A Procedure for Splitting Data-Aware Processes and its Application to Coordination

S.-S.T.Q. Jongmans$,*, D. Clarke$b, J. Proença$b

$Centrum Wiskunde & Informatica, Science Park 123, 1098 XG, Amsterdam, Netherlands
*bKatholieke Universiteit Leuven, Celestijnenlaan 200A, 3001 Heverlee, Belgium

Abstract

We present a procedure for splitting processes in a process algebra with multiactions and data (the untimed subset of the specification language mCRL2). This splitting procedure cuts a process into two processes along a set of actions $A$: roughly, one of these processes contains no actions from $A$, while the other process contains only actions from $A$. We state and prove a theorem asserting that the parallel composition of these two processes is provably equal from a set of axioms (sound and complete with respect to strong bisimilarity) to the original process under some appropriate notion of synchronization.

We apply our splitting procedure to the process algebraic semantics of the coordination language Reo: using this procedure and its related theorem, we formally establish the soundness of splitting Reo connectors along the boundaries of their (a)synchronous regions in implementations of Reo. Such splitting can significantly improve the performance of connectors as shown elsewhere.

1. Motivation

Context. Over the past decades, coordination languages have emerged for the specification and implementation of interaction protocols among entities running concurrently (components, services, threads, etc.). This class of languages includes Reo [2, 3], a graphical language for compositional construction of connectors: communication media through which entities can interact with each other. Figure 1 shows some example Reo connectors in their usual graphical syntax. Intuitively, connectors consist of one or more channels (i.e., the edges of a connector graph), through which data items flow, and a number of nodes (i.e., the vertices of a connector graph), on which channel ends (i.e., the endpoints of edges) meet. Through channel composition—the act of gluing channels together on nodes—engineers can construct complex connectors. Channels often used include the reliable synchronous channel, called $\text{sync}$, and the reliable asynchronous channel $\text{fifo}_n$, which has a buffer of capacity $n$. Importantly, while nodes have a fixed semantics, Reo features an open-ended set of channels. This allows engineers to define their own channels with custom semantics.

To use connectors in real applications, one must derive executable code from graphical specifications of connectors (e.g., those in Figure 1). Roughly two implementation approaches currently exist. In the distributed approach [11, 37, 35, 36], one implements the behavior of each of the $k$ constituents of a connector and runs these $k$ implementations concurrently as a distributed system; in the centralized approach [19, 17, 22], one computes the behavior of a connector as a whole, implements this behavior, and runs this implementation sequentially as a centralized system. For example, in the case of a service-oriented choreography application, the distributed approach seems natural, because the services involved run on different machines and the network between them may play a role in their coordination. However, if coordination involves computation threads running on the same machine in some multithreading application, the centralized approach
appears more appropriate, because it avoids communication among the constituents of a connector at run-
time: in this approach, due to the computation of the behavior of an entire connector at compile time, one
abstracts from the individual, smaller, concurrent constituents of a connector to obtain one big sequential
program for the whole (which can run in its own dedicated thread at run-time, among the computation
threads it coordinates).

One optimization technique applicable to both the distributed and the centralized approach involves
the identification of the synchronous and the asynchronous regions of a connector [36]. A synchronous
region contains exactly those nodes and channels of a connector that synchronize collectively to decide on
their individual behavior; an asynchronous region connects synchronous regions in an asynchronous way,
typically involving a fifo channel. For instance, the connector consisting of a sync channel, a fifo1 channel,
and another sync channel (see Figure 1d) has two synchronous regions, connected by an asynchronous region.

Intuitively, two synchronous regions can run completely independently of each other. In the distributed
approach, this means that nodes and channels need to share information only with those nodes and channels
in the same synchronous region—not with every node or channel in the connector [36]. In the centralized
approach, this means that one does not need to compute the behavior of a connector as a whole, but rather
on a per-region basis [17]. Supplementary, asynchronous regions connect synchronous regions to each other
by transporting data and control information between them. Based on how asynchronous regions do this, one
can distinguish different versions of the region-based optimization technique, with different guarantees and
for different use cases. For example, an asynchronous region can transport control information directly (in
which case transportation starts at the same time as the coordination step that triggered it and ends before
the next), atomically (same as the previous case but transportation can start also after the coordination
step that triggered it), or interleaved (same as the previous case but transportation does not need to end
before the next coordination step). Recent work shows that the region-based optimization technique for Reo
can significantly improve performance [11, 22, 35, 36] (both at compile time and at run-time), to the extent
that its use will become vital for real-world applications: without it, automatically deploying (including
code generation) and running connectors quickly becomes infeasible as their size increases.

Problem. The region-based optimization technique still has a serious problem: although we have reason to
believe (based on intuition and loose informal reasoning) that it preserves the semantics of a connector, we
do not know this for sure by lack of a formal proof.

Contributions of the paper. In this paper, using the existing process algebraic semantics of Reo [27, 24,
25, 26], we prove the correctness of the region-based optimization technique for asynchronous regions with
direct transportation. In this semantics, expressed using the specification language mCRL2 [14, 16], one
associates every connector with a process describing its behavior. Roughly, our proof technique consists of
the formulation of a number of theorems for the untimed subset of mCRL2. We then apply these theorems
to Reo’s process algebraic semantics to prove the region-based optimization technique correct.

Importantly, however, the scope of this paper extends beyond Reo. Because we work on the semantics
level—in terms of process algebra—and because we formulate our proof technique for general processes (not
just those used in Reo’s semantics), our results apply not exclusively to Reo but, instead, to any process
in untimed mCRL2. As a result, we can divide the contributions of this paper into two categories: those concerning mCRL2 in general and those concerning Reo. More concretely:

- **mCRL2**
  - We define a *splitting procedure* for the untimed subset of mCRL2 and prove its correctness. Essentially, this procedure syntactically splits a process into two new processes: one process contains only actions from some set $A$; the other contains only actions from outside $A$.
  - Our work shows the feasibility of using the language mCRL2 (not the associated toolset) for proving properties of a whole language, Reo, rather than of individual concrete connectors. This subtly, yet significantly, differs from work of Kokash et al. [27, 24, 25, 26], who introduced a process algebraic semantics of Reo for verifying concrete connectors (e.g., “this connector never deadlocks”) but obtain no results about Reo as a language. As such, the work presented in this paper also paves the way to proving other properties about Reo using process algebra, including the correctness of others versions of the region-based optimization techniques (in terms of new different splitting procedures).

- **Reo**
  - We formalize the notion of (a)synchronous regions in terms of the process algebraic semantics of Reo.
  - We apply the splitting procedure to the process algebraic semantics of Reo, thereby justifying the region-based optimization technique for Reo implementations. To illustrate this further, we discuss how to implement and use the splitting procedure in the distributed approach, exploiting the local concurrency available on the computational nodes.

Although motivated by Reo, to emphasize the generality of our splitting procedure and theorems, we have organized the rest of this paper from a process algebra perspective; Reo serves as a ‘case study’ exemplifying their usefulness. In Section 2, we give an overview of the untimed subset of mCRL2 we use. In Section 3, to show mCRL2 in action, we summarize the process algebraic semantics of Reo. In Section 4, we introduce our splitting procedure, and in Section 5, we prove its correctness. In Section 6, we apply our splitting procedure to Reo. Section 7 contains related work, and Section 8 ends this paper with a conclusion and future work.

An earlier version of this work appeared in [20], where we considered the untimed data-free subset of mCRL2 and adopted a limited form of recursion. In this paper, by contrast, we do have data and a more general treatment of recursion. As a consequence, in addition to new proofs for new results, we necessarily revised, extended, and sometimes simplified many of our old proofs.

2. A Process Algebra with Multi-actions and Data

The process algebra used in this paper is the untimed subset of mCRL2 [14, 16], a specification language based on ACP [6] and the basis of the process algebraic semantics of Reo. Among other useful constructs, mCRL2 has one feature that makes it particularly well-suited as a semantic formalism for Reo, namely multi-actions: collections of actions that occur at the same time. We postpone an explanation of how to use multi-actions for describing the behavior of connectors until Section 3. In this section, we summarize the untimed subset of mCRL2.

2.1. Data

Before discussing the syntax and semantics of processes, we first give a terse overview of the data language of mCRL2, used to parameterize actions in the algebra (details appear elsewhere [14]). This data language, based on higher-order abstract data types, allows for the definition of sorts. Every sort consists of constructors and maps, which compose into data expressions. Every data expression can be interpreted as
sort \( \mathbb{B} \)
\( \text{cons} \ true : \mathbb{B}, \ false : \mathbb{B} \)
map \( : \mathbb{B} \rightarrow \mathbb{B}, \ \land : \mathbb{B} \times \mathbb{B} \rightarrow \mathbb{B}, \ldots \)
var \( b \)
eqn \( \neg \text{true} = \text{false}, \ \neg \text{false} = \text{true}, \ \neg \neg b = b, \ b \land \text{true} = b, \ b \land \text{false} = \text{false}, \ \text{true} \land b = b, \ \text{false} \land b = \text{false}, \ldots \)

Figure 2: Partial definition of sort \( \mathbb{B} \).

\[
\begin{align*}
\mathbf{a} & ::= \text{any action in } \mathcal{A} \\
\alpha & ::= \mathbf{a}(\mathbf{d}) \mid \tau \mid \alpha \sqcup \alpha \\
\mathbf{a} & ::= \alpha \mid \delta
\end{align*}
\]
(a) Multiactions and deadlock.

\[
\begin{align*}
p & ::= \mathbf{a} \mid P(\mathbf{d}) \mid p + p \mid p \circ p \mid c \rightarrow p \circ p \mid \sum_{\mathbf{d} \in \mathbf{D}} p \\
& \quad \mid p \parallel p \mid p \parallel q \mid p \parallel p \\
& \quad \mid \nabla_V(p) \mid \partial_B(p) \mid \rho_R(p) \mid \Gamma_C(p) \mid T_I(p)
\end{align*}
\]
(b) Processes.

Figure 3: Syntax.

a **data element** of a sort. **Equations**, possibly containing **data variables** (over data expressions), enable one to derive equalities between data expressions (by giving meaning to maps). For example, Figure 2 shows a fragment of the definition of mCRL2’s built-in sort \( \mathbb{B} \) [16], which represents the booleans. Additionally, mCRL2’s collection of built-in sorts includes the natural numbers (\( \mathbb{N} \)) and the real numbers (\( \mathbb{R} \)). Users of mCRL2 can also define their own sorts.

Every sort \( S \) has, among other standard maps, a map \( \approx : S \times S \rightarrow \mathbb{B} \) for equality of data expressions of sort \( S \). For the built-in sorts, this map behaves as expected. For user-defined sorts, the user must provide equations that give meaning to \( \approx \).

Henceforth, let \( c \) range over data expressions of sort \( \mathbb{B} \), let \( d, e, f \) range over arbitrary data expressions, and let \( D, E, F \) range over such sets. Likewise, let \( d, e, f \) range over tuples of data expressions and data variables, and let \( D, E, F \) range over tuples of such sets. Finally, let \( x, y, z \) range over data variables, let \( X, Y, Z \) range over such sets, and let \( x, y, z \) range over such tuples. Furthermore:

**Definition 1.** \( \mathbb{E} \text{lem} \) denotes a global set of and \( \mathbb{V} \text{ar} \) denotes a global set of data variables such that \( \mathbb{E} \text{lem} \cap \mathbb{V} \text{ar} = \emptyset \).

2.2. **Syntax**

Figure 3a shows the syntax of multiactions and deadlock. Let \( \mathcal{A} \) denote a global set of actions, ranged over by \( a, b, c \) (henceforth, whether \( c \) denotes an action or a data expression of sort \( \mathbb{B} \) is always clear from the context). Actions can involve data, specified using the data language from Section 2.1. Note that data variables can occur in the parameter of \( \mathcal{A} \). The distinguished symbol \( \tau \) denotes the empty multiaction, which consists of no observable actions. Operator \( \sqcup \) (associative and commutative) composes multiactions into larger multiactions; let \( \mathcal{M} \mathcal{A} \mathcal{C} \) denote the global set of all multiactions, ranged over by \( \alpha, \beta, \gamma \). The distinguished symbol \( \delta \) denotes the deadlock process, which performs no multiactions; let \( \mathbf{a}, \mathbf{b}, \mathbf{c} \) range over the processes in the set \( \mathcal{M} \mathcal{A} \mathcal{C} \cup \{ \delta \} \).

Figure 3b shows the syntax of processes. Parameterized process references, ranged over by \( P(\mathbf{d}), Q(e), R(f) \), refer to process definitions of the form \( P(x : D) = p \), where \( p \) denotes some process: the process reference \( P(\mathbf{d}) \) behaves as the process resulting from substituting the occurrences of the data variables \( x \) with the data expressions \( \mathbf{d} \) in \( p \), denoted by \( p[\mathbf{d}/x] \). Processes, ranged over by \( p, q, r \), consist of multiactions and process references, composed with a variety of operators as follows.

**Basic operators** Operator + and \( \cdot \) denote alternative and sequential composition in the usual way. Ternary operator \( \rightarrow \) denotes alternative and sequential composition in the usual way. Ternary operator \( \rightarrow \) composes processes into a conditional choice: the process \( c \rightarrow q \circ r \) behaves as \( q \) if the data expression \( c \) equals \( \text{true} \) (in terms of \( \approx \)) and as \( r \) otherwise. Operator \( \sum \) binds, for each
Parallel operators
Operator $\parallel$ interleaves and synchronizes processes. Operator $\parallel$ behaves as $\parallel$, but the first computation step must come from its left-hand argument. Similarly, operator $\cdot$ behaves as $\cdot$, but the first computation step is formed by synchronizing the first multiaction of each of its arguments.

Additional operators
Four additional operators constrain the behavior of processes composed in parallel.

Operator $\Box$ restricts a process $p$ to the multiactions in a set of nonempty multiactions $\mathcal{V} \subseteq \mathcal{MA}\mathcal{C} \setminus \{\text{unary operators}\}$ (modulo commutativity and associativity of $\Box$). Operator $\rho$ blocks those actions in a process $p$ that occur also in a set of actions $\mathcal{B} \subseteq \mathcal{A}\mathcal{C}$. Operator $\rho$ renames the actions in a process $p$ according to a set of renaming rules $\mathcal{R} \subseteq \mathcal{A}\mathcal{C} \times \mathcal{A}\mathcal{C}$. Finally, operator $\Gamma$ applies the communication rules in a set $\mathcal{C} \subseteq \mathcal{M}\mathcal{A}\mathcal{C} \times \mathcal{A}\mathcal{C}$ to a process $p$. We write communication rules as $\alpha \rightarrow a$ and require that $\tau$ does not occur in $\alpha$.

Abstraction operator
Operator $\mathcal{T}$ hides those actions in a process $p$ that occur also in a set of actions $\mathcal{I} \subseteq \mathcal{A}\mathcal{C}$. The act of hiding an action $a$, which means “replacing $a$ by $\tau$,” differs from the act of blocking $a$, which means “replacing $a$ by $\delta$.”

We adopt the following usual operator precedence (in decreasing order): $\cup, |, \cdot, \parallel, \cdot, +$. We write as few parentheses as possible, omitting them also in the case of associative or commutative operators. For example, we write $p \cdot q \cdot r + \alpha + \beta$ instead of $(p \cdot (q \cdot r)) + (\alpha + \beta)$. Furthermore, let symbol $\oplus$ range over the binary operators $+, \cdot, \parallel, \parallel, \|$ and $|$. Similarly, let symbol $f$ range over unary operators $\Box, \rho, \Gamma$, and $\mathcal{T}$.

<table>
<thead>
<tr>
<th>Bound($\alpha$) = $\emptyset$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bound($q + r$), Bound($q \cdot r$), Bound($c \rightarrow q \cdot r$) = Bound($q$) $\cup$ Bound($r$)</td>
</tr>
<tr>
<td>Bound($\sum_{x \in D} q$) = Bound($q$) $\cup$ ${x}$</td>
</tr>
</tbody>
</table>

Figure 4: Definition of Bound.

<table>
<thead>
<tr>
<th>$\alpha \in$ Basic</th>
</tr>
</thead>
<tbody>
<tr>
<td>$q + r$, $q \cdot r$, $c \rightarrow q \cdot r \in$ Basic iff $q$, $r \in$ Basic</td>
</tr>
<tr>
<td>$\sum_{x \in D} q \in$ Basic iff ${q \in$ Basic and $x \notin$ Bound($q)}$</td>
</tr>
</tbody>
</table>

Figure 5: Definition of Basic.

data element in a finite set, a data variable in a process to that particular element and places the resulting processes in an alternative composition: the process $\sum_{x \in \{d_1, \ldots, d_t\}} q$, with $x \in \mathcal{V}$ and $d_1$, $\ldots$, $d_t \in \mathcal{E}$, behaves as $q[d_1/x] + \cdots + q[d_t/x]$ (shortly, we shall state this more explicitly in a proposition).

We associate with every process $p$ built from the operators discussed so far a set $\text{Bound}(p)$ (defined in Figure 4), which contains the data variables bound by occurrences of $\sum$ in $p$. Furthermore, let $\text{Basic}$ (defined in Figure 5) denote the set of basic processes, which consist of only multiactions and the basic operators such that nested occurrences of $\sum$ bind different data variables. The latter restriction, imposed for technical convenience, does not really limit the expressiveness of the algebra, because one can always bring a process to the desired format by applying alpha-conversion (i.e., we consider processes up to alpha-conversion for summation).

(Full mCRL2 contains also the $\text{at}$ basic operator and the initialization basic operator for expressing timed behavior. We skip those operators here, because we use only the untimed subset of mCRL2 in this paper.)
Figure 6: Axioms.
2.3. Semantics

Every process has an associated transition system describing its semantics (SOS rules appear in [14]). Let \( \simeq \) denote processes. Figure 6 shows a sound axiomatization for strong bisimulation of the operators shown in Figure 3 [14]. Let function Free (defined in Figure 7), which occurs in axioms SUM1, SUM2, and SUM5, map processes to the free data variables occurring in them. Note that Figure 6 axiomatizes three additional operators on multiactions: operator \( \setminus \) subtracts the multiaction on its right-hand side from the multiaction on its left-hand side; operator \( \sqsubseteq \) checks if the multiaction on its right-hand side contains the multiaction on its left-hand side; operator \( \_ \) clears a multiaction from data parameters. These three additional operators occur in the definition of the auxiliary function \( C \), used in Axiom C1:

\[
C_C(\alpha) = \begin{cases} 
C_{C_1}(C_{C_2}(\alpha)) & \text{if } C = C_1 \cup C_2 \text{ and } C_1 \cap C_2 = \emptyset \text{ and } C_1, C_2 \neq \emptyset \\
\emptyset & \text{otherwise} \\
\end{cases}
\]

Informally, \( C \) applies the communication rules in a set \( C \) to a multiaction \( \alpha \).

Although we use only a subset of the axioms in Figure 6 in proofs, we show all of them for completeness.

The proof of one of the theorems in Section 5 relies on the recursive specification principle (RSP) [7]. This principle states that every guarded recursive definition has at most one solution. One can formulate this principle in terms of a guarded process operator \( \Phi \)—a function from processes to processes—as follows [16]:

\[
P \simeq \Phi(P) \text{ and } Q \simeq \Phi(Q) \implies P \simeq Q
\]

Thus, if \( \Phi \) has both \( P \) and \( Q \) as fixed points, \( P \) must be provably equal to \( Q \).

Finally, we introduce a “metalevel” operator \( \_ \cup \) to abbreviate arbitrary finite sequences of multiactions composed together: let \( \bigcup_{i=1}^n \alpha_i \) abbreviate the multiaction \( \alpha_1 \cup \cdots \cup \alpha_n \) and identify \( \bigcup_{i=1}^0 \alpha_i \) with \( \tau \). Similarly, we introduce a metalevel operator \( \sum \) (same symbol as the summation operator but with a different, yet related, meaning) to abbreviate alternative compositions consisting of a finite number of processes: let \( \sum_{i=1}^n p_i \) abbreviate the process \( p_1 + \cdots + p_n \) and identify \( \sum_{i=1}^0 p_i \) with \( \delta \). Operators \( \_ \cup \) and \( \sum \) help us in formulating propositions and proofs more concisely. Although strictly different, the latter has a tight connection with the summation operator. The following proposition makes this connection precise.

**Proposition 1 ([16, Section 4.6]).** \( \sum_{x \in \{d_1, \ldots, d_t\}} q \simeq \sum_{i=1}^t q(d_i/x) \)
Every channel has exactly two ends, each of which has one of two types: source ends accept data, while sink ends dispense data. Besides this assumption on the number of ends, Reo makes no assumptions about channels. This means, for example, that Reo allows channels with two source ends.

In the process algebraic semantics of Reo, one associates every channel end with an action. For source ends, represented by the multiaction $\langle a \rangle$, atoms accept data on their source end. For sink ends, represented by the multiaction $\langle b \rangle$, atoms dispense data. Besides this assumption on the number of ends, Reo makes no assumptions about channels. This means, for example, that Reo allows channels with two source ends.

In the process algebraic semantics of Reo, one associates every channel end with an action. For source ends, represented by the multiaction $\langle a \rangle$, atoms accept data on their source end. For sink ends, represented by the multiaction $\langle b \rangle$, atoms dispense data. Besides this assumption on the number of ends, Reo makes no assumptions about channels. This means, for example, that Reo allows channels with two source ends.

In process references, in contrast to the textual syntax in Figure 8, angle brackets have no meaning and give no structure.
the following semantics. which only sink ends coincide, and \( R \) nodes by combining the processes for a binary replicator action

Nodes. Entities communicating through a connector perform I/O operations—writes and takes—on its nodes. Reo features three kinds of nodes: source nodes on which only source ends coincide, \( \text{sink nodes} \) on which only sink ends coincide, and \( \text{mixed nodes} \) on which both kinds of channel end coincide. Nodes have the following semantics.

- A source node \( n \) has replicator semantics. Once an entity attempts to write a data item \( d \) on \( n \), this node first suspends this operation. Subsequently, \( n \) notifies the channels whose source ends coincide on \( n \) that it offers \( d \). Once each of these channels has notified \( n \) that it accepts \( d \), \( n \) resolves the write: atomically, \( n \) dispenses \( d \) to each of its coincident source ends.

- A sink node \( n \) has nondeterministic merger semantics. Once an entity attempts to take a data item from \( n \), this node first suspends this operation. Subsequently, \( n \) notifies the channels whose sink ends coincide on \( n \) that it accepts a data item. Once at least one of these channels has notified \( n \) that it offers a data item, \( n \) resolves the take: atomically, \( n \) fetches this data item from the appropriate channel end and dispenses it to the entity attempting to take. If multiple sink ends offer a data item, \( n \) chooses one of them nondeterministically.

- A mixed node \( n \) has pumping station semantics, which is a combination of the replicator semantics and merger semantics discussed above, where fetching and dispensing occur atomically.

In the process algebraic semantics of Reo, one associates each of the \( n \) source ends of a node with an action \( \text{src}_i \) (1 ≤ \( i \) ≤ \( m \)) and each of its \( n \) sink ends with an action \( \text{snk}_i \) (1 ≤ \( i \) ≤ \( n \)). Then, one can describe nodes by combining the processes for a binary replicator \( R \) (one sink end to two source ends), a binary merger \( M \) (two sink ends to one source end), a one-to-one pumping station \( PS \), and a boundary node \( B \):

\[
\begin{align*}
R(\text{snk}; \text{src}_1, \text{src}_2) &= \sum_{x \in \text{Data}} \text{snk}(x) \cup \text{src}_1(x) \cup \text{src}_2(x) \cdot R(\text{snk}; \text{src}_1, \text{src}_2) \\
M(\text{snk}_1, \text{snk}_2; \text{src}) &= \sum_{x \in \text{Data}} (\text{snk}_1(x) \cup \text{src}(x) + \text{snk}_2(x) \cup \text{src}(x)) \cdot M(\text{snk}_1, \text{snk}_2; \text{src}) \\
PS(\text{snk}; \text{src}) &= \sum_{x \in \text{Data}} \text{snk}(x) \cup \text{src}(x) \cdot PS(\text{snk}; \text{src}) \\
B(\text{bnd}) &= \sum_{x \in \text{Data}} \text{bnd}(x) \cdot B(\text{bnd})
\end{align*}
\]

Connectors. To get the behavior of a connector as a process, one composes the processes of the constituents of that connector in parallel and synchronizes their actions. Below, we give the processes of the connectors in Figures 1a and 1c. More examples may be found in 24, 25, 26, 27.

\[
\begin{align*}
\text{Fig1a} &= \partial_{\{a_1, a_2, a_3, a_4, a_5, a_6\}} (\Gamma_{\{a_1 \cup a_2 \cup a_3 \cup a_4 \cup a_5 \cup a_6\} \rightarrow a_1 \cup a_2 \cup a_3 \cup a_4 \cup a_5 \cup a_6} (q)) \\
\text{Fig1c} &= \partial_{\{a_1, b, c\} \cap \{1, 2, 3\}} (\Gamma_{\{a_1 \cup a_2 \cup a_3 \cup a_4 \cup a_5 \cup a_6 \} \rightarrow \{a, b, c\}} (r))
\end{align*}
\]

\footnote{An extensive overview of context-(in)sensitive semantic formalisms for Reo appears in 13.}
Figure 9: Labeled transition system(s) of the process(es) modeling the connector in Figure 1a. On the left is a graphical representation of the decomposition of that connector into channels and nodes with labeled ends. In the middle are the labeled transition systems of the processes modeling those channels and nodes (without data for simplicity). On the right is the labeled transition system of the parallel composition of those processes (after applying communication and blocking).

For:

\[
q = \left( B(\tilde{a}_1) \parallel \text{Fifo1}(a_1;x_1) \parallel \text{PS}(\tilde{x}_1;x_2) \right) \parallel \text{Fifo1}(x_2;b_1) \parallel B(b_1)
\]

\[
r = \left( B(a_1) \parallel R(\tilde{a}_1;\tilde{a}_2;\tilde{a}_3) \parallel \text{Sync}(a_3;c_1) \parallel M(\tilde{c}_1;\tilde{c}_2;\tilde{c}_3) \parallel B(c_3) \right) \parallel 
\]

\[
B(b_1) \parallel R(b_1;b_2;b_3)
\]

4. Splitting Processes

Recall from Section 1 that we originally aimed at establishing the validity of optimizing implementations of Reo through the identification of (a)synchronous regions. Essentially, we want to show that splitting connectors along the boundaries of their (a)synchronous regions (and running the resulting subconnectors concurrently) neither loses behavior nor gives rise to inadmissible behavior. In this section, we lay the foundation for this kind of splitting in terms of a splitting procedure for processes. Later, in Section 6, we apply this procedure to the process algebraic semantics of Reo, thereby justifying the splitting of connectors. Here, in Section 4.1, we start by explaining the intuition behind our splitting procedure; formal definitions appear in Section 4.2. In Section 5, we investigate and prove properties of our splitting procedure, including a proof of correctness. We note that our notion of “splitting processes” differs from “uniquely decomposing processes” [32]: in our context, neither primality nor uniqueness of processes matters. We discuss the differences in more detail in Section 7.

4.1. Intuition

For simplicity, to convey the intuition behind our splitting procedure, we consider only data-free processes in this subsection (definitions in Section 4.2 do incorporate data).

Let \textit{Act}(p) (defined in Figure 10) denote the set of actions syntactically occurring in a process \( p \). We introduce function \textit{split}, which splits a process \( p \) along a set of actions \( A \) into two processes: one of these
processes contains no actions in $\text{Act}(p) \setminus \mathbb{A}$, while the other process contains no actions in $\mathbb{A}$. We call the former process the $\mathbb{A}$-isolation of $p$ and the latter process the $\mathbb{A}$-coisolation of $p$. We aim at constructing $p$’s isolation and its coisolation such that their parallel composition behaves as $p$ under some appropriate notion of synchronization (defined shortly).

Informally, to construct $p$’s $\mathbb{A}$-isolation, replace every action in $p$ as follows:

- If $a \in \mathbb{A}$, replace $a$ with the multiact $a \sqcup \xi(a)$, where $\xi(a)$ denotes a fresh auxiliary action with respect to $\text{Act}(p)$. Intuitively, $\xi(a)$ represents the act of “disseminating that this process performs $a$.”

- If $b \notin \mathbb{A}$, replace $b$ with the auxiliary action $\overline{\xi}(b)$, where $\overline{\xi}(b)$ denotes a fresh action with respect to $\text{Act}(p)$. Intuitively, $\overline{\xi}(b)$ represents the act of “discovering that another process performs $b$.”

Symmetrically, to construct the $\mathbb{A}$-coisolation of a process $p$, replace in $p$ every $b \in \mathbb{A}$ with $\overline{\xi}(b)$ and every $b \notin \mathbb{A}$ with $b \sqcup \xi(b)$. Note that because the foregoing affects only multiactions, $p$’s isolation and its coisolation have the same syntactic structure as $p$. In other words: the process $p$, its isolation, and its coisolation have the same transition system modulo transition labels.

To illustrate isolation and coisolation, consider an example process $q = a \cdot b$. This process has $q_1 = a \sqcup \xi(a) \cdot \overline{\xi}(b)$ as its $\{a\}$-isolation and $q_2 = \overline{\xi}(a) \cdot b \sqcup \xi(b)$ as its $\{a\}$-coisolation. The parallel composition of $q_1$ and $q_2$, however, does not behave as $q$ yet: to ensure that a process behaves as the parallel composition of its isolation and its coisolation, these two processes should appropriately synchronize on $\xi(a)$ and $\overline{\xi}(a)$ for each $a$. To this end, we apply the communication operator $\Gamma$ to such compositions. In our example, this yields the process $\Gamma_C(q_1 \parallel q_2)$ for $C = \{\xi(a) \sqcup \overline{\xi}(a) \rightarrow \text{tau}, \xi(b) \sqcup \overline{\xi}(b) \rightarrow \text{tau}\}$. The special action $\text{tau}$ serves as a placeholder action for $\tau$, and we can hide it immediately using the abstraction operator $\rho^\Gamma$ henceforth, without loss of generality, we assume $\text{tau} \notin \text{Act}(p)$ for each $p$. In our example, this yields the process $\tau(f_I(\Gamma_C(q_1 \parallel q_2)))$ with $I = \{\text{tau}\}$ and $C$ as before.

But also this process does not behave as $q$ yet: synchronization and abstraction alone do not suffice—we must also block those auxiliary actions whose individual performance “makes no sense.” For instance, we consider every unpaired occurrence of $\overline{\xi}(a)$ in a multiact $\alpha$ nonsensical: intuitively, performing $\overline{\xi}(a)$ suggests that some process discovers that another process performs $a$, even though this does not happen (otherwise, also $\xi(a)$ would occur in $\alpha$). By symmetry, we consider also every unpaired occurrence of $\xi(a)$ nonsensical. To block unpaired occurrences of $\xi(a)$ and $\overline{\xi}(a)$, we apply the blocking operator $\partial$. In our example, this yields the process $\partial_B(f_I(\Gamma_C(q_1 \parallel q_2)))$ with $B = \{\xi(a), \overline{\xi}(a), \xi(b), \overline{\xi}(b)\}$ and $I$ and $C$ as before. This process behaves as $q$, concluding our example.

We proceed with general formal definitions of the splitting procedure just outlined.

### 4.2. Formal Definitions

**Auxiliary actions and substitution environments.** We start with a formal account of the fresh auxiliary actions of the form $\xi(a)$ and $\overline{\xi}(a)$. As suggested by this notation, $\xi$ and $\overline{\xi}$ denote functions that take an action $a$ as input and produce another action as output. We collect such pairs of functions in substitution environments as follows. Let $C^*$ denote the set of finite strings over $C$.

**Definition 2.** 1 and 2 are global symbols such that $1 \neq 2$ and $1, 2 \notin \text{Elem} \cup \forall \text{Var}$.

\begin{center}
\begin{tabular}{|l|}
\hline
$\text{Act}(a(d)) = \{a\}$

$\text{Act}(\tau), \text{Act}(\delta), P(d) = \emptyset$

$\text{Act}(\beta \sqcup \gamma) = \text{Act}(\beta) \cup \text{Act}(\gamma)$

$\text{Act}(q \sqcup r), \text{Act}(c \rightarrow q \circ r) = \text{Act}(q) \cup \text{Act}(r)$

$\text{Act}(\sum_{x \in D} q), \text{Act}(f(q)) = \text{Act}(q)$

\hline
\end{tabular}
\end{center}

Figure 10: Definition of $\text{Act}$.
To formalize the notions of isolation and coisolation.

sets, propositions, lemmas, theorems, and proofs, without mentioning them explicitly. We do the same for

Example.

comm on actions) and image. Function

isolations

substitution environment. Functions

for simplicity. In the general case, however, this information plays a vital role, as explained shortly.

in the example in Section 4.1: because we did not need such an extra string of information, we omitted it

A

take a string over \( \{1, 2\} \cup \text{Elem} \cup \text{Var} \) and a basic process as input.\(^{10}\)

\[\begin{align*}
\text{dom}(\Xi) & = \{a \mid \langle w, a \rangle \in \text{dom}(\xi) \cap \text{dom}(\xi')\} \\
\text{img}(\Xi) & = \text{img}(\xi) \cup \text{img}(\xi') \\
\text{comm}(\Xi) & = \{\xi_w(a) \cup \xi'_w(a) \rightarrow \text{tau} \mid \langle w, a \rangle \in \text{dom}(\xi) \cap \text{dom}(\xi')\}
\end{align*}\]

Figure 11: Definitions of \(\text{dom}, \text{img}, \) and \(\text{comm}\).

\[
\begin{align*}
\text{isol}(w, a(d)) & = a(d) \cup \xi_w(a)(w^b) & \text{if } a \in A \\
\text{isol}(w, b(e)) & = \xi'_w(b)(w^b) & \text{if } b \notin A \\
\hat{\text{isol}}(w, a(d)) & = \hat{\xi}_w(a)(w^b) & \text{if } a \in A \\
\hat{\text{isol}}(w, b(e)) & = b(e) \cup \hat{\xi}_w(b)(w^b) & \text{if } b \notin A \\
\hat{\text{isol}}(w, \tau) & = \tau \\
\hat{\text{isol}}(w, \beta \sqcup \gamma) & = \hat{\text{isol}}(w, \beta) \sqcup \hat{\text{isol}}(w, \gamma) \\
\hat{\text{isol}}(w, \delta) & = \delta \\
\hat{\text{isol}}(w, q + r) & = \hat{\text{isol}}(w, q) + \hat{\text{isol}}(w, r) \\
\hat{\text{isol}}(w, q \cdot r) & = \hat{\text{isol}}(w, q) \cdot \hat{\text{isol}}(w, r) \\
\hat{\text{isol}}(w, c \rightarrow q \circ r) & = c \rightarrow \hat{\text{isol}}(w, q) \circ \hat{\text{isol}}(w, r) \\
\hat{\text{isol}}(w, \sum_{x \in D} q) & = \sum_{x \in D} \hat{\text{isol}}(w, x) q
\end{align*}\]

Figure 12: Definitions of \(\text{isol}\) and \(\hat{\text{isol}}\). Let \(\hat{\text{isol}}\) range over the set \(\{\text{isol}, \hat{\text{isol}}\}\).

\[\begin{align*}
1^\beta & = 1 \\
2^\beta & = 2 \\
\beta^\beta & = \epsilon \\
(wv)^\beta & = w^\beta v^\beta
\end{align*}\]

Figure 13: Definition of \(\beta\).

\[\begin{align*}
1^\beta & = 1 \\
2^\beta & = 2 \\
\beta^\beta & = \epsilon \\
(wv)^\beta & = w^\beta v^\beta
\end{align*}\]

Figure 14: Definition of \(\beta\).

**Definition 3.** A substitution environment, typically denoted by \(\Xi\), is a quadruple \(\langle A, \text{tau}, \xi, \xi'\rangle\) consisting of a set \(A \subseteq \text{Act}\), an action \(\text{tau} \in \text{Act} \setminus A\) and injective functions \(\xi, \xi' : \{1, 2\}^* \times A \rightarrow \text{Act} \setminus (A \cup \{\text{tau}\})\) such that \(\text{img}(\xi) \cap \text{img}(\xi') = \emptyset\).

**Example.**

Henceforth, we write \(\xi_w(a)\) and \(\xi'_w(a)\) instead of \(\xi(w, a)\) and \(\xi'(w, a)\). Note that we dropped the \(w\) subscripts in the example in Section 4.1 because we did not need such an extra string of information, we omitted it for simplicity. In the general case, however, this information plays a vital role, as explained shortly.

Let “dom” and “img” map functions to their domain and image. Figure 11 shows auxiliary functions for substitution environment. Functions \(\text{dom} \) and \(\text{img} \) map substitution environments to their domain (projected on actions) and image. Function \(\text{comm}\) maps substitution environments to communications derivable from them.

**Example.**

Henceforth, to avoid heavy notation, we quantify implicitly over all substitution environments in definitions, propositions, lemmas, theorems, and proofs, without mentioning them explicitly. We do the same for sets \(A\), which contain the actions along which we split processes.

**Isolation and coisolation.** To formalize the notions of \(A\)-isolation and \(A\)-coisolation, we introduce the functions \(\text{isol}\) and \(\hat{\text{isol}}\), ranged over by \(\text{isol}\). Figure 12 shows their definitions. (Recall that we quantify implicitly over all execution environments \(\Xi\) and sets \(A\) without mentioning them explicitly.) Functions \(\text{isol}\) and \(\hat{\text{isol}}\) take a string over \(\{1, 2\} \cup \text{Elem} \cup \text{Var}\) and a basic process as input.\(^{10}\)

\(^{10}\)Strictly speaking, \(\text{isol}\) and \(\hat{\text{isol}}\) also take a substitution environment and a set of actions \(A\) as input.
Before we take a closer look at Figure 12, we explain the purpose of the string over \(\{1, 2\} \cup \text{Elem} \cup \text{Var}\). Essentially, such strings encode information that \(\text{isol}\) and \(\text{isol}\) use to “keep track” of each other’s nondeterministic or data-dependent choices. If they cannot do that, an isolated process and its coisolation run the risk of going “out of sync.” To clarify this, suppose that we want to compose the \(\{a\}\)-isolation and \(\{a\}\)-coisolation of the process \(r = a \cdot b + a \cdot c\) in parallel. For the sake of argument, suppose that \(\text{isol}\) and \(\text{isol}\) take only a basic process as input and no string. We now demonstrate that this can go wrong. We have:

\[
\text{isol}(r) = \frac{a \cup \xi(a) \cdot \bar{\xi}(b) + a \cup \xi(c) \cdot \bar{\xi}(c)}{\text{isol}(r) = \xi(a) \cdot b \cup \xi(b) + \xi(a) \cdot c \cup \xi(c)}
\]

This means that \(\text{isol}(r)\) can erroneously synchronize its left-most multiaction \(a \cup \xi(a)\) with the right-most multiaction \(\bar{\xi}(a)\) of \(\text{isol}(r)\), causing deadlock afterwards (because \(\bar{\xi}(b)\) cannot synchronize with \(\xi(c)\)). To solve this problem, we use strings over \(\{1, 2\} \cup \text{Elem} \cup \text{Var}\): essentially, we associate with every branch of the parse tree of a process a unique such string. This string encodes information about the structure of that process and its data bindings. Moreover, we ensure (e.g., by defining \(\xi\) and \(\bar{\xi}\) as injective functions) that the isolation and the coisolation of a process synchronize auxiliary actions only if they belong to the same branch (in which case they have matching strings). For example:

\[
\text{isol}(\epsilon, r) = a \cup \xi_{11}(a) \cdot \bar{\xi}_{12}(b) + a \cup \xi_{21}(a) \cdot \bar{\xi}_{22}(c)
\]

\[
\text{isol}(\epsilon, r) = \frac{\xi_{11}(a) \cdot b \cup \xi_{12}(b) + \xi_{21}(a) \cdot c \cup \xi_{22}(c)}{}
\]

In this case, assuming some appropriate notion of synchronization that takes strings into account (we define this shortly), \(a \cup \xi_{11}(a)\) can synchronize only with \(\bar{\xi}_{12}(a)\) (they share the same string) and not with \(\bar{\xi}_{21}(a)\) (different string). And so, these two processes do not go out of sync.

Let us now have a closer look at Figure 12. Applied to a string \(w\) and a single action \(a(d)\), depending on whether \(A\) contains \(a\), \(\text{isol}\) and \(\text{isol}\) either compose or replace \(a(d)\) with an auxiliary action using the substitution functions \(\xi\) and \(\bar{\xi}\). However, because \(\xi\) and \(\bar{\xi}\) have \(\{1, 2\}^* \times A\) as domain (see Definition 3), \(\text{isol}\) and \(\text{isol}\) cannot directly use \(w\) in \(\xi\) or \(\bar{\xi}\): because \(w \in \{1, 2\} \cup \text{Elem} \cup \text{Var}\)*, \(\text{isol}\) and \(\text{isol}\) should first filter out the data elements and data variables possibly occurring in \(w\). We introduce an operator denoted by \(\sharp\) for that purpose. Figure 13 shows its definition. Similarly, we introduce an operator denoted by \(\flat\), which does the converse of \(\sharp\): it filters symbols 1 and 2 from a string over \(\{1, 2\} \cup \text{Elem} \cup \text{Var}\). Figure 14 shows its definition. Functions \(\text{isol}\) and \(\text{isol}\) use \(\flat\) to parameterize auxiliary actions with data. This parameterization ensures that the isolation and coisolation of a process of the form \(\sum_{x \in E} q\) do not go out of sync (similar to what we saw in the example above). In Section 4.3, we exemplify this further.

Applied to a composite multiaction \(\beta \cup \gamma\), \(\text{isol}\) and \(\text{isol}\) apply themselves recursively on \(\beta\) and \(\gamma\) without changing \(w\). This differs for processes with a different main composition operator. For instance, for processes of the form \(p + q\), \(\text{isol}\) and \(\text{isol}\) apply themselves recursively on \(w+1\) and \(w2\) instead of \(w\). This ensures that in their parallel composition, if appropriately synchronized, the process \(\text{isol}(w, p + q)\) can track which choice the process \(\sum_{x \in E} q\) makes and vice versa as outlined above.

We make a final remark about the practical computability of \(\text{isol}\) and \(\text{isol}\). Strictly speaking, because we defined \(\xi\) and \(\bar{\xi}\) as functions over \(\{1, 2\}^*\), those functions have infinite domains. This may seem problematic in practice, but fortunately, one can easily fix this. Start by observing that process terms consist of only finitely many operators and actions. This means that for \(\text{isol}(w, p)\) and \(\text{isol}(w, p)\) to be defined (for some \(w\) and \(p\)), functions \(\xi\) and \(\bar{\xi}\) must be defined for only finitely many strings (all of which have \(w\) as a prefix).

One can compute this set of strings \(W\) in a preprocessing step that analyzes the syntax of \(p\) (essentially a dry run of \(\text{isol}\) or \(\text{isol}\)). Then, before actually applying \(\text{isol}\) or \(\text{isol}\), define a finite substitution environment \(\Xi\) such that the domains of \(\xi\) and \(\bar{\xi}\) contain only the strings in \(W\). Thus, rather than one general substitution environment for all processes, we have a tailored substitution environment for every individual process (this generalizes straightforwardly to finite collections of processes). Henceforth, we always assume a finite yet sufficient substitution environment \(\xi\) when we apply \(\text{isol}\) or \(\text{isol}\) to a (collection of) process(es).

\textbf{Splitting.} We build the definition of function \(\text{split}\)—the actual splitting procedure, so to speak—on top of functions \(\text{isol}\) and \(\text{isol}\). Figure 15 shows its definition.
We also introduce an auxiliary operator, denoted by $\dagger$, which represents and ensures “appropriate synchronization” among auxiliary actions: it takes care of the communication, hiding, and blocking necessary to synchronize auxiliary actions such that split preserves semantics (as exemplified in Section 4.1). Recall that we implicitly quantify universally over all substitution environments $\Xi$ in definitions to avoid heavy notation. Then $\dagger$

**Definition 4.** \( ?(p) = \partial_{\text{msg}(\Xi)}(T_{\text{tau}}(\Gamma_{\text{comm}(\Xi)}(p))) \)

**Example.**

The definition of split($w, p$) for $p = P$ may seem odd and requires more explanation, because we make a number of tacit assumptions. First, we assume that if a process reference $R$ occurs in some process $q$, there exists also a process definition $R = r$ (otherwise, $q$ has no meaning). Second, we adopt the notational convention that every process reference with a superscript \( \dagger \) refers to a process definition with a body to which we applied split (for the empty string). For example, $R^\dagger = \text{split}(\epsilon, r)$. Now, the definition of split($w, P$) makes more sense: it means that we replace process references in a split process with process references that refer to other split processes. In that way, the application of split propagates through process definitions. In Section 5.4, we prove the correctness of this definition.

**4.3. More Examples**

To illustrate the usage of split, we give three more examples in this subsection. For the sake of clarity, we use concrete action names for both original actions and auxiliary actions as follows. Define:

\[
\mathcal{A}_{\text{ct}} = \{\text{foo}, \text{bar}, \text{baz}\} \cup \{x_{\text{foo}}, x_{\text{bar}}, x_{\text{baz}}\}
\]

Original actions

\[
\mathcal{A}_{\text{act}} = \{x_{\text{foo}}, x_{\text{bar}}, x_{\text{baz}}\}
\]

Auxiliary actions

Furthermore, let $A = \{\text{foo}, \text{baz}\}$ (i.e., we split along foo and baz), define $\xi$ as follows:

\[
\xi_{1}(\text{foo}) = x_{\text{foo}}, \quad \xi_{2}(\text{bar}) = x_{\text{bar}}, \quad \xi_{21}(\text{bar}) = x_{\text{bar}}, 21
\]

and define $\overline{\xi}$ analogously.

**Example 1.** Let $p_{1} = \text{foo}(1, 2) + \text{bar}(3)$. We derive split($\epsilon, p_{1}$) as follows.

\[
\text{split}(\epsilon, p_{1}) = ?(\text{isol}(\epsilon, \text{foo}(1, 2), \text{bar}(3)) \parallel \text{isol}(\epsilon, \text{foo}(1, 2), \text{bar}(3)))
\]

\[
= ?(\text{isol}(\epsilon, \text{foo}(1, 2) \parallel \text{bar}(3)) \parallel \text{isol}(\epsilon, \text{foo}(1, 2) \parallel \text{bar}(3)))
\]

\[
= ?((\text{foo}(1, 2) \parallel \xi_{1}(\text{foo}) + \xi_{2}(\text{bar})) \parallel (\xi_{1}(\text{foo}) + \text{bar}(3) \parallel \xi_{2}(\text{bar})))
\]

\[
= ?((\text{foo}(1, 2) \parallel x_{\text{foo}}, 1 + x_{\text{bar}}, 2) \parallel (x_{\text{foo}}, 1 + \text{bar}(3) \parallel x_{\text{bar}}, 2))
\]

\[
\]
This example demonstrates how the splitting procedure handles data-dependent processes. Furthermore, note that the auxiliary actions $x_{\text{foo.1}}, x_{\text{foo.1}}, x_{\text{bar.2}},$ and $\bar{x}_{\text{bar.2}}$ have no data parameters in this example, because none of the strings to which we apply $\xi$ and $\xi$ contain symbols outside $\{1, 2\}$. (In those case, by the definition of $b$, auxiliary actions have no parameters.) Next, we consider an example in which data do play a role.

Example 2. Let

$$p_2 = \sum_{x \in D_1} x \leq 28 \leftrightarrow \text{foo(true)} \cdot \sum_{y \in D_2} \text{bar}(x, y) \cdot \text{foo(false)}$$

for $D_1 = \{i \mid 6 \leq i \leq 496\}$ and $D_2 = \{1, 2\}$. We derive $\text{split}(\epsilon, p_2)$ as follows.

$$\text{split}(\epsilon, p_2) = \text{split}(\epsilon, \sum_{x \in D_1} x \leq 28 \leftrightarrow \text{foo(true)} \cdot \sum_{y \in D_2} \text{bar}(x, y) \cdot \text{foo(false)})$$

$$= \text{sum}(\epsilon, \sum_{x \in D_1} x \leq 28 \leftrightarrow \text{foo(true)} \cdot \sum_{y \in D_2} \text{bar}(x, y) \cdot \text{foo(false)}) \ || \\
\sum_{x \in D_1} \text{sum}(\epsilon, x \leq 28 \leftrightarrow \text{foo(true)} \cdot \sum_{y \in D_2} \text{bar}(x, y) \cdot \text{foo(false)})$$

This example demonstrates how the splitting procedure handles data-dependent processes. Furthermore, based on this example, we can illustrate an important property guaranteed by the data parameters of auxiliary actions: the isolation of $p_2$ (left/above of $\parallel$) and the coisolation of $p_2$ (right/below of $\parallel$) terminate successfully only if they bind $x$ (and later $y$) to the same value. To see this, suppose that the isolation binds $x$ to 4, while the coisolation binds $x$ to 28 (such that these processes take the same branch of the conditional choice). Then, because the communication operator $\Gamma$ embedded in $\parallel$ requires that communicating actions have the same data parameters (see Section 2.3, $x_{\text{foo.1}}(4)$ and $x_{\text{foo.1}}(28)$ cannot synchronize with each other. This in turn causes deadlock (effect of the blocking operator in $\parallel$). In contrast, both the isolation and the coisolation bind $x$ to 4, the auxiliary actions parameterized by $x$ can synchronize, after which the whole process terminates successfully.

Example 3.

5. Properties of the Splitting Procedure

In this section, we prove the correctness of the definitions presented in Section 2.3, we establish that processes $p$ and $\text{split}(p)$ are provably equal from the axioms of mCRL2 (see Section 2.3). This implies that $p$ and $\text{split}(p)$ behave indistinguishably under any behavioral congruence satisfying those axioms (e.g., strong bisimilarity).
5.1. Simple Properties I: Deadlock Caused by Split Multiactions

In this subsection, we formulate three propositions that state properties about when split multiactions cause deadlock. Essentially, these propositions formalize when the “appropriate synchronization” operator ? blocks auxiliary actions whose individual execution “makes no sense” (see Section 4.1).

Proposition 2 states that every appropriately synchronized lone (co)isolated multiaction ?(isol(w, α)) causes deadlock. In the formulation of the premise, we write α ∈ TauFree (defined in Figure 17) to express that τ does not occur syntactically in α. Variants of this requirement appear in (nearly) all subsequent propositions, lemmas, and theorems. Fortunately, they limit the applicability of our results only marginally, because τ usually does not occur syntactically in processes (but instead results from hiding). The premise


of Proposition \(2\) also ensures that the domain of the substitution environment contains the actions in \(\alpha\); otherwise, \(\text{isol}(w, \alpha)\) has no meaning.

**Proposition 2 (\(\text{isol}\)-multiactions cause deadlock).**

\[
[\alpha \in \text{TauFree} \text{ and } \text{Act}(\alpha) \subseteq \text{dom}(\Xi)] \implies \text{？}(\text{isol}(w, \alpha)) \simeq \delta
\]

To understand why this proposition holds for \(\text{isol} = \text{isol}\) (the \(\overline{\text{isol}}\) case works similar), observe that every isolated multiaction contains at least one auxiliary action \(\xi\) (this follows immediately from the definition of \(\text{isol}\)). Now, reasoning toward a contradiction, suppose that also the dual of \(\xi\) occurs in \(\text{？}(\text{isol}(w, \alpha))\). Then, the content of \(\mathcal{A}\) must have changed between the construction of \(\xi\) and its dual or vice versa (otherwise, \(\text{isol}\) produces either always \(\xi\) or always its dual). But the content of \(\mathcal{A}\) remains constant across applications of \(\text{isol}\), so \(\mathcal{A}\) cannot have changed. Hence, \(\xi\) has no dual in \(\text{？}(\text{isol}(w, \alpha))\). This means that \(\Gamma_{\text{comm}(\Xi)}\) in \(\text{？}\) does not affect \(\xi\) (because the communications in \(\text{comm}(\Xi)\) involve only pairs of an auxiliary action and its dual). Also \(T_{\text{tau}}\) in \(\text{？}\) does not affect \(\xi\) (because auxiliary actions differ from \(\text{tau}\) by Definition \(3\)). This leaves us with \(\delta_{\text{msg}(\Xi)}\), which does affect \(\xi\); it blocks it. The resulting deadlock then propagates through the entire multiaction. See Section \(B\) page \(17\) for a detailed proof.

**Proposition 3** states that the synchronous composition of an isolated multiaction and a coisolated multiaction under different strings over \(\{1, 2\}\) causes deadlock.

**Proposition 3 (Composed \(\text{isol}\)- and \(\text{isol}\)-multiactions cause deadlock, I).**

\[
[\beta, \gamma \in \text{TauFree} \text{ and } \text{Act}(\beta), \text{Act}(\gamma) \subseteq \text{dom}(\Xi) \text{ and } v^\beta \neq u^\gamma] \implies \text{？}(\text{isol}(v, \beta) \mid \text{isol}(u, \gamma)) \simeq \delta
\]

The validity of this proposition crucially depends on the injectivity of substitution functions (see Definition \(3\)). Essentially, this injectivity ensures that the auxiliary actions in \(\text{isol}(v, \beta)\) and \(\text{isol}(u, \gamma)\) come from different pools: \(\text{isol}(v, \beta)\) and \(\text{isol}(u, \gamma)\) have neither auxiliary actions nor their duals in common. Moreover, by similar reasoning as for Proposition \(2\) we can establish that \(\text{isol}(u, \gamma)\) contains an auxiliary action \(\xi\) but not its dual (the same holds for \(\text{isol}(v, \beta)\) but we do not need it). Thus, neither \(\text{isol}(v, \beta)\) nor \(\text{isol}(u, \gamma)\) contains the dual of \(\xi\). Then, again by similar reasoning as for Proposition \(2\), we can establish that \(\Gamma_{\text{comm}(\Xi)}\) and \(T_{\text{tau}}\) in \(\text{？}\) do not affect \(\xi\) while \(\delta_{\text{msg}(\Xi)}\) does. See Section \(B\) page \(18\) for a detailed proof.

**Proposition 4** states that the synchronous composition of an isolated multiaction and a coisolated multiaction under different data causes deadlock.

**Proposition 4 (Composed \(\text{isol}\)- and \(\text{isol}\)-multiactions cause deadlock, II).**

\[
[\beta, \gamma \in \text{TauFree} \text{ and } \text{Act}(\beta), \text{Act}(\gamma) \subseteq \text{dom}(\Xi) \text{ and } e \neq f] \implies \text{？}(\text{isol}(wv, \beta) \mid \text{isol}(wf, \gamma)) \simeq \delta
\]

Although similar to Proposition \(3\), we prove the validity of this proposition rather differently. In Proposition \(3\) (and also in Proposition \(2\)), deadlock occurred due to lone auxiliary actions. But in this case, it can happen that all auxiliary actions occur with their dual (e.g., if \(\beta = \gamma\) and \(v = u\)). Thus, we need a different strategy. To that end, observe that the premise of Proposition \(4\) ensures that the data parameters of an auxiliary action and its dual differ (because \(e \neq f\)). For instance, if \(b \in \mathcal{A}\), we have \(\text{isol}(e, b) = b \uplus \xi_e(b)(e)\) and \(\overline{\text{isol}(f, b)} = \overline{\xi_e(b)(f)}\). Now, even though \(\xi_e(b)(e)\) and \(\overline{\xi_e(b)(f)}\) are duals, \(\Gamma_{\text{comm}(\Xi)}\) in \(\text{？}\) does not affect their composition, because \(e\) and \(f\) differ (see Axiom C1 and the definition of \(\mathcal{C}\) in Section \(23\)). Because also \(T_{\text{tau}}\) in \(\text{？}\) does not affect these auxiliary actions by the same reasoning as before, again, we end up with \(\delta_{\text{msg}(\Xi)}\), which blocks \(\xi_e(b)(e)\) and \(\overline{\xi_e(b)(f)}\). We can generalize this argument to arbitrary multiactions. See Section \(B\) page \(51\) for a detailed proof.
5.2. Simple Properties II: Deadlock Caused by Split Basic Processes

Next, we generalize the propositions in the previous subsection from multi-actions to basic processes. Each of the proofs of these generalizations exploits the observation that for every (co)isolated basic process \( \tilde{\text{isol}}(w, p) \), there exists a provably equal process with the following structure: \( \sum_{i=1}^{n} \tilde{\text{isol}}(w w_i, \alpha_i) + \sum_{i=1}^{n'} \tilde{\text{isol}}(w w'_i, \alpha'_i) \cdot p'_i' \). Essentially, to establish that such processes cause deadlock, it suffices to show that \( \tilde{\text{isol}}(w w_i, \alpha_i) \) and \( \tilde{\text{isol}}(w w'_i, \alpha'_i) \) cause deadlock for all relevant \( i \) (because of Axioms A6 and A7 in Figure 6). One can show this by applying (some of) the propositions from Section 5.1 for each such \( i \).

The premise of each of the following propositions contains a variant of the requirement \( \text{Bound}(p) \cap w = \emptyset \). The \( \cap \) operator denotes the intersection between the elements in a set (e.g., \( w \)). Thus, \( \text{Bound}(p) \cap w = \emptyset \) means that the data variables that will become bound in \( p \) may not intersect with any of the data variables occurring in \( w \). We forbid this, because if a data variable \( x \) occurs in \( w \), this intuitively means that \( x \) already has been bound (due to how \( \text{isol} \) and \( \text{isol} \) build strings). In other words, if \( \text{Bound}(p) \) and the elements in \( w \) intersect, \( p \) rebinds a data variable, which it should not. The requirement \( \text{Bound}(p) \cap w = \emptyset \) has little consequences in practice: typically, \( w = \epsilon \), in which case it holds vacuously. (Moreover, if necessary, one can avoid rebinding with an \( \alpha \)-conversion preprocessing step.)

**Proposition 5 (isol-processes cause deadlock).**

\[
\left[ \begin{array}{c}
p \in \text{Basic} \text{ and } p \in \text{TauFree} \text{ and } \\
\text{Act}(p) \subseteq \text{dom}(\Xi) \text{ and } \text{Bound}(p) \cap w = \emptyset
\end{array} \right] \text{ implies } \tilde{\text{isol}}(w, p) \simeq \delta
\]

See Section [C] page 65 for a detailed proof.

**Proposition 6 (Composed isol- and \( \tilde{\text{isol}} \)-processes cause deadlock, I).**

\[
\left[ \begin{array}{c}
q, r \in \text{Basic} \text{ and } q, r \in \text{TauFree} \text{ and } \\
\text{Act}(q), \text{Act}(r) \subseteq \text{dom}(\Xi) \text{ and } \\
\text{Bound}(q) \cap v = \emptyset \text{ and } \text{Bound}(r) \cap u = \emptyset \text{ and } \\
v^j \neq u^j
\end{array} \right] \text{ implies } \tilde{\text{isol}}(v, q) \simeq \delta \text{ and }
\tilde{\text{isol}}(u, r) \simeq \delta
\]

See Section [C] page 65 for a detailed proof.

**Proposition 7 (Composed isol- and \( \tilde{\text{isol}} \)-processes cause deadlock, II).**

\[
\left[ \begin{array}{c}
q, r \in \text{Basic} \text{ and } q, r \in \text{TauFree} \text{ and } \\
\text{Act}(q), \text{Act}(r) \subseteq \text{dom}(\Xi) \text{ and } \\
\text{Bound}(q) \cap w = \emptyset \text{ and } \text{Bound}(r) \cap w f = \emptyset \text{ and } \\
e \neq f
\end{array} \right] \text{ implies } \tilde{\text{isol}}(w e, q) \simeq \delta \text{ and }
\tilde{\text{isol}}(w f, r) \simeq \delta
\]

See Section [C] page 65 for a detailed proof.

Although its proof follows the same structure as the proofs of the previous three propositions, we mention Proposition 8 separately for two reasons. First, this proposition does not really generalize a proposition from the previous subsection; second, this proposition plays a crucial role in the proof of an important lemma, Lemma 1 in Section 5.3. Proposition 8 states that if we compose a (co)isolated process \( \tilde{\text{isol}}(w, p) \) using \( \| \) with any other process, deadlock occurs.

\footnote{Alternatively, we could define a function toSet for converting strings to sets and require \( \text{Bound}(p) \cap \text{toSet}(w) = \emptyset \). We favor the \( \cap \)-notation, because it requires a bit less space, especially in proofs.}
Proposition 8 (isol-processes cause deadlock in \([\cdot]\)-ed (left-merged) terms).

\[
\text{if } p \in \text{Basic and } p \in \text{TauFree and } \text{Act}(p) \subseteq \text{dom}(\Xi) \text{ and } \text{Bound}(p) \cap w = \emptyset \text{ implies } \begin{cases}
?((\text{isol}(w, p) \parallel q) \simeq \delta \text{ and } \\
?((\text{isol}(w, p) \cdot p') \parallel q) \simeq \delta
\end{cases}
\]

See Section \[\text{C}\] page \[70\] for a detailed proof.

5.3. Synchronization and Preservation

We proceed with a series of more significant properties that concern synchronization and preservation, starting with the former.

Synchronization. Lemma \[\text{1}\] states that the parallel composition operator, when operating on the isolation and the coisolation of the same process, behaves as the synchronous composition operator. Intuitively, this lemma captures the phenomenon that (co)isolated processes execute in lockstep when appropriately synchronized by \(\cdot\): when composed in parallel, an isolated process and its coisolated sibling always wait for each other until they can perform an auxiliary action and its dual together.

**Lemma 1 (Synchronization lemma).**

\[
[p \in \text{Basic and } p \in \text{TauFree and } \text{Act}(p) \subseteq \text{dom}(\Xi) \text{ and } \text{Bound}(p) \cap w = \emptyset] \implies \begin{cases}
?((\text{isol}(w, p) \parallel \text{isol}(w, p)) \simeq ?((\text{isol}(w, p) \parallel \text{isol}(w, p)) \text{ and } \\
?((\text{isol}(w, p) \cdot p') \parallel (\text{isol}(w, p) \cdot p')) \simeq ?((\text{isol}(w, p) \cdot p') \parallel (\text{isol}(w, p) \cdot p'))
\end{cases}
\]

**Proof (sketch).** By Axiom M, the parallel composition of \(\text{isol}(w, p)\) and \(\text{isol}(w, p)\) is provably equal to a nondeterministic choice among three options. The first two options have the shape \(\text{isol}(w, p) \parallel q\) and \(\text{isol}(w, p) \parallel q\). Distribute \(\parallel\) over + by Axiom Q3, and apply Proposition 8 to conclude that those first two options are provably equal to \(\delta\) (derive the premise of Proposition 8 from the premise of this lemma). After eliminating these \(\delta\)-s by Axiom A6, only the third option of the choice remains, which completes the proof.

See Section \[\text{D}\] page \[71\] for a detailed proof.

**Preservation.** The remaining four lemmas in this subsection concern properties stating that the basic operators of the algebra used are preserved by \text{split} (i.e., \text{split} is homomorphic with respect to the basic operators). These properties make the proof of correctness in Section \[5.4\] relatively straightforward, but in some sense move the main proof obligations (and complexities) to the lemmas in this subsection.

We start with Lemma 2 which states that + is preserved by \text{split} (i.e., \text{split} is homomorphic with respect to \(\parallel\)).

**Lemma 2 (Preservation lemma for +).**

\[
[q + r \in \text{Basic and } q + r \in \text{TauFree and } \text{Act}(q + r) \subseteq \text{dom}(\Xi) \text{ and } \text{Bound}(q + r) \cap w = \emptyset] \implies \text{split}(w, q + r) \simeq \text{split}(w1, q) + \text{split}(w2, r)
\]

**Proof (sketch).** By the definition of \text{split} and by Lemma 1 (derive the premise of Lemma 1 from the premise of this lemma), conclude that \text{split}(w, q + r) is provably equal to \(?((\text{isol}(w, q + r) \parallel \text{isol}(w, q + r))\). Apply the definitions of \text{isol} and \text{isol} to obtain \(?((q1 + r1) \parallel (q2 + r2))\) for \(q1 = \text{isol}(w1, q), r1 = \text{isol}(w2, r), q2 = \text{isol}(w1, q), r2 = \text{isol}(w2, r)\). Distribute \(\parallel\) over + by Axiom S7, and afterwards, distribute \(\parallel\) over + by Axiom Q3. This yields the process \(?((q1 | q2) + ?(q1 | r2) + ?(r1 | q2) + ?(r1 | r2))\). The alternative composition

\[13\] With abuse of terminology, ignoring that \(w\) becomes \(w1\) and \(w2\).
of the first and the last option give the required result (after applying Lemma 1 to each). To get rid of the middle two options, conclude that both of them are provably equal to \(\delta\) by Proposition 6 (derive the premise of Proposition 6 for both of them from the premise of this lemma), and eliminate them by Axiom A6.

See Section D page 72 for a detailed proof.

We continue with Lemma 3 which states that \(\rightarrow, \circ\) is preserved by \(\text{split}\) (i.e., \(\text{split}\) is homomorphic with respect to \(\rightarrow, \circ\)).

**Lemma 3 (Preservation lemma for \(\rightarrow, \circ\)).**

\[
\text{split}(w, c \rightarrow q \circ r) \simeq c \rightarrow \text{split}(w_1, q) \circ \text{split}(w_2, r)
\]

**Proof (sketch).** Distinguish two cases: \(c \equiv \text{true}\) and \(c \equiv \text{false}\). In the former case, by the definition of \(\text{split}\), \(\text{isol}\) and \(\overline{\text{isol}}\), and \(c\), conclude that \(\text{split}(w, c \rightarrow q \circ r)\) is provably equal to \(\overline{?((\text{true} \rightarrow \overline{\text{isol}}(w_1, q) \circ r'))}\). Reduce these processes by Axiom COND1 (from left to right) and apply \(\text{split}\) to obtain \(\text{split}(w_1, q)\). Use Axiom COND1 once more (from right to left this time) to get the required result. The other case follows analogously.

See Section D page 74 for a detailed proof.

The following lemma, Lemma 4, states that \(\sum\) is preserved by \(\text{split}\) (i.e., \(\text{split}\) is homomorphic with respect to \(\sum\)), if the domain of quantification has only finitely many elements. We require finiteness, because otherwise we cannot apply Proposition 1 in the proof, which we do.

**Lemma 4 (Preservation lemma for \(\sum\)).**

\[
\begin{align*}
\sum_{x \in \{d_1, \ldots, d_k\}} q \in \text{Basic} & \quad \text{and} \quad \sum_{x \in \{d_1, \ldots, d_k\}} q \in \text{TauFree} \quad \text{and} \\
\text{Act}(\sum_{x \in \{d_1, \ldots, d_k\}} q) \subseteq \text{dom}(\Xi) & \quad \text{and} \quad \text{Bound}(\sum_{x \in \{d_1, \ldots, d_k\}} q) \cap w = \emptyset
\end{align*}
\]

implies

\[
\text{split}(w, \sum_{x \in \{d_1, \ldots, d_k\}} q) \simeq \sum_{x \in \{d_1, \ldots, d_k\}} \text{split}(wx, q)
\]

**Proof (sketch).** By the definition of \(\text{split}\) and by Lemma 1 (derive the premise of Lemma 1 from the premise of this lemma), conclude that \(\text{split}(w, \sum_{x \in \{d_1, \ldots, d_k\}} q)\) is provably equal to \(\overline{?((\text{isol}(w, \sum_{x \in \{d_1, \ldots, d_k\}} q) | \overline{\text{isol}}(w, \sum_{x \in \{d_1, \ldots, d_k\}} q)))}\). Then apply Proposition 1 from left to right, to obtain the same process but with an ordinary alternative composition: \(\overline{?(\sum_{i=1}^\ell \text{isol}(wd_i, q[d_i/x]) | \sum_{i=1}^\ell \overline{\text{isol}}(wd_i, q[d_i/x]))}\). Distribute \(\lor\) over + by Axiom S7, and afterwards, distribute \(\forall\) over + by Axiom Q3. This yields the process \(\sum_{i=1}^\ell \sum_{j=1}^\ell ?(\text{isol}(wd_i, q[d_i/x]) | \overline{\text{isol}}(wd_j, q[d_j/x]))\). The alternative composition of the processes on the “diagonal” yields the desired result (after applying Q3 and Proposition 1 from right to left). To get rid of the processes \(\forall\) on the diagonal, conclude that each of them is provably equal to \(\delta\) by Proposition 7 (derive the premise of Proposition 7 for each of them from the premise of this lemma), and eliminate them by Axiom A6.

See Section D page 74 for a detailed proof.

The final lemma of this subsection states that \(\cdot\) is preserved by \(\text{split}\) (i.e., \(\text{split}\) is homomorphic with respect to \(\cdot\)). The proof of Lemma 5 requires the application of the other preservation lemmas and, in contrast to those lemmas, involves structural induction. This makes Lemma 5 the most complex among the lemmas in this subsection.

---

14 Actually, the application of Proposition 1 yields \(\overline{?(\sum_{i=1}^\ell \text{isol}(wx, q)[d_i/x] | \sum_{i=1}^\ell \overline{\text{isol}}(wx, q)[d_i/x])}\). However, one can show that this is provably equal to \(\overline{?(\sum_{i=1}^\ell \text{isol}(wx, q)[d_i/x] | \sum_{i=1}^\ell \overline{\text{isol}}(wx, q)[d_i/x])}\) by induction on the structure of \(q\).
Lemma 5 (Preservation lemma for \( \cdot \)).

\[
\begin{align*}
q \cdot r & \in \text{Basic} \text{ and } q \cdot r \in \text{TauFree and} \\
\text{Act}(q \cdot r) & \subseteq \text{dom}(\Xi) \text{ and } \text{Bound}(q \cdot r) \cap \emptyset = \emptyset
\end{align*}
\]
implies

\[
\text{split}(w, q \cdot r) \simeq \text{split}(w_1, q) \cdot \text{split}(w_2, r)
\]

**Proof (sketch).** By the definition of \( \text{split} \) and by Lemma [1] (derive the premise of Lemma [1] from the premise of this lemma), conclude that \( \text{split}(w, q \cdot r) \) is provably equal to \( \text{split}(w_1, q) \cdot \text{split}(w_2, r) \). Apply the definitions of \( \text{TauFree} \) and \( \text{isoTF} \) to obtain \( \text{split}(w_1, q) \cdot \text{split}(w_2, r) \) for \( q_1 = \text{isoTF}(w_1, q) \), \( r_1 = \text{isoTF}(w_2, r) \), \( q_2 = \text{isoTF}(w_1, q) \), \( r_2 = \text{isoTF}(w_2, r) \). Then, proceed by induction on the structure of \( q \) to show that \( \text{split}(w_1, q) \cdot \text{split}(w_2, r) \) is provably equal to \( \text{split}(w_1, q) \cdot \text{split}(w_2, r) \) (afterwards, the required result follows straightforwardly by identifying \( \text{split}(w_2, r) \) with \( ?(r_1 || r_2) \)).

Establish the base of the induction \( (q \neq \text{multiact} \text{ or } \delta) \) by applying Axiom S6, Axiom Q4, and Lemma [1]. To prove the inductive step, set up a case distinction for the main operator of \( q \). Cases \(+, -, \cdot, \circ, \diamond\), and \( \sum \) follow by similar reasoning as in Lemmas [2] and [3]. The key difference between those lemmas and the corresponding cases in the inductive step lies in the presence of \( r_1 \) and \( r_2 \) in the latter. Using the induction hypothesis and the grayed out consequents of Propositions [6] (for \(+\)) and [7] (for \(\sum\)), one can “neutralize” their effect and, basically, follow the same structure as the proofs of the other preservation lemmas. For proving the \( \cdot \) case, the induction hypothesis and Lemma [1] suffice.


5.4. Correctness

Next, we state three theorems which, in increasing level of generality, establish the correctness of our splitting procedure. The first theorem, Theorem [4] states that a split multiaction has the same behavior as the original, unsplit multiaction.

**Theorem 1 (Correctness theorem for multiactions).**

\[
\begin{align*}
\alpha & \in \text{TauFree and } \text{Act}(\alpha) \subseteq \text{dom}(\Xi) \text{ implies } \text{split}(w, \alpha) \simeq \alpha
\end{align*}
\]

**Proof (sketch).** By the definition of \( \text{split} \), Lemma [1] (derive the premise of Lemma [1] from the premise of this lemma), and Axiom SMA, conclude that \( \text{split}(w, \alpha) \) is provably equal to \( ?(\text{isoTF}(w, \alpha) \cup \text{isoTF}(w, \alpha)) \). Then, by straightforward induction on the structure of \( \alpha \), establish:

- \( \alpha \) is provably equal to \( \bigsqcup_{i=1}^{n} a_i(d_i) \cup \bigsqcup_{i=1}^{n'} a'_i(d'_i) \)
- \( \text{isoTF}(w, \alpha) \) is provably equal to \( \bigsqcup_{i=1}^{n} a_i(d_i) \cup \xi_{w_1}(a_i)(w^1) \cup \bigsqcup_{i=1}^{n'} \xi_{w_2}(a'_i)(w^2) \)
- \( \text{isoTF}(w, \alpha) \) is provably equal to \( \bigsqcup_{i=1}^{n} a_i(d_i) \cup \xi_{w_1}(a_i)(w^1) \cup \bigsqcup_{i=1}^{n'} \xi_{w_2}(a'_i)(w^2) \)

Insert the latter two results in \( ?(\text{isoTF}(w, \alpha) \cup \text{isoTF}(w, \alpha)) \), and by Axiom MA2, rearrange the actions in the resulting multiaction to obtain:

\[
\begin{align*}
?(\bigsqcup_{i=1}^{n} a_i(d_i) \cup \xi_{w_1}(a_i)(w^1) \cup \xi_{w_2}(a_i)(w^2)) \cup \bigsqcup_{i=1}^{n'} a'_i(d'_i) \cup \xi_{w_1}(a'_i)(w^1) \cup \xi_{w_2}(a'_i)(w^2))
\end{align*}
\]

Then, because \( ? \) effectively filters out all pairs of an auxiliary action and its dual (e.g., \( \xi_{w_1}(a_i)(w^1) \) and \( \xi_{w_2}(a_i)(w^2) \)), without affecting the original actions (because the sets of auxiliary and original actions do not overlap by Definition [3]), obtain \( \bigsqcup_{i=1}^{n} a_i(d_i) \cup \bigsqcup_{i=1}^{n'} a'_i(d'_i) \), which is provably equal to \( \alpha \) (by the first item in the above itemization).


20
The following theorem states that a split basic process has the same behavior as the original, unsplit process.

**Theorem 2 (Correctness theorem for basic processes).**

\[
\begin{array}{c}
p \in \text{Basic and } P \in \text{Basic and } \text{referring to the premise of Theorem 1 (derive the premise of Theorem 1 from the premise of this theorem) or the definition of split (for \( \delta \)). To prove the inductive step, set up a case distinction for the main operator of \( P \) and prove these cases quickly using the preservation lemmas (derive the premises of Lemmas 2, 3, 4, and 5 from the premise of this theorem). For example,} \\
\text{Pr}_i \text{lie} \text{the premises of Lemmas 2, 3, 4, and 5 from the premise of this theorem). For example,} \\
\text{Pr}_i \text{lie} \text{the premises of Lemmas 2, 3, 4, and 5 from the premise of this theorem). For example,} \\
\text{Pr}_i \text{lie} \text{the premises of Lemmas 2, 3, 4, and 5 from the premise of this theorem). For example,}
\end{array}
\]

**Proof (sketch).** Prove this theorem by a relatively straightforward induction on the structure of \( P \). The base case \((p = 1, Q + \delta)\) follows immediately from Theorem 1 (derive the premise of Theorem 1 from the premise of this theorem) or the definition of split (for \( \delta \)). To prove the inductive step, set up a case distinction for the main operator of \( P \) and prove these cases quickly using the preservation lemmas (derive the premises of Lemmas 2, 3, 4, and 5 from the premise of this theorem). For example, \( (p = q + r) \):

\[
\text{split}(w, p) = \text{split}(w, q + r) \overset{\text{lax-minus}}{\overset{\text{split}(w1, q) + \text{split}(w2, r)}{\overset{\text{split}(w1, q) + \text{split}(w2, r)}}} = q + r = p
\]

See Section 23, page 85, for a detailed proof.

The last theorem of this paper states that split process definitions (potentially mutually recursive) have the same behavior as the original, unsplit process definitions. To prove this theorem, we first present a proposition about the untimed subset of mCRL2, adapted from 39: Proposition 9 states that one can collapse \( k \), potentially mutually recursive, process definitions ( referenced by \( P_1, \ldots, P_k \) ) into a single process definition ( referenced by \( \tilde{P} \)).

**Proposition 9 ([39] Section 4.3].**

\[
\begin{array}{c}
P_1(x_1 \cdot D_1) = p_1, \\
P_2(x_2 \cdot D_2) = p_2, \\
\ldots \\
P_k(x_k \cdot D_k) = p_k, \\
\tilde{P}(y, x : N \times D) = y \approx 1 \rightarrow p_1[P_1(d) := \tilde{P}(1, h(d))] \cdots [P_k(d) := \tilde{P}(k, h(d))] \circ \\
\ldots \\
y \approx k \rightarrow p_k[P_1(d) := \tilde{P}(1, h(d))] \cdots [P_k(d) := \tilde{P}(k, h(d))] \circ \delta \\
\text{and } h = \text{harmonizer}(D_1 \cup \cdots \cup D_k, D) \\
\end{array}
\]

Proposition 9 may look complex, but conceptually, it states a rather simple property. Essentially, it corresponds to the “collapsing into one equation” step of the mCRL2 linearization process [39], as follows. Reference \( \tilde{P} \) has a parameter \( y \) which represents the indices of the \( k \) processes. The body of \( \tilde{P} \) contains a conditional choice dependent on the value of \( y \); if \( y \) equals some index \( i \), the body of \( \tilde{P} \) behaves as the body of \( P_i \). Thus: \( \tilde{P}(i) \approx P_i \). To ensure also that \( \tilde{P} \) contains only references to itself, one should substitute occurrences of \( P_1, \ldots, P_k \) with \( \tilde{P} \) in \( p_i \). To this end, we write \( p_i[P_j(d) := \tilde{P}(j, h(d))] \) for the process resulting from replacing \( P_j(d) \) by \( \tilde{P}(j, h(d)) \) in \( p_i \) (for any \( d \)), for some harmonization function \( h \). Such a function maps data tuples in \( D_1 \cup \cdots \cup D_k \) to data tuples in \( D \). Intuitively, \( h \) transforms the parameters of each of the process definitions \( P_1, \ldots, P_k \) to a single tuple of parameters for \( \tilde{P} \). Neither the precise meaning of harmonization nor the definition of harmonizer matter in the remainder, so we skip them (details appear elsewhere [39]).

We proceed with our final theorem. Let \( \text{Ref}(p) \) (defined in Figure 18) denote the set of references occurring in \( p \).
Theorem 3 (Correctness theorem for process specifications).

\[
\begin{align*}
& P_1(x_1: D_1) = p_1, P_1^\dagger(x_1: D_1) = \text{split}(\epsilon, p_1), \\
& \vdots \\
& P_k(x_k: D_k) = p_k, P_k^\dagger(x_k: D_k) = \text{split}(\epsilon, p_k) \\
& \text{and } p_1, \ldots, p_k \in \text{TauFree} \\
& \text{and } \text{Act}(p_1), \ldots, \text{Act}(p_k) \subseteq \text{dom}(\Xi) \\
& \text{and } [\text{Ref}(p_i) \subseteq \{P_1, \ldots, P_k\} \text{ for all } 1 \leq i \leq k] \\
\end{align*}
\]

implies \[ P_i \simeq P_i^\dagger \text{ for all } 1 \leq i \leq k \]

Proof (sketch). Apply Proposition 9 to collapse the definitions referenced by \( P_1, \ldots, P_k \) into one definition \( \tilde{P} = p \). Similarly, apply Proposition 9 to collapse the definitions referenced by \( P_1^\dagger, \ldots, P_k^\dagger \) into one definition \( \tilde{P}^\dagger = p^\dagger \). To obtain the desired result, show that \( \tilde{P} \) is provably equal to \( \tilde{P}^\dagger \) by demonstrating that some process operator \( \Phi \) has both \( \tilde{P} \) and \( \tilde{P}^\dagger \) as fixed points (and apply RSP). Define \( \Phi(Z) = p[Z := Z] \), and immediately conclude \( \Phi(\tilde{P}) \simeq \tilde{P} \). To show that also \( \Phi(\tilde{P}^\dagger) \simeq \tilde{P}^\dagger \), essentially, it suffices to show that \( p_i \simeq \text{split}(\epsilon, p_i) \). This follows from Theorem 2 (derive the premise of Theorem 2 from the premise of this theorem).

See Section E, page 90, for a detailed proof. We establish the \( \Phi(\tilde{P}^\dagger) \simeq \tilde{P}^\dagger \) step with a separate auxiliary theorem (Theorem 4, page 88). \( \square \)

6. An Application of the Splitting Procedure: Splitting Connectors

Up to now, we have defined a splitting procedure for untimed mCRL2 and proved its correctness, all independent of Reo. Now, as one of its applications, we use this splitting procedure to justify the region-based optimization technique for Reo implementations (i.e., the version with direct transportation of data and control information in asynchronous regions—see Section 1). First, we formalize (a)synchronous regions in terms of process algebra. Afterwards, we split (process algebraic semantic specifications of) connectors.

6.1. Formalization of (A)synchronous Regions

We provide a formal definition of the synchronous regions of a connector, based on the mCRL2 semantics of Reo. Let \( p \) denote a process describing the behavior of a Reo connector, and let \( \rightarrow \rightarrow \) denote its transition relation (labeled with multi-actions). \(^{16}\) Recall that every action in \( p \) represents a channel end or a node end. Let \( a \in \text{Act}(p) \) denote one such end. We define the \( a \)-synchronous region of \( p \) as the smallest set \( X_a \subseteq \text{Act}(p) \) such that:

- \( a \in X_a \)
- \( b \in X_a \Rightarrow [\text{Act}(\beta) \subseteq X_a \text{ for all } \beta \text{ such that } q \xrightarrow{\beta} q' \text{ and } b \in \text{Act}(\beta)] \) \(^{17}\)

\(^{16}\)We have not given the definition of the transition relation (although we showed examples in Figures 9, 19, and 21), because the precise definition does not matter in this paper. See Groote et al. \(^{14}\).

\(^{17}\)Square brackets for readability.
The second rule states that all the ends that occur in the same multiaction belong to the same synchronous region. The third rule states that all the ends that can have flow in some state $q$, but possibly in different transitions leaving $q$, belong to the same synchronous region. In that case, channel ends may exclude each other from flow, which requires them to synchronize and communicate about their behavior.

To exemplify the previous definition, consider the connector modeled by the process $p = a \sqcup b \cdot c + d$. Informally, either this connector has flow through $a$ and $b$, followed by flow through $c$, or it has flow through $d$. Hence, because $a \in X_a$, we add $d$ to $X_a$ (third rule). This concludes the construction: $X_a = X_b = X_d = \{a, b, d\}$ and $X_c = \{c\}$.

We define the set of the synchronous regions of the connector modeled by $p$ as

$$X = \{X_a \mid a \in \text{Act}(p)\}$$

and the set containing its asynchronous regions as

$$Y = \{(a, b) \mid \text{connected}(a, b) \text{ and } a \in X \text{ and } b \in X' \text{ and } X \neq X' \text{ and } X, X' \in X\}$$

where connected$(a, b)$ holds iff ends $a$ and $b$ belong to the same channel.

### 6.2. Splitting Connectors

As motivated in Section 1, we set out to establish the soundness of splitting connectors along the boundaries of their (a)synchronous regions. However, we can split any (syntactically $\tau$-free) process along any set of actions $A$ by Theorem 3. This suggests that regardless of its (a)synchronous regions, one can split a connector in any possible way and preserve its original semantics. While true in theory, there is a catch for implementations of split connectors in practice: the parallel composition of the isolation and the coisolation of a connector process must synchronize appropriately, as represented by the $?$ operator (see Definition 4). Depending on the particular implementation approach, which in turn may depend on the underlying hardware architecture (see Section 1), performing $?$ at run-time may cost an unreasonable amount of resources, if possible at all.

**Region-based splitting.** We start with an example of splitting based on (a)synchronous regions. Suppose that we split fifo$(a, b)$ into two parts: one part contains only $a$, while the other part contains only $b$. Recall from Section 3 that the semantics of this channel is given by the process definition $\text{Fifo}(a; b) = \sum_{x \in \text{Data}}(a(x) \cdot b(x)) \cdot \text{Fifo}(a; b)$. Splitting along $A = \{a\}$ (or equivalently, along $A = \{b\}$) yields:
Here, ? in fact represents the asynchronous region of fifo\((a; b)\), because it synchronizes the two synchronous regions \{a\} and \{b\}. The fact that auxiliary actions happen at the same time as the corresponding original actions represents direct transportation of data and control information in asynchronous regions (see Section 7).

Suppose that we want to implement \(p = \sum_{x \in \text{Data}} (a(x) \cup \xi_1(a)(x) \cdot \xi_2(b)(x))\) and \(q = \sum_{x \in \text{Data}} (\xi_1(a)(x) \cdot b(x) \cup \xi_2(b)(x))\) such that, when run in parallel, they behave as \(\sum_{x \in \text{Data}} (a(x) \cdot b(x))\). Crucially, these implementations should perform the synchronisation implied by ?\(\). Recall from Section 4 that intuitively, \(\xi_1(a)\) represents the act of “disseminating the performance of \(a\),” while \(\xi_2(a)\) represents the act of “discovering the performance of \(a\).” Thus, the implementation of \(p\) should: (1) accept data \(x\) on \(a\) and disseminate this acceptance, and (2) discover the dispersal of \(x\) on \(b\). Meanwhile, the implementation of \(q\) should: (1) discover the acceptance of data \(x\) on \(a\), and (2) dispense \(x\) on \(b\) and disseminate this dispersal. Thus, in each step, the implementations of \(p\) and \(q\) require only unidirectional communication about their behavior to synchronize: first, the implementation of \(p\) performs \(\xi_1(a)(x)\) and the implementation of \(q\) takes notice of this (by performing \(\xi_2(a)(x)\)); afterwards, \(p\) and \(q\) switch roles to perform \(\xi_2(b)(x)\) and \(\xi_2(b)(x)\). This shows that different synchronous regions can decide on their behavior independently of each other: region \{a\} does not need to know that region \{b\} will dispense data before it can accept data—it can decide to do so without communication.

We argue that this can yield performance improvements in practice: although the isolation and the coisolation of a process \(p\) have the same transition system modulo transition labels (i.e., they have the same syntactic structure), benefits can arise when we compose them in parallel with another split process \(q\). In that case, there may exist a transition \(t\) of the isolation of \(p\) that can proceed independently—without communication among the ends involved—of a transition \(t'\) of the coisolation of \(q\). Without splitting, in contrast, communication among the ends involved in \(t\) and \(t'\) must always take place to decide on whether to behave according to \(t\) or \(t'\), or both. But in the split case, the ends can act independently. For instance, if we put two split fifo\(\) instances in sequence (as in Figure 14), the source end \(a\) of the first fifo\(\) can proceed independently of the sink end \(b\) of the second fifo\(\). This means that, if empty, the first fifo\(\) can accept a data item on \(a\) (and place it in its buffer) without communicating with \(b\). Similarly, if full, the second fifo\(\) can dispense a data item on \(b\) (and remove it from its buffer) without communicating with \(a\). In contrast, if we put two unsplit fifo\(\) instances in sequence, the source end \(a\) and the sink end \(b\) communicate with each other to decide on their joint behavior, even though the behavior of those ends does not depend on each other. By splitting, one avoids this unnecessary communication.

**Arbitrary splitting.** To demonstrate that splitting arbitrarily makes no sense, suppose that we split sync\((a; b)\) into two parts: one part contains only \(a\), while the other part contains only \(b\). Recall from Section 3 that the semantics of this channel is given by the process definition \(\text{Sync}(a; b) = \sum_{x \in \text{Data}} a(x) \cup b(x) \cdot \text{Sync}(a; b)\). Splitting along \(A = \{a\}\) (or equivalently, along \(A = \{b\}\)) yields:

\[
\text{Sync}(a; b) = \text{split}(\epsilon, \sum_{x \in \text{Data}} a(x) \cup b(x) \cdot \text{Sync}(a; b))
\]

\[
= \text{split}(\epsilon, \sum_{x \in \text{Data}} a(x) \cup b(x) \cdot \text{Sync}(a; b))
\]

\[
= ?(\text{isol}(\epsilon, \sum_{x \in \text{Data}} a(x) \cup b(x)) \mid \text{isol}(\epsilon, \sum_{x \in \text{Data}} a(x) \cup b(x)))) \cdot \text{Sync}(a; b)
\]

\[
= ?(\sum_{x \in \text{Data}} \text{isol}(x, a(x) \cup b(x)) \mid \sum_{x \in \text{Data}} \text{isol}(x, a(x) \cup b(x)))) \cdot \text{Sync}(a; b)
\]

\[
\text{Sync}(a; b) = \text{split}(\epsilon, \sum_{x \in \text{Data}} a(x) \cup b(x) \cdot \text{Sync}(a; b))
\]
7. Related Work

6.3. Example: Sequencer2

Implementation sketch. We sketch an implementation of the split \( \text{fifo}(a, b) \) on a shared memory machine with multithreading. First, we instantiate two threads, \( A \) and \( B \), for the processes \( p = \sum_{x \in \text{Data}} (a(x) \sqcup \xi_1(a)(x) \sqcup \xi_2(b)(x)) \) and \( q = \sum_{x \in \text{Data}} (\xi_1(a)(x) \sqcup b(x) \sqcup \xi_2(b)(x)) \). Every multiaction \( \alpha \) translates to the atomic execution of a block of code representing the actions occurring in \( \alpha \).

6.3. Example: Sequencer2

7. Related Work

Process decomposition. Closest to the process algebraic work presented in this paper seems the work on processes decomposition, first investigated by Milner and Moller in the late 1980s–early 1990s \[32\]. In that work, Milner and Moller define the notion of a prime process, and they explore what kind of processes \( p \) have a unique decomposition into primes \( p_1, \ldots, p_k \) such that the parallel composition of those primes is strongly bisimilar to \( p \). A process \( p \) qualifies as a prime process if, for all \( q \) and \( r \), it holds that \( p \simeq q \parallel r \) implies that either \( q \) or \( r \)—not both—is equivalent to the neutral element for \( \parallel \) (the algebra used in this paper does not have such an element). In other words, one cannot decompose \( p \) further into nonneutral processes. Among other results, Milner and Moller show that finite processes in the algebra they consider have a unique prime decomposition under strong bisimulation. In his PhD thesis, Moller additionally gives a unique decomposition result with respect to (weak) observational congruence \[33\] Section 4.4\].

\footnote{The parallel composition operator differs slightly from the one in this paper: the operator used by Milner and Moller satisfies \( q \parallel r \simeq q \parallel r + r \parallel q \), while in this paper, we have \( q \parallel r \simeq q \parallel r + r \parallel q + q \parallel r \) (by Axiom M in Figure 8).}
After Milner and Moller, also other researchers investigated process decomposition for various process calculi. This led to some interesting applications. For instance, Lanese et al. proved a prime decomposition theorem for a higher-order process calculus and used it to prove the completeness of the axiomatization of that calculus [29]. Aceto et al. [1] and Christensen [8] used prime decomposition theorems for a similar purpose, among other contributions. Alternatively, Groote and Moller used process decomposition for verification [15]: they showed that instead of checking \( p \simeq q \) directly, in some cases, one can more efficiently check whether the primes of \( p \) and \( q \) are equivalent (while preserving soundness and completeness). The projection operator introduced by Groote and Moller for decomposing processes seems somewhat related to our functions \( \text{isol} \) and \( \text{isol} \), albeit rather distantly. Applied to a process \( p \), similar to \( \text{isol} \) and \( \text{isol} \), this projection operator throws some actions from \( p \) away and keeps others for communicating with other processes. However, those preserved communication actions must already occur in both the original \( p \) and the original other processes; the projection operator does not add auxiliary actions the same way \( \text{isol} \) and \( \text{isol} \) do (more significant differences between process decomposition and process splitting follow shortly).

Other contributions to the theory of process decomposition include the work of Kučera [28] (decidability results and constructions of decompositions), Luttik and Van Oostrom [31] (generalization of decomposition to partial commutative monoids), Luttik [30] (unique parallel decomposition modulo branching and weak bisimilarity), and Dreier et al. [13] (decomposition in the applied \( \pi \)-calculus).

Although related, the work on process decomposition differs significantly from our work on process splitting. For one thing, even though both approaches derive smaller processes from an existing one (such that their parallel composition is equivalent to the original process), the notion of “smaller” in our work does not involve primality. In fact, one could argue that the processes resulting from our splitting procedure are not really smaller than the original process due to the introduction of auxiliary actions. Another difference concerns uniqueness, which plays no explicit role in our splitting procedure. Note, however, that only one isolation and only one coisolation exists for every process under some fixed \( A \) and \( \Xi \) (due to the deterministic definition of \( \text{split} \)). So technically, we have uniqueness. Finally, in process decomposition, one usually requires no additional synchronization on top of the parallel composition of the primes. We, in contrast, needed to introduce the \( ? \) operator to achieve appropriate synchronization between the isolation and the coisolation of a process.

Connector decomposition. In this paper, we developed a process algebraic splitting procedure, which we then applied to Reo’s process algebraic semantics, thereby effectively splitting connectors. Interestingly, different notions of splitting and decomposition of Reo connectors—or their semantics—already exist in the literature. Although inapplicable for our purpose, we discuss them below.

Koehler and Clarke investigated the decomposition of port automata [23], an operational model of connector behavior. The states of a port automaton represent the internal configurations of a connector; its transitions, labeled with sets of firing node names, describe atomic execution steps. Through special product and hiding operators on port automata, one can compositionally construct a connector model from a set of smaller automata for the primitive Reo connectors. Koehler and Clarke showed that they can decompose every port automaton into instances of only two primitive automata. Essentially, this means that one can construct every Reo connector expressible by a port automaton from instances of only two different primitive connectors.

Pourvatan et al. explored the decomposition of complete constraint automata [34], a more expressive operational model of connector behavior than port automata and an extension of ordinary constraint automata [4]. Their approach differs significantly from the work of Koehler and Clarke: Pourvatan et al. develop a notion of inverse for their automata, which allows them to factor out certain parts of a complete constraint automaton based on another such automaton. A typical application of this decomposition technique is connector synthesis. Suppose that we have a specification (as an automaton) of the whole system that we want to build and specifications (also as automata) of the components that this system consists of, but no specification of the connector that should connect those components. We can then factor out the component automata from the system automaton to get the automaton specifying the behavior of the connector. Pourvatan et al. exemplify this with a service-oriented application.

Although not often considered (exceptions exist though—see, e.g., [9]), we remark that Arbab mentioned
a split operation already in his introductory paper on Reo [2]. However, this split operation splits nodes instead of connectors (i.e., sets of nodes). Because our interest lies in splitting connectors, we could not use Arbab’s notion of splitting.

Proença pioneered the work on (a)synchronous regions, region-based optimization techniques for Reo implementations, and connector splitting in this PhD thesis and associated publications [35, 36, 37]. He developed the first working Reo implementation based on these ideas, demonstrated its merits through benchmarks, and invented a new automaton model—behavioral automata [37]—to reason about split connectors. Also, Proença formulated a number of soundness and completeness criteria for when a split behavioral automaton preserves the semantics of the original (but without proofs). Recently, Clarke and Proença explored connector splitting in the context of the connector coloring semantics [11]. They discovered that the standard version of that semantics has undesirable properties in the context of splitting: some split connectors that intuitively should be equivalent to the original connector are not equivalent under the standard model. To address this problem, Clarke and Proença propose a new variant called partial connector coloring, which allows one to better model locality and independencies between different parts of a connector.

8. Conclusion and Future Work

We presented a procedure for splitting processes in a process algebra with multiactions and data (the untimed subset of the specification language mCRL2). This splitting procedure cuts a process into two processes along a set of actions $A$: roughly, one of these processes contains no actions from $A$, while the other process contains only actions from $A$. We stated and proved a theorem asserting that the parallel composition of these two processes is provably equal from a set of axioms (sound and complete with respect to strong bisimilarity) to the original process under some appropriate notion of synchronization.

We applied our splitting procedure to the process algebraic semantics of the coordination language Reo: using this procedure and its related theorem, we formally established the soundness of splitting Reo connectors along the boundaries of their (a)synchronous regions in implementations of Reo. Such splitting can significantly improve the performance of connectors as shown elsewhere [11, 35, 36].

Our work shows the feasibility of using the language mCRL2 (not the associated toolset) for proving properties of a whole language, Reo, rather than of concrete connectors. This subtly, yet significantly, differs from the work presented in [27, 24, 25, 26]. In those paper, Kokash et al. introduce the process algebraic semantics of Reo for verifying concrete connectors (e.g., “this connector never deadlocks”) but obtain no results about Reo as a language.

We identify several directions for future work.

- Implementing the splitting procedure to facilitate automatic splitting of processes, as well as a tool for the automatic detection of (a)synchronous regions of Reo connectors. Combined, they allow us to mechanically split connectors along their (a)synchronous regions. We can then integrate this in one of the code generation frameworks currently under development for Reo.

- Investigating other ways of splitting processes, corresponding to other versions of the region-based optimization technique (see Section 1). The procedure we introduced in this paper splits processes in a synchronous manner such that $\xi(a)$ occurs at the same time as the action $a$ itself. We imagine at least two other ways of splitting processes. In one approach, $\xi(a)$ occurs after $a$ but before the next action. Then, the process $q = a \cdot b$ has $a \cdot \xi(a) \cdot \xi(b)$ as its $\{a\}$-isolation (instead of $a \sqcup \xi(a) \cdot \xi(b)$). In another approach, $\xi(a)$ occurs after $a$ but possibly concurrently with the next action. Then, $q$ has $a \cdot (\xi(a) \parallel \xi(b))$ as its isolation. We speculate that these splitting approaches are sound only under equivalences weaker than strong bisimulation.

This particular line of future work seems related to existing work on delay-insensitive circuits (e.g., [38]) and desynchronization (e.g., [5, 12]), the derivation of an asynchronous system from a synchronous
system: for the class of desynchronizable systems, the original synchronous system and the newly constructed asynchronous system are equivalent. If we use the splitting procedure presented in this paper to obtain such an original synchronous system, we may use—perhaps with modifications—results from desynchronization for more asynchronous splitting.

References


A. More Definitions

\[ d \in d \]
\[ d \in w_1w_2 \iff [d \in w_1 \text{ or } d \in w_2] \]
\[ x \in x \]
\[ x \in w_1w_2 \iff [x \in w_1 \text{ or } x \in w_2] \]

Figure A.22: Definition of \( \varepsilon \).

\[
D \uplus w = \{ d \in D \mid d \in w \}
\]
\[
V \uplus w = \{ v \in V \mid v \in w \}
\]

Figure A.23: Definition of \( \uplus \).

\[
\langle d_1, \ldots, d_\ell \rangle[d/x] = \langle d_1[d/x], \ldots, d_\ell[d/x] \rangle
\]
\[
a(d)[d/x] = a(d[d/x])
\]
\[
\tau[d/x] = \tau
\]
\[
(\beta \sqcup \gamma)[d/x] = \beta[d/x] \sqcup \gamma[d/x]
\]
\[
P(d)[d/x] = P(d[d/x])
\]
\[
\delta[d/x] = \delta
\]
\[
(q \oplus r)[d/x] = q[d/x] \oplus r[d/x]
\]
\[
(c \rightarrow q \circ r)[d/x] = c[d/x] \rightarrow q[d/x] \circ r[d/x]
\]
\[
(\sum_{x \in D} q)[d/x] = \sum_{x \in D} q[d/x]
\]
\[
(\sum_{y \in D} q)[d/x] = \sum_{y \in D} q[d/x]
\]
\[
\text{if } x \neq y
\]
\[
f(q)[d/x] = f(q[d/x])
\]

Figure A.24: Definition of \( [d/\cdot] \).

\[
p[Q :=_g R] = p
\]
\[
P(d)[Q :=_g R] = P(d)
\]
\[
Q(e)[Q :=_g R] = R(g(e))
\]
\[
\delta(Q :=_g R) = \delta
\]
\[
(q \oplus r)[Q :=_g R] = q[Q :=_g R] \oplus r[Q :=_g R]
\]
\[
(c \rightarrow q \circ r)[Q :=_g R] = c \rightarrow q[Q :=_g R] \circ r[Q :=_g R]
\]
\[
(\sum_{x \in D} q)[Q :=_g R] = \sum_{x \in D} q[Q :=_g R]
\]
\[
f(q)[Q :=_g R] = f(q[Q :=_g R])
\]

Figure A.25: Definition of \( [\cdot/\cdot] \).

B. Proofs for Section 5.1

Proposition 10 (\( \text{isol and } [d/\cdot] \) commute on multi-actions).

\[ x \notin w_1, w_2 \text{ implies } \text{isol}(w_1xw_2, \alpha)[d/x] = \text{isol}(w_1dw_2, \alpha[d/x]) \]
PROOF. Assumptions:
• \( x \not\in w_1, w_2 \) (Z1).

By induction on the structure of \( \alpha \).

**Base:** \([\alpha = a(\mathbf{d}) \text{ or } \alpha = \tau] \). Observations:

- Conclude \((w_1xw_2)^1 = w_1^2x^2w_2^1\) by the definition of \( \hat{\xi} \). Then, because \( x^2 = \epsilon \) by the definition of \( \hat{\xi} \), conclude \((w_1xw_2)^1 = w_1^2x^2w_2^1\). Then, because \( d^2 = \epsilon \) by the definition of \( \hat{\xi} \), conclude \((w_1xw_2)^2 = w_1^2d^2w_2^2\). Then, conclude \((w_1xw_2)^2 = (w_1d_2)^2\) by the definition of \( \hat{\xi} \) (Z2).

- Conclude \((w_1xw_2)^2 = w_1^2x^2w_2^2\) by the definition of \( \beta \). Then, because \( x^2 = \epsilon \) by the definition of \( \beta \), conclude \((w_1xw_2)^2 = w_1^2x^2w_2^2\) (Z3).

- Recall \((w_1^2xw_2^2)[d/x] = w_1^2[d/x]x[d/x]w_2^2[d/x]\) by the definition of \([/]\). Then, because \( x \not\in w_1, w_2 \) by Z1, conclude \((w_1^2xw_2^2)[d/x] = w_1^2x[d/x]w_2^2\). Then, conclude \((w_1^2xw_2^2)[d/x] = w_1^2dw_2^2\) by the definition of \([/]\) (Z4).

- Because \( d^2 = d \) by the definition of \( \beta \), conclude \( w_1^2dw_2^2 = w_1^2d^2w_2^2 \). Then, conclude \( w_1^2dw_2^2 = (w_1d_2)^2 \) by the definition of \( \beta \) (Z5).

Proceed by case distinction on the structure of \( \alpha \).

**Case:** \( \alpha = a(\mathbf{d}) \). Proceed by case distinction on the value of \( \tilde{\text{isol}} \).

- **Case:** \( \tilde{\text{isol}} = \text{isol} \). Conclude \([a \in A \text{ or } a \notin A] \) by ZFC—proceed by case distinction.

  **Case:** \( a \in A \). Observations:

  - Recall \( a \in A \) by the definition of this case. Then, conclude \( \text{isol}(w_1xw_2, a(\mathbf{d})) = a(\mathbf{d}) \cup \xi(\mathbf{w}, xw_2)^2(a)((w_1xw_2)^2) \) by the definition of \( \text{isol} \) (Z6).

  - Recall \( a \in A \) by the definition of this case. Then, conclude \( \text{isol}(w_1d_2, a(\mathbf{d})) = a(\mathbf{d}) \cup \xi(\mathbf{w}, d_2)^2(a)((w_1d_2)^2) \) by the definition of \( \text{isol} \) (Z7).

  Conclude:

  \[
  \begin{align*}
  \tilde{\text{isol}}(w_1xw_2, \alpha)[d/x] & = \text{isol}(w_1xw_2, a(\mathbf{d}))[d/x] \\
  & \overset{Z_9}{=} (a(\mathbf{d}) \cup \xi(\mathbf{w}, xw_2)^2(a)((w_1xw_2)^2))[d/x] \\
  & \overset{Z_2}{=} (a(\mathbf{d}) \cup \xi(\mathbf{w}, d_2)^2(a)((w_1d_2)^2))[d/x] \\
  & \overset{Z_5}{=} a(\mathbf{d})[d/x] \cup \xi(\mathbf{w}, d_2)^2(a)((w_1d_2)^2) \\
  & \overset{Z_7}{=} \tilde{\text{isol}}(w_1d_2, a(\mathbf{d}))[d/x] \\
  \end{align*}
  \]

  **Case:** \( a \notin A \). Observations:

  - Recall \( a \notin A \) by the definition of this case. Then, conclude \( \tilde{\text{isol}}(w_1xw_2, a(\mathbf{d})) = \xi(\mathbf{w}, xw_2)^2(a)((w_1xw_2)^2) \) by the definition of \( \tilde{\text{isol}} \) (Z8).

  - Recall \( a \notin A \) by the definition of this case. Then, conclude \( \tilde{\text{isol}}(w_1d_2, a(\mathbf{d})) = \xi(\mathbf{w}, d_2)^2(a)((w_1d_2)^2) \) by the definition of \( \tilde{\text{isol}} \) (Z9).

  Conclude:
Case: $\tilde{\text{isol}} = \text{isol}$. Conclude $[a \in A \text{ or } a \notin A]$ by ZFC—proceed by case distinction.

Case: $a \in A$. Observations:

- Recall $a \in A$ by the definition of this case. Then, conclude $\tilde{\text{isol}}(w_1xw_2, a(d)) = \xi_{(w_1xw_2)}(a)((w_1xw_2)^\alpha)[d/x]$ by the definition of $\tilde{\text{isol}}$ (Z10).
- Recall $a \notin A$ by the definition of this case. Then, conclude $\tilde{\text{isol}}(w_1dw_2, a(d)) = \xi_{(w_1dw_2)}(a)((w_1dw_2)^\alpha)$ by the definition of $\tilde{\text{isol}}$ (Z11).

Conclude:

Case: $a \notin A$. Observations:

- Recall $a \notin A$ by the definition of this case. Then, conclude $\tilde{\text{isol}}(w_1xw_2, a(d)) = a(d) \sqcup \xi_{(w_1xw_2)}(a)((w_1xw_2)^\alpha)$ by the definition of $\tilde{\text{isol}}$ (Z12).
- Recall $a \notin A$ by the definition of this case. Then, conclude $\tilde{\text{isol}}(w_1dw_2, a(d)) = a(d) \sqcup \xi_{(w_1dw_2)}(a)((w_1dw_2)^\alpha)$ by the definition of $\tilde{\text{isol}}$ (Z13).

Conclude:

Case: $\alpha = \tau$. Conclude:

32
\[ \text{Proposition 11 (Normal form for } \tilde{\text{isol}}\text{-multiactions).} \]

\[ \alpha \in \text{TauFree implies} \]

\[ \text{isol}(w, \alpha) \simeq \bigcup_{i=1}^{n} (a_i(d_i) \cup \xi w_1(a_i)(w^p)) \cup \bigcup_{i=1}^{n'} (\xi w_2(a'_i)(w^p)) \text{ and } \text{Act}(\text{isol}(w, \alpha)) = \bigcup_{i=1}^{n} (a_i \cup \xi w_1(a_i)) \cup \bigcup_{i=1}^{n'} (\xi w_2(a'_i)) \]

\[ \text{isol}(w, \alpha) \simeq \bigcup_{i=1}^{n} (a'_i(d'_i) \cup \xi w_2(a'_i)(w^p)) \cup \bigcup_{i=1}^{n'} (\xi w_1(a_i)(w)) \text{ and } \text{Act}(\text{isol}(w, \alpha)) = \bigcup_{i=1}^{n} (a'_i \cup \xi w_2(a'_i)) \cup \bigcup_{i=1}^{n'} (\xi w_1(a_i)) \]

\[ \text{and } \alpha \simeq \bigcup_{i=1}^{n} a_i(d_i) \cup \bigcup_{i=1}^{n'} a'_i(d'_i) \text{ and } n + n' \geq 1 \]

\[ \text{for some } n, n', a_1, \ldots , a_n, d_1, \ldots , d_n, a'_1, \ldots , a'_{n'}, d'_1, \ldots , d'_{n'}. \]
Proof. Assumptions:

- $\alpha \in \text{TauFree} (Z1)$.

Proceed by induction on the structure of $\alpha$.

**Base:** $[\alpha = a(d) \ or \ \alpha = \tau]$. Proceed by case distinction on the structure of $\alpha$.

**Case:** $\alpha = a(d)$. Conclude $[a \in A \ or \ a \notin A]$ by ZFC—proceed by case distinction.

**Case:** $a \in A$. Assumptions:

- $n$, $n'$, $a_1$, $d_1 = 1, 0$, $a, d$ (Z2).

Observations:

- Recall $a \in A$ by the definition of this case. Then, conclude $\text{isol}(w, a(d)) = a(d) \cup \xi_{w^3}(a)(w^3)$ by the definition of $\text{isol}$ (Z3).
- Recall $a \in A$ by the definition of this case. Then, conclude $\overline{\text{isol}}(w, a(d)) = \overline{\xi_{w^3}(a)(w^3)}$ by the definition of $\overline{\text{isol}}$ (Z4).

**Conclude (Z5):**

$$\text{isol}(w, \alpha) = \text{isol}(w, a(d))$$

**Case:** $a \in A$.

- $Z3$: $a \in A$.

**Conclude (Z6):**

$$\text{ZFC} \implies \text{Act}(\text{isol}(w, \alpha))$$

**Case:** $\text{Act}(\text{isol}(w, a))$.

- $Z3$: $a \cup \xi_{w^3}(a)(w^3)$.

**Conclude (Z7):**

$$\overline{\text{isol}}(w, \alpha) = \overline{\text{isol}}(w, a(d))$$

**Case:** $\overline{\text{isol}}(w, a(d))$.

- $Z3$: $\xi_{w^3}(a)(w^3)$.

**Conclude (Z8):**
$\text{Case: } a \not\in A.$ Assumptions:
- $n, n', a_1', d_1' = 0, 1, a, d$ (Z12).

Observations:
- Recall $a \not\in A$ by the definition of this case. Then, conclude $\text{isol}(w, a(d)) = \bar{\xi}_{w^3}(a)(w^5)$ by the definition of $\text{isol}$ (Z13).
- Recall $a \not\in A$ by the definition of this case. Then, conclude $\overline{\text{isol}}(w, a(d)) = a(d) \sqcup \bar{\xi}_{w^3}(a)(w^5)$ by the definition of $\overline{\text{isol}}$ (Z14).

$\text{Conclude (Z15):}$

\[
\text{isol}(w, \alpha) = \text{isol}(w, a(d)) = \bar{\xi}_{w^3}(a)(w^5) = \bar{\xi}_{w^3}(a_1')(w^5) = \tau \sqcup \bar{\xi}_{w^3}(a_1')(w^5)
\]

Conclude the consequent of this proposition by and-ing the results in Z5, Z6, Z7, Z8, Z9, Z10, and Z11.

Case: $a \not\in A.$ Assumptions:
- $n, n', a_1', d_1' = 0, 1, a, d$ (Z12).

Observations:
- Recall $a \not\in A$ by the definition of this case. Then, conclude $\text{isol}(w, a(d)) = \bar{\xi}_{w^3}(a)(w^5)$ by the definition of $\text{isol}$ (Z13).
- Recall $a \not\in A$ by the definition of this case. Then, conclude $\overline{\text{isol}}(w, a(d)) = a(d) \sqcup \bar{\xi}_{w^3}(a)(w^5)$ by the definition of $\overline{\text{isol}}$ (Z14).

$\text{Conclude (Z15):}$

\[
\text{isol}(w, \alpha) = \text{isol}(w, a(d)) = \bar{\xi}_{w^3}(a)(w^5) = \bar{\xi}_{w^3}(a_1')(w^5) = \tau \sqcup \bar{\xi}_{w^3}(a_1')(w^5)
\]

Conclude the consequent of this proposition by and-ing the results in Z5, Z6, Z7, Z8, Z9, Z10, and Z11.
Hence, this case cannot happen.

• Conclude (Z17):

\[ \text{Act} \left( \overline{\text{isol}}(w, \alpha) \right) = \text{Act} \left( \overline{\text{isol}}(w, a) \right) \]
\[ \overset{\text{Z13}}{=} \text{Act} (\overline{\mathcal{C}}_{WZ}(a))(w^\alpha) \]
\[ \overset{\text{Z12}}{=} \text{Act} (\overline{\mathcal{C}}_{WZ}(a')(w^\beta)) \]
\[ \overset{\text{Act}}{=} \{ \xi_{WZ}(a'_1) \} \]
\[ \overset{\text{ZFC}}{=} \bigcup_{i=1}^n \{ a_i, \xi_{WZ}(a_i) \} \bigcup \bigcup_{i=1}^{n'} \{ \xi_{WZ}(a'_i) \} \]

• Conclude (Z18):

\[ \overline{\text{iso}}(w, \alpha) \]
\[ \overset{\text{Z13}}{=} \text{Act} (a(d) \cup \xi_{WZ}(a)(w^\alpha)) \]
\[ \overset{\text{Z12}}{=} a'_1(d'_i) \bigcup \xi_{WZ}(a'_1)(w^\beta) \]
\[ \overset{\text{MA3}}{\cong} a'_1(d'_i) \bigcup \xi_{WZ}(a'_1)(w^\beta) \bigcup \tau \]
\[ \overset{\text{Z12}}{=} \bigcup_{i=1}^1 (a'_1(d'_i) \cup \xi_{WZ}(a'_1)(w^\beta)) \bigcup \bigcup_{i=1}^n \xi_{WZ}(a_i)(w^\alpha) \]
\[ \overset{\text{Z12}}{=} \bigcup_{i=1}^n (a'_1(d'_i) \cup \xi_{WZ}(a'_1)(w^\beta)) \bigcup \bigcup_{i=1}^n \xi_{WZ}(a_i)(w^\alpha) \]

• Conclude (Z19):

\[ \alpha \]
\[ \overset{\text{Z12}}{=} a'_1(d'_i) \]
\[ \overset{\text{MA3}}{\cong} \tau \bigcup a'_1(d'_i) \]
\[ \overset{\text{Z2}}{=} a'_1(d'_i) \bigcup \bigcup_{i=1}^n a'_1(d'_i) \]
\[ \overset{\text{Z2}}{=} \bigcup_{i=1}^n a_i(d_i) \bigcup \bigcup_{i=1}^{n'} a'_i(d'_i) \]

• Conclude (Z20):

\[ \text{Act}(\alpha) = \text{Act}(a) \overset{\text{Z12}}{=} \text{Act}(a'_1) \overset{\text{Act}}{=} \{ a'_1 \} \overset{\text{ZFC}}{=} \bigcup_{i=1}^n a_i \bigcup \bigcup_{i=1}^{n'} a'_i \overset{\text{Z2}}{=} \bigcup_{i=1}^n a_i \bigcup \bigcup_{i=1}^{n'} a'_i \]

• Conclude $0 + 1 \geq 1$ by ZFC. Then, because $n, n' = 0, 1$ by Z12, conclude $n + n' \geq 1$ (Z21).

Conclude the consequent of this proposition by and-ing the results in Z15, Z16, Z17, Z18, Z19, Z20, and Z21.

**Case:** $\alpha = \tau$. Conclude $\tau \notin \text{TauFree}$ by the definition of TauFree. Then, because $\alpha = \tau$ by the definition of this case, conclude $\alpha \notin \text{TauFree}$—a contradiction, because $\alpha \in \text{TauFree}$ by Z1.

Hence, this case cannot happen.

**Step:** $\alpha = \beta \cup \gamma$. Assumptions:
Observations:

• Recall $\alpha \in \text{TauFree}$ by Z1. Then, because $\alpha = \beta \cup \gamma$ by the definition of this step, conclude $\beta \cup \gamma \in \text{TauFree}$. Then, conclude $\beta, \gamma \in \text{TauFree}$ by the definition of TauFree. Then, conclude

$$\begin{align*}
\text{isol}(w, \beta) &\simeq \bigcup_{i=1}^{m} b_i(e_i) \cup \xi_{\text{var}}(b_i)(w) \cup \bigcup_{i=1}^{n} \tilde{\nu}_{\text{var}}(b'_i)(w) \quad \text{and} \\
\text{Act}(\text{isol}(w, \beta)) &\simeq \bigcup_{i=1}^{m} b_i(e_i) \cup \xi_{\text{var}}(b_i) \cup \bigcup_{i=1}^{n} \tilde{\nu}_{\text{var}}(b'_i) \\
\text{isol}(w, \gamma) &\simeq \bigcup_{i=1}^{m} c_i(f_i) \cup \xi_{\text{var}}(c_i)(w) \cup \bigcup_{i=1}^{l'} c'_i(f'_i) \quad \text{and} \\
\text{Act}(\text{isol}(w, \gamma)) &\simeq \bigcup_{i=1}^{m} c_i(f_i) \cup \xi_{\text{var}}(c_i) \cup \bigcup_{i=1}^{l'} c'_i(f'_i)
\end{align*}$$

by IH (Z22).

Assumptions:

• $n, n' = m + l, m' + l'$ (Z23).

• $\alpha, \delta_i = \begin{cases} b_i, e_i & \text{if } 1 \leq i \leq m \\ c_{i-m}, f_{i-m} & \text{if } m + 1 \leq i \leq m + l \end{cases}$ (Z24).

• $\alpha', \delta'_i = \begin{cases} b'_i, e'_i & \text{if } 1 \leq i \leq m' \\ c'_{i-m'}, f'_{i-m'} & \text{if } m' + 1 \leq i \leq m' + l' \end{cases}$ (Z25).

Observations:

• Conclude (Z26):

$$\begin{align*}
\text{isol}(w, \alpha) &\simeq \bigcup_{i=1}^{m} b_i(e_i) \cup \xi_{\text{var}}(b_i)(w) \\
\text{isol}(w, \beta \cup \gamma) &\simeq \bigcup_{i=1}^{m} b_i(e_i) \cup \xi_{\text{var}}(b_i)(w) \\
\text{isol}(w, \beta) \cup \text{isol}(w, \gamma) &\simeq \bigcup_{i=1}^{m} b_i(e_i) \cup \xi_{\text{var}}(b_i)(w) \cup \bigcup_{i=1}^{m} b'_i(e'_i) \cup \xi_{\text{var}}(b'_i)(w) \\
\text{isol}(w, \beta) &\simeq \bigcup_{i=1}^{m} b_i(e_i) \cup \xi_{\text{var}}(b_i)(w) \\
\text{isol}(w, \beta) &\simeq \bigcup_{i=1}^{m} b_i(e_i) \cup \xi_{\text{var}}(b_i)(w) \cup \bigcup_{i=1}^{m} b'_i(e'_i) \cup \xi_{\text{var}}(b'_i)(w) \\
\text{isol}(w, \gamma) &\simeq \bigcup_{i=1}^{m} c_i(f_i) \cup \xi_{\text{var}}(c_i)(w) \\
\text{isol}(w, \beta \cup \gamma) &\simeq \bigcup_{i=1}^{m} c_i(f_i) \cup \xi_{\text{var}}(c_i)(w) \cup \bigcup_{i=1}^{l'} c'_i(f'_i) \cup \xi_{\text{var}}(c'_i)(w) \\
\text{isol}(w, \beta \cup \gamma) &\simeq \bigcup_{i=1}^{m} c_i(f_i) \cup \xi_{\text{var}}(c_i)(w) \cup \bigcup_{i=1}^{l'} c'_i(f'_i) \cup \xi_{\text{var}}(c'_i)(w)
\end{align*}$$

37
\[ Z_{24}, Z_{25} = \bigcup_{i=1}^{m} \{ a_i, \xi_{w^i}(a_i)(w^i) \} \cup \bigcup_{i=m+1}^{m^\prime} \xi_{w^i}(a_i)(w^i) \]
\[ Z_{24}, Z_{25} = \bigcup_{i=1}^{m} \{ a_i, \xi_{w^i}(a_i)(w^i) \} \cup \bigcup_{i=m+1}^{m^\prime + l} \xi_{w^i}(a_i)(w^i) \]
\[ Z_{23} = \bigcup_{i=1}^{m} \{ a_i, \xi_{w^i}(a_i)(w^i) \} \cup \bigcup_{i=m+1}^{m^\prime} \xi_{w^i}(a_i)(w^i) \]

- Conclude (Z27):
  \[ \text{Act(isol}(w, \alpha) \big)
  \]
  \[ \text{Act(isol}(w, \beta \cup \gamma) \big)
  \]
  \[ \text{Act(isol}(w, \beta) \cup \text{isol}(w, \gamma) \big)
  \]

\[ Z_{22} = \bigcup_{i=1}^{m} \{ b_i, \xi_{w^i}(b_i)(w^i) \} \cup \bigcup_{i=1}^{m^\prime} \xi_{w^i}(b_i)(w^i) \]
\[ Z_{22} = \bigcup_{i=1}^{m} \{ b_i, \xi_{w^i}(b_i)(w^i) \} \cup \bigcup_{i=m+1}^{m^\prime + l} \xi_{w^i}(b_i)(w^i) \]

- Conclude (Z28):
  \[ \text{isol}(w, \alpha) \big]
  \[ \text{isol}(w, \beta \cup \gamma) \big)
  \[ \text{isol}(w, \beta) \cup \text{isol}(w, \gamma) \big)

\[ Z_{22} = \bigcup_{i=1}^{m^\prime} \{ b_i', \xi_{w^i}(b_i')(w^i) \} \cup \bigcup_{i=1}^{m} \xi_{w^i}(b_i)(w^i) \]
\[ Z_{22} = \bigcup_{i=1}^{m^\prime + l} \{ b_i', \xi_{w^i}(b_i')(w^i) \} \cup \bigcup_{i=m+1}^{m^\prime + l} \xi_{w^i}(b_i)(w^i) \]

- Conclude (Z29):
  \[ \text{Act(isol}(w, \alpha) \big)
  \[ \text{Act(isol}(w, \beta \cup \gamma) \big)
  \[ \text{Act(isol}(w, \beta) \cup \text{isol}(w, \gamma) \big)

\[ Z_{22} = \bigcup_{i=1}^{m^\prime} \{ b_i', \xi_{w^i}(b_i')(w^i) \} \cup \bigcup_{i=1}^{m} \xi_{w^i}(b_i)(w^i) \]
\[ Z_{22} = \bigcup_{i=1}^{m^\prime + l} \{ b_i', \xi_{w^i}(b_i')(w^i) \} \cup \bigcup_{i=m+1}^{m^\prime + l} \xi_{w^i}(b_i)(w^i) \]
\[
\begin{align*}
\mathsf{ZFC} & \quad \bigcup_{i=1}^{m+l} \{a_i', \xi_{w^3}(a_i')\} \cup \bigcup_{i=1}^{m+l} \{\xi_{\omega^2}(a_i)\} \\
\mathsf{Z23} & \quad \bigcup_{i=1}^{m'} \{a_i', \xi_{w^3}(a_i')\} \cup \bigcup_{i=1}^{n'} \{\xi_{\omega^2}(a_i)\}
\end{align*}
\]

- Conclude (Z30):

\[
\begin{align*}
\text{Step} & \quad \beta \cup \gamma \\
\mathsf{Z22} & \quad \bigcup_{i=1}^{m} b_i(e_i) \cup \bigcup_{i=1}^{m'} b'_i(e'_i) \cup \bigcup_{i=1}^{l} c_i(f_i) \cup \bigcup_{i=1}^{l'} c'_i(f'_i) \\
\mathsf{Z24, Z25} & \quad \bigcup_{i=1}^{m} a_i(d_i) \cup \bigcup_{i=1}^{m'} a'_i(d'_i) \cup \bigcup_{i=m+1}^{m+l+1} a_i(d_i) \cup \bigcup_{i=m+1}^{m+l+1} a'_i(d'_i) \\
\mathsf{Z23} & \quad \bigcup_{i=1}^{m} a_i(d_i) \cup \bigcup_{i=1}^{m'} a'_i(d'_i)
\end{align*}
\]

- Conclude (Z31):

\[
\begin{align*}
\text{Case} & \quad \text{Act}(\alpha) \\
\mathsf{Z31} & \quad \text{Act}(\beta \cup \gamma) \\
\mathsf{Z22} & \quad \bigcup_{i=1}^{m} b_i(e_i) \cup \bigcup_{i=1}^{m'} b'_i(e'_i) \cup \bigcup_{i=1}^{l} c_i(f_i) \cup \bigcup_{i=1}^{l'} c'_i(f'_i) \\
\mathsf{Z24, Z25} & \quad \bigcup_{i=1}^{m} a_i(d_i) \cup \bigcup_{i=1}^{m'} a'_i(d'_i) \cup \bigcup_{i=m+1}^{m+l+1} a_i(d_i) \cup \bigcup_{i=m+1}^{m+l+1} a'_i(d'_i) \\
\mathsf{ZFC} & \quad \bigcup_{i=1}^{m+l+1} a_i \cup \bigcup_{i=1}^{m+l+1} a'_i \\
\mathsf{Z23} & \quad \bigcup_{i=1}^{m} a_i \cup \bigcup_{i=1}^{m'} a'_i
\end{align*}
\]

- Recall \( m + m' + l + l' \geq 1 \) by Z22. Then, conclude \( m + l + m' + l' \geq 1 \) by ZFC. Then, because \( n \), \( n' = m + l \), \( m' + l' \) by Z23, conclude \( n + n' \geq 1 \) (Z32).

Conclude the consequent of this proposition by and-ing the results in Z26, Z27, Z28, Z29, Z30, Z31, and Z32.

\[\square\]

**Proposition 12.** \( \text{Act}(\alpha) \subseteq \text{dom}(\Xi) \) implies \( \xi_{w^3}(a), \xi_{\omega^2}(a) \notin \text{Act}(\alpha) \)

**Proof.** Assumptions:

- \( \text{Act}(\alpha) \subseteq \text{dom}(\Xi) \) (Z1).

Observations:

- Recall \( \text{Act}(\alpha) \subseteq \text{dom}(\Xi) \) by Z1. Then, because \( \text{dom}(\Xi) = \{ a \mid \langle w, a \rangle \in \text{dom}(\xi) \cap \text{dom}(\bar{\xi}) \} \) by the definition of \( \text{dom} \), conclude \( \text{Act}(\alpha) \subseteq \{ a \mid \langle w, a \rangle \in \text{dom}(\xi) \cap \text{dom}(\bar{\xi}) \} \). Then, conclude \( \text{Act}(\alpha) \subseteq \{ a \mid \langle w, a \rangle \in \text{dom}(\xi) \} \) by ZFC. Then, because \( \text{dom}(\xi) \subseteq \{ 1, 2 \}^* \times A \) by Definition 5 conclude \( \text{Act}(\alpha) \subseteq \{ a \mid \langle w, a \rangle \in \{ 1, 2 \}^* \times A \} \). Then, conclude \( \text{Act}(\alpha) \subseteq A \) by ZFC (Z2).

Recall \( \text{img}(\xi), \text{img}(\bar{\xi}) \subseteq \text{Act} \setminus \{ A \cup \{ \text{tau} \} \} \) by Definition 3. Then, because \( \text{Act}(\alpha) \subseteq A \) by Z2, conclude \( \text{img}(\xi), \text{img}(\bar{\xi}) \subseteq \text{Act} \setminus \{ \text{Act}(\alpha) \cup \{ \text{tau} \} \} \). Then, conclude \( \xi_{w^3}(a), \xi_{\omega^2}(a) \notin \text{Act}(\alpha) \) by ZFC.

\[\square\]
Proposition 13.

1. \( \xi_{w^3}(a) \in \text{Act}(\tilde{w}_A, \alpha) \) and \( \text{Act}(\alpha) \subseteq \text{dom}(\Xi) \) implies \[
\begin{align*}
\text{isol} &= \text{isol} \quad \text{and} \quad \alpha \in A \\
\text{or} \quad \text{isol} &= \text{isol} \quad \text{and} \quad \alpha \notin A
\end{align*}
\]

2. \( \xi_{w^3}(a) \in \text{Act}(\tilde{w}_A, \alpha) \) and \( \text{Act}(\alpha) \subseteq \text{dom}(\Xi) \) implies \[
\begin{align*}
\text{isol} &= \text{isol} \quad \text{and} \quad \alpha \notin A \\
\text{or} \quad \text{isol} &= \text{isol} \quad \text{and} \quad \alpha \in A
\end{align*}
\]

3. \( \text{isol} = \text{isol} \quad \text{and} \quad \alpha \in A \) and \( \text{Act}(\alpha) \subseteq \text{dom}(\Xi) \) implies \( \xi_{w^3}(a) \notin \text{Act}(\tilde{w}_A, \alpha) \)

4. \( \text{isol} = \text{isol} \quad \text{and} \quad \alpha \notin A \) and \( \text{Act}(\alpha) \subseteq \text{dom}(\Xi) \) implies \( \xi_{w^3}(a) \notin \text{Act}(\tilde{w}_A, \alpha) \)

5. \( \text{isol} = \text{isol} \quad \text{and} \quad \alpha \notin A \) and \( \text{Act}(\alpha) \subseteq \text{dom}(\Xi) \) implies \( \xi_{w^3}(a) \notin \text{Act}(\tilde{w}_A, \alpha) \)

6. \( \text{isol} = \text{isol} \quad \text{and} \quad \alpha \in A \) and \( \text{Act}(\alpha) \subseteq \text{dom}(\Xi) \) implies \( \xi_{w^3}(a) \notin \text{Act}(\tilde{w}_A, \alpha) \)

Proof.

1. Assumptions:

- \( \xi_{w^3}(a) \in \text{Act}(\tilde{w}_A, \alpha) \) and \( \text{Act}(\alpha) \subseteq \text{dom}(\Xi) \) (Z1).

Proceed by induction on \( \alpha \).

**Base:** \( \alpha = b(e) \) or \( \alpha = \tau \). Proceed by case distinction on the structure of \( \alpha \).

**Case:** \( \alpha = b(e) \). Observations:

- Recall \( \text{Act}(\alpha) \subseteq \text{dom}(\Xi) \) by Z1. Then, conclude \( \xi_{w^3}(a) \notin \text{Act}(\alpha) \) by Proposition 12. Then, because \( \alpha = b(e) \) by the definition of this case, conclude \( \xi_{w^3}(a) \notin \text{Act}(b(e)) \). Then, because \( \text{Act}(b(e)) = \{b\} \) by the definition of \( \text{Act} \), conclude \( \xi_{w^3}(a) \notin \{b\} \) (Z2).

- Recall \( \text{img}(\xi) \cap \text{img}(\tilde{\xi}) = \emptyset \) by Definition 3. Then, conclude \( \xi_{w^3}(a) \notin \{\xi_{w^3}(b)\} \) by ZFC. Then, because \( \text{Act}(\xi_{w^3}(b)(w^3)) = \{\xi_{w^3}(b)\} \) by the definition of \( \text{Act} \), conclude \( \xi_{w^3}(a) \notin \text{Act}(\xi_{w^3}(b)(w^3)) \) (Z3).

Conclude \( \{b = a \text{ or } b \neq a\} \) by ZFC—proceed by case distinction.

**Case:** \( b = a \). Observations:

- Conclude \( \xi_{w^3}(a) \in \{a\} \cup \{\xi_{w^3}(a)\} \) by ZFC. Then, because \( \text{Act}(a(e)) = \{a\} \) and \( \text{Act}(\xi_{w^3}(a)(w^3)) = \{\xi_{w^3}(a)\} \) by the definition of \( \text{Act} \), conclude \( \xi_{w^3}(a) \in \text{Act}(a(e)) \cup \text{Act}(\xi_{w^3}(a)(w^3)) \). Then, because \( \text{Act}(a(e) \cup \xi_{w^3}(a)(w^3)) = \text{Act}(a(e)) \cup \text{Act}(\xi_{w^3}(a)(w^3)) \) by the definition of \( \text{Act} \), conclude \( \xi_{w^3}(a) \in \text{Act}(a(e) \cup \xi_{w^3}(a)(w^3)) \) (Z4).

Proceed by case distinction on the value of \( \text{isol} \).

**Case:** \( \text{isol} = \text{isol} \). Conclude \( \{b \in A \text{ or } b \notin A\} \) by ZFC—proceed by case distinction.

**Case:** \( b \in A \). Observations:

- Recall \( b \in A \) by the definition of this case. Then, because \( b = a \) by the definition of this case, conclude \( a \in A \). Then, conclude \( \text{isol}(w, a(e)) = a(e) \cup \xi_{w^3}(a)(w^3) \) by the definition of \( \text{isol} \) (Z5).
Recall $\xi_{\text{iso}}(a) \in \text{Act}(a(e) \cup \xi_{\text{iso}}(a)(w^3))$ by Z4. Then, because $\text{iso}(w, a(e)) = a(e) \cup \xi_{\text{iso}}(a)(w^3)$ by Z5, conclude $\xi_{\text{iso}}(a) \in \text{Act}(\text{iso}(w, a(e)))$. Then, because $[b = a$ and $\text{iso} = \text{iso}]$ by the definition of this case, conclude $\xi_{\text{iso}}(a) \in \text{Act}(\text{iso}(w, b(e)))$. Then, because $a = b(e)$ by the definition of this base, conclude $\xi_{\text{iso}}(a) \in \text{Act}(\text{iso}(w, \alpha))$.

Case: $b \notin \mathbb{A}$. Observations:
- Recall $b \notin \mathbb{A}$ by the definition of this case. Then, conclude $\text{iso}(w, b(e)) = \xi_{\text{iso}}(b)(w^3)$ by the definition of $\text{iso}$ (Z6).

Recall $\xi_{\text{iso}}(a) \notin \text{Act}(\xi_{\text{iso}}(b)(w^3))$ by Z3. Then, because $\text{iso}(w, b(e)) = \xi_{\text{iso}}(b)(w^3)$ by Z6, conclude $\xi_{\text{iso}}(a) \notin \text{Act}(\text{iso}(w, b(e)))$. Then, because $[a = b(e)$ by the definition of this step] and $[\text{iso} = \text{iso}]$ by the definition of this case], conclude $\xi_{\text{iso}}(a) \notin \text{Act}(\text{iso}(w, \alpha))$—a contradiction, because $\xi_{\text{iso}}(a) \in \text{Act}(\text{iso}(w, \alpha))$ by Z1.

Hence, this case cannot happen.

Hence, conclude $b \in \mathbb{A}$. Then, because $b = a$ by the definition of this case, conclude $a \in \mathbb{A}$.

Case: $\text{iso} = \text{iso}$. Conclude $[a \in \mathbb{A}$ or $a \notin \mathbb{A}]$ by ZFC—proceed by case distinction.

Case: $b \in \mathbb{A}$. Observations:
- Recall $b \in \mathbb{A}$ by the definition of this case. Then, conclude $\text{iso}(w, b(e)) = \xi_{\text{iso}}(b)(w^3)$ by the definition of $\text{iso}$ (Z7).

Recall $\xi_{\text{iso}}(a) \notin \text{Act}(\xi_{\text{iso}}(b)(w^3))$ by Z3. Then, because $\text{iso}(w, b(e)) = \xi_{\text{iso}}(b)(w^3)$ by Z7, conclude $\xi_{\text{iso}}(a) \notin \text{Act}(\text{iso}(w, b(e)))$. Then, because $[a = b(e)$ by the definition of this step] and $[\text{iso} = \text{iso}]$ by the definition of this case], conclude $\xi_{\text{iso}}(a) \notin \text{Act}(\text{iso}(w, \alpha))$—a contradiction, because $\xi_{\text{iso}}(a) \in \text{Act}(\text{iso}(w, \alpha))$ by Z1.

Hence, this case cannot happen.

Case: $b \notin \mathbb{A}$. Observations:
- Recall $b \notin \mathbb{A}$ by the definition of this case. Then, because $b = a$ by the definition of this case, conclude $a \notin \mathbb{A}$.

Then, conclude $\text{iso}(w, a(e)) = a(e) \cup \xi_{\text{iso}}(a)(w^3)$ by the definition of $\text{iso}$ (Z8).

Recall $\xi_{\text{iso}}(a) \in \text{Act}(a(e) \cup \xi_{\text{iso}}(a)(w^3))$ by Z4. Then, because $\text{iso}(w, a(e)) = a(e) \cup \xi_{\text{iso}}(a)(w^3)$ by Z8, conclude $\xi_{\text{iso}}(a) \in \text{Act}(\text{iso}(w, a(e)))$. Then, because $[b = a$ and $\text{iso} = \text{iso}]$ by the definition of this case, conclude $\xi_{\text{iso}}(a) \in \text{Act}(\text{iso}(w, b(e)))$. Then, because $a = b(e)$ by the definition of this case, conclude $\xi_{\text{iso}}(a) \in \text{Act}(\text{iso}(w, \alpha))$.

Hence, conclude $b \notin \mathbb{A}$. Then, because $b = a$ by the definition of this case, conclude $a \notin \mathbb{A}$.

Hence, conclude $[[\text{iso} = \text{iso}$ and $a \in \mathbb{A}]$ or $[\text{iso} = \text{iso}$ and $a \notin \mathbb{A}]]$.

Case: $b \neq a$. Observations:
- Recall $[\xi : \{1, 2\} \times A \rightarrow \text{Act}(A \cup \tau)]$ by Definition [3] and $[a \neq b$ by the definition of this case]. Then, conclude $\xi_{\text{iso}}(a) \notin \{\xi_{\text{iso}}(b)\}$ by ZFC (Z9).

Recall $[\xi_{\text{iso}}(a) \notin \{b\}$ by Z2 and $[\xi_{\text{iso}}(a) \notin \{\xi_{\text{iso}}(b)\}$ by Z9]. Then, conclude $\xi_{\text{iso}}(a) \notin \{b\} \cup \{\xi_{\text{iso}}(b)\}$ by ZFC. Then, because $[\text{Act}(b(e)) = \{b\}$ and $\text{Act}(\xi_{\text{iso}}(b)(w^3)) = \{\xi_{\text{iso}}(b)\}]$ by the definition of Act, conclude $\xi_{\text{iso}}(a) \notin \text{Act}(b(e)) \cup \text{Act}(\xi_{\text{iso}}(b)(w^3))$. Then, because $\text{Act}(b(e) \cup \xi_{\text{iso}}(b)(w^3)) = \text{Act}(b(e)) \cup \text{Act}(\xi_{\text{iso}}(b)(w^3))$ by the definition of Act, conclude $\xi_{\text{iso}}(a) \notin \text{Act}(b(e) \cup \xi_{\text{iso}}(b)(w^3))$ (Z10).

Proceed by case distinction on the value of $\text{iso}$.

Case: $\text{iso} = \text{iso}$. Conclude $[b \in \mathbb{A}$ or $b \notin \mathbb{A}]$ by ZFC—proceed by case distinction.

Case: $b \in \mathbb{A}$. Observations:
- Recall $b \in \mathbb{A}$ by the definition of this case. Then, conclude $\text{iso}(w, b(e)) = b(e) \cup \xi_{\text{iso}}(b)(w^3)$ by the definition of $\text{iso}$ (Z11).
Recall $\xi_{w^t}(a) \notin \text{Act}(b(e) \cup \xi_{w^t}(b)(w^s))$ by Z10. Then, because $\text{isol}(w, b(e)) = b(e) \cup \xi_{w^t}(b)(w^s)$ by Z11, conclude $\xi_{w^t}(a) \notin \text{Act}(\text{isol}(w, b(e)))$. Then, because $[\alpha = b(e)$ by the definition of this step] and $[\text{isol} = \text{isol}$ by the definition of this case], conclude $\xi_{w^t}(a) \notin \text{Act}(\text{isol}(w, \alpha))$—a contradiction, because $\xi_{w^t}(a) \in \text{Act}(\text{isol}(w, \alpha))$ by Z1.

Hence, this case cannot happen.

Case: $b \notin A$. Observations:

- Recall $b \notin A$ by the definition of this case. Then, conclude $\text{isol}(w, b(e)) = \xi_{w^t}(b)(w^s)$ by the definition of isol (Z12).

Recall $\xi_{w^t}(a) \notin \text{Act}(\xi_{w^t}(b)(w^s))$ by Z3. Then, because $\text{isol}(w, b(e)) = \xi_{w^t}(b)(w^s)$ by Z12, conclude $\xi_{w^t}(a) \notin \text{Act}(\text{isol}(w, b(e)))$. Then, because $[\alpha = b(e)$ by the definition of this step] and $[\text{isol} = \text{isol}$ by the definition of this case], conclude $\xi_{w^t}(a) \notin \text{Act}(\text{isol}(w, \alpha))$—a contradiction, because $\xi_{w^t}(a) \in \text{Act}(\text{isol}(w, \alpha))$ by Z1.

Hence, this case cannot happen.

Case: $\text{isol} = \text{isol}$. Conclude $[b \in A$ or $b \notin A$] by ZFC—proceed by case distinction.

Case: $b \in A$. Observations:

- Recall $b \in A$ by the definition of this case. Then, conclude $\text{isol}(w, b(e)) = \xi_{w^t}(b)(w^s)$ by the definition of isol (Z13).

Recall $\xi_{w^t}(a) \notin \text{Act}(\xi_{w^t}(b)(w^s))$ by Z3. Then, because $\text{isol}(w, b(e)) = \xi_{w^t}(b)(w^s)$ by Z13, conclude $\xi_{w^t}(a) \notin \text{Act}(\text{isol}(w, b(e)))$. Then, because $[\alpha = b(e)$ by the definition of this step] and $[\text{isol} = \text{isol}$ by the definition of this case], conclude $\xi_{w^t}(a) \notin \text{Act}(\text{isol}(w, \alpha))$—a contradiction, because $\xi_{w^t}(a) \in \text{Act}(\text{isol}(w, \alpha))$ by Z1.

Hence, this case cannot happen.

Case: $b \notin A$. Observations:

- Recall $b \notin A$ by the definition of this case. Then, conclude $\text{isol}(w, b(e)) = b(e) \cup \xi_{w^t}(b)(w^s)$ by the definition of isol (Z14).

Recall $\xi_{w^t}(a) \notin \text{Act}(b(e) \cup \xi_{w^t}(b)(w^s))$ by Z10. Then, because $\text{isol}(w, b(e)) = b(e) \cup \xi_{w^t}(b)(w^s)$ by Z14, conclude $\xi_{w^t}(a) \notin \text{Act}(\text{isol}(w, b(e)))$. Then, because $[\alpha = b(e)$ by the definition of this step] and $[\text{isol} = \text{isol}$ by the definition of this case], conclude $\xi_{w^t}(a) \notin \text{Act}(\text{isol}(w, \alpha))$—a contradiction, because $\xi_{w^t}(a) \in \text{Act}(\text{isol}(w, \alpha))$ by Z1.

Hence, this case cannot happen.

Hence, this case cannot happen.

Case: $\alpha = \tau$. Conclude $\xi_{w^t}(\alpha) \notin \emptyset$ by ZFC. Then, because Act($\tau$) = $\emptyset$ by the definition of Act, conclude $\xi_{w^t}(\alpha) \notin \text{Act}(\tau)$. Then, because $\text{isol}(w, \tau) = \tau$ definition of isol, conclude $\xi_{w^t}(\alpha) \notin \text{Act}(\text{isol}(w, \tau))$. Then, because $\alpha = \tau$ by the definition of this case, conclude $\xi_{w^t}(\alpha) \notin \text{Act}(\text{isol}(w, \alpha))$—a contradiction, because $\xi_{w^t}(\alpha) \in \text{Act}(\text{isol}(w, \alpha))$ by Z1. Hence, this case cannot happen.

Step: $\alpha = \beta \cup \gamma$. Assumptions:

- Induction hypothesis (IH):

$$\left[\xi_{w^t}(\hat{a}) \in \text{Act}(\text{isol}(\hat{w}, \hat{a}))\right] \implies \left[\left[\text{isol} = \text{isol} \text{ and } \hat{a} \in A\right] \text{ or } \left[\text{isol} = \text{isol} \text{ and } \hat{a} \notin A\right]\right]$$

for all $\hat{a} \in \{\beta, \gamma\}$

Observations:
3. Assumptions:

Proceed by induction on $\alpha$.

Case: Observations:

- Recall $\xi_{w^3}(a) \in \text{Act}(\overline{\text{isol}}(w, \alpha))$ by Z1. Then, because $\text{Act}(\overline{\text{isol}}(w, \alpha)) = \text{Act}(\overline{\text{isol}}(w, \beta)) \cup \text{Act}(\overline{\text{isol}}(w, \gamma))$ by Z15, conclude $\xi_{w^3}(a) \in \text{Act}(\overline{\text{isol}}(w, \beta)) \cup \text{Act}(\overline{\text{isol}}(w, \gamma))$. Then, conclude $[\xi_{w^3}(a) \in \text{Act}(\overline{\text{isol}}(w, \beta))$ or $\xi_{w^3}(a) \in \text{Act}(\overline{\text{isol}}(w, \gamma))]$ by ZFC. Then, conclude $[[\overline{\text{isol}} = \text{isol}$ and $a \in A]$ or $[[\overline{\text{isol}} = \text{isol}$ and $a \notin A]]$ by IH.

2. Likewise.

3. Assumptions:

- $[\overline{\text{isol}} = \text{isol}$ and $a \in A$ and $\text{Act}(\alpha) \subseteq \text{dom}(\Xi)]$ (Z1).

Proceed by induction on $\alpha$.

Base: $[a = b(e)$ or $\alpha = \tau]$. Proceed by case distinction on the structure of $\alpha$.

Case: Observations:

- Recall $\text{Act}(\alpha) \subseteq \text{dom}(\Xi)$ by Z1. Then, conclude $\xi_{w^3}(a) \notin \text{Act}(\alpha)$ by Proposition [12]. Then, because $a = b(e)$ by the definition of this case, conclude $\xi_{w^3}(a) \notin \text{Act}(b(e))$. Then, because $\text{Act}(b(e)) = \{b\}$ by the definition of Act, conclude $\xi_{w^3}(a) \notin \{b\}$ (Z2).
- Recall $\text{img}(\xi) \cap \text{img}(\overline{\xi}) = \emptyset$ by Definition [3]. Then, conclude $\xi_{w^3}(a) \notin \{\xi_{w^3}(b)\}$ by ZFC (Z3).

- Recall $[\xi_{w^3}(a) \notin \{b\}$ by Z2] and $[\xi_{w^3}(a) \notin \{\xi_{w^3}(b)\}$ by Z3]. Then, conclude $\xi_{w^3}(a) \notin \{b\} \cup \{\xi_{w^3}(b)\}$ by ZFC. Then, because $[\text{Act}(b(e)) = \{b\}$ and $\text{Act}(\xi_{w^3}(b)(w^3)) = \{\xi_{w^3}(b)\}]$ by the definition of Act, conclude $\xi_{w^3}(a) \notin \text{Act}(b(e)) \cup \text{Act}(\xi_{w^3}(b)(w^3))$. Then, because $\text{Act}(b(e)) \cup \xi_{w^3}(b)(w^3)) = \text{Act}(b(e)) \cup \text{Act}(\xi_{w^3}(b)(w^3))$ by the definition of Act, conclude $\xi_{w^3}(a) \notin \text{Act}(b(e)) \cup \xi_{w^3}(b)(w^3)$ (Z4).

Conclude $[b = a$ or $b \neq a]$ by ZFC—proceed by case distinction.

Case: $b = a$. Observations:

- Recall $a \in A$ by Z1. Then, conclude $\text{isol}(w, a(e)) = a(e) \cup \xi_{w^3}(a)(w^3)$ by the definition of isol (Z5).

Recall $[\xi_{w^3}(a) \notin \{b\}$ by Z2] and $[\xi_{w^3}(a) \notin \{\xi_{w^3}(b)\}$ by Z3]. Then, conclude $\xi_{w^3}(a) \notin \{b\} \cup \{\xi_{w^3}(b)\}$ by ZFC. Then, because $b = a$ by the definition of this case, conclude $\xi_{w^3}(a) \notin \{a\} \cup \{\xi_{w^3}(a)\}$. Then, because $[\text{Act}(a(e)) = \{a\}$ and $\text{Act}(\xi_{w^3}(a)(w^3)) = \{\xi_{w^3}(a)\}]$ by the definition of Act, conclude $\xi_{w^3}(a) \notin \text{Act}(a(e)) \cup \text{Act}(\xi_{w^3}(a)(w^3))$. Then, because $\text{Act}(a(e)) \cup \xi_{w^3}(a)(w^3)) = \text{Act}(a(e)) \cup \text{Act}(\xi_{w^3}(a)(w^3))$ by the definition of Act, conclude $\xi_{w^3}(a) \notin \text{Act}(a(e)) \cup \xi_{w^3}(a)(w^3))$ (Z5). Then, because $[b = a$ by the definition of this case] and $[[\overline{\text{isol}} = \text{isol}$ by Z1], conclude $\xi_{w^3}(a) \notin \text{Act}(\overline{\text{isol}}(w, b(e)))$. Then, because $a = b(e)$ by the definition of this case, conclude $\xi_{w^3}(a) \notin \text{Act}(\overline{\text{isol}}(w, \alpha))$.

Case: $b \neq a$. Observations:
Recall \([\xi : \{1,2\} \times A \mapsto Act \setminus (A \cup \{\tau a\})]\) by Definition 3 and \([b \neq a\) by the definition of this case\]. Then, conclude \(\xi_{w^3}(a) \notin \xi_{w^3}(b)\) by ZFC. Then, because \(\text{Act}(\xi_{w^3}(b)(w^3)) = \{\xi_{w^3}(b)\}\) by the definition of Act, conclude \(\xi_{w^3}(a) \notin \text{Act}(\xi_{w^3}(b)(w^3))\) (Z6).

Conclude \([b \in A \text{ or } b \notin A]\) by ZFC—proceed by case distinction.

\textbf{Case:} \(b \in A\). Observations:
- Recall \(b \in A\) by the definition of this case. Then, conclude \(\text{isol}(w, b(e)) = b(e) \cup \xi_{w^3}(b)(w^3)\) by the definition of \(\text{isol}\) (Z7).
- Recall \(\xi_{w^3}(a) \notin \text{Act}(\xi_{w^3}(b)(w^3))\) by Z4. Then, because \(\text{isol}(w, b(e)) = b(e) \cup \xi_{w^3}(b)(w^3)\) by Z7, conclude \(\xi_{w^3}(a) \notin \text{Act}(\text{isol}(w, b(e)))\). Then, because \([a = b(e)\) by the definition of this case\] and \([\text{isol} = \text{isol}\) by Z1, conclude \(\xi_{w^3}(a) \notin \text{Act}(\text{isol}(w, a))\).

\textbf{Case:} \(b \notin A\). Observations:
- Recall \(b \notin A\) by the definition of this case. Then, conclude \(\text{isol}(w, b(e)) = \text{isol}(w, b(w^3))\) by the definition of \(\text{isol}\) (Z8).
- Recall \(\xi_{w^3}(a) \notin \text{Act}(\xi_{w^3}(b)(w^3))\) by Z6. Then, because \(\text{isol}(w, b(e)) = \xi_{w^3}(b)(w^3)\) by Z8, conclude \(\xi_{w^3}(a) \notin \text{Act}(\text{isol}(w, b(e)))\). Then, because \([a = b(e)\) by the definition of this case\] and \([\text{isol} = \text{isol}\) by Z1, conclude \(\xi_{w^3}(a) \notin \text{Act}(\text{isol}(w, a))\).

\textbf{Case:} \(a = \tau\). Conclude \(\xi_{w^3}(a) \notin \emptyset\) by ZFC. Then, because \(\text{Act}(\tau) = \emptyset\) by the definition of Act, conclude \(\xi_{w^3}(a) \notin \text{Act}(\tau)\). Then, because \(\text{isol}(w, \tau) = \tau\) by the definition of \(\text{isol}\), conclude \(\xi_{w^3}(a) \notin \text{Act}(\text{isol}(w, \tau))\). Then, because \(a = \tau\) by the definition of this case, conclude \(\xi_{w^3}(a) \notin \text{Act}(\text{isol}(w, a))\).

\textbf{Step:} \(a = \beta \cup \gamma\). Assumptions:
- Induction hypothesis (IH):

\[
\begin{align*}
\text{isol} = \text{isol} & \quad \text{and} \quad a \in A \\
\xi_{w^3}(a) \notin \text{Act}(\text{isol}(w, a))
\end{align*}
\] for all \(a \in \{\beta, \gamma\}\)

Observations:
- Conclude (Z9):

\[
\begin{align*}
\text{Act}(\text{isol}(w, \beta)) & \cup \text{Act}(\text{isol}(w, \gamma)) \\
\equiv & \quad \text{Act}(\text{isol}(w, \beta)) \cup \text{isol}(w, \gamma) \\
\equiv & \quad \text{Act}(\text{isol}(w, \beta \cup \gamma)) \\
\equiv & \quad \text{Act}(\text{isol}(w, a))
\end{align*}
\]

Recall \([\text{isol} = \text{isol} \quad a \in A]\) by Z1. Then, conclude \(\xi_{w^3}(a) \notin \text{Act}(\text{isol}(w, \beta)), \text{Act}(\text{isol}(w, \gamma))\) by IH. Then, conclude \(\xi_{w^3}(a) \notin \text{Act}(\text{isol}(w, \beta)) \cup \text{Act}(\text{isol}(w, \gamma))\) by ZFC. Then, because \(\text{Act}(\text{isol}(w, \beta)) \cup \text{Act}(\text{isol}(w, \gamma)) = \text{isol}(w, a)\) by Z9, conclude \(\xi_{w^3}(a) \notin \text{Act}(\text{isol}(w, a))\).
Proposition 14 (Auxiliary actions exclude their duals).

1. \( \xi_{w^v}(a) \in \text{Act}(\widetilde{\text{isol}}(w, \alpha)) \) and \( \text{Act}(\alpha) \subseteq \text{dom}(\Xi) \) implies \( \overline{\xi}_{w^v}(a) \notin \text{Act}(\widetilde{\text{isol}}(w, \alpha)) \)

2. \( \xi_{w^v}(a) \in \text{Act}(\widetilde{\text{isol}}(w, \alpha)) \) and \( \text{Act}(\alpha) \subseteq \text{dom}(\Xi) \) implies \( \xi_{w^v}(a) \notin \text{Act}(\widetilde{\text{isol}}(w, \alpha)) \)

PROOF.

1. Assumptions:
   - \( \left[ \xi_{w^v}(a) \in \text{Act}(\widetilde{\text{isol}}(w, \alpha)) \text{ and } \text{Act}(\alpha) \subseteq \text{dom}(\Xi) \right] \) (Z1).

   Observations:
   - Recall \( \left[ \xi_{w^v}(a) \in \text{Act}(\widetilde{\text{isol}}(w, \alpha)) \text{ and } \text{Act}(\alpha) \subseteq \text{dom}(\Xi) \right] \) by Z1. Then, conclude \([\text{isol} = \text{isol} \text{ and } a \in \Lambda] \text{ or } [\text{isol} = \overline{\text{isol}} \text{ and } a \notin \Lambda]\) by Proposition 13 (Z2).

   Recall \([\text{isol} = \text{isol} \text{ and } a \in \Lambda] \text{ or } [\text{isol} = \overline{\text{isol}} \text{ and } a \notin \Lambda]\) by Z2 and \([\text{Act}(\alpha) \subseteq \text{dom}(\Xi) \text{ by Z1}] \). Then, conclude \( \xi_{w^v}(a) \notin \text{Act}(\widetilde{\text{isol}}(w, \alpha)) \) by Proposition 13 (Z2).

2. Likewise.

Proposition 15 (Words separate auxiliary actions).

\[ \left[ \text{Act}(\beta) \subseteq \text{dom}(\Xi) \text{ and } v^u \neq u^v \right] \text{ implies } \xi_{u^v}(c), \overline{\xi}_{u^v}(c) \notin \text{Act}(\text{isol}(v, \beta)) \]

PROOF. Assumptions:

- \( \left[ \text{Act}(\beta) \subseteq \text{dom}(\Xi) \text{ and } v^u \neq u^v \right] \) (Z1).

Proceed by induction on the structure of \( \beta \).

Base: \( [\beta = b(e) \text{ or } \beta = \tau] \). Proceed by case distinction on the structure of \( \beta \).

Case: \( \beta = b(e) \). Observations:

- Recall \( \xi, \overline{\xi} : \{ 1, 2 \}^* \times A \rightarrow \text{Act}(A \cup \{ \text{tau} \}) \) by Definition 3. Then, because \( v^u \neq u^v \) by Z1, conclude \( [\xi_{u^v}(c) \neq \xi_{u^v}(b) \text{ and } \xi_{u^v}(c) \neq \overline{\xi}_{u^v}(b)] \). Then, conclude \( [\xi_{u^v}(c) \notin \{ \xi_{u^v}(b) \} \text{ and } \overline{\xi}_{u^v}(c) \notin \{ \overline{\xi}_{u^v}(b)\}] \) by ZFC (Z2).

- Recall \( \text{img}(\xi) \cap \text{img}(\overline{\xi}) = \emptyset \) by Definition 3. Then, conclude \( [\overline{\xi}_{u^v}(c) \notin \{ \xi_{u^v}(b) \} \text{ and } \xi_{u^v}(c) \notin \{ \overline{\xi}_{u^v}(b)\}] \) by ZFC (Z3).

Conclude \( [b \in A \text{ or } b \notin A] \) by ZFC—proceed by case distinction.

Case: \( b \in A \). Observations:

- Recall \( b \in A \) by the definition of this case. Then, conclude \( \text{isol}(v, b(e)) = b(e) \cup \xi_{u^v}(b)(v^u) \) by the definition of \( \text{isol}(\beta) \).

- Recall \( \text{Act}(\beta) \subseteq \text{dom}(\Xi) \) by Z1. Then, conclude \( \xi_{u^v}(c), \overline{\xi}_{u^v}(c) \notin \text{Act}(\beta) \) by Proposition 12. Then, because \( \beta = b(e) \) by the definition of this case, conclude \( \xi_{u^v}(c), \overline{\xi}_{u^v}(c) \notin \text{Act}(b(e)) \). Then, because \( \text{Act}(b(e)) = \{ b \} \) by the definition of \( \text{Act} \), conclude \( \xi_{u^v}(c), \overline{\xi}_{u^v}(c) \notin \{ b \} \) (Z5).
Recall $[\xi_{\text{st}}(c) \notin \{b\}]$ by Z5 and $[\xi_{\text{st}}(c) \notin \{\xi_{\text{st}}(b)\}]$ by Z2. Then, conclude $\xi_{\text{st}}(c) \notin \{b\} \cup \{\xi_{\text{st}}(b)\}$ by ZFC (Z6).

Recall $[\xi_{\text{st}}(c) \notin \{b\}]$ by Z5 and $[\xi_{\text{st}}(c) \notin \{\xi_{\text{st}}(b)\}]$ by Z3. Then, conclude $\xi_{\text{st}}(c) \notin \{b\} \cup \{\xi_{\text{st}}(b)\}$ by ZFC (Z7).

Recall $[\xi_{\text{st}}(c) \notin \{b\} \cup \{\xi_{\text{st}}(b)\}]$ by Z6 and $[\xi_{\text{st}}(c) \notin \{b\} \cup \{\xi_{\text{st}}(b)\}]$ by Z7. Then, conclude $\xi_{\text{st}}(c) , [\xi_{\text{st}}(c) \notin \{b\} \cup \{\xi_{\text{st}}(b)\}]$ by ZFC. Then, because $[\text{Act}(b(e)) = \{b\}$ and $\text{Act}(\xi_{\text{st}}(b)(\nu)) = \{\xi_{\text{st}}(b)\}]$ by the definition of Act, conclude $\xi_{\text{st}}(c) , [\xi_{\text{st}}(c) \notin \{b\} \cup \{\xi_{\text{st}}(b)\}]$ by ZFC. Then, because $\text{Act}(b(e) \cup \xi_{\text{st}}(b)(\nu)) = \text{Act}(b(e)) \cup \text{Act}(\xi_{\text{st}}(b)(\nu))$ by the definition of Act, conclude $\xi_{\text{st}}(c) , [\xi_{\text{st}}(c) \notin \{b\} \cup \{\xi_{\text{st}}(b)\}]$ by Z4. Then, because $\text{isol}(v, b(e)) = b(e) \cup \xi_{\text{st}}(b)(\nu)$ by Z4, conclude $\xi_{\text{st}}(c) , [\xi_{\text{st}}(c) \notin \text{Act}(\text{isol}(v, b(e)))$ by Z5. Then, because $\beta = b(e)$ by the definition of this case, conclude $\xi_{\text{st}}(c) , [\xi_{\text{st}}(c) \notin \text{Act}(\text{isol}(v, \beta))$. 

Case: $b \notin A$. Observations:

Recall $b \notin A$ by the definition of this case. Then, conclude $\text{isol}(v, b(e)) = \xi_{\text{st}}(b)(\nu)$ by the definition of isolated (Z8).

Recall $[\xi_{\text{st}}(c) \notin \{\xi_{\text{st}}(b)\}]$ by Z3 and $[\xi_{\text{st}}(c) \notin \{\xi_{\text{st}}(b)\}]$ by Z3. Then, conclude $\xi_{\text{st}}(c) , [\xi_{\text{st}}(c) \notin \{\xi_{\text{st}}(b)\}]$ by ZFC. Then, because $\text{Act}(\xi_{\text{st}}(b)(\nu)) = \{\xi_{\text{st}}(b)\}$ by the definition of Act, conclude $\xi_{\text{st}}(c) , [\xi_{\text{st}}(c) \notin \text{Act}(\xi_{\text{st}}(b)(\nu))$ by Z4. Then, because $\text{isol}(v, b(e)) = \xi_{\text{st}}(b)(\nu)$ by Z4, conclude $\xi_{\text{st}}(c) , [\xi_{\text{st}}(c) \notin \text{Act}(\text{isol}(v, b(e)))$ by Z5. Then, because $\beta = b(e)$ by the definition of this case, conclude $\xi_{\text{st}}(c) , [\xi_{\text{st}}(c) \notin \text{Act}(\text{isol}(v, \beta))$. 

Case: $\beta = \tau$. Conclude $\xi_{\text{st}}(c) , [\xi_{\text{st}}(c) \notin \emptyset$ by ZFC. Then, because $\text{Act}(\tau) = \emptyset$ by the definition of Act, conclude $\xi_{\text{st}}(c) , [\xi_{\text{st}}(c) \notin \text{Act}(\tau)$ by Z7. Then, because $\text{isol}(v, \tau) = \tau$ by the definition of isolated, conclude $\xi_{\text{st}}(c) , [\xi_{\text{st}}(c) \notin \text{Act}(\text{isol}(v, \tau))$. Then, because $\beta = \tau$ by the definition of this case, conclude $\xi_{\text{st}}(c) , [\xi_{\text{st}}(c) \notin \text{Act}(\text{isol}(v, \beta))$. 

Step: $\beta = \beta_1 \cup \beta_2$. Assumptions:

Induction hypothesis (IH):

$$\left[\text{Act}(\beta) \subseteq \text{dom}(\Xi) \text{ and } \nu^\sharp \neq \bar{\nu}^\sharp \right] \text{ implies } \left[\xi_{\text{st}}(c) , [\xi_{\text{st}}(c) \notin \text{Act}(\text{isol}(v, \beta)) \right] \text{ for all } \beta \in \{\beta_1, \beta_2\}$$

Observations:

Recall $\text{Act}(\beta) \subseteq \text{dom}(\Xi)$ by Z1. Then, because $\beta = \beta_1 \cup \beta_2$ by the definition of this step, conclude $\text{Act}(\beta_1 \cup \beta_2) \subseteq \text{dom}(\Xi)$. Then, because $\text{Act}(\beta_1 \cup \beta_2) = \text{Act}(\beta_1) \cup \text{Act}(\beta_2)$ by the definition of Act, conclude $\text{Act}(\beta_1) \cup \text{Act}(\beta_2) \subseteq \text{dom}(\Xi)$. Then, conclude $\text{Act}(\beta_1) , [\text{Act}(\beta_2) \subseteq \text{dom}(\Xi)$ by ZFC (Z9).

Conclude (Z10):

$$\begin{align*}
\text{Act}(\text{isol}(v, \beta_1)) \cup \text{Act}(\text{isol}(v, \beta_2)) \\
\equiv \\
\text{Act}(\text{isol}(v, \beta_1) \cup \text{isol}(v, \beta_2)) \\
\equiv \\
\text{Act}(\text{isol}(v, \beta_1 \cup \beta_2)) \\
\text{Step} \\
\equiv \\
\text{Act}(\text{isol}(v, \beta))
\end{align*}$$

Recall $[\text{Act}(\beta_1) , [\text{Act}(\beta_2) \subseteq \text{dom}(\Xi)$ by Z9 and $[\nu^\sharp \neq \bar{\nu}^\sharp$ by Z1]. Then, conclude $\xi_{\text{st}}(c) , [\xi_{\text{st}}(c) \notin \text{Act}(\text{isol}(v, \beta_1)) , [\text{Act}(\text{isol}(v, \beta_2))$ by IH. Then, conclude $\xi_{\text{st}}(c) , [\xi_{\text{st}}(c) \notin \text{Act}(\text{isol}(v, \beta_1)) \cup \text{Act}(\text{isol}(v, \beta_2))$ by ZFC. Then, because $\text{Act}(\text{isol}(v, \beta_1)) \cup \text{Act}(\text{isol}(v, \beta_2)) = \text{Act}(\text{isol}(v, \beta))$, conclude $\xi_{\text{st}}(c) , [\xi_{\text{st}}(c) \notin \text{Act}(\text{isol}(v, \beta))$. 

\[\square\]
proof (of proposition 2). assumptions:

- $[\alpha \in \text{TauFree and } \text{Act}(\alpha) \subseteq \text{dom}(\Xi)]$ (Z1).

observations:

- recall $\alpha \in \text{TauFree}$ by Z1. Then, conclude

  \[
  \text{iisol}(w, \alpha) \simeq \bigsqcup_{i=1}^{n} a_i (\xi, \lambda_i (a_i) (w^\beta)) \cup \bigsqcup_{i=1}^{n'} \xi_{w^\beta} (a_i') (w^\beta) \text{ and } \text{Act}(\text{iisol}(w, \alpha)) = \bigsqcup_{i=1}^{n} \{a_i, \xi_{w^\beta} (a_i)\} \cup \bigsqcup_{i=1}^{n'} \{\xi_{w^\beta} (a_i')\} \text{ and } n + n' \geq 1
  \]

  by proposition [1](Z2).

- conclude $\tau \notin \text{Act}(\text{Ptr} \cup \{\text{tau}\})$ by ZFC. then, because $\text{img}(\xi) \subseteq \text{Act}(\text{Ptr} \cup \{\text{tau}\})$ by definition [2], conclude $\tau \notin \text{img}(\xi)$, $\text{img}(\xi) = \text{Act}(\text{iisol}(w, \alpha))$ by ZFC. then, conclude $[\tau (\text{iisol}(\xi_{w^\beta} (a_i) (w^\beta)) \simeq \xi_{w^\beta} (a_i) (w^\beta) \text{ and } \tau (\xi_{w^\beta} (a_i') (w^\beta)) \simeq \xi_{w^\beta} (a_i') (w^\beta)]$

  by H3 (Z3).

- conclude $[\partial_{\text{img}(\Xi) \cup \text{img}(\xi)} (\xi_{w^\beta} (a_i) (w^\beta)) \simeq \xi_{w^\beta} (a_i) (w^\beta) \text{ and } \partial_{\text{img}(\Xi) \cup \text{img}(\xi)} (\xi_{w^\beta} (a_i') (w^\beta)) \simeq \xi_{w^\beta} (a_i') (w^\beta)]$ by B3. Then, because $\text{img}(\Xi) = \text{img}(\xi) \cup \text{img}(\xi)$ by the definition of $\text{img}$, conclude $[\partial_{\text{img}(\Xi)} (\xi_{w^\beta} (a_i) (w^\beta)) \simeq \delta \text{ and } \partial_{\text{img}(\Xi)} (\xi_{w^\beta} (a_i') (w^\beta)) \simeq \delta]$ (Z4).

Recall $n + n' \geq 1$ by Z2. Then, conclude $[n \geq 1 \text{ or } n' \geq 1]$ by ZFC—proceed by case distinction.

Case: $n \geq 1$. observations:

- recall $n \geq 1$ by the definition of this case. Then, conclude $\xi_{w^\beta} (a_n) \in \bigsqcup_{i=1}^{n} \{a_i, \xi_{w^\beta} (a_i)\} \cup \bigsqcup_{i=1}^{n'} \{\xi_{w^\beta} (a_i')\}$ by ZFC. then, because $\text{Act}(\text{iisol}(w, \alpha)) = \bigsqcup_{i=1}^{n} \{a_i, \xi_{w^\beta} (a_i)\} \cup \bigsqcup_{i=1}^{n'} \{\xi_{w^\beta} (a_i')\}$ by Z2, conclude $\xi_{w^\beta} (a_n) \in \text{Act}(\text{iisol}(w, \alpha))$ (Z5).

- recall $[\xi_{w^\beta} (a_n) \in \text{Act}(\text{iisol}(w, \alpha))$ by Z5] and $[\text{Act}(\alpha) \subseteq \text{dom}(\Xi)]$ by Z1]. Then, conclude $\xi_{w^\beta} (a_n) \notin \text{Act}(\text{iisol}(w, \alpha))$ by proposition [1]. Then, because $\text{Act}(\text{iisol}(w, \alpha)) = \bigsqcup_{i=1}^{n} \{a_i, \xi_{w^\beta} (a_i)\} \cup \bigsqcup_{i=1}^{n'} \{\xi_{w^\beta} (a_i')\}$ by Z2, conclude $\xi_{w^\beta} (a_n) \notin \bigsqcup_{i=1}^{n} \{a_i, \xi_{w^\beta} (a_i)\} \cup \bigsqcup_{i=1}^{n'} \{\xi_{w^\beta} (a_i')\}$. Then, conclude

  \[
  \text{comm} (\text{iisol}(w, \alpha)) = \bigsqcup_{i=1}^{n'} \{a_i, \xi_{w^\beta} (a_i)\} \cup \bigsqcup_{i=1}^{n'} \{\xi_{w^\beta} (a_i')\}
  \]

  by the definition of $\text{comm}$ (Z6).

Conclude:

\[
\begin{align*}
\text{Z1} & \implies \text{Z2} \\
\text{Z2} & \implies \text{comm} \\
\text{X} & \implies \text{comm} \\
\text{comm} & \implies \text{comm}
\end{align*}
\]
\[ \delta_{\text{img}(\Xi)}(\mathcal{T}_{\text{tau}}(\alpha')) \cap \mathcal{T}_{\text{w}t}(\alpha_n)(w^\beta) \]

Proof (of Proposition 3). Assumptions:
• \([\beta, \gamma \in \text{TauFree and } \text{Act}(\beta), \text{Act}(\gamma) \subseteq \text{dom}(\Xi) \text{ and } v^\dagger \neq u^\dagger]\) (Z1).

Observations:
• Recall \(\beta, \gamma \in \text{TauFree}\) by Z1. Then, conclude

\[
\begin{align*}
\text{isol}(v, \beta) &\simeq \bigsqcup_{i=1}^{n_1} (b_i(e_i) \cup \xi_{uc}(b_i)(v^\dagger)) \cup \bigsqcup_{i=1}^{m'} \xi_{uc}(b'_i)(v^\dagger) \quad \text{and} \\
\text{Act(isol}(v, \beta)) &\subseteq \bigsqcup_{i=1}^{n_1} \{b_i, \xi_{uc}(b_i)\} \cup \bigsqcup_{i=1}^{m'} \{\xi_{uc}(b'_i)\} \quad \text{and } m + m' \geq 1
\end{align*}
\]

and

\[
\begin{align*}
\text{isol}(u, \gamma) &\simeq \bigsqcup_{i=1}^{l'} (c'_i(f_i) \cup \xi_{uc}(c'_i)(u^\dagger)) \cup \bigsqcup_{i=1}^{l'} \xi_{uc}(c'_i)(u^\dagger) \quad \text{and} \\
\text{Act(isol}(u, \gamma)) &\subseteq \bigsqcup_{i=1}^{l'} \{c'_i, \xi_{uc}(c'_i)\} \cup \bigsqcup_{i=1}^{l'} \{\xi_{uc}(c'_i)\} \quad \text{and } l + l' \geq 1
\end{align*}
\]

by Proposition [11](Z2).

• Recall \([\text{Act}(\beta) \subseteq \text{dom}(\Xi) \text{ and } v^\dagger \neq u^\dagger]\) by Z1. Then, conclude \(\xi_{uc}(c_i), \xi_{uc}(c'_i) \notin \text{Act(isol}(v, \beta))\) by Proposition [15](Z3).

• Conclude \(\tau_\text{u} \notin \text{Act}(A \cup \{\tau_\text{u}\})\) by ZFC. Then, because \(\text{img}(\xi), \text{img}(\xi) \subseteq \text{Act}(A \cup \{\tau_\text{u}\})\) by Definition [3], conclude \(\tau_\text{u} \notin \text{img}(\xi), \text{img}(\xi)\). Then, conclude \([\tau_\text{u} \neq \xi_{uc}(c_i) \text{ and } \tau_\text{u} \neq \xi_{uc}(c'_i)]\) by ZFC. Then, conclude \([T(\tau_\text{u})(\xi_{uc}(c_i)(u^\dagger)) \simeq \xi_{uc}(c_i)(u^\dagger) \text{ and } T(\tau_\text{u})(\xi_{uc}(c'_i)(u^\dagger)) \simeq \xi_{uc}(c'_i)(u^\dagger)]\) by H3 (Z4).

• Conclude \([\partial_{\text{img}(\xi)} \xi_{uc}(c_i)(u^\dagger) \simeq \delta \text{ and } \partial_{\text{img}(\xi)} \xi_{uc}(c'_i)(u^\dagger) \simeq \delta]\) by B3. Then, because \(\text{img}(\xi) = \text{img}(\xi) \cup \text{img}(\xi)\) by the definition of \(\text{img}\), conclude \([\partial_{\text{img}(\xi)} \xi_{uc}(c_i)(u^\dagger) \simeq \delta \text{ and } \partial_{\text{img}(\xi)} \xi_{uc}(c'_i)(u^\dagger) \simeq \delta]\) (Z5).

Recall \(l + l' \geq 1\) by Z2. Then, conclude \([l \geq 1 \text{ or } l' \geq 1]\) by ZFC—proceed by case distinction.

Case: \(l \geq 1\). Observations:
• Recall \(l \geq 1\) by the definition of this case. Then, conclude \(\xi_{uc}(c_i) \in \bigsqcup_{i=1}^{l} \{c'_i, \xi_{uc}(c'_i)\} \cup \bigsqcup_{i=1}^{l'} \{\xi_{uc}(c'_i)\}\) by ZFC. Then, because \(\text{Act(isol}(u, \gamma)) = \bigsqcup_{i=1}^{l} \{c'_i, \xi_{uc}(c'_i)\} \cup \bigsqcup_{i=1}^{l'} \{\xi_{uc}(c'_i)\}\) by Z2, conclude \(\xi_{uc}(c_i) \in \text{Act(isol}(u, \gamma))\) (Z6).

• Recall \(\xi_{uc}(c_i) \in \text{Act(isol}(u, \gamma))\) by Z6 and \([\text{Act}(\gamma) \subseteq \text{dom}(\Xi)\) by Z1]. Then, conclude \(\xi_{uc}(c_i) \notin \text{Act(isol}(u, \gamma))\) by Proposition [14](Z7).

• Recall \([\xi_{uc}(c_i) \notin \text{Act(isol}(v, \beta)) \text{ and } \xi_{uc}(c_i) \notin \text{Act(isol}(u, \gamma))\) by Z7]. Then, conclude \(\xi_{uc}(c_i) \notin \text{Act(isol}(v, \beta)) \cup \text{Act(isol}(u, \gamma))\) by ZFC. Then, because

\[
\begin{align*}
\text{Act(isol}(v, \beta)) &\subseteq \bigsqcup_{i=1}^{n_1} \{b_i, \xi_{uc}(b_i)\} \cup \bigsqcup_{i=1}^{m'} \{\xi_{uc}(b'_i)\} \\
\text{and } \text{Act(isol}(u, \gamma)) &\subseteq \bigsqcup_{i=1}^{l'} \{c'_i, \xi_{uc}(c'_i)\} \cup \bigsqcup_{i=1}^{l'} \{\xi_{uc}(c'_i)\}
\end{align*}
\]

by Z2, conclude

\[
\xi_{uc}(c_i) \notin \bigsqcup_{i=1}^{n_1} \{b_i, \xi_{uc}(b_i)\} \cup \bigsqcup_{i=1}^{m'} \{\xi_{uc}(b'_i)\} \cup \bigsqcup_{i=1}^{l'} \{c'_i, \xi_{uc}(c'_i)\} \cup \bigsqcup_{i=1}^{l'} \{\xi_{uc}(c'_i)\}
\]

Then, conclude

\[
\begin{align*}
\mathcal{C}(\xi_{uc}(c_i) \notin \text{tau}(v^\dagger) \cap \text{dom}(\xi) \cap \text{dom}(\xi)) &\subseteq \\
\bigsqcup_{i=1}^{n_1} \{b_i, \xi_{uc}(b_i)(v^\dagger)\} \cup \bigsqcup_{i=1}^{m'} \{\xi_{uc}(b'_i)(v^\dagger)\} \cup \\
\bigsqcup_{i=1}^{l'} \{c'_i(\gamma_i) \cup \xi_{uc}(c'_i)(u^\dagger)\} \cup \bigsqcup_{i=1}^{l'} \{\xi_{uc}(c'_i)(u^\dagger)\} \simeq \gamma' \cup \xi_{uc}(c_i)(u^\dagger)
\end{align*}
\]

by the definition of \(\mathcal{C}\) (Z8).
Case: $l' \geq 1$. Observations:

- Recall $l' \geq 1$ by the definition of this case. Then, conclude $\xi_{u'}(c_{i'}') \in \bigcup_{i'=1}^{l'} \{c_i', \xi_{u'}(c_{i'})\} \cup \bigcup_{i=1}^{l} \{\xi_{u'}(c_i)\}$ by ZFC. Then, because $\text{Act}(\overline{\text{sol}}(u, \gamma)) = \bigcup_{i=1}^{l'} \{c_i', \xi_{u'}(c_{i'})\} \cup \bigcup_{i=1}^{l} \{\xi_{u'}(c_i)\}$ by Z2, conclude $\xi_{u'}(c_{i'}') \in \text{Act}(\overline{\text{sol}}(u, \gamma))$ by Proposition 1 (Z9).

- Recall $\overline{\text{sol}}(c_{i'}') \not\in \text{Act}(\overline{\text{sol}}(u, \gamma))$ by Z9 and $[\text{Act}(\gamma) \subseteq \text{dom}(\overline{\xi}) \cap \text{dom}(\overline{\xi})]$ by Z1. Then, conclude $\overline{\text{sol}}(c_{i'}') \not\in \text{Act}(\overline{\text{sol}}(u, \gamma))$ by Z3. Then, conclude $\overline{\text{sol}}(c_{i'}') \not\in \text{Act}(\overline{\text{sol}}(u, \gamma)) \cup \text{Act}(\overline{\text{sol}}(u, \gamma))$ by ZFC. Then, because $\text{Act}(\overline{\text{sol}}(v, \beta)) = \bigcup_{i=1}^{m} \{b_i, \xi_{v'}(b_i)\} \cup \bigcup_{i=1}^{m} \{\xi_{v'}(b_i')\}$ and $\text{Act}(\overline{\text{sol}}(u, \gamma)) = \bigcup_{i=1}^{l'} \{c_i', \xi_{u'}(c_{i'})\} \cup \bigcup_{i=1}^{l} \{\xi_{u'}(c_i)\}$.

by Z2, conclude $\overline{\text{sol}}(c_{i'}') \not\in \bigcup_{i=1}^{m} \{b_i, \xi_{v'}(b_i)\} \cup \bigcup_{i=1}^{m} \{\xi_{v'}(b_i')\} \cup \bigcup_{i=1}^{l'} \{c_i', \xi_{u'}(c_{i'})\} \cup \bigcup_{i=1}^{l} \{\xi_{u'}(c_i)\}$.
Then, conclude
\[
\overset{\text{Conclude:}}{Z_{11}} \H_4 \delta
\]
\[
\overset{\text{by the definition of } \mathcal{C}}{(\Xi)}
\]

Conclude:
\[
?_{\text{(isol}(v , \beta) \mid \text{isol}(u , \gamma))}
\]
\[
\overset{\text{SMA}}{\cong}
?_{\text{(isol}(v , \beta) \cup \overline{\text{isol}(u , \gamma))}}
\]
\[
\overset{\text{TauFree}}{\cong}
\overset{\text{Act(}\beta\text{)}}{?_{\text{(}\cup \overline{\text{dom}}(\Xi))}}
\]
\[
\overset{\text{comm}}{\cong}
\overset{\text{Act(}\gamma\text{)}}{?_{\text{(}\cup \overline{\text{dom}}(\Xi))}}
\]
\[
\overset{\text{Z11}}{\cong}
\overset{\text{SMA}}{\cong}
\overset{\text{H4}}{\cong}
\overset{\text{H4}}{\cong}
\overset{\text{R2}}{\cong}
\overset{\text{R2}}{\cong}
\overset{\text{Z4}}{\cong}
\overset{\text{Z4}}{\cong}
\overset{\text{Z4}}{\cong}
\delta
\]

**Proof (of Proposition 4):** Assumptions:

- \([\beta, \gamma \in \text{TauFree and } \text{Act(}\beta\text{), } \text{Act(}\gamma\text{)} \subseteq \text{dom}(\Xi) \text{ and } e \neq f]\) (Z1).

Observations:
Recall $\beta, \gamma \in \text{TauFree}$ by Z1. Then, conclude
\[
\begin{align*}
\text{isol}(\text{wfu}, \beta) &\simeq \bigcup_{i=1}^{m_{l'}} \{b_i(t_i) \cup \xi_{(\text{wfu})(\beta)}((\text{wfu}))\} \cup \bigcup_{i=1}^{m'} \xi_{(\text{wfu})(\beta)}(b'_i((\text{wfu}))) \\
\text{and } \text{Act}(\text{isol}(\text{wfu}, \beta)) &\subseteq \bigcup_{i=1}^{m_{l'}} \{b_i, \xi_{(\text{wfu})(\beta)}(b'_i)\} \cup \bigcup_{i=1}^{m'} \xi_{(\text{wfu})(\beta)}(b'_i) \\
\text{and } \text{Act}(\beta) &\subseteq \bigcup_{i=1}^{m_{l'}} \{b_i\} \cup \bigcup_{i=1}^{m'} \{b'_i\} \text{ and } m+m' \geq 1
\end{align*}
\] and
\[
\begin{align*}
\text{isol}(\text{wfu}, \gamma) &\simeq \bigcup_{i=1}^{l'} \{c'_i(t'_i) \cup \xi_{(\text{wfu})(\gamma)}((\text{wfu}))\} \cup \bigcup_{i=1}^{l} \xi_{(\text{wfu})(\gamma)}(c_i((\text{wfu}))) \\
\text{and } \text{Act}(\text{isol}(\text{wfu}, \gamma)) &\subseteq \bigcup_{i=1}^{l'} \{c'_i, \xi_{(\text{wfu})(\gamma)}(c'_i)\} \cup \bigcup_{i=1}^{l} \xi_{(\text{wfu})(\gamma)}(c_i) \\
\text{and } \text{Act}(\gamma) &\subseteq \bigcup_{i=1}^{l'} \{c'_i\} \cup \bigcup_{i=1}^{l} \{c_i\} \text{ and } l+l' \geq 1
\end{align*}
\] by Proposition [11](Z2).

Conclude $\text{tau} \notin \text{Act}(A \cup \{\text{tau}\})$ by ZFC. Then, because $\text{img}(\xi), \text{img}(\xi) \subseteq \text{Act}(A \cup \{\text{tau}\})$ by Definition [3], conclude $\text{tau} \notin \text{img}(\xi), \text{img}(\xi)$. Then, conclude $\text{tau} \neq \xi_{(\text{wfu})(\gamma)}(c'_i)$ and $\text{tau} \neq \xi_{(\text{wfu})(\gamma)}(c'_i)$ by ZFC. Then, conclude
\[
\begin{align*}
\mathcal{F}_{(\text{tau})}(\xi_{(\text{wfu})(\gamma)}(c_i)((\text{wfu}))) &\simeq \xi_{(\text{wfu})(\gamma)}(c_i)((\text{wfu}))) \\
\text{and } \mathcal{F}_{(\text{tau})}(\xi_{(\text{wfu})(\gamma)}(c'_i)((\text{wfu}))) &\simeq \xi_{(\text{wfu})(\gamma)}(c'_i)((\text{wfu})))
\end{align*}
\]
by H3 (Z3).

Conclude
\[
\begin{align*}
\partial_{\text{img}(\xi) \cup \text{img}(\xi)}(\xi_{(\text{wfu})(\gamma)}(c_i)((\text{wfu}))) &\simeq \delta \\
\text{and } \partial_{\text{img}(\xi) \cup \text{img}(\xi)}(\xi_{(\text{wfu})(\gamma)}(c'_i)((\text{wfu}))) &\simeq \delta
\end{align*}
\] by B3. Then, because $\text{img}(\xi) = \text{img}(\xi) \cup \text{img}(\xi)$ by the definition of $\text{img}$, conclude $\left[\partial_{\text{img}(\xi)}(\xi_{(\text{wfu})(\gamma)}(c_i)((\text{wfu}))) \simeq \delta \text{ and } \partial_{\text{img}(\xi)}(\xi_{(\text{wfu})(\gamma)}(c'_i)((\text{wfu}))) \simeq \delta\right] = (Z4)$.

Recall $l+l' \geq 1$ by Z2. Then, conclude $[l \geq 1 \text{ or } l' \geq 1]$ by ZFC—proceed by case distinction.

**Case:** $l \geq 1$. Observations:

Recall $\text{Act}(\beta) \subseteq \text{dom}(\xi)$. Then, conclude $\xi_{(\text{wfu})(\gamma)}(c_i) \notin \text{Act}(\beta)$ by Proposition [12]. Then, because $\text{Act}(\beta) = \bigcup_{i=1}^{m_{l'}} \{b_i\} \cup \bigcup_{i=1}^{m'} \{b'_i\}$, conclude $\xi_{(\text{wfu})(\gamma)}(c_i) \notin \bigcup_{i=1}^{m_{l'}} \{b_i\} \cup \bigcup_{i=1}^{m'} \{b'_i\}$. Then, conclude $\xi_{(\text{wfu})(\gamma)}(c_i) \notin \bigcup_{i=1}^{m_{l'}} \{b_i\}$ by ZFC (Z5).

Recall $l \geq 1$ by the definition of this case. Then, conclude $\xi_{(\text{wfu})(\gamma)}(c_i) \in \bigcup_{i=1}^{m_{l'}} \{c'_i, \xi_{(\text{wfu})(\gamma)}(c'_i)\} \cup \bigcup_{i=1}^{l} \xi_{(\text{wfu})(\gamma)}(c_i)$ by ZFC. Then, because $\text{Act}(\text{isol}(\text{wfu}, \gamma)) = \bigcup_{i=1}^{l} \{c'_i, \xi_{(\text{wfu})(\gamma)}(c'_i)\} \cup \bigcup_{i=1}^{l} \{\xi_{(\text{wfu})(\gamma)}(c_i)\}$ by Z2, conclude $\xi_{(\text{wfu})(\gamma)}(c_i) \in \text{Act}(\text{isol}(\text{wfu}, \gamma))$ (Z6).

Recall $\xi_{(\text{wfu})(\gamma)}(c_i) \in \text{Act}(\text{isol}(\text{wfu}, \gamma))$ by Z6 and $[\text{Act}(\gamma) \subseteq \text{dom}(\xi) \text{ by Z1}]$. Then, conclude $\xi_{(\text{wfu})(\gamma)}(c_i) \notin \text{Act}(\text{isol}(\text{wfu}, \gamma))$ by Proposition [13]. Then, because $\text{Act}(\text{isol}(\text{wfu}, \gamma)) = \bigcup_{i=1}^{l} \{c'_i, \xi_{(\text{wfu})(\gamma)}(c'_i)\} \cup \bigcup_{i=1}^{l} \xi_{(\text{wfu})(\gamma)}(c_i)$ by Z2, conclude $\xi_{(\text{wfu})(\gamma)}(c_i) \notin \bigcup_{i=1}^{l} \{c'_i, \xi_{(\text{wfu})(\gamma)}(c'_i)\} \cup \bigcup_{i=1}^{l} \{\xi_{(\text{wfu})(\gamma)}(c_i)\}$ (Z7).

Recall $\xi_{(\text{wfu})(\gamma)}(c_i) \notin \bigcup_{i=1}^{m_{l'}} \{b_i\}$ by Z5 and $[\xi_{(\text{wfu})(\gamma)}(c_i) \notin \bigcup_{i=1}^{m_{l'}} \{c'_i, \xi_{(\text{wfu})(\gamma)}(c'_i)\} \cup \bigcup_{i=1}^{l} \{\xi_{(\text{wfu})(\gamma)}(c_i)\}$ by Z7. Then, conclude $\xi_{(\text{wfu})(\gamma)}(c_i) \notin \bigcup_{i=1}^{m_{l'}} \{b_i\} \cup \bigcup_{i=1}^{m_{l'}} \{c'_i, \xi_{(\text{wfu})(\gamma)}(c'_i)\} \cup \bigcup_{i=1}^{l} \{\xi_{(\text{wfu})(\gamma)}(c_i)\}$ by...
ZFC. Then, because \( e \neq f \) by Z1, conclude:

\[
\begin{align*}
\mathcal{C}_{\xi(z,a),\xi(z,a) \rightarrow \tau\{z,a\} \in \text{dom}(\xi) \cap \text{dom}(\mathcal{C})} & \\
\bigcup_{i=1}^{m'} b_i(e_i) \sqcup \xi_{(wev)^{\circ}}(b_i)((wev)^{\circ}) & \bigcup_{i=1}^{m'} \xi_{(wev)^{\circ}}(b_i)((wev)^{\circ}) \sqcup \\
\bigcup_{i=1}^{l'} (c_i'(\xi')) \sqcup \xi_{(wfu)^{\circ}}(c_i)((wfu)^{\circ}) & \bigcup_{i=1}^{l'} \xi_{(wfu)^{\circ}}(c_i)((wfu)^{\circ}) \sqcup \\
\end{align*}
\]

\[\simeq \gamma' \sqcup \xi_{(wfu)^{\circ}}(c_i)((wfu)^{\circ})\]

by the definition of \( \mathcal{C} \) (Z8).

Conclude:

\[
\begin{align*}
\varepsilon \in \text{sol}(wev, \beta) & \\
\varepsilon \sqcup \text{sol}(wfu, \gamma) &
\end{align*}
\]

Case: \( l' \geq 1 \). Observations:

- Recall \( \text{Act}(\beta) \subseteq \text{dom}(\Xi) \). Then, conclude \( \xi_{(wfu)^{\circ}}(c_i') \notin \text{Act}(\beta) \) by Proposition 12. Then, because \( \text{Act}(\beta) = \bigcup_{i=1}^{m'} b_i \sqcup \bigcup_{i=1}^{l'} b_i' \}, \) conclude \( \xi_{(wfu)^{\circ}}(c_i') \notin \bigcup_{i=1}^{m'} b_i \sqcup \bigcup_{i=1}^{l'} b_i' \). Then, conclude \( \xi_{(wfu)^{\circ}}(c_i') \notin \bigcup_{i=1}^{n} \{b_i\} \) by ZFC (Z9).
Recall \( \ell' \geq 1 \) by the definition of this case. Then, conclude \( \xi_{(wfu)j}(e' \ell) \in \bigcup_{\ell=1}^{\ell'} \{ c' \in \xi_{(wfu)j}(e' \ell) \} \) by ZFC. Then, because \( \text{Act}(\overline{\text{iso}}(wfu, \gamma)) = \bigcup_{\ell=1}^{\ell'} \{ c' \in \xi_{(wfu)j}(e' \ell) \} \) by Z2, conclude \( \xi_{(wfu)j}(e' \ell) \in \text{Act}(\overline{\text{iso}}(wfu, \gamma)) \) (Z10).

Recall \( \overline{\xi_{(wfu)j}}(c' \ell) \in \text{Act}(\overline{\text{iso}}(wfu, \gamma)) \) by Z10 and \( \left[ \text{Act}(\gamma) \subseteq \text{dom}(\Xi) \right] \) by Z11. Then, conclude \( \overline{\xi_{(wfu)j}}(c' \ell) \notin \text{Act}(\overline{\text{iso}}(wfu, \gamma)) \) by Z2. Then, because \( \text{Act}(\overline{\text{iso}}(wfu, \gamma)) = \bigcup_{\ell=1}^{\ell'} \{ c' \in \xi_{(wfu)j}(e' \ell) \} \) by ZC, then, conclude \( \xi_{(wfu)j}(e' \ell) \notin \bigcup_{\ell=1}^{\ell'} \{ c' \in \xi_{(wfu)j}(e' \ell) \} \) by ZFC. Then, because \( e \neq f \) by Z1, conclude

\[
\begin{align*}
\overline{\text{iso}}(wvu, \beta) & \models \overline{\text{iso}}(wvu, \gamma) \\
\overline{\text{iso}}(wvu, \beta) & \models \overline{\text{iso}}(wvu, \gamma)
\end{align*}
\]

by the definition of \( \mathcal{C} \) (Z12).

Conclude:

\[
\begin{align*}
\text{Act}(wvu, \beta) & \models \text{Act}(wvu, \gamma) \\
\text{Act}(wvu, \beta) & \models \text{Act}(wvu, \gamma)
\end{align*}
\]
C. Proofs for Section 5.2

Proposition 16.
1. \([x \notin \text{Bound}(p) \text{ and } \text{Bound}(p) \cap w = \emptyset]\) implies \(\text{Bound}(p) \cap wx = \emptyset\)
2. \([d \notin \text{Bound}(p) \text{ and } \text{Bound}(p) \cap w = \emptyset]\) implies \(\text{Bound}(p) \cap wd = \emptyset\)

Proof.
1. Assumptions:
   - \([x \notin \text{Bound}(p) \text{ and } \text{Bound}(p) \cap w = \emptyset]\) (Z1).

   Observations:
   - Suppose \([y \in x \text{ for some } y \in \text{Bound}(p)]\). Then, conclude \([y = x \text{ for some } y \in \text{Bound}(p)]\) by the definition of \(\varepsilon\). Then, conclude \(y \in \text{Bound}(p)\) by ZFC—a contradiction, because \(x \notin \text{Bound}(p)\) by Z1. Hence, \([y \notin x \text{ for all } y \in \text{Bound}(p)]\) (Z2).
   - Recall \(\text{Bound}(p) \cap w = \emptyset\) by Z1. Then, conclude \(\{y \in \text{Bound}(p) \mid y \in w\} = \emptyset\) by the definition of \(\cap\). Then, conclude \([y \notin w \text{ for all } y \in \text{Bound}(p)]\) by ZFC (Z3).

2. Likewise.

Proposition 17 (\(\tilde{\text{isol}}\) and \(\setminus\) commute on processes).
\[
[p \in \text{Basic and } x \notin w_1, w_2 \text{ and } \text{Bound}(p) \cap w_1xw_2 = \emptyset] \implies \tilde{\text{isol}}(w_1xw_2, p)[d/x] = \tilde{\text{isol}}(w_1dw_2, p[d/x])
\]

Proof. Assumptions:
   - \([p \in \text{Basic and } x \notin w_1, w_2 \text{ and } \text{Bound}(p) \cap w_1xw_2 = \emptyset]\) (Z1).

Proceed by induction on the structure of \(p\).

Base: \([p = \alpha \text{ or } p = \delta]\). Proceed by case distinction on the structure of \(p\).

Case: \(p = \alpha\). Recall \(x \notin w_1, w_2\) by Z1. Then, conclude \(\tilde{\text{isol}}(w_1xw_2, \alpha)[d/x] = \tilde{\text{isol}}(w_1dw_2, \alpha[d/x])\) by Proposition 10. Then, because \(p = \alpha\) by the definition of this case, conclude \(\tilde{\text{isol}}(w_1xw_2, p)[d/x] = \tilde{\text{isol}}(w_1dw_2, p[d/x])\).

Case: \(p = \delta\). Conclude:
Step: \( p = q + r \) or \( p = q \cdot r \) or \( p = c \to q \circ r \) or \( p = \sum_{y \in \{d_1, \ldots, d_4\}} q \). Assumptions:

- Induction hypothesis (IH):
  \[
  \left[ \dot{p} \in \text{Basic and } \dot{x} \notin \dot{w}_1, \dot{w}_2 \right] \text{ and } \text{Bound}(\dot{p}) \cap \dot{w}_1 \dot{x} \dot{w}_2 = \emptyset \]
  implies \( \dot{\text{isol}}(\dot{w}_1 \dot{x} \dot{w}_2, \dot{p})[d/\dot{x}] = \dot{\text{isol}}(\dot{w}_1 \dot{d} \dot{w}_2, \dot{p}[d/\dot{x}]) \)
  for all \( \dot{p} \in \{q, r\} \)

Proceed by case distinction on the structure of \( p \).

Case: \( [p = q + r \text{ or } p = q \cdot r \text{ or } p = c \to q \circ r \] \). Observations:

- Recall \( p \in \text{Basic} \) by Z1. Then, because \( \left[ p = q + r \text{ or } p = q \cdot r \text{ or } p = c \to q \circ r \right] \) by the definition of this case, conclude \( \left[ q + r \in \text{Basic or } q \cdot r \in \text{Basic or } c \to q \circ r \in \text{Basic} \right] \). Then, conclude \( q, r \in \text{Basic} \) by the definition of Basic (Z2).

- Conclude \( 1, 2 \notin \text{Var} \) by Definition \([\text{2}]\). Then, because \( x \in \text{Var} \) by the definition of \( x \), conclude \( \left[ x \neq 1 \text{ and } x \neq 2 \right] \). Then, conclude \( x \notin 1, 2 \) by the definition of \( \notin \). Then, because \( x \notin w_2 \) by Z1, conclude \( x \notin w_2, 1, 2 \). Then, conclude \( x \notin w_2, w_2 \) by the definition of \( \notin \). Then, because \( x \notin w_1 \) by Z1, conclude \( x \notin w_1, w_2, w_2 \) (Z3).

- Recall \( \text{Bound}(p) \cap w_1 x w_2 = \emptyset \) by Z1. Then, because \( \left[ p = q + r \text{ or } p = q \cdot r \text{ or } p = c \to q \circ r \right] \) by the definition of this case, conclude
  \[
  \text{Bound}(q + r) \cap w_1 x w_2 = \emptyset \text{ or } \text{Bound}(q \cdot r) \cap w_1 x w_2 = \emptyset \\
  \text{or } \text{Bound}(c \to q \circ r) \cap w_1 x w_2 = \emptyset
  \]
  Then, conclude \( \text{Bound}(q) \cup \text{Bound}(r) \cap w_1 x w_2 = \emptyset \) by the definition of Bound. Then, conclude \( \left[ \text{Bound}(q) \cap w_1 x w_2 = \emptyset \text{ and } \text{Bound}(r) \cap w_1 x w_2 = \emptyset \right] \) by ZFC (Z4).

- Recall \( \left[ q, r \in \text{Basic} \text{ by Z2} \right] \) and \( \left[ x \notin w_1, w_2, w_2 \text{ by Z3} \right] \) and \( \left[ \text{Bound}(q) \cap w_1 x w_2 = \emptyset \text{ and } \text{Bound}(r) \cap w_1 x w_2 = \emptyset \right] \) by Z4. Then, conclude
  \[
  \dot{\text{isol}}(w_1 x w_2, q)[d/x] = \dot{\text{isol}}(w_1 d w_2, q[d/x]) \\
  \text{and } \dot{\text{isol}}(w_1 x w_2, r)[d/x] = \dot{\text{isol}}(w_1 d w_2, r[d/x])
  \]
  by IH (Z5).

Proceed by case distinction on the structure of \( p \).

Case: \( p = q + r \). Conclude:
By the definition of \( q \), conclude
\[
\sum_{y \in \{ d_1, \ldots, d_t \}} q
\]
Observations:

- Recall \( p \in \text{Basic} \) by Z1. Then, because \( p = \sum_{y \in \{ d_1, \ldots, d_t \}} q \) by the definition of this case, conclude \( \sum_{y \in \{ d_1, \ldots, d_t \}} q \in \text{Basic} \). Then, conclude \([ q \in \text{Basic} \text{ and } y \notin \text{Bound}(q) ]\) by the definition of Basic (Z6).

- Recall \( \text{Bound}(p) \cap w_1 x w_2 = \emptyset \) by Z1. Then, because \( p = \sum_{y \in \{ d_1, \ldots, d_t \}} q \) by the definition of this case, conclude \( \text{Bound}(\sum_{y \in \{ d_1, \ldots, d_t \}} q) \cap w_1 x w_2 = \emptyset \). Then, conclude \( \text{Bound}(q) \cup \{ y \} \cap w_1 x w_2 = \emptyset \) by the definition of Bound. Then, conclude \([ \text{Bound}(q) \cap w_1 x w_2 = \emptyset \text{ and } \{ y \} \cap w_1 x w_2 = \emptyset ]\) by ZFC (Z7).

- Recall \( \{ y \} \cap w_1 x w_2 = \emptyset \) by Z7. Then, conclude \( \{ z \in \{ y \} \mid z \in w_1 x w_2 \} = \emptyset \) by the definition of \( \cap \). Then, conclude \( y \notin w_1 x w_2 \) by ZFC. Then, conclude \([ y \notin w_1 \text{ and } y \notin x \text{ and } y \notin w_2 ]\) by the definition of \( \in \). Then, conclude \( y \notin x \) by the definition of \( \in \). Then, conclude
\[
(\sum_{y \in \{ d_1, \ldots, d_t \}} \text{isol}(w_1 x w_2 y, q))[d/x] = \sum_{y \in \{ d_1, \ldots, d_t \}} \text{isol}(w_1 x w_2 y, q)[d/x]\]
\[
\text{and } (\sum_{y \in \{ d_1, \ldots, d_t \}} q)[d/x] = \sum_{y \in \{ d_1, \ldots, d_t \}} q[d/x]\]
by the definition of \([/]\) (Z8).
• Recall \( \{y\} \cap w_1xw_2 = \emptyset \) by Z7. Then, conclude \( \{z \in \{y\} \mid z \in w_1xw_2\} = \emptyset \) by the definition of \( \cap \). Then, conclude \( y \notin w_1xw_2 \) by ZFC. Then, conclude \( \{y \notin w_1 \text{ and } y \notin x \text{ and } y \notin w_2\} \) by the definition of \( \notin \). Then, conclude \( y \neq x \) by the definition of \( \notin \). Then, conclude \( x \neq y \) by the definition of \( \notin \). Then, because \( x \notin w_2 \) by Z1, conclude \( x \notin w_2 \). Then, conclude \( x \notin w_2y \) by the definition of \( \notin \). Then, because \( x \notin w_1 \) by Z1, conclude \( x \notin w_1w_2y \) (Z9).

• Recall \( \{y \notin \text{Bound}(q)\} \) by Z6 and \( \text{Bound}(q) \cap w_1xw_2 = \emptyset \) by Z7. Then, conclude \( \text{Bound}(q) \cap w_1xw_2 = \emptyset \) by Proposition 16 (Z10).

• Recall \( \{q \in \text{Basic}\} \) by Z6 and \( \{x \notin w_1, w_2y\} \) by Z9 and \( \text{Bound}(q) \cap w_1xw_2y = \emptyset \) by Z10. Then, conclude \( \text{iso}(w_1xw_2y, q)\{d/x\} = \text{iso}(w_1dwx_2, q\{d/x\}) \) by IH (Z11).

Conclude:

\[
\begin{align*}
C_{\text{iso}} &= \text{iso}(w_1xw_2, p)\{d/x\} \\
\equiv & \text{iso}(w_1xw_2, \sum_{y \in \{d_1, \ldots, d_k\}} q)\{d/x\} \\
Z &= (\sum_{y \in \{d_1, \ldots, d_k\}} \text{iso}(w_1xw_2y, q))\{d/x\} \\
Z_{11} &= \sum_{y \in \{d_1, \ldots, d_k\}} \text{iso}(w_1xw_2y, q)\{d/x\} \\
Z_{10} &= \text{iso}(w_1dwx_2, \sum_{y \in \{d_1, \ldots, d_k\}} q)\{d/x\} \\
Z &= \text{iso}(w_1dwx_2, (\sum_{y \in \{d_1, \ldots, d_k\}} q)\{d/x\}) \\
C_{\text{iso}} &= \text{iso}(w_1dwx_2, p)\{d/x\}
\end{align*}
\]

\[\square\]

**Proposition 18 (Normal form for iso-processes).**

\[ [p \in \text{Basic and } p \in \text{TauFree and } \text{Act}(p) \subseteq \text{dom}(\Xi) \text{ and } \text{Bound}(p) \cap w = \emptyset] \]

implies

\[ \left[ \begin{array}{l}
\text{iso}(w, p) \simeq \sum_{i=1}^n \text{iso}(ww_1, \alpha_i) + \sum_{i=1}^{n'} \text{iso}(ww'_1, \alpha'_i) \cdot p'_i
\end{array} \right] \]

and

\[ \left[ \begin{array}{l}
\alpha_i \in \text{TauFree and } \text{Act}(\alpha_i) \subseteq \text{dom}(\Xi) \text{ for all } 1 \leq i \leq n
\end{array} \right] \]

and

\[ \left[ \begin{array}{l}
\alpha'_i \in \text{TauFree and } \text{Act}(\alpha'_i) \subseteq \text{dom}(\Xi) \text{ for all } 1 \leq i \leq n'
\end{array} \right] \]

for some \[ [n, n', w_1, \ldots, w_n, \alpha_1, \ldots, \alpha_n, w'_1, \ldots, w'_{n'}, \alpha'_1, \ldots, \alpha'_{n'}, p'_1, \ldots, p'_{n'}] \] (Z1).

**PROOF.** Assumptions:

• \[ [p \in \text{Basic and } p \in \text{TauFree and } \text{Act}(p) \subseteq \text{dom}(\Xi) \text{ and } \text{Bound}(p) \cap w = \emptyset] \] (Z1).

Proceed by induction on the structure of \( p \).

**Base:** \( p = \alpha \). Assumptions:

• \[ [n, n', w_1, \alpha_1 = 1, 0, \epsilon, \alpha] \] (Z2).

**Observations:**

• Conclude (Z3):
Step: \( p = q + r \) or \( p = q \cdot r \) or \( p = c \rightarrow q \circ r \) or \( p = \sum_{x \in \{d_1, \ldots, d_t\}} q \). Assumptions:

- Induction hypothesis (IH):

\[
\begin{bmatrix}
\hat{p} \in \text{Basic and } \hat{p} \in \text{TauFree and } \\
\text{Act}(\hat{p}) \subseteq \text{dom}(\Xi) \text{ and Bound}(\hat{p}) \cap \hat{w} = \emptyset
\end{bmatrix}
\]

implying

\[
\begin{bmatrix}
\text{Base} \\
\text{ZFC} \\
\text{Z2} \\
\text{Z3} \\
\text{Z4} \\
\text{Z5}
\end{bmatrix}
\]

\[
\begin{bmatrix}
\hat{\text{isol}}(w, p) \\
\hat{\text{isol}}(w, \alpha) \\
\sum_{i=1}^{n} \hat{\text{isol}}(w w_i, \alpha_1) \\
\sum_{i=1}^{n} \hat{\text{isol}}(w w_i, \alpha_i) + \delta \\
\sum_{i=1}^{n} \hat{\text{isol}}(w w_i, \alpha_i) + \sum_{i=1}^{n'} \hat{\text{isol}}(w w'_i, \alpha'_i) \cdot \hat{p}'_i \\
\sum_{i=1}^{n} \hat{\text{isol}}(w w_i, \alpha_i) + \sum_{i=1}^{n'} \hat{\text{isol}}(w w'_i, \alpha'_i) \cdot \hat{p}'_i
\end{bmatrix}
\]

- Recall \([p \in \text{TauFree and } \text{Act}(p) \subseteq \text{dom}(\Xi)]\) by Z1. Then, because \( p = \alpha \) by the definition of this case, conclude \([\alpha \in \text{TauFree and } \text{Act}(\alpha) \subseteq \text{dom}(\Xi)]\). Then, because \( \alpha_1 = \alpha \) by Z2, conclude \([\alpha_1 \in \text{TauFree and } \text{Act}(\alpha_1) \subseteq \text{dom}(\Xi)]\) for all \( 1 \leq i \leq 1 \) by ZFC. Then, because \( n = 1 \) by Z2, conclude \([\alpha_1 \in \text{TauFree and } \text{Act}(\alpha_1) \subseteq \text{dom}(\Xi)]\) for all \( 1 \leq i \leq n \) (Z4).

- Conclude \([\alpha'_1 \in \text{TauFree and } \text{Act}(\alpha'_1) \subseteq \text{dom}(\Xi)]\) for all \( 1 \leq i \leq 0 \) by ZFC. Then, because \( n' = 0 \) by Z2, conclude \([\alpha'_1 \in \text{TauFree and } \text{Act}(\alpha'_1) \subseteq \text{dom}(\Xi)]\) for all \( 1 \leq i \leq n' \) (Z5).

Conclude the consequent of this proposition by and-ing the results in Z3, Z4, and Z5.

Case: \([p = q + r \text{ or } p = q \cdot r \text{ or } p = c \rightarrow q \circ r \text{ or } p = \sum_{x \in \{d_1, \ldots, d_t\}} q]\). Observations:

- Recall \( p \in \text{Basic by Z1. Then, because } [p = q + r \text{ or } p = q \cdot r \text{ or } p = c \rightarrow q \circ r \text{ or } p = \sum_{x \in \{d_1, \ldots, d_t\}} q]\) by the definition of this case, conclude \([q + r \in \text{Basic or } q \cdot r \in \text{Basic or } c \rightarrow q \circ r \in \text{Basic}]\). Then, conclude \( q, r \in \text{Basic by the definition of Basic (Z6).} \)

- Recall \( p \in \text{TauFree by Z1. Then, because } [p = q + r \text{ or } p = q \cdot r \text{ or } p = c \rightarrow q \circ r \text{ by the definition of this case, conclude } [q + r \in \text{TauFree or } q \cdot r \in \text{TauFree or } c \rightarrow q \circ r \in \text{TauFree}]\). Then, conclude \( q, r \in \text{TauFree by the definition of TauFree (Z7).} \)

- Recall \( \text{Act}(p) \subseteq \text{dom}(\Xi) \) by Z1. Then, because \([p = q + r \text{ or } p = q \cdot r \text{ or } p = c \rightarrow q \circ r \text{ by the definition of this case, conclude } \text{Act}(q + r) \subseteq \text{dom}(\Xi) \text{ or } \text{Act}(q \cdot r) \subseteq \text{dom}(\Xi) \text{ or } \text{Act}(c \rightarrow q \circ r) \subseteq \text{dom}(\Xi) \)]\). Then, conclude \( \text{Act}(q) \cup \text{Act}(r) \subseteq \text{dom}(\Xi) \) by the definition of Act. Then, conclude \( \text{Act}(q) \), \( \text{Act}(r) \subseteq \text{dom}(\Xi) \) by ZFC (Z8).
• Recall $\text{Bound}(p) \cap w = \emptyset$ by Z1. Then, because $[p = q + r \lor p = q \cdot r \lor p = c \rightarrow q \circ r]$ by the definition of this case, conclude

$$\text{Bound}(q + r) \cap w = \emptyset \lor \text{Bound}(q \cdot r) \cap w = \emptyset$$

or $\text{Bound}(c \rightarrow q \circ r) \cap w = \emptyset$

Then, conclude $\text{Bound}(q) \cup \text{Bound}(r) \cap w = \emptyset$ by the definition of $\text{Bound}$. Then, conclude $[\text{Bound}(q) \cap w = \emptyset \land \text{Bound}(r) \cap w = \emptyset]$ by ZFC (Z9).

• Conclude $1, 2 \notin \mathcal{Var}$ by Definition 2. Then, because $\text{Bound}(q), \text{Bound}(r) \subseteq \mathcal{Var}$ by the definition of $\text{Bound}$, conclude $1, 2 \notin \text{Bound}(q), \text{Bound}(r)$ (Z10).

• Recall $[1, 2 \notin \text{Bound}(q), \text{Bound}(r) \text{by Z10}]$ and $[\text{Bound}(q) \cap w = \emptyset \land \text{Bound}(r) \cap w = \emptyset]$ by Z9. Then, conclude $[\text{Bound}(q) \cap w 1 = \emptyset \land \text{Bound}(r) \cap w 2 = \emptyset]$ by Proposition 16 (Z11).

• Recall $[q, r \in \text{Basic by Z6}]$ and $[q, r \in \text{TauFree by Z7}]$ and $[\text{Act}(q), \text{Act}(r) \subseteq \text{dom}(\Xi)]$ by Z8 and $[\text{Bound}(q) \cap w 1 = \emptyset \land \text{Bound}(r) \cap w 2 = \emptyset]$ by Z11. Then, conclude

$$\left[\begin{array}{l}
\text{isol}(w_1, q) \simeq \sum_{i=1}^{m} \text{isol}(w_1 v_i, \beta_i) + \sum_{i=1}^{m} \text{isol}(w_1 v_i^\prime, \beta_i^\prime \cdot q_i^\prime) \\
\beta_i \in \text{TauFree and Act}(\beta_i) \subseteq \text{dom}(\Xi) \text{ for all } 1 \leq i \leq m
\end{array}\right]$$

and

$$\left[\begin{array}{l}
\beta_i^\prime \in \text{TauFree and Act}(\beta_i^\prime) \subseteq \text{dom}(\Xi) \text{ for all } 1 \leq i \leq m^\prime
\end{array}\right]$$

and

$$\left[\begin{array}{l}
\text{isol}(w_2, r) \simeq \sum_{i=1}^{l} \text{isol}(w_2 u_i, \gamma_i) + \sum_{i=1}^{l} \text{isol}(w_2 u_i^\prime, \gamma_i^\prime \cdot r_i^\prime) \\
\gamma_i \in \text{TauFree and Act}(\gamma_i) \subseteq \text{dom}(\Xi) \text{ for all } 1 \leq i \leq l
\end{array}\right]$$

and

$$\left[\begin{array}{l}
\gamma_i^\prime \in \text{TauFree and Act}(\gamma_i^\prime) \subseteq \text{dom}(\Xi) \text{ for all } 1 \leq i \leq l^\prime
\end{array}\right]$$

by IH (Z12).

Proceed by case distinction on the structure of $p$.

Case: $p = q + r$. Assumptions:

• $[n, n^\prime = m + l, m^\prime + l^\prime]$ (Z13).

• $w_i, \alpha_i = \begin{cases} w_1, \beta_i & \text{if } 1 \leq i \leq m \\ w_{m+1}, \gamma_i & \text{if } m+1 \leq i \leq m + l \end{cases}$ (Z14).

• $w_i^\prime, \alpha_i^\prime, r_i^\prime = \begin{cases} w_1^\prime, \beta_i^\prime, q_i^\prime & \text{if } 1 \leq i \leq m^\prime \\ w_{m+1}^\prime, \gamma_i^\prime & \text{if } m^\prime + 1 \leq i \leq m^\prime + l^\prime \end{cases}$ (Z15).

Observations:

• Conclude (Z16):

$$\begin{array}{l}
\text{isol}(w, p) \\
\simeq \text{isol}(w, q + r) \\
\simeq \text{isol}(w_1, q) + \text{isol}(w_2, r) \\
\simeq \sum_{i=1}^{m} \text{isol}(w_1 v_i, \beta_i) + \sum_{i=1}^{m} \text{isol}(w_1 v_i^\prime, \beta_i^\prime \cdot q_i^\prime) + \\
\sum_{i=1}^{m} \text{isol}(w_{m+1}, \gamma_i) + \sum_{i=1}^{m} \text{isol}(w_{m+1}^\prime, \gamma_i^\prime \cdot r_i^\prime)
\end{array}$$

• Recall $[\beta_i \in \text{TauFree and Act}(\beta_i) \subseteq \text{dom}(\Xi) \text{ for all } 1 \leq i \leq m]$ and $[\gamma_i \in \text{TauFree and Act}(\gamma_i) \subseteq \text{dom}(\Xi) \text{ for all } 1 \leq i \leq l]$.
by Z12. Then, conclude

\[
[\beta_i \in \text{TauFree and } \text{Act}(\beta_i) \subseteq \text{dom}(\Xi)] \quad \text{for all } 1 \leq i \leq m
\]

and

\[
[\gamma_{i-m} \in \text{TauFree and } \text{Act}(\gamma_{i-m}) \subseteq \text{dom}(\Xi)]
\]

for all \( m+1 \leq i \leq m+l \)

by ZFC. Then, conclude \([\alpha_i \in \text{TauFree and } \text{Act}(\alpha_i) \subseteq \text{dom}(\Xi)] \quad \text{for all } 1 \leq i \leq m+l \)

by the definition of \( \alpha_i \) in Z14. Then, because \( n = m+l \) by Z13, conclude \([\alpha_i \in \text{TauFree and } \text{Act}(\alpha_i) \subseteq \text{dom}(\Xi)] \quad \text{for all } 1 \leq i \leq n \) (Z17).

• Recall

\[
[\beta_i' \in \text{TauFree and } \text{Act}(\beta_i') \subseteq \text{dom}(\Xi)] \quad \text{for all } 1 \leq i \leq m' \\
\text{and } [\gamma_i' \in \text{TauFree and } \text{Act}(\gamma_i') \subseteq \text{dom}(\Xi)] \quad \text{for all } 1 \leq i \leq l'
\]

by Z12. Then, conclude

\[
[\beta_i' \in \text{TauFree and } \text{Act}(\beta_i') \subseteq \text{dom}(\Xi)] \quad \text{for all } 1 \leq i \leq m'
\]

and

\[
[\gamma_{i-m'} \in \text{TauFree and } \text{Act}(\gamma_{i-m'}) \subseteq \text{dom}(\Xi)]
\]

for all \( m'+1 \leq i \leq m'+l' \)

by ZFC. Then, conclude \([\alpha_i' \in \text{TauFree and } \text{Act}(\alpha_i') \subseteq \text{dom}(\Xi)] \quad \text{for all } 1 \leq i \leq m'+l' \)

by the definition of \( \alpha_i' \) in Z15. Then, because \( n' = m'+l' \) by Z13, conclude \([\alpha_i' \in \text{TauFree and } \text{Act}(\alpha_i') \subseteq \text{dom}(\Xi)] \quad \text{for all } 1 \leq i \leq n' \) (Z18).

Conclude the consequent of this proposition by and-ing the results in Z16, Z17, and Z18.

**Case:** \( p = q \cdot r \). Assumptions:

• \([n, n' = 0, m, m']\) (Z19).

• \( w_i', \alpha_i', p_i' \) = \[
\begin{cases}
1v_i, \beta_i, \text{isol}(w2, r) & \text{if } 1 \leq i \leq m \\
1v_{i-m}, \beta_{i-m}, q_{i-m} \cdot \text{isol}(w2, r) & \text{if } m+1 \leq i \leq m+m'
\end{cases}
\]

(Z20).

Observations:

• Conclude (Z21):

\[
\text{isol}(w, p)
\]

\[
\text{isol}(w, q \cdot r)
\]

\[
\text{isol}(w2, r)
\]

\[
\text{isol}(w1, q) \cdot \text{isol}(w2, r)
\]

\[
\text{isol}(w1v_i, \beta_i) + \sum_{i=1}^{m'} (\text{isol}(w1v_i', \beta_i') \cdot q_i') \cdot \text{isol}(w2, r)
\]

\[
\sum_{i=1}^{m} (\text{isol}(w1v_i, \beta_i) \cdot \text{isol}(w2, r)) + \sum_{i=1}^{m'} (\text{isol}(w1v_i', \beta_i') \cdot q_i') \cdot \text{isol}(w2, r)
\]

\[
\sum_{i=1}^{m} (\text{isol}(wv_i', \alpha_i') \cdot p_i') + \sum_{i=m+1}^{m+m'} (\text{isol}(wv_i', \alpha_i') \cdot p_i')
\]

\[
\delta + \sum_{i=1}^{m} (\text{isol}(wv_i', \alpha_i') \cdot p_i')
\]

\[
\sum_{i=1}^{n} (\text{isol}(ww_i, \alpha_i) + \sum_{i=m+1}^{m+m'} (\text{isol}(ww_i, \alpha_i') \cdot p_i')
\]

\[
\sum_{i=1}^{n} (\text{isol}(ww_i, \alpha_i) + \sum_{i=1}^{m+m'} (\text{isol}(ww_i, \alpha_i') \cdot p_i')
\]

• Conclude \([\alpha_i \in \text{TauFree and } \text{Act}(\alpha_i) \subseteq \text{dom}(\Xi)] \quad \text{for all } 1 \leq i \leq 0 \) by ZFC. Then, because \( n = 0 \) by Z19, conclude \([\alpha_i \in \text{TauFree and } \text{Act}(\alpha_i) \subseteq \text{dom}(\Xi)] \quad \text{for all } 1 \leq i \leq n \) (Z22).
• Recall \( [\beta_i \in \text{TauFree and Act}\(\beta_i\) \subseteq \text{dom}\(\Xi\) ] \) for all \(1 \leq i \leq m\) and \( [\beta'_i \in \text{TauFree and Act}\(\beta'_i\) \subseteq \text{dom}\(\Xi\) ] \) for all \(1 \leq i \leq m'\).

by Z12. Then, conclude

\[
\begin{align*}
&[\beta_i \in \text{TauFree and Act}\(\beta_i\) \subseteq \text{dom}\(\Xi\) ] \quad \text{for all } 1 \leq i \leq m \\
\quad \text{and} \quad [\beta'_{i-m} \in \text{TauFree and Act}\(\beta'_{i-m}\) \subseteq \text{dom}\(\Xi\) ] \quad \text{for all } m + 1 \leq i \leq m + m'
\end{align*}
\]

by ZFC. Then, conclude \( [\alpha'_i \in \text{TauFree and Act}\(\alpha'_i\) \subseteq \text{dom}\(\Xi\) ] \) for all \(1 \leq i \leq m + m'\) by the definition of \(\alpha'_i\) in Z20. Then, because \(n' = m + m'\) by Z13, conclude \( [\alpha'_i \in \text{TauFree and Act}\(\alpha'_i\) \subseteq \text{dom}\(\Xi\) ] \) for all \(1 \leq i \leq n'\) (Z23).

Conclude the consequent of this proposition by and-ing the results in Z21, Z22, and Z23.

Case: \( p = c \rightarrow q \circ r \). Proceed by case distinction on the value of \( c \).

Case: \( c \equiv true \). Assumptions:

• \([n, n', w_i, \alpha_i, w'_i, \alpha'_i, p'_i = m, m', 1v_i, \beta_i, 1v'_i, \beta'_i, q'_i]\) (Z24).

Observations:

• Conclude (Z25):

\[
\begin{align*}
&\text{isol}(w, p) \\
&\quad = \text{isol}(w, c \rightarrow q \circ r) \\
&\quad = \text{isol}(w, \text{true} \rightarrow q \circ r) \\
&\quad = true \rightarrow \text{isol}(w1, q) \circ \text{isol}(w2, r) \\
&\quad \equiv \text{isol}(w1, q) \\
&\quad \equiv \sum_{i=1}^{n} \text{isol}(w1v_i, \beta_i) + \sum_{i=1}^{m} \sum_{i=1}^{m'} \text{isol}(w1v'_i, \beta'_i) \cdot q'_i \\
&\quad \equiv \sum_{i=1}^{n} \text{isol}(ww_i, \alpha_i) + \sum_{i=1}^{m} \text{isol}(ww'_i, \alpha'_i) \cdot p'_i
\end{align*}
\]

• Recall \( [\beta_i \in \text{TauFree and Act}\(\beta_i\) \subseteq \text{dom}\(\Xi\) ] \) for all \(1 \leq i \leq m\) by Z12. Then, conclude \( [\alpha_i \in \text{TauFree and Act}\(\alpha_i\) \subseteq \text{dom}\(\Xi\) ] \) for all \(1 \leq i \leq m\) by the definition of \(\alpha_i\) in Z24. Then, because \(n = m\) by Z24, conclude \( [\alpha_i \in \text{TauFree and Act}\(\alpha_i\) \subseteq \text{dom}\(\Xi\) ] \) for all \(1 \leq i \leq n\) (Z26).

• Recall \( [\beta'_i \in \text{TauFree and Act}\(\beta'_i\) \subseteq \text{dom}\(\Xi\) ] \) for all \(1 \leq i \leq m'\) by Z12. Then, conclude \( [\alpha'_i \in \text{TauFree and Act}\(\alpha'_i\) \subseteq \text{dom}\(\Xi\) ] \) for all \(1 \leq i \leq m'\) by the definition of \(\alpha'_i\) in Z24. Then, because \(n' = m'\) by Z24, conclude \( [\alpha'_i \in \text{TauFree and Act}\(\alpha'_i\) \subseteq \text{dom}\(\Xi\) ] \) for all \(1 \leq i \leq n'\) (Z27).

Conclude the consequent of this proposition by and-ing the results in Z25, Z26, and Z27.

Case: \( c \equiv false \). Assumptions:

• \([n, n', w_i, \alpha_i, w'_i, \alpha'_i, p'_i = l, l', 2u_i, \gamma_i, 2u'_i, \gamma'_i, r'_i]\) (Z28).

Observations:

• Conclude (Z29):

\[
\begin{align*}
&\text{isol}(w, p) \\
&\quad = \text{isol}(w, c \rightarrow q \circ r) \\
&\quad = \text{isol}(w, \text{false} \rightarrow q \circ r) \\
&\quad = false \rightarrow \text{isol}(w1, q) \circ \text{isol}(w2, r) \\
&\quad \equiv \text{isol}(w2, r) \\
&\quad \equiv \sum_{i=1}^{l} \text{isol}(w2u_i, \gamma_i) + \sum_{i=1}^{m} \sum_{i=1}^{m'} \text{isol}(w2u'_i, \gamma'_i) \cdot r'_i \\
&\quad \equiv \sum_{i=1}^{n} \text{isol}(ww_i, \alpha_i) + \sum_{i=1}^{m} \text{isol}(ww'_i, \alpha'_i) \cdot p'_i
\end{align*}
\]
• Recall \( [\gamma_i \in \text{Ta} \text{uFree} \text{ and } \text{Act}(\gamma_i) \subseteq \text{dom}(\Xi)] \) for all \( 1 \leq i \leq \ell \) by Z12. Then, conclude \( [\alpha_i \in \text{Tu} \text{uFree} \text{ and } \text{Act}(\alpha_i) \subseteq \text{dom}(\Xi)] \) for all \( 1 \leq i \leq \ell \) by the definition of \( \alpha_i \) in Z28. Then, because \( n = l \) by Z28, conclude \( [\alpha_i \in \text{Tu} \text{uFree} \text{ and } \text{Act}(\alpha_i) \subseteq \text{dom}(\Xi)] \) for all \( 1 \leq i \leq n \) (Z30).

• Recall \( [\gamma_i' \in \text{Tu} \text{uFree} \text{ and } \text{Act}(\gamma_i') \subseteq \text{dom}(\Xi)] \) for all \( 1 \leq i \leq \ell' \) by Z12. Then, conclude \( [\alpha_i' \in \text{Tu} \text{uFree} \text{ and } \text{Act}(\alpha_i') \subseteq \text{dom}(\Xi)] \) for all \( 1 \leq i \leq \ell' \) by the definition of \( \alpha_i' \) in Z28. Then, because \( n' = l' \) by Z28, conclude \( [\alpha_i' \in \text{Tu} \text{uFree} \text{ and } \text{Act}(\alpha_i') \subseteq \text{dom}(\Xi)] \) for all \( 1 \leq i \leq n' \) (Z31).

Conclude the consequent of this proposition by and-ing the results in Z29, Z30, and Z31.

\[ \sum_{x \in \{d_1, \ldots, d_\ell\}} q. \] Observations:

- Recall \( p \in \text{Basic} \) by Z1. Then, because \( p = \sum_{x \in \{d_1, \ldots, d_\ell\}} q \) by the definition of this case, conclude \( \sum_{x \in \{d_1, \ldots, d_\ell\}} q \in \text{Basic} \). Then, conclude \( \{q[d_i/x] \in \text{Basic for all } 1 \leq i \leq \ell\} \) by the definition of Basic (Z32).

- Recall \( p \in \text{Tu} \text{uFree} \) by Z1. Then, because \( p = \sum_{x \in \{d_1, \ldots, d_\ell\}} q \) by the definition of this case, conclude \( \sum_{x \in \{d_1, \ldots, d_\ell\}} q \in \text{Tu} \text{uFree} \). Then, conclude \( \{q[d_i/x] \in \text{Tu} \text{uFree for all } 1 \leq i \leq \ell\} \) by the definition of TuFree (Z33).

- Recall \( \text{Act}(p) \subseteq \text{dom}(\Xi) \) by Z1. Then, because \( p = \sum_{x \in \{d_1, \ldots, d_\ell\}} q \) by the definition of this case, conclude \( \text{Act}(\sum_{x \in \{d_1, \ldots, d_\ell\}} q) \subseteq \text{dom}(\Xi) \). Then, conclude \( \{\text{Act}(q[d_i/x]) \subseteq \text{dom}(\Xi) \text{ for all } 1 \leq i \leq \ell\} \) by the definition of Act (Z34).

- Recall \( \text{Bound}(p) \cap w = \emptyset \) by Z1. Then, because \( p = \sum_{x \in \{d_1, \ldots, d_\ell\}} q \) by the definition of this case, conclude \( \text{Bound}(\sum_{x \in \{d_1, \ldots, d_\ell\}} q) \cap w = \emptyset \). Then, conclude \( \text{Bound}(q) \cup \{x \} \cap w = \emptyset \) by the definition of Bound. Then, conclude \( \{\text{Bound}(q) \cap w = \emptyset \text{ and } \{x \} \cap w = \emptyset\} \) by ZFC (Z35).

- Recall \( \{x \} \cap w = \emptyset \) by Z35. Then, conclude \( x \notin w \) by the definition of \( \cap \) (Z36).

- Recall \( \sum_{x \in \{d_1, \ldots, d_\ell\}} q \in \text{Basic} \) by Z1. Then, conclude \( x \notin \text{Bound}(q) \) by the definition of Basic (Z37).

- Recall \( x \notin \text{Bound}(q) \) by Z37 and \( \{\text{Bound}(q) \cap w = \emptyset \} \) by Z35. Then, conclude \( \text{Bound}(q) \cap wx = \emptyset \) by Proposition (Z38).

- Recall \( q \in \text{Basic} \) by Z32 and \( x \notin w \) by Z36 and \( \{\text{Bound}(q) \cap wx = \emptyset \} \) by Z38. Then, conclude \( \{\text{isol}(wx, q)[d_i/x] = \text{isol}(wd_i, q[d_i/x]) \text{ for all } 1 \leq i \leq \ell\} \) by Proposition (Z39).

- Conclude \( d_i \in \text{Elem for all } 1 \leq i \leq \ell \) by the definition of \( \sum \). Then, because \( \{\text{Bound}(q) \subseteq \text{Var}\text{ by the definition of Bound}\} \) and \( \{\text{Elem}\cap\text{Var} = \emptyset \} \) by Definition, conclude \( d_i \notin \text{Bound}(q) \) for all \( 1 \leq i \leq \ell \) (Z40).

- Recall \( [\{d_i \notin \text{Bound}(q) \text{ for all } 1 \leq i \leq \ell\} \) by Z40 and \( \{\text{Bound}(q) \cap w = \emptyset \} \) by Z35. Then, conclude \( \{\text{Bound}(q) \cap wx = \emptyset \} \) by Proposition (Z41). Then, conclude \( \{\text{Bound}(q[d_i/x]) \cap wd_i = \emptyset \text{ for all } 1 \leq i \leq \ell\} \) by the definition of Bound (Z41).

- Recall \( \{d_i \notin \text{Bound}(q) \text{ for all } 1 \leq i \leq \ell\} \text{ by Z32 and } [\{q[d_i/x] \in \text{Tu} \text{uFree for all } 1 \leq i \leq \ell\} \text{ by Z33 and } [\text{Act}(q[d_i/x]) \subseteq \text{dom}(\Xi) \text{ for all } 1 \leq i \leq \ell\} \text{ by Z44 and } [\{\text{Bound}(q[d_i/x]) \cap wx = \emptyset \text{ for all } 1 \leq i \leq \ell\} \text{ by Z41}. \) Then, conclude \( \{\text{isol}(wd_i, q[d_i/x]) = \sum_{j=1}^m \text{isol}(wd_i, q[d_i/x]) \} \) and \( \{\text{Bound}(q[d_i/x]) \cap wx = \emptyset \text{ for all } 1 \leq i \leq \ell\} \) for all \( 1 \leq i \leq \ell \).
by IH (Z42).

Assumptions:

- \([n, n' = \ell m, \ell m']\) (Z43).
- \(w_i, \alpha_i = d_{\lfloor i \mod m \rfloor}, \beta_{\lfloor i \mod m \rfloor} \cdot (w_{i mod m} + 1)\) (Z44).
- \(w'_i, \alpha'_i = d_{\lfloor i \mod m' \rfloor}, \beta'_{\lfloor i \mod m' \rfloor} \cdot (w'_{i mod m'})\) (Z45).

Observations:

- Conclude (Z46):
  \[
  \sum_{i=1}^{\ell} \sum_{j=1}^{m} (\sum_{j=1}^{m'} (\sum_{j=1}^{m'} \sum_{j=1}^{m'} \sum_{j=1}^{m'} (w_{i, j}, \beta_{i,j})), \sum_{j=1}^{m'} (\sum_{j=1}^{m'} \sum_{j=1}^{m'} \sum_{j=1}^{m'} (w'_{i, j}, \beta'_{i,j})), 1 \leq j \leq m) \\
  \text{for all } 1 \leq i \leq \ell
  \]

  by Z12. Then, conclude
  \[
  \beta_{\lfloor i \mod m \rfloor}, \beta_{\lfloor i \mod m + 1 \rfloor} \in \text{TauFree and } \text{Act}(\beta_{\lfloor i \mod m \rfloor} \cdot (i \mod m + 1)) \subseteq \text{dom}(\Xi) \\
  \text{for all } 1 \leq i \leq \ell m
  \]

  by ZFC. Then, conclude \([(\alpha_i \in \text{TauFree and } \text{Act}(\alpha_i) \subseteq \text{dom}(\Xi)) \text{ for all } 1 \leq i \leq \ell m] \text{ by the definition of } \alpha_i \text{ in Z44. Then, because } n = \ell m \text{ by Z43, conclude } [(\alpha_i \in \text{TauFree and } \text{Act}(\alpha_i) \subseteq \text{dom}(\Xi)) \text{ for all } 1 \leq i \leq n] \text{ (Z47).}

- Conclude (Z47):
  \[
  \sum_{i=1}^{\ell} \sum_{j=1}^{m} (\sum_{j=1}^{m'} (\sum_{j=1}^{m'} \sum_{j=1}^{m'} (w_{i, j}, \beta_{i,j})), \sum_{j=1}^{m'} (\sum_{j=1}^{m'} \sum_{j=1}^{m'} \sum_{j=1}^{m'} (w'_{i, j}, \beta'_{i,j})), 1 \leq j \leq m) \\
  \text{for all } 1 \leq i \leq \ell m
  \]

  by Z12. Then, conclude
  \[
  \beta'_{\lfloor i \mod m' \rfloor}, \beta'_{\lfloor i \mod m' + 1 \rfloor} \in \text{TauFree and } \text{Act}(\beta'_{\lfloor i \mod m' \rfloor} \cdot (i \mod m' + 1)) \subseteq \text{dom}(\Xi) \\
  \text{for all } 1 \leq i \leq \ell m'
  \]

  by ZFC. Then, conclude \([(\alpha'_i \in \text{TauFree and } \text{Act}(\alpha'_i) \subseteq \text{dom}(\Xi)) \text{ for all } 1 \leq i \leq \ell m'] \text{ by the definition of } \alpha'_i \text{ in Z45. Then, because } n' = \ell m' \text{ by Z43, conclude } [(\alpha'_i \in \text{TauFree and } \text{Act}(\alpha'_i) \subseteq \text{dom}(\Xi)) \text{ for all } 1 \leq i \leq n'] \text{ (Z48).}

Conclude the consequent of this proposition by and-ing the results in Z46, Z47, and Z48.
Proof (of Proposition 5). Assumptions:

- \[ q, r \in \text{Basic and } q, r \in \text{TauFree and } \text{Act}(q) \subseteq \text{dom}(\Xi) \text{ and } \text{Bound}(q) \cap v = \emptyset \text{ and } \text{Bound}(r) \cap u = \emptyset \text{ and } v = u \]

Observations:

- Recall \[ p \in \text{Basic and } p \in \text{TauFree and } \text{Act}(p) \subseteq \text{dom}(\Xi) \text{ and } \text{Bound}(p) \cap w = \emptyset \] by Z1. Then, conclude

\[
\sum_{i=1}^{n} (\text{isol}(w_i, \alpha_i) + \sum_{i=1}^{n'} (\text{isol}(w_i, \alpha'_i) \cdot p_i))
\]

by Proposition 18 (Z2).

- Recall

\[
\text{Bound}(\Xi) \text{ and } \text{Act}(\Xi) \subseteq \text{dom}(\Xi) \text{ for all } 1 \leq i \leq n
\]

by Z2. Then, conclude

\[
\sum_{i=1}^{n} \delta + \sum_{i=1}^{n'} \delta
\]

by Proposition 2 (Z3).

Conclude:

\[
\sum_{i=1}^{n} \delta + \sum_{i=1}^{n'} \delta
\]

Proof (of Proposition 6). Assumptions:

- \[ q, r \in \text{Basic and } q, r \in \text{TauFree and } \text{Act}(q), \text{Act}(r) \subseteq \text{dom}(\Xi) \text{ and } \text{Bound}(q) \cap v = \emptyset \text{ and } \text{Bound}(r) \cap u = \emptyset \text{ and } v \neq u \]

Observations:

- Recall

\[
q, r \in \text{Basic and } q, r \in \text{TauFree and } \text{Act}(q), \text{Act}(r) \subseteq \text{dom}(\Xi) \text{ and } \text{Bound}(q) \cap v = \emptyset \text{ and } \text{Bound}(r) \cap u = \emptyset \text{ and } v \neq u
\]

by Z1. Then, conclude

\[
\sum_{i=1}^{m} \text{isol}(v_i, \beta_i) + \sum_{i=1}^{m'} (\text{isol}(v_i, \beta'_i) \cdot q_i)
\]

by Proposition 18 (Z2).
by Proposition 18 (Z2).

- Recall $v^x \neq v^y$ by Z1. Then, conclude

\[
\left\{ v^x v^y \neq v^x w^y \text{ and } v^x v^y \neq u^x w^y \text{ and } \forall u^x \neq v^x \right\}
\]

for all $[1 \leq i \leq m \text{ and } 1 \leq j \leq l \text{ and } 1 \leq i' \leq m' \text{ and } 1 \leq j' \leq l']$

by ZFC. Then, conclude

\[
\left\{ (v v_i)^x \neq (w w_j)^x \text{ and } (v v_i)^x \neq u^x w^x_j \text{ and } \forall u^x \neq v^x \right\}
\]

for all $[1 \leq i \leq m \text{ and } 1 \leq j \leq l \text{ and } 1 \leq i' \leq m' \text{ and } 1 \leq j' \leq l']$

by the definition of $\sharp$ (Z3).

- Recall [\[ \beta_i \in \text{TauFree and } \text{Act}(\beta_i) \subseteq \text{dom}(\Xi) \] for all $1 \leq i \leq m$]

and [\[ \beta_i' \in \text{TauFree and } \text{Act}(\beta_i') \subseteq \text{dom}(\Xi) \] for all $1 \leq i \leq m'$]

and [\[ \gamma_i \in \text{TauFree and } \text{Act}(\gamma_i) \subseteq \text{dom}(\Xi) \] for all $1 \leq i \leq l$]

and [\[ \gamma_i' \in \text{TauFree and } \text{Act}(\gamma_i') \subseteq \text{dom}(\Xi) \] for all $1 \leq i \leq l'$]

by Z2] and [\[ \left\{ (v v_i)^x \neq (w w_j)^x \text{ and } (v v_i)^x \neq u^x w^x_j \text{ and } \forall u^x \neq v^x \right\}
\]

for all $[1 \leq i \leq m \text{ and } 1 \leq j \leq l \text{ and } 1 \leq i' \leq m' \text{ and } 1 \leq j' \leq l']$

by Z3. Then, conclude

\[
\left\{ \text{iso}(v v_i, \beta_i) \mid \text{iso}(w w_j, \gamma_i) \right\} \simeq \delta
\]

and \[
\left\{ \text{iso}(v v_i, \beta_i) \mid \text{iso}(u u_j, \gamma_i) \right\} \simeq \delta
\]

and \[
\left\{ \text{iso}(v v_i', \beta_i') \mid \text{iso}(w w_j', \gamma_i') \right\} \simeq \delta
\]

and \[
\left\{ \text{iso}(v v_i', \beta_i') \mid \text{iso}(u u_j, \gamma_i) \right\} \simeq \delta
\]

by Proposition 3 (Z4).

- Conclude (Z5):

\[
\text{iso}(v, q) \mid \text{iso}(u, r)
\]

\[\cong \]

\[
\left( \sum_{i=1}^m \text{iso}(v v_i, \beta_i) + \sum_{i=1}^{m'} \text{iso}(v v_i', \beta_i') \cdot q_i' \right) \mid \sum_{i=1}^{m'} \text{iso}(u u_j, \gamma_i) + \sum_{i=1}^{m' \prime} \text{iso}(u u_j', \gamma_i') \cdot r_i'
\]

\[\cong \]

\[
\sum_{i=1}^m \sum_{j=1}^{l'} \text{iso}(v v_i, \beta_i) \mid \text{iso}(u u_j, \gamma_j) + \sum_{i=1}^m \sum_{j=1}^{l'} \text{iso}(v v_i', \beta_i) \cdot q_i' \mid \text{iso}(u u_j, \gamma_j) + \sum_{i=1}^m \sum_{j=1}^{l'} \text{iso}(v v_i', \beta_i') \cdot q_i' \mid \text{iso}(u u_j, \gamma_j)
\]

\[\cong \]

\[
\sum_{i=1}^m \sum_{j=1}^{l'} \text{iso}(v v_i, \beta_i) \mid \text{iso}(u u_j, \gamma_j) + \sum_{i=1}^m \sum_{j=1}^{l'} \text{iso}(v v_i', \beta_i) \cdot q_i' \mid \text{iso}(u u_j, \gamma_j) + \sum_{i=1}^m \sum_{j=1}^{l'} \text{iso}(v v_i', \beta_i') \cdot q_i' \mid \text{iso}(u u_j, \gamma_j) + \sum_{i=1}^m \sum_{j=1}^{l'} \text{iso}(v v_i', \beta_i') \cdot (q_i' \parallel r_i')
\]

66
\[
\sum_{i=1}^{m} \sum_{j=1}^{l} ?(isol(vu_i, \beta_i) | \overline{isol}(uu_j, \gamma_j)) + \\
\sum_{i=1}^{m} \sum_{j=1}^{l} ?(isol(vu_i, \beta_i) | \overline{isol}(uu'_j, \gamma'_j) \cdot r'_j) + \\
\sum_{i=1}^{m} \sum_{j=1}^{l} ?(isol(vu_i', \beta'_i) | \overline{isol}(uu_j, \gamma_j) \cdot q'_i) + \\
\sum_{i=1}^{m} \sum_{j=1}^{l} ?(isol(vu_i', \beta'_i) | \overline{isol}(uu'_j, \gamma'_j) \cdot (q'_i \parallel r'_j))
\]

\[
\sum_{i=1}^{m} \sum_{j=1}^{l} ?(isol(vu_i, \beta_i) | \overline{isol}(uu_j, \gamma_j)) + \\
\sum_{i=1}^{m} \sum_{j=1}^{l} ?(isol(vu_i, \beta_i) | \overline{isol}(uu'_j, \gamma'_j) \cdot ?(r'_j)) + \\
\sum_{i=1}^{m} \sum_{j=1}^{l} ?(isol(vu'_i, \beta'_i) | \overline{isol}(uu_j, \gamma_j) \cdot ?(q'_i)) + \\
\sum_{i=1}^{m} \sum_{j=1}^{l} ?(isol(vu'_i, \beta'_i) | \overline{isol}(uu'_j, \gamma'_j) \cdot ?(q'_i \parallel r'_j))
\]

\[
\sum_{i=1}^{m} \sum_{j=1}^{l} \delta + \\
\sum_{i=1}^{m} \sum_{j=1}^{l} \delta \cdot ?(r'_j)) + \sum_{i=1}^{m} \sum_{j=1}^{l} \delta \cdot ?(q'_i)) + \sum_{i=1}^{m} \sum_{j=1}^{l} \delta \cdot ?(q'_i \parallel r'_j))
\]

\[
\delta \sum_{i=1}^{m} \sum_{j=1}^{l} \delta + \sum_{i=1}^{m} \sum_{j=1}^{l} \delta + \sum_{i=1}^{m} \sum_{j=1}^{l} \delta + \sum_{i=1}^{m} \sum_{j=1}^{l} \delta
\]

- Conclude (Z6):

\[
?((isol(v, q) \cdot q') | (\overline{isol}(u, r) \cdot r'))
\]

\[
?((\sum_{i=1}^{m} \sum_{j=1}^{l} \sum_{l=1}^{m} (isol(vu_i, \beta_i) \cdot q'_i)) | (\sum_{i=1}^{m} \overline{isol}(uu_i, \gamma_i) \cdot \sum_{l=1}^{m} (isol(vu'_i, \beta'_i) \cdot q'_i) \cdot r'_j)
\]

\[
?((\sum_{i=1}^{m} \sum_{j=1}^{l} \sum_{l=1}^{m} (isol(vu_i, \beta_i) \cdot q'_i)) | (\sum_{i=1}^{m} \overline{isol}(uu_i, \gamma_i) \cdot r'_j) + \sum_{i=1}^{m} \sum_{j=1}^{l} \sum_{l=1}^{m} \overline{isol}(uu'_i, \gamma'_i) \cdot (q'_i \parallel r'_j)
\]

\[
?((\sum_{i=1}^{m} \sum_{j=1}^{l} \sum_{l=1}^{m} (isol(vu_i, \beta_i) \cdot q'_i)) | (\sum_{i=1}^{m} \overline{isol}(uu_i, \gamma_i) \cdot r'_j) + \sum_{i=1}^{m} \sum_{j=1}^{l} \sum_{l=1}^{m} \overline{isol}(uu'_i, \gamma'_i) \cdot (q'_i \parallel r'_j)
\]

\[
?((\sum_{i=1}^{m} \sum_{j=1}^{l} \sum_{l=1}^{m} (isol(vu_i, \beta_i) \cdot q'_i)) | (\sum_{i=1}^{m} \overline{isol}(uu_i, \gamma_i) \cdot (q'_i \parallel r'_j)) + \\
\sum_{i=1}^{m} \sum_{j=1}^{l} \sum_{l=1}^{m} (isol(vu_i, \beta_i) \cdot \overline{isol}(uu_j, \gamma_j) \cdot (q'_i \parallel r'_j) \cdot r'_j) + \\
\sum_{i=1}^{m} \sum_{j=1}^{l} \sum_{l=1}^{m} (isol(vu_i, \beta_i) \cdot \overline{isol}(uu'_j, \gamma'_j) \cdot (q'_i \parallel r'_j) \cdot r'_j))
\]

\[
?((\sum_{i=1}^{m} \sum_{j=1}^{l} \sum_{l=1}^{m} (isol(vu_i, \beta_i) \cdot q'_i)) | (\sum_{i=1}^{m} \overline{isol}(uu_j, \gamma_j) \cdot (q'_i \parallel r'_j) \cdot r'_j) + \\
\sum_{i=1}^{m} \sum_{j=1}^{l} \sum_{l=1}^{m} (isol(vu_i, \beta_i) \cdot \overline{isol}(uu_j, \gamma_j) \cdot (q'_i \parallel r'_j) \cdot r'_j) + \\
\sum_{i=1}^{m} \sum_{j=1}^{l} \sum_{l=1}^{m} (isol(vu_i, \beta_i) \cdot \overline{isol}(uu'_j, \gamma'_j) \cdot (q'_i \parallel r'_j) \cdot r'_j))
\]

\[
?((\sum_{i=1}^{m} \sum_{j=1}^{l} \sum_{l=1}^{m} (isol(vu_i, \beta_i) \cdot q'_i)) | (\sum_{i=1}^{m} \overline{isol}(uu_j, \gamma_j) \cdot (q'_i \parallel r'_j) \cdot r'_j) + \\
\sum_{i=1}^{m} \sum_{j=1}^{l} \sum_{l=1}^{m} (isol(vu_i, \beta_i) \cdot \overline{isol}(uu_j, \gamma_j) \cdot (q'_i \parallel r'_j) \cdot r'_j) + \\
\sum_{i=1}^{m} \sum_{j=1}^{l} \sum_{l=1}^{m} (isol(vu_i, \beta_i) \cdot \overline{isol}(uu'_j, \gamma'_j) \cdot (q'_i \parallel r'_j) \cdot r'_j))
\]

\[
?((\sum_{i=1}^{m} \sum_{j=1}^{l} \sum_{l=1}^{m} (isol(vu_i, \beta_i) \cdot q'_i)) | (\sum_{i=1}^{m} \overline{isol}(uu_j, \gamma_j) \cdot (q'_i \parallel r'_j) \cdot r'_j) + \\
\sum_{i=1}^{m} \sum_{j=1}^{l} \sum_{l=1}^{m} (isol(vu_i, \beta_i) \cdot \overline{isol}(uu_j, \gamma_j) \cdot (q'_i \parallel r'_j) \cdot r'_j) + \\
\sum_{i=1}^{m} \sum_{j=1}^{l} \sum_{l=1}^{m} (isol(vu_i, \beta_i) \cdot \overline{isol}(uu'_j, \gamma'_j) \cdot (q'_i \parallel r'_j) \cdot r'_j))
\]

\[
67
\]
Recall Conclude (Z4): by Proposition 18 (Z2).

\[ \sum_{i=1}^{m} \sum_{j=1}^{l}(\text{iso}(\nu v_{i}, \beta_{i}) | \text{iso}(\nu u_{j}, \gamma_{j})) \cdot \text{iso}(\nu u_{j}, \gamma_{j}) \cdot ?(q' \parallel r')) + \sum_{i=1}^{m} \sum_{j=1}^{l}(\text{iso}(\nu v_{i}, \beta_{i}) | \text{iso}(\nu u_{j}, \gamma_{j})) \cdot ?(q' \parallel r''), \]

Conclude the consequent of this proposition by and-ing the results in Z5 and Z6.

**Proof (of Proposition 7):** Assumptions:

- \[ q, r \in \text{Basic and } q, r \in \text{TauFree and } \text{Act}(q), \text{Act}(r) \subseteq \text{dom}(\Xi) \]
  and \( \text{Bound}(q) \cap \text{we} = \emptyset \) and \( \text{Bound}(r) \cap \text{wf} = \emptyset \) and \( e \neq f \) \( (Z1) \)

Observations:

- Recall \[ q, r \in \text{Basic and } q, r \in \text{TauFree and } \text{Act}(q), \text{Act}(r) \subseteq \text{dom}(\Xi) \]
  and \( \text{Bound}(q) \cap \text{we} = \emptyset \) and \( \text{Bound}(r) \cap \text{wf} = \emptyset \) and \( e \neq f \)

by Z1. Then, conclude

\[ \sum_{i=1}^{m} \sum_{j=1}^{l}(\text{iso}(\nu v_{i}, \beta_{i}) | \text{iso}(\nu u_{j}, \gamma_{j})) \cdot \text{iso}(\nu u_{j}, \gamma_{j}) \cdot ?(q' \parallel r')) + \sum_{i=1}^{m} \sum_{j=1}^{l}(\text{iso}(\nu v_{i}, \beta_{i}) | \text{iso}(\nu u_{j}, \gamma_{j})) \cdot ?(q' \parallel r''), \]

by Proposition 18 (Z2).

- Recall \[ \sum_{i=1}^{m} \sum_{j=1}^{l}(\text{iso}(\nu v_{i}, \beta_{i}) | \text{iso}(\nu u_{j}, \gamma_{j})) \cdot \text{iso}(\nu u_{j}, \gamma_{j}) \cdot ?(q' \parallel r')) + \sum_{i=1}^{m} \sum_{j=1}^{l}(\text{iso}(\nu v_{i}, \beta_{i}) | \text{iso}(\nu u_{j}, \gamma_{j})) \cdot ?(q' \parallel r''), \]

by Z2 and \( e \neq f \) by Z1. Then, conclude

\[ \sum_{i=1}^{m} \sum_{j=1}^{l}(\text{iso}(\nu v_{i}, \beta_{i}) | \text{iso}(\nu u_{j}, \gamma_{j})) \cdot ?(q' \parallel r')) + \sum_{i=1}^{m} \sum_{j=1}^{l}(\text{iso}(\nu v_{i}, \beta_{i}) | \text{iso}(\nu u_{j}, \gamma_{j})) \cdot ?(q' \parallel r''), \]

by Proposition 3 (Z3).

- Conclude (Z4):
\[ (?((\text{isol}(\text{wev}, q) \cdot q') \mid \text{isol}(\text{wfu}, r')) \]

\[ \cong (?((\sum_{i=1}^{m} \text{isol(wev}_i, \beta_i) + \sum_{i=1}^{m'} \text{isol(wev}}_{i'}, \beta'_{i'}) \cdot q_i) \mid \text{isol}(\text{wfu}_i, \gamma_i) + \sum_{i=1}^{l'} (\text{isol}(\text{wfu}_i', \gamma'_{i'}) \cdot r_{i'})) \]

\[ \cong (?((\sum_{i=1}^{m} \text{isol(wev}_i, \beta_i) \mid \text{isol}(\text{wfu}_i, \gamma_i)) + \sum_{i=1}^{m'} \text{isol(wev}}_{i'}, \beta'_{i'}) \cdot q_i) \mid \text{isol}(\text{wfu}_i', \gamma'_{i'}) \cdot r_{i'}) \) + 

\[ \sum_{i=1}^{m'} \sum_{j=1}^{l'} ((\text{isol}(\text{wev}_i', \beta'_{i'}) \cdot q'_j) \mid (\text{isol}(\text{wfu}_i', \gamma'_{i'}) \cdot r'_{j})) \]
Proof (of Proposition 8). Assumptions:

- \( p \in \text{Basic and } p \in \text{TauFree and } \text{Act}(p) \subseteq \text{dom}(\Xi) \text{ and } \text{Bound}(p) \cap w = \emptyset \) (Z1).

Observations:

- Recall

\[
p \in \text{Basic and } p \in \text{TauFree and } \text{Act}(p) \subseteq \text{dom}(\Xi) \text{ and } \text{Bound}(p) \cap w = \emptyset
\]

by Z1. Then, conclude

\[
\sim \sum_{i=1}^{m} \sum_{j=1}^{l} (\text{iso}(wv_i, \beta_i) \cdot \gamma_i) + \sum_{i=1}^{m} \sum_{j=1}^{l} (\text{iso}(wu_j, \alpha_j) \cdot \gamma_j) + \sum_{i=1}^{m} \sum_{j=1}^{l} (\text{iso}(wu_j', \alpha_j') \cdot \gamma_j')
\]

and \( \gamma_i \in \text{TauFree and } \text{Act}(\beta_i) \subseteq \text{dom}(\Xi) \text{ for all } 1 \leq i \leq m \) and \( \beta_i' \in \text{TauFree and } \text{Act}(\alpha_i') \subseteq \text{dom}(\Xi) \text{ for all } 1 \leq i \leq m' \)

by Proposition [8](Z2).
• Conclude $[\text{Bound}(\alpha_i) = \emptyset \forall 1 \leq i \leq \ell]$ by the definition of Bound. Then, conclude $[\text{Bound}(\alpha_i) \cap w\ell_i = \emptyset \forall 1 \leq i \leq \ell]$ by the definition of $\cap$ (Z3).

• Conclude $[\text{Bound}(\alpha_i') = \emptyset \forall 1 \leq i \leq \ell]$ by the definition of Bound. Then, conclude $[\text{Bound}(\alpha_i') \cap w\ell_i' = \emptyset \forall 1 \leq i \leq \ell]$ by the definition of $\cap$ (Z4).

• Recall $[[\alpha_i \in \text{Basic for all } 1 \leq i \leq n]]$ by the definition of Basic and $[[\alpha_i \in \text{TauFree and Act(\alpha_i) \subseteq dom(\Xi)} \forall 1 \leq i \leq n]]$ by Z2 and $[[\text{Bound}(\alpha_i) \cap w\ell_i = \emptyset \forall 1 \leq i \leq \ell]]$ by Z3. Then, conclude $[\exists (\text{isol}(w\ell_i, \alpha_i)) \simeq \delta \forall 1 \leq i \leq n]$ by Proposition 5 (Z5).

• Recall $[[\alpha_i' \in \text{Basic for all } 1 \leq i \leq n']]$ by the definition of Basic and $[[\alpha_i' \in \text{TauFree and Act(\alpha_i') \subseteq dom(\Xi)} \forall 1 \leq i \leq n']]$ by Z2 and $[[\text{Bound}(\alpha_i') \cap w\ell_i' = \emptyset \forall 1 \leq i \leq \ell]]$ by Z4. Then, conclude $[\exists (\text{isol}(w\ell_i', \alpha_i')) \simeq \delta \forall 1 \leq i \leq n]$ by Proposition 5 (Z6).

• Conclude (Z7):

$$
\begin{align*}
\sum_{i=1}^{\ell}(\text{isol}(w_i, \alpha_i) + & \sum_{i=1}^{\ell'}(\text{isol}(w_i', \alpha_i') \cdot p_i')) \parallel q) \\
\sum_{i=1}^{\ell}(\text{isol}(w_i, \alpha_i) \parallel q) + & \sum_{i=1}^{\ell'}(\text{isol}(w_i', \alpha_i') \cdot (p_i' \parallel q)) \\
\sum_{i=1}^{\ell}(\text{isol}(w_i, \alpha_i) \cdot q) + & \sum_{i=1}^{\ell'}(\text{isol}(w_i', \alpha_i') \cdot (p_i' \parallel q)) \\
\sum_{i=1}^{\ell}(\text{isol}(w_i, \alpha_i) \cdot (p_i' \parallel q)) + & \sum_{i=1}^{\ell'}(\text{isol}(w_i', \alpha_i') \cdot (p_i' \parallel q)) \\
\sum_{i=1}^{\ell} \delta' + & \sum_{i=1}^{\ell'} \delta' \\
\text{by the definition of } \text{Bound(\alpha_i)} \parallel q) \\
\sum_{i=1}^{\ell}(\text{isol}(w_i, \alpha_i) \cdot (p_i' \parallel q)) + & \sum_{i=1}^{\ell'}(\text{isol}(w_i', \alpha_i') \cdot (p_i' \parallel q)) \\
\sum_{i=1}^{\ell} \delta' + & \sum_{i=1}^{\ell'} \delta'.
\end{align*}
$$

• Conclude (Z8):

$$
\begin{align*}
\exists (\text{isol}(w, p) \cdot p') \parallel q) \\
\exists (\text{isol}(w, p) \cdot (p' \parallel q)) + \exists (\text{isol}(w, p) \cdot (p' \parallel q)) \\
\exists (\text{isol}(w, p) \cdot (p' \parallel q)) + \exists (\text{isol}(w, p) \cdot (p' \parallel q)) \\
\exists (\text{isol}(w, p) \cdot (p' \parallel q)) + \exists (\text{isol}(w, p) \cdot (p' \parallel q)) \\
\exists (\text{isol}(w, p) \cdot (p' \parallel q)) + \exists (\text{isol}(w, p) \cdot (p' \parallel q)) \\
\exists (\text{isol}(w, p) \cdot (p' \parallel q)) + \exists (\text{isol}(w, p) \cdot (p' \parallel q)).
\end{align*}
$$

Conclude the consequent of this proposition by and-ing the results in Z7 and Z8.  

\textbf{D. Proofs for Section 5.3}

\textbf{Proof (of Lemma 1).} Assumptions:
• \([p \in \text{Basic} \text{ and } p \in \text{TauFree} \text{ and } \text{Act}(p) \subseteq \text{dom}(\Xi) \text{ and } \text{Bound}(p) \cap w = \emptyset]\) (Z1).

**Observations:**

• Recall \([p \in \text{Basic} \text{ and } p \in \text{TauFree} \text{ and } \text{Act}(p) \subseteq \text{dom}(\Xi) \text{ and } \text{Bound}(p) \cap w = \emptyset]\) by Z1. Then, conclude

\[
\vdash \exists p \exists q \exists r \exists w (\exists p \exists q \exists r \exists w \bigl( p \in \text{Basic} \text{ and } p \in \text{TauFree} \text{ and } \text{Act}(p) \subseteq \text{dom}(\Xi) \text{ and } \text{Bound}(p) \cap w = \emptyset \bigr) \rightarrow \neg \exists q \exists r \exists w \exists p \bigl( q \in \text{Basic} \text{ and } q \in \text{TauFree} \bigr) \text{ and } \exists q \exists r \exists w \exists p \bigl( r \in \text{Basic} \text{ and } r \in \text{TauFree} \bigr) \text{ and } \exists q \exists r \exists w \exists p \bigl( \text{Act}(p) \subseteq \text{dom}(\Xi) \text{ and } \text{Bound}(p) \cap w = \emptyset \bigr) \rightarrow \neg \exists q \exists r \exists w \exists p \bigl( \text{Act}(p) \subseteq \text{dom}(\Xi) \text{ and } \text{Bound}(p) \cap w = \emptyset \bigr)
\]

by Proposition 8 (Z2).

• Conclude (Z3):

\[
\vdash \exists p \exists q \exists r \exists w \exists p \bigl( p \in \text{Basic} \text{ and } p \in \text{TauFree} \text{ and } \text{Act}(p) \subseteq \text{dom}(\Xi) \text{ and } \text{Bound}(p) \cap w = \emptyset \bigr) \rightarrow \neg \exists q \exists r \exists w \exists p \bigl( q \in \text{Basic} \text{ and } q \in \text{TauFree} \bigr) \text{ and } \exists q \exists r \exists w \exists p \bigl( r \in \text{Basic} \text{ and } r \in \text{TauFree} \bigr) \text{ and } \exists q \exists r \exists w \exists p \bigl( \text{Act}(p) \subseteq \text{dom}(\Xi) \text{ and } \text{Bound}(p) \cap w = \emptyset \bigr) \rightarrow \neg \exists q \exists r \exists w \exists p \bigl( \text{Act}(p) \subseteq \text{dom}(\Xi) \text{ and } \text{Bound}(p) \cap w = \emptyset \bigr)
\]

• Conclude (Z4):

\[
\vdash \exists p \exists q \exists r \exists w \exists p \bigl( p \in \text{Basic} \text{ and } p \in \text{TauFree} \text{ and } \text{Act}(p) \subseteq \text{dom}(\Xi) \text{ and } \text{Bound}(p) \cap w = \emptyset \bigr) \rightarrow \neg \exists q \exists r \exists w \exists p \bigl( q \in \text{Basic} \text{ and } q \in \text{TauFree} \bigr) \text{ and } \exists q \exists r \exists w \exists p \bigl( r \in \text{Basic} \text{ and } r \in \text{TauFree} \bigr) \text{ and } \exists q \exists r \exists w \exists p \bigl( \text{Act}(p) \subseteq \text{dom}(\Xi) \text{ and } \text{Bound}(p) \cap w = \emptyset \bigr) \rightarrow \neg \exists q \exists r \exists w \exists p \bigl( \text{Act}(p) \subseteq \text{dom}(\Xi) \text{ and } \text{Bound}(p) \cap w = \emptyset \bigr)
\]

Conclude the consequent of this lemma by and-ing the results in Z3 and Z4. \(\square\)

**Proof** (of Lemma 3). Assumptions:

• \([q + r \in \text{Basic and } q + r \in \text{TauFree and } \text{Act}(q + r) \subseteq \text{dom}(\Xi) \text{ and } \text{Bound}(q + r) \cap w = \emptyset]\) (Z1)

**Observations:**

• Recall \([q + r \in \text{Basic}\) by Z1. Then, conclude \(q, r, q + r \in \text{Basic}\) by the definition of Basic (Z2).

• Recall \([q + r \in \text{TauFree}\) by Z1. Then, conclude \(q, r, q + r \in \text{TauFree}\) by the definition of TauFree (Z3).
Recall $\text{Act}(q + r) \subseteq \text{dom}(\exists)$ by Z1. Then, because $\text{Act}(q + r) = \text{Act}(q) \cup \text{Act}(r)$ by the definition of $\text{Act}$, conclude $\text{Act}(q) \cup \text{Act}(r) \subseteq \text{dom}(\exists)$. Then, conclude $\text{Act}(q), \text{Act}(r) \subseteq \text{dom}(\exists)$ by ZFC. Then, because $\text{Act}(q + r) \subseteq \text{dom}(\exists)$ by Z1, conclude $\text{Act}(q), \text{Act}(r), \text{Act}(q + r) \subseteq \text{dom}(\exists)$ (Z4).

Recall $\text{Bound}(q + r) \cap w = \emptyset$ by Z1. Then, because $\text{Bound}(q + r) = \text{Bound}(q) \cup \text{Bound}(r)$ by the definition of $\text{Bound}$, conclude $\text{Bound}(q) \cup \text{Bound}(r) \cap w = \emptyset$. Then, conclude $[\text{Bound}(q) \cap w = \emptyset \text{ and } \text{Bound}(r) \cap w = \emptyset]$ by ZFC (Z5).

Recall $[q, r, q + r \in \text{Basic by Z2}]$ and $[q, r, q + r \in \text{TauFree by Z3}]$ and $[\text{Act}(q), \text{Act}(r), \text{Act}(q + r) \subseteq \text{dom}(\exists) \text{ by Z4}]$ and $[[\text{Bound}(q) \cap w = \emptyset \text{ and } \text{Bound}(r) \cap w = \emptyset \text{ and } \text{Bound}(q + r) \cap w = \emptyset]]$ by Z8. Then, conclude

\[ ?(\text{isol}(w_1, q) \parallel \overline{\text{isol}(w_1, q)}) \simeq ?(\text{isol}(w_1, q) \mid \overline{\text{isol}(w_1, q)}) \]

and

\[ ?(\text{isol}(w_2, r) \parallel \overline{\text{isol}(w_2, r)}) \simeq ?(\text{isol}(w_2, r) \mid \overline{\text{isol}(w_2, r)}) \]

and

\[ ?(\text{isol}(w, q + r) \parallel \overline{\text{isol}(w, q + r)}) \simeq ?(\text{isol}(w, q + r) \mid \overline{\text{isol}(w, q + r)}) \]

by Lemma 1 (Z6).

Conclude $1, 2 \notin \text{VVar}$ by Definition 2 Then, because $\text{Bound}(q), \text{Bound}(r) \subseteq \text{VVar}$ by the definition of $\text{Bound}$, conclude $1, 2 \notin \text{Bound}(q), \text{Bound}(r)$ (Z7).

Recall $[1, 2 \notin \text{Bound}(q), \text{Bound}(r) \text{ by Z7}]$ and $[[\text{Bound}(q) \cap w = \emptyset \text{ and } \text{Bound}(r) \cap w = \emptyset]]$ by Z5. Then, conclude $[\text{Bound}(q) \cap w_1 = \emptyset \text{ and } \text{Bound}(r) \cap w_2 = \emptyset]$ by Proposition 16. Then, because $\text{Bound}(q + r) \cap w = \emptyset$ by Z1, conclude $[\text{Bound}(q) \cap w_1 = \emptyset \text{ and } \text{Bound}(r) \cap w_2 = \emptyset \text{ and } \text{Bound}(q + r) \cap w = \emptyset]$ (Z8).

Conclude $1 \neq 2$ by Definition 2. Then, because $[1^1 = 1 \text{ and } 2^2 = 2]$ by the definition of $g$, conclude $1^1 \neq 2^2$. Then, conclude $w^1 \neq w^2$ by ZFC. Then, conclude $(w_1)^2 \neq (w_2)^2$ by the definition of $g$ (Z9).

Recall $[q, r \in \text{Basic by Z2}]$ and $[q, r \in \text{TauFree by Z3}]$ and $[\text{Act}(q), \text{Act}(r) \subseteq \text{dom}(\exists) \text{ by Z4}]$ and $[[\text{Bound}(q) \cap w_1 = \emptyset \text{ and } \text{Bound}(r) \cap w_2 = \emptyset]]$ by Z8 and $[(w_1)^2 \neq (w_2)^2 \text{ by Z9}]$. Then, conclude

\[ ?(\text{isol}(w_1, q) \mid \text{isol}(w_2, r)) \simeq \delta \text{ and } ?(\text{isol}(w_2, r) \mid \text{isol}(w_1, q)) \simeq \delta \]

by Proposition 4 (Z10).

Conclude:

\[ \text{split}(w, q + r) \]

\[ \overset{\text{-def}}{=} ?(\text{isol}(w, q + r) \parallel \overline{\text{isol}(w, q + r)}) \]

\[ \overset{\text{Z6}}{=} ?(\text{isol}(w, q + r) \mid \overline{\text{isol}(w, q + r)}) \]

\[ \overset{\text{Z7}}{=} ?((\text{isol}(w_1, q) + \text{isol}(w_2, r)) \mid (\overline{\text{isol}(w_1, q)} + \overline{\text{isol}(w_2, r)})) \]

\[ \overset{\text{Z3}}{=} ?(\text{isol}(w_1, q) \mid \overline{\text{isol}(w_1, q)} + \text{isol}(w_1, q) \mid \overline{\text{isol}(w_2, r)} + \text{isol}(w_1, q) \mid \overline{\text{isol}(w_2, r)} + \text{isol}(w_2, r) \mid \overline{\text{isol}(w_2, r)}) \]

\[ \overset{\text{Z3}}{=} ?(\text{isol}(w_2, r) \mid \overline{\text{isol}(w_1, q)} + ?(\text{isol}(w_1, q) \mid \overline{\text{isol}(w_2, r)} + ?(\text{isol}(w_2, r) \mid \overline{\text{isol}(w_2, r)}) \]
Proof (of Lemma 4). By case distinction on the value of $c$.

**Case: $c \approx true$.** Conclude:

\[ \text{split}(w, c \rightarrow q \circ r) \]

\[ = (\text{split}(w, c \rightarrow q \circ r)) \]

\[ = (\text{split}(w, c \rightarrow q \circ r)) \]

\[ \text{split}(w, q) \]

\[ \text{split}(w, q) \]

**Case: $c \approx false$.**

\[ \text{split}(w, c \rightarrow q \circ r) \]

\[ = (\text{split}(w, c \rightarrow q \circ r)) \]

\[ = (\text{split}(w, c \rightarrow q \circ r)) \]

\[ \text{split}(w, q) \]

\[ \text{split}(w, q) \]

Proof (of Lemma 3). By case distinction on the value of $c$.

**Case: $c \approx true$.** Conclude:

\[ \text{split}(w, c \rightarrow q \circ r) \]

\[ = (\text{split}(w, c \rightarrow q \circ r)) \]

\[ = (\text{split}(w, c \rightarrow q \circ r)) \]

\[ \text{split}(w, q) \]

\[ \text{split}(w, q) \]

**Case: $c \approx false$.**

\[ \text{split}(w, c \rightarrow q \circ r) \]

\[ = (\text{split}(w, c \rightarrow q \circ r)) \]

\[ = (\text{split}(w, c \rightarrow q \circ r)) \]

\[ \text{split}(w, q) \]

\[ \text{split}(w, q) \]

Proof (of Lemma 1). Assumptions:

- \[ \sum_{x \in \{d_1, \ldots, d_e\}} q \in \text{Basic} \text{ and } \sum_{x \in \{d_1, \ldots, d_e\}} q \in \text{TauFree} \text{ and } \text{Act}(\sum_{x \in \{d_1, \ldots, d_e\}} q) \subseteq \text{dom}(\Xi) \text{ and Bound}(\sum_{x \in \{d_1, \ldots, d_e\}} q) \cap w = \emptyset \] (Z1).

Observations:

- Recall \( \sum_{x \in \{d_1, \ldots, d_e\}} q \in \text{Basic} \) by Z1. Then, conclude \( q, \sum_{x \in \{d_1, \ldots, d_e\}} q \in \text{Basic} \) by the definition of Basic (Z2).

- Recall \( \sum_{x \in \{d_1, \ldots, d_e\}} q \in \text{TauFree} \) by Z1. Then, conclude \( q, \sum_{x \in \{d_1, \ldots, d_e\}} q \in \text{TauFree} \) by the definition of TauFree (Z3).
• Recall $\text{Act}(\sum_{x \in \{d_1, \ldots, d_\ell\}} q) \subseteq \text{dom}(\Xi)$ by Z1. Then, because $\text{Act}(\sum_{x \in \{d_1, \ldots, d_\ell\}} q) = \text{Act}(q)$ by the definition of $\text{Act}$, conclude $\text{Act}(q) \subseteq \text{dom}(\Xi)$. Then, because $\text{Act}(\sum_{x \in \{d_1, \ldots, d_\ell\}} q) \subseteq \text{dom}(\Xi)$ by Z1, conclude $\text{Act}(q)$, $\text{Act}(\sum_{x \in \{d_1, \ldots, d_\ell\}} q) \subseteq \text{dom}(\Xi)$ (Z4).

• Recall $\text{Bound}(\sum_{x \in \{d_1, \ldots, d_\ell\}} q) \cap w = \emptyset$ by Z1. Then, because $\text{Bound}(\sum_{x \in \{d_1, \ldots, d_\ell\}} q) = \text{Bound}(q) \cup \{x\}$ by the definition of $\text{Bound}$, conclude $\text{Bound}(q) \cup \{x\} \cap w = \emptyset$. Then, conclude $[\text{Bound}(q) \cap w = \emptyset \text{ and } \{x\} \cap w = \emptyset]$ by ZFC (Z5).

• Recall $\sum_{x \in \{d_1, \ldots, d_\ell\}} q \in \text{Basic}$ by Z1. Then, conclude $x \notin \text{Bound}(q)$ by the definition of Basic (Z6).

• Recall $[x \notin \text{Bound}(q) \text{ by Z6}]$ and $[\text{Bound}(q) \cap w = \emptyset \text{ by Z5}]$. Then, conclude $\text{Bound}(q) \cap wx = \emptyset$ by Proposition 16 (Z7).

• Recall $[\text{Bound}(q) \cap w x = \emptyset \text{ by Z7}]$ and $[\text{Bound}(\sum_{x \in \{d_1, \ldots, d_\ell\}} q) \cap w = \emptyset]$ by Z1. Then, conclude $[\text{Bound}(q) \cap w x = \emptyset \text{ and } \text{Bound}(\sum_{x \in \{d_1, \ldots, d_\ell\}} q) \cap w = \emptyset]$ by ZFC (Z8).

• Recall $[g, \sum_{x \in \{d_1, \ldots, d_\ell\}} q \in \text{Basic} \text{ by Z2}]$ and $[g, \sum_{x \in \{d_1, \ldots, d_\ell\}} q \in \text{TauFree} \text{ by Z3}]$ and $[\text{Act}(q), \text{Act}(\sum_{x \in \{d_1, \ldots, d_\ell\}} q) \subseteq \text{dom}(\Xi) \text{ by Z4}]$ and $[[\text{Bound}(q) \cap wx = \emptyset \text{ and } \text{Bound}(\sum_{x \in \{d_1, \ldots, d_\ell\}} q) \cap w = \emptyset]] \text{ by Z8}$. Then, conclude $\exists ! (\text{isol}(wx \cap q) \parallel \overline{\text{isol}}(wx \cap q)) \sim \exists ! (\text{isol}(wx, q) \mid \overline{\text{isol}}(wx, q))$ and $\exists ! (\text{isol}(w, \sum_{x \in \{d_1, \ldots, d_\ell\}} q) \parallel \overline{\text{isol}}(w, \sum_{x \in \{d_1, \ldots, d_\ell\}} q)) \sim \exists ! (\text{isol}(w, \sum_{x \in \{d_1, \ldots, d_\ell\}} q) \mid \overline{\text{isol}}(w, \sum_{x \in \{d_1, \ldots, d_\ell\}} q))$ by Lemma 1 (Z9).

• Recall $[\{x\} \cap w = \emptyset]$ by Z5. Then, conclude $x \notin w$ by the definition of $\cap$ (Z10).

• Recall $[q \in \text{Basic} \text{ by Z2}]$ and $[x \notin w \text{ by Z10}]$ and $[\text{Bound}(q) \cap wx = \emptyset \text{ by Z7}]$. Then, conclude $[\text{isol}(wx, q)[d_i/x] = \text{isol}(wd_i, q[d_i/x]) \text{ for all } 1 \leq i \leq \ell]$ by Proposition 17 (Z11).

• Conclude $[d_i \in \text{Elem} \text{ for all } 1 \leq i \leq \ell]$ by the definition of $\sum$. Then, because $[\text{Bound}(q) \subseteq \text{Var} \text{ by the definition of Bound}]$ and $[\text{Bound} \cap \text{Var} = \emptyset]$ by Definition 1, conclude $[d_i \notin \text{Bound}(q) \text{ for all } 1 \leq i \leq \ell]$ (Z12).

• Recall $[[d_i \notin \text{Bound}(q) \text{ for all } 1 \leq i \leq \ell]$ by Z12] and $[\text{Bound}(q) \cap w = \emptyset]$ by Z5. Then, conclude $[\text{Bound}(q) \cap wd_i = \emptyset \text{ for all } 1 \leq i \leq \ell]$ (Z13).

• Recall $[q \in \text{Basic} \text{ by Z2}]$ and $[q \in \text{TauFree} \text{ by Z3}]$ and $[\text{Act}(q) \subseteq \text{dom}(\Xi) \text{ by Z4}]$ and $[[\text{Bound}(q) \cap wd_i = \emptyset \text{ for all } 1 \leq i \leq \ell]$ by Z13] and $[[d_i \neq d_j \text{ or } i = j] \text{ for all } 1 \leq i, j \leq \ell]$ by ZFC. Then, conclude $[[\exists ! (\text{isol}(wd_i, q[d_i/x]) \mid \overline{\text{isol}}(wd_j, q[d_j/x])) \sim \delta \text{ or } i = j] \text{ for all } 1 \leq i, j \leq \ell]$ by Proposition 7. Then, conclude $\sum_{i=1}^{\ell} \sum_{j=1}^{i-1} ?(\text{isol}(wd_i, q[d_i/x]) \mid \overline{\text{isol}}(wd_j, q[d_j/x])) \sim \delta$ and $\sum_{i=1}^{\ell} \sum_{j=i+1}^{\ell} ?(\text{isol}(wd_i, q[d_i/x]) \mid \overline{\text{isol}}(wd_j, q[d_j/x])) \sim \delta$ by ZFC (Z14).
Conclude:

\[
\text{split}(w, \sum_{x \in \{d_1, \ldots, d_t\}} q) \\
\text{?}(\text{isol}(w, \sum_{x \in \{d_1, \ldots, d_t\}} q) \parallel \text{isol}(w, \sum_{x \in \{d_1, \ldots, d_t\}} q)) \\
\text{?}(\text{isol}(w, \sum_{x \in \{d_1, \ldots, d_t\}} q) \parallel \text{isol}(w, \sum_{x \in \{d_1, \ldots, d_t\}} q)) \\
\text{by Lemma 1 (Z2).}
\]

by Lemma \[\text{Z1}\].

\[ Z1 \]

Lemma 6 (Prepreservation lemma for \(\cdot\)).

\[
[p \in \text{Basic} \text{ and } p \in \text{TauFree and } \text{Act}(p) \subseteq \text{dom}(\Xi) \text{ and } \text{Bound}(p) \cap w = \emptyset] \implies \text{?}((\text{isol}(w, p) \cdot p') \parallel (\text{isol}(w, p) \cdot p')) \simeq \text{split}(w, p) \cdot ?(p' \parallel p')
\]

Proof. Assumptions:

- \([p \in \text{Basic} \text{ and } p \in \text{TauFree and } \text{Act}(p) \subseteq \text{dom}(\Xi) \text{ and } \text{Bound}(p) \cap w = \emptyset] \text{ (Z1).}]

Observations:

- Recall \([p \in \text{Basic} \text{ and } p \in \text{TauFree and } \text{Act}(p) \subseteq \text{dom}(\Xi) \text{ and } \text{Bound}(p) \cap w = \emptyset] \text{ by Z1. Then, conclude}

  \[ \text{?}((\text{isol}(w, p) \cdot p') \parallel (\text{isol}(w, p) \cdot p')) \simeq \text{split}(w, p) \cdot ?(p' \parallel p') \text{ and}
\]

by Lemma \[\text{Z1}\].
Proceed by induction on the structure of $p$.

**Base:** $p = \alpha$. Conclude:

$$\begin{align*}
\top((\text{isol}(w, p) \cdot p') \parallel (\text{isol}(w, p) \cdot p')) \\
\top((\text{isol}(w, p) \cdot p') \mid (\text{isol}(w, p) \cdot p')) \\
\top((\text{isol}(w, \alpha) \cdot p') \mid (\text{isol}(w, \alpha) \cdot p')) \\
\top((\text{isol}(w, \alpha) \mid \text{isol}(w, \alpha) \cdot (p' \parallel p')) \\
\top((\text{isol}(w, p) \mid \text{isol}(w, p)) \cdot (p' \parallel p')) \\
\top((\text{isol}(w, p) \parallel \text{isol}(w, p)) \cdot (p' \parallel p')) \\
\text{split}(w, p) \cdot (p' \parallel p')
\end{align*}$$

**Step:** $[p = q + r$ or $p = q \cdot r$ or $p = c \rightarrow q \circ r$ or $p = \sum_{x \in \{d_1, ..., d_\ell\} q}]$. Assumptions:

- Induction hypothesis (IH):
  
  $\begin{align*}
  \left[ \begin{array}{c}
  \hat{p} \in \text{Basic and } \hat{p} \in \text{TauFree and} \\
  \text{Act}(\hat{p}) \subseteq \text{dom}(\Xi) \text{ and } \text{Bound}(\hat{p}) \cap \hat{w} = \emptyset
  \end{array} \right] \text{ implies } \\
  \top((\text{isol}(\hat{w}, \hat{p}) \cdot \hat{p}') \mid (\text{isol}(\hat{w}, \hat{p}) \cdot \hat{p}')) \simeq \text{split}(\hat{w}, \hat{p}) \cdot (\hat{p}' \parallel \hat{p}')
  \end{align*}$

  for all $\hat{p} \in \{q, r\} \cup \{q[d_i/x] \mid 1 \leq i \leq \ell\}$

Proceed by case distinction on the structure of $p$.

**Case:** $[p = q + r$ or $p = q \cdot r$ or $p = c \rightarrow q \circ r]$. Observations:

- Recall $p \in \text{Basic}$ by Z1. Then, because $[p = q + r$ or $p = q \cdot r$ or $p = c \rightarrow q \circ r]$ by the definition of this case, conclude $[q + r \in \text{Basic or } q \cdot r \in \text{Basic or } c \rightarrow q \circ r \in \text{Basic}]$. Then, conclude $q, r \in \text{Basic}$ by the definition of Basic (Z3).
- Recall $p \in \text{TauFree}$ by Z1. Then, because $[p = q + r$ or $p = q \cdot r$ or $p = c \rightarrow q \circ r]$ by the definition of this case, conclude $[q + r \in \text{TauFree or } q \cdot r \in \text{TauFree or } c \rightarrow q \circ r \in \text{TauFree}]$. Then, conclude $q, r \in \text{TauFree}$ by the definition of TauFree (Z4).
- Recall $\text{Act}(p) \subseteq \text{dom}(\Xi)$ by Z1. Then, because $[p = q + r$ or $p = q \cdot r$ or $p = c \rightarrow q \circ r]$ by the definition of this case, conclude
  
  $\text{Act}(q + r) \subseteq \text{dom}(\Xi)$ or $\text{Act}(q \cdot r) \subseteq \text{dom}(\Xi)$

  or $\text{Act}(c \rightarrow q \circ r) \subseteq \text{dom}(\Xi)$

Then, conclude $\text{Act}(q) \cup \text{Act}(r) \subseteq \text{dom}(\Xi)$ by the definition of Act. Then, conclude $\text{Act}(q), \text{Act}(r) \subseteq \text{dom}(\Xi)$ by ZFC (Z5).
- Recall $\text{Bound}(p) \cap w = \emptyset$ by Z1. Then, because $[p = q + r$ or $p = q \cdot r$ or $p = c \rightarrow q \circ r]$ by the definition of this case, conclude
  
  $\text{Bound}(q + r) \cap w = \emptyset$ or $\text{Bound}(q \cdot r) \cap w = \emptyset$

  or $\text{Bound}(c \rightarrow q \circ r) \cap w = \emptyset$

Then, conclude $\text{Bound}(q) \cup \text{Bound}(r) \cap w = \emptyset$ by the definition of Bound. Then, conclude $[\text{Bound}(q) \cap w = \emptyset$ and $\text{Bound}(r) \cap w = \emptyset]$ by ZFC (Z6).
- Conclude 1, 2 $\notin \text{Var}$ by Definition 2 Then, because $\text{Bound}(q), \text{Bound}(r) \subseteq \text{Var}$ by the definition of Bound, conclude 1, 2 $\notin \text{Bound}(q), \text{Bound}(r)$ (Z7).
• Recall \([1, 2 \notin \text{Bound}(q), \text{Bound}(r) \text{ by } Z7]\) and \([[\text{Bound}(q) \land w = 0 \text{ and Bound}(r) \land w = 0]] \text{ by } Z10\). Then, conclude \([[\text{Bound}(q) \land w = 0 \text{ and Bound}(r) \land w = 0]] \text{ by Proposition 16 (Z8)}.\)

• Recall \([q, r \in \text{Basic by } Z3]\) and \([q, r \in \text{TauFree by } Z4]\) and \([\text{Act}(q), \text{Act}(r) \subseteq \text{dom}(\Xi) \text{ by } Z5]\) and \([[\text{Bound}(q) \land w = 0 \text{ and Bound}(r) \land w = 0]] \text{ by } Z8\). Then, conclude

\[
\begin{align*}
&\neg((\text{iso}(w_1, q) \cdot p') \mid (\text{iso}(w_1, q) \cdot p')) \simeq \text{split}(w_1, q) \cdot \neg(p' \parallel p') \\
&\\text{and } \neg((\text{iso}(w_2, r) \cdot p') \mid (\text{iso}(w_2, r) \cdot p')) \simeq \text{split}(w_2, r) \cdot \neg(p' \parallel p') \\
&\\text{and } \neg((\text{iso}(w_1, q) \cdot \text{iso}(w_2, r)) \mid (\text{iso}(w_1, q) \cdot \text{iso}(w_2, r))) \\
&\simeq \text{split}(w_1, q) \cdot \neg((\text{iso}(w_2, r) \mid (\text{iso}(w_2, r)))
\end{align*}
\]

by IH (Z9).

Proceed by case distinction on the structure of \(p\).

**Case:** \(p = q + r\). Observations:

• Conclude \(1 \neq 2\) by Definition 2. Then, because \([1^2 = 1 \text{ and } 2^2 = 2]\) by the definition of \(\sharp\), conclude \(1^2 \neq 2^2\). Then, conclude \(w^2 \neq w^2\) by ZFC. Then, conclude \((w1)^2 \neq (w2)^4\) by the definition of \(\sharp\) (Z10).

• Recall \([q, r \in \text{Basic by } Z3]\) and \([q, r \in \text{TauFree by } Z4]\) and \([\text{Act}(q), \text{Act}(r) \subseteq \text{dom}(\Xi) \text{ by } Z5]\) and \([[\text{Bound}(q) \land w = 0 \text{ and Bound}(r) \land w = 0]] \text{ by } Z8\) and \([[w1] \neq (w2)^4\) by Z10. Then, conclude

\[
\begin{align*}
&\neg((\text{iso}(w_1, q) \cdot p') \mid (\text{iso}(w_2, r) \cdot p')) \simeq \delta \\
&\\text{and } \neg((\text{iso}(w_2, r) \cdot p') \mid (\text{iso}(w_1, q) \cdot p')) \simeq \delta
\end{align*}
\]

by Proposition 8 (Z11).

• Recall \([p \in \text{Basic by } Z3]\) and \([p \in \text{TauFree by } Z4]\) and \([\text{Act}(p) \subseteq \text{dom}(\Xi) \text{ by } Z5]\) and \([[\text{Bound}(p) \land w = 0]] \text{ by } Z1\). Then, because \(p = q + r\) by the definition of this case, conclude

\(q + r \in \text{Basic and } q + r \in \text{TauFree and } \text{Act}(q + r) \subseteq \text{dom}(\Xi) \text{ and Bound}(q + r) \subseteq w = 0\)

Then, conclude \(\text{split}(w, q + r) \simeq \text{split}(w_1, q) + \text{split}(w_2, r)\) by Lemma 2 (Z12).

Conclude:

\[
\begin{align*}
&l_2 \Rightarrow \neg((\text{iso}(w, p) \cdot p') \mid (\text{iso}(w, p) \cdot p')) \\
&\neg((\text{iso}(w, p) \cdot p') \mid (\text{iso}(w, p) \cdot p')) \\
&\Rightarrow \neg((\text{iso}(w, q + r) \cdot p') \mid (\text{iso}(w, q + r) \cdot p')) \\
&\Rightarrow \neg((\text{iso}(w_1, q) + \text{iso}(w_2, r)) \cdot p') \mid (\text{iso}(w_1, q) + \text{iso}(w_2, r)) \cdot p')) \\
&\Rightarrow \neg((\text{iso}(w_1, q) \cdot p') + (\text{iso}(w_2, r) \cdot p')) \mid (\text{iso}(w_1, q) \cdot p') + (\text{iso}(w_2, r) \cdot p')) \\
&\Rightarrow \neg((\text{iso}(w_1, q) \cdot p') \mid (\text{iso}(w_1, q) \cdot p') + (\text{iso}(w_2, r) \cdot p')) + (\text{iso}(w_2, r) \cdot p') \mid (\text{iso}(w_2, r) \cdot p'))
\end{align*}
\]
\textit{Case: } $p = q \cdot r$. Observations:

- Recall $[r \in \text{Basic by Z3}]$ and $[r \in \text{TauFree by Z4}]$ and $[\text{Act}(r) \subseteq \text{dom}(\Xi) \text{ by Z5}]$ and 
  $[\text{Bound}(r) \cap w = 0 = \emptyset \text{ by Z8}]$. Then, conclude

\[ ?((\text{isol}(w, r) \cdot p') \ | \ (\overline{\text{isol}}(w, r) \cdot p')) \]
\[ \simeq ?((\text{isol}(w, r) \cdot p') \ | \ (\overline{\text{isol}}(w, r) \cdot p')) \]

by Lemma [Z13].

Conclude:

\[ ?((\text{isol}(w, p) \cdot p') \ | \ (\overline{\text{isol}}(w, p) \cdot p')) \]
\[ \simeq ?((\text{isol}(w, p) \cdot p') \ | \ (\overline{\text{isol}}(w, p) \cdot p')) \]
\[ \text{Case } Z2 \]
\[ \simeq ?((\text{isol}(w, q \cdot r) \cdot p') \ | \ (\overline{\text{isol}}(w, q \cdot r) \cdot p')) \]
\[ \text{Case } Z2 \]
\[ \simeq ?((\text{isol}(w, q) \cdot \text{isol}(w, r) \cdot p') \ | \ (\overline{\text{isol}}(w, q) \cdot \overline{\text{isol}}(w, r) \cdot p')) \]
\[ Z2 \]
\[ \text{Case } Z2 \]
\[ \simeq ?(\text{isol}(w, q) \cdot \text{isol}(w, r) \cdot p') \ | \ (\overline{\text{isol}}(w, r) \cdot p')) \]
\[ \text{Case } Z13 \]
\[ \simeq ?(\text{isol}(w, q) \cdot \text{isol}(w, r) \cdot p') \ | \ (\overline{\text{isol}}(w, r) \cdot p')) \]
\[ \text{Case } Z9 \]
\[ \simeq ?((\text{isol}(w, q) \cdot \text{isol}(w, r)) \ | \ (\overline{\text{isol}}(w, q) \cdot \overline{\text{isol}}(w, r))) \cdot \]
\[ ?(p' \ | \ p') \]
\[ \text{Case } Z9 \]
\[ \simeq ?(\text{isol}(w, q \cdot r) \ | \ (\overline{\text{isol}}(w, q \cdot r) \cdot p')) \cdot \]
\[ ?(p' \ | \ p') \]
\[ \text{Case } Z12 \]
\[ \simeq ?((\text{isol}(w, p) \cdot p') \ | \ (\overline{\text{isol}}(w, p) \cdot p')) \]
\[ \text{Case } Z2 \]
\[ \simeq ?((\text{isol}(w, true \rightarrow q \cdot r) \cdot p') \ | \ (\overline{\text{isol}}(w, true \rightarrow q \cdot r) \cdot p')) \]
\[ \text{Case } Z2 \]
\[ \simeq ?((true \rightarrow \text{isol}(w, q) \cdot \text{isol}(w, r)) \cdot p') \ | \]
\[ (true \rightarrow \overline{\text{isol}}(w, q) \cdot \overline{\text{isol}}(w, r)) \cdot p') \]

\textit{Case: } $p = c \rightarrow q \cdot r$. Proceed by case distinction on the value of $c$.

\textit{Case: } $c \simeq \text{true}$. Conclude:

\[ ?((\text{isol}(w, p) \cdot p') \ | \ (\overline{\text{isol}}(w, p) \cdot p')) \]
\[ \simeq ?((\text{isol}(w, p) \cdot p') \ | \ (\overline{\text{isol}}(w, p) \cdot p')) \]
\[ \text{Case } Z2 \]
\[ \simeq ?((\text{isol}(w, true \rightarrow q \cdot r) \cdot p') \ | \ (\overline{\text{isol}}(w, true \rightarrow q \cdot r) \cdot p')) \]
\[ \text{Case } Z2 \]
\[ \simeq ?((true \rightarrow \text{isol}(w, q) \cdot \text{isol}(w, r)) \cdot p') \ | \]
\[ (true \rightarrow \overline{\text{isol}}(w, q) \cdot \overline{\text{isol}}(w, r)) \cdot p') \]

79
Case: $c \approx false$. Conclude:
\[ ((\text{isol}(w, p) \cdot p') | (\text{isol}(w, p) \cdot p')) \]
\[ split(w, p) \cdot (p' \parallel p') \]
\[ \sum \]
\[ = \sum \]
\[ = \sum \]
\[ = \sum \]
\[ by \] Z16.
Then, because $p = \sum_{d \in \{d_1, ..., d_k\}} q$ by the definition of this case, conclude $\sum_{d \in \{d_1, ..., d_k\}} q \in \text{Basic}$ by Proposition 16 (Z20).
\[ [d_i \in \text{Elem for all } 1 \leq i \leq \ell] \]
by the definition of $\sum$. Then, because $[\text{Bound}(q) \subseteq \text{Var by the definition of Bound}]$ and $[\text{Elem} \cap \text{Var} = \emptyset by Definition 1]$, conclude $[d_i \notin \text{Bound}(q) for all 1 \leq i \leq \ell] (Z22).$
• Recall $[d_i \notin \text{Bound}(q) \text{ for all } 1 \leq i \leq \ell]$ by Z22 and $[\text{Bound}(q) \cap w = \emptyset \text{ by Z17}].$ Then, conclude $[\text{Bound}(q) \cap w, d_i = \emptyset \text{ for all } 1 \leq i \leq \ell]$ (Z23).

• Recall $[q \in \text{Basic} \text{ by Z14}]$ and $[q \in \text{TauFree} \text{ by Z15}]$ and $[\text{Act}(q) \subseteq \text{dom}(\Xi) \text{ by Z16}]$ and $[[\text{Bound}(q) \cap w, d_i = \emptyset \text{ for all } 1 \leq i \leq \ell] \text{ by Z23} \text{ and } [[d_i \neq d_j \text{ or } i = j \text{ for all } 1 \leq i, j \leq \ell]] \text{ by ZFC}. \text{ Then, conclude}$

$$\exists ! ((\text{isol}(w, d_i, q(d_i/x)) \cdot p') | (\text{isol}(w, d_j, q(d_j/x)) \cdot p')) \simeq \delta \text{ or } i = j$$

by Proposition 7. Then, conclude

$$\sum_{i=1}^{\ell} \sum_{j=1}^{i-1} \exists ! ((\text{isol}(w, d_i, q(d_i/x)) \cdot p') | (\text{isol}(w, d_j, q(d_j/x)) \cdot p')) \simeq \delta$$

and

$$\sum_{i=1}^{\ell} \sum_{j=i+1}^{\ell} \exists ! ((\text{isol}(w, d_i, q(d_i/x)) \cdot p') | (\text{isol}(w, d_j, q(d_j/x)) \cdot p')) \simeq \delta$$

by ZFC (Z24).

• Recall $[\text{Bound}(q) \cap w, d_i = \emptyset \text{ for all } 1 \leq i \leq \ell] \text{ by Z23}.$ Then, conclude $[\text{Bound}(q[d_i/x]) \cap w, d_i = \emptyset \text{ for all } 1 \leq i \leq \ell] \text{ by the definition of Bound (Z25)}.\text{ Then, conclude}$

$$\exists ! ((\text{isol}(w, d_i, q(d_i/x)) \cdot p') | (\text{isol}(w, d_i, q(d_i/x)) \cdot p')) \simeq \text{split}(w, d_i, q(d_i/x)) \cdot (p' \parallel p')$$

for all $1 \leq i \leq \ell$

by IH (Z26).

• Recall $[p \in \text{Basic} \text{ by Z3}]$ and $[p \in \text{TauFree} \text{ by Z4}]$ and $[\text{Act}(p) \subseteq \text{dom}(\Xi) \text{ by Z5}]$ and $[\text{Bound}(p) \cap w = \emptyset \text{ by Z1}]. \text{ Then, because } p = \sum_{x \in \{d_1, \ldots, d_\ell\}} q \text{ by the definition of this case, conclude}$

$$\sum_{x \in \{d_1, \ldots, d_\ell\}} q \in \text{Basic and } \sum_{x \in \{d_1, \ldots, d_\ell\}} q \in \text{TauFree and} \text{ Act}(\sum_{x \in \{d_1, \ldots, d_\ell\}} q) \subseteq \text{dom}(\Xi) \text{ and Bound}(\sum_{x \in \{d_1, \ldots, d_\ell\}} q) \cap w = \emptyset$$

Then, conclude split$(w, \sum_{x \in \{d_1, \ldots, d_\ell\}} q) \simeq \sum_{x \in \{d_1, \ldots, d_\ell\}} \text{split}(wx, q)$ by Lemma 4 (Z27). Conclude:

$$\sum_{x \in \{d_1, \ldots, d_\ell\}} q \in \text{Basic and } \sum_{x \in \{d_1, \ldots, d_\ell\}} q \in \text{TauFree and} \text{ Act}(\sum_{x \in \{d_1, \ldots, d_\ell\}} q) \subseteq \text{dom}(\Xi) \text{ and Bound}(\sum_{x \in \{d_1, \ldots, d_\ell\}} q) \cap w = \emptyset$$

Then, conclude split$(w, \sum_{x \in \{d_1, \ldots, d_\ell\}} q) \simeq \sum_{x \in \{d_1, \ldots, d_\ell\}} \text{split}(wx, q)$ by Lemma 4 (Z27). Conclude:
Proof (of Lemma 5).

Recall of Bound Act \([q \cdot r \in \text{Basic} \quad \text{and} \quad q \cdot r \in \text{TauFree and} \quad \text{Act}(q \cdot r) \subseteq \text{dom}(\Xi) \quad \text{and} \quad \text{Bound}(q \cdot r) \cap w = \emptyset] \) \(Z1\).

Observations:

- Recall \(q \cdot r \in \text{Basic} \) by Z1. Then, conclude \(q \in \text{Basic} \) by the definition of Basic \((Z2)\).
- Recall \(q \cdot r \in \text{TauFree} \) by Z1. Then, conclude \(q \in \text{TauFree} \) by the definition of TauFree \((Z3)\).
- Recall \(\text{Act}(q \cdot r) \subseteq \text{dom}(\Xi) \) by Z1. Then, because \(\text{Act}(q \cdot r) = \text{Act}(q) \cup \text{Act}(r) \) by the definition of Act, conclude \(\text{Act}(q) \cup \text{Act}(r) \subseteq \text{dom}(\Xi) \). Then, conclude \(\text{Act}(q) \subseteq \text{dom}(\Xi) \) by ZFC \((Z4)\).
- Recall \(\text{Bound}(q \cdot r) \cap w = \emptyset \) by Z1. Then, because \(\text{Bound}(q \cdot r) = \text{Bound}(q) \cup \text{Bound}(r) \) by the definition of Bound, conclude \(\text{Bound}(q) \cup \text{Bound}(r) \cap w = \emptyset \). Then, conclude \(\text{Bound}(q) \cap w = \emptyset \) by ZFC \((Z5)\).

- Conclude \(1 \notin \text{Var} \) by Definition \[\text{Z2}\]. Then, because \(\text{Bound}(q) \subseteq \text{Var} \) by the definition of Bound, conclude \(1 \notin \text{Bound}(q) \) \(Z6\).
- Recall \([1 \notin \text{Bound}(q) \text{ by Z6}] \) and \([\text{Bound}(q) \cap w = \emptyset \text{ by Z5}] \). Then, conclude \(\text{Bound}(q) \cap w1 = \emptyset \) by Proposition \[\text{Z16}\] \(Z7\).
- Recall \([q \in \text{Basic} \text{ by Z2}] \) and \([q \in \text{TauFree by Z3}] \) and \([\text{Act}(q) \subseteq \text{dom}(\Xi) \text{ by Z4}] \) and \([\text{Bound}(q) \cap w1 = \emptyset \text{ by Z7}] \). Then, conclude \(\text{Assumptions})

by Lemma \[\text{Z6}\] \(Z8\).
Conclude:

\[ \text{split}(w, q \cdot r) \]

\[
\begin{align*}
\text{split}(w, q \cdot r) & \equiv \text{split}(w, q \cdot r), \\
\text{iso}(w, q \cdot r) & \equiv \text{iso}(w, q \cdot r), \\
\text{split}(w, q \cdot r) & \equiv \text{split}(w, q \cdot r) \\
\text{split}(w, q \cdot r) & \equiv \text{split}(w, q \cdot r).
\end{align*}
\]

\[ \square \]

E. Proofs for Section 5.4

**Proof (of Theorem 1).** Assumptions:

- \[ [a \in \text{TauFree and } \text{Act}(a) \subseteq \text{dom}(\Xi)] \ (Z1). \]

Observations:

- Conclude \( \emptyset \cap w = \emptyset \) by the definition of \( \cap \). Then, because \( \text{Bound}(\alpha) = \emptyset \) by the definition of \( \text{Bound} \), conclude \( \text{Bound}(\alpha) \cap w = \emptyset \) (Z2).

- Recall \( \alpha \in \text{Basic} \) by the definition of Basic and \( [[\alpha \in \text{TauFree and } \text{Act}(a) \subseteq \text{dom}(\Xi)] \] by Z1] and \( \text{Bound}(\alpha) \cap w = \emptyset \) by Z2]. Then, conclude \( \text{iso}(w, \alpha) \cap \text{iso}(w, \alpha) \cap \text{iso}(w, \alpha) \) by Lemma [1](Z3).

- Recall \( \alpha \in \text{TauFree} \) by Z1. Then, conclude

\[
\begin{align*}
\text{iso}(w, \alpha) & \simeq \bigcup_{i=1}^{n} a_i(d_i) \cup \xi_{\text{act}}(a_i)(w^\alpha) \cup \bigcup_{i=1}^{m'} \xi_{\text{act}}(a_i')(w^\alpha) \\
\text{iso}(w, \alpha) & \simeq \bigcup_{i=1}^{n} a_i(d_i) \cup \xi_{\text{act}}(a_i)(w^\alpha) \cup \bigcup_{i=1}^{m'} \xi_{\text{act}}(a_i')(w^\alpha) \quad \text{and } \alpha \simeq \bigcup_{i=1}^{n} a_i(d_i) \cup \bigcup_{i=1}^{m'} a_i'(d_i') \text{ and } \text{Act}(\alpha) = \bigcup_{i=1}^{n} a_i \cup \bigcup_{i=1}^{m'} a_i'
\end{align*}
\]

by Proposition [1](Z4).

- Recall \( \text{Act}(\alpha) \subseteq \text{dom}(\Xi) \) by Z1. Then, because \( \text{dom}(\Xi) = \{a \mid (w, a) \in \text{dom}(\xi) \cap \text{dom}(\bar{\xi})\} \) by the definition of \( \text{dom} \), conclude \( \text{Act}(\alpha) \subseteq \{a \mid (w, a) \in \text{dom}(\xi) \cap \text{dom}(\bar{\xi})\} \). Then, conclude \( \text{Act}(\alpha) \subseteq \{a \mid (w, a) \in (\text{dom}(\xi) \cap \text{dom}(\bar{\xi}))\} \) by ZFC. Then, because \( \text{dom}(\Xi) \subseteq \{1, 2\}^* \times A \) by Definition 5, conclude \( \text{Act}(\alpha) \subseteq \{a \mid (w, a) \in (\{1, 2\}^* \times A)\} \). Then, conclude \( \text{Act}(\alpha) \subseteq A \) by ZFC (Z5).

- Conclude \( \text{img}(\xi), \text{img}(\bar{\xi}) \subseteq \text{Act} \setminus \{\tau, \omega\} \) by Definition 3. Then, because \( \text{Act}(\alpha) \subseteq A \) by Z5, conclude \( \text{img}(\xi), \text{img}(\bar{\xi}) \subseteq \text{Act} \setminus \{\text{Act}(\alpha) \cup \{\tau, \omega\}\} \). Then, conclude \( \text{Act}(\alpha) \cap (\text{img}(\xi) \cup \text{img}(\bar{\xi})) = \emptyset \) by ZFC. Then, because \( \text{Act}(\alpha) = \bigcup_{i=1}^{n} a_i \cup \bigcup_{i=1}^{m'} a_i' \) by Z4, conclude \( (\bigcup_{i=1}^{n} a_i \cup \bigcup_{i=1}^{m'} a_i') \cap (\text{img}(\xi) \cup \text{img}(\bar{\xi})) = \emptyset \). Then, conclude \( [a_i, a_i' \notin \text{img}(\xi) \cup \text{img}(\bar{\xi}) \text{ for all } 1 \leq i \leq n \text{ and } 1 \leq i' \leq m'] \) by ZFC (Z6).

- Recall \( [a_i, a_i' \notin \text{img}(\xi) \cup \text{img}(\bar{\xi}) \text{ for all } 1 \leq i \leq n \text{ and } 1 \leq i' \leq m'] \) by Z6. Then, conclude

\[
\begin{align*}
\mathcal{C}(\xi, \omega) \cup \tau & = \text{tau} \cup (\xi, \omega) \in \text{dom}(\xi) \cap \text{dom}(\bar{\xi})(\xi, \omega) \\
\mathcal{C}(\xi, \omega) \cup \tau & = \bigcup_{i=1}^{n} a_i(d_i) \cup \xi_{\text{act}}(a_i)(w^\alpha) \cup \bigcup_{i=1}^{m'} \xi_{\text{act}}(a_i')(w^\alpha) \\
\mathcal{C}(\xi, \omega) \cup \tau & = \bigcup_{i=1}^{n} a_i(d_i) \cup \xi_{\text{act}}(a_i)(w^\alpha) \cup \bigcup_{i=1}^{m'} \xi_{\text{act}}(a_i')(w^\alpha) \\
\mathcal{C}(\xi, \omega) \cup \tau & \simeq \bigcup_{i=1}^{n} a_i(d_i) \cup \text{tau} \cup \bigcup_{i=1}^{m'} a_i(d_i') \cup \text{tau}
\end{align*}
\]

by the definition of \( \mathcal{C} \) (Z7).
• Recall $\text{Act}(\alpha) \subseteq A$ by Z5. Then, because $\tau \in \text{Act} \setminus A$ by Definition [3], conclude $\tau \notin \text{Act}(\alpha)$. Then, because $\text{Act}(\alpha) = \bigcup_{i=1}^{n} a_{i} \cup \bigcup_{i'=1}^{n'} a'_{i'}$ by Z4, conclude $\tau \notin \bigcup_{i=1}^{n} a_{i} \cup \bigcup_{i'=1}^{n'} a'_{i'}$. Then, conclude $[a_{i}, a'_{i'} \notin \{\tau\} \text{ for all } [1 \leq i \leq n \text{ and } 1 \leq i' \leq n']$ by ZFC. Then, conclude

$$[\mathcal{T}_{[\tau]}(a_{i}(d_{i})) = a_{i}(d_{i}) \text{ and } \mathcal{T}_{[\tau]}(a'_{i}(d'_{i})) = a'_{i}(d'_{i})]$$

for all $[1 \leq i \leq n \text{ and } 1 \leq i' \leq n']$ by H3 (Z8).

• Recall $[a_{i}, a'_{i'} \notin \text{img}(\xi) \cup \text{img}(\zeta)] \text{ for all } [1 \leq i \leq n \text{ and } 1 \leq i' \leq n']$ by Z6. Then, conclude

$$[\partial_{\text{img}(\xi) \cdot \text{img}(\zeta)}(a_{i}(d_{i})) = a_{i}(d_{i}) \text{ and } \partial_{\text{img}(\xi) \cdot \text{img}(\zeta)}(a'_{i}(d'_{i})) = a'_{i}(d'_{i})]$$

for all $[1 \leq i \leq n \text{ and } 1 \leq i' \leq n']$ by B2. Then, because $\text{img}(\Xi) = \text{img}(\xi) \cup \text{img}(\zeta)$ by the definition of img, conclude (Z9):

$$[\partial_{\text{img}(\Xi)}(a_{i}(d_{i})) = a_{i}(d_{i}) \text{ and } \partial_{\text{img}(\Xi)}(a'_{i}(d'_{i})) = a'_{i}(d'_{i})]$$

for all $[1 \leq i \leq n \text{ and } 1 \leq i' \leq n']$.

Conclude:
Proof (of Theorem 2). Assume:

- \( p \in \text{Basic} \) and \( p \in \text{TauFree} \) and \( \text{Act}(p) \subseteq \text{dom}(\Xi) \) and \( \text{Bound}(p) \cap w = \emptyset \) (Z1).

Proceed by induction on the structure of \( p \).

**Base:** \( p = \alpha \) or \( p = \delta \). Proceed by case distinction on the structure of \( p \).

**Case:** \( p = \alpha \). Recall \( p \in \text{TauFree} \) and \( \text{Act}(p) \subseteq \text{dom}(\Xi) \) by Z1. Then, because \( p = \alpha \) by the definition of this case, conclude \([\alpha \in \text{TauFree} \; \text{and} \; \text{Act}(\alpha) \subseteq \text{dom}(\Xi)]\). Then, conclude split(w , \( \alpha \)) by Theorem 1. Then, because \( p = \alpha \) by the definition of this case, conclude split(w , \( p \)) \( \simeq p \).

**Case:** \( p = \delta \). Conclude:

\[
\begin{align*}
\text{split}(w , p) & \equiv \text{split}(w , \delta) \\
\text{split} & \equiv \text{?}(\text{isol}(w , \delta) \parallel \overline{\text{isol}}(w , \delta)) \\
\text{isol} & \equiv ?(\delta \parallel \delta) \\
\text{LM2, S4} & \equiv ?(\delta +\delta \parallel \delta + \delta \parallel \delta) \\
\text{acts} & \equiv ?(\delta) \\
\text{Case} & \equiv \delta
\end{align*}
\]

**Step:** \( p = q + r \) or \( p = q \cdot r \) or \( p = c \rightarrow q \circ r \) or \( p = \sum_{x \in \{d_1, \ldots, d_k\}} q \). Assumptions:

- Induction hypothesis (IH):
  \[
  \left[ \begin{array}{l}
  \hat{p} \in \text{Basic} \; \text{and} \; \hat{p} \in \text{TauFree} \; \text{and} \\
  \text{Act}(\hat{p}) \subseteq \text{dom}(\Xi) \; \text{and} \; \text{Bound}(\hat{p}) \cap \hat{w} = \emptyset
  \end{array} \right]
  \text{implies split}(\hat{w}, \hat{p}) \simeq \hat{p}
  \]
  \text{for all} \; \hat{p} \in \{q, r\}

Proceed by case distinction on the structure of \( p \).

**Case:** \( p = q + r \) or \( p = q \cdot r \) or \( p = c \rightarrow q \circ r \). Observations:

- Recall \( p \in \text{Basic} \) by Z1. Then, because \( p = q + r \) or \( p = q \cdot r \) or \( p = c \rightarrow q \circ r \) by the definition of this case, conclude \([q + r \in \text{Basic} \; \text{or} \; q \cdot r \in \text{Basic} \; \text{or} \; c \rightarrow q \circ r \in \text{Basic}]\). Then, conclude \( q , r \in \text{Basic} \) by the definition of Basic \( (Z2) \).

- Recall \( p \in \text{TauFree} \) by Z1. Then, because \( p = q + r \) or \( p = q \cdot r \) or \( p = c \rightarrow q \circ r \) by the definition of this case, conclude \([q + r \in \text{TauFree} \; \text{or} \; q \cdot r \in \text{TauFree} \; \text{or} \; c \rightarrow q \circ r \in \text{TauFree}]\). Then, conclude \( q , r \in \text{TauFree} \) by the definition of TauFree \( (Z3) \).

- Recall \( \text{Act}(p) \subseteq \text{dom}(\Xi) \) by Z1. Then, because \( p = q + r \) or \( p = q \cdot r \) or \( p = c \rightarrow q \circ r \) by the definition of this case, conclude \( \text{Act}(q + r) \subseteq \text{dom}(\Xi) \; \text{or} \; \text{Act}(q \cdot r) \subseteq \text{dom}(\Xi) \; \text{or} \; \text{Act}(c \rightarrow q \circ r) \subseteq \text{dom}(\Xi) \)

Then, conclude \( \text{Act}(q) \cup \text{Act}(r) \subseteq \text{dom}(\Xi) \) by the definition of Act. Then, conclude \( \text{Act}(q) , \text{Act}(r) \subseteq \text{dom}(\Xi) \) by ZFC \( (Z4) \).
• Recall \( \text{Bound}(p) \cap w = \emptyset \) by Z1. Then, because \([p = q + r \text{ or } p = q \cdot r \text{ or } p = c \rightarrow q \circ r]\) by the definition of this case, conclude

\[
\text{Bound}(q + r) \cap w = \emptyset \text{ or } \text{Bound}(q \cdot r) \cap w = \emptyset
\]

\[
\text{or } \text{Bound}(c \rightarrow q \circ r) \cap w = \emptyset
\]

Then, conclude \( \text{Bound}(q) \cup \text{Bound}(r) \cap w = \emptyset \) by the definition of \( \text{Bound} \). Then, conclude \([\text{Bound}(q) \cap w = \emptyset \text{ and } \text{Bound}(r) \cap w = \emptyset]\) by ZFC (Z5).

• Conclude \( 1, 2 \notin \text{Var} \) by Definition 2. Then, because \( \text{Bound}(q), \text{Bound}(r) \subseteq \text{Var} \) by the definition of \( \text{Bound} \), conclude \( 1, 2 \notin \text{Bound}(q), \text{Bound}(r) \) (Z6).

• Recall \([1, 2 \notin \text{Bound}(q), \text{Bound}(r) \text{ by Z6} \text{ and } [\text{Bound}(q) \cap w = \emptyset \text{ and } \text{Bound}(r) \cap w = \emptyset]\) by Z10]. Then, conclude \([\text{Bound}(q) \cap w1 = \emptyset \text{ and } \text{Bound}(r) \cap w2 = \emptyset]\) by Proposition 16 (Z7).

• Recall \([q, r \in \text{Basic} \text{ by Z2} \text{ and } [q, r \in \text{TauFree} \text{ by Z3} \text{ and } [\text{Act}(q), \text{Act}(r) \subseteq \text{dom}(\Xi) \text{ by Z4} \text{ and } [\text{Bound}(q) \cap w1 = \emptyset \text{ and } \text{Bound}(r) \cap w2 = \emptyset]\] by Z7]. Then, conclude \([\text{split}(w1, q) \simeq q \text{ and } \text{split}(w2, r) \simeq r]\) by IH (Z8).

Proceed by case distinction on the structure of \( p \).

**Case**: \( p = q + r \). Observations:

• Recall

\[
p \in \text{Basic and } p \in \text{TauFree and}
\]

\[
\text{Act}(p) \subseteq \text{dom}(\Xi) \text{ and } \text{Bound}(p) \cap w = \emptyset
\]

by Z1. Then, because \( p = q + r \) by the definition of this case, conclude

\[
q + r \in \text{Basic and } q + r \in \text{TauFree and}
\]

\[
\text{Act}(q + r) \subseteq \text{dom}(\Xi) \text{ and } \text{Bound}(q + r) \cap w = \emptyset
\]

Then, conclude \( \text{split}(w, q + r) \simeq \text{split}(w1, q) + \text{split}(w2, r) \) by Lemma 2 (Z9).

Conclude:

\[
\text{split}(w, p)
\]

\[
\simeq
\]

\[
\text{split}(w, q + r)
\]

\[
\simeq
\]

\[
\text{split}(w1, q) + \text{split}(w2, r)
\]

\[
\simeq
\]

\[
q + r
\]

\[
\simeq
\]

\[
p
\]

**Case**: \( p = q \cdot r \). Observations:

• Recall

\[
p \in \text{Basic and } p \in \text{TauFree and}
\]

\[
\text{Act}(p) \subseteq \text{dom}(\Xi) \text{ and } \text{Bound}(p) \cap w = \emptyset
\]

by Z1. Then, because \( p = q \cdot r \) by the definition of this case, conclude

\[
q \cdot r \in \text{Basic and } q \cdot r \in \text{TauFree and}
\]

\[
\text{Act}(q \cdot r) \subseteq \text{dom}(\Xi) \text{ and } \text{Bound}(q \cdot r) \cap w = \emptyset
\]

Then, conclude \( \text{split}(w, q \cdot r) \simeq \text{split}(w1, q) \cdot \text{split}(w2, r) \) by Lemma 5 (Z10).

Conclude:

\[
\text{split}(w, p)
\]

\[
\simeq
\]

\[
\text{split}(w, q \cdot r)
\]

\[
\simeq
\]

\[
\text{split}(w1, q) \cdot \text{split}(w2, r)
\]

\[
\simeq
\]

\[
q \cdot r
\]

\[
\simeq
\]

\[
p
\]

86
Case: $p = c \rightarrow q \odot r$. Conclude:

\[
\begin{align*}
\text{split}(w, p) \\
\text{Case} & \quad \sim \quad \text{split}(w, c \rightarrow q \odot r) \\
& \quad \cong \quad c \rightarrow \text{split}(w1, q) \odot \text{split}(w2, r) \\
& \quad \cong \quad c \rightarrow q \odot r \\
\text{Case} & \quad \equiv \quad p
\end{align*}
\]

Then, conclude $\text{Act}(Z1)$. Then, because $\text{Act}(Z1)$, conclude $\text{Act}(Z12)$. Then, conclude $q \in \text{TauFree}$ by the definition of $\text{TauFree}$ (Z12).

Recall $p \in \text{Basic}$ by Z1. Then, because $p = \sum_{x \in \{d_1, \ldots, d_l\}} q$ by the definition of this case, conclude $\sum_{x \in \{d_1, \ldots, d_l\}} q \in \text{Basic}$. Then, conclude $q \in \text{Basic}$ by the definition of $\text{Basic}$ (Z11).

Recall $p \in \text{TauFree}$ by Z1. Then, because $p = \sum_{x \in \{d_1, \ldots, d_l\}} q$ by the definition of this case, conclude $\sum_{x \in \{d_1, \ldots, d_l\}} q \in \text{TauFree}$. Then, conclude $q \in \text{TauFree}$ by the definition of $\text{TauFree}$ (Z12).

Recall $\text{Act}(p) \subseteq \text{dom}(\Xi)$ by Z1. Then, because $p = \sum_{x \in \{d_1, \ldots, d_l\}} q$ by the definition of this case, conclude $\text{Act}(\sum_{x \in \{d_1, \ldots, d_l\}} q) \subseteq \text{dom}(\Xi)$. Then, conclude $\text{Act}(q) \subseteq \text{dom}(\Xi)$ by the definition of $\text{Act}$ (Z13).

Recall $\text{Bound}(p) \cap w = \emptyset$ by Z1. Then, because $p = \sum_{x \in \{d_1, \ldots, d_l\}} q$ by the definition of this case, conclude $\text{Bound}(\sum_{x \in \{d_1, \ldots, d_l\}} q) \cap w = \emptyset$. Then, conclude $\text{Bound}(q) \cup \{x\} \cap w = \emptyset$ by the definition of $\text{Bound}$. Then, conclude $\text{Bound}(q) \cap w = \emptyset$ by ZFC (Z14).

Recall $p \in \text{Basic}$ by Z1. Then, because $p = \sum_{x \in \{d_1, \ldots, d_l\}} q$ by the definition of this case, conclude $\sum_{x \in \{d_1, \ldots, d_l\}} q \in \text{Basic}$. Then, conclude $x \notin \text{Bound}(q)$ by the definition of $\text{Basic}$ (Z15).

Recall $[x \notin \text{Bound}(q)$ by Z15] and $[\text{Bound}(q) \cap w = \emptyset]$ by Z14. Then, conclude $\text{Bound}(q) \cap w = \emptyset$ by Proposition [10] (Z16).

Recall $[q \in \text{Basic}$ by Z11] and $[q \in \text{TauFree}$ by Z12] and $[\text{Act}(q) \subseteq \text{dom}(\Xi)$ by Z13] and $[\text{Bound}(q) \cap w \neq \emptyset]$ by Z16. Then, conclude $\text{split}(wx, q) \cong q$ by IH (Z17).

Recall $p \in \text{Basic}$ and $p \in \text{TauFree}$ and $\text{Act}(p) \subseteq \text{dom}(\Xi)$ and $\text{Bound}(p) \cap w = \emptyset$ by Z1. Then, because $p = \sum_{x \in \{d_1, \ldots, d_l\}} q$ by the definition of this case, conclude

\[
\begin{align*}
\sum_{x \in \{d_1, \ldots, d_l\}} q \in \text{Basic} \quad & \text{and} \quad \sum_{x \in \{d_1, \ldots, d_l\}} q \in \text{TauFree} \quad \text{and} \\
\text{Act}(\sum_{x \in \{d_1, \ldots, d_l\}} q) \subseteq \text{dom}(\Xi) \quad & \text{and} \quad \text{Bound}(\sum_{x \in \{d_1, \ldots, d_l\}} q) \cap w = \emptyset
\end{align*}
\]

Then, conclude $\text{split}(w, \sum_{x \in \{d_1, \ldots, d_l\}} q) \cong \sum_{x \in \{d_1, \ldots, d_l\}} \text{split}(wx, q)$ by Lemma [10] (Z18).

Conclude:

\[
\begin{align*}
\text{split}(w, p) \\
\text{Case} & \quad \cong \quad \text{split}(w, \sum_{x \in \{d_1, \ldots, d_l\}} q) \\
& \quad \cong \quad \sum_{x \in \{d_1, \ldots, d_l\}} \text{split}(wx, q) \\
& \quad \cong \quad \sum_{x \in \{d_1, \ldots, d_l\}} q \\
\text{Case} & \quad \equiv \quad p
\end{align*}
\]

□
Theorem 4 (Precorrectness theorem for process specifications).

\[
[p \in \text{TauFree and } Act(p) \subseteq \text{dom}(\Xi) \text{ and } \text{Ref}(p) \subseteq \{P_1, \ldots, P_k\}] \implies \\
split(\epsilon, p)[P_1^k(x) := \hat{P}^k(1, g(x))] \cdots [P_k^k(x) := \hat{P}^k(k, g(x))] \\
\simeq p[P_1(x) := \hat{P}^1(1, g(x))] \cdots [P_k(x) := \hat{P}^k(k, g(x))]
\]

Proof. Assumptions:

• \( [p \in \text{TauFree and } Act(p) \subseteq \text{dom}(\Xi) \text{ and } \text{Ref}(p) \subseteq \{P_1, \ldots, P_k\}] \) \((Z1)\).

Proceed by induction on the structure of \( p \).

Base: \( [p \in \text{Basic or } p = P] \). Proceed by case distinction on the structure of \( p \).

Case: \( p \in \text{Basic} \). Recall \( [p \in \text{TauFree and } Act(p) \subseteq \text{dom}(\Xi)] \) by \( Z1 \) and \( [\text{Bound}(p) \cap \epsilon = \emptyset \text{ by the definition of } \cap] \). Then, conclude \( \text{split}(\epsilon, p) \simeq p \) by Theorem 2. Then, conclude \( \text{split}(\epsilon, p) \simeq p[P_1(x) := \hat{P}^1(1, g(x))] \cdots [P_k(x) := \hat{P}^k(k, g(x))] \) by the definition of \( :\simeq \).

Case: \( p = Q(y) \). Observations:

• Recall \( \text{Ref}(p) \subseteq \{P_1, \ldots, P_k\} \). Then, because \( p = Q(y) \) by the definition of this case, conclude \( \text{Ref}(Q(y)) \subseteq \{P_1, \ldots, P_k\} \). Then, conclude \( \{Q\} \subseteq \{P_1, \ldots, P_k\} \) by the definition of \( \text{Ref} \). Then, conclude \( [Q = P, \text{ for some } 1 \leq i \leq k] \) by ZFC (Z2).

Conclude:

\[
\text{split}(\epsilon, p)[P_1^k(x) := \hat{P}^1(1, g(x))] \cdots [P_k^k(x) := \hat{P}^k(k, g(x))] \\
\text{split}(\epsilon, Q(y))[P_1^k(x) := \hat{P}^1(1, g(x))] \cdots [P_k^k(x) := \hat{P}^k(k, g(x))] \\
\text{split}(\epsilon, P_i(y))[P_1^k(x) := \hat{P}^1(1, g(x))] \cdots [P_k^k(x) := \hat{P}^k(k, g(x))] \\
\text{split}(\epsilon, \hat{P}_i(x))[P_1^k(x) := \hat{P}^1(1, g(x))] \cdots [P_k^k(x) := \hat{P}^k(k, g(x))] \\
\text{split}(\epsilon, \hat{P}_i(x))[P_1^k(x) := \hat{P}^1(1, g(x))] \cdots [P_k^k(x) := \hat{P}^k(k, g(x))] \\
\text{split}(\epsilon, \hat{P}_i(x))[P_1^k(x) := \hat{P}^1(1, g(x))] \cdots [P_k^k(x) := \hat{P}^k(k, g(x))]
\]

Step: \( [p = q \oplus r \text{ or } p = c \rightarrow q \circ r \text{ or } p = \sum_{d \in D} q \text{ or } p = f(q)] \). Assumptions:

• Induction hypothesis (IH):

\[
\left\llbracket \begin{array}{l}
\hat{p} \in \text{TauFree and } Act(\hat{p}) \subseteq \text{dom}(\Xi) \\
\text{and } \text{Ref}(\hat{p}) \subseteq \{\hat{P}_1, \ldots, \hat{P}_k\}
\end{array} \right\rrbracket \\
\text{implies} \\
\text{split}(\epsilon, \hat{p})[\hat{P}_1^k(x) := \hat{P}^1(1, \hat{g}(x))] \cdots [\hat{P}_k^k(x) := \hat{P}^k(\hat{k}, \hat{g}(x))] \\
\simeq \hat{p}[\hat{P}_1(x) := \hat{P}^1(1, \hat{g}(x))] \cdots [\hat{P}_k(x) := \hat{P}^k(\hat{k}, \hat{g}(x))]
\]

for all \( \hat{p} \in \{\hat{g}, \hat{r}\} \)

Observations:

• Recall \( p \in \text{TauFree} \) by \( Z1 \). Then, because \( [p = q \oplus r \text{ or } p = c \rightarrow q \circ r \text{ or } p = \sum_{d \in D} q \text{ or } p = f(q)] \) by the definition of this case, conclude

\[
q \oplus r \in \text{TauFree or } c \rightarrow q \circ r \in \text{TauFree} \\
or \sum_{d \in D} q \in \text{TauFree or } f(q) \in \text{TauFree}
\]

Then, conclude \( q, r \in \text{TauFree} \) by the definition of \( \text{TauFree} \) (Z3).
• Recall $\text{Act}(p) \subseteq \text{dom}(\Xi)$ by Z1. Then, because $[p = q \oplus r \text{ or } p = c \rightarrow q \circ r \text{ or } p = \sum_{x \in D} q \text{ or } p = f(q)]$ by the definition of this case, conclude

$$\text{Act}(q \oplus r) \subseteq \text{dom}(\Xi) \text{ or } \text{Act}(c \rightarrow q \circ r) \subseteq \text{dom}(\Xi)$$

or $\text{Act}(\sum_{x \in D} q) \subseteq \text{dom}(\Xi) \text{ or } \text{Act}(f(q)) \subseteq \text{dom}(\Xi)$

Then, conclude $\text{Act}(q) \cup \text{Act}(r) \subseteq \text{dom}(\Xi)$ by the definition of $\text{Act}$. Then, conclude $\text{Act}(q), \text{Act}(r) \subseteq \text{dom}(\Xi)$ by ZFC (Z4).

• Recall $\text{Ref}(p) \subseteq \{\hat{P}_1, \ldots, \hat{P}_k\}$ by Z1. Then, because $[p = q \oplus r \text{ or } p = c \rightarrow q \circ r \text{ or } p = \sum_{x \in D} q \text{ or } p = f(q)]$ by the definition of this case, conclude

$$\text{Ref}(q \oplus r) \subseteq \{\hat{P}_1, \ldots, \hat{P}_k\} \text{ or } \text{Ref}(c \rightarrow q \circ r) \subseteq \{\hat{P}_1, \ldots, \hat{P}_k\} \text{ or } \text{Ref}(\sum_{x \in D} q) \subseteq \{\hat{P}_1, \ldots, \hat{P}_k\}$$

or $\text{Ref}(f(q)) \subseteq \{\hat{P}_1, \ldots, \hat{P}_k\}$

Then, conclude $\text{Ref}(q) \cup \text{Ref}(r) \subseteq \{\hat{P}_1, \ldots, \hat{P}_k\}$ by the definition of $\text{Ref}$. Then, conclude $\text{Ref}(q), \text{Ref}(r) \subseteq \{\hat{P}_1, \ldots, \hat{P}_k\}$ by Z5.

• Recall $[q, r \in \text{Taufree} \text{ by Z3}]$ and $[\text{Act}(q), \text{Act}(r) \subseteq \text{dom}(\Xi) \text{ by Z4}]$ and $[\text{Ref}(q), \text{Ref}(r) \subseteq \{\hat{P}_1, \ldots, \hat{P}_k\} \text{ by Z5}]$. Then, conclude

$$\text{split}(\varepsilon, q)[P^1(x) := \tilde{P}^1(1, g(x))] \cdots [P^n(x) := \tilde{P}^1(k, g(x))]$$

$$\simeq q[\tilde{P}^1(1, g(x))] \cdots [\tilde{P}^1(k, g(x))]$$

and $\text{split}(\varepsilon, r)[P^1(x) := \tilde{P}^1(1, g(x))] \cdots [P^n(x) := \tilde{P}^1(k, g(x))]$,

$$\simeq r[\tilde{P}^1(1, g(x))] \cdots [\tilde{P}^1(k, g(x))]$$

by IH (Z6).

Proceed by case distinction on the structure of $p$.

Case: $p = q \oplus r$. Conclude:

$$\text{split}(\varepsilon, p)[P^1(x) := \tilde{P}^1(1, g(x))] \cdots [P^n(x) := \tilde{P}^1(k, g(x))]$$

$$\overset{\text{Case}}{=} \text{split}(\varepsilon, q \oplus r)[P^1(x) := \tilde{P}^1(1, g(x))] \cdots [P^n(x) := \tilde{P}^1(k, g(x))]$$

$$\overset{\text{split}}{=} \text{split}(\varepsilon, q) \oplus \text{split}(\varepsilon, r)$$

$$\overset{\text{Case}}{=} \text{split}(\varepsilon, q)[P^1(x) := \tilde{P}^1(1, g(x))] \cdots [P^n(x) := \tilde{P}^1(k, g(x))]$$

$$\overset{\text{split}}{=} (q \oplus r)[P^1(x) := \tilde{P}^1(1, g(x))] \cdots [P^n(x) := \tilde{P}^1(k, g(x))]$$

Case: $p = c \rightarrow q \circ r$. Conclude:

$$\text{split}(\varepsilon, p)[P^1(x) := \tilde{P}^1(1, g(x))] \cdots [P^n(x) := \tilde{P}^1(k, g(x))]$$

$$\overset{\text{Case}}{=} \text{split}(\varepsilon, c \rightarrow q \circ r)[P^1(x) := \tilde{P}^1(1, g(x))] \cdots [P^n(x) := \tilde{P}^1(k, g(x))]$$

$$\overset{\text{split}}{=} (c \rightarrow \text{split}(\varepsilon, q) \circ \text{split}(\varepsilon, r))$$

$$\overset{\text{Case}}{=} [P^1(x) := \tilde{P}^1(1, g(x))] \cdots [P^n(x) := \tilde{P}^1(k, g(x))]$$
Proof (of Theorem 3). Assumptions:

\[ \begin{align*}
  & P_1(x_1 : D_1) = p_1, P_1(x_1 : D_1) = \text{split}(\epsilon, p_1), \\
  & \vdots \\
  & P_k(x_k : D_k) = p_k, P_k(x_k : D_k) = \text{split}(\epsilon, p_k) \\
  & \text{and } p_1, \ldots, p_k \in \text{TauFree} \\
  & \text{and } \text{Act}(p_1), \ldots, \text{Act}(p_k) \subseteq \text{dom}(\Xi) \text{ and} \\
  & \text{Ref}(p_i) \subseteq \{P_1, \ldots, P_k\} \text{ for all } 1 \leq i \leq k
\end{align*} \] (Z1).

- \( h = \text{harmonizer}(D_1 \cup \cdots \cup D_k, D) \) (Z2).
- \( \bar{P}(y, x : N \times D) = y \approx 1 \rightarrow p_1[P_1(d) := \bar{P}(1, h(d))] \cdots [P_k(d) := \bar{P}(k, h(d))] \circ \delta \)
  \( \vdots \)
  \( y \approx k \rightarrow p_k[P_k(d) := \bar{P}(1, h(d))] \cdots [P_k(d) := \bar{P}(k, h(d))] \circ \delta \) (Z3).
Observations:

- Recall $[P_1(x_1 : D_1) = p_1, \ldots, P_k(x_k : D_k) = p_k$ by Z1] and $[P^t(y, x : \mathbb{N} \times D) = y \approx 1 \mapsto \text{split}(p_1)[P_1^t(d) := \tilde{P}^t(1, h(d)) \ldots [P_k^t(d) := \tilde{P}^t(k, h(d))] \circ$
  
  $\vdash y \approx k \mapsto \text{split}(p_k)[P_1^t(d) := \tilde{P}^t(1, h(d)) \ldots [P_k^t(d) := \tilde{P}^t(k, h(d))] \circ \delta$
  
  (Z4).
- $\Phi(Z) = y \approx 1 \mapsto p_1[P_1(d) := Z(1, h(d)) \ldots [P_k(d) := Z(k, h(d))] \circ$
  
  $\vdash y \approx k \mapsto p_k[P_1(d) := Z(1, h(d)) \ldots [P_k(d) := Z(k, h(d))] \circ \delta$
  
  (Z5).

Observations:

- Recall $[P_1(x_1 : D_1) = p_1, \ldots, P_k(x_k : D_k) = p_k$ by Z1] and $[
  \tilde{P}^t(y, x : \mathbb{N} \times D) = y \approx 1 \mapsto \text{split}(p_1)[P_1^t(d) := \tilde{P}^t(1, h(d)) \ldots [P_k^t(d) := \tilde{P}^t(k, h(d))] \circ$
  
  $\vdash y \approx k \mapsto \text{split}(p_k)[P_1^t(d) := \tilde{P}^t(1, h(d)) \ldots [P_k^t(d) := \tilde{P}^t(k, h(d))] \circ \delta$
  
  by Z3] and $[h = \text{harmonizer}(D_1 \cup \cdots \cup D_k, D)$ by Z2]. Then, conclude $[P_i \simeq \tilde{P}(i) \text{ for all } 1 \leq i \leq k]$ by Proposition [Z6].
- Recall $[P^t_1(x_1 : D_1) = \text{split}(\epsilon, p_1), \ldots, P^t_k(x_k : D_k) = \text{split}(\epsilon, p_k)$ by Z1] and $[
  \tilde{P}^t(y, x : \mathbb{N} \times D) = y \approx 1 \mapsto \text{split}(p_1)[P_1^t(d) := \tilde{P}^t(1, h(d)) \ldots [P_k^t(d) := \tilde{P}^t(k, h(d))] \circ$
  
  $\vdash y \approx k \mapsto \text{split}(p_k)[P_1^t(d) := \tilde{P}^t(1, h(d)) \ldots [P_k^t(d) := \tilde{P}^t(k, h(d))] \circ \delta$
  
  by Z3] and $[h = \text{harmonizer}(D_1 \cup \cdots \cup D_k, D)$ by Z2]. Then, conclude $[P^t_i \simeq \tilde{P}^t(i) \text{ for all } 1 \leq i \leq k]$ by Proposition [Z7].
- Recall
  
  $p_1, \ldots, p_k \in \text{TauFree and Act}(p_1), \ldots, \text{Act}(p_k) \subseteq \text{dom}(\Xi)$
  
  and $[\text{Ref}(p_i) \subseteq \{P_1, \ldots, P_k\}$ for all $1 \leq i \leq k]$ by Z1. Then, conclude
  
  $\text{split}(\epsilon, p_1)[P^t_1(d) := \tilde{P}^t(1, h(d)) \ldots [P^t_k(d) := \tilde{P}^t(k, h(d))] \simeq p_i[P_1(d) := \tilde{P}^t(1, h(d)) \ldots [P_k(d) := \tilde{P}^t(k, h(d)))]$ for all $1 \leq i \leq k$
  
  by Theorem [Z3] (Z8).
- Conclude (Z9):
  
  $\Phi(\tilde{P})$
  
  $\simeq y \approx 1 \mapsto p_1[P_1(d) := \tilde{P}(1, h(d)) \ldots [P_k(d) := \tilde{P}(k, h(d))] \circ$
  
  $\vdash y \approx k \mapsto p_k[P_1(d) := \tilde{P}(1, h(d)) \ldots [P_k(d) := \tilde{P}(k, h(d))] \circ \delta$
\[ \tilde{P} \]

- Conclude (Z10):

\[ \Phi(\tilde{P}) \]

\[ \equiv \]

\[ y \approx k \rightarrow \text{split}(\epsilon, p_k)[P_1^t(d) := \tilde{P}^t(1, h(d))] \cdots [P_k^t(d) := \tilde{P}^t(k, h(d))] \odot \delta \]

Recall \[ \Phi(\tilde{P}) \simeq \tilde{P} \text{ by Z8} \] and \[ \Phi(\tilde{P}^t) \simeq \tilde{P}^t \text{ by Z8} \]. Then, conclude \( \tilde{P} \simeq \tilde{P}^t \) by RSP (Z10).

Recall \[ P_i \simeq \tilde{P}(i) \text{ for all } 1 \leq i \leq k \] by Z6. Then, because \( \tilde{P} \simeq \tilde{P}^t \) by Z10, conclude \[ P_i \simeq \tilde{P}^t(i) \text{ for all } 1 \leq i \leq k \]. Then, because \[ P^t_i \simeq \tilde{P}^t(i) \text{ for all } 1 \leq i \leq k \] by Z7, conclude \[ P_i \simeq P^t_i \text{ for all } 1 \leq i \leq k \]. \( \Box \)