MTR110479

MITRE TECHNICAL REPORT

Completeness of CPSA

October 2011

Moses D. Liskov
Paul D. Rowe
F. Javier Thayer

Sponsor:	**NSA/R2D**	**Contract No.:**	**W15P7T-12-C-F600**
Dept. No.:	**G020**	**Project No.:**	**0712N60B**

MITRE
Center for Integrated Intelligence Systems
Bedford, Massachusetts

Abstract

The Cryptographic Protocol Shapes Analyzer (CPSA) is a program for automatically characterizing the possible executions of a protocol compatible with a specified partial execution. This paper presents a mathematically rigorous theory that backs up the implementation of CPSA in Haskell, and proves the algorithm produces characterizations that are complete, and that the algorithm enumerates these characterizations.

Contents

1 Introduction

The Cryptographic Protocol Shapes Analyzer (CPSA) is a program for automatically characterizing the possible executions of a protocol compatible with a specified partial execution [17]. The purpose of this document is to present a mathematically rigorous theory that backs up the implementation of CPSA in Haskell, and prove the algorithm produces characterizations that are complete, and that the algorithm enumerates these characterizations.

1.1 Previous and Related Work

CPSA is the result of a line of research on the formal analysis of security protocols, typically traced to seminal work of Dolev and Yao [12]. Formal analysis of security protocols treats cryptographic tools such as encryption and digital signatures as abstractions, and thus reduces the problem of analyzing a security protocol to a simpler task. Meadows [16] and Lowe [15] showed that automatic tools for analysis of security protocols are both practical and effective. Numerous tools have been developed since for automated protocol analysis [15, 4, 6, 5, 1, 21, 3, 2, 8, 13].

The cryptographic protocol shapes analyzer is unusual among these tools because it aims to give a *complete characterization* of possible executions, independent of any specific security property to confirm or contradict. CPSA is an automated tool that aims at complete characterization, and works with Strand Space theory [20, 14]. Its structure is described in [9, 11], and the algorithm is more fully specified in [18]. However, no full proof of the algorithm has been given until now.

It is worth noting a few of the similarities and differences from one tool in particular named Scyther [8, 7], created by Cas Cremers. We highlight this tool due to its close similarity to CPSA. Scyther's algorithm is based on the algorithm used in Avispa [1] and also aims to produce complete characterizations of protocols. It is also based on the theory of Strand Spaces, although its semantics is not quite identical to that of CPSA. Scyther's characterizations are sensitive to certain types of adversarial actions, and so these actions are explicitly included in the output, while CPSA focuses solely on the projection of executions onto the regular participants. This difference also manifests itself in the algorithms used by the two tools. In choosing to have characterizations that are insensitive to which adversary actions are used, CPSA must base its algorithm on so-called "authentication tests" [10]. This

appears to add noticeable complexity to the proof of completeness. There has been no systematic comparison of these tools with regard to their performance and expressibility; nor have there been any studies to compare the similarities and differences between the characterizations that the two tools output.

1.2 Document structure

In sections 2, 3, 4, and 5, we describe the formal algebra used to model messages in our protocols and the capabilities of the adversary, and develop some key definitions and reasoning that support the algorithmic design of CPSA. We are mainly concerned with the notion of derivability, specifically, given a message, can the adversary derive it from available messages, and if not, why not? The main result of importance to the proof is the development of the definitions of an *escape set* and a *critical path*, and the relationship between escape sets, critical paths, and derivability, which we prove in Theorem 4.11.

In sections 6, and 7, we describe our mathematical notion of a *protocol* and its roles, and describe *skeletons*, which capture our idea of a (possibly partial) protocol execution. We also describe *homomorphisms*, maps from one skeleton to another, that indicate that the target is an extension of the source as a partial execution. This makes it possible for us to describe *coverage* and to formally explain the goal of the CPSA algorithm.

In sections 8, and 9, we give a mathematical theory of *operators* (transformative operations on skeletons) and *suites* in order to define the *cohort* suite, the main algorithmic operation of CPSA. We also define the overall algorithm of CPSA.

In sections 10, 11, 12, and 13, we formally state and prove the top-level theorems about CPSA we wish to establish: Theorem 10.17, which proves that CPSA gives a *complete* characterization of its input, and Theorem 13.1, which proves that our algorithm enumerates normal characterizations of any input.

2 Preliminaries

2.1 Basic Cryptoalgebra

We use a message algebra called the Basic Cryptoalgebra which is the main algebra used by CPSA.

Sorts

Sorts:	NAME, TEXT, DATA, SKEY, AKEY $<$ MESG
Base sorts:	NAME, TEXT, DATA, SKEY, AKEY

Operations

$\{\!\lvert \cdot \rvert\!\}_{(\cdot)}$	MESG $\times$ MESG $\rightarrow$ MESG	Encryption
$(\cdot, \cdot)$	MESG $\times$ MESG $\rightarrow$ MESG	Pairing
$K_{(\cdot)}$	NAME $\rightarrow$ AKEY	Public key of name
$(\cdot)^{-1}$	AKEY $\rightarrow$ AKEY	Inverse of key
$\mathsf{ltk}(\cdot, \cdot)$	NAME $\times$ NAME $\rightarrow$ SKEY	

Constants

$Tags$	MESG	Tag constants

Equations

$(a^{-1})^{-1} = a$	a: AKEY

The base sorts are pairwise disjoint. Given a set X of generators $\mathfrak{A}_X$ is the free cryptoalgebra generated by X. The set elements of sort base is denoted $\mathfrak{B}_X$.

$$\mathfrak{B} = \mathfrak{A}_{\text{SKEY}} \cup \mathfrak{A}_{\text{AKEY}} \cup \mathfrak{A}_{\text{NAME}} \cup \mathfrak{A}_{\text{TEXT}} \cup \mathfrak{A}_{\text{DATA}} \tag{1}$$

Elements of $\mathfrak{B}$ are called *atoms* in the CPSA Theory paper. The set $\mathfrak{B}$ consists of those terms which are not pairs, not encryptions, not variables of sort message and not tags.

End($\mathfrak{A}$) is the set of homomorphisms $\mathfrak{A} \longrightarrow \mathfrak{A}$. There is a bijective correspondence between elements of End($\mathfrak{A}$) and mappings $X \longrightarrow \mathfrak{A}$. If $\sigma \in \text{End}(\mathfrak{A})$, s-dom σ is the set of variables that are not fixed by σ.

2.2 Path Viewpoint

A *position* π is a finite sequence of whole numbers. We write $\pi^\frown\pi'$ to indicate the concatenation of sequences π and π'. When π' is a prefix of π, we write $(\pi - \pi')$ to indicate the unique sequence π'' such that $\pi' \frown \pi'' = \pi$.

The term in t that *occurs at* π, written $t \mathbin{@} \pi$, is:

$$
\begin{array}{l}
t \mathbin{@} \langle\rangle = t; \\
(t_1, t_2) \mathbin{@} \langle i \rangle \frown \pi = t_i \mathbin{@} \pi \text{ for } i \in \{1, 2\}; \\
\{\!|t_1|\!\}_{t_2} \mathbin{@} \langle i \rangle \frown \pi = t_i \mathbin{@} \pi \text{ for } i \in \{1, 2\}; \\
t^{-1} \mathbin{@} \langle 1 \rangle \frown \pi = t \mathbin{@} \pi.
\end{array}
$$

All references to terms are considered in $\mathfrak{A}$. A term t *occurs in* term t' if $t = t' \mathbin{@} p$ for some p.

We consider the elements of $\mathfrak{A}$ as a directed graph with labeled nodes and arrows; $t \xrightarrow{a} s$ where $t, s \in \mathfrak{A}$ if and only if $t \mathbin{@} \langle a \rangle = s$, where $a \in \{1, 2\}$. Given a term t, the set of paths p from t is denoted by $\mathrm{Path}(t)$.

Remark 2.1. A position π determines a path in the parse tree of a term t. We can associate to each path from t a position π and conversely positions π in a term t determines a path from t. For compatibility with CPSA notation we identify a path p from t with a pair (t, π) where π is a position in t. For any prefix ρ of position π, $t \mathbin{@} \rho$ is a node on the path. We will use similar terminology for paths and positions. For example, a prefix of a path (t, π) is a path (t, ρ) where ρ is a prefix of π. If p is a path from t, then $t \mathbin{@} p$ is the endpoint of the path p. the free algebra.

1. A path $p = (t, \pi)$ *traverses* a term a if $t \mathbin{@} \rho = a$ for some *proper* prefix ρ of π. As a particular case of this, note that any non-null path from t traverses t.

2. A path $p = (t, \pi)$ *terminates* at a if $t \mathbin{@} \pi = a$. Alternative phrases are p leads from t to a or a is an endpoint of p.

3. A path p *visits* a if p traverses a or terminates at a.

4. A path $p = (t, \pi)$ in t *traverses the i-th position* of a function symbol f of arity $n \geq i$ if for some position ρ, $t \mathbin{@} \rho$ is a term with constructor f and $\rho \frown \langle i \rangle$ is a prefix of π. Cases of interest are plaintext edges and key edges of encryptions and key inverse.

5. A path p is *carried* if it does not traverse any key edge or any inverse edge. The set of carried paths at t is denoted $\mathrm{CarPath}(t)$.

6. A term a is *reachable* from t if some path leads from t to a; a is reachable by a carried path if there is a carried path that leads from t to a. In other words, a *occurs in* t if it is reachable from t, and a is *carried by* t if it is reachable from t via a carried path.

3 Upward and Downward Closure

Remarks 3.1 (Notational). *Destructuring* is either one of the two operations which map a pair (a, b) into its components. If a is a term,

$$inv(a) = \begin{cases} a^{-1} & \text{if } a : \text{AKEY} \\ a & \text{if } a \notin X_{\text{MESG}} \\ \bot & \text{otherwise.} \end{cases} \tag{2}$$

Decryption with a key u is the operation $\{\!|a|\!\}_u \mapsto a$. *Encryption with a key u* is the operation $a \mapsto \{\!|a|\!\}_u$.

It is convenient to separate the notion of available terms from the notion of the *context*, which is a set of keys that may be used in derivations. We will use calligraphic fonts $\mathcal{S}$ to emphasize that a set of terms is regarded as a context.

Definition 3.2. *Suppose $\mathcal{S}$ is a set of terms regarded as a cryptographic context. A carried path p is* $\mathcal{S}$-decryptable *if the only encryptions that p* traverses *are of the form $\{\!|b|\!\}_u$ where $inv(u) \in \mathcal{S}$. A carried path p is* $\mathcal{S}$-encryptable *if the only encryptions p traverses are of the form $\{\!|b|\!\}_u$ where $u \in S$.*

An $\mathcal{S}$-decryptable path may terminate at an encryption $\{\!|b|\!\}_u$ with $inv(u) \notin \mathcal{S}$.

Definition 3.3. *A* maximal $\mathcal{S}$-decryptable *carried path is an $\mathcal{S}$-decryptable carried path p which is not a proper prefix of an $\mathcal{S}$-decryptable carried path. Completely analogously, we can define a* maximal $\mathcal{S}$-encryptable *carried path.*

Remark 3.4. Clearly an $\mathcal{S}$-decryptable carried path p is maximal (in the set of $\mathcal{S}$-decryptable carried paths) if and only if it terminates at an encryption $\{\!|b|\!\}_u$ with $inv(u) \notin \mathcal{S}$ or terminates at a term which is not an encryption and not a pair. Similarly an $\mathcal{S}$-encryptable carried path p is maximal (in the set of $\mathcal{S}$-encryptable carried paths) if and only if it terminates at an encryption $\{\!|b|\!\}_u$ with $u \notin \mathcal{S}$ or terminates at a term which is not an encryption and not a pair.

Remark 3.5. For a carried path p from t exactly one of the following alternatives must hold:

1. p is a maximal $\mathcal{S}$-decryptable carried path from t.

2. A proper prefix of p is a maximal $\mathcal{S}$-decryptable carried path from t. This will be the case if and only if p is not $\mathcal{S}$-decryptable.

3. p is a proper prefix of a maximal $\mathcal{S}$-decryptable carried path from t.

There is a corresponding version of the above assertion with "$\mathcal{S}$-decryptable" replaced by "$\mathcal{S}$-encryptable".

Remark 3.6. The maximal $\mathcal{S}$-decryptable prefix of a path p from t if it exists is unique. This is immediate since the set of prefixes of p is totally ordered. However p may have more than one maximal $\mathcal{S}$-decryptable extension. The same remark holds if "$\mathcal{S}$-decryptable" is replaced with "$\mathcal{S}$-encryptable".

Remark 3.7. If a carried path p terminates at a term c such that c is either an encryption $\{\!|b|\!\}_u$ with $inv(u) \notin \mathcal{S}$ or c is not an encryption and not a pair, then p has no proper extensions which are $\mathcal{S}$-decryptable carried paths. In this case some prefix of p (possibly $\langle\rangle$ or p itself) is a maximal $\mathcal{S}$-decryptable carried path from t. In the other direction, if p is an $\mathcal{S}$-decryptable carried path that has no proper extensions which are $\mathcal{S}$-decryptable carried paths, then p terminates at a term c such that c is either an encryption $\{\!|b|\!\}_u$ with $inv(u) \notin \mathcal{S}$ or c is not an encryption and not a pair. Again there is a corresponding statement for maximal $\mathcal{S}$-encryptable carried path: It terminates at a term c such that c is either an encryption $\{\!|b|\!\}_u$ with $u \notin \mathcal{S}$ or c is not an encryption and not a pair.

First we adapt some terminology which is more-or-less standard in the context of paths and graphs: If p is a path, t a term and L a set of terms such that p visits an element of L, the *first L-visit prefix* of p at t is the minimal prefix of p which visits an element of L.

Definition 3.8. *Given a set L of terms, the* depth *of L relative to a term a, denoted* $\mathrm{depth}(L, \mathcal{S}, a)$*, is the supremum over all maximal $\mathcal{S}$-encryptable carried paths p from a of the length of the first L-visit prefix of p. By convention, if there is a maximal $\mathcal{S}$-encryptable carried path* which does not visit L*, then* $\mathrm{depth}(L, \mathcal{S}, a) = +\infty$.

A set L is an $\mathcal{S}$-support for a if $\mathrm{depth}(L, \mathcal{S}, a) < \infty$*. Alternative phrase: a is $\mathcal{S}$-supported by L.*

Remark 3.9. For every term a, $\{a\}$ is an $\mathcal{S}$-support of a. Clearly a is $\mathcal{S}$-supported by L if and only if every maximal $\mathcal{S}$-encryptable carried path visits L.

Definition 3.10. *Let P and $\mathcal{S}$ be sets of terms, where we regard $\mathcal{S}$ as a context. The* $\mathcal{S}$-downclosure of P *denoted* $\mathrm{Cl}^{\downarrow}(P, \mathcal{S})$ *consists of those terms a which are endpoints of an $\mathcal{S}$-decryptable carried path beginning at some element of P. The* $\mathcal{S}$-upclosure of P *denoted* $\mathrm{Cl}^{\uparrow}(P, \mathcal{S})$ *consists of those terms a which are $\mathcal{S}$-supported by P.*

The $\mathcal{S}$-frontier of P *denoted* $\mathrm{Fr}(P, \mathcal{S})$ *consists of those terms which are endpoints of a maximal $\mathcal{S}$-decryptable carried path beginning at some element of P.*

Remark 3.11. Taking the contrapositive of Definition 3.10, $a \notin \mathrm{Cl}^{\downarrow}(P, \mathcal{S})$ if and only if there is no $\mathcal{S}$-decryptable carried path starting at an element of P that reaches a.

Equivalently,

Lemma 3.12. *$a \notin \mathrm{Cl}^{\downarrow}(P, \mathcal{S})$ if and only if every carried path from an element of P to a traverses* $\mathrm{Fr}(P, \mathcal{S})$.

Proof. Suppose $a \notin \mathrm{Cl}^{\downarrow}(P, \mathcal{S})$ and p is a path from t to a. By Definition 3.10, p is not $\mathcal{S}$-decryptable, and therefore has a proper prefix q which is maximal $\mathcal{S}$-decryptable. The endpoint e_q of the path q is an element of $\mathrm{Fr}(P, \mathcal{S})$ and clearly p traverses e_q. Conversely, if $a \in \mathrm{Cl}^{\downarrow}(P, \mathcal{S})$ then there is an $\mathcal{S}$-decryptable path p from an element of P to a. Every encryption c traversed by p is $\mathcal{S}$-decryptable and therefore by Remark 3.4, no such c can be the endpoint of a maximal $\mathcal{S}$-decryptable path *from anywhere*. Therefore no such c can be an element of $\mathrm{Fr}(P, \mathcal{S})$. Thus p traverses no element of $\mathrm{Fr}(P, \mathcal{S})$. □

Remark 3.13. In other words, Lemma 3.12 asserts that $a \notin \mathrm{Cl}^{\downarrow}(P, \mathcal{S})$ if and only for all $t \in P$, a is carried in t *only within* $\mathrm{Fr}(P, \mathcal{S})$. Some elements of $\mathrm{Fr}(P, \mathcal{S})$ may not carry a at all.

Remark 3.14. By Remark 3.4 any element in $\mathrm{Fr}(P, \mathcal{S})$ is either an encryption of the form $\{|b|\}_u$ with $inv(u) \notin \mathcal{S}$ or is not an encryption and not a pair or equivalently, either an encryption or an atom. Every element of $\mathrm{Cl}^{\downarrow}(P, \mathcal{S})$ is visited by an $\mathcal{S}$-decryptable path starting at some element of P.

The frontier has a boundary-like property:

Proposition 3.15. *If $t \in \mathrm{Cl}^{\downarrow}(P, \mathcal{S})$, $a \notin \mathrm{Cl}^{\downarrow}(P, \mathcal{S})$ and p is a path from t to a then p traverses* $\mathrm{Fr}(P, \mathcal{S})$.

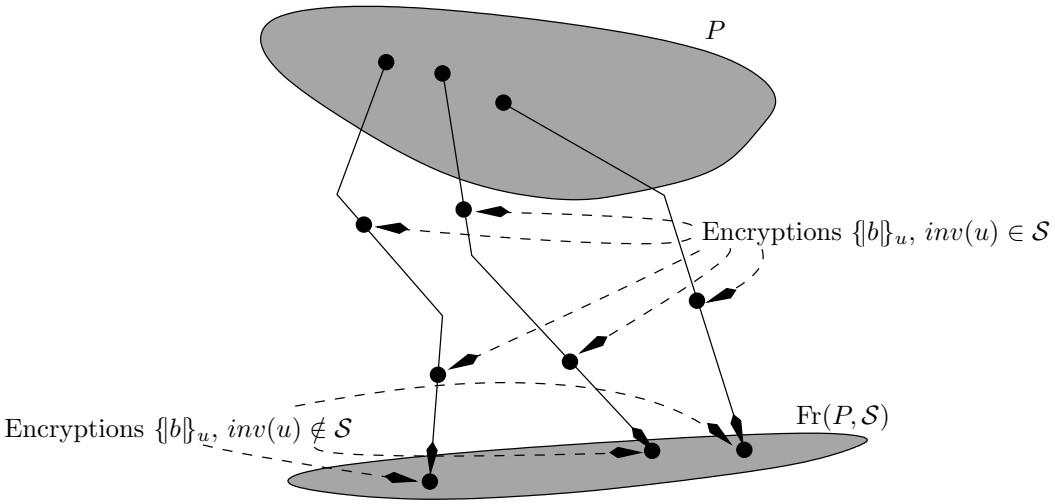


Figure 1: Frontier

Proof. By Definition 3.10, there is an $\mathcal{S}$-decryptable path r from some element of P to t. Now consider the path $r \frown p$. Again by Definition 3.10, $r \frown p$ is not $\mathcal{S}$-decryptable. Therefore $r \frown p$ has a proper maximal $\mathcal{S}$-decryptable prefix q. However q must be an extension of r, possibly r itself. Moreover, the endpoint e_q of q is an element of the frontier $\mathrm{Fr}(P, \mathcal{S})$ and an element of $\mathrm{Cl}^{\downarrow}(P, \mathcal{S})$. Since $r \frown p$ is a path from an element of P to a, e_q is an element of $\mathrm{Fr}(P, \mathcal{S})$. It follows that p visits $e_q \in \mathrm{Fr}(P, \mathcal{S})$ as claimed and since $a \neq e_q$, p traverses e_q. □

3.1 Closure Properties

A set Z of terms is *derivational* if it is closed under pairing, encryption with a key u such that $u \in Z$, destructuring and decryption with a key u such that $inv(u) \in Z$. The smallest derivational set containing P is denoted $D(P)$.

A set Z of terms is $\mathcal{S}$*-constructive* if and only if it is closed under pairing and encryption with a key u such that $u \in \mathcal{S}$. A set Z is $\mathcal{S}$*-deconstructive* if and only if it is closed under destructuring and decryption with a key u such that $inv(u) \in \mathcal{S}$. A set Z is $\mathcal{S}$*-derivational* if and only if it is $\mathcal{S}$-constructive and $\mathcal{S}$-deconstructive. The smallest $\mathcal{S}$-derivational set containing P is $D(P, \mathcal{S})$.

We can characterize these sets in terms of the path heuristic.

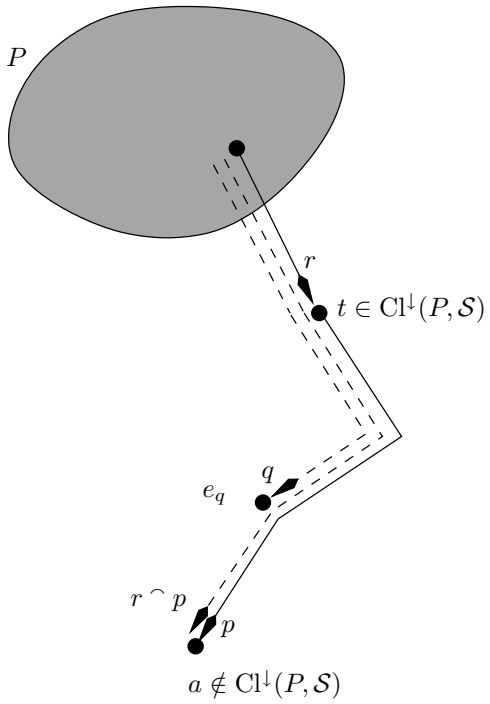


Figure 2: Frontier as a Boundary (Proposition 3.15)

Lemma 3.16. *A set Z is $\mathcal{S}$-deconstructive if and only if every term a which is the endpoint of an $\mathcal{S}$-decryptable carried path p from $t \in Z$, is also an element of Z.*

Proof. Suppose Z is $\mathcal{S}$-deconstructive. We show that if a term a is reachable from $t \in Z$ by an $\mathcal{S}$-decryptable carried path p, then $a \in Z$. We use induction on the length of the path p. If p has length 0, then $p = \langle\rangle$ and $a = t \in Z$. Suppose the claim is true for all carried paths of length n and p has length $n+1$. Then $p = q \frown \langle a \rangle$ where q has length n. q is an $\mathcal{S}$-decryptable carried path and therefore by the induction hypothesis $t \mathbin{@} q \in Z$. If $t \mathbin{@} q$ is an encryption $\{\!|b|\!\}_u$, then by the assumption p is $\mathcal{S}$-decryptable, $inv(u) \in \mathcal{S}$. Since Z is $\mathcal{S}$-deconstructive, $t \mathbin{@} p = b \in Z$. If $t \mathbin{@} q$ is a pair (x, y) then both $x, y \in Z$ and $t \mathbin{@} p$ is either x or y.

Suppose that every a reachable from $t \in Z$ by an $\mathcal{S}$-decryptable carried path p is an element of Z. If $a = (x, y)$ then x and y are reachable from t by the carried paths $p \frown \langle 1 \rangle$, $p \frown \langle 2 \rangle$ and therefore $x, y \in Z$. If $a = \{\!|b|\!\}_u$

with $inv(u) \in \mathcal{S}$, then $p \frown \langle 1 \rangle$ is an $\mathcal{S}$-decryptable path so $b \in Z$. Thus Z is $\mathcal{S}$-deconstructive. □

There is an analogous statement, Lemma 3.17, for $\mathcal{S}$-constructive sets, but this requires the notion of support.

Lemma 3.17. *A set Z is $\mathcal{S}$-constructive if and only if every term t which is $\mathcal{S}$-supported by Z is an element of Z.*

Proof. Suppose Z is $\mathcal{S}$-constructive. We show that for every integer $n \geq 0$, if $\mathrm{depth}(Z, \mathcal{S}, t) \leq n$, then $t \in Z$. We use induction on n. If $n = 0$, then $t \in Z$ and $\{t\}$ is an $\mathcal{S}$-support set for t. Suppose the assertion holds for $n-1$ and $\mathrm{depth}(Z, \mathcal{S}, t) = n \geq 1$. In particular, t is either a pair or an encryption. If $t = (x, y)$ then $\mathrm{depth}(Z, \mathcal{S}, x) \leq n - 1$ and similarly for y. Thus $x \in Z$ and $y \in Z$ by the induction hypothesis. Since Z is $\mathcal{S}$-constructive, $t \in Z$. If $t = \{\!|b|\!\}_u$ with $u \in \mathcal{S}$, then $\mathrm{depth}(Z, \mathcal{S}, b) \leq n - 1$. By the inductive hypothesis $b \in Z$. Since Z is $\mathcal{S}$-constructive, $t \in Z$.

Conversely, if every t which is $\mathcal{S}$-supported by Z is an element of Z it is straightforward to show Z is constructive. □

Corollary 3.18. *The smallest $\mathcal{S}$-constructive set containing P is the set of terms t which are $\mathcal{S}$-supported by P.*

Proposition 3.19. $\mathrm{Cl}^{\downarrow}(P, \mathcal{S})$ *is the smallest $\mathcal{S}$-deconstructive set containing P.*

Proof. Let Z be the smallest $\mathcal{S}$-deconstructive set containing P. By Lemma 3.16, $\mathrm{Cl}^{\downarrow}(P, \mathcal{S})$ is $\mathcal{S}$-deconstructive. Since $\mathrm{Cl}^{\downarrow}(P, \mathcal{S}) \supseteq P$, it follows that $Z \subseteq \mathrm{Cl}^{\downarrow}(P, \mathcal{S})$. Po prove the converse, by definition Z is $\mathcal{S}$-deconstructive and so by Lemma 3.16 if $t \in Z$ and p is an $\mathcal{S}$-decryptable carried path from t then $t@p \in Z$. Since $P \subseteq Z$, it follows $Z \supseteq \mathrm{Cl}^{\downarrow}(P, \mathcal{S})$. □

Analogously, applying Lemma 3.17,

Proposition 3.20. $\mathrm{Cl}^{\uparrow}(P, \mathcal{S})$ *is the smallest $\mathcal{S}$-constructive set containing P.*

Lemma 3.21. *If Z is $\mathcal{S}$-deconstructive, then* $\mathrm{Cl}^{\uparrow}(Z, \mathcal{S})$ *is $\mathcal{S}$-deconstructive.*

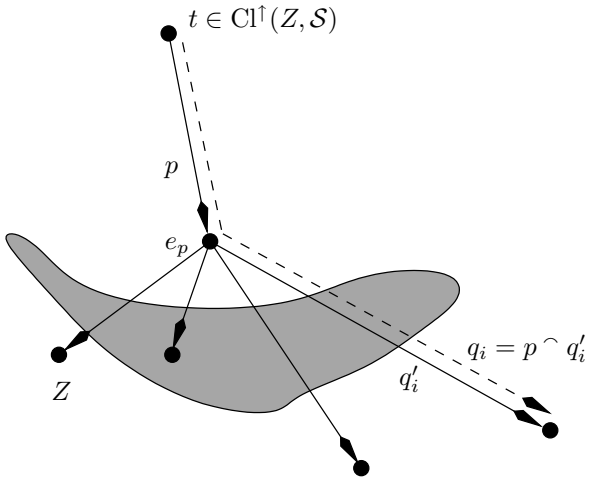


Figure 3: $\mathrm{Cl}^{\uparrow}(Z.\mathcal{S})$ is $\mathcal{S}$-deconstructive (Case (2))

Proof. By Lemma 3.16, it suffices to show that the endpoint of every carried $\mathcal{S}$-decryptable path from $t \in \mathrm{Cl}^{\uparrow}(Z, \mathcal{S})$ is also in $\mathrm{Cl}^{\uparrow}(Z, \mathcal{S})$. Let p be an $\mathcal{S}$-decryptable path from $t \in \mathrm{Cl}^{\uparrow}(Z, \mathcal{S})$ with endpoint e_p. By the $\mathcal{S}$-encryptable version of Remark 3.5, one of two things must hold for p: (1) there is a maximal $\mathcal{S}$-encryptable path q which is a (not necessarily proper) prefix of p, or (2) p has maximal $\mathcal{S}$-encryptable proper extensions. Let $q_1, \ldots, q_\ell$ be all the maximal $\mathcal{S}$-encryptable extensions.

Consider case (1). By Corollary 3.18 and Proposition 3.20, every element of $\mathrm{Cl}^{\uparrow}(Z, \mathcal{S})$ is $\mathcal{S}$-supported by Z, and so some prefix q' of q terminates in an element of Z. Thus we can write $p = q' \frown q''$ where q'' is an $\mathcal{S}$-decryptable path from an element of Z to e_p. Since Z is $\mathcal{S}$-deconstructive, $e_p \in \mathrm{Cl}^{\uparrow}(Z, \mathcal{S})$.

In case (2), we can write $q_i = p \frown q_i'$ for each $1 \leq i \leq \ell$ where each q_i' is a maximal $\mathcal{S}$-encryptable path starting at e_p. Again, since $\mathrm{Cl}^{\uparrow}(Z, \mathcal{S})$ is $\mathcal{S}$-supported by Z, we know that each q_i visits an element of Z. If this happens with some q' which is a prefix of p, then we can argue as we did in case (1) that e_p must be in $\mathrm{Cl}^{\uparrow}(Z, \mathcal{S})$. Otherwise, we are guaranteed that each q_i' visits an element of Z. Since these q_i are the maximal $\mathcal{S}$-encryptable paths of e_p, we know e_p is $\mathcal{S}$-supported by Z. By Lemma 3.17 $e_p \in \mathrm{Cl}^{\uparrow}(Z, \mathcal{S})$. □

The previous results immediately yield:

Proposition 3.22. $D(P, \mathcal{S}) = \mathrm{Cl}^{\uparrow}(\mathrm{Cl}^{\downarrow}(P, \mathcal{S}), \mathcal{S})$.

In particular, by the characterization of the $\mathcal{S}$-upclosure of a set given by Proposition 3.20:

Corollary 3.23. *A necessary and sufficient condition $t \in D(P, \mathcal{S})$ is that t be $\mathcal{S}$-supported by some subset of $\mathrm{Cl}^{\downarrow}(P, \mathcal{S})$.*

Corollary 3.24. *A necessary and sufficient condition $t \in D(P, \mathcal{S})$ is that every maximal $\mathcal{S}$-encryptable path p beginning at t visit some element of $\mathrm{Cl}^{\downarrow}(P, \mathcal{S})$.*

Proof. Apply Remark 3.9 and Corollary 3.23. □

Corollary 3.24 leads to the notion of essential obstruction and critical path in the next section.

Proposition 3.25. *Suppose $t \in D(P, \mathcal{S})$, p is a path from t to a and either (i) $a = \{|b|\}_u$ with $u \notin \mathcal{S}$ or (ii) a is not an encryption and not a pair. Then either p traverses $\mathrm{Fr}(P, \mathcal{S})$ or $a \in \mathrm{Cl}^{\downarrow}(P, \mathcal{S})$.*

Proof. Consider case (i): Suppose $a \notin \mathrm{Cl}^{\downarrow}(P, \mathcal{S})$. Since $u \notin \mathcal{S}$, by Remark 3.7, the path p is has a maximal $\mathcal{S}$-encryptable prefix (possibly p itself.) By Corollary 3.24, that prefix must visit some element b of $\mathrm{Cl}^{\downarrow}(P, \mathcal{S})$; in particular p visits b. Since $a \notin \mathrm{Cl}^{\downarrow}(P, \mathcal{S})$, it is in fact a *proper* prefix q of p that visits b. Let r be the remnant of p after q. r is a path from b to a. By Proposition 3.15, r traverses $\mathrm{Fr}(P, \mathcal{S})$.

In case (ii) the argument is identical, since the only fact used was that the path p is has a maximal $\mathcal{S}$-encryptable prefix. □

4 Critical Path

Definition 4.1. *An* essential obstruction *of t relative to $P, \mathcal{S}$ is a maximal $\mathcal{S}$-encryptable path beginning at t which does not visit an element of $\mathrm{Cl}^{\downarrow}(P, \mathcal{S})$.*

The set of essential obstructions is denoted $\mathrm{Eob}(P, \mathcal{S}, t)$.

Remark 4.2. The content of Corollary 3.24 is that $t \in D(P, \mathcal{S})$ if and only if $\mathrm{Eob}(P, \mathcal{S}, t) = \emptyset$.

For the CPSA search algorithm, in particular for its notion of "progress" the set of essential obstructions is too small. This leads to the notion of *critical path.*

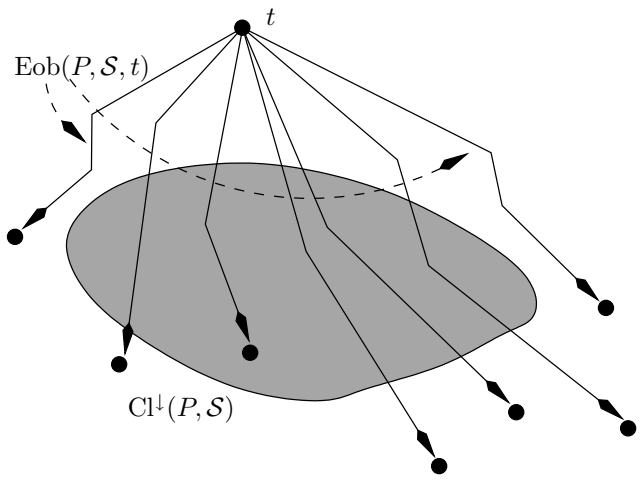


Figure 4: Essential Obstruction

Definition 4.3. *Let $p \in \mathrm{CarPath}(t)$ have endpoint e_p. p is* critical *relative to $P, \mathcal{S}$ if and only (1) e_p is an encryption $\{\!|b|\!\}_u$ with $u \notin \mathcal{S}$ or an element which is not a pair and not an encryption (2) $e_p \notin \mathrm{Cl}^{\downarrow}(P, \mathcal{S})$ and (3) p does not visit $\mathrm{Fr}(P, \mathcal{S})$.*

The set of critical paths is denoted $\mathrm{CritPath}(P, \mathcal{S}, t)$.

Proposition 4.4. *Let $p \in \mathrm{CarPath}(t)$ have endpoint e_p. A necessary and sufficient condition for p to be critical relative to $P, \mathcal{S}$ is that (A) e_p is an encryption $\{\!|b|\!\}_u$ with $u \notin \mathcal{S}$ or an element which is not a pair and not an encryption, (B) p does not visit an element of $\mathrm{Cl}^{\downarrow}(P, \mathcal{S})$.*

Proof. Conditions (1) of Definition 4.3 and (A) are identical. Now suppose $p \in \mathrm{CarPath}(t)$ is such that (1) holds. Then properties (2) and (3) of Definition 4.3 are equivalent to (B): In one direction, suppose (B) holds, that is, p does not visit $\mathrm{Cl}^{\downarrow}(P, \mathcal{S})$. Since p visits e_p, it follows that $e_p \notin \mathrm{Cl}^{\downarrow}(P, \mathcal{S})$ which is (2). By definition $\mathrm{Fr}(P, \mathcal{S}) \subseteq \mathrm{Cl}^{\downarrow}(P, \mathcal{S})$, and therefore p cannot visit any element of $\mathrm{Fr}(P, \mathcal{S})$ proving (3). In the other direction, suppose (2) and (3) hold, but p visits $a \in \mathrm{Cl}^{\downarrow}(P, \mathcal{S})$. Now $a \neq e_p$, for otherwise p is a maximal $\mathcal{S}$-encryptable path from an element of P to a (by Remark 3.7) and therefore $e_p = a \in \mathrm{Fr}(P, \mathcal{S}) \subseteq \mathrm{Cl}^{\downarrow}(P, \mathcal{S})$ which contradicts (2). Otherwise, let q be the remnant of p from a. By the boundary property of $\mathrm{Fr}(P, \mathcal{S})$ (Proposi-

tion 3.15), q visits an element of $\mathrm{Fr}(P, \mathcal{S})$. Therefore p visits an element of $\mathrm{Fr}(P, \mathcal{S})$ which contradicts (3). □

Remark 4.5. Any essential obstruction is a critical path. Proof: By the version of Remark 3.7 for $\mathcal{S}$-encryptable paths, any essential obstruction satisfies (A) of Proposition 4.4. (B) is part of the definition of essential obstruction.

Remark 4.6. If $p \in \mathrm{CritPath}(P, \mathcal{S}, t)$, by Remark 3.6 p has a unique maximal $\mathcal{S}$-encryptable prefix $\mu(p)$. Since p does not visit $\mathrm{Cl}^{\downarrow}(P, \mathcal{S})$, no prefix of p can visit $\mathrm{Cl}^{\downarrow}(P, \mathcal{S})$. In particular, $\mu(p) \in \mathrm{Eob}(P, \mathcal{S}, t)$. Now $\mathrm{Eob}(P, \mathcal{S}, t) \subseteq \mathrm{CritPath}(P, \mathcal{S}, t)$ and μ is the identity on $\mathrm{Eob}(P, \mathcal{S}, t)$ and the mapping $\mu : \mathrm{CritPath}(P, \mathcal{S}, t) \longrightarrow \mathrm{Eob}(P, \mathcal{S}, t)$ is the identity on $\mathrm{Eob}(P, \mathcal{S}, t)$.

Proposition 4.7. *A necessary and sufficient condition $t \in D(P, \mathcal{S})$ is that*

$$\mathrm{CritPath}(P, \mathcal{S}, t) = \emptyset. \tag{3}$$

Proof. Consider the map $\mu : \mathrm{CritPath}(P, \mathcal{S}, t) \longrightarrow \mathrm{Eob}(P, \mathcal{S}, t)$ defined in Remark 4.6. The map μ is surjective and therefore $\mathrm{CritPath}(P, \mathcal{S}, t) \neq \emptyset \iff \mathrm{Eob}(P, \mathcal{S}, t) \neq \emptyset$. The result now follows by Remark 4.2. □

Definition 4.8. *The* escape set *of a relative to $P, \mathcal{S}$, denoted* $\mathrm{Esc}(P, \mathcal{S}, a)$ *is the union of the set of those elements of* $\mathrm{Fr}(P, \mathcal{S})$ *which carry a with the set $\{a\}$ if $a \in \mathrm{Cl}^{\downarrow}(P, \mathcal{S})$.*

Remark 4.9. By Remark 3.14, every element of $\mathrm{Esc}(P, \mathcal{S}, a)$ other than a is either an encryption or an atom.

Proposition 4.10. *If $p \in \mathrm{CarPath}(t)$ with endpoint e_p and $p \in \mathrm{CritPath}(P, \mathcal{S}, t)$, then the following conditions hold:*

1. *p does not visit* $\mathrm{Esc}(P, \mathcal{S}, e_p)$.
2. *Every path from an element of P to e_p visits* $\mathrm{Esc}(P, \mathcal{S}, e_p)$.
3. *For every encryption $\{\!|b|\!\}_u \in \mathrm{Esc}(P, \mathcal{S}, e_p)$, $inv(u) \notin \mathcal{S}$.*
4. *If e_p is an encryption $\{\!|b|\!\}_u$, then $u \notin \mathcal{S}$.*

Proof. Suppose $p \in \mathrm{CritPath}(P, \mathcal{S}, t)$. By Definition 4.3, p does not visit $\mathrm{Fr}(P, \mathcal{S})$. Moreover, also by Definition 4.3, $e_p \notin \mathrm{Cl}^{\downarrow}(P, \mathcal{S})$. Thus $\mathrm{Esc}(P, \mathcal{S}, e_p) \subseteq \mathrm{Fr}(P, \mathcal{S})$ by the Definition 4.8 of escape set. This proves (1). Since $e_p \notin \mathrm{Cl}^{\downarrow}(P, \mathcal{S})$, any path from P to e_p must traverse some element $d \in \mathrm{Fr}(P, \mathcal{S})$ by the boundary property of the frontier (Proposition 3.15). d carries e_p and therefore $d \in \mathrm{Esc}(P, \mathcal{S}, e_p)$. This proves (2). Property (3) is immediate from the definition of frontier. Finally Property (4) follows from the definition of critical path.

□

Theorem 4.11. *Suppose $p \in \mathrm{CarPath}(t)$ with endpoint e_p and $E \subseteq \mathfrak{A}$ consists of encryptions or atoms. If $p \notin \mathrm{CritPath}(P, \mathcal{S}, t)$, then one of the following conditions holds:*

1. *p visits E.*
2. *There is a path from an element of P to e_p which does not visit E.*
3. *For some $\{\!|b|\!\}_u \in E$, $inv(u) \in \mathcal{S}$.*
4. *e_p is an encryption $\{\!|b|\!\}_u$ with $u \in \mathcal{S}$.*

Proof. Since $p \notin \mathrm{CritPath}(P, \mathcal{S}, t)$, by taking the contrapositive of Proposition 4.4, we conclude that the one of the following must hold:

(A) It is not the case that e_p is an encryption of the form $\{\!|b|\!\}_u$ with $u \notin \mathcal{S}$ and e_p is not an atom.

(B) p visits an element of $\mathrm{Cl}^{\downarrow}(P, \mathcal{S}, t)$.

Case (A) is equivalent to: e_p is an encryption of the form $\{\!|b|\!\}_u$ with $u \in \mathcal{S}$. Thus in case (A), condition (4) of the Lemma holds. Thus we may henceforth assume that (A) fails and (B) holds. In particular, p visits an element d of $\mathrm{Cl}^{\downarrow}(P, \mathcal{S}, t)$. Let q be the remnant of p starting at d, and let r be an $\mathcal{S}$-decryptable path from an element of P to d. In this case one of the following statements holds:

(a) p visits E

(b) r visits E

(c) Neither p nor r visit E

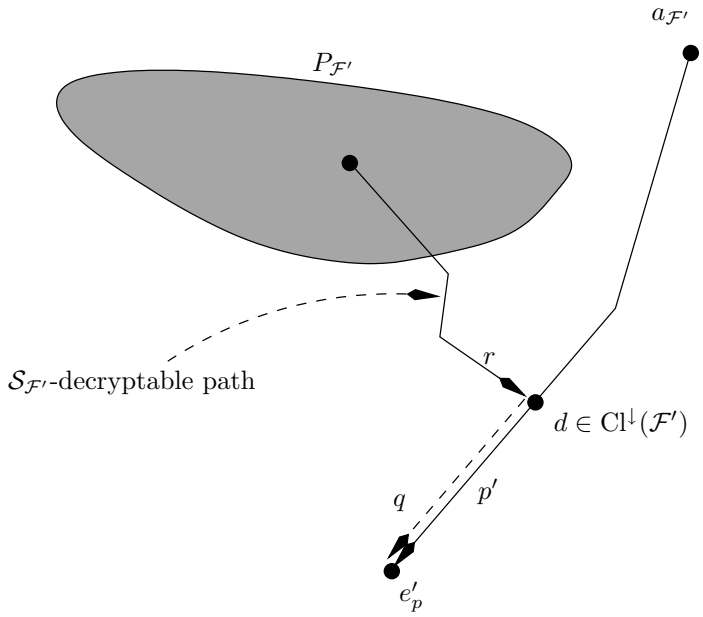


Figure 5: Case (B) of Lemma 5.6

In case (a), this is simply Condition (1) of the lemma. If (a) case does not hold but (b) does hold, then we conclude that r *traverses* an element $x \in E$, for if r terminated at x, then p would visit E contrary to the assumption that (a) does not hold. By hypothesis, x is either an encryption or atom, but x cannot be an atom since no path can traverse an atom. Thus x is of the form $\{\!|c|\!\}_v$. Since r is $\mathcal{S}$-decryptable, $inv(v) \in \mathcal{S}$, thereby meeting Condition (3) of the lemma. In case (c), the carried path $r^\frown q$ is a path from P to e_p which does not visit E, thereby satisfying Condition (2) of the lemma. □

5 Fragments

So far we have considered derivability in the basic cryptoalgebra setting. This section introduces a further refinement in which the available messages and the context are determined by other considerations which occur naturally in the setting of a protocol execution. In the section on the adversary model, it is explained why the penetrator cannot use terms in the what is called the exclusion set. Recall that $\mathfrak{B} = \mathfrak{A}_{\mathrm{SKEY}} \cup \mathfrak{A}_{\mathrm{AKEY}} \cup \mathfrak{A}_{\mathrm{NAME}} \cup \mathfrak{A}_{\mathrm{TEXT}} \cup \mathfrak{A}_{\mathrm{DATA}}$

Definition 5.1. *A* fragment *consists of a tuple* $\mathcal{F} = (T, X, a)$ *where* $X \subseteq \mathfrak{B}$ *and* a *is a term. The* X *in a fragment is called the* exclusion set*. The set of public messages at* $\mathcal{F}$ *is*

$$P_{\mathcal{F}} = T \cup (\mathfrak{B} \setminus X) \cup X_{\text{MESG}} \cup \text{TAGS}\,. \tag{4}$$

The encryption context at $\mathcal{F}$ *is* $\mathcal{S}_{\mathcal{F}} = D(P_{\mathcal{F}})$*. If* $\mathcal{F}$ *is an fragment,* $T_{\mathcal{F}}$*,* $X_{\mathcal{F}}$*, and* $a_{\mathcal{F}}$ *are the components of* $\mathcal{F}$*.*

Definition 5.2. *The critical pathset of a fragment* $\mathcal{F}$ *is the set* $\text{CritPath}(\mathcal{F}) = \text{CritPath}(P_{\mathcal{F}}, \mathcal{S}_{\mathcal{F}}, a_{\mathcal{F}})$*. The critical pathset at* $\mathcal{F}$ *is denoted* $\text{CritPath}(\mathcal{F})$*. The* escape set *of a term* a *in* $\mathcal{F}$ *is* $\text{Esc}(\mathcal{F}, a) = \text{Esc}(P_{\mathcal{F}}, \mathcal{S}_{\mathcal{F}}, a)$*. Similar conventions apply to the sets* $\text{Cl}^{\uparrow}$*,* $\text{Cl}^{\downarrow}$*,*

Remark 5.3. By virtue of (4), $X_{\text{MESG}} \cup \text{TAGS} \subseteq \text{Cl}^{\downarrow}(\mathcal{F})$.

In the intended interpretation of a fragment for skeletons, U corresponds to the declared uniquely originating atoms which actually originate in $\mathbb{A}$.

Definition 5.4. *If* $\mathcal{F} = (T, X, a)$*,* $\mathcal{F}' = (T', X', a')$ *are fragments, a homomorphism* $\sigma : \mathcal{F} \longrightarrow \mathcal{F}'$ *is an algebra homomorphism such that* $\sigma(T) \subseteq T'$*,* $\sigma(X) \subseteq X'$*, and* $\sigma(a) = a'$*.*

We now consider possible ways of resolving a critical path.

Definition 5.5 (Critical Path Solved)**.** *Suppose* $\mathcal{F}$*,* $\mathcal{F}'$ *are fragments and* p *is a critical path with endpoint* e_p *of the term* $a_{\mathcal{F}}$ *at* $\mathcal{F}$*. Let* $\sigma : \mathfrak{A} \longrightarrow \mathfrak{A}$ *be an algebra homomorphism. Let* $E' = \sigma(\text{Esc}(\mathcal{F}, e_p))$*,* $p' = \sigma(p)$ *and* $e'_p = \sigma(e_p)$*. Path* p *from* $a_{\mathcal{F}}$ *is* solved *in* $\mathcal{F}'$ *by* σ*, if and only if one of the following conditions holds:*

- *Sol1.* p' *visits* E'*.*
- *Sol2. There is a carried path from an element of* $T_{\mathcal{F}'}$ *to* e'_p *which does not visit* E'*.*
- *Sol3. For some* $\{\!|b|\!\}_u \in E'$*,* $inv(u) \in \mathcal{S}_{\mathcal{F}'}$*.*
- *Sol4.* $e'_p = \{\!|b|\!\}_u$*, and* $u \in \mathcal{S}_{\mathcal{F}'}$*.*

Let $\text{Solved}(\mathcal{F}, p) = \{(\sigma, \mathcal{F}') : \sigma$ *is a homomorphism* $\mathfrak{A} \longrightarrow \mathfrak{A}$*. and* σ *solves* p *in* $\mathcal{F}'\ \}$

In words, CPSA makes progress by a contraction (Item 1), where messages are identified, an augmentation (Item 2), where something is added to the escape set, or a listener augmentation (Item 3 and Item 4), where an assumption about the lack of the derivability of a key is shown to be invalid.

Lemma 5.6. *Suppose $\mathcal{F}$, $\mathcal{F}'$ are fragments and $\mathfrak{A} \xrightarrow{\sigma} \mathfrak{A}$ is an algebra homomorphism. If $p \in \mathrm{CritPath}(\mathcal{F})$ and $\sigma(p) \notin \mathrm{CritPath}(\mathcal{F}')$, then σ solves p in $\mathcal{F}'$.*

Proof. Let e_p be the endpoint of p. By Proposition 4.4, e_p is either (a) an encryption of the form $\{\!|b|\!\}_u$ with $u \notin \mathcal{S}_{\mathcal{F}}$ or (b) not a pair and not an encryption. The only possibility for e_p in case (b) is $e_p \in \mathfrak{B}$: For by clause (B) of Proposition 4.4 $e_p \notin \mathrm{Cl}^{\downarrow}(\mathcal{F})$ and variables X_{MESG} or TAGS are in $\mathrm{Cl}^{\downarrow}(\mathcal{F})$ by Remark 5.3. Now by Remark 4.9 all elements of $\mathrm{Esc}(\mathcal{F}, e_p)$ other than e_p are either encryptions or atoms. Therefore all elements of $\mathrm{Esc}(\mathcal{F}, e_p)$ are either encryptions or atoms.

Let $E' = \sigma(\mathrm{Esc}(\mathcal{F}, e_p))$, $p' = \sigma(p)$ and $e'_p = \sigma(e_p)$. By the last statement of the previous paragraph, all elements of E' are encryptions or atoms. Now apply Theorem 4.11. □

When dealing with instantiating variables of sort MESG while inferring additional honest behavior, CPSA uses the notion of the set of "target terms," to keeps its behavior finite. First the set of *threshold terms* between a term a and a set of terms S:

$$\mathrm{Thr}(S, a) = \{a\} \cup \{t \mid t \text{ carries } a, t \text{ is a proper carried subterm of some } s \in S\}.$$

Note that a is always a threshold term regardless of whether a is carried within some element of a.

Definition 5.7 (Target terms)**.** *The set of target terms for a critical path p with endpoint t in a fragment $\mathcal{F}$ is the set* $\mathrm{Thr}(\mathrm{Esc}(\mathcal{F}, t), t)$.

Definition 5.8 (Critical Path Weakly Solved)**.** *Suppose $\mathcal{F}$, $\mathcal{F}'$ are fragments and p is a critical path with endpoint e_p of the term $a_{\mathcal{F}}$ at $\mathcal{F}$. Let $\sigma : \mathfrak{A} \longrightarrow \mathfrak{A}$ be an algebra homomorphism. Let $E' = \sigma(\mathrm{Esc}(\mathcal{F}, e_p))$, $p' = \sigma(p)$ and $e'_p = \sigma(e_p)$. Path p from $a_{\mathcal{F}}$ is* weakly solved *in $\mathcal{F}'$ by σ, if and only if p is solved in $\mathcal{F}'$ by σ, or if*

- *Sol5. There is an element of* $\mathrm{Targ}(\mathrm{Esc}(\mathcal{F}', e'_p), e'_p))$ *not in* $\sigma(\mathrm{Targ}(E, e_p))$.

The notion of a path being weakly solved is needed later in the proof of completeness. Sol5 represents a very weak form of progress, where we have only improved our set of target terms.

6 Protocols

A run of a protocol is viewed as an exchange of messages by a finite set of local sessions of the protocol.

An *event* is either a message transmission or a reception. If m is a message, an *outbound message event* is written as $+m$, and inbound message event is written as $-m$. By abuse of language, if e is an event we will write $-e$ to signify that e is an inbound event and similarly for outbound events. A *trace* in $\mathfrak{A}$ is a sequence $\langle e_1, \ldots, e_n \rangle$ of message events. The set of traces over $\mathfrak{A}$ is denoted $\mathfrak{C}_{\mathfrak{A}}$. A *restriction* of a trace $\tau = \langle e_1, \ldots, e_n \rangle$ is any trace $\tau' = \langle e_1, \ldots, e_k \rangle$ for $1 \leq k \leq n$. We will also use the phrase t is an *extension* of t' and the notation $\tau' = \tau|k$. Endomorphisms $\sigma \in \mathrm{End}(\mathfrak{A})$ act on message events and traces: $\sigma(\pm m) = \pm\sigma(m)$ and $\sigma\langle e_1, \ldots, e_n \rangle = \langle \sigma(e_1), \ldots, \sigma(e_n) \rangle$.

Traces of the form $\langle -m, +m \rangle$ where $m \in \mathfrak{A}$ are referred to as *listener traces*. The special trace $\langle -x, +x \rangle$ where x is a variable of sort message matches any listener trace.

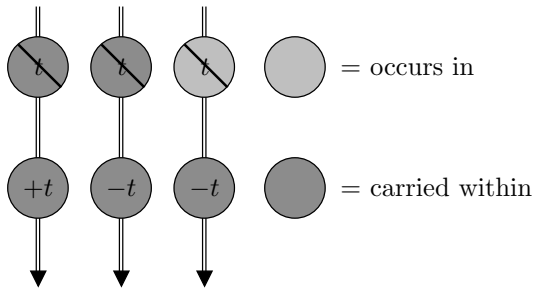


Figure 6: Originates, Gained, Acquired

Let t be a trace, m a message. m *originates* in t if it is carried by some event on t and the first event on t in which it is carried is a transmission. m is *gained* by t if it is carried by some event t and the first event on t in which it is carried is a reception; however, this condition allows for prior *occurrences* of the message m. m is *acquired* by t if it first occurs on t in a reception event and is also carried by that event.

6.1 Roles and Protocols

In a run of a protocol, the behavior of each strand is constrained, in a sense made precise below, by a composite structure called a role. A *protorole* over a cryptoalgebra $\mathfrak{A}$ with atoms $\mathfrak{B}$ is a structure $\rho = (C, N, U)$, where $C \in \mathfrak{C}_{\mathfrak{A}}$, $N \subseteq \mathfrak{B}$, and $U \subseteq \mathfrak{B}$. The trace of the role ρ is C, its non-origination assumptions are N, and its unique origination assumptions are U. In case of ambiguity, the components of ρ are denoted (C_ρ, N_ρ, U_ρ).

Definition 6.1. *A protorole $\rho = (C, N, U)$ is a* role *if*

1. *$t \in N$ implies t is not carried in C, and all variables in N occur in C.*
2. *$t \in U$ implies t originates in C.*
3. *If a variable x occurs in C then x is an atom or it is acquired in C.*

Equivalently, condition (3) *states that a variable x of sort* MESG *occurs in C only if it is acquired in C.*

A *listener trace* is any trace of the form $\langle -m, +m \rangle$ where m is of sort MESG.

Remark 6.2. Any role ρ whose trace is $\langle -x, +x \rangle$, x a variable of sort MESG, must have $N_\rho = U_\rho = \emptyset$. However if a role is of the form $\langle -m, +m \rangle$ where m is not a variable of sort MESG, then N_ρ may be non-empty. For example $\langle -\{\!|x|\!\}_u, +\{\!|x|\!\}_u \rangle$ where $u \in$ SKEY. However, it is always the case that $U_\rho = \emptyset$, since nothing originates on a listener trace.

A *listener role* is one of the form

$$lsn = (\langle -x, +x \rangle, \emptyset, \emptyset). \tag{5}$$

where x is a variable of sort MESG. This is a legitimate role: (1) and (2) are vacuous and the variable x is acquired in the trace.

We introduce a *pseudorole* $\pounds$, not a role, which is used as a special annotation for roles of auxiliary traces introduced by the CPSA search algorithm. This artifice allows us to distinguish between genuine protocol roles which may behave like listener roles and the roles of these auxiliary traces. However $\pounds$ is associated to the listener trace defined previously and accordingly we introduce the following (somewhat abusive) notation:

- $C_{\pounds}$ is the listener trace $\langle -x, +x \rangle$ where x is a variable of sort MESG.

- $U_{\pounds} = N_{\pounds} = \emptyset$.

Essentially a trace is constrained by a role if it agrees with an instance of the role restricted up to the length of the trace. The term we will actually use is role specification which completely determines the trace.

Definition 6.3. *A* role specification *for a trace τ is a structure (ρ, σ) where $\sigma \in \mathrm{End}(\mathfrak{A})$ and either (a) ρ is a role or (b) ρ is the pseudorole $\pounds$, and the following holds*

1. *τ is a restriction of $\sigma(C_\rho)$ and*
2. *$\text{s-dom}\,\sigma$ is the set of variables that occur in the trace $C_\rho \,|\, \text{len}\,\Theta(s)\tau$.*

Remark 6.4. Condition (2) means that the substitution σ transforms precisely those variables that occur within the role up to the height of the strand.

Note that the definition of role specification (ρ, σ) for a trace τ does not involve origination assumptions. However, the origination assumptions of ρ are transported onto τ by the substitution σ. First some notation. Given a $\sigma \in \mathrm{End}(\mathfrak{A})$ and $E \subseteq \mathfrak{A}$, $[\sigma]_* E$ is the set

$$\{\sigma(t) : t \in E, \mathrm{Vars}(t) \subseteq \text{s-dom}(\sigma)\}.$$

Definition 6.5. *Suppose τ is a trace and (ρ, σ) is a role specification for τ. The* origination assumptions inherited by τ via (ρ, σ) *are*

$$N_{(\rho,\sigma)} = [\sigma]_* N_\rho,$$

and

$$U_{(\rho,\sigma)} = \sigma U_s,$$

where U_s are the elements of U_ρ that originate on before the event at position $\text{len}\,\tau$ *on the role strand C_ρ.*

Remark 6.6. Since $N_{\pounds} = U_{\pounds} = \emptyset$, there are no inherited origination assumptions for strands with specification $\pounds$.

A *protocol* is a set of roles. Let $\mathrm{Vars}(P)$ be the set of variables that occur in the traces of the roles in protocol P.

7 Skeletons

Fix a protocol P for the remainder of the paper. The details of penetrator behavior are abstracted away when performing protocol analysis. The abstracted description is called a *realized skeleton*. To define this and as an essential tool of analysis we introduce structures with increasing specificity. In the following $\mathfrak{A}$ denotes a cryptoalgebra.

We begin with a general notion which formalizes a notion of collection of "locally linearly ordered" communication events. A *node space* is a pair (I, Θ) where Θ is a map from I to the set of traces over $\mathfrak{A}$. Associated to a node space $\mathbb{A} = (I, \Theta)$ are the following objects: The *nodes* of $\mathbb{A}$ is the set of pairs (s, i) where $s \in I$ and $i \leq \operatorname{len} \Theta(s)$. By abuse of notation we write $(s, i) \in \mathbb{A}$ to indicate that (s, i) is a node of $\mathbb{A}$. The elements of the index set I are the *strands* of the node space. Nodes (s, i) and (s', j) are on the same strand if $s = s'$. The set of variables occurring in the traces of $\mathbb{A}$ is denoted $\operatorname{Vars}(\mathbb{A})$. If the variables in $\operatorname{Vars}(\mathbb{A})$ are all of base sort, the node space is *instantiated*. For any $t \in \mathfrak{A}$, $\mathcal{O}_{\mathbb{A}}(t)$ is the set of nodes at which t originates in $\mathbb{A}$, $\mathcal{G}_{\mathbb{A}}(t)$ is the set of nodes at which t is gained in $\mathbb{A}$, and $\mathcal{C}_{\mathbb{A}}(t)$ is the set of nodes at which t is carried in $\mathbb{A}$. A *strand* is a node space consisting of a single strand.

There is no intrinsic association between a node space $\mathbb{A}$ and a protocol P. However, we can establish such an association by requiring that each strand of $\mathbb{A}$ be constrained by some specified role. Before stating this condition, we state the blanket *variable hygiene condition* for protoskeletons and protocols which we will assume throughout: $\operatorname{Vars}(P) \cup \operatorname{Vars}(lsn)$ is disjoint from $\operatorname{Vars}(\Theta)$.

A *P-role assignment* for a node space $\mathbb{A}$ is a mapping $\mathcal{A}$ which associates to each strand $s \in I$ a P-role specification $\mathcal{A}(s) = (\rho_s, \sigma_s)$.

Remark 7.1. By the variable hygiene requirement, $\operatorname{Vars}(P)$ and $\operatorname{Vars}(\Theta)$ are disjoint. The substitution σ_s must affect every variable that occurs within the role up to the height of the strand.

A node space may have no P-role assignments or it may have more than one role assignment.

Given a node space $\mathbb{A}$ and a role assignment $\mathcal{A}$ for $\mathbb{A}$, the *origination assumptions inherited by $\mathbb{A}$ from P via $\mathcal{A}$* are

$$N_{\mathcal{A}} = \bigcup_{s \in I} N_{\mathcal{A}(s)}, \quad U_{\mathcal{A}} = \bigcup_{s \in I} U_{\mathcal{A}(s)}.$$

The sets N and U are the *declared* non-origination terms and uniquely originating terms of $\mathbb{A}$.

Definition 7.2. *Suppose P is a protocol. A* protoskeleton *for P is a structure $\mathbb{A} = (I, \Theta, \prec, N, U)$, where (I, Θ) is a node space, $N \subseteq \mathfrak{B}$, $U \subseteq \mathfrak{B}$, and $\prec$ is a relation on the nodes of $\mathbb{A}$ such that there is a role assignment $\mathcal{A}$ with the following properties:*

1. *$N_{\mathcal{A}} \subseteq N$.*
2. *$U_{\mathcal{A}} \subseteq U$.*

The role assignment $\mathcal{A}$ in Definition 7.2 is a protoskeleton validating role assignment. *In contexts where the protocol is understood, we simply refer to a protoskeleton for P as a protoskeleton. If $\mathbb{A}$ is a protoskeleton, $I_{\mathbb{A}}$,$\Theta_{\mathbb{A}}$, $\prec_{\mathbb{A}}$, $N_{\mathbb{A}}$, $U_{\mathbb{A}}$ are the components of $\mathbb{A}$.*

Protoskeleton validating role assignments for a structure $\mathbb{A} = (I, \Theta, \prec, N, U)$ can be mixed:

Remark 7.3. Note that in general, a P-role specification (ρ, σ) for a strand need not respect points of origination, that is if t originates at position i on $C_\rho|n$, $\sigma(t)$ may originate at some earlier node on $\sigma(C_\rho)|n$. However, condition (5) of Definition 7.2 states that a protoskeleton must have at least one P-role assignment such that all role specifications are origination preserving.

Definition 7.4 (Protomorphism)**.** *Let $\mathbb{A}$, $\mathbb{B}$ be protoskeletons. A* protomorphism *from $\mathbb{A}$ to $\mathbb{B}$ is a pair $\lambda = (\varphi, \sigma)$ where φ maps nodes of $\mathbb{A}$ to nodes of $\mathbb{B}$ and $\sigma \in \mathrm{End}(\mathfrak{A})$ is such that:*

1. *There exists a map $\varphi_{\mathrm{str}} : I_{\mathbb{A}} \to I_{\mathbb{B}}$ such that for all $(s, i) \in \mathbb{A}$, $\varphi(s, i) = (\varphi_{\mathrm{str}}(s), i)$.*
2. *$n \in nodes(\mathbb{A})$ implies $\sigma(evt(n)) = evt(\varphi(n))$.*
3. *$\sigma(N_{\mathbb{A}}) \subseteq N_{\mathbb{B}}$;*
4. *$\sigma(U_{\mathbb{A}}) \subseteq U_{\mathbb{B}}$*

Remark 7.5. The mapping φ_{str} is called the *strand mapping*. The strand mapping is unique. By abuse of notation, in most contexts we will use φ to denote the strand mapping.

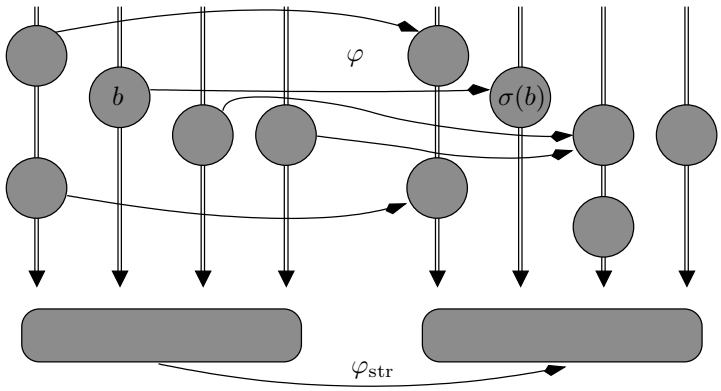


Figure 7: Protomorphism of Protoskeletons

We write $\mathbb{A} \overset{\lambda}{\dashrightarrow} \mathbb{B}$ when λ is protomorphism.

Definition 7.6 (Structure-preserving)**.** *A protomorphism* $\mathbb{A} \overset{\varphi,\sigma}{\dashrightarrow} \mathbb{B}$ *is* structure-preserving *if* $n_0 \prec_{\mathbb{A}} n_1$ *implies* $\varphi(n_0) \prec_{\mathbb{B}} \varphi(n_1)$*.*

Clearly the composition of structure-preserving protomorphisms is a structure-preserving protomorphism.

Definition 7.7 (Point of origination preserving morphisms)**.** *A protomorphism* $\mathbb{A} \overset{\varphi,\sigma}{\dashrightarrow} \mathbb{B}$ preserves points of origination *if* $\varphi(\mathcal{O}_{\mathbb{A}}(t)) \subseteq \mathcal{O}_{\mathbb{B}}(\sigma(t))$ *for all* $t \in U_{\mathbb{A}}$*.*

Definition 7.8 (Homomorphism)**.** *A protomorphism* $\mathbb{A} \overset{\varphi,\sigma}{\dashrightarrow} \mathbb{B}$ *is a* homomorphism *if it is structure-preserving and preserves points of origination.*

Definition 7.8 allows the image of an atom in $U_{\mathbb{A}}$ to originate at more than one point.

We write $\mathbb{A} \overset{\lambda}{\rightarrow} \mathbb{B}$ when λ is a homomorphism. We use $\text{Protom}(\mathbb{A}, \mathbb{B})$ to denote the set of all protomorphisms from $\mathbb{A}$ to $\mathbb{B}$, and $\text{Hom}(\mathbb{A}, \mathbb{B})$ to denote the set of all homomorphisms from $\mathbb{A}$ to $\mathbb{B}$.

Proposition 7.9. *The composition of homomorphisms between protoskeletons is a homomorphism.*

Proof. Suppose $\mathbb{A} \xrightarrow{(\varphi_0,\sigma_0)} \mathbb{B}, \mathbb{B} \xrightarrow{(\varphi_1,\sigma_1)} \mathbb{C}$ are homomorphisms. For each $t \in U_{\mathbb{A}}$, $\sigma_0(t) \in U_{\mathbb{B}}$ by Property (4) of protomorphisms and thus by the definition of homomorphism,

$$\varphi_1(\varphi_0(\mathcal{O}_{\mathbb{A}}(t))) \subseteq \varphi_1(\mathcal{O}_{\mathbb{B}}(\sigma_0(t))) \subseteq \mathcal{O}_{\mathbb{C}}\sigma_1(\sigma_0(t))$$

□

Given a protoskeleton $\mathbb{A} = (I, \Theta, \prec, N, U)$ and a receiving node n of $\mathbb{A}$, we can define a fragment $\mathcal{F}_{\mathbb{A},n} = (T, U, N, a)$ where $-a = evt(n)$ and where

$$T = \{t \mid \exists n' \in \mathbb{A}, n' \prec n \text{ and } evt(n') = +t\}.$$

We write $P_{\mathbb{A},n}$ to refer to $P_{\mathcal{F}_{\mathbb{A},n}}$.

Protoskeletons and protomorphisms form a category as well as protoskeletons and homomorphisms.

7.1 Preskeletons

We now consider two additional subcategories of the protoskeleton categories. These categories will be full subcategories. Accordingly, when we refer to a protomorphism or homomorphism we will always regard it as a pair (φ, σ).

Definition 7.10. *A protoskeleton $\mathbb{A} = (I, \Theta, \prec, N, U)$ for a protocol P is a* preskeleton *if*

1. *Relation $\prec$ is transitive, asymmetric, and includes the strand succession relation: $(s, i) \Rightarrow (s, i + 1)$ for all $(s, i + 1) \in \mathbb{A}$.*
2. *If $s \neq s'$ and $(s, i) \prec (s', j)$ then either $evt(s, i) = +e$ and $evt(s', j) = -e'$, or there exists a node n such that $(s, i) \prec n \prec (s', j)$.*
3. *Each atom in N is carried at no node of $\mathbb{A}$, and each variable in the atom occurs at some node of $\mathbb{A}$.*
4. *Each atom in U is carried at some node of $\mathbb{A}$.*
5. *If s is any strand such that $\rho_s \in P$, and $t \in U_{\rho_s}$ such that t originates in $C_{\rho_s}|\operatorname{len}\Theta(s)$ at event i, then $\sigma(t)$ originates in $\Theta(s)$ at (s, i).*

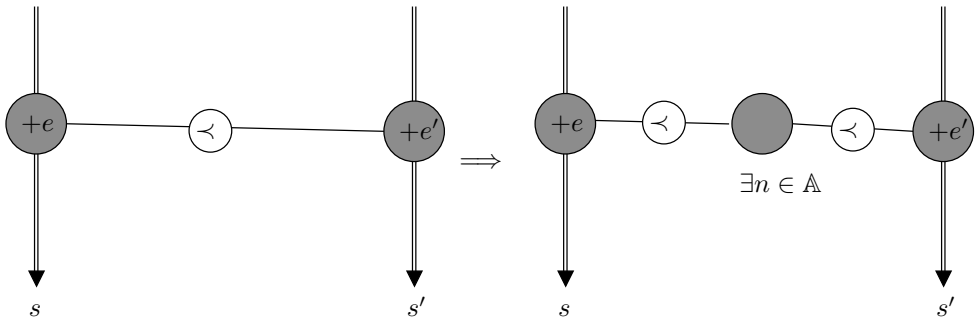


Figure 8: One Case of Intermediate Node Condition: (2) of Definition 7.10. Two remaining cases: $-e, -e'$ and $-e, +e'$.

	Protoskeleton	Preskeleton	Skeleton
Has valid role assignment	•	•	•
Inherited origination nodes	•	•	•
$\prec$ is a strict partial order		•	•
Origination assumptions satisfied			•

Figure 9: Salient differences between Proto, Pre and Proper Skeletons

Definition 7.11. *A preskeleton* $\mathbb{A} = (I, \Theta, \prec, N, U)$ *for* P *is a* skeleton *for* P *if each atom in* U *originates on exactly one strand, and the node of origination precedes each other node that carries the atom, i.e. for every* $t \in U$, $n_0 \in \mathcal{O}_{\mathbb{A}}(t)$ *and* $n_1 \in \mathcal{C}_{\mathbb{A}}(t), n_1 \neq n_0$ *implies* $n_0 \prec n_1$.

In contexts where the protocol is understood, we simply refer to a skeleton for P as a skeleton.

We use PSkel(P), PreSkel(P) and Skel(P) to denote the collections of valid protoskeletons for P, valid preskeletons for P, and valid skeletons for P, respectively. Since the protocol P is fixed for the remainder of the paper, the argument P will be mostly omitted.

7.2 Hierarchies and Commitments

In this section we fix a protocol P. To analyse the CPSA algorithm, we need to introduce additional structure on the basic protoskeleton category.

A *protoskeleton refinement category* is a category $\mathcal{C}$ and a functor $\mathbf{F} : \mathcal{C} \longrightarrow$ PSkel.

7.2.1 Assignment Committed Protoskeletons

A pair $(\mathbb{A}, \mathcal{A})$ consisting of a protoskeleton $\mathbb{A}$ and a P-role assignment $\mathcal{A}$ is an *assignment committed* protoskeleton. To make this class of objects into a category, we first consider the behavior of role assignments under morphisms. There is no universally applicable way that role assignments can be regarded as transforming either covariantly, that is pushing the assignments forward under morphisms or contravariantly, pulling them back. We can however regard them as transforming covariantly in a weaker sense. Suppose $\mathbb{A}$, $\mathbb{B}$ are node spaces, $\lambda = (\varphi, \sigma) : \mathbb{A} \longrightarrow \mathbb{B}$ is a homomorphism of node spaces. If

$$\mathcal{A} = \{(\rho_s^{\mathcal{A}}, \sigma_s^{\mathcal{A}}) : s \in I_{\mathbb{A}}\},$$
$$\mathcal{B} = \{(\rho_t^{\mathcal{B}}, \sigma_t^{\mathcal{B}}) : t \in I_{\mathbb{B}}\}$$

are role assignments for $\mathbb{A}$, $\mathbb{B}$ respectively such that for all $s \in I_{\mathbb{A}}$, $\rho^{\mathcal{B}}_{\varphi(s)} = \rho^{\mathcal{A}}_s$ then *the role assignment $\mathcal{B}$ is a pushforward of $\mathcal{A}$ under λ.* Thus instead of being a function, "pushforward" is a relation between role assignments.

Remark 7.12. The definition of pushforward under $\lambda = (\varphi, \sigma)$ says nothing about the substitutions $\sigma^{\mathcal{A}}$ and $\sigma^{\mathcal{B}}$. However in the basic cryptoalgebra,

$$\sigma \circ \sigma^{\mathcal{A}}(v) = \sigma^{\mathcal{B}}(v) \tag{6}$$

for all variables v which occur in $C_{\rho_s} | \operatorname{len} \Theta(s)$.

Definition 7.13 (Assignment-preserving)**.** *A protomorphism $\mathbb{A} \overset{\varphi,\sigma}{\dashrightarrow} \mathbb{B}$ is a* protomorphism of assignment-committed protoskeletons $(\mathbb{A}, \mathcal{A}_{\mathbb{A}}) \longrightarrow (\mathbb{B}, \mathcal{A}_{\mathbb{B}})$ *if the role assignment $\mathcal{A}_{\mathbb{B}}$ is a pushforward of the role assignment $\mathcal{A}_{\mathbb{A}}$ under (φ, σ).*

Remark 7.14. It follows immediately from the definitions that

$$\mathbf{F}_{\bullet\rightarrow} : (\mathbb{A}, \mathcal{A}) \mapsto \mathbb{A}$$

maps the class of assignment committed protoskeletons *onto* the class of protoskeletons. Any assignment-preserving protomorphism is a protomorphism, so $\mathbf{F}_{\bullet\rightarrow}$ can be considered a functor from the category of assignment-committed protoskeletons to protoskeletons.

We denote the category of assignment-committed protoskeletons by PSkel.

7.2.2 Listener Committed Protoskeletons

Remark 7.15. The definition of protocol P allows for roles ρ such that the trace C_ρ is a listener strand. However, to conform to the established usage in the CPSA specification we will use the term listener to refer to a strand which is specified by the listener pseudorole. Accordingly the *listener set* of a role assignment $\mathcal{A}$ is

$$L(\mathcal{A}) = \{s \in I_{\mathbb{A}} : \exists \sigma\ \mathcal{A}(s) = (\pounds, \sigma)\}. \tag{7}$$

By extension, we will also call this set the listeners of an assignment committed protoskeleton $(\mathbb{A}, \mathcal{A})$. Let $\mathbb{A}$ be a protoskeleton for protocol P. A *valid set of listeners* for $\mathbb{A}$ is a set of strands $L \subseteq I_{\mathbb{A}}$ of the form $L(\mathcal{A})$ for some role assignment $\mathcal{A}$. The protoskeleton $\mathbb{A}$ may contain listener strands not in $L(\mathcal{A})$ if the protocol P has listener roles.

A *listener committed protoskeleton* is a pair $(\mathbb{A}, L_{\mathbb{A}})$ where $L_{\mathbb{A}}$ is a valid set of listeners for $\mathbb{A}$.

Definition 7.16 (Listener-respecting)**.** $\mathbb{A} \overset{\varphi,\sigma}{\dashrightarrow} \mathbb{B}$ *is a* protomorphism of listener-committed protoskeletons from $(\mathbb{A}, L_{\mathbb{A}})$ to $(\mathbb{B}, L_{\mathbb{B}})$ *if for every strand* $s \in \mathbb{A}$, $s \notin L_{\mathbb{A}}$ *implies* $\varphi(s) \notin L_{\mathbb{B}}$.

We write $\mathbb{A}^\circ \overset{\varphi,\sigma}{\dashrightarrow} \mathbb{B}^\circ$ and $\mathbb{A}^\circ \xrightarrow{\varphi,\sigma} \mathbb{B}^\circ$ to denote that (φ, σ) is a protomorphism (respectively homomorphism) of listener-committed protoskeletons from $\mathbb{A}^\circ$ to $\mathbb{B}^\circ$.

Remark 7.17. Similarly, there is a well-defined functor $\mathbf{F}_{\circ\rightarrow}$ from listener committed protoskeletons to protoskeletons which simply forgets the listener set. Clearly

$$\mathbf{F}_{\bullet\rightarrow} = \mathbf{F}_{\circ\rightarrow} \circ \mathbf{F}_{\bullet\rightarrow\circ}. \tag{8}$$

Since $\mathbf{F}_{\bullet\rightarrow}$ is surjective (Remark 7.14) $\mathbf{F}_{\circ\rightarrow}$ is surjective.

Remark 7.18. It follows immediately from the definitions that

$$\mathbf{F}_{\bullet\rightarrow\circ} : (\mathbb{A}, \mathcal{A}) \mapsto (\mathbb{A}, L(\mathcal{A}))$$

maps the class of assignment committed protoskeletons *onto* the class of listener committed protoskeletons. If (φ, σ) is a protomorphism of assignment-committed protoskeletons, then $(\varphi, \sigma)$is also a protomorphism of the corresponding listener-committed protoskeletons. Thus we can consider $\mathbf{F}_{\bullet\rightarrow\circ}$ as a functor.

Definition 7.19. *A protomorphism* $\mathbb{A}^\circ \overset{\lambda}{\dashrightarrow} \mathbb{B}^\circ$ *is* assignment-consistent *if there exist role assignments* $\mathbb{A}^\bullet = (\mathbb{A}, \mathcal{A}_\mathbb{A}), \mathbb{B}^\bullet = (\mathbb{B}, \mathcal{A}_\mathbb{B})$ *such that (1)* $\mathbb{A}^\circ$ *is the listener-committed version of* $\mathbb{A}$ *determined by* $\mathbb{A}^\bullet$*, (2)* $\mathbb{B}^\circ$ *is the listener-committed version of* $\mathbb{B}$ *determined by* $\mathbb{B}^\bullet$*, and* λ *is a protomorphism of assignment-committed protoskeletons from* $\mathbb{A}^\bullet$ *to* $\mathbb{B}^\bullet$*.*

Remark 7.20. $\mathbb{A}^\circ \overset{\lambda}{\dashrightarrow} \mathbb{B}^\circ$ is assignment-consistent if and only if it is of the form $\mathbf{F}_{\bullet\to\circ}(\lambda^\bullet)$ for some $\mathbb{A}^\bullet \overset{\lambda^\bullet}{\dashrightarrow} \mathbb{B}^\bullet$.

7.3 Diagrams

We use $\mathrm{PSkel}^\bullet(P)$, $\mathrm{PreSkel}^\bullet(P)$ and $\mathrm{Skel}^\bullet(P)$ to denote the collections of assignment committed protoskeletons for P, assignment committed preskeletons for P, and assignment committed skeletons for P, respectively. $\mathrm{PSkel}^\circ(P)$, $\mathrm{PreSkel}^\circ(P)$ and $\mathrm{Skel}^\circ(P)$ denote the collections of listener committed protoskeletons for P, listener committed preskeletons for P, and listener committed skeletons for P, respectively.

Remark 7.21. As a notational heuristic, we use $\mathbb{A}^\bullet, \mathbb{B}^\bullet$ and $\mathbb{A}^\circ, \mathbb{B}^\circ$ as symbols to denote assignment committed and listener committed skeletons (possibly proto or pre). Morever, unless explicitly stated to the contrary, *in a context in which either one of the symbols* $\mathbb{A}^\bullet$ *or* $\mathbb{A}^\circ$ *are mentioned, the symbol* $\mathbb{A}$ *refers to the underlying protoskeleton, and, in a context in which* $\mathbb{A}^\bullet$ *is mentioned,* $\mathbb{A}^\circ$ *refers to the listener-restriction of* $\mathbb{A}^\bullet$*.*

Let P be a protocol. We have the following diagram:

$$\begin{array}{ccccc}
\mathrm{PSkel}^\bullet & \Longleftarrow & \mathrm{PreSkel}^\bullet & \Longleftarrow & \mathrm{Skel}^\bullet \\
\Big\downarrow{\scriptstyle \mathbf{F}_{\bullet\to\circ}} & & \Big\downarrow & & \Big\downarrow \\
\mathrm{PSkel}^\circ & \Longleftarrow & \mathrm{PreSkel}^\circ & \Longleftarrow & \mathrm{Skel}^\circ \\
\Big\downarrow{\scriptstyle \mathbf{F}_{\circ\to}} & & \Big\downarrow & & \Big\downarrow \\
\mathrm{PSkel} & \Longleftarrow & \mathrm{PreSkel} & \Longleftarrow & \mathrm{Skel}
\end{array} \tag{9}$$

where the downward arrows are structure removing mappings and leftward arrows are inclusion mappings. At this point of the exposition nothing can be

said about functoriality of the arrows since no morphisms have been defined for any of these collections. That is the object of the next section.

The preceding remarks are summarized in the following proposition:

Proposition 7.22. *In the diagram* (9)*, the mappings* $\mathbf{F}_{\bullet\to\circ}$ *and* $\mathbf{F}_{\circ\to}$ *are defined and surjective on all the columns.*

7.4 Preservation Properties

In arguments involving objects such as protoskeletons and protomorphisms, it is desirable to identify properties of these objects which are preserved under some specific transformation or more generally some relation. Some of the properties we single out are in fact negations of conditions used to define the main categories in the theory.

Non-asymmetry of protoskeleton $\mathbb{A}$ is the property that there are nodes m, n for which $m \prec n$ and $n \prec m$. A non-asymetric protoskeleton is not a skeleton.

Proposition 7.23 (Non-asymmetry preserved under structure-preserving protomorphisms)**.** *If* $\mathbb{A}$ *is a protoskeleton such that* $\prec_{\mathbb{A}}$ *is not asymmetric, then there is no structure-preserving protomorphism* λ *from* $\mathbb{A}$ *to a protoskeleton* $\mathbb{B}$*.*

Proof. If $m \prec n$ and $n \prec m$ in $\mathbb{A}$, and $\lambda = (\varphi, \sigma)$ is structure-preserving, then $\varphi(m) \prec \varphi(n)$ and $\varphi(n) \prec \varphi(m)$ in $\mathbb{B}$, so $\mathbb{B}$ is not a preskeleton. Note that even if $\varphi(m) = \varphi(n)$, $\prec$ is not asymmetric in $\mathbb{B}$. □

Proposition 7.24 (Point of origination non-preservation preserved under extensions)**.** *Suppose* $\mathbb{A} \overset{\varphi,\sigma}{\dashrightarrow} \mathbb{B}$ *and* $t \in U_{\mathbb{A}}$ *are such that* $n \in \mathcal{O}_{\mathbb{A}}(t)$*, but* $\varphi(n) \notin \mathcal{O}_{\mathbb{B}}(\sigma(t))$*. If* $\mathbb{B} \overset{\varphi',\sigma'}{\dashrightarrow} \mathbb{C}$ *is any protomorphism to any protoskeleton, then* $(\varphi' \circ \varphi)(n) \notin \mathcal{O}_{\mathbb{C}}((\sigma' \circ \sigma)(t))$*.*

Proof. Note that the message at n carries t in $\mathbb{A}$, so the message at $\varphi(n)$ carries $\sigma(t)$ in $\mathbb{B}$. If $\varphi(n)$ is not a point of origination of $\sigma(t)$ in $\mathbb{B}$, there must be a node $n' = (s, i)$ where $n = (s, j)$ and $i < j$, where $\varphi(n')$ carries $\sigma(t)$. Then $\varphi'(\varphi(n'))$ carries $\sigma'(\sigma(t))$, so $\varphi'(\varphi(n)) \notin \mathcal{O}_{\mathbb{C}}(\sigma'(\sigma(t)))$. □

Proposition 7.25 (Point of origination preservation preserved under protomorphism factoring)**.** *Suppose we have* $\mathbb{A} \overset{\varphi,\sigma}{\dashrightarrow} \mathbb{B}$ *and we have* $\mathbb{A} \overset{\varphi',\sigma'}{\dashrightarrow} \mathbb{C}$ *where*

there is some $\mathbb{B} \overset{\varphi'',\sigma''}{\dashrightarrow} \mathbb{C}$ *such that* $\varphi = \varphi'' \circ \varphi'$ *and* $\sigma = \sigma'' \circ \sigma'$*. Then if* (φ, σ) *preserves points of origination,* (φ', σ') *preserves points of origination.*

Proof. We prove the contrapositive: that if (φ', σ') does not preserve points of origination then neither does (φ, σ). Suppose (φ', σ') does not preserve points of origination, and suppose $t \in U_{\mathbb{A}}$ and suppose $n \in \mathcal{O}_{\mathbb{A}}(t)$ are such that $\varphi'(n) \notin \mathcal{O}_{\mathbb{C}}(\sigma'(t))$. By Proposition 7.24, $\varphi''(\varphi'(n)) = \varphi(n) \notin \mathcal{O}_{\mathbb{B}}(\sigma''(\sigma'(t))) = \mathcal{O}_{\mathbb{B}}(\sigma(t))$. Thus, (φ, σ) does not preserve points of origination. □

Proposition 7.26 (Violation of role-inherited unique origination constraints preserved under protomorphisms of assignment-committed protoskeletons)**.** *Suppose* $\mathbb{A}^\bullet$ *is a protoskeleton,* $\mathbb{B}^\bullet$ *is a protoskeleton meeting condition* (5) *of the definition of preskeleton and* λ *is a protomorphism of assignment-committed protoskeletons from* $\mathbb{A}^\bullet$ *to* $\mathbb{B}^\bullet$*. Then* $\mathbb{A}^\bullet$ *meets condition* (5) *of the definition of preskeleton.*

Proof. Argue by contradiction. Suppose $\mathbb{A}^\bullet = (\mathbb{A}, \mathcal{A})$ does not meet condition (5) of the definition of preskeleton. Then there is a strand s and a $t \in U_{\rho_s}$ such that t originates in $C_{\rho_s}|\operatorname{len}\Theta_{\mathbb{A}}(s)$ at event i but $\sigma_s(t)$ does not originate in $\Theta_{\mathbb{A}}(s)$ at event i. Thus $\sigma_s(t)$ is carried at an earlier event j on $\Theta_{\mathbb{A}}(s)$. But then $\sigma(\sigma_s(t))$ is carried at event j in strand $\varphi(s)$ in $\mathbb{B}^\bullet$. However, strand $\varphi(s)$ is associated in $\mathbb{B}^\bullet$ with ρ_s and thus $\mathbb{B}^\bullet$ does not meet condition (5) of the definition of preskeleton. □

7.5 Coverage

If $\mathbb{A} = (I, \Theta, \prec, N, U)$ is a protoskeleton and $I' \subset I$ is any subset, then

$$\operatorname{Rmv}_{I'}(\mathbb{A}) = (I \setminus I', \Theta|_{I \setminus I'}, \prec |_{(I \setminus I') \times (I \setminus I')}, N, U).$$

We are particularly interested in the case where I' is a valid listener set for $\mathbb{A}$. In that case, $\operatorname{Rmv}_{I'}(\mathbb{A})$ is a preskeleton (or skeleton) if $\mathbb{A}$ is a preskeleton (or skeleton). If $\mathbb{A}^\circ = (\mathbb{A}, L_{\mathbb{A}})$ is a listener committed protoskeleton, then

$$\operatorname{Rmv} \mathbb{A}^\circ = \operatorname{Rmv}_{L_{\mathbb{A}}}(\mathbb{A}). \tag{10}$$

Remark 7.27. By definition Rmv maps objects in PSkel$^\circ$ to objects in PSkel. If $\mathbb{A}^\circ \overset{\lambda}{\dashrightarrow} \mathbb{B}^\circ$ is a protomorphism of listener-committed skeletons then by Definition 7.16, $\lambda\,|_{\operatorname{Rmv}\mathbb{A}^\circ}$ is a protomorphism $\operatorname{Rmv}\mathbb{A}^\circ \dashrightarrow \operatorname{Rmv}\mathbb{B}^\circ$. Thus

we can view Rmv as a functor in the protomorphism categories $\mathrm{PSkel}^\circ \longrightarrow \mathrm{PSkel}$. We will denote $\lambda\,|_{\mathrm{Rmv}\,\mathbb{A}^\circ}$ by $\mathrm{Rmv}(\lambda)$.

If $\mathbb{A}^\circ \xrightarrow{\lambda} \mathbb{B}^\circ$ is a homomorphism $\mathrm{Rmv}\,\mathbb{A}^\circ \to \mathrm{Rmv}\,\mathbb{B}^\circ$ is a homomorphism. Proof: The order preserving property is obvious. Nothing can originate on a listener strand (Remark 6.2), therefore deleting any set of listener nodes does not affect origination nodes.

Remark 7.28. If $\mathbb{A}^\circ$ is a realized skeleton then $\mathrm{Rmv}(\mathbb{A}^\circ)$ is also realized. Proof: Removing listener strands on a preskeleton has no effect on the fragments at any of the remaining nodes.

Definition 7.29. *The* coverage *of a listener committed skeleton $\mathbb{A}^\circ$, which we denote $[\![\mathbb{A}^\circ]\!]$, is the collection of homomorphisms $\lambda\,|_{\mathrm{Rmv}(\mathbb{A}^\circ)}$ as λ ranges over homomorphisms $\mathbb{A}^\circ \to \mathbb{B}^\circ$ with $\mathbb{B}$ realized.*

Alternatively,

$$[\![\mathbb{A}^\circ]\!] = \{(\mathrm{Rmv}\,\mathbb{B}^\circ, \lambda\,|_{\mathrm{Rmv}(\mathbb{A}^\circ)}) \mid \mathbb{B} \text{ is realized and } \mathbb{A}^\circ \xrightarrow{\lambda} \mathbb{B}^\circ\}. \qquad (11)$$

Thus the coverage is some collection of homomorphisms $\mathrm{Rmv}(\mathbb{A}^\circ) \to \mathbb{B}$ into realized skeletons $\mathbb{B}$.

8 Operators

In this section, we define the notion of an *operator*, which transforms protoskeletons. We will then define the set of operators that CPSA most depends on. In general, an operator on a collection S is a mapping F which whose domain consists pairs (a, τ) where $a \in S$ and τ is an auxiliary parameter.

An *operator* for a protocol P is a self-mapping on $\mathrm{PSkel}(P)$. An *assignment-transforming* operator is a partial self-mapping on $\mathrm{PSkel}^\bullet(P)$. A *listener-transforming* operator is a partial self-mapping on $\mathrm{PSkel}^\circ(P)$. We will also use the generic term "operator" informally to refer to an operator on any of the protoskeleton categories and as a self-mapping on the category of node spaces.

In the assignment-transforming operators f we consider below, the protoskeleton component of $\mathsf{f}(\mathbb{A}, \mathcal{A})$ depends only on $\mathbb{A}$ and the role assignment component depends only on $\mathcal{A}$. We will call the role assignment component the *role-assignment transformation*. A desirable property of the role-assignment transformation is that the set of listeners in $\mathsf{f}(\mathbb{A}, \mathcal{A})$ depends only on the set of listener strands in $\mathcal{A}$.

Definition 8.1. *An assignment-transforming operator* f *is* well-behaved with respect to listeners *if whenever* $\mathcal{A}$ *and* $\mathcal{A}'$ *are role assignments for a protoskeleton* $\mathbb{A}$ *which have the same set of pseudolisteners, the role assignments of* $\mathsf{f}(\mathbb{A}, \mathcal{A})$ *and* $\mathsf{f}(\mathbb{A}, \mathcal{A}')$ *also have the same set of pseudolisteners.*

Whenever f is an assignment-transforming operator well-behaved with respect to listeners, we can view f as describing a well-defined listener-transforming operator, which we will also refer to as f, abusing notation. Specifically, if $L_{\mathbb{A}}$ is a valid listener set for $\mathbb{A}$ then let $\mathcal{A}$ be a role assignment justifying its validity. Since the set of listeners under $\mathsf{f}(\mathcal{A})$ depends only on $L_{\mathbb{A}}$, we can view that set as the well-defined result of applying f to $L_{\mathbb{A}}$.

A *linking map* Λ_{f} associated to an operator f (on any of the protoskeleton categories) associates to any protoskeleton $\mathbb{A}$ a protomorphism $\Lambda_{\mathsf{f}}(\mathbb{A}) = (\varphi_{\mathsf{f}}(\mathbb{A}), \sigma_{\mathsf{f}}(\mathbb{A}))$ from $\mathbb{A}$ to $\mathsf{f}(\mathbb{A})$. For each protoskeleton $\mathbb{A}$, $\Lambda_{\mathsf{f}}(\mathbb{A})$ is called the *linking protomorphism.* In particular, if the operator f acts on the protoskeleton category $\mathrm{PSkel}^{\bullet}$, for each $\mathbb{A}^{\bullet} \in \mathrm{PSkel}^{\bullet}$, the linking protomorphism will be required to be an assignment-preserving protomorphism.

Remark 8.2. Suppose $(\mathbb{A}, \mathcal{A})$ is an assignment-committed protoskeleton with linking protomorphism $\Lambda_{\mathsf{f}}(\mathbb{A}) = (\varphi, \sigma)$. $\Lambda_{\mathsf{f}}(\mathbb{A})$ is assignment-preserving if and only if the role assignment of $\mathcal{B}$ of $\mathsf{f}(\mathbb{A}, \mathcal{A})$ satisfies $\rho^{\mathcal{B}}_{\varphi(s)} = \rho^{\mathcal{A}}_{s}$ for every strand s in $\mathbb{A}$. In particular $\rho^{\mathcal{B}}_{\varphi(s)}$ is a pseudolistener if and only if $\rho^{\mathcal{A}}_{s}$ is a pseudolistener. This remark proves:

Proposition 8.3. *Suppose* f *is an assignment transforming operator with linking protomorphism* Λ_{f}*. If for every assignment committed protoskeleton* $(\mathbb{A}, \mathcal{A})$ *the node mapping component of* $\Lambda_{\mathsf{f}}(\mathbb{A}, \mathcal{A})$ *is surjective, then* f *is well-behaved with respect to listeners.*

CPSA operates at the listener-committed protoskeleton level, so it is important that:

- Operators we use are listener-transforming, and
- Operators we use have linking protomorphisms that are protomorphisms of listener-committed protoskeletons from $\mathbb{A}^{\circ}$ to $\mathsf{f}(\mathbb{A}^{\circ})$.

Most of the operators we use are actually assignment-transforming operators well-behaved with respect to listeners.

8.1 Suites

A *suite* is a map from Pskel° to a set of listener-transforming operators, that is

$$\mathfrak{f} : \text{Pskel}^\circ \longrightarrow \mathcal{P}(\{\mathsf{f} | \mathsf{f} \text{ is a listener-transforming operator}\})$$

We use $\mathfrak{f}[\mathbb{A}^\circ]$ to denote $\{\mathsf{f}(\mathbb{A}^\circ) | \mathsf{f} \in \mathfrak{f}(\mathbb{A}^\circ)\}$.

If $\mathfrak{f}$ and $\mathfrak{g}$ are suites, then $\mathfrak{f} \circ \mathfrak{g}$ is a suite where $\mathfrak{f} \circ \mathfrak{g}(\mathbb{A}^\circ) = \{\mathsf{f} \circ \mathsf{g} | \mathsf{g} \in \mathfrak{g}(\mathbb{A}^\circ), \mathsf{f} \in \mathfrak{f}(\mathsf{g}(\mathbb{A}^\circ))\}$.

At the top level, CPSA is a setwise term reduction system, driven by reductions $\mathfrak{f}[\cdot]$ for various suites.

8.2 Filters

A *filter* is a predicate on pairs $(\mathbb{A}^\circ, \mathsf{f})$ where f is an operator on $\mathbb{A}^\circ$. If F is a filter and $\mathfrak{f}$ is a suite, $\mathfrak{f}^F$ is a suite, where $\mathfrak{f}^F(\mathbb{A}^\circ) = \{\mathsf{f} | \mathsf{f} \in \mathfrak{f}(\mathbb{A}^\circ)$ and $(\mathbb{A}^\circ, \mathsf{f}) \in F\}$.

If F is a filter and f is a listener-transforming operator, then f^F is a suite which, on input $\mathbb{A}^\circ$, is $\{\mathsf{f}\}$ if $(\mathbb{A}^\circ, \mathsf{f}) \in F$ and $\emptyset$ otherwise.

8.3 Primitive Operators

In this section we describe some very simple operators from which our more complicated test-solving operators are built.

Definition 8.4 (Identity operator)**.** *For any protoskeleton* $\mathbb{A}$*,* $\mathsf{Id}(\mathbb{A}) = \mathbb{A}$*. The corresponding role-assignment transformation is* $\mathsf{Id}(\mathcal{A}) = \mathcal{A}$*.*

The linking map Λ_{Id} *is the identity protomorphism:*

$$\Lambda_f(\mathbb{A}) = (\mathsf{Id}_{\mathbb{A}}, \mathsf{Id}_{\mathfrak{A}}).$$

The linking protomorphism for $\mathsf{Id}(\mathbb{A})$ is a protomorphism of assignment-commited protoskeletons.

If σ is a substitution, we can define an operator Sub_σ based on σ: basically, we apply σ to all algebraic parts of $\mathbb{A}$ while leaving its node structure alone.

Definition 8.5 (Substitution operator)**.** *If* $\sigma \in \text{End}(\mathfrak{A})$ *then we define the operator* Sub_σ *as follows. If* $\mathbb{A} = (I, \Theta, \prec, N, U)$ *then*

$$\mathsf{Sub}_\sigma(\mathbb{A}) = (I, \sigma \circ \Theta, \prec, \sigma(N), \sigma(U)).$$

The linking map $\Lambda_{\mathsf{Sub}_\sigma}$ *is defined as follows:*

$$\Lambda_{\mathsf{Sub}_\sigma}(\mathbb{A}) = (\mathsf{Id}_{\mathbb{A}}, \sigma)$$

The role-assignment transformation is defined as follows: If $\mathcal{A} = \{(\rho_s, \sigma_s) : s \in I\}$ *is a role assignment, then* $\mathsf{Sub}_\sigma(\mathcal{A}) = \{(\rho_s, \sigma \circ \sigma_s) : s \in I\}$.

Remark 8.6. If $\mathbb{A} = (I, \Theta, \prec, N, U)$ is a protoskeleton with validating role assignment $\mathcal{A}$, $\mathsf{Sub}_\sigma(\mathbb{A})$ is a protoskeleton with validating role assignment $\mathsf{Sub}_\sigma(\mathcal{A})$. This follows from the definition of protoskeleton (7.2).

$$N_{\mathsf{Sub}_\sigma(\mathbb{A})} = \sigma(N) \supseteq \sigma(N_{\mathcal{A}}) = N_{\mathsf{Sub}_\sigma(\mathcal{A})}$$

and

$$U_{\mathsf{Sub}_\sigma(\mathbb{A})} = \sigma(U) \supseteq \sigma(U_{\mathcal{A}}) = U_{\mathsf{Sub}_\sigma(\mathcal{A})}.$$

Remark 8.7. By Proposition 8.3, the substitution operators are well-behaved with respect to assignments.

Note that the linking protomorphism $\Lambda_{\mathsf{Sub}_\sigma}(\mathbb{A})$ is not necessarily a homomorphism since preservation of nodes of origination may fail. We say that a substitution σ is *homomorphic* if the linking protomorphism of Sub_σ is a homomorphism. Note also that the linking protomorphism is a protomorphism of listener-committed protoskeletons.

The compression operator combines two compatible strands. Recall that a strand is a node space consisting of a single strand.

Definition 8.8 (Compression operator)**.** *Suppose* s, s' *are strands and* $\mathbb{A}$ *is a protoskeleton.* $\mathsf{Comp}_{s,s'}(\mathbb{A})$ *is defined only when* $s, s' \in I_{\mathbb{A}}$ *and* $\Theta(s)$ *is a prefix of* $\Theta(s')$ *or* $\Theta(s')$ *is a prefix of* $\Theta(s)$*. Assume this is the case. If* s, s' *have different lengths, let* $s_{\max}$ *be the strand out of* $\{s, s'\}$ *of greater length and let* $s_{\min}$ *be the other strand; otherwise* $s_{\max} = s$ *and* $s_{\min} = s'$*.*

$$\mathsf{Comp}_{s,s'}(\mathbb{A}) = (I \setminus \{s_{\min}\}, \Theta|_{I \setminus \{s_{\min}\}}, \prec', N, U)$$

where the relation $\prec'$ *is defined based on the linking protomorphism*

$$\Lambda_{\mathsf{Comp}_{s,s'}}(\mathbb{A}) = (\varphi_{\mathsf{Comp}_{s,s'}}, \mathsf{Id}_{\mathfrak{A}})$$

where $\varphi_{\mathsf{Comp}_{s,s'}}$ *is the identity on the nodes in* $I \setminus \{s_{\min}\}$ *but* $\varphi_{\mathsf{Comp}_{s,s'}}(s_{\min}) = s_{\max}$.

The relation $\prec'$ is the smallest transitive relation such that for all $n, n' \in \mathbb{A}$, if $n \prec n'$ then $\varphi_{\mathsf{Comp}_{s,s'}}(n) \prec' \varphi_{\mathsf{Comp}_{s,s'}}(n')$.

The listener transformation of $\mathsf{Comp}_{s,s'}$ is defined as follows

$$\mathsf{Comp}_{s,s'}(L_{\mathbb{A}}) = \begin{cases} L_{\mathbb{A}} \setminus \{s_{\min}\} & \text{if } \{s, s'\} \subset L_{\mathbb{A}} \\ L_{\mathbb{A}} \setminus \{s, s'\} & \text{otherwise} \end{cases}$$

Remark 8.9. The compression operator is not always viewable as assignment-transforming: the main issue is that although $\Theta(s)$ is a prefix of $\Theta(s')$ this does not guarantee that both s and s' are instances of the same role in all role assignments. However, our description of $\mathsf{Comp}_{s,s'}$ as a listener-transforming operator guarantees that the linking protomorphism will always be a protomorphism of listener-committed protoskeletons.

Remark 8.10. First a definition: Given a role assignment $\mathcal{A} = \{(\rho_s, \sigma_s) : s \in I\}$ and strands s, s', $\mathcal{A}(s)$ *is compatible with* $\mathcal{A}(s')$ if and only if $\rho_s = \rho_{s'}$. Suppose s, s' are given. If an an assignment-committed protoskeleton $(\mathbb{A}, \mathcal{A})$ is such that s, s' are such that $\mathcal{A}(s)$ is compatible with $\mathcal{A}(s')$, then one can view $\mathsf{Comp}_{s,s'}$ as an assignment-transforming operator on $(\mathbb{A}, \mathcal{A})$. Moreover by Proposition 8.3 the operator is well-behaved with respect to listeners.

Remark 8.11. It is clear that $\mathsf{Comp}_{s,s'}(\mathbb{A})$ is a protoskeleton, regardless of the relation $\prec'$. However, if $\mathbb{A}$ is a preskeleton, $\mathsf{Comp}_{s,s'}(\mathbb{A})$ need not be a preskeleton because $\prec'$ may have a cycle violating Condition 1 of Definition 7.10.

Given a protoskeleton $\mathbb{A}$, order enrichment only affects the order relation.

Definition 8.12 (Order enrichment operator)**.** *If $\mathbb{A} = (I_{\mathbb{A}}, \Theta_{\mathbb{A}}, \prec', N_{\mathbb{A}}, U_{\mathbb{A}})$, then*

$$\mathsf{OE}(\mathbb{A}) = (I_{\mathbb{A}}, \Theta_{\mathbb{A}}, \prec', N_{\mathbb{A}}, U_{\mathbb{A}})$$

where $\prec'$ is the smallest transitive relation such that

1. *If $n \prec_{\mathbb{A}} n'$ then $n \prec' n'$ and*
2. *If $t \in U$ and t originates in $\mathbb{A}$ at n, and n' is any other node at which t is carried then $n \prec' n'$.*

The role-assignment transformation is the identity.

The linking protomorphism is the identity:

$$\Lambda_{\mathsf{OE}}(\mathbb{A}) = (\mathsf{Id}_{\mathbb{A}}, \mathsf{Id}_{\mathfrak{A}})$$

The linking protomorphism is clearly a protomorphism of assignment-committed protoskeletons.

Remark 8.13. By Proposition 8.3, the order enrichment operator is well-behaved with respect to listeners.

Finally, the augmentation operator adds a strand. Recall that for a role or pseudorole ρ, C_ρ is the trace of ρ.

Definition 8.14 (Augmentation operator)**.** *Let $\mathbb{A} = (I, \Theta, \prec, N, U)$. Let $n \in \mathbb{A}$ be any reception node. Let $\rho \in (P \cup \{\pounds\})$. Let $i \leq |C_\rho|$ such that evt C_ρ, i is a transmission and let σ be any substitution. Let $s^\star \notin I_\mathbb{A}$. Then*

$$\mathsf{Aug}_{n,\rho,i,\sigma,s^\star}(\mathbb{A}) = (I', \Theta', \prec', N', U')$$

where:

- $I' = I \cup \{s^\star\}$.
- $\Theta'(s) = \Theta(s)$ *for* $s \in I$ *and* $\Theta'(s^\star) = \sigma(C_\rho|i)$.
- $(s_1, i_1) \prec' (s_2, i_2)$ *if and only if one of the following holds:*
  1. $s_1, s_2 \in I$ *and* $(s_1, i_1) \prec (s_2, i_2)$,
  2. $s_1 = s_2 = s^\star$ *and* $i_1 < i_2$, *or*
  3. $s_1 = s^\star$, $s_2 \in I$, *and* $n \preceq (s_2, i_2)$.
- $N' = N \cup [\sigma]_* N_\rho$
- $U' = U \cup [\sigma]_* U_\rho$

The augmentation operator has an associated linking protomorphism where

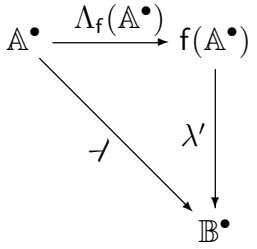
$$\Lambda_{\mathsf{Aug}_{n,\rho,i,\sigma,s^\star}}(\mathbb{A}) = (\mathsf{Id}_\mathbb{A}, \mathsf{Id}_\mathfrak{A}).$$

The role assignment transformation $\mathsf{Aug}_{n,\rho,i,\sigma,s^\star}(\mathcal{A})$ is equal to $\mathcal{A}$ on strands in I but maps $s^\star$ to (ρ, σ).

Note that $\mathsf{Aug}_{n,\rho,i,\sigma,s^\star}(\mathbb{A})$ may not always be a preskeleton even if $\mathbb{A}$ is, but $\Lambda_{\mathsf{Aug}_{n,\rho,i,\sigma,s^\star}(\mathbb{A})}$ is always a homomorphism, and is always a homomorphism of assignment-committed protoskeletons.

Remark 8.15. Any augmentation operator f is well-behaved with respect to listeners. This does not immediately follow from Proposition 8.3 since the node mapping component of the node mapping component of $\Lambda_{\mathsf{f}}(\mathbb{A}, \mathcal{A})$ is not surjective. However $\Lambda_{\mathsf{f}}(\mathbb{A}, \mathcal{A})$ only misses those elements in the new strand $s^\star$ and the role for this new strand is prescribed by ρ which is a parameter in f.

Theorem 8.16 (Preskeleton property preserved under primitive operators)**.** *Let $\mathbb{A}^\bullet$ and $\mathbb{B}^\bullet$ be preskeletons and f be an assignment-transforming primitive operator on $\mathbb{A}^\bullet$ such that there is a commutative diagram:*



(12)

where λ is a structure-preserving protomorphism of assignment-committed preskeletons and λ' is a protomorphism of assignment-committed protoskeletons. Then $\mathsf{f}(\mathbb{A}^\bullet)$ is a preskeleton.

Proof. Asymmetry of $\prec$ (part of Condition (1) of Definition 7.10) is assured by Proposition 7.23 and Condition (5) is assured by Proposition 7.26. For the remaining properties our proof proceeds by cases, one for each primitive operator. Let $\lambda = (\varphi, \sigma)$ and let $\lambda' = (\varphi', \sigma')$.

$\mathsf{f} = \mathsf{Id}$: Since $\mathsf{Id}(\mathbb{A}^\bullet) = \mathbb{A}^\bullet$, $\mathsf{f}(\mathbb{A}^\bullet)$ is a preskeleton, and meets all required conditions.

$\mathsf{f} = \mathsf{Sub}_{\sigma_0}$: Since the set of nodes and the ordering of $\mathsf{f}(\mathbb{A}^\bullet)$ are the same as those of $\mathbb{A}^\bullet$, it should be clear that $\mathsf{f}(\mathbb{A}^\bullet)$ meets condition (2) and satisfies the remaining properties of (1).

Proof of Property (3): Suppose $t \in N_{\mathsf{f}(\mathbb{A}^\bullet)}$. If t is carried at a node n of $\mathsf{f}(\mathbb{A}^\bullet)$ then $\sigma'(t) \in \sigma'(N_{\mathsf{f}(\mathbb{A}^\bullet)}) \subset N_{\mathbb{B}^\bullet}$ is carried at node $\varphi'(n)$. This contradicts $\mathbb{B}^\bullet$ being a preskeleton. Therefore t is carried at no node of $\mathsf{f}(\mathbb{A}^\bullet)$. Next, t is of the form $\sigma_0(s)$ for some some $s \in N_{\mathbb{A}}$. Every variable in s occurs at some node in $\mathbb{A}^\bullet$. Therefore every variable in t occurs at some node in $\mathbb{B}^\bullet$.

Proof of Property (4). Let $t \in U_{\mathsf{f}(\mathbb{A}^\bullet)}$. t is of the form $\sigma_0(s)$ for some some $s \in N_{\mathbb{A}}$. s is carried at a node n in $\mathbb{A}^\bullet$, and therefore t is carried at node n.

$\mathsf{f} = \mathsf{Comp}_{s,s'}$: Note that if f is assignment-transforming in this case then $\mathcal{A}(s)$, $\mathcal{A}(s')$ are compatible (Remark 8.10). The ordering $\prec_{\mathsf{Comp}_{s,s'}(\mathbb{A}^\bullet)}$ is defined to be transitive. Moreover, since the linking homomorphism is surjective, the ordering $\prec_{\mathsf{Comp}_{s,s'}(\mathbb{A}^\bullet)}$ includes the strand ordering. Condition (2) follows from the fact that $\varphi_{\mathsf{Comp}_{s,s'}}(\mathbb{A}^\bullet)$ is defined to be structure-preserving. Conditions (3) and (4) follow from the fact that $N_{\mathsf{Comp}_{s,s'}(\mathbb{A}^\bullet)} = N_{\mathbb{A}^\bullet}$ and $U_{\mathsf{Comp}_{s,s'}(\mathbb{A}^\bullet)} = U_{\mathbb{A}^\bullet}$, and that every event in $\mathbb{A}^\bullet$ is present in $\mathsf{f}(\mathbb{A}^\bullet)$ since we eliminate only completely duplicated nodes.

$\mathsf{f} = \mathsf{OE}$: Like $\mathsf{Comp}_{s,s'}$, $\prec_{\mathsf{OE}(\mathbb{A}^\bullet)}$ is defined to be structure-preserving and transitive, and all events in $\mathbb{A}^\bullet$ are present (at the same node) in $\mathsf{OE}(\mathbb{A}^\bullet)$, so all required conditions are clearly met.

$\mathsf{f} = \mathsf{Aug}_{n,\rho,i,\sigma_0,s^\star}$: Here, transitivity of $\prec_{\mathsf{f}(\mathbb{A}^\bullet)}$ is established as follows. If $n_1 = (s_1, i_1), n_2 = (s_2, i_2), n_3 = (s_3, i_3)$ are nodes in $\mathsf{f}(\mathbb{A}^\bullet)$ and $n_1 \prec n_2$ and $n_2 \prec n_3$ then:

- If $s_2 \in I$ then s_3 must be in I and $n_2 \prec_{\mathbb{A}^\bullet} n_3$. If $n_1 \prec_{\mathbb{A}^\bullet} n_2$ then by transitivity in $\mathbb{A}^\bullet$, $n_1 \prec_{\mathbb{A}^\bullet} n_3$. If $s_1 = s^\star$ then $n \prec_{\mathbb{A}^\bullet} n_2$, so $n \prec_{\mathbb{A}^\bullet} n_3$ and thus $n_1 \prec_{\mathsf{f}(\mathbb{A}^\bullet)} n_3$.
- If $s_2 = s^\star$ then $s_1 = s^\star$ also. If $s_3 = s^\star$ then $i_3 > i_2$ and $i_2 > i_1$ so $n_1 \prec n_3$. Otherwise, $n \prec_{\mathbb{A}^\bullet} n_3$ so $n_1 \prec_{\mathsf{f}(\mathbb{A}^\bullet)} n_3$.

Condition (2) holds in $\mathsf{f}(\mathbb{A}^\bullet)$: if $n_1 = (s_1, i_1)$ and $n_2 = (s_2, i_2)$ and $n_1 \prec n_2$ where $s_1 \neq s_2$ then there are two cases. If $s_1, s_2 \in I$ the property holds because $\mathbb{A}^\bullet$ is a preskeleton. Otherwise, it must be the case that $s_1 = s^\star$ and $s_2 \in I$. If there is no node between n_1 and n_2 it must be that $n_2 = n$ and n_1 is the last node of $s^\star$. But the event at n is a reception and the event at the last node of $s^\star$ is a transmission. □

An obvious corrollary to this theorem is that the same property holds for any *composition* of any number of primitive operators, so long as they are each assignment-transforming.

9 Suites and the Setwise Reduction

CPSA proceeds by maintaining a set of listener-committed skeletons $\mathbb{A}^\circ$ for

P. The initial state consists of one skeleton $\{\mathbb{A}^\circ\}$ which we call the *point of view*[1].

At each iteration, we rewrite the set by replacing an unrealized skeleton in the set with a set of skeletons called the *cohort* of the unrealized skeleton (for a specific critical path in an unrealized node in that skeleton). The cohort calculation takes place in two phases: first, we calculate the *pre-cohort*, which is a set of listener-committed preskeletons. Then, we calculate the *skeletonization* of each pre-cohort member, which produces a set listener-committed skeletons. We describe the cohort, the pre-cohort, and the skeletonization steps as suites.

In addition to these suites, CPSA also makes use of certain algorithms for arbitrary choices. In this document we pay no attention to how these functions are instantiated, we only remark the following:

- $NAME(\mathbb{A})$ is a choice of name for a new strand to add to $\mathbb{A}$. $NAME(\mathbb{A}) \notin I_{\mathbb{A}}$.
- $TEST(\mathbb{A})$ is a choice of a *test* (see Definition 10.5) in an unrealized skeleton.
- $UOI(\mathbb{A})$ is a choice of a unique origination issue. If $\mathbb{A}$ does not satisfy the condition of being a skeleton that every atom in $U_{\mathbb{A}}$ originates at at most one node, then $UOI(\mathbb{A})$ returns a triple (t, n, n') such that $t \in U_{\mathbb{A}}$ and $n \neq n'$ are both in $\mathcal{O}_{\mathbb{A}}(t)$.
- $SEARCH(S)$ is a choice of which skeleton to perform the cohort operation on. Given a set S of listener-committed skeletons, at least one of which is unrealized, $SEARCH(S)$ returns one of the unrealized skeletons in S.
- $FR(\mathbb{A}, \rho, i)$ is a substitution that maps each variable occurring in $C_\rho|_i$ to a distinct variable not occurring in $\mathbb{A}$ or in any role of the protocol.

9.1 The Pre-Cohort Suite

The pre-cohort suite is designed to infer additional honest behavior or restrictions on a skeleton in order to make progress in resolving a critical path.

[1]Actually, CPSA allows the user to specify a pre-skeleton which is then skeletonized, but this is a convenience feature.

Suppose $\mathbb{A}$ is an unrealized skeleton with valid listener set $L_{\mathbb{A}}$, and that n is an unrealized node of $\mathbb{A}$, and that p is a critical path with endpoint e_p of the term $evt_{\mathbb{A}}(n)$ in the fragment $\mathcal{F}_{\mathbb{A},n}$. The pre-cohort suite is

$$\mathfrak{P}_{n,p}(\mathbb{A}^\circ) = \mathfrak{c}_{n,p}(\mathbb{A}^\circ) \cup \mathfrak{a}_{n,p}(\mathbb{A}^\circ) \cup \mathfrak{d}_{n,p}(\mathbb{A}^\circ) \cup \mathfrak{l}_{n,p}(\mathbb{A}^\circ), \tag{13}$$

where the suites in the union are defined as follows.

Definition 9.1 (Contraction suite)**.**

$$\mathfrak{c}_{n,p}(\mathbb{A}^\circ) = \cup_{S \in Z}\{\mathsf{Sub}_\sigma | \sigma \in S\} \tag{14}$$

where Z is a set of $S_{a,b}$ for each $a \in \mathrm{Esc}(\mathcal{F}_{\mathbb{A},n}, e_p)$ and for each b visited by p where $S_{a,b}$ is a complete set of most general unifiers of a, b.

Definition 9.2 (Regular Augmentation suite)**.**

$$\mathfrak{a}_{n,p}(\mathbb{A}^\circ) = \{\mathsf{Sub}_{\sigma_1} \circ \mathsf{Aug}_{n,\rho,i,\sigma_0 \circ FR(\mathbb{A},\rho,i),s^\star}\} \tag{15}$$

where $\sigma_0 \in S_0, \sigma_1 \in S_1, s^\star = NAME(\mathbb{A})$ and $\rho, i, \sigma_0, \sigma_1, S_0, S_1$ are as defined below}.

- *$\mathcal{R} \in P$ and i is such that $C_{\mathcal{R}}(i)$ is defined and is a send event. Let $C = C_{\mathcal{R}}|i$.*
- *There is a path $pp \in \mathrm{CarPath}(C(i))$ and a term tt such that either (i) the endpoint of pp is a variable not of sort* MESG *and $tt = e_p$ or (ii) the endpoint of pp is a variable of sort* MESG *and $tt \in \mathrm{Targ}(\mathrm{Esc}(\mathcal{F}, e_p), e_p)$.*
- *S_0 is a set of most general unifiers of tt with the endpoint of $FR(\mathbb{A}, \rho, i)(pp)$.*
- *S_1 is a set of most general maps σ_1 such that for all $i' < i$ and for all paths $p' \in \mathrm{CarPath}(C(i'))$, if the endpoint of $\sigma_1((\sigma_0 \circ FR(\mathbb{A}, \rho, i))(p'))$ is $\sigma_1(e_p)$ then $\sigma_1(p')$ visits an element of $\sigma_1(\mathrm{Esc}(\mathcal{F}_{\mathbb{A},n}, e_p))$.*

Note that it is not obvious that S_1 exists, let alone that there is a finite set of such maps we can calculate efficiently. However, their existence, and the ability of CPSA *to identify a covering set of them, is proven in Appendix A.*

Definition 9.3 (Displacement suite.)**.**

$$\mathfrak{d}_{n,p}(\mathbb{A}^\circ) = \{\mathsf{Comp}_{s,s^\star} \circ \mathsf{Sub}_{\sigma_2 \circ \sigma_1} \circ \mathsf{Aug}_{n,\rho,i,\sigma_0 \circ FR(\mathbb{A},\rho,i),s^\star}\} \tag{16}$$

where $\rho, i, \sigma_0, \sigma_1, s^\star$ are as in the definition of $\mathfrak{a}_{n,p}(\mathbb{A}^\circ)$, $\sigma_2 \in S$, and s, S have the properties described below.

- *s is a strand in $\mathbb{A}$ and there exists a role assignment $\mathcal{A}$ such that s is associated with role ρ.*
- *S is a set of most general unifiers of the first i' events in s and the first i' events in s' where i' is the smaller of $|s|$ and i.*

Remark 9.4. Displacement is the only pre-cohort sub-suite that involves the compression operator, and is thus the only place where a concern arises as to whether operators in the pre-cohort suite can be viewed as assignment-transforming. A displacement operator $\mathsf{Comp}_{s,s^\star} \circ \mathsf{Sub}_{\sigma_2 \circ \sigma_1} \circ \mathsf{Aug}_{n,\rho,i,\sigma_0,s^\star}$ is assignment-transforming on $(\mathbb{A}, \mathcal{A})$ whenever $\mathcal{A}$ associates strand s with ρ, because in such cases compression occurs between two strands with the same role association, and therefore, the association of the combined strand is unambiguous.

Definition 9.5 (Listener augmentation suite.)**.** $\mathfrak{l}_{n,p}(\mathbb{A}^\circ) = \mathfrak{esl}_{n,p}(\mathbb{A}^\circ) \cup \mathfrak{cpl}_{n,p}(\mathbb{A}^\circ)$ *where:*

- **Escape set listener augmentation.** $\mathfrak{esl}_{n,p}(\mathbb{A}^\circ) = \{\mathsf{Aug}_{n,\mathcal{L},2,\sigma,s^\star} | s^\star = NAME(\mathbb{A}), \{|c|\}_u \in \mathrm{Esc}(\mathcal{F}_{\mathbb{A},n}, e_p)$ *and σ maps the m in the listener role to $inv(u)$*$\}$*.*
- **Critical path listener augmentation.** $\mathfrak{cpl}_{n,p}(\mathbb{A}^\circ) = \{\mathsf{Aug}_{n,\mathcal{L},2,\sigma,s^\star} | s^\star = NAME(\mathbb{A}), e_p = \{|c|\}_u$ *and σ maps the m in the listener role to u*$\}$*. Note that $\mathfrak{cpl}_{n,p}(\mathbb{A}^\circ)$ is $\emptyset$ if e_p is not an encryption.*

9.2 The Skeletonization Suite

The skeletonization suite is designed to rectify pre-skeletons back into skeletons.

The overall process is as follows. If $\mathbb{A}$ is not a skeleton, then either it is not a skeleton because some atom in U originates on more than one strand, or because there are required orderings that are not present. We resolve all

of the former issues before the latter. When a restricted atom originates on multiple strands, we resolve the problem in one of two ways: either we *merge* two strands on which the atom originates, via unification and compression, or we *de-originate* one of the origination points, in which we apply a substitution to the strand so that the atom is gained at an earlier point.

The skeletonization suite is $\mathfrak{S}(\mathbb{A}^\circ) = \{\mathsf{OE} \circ \mathsf{f} | \mathsf{f} \in \mathfrak{ur}(\mathbb{A}^\circ)\}$. The unique origination rectification suite, $\mathfrak{ur}$, is defined to be $\{\mathsf{Id}\}$ if $\mathrm{NUO}(\mathbb{A}^\circ)$ is empty, where $\mathrm{NUO}(\mathbb{A}^\circ)$, the set of non-compliant unique origination constraints, is defined to be $\{(t, n, n') | t \in U_\mathbb{A}$ and n and n' are distinct points of origination of t in $\mathbb{A}\}$, or otherwise defined recursively as follows:

$$\mathfrak{ur}(\mathbb{A}^\circ) = \{\mathsf{f} \circ \mathsf{f}' | (t, n, n') = UOI(\mathbb{A}), \mathsf{f}' \in \mathfrak{ur}_{t,n,n'}(\mathbb{A}^\circ), \mathsf{f} \in \mathfrak{ur}(\mathsf{f}(\mathbb{A}^\circ))\},$$

where $\mathfrak{ur}_{t,n,n'}(\mathbb{A}^\circ) = \mathfrak{M}_{t,n,n'}(\mathbb{A}^\circ) \cup \mathfrak{D}_{t,n,n'}(\mathbb{A}^\circ)$, and where $\mathfrak{M}$ (the merging suite) and $\mathfrak{D}$ (the deorigination suite) are defined below. Recall that $UOI(\mathbb{A})$ outputs a triple in $NUO(\mathbb{A})$ as long as $NUO(\mathbb{A})$ is non-empty.

Definition 9.6 (Merging suite.)**.** *Let n be in strand s and let n' be in strand s'. Let i be the smaller of the lengths of s and s'. Note that s and s' must be distinct because for any t there can be only one point of origination per strand, and n and n' are distinct. Then $\mathfrak{M}_{t,n,n'} = \{\mathsf{Comp}_{s,s'} \circ \mathsf{Sub}_\sigma | \sigma \in U\}$ where U is a set of most general unifiers of $\Theta(s)|i$ with $\Theta(s')|i$.*

Remark 9.7. Merging is the only skeletonization-related suite that involves the compression operator, and is thus the only place where a concern arises as to whether operators in the skeletonization suite can be viewed as assignment-transforming. A merging operator $\mathsf{Comp}_{s,s'} \circ \mathsf{Sub}_\sigma$ is assignment-transforming on $(\mathbb{A}, \mathcal{A})$ whenever $\mathcal{A}$ associates strands s and s' with the same role, because in such cases compression occurs between two strands with the same role association, and therefore, the association of the combined strand is unambiguous.

Definition 9.8 (Deorigination suite.)**.** *Let $n = (s, i)$. Let $V = \{(i', pp) | i' < i$ and $pp \in \mathrm{CarPath}(\mathrm{mesg}(s, i'))\}$. Then let $U_{i',pp}$ be a set of most general unifiers of the endpoint of pp with t, if the endpoint of pp is not a variable of sort* MESG*. If pp does terminate in a variable of sort* MESG*, let $U_{i',pp}$ be a set of most general unifiers of the endpoint of pp with a term that carries t. Then $\mathfrak{D}_{t,n,n'} = \cup_{(i',pp) \in V} \{\mathsf{Sub}_\sigma | \sigma \in U_{i',pp}\}$.*

Remark 9.9. When pp terminates in a variable of sort MESG, $U_{i',pp}$ is not finite. Therefore, if this situation ever comes up, CPSA will not terminate.

9.3 Post-Processing Filters

The post-processing filters filter out operators that are invalid or have failed to make progress. Once these filters are defined, we can define the cohort, which is CPSA's high-level reduction step.

Let $\mathbb{A}$ be a protoskeleton, let $L_{\mathbb{A}}$ be a valid listener set for $\mathbb{A}$, and let f be an operator defined on $\mathbb{A}^\circ$. Let n be an unrealized node of $\mathbb{A}$ and let p be a critical path of $evt_{\mathbb{A}}(n)$ in the fragment $\mathcal{F}_{\mathbb{A},n}$.

Definition 9.10 (Post-processing filter)**.** *The post-processing filter $PP_{n,p}$ is defined to be $WF \cap HC \cap SF_{n,p}$.*

Definition 9.11 (Well-formed filter)**.** $WF = \{(\mathbb{A}^\circ, \mathsf{f})$ *such that* $\mathsf{f}(\mathbb{A})$ *is a preskeleton*$\}$*.*

Definition 9.12 (Homomorphism check)**.** $HC = \{(\mathbb{A}^\circ, \mathsf{f})$ *such that* $\Lambda_{\mathsf{f}}(\mathbb{A})$ *is a homomorphism from* $\mathbb{A}$ *to* $\mathsf{f}(\mathbb{A})\}$*.*

Definition 9.13 (Solved filter)**.** $SF_{n,p}$ *is the set of pairs* $(\mathbb{A}^\circ, \mathsf{f})$ *such that p is weakly solved in $\mathcal{F}_{\mathsf{f}(\mathbb{A}),\varphi(n)}$ by σ, where $\Lambda_{\mathsf{f}(\mathbb{A})} = (\varphi, \sigma)$.*

In our proof later it would be desirable to prove that if $(\mathbb{A}^\circ, \mathsf{f})$ is in $SF_{n,p}$ then so is $(\mathbb{A}^\circ, \mathsf{g}\circ\mathsf{f})$ for all g. This is the case for some of the solved conditions:

Remark 9.14. Let n, p be a test in unrealized skeleton $\mathbb{A}^\circ$ and let $\mathbb{A}^\circ \overset{\varphi,\sigma}{\dashrightarrow} \mathbb{B}^\circ \overset{\varphi',\sigma'}{\dashrightarrow} \mathbb{C}^\circ$, and let $\mathcal{F} = \mathcal{F}_{\mathbb{A},n}$, $\mathcal{F}' = \mathcal{F}_{\mathbb{B},\varphi(n)}$, and $\mathcal{F}'' = \mathcal{F}_{\mathbb{C},\varphi'(\varphi(n))}$. Then:

1. If $p, \mathcal{F}, \mathcal{F}', (\varphi, \sigma)$ meet condition Sol1 then so do $p, \mathcal{F}, \mathcal{F}'', (\varphi' \circ \varphi, \sigma' \circ \sigma)$.
2. If $p, \mathcal{F}, \mathcal{F}', (\varphi, \sigma)$ meet condition Sol3 then so do $p, \mathcal{F}, \mathcal{F}'', (\varphi' \circ \varphi, \sigma' \circ \sigma)$.
3. If $p, \mathcal{F}, \mathcal{F}', (\varphi, \sigma)$ meet condition Sol4 then so do $p, \mathcal{F}, \mathcal{F}'', (\varphi' \circ \varphi, \sigma' \circ \sigma)$.

However, this is not the case generally: progress guaranteed by condition Sol2 or Sol5 can be reversed, for instance, by later unifications. What we can prove is that these properties are preserved under extensions of an operator that factor a map to another protoskeleton in which the same property holds.

Remark 9.15. Let n, p be a test in unrealized skeleton $\mathbb{A}_1^\circ$ and let

$$\mathbb{A}_1^\circ \overset{\varphi_1,\sigma_1}{\dashrightarrow} \mathbb{A}_2^\circ \overset{\varphi_2,\sigma_2}{\dashrightarrow} \mathbb{A}_3^\circ \overset{\varphi_3,\sigma_3}{\dashrightarrow} \mathbb{A}_4^\circ.$$

Let $\mathcal{F}_1 = \mathcal{F}_{\mathbb{A}_1,n}$, let $\mathcal{F}_2 = \mathcal{F}_{\mathbb{A}_2,\varphi_1(n)}$, let $\mathcal{F}_3 = \mathcal{F}_{\mathbb{A}_3,\varphi_2(\varphi_1(n))}$, and let $\mathcal{F}_4 = \mathcal{F}_{\mathbb{A}_4,\varphi_3(\varphi_2(\varphi_1(n)))}$. Let $\lambda = (\varphi, \sigma) = (\varphi_1, \sigma_1)$, $\lambda' = (\varphi', \sigma') = (\varphi_2 \circ \varphi_1, \sigma_2 \circ \sigma_1)$, and $\lambda'' = (\varphi'', \sigma'') = (\varphi_3 \circ \varphi_2 \circ \varphi_1, \sigma_3 \circ \sigma_2 \circ \sigma_1)$. Then:

1. If $p, \mathcal{F}_1, \mathcal{F}_2, \lambda$ meet condition Sol2 with path p', and if $p, \mathcal{F}_1, \mathcal{F}_4, \lambda''$ meet condition Sol2 with path $\sigma_3(\sigma_2(p'))$, then $p, \mathcal{F}_1, \mathcal{F}_3, \lambda'$ meet condition Sol5 with path $\sigma_2(p')$.

2. If $p, \mathcal{F}_1, \mathcal{F}_2, \lambda$ meet condition Sol5 with term tt, and if $p, \mathcal{F}_1, \mathcal{F}_4, \lambda''$ meet condition Sol5 with term $\sigma_3(\sigma_2(tt))$ then $p, \mathcal{F}_1, \mathcal{F}_3, \lambda'$ meet condition Sol5 with term $\sigma_2(tt)$.

Although these claims are complex, they both boil down to the same observation: if two terms have not been unified in $\mathbb{A}_2$ but are unified in $\mathbb{A}_3$, they must be unified in $\mathbb{A}_4$. Therefore, knowing they are not unified in both $\mathbb{A}_2$ and $\mathbb{A}_4$ implies they are not unified in $\mathbb{A}_3$. In the case of Sol2, the pairs of terms are those visited by (t, π) and those in the image of $\mathrm{Esc}(\mathcal{F}_1, e_p)$. In the case of Sol5, the pairs are the new target term and the image of $\mathrm{Targ}(\mathrm{Esc}(\mathcal{F}_1, e_p))$.

9.4 The Cohort and the CPSA Set Reduction

First, we define the cohort, the top-level suite used in the CPSA algorithm.

Definition 9.16 (The cohort)**.** *The cohort,* $\mathfrak{coh}_{n,p}(\mathbb{A}^\circ)$*, is defined to be*

$$\mathfrak{coh}_{n,p}(\mathbb{A}^\circ) = (\mathfrak{S} \circ \mathfrak{P}_{n,p})^{PP_{n,p}}(\mathbb{A}^\circ).$$

Next we define the reduction relation $\twoheadrightarrow$, on sets of listener-committed skeletons.

Definition 9.17 (Setwise reduction)**.** *Let* $S = \{(\mathbb{A}_i^\circ) | 1 \leq i \leq k\}$ *be a set of listener-committed skeletons. If* $\mathbb{A}_i^\circ = SEARCH(S)$ *and* $(n, p) = TEST(\mathbb{A}_i)$ *then* $S \twoheadrightarrow \{\mathbb{A}_j^\circ | 1 \leq j \leq k, j \neq i\} \cup \mathfrak{coh}_{n,p}[\mathbb{A}_i^\circ]$.

The overall operation of CPSA is as follows. The user specifies the *point of view* $\mathbb{A}^\bullet$ along with the protocol P. We calculate the initial set $S = \{\mathbb{A}^\circ\}$. Then we proceed as follows:

1. If $\exists T$ such that $S \twoheadrightarrow T$, let $S \leftarrow T$, and go to 1.

2. Else, output S.

CPSA in fact picks a particular element of S and a particular n and p and chooses the T resulting from *that* choice. However, CPSA is configurable to make different choices of these sorts, so we simply need to understand that CPSA (when it halts) outputs a *normal form* of the reduction $\twoheadrightarrow$, that is, a set S such that $\neg\exists T : S \twoheadrightarrow T$.

10 Suite Completeness

In this section we state and prove varous suite completeness theorems, culminating in a proof that CPSA's overall approach is complete.

First, we state a number of definitions that should help to simplify and clarify the complex theorem statements and proofs to follow. Most of the theorems proving the completeness of CPSA are ones that concern proving that coverage of a certain type can be maintained while advancing from one listener-committed protoskeleton to another via a certain suite. The notion of coverage varies for each theorem, and what is essential to understand from the lemmas are the particular properties of that coverage, which are distinct for each theorem. There is also, in each of these theorems, a complicated logical structure, but one that is largely similar for all the theorems. We first make definitions reflecting this generic logical structure, and then proceed to discuss the various lemmas and theorems.

Definition 10.1 (Coverage property)**.** *A* coverage context *is a tuple* $\mathcal{C} = (\mathfrak{X}, \mathfrak{Y}, \mathrm{Act})$ *where* $\mathfrak{X}, \mathfrak{Y}$ *are sets and* Act *is a mapping* $\mathrm{PSkel} \times \mathfrak{X} \times \mathrm{Opr}^\bullet \to \mathfrak{X}$ *where* $\mathrm{Opr}^\bullet$ *is the collection of assignment-transforming operators. A* coverage property *relative to the coverage context* $\mathcal{C}$ *is a set* $C \subset ((\mathrm{PSkel}^\bullet \times \mathfrak{X}) \times \mathrm{Protom} \times \mathfrak{Y})$ *such that for all* $((\mathbb{A}^\bullet, \beta), \lambda, \alpha) \in C$, λ *is a protomorphism from* $\mathbb{A}$*.*

If $\beta \in \mathfrak{X}$*, we use the notation* $\mathsf{f}(\mathbb{A}).\beta$ *to refer to* $\mathrm{Act}(\mathbb{A}, \beta, \mathsf{f})$*. In cases where* $\mathbb{A}$ *is unambiguous, we sometimes use the notation* $\mathsf{f}.\beta$*.*

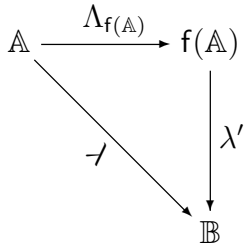
If $\mathfrak{X}$ or $\mathfrak{Y}$ are sets of tuples themselves, we omit the extra nexting of parentheses, as in $((\mathbb{A}^\bullet, t, n, n'), \lambda, \alpha_1, \alpha_2)$.

Definition 10.2 (Suite factoring protomorphisms from protoskeleton under conditions P guaranteeing conditions Q)**.** *Fix a coverage context* $\mathcal{C} = (\mathfrak{X}, \mathfrak{Y}, \mathrm{Act})$*. Let* $\mathfrak{s}$ *be a suite,* P *and* Q *coverage properties relative to* $\mathcal{C}$*,* $\mathbb{A}^\circ$ *a listener-committed protoskeleton, and* $\beta \in \mathfrak{X}$*.*

We say $\mathfrak{s}$ factors protomorphisms from $(\mathbb{A}^\circ, \beta)$ under conditions P guaranteeing conditions Q, expressed as

$$(\mathbb{A}^\circ, \beta) : [P \overset{\mathfrak{s}}{\Longrightarrow} Q] \tag{17}$$

if for all $\lambda, \mathbb{A}^\bullet, \alpha$ such that $((\mathbb{A}^\bullet, \beta), \lambda, \alpha) \in P$ and (adhering to the convention adopted in Remark 7.21) $\mathbb{A}^\bullet$ is an assignment committed protoskeleton such that $\mathbf{F}_{\bullet\to\circ}(\mathbb{A}^\bullet) = \mathbb{A}^\circ$, there is a commutative diagram



(18)

where $\mathsf{f} \in \mathfrak{s}(\mathbb{A}^\circ)$ is assignment-transforming on $(\mathbb{A}, \mathcal{A})$ and

$$((\mathsf{f}(\mathbb{A}^\bullet), \mathrm{Act}(\mathbb{A}, \beta, \mathsf{f})), \lambda', \alpha) \in Q.$$

Definition 10.3 (Suite factoring coverings from protoskeleton under conditions P guaranteeing conditions Q)**.** *Fix a coverage context $\mathcal{C} = (\mathfrak{X}, \mathfrak{Y}, \mathrm{Act})$. Let $\mathfrak{s}$ be a suite, P and Q coverage properties relative to $\mathcal{C}$, $\mathbb{A}^\circ$ a listener-committed protoskeleton, and $\beta \in \mathfrak{X}$.*

We say $\mathfrak{s}$ factors coverings from $(\mathbb{A}^\circ, \beta)$ under conditions P guaranteeing conditions Q expressed as

$$(\mathbb{A}^\circ, \beta) : [\![P \overset{\mathfrak{s}}{\Longrightarrow} Q]\!] \tag{19}$$

if for all $\lambda, \mathbb{A}^\bullet, \alpha$ such that $((\mathbb{A}^\bullet, \beta), \lambda, \alpha) \in P$ there exists an operator $\mathsf{f} \in \mathfrak{s}(\mathbb{A}^\circ)$ that is assignment-transforming on $(\mathbb{A}, \mathcal{A})$ and a λ' such that

$$\lambda|_{\mathrm{Rmv}(\mathbb{A}^\circ)} = (\lambda' \circ \Lambda_{\mathsf{f}(\mathbb{A}^\circ)})|_{\mathrm{Rmv}(\mathbb{A}^\circ)} \tag{20}$$

and $((\mathsf{f}(\mathbb{A}^\bullet), \mathrm{Act}(\mathbb{A}, \beta, \mathsf{f})), \lambda', \alpha) \in Q$.

Remark 10.4. This notion is nearly identical to the idea of a suite factoring a protomorphism from protoskeleton w.r.t. a predicate, except that the condition expressed by the commutative diagram (18) can only be considered to be true modulo listeners. Note the difference in notation: $\mathbb{A}^\circ : [\![P \overset{\mathfrak{s}}{\Longrightarrow} Q]\!]$ uses double brackets, as in our notation for coverage, whereas $\mathbb{A}^\circ : [P \overset{\mathfrak{s}}{\Longrightarrow} Q]$ does not.

Definition 10.5 (Test)**.** *A* test *of an unrealized skeleton* $\mathbb{A}^\circ$ *is a pair* (n, p) *where* n *is an unrealized node of* $\mathbb{A}^\circ$*, and* p *is a critical path of* $evt_{\mathbb{A}^\circ}(n)$ *in the fragment* $\mathcal{F}_{\mathbb{A}^\circ,n}$*.*

Definition 10.6 (Equivalence modulo listeners)**.** $\mathbb{A}^\circ \overset{R}{\equiv} \mathbb{B}^\circ$ *if and only if* $\mathrm{Rmv}(\mathbb{A}^\circ) = \mathrm{Rmv}(\mathbb{B}^\circ)$*.*

In what follows, we only use $\overset{R}{\equiv}$ among realized skeletons.

Definition 10.7 (Strict role-generated uniques)**.** *Let* $\mathbb{A}^\bullet = (\mathbb{A}, \mathcal{A})$ *be a preskeleton and let* $\mathbb{A}'^\circ$ *be a protoskeleton and* (φ, σ) *be a protomorphism where* $\mathbb{A}'^\circ \overset{\varphi,\sigma}{\dashrightarrow} \mathbb{A}^\circ$*. Then* $\mathbb{A}^\bullet$ *has strictly role-generated unique origination assumptions over* $(\mathbb{A}'^\circ, (\varphi, \sigma))$ *if* $U_\mathbb{A} = \sigma(U_{\mathbb{A}'}) \cup U_\mathcal{A}$*.*

Definition 10.8 (Coverage modulo listeners context)**.** *Let the* coverage modulo listeners context $\mathcal{C}_M$ *be* $(\mathrm{Opr}, (\mathrm{PSkel}^\circ \times \mathrm{PSkel}^\circ), \mathrm{Act}_M)$ *where* $Act_M(\mathbb{A}, \mathsf{g}, \mathsf{f}) = f \circ g$*.*

Definition 10.9 (Precohort coverage property)**.** *The precohort coverage property* $\mathrm{PCoh}_{n,p}$ *is defined with respect to the context* $\mathcal{C}_M$*, and includes the set of 4-tuples* $((\mathbb{A}^\bullet, \mathsf{g}), \lambda, \mathbb{B}^\circ, \mathbb{A}'^\circ)$ *such that*

- *P1.* $\mathbb{A}^\bullet$ *is a preskeleton.*
- *P2.* $\mathbb{B}^\circ$ *is a realized skeleton.*
- *P3. There exists* $\mathbb{B}'^\bullet$*, a realized skeleton, with* $\mathbb{B}^\circ \overset{R}{\equiv} \mathbb{B}'^\circ$ *such that* λ *is structure-preserving and* $\mathbb{A}^\bullet \overset{\lambda}{\dashrightarrow} \mathbb{B}'^\bullet$*.*
- *P4.* $\mathsf{g}(\mathbb{A}'^\circ) = \mathbb{A}^\circ$*.*
- *P5.* $\mathbb{A}^\bullet$ *has strictly role-generated unique originations assumptions over* $(\mathbb{A}'^\circ, \Lambda_{\mathsf{g}(\mathbb{A}'^\circ)})$*.*
- *P6. For all* λ' *such that* $\mathbb{A} \overset{\lambda'}{\dashrightarrow} \mathbb{C} \overset{\lambda''}{\dashrightarrow} \mathbb{B}'$ *with* $\lambda = \lambda'' \circ \lambda'$*,* p *is weakly solved in* $\mathcal{F}_{\mathbb{C},\varphi'(n)}$ *by* σ' *where* $\lambda' = (\varphi', \sigma')$*.*

Definition 10.10 (Cohort coverage property)**.** *The cohort coverage predicate* Coh *is defined with respect to the context* $\mathcal{C}_M$*, and includes the set of 4-tuples* $((\mathbb{A}^\bullet, \mathsf{g}), \lambda, \mathbb{B}^\circ, \mathbb{A}'^\circ)$ *such that*

- *C1.* $\mathbb{A}^\bullet$ *is a skeleton.*
- *P2.* $\mathbb{B}^\circ$ *is a realized skeleton.*
- *C3. There exists* $\mathbb{B}'^\bullet$*, a realized skeleton, with* $\mathbb{B}^\circ \overset{R}{\equiv} \mathbb{B}'^\circ$ *such that* $\mathbb{A}^\bullet \xrightarrow{\lambda} \mathbb{B}'^\bullet$.
- *P4.* $\mathsf{g}(\mathbb{A}'^\circ) = \mathbb{A}^\circ$.

In the definition of the cohort coverage property, we need to allow for existence of some $\mathbb{B}'$ rather than simply use $\mathbb{B}$, due to the the case in which f adds a listener strand that had no available image in $\mathbb{B}$.

The main theorem to be established first is the completeness of the CPSA cohort suite. This is proven given two central lemmas, which we state here but prove later. The pre-cohort completeness lemma establishes that the pre-cohort produces a complete (in terms of coverage) set of preskeleton outputs that pass the solved filter.

Lemma 10.11 (Pre-cohort completeness)**.** *Let* $\mathbb{A}^\circ$ *be an unrealized skeleton and let* (n, p) *be any test of* $\mathbb{A}^\circ$*. Then* $(\mathbb{A}^\circ, \mathsf{Id}) : [\![\mathrm{Coh} \xrightarrow{\mathfrak{P}^{PP_{n,p}}} \mathrm{PCoh}_{n,p}]\!]$.

We prove Lemma 10.11 in Section 12.

The skeletonization completeness lemma establishes that skeletonization is complete with regard to homomorphisms to a skeleton, and produces only skeletons. Informally: when $\mathbb{A} \to \mathbb{A}'$ where $\mathbb{A}$ is a skeleton but $\mathbb{A}'$ is only presumed to be a preskeleton, then any homomorphism from $\mathbb{A}$ to a skeleton that factors through the map to $\mathbb{A}'$ factors, further, through the linking protomorphism of some element of the skeletonization suite on $\mathbb{A}'$.

Definition 10.12 (Ancestor-aware coverage context)**.** *Let the* ancestor-aware coverage context $\mathcal{C}_A$ *be* $(\mathrm{Protom}, \mathrm{PSkel}^\bullet, \mathrm{Act}_A)$ *where* $Act_A(\mathbb{A}, \lambda, \mathsf{f}) = \Lambda_{\mathsf{f}(\mathbb{A})} \circ \lambda$.

Definition 10.13 (Skeleton coverage property)**.** *Let* $\mathbb{A}_0^\circ$ *be a skeleton. Then the skeleton coverage property* $\mathrm{Skl}_{\mathbb{A}_0}$ *is defined with respect to the context* $\mathcal{C}_A$*, and includes the set of 3-tuples* $((\mathbb{A}^\bullet, \lambda_0), \lambda, \mathbb{B}^\bullet)$ *such that*

- *S1.* $\mathbb{A}^\bullet$ *is a preskeleton.*
- *S2.* $\mathbb{B}^\bullet$ *is a skeleton.*
- *S3.* $\mathbb{A}_0^\circ \xrightarrow{\lambda_0} \mathbb{A}^\circ$.

- *S4.* $\mathbb{A}_0^\circ \xrightarrow{\lambda \circ \lambda_0} \mathbb{B}^\circ$.
- *S5.* $\mathbb{A}^\bullet \overset{\lambda}{\dashrightarrow} \mathbb{B}^\bullet$ *and* λ *is structure-preserving.*
- *S6.* $\mathbb{A}^\bullet$ *has strictly role-generated unique origination assumptions over* $(\mathbb{A}_0^\circ, \lambda_0)$.

Definition 10.14 (Skeleton homomorphism coverage property)**.** *Let* $\mathbb{A}_0^\circ$ *be a skeleton. Then the skeleton homomorphism coverage property* $\text{SklHom}_{\mathbb{A}_0}$ *is defined with respect to the context* $\mathcal{C}_A$*, and includes the set of 3-tuples* $((\mathbb{A}^\bullet, \lambda_0), \lambda, \mathbb{B}^\bullet)$ *that meet conditions S2, S3, S4, S6, and*

- *SH1.* $\mathbb{A}^\bullet$ *is a skeleton.*
- *SH5.* $\mathbb{A}^\bullet \xrightarrow{\lambda} \mathbb{B}^\bullet$.

In order to discuss the homomorphism check filter properly in the skeletonization completeness lemma, we must first describe the homomorphism check filter in a different way. Let $HC_{\mathbb{A},\lambda} = \{(\mathbb{A}'^\circ, \mathsf{f}) | \Lambda_{\mathsf{f}}(\mathbb{A}') \circ \lambda \text{ is a homomorphism from } \mathbb{A} \text{ to } \mathsf{f}(\mathbb{A}')\}$.

Lemma 10.15 (Skeletonization completeness)**.** *Let* $\mathbb{A}_0^\circ$ *be a skeleton and let* $\mathbb{A}^\circ$ *be a preskeleton such that* $\mathbb{A}_0^\circ \xrightarrow{\lambda_0} \mathbb{A}^\circ$. *Then* $(\mathbb{A}^\circ, \lambda_0) : [\text{Skl}_{\mathbb{A}_0} \xRightarrow{\mathfrak{S}^{WF \cap HC_{\mathbb{A}_0,\lambda_0}}} \text{SklHom}_{\mathbb{A}_0}]$.

We prove Lemma 10.15 in Section 11.

From these, we can give a proof of Theorem 10.16.

Theorem 10.16 (CPSA cohort completeness)**.** *Let* $\mathbb{A}^\circ$ *be an unrealized skeleton and let* (n, p) *be any test of* $\mathbb{A}^\circ$. *Then* $(\mathbb{A}^\circ, \mathsf{Id}) : [\![\text{Coh} \xRightarrow{\mathfrak{coh}_{n,p}} \text{Coh}]\!]$.

Proof. Let $\mathbb{A}^\circ$ be an unrealized skeleton and let (n, p) be a test of $\mathbb{A}^\circ$. Let $((\mathbb{A}^\bullet, \mathsf{Id}), \lambda, \mathbb{B}^\circ, \mathbb{A}^\circ) \in \text{Coh}$, with $(\mathbb{B}'^\bullet)$ satisfying condition C3.

By Lemma 10.11, $(\mathbb{A}^\circ, \mathsf{Id}) : [\![\text{Coh} \xRightarrow{\mathfrak{P}_{n,p}^{PP_{n,p}}} \text{PCoh}_{n,p}]\!]$. Let g be an assignment-transforming operator in $\mathfrak{P}_{n,p}^{PP_{n,p}}$ and λ' be such that $((\mathsf{g}(\mathbb{A}^\bullet), \mathsf{g})\lambda', \mathbb{B}^\circ, \mathbb{A}^\circ) \in \text{PCoh}_{n,p}$ and $\lambda|_{\text{Rmv}(\mathbb{A}^\circ)} = (\lambda' \circ \Lambda_{\mathsf{g}(\mathbb{A})})|_{\text{Rmv}(\mathbb{A}^\circ)}$. Let $\mathbb{B}''^\bullet = (\mathbb{B}'', \mathcal{B}'')$ be an assignment-committed realized skeleton satisfying condition P3.

We claim that the 3-tuple $((\mathsf{g}(\mathbb{A}^\bullet), \Lambda_{\mathsf{g}(\mathbb{A})}), \lambda', \mathbb{B}''^\bullet)$ is in $\text{Skl}_{\mathbb{A}}$.

- S1. Guaranteed because $(\mathbb{A}^\circ, \mathsf{g}) \in WF$.

- S2. We already know $\mathbb{B}''^{\bullet}$ is a skeleton, so S2 is met.
- S3. Guaranteed because $(\mathbb{A}^{\circ}, \mathsf{g}) \in HC$.
- S4. We know that $\lambda|_{\mathrm{Rmv}(\mathbb{A}^{\circ})} = (\lambda' \circ \Lambda_{\mathsf{g}(\mathbb{A})})|_{\mathrm{Rmv}(\mathbb{A}^{\circ})}$, and we know $\mathbb{A}_0^{\circ} \overset{\lambda' \circ \Lambda_{\mathsf{g}(\mathbb{A})}}{\dashrightarrow} \mathbb{B}''^{\circ}$. As noted below, λ' is structure-preserving, and so is Λ_{g} since it is a composition of linking maps of primitive operators. Therefore, $\lambda' \circ \Lambda_{\mathsf{g}(\mathbb{A})}$ is structure-preserving. Furthermore, it preserves points of origination, since all points of origination occur in $\mathrm{Rmv}(\mathbb{A}^{\circ})$. Therefore, $\lambda' \circ \Lambda_{\mathsf{g}(\mathbb{A})}$ is a homomorphism.
- S5. $\mathsf{g}(\mathbb{A}^{\bullet}) \overset{\lambda'}{\dashrightarrow} \mathbb{B}''^{\bullet}$ and λ' is structure-preserving by condition P3.
- S6. Guaranteed because of conditions P4 and P5.

Thus, by Lemma 10.15, there is an assignment-transforming $\mathsf{h} \in \mathfrak{S}^{WF \cap HC_{\mathbb{A}, \Lambda_{\mathsf{g}(\mathbb{A})}}}$ and a λ'' such that $((\mathsf{h}(\mathsf{g}(\mathbb{A}^{\bullet})), \Lambda_{\mathsf{g}(\mathbb{A})} \circ \lambda_0), \lambda'', \mathbb{B}''^{\bullet})$ is in $\mathrm{SklHom}_{\mathbb{A}}$. Let $\mathsf{f} = \mathsf{h} \circ \mathsf{g} \in \mathfrak{coh}_{n,p}(\mathbb{A}^{\circ})$. We claim that the 4-tuple $((\mathsf{f}(\mathbb{A}^{\bullet}), \mathsf{f}), \lambda'', \mathbb{B}^{\circ}, \mathbb{A}^{\circ})$ is in Coh.

- C1. Guaranteed by condition SH1.
- P2. Guaranteed by condition P2 for the 4-tuple $((\mathbb{A}^{\bullet}, \mathsf{Id}), \lambda, \mathbb{B}^{\circ}, \mathbb{A}^{\circ})$.
- C3. $\mathbb{B}''^{\bullet}$ satisfies this condition. We already know $\mathbb{B}''^{\bullet}$ is a realized skeleton and that $\mathbb{B}^{\circ} \overset{R}{\equiv} \mathbb{B}''^{\circ}$, and condition SH5 guarantees that $\mathbb{A}^{\bullet} \xrightarrow{\lambda''} \mathbb{B}''^{\bullet}$.
- P4. Applying f to $\mathbb{A}^{\circ}$ gives $\mathsf{f}(\mathbb{A}^{\circ})$.

We know that $\mathsf{f} \in \mathfrak{S} \circ \mathfrak{P}_{n,p}$; we need only prove that $(\mathbb{A}^{\circ}, \mathsf{f}) \in PP_{n,p}$. We know $(\mathbb{A}^{\circ}, \mathsf{f}) \in WF$ from condition C1 and $(\mathbb{A}^{\circ}, \mathsf{f}) \in HC$ from condition C3. Also, we know that $(\mathbb{A}^{\circ}, \mathsf{f}) \in SF_{n,p}$ by condition P6. □

10.1 Top-level Completeness Proof

Assuming the results of Theorem 10.16, we can prove the main completeness result:

Theorem 10.17 (CPSA overall completeness)**.** *Let $\mathbb{A}^\circ$ be the input to* CPSA *and suppose* CPSA *produces a set S in a finite number of steps during its setwise reduction. Let $H_{\mathbb{B}^\circ}$ be the set of homomorphisms $\mathbb{A}^\circ \xrightarrow{\lambda} \mathbb{B}^\circ$. Then*

$$[\![\mathbb{A}^\circ]\!] = \bigcup_{\mathbb{B}^\circ \in S} \{(\mathbb{A}', \lambda' \circ (\lambda|_{\mathrm{Rmv}(\mathbb{A}^\circ)}))|(\mathbb{A}', \lambda') \in [\![\mathbb{B}^\circ]\!], \lambda \in H_{\mathbb{B}^\circ}\}$$

Proof. Let $\mathbb{A}^\circ$ be the input to CPSA and suppose CPSA produces S during its setwise reduction in a finite number, n, of steps. Let $S_1, \ldots, S_n$ be the sequence of sets CPSA calculates, where $S_1 = \{\mathbb{A}^\circ\}$ and $S_n = S$. For each i from 1 to $n-1$, let $\mathbb{A}_i^\circ, n_i, p_i$ be such that $S_{i+1} = (S_i \setminus \{\mathbb{A}_i^\circ\}) \cup \mathfrak{coh}_{n_i,p_i}[\mathbb{A}_i^\circ]$, where $\mathbb{A}_i^\circ = SEARCH(S_i)$ and $(n_i, p_i) = TEST(\mathbb{A}_i)$.

Now we must prove that

$$[\![\mathbb{A}^\circ]\!] = \bigcup_{\mathbb{B}^\circ \in S} \{(\mathbb{A}', \lambda' \circ (\lambda|_{\mathrm{Rmv}(\mathbb{A}^\circ)}))|(\mathbb{A}', \lambda') \in [\![\mathbb{B}^\circ]\!], \lambda \in H_{\mathbb{B}^\circ}\}.$$

Let $(\mathbb{A}', \mu) \in [\![\mathbb{A}^\circ]\!]$, and let $\mathbb{B}'^\bullet$ be any realized skeleton such that $\mathrm{Rmv}(\mathbb{B}'^\circ) = \mathbb{A}'$, and let $\mathbb{A}^\bullet$ and ν be such that $\mathbb{A}^\bullet \xrightarrow{\nu} \mathbb{B}'^\bullet$ such that $\nu|_{\mathrm{Rmv}(\mathbb{A}^\circ)} = \mu$.

We define a sequence of tuples $(\mathbb{C}_i^\bullet, \lambda_i, \mathbb{B}_i^\bullet, \nu_i)$ such that $((\mathbb{C}_i^\bullet, \mathsf{Id}), \nu_i, \mathbb{B}'^\circ, \mathbb{C}_i^\circ) \in \mathrm{Coh}$ where $(\mathbb{B}_i^\bullet)$ satisfies condition C3, such that $\mathbb{A}^\bullet \xrightarrow{\lambda_i} \mathbb{C}_i^\bullet$, and $(\nu_i \circ \lambda_i)|_{\mathrm{Rmv}(\mathbb{A}^\circ)} = \mu$. The sequence is defined as follows:

- $\mathbb{B}_1^\bullet = \mathbb{B}'^\bullet$, $\mathbb{C}_1^\bullet = \mathbb{A}^\bullet$, $\lambda_1 = \Lambda_{\mathsf{Id}}$, and $\nu_1 = \nu$. The tuple $(\mathbb{C}_1^\bullet, \lambda_1, \mathbb{B}_1^\bullet, \nu_1)$ clearly has the required properties.

- If $\mathbb{C}_i^\circ \neq \mathbb{A}_i^\circ$ then $\mathbb{C}_{i+1}^\bullet = \mathbb{C}_i^\bullet$, $\mathbb{B}_{i+1}^\bullet = \mathbb{B}_i^\bullet$, $\lambda_{i+1} = \lambda_i$, and $\nu_{i+1} = \nu_i$. All required properties of the tuple are clear.

- If $\mathbb{C}_i^\circ = \mathbb{A}_i^\circ$, then by Theorem 10.16 there is an $\mathsf{f} \in \mathfrak{coh}_{n_i,p_i}(\mathbb{A}_i^\circ)$ and a ν_{i+1} such that $((\mathsf{f}(\mathbb{C}_i^\bullet), \mathsf{f}), \nu_{i+1}, \mathbb{B}'^\circ, \mathbb{C}_i^\circ) \in \mathrm{Coh}$ and a $\mathbb{B}_{i+1}^\bullet$ that satisfies the conditions of property C3. Note that $((\mathsf{f}(\mathbb{C}_i^\bullet), \mathsf{Id}), \nu_{i+1}, \mathbb{B}'^\circ, \mathsf{f}(\mathbb{C}_i^\circ)) \in \mathrm{Coh}$ and the same $\mathbb{B}_{i+1}^\bullet$ satisfies conditions C3, because the only condition affected by the changed fields is P4.

  Define $\mathbb{C}_{i+1}^\bullet = \mathsf{f}(\mathbb{C}_i^\bullet)$ and define $\lambda_{i+1} = \Lambda_{\mathsf{f}(\mathbb{A}_i)} \circ \lambda_i$. Then the tuple $(\mathbb{C}_{i+1}^\bullet, \lambda_{i+1}, \mathbb{B}_{i+1}^\bullet, \nu_{i+1})$ has the required properties:

– We already know $((\mathsf{f}(\mathbb{C}_i^\bullet), \mathsf{Id}), \nu_{i+1}, \mathbb{B}'^\circ, \mathsf{f}(\mathbb{C}_i^\circ)) \in \text{Coh}$ where $(\mathbb{B}_{i+1}^\bullet)$ satisfies condition C3.

– We know that $(\nu_{i+1} \circ \Lambda_{\mathsf{f}(\mathbb{A}_i)})|_{\text{Rmv}(\mathbb{A}_i^\circ)} = \nu_i|_{\text{Rmv}(\mathbb{A}_i^\circ)}$ and $(\nu_i \circ \lambda_i)|_{\text{Rmv}(\mathbb{A}^\circ)} = \mu$. Because λ_i is a homomorphism of listener-committed skeletons, all non-listener strands of $\mathbb{A}^\circ$ map to non-listener strands of $\mathbb{A}_i^\circ$. Thus, $(\nu_{i+1} \circ \Lambda_{\mathsf{f}(\mathbb{A}_i^\circ)} \circ \lambda_i)|_{\text{Rmv}(\mathbb{A}^\circ)} = (\nu_i \circ \lambda_i)|_{\text{Rmv}(\mathbb{A}^\circ)} = \mu$, and since $\Lambda_{\mathsf{f}(\mathbb{A}_i^\circ)} \circ \lambda_i = \lambda_{i+1}$, we have that $(\nu_{i+1} \circ \lambda_{i+1})|_{\text{Rmv}(\mathbb{A}^\circ)} = \mu$.

Thus, we have $(\mathbb{C}_n^\bullet, \lambda_n, \mathbb{B}_n^\bullet, \nu_n)$ with these properties, where $\mathbb{C}_n^\circ \in S_n = S$. Note that since $\mathbb{C}_n^\circ \xrightarrow{\nu_n} \mathbb{B}_n^\circ$ where $\mathbb{B}_n^\circ$ is realized, $(\text{Rmv}(\mathbb{B}_n^\circ), \nu_n|_{\text{Rmv}(\mathbb{C}_n^\circ)}) \in [\![\mathbb{C}_n^\circ]\!]$. Furthermore, $\lambda_n \in H_{\mathbb{C}_n^\circ}$ and $(\nu_n \circ \lambda_n)|_{\text{Rmv}(\mathbb{A}^\circ)} = \mu$. Since λ_n is a homomorphism of listener-committed skeletons, all non-listener strands of $\mathbb{A}^\circ$ map to non-listener strands of $\mathbb{C}_n^\circ$, so $(\nu_n|_{\text{Rmv}(\mathbb{C}_n^\circ)} \circ \lambda_n)|_{\text{Rmv}(\mathbb{A}^\circ)} = (\nu_n \circ \lambda_n)|_{\text{Rmv}(\mathbb{A}^\circ)} = \mu$.

This proves that $[\![\mathbb{A}^\circ]\!] \subseteq \bigcup_{\mathbb{B}^\circ \in S}\{(\mathbb{A}', \lambda' \circ (\lambda|_{\text{Rmv}(\mathbb{A}^\circ)}))|(\mathbb{A}', \lambda') \in [\![\mathbb{B}^\circ]\!], \lambda \in H_{\mathbb{B}^\circ}\}$. To prove equality we must also establish the other inclusion.

Suppose $(\mathbb{A}', \mu) \in [\![\mathbb{B}^\circ]\!]$ where $\mathbb{B}^\circ \in S$, and suppose $\lambda \in H_{\mathbb{B}^\circ}$. We know $\mathbb{A}^\circ \xrightarrow{\lambda} \mathbb{B}^\circ$. Let $\mathbb{B}'^\circ$ and ν be such that $\mathbb{B}^\circ \xrightarrow{\nu} \mathbb{B}'^\circ$, $\text{Rmv}(\mathbb{B}'^\circ) = \mathbb{A}'$, and $\mu = \nu|_{\text{Rmv}(\mathbb{B}^\circ)}$. Then $\mathbb{A}^\circ \xrightarrow{\nu \circ \lambda} \mathbb{B}'^\circ$, and $(\nu \circ \lambda)|_{\text{Rmv}(\mathbb{A}^\circ)} = (\mu \circ (\lambda|_{\text{Rmv}(\mathbb{A}^\circ)}))$ because λ is a homomorphism of listener-committed skeletons. Thus, $(\text{Rmv}(\mathbb{B}'^\circ), (\mu \circ (\lambda|_{\text{Rmv}(\mathbb{A}^\circ)}))) = (\mathbb{A}', (\mu \circ (\lambda|_{\text{Rmv}(\mathbb{A}^\circ)}))) \in [\![\mathbb{A}^\circ]\!]$. This completes the proof. □

11 Skeletonization

In this section we prove Lemma 10.15.

Definition 11.1 (Ancestor-aware coverage context with term and node pair)**.** *Let N be the set of nodes appearing in members of* PSkel*. Let the* ancestor-aware coverage context with term and node pair $\mathcal{C}_{A_2}$ *be* $((\text{Protom} \times \mathfrak{A} \times N \times N), \text{PSkel}^\bullet, \text{Act}_{A_2})$ *where* $Act_{A_2}(\mathbb{A}, (t, n, n', \lambda), \mathsf{f}) = (\sigma_{\mathsf{f}(\mathbb{A})}(t), \varphi_{\mathsf{f}(\mathbb{A})}(n), \varphi_{\mathsf{f}(\mathbb{A})}(n'), \Lambda_{\mathsf{f}(\mathbb{A})} \circ \lambda)$.

Definition 11.2 (Unique origination issue coverage property)**.** *Let $\mathbb{A}_0^\circ$ be a skeleton and let $\mathbb{A}_0^\circ \xrightarrow{\lambda_0} \mathbb{A}^\circ$. Then the unique origination issue coverage property* $\text{UrI}_{\mathbb{A}_0}$ *is defined with respect to the context $\mathcal{C}_{A_2}$, and includes the set of 3-tuples* $((\mathbb{A}^\bullet, t, n, n', \lambda_0), \lambda, \mathbb{B}^\bullet)$ *such that* $((\mathbb{A}^\circ, \lambda_0), \lambda, \mathbb{B}^\bullet) \in \text{Skl}_{\mathbb{A}_0}$ *and*

- *UI1. n and n' are distinct and are both points of origination of t in $\mathbb{A}$.*

The unique origination issue predicate refers to coverage (in the same sense as the skeleton coverage) in which there is a violation of a unique origination specification (n, t, t'). The next predicate indicates the same kind of coverage but with the violation resolved.

Definition 11.3 (Unique origination issue resolved coverage property)**.** *Let $\mathbb{A}_0^\circ$ be a skeleton and let $\mathbb{A}_0^\circ \xrightarrow{\lambda_0} \mathbb{A}^\circ$. Then the unique origination issue resolved property $\mathrm{UrR}_{\mathbb{A}_0}$ is defined with respect to the context $\mathcal{C}_{A_2}$, and includes the set of 3-tuples $((\mathbb{A}^\bullet, t, n, n', \lambda_0), \lambda, \mathbb{B}^\bullet)$ such that $((\mathbb{A}^\circ, \lambda_0), \lambda, \mathbb{B}^\bullet) \in \mathrm{Skl}_{\mathbb{A}_0}$ and*

- *UR1. $n = n'$ or one of n, n' are not points of origination of t in $\mathbb{A}$.*

First we state a main lemma about the unique origination rectification suite:

Lemma 11.4 (ur-universality)**.** *Let $\mathbb{A}_0^\circ$ be a skeleton and let $\mathbb{A}^\circ$ be a pre-skeleton such that $\mathbb{A}_0^\circ \xrightarrow{\lambda_0} \mathbb{A}^\circ$. Let t be a term in $U_{\mathbb{A}}$ and let n and n' be distinct nodes in $\mathbb{A}$ which are both points of origination of t in $\mathbb{A}$. Then $(\mathbb{A}^\circ, t, n, n', \lambda_0) : [\mathrm{UrI}_{\mathbb{A}_0} \xrightarrow{\mathfrak{ur}_{t,n,n'}} \mathrm{UrR}_{\mathbb{A}_0}]$.*

In other words, $\mathfrak{ur}_{t,n,n'}$ can resolve unique origination issues while maintaining skeleton coverage.

The proof is largely split into two cases: one for using the merging suite and one for using the deorigination suite. With a little work these can be their own lemmas.

Definition 11.5 (Unique origination issue (merging) coverage property)**.** *Let $\mathbb{A}_0^\circ$ be a skeleton and let $\mathbb{A}_0^\circ \xrightarrow{\lambda_0} \mathbb{A}^\circ$. Then the unique origination issue (merging) predicate $\mathrm{UrIM}_{\mathbb{A}_0}$ is defined with respect to the context $\mathcal{C}_{A_2}$, and includes the set of 3-tuples $((\mathbb{A}^\bullet, t, n, n', \lambda_0), \lambda, \mathbb{B}^\bullet)$ such that $((\mathbb{A}^\bullet, t, n, n', \lambda_0), \lambda, \mathbb{B}^\bullet) \in \mathrm{UrI}_{\mathbb{A}_0}$ and*

- *UIM1. $\varphi(n) = \varphi(n')$ where $\lambda = (\varphi, \sigma)$.*

Definition 11.6 (Unique origination issue (deorig) coverage property)**.** *Let $\mathbb{A}_0^\circ$ be a skeleton and let $\mathbb{A}_0^\circ \xrightarrow{\lambda_0} \mathbb{A}^\circ$. Then the unique origination issue (deorig) predicate $\mathrm{UrID}_{\mathbb{A}_0}$ is defined with respect to the context $\mathcal{C}_{A_2}$, and includes the set of 3-tuples $((\mathbb{A}^\bullet, t, n, n', \lambda_0), \lambda, \mathbb{B}^\bullet)$ such that $((\mathbb{A}^\bullet, t, n, n', \lambda_0), \lambda, \mathbb{B}^\bullet) \in \mathrm{UrI}_{\mathbb{A}_0}$ and*

- $\neg$*UIM1.* $\varphi(n) \neq \varphi(n')$ *where* $\lambda = (\varphi, \sigma)$.

Lemma 11.7 (Merging universality)**.** *Let* $\mathbb{A}_0^\circ$ *be a skeleton and let* $\mathbb{A}^\circ$ *be a preskeleton such that* $\mathbb{A}_0^\circ \xrightarrow{\lambda_0} \mathbb{A}^\circ$. *Let* t *be a term in* $U_\mathbb{A}$ *and let* n *and* n' *be distinct nodes in* $\mathbb{A}$ *which are both points of origination of* t *in* $\mathbb{A}$. *Then* $(\mathbb{A}^\circ, t, n, n', \lambda_0) : [\mathrm{UrIM}_{\mathbb{A}_0} \xrightarrow{\mathfrak{M}_{t,n,n'}} \mathrm{UrR}_{\mathbb{A}_0}]$.

Proof. Let $\mathbb{A}_0^\circ$ be a skeleton and let $\mathbb{A}^\circ$ be a preskeleton such that $\mathbb{A}_0^\circ \xrightarrow{\lambda_0} \mathbb{A}^\circ$. Let t be a term in $U_\mathbb{A}$ and let n and n' be distinct nodes in $\mathbb{A}$ which are both points of origination of t in $\mathbb{A}$. Suppose we have $((\mathbb{A}^\bullet, t, n, n', \lambda_0), \lambda, \mathbb{B}^\bullet) \in \mathrm{UrI}_{\mathbb{A}_0}$, and let $(\varphi, \sigma) = \lambda$ and $(s, i) = n$.

Observe that n and n' are in different strands (since each strand can only originate any atom once), and that φ maps these two strands to the same strand in $\mathbb{B}^\circ$. Let l be the minimum of the lengths of strands s and s' in $\mathbb{A}$. Then in order for (φ, σ) to be a well-defined protomorphism, it must be that σ is a unifier of $msg_\mathbb{A}(s, j)$ and $msg_\mathbb{A}(s', j)$ for every $1 \leq j \leq l$. Let σ_0 be a most general unifier of $\Theta(s)|l$ with $\Theta(s')|l$ such that $\sigma = \sigma' \circ \sigma_0$ for some σ'.

Let $\mathsf{f} = \mathsf{Comp}_{s,s'} \circ \mathsf{Sub}_{\sigma_0} \in \mathfrak{M}_{t,n,n'}(\mathbb{A}^\circ)$. Let $(\varphi_\mathsf{f}, \sigma_\mathsf{f}) = \Lambda_{\mathsf{f}(\mathbb{A}^\circ)}$. Let φ' be defined strandwise on $\mathsf{f}(\mathbb{A}^\circ)$ so that $\varphi'(\varphi_\mathsf{f}(s)) = \varphi(s)$ for all s; note that this is possible in $\mathsf{f}(\mathbb{A})$ since $\varphi(s) = \varphi(s')$. Note that since (φ, σ) is a protomorphism of assignment-committed protoskeletons, $\mathcal{A}$ assigns both s and s' to the same role, so f in this situation is assignment-transforming.

Then we claim that $((\mathsf{f}(\mathbb{A}^\bullet), \sigma_\mathsf{f}(t), \varphi_\mathsf{f}(n), \varphi_\mathsf{f}(n'), \Lambda_{\mathsf{f}(\mathbb{A})} \circ \lambda_0), \lambda', \mathbb{B}^\bullet) \in \mathrm{UrR}_{\mathbb{A}_0}$, where $\lambda' = (\varphi', \sigma')$. We have that $\varphi_\mathsf{f}(n) = \varphi_\mathsf{f}(n')$, so it remains for us to prove that $((\mathsf{f}(\mathbb{A}^\bullet), \Lambda_{\mathsf{f}(\mathbb{A}^\circ)} \circ \lambda_0), \lambda', \mathbb{B}^\bullet) \in \mathrm{Skl}_{\mathbb{A}_0}$. We proceed through each requirement in the definition of the skeleton coverage predicate:

- S1. $\mathsf{f}(\mathbb{A}^\bullet)$ is a preskeleton by Theorem 8.16.
- S2. Guaranteed because $((\mathbb{A}^\bullet, \lambda_0), \lambda, \mathbb{B}^\bullet) \in \mathrm{Skl}_{\mathbb{A}_0}$.
- S3. We must prove $\Lambda_{\mathsf{f}(\mathbb{A}^\circ)} \circ \lambda_0$ is structure-preserving and that it preserves points of origination. The former property is guaranteed since f is a composition of primitive operators and because λ_0 is structure-preserving. Furthermore, it preserves points of origination by Lemma 7.25.
- S4. $\lambda' \circ \Lambda_{\mathsf{f}(\mathbb{A}^\circ)} \circ \lambda_0 = \lambda \circ \lambda_0$, and by property S4 we previously know that $\mathbb{A}_0^\circ \xrightarrow{\lambda \circ \lambda_0} \mathbb{B}^\circ$.

- S5. We know λ is a protomorphism of assignment-committed protoskeletons; only the compression operator could have an effect on this, and we have already remarked that the two strands we compress both were assigned to the same role.

  Let $\prec$ refer to $\prec_{\mathbb{A}^\circ}$. Let $\prec_M$ be such that $n \prec_M n'$ only when $n = (s, i)$ and $n' = (s', j)$ or $n = (s', i)$ and $n' = (s, j)$ for some $i < j$. The ordering $\prec_{\mathsf{f}(\mathbb{A}^\circ)}$ is the transitive closure of $\prec \cup \prec_M$. Thus, $n \prec_{\mathsf{f}(\mathbb{A}^\circ)} n'$ if and only if we can define a sequence $n = n_0, n_1, \ldots, n_l = n'$ such that for every $1 \leq i \leq l, n_{i-1} \prec n_i$ or $n_{i-1} \prec_M n_i$.

  If $n_{i-1} \prec_M n_i$, then $\varphi'(n_{i-1}) \prec_{\mathbb{B}} \varphi'(n_i)$ because $\varphi'(n_{i-1})$ will precede $\varphi'(n_i)$ in the same strand in $\mathbb{B}$. If $n_{i-1} \prec_{\mathbb{A}} n_i$ then $\varphi'(n_{i-1}) \prec \varphi'(n_i)$ because $\varphi'(n_{i-1}) = \varphi(n_{i-1})$ and $\varphi'(n_i) = \varphi(n_i)$, and λ is structure-preserving. So we have that $\varphi(n) = \varphi(n_0) \prec_{\mathbb{B}} \ldots \prec_{\mathbb{B}} \varphi(n_l) = \varphi(n')$ and so $\varphi(n) \prec_{\mathbb{B}} \varphi(n')$ because $\prec_{\mathbb{B}}$ is transitive.

- S6. We know that $\mathsf{f}(\mathbb{A}^\bullet)$ has strictly role-generated unique origination assumptions over $(\mathbb{A}_0, \Lambda_{\mathsf{f}(\mathbb{A}^\circ)} \circ \lambda_0)$ because $(\mathbb{A}^\bullet)$ did over $(\mathbb{A}_0^\circ, \lambda_0)$ and $U_{\mathsf{f}(\mathbb{A}^\circ)} = \sigma_{\mathsf{f}(\mathbb{A}^\circ)}(U_{\mathbb{A}^\circ})$.

□

Lemma 11.8 (Deorigination universality)**.** *Let $\mathbb{A}_0^\circ$ be a skeleton and let $\mathbb{A}^\circ$ be a preskeleton such that $\mathbb{A}_0^\circ \xrightarrow{\lambda_0} \mathbb{A}^\circ$. Let t be a term in $U_{\mathbb{A}}$ and let n and n' be distinct nodes in $\mathbb{A}$ which are both points of origination of t in $\mathbb{A}$. Then $(\mathbb{A}^\circ, t, n, n', \lambda_0) : [\mathrm{UrID}_{\mathbb{A}_0} \xrightarrow{\mathfrak{D}_{t,n,n'}} \mathrm{UrR}_{\mathbb{A}_0}]$.*

Proof. Let $\mathbb{A}_0^\circ$ be a skeleton and let $\mathbb{A}^\circ$ be a preskeleton such that $\mathbb{A}_0^\circ \xrightarrow{\lambda_0} \mathbb{A}^\circ$. Let t be a term in $U_{\mathbb{A}}$ and let n and n' be distinct nodes in $\mathbb{A}$ which are both points of origination of t in $\mathbb{A}$. Suppose we have $((\mathbb{A}^\bullet, t, n, n', \lambda_0), \lambda, \mathbb{B}^\bullet) \in \mathrm{UrI}_{\mathbb{A}_0}$, and let $(\varphi, \sigma) = \lambda$ and $(s, i) = n$.

Since $\mathbb{B}$ is a skeleton, $\sigma(t)$ cannot originate at both $\varphi(n)$ and $\varphi(n')$. Without loss of generality, assume that $\varphi(n)$ is not a point of origination of $\sigma(t)$ in $\mathbb{B}$.

Note that $\sigma(t)$ is carried at node $\varphi(n)$ so it must be that there is an earlier node $(s, i') \prec n$ such that $\sigma(t)$ is the termination point of a carried path $(msg_{\mathbb{B}}(\varphi(s, i')), \pi)$. Let π' be the largest prefix of π such that $(msg_{\mathbb{A}^\circ}(s, i'), \pi')$ is a well-defined path; note also that this path is a carried path. Then either $\pi = \pi'$ or $(msg_{\mathbb{B}^\circ}(s, i'), \pi')$ terminates at a variable m of sort MESG. (Note

that in the latter case we are guaranteed to terminate at a variable of sort MESG because of the stipulation that the path is carried.) In the former case, σ is a unifier of the endpoint of $(msg_{\mathbb{A}}(s, i'), \pi)$ with t, and in the latter case σ is a unifier of m with a term that carries t. Either way, we can write $\sigma = \sigma' \circ \sigma_0$ where $\mathsf{f} = \mathsf{Sub}_{\sigma_0} \in \mathfrak{D}_{t,n,n'}(\mathbb{A}^\circ)$. Let $\varphi' = \varphi$.

We will show that $((\mathsf{f}(\mathbb{A}^\bullet), \sigma_0(t), n, n', \Lambda_{\mathsf{f}(\mathbb{A})} \circ \lambda_0), \lambda', \mathbb{B}^\bullet) \in \mathrm{UrR}_{\mathbb{A}_0}$, where $\lambda' = (\varphi', \sigma')$. We know that $\sigma_0(t)$ is carried in $msg_{\mathsf{f}(\mathbb{A}^\circ)}(s, i')$, so $\sigma_0(t)$ does not originate at node n in $\mathsf{f}(\mathbb{A}^\circ)$, so we need only prove that $((\mathsf{f}(\mathbb{A}^\bullet), \Lambda_{\mathsf{f}(\mathbb{A})} \circ \lambda_0), \lambda', \mathbb{B}^\bullet) \in \mathrm{Skl}_{\mathbb{A}_0}$. We proceed through each requirement in the definition of the skeleton coverage predicate:

- S1. $\mathsf{f}(\mathbb{A}^\bullet)$ is a preskeleton by Theorem 8.16.
- S2. Guaranteed because $((\mathbb{A}^\bullet, \lambda_0), \lambda, \mathbb{B}^\bullet) \in \mathrm{Skl}_{\mathbb{A}_0}$.
- S3. We must prove $\Lambda_{\mathsf{f}(\mathbb{A}^\circ)} \circ \lambda_0$ is structure-preserving and that it preserves points of origination. The former property is guaranteed since f is a composition of primitive operators and because λ_0 is structure-preserving. Furthermore, it preserves points of origination by Lemma 7.25.
- S4. $\lambda' \circ \Lambda_{\mathsf{f}(\mathbb{A}^\circ)} \circ \lambda_0 = \lambda \circ \lambda_0$, and by property S4 we previously know that $\mathbb{A}_0^\circ \xrightarrow{\lambda \circ \lambda_0} \mathbb{B}^\circ$.
- S5. Since it is the same as λ on strands, we already know λ' is a protomorphism of assignment-committed protoskeletons. Furthermore, the orderings in $\mathsf{f}(\mathbb{A})$ are the same as the orderings in $\mathbb{A}$, so λ' must be structure-preserving.
- S6. We know that $\mathsf{f}(\mathbb{A}^\bullet)$ has strictly role-generated unique origination assumptions over $(\mathbb{A}_0, \Lambda_{\mathsf{f}(\mathbb{A}^\circ)} \circ \lambda_0)$ because $(\mathbb{A}^\bullet)$ did over $(\mathbb{A}_0^\circ, \lambda_0)$ and $U_{\mathsf{f}(\mathbb{A}^\circ)} = \sigma_{\mathsf{f}(\mathbb{A}^\circ)}(U_{\mathbb{A}^\circ})$.

□

We now give the proof of Lemma 11.4.

proof of Lemma 11.4. Let $\mathbb{A}_0^\circ$ be a skeleton and let $\mathbb{A}^\circ$ be a preskeleton such that $\mathbb{A}_0^\circ \xrightarrow{\lambda_0} \mathbb{A}^\circ$. Let t be a term in $U_{\mathbb{A}}$ and let n and n' be distinct nodes in $\mathbb{A}$ which are both points of origination of t in $\mathbb{A}$. Suppose we have $((\mathbb{A}^\bullet, t, n, n', \lambda_0), \lambda, \mathbb{B}^\bullet) \in \mathrm{UrI}_{\mathbb{A}_0}$, and let $(\varphi, \sigma) = \lambda$ and $(s, i) = n$.

The proof proceeds by cases. One of the following must be the case:
Case 1: $\varphi(n) = \varphi(n')$. In this case, $((\mathbb{A}^\bullet, t, n, n', \lambda_0), \lambda, \mathbb{B}^\bullet) \in \mathrm{UrI}_{\mathbb{A}_0}$, so by Lemma 11.7 there exists an $\mathsf{f} \in \mathfrak{M}_{t,n,n'}(\mathbb{A}^\circ) \subset \mathfrak{ur}_{t,n,n'}(\mathbb{A}^\circ)$ and a λ' such that $((\mathsf{f}(\mathbb{A}^\bullet), \mathsf{f}(t), \mathsf{f}(n), \mathsf{f}(n'), \mathsf{f}(\lambda_0)), \lambda', \mathbb{B}^\bullet) \in \mathrm{UrR}_{\mathbb{A}_0}$ and such that $\lambda = \lambda' \circ \Lambda_{\mathsf{f}(\mathbb{A})}$.
Case 2: $\varphi(n) \neq \varphi(n')$. In this case, $((\mathbb{A}^\bullet, t, n, n', \lambda_0), \lambda, \mathbb{B}^\bullet) \in \mathrm{UrI}_{\mathbb{A}_0}$, so by Lemma 11.8 there exists an $\mathsf{f} \in \mathfrak{D}_{t,n,n'}(\mathbb{A}^\circ) \subset \mathfrak{ur}_{t,n,n'}(\mathbb{A}^\circ)$ and a λ' such that $((\mathsf{f}(\mathbb{A}^\bullet), \mathsf{f}(t), \mathsf{f}(n), \mathsf{f}(n'), \mathsf{f}(\lambda_0)), \lambda', \mathbb{B}^\bullet) \in \mathrm{UrR}_{\mathbb{A}_0}$ and such that $\lambda = \lambda' \circ \Lambda_{\mathsf{f}(\mathbb{A})}$. □

Next, we prove a lemma about the order enrichment operator.

Definition 11.9 (Skeleton coverage without unique origination issues)**.** *Let $\mathbb{A}_0^\circ$ be a skeleton. Then skeleton coverage without unique origination issues, $\mathrm{SklU}_{\mathbb{A}_0}$, is defined with respect to the context $\mathcal{C}_A$, and includes the set of 3-tuples $((\mathbb{A}^\bullet, \lambda_0), \lambda, \mathbb{B}^\bullet)$ in $\mathrm{Skl}_{\mathbb{A}_0}$ such that $NUO(\mathbb{A}) = \emptyset$.*

Lemma 11.10 (Order enrichment)**.** *Let $\mathbb{A}_0^\circ$ be a skeleton and let $\mathbb{A}^\circ$ be a preskeleton such that $\mathbb{A}_0^\circ \xrightarrow{\lambda_0} \mathbb{A}^\circ$. Then $(\mathbb{A}^\circ, \lambda_0) : [\mathrm{SklU}_{\mathbb{A}_0} \overset{\mathfrak{oe}}{\Longrightarrow} \mathrm{SklHom}_{\mathbb{A}_0}]$, where $\mathfrak{oe} = \{\mathsf{OE}\}$.*

Proof. Let $\mathbb{A}_0^\circ$ be a skeleton and let $\mathbb{A}^\circ$ be a preskeleton such that $\mathbb{A}_0^\circ \xrightarrow{\lambda_0} \mathbb{A}^\circ$. Let $\lambda_0 = (\varphi_0, \sigma_0)$ and let $\lambda = (\varphi, \sigma)$.

Let $((\mathbb{A}^\bullet, \lambda_0), \lambda, \mathbb{B}^\bullet) \in \mathrm{SklU}_{\mathbb{A}_0}$. Then we claim that $((\mathsf{OE}(\mathbb{A}^\bullet), \lambda_0), \lambda, \mathbb{B}^\bullet) \in \mathrm{SklHom}_{\mathbb{A}_0}$; note that $\Lambda_{\mathsf{OE}(\mathbb{A}^\circ)} \circ \lambda_0 = \lambda_0$. We need only establish that $\mathsf{OE}(\mathbb{A}^\bullet)$ is a skeleton and that λ is a homomorphism.

We know that $\mathsf{OE}(\mathbb{A}^\circ)$ is a preskeleton by Theorem 8.16. To prove that $\mathsf{OE}(\mathbb{A}^\circ)$ is a skeleton, we need to check two things: first, that each uniquely-originating atom originates on exactly one strand, and second, that the node of origination precedes each other node that carries that atom. The latter condition is guaranteed because all such instances of that requirement are ensured to be in the ordering after applying OE.

Each uniquely-originating atom originates on at most one strand in $\mathbb{A}^\circ$ since $\mathrm{NUO}(\mathbb{A}) = \emptyset$. To prove that each uniquely-originating atom originates on at least one strand, we appeal to the fact that $\mathbb{A}^\bullet$ has strictly role-generated unique origination assumptions over $(\mathbb{A}_0^\circ, \lambda_0)$. For every t in $U_{\mathsf{f}(\mathbb{A}^\circ)}$, either:

Case 1: There exists a $t' \in U_{\mathbb{A}_0^\circ}$ such that $t = \sigma_0(t)$. In this case, because $\mathbb{A}_0$ is a skeleton, t' must originate at a node $n \in \mathbb{A}_0$. Therefore, t

originates at $\varphi_0(n)$; this point of origination must be preserved because it is preserved under an extension, namely $\lambda \circ \lambda_0$.

Case 2: There exists a strand $s \in \mathsf{OE}(\mathbb{A}^\circ)$ with $\mathcal{A}(s) = (\rho_s, \sigma_s)$, and a $t' \in U_{\rho_s}$ such that $\sigma_s(t') = t$, such that t' originates in $C_{\rho_s}|\operatorname{len}(\Theta_{\mathsf{OE}(\mathbb{A}^\circ)}(s))$ at event i. Since $\mathsf{OE}(\mathbb{A}^\circ)$ is a preskeleton, $\sigma_s(t)$ originates in $\mathsf{OE}(\mathbb{A}^\circ)$ at (s, i).

Next we must prove that λ is a homomorphism.

First we prove that λ preserves points of origination. Let $t \in U_{\mathsf{OE}(\mathbb{A}^\circ)}$ and let n be a point of origination of t in $\mathsf{OE}(\mathbb{A}^\circ)$. There are two cases. Either:

Case 1: There is a $t_0 \in U_{\mathbb{A}_0}$ and a point $n_0 \in \mathbb{A}_0$ such that $\sigma_0(t_0) = t$ and t_0 originates at n_0 in $\mathbb{A}_0$. This point of origination must be preserved because it is preserved under $\lambda \circ \lambda_0$.

Case 2: Let $n = (s, i')$. Then $\mathcal{A}(s) = (\rho_s, \sigma_s)$, and there is a $t' \in U_{\rho_s}$ such that $\sigma_s(t') = t$ and such that t' originates in $C_{\rho_s}|\operatorname{len}(\Theta_{\mathbb{A}^\circ}(s))$ at event i. Since $\mathsf{OE}(\mathbb{A}^\circ)$ is a preskeleton, $\sigma_s(t)$ originates in $\mathsf{OE}(\mathbb{A}^\circ)$ at (s, i), and thus $i' = i$. Since λ is a protomorphism of assignment-committed protoskeletons, and since $\mathbb{B}^\bullet$ is a preskeleton, $\sigma(t)$ must originate at node $\varphi(n)$.

These cases are exhaustive since $\mathbb{A}^\bullet$ has strictly role-generated unique origination assumptions over $(\mathbb{A}_0, \lambda_0)$, and thus so does $\mathsf{OE}(\mathbb{A}^\bullet)$ since it only differs from $\mathbb{A}^\bullet$ in its orderings.

All that is left is to prove that λ is a structure-preserving protomorphism from $\mathsf{OE}(\mathbb{A}^\circ)$ to $\mathbb{B}^\circ$. We already know that λ is a structure-preserving protomorphism from $\mathbb{A}^\circ$ to $\mathbb{B}^\circ$, because we know $((\mathbb{A}^\circ, \lambda_0), \lambda, \mathbb{B}^\bullet) \in \mathrm{Skl}_{\mathbb{A}_0}$. However, $\mathsf{OE}(\mathbb{A}^\circ)$ may have additional orderings and we must prove these are preserved under λ.

Let $\prec$ refer to $\prec_{\mathbb{A}^\circ}$, that is, $\prec$ refers to the ordering before order-enrichment. Let $\prec_{\mathsf{OE}}$ be such that $n \prec_{\mathsf{OE}} n'$ where $n, n' \in \mathsf{OE}(\mathbb{A}^\circ)$ if and only if there is a $t \in U_{\mathsf{OE}(\mathbb{A}^\circ)}$ that originates at n and is carried at $n' \neq n$. Recall that $\prec_{\mathsf{OE}(\mathbb{A}^\circ)}$ is the transitive closure of $\prec \cup \prec_{\mathsf{OE}}$. Thus, $n \prec_{\mathsf{OE}(\mathbb{A}^\circ)} n'$ if and only if we can define a sequence $n = n_0, n_1, \ldots, n_l = n'$ such that for every $1 \leq i \leq l, n_{i-1} \prec n_i$ or $n_{i-1} \prec_{\mathsf{OE}} n_i$.

If $n_{i-1} \prec_{\mathsf{OE}} n_i$, there is a term t_{i-1} originating at n_{i-1} in $\mathsf{f}(\mathbb{A}^\circ)$ that is carried at n_i. Since λ preserves points of origination, $\sigma(t_{i-1}) \in U_{\mathbb{B}}$, $\sigma(t_{i-1})$

originates at $\varphi(n_{i-1})$ in $\mathbb{B}$, and is carried at $\varphi(n'_i)$ in $\mathbb{B}$. Since $\mathbb{B}$ is a skeleton, $\varphi(n_{i-1}) \prec_{\mathbb{B}} \varphi(n_i)$. Of course, if $n_{i-1} \prec n_i$ then $\varphi(n_{i-1}) \prec_{\mathbb{B}} \varphi(n_i)$ since λ is structure preserving from $\mathbb{A}^\circ$ to $\mathbb{B}^\circ$. So we have that $\varphi(n) = \varphi(n_0) \prec_{\mathbb{B}} \ldots \prec_{\mathbb{B}} \varphi(n_l) = \varphi(n')$ and so $\varphi(n) \prec_{\mathbb{B}} \varphi(n')$ because $\prec_{\mathbb{B}}$ is transitive. This completes the proof. □

Now we can prove Lemma 10.15.

Lemma 10.15 (Skeletonization completeness). *Let $\mathbb{A}_0^\circ$ be a skeleton and let $\mathbb{A}^\circ$ be a preskeleton such that $\mathbb{A}_0^\circ \xrightarrow{\lambda_0} \mathbb{A}^\circ$. Then $(\mathbb{A}^\circ, \lambda_0) : [\mathrm{Skl}_{\mathbb{A}_0} \xRightarrow{\mathfrak{S}^{WF \cap HC_{\mathbb{A}_0, \lambda_0}}} \mathrm{SklHom}_{\mathbb{A}_0}]$*

Proof. Let $\mathbb{A}_0^\circ$ be a skeleton and let $\mathbb{A}^\circ$ be a preskeleton such that $\mathbb{A}_0^\circ \xrightarrow{\lambda_0} \mathbb{A}^\circ$, and assume $((\mathbb{A}^\bullet, \lambda_0), \mu, \mathbb{B}^\bullet) \in \mathrm{Skl}_{\mathbb{A}_0}$.

We define a sequence of tuples $((\mathbb{A}_i^\bullet, \lambda_i), \mu_i, \mathbb{B}^\bullet, \mathsf{f}_i)$ with $1 \leq i \leq k$ as follows:

1. $\mathbb{A}_1^\bullet = \mathbb{A}^\bullet, \lambda_1 = \lambda_0, \mu_1 = \mu$. Note that $((\mathbb{A}_1^\bullet, \lambda_1), \mu_1, \mathbb{B}^\bullet) \in \mathrm{Skl}_{\mathbb{A}_0}$. We do not yet define f_1; nonetheless, note that $\mu_1 = \mu \circ (\Lambda_{\mathsf{f}_{i-1}(\mathbb{A}_{i-1}^\circ)} \circ \ldots \circ \Lambda_{\mathsf{f}_1(\mathbb{A}_1^\circ)})$ for $i = 1$, since the sequence is empty.

2. For $i \geq 1$, if $\mathrm{NUO}(\mathbb{A}_i)$ is nonempty, let (t_i, n_i, n'_i) be $UOI(\mathbb{A})$. We know that $((\mathbb{A}_i^\bullet, t_i, n_i, n'_i, \lambda_i), \mu_i, \mathbb{B}^\bullet) \in \mathrm{UrI}_{\mathbb{A}_0}$. By Lemma 11.4 that there is an $\mathsf{f}_i \in \mathfrak{ur}_{t_i, n_i, n'_i}(\mathbb{A}_i^\circ)$ and a μ_{i+1} such that $((\mathsf{f}_i(\mathbb{A}_i^\bullet), \mathsf{f}_i(t_i), \mathsf{f}_i(n_i), \mathsf{f}_i(n'_i), \mathsf{f}_i(\lambda_i)), \mu_{i+1}, \mathbb{B}^\bullet) \in \mathrm{UrR}_{\mathbb{A}_0}$ and $\mu = \mu_i \circ (\Lambda_{\mathsf{f}_{i-1}(\mathbb{A}_{i-1}^\circ)} \circ \ldots \circ \Lambda_{\mathsf{f}_1(\mathbb{A}_1^\circ)})$.

   If we let $\mathbb{A}_{i+1}^\bullet = \mathsf{f}_i(\mathbb{A}_i^\bullet)$ and $\lambda_{i+1} = \Lambda_{\mathsf{f}_i(\mathbb{A}_i^\circ)} \circ \lambda_i$ then $((\mathbb{A}_{i+1}^\bullet, \lambda_{i+1}), \mu_{i+1}, \mathbb{B}^\bullet) \in \mathrm{Skl}_{\mathbb{A}_0}$.

3. If $i \geq 1$ but $\mathrm{NUO}(\mathbb{A}_i)$ is empty then let $k = i$ and let $\mathsf{f}_i = \mathsf{Id}$.

By assuming that a k exists we are effectively assuming that this part of skeletonization terminates in finitely many steps. However, it clearly must: at every step, the sum of the heights of each point of origination of a uniquely originating term is strictly decreasing, because when we use the merging suite, we reduce the sum by the height of one of the origination points that merge, and when we use the deorigination suite, we either destroy a point of origination without replacing it, or we replace it at an earlier node in the same strand.

Note that $\mathsf{f}_k \in \mathfrak{ur}(\mathbb{A}_k^\circ)$ since $\text{NUO}(\mathbb{A}_k) = \emptyset$. Also note that for $1 \leq i < k$, if $\mathsf{f}_k \circ \ldots \circ \mathsf{f}_{i+1} \in \mathfrak{ur}(\mathbb{A}_{i+1}^\circ)$ then $\mathsf{f}_k \circ \ldots \circ \mathsf{f}_i \in \mathfrak{ur}(\mathbb{A}_i^\circ)$. Therefore let $\mathsf{g} = \mathsf{f}_k \circ \ldots \circ \mathsf{f}_1 \in \mathfrak{ur}(\mathbb{A}_1^\circ) = \mathfrak{ur}(\mathbb{A}^\circ)$, and let $\mathsf{f} = \mathsf{OE} \circ \mathsf{g} \in \mathfrak{S}(\mathbb{A}^\circ)$. Note that as a composition of assignment-transforming operators, f is assignment-transforming.

Note that we already know that $((\mathsf{g}(\mathbb{A}^\bullet), \Lambda_{\mathsf{g}(\mathbb{A}^\circ)} \circ \lambda_0), \mu', \mathbb{B}^\bullet) \in \text{SklU}_{\mathbb{A}_0}$. Therefore, $((\mathsf{f}(\mathbb{A}^\bullet), \Lambda_{\mathsf{f}(\mathbb{A}^\circ)} \circ \lambda_0), \mu', \mathbb{B}^\bullet) \in \text{SklHom}_{\mathbb{A}_0}$ by Lemma 11.10.

We must only prove that $(\mathbb{A}^\circ, \mathsf{f}) \in WF \cap HC_{\mathbb{A}_0,\lambda_0}$. $\mathsf{f}(\mathbb{A}^\circ)$ is clearly a preskeleton by Theorem 8.16. Also, $\Lambda_{\mathsf{f}(\mathbb{A}^\circ)} \circ \lambda_0$ is a homomorphism since $((\mathsf{f}(\mathbb{A}^\bullet), \lambda), \mu', \mathbb{B}^\bullet) \in \text{SklHom}_{\mathbb{A}_0}$.

This completes the proof that $(\mathbb{A}^\circ, \lambda_0) : [\text{Skl}_{\mathbb{A}_0} \xRightarrow{\mathfrak{S}^{WF \cap HC_{\mathbb{A}_0,\lambda_0}}} \text{SklHom}_{\mathbb{A}_0}]$. □

12 Pre-Cohort Completeness

In this section we build up to a proof of Lemma 10.11.

12.1 Preliminaries

First, some definitions. Earlier we defined cohort coverage and precohort coverage, which were defined in terms of a protomorphism to a realized skeleton remove-equivalent to a target. In two of the main lemmas needed to prove Lemma 10.11, we get factorization of the homomorphism to the same target; we only need that flexibility for the case where we employ listener augmentation. As a precursor to the definitions we use for these two lemmas, we first define *direct* cohort and precohort coverage.

Definition 12.1 (Direct coverage context)**.** *Let the* direct coverage context $\mathcal{C}_D$ *be* $(\text{Opr}, (\text{PSkel}^\bullet \times \text{PSkel}^\circ), \text{Act}_D)$ *where* $Act_D(\mathbb{A}, \mathsf{g}, \mathsf{f}) = f \circ g$.

Definition 12.2 (Direct cohort coverage)**.** *The direct cohort coverage property* DCoh *is defined with respect to the context* $\mathcal{C}_D$*, and includes the set of 4-tuples* $((\mathbb{A}^\bullet, \mathsf{g}), \lambda, \mathbb{B}^\bullet, \mathbb{A}'^\circ)$ *such that:*

- *DC1* $\mathbb{A}^\bullet$ *is a skeleton.*
- *DC2* $\mathbb{B}^\bullet$ *is a skeleton.*
- *DC3* $\mathbb{A}^\bullet \xrightarrow{\lambda} \mathbb{B}^\bullet$.

- *DC4* $\mathsf{g}(\mathbb{A}'^{\circ}) = \mathbb{A}^{\circ}$.

Definition 12.3 (Direct precohort coverage)**.** *The direct precohort coverage property* $\mathrm{DPCoh}_{n,p}$ *is defined with respect to the context* $\mathcal{C}_D$*, and includes the set of 4-tuples* $((\mathbb{A}^{\bullet}, \mathsf{g}), \lambda, \mathbb{B}^{\bullet}, \mathbb{A}'^{\circ})$ *such that:*

- *DPC1* $\mathbb{A}^{\bullet}$ *is a preskeleton.*
- *DC2* $\mathbb{B}^{\bullet}$ *is a skeleton.*
- *DPC3* $\mathbb{A}^{\bullet} \overset{\lambda}{\dashrightarrow} \mathbb{B}^{\bullet}$*, where* λ *is structure-preserving.*
- *DC4* $\mathsf{g}(\mathbb{A}'^{\circ}) = \mathbb{A}^{\circ}$.
- *DPC5* $\mathbb{A}^{\bullet}$ *has strictly role-generated unique origination assumptions over* $(\mathbb{A}'^{\circ}, \Lambda_{g(\mathbb{A}'^{\circ})})$.
- *DPC6. For all* λ' *such that* $\mathbb{A} \overset{\lambda'}{\dashrightarrow} \mathbb{C} \overset{\lambda''}{\dashrightarrow} \mathbb{B}$ *with* $\lambda = \lambda'' \circ \lambda'$*,* p *is weakly solved in* $\mathcal{F}_{\mathbb{C},\varphi'(n)}$ *by* σ' *where* $\lambda' = (\varphi', \sigma')$.

The three main lemmas in this section will correspond to the contraction, regular augmentation, and listener augmentation suites. In each case, we will state the lemma as a suite factoring statement; the conditions before will be more specific than cohort coverage (or direct cohort coverage) and the conditions after will be either precohort or direct precohort coverage.

12.2 Contractions

For the contraction case, the condition that guarantees a contraction will work is the following:

Definition 12.4 (Direct cohort coverage with test destroyed)**.** *Let* $\mathbb{A}^{\circ}$ *be an unrealized skeleton and let* (n, p) *be a test. Then the direct cohort coverage with test destroyed coverage property* $\mathrm{DCohTD}_{n,p}$ *is defined with respect to the context* $\mathcal{C}_D$*, and includes the set of 4-tuples* $((\mathbb{A}^{\bullet}, \mathsf{g}), \lambda, \mathbb{B}^{\bullet}, \mathbb{A}'^{\circ})$ *such that* $((\mathbb{A}^{\bullet}, \mathsf{g}), \lambda, \mathbb{B}^{\bullet}, \mathbb{A}'^{\circ}) \in \mathrm{DCoh}$ *and*

- *TD1* $\sigma(p)$ *visits* $\sigma(\mathrm{Esc}(\mathcal{F}_{\mathbb{A}^{\circ},n}, e_p))$*, where* $(\varphi, \sigma) = \lambda$.

This allows us to state and prove the completeness lemma for the contraction suite.

Lemma 12.5 (Contraction completeness)**.** *Let $\mathbb{A}^\circ$ be an unrealized skeleton and let (n,p) be a test. Then $(\mathbb{A}^\circ, \mathsf{Id}) : [\mathrm{DCohTD}_{n,p} \xrightarrow{\mathfrak{c}^{HC}_{n,p}} \mathrm{DPCoh}_{n,p}]$.*

Proof. Let $((\mathbb{A}^\bullet, \mathsf{Id}), \lambda, \mathbb{B}^\bullet, \mathbb{A}^\circ) \in \mathrm{DCohTD}_{n,p}$, where $\lambda = (\varphi, \sigma)$ and $p = (t, \pi)$. Let $p' = (\sigma(t), \pi)$. Since we know p' visits $\sigma(\mathrm{Esc}(\mathcal{F}_{\mathbb{A},n}, e_p))$, σ unifies a b visited by p and an a in $\mathrm{Esc}(\mathcal{F}_{\mathbb{A},n}, e_p)$. Therefore, there is a $\sigma_0 \in \cup_{S \in Z} S$ such that σ_0 unifies a and b, and such that $\sigma = \sigma' \circ \sigma_0$ for some σ', and $\mathsf{Sub}_{\sigma_0} \in \mathfrak{c}_{n,p}$. Let $\mathsf{f} = \mathsf{Sub}_{\sigma_0}$, and let $\lambda' = (\varphi, \sigma')$.

We claim that $((\mathsf{f}(\mathbb{A}^\bullet), \mathsf{f}), \lambda', \mathbb{B}^\bullet, \mathbb{A}^\circ) \in \mathrm{DPCoh}_{n,p}$.

1. DPC1 is guaranteed by Theorem 8.16.
2. DC2 is known since $((\mathbb{A}^\bullet, \mathsf{Id}), \lambda, \mathbb{B}^\bullet, \mathbb{A}^\circ) \in \mathrm{DCohTD}_{n,p}$.
3. DPC3 Whether λ' is a structure-preserving protomorphism or not, and whether it is a protomorphism of assignment-committed protoskeletons or not, depends only on its strand mapping, φ. Since λ is a homomorphism, it has these properties and thus so does λ'.
4. DC4 is obvious.
5. DPC5 Since $U_{\mathsf{f}(\mathbb{A})} = U_{\mathbb{A}}$, all unique origination assumptions are directly inherited, so this condition is met.
6. DPC6 Note that since σ_0 unifies a and b, it is immediately clear that $(\sigma_0(t), \pi)$ visits $\sigma_0(\mathrm{Esc}(\mathcal{F}_{\mathbb{A},n}, e_p))$. By Remark 9.14, this remains true under any extension.

We need only prove that $\mathbb{A}^\circ \xrightarrow{\Lambda_{\mathsf{f}(\mathbb{A})}} \mathsf{f}(\mathbb{A})$. It is obvious that $\Lambda_{\mathsf{f}(\mathbb{A})}$ is structure-preserving, and by Lemma 7.25, since λ preserves points of origination, so does $\Lambda_{\mathsf{f}(\mathbb{A})}$.

□

12.3 Listener Augmentation

Next, we address the case of listener augmentation. For this suite, we cannot make use of direct versions of our coverage properties, because if we add a listener but no corresponding listener is present in $\mathbb{B}$, there is no homomorphism. Rather, we find a proper place to *add* a listener to $\mathbb{B}$.

Definition 12.6 (Cohort coverage with certain values derivable)**.** *Let* $\mathbb{A}^\circ$ *be a skeleton and let* (n,p) *be a test and let* e_p *be the endpoint of* p*. The cohort coverage with certain values derivable coverage property* $\mathrm{CohCVD}_{n,p}$ *is defined with respect to the context* $\mathcal{C}_M$*, and includes the set of 4-tuples* $((\mathbb{A}^\bullet, \mathsf{g}), \lambda, \mathbb{B}^\circ, \mathbb{A}'^\circ)$ *in* Coh *such that*

- *CVD. Let* $\lambda = (\varphi, \sigma)$*. Then either:*
  - *EL. There exists* $\{\!|c|\!\}_u \in \mathrm{Esc}(\mathcal{F}_{\mathbb{A},n}, e_p)$ *such that* $\mathsf{inv}(\sigma(u)) \in \mathcal{S}_{\mathcal{F}_{\mathbb{B}',\varphi(n)}}$.
  - *CL.* $e_p = \{\!|c|\!\}_u$ *and* $\sigma(u) \in \mathcal{S}_{\mathcal{F}_{\mathbb{B}',\varphi(n)}}$.

Lemma 12.7 (Listener augmentation completeness)**.** *Let* $\mathbb{A}^\circ$ *be an unrealized skeleton and let* (n,p) *be a test. Then* $(\mathbb{A}^\circ, \mathsf{Id}) : [\![\mathrm{CohCVD}_{n,p} \xRightarrow{\mathfrak{l}^{HC}_{n,p}} \mathrm{PCoh}_{n,p}]\!]$.

Proof. Let $((\mathbb{A}^\bullet, \mathsf{Id}), \lambda, \mathbb{B}^\circ, \mathbb{A}^\circ)$ be in $\mathrm{CohCVD}_{n,p}$ with $\lambda = (\varphi, \sigma)$. Let $\mathbb{B}'^\bullet$ satisfy condition C3. There are two cases: either EL holds or CL holds. Let $s^\star$ be $NAME(\mathbb{A})$. Let $s'^\star \notin I_{\mathbb{B}'}$.

If condition EL holds for $\{\!|c|\!\}_u \in \mathrm{Esc}\,\mathcal{F}_{\mathbb{A},n}, e_p$ then let $\mathsf{f} = \mathsf{Aug}_{n,lsn,2,\sigma^\star,s^\star}$ where $\sigma^\star$ maps the m in the listener role to $inv(u)$. Note that $\mathsf{f} \in \mathfrak{es}\mathfrak{l}_{n,p}(\mathbb{A}^\circ)$. In this case, let $t = \mathsf{inv}(u)$. Note that $\sigma(t) \in D(P_{\mathbb{B}',\varphi(n)}) = \mathcal{S}_{\mathcal{F}_{\mathbb{B}',\varphi(n)}}$ because $\sigma(\mathsf{inv}(u)) = \mathsf{inv}(\sigma(u))$ as u is not a variable of sort MESG.

If condition CL holds then let $\mathsf{f} = \mathsf{Aug}_{n,\pounds,2,\sigma^\star,s^\star}$ where $\sigma^\star$ maps the m in the listener role to u. Note that $\mathsf{f} \in \mathfrak{cp}\mathfrak{l}_{n,p}(\mathbb{A}^\circ)$. In this case, let $t = u$. Note that $\sigma(t) \in \mathcal{S}_{\mathcal{F}_{\mathbb{B}',\varphi(n)}}$ by condition CL.

Either way, note that $\mathsf{f} \in \mathfrak{l}_{n,p}(\mathbb{A}^\circ)$. Let $\lambda' = (\varphi', \sigma)$ where $\varphi' = \varphi$ on all strands in $\mathbb{A}^\bullet$ and $\varphi'(s^\star) = s'^\star$. Let $\mathsf{f}' = \mathsf{F}_{n,(s'^\star,1)} \circ \mathsf{Aug}_{\varphi(n),\pounds,2,\sigma\circ\sigma^\star,s'^\star}$, where $\mathsf{F}_{n,n'}$ is a (non-primitive) operator that forces all transmissions before n to be before n'. Formally, $\mathsf{F}_{n,n'}(\mathbb{A}, \mathcal{A}) = ((I_\mathbb{A}, \Theta_\mathbb{A}, \prec', N_\mathbb{A}, U_\mathbb{A}), \mathcal{A})$ where $\prec'$ is the transitive closure of $\prec_\mathbb{A} \cup \prec_{n,n'}$ where $n_1 \prec_{n,n'} n'$ whenever n_1 is a transmission node such that $n_1 \prec_\mathbb{A} n$.

We claim that $((\mathsf{f}(\mathbb{A}^\bullet), \mathsf{f}), \lambda', \mathbb{B}^\circ, \mathbb{A}^\circ) \in \mathrm{PCoh}_{n,p}$, with $\mathsf{f}'(\mathbb{B}'^\bullet)$ satisfying condition P3.

- P1. We know $\mathsf{f}(\mathbb{A}^\bullet)$ is a preskeleton by Theorem 8.16.
- P2. $\mathbb{B}^\circ$ is a realized skeleton because $((\mathbb{A}^\bullet, (Id)), \lambda, \mathbb{B}^\circ, \mathbb{A}^\circ) \in \mathrm{Coh}$.
- P3. To see that $\mathsf{f}'(\mathbb{B}'^\bullet)$ is a skeleton, we need to establish that the nodes in $s'^\star$ are ordered after all the points of origination of any uniquely-originating atoms they carry. However, we know that $\sigma(t) \in D(P_{\mathbb{B}',\varphi(n)})$

and thus every uniquely-originating atom carried by $\sigma(t)$ is also carried by a transmission prior to n in $\mathbb{B}'$. Since $\mathbb{B}'$ is a skeleton, the origination point of that atom is before that transmission, and that transmission is before both nodes in the new listener in $\mathsf{f}'(\mathbb{B}'^{\bullet})$. The only reception in $\mathsf{f}'(\mathbb{B}'^{\bullet})$ not in $\mathbb{B}^{\bullet}$ is the reception in the listener strand $s'^{\star}$, of $\sigma(t)$, and $\sigma(t) \in D(P_{\mathbb{B}',\varphi(n)})$. Since $P_{\mathsf{f}'(\mathbb{B}'),(s'^{\star},1)} = P_{\mathbb{B}',\varphi'(n)}$, $\sigma(t) \in D(P_{\mathsf{f}'(\mathbb{B}'),(s'^{\star},1)})$, so the additional reception is realized, and all other receptions in $\mathsf{f}'(\mathbb{B}')$ are realized because they are realized in $\mathbb{B}'$. Therefore, $\mathbb{B}'^{\bullet}$ is realized.

It should be obvious that $\mathbb{B}'^{\circ} \overset{R}{\equiv} \mathsf{f}'(\mathbb{B}'^{\circ})$ and thus $\mathbb{B}^{\circ} \overset{R}{\equiv} \mathsf{f}'(\mathbb{B}'^{\circ})$.

It should be clear that λ' is a protomorphism $\mathsf{f}(\mathbb{A}^{\bullet}) \overset{\lambda'}{\dashrightarrow} \mathsf{f}'(\mathbb{B}'^{\bullet})$; the only role not known to be preserved by the fact that λ is assignment-preserving is the role of $s^{\star}$ which is *lsn* in both $\mathsf{f}(\mathbb{A}^{\bullet})$ and $\mathsf{f}'(\mathbb{B}'^{\bullet})$. λ' is structure-preserving: if $n_1 \prec n_2$ in $\mathsf{f}(\mathbb{A}^{\bullet})$ there are two possibilities. The first is that $n_1, n_2 \in \mathbb{A}^{\bullet}$ in which case, λ' maps both n_1 and n_2 just as λ does, and since λ is structure-preserving, $\varphi'(n_1) \prec_{\mathbb{B}'} \varphi'(n_2)$ so $\varphi'(n_1) \prec_{\mathsf{f}'(\mathbb{B}')} \varphi'(n_2)$. The other is that n_1 is in the strand $s^{\star}$ and $n \preceq n_2$. But then $\varphi'(n_1)$ is in the strand $s^{\star}$ and since $n, n_2 \in \mathbb{A}^{\bullet}$, $\varphi'(n) \prec_{\mathsf{f}'(\mathbb{B}')} \varphi'(n_2)$, and since by the operation of Aug, $\varphi'(n_1) \prec_{\mathsf{f}'(\mathbb{B}')} \varphi'(n)$, we have that $\varphi'(n_1) \prec_{\mathsf{f}'(\mathbb{B}')} \varphi'(n_2)$.

- P4 is obvious.
- P5 is true because since $\mathcal{L}$ has no uniquely originating atoms, $U_{\mathsf{f}(\mathbb{A})} = U_{\mathbb{A}}$.
- P6 Since t is in $S_{\mathcal{F}_{\mathsf{f}(\mathbb{A}),n}}$, we meet either condition Sol3 or Sol4 of Definition 5.5. By Remark 9.14 this remains true in any extension.

Furthermore, $\lambda|_{\mathrm{Rmv}(\mathbb{A}^{\circ})} = (\lambda' \circ \Lambda_{\mathsf{f}(\mathbb{A})})|_{\mathrm{Rmv}(\mathbb{A}^{\circ})}$ because $\Lambda_{\mathsf{f}(\mathbb{A})}|_{\mathrm{Rmv}(\mathbb{A}^{\circ})}$ is the identity and λ' is the same as λ on all strands in $\mathbb{A}$.

We need only prove that $\mathbb{A}^{\circ} \xrightarrow{\Lambda_{\mathsf{f}(\mathbb{A})}} \mathsf{f}(\mathbb{A})$. It is obvious that $\Lambda_{\mathsf{f}(\mathbb{A})}$ is structure-preserving. Since $\Lambda_{\mathsf{f}(\mathbb{A})}$ is the identity homomorphism on the algebra, it preserves points of origination.

Thus, $(\mathbb{A}^{\circ}, \mathsf{Id}) : [\![\mathrm{CohCVD}_{n,p} \overset{\mathfrak{l}_{n,p}}{\Longrightarrow} \mathrm{PCoh}_{n,p}]\!]$. □

12.4 Regular Augmentation

Third, we address the case of regular augmentation. Here, we can use the direct form of coverage again, but this time we make very strong assumptions about $\mathbb{B}$.

Definition 12.8 (Direct cohort coverage with good augmentation candidate)**.** *Let $\mathbb{A}^\circ$ be an unrealized skeleton and let (n, p) be a test. Then the direct cohort coverage with good augmentation candidate property* $\mathrm{DCohGAC}_{n,p}$ *is defined with respect to the context $\mathcal{C}_D$, and includes the set of 4-tuples $((\mathbb{A}^\bullet, \mathsf{g}), \lambda, \mathbb{B}^\bullet, \mathbb{A}'^\circ)$ with $\lambda = (\varphi, \sigma)$ such that $((\mathbb{A}^\bullet, \mathsf{g}), \lambda, \mathbb{B}^\bullet, \mathbb{A}'^\circ) \in \mathrm{DCoh}$ and there exists a strand $s_\mathbb{B}$ in $I_\mathbb{B}$ and a 4-tuple (ρ, i, π, tt) with ρ a protocol role, $i \leq |C_\rho|$ with event i in C_ρ being a transmission, $(C_\rho(i), \pi)$ is a carried path ending at a variable, and $tt \in \mathrm{Targ}(\mathrm{Esc}(\mathcal{F}_{\mathbb{A},n}, e_p), e_p)$ with (tt, π^{tt}) a carried path ending at e_p such that:*

- *WFC1. If the endpoint of $(C_\rho(i), \pi)$ is not a variable of sort* MESG *then $tt = e_p$.*
- *FC1. $s_\mathbb{B}$ is associated with role ρ in $\mathbb{B}^\bullet$, and $|\Theta_\mathbb{B}(s_\mathbb{B})| \geq i$.*
- *FC2. $(s_\mathbb{B}, i) \prec_\mathbb{B} \varphi(n)$.*
- *FC3. The endpoint of $(msg_\mathbb{B}(s_\mathbb{A}, i), \pi)$ is $\sigma(tt)$.*
- *WFC2. For all $i' < i$, for all carried paths $p' = (msg_\mathbb{B}(s_\mathbb{B}, i'), \pi')$ with endpoint $\sigma(e_p)$, p' visits $\sigma(\mathrm{Esc}(\mathcal{F}_{\mathbb{A},n}, e_p))$.*
- *WFC3. Either:*
  - *WFC3a. The path $(msg_\mathbb{B}(s_\mathbb{A}, i), \pi^\frown\pi^{tt})$ does not visit $\sigma(\mathrm{Esc}(\mathcal{F}_{\mathbb{A},n}, e_p))$, or*
  - *WFC3b. There is a prefix π' of $\pi \frown \pi^{tt}$ such that $(msg_\mathbb{B}(s_\mathbb{B}, i), \pi')$ traverses a term in $\mathrm{Esc}(\mathcal{F}_{\mathbb{B},\varphi(n)}, \sigma(e_p))$ and the endpoint of $(msg_\mathbb{B}(s_\mathbb{A}, i), \pi')$ is not in $\sigma(\mathsf{Targ}(\mathrm{Esc}(\mathcal{F}_{\mathbb{A},n}, e_p), e_p))$.*

A note on the naming of these conditions: WFC stands for "well-formed candidate." These are the conditions that guarantee that our candidate augmentation will be in the $\mathfrak{a}_{n,p}$ suite. FC stands for "factoring candidate." These conditions guarantee that the candidate augmentation results in a factoring of the coverage we care about.

Lemma 12.9 (Regular augmentation completeness)**.** *Let $\mathbb{A}^\circ$ be an unrealized skeleton and let (n,p) be a test. Then $(\mathbb{A}^\circ, \mathsf{Id}) : [\mathrm{DCohGAC}_{n,p} \xrightarrow{\mathfrak{a}^{HC}_{n,p}} \mathrm{PCoh}_{n,p}]$.*

Proof. Suppose $((\mathbb{A}^\bullet, \mathsf{Id}), \lambda, \mathbb{B}^\bullet, \mathbb{A}^\circ)$ is in $\mathrm{DCohGAC}_{n,p}$, with $s_\mathbb{B}, \rho, i, \pi, tt$ as specified in the definition. Let $s^\star = NAME(\mathbb{A})$.

Note that the endpoint of $(C_\rho(i), \pi)$ is a variable v. There is a unique most general unifier of $FR(\mathbb{A}, \rho, i)(v)$ with tt, call it σ_0. Let $\mathbb{A}'^\bullet = \mathsf{Aug}_{n,\rho,i,\sigma_0 \circ FR(\mathbb{A},\rho,i),NAME(\mathbb{A})}(\mathbb{A}^\bullet)$. We extend $\lambda = (\varphi, \sigma)$ to a map (φ', σ') from $\mathbb{A}'^\bullet$ to $\mathbb{B}^\bullet$ as follows: $\varphi' = \varphi$ on all strands other than $NAME(\mathbb{A})$, and $\varphi'(NAME(\mathbb{A})) = s_\mathbb{B}$. Since $s_\mathbb{B}$ is an instance of role ρ, φ' preserves role associations. Let $\sigma' = \sigma$ on all variables occurring in $\mathbb{A}$. The only variables occurring in $\mathbb{A}'$ that do not occur in $\mathbb{A}$ are the variables $FC(\mathbb{A}, \rho, i)(v')$ for v' a variable occurring in $C_\rho|_i$ other than $v' = v$. $\sigma'(FC(\mathbb{A}, \rho, i)(v')) = \sigma_{\mathcal{B}(s_\mathbb{B})}(v')$. Thus we have $\mathbb{A}'^\bullet \overset{\varphi', \sigma'}{\dashrightarrow} \mathbb{B}^\bullet$.

Note, by WFC2, that σ' is a map such that for all $i' < i$, for all carried paths $p' \in \mathrm{CarPath}(C(i'))$ such that the endpoint of $\sigma'((\sigma_0 \circ FR(\mathbb{A}, \rho, i))(p'))$ is $\sigma'(e_p)$, p' visits $\sigma'(\mathrm{Esc}(\mathcal{F}_{\mathbb{A},n}, e_p))$. Let σ_1 be a most general map with this property more general than σ', so that $\sigma' = \sigma'' \circ \sigma_1$.

Then $\mathsf{f} = \mathsf{Sub}_{\sigma_1} \circ \mathsf{Aug}_{n,\rho,i,\sigma_0 \circ FR(\mathbb{A},\rho,i),NAME(\mathbb{A})}$; we claim that $\mathsf{f} \in \mathfrak{a}_{n,p}(\mathbb{A})$. Observe:

- $C_\rho(i)$ is assumed to be a send event in Definition 12.8.
- If $pp = (C_\rho(i), \pi)$ then the endpoint of pp is a variable v, and if that variable is not of sort MESG then $tt = e_p$, by condition WFC1.
- σ_0 is *the* most general unifier of tt with v, and thus $S_0 = \{\sigma_0\}$ so in particular $\sigma_0 \in S_0$.
- σ_1 is a most general map such that for all $i' < i$ and for all carried paths $p' \in \mathrm{CarPath}(C(i'))$, if the endpoint of $\sigma_1((\sigma_0 \circ FR(\mathbb{A}, \rho, i))(p'))$ is $\sigma_1(e_p)$ then $\sigma_1(p')$ visits an element of $\sigma_1(\mathrm{Esc}(\mathcal{F}_{\mathbb{A},n}, e_p))$.

Let $\lambda' = (\varphi', \sigma'')$. We claim that $((\mathsf{f}(\mathbb{A}^\bullet), \mathsf{f}), \lambda', \mathbb{B}^\bullet, \mathbb{A}^\circ) \in \mathrm{DPCoh}_{n,p}$.

- DPC1 holds by Theorem 8.16.
- DC2 is already known to be true.

- We know $\mathsf{f}(\mathbb{A}^\bullet) \xrightarrow{\lambda'} \mathbb{B}^\bullet$. To see that λ' is structure-preserving, note that this property depends only on φ', and that $\varphi' = \varphi$ for all nodes other than those in $NAME(\mathbb{A})$. To see that φ' is structure-preserving, we need only establish that it is structure-preserving for orderings of the form $n_1 \prec n_2$ with n_1 in the strand $NAME(\mathbb{A})$. Either n_2 is also in that strand, in which case $\varphi'(n_1) \prec \varphi'(n_2)$ by the inclusion of the strand succession relation, or n_2 is not in that strand and $n \preceq n_2$. But $(s_\mathbb{B}, i) \prec \varphi(n)$ so by the structure-preserving property of φ and transitivity, $\varphi'(n_1) \prec_\mathbb{B} \varphi'(n_2)$. This establishes DPC3.

- DC4 is obvious.

- DPC5 is guaranteed, because f adds to $U_\mathbb{A}$ only where required to by $\mathsf{Aug}_{n,\rho,i,\sigma_0 \circ FR(\mathbb{A},\rho,i),NAME(\mathbb{A})}$.

- DPC6 is the most complicated to prove.

  If WFC3a holds, then condition Sol2 applies in both $\mathsf{f}(\mathbb{A}^\bullet)$ and $\mathbb{B}^\bullet$, so it must apply in any intermediate factorization by Remark 9.15.

  If WFC3b holds, then condition Sol5 applies in $\mathbb{B}^\bullet$; the fact that $(msg_\mathbb{B}(s_\mathbb{B}, i), \pi')$ traverses rather than visits the escape set establishes that the new potential target term is a *proper* carried substring of an escape set member. Sol5 applies in $\mathsf{f}(\mathbb{A}^\bullet)$, because $(msg_{\mathsf{f}(\mathbb{A})}(NAME(\mathbb{A}), i), \pi \frown \pi^{tt})$ is a carried path ending at t' and thus must have some maximal decryptable subpath $(msg_{\mathsf{f}(\mathbb{A})}(NAME(\mathbb{A}), i), \pi'')$. Furthermore, $(msg_\mathbb{B}(s_\mathbb{B}, i), \pi'')$ must be decryptable. Since $(msg_\mathbb{B}(s_\mathbb{B}, i), \pi')$ traverses the maximal decryptable subpath of $(msg_\mathbb{B}(s_\mathbb{B}, i), \pi \frown \pi^{tt})$, so does $(msg_{\mathsf{f}(\mathbb{A})}(NAME(\mathbb{A}), i), \pi')$. Therefore, Sol5 must apply in any intermediate factorization by Remark 9.15.

We need only prove that $\mathbb{A}^\circ \xrightarrow{\Lambda_{\mathsf{f}(\mathbb{A})}} \mathsf{f}(\mathbb{A})$. It is obvious that $\Lambda_{\mathsf{f}(\mathbb{A})}$ is structure-preserving, and by Lemma 7.25, since λ preserves points of origination, so does $\Lambda_{\mathsf{f}(\mathbb{A})}$.

□

12.5 Exhaustivity of the Cases

In this section we prove that cohort coverage implies one of the three critical coverage properties used in lemmas 12.5, 12.7, and 12.9.

The proof intimiately deals with the four conditions for a test being solved. Let $\mathbb{A}$ be an unrealized skeleton with test (n, p) and let $\mathbb{B}$ be a realized skeleton with $\mathbb{A} \xrightarrow{\varphi,\sigma} \mathbb{B}$. The four conditions (from Definition 5.5) are:

- Sol1. $\sigma(p)$ visits $\sigma(\mathrm{Esc}(\mathcal{F}_{\mathbb{A},n}, e_p))$.
- Sol2. There is a carried path from an element of $T_{\mathcal{F}_{\mathbb{A},n}}$ to $\sigma(e_p)$ which does not visit $\sigma(\mathrm{Esc}(\mathcal{F}_{\mathbb{A},n}, e_p))$.
- Sol3. There exists a $\{\!|b|\!\}_u \in \mathrm{Esc}(\mathcal{F}_{\mathbb{A},n}, e_p)$ such that $\sigma(\mathsf{inv}(u)) \in \mathcal{S}_{\mathcal{F}_{\mathbb{B},\varphi(n)}}$.
- Sol4. $e_p = \{\!|b|\!\}_u$ and $\sigma(u) \in \mathcal{S}_{\mathcal{F}_{\mathbb{B},\varphi(n)}}$.

First we state and prove the top-level proof, assuming a lemma we will prove later.

Lemma 12.10 (Case exhaustivity)**.** *Let $\mathbb{A}^\circ$ be an unrealized skeleton with test (n, p), and let $((\mathbb{A}^\bullet, \mathsf{Id}), \lambda, \mathbb{B}^\circ, \mathbb{A}^\circ) \in \mathrm{Coh}$, with $\mathbb{B}'^\bullet$ satisfying condition C3. Then one of the following holds:*

1. $((\mathbb{A}^\bullet, \mathsf{Id}), \lambda, \mathbb{B}'^\bullet, \mathbb{A}^\circ) \in \mathrm{DCohTD}_{n,p}$,
2. $((\mathbb{A}^\bullet, \mathsf{Id}), \lambda, \mathbb{B}'^\bullet, \mathbb{A}^\circ) \in \mathrm{DCohGAC}_{n,p}$, *or*
3. $((\mathbb{A}^\bullet, \mathsf{Id}), \lambda, \mathbb{B}^\circ, \mathbb{A}^\circ) \in \mathrm{CohCVD}_{n,p}$.

Proof. The main observation is that since $\sigma(p)$ is not a critical path in $\mathbb{B}$ at $\varphi(n)$ (where $\lambda = (\varphi, \sigma)$), by Theorem 5.6, p is solved in $\mathcal{F}_{\mathbb{B},\varphi(n)}$ by σ.

Note that Sol1 is just the same as TD1, so in that case, $((\mathbb{A}^\bullet, \mathsf{Id}), \lambda, \mathbb{B}'^\bullet, \mathbb{A}^\circ) \in \mathrm{DCohTD}_{n,p}$.

Note also that Sol3 is the same as EL (since $\sigma(\mathsf{inv}(u)) = \mathsf{inv}(\sigma(u))$), and Sol4 is the same as CL, so in either of those two cases, $((\mathbb{A}^\bullet, \mathsf{Id}), \lambda, \mathbb{B}^\circ, \mathbb{A}^\circ) \in \mathrm{CohCVD}_{n,p}$.

By Lemma 12.13, if neither of the two cases above apply, then there exists a good augmentation candidate (ρ, i, π, tt) and a target strand $s_{\mathbb{B}'}$. Therefore, $((\mathbb{A}^\bullet, \mathsf{Id}), \lambda, \mathbb{B}'^\bullet, \mathbb{A}^\circ) \in \mathrm{DCohGAC}_{n,p}$. □

Before we state and prove our lemma guaranteeing the existance of a good augmentation candidate, we state and prove two lemmas that will be helpful.

Lemma 12.11 (Image of the escape set without Sol3)**.** *Let $\mathbb{A}$ be an unrealized protoskeleton with test (n, p) and let $\mathbb{B}$ be a protoskeleton such that $\mathbb{A} \overset{\varphi,\sigma}{\dashrightarrow} \mathbb{B}$, and suppose that Sol3 does not hold. Then we have that $\sigma(\mathrm{Esc}(\mathcal{F}_{\mathbb{A},n}, e_p)) \subset \mathrm{Esc}(\mathcal{F}_{\mathbb{B},\varphi(n)}, \sigma(e_p))$.*

Proof. Let $t \in \mathrm{Esc}(\mathcal{F}_{\mathbb{A},n}, e_p)$. Then either $t = e_p$ and $e_p \in \mathrm{Cl}^{\downarrow}(P_{\mathcal{F}_{\mathbb{A},n}}, \mathcal{S}_{\mathcal{F}_{\mathbb{A},n}})$, or t is a term in $\mathrm{Fr}(P_{\mathcal{F}_{\mathbb{A},n}}, \mathcal{S}_{\mathcal{F}_{\mathbb{A},n}})$ that carries e_p.

First we observe that $\sigma(\mathrm{Cl}^{\downarrow}(P_{\mathcal{F}_{\mathbb{A},n}}, \mathcal{S}_{\mathcal{F}_{\mathbb{A},n}})) \subset \mathrm{Cl}^{\downarrow}(P_{\mathcal{F}_{\mathbb{B},\varphi(n)}}, \mathcal{S}_{\mathcal{F}_{\mathbb{B},\varphi(n)}})$, which takes care of the former case. In the latter case, we also need to show that t is not the endpoint of a path that is the proper prefix of another carried path in the downward closure. However since we know that in such a case t is an encryption $\{\!|b|\!\}_u$, we know that $\sigma(\mathsf{inv}(u)) \notin \mathcal{S}_{\mathcal{F}_{\mathbb{B},\varphi(n)}}$ because Sol3 does not hold. Therefore, $\sigma(t)$ is in the frontier. □

Lemma 12.12 (Critical derivability property preserved without Sol4)**.** *Let $\mathbb{A}$ be an unrealized protoskeleton with test (n, p) and let $\mathbb{B}$ be a realized skeleton such that $\mathbb{A}^\bullet \overset{\varphi,\sigma}{\dashrightarrow} \mathbb{B}^\bullet$, and suppose that Sol4 does not hold. Then of $n' \prec \varphi(n)$ is any reception node in $\mathbb{B}$ and p' is any carried path from $msg_{\mathbb{B}}(n')$ to $\sigma(e_p)$, p' visits $\mathrm{Esc}(P_{\mathcal{F}_{\mathbb{B},n'}}, \mathcal{S}_{\mathcal{F}_{\mathbb{B},\varphi(n)}}, \sigma(e_p))$.*

Proof. Since $\mathbb{B}$ is realized we know that $msg_{\mathbb{B}}(n') \in D(P_{\mathcal{F}_{\mathbb{B},n'}}, \mathcal{S}_{\mathcal{F}_{\mathbb{B},n'}})$. A weaker constraint is that $msg_{\mathbb{B}}(n') \in D(P_{\mathcal{F}_{\mathbb{B},n'}}, \mathcal{S}_{\mathcal{F}_{\mathbb{B},\varphi(n)}})$. By Proposition 4.7, p' must not be a critical path. Since Sol4 does not hold, if e_p is an encryption $\{\!|b|\!\}_u$ we know that $\sigma(u) \notin \mathcal{S}_{\mathcal{F}_{\mathbb{B},\varphi(n)}}$. Therefore, either $\sigma(e_p) \in \mathrm{Cl}^{\downarrow}(P_{\mathcal{F}_{\mathbb{B},n'}}, \mathcal{S}_{\mathcal{F}_{\mathbb{B},\varphi(n)}})$ or p visits $\mathrm{Fr}(P_{\mathcal{F}_{\mathbb{B},n'}}, \mathcal{S}_{\mathcal{F}_{\mathbb{B},\varphi(n)}})$. In the former case, $\sigma e_p \in \mathrm{Esc}(P_{\mathcal{F}_{\mathbb{B},n'}}, \mathcal{S}_{\mathcal{F}_{\mathbb{B},\varphi(n)}}, \sigma(e_p))$ and thus in either case p visits $\mathrm{Esc}(P_{\mathcal{F}_{\mathbb{B},n'}}, \mathcal{S}_{\mathcal{F}_{\mathbb{B},\varphi(n)}}, \sigma(e_p))$.

□

Lemma 12.13 (Existence of a good augmentation candidate)**.** *Let $\mathbb{A}^\bullet$ be an unrealized skeleton and let $\mathbb{B}^\bullet$ be a realized skeleton with $\mathbb{A}^\bullet \xrightarrow{\varphi,\sigma} \mathbb{B}^\bullet$ with $\lambda = (\varphi, \sigma)$. Let Sol2 hold, but let neither Sol3 nor Sol4 hold.*

Then there exists a $s_{\mathbb{B}}$ and a 4-tuple (ρ, i, π, tt) such that conditions WFC1, WFC2, WFC3, FC1, FC2, and FC3 are satisfied.

Proof. We will show that if Sol2 and neither Sol3 nor Sol4, then $((\mathbb{A}^\bullet, \mathsf{Id}), \lambda, \mathbb{B}'^\bullet, \mathbb{A}^\circ) \in \mathrm{DCohGAC}_{n,p}$, however this is far from straightforward. In brief, the problem is that while Sol2 establishes that there is some transmission outside the escape set, that does not make that transmission

a candidate for augmentation now. What we need to find is a transmission outside the escape set that can serve as a "transforming node" from the current escape set: a *first* new transmission.

Let $t = e_p$ and let $t' = \sigma(t)$. Let $E = \mathrm{Esc}(\mathcal{F}_{\mathbb{A},n}, t)$ and let $E' = \sigma(E)$. We define a sequence of tuples $(T_i, t_i, \pi_i, \pi'_i, n_i)$ for $0 \leq i \leq k$ with the following properties:

1. For $i > 0, T_i = E' \setminus \{t_1, \ldots, t_i\}$, and $t_i \in T_{i-1}$.
2. For $i > 0$, π'_i is a prefix of π_i.
3. (t_i, π_i) is a carried path that ends at t', and for $i > 0$, the endpoint of (t_i, π'_i) is not in $\sigma(\mathrm{Targ}(E, t))$.
4. For all $i \geq 0, n_i \prec \varphi(n)$.
5. There is a carried path from $msg_{\mathbb{B}}(n_i)$ ending at t' that does not visit T_i.

Note that because E' is finite, only a finite sequence can satisfy property 1.

To define the sequence, we let $T_0 = E'$. Since Sol2 holds we know there is a carried path from an element of $T_{\mathcal{F}_{\mathbb{A},n}}$ to t' that does not visit T_0; let n_0 be a any node transmitting such an element, let $t_0 = t'$, and let $\pi_0 = \pi'_0 = \langle\rangle$. Note that $n_0 \neq \varphi(n)$ because $\varphi(n)$ is a reception. Note that properties 1, 2, and the latter part of 3 are trivial for $i = 0$, and properties 4 and 5 and the first part of 3 are easily observed to be true for $i = 0$.

Let n'_i be a minimal node such that $msg_{\mathbb{B}}(n'_i)$ contains a carried path ending at t' that does not visit T_i.

Claim 1. Node n'_i must contain a transmission event.

Proof of Claim 1. If n'_i is a reception then since $\mathbb{B}$ is realized and Sol4 does not hold, by Lemma 12.12, all carried paths ending at t' in $msg_{\mathbb{B}}(n'_i)$ must visit $\mathrm{Esc}(P_{\mathcal{F}_{\mathbb{B},n'_i}}, \mathcal{S}_{\mathcal{F}_{\mathbb{B},\varphi(n)}}, t')$. Since there is a path ending at t' in $msg_{\mathbb{B}}(n'_i)$ that does not visit T_i, we must conclude that the first element, e, of $\mathrm{Esc}(P_{\mathcal{F}_{\mathbb{B},n'_i}}, \mathcal{S}_{\mathcal{F}_{\mathbb{B},\varphi(n)}}, t')$ visited by that path is not in T_i, and that e contains a carried path ending at t' that does not visit T_i. However, e can only be in $\mathrm{Esc}(P_{\mathcal{F}_{\mathbb{B},n'_i}}, \mathcal{S}_{\mathcal{F}_{\mathbb{B},\varphi(n)}}, t')$ if an earlier node n' transmitted a term with a carried

path q visiting e, such that q does not visit T_i after it visits e, and such that every plaintext edge q traverses before e is one for which the inverse key is in $\mathcal{S}_{\mathcal{F}_{\mathbb{B},\varphi(n)}}$. If q visits T_i it must do so before it visits e, but then it would visit T_i at a term $\sigma(\{\!|b|\!\}_u)$ for which $\mathsf{inv}(u) \in \mathcal{S}_{\mathcal{F}_{\mathbb{B},\varphi(n)}}$. Since Sol3 does not hold this cannot happen, so n' is an earlier node than n'_i such that $msg_{\mathbb{B}}(n')$ contains a carried path ending at t' that does not visit T_i.

□

Define a carried path $p_i = (msg_{\mathbb{B}}(n'_i), \alpha_i)$ to t' as follows:

1. If there is a carried path from $msg_{\mathbb{B}}(n'_i)$ to t' that does not visit E', let p_i be such a path.

2. If there is a carried path from $msg_{\mathbb{B}}(n'_i)$ to t' that visits E' but does not visit T_i, it must visit E' at a point t_j for $1 \leq j \leq i$. Let α_i be such that p_i visits t_j and such that π_j is a suffix of α_i.

Let $n'_i = (s_i, h_i)$ and let ρ_i be the role associated with s_i in $\mathcal{B}$. Let β_i be the maximal prefix of α_i such that $(C_{\rho_i}(h_i), \beta_i)$ is well-defined. There are three cases.

1. If $\beta_i = \alpha_i$ then let $tt = t$ and terminate the sequence, that is, set $k = i$.

   Otherwise, $(C_{\rho_i}(h_i), \beta_i)$ ends at a variable m_i of sort MESG.

2. If $(msg_{\mathbb{B}}(n'_i), \beta_i)$ ends at an element of $\sigma(\mathrm{Targ}(E, t))$, let $tt \in \mathrm{Targ}(E, t)$ such that $\sigma(tt)$ is the endpoint of $(msg_{\mathbb{B}}(n'_i), \beta_i)$ and set $k = i$.

3. Otherwise, $(msg_{\mathbb{B}}(n'_i), \beta_i)$ ends at a non-element of $\sigma(\mathrm{Targ}(E, t))$. Because ρ_i meets condition 3 of Definition 6.1, m_i must be acquired in ρ_i. Let h'_i be the node at which m_i first occurs in C_{ρ_i}, and let $(C_{\rho_i}(h'_i), \gamma_i)$ be a carried path ending at m_i. Note that $h'_i < h_i$. Let $n''_i = (s_i, h'_i)$, and note that $n''_i \prec n'_i \preceq n_i$. Since $\sigma_{\mathcal{B}(s_i)}(m_i)$ is both the endpoint of $(msg_{\mathbb{B}}(n'_i), \beta_i)$ and the endpoint of $(msg_{\mathbb{B}}(n''_i), \gamma_i)$, we may conclude that $q_i = (msg_{\mathbb{B}}(n''_i), \gamma_i \frown (\alpha_i - \beta_i))$ is a carried path ending at t'. By the minimality of n'_i, q_i visits T_i; let δ_i be the largest prefix of $(\gamma_i \frown (\alpha_i - \beta_i))$ such that $msg_{\mathbb{B}}(n''_i) @ \delta_i = t_{i+1} \in T_i$.

   Note that if γ_i is a prefix of δ_i, then $\sigma_{\mathcal{B}(s_i)}(m_i) @ (\delta_i - \gamma_i) \in T_i$ and therefore, p_i visits T_i, which contradicts our earlier assumption. Therefore, δ_i is a proper prefix of γ_i.

Let $\pi'_{i+1} = (\gamma_i - \delta_i)$. Let $\pi_{i+1} = \pi'_{i+1} \frown (\alpha_i - \beta_i)$. Let $T_{i+1} = E' \setminus \{t_1, \ldots, t_{i+1}\}$. Since Sol3 does not hold, by Lemma 12.11 we know that $t_{i+1} \in T_i \subseteq E' \subseteq \mathrm{Esc}(\mathcal{F}_{\mathbb{B},\varphi(n)}, t')$. Since we know $t_{i+1} \neq t'$, t_{i+1} must be in $\mathrm{Fr}(P_{\mathcal{F}_{\mathbb{B},\varphi(n)}}, \mathcal{S}_{\mathcal{F}_{\mathbb{B},\varphi(n)}})$. Therefore, t_{i+1} must be the endpoint of a maximal $\mathcal{S}_{\mathcal{F}_{\mathbb{B},\varphi(n)}}$-decryptable path beginning at an element $t^\star$ of $P_{\mathcal{F}_{\mathbb{B},\varphi(n)}}$. Since $t^\star$ cannot be an atom, a variable or sort MESG, or a tag, $t^\star$ must be a transmission, and therefore there is a node $n_{i+1} \prec \varphi(n)$ that transmits $t^\star$.

Claim 2. When we are in case 3 above, $(T_{i+1}, t_{i+1}, \pi_{i+1}, \pi'_{i+1}, n_{i+1})$ satisfy properties 1-5.

Proof of Claim 2. Consider the properties one at a time.

1. We chose $T_{i+1} = E' \setminus \{t_1, \ldots, t_{i+1}\}$, and chose $t_{i+1} \in T_i$.

2. Here, π_{i+1} was explicitly described as an extension of π'_{i+1}, so this is obvious.

3. First, note that (t_{i+1}, π_{i+1}) is a carried path, since it is a subpath of $(msg_{skelB}(n''_i), (\delta_i \frown (\alpha_i - \beta_i)))$, which we know is a carried path. Also, (t_{i+1}, π_{i+1}) ends at t':

$$\begin{aligned}
t_{i+1} \mathbin{@} \pi_{i+1} &= (msg_{\mathbb{B}}(n''_i) \mathbin{@} \delta_i) \mathbin{@} ((\gamma_i - \delta_i) \frown (\alpha_i - \beta_i)) \\
&= msg_{\mathbb{B}}(n''_i) \mathbin{@} (\delta_i \frown (\gamma_i - \delta_i) \frown (\alpha_i - \beta_i)) \\
&= msg_{\mathbb{B}}(n''_i) \mathbin{@} (\gamma_i \frown (\alpha_i - \beta_i)) \\
&= \sigma_{\mathcal{B}(s_i)}(m_i) \mathbin{@} (\alpha_i - \beta_i) \\
&= msg_{\mathbb{B}}(n'_i) \mathbin{@} (\beta_i \frown (\alpha_i - \beta_i)) \\
&= msg_{\mathbb{B}}(n'_i) \mathbin{@} \alpha_i \\
&= t'
\end{aligned}$$

   The endpoint of $(t_{i+1}, \pi'_{i+1}) = msg_{\mathbb{B}}(n'_i) \mathbin{@} \beta_i$, which we know is not an element of $\sigma(\mathrm{Targ}(E, t))$ since we are in case 3.

4. We have already noted that $n_i \prec \varphi(n)$.

5. There is a carried path from $msg_{\mathbb{B}}(n_i)$ ending at t' that does not visit T_i, namely the one whose maximal decryptable subpath is t_i.

□

Now we prove that, $(\rho_k, h_k, \beta_k, tt)$ is a good augmentation candidate with $s_{\mathbb{B}} = s_k$ and $\pi^{tt} = (\alpha_k - \beta_k)$:

- WFC1. If the endpoint of $(C_{\rho_k}(h_k), \beta_k)$ is not a variable of sort MESG then either $tt = t$ or $(msg_{\mathbb{B}}(n'), \alpha_k)$ would not be a carried path.
- FC1 is obvious.
- FC2. $n'_k = (s_{\mathbb{B}}, h_k) \prec \varphi(n)$ by property 5 of our sequence.
- FC3 is true by construction in the latter case; when $\alpha_k = \beta_k$, the endpoint is $t' = \sigma(t) = \sigma(tt)$.
- WFC2 is guaranteed by the minimality of n'_k.
- WFC3. Note that (tt, π^{tt}) is a carried path ending at t. Note further that $(msg_{\mathbb{B}}(n'_k), (\beta_k) \frown (\alpha_k - \beta_k)) = p_k$. Thus, if p_k does not visit E', we meet WFC3a.

  Otherwise, p_k visits some t_j, and π_j is a suffix of α_k, so we can write $\alpha_k = \alpha'_k \frown \pi_j$. Let $\pi' = \alpha'_k \frown \pi'_j$. We know that $(msg_{\mathbb{B}}(n'_k), \pi')$ traverses E' since it visits $t_j \in E'$, and by Lemma 12.11, $(msg_{\mathbb{B}}(n'_k), \pi')$ traverses $\mathrm{Esc}(\mathcal{F}_{\mathbb{B},\varphi(n)}, t')$. We also know, by property 3 of our sequence, that the endpoint of $(msg_{\mathbb{B}}(n'_k), \pi')$, which is the same as the endpoint of (t_j, π'_j), is not in $\sigma(\mathrm{Targ}(E, t))$, so we meet WFC3b.

This completes the proof of Lemma 12.13. □

12.6 Proof of Lemma 10.11

Lemma 10.11. (Pre-cohort completeness) *Let $\mathbb{A}^\circ$ be an unrealized skeleton and let (n, p) be any test of $\mathbb{A}^\circ$. Then $(\mathbb{A}^\circ, \mathsf{Id}) : [\![\mathrm{Coh} \xrightarrow{\mathfrak{P}^{PP_{n,p}}} \mathrm{PCoh}_{n,p}]\!]$.*

Proof. Suppose $((\mathbb{A}^\bullet, \mathsf{Id}), \lambda, \mathbb{B}^\circ, \mathbb{A}^\circ) \in \mathrm{Coh}$, with $\mathbb{B}'^\bullet$ satisfying condition C3. By Lemma 12.10, one of the following three cases applies:

1. $((\mathbb{A}^\bullet, \mathsf{Id}), \lambda, \mathbb{B}'^\bullet, \mathbb{A}^\circ) \in \mathrm{DCohTD}_{n,p}$.

   In this case, by Lemma 12.5, there exists an $\mathsf{f} \in \mathfrak{c}_{n,p}(\mathbb{A}^\circ) \subset \mathfrak{P}_{n,p}$ and a λ' such that $((\mathsf{f}(\mathbb{A}^\bullet), \mathsf{f}), \lambda', \mathbb{B}'^\bullet, \mathbb{A}^\circ) \in \mathrm{DPCoh}_{n,p}$.

2. $((\mathbb{A}^\bullet, \mathsf{Id}), \lambda, \mathbb{B}'^\bullet, \mathbb{A}^\circ) \in \mathrm{DCohGAC}_{n,p}$, or

   In this case, by Lemma 12.9, there exists an $\mathsf{f} \in \mathfrak{a}_{n,p}(\mathbb{A}^\circ) \subset \mathfrak{P}_{n,p}$ and a λ' such that $((\mathsf{f}(\mathbb{A}^\bullet), \mathsf{f}), \lambda', \mathbb{B}'^\bullet, \mathbb{A}^\circ) \in \mathrm{DPCoh}_{n,p}$.

   In either case 1 or case 2, we have $\mathsf{f} \in \mathfrak{P}_{n,p}$ and a λ' such that $((\mathsf{f}(\mathbb{A}^\bullet), \mathsf{f}), \lambda', \mathbb{B}'^\bullet, \mathbb{A}^\circ) \in \mathrm{DPCoh}_{n,p}$. Therefore, $((\mathsf{f}(\mathbb{A}^\bullet), \mathsf{f}), \lambda', \mathbb{B}^\circ, \mathbb{A}^\circ) \in \mathrm{PCoh}_{n,p}$ with $\mathbb{B}'^\bullet$ satisfying condition P3.

3. $((\mathbb{A}^\bullet, \mathsf{Id}), \lambda, \mathbb{B}^\circ, \mathbb{A}^\circ) \in \mathrm{CohCVD}_{n,p}$.

   In this case, by Lemma 12.9, there exists an $\mathsf{f} \in \mathfrak{l}_{n,p}(\mathbb{A}^\circ) \subset \mathfrak{P}_{n,p}$ and a λ' such that $((\mathsf{f}(\mathbb{A}^\bullet), \mathsf{f}), \lambda', \mathbb{B}^\circ, \mathbb{A}^\circ) \in \mathrm{PCoh}_{n,p}$.

We need only prove that $(\mathbb{A}^\circ, \mathsf{f}) \in PP_{n,p}$ to complete the theorem. We know that $\mathsf{f}(\mathbb{A}^\circ)$ is a preskeleton by Theorem 8.16. We also know that $(\mathbb{A}^\circ, \mathsf{f}) \in SF_{n,p}$ by property P6, with $\mathsf{f}(\mathbb{A})$ itself as the intermediate factoring. Finally, all three of the lemmas established that $(\mathbb{A}^\circ, \mathsf{f}) \in HC$. This completes the proof. □

13 Enumerability

In this section, we go beyond the concept of completeness and show that CPSA *enumerates* covering realized skeletons.

Theorem 13.1 (CPSA enumerates)**.** *For all $(\mathbb{A}', \lambda) \in [\![\mathbb{A}^\circ]\!]$, there exists an $n \geq 0$ such that if S is the set of skeletons* CPSA *produces after n setwise reduction operations, there is a realized $\mathbb{B}'^\circ \in S$ and a homomorphism $\mathbb{A}^\circ \xrightarrow{\lambda'} \mathbb{B}^\circ$ such that $(\mathbb{A}', \lambda'') \in [\![\mathbb{B}^\circ]\!]$ and $\lambda = \lambda'' \circ (\lambda'|_{\mathrm{Rmv}(\mathbb{A}^\circ)})$.*

To accomplish this cleanly, we need to modify the CPSA algorithm slightly. Our approach to proving enumerability is to prove that CPSA maintains a nodewise-injective map to the desired target through the cohort completeness arguments. However, to ensure that we have a nodewise-injective map initially, we must modify CPSA to do all possible merges of strands in the initial input first.

Definition 13.2 (Merge-all suite)**.** *Let*

$$\mathfrak{X}(\mathbb{A}^\circ) = \{\mathsf{Id}\} \cup \{\mathsf{Comp}_{s_1,s_2} \circ \mathsf{Sub}_\sigma | s_1 \neq s_2 \in I_\mathbb{A}, \sigma \in U_{s_1,s_2}\},$$

where U_{s_1,s_2} is a set of most general unifiers of $\Theta(s_1)|i$ and $\Theta(s_2)|i$ for $i = min(|\Theta(s_1)|, |\Theta(s_2)|)$.

Then the merge-all suite is defined to be

$$\mathfrak{Ma}(\mathbb{A}^\circ) = \mathfrak{X}(\mathbb{A}^\circ)^{|I_\mathbb{A}|}.$$

Lemma 13.3 (Initial nodewise injectivity)**.** *If there is a homomorphism from the user's input $\mathbb{A}^\circ \xrightarrow{\lambda} \mathbb{B}^\circ$ with $\mathbb{B}^\circ$ realized, there is an $\mathsf{f} \in (\mathfrak{S} \circ \mathfrak{Ma})(\mathbb{A}^\circ)$ such that $\mathsf{f}(\mathbb{A}^\circ)$ is a skeleton and there is a nodewise injective λ' such that $\mathsf{f}(\mathbb{A}^\circ) \xrightarrow{\lambda'} \mathbb{B}^\circ$ and $\lambda = \lambda' \circ \Lambda_{\mathsf{f}(\mathbb{A})}$.*

Proof. Simply put, for every pair of strands in $\mathbb{A}$ that are unified in $\mathbb{B}$, we merge one pair of these strands at a time in each application of $\mathfrak{X}$ until no more are needed. After that, we choose $\mathsf{Id} \in \mathfrak{X}$. The result is an operator that merges all strands in $\mathbb{A}^\circ$ that are merged in $\mathbb{B}^\circ$. After this, the remaining factorization of the original map must be nodewise injective.

By Lemma 10.15, λ' is structure-preserving. Note that $\mathsf{f}(\mathbb{A})$ has only inherited unique origination assumptions from $\mathbb{A}$ under $\Lambda_{\mathsf{f}(\mathbb{A})}$.

□

Lemma 13.4. *If $\mathbb{A} \overset{\lambda}{\dashrightarrow} \mathbb{B}$ and λ is nodewise-injective, and $\mathsf{f} = \mathsf{OE}$ or $\mathsf{f} = \mathsf{Sub}_\sigma$ for any σ, such that $\mathsf{f}(\mathbb{A}) \overset{\lambda'}{\dashrightarrow} \mathbb{B}$ where $\lambda = \lambda' \circ \Lambda_\mathsf{f}$, λ' is nodewise-injective.*

Proof. For all such operators, the nodes in $\mathsf{f}(\mathbb{A})$ are in one-to-one correspondance with the nodes in $\mathbb{A}$. Thus, if two nodes $\mathsf{f}(n_1)$ and $\mathsf{f}(n_2)$ in $\mathsf{f}(\mathbb{A})$ map to the same node in $\mathbb{B}$, then so did n_1 and n_2 in $\mathbb{A}$, which we assumed was not the case. □

This already shows that much of the proof of completeness holds if we add the requirement that λ be nodewise-injective to all the coverage properties. What remains is to handle those cases where we use the compression or augmentation primitive operators. Fortunately, these come up in only three places.

1. In skeletonization, we use the compression operator in the merging suite $\mathfrak{m}_{n,p}$. However, the precondition for Lemma 11.7 renders the merging suite moot when we require a nodewise injective λ, because it requires that $\varphi(n) = \varphi(n')$ for distinct nodes n, n'.

2. In listener augmentation, we use the augmentation operator to add a listener. If the listener were to map to a node already in the image of $\mathbb{A}$, this would break node injectivity. Fortunately during listener augmentation, we adjust the target $\mathbb{B}$ to include a new listener specifically to be the image of the new listener added to $\mathbb{A}$, so node-injectivity is always preserved.

3. In regular augmentation, we may sometimes augment with a strand whose image is already in $\lambda(\mathbb{A})$. Specifically, suppose that we have $\mathsf{f} = \mathsf{Sub}_{\sigma_1} \circ \mathsf{Aug}_{n,\rho,i,\sigma_0 \circ FR(\mathbb{A},\rho,i),NAME(\mathbb{A})}$ and the corresponding λ', a homomorphism from $\mathsf{f}(\mathbb{A}^\bullet)$ to $\mathbb{B}^\bullet$, but that λ' is not nodewise-injective. Since λ was nodewise-injective, no node collision can occur unless one of the nodes is in the new strand $NAME(\mathbb{A})$. Suppose $\varphi'(NAME(\mathbb{A})) = \varphi'(s)$ for some other strand $s \in I_{\mathbb{A}}$. Note that σ' is a unifier of s with $NAME(\mathbb{A})$ up to the minimum of their heights; let σ_2 be a most general unifier of those terms more general than σ'. Then consider $\mathsf{f}' = \mathsf{Comp}_{s,NAME(\mathbb{A})} \circ \mathsf{Sub}_{\sigma_2} \circ \mathsf{f}$. Clearly, there is a λ'' such that $\lambda' = \lambda'' \circ \Lambda_{\mathsf{Comp}_{s,NAME(\mathbb{A})} \circ \mathsf{Sub}_{\sigma_2}(\mathbb{A})}$. $\mathsf{f}' \in \mathfrak{d}_{n,p}(\mathbb{A}^\circ)$ by our definition of the deorigination suite.

   $((\mathsf{f}'(\mathbb{A}^\bullet), \mathsf{f}'), \lambda'', \mathbb{B}^\bullet, \mathbb{A}^\circ) \in \mathrm{DPCoh}_{n,p}$. Most conditions are easy to establish given that we already know $((\mathsf{f}(\mathbb{A}^\bullet), \mathsf{f}), \lambda', \mathbb{B}^\bullet, \mathbb{A}^\circ) \in \mathrm{DPCoh}_{n,p}$; the only one that is non-trivial is that λ'' is structure-preserving. The proof that λ'' must be structure-preserving essentially follows the argument that merging satisfies condition S5 in Lemma 11.7.

   Finally, $\mathbb{A}^\circ \xrightarrow{\Lambda_{\mathsf{f}'(\mathbb{A})}} \mathsf{f}'(\mathbb{A}^\circ)$. Again it is obvious that $\Lambda_{\mathsf{f}'(\mathbb{A})}$ is structure-preserving, and by Lemma 7.25, it preserves points of origination. This allows us to conclude that $(\mathbb{A}^\circ, \mathsf{Id}) : [\mathrm{NIDCohGAC}_{n,p} \xRightarrow{(\mathfrak{a}_{n,p} \cup \mathfrak{d}_{n,p})^{HC_{n,p}}} \mathrm{NIPCoh}_{n,p}]$, where NIDCohGAC and NIPCoh, respectively, are the equivalent coverage properties with the additional requirement that the map λ be nodewise-injective.

   Since $\mathfrak{a}_{n,p} \cup \mathfrak{d}_{n,p} \subset \mathfrak{P}_{n,p}$, this weakened statement is still strong enough for use in the proof of Lemma 10.11.

Finally, we prove that maintaining nodewise-injective maps guarantees a finite number of cohort steps to a realized skeleton. Note that since we maintain nodewise-injective maps, we can use only finitely many operators that add nodes to our current skeleton while maintaining coverage of the

target realized skeleton. There are only finitely many missing orderings, so again we can only use finitely many operators that add additional orderings. Since no cohort step consists of only Id, the only way we can fail to reach the target (or something covering it) in finitely may steps is for there to be an infinite sequence of substitutions, each more specific than the last, factoring a specific substitution. And to be more specific, every subsequent substitution is a unification. However, it is known (see Lemma A.10) that this cannot be the case for our algebra. Thus, either we reach our target in a finite number of steps, or we reach something realized before that, in which case we reach something covering our target in finitely many steps.

References

[1] A. Armando, D. Basin, Y. Boichut, Y. Chevalier, L. Compagna, J. Cuellar, P. Hankes Drielsma, P.C. Hem, O. Kouchnarenko, J. Mantovani, S. Mdersheim, D. von Oheimb, M. Rusinowitch, J. Santiago, M. Turuani, L. Vigan, and L. Vigneron. The avispa tool for the automated validation of internet security protocols and applications. In Kousha Etessami and Sriram K. Rajamani, editors, *Computer Aided Verification*, volume 3576 of *Lecture Notes in Computer Science*, pages 135–165. Springer Berlin / Heidelberg, 2005.

[2] A. Armando and L. Compagna. Sat-based model checking for security protocols analysis. *International Journal of Information Security*, 7(1):3–32, 2008.

[3] M. Backes, S. Lorenz, M. Maffei, and K. Pecina. The CASPA tool: Causality-based abstraction for security protocol analysis. In A. Gupta and S. Malik, editors, *CAV*, volume 5123 of *Lecture Notes in Computer Science*, pages 419–422. Springer, 2008.

[4] Bruno Blanchet. An efficient cryptographic protocol verifier based on prolog rules. In *14th IEEE Computer Security Foundations Workshop (CSFW-14 2001)*, pages 82–96, 2001.

[5] Y. Boichut, P.C. Heam, O. Kouchnarenko, and F. Oehl. Improvements on the genet and klay technique to automatically verify security protocols. In *Proc. International Workshop on Automatic Verification of Infinite-State Systems (AVIS'04)*, 2004.

[6] R. Corin and S. Etalle. An improved constraint-based system for the verification of security protocols. In *Proc. 9th International Static Analysis Symposium (SAS)*, volume 2477 of *Lecture Notes in Computer Science*, pages 326–341. Springer, September 2002.

[7] C.J.F. Cremers. *Scyther - Semantics and Verification of Security Protocols*. Ph.D. dissertation, Eindhoven University of Technology, 2006.

[8] C.J.F. Cremers. The Scyther Tool: Verification, falsification, and analysis of security protocols. In *Computer Aided Verification, 20th International Conference, CAV 2008, Princeton, USA, Proc.*, volume 5123/2008 of *Lecture Notes in Computer Science*, pages 414–418. Springer, 2008.

[9] Shaddin F. Doghmi, Joshua D. Guttman, and F. Javier Thayer. Searching for shapes in cryptographic protocols (extended version). `http://eprint.iacr.org/2006/435`, November 2006.

[10] Shaddin F. Doghmi, Joshua D. Guttman, and F. Javier Thayer. Completeness of the authentication tests. In J. Biskup and J. Lopez, editors, *European Symposium on Research in Computer Security (ESORICS)*, number 4734 in LNCS, pages 106–121. Springer-Verlag, September 2007.

[11] Shaddin F. Doghmi, Joshua D. Guttman, and F. Javier Thayer. Searching for shapes in cryptographic protocols. In *Tools and Algorithms for Construction and Analysis of Systems (TACAS)*, number 4424 in LNCS, pages 523–538. Springer, March 2007. Extended version at `http://eprint.iacr.org/2006/435`.

[12] Daniel Dolev and Andrew Yao. On the security of public-key protocols. *IEEE Transactions on Information Theory*, 29:198–208, 1983.

[13] Santiago Escobar, Catherine Meadows, and Jos Meseguer. Maude-NPA: Cryptographic protocol analysis modulo equational properties. In Alessandro Aldini, Gilles Barthe, and Roberto Gorrieri, editors, *Foundations of Security Analysis and Design V*, volume 5705 of *Lecture Notes in Computer Science*, pages 1–50. Springer Berlin / Heidelberg, 2009.

[14] Joshua D. Guttman and F. Javier Thayer. Authentication tests and the structure of bundles. *Theor. Comput. Sci.*, 283(2):333–380, 2002.

[15] G. Lowe. Breaking and fixing the needham-schroeder public-key protocol using FDR. In *Proc. 2nd International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, volume 1055 of *Lecture Notes in Computer Science*, pages 147–166, 1996.

[16] C. Meadows. The NRL protocol analyzer: An overview. *Journal of Logical Programming*, 26(2):113–131, 1996.

[17] John D. Ramsdell and Joshua D. Guttman. CPSA: A cryptographic protocol shapes analyzer. In *Hackage*. The MITRE Corporation, 2009. `http://hackage.haskell.org/package/cpsa`.

[18] John D. Ramsdell, Joshua D. Guttman, and Paul D. Rowe. *The CPSA Specification: A Reduction System for Searching for Shapes in Cryptographic Protocols*. The MITRE Corporation, 2009. In `http://hackage.haskell.org/package/cpsa` source distribution, `doc` directory.

[19] Alan Robinson and Andrei Voronkov. *Handbook of Automated Reasoning*. The MIT Press, 2001.

[20] F. Javier Thayer, Jonathan C. Herzog, and Joshua D. Guttman. Strand spaces: Proving security protocols correct. *Journal of Computer Security*, 7(1), 1999.

[21] M. Turuani. The CL-Atse protocol analyser. In *Proc. RTA '06*, volume 4098 of *Lecture Notes in Computer Science*, pages 227–286. Springer, August 2006.

A The `cowt` Algorithm

This section describes the algebra specific part of the computation used in regular augmentation and displacement.

First we recall the definition of the regular augmentation suite:

Definition 9.2 (Regular augmentation suite)

$$\mathfrak{a}_{n,p}(\mathbb{A}^\circ) = \{\mathsf{Sub}_{\sigma_1} \circ \mathsf{Aug}_{n,\rho,i,\sigma_0 \circ FR(\mathbb{A},\rho,i),s^\star}\}$$

where $\sigma_0 \in S_0, \sigma_1 \in S_1, s^\star = NAME(\mathbb{A})$ *and* $\rho, i, \sigma_0, \sigma_1, S_0, S_1$ *are as defined below*$\}$.

- $\mathcal{R} \in P$ *and* i *is such that* $C_{\mathcal{R}}(i)$ *is defined and is a send event. Let* $C = C_{\mathcal{R}}|i$.
- *There is a path* $pp \in \mathrm{CarPath}(C(i))$ *and a term* tt *such that either (i) the endpoint of* pp *is a variable not of sort* MESG *and* $tt = e_p$ *or (ii) the endpoint of* pp *is a variable of sort* MESG *and* $tt \in \mathrm{Targ}(\mathrm{Esc}(\mathcal{F}, e_p), e_p)$.
- S_0 *is a set of most general unifiers of* tt *with the endpoint of* $FR(\mathbb{A}, \rho, i)(pp)$.
- S_1 *is a set of most general maps* σ_1 *such that for all* $i' < i$ *and for all paths* $p' \in \mathrm{CarPath}(C(i'))$, *if the endpoint of* $\sigma_1((\sigma_0 \circ FR(\mathbb{A}, \rho, i))(p'))$ *is* $\sigma_1(e_p)$ *then* $\sigma_1(p')$ *visits an element of* $\sigma_1(\mathrm{Esc}(\mathcal{F}_{\mathbb{A},n}, e_p))$.

Note that it is not obvious that S_1 exists, let alone that there is a finite set of such maps we can calculate efficiently.

Definition A.1 (Carried only within)**.** *Message* t *is* carried only within *set* T *in* t', *if for all carried paths* p *ending at* t *in* t', p *visits* T.

A message t *is carried only within set* T *in a set of messages* T' *if for all* $t' \in T'$, t *is carried only within* T *in* t'.

In calculating the regular augmentation suite, we must find a set S_1 of most general maps σ_1 such that $\sigma_1(e_p)$ is carried only within $\sigma_1(\mathrm{Esc}(\mathcal{F}_{\mathbb{A},n}, e_p))$ in $\{(\sigma_0 \circ FR(\mathbb{A}, \rho, i))(C|_{i-1})\}$.

Definition A.2 (Carried only within problem)**.** *A* carried only within problem *is a triple* (t, T, T').

Definition A.3 (Carried only within problem solution)**.** *The solution to a carried only within problem* (t, T, T') *is a complete set of most general unifiers* S *such that for every* $\sigma \in S$, $\sigma(t)$ *is carried only within* $\sigma(T)$ *in* $\sigma(T')$.

The notion of a carried only within problem and its solution allows us to describe the CPSA approach to computing the regular augmentation suite as its own algorithm.

A *unification problem* is a finite set

$$E = \{t_1 \stackrel{?}{=} t'_1, \ldots, t_n \stackrel{?}{=} t'_n\},$$

and a *unifier* of E is a substitution σ such that $\sigma(t_1) \equiv \sigma(t'_1), \ldots, \sigma(t_n) \equiv \sigma(t'_n)$. Definitions in this section follow [19, Chapter 9].

When solving a unification problem E, we rely on the fact that the order in which the equations are solved is irrelevant in the following sense. Let $\mathcal{C}(E)$ be a finite complete set of unifiers for E. Let $E = E_0 \cup E_1$ be a decomposition of E, and let $S = \{\sigma_1 \circ \sigma_0 \mid \sigma_0 \in \mathcal{C}(E_0) \wedge \sigma_1 \in \mathcal{C}(\sigma_0(E_1))\}$. Then S is a finite complete set of unifiers for E. Decompositions $E = E_0 \cup E_1$ in this section often are such that $E_0 \subset E$, and $E_1 = E$. Thus, solving a unification problem can be done from a function *unify* that, on input t and t', finds a set of most general substitutions σ such that $\sigma(t) = \sigma(t')$.

In what follows, $\mathcal{C}(\{\}) = \{\mathsf{Id}_{\mathfrak{A}}\}$.

A carried only within solution cannot be directly computed. Given terms t and t', the *unify* function finds a most general set of substitutions σ such that $\sigma(t) \equiv \sigma(t')$, however, the set of carried paths ending at t may become larger after we apply a unifying substitution.

The remainder of this section describes an iterative procedure that breaks the cyclic dependencies. Each step of the iteration improves an approximation of a solution to the problem.

Definition A.4 (Carried only within at a substitution)**.** *Message t is* carried only within T *in t'* at substitution σ *if for all carried paths p ending at t in t', $\sigma(p)$ visits $\sigma(T)$.*

Each step in the iterative procedure involves finding subsequently more specific substitutions such that t is COW T in t' at σ for each $t' \in T'$. The sense in which each step approximates the solution is captured by the following lemmas.

The algorithm uses specific terms rather than equivalence classes of terms.

Lemma A.5. $\sigma(\mathrm{CarPath}(t, t')) \subseteq \mathrm{CarPath}(\sigma(t), \sigma(t'))$.

Proof. Let $p = (t', \pi)$ be a path that ends at t. Then $\sigma(p) = (\sigma(t'), \pi)$, which is a path that ends at $\sigma(t)$. Moreover, p is a carried path if and only if $\sigma(p)$ is. □

The case in which $t = x$ and $t' = \langle x, y \rangle$, and σ unifies x and y provides an example in which the subset relation is proper.

Lemma A.5 can be used to show why the problem of finding carried only within solutions is non-trivial. If $\sigma(t) \equiv \sigma(t')$ and $\sigma \trianglelefteq \sigma'$, then $\sigma'(t) \equiv \sigma'(t')$, however, if $\sigma(t)$ is COW $\sigma(T)$ in $\sigma(t')$ and $\sigma \trianglelefteq \sigma'$, one cannot conclude that $\sigma'(t)$ is COW $\sigma'(T)$ in $\sigma'(t')$, because by Lemma A.5, it is possible that $\mathrm{CarPath}(\sigma(t), \sigma(t')) \subsetneq \mathrm{CarPath}(\sigma'(t), \sigma'(t'))$.

The implementation must consider all possible solutions to the equations. It does so by operating on sets of terms and substitutions. Given solutions S to some other equations, $solve(T, T', S)$ extends them to include solutions to one pair from T and T'.

$$\begin{array}{l} solve(T, T', S) = \\ \quad \{\sigma' \mid t \in T, t' \in T', \sigma \in S, \sigma' \in unify_0(t, t', \sigma)\} \\ unify_0(t, t', \sigma) = \{\sigma' \circ \sigma \mid \sigma' \in unify(\sigma(t), \sigma(t'))\} \end{array}$$

An algebra module in CPSA exports $unify_0$, not $unify$, and substitution composition is intertwined with unification steps. Obviously, if $\sigma' \in unify_0(t, t', \sigma)$ then $\sigma \preceq \sigma'$.

The implementation combines the solutions for single equations by folding the substitutions produced by the *solve* function. The set-oriented version of the COW at a substitution function is:

$$\begin{array}{l} fold(t, T, t', \sigma) = \\ \quad fold_0(t, T, t', \sigma, \{\mathsf{Id}_{\mathfrak{A}}\}, \mathrm{CarPath}(\sigma(t), \sigma(t'))) \end{array}$$

$$\begin{array}{l} fold_0(t, T, t', \sigma, S, \{\}) = \{\sigma' \circ \sigma | \sigma' \in S\} \\ fold_0(t, T, t', \sigma, S, \{p\} \cup P) = \\ \quad fold_0(t, T, t', \sigma, solve(anc(\sigma(t'), p), \sigma(T), S), P) \end{array}$$

The *fold* function on (t, T, t', σ) is meant to return the set of substitutions $\{\sigma' \circ \sigma | \sigma(t)$ is COW $\sigma(T)$ in $\sigma(t')$ at $\sigma'\}$. However, we do not need to prove this behavior specifically to establish the theorems we want to prove. The important observation is that t being COW T in t' at σ is insufficient to guarantee that $\sigma(t)$ is COW $\sigma(T)$ in $\sigma(t')$.

Iterating the *fold* function can be used to find contractions. Potential contractions are in $cows(t, T, t')$, where

$$\begin{array}{l} cows(t, T, t') = \\ \quad cows_0(t, T, t', \mathsf{Id}_{\mathfrak{A}}) \end{array}$$

$$\begin{array}{l} cows_0(t, T, t', \sigma) = \\ \quad \textbf{if } \sigma(t) \text{ is COW } \sigma(T) \text{ at } \sigma(t') \textbf{ then} \\ \quad\quad \{\sigma\} \\ \quad \textbf{else} \\ \quad\quad \textbf{let } S = fold(t, T, t', \sigma) \textbf{ in} \\ \quad\quad \bigcup_{\sigma' \in S} cows_0(t, T, t', \sigma') \end{array}$$

We now show the *cows* function produces the unifiers that make up a carried only within solution. It may also produce non-minimal unifiers. An additional step is required to remove these unifiers.

The *cows* function terminates because each step in the iteration reduces the number of variables in the problem statement. Several lemmas are required to show termination.

Lemma A.6. *$\sigma' \in unify_0(t, t', \sigma)$ and $\sigma(t) \not\equiv \sigma(t')$ implies $\sigma \lhd \sigma'$.*

Proof. We know $\sigma \trianglelefteq \sigma'$. Assume the negation of the conclusion, that there is a substitution σ'' such that $\sigma = \sigma'' \circ \sigma'$. The first hypothesis implies $\sigma'(t) \equiv \sigma'(t')$, so $\sigma''(\sigma'(t)) \equiv \sigma''(\sigma'(t'))$, a contradiction. □

Lemma A.7. *$\sigma' \in fold(t, T, t', \sigma)$ and $\sigma(t)$ not* COW *$\sigma(T)$ in $\sigma(t')$ implies $\sigma \lhd \sigma'$.*

Proof. If $\sigma(t)$ not COW $\sigma(T)$ in $\sigma(t')$, there is some position p in $\mathrm{CarPath}(\sigma(t), \sigma(t'))$ such that no ancestor of $\sigma(t') @ p$ is equivalent to any member of $\sigma(T)$. We know $\sigma \trianglelefteq \sigma'$ (by observation, all outputs of *fold* extend the initial σ), so assume the negation of the conclusion, that there is a substitution σ'' such that $\sigma = \sigma'' \circ \sigma'$. By the fact that $\sigma' \in fold(t, T, t', \sigma)$ we know that t is COW T in t' at σ', so there is a $t'' \in T$ and a proper prefix p' of p such that $\sigma'(t'') \equiv \sigma'(t' @ p')$. Thus, $\sigma(t'') = \sigma''(\sigma'(t'')) \equiv \sigma''(\sigma'(t' @ p')) = \sigma(t' @ p')$ which contradicts what we know about p, as $\sigma(t' @ p') = \sigma(t') @ p'$ is an ancestor of $\sigma(t')$ equivalent to $\sigma(t'')$ which is in $\sigma(T)$.

□

Now we prove our three main results about *cows*. We give sufficient conditions to guarantee that *cows* terminates (Theorem A.8), we prove that *cows* gives answers with the property we want (Theorem A.9), and we prove that *cows* produces a complete set of such outputs (Theorem A.11).

Theorem A.8. *If the algebra $\mathfrak{A}$ has variable-reducing, finitary unification, the function $cows(t, T, t')$ terminates on all inputs.*

Proof. By an examination of *fold*, each substitution produced is a unification of a set of equations. By Lemma A.7, each substitution produced is strictly less general than $\mathsf{Id}_{\mathfrak{A}}$. Therefore, every substitution produced by *fold* is a non-trivial unification. Since every unification is variable-reducing, there is a maximum number of successive non-trivial unifications that can be applied

before further unification becomes impossible, namely, the number of variables appearing in the original t, T, and t'. Furthermore, since the unification is finitary, we know that every node of the tree of substitutions we explore has a finite branching factor. Thus, the entire tree of potential solutions is finite, so *cows* terminates. □

Theorem A.9. *$\sigma \in cows(t, T, t')$ implies $\sigma(t)$ is* COW *$\sigma(T)$ in $\sigma(t')$.*

Proof. We prove this by structural induction on the execution of *cows*. If $\sigma \in cows(t, T, t')$ and $cows_0$ outputs without recursing then $\sigma = \mathsf{Id}_{\mathfrak{A}}$ and the property holds by the required condition in the algorithm.

If $\sigma \in cows(t, T, t')$ and $cows_0$ outputs after recursing, then $\exists \sigma', \sigma''$ such that $\sigma = \sigma'' \circ \sigma'$ where $\sigma' \in S$ and $\sigma'' \in cows_0(t, T, t', \sigma')$. By inductive assumption, $\sigma''(\sigma'(t))$ is COW $\sigma''(\sigma'(T))$ in $\sigma''(\sigma'(t'))$. This proves that $\sigma(t)$ is COW $\sigma(T)$ in $\sigma(t')$. □

Lemma A.10. *For any strand space algebra with variable-reducing unification, the following holds: Let $\mathsf{Id}_{\mathfrak{A}} = \sigma_0 \trianglelefteq \sigma_1 \trianglelefteq \ldots$ be an infinite sequence of substitutions all generated over the same finite set of variables, such that for every σ_i, $\sigma_i \trianglelefteq \sigma$ and such that every σ_i' (where $\sigma_{i+1} = \sigma_i' \circ \sigma_i$) is a most general unification. Then for at most finitely many $i \geq 0$, $\sigma_i \triangleleft \sigma_{i+1}$.*

Proof. Let $\{x_1, \ldots, x_n\}$ be the finite set of variables over which the sequence of substitutions is defined. For each i, define v_i to be $|Vars(\sigma_i(x_1, \ldots, x_n))|$ with $v_0 = n$, and let $v = |Vars(\sigma(x_1, \ldots, x_n))|$. For each i, we know that σ_i' is a most general unification; if it is a unification of terms which are all already equivalent then σ_i' must be a renaming and then it cannot be the case that $\sigma_i \triangleleft \sigma_{i+1}$. If σ_i' is a unification of one or more pairs of non-equivalent terms, it is variable-reducing, so $v_i > v_{i+1}$. However, this sort of step can be taken at most $n - v$ times. □

Theorem A.11. *If the algebra $\mathfrak{A}$ has variable-reducing unification, $\sigma(t)$ is* COW *$\sigma(T)$ in $\sigma(t')$ implies there exists a substitution σ' such that $\sigma' \trianglelefteq \sigma$ and $\sigma' \in cows(t, T, t')$.*[2]

Proof. We aim to define a sequence of substitutions $\mathsf{Id}_{\mathfrak{A}} = \sigma_0 \trianglelefteq \sigma_1 \ldots \trianglelefteq \sigma_n \trianglelefteq \sigma$, such that $\sigma_n(t)$ is COW $\sigma_n(T)$ in $\sigma_n(t')$, where each σ_i is produced

[2]Note that $cows(t, T, t')$ is a well-defined set regardless of whether a computer could calculate it in a finite number of steps.

incrementally during the *cows* computation, where σ_n will serve as the σ' in the theorem.

By saying that σ_i is "produced incrementally during the *cows* computation" we mean that (i) if $i > 0$, σ_i is an output of $\mathit{fold}(t, T, t', \sigma_{i-1})$, and (ii) if $i < n$, $\mathit{cows}_0(t, T, t', \sigma_i)$ is a recursive call within the execution of $\mathit{cows}_0(t, T, t', \mathsf{Id}_{\mathfrak{A}})$.

We develop the sequence inductively. Note that $\sigma_0 = \mathsf{Id}_{\mathfrak{A}} \trianglelefteq \sigma$ for any σ, and that our initial call of $\mathit{cows}(t, T, t')$ establishes that we run cows_0 on (t, T, t', σ_0).

Suppose that we have $\mathsf{Id}_{\mathfrak{A}} = \sigma_0 \trianglelefteq \ldots \trianglelefteq \sigma_i \trianglelefteq \sigma$ and that we run $\mathit{cows}_0(t, T, t', \sigma_i)$. If $\sigma_i(t)$ is COW $\sigma_i(T)$ in $\sigma_i(t')$ then $n = i$ and our sequence is complete. Otherwise, we calculate $S = \mathit{fold}(t, T, t', \sigma_i)$; we wish to prove that there exists a $\sigma_{i+1} \in S$ such that $\sigma_i \trianglelefteq \sigma_{i+1} \trianglelefteq \sigma$.

By Lemma A.5, we know that $\mathrm{CarPath}(\sigma_i(t), \sigma_i(t')) \subseteq \mathrm{CarPath}(\sigma(t), \sigma(t'))$. Let $p_1, \ldots, p_k$ be the positions in $\mathrm{CarPath}(\sigma_i(t), \sigma_i(t'))$. In fold_0, the set S initially contains $\varphi_0 = \mathsf{Id}_{\mathfrak{A}}$; note that $\varphi_0 \circ \sigma_i \trianglelefteq \sigma$. In the next paragraph we show how to define a sequence of substitutions $\varphi_0 \trianglelefteq \ldots \trianglelefteq \varphi_k$ in S in successive calls to fold_0, such that $\varphi_l \circ \sigma_i \trianglelefteq \sigma$. This sequence will serve to bridge the difference between σ_i and σ_{i+1}.

Suppose that $\varphi_l \in S$ in fold_0 when P consists of $p_{l+1}, \ldots, p_k$, and that $\varphi_l \circ \sigma_i \trianglelefteq \sigma$. Since $\sigma(t)$ is COW $\sigma(T)$ in $\sigma(t')$, there exists a proper prefix p'_{l+1} of p_{l+1} and a $t_e \in T$ such that $\sigma(t' @ p'_{l+1}) \equiv \sigma(t_e)$. Note that $\sigma_i(t_e) \in \sigma_i(T)$, and that $\sigma_i(t') @ p'_{l+1} \in \mathit{anc}(\sigma_i(t') @ p_{l+1}))$. Thus, fold_0 calls $\mathit{solve}(X, Y, S)$ where $\sigma_i(t_e) \in X$, $\sigma_i(t' @ p'_{l+1}) \in Y$, and $\varphi_l \in S$.

Write $\sigma = \psi_l \circ \varphi_l \circ \sigma_i$: note that ψ_l is a unifier of $\varphi_l(t_e)$ and $\varphi_l(t' @ p'_{l+1})$. Thus, there exists a $\psi'_l \in \mathit{unify}(\varphi_l(t_e), \varphi_l(t' @ p'_{l+1}))$ such that $\psi'_l \trianglelefteq \psi_l$. Let $\varphi_{l+1} = \psi'_l \circ \varphi_l$: note that $\psi'_l \circ \varphi_l \trianglelefteq \psi_l \circ \varphi_l$ so $\sigma_i \trianglelefteq \varphi_l \trianglelefteq \varphi_{l+1} \trianglelefteq \sigma$. Note further that $\varphi_{l+1} \in \mathit{solve}(X, Y, S)$ and thus φ_{l+1} is in S in fold_0 when P consists of $p_{l+2}, \ldots, p_k$.

Consider φ_k: note that $\sigma_i \trianglelefteq \varphi_k \trianglelefteq \sigma$ and that $\varphi_k \in S$ in fold_0 when $P = \{\}$, so $\varphi_k \in \mathit{fold}(t, T, t', \sigma_i)$. Let $\sigma_{i+1} = \varphi_k$. Because φ_k is in the output of $\mathit{fold}(t, T, t', \sigma_i)$, note that we make a recursive call to cows_0 on (t, T, t', σ_{i+1}).

In this way we define a (potentially infinite) sequence $\mathsf{Id}_{\mathfrak{A}} = \sigma_0 \trianglelefteq \sigma_1 \trianglelefteq \ldots \trianglelefteq \sigma_i \trianglelefteq \ldots \trianglelefteq \sigma$. Moreover, by Lemma A.7, we know that $\sigma_0 \triangleleft \ldots \triangleleft \sigma_i \triangleleft \ldots \trianglelefteq \sigma$. Note that each substitution is produced by composing a unification with the previous substitution. Thus, by Lemma A.10, the sequence must be finite, with σ_n being the last substitution before σ. But the sequence can only end at σ_n if $\sigma_n(t)$ is COW $\sigma_n(T)$ in $\sigma_n(t')$. If this is the case, then in the

recursive call to *cows* on $\sigma_n(t), \sigma_n(T), \sigma_n(t')$, $\mathsf{Id}_{\mathfrak{A}}$ is returned, so $\sigma_n \circ \mathsf{Id}_{\mathfrak{A}} = \sigma_n \in \mathit{cows}(t, T, t')$. This completes the proof, with $\sigma' = \sigma_n$. □

The *cows* function finds solutions to a single carried only within problem. To calculate substitutions, we need the following function, *cowt*, which finds solutions to a set of carried only within problems.

$$\begin{aligned} &\mathit{cowt}(t, T, T') = \\ &\quad \mathit{cows}(t, T, \mathit{concat}(T')) \end{aligned}$$

where $\mathit{concat}(T')$ is a concatenation, by successive pairings, of the messages in T'.

Lemma A.12. *cowt complete terminates* $\mathit{cowt}(t, T, T')$ *is a most general set of substitutions* σ *such that* $\sigma(t)$ *is carried only within* $\sigma(T)$ *in* $\sigma(T')$*, and* $\mathit{cowt}(t, T, T')$ *can be calculated in finitely many steps.*

Proof. It should be obvious that *cowt* will terminate. Note that the carried paths of $\mathit{concat}(T')$ are in one-to-one correspondance with the carried paths of members of T'; this establishes that *cowt* gives a complete set of most general substitutions such that $\sigma(t)$ is carried only within $\sigma(T)$ in $\sigma(T')$, because *cows* is complete. □