c}_5!\langle \rangle \mid \mathcal{A}_\epsilon^5(A') \mid a_1?(y_1).y_1?(z_1).\bar{c}_3!\langle z_1 \rangle \mid \\ &\quad c_3?(z_1).z_1?(x).\bar{c}_4!\langle x \rangle \mid c_4?(x).(v s)(x!\langle s \rangle.\bar{s}!\langle w_1, w_2 \rangle))) \\ \mathcal{A}_\epsilon^5(A') &= c_5?().\bar{w}_1?(y_2).\bar{c}_6!\langle y_2 \rangle \mid (v s_1)(c_6?(y_2).\bar{c}_7!\langle y_2 \rangle.\bar{c}_8!\langle \rangle \mid \\ &\quad c_7?(y_2).(v s')(y_2!\langle s' \rangle.\bar{s}'!\langle s_1 \rangle.\mathbf{0}) \mid c_8?().(v a_2)(\bar{s}_1!\langle a_2 \rangle.\bar{c}_{10}!\langle \rangle \mid c_{10}?().\mathbf{0} \mid \\ &\quad a_2?(y_3).y_3?(t_1).\bar{c}_9!\langle \rangle \mid \mathcal{A}_\epsilon^9(A'')))) \\ \mathcal{A}_\epsilon^9(A'') &= c_9?().(v a)(\bar{w}_2!\langle a \rangle.\bar{c}_{12}!\langle \rangle \mid c_{12}?().\mathbf{0} \mid \\ &\quad a?(y).y?(z_1).\bar{c}_{11}!\langle z_1 \rangle \mid c_{10}?(z_1).z_1?(x).\bar{c}_{11}!\langle x \rangle \mid c_{11}?(x).(v s)(x!\langle s \rangle.\bar{s}!\langle \text{odd}(t) \rangle))) \end{aligned}$$

The breakdown of B is:

$$\begin{aligned} \mathcal{A}_\epsilon^{13}(B) &= c_{13}?().\bar{u}_1?(y_4).\bar{c}_{14}!\langle y_4 \rangle \mid (v s_1)(c_{14}?(y).\bar{c}_{15}!\langle y \rangle.\bar{c}_{16}!\langle \rangle \mid \\ &\quad c_{15}?(y_4).(v s'')(y_4!\langle s'' \rangle.\bar{s}''!\langle s_1 \rangle.\mathbf{0}) \mid c_{16}?(().(v a_3)(s_1!\langle a_3 \rangle).\bar{c}_{21}!\langle \rangle \mid \\ &\quad c_{21}?().\mathbf{0} \mid a_3?(y_5).y_5?(x_1, x_2).\bar{c}_{17}!\langle \rangle \mid \mathcal{A}_\epsilon^{17}(B')))) \\ \mathcal{A}_\epsilon^{17}(B') &= c_{17}?(().(v a_4)(x_1!\langle a_4 \rangle).\bar{c}_{20}!\langle \rangle \mid \mathcal{A}_\epsilon^{20}(B'') \mid a_4?(y_6).y_6?(z_1).\bar{c}_{18}!\langle z_1 \rangle \mid \\ &\quad c_{18}?(z_1).z_1?(x).\bar{c}_{19}!\langle x \rangle \mid c_{19}?(x).(v s''')(x!\langle s''' \rangle.\bar{s}'''!\langle 5 \rangle))) \\ \mathcal{A}_\epsilon^{20}(B'') &= c_{20}?(().x_2?(y).\bar{c}_{21}!\langle y \rangle \mid \\ &\quad (v s_1)(c_{21}?(y).\bar{c}_{22}!\langle y \rangle.\bar{c}_{23}!\langle \rangle \mid c_{22}?(y).(v s)(y!\langle s \rangle.\bar{s}!\langle s_1 \rangle) \mid \\ &\quad c_{23}?(x).\bar{c}_{24}!\langle x \rangle.\bar{c}_{25}!\langle \rangle \mid c_{25}?().\mathbf{0} \mid \\ &\quad a?(y').y'?(b_1).\bar{c}_{24}!\langle \rangle \mid c_{24}?().\mathbf{0}))) \end{aligned}$$

Names \bar{w}_1 and \bar{w}_2 are typed with $M_1 = ?(\langle ?(\langle ?(\text{Int}); \text{end} \rangle); \text{end}; \text{end} \rangle); \text{end}$ and $M_2 = !(\langle ?(\langle ?(\langle ?(\text{Bool}); \text{end} \rangle); \text{end}; \text{end} \rangle); \text{end}; \text{end} \rangle); \text{end}$, respectively. Name u_1 is typed with $!\langle ?(\langle ?(\langle ?(\bar{M}_1, \bar{M}_2); \text{end} \rangle); \text{end}; \text{end} \rangle); \text{end}$. Now let us observe the reduction chain. The collection of processes synchronizes on c_1, c_2, c_{13} after three reductions.

$$\begin{aligned} \mathcal{F}(P) &\longrightarrow^3 (v \bar{c})(v a_1)(\bar{u}_1!\langle a_1 \rangle.\bar{c}_5!\langle \rangle \mid \mathcal{A}_\epsilon^5(A') \mid a_1?(y_1).y_1?(z_1).\bar{c}_3!\langle z_1 \rangle \mid c_3?(z_1).z_1?(x).\bar{c}_4!\langle x \rangle \mid \\ &\quad c_4?(x).(v s)(x!\langle s \rangle.\bar{s}!\langle w_1, w_2 \rangle)) \mid \bar{u}_1?(y_4).\bar{c}_{14}!\langle y_4 \rangle \mid (v s_1)(c_{14}?(y).\bar{c}_{15}!\langle y \rangle.\bar{c}_{16}!\langle \rangle \mid \\ &\quad c_{15}?(y_4).(v s'')(y_4!\langle s'' \rangle.\bar{s}''!\langle s_1 \rangle.\mathbf{0}) \mid c_{16}?(().(v a_3)(s_1!\langle a_3 \rangle).\bar{c}_{21}!\langle \rangle \mid \\ &\quad c_{21}?().\mathbf{0} \mid a_3?(y_5).y_5?(x_1, x_2).\bar{c}_{17}!\langle \rangle \mid \mathcal{A}_\epsilon^{17}(B')))) \\ &\text{where } \bar{c} = (c_3, \dots, c_{12}, c_{14}, \dots, c_{25}) \end{aligned}$$

Then, the broken down process A communicates with process B through channel u_1 by passing name a_1 (highlighted above). Here we notice that the original transmission of value w is not immediately mimicked on channel u , but it is delegated to some other channel through series of channel redirections starting with the transmission of name a_1 . Further, received name a_1 is locally propagated by c_{14} and c_{15} . This represents redundant communications on propagators induced by breaking down sequential prefixes produced by two encodings $\llbracket \cdot \rrbracket_g^1$ and $\llbracket \cdot \rrbracket^2$ (not present in the original process). Another synchronization occurs on c_{16} .

$$\begin{aligned} \mathcal{F}(P) \longrightarrow^7 & (v \tilde{c}_*)(v a_1)(\overline{c_5!}\langle \rangle \mid \mathcal{A}_e^5(A') \mid a_1?(y_1). y_1?(z_1). \overline{c_3!}\langle z_1 \rangle \mid c_3?(z_1). z_1?(x). \overline{c_4!}\langle x \rangle \mid \\ & c_4?(x).(v s)(x!\langle s \rangle. \overline{s!}\langle w_1, w_2 \rangle) \mid (v s_1)((v s'')(\overline{a_1!}\langle s'' \rangle. \overline{s''!}\langle s_1 \rangle. \mathbf{0} \mid (v a_3)(s_1!\langle a_3 \rangle. (\overline{c_{21}}!)\langle \rangle \mid \\ & c_{21}?(.) . \mathbf{0} \mid a_3?(y_5). y_5?(x_1, x_2). (\overline{c_{17}}!)\langle \rangle \mid \mathcal{A}_e^{17}(B')))) \\ & \text{where } \tilde{c}_* = (c_3, \dots, c_{12}, c_{17}, \dots, c_{25}) \end{aligned}$$

The next step involves communication on a_1 : session name s'' is passed and substitutes variable y_1 .

$$\begin{aligned} \mathcal{F}(P) \longrightarrow^8 & (v \tilde{c}_*)(v s'')(\overline{c_5!}\langle \rangle \mid \mathcal{A}_e^5(A') \mid s''?(z_1). \overline{c_3!}\langle z_1 \rangle \mid c_3?(z_1). z_1?(x). \overline{c_4!}\langle x \rangle \mid \\ & c_4?(x).(v s)(x!\langle s \rangle. \overline{s!}\langle w_1, w_2 \rangle) \mid (v s_1)(\overline{s''!}\langle s_1 \rangle. \mathbf{0} \mid (v a_3)(s_1!\langle a_3 \rangle. (\overline{c_{21}}!)\langle \rangle \mid c_{21}?(.) . \mathbf{0} \mid \\ & a_3?(y_5). y_5?(x_1, x_2). (\overline{c_{17}}!)\langle \rangle \mid \mathcal{A}_e^{17}(B')))) \end{aligned}$$

The process then synchronizes on channel s'' . After z_1 is replaced by s_1 , it is further sent to the next parallel process through the propagator c_3 .

$$\begin{aligned} \mathcal{F}(P) \longrightarrow^{10} & (v \tilde{c}_{**})(v s_1)(\overline{c_5!}\langle \rangle \mid \mathcal{A}_e^5(A') \mid s_1?(x). \overline{c_4!}\langle x \rangle \mid c_4?(x).(v s)(x!\langle s \rangle. \overline{s!}\langle w_1, w_2 \rangle) \mid \\ & (v a_3)(s_1!\langle a_3 \rangle. (\overline{c_{21}}!)\langle \rangle \mid c_{21}?(.) . \mathbf{0} \mid \\ & a_3?(y_5). y_5?(x_1, x_2). (\overline{c_{17}}!)\langle \rangle \mid \mathcal{A}_e^{17}(B')))) \\ & \text{where } \tilde{c}_{**} = (c_4, \dots, c_{12}, c_{17}, \dots, c_{25}) \end{aligned}$$

Communication on s_1 leads to variable x being substituted by name a_3 , which is then passed on c_4 to the next process. In addition, inaction is simulated by synchronization on c_{21} .

$$\begin{aligned} \mathcal{F}(P) \longrightarrow^{13} & (v \tilde{c}_\bullet)(v a_3)(\overline{c_5!}\langle \rangle \mid \mathcal{A}_e^5(A') \mid (v s)(a_3!\langle s \rangle. \overline{s!}\langle w_1, w_2 \rangle) \mid \\ & a_3?(y_5). y_5?(x_1, x_2). (\overline{c_{17}}!)\langle \rangle \mid \mathcal{A}_e^{17}(B')) \\ & \text{where } \tilde{c}_\bullet = (c_5, \dots, c_{12}, c_{17}, \dots, c_{25}) \end{aligned}$$

Now, the distribution of the decomposition of w from one process to another can finally be simulated by two reductions: first, a synchronization on a_3 sends the endpoint of session s , which replaces variable y_5 ; afterwards, the dual endpoint is used to send the names w_1, w_2 , substituting the variables x_1, x_2 .

$$\begin{aligned} \mathcal{F}(P) \longrightarrow^{14} & (v \tilde{c}_{\bullet\bullet})(v s)(\overline{c_5!}\langle \rangle \mid \mathcal{A}_e^5(A') \mid \\ & \overline{s!}\langle w_1, w_2 \rangle \mid s?(x_1, x_2). (\overline{c_{17}}!)\langle \rangle \mid \mathcal{A}_e^{17}(B')) \\ \mathcal{F}(P) \longrightarrow^{15} & (v \tilde{c}_{\bullet\bullet})(\overline{c_5!}\langle \rangle \mid \mathcal{A}_e^5(A') \mid \\ & \overline{c_{17}}!)\langle \rangle \mid \mathcal{A}_e^{17}(B')\{w_1 w_2/x_1 x_2\} = Q \\ & \text{where } \tilde{c}_{\bullet\bullet} = (c_5, \dots, c_{12}, c_{17}, \dots, c_{25}) \end{aligned}$$

Here, we remark that as prefix $s?(x_1, x_2)$ bounds variables x_1, x_2 in the breakdown of the continuation $(\mathcal{A}_e^{17}(B'))$. Thus, there is no need for propagators to pass contexts: propagators here only serve to enforce the ordering of actions. On the other hand, this rely on arbitrary process nesting which is induced by the final application of encoding $\llbracket \cdot \rrbracket^2$ in the composition. Thus the trio structure is lost.

Undoubtedly, the first action of the original first-order process has been simulated. We may notice that in Q names w_1, w_2 substitute x_1, x_2 and subsequent τ -action on w can be simulated on name w_1 . The following reductions follow the same pattern. Thus, the outcome of our decomposition function is a functionally equivalent process that is typed with minimal session types.

A.2 First Decomposition: Labelled Choice

We now illustrate how the approach in § 3 can be extended to account for labelled (deterministic) choice (selection and branching constructs). In $\text{HO}\pi$, these constructs are denoted

$$u \triangleleft l.P \quad u \triangleright \{l_i : P_i\}_{i \in I}$$

with reduction rule

$$n \triangleleft l_j.Q \mid \bar{n} \triangleright \{l_i : P_i\}_{i \in I} \longrightarrow Q \mid P_j \quad (j \in I)$$

Example A.2 (Labelled Choice). Let P and Q be π processes which incorporate branching and selection:

$$\begin{aligned} P &= u \triangleright \{l_1 : (\nu h)(h!\langle m \rangle.h?(t).\mathbf{0} \mid \bar{h}?(x).\bar{h}!\langle x \rangle.\mathbf{0}), l_2 : s!\langle m \rangle.\bar{s}?(x).\mathbf{0}\} \\ Q &= \bar{u} \triangleleft l_1.\mathbf{0} \end{aligned}$$

Let R denote the restricted parallel composition of P and Q : $R = (\nu u)(P \mid Q)$. The respective process degrees have the same values as before. The decomposition of R is:

$$\begin{aligned} \mathcal{F}(R) &= (\nu c_1, \dots, c_5)(\bar{c}_1!\langle \rangle.\mathbf{0} \mid \mathcal{A}_\epsilon^1((\nu u)(P \mid Q)\sigma)_g), \text{ where } \sigma = \{u_1\bar{u}_1/u\bar{u}\} \\ &= (\nu c_1, \dots, c_5)(\bar{c}_1!\langle \rangle.\mathbf{0} \mid (\nu u_1)(c_1?().\bar{c}_2!\langle \rangle.\bar{c}_3!\langle \rangle \mid \mathcal{A}_\epsilon^2(P)_g \mid \mathcal{A}_\epsilon^3(Q)_g)) \end{aligned}$$

We consider P and Q separately. We begin with the breakdown of Q .

$$\begin{aligned} \mathcal{A}_\epsilon^3(Q)_g &= c_3?().(\nu a_1)(\bar{c}_4!\langle a_1 \rangle.((\nu u_2)(c_4?(y).(\nu s)(y!\langle s \rangle.\bar{s}!\langle \bar{u}_2 \rangle)) \mid c_5?().\mathbf{0} \mid \\ &\quad a_1?(y').y'(y_1).u_1 \triangleleft l_1.u_1?(z).\bar{c}_5!\langle \rangle.(\nu s')(z!\langle s' \rangle.\bar{s}'!\langle y_1 \rangle))) \end{aligned}$$

The breakdown of process P is:

$$\begin{aligned} \mathcal{A}_\epsilon^2(P)_g &= c_2?().u_1 \triangleright \{l_1 : (\nu a_2)u_1!\langle a_2 \rangle.a_2?(y'').y''?(y_1).(\nu c_3, \dots, c_{21})(\bar{c}_3!\langle \rangle \mid \mathcal{A}_\epsilon^3(P_1\{y_1/u_1\})_g), \\ &\quad l_2 : (\nu a_2)u_1!\langle a_2 \rangle.a_2?(y'').y''?(y_1).(\nu c_3, \dots, c_{11})(\bar{c}_3!\langle \rangle \mid \mathcal{A}_\epsilon^3(P_2\{y_1/u_1\})_g)\} \end{aligned}$$

As in the previous approach of this example, we are interested in the breakdown of P_1 , because it belongs to the selected branch.

$$\begin{aligned} \mathcal{A}_\epsilon^3(P_1\{y_1/u_1\})_g &= (\nu h_1, h_2)\mathcal{A}_\epsilon^3(h_1!\langle m_1 \rangle.h_1?(t).\mathbf{0} \mid \bar{h}_1?(x).\bar{h}_1!\langle x \rangle.\mathbf{0})_g \\ &= (\nu h_1, h_2)(c_3?().\bar{c}_4!\langle \rangle.\bar{c}_{13}!\langle \rangle \mid \mathcal{A}_\epsilon^4(h_1!\langle m_1 \rangle.h_1?(t).\mathbf{0})_g \mid \mathcal{A}_\epsilon^{13}(\bar{h}_1?(x).\bar{h}_1!\langle x \rangle.\mathbf{0})_g) \end{aligned}$$

$$\begin{aligned} \mathcal{A}_\epsilon^4(h_1!\langle m_1 \rangle.g.h_1?(t).\mathbf{0}) &= c_4?().(\nu a_3)(h_1!\langle a_3 \rangle.(\bar{c}_7!\langle \rangle \mid \mathcal{A}_\epsilon^7(h_2?(t).\mathbf{0})_g \mid a_3?(y_2).y_2?(z_1).\bar{c}_5!\langle z_1 \rangle \mid \\ &\quad c_5?(z_1).z_1?(x).\bar{c}_6!\langle x \rangle \mid c_6?(x).(\nu s_1)(x!\langle s_1 \rangle.\bar{s}_1!\langle m_1 \rangle.\mathbf{0})) \end{aligned}$$

$$\begin{aligned} \mathcal{A}_\epsilon^7(h_2?(t).\mathbf{0})_g &= c_7?().h_2?(y_3).\bar{c}_8!\langle y_3 \rangle \mid (\nu s_2)(c_8?(y_3).\bar{c}_9!\langle y_3 \rangle.\bar{c}_{10}!\langle \rangle \mid \\ &\quad c_9?(y_3).(\nu s_3)(y_3!\langle s_3 \rangle.\bar{s}_3!\langle s_2 \rangle.\mathbf{0}) \mid c_{10}?(().(\nu a_4)(\bar{s}_2!\langle a_4 \rangle.(\bar{c}_{12}!\langle \rangle \mid \\ &\quad c_{12}?(().\mathbf{0} \mid a_4?(y_4).y_4?(t_1).\bar{c}_{11}!\langle \rangle \mid c_{11}?(().\mathbf{0}))) \end{aligned}$$

$$\begin{aligned} \mathcal{A}_\epsilon^{13}(\bar{h}_1?(x).\bar{h}_1!\langle x \rangle.\mathbf{0})_g &= c_{13}?(().\bar{h}_1?(y_5).\bar{c}_{14}!\langle y_5 \rangle \mid (\nu s_4)(c_{14}(y_5).\bar{c}_{15}!\langle y_5 \rangle.\bar{c}_{16}!\langle \rangle \mid \\ &\quad c_{15}(y_5).(\nu s_5)(y_5!\langle s_5 \rangle.\bar{s}_5!\langle s_4 \rangle.\mathbf{0}) \mid c_{16}?(().(\nu a_5)(\bar{s}_4!\langle a_5 \rangle.(\bar{c}_{21}!\langle \rangle \mid \\ &\quad c_{21}?(().\mathbf{0} \mid a_5?(y_6).y_6?(x_1).\bar{c}_{17}!\langle x_1 \rangle \mid \mathcal{A}_{x_1}^{17}(\bar{h}_2!\langle x_1 \rangle.\mathbf{0})_g))) \end{aligned}$$

$$\begin{aligned} \mathcal{A}_{x_1}^{17}(\bar{h}_2!\langle x_1 \rangle.\mathbf{0})_g &= c_{17}?(x_1).(\nu a_6)(\bar{h}_2!\langle a_6 \rangle.(\bar{c}_{20}!\langle x_1 \rangle \mid c_{20}?(().\mathbf{0} \mid a_6?(y_7).y_7?(z_2).\bar{c}_{18}!\langle z_2 \rangle \mid \\ &\quad c_{18}?(z_2).z_2?(x').\bar{c}_{19}!\langle x' \rangle \mid c_{19}?(x').(\nu s_6)(x'!\langle s_6 \rangle.\bar{s}_6!\langle x_1 \rangle.\mathbf{0})) \end{aligned}$$

In order to verify that the decomposition of R simulates the behaviour of the initial process, we refer to its reduction chain. The first transitions involve synchronization on c_1, c_2, c_3 and c_4 . The trigger name a_2 is passed through propagator c_4 , such that it replaces variable y .

$$\begin{aligned} \mathcal{F}(R) &\longrightarrow^4 (\nu c_5)(\nu u_1)(u_1 \triangleright \{l_1 : (\nu a_2) u_1!\langle a_2 \rangle.a_2?(y'').y''?(y_1).(\nu c_3, \dots, c_{21})(\bar{c}_3!\langle \rangle \mid \\ &\quad \mathcal{A}_\epsilon^3(P_1\{y_1/u_1\})_g, l_2 : (\nu a_2) u_1!\langle a_2 \rangle.a_2?(y'').y''?(y_1).(\nu c_3, \dots, c_{11})(\bar{c}_3!\langle \rangle \mid \\ &\quad \mathcal{A}_\epsilon^3(P_2\{y_1/u_1\})_g)\} \mid (\nu a_1)((\nu u_2)((\nu s)(a_1!\langle s \rangle.\bar{s}!\langle \bar{u}_2 \rangle)) \mid c_5?().\mathbf{0} \mid \\ &\quad a_1?(y').y'(y_1).u_1 \triangleleft l_1.u_1?(z).\bar{c}_5!\langle \rangle.(\nu s')(z!\langle s' \rangle.\bar{s}'!\langle y_1 \rangle))) \end{aligned}$$

The next step involves synchronization on a_1 , which leads to the variable y' being substituted by session name s .

$$\begin{aligned} \mathcal{F}(R) \longrightarrow^5 & (v c_5)(v u_1)(u_1 \triangleright \{l_1 : (v a_2) u_1! \langle a_2 \rangle . a_2?(y'') . y''?(y_1) . (v c_3, \dots, c_{21})(\overline{c_3}! \langle \rangle \mid \\ & \mathcal{A}_\epsilon^3(P_1\{y_1/u_1\})_g, l_2 : (v a_2) u_1! \langle a_2 \rangle . a_2?(y'') . y''?(y_1) . (v c_3, \dots, c_{11})(\overline{c_3}! \langle \rangle \mid \\ & \mathcal{A}_\epsilon^3(P_2\{y_1/u_1\})_g\} \mid (v s)((v u_2)(\overline{s}! \langle \overline{u_2} \rangle) \mid c_5?(). \mathbf{0} \mid \\ & s?(y_1) . u_1 \triangleleft l_1 . u_1?(z) . \overline{c_5}! \langle \rangle . (v s')(z! \langle s' \rangle . \overline{s}'! \langle y_1 \rangle))) \end{aligned}$$

Now, session name s and its dual endpoint can communicate - channel name $\overline{u_2}$ is sent, replacing y_1 .

$$\begin{aligned} \mathcal{F}(R) \longrightarrow^6 & (v c_5)(v u_1)(u_1 \triangleright \{l_1 : (v a_2) u_1! \langle a_2 \rangle . a_2?(y'') . y''?(y_1) . (v c_3, \dots, c_{21})(\overline{c_3}! \langle \rangle \mid \\ & \mathcal{A}_\epsilon^3(P_1\{y_1/u_1\})_g, l_2 : (v a_2) u_1! \langle a_2 \rangle . a_2?(y'') . y''?(y_1) . (v c_3, \dots, c_{11})(\overline{c_3}! \langle \rangle \mid \\ & \mathcal{A}_\epsilon^3(P_2\{y_1/u_1\})_g\} \mid (v u_2)(c_5?(). \mathbf{0} \mid u_1 \triangleleft l_1 . u_1?(z) . \overline{c_5}! \langle \rangle . (v s')(z! \langle s' \rangle . \overline{s}'! \langle \overline{u_2} \rangle))) \end{aligned}$$

At this point, the action of choosing the first branch can be simulated by selecting label l_1 on channel name u_1 .

$$\begin{aligned} \mathcal{F}(R) \longrightarrow^7 & (v c_5)(v u_1)((v a_2) u_1! \langle a_2 \rangle . a_2?(y'') . y''?(y_1) . (v c_3, \dots, c_{21})(\overline{c_3}! \langle \rangle \mid \\ & \mathcal{A}_\epsilon^3(P_1\{y_1/u_1\})_g \mid (v u_2)(c_5?(). \mathbf{0} \mid u_1?(z) . \overline{c_5}! \langle \rangle . (v s')(z! \langle s' \rangle . \overline{s}'! \langle \overline{u_2} \rangle))) \end{aligned}$$

The following two reductions consist of synchronizations on u_1 and c_5 . As a result, variable z is replaced by the trigger name a_2 .

$$\begin{aligned} \mathcal{F}(R) \longrightarrow^9 & (v a_2)(a_2?(y'') . y''?(y_1) . (v c_3, \dots, c_{21})(\overline{c_3}! \langle \rangle \mid \\ & \mathcal{A}_\epsilon^3(P_1\{y_1/u_1\})_g \mid (v u_2)((v s')(a_2! \langle s' \rangle . \overline{s}'! \langle \overline{u_2} \rangle))) \end{aligned}$$

Communication on a_2 allows for synchronization on s' . Channel a_2 receives the session name s' , which further receives the dual endpoint $\overline{u_2}$.

$$\begin{aligned} \mathcal{F}(R) \longrightarrow^{11} & (v c_3, \dots, c_{21})(v u_2)(\overline{c_3}! \langle \rangle \mid \mathcal{A}_\epsilon^3(P_1\{\overline{u_2}/u_1\})_g) \\ & = (v c_3, \dots, c_{21})(v u_2)(\overline{c_3}! \langle \rangle \mid (v h_1, h_2)(c_3?(). \overline{c_4}! \langle \rangle . \overline{c_{13}}! \langle \rangle \mid \\ & \mathcal{A}_\epsilon^4(h_1! \langle m_1 \rangle . h_1?(t) . \mathbf{0})_g \mid \mathcal{A}_\epsilon^{13}(\overline{h_1}?(x) . \overline{h_1}! \langle x \rangle . \mathbf{0})_g)) \end{aligned}$$

Remarkably, the obtained result reflects the broken down version of process P_1 . Following three more reductions, in which c_3 , c_4 and c_{13} synchronize with their respective duals, the breakdown of each parallel sub-process will be activated. We refrain from presenting the remaining reduction steps, as they represent a standard case of name-passing. Clearly, the decomposed process $\mathcal{F}(R)$ behaves in a similar manner to the initial process R .

B RELEVANT NOTIONS FROM [10]

B.1 Session Types for $\text{HO}\pi$

$\frac{}{\Gamma; \emptyset; \{u : S\} \vdash u \triangleright S} \text{ (SESS)}$	$\frac{}{\Gamma, u : U; \emptyset; \emptyset \vdash u \triangleright U} \text{ (SH)}$	
$\frac{}{\Gamma; \{x : C \multimap \diamond\}; \emptyset \vdash x \triangleright C \multimap \diamond} \text{ (LVAR)}$	$\frac{}{\Gamma, X : \Delta; \emptyset; \Delta \vdash X \triangleright \diamond} \text{ (RVAR)}$	
$\frac{\Gamma; \Lambda; \Delta_1 \vdash P \triangleright \diamond \quad \Gamma; \emptyset; \Delta_2 \vdash x \triangleright C}{\Gamma \setminus x; \Lambda; \Delta_1 \setminus \Delta_2 \vdash \lambda x. P \triangleright C \multimap \diamond} \text{ (ABS)}$	$\frac{\Gamma; \Lambda; \Delta_1 \vdash V \triangleright C \rightsquigarrow \diamond \quad \rightsquigarrow \in \{-\circ, \rightarrow\} \quad \Gamma; \emptyset; \Delta_2 \vdash u \triangleright C}{\Gamma; \Lambda; \Delta_1, \Delta_2 \vdash V u \triangleright \diamond} \text{ (APP)}$	
$\frac{\Gamma; \emptyset; \emptyset \vdash V \triangleright C \multimap \diamond}{\Gamma; \emptyset; \emptyset \vdash V \triangleright C \multimap \diamond} \text{ (PROM)}$	$\frac{\Gamma; \Lambda, x : C \multimap \diamond; \Delta \vdash P \triangleright \diamond}{\Gamma, x : C \multimap \diamond; \Lambda; \Delta \vdash P \triangleright \diamond} \text{ (EPROM)}$	$\frac{\Gamma; \Lambda; \Delta \vdash P \triangleright T \quad u \notin \text{dom}(\Gamma, \Lambda, \Delta)}{\Gamma; \Lambda; \Delta, u : \text{end} \vdash P \triangleright \diamond} \text{ (END)}$
$\frac{\Gamma, X : \Delta; \emptyset; \Delta \vdash P \triangleright \diamond}{\Gamma; \emptyset; \Delta \vdash \mu X. P \triangleright \diamond} \text{ (REC)}$	$\frac{\Gamma; \Lambda_i; \Delta_i \vdash P_i \triangleright \diamond \quad i = 1, 2}{\Gamma; \Lambda_1, \Lambda_2; \Delta_1, \Delta_2 \vdash P_1 \mid P_2 \triangleright \diamond} \text{ (PAR)}$	$\frac{}{\Gamma; \emptyset; \emptyset \vdash \mathbf{0} \triangleright \diamond} \text{ (NIL)}$
$\frac{u : S \in \Delta_1, \Delta_2 \quad \Gamma; \Lambda_1; \Delta_1 \vdash P \triangleright \diamond \quad \Gamma; \Lambda_2; \Delta_2 \vdash V \triangleright U}{\Gamma; \Lambda_1, \Lambda_2; ((\Delta_1, \Delta_2) \setminus u : S), u : \langle U \rangle; S \vdash u! \langle V \rangle. P \triangleright \diamond} \text{ (SEND)}$		
$\frac{\Gamma; \Lambda; \Delta_1 \vdash P \triangleright \diamond \quad \Gamma; \emptyset; \emptyset \vdash u \triangleright \langle \mathcal{U} \rangle \quad \Gamma; \emptyset; \Delta_2 \vdash V \triangleright \mathcal{U} \quad \mathcal{U} \in \{S, L\}}{\Gamma; \Lambda; \Delta_1, \Delta_2 \vdash u! \langle V \rangle. P \triangleright \diamond} \text{ (REQ)}$		
$\frac{\Gamma; \Lambda_1; \Delta_1, u : S \vdash P \triangleright \diamond \quad \Gamma; \Lambda_2; \Delta_2 \vdash x \triangleright U}{\Gamma \setminus x; \Lambda_1 \setminus \Lambda_2; \Delta_1 \setminus \Delta_2, u : ? \langle U \rangle; S \vdash u?(x). P \triangleright \diamond} \text{ (RCV)}$	$\frac{\Gamma; \Lambda_1; \Delta_1 \vdash P \triangleright \diamond \quad \Gamma; \emptyset; \emptyset \vdash u \triangleright \langle \mathcal{U} \rangle}{\Gamma; \Lambda_2; \Delta_2 \vdash x \triangleright \mathcal{U} \quad \mathcal{U} \in \{S, L\}} \text{ (ACC)}$	
$\frac{\Gamma \setminus x; \Lambda_1 \setminus \Lambda_2; \Delta_1 \setminus \Delta_2 \vdash u?(x). P \triangleright \diamond}{\Gamma \setminus x; \Lambda_1 \setminus \Lambda_2; \Delta_1 \setminus \Delta_2 \vdash u?(x). P \triangleright \diamond} \text{ (BRA)}$		
$\frac{\forall i \in I \quad \Gamma; \Lambda; \Delta, u : S_i \vdash P_i \triangleright \diamond}{\Gamma; \Lambda; \Delta, u : \& \{l_i : S_i\}_{i \in I} \vdash u \triangleright \{l_i : P_i\}_{i \in I} \triangleright \diamond} \text{ (BRA)}$		$\frac{\Gamma; \Lambda; \Delta, u : S_j \vdash P \triangleright \diamond \quad j \in I}{\Gamma; \Lambda; \Delta, u : \oplus \{l_i : S_i\}_{i \in I} \vdash u \triangleleft l_j. P \triangleright \diamond} \text{ (SEL)}$
$\frac{\Gamma; \Lambda; \Delta, s : S_1, \bar{s} : S_2 \vdash P \triangleright \diamond \quad S_1 \text{ dual } S_2}{\Gamma; \Lambda; \Delta \vdash (vs)P \triangleright \diamond} \text{ (RES)}$		$\frac{\Gamma, a : \langle S \rangle; \Lambda; \Delta \vdash P \triangleright \diamond}{\Gamma; \Lambda; \Delta \vdash (va)P \triangleright \diamond} \text{ (RES)}$

Figure 13: Typing Rules for $\text{HO}\pi$ (including selection and branching constructs).

Remark 2. We derive polyadic rules for typing π as an expected extension of π typing rules in Fig. 14.

B.2 Labelled Transition System for $\text{HO}\pi$

The typed LTS combines the LTSs in Fig. 15 and Fig. 16.

Definition B.1 (Typed Transition System). A typed transition relation is a typed relation $\Gamma; \Delta_1 \vdash P_1 \xrightarrow{\ell} \Delta_2 \vdash P_2$ where:

- (1) $P_1 \xrightarrow{\ell} P_2$ and
- (2) $(\Gamma, \emptyset, \Delta_1) \xrightarrow{\ell} (\Gamma, \emptyset, \Delta_2)$ with $\Gamma; \emptyset; \Delta_i \vdash P_i \triangleright \diamond$ ($i = 1, 2$).

We write \Longrightarrow for the reflexive and transitive closure of $\xrightarrow{\ell}$, \Longrightarrow^{ℓ} for the transitions $\Longrightarrow \xrightarrow{\ell} \Longrightarrow$, and $\xRightarrow{\ell}$ for $\xrightarrow{\ell}$ if $\ell \neq \tau$ otherwise \Longrightarrow .

(POLYVAR) $\Gamma, \tilde{x} : \tilde{M}_x; \tilde{y} : \tilde{M}_y \vdash \tilde{x}\tilde{y} \triangleright \tilde{M}_x \tilde{M}_y$	
(POLYSEND) $\frac{\Gamma; \Delta_1, u : S \vdash P \triangleright \diamond \quad \Gamma; \Delta_2 \vdash \tilde{x} \triangleright \tilde{C}}{\Gamma; \Delta_1, \Delta_2, u : !(\tilde{C}); S \vdash u!(\tilde{x}).P \triangleright \diamond}$	(POLYRCV) $\frac{\Gamma; \Delta_1, u : S \vdash P \triangleright \diamond \quad \Gamma; \Delta_2 \vdash \tilde{x} \triangleright \tilde{C}}{\Gamma \setminus \tilde{x}; \Delta_1 \setminus \Delta_2, u : ?(\tilde{C}); \tilde{S} \vdash u?(\tilde{x}).P \triangleright \diamond}$
(POLYRESS) $\frac{\Gamma; \Delta, \tilde{s} : \tilde{S}_1, \tilde{s} : \tilde{S}_2 \vdash P \triangleright \diamond \quad S_1 \text{ dual } S_2}{\Gamma; \Delta \vdash (v \tilde{s})P \triangleright \diamond}$	(POLYRES) $\frac{\Gamma; \Delta, \tilde{a} : \tilde{C} \vdash P \triangleright \diamond}{\Gamma; \Delta \vdash (v \tilde{a})P \triangleright \diamond}$
(REQ) $\frac{\Gamma; \emptyset \vdash u \triangleright \langle \tilde{S} \rangle \quad \Gamma; \Delta_1 \vdash P \triangleright \diamond \quad \Gamma; \Delta_2 \vdash \tilde{x} \triangleright \tilde{S}}{\Gamma; \Delta_1, \Delta_2 \vdash u!(\tilde{x}).P \triangleright \diamond}$	(ACC) $\frac{\Gamma; \emptyset \vdash u \triangleright \langle \tilde{S} \rangle \quad \Gamma; \Delta_1 \vdash P \triangleright \diamond \quad \Gamma; \Delta_2 \vdash \tilde{x} \triangleright \tilde{S}}{\Gamma \setminus \tilde{x}; \Delta_1 \setminus \Delta_2 \vdash u?(\tilde{x}).P \triangleright \diamond}$

 Figure 14: Polyadic typing rules for π .

$\langle \text{APP} \rangle$ $\frac{}{(\lambda x.P)V \xrightarrow{\tau} P\{V/x\}}$	$\langle \text{SND} \rangle$ $\frac{}{n!\langle V \rangle.P \xrightarrow{n!\langle V \rangle} P}$	$\langle \text{RV} \rangle$ $\frac{}{n?(x).P \xrightarrow{n?\langle V \rangle} P\{V/x\}}$	$\langle \text{SEL} \rangle$ $\frac{}{s \triangleleft l.P \xrightarrow{s \oplus l} P}$	$\langle \text{BRA} \rangle$ $\frac{j \in I}{s \triangleright \{l_i : P_i\}_{i \in I} \xrightarrow{s \& l_j} P_j}$
$\langle \text{ALPHA} \rangle$ $\frac{P \equiv_{\alpha} Q \quad Q \xrightarrow{\ell} P'}{P \xrightarrow{\ell} P'}$	$\langle \text{RES} \rangle$ $\frac{P \xrightarrow{\ell} P' \quad n \notin \text{fn}(\ell)}{(v n)P \xrightarrow{\ell} (v n)P'}$	$\langle \text{NEW} \rangle$ $\frac{P \xrightarrow{(v \tilde{m})n!\langle V \rangle} P' \quad m_1 \in \text{fn}(V)}{(v m_1)P \xrightarrow{(v m_1, \tilde{m})n!\langle V \rangle} P'}$	$\langle \text{PARL} \rangle$ $\frac{P \xrightarrow{\ell} P' \quad \text{bn}(\ell) \cap \text{fn}(Q) = \emptyset}{P \mid Q \xrightarrow{\ell} P' \mid Q}$	
$\langle \text{TAU} \rangle$ $\frac{P \xrightarrow{\ell_1} P' \quad Q \xrightarrow{\ell_2} Q' \quad \ell_1 \asymp \ell_2}{P \mid Q \xrightarrow{\tau} (v \text{bn}(\ell_1) \cup \text{bn}(\ell_2))(P' \mid Q')}$			$\langle \text{REC} \rangle$ $\frac{P\{\mu X.P/X\} \xrightarrow{\ell} P'}{\mu X.P \xrightarrow{\ell} P'}$	

 Figure 15: The Untyped LTS for HO π processes. We omit Rule $\langle \text{PARR} \rangle$.

In this section we recall relevant notions from [10].

B.3 Encoding of HO π into HO

Definition B.2 (Auxiliary Mappings). We define mappings $\|\cdot\|$ and $\llbracket \cdot \rrbracket_{\sigma}$ as follows:

- $\|\cdot\| : 2^N \rightarrow \mathcal{V}^{\omega}$ is a map of sequences of lexicographically ordered names to sequences of variables, defined inductively as:

$$\begin{aligned} \|\epsilon\| &= \epsilon \\ \|n, \tilde{m}\| &= x_n, \|\tilde{m}\| \quad (x \text{ fresh}) \end{aligned}$$

- Given a set of session names and variables σ , the map $\llbracket \cdot \rrbracket_{\sigma} : \text{HO} \rightarrow \text{HO}$ is as in Fig. 17.

$$\begin{array}{c}
\text{[SRV]} \quad \frac{\bar{s} \notin \text{dom}(\Delta) \quad \Gamma; \Lambda'; \Delta' \vdash V \triangleright U}{(\Gamma; \Lambda; \Delta, s : ?\langle U \rangle; S) \xrightarrow{s? \langle V \rangle} (\Gamma; \Lambda, \Lambda'; \Delta, \Delta', s : S)} \\
\text{[SHRV]} \quad \frac{\Gamma; \emptyset; \emptyset \vdash a \triangleright \langle U \rangle \quad \Gamma; \Lambda'; \Delta' \vdash V \triangleright U}{(\Gamma; \Lambda; \Delta) \xrightarrow{a? \langle V \rangle} (\Gamma; \Lambda, \Lambda'; \Delta, \Delta')} \\
\text{[SSND]} \quad \frac{\Gamma, \Gamma'; \Lambda'; \Delta' \vdash V \triangleright U \quad \Gamma'; \emptyset; \Delta_j \vdash m_j \triangleright U_j \quad \bar{s} \notin \text{dom}(\Delta) \quad \Delta' \setminus (\cup_j \Delta_j) \subseteq \Delta \quad \Gamma'; \emptyset; \Delta'_j \vdash \bar{m}_j \triangleright U'_j \quad \Lambda' \subseteq \Lambda}{(\Gamma; \Lambda; \Delta, s : !\langle U \rangle; S) \xrightarrow{(\nu \bar{m})s! \langle V \rangle} (\Gamma, \Gamma'; \Lambda \setminus \Lambda'; (\Delta, s : S, \cup_j \Delta'_j) \setminus \Delta')} \\
\text{[SHSND]} \quad \frac{\Gamma, \Gamma'; \Lambda'; \Delta' \vdash V \triangleright U \quad \Gamma'; \emptyset; \Delta_j \vdash m_j \triangleright U_j \quad \Gamma; \emptyset; \emptyset \vdash a \triangleright \langle U \rangle \quad \Delta' \setminus (\cup_j \Delta_j) \subseteq \Delta \quad \Gamma'; \emptyset; \Delta'_j \vdash \bar{m}_j \triangleright U'_j \quad \Lambda' \subseteq \Lambda}{(\Gamma; \Lambda; \Delta) \xrightarrow{(\nu \bar{m})a! \langle V \rangle} (\Gamma, \Gamma'; \Lambda \setminus \Lambda'; (\Delta, \cup_j \Delta'_j) \setminus \Delta')} \\
\text{[SEL]} \quad \frac{\bar{s} \notin \text{dom}(\Delta) \quad j \in I}{(\Gamma; \Lambda; \Delta, s : \oplus \{l_i : S_i\}_{i \in I}) \xrightarrow{s \oplus l_j} (\Gamma; \Lambda; \Delta, s : S_j)} \\
\text{[BRA]} \quad \frac{\bar{s} \notin \text{dom}(\Delta) \quad j \in I}{(\Gamma; \Lambda; \Delta, s : \& \{l_i : T_i\}_{i \in I}) \xrightarrow{s \& l_j} (\Gamma; \Lambda; \Delta, s : S_j)} \\
\text{[TAU]} \quad \frac{\Delta_1 \longrightarrow \Delta_2 \vee \Delta_1 = \Delta_2}{(\Gamma; \Lambda; \Delta_1) \xrightarrow{\tau} (\Gamma; \Lambda; \Delta_2)}
\end{array}$$

Figure 16: Labelled Transition System for Typed Environments.

$$\begin{array}{l}
\llbracket w! \langle \lambda x. Q \rangle. P \rrbracket_\sigma \stackrel{\text{def}}{=} u! \langle \lambda x. \llbracket Q \rrbracket_{\sigma, x} \cdot \llbracket P \rrbracket_\sigma \\
\llbracket w?(x). P \rrbracket_\sigma \stackrel{\text{def}}{=} u?(x). \llbracket P \rrbracket_\sigma \\
\llbracket (\nu n)P \rrbracket_\sigma \stackrel{\text{def}}{=} (\nu n) \llbracket P \rrbracket_{\sigma, n} \\
\llbracket P \mid Q \rrbracket_\sigma \stackrel{\text{def}}{=} \llbracket P \rrbracket_\sigma \mid \llbracket Q \rrbracket_\sigma \\
\llbracket 0 \rrbracket_\sigma \stackrel{\text{def}}{=} 0 \\
\llbracket w \triangleright \{l_i : P_i\}_{i \in I} \rrbracket_\sigma \stackrel{\text{def}}{=} u \triangleright \{l_i : \llbracket P_i \rrbracket_\sigma\}_{i \in I} \\
\llbracket w \triangleleft l. P \rrbracket_\sigma \stackrel{\text{def}}{=} u \triangleleft l. \llbracket P \rrbracket_\sigma \\
\llbracket (\lambda x. Q) w \rrbracket_\sigma \stackrel{\text{def}}{=} (\lambda x. \llbracket Q \rrbracket_{\sigma, x}) u \\
\llbracket x w \rrbracket_\sigma \stackrel{\text{def}}{=} x u
\end{array}$$

In all cases: $u = \begin{cases} x_n & \text{if } w \text{ is a name } n \text{ and } n \notin \sigma \text{ (} x \text{ fresh)} \\ w & \text{otherwise: } w \text{ is a variable or a name } n \text{ and } n \in \sigma \end{cases}$

Figure 17: Auxiliary mapping used to encode HO π into HO (Def. B.2).

C PROOFS

C.1 Auxiliary Results

Lemma C.1 (Substitution Lemma). $\Gamma; \Delta, x : S \vdash P \triangleright \diamond$ and $u \notin \text{dom}(\Gamma, \Delta)$ implies $\Gamma; \Delta, u : S \vdash P\{u/x\} \triangleright \diamond$.

The following property follows immediately from Def. 4.1 and Def. 4.4:

Lemma C.2 (Typing Broken-down Variables). If $\Gamma; \Delta \vdash z_i \triangleright C$ then $\mathcal{G}(\Gamma); \mathcal{G}(\Delta) \vdash \tilde{z} \triangleright \mathcal{G}(C)$ where $\tilde{z} = (z_1, \dots, z_{i+|\mathcal{G}(C)|-1})$.

Lemma C.3 (Shared environment weakening). If $\Gamma; \Delta \vdash P \triangleright \diamond$ then $\Gamma, u : \langle C \rangle; \Delta \vdash P \triangleright \diamond$.

Lemma C.4 (Shared environment strengthening). If $\Gamma; \Delta \vdash P \triangleright \diamond$ and $u \notin \text{fn}(P)$ then $\Gamma \setminus u; \Delta \vdash P \triangleright \diamond$.

Lemma C.5. Let \tilde{r} be tuple of channel names and S a recursive session type. If $\tilde{r} : \mathcal{R}^*(!\langle C \rangle; S)$ and $k = f(!\langle C \rangle; S)$ then $r_k : \mu t. !\langle \mathcal{G}(C) \rangle; t$.

C.2 Decompose by Composition

THEOREM 1 (TYPABILITY OF BREAKDOWN). Let P be an initialized π process. If $\Gamma; \Delta, \Delta_\mu \vdash P \triangleright \diamond$, then $\mathcal{H}(\Gamma'), \Phi'; \mathcal{H}(\Delta), \Theta' \vdash \mathcal{A}_e^k(P)_g \triangleright \diamond$, where $k > 0; \tilde{r} = \text{dom}(\Delta_\mu); \Phi' = \prod_{r \in \tilde{r}} c^r : \langle \langle ?(\mathcal{R}^{*}(\Delta_\mu(r))); \text{end} \rangle \rangle$; and $\text{balanced}(\Theta')$ with

$$\text{dom}(\Theta') = \{c_k, c_{k+1}, \dots, c_{k+[P]-1}\} \cup \{\overline{c_{k+1}}, \dots, \overline{c_{k+[P]-1}}\}$$

such that $\Theta'(c_k) = ?(\cdot)$; end.

PROOF.

$$\begin{aligned}
 \Gamma; \Delta, \Delta_\mu \vdash P \triangleright \diamond & \quad \text{(Assumption)} & (1) \\
 \langle \Gamma \rangle^1; \langle \Delta \rangle^1, \langle \Delta_\mu \rangle^1 \vdash \llbracket P \rrbracket_g^1 \triangleright \diamond & \quad \text{(Theorem 5.1 [10], (10))} & (2) \\
 \mathcal{G}(\langle \Gamma \rangle^1), \Phi; \mathcal{G}(\langle \Delta \rangle^1), \Theta \vdash \mathcal{B}_\epsilon^k(\llbracket P \rrbracket_g^1) \triangleright \diamond & \quad \text{(Theorem 3.27 [1], (11))} & (3) \\
 \llbracket \mathcal{G}(\langle \Gamma \rangle^1) \rrbracket^2, \llbracket \Phi \rrbracket^2; \llbracket \mathcal{G}(\langle \Delta \rangle^1) \rrbracket^2, \llbracket \Theta \rrbracket^2 \vdash \llbracket \mathcal{B}_\epsilon^k(\llbracket P \rrbracket_g^1) \rrbracket^2 \triangleright \diamond & \quad \text{(Theorem 5.2 [10], (12))} & (4) \\
 \mathcal{H}(\Gamma), \Phi'; \mathcal{H}(\Delta), \Theta' \vdash \mathcal{A}_\epsilon^k(P)_g \triangleright \diamond & \quad \text{(Definition of } \mathcal{A}_\epsilon^k(\cdot)_g, \\ & \quad \text{Definition 3.5, Definition 3.9, (4))} & (5)
 \end{aligned}$$

□

THEOREM 2 (MINIMALITY RESULT FOR π). *Let P be a closed π process, with $\tilde{u} = \text{fn}(P)$ and $\tilde{v} = \text{rn}(P)$. If $\Gamma; \Delta, \Delta_\mu \vdash P \triangleright \diamond$, where Δ_μ only involves recursive session types, then $\mathcal{H}(\Gamma\sigma); \mathcal{H}(\Delta\sigma), \mathcal{H}(\Delta_\mu\sigma) \vdash \mathcal{F}(P) \triangleright \diamond$, where $\sigma = \{\text{init}(\tilde{u})/\tilde{u}\}$.*

PROOF.

$$\begin{aligned}
 \Gamma; \Delta \vdash P \triangleright \diamond & \quad \text{(Assumption)} & (6) \\
 \Gamma\sigma; \Delta\sigma \vdash P\sigma \triangleright \diamond & \quad \text{(Lemma C.1, (6))} & (7) \\
 \mathcal{H}(\Gamma\sigma), \Phi'; \mathcal{H}(\Delta\sigma), \Theta' \vdash \mathcal{A}_\epsilon^k(P\sigma)_g \triangleright \diamond & \quad \text{(P is initialized, Theorem 1)} & (8)
 \end{aligned}$$

To complete the proof, let us construct a well-formed derivation tree by using the appropriate typing rules for the higher-order calculus. Note that we apply one of the polyadic rules, i.e. PolyResS, used for typing HO π , derived by Arslanagić et al. [1].

$$\text{Par } (|\tilde{v}| - 1 \text{ times}) \frac{\text{for } r \in \tilde{v} \quad \mathcal{H}(\Gamma\sigma), \Phi'; \tilde{r} : \mathcal{H}(S) \vdash c^r?(b).(v s)(b!(s).\bar{s}!(\tilde{r})) \triangleright \diamond}{\mathcal{H}(\Gamma\sigma), \Phi'; \mathcal{H}(\Delta_\mu\sigma) \vdash \prod_{r \in \tilde{v}} P^r} \quad (9)$$

where $S = \Delta_\mu(r)$. By the definition of Φ' we have $c^r : \langle \langle ?(\mathcal{R}'^*(\Delta_\mu(r))); \text{end} \rangle \rangle \in \Phi'$ and by Def. 3.5 we have $\mathcal{R}'^*(\Delta_\mu(r)) = \mathcal{H}(S)$, so the right-hand of (9) is well-typed.

$$\begin{aligned}
 & \text{Nil } \frac{}{\mathcal{H}(\Gamma\sigma), \Phi'; \mathcal{H}(\Delta\sigma) \vdash \mathbf{0} \triangleright \diamond} \\
 & \text{Send } \frac{}{\mathcal{H}(\Gamma\sigma), \Phi'; \mathcal{H}(\Delta\sigma), \bar{c}_k!(\cdot); \text{end} \vdash \bar{c}_k!(\cdot)} \quad (9) \\
 & \text{Par } \frac{}{\mathcal{H}(\Gamma\sigma), \Phi'; \mathcal{H}(\Delta\sigma), \mathcal{H}(\Delta_\mu\sigma), \bar{c}_k!(\cdot); \text{end} \vdash \bar{c}_k!(\cdot).\mathbf{0} \mid \prod_{r \in \tilde{v}} P^r} \quad (10) \\
 & \text{PolyResS } \frac{\text{Par } \frac{}{\mathcal{H}(\Gamma\sigma), \Phi'; \mathcal{H}(\Delta\sigma), \mathcal{H}(\Delta_\mu\sigma), \bar{c}_k!(\cdot); \text{end}, \Theta' \vdash \bar{c}_k!(\cdot).\mathbf{0} \mid \prod_{r \in \tilde{v}} P^r \mid \mathcal{A}_\epsilon^k(P\sigma)_g \triangleright \diamond}{} \quad (10) \quad (8)}{\mathcal{H}(\Gamma\sigma); \mathcal{H}(\Delta\sigma), \mathcal{H}(\Delta_\mu\sigma) \vdash (v \tilde{c})(v \tilde{c}_r)(\prod_{r \in \tilde{v}} P^r \mid \bar{c}_k!(\cdot).\mathbf{0} \mid \mathcal{A}_\epsilon^k(P\sigma)_g)}
 \end{aligned}$$

□

C.3 Measuring the Optimization

Proposition 4.1. *If P is in normal form then $\#(P) \geq \frac{5}{3} \cdot \#^*(P)$.*

PROOF. Because $\#(Q) \geq \lfloor P \rfloor + 2 \cdot |\text{brn}(P)|$, it suffices to show that $\lfloor P \rfloor + 2 \cdot |\text{brn}(P)| \geq \frac{5}{3} \cdot \#^*(P)$. The proof is by induction on structure of P . We show one base case (output prefix followed by inaction) and three inductive cases (input, restriction, and parallel composition); other cases are similar:

- Case $P = u!(x).\mathbf{0}$. Then $\lfloor P \rfloor = 4$, $|\text{brn}(P)| = 0$, and $\lfloor P \rfloor^* = 2$. So, we have $\lfloor P \rfloor + 2 \cdot |\text{brn}(P)| \geq \frac{5}{3} \cdot \#^*(P)$
- Case $P = u!(x).P'$ with $P' \neq \mathbf{0}$. By IH we know $\lfloor P' \rfloor + |\text{brn}(P')| \geq \frac{5}{3} \cdot \#^*(P')$. We know $\lfloor P \rfloor = \lfloor P' \rfloor + 3$, $\lfloor P \rfloor^* = \lfloor P' \rfloor^* + 1$ and $\#_X(P) = \#_X(P')$. So, $\lfloor P \rfloor + 2 \cdot |\text{brn}(P)| + 3 \geq \frac{5}{3} \cdot \#^*(P) + 1 > \frac{5}{3} \cdot (\#^*(P) + 1)$.
- Case $P = (v r : S)P'$ with $\text{tr}(S)$. Then $\lfloor P \rfloor = \lfloor P' \rfloor$ (by Def. 3.2) and $|\text{brn}(P)| = |\text{brn}(P')| + 1$. Further, we have $\#_X(P) = \#_X(P')$ and $\lfloor P \rfloor^* = \lfloor P' \rfloor^* + 1$. Now, by IH we can conclude that $\lfloor P \rfloor + 2 \cdot (|\text{brn}(P)| + 1) \geq \frac{5}{3} \cdot \#^*(P) + 1 > \frac{5}{3} \cdot (\#^*(P) + 1)$.

- Case $P = P_1 \mid \dots \mid P_n$. By IH we know

$$\lfloor P_i \rfloor + 2 \cdot |\text{brn}(P_i)| \geq \frac{5}{3} \cdot \#^*(P_i) \quad (11)$$

for $i \in \{1, \dots, n\}$. We know $\lfloor P \rfloor = \sum_{i=1}^n \lfloor P_i \rfloor + (n-1)$, $|\text{brn}(P)| = \sum_{i=1}^n |\text{brn}(P_i)|$, and $\lfloor P \rfloor^* = \sum_{i=1}^n \#^*(P_i) + (n-1)$. So, we should show

$$\sum_{i=1}^n \lfloor P_i \rfloor + 2 \cdot \sum_{i=1}^n |\text{brn}(P_i)| + (n-1) \geq \frac{5}{3} \cdot \left(\sum_{i=1}^n \#^*(P_i) + (n-1) \right) \quad (12)$$

That is,

$$\sum_{i=1}^n (\lfloor P_i \rfloor + 2 \cdot |\text{brn}(P_i)| + 1) \geq \frac{5}{3} \cdot \sum_{i=1}^n (\#^*(P_i) + 1)$$

Equivalent to

$$\sum_{i=1}^n (\lfloor P_i \rfloor + 2 \cdot |\text{brn}(P_i)| + 1) \geq \sum_{i=1}^n \left(\frac{5}{3} \cdot (\#^*(P_i) + 1) \right)$$

We show that for each $i \in \{1, \dots, n\}$ the following holds:

$$\lfloor P_i \rfloor + 2 \cdot |\text{brn}(P_i)| + 1 \geq \frac{5}{3} \cdot (\#^*(P_i) + 1)$$

That is

$$A = \frac{\lfloor P_i \rfloor + 2 \cdot |\text{brn}(P_i)| + 1}{\#^*(P_i) + 1} \geq \frac{5}{3}$$

As P is in the normal form, we know that $P_i \equiv \alpha.P'_i$ where α is some sequential prefix. So, by Def. 3.2 and Def. 4.5 we know that for some p^* and p we have $\lfloor P_i \rfloor^* = p^* + \lfloor P'_i \rfloor^*$, $\#_X(P_i) = \#_X(P'_i)$, and $\lfloor P_i \rfloor = p + \lfloor P'_i \rfloor$. We can distinguish two sub-cases: (i) $P'_i \neq \mathbf{0}$ and (ii) otherwise. We consider sub-case (i). By (11) we have:

$$A \geq \frac{\frac{5}{3} \cdot \#^*(P'_i) + p + 1}{\#^*(P'_i) + p^* + 1} \geq \frac{5}{3}$$

We need to find prefix α such that p and $\frac{p}{p^*}$ are the least. We notice that for the output prefix we have $p = 3$ and $\frac{p}{p^*} = \frac{3}{1}$. So, the following holds

$$\frac{\frac{5}{3} \cdot \#^*(P'_i) + 4}{\#^*(P'_i) + 2} = \frac{5}{3} \cdot \frac{\#^*(P'_i) + \frac{12}{5}}{\#^*(P'_i) + 2} \geq \frac{5}{3}$$

Now, we consider sub-case (ii) when $P'_i = \mathbf{0}$. In this case we have $\lfloor P'_i \rfloor + 2 \cdot \text{brn}(P'_i) = \#^*(P'_i) = 1$. We pick p and p^* as in the previous sub-case. So, we have

$$\frac{\lfloor P_i \rfloor + 2 \cdot |\text{brn}(P_i)| + 1}{\#^*(P_i) + 1} = \frac{1 + 3 + 1}{1 + 1} = \frac{5}{3}$$

So, we can conclude that inequality (12) holds. \square

C.4 Typability of the Optimized Breakdown Function

Lemma C.6 (Typability of Breakdown: $\mathbb{A}_{\text{rec}}^k \tilde{y}(\cdot)_g$). *Let P be an initialized process. If $\Gamma \cdot X : \Delta_\mu ; \Delta \vdash P \triangleright \diamond$ then:*

$$\mathcal{G}(\Gamma \setminus \tilde{x}); \Theta \vdash \mathbb{A}_{\text{rec}}^k \tilde{y}(P)_g \triangleright \diamond \quad (k > 0)$$

where $\tilde{x} \subseteq \text{fn}(P)$ such that $\Delta \setminus \tilde{x} = \emptyset$, $\tilde{y} = \tilde{v} \cdot \tilde{m}$ where \tilde{v} is such that and indexed $\Gamma, \Delta(\tilde{v}, \tilde{x})$ and $\tilde{m} = \text{codom}(g)$. Also, $\Theta = \Theta_\mu, \Theta_X(g)$ where balanced (Θ_μ) with $\text{dom}(\Theta_\mu) = \{c_k^r, c_{k+1}^r, \dots, c_{k+\lfloor P \rfloor^*-1}^r\} \cup \{c_{k+1}^r, \dots, c_{k+\lfloor P \rfloor^*-1}^r\}$ and $\Theta_\mu(c_k^r) = \mu.t.?(N); t$ if $g \neq \emptyset$, otherwise $\Theta_\mu(c_k^r) = \langle \tilde{N} \rangle$ where $\tilde{N} = (\mathcal{G}(\Gamma), \mathcal{G}(\Delta_\mu \cdot \Delta))(\tilde{y})$. and $\Theta_X(g) = \bigcup_{X \in \text{dom}(g)} c_X^r : \langle \tilde{M}_X \rangle$ where $\tilde{M}_X = (\mathcal{G}(\Gamma), \mathcal{G}(\Delta_\mu))(g(X))$.

PROOF. The proof is by the induction of the structure of P . We consider cases when $g \neq \emptyset$, as cases when $g = \emptyset$ are similar.

(1) Case $P = X$. The only rule that can be applied here is RVar:

$$\text{RVar} \frac{}{\Gamma \cdot X : \Delta ; \Delta \vdash X \triangleright \diamond} \quad (13)$$

Let $\tilde{x} = \emptyset$ as $\text{fn}(P) = \emptyset$. So, $\tilde{y} = \tilde{m}$ where $\tilde{m} = g(X)$. Since $\lfloor X \rfloor^* = 1$ we have $\Theta_\mu = \{c_k^r : \mu.t.?(N); t\}$ where $\tilde{N} = (\mathcal{G}(\Gamma), \mathcal{G}(\Delta))(\tilde{y})$. In this case $\Delta_\mu = \Delta$, thus $\tilde{N} = \tilde{M}$. We shall then prove the following judgment:

$$\mathcal{G}(\Gamma \setminus x); \Theta \vdash \mathbb{A}_{\text{rec}}^k \tilde{y}(X)_g \triangleright \diamond \quad (14)$$

By Tab. 3, we have:

$$\begin{array}{c} \mathbb{A}_{\text{rec}}^k \tilde{y}(X)_g = \mu X. c_k^r?(\tilde{y}). c_X^r! \langle \tilde{y} \rangle. X \\ \text{Req} \frac{\text{RVar} \frac{\mathcal{G}(\Gamma \setminus x), X : \Theta; \Theta \vdash X}{\mathcal{G}(\Gamma \setminus x), X : \Theta; \Theta \vdash X} \quad \text{PolySend} \frac{\mathcal{G}(\Gamma \setminus x), X : \Theta; \tilde{y} : \tilde{M} \vdash \tilde{y} \triangleright \tilde{M}}{\mathcal{G}(\Gamma \setminus x), X : \Theta; \tilde{y} : \tilde{M} \vdash \tilde{y} \triangleright \tilde{M}}}{\mathcal{G}(\Gamma \setminus x), X : \Theta; c_X^r : \langle \tilde{M} \rangle, \tilde{y} : \tilde{M} \vdash c_X^r! \langle \tilde{y} \rangle. X} \end{array} \quad (15)$$

The following tree proves this case:

$$\begin{array}{c} \text{Rcv} \frac{\text{PolySend} \frac{(15) \quad \mathcal{G}(\Gamma \setminus x), X : \Theta; \tilde{y} : \tilde{M} \vdash \tilde{y} \triangleright \tilde{M}}{\mathcal{G}(\Gamma \setminus x), X : \Theta; \Theta \vdash c_k^r?(\tilde{y}). c_X^r! \langle \tilde{y} \rangle. X \triangleright \diamond}}{\mathcal{G}(\Gamma \setminus x); \Theta \vdash \mu X. c_k^r?(\tilde{y}). c_X^r! \langle \tilde{y} \rangle. X \triangleright \diamond} \\ \text{Rec} \end{array} \quad (16)$$

where $\Theta = \Theta_\mu, \overline{c_X^r} : \mu t. ! \langle \tilde{M} \rangle; t$.

- (2) Case $P = u_i! \langle z_j \rangle. P'$. Let $u_i : C$. We distinguish three sub-cases: (i) $C = S = \mu t. ! \langle C_z \rangle; S'$, (ii) $C = S = ! \langle C_z \rangle; S'$, and (iii) $C = \langle C_z \rangle$. We consider first two sub-cases, as (iii) is shown similarly. The only rule that can be applied here is SEND:

$$\text{Send} \frac{\Gamma \cdot X : \Delta_\mu; \Delta \cdot u_i : S' \vdash P' \triangleright \diamond \quad \Gamma \cdot X : \Delta_\mu; \Delta_z \vdash z_j \triangleright C}{\Gamma \cdot X : \Delta_\mu; \Delta \cdot u_i : S \cdot \Delta_z \vdash u_i! \langle z_j \rangle. P' \triangleright \diamond} \quad (17)$$

Then, by IH on the right assumption of (17) we have:

$$\mathcal{G}(\Gamma \setminus \tilde{x}'); \Theta' \vdash \mathbb{A}_{\text{rec}}^{k+1} \tilde{y}'(P')_g \triangleright \diamond \quad (18)$$

where $\tilde{x}' \subseteq \text{fn}(P)$ such that $(\Delta, r : S') \setminus \tilde{x}' = \emptyset$, and $\tilde{y}' = \tilde{v}' \cup \tilde{m}$ where $\text{indexed}_{\Gamma, \Delta, r, S'}(\tilde{v}', \tilde{x}')$. Also, $\Theta' = \Theta'_\mu, \Theta_X$ where Θ'_μ such that $\text{balanced}(\Theta'_\mu)$ with

$$\text{dom}(\Theta'_\mu) = \{c_{k+1}^r, c_{k+2}^r, \dots, c_{k+[P']^*+1}^r\} \cup \{\overline{c_{k+2}^r}, \dots, \overline{c_{k+[P']^*-1}^r}, \overline{c_{k+[P']^*+1}^r}\}$$

and $\Theta'_\mu(c_{k+1}^r) = \mu t. ?(\tilde{N}'); t$ where $\tilde{N}' = (\mathcal{G}(\Gamma), \mathcal{G}(\Delta_\mu \cdot \Delta \cdot u_i : S))(\tilde{y}')$. By applying Lem. C.2 on the second assumption of (17) we have:

$$\mathcal{G}(\Gamma \cdot X : \Delta_\mu); \mathcal{G}(\Delta_z) \vdash \tilde{z} \triangleright \mathcal{G}(C_z) \triangleright \diamond \quad (19)$$

Let $\sigma = \text{next}(u_i)$ and in sub-case (i) $\sigma_1 = \{\tilde{n}/\tilde{u}\}$ where $\tilde{n} = (u_{i+1}, \dots, u_{i+\mathcal{G}(S)})$ and $\tilde{u} = (u_i, \dots, u_{i+\mathcal{G}(S)-1})$, otherwise (ii) $\sigma_1 = \epsilon$. We define $\tilde{x} = \tilde{x}', z$ and $\tilde{y} = \tilde{y}' \sigma_1, \tilde{z} \cdot u_i$.

By construction $\tilde{x} \subseteq P$ and $(\Delta \cdot u_i : S \cdot \Delta_z) \setminus \tilde{x} = \emptyset$. Further, we may notice that $\tilde{y} = \tilde{v} \cdot \tilde{m}$ such that $\text{indexed}_{\Gamma, \Delta, u_i : S, \Delta_z}(\tilde{v}, \tilde{x})$ and $\tilde{y}' = \tilde{m} \cup \text{fnb}(P', \tilde{y})$. Let $\Theta = \Theta_\mu, \Theta_X$ where

$$\Theta_\mu = \Theta'_\mu, c_k^r : \mu t. ?(\tilde{N}); t, \overline{c_{k+1}^r} : \mu t. ! \langle \tilde{N}' \rangle; t$$

where $\tilde{N} = (\mathcal{G}(\Gamma), \mathcal{G}(\Delta_\mu \cdot \Delta \cdot u_i : S \cdot \Delta_z))(\tilde{y})$. By construction and since $[P]^* = [P']^* + 1$ we have

$$\text{dom}(\Theta_\mu) = \{c_k^r, c_{k+1}^r, \dots, c_{k+[P]^*-1}^r\} \cup \{\overline{c_{k+1}^r}, \dots, \overline{c_{k+[P]^*-1}^r}\}$$

and $\text{balanced}(\Theta_\mu)$. By Tab. 3 we have:

$$\mathbb{A}_{\text{rec}}^k \tilde{y}(P)_g = \mu X. c_k^r?(\tilde{y}). u_i! \langle \tilde{z} \rangle. c_{k+1}^r! \langle \tilde{y}' \sigma_1 \rangle. X \mid \mathbb{A}_{\text{rec}}^{k+1} \tilde{y}' \sigma_1 (P') \sigma_1 g \quad (20)$$

where in sub-case (i) $l = i$ and in sub-case (ii) $l = f(S)$. We shall prove the following judgment:

$$\mathcal{G}(\Gamma \setminus \tilde{x}); \Theta \vdash \mathbb{A}_{\text{rec}}^k \tilde{y}(P)_g \triangleright \diamond \quad (21)$$

Let $\Delta_1 = \Delta, u_i : S, \Delta_z$ and $\tilde{u}_i = \text{bn}(u_i : S)$. We use some auxiliary sub-trees:

$$\text{RVar} \frac{\mathcal{G}(\Gamma \setminus \tilde{x}) \cdot X : \Theta; \Theta \vdash X \triangleright \diamond}{\mathcal{G}(\Gamma \setminus \tilde{x}) \cdot X : \Theta; \Theta \vdash X \triangleright \diamond} \quad (22)$$

$$\text{PolySend} \frac{\text{PolyVar} \frac{(22) \quad \mathcal{G}(\Gamma \setminus \tilde{x}) \cdot X : \Theta; \mathcal{G}(\Delta, u_i \sigma : S') \vdash \tilde{y}' \sigma_1 \triangleright \tilde{N}'}{\mathcal{G}(\Gamma \setminus \tilde{x}) \cdot X : \Theta; \mathcal{G}(\Delta, u_i \sigma : S') \vdash \tilde{y}' \sigma_1 \triangleright \tilde{N}'} \quad \text{PolySend} \frac{\mathcal{G}(\Gamma \setminus \tilde{x}) \cdot X : \Theta; \Theta, \mathcal{G}(\Delta, u_i \sigma : S') \vdash c_{k+1}^r! \langle \tilde{y}' \sigma_1 \rangle. X \triangleright \diamond}{\mathcal{G}(\Gamma \setminus \tilde{x}) \cdot X : \Theta; \Theta, \mathcal{G}(\Delta, u_i \sigma : S') \vdash c_{k+1}^r! \langle \tilde{y}' \sigma_1 \rangle. X \triangleright \diamond}}{\mathcal{G}(\Gamma \setminus \tilde{x}) \cdot X : \Theta; \Theta, \mathcal{G}(\Delta, u_i \sigma : S') \vdash c_{k+1}^r! \langle \tilde{y}' \sigma_1 \rangle. X \triangleright \diamond} \quad (23)$$

Here, in typing the right-hand assumption, we may notice that in sub-case (i) $\mathcal{G}(u_i : S) = \tilde{u}_i : ! \langle \mathcal{G}(C_z) \rangle$; end, $\mathcal{G}(S')$ and $\mathcal{G}(u_i \sigma : S') = \tilde{u}_{i+1} : \mathcal{G}(S')$ where $\tilde{u}_{i+1} = (u_{i+1}, \dots, u_{i+|\mathcal{G}(S')|})$. So right-hand side follows by Def. 4.12 and Lem. C.1. Otherwise, in sub-case (ii) by Def. 4.4 and Def. 12 we know $\mathcal{G}(u_i : S) = \mathcal{G}(u_i \sigma : S')$ and by definition $\tilde{u}_i \subseteq \tilde{m} \subseteq \tilde{y}' \sigma_1$.

$$\text{PolySend} \frac{(23) \quad (19)}{\mathcal{G}(\Gamma \setminus \bar{x}) \cdot X : \Theta; \Theta, \mathcal{G}(\Delta_1) \vdash u_l! \langle \bar{z} \rangle . c_{k+1}^r \overline{! \langle \bar{y}' \sigma_1 \rangle} . X \triangleright \diamond} \quad (24)$$

$$\text{PolyRev} \frac{(24) \quad \text{PolyVar} \frac{\mathcal{G}(\Gamma \setminus \bar{x}) \cdot X : \Theta; \mathcal{G}(\Delta_1) \vdash \bar{y} \triangleright \bar{N}}{\mathcal{G}(\Gamma \setminus \bar{x}) \cdot X : \Theta; \Theta \vdash c_k^r ? (\bar{y}) . u_l! \langle \bar{z} \rangle . c_{k+1}^r \overline{! \langle \bar{y}' \sigma_1 \rangle} . X \triangleright \diamond}}{\mathcal{G}(\Gamma \setminus \bar{x}) \cdot X : \Theta; \Theta \vdash c_k^r ? (\bar{y}) . u_l! \langle \bar{z} \rangle . c_{k+1}^r \overline{! \langle \bar{y}' \sigma_1 \rangle} . X \triangleright \diamond}} \quad (25)$$

We may notice that by Def. 4.4 and Def. 12 we have $\mathcal{G}(\Delta_1) = \mathcal{G}(\Delta), \bar{u}_i : \mathcal{G}(S), \mathcal{G}(\Delta_z)$. So, in sub-case (i) as $u_i = u_l$ we have $\mathcal{G}(\Delta_1)(u_l) = ! \langle \mathcal{G}(C_z) \rangle$; end. In sub-case (ii), by Lem. C.5 and as $u_l = u_{f(S)}$ we know $\mathcal{G}(\Delta_1)(u_l) = \mu t. ! \langle \mathcal{G}(C_z) \rangle$; t. The following tree proves this case:

$$\text{Par} \frac{(25) \quad (18)}{\mathcal{G}(\Gamma \setminus \bar{x}); \Theta \vdash \mu X . c_k^r ? (\bar{y}) . u_l! \langle \bar{z} \rangle . c_{k+1}^r \overline{! \langle \bar{y}' \sigma_1 \rangle} . X \mid \mathbb{A}_{\text{rec}}^{k+1} \bar{y}' \sigma_1 (P' \sigma)_g \triangleright \diamond} \quad (26)$$

Note that we have used the following for the right assumption of (26):

$$\mathbb{A}_{\text{rec}}^{k+1} \bar{y}' (P') \equiv \alpha \mathbb{A}_{\text{rec}}^{k+1} \bar{y}' \sigma_1 (P' \sigma)$$

- (3) Case $P = u_i ? (z) . P'$. We distinguish three sub-cases: (i) $C = S = \mu t. ? (C_z); S'$, (ii) $C = S = ? (C_z); S'$, and (iii) $C = \langle C_z \rangle$. The only rule that can be applied here is Rcv:

$$\text{Rcv} \frac{\Gamma \cdot X : \Delta_\mu; \Delta, u_i : S', \Delta_z \vdash P' \triangleright \diamond \quad \Gamma \cdot X : \Delta_\mu; \Delta_z \vdash z \triangleright C}{(\Gamma \setminus z) \cdot X : \Delta_\mu; \Delta, u_i : S \vdash u_i ? (z) . P' \triangleright \diamond} \quad (27)$$

Let $\bar{x}' \subseteq \text{fn}(P')$ such that $(\Delta \cdot u_i : S' \cdot \Delta_z) \setminus \bar{x}' = \emptyset$ and $\bar{y}' = \bar{v} \cup \bar{m}$ such that $\text{indexed}_{\Gamma, \Delta, u_i; S'}(\bar{v}', \bar{x}')$. Also, $\Theta' = \Theta'_\mu, \Theta_X(g)$ where $\text{balanced}(\Theta'_\mu)$ with

$$\text{dom}(\Theta'_\mu) = \{c_{k+1}^r, c_{k+2}^r, \dots, c_{k+[P']^*+1}^r\} \cup \{\overline{c_{k+2}^r}, \dots, \overline{c_{k+[P']^*+1}^r}\}$$

and $\Theta'_\mu(c_{k+1}^r) = \mu t. ? (\bar{N}')$; t, where $\bar{N}' = (\mathcal{G}(\Gamma), \mathcal{G}(\Delta_\mu \cdot \Delta \cdot u_i : S))(\bar{y}')$. Then, by IH on the right assumption of (17) we have:

$$\mathcal{G}(\Gamma \setminus \bar{x}'); \Theta'_\mu \vdash \mathbb{A}_{\text{rec}}^{k+1} \bar{y}' (P') \triangleright \diamond \quad (28)$$

By applying Lem. C.2 to the second assumption of (17), we have:

$$\mathcal{G}(\Gamma) \cdot X : \Theta_\mu; \mathcal{G}(\Delta_z) \vdash \bar{z} \triangleright \mathcal{G}(C) \quad (29)$$

Let $\sigma = \text{next}(u_i)$ and in sub-case (i) $\sigma_1 = \{\bar{n}/\bar{u}\}$ where $\bar{n} = (u_{i+1}, \dots, u_{i+\mathcal{G}(S)})$ and $\bar{u} = (u_i, \dots, u_{i+\mathcal{G}(S)-1})$, otherwise in sub-case (ii) $\sigma_1 = \epsilon$. We define $\bar{x} = \bar{x}' \sigma \setminus z$ and $\bar{y} = \bar{y}' \sigma_1 \setminus \bar{z}$ with $|\bar{z}| = |\mathcal{G}(C)|$. By construction $\bar{x} \subseteq \text{fn}(P)$ and $(\Delta, r : S) \setminus \bar{x} = \emptyset$. Further, we may notice that $\bar{y} = \bar{v} \cdot \bar{m}$ where \bar{v} is such that $\text{indexed}_{\Gamma, \Delta, r; S, \Delta_z}(\bar{v}, \bar{x})$ and $\bar{y} = \bar{m} \cup \text{fnb}(P', \bar{y}\bar{z})$. Let $\Theta = \Theta_\mu, \Theta_X$ where

$$\Theta_\mu = \Theta'_\mu, c_k^r : \mu t. ? (\bar{N}); t, \overline{c_{k+1}^r} : \mu t. ! \langle \bar{N}' \rangle; t$$

where $\bar{N} = (\mathcal{G}(\Gamma), \mathcal{G}(\Delta_\mu \cdot \Delta \cdot u_i : S, \Delta_z))(\bar{y})$. By construction and since $[P]^* = [P']^* + 1$ we have

$$\text{dom}(\Theta_\mu) = \{c_k^r, c_{k+1}^r, \dots, c_{k+[P]^*-1}^r\} \cup \{\overline{c_{k+1}^r}, \dots, \overline{c_{k+[P]^*-1}^r}\}$$

and $\text{balanced}(\Theta_\mu)$. By Tab. 3, we have:

$$\mathbb{A}_{\text{rec}}^k \bar{y} (P)_g = \mu X . c_k^r ? (\bar{y}) . u_l! \langle \bar{z} \rangle . c_{k+1}^r \overline{! \langle \bar{y}' \sigma_1 \rangle} . X \mid \mathbb{A}_{\text{rec}}^{k+1} \bar{y} (P' \sigma)_g \quad (30)$$

where in sub-case (i) $l = i$ and in sub-case (ii) $l = f(S)$. Let $\Gamma_1 = \Gamma \setminus \bar{x}$. We shall prove the following judgment:

$$\mathcal{G}(\Gamma_1 \setminus z); \Theta \vdash \mathbb{A}_{\text{rec}}^k \bar{y} (P)_g \triangleright \diamond \quad (31)$$

We use some auxiliary sub-trees:

$$\text{RVar} \frac{\mathcal{G}(\Gamma) \cdot X : \Theta; \Theta \vdash X \triangleright \diamond}{\mathcal{G}(\Gamma) \cdot X : \Theta; \Theta \vdash X \triangleright \diamond} \quad (32)$$

$$\text{PolySend} \frac{(32) \quad \text{PolyVar} \frac{\mathcal{G}(\Gamma) \cdot X : \Theta; \mathcal{G}(\Delta, u_l : S', \Delta_z) \vdash \bar{y}' \sigma_1 \triangleright \bar{N}'}{\mathcal{G}(\Gamma) \cdot X : \Theta; \Theta, \mathcal{G}(\Delta_1) \vdash c_{k+1}^r \overline{! \langle \bar{y}' \sigma_1 \rangle} . X \triangleright \diamond}}{\mathcal{G}(\Gamma) \cdot X : \Theta; \Theta, \mathcal{G}(\Delta_1) \vdash c_{k+1}^r \overline{! \langle \bar{y}' \sigma_1 \rangle} . X \triangleright \diamond} \quad (33)$$

Here, in typing the right-hand assumption, we may notice that in sub-case (i) $\mathcal{G}(u_i : S) = \tilde{u}_i : ?(\mathcal{G}(C_z))$; end, $\mathcal{G}(S')$ and $\mathcal{G}(u_i \sigma : S') = \tilde{u}_{i+1} : \mathcal{G}(S')$ where $\tilde{u}_{i+1} = (u_{i+1}, \dots, u_{i+|\mathcal{G}(S')|})$ and by definition $\tilde{u}_{i+1} \subseteq \tilde{y}'\sigma_1$. So right-hand side follows by Def. 4.12 and Lem. C.1. Otherwise, in sub-case (ii) by Def. 4.4 and Def. 12 we know $\mathcal{G}(u_i : S) = \mathcal{G}(u_i \sigma : S')$.

$$\text{PolyRcv} \frac{(33) \quad (29)}{\mathcal{G}(\Gamma \setminus z) \cdot X : \Theta; \Theta, \mathcal{G}(\Delta, u_i : S) \vdash u_l?(\tilde{z}).c_{k+1}^r! \langle \tilde{y}'\sigma_1 \rangle . X \triangleright \diamond} \quad (34)$$

$$\text{PolyRcv} \frac{(34) \quad \text{PolyVar} \frac{\mathcal{G}(\Gamma \setminus z) \cdot X : \Theta; \mathcal{G}(\Delta, r : S) \vdash \tilde{y} \triangleright \tilde{N}}{\mathcal{G}(\Gamma_1 \setminus z) \cdot X : \Theta; \Theta \vdash c_k^r?(\tilde{y}).u_l?(\tilde{z}).c_{k+1}^r! \langle \tilde{y}'\sigma_1 \rangle . X \triangleright \diamond}}{\mathcal{G}(\Gamma_1 \setminus z); \Theta \vdash \mu X.c_k^r?(\tilde{y}).u_l?(\tilde{z}).c_{k+1}^r! \langle \tilde{y}'\sigma_1 \rangle . X \triangleright \diamond} \quad (35)$$

We may notice that by Def. 4.4 and Def. 12 we have $\mathcal{G}(\Delta_1) = \mathcal{G}(\Delta, \tilde{u}_i : \mathcal{G}(S), \mathcal{G}(\Delta_z))$. So, in sub-case (i) as $u_i = u_l$ we have $\mathcal{G}(\Delta_1)(u_l) = ?(\mathcal{G}(C_z))$; end. In sub-case (ii), by Lem. C.5 and as $u_l = u_{f(S)}$ we know $\mathcal{G}(\Delta_1)(u_l) = \mu t.?(\mathcal{G}(C_z)); t$. The following tree proves this case:

$$\text{Par} \frac{(35) \quad (28)}{\mathcal{G}(\Gamma_1 \setminus z); \Theta \vdash \mu X.c_k^r?(\tilde{y}).u_l?(\tilde{z}).c_{k+1}^r! \langle \tilde{y}'\sigma_1 \rangle . X \mid \mathbb{A}_{\text{rec}}^{k+1} \tilde{y}'\sigma_1 (P'\sigma)_g} \quad (36)$$

Note that we have used the following for the right assumption of (36):

$$\mathbb{A}_{\text{rec}}^{k+1} \tilde{y}'\sigma_1 (P') \equiv_{\alpha} \mathbb{A}_{\text{rec}}^{k+1} \tilde{y}'\sigma_1 (P'\sigma)$$

(4) Case $P = Q_1 \mid Q_2$. The only rule that can be applied here is PAR:

$$\text{Par} \frac{\Gamma \cdot X : \Delta_\mu; \Delta_1 \vdash Q_1 \triangleright \diamond \quad \Gamma \cdot X : \Delta_\mu; \Delta_2 \vdash Q_2 \triangleright \diamond}{\Gamma \cdot X : \Delta_\mu; \Delta_1, \Delta_2 \vdash Q_1 \mid Q_2 \triangleright \diamond} \quad (37)$$

Here we assume $\text{frv}(Q_1)$. So, by IH on the first and second assumption of (37) we have:

$$\mathcal{G}(\Gamma \setminus x_1); \Theta_1 \vdash \mathbb{A}_{\text{rec}}^k \tilde{y}_1 (Q_1)_g \triangleright \diamond \quad (38)$$

$$\mathcal{G}(\Gamma \setminus x_1); \Theta_2 \vdash \mathbb{A}_{\text{rec}}^{k+l+1} \tilde{y}_2 (Q_2)_\emptyset \triangleright \diamond \quad (39)$$

where for $i \in \{1, 2\}$ we have $\tilde{x}_i \subseteq \text{fn}(Q_i)$ such that $\Delta_i \setminus \tilde{x} = \emptyset$, and $\tilde{y}_i = \tilde{v}_i \cdot \tilde{m}_i$ where \tilde{v}_i is such that $\text{indexed}_{\Gamma, \Delta_i}(\tilde{v}_i, \tilde{x}_i)$ and $\tilde{m}_i = \text{codom}(g_i)$. Further, $\Theta_i = \Theta_\mu^i \cdot \Theta_X(g_i)$ where

$$\text{dom}(\Theta_\mu^1) = \{c_{k+1}^r, c_{k+2}^r, \dots, c_{k+[Q_1]^*}^r\} \cup \{\overline{c_{k+2}^r}, \dots, \overline{c_{k+[Q_1]^*}^r}\}$$

$$\text{dom}(\Theta_\mu^2) = \{c_{k+l+1}^r, c_{k+l+2}^r, \dots, c_{k+l+[Q_2]^*}^r\}$$

and $\Theta_\mu^1(c_{k+1}^r) = \mu t.?(\tilde{N}^1); t$ and $\Theta_\mu^2(c_{k+l+1}^r) = \langle \tilde{N}^2 \rangle$ with $\tilde{N}^i = (\mathcal{G}(\Gamma), \mathcal{G}(\Delta_\mu \cdot \Delta_i))(\tilde{y}_i)$. Let $\Delta = \Delta_1 \cdot \Delta_2$. We define $\tilde{x} = \tilde{x}_1 \cdot \tilde{x}_2$ and $\tilde{y} = \tilde{y}_1 \cdot \tilde{y}_2$. By construction $\tilde{x} \subseteq \text{fn}(P)$ and $(\Delta_1 \cdot \Delta_2) \setminus \tilde{x} = \emptyset$. Further, we may notice that $\tilde{y} = \tilde{v} \cdot \tilde{m}$ where $\tilde{m} = \text{codom}(g)$ and $\text{indexed}_{\Gamma, \Delta}(\tilde{v}, \tilde{x})$. We shall prove the following judgment:

$$\mathcal{G}(\Gamma \setminus \tilde{x}); \Theta \vdash \mathbb{A}_{\text{rec}}^k \tilde{y} (P)_g \triangleright \diamond \quad (40)$$

where $\Theta = \Theta_1 \cdot \Theta_2 \cdot \Theta'$ with

$$\Theta' = c_k^r : \mu t.?(\tilde{N}); t \cdot \overline{c_{k+1}^r} : \mu t.!(\tilde{N}_1); t$$

By Tab. 3 we have:

$$\mathbb{A}_{\text{rec}}^k \tilde{y} (P)_g = \mu X.c_k^r?(\tilde{y}).(\overline{c_{k+1}^r! \langle \tilde{y}_1 \rangle . X} \mid c_{k+l+1}^r! \langle \tilde{y}_2 \rangle) \mid \mathbb{A}_{\text{rec}}^{k+1} \tilde{y}_1 (Q_1)_g \mid \mathbb{A}_{\text{rec}}^{k+l+1} \tilde{y}_2 (Q_2)_\emptyset$$

We use some auxiliary sub-trees:

$$\text{Nil} \frac{\mathcal{G}(\Gamma \setminus \tilde{x}) \cdot X : \Theta' \vdash \mathbf{0} \triangleright \diamond}{\mathcal{G}(\Gamma \setminus \tilde{x}) \cdot X : \Theta'; \tilde{y}_2 : \tilde{N}_2 \vdash \tilde{y}_2 \triangleright \tilde{N}_2} \quad \text{PolyVar} \frac{\mathcal{G}(\Gamma \setminus \tilde{x}) \cdot X : \Theta'; \tilde{y}_2 : \tilde{N}_2 \vdash \tilde{y}_2 \triangleright \tilde{N}_2}{\mathcal{G}(\Gamma \setminus \tilde{x}) \cdot X : \Theta'; \tilde{y}_2 : \tilde{N}_2 \vdash c_{k+l+1}^r! \langle \tilde{y}_2 \rangle} \quad (41)$$

$$\text{Rvar} \frac{\mathcal{G}(\Gamma \setminus \tilde{x}) \cdot X : \Theta'; \Theta' \vdash X \triangleright \diamond}{\mathcal{G}(\Gamma \setminus \tilde{x}) \cdot X : \Theta'; \Theta' \cdot \tilde{y}_1 : \tilde{N}_1 \vdash \tilde{y}_1 \triangleright \tilde{N}_1} \quad \text{PolyVar} \frac{\mathcal{G}(\Gamma \setminus \tilde{x}) \cdot X : \Theta'; \tilde{y}_1 : \tilde{N}_1 \vdash \tilde{y}_1 \triangleright \tilde{N}_1}{\mathcal{G}(\Gamma \setminus \tilde{x}) \cdot X : \Theta'; \Theta' \cdot \tilde{y}_1 : \tilde{N}_1 \vdash \overline{c_{k+1}^r! \langle \tilde{y}_1 \rangle} . X \triangleright \diamond} \quad (42)$$

$$\text{Par} \frac{\mathcal{G}(\Gamma \setminus \tilde{x}) \cdot X : \Theta'; \Theta' \cdot \tilde{y}_1 : \tilde{N}_1 \vdash \overline{c_{k+1}^r! \langle \tilde{y}_1 \rangle} . X \triangleright \diamond \quad \mathcal{G}(\Gamma \setminus \tilde{x}) \cdot X : \Theta'; \Theta' \cdot \tilde{y}_2 : \tilde{N}_2 \vdash c_{k+l+1}^r! \langle \tilde{y}_2 \rangle \triangleright \diamond}{\mathcal{G}(\Gamma \setminus \tilde{x}) \cdot X : \Theta'; \Theta' \cdot \tilde{y} : \tilde{N} \vdash \overline{c_{k+1}^r! \langle \tilde{y}_1 \rangle} . X \mid c_{k+l+1}^r! \langle \tilde{y}_2 \rangle \triangleright \diamond} \quad (41)$$

$$\begin{array}{c}
 \text{PolyVar} \frac{(42) \quad \mathcal{G}(\Gamma \setminus \bar{x}) \cdot X : \Theta'; \bar{y} : \bar{N} \vdash \bar{y} \triangleright \bar{N}}{\mathcal{G}(\Gamma \setminus \bar{x}) \cdot X : \Theta'; \Theta' \vdash c_k^r(\bar{y}).(\overline{c_{k+1}^r}!\langle \bar{y}_1 \rangle).X \mid c_{k+l+1}^r!\langle \bar{y}_2 \rangle) \triangleright \diamond} \\
 \text{Rv} \\
 \text{Rec} \frac{\mathcal{G}(\Gamma \setminus \bar{x}); \Theta' \vdash \mu X.c_k^r(\bar{y}).(\overline{c_{k+1}^r}!\langle \bar{y}_1 \rangle).X \mid c_{k+l+1}^r!\langle \bar{y}_2 \rangle) \triangleright \diamond}{\mathcal{G}(\Gamma \setminus \bar{x}); \Theta' \vdash \mu X.c_k^r(\bar{y}).(\overline{c_{k+1}^r}!\langle \bar{y}_1 \rangle).X \mid c_{k+l+1}^r!\langle \bar{y}_2 \rangle) \triangleright \diamond}
 \end{array} \quad (43)$$

The following tree proves this case:

$$\begin{array}{c}
 \text{Par} \frac{\text{Par} \frac{(38) \quad (39) \quad \mathcal{G}(\Gamma \setminus \bar{x}); \Theta \vdash \mathbb{A}_{\text{rec}}^{k+1}(\bar{Q}_1)_{g_1} \mid \mathbb{A}_{\text{rec}}^{k+l+1}(\bar{Q}_2)_{g_2} \triangleright \diamond}{\mathcal{G}(\Gamma \setminus \bar{x}); \Theta_1 \cdot \Theta_2 \vdash \mu X.c_k^r(\bar{y}).(\overline{c_{k+1}^r}!\langle \bar{y}_1 \rangle).X \mid c_{k+l+1}^r!\langle \bar{y}_2 \rangle) \mid \mathbb{A}_{\text{rec}}^{k+1}(\bar{Q}_1)_g \mid \mathbb{A}_{\text{rec}}^{k+l+1}(\bar{Q}_2)_\emptyset \triangleright \diamond}}{\mathcal{G}(\Gamma \setminus \bar{x}); \Theta_1 \cdot \Theta_2 \vdash \mu X.c_k^r(\bar{y}).(\overline{c_{k+1}^r}!\langle \bar{y}_1 \rangle).X \mid c_{k+l+1}^r!\langle \bar{y}_2 \rangle) \mid \mathbb{A}_{\text{rec}}^{k+1}(\bar{Q}_1)_g \mid \mathbb{A}_{\text{rec}}^{k+l+1}(\bar{Q}_2)_\emptyset \triangleright \diamond}
 \end{array}$$

- (5) Case $P = (\nu s : C)P'$. We distinguish two sub-cases: (i) $C = S$ and (ii) $C = \langle C' \rangle$. We only consider sub-case (i) as the other is similar. First, we α -convert P as follows:

$$P \equiv_\alpha (\nu s_1 : C)P' \{s_1 \bar{s}_1 / s \bar{s}\}$$

The only can that can be applied is ResS:

$$\text{ResS} \frac{\Gamma; \Delta_\mu \cdot \Delta \cdot s_1 : S \cdot \bar{s}_1 : \bar{S} \vdash P' \{s_1 \bar{s}_1 / s \bar{s}\} \triangleright \diamond}{\Gamma; \Delta_\mu \cdot \Delta \vdash (\nu s_1 : S)P' \{s_1 \bar{s}_1 / s \bar{s}\} \triangleright \diamond} \quad (44)$$

By IH on the assumption of (44) we have:

$$\mathcal{G}(\Gamma \setminus \bar{x}); \Theta' \vdash \mathbb{A}_{\text{rec}}^k \bar{y}'(P' \{s_1 \bar{s}_1 / s \bar{s}\})_g \triangleright \diamond \quad (45)$$

where $\bar{x}' \subseteq \text{fn}(P)$ such that $(\Delta \cdot s_1 : S \cdot \bar{s}_1 : \bar{S}) \setminus \bar{x}' = \emptyset$, and $\bar{y}' = \bar{v}' \cup \bar{m}$ where indexed $_{\Gamma, \Delta \cdot s_1 : S \cdot \bar{s}_1 : \bar{S}}(\bar{v}', \bar{x}')$. Also, $\Theta' = \Theta'_\mu, \Theta_X$ where Θ'_μ such that balanced (Θ'_μ) with

$$\text{dom}(\Theta'_\mu) = \{c_{k+1}^r, c_{k+2}^r, \dots, c_{k+[P]^*+1}^r\} \cup \{\overline{c_{k+2}^r}, \dots, \overline{c_{k+[P]^*+1}^r}\}$$

and $\Theta'_\mu(c_{k+1}^r) = \mu t.?(\bar{N}'); t$ where $\bar{N}' = (\mathcal{G}(\Gamma), \mathcal{G}(\Delta_\mu \cdot \Delta \cdot s_1 : S \cdot \bar{s}_1 : \bar{S}))(\bar{y}')$.

We define $\bar{x} = \bar{x}' \setminus (s_1, \bar{s}_1)$ and $\bar{y} = \bar{y}' \setminus (\bar{s} \cdot \bar{s})$ where $\bar{s} = \text{bn}(s_1 : S)$ and $\bar{s} = \text{bn}(\bar{s}_1 : \bar{S})$. By construction $\bar{x} \subseteq \text{fn}(P)$ and $\Delta \setminus \bar{x} = \emptyset$. Further, $\bar{y} = \bar{v} \cdot \bar{m}$ where indexed $_{\Gamma, \Delta}(\bar{v}, \bar{x})$.

Let $\Theta = \Theta', \Theta''_\mu$ where

$$\Theta''_\mu = c_k^r : \mu t.?(\bar{N}); t, \overline{c_{k+1}^r} : \mu t.!(\bar{N}'); t$$

where $\bar{N} = (\mathcal{G}(\Gamma), \mathcal{G}(\Delta_\mu \cdot \Delta \cdot u_i : S))(\bar{y})$. By construction and since $[P]^* = [P']^* + 1$ we have

$$\text{dom}(\Theta_\mu) = \{c_k^r, c_{k+1}^r, \dots, c_{k+[P]^*-1}^r\} \cup \{\overline{c_{k+1}^r}, \dots, \overline{c_{k+[P]^*-1}^r}\}$$

By Tab. 3 we have:

$$\mathbb{A}_{\text{rec}}^k \bar{y}(P)_g = \mu X.(\nu \bar{s} : \mathcal{G}(C))c_k^r(\bar{y}).\overline{c_{k+1}^r}!\langle \bar{y}' \rangle.X \mid \mathbb{A}_{\text{rec}}^k \bar{y}'(P' \{s_1 \bar{s}_1 / s \bar{s}\})_g$$

We shall prove the following judgment:

$$\mathcal{G}(\Gamma \setminus \bar{x}); \Theta \vdash \mathbb{A}_{\text{rec}}^k \bar{y}(P)_g \triangleright \diamond \quad (46)$$

We use some auxiliary sub-tree:

$$\begin{array}{c}
 \text{Rvar} \frac{\mathcal{G}(\Gamma) \cdot X : \Theta_\mu; \Theta''_\mu \vdash X \triangleright \diamond}{\mathcal{G}(\Gamma) \cdot X : \Theta_\mu; \Theta_\mu \cdot \mathcal{G}(\Delta), \bar{s} : \mathcal{G}(S), \bar{s} : \mathcal{G}(\bar{S}) \vdash \bar{y}' \triangleright \bar{N}'} \\
 \text{PolySend} \frac{\text{PolyVar} \frac{\mathcal{G}(\Gamma) \cdot X : \Theta_\mu; \Theta_\mu \cdot \mathcal{G}(\Delta), \bar{s} : \mathcal{G}(S), \bar{s} : \mathcal{G}(\bar{S}) \vdash \overline{c_{k+1}^r}!\langle \bar{y}' \rangle.X \triangleright \diamond}{\mathcal{G}(\Gamma) \cdot X : \Theta_\mu; \Theta_\mu \cdot \mathcal{G}(\Delta), \bar{s} : \mathcal{G}(S), \bar{s} : \mathcal{G}(\bar{S}) \vdash \bar{y}' \triangleright \bar{N}'}}{\mathcal{G}(\Gamma) \cdot X : \Theta_\mu; \Theta_\mu \cdot \mathcal{G}(\Delta), \bar{s} : \mathcal{G}(S), \bar{s} : \mathcal{G}(\bar{S}) \vdash \bar{y}' \triangleright \bar{N}'}
 \end{array} \quad (47)$$

The following tree proves this case:

$$\begin{array}{c}
 \text{PolyRcv} \frac{\text{PolyVar} \frac{(47) \quad \mathcal{G}(\Gamma); \mathcal{G}(\Delta) \vdash \tilde{y} \triangleright \tilde{N}}{\mathcal{G}(\Gamma); \mathcal{G}(\Delta) \vdash \tilde{y} \triangleright \tilde{N}}}{\mathcal{G}(\Gamma \setminus \tilde{x}) \cdot X : \Theta''_{\mu}; \Theta''_{\mu} \cdot \tilde{s} : \mathcal{G}(S), \tilde{s} : \mathcal{G}(\tilde{S}) \vdash c_k?(y).c_{k+1}!(\tilde{y}').X \triangleright \diamond}}{\mathcal{G}(\Gamma \setminus \tilde{x}) \cdot X : \Theta''_{\mu}; \Theta''_{\mu} \vdash (\nu \tilde{s} : \mathcal{G}(S))c_k?(y).c_{k+1}!(\tilde{y}').X \triangleright \diamond} \\
 \text{PolyResS} \frac{\mathcal{G}(\Gamma \setminus \tilde{x}) \cdot X : \Theta''_{\mu}; \Theta''_{\mu} \vdash (\nu \tilde{s} : \mathcal{G}(S))c_k?(y).c_{k+1}!(\tilde{y}').X \triangleright \diamond}{\mathcal{G}(\Gamma \setminus \tilde{x}); \Theta \vdash \mu X.(\nu \tilde{s} : \mathcal{G}(S))c_k?(y).c_{k+1}!(\tilde{y}').X \triangleright \diamond} \\
 \text{Rec} \frac{\mathcal{G}(\Gamma \setminus \tilde{x}); \Theta \vdash \mu X.(\nu \tilde{s} : \mathcal{G}(S))c_k?(y).c_{k+1}!(\tilde{y}').X \triangleright \diamond}{\mathcal{G}(\Gamma \setminus \tilde{x}); \Theta \vdash \mu X.(\nu \tilde{s} : \mathcal{G}(S))c_k?(y).c_{k+1}!(\tilde{y}').X \triangleright \diamond} \\
 \text{Par} \frac{\mathcal{G}(\Gamma \setminus \tilde{x}); \Theta \vdash \mu X.(\nu \tilde{s} : \mathcal{G}(S))c_k?(y).c_{k+1}!(\tilde{y}').X \triangleright \diamond}{\mathcal{G}(\Gamma \setminus \tilde{x}); \Theta \vdash \mu X.(\nu \tilde{s} : \mathcal{G}(S))c_k?(y).c_{k+1}!(\tilde{y}').X \triangleright \diamond} \quad (45)
 \end{array}$$

This concludes the proof. \square

THEOREM 3 (TYPABILITY OF BREAKDOWN). *Let P be an initialized process. If $\Gamma; \Delta \vdash P \triangleright \diamond$ then*

$$\mathcal{H}^*(\Gamma \setminus \tilde{x}); \mathcal{H}^*(\Delta \setminus \tilde{x}), \Theta \vdash \mathbb{A}_{\tilde{y}}^k(P) \triangleright \diamond \quad (k > 0)$$

where $\tilde{x} \subseteq \text{fn}(P)$ and \tilde{y} such that $\text{indexed}_{\Gamma, \Delta}(\tilde{y}, \tilde{x})$. Also, $\text{balanced}(\Theta)$ with

$$\text{dom}(\Theta) = \{c_k, c_{k+1}, \dots, c_{k+|P|-1}\} \cup \{\overline{c_{k+1}}, \dots, \overline{c_{k+|P|-1}}\}$$

and $\Theta(c_k) = ?(\tilde{M}); \text{end}$, where $\tilde{M} = (\mathcal{H}^*(\Gamma), \mathcal{H}^*(\Delta))(\tilde{y})$.

PROOF. By induction on the structure of P . By assumption $\Gamma; \Delta \vdash P \triangleright \diamond$. In total we consider nine cases. We separately treat Input and Output cases depending on whether a subject name of the prefix is recursive.

(1) Case $P = 0$. The only rule that can be applied here is **Nil**. By inversion of this rule, we have: $\Gamma; \emptyset \vdash 0$. We shall then prove the following judgment:

$$\mathcal{G}(\Gamma); \Theta \vdash \mathcal{B}_{\tilde{y}}^k(0) \triangleright \diamond \quad (48)$$

where $\tilde{x} \subseteq \text{fn}(0) = \epsilon$ and $\Theta = \{c_k : ?(\langle \text{end} \rangle); \text{end}\}$. Since by Rem. 1 we know that $c_k?(y).0$ stands for $c_k?(y).0$ with $c_k : ?(\langle \text{end} \rangle); \text{end}$. By Tab. 3: $\mathcal{B}_{\epsilon}^k(0) = c_k?(y).0$.

The following tree proves this case:

$$\begin{array}{c}
 \text{Nil} \frac{\Gamma'; \emptyset; \emptyset \vdash 0 \triangleright \diamond}{\Gamma'; \emptyset; \emptyset \vdash 0 \triangleright \diamond} \quad c_k \notin \text{dom}(\Gamma) \\
 \text{End} \frac{\Gamma'; \emptyset; \emptyset \vdash 0 \triangleright \diamond}{\Gamma'; \emptyset; c_k : \text{end} \vdash 0 \triangleright \diamond} \quad \text{Sh} \frac{\Gamma'; \emptyset \vdash y \triangleright \langle \text{end} \rangle}{\Gamma'; \emptyset \vdash y \triangleright \langle \text{end} \rangle} \\
 \text{Rev} \frac{\Gamma'; \emptyset; c_k : \text{end} \vdash 0 \triangleright \diamond}{\mathcal{G}(\Gamma), y : \langle \text{end} \rangle; \Theta \vdash c_k?(y).0 \triangleright \diamond}
 \end{array}$$

where $\Gamma' = \mathcal{G}(\Gamma), y : \langle \text{end} \rangle$. We know $c_k \notin \text{dom}(\Gamma)$ since we use reserved names for propagators channels.

(2) Case $P = u_i?(z).P'$. We distinguish two sub-cases, depending on whether u_i is linear or not: (i) $u_i \in \text{dom}(\Delta)$ and (ii) $u_i \in \text{dom}(\Gamma)$. We consider sub-case (i) first. For this case Rule **Rcv** can be applied:

$$\text{Rcv} \frac{\Gamma; \Delta, u_i : S, \Delta_z \vdash P' \triangleright \diamond \quad \Gamma; \Delta_z \vdash z \triangleright C}{\Gamma \setminus z; \Delta, u_i : ?(C); S \vdash u_i?(z).P' \triangleright \diamond} \quad (49)$$

By IH on the first assumption of (49) we know:

$$\mathcal{G}(\Gamma'_1); \mathcal{G}(\Delta'_1), \Theta_1 \vdash \mathcal{B}_{\tilde{y}'}^{k+1}(P') \triangleright \diamond \quad (50)$$

where $\tilde{x}' \subseteq \text{fn}(P')$ and \tilde{y}' such that $\text{indexed}_{\Gamma, \Delta}(\tilde{y}', \tilde{x}')$. Also, $\Gamma'_1 = \Gamma \setminus \tilde{x}'$, $\Delta'_1 = \Delta \setminus \tilde{x}'$, and $\text{balanced}(\Theta_1)$ with

$$\text{dom}(\Theta_1) = \{c_{k+1}, \dots, c_{k+|P'|}\} \cup \{\overline{c_{k+2}}, \dots, \overline{c_{k+|P'|}}\}$$

and $\Theta_1(c_{k+1}) = ?(\tilde{M}')$; end where $\tilde{M}' = (\mathcal{G}(\Gamma), \mathcal{G}(\Delta, u_i : S, \Delta_z))(\tilde{y}')$.

By applying Lem. C.2 to the second assumption of (49) we have:

$$\mathcal{G}(\Gamma); \mathcal{G}(\Delta_z) \vdash \tilde{z} \triangleright \mathcal{G}(C) \quad (51)$$

Let $\tilde{x} = \tilde{x}'$, $u \setminus z$ and $\tilde{y} = \tilde{y}' \sigma$, $u_i \setminus \tilde{z}$ such that $|\tilde{z}| = \mathcal{G}(C)$ where $\sigma = \{\tilde{n}/\tilde{u}\}$ with $\tilde{n} = (u_{i+1}, \dots, u_{i+|\mathcal{G}(S)|})$ and $\tilde{u} = (u_i, \dots, u_{i+|\mathcal{G}(S)|-1})$. We may notice that by Def. 4.12 $\text{indexed}_{\Gamma, \Delta}(\tilde{y}, \tilde{x})$ holds. We define $\Theta = \Theta_1, \Theta'$, where

$$\Theta' = c_k : ?(\tilde{M}); \text{end}, \overline{c_{k+1}} : !(\tilde{M}'); \text{end}$$

with $\tilde{M} = (\mathcal{G}(\Gamma), \mathcal{G}(\Delta, u_i : ?(C); S))(\tilde{y})$. By Def. 4.5, $[P]^* = [P']^* + 1$ so

$$\text{dom}(\Theta) = \{c_k, \dots, c_{k+|P|^*}\} \cup \{\overline{c_{k+1}}, \dots, \overline{c_{k+|P|^*}}\}$$

and Θ is balanced since $\Theta(c_{k+1})$ dual $\Theta(\overline{c_{k+1}})$ and Θ_1 is balanced. By Tab. 3:

$$\mathcal{B}_{\tilde{y}}^k(u_i?(z).P') = c_k?(\tilde{y}).u_i?(\tilde{z}).\overline{c_{k+1}}!\langle\tilde{y}'\sigma\rangle.\mathbf{0} \mid \mathcal{B}_{\tilde{y}'\sigma}^{k+1}(P'\{u_{i+1}/u_i\})$$

Also, let $\Gamma_1 = \Gamma \setminus \tilde{x}$ and $\Delta_1 = \Delta \setminus \tilde{x}$. We may notice that $\Delta_1 = \Delta'_1$. We shall prove the following judgment:

$$\mathcal{G}(\Gamma_1 \setminus z); \mathcal{G}(\Delta_1), \Theta \vdash \mathcal{B}_{\tilde{y}}^k(u_i?(z).P')$$

We type sub-process $c_k?(\tilde{y}).u_i?(\tilde{z}).\overline{c_{k+1}}!\langle\tilde{y}'\sigma\rangle.\mathbf{0}$ with some auxiliary derivations:

$$\text{End} \frac{\text{Nil} \overline{\mathcal{G}(\Gamma); \emptyset \vdash \mathbf{0} \triangleright \diamond}}{\mathcal{G}(\Gamma); \overline{c_{k+1}} : \text{end} \vdash \mathbf{0} \triangleright \diamond} \quad (52)$$

$$\text{PolySend} \frac{(52) \quad \text{PolyVar} \overline{\mathcal{G}(\Gamma); \mathcal{G}(\Delta \setminus \Delta_1, u_{i+1} : S), \mathcal{G}(\Delta_2) \vdash \tilde{y}'\sigma \triangleright \tilde{M}'}}{\mathcal{G}(\Gamma); \overline{c_{k+1}} : !\langle\tilde{M}'\rangle; \text{end}, \mathcal{G}(\Delta \setminus \Delta_1, u_{i+1} : S), \mathcal{G}(\Delta_2) \vdash \overline{c_{k+1}}!\langle\tilde{y}'\sigma\rangle.\mathbf{0} \triangleright \diamond} \quad (53)$$

$$\text{End} \frac{\mathcal{G}(\Gamma); \overline{c_{k+1}} : !\langle\tilde{M}'\rangle; \text{end}, u_i : \text{end}, \mathcal{G}(\Delta \setminus \Delta_1, u_{i+1} : S), \mathcal{G}(\Delta_2) \vdash \overline{c_{k+1}}!\langle\tilde{y}'\sigma\rangle.\mathbf{0} \triangleright \diamond}{\mathcal{G}(\Gamma); \overline{c_{k+1}} : !\langle\tilde{M}'\rangle; \text{end}, c_k : \text{end}, \mathcal{G}(\Delta_2) \vdash u_i?(\tilde{z}).\overline{c_{k+1}}!\langle\tilde{y}'\sigma\rangle.\mathbf{0} \triangleright \diamond} \quad (54)$$

$$\text{PolyRev} \frac{(53) \quad (51) \quad \mathcal{G}(\Gamma \setminus z); u_i : ?(\mathcal{G}(U)); \text{end}, \overline{c_{k+1}} : !\langle\tilde{M}'\rangle; \text{end}, \mathcal{G}(\Delta_2) \vdash u_i?(\tilde{z}).\overline{c_{k+1}}!\langle\tilde{y}'\sigma\rangle.\mathbf{0} \triangleright \diamond}{\mathcal{G}(\Gamma \setminus z); \overline{c_{k+1}} : !\langle\tilde{M}'\rangle; \text{end}, c_k : \text{end}, \mathcal{G}(\Delta_2) \vdash u_i?(\tilde{z}).\overline{c_{k+1}}!\langle\tilde{y}'\sigma\rangle.\mathbf{0} \triangleright \diamond} \quad (55)$$

$$\text{PolyRev} \frac{(54) \quad \text{PolyVar} \overline{\mathcal{G}(\Gamma \setminus z); \mathcal{G}(\Delta_2) \vdash \tilde{y} \triangleright \tilde{M}'}}{\mathcal{G}(\Gamma_1 \setminus z); \Theta' \vdash c_k?(\tilde{y}).u_i?(z).\overline{c_{k+1}}!\langle\tilde{y}'\sigma\rangle.\mathbf{0} \triangleright \diamond} \quad (56)$$

where $\Delta_2 = \Delta, u_i : ?(C); S \setminus \Delta_1$. Using (55), the following tree proves this case:

$$\text{Par} \frac{(50) \quad \mathcal{G}(\Gamma'_1 \setminus z); \mathcal{G}(\Delta_1), \Theta_1 \vdash \mathcal{B}_{\tilde{y}'\sigma}^{k+1}(P'\{u_{i+1}/u_i\}) \triangleright \diamond}{\mathcal{G}(\Gamma_1 \setminus z); \mathcal{G}(\Delta_1), \Theta \vdash c_k?(\tilde{y}).u_i?(\tilde{z}).\overline{c_{k+1}}!\langle\tilde{y}'\sigma\rangle.\mathbf{0} \mid \mathcal{B}_{\tilde{y}'\sigma}^{k+1}(P'\{u_{i+1}/u_i\}) \triangleright \diamond} \quad (56)$$

Note that we have used the following for the right assumption of (56):

$$\mathcal{B}_{\tilde{y}'\sigma}^{k+1}(P') \equiv_{\alpha} \mathcal{B}_{\tilde{y}'\sigma}^{k+1}(P'\{u_{i+1}/u_i\})$$

Next, we comment the case when $u_i \notin \tilde{x}$. In this case $\Delta_1 = \Delta \setminus \tilde{x}, u_i : ?(C); S$. Hence, in the right hand-side of (56) instead of $\mathcal{G}(\Delta_1)$ we would have $\mathcal{G}(\Delta \setminus \tilde{x}, u_{i+1} : S)$ and in the left-hand side we have $u_i : ?(\mathcal{G}(C)); \text{end}$ as a linear environment. Then, we would need to apply (C.1) with $\{u_i/u_{i+1}\}$ to right-hand side before invoking (50). We remark that similar provisos apply to the following cases when the assumption is $u_i \notin \tilde{x}$.

This concludes sub-case (i). We now consider sub-case (ii), i.e., $u_i \in \text{dom}(\Gamma)$. Here Rule Acc can be applied:

$$\text{Acc} \frac{\Gamma; \emptyset \vdash u_i \triangleright \langle C \rangle \quad \Gamma; \Delta, z : C \vdash P' \triangleright \diamond \quad \Gamma; z : C \vdash z \triangleright C}{\Gamma; \Delta \vdash u_i?(z).P' \triangleright \diamond} \quad (57)$$

By IH on the second assumption of (57) we have:

$$\mathcal{G}(\Gamma'_1); \mathcal{G}(\Delta'_1), \Theta_1 \vdash \mathcal{B}_{\tilde{y}'\sigma}^{k+1}(P') \triangleright \diamond \quad (58)$$

where \tilde{x}' and \tilde{y}' are as in sub-case (i). Also, $\Gamma'_1 = \Gamma \setminus \tilde{x}'$ and $\Delta'_1 = \Delta \setminus \tilde{y}'$ and balanced(Θ_1) with

$$\text{dom}(\Theta_1) = \{c_{k+1}, \dots, c_{k+\lfloor P' \rfloor^*}\} \cup \{\overline{c_{k+2}}, \dots, \overline{c_{k+\lfloor P' \rfloor^*}}\}$$

and $\Theta_1(c_{k+1}) = ?(\tilde{M}')$; end where $\tilde{M}' = (\mathcal{G}(\Gamma), \mathcal{G}(\Delta, z : C))(\tilde{y}')$.

By applying Lem. C.2 to the first and third assumptions of (57) we have:

$$\mathcal{G}(\Gamma); \emptyset \vdash u_i \triangleright \langle \mathcal{G}(C) \rangle \quad (59)$$

$$\mathcal{G}(\Gamma); \mathcal{G}(z : C) \vdash \tilde{z} \triangleright \mathcal{G}(C) \quad (60)$$

We define $\tilde{x} = \tilde{x}' \cup u \setminus z$ and $\tilde{y} = \tilde{y}' \cup u_i \setminus \tilde{z}$ where $|\tilde{z}| = |\mathcal{G}(C)|$. Notice that $\text{indexed}_{\Gamma, \Delta}(\tilde{y}, \tilde{x})$. Let $\Gamma_1 = \Gamma \setminus x$. We define $\Theta = \Theta_1, \Theta'$, where

$$\Theta' = c_k : ?(\tilde{M}); \text{end}, \overline{c_{k+1}} : !(\tilde{M}'); \text{end}$$

with $\tilde{M} = (\mathcal{G}(\Gamma), \mathcal{G}(\Delta))(\tilde{y})$. By Def. 4.5, $[P]^* = [P']^* + 1$ so

$$\text{dom}(\Theta) = \{c_k, \dots, c_{k+[P]^*-1}\} \cup \{\overline{c_{k+1}}, \dots, \overline{c_{k+[P]^*-1}}\}$$

and Θ is balanced since $\Theta(c_{k+1})$ dual $\Theta(\overline{c_{k+1}})$ and Θ_1 is balanced.

By Tab. 3, we have:

$$\mathcal{B}_{\tilde{y}}^k(u_i?(z).P') = c_k?(y).u_i?(z).\overline{c_{k+1}}!(\tilde{y}').\mathbf{0} \mid \mathcal{B}_{\tilde{y}'}^{k+1}(P') \quad (61)$$

We shall prove the following judgment:

$$\mathcal{G}(\Gamma_1); \mathcal{G}(\Delta_1), \Theta \vdash \mathcal{B}_{\tilde{y}}^k(u_i?(z).P') \triangleright \diamond \quad (62)$$

To this end, we use some auxiliary derivations:

$$\text{End} \frac{\text{Nil} \frac{\mathcal{G}(\Gamma); \mathbf{0}; \mathbf{0} \vdash \mathbf{0} \triangleright \diamond}{\mathcal{G}(\Gamma); \mathbf{0}; \mathbf{0} \vdash \mathbf{0} \triangleright \diamond}}{\mathcal{G}(\Gamma); \mathbf{0}; \overline{c_{k+1}} : \text{end} \vdash \mathbf{0} \triangleright \diamond} \quad (63)$$

$$\text{PolySend} \frac{\text{PolyVar} \frac{\mathcal{G}(\Gamma); \mathcal{G}(\Delta_2), z : \mathcal{G}(C) \vdash \tilde{y}' \triangleright \tilde{M}'}{\mathcal{G}(\Gamma); \mathcal{G}(\Delta_2), z : \mathcal{G}(C) \vdash \overline{c_{k+1}}!(\tilde{y}').\mathbf{0} \triangleright \diamond}}{\mathcal{G}(\Gamma); \overline{c_{k+1}} : !(\tilde{M}'); \text{end}, \mathcal{G}(\Delta_2), z : \mathcal{G}(C) \vdash \overline{c_{k+1}}!(\tilde{y}').\mathbf{0} \triangleright \diamond} \quad (64)$$

$$\text{PolyAcc} \frac{\text{PolyAcc} \frac{\mathcal{G}(\Gamma); \overline{c_{k+1}} : !(\tilde{M}'); \text{end}, \mathcal{G}(\Delta_2) \vdash u_i?(z).\overline{c_{k+1}}!(\tilde{y}').\mathbf{0} \triangleright \diamond}{\mathcal{G}(\Gamma); \overline{c_{k+1}} : !(\tilde{M}'); \text{end}, c_k : \text{end}, \mathcal{G}(\Delta_2) \vdash u_i?(z).\overline{c_{k+1}}!(\tilde{y}').\mathbf{0} \triangleright \diamond}}{\mathcal{G}(\Gamma); \overline{c_{k+1}} : !(\tilde{M}'); \text{end}, c_k : \text{end}, \mathcal{G}(\Delta_2) \vdash u_i?(z).\overline{c_{k+1}}!(\tilde{y}').\mathbf{0} \triangleright \diamond} \quad (65)$$

$$\text{PolyRev} \frac{\text{PolyVar} \frac{\mathcal{G}(\Gamma); \mathcal{G}(\Delta_2) \vdash \tilde{y} \triangleright \tilde{M}}{\mathcal{G}(\Gamma); \mathcal{G}(\Delta_2) \vdash \tilde{y} \triangleright \tilde{M}}}{\mathcal{G}(\Gamma_1); \Theta' \vdash c_k?(y).u_i?(z).\overline{c_{k+1}}!(\tilde{y}').\mathbf{0} \triangleright \diamond} \quad (66)$$

where $\Delta_2 = \Delta \setminus \Delta_1$. Using (58) and (66), the following tree proves this sub-case:

$$\text{Par} \frac{\text{Par} \frac{\mathcal{G}(\Gamma_1); \mathcal{G}(\Delta_1), \Theta \vdash c_k?(y).u_i?(z).\overline{c_{k+1}}!(\tilde{y}').\mathbf{0} \mid \mathcal{B}_{\tilde{y}'}^{k+1}(P') \triangleright \diamond}{\mathcal{G}(\Gamma_1); \mathcal{G}(\Delta_1), \Theta \vdash c_k?(y).u_i?(z).\overline{c_{k+1}}!(\tilde{y}').\mathbf{0} \mid \mathcal{B}_{\tilde{y}'}^{k+1}(P') \triangleright \diamond}}{\mathcal{G}(\Gamma_1); \mathcal{G}(\Delta_1), \Theta \vdash c_k?(y).u_i?(z).\overline{c_{k+1}}!(\tilde{y}').\mathbf{0} \mid \mathcal{B}_{\tilde{y}'}^{k+1}(P') \triangleright \diamond} \quad (67)$$

Note that if $u \notin \text{fn}(P')$ we need to apply Lem. C.4 with u_i to the right assumption of (67) before applying (58). This concludes the analysis of the input case.

- (3) Case $P = u_i!(z_j).P'$. We distinguish two sub-cases: (i) $u_i \in \text{dom}(\Delta)$ and (ii) $u_i \in \text{dom}(\Gamma)$. We consider sub-case (i) first. For this case Rule SEND can be applied:

$$\text{Send} \frac{\Gamma; \Delta_1, u_i : S \vdash P' \triangleright \diamond \quad \Gamma; \Delta_z \vdash z_j \triangleright C \quad u_i : S \in \Delta}{\Gamma; \Delta \vdash u_i!(z_j).P' \triangleright \diamond} \quad (68)$$

where $\Delta = \Delta_1, \Delta_z, u_i : !\langle C \rangle; S$.

By IH on the first assumption of (68) we have:

$$\mathcal{G}(\Gamma'_1); \mathcal{G}(\Delta'_1), \Theta_1 \vdash \mathcal{B}_{\tilde{y}'}^{k+1}(P') \triangleright \diamond \quad (69)$$

where $\tilde{x}' \subseteq \text{fv}(P')$ and \tilde{y}' such that $\text{indexed}_{\Gamma, \Delta_1, u_i : S}(\tilde{y}', \tilde{x}')$. Also, $\Gamma'_1 = \Gamma \setminus \tilde{y}'$, $\Delta'_1 = \Delta_1 \setminus \tilde{y}'$, and $\text{balanced}(\Theta_1)$ with

$$\text{dom}(\Theta_1) = \{c_{k+1}, \dots, c_{k+[P']^*}\} \cup \{\overline{c_{k+2}}, \dots, \overline{c_{k+[P']^*}}\}$$

and $\Theta_1(c_{k+1}) = ?(\tilde{M}_1); \text{end}$ where $\tilde{M}_1 = (\mathcal{G}(\Gamma), \mathcal{G}(\Delta_1, u_i : S))(\tilde{y}')$.

By Lem. C.2 and the first assumption of (68) we have:

$$\mathcal{G}(\Gamma); \mathcal{G}(\Delta_z) \vdash \tilde{z} \triangleright \mathcal{G}(C) \quad (70)$$

where $\tilde{z} = (z_j, \dots, z_{j+|\mathcal{G}(C)|-1})$. We assume $\tilde{x} = \tilde{x}', u, z$. Since $\tilde{x}' \subseteq \text{fn}(P')$ follows that $\tilde{x} \subseteq \text{fn}(P)$. Let $\tilde{y} = \tilde{y}'\sigma, u_i, \tilde{z}$ where $\sigma = \{\tilde{n}/\tilde{u}\}$ with $\tilde{n} = (u_{i+1}, \dots, u_{i+|\mathcal{G}(S)|})$ and $\tilde{u} = (u_i, \dots, u_{i+|\mathcal{G}(S)|-1})$. We have $\tilde{z} = \text{fnb}(z, \tilde{y})$. By Def. 4.12 follows that $\text{indexed}_{\Gamma, \Delta}(\tilde{y}, \tilde{x})$. We define $\Theta = \Theta_1, \Theta'$, where:

$$\Theta' = c_k : ?(\tilde{M}); \text{end}, \overline{c_{k+1}} : !(\tilde{M}_1); \text{end}$$

with $\tilde{M} = (\mathcal{G}(\Gamma), \mathcal{G}(\Delta))(\tilde{y})$. By Def. 4.5, we know $\lfloor P \rfloor^* = \lfloor P' \rfloor^* + 1$, so

$$\text{dom}(\Theta) = \{c_k, \dots, c_{k+\lfloor P \rfloor^* - 1}\} \cup \{\overline{c_{k+1}}, \dots, \overline{c_{k+\lfloor P \rfloor^* - 1}}\}$$

and Θ is balanced since $\Theta(c_{k+1})$ dual $\Theta(\overline{c_{k+1}})$ and Θ_1 is balanced.

By Tab. 3, we have:

$$\mathcal{B}_{\tilde{y}}^k(u_i! \langle z \rangle . P') = c_k?(\tilde{y}).u_i! \langle \tilde{z} \rangle . \overline{c_{k+1}}! \langle \tilde{y}' \sigma \rangle . \mathbf{0} \mid \mathcal{B}_{\tilde{y}'\sigma}^{k+1}(P' \{u_{i+1}/u_i\})$$

Let $\Gamma_1 = \Gamma \setminus \tilde{x} = \Gamma'_1$ and $\Delta \setminus \tilde{x} = \Delta'_1$.

We shall prove the following judgment:

$$\mathcal{G}(\Gamma_1); \mathcal{G}(\Delta'_1), \Theta \vdash \mathcal{B}_{\tilde{y}}^k(u_i! \langle z \rangle . P') \triangleright \diamond \quad (71)$$

To type the sub-process $c_k?(\tilde{y}).u_i! \langle \tilde{z} \rangle . \overline{c_{k+1}}! \langle \tilde{y}' \rangle . \mathbf{0}$ of $\mathcal{B}_{\tilde{y}}^k(u_i! \langle z \rangle . P')$ we use some auxiliary derivations:

$$\text{End} \frac{\text{Nil} \frac{\mathcal{G}(\Gamma); \emptyset \vdash \mathbf{0} \triangleright \diamond}{\mathcal{G}(\Gamma); \overline{c_{k+1}} : \text{end} \vdash \mathbf{0} \triangleright \diamond}}{\mathcal{G}(\Gamma); \overline{c_{k+1}} : \text{end} \vdash \mathbf{0} \triangleright \diamond} \quad (72)$$

$$\text{PolySend} \frac{(72) \quad \text{PolyVar} \frac{\mathcal{G}(\Gamma); \mathcal{G}(\Delta_1, u_{i+1} : S \setminus \Delta'_1) \vdash \tilde{y}'\sigma \triangleright \tilde{M}_1}{\mathcal{G}(\Gamma); \overline{c_{k+1}} : ! \langle \tilde{M}_1 \rangle; \text{end}, \mathcal{G}(\Delta_1, u_{i+1} : S \setminus \Delta'_1) \vdash \overline{c_{k+1}}! \langle \tilde{y}' \sigma \rangle . \triangleright \diamond}}{\mathcal{G}(\Gamma); \overline{c_{k+1}} : ! \langle \tilde{M}_1 \rangle; \text{end}, u_i : \text{end}, \mathcal{G}(\Delta_1, u_{i+1} : S \setminus \Delta'_1) \vdash \overline{c_{k+1}}! \langle \tilde{y}' \sigma \rangle . \triangleright \diamond} \quad (73)$$

$$\text{Send} \frac{(73) \quad (70)}{\mathcal{G}(\Gamma); \mathcal{G}(\Delta \setminus \Delta'_1), \overline{c_{k+1}} : ! \langle \tilde{M}_1 \rangle; \text{end} \vdash u_i! \langle \tilde{z} \rangle . \overline{c_{k+1}}! \langle \tilde{y}' \rangle . \mathbf{0} \triangleright \diamond} \quad (74)$$

$$\text{End} \frac{\mathcal{G}(\Gamma); \mathcal{G}(\Delta \setminus \Delta'_1), \overline{c_{k+1}} : ! \langle \tilde{M}_1 \rangle; \text{end}, c_k : \text{end} \vdash u_i! \langle \tilde{z} \rangle . \overline{c_{k+1}}! \langle \tilde{y}' \rangle . \mathbf{0} \triangleright \diamond}{\mathcal{G}(\Gamma); \mathcal{G}(\Delta \setminus \Delta'_1) \vdash \tilde{y} : \tilde{M}} \quad (75)$$

Using (69) and (75), the following tree proves this case:

$$\text{Par} \frac{(75) \quad \frac{(69)}{\mathcal{G}(\Gamma_1); \mathcal{G}(\Delta'_1), \Theta_1 \vdash \mathcal{B}_{\tilde{y}'\sigma}^{k+1}(P' \{u_{i+1}/u_i\}) \triangleright \diamond}}{\mathcal{G}(\Gamma_1); \mathcal{G}(\Delta'_1), \Theta \vdash \mathcal{B}_{\tilde{y}}^k(u_i! \langle z \rangle . P') \triangleright \diamond}}{\mathcal{G}(\Gamma_1); \mathcal{G}(\Delta'_1), \Theta \vdash \mathcal{B}_{\tilde{y}}^k(u_i! \langle z \rangle . P') \triangleright \diamond} \quad (76)$$

Note that we have used the following for the right assumption of (56):

$$\mathcal{B}_{\tilde{y}}^{k+1}(P') \equiv_{\alpha} \mathcal{B}_{\tilde{y}'\sigma}^{k+1}(P' \{u_{i+1}/u_i\})$$

We now consider sub-case (ii). For this sub-case Rule REQ can be applied:

$$\text{Req} \frac{\Gamma; \emptyset \vdash u \triangleright \langle C \rangle \quad \Gamma; \Delta_1 \triangleright P' \triangleright \diamond \quad \Gamma; \Delta_z \vdash z \triangleright C}{\Gamma; \Delta_1, \Delta_z \vdash u_i! \langle z \rangle . P' \triangleright \diamond} \quad (77)$$

Let $\tilde{x}' \subseteq \text{fn}(P')$ and \tilde{y} such that $\text{indexed}_{\Gamma, \Delta_1}(\tilde{y}', \tilde{x}')$. Further, let $\Gamma'_1 = \Gamma \setminus \tilde{x}'$ and $\Delta'_1 = \Delta_1 \setminus \tilde{x}'$. Also, let Θ_1 be environment defined as in sub-case (i).

By IH on the second assumption of (77) we have:

$$\mathcal{G}(\Gamma'_1); \mathcal{G}(\Delta'_1), \Theta_1 \vdash \mathcal{B}_{\tilde{y}'\sigma}^{k+1}(P') \triangleright \diamond \quad (78)$$

By Lem. C.2 and the first and third assumptions of (77) we have:

$$\mathcal{G}(\Gamma); \emptyset \vdash u_i \triangleright \mathcal{G}(\langle C \rangle) \quad (79)$$

$$\mathcal{G}(\Gamma); \mathcal{G}(\Delta_z) \vdash \tilde{z} \triangleright \mathcal{G}(C) \quad (80)$$

We define $\tilde{x} = \tilde{x}' \cup z \cup u$ and $\tilde{y} = \tilde{y}' \cup \tilde{z} \cup u_i$ where $|\tilde{z}| = |\mathcal{G}(C)|$. Notice that $\text{indexed}_{\Gamma, \Delta}(\tilde{y}, \tilde{x})$. Let $\Gamma_1 = \Gamma \setminus \tilde{x} = \Gamma'_1 \setminus u$ and $\Delta \setminus \tilde{x} = \Delta'_1$. By Tab. 3, we have:

$$\mathcal{B}_{\tilde{y}}^k(u_i!(z).P') = c_k?(y).u_i!(z).\overline{c_{k+1}}!(y').\mathbf{0} \mid \mathcal{B}_{\tilde{y}'}^{k+1}(P')$$

We shall prove the following judgment:

$$\mathcal{G}(\Gamma_1); \mathcal{G}(\Delta'_1), \Theta \vdash \mathcal{B}_{\tilde{y}}^k(u_i!(z).P')$$

We use some auxiliary derivations:

$$\text{PolySend} \frac{\text{End} \frac{\text{Nil} \frac{\mathcal{G}(\Gamma); \emptyset; \mathbf{0} \vdash \mathbf{0} \triangleright \diamond}{\mathcal{G}(\Gamma); \emptyset; \mathbf{0} \vdash \mathbf{0} \triangleright \diamond}}{\mathcal{G}(\Gamma); \emptyset; \overline{c_{k+1}} : \text{end} \vdash \mathbf{0} \triangleright \diamond}}{\mathcal{G}(\Gamma); \mathcal{G}(\Delta_1 \setminus \Delta'_1) \vdash \tilde{y}' : \tilde{M}_1}}{\mathcal{G}(\Gamma); \mathcal{G}(\Delta_1 \setminus \Delta'_1), \overline{c_{k+1}} :!(\tilde{M}_1); \text{end} \vdash \overline{c_{k+1}}!(\tilde{y}').\mathbf{0} \triangleright \diamond}} \quad (81)$$

$$\text{PolyReq} \frac{(79) \quad (81) \quad (80)}{\mathcal{G}(\Gamma); \mathcal{G}(\Delta_1 \setminus \Delta'_1), \overline{c_{k+1}} :!(\tilde{M}_1); \text{end}, \tilde{z} : \mathcal{G}(C) \vdash u_i!(z).\overline{c_{k+1}}!(\tilde{y}').\mathbf{0} \triangleright \diamond} \quad (82)$$

$$\text{PolyRcv} \frac{(82) \quad \text{PolyVar} \frac{\mathcal{G}(\Gamma); \Delta_1 \setminus \Delta'_1, \tilde{z} : \mathcal{G}(C) \vdash \tilde{y} \triangleright \tilde{M}}{\mathcal{G}(\Gamma_1); \Theta' \vdash c_k?(y).u_i!(z).\overline{c_{k+1}}!(\tilde{y}').\mathbf{0} \triangleright \diamond}}{\mathcal{G}(\Gamma_1); \Theta' \vdash c_k?(y).u_i!(z).\overline{c_{k+1}}!(\tilde{y}').\mathbf{0} \triangleright \diamond}} \quad (83)$$

The following tree proves this case:

$$\text{Par} \frac{(83) \quad \frac{(78)}{\mathcal{G}(\Gamma_1); \mathcal{G}(\Delta'_1), \Theta_1 \vdash \mathcal{B}_{\tilde{y}'}^{k+1}(P') \triangleright \diamond}}{\mathcal{G}(\Gamma_1); \mathcal{G}(\Delta'_1), \Theta \vdash \mathcal{B}_{\tilde{y}}^k(u_i!(z).P') \triangleright \diamond}}{\mathcal{G}(\Gamma_1); \mathcal{G}(\Delta'_1), \Theta \vdash \mathcal{B}_{\tilde{y}}^k(u_i!(z).P') \triangleright \diamond}} \quad (84)$$

We remark that if $u_i \notin \text{fn}(P')$ we need to apply Lem. C.4 with u_i to the right assumption of (84) before applying (78). This concludes the analysis for the output case $P = u_i!(z).P'$.

(4) Case $P = (\nu s : C)P'$. We distinguish two sub-cases: (i) $C = S$ and (ii) $C = \langle C \rangle$. First, we α -convert P as follows:

$$P \equiv_{\alpha} (\nu s_1 : C)P' \{s_1 \bar{s}_1 / s \bar{s}\}$$

We consider sub-case (i) first. For this case Rule ResS can be applied:

$$\text{ResS} \frac{\Gamma; \Delta, s_1 : S, \bar{s}_1 : \bar{S} \vdash P' \{s_1 \bar{s}_1 / s \bar{s}\} \triangleright \diamond}{\Gamma; \Delta \vdash (\nu s_1 : S)P' \{s_1 \bar{s}_1 / s \bar{s}\} \triangleright \diamond} \quad (85)$$

By IH on the assumption of (85) we have:

$$\mathcal{G}(\Gamma \setminus \tilde{x}); \mathcal{G}(\Delta \setminus \tilde{x}, s_1 : S, \bar{s}_1 : \bar{S}), \Theta_1 \vdash \mathcal{B}_{\tilde{y}}^k(P' \{s_1 \bar{s}_1 / s \bar{s}\}) \triangleright \diamond \quad (86)$$

where $\tilde{x} \subseteq \text{fn}(P')$ such that $s_1, \bar{s}_1 \notin \tilde{x}$ and \tilde{y} such that $\text{indexed}_{\Gamma, \Delta}(\tilde{y}, \tilde{x})$. Also, $\text{balanced}(\Theta_1)$ with

$$\text{dom}(\Theta_1) = \{c_k, \dots, c_{k+\lfloor P' \rfloor^* - 1}\} \cup \{\overline{c_{k+1}}, \dots, \overline{c_{k+\lfloor P' \rfloor^* - 1}}\}$$

and $\Theta_1(c_k) = ?(\tilde{M}); \text{end}$ with $\tilde{M} = (\mathcal{G}(\Gamma), \mathcal{G}(\Delta))(\tilde{y})$.

Note that we take $\Theta = \Theta_1$ since $\lfloor P \rfloor^* = \lfloor P' \rfloor^*$. By Def. 4.1 and Def. 4.4 and (86), we know that:

$$\mathcal{G}(\Gamma \setminus \tilde{x}); \mathcal{G}(\Delta \setminus \tilde{x}), \tilde{s} : \mathcal{G}(S), \bar{\tilde{s}} : \mathcal{G}(\bar{S}) \vdash \mathcal{B}_{\tilde{y}}^k(P' \{s_1 \bar{s}_1 / s \bar{s}\}) \triangleright \diamond \quad (87)$$

where $\tilde{s} = (s_1, \dots, s_{|\mathcal{G}(S)|})$ and $\bar{\tilde{s}} = (\bar{s}_1, \dots, \bar{s}_{|\mathcal{G}(S)|})$. By Tab. 3, we have:

$$\mathcal{B}_{\tilde{y}}^k((\nu s)P') = (\nu \tilde{s} : \mathcal{G}(S))\mathcal{B}_{\tilde{y}}^k(P' \{s_1 \bar{s}_1 / s \bar{s}\})$$

The following tree proves this sub-case:

$$\text{PolyResS} \frac{(87)}{\mathcal{G}(\Gamma \setminus \tilde{x}); \mathcal{G}(\Delta \setminus \tilde{x}) \vdash (\nu \tilde{s} : \mathcal{G}(S))\mathcal{B}_{\tilde{y}}^k(P' \{s_1 \bar{s}_1 / s \bar{s}\}) \triangleright \diamond} \quad (88)$$

We now consider sub-case (ii). Similarly to sub-case (i) we first α -convert P as follows:

$$P \equiv_{\alpha} (\nu s_1)P' \{s_1 / s\}$$

For this sub-case Rule Res can be applied:

$$\text{Res} \frac{\Gamma, s_1 : \langle C \rangle; \Delta \vdash P' \{s_1 / s\} \triangleright \diamond}{\Gamma; \Delta \vdash (\nu s_1)P' \{s_1 / s\} \triangleright \diamond} \quad (89)$$

By IH on the first assumption of (89) we have:

$$\mathcal{G}(\Gamma \setminus \tilde{x}, s_1 : \langle C \rangle); \mathcal{G}(\Delta \setminus \tilde{x}), \Theta_1 \vdash \mathcal{B}_{\tilde{y}}^k(P' \{s_1/s\}) \triangleright \diamond \quad (90)$$

where $\tilde{x} \subseteq \text{fn}(P')$ such that $s_1 \notin \tilde{x}$ and \tilde{y} such that $\text{indexed}_{\Gamma, \Delta}(\tilde{y}, \tilde{x})$. Also, $\text{balanced}(\Theta_1)$ with

$$\text{dom}(\Theta_1) = \{c_k, \dots, c_{k+\lfloor P' \rfloor^* - 1}\} \cup \{\overline{c_{k+1}}, \dots, \overline{c_{k+\lfloor P' \rfloor^* - 1}}\}$$

and $\Theta_1(c_k) = ?(\tilde{M})$; end with $\tilde{M} = (\mathcal{G}(\Gamma), \mathcal{G}(\Delta))(\tilde{y})$.

Here we also take $\Theta = \Theta_1$ since $\lfloor P \rfloor^* = \lfloor P' \rfloor^*$. We notice that by Def. 4.1 and Def. 4.4 and (90):

$$\mathcal{G}(\Gamma \setminus \tilde{x}, s_1 : \mathcal{G}(\langle C \rangle); \mathcal{G}(\Delta \setminus \tilde{x}), \Theta_1 \vdash \mathcal{B}_{\tilde{y}}^k(P' \{s_1/s\}) \triangleright \diamond \quad (91)$$

By Tab. 3, we have:

$$\mathcal{B}_{\tilde{y}}^k((\nu s)P') = (\nu s_1 : \mathcal{G}(\langle C \rangle)) \mathcal{B}_{\tilde{y}}^k(P' \{s_1/s\})$$

The following tree proves this sub-case:

$$\text{PolyRes} \frac{(91)}{\mathcal{G}(\Gamma \setminus \tilde{x}); \mathcal{G}(\Delta \setminus \tilde{x}), \Theta \vdash (\nu s_1 : \mathcal{G}(\langle C \rangle)) \mathcal{B}_{\tilde{y}}^k(P' \{s_1/s\}) \triangleright \diamond} \quad (92)$$

(5) Case $P = Q \mid R$. For this case only Rule PAR can be applied:

$$\text{Par} \frac{\Gamma; \Delta_1 \vdash Q \triangleright \diamond \quad \Gamma; \Delta_2 \vdash R \triangleright \diamond}{\Gamma; \Delta_1, \Delta_2 \vdash Q \mid R \triangleright \diamond} \quad (93)$$

By IH on the first assumption of (93) we have:

$$\mathcal{G}(\Gamma'_1); \mathcal{G}(\Delta'_1), \Theta_1 \vdash \mathcal{B}_{\tilde{y}_1}^{k+1}(Q) \triangleright \diamond \quad (94)$$

where $\tilde{x}_1 \subseteq \text{fn}(Q)$ and \tilde{y}_1 such that $\text{indexed}_{\Gamma, \Delta_1}(\tilde{y}_1, \tilde{x}_1)$. Also, $\Gamma'_1 = \Gamma \setminus \tilde{x}_1$, $\Delta'_1 = \Delta_1 \setminus \tilde{x}_1$, and $\text{balanced}(\Theta_1)$ with

$$\text{dom}(\Theta_1) = \{c_{k+1}, \dots, c_{k+\lfloor Q \rfloor^*}\} \cup \{\overline{c_{k+2}}, \dots, \overline{c_{k+\lfloor Q \rfloor^*}}\}$$

and $\Theta_1(c_{k+1}) = ?(\tilde{M}_1)$; end with $\tilde{M}_1 = (\mathcal{G}(\Gamma), \mathcal{G}(\Delta_1))(\tilde{y}_1)$.

By IH on the second assumption of (93) we have:

$$\mathcal{G}(\Gamma'_2); \mathcal{G}(\Delta'_2), \Theta_2 \vdash \mathcal{B}_{\tilde{y}_2}^{k+l+1}(R) \triangleright \diamond \quad (95)$$

where $\tilde{x}_2 \subseteq \text{fn}(R)$ and \tilde{y}_2 such that $\text{indexed}_{\Gamma, \Delta_2}(\tilde{y}_2, \tilde{x}_2)$ and $l = |Q|$. Also, $\Gamma'_2 = \Gamma \setminus \tilde{x}_2$, $\Delta'_2 = \Delta_2 \setminus \tilde{x}_2$ and $\text{balanced}(\Theta_2)$ with

$$\text{dom}(\Theta_2) = \{c_{k+l+1}, \dots, c_{k+l+\lfloor R \rfloor^*}\} \cup \{\overline{c_{k+l+2}}, \dots, \overline{c_{k+l+\lfloor R \rfloor^*}}\}$$

and $\Theta_2(c_{k+l+1}) = ?(\tilde{M}_2)$; end with $\tilde{M}_2 = (\mathcal{G}(\Gamma), \mathcal{G}(\Delta_2))(\tilde{y}_2)$.

We define $\tilde{x} = \tilde{x}_1 \cup \tilde{x}_2$. We may notice that $\tilde{x} \subseteq \text{fn}(P)$ since $\text{fn}(P) = \text{fn}(Q) \cup \text{fn}(R)$. Accordingly, we define $\tilde{y} = \tilde{y}_1, \tilde{y}_2$. By definition, $\text{indexed}_{\Gamma, \Delta_1, \Delta_2}(\tilde{y}, \tilde{x})$ holds. Further, let $\tilde{M} = (\mathcal{G}(\Gamma), \mathcal{G}(\Delta_1, \Delta_2))(\tilde{y})$. We define $\Theta = \Theta_1, \Theta_2, \Theta'$ where:

$$\Theta' = c_k : ?(\tilde{M}); \text{end}, \overline{c_{k+1}} : !(\tilde{M}_1); \text{end}, \overline{c_{k+l+1}} : !(\tilde{M}_2); \text{end}$$

By construction Θ is balanced since $\Theta(c_{k+1})$ dual $\Theta(\overline{c_{k+1}})$, $\Theta(c_{k+l+1})$ dual $\Theta(\overline{c_{k+l+1}})$, and Θ_1 and Θ_2 are balanced.

By Tab. 3 we have:

$$\mathcal{B}_{\tilde{y}}^k(Q \mid R) = c_k ?(\tilde{y}). \overline{c_{k+1}} !(\tilde{y}_1). \overline{c_{k+l+1}} !(\tilde{y}_2). \mathbf{0} \mid \mathcal{B}_{\tilde{y}_1}^{k+1}(Q) \mid \mathcal{B}_{\tilde{y}_2}^{k+l+1}(R)$$

We may notice that $\tilde{y}_1 = \text{fnb}(Q, \tilde{y})$ and $\tilde{y}_2 = \text{fnb}(R, \tilde{y})$ hold by the construction of \tilde{y} . Let $\Gamma_1 = \Gamma \setminus \tilde{x}$. We shall prove the following judgment:

$$\mathcal{G}(\Gamma_1); \mathcal{G}(\Delta'_1, \Delta'_2), \Theta \vdash c_k ?(\tilde{y}). \overline{c_{k+1}} !(\tilde{y}_1). \overline{c_{k+l+1}} !(\tilde{y}_2). \mathbf{0} \mid \mathcal{B}_{\tilde{y}_1}^{k+1}(Q) \mid \mathcal{B}_{\tilde{y}_2}^{k+l+1}(R) \triangleright \diamond$$

To type $c_k ?(\tilde{y}). \overline{c_{k+1}} !(\tilde{y}_1). \overline{c_{k+l+1}} !(\tilde{y}_2). \mathbf{0}$, we use some auxiliary derivations:

$$\begin{array}{c} \text{Nil} \frac{}{\mathcal{G}(\Gamma); \mathbf{0} \vdash \mathbf{0}} \\ \text{End} \frac{}{\mathcal{G}(\Gamma); \overline{c_{k+1}} : \text{end} \vdash \mathbf{0}} \\ \text{PolySend} \frac{}{\mathcal{G}(\Gamma); \mathcal{G}(\Delta_2), \vdash \tilde{z} \triangleright \tilde{M}_2} \\ \text{PolyVar} \frac{}{\mathcal{G}(\Gamma); \mathcal{G}(\Delta_2), \vdash \tilde{z} \triangleright \tilde{M}_2} \\ \text{End} \frac{}{\mathcal{G}(\Gamma); \mathcal{G}(\Delta'_2), \overline{c_{k+l+1}} : !(\tilde{M}_2); \text{end} \vdash \overline{c_{k+l+1}} !(\tilde{z}). \mathbf{0} \triangleright \diamond} \\ \text{End} \frac{}{\mathcal{G}(\Gamma); \mathcal{G}(\Delta'_2), \overline{c_{k+1}} : !(\tilde{M}_2); \text{end}, \overline{c_{k+1}} : \text{end} \vdash \overline{c_{k+l+1}} !(\tilde{z}). \mathbf{0} \triangleright \diamond} \end{array} \quad (96)$$

$$\text{PolySend} \frac{\text{PolyVar} \frac{(96) \quad \mathcal{G}(\Gamma); \mathcal{G}(\Delta_1) \vdash \tilde{y}_1 \triangleright \tilde{M}_1}{\mathcal{G}(\Gamma); \mathcal{G}(\Delta'_1, \Delta'_2), \overline{c_{k+1}} : !\langle \tilde{M}_1 \rangle; \text{end}, \overline{c_{k+l+1}} : !\langle \tilde{M}_2 \rangle; \text{end} \vdash \overline{c_{k+1}} ! \langle \tilde{y}_1 \rangle . \overline{c_{k+l+1}} ! \langle \tilde{y}_2 \rangle . \mathbf{0} \triangleright \diamond}}{\mathcal{G}(\Gamma); \mathcal{G}(\Delta'_1, \Delta'_2), \overline{c_{k+1}} : !\langle \tilde{M}_1 \rangle; \text{end}, \overline{c_{k+l+1}} : !\langle \tilde{M}_2 \rangle; \text{end}, c_k : \text{end} \vdash \overline{c_{k+1}} ! \langle \tilde{y}_1 \rangle . \overline{c_{k+l+1}} ! \langle \tilde{y}_2 \rangle . \mathbf{0} \triangleright \diamond}}{\text{End}} \quad (97)$$

$$\text{PolyRecv} \frac{\text{PolyVar} \frac{(97) \quad \mathcal{G}(\Gamma); \mathcal{G}(\Delta'_1, \Delta'_2) \vdash \tilde{x} \triangleright \tilde{M}}{\mathcal{G}(\Gamma_1); \Theta' \vdash c_k ? (\tilde{y}) . \overline{c_{k+1}} ! \langle \tilde{y}_1 \rangle . \overline{c_{k+l+1}} ! \langle \tilde{y}_2 \rangle . \mathbf{0} \triangleright \diamond}}{\mathcal{G}(\Gamma_1); \Theta' \vdash c_k ? (\tilde{y}) . \overline{c_{k+1}} ! \langle \tilde{y}_1 \rangle . \overline{c_{k+l+1}} ! \langle \tilde{y}_2 \rangle . \mathbf{0} \triangleright \diamond}}{\text{End}} \quad (98)$$

$$\text{(Lem. C.3) with } \Gamma'_1 \setminus \Gamma_1 \frac{(94)}{\mathcal{G}(\Gamma_1); \mathcal{G}(\Delta_1), \Theta_1 \vdash \mathcal{B}_{\tilde{y}_1}^{k+1}(Q) \triangleright \diamond} \quad (99)$$

$$\text{(Lem. C.3) with } \Gamma'_2 \setminus \Gamma_1 \frac{(95)}{\mathcal{G}(\Gamma_1); \mathcal{G}(\Delta'_2), \Theta_2 \vdash \mathcal{B}_{\tilde{y}_2}^{k+l+1}(R) \triangleright \diamond} \quad (100)$$

$$\text{Par} \frac{(99) \quad (100)}{\mathcal{G}(\Gamma_1); \mathcal{G}(\Delta'_1, \Delta'_2), \Theta_1, \Theta_2 \vdash \mathcal{B}_{\tilde{y}}^{k+1}(Q) \mid \mathcal{B}_{\tilde{z}}^{k+l+1}(R) \triangleright \diamond} \quad (101)$$

The following tree proves this case:

$$\text{Par} \frac{(98) \quad (101)}{\mathcal{G}(\Gamma_1); \mathcal{G}(\Delta'_1, \Delta'_2), \Theta \vdash c_k ? (\tilde{y}) . \overline{c_{k+1}} ! \langle \tilde{y}_1 \rangle . \overline{c_{k+l+1}} ! \langle \tilde{y}_2 \rangle . \mathbf{0} \mid \mathcal{B}_{\tilde{y}_1}^{k+1}(Q) \mid \mathcal{B}_{\tilde{y}_2}^{k+l+1}(R) \triangleright \diamond}$$

(6) Case $P = \mu X.P'$. The only rule that can be applied here is REC:

$$\frac{\Gamma, X : \Delta; \Delta \vdash P' \triangleright \diamond}{\Gamma; \Delta \vdash \mu X.P'} \quad (102)$$

We remark that by (102) we know $x : S \in \Delta \implies \text{tr}(S)$. Then, by applying Lem. C.6 on the assumption of (102) we have:

$$\mathcal{G}(\Gamma \setminus \tilde{x}); \Theta \vdash \mathbb{A}_{\text{rec}}^{k+1} \tilde{y} (P')_g \triangleright \diamond \quad (103)$$

We take $\tilde{x}' = \text{fn}(P')$, so we have $\Delta \setminus \tilde{x}' = \emptyset$. Further, \tilde{y}' is such that $\text{indexed}_{\Gamma, \Delta}(\tilde{y}', \tilde{x}')$. Let $\Theta' = \Theta'_\mu, \Theta_X(g)$ with $\Theta_X(g) = c'_X : \langle \tilde{M}' \rangle$ with $\tilde{M}' = (\mathcal{G}(\Gamma), \mathcal{G}(\Delta))(\tilde{y}')$. Further, $\tilde{x}' \subseteq P'$ and \tilde{y}' such that $\Delta \setminus \tilde{x}' = \emptyset$ and $\text{indexed}_{\Gamma, \Delta}(\tilde{y}', \tilde{x}')$. Also, $\text{balanced}(\Theta'_\mu)$ with

$$\text{dom}(\Theta'_\mu) = \{c'_{k+1}, \dots, c'_{k+|P'|^*}\} \cup \{\overline{c'_{k+1}}, \dots, \overline{c'_{k+|P'|^*}}\}$$

where $\Theta'_\mu(c_{k+1}) = c_k : ?(\tilde{M}')$; end.

Let $l = |P'|^*$ and \tilde{z} such that $|\tilde{y}'| = |\tilde{z}|$. Further, let $\tilde{x} \subseteq \text{fn}(P)$ and $\tilde{y} = \text{bn}(\tilde{x} : \tilde{S})$. By definition we have $\tilde{x} \subseteq \tilde{x}'$. By Tab. 3, we have:

$$\mathcal{B}_{\tilde{y}}^k(P) = (\nu c'_X) (c_k ? (\tilde{y}) . \overline{c'_{k+1}} ! \langle \tilde{y}' \rangle . \mu X . c'_X ? (\tilde{z}) . \overline{c'_{k+1}} ! \langle \tilde{z} \rangle . X \mid \mathbb{A}_{\text{rec}}^{k+1} \tilde{y} (P')_g)$$

Let $\Theta_\mu = \Theta'_\mu \cdot \Theta''_\mu$ where:

$$\Theta''_\mu = \overline{c'_k} : \mu t. ?(\tilde{M}'); t \cdot \overline{c'_{k+1}} : \mu t. !(\tilde{M}'); t$$

We shall prove the following judgment:

$$\mathcal{G}(\Gamma \setminus \tilde{x}); \mathcal{G}(\Delta \setminus \tilde{x}) \cdot \Delta_\mu \vdash \mathcal{B}_{\tilde{y}}^k(P) \triangleright \diamond \quad (104)$$

Let $\Theta'_X = c'_X : \mu t. ?(\tilde{M}'); t \cdot \overline{c'_X} : \mu t. !(\tilde{M}'); t$. We use some auxiliary sub-trees:

$$\text{RVar} \frac{\mathcal{G}(\Gamma \setminus \tilde{x}), X : \Theta''; \Theta'' \vdash X \triangleright \diamond}{\mathcal{G}(\Gamma \setminus \tilde{x}), X : \Theta''; \mathcal{G}(\Delta), \Theta'' \vdash \overline{c'_{k+1}} ! \langle \tilde{z} \rangle . X \triangleright \diamond} \quad \text{PolyVar} \frac{\mathcal{G}(\Gamma \setminus \tilde{x}), X : \Theta''; \mathcal{G}(\Delta) \vdash \tilde{z} \triangleright \tilde{M}}{\mathcal{G}(\Gamma \setminus \tilde{x}), X : \Theta''; \mathcal{G}(\Delta), \Theta'' \vdash \overline{c'_{k+1}} ! \langle \tilde{z} \rangle . X \triangleright \diamond} \quad (105)$$

$$\begin{array}{c}
 \text{PolyRev} \frac{\text{PolyVar} \frac{(105) \quad \mathcal{G}(\Gamma \setminus \bar{x}); \mathcal{G}(\Delta) \vdash \bar{z} \triangleright \bar{M}}{\mathcal{G}(\Gamma \setminus \bar{x}); \mathcal{G}(\Delta) \vdash \bar{z} \triangleright \bar{M}}}{\mathcal{G}(\Gamma \setminus \bar{x}), X : \Theta''; \Theta'' \vdash c_X^r ?(\bar{z}). \overline{c_{k+1}^r} !(\bar{z}). X \triangleright \diamond} \\
 \text{Rec} \frac{}{\mathcal{G}(\Gamma \setminus \bar{x}); \Theta'' \vdash \mu X. c_X^r ?(\bar{z}). \overline{c_{k+1}^r} !(\bar{z}). X \triangleright \diamond}
 \end{array} \quad (106)$$

where $\Theta'' = \overline{c_{k+1}^r} : \mu t. !(\bar{M}); t, c_X^r : \mu t. ?(\bar{M}); t$.

$$\text{PolySend} \frac{\text{PolyVar} \frac{(106) \quad \mathcal{G}(\Gamma \setminus \bar{x}); \mathcal{G}(\Delta) \vdash \bar{y}' \triangleright \bar{M}'}{\mathcal{G}(\Gamma \setminus \bar{x}); \mathcal{G}(\Delta) \vdash \bar{y}' \triangleright \bar{M}'}}{\mathcal{G}(\Gamma \setminus \bar{x}); \mathcal{G}(\Delta), \overline{c_{k+1}^r} : \mu t. !(\bar{M}); t, c_X^r : \mu t. ?(\bar{M}); t \vdash c_{k+1}^r !(\bar{y}'). \mu X. c_X^r ?(\bar{z}). \overline{c_{k+1}^r} !(\bar{z}). X \triangleright \diamond} \quad (107)$$

$$\text{PolyRev} \frac{\text{PolyVar} \frac{(107) \quad \mathcal{G}(\Gamma \setminus \bar{x}); \bar{y} : \bar{M} \vdash \bar{y} \triangleright \bar{M}}{\mathcal{G}(\Gamma \setminus \bar{x}); \bar{y} : \bar{M} \vdash \bar{y} \triangleright \bar{M}}}{\mathcal{G}(\Gamma \setminus \bar{x}); \mathcal{G}(\Delta \setminus \bar{x}) \cdot \Theta'' \cdot c_X^r : \mu t. ?(\bar{M}); t \vdash c_k ?(\bar{y}). \overline{c_{k+1}^r} !(\bar{y}'). \mu X. c_X^r ?(\bar{z}). \overline{c_{k+1}^r} !(\bar{z}). X \triangleright \diamond} \quad (108)$$

The following tree proves this case:

$$\begin{array}{c}
 \text{Par} \frac{\text{ResS} \frac{\text{PolyVar} \frac{(108) \quad (103) \quad \mathcal{G}(\Gamma \setminus \bar{x}); \mathcal{G}(\Delta \setminus \bar{x}) \cdot \Theta'' \cdot c_X^r : \mu t. ?(\bar{M}); t \vdash c_k ?(\bar{y}). \overline{c_{k+1}^r} !(\bar{y}'). \mu X. c_X^r ?(\bar{z}). \overline{c_{k+1}^r} !(\bar{z}). X \triangleright \diamond \mid \mathbf{A}_{\text{rec}}^{k+1} \bar{y} (P') \triangleright \diamond}{\mathcal{G}(\Gamma \setminus \bar{x}); \mathcal{G}(\Delta \setminus \bar{x}) \cdot \Theta'' \cdot c_X^r : \mu t. ?(\bar{M}); t \vdash c_k ?(\bar{y}). \overline{c_{k+1}^r} !(\bar{y}'). \mu X. c_X^r ?(\bar{z}). \overline{c_{k+1}^r} !(\bar{z}). X \triangleright \diamond \mid \mathbf{A}_{\text{rec}}^{k+1} \bar{y} (P') \triangleright \diamond}}{\mathcal{G}(\Gamma \setminus \bar{x}); \mathcal{G}(\Delta \setminus \bar{x}) \cdot \Theta'' \cdot c_X^r : \mu t. ?(\bar{M}); t \vdash c_k ?(\bar{y}). \overline{c_{k+1}^r} !(\bar{y}'). \mu X. c_X^r ?(\bar{z}). \overline{c_{k+1}^r} !(\bar{z}). X \triangleright \diamond \mid \mathbf{A}_{\text{rec}}^{k+1} \bar{y} (P') \triangleright \diamond}}{\mathcal{G}(\Gamma \setminus \bar{x}); \mathcal{G}(\Delta \setminus \bar{x}) \cdot \Theta'' \cdot c_X^r : \mu t. ?(\bar{M}); t \vdash c_k ?(\bar{y}). \overline{c_{k+1}^r} !(\bar{y}'). \mu X. c_X^r ?(\bar{z}). \overline{c_{k+1}^r} !(\bar{z}). X \triangleright \diamond \mid \mathbf{A}_{\text{rec}}^{k+1} \bar{y} (P') \triangleright \diamond}
 \end{array} \quad (109)$$

(7) Case $P = r!(z).P'$ when $\text{tr}(r)$. For this case Rule SEND can be applied:

$$\text{Send} \frac{\Gamma; \Delta, r : S' \vdash P' \triangleright \diamond \quad \Gamma; \Delta_z \vdash z \triangleright C}{\Gamma; \Delta, r : S, \Delta_z \vdash r!(z).P' \triangleright \diamond} \quad (110)$$

where $S = !\langle C \rangle; S'$.

Then, by IH on the first assumption of (110) we have:

$$\mathcal{G}(\Gamma_1'); \mathcal{G}(\Delta_1'), \Theta_1 \vdash \mathcal{B}_{\bar{y}'}^{k+1}(P') \triangleright \diamond \quad (111)$$

where $\bar{x}' \subseteq \text{fn}(P')$ such that $r \in \bar{x}'$ and \bar{y}' such that $\text{indexed}_{\Gamma, \Delta, r, S}(\bar{y}', \bar{x}')$. By this follows that $(r_1, \dots, r_{|\mathcal{G}(S)|}) \subseteq \bar{y}'$. Also, $\Gamma_1' = \Gamma \setminus \bar{x}'$, $\Delta_1' = \Delta \setminus \bar{y}'$, and $\text{balanced}(\Theta_1)$ with

$$\text{dom}(\Theta_1) = \{c_{k+1}, \dots, c_{k+|P'|}\} \cup \{\overline{c_{k+2}}, \dots, \overline{c_{k+|P'|}}\}$$

and $\Theta_1(c_{k+1}) = ?(\bar{M}_1)$; end where $\bar{M}_1 = (\mathcal{G}(\Gamma), \mathcal{G}(\Delta, r : S))(\bar{y}')$.

By applying Lem. C.2 on the assumption of (68) we have:

$$\mathcal{G}(\Gamma); \mathcal{G}(\Delta_z) \vdash \bar{z} \triangleright \mathcal{G}(C) \quad (112)$$

We assume $\bar{x} = \bar{x}', z$. Since $\bar{x}' \subseteq \text{fn}(P')$ follows that $\bar{x} \subseteq \text{fn}(P)$. Let $\bar{y} = \bar{y}', \bar{z}$ where $|\bar{z}| = |\mathcal{G}(C)|$. By Def. 4.12 follows that $\text{indexed}_{\Gamma, \Delta, r, S, \Delta_z}(\bar{y}, \bar{x})$. We define $\Theta = \Theta_1, \Theta'$, where:

$$\Theta = c_k : ?(\bar{M}); \text{end}, \overline{c_{k+1}} : !(\bar{M}_1); \text{end}$$

with $\bar{M} = (\mathcal{G}(\Gamma), \mathcal{G}(\Delta, r : S, \Delta_z))(\bar{y})$. By Def. 4.5, we know $[P]^* = [P']^* + 1$, so

$$\text{dom}(\Theta) = \{c_k, \dots, c_{k+|P|^* - 1}\} \cup \{\overline{c_{k+1}}, \dots, \overline{c_{k+|P|^* - 1}}\}$$

and Θ is balanced since $\Theta(c_{k+1})$ dual $\Theta(\overline{c_{k+1}})$ and Θ_1 is balanced.

By Tab. 3, we have:

$$\mathcal{B}_{\bar{y}'}^k(r!(z).P') = c_k ?(\bar{y}). r_f(S) !(\bar{z}). \overline{c_{k+1}} !(\bar{y}'). \mathbf{0} \mid \mathcal{B}_{\bar{y}'}^{k+1}(P')$$

Let $\Gamma_1 = \Gamma \setminus \bar{x} = \Gamma_1'$ and $\Delta \setminus \bar{x} = \Delta_1'$.

We shall prove the following judgment:

$$\mathcal{G}(\Gamma_1); \mathcal{G}(\Delta_1'), \Theta \vdash \mathcal{B}_{\bar{y}'}^k(r!(z).P') \triangleright \diamond \quad (113)$$

Let $\Delta_1 = \Delta, r : S, \Delta_z$. To type the left-hand side component of $\mathcal{B}_{\bar{y}'}^k(r!(z).P')$ we use some auxiliary derivations:

$$\begin{array}{c}
 \text{End} \frac{\text{Nil} \frac{}{\mathcal{G}(\Gamma); \mathbf{0} \triangleright \diamond}}{\mathcal{G}(\Gamma); \overline{c_{k+1}} : \text{end} \triangleright \mathbf{0} \triangleright \diamond} \quad \text{PolyVar} \frac{}{\mathcal{G}(\Gamma); \mathcal{G}(\Delta, r : S \setminus \Delta_1') \vdash \bar{y}' \triangleright \bar{M}_1}}{\text{PolySend} \frac{}{\mathcal{G}(\Gamma); \overline{c_{k+1}} : !(\bar{M}_1); \text{end}, \mathcal{G}(\Delta, r : S \setminus \Delta_1') \vdash \overline{c_{k+1}} !(\bar{y}'). \triangleright \diamond}} \quad (114)
 \end{array}$$

$$\text{PolySend} \frac{\text{End} \frac{\mathcal{G}(\Gamma); \mathcal{G}(\Delta_1 \setminus \Delta'_1, \overline{c_{k+1}} :! \langle \widetilde{M}_1 \rangle); \text{end} \vdash r_{f(S)}! \langle \widetilde{z} \rangle . \overline{c_{k+1}}! \langle \widetilde{y}' \rangle . \mathbf{0} \triangleright \diamond}{\mathcal{G}(\Gamma); \mathcal{G}(\Delta_1 \setminus \Delta'_1, \overline{c_{k+1}} :! \langle \widetilde{M}_1 \rangle); \text{end}, c_k : \text{end} \vdash r_{f(S)}! \langle \widetilde{z} \rangle . \overline{c_{k+1}}! \langle \widetilde{y}' \rangle . \mathbf{0} \triangleright \diamond}}{(114) \quad (112)} \quad (115)$$

$$\text{PolyRev} \frac{\text{PolyVar} \frac{\mathcal{G}(\Gamma); \mathcal{G}(\Delta_1 \setminus \Delta'_1) \vdash \widetilde{y} \triangleright \widetilde{M}}{(115)}}{\mathcal{G}(\Gamma_1); \Theta' \vdash c_k ? (\widetilde{y}) . r_{f(S)}! \langle \widetilde{z} \rangle . \overline{c_{k+1}}! \langle \widetilde{y}' \rangle . \mathbf{0} \triangleright \diamond} \quad (116)$$

We may notice that by Def. 4.4 and Def. 12 we have $\mathcal{G}(\Delta_1 \setminus \Delta'_1) = \mathcal{G}(\Delta)$, $\widetilde{r} : \mathcal{R}^*(S)$, $\mathcal{G}(\Delta_z) \setminus \mathcal{G}(\Delta'_1)$. Further, by Lem. C.5 we know $\mathcal{G}(\Delta_1)(r_{f(S)}) = \mu t. ! \langle \mathcal{G}(C) \rangle; t$.

The following tree proves this case:

$$\text{Par} \frac{\text{PolyRev} \frac{\text{PolyVar} \frac{\mathcal{G}(\Gamma_1); \mathcal{G}(\Delta'_1), \Theta_1 \vdash \mathcal{B}_{\widetilde{y}'}^{k+1}(P') \triangleright \diamond}{(116)}}{\mathcal{G}(\Gamma_1); \mathcal{G}(\Delta'_1), \Theta \vdash \mathcal{B}_{\widetilde{y}}^k(r! \langle z \rangle . P') \triangleright \diamond}}{(111)}}{\mathcal{G}(\Gamma_1); \mathcal{G}(\Delta'_1), \Theta \vdash \mathcal{B}_{\widetilde{y}}^k(r! \langle z \rangle . P') \triangleright \diamond} \quad (117)$$

(8) Case $P = r?(z).P'$ when $\text{tr}(r)$. For this case Rule Rcv can be applied:

$$\text{Rev} \frac{\Gamma; \Delta, r : S', \Delta_z \vdash P' \triangleright \diamond \quad \Gamma; \Delta_z \vdash z \triangleright C}{\Gamma \setminus z; \Delta, r : S \vdash r?(z).P' \triangleright \diamond} \quad (118)$$

where $S = ?(C); S'$.

Then, by IH on the first assumption of (49) we know:

$$\mathcal{G}(\Gamma'_1); \mathcal{G}(\Delta'_1), \Theta_1 \vdash \mathcal{B}_{\widetilde{y}'}^{k+1}(P') \triangleright \diamond \quad (119)$$

where $\widetilde{x}' \subseteq \text{fn}(P')$ such that $r \in \widetilde{x}'$ and \widetilde{y}' such that $\text{indexed}_{\Gamma, \Delta, r, S}(\widetilde{y}', \widetilde{x}')$. By this follows that $(r_1, \dots, r_{|\mathcal{G}(S)|}) \subseteq \widetilde{y}'$. Also, $\Gamma'_1 = \Gamma \setminus \widetilde{x}'$, $\Delta'_1 = \Delta \setminus \widetilde{x}'$, and $\text{balanced}(\Theta_1)$ with

$$\text{dom}(\Theta_1) = \{c_{k+1}, \dots, c_{k+|P'|^*}\} \cup \{\overline{c_{k+2}}, \dots, \overline{c_{k+|P'|^*}}\}$$

and $\Theta_1(c_{k+1}) = ?(\widetilde{M}')$; end where $\widetilde{M}' = (\mathcal{G}(\Gamma), \mathcal{G}(\Delta, r : S, \Delta_z))(\widetilde{y}')$.

By applying Lem. C.2 to the second assumption of (49) we have:

$$\mathcal{G}(\Gamma); \mathcal{G}(\Delta_z) \vdash z \triangleright \mathcal{G}(C) \quad (120)$$

Let $\widetilde{x} = \widetilde{x}' \setminus z$ and $\widetilde{y} = \widetilde{y}' \setminus \widetilde{z}$ such that $|\widetilde{z}| = \mathcal{G}(C)$ where. We may notice that by Def. 4.12 $\text{indexed}_{\Gamma, \Delta, r, S}(\widetilde{y}, \widetilde{x})$ holds. We define $\Theta = \Theta_1, \Theta'$, where

$$\Theta' = c_k : ?(\widetilde{M}'); \text{end}, \overline{c_{k+1}} : ! \langle \widetilde{M}' \rangle; \text{end}$$

with $\widetilde{M} = (\mathcal{G}(\Gamma), \mathcal{G}(\Delta, u_i : ?(C); S))(\widetilde{y})$. By Def. 4.5, $[P]^* = [P']^* + 1$ so

$$\text{dom}(\Theta) = \{c_k, \dots, c_{k+|P|^*-1}\} \cup \{\overline{c_{k+1}}, \dots, \overline{c_{k+|P|^*-1}}\}$$

and Θ is balanced since $\Theta(c_{k+1})$ dual $\Theta(\overline{c_{k+1}})$ and Θ_1 is balanced. By Tab. 3:

$$\mathcal{B}_{\widetilde{y}}^k(r?(z).P') = c_k ? (\widetilde{y}) . r_{f(S)} ? (\widetilde{z}) . \overline{c_{k+1}}! \langle \widetilde{y}' \rangle . \mathbf{0} \mid \mathcal{B}_{\widetilde{y}'}^{k+1}(P')$$

Also, let $\Gamma_1 = \Gamma \setminus \widetilde{x}$ and $\Delta_1 = \Delta \setminus \widetilde{x}$. We may notice that $\Delta_1 = \Delta'_1$. We shall prove the following judgment:

$$\mathcal{G}(\Gamma_1 \setminus z); \mathcal{G}(\Delta_1), \Theta \vdash \mathcal{B}_{\widetilde{y}}^k(r?(z).P')$$

Let $\Delta_1 = \Delta, r : S$. The left-hand side component of $\mathcal{B}_{\widetilde{y}}^k(r?(z).P')$ is typed using some auxiliary derivations:

$$\text{PolySend} \frac{\text{End} \frac{\text{Nil} \frac{\mathcal{G}(\Gamma); \mathbf{0} \vdash \mathbf{0} \triangleright \diamond}{\mathcal{G}(\Gamma); \overline{c_{k+1}} : \text{end} \vdash \mathbf{0} \triangleright \diamond}}{\mathcal{G}(\Gamma); \overline{c_{k+1}} : ! \langle \widetilde{M}' \rangle; \text{end}, \mathcal{G}(\Delta \setminus \Delta_1) \vdash \overline{c_{k+1}}! \langle \widetilde{y}' \rangle . \mathbf{0} \triangleright \diamond}}{\text{PolyVar} \frac{\mathcal{G}(\Gamma); \mathcal{G}(\Delta \setminus \Delta_1) \vdash \widetilde{y}' \triangleright \widetilde{M}'}}{\mathcal{G}(\Gamma); \mathcal{G}(\Delta_1), \Theta \vdash \mathcal{B}_{\widetilde{y}}^k(r?(z).P') \triangleright \diamond}} \quad (121)$$

$$\text{End} \frac{\text{Rcv} \frac{\mathcal{G}(\Gamma \setminus z); \overline{c_{k+1}} : ! \langle \widetilde{M}' \rangle; \text{end}, \mathcal{G}(\Delta_2) \vdash r_{f(S)} ? (\widetilde{z}) . \overline{c_{k+1}}! \langle \widetilde{y}' \rangle . \mathbf{0} \triangleright \diamond}{\mathcal{G}(\Gamma \setminus z); \overline{c_{k+1}} : ! \langle \widetilde{M}' \rangle; \text{end}, c_k : \text{end}, \mathcal{G}(\Delta_2) \vdash r_{f(S)} ? (\widetilde{z}) . \overline{c_{k+1}}! \langle \widetilde{y}' \rangle . \mathbf{0} \triangleright \diamond}}{(121) \quad (120)}}{\mathcal{G}(\Gamma \setminus z); \overline{c_{k+1}} : ! \langle \widetilde{M}' \rangle; \text{end}, c_k : \text{end}, \mathcal{G}(\Delta_2) \vdash r_{f(S)} ? (\widetilde{z}) . \overline{c_{k+1}}! \langle \widetilde{y}' \rangle . \mathbf{0} \triangleright \diamond} \quad (122)$$

$$\text{PolyRev} \frac{\text{PolyVar} \frac{(122) \quad \mathcal{G}(\Gamma \setminus z); \mathcal{G}(\Delta_2) \vdash \tilde{y} \triangleright \tilde{M}}{\mathcal{G}(\Gamma_1 \setminus z); \Theta' \vdash c_k ?(\tilde{y}). r_{f(S)} ?(z). \overline{c_{k+1}} !(\tilde{y}'). \mathbf{0} \triangleright \diamond}}{\mathcal{G}(\Gamma_1 \setminus z); \mathcal{G}(\Delta_1), \Theta_1 \vdash \mathcal{B}_{\tilde{y}'}^{k+1}(P') \triangleright \diamond}}}{\mathcal{G}(\Gamma_1 \setminus z); \mathcal{G}(\Delta_1), \Theta \vdash c_k ?(\tilde{y}). r_{f(S)} ?(\tilde{z}). \overline{c_{k+1}} !(\tilde{y}'). \mathbf{0} \mid \mathcal{B}_{\tilde{y}'}^{k+1}(P') \triangleright \diamond}} \quad (123)$$

where $\Delta_2 = \Delta, r : S \setminus \Delta_1$. We may notice that by Def. 4.4 and Def. 12 we have $\mathcal{G}(\Delta_2) = \mathcal{G}(\Delta), \tilde{r} : \mathcal{R}^*(S) \setminus \mathcal{G}(\Delta_1)$. Further, by Lem. C.5 we know $\mathcal{G}(\Delta_2)(r_{f(S)}) = \mu t. ?(\mathcal{G}(C)); t$.

The following tree proves this case:

$$\text{Par} \frac{\text{PolyRev} \frac{(123) \quad \mathcal{G}(\Gamma_1' \setminus z); \mathcal{G}(\Delta_1), \Theta_1 \vdash \mathcal{B}_{\tilde{y}'}^{k+1}(P') \triangleright \diamond}{\mathcal{G}(\Gamma_1 \setminus z); \mathcal{G}(\Delta_1), \Theta \vdash c_k ?(\tilde{y}). r_{f(S)} ?(\tilde{z}). \overline{c_{k+1}} !(\tilde{y}'). \mathbf{0} \mid \mathcal{B}_{\tilde{y}'}^{k+1}(P') \triangleright \diamond}}{\mathcal{G}(\Gamma \setminus \tilde{x}'); \mathcal{G}(\Delta \setminus \tilde{x}'), \Theta_1 \vdash \mathcal{B}_{\tilde{y}'}^{k+1}(P') \triangleright \diamond}}}{\mathcal{G}(\Gamma \setminus \tilde{x}'); \mathcal{G}(\Delta \setminus \tilde{x}'), \Theta \vdash c_k ?(\tilde{y}). r_{f(S)} ?(\tilde{z}). \overline{c_{k+1}} !(\tilde{y}'). \mathbf{0} \mid \mathcal{B}_{\tilde{y}'}^{k+1}(P') \triangleright \diamond}} \quad (119)$$

(9) Case $P = (\nu s : \mu t.S)P'$. For this case Rule RESS can be applied:

$$\text{ResS} \frac{\Gamma; \Delta, s : \mu t.S, \bar{s} : \overline{\mu t.S} \vdash P'}{\Gamma; \Delta \vdash (\nu s : \mu t.S)P'} \quad (125)$$

By IH on the assumption of (125) we have:

$$\mathcal{G}(\Gamma \setminus \tilde{x}'); \mathcal{G}(\Delta \setminus \tilde{x}'), \Theta_1 \vdash \mathcal{B}_{\tilde{y}'}^{k+1}(P') \quad (126)$$

where we take \tilde{y}' such that $\tilde{s}, \bar{\tilde{s}} \subseteq \tilde{y}'$ with $\tilde{s} = (s_1, \dots, s_{|\mathcal{R}(S)|})$ and $\bar{\tilde{s}} = (\bar{s}_1, \dots, \bar{s}_{|\mathcal{R}(S)|})$. Accordingly, \tilde{x}' is such that $s, \bar{s} \subseteq \tilde{x}'$. Since $\text{lin}(s)$ and $\text{lin}(\bar{s})$ we know $\tilde{x}' \subseteq \text{fn}(P')$. Also, $\text{indexed}_{\Gamma, \Delta_1}(\tilde{y}', \tilde{x}')$ where $\Delta_1 = \Delta, s : \mu t.S, \bar{s} : \overline{\mu t.S}$. Also, $\text{balanced}(\Theta_1)$ with

$$\text{dom}(\Theta_1) = \{c_{k+1}, \dots, c_{k+|\mathcal{P}'|}\} \cup \{\overline{c_{k+2}}, \dots, \overline{c_{k+|\mathcal{P}'|}}\}$$

and $\Theta_1(c_{k+1}) = ?(\tilde{M}')$; end where $\tilde{M} = (\mathcal{G}(\Gamma), \mathcal{G}(\Delta))(\tilde{y}')$.

Let $\tilde{y} = \tilde{y}' \setminus (\tilde{s}, \bar{\tilde{s}})$ and $\tilde{x} = \tilde{x}' \setminus (s, \bar{s})$. Since $s, \bar{s} \notin \text{fn}(P)$ we know $\tilde{x} \subseteq \text{fn}(P)$ and $\text{indexed}_{\Gamma, \Delta}(\tilde{y}, \tilde{x})$.

We define $\Theta = \Theta_1, \Theta'$ where

$$\Theta' = c_k : ?(\tilde{M}); \text{end}, \overline{c_{k+1}} : !(\tilde{M}'); \text{end}$$

By Tab. 3, we have:

$$\mathcal{B}_{\tilde{y}}^k(P) = (\nu \bar{s} : \mathcal{R}(S))(c_k ?(\tilde{y}). \overline{c_{k+1}} !(\tilde{y}'). \mathbf{0} \mid \mathcal{B}_{\tilde{y}'}^k(P'))$$

We should prove the following judgment:

$$\mathcal{G}(\Gamma \setminus \tilde{x}); \mathcal{G}(\Delta \setminus \tilde{x}), \Theta \vdash \mathcal{B}_{\tilde{y}}^k(P) \triangleright \diamond$$

We use some auxiliary sub-tree:

$$\text{PolySend} \frac{\text{Nil} \frac{\mathcal{G}(\Gamma); \mathbf{0} \vdash \mathbf{0} \triangleright \diamond}{\mathcal{G}(\Gamma); \mathcal{G}(\Delta \cap \tilde{x}), \bar{s} : \mathcal{R}(S), \bar{\tilde{s}} : \mathcal{R}(\bar{S}) \vdash \overline{c_{k+1}} !(\tilde{y}'). \mathbf{0} \triangleright \diamond}}{\mathcal{G}(\Gamma); \mathcal{G}(\Delta \cap \tilde{x}') \vdash \tilde{y}' \triangleright \tilde{M}'} \quad \text{PolyVar} \frac{\mathcal{G}(\Gamma); \mathcal{G}(\Delta_1 \cap \tilde{x}') \vdash \tilde{y}' \triangleright \tilde{M}'}{\mathcal{G}(\Gamma); \mathcal{G}(\Delta \cap \tilde{x}), \bar{s} : \mathcal{R}(S), \bar{\tilde{s}} : \mathcal{R}(\bar{S}) \vdash \overline{c_{k+1}} !(\tilde{y}'). \mathbf{0} \triangleright \diamond}}{\mathcal{G}(\Gamma); \mathcal{G}(\Delta \cap \tilde{x}), \bar{s} : \mathcal{R}(S), \bar{\tilde{s}} : \mathcal{R}(\bar{S}) \vdash \overline{c_{k+1}} !(\tilde{y}'). \mathbf{0} \triangleright \diamond}} \quad (127)$$

here we may notice that

$$\mathcal{G}(\Delta \cap \tilde{x}), \bar{s} : \mathcal{R}(S), \bar{\tilde{s}} : \mathcal{R}(\bar{S}) = \mathcal{G}(\Delta_1 \cap \tilde{x}')$$

The following tree proves this case:

$$\text{PolyResS} \frac{\text{Par} \frac{\text{PolyRev} \frac{\text{PolyVar} \frac{(127) \quad \mathcal{G}(\Gamma); \mathcal{G}(\Delta \cap \tilde{x}) \vdash \tilde{y}' \triangleright \tilde{M}}{\mathcal{G}(\Gamma \setminus \tilde{x}); \Theta', \bar{s} : \mathcal{R}(S), \bar{\tilde{s}} : \mathcal{R}(\bar{S}) \vdash c_k ?(\tilde{y}). \overline{c_{k+1}} !(\tilde{y}'). \mathbf{0} \triangleright \diamond} \quad (126)}{\mathcal{G}(\Gamma \setminus \tilde{x}); \mathcal{G}(\Delta \setminus \tilde{x}), \Theta, \bar{s} : \mathcal{R}(S), \bar{\tilde{s}} : \mathcal{R}(\bar{S}) \vdash c_k ?(\tilde{y}). \overline{c_{k+1}} !(\tilde{y}'). \mathbf{0} \mid \mathcal{B}_{\tilde{y}'}^k(P') \triangleright \diamond}}{\mathcal{G}(\Gamma \setminus \tilde{x}); \mathcal{G}(\Delta \setminus \tilde{x}), \Theta \vdash (\nu \bar{s} : \mathcal{R}(S))(c_k ?(\tilde{y}). \overline{c_{k+1}} !(\tilde{y}'). \mathbf{0} \mid \mathcal{B}_{\tilde{y}'}^k(P')) \triangleright \diamond}}}{\mathcal{G}(\Gamma \setminus \tilde{x}); \mathcal{G}(\Delta \setminus \tilde{x}), \Theta \vdash (\nu \bar{s} : \mathcal{R}(S))(c_k ?(\tilde{y}). \overline{c_{k+1}} !(\tilde{y}'). \mathbf{0} \mid \mathcal{B}_{\tilde{y}}^k(P)) \triangleright \diamond}} \quad (128)$$

□
□

C.5 Proof of Thm. 4 (Minimality Result, Optimized)

THEOREM 4 (MINIMALITY RESULT FOR π , OPTIMIZED). *Let P be a π process with $\tilde{u} = \text{fn}(P)$. If $\Gamma; \Delta \vdash P \triangleright \diamond$ then $\mathcal{H}^*(\Gamma\sigma); \mathcal{H}^*(\Delta\sigma) \vdash \mathcal{F}^*(P) \triangleright \diamond$, where $\sigma = \{\text{init}(\tilde{u})/\tilde{u}\}$.*

PROOF. By assumption, $\Gamma; \Delta \vdash P \triangleright \diamond$. Then by applying Lem. C.1 we have:

$$\Gamma\sigma; \Delta\sigma \vdash P\sigma \triangleright \diamond \quad (129)$$

By Def. 4.8, we shall prove the following judgment:

$$\mathcal{G}(\Gamma\sigma); \mathcal{G}(\Delta\sigma) \vdash (v\tilde{c})(\overline{c_k}!\langle\tilde{r}\rangle.\mathbf{0} \mid \mathcal{B}_{\tilde{r}}^k(P\sigma)) \triangleright \diamond \quad (130)$$

where $\tilde{c} = (c_k, \dots, c_{k+[P]^*-1})$; $k > 0$; and $\tilde{r} = \bigcup_{v \in \tilde{v}} \{v_1, \dots, v_{|\mathcal{G}(S)|}\}$ with $v : S$. Since $\tilde{v} \subseteq \text{fn}(P)$ we know $\text{indexed}_{\Gamma\sigma, \Delta\sigma}(\tilde{r}, \tilde{v})$. Since $P\sigma$ is an initialized process, we apply Thm. 3 to (129) to get:

$$\mathcal{G}(\Gamma\sigma); \mathcal{G}(\Delta\sigma \setminus \tilde{v}), \Theta \vdash \mathcal{B}_{\tilde{r}}^k(P\sigma) \triangleright \diamond \quad (131)$$

where Θ is balanced with $\text{dom}(\Theta) = \{c_k, c_{k+1}, \dots, c_{k+[P]^*-1}\} \cup \{\overline{c_{k+1}}, \dots, \overline{c_{k+[P]^*-1}}\}$ and $\Theta(c_k) = ?(\tilde{M})$; end with $\tilde{M} = \mathcal{G}(\Delta)(\tilde{r})$.

The following tree proves this case:

$$\begin{array}{c} \text{Nil} \frac{}{\mathcal{G}(\Gamma\sigma); \emptyset \vdash \mathbf{0} \triangleright \diamond} \\ \text{End} \frac{}{\mathcal{G}(\Gamma\sigma); c_k : \text{end} \vdash \mathbf{0} \triangleright \diamond} \quad \text{PolyVar} \frac{}{\mathcal{G}(\Gamma\sigma); \tilde{r} : \tilde{M} \vdash \tilde{r} \triangleright \tilde{M}} \\ \text{Send} \frac{}{\mathcal{G}(\Gamma\sigma); \overline{c_k} : !(\tilde{M}); \text{end}, \tilde{r} : \tilde{M} \vdash \overline{c_k}!\langle\tilde{r}\rangle.\mathbf{0} \triangleright \diamond} \quad (131) \\ \text{Par} \frac{}{\mathcal{G}(\Gamma\sigma); \mathcal{G}(\Delta\sigma), \overline{c_k} : !(\tilde{M}); \text{end}, \Theta \vdash \overline{c_k}!\langle\tilde{r}\rangle.\mathbf{0} \mid \mathcal{B}_{\tilde{r}}^k(P\sigma) \triangleright \diamond} \\ \text{ResS} \frac{}{\mathcal{G}(\Gamma\sigma); \mathcal{G}(\Delta\sigma) \vdash (v\tilde{c})(\overline{c_k}!\langle\tilde{r}\rangle.\mathbf{0} \mid \mathcal{B}_{\tilde{r}}^k(P\sigma)) \triangleright \diamond} \end{array}$$

□

C.6 MST Bisimilarity

Now, we are ready to relate processes, modulo indexed names (Def. 4.13), using the relation \diamond defined as follows:

Definition C.1 (\diamond Indexed process relation). We define the relation \diamond as

$$\begin{array}{c} \text{[IPSNd]} \\ \frac{P\sigma \diamond P' \quad v\sigma \triangleright_c \tilde{v} \quad \sigma = \text{next}(n_i)}{n_i!\langle v \rangle.P \diamond n_i!\langle \tilde{v} \rangle.P'} \quad \text{[IPINACT]} \\ \frac{}{\mathbf{0} \diamond \mathbf{0}} \\ \text{[IPrcv]} \\ \frac{P\sigma \diamond P' \quad v\sigma \triangleright_c \tilde{v} \quad \sigma = \text{next}(n_i)}{n_i?(y).P \diamond n_i?(\tilde{y}).P'} \quad \text{[IPNews]} \\ \frac{P \diamond P' \quad \tilde{m}_1 \triangleright_c \tilde{m}_2}{(v\tilde{m}_1)P \diamond (v\tilde{m}_2)P'} \end{array}$$

Definition C.2 (Minimal characteristic processes).

$$\begin{array}{l} \langle ?(C); S \rangle_i^u \stackrel{\text{def}}{=} u_i?(x).(t!\langle u_{i+1}, \dots, u_{i+|\mathcal{G}(S)|} \rangle.\mathbf{0} \mid \langle C \rangle_i^x) \\ \langle !\langle C \rangle; S \rangle_i^u \stackrel{\text{def}}{=} u_i!\langle \langle C \rangle_c \rangle.t!\langle u_{i+1}, \dots, u_{i+|\mathcal{G}(S)|} \rangle.\mathbf{0} \\ \langle \text{end} \rangle_i^u \stackrel{\text{def}}{=} \mathbf{0} \\ \langle \langle C \rangle \rangle_i^u \stackrel{\text{def}}{=} u_i!\langle \langle C \rangle_c \rangle.t!\langle u_i \rangle.\mathbf{0} \\ \langle \mu t.S \rangle_i^u \stackrel{\text{def}}{=} \langle S\{\text{end}/t\} \rangle_i^u \\ \langle S \rangle_c \stackrel{\text{def}}{=} \tilde{s} \ (|\tilde{s}| = |\mathcal{G}(S)|, \tilde{s} \text{ fresh}) \\ \langle \langle C \rangle \rangle_c \stackrel{\text{def}}{=} a_1 \ (a_1 \text{ fresh}) \end{array}$$

Definition C.3 (Trigger Collections). We let H, H' to range over *trigger collections*: processes of the form $P_1 \mid \dots \mid P_n$ (with $n \geq 1$), where each P_i is a trigger process or a process that originates from a trigger process.

Example C.4. Let $H_1 = t \leftarrow_c v : C \mid \llbracket C \rrbracket^{u_1} \mid t_2!\langle u_2 \rangle.\mathbf{0}$ where v, t_1, t_2, u_1, u_2 are channel names, C a channel type. We could see that, $\llbracket C \rrbracket^{u_1}$ and $t_2!\langle u_2 \rangle.\mathbf{0}$ originates from a trigger process. Thus, H_1 is a trigger collection.

Definition C.5 (Process in parallel with a trigger or a characteristic process). We write $P \parallel Q$ to stand for $P \mid Q$ where either P or Q is a trigger collection.

P	$C_{\tilde{x}}^{\tilde{u}}(P)$	
$Q_1 \parallel Q_2$	$\{R_1 \parallel R_2 : R_1 \in C_{\tilde{y}}^{\tilde{u}_1}(Q_1), R_2 \in C_{\tilde{z}}^{\tilde{u}_2}(Q_2)\}$	$\tilde{y} = \text{fnb}(Q_1, \tilde{x}), \tilde{z} = \text{fnb}(Q_2, \tilde{x})$ $\{\tilde{u}/\tilde{x}\} = \{\tilde{u}_1/\tilde{y}\} \cdot \{\tilde{u}_2/\tilde{z}\}$
$(\nu s : C)Q$	$\{(\nu \tilde{s} : \mathcal{G}(C)) R : C_{\tilde{x}}^{\tilde{u}}(Q\sigma)\}$	$\tilde{s} = (s_1, \dots, s_{ \mathcal{G}(C) })$ $\sigma = \{s_1 \tilde{s}_1 / s \tilde{s}\}$
Q	$\{(\nu \tilde{c}) R : R = \overline{c_k}! \langle \tilde{u} \rangle \mid \mathcal{B}_{\tilde{x}}^k(Q)\} \cup \{(\nu \tilde{c}) R : R \in \mathcal{J}_{\tilde{x}}^{\tilde{u}}(Q)\}$	$\tilde{c} = \text{fpn}(R)$
$\mathbf{0}$	$\{\mathbf{0}\}$	
P	$\mathcal{J}_{\tilde{x}}^{\tilde{u}}(P)$	
$n_i?(y).Q$	$\{n_i \rho?(\tilde{y}).\overline{c_{k+1}}! \langle \tilde{z} \rho \rangle \mid \mathcal{B}_{\tilde{z}}^{k+1}(Q\sigma)\}$	$y : S \wedge \tilde{y} = (y_1, \dots, y_{ S })$ $\tilde{w} = (\text{lin}(n_i))? \{n_i\} : \epsilon$ $\tilde{z} = \text{fnb}(Q, \tilde{x} \setminus \tilde{w})$ $\rho = \{\tilde{u}/\tilde{x}\}$ $\sigma = \text{next}(n_i) \cdot \{y_1/y\}$ $l = (\text{tr}(S))? f(S) : i$
$n_i!(y_j).Q$	$\{n_i \rho! \langle \tilde{y} \rho \rangle.\overline{c_{k+1}}! \langle \tilde{z} \rho \rangle \mid \mathcal{B}_{\tilde{z}}^{k+1}(Q\sigma)\}$	$y : S \wedge \tilde{y} = (y_j, \dots, y_{j+ \mathcal{G}(S) -1})$ $\tilde{w} = (\text{lin}(n_i))? \{n_i\} : \epsilon$ $\tilde{z} = \text{fnb}(Q, \tilde{x} \setminus \tilde{w})$ $\rho = \{\tilde{u}/\tilde{x}\}$ $\sigma = \text{next}(n_i)$ $l = (\text{tr}(S))? f(S) : i$
$Q_1 \mid Q_2$	$\{\overline{c_k}! \langle \tilde{y} \rho \rangle.\overline{c_{k+1}}! \langle \tilde{z} \rho \rangle \mid \mathcal{B}_{\tilde{y}}^k(Q_1) \mid \mathcal{B}_{\tilde{z}}^{k+1}(Q_2)\}$ \cup $\{(R_1 \mid R_2) : C_{\tilde{y}}^{\tilde{u}_1}(Q_1), R_2 \in C_{\tilde{z}}^{\tilde{u}_2}(Q_2)\}$	$\tilde{y} = \text{fnb}(Q_1, \tilde{x})$ $\tilde{z} = \text{fnb}(Q_2, \tilde{x})$ $\rho = \{\tilde{u}/\tilde{x}\}$ $l = \lfloor Q_1 \rfloor^*$
$\mu X.P$	$\{\overline{c_{k+1}}! \langle \tilde{z} \rho \rangle.\mu X.c_X^r?(\tilde{y}).\overline{c_{k+1}}! \langle \tilde{y} \rangle.X \mid$ $\mathcal{A}_{\text{rec } \tilde{x}}^{k+1}(P)g\}$	$\tilde{n} = \text{fn}(P)$ $\tilde{n} : \tilde{C} \wedge \tilde{m} = \text{bn}(\tilde{n} : \tilde{C})$ $\tilde{z} = \tilde{x} \cup \tilde{m}, \tilde{z} = \tilde{y} $ $g = \{X \mapsto \tilde{m}\}$
$\mathbf{0}$	$\{\mathbf{0}\}$	

Table 4: The definition of $C_{\tilde{x}}^{\tilde{u}}(P)$ and $\mathcal{J}_{\tilde{x}}^{\tilde{u}}(P)$ (core fragment).

Definition C.6 (Propagators of P). We define $\text{fpn}(P)$ to denote the set of free propagator names in P .

Definition C.7 (Depth of recursive variable). Let P be a recursive process. We define $\text{recd}(P)$ to count sequential prefixes in P preceding recursive variable X or a subprocess of shape $\mu X.Q$. Function $\text{recd}(P)$ and $\text{recd}'(P)$ are mutually defined as follows:

$$\text{recd}(P) = \begin{cases} \text{recd}'(P) & \text{if } \text{frv}(P) \\ 0 & \text{otherwise} \end{cases}$$

$$\text{recd}'(P) = \begin{cases} \text{recd}(Q) + 1 & \text{if } P = \alpha.Q \\ \text{recd}(Q) + \text{recd}(R) & \text{if } P = Q \mid R \\ 0 & \text{if } P = \mu X.Q \text{ or } P = X \\ \text{recd}(Q) & \text{if } P = (\nu s)Q \end{cases}$$

Definition C.8 (Name breakdown). Let $u_i : C$ be an indexed name with its session type. We write $\text{bn}(u_i : C)$ to denote

$$\text{bn}(u_i : C) = (u_i, \dots, u_{i+|\mathcal{G}(C)|-1})$$

Function $\text{bn}(\cdot)$ is extended to work on a list element-wise, as follows:

$$\text{bn}((u_i^1, \dots, u_j^n) : (C_1, \dots, C_n)) = \text{bn}(u_i^1 : C_1) \cdot \dots \cdot \text{bn}(u_j^n : C_n)$$

P	$C_{\tilde{x}}^{\tilde{u}}(P)$	
Q	$C_{\text{rec } \tilde{x}}^{\tilde{u}}(\mu X.Q^*)^d$ if $Q \equiv \alpha_d.\alpha_{d-1}.\dots.\alpha_{d-p}.(P_X \mid R)$ and $\mu X.Q^*$ is a subprocess of P_X $\{(v\tilde{c})R : R \in \{\overline{c_k}!\langle\tilde{u}\rangle \mid \mathcal{B}_{\tilde{x}}^k(Q), c_k^r!\langle\tilde{u}\rangle \mid A_{\text{rec } \tilde{x}}^k(Q)\}\}$ $\cup \{(v\tilde{c})R : R \in \mathcal{J}_{\text{rec } \tilde{x}, \rho}^{\tilde{u}}(Q) \cup \mathcal{J}_{\text{rec } \tilde{x}, \rho}^k(Q)\}_g^d$ otherwise	$\tilde{c} = \text{fpn}(R)$ $\rho = \{\tilde{u}/\tilde{x}\}$
$C_{\text{rec } \tilde{x}}^{\tilde{u}}(Q)^d$	$\{(v\tilde{c})(\mu X.c_X^r?(\tilde{y}).\overline{c_{k+1}^r}!\langle\tilde{y}\rangle.X \mid R \mid c_k^r?(\tilde{x}).c_X^r!\langle\tilde{x}\rangle.X : R \in \mathcal{J}_{\text{rec } \tilde{x}}^k(Q)_g^d) \cup N$ where: $M = \begin{cases} \{(v\tilde{c})(\mu X.c_X^r?(\tilde{y}).\overline{c_{k+1}^r}!\langle\tilde{y}\rangle.X \mid R \mid \\ c_X^r!\langle\tilde{x}\rho\rangle.\mu X.c_k^r?(\tilde{x}).c_X^r!\langle\tilde{x}\rangle.X) : R \in \mathcal{J}_{\text{rec } \tilde{x}}^k(Q)_g^0\} & \text{if } g \neq \emptyset \\ \{(v\tilde{c})(\mu X.c_X^r?(\tilde{y}).\overline{c_{k+1}^r}!\langle\tilde{y}\rangle.X \mid R \mid \\ c_X^r!\langle\tilde{y}\rangle \mid \mu X.c_k^r?(\tilde{x}).(c_X^r!\langle\tilde{y}\rangle \mid X)) : R \in \mathcal{J}_{\text{rec } \tilde{x}}^k(Q)_g^0\} & \text{otherwise} \end{cases}$ $N = \begin{cases} \{(v\tilde{c})(\overline{c_k^r}!\langle\tilde{z}\rho\rangle.\mu X.c_X^r?(\tilde{y}).\overline{c_k^r}!\langle\tilde{y}\rangle.X \mid A_{\text{rec } \tilde{x}}^k(Q)_g)\} \cup M & \text{if } d = 0 \\ \emptyset & \text{otherwise} \end{cases}$	$\tilde{c} = \text{fpn}(R)$ $Q \equiv \mu X.Q^*$ $\tilde{x} = \text{fs}(Q^*)$ $g = \{X \mapsto \tilde{x}\}$ $\rho = \{\tilde{u}/\tilde{x}\}$
P	$\mathcal{J}_{\text{rec } \tilde{x}, \rho}^k(P)_g^d$ when $g \neq \emptyset$	
$n_i!\langle y_j \rangle.Q$	$\{n_l!\langle\tilde{y}\rho\rangle.\overline{c_{k+1}^r}!\langle\tilde{z}\rho\rangle.\mu X.c_k^r?(\tilde{x}).n_l!\langle\tilde{y}\rangle.\overline{c_{k+1}^r}!\langle\tilde{z}\rangle.X \mid A_{\text{rec } \tilde{z}}^{k+1}(Q\sigma)_g\}$ if $\text{recd}(Q) = d$ $N \cup \{\mu X.c_k^r?(\tilde{x}).n_l!\langle\tilde{y}\rangle.\overline{c_{k+1}^r}!\langle\tilde{z}\rangle.X \mid R : R \in \mathcal{J}_{\text{rec } \tilde{z}, \rho}^{k+1}(Q\sigma)_g^d\}$ otherwise where: $N = \begin{cases} \{\overline{c_{k+1}^r}!\langle\tilde{z}\rho\rangle.\mu X.c_k^r?(\tilde{x}).n_l!\langle\tilde{y}\rangle.\overline{c_{k+1}^r}!\langle\tilde{z}\rangle.X \mid A_{\text{rec } \tilde{z}}^{k+1}(Q\sigma)_g\} & \text{if } \text{recd}(Q) = d + 1 \\ \emptyset & \text{otherwise} \end{cases}$	$y : T \wedge (y_j, \dots, y_{j+ \mathcal{G}(T) -1})$ $\tilde{w} = (\text{lin}(u_i)) ? \{u_i\} : \epsilon$ $\tilde{z} = g(X) \cup \text{fnb}(Q, \tilde{x}\tilde{y} \setminus \tilde{w})$ $l = (\text{tr}(u_i)) ? f(S) : i$ $\sigma = \text{next}(u_i)$
$n_i?(y).Q$	$\{n_l!\langle\tilde{y}\rangle.\overline{c_{k+1}^r}!\langle\tilde{z}\rho\rangle.\mu X.c_k^r?(\tilde{x}).n_l!\langle\tilde{y}\rangle.\overline{c_{k+1}^r}!\langle\tilde{z}\rangle.X \mid A_{\text{rec } \tilde{z}}^{k+1}(Q\sigma)_g\}$ if $\text{recd}(Q) = d$ $N \cup \{\mu X.c_k^r?(\tilde{x}).n_l!\langle\tilde{y}\rangle.\overline{c_{k+1}^r}!\langle\tilde{z}\rangle.X \mid R : R \in \mathcal{J}_{\text{rec } \tilde{z}, \rho}^{k+1}(Q\sigma)_g^d\}$ $N = \begin{cases} \{\overline{c_{k+1}^r}!\langle\tilde{z}\rho\rangle.\mu X.c_k^r?(\tilde{x}).n_l!\langle\tilde{y}\rangle.\overline{c_{k+1}^r}!\langle\tilde{z}\rangle.X \mid A_{\text{rec } \tilde{z}}^{k+1}(Q\sigma)_g\} & \text{if } \text{recd}(Q) = d + 1 \\ \emptyset & \text{otherwise} \end{cases}$	$\tilde{w} = (\text{lin}(u_i)) ? \{u_i\} : \epsilon$ $\tilde{z} = g(X) \cup \text{fnb}(Q, \tilde{x}\tilde{y} \setminus \tilde{w})$ $l = (\text{tr}(u_i)) ? f(S) : i$ $\sigma = \text{next}(u_i) \cdot \{y_l/y\}$
$Q_1 \mid Q_2$	$N \cup \{\mu X.c_k^r?(\tilde{x}).(\overline{c_{k+1}^r}!\langle\tilde{y}_1\rangle.X \mid c_{k+l+1}^r!\langle\tilde{y}_2\rangle) \mid R_1 \mid R_2 : \\ R_1 \in \mathcal{J}_{\text{rec } \tilde{y}_1, \rho}^{k+1}(Q_1)_{g_1}^d, R_2 \in \mathcal{J}_{\text{rec } \tilde{y}_2, \rho}^{k+l+1}(Q_2)_{g_2}^d\}$ $N = \begin{cases} \{\overline{c_{k+1}^r}!\langle\tilde{y}_1\rho\rangle.\mu X.c_k^r?(\tilde{x}).(\overline{c_{k+1}^r}!\langle\tilde{y}_1\rangle.X \mid c_{k+l+1}^r!\langle\tilde{y}_2\rangle) \mid \\ A_{\text{rec } \tilde{y}_1}^{k+1}(Q_1)_{g_1} \mid A_{\text{rec } \tilde{y}_2}^{k+l+1}(Q_2)_{g_2}\} & \text{if } \text{recd}(Q_1) = d \\ \emptyset & \text{otherwise} \end{cases}$	$\text{frv}(Q_1)$ $\tilde{y}_1 = g(X) \cup \text{fnb}(Q_1, \tilde{x})$ $\tilde{y}_2 = \text{fnb}(Q_2, \tilde{x})$ $l = \lfloor Q_1 \rfloor^*$
$(v s : C)Q$	$N \cup \{\mu X.(v\tilde{s} : \mathcal{G}(C))c_k?(\tilde{x}).\overline{c_{k+1}^r}!\langle\tilde{z}\rangle.X \mid R : R \in \mathcal{J}_{\text{rec } \tilde{z}, \rho}^{k+1}(Q\sigma)_g^d\}$ $R = \overline{c_{k+1}^r}!\langle\tilde{z}\rho\rangle.\mu X.(v\tilde{s} : \mathcal{G}(C))c_k?(\tilde{x}).\overline{c_{k+1}^r}!\langle\tilde{z}\rangle.X$ $N = \begin{cases} \{(v\tilde{s} : \mathcal{G}(C))R \mid A_{\text{rec } \tilde{z}}^{k+1}(Q\sigma)_g\} & \text{if } \text{recd}(Q_1) = d \\ \emptyset & \text{otherwise} \end{cases}$	$\tilde{s} = (s_1, \dots, s_{ \mathcal{G}(S) })$ $\tilde{s} = (\overline{s_1}, \dots, \overline{s_{ \mathcal{G}(S) }})$ $\tilde{n} = (\text{lin}(s)) ? \tilde{n} : \epsilon$ $\tilde{z} = \tilde{x}, \tilde{s}, \tilde{n}$ $\sigma = \{s_1\overline{s_1}/\tilde{s}\tilde{s}\}$
X	$\mathbf{0}$	
P	$\mathcal{J}_{\text{rec } \tilde{x}, \rho}^k(P)_g^d$ when $g = \emptyset$	
$n_i!\langle y_j \rangle.Q$	$\{n_l!\langle\tilde{y}\rho\rangle.\overline{c_{k+1}^r}!\langle\tilde{z}\rho\rangle \mid \mu X.c_k^r?(\tilde{x}).(n_l!\langle\tilde{y}\rangle.\overline{c_{k+1}^r}!\langle\tilde{z}\rangle \mid X) \mid A_{\text{rec } \tilde{z}}^{k+1}(Q\sigma)_g\}$	
$n_i?(y).Q$	$\{n_l!\langle\tilde{y}\rangle.\overline{c_{k+1}^r}!\langle\tilde{z}\rho\rangle \mid \mu X.c_k^r?(\tilde{x}).(n_l!\langle\tilde{y}\rangle.\overline{c_{k+1}^r}!\langle\tilde{z}\rangle \mid X) \mid A_{\text{rec } \tilde{z}}^{k+1}(Q\sigma)_g\}$	
$Q_1 \mid Q_2$	$\{\overline{c_{k+1}^r}!\langle\tilde{y}_1\rho\rangle \mid \overline{c_{k+l+1}^r}!\langle\tilde{y}_2\rho\rangle \mid \mu X.c_k^r?(\tilde{x}).(\overline{c_{k+1}^r}!\langle\tilde{y}_1\rangle \mid \overline{c_{k+l+1}^r}!\langle\tilde{y}_2\rangle) \mid X \mid \\ A_{\text{rec } \tilde{y}_1}^{k+1}(Q_1)_{g_1} \mid A_{\text{rec } \tilde{y}_2}^{k+l+1}(Q_2)_{g_2}\}$	
$(v s : C)Q$	$\{\overline{c_{k+1}^r}!\langle\tilde{z}\rho\rangle \mid \mu X.(v\tilde{s} : \mathcal{G}(C))c_k?(\tilde{x}).(\overline{c_{k+1}^r}!\langle\tilde{z}\rangle \mid X) \mid A_{\text{rec } \tilde{z}}^{k+1}(Q)_g\}$	

Table 5: The extension of $C_{\tilde{x}}^{\tilde{u}}(P)$ and definition of $\mathcal{J}_{\text{rec } \tilde{x}, \rho}^k(P)_g^d$ (recursion extension). Side conditions for last four rows ($g = \emptyset$) are the same as for corresponding cases when $g \neq \emptyset$.

Definition C.9 (Indexed names substitutions). Let

$$\tilde{u} = (a, b, s, \bar{s}, s', r, r', \dots)$$

be a finite tuple of names. We write $\text{index}(\tilde{u})$ to denote

$$\text{index}(\tilde{u}) = \{\{a_1, b_1, s_i, \bar{s}_i, s'_j, \bar{s}'_j, r_1, r'_1, \dots / a, b, s, s', r, r', \dots\} : i, j, \dots > 0\}$$

Definition C.10 (Relation \mathcal{S}). Let $P\{\tilde{u}/\tilde{x}\}$ be a well-typed process and let the \mathcal{C} -set be as in Tab. 4. We define the relation \mathcal{S} as follows:

$$\begin{aligned} \mathcal{S} = \{ & (P\{\tilde{u}/\tilde{x}\}, R) : R \in C_{\tilde{x}}^{\tilde{u}_*}(P\sigma) \\ & \text{with } \tilde{n} = \text{fn}(P) \cup \tilde{u} \cup \tilde{x}, \sigma \in \text{index}(\tilde{n}), \\ & \tilde{u}_* = \text{bn}(\tilde{u}\sigma : \tilde{C}), \tilde{x}_* = \text{bn}(\tilde{x}\sigma : \tilde{C}) \} \end{aligned}$$

where $\tilde{u} : \tilde{C}$.

For the convenience, we define function $\widehat{C}_-^-(\cdot)$ relying on $C_-^-(\cdot)$ as follows:

Definition C.11 (Function $\widehat{C}_-^-(\cdot)$). Let P be a process, ρ be a name substitution, and σ be an indexed name substitution. We define $\widehat{C}_\sigma^\rho(P)$ as follows:

$$\begin{aligned} \widehat{C}_\sigma^\rho(P) &= C_{\tilde{x}_*}^{\tilde{u}_*}(P\sigma) \\ & \text{with } \rho = \{\tilde{u}/\tilde{x}\}, \tilde{u}_* = \text{bn}(\tilde{u}\sigma : \tilde{C}), \tilde{x}_* = \text{bn}(\tilde{x}\sigma : \tilde{C}) \end{aligned}$$

where $\tilde{u} : \tilde{C}$.

C.7 Operational correspondence

LEMMA C.12. *Given an indexed process $P_1\{\tilde{u}/\tilde{x}\}$, the set $C_{\tilde{x}}^{\tilde{u}}(P_1)$ is closed under τ transitions on non-essential prefixes. That is, if $R_1 \in C_{\tilde{x}}^{\tilde{u}}(P_1)$ and $R_1 \xrightarrow{\tau} R_2$ is inferred from the actions on non-essential prefixes, then $R_2 \in C_{\tilde{x}}^{\tilde{u}}(P_1)$.*

SKETCH. By the induction on the structure of P_1 and the inspection of definition of $C_-^-(\cdot)$ and $\mathcal{J}_-^-(\cdot)$ in Tab. 4 and Tab. 5. \square

LEMMA C.13. *Relation \diamond is a MST bisimulation.*

SKETCH. Straightforward by the transition induction and Def. C.1. \square

Lemma C.7. *Assume $P_1\{\tilde{u}/\tilde{x}\}$ is a well-formed process and $P_1\{\tilde{u}/\tilde{x}\} \mathcal{S} Q_1$.*

(1) *Whenever $P_1\{\tilde{u}/\tilde{x}\} \xrightarrow{(v\tilde{m}_1)n!(v:C_1)} P_2$ then there exist Q_2 and σ_v such that $Q_1 \xrightarrow{(v\tilde{m}_2)\tilde{n}!(\tilde{v}:\mathcal{G}(C_1))} Q_2$ where $v\sigma_v \vDash_C \tilde{v}$ and, for a fresh t ,*

$$(v\tilde{m}_1)(P_2 \parallel t \Leftarrow_C v : C_1) \mathcal{S} (v\tilde{m}_2)(Q_2 \parallel t \Leftarrow_m v\sigma_v : C_1)$$

(2) *Whenever $P_1\{\tilde{u}/\tilde{x}\} \xrightarrow{n?(v)} P_2$ then there exist Q_2 and σ_v such that $Q_1 \xrightarrow{\tilde{n}?(v)} Q_2$ where $v\sigma_v \vDash \tilde{v}$ and $P_2 \mathcal{S} Q_2$,*

(3) *Whenever $P_1 \xrightarrow{\tau} P_2$ then there exists Q_2 such that $Q_1 \xrightarrow{\tau} Q_2$ and $P_2 \mathcal{S} Q_2$.*

PROOF. By transition induction. Let $\rho_1 = \{\tilde{u}/\tilde{x}\}$. By inversion of $P_1 \mathcal{S} Q_1$ we know there is $\sigma_1 \in \text{index}(\text{fn}(P_1) \cup \tilde{u} \cup \tilde{x})$, such that $Q_1 \in \widehat{C}_{\sigma_1}^{\rho_1}(P_1)$. Then, if $P_1 \xrightarrow{\ell} P_2$ then we show there exists Q_2 such that $Q_1 \xrightarrow{\check{\ell}} Q_2$ and $Q_2 \in \widehat{C}_{\sigma_2}^{\rho_2}(P_2)$ for some ρ_2 , and for some σ_2' we have $\sigma_2 = \sigma_2' \cdot \sigma_1 \cdot \{n_{i+1}/n_i\}$ with $n_i = \text{subj}(\check{\ell})$. First, we consider two base cases: Rules Snd and Rv. Then, we consider inductive cases: Rules Par $_{\perp}$ and Tau. We omit inductive cases New and Res as they follow directly by the inductive hypothesis and the definition of restriction case of $C_-^-(\cdot)$ (Tab. 4). Finally, we separately treat cases when a process is recursive. We show two cases (Rv and Par $_{\perp}$) emphasizing specifics of the recursion breakdown.

(1) Case Snd. We distinguish two sub-cases: (i) $P_1 = n!(v).P_2$ and (ii) $P_1 = n!(y).P_2$ with $\{v/y\} \in \rho_1$. In both sub-cases we know there is $\rho_2 = \{\tilde{u}_2/\tilde{x}_2\}$ such that

$$P_1\rho_1 = n\rho_1!(v).P_2\rho_2$$

We have the following transition:

$$\langle \text{Snd} \rangle \frac{}{P_1\rho_1 \xrightarrow{n\rho_1!(v)} P_2\rho_2}$$

Let $\sigma_1 = \sigma' \cdot \{n_i/n\}$ where $\sigma' \in \text{index}(\text{fn}(P_1) \cup \tilde{u} \cup \tilde{x})$ and $\sigma_2 = \sigma_1 \cdot \sigma$ where $\sigma = \text{next}(n_i)$. Here we take $\sigma_v = \sigma_2$. Further, let $\tilde{u}_* = \text{bn}(\tilde{u}\sigma_1 : \tilde{C})$ and $\tilde{x}_* = \text{bn}(\tilde{x}\sigma_1 : \tilde{C})$ with $\tilde{u} : \tilde{C}$. Also, let $\tilde{z} = \text{fnb}(P_2, \tilde{x}_* \setminus \tilde{w})$ where $\tilde{w} = \{n_i\}$ if $\text{lin}(n_i)$ otherwise $\tilde{w} = \epsilon$. Then, in both sub-cases, there are two possibilities for the shape of Q_1 , namely:

$$\begin{aligned} Q_1^1 &= (v \tilde{c}_k)(\overline{c_k!} \langle \tilde{u}_* \rangle \mid \mathcal{B}_{\tilde{x}_*}^k(P_1\sigma_1)) \\ Q_1^2 &= (v \tilde{c}_{k+1})n_i\rho! \langle \tilde{v} \rangle . \overline{c_{k+1}!} \langle \tilde{z}\rho \rangle \mid \mathcal{B}_{\tilde{z}}^{k+1}(P_2\sigma_2) \end{aligned}$$

where $v\sigma_v \triangleright_{\epsilon} \tilde{v}$. By Lem. C.12 we know that $Q_1^1 \xrightarrow{\tau} Q_1^2$. Thus, we only consider how Q_1^2 evolves. We can infer the following:

$$Q_1^2 \xrightarrow{n_i\rho! \langle \tilde{v} \rangle} Q_2$$

where $Q_2 = (v \tilde{c}_{k+1})\overline{c_{k+1}!} \langle \tilde{z}\rho \rangle \mid \mathcal{B}_{\tilde{z}}^{k+1}(P_2\sigma_2)$. Then, we should show the following:

$$(P_2 \parallel t \leftarrow_{\epsilon} v : C_1)\rho_2 \mathcal{S} Q_2 \parallel t \leftarrow_m v\sigma_v : C_1 \quad (132)$$

By Tab. 4 we can see that

$$Q_2 \in C_{\tilde{z}}^{\tilde{z}\rho_2}(P_2\sigma_2) \quad (133)$$

Here we remark that by construction $\sigma_2 = \sigma_1 \cdot \{n_{i+1}/n_i\}$.

By Def. 4.6 and Def. C.8 we know that

$$\tilde{z} = \text{fnb}(P_2, \text{bn}(\tilde{x} : \tilde{C}) \setminus \tilde{w}) = \text{bn}(\tilde{x}_2\sigma_1 \cdot \sigma : \tilde{C}_2)$$

with $\tilde{x}_2 : \tilde{C}_2$. Similarly, we have

$$\tilde{z}\rho = \text{bn}(\tilde{x}_2\rho_2\sigma_1 \cdot \sigma : \tilde{C}_2)$$

Further, we know there is

$$\sigma'_2 \in \text{index}(\text{fn}(P_2) \cup \tilde{u}_2 \cup \tilde{x}_2)$$

such that $P_2\{\tilde{u}_2/\tilde{x}_2\}\sigma_2 = P_2\{\tilde{u}_2/\tilde{x}_2\}\sigma'_2$. Thus, by this and (133) we have that $P_2\rho_2 \mathcal{S} Q_2$. Next, by the definition we have $t \leftarrow_{\epsilon} v : C_1 \mathcal{S} t \leftarrow_m v\rho\sigma : C_1$. Thus, the goal (132) follows. This concludes Snd case.

(2) Case Rv. Here we know $P_1 = n?(y).P_2$. We know there is $\rho_2 = \{\tilde{u}_2/\tilde{x}_2\}$ such that

$$P_1\rho_1 = n\rho_1?(y).P_2\rho_2$$

The transition is as follows:

$$\langle \text{Rv} \rangle \frac{}{n\rho_1?(y).P_2\rho_2 \xrightarrow{n?(v)} P_2\rho_2 \cdot \{v/y\}}$$

Let $\sigma_1 = \sigma' \cdot \{n_i/n\}$ where $\sigma' \in \text{index}(\text{fn}(P_1) \cup \tilde{u} \cup \tilde{x})$ and $\sigma_2 = \sigma_1 \cdot \sigma$ where $\sigma = \text{next}(n_i) \cdot \{y_1/y\}$. Further, let $\tilde{u}_* = \text{bn}(\tilde{u}\sigma_1 : \tilde{C})$ and $\tilde{x}_* = \text{bn}(\tilde{x}\sigma_1 : \tilde{C})$ with $\tilde{u} : \tilde{C}$. Also, let $\tilde{y} = (y_1, \dots, y_{|\mathcal{G}(S)|})$ and $\tilde{z} = \text{fnb}(P_2, \tilde{x}_* \setminus \tilde{w})$ where $\tilde{w} = \{n_i\}$ if $\text{lin}(n_i)$ otherwise $\tilde{w} = \epsilon$. Then, there are two possibilities for the shape of Q_1 , namely:

$$\begin{aligned} Q_1^1 &= (v \tilde{c}_k)(\overline{c_k!} \langle \tilde{u}_* \rangle \mid \mathcal{B}_{\tilde{x}_*}^k(P_1\sigma_1)) \\ Q_1^2 &= (v \tilde{c}_{k+1})n_i\rho?(y) \cdot \overline{c_{k+1}!} \langle \tilde{z}\rho \rangle \mid \mathcal{B}_{\tilde{z}}^{k+1}(P_2\sigma_2) \end{aligned}$$

By Lem. C.12 we know that $Q_1^1 \xrightarrow{\tau} Q_1^2$. Thus, we only consider how Q_1^2 evolves. We can infer the following:

$$Q_1^2 \xrightarrow{n_i\rho?(y)} Q_2$$

where $Q_2 = (v \tilde{c}_{k+1})\overline{c_{k+1}!} \langle \tilde{z}\rho \cdot \{\tilde{v}/\tilde{y}\} \rangle \mid \mathcal{B}_{\tilde{z}}^{k+1}(P_2\sigma_2)$ and $v\sigma_v \triangleright_{\epsilon} \tilde{v}$ for some $\sigma_v \in \text{index}(v)$. Now, we should show the following:

$$P_2\rho_2 \cdot \{v/y\} \mathcal{S} Q_2 \quad (134)$$

By Tab. 4 we can see that

$$Q_2 \in C_{\tilde{z}}^{\tilde{z}\rho_2 \cdot \{\tilde{v}/\tilde{y}\}}(P_2\sigma_2 \cdot \sigma_v) \quad (135)$$

Here we remark that $\sigma_2 \cdot \sigma_v = \sigma_v \cdot \sigma_1 \cdot \{n_{i+1}/n_i\}$. Further, we may notice that

$$\sigma_2 \cdot \sigma_v \in \text{index}(\text{fn}(P_2) \cup \tilde{u}_2 \cup \tilde{x}_2)$$

By Def. 4.6 and Def. C.8 we have that

$$\begin{aligned} \tilde{z} &= \text{fnb}(P_2, \text{bn}(\tilde{x} : \tilde{C}) \setminus \tilde{y} \setminus \tilde{w}) \\ &= \text{bn}(\tilde{x}\tilde{y}(\sigma_1 \cdot \sigma) : \tilde{C}S) \end{aligned}$$

with $y : S$. Similarly, we have

$$\bar{z}\rho_2 \cdot \{\tilde{v}/\tilde{y}\} = \text{bn}(\bar{x}y\rho_2 \cdot \{v/y\}(\sigma_2 \cdot \sigma_v) : \bar{C}S)$$

Thus, by this and (135) the goal (134) follows. This concludes Rv case (and the base cases). We remark that base cases concerning triggers collection processes follow by Lem. C.13. Next, we consider inductive cases.

(3) Case Par_L . Here we know $P_1 = P'_1 \mid P''_1$. The final rule in the inference tree is as follows:

$$\langle \text{Par}_L \rangle \frac{P'_1\{\tilde{u}'_1/\tilde{y}_1\} \xrightarrow{\ell} P'_2\{\tilde{u}'_2/\tilde{y}_2\} \quad \text{bn}(\ell) \cap \text{fn}(P''_1) = \emptyset}{P'_1\{\tilde{u}'_1/\tilde{y}_1\} \mid P''_1\{\tilde{u}''_1/\tilde{z}_1\} \xrightarrow{\ell} P'_2\{\tilde{u}'_2/\tilde{y}_2\} \mid P''_1\{\tilde{u}''_1/\tilde{z}_2\}}$$

Let $\sigma_1 \in \text{index}(\text{fn}(P_1) \cup \tilde{u} \cup \tilde{x})$. Further, let $\tilde{y} = \text{fnb}(P'_1, \tilde{x}_*)$, $\tilde{z} = \text{fnb}(P''_1, \tilde{x}_*)$, $\tilde{u}_* = \text{bn}(\tilde{u}\sigma_1 : \bar{C})$, $\tilde{x}_* = \text{bn}(\tilde{x}\sigma_1 : \bar{C})$ with $\tilde{u} : \bar{C}$, and $\rho_m = \{\tilde{u}_*/\tilde{x}_*\}$. In sub-case (i), by the definition of \mathcal{S} (Tab. 4), there are following possibilities for Q_1 :

$$\begin{aligned} Q_1^1 &= (v \bar{c}_k)(\bar{c}_k! \langle \tilde{u}_* \rangle \mid \mathcal{B}_{\tilde{x}_*}^k(P_1\sigma_1)) \\ Q_1^2 &= \bar{c}_k! \langle \tilde{y}\rho_m \rangle . \bar{c}_k! \langle \tilde{z}\rho_m \rangle \mid \mathcal{B}_{\tilde{y}}^k(P'_1\sigma_1) \mid \mathcal{B}_{\tilde{z}}^{k+l}(P''_1\sigma_1) \\ N_1^3 &= \{(R'_1 \mid R''_1) : C_{\tilde{y}}^{\tilde{y}\rho_m}(P'_1\sigma_1), R''_1 \in C_{\tilde{z}}^{\tilde{z}\rho_m}(P''_1\sigma_1)\} \end{aligned}$$

By Lem. C.12 there exist

$$Q'_1 \in C_{\tilde{y}}^{\tilde{y}\rho_m}(P'_1\sigma_1) \quad (136)$$

$$Q''_1 \in C_{\tilde{z}}^{\tilde{z}\rho_m}(P''_1\sigma_1) \quad (137)$$

such that

$$Q_1^1 \xrightarrow{\tau} Q_1^2 \xrightarrow{\tau} Q'_1 \mid Q''_1$$

Thus, in both cases we consider how $Q'_1 \mid Q''_1$ evolves. By the definition of \mathcal{S} we have

$$P'_1\rho'_1 \mathcal{S} Q'_1 \quad (138)$$

$$P''_1\rho''_1 \mathcal{S} Q''_1 \quad (139)$$

To apply IH we do the case analysis on the action ℓ :

- Sub-case $\ell \equiv n? \langle v \rangle$. If $v \in \tilde{u}$ then we take $\sigma_v = \sigma_1$, otherwise $\sigma_v = \{v_j/v\}$ for $j > 0$. Then, by applying IH to (138) we know there is Q'_2 such that $Q'_1 \xrightarrow{n? \langle \tilde{v} \rangle} Q'_2$ and

$$P'_2\{\tilde{u}'_2/\tilde{y}_2\} \mathcal{S} Q'_2 \quad (140)$$

where $v\sigma_v \triangleright_c \bar{v}$ and $\tilde{u}'_2 = \tilde{u}'_1 \cdot \bar{v}$. We should show that

$$P'_2\rho'_2 \mid P''_1\rho''_1 \mathcal{S} Q'_2 \mid Q''_1 \quad (141)$$

where $\rho'_2 = \{\tilde{u}'_2/\tilde{y}_2\}$ and $\rho''_1 = \{\tilde{u}''_1/\tilde{z}_1\}$. Let $\sigma_2 = \sigma_1 \cdot \text{next}(n_i)$ such that $\{n_i/n\} \in \sigma_1$. Then, by (140) we know

$$Q'_2 \in \widehat{C}_{\sigma_2 \cdot \sigma_v}^{\rho'_2}(P'_2) \quad (142)$$

By the definition of σ_v and σ_2 we may see that

$$\widehat{C}_{\sigma_2 \cdot \sigma_v}^{\rho'_1}(P''_1) = \widehat{C}_{\sigma_1}^{\rho''_1}(P''_1)$$

So, by (142) and (137) we have

$$Q'_2 \mid Q''_1 \in \widehat{C}_{\sigma_2 \cdot \sigma_v}^{\rho'_2 \cdot \rho''_1}(P'_2 \mid P''_1) \quad (143)$$

Further, we may notice that $\sigma_2 \cdot \sigma_v \in \text{index}(\text{fn}(P'_2 \mid P''_1) \cup \rho'_2 \cup \rho''_1)$. So, by this and (143) the goal (141) follows.

- Sub-case $\ell = \tau$. By applying IH to (138) we know there is Q'_2 such that $Q'_1 \xrightarrow{\ell} Q'_2$ and

$$P'_2\rho'_2 \mathcal{S} Q'_2 \quad (144)$$

where $\rho'_2 = \{\tilde{u}'_2/\tilde{y}_2\}$. We should show that

$$P'_2\rho'_2 \mid P''_1\rho''_1 \mathcal{S} Q'_2 \mid Q''_1 \quad (145)$$

By (144), we know there is $\sigma_2 \in \text{index}(\text{fn}(P_2) \cup \rho'_2)$ such that

$$Q'_2 \in \widehat{C}_{\sigma_2}^{\rho'_2}(P'_2)$$

Further, by remark we know that either $\sigma'_2 = \sigma_1 \cdot \{n_{i+1}/n_i\} \cdot \{\bar{n}_{i+1}/\bar{n}_i\}$ for some $n_i, \bar{n}_i \in \text{fn}(P'_2 \rho'_2)$ and $n_i, \bar{n}_i \notin \text{fn}(P''_1 \rho''_1)$ or $\sigma'_2 = \sigma_1$ such that

$$\widehat{C}_{\sigma'_2}^{\rho'_2}(P'_2) = \widehat{C}_{\sigma'_2}^{\rho'_2}(P'_2)$$

By this we have that

$$\widehat{C}_{\sigma'_2}^{\rho'_1}(P'_1) = \widehat{C}_{\sigma_1}^{\rho'_1}(P'_1)$$

So, we know that

$$Q'_2 \mid Q''_1 \in \widehat{C}_{\sigma'_2}^{\rho'_2 \cdot \rho''_1}(P'_2 \mid P''_1)$$

By (144) and (137) and the definition of σ'_2 the goal (145) follows.

- Sub-case $\ell \equiv (\nu \bar{m}_1) n! \langle v : C_1 \rangle$. Here we omit details on substitutions as they are similar to the first sub-case. By applying IH to (138) we know there is Q'_2 such that $Q'_1 \xRightarrow{\ell} Q'_2$ and

$$(\nu \bar{m}_1)(P'_2 \parallel t \leftarrow_c v : C_1) \mathcal{S} \quad (146)$$

$$(\nu \bar{m}_2)(Q'_2 \parallel t \leftarrow_m v \sigma_1 : C_1) \quad (147)$$

We should show that

$$(\nu \bar{m}_1)(P'_2 \mid P''_1 \parallel t \leftarrow_c v : C_1) \mathcal{S} \quad (148)$$

$$(\nu \bar{m}_2)(Q'_2 \mid P''_1 \parallel t \leftarrow_m v \sigma_1 : C_1) \quad (149)$$

By Def. C.10 and Tab. 4 from (146) we can infer the following:

$$P'_2 \{\tilde{u}'_i / \tilde{y}'\} \mathcal{S} Q'_2 \quad (150)$$

So, by (139) and (150) the goal (148) follows.

This concludes Par_\perp case.

- (4) Case Tau. Here we know $P_1 = P'_1 \mid P''_1$. Without the loss of generality, we assume $\ell_1 = (\nu \bar{m}_1) \bar{n}_i! \langle v_1 \rangle$ and $\ell_2 = n_i? \langle v_1 \rangle$. Let $\rho'_1 = \{\tilde{u}'_i / \tilde{y}\}$ and $\rho''_1 = \{\tilde{u}''_i / \tilde{z}\}$ such that

$$P_1 \rho_1 = P'_1 \rho'_1 \mid P''_1 \rho''_1$$

Then, the final rule in the inference tree is as follows:

$$\langle \text{Tau} \rangle \frac{P'_1 \rho'_1 \xrightarrow{\ell_1} P'_2 \rho'_2 \quad P''_1 \rho''_1 \xrightarrow{\ell_2} P''_2 \rho''_2 \quad \ell_1 \times \ell_2}{P'_1 \rho'_1 \mid P''_1 \rho''_1 \xrightarrow{\tau} (\nu \bar{m}_1)(P'_2 \rho'_2 \mid P''_2 \rho''_2)}$$

Let $\sigma_1, \tilde{y}, \tilde{z}, \tilde{u}_*, \tilde{x}_*$, and ρ_m be defined as in the previous case. Further, Q_1 can have shapes: Q_1^1, Q_1^2 , and N_1^3 as in the previous case. As in the previous case, we know that there are

$$Q'_1 \in C_{\tilde{y}}^{\tilde{y} \rho_m}(P'_1 \sigma_1) \quad (151)$$

$$Q''_1 \in C_{\tilde{z}}^{\tilde{z} \rho_m}(P''_1 \sigma_1) \quad (152)$$

such that

$$Q_1^1 \xrightarrow{\tau} Q_1^2 \xrightarrow{\tau} Q'_1 \mid Q''_1$$

Thus, in both cases we consider how $Q'_1 \mid Q''_1$ evolves. By the definition of \mathcal{S} we have

$$P'_1 \rho'_1 \mathcal{S} Q'_1 \quad (153)$$

$$P''_1 \rho''_1 \mathcal{S} Q''_1 \quad (154)$$

We apply IH component-wise:

- By applying IH to (153) we know there is Q'_2 such that $Q'_1 \xrightarrow{(\nu \bar{m}_2) \bar{n}_i! \langle \tilde{v} \rangle} Q'_2$ and

$$(\nu \bar{m}_1)(P'_2 \parallel t \leftarrow_c v : C_1) \rho'_2 \mathcal{S} \quad (155)$$

$$(\nu \bar{m}_2)(Q'_2 \parallel t \leftarrow_m v \sigma_1 : C_1) \quad (156)$$

where $v \sigma_1 \times_c \tilde{v}$.

- By applying IH to (154) we know there is Q''_2 such that $Q''_1 \xrightarrow{n_i? \langle \tilde{v} \rangle} Q''_2$ and

$$P''_2 \rho''_2 \mathcal{S} Q''_2 \quad (157)$$

Now, by (155) we can infer:

$$P'_2 \rho'_2 \mathcal{S} Q'_2$$

By this we know there is $\sigma'_2 \in \text{index}(\text{fn}(P'_2) \cup \rho'_2)$ such that

$$Q'_2 \in \widehat{C}_{\sigma'_2}^{\rho'_2}(P'_2)$$

Moreover, we know that it must be $\sigma'_2 \subseteq \sigma_1 \cdot \text{next}(\bar{n}_i)$. So, we have

$$Q'_2 \in \widehat{C}_{\sigma_1 \cdot \text{next}(\bar{n}_i)}^{\rho'_2}(P'_2)$$

Similarly, there is $\sigma''_2 \in \text{index}(\text{fn}(P''_2) \cup \rho''_2)$ and $\sigma''_2 \subseteq \sigma_1 \cdot \text{next}(\bar{n}_i)$, so we have the following

$$Q''_2 \in \widehat{C}_{\sigma_1 \cdot \text{next}(n_i)}^{\rho''_2}(P''_2)$$

So, we have

$$Q'_2 \mid Q''_2 \in \widehat{C}_{\sigma_1 \cdot \text{next}(n_i) \cdot \text{next}(\bar{n}_i)}^{\rho'_2 \cdot \rho''_2}(P'_2 \mid P''_2)$$

By definition of Def. C.9, there is $\sigma_2 \in \text{index}(\text{fn}(P'_2 \mid P''_2) \cup \rho'_2 \cup \rho''_2)$ such that $\sigma_2 \subseteq \sigma_1 \cdot \text{next}(n_i) \cdot \text{next}(\bar{n}_i)$. That is, we know

$$\widehat{C}_{\sigma_2}^{\rho'_2 \cdot \rho''_2}(P'_2 \mid P''_2) = \widehat{C}_{\sigma_1 \cdot \text{next}(n_i) \cdot \text{next}(\bar{n}_i)}^{\rho'_2 \cdot \rho''_2}(P'_2 \mid P''_2)$$

Thus, we can conclude

$$(v \bar{m}_1)(P'_2 \mid P''_2) \rho'_2 \cdot \rho''_2 \mathcal{S} (v \bar{m}_2)(Q'_2 \mid Q''_2)$$

This concludes Tau case.

Recursion cases Now, we consider cases where $P'_1 \equiv \mu X.P_1^*$ is a parallel component of P_1 . We focus on cases that highlight specifics of the breakdown of recursive processes and omit details that are similar to the corresponding non-recursive cases.

(1) Case Rv. Here we know $P_1 = n?(y).P_2$ such that $P_1 \equiv \mu X.P_1^*$. Let

$$P_1^1 = \alpha_d \cdot \alpha_{d-1} \cdot \dots \cdot \alpha_{d-p} \cdot (P_X \mid R)$$

where R and P_X are some processes, P_X contains X and $\alpha_d = n?(y)$. As P_1 is a recursive process, we know that there is $\mu X.P^*$ such that P_1^1 is its sub-processes and

$$P_1 \equiv P_1^1 \{ \mu X.P^* / X \}$$

Here we can distinguish two sub-cases: (i) $p > 0$ and (ii) $p = 0$ and $R \neq \mathbf{0}$. We know there is $\rho_2 = \{\tilde{u}_2 / \tilde{x}_2\}$ such that

$$P_1 \rho_1 = n \rho_1 ?(y).P_2 \rho_2$$

The transition is as follows:

$$\langle \text{Rv} \rangle \frac{}{P_1^1 \{ \mu X.P^* / X \} \xrightarrow{n?(v)} P_2 \rho_2 \cdot \{v/y\}}$$

Let $\sigma_1 = \sigma' \cdot \{n_i/n\}$ where $\sigma' \in \text{index}(\text{fn}(P_1) \cup \bar{u} \cup \bar{x})$ and $\sigma_2 = \sigma_1 \cdot \sigma$ where $\sigma = \text{next}(n_i) \cdot \{y_1/y\}$. Further, let $\bar{u}_* = \text{bn}(\bar{u}\sigma_1 : \bar{C})$ and $\bar{x}_* = \text{bn}(\bar{x}\sigma_1 : \bar{C})$ with $\bar{u} : \bar{C}$. Also, let $\bar{y} = (y_1, \dots, y_{|\mathcal{G}(S)|})$ and $\bar{z} = \text{fnb}(P_2, \bar{x}_* \bar{y} \setminus \bar{w})$ where $\bar{w} = \{n_i\}$ if $\text{lin}(n_i)$ otherwise $\bar{w} = \epsilon$.

We could see that $d = \text{recd}(\alpha_d, \alpha_{d-1}, \dots, \alpha_1, \mu X.P_1^*)$. So, by Tab. 5 there are following possibilities for the shape of Q_1 , namely elements in N defined as:

$$N = \{ (v \bar{c})R : R \in C_{\text{rec } \bar{x}_*, \rho}^{\bar{u}_*}(\mu X.P^*)^d \}$$

Let

$$Q_1^1 = (v \bar{c})B_1 \mid n_l ?(\bar{y}). \overline{c_{k+1}^r} !(\bar{z}\rho) \cdot \mu X.c_k^r ?(\bar{x}). n_l ?(\bar{y}). \overline{c_{k+1}^r} !(\bar{z}). X \mid \mathbb{A}_{\text{rec } \bar{z}}^{k+1}(\alpha_d, \alpha_{d-1}.P_1^2 \sigma)_g$$

where B_1 is, intuitively, trios mimicking prefixes before α_d and P_1^2 is such that $P_1^1 = \alpha_d \cdot \alpha_{d-1} \cdot P_1^2$. By unfolding definition of $C_{\text{rec } \bar{x}_*, \rho}^{\bar{u}_*}(\mu X.P^*)^d$ we have that $Q_1^1 \in N$. Further, if $R \in C_{\text{rec } \bar{x}_*, \rho}^{\bar{u}_*}(\mu X.P^*)^d$ then $R \xrightarrow{\tau} Q_1^1$. So, we only analyze how Q_1^1 evolves. We can infer the following:

$$Q_1^1 \xrightarrow{n_i \rho ?(\bar{v})} Q_2$$

where

$$Q_2 = B_1 \mid \overline{c_{k+1}^r} !(\bar{z}\rho \cdot \{\bar{v}/\bar{y}\}) \cdot \mu X.c_k^r ?(\bar{x}). n_l ?(\bar{y}). \overline{c_{k+1}^r} !(\bar{z}). X \mid \mathbb{A}_{\text{rec } \bar{z}}^{k+1}(\alpha_{d-1}.P_1^2 \sigma)_g$$

with $v\sigma_v \triangleright_c \tilde{v}$ for some $\sigma_v \in \text{index}(v)$. Now, we should show that

$$P_2\rho_2 \cdot \{v/y\} \mathcal{S} Q_2 \quad (158)$$

Let $\tilde{u}_z = \tilde{z}\rho_2 \cdot \{\tilde{v}/\tilde{y}\}$. In sub-case (i) we should show that $Q_2 \in C_{\tilde{z}}^{\tilde{u}_z}(\alpha_{d-1}.P_1^2)$. So, we should show that

$$Q_2 \in C_{\text{rec } \tilde{z}, \rho_2, \{\tilde{v}/\tilde{y}\}}^{\tilde{u}_z} (P^*)_g^{d-1}$$

This follows by the inspection of definition $\mathcal{J}_{\text{rec } \tilde{z}, \rho_2, \{\tilde{v}/\tilde{y}\}}^k (P^*)_g^{d-1}$. That is, we can notice that

$$Q_2^1 = B_1 \mid B_2$$

where $B_2 \in \mathcal{J}_{\text{rec } \tilde{z}, \rho_2, \{\tilde{v}/\tilde{y}\}}^k (\alpha_{d-1}.P_1^2\sigma)_g^{d-1}$ as $(d-1) + 1 = \text{recd}(\alpha_d.\alpha_{d-1}.P_1^2\sigma)$. So, (158) follows. This concludes sub-case (i).

In sub-case (ii) we know $P_2 \equiv \alpha_{d-1}.P_1^2 \mid R$. Now, by Tab. 5 we have $Q_2 \xrightarrow{\tau} Q_2^2$ where

$$Q_2^2 = (v\tilde{c})V_1 \mid c_{k+l+1}^r \langle \tilde{u}_{z_2} \rangle \mid V_2$$

where

$$\begin{aligned} V_1 &= B_1 \mid \mu X.c_k^r \langle \tilde{x} \rangle . n_l \langle \tilde{y} \rangle . \overline{c_{k+1}^r \langle \tilde{z} \rangle} . X \mid \overline{c_{k+1}^r \langle \tilde{u}_{z_1} \rangle} . \mu X.c_k^r \langle \tilde{x} \rangle . (\overline{c_{k+1}^r \langle \tilde{z}_1 \rangle} . X \mid c_{k+l+1}^r \langle \tilde{z}_2 \rangle) \\ V_2 &= \mathbb{A}_{\text{rec } \tilde{z}_1}^{k+1} (PX)_{g_1} \mid \mathbb{A}_{\text{rec } \tilde{z}_2}^{k+l+1} (R) \emptyset \end{aligned}$$

Now, we should show that $Q_2^2 \in C_{\tilde{z}}^{\tilde{u}_z}(\alpha_{d-1}.P_1^2 \mid R)$. By Tab. 5 we have that

$$\{R_1 \mid R_2 : R_1 \in C_{\text{rec } \tilde{z}_1}^{\tilde{u}_{z_1}}(\alpha_{d-1}.P_1^2), R_2 \in C_{\tilde{z}_2}^{\tilde{u}_{z_2}}(R)\} \subseteq C_{\tilde{z}}^{\tilde{u}_z}(\alpha_{d-1}.P_1^2 \mid R) \quad (159)$$

Further, we know that

$$(v\tilde{c}_{k+l+1})c_{k+l+1}^r \langle \tilde{u}_z \rangle \mid \mathbb{A}_{\text{rec } \tilde{z}_2}^{k+l+1} (R) \emptyset \in C_{\tilde{z}_2}^{\tilde{u}_{z_2}}(R) \quad (160)$$

and

$$V_1 \mid V_2 \in C_{\text{rec } \tilde{z}_1}^{\tilde{u}_{z_1}}(\alpha_{d-1}.P_1^2) \quad (161)$$

Thus, by (159), (160), and (161) we have the following:

$$V_1 \mid V_2 \mid (v\tilde{c}_{k+l+1})c_{k+l+1}^r \langle \tilde{u}_z \rangle \mid \mathbb{A}_{\text{rec } \tilde{z}_2}^{k+l+1} (R) \emptyset \in C_{\tilde{z}}^{\tilde{u}_z}(\alpha_{d-1}.P_1^2 \mid R)$$

Now, we can notice that $\tilde{c}_{k+l+1} \cap \text{fpn}(V_1) = \emptyset$ and $\tilde{c}_{k+l+1} \cap \text{fpn}(\mathbb{A}_{\text{rec } \tilde{z}_1}^{k+1} (PX)_{g_1}) = \emptyset$. Further, we have

$$(v\tilde{c}_{k+l+1})\mathbb{A}_{\text{rec } \tilde{z}_2}^{k+l+1} (R) \emptyset \approx^c \mathbf{0}$$

(where \approx^c is defined in Definition 18 [9]) as first shared trigger c_{k+l+1} in the breakdown of R is restricted so it could not get activated. Thus, we have

$$Q_2^2 \equiv V_1 \mid V_2 \mid (v\tilde{c}_{k+l+1})c_{k+l+1}^r \langle \tilde{u}_z \rangle \mid \mathbb{A}_{\text{rec } \tilde{z}_2}^{k+l+1} (R) \emptyset$$

This concludes Rv case. Now, we consider the inductive case.

(2) Case Par_\perp . Here we know $P_1 = P'_1 \mid P''_1$. Similarly to the previous case, let

$$P_1^1 = \alpha_d.\alpha_{d-1} \dots \alpha_{d-p} . (PX \mid R)$$

We assume P'_1 is a recursive process. So, we know there is $\mu X.P^*$ such that P_1^1 is its subprocess and

$$P'_1 \equiv P_1^1 \{ \mu X.P^* / X \}$$

Here, we can distinguish two sub-cases: (i) $P'_1 \equiv R$ and (ii) $P'_1 \not\equiv R$. Here, we consider sub-case (i) as it is an interesting case. The sub-case (ii) is similar to the corresponding case of non-recursive process. As in the previous case, we can further distinguish cases in which $p = 0$ and $p > 0$. We consider $p = 0$ and $\ell = n \langle v \rangle$.

The final rule in the inference tree is as follows:

$$\langle \text{Par}_\perp \rangle \frac{P'_1 \{ \tilde{u}'_1 / \tilde{y}_1 \} \xrightarrow{\ell} P'_2 \{ \tilde{u}'_2 / \tilde{y}_2 \} \mid R \{ \tilde{w}_1 / \tilde{z}_1 \} \cdot \{ \tilde{u}'_R / \tilde{w}_1 \} \quad \text{bn}(\ell) \cap \text{fn}(P''_1) = \emptyset}{P'_1 \{ \tilde{u}'_1 / \tilde{y}_1 \} \mid R \{ \tilde{u}''_1 / \tilde{z}_1 \} \xrightarrow{\ell} P'_2 \{ \tilde{u}'_2 / \tilde{y}_2 \} \mid R \{ \tilde{u}'_1 / \tilde{z}_1 \} \mid R \{ \tilde{w}_1 / \tilde{z}_1 \} \cdot \{ \tilde{u}'_R / \tilde{w}_1 \}}$$

Let $\sigma_1 \in \text{index}(\text{fn}(P_1) \cup \tilde{u} \cup \tilde{x})$. Further, let $\tilde{y}_{*1} = \text{fnb}(P'_1, \tilde{x}_*)$, $\tilde{z}_{*1} = \text{fnb}(R, \tilde{x}_*)$, $\tilde{u}_* = \text{bn}(\tilde{u}\sigma_1 : \tilde{C})$ where $\tilde{x}_* = \text{bn}(\tilde{x}\sigma_1 : \tilde{C})$ with $\tilde{u} : \tilde{C}$, and $\rho_m = \{\tilde{u}_*/\tilde{x}_*\}$. By the definition of \mathcal{S} (Tab. 5), there are following possibilities for Q_1 :

$$\begin{aligned} Q_1^1 &= (\nu \tilde{c}_k)(\overline{c_k}!\langle \tilde{u}_* \rangle \mid \mathcal{B}_{\tilde{x}_*}^k(P_1\sigma_1)) \\ Q_1^2 &= (\nu \tilde{c}_k)\overline{c_k}!\langle \tilde{u}_{*2}' \rangle.\overline{c_{k+l}}!\langle \tilde{u}_{*1}'' \rangle \mid \mathcal{B}_{\tilde{y}_{*1}}^k(P'_1\sigma_1) \mid \mathcal{B}_{\tilde{z}_{*1}}^{k+l}(R\sigma_1) \\ N_1^3 &= \{(R'_1 \mid R''_1) : C_{\text{rec } \tilde{y}_{*1}}^{\tilde{u}_{*2}'}(\mu X.P^*\sigma_1)_g^d, R''_1 \in C_{\tilde{z}_{*1}}^{\tilde{u}_{*1}''}(R\sigma_1)\} \end{aligned}$$

By Lem. C.12 there exist

$$Q'_1 \in C_{\text{rec } \tilde{y}_{*1}}^{\tilde{u}_{*2}'}(\mu X.P^*\sigma_1)_g^d \quad (162)$$

$$Q''_1 \in C_{\tilde{z}_{*1}}^{\tilde{u}_{*1}''}(R\sigma_1) \quad (163)$$

such that

$$Q_1^1 \xrightarrow{\tau} Q_1^2 \xrightarrow{\tau} Q'_1 \mid Q''_1$$

The interesting case is to consider a process Q_1^3 defined as:

$$Q_1^3 = (\nu \tilde{c})V_1 \mid c_{k+l+1}^r!\langle \tilde{u}_{*1}'' \rangle \mid V_2$$

where

$$\begin{aligned} V_1 &= B_1 \mid \overline{c_{k+1}^r}!\langle \tilde{u}_{*2}' \rangle.\mu X.c_k^r?(\tilde{y}_{*1}).(\overline{c_{k+1}^r}!\langle \tilde{y}_{*2} \rangle.X \mid c_{k+l+1}^r!\langle \tilde{z}_{*2} \rangle) \\ V_2 &= \mathbb{A}_{\text{rec } \tilde{y}_{*2}}^{k+1}(P_X)_{g_1} \mid \mathbb{A}_{\text{rec } \tilde{w}_{*2}}^{k+l+1}(R\{\tilde{w}_2/\tilde{z}_2\})_{\emptyset} \end{aligned}$$

We have $Q_1^3 \approx^c Q'_1 \mid Q''_1$ (\approx^c is defined in Definition 18 [9]). We may notice that Q_1^3 can be a descendent of the recursive process (following the similar reasoning as in the previous case). So, we consider how Q_1^3 evolves. As in the corresponding case of non-recursive processes, we do the case analysis on ℓ . If $v \in \tilde{u}$ then we take $\sigma_v = \sigma_1$, otherwise $\sigma_v = \{v_j/v\}$ for $j > 0$. Now, we could see that

$$Q_1^3 \xrightarrow{n?(\tilde{v})} Q_2$$

where

$$Q_2 = (\nu \tilde{c})V_1 \mid \overline{c_{k+1}^r}!\langle \tilde{u}_{*2}' \rangle.\mu X.c_k^r?(\tilde{y}_{*1}).(\overline{c_{k+1}^r}!\langle \tilde{y}_{*2} \rangle.X \mid c_{k+l+1}^r!\langle \tilde{z}_{*1} \rangle) \mid c_{k+l+1}^r!\langle \tilde{u}_{*R}' \rangle \mid V_2$$

We define Q_2^1 as follows

$$\begin{aligned} Q_2^1 &= (\nu \tilde{c})V_1 \mid \overline{c_{k+1}^r}!\langle \tilde{u}_{*2}' \rangle.\mu X.c_k^r?(\tilde{y}_{*1}).(\overline{c_{k+1}^r}!\langle \tilde{y}_{*2} \rangle.X \mid c_{k+l+1}^r!\langle \tilde{z}_{*1} \rangle) \mid V_2 \mid \\ &\quad (\nu \tilde{c}_k)(c_k^r!\langle \tilde{u}_{*1}'' \rangle \mid \mathbb{A}_{\text{rec } \tilde{z}_{*1}}^k(R)_{\emptyset}) \mid (\nu \tilde{c}_k)(c_k^r!\langle \tilde{u}_{*R}' \rangle \mid \mathbb{A}_{\text{rec } \tilde{w}_{*1}}^k(R\{\tilde{w}_1/\tilde{z}_1\})_{\emptyset}) \end{aligned}$$

By definition, we could see that $Q_2^1 \in C_{\tilde{z}_{*1}}^{\tilde{u}_{*2}'}(P'_2 \mid R \mid R\{\tilde{w}_1/\tilde{z}_1\})$. Now, by the definition of $\mathbb{A}_{\text{rec } -}^k(\cdot)_{\emptyset}$ we have

$$\begin{aligned} c_{k+l+1}^r!\langle \tilde{u}_{*1}'' \rangle \mid c_{k+l+1}^r!\langle \tilde{u}_{*R}' \rangle \mid \mathbb{A}_{\text{rec } \tilde{z}_{*1}}^{k+l+1}(R)_{\emptyset} &\approx^c \\ (\nu \tilde{c}_k)(c_k^r!\langle \tilde{u}_{*1}'' \rangle \mid \mathbb{A}_{\text{rec } \tilde{z}_{*1}}^k(R)_{\emptyset}) \mid (\nu \tilde{c}_k)(c_k^r!\langle \tilde{u}_{*R}' \rangle \mid \mathbb{A}_{\text{rec } \tilde{w}_{*1}}^k(R\{\tilde{w}_1/\tilde{z}_1\})_{\emptyset}) & \end{aligned}$$

As each trio in $\mathbb{A}_{\text{rec } \tilde{z}_{*1}}^{k+l+1}(R)_{\emptyset}$ makes a replica of itself when triggered along a propagator. So, finally we have

$$Q_2 \approx^c Q_2^1$$

This concludes the proof. \square

Lemma C.8. Assume $P_1\{\tilde{u}/\tilde{x}\}$ is a well-formed process and $P_1\{\tilde{u}/\tilde{x}\} \mathcal{S} Q_1$.

- (1) Whenever $Q_1 \xrightarrow{(\nu \tilde{m}_2)\tilde{n}!(\tilde{v};\mathcal{G}(C_1))} Q_2$ then there exist P_2 and σ_v such that $P_1\{\tilde{u}/\tilde{x}\} \xrightarrow{(\nu \tilde{m}_1)\tilde{n}!(\tilde{v};C_1)} P_2$ where $v\sigma_v \vDash \tilde{v}$ and, for a fresh t ,

$$(\nu \tilde{m}_1)(P_2 \parallel t \Leftarrow_C v : C_1) \mathcal{S} (\nu \tilde{m}_2)(Q_2 \parallel t \Leftarrow_m v\sigma_v : C_1)$$
- (2) Whenever $Q_1 \xrightarrow{\tilde{n}?(v)} Q_2$ then there exist P_2 and σ_v such that $P_1\{\tilde{u}/\tilde{x}\} \xrightarrow{\tilde{n}?(v)} P_2$ where $v\sigma_v \vDash \tilde{v}$ and $P_2 \mathcal{S} Q_2$.
- (3) Whenever $Q_1 \xrightarrow{\tau} Q_2$ either (i) $P_1\{\tilde{u}/\tilde{x}\} \mathcal{S} Q_2$ or (ii) there exists P_2 such that $P_1 \xrightarrow{\tau} P_2$ and $P_2 \mathcal{S} Q_2$.

SKETCH. Following Parrow, we refer to prefixes corresponding to prefixes of the original process as *essential prefixes*. We remark that a prefix in $\mathcal{D}(P)$ is non-essential if and only if is a prefix on a propagator name. First, we discuss case when transition is inferred without any actions from essential prefixes. In this case we know that an action can only involve propagator prefixes and by inspection of definition of $C_{\tilde{x}}^{\tilde{u}}(P_1)$ that $\ell = \tau$. This concerns the sub-case (i) of Part 3 and it directly follows by Lem. C.12.

Now, assume $Q_1 \xrightarrow{\ell} Q_2$ when ℓ involves essential prefixes. This concerns Part 1, Part 2, and sub-case (ii) of Part 3. This case is mainly the inverse of the proof of Lem. C.7. As Parrow, here we note that an essential prefix is unguarded in Q_1 if and only if it is unguarded in P_1 . That is, by inspection of the definition, function $C_{\tilde{x}}^{\tilde{u}}(P_1)$ does not unguard essential prefixes of P_1 that its members mimic (the propagators serve as guards). □

THEOREM 5 (OPERATIONAL CORRESPONDENCE). *Let P be a π process such that $\Gamma_1; \Delta_1 \vdash P_1$. We have*

$$\Gamma; \Delta \vdash P \approx^M \mathcal{H}^*(\Gamma); \mathcal{H}^*(\Delta) \vdash \mathcal{F}^*(P)$$

PROOF. By Lem. C.7 and Lem. C.8 we know \mathcal{S} is a MST bisimulation. So, we need to show $(P, \mathcal{F}^*(P)) \in \mathcal{S}$. Let P_1 be such that $P_1\{\tilde{r}/\tilde{x}\} = P$ where $\tilde{r} = \text{rn}(P)$. Further, let $\sigma = \bigcup_{v \in \tilde{r}} \{v_1/v\}$, so we have $\sigma \in \text{index}(\tilde{r})$. Then, let $\tilde{r}_* = \text{bn}(\tilde{r} : \tilde{S})$ and $\tilde{x}_* = \text{bn}(\tilde{x} : \tilde{S})$ where $\tilde{r} : \tilde{S}$. Therefore, by Def. 4.8 and Tab. 4 we have $\mathcal{F}^*(P) \in C_{\tilde{x}_*}^{\tilde{r}_*}(P_1)$. Finally, by Def. C.10 we have $(P, \mathcal{F}^*(P)) \in \mathcal{S}$. □