Decidable Inductive Invariants for Verification of Cryptographic Protocols with Unbounded Sessions

Emanuele D'Osualdo
Imperial College London
e.dosualdo@ic.ac.uk

Felix M. Stutz
University of Saarland
fstutz@mpi-inf.mpg.de

Abstract

We develop a theory of inductive invariants for automatic verification of cryptographic protocols with unbounded sessions/nonces, expressed in a variant of the applied π-calculus. Invariants can prove trace properties of protocols such as secrecy of messages. Our main contribution is a class of expressions that are sound and complete for representing infinite sets of configurations of depth-bounded protocols. We provide direct algorithms to prove that some expression is an inductive invariant for a protocol. Inductive invariants of this form can be inferred, and represent an independently checkable certificate of correctness. We implemented the approach in a prototype tool that we used over some illustrative examples.

2012 ACM Subject Classification Theory of computation → Program verification

Keywords and phrases Security Protocols, Infinite-state Verification, Ideal Completions for WSTS

1 Introduction

Security protocols are distributed programs that implement secure communication over insecure channels, by using cryptography. Despite currently underpinning virtually every communication over the internet, new flaws that compromise security are routinely discovered in deployed systems. Many of these flaws do not require breaking the cryptographic primitives, but instead exploit mistakes in the logic of their use in the protocol. The main challenge is to prove that a protocol will be secure even against any possible attack of an intruder that has full control of the insecure channel. Protocols are typically run in parallel by many participants, who create many interleaved sessions of the protocol generating unboundedly many distinct nonces/encryption keys. This leads easily to Turing-complete models of computation, for which any verification problem is undecidable.

A number of dedicated automatic verification tools has been developed to catch these programming bugs early in the design. They remedy to the obstacle of undecidability in two ways: either they consider boundedly many sessions, or they give approximate answers without termination guarantees. Several tools (e.g. DEEPSEC [7], AVISPA [1], SPEC [17], AKISS [6]) apply model checking techniques to analyse protocols under the assumption of a given bound on the number of sessions/nonces. This restriction makes many verification problems decidable, supporting powerful classes of equivalence properties and cryptographic primitives, at the price of missing possible attacks.

The ProVerif [2], Tamarin [14], and Maude NPA [12] tools support unbounded sessions: as a consequence of dealing with an undecidable problem, all of them may not terminate and use over-approximations that may cause false positives.

Our contributions In this paper we start a systematic study of the structure of cryptographic protocols with unbounded sessions/nonces, by contributing a principled theory of decidable inductive invariants for a rich class of protocols.

To verify correctness of a protocol, it is often necessary to establish a number of auxiliary properties of the reachable configurations, from which one builds some further correctness argument. For example, some equivalence property may hinge on the fact that some key k can never be leaked to the intruder, an example of invariant of the protocol. Formally, we call an *invariant* any set that includes the reachable configurations. Our methodology is based on proving properties using *inductive invariants*, i.e. invariants that are closed under protocol steps. Our main result is that, for a rich class of protocols with unbounded sessions, it is decidable whether a set is an inductive invariant for a protocol. Furthermore, the method supports inference of inductive invariants starting from the description of the protocol.

To develop our theory of invariants, we make several technical contributions: (1) we generalise the class of depth-bounded processes of D'Osualdo, Ong & Tiu [11], by parametrising it in the choice of cryptographic primitives; (2) we prove that a class of expressions we call *limits* is sound and complete to represent downward-closed depth-bounded invariants; (3) we show decidability of inductiveness and use it to construct algorithms to decide secrecy and boundedness. We built a proof-of-concept prototype tool and verified benchmark protocols.

Our approach has a number of notable properties. First, once a suitable inductive invariant has been found, it can be provided as a certificate of correctness that can be independently checked. Second, the search of a suitable invariant can be performed both automatically (with a trade-off between precision and performance) or interactively. Third, supporting unbounded nonces makes it possible to reason about properties like susceptibility to known-/chosen-plaintext attacks (Section 3.1). Finally, even coarse invariants inferred with our method can be used to prune the search space of other model checking procedures.

Related work We focus on methods that handle unbounded sessions. In ProVerif, Tamarin and Maude NPA, as a consequence of non-termination and over-approximations, there is no clear and general characterisation of the protocols for which a precise answer is guaranteed. Our contribution can be seen as a first step in establishing a solid protocol class for which termination and precision can be proven. Additionally, the invariants produced by our methodology can be independently checked and provide a finite representation for over-approximations of the reachable configurations; they can in principle be used by other tools, to prune the search space without bounding sessions.

There has been work on proving security properties using type systems [10, 8, 9]. Types try to capture and generalise common safe usages of cryptographic primitives, and reduce verification to constraints which can be solved efficiently. We speculate that our domain of limits and the associated algorithms could be used to define an expressive class of solvable constraints that could be integrated in type systems.

Manual verification using invariants was pioneered by [15], with a method mechanised in Isabelle/HOL. We contribute decidable invariants that can be integrated in this methodology.

Our theory of invariants is framed in terms of ideal completions for well-structured transition systems [13], which, to the best of our knowledge, has not been linked to cryptographic protocols before. Our decidability proofs introduce substantial new proof techniques to deal with an active intruder while being parametric on the cryptographic primitives. Even stripped of the complexity of handling cryptography, when applied to the simpler case of the pure π-calculus, our results contribute new, more direct and easy to implement algorithms for verifying depth-bounded concurrent systems, compared to e.g. [18].

Outline Section 2 introduces the formal model and our generalisation of depth-bounded protocols. Section 3 presents our main theoretical results. In Section 4 we discuss implementation and optimisations, and report on experiments with our tool. Section 5 discusses limitations and future work. The omitted proofs and definitions can be found in the Appendix.

2 Formal model

We introduce a variant of the Applied π-calculus as our formal model of protocols. Following the Dolev-Yao intruder model, we treat cryptographic primitives algebraically. Let $\mathcal{N}$ be an enumerable set of *names* $a, b, \cdots \in \mathcal{N}$, which here will abstractly model data, nonces, and encryption keys. A signature Σ of *constructors*, is a finite set of symbols f with their arity $\mathrm{ar}(\mathsf{f}) \in \mathbb{N}$. The set of messages over Σ, is the smallest set $\mathbb{M}^\Sigma$ which contains all names, and such that if $\mathsf{f} \in \Sigma$ with $\mathrm{ar}(\mathsf{f}) = n$ and $M_1, \ldots, M_n \in \mathbb{M}^\Sigma$ then $\mathsf{f}(M_1, \ldots, M_n) \in \mathbb{M}^\Sigma$. We define $\mathrm{size}(a) := 1$, $\mathrm{size}(\mathsf{f}(M_1, \ldots, M_n)) := 1 + \max\{\mathrm{size}(M_1), \ldots, \mathrm{size}(M_n)\}$, and $\mathrm{names}(a) := \{a\}$, $\mathrm{names}(\mathsf{f}(M_1, \ldots, M_n)) := \mathrm{names}(M_1) \cup \ldots \cup \mathrm{names}(M_n)$. Given $X \subseteq \mathcal{N}$ and $s \in \mathbb{N}$, we define $\mathbb{M}_s^{\Sigma,X} := \{M \in \mathbb{M}^\Sigma \mid \mathrm{names}(M) \subseteq X, \mathrm{size}(M) \leq s\}$. We write $\mathbb{K}^\Sigma$ for the set of finite sets of messages, $\mathbb{K}^\Sigma := \mathscr{P}_f(\mathbb{M}^\Sigma)$; Γ, Γ' and Γ, M stand for $\Gamma \cup \Gamma'$ and $\Gamma \cup \{M\}$ respectively.

A *substitution* is a finite partial function $\theta \colon \mathcal{N} \rightharpoonup \mathbb{M}^\Sigma$; we write $\theta = [M_1/x_1, \ldots, M_n/x_n]$, often abbreviated with $[\vec{M}/\vec{x}]$, for the substitution with $\theta(x_i) = M_i$ for all $1 \leq i \leq n$. We write $M\theta$ for the application of substitution θ to the message M, and extend the notation to sets of messages $\Gamma\theta := \{M\theta \mid M \in \Gamma\}$. A substitution θ is a *renaming of* $X \subseteq \mathcal{N}$ if it is defined on X, injective, and with $\theta(X) \subseteq \mathcal{N}$.

▶ **Definition 1** (Intruder model). *A* derivability relation *for a signature Σ, is a relation $\vdash \subseteq \mathbb{K}^\Sigma \times \mathbb{M}^\Sigma$. The pair $\mathbb{I} = (\Sigma, \vdash)$ is an* (effective) intruder model *if $\vdash$ is a (decidable) derivability relation for Σ, and for all $M, N \in \mathbb{M}^\Sigma$, $\Gamma, \Gamma' \in \mathbb{K}^\Sigma$, $a \in \mathcal{N}$:*

$$M \vdash M \tag{Id}$$
$$\Gamma \subseteq \Gamma' \wedge \Gamma \vdash M \implies \Gamma' \vdash M \tag{Mon}$$
$$\Gamma \vdash M \wedge \Gamma, M \vdash N \implies \Gamma \vdash N \tag{Cut}$$
$$M_1, \ldots, M_n \vdash \mathsf{f}(M_1, \ldots, M_n) \quad \textit{for every } \mathsf{f} \in \Sigma \textit{ with } \mathrm{ar}(\mathsf{f}) = n \tag{Constr}$$
$$\Gamma \vdash M \implies \mathrm{names}(M) \subseteq \mathrm{names}(\Gamma) \tag{Locality}$$
$$\Gamma\theta \vdash M\theta \iff \Gamma \vdash M \quad \textit{for any } \theta \textit{ renaming of } \mathrm{names}(\Gamma) \tag{Alpha}$$
$$\Gamma, a \vdash M \wedge a \notin \mathrm{names}(\Gamma, M) \implies \Gamma \vdash M \tag{Relevancy}$$

The knowledge ordering *for $\mathbb{I}$ is the relation $\leq_{\mathsf{kn}} \subseteq \mathbb{K}^\Sigma \times \mathbb{K}^\Sigma$ such that $\Gamma_1 \leq_{\mathsf{kn}} \Gamma_2$ if and only if $\forall M \in \mathbb{M}^\Sigma \colon \Gamma_1 \vdash M \implies \Gamma_2 \vdash M$. We write $\Gamma_1 \sim_{\mathsf{kn}} \Gamma_2$ if $\Gamma_1 \leq_{\mathsf{kn}} \Gamma_2$ and $\Gamma_2 \leq_{\mathsf{kn}} \Gamma_1$.*

Motivation for the intruder axioms The first three axioms deal exclusively with what it means to be a deduction relation: what is known can be derived (Id); the more is known the more can be derived (Mon); what can be derived is known (Cut)—interpreted as a deduction rule this is the CUT rule. The (Constr) axiom ensures the intruder is able to construct arbitrary messages by composing known messages. The (Locality) axiom forbids derivability to create new arbitrary names. The creation of names is controlled via name restrictions in our calculus, not by derivability. The (Alpha) axiom reflects the immateriality of the choice of names; it will justify α-renaming in our calculus. The (Relevancy) axiom will allow us to only consider boundedly many nonces maliciously injected by the intruder, at each step of the protocol.

In the rest of the paper, unless otherwise specified, we fix an arbitrary effective intruder model $\mathbb{I}$ and omit the corresponding superscripts.

▶ **Proposition 2.** *Given $\Gamma_1, \Gamma_2 \in \mathbb{K}^\Sigma$, $\Gamma_1 \leq_{\mathsf{kn}} \Gamma_2$ if and only if $\forall M \in \Gamma_1 \colon \Gamma_2 \vdash M$. As a consequence, if $\vdash$ is decidable, so is $\leq_{\mathsf{kn}}$.*

$$\frac{M \in \Gamma}{\Gamma \vdash M}\ \text{Id} \qquad \frac{\Gamma, (M,N), M, N \vdash M'}{\Gamma, (M,N) \vdash M'}\ \text{P}_L \qquad \frac{\Gamma \vdash M \qquad \Gamma \vdash N}{\Gamma \vdash (M,N)}\ \text{P}_R$$

$$\frac{\Gamma, \mathsf{e}(M)_K \vdash K \qquad \Gamma, \mathsf{e}(M)_K, M, K \vdash N}{\Gamma, \mathsf{e}(M)_K \vdash N}\ \text{E}_L \qquad \frac{\Gamma \vdash M \qquad \Gamma \vdash K}{\Gamma \vdash \mathsf{e}(M)_K}\ \text{E}_R$$

Figure 1 Deduction rules for the derivability relation of $\mathbb{I}_{\mathsf{sy}}$

Our framework uses the derivability relation as a black box and does not rely on the way it is specified (e.g. with a rewriting system or a deduction system). In our examples we find it convenient to specify it with a sequent calculus in the style of [16].

▶ **Definition 3** (Symmetric encryption). *A model of symmetric encryption can be given using the signature* $\Sigma_{\mathsf{sy}} = \{(\cdot,\cdot), \mathsf{e}(\cdot).\}$*, where* (M,N) *pairs messages* M *and* N*, and* $\mathsf{e}(M)_N$ *represents the message* M *encrypted with key* N*. The intruder model for symmetric encryption is the model* $\mathbb{I}_{\mathsf{sy}} = (\Sigma_{\mathsf{sy}}, \vdash)$ *where* $\vdash$ *is defined by the deduction rules in Figure 1.*

▶ **Proposition 4.** *The model* Σ_{sy} *is an effective intruder model.*

2.1 A calculus for cryptographic protocols

A common approach to model cryptographic primitives is to consider both constructors (e.g. encryption) and destructors (e.g. decryption). Here messages only contain constructors, and "destruction" is represented by pattern matching. Fix a finite signature $\mathcal{Q}$ of process names (ranged over by Q) each of which has a fixed arity $\mathrm{ar}(\mathsf{Q}) \in \mathbb{N}$. A protocol specification consists of an initial process P and a finite set Δ of definitions of the form $\mathsf{Q}[x_1, \ldots, x_n] := A$, with $\mathrm{ar}(\mathsf{Q}) = n$, where the syntax of P and A follows the grammar:

$$P ::= \mathbf{0} \mid \nu x.P \mid P \parallel P \mid \langle M \rangle \mid \mathsf{Q}[\vec{M}] \qquad A ::= \mathbf{in}(\vec{x} : M).P \mid A + A \quad \text{(actions)}$$

We use the vector notation $\vec{x} = x_1, \ldots, x_n$ for lists of pairwise distinct names. In an action $\mathbf{in}(\vec{x} : M).P$, we call $\vec{x} : M$ the *pattern*, and P the *continuation*; processes $\mathsf{Q}[\vec{M}]$ are called *process calls*. If $\Gamma = \{M_1, \ldots, M_k\}$ then $\langle \Gamma \rangle := \langle M_1 \rangle \parallel \ldots \parallel \langle M_k \rangle$. We define $P^0 := \mathbf{0}$ and $P^{n+1} := P \parallel P^n$. The internal action $\boldsymbol{\tau}$, is an abbreviation for $\mathbf{in}(x : x)$, for a fresh x. Processes of the form $\langle M \rangle$ or $\mathsf{Q}[\vec{a}]$ are called *sequential*. The names $\vec{x}$ are bound in both $\nu\vec{x}.P$ and $\mathbf{in}(\vec{x} : M).P$. We denote the set of free names of a term P with $\mathrm{fn}(P)$ and the set of bound names with $\mathrm{bn}(P)$. As is standard, we require, wlog, that $\mathrm{fn}(P) \cap \mathrm{bn}(P) = \emptyset$. When nesting restrictions $\nu\vec{x}.\nu\vec{y}.P$, we implicitly assume wlog that $\vec{x}$ and $\vec{y}$ are disjoint. We assume there is at most one definition for each $\mathsf{Q} \in \mathcal{Q}$, and that for each definition $\mathsf{Q}[x_1, \ldots, x_n] := A$, $\mathrm{fn}(A) \subseteq \{x_1, \ldots, x_n\}$. The set $\mathbb{P}$ consists of all processes over an underlying signature $\mathcal{Q}$.

Structural congruence We write $\overset{\alpha}{=}$ for standard α-equivalence. Structural congruence, $\equiv$, is the smallest congruence relation that includes $\overset{\alpha}{=}$, and is associative and commutative with respect to $\parallel$ and $+$ with $\mathbf{0}$ as the neutral element, and satisfies the standard laws:

$$\nu a.\mathbf{0} \equiv \mathbf{0} \qquad \nu a.\nu b.P \equiv \nu b.\nu a.P \qquad P \parallel \nu a.Q \equiv \nu a.(P \parallel Q) \quad (\text{if } a \notin \mathrm{fn}(P))$$

The second law is called *exchange*, the third *scope extrusion*. Note that the only structural law that modifies the messages is α-equivalence; the axiom (Alpha) ensures that α-equivalence preserves derivability. Every process P is congruent to a process in *standard form*:

$$\nu\vec{x}.\big(\langle M_1 \rangle \parallel \cdots \parallel \langle M_m \rangle \parallel \mathsf{Q}_1[\vec{N}_1] \parallel \cdots \parallel \mathsf{Q}_k[\vec{N}_k]\big) \qquad \text{(SF)}$$

where every name in $\vec{x}$ occurs free in some subterm. We write $\mathrm{sf}(P)$ for the standard form of P, which is unique up to α-equivalence, and associativity and commutativity of parallel. We abbreviate standard forms with $\nu\vec{x}.(\langle\Gamma\rangle \parallel Q)$ where all the active messages are collected in Γ, and Q is a parallel composition of process calls. Let $\mathrm{sf}(P)$ be the expression (SF), we define $\mathrm{msg}(P) = \{M_1, \ldots, M_m\} \cup \bigcup_{i=1}^{n} \vec{N}_i$. Thus $\mathrm{msg}(P)$ is the set of messages appearing in a term. When $m = 0, k = 0, \vec{x} = \emptyset$, the expression (SF) is $\mathbf{0}$.

Reduction semantics One can think of standard forms $\nu\vec{x}.(\langle\Gamma\rangle \parallel Q)$ as runtime configurations of the protocol. They capture, at a specific point in time, the current relevant names (which encode nonces/keys/data), the knowledge of the intruder Γ, and the local state of each participant. A sequential term $\mathsf{Q}[\vec{N}]$ represents a single participant in control state Q with local knowledge of messages $\vec{N}$.

Principals can only communicate through an insecure global channel; we model this by mediating every communication by the intruder. An input action can be fired if the intruder can produce a message that matches the action's pattern. Since the intruder can also replay genuine messages as they are, this interaction models both legitimate and malicious communications.

We write $\mathsf{Q}[\vec{M}] \triangleq A$ if $\mathsf{Q}[\vec{x}] := A' \in \Delta$ and $A \overset{\alpha}{=} A'[\vec{M}/\vec{x}]$, up to commutativity and associativity of $+$. The transition relation $\to_\Delta$ defines the semantics of our calculus:

$$\frac{\mathsf{Q}[\vec{M}] \triangleq \mathbf{in}(\vec{x} : N).P' + A \qquad \Gamma, \vec{z} \vdash N[\vec{M}'/\vec{x}] \qquad \vec{z} \text{ fresh}}{\nu\vec{a}.(\langle\Gamma\rangle \parallel \mathsf{Q}[\vec{M}] \parallel C) \to_\Delta \nu\vec{a}.\nu\vec{z}.(\langle\Gamma\rangle \parallel \langle\vec{z}\rangle \parallel P'[\vec{M}'/\vec{x}] \parallel C)} \qquad \frac{P \equiv P' \to_\Delta Q' \equiv Q}{P \to_\Delta Q}$$

Because of (Relevancy), in the first rule we can assume wlog that $\vec{z}$ includes only names that appear in $\vec{M}'$. The set $\mathrm{traces}_\Delta(P) := \{Q_0 \cdots Q_n \mid P \equiv_{\mathsf{kn}} Q_0 \to_\Delta \cdots \to_\Delta Q_n\}$ collects all the transition sequences from P under Δ. The set $\mathrm{reach}_\Delta(P) := \{Q \mid P \to_\Delta^* Q\}$ is the set of processes reachable from P. We often omit Δ when it is clear from the context.

▶ Remark 5 (Implementable Patterns). Our calculus represents decryption with pattern matching. However, general pattern matching is too powerful: a pattern like $\mathbf{in}(x, k : \mathsf{e}(x)_k)$ would obtain *both* the key k and the plaintext x from an encrypted message! This is only a modelling problem: one should make sure all patterns can be implemented using the cryptographic primitives. Consider a pattern $\vec{x} : M$ and let $Z = \mathrm{names}(M) \setminus \vec{x}$; the pattern is *implementable*, if, for all substitutions $\theta \colon Z \to \mathbb{M}$, we have $M\theta, Z\theta \vdash y$ for all $y \in \vec{x}$.

Knowledge congruence We introduce a coarser relation than structural congruence that we call *knowledge congruence* $P \equiv_{\mathsf{kn}} Q$, which is the smallest congruence that includes $\equiv$ and such that $\langle\Gamma_1\rangle \equiv_{\mathsf{kn}} \langle\Gamma_2\rangle$ if $\Gamma_1 \sim_{\mathsf{kn}} \Gamma_2$. Knowledge congruence can also be characterised by

$$P_1 \equiv_{\mathsf{kn}} P_2 \iff \mathrm{sf}(P_1) \overset{\alpha}{=} \nu\vec{x}.(\langle\Gamma_1\rangle \parallel Q) \wedge \mathrm{sf}(P_2) \overset{\alpha}{=} \nu\vec{x}.(\langle\Gamma_2\rangle \parallel Q) \wedge \Gamma_1 \sim_{\mathsf{kn}} \Gamma_2.$$

Intuitively, modulo derivability, two processes $P \equiv_{\mathsf{kn}} Q$ are indistinguishable to the intruder and to the principals. Formally, if $P \equiv_{\mathsf{kn}} Q$ then the transitions systems $(P, \to_\Delta)$ and $(Q, \to_\Delta)$ are isomorphic. We thus close the reduction semantics under knowledge congruence: we add the rule that if $P \equiv_{\mathsf{kn}} P' \to_\Delta Q' \equiv_{\mathsf{kn}} Q$ then $P \to_\Delta Q$.

While knowledge congruence captures when two configurations are essentially the same, knowledge embedding formalises the notion of "sub-configuration".

▶ **Definition 6** (Knowledge embedding). *The* knowledge embedding *relation* $P_1 \sqsubseteq_{\mathsf{kn}} P_2$ *holds if* $P_1 \equiv \nu\vec{x}.(\langle\Gamma_1\rangle \parallel Q)$, $P_2 \equiv \nu\vec{x}.\nu\vec{y}.(\langle\Gamma_2\rangle \parallel Q \parallel Q')$ *and* $\Gamma_1 \leq_{\mathsf{kn}} \Gamma_2$.

▶ **Proposition 7.** *$P_1 \equiv_{\mathsf{kn}} P_2$ if and only if $P_1 \sqsubseteq_{\mathsf{kn}} P_2$ and $P_2 \sqsubseteq_{\mathsf{kn}} P_1$.*

$$
\begin{aligned}
\mathsf{S}[a,b,k_{as},k_{bs}] &:= \mathbf{in}(n_a : (n_a,b)).\nu k.\big(\langle \mathsf{e}(k)_{k_{bs}}\rangle \parallel \big\langle \mathsf{e}(k)_{(n_a,k_{as})}\big\rangle \parallel \mathsf{S}[a,b,k_{as},k_{bs}]\big)\\
\mathsf{A}_1[a,b,k_{as}] &:= \boldsymbol{\tau}.\nu n_a.(\langle (n_a,b)\rangle \parallel \mathsf{A}_2[a,b,k_{as},n_a])\\
\mathsf{A}_2[a,b,k_{as},n_a] &:= \mathbf{in}\big(k : \mathsf{e}(k)_{(n_a,k_{as})}\big).\mathsf{A}_3[a,b,k_{as},k]\\
\mathsf{A}_3[a,b,k_{as},k] &:= \mathbf{in}(n_b : \mathsf{e}(n_b)_k).\langle \mathsf{e}(n_b)_{(k,k)}\rangle\\
\mathsf{B}_1[a,b,k_{bs}] &:= \mathbf{in}(k : \mathsf{e}(k)_{k_{bs}}).\nu n_b.\big(\langle \mathsf{e}(n_b)_k\rangle \parallel \mathsf{B}_2[a,b,k_{bs},n_b,k]\big)\\
\mathsf{B}_2[a,b,k_{bs},n_b,k] &:= \mathbf{in}(\mathsf{e}(n_b)_{(k,k)}).\mathsf{Secret}[k]
\end{aligned}
$$

Figure 2 Formal model of Example 9.

▶ **Theorem 8.** *Knowledge embedding is a simulation, that is, for all P, P' and Q, if $P \to Q$ and $P \sqsubseteq_{\mathsf{kn}} P'$ then there is a Q' such that $P' \to Q'$ and $Q \sqsubseteq_{\mathsf{kn}} Q'$.*

▶ **Example 9.** Consider the following toy protocol, given in Alice&Bob notation, meant to establish a new session key K between A and B through a trusted server S:

$$
\begin{array}{llll}
(1) & A \to S: & N_A, B & \\
(2) & S \to B: & \mathsf{e}(K)_{(N_A,K_{AS})}, \mathsf{e}(K)_{K_{BS}} & \\
(3) & B \to A: & \mathsf{e}(K)_{(N_A,K_{AS})}, \mathsf{e}(N_B)_K & \\
(4) & A \to B: & \mathsf{e}(N_B)_{(K,K)} &
\end{array}
$$

Figure 2 shows the protocol formalised in our calculus. Assume the initial state is $P_0 = \nu a,b,k_{as},k_{bs}.(\mathsf{S}[a,b,k_{as},k_{bs}] \parallel \mathsf{A}_1[a,b,k_{as}] \parallel \mathsf{B}_1[a,b,k_{bs}] \parallel \langle a,b\rangle)$. Step (1) is initiated by A_1 which sends some new name n_a to the server; since communication is over an insecure channel, the message is just output without indicating the intended recipient. The server receives the message (or any message the intruder may decide to forge instead) and outputs the fresh key k encrypted with k_{bs} (the long-term key between B and S) and with the pair (n_a, k_{as}) (note the use of non-atomic encryption keys). In the protocol, these two messages are sent to B but we model step (2) by B_1 which just receives the message relevant to B. The forwarding of $\mathsf{e}(k)_{(n_a,k_{as})}$ from S to A is performed by the intruder instead of B in the model.

In the last two steps, modelled by $\mathsf{B}_2, \mathsf{A}_3$, B sends a nonce n_b encrypted with k, to challenge A to prove she knows k, which she does by sending back $\mathsf{e}(n_b)_{(k,k)}$. At this point, B is convinced that by encrypting messages with k they will be only accessible to A. We model this by making B_2 transition to $\mathsf{Secret}[k]$ after a successful challenge. We always assume the definition $\mathsf{Secret}[k] := \mathbf{in}(k).\mathsf{Leak}[k]$. A transition to $\mathsf{Leak}[k]$ is only possible when the intruder can derive k so we can check whether the secrecy assertion holds by checking that in no reachable process contains a call to $\mathsf{Leak}[k]$.

2.2 Depth-Bounded Protocols

We generalise the definition of *depth-bounded protocols*, first introduced in [11]. The theory of invariants we will study is sound and complete for this class of protocols.

▶ **Definition 10** (Depth). *The* nesting of restrictions *of a term is given by the function* $\mathrm{nest}_\nu(\mathsf{Q}[\vec{a}]) := \mathrm{nest}_\nu(\langle M\rangle) := \mathrm{nest}_\nu(\mathbf{0}) := 0$, $\mathrm{nest}_\nu(\nu x.P) := 1 + \mathrm{nest}_\nu(P)$, $\mathrm{nest}_\nu(P \parallel Q) := \max(\mathrm{nest}_\nu(P), \mathrm{nest}_\nu(Q))$. *The* depth *of a term is defined as the minimal nesting of restrictions in its knowledge congruence class,* $\mathrm{depth}(P) := \min\{\mathrm{nest}_\nu(Q) \mid Q \equiv_{\mathsf{kn}} P\}$.

▶ **Lemma 11.** *Every Q is α-equivalent to a process Q' such that $|\mathrm{bn}(Q')| \le \mathrm{nest}_\nu(Q)$.*

Consider for example $P = \nu a,b,c.(\langle a\rangle \parallel \langle \mathsf{e}(b)_a\rangle \parallel \langle \mathsf{e}(c)_b\rangle \parallel \langle c\rangle)$ which has $\mathrm{nest}_\nu(P) = 3$. The process P is knowledge-congruent to $Q = (\nu a.\langle a\rangle \parallel \nu b.\langle b\rangle \parallel \nu c.\langle c\rangle)$ which has $\mathrm{nest}_\nu(Q) = 1$;

this gives us $\mathrm{depth}(P) = \mathrm{nest}_{\nu}(Q) = 1$. Although $\mathrm{bn}(Q) = \{a, b, c\}$, by α-renaming all names to x we obtain $Q' = (\nu x.\langle x\rangle \parallel \nu x.\langle x\rangle \parallel \nu x.\langle x\rangle)$ which has the property $|\mathrm{bn}(Q')| \leq \mathrm{nest}_{\nu}(Q) \leq \mathrm{depth}(P)$. More generally, Lemma 11 says that processes of depth k can always be represented by only using at most k unique names, by reusing names in disjoint scopes.

Let $\mathbb{S}_s := \{P \in \mathbb{P} \mid \forall M \in \mathrm{msg}(P)\colon \mathrm{size}(M) \leq s\}$ be the set of processes containing messages of size at most s. Let $\mathbb{D}^X_{s,k} := \{P \in \mathbb{S}_s \mid \mathrm{fn}(P) \subseteq X, \exists Q \in \mathbb{S}_s\colon Q \equiv_{\mathsf{kn}} P \wedge \mathrm{nest}_{\nu}(Q) \leq k\}$ be the set of processes of depth at most $k \in \mathbb{N}$, with free names in X, and messages not exceeding size s. When starting from some initial process P_0, every reachable process P has $\mathrm{fn}(P) \subseteq \mathrm{fn}(P_0)$ so X can always be fixed to be $\mathrm{fn}(P_0)$. We therefore omit X from the superscripts to unclutter notation. The set of processes reachable from P while respecting a size bound s is the set $\mathrm{reach}^s_{\Delta}(P) := \{Q \mid P \cdots Q \in \mathrm{traces}_{\Delta}(P) \cap \mathbb{S}^*_s\}$.

▶ **Definition 12.** *For some $s, k \in \mathbb{N}$, we say the process P is (s,k)-*bounded *(w.r.t. a finite set Δ of definitions) if $\mathrm{reach}^s_{\Delta}(P) \subseteq \mathbb{D}_{s,k}$, i.e. from P only processes of depth at most k can be reached, in traces respecting the size bound s.*

Note that the two bounds s and k are very different in nature. Bounding the size of the messages restricts the traces that we are going to consider: if some behaviour can only be exhibited by using messages exceeding size s, then that behaviour is ignored. When considering the rest of the behaviour, we then check if all the reachable processes have depth at most k. If so, the initial process is (s,k)-bounded. This definition has two important implications. First, we are going to only be sound with respect to bugs that can be triggered without using messages that exceed s in size. Second, the property of being (s,k)-bounded is not static. Proving that P *is* (s,k)-bounded, requires characterising the whole set of reachable processes. Section 3.1 explains how our theory of invariants can prove boundedness.

The protocol in Example 9 is $(3,7)$-bounded. We defer the proof of this fact to Section 3.6.

▶ **Example 13** (Encryption Oracle)**.** The definition $\mathsf{E}[k] := \mathbf{in}(x : x).(\langle \mathsf{e}(x)_k\rangle \parallel E[k])$ leads to unboundedness as soon as the initial process contains $\mathsf{E}[k]$ for some k not known to the intruder, and size bound such that x can match messages of size greater than 1. In such case, the intruder can inject messages (c_i, c_{i+1}) for unboundedly many i, where c_i are intruder-generated nonces. Since k is secret, the resulting reachable configurations would contain "encryption chains" of the form $\nu k.\nu c_1, \ldots, c_n.(\langle \mathsf{e}(c_1, c_2)_k\rangle \parallel \langle \mathsf{e}(c_2, c_3)_k\rangle \parallel \ldots \langle \mathsf{e}(c_{n-1}, c_n)_k\rangle)$. When such chains appear in a set for unboundedly many $n \in \mathbb{N}$, the set is not depth-bounded. This *encryption oracle* pattern could be considered an anti-pattern because it can be exploited for a chosen-plaintext attack on the key k. The pattern can be usually modified or constrained to obtain a bounded protocol. One option is to limit the verification to only consider traces where x is of size 1. Another is to modify the input action to $\mathbf{in}(x : \mathsf{e}(x)_e)$ for some key e shared between an honest principal and the oracle. In Section 5, we elaborate on extensions of boundedness that can handle the oracle pattern without these patches.

3 Ideal Completions for Security Protocols

Our main technical contributions are the proofs needed to show that (s,k)-bounded protocols form a completion-post-effective class of WSTS, in the sense of [5]. First we outline the significance and applications of this result, and then proceed with the proofs.

3.1 Downward-closed Invariants and Security Properties

Suppose we want to establish that a protocol P fulfills some security requirement. In a typical proof, one needs to establish many intermediate facts about executions of the protocol.

For example, part of the argument may hinge on some key k being always unknown to the intruder. This kind of property is an *invariant* of the protocol: it holds at every step of an execution. Formally, an invariant of P (under definitions Δ and size constraint s) is any set of processes that includes $\text{reach}^s_\Delta(P)$. For example, k is never leaked to the intruder in executions of the protocol P if the set of processes $\mathcal{S}_k := \{Q \mid \langle k \rangle \not\sqsubseteq_{\mathsf{kn}} Q\}$ —i.e. all the processes where k is not public— is an invariant of P. We will focus here on the class of $\sqsubseteq_{\mathsf{kn}}$-downward-closed invariants. Formally, given a set of processes X, its $\sqsubseteq_{\mathsf{kn}}$-*downward closure* is the set $X{\downarrow} := \{Q \mid \exists P \in X \colon Q \sqsubseteq_{\mathsf{kn}} P\}$. A set X is $\sqsubseteq_{\mathsf{kn}}$-*downward closed* if $X = X{\downarrow}$. Many properties of interest are naturally downward closed. For example, the set $\mathcal{S}_k$ above is downward-closed as $\langle k \rangle \not\sqsubseteq_{\mathsf{kn}} Q$ and $Q' \sqsubseteq_{\mathsf{kn}} Q$ implies $\langle k \rangle \not\sqsubseteq_{\mathsf{kn}} Q'$. Intuitively, if k is secret in Q and $Q' \sqsubseteq_{\mathsf{kn}} Q$ then k is secret in Q'.

The problem we need to solve is, then, how to show that a given downward-closed set X is an invariant for a given protocol. Formally, that corresponds to checking $X \supseteq \text{reach}^s_\Delta(P)$ which, by downward-closure of X, is equivalent to checking $X \supseteq \text{reach}^s_\Delta(P){\downarrow}$. To prove the latter inclusion, our strategy is to find an inductive invariant that includes the initial state P and that is included in X. Let $\text{post}^s_\Delta(X) := \{Q' \mid \exists Q \in X, Q \to Q' \in \mathbb{S}_s\}$ be the set of processes reachable in one step from processes in X. An invariant X is *inductive* if $X \supseteq \text{post}^s_\Delta(X)$, which is equivalent to requiring $X \supseteq \text{post}^s_\Delta(X){\downarrow}$ if X is downward-closed. Any inductive invariant that contains the initial process P will include $\text{reach}^s_\Delta(P)$.
To turn this proof strategy into an algorithm, we need three ingredients:

1. a (recursively enumerable) finite representation of downward-closed sets,
2. a way to decide inclusion between two downward-closed sets, given their representation,
3. an algorithm (called $\widehat{\text{post}}^s_\Delta$) to compute, given the finite representation of a downward-closed set D, the finite representation of $\text{post}^s_\Delta(D){\downarrow}$.

Unfortunately, downward-closed sets cannot be finitely represented in general, especially if one considers unbounded sessions/nonces. We will show, however, that we can devise solutions to all three items above for downward-closed subsets of $\mathbb{D}_{s,k}$, under a mild restriction on the intruder model.

Solving problems 1 to 3 amounts to proving that $\mathbb{D}_{s,k}$ admits a *post-effective ideal completion* in the sense of [13, 5]. Before moving to the proofs, we give some examples of applications. Suppose we provide solutions for problems 1 to 3 above for $\mathbb{D}_{s,k}$. The following problems are then solvable.

Deciding (s,k)-boundedness Given a protocol with initial process P, we can decide whether P is (s,k)-bounded by enumerating all finite representations of downward-closed sets of $\mathbb{D}_{s,k}$. For each we check if (1) it is inductive, and (2) it contains P. If we find such a set I, then we have proven (s,k)-boundedness of P: $\mathbb{D}_{s,k} \supseteq I \supseteq \text{reach}^s_\Delta(P){\downarrow}$. In parallel to the enumeration, we run a fair exploration of the set $\text{traces}(P) \cap \mathbb{S}^*_s$, terminating with a negative answer if we reach a process with depth greater than k. If P is (s,k)-bounded then in the worst case we will eventually consider the finite representation of $\text{reach}^s_\Delta(P){\downarrow}$, which satisfies (1) and (2). If P is not (s,k)-bounded, the downward-closed set enumeration would not terminate, but the exploration eventually would.

Deciding control-state reachability and secrecy Control-state reachability asks whether there is an execution of the protocol which reaches a process containing a process call $\mathsf{Q}[\ldots]$ for some given Q. Secrecy can be reduced to control-state reachability by introducing a definition $\mathsf{Secret}[m] := \mathbf{in}(m).\mathsf{Leak}[m]$ (for a special process identifier Leak with no definition). In the definition of the protocol one can call $\mathsf{Secret}[m]$ to mark some message m as a secret, and secrecy corresponds to asking control-state (un)reachability for Leak.

Assume P is (s,k)-bounded. To decide control-state reachability for Q from P, we

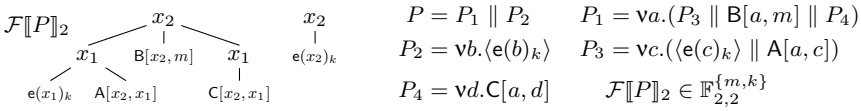

Figure 3 Forest encoding of P. Restrictions at nesting level k are renamed canonically to x_{2-k}.

can replicate the same enumeration scheme as for checking (s, k)-boundedness. For each inductive invariant of P found, we also check whether it contains a call to Q. If it does then we continue with the enumeration, otherwise we declare the protocol correct. For the reachable processes exploration we would similarly terminate, with a negative answer, if we reach a process containing a call to Q. Note that we could also start from an arbitrary P and check (s, k)-boundedness and control-state reachability at the same time.

Susceptibility to known-/chosen-plaintext attacks Assume we find an inductive invariant for P, which does not include the downward-closed set $\left\{ \nu k.\big(\nu m.(\langle m \rangle \parallel \langle \mathsf{e}(m)_k \rangle)\big)^n \mid n \in \mathbb{N} \right\}$. Then we can be sure that the intruder cannot obtain arbitrarily many pairs of plaintext m and its encrypted form $\mathsf{e}(m)_k$ for some fixed k; if that was possible than there would be higher chances for a cryptanalysis attack to succeed in breaking k. Contrary to the previous two algorithms, this is just a semi-decision algorithm, in that if the protocol *is* susceptible to such attack, there is no finite witness of this fact. Interestingly, the property becomes meaningful only when considering unbounded number of nonces.

Note how, in all three cases, if the protocol is found to satisfy the property, the algorithm can output an inductive invariant acting as an independently checkable certificate of correctness and providing the foundations to prove further properties.

The presented algorithms rely on expensive enumeration schemes, which are mainly a theoretical device to prove decidability. We will describe practical optimisations in Section 4.

3.2 Bounded processes are well-quasi-ordered

We will construct finite representations of downward-closed invariants by making use of the algebraic structure of the quasi-order $(\mathbb{D}_{s,k}, \sqsubseteq_{\mathsf{kn}})$. A relation $\sqsubseteq \subseteq S \times S$ over some set S is a *quasi-order* (qo) if it is reflexive and transitive. An infinite sequence $s_0, s_1, \ldots$ of elements of S is called *good* if there are two indexes $i < j$ such that $s_i \sqsubseteq s_j$. The sequence is called *bad* if it is not good. When the qo $(S, \sqsubseteq)$ has no bad sequences it is called a *well quasi order* (wqo). In [11] it is shown that $(\mathbb{D}_{s,k}, \sqsubseteq_{\mathsf{kn}})$ is a wqo, when considering the intruder model $\mathbb{I}_{\mathsf{sy}}$. We refine the result, and generalise it to arbitrary intruder models.

We prove $(\mathbb{D}_{s,k}, \sqsubseteq_{\mathsf{kn}})$ is a wqo for any intruder model, by showing a correspondence between processes in $\mathbb{D}_{s,k}$ and finitely-labelled forests of height at most k. The function $\mathcal{F}[\![\text{-}]\!]_k$, defined in the Appendix, transforms a process P with $\mathrm{nest}_{\nu}(P) \leq k$ into its forest encoding, as illustrated in Figure 3. The forests in $\mathbb{F}_{s,k}^{Y}$ have sequential processes as leaves (with messages have size smaller than s), and free names $Y \uplus \{x_1, \ldots, x_k\}$. Lemma 11 is used to only have to use k distinct bound names: this makes the forests finitely labelled. We represent these bounded-height forests using nested multisets. Multisets over a wqo are wqo. This induces a wqo $(\mathbb{F}_{s,k}^{Y}, \sqsubseteq_{\mathbb{F}})$ which, as we show, implies that $\sqsubseteq_{\mathsf{kn}}$ is a wqo.

▶ **Lemma 14.** *If* $Q \in \mathbb{S}_s^Y$ *and* $\mathrm{nest}_{\nu}(Q) \leq k$*, then* $\mathcal{F}[\![Q]\!]_k \in \mathbb{F}_{s,k}^{Y}$.

▶ **Lemma 15.** *Assume* $Q_1, Q_2 \in \mathbb{S}_s^Y$ *with* $\mathrm{nest}_{\nu}(Q_1) \leq k$ *and* $\mathrm{nest}_{\nu}(Q_2) \leq k$*. Then* $\mathcal{F}[\![Q_1]\!]_k \sqsubseteq_{\mathbb{F}} \mathcal{F}[\![Q_2]\!]_k$ *implies* $Q_1 \sqsubseteq_{\mathsf{kn}} Q_2$.

$$\lceil L\rceil^n := \begin{cases} L & \text{if } L \text{ is sequential or } \mathbf{0} \\ \lceil L_1\rceil^n \parallel \lceil L_2\rceil^n & \text{if } L = L_1 \parallel L_2 \\ \nu x.(\lceil L'\rceil^n) & \text{if } L = \nu x.L' \\ (\lceil B\rceil^n)^n & \text{if } L = B^\omega \end{cases} \qquad L \otimes n := \begin{cases} L & \text{if } L \text{ is sequential or } \mathbf{0} \\ L_1 \otimes n \parallel L_2 \otimes n & \text{if } L = L_1 \parallel L_2 \\ \nu x.(L' \otimes n) & \text{if } L = \nu x.L' \\ (B \otimes n)^n \parallel B^\omega & \text{if } L = B^\omega \end{cases}$$

Figure 4 The grounding $\lceil\text{-}\rceil^n : \mathbb{L}_{s,k} \to \mathbb{D}_{s,k}$ and extension $\text{-}\otimes n : \mathbb{L}_{s,k} \to \mathbb{L}_{s,k}$ operations on limits.

▶ **Theorem 16.** *For every finite $Y \subseteq \mathcal{N}$ and $s, k \in \mathbb{N}$, $(\mathbb{D}^Y_{s,k}, \sqsubseteq_{\mathsf{kn}})$ is a wqo.*

Proof. Let $P_1, P_2, \ldots$ be an infinite sequence of processes in $\mathbb{D}^Y_{s,k}$. By definition, for all $i \in \mathbb{N}$ there exists a process $Q_i \in \mathbb{S}_s$ such that $Q_i \equiv_{\mathsf{kn}} P_i$ and $\text{nest}_\nu(Q_i) \leq k$, which implies by Lemma 14 that $\mathcal{F}[\![Q_i]\!]_k \in \mathbb{F}^Y_{s,k}$. The sequence $\mathcal{F}[\![Q_1]\!]_k, \mathcal{F}[\![Q_2]\!]_k, \ldots$ must then be a good sequence since $(\mathbb{F}^Y_{s,k}, \sqsubseteq_{\mathbb{F}})$ is a wqo. Therefore, there are $i, j \in \mathbb{N}$ with $i < j$, such that $\mathcal{F}[\![Q_i]\!]_k \sqsubseteq_{\mathbb{F}} \mathcal{F}[\![Q_j]\!]_k$, which by Lemma 15 implies $Q_i \sqsubseteq_{\mathsf{kn}} Q_j$. We therefore have $P_i \sqsubseteq_{\mathsf{kn}} P_j$ which proves that the sequence $P_1, P_2, \ldots$ is good. ◀

3.3 Limits and Ideal Decompositions

In this section, we exploit the wqo structure of $\mathbb{D}_{s,k}$ to provide a finite representation for its downward-closed sets. Let $(S, \sqsubseteq)$ be a qo. A set $D \subseteq S$ is an *ideal* if it is downward-closed and directed, i.e. for all $x, y \in D$ there is a $z \in D$ such that $x \sqsubseteq z$ and $y \sqsubseteq z$. We write $\text{Idl}(S)$ for the set of ideals of S. It is well-known that in a well-quasi-order, every downward-closed set is equal to a canonical minimal finite union of ideals, its *ideal decomposition*. To represent downward-closed sets of $\mathbb{D}_{s,k}$ we will only need to provide finite representations of its ideals. We represent ideals using *limits*, which have the same syntax as processes augmented with a construct -^ω to represent processes with arbitrary number of parallel components.

▶ **Definition 17** (Limits). *We call* limits *the terms L formed according to the grammar:*

$$\mathbb{L} \ni L ::= \mathbf{0} \mid (R_1 \parallel \cdots \parallel R_n) \qquad R ::= B \mid B^\omega \qquad B ::= \langle M\rangle \mid \mathsf{Q}[\vec{M}] \mid \nu x.L$$

▶ **Definition 18** (Denotation of limits). *The denotation of L is the set $[\![L]\!] := [\,L\,]{\downarrow}$ where:*

$$[\,\mathbf{0}\,] := \{\mathbf{0}\} \qquad [\,L_1 \parallel L_2\,] := \{(P_1 \parallel P_2) \mid P_1 \in [\,L_1\,], P_2 \in [\,L_2\,]\}$$
$$[\,\langle M\rangle\,] := \{\langle M\rangle\} \qquad [\,\mathsf{Q}[\vec{M}]\,] := \{\mathsf{Q}[\vec{M}]\}$$
$$[\,\nu x.L\,] := \{\nu x.P \mid P \in [\,L\,]\} \qquad [\,B^\omega\,] := \textstyle\bigcup_{n\in\mathbb{N}} \{(P_1 \parallel \cdots \parallel P_n) \mid \forall i \leq n : P_i \in [\,B\,]\}$$

We call the processes in $[\![L]\!]$ instances *of L. Define $\text{nest}_\nu(L)$ to be defined as nest_ν on processes with the addition of the case $\text{nest}_\nu(L^\omega) := \text{nest}_\nu(L)$. It is easy to check that for each $P \in [\![L]\!]$, $\text{depth}(P) \leq \text{nest}_\nu(L)$. We write $\mathbb{L}^X_{s,k}$ for the set of limit expressions L with free names in X that have $\text{nest}_\nu(L) \leq k$ and do not contain messages of size exceeding s. We often omit X and understand it is a fixed finite set of names.*

We now prove that each limit is the denotation of an ideal, and every ideal is a denotation of some limit, i.e. $\text{Idl}(\mathbb{D}_{s,k}) = [\![\mathbb{L}_{s,k}]\!]$. This makes finite unions of limits a sound and complete representation of downward-closed sets of $\mathbb{D}_{s,k}$.

To better manipulate limits, we introduce the n-th grounding $\lceil L\rceil^n$ of a limit L, defined in Figure 4. Grounding can be thought as replacing each -^ω with -^n; e.g. $\lceil(\nu x.\langle x\rangle \parallel (\nu y.\mathsf{Q}[x,y])^\omega)^\omega\rceil^2 = ((\nu x.\langle x\rangle \parallel (\nu y.\mathsf{Q}[x,y]) \parallel (\nu y.\mathsf{Q}[x,y])) \parallel (\nu x.\langle x\rangle \parallel (\nu y.\mathsf{Q}[x,y]) \parallel (\nu y.\mathsf{Q}[x,y])))$.

▶ **Theorem 19.** *For every $L \in \mathbb{L}_{s,k}$, the set $[\![L]\!]$ is an ideal of $(\mathbb{D}_{s,k}, \sqsubseteq_{\mathsf{kn}})$.*

▶ **Theorem 20.** *Every ideal in* $\mathrm{Idl}(\mathbb{D}_k \cap \mathbb{S}_s)$ *is the denotation of some limit* $L \in \mathbb{L}_{s,k}$.

The idea of the proof is to use the forest encoding $\mathcal{F}[\![\text{-}]\!]$ to apply known facts about the ideal completion for multisets to our setting. We take an ideal of $\mathbb{D}_{s,k}$ and show it corresponds to a downward closed set of $\mathbb{F}_{s,k}$, via forest encoding. For multisets, we know exactly how to represent ideals [13], i.e. $\circledast$-products. These products are expressions that can be translated to limits.

3.4 Decidability of Inclusion

We have established that we can finitely represent downward-closed sets of $\mathbb{D}_{s,k}$ using limits. Now we turn to decidability of inclusion between downward-closed sets. We will make use of the following well-known property of ideals: for any D, D_1, D_2 ideals of a wqo, $D \subseteq D_1 \cup D_2 \iff D \subseteq D_1$ or $D \subseteq D_2$. Assume we are given the ideal decomposition of two downward-closed sets $D_1 = I_1 \cup \ldots \cup I_n$ and $D_2 = J_1 \cup \ldots \cup J_m$. We have $D_1 \subseteq D_2$ if and only if for all $1 \leq i \leq n$, there is a $1 \leq j \leq m$, such that $I_i \subseteq J_j$. As a consequence, it suffices to show that inclusion of ideals is decidable.

We extend structural congruence to limits in the obvious way, with laws for parallel and restriction analogous to the process case, and the addition of the law $\langle M \rangle^\omega \equiv \langle M \rangle$. It is easy to check that $L \equiv L'$ implies $[\![L]\!] = [\![L']\!]$. We can define a standard form for limits: every limit is structurally congruent to a limit of the form $\nu\vec{x}.(\langle\Gamma\rangle \parallel \prod_{i\in I} \mathsf{Q}_i[\vec{M}_i] \parallel \prod_{j\in J} B_j^\omega)$ where every name in $\vec{x}$ occurs free at least once in the scope of the restriction, and for all $j \in J$, B_j is also in standard form. When we write $\mathrm{sf}(L) \overset{\alpha}{=} \nu\vec{x}.(\langle\Gamma\rangle \parallel Q \parallel R)$ we imply that Q is a parallel composition of process calls $\prod_{i\in I} \mathsf{Q}_i[\vec{M}_i]$ (in which case we write $|Q|$ for $|I|$) and R is a parallel composition of iterated limits $\prod_{j\in J} B_j^\omega$.

The absorption axiom The decidability proof hinges on a characterisation of inclusion that requires an additional hypothesis on the intruder model.

▶ **Definition 21** (Absorbing intruder). *Fix an intruder model* $\mathbb{I} = (\Sigma, \vdash)$. *Let* $\vec{x}$ *and* $\vec{y}$ *be two lists of pairwise distinct names,* Γ *be a finite set of messages, and* $\Gamma' = \Gamma[\vec{y}/\vec{x}]$. *Moreover, assume that* $\mathrm{names}(\Gamma) \cap \vec{y} = \emptyset$. *We say* $\mathbb{I}$ *is* absorbing *if, for all messages* M *with* $\mathrm{names}(M) \subseteq \mathrm{names}(\Gamma)$, *we have that* $\Gamma, \Gamma' \vdash M$ *if and only if* $\Gamma \vdash M$.

▶ **Lemma 22.** $\mathbb{I}_{\mathsf{sy}}$ *is absorbing.*

For all the results in the rest of the paper, we assume an absorbing intruder model.

The absorption axiom has a technical definition, which becomes more intuitive if understood in the context of limits of the form $L = \big(\nu\vec{x}.(\langle\Gamma\rangle \parallel Q)\big)^\omega$. Imagine comparing the difference in knowledge between $\lceil L \rceil^1$ and $\lceil L \rceil^2$: we have $\mathrm{sf}(\lceil L \rceil^2) \overset{\alpha}{=} \big(\nu\vec{x}.\nu\vec{x}'.(\langle\Gamma\rangle \parallel \langle\Gamma'\rangle \parallel Q \parallel Q')\big)$, where $\Gamma' = \Gamma[\vec{x}'/\vec{x}]$. The absorption axiom tells us that if we want to check whether a process $\nu\vec{x}.\langle M \rangle$ is embedded in $\mathrm{sf}(\lceil L \rceil^2)$, we only need to check if M is derivable from Γ and we can ignore Γ'. In other words, we only need to check if $\nu\vec{x}.\langle M \rangle$ is embedded in $\lceil L \rceil^1$.

To see how this can be used to prove inclusion, consider the simpler problem of deciding if $P \in [\![L]\!]$ given $P \in \mathbb{P}$ and $L \in \mathbb{L}$. We know that $P \in [\![L]\!]$ if and only if $P \sqsubseteq_{\mathsf{kn}} \lceil L \rceil^n$ for some n. Intuitively, there should be some bound $b \in \mathbb{N}$ if P failed to be embedded in $\lceil L \rceil^b$, then there is no chance to embed it in any higher grounding of L. We will prove that this bound is given by the sum of the number of bound names and the number of process calls in P. To prove that grounding $\lceil L \rceil^n$ for $n > b$ is not necessary, we make use of the absorption axiom to show that any process produced by further unfolding a sublimit under ω that was already unfolded b times, is not going to provide new knowledge that can be relevant to P.

$$
\begin{aligned}
L &= \nu a, b, k_{as}, k_{bs}.(\langle a, b\rangle \parallel \mathsf{A}_1[a, b, k_{as}] \parallel \mathsf{B}_1[a, b, k_{bs}] \parallel \mathsf{S}[a, b, k_{as}, k_{bs}]^\omega \parallel L_1^\omega)\\
L_1 &= \nu n_a.\big(\langle n_a\rangle \parallel \mathsf{A}_2[a, b, k_{as}, n_a] \parallel L_2^\omega\big)\\
L_2 &= \nu k.\big(\langle \mathsf{e}(k)_{(a,k_{as})}\rangle \parallel \langle \mathsf{e}(k)_{(b,k_{as})}\rangle \parallel \langle \mathsf{e}(k)_{(n_a,k_{as})}\rangle \parallel \langle \mathsf{e}(k)_{k_{bs}}\rangle \parallel \mathsf{Secret}[k]^\omega \parallel \mathsf{A}_3[a, b, k_{as}, k]^\omega \parallel L_3^\omega\big)\\
L_3 &= \nu n_b.\big(\langle \mathsf{e}(n_b)_{(k,k)}\rangle \parallel \langle \mathsf{e}(n_b)_k\rangle \parallel \mathsf{B}_2[a, b, k_{bs}, n_b, k]\big)
\end{aligned}
$$

Figure 5 An inductive invariant for Example 9.

We introduce a variant of grounding called the n-th extension $L \otimes n$, defined in Figure 4. An extension $L \otimes n$ produces a new limit with each sublimit B^ω unfolded n times, while also keeping the sublimit B^ω. Note that extension does not alter semantics: $[\![L]\!] = [\![L \otimes n]\!]$.

▶ **Theorem 23** (Characterisation of Limits Inclusion)**.** *Let L_1 and L_2 be two limits, with $\mathrm{sf}(L_1) \stackrel{\alpha}{=} \nu\vec{x}_1.(\langle\Gamma_1\rangle \parallel Q_1 \parallel \prod_{i\in I} B_i^\omega)$, and let $n = |\vec{x}_1| + |Q_1| + 1$. Then:*

$$
[\![L_1]\!] \subseteq [\![L_2]\!] \iff \begin{cases} \mathrm{sf}(L_2 \otimes n) \stackrel{\alpha}{=} \nu\vec{x}_1, \vec{x}_2.(\langle\Gamma_2\rangle \parallel Q_1 \parallel Q_2 \parallel R_2) \;\; \textit{and} \;\; \Gamma_1 \leq_{\mathsf{kn}} \Gamma_2 & \textbf{(A)} \\ [\![\langle\Gamma_1\rangle \parallel \prod_{i\in I} B_i]\!] \subseteq [\![\langle\Gamma_2\rangle \parallel R_2]\!] & \textbf{(B)} \end{cases}
$$

▶ **Theorem 24.** *Given $L_1, L_2 \in \mathbb{L}$ it is decidable whether $[\![L_1]\!] \subseteq [\![L_2]\!]$.*

Proof. Theorem 23 leads to a recursive algorithm. Given L_1 and L_2, one computes $\mathrm{sf}(L_1)$ and $\mathrm{sf}(L_2 \otimes n)$. For every α-renaming that makes condition **(A)** hold, one checks condition **(B)** (recursively). If no renaming makes both true then the inclusion does not hold. In the recursive case, there are fewer occurrences of ω in the limit on the left, eventually leading to the case where L_1 has no occurrence of ω, and only condition **(A)** needs to be checked. ◀

3.5 Computing Post-Hat

The last result we need is the decidability of a function $\widehat{\mathrm{post}}^s_\Delta(L)$ which, given a limit L, returns a finite set of limits $\{L_1, \ldots, L_n\}$ such that $\mathrm{post}^s_\Delta([\![L]\!])\!\downarrow = [\![L_1]\!] \cup \ldots \cup [\![L_n]\!]$. The challenge is representing all the possible successors of processes in $[\![L]\!]$ without having to enumerate $[\![L]\!]$. The key idea hinges again on the absorption axiom: we observe that to consider all possible process calls that may cause a transition, it is enough to unfold each ω in L by some bounded number b. Any transition taken from further unfoldings will give rise to successors that are congruent to some of the ones already considered.

The bound b used in extending a limit, is defined as a function of the process definitions and the intruder model. The arity of a pattern $\vec{x} : M$ is $|\vec{x}|$. Given a set of definitions Δ, $\beta(\Delta)$ is the maximum arity of patterns in Δ. The function $\gamma(\mathbb{I})$ returns the maximum arity of the constructors in the signature of $\mathbb{I}$.

▶ **Definition 25** ($\widehat{\mathrm{post}}$)**.** *Let $b = \beta(\Delta) \cdot \gamma(\mathbb{I})^{s-1} + 1$ and $\mathrm{sf}(L \otimes b) = \nu\vec{x}.(\langle\Gamma\rangle \parallel Q \parallel R)$, then*

$$
\widehat{\mathrm{post}}^s_\Delta(L) := \big\{\; \nu\vec{y}.\big(\langle\Gamma'\rangle \parallel Q' \parallel R\big) \;\big|\; \nu\vec{x}.\big(\langle\Gamma\rangle \parallel Q\big) \to_\Delta \nu\vec{y}.\big(\langle\Gamma'\rangle \parallel Q'\big) \in \mathbb{S}_s \;\big\}
$$

▶ **Theorem 26.** $\widehat{\mathrm{post}}^s_\Delta(L) = \{L_1, \ldots, L_n\} \implies \mathrm{post}^s_\Delta([\![L]\!])\!\downarrow = [\![L_1]\!] \cup \ldots \cup [\![L_n]\!]$.

3.6 Example

The limit L in Figure 5 represents an inductive invariant for Example 9, under size bound 3. It can be proven invariant by checking that $\widehat{\mathrm{post}}^3_\Delta(L)$ is included in L. The invariant certificates that any process in $[\![L]\!]$ is $(3, 7)$-bounded (note $\mathrm{nest}_\nu(L) = 7$) and satisfies secrecy. This applies to the initial process of Example 9 as $P_0 \in [\![L]\!]$. In fact, L^ω is also inductive, proving

Table 1 Experimental results. Columns: **Infer**ence of invariant fully automatic (F) or interactive (I); **C**heck of inductiveness; **S**ecrecy proved (✓), not holding (×), not modelled (○).

Name	Infer	C	S	Name	Infer	C	S	Name	Infer	C	S
Ex.9	2.6s F	1.0s	✓	ARPC	0.4s F	0.1s	✓	NHS	8.5s F	1.7s	○
OR	4.0s F	2.1s	○	YAH	8.6s F	2.8s	○	NHSL	2.4s I	16.1s	✓
ORl	37.5s F	3.9s	×	YAHsI	13.4s F	2.8s	✓	KSL	45.0s F	11.2s	✓
ORs	15.0s F	2.3s	✓	YAHs2	9.7s F	2.2s	✓	KSLR	175.1s F	36.2s	✓

boundedness and secrecy for any process in $(P_0)^\omega$. Since $[\![\nu k.(\nu x.(\langle x, \mathsf{e}(x)_k\rangle))^\omega]\!] \not\subseteq [\![L^\omega]\!]$, L^ω provides proof that the protocol is not susceptible to known-/chosen-plaintext attacks. The invariant L was automatically inferred from P_0 using our prototype tool.

4 Algorithmic Aspects and Evaluation

In Section 3 we layed down the foundations for a theory of invariants for depth-bounded protocols. The algorithms presented in Section 3.1 have, however, very high complexity; in this section we summarise strategies that can be employed to obtain more efficient algorithms.

The algorithms for inclusion and $\widehat{\text{post}}$ can be used to check inductiveness given a candidate invariant. This leaves open the question of how to efficiently enumerate candidate invariants. An initial answer can be found in [19] which presents a framework for widening operators on ideal completions. It is not difficult to design a widening operator for bounded protocols inspired by the widening for the depth-bounded π-calculus of [19]. The basic observation is that from a sequence of transitions $P_1 \to^* P_2$ with $P_1 \sqsubseteq_{\mathsf{kn}} P_2$, one can deduce that the same sequence can be simulated from P_2 (by Theorem 8) obtaining a P_3 with $P_2 \sqsubseteq_{\mathsf{kn}} P_3$ and so on. One can then analyse the embedding between P_1 and P_2, justified by $P_1 \equiv_{\mathsf{kn}} \nu\vec{x}.(\langle\Gamma_1\rangle \parallel Q)$, $P_2 \equiv_{\mathsf{kn}} \nu\vec{x}.(\langle\Gamma_1\rangle \parallel Q \parallel P)$ extrapolate the difference P and accellerate the sequence of transitions by constructing the limit $\nu\vec{x}.(\langle\Gamma_1\rangle \parallel Q \parallel P^\omega)$. The widening can be extended to limits. The end product is a finite union of limits which is inductive by construction. This procedure is still very computationally intensive: many inclusion checks are needed and the exploration of transitions can be costly. To obtain a more practical algorithm, we devised two techniques: inductiveness checks through "incorporation", and a coarser widening.

Consider the inductiveness check $[\![\widehat{\text{post}}^s_\Delta(L)]\!] \subseteq [\![L]\!]$ implemented by checking that, for each transition considered by $\widehat{\text{post}}$ the resulting limit L' is included by L. We observe that L' and L will share the context of the transition: L can be rewitten to $C[\mathsf{Q}[\vec{M}]]$ and L' to $C[P]$ for some P. To prove inclusion of $C[P]$ in $C[\mathsf{Q}[\vec{M}]]$ it is then sufficient to show that P is embedded in $C[\mathsf{Q}[\vec{M}]]$. We call this check an *incorporation* of P in C. See the Appendix for an example. In many cases, this check is sufficient to prove inductiveness.

Incorporation lowers the complexity of inductiveness dramatically. There are cases, however, where the incorporation check fails on inductive invariants. Consider for example an inductive invariant represented by the union of two incomparable limits L_1, L_2. Suppose that for some $P \in [\![L_1]\!]$ there is P' with $P \to P' \in [\![L_2]\!] \setminus [\![L_1]\!]$. Then, incorporation of P' in L_1 would fail. To side-step this problem we replace union of limits with parallel composition: $[\![L_1]\!] \cup [\![L_2]\!] \subseteq [\![L_1 \parallel L_2]\!]$ by downward closure of $[\![\text{-}]\!]$. Using this over-approximation, we can try to aim for an inductive invariant which consists of exactly one limit, and for which the incorporation check suffices. The over-approximation is incomplete: there are protocols for which there are only inductive invariants consisting of unions of incomparable limits.

Evaluation We built a proof-of-concept prototype tool implementing limit inclusion, inductivity check, incorporation and a coarse widening. Currently, the tool only supports

symmetric encryption, but we plan to add support for asymmetric encryption, signatures and hashing. The sourcecode and all the test protocol models are available at `https://bitbucket.org/bordaigorl/lemma9`. We summarise our experiments in Table 1. The times are taken by running the tool with Python 2.7, z3-solver v4.8, 8GB RAM, Intel CPU i5, on Linux. The analysed models are variants of well-known protocols: Needham-Schröder(-Lowe) (NHS, NHSL), Yahalom (YAH), Andrew's RPC (ARPC), Otway-Rees (OR), Kehne-Schönwälder-Landendörfer (KSL, KSLR with reauthentification). With the exception of NHSL, all the invariants were obtained fully automatically. For NHSL we had to interactively adjust the widened invariant. To simulate using invariants as correctness certificates, we re-checked them for inductiveness.

5 Conclusions and Future work

This paper presented a principled first step in understanding automatic verification of cryptographic protocols without bounding the number of sessions/nonces.

Our approach is parametric in the intruder model. Although we illustrated the theory using symmetric encryption in our examples, Definition 1 can capture models of many other cryptographic primitives such as asymmetric encryption, signatures, hashing. Handling XOR would require a relaxation of the axioms that we are currently exploring. The current limitation of our framework with respect to more powerful primitives, such as the notoriously difficult to handle Diffie–Hellman exponentiation, is given by the (s, k)-bounded restriction: protocols like Diffie–Hellman Key Exchange, as modelled in our calculus, is not depth-bounded. At the core of the issue is the encryption oracle pattern presented in Example 13. We are currently investigating extensions of the class of (s, k)-bounded protocols that accommodate safe uses of encryption oracles. The main idea is that, while oracles can be used by the intruder to create unbounded chains, these chains may never be inspected by principals, and thus are effectively irrelevant. To weed out irrelevant chains we can make the intruder *lazy*: instead of generating full messages matching a pattern in a transition, the intruder syntesizes a *symbolic* message which is specified only up to what is needed to make it match the pattern. Pattern variables will be bound to symbolic messages that can later be instantiated only if any principal will be able to pattern match on them. Developing a theory of symbolic messages for depth-bounded invariants is a topic of future research.

Another direction for further improvement is extending the class of supported properties. In particular, we plan to study how invariants can be used to automatically prove diff-equivalence [3] without bounding sessions/nonces.

On the practical side, we implemented an inductiveness checker in a proof-of-concept tool. The current version can be used interactively to construct candidate invariants, also with the help of a coarse widening. Our plan is to improve the widening by implementing a refinement loop that iteratively refines an initial coarse invariant candidate, until one is found that proves the desired properties.

Finally, we intend to explore ways in which our invariants can be integrated in existing tools such as ProVerif and Tamarin. For instance, Tamarin performs a backwards search to find possible attacks. Our invariants could provide a pruning technique to avoid exploring paths that are unreachable from the initial state. Similar combinations of forward and backward search has been shown to improve performance dramatically for verification of infinite-state systems such as Petri nets [4].

References

1 Alessandro Armando, David A. Basin, Yohan Boichut, Yannick Chevalier, Luca Compagna, Jorge Cuéllar, Paul Hankes Drielsma, Pierre-Cyrille Héam, Olga Kouchnarenko, Jacopo Mantovani, Sebastian Mödersheim, David von Oheimb, Michaël Rusinowitch, Judson Santiago, Mathieu Turuani, Luca Viganò, and Laurent Vigneron. The AVISPA tool for the automated validation of internet security protocols and applications. In *CAV*, volume 3576 of *Lecture Notes in Computer Science*, pages 281–285. Springer, 2005.

2 Bruno Blanchet. Modeling and verifying security protocols with the applied pi calculus and proverif. *Foundations and Trends in Privacy and Security*, 1(1-2):1–135, 2016.

3 Bruno Blanchet, Martín Abadi, and Cédric Fournet. Automated verification of selected equivalences for security protocols. *J. Log. Algebr. Program.*, 75(1):3–51, 2008. doi:10.1016/j.jlap.2007.06.002.

4 Michael Blondin, Alain Finkel, Christoph Haase, and Serge Haddad. Approaching the coverability problem continuously. In *TACAS*, volume 9636 of *Lecture Notes in Computer Science*, pages 480–496. Springer, 2016.

5 Michael Blondin, Alain Finkel, and Pierre McKenzie. Handling infinitely branching well-structured transition systems. *Inf. Comput.*, 258:28–49, 2018.

6 Rohit Chadha, Vincent Cheval, Ştefan Ciobâcă, and Steve Kremer. Automated verification of equivalence properties of cryptographic protocols. *ACM Trans. Comput. Log.*, 17(4):23:1–23:32, 2016.

7 Vincent Cheval, Steve Kremer, and Itsaka Rakotonirina. DEEPSEC: deciding equivalence properties in security protocols theory and practice. In *IEEE Symposium on Security and Privacy*, pages 529–546. IEEE Computer Society, 2018.

8 Rémy Chrétien, Véronique Cortier, and Stéphanie Delaune. Decidability of trace equivalence for protocols with nonces. In *CSF'15*, pages 170–184. IEEE Computer Society, 2015.

9 Véronique Cortier, Niklas Grimm, Joseph Lallemand, and Matteo Maffei. A type system for privacy properties. In *ACM Conference on Computer and Communications Security*, pages 409–423. ACM, 2017.

10 Morten Dahl, Naoki Kobayashi, Yunde Sun, and Hans Hüttel. Type-based automated verification of authenticity in asymmetric cryptographic protocols. In *ATVA*, volume 6996 of *Lecture Notes in Computer Science*, pages 75–89. Springer, 2011.

11 Emanuele D'Osualdo, Luke Ong, and Alwen Tiu. Deciding secrecy of security protocols for an unbounded number of sessions: The case of depth-bounded processes. In *CSF*, pages 464–480. IEEE Computer Society, 2017.

12 Santiago Escobar, Catherine A. Meadows, and José Meseguer. A rewriting-based inference system for the NRL protocol analyzer and its meta-logical properties. *Theor. Comput. Sci.*, 367(1-2):162–202, 2006.

13 Alain Finkel and Jean Goubault-Larrecq. Forward analysis for WSTS, part I: completions. In *STACS*, volume 3 of *LIPIcs*, pages 433–444, 2009.

14 Simon Meier, Benedikt Schmidt, Cas Cremers, and David A. Basin. The TAMARIN prover for the symbolic analysis of security protocols. In *CAV*, volume 8044 of *Lecture Notes in Computer Science*, pages 696–701. Springer, 2013.

15 Lawrence C. Paulson. The inductive approach to verifying cryptographic protocols. *Journal of Computer Security*, 6(1-2):85–128, 1998.

16 Alwen Tiu, Rajeev Goré, and Jeremy E. Dawson. A proof theoretic analysis of intruder theories. *Logical Methods in Computer Science*, 6(3), 2010.

17 Alwen Tiu, Nam Nguyen, and Ross Horne. SPEC: an equivalence checker for security protocols. In *APLAS*, volume 10017 of *Lecture Notes in Computer Science*, pages 87–95, 2016.

18 Thomas Wies, Damien Zufferey, and Thomas A. Henzinger. Forward analysis of depth-bounded processes. In *FoSSaCS*, volume 6014 of *Lecture Notes in Computer Science*, pages 94–108. Springer, 2010.

19 Damien Zufferey, Thomas Wies, and Thomas A. Henzinger. Ideal abstractions for well-structured transition systems. In *VMCAI'12'*, volume 7148 of *LNCS*, pages 445–460. Springer, 2012.

Appendix

A Proof of Proposition 4

Rule ID implies (Id); it is easy to check that any derivation for $\Gamma \vdash M$ is also a derivation (of same depth) for $\Gamma, \Gamma' \vdash M$, which implies (Mon); (Cut) is a consequence of the admissibility of the CUT rule proven in [16]. Rules P_R and E_R directly entail (Constr). All rules are invariant under renamings so (Alpha) holds. (Relevancy) can be proven by a straightforward induction on the depth of derivations. Since the premises in every rule involve only sub-terms of the consequence, derivability is decidable.

B Proof of Proposition 7

The "only if" direction is trivial. The "if" direction is proved as follows. Let $\mathrm{sf}(P_i) \stackrel{\alpha}{=} \nu\vec{x}_i.(\langle\Gamma_i\rangle \parallel Q_i)$ for $i = 1, 2$. It is easy to check that $P_1 \sqsubseteq_{\mathsf{kn}} P_2$ if and only if $\Gamma_1 \leq_{\mathsf{kn}} \Gamma_2$, $\vec{x}_2 = \vec{x}_1\vec{y}_2$ for some $\vec{y}_2$, and $Q_2 = Q_1 \parallel Q_2'$ for some parallel composition of process calls Q_2'. Similarly, $P_2 \sqsubseteq_{\mathsf{kn}} P_1$ if and only if $\Gamma_2 \leq_{\mathsf{kn}} \Gamma_1$, $\vec{x}_1 = \vec{x}_2\vec{y}_1$, and $Q_1 = Q_2 \parallel Q_1'$ for some $\vec{y}_1, Q_1'$. We get $\Gamma_1 \sim_{\mathsf{kn}} \Gamma_2$, $x_1 = x_2$ and $Q_1 = Q_2$ as required to prove $P_1 \equiv_{\mathsf{kn}} P_2$.

C Auxiliary results

C.1 Properties of Knowledge

▶ **Lemma 27.** *Let $\Gamma_1, \Gamma_2, \Gamma$ be sets of messages such that $\Gamma_1 \leq_{\mathsf{kn}} \Gamma_2$. Then, $\Gamma, \Gamma_1 \leq_{\mathsf{kn}} \Gamma, \Gamma_2$.*

Proof. Let $\Gamma_1 = \{M_1, \ldots, M_n\}$. We have to show that for all N, if $\Gamma, \Gamma_1 \vdash N$ then $\Gamma_2 \vdash N$. We apply (Cut) n times obtaining

$$\dfrac{\Gamma, \Gamma_2 \vdash M_1 \qquad \dfrac{\Gamma, \Gamma_2, M_1 \vdash M_2 \qquad \dfrac{\iddots \quad \dfrac{\Gamma, \Gamma_2, (\Gamma_1 \setminus M_n) \vdash M_n \qquad \Gamma, \Gamma_1, \Gamma_2 \vdash N}{\vdots}\,\textsc{Cut}}{\Gamma, \Gamma_2, M_1, M_2 \vdash N}\,\textsc{Cut}}{\Gamma, \Gamma_2, M_1 \vdash N}\,\textsc{Cut}}{\Gamma, \Gamma_2 \vdash N}\,\textsc{Cut}$$

From $\Gamma_1 \leq_{\mathsf{kn}} \Gamma_2$ we have $\forall i \leq n\colon \Gamma_2 \vdash M_i$, which implies, by (Mon), all the left-most premises. By (Mon), from $\Gamma, \Gamma_1 \vdash N$ we know $\Gamma, \Gamma_1, \Gamma_2 \vdash N$, which completes the derivation showing $\Gamma, \Gamma_2 \vdash N$. ◀

▶ **Corollary 28.** *Let $\Delta_1, \Delta_2, \Gamma_1, \Gamma_2$ be sets of messages s.t. $\Delta_1 \leq_{\mathsf{kn}} \Delta_2$ and $\Gamma_1 \leq_{\mathsf{kn}} \Gamma_2$. Then, $\Delta_1, \Gamma_1 \leq_{\mathsf{kn}} \Delta_2, \Gamma_2$.*

Proof. Easy corollary of Lemma 27: $\Gamma_1, \Delta_1 \leq_{\mathsf{kn}} \Gamma_2, \Delta_1 \leq_{\mathsf{kn}} \Gamma_2, \Delta_2$. ◀

▶ **Lemma 29.** *Let Γ be a set of messages. Then $\langle\Gamma\rangle \parallel \langle\Gamma\rangle \equiv_{\mathsf{kn}} \langle\Gamma\rangle$.*

C.2 Properties of $\sqsubseteq_{\mathsf{kn}}$

▶ **Corollary 30.** *Let Γ be a set of messages, $P_1, P_2 \in \mathbb{P}$ two processes for which $\mathrm{sf}(P_i) = \nu\vec{x}_i.(\langle\Gamma_i\rangle \parallel Q_i)$ for $i \in \{1,2\}$ and $P_1 \sqsubseteq_{\mathsf{kn}} P_2$. Then, $\langle\Gamma\rangle \parallel P_1 \sqsubseteq_{\mathsf{kn}} \langle\Gamma\rangle \parallel P_2$.*

▶ **Lemma 31.** *Let P_1, P_2 be two processes and $n \in \mathbb{N}$. If $P_1 \sqsubseteq_{\mathsf{kn}} P_2$, then $P_1^n \sqsubseteq_{\mathsf{kn}} P_2^n$.*

▶ **Lemma 32.** *Let $L' \in \mathbb{L}$ a limit s.t. $L' = L^\omega$ for some $L \in \mathbb{L}$. Then, $(\lceil L' \rceil^n)^m \sqsubseteq_{\mathsf{kn}} \lceil L' \rceil^{m*n}$ holds for every $m, n \in \mathbb{N}$,*

C.3 Properties of grounding

▶ **Lemma 33.** *For every $n \in \mathbb{N}$, $\lceil L \rceil^n \in [\, L \,]$.*

▶ **Lemma 34.** *For every $n \leq m \in \mathbb{N}$, $\lceil L \rceil^n \sqsubseteq_{\mathsf{kn}} \lceil L \rceil^m$.*

▶ **Lemma 35.** *For every $P \in [\![L]\!]$ there exists an $n \in \mathbb{N}$ such that $P \sqsubseteq_{\mathsf{kn}} \lceil L \rceil^n$.*

▶ **Lemma 36.** *Let $L_1, L_2 \in \mathbb{L}$, then $[\![L_1]\!] \subseteq [\![L_2]\!] \iff \forall n \in \mathbb{N}\colon \exists m \in \mathbb{N}\colon \lceil L_1 \rceil^n \sqsubseteq_{\mathsf{kn}} \lceil L_2 \rceil^m$.*

C.4 Properties of extension

▶ **Lemma 37.** *For every limit L and $n \in \mathbb{N}$, $[\![L]\!] = [\![L \otimes n]\!]$.*

▶ **Lemma 38.** *For every limit L and $n \in \mathbb{N}$, $\lceil L \rceil^n = \lceil L \otimes n \rceil^0$.*

D Soundness and Completeness of Limits

Multisets We say a function $\mu\colon X \to \mathbb{N}$ is finite-support if its support $\mathrm{supp}(\mu) := \{x \mid \mu(x) > 0\}$, is finite. Given a set X, the multisets over X, $\mu \in \mathcal{M}(X)$, are the finite-support functions from X to $\mathbb{N}$. Assume $(X, \sqsubseteq)$ is a quasi order; its *multiset extension* is the quasi order $\sqsubseteq_{\mathcal{M}}$ over $\mathcal{M}(X)$ defined as follows; for two multisets $\mu_1, \mu_2 \in \mathcal{M}(X)$, $\mu_1 \sqsubseteq_{\mathcal{M}} \mu_2$ holds just if there is an injective function $f\colon \mathrm{supp}(\mu_1) \to \mathrm{supp}(\mu_2)$ such that for each $x \in \mathrm{supp}(\mu_1)$, we have $x \sqsubseteq f(x)$ and $\mu_1(x) \leq \mu_2(f(x))$. When X is a finite set ordered by equality, $\sqsubseteq_{\mathcal{M}}$ coincides with multiset inclusion. If $(X, \sqsubseteq)$ is a wqo then $(\mathcal{M}(X), \sqsubseteq_{\mathcal{M}})$ is a wqo [13].

Forests We define a domain of forests $\mathbb{F}^X_{s,k}$ with sequential processes in $\mathcal{B}_s(X)$ as leaves:

$$\mathcal{B}_s(X) := \{\mathsf{Q}[\vec{M}] \mid Q \in \mathcal{Q}, \mathrm{ar}(\mathsf{Q}) = |\vec{M}|, \vec{M} \subseteq \mathbb{M}^X_s\} \cup \mathbb{M}^X_s$$
$$\mathbb{F}^X_{s,0} := \mathcal{M}(\mathcal{B}_s(X)) \qquad \mathbb{F}^X_{s,k+1} := \mathcal{M}\Big(\mathcal{B}_s(X) \uplus \mathbb{F}^{X \cup \{x_{k+1}\}}_{s,k}\Big) \qquad \text{(assuming } x_{k+1} \notin X\text{)}$$

The set $\mathcal{B}_s(X)$ contains all sequential processes of $\mathbb{D}^X_{s,k}$. In the base case, forests of height 0, $\mathbb{F}^X_{s,0}$ is simply a multiset of sequential processes. Forests of height $k+1$ are represented in $\mathbb{F}^X_{s,k+1}$ by a multiset of sequential processes and of subforests of height k.

For any given $s \in \mathbb{N}$ and finite $X \subseteq \mathcal{N}$, $\mathcal{B}_s(X)$ is a finite set and thus forms a wqo with equality. With $\sqsubseteq_{\mathbb{F}}$ (forest embedding) we denote the qo over $\mathbb{F}^X_{s,k}$ defined as follows: on $\mathbb{F}^X_{s,0}$ it coincides with the multiset extension of equality on $\mathcal{B}_s(X)$; on $\mathbb{F}^X_{s,k+1}$ it coincides with the multiset extension of the disjoint union of equality on $\mathcal{B}_s(X)$, and the forest embedding on $\mathbb{F}^{X \cup \{x_{k+1}\}}_{s,k}$. By iterating the result on multisets, we obtain that for any $s, k \in \mathbb{N}$ and finite $X \subseteq \mathcal{N}$, $(\mathbb{F}^X_{s,k}, \sqsubseteq_{\mathbb{F}})$ is a wqo.

▶ **Definition 39** (Forest encoding). *The function $\mathcal{F}[\![\text{-}]\!]_k$ transforms a process P with $\mathrm{nest}_{\mathsf{v}}(P) \leq k$ into a forest:*

$$\mathcal{F}[\![P]\!]_k := \begin{cases} \emptyset & \text{if } P = \mathbf{0} \\ \{\!|P|\!\} & \text{if } P \text{ is sequential} \\ \{\!|\mathcal{F}[\![Q[x_k/x]]\!]_{k-1}|\!\} & \text{if } P = \mathsf{v}x.Q \\ \mathcal{F}[\![Q_1]\!]_k \oplus \mathcal{F}[\![Q_2]\!]_k & \text{if } P = Q_1 \parallel Q_2 \end{cases}$$

assuming $\{x_1, \ldots, x_k\} \cap (\mathrm{bn}(P) \cup \mathrm{fn}(P)) = \emptyset$; when this assumption is not met because of bound names, we implicitly α-rename the term before applying the definition. When $\mathrm{nest}_{\mathsf{v}}(P) > k$, $\mathcal{F}[\![P]\!]_k$ is undefined.

D.1 Proof of Lemma 15

First we strengthen the statement: we can prove that if $\mathcal{F}[\![Q_1]\!]_k \sqsubseteq_{\mathbb{F}} \mathcal{F}[\![Q_2]\!]_k$ then $\mathrm{sf}(Q_1) = \mathsf{v}\vec{y}.R$ and $\mathrm{sf}(Q_2) = \mathsf{v}\vec{y}.\mathsf{v}\vec{z}.(R \parallel R')$, which clearly implies $Q_1 \sqsubseteq_{\mathsf{kn}} Q_2$.

We proceed by induction on k. The base case is a special case of the induction step, so let us consider the latter first. Assume $\mathrm{nest}_{\mathsf{v}}(Q_1), \mathrm{nest}_{\mathsf{v}}(Q_2) \leq k+1$ and $\varphi_1 = \mathcal{F}[\![Q_1]\!]_{k+1} \sqsubseteq_{\mathbb{F}} \mathcal{F}[\![Q_2]\!]_{k+1} = \varphi_2$, and let $B = \mathrm{supp}(\varphi_1) \cap \mathcal{B}_Y(s)$ and $C = \mathrm{supp}(\varphi_1) \setminus B$ (note that $C \subseteq \mathbb{F}_{s,k}^{Y \cup \{x_{k+1}\}}$). Then, by definition, there is an injective function $f\colon \mathrm{supp}(\varphi_1) \to \mathrm{supp}(\varphi_2)$ such that for each $P \in B$, $P = f(P)$ and $\varphi_1(P) \leq \varphi_2(P)$; moreover, for each $\varphi \in C$, $\varphi \sqsubseteq_{\mathbb{F}} f(\varphi)$ and $\varphi_1(\varphi) \leq \varphi_2(\varphi)$. By definition of $\mathcal{F}[\![\text{-}]\!]$ we know that each $\varphi \in C$ is the forest encoding $\mathcal{F}[\![P_1]\!]_k$ for some subterm (α-equivalent to) $\mathsf{v}x_{k+1}.P_\varphi$ of Q_1 with $\mathrm{nest}_{\mathsf{v}}(P_\varphi) \leq \mathrm{nest}_{\mathsf{v}}(Q_1) - 1 \leq k$. Similarly for Q_2 we have $f(\varphi) = \mathcal{F}[\![P_{f(\varphi)}]\!]_k$ for some subterm $\mathsf{v}x_{k+1}.P_{f(\varphi)}$ with $\mathrm{nest}_{\mathsf{v}}(P_{f(\varphi)}) \leq \mathrm{nest}_{\mathsf{v}}(Q_2) - 1 \leq k$. We can therefore apply the induction hypothesis and get $\varphi \sqsubseteq_{\mathbb{F}} f(\varphi)$ implies that $\mathrm{sf}(P_\varphi) = \mathsf{v}\vec{y}_\varphi.R_\varphi$ and $\mathrm{sf}(P_{f(\varphi)}) = \mathsf{v}\vec{y}_\varphi.\mathsf{v}\vec{z}_\varphi.(R_\varphi \parallel R'_\varphi)$. As a consequence of this and of $\varphi_1(\varphi) \leq \varphi_2(\varphi)$, we have

$$Q_1 \equiv_{\mathsf{kn}} \left(\textstyle\prod_{P \in B} P^{\varphi_1(P)} \parallel \prod_{\varphi \in C} (\mathsf{v}x_{k+1}.\mathsf{v}\vec{y}_\varphi.R_\varphi)^{\varphi_1(\varphi)}\right)$$
$$Q_2 \equiv_{\mathsf{kn}} \left(\textstyle\prod_{P \in B} P^{\varphi_1(P)} \parallel \prod_{\varphi \in C} (\mathsf{v}x_{k+1}.\mathsf{v}\vec{y}_\varphi.\mathsf{v}\vec{z}_\varphi.(R_\varphi \parallel R'_\varphi))^{\varphi_1(\varphi)} \parallel R'\right)$$

which clearly entails the claim, by application of α-renaming and scope extrusion to get the two standard forms. In the base case, $C = \emptyset$ from which the claim follows straightforwardly.

D.2 Proof of Theorem 19

We want to prove that for every $L \in \mathbb{L}_{s,k}$, we have $[\![L]\!] \subseteq \mathbb{D}_{s,k}$ and $[\![L]\!]$ is $\sqsubseteq_{\mathsf{kn}}$-downward closed and directed.

Clearly $[\![L]\!] \subseteq \mathbb{D}_{s,k}$ since every $P \in [\, L \,]$ has $\mathrm{nest}_{\mathsf{v}}(P) \leq \mathrm{nest}_{\mathsf{v}}(L) \leq k$ and its active messages are the messages of L up to renaming (which does not alter size).

$[\![L]\!]$ is downward closed by definition. Proving that $[\![L]\!]$ is directed requires showing that for each $P_1, P_2 \in [\![L]\!]$, there is $Q \in [\![L]\!]$ such that $P_1 \sqsubseteq_{\mathsf{kn}} Q$ and $P_2 \sqsubseteq_{\mathsf{kn}} Q$. Such Q can be constructed by applying Lemma 35 to P_1 and P_2, obtaining n_1 and n_2 such that $P_i \sqsubseteq_{\mathsf{kn}} \lceil L \rceil^{n_i}$, for $i = 1, 2$. Then $Q = \lceil L \rceil^{\max(n_1, n_2)}$ is the required process by Lemma 34.

D.3 Proof of Theorem 20

The idea of the proof is to use the forest encoding $\mathcal{F}[\![\text{-}]\!]$ to apply known facts about the ideal completion for multisets to our setting. We take an ideal of $\mathbb{D}_{s,k}$ and show it corresponds to a downward closed set of $\mathbb{F}_{s,k}$, via forest encoding. For multisets we know exactly how

to represent ideals [13], i.e. $\circledast$-products. These products are expressions that can be readily seen as limits. Let us recall the result on multisets first and then proceed with the proof.

Let X be a wqo with a finite representation of ideals $\mathbb{I}(X)$. Then we can finitely represent ideals of $\mathcal{M}(X)$ by expressions of the form

$$p = I_1^{\circledast} \odot \cdots \odot I_n^{\circledast} \odot J_1^{?} \odot \cdots \odot J_m^{?}$$

where $I_i, J_j \in \mathbb{I}(X)$ for all $1 \leq i \leq n$, $1 \leq j \leq m$. When $n, m = 0$ we write $p = \epsilon$. Such expressions are called $\circledast$-products. Their denotations are the ideals of $\mathcal{M}(X)$, via the map

$$\begin{aligned} [\![\epsilon]\!] &:= \{\emptyset\} & [\![p_1 \odot p_2]\!] &:= \{\mu_1 \oplus \mu_2 \mid \mu_1 \in [\![p_1]\!], \mu_2 \in [\![p_2]\!]\} \\ [\![I^{\circledast}]\!] &:= \mathcal{M}([\![I]\!]) & [\![I^{?}]\!] &:= \{ \{\!\{x\}\!\} \mid x \in [\![I]\!] \} \cup \{\emptyset\} \end{aligned}$$

In a wqo, downward closed sets are finite unions of ideals, so any downward closed set of $\mathcal{M}(X)$ can be described as the union of the denotation of finitely many $\circledast$-products. For the details[1] see [13].

Another well-known property of ideals we are going to use is the following. Let D be a downward closed set, then the following are equivalent:

(1) D is an ideal;
(2) for all downward closed sets $D_1, D_2 \subseteq X$, if $D \subseteq D_1 \cup D_2$ then $D \subseteq D_1$ or $D \subseteq D_2$;
(3) for all downward closed sets $D_1, D_2 \subseteq X$, if $D = D_1 \cup D_2$ then $D = D_1$ or $D = D_2$.

Now we turn to our theorem. Let $D \subseteq \mathbb{D}_{s,k}$, we define

$$\mathcal{F}[\![D]\!]_k := \{\mathcal{F}[\![P]\!] \mid P \in D, \mathrm{nest}_{\mathsf{v}}(P) \leq k\}.$$

By definition, $\mathcal{F}[\![D]\!]_k \subseteq \mathbb{F}_{s,k}$.

We also define the opposite mapping, from forests to processes in the expected way. Let $\varphi \in \mathbb{F}^X_{s,k}$, $B(\varphi) := \mathrm{supp}(\varphi) \cap \mathcal{B}_s(X)$ and $C(\varphi) := \mathrm{supp}(\varphi) \setminus B(\varphi)$. Then

$$\mathcal{P}[\![\varphi]\!]_k := \left(\prod_{P \in B(\varphi)} P^{\varphi(P)} \parallel \prod_{\psi \in C(\varphi)} (\mathsf{v} x_k.\mathcal{P}[\![\psi]\!]_{k-1})^{\varphi(\psi)}\right)$$

Since when $k = 0$, $C(\varphi) = \emptyset$, the expression is well-defined. We have $\mathcal{P}[\![\varphi]\!]_k \subseteq \mathbb{D}_{s,k}$ and[2] $\mathcal{F}\left[\!\left[\mathcal{P}[\![\varphi]\!]_k\right]\!\right]_k = \varphi$ and $\mathcal{P}\left[\!\left[\mathcal{F}[\![Q]\!]_k\right]\!\right]_k \equiv Q$ (when defined).

▶ **Lemma 40.** *For any $\sqsubseteq_{\mathsf{kn}}$-downward closed $D \subseteq \mathbb{D}_{s,k}$, $\mathcal{F}[\![D]\!]_k$ is $\sqsubseteq_{\mathbb{F}}$-downward closed.*

Proof. First note that for each $P \in D$ there is a $Q \equiv_{\mathsf{kn}} P$ in D such that $\mathcal{F}[\![Q]\!] \in \mathcal{F}[\![D]\!]_k$.

Towards a contradiction, assume there is a forest $\varphi \in \mathbb{F}_{s,k}$ such that there is a $\psi \in \mathcal{F}[\![D]\!]_k$ with $\varphi \sqsubseteq_{\mathbb{F}} \psi$ but $\varphi \notin \mathcal{F}[\![D]\!]_k$. We know that $\psi = \mathcal{F}[\![P]\!]_k$ for some $P \in D$. From Lemma 15 we can derive that $\mathcal{P}[\![\varphi]\!]_k \sqsubseteq_{\mathsf{kn}} P$. But since D is downward closed, $\mathcal{P}[\![\varphi]\!]_k \in D$, which contradicts the assumption that $\varphi \notin \mathcal{F}[\![D]\!]_k$. ◀

Proof of Theorem 20 We prove, by induction on k, that $\mathrm{Idl}(\mathbb{D}^X_{s,k}) \subseteq [\![\mathbb{L}_{s,k}]\!]$. Again, the base case is a special case of the induction step. Let $k > 0$ and $D \in \mathrm{Idl}(\mathbb{D}^X_{s,k})$; By induction hypothesis, we can assume that the ideals of $\mathbb{D}^X_{s,k-1}$ are described by limits in $\mathbb{L}^X_{s,k-1}$. Since

[1] here we use a variation of $\circledast$-products that can be seen equivalent to the one in [13] by using $(C_1 \cup C_2)^{\circledast} = C_1^{\circledast} \odot C_2^{\circledast}$.
[2] Note the implicit α-renaming in the application of $\mathcal{F}[\![\text{-}]\!]$.

D is downward closed, by Lemma 40, $\mathcal{F}[\![D]\!]_k \subseteq \mathbb{F}_{s,k}$ is downward closed as well, and thus is the finite union of some ideals of $\mathbb{F}^X_{s,k}$. Now we have,

$$\mathrm{Idl}\big(\mathbb{F}^X_{s,k}\big) = \mathrm{Idl}\Big(\mathcal{M}\Big(\mathcal{B}_s(X) \uplus \mathbb{F}^{X\cup\{x_{k+1}\}}_{s,k}\Big)\Big)$$

so we can represent $\mathrm{Idl}(\mathbb{F}^X_{s,k})$ with $\circledast$-products over $\mathcal{B}_s(X) \uplus \mathbb{L}^{X\cup\{x_{k+1}\}}_{s,k}$. Let $p_1, \ldots, p_n$ be such that $\bigcup_{i=1}^n [\![p_i]\!] = \mathcal{F}[\![D]\!]_k$. From each p_i we can obtain a limit L_{p_i} by replacing:

1. $\odot$ with $\|$;
2. each $B^?$ with B and each $B^\circledast$ with B^ω, if $B \in \mathcal{B}_s(X)$;
3. each $L^?$ with $\nu x_k.L$ and each $L^\circledast$ with $(\nu x_k.L)^\omega$, if $L \in \mathbb{L}^{X\cup\{x_{k+1}\}}_{s,k}$.

We now show that $D = \bigcup_{i=1}^n [\![L_{p_i}]\!]$. It is important to remember that $[\![p_i]\!]$ are $\sqsubseteq_{\mathbb{F}}$-ideals while $[\![L_{p_i}]\!]$ are $\sqsubseteq_{\mathsf{kn}}$-ideals. One can show, by induction on the structure of p_i, that $Q \in [\,L_{p_i}\,]$ if and only if $Q \equiv Q'$ for some $Q' \in \mathcal{P}[\![p_i]\!]$. It then follows that $[\![L_{p_i}]\!] = \{Q \mid \exists \varphi \in [\![p_i]\!] : Q \sqsubseteq_{\mathsf{kn}} \mathcal{P}[\![\varphi]\!]\}$ and therefore

$$\begin{aligned} D &= \{Q \mid \exists \varphi \in \mathcal{F}[\![D]\!] : Q \sqsubseteq_{\mathsf{kn}} \mathcal{P}[\![\varphi]\!]\} \\ &= \bigcup_{i=1}^n \{Q \mid \exists \varphi \in [\![p_i]\!] : Q \sqsubseteq_{\mathsf{kn}} \mathcal{P}[\![\varphi]\!]\} = \bigcup_{i=1}^n [\![L_{p_i}]\!] \end{aligned}$$

We thus established that $D = \bigcup_{i=1}^n [\![L_{p_i}]\!]$. Since D is an ideal and each $[\![L_{p_i}]\!]$ is downward closed, we have that $D = [\![L_{p_j}]\!]$ for some $1 \le j \le n$.

E Proof of Lemma 22

Let $\vec{x}$ and $\vec{y}$ be two lists of pairwise distinct names, $\Gamma \subseteq \mathbb{M}^{\Sigma_{\mathrm{sy}}}$ be a finite set of messages, and $\Gamma' = \Gamma[\vec{y}/\vec{x}]$. Moreover, assume that $\mathrm{names}(\Gamma) \cap \vec{y} = \emptyset$. We have to show that, for all messages M with $\mathrm{names}(M) \subseteq \mathrm{names}(\Gamma)$, we have that $\Gamma, \Gamma' \vdash M$ if and only if $\Gamma \vdash M$.

The $\Leftarrow$ direction is a direct consequence of (Mon). For the other direction, let $\sigma = [\vec{y}/\vec{x}]$ and $\sigma' = [\vec{x}/\vec{y}]$, so $N\sigma' \in \Gamma$ if and only if $N \in \Gamma'$. Note that $\Gamma\sigma' = \Gamma'\sigma' = \Gamma$. Moreover, if $\mathrm{names}(M) \subseteq \mathrm{names}(\Gamma)$, $M = M\sigma'$ because, by assumption, $\mathrm{names}(\Gamma) \cap \vec{y} = \emptyset$.

We proceed by induction on the depth of the derivation for $\Gamma, \Gamma' \vdash M$. We assume the statement holds for derivations of depth d and do a case analysis on the last rule applied in the $(d+1)$-deep derivation for $\Gamma, \Gamma' \vdash M$.

Rule Id: $M \in \Gamma \cup \Gamma'$. If $M \in \Gamma$ we are done. Otherwise, if $M \in \Gamma'$, we have $M\sigma' \in \Gamma$ but $M = M\sigma'$ so $M \in \Gamma$.

Rule P_L: $(N_1, N_2) \in \Gamma \cup \Gamma'$ and $\dfrac{\Gamma, \Gamma', N_1, N_2 \vdash M}{\Gamma, \Gamma' \vdash M}$.

We then have two cases:

Case $(N_1, N_2) \in \Gamma$. We can then transform the d-deep derivation for $\Gamma, \Gamma', N_1, N_2 \vdash M$ into a derivation for $\Gamma, N_1, N_2, \Gamma', N_1\sigma, N_2\sigma \vdash M$ of the same depth by (Mon). Since $M \subseteq \mathrm{names}(\Gamma, N_1, N_2) = \mathrm{names}(\Gamma)$ and $\Gamma', N_1\sigma, N_2\sigma = (\Gamma, N_1, N_2)\sigma$, we can apply the induction hypothesis, obtaining $\Gamma, N_1, N_2 \vdash M$ which proves $\Gamma \vdash M$ by application of Rule P_L.

Case $(N_1, N_2) \in \Gamma'$. From $\Gamma, \Gamma', N_1, N_2 \vdash M$ we get $(\Gamma, \Gamma', N_1, N_2)\sigma' \vdash M\sigma'$ by (Alpha). Since we have $\Gamma\sigma' = \Gamma'\sigma' = \Gamma$ and $M\sigma' = M$, we can infer that $\Gamma, N_1\sigma', N_2\sigma' \vdash M$ which proves, together with $(N_1\sigma', N_2\sigma') \in \Gamma$, that $\Gamma \vdash M$ by application of Rule P_L.

Rule E_L: $\mathsf{e}(N)_K \in \Gamma \cup \Gamma'$ and $\dfrac{\Gamma, \Gamma' \vdash K \qquad \Gamma, \Gamma', N, K \vdash M}{\Gamma, \Gamma' \vdash M}$.

We distinguish two cases:

Case $\mathsf{e}(N)_K \in \Gamma$. Then $\mathrm{names}(K) \subseteq \mathrm{names}(\Gamma)$ and by induction hypothesis we have $\Gamma \vdash K$. We also have $\mathrm{names}(N) \subseteq \mathrm{names}(\Gamma)$, $(\Gamma, N, K)\sigma = \Gamma', N\sigma, K\sigma$ and $\mathrm{names}(M) \subseteq \mathrm{names}(\Gamma, N, K)$. We can transform the d-deep derivation for $\Gamma, \Gamma', N, K \vdash M$ into a derivation for $\Gamma, N, K, \Gamma', N\sigma, K\sigma \vdash M$ with the same depth, so by induction hypothesis we obtain $\Gamma, N, K \vdash M$. By applying Rule E_L to $\Gamma \vdash K$ and $\Gamma, N, K \vdash M$ we obtain $\Gamma \vdash M$.

Case $\mathsf{e}(N)_K \in \Gamma'$. Then by (Alpha), $\Gamma, \Gamma' \vdash K$ implies $\Gamma\sigma', \Gamma'\sigma' \vdash K\sigma'$. Since $\Gamma\sigma' = \Gamma$ and $\Gamma'\sigma' = \Gamma$ we have $\Gamma \vdash K\sigma'$. Similarly, $\Gamma, \Gamma', N, K \vdash M$ implies $\Gamma, N\sigma', K\sigma' \vdash M\sigma'$. Since $M\sigma' = M$, we have $\Gamma, N\sigma', K\sigma' \vdash M$, which in conjuction with $\Gamma \vdash K\sigma'$ and $\mathsf{e}(N\sigma')_{K\sigma'} \in \Gamma$, proves $\Gamma \vdash M$ by application of Rule E_L.

Rule P_R: $M = (N_1, N_2)$ and $\dfrac{\Gamma, \Gamma' \vdash N_1 \qquad \Gamma, \Gamma' \vdash N_2}{\Gamma, \Gamma' \vdash M}$.

We have $\mathrm{names}(N_1), \mathrm{names}(N_1) \subseteq \mathrm{names}(\Gamma)$ so we can apply the induction hypothesis to the two premises and get $\Gamma \vdash N_1$ and $\Gamma \vdash N_2$, which proves $\Gamma \vdash M$ by Rule P_R.

Rule E_R: Analogous to the case for Rule P_R.

F Proof of Characterisation of Limits Inclusion

For any limit L with standard form $\nu\vec{x}.(\langle\Gamma\rangle \parallel Q \parallel \prod_{i\in I} B_i^\omega)$, we call Γ and Q non-iterated, also fixed, components while B_i for $i \in I$ are iterated components.

We recall the statement: Let L_1 and L_2 be two limits, with $\mathrm{sf}(L_1) \overset{\alpha}{=} \nu\vec{x}_1.(\langle\Gamma_1\rangle \parallel Q_1 \parallel \prod_{i\in I} B_i^\omega)$, and let $n = |\vec{x}_1| + |Q_1| + 1$. Then:

$$[\![L_1]\!] \subseteq [\![L_2]\!] \iff \begin{cases} \mathrm{sf}(L_2 \otimes n) \overset{\alpha}{=} \nu\vec{x}_1, \vec{x}_2.(\langle\Gamma_2\rangle \parallel Q_1 \parallel Q_2 \parallel R_2) \text{ and } \Gamma_1 \leq_{\mathsf{kn}} \Gamma_2 & \textbf{(A)} \\ [\![\langle\Gamma_1\rangle \parallel \prod_{i\in I} B_i]\!] \subseteq [\![\langle\Gamma_2\rangle \parallel R_2]\!] & \textbf{(B)} \end{cases}$$

F.1 Sufficient Conditions

▶ **Lemma 41** (Sufficient Conditions). *If conditions* **(A)** *and* **(B)** *hold, then* $[\![L_1]\!] \subseteq [\![L_2]\!]$.

Proof. Given that **(A)** and **(B)** hold, it suffices to show that $\forall n_1 \in \mathbb{N}, \exists n_2 \in \mathbb{N}. \lceil L_1 \rceil^{n_1} \sqsubseteq_{\mathsf{kn}} \lceil L_2 \otimes n \rceil^{n_2}$ by Lemmas 36 and 37. Let n_1 be an arbitrary number. With $R_1 := \prod_{i\in I} B_i^\omega$, we have $\lceil \mathrm{sf}(L_1) \rceil^{n_1} = \nu\vec{x}_1.(\langle\Gamma_1\rangle \parallel Q_1 \parallel \lceil R_1 \rceil^{n_1})$ and $\mathrm{sf}(\lceil L_2 \otimes n \rceil^0) \overset{\alpha}{=} \nu\vec{x}_1, \vec{x}_2.(\langle\Gamma_2\rangle \parallel Q_1 \parallel Q_2)$ by **(A)**.

We show that $\nu\vec{x}_1.(\langle\Gamma_1\rangle \parallel Q_1 \parallel \lceil R_1 \rceil^{n_1}) \sqsubseteq_{\mathsf{kn}} \nu\vec{x}_1, \vec{x}_2.(\langle\Gamma_2\rangle \parallel Q_1 \parallel Q_2 \parallel \lceil R_2 \rceil^{n_2})$ for some n_2. Condition **(A)** induces a knowledge order for both fixed parts. Condition **(B)** gives rise to some relation between the iterated components we will exploit. This relation is captured in the following fact. For some α-renaming in **(A)** and some n_2:

$$\langle\Gamma_1\rangle \parallel \lceil R_1 \rceil^{n_1} \sqsubseteq_{\mathsf{kn}} \langle\Gamma_2\rangle \parallel \lceil R_2 \rceil^{n_2} \qquad (*)$$

Prior to proving (*), we show that this fact and **(A)** suffice to prove our goal. With $\mathrm{sf}(\lceil R_i \rceil^{n_i}) = \nu\vec{y}_i.(\Gamma_{R_i^{n_i}} \parallel Q_i')$, we call $\Gamma_{R_i^{n_i}}$ the knowledge of $\lceil R_i \rceil^{n_i}$ for $i \in \{1, 2\}$. Because of $(*)$, we can choose $\vec{y}_2$ so that $\vec{y}_2 = \vec{y}_1, \vec{z}_2$ and there is no need to rename for the embedding anymore. Equipped with this abbreviation, the knowledge of the left hand side $(\nu\vec{x}_1.(\langle\Gamma_1\rangle \parallel Q_1 \parallel \lceil R_1 \rceil^{n_1})$ is $(\Gamma_1, \Gamma_{R_1^{n_1}})$ while $\Gamma_2, \Gamma_{R_2^{n_2}}$ is the knowledge of the right hand

side $\nu\vec{x}_1, \vec{x}_2.(\langle\Gamma_2\rangle \parallel Q_1 \parallel Q_2 \parallel \lceil R_2\rceil^{n_2})$. This is also exactly the knowledge contained in $(\langle\Gamma_1\rangle \parallel \lceil R_1\rceil^{n_1})$ respectively $(\langle\Gamma_2\rangle \parallel \lceil R_2\rceil^{n_2})$, hence $\Gamma_1, \Gamma_{R_1^{n_1}} \leq_{\mathsf{kn}} \Gamma_2, \Gamma_{R_2^{n_2}}$ by (*). It remains to take care of names and process calls for the embedding. Condition **(A)** already takes care of non-iterated names and process calls. With our assumption (*), we obtain

$$\nu\vec{x}_1.(\langle\Gamma_1\rangle \parallel Q_1 \parallel \langle\Gamma_1\rangle \parallel \lceil R_1\rceil^{n_1}) \sqsubseteq_{\mathsf{kn}} \nu\vec{x}_1, \vec{x}_2.(\langle\Gamma_2\rangle \parallel Q_1 \parallel Q_2 \parallel \langle\Gamma_2\rangle \parallel \lceil R_2\rceil^{n_2}).$$

Both parts are knowledge congruent to our goal's sides by Lemma 29 which proves the goal.

It remains to show that (*), i.e. $(\langle\Gamma_1\rangle \parallel \lceil R_1\rceil^{n_1}) \sqsubseteq_{\mathsf{kn}} (\langle\Gamma_2\rangle \parallel \lceil R_2\rceil^{n_2})$ for some n_2. We start with **(B)** and apply Lemma 36 so that we know that for every $n_1 \in \mathbb{N}$, there is a $m \in \mathbb{N}$ such that $\langle\Gamma_1\rangle \parallel \lceil\prod_{i\in I} B_1\rceil^{n_1} \sqsubseteq_{\mathsf{kn}} \langle\Gamma_2\rangle \parallel \lceil R_2\rceil^m$. By Lemma 31, we multiply on both sides:

$$(\langle\Gamma_1\rangle \parallel \lceil\prod_{i\in I} B_1\rceil^{n_1})^{n_1} \sqsubseteq_{\mathsf{kn}} (\langle\Gamma_2\rangle \parallel \lceil R_2\rceil^m)^{n_1}.$$

By Lemma 29, we pull the messages out of both replications:

$$\langle\Gamma_1\rangle \parallel (\lceil\prod_{i\in I} B_1\rceil^{n_1})^{n_1} \sqsubseteq_{\mathsf{kn}} \langle\Gamma_2\rangle \parallel (\lceil R_2\rceil^m)^{n_1}.$$

We continue on both sides individually. On the left, we omit messages for simplicity as we already know that $\Gamma_1 \leq_{\mathsf{kn}} \Gamma_2$.

$$(\lceil\prod_{i\in I} B_1\rceil^{n_1})^{n_1} = (\prod_{i\in I}\lceil B_1\rceil^{n_1})^{n_1} = \lceil R_1\rceil^{n_1}.$$

While we use Lemma 32 and Corollary 30 on the right in order to obtain that:

$$\langle\Gamma_2\rangle \parallel (\lceil R_2\rceil^m)^{n_1} \sqsubseteq_{\mathsf{kn}} \langle\Gamma_2\rangle \parallel \lceil R_2\rceil^{n_1\cdot m}$$

Combining both paths leads to: $\langle\Gamma_1\rangle \parallel \lceil R_1\rceil^{n_1} \sqsubseteq_{\mathsf{kn}} \langle\Gamma_2\rangle \parallel \lceil R_2\rceil^{n_1\cdot m}$. By choosing $n_2 = n_1 \cdot m$, the claim follows. ◀

F.2 Necessary Conditions

▶ **Lemma 42** (Necessary Conditions). *If $[\![L_1]\!] \subseteq [\![L_2]\!]$, then conditions* **(A)** *and* **(B)** *hold.*

For this proof we are assuming the intruder model is absorbing. Before we give the proof, we need some auxiliary definitions and results.

▶ **Corollary 43.** *Let $\vec{x}$ and $\vec{y}$ be two lists of pairwise distinct names, Γ, Γ_1 be two finite sets of messages, and $\Gamma_2 = \Gamma_1[\vec{y}/\vec{x}]$. Moreover, assume that* $\mathrm{names}(\Gamma_1) \cap \vec{y} = \emptyset$ *and* $\mathrm{names}(\Gamma) \cap \vec{y} = \emptyset = \mathrm{names}(\Gamma) \cap \vec{x}$. *Then, for all messages M with* $\mathrm{names}(M) \subseteq \mathrm{names}(\Gamma, \Gamma_1)$, *we have that $\Gamma, \Gamma_1, \Gamma_2 \vdash M$ if and only if $\Gamma, \Gamma_1 \vdash M$.*

Proof. The direction from right to left is obvious. For the reverse direction, it is equivalent to show that $\Gamma, \Gamma_1, \Gamma, \Gamma_2 \vdash M$. Now, $\Gamma, \Gamma_1 = (\Gamma, \Gamma_2)[\vec{y}/\vec{x}]$. The claim follows by the assumption that the intruder is absorbing (Definition 21). ◀

▶ **Lemma 44.** *Let L, L' be two limits such that $L \equiv L'$. Then for every $n \in \mathbb{N} : \lceil L\rceil^n \equiv \lceil L'\rceil^n$.*

Proof. The proof is a straightforward structural induction on L. ◀

We introduce a refinement of grounding and a function folding the right hand side but preserving the knowledge embedding. The (n, k, m)-th grounding takes a limit and unfolds each ω n times for the outer k nested levels of ω, and m times for the inner ones.

▶ **Definition 45** (Step-indexed grounding). *For a limit L in standard form, we define the (n,k,m)-th grounding of L to be the process $\lceil L \rceil^{n,k,m}$ recursively:*

$$\lceil L \rceil^{n,k,m} := \begin{cases} L & \text{if } L \text{ is sequential or } \mathbf{0} \\ \lceil L_1 \rceil^{n,k,m} \parallel \lceil L_2 \rceil^{n,k,m} & \text{if } L = L_1 \parallel L_2 \\ \nu x.(\lceil L' \rceil^{n,k,m}) & \text{if } L = \nu x.L' \\ \left(\lceil B \rceil^{n,k-1,m}\right)^n & \text{if } L = B^\omega \wedge k > 0 \\ (\lceil B \rceil^m)^m & \text{if } L = B^\omega \wedge k = 0 \end{cases}$$

▶ **Definition 46** (ω-height). *For a limit L, we define the ω-height as follows:*

$$\omega\text{-height}(L) := \begin{cases} 0 & \text{if } L \text{ is sequential or } \mathbf{0} \\ \max\left(\omega\text{-height}(R_1), \cdots, \omega\text{-height}(R_n)\right) & \text{if } L = R_1 \parallel \cdots \parallel R_n \\ \omega\text{-height}(L') & \text{if } L = \nu x.L' \\ \omega\text{-height}(B) + 1 & \text{if } L = B^\omega \end{cases}$$

▶ **Lemma 47.** *Let L be a limit and $m, n, k \in \mathbb{N}$. If $k \geq \omega\text{-height}(L)$, then $\lceil L \rceil^{n,k,m} = \lceil L \rceil^n$.*

Proof. The parameter k only decreases when recursing into a limit under ω. If $k \geq \omega$-height(L) the last case of the definition will never apply, which makes the definition coincide with the one of n-grounding. ◀

The idea of the following parametrised function is to fold an m-grounding to an n-grounding up to a certain ω-height k of the limit. Since we want to be very specific about the domains, we define some sets of groundings.

▶ **Definition 48** (Set of groundings). *Let $L \in \mathbb{L}$ and $m, n, k \in \mathbb{N}$.*
We define the following two sets of processes:
- $\mathcal{R}_L^m := \{\lceil \mathrm{sf}(L) \rceil^m\}$
- $\mathcal{R}_L^{n,k,m} := \{\lceil \mathrm{sf}(L) \rceil^{n,k,m}\}$

▶ **Definition 49.** *Let L be a limit in recursive standard form and $m, k, n \in \mathbb{N}$. The function $\Phi_{k,L}^{n,m} : \mathcal{R}_L^m \to \mathcal{R}_L^{n,k,m}$ and is parametrised in all four variables.*

$$\Phi_{k,P}^{n,m}(P) := \begin{cases} P & \text{if } k = 0 \\ \nu \vec{x}.(\langle \Gamma \rangle \parallel Q \parallel \prod_{j \in J} \left(\Phi_{k,L_j}^{n,m}(\lceil L_j^\omega \rceil^m)\right) & \text{if } k > 0 \text{ and } P = \nu \vec{x}.(\langle \Gamma \rangle \parallel Q \parallel \prod_{j \in J} L_j^\omega) \\ \left(\Phi_{k-1,L}^{n,m}(\lceil L \rceil^m)\right)^n & \text{if } k > 0 \text{ and } P = \lceil L^\omega \rceil^m \end{cases}$$

We may omit the parameters n and m in the following if obvious from context as they do not change over the process of folding.

▶ **Lemma 50** (Folding is sound wrt. knowledge embedding). *Let L_1, L_2 be two limits in standard form $L_i = \nu \vec{x_i}.(\langle \Gamma_i \rangle \parallel Q_i \parallel R_i)$ with $n = |\vec{x_1}| + |Q_1| + 1$. If there is a $m \in \mathbb{N}$ such that $m > n$ and $\lceil L_1 \rceil^0 \sqsubseteq_{\mathsf{kn}} \lceil L_2 \rceil^m$, then $\forall k : \lceil L_1 \rceil^0 \sqsubseteq_{\mathsf{kn}} \Phi_{k,L_2}^{n,m}(\lceil L_2 \rceil^m)$.*

Proof. We prove the claim by induction on k.

For k = 0, the claim trivially follows as $\Phi_{0,L_2}^{n,m}(\lceil L_2 \rceil^m) = \lceil L_2 \rceil^m$ by the assumption that $\lceil L_1 \rceil^0 \sqsubseteq_{\mathsf{kn}} \lceil L_2 \rceil^m$ holds.

For the induction step, we assume that $\lceil L_1 \rceil^0 \sqsubseteq_{\mathsf{kn}} \Phi^{n,m}_{k,L_2}(\lceil L_2 \rceil^m)$ and prove that

$\lceil L_1 \rceil^0 \sqsubseteq_{\mathsf{kn}} \Phi^{n,m}_{k+1,L_2}(\lceil L_2 \rceil^m)$.

By definition of folding, both $\Phi_{k,L}(\lceil L_2 \rceil^m)$ and $\Phi_{k+1,L}(\lceil L_2 \rceil^m)$ are folding in exactly the same way up to the k-th recursive calls, i.e. calls in which k decreases. This means that up to the calls to $\Phi_{0,L'}$ and $\Phi_{1,L'}$, both $\Phi_{k,L}(\lceil L_2 \rceil^m)$ and $\Phi_{k+1,L}(\lceil L_2 \rceil^m)$ will have constructed the same context $C[-, \cdots, -]$ around these final calls. We thus characterise $A = \Phi^{n,m}_{k,L_2}(\lceil L_2 \rceil^m)$, and $B = \Phi^{n,m}_{k+1,L_2}(\lceil L_2 \rceil^m)$ as follows:

$$\begin{aligned} A &= C[\Phi^{n,m}_{0,F_1^\omega}(F_1^\omega), \cdots, \Phi^{n,m}_{0,F_j^\omega}(F_j^\omega)] \\ &= C[\lceil F_1^\omega \rceil^m, \cdots, \lceil F_j^\omega \rceil^m] = C[(\lceil F_1 \rceil^m)^m, \cdots, (\lceil F_j \rceil^m)^m] \\ B &= C[\Phi^{n,m}_{1,F_1^\omega}(F_1^\omega), \cdots, \Phi^{n,m}_{1,F_j^\omega}(F_j^\omega)] = C[(\lceil F_1 \rceil^m)^n, \cdots, (\lceil F_j \rceil^m)^n] \end{aligned}$$

Recall that $L_1 = \nu\vec{x}_1.(\langle \Gamma_1 \rangle \parallel Q_1 \parallel R_1)$ and hence $\lceil L_1 \rceil^0 = \nu\vec{x}_1.(\langle \Gamma_1 \rangle \parallel Q_1)$. By assumption, we have

$$\nu\vec{x}_1.(\langle \Gamma_1 \rangle \parallel Q_1) \sqsubseteq_{\mathsf{kn}} \Phi^{n,m}_{k,L_2}(\lceil L_2 \rceil^m) = A \tag{i}$$

and want to prove that

$$\nu\vec{x}_1.(\langle \Gamma_1 \rangle \parallel Q_1) \sqsubseteq_{\mathsf{kn}} \Phi^{n,m}_{k+1,L_2}(\lceil L_2 \rceil^m) = B. \tag{ii}$$

Intuitively, we have to find a way to preserve the knowledge embedding from (i) when removing some branches in the holes of the context to get from A to B. Let us show what A and B look like explicitly with their context:

$$\begin{aligned} A &\equiv \nu\vec{c}.(\langle \Gamma_c \rangle \parallel Q_c \parallel D) && \text{with } D = (\lceil F_1 \rceil^m)^m \parallel \cdots \parallel (\lceil F_j \rceil^m)^m \\ B &\equiv \nu\vec{c}.(\langle \Gamma_c \rangle \parallel Q_c \parallel E) && \text{with } E = (\lceil F_1 \rceil^m)^n \parallel \cdots \parallel (\lceil F_j \rceil^m)^n \end{aligned}$$

We call Γ_c the knowledge of the context.

Our goal is to show that reducing the number of iterations in each of D's parallel components does not affect the knowledge embedding described in (i) and thereby obtain (ii). Let us first consider names and process calls. We can split $\lceil L_1 \rceil^0$ in the following way: $\lceil L_1 \rceil^0 = \nu\vec{x}_1.(\langle \Gamma_1 \rangle \parallel Q_1) = \nu\vec{y}.\nu\vec{z}.(\langle \Gamma_1 \rangle \parallel Q_y \parallel Q_z)$ where $\vec{y} \subseteq \vec{c}$ and $Q_y \subseteq Q_c$. The intention is to distinguish names and process calls that are already matched in the context. This is why we do not require all process calls that are only using names from $\vec{y}$ to be in Q_y but some of them might be in Q_z. The goal follows with the following claim immediately.

Claim I: In every hole of context $C[-, \cdots, -]$, we can reduce the number of branches to at most n, i.e. for every $1 \leq l \leq j$ we can have a grounding $\lceil F_l \rceil^n$ instead of $\lceil F_j \rceil^m$.

Proof of Claim I. Towards a contradiction, assume that there is a hole in which we cannot remove $m - n$ branches. W.l.o.g. let $\lceil L_l \rceil^m$ for $1 \leq l \leq j$ be the sublimit in this hole. There might be three reasons for this: names, process calls and knowledge.

Considering the names and process calls, we know that $|\vec{x_1}| + |Q_1| < n$ by definition and hence $|\vec{z}| + |Q_z| < n$. Now, we investigate the components that $\vec{z}$ and Q_z are matched to. In the worst case, all of them are mapped to this hole but we still delete $m - n$ branches that are not used to match the names $\vec{z}$. For the process calls in Q_z, we have to distinguish two cases. First, if a process call uses any name from $\vec{z}$, it is fine as we will leave them anyway. Second, if a process call does not use any name from $\vec{z}$, it is fine to delete this branch as there will be enough copies in the remaining branches to cover this process call.

It remains to reason about knowledge. We make the knowledge of D, i.e. the one having budget k, explicit: $\mathrm{sf}(D) = \nu\vec{a}.(\Gamma_m \parallel \cdots)$ and factor out the knowledge from sublimit $\lceil F_l \rceil^m$: $\Gamma_m = \Gamma'_m, \Gamma_{l,m}$ so that $\Gamma_{l,m}$ was the knowledge obtained through $\lceil F_l \rceil^m$. For knowledge, we have to prove that $\forall M\colon \Gamma_1 \vdash M \implies \Gamma_c, \Gamma'_m, \Gamma_{l,n} \vdash M$ given that $\Gamma_c, \Gamma'_m, \Gamma_{l,m} \vdash M$. The idea now is to reduce the knowledge from $\Gamma_{l,m}$ to $\Gamma_{l,n}$ by Corollary 43, which is a corollary of the absorbing intruder. Let us define $\Gamma'_{c,m} = \Gamma_c, \Gamma'_m$ indicating the context of the hole we are considering. As m might be bigger than $2n$, we have to iterate the process of reducing the number of branches. Hence, we generalise the notation of $\Gamma_{l,m}$ and $\Gamma_{l,n}$ in the following way: $(\lceil F_l \rceil^m)^i \equiv \nu\vec{a_i}.(\Gamma_{l,i} \parallel \cdots)$.

Claim II: $\forall m > n,\ \Gamma_{l,m} \vdash M \implies \Gamma_{l,m-1} \vdash M$.

Proof of Claim II. For convenience, we rename $\Gamma_{l,i}$ to Λ_i. The main observation is that we can split the knowledge Λ_m into Λ_{m-1} and a remainder Λ'. We can choose a branch which does not use names from $\vec{z}$ to contribute to Λ'. Since we know that $n \geq 1$, we know that $m > 1$ by assumption. Therefore, we can split Λ_{m-1} again and obtain the knowledge stemming from one branch which we call Λ''. Let us recall the assumption and goal after these rewriting steps: Given that

$$\Gamma'_{c,m}, \Lambda_{m-2}, \Lambda', \Lambda'' \vdash M \tag{iii}$$

holds, we want to prove that $\Gamma'_{c,m}, \Lambda_{m-2}, \Lambda'' \vdash M$. Let $\vec{w}'$ and $\vec{w}''$ be the names only used in Λ' and Λ'' respectively so that:

$$\vec{w}' \cap \vec{w}'' = \emptyset \text{ and } \mathrm{names}(\Gamma'_{c,m}, \Lambda_{m-2}) \cap \vec{w}' = \emptyset = \mathrm{names}(\Gamma'_{c,m}, \Lambda_{m-2}) \cap \vec{w}'' \tag{iv}$$

Λ' and Λ'' have been obtained from a branch of the same sublimit, so we can infer that

$$\Lambda'' = \Lambda'[\vec{w}''/\vec{w}']. \tag{v}$$

Notice that $\vec{w}' \cap \vec{z} = \emptyset$ by the fact how we have chosen the branch for Λ'. Furthermore, $\vec{w}' \cap \vec{y} = \emptyset$ by $\vec{y} \subseteq \vec{c}$. Combining these observations, we get that $\vec{x_1} \cap \vec{w}' = \emptyset$. By the Locality-Axiom and $\Gamma_1 \vdash M$, we get $\mathrm{names}(M) \subseteq \mathrm{names}(\Gamma_1) \subseteq \vec{x_1}$. Therefore, $\mathrm{names}(M) \cap \vec{w'} = \emptyset$ which implies that

$$\mathrm{names}(M) \subseteq \mathrm{names}(\Gamma'_{c,m}, \Lambda_{m-2}, \Lambda'') \tag{vi}$$

The facts (iii) to (vi) fulfil the conditions for Corollary 43 resulting in $\Gamma'_{c,m}, \Lambda_{m-2}, \Lambda'' \vdash M$ which reads $\Gamma'_{c,m}, \Lambda_{m-1} \vdash M$ when folding back which is the goal of Claim II. By this, we have shown that we can remove $m - n$ branches which leads to a contradiction which is why Claim I holds. As $\lceil F_l \rceil^m$ was chosen arbitrarily, we have shown that we can remove $m - n$ branches in every hole of the context $C[-, \cdots, -]$ which concludes this proof. ◀

▶ **Corollary 51.** *Let L_1, L_2 be two limits with $\mathrm{sf}(L_1) = \nu\vec{x}_1.(\langle\Gamma_1\rangle \parallel Q_1 \parallel R_1)$ and $[\![L_1]\!] \subseteq [\![L_2]\!]$. Then, $\mathrm{sf}(L_2 \otimes n) \overset{\alpha}{=} \nu\vec{x}_1, \vec{x}_2.(\langle\Gamma_2\rangle \parallel Q_1 \parallel Q_2 \parallel R_2)$ and $\Gamma_1 \leq_{\mathsf{kn}} \Gamma_2$. for $n = |\vec{x}_1| + |Q_1| + 1$.*

Proof. First, we show that $\lceil \mathrm{sf}(L_1) \rceil^0 \sqsubseteq_{\mathsf{kn}} \lceil L_2 \rceil^n$. By $[\![L_1]\!] \subseteq [\![L_2]\!]$, we know that there is an m so that $\lceil \mathrm{sf}(L_1) \rceil^0 \sqsubseteq_{\mathsf{kn}} \lceil L_2 \rceil^m$ by Lemma 36. From Lemma 50, we obtain that that $\lceil \mathrm{sf}(L_1) \rceil^0 \sqsubseteq_{\mathsf{kn}} \Phi^{n,m}_{k,L_2}(\lceil L_2 \rceil^m)$ for every k. Substituting $\omega\text{-height}(L_2)$ for k leads to $\Phi^{n,m}_{\omega\text{-height},L_2}(\lceil L_2 \rceil^m)$. This is $\lceil L_2 \rceil^n$ by Lemma 47. Using this knowledge embedding, we get $\lceil L_2 \rceil^n \overset{\alpha}{=} \nu\vec{x}_1, \vec{x}_2.(\langle\Gamma_2\rangle \parallel Q_1 \parallel Q_2)$ with $\Gamma_1 \leq_{\mathsf{kn}} \Gamma_2$. With Lemma 38, we observe that $\lceil L_2 \rceil^n = \lceil L_2 \otimes n \rceil^0$. By this, the claim follows as $\lceil - \rceil^0$ just omits the iterated parts, i.e. R_2, from $\mathrm{sf}(L_2 \otimes n)$. ◀

▶ **Lemma 52** (Necessary Conditions for Inclusion). *Let L_1 and L_2 be two limits, with $\mathrm{sf}(L_1) \overset{\alpha}{=} \nu\vec{x}_1.(\langle\Gamma_1\rangle \parallel Q_1 \parallel R_1)$ with $R_1 = \prod_{i\in I} B_i^\omega$, and let $n = |\vec{x}_1|+|Q_1|+1$. Given that the inclusion $[\![L_1]\!] \subseteq [\![L_2]\!]$, both conditions* **(A)** *and* **(B)** *hold.*

Proof. For **(A)**, the claim follows by Corollary 51 which implicitly gives a renaming for L_2. For **(B)**, we want to show that $[\![\langle\Gamma_1\rangle \parallel \prod_{i\in I} B_i]\!] \subseteq [\![\langle\Gamma_2\rangle \parallel R_2]\!]$.
Let $N_i = \langle\Gamma_i\rangle \parallel R_i$. It is straightforward to see that $[\![\langle\Gamma_1\rangle \parallel \prod_{i\in I} B]\!] \subseteq [\![N_1]\!]$. This is why it is enough to show that $[\![N_1]\!] \subseteq [\![N_2]\!]$ by transitivity.
Towards a contradiction, we assume that for all possible renaming so that **(A)** is satisfied, $[\![N_1]\!] \not\subseteq [\![N_2]\!]$. By Lemma 36,

$$\exists m_1', \forall m_1 \geq m_1', \forall m_2. \lceil N_1 \rceil^{m_1} \not\sqsubseteq_{\mathsf{kn}} \lceil N_2 \rceil^{m_2}. \tag{$*$}$$

First, knowledge could break the embedding. Let us define

$$\mathrm{sf}(\lceil N_i \rceil^{m_i}) = \nu y_i.(\Gamma_i \parallel \Gamma_i' \parallel Q_i').$$

Notice that $\mathrm{names}(\Gamma_i) \cap \vec{y_i} = \emptyset$. There might be two reasons why the knowledge embedding does not hold.

First, the embedding breaks because of knowledge. This is impossible as Γ_1, Γ_1' and Γ_2, Γ_2' represent the knowledge of groundings of the two limits L_1 and L_2 as their top level knowledge is replicated in **(B)**. Therefore, the inclusion $[\![L_1]\!] \subseteq [\![L_2]\!]$ would also break which is a contradiction.

Second, names or process calls can hence be the only reasons why the knowledge embedding $\lceil N_1 \rceil^{m_1'} \sqsubseteq_{\mathsf{kn}} \lceil N_2 \rceil^{m_2}$ does not hold for any m_2. We will derive a contradiction by choosing $n_1 = 2 \cdot max(|\vec{x}_2|+|Q_2|+1, m_1')$. We incorporate $\lceil R_1 \rceil^{n_1}$ into $\mathrm{sf}(L_1)$: $\lceil \mathrm{sf}(L_1) \rceil^{n_1} = \nu\vec{x}_1.(\Gamma_1 \parallel Q_1 \parallel \lceil R_1 \rceil^{n_1})$. Recall that $\lceil \mathrm{sf}(L_2 \otimes n) \rceil^{m_2} \overset{\alpha}{=} \nu\vec{x}_1, \vec{x}_2.(\langle\Gamma_2\rangle \parallel Q_1 \parallel Q_2 \parallel \lceil R_2 \rceil^{m_2})$. By the size of n_1, at least half of the names and process calls of $\lceil R_1 \rceil^{n_1}$ have to be covered by $\lceil R_2 \rceil^{m_2}$ as the non-iterated part $\vec{x_2}$ and Q_2 cannot do more than half. But by definition $\frac{n_1}{2} \geq m_1'$ so $\forall m_2, \lceil R_1 \rceil^{\frac{n_1}{2}} \not\sqsubseteq_{\mathsf{kn}} \lceil R_2 \rceil^{m_2}$ as knowledge cannot be the reason for $(*)$ to break. Altogether, this entails that there is a n_1 such that for all m_2:

$$\lceil \mathrm{sf}(L_1) \rceil^{n_1} = \nu\vec{x}_1.(\Gamma_1 \parallel Q_1 \parallel \lceil R_1 \rceil^{n_1}) \not\sqsubseteq_{\mathsf{kn}} \nu\vec{x}_1, \vec{x}_2.(\langle\Gamma_2\rangle \parallel Q_1 \parallel Q_2 \parallel \lceil R_2 \rceil^{m_2}) \overset{\alpha}{=} \lceil \mathrm{sf}(L_2 \otimes n) \rceil^{m_2}$$

Using Lemma 37, this implies that $\lceil L_1 \rceil^{n_1} \not\sqsubseteq_{\mathsf{kn}} \lceil L_2 \rceil^{m_2}$ for every m_2. In turn, this entails that $[\![L_1]\!] \not\subseteq [\![L_2]\!]$ by Lemma 36 which is a contradiction. ◀

F.3 Correctness of $\widehat{\mathrm{post}}$

▶ **Definition 53** ($\beta(\Delta)$ and b). *Let Δ be a set of definitions. We define $\beta(\Delta)$ and $\gamma(\mathbb{I})$ as follows:*

$$\beta(\Delta) := \max\{\, |\vec{x}| \mid (\mathsf{Q}[\vec{y}] := A + \mathbf{in}(\vec{x} : M).P + A') \in \Delta \,\}$$
$$\gamma(\mathbb{I}) := \max\{\, \mathrm{ar}(\mathsf{f}) \mid \mathsf{f} \in \Sigma) \,\} \text{ with } \mathbb{I} = (\Sigma, \vdash)$$

then $b := \beta(\Delta) \cdot \gamma(\mathbb{I})^{s-1} + 1$.

Recall the definition of $\widehat{\mathrm{post}}$ from Definition 25. Let $\mathrm{sf}(L \otimes b) = \nu\vec{x}.(\langle\Gamma\rangle \parallel Q \parallel R)$, then

$$\widehat{\mathrm{post}}^s_\Delta(L) := \left\{\; \nu\vec{y}.\big(\langle\Gamma'\rangle \parallel Q' \parallel R\big) \;\middle|\; \nu\vec{x}.\big(\langle\Gamma\rangle \parallel Q\big) \to_\Delta \nu\vec{y}.\big(\langle\Gamma'\rangle \parallel Q'\big) \in \mathbb{S}_s \;\right\}$$

▶ **Corollary 54** (Post of Expansion is enough). *Let L be a limit and $P \in [\![L]\!]$. Then, for every $P_1 \in \mathrm{post}(P)$, there is a $P_2 \in \mathrm{post}(\lceil L \rceil^n)$ for some $n \in \mathbb{N}$ such that $P_1 \sqsubseteq_{\mathsf{kn}} P_2$.*

Proof. As $P \in [\![L]\!]$, we know that there is a $n' \in \mathbb{N}$ such that $P \sqsubseteq_{\mathsf{kn}} \lceil L \rceil^{n'}$ by Lemma 35. We choose this n' to be n. We have that $P \to P_1$ and $P \sqsubseteq_{\mathsf{kn}} \lceil L \rceil^n$. As $\sqsubseteq_{\mathsf{kn}}$ is a simulation (Theorem 8), we know that there is a P_2 so that $\lceil L \rceil^n \to P_2$ and $P_1 \sqsubseteq_{\mathsf{kn}} P_2$. ◀

Let us recall the theorem we want to prove:
For all $L \in \mathbb{L}_{s,k}$, $\widehat{\mathrm{post}}^s_\Delta(L) = \{L_1, \ldots, L_n\} \implies \big(\mathrm{post}([\![L]\!]) \cap \mathbb{S}_s\big)\downarrow = [\![L_1]\!] \cup \ldots \cup [\![L_n]\!]$.

Proof of Theorem 26. We assume that $\widehat{\mathrm{post}}^s_\Delta(L) = \{L_1, \ldots, L_n\}$ and prove the set equality by two inclusions.

First, we show that $\big(\mathrm{post}([\![L]\!]) \cap \mathbb{S}_s\big)\downarrow \supseteq [\![L_1]\!] \cup \ldots \cup [\![L_n]\!]$. It suffices to show that $[\![L_j]\!] \subseteq \big(\mathrm{post}([\![L]\!]) \cap \mathbb{S}_s\big)\downarrow$ for every $j \in \{1, \cdots, n\}$. We choose j arbitrarily and prove the latter statement. By definition, we know that L_j stems from at least one transition in the non-iterated part of $\mathrm{sf}(L \otimes b) = \nu\vec{x}.(\langle\Gamma\rangle \parallel Q \parallel R)$. W.l.o.g. let $P_1 := \nu\vec{x}.\big(\langle\Gamma\rangle \parallel Q\big) \to_\Delta \nu\vec{y}.\big(\langle\Gamma'\rangle \parallel Q'\big) =: P_2$ be this transition in $\mathbb{S}_s$. Hence, $L_j = \nu\vec{y}.\big(\langle\Gamma'\rangle \parallel Q' \parallel R\big)$. By Lemma 37, $[\![L]\!] = [\![L \otimes b]\!]$ and therefore $\lceil \nu\vec{x}.(\langle\Gamma\rangle \parallel Q \parallel R)\rceil^n = \nu\vec{x}.(\langle\Gamma\rangle \parallel Q \parallel \lceil R\rceil^n) \in [\![L]\!]$ for every n. Because of $P_1 \to P_2$, we know that $\lceil \nu\vec{x}.(\langle\Gamma\rangle \parallel Q \parallel R)\rceil^n \to \lceil \nu\vec{y}.(\langle\Gamma'\rangle \parallel Q' \parallel R)\rceil^n$. This is why $\lceil L_j \rceil^n = \lceil \nu\vec{y}.(\langle\Gamma'\rangle \parallel Q' \parallel R)\rceil^n \in \mathrm{post}([\![L]\!])$ for every n. With Lemma 35, we know that for every $P \in [\![L_j]\!]$, there is a m so that $P \sqsubseteq_{\mathsf{kn}} \lceil L_j \rceil^m$. By downward-closure, we infer that $P \in \big([\![\mathrm{post}(L)]\!] \cap \mathbb{S}_s\big)\downarrow$.

Second, we show that $\big(\mathrm{post}([\![L]\!]) \cap \mathbb{S}_s\big)\downarrow \subseteq [\![L_1]\!] \cup \ldots \cup [\![L_n]\!]$. Let $\lceil L \rceil^b$ has standard form: $\mathrm{sf}(\lceil L \rceil^b) \stackrel{\alpha}{=} \nu\vec{x}_1.(\Gamma_1 \parallel \mathsf{Q}[\vec{M}] \parallel C_1)$. By Corollary 54, it suffices to consider only successors of groundings of L. Therefore, let $b < m \in \mathbb{N}$ with $\mathrm{sf}(\lceil L \rceil^m) \stackrel{\alpha}{=} \nu\vec{x}_1, \vec{x_2}.(\langle\Gamma_1\rangle \parallel \langle\Gamma_2\rangle \parallel \mathsf{Q}[\vec{M}] \parallel C_2)$ and $\mathsf{Q}[\vec{M}] := \mathbf{in}(\vec{p} : N).P_1$. W.l.o.g, we derive the message which is matched using some fresh intruder names $\vec{c}$: $\Gamma_1, \vec{c} \vdash N[\vec{M'}/\vec{x}]$. Overall, we get the following transition:

$$\lceil L \rceil^m \quad \xrightarrow[\vec{p} \to \vec{M'}]{\mathsf{Q}[\vec{M}] = \mathbf{in}(\vec{p}:N).P_1 + A} \quad \nu\vec{x}_1, \vec{x}_2, \vec{c}.(\langle\Gamma_1\rangle \parallel \langle\Gamma_2\rangle \parallel \langle\vec{c}\rangle \parallel P_1[\vec{M'}/\vec{p}] \parallel C_1) =: Q' \in \mathrm{post}^s(L)$$

where the annotations explicitly state which process call and action was used with which substitution . We want to show that we can have the same reduction in $\lceil L \rceil^b$. We do so by proving that this transition is still enabled.

Claim I: In every hole of context $C[-, \cdots, -]$, we can reduce the number of branches to at most n, i.e. for every j we can have a grounding $\lceil F_j \rceil^n$ instead of $\lceil F_j \rceil^m$. for every $k \in \mathbb{N}$:

$$\exists \vec{y}_k, \Delta_k, D_k : \Phi^{b,n}_{k,L}(\lceil L \rceil^n) \quad \xrightarrow[\vec{p} \to \vec{M'}]{\mathsf{Q}[\vec{M}] = \mathbf{in}(\vec{p}:N).P_1 + A} \quad \nu\vec{y}_k, \vec{c}.(\langle\Delta_k\rangle \parallel \langle\vec{c}\rangle \parallel P_1[\vec{M'}/\vec{p}] \parallel D_k) \quad \text{with } \Delta, \vec{c} \vdash N[\vec{M'}/\vec{p}] \qquad (*)$$

Proof of Claim I by induction on k: For the base case in which $k = 0$, the claim holds by assumption. For the induction step, we assume that $(*)$ holds for k and we prove it for $k+1$. Similar to the proof of Lemma 50, we use a multi-hole context $C[-, \cdots, -]$ to distinguish between having budget k or $k+1$. Let $F_1, \cdots, F_j$ be j limits and $C[-, \cdots, -]$ a multi-hole context so that:

$$\begin{aligned}
\Phi^{b,n}_{k,L}(\lceil L \rceil^n) &\equiv C[\Phi^{b,n}_{0,F_1^\omega}(F_1^\omega), \cdots, \Phi^{b,n}_{0,F_j^\omega}(F_j^\omega)] = C[\lceil F_1^\omega \rceil^n, \cdots, \lceil F_j^\omega \rceil^n] \\
&= C[(\lceil F_1 \rceil^n)^n, \cdots, (\lceil F_j \rceil^n)^n] = A \\
\Phi^{b,n}_{k+1,L}(\lceil L \rceil^n) &\equiv C[\Phi^{b,n}_{1,F_1^\omega}(F_1^\omega), \cdots, \Phi^{b,n}_{1,F_j^\omega}(F_j^\omega)] \\
&= C[(\lceil F_1 \rceil^n)^b, \cdots, (\lceil F_j \rceil^n)^b] = B
\end{aligned}$$

We want to show that it suffices to have b copies of F_l for any $1 \leq l \leq j$. Since $\mathsf{Q}[\vec{M}]$ also occurs in $\mathrm{sf}(\lceil L \rceil^b)$, it is trivial to keep the process call which is reduced. It remains to argue that the same redex is enabled in B. Towards a contradiction: Assume that there is a hole in which b copies are not sufficient. W.l.o.g. let L_l be the limit in this hole.

We did not explicitly state the message $N[\vec{M'}/\vec{p}]$ but the substitution $\vec{p} \to \vec{M'}$ for the continuation since the substitution is solely determining the successor.

We know that A's knowledge is Δ_k and factor out the knowledge from sublimit $\lceil F_l \rceil^m$: $\Delta_k = \Delta'_k, \Delta_{l,m}$ so that $\Delta_{l,m}$ was the knowledge obtained through $\lceil F_l \rceil^m$. We want to prove that $\Delta'_k, \Delta_{l,m}, \vec{c} \vdash N[\vec{M'}/\vec{p}] \implies \Delta'_k, \Delta_{l,b}, \vec{c} \vdash N[\vec{M'}/\vec{p}]$ where $\Delta_{l,b}$ denotes the knowledge obtained through $\lceil F_l \rceil^b$ respectively.

Now, we consider the different names used in $N[\vec{M'}/\vec{p}]$. It is straightforward to see that $\mathrm{names}(N) \subseteq \vec{x}_1$. Recall the definition of b:

$$b := \beta(\Delta) \cdot \gamma(\mathbb{I})^{s-1} + 1$$

By this, we know that $\beta(\Delta) \geq |\vec{p}|$ and hence $b \geq |\vec{p}| \cdot \gamma(\mathbb{I})^{s-1} + 1$. Intuitively, this ensures that there are enough distinct names for every single parameter in the parameter list as the size determines the maximum depth of the syntax tree of a message. As $\mathrm{names}(\vec{p}) < b$, we can remove at least $m - b$ branches without loosing names used in $\vec{p}$. Therefore, we can assume that $\mathrm{names}(N[\vec{M}/\vec{p}]) \in \vec{x}_1$. The idea is to reduce the knowledge from $\Delta_{l,m}$ to $\Delta_{l,b}$ by Corollary 43, which is a corollary of the absorbing intruder. As m might be bigger than $2b$, we have to iterate the process of reducing the number of branches. Hence, we generalise the notation of $\Delta_{l,m}$ and $\Delta_{l,b}$ in the obvious way: $(\lceil F_l \rceil^m)^i \equiv \nu\vec{a_i}.(\Delta_{l,i} \parallel \cdots)$.

Claim II: $\forall m > b$, $\Delta_{l,m} \vdash N[\vec{M}/\vec{p}] \implies \Delta_{l,m-1} \vdash N[\vec{M}/\vec{p}]$.

Proof of Claim II. For convenience, we rename $\Delta_{l,i}$ to Λ_i. The main observation is that we can split the knowledge Λ_m into Λ_{m-1} and a remainder Λ'. We can choose a branch which does not use names from $\vec{x}_1$ to contribute to Λ'. Since we know that $n \geq 1$, we know that $m > 1$ by assumption. Therefore, we can split Λ_{m-1} again and obtain the knowledge stemming from one branch which we call Λ''. Let us recall the assumption and goal after these rewriting steps: Given that

$$\Delta'_k, \Lambda_{m-2}, \Lambda', \Lambda'' \vdash N[\vec{M}/\vec{p}] \tag{iii}$$

holds, we want to prove that $\Delta'_k, \Lambda_{m-2}, \Lambda'' \vdash N[\vec{M}/\vec{p}]$. Let $\vec{w'}$ and $\vec{w''}$ be the names only used in Λ' and Λ'' respectively so that:

$$\vec{w'} \cap \vec{w''} = \emptyset \text{ and } \mathrm{names}(\Gamma'_{c,m}, \Lambda_{m-2}) \cap \vec{w'} = \emptyset = \mathrm{names}(\Gamma'_{c,m}, \Lambda_{m-2}) \cap \vec{w''} \tag{iv}$$

Λ' and Λ'' have been obtained from a branch of the same sublimit, so we can infer that

$$\Lambda'' = \Lambda'[\vec{w''}/\vec{w'}]. \tag{v}$$

Notice that $\vec{w'} \cap \vec{x}_1 = \emptyset$ by the fact how we have chosen the the branch for Λ'. Because of $\mathrm{names}(N[\vec{M}/\vec{p}]) \subseteq \vec{x_1}$, we can infer that $\mathrm{names}(N[\vec{M}/\vec{p}]) \cap \vec{w'} = \emptyset$ which implies that

$$\mathrm{names}(N[\vec{M}/\vec{p}]) \subseteq \mathrm{names}(\Delta'_k, \Lambda_{m-2}, \Lambda'') \tag{vi}$$

The facts (iii) to (vi) fulfil the conditions for Corollary 43 resulting in $\Delta'_k, \Lambda_{m-2}, \Lambda'' \vdash N[\vec{M}/\vec{p}]$ which reads $\Delta'_k, \Lambda_{m-1} \vdash N[\vec{M}/\vec{p}]$ when folding back which is the goal of Claim II. In turn, this concludes the proof of Claim I.

Instantiating the statement of Claim I with $k > \omega\text{-height}(L)$ shows that this transition is still enabled in $\lceil L \rceil^b$. Consider the extension $L \otimes b$ of limit L whose standard form we

choose to resemble the correlation with $\lceil L \rceil^b$: $\mathrm{sf}(L \otimes b) \overset{\alpha}{=} \nu\vec{x}_1.(\Gamma_1 \parallel \mathsf{Q}[\vec{M}] \parallel C_1 \parallel R_1)$. By the definition of $\widehat{\mathrm{post}}(L)$, we know that

$$L' := \nu\vec{x}_1.(\langle\Gamma_1\rangle \parallel P_1[\vec{M}/\vec{p}] \parallel C_1 \parallel R_1) \in \widehat{\mathrm{post}}(L).$$

Recall that $Q' = \nu\vec{x}_1, \vec{x}_2, \vec{c}.(\langle\Gamma_1\rangle \parallel \langle\Gamma_2\rangle \parallel \langle\vec{c}\rangle \parallel P_1[\vec{M}'/\vec{p}] \parallel C_1)$ was the successor of $\lceil L \rceil^m$ from the beginning. It remains to show that $Q' \in [\![L']\!]$. As before for $L \otimes b$, we relate the standard form of $L \otimes m$ to $\lceil L \rceil^m$ and get the following:

$$\mathrm{sf}(L \otimes m) \overset{\alpha}{=} \nu\vec{x}_1, \vec{x}_2.(\langle\Gamma_1\rangle \parallel \langle\Gamma_2\rangle \parallel \mathsf{Q}[\vec{M}] \parallel C_2 \parallel R_2)$$

By Lemma 37, we have that $[\![L \otimes m]\!] = [\![L \otimes b]\!]$ and taking one step hence leads to

$$\begin{aligned} K' &:= [\![\nu\vec{x}_1, \vec{x}_2, \vec{c}.(\langle\Gamma_1\rangle \parallel \langle\Gamma_2\rangle \parallel \langle\vec{c}\rangle \parallel P_1[\vec{M}'/\vec{p}] \parallel C_2 \parallel R_2)]\!] \\ &= [\![\nu\vec{x}_1.(\langle\Gamma_1\rangle \parallel \langle\vec{c}\rangle \parallel P_1[\vec{M}'/\vec{p}] \parallel C_1 \parallel R_1)]\!] \qquad =: L' \end{aligned}$$

As $Q' \in K'$, it also holds that $Q' \in L'$ which concludes this proof.

◀

G Example of Incorporation

Consider the limit $L = \nu y.\Big((\nu x.(\mathsf{A}[x] \parallel \mathsf{B}[y,x]^\omega))^\omega\Big)$. To compute $\widehat{\mathrm{post}}$ we consider the limit $L \otimes 1 = \nu y.\Big((\nu x.(\mathsf{A}[x] \parallel \mathsf{B}[y,x] \parallel \mathsf{B}[y,x]^\omega)) \parallel (\nu x.(\mathsf{A}[x] \parallel \mathsf{B}[y,x]^\omega))^\omega\Big)$ and assume

$$\mathsf{A}[x] \to \nu z.(A[z] \parallel \mathsf{B}[y,z]) \parallel \mathsf{B}[y,x] = P.$$

Then, $\widehat{\mathrm{post}}$ will contain the limit $L' = C[P]$ for $C[\bullet] = \nu y.\Big((\nu x.(\ \bullet \parallel \mathsf{B}[y,x] \parallel \mathsf{B}[y,x]^\omega)) \parallel (\nu x.(\mathsf{A}[x] \parallel \mathsf{B}[y,x]^\omega))^\omega\Big)$. Since we are trying to prove inclusion between L and $L' = C[P]$, and L is equivalent to $L \otimes 1 = C[\mathsf{A}[x]]$, it seems likely that the inclusion could be proven by matching C with C in the two limits, and by matching P in $L \otimes 1$. Intuitively:

$$\nu y.\Big((\nu x.(\ \mathsf{A}[x]\ \parallel \mathsf{B}[y,x] \parallel \underbrace{\mathsf{B}[y,x]^\omega}\ )) \parallel (\underbrace{\nu x.(\mathsf{A}[x] \parallel \mathsf{B}[y,x]^\omega)}\ )^\omega\Big)$$

$$\nu y.\Big((\nu x.(\overbrace{\nu z.(\mathsf{A}[z] \parallel \mathsf{B}[y,z])}\ \parallel\ \overbrace{\mathsf{B}[y,x]}\ \parallel \mathsf{B}[y,x] \parallel\ \mathsf{B}[y,x]^\omega\ )) \parallel (\nu x.(\mathsf{A}[x] \parallel \mathsf{B}[y,x]^\omega))^\omega\Big)$$

where the part in blue is P and the context C is the part in black. The arrows indicate the sublimits that can be further unfolded to show the inclusion of $C[P]$.

Formally, we want to check that $\forall L' \in \widehat{\mathrm{post}}^s_\Delta(L) : [\![L']\!] \subseteq [\![L]\!]$. By construction, $L' = C[P]$ for some context C such that $L \otimes b = C[\mathsf{Q}[\vec{M}]]$. Then, if we can find P in the fixed part of $\mathrm{sf}(L \otimes n)$, conditions **(A)** and **(B)** of Theorem 23 are automatically satisfied because of the shared context C.