Minimal Session Types for the π-calculus

Alen Arslanagić
University of Groningen
The Netherlands

Anda-Amelia Palamariuc
University of Groningen
The Netherlands

Jorge A. Pérez
University of Groningen
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 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 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. This new minimality result for the session π-calculus can be obtained by composing existing results. We develop associated optimizations and enhancements of this result, and establish its static and dynamic correctness.

KEYWORDS

Concurrency, Session Types, Process Calculi

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 $S =?(\mathsf{Int}); ?(\mathsf{Int}); !\langle\mathsf{Bool}\rangle; \mathsf{end}$, 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 $[\![\cdot]\!]^1_g$) and from HO to π (denoted $[\![\cdot]\!]^2$). Therefore, given $\mathcal{D}(\cdot)$, $[\![\cdot]\!]^1_g$,
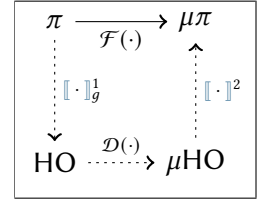

Figure 1: Decomposition by composition.

and $[\![\cdot]\!]^2$, to define a decomposition function $\mathcal{F}(\cdot) : \pi \to \mu\pi$, it suffices to follow Figure 1. This is sound for our purposes, because $[\![\cdot]\!]^1_g$ and $[\![\cdot]\!]^2$ 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 $\mathcal{D}(\cdot)$, $[\![\cdot]\!]^1_g$, and $[\![\cdot]\!]^2$ 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.

Summing up, 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 $\mathcal{F}^*(\cdot)$.

Note to Reviewers: Due to space limits, omitted material (definitions, proofs, additional examples) can be found in the appendices.

2 PRELIMINARIES

HO and π are actually sub-languages of HOπ [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].

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

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

$$
\begin{array}{ll}
(\lambda x.\, P)\, u \longrightarrow P\{u/x\} & [\text{App}] \\[4pt]
n!\langle V \rangle.P \mid \bar{n}?(x).Q \longrightarrow P \mid Q\{V/x\} & [\text{Pass}] \\[4pt]
P \longrightarrow P' \Rightarrow (\nu n)P \longrightarrow (\nu n)P' & [\text{Res}] \\[4pt]
P \longrightarrow P' \Rightarrow P \mid Q \longrightarrow P' \mid Q & [\text{Par}] \\[4pt]
P \equiv Q \longrightarrow Q' \equiv P' \Rightarrow P \longrightarrow P' & [\text{Cong}] \\[4pt]
\end{array}
$$

$$
\begin{array}{ll}
P_1 \mid P_2 \equiv P_2 \mid P_1 & P_1 \mid (P_2 \mid P_3) \equiv (P_1 \mid P_2) \mid P_3 \\[4pt]
P \mid \mathbf{0} \equiv P & P \mid (\nu n)Q \equiv (\nu n)(P \mid Q)\ (n \notin \mathsf{fn}(P)) \\[4pt]
(\nu n)\mathbf{0} \equiv \mathbf{0} & \mu X.P \equiv P\{\mu X.P/X\} \qquad P \equiv Q \text{ if } P \equiv_\alpha Q
\end{array}
$$

Figure 3: Operational Semantics of HOπ.

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

Fig. 2 gives the **syntax** of processes $P, Q, \ldots$, values $V, W, \ldots$, and conventions for names. Identifiers $a, b, c, \ldots$ denote *shared names*, while $s, \bar{s}, \ldots$ are used for *session names*. Duality is defined only on session names, thus $\bar{\bar{s}} = s$, but $\bar{a} = a$. *Names* (shared or sessions) are denoted by $m, n \ldots$, and $x, y, z, \ldots$ range over variables. We write $\widetilde{x}$ to denote a tuple $(x_1, \ldots, x_n)$, 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 $(\nu n)P$ 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 $\mathsf{fv}(P)$, $\mathsf{fs}(P)$, and $\mathsf{fn}(P)$. A process P is *closed* if $\mathsf{fv}(P) = \emptyset$. 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 HOπ differ as follows: application $V u$ is only present in HO; constructs for recursion $\mu X.P$ are present in π but not in HO.

The **operational semantics** of HOπ, 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

$$
\begin{aligned}
U &::= C \mid L \\
C &::= S \mid \langle S \rangle \mid \langle L \rangle \\
L &::= U \to \diamond \mid U \multimap \diamond \\
S &::= !\langle U \rangle; S \mid ?(U); S \mid \mu t.S \mid t \mid \mathsf{end} \\
\end{aligned}
$$

$\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots$

$$
\begin{aligned}
U &::= \widetilde{C} \to \diamond \mid \widetilde{C} \multimap \diamond \\
C &::= M \mid \langle U \rangle \\
M &::= \gamma \mid !\langle \widetilde{U} \rangle; \gamma \mid ?(\widetilde{U}); \gamma \mid \mu t.M \\
\gamma &::= \mathsf{end} \mid t
\end{aligned}
$$

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

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 HOπ

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 denote the *process type*. The functional types $U \to \diamond$ and $U \multimap \diamond$ 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:

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

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 $\mathsf{dom}(\Gamma)$, $\mathsf{dom}(\Lambda)$, and $\mathsf{dom}(\Delta)$) are assumed pairwise distinct.

Given Γ, we write $\Gamma \backslash x$ to denote the environment obtained from Γ by removing the assignment $x : U \to \diamond$, for some U. This notation applies similarly to Δ and Λ; we write $\Delta \backslash \Delta'$ (and $\Lambda \backslash \Lambda'$) 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:

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

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 HOπ 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:

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

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* $\Delta = \Delta'$ *or* $\Delta \longrightarrow \Delta'$ *with* Δ' *balanced.*

2.3 Mutual Encodings between π and HO

The encodings $[\![\cdot]\!]^1_g : \pi \to$ HO and $[\![\cdot]\!]^2 :$ HO $\to \pi$ are *typed*: each consists of a translation on processes and a translation on types. This way, $\langle\!\langle \cdot \rangle\!\rangle^1$ translates session types for first-order processes into session types for higher-order processes, while $\langle\!\langle \cdot \rangle\!\rangle^2$ operates in the opposite direction. Their definitions are recalled in 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 $[\![\cdot]\!]^1_g$ 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, $\|\cdot\|$ maps sequences of session names into sequences of variables, and $\big\lfloor \cdot \big\rfloor_\emptyset$ maps processes with free names to processes without free names (but with free variables instead).

The encoding on types $\langle\!\langle \cdot \rangle\!\rangle^1$ depends on an auxiliary function, $\lfloor \cdot \rfloor^1$, 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 $[\![\cdot]\!]^2$ 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 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 name. The encoding of application is also in two cases; both of

Terms:

$$[\![u!\langle w\rangle.P]\!]^1_g \overset{\text{def}}{=} u!\langle \lambda z.\ z?(x).(x\ w)\rangle.[\![P]\!]^1_g$$

$$[\![u?(x{:}C).Q]\!]^1_g \overset{\text{def}}{=} u?(y).(\nu s)(y\ s\ |\ \bar{s}!\langle \lambda x.\ [\![Q]\!]^1_g\rangle.\mathbf{0})$$

$$[\![P\ |\ Q]\!]^1_g \overset{\text{def}}{=} [\![P]\!]^1_g\ |\ [\![Q]\!]^1_g$$

$$[\![(\nu n)P]\!]^1_g \overset{\text{def}}{=} (\nu n)[\![P]\!]^1_g$$

$$[\![\mathbf{0}]\!]^1_g \overset{\text{def}}{=} \mathbf{0}$$

$$[\![\mu X.P]\!]^1_g \overset{\text{def}}{=} (\nu s)(\bar{s}!\langle V\rangle.\mathbf{0}\ |\ s?(z_X).[\![P]\!]^1_{g,\{X\to\tilde{n}\}})$$
$$\text{where } (\tilde{n} = \mathsf{fn}(P))$$
$$V = \lambda(\|\tilde{n}\|, y).\ y?(z_X).\big\lfloor [\![P]\!]^1_{g,\{X\to\tilde{n}\}} \big\rfloor_\emptyset$$

$$[\![X]\!]^1_g \overset{\text{def}}{=} (\nu s)(z_X\ (\tilde{n}, s)\ |\ \bar{s}!\langle z_X\rangle.\mathbf{0})\quad (\tilde{n} = g(X))$$

Types:

$$\lfloor S\rfloor^1 \overset{\text{def}}{=} (?(\langle\!\langle S\rangle\!\rangle^1 \multimap \diamond); \mathsf{end}) \multimap \diamond$$

$$\lfloor\langle S\rangle\rfloor^1 \overset{\text{def}}{=} (?(\langle\langle\!\langle S\rangle\!\rangle^1\rangle \to \diamond); \mathsf{end}) \multimap \diamond$$

$$\langle\!\langle !\langle U\rangle; S\rangle\!\rangle^1 \overset{\text{def}}{=} !\langle\lfloor U\rfloor^1\rangle; \langle\!\langle S\rangle\!\rangle^1$$

$$\langle\!\langle ?(U); S\rangle\!\rangle^1 \overset{\text{def}}{=} ?(\lfloor U\rfloor^1); \langle\!\langle S\rangle\!\rangle^1$$

$$\langle\!\langle\langle S\rangle\rangle\!\rangle^1 \overset{\text{def}}{=} \langle\langle\!\langle S\rangle\!\rangle^1\rangle$$

$$\langle\!\langle \mathsf{end}\rangle\!\rangle^1 \overset{\text{def}}{=} \mathsf{end}$$

$$\langle\!\langle \mu t.S\rangle\!\rangle^1 \overset{\text{def}}{=} \mu t.\langle\!\langle S\rangle\!\rangle^1$$

$$\langle\!\langle t\rangle\!\rangle^1 \overset{\text{def}}{=} t$$

Figure 5: Typed encoding of π into HO, selected cases from [10]. Above, $\mathsf{fn}(P)$ is a lexicographically ordered sequence of free names in P. Maps $\|\cdot\|$ and $\big\lfloor \cdot \big\rfloor_\sigma$ are in Def. B.2 and Fig. 16.

Terms:

$$[\![u!\langle\lambda x.\ Q\rangle.P]\!]^2 \overset{\text{def}}{=}$$

$$\begin{cases} (\nu a)(u!\langle a\rangle.([\![P]\!]^2\ |\ *\ a?(y).y?(x).[\![Q]\!]^2)) & \text{if } \mathsf{fs}(Q) = \emptyset \\ (\nu a)(u!\langle a\rangle.([\![P]\!]^2\ |\ a?(y).y?(x).[\![Q]\!]^2)) & \text{otherwise} \end{cases}$$

$$[\![u?(x).P]\!]^2 \overset{\text{def}}{=} u?(x).[\![P]\!]^2$$

$$[\![x\ u]\!]^2 \overset{\text{def}}{=} (\nu s)(x!\langle s\rangle.\bar{s}!\langle u\rangle.\mathbf{0})$$

$$[\![(\lambda x.\ P)\ u]\!]^2 \overset{\text{def}}{=} (\nu s)(s?(x).[\![P]\!]^2\ |\ \bar{s}!\langle u\rangle.\mathbf{0})$$

Types:

$$\langle\!\langle !\langle S \multimap \diamond\rangle; S_1\rangle\!\rangle^2 \overset{\text{def}}{=} !\langle\langle ?(\langle\!\langle S\rangle\!\rangle^2); \mathsf{end}\rangle\rangle; \langle\!\langle S_1\rangle\!\rangle^2$$

$$\langle\!\langle ?(S \multimap \diamond); S_1\rangle\!\rangle^2 \overset{\text{def}}{=} ?(\langle ?(\langle\!\langle S\rangle\!\rangle^2); \mathsf{end}\rangle); \langle\!\langle S_1\rangle\!\rangle^2$$

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

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 $\widetilde{C} \rightarrow \diamond$ and $\widetilde{C} \multimap \diamond$. Similarly, minimal session types for output and input are of the form $!\langle \widetilde{U} \rangle; \mathsf{end}$ and $?(\widetilde{U}); \mathsf{end}$: they communicate tuples of values but lack a continuation.

Following Parrow [11], $\mathcal{D}(\cdot)$ is defined in terms of a *breakdown function* $\mathcal{B}_{\widetilde{x}}^k(\cdot)$, 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 $\widetilde{x}$, possibly empty, which makes variable bindings explicit. We use a reserved set of *propagator names* (or simply *propagators*), denoted by $c_k, c_{k+1}, \ldots$, to carry contexts and trigger the subsequent trio. Propagators with recursive types are denoted by $c_k^r, c_{k+1}^r, \ldots$. 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 $\mathcal{B}_{\widetilde{x}}^k(P)$, where k is the index for the propagators c_k, and $\widetilde{x}$ is the context to be received by the previous trio. Similarly, the breakdown of a value V is denoted by $\mathcal{V}_{\widetilde{x}}^k(V)$. Figure 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 $\mathcal{B}_{\widetilde{x}}^k(\cdot)$, we need some notation. Let $\widetilde{u} = (a, b, s, s', \ldots)$ be a finite tuple of names. We shall write $\mathsf{init}(\widetilde{u})$ to denote the tuple $(a_1, b_1, s_1, s'_1, \ldots)$. 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 $\widetilde{u} = \mathsf{fn}(P)$. The *decomposition* of P, denoted $\mathcal{D}(P)$, is defined as:

$$\mathcal{D}(P) = (\nu \widetilde{c})\big(\overline{c_k}!\langle\rangle.\mathbf{0} \mid \mathcal{B}_{\epsilon}^k(P\sigma)\big)$$

where: $k > 0$; $\widetilde{c} = (c_k, \ldots, c_{k+|P|-1})$; $\sigma = \{\mathsf{init}(\widetilde{u})/\widetilde{u}\}$; and the breakdown function $\mathcal{B}_{\widetilde{x}}^k(\cdot)$, 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 $\mathcal{B}_{\widetilde{x}}^k(P)$.

$$\mathcal{G}(!\langle U \rangle; S) = \begin{cases} !\langle \mathcal{G}(U) \rangle; \mathsf{end} & \text{if } S = \mathsf{end} \\ !\langle \mathcal{G}(U) \rangle; \mathsf{end}, \mathcal{G}(S) & \text{otherwise} \end{cases}$$

$$\mathcal{G}(?(U); S) = \begin{cases} ?(\mathcal{G}(U)); \mathsf{end} & \text{if } S = \mathsf{end} \\ ?(\mathcal{G}(U)); \mathsf{end}, \mathcal{G}(S) & \text{otherwise} \end{cases}$$

$$\mathcal{G}(\mathsf{end}) = \mathsf{end}$$

$$\mathcal{G}(C \multimap \diamond) = \mathcal{G}(C) \multimap \diamond$$

$$\mathcal{G}(C \rightarrow \diamond) = \mathcal{G}(C) \rightarrow \diamond$$

$$\mathcal{G}(\langle U \rangle) = \langle \mathcal{G}(U) \rangle$$

$$\mathcal{G}(S_1, \ldots, S_n) = \mathcal{G}(S_1), \ldots, \mathcal{G}(S_n)$$

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

THEOREM 2.5 (MINIMALITY RESULT [1]). *Let P be a closed* HO *process with $\widetilde{u} = \mathsf{fn}(P)$ and $\sigma = \{\mathsf{init}(\widetilde{u})/\widetilde{u}\}$. 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 $\mathcal{A}_{\widetilde{x}}^k(\cdot)_g$ (cf. Tab. 2). Following Figure 1, this breakdown function will result from the composition of $[\![\cdot]\!]_g^1$, $\mathcal{B}_{\widetilde{x}}^k(\cdot)$, and $[\![\cdot]\!]^2$, i.e., $\mathcal{A}_{\widetilde{x}}^k(\cdot)_g = [\![\mathcal{B}_{\widetilde{x}}^k([\![\cdot]\!]_g^1)]\!]^2$. 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 $\mathcal{B}_{\widetilde{x}}^k([\![\cdot]\!]_g^1)$, which returns a process in μHO; second, a step that transforms that μHO process into a $\mu\pi$ process using $[\![\cdot]\!]^2$. We illustrate these two steps for output and input processes.

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

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

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

$$W = \lambda z_1. (\overline{c_{k+1}}!\langle\rangle \mid c_{k+1}?().z_1?(x).\overline{c_{k+2}}!\langle x \rangle \mid c_{k+2}?(x).(x\,\widetilde{w}))$$

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

$$c_k?(\widetilde{x}).(\nu\,a)\big(u_i!\langle a \rangle.(\overline{c_{k+3}}!\langle \widetilde{x} \rangle \mid \mathcal{A}_{\widetilde{x}}^{k+3}(Q\sigma)_g \mid$$
$$a?(y).y?(z_1).\overline{c_{k+1}}!\langle z_1 \rangle \mid c_{k+1}?(z_1).z_1?(x).\overline{c_{k+2}}!\langle x \rangle \mid$$
$$c_{k+2}?(x).(\nu\,s)(x!\langle s \rangle.\overline{s}!\langle \widetilde{w} \rangle)))$$

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 $\widetilde{x}$ is sent to breakdown of continuation Q.

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

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

We refer to names typed with tail-recursive type as *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.

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

$$c_k?(\tilde{x}).u_i?(y).\overline{c_{k+1}}!\langle\tilde{x}, y\rangle \mid$$
$$(\nu s_1)\big(c_{k+1}?(\tilde{x}, y).\overline{c_{k+2}}!\langle y\rangle.\overline{c_{k+3}}!\langle\tilde{x}\rangle \mid$$
$$c_{k+2}?(y).(\nu s)(y!\langle s\rangle.\overline{s}!\langle s_1\rangle) \mid$$
$$c_{k+3}?(\tilde{x}).(\nu a)(\overline{s_1}!\langle a\rangle.(\overline{c_{k+l+4}}!\langle\rangle \mid c_{k+l+4}?().\mathbf{0} \mid \widehat{Q}_{\tilde{x}})))$$

where $\widehat{Q}_{\tilde{x}} = a?(y').y'?(\tilde{w}).\big(\overline{c_{k+4}}!\langle\tilde{x}\rangle \mid \mathcal{A}_{\tilde{x}}^{k+4}\big(Q\{w_1/w\}\sigma\big)_g\big)$.

After activation on c_k, the input on u_i is enabled. As in the output case, the actual input is redirected to a different channel. After several redirections (with actions matching the breakdown of output), the actual input of variables $\tilde{w}$ is on a name received for y', which binds them in $\widehat{Q}_{\tilde{x}}$. 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: special 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.

$$\boxed{\begin{aligned} C &::= M \mid \langle M\rangle \\ \gamma &::= \text{end} \mid t \\ M &::= \gamma \mid !\langle\widetilde{C}\rangle;\gamma \mid ?(\widetilde{C});\gamma \mid \mu t.M \end{aligned}}$$

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

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

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

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 $\text{next}(n_i) = (\text{lin}(n_i))\,?\,\{n_{i+1}/n_i\}\!:\!\{\}$.

The following definition explains 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 $(\!(\,\cdot\,)\!)^1$, $\mathcal{G}(\cdot)$, and $(\!(\,\cdot\,)\!)^2$, 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 $\mathcal{R}'^{\star}(\cdot)$, 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 $\mathcal{R}'(\cdot)$.

P	$\mathcal{A}_{\tilde{x}}^k(P)_g$	
$u_i!\langle w_j\rangle.Q$	$c_k?(\widetilde{x}).(\nu a)(u_i!\langle a\rangle.(\overline{c_{k+3}}!\langle\widetilde{x}\rangle \mid \mathcal{A}_{\tilde{x}}^{k+3}(Q\sigma)_g \mid$ $\qquad a?(y).y?(z_1).\overline{c_{k+1}}!\langle z_1\rangle \mid c_{k+1}?(z_1).z_1?(x).\overline{c_{k+2}}!\langle x\rangle \mid$ $\qquad\qquad c_{k+2}?(x).(\nu s)(x!\langle s\rangle.\overline{s}!\langle\widetilde{w}\rangle)))$	$w_j : C$ $\widetilde{w} = (w_j,\dots,w_{j+|\mathcal{H}(C)|-1})$ $\sigma = \mathsf{next}(u_i)$
$u_i?(w).Q$	$c_k?(\widetilde{x}).u_i?(y).\overline{c_{k+1}}!\langle\widetilde{x},y\rangle \mid$ $(\nu s_1)(c_{k+1}?(\widetilde{x},y).\overline{c_{k+2}}!\langle y\rangle.\overline{c_{k+3}}!\langle\widetilde{x}\rangle \mid$ $\qquad c_{k+2}?(y).(\nu s)(y!\langle s\rangle.\overline{s}!\langle s_1\rangle) \mid$ $\qquad c_{k+3}?(\widetilde{x}).(\nu a)(\overline{s_1}!\langle a\rangle.(\overline{c_{k+l+4}}!\langle\rangle \mid c_{k+l+4}?().\mathbf{0} \mid \widehat{Q}_{\tilde{x}})))$ where: $\widehat{Q}_{\tilde{x}} = a?(y').y'?(\widetilde{w}).(\overline{c_{k+4}}!\langle\widetilde{x}\rangle \mid \mathcal{A}_{\tilde{x}}^{k+4}(Q\{w_l/w\}\sigma)_g)$	$w : C$ $\widetilde{w} = (w_1,\dots,w_{|\mathcal{H}(C)|})$ $l = \lfloor Q\rfloor$ $\sigma = (u_i : S)\,?\,\{u_{i+1}/u_i\}\!:\!\epsilon\;\;\sigma = \mathsf{next}(u_i)$
$r_i!\langle w_j\rangle.P$	$c_k?(\widetilde{x}).(\nu a_1)c^r!\langle a_1\rangle.(\mathcal{A}_{\tilde{x}}^{k+3}(P)_g \mid a_1?(y_1).y_1?(\widetilde{z}).W)$ where: $W = (\nu a_2)(z_{f(S)}!\langle a_2\rangle.(\overline{c_{k+3}}!\langle\widetilde{x}\rangle.c^r?(b).(\nu s)(b!\langle s\rangle.\overline{s}!\langle\widetilde{z}\rangle) \mid$ $\qquad a_2?(y_2).y_2?(z_1').(\nu\widetilde{c})(\overline{c_{k+1}}!\langle\rangle \mid$ $\qquad c_{k+1}?().z_1'?(x).\overline{c_{k+2}}!\langle x\rangle \mid c_{k+2}?(x).(\nu s')(x!\langle s'\rangle.\overline{s'}!\langle\widetilde{w}\rangle))))$	$r : S \wedge \mathsf{tr}(S)$ $\widetilde{z} = (z_1,\dots,z_{|\mathcal{R}'^\star(S)|})$ $\widetilde{c} = (c_{k+1},c_{k+2})$ $w : C \wedge \widetilde{w} = (w_j,\dots,w_{j+|\mathcal{H}(C)|-1})$
$r_i?(w).P$	$c_k?(\widetilde{x}).(\nu a_1)(c^r!\langle a_1\rangle.((\nu s_1)(c_{k+1}?(y).\overline{c_{k+2}}!\langle y\rangle.\overline{c_{k+3}}!\langle\rangle \mid$ $\qquad c_{k+2}?(y).(\nu s)(y!\langle s\rangle.\overline{s}!\langle s_1\rangle) \mid$ $\qquad c_{k+3}?().(\nu a_2)(s_1!\langle a_2\rangle.(\overline{c_{k+l+4}}!\langle\rangle \mid c_{k+l+4}?().\mathbf{0} \mid$ $\qquad a_2?(y_2).y_2?(\widetilde{w}).(\nu\widetilde{c})(\overline{c_{k+4}}!\langle\widetilde{x}\rangle \mid \mathcal{A}_{\tilde{x}}^{k+4}(P\{w_l/w\})_g))) \mid$ $\qquad a_1?(y_1).y_1?(\widetilde{z}).z_{f(S)}?(y).$ $\qquad\qquad \overline{c_{k+1}}!\langle y\rangle.c^r?(b).(\nu s')(b!\langle s'\rangle.\overline{s'}!\langle\widetilde{z}\rangle))))$	$r : S \wedge \mathsf{tr}(S)$ $\widetilde{z} = (z_1,\dots,z_{|\mathcal{R}'^\star(S)|})$ $l = \lfloor P\rfloor$ $w : C \wedge \widetilde{w} = (w_1,\dots,w_{|\mathcal{H}(C)|})$
$(\nu s : C)P'$	$(\nu\widetilde{s} : \mathcal{H}(C))\,\mathcal{A}_{\tilde{x}}^k(P'\sigma)_g$	$\widetilde{s} = (s_1,\dots,s_{|\mathcal{H}(C)|})\quad \sigma = \{s_1\overline{s_1}/s\overline{s}\}$
$Q_1 \mid Q_2$	$c_k?(\tilde{x}).\overline{c_{k+1}}!\langle\tilde{y}\rangle.\overline{c_{k+l+1}}!\langle\tilde{z}\rangle \mid \mathcal{A}_{\tilde{y}}^{k+1}(Q_1)_g \mid \mathcal{A}_{\tilde{z}}^{k+l+1}(Q_2)_g$	$\widetilde{y} = \mathsf{fv}(Q_1)\quad \widetilde{z} = \mathsf{fv}(Q_2)\quad l = \lfloor Q\rfloor$
$\mathbf{0}$	$c_k?().\mathbf{0}$	
$\mu X.P$	$(\nu s_1)(c_k?(\widetilde{x}).\overline{c_{k+1}}!\langle\widetilde{x}\rangle.\overline{c_{k+3}}!\langle\widetilde{x}\rangle \mid$ $\qquad c_{k+1}?(\widetilde{x}).(\nu a_1)(s_1!\langle a_1\rangle.(\overline{c_{k+2}}!\langle\rangle \mid c_{k+2}?().\mathbf{0} \mid$ $\qquad c_{k+3}?(\widetilde{x}).s_1?(z_x).\overline{c_{k+4}}!\langle\widetilde{x},z_x\rangle \mid$ $\qquad \mathcal{A}_{\tilde{x},z_x}^{k+4}(P)_{g,\{X\to\tilde{n}\}} \mid$ $\qquad * a_1?(y_1').y_1'?(\|\widetilde{n}^1\|,\dots,\|\widetilde{n}^m\|,y_1).\widehat{P})))$ where: $\widehat{P} = (\nu\widetilde{c})\prod_{0<i\le m}(c^{n_i}?(b).(\nu s')(b!\langle s'\rangle.\overline{s'}!\langle\|\widetilde{n}^i\|\rangle) \mid \overline{c_{k+2}}!\langle\widetilde{x}\rangle$ $\qquad c_{k+2}?(\widetilde{x}).y_1?(z_x).\overline{c_{k+3}}!\langle\widetilde{x},z_x\rangle \mid \llbracket\mathcal{A}_{\tilde{x},z_x}^{k+3}(P)_{g,\{X\to\tilde{n}\}}\rrbracket_{\tilde{c},\tilde{c}_r})$	$\widetilde{n} = \mathsf{fn}(P)$ $m = |\widetilde{n}|$ $\|\widetilde{n}\| = (\|n_1\|,\dots,\|n_m\|)$ $i \in \{1,\dots,m\}.$ $\qquad \|n_i\| : S_i$ $\qquad \|\widetilde{n}^i\| = (\|n_1^i\|,\dots,\|n_{|\mathcal{H}(S_i)|}^i\|)$ $\widetilde{c} = (c_{k+2},\dots,c_{k+\lfloor\|P\|_{g,\{X\to\tilde{n}\}}^1\rfloor+1})$ $\widetilde{c}_r = \bigcup_{v\in\widetilde{n}} c^v$
X	$(\nu s_1)(c_k?(z_x).\overline{c_{k+1}}!\langle z_x\rangle.\overline{c_{k+2}}!\langle z_x\rangle \mid$ $\qquad c_{k+1}?(z_x).(\nu a_1)(c^{n_1}!\langle a_1\rangle.(c_{k+2}?(z_x).\overline{s_1}!\langle z_x\rangle.\overline{c_{k+3}}!\langle\rangle \mid c_{k+3}?().\mathbf{0} \mid$ $\qquad\qquad a_1?(y_1).y_1?(\widetilde{z_1}).\dots.(\nu a_j)Q)))$ where: $Q = (c^{n_j}!\langle a_j\rangle.(a_j?(y_j).y_j?(\widetilde{z_j}).(\nu s')(z_x!\langle s'\rangle.\overline{s'}!\langle\widetilde{z_1},\dots,\widetilde{z_j},s_1\rangle)))$	$\widetilde{n} = g(X)$ $|\widetilde{n}| = j$ $i \in \{1,\dots,j\}$ $\qquad n_i : S \wedge \mathsf{tr}(S_i)$ $\qquad \widetilde{z_i} = (z_1^i,\dots,z_{|\mathcal{R}'^\star(S_i)|}^i)$

Table 2: Decompose by composition: Breakdown function for π processes (cf. Definition 3.6).

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 $\widetilde{u} = \mathsf{fn}(P)$ and $\widetilde{v} = \mathsf{rn}(P)$. Its decomposition is defined as:

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

where: $k > 0$, $\widetilde{c} = (c_k,\dots,c_{k+\lfloor P\rfloor-1})$; $\widetilde{c}_r = \bigcup_{r\in\widetilde{v}} c^r$; $\sigma = \{\mathsf{init}(\widetilde{u})/\widetilde{u}\}$; $P^r = c^r?(b).(\nu s)(b!\langle s\rangle.\overline{s}!\langle\overline{r}\rangle)$ with $r : S$ and $\widetilde{r} = r_1,\dots,r_{|\mathcal{G}(S)|}$. Also, the breakdown function $\mathcal{A}_{\tilde{x}}^k(\cdot)_g$ is as defined in Table 2.

3.3 Examples

Example 3.7 (Core Fragment). Let P be a π process which incorporates name-passing and w be a channel through which an integer value is sent. The degree of P is the same, mainly $\lfloor P\rfloor = 19$.

$$P = (\nu u)(u!\langle w\rangle.\overline{w}?(t).\mathbf{0} \mid \overline{u}?(x).x!\langle 5\rangle.\mathbf{0}) = (\nu u)(A \mid B)$$

Then, $\mathcal{F}(P) = (\nu c_1,\dots,c_{19})(\overline{c_1}!\langle\rangle.\mathbf{0} \mid (\nu u_1)\mathcal{A}_\epsilon^1((A \mid B)\sigma')_g)$, where $\sigma' = \{u_1\overline{u_1}/u\overline{u}\}$. We have:

$$\mathcal{A}_\epsilon^1((A \mid B)\sigma')_g) = c_1?().\overline{c_2}!\langle\rangle.\overline{c_{11}}!\langle\rangle \mid \mathcal{A}_\epsilon^2(A\sigma')_g \mid \mathcal{A}_\epsilon^{11}(B\sigma')_g$$

$$\mathcal{H}(\langle S \rangle) = \langle \mathcal{H}(S) \rangle$$

$$\mathcal{H}(!\langle S \rangle; S') = \begin{cases} M & \text{if } S' = \text{end} \\ M, \mathcal{H}(S') & \text{otherwise} \end{cases}$$

$$\text{where } M = !\langle \langle ?(?(\langle ?(\mathcal{H}(S)); \text{end} \rangle); \text{end}); \text{end} \rangle \rangle; \text{end}$$

$$\mathcal{H}(?(S); S') = \begin{cases} M & \text{if } S' = \text{end} \\ M, \mathcal{H}(S') & \text{otherwise} \end{cases}$$

$$\text{where } M = ?(\langle ?(?(\langle ?(\mathcal{H}(S)); \text{end} \rangle); \text{end}); \text{end} \rangle); \text{end}$$

$$\mathcal{H}(\text{end}) = \text{end}$$

$$\mathcal{H}(S_1, \ldots, S_n) = \mathcal{H}(S_1), \ldots, \mathcal{H}(S_n)$$

....

$$\mathcal{H}(\mu t.S) = \begin{cases} \mathcal{R}'(S) & \text{if } \mu t.S \text{ is tail-recursive} \\ \mu t.\mathcal{H}(S) & \text{otherwise} \end{cases}$$

$$\mathcal{R}'(!\langle S \rangle; S') = \mu t.!\langle \langle ?(?(\langle ?(\mathcal{H}(S)); \text{end} \rangle); \text{end}); \text{end} \rangle \rangle; t, \mathcal{R}'(S')$$

$$\mathcal{R}'(?(S); S') = \mu t.?(\langle ?(?(\langle ?(\mathcal{H}(S)); \text{end} \rangle); \text{end}); \text{end} \rangle); t, \mathcal{R}'(S')$$

$$\mathcal{H}(t) = t \qquad \mathcal{R}'(t) = \epsilon$$

....

$$\mathcal{R}'^{\star}(?(S); S') = \mathcal{R}'^{\star}(S') \qquad \mathcal{R}'^{\star}(!\langle S \rangle; S') = \mathcal{R}'^{\star}(S')$$

$$\mathcal{R}'^{\star}(\mu t.S) = \mathcal{R}'^{\star}(S)$$

Figure 9: Decomposition of types (cf. Def. 3.5)

The breakdown of A is:

$$\mathcal{A}_\epsilon^2(A)_g = c_2?().(\nu a_1)(u_1!\langle a_1 \rangle.(\overline{c_5}!\langle\rangle \mid \mathcal{A}_\epsilon^5(\overline{w_1}?(t).\mathbf{0})_g \mid$$
$$a_1?(y_1).y_1?(z_1).\overline{c_3}!\langle z_1 \rangle \mid$$
$$c_3?(z_1).z_1?(x).\overline{c_4}!\langle x \rangle \mid c_4?(x).(\nu s)(x!\langle s \rangle.\overline{s}!\langle w_1 \rangle)))$$

$$\mathcal{A}_\epsilon^5(\overline{w_1}?(t).\mathbf{0})_g = c_5?().\overline{w_1}?(y_2).\overline{c_6}!\langle y_2 \rangle \mid (\nu s_1)(c_6?(y_2).\overline{c_7}!\langle y_2 \rangle.\overline{c_8}!\langle\rangle \mid$$
$$c_7?(y_2).(\nu s')(y_2!\langle s' \rangle.\overline{s'}!\langle s_1 \rangle.\mathbf{0})) \mid$$
$$c_8?().(\nu a_2)(\overline{s_1}!\langle a_2 \rangle.(\overline{c_{10}}!\langle\rangle \mid c_{10}?().\mathbf{0} \mid$$
$$a_2?(y_3).y_3?(t_1).(\overline{c_9}!\langle\rangle \mid c_9?().\mathbf{0})))$$

The breakdown of B is similar and given in App. A.1.

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

$$r_1 : \mu t.?(\langle ?(?(\langle ?(\text{Int}); \text{end} \rangle); \text{end}); \text{end} \rangle); t$$
$$r_2 : \mu t.!\langle \langle ?(?(\langle ?(\text{Int}); \text{end} \rangle); \text{end}); \text{end} \rangle \rangle; t$$

The process $\mathcal{A}_{\tilde{x}}^k(P)_g$ is given in Fig. 11 (App. A.2).

3.4 Results

We establish the minimality result for π using 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., $c_k, c_{k+1}, \ldots$) generated by the breakdown function $\mathcal{B}^-(\cdot)$, environment Φ types shared propagators used in trios that propagate breakdown of recursive names (e.g., $c^r, c^v, \ldots$ 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 $(\!(\cdot)\!)^2$, we define Θ' and Φ' as follows: $\Theta' = (\!(\Theta)\!)^2, \Phi' = (\!(\Phi)\!)^2$.

We can use $\Theta' = (\!(\Theta)\!)^2$ in the following statement, where we state the typability result for the decomposition function. We find it convenient to handle finite processes first:

We now consider typability for processes with recursion, using $\Phi' = (\!(\Phi)\!)^2$ as in Definition 3.9:

Theorem 1 (Typability of Breakdown). *Let P be an initialized π process. If $\Gamma; \Delta, \Delta_\mu \vdash P \triangleright \diamond$, then $\mathcal{H}(\Gamma'), \Phi'; \mathcal{H}(\Delta), \Theta' \vdash \mathcal{A}_\epsilon^k(P)_g \triangleright \diamond$, where $k > 0; \widetilde{r} = \text{dom}(\Delta_\mu); \Phi' = \prod_{r \in \widetilde{r}} c^r : \langle ?(\mathcal{R}'^{\star}(\Delta_\mu(r))); \text{end} \rangle$; and* balanced$(\Theta')$ *with*

$$\text{dom}(\Theta') = \{c_k, c_{k+1}, \ldots, c_{k+\lceil P \rceil - 1}\} \cup \{\overline{c_{k+1}}, \ldots, \overline{c_{k+\lceil P \rceil - 1}}\}$$

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

Proof. Directly by using Theorem 5.1 [10], Theorem 3.27 [1], and Theorem 5.2 [10]. See Appendix C.2. □

Theorem 2 (Minimality Result for π). *Let P be a closed π process, with $\widetilde{u} = \text{fn}(P)$ and $\widetilde{v} = \text{rn}(P)$. 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 $\sigma = \{\text{init}(\widetilde{u})/\widetilde{u}\}$.*

Proof. Directly by using Thm. 1. See Appendix C.2. □

4 OPTIMIZATIONS

Although conceptually simple, the composition approach to decomposition induces redundancies. Here we propose an optimized variant of the breakdown function $\mathcal{A}_{\tilde{x}}^k(\cdot)_g$ 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 $\mathcal{A}_{\tilde{x}}^k(u_i?(w).Q)_g$ as presented in §3.1 and Tab. 2. We recognize 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 another the restricted name s_1 and so on. Finally, the name received for y' is receives $\widetilde{w}$, 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 $[\![\cdot]\!]_g^1$ and $[\![\cdot]\!]^2$.

Also, $\mathcal{A}_{\tilde{x}}^k(u_i?(w).Q)_g$ 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 $[\![\cdot]\!]^1_g$ (not present in the original process). Last but not least, the trio structure is lost as subprocess $\widehat{Q}_{\widetilde{x}}$ 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 $[\![\cdot]\!]^2$ in the composition.

The non-optimality of $\mathcal{A}^k_{\widetilde{x}}(\cdot)_g$ is more prominent in the treatment of recursive processes and recursive names. As HO does not feature recursion constructs, $[\![\cdot]\!]^1_g$ encodes recursive behaviors by relying on abstraction passing and shared abstractions. Then, going back to π via $[\![\cdot]\!]^2_g$, 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 that avoids the redundancies described above. The optimized decomposition, denoted $\mathcal{F}^*(\cdot)$ (Def. 4.10), 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* $\mathcal{G}(\cdot)$ is defined as in Figure 10.

Example 4.2 (Decomposing a Recursive Type). Let $S = \mu t.S'$ be a recursive session type, with $S' =?(\text{Int}); ?(\text{Bool}); !\langle \text{Bool} \rangle; t$. By Fig. 10, since S is tail-recursive, $\mathcal{G}(S) = \mathcal{R}(S')$. Further,

$$\mathcal{R}(S') = \mu t.?(\mathcal{G}(\text{Int})); t, \mathcal{R}(?(\text{Bool}); !\langle \text{Bool} \rangle; t)$$

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

$$\mathcal{G}(S) = \mu t.?(\text{Int}); t, \mu t.?(\text{Bool}); t, \mu t.!\langle \text{Bool} \rangle; t, \mathcal{R}(t)$$

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

$$\mathcal{G}(S) = \mu t.?(\text{Int}); t, \mu t.?(\text{Bool}); t, \mu t.!\langle \text{Bool} \rangle; t$$

Example 4.3 (Decomposing an Unfolded Recursive Type). Let $T = ?(\text{Bool}); !\langle \text{Bool} \rangle; S$ be a derived unfolding of S from Exam. 4.2. Then, by Fig. 10, $\mathcal{R}^\star(T)$ is the list of minimal recursive types obtained as follows: first, $\mathcal{R}^\star(T) = \mathcal{R}^\star(!\langle \text{Bool} \rangle; \mu t.S')$ and after one more step, $\mathcal{R}^\star(!\langle \text{Bool} \rangle; \mu t.S') = \mathcal{R}^\star(\mu t.S')$. Finally, we have $\mathcal{R}^\star(\mu t.S') = \mathcal{R}(S')$. We get the same list of minimal types as in Exam. 4.2:

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

Definition 4.4 (Decomposing Environments). Let Γ and Δ be typing environments. We define $\mathcal{G}(\Gamma)$ and $\mathcal{G}(\Delta)$ inductively as follows:

$$\mathcal{G}(\Delta, u_i : S) = \mathcal{G}(\Delta), (u_i, \ldots, u_{i+|\mathcal{G}(S)|-1}) : \mathcal{G}(S) \qquad \mathcal{G}(\emptyset) = \emptyset$$
$$\mathcal{G}(\Gamma, u_i : \langle S \rangle) = \mathcal{G}(\Gamma), u_i : \mathcal{G}(\langle S \rangle)$$

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

Figure 10: Type decomposition

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

$$\begin{cases} |Q| + 1 & \text{if } P = u_i!\langle y \rangle.Q \text{ or } P = u_i?(y).Q \\ |Q| + 1 & \text{if } P = (\nu s : S)Q \\ |Q| + |R| + 1 & \text{if } P = Q \mid R \\ 1 & \text{if } P = \mathbf{0} \\ |Q| + 1 & \text{if } P = \mu X.Q \\ 1 & \text{if } P = X \end{cases}$$

Definition 4.6 (Process Initialization). Let $\widetilde{u} = (a, b, s, s', \ldots)$ be a finite tuple of names. We shall write $\text{init}(\widetilde{u})$ to denote the tuple $(a_1, b_1, s_1, s'_1, \ldots)$. We will say that a process has been *initialized* if all of its names have some index.

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

Definition 4.7 (Free indexed names). Let $\widetilde{u}$ and $\widetilde{x}$ be two tuples of names. We define the set $\text{fnb}(\widetilde{u}, \widetilde{x})$ as $\{z_k : z_i \in \widetilde{u} \land z_k \in \widetilde{x}\}$.

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

Remark 1. *Whenever $c_k?(\widetilde{y})$ (resp. $\overline{c_k}!\langle \widetilde{y} \rangle$) with $\widetilde{y} = \epsilon$, 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. $\overline{c_k} :!\langle \langle \text{end} \rangle \rangle; \text{end}$).*

Definition 4.8 (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: $f_l'(!\langle U \rangle; S) = f_{l+1}'(S)$, $f_l'(?(U); S) = f_{l+1}'(S)$, and
$f_l'(\mu t.S) = |\mathcal{R}(S)| - l + 1$.

Definition 4.9 (Recursive subprocess predicate). Let P be a process. We write $\mathsf{X}(P)$ 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 $\mathcal{B}_{\widetilde{x}}^k(\cdot)$ (cf. § 4.3.1). Given a context $\widetilde{x}$ and a $k > 0$, $\mathcal{B}_{\widetilde{x}}^k(\cdot)$ is defined on initialized processes (cf. Def. 4.6). Table 3 gives the definition: we use an auxiliary function for recursive processes, denoted $\mathcal{B}_{\mathrm{rec}\ \widetilde{x}}^k(\cdot)_g$ (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 notationally heavy.

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

1. Input Process $\mathcal{B}_{\widetilde{x}}^k(u_i?(y).Q)$ consists of a leading trio that mimics the input and runs in parallel with the breakdown of Q. In the trio, a context $\widetilde{x}$ is expected along c_k. Then, an input on u_l mimics the input action: it expects the *decomposition* of name y, denoted $\widetilde{y}$. To decompose y we use its type: if $y : S$ then $\widetilde{y} = (y_1, \ldots, y_{|\mathcal{G}(S)|})$. The index of u_l depends on the type of u_i. Intuitively, if u_i is tail-recursive then $l = f(S)$ (Def. 4.8) 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 $\widetilde{z} = \mathsf{fnb}(Q, \widetilde{xy} \setminus \widetilde{w})$ is propagated, where $\widetilde{w} = (u_i)$ or $\widetilde{w} = \epsilon$.

2. Output Process $\mathcal{B}_{\widetilde{x}}^k(u_i!\langle y_j \rangle.Q)$ 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: $\widetilde{y} = (y_j, \ldots, y_{j+|\mathcal{G}(S)|-1})$. The context to be propagated is $\widetilde{z} = \mathsf{fnb}(P, \widetilde{x} \setminus \widetilde{w})$, where $\widetilde{w}$ and σ are as in the input case.

3. Restriction (Non-recursive name) The breakdown of process $(\nu s : C)Q$ is $(\nu \widetilde{s} : \mathcal{G}(C)) \mathcal{B}_{\widetilde{x}}^k(Q\sigma)$, where s is decomposed using C: $\widetilde{s} = (s_1, \ldots, s_{|\mathcal{G}(C)|})$. Since (νs) binds s and its dual $\bar{s}$ (or only s if C is a shared type) the substitution σ is simply $\{s_1\bar{s}_1/s\bar{s}\}$ 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 $\widetilde{s}$ 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 $\widetilde{s}$ and $\widetilde{\bar{s}}$ to the context, i.e., $\widetilde{z} = \widetilde{x}, \widetilde{s}, \widetilde{\bar{s}}$.

5. Composition The breakdown of process $Q_1 \mid Q_2$ 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., $\widetilde{x} = \epsilon$).

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

$$(\nu c_X^r)(c_k?(\widetilde{x}).\overline{c_{k+1}^r}!\langle \widetilde{z} \rangle.\mu X.c_X^r?(\widetilde{y}).\overline{c_{k+1}^r}!\langle \widetilde{y} \rangle.X \mid \mathcal{B}_{\mathrm{rec}\ \widetilde{z}}^{k+1}(P)_g)$$

We have a control trio and the breakdown of P, obtained using $\mathcal{B}_{\mathrm{rec}\ \widetilde{x}}^k(\cdot)_g$ (§ 4.3.2). The trio receives the context $\widetilde{x}$ on c_k and propagates it to the next trio. To ensure typability, we bind all session free names of P using the context $\widetilde{z}$, which contains the

decomposition of those free names. This context is needed to break down P, and so we record it as $g = \{X \mapsto \widetilde{z}\}$ in the definition of $\mathcal{B}_{\mathrm{rec}\ \widetilde{x}}^{k+1}(P)_g$. This way, $\widetilde{z}$ will be propagated all the way until reaching X.

Next, the recursive trio is enabled, and receives $\widetilde{y}$ along c_X^r, with $|\widetilde{z}| = |\widetilde{y}|$ and $l = |P|$. The tuple $\widetilde{y}$ is propagated to the first trio of $\mathcal{B}_{\mathrm{rec}\ \widetilde{x}}^{k+1}(P)_g$. By definition of $\mathcal{B}_{\mathrm{rec}\ \widetilde{x}}^{k+1}(P)_g$, 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 $\mathcal{B}_{\mathrm{rec}\ \widetilde{x}}^{k+1}(P)_g$.

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 μX.P. As already mentioned, we use the auxiliary function $\mathcal{B}_{\mathrm{rec}\ \widetilde{x}}^k(\cdot)_g$ 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 $\widetilde{z}$ 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 $\mathsf{X}(Q_1)$. We take $\widetilde{y}_1 = g(X) \cup \mathsf{fnb}(Q_1, \widetilde{x})$ 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 $\widetilde{x}$ 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 $\widetilde{x} = g(X)$.

We may now define the optimized process decomposition:

Definition 4.10 (Decomposing Processes). Let P be a π process with $\widetilde{u} = \mathsf{fn}(P)$ and $\widetilde{v} = \mathsf{rn}(P)$. The *decomposition* of P, denoted $\mathcal{F}^*(P)$, is defined as $\mathcal{F}^*(P) = (\nu \widetilde{c})(\overline{c_k}!\langle \widetilde{r} \rangle.\mathbf{0} \mid \mathcal{B}_{\widetilde{r}}^k(P\sigma))$, where: $k > 0$; $\widetilde{c} = (c_k, \ldots, c_{k+|P|-1})$; $\widetilde{r}$ such that for $v \in \widetilde{v}$ and $v : S$ $(v_1, \ldots, v_{|\mathcal{R}(S)|}) \subseteq \widetilde{r}$; and $\sigma = \{\mathsf{init}(\widetilde{u})/\widetilde{u}\}$.

4.4 Examples

We illustrate $\mathcal{F}^*(\cdot)$, $\mathcal{B}_{\widetilde{x}}^k(\cdot)$, and $\mathcal{B}_{\mathrm{rec}\ \widetilde{x}}^k(\cdot)_g$ by means of examples.

Example 4.11. Consider a simple mathematical server that first receives a channel along which it subsequently receives two integers to compare. Let $S = !\langle \mathsf{Int} \rangle; !\langle \mathsf{Int} \rangle; \mathsf{end}$ and $T = ?(\overline{S}); !\langle \mathsf{Bool} \rangle; \mathsf{end}$.

	P	$\mathcal{B}^k_{\tilde{x}}(P)$	
1	$u_i?(y).Q$	$c_k?(\tilde{x}).u_l?(\tilde{y}).\overline{c_{k+1}}!\langle\tilde{z}\rangle \mid \mathcal{B}^{k+1}_{\tilde{z}}(Q\sigma)$	$y_j : S \wedge \tilde{y} = (y_1,\ldots,y_{\|S\|})$ $\tilde{w} = (\mathtt{lin}(u_i)) ? \{u_i\} : \epsilon$ $\tilde{z} = \mathsf{fnb}(Q, \widetilde{xy} \setminus \tilde{w})$ $l = (\mathtt{tr}(u_i)) ? f(S) : i$ $\sigma = \mathsf{next}(u_i) \cdot \{y_1/y\}$
2	$u_i!\langle y_j\rangle.Q$	$c_k?(\tilde{x}).u_l!\langle\tilde{y}\rangle.\overline{c_{k+1}}!\langle\tilde{z}\rangle \mid \mathcal{B}^{k+1}_{\tilde{z}}(Q\sigma)$	$y_j : S \wedge \tilde{y} = (y_j,\ldots,y_{j+\|\mathcal{G}(S)\|-1})$ $\tilde{w} = (\mathtt{lin}(u_i)) ? \{u_i\} : \epsilon$ $\tilde{z} = \mathsf{fnb}(Q, \tilde{x} \setminus \tilde{w})$ $l = (\mathtt{tr}(u_i)) ? f(S) : i$ $\sigma = \mathsf{next}(u_i)$
3	$(\nu s : C)Q$	$(\nu\tilde{s} : \mathcal{G}(C))\, \mathcal{B}^k_{\tilde{x}}(Q\sigma)$	$\tilde{s} = (s_1,\ldots,s_{\|\mathcal{G}(C)\|})$ $\sigma = \{s_1\overline{s_1}/s\overline{s}\}$
4	$(\nu s : \mu t.S)Q$	$(\nu\tilde{s} : \mathcal{R}(S)) (c_k?(\tilde{x}).\overline{c_{k+1}}!\langle\tilde{z}\rangle.\mathbf{0} \mid \mathcal{B}^{k+1}_{\tilde{z}}(Q))$	$\mathtt{tr}(\mu t.S) \quad \tilde{s} = (s_1,\ldots,s_{\|\mathcal{R}(S)\|})$ $\tilde{z} = \tilde{x}, \tilde{s}, \tilde{\tilde{s}} \quad \tilde{\tilde{s}} = (\overline{s_1},\ldots,\overline{s_{\|\mathcal{R}(S)\|}})$
5	$Q_1 \mid Q_2$	$c_k?(\tilde{x}).\overline{c_{k+1}}!\langle\tilde{y}\rangle.\overline{c_{k+l+1}}!\langle\tilde{z}\rangle \mid \mathcal{B}^{k+1}_{\tilde{y}}(Q_1) \mid \mathcal{B}^{k+l+1}_{\tilde{z}}(Q_2)$	$\tilde{y} = \mathsf{fnb}(Q_1,\tilde{x}) \quad \tilde{z} = \mathsf{fnb}(Q_2,\tilde{x})$ $l = \|Q_1\|$
6	$\mathbf{0}$	$c_k?().\mathbf{0}$	
7	$\mu X.P$	$(\nu c^r_X)(c_k?(\tilde{x}).\overline{c^r_{k+1}}!\langle\tilde{z}\rangle.\mu X.c^r_X?(\tilde{y}).\overline{c^r_{k+1}}!\langle\tilde{y}\rangle.X \mid \mathcal{B}^{k+1}_{\mathsf{rec}\ \tilde{z}}(P)_g)$	$\tilde{n} = \mathsf{fs}(P) \quad \tilde{n} : \tilde{C} \wedge \tilde{z} = \mathsf{bn}(\tilde{n} : \tilde{C})$ $\|\tilde{z}\| = \|\tilde{y}\| \quad g = \{X \mapsto \tilde{z}\}$

	P	$\mathcal{B}^k_{\mathsf{rec}\ \tilde{x}}(P)_g$	
8	$u_i!\langle y_j\rangle.Q$	$\mu X.c^r_k?(\tilde{x}).u_l!\langle\tilde{y}\rangle.\overline{c^r_{k+1}}!\langle\tilde{z}\rangle.X \mid \mathcal{B}^{k+1}_{\mathsf{rec}\ \tilde{z}}(Q\sigma)_g \qquad (\text{if } g \neq \emptyset)$ $\mu X.c^r_k?(\tilde{x}).(u_l!\langle\tilde{y}\rangle.\overline{c^r_{k+1}}!\langle\tilde{z}\rangle \mid X) \mid \mathcal{B}^{k+1}_{\mathsf{rec}\ \tilde{z}}(Q\sigma)_g \qquad (\text{if } g = \emptyset)$	$y : T \wedge (y_j,\ldots,y_{j+\|\mathcal{G}(T)\|-1})$ $\tilde{w} = (\mathtt{lin}(u_i)) ? \{u_i\} : \epsilon$ $\tilde{z} = g(X) \cup \mathsf{fnb}(Q, \tilde{x} \setminus \tilde{w})$ $l = (\mathtt{tr}(u_i)) ? f(S) : i$ $\sigma = \mathsf{next}(u_i)$
9	$u_i?(y).Q$	$\mu X.c^r_k?(\tilde{x}).u_l?(\tilde{y}).\overline{c^r_{k+1}}!\langle\tilde{z}\rangle.X \mid \mathcal{B}^{k+1}_{\mathsf{rec}\ \tilde{z}}(Q\sigma)_g \qquad (\text{if } g \neq \emptyset)$ $\mu X.c^r_k?(\tilde{x}).(u_l?(\tilde{y}).\overline{c^r_{k+1}}!\langle\tilde{z}\rangle \mid X) \mid \mathcal{B}^{k+1}_{\mathsf{rec}\ \tilde{z}}(Q\sigma)_g \qquad (\text{if } g = \emptyset)$	$\tilde{w} = (\mathtt{lin}(u_i)) ? \{u_i\} : \epsilon$ $\tilde{z} = g(X) \cup \mathsf{fnb}(Q, \widetilde{xy} \setminus \tilde{w})$ $l = (\mathtt{tr}(u_i)) ? f(S) : i$ $\sigma = \mathsf{next}(u_i) \cdot \{y_1/y\}$
10	$Q_1 \mid Q_2$	$\mu X.c^r_k?(\tilde{x}).(\overline{c^r_{k+1}}!\langle\tilde{y_1}\rangle.X \mid c^r_{k+l+1}!\langle\tilde{y_2}\rangle) \mid \qquad (\text{if } g \neq \emptyset)$ $\quad \mathcal{B}^{k+1}_{\mathsf{rec}\ \tilde{y_1}}(Q_1)_g \mid \mathcal{B}^{k+l+1}_{\mathsf{rec}\ \tilde{y_2}}(Q_2)_\emptyset$ $\mu X.c^r_k?(\tilde{x}).(c^r_{k+1}!\langle\tilde{y_1}\rangle \mid c^r_{k+l+1}!\langle\tilde{y_2}\rangle \mid X) \mid$ $\quad \mathcal{B}^{k+1}_{\mathsf{rec}\ \tilde{y_1}}(Q_1)_\emptyset \mid \mathcal{B}^{k+l+1}_{\mathsf{rec}\ \tilde{y_2}}(Q_2)_\emptyset \qquad (\text{if } g = \emptyset)$	$X(Q_1)$ $\tilde{y_1} = g(X) \cup \mathsf{fnb}(Q_1,\tilde{x})$ $\tilde{y_2} = \mathsf{fnb}(Q_2,\tilde{x})$ $l = \|Q_1\|$
11	$(\nu s : C)Q$	$\mu X.(\nu\tilde{s} : \mathcal{G}(C))c^r_k?(\tilde{x}).\overline{c^r_{k+1}}!\langle\tilde{z}\rangle.X \mid \mathcal{B}^{k+1}_{\mathsf{rec}\ \tilde{z}}(Q\sigma)_g \qquad (\text{if } g \neq \emptyset)$ $\mu X.(\nu\tilde{s} : \mathcal{G}(C))c^r_k?(\tilde{x}).(c^r_{k+1}!\langle\tilde{z}\rangle \mid X) \mid \mathcal{B}^{k+1}_{\mathsf{rec}\ \tilde{z}}(Q\sigma)_g \qquad (\text{if } g = \emptyset)$	$\tilde{s} = (s_1,\ldots,s_{\|\mathcal{G}(S)\|})$ $\tilde{\tilde{s}} = (\mathtt{lin}(S)) ? (\overline{s_1},\ldots,\overline{s_{\|\mathcal{G}(S)\|}}) : \epsilon$ $\tilde{z} = \tilde{x}, \tilde{s}, \tilde{\tilde{s}} \qquad \sigma = \{s_1\overline{s_1}/s\overline{s}\}$
12	X	$\mu X.c^r_k?(\tilde{x}).\overline{c^r_X}!\langle\tilde{x}\rangle X \qquad (\text{if } g \neq \emptyset)$ $\mu X.c^r_k?().(\overline{c^r_X}!\langle\rangle \mid X) \qquad (\text{if } g = \emptyset)$	
13	$\mathbf{0}$	$c^r_k?().\mathbf{0}$	

Table 3: Optimized breakdown function for processes, and auxiliary function for recursive processes.

Further, let Q be a process implementing T along name v:

$$Q = v?(x).x?(y).x?(z).v!\langle y == z\rangle.\mathbf{0}$$

Let $R = (\nu u : S)\overline{v}!\langle\overline{u}\rangle.u!\langle 5\rangle.u!\langle 3\rangle.\overline{v}?(eq).\mathbf{0}$ be a client for the server R. We then have:

$$\mathcal{F}^*(Q \mid R) = (\nu\tilde{c})(\overline{c_1}!\langle\rangle \mid c_1?().\overline{c_2}!\langle\rangle.\overline{c_7}!\langle\rangle \mid \mathcal{B}^1_\epsilon(Q) \mid \mathcal{B}^7_\epsilon(R))$$

$$\mathcal{B}^1_\epsilon(Q) = c_2?().v_1?(x_1, x_2).\overline{c_3}!\langle x_1, x_2\rangle \mid$$
$$c_3?(x_1, x_2).x_1?(y_1).\overline{c_4}!\langle x_2, y_1\rangle \mid c_4?(x_2, y_1).x_2?(z_1).\overline{c_5}!\langle y_1, z_1\rangle \mid$$
$$c_5?(y_1, z_1).v_2!\langle y_1 == z_1\rangle.\overline{c_6}!\langle\rangle \mid c_6?().\mathbf{0}$$

$$\mathcal{B}_\epsilon^7(R) = (\nu\, u_1, u_2) c_7?().\overline{v}_1!\langle \overline{u}_1, \overline{u}_2\rangle.\overline{c_8}!\langle\rangle \mid c_8?().u_1!\langle 5\rangle.\overline{c_9}!\langle\rangle \mid$$
$$c_9?().u_2!\langle 3\rangle.\overline{c_{10}}!\langle\rangle \mid c_{10}?().\overline{v}_1?(eq).\overline{c_{11}}!\langle\rangle \mid c_{11}?().\mathbf{0}$$

Name v is decomposed as the indexed names v_1, v_2, whose MSTs are, respectively, ?(?(Int); end, ?(Int); end); end and !⟨Bool⟩; end. Let us analyze the reductions of $\mathcal{F}^*(Q \mid R)$ using some abbreviations for subprocesses of Q and R:

$$Q = v?(x).Q' \qquad Q' = x?(y).Q'' \qquad Q'' = x?(z).v!\langle y == z\rangle.\mathbf{0}$$
$$R = (\nu\, u : S)\overline{v}!\langle \overline{u}\rangle.R' \qquad R' = u!\langle 5\rangle.R'' \qquad R'' = u!\langle 3\rangle.\overline{v}?(eq).\mathbf{0}$$

After a few administrative reductions, $\mathcal{F}^*(Q \mid R)$ mimics the first source communication:

$$\mathcal{F}^*(Q \mid R) \longrightarrow^* v_1?(x_1, x_2).\overline{c_3}!\langle x_1, x_2\rangle \mid \mathcal{B}_{x_1, x_2}^2(Q') \mid$$
$$\overline{v}_1!\langle \overline{u}_1, \overline{u}_2\rangle.\overline{c_8}!\langle\rangle \mid \mathcal{B}_\epsilon^8(R')$$
$$\longrightarrow \overline{c_3}!\langle \overline{u}_1, \overline{u}_2\rangle \mid \mathcal{B}_{x_1, x_2}^2(Q') \mid \overline{c_8}!\langle\rangle \mid \mathcal{B}_\epsilon^8(R') = P_1$$

Notice that the decomposition of $\overline{u}$ (i.e., $\overline{u}_1, \overline{u}_2$) that is received in the first leading trio is propagated as a context to the first trio in $\mathcal{B}_{x_1, x_2}^2(Q')$. After two reductions (on c_3 and c_8), name $\overline{u}_1$ substitutes x_1 and the communication along u_1 can be mimicked:
$$P_1 \longrightarrow^* \overline{u}_1?(y_1).\overline{c_4}!\langle x_2, y_1\rangle \mid \mathcal{B}_{x_2, y_1}^3(Q'') \mid u_1!\langle 5\rangle.\overline{c_9}!\langle\rangle \mid \mathcal{B}_\epsilon^9(R'').$$

Example 4.12 (Example 3.8, Revisited). Consider again the tail-recursive session type $S = \mu t.?(Int);!\langle Int\rangle;t$. Also, let R be a process implementing a channel r with type with S as follows:

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

We decompose r using S and obtain two channels typed with MSTs as in Fig. 10. We have: $r_1 : \mu t.?(Int);t$ and $r_2 : \mu t.!\langle Int\rangle;t$.

The trios produced by $\mathcal{B}_\epsilon^k(R)$ 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 $\widetilde{x}$ 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 $\mathcal{B}_{r_1, r_2}^1(R)$:

$$c_1?(r_1, r_2).\overline{c_2^r}!\langle r_1, r_2\rangle.\mu X.c_X^r?(y_1, y_2).\overline{c_2^r}!\langle y_1, y_2\rangle.X \mid \mathcal{B}_{\mathsf{rec}\ r_1, r_2}^2(R')$$

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

$$\mu X.c_2^r?(y_1, y_2).r_1?(z_1).\overline{c_3^r}!\langle y_1, y_2, z_1\rangle.X \mid$$
$$\mu X.c_3^r?(y_1, y_2, z_1).r_2?(-z_1).\overline{c_4^r}!\langle y_1, y_2\rangle.X \mid$$
$$\mu X.c_4^r?(y_1, y_2).\overline{c_X^r}!\langle y_1, y_2\rangle.X$$

We can see that 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 action of the recursion body. Finally, the last recursive trio is able to send 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 Results

4.5.1 Static Correctness.
To state the typability of $\mathcal{B}_{\widetilde{x}}^k(\cdot)$, we define an auxiliary predicate:

Definition 4.13 (Indexed Names). Suppose some typing environments Γ, Δ. Let $\widetilde{x}, \widetilde{y}$ be two tuples of indexed names. We write $\mathsf{indexed}_{\Gamma, \Delta}(\widetilde{y}, \widetilde{x})$ for the predicate

$$\forall z_i.(z_i \in \widetilde{x} \Leftrightarrow ((z_i, \ldots, z_{i+m-1}) \subseteq \widetilde{y}) \wedge m = |\mathcal{G}((\Gamma, \Delta)(z_i))|))$$

The following theorem ensures typability of $\mathcal{B}_{\widetilde{x}}^k(\cdot)$ (see App. C.3):

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

$$\mathcal{G}(\Gamma \setminus \widetilde{x}); \mathcal{G}(\Delta \setminus \widetilde{x}), \Theta \vdash \mathcal{B}_{\widetilde{y}}^k(P) \triangleright \diamond \quad (k > 0)$$

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

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

and $\Theta(c_k) = ?(\widetilde{M}); end,$ where $\widetilde{M} = (\mathcal{G}(\Gamma), \mathcal{G}(\Delta))(\widetilde{y})$.

PROOF. By the induction of the structure of P; see § C.3. □

We now (re)state the minimality result for π, now based on the optimized decomposition in Def. 4.10.

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

PROOF. Direct by using Lem. 3 (see App. C.4). □

4.5.2 Dynamic Correctness.
We have established that P and $\mathcal{F}^*(P)$ are behaviorally equivalent, a result that complements the minimality result by specifying the dynamic correctness of our optimized decomposition. Below we briefly overview the required notions and the correctness result; see App. C.5 and App. C.6 for details.

The operational correspondence in Thm. 5 relies on *MST-bisimilarity* (cf. Def. 4.18, $\approx^{\mathsf{M}}$), 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. We need the following notion:

Definition 4.14 (Indexed name). Given a name n, we write $\check{n}$ to either denote n or any indexed name n_i, with $i > 0$.

Suppose we wish to relate P and Q using $\approx^{\mathsf{M}}$, and that P performs an output action involving name v. In our setting, Q should send a *tuple* of names, namely the decomposition of v. The second difference is that output objects should be related by a relation $\bowtie_{\mathsf{c}}$ (Def. 4.15) defined as follows:

Definition 4.15 (Relating names). We define $\bowtie_{\mathsf{c}}$ as the relation on names defined as

$$\frac{}{\epsilon \bowtie_{\mathsf{c}} \epsilon} \qquad \frac{\Gamma; \Delta \vdash n_i \triangleright C}{n_i \bowtie_{\mathsf{c}} (n_i, \ldots, n_{i+|\mathcal{G}(C)|-1})} \qquad \frac{\check{n} \bowtie_{\mathsf{c}} \tilde{m}_1 \quad n_i \bowtie_{\mathsf{c}} \tilde{m}_2}{\check{n}, n_i \bowtie_{\mathsf{c}} \tilde{m}_1, \tilde{m}_2}$$

where ϵ denotes the empty list.

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

$$t \Leftarrow_{\mathsf{C}} v : C \overset{\mathsf{def}}{=} t?(x).(\nu s)(s?(y).[\![C]\!]^y \mid \overline{s}!\langle v\rangle.\mathbf{0})$$

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

Our variant of trigger processes is defined as follows:

Definition 4.17 (Minimal characteristic trigger process). Given a type C, the trigger process is

$$t \Leftarrow_{\mathsf{m}} v_i : C \overset{\mathsf{def}}{=} t_1?(x).(\nu s_1)(s_1?(\widetilde{y}).\langle C\rangle_i^y \mid \overline{s_1}!\langle \widetilde{v}\rangle.\mathbf{0})$$

where $v_i \bowtie_{\mathsf{c}} \widetilde{v}$, $y_i \bowtie_{\mathsf{c}} \widetilde{y}$, 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 4.18 (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 $\Gamma_1; \Delta_1 \vdash P_1 \xrightarrow{(\nu \widetilde{m_1})n!\langle v:C_1\rangle} \Delta_1'; \Lambda_1' \vdash P_2$ then there exist Q_2, Δ_2', and σ_v such that $\Gamma_2; \Delta_2 \vdash Q_1 \xRightarrow{(\nu \widetilde{m_2})\check{n}!\langle \check{v}:\mathcal{G}(C)\rangle} \Delta_2' \vdash Q_2$ where $v\sigma_v \bowtie_{\mathsf{c}} \widetilde{v}$ and, for a fresh t,

$$\Gamma; \Delta_1'' \vdash (\nu \widetilde{m_1})(P_2 \parallel t \Leftarrow_{\mathsf{C}} v : C_1) \mathfrak{R}$$
$$\Delta_2'' \vdash (\nu \widetilde{m_2})(Q_2 \parallel t \Leftarrow_{\mathsf{m}} v\sigma : C_1)$$

(2) Whenever $\Gamma_1; \Delta_1 \vdash P_1 \xrightarrow{n?(v)} \Delta_1' \vdash P_2$ then there exist Q_2, Δ_2', and σ_v such that $\Gamma_2; \Delta_2 \vdash Q_1 \xRightarrow{\check{n}?(\check{v})} \Delta_2' \vdash Q_2$ where $v\sigma_v \bowtie_{\mathsf{c}} \widetilde{v}$ 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 $\mathsf{sub}(\ell) = n$ implies $\mathsf{sub}(\hat{\ell}) = \check{n}$.

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

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

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*

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

PROOF. By coinduction: we exhibit a binary relation $\mathcal{S}$ that contains $(P, \mathcal{F}^*(P))$ and prove that it is an MST bisimulation. We define $\mathcal{S}$ (Def. C.10) using two auxiliary functions $C_-^-(\,\cdot\,)$ and $\mathcal{J}_-^-(\,\cdot\,)$ (Tab. 4 and Tab. 5). The proof that $\mathcal{S}$ is an MST bisimilarity is given by Lem. C.7 and Lem. C.8 (see App. C.6 for details). □

5 DISCUSSION

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 work complements the results in [1], and shows that the minimality result is independent from the kind of communicated objects (names or abstractions). From a practical standpoint, the decomposition used in our approach can be seen as a technique for implementing session types into programming languages whose type systems do not support sequentiality.

Sequentiality is the key distinguishing feature in the specification of message-passing programs using session types. We remark that by our minimality results, based on MSTs, do not mean that sequentiality in session types is redundant in *modeling and specifying* processes; rather, we claim that it is not indispensable as a primitive concept in *type-checking* them. Indeed, by decomposing processes, we are able to type-check *session typed* processes using type systems that do not directly support sequencing in types.

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.3 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. Our work relies on the trios decompositions proposed by Parrow [11], which he studied for an untyped π-calculus with replication and without choices; in contrast, π processes feature recursion. We stress that our goal is to clarify the role of sequentiality in session types by using processes typable with MSTs, which lack sequentiality. While trios processes elegantly induce processes typable with MSTs, developing trios decompositions for π is just one possible path towards our goal.

Our work differs significantly with respect to [1]. As already discussed, 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 to our approach is the work by Dardha et al. [2]. They compile a session π-calculus down into a π-calculus equipped with the linear type system of [7] extended with variant types. They represent sequentiality in types 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 communications. 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. A detailed comparison between our approach and [2] is interesting follow-up work.

REFERENCES

[1] Alen Arslanagic, 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 pi-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

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 w be a channel through which an integer value is sent. The degree of P is the same, mainly $\lfloor P \rfloor = 19$.

$$P = (\nu u)(u!\langle w \rangle.\overline{w}?(t).\mathbf{0} \mid \overline{u}?(x).x!\langle 5 \rangle.\mathbf{0}) = (\nu u)(A \mid B)$$

Then, the decomposition of P into a collection of first-order processes typed with minimal session types is:

$$\mathcal{F}(P) = (\nu c_1, \ldots, c_{19})\big(\overline{c_1}!\langle\rangle.\mathbf{0} \mid \mathcal{A}_\epsilon^1(P\sigma)_g\big)$$

$$= (\nu c_1, \ldots, c_{19})\big(\overline{c_1}!\langle\rangle.\mathbf{0} \mid \mathcal{A}_\epsilon^1((\nu u)(A \mid B)\sigma)_g\big)$$

$$= (\nu c_1, \ldots, c_{19})\big(\overline{c_1}!\langle\rangle.\mathbf{0} \mid (\nu u_1)\mathcal{A}_\epsilon^1((A \mid B)\sigma')_g\big), \text{ where } \sigma' = \{u_1\overline{u_1}/u\overline{u}\}$$

We have:

$$\mathcal{A}_\epsilon^1((A \mid B)\sigma')_g = c_1?().\overline{c_2}!\langle\rangle.\overline{c_{11}}!\langle\rangle \mid \mathcal{A}_\epsilon^2(A\sigma')_g \mid \mathcal{A}_\epsilon^{11}(B\sigma')_g$$

The breakdown of A is:

$$\mathcal{A}_\epsilon^2(A)_g = c_2?().(\nu a_1)(u_1!\langle a_1 \rangle.(\overline{c_5}!\langle\rangle \mid \mathcal{A}_\epsilon^5(\overline{w_1}?(t).\mathbf{0})_g \mid a_1?(y_1).y_1?(z_1).\overline{c_3}!\langle z_1 \rangle \mid$$
$$c_3?(z_1).z_1?(x).\overline{c_4}!\langle x \rangle \mid c_4?(x).(\nu s)(x!\langle s \rangle.\overline{s}!\langle w_1 \rangle))))$$

$$\mathcal{A}_\epsilon^5(\overline{w_1}?(t).\mathbf{0})_g = c_5?().\overline{w_1}?(y_2).\overline{c_6}!\langle y_2 \rangle \mid (\nu s_1)(c_6?(y_2).\overline{c_7}!\langle y_2 \rangle.\overline{c_8}!\langle\rangle \mid$$
$$c_7?(y_2).(\nu s')(y_2!\langle s' \rangle.\overline{s'}!\langle s_1 \rangle.\mathbf{0})) \mid c_8?().(\nu a_2)(\overline{s_1}!\langle a_2 \rangle.(\overline{c_{10}}!\langle\rangle \mid c_{10}?().\mathbf{0} \mid$$
$$a_2?(y_3).y_3?(t_1).(\overline{c_9}!\langle\rangle \mid c_9?().\mathbf{0})))$$

The breakdown of B is:

$$\mathcal{A}_\epsilon^{11}(B)_g = c_{11}?().\overline{u_1}?(y_4).\overline{c_{12}}!\langle y_4 \rangle \mid (\nu s_1)(c_{12}?(y).\overline{c_{13}}!\langle y \rangle.\overline{c_{14}}!\langle\rangle \mid$$
$$c_{13}?(y_4).(\nu s'')(y_4!\langle s'' \rangle.\overline{s''}!\langle s_1 \rangle.\mathbf{0}) \mid c_{14}?().(\nu a_3)(s_1!\langle a_3 \rangle.(\overline{c_{19}}!\langle\rangle \mid$$
$$c_{19}?().\mathbf{0} \mid a_3?(y_5).y_5?(x_1).(\overline{c_{15}}!\langle\rangle \mid \mathcal{A}_\epsilon^{15}(x_1!\langle 5 \rangle.\mathbf{0})_g))))$$

$$\mathcal{A}_\epsilon^{15}(x_1!\langle 5 \rangle.\mathbf{0})_g = c_{15}?().(\nu a_4)(x_1!\langle a_4 \rangle.(\overline{c_{18}}!\langle\rangle \mid c_{18}?().\mathbf{0} \mid a_4?(y_6).y_6?(z_1).\overline{c_{16}}!\langle z_1 \rangle \mid$$
$$c_{16}?(z_1).z_1?(x).\overline{c_{17}}!\langle x \rangle \mid c_{17}?(x).(\nu s''')(x!\langle s''' \rangle.\overline{s'''}!\langle 5 \rangle)))$$

Now let us observe the reduction chain. As before, the collection of processes synchronizes on c_1, c_2, c_{11} after three reductions.

$$\mathcal{F}(P) \longrightarrow^3 (\nu \widetilde{c})((\nu a_1)(u_1!\langle a_1 \rangle.(\overline{c_5}!\langle\rangle \mid \mathcal{A}_\epsilon^5(\overline{w_1}?(t).\mathbf{0})_g \mid a_1?(y_1).y_1?(z_1).\overline{c_3}!\langle z_1 \rangle \mid$$
$$c_3?(z_1).z_1?(x).\overline{c_4}!\langle x \rangle \mid c_4?(x).(\nu s)(x!\langle s \rangle.\overline{s}!\langle w_1 \rangle)))) \mid u_1?(y_4).\overline{c_{12}}!\langle y_4 \rangle \mid$$
$$(\nu s_1)(c_{12}?(y_4).\overline{c_{13}}!\langle y_4 \rangle.\overline{c_{14}}!\langle\rangle \mid c_{13}?(y_4).(\nu s'')(y_4!\langle s'' \rangle.\overline{s''}!\langle s_1 \rangle.\mathbf{0}) \mid$$
$$c_{14}?().(\nu a_3)(s_1!\langle a_3 \rangle.(\overline{c_{19}}!\langle\rangle \mid c_{19}?().\mathbf{0} \mid a_3?(y_5).y_5?(x_1).(\overline{c_{15}}!\langle\rangle \mid \mathcal{A}_\epsilon^{15}(x_1!\langle 5 \rangle.\mathbf{0})_g))))))$$
$$\text{where } \widetilde{c} = (c_3, \ldots, c_{10}, c_1, \ldots, c_{19})$$

Then, the broken down process A communicates with process B through channel u_1 by passing name a_1, which is further sent on c_{12} and c_{13}. Another synchronization occurs on c_{14}.

$$\mathcal{F}(P) \longrightarrow^7 (\nu \widetilde{c})(\nu a_1)(\overline{c_5}!\langle\rangle \mid \mathcal{A}_\epsilon^5(\overline{w_1}?(t).\mathbf{0})_g \mid a_1?(y_1).y_1?(z_1).\overline{c_3}!\langle z_1 \rangle \mid c_3?(z_1).z_1?(x).\overline{c_4}!\langle x \rangle \mid$$
$$c_4?(x).(\nu s)(x!\langle s \rangle.\overline{s}!\langle w_1 \rangle)) \mid (\nu s_1)((\nu s'')(a_1!\langle s'' \rangle.\overline{s''}!\langle s_1 \rangle.\mathbf{0}) \mid$$
$$(\nu a_3)(s_1!\langle a_3 \rangle.(\overline{c_{19}}!\langle\rangle \mid c_{19}?().\mathbf{0} \mid a_3?(y_5).y_5?(x_1).(\overline{c_{15}}!\langle\rangle \mid \mathcal{A}_\epsilon^{15}(x_1!\langle 5 \rangle.\mathbf{0})_g)))))$$
$$\text{where } \widetilde{c} = (c_3, \ldots, c_{10}, c_{15}, \ldots, c_{19})$$

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

$$\mathcal{F}(P) \longrightarrow^8 (\nu \widetilde{c})(\nu s'')(\overline{c_5}!\langle\rangle \mid \mathcal{A}_\epsilon^5(\overline{w_1}?(t).\mathbf{0})_g \mid s''?(z_1).\overline{c_3}!\langle z_1 \rangle \mid c_3?(z_1).z_1?(x).\overline{c_4}!\langle x \rangle \mid$$
$$c_4?(x).(\nu s)(x!\langle s \rangle.\overline{s}!\langle w_1 \rangle)) \mid (\nu s_1)(\overline{s''}!\langle s_1 \rangle.\mathbf{0} \mid (\nu a_3)(s_1!\langle a_3 \rangle.(\overline{c_{19}}!\langle\rangle \mid c_{19}?().\mathbf{0} \mid$$
$$a_3?(y_5).y_5?(x_1).(\overline{c_{15}}!\langle\rangle \mid \mathcal{A}_\epsilon^{15}(x_1!\langle 5 \rangle.\mathbf{0})_g)))))$$
$$\text{where } \widetilde{c} = (c_3, \ldots, c_{10}, c_{15}, \ldots, c_{19})$$

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.

$$\mathcal{F}(P) \longrightarrow^{10} (\nu \widetilde{c})(\nu s_1)(\ \overline{c_5}!\langle\rangle \mid \mathcal{A}_\epsilon^5\big(\overline{w_1}?(t).\mathbf{0}\big)_g \mid s_1?(x).\overline{c_4}!\langle x\rangle \mid c_4?(x).(\nu s)(x!\langle s\rangle.\overline{s}!\langle w_1\rangle) \mid$$
$$(\nu a_3)(s_1!\langle a_3\rangle.\big(\overline{c_{19}}!\langle\rangle \mid c_{19}?().\mathbf{0} \mid a_3?(y_5).y_5?(x_1).\big(\overline{c_{15}}!\langle\rangle \mid \mathcal{A}_\epsilon^{15}\big(x_1!\langle 5\rangle.\mathbf{0}\big)_g\big)\big)\big)\ )$$
$$\text{where } \widetilde{c} = (c_4, \ldots, c_{10}, c_{15}, \ldots, c_{19})$$

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_{19}.

$$\mathcal{F}(P) \longrightarrow^{13} (\nu \widetilde{c})(\nu a_3)(\ \overline{c_5}!\langle\rangle \mid \mathcal{A}_\epsilon^5\big(\overline{w_1}?(t).\mathbf{0}\big)_g \mid (\nu s)(a_3!\langle s\rangle.\overline{s}!\langle w_1\rangle) \mid$$
$$a_3?(y_5).y_5?(x_1).\big(\overline{c_{15}}!\langle\rangle \mid \mathcal{A}_\epsilon^{15}\big(x_1!\langle 5\rangle.\mathbf{0}\big)_g\big)\ )$$
$$\text{where } \widetilde{c} = (c_5, \ldots, c_{10}, c_{15}, \ldots, c_{18})$$

Now, the distribution of w_1 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 name w_1, substituting the variable x_1:

$$\mathcal{F}(P) \longrightarrow^{14} (\nu \widetilde{c})(\nu s)(\ \overline{c_5}!\langle\rangle \mid \mathcal{A}_\epsilon^5\big(\overline{w_1}?(t).\mathbf{0}\big)_g \mid \overline{s}!\langle w_1\rangle \mid s?(x_1).\big(\overline{c_{15}}!\langle\rangle \mid \mathcal{A}_\epsilon^{15}\big(x_1!\langle 5\rangle.\mathbf{0}\big)_g\big)\ )$$
$$\text{where } \widetilde{c} = (c_5, \ldots, c_{10}, c_{15}, \ldots, c_{18})$$
$$\mathcal{F}(P) \longrightarrow^{15} (\nu \widetilde{c})(\ \overline{c_5}!\langle\rangle \mid \mathcal{A}_\epsilon^5\big(\overline{w_1}?(t).\mathbf{0}\big)_g \mid \overline{c_{15}}!\langle\rangle \mid \mathcal{A}_\epsilon^{15}\big(x_1!\langle 5\rangle.\mathbf{0}\big)_g\{w_1/x_1\}\ )$$
$$\text{where } \widetilde{c} = (c_5, \ldots, c_{10}, c_{15}, \ldots, c_{18})$$

Replacing $\mathcal{A}_\epsilon^5\big(\overline{w_1}?(t).\mathbf{0}\big)_g$ and $\mathcal{A}_\epsilon^{15}\big(x_1!\langle 5\rangle.\mathbf{0}\big)_g$ by their definitions, as well as synchronizing on c_5 and c_{15}, we have:

$$\mathcal{F}(P) \longrightarrow^{17} (\nu \widetilde{c})(\ \overline{w_1}?(y_2).\overline{c_6}!\langle y_2\rangle \mid (\nu s_1)(c_6?(y_2).\overline{c_7}!\langle y_2\rangle.\overline{c_8}!\langle\rangle \mid c_7?(y_2).(\nu s')(y_2!\langle s'\rangle.\overline{s'}!\langle s_1\rangle.\mathbf{0})) \mid$$
$$c_8?().(\nu a_2)\big(\overline{s_1}!\langle a_2\rangle.\big(\overline{c_{10}}!\langle\rangle \mid c_{10}?().\mathbf{0} \mid a_2?(y_3).y_3?(t_1).\big(\overline{c_9}!\langle\rangle \mid c_9?().\mathbf{0}\big)\big)\big) \mid$$
$$(\nu a_4)(w_1!\langle a_4\rangle.\big(\overline{c_{18}}!\langle\rangle \mid c_{18}?().\mathbf{0} \mid a_4?(y_6).y_6?(z_1).\overline{c_{16}}!\langle z_1\rangle \mid$$
$$c_{16}?(z_1).z_1?(x).\overline{c_{17}}!\langle x\rangle \mid c_{17}?(x).(\nu s''')(x!\langle s'''\rangle.\overline{s'''}!\langle 5\rangle)))\ )$$
$$\text{where } \widetilde{c} = (c_6, \ldots, c_{10}, c_{16}, \ldots, c_{18})$$

Clearly, the following reduction involves communication on channel w_1, through which the name a_4 is passed. This name is further passed via c_6 and c_7, eventually substituting the variable y_2.

$$\mathcal{F}(P) \longrightarrow^{20} (\nu \widetilde{c})(\nu a_4)(\ (\nu s_1)(\overline{c_8}!\langle\rangle \mid (\nu s')(a_4!\langle s'\rangle.\overline{s'}!\langle s_1\rangle.\mathbf{0})) \mid$$
$$c_8?().(\nu a_2)\big(\overline{s_1}!\langle a_2\rangle.\big(\overline{c_{10}}!\langle\rangle \mid c_{10}?().\mathbf{0} \mid a_2?(y_3).y_3?(t_1).\big(\overline{c_9}!\langle\rangle \mid c_9?().\mathbf{0}\big)\big)\big) \mid$$
$$\big(\overline{c_{18}}!\langle\rangle \mid c_{18}?().\mathbf{0} \mid a_4?(y_6).y_6?(z_1).\overline{c_{16}}!\langle z_1\rangle \mid$$
$$c_{16}?(z_1).z_1?(x).\overline{c_{17}}!\langle x\rangle \mid c_{17}?(x).(\nu s''')(x!\langle s'''\rangle.\overline{s'''}!\langle 5\rangle))\ )$$
$$\text{where } \widetilde{c} = (c_8, \ldots, c_{10}, c_{16}, \ldots, c_{18})$$

Now, channel a_4 can send the session s' to its dual endpoint. Thus, s' replaces variable y_6:

$$\mathcal{F}(P) \longrightarrow^{21} (\nu \widetilde{c})(\nu s')(\ (\nu s_1)(\overline{c_8}!\langle\rangle \mid \overline{s'}!\langle s_1\rangle.\mathbf{0}) \mid c_8?().(\nu a_2)\big(\overline{s_1}!\langle a_2\rangle.\big(\overline{c_{10}}!\langle\rangle \mid$$
$$c_{10}?().\mathbf{0} \mid a_2?(y_3).y_3?(t_1).\big(\overline{c_9}!\langle\rangle \mid c_9?().\mathbf{0}\big)\big)\big) \mid \overline{c_{18}}!\langle\rangle \mid c_{18}?().\mathbf{0} \mid s'?(z_1).\overline{c_{16}}!\langle z_1\rangle \mid$$
$$c_{16}?(z_1).z_1?(x).\overline{c_{17}}!\langle x\rangle \mid c_{17}?(x).(\nu s''')(x!\langle s'''\rangle.\overline{s'''}!\langle 5\rangle)\ )$$
$$\text{where } \widetilde{c} = (c_8, \ldots, c_{10}, c_{16}, \ldots, c_{18})$$

The next immediate step consists in communication between s' and its dual, implying that s_1 substitutes z_1. Following this interaction, s_1 is further sent along c_{16}. Additionally, propagators c_8 and c_{18} also synchronize with their respective duals.

$$\mathcal{F}(P) \longrightarrow^{25} (\nu \widetilde{c})(\nu s_1)(\ (\nu a_2)\big(\overline{s_1}!\langle a_2\rangle.\big(\overline{c_{10}}!\langle\rangle \mid c_{10}?().\mathbf{0} \mid a_2?(y_3).y_3?(t_1).\big(\overline{c_9}!\langle\rangle \mid c_9?().\mathbf{0}\big)\big)\big) \mid$$
$$s_1?(x).\overline{c_{17}}!\langle x\rangle \mid c_{17}?(x).(\nu s''')(x!\langle s'''\rangle.\overline{s'''}!\langle 5\rangle)\ )$$
$$\text{where } \widetilde{c} = (c_9, c_{10}, c_{17})$$

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

where:

$$\widehat{P} = (\nu\widetilde{c})\big(c^r?(b).(\nu s')(b!\langle s'\rangle.\overline{s}'!\langle x_{r_1},x_{r_2}\rangle) \mid \overline{c_1}!\langle\rangle \mid c_1?().y_1?(z_x).\overline{c_2}!\langle z_x\rangle \mid R^2\{x_{r_1},x_{r_2}/r_1,r_2\}\big)$$

$$R^k = c_k?(z_x).(\nu a_1)\big(c^r!\langle a_1\rangle.((\nu s_1)(c_{k+1}?(y).\overline{c_{k+2}}!\langle y\rangle.\overline{c_{k+3}}!\langle\rangle \mid$$
$$c_{k+2}?(y).(\nu s)(y!\langle s\rangle.\overline{s}!\langle s_1\rangle) \mid c_{k+3}?().(\nu a_2)(s_1!\langle a_2\rangle.(\overline{c_{k+l+4}}!\langle\rangle \mid c_{k+l+4}?().\mathbf{0} \mid$$
$$a_2?(y_2).\boxed{y_2?(w_1).}(\nu\widetilde{c})(\overline{c_{k+4}}!\langle z_x\rangle \mid \mathcal{A}_{z_x}^{k+4}\big(r_2!\langle -w_1\rangle.X\big)_g))) \mid$$
$$a_1?(y_1).y_1?(z_1,z_2).\boxed{z_1?(y).}\overline{c_{k+1}}!\langle y\rangle.c^r?(b).(\nu s')(b!\langle s'\rangle.\overline{s}'!\langle z_1,z_2\rangle)))))$$

$$\mathcal{A}_{z_x}^{k+4}\big(r_2!\langle -w_1\rangle.X\big)_g = c_k?(z_x).(\nu a_1)c^r!\langle a_1\rangle.(\mathcal{A}_{z_x}^{k+7}\big(X\big)_g \mid a_1?(y_1).y_1?(\widetilde{z}).W)$$

$$W = (\nu a_2)\big(\boxed{z_2!\langle a_2\rangle.}(\overline{c_{k+7}}!\langle z_x\rangle.c^r?(b).(\nu s)(b!\langle s\rangle.\overline{s}!\langle\widetilde{z}\rangle)) \mid$$
$$a_2?(y_2).y_2?(z_1').(\nu\widetilde{c})(\overline{c_{k+5}}!\langle\rangle \mid$$
$$c_{k+5}?().z_1'?(x).\overline{c_{k+6}}!\langle x\rangle \mid c_{k+6}?(x).(\nu s')(x!\langle s'\rangle.\boxed{\overline{s}'!\langle -w1\rangle})))))$$

$$\mathcal{A}_{z_x}^{k+7}\big(X\big)_g = (\nu s_1)\big(c_{k+7}?(z_x).\overline{c_{k+8}}!\langle z_x\rangle.\overline{c_{k+9}}!\langle z_x\rangle \mid$$
$$c_{k+8}?(z_x).(\nu a_1)(c^r!\langle a_1\rangle.(c_{k+9}?(z_x).\overline{s}_1!\langle z_x\rangle.\overline{c_{k+10}}!\langle\rangle \mid c_{k+10}?().\mathbf{0} \mid$$
$$a_1?(y_1).y_1?(r_1,r_2).(\nu s')(z_x!\langle s'\rangle.\overline{s}'!\langle r_1,r_2,s_1\rangle)))))$$

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

Figure 11: Breakdown of recursive process

Then, the name a_2 is sent by $\overline{s_1}$, after which is further communicated by c_{17}, such that it substitutes variable x.

$$\mathcal{F}(P) \longrightarrow^{27} (\nu\widetilde{c})(\nu a_2)(\,\overline{c_{10}}!\langle\rangle \mid c_{10}?().\mathbf{0} \mid a_2?(y_3).y_3?(t_1).(\overline{c_9}!\langle\rangle \mid c_9?().\mathbf{0}) \mid$$
$$(\nu s''')(a_2!\langle s'''\rangle.\overline{s'''}!\langle 5\rangle)\,)$$
$$\text{where } \widetilde{c} = (c_9,c_{10})$$

The channel a_2 can now pass s''', which replaces variable y_3:

$$\mathcal{F}(P) \longrightarrow^{28} (\nu\widetilde{c})(\nu s''')(\,\overline{c_{10}}!\langle\rangle \mid c_{10}?().\mathbf{0} \mid s'''?(t_1).(\overline{c_9}!\langle\rangle \mid c_9?().\mathbf{0})\overline{s'''}!\langle 5\rangle\,)$$
$$\text{where } \widetilde{c} = (c_9,c_{10})$$

Finally, the process simulates the distribution on integer 5 by using s''' as a communication channel. In addition, propagator c_{10} also synchronizes with its corresponding endpoint.

$$\mathcal{F}(P) \longrightarrow^{28} (\nu\widetilde{c})(\,\overline{c_9}!\langle\rangle \mid c_9?().\mathbf{0}\,), \text{ where } \widetilde{c} = (c_9)$$

The ending interaction occurs on propagator c_9, simulating inaction. Undoubtedly, the behaviour of the initial first-order process has been fully simulated. Thus, the outcome of our decomposition function is a functionally equivalent process that is typed with minimal session types.

A.2 First Decomposition: Recursion

Example A.2 (Recursive process (Example 3.8)). Let $S = \mu t.?(\text{Int});!\langle\text{Int}\rangle;t$ be a tail-recursive session type. Also, let $P = \mu X.P'$ be a process implementing a channel r with type with S, with $P' = r?(w).r!\langle -w\rangle.X$. We decompose r using S and obtain two channels typed with MSTs as in Fig. 9. We have:

$$r_1 : \mu t.?\big(\langle?(?(\langle?(\text{Int});\text{end}\rangle);\text{end});\text{end}\rangle\big);t$$
$$r_2 : \mu t.!\big\langle\langle?(?(\langle?(\text{Int});\text{end}\rangle);\text{end});\text{end}\rangle\big\rangle;t$$

We decomposition of P is

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

where $\mathcal{A}_\epsilon^1\big(P\{r_1/r\}\big)_\emptyset$ is in Fig. 11.

We can see that $\mathcal{A}_\epsilon^1\big(P\{r_1/r\}\big)$ simulates a behavior of a recursive process by the replication. In $\mathcal{A}_\epsilon^1\big(P\{r_1/r\}\big)$ subprocess R^5 is a breakdown of the first instance of the recursive body. After a few reductions on propagators, this process gets activated by receiving a_1 for z_x along c_5. Intuitively, R^5 will send recursive names along name a_1 for use in a next instance. Replication guarded by a_1 produces a next instance: that is process $R^2\{x_{r_1}, x_{r_2}/r_1, r_2\}$ nested in $\widehat{P}$. By communication on a_1 and a few more reductions on propagators, it gets activated: along a_1 it first receives a name for y_1' along which it further retrieves: (i) recursive name r_1, r_2 for variables x_{r_1}, x_{r_2}, and (ii) a name for y_1 along which it will receive name a_1 again, as can be seen in $\mathcal{A}_{z_x}^{k+7}(X)_g$, for future instances.

Process R^k is parameterized in index k (used in propagators to avoid name clashes), which mimics actions of recursive body. Process R^k first gets a decomposition of the recursive name r by communication 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 R^k, cf Fig. 11). After 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 highlighted in W, cf. Fig. 11).

A.3 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 HOπ, these constructs are denoted

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

with reduction rule

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

Example A.3 (Labelled Choice). Let P and Q be π processes which incorporate branching and selection:

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

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:

$$\mathcal{F}(R) = (\nu\,c_1, \ldots, c_5)(\overline{c_1}!\langle\rangle.\mathbf{0} \mid \mathcal{A}_\epsilon^1\big((\nu\,u)(P \mid Q)\sigma\big)_g), \text{ where } \sigma = \{u_1\overline{u_1}/u\overline{u}\}$$
$$= (\nu\,c_1, \ldots, c_5)(\overline{c_1}!\langle\rangle.\mathbf{0} \mid (\nu\,u_1)(c_1?().\overline{c_2}!\langle\rangle.\overline{c_3}!\langle\rangle \mid \mathcal{A}_\epsilon^2(P)_g \mid \mathcal{A}_\epsilon^3(Q)_g))$$

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

$$\mathcal{A}_\epsilon^3(Q)_g = c_3?().(\nu\,a_1)(\overline{c_4}!\langle a_1\rangle.\big((\nu\,u_2)(c_4?(y).(\nu\,s)(y!\langle s\rangle.\overline{s}!\langle\overline{u_2}\rangle))\big) \mid c_5?().\mathbf{0} \mid$$
$$a_1?(y').y'?(y_1).u_1 \triangleleft l_1.u_1?(z).\overline{c_5}!\langle\rangle.(\nu\,s')(z!\langle s'\rangle.\overline{s}'!\langle y_1\rangle)))$$

The breakdown of process P is:

$$\mathcal{A}_\epsilon^2(P)_g = c_2?().u_1 \triangleright \{l_1 : (\nu\,a_2)u_1!\langle a_2\rangle.a_2?(y'').y''?(y_1).(\nu\,c_3, \ldots, c_{21})(\overline{c_3}!\langle\rangle \mid \mathcal{A}_\epsilon^3\big(P_1\{y_1/u_1\}\big)_g),$$
$$l_2 : (\nu\,a_2)u_1!\langle a_2\rangle.a_2?(y'').y''?(y_1).(\nu\,c_3, \ldots, c_{11})(\overline{c_3}!\langle\rangle \mid \mathcal{A}_\epsilon^3\big(P_2\{y_1/u_1\}\big)_g)\}$$

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

$$\mathcal{A}_\epsilon^3\big(P_1\{y_1/u_1\}\big)_g = (\nu\,h_1, h_2)\mathcal{A}_\epsilon^3\big(h_1!\langle m_1\rangle.h_1?(t).\mathbf{0} \mid \overline{h_1}?(x).\overline{h_1}!\langle x\rangle.\mathbf{0}\big)_g$$
$$= (\nu\,h_1, h_2)(c_3?().\overline{c_4}!\langle\rangle.\overline{c_{13}}!\langle\rangle \mid \mathcal{A}_\epsilon^4\big(h_1!\langle m_1\rangle.h_1?(t).\mathbf{0}\big)_g \mid \mathcal{A}_\epsilon^{13}\big(\overline{h_1}?(x).\overline{h_1}!\langle x\rangle.\mathbf{0}\big)_g)$$

$$\mathcal{A}^4_\epsilon\big(h_1!\langle m_1\rangle.g\,h_1?(t).\mathbf{0}\big) = c_4?().(\nu\,a_3)(h_1!\langle a_3\rangle.\big(\overline{c_7}!\langle\rangle \mid \mathcal{A}^7_\epsilon\big(h_2?(t).\mathbf{0}\big)_g \mid a_3?(y_2).y_2?(z_1).\overline{c_5}!\langle z_1\rangle \mid$$
$$c_5?(z_1).z_1?(x).\overline{c_6}!\langle x\rangle \mid c_6?(x).(\nu\,s_1)(x!\langle s_1\rangle.\overline{s_1}!\langle m_1\rangle.\mathbf{0})))$$

$$\mathcal{A}^7_\epsilon\big(h_2?(t).\mathbf{0}\big)_g = c_7?().h_2?(y_3).\overline{c_8}!\langle y_3\rangle \mid (\nu\,s_2)(c_8?(y_3).\overline{c_9}!\langle y_3\rangle.\overline{c_{10}}!\langle\rangle \mid$$
$$c_9?(y_3).(\nu\,s_3)(y_3!\langle s_3\rangle.\overline{s_3}!\langle s_2\rangle.\mathbf{0}) \mid c_{10}?().(\nu\,a_4)\big(\overline{s_2}!\langle a_4\rangle.\big(\overline{c_{12}}!\langle\rangle \mid$$
$$c_{12}?().\mathbf{0} \mid a_4?(y_4).y_4?(t_1).\overline{c_{11}}!\langle\rangle \mid c_{11}?().\mathbf{0}\,))$$

$$\mathcal{A}^{13}_\epsilon\big(\overline{h_1}?(x).\overline{h_1}!\langle x\rangle.\mathbf{0}\big)_g = c_{13}?().\overline{h_1}?(y_5).\overline{c_{14}}!\langle y_5\rangle \mid (\nu\,s_4)(c_{14}?(y_5).\overline{c_{15}}!\langle y_5\rangle.\overline{c_{16}}!\langle\rangle \mid$$
$$c_{15}?(y_5).(\nu\,s_5)(y_5!\langle s_5\rangle.\overline{s_5}!\langle s_4\rangle.\mathbf{0}) \mid c_{16}?().(\nu\,a_5)\big(\overline{s_4}!\langle a_5\rangle.\big(\overline{c_{21}}!\langle\rangle \mid$$
$$c_{21}?().\mathbf{0} \mid a_5?(y_6).y_6?(x_1).\overline{c_{17}}!\langle x_1\rangle \mid \mathcal{A}^{17}_{x_1}\big(\overline{h_2}!\langle x_1\rangle.\mathbf{0}\big)_g\,)))$$

$$\mathcal{A}^{17}_{x_1}\big(\overline{h_2}!\langle x_1\rangle.\mathbf{0}\big)_g = c_{17}?(x_1).(\nu\,a_6)(\overline{h_2}!\langle a_6\rangle.\big(\overline{c_{20}}!\langle x_1\rangle \mid c_{20}?().\mathbf{0} \mid a_6?(y_7).y_7?(z_2).\overline{c_{18}}!\langle z_2\rangle \mid$$
$$c_{18}?(z_2).z_2?(x').\overline{c_{19}}!\langle x'\rangle \mid c_{19}?(x').(\nu\,s_6)(x'!\langle s_6\rangle.\overline{s_6}!\langle x_1\rangle.\mathbf{0})))$$

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.

$$\mathcal{F}(R) \longrightarrow^4 (\nu\,c_5)(\nu\,u_1)(u_1 \triangleright \{l_1 : (\nu\,a_2)\,u_1!\langle a_2\rangle.a_2?(y'').y''?(y_1).(\nu\,c_3,\dots,c_{21})(\overline{c_3}!\langle\rangle \mid$$
$$\mathcal{A}^3_\epsilon\big(P_1\{y_1/u_1\}\big)_g), \; l_2 : (\nu\,a_2)\,u_1!\langle a_2\rangle.a_2?(y'').y''?(y_1).(\nu\,c_3,\dots,c_{11})(\overline{c_3}!\langle\rangle \mid$$
$$\mathcal{A}^3_\epsilon\big(P_2\{y_1/u_1\}\big)_g)\} \mid (\nu\,a_1)\big((\nu\,u_2)((\nu\,s)(a_1!\langle s\rangle.\overline{s}!\langle\overline{u_2}\rangle)) \mid c_5?().\mathbf{0} \mid$$
$$a_1?(y').y'?(y_1).u_1 \triangleleft l_1.u_1?(z).\overline{c_5}!\langle\rangle.(\nu\,s')(z!\langle s'\rangle.\overline{s'}!\langle y_1\rangle))))$$

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

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

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

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

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

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

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.

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

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

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

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 $\mathsf{HO}\pi$

$$\text{(Sess)} \qquad \text{(Sh)}$$
$$\Gamma; \emptyset; \{u : S\} \vdash u \triangleright S \qquad \Gamma, u : U; \emptyset; \emptyset \vdash u \triangleright U$$

$$\text{(LVar)} \qquad\qquad \text{(RVar)}$$
$$\Gamma; \{x : C \multimap \diamond\}; \emptyset \vdash x \triangleright C \multimap \diamond \qquad \Gamma, X : \Delta; \emptyset; \Delta \vdash X \triangleright \diamond$$

$$\text{(Abs)} \qquad\qquad\qquad\qquad \text{(App)}$$
$$\frac{\Gamma; \Lambda; \Delta_1 \vdash P \triangleright \diamond \quad \Gamma; \emptyset; \Delta_2 \vdash x \triangleright C}{\Gamma \backslash x; \Lambda; \Delta_1 \backslash \Delta_2 \vdash \lambda x. P \triangleright C \multimap \diamond} \qquad \frac{\Gamma; \Lambda; \Delta_1 \vdash V \triangleright C \rightsquigarrow \diamond \quad \rightsquigarrow \in \{\multimap, \rightarrow\} \quad \Gamma; \emptyset; \Delta_2 \vdash u \triangleright C}{\Gamma; \Lambda; \Delta_1, \Delta_2 \vdash V u \triangleright \diamond}$$

$$\text{(Prom)} \qquad\qquad \text{(EProm)} \qquad\qquad \text{(End)}$$
$$\frac{\Gamma; \emptyset; \emptyset \vdash V \triangleright C \multimap \diamond}{\Gamma; \emptyset; \emptyset \vdash V \triangleright C \rightarrow \diamond} \qquad \frac{\Gamma, \Lambda, x : C \multimap \diamond; \Delta \vdash P \triangleright \diamond}{\Gamma, x : C \rightarrow \diamond; \Lambda; \Delta \vdash P \triangleright \diamond} \qquad \frac{\Gamma; \Lambda; \Delta \vdash P \triangleright T \quad u \notin \mathrm{dom}(\Gamma, \Lambda, \Delta)}{\Gamma; \Lambda; \Delta, u : \mathsf{end} \vdash P \triangleright \diamond}$$

$$\text{(Rec)} \qquad\qquad \text{(Par)} \qquad\qquad\qquad \text{(Nil)}$$
$$\frac{\Gamma, X : \Delta; \emptyset; \Delta \vdash P \triangleright \diamond}{\Gamma; \emptyset; \Delta \vdash \mu X. P \triangleright \diamond} \qquad \frac{\Gamma; \Lambda_i; \Delta_i \vdash P_i \triangleright \diamond \quad i = 1, 2}{\Gamma; \Lambda_1, \Lambda_2; \Delta_1, \Delta_2 \vdash P_1 \mid P_2 \triangleright \diamond} \qquad \frac{}{\Gamma; \emptyset; \emptyset \vdash \mathbf{0} \triangleright \diamond}$$

$$\text{(Send)}$$
$$\frac{u : S \in \Delta_1, \Delta_2 \quad \Gamma; \Lambda_1; \Delta_1 \vdash P \triangleright \diamond \quad \Gamma; \Lambda_2; \Delta_2 \vdash V \triangleright U}{\Gamma; \Lambda_1, \Lambda_2; ((\Delta_1, \Delta_2) \setminus u : S), u :! \langle U \rangle; S \vdash u! \langle V \rangle. P \triangleright \diamond}$$

$$\text{(Req)}$$
$$\frac{\Gamma; \Lambda; \Delta_1 \vdash P \triangleright \diamond \quad \Gamma; \emptyset; \emptyset \vdash u \triangleright \langle \mathcal{U} \rangle \quad \Gamma; \emptyset; \Delta_2 \vdash V \triangleright \mathcal{U} \quad \mathcal{U} \in \{S, L\}}{\Gamma; \Lambda; \Delta_1, \Delta_2 \vdash u! \langle V \rangle. P \triangleright \diamond}$$

$$\text{(Acc)}$$
$$\text{(Rcv)} \qquad\qquad\qquad\qquad\qquad \frac{\Gamma; \Lambda_1; \Delta_1 \vdash P \triangleright \diamond \quad \Gamma; \emptyset; \emptyset \vdash u \triangleright \langle \mathcal{U} \rangle}{}$$
$$\frac{\Gamma; \Lambda_1; \Delta_1, u : S \vdash P \triangleright \diamond \quad \Gamma; \Lambda_2; \Delta_2 \vdash x \triangleright U}{\Gamma \backslash x; \Lambda_1 \backslash \Lambda_2; \Delta_1 \backslash \Delta_2, u :?(U); S \vdash u?(x). P \triangleright \diamond} \qquad \frac{\Gamma; \Lambda_2; \Delta_2 \vdash x \triangleright \mathcal{U} \quad \mathcal{U} \in \{S, L\}}{\Gamma \backslash x; \Lambda_1 \backslash \Lambda_2; \Delta_1 \backslash \Delta_2 \vdash u?(x). P \triangleright \diamond}$$

$$\text{(Bra)} \qquad\qquad\qquad\qquad\qquad \text{(Sel)}$$
$$\frac{\forall i \in I \quad \Gamma; \Lambda; \Delta, u : S_i \vdash P_i \triangleright \diamond}{\Gamma; \Lambda; \Delta, u : \&\{l_i : S_i\}_{i \in I} \vdash u \triangleright \{l_i : P_i\}_{i \in I} \triangleright \diamond} \qquad \frac{\Gamma; \Lambda; \Delta, u : S_j \vdash P \triangleright \diamond \quad j \in I}{\Gamma; \Lambda; \Delta, u : \oplus\{l_i : S_i\}_{i \in I} \vdash u \triangleleft l_j. P \triangleright \diamond}$$

$$\text{(ResS)} \qquad\qquad\qquad\qquad \text{(Res)}$$
$$\frac{\Gamma; \Lambda; \Delta, s : S_1, \overline{s} : S_2 \vdash P \triangleright \diamond \quad S_1 \text{ dual } S_2}{\Gamma; \Lambda; \Delta \vdash (\nu s) P \triangleright \diamond} \qquad \frac{\Gamma, a : \langle S \rangle; \Lambda; \Delta \vdash P \triangleright \diamond}{\Gamma; \Lambda; \Delta \vdash (\nu a) P \triangleright \diamond}$$

Figure 12: Typing Rules for $\mathsf{HO}\pi$ (including selection and branching constructs).

Remark 2. *We derive polyadic rules for typing π as an expected extension of π typing rules in Fig. 13.*

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

The typed LTS combines the LTSs in Fig. 14 and Fig. 15.

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 $\longrightarrow$, $\xoverset{\ell}{\Longrightarrow}$ for the transitions $\Longrightarrow \xrightarrow{\ell} \Longrightarrow$, and $\xoverset{\hat{\ell}}{\Longrightarrow}$ for $\xoverset{\ell}{\Longrightarrow}$ if $\ell \neq \tau$ otherwise $\Longrightarrow$.

(PolyVar)
$$\Gamma, \widetilde{x} : \widetilde{M}_x; \widetilde{y} : \widetilde{M}_y \vdash \widetilde{xy} \triangleright \widetilde{M}_x \widetilde{M}_y$$

(PolySend)
$$\frac{\Gamma; \Delta_1, u : S \vdash P \triangleright \diamond \quad \Gamma; \Delta_2 \vdash \widetilde{x} \triangleright \widetilde{C}}{\Gamma; \Delta_1, \Delta_2, u : !\langle \widetilde{C} \rangle; S \vdash u!\langle \widetilde{x} \rangle.P \triangleright \diamond}$$

(PolyRcv)
$$\frac{\Gamma; \Delta_1, u : S \vdash P \triangleright \diamond \quad \Gamma; \Delta_2 \vdash \widetilde{x} \triangleright \widetilde{C}}{\Gamma \backslash \widetilde{x}; \Delta_1 \backslash \Delta_2, u : ?(\widetilde{C}); S \vdash u?(\widetilde{x}).P \triangleright \diamond}$$

(PolyResS)
$$\frac{\Gamma; \Delta, \widetilde{s} : \widetilde{S}_1, \widetilde{\overline{s}} : \widetilde{S}_2 \vdash P \triangleright \diamond \quad S_1 \text{ dual } S_2}{\Gamma; \Delta \vdash (\nu \widetilde{s})P \triangleright \diamond}$$

(PolyRes)
$$\frac{\Gamma; \Delta, \widetilde{a} : \widetilde{C} \vdash P \triangleright \diamond}{\Gamma; \Delta \vdash (\nu \widetilde{a})P \triangleright \diamond}$$

(Req)
$$\frac{\Gamma; \emptyset \vdash u \triangleright \langle \widetilde{S} \rangle \quad \Gamma; \Delta_1 \vdash P \triangleright \diamond \quad \Gamma; \Delta_2 \vdash \widetilde{x} \triangleright \widetilde{S}}{\Gamma; \Delta_1, \Delta_2 \vdash u!\langle \widetilde{x} \rangle.P \triangleright \diamond}$$

(Acc)
$$\frac{\Gamma; \emptyset \vdash u \triangleright \langle \widetilde{S} \rangle \quad \Gamma; \Delta_1 \vdash P \triangleright \diamond \quad \Gamma; \Delta_2 \vdash \widetilde{x} \triangleright \widetilde{S}}{\Gamma \backslash \widetilde{x}; \Delta_1 \backslash \Delta_2 \vdash u?(\widetilde{x}).P \triangleright \diamond}$$

Figure 13: Polyadic typing rules for π.

$\langle\text{App}\rangle$
$$(\lambda x.P)\, V \xrightarrow{\tau} P\{V/x\}$$

$\langle\text{Snd}\rangle$
$$n!\langle V \rangle.P \xrightarrow{n!\langle V \rangle} P$$

$\langle\text{Rv}\rangle$
$$n?(x).P \xrightarrow{n?\langle V \rangle} P\{V/x\}$$

$\langle\text{Sel}\rangle$
$$s \triangleleft l.P \xrightarrow{s \oplus l} P$$

$\langle\text{Bra}\rangle$
$$\frac{j \in I}{s \triangleright \{l_i : P_i\}_{i \in I} \xrightarrow{s \& l_j} P_j}$$

$\langle\text{Alpha}\rangle$
$$\frac{P \equiv_\alpha Q \quad Q \xrightarrow{\ell} P'}{P \xrightarrow{\ell} P'}$$

$\langle\text{Res}\rangle$
$$\frac{P \xrightarrow{\ell} P' \quad n \notin \mathsf{fn}(\ell)}{(\nu n)P \xrightarrow{\ell} (\nu n)P'}$$

$\langle\text{New}\rangle$
$$\frac{P \xrightarrow{(\nu \widetilde{m})n!\langle V \rangle} P' \quad m_1 \in \mathsf{fn}(V)}{(\nu m_1)P \xrightarrow{(\nu m_1, \widetilde{m})n!\langle V \rangle} P'}$$

$\langle\text{Par}_L\rangle$
$$\frac{P \xrightarrow{\ell} P' \quad \mathsf{bn}(\ell) \cap \mathsf{fn}(Q) = \emptyset}{P \mid Q \xrightarrow{\ell} P' \mid Q}$$

$\langle\text{Tau}\rangle$
$$\frac{P \xrightarrow{\ell_1} P' \quad Q \xrightarrow{\ell_2} Q' \quad \ell_1 \asymp \ell_2}{P \mid Q \xrightarrow{\tau} (\nu \, \mathsf{bn}(\ell_1) \cup \mathsf{bn}(\ell_2))(P' \mid Q')}$$

$\langle\text{Rec}\rangle$
$$\frac{P\{\mu X.P/X\} \xrightarrow{\ell} P'}{\mu X.P \xrightarrow{\ell} P'}$$

Figure 14: The Untyped LTS for $\mathsf{HO}\pi$ processes. We omit Rule $\langle\text{Par}_R\rangle$.

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

B.3 Encoding of $\mathsf{HO}\pi$ into HO

Definition B.2 (Auxiliary Mappings). We define mappings $\| \cdot \|$ and $\left\| \cdot \right\|_\sigma$ as follows:

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

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

- Given a set of session names and variables σ, the map $\left\| \cdot \right\|_\sigma : \mathsf{HO} \to \mathsf{HO}$ is as in Fig. 16.

$$[\text{SRv}]$$
$$\frac{\overline{s} \notin \mathsf{dom}(\Delta) \qquad \Gamma; \Lambda'; \Delta' \vdash V \rhd U}{(\Gamma; \Lambda; \Delta, s :?(U); S) \xrightarrow{s?\langle V \rangle} (\Gamma; \Lambda, \Lambda'; \Delta, \Delta', s : S)}$$

$$[\text{ShRv}]$$
$$\frac{\Gamma; \emptyset; \emptyset \vdash a \rhd \langle U \rangle \qquad \Gamma; \Lambda'; \Delta' \vdash V \rhd U}{(\Gamma; \Lambda; \Delta) \xrightarrow{a?\langle V \rangle} (\Gamma; \Lambda, \Lambda'; \Delta, \Delta')}$$

$$[\text{SSnd}]$$
$$\frac{\begin{array}{cc} \Gamma, \Gamma'; \Lambda'; \Delta' \vdash V \rhd U \qquad \Gamma'; \emptyset; \Delta_j \vdash m_j \rhd U_j \qquad \overline{s} \notin \mathsf{dom}(\Delta) \\ \Delta' \backslash (\cup_j \Delta_j) \subseteq (\Delta, s : S) \qquad \Gamma'; \emptyset; \Delta'_j \vdash \overline{m}_j \rhd U'_j \qquad \Lambda' \subseteq \Lambda \end{array}}{(\Gamma; \Lambda; \Delta, s :!\langle U \rangle; S) \xrightarrow{(\nu \widetilde{m}) s!\langle V \rangle} (\Gamma, \Gamma'; \Lambda \backslash \Lambda'; (\Delta, s : S, \cup_j \Delta'_j) \backslash \Delta')}$$

$$[\text{ShSnd}]$$
$$\frac{\begin{array}{ccc} \Gamma, \Gamma'; \Lambda'; \Delta' \vdash V \rhd U & \Gamma'; \emptyset; \Delta_j \vdash m_j \rhd U_j & \Gamma; \emptyset; \emptyset \vdash a \rhd \langle U \rangle \\ \Delta' \backslash (\cup_j \Delta_j) \subseteq \Delta & \Gamma'; \emptyset; \Delta'_j \vdash \overline{m}_j \rhd U'_j & \Lambda' \subseteq \Lambda \end{array}}{(\Gamma; \Lambda; \Delta) \xrightarrow{(\nu \widetilde{m}) a!\langle V \rangle} (\Gamma, \Gamma'; \Lambda \backslash \Lambda'; (\Delta, \cup_j \Delta'_j) \backslash \Delta')}$$

$$[\text{Sel}]$$
$$\frac{\overline{s} \notin \mathsf{dom}(\Delta) \qquad j \in I}{(\Gamma; \Lambda; \Delta, s : \oplus\{l_i : S_i\}_{i \in I}) \xrightarrow{s \oplus l_j} (\Gamma; \Lambda; \Delta, s : S_j)}$$

$$[\text{Bra}]$$
$$\frac{\overline{s} \notin \mathsf{dom}(\Delta) \qquad j \in I}{(\Gamma; \Lambda; \Delta, s : \&\{l_i : T_i\}_{i \in I}) \xrightarrow{s \& l_j} (\Gamma; \Lambda; \Delta, s : S_j)}$$

$$[\text{Tau}]$$
$$\frac{\Delta_1 \longrightarrow \Delta_2 \vee \Delta_1 = \Delta_2}{(\Gamma; \Lambda; \Delta_1) \xrightarrow{\tau} (\Gamma; \Lambda; \Delta_2)}$$

Figure 15: Labelled Transition System for Typed Environments.

$$\llbracket w!\langle \lambda x. Q \rangle . P \rrbracket_\sigma \stackrel{\mathsf{def}}{=} u!\langle \lambda x. \llbracket Q \rrbracket_{\sigma, x} \rangle . \llbracket P \rrbracket_\sigma \qquad\qquad \llbracket w \rhd \{l_i : P_i\}_{i \in I} \rrbracket_\sigma \stackrel{\mathsf{def}}{=} u \rhd \{l_i : \llbracket P_i \rrbracket_\sigma\}_{i \in I}$$

$$\llbracket w?(x).P \rrbracket_\sigma \stackrel{\mathsf{def}}{=} u?(x). \llbracket P \rrbracket_\sigma \qquad\qquad \llbracket w \lhd l.P \rrbracket_\sigma \stackrel{\mathsf{def}}{=} u \lhd l. \llbracket P \rrbracket_\sigma$$

$$\llbracket (\nu n) P \rrbracket_\sigma \stackrel{\mathsf{def}}{=} (\nu n) \llbracket P \rrbracket_{\sigma, n} \qquad\qquad \llbracket (\lambda x. Q) w \rrbracket_\sigma \stackrel{\mathsf{def}}{=} (\lambda x. \llbracket Q \rrbracket_{\sigma, x}) u$$

$$\llbracket P \mid Q \rrbracket_\sigma \stackrel{\mathsf{def}}{=} \llbracket P \rrbracket_\sigma \mid \llbracket Q \rrbracket_\sigma \qquad\qquad \llbracket x \, w \rrbracket_\sigma \stackrel{\mathsf{def}}{=} x \, u$$

$$\llbracket \mathbf{0} \rrbracket_\sigma \stackrel{\mathsf{def}}{=} \mathbf{0}$$

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

Figure 16: Auxiliary mapping used to encode $\mathsf{HO}\pi$ into HO (Def. B.2).

C PROOFS

C.1 Auxiliary Results

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

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

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

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

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

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

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 $\mathcal{H}(\Gamma'), \Phi'; \mathcal{H}(\Delta), \Theta' \vdash \mathcal{A}_\epsilon^k(P)_g \triangleright \diamond$, where $k > 0; \widetilde{r} = \text{dom}(\Delta_\mu); \Phi' = \prod_{r \in \widetilde{r}} c^r : \langle\langle ?(\mathcal{R}'^\star(\Delta_\mu(r))); \text{end}\rangle\rangle;$ and $\text{balanced}(\Theta')$ with*

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

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

PROOF.

$$\Gamma; \Delta, \Delta_\mu \vdash P \triangleright \diamond \qquad\qquad \text{(Assumption)} \qquad (1)$$

$$\langle\!\langle\Gamma\rangle\!\rangle^1; \langle\!\langle\Delta\rangle\!\rangle^1, \langle\!\langle\Delta_\mu\rangle\!\rangle^1 \vdash \llbracket P \rrbracket_g^1 \triangleright \diamond \qquad\qquad \text{(Theorem 5.1 [10], (10))} \qquad (2)$$

$$\mathcal{G}(\langle\!\langle\Gamma\rangle\!\rangle^1), \Phi;\ \mathcal{G}(\langle\!\langle\Delta\rangle\!\rangle^1), \Theta \vdash \mathcal{B}_\epsilon^k(\llbracket P \rrbracket_g^1) \triangleright \diamond \qquad\qquad \text{(Theorem 3.27 [1], (11))} \qquad (3)$$

$$\llbracket \mathcal{G}(\langle\!\langle\Gamma\rangle\!\rangle^1) \rrbracket^2, \llbracket \Phi \rrbracket^2; \llbracket \mathcal{G}(\langle\!\langle\Delta\rangle\!\rangle^1) \rrbracket^2,\ \llbracket \Theta \rrbracket^2 \vdash \llbracket \mathcal{B}_\epsilon^k(\llbracket P \rrbracket_g^1) \rrbracket^2 \triangleright \diamond \qquad\qquad \text{(Theorem 5.2 [10], (12))} \qquad (4)$$

$$\mathcal{H}(\Gamma), \Phi'; \mathcal{H}(\Delta), \Theta' \vdash \mathcal{A}_\epsilon^k(P)_g \triangleright \diamond \qquad\qquad (\text{ Definition of } \mathcal{A}_\epsilon^k(\cdot)_g, \qquad (5)$$

$$\text{Definition 3.5, Definition 3.9, (4))}$$

$$\square$$

THEOREM 2 (MINIMALITY RESULT FOR π). *Let P be a closed π process, with $\widetilde{u} = \text{fn}(P)$ and $\widetilde{v} = \text{rn}(P)$. 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 $\sigma = \{\text{init}(\widetilde{u})/\widetilde{u}\}$.*

PROOF.

$$\Gamma; \Delta \vdash P \triangleright \diamond \qquad\qquad \text{(Assumption)} \qquad (6)$$

$$\Gamma\sigma; \Delta\sigma \vdash P\sigma \triangleright \diamond \qquad\qquad \text{(Lemma C.1, (6))} \qquad (7)$$

$$\mathcal{H}(\Gamma\sigma), \Phi'; \mathcal{H}(\Delta\sigma), \Theta' \vdash \mathcal{A}_\epsilon^k(P\sigma)_g \triangleright \diamond \qquad\qquad \text{(P is initialized, Theorem 1)} \qquad (8)$$

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 HOπ, derived by Arslanagić et al. [1].

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

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

$$\text{PolyResS} \frac{\text{Par} \frac{\text{Par} \frac{\text{Send} \frac{\text{Nil} \frac{}{\mathcal{H}(\Gamma\sigma), \Phi'; \mathcal{H}(\Delta\sigma) \vdash \mathbf{0} \triangleright \diamond}}{\mathcal{H}(\Gamma\sigma), \Phi'; \mathcal{H}(\Delta\sigma), \overline{c_k}! \langle \cdot \rangle; \text{end} \vdash \overline{c_k}! \langle \cdot \rangle} \qquad (9)}{\mathcal{H}(\Gamma\sigma), \Phi'; \mathcal{H}(\Delta\sigma), \mathcal{H}(\Delta_\mu\sigma), \overline{c_k}! \langle \cdot \rangle; \text{end} \vdash \overline{c_k}! \langle \cdot \rangle . \mathbf{0} \mid \prod_{r \in \widetilde{v}} P^r} \qquad (10)\qquad (8)}{\mathcal{H}(\Gamma\sigma), \Phi'; \mathcal{H}(\Delta\sigma), \mathcal{H}(\Delta_\mu\sigma), \overline{c_k}! \langle \cdot \rangle; \text{end}, \Theta' \vdash \overline{c_k}! \langle \cdot \rangle . \mathbf{0} \mid \prod_{r \in \widetilde{v}} P^r \mid \mathcal{A}_\epsilon^k(P\sigma)_g \triangleright \diamond}}{\mathcal{H}(\Gamma\sigma); \mathcal{H}(\Delta\sigma), \mathcal{H}(\Delta_\mu\sigma) \vdash (\nu \widetilde{c})(\nu \widetilde{c_r})\big(\prod_{r \in \widetilde{v}} P^r \mid \overline{c_k}! \langle \rangle . \mathbf{0} \mid \mathcal{A}_\epsilon^k(P\sigma)_g\big)}$$

$$\square$$

C.3 Typability of the Optimized Breakdown Function

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

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

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

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:

$$\text{RVar} \; \frac{}{\Gamma \cdot X : \Delta; \Delta \vdash X \triangleright \diamond} \tag{11}$$

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

$$\mathcal{G}(\Gamma \setminus x); \Theta \vdash \mathcal{B}_{\mathsf{rec} \; \tilde{y}}^k (X)_g \triangleright \diamond \tag{12}$$

By Tab. 3, we have:

$$\mathcal{B}_{\mathsf{rec} \; \tilde{y}}^k (X)_g = \mu X.c_k^r?(\widetilde{y}).c_X^r!\langle \widetilde{y}\rangle.X$$

$$\text{Req} \; \frac{\text{RVar} \; \dfrac{}{\mathcal{G}(\Gamma \setminus x), X : \Theta; \Theta \vdash X} \qquad \text{PolySend} \; \dfrac{}{\mathcal{G}(\Gamma \setminus x), X : \Theta; \widetilde{y} : \widetilde{M} \vdash \widetilde{y} \triangleright \widetilde{M}}}{\mathcal{G}(\Gamma \setminus x), X : \Theta; c_X^r : \langle \widetilde{M}\rangle, \widetilde{y} : \widetilde{M} \vdash c_X^r!\langle \widetilde{y}\rangle.X} \tag{13}$$

The following tree proves this case:

$$\text{Rec} \; \frac{\text{Rcv} \; \dfrac{(13) \qquad \text{PolySend} \; \dfrac{}{\mathcal{G}(\Gamma \setminus x), X : \Theta; \widetilde{y} : \widetilde{M} \vdash \widetilde{y} \triangleright \widetilde{M}}}{\mathcal{G}(\Gamma \setminus x), X : \Theta; \Theta \vdash c_k^r?(\widetilde{y}).c_X^r!\langle \widetilde{y}\rangle.X \triangleright \diamond}}{\mathcal{G}(\Gamma \setminus x); \Theta \vdash \mu X.c_k^r?(\widetilde{y}).c_X^r!\langle \widetilde{y}\rangle.X \triangleright \diamond} \tag{14}$$

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

(2) Case $P = u_i!\langle z_j\rangle.P'$. Let $u_i : C$. We distinguish three sub-cases: (i) $C = S = \mu\mathsf{t}.!\langle C_z\rangle; S'$, (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:

$$\text{Send} \; \frac{\Gamma \cdot X : \Delta_\mu; \Delta \cdot u_i : S' \vdash P' \triangleright \diamond \qquad \Gamma \cdot X : \Delta_\mu; \Delta_z \vdash z_j \triangleright C}{\Gamma \cdot X : \Delta_\mu; \Delta \cdot u_i : S \cdot \Delta_z \vdash u_i!\langle z_j\rangle.P' \triangleright \diamond} \tag{15}$$

Then, by IH on the right assumption of (15) we have:

$$\mathcal{G}(\Gamma \setminus \widetilde{x}'); \Theta' \vdash \mathcal{B}_{\mathsf{rec} \; \tilde{y}'}^{k+1} (P')_g \triangleright \diamond \tag{16}$$

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

$$\mathsf{dom}(\Theta_\mu') = \{c_{k+1}^r, c_{k+2}^r, \ldots, c_{k+|P'|+1}^r\} \cup \{\overline{c_{k+2}^r}, \ldots, \overline{c_{k+|P'|-1}^r}, \overline{c_{k+|P'|+1}^r}\}$$

and $\Theta_\mu'(c_{k+1}^r) = \mu\mathsf{t}.?(\widetilde{N}'); \mathsf{t}$ where $\widetilde{N}' = (\mathcal{G}(\Gamma), \mathcal{G}(\Delta_\mu \cdot \Delta \cdot u_i : S))(\widetilde{y}')$. By applying Lem. C.2 on the second assumption of (15) we have:

$$\mathcal{G}(\Gamma \cdot X : \Delta_\mu); \mathcal{G}(\Delta_z) \vdash \widetilde{z} \triangleright \mathcal{G}(C_z) \triangleright \diamond \tag{17}$$

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

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

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

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

$$\mathsf{dom}(\Theta_\mu) = \{c_k^r, c_{k+1}^r, \ldots, c_{k+|P|-1}^r\} \cup \{\overline{c_{k+1}^r}, \ldots, \overline{c_{k+|P|-1}^r}\}$$

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

$$\mathcal{B}_{\mathsf{rec} \; \tilde{y}}^k (P)_g = \mu X.c_k^r?(\widetilde{y}).u_l!\langle \widetilde{z}\rangle.c_{k+1}^r!\langle \widetilde{y}'\sigma_1\rangle.X \mid \mathcal{B}_{\mathsf{rec} \; \tilde{y}'\sigma_1}^{k+1} (P'\sigma)_g \tag{18}$$

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

$$\mathcal{G}(\Gamma \setminus \widetilde{x}); \Theta \vdash \mathcal{B}_{\mathsf{rec} \; \tilde{y}}^k (P)_g \triangleright \diamond \tag{19}$$

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

$$\text{RVar} \; \frac{}{\mathcal{G}(\Gamma \setminus \widetilde{x}) \cdot X : \Theta; \Theta \vdash X \triangleright \diamond} \tag{20}$$

$$\text{PolySend} \; \frac{\text{(20)} \qquad \text{PolyVar} \; \dfrac{}{\mathcal{G}(\Gamma \setminus \widetilde{x}) \cdot X : \Theta; \mathcal{G}(\Delta, u_i \sigma : S') \vdash \widetilde{y}' \sigma_1 \triangleright \widetilde{N}'}}{\mathcal{G}(\Gamma \setminus \widetilde{x}) \cdot X : \Theta; \Theta, \mathcal{G}(\Delta, u_i \sigma : S') \vdash \overline{c^r_{k+1}} ! \langle \widetilde{y}' \sigma_1 \rangle . X \triangleright \diamond} \tag{21}$$

Here, in typing the right-hand assumption, we may notice that in sub-case (i) $\mathcal{G}(u_i : S) = \widetilde{u}_i \, ! \langle \mathcal{G}(C_z) \rangle;$ end, $\mathcal{G}(S')$ and $\mathcal{G}(u_i \sigma : S') = \widetilde{u}_{i+1} : \mathcal{G}(S')$ where $\widetilde{u}_{i+1} = (u_{i+1}, \ldots, u_{i+|\mathcal{G}(S')|})$. So right-hand side follows by Def. 4.13 and Lem. C.1. Otherwise, in sub-case (ii) by Def. 4.4 and Def. 10 we know $\mathcal{G}(u_i : S) = \mathcal{G}(u_i \sigma : S')$ and by definition $\widetilde{u}_i \subseteq \widetilde{m} \subseteq \widetilde{y}' \sigma_1$.

$$\text{PolySend} \; \frac{\text{(21)} \qquad \text{(17)}}{\mathcal{G}(\Gamma \setminus \widetilde{x}) \cdot X : \Theta; \Theta, \mathcal{G}(\Delta_1) \vdash u_l ! \langle \widetilde{z} \rangle . \overline{c^r_{k+1}} ! \langle \widetilde{y}' \sigma_1 \rangle . X \triangleright \diamond} \tag{22}$$

$$\text{Rec} \; \frac{\text{PolyRcv} \; \dfrac{\text{(22)} \qquad \text{PolyVar} \; \dfrac{}{\mathcal{G}(\Gamma \setminus \widetilde{x}) \cdot X : \Theta; \mathcal{G}(\Delta_1) \vdash \widetilde{y} \triangleright \widetilde{N}}}{\mathcal{G}(\Gamma \setminus \widetilde{x}) \cdot X : \Theta; \Theta \vdash c^r_k ? (\widetilde{y}) . u_l ! \langle \widetilde{z} \rangle . \overline{c^r_{k+1}} ! \langle \widetilde{y}' \sigma_1 \rangle . X \triangleright \diamond}}{\mathcal{G}(\Gamma \setminus \widetilde{x}); \Theta \vdash \mu X . c^r_k ? (\widetilde{y}) . u_l ! \langle \widetilde{z} \rangle . \overline{c^r_{k+1}} ! \langle \widetilde{y}' \sigma_1 \rangle . X \triangleright \diamond} \tag{23}$$

We may notice that by Def. 4.4 and Def. 10 we have $\mathcal{G}(\Delta_1) = \mathcal{G}(\Delta), \widetilde{u}_i : \mathcal{G}(S), \mathcal{G}(\Delta_z)$ 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:

$$\text{Par} \; \frac{\text{(23)} \qquad \text{(16)}}{\mathcal{G}(\Gamma \setminus \widetilde{x}); \Theta \vdash \mu X . c^r_k ? (\widetilde{y}) . u_l ! \langle \widetilde{z} \rangle . c^r_{k+1} ! \langle \widetilde{y}' \sigma_1 \rangle . X \mid \mathcal{B}^{k+1}_{\text{rec } \widetilde{y}' \sigma_1} (P' \sigma)_g \triangleright \diamond} \tag{24}$$

Note that we have used the following for the right assumption of (24):

$$\mathcal{B}^{k+1}_{\text{rec } \widetilde{y}'} (P') \equiv_\alpha \mathcal{B}^{k+1}_{\text{rec } \widetilde{y}' \sigma_1} (P' \sigma)$$

(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:

$$\text{Rcv} \; \frac{\Gamma \cdot X : \Delta_\mu; \Delta, u_i : S', \Delta_z \vdash P' \triangleright \diamond \qquad \Gamma \cdot X : \Delta_\mu; \Delta_z \vdash z \triangleright C}{(\Gamma \setminus z) \cdot X : \Delta_\mu; \Delta, u_i : S \vdash u_i ? (z) . P' \triangleright \diamond} \tag{25}$$

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

$$\text{dom}(\Theta'_\mu) = \{c^r_{k+1}, c^r_{k+2}, \ldots, c^r_{k+|P'|+1}\} \cup \{\overline{c^r_{k+2}}, \ldots, \overline{c^r_{k+|P'|+1}}\}$$

and $\Theta'_\mu(c^r_{k+1}) = \mu t . ? (\widetilde{N}'); t$, where $\widetilde{N}' = (\mathcal{G}(\Gamma), \mathcal{G}(\Delta_\mu \cdot \Delta \cdot u_i : S))(\widetilde{y}')$. Then, by IH on the right assumption of (15) we have:

$$\mathcal{G}(\Gamma \setminus \widetilde{x}'); \Theta'_\mu \vdash \mathcal{B}^{k+1}_{\text{rec } \widetilde{y}'} (P') \triangleright \diamond \tag{26}$$

By applying Lem. C.2 to the second assumption of (15), we have:

$$\mathcal{G}(\Gamma) \cdot X : \Theta_\mu; \mathcal{G}(\Delta_z) \vdash \widetilde{z} \triangleright \mathcal{G}(C) \tag{27}$$

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

$$\Theta_\mu = \Theta'_\mu, c^r_k : \mu t . ? (\widetilde{N}); t, \overline{c^r_{k+1}} : \mu t . ! \langle \widetilde{N}' \rangle; t$$

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

$$\text{dom}(\Theta_\mu) = \{c^r_k, c^r_{k+1}, \ldots, c^r_{k+|P|-1}\} \cup \{\overline{c^r_{k+1}}, \ldots, \overline{c^r_{k+|P|-1}}\}$$

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

$$\mathcal{B}^k_{\text{rec } \widetilde{y}} (P)_g = \mu X . c^r_k ? (\widetilde{y}) . u_l ? (\widetilde{z}) . c^r_{k+1} ! \langle \widetilde{y}' \sigma_1 \rangle . X \mid \mathcal{B}^{k+1}_{\text{rec } \widetilde{y}} (P' \sigma)_g \tag{28}$$

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

$$\mathcal{G}(\Gamma_1 \setminus z); \Theta \vdash \mathcal{B}^k_{\text{rec } \widetilde{y}} (P)_g \triangleright \diamond \tag{29}$$

We use some auxiliary sub-trees:

$$\text{RVar} \frac{}{\mathcal{G}(\Gamma) \cdot X : \Theta; \Theta \vdash X \triangleright \diamond} \tag{30}$$

$$\text{PolySend} \frac{(30) \qquad \text{PolyVar} \frac{}{\mathcal{G}(\Gamma) \cdot X : \Theta; \mathcal{G}(\Delta, u_l : S', \Delta_z) \vdash \widetilde{y}'\sigma_1 \triangleright \widetilde{N}'}}{\mathcal{G}(\Gamma) \cdot X : \Theta; \Theta, \mathcal{G}(\Delta_1) \vdash c^r_{k+1}!\langle \widetilde{y}'\sigma_1 \rangle. X \triangleright \diamond} \tag{31}$$

Here, in typing the right-hand assumption, we may notice that in sub-case (i) $\mathcal{G}(u_i : S) = \widetilde{u}_i :?(\mathcal{G}(C_z))$; end, $\mathcal{G}(S')$ and $\mathcal{G}(u_i\sigma : S') = \widetilde{u}_{i+1} : \mathcal{G}(S')$ where $\widetilde{u}_{i+1} = (u_{i+1}, \ldots, u_{i+|\mathcal{G}(S')|})$ and by definition $\widetilde{u}_{i+1} \subseteq \widetilde{y}'\sigma_1$. So right-hand side follows by Def. 4.13 and Lem. C.1. Otherwise, in sub-case (ii) by Def. 4.4 and Def. 10 we know $\mathcal{G}(u_i : S) = \mathcal{G}(u_i\sigma : S')$.

$$\text{PolyRcv} \frac{(31) \qquad (27)}{\mathcal{G}(\Gamma \setminus z) \cdot X : \Theta; \Theta, \mathcal{G}(\Delta, u_i : S) \vdash u_l?(\widetilde{z}).c^r_{k+1}!\langle \widetilde{y}'\sigma_1 \rangle. X \triangleright \diamond} \tag{32}$$

$$\text{Rec} \frac{\text{PolyRcv} \frac{(32) \qquad \text{PolyVar} \frac{}{\mathcal{G}(\Gamma \setminus z) \cdot X : \Theta; \mathcal{G}(\Delta, r : S) \vdash \widetilde{y} \triangleright \widetilde{N}}}{\mathcal{G}(\Gamma_1 \setminus z) \cdot X : \Theta; \Theta \vdash c^r_k?(\widetilde{y}).u_l?(\widetilde{z}).c^r_{k+1}!\langle \widetilde{y}'\sigma_1 \rangle. X \triangleright \diamond}}{\mathcal{G}(\Gamma_1 \setminus z); \Theta \vdash \mu X. c^r_k?(\widetilde{y}).u_l?(\widetilde{z}).c^r_{k+1}!\langle \widetilde{y}'\sigma_1 \rangle. X \triangleright \diamond} \tag{33}$$

We may notice that by Def. 4.4 and Def. 10 we have $\mathcal{G}(\Delta_1) = \mathcal{G}(\Delta), \widetilde{u}_i : \mathcal{G}(S), \mathcal{G}(\Delta_z)$ 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:

$$\text{Par} \frac{(33) \qquad (26)}{\mathcal{G}(\Gamma_1 \setminus z); \Theta \vdash \mu X. c^r_k?(\widetilde{y}).u_l?(\widetilde{z}).c^r_{k+1}!\langle \widetilde{y}'\sigma_1 \rangle. X \mid \mathcal{B}^{k+1}_{\text{rec } \widetilde{y}'\sigma_1}(P'\sigma)_g} \tag{34}$$

Note that we have used the following for the right assumption of (34):

$$\mathcal{B}^{k+1}_{\text{rec } \widetilde{y}'}(P') \equiv_\alpha \mathcal{B}^{k+1}_{\text{rec } \widetilde{y}'\sigma_1}(P'\sigma)$$

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

$$\text{Par} \frac{\Gamma \cdot X : \Delta_\mu; \Delta_1 \vdash Q_1 \triangleright \diamond \qquad \Gamma \cdot X : \Delta_\mu; \Delta_2 \vdash Q_2 \triangleright \diamond}{\Gamma \cdot X : \Delta_\mu; \Delta_1, \Delta_2 \vdash Q_1 \mid Q_2 \triangleright \diamond} \tag{35}$$

Here we assume $X(Q_1)$. So, by IH on the first and second assumption of (35) we have:

$$\mathcal{G}(\Gamma \setminus x_1); \Theta_1 \vdash \mathcal{B}^k_{\text{rec } \widetilde{y}_1}(Q_1)_g \triangleright \diamond \tag{36}$$

$$\mathcal{G}(\Gamma \setminus x_1); \Theta_2 \vdash \mathcal{B}^{k+l+1}_{\text{rec } \widetilde{y}_2}(Q_2)_\emptyset \triangleright \diamond \tag{37}$$

where for $i \in \{1, 2\}$ we have $\widetilde{x}_i \subseteq \text{fn}(Q_i)$ such that $\Delta_i \setminus \widetilde{x} = \emptyset$, and $\widetilde{y}_i = \widetilde{v}_i \cdot \widetilde{m}_i$ where $\widetilde{v}_i$ is such that $\text{indexed}_{\Gamma, \Delta_i}(\widetilde{v}_1, \widetilde{x}_i)$ and $\widetilde{m}_i = \text{codom}(g_i)$. Further, $\Theta_i = \Theta^i_\mu \cdot \Theta_X(g_i)$ where

$$\text{dom}(\Theta^1_\mu) = \{c^r_{k+1}, c^r_{k+2}, \ldots, c^r_{k+|Q_1|}\} \cup \{\overline{c^r_{k+2}}, \ldots, \overline{c^r_{k+|Q_1|}}\}$$

$$\text{dom}(\Theta^2_\mu) = \{c^r_{k+l+1}, c^r_{k+l+2}, \ldots, c^r_{k+l+|Q_2|}\}$$

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

$$\mathcal{G}(\Gamma \setminus \widetilde{x}); \Theta \vdash \mathcal{B}^k_{\text{rec } \widetilde{y}}(P)_g \triangleright \diamond \tag{38}$$

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

$$\Theta' = c^r_k : \mu t.?(\widetilde{N}); t \cdot \overline{c^r_{k+1}} : \mu t.!\langle \widetilde{N}_1 \rangle; t$$

By Tab. 3 we have:

$$\mathcal{B}^k_{\text{rec } \widetilde{y}}(P)_g = \mu X. c^r_k?(\widetilde{y}).(\overline{c^r_{k+1}}!\langle \widetilde{y}_1 \rangle. X \mid c^r_{k+l+1}!\langle \widetilde{y}_2 \rangle) \mid \mathcal{B}^{k+1}_{\text{rec } \widetilde{y}_1}(Q_1)_g \mid \mathcal{B}^{k+l+1}_{\text{rec } \widetilde{y}_2}(Q_2)_\emptyset$$

We use some auxiliary sub-trees:

$$\text{Snd} \frac{\text{Nil} \dfrac{}{\mathcal{G}(\Gamma \setminus \widetilde{x}) \cdot X : \Theta' \vdash \mathbf{0} \rhd \diamond} \qquad \text{PolyVar} \dfrac{}{\mathcal{G}(\Gamma \setminus \widetilde{x}) \cdot X : \Theta'; \widetilde{y}_2 : \widetilde{N}_2 \vdash \widetilde{y}_2 \rhd \widetilde{N}_2}}{\mathcal{G}(\Gamma \setminus \widetilde{x}) \cdot X : \Theta'; \cdot \widetilde{y}_2 : \widetilde{N}_2 \vdash c^r_{k+l+1}!\langle \widetilde{y}_2 \rangle} \tag{39}$$

$$\text{Par} \frac{\text{Snd} \dfrac{\text{Rvar} \dfrac{}{\mathcal{G}(\Gamma \setminus \widetilde{x}) \cdot X : \Theta'; \Theta' \vdash X \rhd \diamond} \qquad \text{PolyVar} \dfrac{}{\mathcal{G}(\Gamma \setminus \widetilde{x}) \cdot X : \Theta'; \widetilde{y}_1 : \widetilde{N}_1 \vdash \widetilde{y}_1 \rhd \widetilde{N}}}{\mathcal{G}(\Gamma \setminus \widetilde{x}) \cdot X : \Theta'; \Theta' \cdot \widetilde{y}_1 : \widetilde{N}_1 \vdash \overline{c^r_{k+1}}!\langle \widetilde{y}_1 \rangle.X \rhd \diamond} \qquad (39)}{\mathcal{G}(\Gamma \setminus \widetilde{x}) \cdot X : \Theta'; \Theta' \cdot \widetilde{y} : \widetilde{N} \vdash \overline{c^r_{k+1}}!\langle \widetilde{y}_1 \rangle.X \mid \overline{c^r_{k+l+1}}!\langle \widetilde{y}_2 \rangle. \rhd \diamond} \tag{40}$$

$$\text{Rec} \frac{\text{Rv} \dfrac{(40) \qquad \text{PolyVar} \dfrac{}{\mathcal{G}(\Gamma \setminus \widetilde{x}) \cdot X : \Theta'; \widetilde{y} : \widetilde{N} \vdash \widetilde{y} \rhd \widetilde{N}}}{\mathcal{G}(\Gamma \setminus \widetilde{x}) \cdot X : \Theta'; \Theta' \vdash c^r_k?(\widetilde{y}).(\overline{c^r_{k+1}}!\langle \widetilde{y}_1 \rangle.X \mid c^r_{k+l+1}!\langle \widetilde{y}_2 \rangle) \rhd \diamond}}{\mathcal{G}(\Gamma \setminus \widetilde{x}); \Theta' \vdash \mu X.c^r_k?(\widetilde{y}).(\overline{c^r_{k+1}}!\langle \widetilde{y}_1 \rangle.X \mid c^r_{k+l+1}!\langle \widetilde{y}_2 \rangle) \rhd \diamond} \tag{41}$$

The following tree proves this case:

$$\text{Par} \frac{(41) \qquad \text{Par} \dfrac{(36) \qquad (37)}{\mathcal{G}(\Gamma \setminus \widetilde{x}); \Theta \vdash \mathcal{B}^{k+1}_{\text{rec } \widetilde{y}_1}(Q_1)_{g_1} \mid \mathcal{B}^{k+l+1}_{\text{rec } \widetilde{y}_2}(Q_2)_{g_2} \rhd \diamond}}{\mathcal{G}(\Gamma \setminus \widetilde{x}); \Theta_1 \cdot \Theta_2 \vdash \mu X.c^r_k?(\widetilde{y}).(\overline{c^r_{k+1}}!\langle \widetilde{y}_1 \rangle.X \mid c^r_{k+l+1}!\langle \widetilde{y}_2 \rangle) \mid \mathcal{B}^{k+1}_{\text{rec } \widetilde{y}_1}(Q_1)_g \mid \mathcal{B}^{k+l+1}_{\text{rec } \widetilde{y}_2}(Q_2)_{\emptyset} \rhd \diamond}$$

(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:

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

The only can that can be applied is RESS:

$$\text{ResS} \frac{\Gamma; \Delta_\mu \cdot \Delta \cdot s_1 : S \cdot \overline{s_1} : \overline{S} \vdash P'\{s_1 \overline{s_1}/s \overline{s}\} \rhd \diamond}{\Gamma; \Delta_\mu \cdot \Delta \vdash (\nu s_1 : S)P'\{s_1 \overline{s_1}/s \overline{s}\} \rhd \diamond} \tag{42}$$

By IH on the assumption of (42) we have:

$$\mathcal{G}(\Gamma \setminus \widetilde{x}); \Theta' \vdash \mathcal{B}^k_{\text{rec } \widetilde{y}'}\big(P'\{s_1 \overline{s_1}/s \overline{s}\}\big)_g \rhd \diamond \tag{43}$$

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

$$\text{dom}(\Theta'_\mu) = \{c^r_{k+1}, c^r_{k+2}, \ldots, c^r_{k+|P'|+1}\} \cup \{\overline{c^r_{k+2}}, \ldots, \overline{c^r_{k+|P'|+1}}\}$$

and $\Theta'_\mu(c^r_{k+1}) = \mu t.?(\widetilde{N}'); t$ where $\widetilde{N}' = (\mathcal{G}(\Gamma), \mathcal{G}(\Delta_\mu \cdot \Delta \cdot s_1 : S \cdot \overline{s_1} : \overline{S}))(\widetilde{y}')$.
We define $\widetilde{x} = \widetilde{x}' \setminus (s_1, \overline{s_1})$ and $\widetilde{y} = \widetilde{y}' \setminus (\widetilde{s} \cdot \widetilde{\overline{s}})$ where $\widetilde{s} = \text{bn}(s_1 : S)$ and $\widetilde{\overline{s}} = \text{bn}(\overline{s_1} : \overline{S})$. By construction $\widetilde{x} \subseteq \text{fn}(P)$ and $\Delta \setminus \widetilde{x} = \emptyset$.
Further, $\widetilde{y} = \widetilde{v} \cdot \widetilde{m}$ where $\text{indexed}_{\Gamma, \Delta}(\widetilde{v}, \widetilde{x})$.
Let $\Theta = \Theta', \Theta''_\mu$ where

$$\Theta''_\mu = c^r_k : \mu t.?(\widetilde{N}); t, \overline{c^r_{k+1}} : \mu t.!\langle \widetilde{N}' \rangle; t$$

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

$$\text{dom}(\Theta_\mu) = \{c^r_k, c^r_{k+1}, \ldots, c^r_{k+|P|-1}\} \cup \{\overline{c^r_{k+1}}, \ldots, \overline{c^r_{k+|P|-1}}\}$$

By Tab. 3 we have:

$$\mathcal{B}^k_{\text{rec } \widetilde{y}}(P)_g = \mu X.(\nu \widetilde{s} : \mathcal{G}(C))c_k?(\widetilde{y}).\overline{c_{k+1}}!\langle \widetilde{y}' \rangle.X \mid \mathcal{B}^k_{\text{rec } \widetilde{y}'}\big(P'\{s_1 \overline{s_1}/s \overline{s}\}\big)_g$$

We shall prove the following judgment:

$$\mathcal{G}(\Gamma \setminus \widetilde{x}); \Theta \vdash \mathcal{B}^k_{\text{rec } \widetilde{y}}(P)_g \rhd \diamond \tag{44}$$

We use some auxiliary sub-tree:

$$\text{PolySend} \frac{\text{Rvar} \dfrac{}{\mathcal{G}(\Gamma) \cdot X : \Theta_\mu; \Theta''_\mu \vdash X \rhd \diamond} \qquad \text{PolyVar} \dfrac{}{\mathcal{G}(\Gamma) \cdot X : \Theta''_\mu; \mathcal{G}(\Delta), \widetilde{s} : \mathcal{G}(S), \widetilde{\overline{s}} : \mathcal{G}(\overline{S}) \vdash \widetilde{y}' \rhd \widetilde{N}'}}{\mathcal{G}(\Gamma) \cdot X : \Theta_\mu; \Theta_\mu \cdot \mathcal{G}(\Delta), \widetilde{s} : \mathcal{G}(S), \widetilde{\overline{s}} : \mathcal{G}(\overline{S}) \vdash \overline{c_{k+1}}!\langle \widetilde{y}' \rangle.X \rhd \diamond} \tag{45}$$

The following tree proves this case:

$$
\begin{array}{c}
\text{PolyRcv} \cfrac{
  (45) \qquad
  \text{PolyVar} \cfrac{}{\mathcal{G}(\Gamma); \mathcal{G}(\Delta) \vdash \widetilde{y} \triangleright \widetilde{N}}
}{
  \text{PolyResS} \cfrac{
    \mathcal{G}(\Gamma \setminus \widetilde{x}) \cdot X : \Theta''_\mu; \Theta''_\mu \cdot \widetilde{s} : \mathcal{G}(S), \widetilde{\overline{s}} : \mathcal{G}(\overline{S}) \vdash c_k?(\widetilde{y}).\overline{c_{k+1}}!\langle \widetilde{y}' \rangle.X \triangleright \diamond
  }{
    \text{Rec} \cfrac{
      \mathcal{G}(\Gamma \setminus \widetilde{x}) \cdot X : \Theta''_\mu; \Theta''_\mu \vdash (\nu \widetilde{s} : \mathcal{G}(S)) c_k?(\widetilde{y}).\overline{c_{k+1}}!\langle \widetilde{y}' \rangle.X \triangleright \diamond
    }{
      \text{Par} \cfrac{
        \mathcal{G}(\Gamma \setminus \widetilde{x}); \Theta''_\mu \vdash \mu X.(\nu \widetilde{s} : \mathcal{G}(S)) c_k?(\widetilde{y}).\overline{c_{k+1}}!\langle \widetilde{y}' \rangle.X \triangleright \diamond \qquad (43)
      }{
        \mathcal{G}(\Gamma \setminus \widetilde{x}); \Theta \vdash \mu X.(\nu \widetilde{s} : \mathcal{G}(S)) c_k?(\widetilde{y}).\overline{c_{k+1}}!\langle \widetilde{y}' \rangle.X \mid \mathcal{B}^k_{\text{rec }\widetilde{y}'}\big(P'\{s_1\overline{s_1}/s\overline{s}\}\big)_g \triangleright \diamond
      }
    }
  }
\end{array}
$$

This concludes the proof.

$\square$

Theorem 3 (Typability of Breakdown). *Let P be an initialized process. If $\Gamma; \Delta \vdash P \triangleright \diamond$ then*

$$
\mathcal{G}(\Gamma \setminus \widetilde{x}); \mathcal{G}(\Delta \setminus \widetilde{x}), \Theta \vdash \mathcal{B}^k_{\widetilde{y}}(P) \triangleright \diamond \quad (k > 0)
$$

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

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

and $\Theta(c_k) = ?(\widetilde{M}); \mathsf{end}$, where $\widetilde{M} = (\mathcal{G}(\Gamma), \mathcal{G}(\Delta))(\widetilde{y})$.

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

$$
\mathcal{G}(\Gamma); \Theta \vdash \mathcal{B}^k_{\widetilde{y}}(\mathbf{0}) \triangleright \diamond \tag{46}
$$

where $\widetilde{x} \subseteq \mathsf{fn}(\mathbf{0}) = \epsilon$ and $\Theta = \{c_k : ?(\langle \mathsf{end} \rangle); \mathsf{end}\}$. Since by Rem. 1 we know that $c_k?().\mathbf{0}$ stands for $c_k?(y).\mathbf{0}$ with $c_k : ?(\langle \mathsf{end} \rangle); \mathsf{end}$. By Tab. 3: $\mathcal{B}^k_\epsilon(\mathbf{0}) = c_k?().\mathbf{0}$.
The following tree proves this case:

$$
\text{Rcv} \cfrac{
  \text{End} \cfrac{
    \text{Nil} \cfrac{}{\Gamma'; \emptyset; \emptyset \vdash \mathbf{0} \triangleright \diamond} \qquad c_k \notin dom(\Gamma)
  }{
    \Gamma'; \emptyset; c_k : \mathsf{end} \vdash \mathbf{0} \triangleright \diamond
  } \qquad
  \text{Sh} \cfrac{}{\Gamma'; \emptyset \vdash y \triangleright \langle \mathsf{end} \rangle}
}{
  \mathcal{G}(\Gamma), y : \langle \mathsf{end} \rangle; \Theta \vdash c_k?().\mathbf{0} \triangleright \diamond
}
$$

where $\Gamma' = \mathcal{G}(\Gamma), y : \langle \mathsf{end} \rangle$. We know $c_k \notin dom(\Gamma)$ 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) $u_i \in dom(\Delta)$ and (ii) $u_i \in dom(\Gamma)$. We consider sub-case (i) first. For this case Rule Rcv can be applied:

$$
\text{Rcv} \cfrac{
  \Gamma; \Delta, u_i : S, \Delta_z \vdash P' \triangleright \diamond \qquad \Gamma; \Delta_z \vdash z \triangleright C
}{
  \Gamma \setminus z; \Delta, u_i : ?(C); S \vdash u_i?(z).P' \triangleright \diamond
} \tag{47}
$$

By IH on the first assumption of (47) we know:

$$
\mathcal{G}(\Gamma'_1); \mathcal{G}(\Delta'_1), \Theta_1 \vdash \mathcal{B}^{k+1}_{\widetilde{y}'}(P') \triangleright \diamond \tag{48}
$$

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

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

and $\Theta_1(c_{k+1}) = ?(\widetilde{M}'); \mathsf{end}$ where $\widetilde{M}' = (\mathcal{G}(\Gamma), \mathcal{G}(\Delta, u_i : S, \Delta_z))(\widetilde{y}')$.
By applying Lem. C.2 to the second assumption of (47) we have:

$$
\mathcal{G}(\Gamma); \mathcal{G}(\Delta_z) \vdash \widetilde{z} \triangleright \mathcal{G}(C) \tag{49}
$$

Let $\widetilde{x} = \widetilde{x}', u \setminus z$ and $\widetilde{y} = \widetilde{y}'\sigma, u_i \setminus \widetilde{z}$ such that $|\widetilde{z}| = \mathcal{G}(C)$ where $\sigma = \{\widetilde{n}/\widetilde{u}\}$ with $\widetilde{n} = (u_{i+1}, \ldots, u_{i+|\mathcal{G}(S)|})$ and $\widetilde{u} = (u_i, \ldots, u_{i+|\mathcal{G}(S)|-1})$. We may notice that by Def. 4.13 $\mathsf{indexed}_{\Gamma, \Delta}(\widetilde{y}, \widetilde{x})$ holds. We define $\Theta = \Theta_1, \Theta'$, where

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

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

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

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

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

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:

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

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

$$\text{End} \frac{\text{Nil} \dfrac{}{\mathcal{G}(\Gamma); \emptyset \vdash \mathbf{0} \triangleright \diamond}}{\mathcal{G}(\Gamma); \overline{c_{k+1}} : \text{end} \vdash \mathbf{0} \triangleright \diamond} \tag{50}$$

$$\text{End} \frac{\text{PolySend} \dfrac{(50) \qquad \text{PolyVar} \dfrac{}{\mathcal{G}(\Gamma); \mathcal{G}(\Delta \setminus \Delta_1, u_{i+1} : S), \mathcal{G}(\Delta_z) \vdash \widetilde{y}'\sigma \triangleright \widetilde{M}'}}{\mathcal{G}(\Gamma); \overline{c_{k+1}} :!\langle\widetilde{M}'\rangle; \text{end}, \mathcal{G}(\Delta \setminus \Delta_1, u_{i+1} : S), \mathcal{G}(\Delta_z) \vdash}{\overline{c_{k+1}}!\langle\widetilde{y}'\sigma\rangle.\mathbf{0} \triangleright \diamond}}{\mathcal{G}(\Gamma); \overline{c_{k+1}} :!\langle\widetilde{M}'\rangle; \text{end}, u_i : \text{end}, \mathcal{G}(\Delta \setminus \Delta_1, u_{i+1} : S), \mathcal{G}(\Delta_z) \vdash}{\overline{c_{k+1}}!\langle\widetilde{y}'\sigma\rangle.\mathbf{0} \triangleright \diamond} \tag{51}$$

$$\text{End} \frac{\text{PolyRcv} \dfrac{(51) \qquad (49)}{\mathcal{G}(\Gamma \setminus z); u_i :?(\mathcal{G}(U)); \text{end}, \overline{c_{k+1}} :!\langle\widetilde{M}'\rangle; \text{end}, \mathcal{G}(\Delta_2) \vdash}{u_i?(\widetilde{z}).\overline{c_{k+1}}!\langle\widetilde{y}'\sigma\rangle.\mathbf{0} \triangleright \diamond}}{\mathcal{G}(\Gamma \setminus z); \overline{c_{k+1}} :!\langle\widetilde{M}'\rangle; \text{end}, c_k : \text{end}, \mathcal{G}(\Delta_2) \vdash}{u_i?(\widetilde{z}).\overline{c_{k+1}}!\langle\widetilde{y}'\sigma\rangle.\mathbf{0} \triangleright \diamond} \tag{52}$$

$$\text{PolyRcv} \frac{(52) \qquad \text{PolyVar} \dfrac{}{\mathcal{G}(\Gamma \setminus z); \mathcal{G}(\Delta_2) \vdash \widetilde{y} \triangleright \widetilde{M}}}{\mathcal{G}(\Gamma_1 \setminus z); \Theta' \vdash c_k?(\widetilde{y}).u_i?(z).\overline{c_{k+1}}!\langle\widetilde{y}'\sigma\rangle.\mathbf{0} \triangleright \diamond} \tag{53}$$

where $\Delta_2 = \Delta, u_i :?(C); S \setminus \Delta_1$. Using (53), the following tree proves this case:

$$\text{Par} \frac{(53) \qquad \dfrac{(48)}{\begin{array}{c}\mathcal{G}(\Gamma'_1 \setminus z); \mathcal{G}(\Delta_1), \Theta_1 \vdash \\ \mathcal{B}_{\widetilde{y}'\sigma}^{k+1}\big(P'\{u_{i+1}/u_i\}\big) \triangleright \diamond\end{array}}}{\begin{array}{c}\mathcal{G}(\Gamma_1 \setminus z); \mathcal{G}(\Delta_1), \Theta \vdash \\ c_k?(\widetilde{y}).u_i?(\widetilde{z}).\overline{c_{k+1}}!\langle\widetilde{y}'\sigma\rangle.\mathbf{0} \mid \mathcal{B}_{\widetilde{y}'\sigma}^{k+1}\big(P'\{u_{i+1}/u_i\}\big) \triangleright \diamond\end{array}} \tag{54}$$

Note that we have used the following for the right assumption of (54):

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

Next, we comment the case when $u_i \notin \widetilde{x}$. In this case $\Delta_1 = \Delta \setminus \widetilde{x}, u_i :?(C); S$. Hence, in the right hand-side of (54) instead of $\mathcal{G}(\Delta_1)$ we would have $\mathcal{G}(\Delta \setminus \widetilde{x}, u_{i+1} : S)$ 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 (48). We remark that similar provisos apply to the following cases when the assumption is $u_i \notin \widetilde{x}$.

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:

$$\text{Acc} \frac{\Gamma; \emptyset \vdash u_i \triangleright \langle C\rangle \qquad \Gamma; \Delta, z : C \vdash P' \triangleright \diamond \qquad \Gamma; z : C \vdash z \triangleright C}{\Gamma; \Delta \vdash u_i?(z).P' \triangleright \diamond} \tag{55}$$

By IH on the second assumption of (55) we have:

$$\mathcal{G}(\Gamma'_1); \mathcal{G}(\Delta'_1), \Theta_1 \vdash \mathcal{B}_{\widetilde{y}'}^{k+1}\big(P'\big) \triangleright \diamond \tag{56}$$

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

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

and $\Theta_1(c_{k+1}) =?(\widetilde{M}'); \text{end}$ where $\widetilde{M}' = (\mathcal{G}(\Gamma), \mathcal{G}(\Delta, z : C))(\widetilde{y}')$.
By applying Lem. C.2 to the first and third assumptions of (55) we have:

$$\mathcal{G}(\Gamma); \emptyset \vdash u_i \triangleright \langle\mathcal{G}(C)\rangle \tag{57}$$

$$\mathcal{G}(\Gamma); \mathcal{G}(z : C) \vdash \widetilde{z} \triangleright \mathcal{G}(C) \tag{58}$$

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

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

with $\widetilde{M} = (\mathcal{G}(\Gamma), \mathcal{G}(\Delta))(\widetilde{y})$. By Def. 4.5, $|P| = |P'| + 1$ so

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

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:

$$\mathcal{B}_{\widetilde{y}}^k(u_i?(z).P') = c_k?(\widetilde{y}).u_i?(z).\overline{c_{k+1}}!\langle \widetilde{y}' \rangle.\mathbf{0} \mid \mathcal{B}_{\widetilde{y}'}^{k+1}(P') \tag{59}$$

We shall prove the following judgment:

$$\mathcal{G}(\Gamma_1); \mathcal{G}(\Delta_1), \Theta \vdash \mathcal{B}_{\widetilde{y}}^k(u_i?(z).P') \triangleright \diamond \tag{60}$$

To this end, we use some auxiliary derivations:

$$\mathsf{End} \dfrac{\mathsf{Nil} \dfrac{}{\mathcal{G}(\Gamma); \emptyset; \emptyset \vdash \mathbf{0} \triangleright \diamond}}{\mathcal{G}(\Gamma); \emptyset; \overline{c_{k+1}} : \mathsf{end} \vdash \mathbf{0} \triangleright \diamond} \tag{61}$$

$$\mathsf{PolySend} \dfrac{(61) \qquad \mathsf{PolyVar} \dfrac{}{\mathcal{G}(\Gamma); \mathcal{G}(\Delta_2), z : \mathcal{G}(C) \vdash \widetilde{y}' \triangleright \widetilde{M}'}}{\mathcal{G}(\Gamma); \overline{c_{k+1}} :!\langle \widetilde{M}' \rangle; \mathsf{end}, \mathcal{G}(\Delta_2), z : \mathcal{G}(C) \vdash \overline{c_{k+1}}!\langle \widetilde{y}' \rangle.\mathbf{0} \triangleright \diamond} \tag{62}$$

$$\mathsf{End} \dfrac{\mathsf{PolyAcc} \dfrac{(57) \qquad (62) \qquad (58)}{\mathcal{G}(\Gamma); \overline{c_{k+1}} :!\langle \widetilde{M}' \rangle; \mathsf{end}, \mathcal{G}(\Delta_2) \vdash u_i?(z).\overline{c_{k+1}}!\langle \widetilde{y}' \rangle.\mathbf{0} \triangleright \diamond}}{\mathcal{G}(\Gamma); \overline{c_{k+1}} :!\langle \widetilde{M}' \rangle; \mathsf{end}, c_k : \mathsf{end}, \mathcal{G}(\Delta_2) \vdash u_i?(z).\overline{c_{k+1}}!\langle \widetilde{y}' \rangle.\mathbf{0} \triangleright \diamond} \tag{63}$$

$$\mathsf{PolyRcv} \dfrac{(63) \qquad \mathsf{PolyVar} \dfrac{}{\mathcal{G}(\Gamma); \mathcal{G}(\Delta_2) \vdash \widetilde{y} \triangleright \widetilde{M}}}{\mathcal{G}(\Gamma_1); \Theta' \vdash c_k?(\widetilde{y}).u_i?(z).\overline{c_{k+1}}!\langle \widetilde{y}' \rangle.\mathbf{0} \triangleright \diamond} \tag{64}$$

where $\Delta_2 = \Delta \setminus \Delta_1$. Using (56) and (64), the following tree proves this sub-case:

$$\mathsf{Par} \dfrac{(64) \qquad (56)}{\mathcal{G}(\Gamma_1); \mathcal{G}(\Delta_1), \Theta \vdash c_k?(\widetilde{y}).u_i?(z).\overline{c_{k+1}}!\langle \widetilde{y}' \rangle.\mathbf{0} \mid \mathcal{B}_{\widetilde{y}'}^{k+1}(P') \triangleright \diamond} \tag{65}$$

Note that if $u \notin \mathsf{fn}(P')$ we need to apply Lem. C.4 with u_i to the right assumption of (65) before applying (56). This concludes the analysis of the input case.

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

$$\mathsf{Send} \dfrac{\Gamma; \Delta_1, u_i : S \vdash P' \triangleright \diamond \qquad \Gamma; \Delta_z \vdash z_j \triangleright C \qquad u_i : S \in \Delta}{\Gamma; \Delta \vdash u_i!\langle z_j \rangle.P' \triangleright \diamond} \tag{66}$$

where $\Delta = \Delta_1, \Delta_z, u_i :!\langle C \rangle; S$.
By IH on the first assumption of (66) we have:

$$\mathcal{G}(\Gamma_1'); \mathcal{G}(\Delta_1'), \Theta_1 \vdash \mathcal{B}_{\widetilde{y}'}^{k+1}(P') \triangleright \diamond \tag{67}$$

where $\widetilde{x}' \subseteq \mathsf{fv}(P')$ and $\widetilde{y}'$ such that $\mathsf{indexed}_{\Gamma,\Delta_1,u_i:S}(\widetilde{y}', \widetilde{x}')$. Also, $\Gamma_1' = \Gamma \setminus \widetilde{y}'$, $\Delta_1' = \Delta_1 \setminus \widetilde{y}'$, and $\mathsf{balanced}(\Theta_1)$ with

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

and $\Theta_1(c_{k+1}) =?(\widetilde{M}_1); \mathsf{end}$ where $\widetilde{M}_1 = (\mathcal{G}(\Gamma), \mathcal{G}(\Delta_1, u_i : S))(\widetilde{y}')$.
By Lem. C.2 and the first assumption of (66) we have:

$$\mathcal{G}(\Gamma); \mathcal{G}(\Delta_z) \vdash \widetilde{z} \triangleright \mathcal{G}(C) \tag{68}$$

where $\widetilde{z} = (z_j, \ldots, z_{j+|\mathcal{G}(C)|-1})$. We assume $\widetilde{x} = \widetilde{x}', u, z$. Since $\widetilde{x}' \subseteq \mathsf{fn}(P')$ follows that $\widetilde{x} \subseteq \mathsf{fn}(P)$. Let $\widetilde{y} = \widetilde{y}'\sigma, u_i, \widetilde{z}$ where $\sigma = \{\widetilde{n}/\widetilde{u}\}$ with $\widetilde{n} = (u_{i+1}, \ldots, u_{i+|\mathcal{G}(S)|})$ and $\widetilde{u} = (u_i, \ldots, u_{i+|\mathcal{G}(S)|-1})$. We have $\widetilde{z} = \mathsf{fnb}(z, \widetilde{y})$. By Def. 4.13 follows that $\mathsf{indexed}_{\Gamma,\Delta}(\widetilde{y}, \widetilde{x})$. We define $\Theta = \Theta_1, \Theta'$, where:

$$\Theta' = c_k :?(\widetilde{M}); \mathsf{end}, \overline{c_{k+1}} :!\langle \widetilde{M}_1 \rangle; \mathsf{end}$$

with $\widetilde{M} = (\mathcal{G}(\Gamma), \mathcal{G}(\Delta))(\widetilde{y})$. By Def. 4.5, we know $|P| = |P'| + 1$, so

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

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:

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

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

We shall prove the following judgment:

$$\mathcal{G}(\Gamma_1); \mathcal{G}(\Delta_1'), \Theta \vdash \mathcal{B}_{\widetilde{y}}^{k}\big(u_i!\langle z\rangle.P'\big) \triangleright \diamond \tag{69}$$

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

$$\text{End} \frac{\text{Nil} \dfrac{}{\mathcal{G}(\Gamma); \emptyset \vdash \mathbf{0} \triangleright \diamond}}{\mathcal{G}(\Gamma); \overline{c_{k+1}} : \mathsf{end} \vdash \mathbf{0} \triangleright \diamond} \tag{70}$$

$$\text{End} \frac{\text{PolySend} \dfrac{(70) \qquad \text{PolyVar} \dfrac{}{\mathcal{G}(\Gamma); \mathcal{G}(\Delta_1, u_{i+1} : S \setminus \Delta_1') \vdash \widetilde{y}'\sigma \triangleright \widetilde{M}_1}}{\mathcal{G}(\Gamma); \overline{c_{k+1}} :!\langle \widetilde{M}_1\rangle; \mathsf{end}, \mathcal{G}(\Delta_1, u_{i+1} : S \setminus \Delta_1') \vdash \overline{c_{k+1}}!\langle \widetilde{y}'\sigma\rangle. \triangleright \diamond}}{\mathcal{G}(\Gamma); \overline{c_{k+1}} :!\langle \widetilde{M}_1\rangle; \mathsf{end}, u_i : \mathsf{end}, \mathcal{G}(\Delta_1, u_{i+1} : S \setminus \Delta_1') \vdash \overline{c_{k+1}}!\langle \widetilde{y}'\sigma\rangle.\mathbf{0} \triangleright \diamond} \tag{71}$$

$$\text{End} \frac{\text{Send} \dfrac{(71) \qquad (68)}{\mathcal{G}(\Gamma); \mathcal{G}(\Delta \setminus \Delta_1'), \overline{c_{k+1}} :!\langle \widetilde{M}_1\rangle; \mathsf{end} \vdash u_i!\langle \widetilde{z}\rangle.\overline{c_{k+1}}!\langle \widetilde{y}'\rangle.\mathbf{0} \triangleright \diamond}}{\mathcal{G}(\Gamma); \mathcal{G}(\Delta \setminus \Delta_1'), \overline{c_{k+1}} :!\langle \widetilde{M}_1\rangle; \mathsf{end}, c_k : \mathsf{end} \vdash u_i!\langle \widetilde{z}\rangle.\overline{c_{k+1}}!\langle \widetilde{y}'\rangle.\mathbf{0} \triangleright \diamond} \tag{72}$$

$$\text{PolyRcv} \frac{(72) \qquad \text{PolyVar} \dfrac{}{\mathcal{G}(\Gamma); \mathcal{G}(\Delta \setminus \Delta_1') \vdash \widetilde{y} : \widetilde{M}}}{\mathcal{G}(\Gamma_1); \Theta' \vdash c_k?(\widetilde{y}).u_i!\langle \widetilde{z}\rangle.\overline{c_{k+1}}!\langle \widetilde{y}'\rangle.\mathbf{0} \triangleright \diamond} \tag{73}$$

Using (67) and (73), the following tree proves this case:

$$\text{Par} \frac{(73) \qquad \dfrac{(67)}{\mathcal{G}(\Gamma_1); \mathcal{G}(\Delta_1'), \Theta_1 \vdash \mathcal{B}_{\widetilde{y}'\sigma}^{k+1}\big(P'\{u_{i+1}/u_i\}\big) \triangleright \diamond}}{\mathcal{G}(\Gamma_1); \mathcal{G}(\Delta_1'), \Theta \vdash \mathcal{B}_{\widetilde{y}}^{k}\big(u_i!\langle z\rangle.P'\big) \triangleright \diamond} \tag{74}$$

Note that we have used the following for the right assumption of (54):

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

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

$$\text{Req} \frac{\Gamma; \emptyset \vdash u \triangleright \langle C\rangle \qquad \Gamma; \Delta_1 \triangleright P' \triangleright \diamond \qquad \Gamma; \Delta_z \vdash z \triangleright C}{\Gamma; \Delta_1, \Delta_z \vdash u_i!\langle z\rangle.P' \triangleright \diamond} \tag{75}$$

Let $\widetilde{x}' \subseteq \mathsf{fn}(P')$ and $\widetilde{y}$ such that $\mathsf{indexed}_{\Gamma, \Delta_1}(\widetilde{y}', \widetilde{x}')$. Further, let $\Gamma_1' = \Gamma \setminus \widetilde{x}'$ and $\Delta_1' = \Delta_1 \setminus \widetilde{x}'$. Also, let Θ_1 be environment defined as in sub-case (i).

By IH on the second assumption of (75) we have:

$$\mathcal{G}(\Gamma_1'); \mathcal{G}(\Delta_1'), \Theta_1 \vdash \mathcal{B}_{\widetilde{y}'}^{k+1}\big(P'\big) \triangleright \diamond \tag{76}$$

By Lem. C.2 and the first and third assumptions of (75) we have:

$$\mathcal{G}(\Gamma); \emptyset \vdash u_i \triangleright \mathcal{G}(\langle C\rangle) \tag{77}$$

$$\mathcal{G}(\Gamma); \mathcal{G}(\Delta_z) \vdash \widetilde{z} \triangleright \mathcal{G}(C) \tag{78}$$

We define $\widetilde{x} = \widetilde{x}' \cup z \cup u$ and $\widetilde{y} = \widetilde{y}' \cup \widetilde{z} \cup u_i$ where $|\widetilde{z}| = |\mathcal{G}(C)|$. Notice that $\mathsf{indexed}_{\Gamma,\Delta}(\widetilde{y}, \widetilde{x})$. Let $\Gamma_1 = \Gamma \setminus \widetilde{x} = \Gamma_1' \setminus u$ and $\Delta \setminus \widetilde{x} = \Delta_1'$. By Tab. 3, we have:

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

We shall prove the following judgment:

$$\mathcal{G}(\Gamma_1); \mathcal{G}(\Delta_1'), \Theta \vdash \mathcal{B}_{\widetilde{y}}^k\big(u_i!\langle z\rangle.P'\big)$$

We use some auxiliary derivations:

$$\text{PolySend} \cfrac{\text{End} \cfrac{\text{Nil} \cfrac{}{\mathcal{G}(\Gamma); \emptyset; \emptyset \vdash \mathbf{0} \triangleright \diamond}}{\mathcal{G}(\Gamma); \emptyset; \overline{c_{k+1}} : \mathsf{end} \vdash \mathbf{0} \triangleright \diamond} \qquad \text{PolyVar} \cfrac{}{\mathcal{G}(\Gamma); \mathcal{G}(\Delta_1 \setminus \Delta_1') \vdash \widetilde{y}' : \widetilde{M_1}}}{\mathcal{G}(\Gamma); \mathcal{G}(\Delta_1 \setminus \Delta_1'), \overline{c_{k+1}} :!\langle\widetilde{M_1}\rangle; \mathsf{end} \vdash \overline{c_{k+1}}!\langle\widetilde{y}'\rangle.\mathbf{0} \triangleright \diamond} \tag{79}$$

$$\text{PolyReq} \cfrac{(77) \qquad (79) \qquad (78)}{\mathcal{G}(\Gamma); \mathcal{G}(\Delta_1 \setminus \Delta_1'), \overline{c_{k+1}} :!\langle\widetilde{M_1}\rangle; \mathsf{end}, \widetilde{z} : \mathcal{G}(C) \vdash u_i!\langle\widetilde{z}\rangle.\overline{c_{k+1}}!\langle\widetilde{y}'\rangle.\mathbf{0} \triangleright \diamond} \tag{80}$$

$$\text{PolyRcv} \cfrac{(80) \qquad \text{PolyVar} \cfrac{}{\mathcal{G}(\Gamma); \Delta_1 \setminus \Delta_1', \widetilde{z} : \mathcal{G}(C) \vdash \widetilde{y} \triangleright \widetilde{M}}}{\mathcal{G}(\Gamma_1); \Theta' \vdash c_k?(\widetilde{y}).u_i!\langle\widetilde{z}\rangle.\overline{c_{k+1}}!\langle\widetilde{y}'\rangle.\mathbf{0} \triangleright \diamond} \tag{81}$$

The following tree proves this case:

$$\text{Par} \cfrac{(81) \qquad \cfrac{(76)}{\mathcal{G}(\Gamma_1); \mathcal{G}(\Delta_1'), \Theta_1 \vdash \mathcal{B}_{\widetilde{y}'}^{k+1}(P') \triangleright \diamond}}{\mathcal{G}(\Gamma_1); \mathcal{G}(\Delta_1'), \Theta \vdash \mathcal{B}_{\widetilde{y}}^k\big(u_i!\langle z\rangle.P'\big) \triangleright \diamond} \tag{82}$$

We remark that if $u_i \notin \mathsf{fn}(P')$ we need to apply Lem. C.4 with u_i to the right assumption of (82) before applying (76). This concludes the analysis for the output case $P = u_i!\langle z\rangle.P'$.

(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:

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

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

$$\text{ResS} \cfrac{\Gamma; \Delta, s_1 : S, \overline{s_1} : \overline{S} \vdash P'\{s_1\overline{s_1}/s\overline{s}\} \triangleright \diamond}{\Gamma; \Delta \vdash (\nu s_1 : S)P'\{s_1\overline{s_1}/s\overline{s}\} \triangleright \diamond} \tag{83}$$

By IH on the assumption of (83) we have:

$$\mathcal{G}(\Gamma \setminus \widetilde{x}); \mathcal{G}(\Delta \setminus \widetilde{x}, s_1 : S, \overline{s_1} : \overline{S}), \Theta_1 \vdash \mathcal{B}_{\widetilde{y}}^k\big(P'\{s_1\overline{s_1}/s\overline{s}\}\big) \triangleright \diamond \tag{84}$$

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

$$\mathsf{dom}(\Theta_1) = \{c_k, \ldots, c_{k+|P'|-1}\} \cup \{\overline{c_{k+1}}, \ldots, \overline{c_{k+|P'|-1}}\}$$

and $\Theta_1(c_k) =?(\widetilde{M}); \mathsf{end}$ with $\widetilde{M} = (\mathcal{G}(\Gamma), \mathcal{G}(\Delta))(\widetilde{y})$.
Note that we take $\Theta = \Theta_1$ since $|P| = |P'|$. By Def. 4.1 and Def. 4.4 and (84), we know that:

$$\mathcal{G}(\Gamma \setminus \widetilde{x}); \mathcal{G}(\Delta \setminus \widetilde{x}), \widetilde{s} : \mathcal{G}(S), \overline{\widetilde{s}} : \mathcal{G}(\overline{S}) \vdash \mathcal{B}_{\widetilde{y}}^k\big(P'\{s_1\overline{s_1}/s\overline{s}\}\big) \triangleright \diamond \tag{85}$$

where $\widetilde{s} = (s_1, \ldots, s_{|\mathcal{G}(S)|})$ and $\overline{\widetilde{s}} = (\overline{s_1}, \ldots, \overline{s_{|\mathcal{G}(S)|}})$. By Tab. 3, we have:

$$\mathcal{B}_{\widetilde{y}}^k\big((\nu s)P'\big) = (\nu \widetilde{s} : \mathcal{G}(S))\mathcal{B}_{\widetilde{y}}^k\big(P'\{s_1\overline{s_1}/s\overline{s}\}\big)$$

The following tree proves this sub-case:

$$\text{PolyResS} \cfrac{(85)}{\mathcal{G}(\Gamma \setminus \widetilde{x}); \mathcal{G}(\Delta \setminus \widetilde{x}) \vdash (\nu \widetilde{s} : \mathcal{G}(S))\mathcal{B}_{\widetilde{y}}^k\big(P'\{s_1\overline{s_1}/s\overline{s}\}\big) \triangleright \diamond} \tag{86}$$

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:

$$\text{Res} \cfrac{\Gamma, s_1 : \langle C\rangle; \Delta \vdash P'\{s_1/s\} \triangleright \diamond}{\Gamma; \Delta \vdash (\nu s_1)P'\{s_1/s\} \triangleright \diamond} \tag{87}$$

By IH on the first assumption of (87) we have:

$$\mathcal{G}(\Gamma \setminus \widetilde{x}, s_1 : \langle C \rangle); \mathcal{G}(\Delta \setminus \widetilde{x}), \Theta_1 \vdash \mathcal{B}_{\widetilde{y}}^k \big(P'\{s_1/s\}\big) \triangleright \diamond \tag{88}$$

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

$$\mathsf{dom}(\Theta_1) = \{c_k, \ldots, c_{k+|P'|-1}\} \cup \{\overline{c_{k+1}}, \ldots, \overline{c_{k+|P'|-1}}\}$$

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

Here we also take $\Theta = \Theta_1$ since $|P| = |P'|$. We notice that by Def. 4.1 and Def. 4.4 and (88):

$$\mathcal{G}(\Gamma \setminus \widetilde{x}), s_1 : \mathcal{G}(\langle C \rangle); \mathcal{G}(\Delta \setminus \widetilde{x}), \Theta_1 \vdash \mathcal{B}_{\widetilde{y}}^k \big(P'\{s_1/s\}\big) \triangleright \diamond \tag{89}$$

By Tab. 3, we have:

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

The following tree proves this sub-case:

$$\mathsf{PolyRes} \; \frac{(89)}{\mathcal{G}(\Gamma \setminus \widetilde{x}); \mathcal{G}(\Delta \setminus \widetilde{x}), \Theta \vdash (\nu s_1 : \mathcal{G}(\langle C \rangle)) \mathcal{B}_{\widetilde{y}}^k \big(P'\{s_1/s\}\big) \triangleright \diamond} \tag{90}$$

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

$$\mathsf{Par} \; \frac{\Gamma; \Delta_1 \vdash Q \triangleright \diamond \qquad \Gamma; \Delta_2 \vdash R \triangleright \diamond}{\Gamma; \Delta_1, \Delta_2 \vdash Q \mid R \triangleright \diamond} \tag{91}$$

By IH on the first assumption of (91) we have:

$$\mathcal{G}(\Gamma_1'); \mathcal{G}(\Delta_1'), \Theta_1 \vdash \mathcal{B}_{\widetilde{y}_1}^{k+1}(Q) \triangleright \diamond \tag{92}$$

where $\widetilde{x}_1 \subseteq \mathsf{fn}(Q)$ and $\widetilde{y}_1$ such that $\mathsf{indexed}_{\Gamma,\Delta_1}(\widetilde{y}_1, \widetilde{x}_1)$. Also, $\Gamma_1' = \Gamma \setminus \widetilde{x}_1$, $\Delta_1' = \Delta_1 \setminus \widetilde{x}_1$, and $\mathsf{balanced}(\Theta_1)$ with

$$\mathsf{dom}(\Theta_1) = \{c_{k+1}, \ldots, c_{k+|Q|}\} \cup \{\overline{c_{k+2}}, \ldots, \overline{c_{k+|Q|}}\}$$

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

By IH on the second assumption of (91) we have:

$$\mathcal{G}(\Gamma_2'); \mathcal{G}(\Delta_2'), \Theta_2 \vdash \mathcal{B}_{\widetilde{y}_2}^{k+l+1}(R) \triangleright \diamond \tag{93}$$

where $\widetilde{x}_2 \subseteq \mathsf{fn}(R)$ and $\widetilde{y}_2$ such that $\mathsf{indexed}_{\Gamma,\Delta_2}(\widetilde{y}_2, \widetilde{x}_2)$ and $l = |Q|$. Also, $\Gamma_2' = \Gamma \setminus \widetilde{x}_2$, $\Delta_2' = \Delta_2 \setminus \widetilde{x}_2$ and $\mathsf{balanced}(\Theta_2)$ with

$$\mathsf{dom}(\Theta_2) = \{c_{k+l+1}, \ldots, c_{k+l+|R|}\} \cup \{\overline{c_{k+l+2}}, \ldots, \overline{c_{k+l+|R|}}\}$$

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

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

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

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:

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

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

$$\mathcal{G}(\Gamma_1); \mathcal{G}(\Delta_1', \Delta_2'), \Theta \vdash c_k?(\widetilde{y}).\overline{c_{k+1}}!\langle \widetilde{y}_1 \rangle.\overline{c_{k+l+1}}!\langle \widetilde{y}_2 \rangle.\mathbf{0} \mid \mathcal{B}_{\widetilde{y}_1}^{k+1}(Q) \mid \mathcal{B}_{\widetilde{y}_2}^{k+l+1}(R) \triangleright \diamond$$

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

$$\mathsf{End} \; \frac{\mathsf{PolySend} \; \frac{\mathsf{End} \; \frac{\mathsf{Nil} \; \frac{}{\mathcal{G}(\Gamma); \emptyset \vdash \mathbf{0}}}{\mathcal{G}(\Gamma); \overline{c_{k+l+1}} : \mathsf{end} \vdash \mathbf{0}} \qquad \mathsf{PolyVar} \; \frac{}{\mathcal{G}(\Gamma); \mathcal{G}(\Delta_2), \vdash \widetilde{z} \triangleright \widetilde{M}_2}}{\mathcal{G}(\Gamma); \mathcal{G}(\Delta_2'), \overline{c_{k+l+1}} :!\langle \widetilde{M}_2 \rangle; \mathsf{end} \vdash \overline{c_{k+l+1}}!\langle \widetilde{z} \rangle.\mathbf{0} \triangleright \diamond}}{\mathcal{G}(\Gamma); \mathcal{G}(\Delta_2'), \overline{c_{k+l+1}} :!\langle \widetilde{M}_2 \rangle; \mathsf{end}, \overline{c_{k+1}} : \mathsf{end} \vdash \overline{c_{k+l+1}}!\langle \widetilde{z} \rangle.\mathbf{0} \triangleright \diamond} \tag{94}$$

$$\text{PolySend} \frac{(94) \qquad \text{PolyVar} \dfrac{}{\mathcal{G}(\Gamma); \mathcal{G}(\Delta_1) \vdash \widetilde{y}_1 \rhd \widetilde{M}_1}}{\mathcal{G}(\Gamma); \mathcal{G}(\Delta_1', \Delta_2'), \overline{c_{k+1}} :! \langle \widetilde{M}_1 \rangle; \text{end}, \overline{c_{k+l+1}} :! \langle \widetilde{M}_2 \rangle; \text{end} \vdash}{\overline{c_{k+1}}! \langle \widetilde{y}_1 \rangle . \overline{c_{k+l+1}}! \langle \widetilde{y}_2 \rangle . \mathbf{0} \rhd \diamond}$$

$$\text{End} \frac{}{\mathcal{G}(\Gamma); \mathcal{G}(\Delta_1', \Delta_2'), \overline{c_{k+1}} :! \langle \widetilde{M}_1 \rangle; \text{end}, \overline{c_{k+l+1}} :! \langle \widetilde{M}_2 \rangle; \text{end}, c_k : \text{end} \vdash}{\overline{c_{k+1}}! \langle \widetilde{y}_1 \rangle . \overline{c_{k+l+1}}! \langle \widetilde{y}_2 \rangle . \mathbf{0} \rhd \diamond} \tag{95}$$

$$\text{PolyRcv} \frac{(95) \qquad \text{PolyVar} \dfrac{}{\mathcal{G}(\Gamma); \mathcal{G}(\Delta_1', \Delta_2') \vdash \widetilde{x} \rhd \widetilde{M}}}{\mathcal{G}(\Gamma_1); \Theta' \vdash c_k?(\widetilde{y}).\overline{c_{k+1}}! \langle \widetilde{y}_1 \rangle . \overline{c_{k+l+1}}! \langle \widetilde{y}_2 \rangle . \mathbf{0} \rhd \diamond} \tag{96}$$

$$\text{(Lem. C.3) with } \Gamma_1' \setminus \Gamma_1 \frac{(92)}{\mathcal{G}(\Gamma_1); \mathcal{G}(\Delta_1), \Theta_1 \vdash \mathcal{B}_{\widetilde{y}_1}^{k+1}(Q) \rhd \diamond} \tag{97}$$

$$\text{(Lem. C.3) with } \Gamma_2' \setminus \Gamma_1 \frac{(93)}{\mathcal{G}(\Gamma_1); \mathcal{G}(\Delta_2'), \Theta_2 \vdash \mathcal{B}_{\widetilde{y}_2}^{k+l+1}(R) \rhd \diamond} \tag{98}$$

$$\text{Par} \frac{(97) \qquad (98)}{\mathcal{G}(\Gamma_1); \mathcal{G}(\Delta_1', \Delta_2'), \Theta_1, \Theta_2 \vdash \mathcal{B}_{\widetilde{y}}^{k+1}(Q) \mid \mathcal{B}_{\widetilde{z}}^{k+l+1}(R) \rhd \diamond} \tag{99}$$

The following tree proves this case:

$$\text{Par} \frac{(96) \qquad (99)}{\mathcal{G}(\Gamma_1); \mathcal{G}(\Delta_1', \Delta_2'), \Theta \vdash}{c_k?(\widetilde{y}).\overline{c_{k+1}}! \langle \widetilde{y}_1 \rangle . \overline{c_{k+l+1}}! \langle \widetilde{y}_2 \rangle . \mathbf{0} \mid \mathcal{B}_{\widetilde{y}_1}^{k+1}(Q) \mid \mathcal{B}_{\widetilde{y}_2}^{k+l+1}(R) \rhd \diamond}$$

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

$$\frac{\Gamma, X : \Delta; \Delta \vdash P' \rhd \diamond}{\Gamma; \Delta \vdash \mu X.P'} \tag{100}$$

We remark that by (100) we know $x : S \in \Delta \implies \text{tr}(S)$. Then, by applying Lem. C.6 on the assumption of (100) we have:

$$\mathcal{G}(\Gamma \setminus \widetilde{x}); \Theta \vdash \mathcal{B}_{\text{rec } \widetilde{y}'}^{k+1}(P')_g \rhd \diamond \tag{101}$$

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

$$\text{dom}(\Theta'_\mu) = \{c_{k+1}^r, \ldots, c_{k+|P'|}^r\} \cup \{\overline{c_{k+1}^r}, \ldots, \overline{c_{k+|P'|}^r}\}$$

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

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

$$\mathcal{B}_{\widetilde{y}}^k(P) = (\nu c_X^r)(c_k?(\widetilde{y}).\overline{c_{k+1}^r}! \langle \widetilde{y}' \rangle . \mu X. c_X^r?(\widetilde{z}).\overline{c_{k+1}^r}! \langle \widetilde{z} \rangle . X \mid \mathcal{B}_{\text{rec } \widetilde{y}'}^{k+1}(P')_g)$$

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

$$\Theta''_\mu = \overline{c_k^r} : \mu t.?(\widetilde{M}); t \cdot \overline{c_{k+1}^r} : \mu t.! \langle \widetilde{M}' \rangle; t$$

We shall prove the following judgment:

$$\mathcal{G}(\Gamma \setminus \widetilde{x}); \mathcal{G}(\Delta \setminus \widetilde{x}) \cdot \Delta_\mu \vdash \mathcal{B}_{\widetilde{y}}^k(P) \rhd \diamond \tag{102}$$

Let $\Theta'_X = c_X^r : \mu t.?(\widetilde{M}'); t \cdot \overline{c_X^r} : \mu t.! \langle \widetilde{M}' \rangle; t$. We use some auxiliary sub-trees:

$$\text{Send} \frac{\text{RVar} \dfrac{}{\mathcal{G}(\Gamma \setminus \widetilde{x}), X : \Theta''; \Theta'' \vdash X \rhd \diamond} \qquad \text{PolyVar} \dfrac{}{\mathcal{G}(\Gamma \setminus \widetilde{x}), X : \Theta''; \mathcal{G}(\Delta) \vdash \widetilde{z} \rhd \widetilde{M}}}{\mathcal{G}(\Gamma \setminus \widetilde{x}), X : \Theta''; \mathcal{G}(\Delta), \Theta'' \vdash \overline{c_{k+1}^r}! \langle \widetilde{z} \rangle . X \rhd \diamond} \tag{103}$$

$$\text{PolyRcv} \cfrac{(103) \qquad \text{PolyVar} \cfrac{}{\mathcal{G}(\Gamma \setminus \widetilde{x}); \mathcal{G}(\Delta) \vdash \widetilde{z} \rhd \widetilde{M}}}{\text{Rec} \cfrac{\mathcal{G}(\Gamma \setminus \widetilde{x}), X : \Theta''; \Theta'' \vdash c_X^r?(\widetilde{z}).\overline{c_{k+1}^r}!\langle\widetilde{z}\rangle.X \rhd \diamond}{\mathcal{G}(\Gamma \setminus \widetilde{x}); \Theta'' \vdash \mu X.c_X^r?(\widetilde{z}).\overline{c_{k+1}^r}!\langle\widetilde{z}\rangle.X \rhd \diamond}} \tag{104}$$

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

$$\text{PolySend} \cfrac{(104) \qquad \text{PolyVar} \cfrac{}{\mathcal{G}(\Gamma \setminus \widetilde{x}); \mathcal{G}(\Delta) \vdash \widetilde{y}' \rhd \widetilde{M}'}}{\mathcal{G}(\Gamma \setminus \widetilde{x}); \mathcal{G}(\Delta), \overline{c_{k+1}^r} : \mu t.!\langle\widetilde{M}\rangle; t, c_X^r : \mu t.?(\widetilde{M}); t \vdash \overline{c_{k+1}^r}!\langle\widetilde{y}\rangle.\mu X.c_X^r?(\widetilde{z}).\overline{c_{k+1}^r}!\langle\widetilde{z}\rangle.X \rhd \diamond} \tag{105}$$

$$\text{PolyRcv} \cfrac{(105) \qquad \text{PolyVar} \cfrac{}{\mathcal{G}(\Gamma \setminus \widetilde{x}); \widetilde{y} : \widetilde{M} \vdash \widetilde{y} \rhd \widetilde{M}}}{\mathcal{G}(\Gamma \setminus \widetilde{x}); \mathcal{G}(\Delta \setminus \widetilde{x}) \cdot \Theta''_\mu \cdot c_X^r : \mu t.?(\widetilde{M}); t \vdash c_k?(\widetilde{y}).\overline{c_{k+1}^r}!\langle\widetilde{y}'\rangle.\mu X.c_X^r?(\widetilde{z}).\overline{c_{k+1}^r}!\langle\widetilde{z}\rangle.X \rhd \diamond} \tag{106}$$

The following tree proves this case:

$$\text{ResS} \cfrac{\text{Par} \cfrac{(106) \qquad (101)}{\mathcal{G}(\Gamma \setminus \widetilde{x}); \mathcal{G}(\Delta \setminus \widetilde{x}) \cdot \Theta_\mu \cdot \Theta'_X \vdash c_k?(\widetilde{y}).\overline{c_{k+1}^r}!\langle\widetilde{y}'\rangle.X\mu X.c_X^r?(\widetilde{z}).\overline{c_{k+1}^r}!\langle\widetilde{z}\rangle.X \mid \mathcal{B}_{\text{rec } \widetilde{y}}^{k+1}(P') \rhd \diamond}}{\mathcal{G}(\Gamma \setminus \widetilde{x}); \mathcal{G}(\Delta \setminus \widetilde{x}) \cdot \Theta_\mu \vdash (\nu\, c_X^r)(c_k?(\widetilde{y}).\overline{c_{k+1}^r}!\langle\widetilde{y}'\rangle.X\mu X.c_X^r?(\widetilde{z}).\overline{c_{k+1}^r}!\langle\widetilde{z}\rangle.X \mid \mathcal{B}_{\text{rec } \widetilde{y}'}^{k+1}(P')) \rhd \diamond} \tag{107}$$

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

$$\text{Send} \cfrac{\Gamma; \Delta, r : S' \vdash P' \rhd \diamond \qquad \Gamma; \Delta_z \vdash z \rhd C}{\Gamma; \Delta, r : S, \Delta_z \vdash r!\langle z\rangle.P' \rhd \diamond} \tag{108}$$

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

Then, by IH on the first assumption of (108) we have:

$$\mathcal{G}(\Gamma_1'); \mathcal{G}(\Delta_1'), \Theta_1 \vdash \mathcal{B}_{\widetilde{y}'}^{k+1}(P') \rhd \diamond \tag{109}$$

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 $(r_1, \ldots, r_{|\mathcal{G}(S)|}) \subseteq \widetilde{y}'$. Also, $\Gamma_1' = \Gamma \setminus \widetilde{x}'$, $\Delta_1' = \Delta \setminus \widetilde{y}'$, and $\text{balanced}(\Theta_1)$ with

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

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

By applying Lem. C.2 on the assumption of (66) we have:

$$\mathcal{G}(\Gamma); \mathcal{G}(\Delta_z) \vdash \widetilde{z} \rhd \mathcal{G}(C) \tag{110}$$

We assume $\widetilde{x} = \widetilde{x}', z$. Since $\widetilde{x}' \subseteq \text{fn}(P')$ follows that $\widetilde{x} \subseteq \text{fn}(P)$. Let $\widetilde{y} = \widetilde{y}', \widetilde{z}$ where $|\widetilde{z}| = |\mathcal{G}(C)|$. By Def. 4.13 follows that $\text{indexed}_{\Gamma, \Delta, r:S, \Delta_z}(\widetilde{y}, \widetilde{x})$. We define $\Theta = \Theta_1, \Theta'$, where:

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

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

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

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:

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

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

We shall prove the following judgment:

$$\mathcal{G}(\Gamma_1); \mathcal{G}(\Delta_1'), \Theta \vdash \mathcal{B}_{\widetilde{y}}^k(r!\langle z\rangle.P') \rhd \diamond \tag{111}$$

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

$$\text{PolySend} \cfrac{\text{End} \cfrac{\text{Nil} \cfrac{}{\mathcal{G}(\Gamma); \emptyset \vdash \mathbf{0} \rhd \diamond}}{\mathcal{G}(\Gamma); \overline{c_{k+1}} : \text{end} \vdash \mathbf{0} \rhd \diamond} \qquad \text{PolyVar} \cfrac{}{\mathcal{G}(\Gamma); \mathcal{G}(\Delta, r : S \setminus \Delta_1') \vdash \widetilde{y}' \rhd \widetilde{M}_1}}{\mathcal{G}(\Gamma); \overline{c_{k+1}} :!\langle\widetilde{M}_1\rangle; \text{end}, \mathcal{G}(\Delta, r : S \setminus \Delta_1') \vdash \overline{c_{k+1}}!\langle\widetilde{y}'\rangle. \rhd \diamond} \tag{112}$$

$$\text{End} \frac{\text{PolySend} \dfrac{(112) \qquad (110)}{\mathcal{G}(\Gamma); \mathcal{G}(\Delta_1 \setminus \Delta_1'), \overline{c_{k+1}} :! \langle \widetilde{M_1} \rangle; \text{end} \vdash r_{f(S)}! \langle \widetilde{z} \rangle.\overline{c_{k+1}}! \langle \widetilde{y'} \rangle.\mathbf{0} \rhd \diamond}}{\mathcal{G}(\Gamma); \mathcal{G}(\Delta_1 \setminus \Delta_1'), \overline{c_{k+1}} :! \langle \widetilde{M_1} \rangle; \text{end}, c_k : \text{end} \vdash r_{f(S)}! \langle \widetilde{z} \rangle.\overline{c_{k+1}}! \langle \widetilde{y'} \rangle.\mathbf{0} \rhd \diamond} \tag{113}$$

$$\text{PolyRcv} \frac{(113) \qquad \text{PolyVar} \dfrac{}{\mathcal{G}(\Gamma); \mathcal{G}(\Delta_1 \setminus \Delta_1') \vdash \widetilde{y} \rhd \widetilde{M}}}{\mathcal{G}(\Gamma_1); \Theta' \vdash c_k?(\widetilde{y}).r_{f(S)}! \langle \widetilde{z} \rangle.\overline{c_{k+1}}! \langle \widetilde{y'} \rangle.\mathbf{0} \rhd \diamond} \tag{114}$$

We may notice that by Def. 4.4 and Def. 10 we have $\mathcal{G}(\Delta_1 \setminus \Delta_1') = \mathcal{G}(\Delta), \widetilde{r} : \mathcal{R}^\star(S), \mathcal{G}(\Delta_z) \setminus \mathcal{G}(\Delta_1')$. 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:

$$\text{Par} \frac{(114) \qquad \dfrac{(109)}{\mathcal{G}(\Gamma_1); \mathcal{G}(\Delta_1'), \Theta_1 \vdash \mathcal{B}_{\widetilde{y'}}^{k+1}(P') \rhd \diamond}}{\mathcal{G}(\Gamma_1); \mathcal{G}(\Delta_1'), \Theta \vdash \mathcal{B}_{\widetilde{y}}^{k}(r! \langle z \rangle.P') \rhd \diamond} \tag{115}$$

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

$$\text{Rcv} \frac{\Gamma; \Delta, r : S', \Delta_z \vdash P' \rhd \diamond \qquad \Gamma; \Delta_z \vdash z \rhd C}{\Gamma \setminus z; \Delta, r : S \vdash r?(z).P' \rhd \diamond} \tag{116}$$

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

Then, by IH on the first assumption of (47) we know:

$$\mathcal{G}(\Gamma_1'); \mathcal{G}(\Delta_1'), \Theta_1 \vdash \mathcal{B}_{\widetilde{y'}}^{k+1}(P') \rhd \diamond \tag{117}$$

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 $(r_1, \ldots, r_{|\mathcal{G}(S)|}) \subseteq \widetilde{y'}$.

Also, $\Gamma_1' = \Gamma \setminus \widetilde{x'}$, $\Delta_1' = \Delta \setminus \widetilde{x'}$, and $\text{balanced}(\Theta_1)$ with

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

and $\Theta_1(c_{k+1}) = ?(\widetilde{M'}); \text{end}$ 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 (47) we have:

$$\mathcal{G}(\Gamma); \mathcal{G}(\Delta_z) \vdash z \rhd \mathcal{G}(C) \tag{118}$$

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. 4.13 $\text{indexed}_{\Gamma, \Delta, r:S}(\widetilde{y}, \widetilde{x})$ 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

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

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:

$$\text{PolySend} \frac{\text{End} \dfrac{\text{Nil} \dfrac{}{\mathcal{G}(\Gamma); \emptyset \vdash \mathbf{0} \rhd \diamond}}{\mathcal{G}(\Gamma); \overline{c_{k+1}} : \text{end} \vdash \mathbf{0} \rhd \diamond} \qquad \text{PolyVar} \dfrac{}{\mathcal{G}(\Gamma); \mathcal{G}(\Delta \setminus \Delta_1) \vdash \widetilde{y'} \rhd \widetilde{M'}}}{\mathcal{G}(\Gamma); \overline{c_{k+1}} :! \langle \widetilde{M'} \rangle; \text{end}, \mathcal{G}(\Delta \setminus \Delta_1) \vdash \overline{c_{k+1}}! \langle \widetilde{y'} \rangle.\mathbf{0} \rhd \diamond} \tag{119}$$

$$\text{End} \frac{\text{Rcv} \dfrac{(119) \qquad (118)}{\mathcal{G}(\Gamma \setminus z); \overline{c_{k+1}} :! \langle \widetilde{M'} \rangle; \text{end}, \mathcal{G}(\Delta_2) \vdash r_{f(S)}?(\widetilde{z}).\overline{c_{k+1}}! \langle \widetilde{y'} \rangle.\mathbf{0} \rhd \diamond}}{\mathcal{G}(\Gamma \setminus z); \overline{c_{k+1}} :! \langle \widetilde{M'} \rangle; \text{end}, c_k : \text{end}, \mathcal{G}(\Delta_2) \vdash r_{f(S)}?(\widetilde{z}).\overline{c_{k+1}}! \langle \widetilde{y'} \rangle.\mathbf{0} \rhd \diamond} \tag{120}$$

$$\text{PolyRcv} \;\; \dfrac{(120) \qquad \text{PolyVar} \;\; \dfrac{}{\mathcal{G}(\Gamma \setminus z); \mathcal{G}(\Delta_2) \vdash \widetilde{y} \rhd \widetilde{M}}}{\mathcal{G}(\Gamma_1 \setminus z); \Theta' \vdash c_k?(\widetilde{y}).r_{f(S)}?(z).\overline{c_{k+1}}!\langle \widetilde{y}' \rangle.\mathbf{0} \rhd \diamond} \tag{121}$$

where $\Delta_2 = \Delta, r : S \setminus \Delta_1$. We may notice that by Def. 4.4 and Def. 10 we have $\mathcal{G}(\Delta_2) = \mathcal{G}(\Delta), \widetilde{r} : \mathcal{R}^\star(S) \setminus \mathcal{G}(\Delta_1)$. 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:

$$\text{Par} \;\; \dfrac{(121) \qquad \dfrac{(117)}{\mathcal{G}(\Gamma_1' \setminus z); \mathcal{G}(\Delta_1), \Theta_1 \vdash \mathcal{B}^{k+1}_{\widetilde{y}'}(P') \rhd \diamond}}{\mathcal{G}(\Gamma_1 \setminus z); \mathcal{G}(\Delta_1), \Theta \vdash c_k?(\widetilde{y}).r_{f(S)}?(\widetilde{z}).\overline{c_{k+1}}!\langle \widetilde{y}' \rangle.\mathbf{0} \mid \mathcal{B}^{k+1}_{\widetilde{y}'}(P') \rhd \diamond} \tag{122}$$

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

$$\text{ResS} \;\; \dfrac{\Gamma; \Delta, s : \mu t.S, \overline{s} : \overline{\mu t.S} \vdash P'}{\Gamma; \Delta \vdash (\nu s : \mu t.S)P'} \tag{123}$$

By IH on the assumption of (123) we have:

$$\mathcal{G}(\Gamma \setminus \widetilde{x}'); \mathcal{G}(\Delta \setminus \widetilde{x}'), \Theta_1 \vdash \mathcal{B}^{k+1}_{\widetilde{y}'}(P') \tag{124}$$

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

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

and $\Theta_1(c_{k+1}) = ?(\widetilde{M}'); \text{end}$ where $\widetilde{M} = (\mathcal{G}(\Gamma), \mathcal{G}(\Delta))(\widetilde{y}')$.
Let $\widetilde{y} = \widetilde{y}' \setminus (\widetilde{s}, \widetilde{\overline{s}})$ and $\widetilde{x} = \widetilde{x}' \setminus (s, \overline{s})$. Since $s, \overline{s} \notin \text{fn}(P)$ we know $\widetilde{x} \subseteq \text{fn}(P)$ and $\text{indexed}_{\Gamma, \Delta}(\widetilde{y}, \widetilde{x})$.
We define $\Theta = \Theta_1, \Theta'$ where

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

By Tab. 3, we have:

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

We should prove the following judgment:

$$\mathcal{G}(\Gamma \setminus \widetilde{x}); \mathcal{G}(\Delta \setminus \widetilde{x}), \Theta \vdash \mathcal{B}^k_{\widetilde{y}}(P) \rhd \diamond$$

We use some auxiliary sub-tree:

$$\text{PolySend} \;\; \dfrac{\text{Nil} \; \dfrac{}{\mathcal{G}(\Gamma); \emptyset \vdash \mathbf{0} \rhd \diamond} \qquad \text{PolyVar} \; \dfrac{}{\mathcal{G}(\Gamma); \mathcal{G}(\Delta_1 \cap \widetilde{x}') \vdash \widetilde{y}' \rhd \widetilde{M}'}}{\mathcal{G}(\Gamma); \mathcal{G}(\Delta \cap \widetilde{x}), \widetilde{s} : \mathcal{R}(S), \widetilde{\overline{s}} : \mathcal{R}(\overline{S}) \vdash \overline{c_{k+1}}!\langle \widetilde{y}' \rangle.\mathbf{0} \rhd \diamond} \tag{125}$$

here we may notice that

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

The following tree proves this case:

$$\text{PolyResS} \;\; \dfrac{\text{Par} \;\; \dfrac{\text{PolyRcv} \;\; \dfrac{(125) \qquad \text{PolyVar} \; \dfrac{}{\mathcal{G}(\Gamma); \mathcal{G}(\Delta \cap \widetilde{x}) \vdash \widetilde{y} \rhd \widetilde{M}}}{\mathcal{G}(\Gamma \setminus \widetilde{x}); \Theta', \widetilde{s} : \mathcal{R}(S), \widetilde{\overline{s}} : \mathcal{R}(\overline{S}) \vdash c_k?(\widetilde{y}).\overline{c_{k+1}}!\langle \widetilde{y}' \rangle.\mathbf{0} \rhd \diamond} \qquad (124)}{\mathcal{G}(\Gamma \setminus \widetilde{x}); \mathcal{G}(\Delta \setminus \widetilde{x}), \Theta, \widetilde{s} : \mathcal{R}(S), \widetilde{\overline{s}} : \mathcal{R}(\overline{S}) \vdash c_k?(\widetilde{y}).\overline{c_{k+1}}!\langle \widetilde{y}' \rangle.\mathbf{0} \mid \mathcal{B}^k_{\widetilde{y}'}(P') \rhd \diamond}}{\mathcal{G}(\Gamma \setminus \widetilde{x}); \mathcal{G}(\Delta \setminus \widetilde{x}), \Theta \vdash (\nu \widetilde{s} : \mathcal{R}(S))(c_k?(\widetilde{y}).\overline{c_{k+1}}!\langle \widetilde{y}' \rangle.\mathbf{0} \mid \mathcal{B}^k_{\widetilde{y}'}(P')) \rhd \diamond} \tag{126}$$

$\square$

$\square$

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

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

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

$$\Gamma\sigma; \Delta\sigma \vdash P\sigma \triangleright \diamond \tag{127}$$

By Def. 4.10, we shall prove the following judgment:

$$\mathcal{G}(\Gamma\sigma); \mathcal{G}(\Delta\sigma) \vdash (\nu\widetilde{c})(\overline{c_k}!\langle\widetilde{r}\rangle.\mathbf{0} \mid \mathcal{B}_{\widetilde{r}}^k(P\sigma)) \triangleright \diamond \tag{128}$$

where $\widetilde{c} = (c_k, \ldots, c_{k+|P|-1})$; $k > 0$; and $\widetilde{r} = \bigcup_{v \in \widetilde{v}}\{v_1, \ldots, v_{|\mathcal{G}(S)|}\}$ with $v : S$. Since $\widetilde{v} \subseteq \mathsf{fn}(P)$ we know $\mathsf{indexed}_{\Gamma\sigma,\Delta\sigma}(\widetilde{r}, \widetilde{v})$. Since $P\sigma$ is an initialized process, we apply Thm. 3 to (127) to get:

$$\mathcal{G}(\Gamma\sigma); \mathcal{G}(\Delta\sigma \setminus \widetilde{v}), \Theta \vdash \mathcal{B}_{\widetilde{r}}^k(P\sigma) \triangleright \diamond \tag{129}$$

where Θ is balanced with $\mathsf{dom}(\Theta) = \{c_k, c_{k+1}, \ldots, c_{k+|P|-1}\} \cup \{\overline{c_{k+1}}, \ldots, \overline{c_{k+|P|-1}}\}$ and $\Theta(c_k) = ?(\widetilde{M}); \mathsf{end}$ with $\widetilde{M} = \mathcal{G}(\Delta)(\widetilde{r})$.

The following tree proves this case:

$$
\cfrac{
  \cfrac{
    \cfrac{
      \cfrac{\text{Nil} \quad \cfrac{}{\mathcal{G}(\Gamma\sigma); \emptyset \vdash \mathbf{0} \triangleright \diamond}}
      {\mathcal{G}(\Gamma\sigma); c_k : \mathsf{end} \vdash \mathbf{0} \triangleright \diamond}\text{End} \qquad \text{PolyVar} \cfrac{}{\mathcal{G}(\Gamma\sigma); \widetilde{r} : \widetilde{M} \vdash \widetilde{r} \triangleright \widetilde{M}}
    }
    {\mathcal{G}(\Gamma\sigma); \overline{c_k} :!\langle\widetilde{M}\rangle; \mathsf{end}, \widetilde{r} : \widetilde{M} \vdash \overline{c_k}!\langle\widetilde{r}\rangle.\mathbf{0} \triangleright \diamond}\text{Send} \qquad (129)
  }
  {\mathcal{G}(\Gamma\sigma); \mathcal{G}(\Delta\sigma), \overline{c_k} :!\langle\widetilde{M}\rangle; \mathsf{end}, \Theta \vdash \overline{c_k}!\langle\widetilde{r}\rangle.\mathbf{0} \mid \mathcal{B}_{\widetilde{r}}^k(P\sigma) \triangleright \diamond}\text{Par}
}
{\mathcal{G}(\Gamma\sigma); \mathcal{G}(\Delta\sigma) \vdash (\nu\widetilde{c})(\overline{c_k}!\langle\widetilde{r}\rangle.\mathbf{0} \mid \mathcal{B}_{\widetilde{r}}^k(P\sigma)) \triangleright \diamond}\text{ResS}
$$

□

C.5 MST Bisimilarity

Now, we are ready to relate processes, modulo indexed names (Def. 4.14), using the relation $\diamond$ defined as follows:

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

$$
\begin{array}{cc}
\text{[IPSND]} & \\
\cfrac{P\sigma \diamond P' \qquad v\sigma \bowtie_{\mathsf{c}} \widetilde{v} \qquad \sigma = \mathsf{next}(n_i)}{n_i!\langle v\rangle.P \diamond n_i!\langle\widetilde{v}\rangle.P'} & \quad \cfrac{\text{[IPINACT]}}{\mathbf{0} \diamond \mathbf{0}}
\end{array}
$$

$$
\begin{array}{cc}
\text{[IPRCV]} & \text{[IPNEWS]} \\
\cfrac{P\sigma \diamond P' \qquad v\sigma \bowtie_{\mathsf{c}} \widetilde{v} \qquad \sigma = \mathsf{next}(n_i)}{n_i?(y).P \diamond n_i?(\widetilde{y}).P'} & \cfrac{P \diamond P' \qquad \widetilde{m}_1 \bowtie_{\mathsf{c}} \widetilde{m}_2}{(\nu\,\widetilde{m}_1)P \diamond (\nu\,\widetilde{m}_2)P'}
\end{array}
$$

Definition C.2 (Minimal characteristic processes).

$$\langle ?(C); S\rangle_i^u \overset{\mathsf{def}}{=} u_i?(x).(t!\langle u_{i+1}, \ldots, u_{i+|\mathcal{G}(S)|}\rangle.\mathbf{0} \mid \langle C\rangle_i^x)$$

$$\langle !\langle C\rangle; S\rangle_i^u \overset{\mathsf{def}}{=} u_i!\langle\langle C\rangle_{\mathsf{c}}\rangle.t!\langle u_{i+1}, \ldots, u_{i+|\mathcal{G}(S)|}\rangle.\mathbf{0}$$

$$\langle \mathsf{end}\rangle_i^u \overset{\mathsf{def}}{=} \mathbf{0}$$

$$\langle\langle C\rangle\rangle_i^u \overset{\mathsf{def}}{=} u_1!\langle\langle C\rangle_{\mathsf{c}}\rangle.t!\langle u_1\rangle.\mathbf{0}$$

$$\langle \mu t.S\rangle_i^u \overset{\mathsf{def}}{=} \langle S\{\mathsf{end}/t\}\rangle_i^u$$

$$\langle S\rangle_{\mathsf{c}} \overset{\mathsf{def}}{=} \widetilde{s} \ (|\widetilde{s}| = |\mathcal{G}(S)|, \widetilde{s} \text{ fresh})$$

$$\langle\langle C\rangle\rangle_{\mathsf{c}} \overset{\mathsf{def}}{=} a_1 \ (a_1 \text{ fresh})$$

Definition C.3 (Trigger Collections). We let H, H' to range over *trigger collections*: processes of the form $P_1 \mid \cdots \mid P_n$ (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 $H_1 = t \Leftarrow_{\mathsf{c}} v : C \mid [\![C]\!]^{u_1} \mid t_2!\langle u_2\rangle.\mathbf{0}$ where v, t_1, t_2, u_1, u_2 are channel names, C a channel type. We could see that, $[\![C]\!]^{u_1}$ 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	$C_{\tilde{x}}^{\tilde{u}}(P)$	
$Q_1 \parallel Q_2$	$\{R_1 \parallel R_2 : R_1 \in C_{\tilde{y}}^{\tilde{u}_1}(Q_1), R_2 \in C_{\tilde{z}}^{\tilde{u}_2}(Q_2)\}$	$\widetilde{y} = \mathsf{fnb}(Q_1, \tilde{x}), \widetilde{z} = \mathsf{fnb}(Q_2, \tilde{x})$ $\{\tilde{u}/\tilde{x}\} = \{\tilde{u}_1/\tilde{y}\} \cdot \{\tilde{u}_2/\tilde{z}\}$
$(\nu s : C)Q$	$\{(\nu \widetilde{s} : \mathcal{G}(C))\, R : C_{\tilde{x}}^{\tilde{u}}(Q\sigma)\}$	$\widetilde{s} = (s_1, \ldots, s_{|\mathcal{G}(C)|})$ $\sigma = \{s_1 \overline{s_1}/s\overline{s}\}$
Q	$\{(\nu \widetilde{c})R : R = \overline{c_k}!\langle \tilde{u}\rangle \mid \mathcal{B}_{\tilde{x}}^{k}(Q)\} \cup \{(\nu \widetilde{c})R : R \in \mathcal{J}_{\tilde{x}}^{\tilde{u}}(Q)\}$	$\widetilde{c} = \mathsf{fpn}(R)$
$\mathbf{0}$	$\{\mathbf{0}\}$	
P	$\mathcal{J}_{\tilde{x}}^{\tilde{u}}(P)$	
$n_i?(y).Q$	$\{n_l\rho?(\widetilde{y}).\overline{c_{k+1}}!\langle \widetilde{z}\rho\rangle \mid \mathcal{B}_{\tilde{z}}^{k+1}(Q\sigma)\}$	$y : S \wedge \widetilde{y} = (y_1, \ldots, y_{|S|})$ $\widetilde{w} = (\mathtt{lin}(n_i))\,?\,\{n_i\}:\epsilon$ $\widetilde{z} = \mathsf{fnb}(Q, \widetilde{x}\widetilde{y} \setminus \widetilde{w})$ $\rho = \{\tilde{u}/\tilde{x}\}$ $\sigma = \mathsf{next}(n_i) \cdot \{y_1/y\}$ $l = (\mathtt{tr}(S))\,?\,f(S):i$
$n_i!\langle y_j\rangle.Q$	$\{n_l\rho!\langle \widetilde{y}\rho\rangle.\overline{c_{k+1}}!\langle \widetilde{z}\rho\rangle \mid \mathcal{B}_{\tilde{z}}^{k+1}(Q\sigma)\}$	$y : S \wedge \widetilde{y} = (y_j, \ldots, y_{j+|\mathcal{G}(S)|-1})$ $\widetilde{w} = (\mathtt{lin}(n_i))\,?\,\{n_i\}:\epsilon$ $\widetilde{z} = \mathsf{fnb}(Q, \tilde{x} \setminus \widetilde{w})$ $\rho = \{\tilde{u}/\tilde{x}\}$ $\sigma = \mathsf{next}(n_i)$ $l = (\mathtt{tr}(S))\,?\,f(S):i$
$Q_1 \mid Q_2$	$\{\overline{c_k}!\langle \widetilde{y}\rho\rangle.\overline{c_{k+l}}!\langle \widetilde{z}\rho\rangle \mid \mathcal{B}_{\tilde{y}}^{k}(Q_1) \mid \mathcal{B}_{\tilde{z}}^{k+l}(Q_2)\}$ $\cup$ $\{(R_1 \mid R_2) : C_{\tilde{y}}^{\tilde{u}_1}(Q_1), R_2 \in C_{\tilde{z}}^{\tilde{u}_2}(Q_2)\}$	$\widetilde{y} = \mathsf{fnb}(Q_1, \widetilde{x})$ $\widetilde{z} = \mathsf{fnb}(Q_2, \widetilde{x})$ $\rho = \{\tilde{u}/\tilde{x}\}$ $l = |Q_1|$
$\mu X.P$	$\{\overline{c_{k+1}^r}!\langle \widetilde{z}\rho\rangle.\mu X.c_X^r?(\widetilde{y}).\overline{c_{k+1}^r}!\langle \widetilde{y}\rangle.X \mid$ $\mathcal{B}_{\mathsf{rec}\,\tilde{x}}^{k+1}(P)_g\}$	$\widetilde{n} = \mathsf{fn}(P)$ $\widetilde{n} : \widetilde{C} \wedge \widetilde{m} = \mathsf{bn}(\widetilde{n} : \widetilde{C})$ $\widetilde{z} = \widetilde{x} \cup \widetilde{m}, |\widetilde{z}| = |\widetilde{y}|$ $g = \{X \mapsto \widetilde{m}\}$
$\mathbf{0}$	$\{\mathbf{0}\}$	

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 $\mathsf{fpn}(P)$ 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 $\mathsf{recd}(P)$ to count sequential prefixes in P preceding recursive variable X or a subprocess of shape $\mu X.Q$. Function $\mathsf{recd}(P)$ and $\mathsf{recd}'(P)$ are mutually defined as follows:

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

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

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

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

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

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

P	$C_{\tilde{x}}^{\tilde{u}}(P)$	
Q	$C_{\mathrm{rec}\ \tilde{x}}^{\tilde{u}}\big(\mu X.Q^*\big)^d$ if $Q \equiv \alpha_d.\alpha_{d-1}.\ldots.\alpha_{d-p}.(P_X \mid R)$ and $\mu X.Q^*$ is a subprocess of P_X $\{(\nu\widetilde{c})R\ :\ R \in \{\overline{c_k}!\langle\tilde{u}\rangle \mid \mathcal{B}_{\tilde{x}}^k(Q), c_k^r!\langle\tilde{u}\rangle \mid \mathcal{B}_{\mathrm{rec}\ \tilde{x}}^k(Q)\}\}$ $\cup\ \{(\nu\widetilde{c})R\ :\ R \in \mathcal{J}_{\tilde{x}}^{\tilde{u}}(Q) \cup \mathcal{J}_{\mathrm{rec}\ \tilde{x},\rho}^k(Q)_{\emptyset}^d\}$ otherwise	$\widetilde{c} = \mathsf{fpn}(R)$ $\rho = \{\tilde{u}/\tilde{x}\}$
$C_{\mathrm{rec}\ \tilde{x}}^{\tilde{u}}(Q)^d$	$\{(\nu\widetilde{c})(\mu X.c_X^r?(\widetilde{y}).\overline{c_{k+1}^r}!\langle\widetilde{y}\rangle.X \mid R \mid c_k^r?(\widetilde{x}).c_X^r!\langle\widetilde{x}\rangle.X\ :\ R \in \mathcal{J}_{\mathrm{rec}\ \tilde{x}}^k(Q)_g^d\}\ \cup N$ where: $M = \begin{cases} \{(\nu\widetilde{c})(\mu X.c_X^r?(\widetilde{y}).\overline{c_{k+1}^r}!\langle\widetilde{y}\rangle.X \mid R \mid \\ \quad c_X^r!\langle\widetilde{x}\rho\rangle.\mu X.c_k^r?(\widetilde{x}).c_X^r!\langle\widetilde{x}\rangle.X)\ :\ R \in \mathcal{J}_{\mathrm{rec}\ \tilde{x}}^k(Q)_g^0\} & \text{if } g \neq \emptyset \\ \{(\nu\widetilde{c})(\mu X.c_X^r?().\overline{c_{k+1}^r}!\langle\rangle.X \mid R \mid \\ \quad c_X^r!\langle\rangle \mid \mu X.c_k^r?().(c_X^r!\langle\rangle \mid X))\ :\ R \in \mathcal{J}_{\mathrm{rec}\ \tilde{x}}^k(Q)_g^0\} & \text{otherwise} \end{cases}$ $N = \begin{cases} \{(\nu\widetilde{c})(\overline{c_k^r}!\langle\widetilde{z}\rho\rangle.\mu X.c_X^r?(\widetilde{y}).\overline{c_k^r}!\langle\widetilde{y}\rangle.X \mid \mathcal{B}_{\mathrm{rec}\ \tilde{x}}^k(Q)_g)\} \cup M & \text{if } d = 0 \\ \emptyset & \text{otherwise} \end{cases}$	$\widetilde{c} = \mathsf{fpn}(R)$ $Q \equiv \mu X.Q^*$ $\widetilde{x} = \mathsf{fs}(Q^*)$ $g = \{X \mapsto \widetilde{x}\}$ $\rho = \{\tilde{u}/\tilde{x}\}$
P	$\mathcal{J}_{\mathrm{rec}\ \tilde{x},\rho}^k(P)_g^d$ when $g \neq \emptyset$	
$n_i!\langle y_j\rangle.Q$	$\{n_l!\langle\widetilde{y}\rho\rangle.\overline{c_{k+1}^r}!\langle\widetilde{z}\rho\rangle.\mu X.c_k^r?(\widetilde{x}).n_l!\langle\widetilde{y}\rangle.\overline{c_{k+1}^r}!\langle\widetilde{z}\rangle.X \mid \mathcal{B}_{\mathrm{rec}\ \tilde{z}}^{k+1}(Q\sigma)_g\}$ if $\mathsf{recd}(Q) = d$ $N \cup \{\mu X.c_k^r?(\widetilde{x}).n_l!\langle\widetilde{y}\rangle.\overline{c_{k+1}^r}!\langle\widetilde{z}\rangle.X \mid R : R \in \mathcal{J}_{\mathrm{rec}\ \tilde{z},\rho}^{k+1}(Q\sigma)_g^d\}$ otherwise where: $N = \begin{cases} \{\overline{c_{k+1}^r}!\langle\widetilde{z}\rho\rangle.\mu X.c_k^r?(\widetilde{x}).n_l!\langle\widetilde{y}\rangle.\overline{c_{k+1}^r}!\langle\widetilde{z}\rangle.X \mid \mathcal{B}_{\mathrm{rec}\ \tilde{z}}^{k+1}(Q\sigma)_g\} & \text{if } \mathsf{recd}(Q) = d+1 \\ \emptyset & \text{otherwise} \end{cases}$	$y : T \wedge (y_j,\ldots,y_{j+|\mathcal{G}(T)|-1})$ $\widetilde{w} = (\mathsf{lin}(u_i))\,?\,\{u_i\}{:}\,\epsilon$ $\widetilde{z} = g(X) \cup \mathsf{fnb}(Q,\widetilde{x}\setminus\widetilde{w})$ $l = (\mathsf{tr}(u_i))\,?\,f(S){:}\,i$ $\sigma = \mathsf{next}(u_i)$
$n_i?(y).Q$	$\{n_l?(\widetilde{y}).\overline{c_{k+1}^r}!\langle\widetilde{z}\rho\rangle.\mu X.c_k^r?(\widetilde{x}).n_l?(\widetilde{y}).\overline{c_{k+1}^r}!\langle\widetilde{z}\rangle.X \mid \mathcal{B}_{\mathrm{rec}\ \tilde{z}}^{k+1}(Q\sigma)_g\}$ if $\mathsf{recd}(Q) = d$ $N \cup \{\mu X.c_k^r?(\widetilde{x}).n_l?(\widetilde{y}).\overline{c_{k+1}^r}!\langle\widetilde{z}\rangle.X \mid R : R \in \mathcal{J}_{\mathrm{rec}\ \tilde{z},\rho}^{k+1}(Q\sigma)_g^d\}$ $N = \begin{cases} \{\overline{c_{k+1}^r}!\langle\widetilde{z}\rho\rangle.\mu X.c_k^r?(\widetilde{x}).n_l?(\widetilde{y}).\overline{c_{k+1}^r}!\langle\widetilde{z}\rangle.X \mid \mathcal{B}_{\mathrm{rec}\ \tilde{z}}^{k+1}(Q\sigma)_g\} & \text{if } \mathsf{recd}(Q) = d+1 \\ \emptyset & \text{otherwise} \end{cases}$	$\widetilde{w} = (\mathsf{lin}(u_i))\,?\,\{u_i\}{:}\,\epsilon$ $\widetilde{z} = g(X) \cup \mathsf{fnb}(Q,\widetilde{xy}\setminus\widetilde{w})$ $l = (\mathsf{tr}(u_i))\,?\,f(S){:}\,i$ $\sigma = \mathsf{next}(u_i) \cdot \{y_1/y\}$
$Q_1 \mid Q_2$	$N \cup \{\mu X.c_k^r?(\widetilde{x}).(\overline{c_{k+1}^r}!\langle\widetilde{y}_1\rangle.X \mid c_{k+l+1}^r!\langle\widetilde{y}_2\rangle) \mid R_1 \mid R_2 :$ $R_1 \in \mathcal{J}_{\mathrm{rec}\ \tilde{y}_1,\rho}^{k+1}(Q_1)_{g_1}^d, R_2 \in \mathcal{J}_{\mathrm{rec}\ \tilde{y}_2,\rho}^{k+l+1}(Q_2)_{g_2}^d\}$ $N = \begin{cases} \{\overline{c_{k+1}^r}!\langle\widetilde{y}_1\rho\rangle.\mu X.c_k^r?(\widetilde{x}).(\overline{c_{k+1}^r}!\langle\widetilde{y}_1\rangle.X \mid c_{k+l+1}^r!\langle\widetilde{y}_2\rangle) \mid \\ \quad \mathcal{B}_{\mathrm{rec}\ \tilde{y}_1}^{k+1}(Q_1)_{g_1} \mid \mathcal{B}_{\mathrm{rec}\ \tilde{y}_2}^{k+l+1}(Q_2)_{g_2}\} & \text{if } \mathsf{recd}(Q_1) = d \\ \emptyset & \text{otherwise} \end{cases}$	$X(Q_1)$ $\widetilde{y}_1 = g(X) \cup \mathsf{fnb}(Q_1,\widetilde{x})$ $\widetilde{y}_2 = \mathsf{fnb}(Q_2,\widetilde{x})$ $l = |Q_1|$
$(\nu s : C)Q$	$N \cup \{\mu X.(\nu\widetilde{s} : \mathcal{G}(C))c_k^r?(\widetilde{x}).\overline{c_{k+1}^r}!\langle\widetilde{z}\rangle.X \mid R : R \in \mathcal{J}_{\mathrm{rec}\ \tilde{z},\rho}^{k+1}(Q\sigma)_g^d\}$ $R = \overline{c_{k+1}^r}!\langle\widetilde{z}\rho\rangle.\mu X.(\nu\widetilde{s} : \mathcal{G}(C))c_k^r?(\widetilde{x}).\overline{c_{k+1}^r}!\langle\widetilde{z}\rangle.X$ $N = \begin{cases} \{(\nu\widetilde{s} : \mathcal{G}(C))R \mid \mathcal{B}_{\mathrm{rec}\ \tilde{z}}^{k+1}(Q\sigma)_g\} & \text{if } \mathsf{recd}(Q_1) = d \\ \emptyset & \text{otherwise} \end{cases}$	$\widetilde{s} = (s_1,\ldots,s_{|\mathcal{G}(S)|})$ $\overline{\widetilde{s}} = (\overline{s_1},\ldots,\overline{s_{|\mathcal{G}(S)|}})$ $\widetilde{n} = (\mathsf{lin}(s))\,?\,\widetilde{n}{:}\,\epsilon$ $\widetilde{z} = \widetilde{x},\widetilde{s},\widetilde{n}$ $\sigma = \{s_1\overline{s_1}/s\overline{s}\}$
X	$\mathbf{0}$	
P	$\mathcal{J}_{\mathrm{rec}\ \tilde{x},\rho}^k(P)_g^d$ when $g = \emptyset$	
$n_i!\langle y_j\rangle.Q$	$\{n_l!\langle\widetilde{y}\rho\rangle.c_{k+1}^r!\langle\widetilde{z}\rho\rangle \mid \mu X.c_k^r?(\widetilde{x}).(n_l!\langle\widetilde{y}\rangle.c_{k+1}^r!\langle\widetilde{z}\rangle \mid X) \mid \mathcal{B}_{\mathrm{rec}\ \tilde{z}}^{k+1}(Q\sigma)_g\}$	
$n_i?(y).Q$	$\{n_l?(\widetilde{y}).c_{k+1}^r!\langle\widetilde{z}\rho\rangle \mid \mu X.c_k^r?(\widetilde{x}).(n_l?(\widetilde{y}).c_{k+1}^r!\langle\widetilde{z}\rangle \mid X) \mid \mathcal{B}_{\mathrm{rec}\ \tilde{z}}^{k+1}(Q\sigma)_g\}$	
$Q_1 \mid Q_2$	$\{c_{k+1}^r!\langle\widetilde{y}_1\rho\rangle \mid c_{k+l+1}^r!\langle\widetilde{y}_2\rho\rangle \mid \mu X.c_k^r?(\widetilde{x}).(c_{k+1}^r!\langle\widetilde{y}_1\rangle \mid c_{k+l+1}^r!\langle\widetilde{y}_2\rangle \mid X) \mid$ $\mathcal{B}_{\mathrm{rec}\ \tilde{y}_1}^{k+1}(Q_1)_{g_1} \mid \mathcal{B}_{\mathrm{rec}\ \tilde{y}_2}^{k+l+1}(Q_2)_{g_2}\}$	
$(\nu s : C)Q$	$\{c_{k+1}^r!\langle\widetilde{z}\rho\rangle \mid \mu X.(\nu\widetilde{s} : \mathcal{G}(C))c_k^r?(\widetilde{x}).(c_{k+1}^r!\langle\widetilde{z}\rangle \mid X) \mid \mathcal{B}_{\mathrm{rec}\ \tilde{z}}^{k+1}(Q)_g\}$	

Table 5: The extension of $C_{\tilde{x}}^{\tilde{u}}(P)$ and definition of $\mathcal{J}_{\mathrm{rec}\ \tilde{x},\rho}^k(P)_g^d$ (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

$$\widetilde{u} = (a, b, s, \bar{s}, \bar{s}', s', r, r', \ldots)$$

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

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

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

$$\mathcal{S} = \big\{(P\{\tilde{u}/\tilde{x}\}, R) : R \in C^{\tilde{u}_*}_{\tilde{x}_*}(P\sigma)$$
$$\text{with } \widetilde{n} = \mathsf{fn}(P) \cup \widetilde{u} \cup \widetilde{x}, \sigma \in \mathsf{index}(\widetilde{n}),$$
$$\widetilde{u_*} = \mathsf{bn}(\widetilde{u}\sigma : \widetilde{C}), \widetilde{x_*} = \mathsf{bn}(\widetilde{x}\sigma : \widetilde{C})\big\}$$

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

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 $\widehat{C}^{\rho}_{\sigma}(P)$ as follows:

$$\widehat{C}^{\rho}_{\sigma}(P_1) = C^{\tilde{u}_*}_{\tilde{x}_*}(P\sigma)$$
$$\text{with } \rho = \{\tilde{u}/\tilde{x}\}, \widetilde{u_*} = \mathsf{bn}(\widetilde{u}\sigma : \widetilde{C}), \widetilde{x_*} = \mathsf{bn}(\widetilde{x}\sigma : \widetilde{C})$$

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

C.6 Operational correspondence

LEMMA C.12. *Given an indexed process $P_1\{\tilde{u}/\tilde{x}\}$, the set $C^{\tilde{u}}_{\tilde{x}}(P_1)$ is closed under τ transitions on non-essential prefixes. That is, if $R_1 \in C^{\tilde{u}}_{\tilde{x}}(P_1)$ and $R_1 \xrightarrow{\tau} R_2$ is inferred from the actions on non-essential prefixes, then $R_2 \in C^{\tilde{u}}_{\tilde{x}}(P_1)$.*

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. □

LEMMA C.13. *Relation $\diamond$ is a MST bisimulation.*

SKETCH. Straightforward by the transition induction and Def. C.1. □

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 $P_1\{\tilde{u}/\tilde{x}\} \xrightarrow{(\nu\widetilde{m_1})n!\langle v:C_1\rangle} P_2$ then there exist Q_2 and σ_v such that $Q_1 \xrightarrow{(\nu\widetilde{m_2})\check{n}!\langle \widetilde{v}:\mathcal{G}(C_1)\rangle} Q_2$ where $v\sigma_v \bowtie_{\mathsf{c}} \widetilde{v}$ and, for a fresh t,*

$$(\nu\widetilde{m_1})(P_2 \parallel t \Leftarrow_{\mathsf{C}} v:C_1) \,\mathcal{S}\, (\nu\widetilde{m_2})(Q_2 \parallel t \Leftarrow_{\mathsf{m}} v\sigma_v:C_1)$$

(2) *Whenever $P_1\{\tilde{u}/\tilde{x}\} \xrightarrow{n?(v)} P_2$ then there exist Q_2 and σ_v such that $Q_1 \xrightarrow{\check{n}?(\tilde{v})} Q_2$ where $v\sigma_v \bowtie \widetilde{v}$ and $P_2 \,\mathcal{S}\, Q_2$,*
(3) *Whenever $P_1 \xrightarrow{\tau} P_2$ then there exists Q_2 such that $Q_1 \Longrightarrow Q_2$ 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 $\sigma_1 \in \mathsf{index}(\mathsf{fn}(P_1) \cup \widetilde{u} \cup \widetilde{x})$, such that $Q_1 \in \widehat{C}^{\rho_1}_{\sigma_1}(P_1)$. 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 $Q_2 \in \widehat{C}^{\rho_2}_{\sigma_2}(P_2)$ 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 $n_i = \mathsf{subj}(\check{\ell})$. First, we consider two base cases: Rules Snd and Rv. Then, we consider inductive cases: Rules Par$_\mathsf{L}$ 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$_\mathsf{L}$) emphasizing specifics of the recursion breakdown.

(1) Case Snd. We distinguish two sub-cases: (i) $P_1 = n!\langle v\rangle.P_2$ and (ii) $P_1 = n!\langle y\rangle.P_2$ with $\{v/y\} \in \rho_1$. In both sub-cases we know there is $\rho_2 = \{\tilde{u}_2/\tilde{x}_2\}$ such that

$$P_1\rho_1 = n\rho_1!\langle v\rangle.P_2\rho_2$$

We have the following transition:

$$\langle\mathsf{Snd}\rangle \; \frac{}{P_1\rho_1 \xrightarrow{n\rho!\langle v\rangle} P_2\rho_2}$$

Let $\sigma_1 = \sigma' \cdot \{n_i/n\}$ where $\sigma' \in \text{index}(\text{fn}(P_1) \cup \widetilde{u} \cup \widetilde{x})$ and $\sigma_2 = \sigma_1 \cdot \sigma$ where $\sigma = \text{next}(n_i)$. Here we take $\sigma_v = \sigma_2$. Further, let $\widetilde{u_*} = \text{bn}(\widetilde{u}\sigma_1 : \widetilde{C})$ and $\widetilde{x_*} = \text{bn}(\widetilde{x}\sigma_1 : \widetilde{C})$ with $\widetilde{u} : \widetilde{C}$. Also, let $\widetilde{z} = \text{fnb}(P_2, \widetilde{x_*} \setminus \widetilde{w})$ where $\widetilde{w} = \{n_i\}$ if $\text{lin}(n_i)$ otherwise $\widetilde{w} = \epsilon$. Then, in both sub-cases, there are two possibilities for the shape of Q_1, namely:

$$Q_1^1 = (\nu \widetilde{c_k})(\overline{c_k}!\langle \widetilde{u_*} \rangle \mid \mathcal{B}_{\widetilde{x_*}}^k(P_1\sigma_1))$$

$$Q_1^2 = (\nu \widetilde{c_{k+1}})n_i\rho!\langle \widetilde{v} \rangle.\overline{c_{k+1}}!\langle \widetilde{z}\rho \rangle \mid \mathcal{B}_{\widetilde{z}}^{k+1}(P_2\sigma_2)$$

where $v\sigma_v \bowtie_{\mathsf{c}} \widetilde{v}$. 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:

$$Q_1^2 \xrightarrow{n_i\rho!\langle \widetilde{v} \rangle} Q_2$$

where $Q_2 = (\nu \widetilde{c_{k+1}})\overline{c_{k+1}}!\langle \widetilde{z}\rho \rangle \mid \mathcal{B}_{\widetilde{z}}^{k+1}(P_2\sigma_2)$. Then, we should show the following:

$$(P_2 \parallel t \Leftarrow_{\mathsf{C}} v : C_1)\rho_2 \; \mathcal{S} \; Q_2 \parallel t \Leftarrow_{\mathsf{m}} v\sigma_v : C_1 \tag{130}$$

By Tab. 4 we can see that

$$Q_2 \in C_{\widetilde{z}}^{\widetilde{z}\rho_2}(P_2\sigma_2) \tag{131}$$

Here we remark that by construction $\sigma_2 = \sigma_1 \cdot \{n_{i+1}/n_i\}$.
By Def. 4.7 and Def. C.8 we know that

$$\widetilde{z} = \text{fnb}(P_2, \text{bn}(\widetilde{x} : \widetilde{C}) \setminus \widetilde{w}) = \text{bn}(\widetilde{x}_2\sigma_1 \cdot \sigma : \widetilde{C}_2)$$

with $\widetilde{x}_2 : \widetilde{C}_2$. Similarly, we have

$$\widetilde{z}\rho = \text{bn}(\widetilde{x}_2\rho_2\sigma_1 \cdot \sigma : \widetilde{C}_2)$$

Further, we know there is

$$\sigma_2' \in \text{index}(\text{fn}(P_2) \cup \widetilde{u}_2 \cup \widetilde{x}_2)$$

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 (131) we have that $P_2\rho_2 \, \mathcal{S} \, Q_2$. Next, by the definition we have $t \Leftarrow_{\mathsf{C}} v : C_1 \, \mathcal{S} \, t \Leftarrow_{\mathsf{m}} v\rho\sigma : C_1$. Thus, the goal (130) 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:

$$\langle \mathsf{Rv} \rangle \; \frac{}{n\rho_1?(y).P_2\rho_2 \xrightarrow{n?(v)} P_2\rho_2 \cdot \{v/y\}}$$

Let $\sigma_1 = \sigma' \cdot \{n_i/n\}$ where $\sigma' \in \text{index}(\text{fn}(P_1) \cup \widetilde{u} \cup \widetilde{x})$ and $\sigma_2 = \sigma_1 \cdot \sigma$ where $\sigma = \text{next}(n_i) \cdot \{y_1/y\}$. Further, let $\widetilde{u_*} = \text{bn}(\widetilde{u}\sigma_1 : \widetilde{C})$ and $\widetilde{x_*} = \text{bn}(\widetilde{x}\sigma_1 : \widetilde{C})$ with $\widetilde{u} : \widetilde{C}$. Also, let $\widetilde{y} = (y_1, \ldots, y_{|\mathcal{G}(S)|})$ and $\widetilde{z} = \text{fnb}(P_2, \widetilde{x_*}\widetilde{y} \setminus \widetilde{w})$ where $\widetilde{w} = \{n_i\}$ if $\text{lin}(n_i)$ otherwise $\widetilde{w} = \epsilon$. Then, there are two possibilities for the shape of Q_1, namely:

$$Q_1^1 = (\nu \widetilde{c_k})(\overline{c_k}!\langle \widetilde{u_*} \rangle \mid \mathcal{B}_{\widetilde{x_*}}^k(P_1\sigma_1))$$

$$Q_1^2 = (\nu \widetilde{c_{k+1}})n_i\rho?(\widetilde{y}).\overline{c_{k+1}}!\langle \widetilde{z}\rho \rangle \mid \mathcal{B}_{\widetilde{z}}^{k+1}(P_2\sigma_2)$$

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:

$$Q_1^2 \xrightarrow{n_i\rho?\langle \widetilde{v} \rangle} Q_2$$

where $Q_2 = (\nu \widetilde{c_{k+1}})\overline{c_{k+1}}!\langle \widetilde{z}\rho \cdot \{\tilde{v}/\tilde{y}\} \rangle \mid \mathcal{B}_{\widetilde{z}}^{k+1}(P_2\sigma_2)$ and $v\sigma_v \bowtie_{\mathsf{c}} \widetilde{v}$ for some $\sigma_v \in \text{index}(v)$. Now, we should show the following:

$$P_2\rho_2 \cdot \{v/y\} \, \mathcal{S} \, Q_2 \tag{132}$$

By Tab. 4 we can see that

$$Q_2 \in C_{\widetilde{z}}^{\widetilde{z}\rho_2 \cdot \{\tilde{v}/\tilde{y}\}}(P_2\sigma_2 \cdot \sigma_v) \tag{133}$$

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

$$\sigma_2 \cdot \sigma_v \in \text{index}(\text{fn}(P_2) \cup \widetilde{u}_2 \cup \widetilde{x}_2)$$

By Def. 4.7 and Def. C.8 we have that

$$\widetilde{z} = \text{fnb}(P_2, \text{bn}(\widetilde{x} : \widetilde{C}) \cdot \widetilde{y} \setminus \widetilde{w})$$

$$= \text{bn}(\widetilde{x}y(\sigma_1 \cdot \sigma) : \widetilde{C}S)$$

with $y : S$. Similarly, we have

$$\widetilde{z}\rho_2 \cdot \{\widetilde{v}/\widetilde{y}\} = \mathsf{bn}(\widetilde{x}y\rho_2 \cdot \{v/y\}(\sigma_2 \cdot \sigma_v) : \widetilde{C}S)$$

Thus, by this and (133) the goal (132) 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 $\mathsf{Par_L}$. Here we know $P_1 = P_1' \mid P_1''$. The final rule in the inference tree is as follows:

$$\langle \mathsf{Par}_L \rangle \quad \frac{P_1'\{\tilde{u}_1'/\tilde{y}_1\} \overset{\ell}{\longrightarrow} P_2'\{\tilde{u}_2'/\tilde{y}_2\} \qquad \mathsf{bn}(\ell) \cap \mathsf{fn}(P_1'') = \emptyset}{P_1'\{\tilde{u}_1'/\tilde{y}_1\} \mid P_1''\{\tilde{u}_1''/\tilde{z}_1\} \overset{\ell}{\longrightarrow} P_2'\{\tilde{u}_2'/\tilde{y}_2\} \mid P_1''\{\tilde{u}_1''/\tilde{z}_2\}}$$

Let $\sigma_1 \in \mathsf{index}(\mathsf{fn}(P_1) \cup \tilde{u} \cup \tilde{x})$. Further, let $\tilde{y} = \mathsf{fnb}(P_1', \tilde{x}_*)$, $\tilde{z} = \mathsf{fnb}(P_1'', \tilde{x}_*)$, $\widetilde{u_*} = \mathsf{bn}(\widetilde{u}\sigma_1 : \widetilde{C})$, $\widetilde{x_*} = \mathsf{bn}(\widetilde{x}\sigma_1 : \widetilde{C})$ with $\tilde{u} : \widetilde{C}$, and $\rho_\mathsf{m} = \{\tilde{u}_*/\tilde{x}_*\}$. In sub-case (i), by the definition of $\mathcal{S}$ (Tab. 4), there are following possibilities for Q_1:

$$Q_1^1 = (\nu \widetilde{c}_k)(\overline{c_k}!\langle \widetilde{u_*}\rangle \mid \mathcal{B}_{\tilde{x}_*}^k(P_1\sigma_1))$$

$$Q_1^2 = \overline{c_k}!\langle \tilde{y}\rho_\mathsf{m}\rangle.\overline{c_{k+l}}!\langle \tilde{z}\rho_\mathsf{m}\rangle \mid \mathcal{B}_{\tilde{y}}^k(P_1'\sigma_1) \mid \mathcal{B}_{\tilde{z}}^{k+l}(P_1''\sigma_1)$$

$$N_1^3 = \{(R_1' \mid R_1'') : C_{\tilde{y}}^{\tilde{y}\rho_\mathsf{m}}(P_1'\sigma_1), R_1'' \in C_{\tilde{z}}^{\tilde{z}\rho_\mathsf{m}}(P_1''\sigma_1)\}$$

By Lem. C.12 there exist

$$Q_1' \in C_{\tilde{y}}^{\tilde{y}\rho_\mathsf{m}}(P_1'\sigma_1) \tag{134}$$

$$Q_1'' \in C_{\tilde{z}}^{\tilde{z}\rho_\mathsf{m}}(P_1''\sigma_1) \tag{135}$$

such that

$$Q_1^1 \overset{\tau}{\Longrightarrow} Q_1^2 \overset{\tau}{\Longrightarrow} Q_1' \mid Q_1''$$

Thus, in both cases we consider how $Q_1' \mid Q_1''$ evolves. By the definition of $\mathcal{S}$ we have

$$P_1'\rho_1' \, \mathcal{S} \, Q_1' \tag{136}$$

$$P_1''\rho_1'' \, \mathcal{S} \, Q_1'' \tag{137}$$

To apply IH we do the case analysis on the action ℓ:

- Sub-case $\ell \equiv n?\langle v \rangle$. If $v \in \widetilde{u}$ then we take $\sigma_v = \sigma_1$, otherwise $\sigma_v = \{v_j/v\}$ for $j > 0$. Then, by applying IH to (136) we know there is Q_2' such that $Q_1' \overset{n?\langle \tilde{v}\rangle}{\Longrightarrow} Q_2'$ and

$$P_2'\{\tilde{u}_2'/\tilde{y}_2\} \, \mathcal{S} \, Q_2' \tag{138}$$

where $v\sigma_v \bowtie_\mathsf{c} \widetilde{v}$ and $\widetilde{u}_2' = \widetilde{u}_1' \cdot \widetilde{v}$. We should show that

$$P_2'\rho_2' \mid P_1''\rho_1'' \, \mathcal{S} \, Q_2' \mid Q_1'' \tag{139}$$

where $\rho_2' = \{\tilde{u}_2'/\tilde{y}_1\}$ and $\rho_1'' = \{\tilde{u}_1''/\tilde{z}_1\}$. Let $\sigma_2 = \sigma_1 \cdot \mathsf{next}(n_i)$ such that $\{n_i/n\} \in \sigma_1$. Then, by (138) we know

$$Q_2' \in \widehat{C}_{\sigma_2 \cdot \sigma_v}^{\rho_2'}(P_2') \tag{140}$$

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 (140) and (135) we have

$$Q_2' \mid Q_1'' \in \widehat{C}_{\sigma_2 \cdot \sigma_v}^{\rho_2' \cdot \rho_1'}(P_2' \mid P_1'') \tag{141}$$

Further, we may notice that $\sigma_2 \cdot \sigma_v \in \mathsf{index}(\mathsf{fn}(P_2' \mid P_1'') \cup \rho_2' \cup \rho_1'')$. So, by this and (141) the goal (139) follows.

- Sub-case $\ell = \tau$. By applying IH to (136) we know there is Q_2' such that $Q_1' \overset{\ell}{\Longrightarrow} Q_2'$ and

$$P_2'\rho_2' \, \mathcal{S} \, Q_2' \tag{142}$$

where $\rho_2' = \{\tilde{u}_2'/\tilde{y}_2\}$. We should show that

$$P_2'\rho_2' \mid P_1''\rho_1'' \, \mathcal{S} \, Q_2' \mid Q_1'' \tag{143}$$

By (142), we know there is $\sigma_2 \in \mathsf{index}(\mathsf{fn}(P_2) \cup \rho_2')$ such that

$$Q_2' \in \widehat{C}_{\sigma_2}^{\rho_2'}(P_2')$$

Further, by remark we know that either $\sigma'_2 = \sigma_1 \cdot \{n_{i+1}/n_i\} \cdot \{\overline{n}_{i+1}/\overline{n}_i\}$ for some $n_i, \overline{n}_i \in \mathsf{fn}(P'_2\rho'_2)$ and $n_i, \overline{n}_i \notin \mathsf{fn}(P''_1\rho''_1)$ 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 (142) and (135) and the definition of σ'_2 the goal (143) follows.

- Sub-case $\ell \equiv (\nu\,\widetilde{m}_1)n!\langle v : C_1\rangle$. Here we omit details on substitutions as they are similar to the first sub-case. By applying IH to (136) we know there is Q'_2 such that $Q'_1 \stackrel{\ell}{\Longrightarrow} Q'_2$ and

$$(\nu\,\widetilde{m_1})(P'_2 \parallel t \Leftarrow_{\mathsf{C}} v : C_1)\ \mathcal{S} \tag{144}$$

$$(\nu\,\widetilde{m_2})(Q'_2 \parallel t \Leftarrow_{\mathsf{m}} v\sigma_1 : C_1) \tag{145}$$

We should show that

$$(\nu\,\widetilde{m_1})(P'_2 \mid P''_1 \parallel t \Leftarrow_{\mathsf{C}} v : C_1)\ \mathcal{S} \tag{146}$$

$$(\nu\,\widetilde{m_2})(Q'_2 \mid P''_1 \parallel t \Leftarrow_{\mathsf{m}} v\sigma_1 : C_1) \tag{147}$$

By Def. C.10 and Tab. 4 from (144) we can infer the following:

$$P'_2\{\tilde{u}'_1/\tilde{y}'\}\ \mathcal{S}\ Q'_2 \tag{148}$$

So, by (137) and (148) the goal (146) follows.

This concludes $\mathsf{Par}_{\mathsf{L}}$ case.

(4) Case Tau. Here we know $P_1 = P'_1 \mid P''_1$. Without the loss of generality, we assume $\ell_1 = (\nu\,\widetilde{m}_1)\overline{n}_i!\langle v_1\rangle$ and $\ell_2 = n_i?(v_1)$. Let $\rho'_1 = \{\tilde{u}'_1/\tilde{y}\}$ and $\rho''_1 = \{\tilde{u}''_1/\tilde{z}\}$ 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:

$$\langle\mathsf{Tau}\rangle\ \frac{P'_1\rho'_1 \stackrel{\ell_1}{\longrightarrow} P'_2\rho'_2 \qquad P''_1\rho''_1 \stackrel{\ell_2}{\longrightarrow} P''_2\rho'_2 \qquad \ell_1 \asymp \ell_2}{P'_1\rho'_1 \mid P''_1\rho''_1 \stackrel{\tau}{\longrightarrow} (\nu\,\widetilde{m}_1)(P'_2\rho'_2 \mid P''_2\rho'_2)}$$

Let $\sigma_1, \widetilde{y}, \widetilde{z}, \widetilde{u}_*, \widetilde{x}_*$, and ρ_{m} be defined as in the previous case. Further, Q_1 can have shapes: Q^1_1, Q^2_1, and N^3_1 as in the previous case. As in the previous case, we know that there are

$$Q'_1 \in C_{\tilde{y}}^{\tilde{y}\rho_{\mathsf{m}}}(P'_1\sigma_1) \tag{149}$$

$$Q''_1 \in C_{\tilde{z}}^{\tilde{z}\rho_{\mathsf{m}}}(P''_1\sigma_1) \tag{150}$$

such that

$$Q^1_1 \stackrel{\tau}{\Longrightarrow} Q^2_1 \stackrel{\tau}{\Longrightarrow} Q'_1 \mid Q''_1$$

Thus, in both cases we consider how $Q'_1 \mid Q''_1$ evolves. By the definition of $\mathcal{S}$ we have

$$P'_1\rho'_1\ \mathcal{S}\ Q'_1 \tag{151}$$

$$P''_1\rho''_1\ \mathcal{S}\ Q''_1 \tag{152}$$

We apply IH component-wise:

- By applying IH to (151) we know there is Q'_2 such that $Q'_1 \stackrel{(\nu\,\widetilde{m}_2)\overline{n}_i!\langle\tilde{v}\rangle}{\Longrightarrow} Q'_2$ and

$$(\nu\,\widetilde{m_1})(P'_2 \parallel t \Leftarrow_{\mathsf{C}} v : C_1)\rho'_2\ \mathcal{S} \tag{153}$$

$$(\nu\,\widetilde{m_2})(Q'_2 \parallel t \Leftarrow_{\mathsf{m}} v\sigma_1 : C_1) \tag{154}$$

where $v\sigma_1 \bowtie_{\mathsf{C}} \tilde{v}$.

- By applying IH to (152) we know there is Q''_2 such that $Q''_1 \stackrel{n_i?\langle\tilde{v}\rangle}{\Longrightarrow} Q''_2$ and

$$P''_2\rho''_2\ \mathcal{S}\ Q''_2 \tag{155}$$

Now, by (153) 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 $\sigma_2' \subseteq \sigma_1 \cdot \text{next}(\overline{n}_i)$. So, we have

$$Q_2' \in \widehat{C}_{\sigma_1 \cdot \text{next}(\overline{n}_i)}^{\rho_2'}(P_2')$$

Similarly, there is $\sigma_2'' \in \text{index}(\text{fn}(P_2'') \cup \rho_2'')$ and $\sigma_2'' \subseteq \sigma_1 \cdot \text{next}(\overline{n}_i)$, so we have the following

$$Q_2'' \in \widehat{C}_{\sigma_1 \cdot \text{next}(n_i)}^{\rho_2''}(P_2'')$$

So, we have

$$Q_2' \mid Q_2'' \in \widehat{C}_{\sigma_1 \cdot \text{next}(n_i) \cdot \text{next}(\overline{n}_i)}^{\rho_2' \cdot \rho_2''}(P_2' \mid P_2'')$$

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 $\sigma_2 \subseteq \sigma_1 \cdot \text{next}(n_i) \cdot \text{next}(\overline{n}_i)$. That is, we know

$$\widehat{C}_{\sigma_2}^{\rho_2' \cdot \rho_2''}(P_2' \mid P_2'') = \widehat{C}_{\sigma_1 \cdot \text{next}(n_i) \cdot \text{next}(\overline{n}_i)}^{\rho_2' \cdot \rho_2''}(P_2' \mid P_2'')$$

Thus, we can conclude

$$(\nu \widetilde{m_1})(P_2' \mid P_2'')\rho_2' \cdot \rho_2'' \; \mathcal{S} \; (\nu \widetilde{m_2})(Q_2' \mid Q_2'')$$

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

$$P_1^1 = \alpha_d.\alpha_{d-1}.\ldots.\alpha_{d-p}.(P_X \mid R)$$

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 $R \not\equiv \mathbf{0}$. 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 $\sigma' \in \text{index}(\text{fn}(P_1) \cup \widetilde{u} \cup \widetilde{x})$ and $\sigma_2 = \sigma_1 \cdot \sigma$ where $\sigma = \text{next}(n_i) \cdot \{y_1/y\}$. Further, let $\widetilde{u}_* = \text{bn}(\widetilde{u}\sigma_1 : \widetilde{C})$ and $\widetilde{x}_* = \text{bn}(\widetilde{x}\sigma_1 : \widetilde{C})$ with $\widetilde{u} : \widetilde{C}$. Also, let $\widetilde{y} = (y_1, \ldots, y_{|\mathcal{G}(S)|})$ and $\widetilde{z} = \text{fnb}(P_2, \widetilde{x_*}\widetilde{y} \setminus \widetilde{w})$ where $\widetilde{w} = \{n_i\}$ if $\text{lin}(n_i)$ otherwise $\widetilde{w} = \epsilon$.
We could see that $d = \text{recd}(\alpha_d, \alpha_{d-1}, \ldots, \alpha_1.\mu X.P_1^*)$. So, by Tab. 5 there are following possibilities for the shape of Q_1, namely elements in N defined as:

$$N = \{(\nu \widetilde{c})R : R \in C_{\text{rec} \; \widetilde{x}_*,\rho}^{\widetilde{u}_*}(\mu X.P^*)^d\}$$

Let

$$Q_1^1 = (\nu \widetilde{c})B_1 \mid n_l?(\widetilde{y}).\overline{c_{k+1}^r}!\langle \widetilde{z}\rho \rangle.\mu X.c_k^r?(\widetilde{x}).n_l?(\widetilde{y}).\overline{c_{k+1}^r}!\langle \widetilde{z} \rangle.X \mid \mathcal{B}_{\text{rec} \; \widetilde{z}}^{k+1}(\alpha_d, \alpha_{d-1}.P_1^2\sigma)_g$$

where B_1 is, intuitively, trios mimicking prefixes before α_d and P_1^2 is such that $P_1^1 = \alpha_d.\alpha_{d-1}.P_1^2$. By unfolding definition of $C_{\text{rec} \; \widetilde{x}_*,\rho}^{\widetilde{u}_*}(\mu X.P^*)^d$ we have that $Q_1^1 \in N$. Further, if $R \in C_{\text{rec} \; \widetilde{x}_*,\rho}^{\widetilde{u}_*}(\mu X.P^*)^d$ then $R \xrightarrow{\tau} Q_1^1$. So, we only analyze how Q_1^1 evolves. We can infer the following:

$$Q_1^1 \xrightarrow{n_i\rho?\langle \widetilde{v} \rangle} Q_2$$

where

$$Q_2 = B_1 \mid \overline{c_{k+1}^r}!\langle \widetilde{z}\rho \cdot \{\tilde{v}/\tilde{y}\} \rangle.\mu X.c_k^r?(\widetilde{x}).n_l?(\widetilde{y}).\overline{c_{k+1}^r}!\langle \widetilde{z} \rangle.X \mid \mathcal{B}_{\text{rec} \; \widetilde{z}}^{k+1}(\alpha_{d-1}.P_1^2\sigma)_g$$

with $v\sigma_v \bowtie_c \widetilde{v}$ for some $\sigma_v \in \text{index}(v)$. Now, we should show that

$$P_2\rho_2 \cdot \{^v/_y\} \, \mathcal{S} \, Q_2 \tag{156}$$

Let $\widetilde{u}_z = \widetilde{z}\rho_2 \cdot \{^{\widetilde{v}}/_{\widetilde{y}}\}$. In sub-case (i) we should show that $Q_2 \in C_{\widetilde{z}}^{\widetilde{u}_z}(\alpha_{d-1}.P_1^2)$. So, we should show that

$$Q_2 \in C_{\text{rec } \widetilde{z}, \rho_2 \cdot \{^{\widetilde{v}}/_{\widetilde{y}}\}}^{\widetilde{u}_z}(P^*)_g^{d-1}$$

This follows by the inspection of definition $\mathcal{J}_{\text{rec } \widetilde{z}, \rho_2 \cdot \{^{\widetilde{v}}/_{\widetilde{y}}\}}^k (P^*)_g^{d-1}$. That is, we can notice that

$$Q_2^1 = B_1 \mid B_2$$

where $B_2 \in \mathcal{J}_{\text{rec } \widetilde{z}, \rho_2 \cdot \{^{\widetilde{v}}/_{\widetilde{y}}\}}^k (\alpha_{d-1}.P_1^2\sigma)_g^{d-1}$ as $(d-1)+1 = \text{recd}(\alpha_d.\alpha_{d-1}.P_1^2\sigma)$. So, (156) 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

$$Q_2^2 = (\nu \, \widetilde{c})V_1 \mid c_{k+l+1}^r !\langle \widetilde{u}_{z_2} \rangle \mid V_2$$

where

$$V_1 = B_1 \mid \mu X.c_k^r?(\widetilde{x}).n_l?(\widetilde{y}).\overline{c_{k+1}^r}!\langle \widetilde{z} \rangle.X \mid \overline{c_{k+1}^r}!\langle \widetilde{u}_{z_1} \rangle.\mu X.c_k^r?(\widetilde{x}).(\overline{c_{k+1}^r}!\langle \widetilde{z}_1 \rangle.X \mid c_{k+l+1}^r !\langle \widetilde{z}_2 \rangle)$$
$$V_2 = \mathcal{B}_{\text{rec } \widetilde{z}_1}^{k+1}(P_X)_{g_1} \mid \mathcal{B}_{\text{rec } \widetilde{z}_2}^{k+l+1}(R)_\emptyset$$

Now, we should show that $Q_2^2 \in C_{\widetilde{z}}^{\widetilde{u}_z}(\alpha_{d-1}.P_1^2 \mid R)$. By Tab. 5 we have that

$$\{R_1 \mid R_2 : R_1 \in C_{\text{rec } \widetilde{z}_1}^{\widetilde{u}_{z_1}}(\alpha_{d-1}.P_1^2), \ R_2 \in C_{\widetilde{z}_2}^{\widetilde{u}_{z_2}}(R)\} \subseteq C_{\widetilde{z}}^{\widetilde{u}_z}(\alpha_{d-1}.P_1^2 \mid R) \tag{157}$$

Further, we know that

$$(\nu \widetilde{c}_{k+l+1})c_{k+l+1}^r !\langle \widetilde{u}_z \rangle \mid \mathcal{B}_{\text{rec } \widetilde{z}_2}^{k+l+1}(R)_\emptyset \in C_{\widetilde{z}_2}^{\widetilde{u}_{z_2}}(R) \tag{158}$$

and

$$V_1 \mid V_2 \in C_{\text{rec } \widetilde{z}_1}^{\widetilde{u}_{z_1}}(\alpha_{d-1}.P_1^2) \tag{159}$$

Thus, by (157), (158), and (159) we have the following:

$$V_1 \mid V_2 \mid (\nu \widetilde{c}_{k+l+1})c_{k+l+1}^r !\langle \widetilde{u}_z \rangle \mid \mathcal{B}_{\text{rec } \widetilde{z}_2}^{k+l+1}(R)_\emptyset \in C_{\widetilde{z}}^{\widetilde{u}_z}(\alpha_{d-1}.P_1^2 \mid R)$$

Now, we can notice that $\widetilde{c}_{k+l+1} \cap \text{fpn}(V_1) = \emptyset$ and $\widetilde{c}_{k+l+1} \cap \text{fpn}(\mathcal{B}_{\text{rec } \widetilde{z}_1}^{k+1}(P_X)_{g_1}) = \emptyset$. Further, we have

$$(\nu \widetilde{c}_{k+l+1})\mathcal{B}_{\text{rec } \widetilde{z}_2}^{k+l+1}(R)_\emptyset \approx^\mathsf{C} \mathbf{0}$$

(where $\approx^\mathsf{C}$ 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

$$Q_2^2 \equiv V_1 \mid V_2 \mid (\nu \widetilde{c}_{k+l+1})c_{k+l+1}^r !\langle \widetilde{u}_z \rangle \mid \mathcal{B}_{\text{rec } \widetilde{z}_2}^{k+l+1}(R)_\emptyset$$

This concludes Rv case. Now, we consider the inductive case.

(2) Case Par_L. Here we know $P_1 = P_1' \mid P_1''$. Similarly to the previous case, let

$$P_1^1 = \alpha_d.\alpha_{d-1}.\ldots.\alpha_{d-p}.(P_X \mid R)$$

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:

$$\langle \text{Par}_L \rangle \ \frac{P_1'\{\widetilde{u}_1'/\widetilde{y}_1\} \xrightarrow{\ell} P_2'\{\widetilde{u}_2'/\widetilde{y}_2\} \mid R\{\widetilde{w}_1/\widetilde{z}_1\} \cdot \{\widetilde{u}_R'/\widetilde{w}_1\} \qquad \text{bn}(\ell) \cap \text{fn}(P_1'') = \emptyset}{P_1'\{\widetilde{u}_1'/\widetilde{y}_1\} \mid R\{\widetilde{u}_1''/\widetilde{z}_1\} \xrightarrow{\ell} P_2'\{\widetilde{u}_2'/\widetilde{y}_2\} \mid R\{\widetilde{u}_1''/\widetilde{z}_1\} \mid R\{\widetilde{w}_1/\widetilde{z}_1\} \cdot \{\widetilde{u}_R'/\widetilde{w}_1\}}$$

Let $\sigma_1 \in \text{index}(\text{fn}(P_1) \cup \widetilde{u} \cup \widetilde{x})$. Further, let $\widetilde{y}_{*1} = \text{fnb}(P'_1, \widetilde{x}_*)$, $\widetilde{z}_{*1} = \text{fnb}(R, \widetilde{x}_*)$, $\widetilde{u}_* = \text{bn}(\widetilde{u}\sigma_1 : \widetilde{C})$ where $\widetilde{x}_* = \text{bn}(\widetilde{x}\sigma_1 : \widetilde{C})$ with $\widetilde{u} : \widetilde{C}$, and $\rho_{\mathfrak{m}} = \{\widetilde{u}_*/\widetilde{x}_*\}$. By the definition of $\mathcal{S}$ (Tab. 5), there are following possibilities for Q_1:

$$Q_1^1 = (\nu \widetilde{c}_k)(\overline{c_k}!\langle \widetilde{u}_* \rangle \mid \mathcal{B}_{\widetilde{x}_*}^k (P_1 \sigma_1))$$

$$Q_1^2 = (\nu \widetilde{c}_k)\overline{c_k}!\langle \widetilde{u}_{*2}' \rangle.\overline{c_{k+l}}!\langle \widetilde{u}_{*1}'' \rangle \mid \mathcal{B}_{\widetilde{y}_{*1}}^k (P'_1 \sigma_1) \mid \mathcal{B}_{\widetilde{z}_{*1}}^{k+l} (R \sigma_1)$$

$$N_1^3 = \{(R'_1 \mid R''_1) : C_{\text{rec } \widetilde{y}_{*1}}^{\widetilde{u}_{*2}'} (\mu X.P^* \sigma_1)_g^d, \ R''_1 \in C_{\widetilde{z}_{*1}}^{\widetilde{u}_{*1}''} (R \sigma_1)\}$$

By Lem. C.12 there exist

$$Q'_1 \in C_{\text{rec } \widetilde{y}_{*1}}^{\widetilde{u}_{*2}'} (\mu X.P^* \sigma_1)_g^d \tag{160}$$

$$Q''_1 \in C_{\widetilde{z}_{*1}}^{\widetilde{u}_{*1}''} (R \sigma_1) \tag{161}$$

such that

$$Q_1^1 \overset{\tau}{\Longrightarrow} Q_1^2 \overset{\tau}{\Longrightarrow} Q'_1 \mid Q''_1$$

The interesting case is to consider a process Q_1^3 defined as:

$$Q_1^3 = (\nu \widetilde{c})V_1 \mid c_{k+l+1}^r!\langle \widetilde{u}_{*1}'' \rangle \mid V_2$$

where

$$V_1 = B_1 \mid \overline{c_{k+1}^r}!\langle \widetilde{u}_{*2}' \rangle.\mu X.c_k^r?(\widetilde{y}_{*1}).(\overline{c_{k+1}^r}!\langle \widetilde{y}_{*2} \rangle.X \mid c_{k+l+1}^r!\langle \widetilde{z}_{*2} \rangle)$$

$$V_2 = \mathcal{B}_{\text{rec } \widetilde{y}_{*2}}^{k+1} (P_X)_{g_1} \mid \mathcal{B}_{\text{rec } \widetilde{w}_{*2}}^{k+l+1} (R\{\widetilde{w}_2/\widetilde{z}_2\})_\emptyset$$

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 $v \in \widetilde{u}$ then we take $\sigma_v = \sigma_1$, otherwise $\sigma_v = \{v_j/v\}$ for $j > 0$. Now, we could see that

$$Q_1^3 \overset{n?\langle \widetilde{v} \rangle}{\Longrightarrow} Q_2$$

where

$$Q_2 = (\nu \widetilde{c})V_1 \mid \overline{c_{k+1}^r}!\langle \widetilde{u}_{*2}' \rangle.\mu X.c_k^r?(\widetilde{y}_{*1}).(\overline{c_{k+1}^r}!\langle \widetilde{y}_{*2} \rangle.X \mid c_{k+l+1}^r!\langle \widetilde{z}_{*1} \rangle) \mid c_{k+l+1}^r!\langle \widetilde{u}_{*R}' \rangle \mid V_2$$

We define Q_2^1 as follows

$$Q_2^1 = (\nu \widetilde{c})V_1 \mid \overline{c_{k+1}^r}!\langle \widetilde{u}_{*2}' \rangle.\mu X.c_k^r?(\widetilde{y}_{*1}).(\overline{c_{k+1}^r}!\langle \widetilde{y}_{*2} \rangle.X \mid c_{k+l+1}^r!\langle \widetilde{z}_{*1} \rangle) \mid V_2 \mid$$
$$(\nu \widetilde{c}_k)(c_k^r!\langle \widetilde{u}_{*1}'' \rangle \mid \mathcal{B}_{\text{rec } \widetilde{z}_{*1}}^k (R)_\emptyset) \mid (\nu \widetilde{c}_k)(c_k^r!\langle \widetilde{u}_{*R}' \rangle \mid \mathcal{B}_{\text{rec } \widetilde{w}_{*1}}^k (R\{\widetilde{w}_1/\widetilde{z}_1\})_\emptyset)$$

By definition, we could see that $Q_2^1 \in C_{\widetilde{z}_{*1}}^{\widetilde{u}_{*2}} (P'_2 \mid R \mid R\{\widetilde{w}_1/\widetilde{z}_1\})$. Now, by the definition of $\mathcal{B}_{\text{rec} \ -}^- (\cdot)_\emptyset$ we have

$$c_{k+l+1}^r!\langle \widetilde{u}_{*1}'' \rangle \mid c_{k+l+1}^r!\langle \widetilde{u}_{*R}' \rangle \mid \mathcal{B}_{\text{rec } \widetilde{z}_{*1}}^{k+l+1} (R)_\emptyset \approx^C$$
$$(\nu \widetilde{c}_k)(c_k^r!\langle \widetilde{u}_{*1}'' \rangle \mid \mathcal{B}_{\text{rec } \widetilde{z}_{*1}}^k (R)_\emptyset) \mid (\nu \widetilde{c}_k)(c_k^r!\langle \widetilde{u}_{*R}' \rangle \mid \mathcal{B}_{\text{rec } \widetilde{w}_{*1}}^k (R\{\widetilde{w}_1/\widetilde{z}_1\})_\emptyset)$$

As each trio in $\mathcal{B}_{\text{rec } \widetilde{z}_{*1}}^{k+l+1} (R)_\emptyset$ 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 $P_1\{\widetilde{u}/\widetilde{x}\}$ is a well-formed process and $P_1\{\widetilde{u}/\widetilde{x}\} \mathcal{S} Q_1$.*

(1) *Whenever $Q_1 \overset{(\nu \widetilde{m}_2)\check{n}!\langle \widetilde{v}:\mathcal{G}(C_1) \rangle}{\longrightarrow} Q_2$ then there exist P_2 and σ_v such that $P_1\{\widetilde{u}/\widetilde{x}\} \overset{(\nu \widetilde{m}_1)n!\langle v:C_1 \rangle}{\longrightarrow} P_2$ where $v\sigma_v \bowtie_c \widetilde{v}$ and, for a fresh t,*

$$(\nu \widetilde{m}_1)(P_2 \parallel t \Leftarrow_C v:C_1) \ \mathcal{S} \ (\nu \widetilde{m}_2)(Q_2 \parallel t \Leftarrow_{\mathfrak{m}} v\sigma_v:C_1)$$

(2) *Whenever $Q_1 \overset{\check{n}?(\widetilde{v})}{\longrightarrow} Q_2$ then there exist P_2 and σ_v such that $P_1\{\widetilde{u}/\widetilde{x}\} \overset{n?(v)}{\longrightarrow} P_2$ where $v\sigma_v \bowtie \widetilde{v}$ and $P_2 \ \mathcal{S} \ Q_2$,*

(3) *Whenever $Q_1 \overset{\tau}{\longrightarrow} Q_2$ either (i) $P_1\{\widetilde{u}/\widetilde{x}\} \ \mathcal{S} \ Q_2$ or (ii) there exists P_2 such that $P_1 \overset{\tau}{\longrightarrow} P_2$ 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*

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

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 $\tilde{r} = \mathsf{rn}(P)$. Further, let $\sigma = \bigcup_{v \in \tilde{r}}\{v_1/v\}$, so we have $\sigma \in \mathsf{index}(\tilde{r})$. Then, let $\widetilde{r_*} = \mathsf{bn}(\tilde{r} : \widetilde{S})$ and $\widetilde{x_*} = \mathsf{bn}(\tilde{x} : \widetilde{S})$ where $\tilde{r} : \widetilde{S}$. Therefore, by Def. 4.10 and Tab. 4 we have $\mathcal{F}^*(P) \in C_{\widetilde{x_*}}^{\widetilde{r_*}}(P_1)$. Finally, by Def. C.10 we have $(P, \mathcal{F}^*(P)) \in \mathcal{S}$.

□