A Core Calculus for Equational Proofs of Distributed Cryptographic Protocols: Technical Report

Kristina Sojakova Mihai Codescu Joshua Gancher

November 28, 2024

Acknowledgement

This project was funded through the NGI Assure Fund, a fund established by NLnet with financial support from the European Commission's Next Generation Internet programme, under the aegis of DG Communications Networks, Content and Technology under grant agreement No. 957073.

1 Syntax of IPDL

IPDL is built from three layers: *protocols* are networks of mutually interacting *reactions*, which are simple monadic programs probabilistically computing an *expression*. In the context of a protocol, a reaction operates on a unique *channel* and may read from other channels, thereby utilizing computations coming from other reactions. The syntax and judgements of IPDL are outlined in Figures **??**, **??**, respectively, and are parameterized by a user-defined *signature* Σ:

Definition 1 (Signature). *An IPDL signature Σ is a finite collection of:*

- *type constants* t*;*
- *function symbols* $\mathsf{f} : \sigma \to \tau$*; and*
- *distribution symbols* $\mathsf{d} : \sigma \twoheadrightarrow \tau$*.*

We have a minimal set of data types, including the unit type 1, Booleans, products, as well as arbitrary type symbols t, drawn from the signature Σ. Expressions are used for non-probabilistic computations, and are standard. All values in IPDL are bitstrings of a length given by data types, so we annotate the operations $\mathsf{fst}_{\tau\times\sigma}$ and $\mathsf{snd}_{\tau\times\sigma}$ with the type of the pair to determine the index to split the pair into two; for readability we omit this subscript whenever appropriate. All function symbols f and distribution symbols d must be declared in the signature Σ. Substitutions $\theta : \Gamma_1 \to \Gamma_2$ between type contexts are standard.

As mentioned above, reactions are monadic programs which may return expressions, perform probabilistic sampling, read from channels, branch on a value of type Bool, and sequentially compose. Protocols in IPDL are given by a simple but expressive syntax: channel assignment $o := R$ assigns the reaction R to channel o; parallel composition $P \parallel Q$ allows P and Q to freely interact concurrently; and channel generation $\mathsf{new}\ o : \tau\ \mathsf{in}\ P$ creates a new, internal channel for use in P. *Embeddings* $\phi : \Delta_1 \to \Delta_2$ between channel contexts are injective, type-preserving mappings that specify how to rename channels in Δ_2 to fit in the larger context Δ_1.

Formally, references $\mathsf{var}(x : \tau)$ to variables and $\mathsf{read}(c : \tau)$ to channels include a typing annotation. This will come in handy later when we encode an IPDL construct as a sequence of symbols on a Turing Machine tape; knowing the type τ will allow us to allocate the correct number of bits for the variable x or the channel c. In almost all other instances, we simply write x and $\mathsf{read}\ c$. Similarly, we often write $\mathsf{f}\ e$ instead of $\mathsf{app}_{\sigma\to\tau}\ \mathsf{f}\ e$ and $\mathsf{samp}\ \mathsf{d}\ e$ instead of $\mathsf{samp}_{\sigma\twoheadrightarrow\tau}\ \mathsf{d}\ e$. For a constant $\mathsf{f} : 1 \to \tau$, we write f in place of $\mathsf{f}\ \checkmark$, and for a constant $\mathsf{d} : 1 \twoheadrightarrow \tau$, we write d instead of $\mathsf{d}\ \checkmark$. We also frequently omit the type of the bound variable in a sequential composition. Finally, we might omit the typing subscript in expressions $\mathsf{fst}_{\sigma\times\tau}$ and $\mathsf{snd}_{\sigma\times\tau}$ if the types can be inferred from the context or are irrelevant.

Variables	$x, y, z+$		
Channels	i, o, c		
Channel Sets	I, O	$::=$	$\{c_1, \ldots, c_n\}$
Data Types	τ, σ	$::=$	$\mathsf{t} \mid 1 \mid \mathsf{Bool} \mid \tau_1 \times \tau_2$
Expressions	e	$::=$	$\mathsf{var}(x : \tau) \mid \checkmark \mid \mathsf{true} \mid \mathsf{false} \mid \mathsf{app}_{\sigma \to \tau}\ \mathsf{f}\ e \mid (e_1, e_2) \mid \mathsf{fst}_{\sigma \times \tau}\ e \mid \mathsf{snd}_{\sigma \times \tau}\ e$
Reactions	R, S	$::=$	$\mathsf{ret}\ e \mid \mathsf{samp}_{\sigma \twoheadrightarrow \tau}\ \mathsf{d}\ e \mid \mathsf{read}(c : \tau) \mid \mathsf{if}\ e\ \mathsf{then}\ R_1\ \mathsf{else}\ R_2 \mid x : \sigma \leftarrow R;\ S$
Protocols	P, Q	$::=$	$0 \mid o := R \mid P \parallel Q \mid \mathsf{new}\ o : \tau\ \mathsf{in}\ P$
Type Contexts	Γ	$::=$	$\cdot \mid \Gamma, x : \tau$
Channel Contexts	Δ	$::=$	$\cdot \mid \Delta, c : \tau$

Figure 1: Syntax of IPDL.

Expression Typing	$\Gamma \vdash e : \tau$
Reaction Typing	$\Delta;\ \Gamma \vdash R : I \to \tau$
Protocol Typing	$\Delta \vdash P : I \to O$
Substitutions	$\theta : \Gamma_1 \to \Gamma_2$
Embeddings	$\phi : \Delta_1 \to \Delta_2$
Expression Equality	$\Gamma \vdash e_1 = e_2 : \tau$
Reaction Equality	$\Delta;\ \Gamma \vdash R_1 = R_2 : I \to \tau$
Protocol Equality (Strict)	$\Delta \vdash P_1 = P_2 : I \to O$

Figure 2: Judgements of the exact fragment of IPDL.

1.1 Typing

We restrict our attention to well-typed IPDL constructs. In addition to respecting data types, the typing judgments guarantee that all reads from channels in reactions are in scope, and that all channels are assigned at most one reaction in protocols. The typing $\Gamma \vdash e : \tau$ for expressions is standard, see Figure **??**. Figure **??** shows the typing rules for reactions. Intuitively, $\Delta;\ \Gamma \vdash R : I \to \tau$ holds when R uses variables in Γ, reads from channels in I typed according to Δ, and returns a value of type τ. Figure **??** gives the typing rules for protocols: $\Delta \vdash P : I \to O$ holds when P uses inputs in I to assign reactions to the channels in O, all typed according to Δ.

Channel assignment $o := R$ has the type $I \to \{o\}$ when R is well-typed with an empty variable context, making use of inputs from I as well as of o. We allow R to read from its own output o to express divergence: the protocol $o := \mathsf{read}\ o$ cannot reduce, which is useful for (conditionally) deactivating certain outputs. The typing rule for parallel composition $P \parallel Q$ states that P may use the outputs of Q as inputs while defining its own outputs, and vice versa. Importantly, the typing rules ensure that the outputs of P and Q are disjoint so that each channel carries a unique reaction. Finally, the rule for channel generation allows a protocol to select a fresh channel name o, assign it a type τ, and use it for internal computation and communication. Protocol typing plays a crucial role for modeling security. Simulation-based security in IPDL is modeled by the existence of a *simulator* with an appropriate typing judgment, $\Delta \vdash \mathsf{Sim} : I \to O$. Restricting the behavior of Sim to only use inputs along I is necessary to rule out trivial results (*e.g.*, Sim simply copies a secret from the specification).

1.2 Equational Logic

We now present the equational logic of IPDL. As mentioned above, the logic is divided into *exact* rules that establish semantic equality between protocols, and *approximate* rules that are used to discharge computational indistinguishability assumptions.

$$\boxed{\Gamma \vdash e : \tau}$$

$$\frac{x : \tau \in \Gamma}{\Gamma \vdash \mathsf{var}(x : \tau) : \tau} \qquad \frac{}{\Gamma \vdash \checkmark : \mathsf{1}} \qquad \frac{}{\Gamma \vdash \mathsf{true} : \mathsf{Bool}} \qquad \frac{}{\Gamma \vdash \mathsf{false} : \mathsf{Bool}} \qquad \frac{\mathsf{f} : \sigma \to \tau \in \Sigma \qquad \Gamma \vdash e : \sigma}{\Gamma \vdash \mathsf{app}_{\sigma \to \tau}\ \mathsf{f}\ e : \tau}$$

$$\frac{\Gamma \vdash e_1 : \tau_1 \qquad \Gamma \vdash e_2 : \tau_2}{\Gamma \vdash (e_1, e_2) : \tau_1 \times \tau_2} \qquad \frac{\Gamma \vdash e : \sigma \times \tau}{\Gamma \vdash \mathsf{fst}_{\sigma \times \tau}\ e : \sigma} \qquad \frac{\Gamma \vdash e : \sigma \times \tau}{\Gamma \vdash \mathsf{snd}_{\sigma \times \tau}\ e : \tau}$$

Figure 3: Typing for IPDL expressions.

$$\boxed{\Delta;\ \Gamma \vdash R : I \to \tau}$$

$$\frac{\Gamma \vdash e : \tau}{\Delta;\ \Gamma \vdash \mathsf{ret}\ e : I \to \tau} \qquad \frac{\mathsf{d} : \sigma \twoheadrightarrow \tau \in \Sigma \qquad \Gamma \vdash e : \sigma}{\Delta;\ \Gamma \vdash \mathsf{samp}_{\sigma \twoheadrightarrow \tau}\ \mathsf{d}\ e : I \to \tau} \qquad \frac{i : \tau \in \Delta \qquad i \in I}{\Delta;\ \Gamma \vdash \mathsf{read}(i : \tau) : I \to \tau}$$

$$\frac{\Gamma \vdash e : \mathsf{Bool} \qquad \Delta;\ \Gamma \vdash R_1 : I \to \tau \qquad \Delta;\ \Gamma \vdash R_2 : I \to \tau}{\Delta;\ \Gamma \vdash \mathsf{if}\ e\ \mathsf{then}\ R_1\ \mathsf{else}\ R_2 : I \to \tau} \qquad \frac{\Delta;\ \Gamma \vdash R : I \to \sigma \qquad \Delta;\ \Gamma, x : \sigma \vdash S : I \to \tau}{\Delta;\ \Gamma \vdash (x : \sigma \leftarrow R;\ S) : I \to \tau}$$

Figure 4: Typing for IPDL reactions.

$$\boxed{\Delta \vdash P : I \to O}$$

$$\frac{}{\Delta \vdash \mathsf{0} : I \to \varnothing} \qquad \frac{o \notin I \qquad o : \tau \in \Delta \qquad \Delta;\ \cdot \vdash R : I \cup \{o\} \to \tau}{\Delta \vdash \big(o := R\big) : I \to \{o\}}$$

$$\frac{\Delta \vdash P : I \cup O_2 \to O_1 \qquad \Delta \vdash Q : I \cup O_1 \to O_2}{\Delta \vdash P \parallel Q : I \to O_1 \cup O_2} \qquad \frac{\Delta, o : \tau \vdash P : I \to O \cup \{o\}}{\Delta \vdash \big(\mathsf{new}\ o : \tau\ \mathsf{in}\ P\big) : I \to O}$$

Figure 5: Typing for IPDL protocols.

1.2.1 Exact Equality

The majority of the reasoning in IPDL is done using exact equalities. At the expression level, we assume an ambient finite set $\mathbb{T}_{\mathsf{exp}}$ of axioms of the form $\Gamma \vdash e_1 = e_2 : \tau$, where $\Gamma \vdash e_1 : \tau$ and $\Gamma \vdash e_2 : \tau$. The rules for the equality of expressions are standard, see Figure **??**.

At the reaction level, we similarly assume an ambient finite set $\mathbb{T}_{\mathsf{dist}}$ of axioms of the form $\Gamma \vdash R_1 = R_2 : \tau$, where $\cdot\ ;\ \Gamma \vdash R_1 : \varnothing \to \tau$ and $\cdot\ ;\ \Gamma \vdash R_2 : \varnothing \to \tau$. The absence of any input channels means that the reactions R_1 and R_2 are in fact *distributions*, represented using monadic syntax. We will therefore refer to axioms of this form as distribution-level axioms. The rules for reaction equality, shown in Figures **??** and **??**, ensure in particular that reactions form a *commutative monad*: we have

$$\big(x \leftarrow R_1;\ y \leftarrow R_2;\ S(x,y)\big) = \big(y \leftarrow R_2;\ x \leftarrow R_1;\ S(x,y)\big)$$

whenever R_2 does not depend on x. All expected equalities for commutative monads hold for reactions, including the usual monad laws and congruence of equality under monadic bind. The SAMP-PURE rule allows us to drop an unused sampling, and the READ-DET rule allows us to replace two reads from the same channel by a single one. The rules IF-LEFT, IF-RIGHT, and IF-EXT allow us to manipulate conditionals.

At the protocol level, we again assume an ambient finite set $\mathbb{T}_{\mathsf{prot}}$ of axioms of the form $\Delta \vdash P_1 = P_2 : I \to O$, where $\Delta \vdash P_1 : I \to O$ and $\Delta \vdash P_2 : I \to O$. We use these axioms to specify user-defined functional assumptions, *e.g.*, the correctness of decryption. Exact protocol equalities allow us to reason about communication between subprotocols and functional correctness, and to simplify intermediate computations. We will see later that exact equality implies the existence of a *bisimulation* on protocols, which in turn implies perfect computational indistinguishability against an arbitrary distinguisher. The rules for the exact equality of protocols are in Figures **??**, **??**; we now describe them informally.

The COMP-NEW rule allows us to permute parallel composition and the creation of a new channel, and the same as *scope extrusion* in process calculi [**?**]. The ABSORB-LEFT rule allows us to discard a component in a parallel composition if it has no outputs; this allows us to eliminate internal channels once they are no longer used. The DIVERGE rule allows us to simplify diverging reactions: if a channel reads from itself and continues as an arbitrary reaction R, then we can safely discard R as we will never reach it in the first place. The three (un)folding rules FOLD-IF-LEFT, FOLD-IF-RIGHT, and FOLD-BIND allow us to simplify composite reactions by bringing their components into the protocol level as separate internal channels. The rule SUBSUME states that channel dependency is transitive: if we depend on o_1, and o_1 in turn depends on o_0, then we also depend on o_0, and this dependency can be made explicit. The SUBST rule allows us to inline certain reactions into read commands. Inlining $o_1 := R_1$ into $o_2 := x \leftarrow \mathsf{read}\ o_1;\ R_2$ is sound provided R_1 is *duplicable*: observing two independent results of evaluating R_1 is equivalent to observing the same result twice. This side condition is easily discharged whenever R_1 does not contain probabilistic sampling. Finally, the DROP rule allows dropping unused reads from channels in certain situations. Due to timing dependencies among channels, we only allow dropping reads from the channel $o_1 := R_1$ in the context of $o_2 := \_ \leftarrow \mathsf{read}\ o_1;\ R_2$ when we have that $(\_ \leftarrow R_1;\ R_2) = R_2$. This side condition is met in particular whenever all reads present in R_1 are also present in R_2.

1.2.2 Approximate Equality

The equational theory for the approximate fragment of IPDL consists of two layers: one for the *approximate equality* of protocols, and one for the *asymptotic equality* of protocol families as functions of the security parameter $\lambda \in \mathbb{N}$. Informally, if two protocol families are asymptotically equal, then any *resource-bounded* adversary cannot distinguish them with greater than negligible error. Analogously to exact protocol equality, for approximate equality we assume an ambient finite set of *approximate axioms* of the form $\Delta \vdash P \approx Q : I \to O$, where $\Delta \vdash P : I \to O$ and $\Delta \vdash Q : I \to O$. These axioms capture cryptographic assumptions on computational indistinguishability. The approximate equality of two such protocols has the form $\Delta \vdash P \approx Q : I \to O$ $\mathsf{wid}\ k\ \mathsf{len}\ l$, and we think of this proof as corresponding to a specific security parameter λ. In the asymptotic equality judgement, both parameters k, l become functions of the security parameter λ, and must be bounded by a polynomial in λ.

Figure **??** shows the rules for the approximate equality of IPDL protocols, where we chain together a sequence of strict equalities and *approximate congruence* transformations, see Figure **??**. The parameter $k \in \mathbb{N}$ counts the number of axiom invocations. Applying a single approximate axiom incurs $k = 1$ (rule APPROX-CONG, and the use of transitivity requires us to add up the respective values of k (rule TRANS). Even though each individual axiom

$$\boxed{\Gamma \vdash e_1 = e_2 : \tau}$$

$$\frac{\Gamma \vdash e : \tau}{\Gamma \vdash e = e : \tau}\ \textsc{Refl} \qquad \frac{\Gamma \vdash e_1 = e_2 : \tau}{\Gamma \vdash e_2 = e_1 : \tau}\ \textsc{Sym} \qquad \frac{\Gamma \vdash e_1 = e_2 : \tau \qquad \Gamma \vdash e_2 = e_3 : \tau}{\Gamma \vdash e_1 = e_3 : \tau}\ \textsc{Trans}$$

$$\frac{\Gamma \vdash e_1 = e_2 : \tau\ \mathsf{axiom}}{\Gamma \vdash e_1 = e_2 : \tau}\ \textsc{Axiom} \qquad \frac{\theta : \Gamma_1 \to \Gamma_2 \qquad \Gamma_2 \vdash e_1 = e_2 : \tau}{\Gamma_1 \vdash \theta^\star(e_1) = \theta^\star(e_2) : \tau}\ \textsc{Subst}$$

$$\frac{\mathsf{f} : \sigma \to \tau \in \Sigma \qquad \Gamma \vdash e = e' : \sigma}{\Gamma \vdash \mathsf{app}_{\sigma \to \tau}\ \mathsf{f}\ e = \mathsf{app}_{\sigma \to \tau}\ \mathsf{f}\ e' : \tau}\ \textsc{App-Cong} \qquad \frac{\Gamma \vdash e_1 = e_1' : \tau_1 \qquad \Gamma \vdash e_2 = e_2' : \tau_2}{\Gamma \vdash (e_1, e_2) = (e_1', e_2') : \tau_1 \times \tau_2}\ \textsc{Pair-Cong}$$

$$\frac{\Gamma \vdash e = e' : \sigma \times \tau}{\Gamma \vdash \mathsf{fst}_{\sigma \times \tau}\ e = \mathsf{fst}_{\sigma \times \tau}\ e' : \sigma}\ \textsc{Fst-Cong} \qquad \frac{\Gamma \vdash e = e' : \sigma \times \tau}{\Gamma \vdash \mathsf{snd}_{\sigma \times \tau}\ e = \mathsf{snd}_{\sigma \times \tau}\ e' : \tau}\ \textsc{Snd-Cong}$$

$$\frac{\Gamma \vdash e_1 : \tau_1 \qquad \Gamma \vdash e_2 : \tau_2}{\Gamma \vdash \mathsf{fst}_{\tau_1 \times \tau_2}\ (e_1, e_2) = e_1 : \tau_1}\ \textsc{Fst-Pair} \qquad \frac{\Gamma \vdash e_1 : \tau_1 \qquad \Gamma \vdash e_2 : \tau_2}{\Gamma \vdash \mathsf{snd}_{\tau_1 \times \tau_2}\ (e_1, e_2) = e_2 : \tau_2}\ \textsc{Snd-Pair}$$

$$\frac{\Gamma \vdash e : \sigma \times \tau}{\Gamma \vdash e = \big(\mathsf{fst}_{\sigma \times \tau}\ e,\ \mathsf{snd}_{\sigma \times \tau}\ e\big) : \sigma \times \tau}\ \textsc{Pair-Ext} \qquad \frac{\Gamma \vdash e : \mathsf{1}}{\Gamma \vdash e = \checkmark : \mathsf{1}}\ \textsc{Unit-Ext}$$

Figure 6: Equality for IPDL expressions.

$$\boxed{\Delta;\ \Gamma \vdash R_1 = R_2 : I \to \tau}$$

$$\frac{\Delta;\ \Gamma \vdash R : I \to \tau}{\Delta;\ \Gamma \vdash R = R : I \to \tau}\ \textsc{Refl} \qquad \frac{\Delta;\ \Gamma \vdash R_1 = R_2 : I \to \tau}{\Delta;\ \Gamma \vdash R_2 = R_1 : I \to \tau}\ \textsc{Sym}$$

$$\frac{\Delta;\ \Gamma \vdash R_1 = R_2 : I \to \tau \qquad \Delta;\ \Gamma \vdash R_2 = R_3 : I \to \tau}{\Delta;\ \Gamma \vdash R_1 = R_3 : I \to \tau}\ \textsc{Trans} \qquad \frac{\Gamma \vdash R_1 = R_2 : \tau\ \mathsf{axiom}}{\cdot\ ;\ \Gamma \vdash R_1 = R_2 : \varnothing \to \tau}\ \textsc{Axiom}$$

$$\frac{i \notin I \qquad \Delta;\ \Gamma \vdash R_1 = R_2 : I \to \tau}{\Delta;\ \Gamma \vdash R_1 = R_2 : I \cup \{i\} \to \tau}\ \textsc{Input-Unused} \qquad \frac{\phi : \Delta_1 \to \Delta_2 \qquad \Delta_2 \vdash R_1 = R_2 : I \to \tau}{\Delta_1 \vdash \phi^\star(R_1) = \phi^\star(R_2) : \phi^\star(I) \to \tau}\ \textsc{Embed}$$

$$\frac{\theta : \Gamma_1 \to \Gamma_2 \qquad \Delta;\ \Gamma_2 \vdash R_1 = R_2 : I \to \tau}{\Delta;\ \Gamma_1 \vdash \theta^\star(R_1) = \theta^\star(R_2) : I \to \tau}\ \textsc{Subst} \qquad \frac{\Gamma \vdash e = e' : \tau}{\Delta;\ \Gamma \vdash \mathsf{ret}\ e = \mathsf{ret}\ e' : I \to \tau}\ \textsc{Cong-Ret}$$

$$\frac{\mathsf{d} : \sigma \twoheadrightarrow \tau \in \Sigma \qquad \Gamma \vdash e = e' : \sigma}{\Delta;\ \Gamma \vdash \mathsf{samp}_{\sigma \twoheadrightarrow \tau}\ \mathsf{d}\ e = \mathsf{samp}_{\sigma \twoheadrightarrow \tau}\ \mathsf{d}\ e' : I \to \tau}\ \textsc{Cong-Samp}$$

$$\frac{\Gamma \vdash e = e' : \mathsf{Bool} \qquad \Delta;\ \Gamma \vdash R_1 = R_1' : I \to \tau \qquad \Delta;\ \Gamma \vdash R_2 = R_2' : I \to \tau}{\Delta;\ \Gamma \vdash \big(\mathsf{if}\ e\ \mathsf{then}\ R_1\ \mathsf{else}\ R_2\big) = \big(\mathsf{if}\ e'\ \mathsf{then}\ R_1'\ \mathsf{else}\ R_2'\big) : I \to \tau}\ \textsc{Cong-If}$$

$$\frac{\Delta;\ \Gamma \vdash R = R' : I \to \sigma \qquad \Delta;\ \Gamma, x : \sigma \vdash S = S' : I \to \tau}{\Delta;\ \Gamma \vdash (x : \sigma \leftarrow R;\ S) = (x : \sigma \leftarrow R';\ S') : I \to \tau}\ \textsc{Cong-Bind}$$

Figure 7: Equality for IPDL reactions. Additional rules are given in Figure **??**.

$$\boxed{\Delta;\ \Gamma \vdash R_1 = R_2 : I \to \tau}$$

$$\frac{\Gamma \vdash e : \sigma \qquad \Delta;\ \Gamma, x : \sigma \vdash R : I \to \tau}{\Delta;\ \Gamma \vdash (x : \sigma \leftarrow \mathsf{ret}\ e;\ R) = R[x := e] : I \to \tau}\ \textsc{ret-bind} \qquad \frac{\Delta;\ \Gamma \vdash R : I \to \tau}{\Delta;\ \Gamma \vdash (x : \tau \leftarrow R;\ \mathsf{ret}\ x) = R : I \to \tau}\ \textsc{bind-ret}$$

$$\frac{\Delta;\ \Gamma \vdash R_1 : I \to \sigma_1 \qquad \Delta;\ \Gamma, x_1 : \sigma_1 \vdash R_2 : I \to \sigma_2 \qquad \Delta;\ \Gamma, x_2 : \sigma_2 \vdash S : I \to \tau}{\Delta;\ \Gamma \vdash \big(x_2 : \sigma_2 \leftarrow (x_1 : \sigma_1 \leftarrow R_1;\ R_2);\ S\big) = \big(x_1 : \sigma_1 \leftarrow R_1;\ x_2 : \sigma_2 \leftarrow R_2;\ S\big) : I \to \tau}\ \textsc{bind-bind}$$

$$\frac{\Delta;\ \Gamma \vdash R_1 : I \to \sigma_1 \qquad \Delta;\ \Gamma \vdash R_2 : I \to \sigma_2 \qquad \Delta;\ \Gamma, x_1 : \sigma_1, x_2 : \sigma_2 \vdash S : I \to \tau}{\Delta;\ \Gamma \vdash \big(x_1 : \sigma_1 \leftarrow R_1;\ x_2 : \sigma_2 \leftarrow R_2;\ S\big) = \big(x_2 : \sigma_2 \leftarrow R_2;\ x_1 : \sigma_1 \leftarrow R_1;\ S\big) : I \to \tau}\ \textsc{exch}$$

$$\frac{\mathsf{d} : \rho \twoheadrightarrow \sigma \in \Sigma \qquad \Gamma \vdash e : \rho \qquad \Delta;\ \Gamma \vdash R : I \to \tau}{\Delta;\ \Gamma \vdash (x : \sigma \leftarrow \mathsf{samp}_{\rho \twoheadrightarrow \sigma}\ \mathsf{d}\ e;\ R) = R : I \to \tau}\ \textsc{samp-pure}$$

$$\frac{i : \sigma \in \Delta \qquad i \in I \qquad \Delta;\ \Gamma, x : \sigma, y : \sigma \vdash R : I \to \tau}{\Delta;\ \Gamma \vdash \big(x : \sigma \leftarrow \mathsf{read}\ i;\ y : \sigma \leftarrow \mathsf{read}\ i;\ R\big) = \big(x : \sigma \leftarrow \mathsf{read}\ i;\ R[y := x]\big) : I \to \tau}\ \textsc{read-det}$$

$$\frac{\Delta;\ \Gamma \vdash R_1 : I \to \tau \qquad \Delta;\ \Gamma \vdash R_2 : I \to \tau}{\Delta;\ \Gamma \vdash \big(\mathsf{if\ true\ then}\ R_1\ \mathsf{else}\ R_2\big) = R_1 : I \to \tau}\ \textsc{if-left}$$

$$\frac{\Delta;\ \Gamma \vdash R_1 : I \to \tau \qquad \Delta;\ \Gamma \vdash R_2 : I \to \tau}{\Delta;\ \Gamma \vdash \big(\mathsf{if\ false\ then}\ R_1\ \mathsf{else}\ R_2\big) = R_2 : I \to \tau}\ \textsc{if-right}$$

$$\frac{\Delta;\ \Gamma, x : \mathsf{Bool} \vdash R : I \to \tau \qquad \Gamma \vdash e : \mathsf{Bool}}{\Delta;\ \Gamma \vdash \big(\mathsf{if}\ e\ \mathsf{then}\ R[x := \mathsf{true}]\ \mathsf{else}\ R[x := \mathsf{false}]\big) = R[x := e] : I \to \tau}\ \textsc{if-ext}$$

Figure 8: Equality for IPDL reactions.

$$\boxed{\Delta \vdash P_1 = P_2 : I \to O}$$

$$\frac{\Delta \vdash P : I \to O}{\Delta \vdash P = P : I \to O}\ \textsc{refl} \qquad \frac{\Delta \vdash P_1 = P_2 : I \to O}{\Delta \vdash P_2 = P_1 : I \to O}\ \textsc{sym}$$

$$\frac{\Delta \vdash P_1 = P_2 : I \to O \qquad \Delta \vdash P_2 = P_3 : I \to O}{\Delta \vdash P_1 = P_3 : I \to O}\ \textsc{trans} \qquad \frac{\Delta \vdash P_1 = P_2 : I \to O\ \textsf{axiom}}{\Delta \vdash P_1 = P_2 : I \to O}\ \textsc{axiom}$$

$$\frac{i \notin I \cup O \qquad \Delta \vdash P_1 = P_2 : I \to O}{\Delta \vdash P_1 = P_2 : I \cup \{i\} \to O}\ \textsc{input-unused} \qquad \frac{\phi : \Delta_1 \to \Delta_2 \qquad \Delta_2 \vdash P_1 = P_2 : I \to O}{\Delta_1 \vdash \phi^\star(P_1) = \phi^\star(P_2) : \phi^\star(I) \to \phi^\star(O)}\ \textsc{embed}$$

$$\frac{o \notin I \qquad o : \tau \in \Delta \qquad \Delta;\ \cdot \vdash R = R' : I \cup \{o\} \to \tau}{\Delta \vdash \big(o := R\big) = \big(o := R'\big) : I \to \{o\}}\ \textsc{cong-react}$$

$$\frac{\Delta \vdash P = P' : I \cup O_2 \to O_1 \qquad \Delta \vdash Q : I \cup O_1 \to O_2}{\Delta \vdash P \parallel Q = P' \parallel Q : I \to O_1 \cup O_2}\ \textsc{cong-comp-left}$$

$$\frac{\Delta, o : \tau \vdash P = P' : I \to O \cup \{o\}}{\Delta \vdash \big(\textsf{new}\ o : \tau\ \textsf{in}\ P\big) = \big(\textsf{new}\ o : \tau\ \textsf{in}\ P'\big) : I \to O}\ \textsc{cong-new}$$

$$\frac{\Delta \vdash P_1 : I \cup O_2 \to O_1 \qquad \Delta \vdash P_2 : I \cup O_1 \to O_2}{\Delta \vdash P_1 \parallel P_2 = P_2 \parallel P_1 : I \to O_1 \cup O_2}\ \textsc{comp-comm}$$

$$\frac{\Delta \vdash P_1 : I \cup O_2 \cup O_3 \to O_1 \qquad \Delta \vdash P_2 : I \cup O_1 \cup O_3 \to O_2 \qquad \Delta \vdash P_3 : I \cup O_1 \cup O_2 \to O_3}{\Delta \vdash (P_1 \parallel P_2) \parallel P_3 = P_1 \parallel (P_2 \parallel P_3) : I \to O_1 \cup O_2 \cup O_3}\ \textsc{comp-assoc}$$

$$\frac{\Delta, o_1 : \tau_1, o_2 : \tau_2 \vdash P : I \to O \cup \{o_1, o_2\}}{\Delta \vdash \big(\textsf{new}\ o_1 : \tau_1\ \textsf{in new}\ o_2 : \tau_2\ \textsf{in}\ P\big) = \big(\textsf{new}\ o_2 : \tau_2\ \textsf{in new}\ o_1 : \tau_1\ \textsf{in}\ P\big) : I \to O}\ \textsc{new-exch}$$

$$\frac{\Delta \vdash P : I \cup O_2 \to O_1 \qquad \Delta, o : \tau \vdash Q : I \cup O_1 \to O_2 \cup \{o\}}{\Delta \vdash P \parallel \big(\textsf{new}\ o : \tau\ \textsf{in}\ Q\big) = \textsf{new}\ o : \tau\ \textsf{in}\ (P \parallel Q) : I \to O_1 \cup O_2}\ \textsc{comp-new}$$

$$\frac{\Delta \vdash P : I \to O \qquad \Delta \vdash Q : I \cup O \to \varnothing}{\Delta \vdash P \parallel Q = P : I \to O}\ \textsc{absorb-left}$$

Figure 9: Exact equality for IPDL protocols. Additional rules are given in Figure ??.

$$\boxed{\Delta \vdash P = Q : I \to O}$$

$$\frac{o : \tau \in \Delta \qquad \Delta;\ \cdot \vdash R : I \to \tau}{\Delta \vdash \big(o := x \leftarrow \mathsf{read}\ o;\ R\big) = \big(o := \mathsf{read}\ o\big) : I \setminus \{o\} \to \{o\}}\ \textsc{Diverge}$$

$$\frac{o : \tau \in \Delta \qquad \Delta;\ \cdot \vdash R : I \to \mathsf{Bool} \qquad \Delta;\ \cdot \vdash S_1 : I \to \tau \qquad \Delta;\ \cdot \vdash S_2 : I \to \tau}{\begin{array}{r}\Delta \vdash \big(\mathsf{new}\ l : \tau\ \mathsf{in}\ o := x \leftarrow R;\ \mathsf{if}\ x\ \mathsf{then}\ \mathsf{read}\ l\ \mathsf{else}\ S_2 \parallel l := S_1\big) = \\ \big(o := x \leftarrow R;\ \mathsf{if}\ x\ \mathsf{then}\ S_1\ \mathsf{else}\ S_2\big) : I \setminus \{o\} \to \{o\}\end{array}}\ \textsc{Fold-If-Left}$$

$$\frac{o : \tau \in \Delta \qquad \Delta;\ \cdot \vdash R : I \to \mathsf{Bool} \qquad \Delta;\ \cdot \vdash S_1 : I \to \tau \qquad \Delta;\ \cdot \vdash S_2 : I \to \tau}{\begin{array}{r}\Delta \vdash \big(\mathsf{new}\ r : \tau\ \mathsf{in}\ o := x \leftarrow R;\ \mathsf{if}\ x\ \mathsf{then}\ S_1\ \mathsf{else}\ \mathsf{read}\ r \parallel r := S_2\big) = \\ \big(o := x \leftarrow R;\ \mathsf{if}\ x\ \mathsf{then}\ S_1\ \mathsf{else}\ S_2\big) : I \setminus \{o\} \to \{o\}\end{array}}\ \textsc{Fold-If-Right}$$

$$\frac{o : \tau \in \Delta \qquad \Delta;\ \cdot \vdash R : I \to \sigma \qquad \Delta;\ x : \sigma \vdash S : I \to \tau}{\Delta \vdash \big(\mathsf{new}\ c : \sigma\ \mathsf{in}\ o := x \leftarrow \mathsf{read}\ c;\ S \parallel c := R\big) = \big(o := x \leftarrow R;\ S\big) : I \setminus \{o\} \to \{o\}}\ \textsc{Fold-Bind}$$

$$\frac{\begin{array}{c}o_1 \neq o_2 \qquad o_1 : \tau_1, o_2 : \tau_2 \in \Delta \qquad \Delta;\ \cdot \vdash R_1 : I \to \tau_1 \qquad \Delta;\ x_1 : \tau_1 \vdash R_2 : I \to \tau_2 \\ \Delta;\ \cdot \vdash \big(x_1 \leftarrow R_1;\ x_1' \leftarrow R_1;\ \mathsf{ret}\ (x_1, x_1')\big) = \big(x_1 \leftarrow R_1;\ \mathsf{ret}\ (x_1, x_1)\big) : I \to \tau_1 \times \tau_1\end{array}}{\Delta \vdash \big(o_1 := R_1 \parallel o_2 := x_1 \leftarrow \mathsf{read}\ o_1;\ R_2\big) = \big(o_1 := R_1 \parallel o_2 := x_1 \leftarrow R_1;\ R_2\big) : I \setminus \{o_1, o_2\} \to \{o_1, o_2\}}\ \textsc{Subst}$$

$$\frac{\begin{array}{c}o_1 \neq o_2 \qquad o_1 : \tau_1, o_2 : \tau_2 \in \Delta \\ \Delta;\ \cdot \vdash R_1 : I \to \tau_1 \qquad \Delta;\ \cdot \vdash R_2 : I \to \tau_2 \qquad \Delta;\ \cdot \vdash \big(x_1 \leftarrow R_1;\ R_2\big) = R_2 : I \to \tau_2\end{array}}{\Delta \vdash \big(o_1 := R_1 \parallel o_2 := x_1 \leftarrow \mathsf{read}\ o_1;\ R_2\big) = \big(o_1 := R_1 \parallel o_2 := R_2\big) : I \setminus \{o_1, o_2\} \to \{o_1, o_2\}}\ \textsc{Drop}$$

Figure 10: Additional rules for exact equality of IPDL protocols. Distinguishing changes of equalities are highlighed in red.

$$\boxed{\Delta \vdash P \cong Q : I \to O\ \mathsf{len}\ l}$$

$$\frac{\Delta \vdash P \approx Q : I \to O\ \mathsf{axiom}}{\Delta \vdash P \cong Q : I \to O\ \mathsf{len}\ 0}\ \textsc{Axiom} \qquad \frac{i \notin I \cup O \qquad \Delta \vdash P \cong Q : I \to O\ \mathsf{len}\ l}{\Delta \vdash P \cong Q : I \cup \{i\} \to O\ \mathsf{len}\ l}\ \textsc{Input-Unused}$$

$$\frac{\phi : \Delta_1 \to \Delta_2 \qquad \Delta_2 \vdash P \cong Q : I \to O\ \mathsf{len}\ l}{\Delta_1 \vdash \phi^\star(P) \cong \phi^\star(Q) : \phi^\star(I) \to \phi^\star(O)\ \mathsf{len}\ l}\ \textsc{Embed}$$

$$\frac{\Delta \vdash P \cong P' : I \cup O_2 \to O_1\ \mathsf{len}\ l \qquad \Delta \vdash Q : I \cup O_1 \to O_2}{\Delta \vdash P \parallel Q \cong P' \parallel Q : I \to O_1 \cup O_2\ \mathsf{len}\ l + \|Q\|_{\mathsf{TM}} + 3}\ \textsc{Cong-Comp-Left}$$

$$\frac{\Delta, o : \tau \vdash P \cong P' : I \to O \cup \{o\}\ \mathsf{len}\ l}{\Delta \vdash \big(\mathsf{new}\ o : \tau\ \mathsf{in}\ P\big) \cong \big(\mathsf{new}\ o : \tau\ \mathsf{in}\ P'\big) : I \to O\ \mathsf{len}\ l}\ \textsc{Cong-New}$$

Figure 11: Approximate congruence of IPDL protocols.

$$\boxed{\Delta \vdash P \approx Q : I \to O \text{ wid } k \text{ len } l}$$

$$\frac{\Delta \vdash P = Q : I \to O}{\Delta \vdash P \approx Q : I \to O \text{ wid } 0 \text{ len } 0}\ \textsc{strict} \qquad \frac{\Delta \vdash P \cong Q : I \to O \text{ len } l}{\Delta \vdash P \approx Q : I \to O \text{ wid } 1 \text{ len } l}\ \textsc{approx-cong}$$

$$\frac{\Delta \vdash P_1 \approx P_2 : I \to O \text{ wid } k \text{ len } l}{\Delta \vdash P_2 \approx P_1 : I \to O \text{ wid } k \text{ len } l}\ \textsc{sym}$$

$$\frac{\Delta \vdash P_1 \approx P_2 : I \to O \text{ wid } k_1 \text{ len } l_1 \qquad \Delta \vdash P_2 \approx P_3 : I \to O \text{ wid } k_2 \text{ len } l_2}{\Delta \vdash P_1 \approx P_3 : I \to O \text{ wid } k_1 + k_2 \text{ len } \mathsf{max}(l_1, l_2)}\ \textsc{trans}$$

Figure 12: Approximate equality for IPDL protocols.

invocation introduces a negligible error, the sum of exponentially many negligible errors might not be negligible, which is why we later impose a polynomial bound on k.

The parameter l tracks the increase in adversarial resources incurred by the proof. The bulk of the reasoning in IPDL is done in the exact fragment, where a typical proof step transforms the protocol into a form where an approximate axiom applies. We subsequently carry out an approximate congruence step, where we use the approximate axiom to replace a small protocol fragment nested inside a larger program context by its computationally indistinguishable counterpart. The program context is formally a part of the adversary, and as such it must be resource-bounded for the indistinguishability assumption to apply. Some nesting patterns do not effect any change on the adversary's resources: for example, a simple renaming of channels (rule EMBED); the formal addition of an unused channel i to the protocol's inputs I (rule INPUT-UNUSED), in which case any value assigned by the adversary to channel i will leave the protocol unchanged; or the introduction of an internal channel $o : \tau$ (rule CONG-NEW), in which case the adversary will never query o because internal channels are only visible in the scope of their declaration.

On the other hand, composing two approximately equal protocols $P \approx P'$ with another protocol Q requires the adversary to simulate the interaction of the common protocol Q with P versus P'. In other words, the adversary *absorbs* Q and the protocol becomes part of the new adversary's code. In particular, the number of symbols needed to encode the adversary's code on a Turing Machine tape increases, and the parameter l measures this increase. As we can see in rule CONG-COMP-LEFT, we use $\|Q\|_{\mathsf{TM}} + 3$ additional symbols: $\|Q\|_{\mathsf{TM}}$ symbols for encoding the protocol Q; a parallel composition symbol to combine the original adversary code with the protocol Q; and two parenthesis symbols "(", ")" for enclosing the composition. We emphasize that the exact numbers here are not crucial; what matters is that we eventually deliver a polynomial in λ.

List the type constants declared in the signature Σ as $\mathsf{t}_1, \ldots, \mathsf{t}_{|\Sigma_\mathsf{t}|}$. Unlike the parameter $k \in \mathbb{N}$, the parameter l is not a natural number but a function $l(t_1, \ldots, t_{|\Sigma_\mathsf{t}|}) : \mathbb{N}^{|\Sigma_\mathsf{t}|} \to \mathbb{N}$ that is *monotonically increasing in each argument.* When encoding a protocol Q as a sequence of symbols on a Turing Machine tape, we invariably encounter type annotations such as $\mathsf{var}(x : \tau)$. At this point, we do not know how many bits we will need to encode values of type τ, because the type constants $\mathsf{t} \in \Sigma$ are as of yet uninterpreted. Instead, we leave the size of each type constant as a variable to the function l, which will later be instantiated by the appropriate natural number according to $[\![-]\!]$. With this proviso, the Turing Machine bound of a type τ is straightforward:

$$\begin{aligned} \|\mathsf{t}_i\|_{\mathsf{TM}} &:= t_i \\ \|1\|_{\mathsf{TM}} &:= 0 \\ \|\mathsf{Bool}\|_{\mathsf{TM}} &:= 1 \\ \|\tau_1 \times \tau_2\|_{\mathsf{TM}} &:= \|\tau_1\|_{\mathsf{TM}} + \|\tau_2\|_{\mathsf{TM}} \end{aligned}$$

For variables $\mathsf{var}(x : \tau)$, we use the symbols "(", "var", ":", ")" in addition to the de Bruijn index of the variable x, encoded as a single symbol, and the encoding of the type annotation τ. For expressions ✓, true, false, we use the corresponding symbols "✓", "true", "false" and the two parenthesis symbols "(", ")". For an application $\mathsf{app}_{\sigma \to \tau}\ \mathsf{f}\ e$, we use the symbols "(", "app", "→", ")" in addition to the function symbol f, encoded as a single symbol, and the encodings of the two type annotations σ, τ and the expression e. To encode a pair (e_1, e_2), we will only need the encodings of the two expressions e_1 and e_2. Finally, to encode first and second projections, we will use the symbols “(”, “fst” or “snd”, “×”,)” in addition to the encodings of the two type annotations σ, τ and the expression e.

$$\begin{aligned}
\|\mathsf{var}(x : \tau)\|_{\mathsf{TM}} &:= \|\tau\|_{\mathsf{TM}} + 5 \\
\|\checkmark\|_{\mathsf{TM}} &:= 3 \\
\|\mathsf{true}\|_{\mathsf{TM}} &:= 3 \\
\|\mathsf{false}\|_{\mathsf{TM}} &:= 3 \\
\|\mathsf{app}_{\sigma \to \tau}\ \mathsf{f}\ e\|_{\mathsf{TM}} &:= \|\sigma\|_{\mathsf{TM}} + \|\tau\|_{\mathsf{TM}} + \|e\|_{\mathsf{TM}} + 5 \\
\|(e_1, e_2)\|_{\mathsf{TM}} &:= \|e_1\|_{\mathsf{TM}} + \|e_2\|_{\mathsf{TM}} \\
\|\mathsf{fst}_{\sigma \times \tau}\ e\|_{\mathsf{TM}} &:= \|\sigma\|_{\mathsf{TM}} + \|\tau\|_{\mathsf{TM}} + \|e\|_{\mathsf{TM}} + 5 \\
\|\mathsf{snd}_{\sigma \times \tau}\ e\|_{\mathsf{TM}} &:= \|\sigma\|_{\mathsf{TM}} + \|\tau\|_{\mathsf{TM}} + \|e\|_{\mathsf{TM}} + 5
\end{aligned}$$

For a return $\mathsf{ret}\ e$, we use the symbols “(”, “ret”,)” in addition to the encoding of the expression e. For a sampling $\mathsf{samp}_{\sigma \twoheadrightarrow \tau}\ \mathsf{d}\ e$, we use the symbols “(”, “samp”, “↠”, “)” in addition to the distribution symbol d, encoded as a single symbol, and the encodings of the two type annotations σ, τ and the expression e. For a read $\mathsf{read}(c : \tau)$, we use the symbols “(”, “read”, “:”, “)” in addition to the de Bruijn index of the channel c, encoded as a single symbol, and the encoding of the type annotation τ. Furthermore, we will need one extra symbol: one of “input-to-query”, “input-queried”, “input-not-to-query”. When encoding a protocol $Q : I \cup O_1 \to O_2$ coming from the COMP-CONG-LEFT rule, we use “input-to-query” or “input-queried” if we are reading from a channel $o_1 \in O_1$, according to whether we have already queried the channel o_1, and “input-not-to-query” otherwise. For a conditional $\mathsf{if}\ e\ \mathsf{then}\ R_1\ \mathsf{else}\ R_2$, we use the symbols “(”, “if”, “then”, “else”, “)” in addition to the encodings of the expression e and the two reactions R_1, R_2. Finally, to encode a bind, we use the symbols “{”, “_”, “:”, “←”, “;”, “}” in addition to the encodings of the type annotation σ and the two reactions R and S. The symbol “_” is used in lieu of the bound variable name x and stands for de Bruijn index 0.

$$\begin{aligned}
\|\mathsf{ret}\ e\|_{\mathsf{TM}} &:= \|e\|_{\mathsf{TM}} + 3 \\
\|\mathsf{samp}_{\sigma \twoheadrightarrow \tau}\ \mathsf{d}\ e\|_{\mathsf{TM}} &:= \|\sigma\|_{\mathsf{TM}} + \|\tau\|_{\mathsf{TM}} + \|e\|_{\mathsf{TM}} + 5 \\
\|\mathsf{read}(c : \tau)\|_{\mathsf{TM}} &:= \|\tau\|_{\mathsf{TM}} + 6 \\
\|\mathsf{if}\ e\ \mathsf{then}\ R_1\ \mathsf{else}\ R_2\|_{\mathsf{TM}} &:= \|e\|_{\mathsf{TM}} + \|R_1\|_{\mathsf{TM}} + \|R_2\|_{\mathsf{TM}} + 5 \\
\|x : \sigma \leftarrow R;\ S\|_{\mathsf{TM}} &:= \|\sigma\|_{\mathsf{TM}} + \|R\|_{\mathsf{TM}} + \|S\|_{\mathsf{TM}} + 6
\end{aligned}$$

To encode the zero protocol $\mathsf{0}$, we use the single symbol “0”. For an assignment $o := R$, we use the symbols “[”, “:=”, “react”, “]” in addition to the de Bruijn index of the channel c, encoded as a single symbol, and the encoding of the reaction R. For a parallel composition $P \parallel Q$, we use the symbols “(”, “‖”, “)” in addition to the encodings of the two protocols P and Q. Finally, for the declaration of a new channel $\mathsf{new}\ o : \tau\ \mathsf{in}\ P$, we use the symbols “new”, “_”, “:”, “in”, “wen” in addition to the encodings of the typing annotation τ and the protocol P. The symbol “_” is used in lieu of the bound channel name c and stands for de Bruijn index 0. The symbol “wen” indicates the end of the binding scope.

$$\begin{aligned}
\|\mathsf{0}\|_{\mathsf{TM}} &:= 1 \\
\|o := R\|_{\mathsf{TM}} &:= \|R\|_{\mathsf{TM}} + 5 \\
\|P \parallel Q\|_{\mathsf{TM}} &:= \|P\|_{\mathsf{TM}} + \|Q\|_{\mathsf{TM}} + 3 \\
\|\mathsf{new}\ c : \tau\ \mathsf{in}\ P\|_{\mathsf{TM}} &:= \|\tau\|_{\mathsf{TM}} + \|P\|_{\mathsf{TM}} + 5
\end{aligned}$$

We note that the Turing Machine bound of each construct is invariant under embeddings.

To make the ambient approximate theory with axioms $\Delta^1 \vdash P^1 \approx Q^1 : I^1 \to O^1, \ldots, \Delta^n \vdash P^n \approx Q^n : I^n \to O^n$ explicit, we write the approximate equality judgement as

$$\Delta^1 \vdash P^1 \approx Q^1 : I^1 \to O^1, \ldots, \Delta^n \vdash P^n \approx Q^n : I^n \to O^n \Rightarrow \Delta \vdash P \approx Q : I \to O\ \mathsf{wid}\ k\ \mathsf{len}\ l.$$

We also recall that the exact fragment of IPDL is formulated with respect to ambient theories $\mathbb{T}_{\mathsf{exp}}$, $\mathbb{T}_{\mathsf{dist}}$, and $\mathbb{T}_{\mathsf{prot}}$ for expressions, distributions, and protocols. If we want to make these explicit, we combine them into a single exact

$$\boxed{\left\{\Delta^1_\lambda \vdash P^1_\lambda \approx Q^1_\lambda : I^1_\lambda \to O^1_\lambda\right\}_{\lambda\in\mathbb{N}}, \dots, \left\{\Delta^n_\lambda \vdash P^n_\lambda \approx Q^n_\lambda : I^n_\lambda \to O^n_\lambda\right\}_{\lambda\in\mathbb{N}} \Rrightarrow \left\{\Delta_\lambda \vdash P_\lambda \approx Q_\lambda : I_\lambda \to O_\lambda\right\}_{\lambda\in\mathbb{N}}}$$

$$\frac{\begin{array}{c}\forall\lambda, \Delta^1_\lambda \vdash P^1_\lambda \approx Q^1_\lambda : I^1_\lambda \to O^1_\lambda, \dots, \Delta^n_\lambda \vdash P^n_\lambda \approx Q^n_\lambda : I^n_\lambda \to O^n_\lambda \Rrightarrow \Delta_\lambda \vdash P_\lambda \approx Q_\lambda : I_\lambda \to O_\lambda \ \mathsf{wid}\ k_\lambda \ \mathsf{len}\ l_\lambda \\ k_\lambda = \mathsf{O}(\mathsf{poly}(\lambda)) \qquad l_\lambda = \mathsf{O}\big(\mathsf{poly}(\lambda, t_1, \dots, t_{|\Sigma_\mathsf{t}|})\big)\end{array}}{\left\{\Delta^1_\lambda \vdash P^1_\lambda \approx Q^1_\lambda : I^1_\lambda \to O^1_\lambda\right\}_{\lambda\in\mathbb{N}}, \dots, \left\{\Delta^n_\lambda \vdash P^n_\lambda \approx Q^n_\lambda : I^n_\lambda \to O^n_\lambda\right\}_{\lambda\in\mathbb{N}} \Rrightarrow \left\{\Delta_\lambda \vdash P_\lambda \approx Q_\lambda : I_\lambda \to O_\lambda\right\}_{\lambda\in\mathbb{N}}}$$

Figure 13: Asymptotic equality for IPDL protocol families.

IPDL theory $\mathbb{T}_= := (\mathbb{T}_{\mathsf{exp}}, \mathbb{T}_{\mathsf{dist}}, \mathbb{T}_{\mathsf{prot}})$, and write the approximate equality judgement as

$$\mathbb{T}_=;\, \Delta^1 \vdash P^1 \approx Q^1 : I^1 \to O^1, \dots, \Delta^n \vdash P^n \approx Q^n : I^n \to O^n \Rrightarrow \Delta \vdash P \approx Q : I \to O \ \mathsf{wid}\ k \ \mathsf{len}\ l.$$

For the asymptotic equality of IPDL protocols, we assume a finite set $\mathbb{T}_\approx$ of *approximate axiom families* of the form $\left\{\Delta_\lambda \vdash P_\lambda \approx Q_\lambda : I_\lambda \to O_\lambda\right\}_{\lambda\in\mathbb{N}}$, where $\left\{\Delta_\lambda \vdash P_\lambda : I_\lambda \to O_\lambda\right\}_{\lambda\in\mathbb{N}}$ and $\left\{\Delta_\lambda \vdash Q_\lambda : I_\lambda \to O_\lambda\right\}_{\lambda\in\mathbb{N}}$ are two protocol families with pointwise-identical typing judgements. The asymptotic equality of two such protocol families has the form $\mathbb{T}_\approx \Rrightarrow \left\{\Delta_\lambda \vdash P_\lambda \approx Q_\lambda : I_\lambda \to O_\lambda\right\}_{\lambda\in\mathbb{N}}$, see Figure **??**, where the left-hand side of $\Rrightarrow$ lists the approximate axiom families comprising the asymptotic IPDL theory $\mathbb{T}_\approx$.

Specifically, for any fixed λ we obtain an approximate theory by selecting from each axiom family the particular axiom corresponding to λ. Similarly, from each of the two protocol families we select the protocol corresponding to λ, which gives us two concrete protocols to equate approximately. We recall that an approximate equality judgement is tagged by a pair of parameters $k \in \mathbb{N}$ and $l(t_1, \dots, t_{|\Sigma_\mathsf{t}|}) : \mathbb{N}^{|\Sigma_\mathsf{t}|} \to \mathbb{N}$, where $|\Sigma_\mathsf{t}|$ is the number of type constants declared in our ambient signature Σ. Letting $\lambda \in \mathbb{N}$ vary thus gives us two functions $k_\lambda : \mathbb{N} \to \mathbb{N}$ and $l_\lambda : \mathbb{N}^{|\Sigma_\mathsf{t}|+1} \to \mathbb{N}$, and we require that these be bounded by polynomials in the appropriate number of variables.

Whenever we want to make the underlying exact theory explicit, we write the asymptotic equality judgement as $\mathbb{T}_=;\, \mathbb{T}_\approx \Rrightarrow \left\{\Delta_\lambda \vdash P_\lambda \approx Q_\lambda : I_\lambda \to O_\lambda\right\}_{\lambda\in\mathbb{N}}$.

2 Operational Semantics of IPDL

In this section we define an operational semantics for IPDL expressions, reactions, and protocols. This semantics will validate the *exact* fragment of our equational logic and prove perfect observational equivalence. To give semantics to user-defined symbols, we define interpretations:

Definition 2 (Interpretation). *An interpretation* $[\![-]\!]$ *for a signature* Σ *associates to:*

- *each type symbol* t *a subset* $\subseteq \{0,1\}^{|\mathsf{t}|}$ *of bitstrings of a fixed length* $|\mathsf{t}| \geqslant 0$*;*
- *each function symbol* $\mathsf{f} : \sigma \to \tau$ *a function* $[\![\mathsf{f}]\!]$ *from* $[\![\sigma]\!]$ *to* $[\![\tau]\!]$*;*
- *each distribution symbol* $\mathsf{d} : \sigma \twoheadrightarrow \tau$ *a function* $[\![\mathsf{d}]\!]$ *from* $[\![\sigma]\!]$ *to* distributions *on* $[\![\tau]\!]$.

In the above, we naturally lift the interpretation $[\![-]\!]$ to all types by setting

$$\begin{aligned}[\![1]\!] &:= \{()\} \\ [\![\mathsf{Bool}]\!] &:= \{0,1\} \\ [\![\tau \times \sigma]\!] &:= \left\{v_1 v_2 \,|\, v_1 \in [\![\tau]\!], v_2 \in [\![\sigma]\!]\right\}\end{aligned}$$

where $()$ denotes the empty bitstring and $v_1 v_2$ stands for the concatenation of bitstrings v_1 and v_2. We similarly define the length $|\sigma|$ of a type σ in the obvious way, letting $|1| := 0$, $|\mathsf{Bool}| := 1$, and $|\tau \times \sigma| := |\tau| + |\sigma|$.

To handle partial computations, we augment the syntax of IPDL expressions, reactions, and protocols to contain intermediate bitstring values $v \in \{0,1\}^\star$:

$$\begin{array}{lll} \text{Valued Expressions} & e & ::= \ v \mid \dots \\ \text{Valued Reactions} & R, S & ::= \ \mathsf{val}\ v \mid \dots \\ \text{Valued Protocols} & P, Q & ::= \ o := v \mid \dots \end{array}$$

We extend the notion of a Turing Machine bound to valued IPDL constructs as follows, where $|v|$ denotes the length of the bitstring v:

$$\begin{aligned} \|v\|_{\mathsf{TM}} &:= |v| \\ \|\mathsf{val}\ v\|_{\mathsf{TM}} &:= |v| + 3 \\ \|o := v\|_{\mathsf{TM}} &:= |v| + 4 \end{aligned}$$

Given an ambient interpretation $[\![-]\!]$ for the signature Σ, we can type the valued counterpart of IPDL constructs as expected: in addition to the regular typing rules, we have

$$\frac{v \in [\![\tau]\!]}{\Gamma \vdash v : \tau} \qquad \frac{v \in [\![\tau]\!]}{\Delta;\ \Gamma \vdash \mathsf{val}\ v : I \to \tau} \qquad \frac{o : \tau \in \Delta \qquad o \notin I \qquad v \in [\![\tau]\!]}{\Delta \vdash \big(o := v\big) : I \to \{o\}}$$

2.1 Small-Step Operational Semantics of IPDL

The small-step semantics $e \to e'$ for expressions is standard, see Figure **??**. Pairing is given by bitstring concatenation (rule PAIR), and the projections $\mathsf{fst}_{\sigma\times\tau}$ and $\mathsf{snd}_{\sigma\times\tau}$ unambiguously split the pair according to $|\sigma|$ and $|\tau|$, respectively (rules FST-EVAL and SND-EVAL).

Reactions have a straightforward small-step semantics of the form $R \to \eta$, where η is a probability distribution over reactions. Figure **??** shows the rules, where we write $1[R]$ for the distribution with unit mass at the reaction R, and freely use a distribution in place of a value (rule SAMP-EVAL) or a reaction (rule BIND-REACT) to indicate the obvious lifting of the corresponding construct to distributions on reactions. All distributions are implicitly finitely supported. Crucially, there is no semantic rule for stepping the reaction $\mathsf{read}\ c$; we model communication via semantics for protocols, which substitute all instances of read for values.

We give semantics to protocols via two main small-step rules, see Figure **??**, where we analogously write $1[P]$ for the distribution with unit mass at the protocol P, and freely use a distribution in place of a reaction (in rule STEP-REACT) or a protocol (rules STEP-COMP-LEFT, STEP-COMP-RIGHT, and STEP-NEW) to indicate the obvious lifting of the corresponding construct to distributions on protocols.

First we have the *output* relation $P \xmapsto{o := v} Q$, which is enabled when the reaction for channel o in P terminates, resulting in value v (rule OUT-VAL). When this happens, the value of o is broadcast through the protocol context enveloping P (rules OUT-COMP-LEFT, OUT-COMP-LEFT, and OUT-NEW), resulting in each $\mathsf{read}\ o$ command in other reactions to be substituted with $\mathsf{val}\ v$. We note that the value of o is not broadcast above the new quantifier when the local channel introduced is equal to o.

Next we have the *internal stepping* relation $P \to \eta$, specified similarly to the small-step relation for reactions. The rule STEP-REACT lifts the stepping relation for R to the stepping relation for $o := R$, while the three rules STEP-COMP-LEFT, STEP-COMP-RIGHT, STEP-NEW simply propagate the stepping relation through parallel composition and the new quantifier. The last rule OUT-NEW-HIDE links the output relation with the stepping relation: whenever P steps to P', resulting in the output $o := v$, we have that $\mathsf{new}\ o : \tau\ \mathsf{in}\ P$ steps with unit mass to $\mathsf{new}\ o : \tau\ \mathsf{in}\ P'$.

2.2 Big-Step Operational Semantics of IPDL

The big-step semantics for expressions $e \Downarrow v$, see Figure **??**, performs a sequence of small steps to compute e to a value. The big-step semantics for reactions $R \Downarrow \eta$, see Figure **??**, performs as many steps as possible in an attempt to compute R, resulting in a distribution η on reactions. A reaction that cannot step any further is *final.* We can syntactically describe final reactions as those that have either yielded a value or have an unresolved read in the leading position (*i.e.*, are *stuck*). The big-step operational semantics for protocols $P \Downarrow \eta$, see Figure **??**, performs as many output and internal steps as possible in an attempt to compute P, resulting in a distribution η on protocols. A protocol that cannot step any further using either of the two stepping relations is *final.* We can syntactically describe final protocols as those where every channel, including the internal ones, carries either a value or a reaction that is stuck.

Note that while the small-step semantics for reactions is sequential, both output and internal step relations for protocols are non-deterministic. Indeed, any two channels in a protocol may output in any order. Ordinarily,

$$\boxed{e \to e'}$$

$$\frac{}{\checkmark \to ()}\ \textsc{check} \qquad \frac{}{\mathsf{true} \to 1}\ \textsc{true} \qquad \frac{}{\mathsf{false} \to 0}\ \textsc{false} \qquad \frac{e \to e'}{\mathsf{f}\ e \to \mathsf{f}\ e'}\ \textsc{app-cong} \qquad \frac{}{\mathsf{f}\ v \to [\![\mathsf{f}]\!](v)}\ \textsc{app-eval}$$

$$\frac{e_1 \to e_1'}{(e_1, e_2) \to (e_1', e_2)}\ \textsc{pair-cong-fst} \qquad \frac{e_2 \to e_2'}{(e_1, e_2) \to (e_1, e_2')}\ \textsc{pair-cong-snd} \qquad \frac{}{(v_1, v_2) \to v_1 v_2}\ \textsc{pair-eval}$$

$$\frac{e \to e'}{\mathsf{fst}_{\tau_1 \times \tau_2}\ e \to \mathsf{fst}_{\tau_1 \times \tau_2}\ e'}\ \textsc{fst-cong} \qquad \frac{v_1 \in \{0,1\}^{|\tau_1|} \qquad v_2 \in \{0,1\}^{|\tau_2|}}{\mathsf{fst}_{\tau_1 \times \tau_2}\ v_1 v_2 \to v_1}\ \textsc{fst-eval}$$

$$\frac{e \to e'}{\mathsf{snd}_{\tau_1 \times \tau_2}\ e \to \mathsf{snd}_{\tau_1 \times \tau_2}\ e'}\ \textsc{snd-cong} \qquad \frac{v_1 \in \{0,1\}^{|\tau_1|} \qquad v_2 \in \{0,1\}^{|\tau_2|}}{\mathsf{snd}_{\tau_1 \times \tau_2}\ v_1 v_2 \to v_2}\ \textsc{snd-eval}$$

Figure 14: Small-step operational semantics for IPDL expressions.

$$\boxed{R \to \eta}$$

$$\frac{e \to e'}{\mathsf{ret}\ e \to 1[\mathsf{ret}\ e']}\ \textsc{ret-step} \qquad \frac{}{\mathsf{ret}\ v \to 1[\mathsf{val}\ v]}\ \textsc{ret-eval} \qquad \frac{e \to e'}{\mathsf{samp}\ \mathsf{d}\ e \to 1[\mathsf{samp}\ \mathsf{d}\ e']}\ \textsc{samp-step}$$

$$\frac{}{\mathsf{samp}\ \mathsf{d}\ v \to \mathsf{val}\ [\![\mathsf{d}]\!](v)}\ \textsc{samp-eval} \qquad \frac{e \to e'}{\big(\mathsf{if}\ e\ \mathsf{then}\ R_1\ \mathsf{else}\ R_2\big) \to 1[\mathsf{if}\ e'\ \mathsf{then}\ R_1\ \mathsf{else}\ R_2]}\ \textsc{if-step}$$

$$\frac{}{\big(\mathsf{if}\ 1\ \mathsf{then}\ R_1\ \mathsf{else}\ R_2\big) \to 1[R_1]}\ \textsc{if-true} \qquad \frac{}{\big(\mathsf{if}\ 0\ \mathsf{then}\ R_1\ \mathsf{else}\ R_2\big) \to 1[R_2]}\ \textsc{if-false}$$

$$\frac{}{\big(x : \sigma \leftarrow \mathsf{val}\ v;\ S\big) \to 1\big[S[x := v]\big]}\ \textsc{bind-val} \qquad \frac{R \to \eta}{\big(x : \sigma \leftarrow R;\ S\big) \to \big(x : \sigma \leftarrow \eta;\ S\big)}\ \textsc{bind-react}$$

Figure 15: Small-step operational semantics for IPDL reactions.

$$\boxed{P \xmapsto{o := v} Q}$$

$$\frac{}{\big(o := \mathsf{val}\ v\big) \xmapsto{o := v} \big(o := v\big)}\ \textsc{out-react} \qquad \frac{P \xmapsto{o := v} P'}{P \parallel Q \xmapsto{o := v} P' \parallel Q[\mathsf{read}\ o := \mathsf{val}\ v]}\ \textsc{out-comp-left}$$

$$\frac{Q \xmapsto{o := v} Q'}{P \parallel Q \xmapsto{o := v} P[\mathsf{read}\ o := \mathsf{val}\ v] \parallel Q'}\ \textsc{out-comp-right} \qquad \frac{P \xmapsto{o := v} P' \qquad o \neq c}{\big(\mathsf{new}\ c : \tau\ \mathsf{in}\ P\big) \xmapsto{o := v} \big(\mathsf{new}\ c : \tau\ \mathsf{in}\ P'\big)}\ \textsc{out-new}$$

$$\boxed{P \to \eta}$$

$$\frac{R \to \eta}{\big(o := R\big) \to \big(o := \eta\big)}\ \textsc{step-react} \qquad \frac{P \to \eta}{P \parallel Q \to \eta \parallel Q}\ \textsc{step-comp-left} \qquad \frac{Q \to \eta}{P \parallel Q \to P \parallel \eta}\ \textsc{step-comp-right}$$

$$\frac{P \to \eta}{\big(\mathsf{new}\ c : \tau\ \mathsf{in}\ P\big) \to \big(\mathsf{new}\ c : \tau\ \mathsf{in}\ \eta\big)}\ \textsc{step-new} \qquad \frac{P \xmapsto{c := v} P'}{\big(\mathsf{new}\ c : \tau\ \mathsf{in}\ P\big) \to 1[\mathsf{new}\ c : \tau\ \mathsf{in}\ P']}\ \textsc{out-new-hide}$$

Figure 16: Small-step operational semantics for IPDL protocols.

$$\boxed{e \Downarrow v}$$

$$\frac{}{v \Downarrow v} \qquad \frac{e \to e' \qquad e' \Downarrow v}{e \Downarrow v}$$

Figure 17: Big-step operational semantics for IPDL expressions.

$$\boxed{R\ \mathsf{stuck}}$$

$$\frac{}{\big(\mathsf{read}\ c\big)\ \mathsf{stuck}} \qquad \frac{R\ \mathsf{stuck}}{\big(x : \sigma \leftarrow R;\ S\big)\ \mathsf{stuck}}$$

$$\boxed{R\ \mathsf{final}}$$

$$\frac{}{\big(\mathsf{val}\ v\big)\ \mathsf{final}} \qquad \frac{R\ \mathsf{stuck}}{R\ \mathsf{final}}$$

$$\boxed{R \Downarrow \eta}$$

$$\frac{R\ \mathsf{final}}{R \Downarrow 1[R]}$$

$$\frac{R \to \sum_i c_i * 1[R_i] \qquad R_i \Downarrow \eta_i}{R \Downarrow \sum_i c_i * \eta_i}$$

Figure 18: Big-step operational semantics for IPDL reactions.

$$\boxed{P \ \mathsf{final}}$$

$$\frac{}{\mathsf{0\ final}} \qquad \frac{}{\big(o := v\big)\ \mathsf{final}} \qquad \frac{R\ \mathsf{stuck}}{\big(o := R\big)\ \mathsf{final}} \qquad \frac{P\ \mathsf{final} \qquad Q\ \mathsf{final}}{P \parallel Q\ \mathsf{final}} \qquad \frac{P\ \mathsf{final}}{\big(\mathsf{new}\ o : \tau\ \mathsf{in}\ P\big)\ \mathsf{final}}$$

$$\boxed{P \Downarrow \eta}$$

$$\frac{P\ \mathsf{final}}{P \Downarrow 1[P]} \qquad \frac{P \xmapsto{o := v} Q \qquad Q \Downarrow \eta}{P \Downarrow \eta}$$

$$\frac{P \to \sum_i c_i * 1[P_i] \qquad P_i \Downarrow \eta_i}{P \Downarrow \sum_i c_i * \eta_i}$$

Figure 19: Big-step operational semantics for IPDL protocols.

this presents a problem for reasoning about cryptography, since non-deterministic choice may present a security leak. However, our language introduces *no* way to exploit this extra non-determinism, essentially due to the read commands in reactions being blocking. This is formalized by a number of *confluence* results for IPDL:

Lemma 1 (Confluence for expressions)**.** *For any expression $\Gamma \vdash e : \tau$ we have the following:*

- *If $e \to e_1$ and $e \to e_2$ then either $e_1 = e_2$ or there is e' such that $e_1 \to e'$ and $e_2 \to e'$.*
- *If $e \to e'$ and $e \Downarrow v$ then $e' \Downarrow v$.*

Lemma 2 (Confluence for reactions)**.** *For any reaction $\Delta;\ \Gamma \vdash R : I \to \tau$ we have the following:*

- *We have $R[\mathsf{read}\ i := \mathsf{val}\ v][\mathsf{read}\ i := \mathsf{val}\ v] = R[\mathsf{read}\ i := \mathsf{val}\ v]$.*
- *Let $i_1 \neq i_2$. Then $R[\mathsf{read}\ i_1 := \mathsf{val}\ v_1][\mathsf{read}\ i_2 := \mathsf{val}\ v_2] = R[\mathsf{read}\ i_2 := \mathsf{val}\ v_2][\mathsf{read}\ i_1 := \mathsf{val}\ v_1]$.*
- *If $R \to \eta$ then $R[\mathsf{read}\ i := \mathsf{val}\ v] \to \eta[\mathsf{read}\ i := \mathsf{val}\ v]$.*

Lemma 3 (Confluence for protocols, part I)**.** *For any protocol $\Delta \vdash P : I \to O$ we have the following:*

- *We have $P[\mathsf{read}\ i := \mathsf{val}\ v][\mathsf{read}\ i := \mathsf{val}\ v] = P[\mathsf{read}\ i := \mathsf{val}\ v]$.*
- *Let $i_1 \neq i_2$. Then $P[\mathsf{read}\ i_1 := \mathsf{val}\ v_1][\mathsf{read}\ i_2 := \mathsf{val}\ v_2] = P[\mathsf{read}\ i_2 := \mathsf{val}\ v_2][\mathsf{read}\ i_1 := \mathsf{val}\ v_1]$.*
- *Let $o_1 \neq o_2$. If $P \xmapsto{o_1 := v_1} Q$ then $P[\mathsf{read}\ o_2 := \mathsf{val}\ v_2] \xmapsto{o_1 := v_1} Q[\mathsf{read}\ o_2 := \mathsf{val}\ v_2]$.*
- *If $P \to \eta$ then $P[\mathsf{read}\ o := \mathsf{val}\ v] \to \eta[\mathsf{read}\ o := \mathsf{val}\ v]$.*

Lemma 4 (Confluence for protocols, part II)**.** *For any protocol $\Delta \vdash P : I \to O$ we have the following:*

- *Let $o_1 \neq o_2$. If $P \xmapsto{o_1 := v_1} Q$ and $P \xmapsto{o_2 := v_2} P'$ then there is Q' such that $Q \xmapsto{o_2 := v_2} Q'$ and $P' \xmapsto{o_1 := v_1} Q'$.*
- *If $P \xmapsto{o := v} P'$ and $P \to \eta$ then there is η' such that $\eta \xmapsto{o := v} \eta'$ and $P' \to \eta'$.*
- *If $P \to \eta_1$ and $P \to \eta_2$ then either $\eta_1 = \eta_2$ or there is η such that $\eta_1 \to \eta$ and $\eta_2 \to \eta$.*

In the above lemma, we lift the two protocol stepping relations $\xmapsto{o := v}$ and → to distributions in the natural way.

Lemma 5 (Confluence for protocols, part III)**.** *For any protocol $\Delta \vdash P : I \to O$ we have the following:*

- *If* $P \xmapsto{o:=v} Q$ *and* $P \Downarrow \varepsilon$ *then* $Q \Downarrow \varepsilon$.
- *If* $P \to \eta$ *and* $P \Downarrow \varepsilon$ *then* $\eta \Downarrow \varepsilon$.

To guarantee termination (for protocols especially), we introduce the notion of a *structure bound.* The structure bound for expressions, defined below, is invariant under substitutions.

$$\begin{aligned}
\|v\|_{\mathsf{str}} &:= 0 \\
\|\checkmark\|_{\mathsf{str}} &:= 1 \\
\|\mathsf{true}\|_{\mathsf{str}} &:= 1 \\
\|\mathsf{false}\|_{\mathsf{str}} &:= 1 \\
\|\mathsf{f}\ e\|_{\mathsf{str}} &:= \|e\|_{\mathsf{str}} + 1 \\
\|(e_1, e_2)\|_{\mathsf{str}} &:= \|e_1\|_{\mathsf{str}} + \|e_2\|_{\mathsf{str}} + 1 \\
\|\mathsf{fst}\ e\|_{\mathsf{str}} &:= \|e\|_{\mathsf{str}} + 1 \\
\|\mathsf{snd}\ e\|_{\mathsf{str}} &:= \|e\|_{\mathsf{str}} + 1
\end{aligned}$$

The structure bound for reactions, defined below, is invariant under substitutions, embeddings, and input assignment.

$$\begin{aligned}
\|\mathsf{val}\ v\|_{\mathsf{str}} &:= 0 \\
\|\mathsf{ret}\ e\|_{\mathsf{str}} &:= \|e\|_{\mathsf{str}} + 1 \\
\|\mathsf{samp}\ \mathsf{d}\ e\|_{\mathsf{str}} &:= \|e\|_{\mathsf{str}} + 1 \\
\|\mathsf{read}\ c\|_{\mathsf{str}} &:= 0 \\
\|\mathsf{if}\ e\ \mathsf{then}\ R_1\ \mathsf{else}\ R_2\|_{\mathsf{str}} &:= \|e\|_{\mathsf{str}} + \mathsf{max}\left(\|R_1\|_{\mathsf{str}}, \|R_2\|_{\mathsf{str}}\right) + 1 \\
\|x \leftarrow R;\ S\|_{\mathsf{str}} &:= \|R\|_{\mathsf{str}} + \|S\|_{\mathsf{str}} + 1
\end{aligned}$$

The structure bound for protocols, defined below, is invariant under embeddings and input assignment.

$$\begin{aligned}
\|0\|_{\mathsf{str}} &:= 0 \\
\|o := v\|_{\mathsf{str}} &:= 0 \\
\|o := R\|_{\mathsf{str}} &:= \|R\|_{\mathsf{str}} + 1 \\
\|P \parallel Q\|_{\mathsf{str}} &:= \|P\|_{\mathsf{str}} + \|Q\|_{\mathsf{str}} \\
\|\mathsf{new}\ c : \tau\ \mathsf{in}\ P\|_{\mathsf{str}} &:= \|P\|_{\mathsf{str}}
\end{aligned}$$

Lemma 6 (Progress for expressions)**.** *For any expression* $\cdot \vdash e : \tau$ *we have the following:*

- *If* $e \to e'$ *then* $\|e'\|_{\mathsf{str}} < \|e\|_{\mathsf{str}}$.

Lemma 7 (Progress for reactions)**.** *For any reaction* $\Delta;\ \cdot \vdash R : I \to \tau$ *we have the following:*

- *If* $R \to \sum_i c_i * 1[R_i]$ *with* $c_i \neq 0$*, then* $\|R_i\|_{\mathsf{str}} < \|R\|_{\mathsf{str}}$.

Lemma 8 (Progress for protocols)**.** *For any protocol* $\Delta \vdash P : I \to O$ *we have the following:*

- *If* $P \xmapsto{o:=v} Q$ *then* $\|Q\|_{\mathsf{str}} < \|P\|_{\mathsf{str}}$.
- *If* $P \to \sum_i c_i * 1[P_i]$ *with* $c_i \neq 0$*, then* $\|P_i\|_{\mathsf{str}} < \|P\|_{\mathsf{str}}$.

The confluence and progress lemmas together imply that our semantics is indeed well-defined:

Corollary 1 (Determinism of $\Downarrow$ for expressions)**.** *For any expression* $\Gamma \vdash e : \tau$*, there exists a unique bitstring* v *such that* $e \Downarrow v$*. We will denote* v *by* $e\Downarrow$.

Corollary 2 (Determinism of $\Downarrow$ for reactions)**.** *For any reaction* $\Delta;\ \cdot \vdash R : I \to \tau$*, there exists a unique distribution* η *on reactions such that* $R \Downarrow \eta$*. We will denote* η *by* $R\Downarrow$.

Corollary 3 (Determinism of $\Downarrow$ for protocols)**.** *For any protocol* $\Delta \vdash P : I \to O$*, there exists a unique distribution* η *on protocols such that* $P \Downarrow \eta$*. We will denote* η *by* $P\Downarrow$.

3 Soundness of Exact Equality in IPDL

Soundness of equality at the expression level means that if we substitute the same valued expression for each free variable, the resulting closed expressions will compute to the same value:

Definition 3. *An axiom* $\Gamma \vdash e_1 = e_2 : \tau$ *is* sound *if for any valued substitution* $\theta : \cdot \to \Gamma$ *we have* $\theta^\star(e_1)\Downarrow = \theta^\star(e_2)\Downarrow$*.*

The ambient IPDL theory $\mathbb{T}_{\mathsf{exp}}$ for expressions is said to be sound if each of its axioms is sound. It is straightforward to show that this implies overall soundness:

Lemma 9 (Soundness of equality of expressions)**.** *If the ambient IPDL theory for expressions is sound, then for any expressions* $\Gamma \vdash e_1 = e_2 : \tau$ *and any valued substitution* $\theta : \cdot \to \Gamma$ *we have that* $\theta^\star(e_1)\Downarrow = \theta^\star(e_2)\Downarrow$*.*

At the reaction level, two equal reactions should behave in a way that is perfectly indistinguishable by an external observer. We formally capture this notion of indistinguishability by a logical relation known as a *bisimulation* – a binary relation on distributions on reactions that satisfies certain closure properties, together with the crucial *valuation property* that allows us to jointly partition two related final distributions so that any two corresponding components are again related and have the same *value*: a final reaction $\Delta;\ \cdot \vdash R : I \to \tau$ is said to have value $v \in [\![\tau]\!]$ if R is of the form $\mathsf{val}\ v$; if R is stuck we define the value to be $\bot$. Given a $v_\bot \in \{\bot\} \cup [\![\tau]\!]$, we write $R\,|_{\mathsf{val}} = v_\bot$ to indicate that the value of R is $v_\bot$, and lift this notation to distributions in the obvious way.

We emphasize that at the reaction level, we only require the valuation property to hold for those distributions that are *final*, *i.e.*, no reaction in the support steps.

Definition 4 (Reaction bisimulation)**.** *A* reaction bisimulation $\sim$ *is a binary relation on distributions on reactions of type* $\Delta;\ \cdot \vdash I \to \tau$ *satisfying the following conditions:*

- Closure under convex combinations*: For any distributions* $\eta_1 \sim \varepsilon_1$ *and* $\eta_2 \sim \varepsilon_2$*, and any coefficients* $c_1, c_2 > 0$ *with* $c_1 + c_2 = 1$*, we have* $(c_1 * \eta_1 + c_2 * \eta_2) \sim (c_1 * \varepsilon_1 + c_2 * \varepsilon_2)$*.*
- Closure under input assignment*: For any distributions* $\eta \sim \varepsilon$*, input channel* $i \in I$ *with* $i : \tau \in \Delta$*, and value* $v \in [\![\tau]\!]$*, we have* $\eta[\mathsf{read}\ i := \mathsf{val}\ v] \sim \varepsilon[\mathsf{read}\ i := \mathsf{val}\ v]$*.*
- Closure under computation*: For any distributions* $\eta \sim \varepsilon$*, we have* $(\eta\Downarrow) \sim (\varepsilon\Downarrow)$*.*
- Valuation property*: For any final distributions* $\eta \sim \varepsilon$*, there exists a joint convex combination*

$$\eta = \sum_i c_i * \eta_i \ \sim \ \sum_i c_i * \varepsilon_i = \varepsilon$$

  with $c_i > 0$ *and* $\sum_i c_i = 1$ *such that*

  – *the respective components* $\eta_i \sim \varepsilon_i$ *are again related, and*

  – $\eta_i\,|_{\mathsf{val}} = v_\bot = \varepsilon_j\,|_{\mathsf{val}}$ *for the same* $v_\bot \in \{\bot\} \cup [\![\tau]\!]$ *if and only if* $i = j$*.*

Crucially, we note that the joint convex combination in the valuation property is unique up to the order of the summands.

Lemma 10. *We have the following:*

- *The identity relation is a reaction bisimulation.*
- *The inverse of a reaction bisimulation is a reaction bisimulation.*
- *The composition of two reaction bisimulations is a reaction bisimulation.*

We now describe one canonical way to construct reaction bisimulations:

Definition 5. *Let* $\sim$ *be an arbitrary binary relation on distributions on reactions of type* $\Delta;\ \cdot \vdash I \to \tau$*. The* expansion $\mathcal{L}(\sim)$ *is the closure of* $\sim$ *under joint convex combinations. Explicitly,* $\mathcal{L}(\sim)$ *is defined by*

$$\Big(\sum_i c_i * \eta_i\Big)\ \mathcal{L}(\sim)\ \Big(\sum_i c_i * \varepsilon_i\Big)$$

for coefficients $c_i > 0$ *with* $\sum_i c_i = 1$ *and distributions* $\eta_i \sim \varepsilon_i$*.*

Lemma 11. *Let $\sim$ be a binary relation on distributions on reactions of type $\Delta;\ \cdot \vdash I \to \tau$ with the following properties:*

- Closure under input assignment*: For any distributions $\eta \sim \varepsilon$, input channel $i \in I$ with $i : \tau \in \Delta$, and value $v \in [\![\tau]\!]$, we have $\eta[\mathsf{read}\ i := \mathsf{val}\ v] \sim \varepsilon[\mathsf{read}\ i := \mathsf{val}\ v]$.*
- Expansion closure under computation*: For any distributions $\eta \sim \varepsilon$, we have $(\eta\Downarrow)\ \mathcal{L}(\sim)\ (\varepsilon\Downarrow)$.*
- Valuation property*: For any final distributions $\eta \sim \varepsilon$, there exists a joint convex combination*

$$\eta = \sum_i c_i * \eta_i \ \sim\ \sum_i c_i * \varepsilon_i = \varepsilon$$

  with $c_i > 0$ and $\sum_i c_i = 1$ such that

  - *the respective components $\eta_i \sim \varepsilon_i$ are again related, and*
  - *$\eta_i\,|_{\mathsf{val}} = v_\perp = \varepsilon_j\,|_{\mathsf{val}}$ for the same $v_\perp \in \{\perp\} \cup [\![\tau]\!]$ if and only if $i = j$.*

Then the expansion $\mathcal{L}(\sim)$ is a reaction bisimulation.

Example 1. *Fix two expressions $\cdot \vdash e_1 : \sigma$ and $\cdot \vdash e_2 : \sigma$ such that $e_1\Downarrow = e_2\Downarrow$. Then the relation $\sim$ defined by*

- $1[R(x := e_1)] \sim 1[R(x := e_2)]$ *for reaction $\Delta;\ x : \sigma \vdash R : I \to \tau$*

satisfies the hypotheses of Lemma **??** *(and its lifting is hence a reaction bisimulation).*

Having defined reaction bisimulations, we can now formally state what it means for reaction equality to be sound:

Definition 6. *An axiom $\Gamma \vdash R_1 = R_2 : \tau$ is* sound *if there is a reaction bisimulation $\sim$ such that for any valued substitution $\theta : \cdot \to \Gamma$ we have $1[\theta^\star(R_1)] \sim 1[\theta^\star(R_2)]$.*

The ambient IPDL theory $\mathbb{T}_{\mathsf{dist}}$ for reactions is said to be sound if each of its axioms is sound. We now show that this implies overall soundness:

Lemma 12 (Soundness of equality of reactions)**.** *If the ambient IPDL theories for expressions and reactions are sound, then for any reactions $\Delta;\ \Gamma \vdash R_1 = R_2 : I \to \tau$ there exists a reaction bisimulation $\sim$ such that for any valued substitution $\theta : \cdot \to \Gamma$ we have $1[\theta^\star(R_1)] \sim 1[\theta^\star(R_2)]$.*

Proof. We first replace the exchange rule EXCH by the three rules EXCH-SAMP-SAMP, EXCH-SAMP-READ, and EXCH-READ-READ in Figure **??**; it is easy to see that this new set of rules is equivalent to the original one. We now proceed by induction on the alternative set of rules for reaction equality. We will freely use a distribution in place of a value (rule EXCH-SAMP-READ) or a reaction (rules EMBED, CONG-BIND) to indicate the obvious lifting of the corresponding construct to distributions on reactions.

- REFL: Our desired bisimulation is the identity relation.
- SYM: Our desired bisimulation is the inverse of the bisimulation obtained from the premise.
- TRANS: Our desired bisimulation is the composition of the two bisimulations obtained from the two premises.
- AXIOM: Our desired bisimulation is precisely the bisimulation obtained from the soundness of the axiom.
- INPUT-UNUSED: Our desired bisimulation is precisely the bisimulation obtained from the premise, seen as a bisimulation on distributions on reactions with the additional input i.
- SUBST: Our desired bisimulation is precisely the bisimulation obtained from the premise.
- EMBED: Let $\sim$ be the bisimulation obtained from the premise. Our desired bisimulation $\sim_\phi$ is defined by
  - $\phi^\star(\eta) \sim_\phi \phi^\star(\varepsilon)$ if $\eta \sim \varepsilon$
- CONG-RET: Our desired bisimulation is the expansion of the relation $\sim$ defined by

- $1[\mathsf{ret}\ e] \sim 1[\mathsf{ret}\ e']$ for expressions $\cdot \vdash e : \tau$ and $\cdot \vdash e' : \tau$ such that $e\Downarrow = e'\Downarrow$
- $1[\mathsf{val}\ v] \sim 1[\mathsf{val}\ v]$ for value $v \in [\![\tau]\!]$

- CONG-SAMP: Our desired bisimulation is the expansion of the relation $\sim$ defined by
  - $1[\mathsf{samp}\ \mathsf{d}\ e] \sim 1[\mathsf{samp}\ \mathsf{d}\ e']$ for expressions $\cdot \vdash e : \sigma$ and $\cdot \vdash e' : \sigma$ such that $e\Downarrow = e'\Downarrow$
  - $1[\mathsf{val}\ v] \sim 1[\mathsf{val}\ v]$ for value $v \in [\![\tau]\!]$

- CONG-IF: Let $\sim_1$ and $\sim_2$ be the two bisimulations obtained from the two premises. Our desired bisimulation is the expansion of the relation $\sim_{\mathsf{if}}$ defined by
  - $1[\mathsf{if}\ e\ \mathsf{then}\ R_1\ \mathsf{else}\ R_2] \sim_{\mathsf{if}} 1[\mathsf{if}\ e'\ \mathsf{then}\ R'_1\ \mathsf{else}\ R'_2]$ for
    * expressions $\cdot \vdash e : \mathsf{Bool}$ and $\cdot \vdash e' : \mathsf{Bool}$ such that $e\Downarrow = e'\Downarrow$
    * reactions $\Delta;\ \cdot \vdash R_1 : I \to \tau$ and $\Delta;\ \cdot \vdash R'_1 : I \to \tau$ such that $1[R_1] \sim_1 1[R'_1]$
    * reactions $\Delta;\ \cdot \vdash R_2 : I \to \tau$ and $\Delta;\ \cdot \vdash R'_2 : I \to \tau$ such that $1[R_2] \sim_2 1[R'_2]$
  - $\eta_1 \sim_{\mathsf{if}} \eta'_1$ if $\eta_1 \sim_1 \eta'_1$
  - $\eta_2 \sim_{\mathsf{if}} \eta'_2$ if $\eta_2 \sim_2 \eta'_2$

- CONG-BIND: Let $\sim_1$ and $\sim_2$ be the two bisimulations obtained from the two premises. Our desired bisimulation is the expansion of the relation $\sim_{\mathsf{bind}}$ defined by
  - $\big(x \leftarrow \eta;\ S\big) \sim_{\mathsf{bind}} \big(x \leftarrow \eta';\ S'\big)$ for
    * distributions $\eta \sim_1 \eta'$
    * reactions $\Delta;\ x : \sigma \vdash S : I \to \tau$ and $\Delta;\ x : \sigma \vdash S' : I \to \tau$ such that for any value $v \in [\![\sigma]\!]$ we have $1[S(x := v)] \sim_2 1[S'(x := v)]$
  - $\varepsilon \sim_{\mathsf{bind}} \varepsilon'$ if $\varepsilon \sim_2 \varepsilon'$

- RET-BIND: Our desired bisimulation is the expansion of the relation $\sim$ defined by
  - $1[x \leftarrow \mathsf{ret}\ e;\ R] \sim 1[R(x := e)]$ for expression $\cdot \vdash e : \sigma$ and reaction $\Delta;\ x : \sigma \vdash R : I \to \tau$
  - $1[R(x := e\Downarrow)] \sim 1[R(x := e)]$ for reaction $\Delta;\ x : \sigma \vdash R : I \to \tau$ and expression $\cdot \vdash e : \sigma$

- BIND-RET: Our desired bisimulation is the expansion of the relation $\sim$ defined by
  - $1[x \leftarrow R;\ \mathsf{ret}\ x] \sim 1[R]$ for reaction $\Delta;\ \cdot \vdash R : I \to \tau$
  - $1[\mathsf{val}\ v] \sim 1[\mathsf{val}\ v]$ for value $v \in [\![\tau]\!]$

- BIND-BIND: Our desired bisimulation is the expansion of the relation $\sim$ defined by
  - $1[x_2 \leftarrow (x_1 \leftarrow R_1;\ R_2);\ S] \sim 1[x_1 \leftarrow R_1;\ x_2 \leftarrow R_2;\ S]$ for
    * reaction $\Delta;\ \cdot \vdash R_1 : I \to \sigma_1$
    * reaction $\Delta;\ x_1 : \sigma_1 \vdash R_2 : I \to \sigma_2$
    * reaction $\Delta;\ x_2 : \sigma_2 \vdash S : I \to \tau$
  - $1[x_2 \leftarrow R_2;\ S] \sim 1[x_2 \leftarrow R_2;\ S]$ for reactions $\Delta;\ \cdot \vdash R_2 : I \to \sigma_2$ and $\Delta;\ x_2 : \sigma_2 \vdash S : I \to \tau$
  - $1[S] \sim 1[S]$ for reaction $\Delta;\ \cdot \vdash S : I \to \tau$

- SAMP-PURE: Our desired bisimulation is the expansion of the relation $\sim$ defined by
  - $1[x \leftarrow \mathsf{samp}\ \mathsf{d}\ e;\ R] \sim 1[R]$ for expression $\cdot \vdash e : \rho$ and reaction $\Delta;\ \cdot \vdash R : I \to \tau$
  - $1[R] \sim 1[R]$ for reaction $\Delta;\ \cdot \vdash R : I \to \tau$

- READ-DET: Our desired bisimulation is the expansion of the relation $\sim$ defined by
  - $1[x \leftarrow \mathsf{read}\ i;\ y \leftarrow \mathsf{read}\ i;\ R] \sim 1[x \leftarrow \mathsf{read}\ i;\ R(y := x)]$ for reaction $\Delta;\ x : \sigma, y : \sigma \vdash R : I \to \tau$
  - $1[x \leftarrow \mathsf{val}\ v;\ y \leftarrow \mathsf{val}\ v;\ R] \sim 1[x \leftarrow \mathsf{val}\ v;\ R(y := x)]$ for

    - reaction $\Delta;\ x : \sigma, y : \sigma \vdash R : I \to \tau$
    - value $v \in [\![\sigma]\!]$
  - $1[R] \sim 1[R]$ for reaction $\Delta;\ \cdot \vdash R : I \to \tau$

- IF-LEFT: Our desired bisimulation is the expansion of the relation $\sim$ defined by
  - $1[\mathsf{if\ true\ then}\ R_1\ \mathsf{else}\ R_2] \sim 1[R_1]$ for reactions $\Delta;\ \cdot \vdash R_1 : I \to \tau$ and $\Delta;\ \cdot \vdash R_2 : I \to \tau$
  - $1[R_1] \sim 1[R_1]$ for reaction $\Delta;\ \cdot \vdash R_1 : I \to \tau$

- IF-RIGHT: Our desired bisimulation is the expansion of the relation $\sim$ defined by
  - $1[\mathsf{if\ false\ then}\ R_1\ \mathsf{else}\ R_2] \sim 1[R_2]$ for reactions $\Delta;\ \cdot \vdash R_1 : I \to \tau$ and $\Delta;\ \cdot \vdash R_2 : I \to \tau$
  - $1[R_2] \sim 1[R_2]$ for reaction $\Delta;\ \cdot \vdash R_2 : I \to \tau$

- IF-EXT: Our desired bisimulation is the expansion of the relation $\sim$ defined by
  - $1[\mathsf{if}\ e\ \mathsf{then}\ R(x := \mathsf{true})\ \mathsf{else}\ R(x := \mathsf{false})] \sim 1[R(x := e)]$ for
    - reaction $\Delta;\ x : \mathsf{Bool} \vdash R : I \to \tau$
    - expression $\cdot \vdash e : \mathsf{Bool}$
  - $1[R(x := \mathsf{true})] \sim 1[R(x := e)]$ for
    - reaction $\Delta;\ x : \mathsf{Bool} \vdash R : I \to \tau$
    - expression $\cdot \vdash e : \mathsf{Bool}$ such that $e \Downarrow 1$
  - $1[R(x := \mathsf{false})] \sim 1[R(x := e)]$ for
    - reaction $\Delta;\ x : \mathsf{Bool} \vdash R : I \to \tau$
    - expression $\cdot \vdash e : \mathsf{Bool}$ such that $e \Downarrow 0$

- EXCH-SAMP-SAMP: Our desired bisimulation is the expansion of the relation $\sim$ defined by
  - $1[x_1 \leftarrow \mathsf{samp}\ \mathsf{d}_1\ e_1;\ x_2 \leftarrow \mathsf{samp}\ \mathsf{d}_2\ e_2;\ \mathsf{ret}\ (x_1, x_2)] \sim$
    $1[x_2 \leftarrow \mathsf{samp}\ \mathsf{d}_2\ e_2;\ x_1 \leftarrow \mathsf{samp}\ \mathsf{d}_1\ e_1;\ \mathsf{ret}\ (x_1, x_2)]$ for
    - expressions $\cdot \vdash e_1 : \sigma_1$ and $\cdot \vdash e_2 : \sigma_2$
  - $1[\mathsf{val}\ v_1 v_2] \sim 1[\mathsf{val}\ v_1 v_2]$ for values $v_1 \in [\![\tau_1]\!]$ and $v_2 \in [\![\tau_2]\!]$

- EXCH-SAMP-READ: Our desired bisimulation is the expansion of the relation $\sim$ defined by
  - $1[x_1 \leftarrow \mathsf{samp\ d}\ e;\ x_2 \leftarrow \mathsf{read}\ i;\ \mathsf{ret}\ (x_1, x_2)] \sim 1[x_2 \leftarrow \mathsf{read}\ i;\ x_1 \leftarrow \mathsf{samp\ d}\ e;\ \mathsf{ret}\ (x_1, x_2)]$ for
    - expression $\cdot \vdash e : \sigma$
  - $1[x_1 \leftarrow \mathsf{samp\ d}\ e;\ x_2 \leftarrow \mathsf{val}\ v_2;\ \mathsf{ret}\ (x_1, x_2)] \sim 1[x_2 \leftarrow \mathsf{val}\ v_2;\ x_1 \leftarrow \mathsf{samp\ d}\ e;\ \mathsf{ret}\ (x_1, x_2)]$ for
    - expression $\cdot \vdash e : \sigma$
    - value $v_2 \in [\![\tau_2]\!]$
  - $\big(x_2 \leftarrow \mathsf{read}\ i;\ \mathsf{ret}\ ([\![\mathsf{d}]\!](e\Downarrow), x_2)\big) \sim 1[x_2 \leftarrow \mathsf{read}\ i;\ x_1 \leftarrow \mathsf{samp\ d}\ e;\ \mathsf{ret}\ (x_1, x_2)]$ for
    - expression $\cdot \vdash e : \sigma$
  - $\big(x_2 \leftarrow \mathsf{val}\ v_2;\ \mathsf{ret}\ ([\![\mathsf{d}]\!](e\Downarrow), x_2)\big) \sim 1[x_2 \leftarrow \mathsf{val}\ v_2;\ x_1 \leftarrow \mathsf{samp\ d}\ e;\ \mathsf{ret}\ (x_1, x_2)]$ for
    - expression $\cdot \vdash e : \sigma$
    - value $v_2 \in [\![\tau_2]\!]$
  - $1[\mathsf{val}\ v_1 v_2] \sim 1[\mathsf{val}\ v_1 v_2]$ for values $v_1 \in [\![\tau_1]\!]$ and $v_2 \in [\![\tau_2]\!]$

- EXCH-READ-READ: Our desired bisimulation is the expansion of the relation $\sim$ defined by
  - $1[x_1 \leftarrow \mathsf{read}\ i_1;\ x_2 \leftarrow \mathsf{read}\ i_2;\ \mathsf{ret}\ (x_1, x_2)] \sim 1[x_2 \leftarrow \mathsf{read}\ i_2;\ x_1 \leftarrow \mathsf{read}\ i_1;\ \mathsf{ret}\ (x_1, x_2)]$
  - $1[x_1 \leftarrow \mathsf{val}\ v_1;\ x_2 \leftarrow \mathsf{read}\ i_2;\ \mathsf{ret}\ (x_1, x_2)] \sim 1[x_2 \leftarrow \mathsf{read}\ i_2;\ x_1 \leftarrow \mathsf{val}\ v_1;\ \mathsf{ret}\ (x_1, x_2)]$ for value $v_1 \in [\![\tau_1]\!]$
  - $1[x_1 \leftarrow \mathsf{read}\ i_1;\ x_2 \leftarrow \mathsf{val}\ v_2;\ \mathsf{ret}\ (x_1, x_2)] \sim 1[x_2 \leftarrow \mathsf{val}\ v_2;\ x_1 \leftarrow \mathsf{read}\ i_1;\ \mathsf{ret}\ (x_1, x_2)]$ for value $v_2 \in [\![\tau_2]\!]$

$$\frac{\mathsf{d}_1 : \sigma_1 \twoheadrightarrow \tau_1, \mathsf{d}_2 : \sigma_2 \twoheadrightarrow \tau_2 \in \Sigma \qquad \Gamma \vdash e_1 : \sigma_1 \qquad \Gamma \vdash e_2 : \sigma_2}{\begin{array}{l}\Delta;\ \Gamma \vdash \big(x_1 : \tau_1 \leftarrow \mathsf{samp}_{\sigma_1 \twoheadrightarrow \tau_1}\ \mathsf{d}_1\ e_1;\ x_2 : \tau_2 \leftarrow \mathsf{samp}_{\sigma_2 \twoheadrightarrow \tau_2}\ \mathsf{d}_2\ e_2;\ \mathsf{ret}\ (x_1, x_2)\big) = \\ \qquad \big(x_2 : \tau_2 \leftarrow \mathsf{samp}_{\sigma_2 \twoheadrightarrow \tau_2}\ \mathsf{d}_2\ e_2;\ x_1 : \tau_1 \leftarrow \mathsf{samp}_{\sigma_1 \twoheadrightarrow \tau_1}\ \mathsf{d}_1\ e_1;\ \mathsf{ret}\ (x_1, x_2)\big) : I \to \tau_1 \times \tau_2\end{array}}\ \textsc{exch-samp-samp}$$

$$\frac{\mathsf{d} : \sigma \twoheadrightarrow \tau_1 \in \Sigma \qquad \Gamma \vdash e : \sigma \qquad i : \tau_2 \in \Delta \qquad i \in I}{\begin{array}{l}\Delta;\ \Gamma \vdash \big(x_1 : \tau_1 \leftarrow \mathsf{samp}_{\sigma \twoheadrightarrow \tau_1}\ \mathsf{d}\ e;\ x_2 : \tau_2 \leftarrow \mathsf{read}\ i;\ \mathsf{ret}\ (x_1, x_2)\big) = \\ \qquad \big(x_2 : \tau_2 \leftarrow \mathsf{read}\ i;\ x_1 : \tau_1 \leftarrow \mathsf{samp}_{\sigma \twoheadrightarrow \tau_1}\ \mathsf{d}\ e;\ \mathsf{ret}\ (x_1, x_2)\big) : I \to \tau_1 \times \tau_2\end{array}}\ \textsc{exch-samp-read}$$

$$\frac{i_1 : \tau_1, i_2 : \tau_2 \in \Delta \qquad i_1, i_2 \in I}{\begin{array}{l}\Delta;\ \Gamma \vdash \big(x_1 : \tau_1 \leftarrow \mathsf{read}\ i_1;\ x_2 : \tau_2 \leftarrow \mathsf{read}\ i_2;\ \mathsf{ret}\ (x_1, x_2)\big) = \\ \qquad \big(x_2 : \tau_2 \leftarrow \mathsf{read}\ i_2;\ x_1 : \tau_1 \leftarrow \mathsf{read}\ i_1;\ \mathsf{ret}\ (x_1, x_2)\big) : I \to \tau_1 \times \tau_2\end{array}}\ \textsc{exch-read-read}$$

Figure 20: Alternative formulation of the EXCH rule for reaction equality.

- $1[x_1 \leftarrow \mathsf{val}\ v_1;\ x_2 \leftarrow \mathsf{val}\ v_2;\ \mathsf{ret}\ (x_1, x_2)] \sim 1[x_2 \leftarrow \mathsf{val}\ v_2;\ x_1 \leftarrow \mathsf{val}\ v_1;\ \mathsf{ret}\ (x_1, x_2)]$ for
  - values $v_1 \in \llbracket \tau_1 \rrbracket$ and $v_2 \in \llbracket \tau_2 \rrbracket$
- $1[x_2 \leftarrow \mathsf{read}\ i_2;\ \mathsf{ret}\ (v_1, x_2)] \sim 1[x_2 \leftarrow \mathsf{read}\ i_2;\ x_1 \leftarrow \mathsf{val}\ v_1;\ \mathsf{ret}\ (x_1, x_2)]$ for value $v_1 \in \llbracket \tau_1 \rrbracket$
- $1[x_1 \leftarrow \mathsf{read}\ i_1;\ x_2 \leftarrow \mathsf{val}\ v_2;\ \mathsf{ret}\ (x_1, x_2)] \sim 1[x_1 \leftarrow \mathsf{read}\ i_1;\ \mathsf{ret}\ (x_1, v_2)]$ for value $v_2 \in \llbracket \tau_2 \rrbracket$
- $1[x_2 \leftarrow \mathsf{val}\ v_2;\ \mathsf{ret}\ (v_1, x_2)] \sim 1[x_2 \leftarrow \mathsf{val}\ v_2;\ x_1 \leftarrow \mathsf{val}\ v_1;\ \mathsf{ret}\ (x_1, x_2)]$ for values $v_1 \in \llbracket \tau_1 \rrbracket$ and $v_2 \in \llbracket \tau_2 \rrbracket$
- $1[x_1 \leftarrow \mathsf{val}\ v_1;\ x_2 \leftarrow \mathsf{val}\ v_2;\ \mathsf{ret}\ (x_1, x_2)] \sim 1[x_1 \leftarrow \mathsf{val}\ v_1;\ \mathsf{ret}\ (x_1, v_2)]$ for values $v_1 \in \llbracket \tau_1 \rrbracket$ and $v_2 \in \llbracket \tau_2 \rrbracket$
- $1[\mathsf{val}\ v_1 v_2] \sim 1[\mathsf{val}\ v_1 v_2]$ for values $v_1 \in \llbracket \tau_1 \rrbracket$ and $v_2 \in \llbracket \tau_2 \rrbracket$

□

At last we get to the protocol level. A protocol $\Delta \vdash P : I \to O$ is said to have value $v \in \llbracket \tau \rrbracket$ on channel $o \in O$ with $o : \tau \in \Delta$ if P contains the assignment $o := v$; otherwise we define the value to be $\bot$. We write $P|_{\mathsf{val}(o)} = v_\bot$ to indicate that the value of P on o is $v_\bot \in \{\bot\} \cup \llbracket \tau \rrbracket$, and lift this notation to distributions in the obvious way. In the case when $P|_{\mathsf{val}(o)} = \bot$, P contains the assignment $o := R$ for some reaction $\Delta;\ \cdot \vdash R : I \to \tau$. The value of R will be called the *local value* of P at o (this terminology indicates that the computation of the channel o to v has not yet been communicated to the rest of the protocol). We write $P|^{\mathsf{react}}_{\mathsf{val}(o)} = v_\bot$ to indicate that the local value of P on o is $v_\bot \in \{\bot\} \cup \llbracket \tau \rrbracket$ (therefore $R\,|_{\mathsf{val}} = v_\bot$), and lift this notation to distributions η in the obvious way. Thus, $\eta|^{\mathsf{react}}_{\mathsf{val}(o)} = v$ indicates that each protocol in the support of η contains the assignment $o := \mathsf{val}\ v$, whereas $\eta|^{\mathsf{react}}_{\mathsf{val}(o)} = \bot$ indicates that each protocol in the support of η carries a stuck reaction on o.

A protocol bisimulation is entirely analogous to a reaction bisimulation, except we require the valuation property to hold: *i)* per output channel o, and *ii)* for all distributions (not necessarily final).

Definition 7 (Protocol bisimulation)**.** *A* protocol bisimulation $\sim$ *is a binary relation on distributions on protocols of type* $\Delta \vdash I \to O$ *satisfying the following conditions:*

- Closure under convex combinations*: For any distributions* $\eta_1 \sim \varepsilon_1$ *and* $\eta_2 \sim \varepsilon_2$*, and any coefficients* $c_1, c_2 > 0$ *with* $c_1 + c_2 = 1$*, we have* $(c_1 * \eta_1 + c_2 * \eta_2) \sim (c_1 * \varepsilon_1 + c_2 * \varepsilon_2)$*.*
- Closure under input assignment*: For any distributions* $\eta \sim \varepsilon$*, input channel* $i \in I$ *with* $i : \tau \in \Delta$*, and value* $v \in \llbracket \tau \rrbracket$*, we have* $\eta[\mathsf{read}\ i := \mathsf{val}\ v] \sim \varepsilon[\mathsf{read}\ i := \mathsf{val}\ v]$*.*
- Closure under computation*: For any distributions* $\eta \sim \varepsilon$*, we have* $(\eta\Downarrow) \sim (\varepsilon\Downarrow)$*.*
- Valuation property*: For any output channel* $o \in O$ *with* $o : \tau \in \Delta$ *and any distributions* $\eta \sim \varepsilon$*, there exists a joint convex combination*

  $$\eta = \sum_i c_i * \eta_i \ \sim\ \sum_i c_i * \varepsilon_i = \varepsilon$$

  with $c_i > 0$ *and* $\sum_i c_i = 1$ *such that*

- *the respective components $\eta_i \sim \varepsilon_i$ are again related, and*
- $\eta_i|_{\mathsf{val}(o)} = v_\perp = \varepsilon_j|_{\mathsf{val}(o)}$ *for the same $v_\perp \in \{\perp\} \cup [\![\tau]\!]$ if and only if $i = j$.*

We have the analogous results for bisimulations on the protocol level:

Lemma 13. *We have the following:*

- *The identity relation is a protocol bisimulation.*
- *The inverse of a protocol bisimulation is a protocol bisimulation.*
- *The composition of two protocol bisimulations is a protocol bisimulation.*

Definition 8. *Let $\sim$ be an arbitrary binary relation on distributions on protocols of type $\Delta \vdash I \to O$. The* expansion *$\mathcal{L}(\sim)$ is the closure of $\sim$ under joint convex combinations. Explicitly, $\mathcal{L}(\sim)$ is defined by*

$$\Big(\sum_i c_i * \eta_i\Big)\ \mathcal{L}(\sim)\ \Big(\sum_i c_i * \varepsilon_i\Big)$$

for coefficients $c_i > 0$ with $\sum_i c_i = 1$ and distributions $\eta_i \sim \varepsilon_i$.

Lemma 14. *Let $\sim$ be a binary relation on distributions on protocols of type $\Delta \vdash I \to O$ with the following properties:*

- Closure under input assignment*: For any distributions $\eta \sim \varepsilon$, input channel $i \in I$ with $i : \tau \in \Delta$, and value $v \in [\![\tau]\!]$, we have $\eta[\mathsf{read}\ i := \mathsf{val}\ v] \sim \varepsilon[\mathsf{read}\ i := \mathsf{val}\ v]$.*
- Expansion closure under computation*: For any distributions $\eta \sim \varepsilon$, we have $(\eta\Downarrow)\ \mathcal{L}(\sim)\ (\varepsilon\Downarrow)$.*
- Valuation property*: For any output channel $o \in O$ with $o : \tau \in \Delta$ and any distributions $\eta \sim \varepsilon$, there exists a joint convex combination*

  $$\eta = \sum_i c_i * \eta_i\ \sim\ \sum_i c_i * \varepsilon_i = \varepsilon$$

  with $c_i > 0$ and $\sum_i c_i = 1$ such that

  - *the respective components $\eta_i \sim \varepsilon_i$ are again related, and*
  - $\eta_i|_{\mathsf{val}(o)} = v_\perp = \varepsilon_j|_{\mathsf{val}(o)}$ *for the same $v_\perp \in \{\perp\} \cup [\![\tau]\!]$ if and only if $i = j$.*

Then the expansion $\mathcal{L}(\sim)$ is a protocol bisimulation.

We now formally state what it means for exact protocol equality to be sound:

Definition 9. *An axiom $\Delta \vdash P_1 = P_2 : I \to O$ is* sound *if there is a protocol bisimulation $\sim$ such that* $1[P_1] \sim 1[P_2]$.

The ambient exact IPDL theory $\mathbb{T}_{\mathsf{prot}}$ for protocols is said to be sound if each of its axioms is sound. We now show that this implies overall soundness for exact equality:

Lemma 15 (Soundness of exact equality of protocols)**.** *If the ambient (exact) IPDL theories for expressions, reactions, and protocols are sound, then for any protocols $\Delta \vdash P_1 = P_2 : I \to O$ there exists a protocol bisimulation $\sim$ such that $1[P_1] \sim 1[P_2]$.*

Proof. We first replace the rules FOLD-IF-LEFT and FOLD-IF-RIGHT by the equivalent formulation in Figure **??**. We now proceed by induction on this alternative set of rules for exact protocol equality. We will freely use a distribution in place of a reaction (rule CONG-REACT) or a protocol (rules EMBED, ABSORB-LEFT) to indicate the obvious lifting of the corresponding construct to distributions on protocols.

- REFL: Our desired bisimulation is the identity relation.
- SYM: Our desired bisimulation is the inverse of the bisimulation obtained from the premise.
- TRANS: Our desired bisimulation is the composition of the two bisimulations obtained from the two premises.

- AXIOM: Our desired bisimulation is precisely the bisimulation obtained from the soundness of the axiom.
- INPUT-UNUSED: Our desired bisimulation is precisely the bisimulation obtained from the premise, seen as a bisimulation on distributions on protocols with the additional input i.
- EMBED: Let $\sim$ be the bisimulation obtained from the premise. Our desired bisimulation $\sim_\phi$ is defined by
  - $\phi^\star(\eta) \sim_\phi \phi^\star(\varepsilon)$ if $\eta \sim \varepsilon$
- CONG-REACT: Let $\sim$ be the reaction bisimulation obtained from the premise. Our desired bisimulation is the expansion of the relation $\sim_{\mathsf{react}}$ defined by
  - $(o \coloneqq \eta) \sim_{\mathsf{react}} (o \coloneqq \eta')$ for distributions $\eta \sim \eta'$
  - $1[o \coloneqq v] \sim_{\mathsf{react}} 1[o \coloneqq v]$ for value $v \in [\![\tau]\!]$
- CONG-COMP-LEFT: Let $\sim$ be the bisimulation obtained from the premise. The expansion of the relation $\sim_{\mathsf{par}}$ defined by
  - $(\eta \parallel Q) \sim_{\mathsf{par}} (\eta' \parallel Q)$ for $\eta \sim \eta'$ and protocol $\Delta \vdash Q : I \cup O_1 \to O_2$

  is the natural candidate for our desired bisimulation. Proving that this is indeed a bisimulation requires a fair amount of work (see Lemma **??**).
- CONG-NEW: Let $\sim$ be the bisimulation obtained from the premise. Our desired bisimulation $\sim_{\mathsf{new}}$ is defined by
  - $\big(\mathsf{new}\ o : \tau\ \mathsf{in}\ \eta\big) \sim_{\mathsf{new}} \big(\mathsf{new}\ o : \tau\ \mathsf{in}\ \eta'\big)$ if $\eta \sim \eta'$
- COMP-COMM: Our desired bisimulation is the expansion of the relation $\sim$ defined by
  - $1[P_1 \parallel P_2] \sim 1[P_2 \parallel P_1]$ for protocols $\Delta \vdash P_1 : I \cup O_2 \to O_1$ and $\Delta \vdash P_2 : I \cup O_1 \to O_2$
- COMP-ASSOC: Our desired bisimulation is the expansion of the relation $\sim$ defined by
  - $1\big[(P_1 \parallel P_2) \parallel P_3\big] \sim 1\big[P_1 \parallel (P_2 \parallel P_3)\big]$ for
    - protocol $\Delta \vdash P_1 : I \cup O_2 \cup O_3 \to O_1$
    - protocol $\Delta \vdash P_2 : I \cup O_1 \cup O_3 \to O_2$
    - protocol $\Delta \vdash P_3 : I \cup O_1 \cup O_2 \to O_3$
- NEW-EXCH: The desired bisimulation is the expansion of the relation $\sim$ defined by
  - $1[\mathsf{new}\ o_1 : \tau_1\ \mathsf{in}\ \mathsf{new}\ o_2 : \tau_2\ \mathsf{in}\ P] \sim 1[\mathsf{new}\ o_2 : \tau_2\ \mathsf{in}\ \mathsf{new}\ o_1 : \tau_1\ \mathsf{in}\ P]$ for
    - protocol $\Delta, o_1 : \tau_1, o_2 : \tau_2 \vdash P : I \to O \cup \{o_1, o_2\}$
- COMP-NEW: Our desired bisimulation is the expansion of the relation $\sim$ defined by
  - $1[P \parallel (\mathsf{new}\ o : \tau\ \mathsf{in}\ Q)] \sim 1[\mathsf{new}\ o : \tau\ \mathsf{in}\ (P \parallel Q)]$ for
    - protocol $\Delta \vdash P : I \cup O_2 \to O_1$
    - protocol $\Delta, o : \tau \vdash Q : I \cup O_1 \to O_2 \cup \{o\}$
- ABSORB-LEFT: Our desired bisimulation is the expansion of the relation $\sim$ defined by
  - $1[P \parallel Q] \sim 1[P]$ for protocols $\Delta \vdash P : I \to O$ and $\Delta \vdash Q : I \cup O \to \varnothing$
- DIVERGE: Our desired bisimulation is the expansion of the relation $\sim$ defined by
  - $1[o \coloneqq x \leftarrow \mathsf{read}\ o;\ R] \sim 1[o \coloneqq \mathsf{read}\ o]$ for reaction $\Delta;\ \cdot \vdash R : I \to \tau$
- FOLD-IF-LEFT: Our desired bisimulation is the expansion of the relation $\sim$ defined by

  - $1[\mathsf{new}\ l : \tau\ \mathsf{in}\ o \coloneqq x \leftarrow \mathsf{read}\ b;\ \mathsf{if}\ x\ \mathsf{then}\ \mathsf{read}\ l\ \mathsf{else}\ S_2 \parallel l \coloneqq x \leftarrow \mathsf{read}\ b;\ S_1] \sim$ $1[o \coloneqq x \leftarrow \mathsf{read}\ b;\ \mathsf{if}\ x\ \mathsf{then}\ S_1\ \mathsf{else}\ S_2]$ for
    * reaction $\Delta;\ \cdot \vdash S_1 : I \to \tau$
    * reaction $\Delta;\ \cdot \vdash S_2 : I \to \tau$
  - $1[\mathsf{new}\ l : \tau\ \mathsf{in}\ o \coloneqq x \leftarrow \mathsf{val}\ v;\ \mathsf{if}\ x\ \mathsf{then}\ \mathsf{read}\ l\ \mathsf{else}\ S_2 \parallel l \coloneqq x \leftarrow \mathsf{val}\ v;\ S_1] \sim$ $1[o \coloneqq x \leftarrow \mathsf{val}\ v;\ \mathsf{if}\ x\ \mathsf{then}\ S_1\ \mathsf{else}\ S_2]$ for
    * value $v \in \{0, 1\}$
    * reaction $\Delta;\ \cdot \vdash S_1 : I \to \tau$
    * reaction $\Delta;\ \cdot \vdash S_2 : I \to \tau$
  - $1[\mathsf{new}\ l : \tau\ \mathsf{in}\ o \coloneqq \mathsf{read}\ l \parallel l \coloneqq S_1] \sim 1[o \coloneqq S_1]$ for reaction $\Delta;\ \cdot \vdash S_1 : I \to \tau$
  - $1[\mathsf{new}\ l : \tau\ \mathsf{in}\ o \coloneqq S_2 \parallel l \coloneqq S_1] \sim 1[o \coloneqq S_2]$ for reactions $\Delta;\ \cdot \vdash S_1 : I \to \tau$ and $\Delta;\ \cdot \vdash S_2 : I \to \tau$
  - $1[\mathsf{new}\ l : \tau\ \mathsf{in}\ o \coloneqq v_2 \parallel l \coloneqq S_1] \sim 1[o \coloneqq v_2]$ for reaction $\Delta;\ \cdot \vdash S_1 : I \to \tau$ and value $v_2 \in [\![\tau]\!]$
  - $1[\mathsf{new}\ l : \tau\ \mathsf{in}\ o \coloneqq S_2 \parallel l \coloneqq v_1] \sim 1[o \coloneqq S_2]$ for value $v_1 \in [\![\tau]\!]$ and reaction $\Delta;\ \cdot \vdash S_2 : I \to \tau$
  - $1[\mathsf{new}\ l : \tau\ \mathsf{in}\ o \coloneqq v_2 \parallel l \coloneqq v_1] \sim 1[o \coloneqq v_2]$ for values $v_1, v_2 \in [\![\tau]\!]$

- FOLD-IF-RIGHT: Our desired bisimulation is the expansion of the relation $\sim$ defined by
  - $1[\mathsf{new}\ r : \tau\ \mathsf{in}\ o \coloneqq x \leftarrow \mathsf{read}\ b;\ \mathsf{if}\ x\ \mathsf{then}\ S_1\ \mathsf{else}\ \mathsf{read}\ r \parallel r \coloneqq x \leftarrow \mathsf{read}\ b;\ S_2] \sim$ $1[o \coloneqq x \leftarrow \mathsf{read}\ b;\ \mathsf{if}\ x\ \mathsf{then}\ S_1\ \mathsf{else}\ S_2]$ for
    * reaction $\Delta;\ \cdot \vdash S_1 : I \to \tau$
    * reaction $\Delta;\ \cdot \vdash S_2 : I \to \tau$
  - $1[\mathsf{new}\ r : \tau\ \mathsf{in}\ o \coloneqq x \leftarrow \mathsf{val}\ v;\ \mathsf{if}\ x\ \mathsf{then}\ S_1\ \mathsf{else}\ \mathsf{read}\ r \parallel r \coloneqq x \leftarrow \mathsf{val}\ v;\ S_2] \sim$ $1[o \coloneqq x \leftarrow \mathsf{val}\ v;\ \mathsf{if}\ x\ \mathsf{then}\ S_1\ \mathsf{else}\ S_2]$ for
    * value $v \in \{0, 1\}$
    * reaction $\Delta;\ \cdot \vdash S_1 : I \to \tau$
    * reaction $\Delta;\ \cdot \vdash S_2 : I \to \tau$
  - $1[\mathsf{new}\ r : \tau\ \mathsf{in}\ o \coloneqq \mathsf{read}\ r \parallel r \coloneqq S_2] \sim 1[o \coloneqq S_2]$ for reaction $\Delta;\ \cdot \vdash S_2 : I \to \tau$
  - $1[\mathsf{new}\ r : \tau\ \mathsf{in}\ o \coloneqq S_1 \parallel r \coloneqq S_2] \sim 1[o \coloneqq S_1]$ for reactions $\Delta;\ \cdot \vdash S_1 : I \to \tau$ and $\Delta;\ \cdot \vdash S_2 : I \to \tau$
  - $1[\mathsf{new}\ r : \tau\ \mathsf{in}\ o \coloneqq S_1 \parallel r \coloneqq v_2] \sim 1[o \coloneqq S_1]$ for reaction $\Delta;\ \cdot \vdash S_1 : I \to \tau$ and value $v_2 \in [\![\tau]\!]$
  - $1[\mathsf{new}\ r : \tau\ \mathsf{in}\ o \coloneqq v_1 \parallel r \coloneqq S_2] \sim 1[o \coloneqq v_1]$ for value $v_1 \in [\![\tau]\!]$ and reaction $\Delta;\ \cdot \vdash S_2 : I \to \tau$
  - $1[\mathsf{new}\ r : \tau\ \mathsf{in}\ o \coloneqq v_1 \parallel r \coloneqq v_2] \sim 1[o \coloneqq v_1]$ for values $v_1, v_2 \in [\![\tau]\!]$

- FOLD-BIND: Our desired bisimulation is the expansion of the relation $\sim$ defined by
  - $1[\mathsf{new}\ c : \sigma\ \mathsf{in}\ o \coloneqq x \leftarrow \mathsf{read}\ c;\ S \parallel c \coloneqq R] \sim 1[o \coloneqq x \leftarrow R;\ S]$ for
    * reaction $\Delta;\ \cdot \vdash R : I \to \sigma$
    * reaction $\Delta;\ x : \sigma \vdash S : I \to \tau$
  - $1[\mathsf{new}\ c : \sigma\ \mathsf{in}\ o \coloneqq S \parallel c \coloneqq u] \sim 1[o \coloneqq S]$ for value $u \in [\![\sigma]\!]$ and reaction $\Delta;\ \cdot \vdash S : I \to \tau$
  - $1[\mathsf{new}\ c : \sigma\ \mathsf{in}\ o \coloneqq v \parallel c \coloneqq u] \sim 1[o \coloneqq u]$ for values $u \in [\![\sigma]\!]$ and $v \in [\![\tau]\!]$

- SUBST: Let $\sim$ be the reaction bisimulation obtained from the premise

  $$\Delta;\ \cdot \vdash \big(x_1 \leftarrow R_1;\ x_1' \leftarrow R_1;\ \mathsf{ret}\ (x_1, x_1')\big) = \big(x_1 \leftarrow R_1;\ \mathsf{ret}\ (x_1, x_1)\big) : I \to \tau_1 \times \tau_1$$

  Our desired bisimulation is the expansion of the relation $\sim_{\mathsf{subst}}$ defined by
  - $\big(o_1 \coloneqq \eta \parallel o_2 \coloneqq x_1 \leftarrow \mathsf{read}\ o_1;\ R_2\big) \sim_{\mathsf{subst}} \big(o_1 \coloneqq \eta \parallel o_2 \coloneqq x_1 \leftarrow \eta;\ R_2\big)$ for
    * distribution η on reactions $\Delta;\ \cdot \vdash R_1 : I \to \tau_1$
    * reaction $\Delta;\ \cdot \vdash R_1 : I \to \tau_1$ such that $R_1{\Downarrow} = \eta{\Downarrow}$

$$\frac{b \neq o \qquad b \in I \qquad b : \mathsf{Bool}, o : \tau \in \Delta \qquad \Delta;\ \cdot \vdash S_1 : I \to \tau \qquad \Delta;\ \cdot \vdash S_2 : I \to \tau}{\begin{array}{c}\Delta \vdash \big(\mathsf{new}\ l : \tau\ \mathsf{in}\ o := x \leftarrow \mathsf{read}\ b;\ \mathsf{if}\ x\ \mathsf{then}\ \mathsf{read}\ l\ \mathsf{else}\ S_2 \parallel l := x \leftarrow \mathsf{read}\ b;\ S_1\big) = \\ \big(o := x \leftarrow \mathsf{read}\ b;\ \mathsf{if}\ x\ \mathsf{then}\ S_1\ \mathsf{else}\ S_2\big) : I \setminus \{o\} \to \{o\}\end{array}}\ \textsc{fold-if-left}$$

$$\frac{b \neq o \qquad b \in I \qquad b : \mathsf{Bool}, o : \tau \in \Delta \qquad \Delta;\ \cdot \vdash S_1 : I \to \tau \qquad \Delta;\ \cdot \vdash S_2 : I \to \tau}{\begin{array}{c}\Delta \vdash \big(\mathsf{new}\ r : \tau\ \mathsf{in}\ o := x \leftarrow \mathsf{read}\ b;\ \mathsf{if}\ x\ \mathsf{then}\ S_1\ \mathsf{else}\ \mathsf{read}\ r \parallel r := x \leftarrow \mathsf{read}\ b;\ S_2\big) = \\ \big(o := x \leftarrow \mathsf{read}\ b;\ \mathsf{if}\ x\ \mathsf{then}\ S_1\ \mathsf{else}\ S_2\big) : I \setminus \{o\} \to \{o\}\end{array}}\ \textsc{fold-if-right}$$

Figure 21: Alternative formulation of the FOLD-IF-LEFT and FOLD-IF-RIGHT rules.

    - reaction $\Delta;\ x_1 : \tau_1 \vdash R_2 : I \to \tau_2$

  such that $1[x_1 \leftarrow R_1;\ x_1' \leftarrow R_1;\ \mathsf{ret}\ (x_1, x_1')] \sim 1[x_1 \leftarrow R_1;\ \mathsf{ret}\ (x_1, x_1)]$

  - $1[o_1 := v_1 \parallel o_2 := R_2] \sim_{\mathsf{subst}} 1[o_1 := v_1 \parallel o_2 := R_2]$ for value $v_1 \in [\![\tau_1]\!]$ and reaction $\Delta;\ \cdot \vdash R_2 : I \to \tau_2$
  - $1[o_1 := v_1 \parallel o_2 := v_2] \sim_{\mathsf{subst}} 1[o_1 := v_1 \parallel o_2 := v_2]$ for values $v_1 \in [\![\tau_1]\!]$ and $v_2 \in [\![\tau_2]\!]$

- DROP: Let $\sim$ be the reaction bisimulation obtained from the premise

$$\Delta;\ \cdot \vdash \big(x_1 \leftarrow R_1;\ R_2\big) = R_2 : I \to \tau_2$$

  Our desired bisimulation is the expansion of the relation $\sim_{\mathsf{drop}}$ defined by

  - $\big(o_1 := \eta_1 \parallel o_2 := x_1 \leftarrow \mathsf{read}\ o_1;\ R_2\big) \sim_{\mathsf{drop}} \big(o_1 := \eta_1 \parallel o_2 := \eta_2\big)$ for
    - distribution η_1 on reactions $\Delta;\ \cdot \vdash R_1 : I \to \tau_1$
    - reaction $\Delta;\ \cdot \vdash R_1 : I \to \tau_1$ such that either

      i) $R_1\Downarrow = \eta_1\Downarrow$, or

      ii) $R_1\Downarrow = c_1(\eta_1\Downarrow) + c_2\mu$ for some distribution μ and some $c_1, c_2 > 0$ with $c_1 + c_2 = 1$
    - distribution η_2 on reactions $\Delta;\ \cdot \vdash R_2 : I \to \tau_2$
    - reaction $\Delta;\ \cdot \vdash R_2 : I \to \tau_2$ such that $R_2\Downarrow = \eta_2\Downarrow$

    such that $1[x_1 \leftarrow R_1;\ R_2] \sim 1[R_2]$
  - $1[o_1 := v_1 \parallel o_2 := R_2] \sim_{\mathsf{drop}} 1[o_1 := v_1 \parallel o_2 := R_2]$ for value $v_1 \in [\![\tau_1]\!]$ and reaction $\Delta;\ \cdot \vdash R_2 : I \to \tau_2$
  - $1[o_1 := v_1 \parallel o_2 := v_2] \sim_{\mathsf{drop}} 1[o_1 := v_1 \parallel o_2 := v_2]$ for values $v_1 \in [\![\tau_1]\!]$ and $v_2 \in [\![\tau_2]\!]$

□

The remainder of this section is devoted to proving the following lemma:

Lemma 16 (Composability for bisimulations)**.** *Let $\sim$ be a bisimulation on protocols of type $\Delta \vdash I \cup O_2 \to O_1$. Then the expansion of the relation $\sim_{\mathsf{par}}$ defined by*

- *$(\eta \parallel Q) \sim_{\mathsf{par}} (\eta' \parallel Q)$ for $\eta \sim \eta'$ and protocol $\Delta \vdash Q : I \cup O_1 \to O_2$*

is again a protocol bisimulation.

The one property hard to verify is expansion closure under computation: *for any protocol $\Delta \vdash Q : I \cup O_1 \to O_2$ and any distributions $\eta \sim \eta'$, we have $\big(\eta \parallel Q\big)\Downarrow\ \mathcal{L}(\sim_{\mathsf{par}})\ \big(\eta' \parallel Q\big)\Downarrow$.* The difficulty arises from the global nature of the protocol semantics: in the composition $P \parallel Q$, a step of the form $P \xmapsto{o := v} P'$ changes the protocol Q (specifically to $Q[\mathsf{read}\ o := \mathsf{val}\ v]$). This makes it hard to express the computation of $P \parallel Q$ in terms of the computation of P, because in the course of the latter we are simultaneously probabilistically updating Q.

We solve this problem by defining an alternate *local* form of operational semantics for IPDL protocols, where we can use local values (that is, an assignment of the form $o := \mathsf{val}\ v$ rather than $o := v$) to compute in P without having to update Q at the same time. The relation $P \xhookrightarrow{o\,:=\,v} Q$ ("*pull*") for protocols formalizes this notion.

$$\frac{}{\big(o := \mathsf{val}\ v\big) \xhookrightarrow{o\,:=\,v} \big(o := \mathsf{val}\ v\big)}\ \textsc{pull-react} \qquad \frac{P \xhookrightarrow{o\,:=\,v} P'}{P \parallel Q \xhookrightarrow{o\,:=\,v} P' \parallel Q[\mathsf{read}\ o := \mathsf{val}\ v]}\ \textsc{pull-comp-left}$$

$$\frac{Q \xhookrightarrow{o\,:=\,v} Q'}{P \parallel Q \xhookrightarrow{o\,:=\,v} P[\mathsf{read}\ o := \mathsf{val}\ v] \parallel Q'}\ \textsc{pull-comp-right} \qquad \frac{P \xhookrightarrow{o\,:=\,v} P' \qquad o \neq c}{\big(\mathsf{new}\ c : \tau\ \mathsf{in}\ P\big) \xhookrightarrow{o\,:=\,v} \big(\mathsf{new}\ c : \tau\ \mathsf{in}\ P'\big)}\ \textsc{pull-new}$$

The only difference between $P \xhookrightarrow{o\,:=\,v} Q$ and $P \xmapsto{o\,:=\,v} Q$ is that the latter turns the assignment $o := \mathsf{val}\ v$ into $o := v$. We extract this simple extra step into the dual relation $P \downarrow_{o\,:=\,v} Q$ ("*local assign*"), which, crucially, does not involve any manipulation of reads.

$$\frac{}{\big(o := \mathsf{val}\ v\big) \downarrow_{o\,:=\,v} \big(o := v\big)}\ \textsc{loc-assign-react} \qquad \frac{P \downarrow_{o\,:=\,v} P'}{\big(P \parallel Q\big) \downarrow_{o\,:=\,v} \big(P' \parallel Q\big)}\ \textsc{loc-assign-comp-left}$$

$$\frac{Q \downarrow_{o\,:=\,v} Q'}{\big(P \parallel Q\big) \downarrow_{o\,:=\,v} \big(P \parallel Q'\big)}\ \textsc{loc-assign-comp-right} \qquad \frac{P \downarrow_{o\,:=\,v} P' \qquad o \neq c}{\big(\mathsf{new}\ c : \tau\ \mathsf{in}\ P\big) \downarrow_{o\,:=\,v} \big(\mathsf{new}\ c : \tau\ \mathsf{in}\ P'\big)}\ \textsc{loc-assign-new}$$

The big-step form $P \Rightarrow \eta$ of our local operational semantics strings together a sequence of internal and pull steps.

$$\frac{}{P \Rightarrow 1[P]} \qquad \frac{P \xhookrightarrow{o\,:=\,v} Q \qquad Q \Rightarrow \eta}{P \Rightarrow \eta}$$

$$\frac{P \to \sum_i c_i * 1[P_i] \qquad P_i \Rightarrow \eta_i}{P \Rightarrow \sum_i c_i * \eta_i}$$

To bridge the gap between the local and the global versions of our operational semantics, we use the big-step form $P \mapsdown Q$, which strings together a sequence of local assign steps *that coincide with output steps.*

$$\frac{}{P \mapsdown P} \qquad \frac{P \downarrow_{o\,:=\,v} P' \qquad P \xmapsto{o\,:=\,v} P' \qquad P' \mapsdown Q}{P \mapsdown Q}$$

So if $P \mapsdown Q$ then Q is a computation of P that has been obtained chiefly by performing local assign steps.

Lemma 17 (Lifting)**.** *For any protocol $\Delta \vdash P : I \to O$ we have the following:*

- *If $P \downarrow_{o_1\,:=\,v_1} Q$ and $Q \xmapsto{o_2\,:=\,v_2} Q'$ then there is P' such that $P \xmapsto{o_2\,:=\,v_2} P'$ and $P' \downarrow_{o_1\,:=\,v_1} Q'$.*
- *If $P \downarrow_{o\,:=\,v} P'$ and $P' \to \eta'$ then there is η such that $P \to \eta$ and $\eta \downarrow_{o\,:=\,v} \eta'$.*
- *If $P \downarrow_{o_1\,:=\,v_1} Q$ and $P \xmapsto{o_1\,:=\,v_1} Q$, and $Q \downarrow_{o_2\,:=\,v_2} Q'$ and $Q \xmapsto{o_2\,:=\,v_2} Q'$, then there is P' such that $P \downarrow_{o_2\,:=\,v_2} P'$ and $P \xmapsto{o_2\,:=\,v_2} P'$, and $P' \downarrow_{o_1\,:=\,v_1} Q'$ and $P' \xmapsto{o_1\,:=\,v_1} Q'$.*
- *If $P \mapsdown Q$ and $Q \xmapsto{o\,:=\,v} Q'$ then there is P' such that $P \xmapsto{o\,:=\,v} P'$ and $P' \mapsdown Q'$.*
- *If $P \mapsdown P'$ and $P' \to \eta'$ then there is η such that $P \to \eta$ and $\eta \mapsdown \eta'$.*

In the above lemma, we lift the relations $\downarrow_{o\,:=\,v}$ and $\mapsdown$ to distributions in the natural way.

Lemma 18 (Approximation)**.** *For any protocol $\Delta \vdash P : I \to O$ we have the following:*

- *If $P \xhookrightarrow{o:=v} P'$ then there is Q such that $P \xmapsto{o:=v} Q$ and $P' \xmapsto{o:=v} Q$.*

Lemma 19 (Factoring)**.** *For any protocol $\Delta \vdash P : I \to O$ we have the following:*

- *If $P \xmapsto{o:=v} Q$ then there is P' such that $P \xhookrightarrow{o:=v} P'$ and $P' \downarrow_{o:=v} Q$ and $P' \xmapsto{o:=v} Q$.*

Lemma 20 (Correctness)**.** *For any protocol $\Delta \vdash P : I \to O$ we have the following:*

- *If $P \Rightarrow \eta$ then $P{\Downarrow} = \eta{\Downarrow}$.*
- *If $P \mapsdown Q$ then $P{\Downarrow} = Q{\Downarrow}$.*

The following lemma expresses the computation of $P \parallel Q$ in terms of the local computation of P.

Lemma 21 (Composability for local semantics)**.** *For protocols $\Delta \vdash P : I \cup O_2 \to O_1$ and $\Delta \vdash Q : I \cup O_1 \to O_2$, if $P \Rightarrow \eta$ then $(P \parallel Q){\Downarrow} = \big(\eta \parallel Q\big){\Downarrow}$.*

Lemma 22 (Termination)**.** *For any protocol $\Delta \vdash P : I \to O$ there are distributions η and ε such that $P \Rightarrow \eta$ and $\eta \mapsdown \varepsilon$ and ε is final.*

Sketch. We generalize the statement. *Given any $n \in \mathbb{N}$, any protocol $\Delta \vdash P : I \to O$, and any protocol Q such that $P \mapsdown Q$ and $\|Q\|_{\mathsf{str}} = n$, there are distributions η and ε such that $P \Rightarrow \eta$ and $\eta \mapsdown \varepsilon$ and ε is final.* We prove this statement by induction on the structure bound n using the lifting and factoring lemmas. □

We now have all the preliminaries necessary to prove that $\sim_{\mathsf{par}}$ enjoys expansion closure under computation.

Proof. We proceed by induction on the structure bound of the protocol Q:

Claim 1 (Expansion closure under computation)**.** *Given any $n \in \mathbb{N}$, any protocol $\Delta \vdash Q : I \cup O_1 \to O_2$ such that $\|Q\|_{\mathsf{str}} = n$, and any distributions $\eta \sim \eta'$ on protocols of type $\Delta \vdash I \cup O_2 \to O_1$, we have $\big(\eta \parallel Q\big){\Downarrow}\ \mathcal{L}(\sim_{\mathsf{par}})\ \big(\eta' \parallel Q\big){\Downarrow}$.*

In the remainder of this section we will work with a fixed $n \in \mathbb{N}$. Since the set O_1 is finite, we can start off by successively applying the valuation property of the bisimulation $\sim$ to $\eta \sim \eta'$ for each output channel $o \in O_1$, until we end up with the special case when η and η' have the same value on each o. In other words, it suffices to prove the following:

Claim 2. *Given any protocol $\Delta \vdash Q : I \cup O_1 \to O_2$ such that $\|Q\|_{\mathsf{str}} = n$, and any distributions $\eta \sim \eta'$ on protocols of type $\Delta \vdash I \cup O_2 \to O_1$ such that*

- *for each channel $o \in O_1$ with $o : \tau \in \Delta$ we have $\eta|_{\mathsf{val}(o)} = v_\perp = \eta'|_{\mathsf{val}(o)}$ for some $v_\perp \in \{\perp\} \cup [\![\tau]\!]$,*

we have $\big(\eta \parallel Q\big){\Downarrow}\ \mathcal{L}(\sim_{\mathsf{par}})\ \big(\eta' \parallel Q\big){\Downarrow}$.

By performing internal and pull steps on the distributions η and η', we can approximate their computations without changing any channel valuations. The resulting distributions will not be necessarily related by $\sim$ but that's okay: their computations will again be related, as these coincide with the computations of η and η', respectively. Specifically, by the correctness, composability, and termination lemmas for our local semantics it suffices to prove the following:

Claim 3. *Given any protocol $\Delta \vdash Q : I \cup O_1 \to O_2$ such that $\|Q\|_{\mathsf{str}} = n$, any distributions η and η' on protocols of type $\Delta \vdash I \cup O_2 \to O_1$ such that*

- *for each channel $o \in O_1$ with $o : \tau \in \Delta$ we have $\eta|_{\mathsf{val}(o)} = v_\perp = \eta'|_{\mathsf{val}(o)}$ for some $v_\perp \in \{\perp\} \cup [\![\tau]\!]$,*

and any final distributions $\varepsilon \sim \varepsilon'$ such that $\eta \mapsdown \varepsilon$ and $\eta' \mapsdown \varepsilon'$, we have $\big(\eta \parallel Q\big){\Downarrow}\ \mathcal{L}(\sim_{\mathsf{par}})\ \big(\eta' \parallel Q\big){\Downarrow}$.

We now establish an analogue of the valuation property for the local semantics of protocols:

Claim 4 (Local valuation property)**.** *For any channel $o \in O_1$ with $o : \tau \in \Delta$, any distributions η and η' on protocols of type $\Delta \vdash I \cup O_2 \to O_1$ such that $\eta|_{\mathsf{val}(o)} = \perp = \eta'|_{\mathsf{val}(o)}$, and any final distributions $\varepsilon \sim \varepsilon'$ such that $\eta \mapsdown \varepsilon$ and $\eta' \mapsdown \varepsilon'$, there exist convex combinations*

$$\eta = \sum_i c_i * \eta_i, \quad \eta' = \sum_i c_i * \eta_i', \quad \varepsilon = \sum_i c_i * \varepsilon_i, \quad \varepsilon' = \sum_i c_i * \varepsilon_i'$$

with $c_i > 0$ and $\sum_i c_i = 1$ such that

- $\eta_i \mapsdown \varepsilon_i$ *and* $\eta'_i \mapsdown \varepsilon'_i$,
- *the respective components* $\varepsilon_i \sim \varepsilon'_i$ *are again related, and*
- $\eta_i|^{\mathsf{react}}_{\mathsf{val}(o)} = v_\perp = \eta'_j|^{\mathsf{react}}_{\mathsf{val}(o)}$ *for the same* $v_\perp \in \{\perp\} \cup [\![\tau]\!]$ *if and only if* $i = j$.

The local valuation property follows easily from the valuation property of the bisimulation $\sim$ and the fact that the relation $P \mapsdown Q$ turns local values on $o \in O$ into global ones: if $P|^{\mathsf{react}}_{\mathsf{val}(o)} = v_\perp$ and Q is final then $Q|_{\mathsf{val}(o)} = v_\perp$.

We can now carry out the analogous argument from earlier but for local valuation: since the set O_1 of output channels is finite, we can successively apply the local valuation property to η, η' for each output channel o where $\eta|_{\mathsf{val}(o)} = \perp = \eta'|_{\mathsf{val}(o)}$, until we end up with the special case when η and η' have the same local value on each such o. In other words, it suffices to prove the following:

Claim 5. *Given any protocol* $\Delta \vdash Q : I \cup O_1 \to O_2$ *such that* $\|Q\|_{\mathsf{str}} = n$*, any distributions* η *and* η' *on protocols of type* $\Delta \vdash I \cup O_2 \to O_1$ *such that*

- *for each channel* $o \in O_1$ *with* $o : \tau \in \Delta$ *we have* $\eta|_{\mathsf{val}(o)} = v_\perp = \eta'|_{\mathsf{val}(o)}$ *for some* $v_\perp \in \{\perp\} \cup [\![\tau]\!]$*, and*
  - *in the case that* $\eta|_{\mathsf{val}(o)} = \perp = \eta'|_{\mathsf{val}(o)}$*, we have* $\eta|^{\mathsf{react}}_{\mathsf{val}(o)} = v_\perp = \eta'|^{\mathsf{react}}_{\mathsf{val}(o)}$ *for some* $v_\perp \in \{\perp\} \cup [\![\tau]\!]$*,*

and any final distributions $\varepsilon \sim \varepsilon'$ *such that* $\eta \mapsdown \varepsilon$ *and* $\eta' \mapsdown \varepsilon'$*, we have* $\big(\eta \parallel Q\big)\Downarrow \;\mathcal{L}(\sim_{\mathsf{par}})\; \big(\eta' \parallel Q\big)\Downarrow$.

To prove the latest claim, we let η and η' step simultaneously on the distinct output channels $o_1, \ldots, o_n \in O_1$ as follows, where the lifting lemma guarantees that the order in which we execute these steps is immaterial:

- $\eta = \mu_0 \downarrow_{o_1 := v_1} \mu_1 \downarrow_{o_2 := v_2} \cdots \downarrow_{o_{n-1} := v_{n-1}} \mu_{n-1} \downarrow_{o_n := v_n} \mu_n = \varepsilon$,
- $\eta = \mu_0 \xmapsto{o_1 := v_1} \mu_1 \xmapsto{o_2 := v_2} \ldots \xmapsto{o_{n-1} := v_{n-1}} \mu_{n-1} \xmapsto{o_n := v_n} \mu_n = \varepsilon$, and
- $\eta' = \mu'_0 \downarrow_{o_1 := v_1} \mu'_1 \downarrow_{o_2 := v_2} \cdots \downarrow_{o_{n-1} := v_{n-1}} \mu'_{n-1} \downarrow_{o_n := v_n} \mu'_n = \varepsilon'$,
- $\eta' = \mu'_0 \xmapsto{o_1 := v_1} \mu'_1 \xmapsto{o_2 := v_2} \ldots \xmapsto{o_{n-1} := v_{n-1}} \mu'_{n-1} \xmapsto{o_n := v_n} \mu'_n = \varepsilon'$.

The valuation assumptions on η and η', and consequently on μ_i and μ'_i, guarantee that the corresponding steps $\mu_i \xmapsto{o_{i+1} := v_{i+1}} \mu_{i+1}$ and $\mu'_i \xmapsto{o_{i+1} := v_{i+1}} \mu'_{i+1}$ exert the same effect on the common context, thereby yielding the same sequence $Q = Q_0, \ldots, Q_n$ of updates: we have

$$\begin{aligned} \mu_i \parallel Q_i &\xmapsto{o_{i+1} := v_{i+1}} \mu_{i+1} \parallel Q_i[\mathsf{read}\ o_{i+1} := \mathsf{val}\ v_{i+1}] \\ \mu'_i \parallel Q_i &\xmapsto{o_{i+1} := v_{i+1}} \mu'_{i+1} \parallel Q_i[\mathsf{read}\ o_{i+1} := \mathsf{val}\ v_{i+1}] \end{aligned}$$

and hence $Q_{i+1} := Q_i[\mathsf{read}\ o_{i+1} := \mathsf{val}\ v_{i+1}]$. It thus suffices to prove the following:

Claim 6. *Given any protocol* $\Delta \vdash Q : I \cup O_1 \to O_2$ *with* $\|Q\|_{\mathsf{str}} = n$*, and any final distributions* $\eta \sim \eta'$*, we have* $\big(\eta \parallel Q\big)\Downarrow \;\mathcal{L}(\sim_{\mathsf{par}})\; \big(\eta' \parallel Q\big)\Downarrow$.

At last we have the opportunity to use the induction hypothesis: either Q is itself final, in which case the claim follows at once from the definition of $\sim_{\mathsf{par}}$, or Q takes a step using one of the stepping relations $\to$ and $\xmapsto{o := v}$, in which case its structure bound becomes strictly smaller and the induction hypothesis applies. □

4 Computational Semantics of IPDL

When using Turing Machines to compute (probabilistic) functions, we only consider (probabilistic) Turing Machines that have a finite runtime $N \in \mathbb{N}$; *i.e.*, for every input in the domain, after N (probabilistic) transitions the TM ends up in a configuration where no further transitions are possible. That is, the TM has either reached an accepting state or it has halted after reading a symbol for which no transition is possible in the current state.

Intuitively, a family of interpretations is PPT (probabilistic polynomial-time) if it assigns polynomial lengths to type symbols t, and PPT-computable functions to function symbols f and distribution symbols d. A small caveat is that a random distribution on a subset $S \subseteq \{0,1\}^n$ of bitstrings is in general computable by a probabilistic Turing Machine only up to a small error ε, which is the probability that the TM does not end up in an accepting state. In effect, the TM computes a distribution μ on $S \cup \{\perp\}$ with $\mu(\perp) = \varepsilon$. To relate μ to our original distribution on S, we introduce the following:

Definition 10 (Approximating distributions)**.** *Let $S \subseteq \{0,1\}^n$ be a subset of bitstrings of a fixed length. We say that a distribution μ_1 on $S \cup \{\bot\}$* approximates *a distribution μ_2 on S with error $0 \leqslant \varepsilon \leqslant 1$ if there are distributions η_1, η_2 on S such that $\mu_1 = (1-\varepsilon)\eta_1 + \varepsilon 1[\bot]$ and $\mu_2 = (1-\varepsilon)\eta_1 + \varepsilon\eta_2$.*

We recall that a function $\varepsilon : \mathbb{N} \to \mathbb{Q}_{\geqslant 0}$ is negligible if it is eventually smaller than the inverse of any polynomial: *for any $n \in \mathbb{N}$, there exists $N \in \mathbb{N}$ such that for all $\lambda \geqslant N$ we have $\varepsilon(\lambda) \leqslant \frac{1}{\lambda^n}$.*

Definition 11 (PPT family of interpretations)**.** *Let Σ be an* IPDL *signature. A family $\{[\![-]\!]_\lambda\}_{\lambda \in \mathbb{N}}$ of interpretations for Σ is* probabilistic polynomial-time (PPT) *if there is a polynomial $p(\lambda)$, a negligible function $\eta(\lambda)$, and a natural number $N \in \mathbb{N}$ such that the following holds:*

- *For all type symbols t, $|\mathsf{t}|_\lambda \leqslant p(\lambda)$ if $\lambda \geqslant N$.*
- *For all function symbols $\mathsf{f} : \sigma \to \tau$, the function $[\![\mathsf{f}]\!]_\lambda$ from bitstrings $[\![\sigma]\!]_\lambda$ to bitstrings $[\![\tau]\!]_\lambda$ is computable by a deterministic Turing Machine TM_λ with symbols $\mathsf{0}, \mathsf{1}$. Both the number of states and the runtime of TM_λ are $\leqslant p(\lambda)$ if $\lambda \geqslant N$.*
- *For all distribution symbols $\mathsf{d} : \sigma \twoheadrightarrow \tau$, the function $[\![\mathsf{d}]\!]_\lambda$ from bitstrings $[\![\sigma]\!]_\lambda$ to distributions on bitstrings $[\![\tau]\!]_\lambda$ is computable up to an error $\eta(\lambda)$ by a probabilistic Turing Machine TM_λ with symbols $\mathsf{0}, \mathsf{1}$. Specifically, for every $v \in [\![\sigma]\!]_\lambda$, the distribution $\mathsf{TM}_\lambda(v)$ on $[\![\tau]\!]_\lambda \cup \{\bot\}$ approximates $[\![\mathsf{d}]\!]_\lambda(v)$ with error $\leqslant \eta(\lambda)$. Both the number of states and the runtime of TM_λ are $\leqslant p(\lambda)$ if $\lambda \geqslant N$.*

To seamlessly account for the possible renaming of channel names, a distinguisher for protocols of type $\Delta \vdash I \to O$ is allowed to operate in a larger context Δ' that subsumes the original context Δ via an embedding $\phi : \Delta' \to \Delta$. In this larger context, we specify the adversarial input channels I' that the distinguisher will query for a value, and the adversarial output channels O' that the distinguisher will assign values to. The adversarial inputs I' will be a subset of the protocol outputs O, appropriately translated along ϕ. Dually, the protocol inputs I, appropriately translated along ϕ, will be a subset of the adversarial outputs O'. In the interaction between the adversary and the protocol, every query for a value of a channel $o \in I'$ will extract the value of the channel $o \in \phi^\star(O)$ as computed by the the protocol, and pass it on to the distinguisher. Conversely, an input on channel $i \in \phi^\star(I)$ to the protocol occurs after the distinguisher computes the value of the channel $i \in O'$.

Since our distinguishers will be resource-bounded, we need to bind the number of interactions or *rounds* between the distinguisher and the protocol. In each round, the adversary examines its internal state to determine the type of interaction to perform next, and steps to a new state. This transition function is a partial probabilistic function of type $\mathsf{St} \rightharpoonup \big(\{\bot\} \cup I' \cup O'\big) \times \mathsf{St}$. That is, for any internal state s the adversary probabilistically decides among: *1)* no interaction, coupled with stepping to a new state s'; *2)* querying a channel $o \in I'$, coupled with stepping to a new state s'; *3)* an assignment to a channel $i \in O'$, coupled with stepping to a new state s'; or *4)* halting, in which case the game between the distinguisher and the protocol ends without a decision Boolean. We use this last option to capture probabilistic computations that only succeed up to a negligible error.

If a distinguisher chooses to query the channel $o \in I'$ and receives a value v as a response to the query, it will update its internal state according to an input assignment function of type $[\![\tau]\!] \times \mathsf{St} \to \mathsf{St}$, where τ is the type of the channel o in Δ'. That is, for any value $v \in [\![\tau]\!]$ and any state s, the distinguisher steps to a new state s' that records the value v as a result of the query. If a distinguisher chooses a value assignment to a channel $i \in O'$, the value v – if any – is determined by an output valuation function of type $\mathsf{St} \to [\![\tau]\!] \cup \{\bot\}$, where τ is the type of the channel i in Δ'. After completing the designated number of rounds, the distinguisher converts its internal state to a final decision Boolean according to a function of type $\mathsf{St} \to \{0,1\}$.

To bind the complexity of the aforementioned operations, we implement them as Turing Machines. For convenience, we allow TMs with multiple tapes. As is standard, in the initial configuration all tapes except the first are fully blank. The internal state of the distinguisher is typically encoded as a bitstring, containing *e.g.*, register values together with the sequence of instructions to be executed, if we view the distinguisher as an essentially arbitrary probabilistic program. For our purposes, it will be convenient to allow additional symbols besides 0, 1 on our TM tapes: when justifying the COMP-CONG-LEFT rule of the approximate fragment of IPDL, we will suitably compose the distinguisher with the common context Q, which is an IPDL protocol. The protocol Q thus becomes integrated into the new distinguisher's code. Instead of encoding IPDL protocols as bitstrings, we will suitably enrich our baseline set of symbols so that we can faithfully capture IPDL code.

Definition 12 (Distinguishers)**.** *Fix an IPDL signature Σ and an interpretation $[\![-]\!]$ for Σ. A* distinguisher Adv *for protocols of type $\Delta \vdash I \to O$ is a tuple $\big(\Delta', I', O', \phi, \#_{\mathsf{round}}, \#_{\mathsf{tape}}, \mathsf{Symb}, \mathsf{St}, s_\star, \mathsf{T}, \{\mathsf{I}_o\}_{o \in I'}, \{\mathsf{O}_i\}_{i \in O'}, \mathsf{D}\big)$, where*

- *Δ' is a channel context;*
- *$I' \subseteq \Delta'$ is a set of channels that the adversary can query for a value;*
- *$O' \subseteq \Delta'$ is a set of channels to which the adversary can assign a value;*
- *$\phi : \Delta' \to \Delta$ is an embedding of Δ into Δ';*
- *$\#_{\mathsf{round}} \geqslant 1$ is the number of rounds the adversary will perform;*
- *$\#_{\mathsf{tape}} \geqslant 1$ is the number of TM tapes at our disposal;*
- *Symb is a finite set of additional symbols that will be used to encode the distinguisher's internal state;*
- *$\mathsf{St} \subseteq \big(\{0,1\} \bigsqcup \mathsf{Symb}\big)^k$ is a set of strings of a fixed length $k \geqslant 1$ consisting of symbols drawn from the disjoint union of the sets $\{0,1\}$ and Symb;*
- *$s_\star \in \mathsf{St}$ is the initial state;*
- *T is a probabilistic TM that computes a partial function $\mathsf{St} \rightharpoonup \big(\{\bot\} \cup I' \cup O'\big) \times \mathsf{St}$, with $\#_{\mathsf{tape}}$-many tapes and using symbols from the set $\{0,1\} \bigsqcup \{\bot\} \bigsqcup (I' \cup O') \bigsqcup \mathsf{Symb}$,*
- *each I_o with $o : \tau \in \Delta$ is a deterministic TM that computes a function $\mathsf{St} \times [\![\tau]\!] \to \mathsf{St}$, with $\#_{\mathsf{tape}}$-many tapes and using symbols from the set $\{0,1\} \bigsqcup \mathsf{Symb}$,*
- *each O_i with $i : \tau \in \Delta$ is a deterministic TM that computes a function $\mathsf{St} \to [\![\tau]\!] \cup \{\bot\}$, with $\#_{\mathsf{tape}}$-many tapes and using symbols from the set $\{0,1\} \bigsqcup \{\bot\} \bigsqcup \mathsf{Symb}$,*
- *D is a deterministic TM that computes a function $\mathsf{St} \to \{0,1\}$, with $\#_{\mathsf{tape}}$-many tapes and using symbols from the set $\{0,1\} \bigsqcup \mathsf{Symb}$.*

We furthermore require that

- *$I' \subseteq \phi^\star(O)$, and*
- *$\phi^\star(I) \subseteq O'$,*
- *$\phi^\star(O) \cap O' = \varnothing$.*

The probabilistic Turing Machine T that computes the transition function can terminate in a non-accepting state with probability > 0. We will be interested in families of distinguishers where this *error* as a function of the security parameter is negligible.

Definition 13 (Distinguisher error)**.** *Fix a distinguisher Adv as in Definition* **??**. *We say that Adv* has error up to *$\varepsilon \in \mathbb{Q}_{\geqslant 0}$, written $\mathsf{err}(\mathsf{Adv}) \leqslant \varepsilon$, if for any state $s \in \mathsf{St}$ the transition function $\mathsf{T}(s)$ is undefined with probability $\leqslant \varepsilon$. In other words, when T is run with the initial tape contents s, it halts in a non-accepting state with probability $\leqslant \varepsilon$.*

To ensure that a distinguisher does not have access to computationally expensive functions such as the discrete logarithm, we need to impose a bound on its computational resources. We will be interested in families of distinguishers where the bound as the function of the security parameter is polynomial.

Definition 14 (Resource-bounded distinguishers)**.** *Fix a distinguisher Adv as in Definition* **??**. *We say that Adv* is bounded by *$K \in \mathbb{N}$, written $|\mathsf{Adv}| \leqslant K$, if:*

- *$\#_{\mathsf{round}}$, $\#_{\mathsf{tape}} \leqslant K$;*
- *$|I'| \leqslant K$ and for each $o \in I'$ with $o : \tau \in \Delta'$, we have $|\tau| \leqslant K$,*
- *$|O'| \leqslant K$ and for each $i \in O'$ with $i : \tau \in \Delta'$, we have $|\tau| \leqslant K$,*

```
Algorithm $\mathsf{Adv} \overset{[\![-]\!]}{\rightleftharpoons} P$:

$s := s_\star$
$P := \phi^\star(P)$
for $\#_{\mathsf{round}}$ rounds
    $P' \leftarrow P\Downarrow$
    $(q, s') \leftarrow \mathsf{T}(s)$
    if $q = \bot$ then
        $s := s'$
        $P := P'$
    else if $q = i \in O'$ then
        if $\mathsf{O}_i(s') = v$ for some $v$ then
            $s := s'$
            $P := P'[\mathsf{read}\ i := \mathsf{val}\ v]$
        else
            $s := s'$
            $P := P'$
    else if $q = o \in I'$ then
        if $(o := v) \in P'$ for some $v$ then
            $s := \mathsf{I}_o(v, s')$
            $P := P'$
        else
            $s := s'$
            $P := P'$
return $\mathsf{D}(s)$
```

Figure 22: Interaction of an IPDL protocol $\Delta \vdash P : I \to O$ with a distinguisher Adv.

- $|\mathsf{Symb}| \leqslant K$;
- *the length k of a state $s \in \mathsf{St}$ is $\leqslant K$;*
- *the number of states of each TM $\mathsf{T}, \mathsf{I}_o, \mathsf{O}_i, \mathsf{D}$ is $\leqslant K$;*
- *the runtime of each TM $\mathsf{T}, \mathsf{I}_o, \mathsf{O}_i, \mathsf{D}$ is $\leqslant K$.*

We note that instead of having a separate Turing Machine I_o for each channel $o \in I'$ we could have required a single Turing Machine that performs the computation across all channels in I', and analogously for O_i. However, this is unnecessary as the number of channels in I' and O' is $\mathsf{O}(\mathsf{poly}(\lambda))$, and the current formulation is more convenient for our purposes. We can now formally define the interaction between a distinguisher Adv and a protocol P.

Definition 15 (Interaction)**.** *Fix an IPDL signature Σ and an interpretation $[\![-]\!]$ for Σ. Let Adv be a distinguisher for protocols of type $\Delta \vdash I \to O$ and let $\Delta \vdash P : I \to O$. We define $\mathsf{Adv} \overset{[\![-]\!]}{\rightleftharpoons} P$ to be the probability sub-distribution on Booleans induced by the algorithm in Figure* **??***.*

In Figure **??**, the distinguisher interacts with the protocol through the specified number of rounds. The algorithm maintains a state variable s and a protocol variable P, which we respectively initialize to the initial state $s_\star$ and the original protocol P, appropriately embedded in Δ'. In each round, the protocol P probabilistically evolves to a new protocol P'. Independently, the distinguisher probabilistically computes the type of interaction $q \in \{\bot\} \cup I' \cup O'$

together with a new state s' according to $\mathsf{T}(s)$. If $q = \bot$, in which case no interaction takes place, the state and the protocol are updated to s' and P'. If $q = i$ for some $i \in O'$, we compute $\mathsf{O}_i(s')$ to see if in the distinguisher's current state s' the channel i carries a value v. If this is the case, we update the state to s' while computing a new protocol $P'[\mathsf{read}\ i := \mathsf{val}\ v]$. Otherwise we update the state and the protocol to s' and P'. Finally, if $q = o$ for some $o \in I'$, we examine the protocol P' to see if the output channel o carries a value v. If this is the case, we compute a new distinguisher state $\mathsf{I}_o(v, s')$, while updating the protocol to P'. Otherwise we update the state and the protocol to s' and P'. After completing the prescribed number of rounds, we obtain a decision Boolean $\mathsf{D}(s)$ based on the distinguisher's current state.

Note: Strictly speaking, the interaction $\mathsf{Adv} \overset{[\![-]\!]}{\rightleftharpoons} P$ is only a *sub*-distribution on Booleans, since $\mathsf{T}(s)$ may halt without a result. Since the probability of this happening will be negligible, this technical point is inessential. We are now ready to give the definition of computational indistinguishability.

Definition 16 (Computational Indistinguishability). *Let Σ be an IPDL signature and consider a family $\{[\![-]\!]_\lambda\}_{\lambda\in\mathbb{N}}$ of interpretations for Σ. Let $\{\Delta_\lambda \vdash P_\lambda : I_\lambda \to O_\lambda\}_{\lambda\in\mathbb{N}}$ and $\{\Delta_\lambda \vdash Q_\lambda : I_\lambda \to O_\lambda\}_{\lambda\in\mathbb{N}}$ be two protocol families with identical typing judgments. We say that $\{P_\lambda\}$ and $\{Q_\lambda\}$ are* indistinguishable *under $\{[\![-]\!]_\lambda\}$, written*

$$\{[\![-]\!]_\lambda\}_{\lambda\in\mathbb{N}} \models \{\Delta_\lambda \vdash P_\lambda \approx Q_\lambda : I_\lambda \to O_\lambda\}_{\lambda\in\mathbb{N}}$$

if for any polynomial $p(\lambda)$ and negligible function $\eta(\lambda)$, there exists a negligible function $\varepsilon(\lambda)$ with a natural number $N \in \mathbb{N}$ such that for any $\lambda \geqslant N$ and any distinguisher Adv for protocols of type $\Delta_\lambda \vdash I_\lambda \to O_\lambda$ with respect to the interpretation $[\![-]\!]_\lambda$, such that $|\mathsf{Adv}| \leqslant p(\lambda)$ and $\mathsf{err}(\mathsf{Adv}) \leqslant \eta(\lambda)$, we have

$$\left|\mathsf{Pr}\left[\mathsf{Adv} \overset{[\![-]\!]_\lambda}{\rightleftharpoons} P_\lambda = 1\right] - \mathsf{Pr}\left[\mathsf{Adv} \overset{[\![-]\!]_\lambda}{\rightleftharpoons} Q_\lambda = 1\right]\right| \leqslant \varepsilon(\lambda).$$

Instead of comparing the probabilities that the decision Boolean is 1, we could have likewise used 0: the probability $\mathsf{Pr}[b = 0]$ that the decision Boolean b is 0 is only negligibly different from $1 - \mathsf{Pr}[b = 1]$, since the probability that the game ends without a decision Boolean is negligible. This follows from the fact that the error is negligible and the number of rounds is $\mathsf{O}(\mathsf{poly}(\lambda))$.

5 Soundness of Approximate Equality in IPDL

We will say that the underlying exact IPDL theory $\mathbb{T}_=$ is sound with respect to an interpretation $[\![-]\!]$ if each of its constituent theories $\mathbb{T}_{\mathsf{exp}}, \mathbb{T}_{\mathsf{dist}}, \mathbb{T}_{\mathsf{prot}}$ is sound with respect to $[\![-]\!]$. We will use the judgement $[\![-]\!] \models \mathbb{T}_=$ to denote this. We now define what it means for an asymptotic theory to be sound with respect to a family of interpretations.

Definition 17. *Fix an IPDL signature Σ. An approximate axiom family $\{\Delta_\lambda \vdash P_\lambda \approx Q_\lambda : I_\lambda \to O_\lambda\}_{\lambda\in\mathbb{N}}$ is* sound *with respect to a family of interpretations $\{[\![-]\!]_\lambda\}_{\lambda\in\mathbb{N}}$ if $\{[\![-]\!]_\lambda\}_{\lambda\in\mathbb{N}} \models \{\Delta_\lambda \vdash P_\lambda \approx Q_\lambda : I_\lambda \to O_\lambda\}_{\lambda\in\mathbb{N}}$.*

The asymptotic IPDL theory $\mathbb{T}_\approx$ is said to be sound if each of its axioms is sound, and we will utilize the judgement $\{[\![-]\!]_\lambda\}_{\lambda\in\mathbb{N}} \models \mathbb{T}_\approx$ to denote this. Our goal in this section is to prove that this implies overall soundness:

Theorem 1 (Soundness of asymptotic equality of protocols). *Assume*

- *an IPDL signature Σ with type symbols $\mathsf{t}_1, \ldots, \mathsf{t}_{|\Sigma_\mathsf{t}|}$,*
- *two protocol families $\{\Delta_\lambda \vdash P_\lambda : I_\lambda \to O_\lambda\}_{\lambda\in\mathbb{N}}$ and $\{\Delta_\lambda \vdash Q_\lambda : I_\lambda \to O_\lambda\}_{\lambda\in\mathbb{N}}$ with identical typing judgments,*
- *a PPT family of interpretations $\{[\![-]\!]_\lambda\}_{\lambda\in\mathbb{N}}$,*
- *a strict IPDL theory $\mathbb{T}_=$ such that for each $\lambda \in \mathbb{N}$, we have $[\![-]\!]_\lambda \models \mathbb{T}_=$, and*
- *an asymptotic IPDL theory $\mathbb{T}_\approx$ such that $\{[\![-]\!]_\lambda\}_{\lambda\in\mathbb{N}} \models \mathbb{T}_\approx$.*

Then

$$\mathbb{T}_=; \mathbb{T}_\approx \Rightarrow \{\Delta_\lambda \vdash P_\lambda \approx Q_\lambda : I_\lambda \to O_\lambda\}_{\lambda\in\mathbb{N}}$$

implies

$$\{[\![-]\!]_\lambda\}_{\lambda\in\mathbb{N}} \models \{\Delta_\lambda \vdash P_\lambda \approx Q_\lambda : I_\lambda \to O_\lambda\}_{\lambda\in\mathbb{N}}.$$

We begin by restructuring our proofs of approximate equality so that all invocations of the rule EMBED are carried out first, followed by applications of the rule INPUT-UNUSED, which are in turn followed by invocations of the rule CONG-COMP-LEFT, and finally by applications of the rule CONG-NEW. Crucially, a sequence of applications of the CONG-COMP-LEFT rule with common contexts $Q_1, \ldots, Q_n$ can be collapsed into a single application with the common context $Q_1 \mid\mid \ldots \mid\mid Q_n$. An analogous observation holds for a sequence of applications of the EMBED rule with embeddings $\phi_1, \ldots, \phi_n$, which can be combined into a single application with the embedding $\phi_n \circ \ldots \circ \phi_1$. The new layered form of our approximate judgements is shown in Figures **??** and **??**.

Lemma 23. *For any protocols $\Delta \vdash P : I \to O$ and $\Delta \vdash Q : I \to O$ we have $\Delta \vdash P \approx Q : I \to O$ wid k len l if and only if $\Delta \vdash P \approx_5 Q : I \to O$ wid k len l.*

Sketch. For the right-to-left direction, $\Delta \vdash P \cong_4 Q : I \to O$ len l clearly implies $\Delta \vdash P \cong Q : I \to O$ len l. Thus we have that $\Delta \vdash P \cong_5 Q : I \to O$ len l implies $\Delta \vdash P \approx Q : I \to O$ wid 1 len l. Hence $\Delta \vdash P \approx_5 Q : I \to O$ wid k len l implies $\Delta \vdash P \approx Q : I \to O$ wid k len l, as desired.

For the left-to-right direction, we first show that $\Delta \vdash P \cong Q : I \to O$ len l implies $\Delta \vdash P \cong_5 Q : I \to O$ len l by induction on the former derivation. That $\Delta \vdash P \approx Q : I \to O$ wid k len l implies $\Delta \vdash P \approx_5 Q : I \to O$ wid k len l now follows immediately. □

We can now prove a soundness theorem for approximate equality. Roughly speaking, if P is approximately equal to Q, then the advantage that an adversary Adv has in distinguishing P and Q is a reasonable combination of the distinguishing advantages against each approximate axiom by an adversary whose computational resources are only slightly larger than those of the original adversary Adv. We start by proving that strict equality of protocols implies perfect indistinguishability against any adversary (not just a resource-bounded one).

Lemma 24 (Soundness of approximate equality of protocols: Perfect indistinguishability)**.** *For any IPDL signature Σ, interpretation $[\![-]\!]$ for Σ, strict IPDL theory $\mathbb{T}_=$ such that $[\![-]\!] \models \mathbb{T}_=$, derivation $\mathbb{T}_=$; $\Delta \vdash P = Q : I \to O$, and distinguisher Adv for protocols of type $\Delta \vdash I \to O$ under the interpretation $[\![-]\!]$, we have*

$$\left|\mathsf{Pr}\big[\mathsf{Adv} \overset{[\![-]\!]}{\Longleftrightarrow} P = 1\big] - \mathsf{Pr}\big[\mathsf{Adv} \overset{[\![-]\!]}{\Longleftrightarrow} Q = 1\big]\right| = 0.$$

Proof. Fix a distinguisher Adv as in Definition **??**. By assumption, we have a proof $\Delta \vdash P = Q : I \to O$, which means we also have a proof that $\Delta' \vdash \phi^\star(P) = \phi^\star(Q) : \phi^\star(I) \to \phi^\star(O)$. The soundness theorem for strict equality of protocols applied to this proof gives us a bisimulation $\sim$ such that $1[\phi^\star(P)] \sim 1[\phi^\star(Q)]$. Now let $\sim_{\mathsf{adv}}$ be a binary relation on sub-distributions on pairs where the first element is a distinguisher state and the second is a protocol of type $\Delta \vdash I \to O$, defined as follows:

- $(s, \eta) \sim_{\mathsf{adv}} (s, \varepsilon)$ if $s \in \mathsf{St}$ and $\eta \sim \varepsilon$, where we use a distribution in place of a protocol to indicate the obvious lifting to sub-distributions on pairs of the the aforementioned form, and
- $1[\bot] \sim_{\mathsf{adv}} 1[\bot]$, where $\bot$ indicates that the security game between the distinguisher and the protocol halted without a decision Boolean.

Let $\mathcal{L}_{\sim_{\mathsf{adv}}}$ be the closure of $\sim_{\mathsf{adv}}$ under joint convex combinations. Explicitly, $\mathcal{L}_{\sim_{\mathsf{adv}}}$ is defined by

$$\Big(\sum_i c_i\, \eta_i\Big)\ \mathcal{L}_{\sim_{\mathsf{adv}}}\ \Big(\sum_i c_i\, \varepsilon_i\Big)$$

for coefficients $c_i > 0$ with $\sum_i c_i = 1$ and distributions $\eta_i \sim_{\mathsf{adv}} \varepsilon_i$. We now establish a loop invariant for the algorithm in Figure **??**. Before starting the first round, the initial distributions are suitably related: by assumption, we have $1[\phi^\star(P)] \sim 1[\phi^\star(Q)]$, which means that

$$1\big[(s_\star, \phi^\star(P))\big]\ \mathcal{L}_{\sim_{\mathsf{adv}}}\ 1\big[(s_\star, \phi^\star(Q))\big]$$

as the two distributions are already related under $\sim_{\mathsf{adv}}$. Now assume that we have two sub-distributions related by $\mathcal{L}_{\sim_{\mathsf{adv}}}$. We prove that performing a single round yields sub-distributions that are again related by $\mathcal{L}_{\sim_{\mathsf{adv}}}$. It suffices to show this for the case $(s, \eta) \sim_{\mathsf{adv}} (s, \varepsilon)$, where $s \in \mathsf{St}$ and $\eta \sim \varepsilon$. We first compute the distributions $\eta' := \eta\Downarrow$ and $\varepsilon' := \varepsilon\Downarrow$. By definition of $\sim$ we have $\eta' \sim \varepsilon'$. Independently, we probabilistically compute the type of interaction

$$\boxed{\Delta \vdash P \cong_0 Q : I \to O \ \mathsf{len}\ l}$$

$$\frac{\Delta \vdash P \approx Q : I \to O \ \mathsf{axiom}}{\Delta \vdash P \cong_0 Q : I \to O \ \mathsf{len}\ 0}\ \textsc{axiom}$$

$$\boxed{\Delta \vdash P \cong_1 Q : I \to O \ \mathsf{len}\ l}$$

$$\frac{\Delta \vdash P \cong_0 Q : I \to O \ \mathsf{len}\ l}{\Delta \vdash P \cong_1 Q : I \to O \ \mathsf{len}\ l}\ \textsc{subsume} \qquad \frac{\phi : \Delta_1 \to \Delta_2 \qquad \Delta_2 \vdash P \cong_0 Q : I \to O \ \mathsf{len}\ l}{\Delta_1 \vdash \phi^\star(P) \cong_1 \phi^\star(Q) : \phi^\star(I) \to \phi^\star(O) \ \mathsf{len}\ l}\ \textsc{embed}$$

$$\boxed{\Delta \vdash P \cong_2 Q : I \to O \ \mathsf{len}\ l}$$

$$\frac{\Delta \vdash P \cong_1 Q : I \to O \ \mathsf{len}\ l}{\Delta \vdash P \cong_2 Q : I \to O \ \mathsf{len}\ l}\ \textsc{subsume} \qquad \frac{i \notin I \cup O \qquad \Delta \vdash P \cong_2 Q : I \to O \ \mathsf{len}\ l}{\Delta \vdash P \cong_2 Q : I \cup \{i\} \to O \ \mathsf{len}\ l}\ \textsc{input-unused}$$

$$\boxed{\Delta \vdash P \cong_3 Q : I \to O \ \mathsf{len}\ l}$$

$$\frac{\Delta \vdash P \cong_2 Q : I \to O \ \mathsf{len}\ l}{\Delta \vdash P \cong_3 Q : I \to O \ \mathsf{len}\ l}\ \textsc{subsume}$$

$$\frac{\Delta \vdash P \cong_2 P' : I \cup O_2 \to O_1 \ \mathsf{len}\ l \qquad \Delta \vdash Q : I \cup O_1 \to O_2}{\Delta \vdash P \parallel Q \cong_3 P' \parallel Q : I \to O_1 \cup O_2 \ \mathsf{len}\ l + \|Q\|_{\mathsf{TM}} + 3}\ \textsc{cong-comp-left}$$

$$\boxed{\Delta \vdash P \cong_4 Q : I \to O \ \mathsf{len}\ l}$$

$$\frac{\Delta \vdash P \cong_3 Q : I \to O \ \mathsf{len}\ l}{\Delta \vdash P \cong_4 Q : I \to O \ \mathsf{len}\ l}\ \textsc{subsume} \qquad \frac{\Delta, o : \tau \vdash P \cong_4 P' : I \to O \cup \{o\} \ \mathsf{len}\ l}{\Delta \vdash \big(\mathsf{new}\ o : \tau\ \mathsf{in}\ P\big) \cong_4 \big(\mathsf{new}\ o : \tau\ \mathsf{in}\ P'\big) : I \to O \ \mathsf{len}\ l}\ \textsc{cong-new}$$

$$\boxed{\Delta \vdash P \cong_5 Q : I \to O \ \mathsf{len}\ l}$$

$$\frac{\Delta \vdash P = P' : I \to O \qquad \Delta \vdash P' \cong_4 Q' : I \to O \ \mathsf{len}\ l \qquad \Delta \vdash Q' = Q : I \to O}{\Delta \vdash P \cong_5 Q : I \to O \ \mathsf{len}\ l}\ \textsc{strict-subsume}$$

Figure 23: Layered approximate congruence for IPDL protocols.

$$\boxed{\Delta \vdash P \approx_5 Q : I \to O \ \mathsf{wid}\ k \ \mathsf{len}\ l}$$

$$\frac{\Delta \vdash P = Q : I \to O}{\Delta \vdash P \approx_5 Q : I \to O \ \mathsf{wid}\ 0 \ \mathsf{len}\ 0}\ \textsc{strict} \qquad \frac{\Delta \vdash P \cong_5 Q : I \to O \ \mathsf{len}\ l}{\Delta \vdash P \approx_5 Q : I \to O \ \mathsf{wid}\ 1 \ \mathsf{len}\ l}\ \textsc{approx-cong}$$

$$\frac{\Delta \vdash P_1 \approx_5 P_2 : I \to O \ \mathsf{wid}\ k \ \mathsf{len}\ l}{\Delta \vdash P_2 \approx_5 P_1 : I \to O \ \mathsf{wid}\ k \ \mathsf{len}\ l}\ \textsc{sym}$$

$$\frac{\Delta \vdash P_1 \approx_5 P_2 : I \to O \ \mathsf{wid}\ k_1 \ \mathsf{len}\ l_1 \qquad \Delta \vdash P_2 \approx_5 P_3 : I \to O \ \mathsf{wid}\ k_2 \ \mathsf{len}\ l_2}{\Delta \vdash P_1 \approx_5 P_3 : I \to O \ \mathsf{wid}\ k_1 + k_2 \ \mathsf{len}\ \mathsf{max}(l_1, l_2)}\ \textsc{trans}$$

Figure 24: Layered approximate equality for IPDL protocols.

to perform together with a new distinguisher state s'. If no interaction has been chosen, the resulting distributions are (s', η') and (s', ε'). We have

$$(s', \eta') \ \mathcal{L}_{\sim_{\mathsf{adv}}} \ (s', \varepsilon')$$

as desired, as the two distributions are already related under $\sim_{\mathsf{adv}}$. If the interaction is an input on channel i, we compute $\mathsf{O}_i(s')$ to see if in the distinguisher's current state s' the channel i carries a value. If this is not the case, the resulting distributions are (s', η') and (s', ε'). Here we again have $(s', \eta') \ \mathcal{L}_{\sim_{\mathsf{adv}}} \ (s', \varepsilon')$, as desired. On the other hand, if the channel i carries a value v, the resulting distributions are $\big(s', \eta'[\mathsf{read}\ i := \mathsf{val}\ v]\big)$ and $\big(s', \varepsilon'[\mathsf{read}\ i := \mathsf{val}\ v]\big)$. Now because $\eta' \sim \varepsilon'$, by definition of $\sim$ we have $\eta'[\mathsf{read}\ i := \mathsf{val}\ v] \sim \varepsilon'[\mathsf{read}\ i := \mathsf{val}\ v]$. Thus we have

$$\big(s', \eta'[\mathsf{read}\ i := \mathsf{val}\ v]\big) \ \mathcal{L}_{\sim_{\mathsf{adv}}} \ \big(s', \varepsilon'[\mathsf{read}\ i := \mathsf{val}\ v]\big)$$

as desired, as the two distributions are already related under $\sim_{\mathsf{adv}}$. Finally, if the interaction is a query for an output channel o, we recall that the valuation property of the bisimulation $\sim$ allows us to jointly partition the distributions $\eta' \sim \varepsilon'$ into a joint convex combination

$$\eta' = \sum_i c_i\, \eta'_i \ \sim \ \sum_i c_i\, \varepsilon'_i = \varepsilon'$$

with $c_i > 0$ and $\sum_i c_i = 1$ such that

- the respective components $\eta'_i \sim \varepsilon'_i$ are again related, and
- $\eta'_i|_{\mathsf{val}(o)} = v_\perp = \varepsilon'_i|_{\mathsf{val}(o)}$ for the same $v_\perp \in \{\perp\} \cup [\![\tau]\!]$ where $o : \tau$ in Δ'.

Therefore, it suffices to consider the respective components $\eta'_i \sim \varepsilon'_i$ with the same $v_\perp$. If $v_\perp$ is $\perp$, then the resulting distributions are (s', η'_i) and (s', ε'_i). Here we again have $(s', \eta'_i) \ \mathcal{L}_{\sim_{\mathsf{adv}}} \ (s', \varepsilon'_i)$, as desired. On the other hand, if $v_\perp$ is a value v, then the resulting distributions are $\big(\mathsf{I}_o(v, s'), \eta'_i\big)$ and $\big(\mathsf{I}_o(v, s'), \varepsilon'_i\big)$. Thus we have

$$\big(\mathsf{I}_o(v, s'), \eta'_i\big) \ \mathcal{L}_{\sim_{\mathsf{adv}}} \ \big(\mathsf{I}_o(v, s'), \varepsilon'_i\big)$$

as desired, as the two distributions are already related under $\sim_{\mathsf{adv}}$. This proves that after completing the required number of rounds, we end up with two sub-distributions related by $\mathcal{L}_{\sim_{\mathsf{adv}}}$. It is now easy to see that they induce the same sub-distribution on decision Booleans. It suffices to prove this for the case $(s, \eta) \sim_{\mathsf{adv}} (s, \varepsilon)$, where $s \in \mathsf{St}$ and $\eta \sim \varepsilon$. But the state s is the same for both distributions, so the resulting distribution on decision Booleans is $1[\mathsf{D}(s)]$. This finishes the proof. □

We now establish soundness of approximate congruence of protocols as a sequence of lemmas, one for each level in Figure **??**. We note that at levels $0 - 2$, the length l of the derivation does not factor into the final distinguishing advantage because it is always 0 by construction. If we wish to make the ambient strict theory $\mathbb{T}_=$ and the ambient approximate theory with axioms $\Delta^1 \vdash P^1 \approx Q^1 : I^1 \to O^1, \ldots, \Delta^n \vdash P^n \approx Q^n : I^n \to O^n$ explicit, we write the approximate congruence judgement as

$$\mathbb{T}_=;\ \Delta^1 \vdash P^1 \approx Q^1 : I^1 \to O^1, \ldots, \Delta^n \vdash P^n \approx Q^n : I^n \to O^n \Rightarrow \Delta \vdash P \cong_i Q : I \to O \ \mathsf{len}\ l$$

for each level $i = 0, \ldots, 5$, and the approximate equality judgement as

$$\mathbb{T}_=;\ \Delta^1 \vdash P^1 \approx Q^1 : I^1 \to O^1, \ldots, \Delta^n \vdash P^n \approx Q^n : I^n \to O^n \Rightarrow \Delta \vdash P \approx_5 Q : I \to O \ \mathsf{wid}\ k\ \mathsf{len}\ l.$$

Lemma 25 (Soundness of approximate congruence of protocols: Level 0)**.** *For any IPDL signature Σ, interpretation $[\![-]\!]$ for Σ, strict IPDL theory $\mathbb{T}_=$ such that $[\![-]\!] \models \mathbb{T}_=$, and any*

- *approximate IPDL theory with axioms $\Delta^1 \vdash P^1 \approx Q^1 : I^1 \to O^1, \ldots, \Delta^n \vdash P^n \approx Q^n : I^n \to O^n$,*
- *derivation $\mathbb{T}_=;\ \Delta^1 \vdash P^1 \approx Q^1 : I^1 \to O^1, \ldots, \Delta^n \vdash P^n \approx Q^n : I^n \to O^n \Rightarrow \Delta \vdash P \cong_0 Q : I \to O\ \mathsf{len}\ l$,*
- *distinguisher Adv for protocols of type $\Delta \vdash I \to O$ under the interpretation $[\![-]\!]$ for which there exist $K_{\mathsf{adv}} \in \mathbb{N}$ and $\eta_{\mathsf{adv}} \in \mathbb{Q}_{\geqslant 0}$ such that $|\mathsf{Adv}| \leqslant K_{\mathsf{adv}}$ and $\mathsf{err}(\mathsf{Adv}) \leqslant \eta_{\mathsf{adv}}$,*

- *bounds* $\varepsilon^1, \dots, \varepsilon^n \in \mathbb{Q}_{\geqslant 0}$ *with the property that for any distinguisher* Adv^i *for protocols of type* $\Delta^i \vdash I^i \to O^i$ *with respect to the interpretation* $\llbracket - \rrbracket$ *such that* $|\mathsf{Adv}^i| \leqslant K_{\mathsf{adv}}$ *and* $\mathsf{err}(\mathsf{Adv}^i) \leqslant \eta_{\mathsf{adv}}$, *we have*

$$\left|\mathsf{Pr}\left[\mathsf{Adv}^i \xrightleftharpoons{\llbracket - \rrbracket} P^i = 1\right] - \mathsf{Pr}\left[\mathsf{Adv}^i \xrightleftharpoons{\llbracket - \rrbracket} Q^i = 1\right]\right| \leqslant \varepsilon^i,$$

we have

$$\left|\mathsf{Pr}\left[\mathsf{Adv} \xrightleftharpoons{\llbracket - \rrbracket} P = 1\right] - \mathsf{Pr}\left[\mathsf{Adv} \xrightleftharpoons{\llbracket - \rrbracket} Q = 1\right]\right| \leqslant \mathsf{max}(\varepsilon^1, \dots, \varepsilon^n).$$

Proof. We proceed by induction on the derivation of $\cong_0$. The AXIOM rule is clear. □

Lemma 26 (Soundness of approximate congruence of protocols: Level 1)**.** *For any IPDL signature* Σ, *interpretation* $\llbracket - \rrbracket$ *for* Σ, *strict IPDL theory* $\mathbb{T}_=$ *such that* $\llbracket - \rrbracket \vDash \mathbb{T}_=$, *and any*

- *approximate IPDL theory with axioms* $\Delta^1 \vdash P^1 \approx Q^1 : I^1 \to O^1, \dots, \Delta^n \vdash P^n \approx Q^n : I^n \to O^n$,
- *derivation* $\mathbb{T}_=; \Delta^1 \vdash P^1 \approx Q^1 : I^1 \to O^1, \dots, \Delta^n \vdash P^n \approx Q^n : I^n \to O^n \Rightarrow \Delta \vdash P \cong_1 Q : I \to O \; \mathsf{len}\; l$,
- *distinguisher* Adv *for protocols of type* $\Delta \vdash I \to O$ *under the interpretation* $\llbracket - \rrbracket$ *for which there exist* $K_{\mathsf{adv}} \in \mathbb{N}$ *and* $\eta_{\mathsf{adv}} \in \mathbb{Q}_{\geqslant 0}$ *such that* $|\mathsf{Adv}| \leqslant K_{\mathsf{adv}}$ *and* $\mathsf{err}(\mathsf{Adv}) \leqslant \eta_{\mathsf{adv}}$,
- *bounds* $\varepsilon^1, \dots, \varepsilon^n \in \mathbb{Q}_{\geqslant 0}$ *with the property that for any distinguisher* Adv^i *for protocols of type* $\Delta^i \vdash I^i \to O^i$ *with respect to the interpretation* $\llbracket - \rrbracket$ *such that* $|\mathsf{Adv}^i| \leqslant K_{\mathsf{adv}}$ *and* $\mathsf{err}(\mathsf{Adv}^i) \leqslant \eta_{\mathsf{adv}}$, *we have*

$$\left|\mathsf{Pr}\left[\mathsf{Adv}^i \xrightleftharpoons{\llbracket - \rrbracket} P^i = 1\right] - \mathsf{Pr}\left[\mathsf{Adv}^i \xrightleftharpoons{\llbracket - \rrbracket} Q^i = 1\right]\right| \leqslant \varepsilon^i,$$

we have

$$\left|\mathsf{Pr}\left[\mathsf{Adv} \xrightleftharpoons{\llbracket - \rrbracket} P = 1\right] - \mathsf{Pr}\left[\mathsf{Adv} \xrightleftharpoons{\llbracket - \rrbracket} Q = 1\right]\right| \leqslant \mathsf{max}(\varepsilon^1, \dots, \varepsilon^n).$$

Proof. We proceed by induction on the derivation of $\cong_1$. The SUBSUME rule follows immediately from the preceding lemma. In the case of the EMBED rule, let $\mathsf{Adv} \coloneqq \left(\Delta', I', O', \phi', \#_{\mathsf{round}}, \#_{\mathsf{tape}}, \mathsf{Symb}, \mathsf{St}, s_\star, \mathsf{T}, \left\{\mathsf{I}_o\right\}_{o \in I'}, \left\{\mathsf{O}_i\right\}_{i \in O'}, \mathsf{D}\right)$ be the original adversary for protocols of type $\Delta_1 \vdash \phi^\star(I) \to \phi^\star(O)$. Let

$$\mathsf{Adv}_{\mathcal{R}} \coloneqq \left(\Delta', I', O', \phi \circ \phi', \#_{\mathsf{round}}, \#_{\mathsf{tape}}, \mathsf{Symb}, \mathsf{St}, s_\star, \mathsf{T}, \left\{\mathsf{I}_o\right\}_{o \in I'}, \left\{\mathsf{O}_i\right\}_{i \in O'}, \mathsf{D}\right)$$

be the new adversary for protocols of type $\Delta_2 \vdash I \to O$. Clearly, $|\mathsf{Adv}_{\mathcal{R}}| \leqslant K_{\mathsf{adv}}$ and $\mathsf{err}(\mathsf{Adv}_{\mathcal{R}}) \leqslant \eta_{\mathsf{adv}}$, and

$$\mathsf{Pr}\left[\mathsf{Adv}_{\mathcal{R}} \xrightleftharpoons{\llbracket - \rrbracket} P = 1\right] = \mathsf{Pr}\left[\mathsf{Adv} \xrightleftharpoons{\llbracket - \rrbracket} \phi^\star(P) = 1\right]$$

$$\mathsf{Pr}\left[\mathsf{Adv}_{\mathcal{R}} \xrightleftharpoons{\llbracket - \rrbracket} Q = 1\right] = \mathsf{Pr}\left[\mathsf{Adv} \xrightleftharpoons{\llbracket - \rrbracket} \phi^\star(Q) = 1\right]$$

To finish the proof, we invoke the preceding lemma with the premise $\Delta_2 \vdash P \cong_0 Q : I \to O \; \mathsf{len}\; l$ and the new adversary $\mathsf{Adv}_{\mathcal{R}}$. □

Lemma 27 (Soundness of approximate congruence of protocols: Level 2)**.** *For any IPDL signature* Σ, *interpretation* $\llbracket - \rrbracket$ *for* Σ, *strict IPDL theory* $\mathbb{T}_=$ *such that* $\llbracket - \rrbracket \vDash \mathbb{T}_=$, *and any*

- *approximate IPDL theory with axioms* $\Delta^1 \vdash P^1 \approx Q^1 : I^1 \to O^1, \dots, \Delta^n \vdash P^n \approx Q^n : I^n \to O^n$,
- *derivation* $\mathbb{T}_=; \Delta^1 \vdash P^1 \approx Q^1 : I^1 \to O^1, \dots, \Delta^n \vdash P^n \approx Q^n : I^n \to O^n \Rightarrow \Delta \vdash P \cong_2 Q : I \to O \; \mathsf{len}\; l$,
- *distinguisher* Adv *for protocols of type* $\Delta \vdash I \to O$ *under the interpretation* $\llbracket - \rrbracket$ *for which there exist* $K_{\mathsf{adv}} \in \mathbb{N}$ *and* $\eta_{\mathsf{adv}} \in \mathbb{Q}_{\geqslant 0}$ *such that* $|\mathsf{Adv}| \leqslant K_{\mathsf{adv}}$ *and* $\mathsf{err}(\mathsf{Adv}) \leqslant \eta_{\mathsf{adv}}$,
- *bounds* $\varepsilon^1, \dots, \varepsilon^n \in \mathbb{Q}_{\geqslant 0}$ *with the property that for any distinguisher* Adv^i *for protocols of type* $\Delta^i \vdash I^i \to O^i$ *with respect to the interpretation* $\llbracket - \rrbracket$ *such that* $|\mathsf{Adv}^i| \leqslant K_{\mathsf{adv}}$ *and* $\mathsf{err}(\mathsf{Adv}^i) \leqslant \eta_{\mathsf{adv}}$, *we have*

$$\left|\mathsf{Pr}\left[\mathsf{Adv}^i \xrightleftharpoons{\llbracket - \rrbracket} P^i = 1\right] - \mathsf{Pr}\left[\mathsf{Adv}^i \xrightleftharpoons{\llbracket - \rrbracket} Q^i = 1\right]\right| \leqslant \varepsilon^i,$$

we have

$$\left|\mathsf{Pr}\left[\mathsf{Adv} \overset{[\![-]\!]}{\rightleftharpoons} P = 1\right] - \mathsf{Pr}\left[\mathsf{Adv} \overset{[\![-]\!]}{\rightleftharpoons} Q = 1\right]\right| \leqslant \mathsf{max}(\varepsilon^1, \dots, \varepsilon^n).$$

Proof. We proceed by induction on the derivation of $\cong_2$. The SUBSUME rule follows immediately from the preceding lemma. For the INPUT-UNUSED rule, let $\mathsf{Adv} := \left(\Delta', I', O', \phi, \#_{\mathsf{round}}, \#_{\mathsf{tape}}, \mathsf{Symb}, \mathsf{St}, s_\star, \mathsf{T}, \left\{\mathsf{I}_o\right\}_{o \in I'}, \left\{\mathsf{O}_i\right\}_{i \in O'}, \mathsf{D}\right)$ be the original adversary for protocols of type $\Delta \vdash I \cup \{i\} \to O$. Then $\mathsf{Adv}_{\mathcal{R}} := \mathsf{Adv}$ is also an adversary for protocols of type $\Delta \vdash I \to O$, and we have

$$\mathsf{Pr}\left[\mathsf{Adv}_{\mathcal{R}} \overset{[\![-]\!]}{\rightleftharpoons} P = 1\right] = \mathsf{Pr}\left[\mathsf{Adv} \overset{[\![-]\!]}{\rightleftharpoons} P = 1\right]$$

$$\mathsf{Pr}\left[\mathsf{Adv}_{\mathcal{R}} \overset{[\![-]\!]}{\rightleftharpoons} Q = 1\right] = \mathsf{Pr}\left[\mathsf{Adv} \overset{[\![-]\!]}{\rightleftharpoons} Q = 1\right]$$

To finish the proof, we appeal to the inductive hypothesis for the premise $\Delta \vdash P \cong_2 Q : I \to O \text{ len } l$ and the adversary $\mathsf{Adv}_{\mathcal{R}}$. □

To prove soundness of the approximate COMP-CONG-LEFT rule, we need the following crucial lemma, the proof of which we defer to the end of this section.

Lemma 28 (Absorption)**.** *There exists a polynomial $\mathcal{P}(x, y, z) \geqslant y$ such that for any*

- *IPDL signature Σ with type symbols $\mathsf{t}_1, \dots, \mathsf{t}_{|\Sigma_\mathsf{t}|}$,*
- *interpretation $[\![-]\!]$ for Σ for which there exist $K_{\mathsf{sem}} \in \mathbb{N}$ and $\eta_{\mathsf{sem}} \in \mathbb{Q}_{\geqslant 0}$ such that*
  - *for all type symbols t, $|\mathsf{t}| \leqslant K_{\mathsf{sem}}$,*
  - *for all function symbols f, $[\![\mathsf{f}]\!]$ is computable by a deterministic TM with symbols $0, 1$ such that the number of states and the runtime are $\leqslant K_{\mathsf{sem}}$, and*
  - *for all distribution symbols d, $[\![\mathsf{d}]\!]$ is computable up to error η_{sem} by a probabilistic TM with symbols $0, 1$ such that the number of states and the runtime are $\leqslant K_{\mathsf{sem}}$,*
- *distinguisher Adv for protocols of type $\Delta \vdash I \to O_1 \cup O_2$ under the interpretation $[\![-]\!]$ for which there exist $K_{\mathsf{adv}} \in \mathbb{N}$ and $\eta_{\mathsf{adv}} \in \mathbb{Q}_{\geqslant 0}$ such that $|\mathsf{Adv}| \leqslant K_{\mathsf{adv}}$ and $\mathsf{err}(\mathsf{Adv}) \leqslant \eta_{\mathsf{adv}}$,*
- *protocol $\Delta \vdash Q : I \cup O_1 \to O_2$,*

we have a new distinguisher $\mathsf{Adv}_{\mathcal{R}}$ for protocols of type $\Delta \vdash I \cup O_2 \to O_1$ with

$$|\mathsf{Adv}_{\mathcal{R}}| \leqslant \mathcal{P}\left(K_{\mathsf{sem}}, K_{\mathsf{adv}}, \|Q\|_{\mathsf{TM}}(|\mathsf{t}_1|, \dots, |\mathsf{t}_{|\Sigma_\mathsf{t}|}|)\right)$$

and $\mathsf{err}(\mathsf{Adv}_{\mathcal{R}}) \leqslant \mathsf{max}(\eta_{\mathsf{sem}}, \eta_{\mathsf{adv}})$ such that for any protocol $\Delta \vdash P : I \cup O_2 \to O_1$ we have

$$\left|\mathsf{Pr}\left[\mathsf{Adv} \overset{[\![-]\!]}{\rightleftharpoons} P \parallel Q = 1\right] - \mathsf{Pr}\left[\mathsf{Adv}_{\mathcal{R}} \overset{[\![-]\!]}{\rightleftharpoons} P = 1\right]\right| \leqslant \|Q\|_{\mathsf{TM}}(|\mathsf{t}_1|, \dots, |\mathsf{t}_{|\Sigma_\mathsf{t}|}|) * \eta_{\mathsf{sem}}.$$

Lemma 29 (Soundness of approximate equality of protocols: Level 3)**.** *For any*

- *IPDL signature Σ with type symbols $\mathsf{t}_1, \dots, \mathsf{t}_{|\Sigma_\mathsf{t}|}$,*
- *interpretation $[\![-]\!]$ for Σ for which there exist $K_{\mathsf{sem}} \in \mathbb{N}$ and $\eta_{\mathsf{sem}} \in \mathbb{Q}_{\geqslant 0}$ such that*
  - *for all type symbols t, $|\mathsf{t}| \leqslant K_{\mathsf{sem}}$,*
  - *for all function symbols f, $[\![\mathsf{f}]\!]$ is computable by a deterministic TM with symbols $0, 1$ such that the number of states and the runtime are $\leqslant K_{\mathsf{sem}}$, and*
  - *for all distribution symbols d, $[\![\mathsf{d}]\!]$ is computable up to error η_{sem} by a probabilistic TM with symbols $0, 1$ such that the number of states and the runtime are $\leqslant K_{\mathsf{sem}}$,*
- *strict IPDL theory $\mathbb{T}_=$ such that $[\![-]\!] \models \mathbb{T}_=$,*
- *approximate IPDL theory with axioms $\Delta^1 \vdash P^1 \approx Q^1 : I^1 \to O^1, \dots, \Delta^n \vdash P^n \approx Q^n : I^n \to O^n$,*

- *derivation*

$$\mathbb{T}_{=}; \Delta^1 \vdash P^1 \approx Q^1 : I^1 \to O^1, \ldots, \Delta^n \vdash P^n \approx Q^n : I^n \to O^n \Rightarrow \Delta \vdash P \cong_3 Q : I \to O \ \mathsf{len}\ l,$$

- *distinguisher* Adv *for protocols of type* $\Delta \vdash I \to O$ *under the interpretation* $\llbracket - \rrbracket$ *for which there exist* $K_{\mathsf{adv}} \in \mathbb{N}$ *and* $\eta_{\mathsf{adv}} \in \mathbb{Q}_{\geqslant 0}$ *such that* $|\mathsf{Adv}| \leqslant K_{\mathsf{adv}}$ *and* $\mathsf{err}(\mathsf{Adv}) \leqslant \eta_{\mathsf{adv}}$*,*
- *bound* $K_{\mathsf{len}} \in \mathbb{N}$ *such that* $l(|\mathsf{t}_1|, \ldots, |\mathsf{t}_{|\Sigma_{\mathsf{t}}|}|) \leqslant K_{\mathsf{len}}$*, and*
- *bounds* $\varepsilon^1, \ldots, \varepsilon^n \in \mathbb{Q}_{\geqslant 0}$ *with the property that for any distinguisher* Adv^i *for protocols of type* $\Delta^i \vdash I^i \to O^i$ *with respect to the interpretation* $\llbracket - \rrbracket$ *such that* $|\mathsf{Adv}^i| \leqslant \mathcal{P}(K_{\mathsf{sem}}, K_{\mathsf{adv}}, K_{\mathsf{len}})$ *and* $\mathsf{err}(\mathsf{Adv}^i) \leqslant \mathsf{max}(\eta_{\mathsf{sem}}, \eta_{\mathsf{adv}})$*, we have*
$$\left|\mathsf{Pr}\left[\mathsf{Adv}^i \xrightleftharpoons{\llbracket - \rrbracket} P^i = 1\right] - \mathsf{Pr}\left[\mathsf{Adv}^i \xrightleftharpoons{\llbracket - \rrbracket} Q^i = 1\right]\right| \leqslant \varepsilon^i,$$

we have
$$\left|\mathsf{Pr}\left[\mathsf{Adv} \xrightleftharpoons{\llbracket - \rrbracket} P = 1\right] - \mathsf{Pr}\left[\mathsf{Adv} \xrightleftharpoons{\llbracket - \rrbracket} Q = 1\right]\right| \leqslant \mathsf{max}(\varepsilon^1, \ldots, \varepsilon^n) + 2 * K_{\mathsf{len}} * \eta_{\mathsf{sem}}.$$

Proof. We proceed by induction on the derivation of $\cong_3$. The SUBSUME rule follows from Lemma **??** instantiated with $K_{\mathsf{adv}} := \mathcal{P}(K_{\mathsf{sem}}, K_{\mathsf{adv}}, K_{\mathsf{len}})$ and $\eta_{\mathsf{adv}} := \mathsf{max}(\eta_{\mathsf{sem}}, \eta_{\mathsf{adv}})$. We can do this because

$$\begin{aligned} |\mathsf{Adv}| &\leqslant K_{\mathsf{adv}} \leqslant \mathcal{P}(K_{\mathsf{sem}}, K_{\mathsf{adv}}, K_{\mathsf{len}}), \\ \mathsf{err}(\mathsf{Adv}) &\leqslant \eta_{\mathsf{adv}} \leqslant \mathsf{max}(\eta_{\mathsf{sem}}, \eta_{\mathsf{adv}}). \end{aligned}$$

Then we have

$$\left|\mathsf{Pr}\left[\mathsf{Adv} \xrightleftharpoons{\llbracket - \rrbracket} P = 1\right] - \mathsf{Pr}\left[\mathsf{Adv} \xrightleftharpoons{\llbracket - \rrbracket} Q = 1\right]\right| \leqslant \mathsf{max}(\varepsilon^1, \ldots, \varepsilon^n) \leqslant \mathsf{max}(\varepsilon^1, \ldots, \varepsilon^n) + 2 * K_{\mathsf{len}} * \eta_{\mathsf{sem}},$$

where the first inequality is the bound we got from Lemma **??**.

For the CONG-COMP-LEFT rule, we appeal to the absorption lemma to give us a new adversary $\mathsf{Adv}_{\mathcal{R}}$ for protocols of type $\Delta \vdash I \cup O_2 \to O_1$ with $|\mathsf{Adv}_{\mathcal{R}}| \leqslant \mathcal{P}\big(K_{\mathsf{sem}}, K_{\mathsf{adv}}, \|Q\|_{\mathsf{TM}}(|\mathsf{t}_1|, \ldots, |\mathsf{t}_{|\Sigma_{\mathsf{t}}|}|)\big)$ and $\mathsf{err}(\mathsf{Adv}_{\mathcal{R}}) \leqslant \mathsf{max}(\eta_{\mathsf{sem}}, \eta_{\mathsf{adv}})$ such that

$$\begin{aligned} \left|\mathsf{Pr}\left[\mathsf{Adv} \xrightleftharpoons{\llbracket - \rrbracket} P \parallel Q = 1\right] - \mathsf{Pr}\left[\mathsf{Adv}_{\mathcal{R}} \xrightleftharpoons{\llbracket - \rrbracket} P = 1\right]\right| &\leqslant \|Q\|_{\mathsf{TM}}(|\mathsf{t}_1|, \ldots, |\mathsf{t}_{|\Sigma_{\mathsf{t}}|}|) * \eta_{\mathsf{sem}}, \\ \left|\mathsf{Pr}\left[\mathsf{Adv} \xrightleftharpoons{\llbracket - \rrbracket} P' \parallel Q = 1\right] - \mathsf{Pr}\left[\mathsf{Adv}_{\mathcal{R}} \xrightleftharpoons{\llbracket - \rrbracket} P' = 1\right]\right| &\leqslant \|Q\|_{\mathsf{TM}}(|\mathsf{t}_1|, \ldots, |\mathsf{t}_{|\Sigma_{\mathsf{t}}|}|) * \eta_{\mathsf{sem}}. \end{aligned}$$

We can now appeal to Lemma **??** for the premise $\Delta \vdash P \approx_2 P' : I \cup O_2 \to O_1 \ \mathsf{wid}\ k \ \mathsf{len}\ l$ and the adversary $\mathsf{Adv}_{\mathcal{R}}$, instantiated with $K_{\mathsf{adv}} := \mathcal{P}(K_{\mathsf{sem}}, K_{\mathsf{adv}}, K_{\mathsf{len}})$ and $\eta_{\mathsf{adv}} := \mathsf{max}(\eta_{\mathsf{sem}}, \eta_{\mathsf{adv}})$. We can do this because

$$|\mathsf{Adv}_{\mathcal{R}}| \leqslant \mathcal{P}\big(K_{\mathsf{sem}}, K_{\mathsf{adv}}, \|Q\|_{\mathsf{TM}}(|\mathsf{t}_1|, \ldots, |\mathsf{t}_{|\Sigma_{\mathsf{t}}|}|)\big) \leqslant \mathcal{P}(K_{\mathsf{sem}}, K_{\mathsf{adv}}, K_{\mathsf{len}}),$$

where the second inequality follows from the assumption

$$\|Q\|_{\mathsf{TM}}(|\mathsf{t}_1|, \ldots, |\mathsf{t}_{|\Sigma_{\mathsf{t}}|}|) + 3 \leqslant K_{\mathsf{len}}.$$

We recall that l does not appear in the above because it comes from level 2, and is thus necessarily 0. Lemma **??** gives us the bound
$$\left|\mathsf{Pr}\left[\mathsf{Adv}_{\mathcal{R}} \xrightleftharpoons{\llbracket - \rrbracket} P = 1\right] - \mathsf{Pr}\left[\mathsf{Adv}_{\mathcal{R}} \xrightleftharpoons{\llbracket - \rrbracket} P' = 1\right]\right| \leqslant \mathsf{max}(\varepsilon^1, \ldots, \varepsilon^n).$$

Combining the three bounds

$$\begin{aligned} \left|\mathsf{Pr}\left[\mathsf{Adv} \xrightleftharpoons{\llbracket - \rrbracket} P \parallel Q = 1\right] - \mathsf{Pr}\left[\mathsf{Adv}_{\mathcal{R}} \xrightleftharpoons{\llbracket - \rrbracket} P = 1\right]\right| &\leqslant \|Q\|_{\mathsf{TM}}(|\mathsf{t}_1|, \ldots, |\mathsf{t}_{|\Sigma_{\mathsf{t}}|}|) * \eta_{\mathsf{sem}}, \\ \left|\mathsf{Pr}\left[\mathsf{Adv}_{\mathcal{R}} \xrightleftharpoons{\llbracket - \rrbracket} P = 1\right] - \mathsf{Pr}\left[\mathsf{Adv}_{\mathcal{R}} \xrightleftharpoons{\llbracket - \rrbracket} P' = 1\right]\right| &\leqslant \mathsf{max}(\varepsilon^1, \ldots, \varepsilon^n), \\ \left|\mathsf{Pr}\left[\mathsf{Adv}_{\mathcal{R}} \xrightleftharpoons{\llbracket - \rrbracket} P' = 1\right] - \mathsf{Pr}\left[\mathsf{Adv} \xrightleftharpoons{\llbracket - \rrbracket} P' \parallel Q = 1\right]\right| &\leqslant \|Q\|_{\mathsf{TM}}(|\mathsf{t}_1|, \ldots, |\mathsf{t}_{|\Sigma_{\mathsf{t}}|}|) * \eta_{\mathsf{sem}} \end{aligned}$$

yields the following:

$$\left|\mathsf{Pr}\left[\mathsf{Adv} \xrightleftharpoons{[\![-]\!]} P \parallel Q = 1\right] - \mathsf{Pr}\left[\mathsf{Adv} \xrightleftharpoons{[\![-]\!]} P' \parallel Q = 1\right]\right| \leqslant \mathsf{max}(\varepsilon^1, \dots, \varepsilon^n) + 2 * \|Q\|_{\mathsf{TM}}(|\mathsf{t}_1|, \dots, |\mathsf{t}_{|\Sigma_\mathsf{t}|}|) * \eta_{\mathsf{sem}}$$

$$\leqslant \mathsf{max}(\varepsilon^1, \dots, \varepsilon^n) + 2 * K_{\mathsf{len}} * \eta_{\mathsf{sem}}.$$

□

Lemma 30 (Soundness of approximate congruence of protocols: Level 4)**.** *For any*

- *IPDL signature Σ with type symbols $\mathsf{t}_1, \dots, \mathsf{t}_{|\Sigma_\mathsf{t}|}$,*
- *interpretation $[\![-]\!]$ for Σ for which there exist $K_{\mathsf{sem}} \in \mathbb{N}$ and $\eta_{\mathsf{sem}} \in \mathbb{Q}_{\geqslant 0}$ such that*
  - *for all type symbols t, $|\mathsf{t}| \leqslant K_{\mathsf{sem}}$,*
  - *for all function symbols f, $[\![\mathsf{f}]\!]$ is computable by a deterministic Turing Machine with symbols $\mathsf{0}, \mathsf{1}$ such that the number of states and the runtime are $\leqslant K_{\mathsf{sem}}$, and*
  - *for all distribution symbols d, $[\![\mathsf{d}]\!]_\lambda$ is computable up to an error η_{sem} by a probabilistic Turing Machine with symbols $\mathsf{0}, \mathsf{1}$ such that the number of states and the runtime are $\leqslant K_{\mathsf{sem}}$,*
- *strict IPDL theory $\mathbb{T}_=$ such that $[\![-]\!] \vDash \mathbb{T}_=$,*
- *approximate IPDL theory with axioms $\Delta^1 \vdash P^1 \approx Q^1 : I^1 \to O^1, \dots, \Delta^n \vdash P^n \approx Q^n : I^n \to O^n$,*
- *derivation*

$$\mathbb{T}_=;\, \Delta^1 \vdash P^1 \approx Q^1 : I^1 \to O^1, \dots, \Delta^n \vdash P^n \approx Q^n : I^n \to O^n \Rrightarrow \Delta \vdash P \cong_4 Q : I \to O \;\mathsf{len}\; l,$$

- *distinguisher Adv for protocols of type $\Delta \vdash I \to O$ under the interpretation $[\![-]\!]$ for which there exist $K_{\mathsf{adv}} \in \mathbb{N}$ and $\eta_{\mathsf{adv}} \in \mathbb{Q}_{\geqslant 0}$ such that $|\mathsf{Adv}| \leqslant K_{\mathsf{adv}}$ and $\mathsf{err}(\mathsf{Adv}) \leqslant \eta_{\mathsf{adv}}$,*
- *bound $K_{\mathsf{len}} \in \mathbb{N}$ such that $l(|\mathsf{t}_1|, \dots, |\mathsf{t}_{|\Sigma_\mathsf{t}|}|) \leqslant K_{\mathsf{len}}$, and*
- *bounds $\varepsilon^1, \dots, \varepsilon^n \in \mathbb{Q}_{\geqslant 0}$ with the property that for any distinguisher Adv^i for protocols of type $\Delta^i \vdash I^i \to O^i$ with respect to the interpretation $[\![-]\!]$ such that $|\mathsf{Adv}^i| \leqslant \mathcal{P}(K_{\mathsf{sem}}, K_{\mathsf{adv}}, K_{\mathsf{len}})$ and $\mathsf{err}(\mathsf{Adv}^i) \leqslant \mathsf{max}(\eta_{\mathsf{sem}}, \eta_{\mathsf{adv}})$, we have*

$$\left|\mathsf{Pr}\left[\mathsf{Adv}^i \xrightleftharpoons{[\![-]\!]} P^i = 1\right] - \mathsf{Pr}\left[\mathsf{Adv}^i \xrightleftharpoons{[\![-]\!]} Q^i = 1\right]\right| \leqslant \varepsilon^i,$$

we have

$$\left|\mathsf{Pr}\left[\mathsf{Adv} \xrightleftharpoons{[\![-]\!]} P = 1\right] - \mathsf{Pr}\left[\mathsf{Adv} \xrightleftharpoons{[\![-]\!]} Q = 1\right]\right| \leqslant \mathsf{max}(\varepsilon^1, \dots, \varepsilon^n) + 2 * K_{\mathsf{len}} * \eta_{\mathsf{sem}}.$$

Proof. We proceed by induction on the derivation of $\cong_4$. The SUBSUME rule follows immediately from the preceding lemma. To prove the CONG-NEW rule, let $\mathsf{Adv} := \left(\Delta', I', O', \phi, \#_{\mathsf{round}}, \#_{\mathsf{tape}}, \mathsf{Symb}, \mathsf{St}, s_\star, \mathsf{T}, \{\mathsf{I}_o\}_{o \in I'}, \{\mathsf{O}_i\}_{i \in O'}, \mathsf{D}\right)$ be the original adversary for protocols of type $\Delta \vdash I \to O$. We define a new adversary $\mathsf{Adv}_{\mathcal{R}}$ for protocols of type $\Delta, o : \tau \vdash I \to O \cup \{o\}$ as follows:

$$\mathsf{Adv}_{\mathcal{R}} := \left(\Delta'_{\mathcal{R}}, I'_{\mathcal{R}}, O'_{\mathcal{R}}, \phi_{\mathcal{R}}, \#_{\mathsf{round}}, \#_{\mathsf{tape}}, \mathsf{Symb}, \mathsf{St}, s_\star, \mathsf{T}^{\mathcal{R}}, \{\mathsf{I}^{\mathcal{R}}_o\}_{o \in I'_{\mathcal{R}}}, \{\mathsf{O}^{\mathcal{R}}_i\}_{i \in O'_{\mathcal{R}}}, \mathsf{D}\right).$$

In the above definition, we have the following, where we recall that each channel name stands for a de Bruijn index:

- $\Delta'_{\mathcal{R}} := \Delta', o : \tau$ is the channel context Δ' extended with the type τ,
- $I'_{\mathcal{R}}$ is the set $\{o + 1 \,|\, o \in I'\}$ of de Bruijn indices in I' increased by 1,
- $O'_{\mathcal{R}}$ is the set $\{i + 1 \,|\, i \in O'\}$ of de Bruijn indices in O' increased by 1,
- $\phi_{\mathcal{R}} := \phi \times \mathsf{id}_\tau$ from the context $\Delta', o : \tau$ to the context $\Delta, o : \tau$ is the extended channel embedding,
- each $\mathsf{I}^{\mathcal{R}}_{o+1}$ is the Turing Machine I_o corresponding to the original index $o \in I'$,

- each $\mathsf{O}^{\mathcal{R}}_{i+1}$ is the Turing Machine O_i corresponding to the original index $i \in O'$,
- $\mathsf{T}^{\mathcal{R}}$ is the Turing Machine T with each index $o \in I'$ and $i \in O'$ renamed to $o+1$ and $i+1$, respectively.

Since all we did was rename channel indices, it is easy to see that we have

$$\mathsf{Pr}\big[\mathsf{Adv}_{\mathcal{R}} \overset{[\![-]\!]}{\rightleftharpoons} P = 1\big] = \mathsf{Pr}\big[\mathsf{Adv} \overset{[\![-]\!]}{\rightleftharpoons} \big(\mathsf{new}\ o : \tau\ \mathsf{in}\ P\big) = 1\big]$$

$$\mathsf{Pr}\big[\mathsf{Adv}_{\mathcal{R}} \overset{[\![-]\!]}{\rightleftharpoons} Q = 1\big] = \mathsf{Pr}\big[\mathsf{Adv} \overset{[\![-]\!]}{\rightleftharpoons} \mathsf{new}\ o : \tau\ \mathsf{in}\ Q = 1\big]$$

In particular, the new adversary $\mathsf{Adv}_{\mathcal{R}}$ will never query for the newly exposed channel $o : \tau$, since this channel is hidden from the original adversary's view. To finish the proof, we appeal to the inductive hypothesis for the premise $\Delta, o : \tau \vdash P \cong_4 P' : I \to O \cup \{o\}\ \mathsf{len}\ l$ and the adversary $\mathsf{Adv}_{\mathcal{R}}$. □

Lemma 31 (Soundness of approximate congruence of protocols: Level 5). *For any*

- *IPDL signature* Σ *with type symbols* $\mathsf{t}_1, \dots, \mathsf{t}_{|\Sigma_\mathsf{t}|}$,
- *interpretation* $[\![-]\!]$ *for* Σ *for which there exist* $K_\mathsf{sem} \in \mathbb{N}$ *and* $\eta_\mathsf{sem} \in \mathbb{Q}_{\geqslant 0}$ *such that*
  - *for all type symbols* t, $|\mathsf{t}| \leqslant K_\mathsf{sem}$,
  - *for all function symbols* f, $[\![\mathsf{f}]\!]$ *is computable by a deterministic Turing Machine with symbols* $\mathsf{0}, \mathsf{1}$ *such that the number of states and the runtime are* $\leqslant K_\mathsf{sem}$, *and*
  - *for all distribution symbols* d, $[\![\mathsf{d}]\!]_\lambda$ *is computable up to an error* η_sem *by a probabilistic Turing Machine with symbols* $\mathsf{0}, \mathsf{1}$ *such that the number of states and the runtime are* $\leqslant K_\mathsf{sem}$,
- *strict IPDL theory* $\mathbb{T}_=$ *such that* $[\![-]\!] \models \mathbb{T}_=$,
- *approximate IPDL theory with axioms* $\Delta^1 \vdash P^1 \approx Q^1 : I^1 \to O^1, \dots, \Delta^n \vdash P^n \approx Q^n : I^n \to O^n$,
- *derivation*

$$\mathbb{T}_=; \Delta^1 \vdash P^1 \approx Q^1 : I^1 \to O^1, \dots, \Delta^n \vdash P^n \approx Q^n : I^n \to O^n \Rightarrow \Delta \vdash P \cong_5 Q : I \to O\ \mathsf{len}\ l,$$

- *distinguisher* Adv *for protocols of type* $\Delta \vdash I \to O$ *under the interpretation* $[\![-]\!]$ *for which there exist* $K_\mathsf{adv} \in \mathbb{N}$ *and* $\eta_\mathsf{adv} \in \mathbb{Q}_{\geqslant 0}$ *such that* $|\mathsf{Adv}| \leqslant K_\mathsf{adv}$ *and* $\mathsf{err}(\mathsf{Adv}) \leqslant \eta_\mathsf{adv}$,
- *bound* $K_\mathsf{len} \in \mathbb{N}$ *such that* $l(|\mathsf{t}_1|, \dots, |\mathsf{t}_{|\Sigma_\mathsf{t}|}|) \leqslant K_\mathsf{len}$, *and*
- *bounds* $\varepsilon^1, \dots, \varepsilon^n \in \mathbb{Q}_{\geqslant 0}$ *with the property that for any distinguisher* Adv^i *for protocols of type* $\Delta^i \vdash I^i \to O^i$ *with respect to the interpretation* $[\![-]\!]$ *such that* $|\mathsf{Adv}^i| \leqslant \mathcal{P}(K_\mathsf{sem}, K_\mathsf{adv}, K_\mathsf{len})$ *and* $\mathsf{err}(\mathsf{Adv}^i) \leqslant \mathsf{max}(\eta_\mathsf{sem}, \eta_\mathsf{adv})$, *we have*

$$\Big|\mathsf{Pr}\big[\mathsf{Adv}^i \overset{[\![-]\!]}{\rightleftharpoons} P^i = 1\big] - \mathsf{Pr}\big[\mathsf{Adv}^i \overset{[\![-]\!]}{\rightleftharpoons} Q^i = 1\big]\Big| \leqslant \varepsilon^i,$$

we have

$$\Big|\mathsf{Pr}\big[\mathsf{Adv} \overset{[\![-]\!]}{\rightleftharpoons} P = 1\big] - \mathsf{Pr}\big[\mathsf{Adv} \overset{[\![-]\!]}{\rightleftharpoons} Q = 1\big]\Big| \leqslant \mathsf{max}(\varepsilon^1, \dots, \varepsilon^n) + 2 * K_\mathsf{len} * \eta_\mathsf{sem}.$$

Proof. We proceed by induction on the derivation of $\cong_5$. To prove the STRICT-SUBSUME rule, we invoke the preceding lemma with the premise $\Delta \vdash P' \cong_4 Q' : I \to O\ \mathsf{len}\ l$ to give us the bound

$$\Big|\mathsf{Pr}\big[\mathsf{Adv} \overset{[\![-]\!]}{\rightleftharpoons} P' = 1\big] - \mathsf{Pr}\big[\mathsf{Adv} \overset{[\![-]\!]}{\rightleftharpoons} Q' = 1\big]\Big| \leqslant \mathsf{max}(\varepsilon^1, \dots, \varepsilon^n) + 2 * K_\mathsf{len} * \eta_\mathsf{sem}.$$

Invoking Lemma **??** on the two premises $\Delta \vdash P = P' : I \to O$ and $\Delta \vdash Q' = Q : I \to O$ gives us the two respective bounds

$$\Big|\mathsf{Pr}\big[\mathsf{Adv} \overset{[\![-]\!]}{\rightleftharpoons} P = 1\big] - \mathsf{Pr}\big[\mathsf{Adv} \overset{[\![-]\!]}{\rightleftharpoons} P' = 1\big]\Big| = 0,$$

$$\Big|\mathsf{Pr}\big[\mathsf{Adv} \overset{[\![-]\!]}{\rightleftharpoons} Q' = 1\big] - \mathsf{Pr}\big[\mathsf{Adv} \overset{[\![-]\!]}{\rightleftharpoons} Q = 1\big]\Big| = 0.$$

Combining the three bounds

$$\left|\mathsf{Pr}\big[\mathsf{Adv} \overset{[\![-]\!]}{\rightleftarrows} P = 1\big] - \mathsf{Pr}\big[\mathsf{Adv} \overset{[\![-]\!]}{\rightleftarrows} P' = 1\big]\right| = 0,$$

$$\left|\mathsf{Pr}\big[\mathsf{Adv} \overset{[\![-]\!]}{\rightleftarrows} P' = 1\big] - \mathsf{Pr}\big[\mathsf{Adv} \overset{[\![-]\!]}{\rightleftarrows} Q' = 1\big]\right| \leqslant \mathsf{max}(\varepsilon^1, \ldots, \varepsilon^n) + 2 * K_{\mathsf{len}} * \eta_{\mathsf{sem}},$$

$$\left|\mathsf{Pr}\big[\mathsf{Adv} \overset{[\![-]\!]}{\rightleftarrows} Q' = 1\big] - \mathsf{Pr}\big[\mathsf{Adv} \overset{[\![-]\!]}{\rightleftarrows} Q = 1\big]\right| = 0$$

yields the following:

$$\left|\mathsf{Pr}\big[\mathsf{Adv} \overset{[\![-]\!]}{\rightleftarrows} P = 1\big] - \mathsf{Pr}\big[\mathsf{Adv} \overset{[\![-]\!]}{\rightleftarrows} Q = 1\big]\right| \leqslant \mathsf{max}(\varepsilon^1, \ldots, \varepsilon^n) + 2 * K_{\mathsf{len}} * \eta_{\mathsf{sem}}.$$

□

Lemma 32 (Soundness of approximate equality of protocols)**.** *For any*

- *IPDL signature* Σ *with type symbols* $\mathsf{t}_1, \ldots, \mathsf{t}_{|\Sigma_\mathsf{t}|}$*,*
- *interpretation* $[\![-]\!]$ *for* Σ *for which there exist* $K_{\mathsf{sem}} \in \mathbb{N}$ *and* $\eta_{\mathsf{sem}} \in \mathbb{Q}_{\geqslant 0}$ *such that*
  - *for all type symbols* t*,* $|\mathsf{t}| \leqslant K_{\mathsf{sem}}$*,*
  - *for all function symbols* f*,* $[\![\mathsf{f}]\!]$ *is computable by a deterministic Turing Machine with symbols* $\mathsf{0}, \mathsf{1}$ *such that the number of states and the runtime are* $\leqslant K_{\mathsf{sem}}$*, and*
  - *for all distribution symbols* d*,* $[\![\mathsf{d}]\!]_\lambda$ *is computable up to an error* η_{sem} *by a probabilistic Turing Machine with symbols* $\mathsf{0}, \mathsf{1}$ *such that the number of states and the runtime are* $\leqslant K_{\mathsf{sem}}$*,*
- *strict IPDL theory* $\mathbb{T}_=$ *such that* $[\![-]\!] \vDash \mathbb{T}_=$*,*
- *approximate IPDL theory with axioms* $\Delta^1 \vdash P^1 \approx Q^1 : I^1 \to O^1, \ldots, \Delta^n \vdash P^n \approx Q^n : I^n \to O^n$*,*
- *derivation*

$$\mathbb{T}_=;\ \Delta^1 \vdash P^1 \approx Q^1 : I^1 \to O^1, \ldots, \Delta^n \vdash P^n \approx Q^n : I^n \to O^n \Rightarrow \Delta \vdash P \approx_5 Q : I \to O \ \mathsf{wid}\ k\ \mathsf{len}\ l,$$

- *distinguisher* Adv *for protocols of type* $\Delta \vdash I \to O$ *under the interpretation* $[\![-]\!]$ *for which there exist* $K_{\mathsf{adv}} \in \mathbb{N}$ *and* $\eta_{\mathsf{adv}} \in \mathbb{Q}_{\geqslant 0}$ *such that* $|\mathsf{Adv}| \leqslant K_{\mathsf{adv}}$ *and* $\mathsf{err}(\mathsf{Adv}) \leqslant \eta_{\mathsf{adv}}$*,*
- *bound* $K_{\mathsf{len}} \in \mathbb{N}$ *such that* $l(|\mathsf{t}_1|, \ldots, |\mathsf{t}_{|\Sigma_\mathsf{t}|}|) \leqslant K_{\mathsf{len}}$*, and*
- *bounds* $\varepsilon^1, \ldots, \varepsilon^n \in \mathbb{Q}_{\geqslant 0}$ *with the property that for any distinguisher* Adv^i *for protocols of type* $\Delta^i \vdash I^i \to O^i$ *with respect to the interpretation* $[\![-]\!]$ *such that* $|\mathsf{Adv}^i| \leqslant \mathcal{P}(K_{\mathsf{sem}}, K_{\mathsf{adv}}, K_{\mathsf{len}})$ *and* $\mathsf{err}(\mathsf{Adv}^i) \leqslant \mathsf{max}(\eta_{\mathsf{sem}}, \eta_{\mathsf{adv}})$*, we have*

$$\left|\mathsf{Pr}\big[\mathsf{Adv}^i \overset{[\![-]\!]}{\rightleftarrows} P^i = 1\big] - \mathsf{Pr}\big[\mathsf{Adv}^i \overset{[\![-]\!]}{\rightleftarrows} Q^i = 1\big]\right| \leqslant \varepsilon^i,$$

we have

$$\left|\mathsf{Pr}\big[\mathsf{Adv} \overset{[\![-]\!]}{\rightleftarrows} P = 1\big] - \mathsf{Pr}\big[\mathsf{Adv} \overset{[\![-]\!]}{\rightleftarrows} Q = 1\big]\right| \leqslant k * \big(\mathsf{max}(\varepsilon^1, \ldots, \varepsilon^n) + 2 * K_{\mathsf{len}} * \eta_{\mathsf{sem}}\big).$$

Proof. We proceed by induction on the derivation of $\approx_5$. The STRICT rule follows immediately from Lemma **??**. The APPROX-CONG rule follows immediately from Lemma **??**. The SYM rule follows easily from the inductive hypothesis for the premise $\Delta \vdash P_1 \approx_5 P_2 : I \to O\ \mathsf{wid}\ k\ \mathsf{len}\ l$. For the TRANS rule, we use the inductive hypotheses for the two premises $\Delta \vdash P_1 \approx_5 P_2 : I \to O\ \mathsf{wid}\ k_1\ \mathsf{len}\ l_1$ and $\Delta \vdash P_2 \approx_5 P_3 : I \to O\ \mathsf{wid}\ k_2\ \mathsf{len}\ l_2$, both invoked with K_{len}. We can do this because

$$l_1(|\mathsf{t}_1|, \ldots, |\mathsf{t}_{|\Sigma_\mathsf{t}|}|) \leqslant \mathsf{max}\big(l_1(|\mathsf{t}_1|, \ldots, |\mathsf{t}_{|\Sigma_\mathsf{t}|}|), l_2(|\mathsf{t}_1|, \ldots, |\mathsf{t}_{|\Sigma_\mathsf{t}|}|)\big) \leqslant K_{\mathsf{len}},$$

$$l_2(|\mathsf{t}_1|, \ldots, |\mathsf{t}_{|\Sigma_\mathsf{t}|}|) \leqslant \mathsf{max}\big(l_1(|\mathsf{t}_1|, \ldots, |\mathsf{t}_{|\Sigma_\mathsf{t}|}|), l_2(|\mathsf{t}_1|, \ldots, |\mathsf{t}_{|\Sigma_\mathsf{t}|}|)\big) \leqslant K_{\mathsf{len}}.$$

This gives us the two bounds

$$\Big|\mathsf{Pr}\big[\mathsf{Adv} \xrightleftharpoons{\llbracket - \rrbracket} P_1 = 1\big] - \mathsf{Pr}\big[\mathsf{Adv} \xrightleftharpoons{\llbracket - \rrbracket} P_2 = 1\big]\Big| \leqslant k_1 * \big(\mathsf{max}(\varepsilon^1, \dots, \varepsilon^n) + 2 * K_{\mathsf{len}} * \eta_{\mathsf{sem}}\big),$$
$$\Big|\mathsf{Pr}\big[\mathsf{Adv} \xrightleftharpoons{\llbracket - \rrbracket} P_2 = 1\big] - \mathsf{Pr}\big[\mathsf{Adv} \xrightleftharpoons{\llbracket - \rrbracket} P_3 = 1\big]\Big| \leqslant k_2 * \big(\mathsf{max}(\varepsilon^1, \dots, \varepsilon^n) + 2 * K_{\mathsf{len}} * \eta_{\mathsf{sem}}\big).$$

Combining these yields the following:

$$\Big|\mathsf{Pr}\big[\mathsf{Adv} \xrightleftharpoons{\llbracket - \rrbracket} P_1 = 1\big] - \mathsf{Pr}\big[\mathsf{Adv} \xrightleftharpoons{\llbracket - \rrbracket} P_3 = 1\big]\Big| \leqslant (k_1 + k_2) * \big(\mathsf{max}(\varepsilon^1, \dots, \varepsilon^n) + 2 * K_{\mathsf{len}} * \eta_{\mathsf{sem}}\big).$$

□

Corollary 4 (Soundness of approximate equality of protocols)**.** *For any*

- *IPDL signature Σ with type symbols $\mathsf{t}_1, \dots, \mathsf{t}_{|\Sigma_\mathsf{t}|}$,*
- *interpretation $\llbracket - \rrbracket$ for Σ for which there exist $K_{\mathsf{sem}} \in \mathbb{N}$ and $\eta_{\mathsf{sem}} \in \mathbb{Q}_{\geqslant 0}$ such that*
    - *for all type symbols t, $|\mathsf{t}| \leqslant K_{\mathsf{sem}}$,*
    - *for all function symbols f, $\llbracket \mathsf{f} \rrbracket$ is computable by a deterministic Turing Machine with symbols $0, 1$ such that the number of states and the runtime are $\leqslant K_{\mathsf{sem}}$, and*
    - *for all distribution symbols d, $\llbracket \mathsf{d} \rrbracket_\lambda$ is computable up to an error η_{sem} by a probabilistic Turing Machine with symbols $0, 1$ such that the number of states and the runtime are $\leqslant K_{\mathsf{sem}}$,*
- *strict IPDL theory $\mathbb{T}_=$ such that $\llbracket - \rrbracket \vDash \mathbb{T}_=$,*
- *approximate IPDL theory with axioms $\Delta^1 \vdash P^1 \approx Q^1 : I^1 \to O^1, \dots, \Delta^n \vdash P^n \approx Q^n : I^n \to O^n$,*
- *derivation*

$$\mathbb{T}_=;\ \Delta^1 \vdash P^1 \approx Q^1 : I^1 \to O^1, \dots, \Delta^n \vdash P^n \approx Q^n : I^n \to O^n \Rrightarrow \Delta \vdash P \approx Q : I \to O \ \mathsf{wid}\ k\ \mathsf{len}\ l,$$

- *distinguisher Adv for protocols of type $\Delta \vdash I \to O$ under the interpretation $\llbracket - \rrbracket$ for which there exist $K_{\mathsf{adv}} \in \mathbb{N}$ and $\eta_{\mathsf{adv}} \in \mathbb{Q}_{\geqslant 0}$ such that $|\mathsf{Adv}| \leqslant K_{\mathsf{adv}}$ and $\mathsf{err}(\mathsf{Adv}) \leqslant \eta_{\mathsf{adv}}$,*
- *bound $K_{\mathsf{len}} \in \mathbb{N}$ such that $l(|\mathsf{t}_1|, \dots, |\mathsf{t}_{|\Sigma_\mathsf{t}|}|) \leqslant K_{\mathsf{len}}$, and*
- *bounds $\varepsilon^1, \dots, \varepsilon^n \in \mathbb{Q}_{\geqslant 0}$ with the property that for any distinguisher Adv^i for protocols of type $\Delta^i \vdash I^i \to O^i$ with respect to the interpretation $\llbracket - \rrbracket$ such that $|\mathsf{Adv}^i| \leqslant \mathcal{P}(K_{\mathsf{sem}}, K_{\mathsf{adv}}, K_{\mathsf{len}})$ and $\mathsf{err}(\mathsf{Adv}^i) \leqslant \mathsf{max}(\eta_{\mathsf{sem}}, \eta_{\mathsf{adv}})$, we have*
$$\Big|\mathsf{Pr}\big[\mathsf{Adv}^i \xrightleftharpoons{\llbracket - \rrbracket} P^i = 1\big] - \mathsf{Pr}\big[\mathsf{Adv}^i \xrightleftharpoons{\llbracket - \rrbracket} Q^i = 1\big]\Big| \leqslant \varepsilon^i,$$

we have
$$\Big|\mathsf{Pr}\big[\mathsf{Adv} \xrightleftharpoons{\llbracket - \rrbracket} P = 1\big] - \mathsf{Pr}\big[\mathsf{Adv} \xrightleftharpoons{\llbracket - \rrbracket} Q = 1\big]\Big| \leqslant k * \big(\mathsf{max}(\varepsilon^1, \dots, \varepsilon^n) + 2 * K_{\mathsf{len}} * \eta_{\mathsf{sem}}\big).$$

Proof. Follows immediately from Lemmas **??** and **??**. □

We are now ready to prove our main soundness theorem.

Theorem 2 (Soundness of asymptotic equality of protocols)**.** *Assume*

- *an IPDL signature Σ with type symbols $\mathsf{t}_1, \dots, \mathsf{t}_{|\Sigma_\mathsf{t}|}$,*
- *two protocol families $\{\Delta_\lambda \vdash P_\lambda : I_\lambda \to O_\lambda\}_{\lambda \in \mathbb{N}}$ and $\{\Delta_\lambda \vdash Q_\lambda : I_\lambda \to O_\lambda\}_{\lambda \in \mathbb{N}}$ with identical typing judgments,*
- *a PPT family of interpretations $\{\llbracket - \rrbracket_\lambda\}_{\lambda \in \mathbb{N}}$,*
- *a strict IPDL theory $\mathbb{T}_=$ such that for each $\lambda \in \mathbb{N}$, we have $\llbracket - \rrbracket_\lambda \vDash \mathbb{T}_=$, and*

- *an asymptotic IPDL theory $\mathbb{T}_{\approx}$ such that $\{[\![-]\!]_\lambda\}_{\lambda\in\mathbb{N}} \models \mathbb{T}_{\approx}$.*

Then

$$\mathbb{T}_{=}; \mathbb{T}_{\approx} \Rightarrow \{\Delta_\lambda \vdash P_\lambda \approx Q_\lambda : I_\lambda \to O_\lambda\}_{\lambda\in\mathbb{N}}$$

implies

$$\{[\![-]\!]_\lambda\}_{\lambda\in\mathbb{N}} \models \{\Delta_\lambda \vdash P_\lambda \approx Q_\lambda : I_\lambda \to O_\lambda\}_{\lambda\in\mathbb{N}}.$$

Proof. Let $\{\Delta^1_\lambda \vdash P^1_\lambda \approx Q^1_\lambda : I^1_\lambda \to O^1_\lambda\}_{\lambda\in\mathbb{N}}, \ldots, \{\Delta^n_\lambda \vdash P^n_\lambda \approx Q^n_\lambda : I^n_\lambda \to O^n_\lambda\}_{\lambda\in\mathbb{N}}$ be the axioms comprising the theory $\mathbb{T}_{\approx}$. The top-level asymptotic equality judgement

$$\mathbb{T}_{=}; \mathbb{T}_{\approx} \Rightarrow \{\Delta_\lambda \vdash P_\lambda \approx Q_\lambda : I_\lambda \to O_\lambda\}_{\lambda\in\mathbb{N}}$$

gives us functions $k = \mathsf{O}(\mathsf{poly}(\lambda))$ and $l = \mathsf{O}\big(\mathsf{poly}(\lambda, t_1, \ldots, t_{|\Sigma_\mathsf{t}|})\big)$ such that

$$\mathbb{T}_{=}; \Delta^1_\lambda \vdash P^1_\lambda \approx Q^1_\lambda : I^1_\lambda \to O^1_\lambda, \ldots, \Delta^n_\lambda \vdash P^n_\lambda \approx Q^n_\lambda : I^n_\lambda \to O^n_\lambda \Rightarrow \Delta_\lambda \vdash P_\lambda \approx Q_\lambda : I_\lambda \to O_\lambda \text{ wid } k(\lambda) \text{ len } l(\lambda) \quad (\star)$$

In particular, there exists a polynomial $p_{\mathsf{wid}}(\lambda)$ with an index $N_{\mathsf{wid}} \in \mathbb{N}$ such that $k(\lambda) \leqslant p_{\mathsf{wid}}(\lambda)$ if $\lambda \geqslant N_{\mathsf{wid}}$, and a polynomial $p_{\mathsf{len}}(\lambda, t_1, \ldots, t_{|\Sigma_\mathsf{t}|})$ with an index $N_{\mathsf{len}} \in \mathbb{N}$ such that $l(\lambda, t_1, \ldots, t_{|\Sigma_\mathsf{t}|}) \leqslant p_{\mathsf{len}}(\lambda, t_1, \ldots, t_{|\Sigma_\mathsf{t}|})$ if $\lambda \geqslant N_{\mathsf{len}}$ and $t_i \geqslant N_{\mathsf{len}}$. Since the family $\{[\![-]\!]_\lambda\}_{\lambda\in\mathbb{N}}$ of interpretations is PPT, we have a polynomial $K_{\mathsf{sem}}(\lambda)$, a negligible function $\eta_{\mathsf{sem}}(\lambda)$, and a natural number $N_{\mathsf{sem}} \in \mathbb{N}$ such that:

- *for all type symbols* t, $|\mathsf{t}|_\lambda \leqslant K_{\mathsf{sem}}(\lambda)$ *if* $\lambda \geqslant N_{\mathsf{sem}}$,
- *for all function symbols* f, $[\![\mathsf{f}]\!]_\lambda$ *is computable by a deterministic Turing Machine* TM_λ *with symbols* $0, 1$, *and both the number of states and the runtime of* TM_λ *are* $\leqslant K_{\mathsf{sem}}(\lambda)$ *if* $\lambda \geqslant N_{\mathsf{sem}}$, *and*
- *for all distribution symbols* d, $[\![\mathsf{d}]\!]_\lambda$ *is computable up to an error* $\eta_{\mathsf{sem}}(\lambda)$ *by a probabilistic Turing Machine* TM_λ *with symbols* $0, 1$, *and both the number of states and the runtime of* TM_λ *are* $\leqslant K_{\mathsf{sem}}(\lambda)$ *if* $\lambda \geqslant N_{\mathsf{sem}}$.

To prove

$$\{[\![-]\!]_\lambda\}_{\lambda\in\mathbb{N}} \models \{\Delta_\lambda \vdash P_\lambda \approx Q_\lambda : I_\lambda \to O_\lambda\}_{\lambda\in\mathbb{N}},$$

we fix a polynomial $K_{\mathsf{adv}}(\lambda)$ and a negligible function $\eta_{\mathsf{adv}}(\lambda)$. Since $\mathbb{T}_{\approx}$ is sound under the family of interpretations $\{[\![-]\!]_\lambda\}_{\lambda\in\mathbb{N}}$, we have the computational indistinguishability assumption

$$\{[\![-]\!]_\lambda\}_{\lambda\in\mathbb{N}} \models \{\Delta^i_\lambda \vdash P^i_\lambda \approx Q^i_\lambda : I^i_\lambda \to O^i_\lambda\}_{\lambda\in\mathbb{N}}.$$

Define

$$K_{\mathsf{len}}(\lambda) := p_{\mathsf{len}}\big(\lambda, K_{\mathsf{sem}}(\lambda) + N_{\mathsf{len}}, \ldots, K_{\mathsf{sem}}(\lambda) + N_{\mathsf{len}}\big),$$

and apply the computational indistinguishability assumption to the polynomial $\mathcal{P}\big(K_{\mathsf{sem}}(\lambda), K_{\mathsf{adv}}(\lambda), K_{\mathsf{len}}(\lambda)\big)$ and the negligible function $\mathsf{max}\big(\eta_{\mathsf{sem}}(\lambda), \eta_{\mathsf{adv}}(\lambda)\big)$. We get a negligible function $\varepsilon^i(\lambda)$ with a natural number $N^i \in \mathbb{N}$ such that for any $\lambda \geqslant N^i$ and any distinguisher Adv^i for protocols of type $\Delta^i_\lambda \vdash I^i_\lambda \to O^i_\lambda$ under the interpretation $[\![-]\!]_\lambda$ such that $|\mathsf{Adv}^i| \leqslant \mathcal{P}\big(K_{\mathsf{sem}}(\lambda), K_{\mathsf{adv}}(\lambda), K_{\mathsf{len}}(\lambda)\big)$ and $\mathsf{err}(\mathsf{Adv}^i) \leqslant \mathsf{max}\big(\eta_{\mathsf{sem}}(\lambda), \eta_{\mathsf{adv}}(\lambda)\big)$, we have

$$\Big|\mathsf{Pr}\big[\mathsf{Adv} \xleftrightarrow{[\![-]\!]_\lambda} P^i_\lambda = 1\big] - \mathsf{Pr}\big[\mathsf{Adv} \xleftrightarrow{[\![-]\!]_\lambda} Q^i_\lambda = 1\big]\Big| \leqslant \varepsilon^i(\lambda).$$

We can now define our desired negligible function as

$$\varepsilon(\lambda) := k(\lambda) * \big(\mathsf{max}(\varepsilon^1(\lambda), \ldots, \varepsilon^n(\lambda)) + 2 * K_{\mathsf{len}}(\lambda) * \eta_{\mathsf{sem}}(\lambda)\big).$$

The negligibility of $\varepsilon(\lambda)$ follows easily: if $\lambda \geqslant N_{\mathsf{wid}}$ then

$$\varepsilon(\lambda) \leqslant p_{\mathsf{wid}}(\lambda) * \big(\mathsf{max}(\varepsilon^1(\lambda), \ldots, \varepsilon^n(\lambda)) + 2 * K_{\mathsf{len}}(\lambda) * \eta_{\mathsf{sem}}(\lambda)\big),$$

so it suffices to show that this latter function is negligible. But this is immediate from the negligibility of each $\varepsilon^i(\lambda)$, the negligibility of $\eta_{\mathsf{sem}}(\lambda)$, and the fact that $p_{\mathsf{wid}}(\lambda)$ and $K_{\mathsf{len}}(\lambda)$ are polynomials. Define

$$N := \mathsf{max}\big(N_{\mathsf{len}}, N_{\mathsf{sem}}, N^1, \ldots, N^n\big).$$

Assume $\lambda \geqslant N$ and take any distinguisher Adv for protocols of type $\Delta_\lambda \vdash I_\lambda \to O_\lambda$ under the interpretation $[\![-]\!]_\lambda$, such that $|\mathsf{Adv}| \leqslant K_{\mathsf{adv}}(\lambda)$ and $\mathsf{err}(\mathsf{Adv}) \leqslant \eta_{\mathsf{adv}}(\lambda)$. We aim to show that

$$\left|\mathsf{Pr}\big[\mathsf{Adv} \xleftrightarrow{[\![-]\!]_\lambda} P_\lambda = 1\big] - \mathsf{Pr}\big[\mathsf{Adv} \xleftrightarrow{[\![-]\!]_\lambda} Q_\lambda = 1\big]\right| \leqslant k(\lambda) * \big(\mathsf{max}(\varepsilon^1(\lambda), \ldots, \varepsilon^n(\lambda)) + 2 * K_{\mathsf{len}}(\lambda) * \eta_{\mathsf{sem}}(\lambda)\big).$$

But this is precisely the conclusion of Lemma **??** applied to the derivation ($\star$). It thus suffices to prove the hypotheses of Lemma **??**. Among these, the only non-trivial assumption is

$$l\big(\lambda, |\mathsf{t}_1|, \ldots, |\mathsf{t}_{|\Sigma_\mathsf{t}|}|\big) \leqslant K_{\mathsf{len}}(\lambda).$$

We show this via the following sequence of inequalities:

$$\begin{aligned} l\big(\lambda, |\mathsf{t}_1|_\lambda, \ldots, |\mathsf{t}_{|\Sigma_\mathsf{t}|}|_\lambda\big) &\leqslant l\big(\lambda, K_{\mathsf{sem}}(\lambda), \ldots, K_{\mathsf{sem}}(\lambda)\big) \\ &\leqslant l\big(\lambda, K_{\mathsf{sem}}(\lambda) + N_{\mathsf{len}}, \ldots, K_{\mathsf{sem}}(\lambda) + N_{\mathsf{len}}\big) \\ &\leqslant p_{\mathsf{len}}\big(\lambda, K_{\mathsf{sem}}(\lambda) + N_{\mathsf{len}}, \ldots, K_{\mathsf{sem}}(\lambda) + N_{\mathsf{len}}\big) \\ &= K_{\mathsf{len}}. \end{aligned}$$

The first inequality follows from the fact that the function $l(\lambda) : \mathbb{N}^{|\Sigma_\mathsf{t}|} \to \mathbb{N}$ is monotonically increasing in each argument and $|\mathsf{t}_i|_\lambda \leqslant K_{\mathsf{sem}}(\lambda)$ by assumption since $\lambda \geqslant N_{\mathsf{sem}}$. The second inequality is again monotonicity of $l(\lambda)$, and the third follows from the definition of p_{len} since $\lambda \geqslant N_{\mathsf{len}}$ and $K_{\mathsf{sem}}(\lambda) + N_{\mathsf{len}} \geqslant N_{\mathsf{len}}$. □

The remainder of this section is devoted to the proof of the absorption lemma that we promised earlier.

Lemma 33 (Absorption)**.** *There exists a polynomial $\mathcal{P}(x, y, z) \geqslant y$ such that for any*

- *IPDL signature Σ with type symbols $\mathsf{t}_1, \ldots, \mathsf{t}_{|\Sigma_\mathsf{t}|}$,*
- *interpretation $[\![-]\!]$ for Σ for which there exist $K_{\mathsf{sem}} \in \mathbb{N}$ and $\eta_{\mathsf{sem}} \in \mathbb{Q}_{\geqslant 0}$ such that*
  - *for all type symbols t, $|\mathsf{t}| \leqslant K_{\mathsf{sem}}$,*
  - *for all function symbols f, $[\![\mathsf{f}]\!]$ is computable by a deterministic TM with symbols $0, 1$ such that the number of states and the runtime are $\leqslant K_{\mathsf{sem}}$, and*
  - *for all distribution symbols d, $[\![\mathsf{d}]\!]$ is computable up to error η_{sem} by a probabilistic TM with symbols $0, 1$ such that the number of states and the runtime are $\leqslant K_{\mathsf{sem}}$,*
- *distinguisher Adv for protocols of type $\Delta \vdash I \to O_1 \cup O_2$ under the interpretation $[\![-]\!]$ for which there exist $K_{\mathsf{adv}} \in \mathbb{N}$ and $\eta_{\mathsf{adv}} \in \mathbb{Q}_{\geqslant 0}$ such that $|\mathsf{Adv}| \leqslant K_{\mathsf{adv}}$ and $\mathsf{err}(\mathsf{Adv}) \leqslant \eta_{\mathsf{adv}}$,*
- *protocol $\Delta \vdash Q : I \cup O_1 \to O_2$,*

we have a new distinguisher $\mathsf{Adv}_{\mathcal{R}}$ for protocols of type $\Delta \vdash I \cup O_2 \to O_1$ with

$$|\mathsf{Adv}_{\mathcal{R}}| \leqslant \mathcal{P}\big(K_{\mathsf{sem}}, K_{\mathsf{adv}}, \|Q\|_{\mathsf{TM}}(|\mathsf{t}_1|, \ldots, |\mathsf{t}_{|\Sigma_\mathsf{t}|}|)\big)$$

and $\mathsf{err}(\mathsf{Adv}_{\mathcal{R}}) \leqslant \mathsf{max}(\eta_{\mathsf{sem}}, \eta_{\mathsf{adv}})$ such that for any protocol $\Delta \vdash P : I \cup O_2 \to O_1$ we have

$$\left|\mathsf{Pr}\big[\mathsf{Adv} \xleftrightarrow{[\![-]\!]} P \parallel Q = 1\big] - \mathsf{Pr}\big[\mathsf{Adv}_{\mathcal{R}} \xleftrightarrow{[\![-]\!]} P = 1\big]\right| \leqslant \|Q\|_{\mathsf{TM}}(|\mathsf{t}_1|, \ldots, |\mathsf{t}_{|\Sigma_\mathsf{t}|}|) * \eta_{\mathsf{sem}}.$$

To encode IPDL protocols on a Turing Machine tape, we will make use of the following sets of symbols:

- Punc with symbols “⟨”, “⟩”, “(”, “)”, “{”, “}”, “[”, ‘]”, “_”, “:”, “·”, “;”, “→”, “↠”, “←”, “×”, “:=”, “‖”,
- KeyWords with symbols “var”, “✓”, “true”, “false”, “app”, “fst”, “snd”, “of”, “val”, “ret”, “samp”, “read”, “input-to-query”, “input-queried”, “input-not-to-query”, “if”, “then”, “else”, “0”, “new”, “in”, “wen”.

We will also need a finite set of de Bruijn indices in lieu of channel and variable names. To derive an upper bound on how many indices we will need, we statically count the maximum depth of variable and channel declarations, giving us a *variable-index bound* and a *channel-index bound*, respectively. The variable-index bound for reactions is invariant under substitutions, embeddings, and input assignment.

$$\begin{aligned}
\|\mathsf{val}\ v\|_{\mathsf{var}} &:= 0\\
\|\mathsf{ret}\ e\|_{\mathsf{var}} &:= 0\\
\|\mathsf{samp}\ \mathsf{d}\ e\|_{\mathsf{var}} &:= 0\\
\|\mathsf{read}\ c\|_{\mathsf{var}} &:= 0\\
\|\mathsf{if}\ e\ \mathsf{then}\ R_1\ \mathsf{else}\ R_2\|_{\mathsf{var}} &:= \mathsf{max}(\|R_1\|_{\mathsf{var}}, \|R_2\|_{\mathsf{var}})\\
\|x \leftarrow R;\ S\|_{\mathsf{var}} &:= \mathsf{max}(\|R\|_{\mathsf{var}}, \|S\|_{\mathsf{var}} + 1)
\end{aligned}$$

The variable-index bound for protocols is invariant under embeddings and input assignment.

$$\begin{aligned}
\|\mathsf{0}\|_{\mathsf{var}} &:= 0\\
\|o := v\|_{\mathsf{var}} &:= 0\\
\|o := R\|_{\mathsf{var}} &:= \|R\|_{\mathsf{var}}\\
\|P \parallel Q\|_{\mathsf{var}} &:= \mathsf{max}(\|P\|_{\mathsf{var}}, \|Q\|_{\mathsf{var}})\\
\|\mathsf{new}\ c : \tau\ \mathsf{in}\ P\|_{\mathsf{var}} &:= \|P\|_{\mathsf{var}}
\end{aligned}$$

Likewise, the channel-index bound for protocols is invariant under embeddings and input assignment.

$$\begin{aligned}
\|\mathsf{0}\|_{\mathsf{chan}} &:= 0\\
\|o := v\|_{\mathsf{chan}} &:= 0\\
\|o := R\|_{\mathsf{chan}} &:= 0\\
\|P \parallel Q\|_{\mathsf{chan}} &:= \mathsf{max}(\|P\|_{\mathsf{chan}}, \|Q\|_{\mathsf{chan}})\\
\|\mathsf{new}\ c : \tau\ \mathsf{in}\ P\|_{\mathsf{chan}} &:= \|P\|_{\mathsf{chan}} + 1
\end{aligned}$$

To avoid an infinite loop, an adversary executing the absorbed protocol will need to keep track of which channels have already been queried for a value. We store this information inside the protocol in the form of an annotation: for each channel read $\mathsf{read}(c : \tau)$, we denote whether the channel c has already been queried for a value, if applicable:

$$\begin{array}{llll}
\text{Query Annotations} & a & ::= & \mathsf{input\text{-}to\text{-}query} \mid \mathsf{input\text{-}queried} \mid \mathsf{input\text{-}not\text{-}to\text{-}query}\\
\text{Query-Annotated Reactions} & R, S & ::= & \ldots \mid \mathsf{read}[a](c : \tau) \mid \ldots\\
\text{Query-Annotated Protocols} & P, Q & ::= & \ldots \mid o := R \mid \ldots
\end{array}$$

By erasing the annotations from a query-annotated reaction or protocol, we obtain the underlying (valued) IPDL construct.

Given an ambient interpretation $[\![-]\!]$ for the IPDL signature Σ, we now show how to encode IPDL constructs as a sequence of symbols on a Turing Machine tape. For types, the encoding $\mathsf{Enc}_{\mathsf{TM}}[\tau]$ consists of the symbol “$\cdot$” repeated $|\tau|$-many times. For expressions, we have the encoding below, where + denotes string concatenation. We recall that each variable name is represented as a de Bruijn index, and is in particular a natural number.

$$\begin{aligned}
\mathsf{Enc}_{\mathsf{TM}}[v] &:= v\\
\mathsf{Enc}_{\mathsf{TM}}[\mathsf{var}(x : \tau)] &:= \text{“(”} + \text{“}\mathsf{var}\text{”} + x + \text{“:”} + \mathsf{Enc}_{\mathsf{TM}}[\tau] + \text{“)”}\\
\mathsf{Enc}_{\mathsf{TM}}[\checkmark] &:= \text{“(”} + \text{“}\checkmark\text{”} + \text{“)”}\\
\mathsf{Enc}_{\mathsf{TM}}[\mathsf{true}] &:= \text{“(”} + \text{“}\mathsf{true}\text{”} + \text{“)”}\\
\mathsf{Enc}_{\mathsf{TM}}[\mathsf{false}] &:= \text{“(”} + \text{“}\mathsf{false}\text{”} + \text{“)”}\\
\mathsf{Enc}_{\mathsf{TM}}[\mathsf{app}_{\sigma \to \tau}\ \mathsf{f}\ e] &:= \text{“(”} + \text{“}\mathsf{app}\text{”} + \mathsf{Enc}_{\mathsf{TM}}[\sigma] + \text{“}\rightarrow\text{”} + \mathsf{Enc}_{\mathsf{TM}}[\tau] + \mathsf{f} + \mathsf{Enc}_{\mathsf{TM}}[e] + \text{“)”}\\
\mathsf{Enc}_{\mathsf{TM}}[(e_1, e_2)] &:= \mathsf{Enc}_{\mathsf{TM}}[e_1] + \mathsf{Enc}_{\mathsf{TM}}[e_2]\\
\mathsf{Enc}_{\mathsf{TM}}[\mathsf{fst}_{\sigma \times \tau}\ e] &:= \text{“(”} + \text{“}\mathsf{fst}\text{”} + \mathsf{Enc}_{\mathsf{TM}}[\sigma] + \text{“}\times\text{”} + \mathsf{Enc}_{\mathsf{TM}}[\tau] + \text{“}\mathsf{of}\text{”} + \mathsf{Enc}_{\mathsf{TM}}[e] + \text{)”}\\
\mathsf{Enc}_{\mathsf{TM}}[\mathsf{snd}_{\sigma \times \tau}\ e] &:= \text{“(”} + \text{“}\mathsf{snd}\text{”} + \mathsf{Enc}_{\mathsf{TM}}[\sigma] + \text{“}\times\text{”} + \mathsf{Enc}_{\mathsf{TM}}[\tau] + \text{“}\mathsf{of}\text{”} + \mathsf{Enc}_{\mathsf{TM}}[e] + \text{)”}
\end{aligned}$$

The encoding $\mathsf{Enc}_{\mathsf{TM}}[a]$ of an annotation $a = \mathsf{input\text{-}to\text{-}query}, \mathsf{input\text{-}queried}$, or $\mathsf{input\text{-}not\text{-}to\text{-}query}$ is the single symbol "$\mathsf{input\text{-}to\text{-}query}$", "$\mathsf{input\text{-}queried}$", or "$\mathsf{input\text{-}not\text{-}to\text{-}query}$", respectively. For reactions, we have the following encoding, where we recall that each channel name is represented as a de Bruijn index, and is in particular a natural number.

$$\begin{aligned}
\mathsf{Enc}_{\mathsf{TM}}[\mathsf{val}\ v] &:= \text{“}\langle\text{”} + \text{“}\mathsf{val}\text{”} + v + \text{“}\rangle\text{”}\\
\mathsf{Enc}_{\mathsf{TM}}[\mathsf{ret}\ e] &:= \text{“(”} + \text{“}\mathsf{ret}\text{”} + \mathsf{Enc}_{\mathsf{TM}}[e] + \text{“)”}\\
\mathsf{Enc}_{\mathsf{TM}}[\mathsf{samp}_{\sigma \twoheadrightarrow \tau}\ \mathsf{d}\ e] &:= \text{“(”} + \text{“}\mathsf{samp}\text{”} + \mathsf{Enc}_{\mathsf{TM}}[\sigma] + \text{“}\twoheadrightarrow\text{”} + \mathsf{Enc}_{\mathsf{TM}}[\tau] + \mathsf{d} + \mathsf{Enc}_{\mathsf{TM}}[e] + \text{“)”}\\
\mathsf{Enc}_{\mathsf{TM}}[\mathsf{read}[a](c : \tau)] &:= \text{“(”} + \text{“}\mathsf{read}\text{”} + \mathsf{Enc}_{\mathsf{TM}}[a] + c + \text{“:”} + \mathsf{Enc}_{\mathsf{TM}}[\tau] + \text{“)”}\\
\mathsf{Enc}_{\mathsf{TM}}[\mathsf{if}\ e\ \mathsf{then}\ R_1\ \mathsf{else}\ R_2] &:= \text{“(”} + \text{“}\mathsf{if}\text{”} + \mathsf{Enc}_{\mathsf{TM}}[e] + \text{“}\mathsf{then}\text{”} + \mathsf{Enc}_{\mathsf{TM}}[R_1] + \text{“}\mathsf{else}\text{”} + \mathsf{Enc}_{\mathsf{TM}}[R_2] + \text{“)”}\\
\mathsf{Enc}_{\mathsf{TM}}[x : \sigma \leftarrow R;\ S] &:= \text{“\{”} + \text{“\_”} + \text{“:”} + \mathsf{Enc}_{\mathsf{TM}}[\sigma] + \text{“}\leftarrow\text{”} + \mathsf{Enc}_{\mathsf{TM}}[R] + \text{“;”} + \mathsf{Enc}_{\mathsf{TM}}[S] + \text{“\}”}
\end{aligned}$$

Finally, for protocols we have the encoding below.

$$\begin{aligned}
\mathsf{Enc}_{\mathsf{TM}}[0] &:= \text{“}\mathsf{0}\text{”}\\
\mathsf{Enc}_{\mathsf{TM}}[o := v] &:= \text{“[”} + o + \text{“:=”} + v + \text{“]”}\\
\mathsf{Enc}_{\mathsf{TM}}[o := R] &:= \text{“(”} + o + \text{“:=”} + \text{“}\mathsf{react}\text{”} + \mathsf{Enc}_{\mathsf{TM}}[R] + \text{“)”}\\
\mathsf{Enc}_{\mathsf{TM}}[P \parallel Q] &:= \text{“(”} + \mathsf{Enc}_{\mathsf{TM}}[P] + \text{“}\parallel\text{”} + \mathsf{Enc}_{\mathsf{TM}}[Q] + \text{“)”}\\
\mathsf{Enc}_{\mathsf{TM}}[\mathsf{new}\ c : \tau\ \mathsf{in}\ P] &:= \text{“}\mathsf{new}\text{”} + \text{“\_”} + \text{“:”} + \mathsf{Enc}_{\mathsf{TM}}[\tau] + \text{“}\mathsf{in}\text{”} + \mathsf{Enc}_{\mathsf{TM}}[P] + \text{“}\mathsf{wen}\text{”}
\end{aligned}$$

To avoid having to shift the tape contents when executing IPDL protocols on a Turing Machine tape, we will make use of the white-space symbol " ", which we consider as distinct from the symbol *blank*. The former will be used as a placeholder so that our protocol encoding remains at a constant length throughout the execution. For this reason, we extend our notion of encoding to allow extra white-spaces around the encoding of a valued expression e or a query-annotated reaction R occurring inside a query-annotated protocol P.

Given a query-annotated IPDL construct, its *query bound* $\| - \|_{\mathsf{query}}$ is the number of occurrences of the annotation $\mathsf{input\text{-}to\text{-}query}$ inside the construct. Furthermore, given a channel set C, we define $\mathsf{QueryAnn}_C[R]$ and $\mathsf{QueryAnn}_C[P]$ to be the query-annotated version of R and P that annotates every read from a channel in C with the annotation $\mathsf{input\text{-}to\text{-}query}$ and every read from a channel not in C with the annotation $\mathsf{input\text{-}not\text{-}to\text{-}query}$. Additionally, given a channel set C, we define the minimal set $\mathsf{MinQueryIn}_C[R] \subseteq C$ of query inputs to the reaction R as follows:

$$\begin{aligned}
\mathsf{MinQueryIn}_C[\mathsf{val}\ v] &:= \varnothing\\
\mathsf{MinQueryIn}_C[\mathsf{ret}\ e] &:= \varnothing\\
\mathsf{MinQueryIn}_C[\mathsf{samp}\ \mathsf{d}\ e] &:= \varnothing\\
\mathsf{MinQueryIn}_C[\mathsf{read}\ c] &:= \{c\} \text{ if } c \in C\\
\mathsf{MinQueryIn}_C[\mathsf{read}\ i] &:= \{\varnothing\} \text{ if } i \notin C\\
\mathsf{MinQueryIn}_C[\mathsf{if}\ e\ \mathsf{then}\ R_1\ \mathsf{else}\ R_2] &:= \mathsf{MinQueryIn}_C[R_1] \bigcup \mathsf{MinQueryIn}_C[R_2]\\
\mathsf{MinQueryIn}_C[x \leftarrow R;\ S] &:= \mathsf{MinQueryIn}_C[R] \bigcup \mathsf{MinQueryIn}_C[S]
\end{aligned}$$

Similarly, given a channel set C, we define the minimal set $\mathsf{MinQueryIn}_C[P] \subseteq C$ of query inputs to the protocol P as follows:

$$\begin{aligned}
\mathsf{MinQueryIn}_C[0] &:= \varnothing\\
\mathsf{MinQueryIn}_C[o := v] &:= \varnothing\\
\mathsf{MinQueryIn}_C[o := R] &:= \mathsf{MinQueryIn}_C[R]\\
\mathsf{MinQueryIn}_C[P \parallel Q] &:= \mathsf{MinQueryIn}_C[P] \bigcup \mathsf{MinQueryIn}_C[Q]\\
\mathsf{MinQueryIn}_C[\mathsf{new}\ c : \tau\ \mathsf{in}\ P] &:= \mathsf{MinQueryIn}_C[P]
\end{aligned}$$

The typing of query-annotated IPDL constructs has the form $\Delta;\ \Gamma \vdash R : I \to \tau\ \mathsf{query}\ C$ and $\Delta \vdash P : I \to O\ \mathsf{query}\ C$, where $C \subseteq I$ is the set of *query inputs*. For reactions, reading from a query input must be annotated with either

$\mathsf{input\text{-}to\text{-}query}$ or $\mathsf{input\text{-}queried}$, and reading from any other channel must carry the annotation $\mathsf{input\text{-}not\text{-}to\text{-}query}$.

$$\frac{c : \tau \in \Delta \qquad c \in C \qquad a \in \{\mathsf{input\text{-}to\text{-}query}, \mathsf{input\text{-}queried}\}}{\Delta;\ \Gamma \vdash \mathsf{read}[a](c : \tau) : I \to \tau \ \mathsf{query}\ C}$$

$$\frac{i : \tau \in \Delta \qquad i \in I \setminus C \qquad a \in \{\mathsf{input\text{-}not\text{-}to\text{-}query}\}}{\Delta;\ \Gamma \vdash \mathsf{read}[a](i : \tau) : I \to \tau \ \mathsf{query}\ C}$$

For protocols, the interesting rules are shown below, where we propagate the set of query inputs throughout.

$$\frac{o \notin I \qquad o : \tau \in \Delta \qquad \Delta;\ \cdot \vdash R : I \cup \{o\} \to \tau \ \mathsf{query}\ C}{\Delta \vdash \big(o := R\big) : I \to \{o\} \ \mathsf{query}\ C}$$

$$\frac{\Delta \vdash P : I \cup O_2 \to O_1 \ \mathsf{query}\ C \qquad \Delta \vdash Q : I \cup O_1 \to O_2 \ \mathsf{query}\ C}{\Delta \vdash P \parallel Q : I \to O_1 \cup O_2 \ \mathsf{query}\ C}$$

$$\frac{\Delta, o : \tau \vdash P : I \to O \cup \{o\} \ \mathsf{query}\ C}{\Delta \vdash \big(\mathsf{new}\ o : \tau\ \mathsf{in}\ P\big) : I \to O \ \mathsf{query}\ C}$$

Proof. Let $O_1^{\mathsf{min}} = \mathsf{MinQueryIn}_{O_1}[Q]$ be the minimal set of query inputs to Q from among O_1. In other words, O_1^{min} contains precisely those channels of O_1 that Q reads from. Then $\Delta \vdash \mathsf{QueryAnn}_{O_1}[Q] : I \cup O_1^{\mathsf{min}} \to Q_2 \ \mathsf{query}\ O_1^{\mathsf{min}}$. The reason for replacing O_1 with O_1^{min} is that we do not have a bound on the size of the former. The size of the latter is bounded by the query bound $\|Q\|_{\mathsf{query}}$, and this is in turn bounded by $\|Q\|_{\mathsf{TM}}(|\mathsf{t}_1|, \ldots, |\mathsf{t}_{|\Sigma_\mathsf{t}|}|)$. Let $\mathsf{ProtEncSymb}$ be the disjoint union of the sets

- $\{\text{“ ”}\}$,
- Punc,
- $\mathsf{KeyWords}$,
- *the set* Σ_f *of function symbols declared in* Σ,
- *the set* Σ_d *of distribution symbols declared in* Σ,
- $\big\{n \mid 0 \leqslant n < \|Q\|_{\mathsf{var}}\big\}$,
- $\big\{n \mid 0 \leqslant n < \|Q\|_{\mathsf{chan}}\big\} \bigsqcup \big\{m + n \mid m \in \phi^\star(I \cup O_1^{\mathsf{min}} \cup O_2) \text{ and } 0 \leqslant n \leqslant \|Q\|_{\mathsf{chan}}\big\}$.

Let $\mathsf{Adv} := \big(\Delta', I', O', \phi, \#_{\mathsf{round}}, \#_{\mathsf{tape}}, \mathsf{Symb}, \mathsf{St}, s_\star, \mathsf{T}, \{\mathsf{I}_o\}_{o \in I'}, \{\mathsf{O}_i\}_{i \in O'}, \mathsf{D}\big)$ be the adversary for protocols of type $\Delta \vdash I \to O_1 \cup O_2$. We define a new adversary $\mathsf{Adv}_{\mathcal{R}}$ for protocols of type $\Delta \vdash I \cup O_2 \to O_1$ as follows:

$$\mathsf{Adv}_{\mathcal{R}} := \big(\Delta', I'_{\mathcal{R}}, O'_{\mathcal{R}}, \phi, \#^{\mathcal{R}}_{\mathsf{round}}, \#^{\mathcal{R}}_{\mathsf{tape}}, \mathsf{Symb}^{\mathcal{R}}, \mathsf{St}^{\mathcal{R}}, s^{\mathcal{R}}_\star, \mathsf{T}^{\mathcal{R}}, \{\mathsf{I}^{\mathcal{R}}_o\}_{o \in I'_{\mathcal{R}}}, \{\mathsf{O}^{\mathcal{R}}_i\}_{i \in O'_{\mathcal{R}}}, \mathsf{D}^{\mathcal{R}}\big),$$

where

- $I'_{\mathcal{R}} := (I' \setminus \phi^\star(O_2)) \cup (\phi^\star(O_1^{\mathsf{min}}) \setminus I')$;
- $O'_{\mathcal{R}} := O' \cup \phi^\star(O_2)$;
- $\#^{\mathcal{R}}_{\mathsf{round}} := \#_{\mathsf{round}} * \|Q\|_{\mathsf{TM}}(|\mathsf{t}_1|, \ldots, |\mathsf{t}_{|\Sigma_\mathsf{t}|}|)^2 + \#_{\mathsf{round}}$;
- $\#^{\mathcal{R}}_{\mathsf{tape}} := \#_{\mathsf{tape}} + 1$;
- $\mathsf{Symb}^{\mathcal{R}} := \mathsf{ProtEncSymb} \bigsqcup \{\text{“}\rightleftharpoons\text{”}, \text{“}\#\text{”}\} \bigsqcup \mathsf{Symb}$;
- $\mathsf{St}^{\mathcal{R}} := \big\{\text{“(”} + b + s_{\mathsf{prot}} + \text{“}\rightleftharpoons\text{”} + s_{\mathsf{adv}} + \text{“)”}\big\}$ *contains strings of the specified form, where* $b \in \{0, 1\}$, $s_{\mathsf{adv}} \in \mathsf{St}$ *is a distinguisher state, and* s_{prot} *is a string of length* $\|Q\|_{\mathsf{TM}}(|\mathsf{t}_1|, \ldots, |\mathsf{t}_{|\Sigma_\mathsf{t}|}|)$ *with symbols drawn from* $\mathsf{ProtEncSymb}$, *that encodes a protocol* $\phi^\star(Q')$, *where* $\Delta \vdash Q' : I \cup O_1^{\mathsf{min}} \to O_2 \ \mathsf{query}\ O_1^{\mathsf{min}}$ *is such that* $\|Q'\|_{\mathsf{var}} \leqslant \|Q\|_{\mathsf{var}}$ *and* $\|Q'\|_{\mathsf{chan}} \leqslant \|Q\|_{\mathsf{chan}}$;

- $s^{\mathcal{R}}_{\star} :=$ “(” + “1” + $\mathsf{Enc}_{\mathsf{TM}}[\phi^{\star}(\mathsf{QueryAnn}_{O_1}[Q])]$ + “⇌” + $s_{\star}$ + “)”;

- $\mathsf{D}^{\mathcal{R}}$ *processes a state of the form* “(” + b + s_{prot} + “⇌” + s_{adv} + “)” *by applying* D *to* s_{adv};

- $\mathsf{O}^{\mathcal{R}}_{i}$ *for* $i \in O'$ *processes a state of the form* “(” + b + s_{prot} + “⇌” + s_{adv} + “)” *by applying* O_i *to* s_{adv};

- $\mathsf{O}^{\mathcal{R}}_{o_2}$ *for* $o_2 \in \phi^{\star}(O_2)$ *processes a state of the form* “(” + b + s_{prot} + “⇌” + s_{adv} + “)” *by traversing through* s_{prot} *to locate the assignment to channel* o_2*. If* o_2 *is assigned a value* v*, it returns* v*; otherwise it returns* $\perp$;

- $\mathsf{I}^{\mathcal{R}}_{o_1}$ *for* $o_1 \in I' \setminus \phi^{\star}(O_2) \setminus \phi^{\star}(O_1^{\mathsf{min}})$ *processes a state of the form* “(” + 0 + s_{prot} + “⇌” + s_{adv} + “)” *by applying* I_{o_1} *to* s_{adv};

- $\mathsf{I}^{\mathcal{R}}_{o_1}$ *for* $o_1 \in I' \setminus \phi^{\star}(O_2) \setminus \phi^{\star}(O_1^{\mathsf{min}})$ *leaves a state of the form* “(” + 1 + s_{prot} + “⇌” + s_{adv} + “)” *unchanged*;

- $\mathsf{I}^{\mathcal{R}}_{o_1}$ *for* $o_1 \in \phi^{\star}(O_1^{\mathsf{min}}) \setminus I'$ *leaves a state of the form* “(” + 0 + s_{prot} + “⇌” + s_{adv} + “)” *unchanged*;

- $\mathsf{I}^{\mathcal{R}}_{o_1}$ *for* $o_1 \in \phi^{\star}(O_1^{\mathsf{min}}) \setminus I'$ *processes a state of the form* “(” + 1 + s_{prot} + “⇌” + s_{adv} + “)” *by traversing through* s_{prot} *and replacing every* read *from the channel* o_1 *by the assigned input value* v;

- $\mathsf{I}^{\mathcal{R}}_{o_1}$ *for* $o_1 \in I' \setminus \phi^{\star}(O_2) \cap \phi^{\star}(O_1^{\mathsf{min}})$ *processes a state of the form* “(” + 0 + s_{prot} + “⇌” + s_{adv} + “)” *by applying* I_{o_1} *to* s_{adv};

- $\mathsf{I}^{\mathcal{R}}_{o_1}$ *for* $o_1 \in I' \setminus \phi^{\star}(O_2) \cap \phi^{\star}(O_1^{\mathsf{min}})$ *processes a state of the form* “(” + 1 + s_{prot} + “⇌” + s_{adv} + “)” *by traversing through* s_{prot} *and replacing every* read *from the channel* o_1 *by the assigned input value* v;

□

References

[1] R. Milner, J. Parrow, and D. Walker, “A Calculus of Mobile Processes, I,” *Information and Computation*, vol. 100, no. 1, pp. 1–40, 1992.