Checking Entailment Between Separation Logic Symbolic Heaps: Beyond Connected and Established Systems

Nicolas Peltier, Radu Iosif, Mnacho Echenim

Checking Entailment Between Separation Logic Symbolic Heaps: Beyond Connected and Established Systems

Mnacho Echenim, Radu Iosif and Nicolas Peltier
Univ. Grenoble Alpes, CNRS, LIG, F-38000 Grenoble France
Univ. Grenoble Alpes, CNRS, VERIMAG, F-38000 Grenoble France

January 15, 2021

Abstract

We show that the entailment problem $\varphi \models \psi$ in Separation Logic is decidable for separated conjunctions of atoms φ and ψ, that contain predicate symbols whose interpretation is given inductively by a set of recursive rules. The proof is based on a reduction to a class of entailment problems shown to be decidable in [9]. In contrast with the works of [9, 12, 13], the considered inductive rules may introduce memory locations without allocating them, which strongly extends the class of structures that can be constructed. Moreover, the result is more general than the one given in [8], because the conditions on the inductive rules corresponding to the left-hand side of the considered entailment are strongly relaxed: it is only assumed that the rules are progressing, i.e. that they allocate exactly one memory location.

1 Introduction

Separation Logic [11, 14] was introduced in order to reason efficiently about programs manipulating recursively linked data structures. It forms the basis of several industrial-scale static program analysis techniques [4, 3, 5]. Given a set $\mathcal{L}$ of memory locations (e.g., addresses), the formulas in this logic describe *heaps* that are finite partial functions mapping some locations to records of locations. A location ℓ is *allocated* if it occurs in the domain of the heap. An atom $x \mapsto (y_1, \ldots, y_\kappa)$ states that the location associated with x refers to the tuple of locations associated with $(y_1, \ldots, y_\kappa)$. The *separating conjunction* $\phi * \psi$ states that the formulæ ϕ and ψ hold in non-overlapping parts of the heap, that have disjoint domains. This connective allows for modular program analyses, because the formulæ specifying the behaviour of a program statement refer only to the small (local) set of locations that are manipulated by that statement, with no concern for the rest of the program's state.

To reason about recursive data structures of unbounded sizes (lists, trees, etc.), the base logic is enriched by predicate symbols, with a semantics specified by user-defined inductive rules.

An important problem in program verification, arising during construction of Hoare-style correctness proofs, is the discharge of verification conditions, that are entailments of the form $\phi \vdash \psi$, where ϕ and ψ are Separation Logic formulas.

In general, the entailment problem is undecidable for formulas containing inductive defined predicates [10, 1]. A first decidable class of entailment problems is described in [9] and involves three restrictions on the SID rules: *progress*, *connectivity* and *establishment*, formally defined later. Intuitively, the progress condition (P) states that every rule allocates exactly one location, the connectivity condition (C) states that the set of allocated locations has a tree-shaped structure, and the establishment condition (E) states that every existential variable is (eventually) allocated. In [12, 13] a 2EXPTIME algorithm was proposed for testing the validity of PCE entailment problems, and in [6], a 2EXPTIME-hardness proof is provided. In [8], we relaxed the condition ensuring decidability and showed that entailment is still in 2EXPTIME if the establishment condition is removed and is replaced by a strictly less restrictive condition, called *restrictedness*, which imposes some conditions on the form of the (dis)equations occurring in the problem. One interesting feature of this class of entailment is that it allows one to handle rules generating data structures with "dangling" edges, for instance rules with pending elements. We also slightly generalized the connectivity condition, by allowing forests (rooted on free variables) instead of trees. In the present paper, we strongly generalize this result by showing that the connectivity and restrictedness conditions need only to be imposed for the right-hand side of the entailment. Hence no condition (other than the progress condition) needs to be imposed on the left-hand side of the entailment, which is a major improvement. To this aim, we prove that every entailment in which the left-hand side formula is progressing, and the right-hand side is progressing, connected and restricted can be reduced to a PCE entailment (although the class of data structures that can be described is strictly bigger).

2 Definitions

For a (partial) function $f : A \to B$, we denote by $\mathrm{dom}(f)$ and $\mathrm{rng}(f)$ its domain and range, respectively. A function f is finite if $|\mathrm{dom}(f)| < \infty$, where $|S|$ denotes the cardinality of the set S. The subset $\{k, k+1, \ldots, \ell\}$ of the set $\mathbb{N}$ of natural numbers is denoted as $[\![k, \ell]\!]$; note that $[\![k, \ell]\!] = \emptyset$ whenever $\ell < k$. For a relation $R \subseteq A \times A$, we denote by R^* its reflexive and transitive closure, i.e. $R^* \stackrel{\text{def}}{=} \{(x_1, x_n) \mid n \geq 1,\ \forall i \in [\![1, n-1]\!] \,.\, (x_i, x_{i+1}) \in R\}$.

Let κ be a fixed natural number and let P be a countably infinite set of predicate symbols. Each predicate symbol $p \in \mathsf{P}$ is associated a unique arity $ar(p)$. Let V be a countably infinite set of variables. For technical convenience, we also consider a special constant $\bot$, which will be used to denote "empty"

record fields. Formulæ are built inductively according to the following syntax:

$$\phi := x \not\approx y \mid x \approx y \mid x \mapsto (y_1, \ldots, y_\kappa) \mid p(x_1, \ldots, x_n) \mid \phi_1 * \phi_2 \mid \phi_1 \vee \phi_2 \mid \exists x \,.\, \phi_1$$

where $p \in \mathsf{P}$ is a predicate symbol of arity $n = ar(p)$, $x, x_1, \ldots, x_n \in \mathsf{V}$ are variables and $y_1, \ldots, y_\kappa \in \mathsf{V} \cup \{\bot\}$ are *terms*, that is either variables or $\bot$. The set of variables freely occurring in a formula ϕ is denoted by $\mathrm{fv}(\phi)$ (we assume by α-equivalence that the same variable cannot occur both free and bound in the same formula ϕ, and that distinct quantifiers bind distinct variables). The size $|\phi|$ of a formula ϕ is the number of occurrences of symbols in ϕ.

A formula $x \mapsto (y_1, \ldots, y_\kappa)$ is a *points-to atom*, whereas a formula $p(x_1, \ldots, x_n)$ is a *predicate atom*. A formula is *predicate-less* if no predicate atom occurs in it. A *symbolic heap* is a formula containing no disjunctions, i.e. of the form $\exists \vec{x} \,.\, \ast_{j=1}^{m} \alpha_i$, where each α_i is either a points-to or a predicate atom.

Definition 1. *A variable x is* allocated by a symbolic heap ϕ *iff ϕ contains a sequence of equalities $x_1 \approx x_2 \approx \ldots \approx x_{n-1} \approx x_n$, for $n \geq 1$, such that $x = x_1$ and $x_n \mapsto (y_1, \ldots, y_\kappa)$ occurs in ϕ, for some terms $y_1, \ldots, y_\kappa \in \mathsf{V} \cup \{\bot\}$.*

A *substitution* is a partial function mapping variables to terms . If σ is a substitution and ϕ is a formula, a variable or a tuple, then $\phi\sigma$ denotes the formula, the variable or the tuple obtained from ϕ by replacing every free occurrence of a variable $x \in \mathrm{dom}(\sigma)$ by $\sigma(x)$, respectively. We denote by $\{\langle x_i, y_i \rangle \mid i \in [\![1, n]\!]\}$ the substitution with domain $\{x_1, \ldots, x_n\}$ that maps x_i to y_i, for each $i \in [\![1, n]\!]$.

Let $\mathcal{L}$ be a countably infinite set of *locations* containing, in particular, a special location $\bot$. A *structure* is a pair $(\mathfrak{s}, \mathfrak{h})$, where:

- $\mathfrak{s}$ is a partial function from $\mathsf{V} \cup \{\bot\}$ to $\mathcal{L}$, called a *store*, such that $\bot \in \mathrm{dom}(\mathfrak{s})$ and $\mathfrak{s}(x) = \bot \iff x = \bot$, and
- $\mathfrak{h}$ is a finite partial function from $\mathcal{L}$ to $\mathcal{L}^\kappa$, such that $\bot \notin \mathrm{dom}(\mathfrak{h})$.

Given a heap $\mathfrak{h}$, we define $\mathrm{ref}(\mathfrak{h}) \stackrel{\text{def}}{=} \bigcup_{l \in \mathrm{dom}(\mathfrak{h})} \{\ell_i \mid \mathfrak{h}(\ell) = (\ell_1, \ldots, \ell_\kappa), i \in [\![1, \kappa]\!]\}$ and $\mathrm{loc}(\mathfrak{h}) \stackrel{\text{def}}{=} \mathrm{dom}(\mathfrak{h}) \cup \mathrm{ref}(\mathfrak{h})$. Two heaps $\mathfrak{h}_1$ and $\mathfrak{h}_2$ are *disjoint* iff $\mathrm{dom}(\mathfrak{h}_1) \cap \mathrm{dom}(\mathfrak{h}_2) = \emptyset$, in which case $\mathfrak{h}_1 \uplus \mathfrak{h}_2$ denotes the union of $\mathfrak{h}_1$ and $\mathfrak{h}_2$ (undefined if $\mathfrak{h}_1$ and $\mathfrak{h}_2$ are not disjoint).

If $x_1, \ldots, x_n$ are pairwise distinct variables and $\ell_1, \ldots, \ell_n \in \mathcal{L}$ are locations, we denote by $\mathfrak{s}[x_i \leftarrow \ell_i \mid 1 \leq i \leq n]$ the store $\mathfrak{s}'$ defined by $\mathrm{dom}(\mathfrak{s}') = \mathrm{dom}(\mathfrak{s}) \cup \{x_1, \ldots, x_n\}$, $\mathfrak{s}'(y) = \ell_i$ if $y = x_i$ for some $i \in [\![1, n]\!]$, and $\mathfrak{s}'(y) = \mathfrak{s}(x)$ otherwise. If $x_1, \ldots, x_n \notin \mathrm{dom}(\mathfrak{s})$, then the store $\mathfrak{s}'$ is called an *extension* of $\mathfrak{s}$ to $\{x_1, \ldots, x_n\}$.

A *system of inductive definitions* (SID) is a set $\mathcal{R}$ of rules of the form $p(x_1, \ldots, x_n) \Leftarrow \pi$, where $p \in \mathsf{P}$, $n = ar(p)$, $x_1, \ldots, x_n$ are pairwise distinct variables and π is a quantifier-free symbolic heap. The predicate atom $p(x_1, \ldots, x_n)$ is the *head* of the rule and $\mathcal{R}(p)$ denotes the subset of $\mathcal{R}$ consisting of rules with head $p(x_1, \ldots, x_n)$ (the choice of $x_1, \ldots, x_n$ is not important). The variables in $\mathrm{fv}(\pi) \setminus \{x_1, \ldots, x_n\}$ are called the *existential variables of the rule*. Note that, by definition, these variables are not explicitly quantified inside π and that π is quantifier-free. For simplicity, we denote by $p(x_1, \ldots, x_n) \Leftarrow_{\mathcal{R}} \pi$ the fact

that the rule $p(x_1, \ldots, x_n) \Leftarrow \pi$ belongs to $\mathcal{R}$. The *size* of $\mathcal{R}$ is defined as $|\mathcal{R}| \stackrel{\text{def}}{=} \sum_{p(x_1,\ldots,x_n) \Leftarrow_{\mathcal{R}} \pi} |\pi|$ and its *width* as $\mathrm{w}(\mathcal{R}) \stackrel{\text{def}}{=} \max_{p(x_1,\ldots,x_n) \Leftarrow_{\mathcal{R}} \pi} |\pi|$.

Given predicate symbols $p, q \in \mathsf{P}$, we write $p \succeq_{\mathcal{R}} q$ iff $\mathcal{R}$ contains a rule of the form $p(x_1, \ldots, x_n) \Leftarrow \pi$, and q occurs in π. The relation $p \succeq^*_{\mathcal{R}} q$ is the reflexive and transitive closure of $\succeq_{\mathcal{R}}$, in which case we say that p *depends on* q. For a formula ϕ, we denote by $\mathcal{P}(\phi)$ the set of predicate symbols q, such that p occurs in ϕ and $p \succeq^*_{\mathcal{R}} q$.

Given formulæ ϕ and ψ, we write $\phi \Leftarrow_{\mathcal{R}} \psi$ if ψ is obtained from ϕ by replacing an atom $p(u_1, \ldots, u_n)$ by $\pi \{\langle x_1, u_1 \rangle, \ldots, \langle x_n, u_n \rangle\}$, where $\mathcal{R}$ contains a rule $p(x_1, \ldots, x_n) \Leftarrow \pi$. We assume that each unfolding step introduces fresh existential variables. We say that ψ is an *unfolding* of ϕ iff $\phi \Leftarrow^*_{\mathcal{R}} \psi$.

Lemma 2. *Every unfolding of a symbolic heap is again a symbolic heap.*

Proof. By an easy induction on the length of the unfolding sequence. □

Given an SID $\mathcal{R}$, $(\mathfrak{s}, \mathfrak{h}) \models_{\mathcal{R}} \phi$ is the least relation between structures and formulæ, such that, whenever $(\mathfrak{s}, \mathfrak{h}) \models_{\mathcal{R}} \phi$, we have $\mathrm{fv}(\phi) \subseteq \mathrm{dom}(\mathfrak{s})$ and the following conditions hold:

$(\mathfrak{s}, \mathfrak{h})$	$\models_{\mathcal{R}}$	$x \approx y$	if $\mathrm{dom}(\mathfrak{h}) = \emptyset$ and $\mathfrak{s}(x) = \mathfrak{s}(y)$
$(\mathfrak{s}, \mathfrak{h})$	$\models_{\mathcal{R}}$	$x \not\approx y$	if $\mathrm{dom}(\mathfrak{h}) = \emptyset$ and $\mathfrak{s}(x) \neq \mathfrak{s}(y)$
$(\mathfrak{s}, \mathfrak{h})$	$\models_{\mathcal{R}}$	$x \mapsto (y_1, \ldots, y_\kappa)$	if $\mathrm{dom}(\mathfrak{h}) = \{\mathfrak{s}(x)\}$ and $\mathfrak{h}(\mathfrak{s}(x)) = \langle \mathfrak{s}(y_1), \ldots, \mathfrak{s}(y_\kappa) \rangle$
$(\mathfrak{s}, \mathfrak{h})$	$\models_{\mathcal{R}}$	$\phi_1 * \phi_2$	if there exist disjoint heaps $\mathfrak{h}_1$ and $\mathfrak{h}_2$ such that $\mathfrak{h} = \mathfrak{h}_1 \uplus \mathfrak{h}_2$ and $(\mathfrak{s}, \mathfrak{h}_i) \models_{\mathcal{R}} \phi_i$, for both $i = 1, 2$
$(\mathfrak{s}, \mathfrak{h})$	$\models_{\mathcal{R}}$	$\phi_1 \vee \phi_2$	if $(\mathfrak{s}, \mathfrak{h}) \models_{\mathcal{R}} \phi_i$, for some $i = 1, 2$
$(\mathfrak{s}, \mathfrak{h})$	$\models_{\mathcal{R}}$	$p(x_1, \ldots, x_n)$	if $p(x_1, \ldots, x_n) \Leftarrow_{\mathcal{R}} \phi$, $\{x_1, \ldots, x_n\} \subseteq \mathrm{dom}(\mathfrak{s})$ and there exists an extension $\mathfrak{s}_e$ of $\mathfrak{s}$ to $\mathrm{fv}(\phi) \setminus \mathrm{dom}(\mathfrak{s})$ such that $(\mathfrak{s}_e, \mathfrak{h}) \models \phi$ — we assume by renaming that $(\mathrm{fv}(\phi) \setminus \{x_1, \ldots, x_n\}) \cap \mathrm{dom}(\mathfrak{s}) = \emptyset$
$(\mathfrak{s}, \mathfrak{h})$	$\models_{\mathcal{R}}$	$\exists x \,.\, \phi$	if there exists $\ell \in \mathcal{L}$ such that $(\mathfrak{s}[x \leftarrow \ell], \mathfrak{h}) \models \phi$

Given formulæ ϕ and ψ, we write $\phi \models_{\mathcal{R}} \psi$ whenever $(\mathfrak{s}, \mathfrak{h}) \models_{\mathcal{R}} \phi \Rightarrow (\mathfrak{s}, \mathfrak{h}) \models_{\mathcal{R}} \psi$, for all structures $(\mathfrak{s}, \mathfrak{h})$ and $\phi \equiv_{\mathcal{R}} \psi$ for $(\phi \models_{\mathcal{R}} \psi$ and $\psi \models_{\mathcal{R}} \phi)$. We omit the subscript $\mathcal{R}$ whenever these relations hold for any SID.

It is easy to check that, for all formulas ϕ_1, ϕ_2, ψ, $(\phi_1 \vee \phi_2) * \psi \equiv (\phi_1 * \psi) \vee (\phi_2 * \psi)$ and $(\exists x.\phi_1) * \phi_2 \equiv \exists x \,.\, \phi_1 * \phi_2$. Consequently, each formula can be transformed into an equivalent finite disjunction of symbolic heaps.

Definition 3. *An* entailment problem *is a triple* $\mathfrak{P} \stackrel{\text{def}}{=} \phi \vdash_{\mathcal{R}} \psi$*, where* ϕ *is a quantifier-free formula,* ψ *is a formula and* $\mathcal{R}$ *is an SID. The problem* $\mathfrak{P}$ *is said to be* valid valid *if and only if* $\phi \models_{\mathcal{R}} \psi$*. The* size *of the problem* $\mathfrak{P}$ *is defined as* $|\mathfrak{P}| \stackrel{\text{def}}{=} |\phi| + |\psi| + |\mathcal{R}|$ *and its* width *is defined as* $\mathrm{w}(\mathfrak{P}) \stackrel{\text{def}}{=} \max(|\phi|, |\psi|, \mathrm{w}(\mathcal{R}))$.

Note that considering ϕ to be quantifier-free loses no generality, because $\exists x.\phi \models_{\mathcal{R}} \psi \iff \phi \models_{\mathcal{R}} \psi$.

3 Decidable Entailment Problems

The class of general entailment problems is undecidable, the initial proofs from [10, 1] are refined by Theorem 6 below. A first attempt to define a natural decidable class of entailment problems is described in [9] and involves three restrictions on the SID rules, called *progress*, *connectivity* and *establishment*:

Definition 4. *A rule* $p(x_1, \ldots, x_n) \Leftarrow \pi$ *is:*

1. progressing *iff* $\pi = x_1 \mapsto (y_1, \ldots, y_\kappa) * \rho$ *and* ρ *contains no points-to atoms,*
2. connected *iff it is progressing,* $\pi = x_1 \mapsto (y_1, \ldots, y_\kappa) * \rho$ *and every predicate atom in* ρ *is of the form* $q(y_i, \vec{u})$*, for some* $i \in [\![1, \kappa]\!]$*,*
3. established *iff every existential variable* $x \in \mathrm{fv}(\pi) \setminus \{x_1, \ldots, x_n\}$ *is allocated by every predicate-less unfolding*[1] $\pi \Leftarrow^*_{\mathcal{R}} \phi$*.*

An SID $\mathcal{R}$ *is* progressing *(*connected, established*)* for a formula ϕ *iff every rule in* $\bigcup_{p \in \mathcal{P}(\phi)} \mathcal{R}(p)$ *is progressing (resp. connected, established). An entailment problem* $\phi \vdash_{\mathcal{R}} \psi$ *is* left *(resp.* right*)* progressing *(resp.* connected, established*) iff* $\mathcal{R}$ *is progressing (resp. connected, established) for* ϕ *(resp.* ψ*). An entailment problem is* progressing *(resp.* connected, established*) iff it is both left and right progressing (resp. connected, established).*

The decidability of progressing, connected and left-established entailment problems is an immediate consequence of the result of [9]. Moreover, an analysis of the proof [9] leads to an elementary recursive complexity upper bound, which has been recently tighten down to 2EXPTIME-complete [13, 8, 6]. In the following, we refer to Table 1 for a recap of the complexity results for the entailment problem.

Table 1: Decidability and Complexity Results for the Entailment Problem — ✓ (resp. λ) means the condition (resp. λ-condition) holds both left and right.

Reference	Progress	Connected	Established	Restricted	Complexity
Theorem 5	✓	✓	left	-	2EXP-co.
Theorem 6	✓	left	✓	-	undec.
[7, Theorem 6]	✓	✓	-	-	undec.
[8, Theorem 32]	✓	λ	-	λ	2EXP-co.
Theorem 44	✓	λ-right	-	λ-right	2EXP-co.

Theorem 5. *The progressing, connected and left-established entailment problem is* 2*EXPTIME-complete. Moreover, given an instance* $\mathfrak{P}$ *of this problem, there exists an algorithm that runs in time* $2^{2^{\mathcal{O}(\mathrm{w}(\mathfrak{P})^8 \cdot \log |\mathfrak{P}|)}}$.

Proof. The 2EXPTIME-hardness lower bound is given in [6, Theorem 18]. The upper bound is explained in the proof of [8, Theorem 32]. □

[1] Note that, by Lemma 2, ϕ is a symbolic heap.

A natural question arises in this context: which of the restrictions from the above theorem can be relaxed and what is the price (in terms of computational complexity) of relaxing (some of) them? In the light of Theorem 6 below, the connectivity restriction cannot be completely dropped. Further, if we drop the establishment condition, the problem becomes undecidable [7, Theorem 6], even if both the left/right progress and connectivity conditions apply.

Theorem 6. *The progressing, left-connected and established entailment problem is undecidable.*

Proof. By a reduction from the known undecidable problem of universality of context-free languages. A context-free grammar $G = \langle N, T, S, \Delta \rangle$ consists of a finite set N of nonterminals, a finite set T of terminals, a start symbol $S \in N$ and a finite set Δ of productions of the form $A \to w$, where $A \in N$ and $w \in (N \cup T)^*$. Given finite strings $u, v \in (N \cup T)^*$, the step relation $u \Rightarrow v$ replaces a nonterminal A of u by the right-hand side w of a production $A \to w$ and $\Rightarrow^*$ denotes the reflexive and transitive closure of $\Rightarrow$. The language of G is the set $\mathcal{L}(G)$ of finite strings $w \in T^*$, such that $s \Rightarrow^* w$. The problem $T^* \subseteq \mathcal{L}(G)$ is known as the universality problem, known to be undecidable [2]. W.l.o.g. we assume further that:

- $T = \{0, 1\}$, because every terminal can be encoded as a binary string,
- $\mathcal{L}(G)$ does not contain the empty string ϵ, because computing a grammar G' such that $\mathcal{L}(G') = \mathcal{L}(G) \cap T^+$ is possible in polynomial time and, moreover, we can reduce from the modified universality problem problem $T^+ \subseteq \mathcal{L}(G')$ instead of the original $T^* \subseteq \mathcal{L}(G)$,
- G is in Greibach normal form and contains only production rules of the form $A_0 \to aA_1 \dots A_n$, where $A_0, \dots A_n \in N$, for some $n \geq 0$ and $a \in T$.

We use the special variables $\hat{0}$ and $\hat{1}$ to denote the binary digits 0 and 1. For each nonterminal $A_0 \in N$, we have a predicate $A_0(x, y, \hat{0}, \hat{1})$ and a rule $A_0(x, y, \hat{0}, \hat{1}) \Leftarrow x \mapsto (\hat{a}, x_1) * A_1(x_1, x_2, \hat{0}, \hat{1}) * \dots A_n(x_n, y, \hat{0}, \hat{1})$, for each rule $A_0 \to aA_1 \dots A_n$ of G. Moreover, we consider the rules $T(x, y, \hat{0}, \hat{1}) \Leftarrow x \mapsto (\hat{a}, z) * T(z, y, \hat{0}, \hat{1})$, for all $a \in \{0, 1\}$ and let $\mathcal{R}$ be the resulting SID. It is easy to check that the SID is progressing and established and, moreover, the rules for T are connected. Finally, the entailment $\hat{0} \not\approx \hat{1} * T(x, y) \vdash_{\mathcal{R}} S(x, y)$ is valid if and only if $T^+ \subseteq \mathcal{L}(G)$. □

The second decidable class of entailment problems relaxes the connectivity condition and replaces the establishment with a syntactic condition (that can be checked in linear time in the size of the SID), while remaining 2EXPTIME-complete [8]. To define this class, we consider $\mathcal{R}$-*positional functions*, i.e. functions that map every n-ary predicate symbol p occurring in $\mathcal{R}$ to a subset of $[\![1, n]\!]$. Given an $\mathcal{R}$-positional function λ and a formula ϕ, we denote by $\mathsf{V}_\lambda(\phi)$ the set of variables x_i such that ϕ contains a predicate atom $p(x_1, \dots, x_n)$ with $i \in \lambda(p)$. Note that V_λ is stable under substitutions, i.e. $\mathsf{V}_\lambda(\phi\sigma) = (\mathsf{V}_\lambda(\phi))\sigma$, for each formula ϕ and each substitution σ.

Definition 7. *Let ψ be a formula and $\mathcal{R}$ be an SID. The* fv-profile *of the pair $(\psi, \mathcal{R})$ is the $\mathcal{R}$-positional function λ, where $\lambda(p)$, $p \in \mathsf{P}$ are the maximal sets satisfying the following conditions:*

1. $\mathsf{V}_\lambda(\psi) \subseteq \mathrm{fv}(\psi)$.
2. *For all predicate symbols $p \in \mathcal{P}(\psi)$, all rules $p(x_1, \ldots, x_n) \Leftarrow \pi$ in $\mathcal{R}$, all predicate atoms $q(y_1, \ldots, y_m)$ in π and all $i \in \lambda(q)$, there exists $j \in \lambda(p)$ such that $x_j = y_i$.*

The fv-profile of $(\psi, \mathcal{R})$ is denoted by $\lambda^\psi_\mathcal{R}$.

Intuitively, given a predicate $p \in \mathsf{P}$, the set $\lambda^\psi_\mathcal{R}(p)$ denotes the formal parameters of p that, in every unfolding of ψ, will always be substituted by variables occurring freely in ψ. It is easy to check that $\lambda^\psi_\mathcal{R}$ can be computed in polynomial time w.r.t. $|\psi| + |\mathcal{R}|$, using a straightforward greatest fixpoint algorithm. The algorithm starts with a function mapping every predicate p of arity n to $[\![1, n]\!]$ and repeatedly removes elements from the sets $\lambda(p)$ to ensure that the above conditions hold. In the worst case, we may have eventually $\lambda(p) = \emptyset$ for all predicate symbols p.

Definition 8. *Let λ be an $\mathcal{R}$-positional function, and V be a set of variables. A formula ϕ is* λ-restricted *w.r.t. V iff the following hold:*

1. *for every disequation $y \not\approx z$ in ϕ, we have $\{y, z\} \cap V \neq \emptyset$, and*
2. $\mathsf{V}_\lambda(\phi) \subseteq V$.

A rule $p(x_1, \ldots, x_n) \Leftarrow x \mapsto (y_1, \ldots, y_\kappa) * \rho$ is:

- λ-connected *iff for every atom $q(z_1, \ldots, z_m)$ occurring in ρ, we have $z_1 \in \mathsf{V}_\lambda(p(x_1, \ldots, x_n)) \cup \{y_1, \ldots, y_\kappa\}$,*
- λ-restricted *iff ρ is λ-restricted w.r.t. $\mathsf{V}_\lambda(p(x_1, \ldots, x_n))$.*

An SID $\mathcal{R}$ is progressing *(resp.* λ-connected, λ-restricted*)* for a formula ϕ *iff every rule in $\bigcup_{p \in \mathcal{P}(\phi)} \mathcal{R}(p)$ is progressing (resp. λ-connected, λ-restricted). An SID $\mathcal{R}$ is* λ-connected *(*λ-restricted*)* for a formula ϕ *iff every rule in $\bigcup_{p \in \mathcal{P}(\phi)} \mathcal{R}(p)$ is λ-connected (λ-restricted). An entailment problem $\phi \vdash_\mathcal{R} \psi$ is* left *(*right*)* λ-connected, *(*λ-restricted*)* *iff $\mathcal{R}$ is λ-connected (λ-restricted) for ϕ (ψ), where λ is considered to be $\lambda^\phi_\mathcal{R}$ ($\lambda^\psi_\mathcal{R}$). An entailment problem is* λ-connected *(*λ-restricted*)* *iff it is both left and right λ-connected (λ-restricted).*

The class of progressing, λ-connected and λ-restricted entailment problems has been shown to be a generalization of the class of progressing, connected and left-established problems, because the latter can be reduced to the former by a many-one reduction [8, Theorem 13] that runs in time $|\mathfrak{P}| \cdot 2^{\mathcal{O}(\mathrm{w}(\mathfrak{P})^2)}$ on input $\mathfrak{P}$ (Figure 1) and preserves the width asymptotically.

In the rest of this paper we close the loop by defining a syntactic extension of λ-progressing, λ-connected and λ-restricted entailment problems that can be reduced to the class of progressing, connected and left-established entailment problems by a many-one reduction.

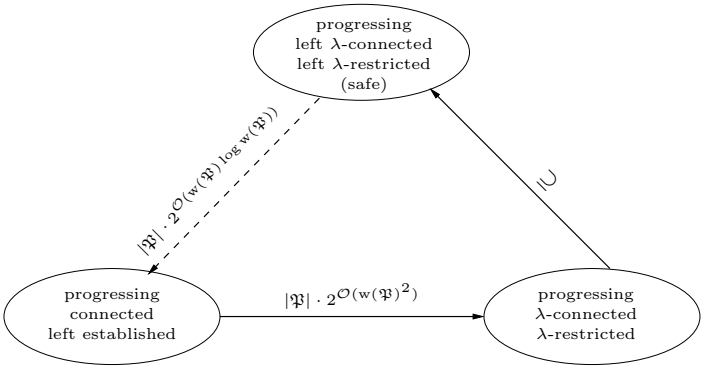

Figure 1: Many-one Reductions between Decidable Entailment Problems

Definition 9. *An entailment problem* $\phi \vdash_{\mathcal{R}} \psi$ *is* safe *if, by letting* $\lambda \stackrel{\text{def}}{=} \lambda^{\psi}_{\mathcal{R}}$*, it is the case that:*

1. *every rule in* $\mathcal{R}$ *is progressing,*
2. ψ *is* λ*-restricted w.r.t.* $\mathrm{fv}(\phi)$*,*
3. *all the rules from* $\bigcup_{p \in \mathcal{P}(\psi)} \mathcal{R}(p)$ *are* λ*-connected and* λ*-restricted.*

Note that there is no condition on the formula ϕ, or on the rules defining the predicates occurring only in ϕ (other than the progress condition). Essentially, the conditions in Definition 9 ensure that all the disequations occurring in any unfolding of ψ involve at least one variable that is free in ϕ. Further, the heaps of the model of ψ must be *forests*, i.e. unions of trees, the roots of which are associated with the first argument of the predicate atoms in ψ or to free variables from ϕ.

We refer the reader to Figure 1 for a general picture of the entailment problems considered so far and of the many-one reductions between them, where the reduction corresponding to the dashed arrow is the concern of the next section. Importantly, since all reductions are many-one, taking time polynomial in the size and exponential in the width of the input problem, while preserving its width asymptotically, the three classes from Figure 1 can be joined into a single, general, 2EXPTIME-complete class of entailment problems.

4 From Safe To Established Entailment

In this section we consider an instance of the safe entailment problem $\mathfrak{P} = \phi \vdash_{\mathcal{R}} \psi$ and let λ denote the $\mathcal{R}$-positional function $\lambda^{\psi}_{\mathcal{R}}$, i.e. the fv-profile of $(\psi, \mathcal{R})$ (Definition 7). Let $\{w_1, \ldots, w_\nu\} \stackrel{\text{def}}{=} \mathrm{fv}(\phi) \cup \mathrm{fv}(\psi)$ and $\vec{w} \stackrel{\text{def}}{=} (w_1, \ldots, w_\nu)$, the order of variables does not matter and may be chosen arbitrarily. We further assume, w.l.o.g., that $\nu > 0$. Note that, by definition, the right-hand side of every rule in $\mathcal{R}$ is a symbolic heap (i.e., $\mathcal{R}$ contains no disjunction).

Let $\mathcal{P}_l \stackrel{\text{def}}{=} \mathcal{P}(\phi)$ and $\mathcal{P}_r \stackrel{\text{def}}{=} \mathcal{P}(\psi)$ be the predicate symbols that depend on the

predicate symbols occurring in the left- and right-hand side of the entailment, respectively. We assume that ϕ and ψ contain no points-to atoms and that $\mathcal{P}_l \cap \mathcal{P}_r = \emptyset$. Again, these assumptions lose no generality, because a points-to atom $x \mapsto (y_1, \ldots, y_\kappa)$ can be replaced by a predicate atom $p(x, y_1, \ldots, y_\kappa)$, where p is a fresh predicate symbol associated with the rule $p(u, v_1, \ldots, v_\kappa) \Leftarrow u \mapsto (v_1, \ldots, v_\kappa)$. Moreover the condition $\mathcal{P}_l \cap \mathcal{P}_r \neq \emptyset$ may be enforced by considering two copies of each predicate, for the left-hand side and for the right-hand side, respectively. Finally, we assume that every rule contains exactly μ existential variables, for some fixed $\mu \in \mathbb{N}$; this condition can be enforced by adding dummy literals $x \approx x$ if needed.

We describe a reduction of $\mathfrak{P}$ to an equivalent progressing, connected, and left-established entailment problem. The reduction will add $\nu + \mu$ record fields to the heap. In what follows, we shall therefore often consider heaps and points-to atoms having $\kappa + \nu + \mu$ record fields, where the formal definitions are similar to those given before. Usually such formulas and heaps will be written with a prime. These additional record fields will be used to ensure that the constructed system is connected, by adding all the existential variables of a given rule (as well as the variables in $w_1, \ldots, w_\nu$) into the image of the location allocated by the considered rule. Furthermore, the left-establishment condition will be enforced by adding predicates and rules in order to allocate all the locations that correspond to existential quantifiers and that are not already allocated, making such locations point to a dummy vector $\vec{\bot} \stackrel{\text{def}}{=} (\bot, \ldots, \bot)$, of length $\kappa + \nu + \mu$, where $\bot$ is the special constant denoting empty heap entries . To this aim, we shall use a predicate symbol $\underline{\bot}$ associated with the rule $\underline{\bot}(x) \Leftarrow x \mapsto \vec{\bot}$, where $\vec{\bot} = (\bot, \ldots, \bot)$. Note that allocating all these locations will entail (by definition of the separating conjunction) that they are distinct, thus the addition of such predicates and rules will reduce the number of satisfiable unfoldings. However, due to the restrictions on the use of disequations, we shall see that this does not change the status of the entailment problem.

Definition 10. *For any total function $\gamma : \mathcal{L} \to \mathcal{L}$ and any tuple $\vec{\ell} = \langle \ell_1, \ldots, \ell_n \rangle \in \mathcal{L}^n$, we denote by $\gamma(\vec{\ell})$ the tuple $\langle \gamma(\ell_1), \ldots, \gamma(\ell_n) \rangle$. If $\mathfrak{s}$ is a store, then $\gamma(\mathfrak{s})$ denotes the store with domain $\mathrm{dom}(\mathfrak{s})$, such that $\gamma(\mathfrak{s})(x) \stackrel{\text{def}}{=} \gamma(\mathfrak{s}(x))$, for all $x \in \mathrm{dom}(\mathfrak{s})$. Consider a heap $\mathfrak{h}$ such that for all $\ell \neq \ell' \in \mathrm{dom}(\mathfrak{h})$, we have $\gamma(\ell) \neq \gamma(\ell')$. Then $\gamma(\mathfrak{h})$ denotes the heap with domain $\mathrm{dom}(\gamma(\mathfrak{h})) = \{\gamma(\ell) \mid \ell \in \mathrm{dom}(\mathfrak{h})\}$, such that $\gamma(\mathfrak{h})(\gamma(\ell)) \stackrel{\text{def}}{=} \gamma(\mathfrak{h}(\ell))$, for all $\ell \in \mathrm{dom}(\mathfrak{h})$.*

Proposition 11. *Let λ be an $\mathcal{R}$-positional function and let V be a set of variables. Consider an atom $p(u_1, \ldots, u_n)$ such that $\mathsf{V}_\lambda(p(u_1, \ldots, u_n)) \subseteq V$ and a λ-restricted rule $p(x_1, \ldots, x_n) \Leftarrow \pi$. Then the formula $\pi\,\{\langle x_1, u_1 \rangle, \ldots, \langle x_n, u_n \rangle\}$ is λ-restricted w.r.t. V.*

Proof. Let $\theta = \{\langle x_1, u_1 \rangle, \ldots, \langle x_n, u_n \rangle\}$. By hypothesis π is λ-restricted w.r.t. $\mathsf{V}_\lambda(p(x_1, \ldots, x_n))$, thus we have $\mathsf{V}_\lambda(\pi) \subseteq \mathsf{V}_\lambda(p(x_1, \ldots, x_n))$ and we deduce that

$$\mathsf{V}_\lambda(\pi\theta) \;=\; \mathsf{V}_\lambda(\pi)\theta \;\subseteq\; \mathsf{V}_\lambda(p(x_1, \ldots, x_n))\theta \;=\; \mathsf{V}_\lambda(p(u_1, \ldots, u_n)) \;\subseteq\; V.$$

Consider an atom $y\theta \not\approx z\theta$ occurring in $\pi\theta$. Then necessarily $y \not\approx z$ occurs in π, hence $\{y, z\} \cap \mathsf{V}_\lambda(p(x_1, \ldots, x_n)) \neq \emptyset$ and $\{y\theta, z\theta\} \cap \mathsf{V}_\lambda(p(u_1, \ldots, u_n)) \neq \emptyset$. Since $\mathsf{V}_\lambda(p(u_1, \ldots, u_n)) \subseteq V$, we have the result. □

Lemma 12. *Given a set of variables V, let α be a formula that is λ-restricted w.r.t. V, such that $\mathcal{P}(\alpha) \subseteq \mathcal{P}_r$ and let $(\mathfrak{s}, \mathfrak{h})$ be an $\mathcal{R}$-model of α. For every mapping $\gamma : \mathcal{L} \to \mathcal{L}$ such that $\gamma(\ell) = \gamma(\ell') \Rightarrow \ell = \ell'$ holds whenever either $\{\ell, \ell'\} \subseteq \mathrm{dom}(\mathfrak{h})$ or $\{\ell, \ell'\} \cap \mathfrak{s}(V) \neq \emptyset$, we have $(\gamma(\mathfrak{s}), \gamma(\mathfrak{h})) \models_{\mathcal{R}} \alpha$.*

Proof. The proof is by structural induction on the definition of the relation $\models_{\mathcal{R}}$. We distinguish the following cases:

- If $\alpha = (x \approx y)$, then $\mathfrak{s}(x) = \mathfrak{s}(y)$ and $\mathfrak{h} = \emptyset$, thus $\gamma(\mathfrak{s})(x) = \gamma(\mathfrak{s})(y)$ and $\gamma(\mathfrak{h}) = \emptyset$. Therefore, $(\gamma(\mathfrak{s}), \gamma(\mathfrak{h})) \models_{\mathcal{R}} x \approx y$.
- If $\alpha = (x \not\approx y)$, then $\mathfrak{s}(x) \neq \mathfrak{s}(y)$ and $\mathfrak{h} = \emptyset$, hence $\gamma(\mathfrak{h}) = \emptyset$. Since α is λ-restricted w.r.t. V, necessarily one of the variables x or y occurs in V, hence $\{\mathfrak{s}(x), \mathfrak{s}(y)\} \cap \mathfrak{s}(V) \neq \emptyset$. By the hypotheses of the lemma this entails that $\gamma(\mathfrak{s})(x) \neq \gamma(\mathfrak{s})(y)$. Thus $(\gamma(\mathfrak{s}), \gamma(\mathfrak{h})) \models_{\mathcal{R}} x \not\approx y$.
- If $\alpha = x \mapsto (y_1, \ldots, y_\kappa)$ then we have $\mathrm{dom}(\mathfrak{h}) = \{\mathfrak{s}(x)\}$ and $\mathfrak{h}(\mathfrak{s}(x)) = \langle \mathfrak{s}(y_1), \ldots, \mathfrak{s}(y_n) \rangle$. Therefore $\mathrm{dom}(\gamma(\mathfrak{h})) = \{\gamma(\mathfrak{s}(x))\}$ and $\gamma(\mathfrak{h})(\mathfrak{s}(x)) = \gamma(\langle \mathfrak{s}(y_1), \ldots, \mathfrak{s}(y_n) \rangle)$, thus $(\gamma(\mathfrak{s}), \gamma(\mathfrak{h})) \models_{\mathcal{R}} \alpha$.
- If $\alpha = \alpha_1 * \alpha_2$ then there exists $\mathfrak{h}_1, \mathfrak{h}_2$, such that $\mathfrak{h} = \mathfrak{h}_1 \uplus \mathfrak{h}_2$ and $(\mathfrak{s}, \mathfrak{h}_i) \models_{\mathcal{R}} \alpha_i$, for $i = 1, 2$. Since α_i is λ-restricted w.r.t. V and that $\mathcal{P}(\alpha_i) \subseteq \mathcal{P}_r$, by the induction hypothesis, we obtain that $(\gamma(\mathfrak{s}), \gamma(\mathfrak{h}_i)) \models_{\mathcal{R}} \alpha_i$, for $i = 1, 2$. For all $\ell_i \in \mathrm{dom}(\mathfrak{h}_i)$ with $i = 1, 2$, we have $\ell_1, \ell_2 \in \mathrm{dom}(\mathfrak{h})$ and $\ell_1 \neq \ell_2$, thus by the hypothesis of the lemma $\gamma(\ell_1) \neq \gamma(\ell_2)$. Hence $\mathrm{dom}(\gamma(\mathfrak{h}_1))$ and $\mathrm{dom}(\gamma(\mathfrak{h}_2))$ are disjoint, we have $\gamma(\mathfrak{h}) = \gamma(\mathfrak{h}_1) \uplus \gamma(\mathfrak{h}_2)$ and therefore $(\gamma(\mathfrak{s}), \gamma(\mathfrak{h})) \models_{\mathcal{R}} \alpha$.
- If $\alpha = p(u_1, \ldots, u_n)$ then $\mathcal{R}$ contains a rule $p(x_1, \ldots, x_n) \Leftarrow \pi$ and $(\mathfrak{s}_e, \mathfrak{h}) \models_{\mathcal{R}} \pi\theta$, for some extension $\mathfrak{s}_e$ of $\mathfrak{s}$, with $\theta \stackrel{\text{def}}{=} \{\langle x_i, u_i \rangle \mid i \in [\![1, n]\!]\}$. Since $\mathcal{P}(\alpha) \subseteq \mathcal{P}_r$, the rule must be λ-restricted by Condition 3 in Definition 9. By Proposition 11, we deduce that $\pi\theta$ is λ-restricted w.r.t. V. Moreover, we have $\mathcal{P}(\pi\theta) = \mathcal{P}(\pi) \subseteq \mathcal{P}_r$, thus by the induction hypothesis $(\gamma(\mathfrak{s}_e), \gamma(\mathfrak{h})) \models_{\mathcal{R}} \pi\theta$, and therefore $(\gamma(\mathfrak{s}), \gamma(\mathfrak{h})) \models_{\mathcal{R}} \alpha$, because $\gamma(\mathfrak{s}_e)$ is an extension of $\gamma(\mathfrak{s})$.

□

If γ is injective then Lemma 12 holds for any formula:

Lemma 13. *Let α be a formula and let $(\mathfrak{s}, \mathfrak{h})$ be an $\mathcal{R}$-model of α. For every injective mapping $\gamma : \mathcal{L} \to \mathcal{L}$ we have $(\gamma(\mathfrak{s}), \gamma(\mathfrak{h})) \models_{\mathcal{R}} \alpha$.*

Proof. The proof is similar to that of Lemma 12; it is in fact simpler as the conditions on α are useless, since the implication $\gamma(\ell) = \gamma(\ell') \Rightarrow \ell = \ell'$ holds for all ℓ, ℓ'. □

4.1 A Relation On Structures

We introduce a so-called *expansion* relation on structures, as well as a *truncation* operation on heaps. Intuitively, the expansion of a structure is a structure with the same store and whose heap is augmented with new allocated locations (each pointing to $\vec{\bot}$) and additional record fields, referring in particular to all the newly added allocated locations. These locations are introduced to accommodate all the existential variables of the predicate-less unfolding of the left-hand side of the entailment (to ensure that the obtained entailment is left-established). Conversely, the truncation of a heap is the heap obtained by removing these extra locations. We also introduce the notion of a γ-expansion which is a structure whose image by γ is an expansion. We recall that $\vec{w} \stackrel{\text{def}}{=} (w_1, \ldots, w_\nu)$ is the tuple of free variables of the instance $\mathfrak{P}$ of the entailment problem, taken in some fixed order, of no particular importance.

Definition 14. *Let $\gamma : \mathcal{L} \to \mathcal{L}$ be a total mapping. A structure $(\mathfrak{s}, \mathfrak{h}')$ is a* γ-expansion *(or simply an* expansion *if $\gamma = id$) of some structure $(\mathfrak{s}, \mathfrak{h})$, denoted $(\mathfrak{s}, \mathfrak{h}') \rhd_\gamma (\mathfrak{s}, \mathfrak{h})$, if there exist two disjoint heaps $\mathrm{main}(\mathfrak{h}')$ and $\mathrm{aux}(\mathfrak{h}')$ such that $\mathfrak{h}' = \mathrm{main}(\mathfrak{h}') \uplus \mathrm{aux}(\mathfrak{h}')$ and the following hold:*

1. *for all $\ell_1, \ell_2 \in \mathrm{dom}(\mathrm{main}(\mathfrak{h}'))$, if $\gamma(\ell_1) = \gamma(\ell_2)$ then $\ell_1 = \ell_2$,*
2. *$\gamma(\mathrm{dom}(\mathrm{main}(\mathfrak{h}'))) = \mathrm{dom}(\mathfrak{h})$,*
3. *for each $\ell \in \mathrm{dom}(\mathrm{main}(\mathfrak{h}'))$, we have $\mathfrak{h}'(\ell) = \langle \vec{a}, \mathfrak{s}(\vec{w}), b_1^\ell, \ldots, b_\mu^\ell \rangle$, for some $b_1^\ell, \ldots, b_\mu^\ell \in \mathcal{L}$ and $\gamma(\vec{a}) = \mathfrak{h}(\gamma(\ell))$,*
4. *for each $\ell \in \mathrm{dom}(\mathrm{aux}(\mathfrak{h}'))$, we have $\mathfrak{h}'(\ell) = \vec{\bot}$ and there exists $\ell' \in \mathrm{dom}(\mathrm{main}(\mathfrak{h}'))$ such that $\mathrm{main}(\mathfrak{h}')(\ell')$ is of the form $\langle \vec{a}, \vec{\omega}, b_1^{\ell'}, \ldots, b_\mu^{\ell'} \rangle$ and $\ell = b_i^{\ell'}$, for some $i \in [\![1, \mu]\!]$. The element ℓ' is called the* connection of ℓ in $\mathfrak{h}'$ *and is denoted by $\mathrm{C}_{\mathfrak{h}'}(\ell)$*[2]

Let $(\mathfrak{s}, \mathfrak{h}')$ be a γ-expansion of $(\mathfrak{s}, \mathfrak{h})$ and let $\ell \in \mathrm{dom}(\mathrm{main}(\mathfrak{h}'))$ be a location. Since $\nu > 0$ and for all $i \in [\![1, \nu]\!]$, $\mathfrak{s}(w_i)$ occurs in $\mathfrak{h}'(\ell)$, and since we assume that $\mathfrak{s}(w_i) \neq \bot = \mathfrak{s}(\bot)$ for every $i \in [\![1, \nu]\!]$, necessarily $\mathrm{main}(\mathfrak{h}')(\ell) \neq \vec{\bot}$. This entails that the decomposition $\mathfrak{h}' = \mathrm{main}(\mathfrak{h}') \uplus \mathrm{aux}(\mathfrak{h}')$ is unique : $\mathrm{dom}(\mathrm{main}(\mathfrak{h}'))$ and $\mathrm{dom}(\mathrm{aux}(\mathfrak{h}'))$ are the set of locations ℓ in $\mathrm{dom}(\mathfrak{h}')$ such that $\mathfrak{h}'(\ell) \neq \vec{\bot}$ and $\mathfrak{h}'(\ell) = \vec{\bot}$, respectively.

Proposition 15. *For all stores $\mathfrak{s}$ and functions γ, we have $(\mathfrak{s}, \emptyset) \rhd_\gamma (\mathfrak{s}, \emptyset)$.*

Lemma 16. *Consider a total mapping $\gamma : \mathcal{L} \to \mathcal{L}$, a store $\mathfrak{s}$ and two heaps $\mathfrak{h}$ and $\mathfrak{h}'$, such that $(\mathfrak{s}, \mathfrak{h}') \rhd_\gamma (\mathfrak{s}, \mathfrak{h})$. We have $\mathfrak{h}' = \emptyset$ if and only if $\mathfrak{h} = \emptyset$.*

Proof. Since $(\mathfrak{s}, \mathfrak{h}') \rhd_\gamma (\mathfrak{s}, \mathfrak{h})$, following the notations of Definition 14, $\mathfrak{h}'$ is of the form $\mathrm{main}(\mathfrak{h}') \uplus \mathrm{aux}(\mathfrak{h}')$. First assume $\mathfrak{h}' = \emptyset$. Then necessarily $\mathrm{main}(\mathfrak{h}') = \emptyset$ and by Condition 2 of Definition 14, we deduce that $\mathfrak{h} = \emptyset$. Conversely, if $\mathfrak{h} = \emptyset$, then by Condition 2 of Definition 14 we must have $\mathrm{main}(\mathfrak{h}') = \emptyset$ and by Condition 4, we also have $\mathrm{aux}(\mathfrak{h}') = \emptyset$. □

[2] Note that ℓ' does not depend on γ, and if several such locations exist then one is chosen arbitrarily.

Lemma 17. *Consider a store* $\mathfrak{s}$*, heaps* $\mathfrak{h}_1, \mathfrak{h}'_1, \mathfrak{h}_2, \mathfrak{h}'_2$ *and total mappings* $\gamma, \gamma_1, \gamma_2 : \mathcal{L} \to \mathcal{L}$*, such that the following hold:*

1. $\mathrm{dom}(\mathfrak{h}'_1) \cap \mathrm{dom}(\mathfrak{h}'_2) = \emptyset$ *and* $\mathrm{dom}(\mathfrak{h}_1) \cap \mathrm{dom}(\mathfrak{h}_2) = \emptyset$.
2. $(\mathfrak{s}, \mathfrak{h}'_i) \rhd_{\gamma_i} (\mathfrak{s}, \mathfrak{h}_i)$*, for* $i = 1, 2$,
3. *for all* $\ell \in \mathrm{loc}(\mathfrak{h}'_i)$, $\gamma(\ell) = \gamma_i(\ell)$*, for* $i = 1, 2$.

By letting $\mathfrak{h}' \stackrel{\mathsf{def}}{=} \mathfrak{h}'_1 \uplus \mathfrak{h}'_2$ *and* $\mathfrak{h} \stackrel{\mathsf{def}}{=} \mathfrak{h}_1 \uplus \mathfrak{h}_2$*, we have* $\mathfrak{h}' \rhd_\gamma \mathfrak{h}$.

Proof. By Point (2), we have $\mathfrak{h}'_i = \mathrm{main}(\mathfrak{h}'_i) \uplus \mathrm{aux}(\mathfrak{h}'_i)$, for $i = 1, 2$. By Point (1), $\mathrm{dom}(\mathrm{main}(\mathfrak{h}'_1)) \cap \mathrm{dom}(\mathrm{main}(\mathfrak{h}'_2)) = \mathrm{dom}(\mathrm{aux}(\mathfrak{h}'_1)) \cap \mathrm{dom}(\mathrm{aux}(\mathfrak{h}'_2)) = \emptyset$. Let $\mathrm{main}(\mathfrak{h}') = \mathrm{main}(\mathfrak{h}'_1) \uplus \mathrm{main}(\mathfrak{h}'_2)$ and $\mathrm{aux}(\mathfrak{h}') = \mathrm{aux}(\mathfrak{h}'_1) \uplus \mathrm{aux}(\mathfrak{h}'_2)$. We prove that these heaps satisfy the conditions of Definition 14:

1. Let $\ell_1, \ell_2 \in \mathrm{dom}(\mathrm{main}(\mathfrak{h}'))$. If $\ell_1 \in \mathrm{dom}(\mathrm{main}(\mathfrak{h}'_1))$ and $\ell_2 \in \mathrm{dom}(\mathrm{main}(\mathfrak{h}'_2))$, then $\gamma(\ell_1) = \gamma_1(\ell_1) \in \mathrm{dom}(\mathfrak{h}_1)$, and $\gamma(\ell_2) = \gamma_2(\ell_2) \in \mathrm{dom}(\mathfrak{h}_2)$. Since both sets are disjoint, it is impossible to have $\gamma(\ell_1) = \gamma(\ell_2)$. Else, if $\ell_1, \ell_2 \in \mathrm{dom}(\mathrm{main}(\mathfrak{h}'_1))$, then $\gamma(\ell_1) = \gamma_1(\ell_1)$ and $\gamma(\ell_2) = \gamma_1(\ell_2)$, thus $\gamma_1(\ell_1) = \gamma_1(\ell_2)$ and $\ell_1 = \ell_2$ follows from point (2). The other cases are symmetric.
2. We compute:
$$\begin{array}{rcl} \gamma(\mathrm{dom}(\mathrm{main}(\mathfrak{h}'))) & = & \gamma(\mathrm{dom}(\mathrm{main}(\mathfrak{h}'_1)) \uplus \mathrm{dom}(\mathrm{main}(\mathfrak{h}'_2))) \\ & = & \gamma_1(\mathrm{dom}(\mathrm{main}(\mathfrak{h}'_1))) \uplus \gamma_2(\mathrm{dom}(\mathrm{main}(\mathfrak{h}'_2))) \\ & = & \mathrm{dom}(\mathfrak{h}_1) \uplus \mathrm{dom}(\mathfrak{h}_2) \\ & = & \mathrm{dom}(\mathfrak{h}). \end{array}$$
3. Let $\ell \in \mathrm{dom}(\mathrm{main}(\mathfrak{h}'))$ and assume $\ell \in \mathrm{dom}(\mathrm{main}(\mathfrak{h}'_1))$; the other case is symmetric. Then by construction $\mathrm{main}(\mathfrak{h}')(\ell) = \mathrm{main}(\mathfrak{h}'_1)(\ell)$ is of the form $\langle \vec{a}, \mathfrak{s}(\vec{w}), b^\ell_1, \ldots, b^\ell_\mu \rangle$ where $\gamma_1(\vec{a}) = \mathfrak{h}_1(\gamma_1(\ell))$. Since $\vec{a}, \ell \in \mathrm{loc}(\mathfrak{h}'_1)$, we deduce that $\gamma(\vec{a}) = \mathfrak{h}_1(\gamma(\ell))$, hence $\gamma(\vec{a}) = \mathfrak{h}(\gamma(\ell))$.
4. Let $\ell \in \mathrm{dom}(\mathrm{aux}(\mathfrak{h}'))$. By definition, $\ell \in \mathrm{dom}(\mathrm{aux}(\mathfrak{h}'_i))$, for some $i = 1, 2$, and by Point 2, we deduce that $\mathfrak{h}'_i(\ell) = \vec{\bot}$ and that there exists a connection $\ell' \in \mathrm{dom}(\mathrm{main}(\mathfrak{h}'_i))$ of ℓ in $\mathfrak{h}'_i$. Then $\mathfrak{h}'(\ell) = \mathfrak{h}'_i(\ell) = \vec{\bot}$ and $\ell' \in \mathrm{dom}(\mathrm{main}(\mathfrak{h}'))$, hence ℓ' is also a connection of ℓ in $\mathfrak{h}'$.

□

Lemma 18. *Assume* $(\mathfrak{s}, \mathfrak{h}') \rhd_\gamma (\mathfrak{s}, \mathfrak{h})$ *and let* $\eta : \mathcal{L} \to \mathcal{L}$ *be a bijection such that* $\eta(\bot) = \bot$*. If* $\gamma' = \gamma \circ \eta^{-1}$*, then* $(\eta(\mathfrak{s}), \eta(\mathfrak{h}')) \rhd_{\gamma'} (\eta(\mathfrak{s}), \mathfrak{h})$.

Proof. Note that $\eta(\mathfrak{h}')$ is well-defined, since η is a bijection. We show that $\eta(\mathrm{main}(\mathfrak{h}'))$ and $\eta(\mathrm{aux}(\mathfrak{h}'))$ satisfy the conditions of Definition 14, thus proving the result:

1. Let $\ell, \ell' \in \mathrm{dom}(\eta(\mathrm{main}(\mathfrak{h}')))$ and assume that $\gamma'(\ell) = \gamma'(\ell')$. Then there exist $\ell_1, \ell_2 \in \mathrm{dom}(\mathrm{main}(\mathfrak{h}'))$ such that $\ell = \eta(\ell_1)$ and $\ell' = \eta(\ell_2)$, hence
$$\gamma(\ell_1) \;=\; \gamma \circ \eta^{-1}(\eta(\ell_1)) \;=\; \gamma'(\ell) \;=\; \gamma'(\ell') \;=\; \gamma \circ \eta^{-1}(\eta(\ell_2)) \;=\; \gamma(\ell_2),$$
so that $\ell_1 = \ell_2$, by point (1) of Definition 14, leading to $\ell = \ell'$.

2. We compute:

$$\begin{array}{rcl} \gamma'(\mathrm{dom}(\eta(\mathrm{main}(\mathfrak{h}')))) & = & \gamma'(\eta(\mathrm{dom}(\mathrm{main}(\mathfrak{h}')))) \quad = \quad \gamma(\mathrm{dom}(\mathrm{main}(\mathfrak{h}'))) \\ & = & \mathrm{dom}(\mathfrak{h}). \end{array}$$

3. Let $\ell \in \mathrm{dom}(\eta(\mathrm{main}(\mathfrak{h}')))$. Then there exists $\ell_1 \in \mathrm{dom}(\mathrm{main}(\mathfrak{h}'))$ such that $\ell = \eta(\ell_1)$, and $\mathrm{main}(\mathfrak{h}')(\ell_1)$ is of the form $\langle \vec{a}, \mathfrak{s}(\vec{w}), b_1, \ldots, b_\mu \rangle$, for some $b_i \in \mathcal{L}$ $(1 \leq i \leq \mu)$ and $\gamma(\vec{a}) = \mathfrak{h}(\gamma(\ell_1))$. Hence, $\eta(\mathrm{main}(\mathfrak{h}'))(\ell) = \langle \eta(\vec{a}), \eta(\mathfrak{s})(\vec{w}), \eta(b_1), \ldots, \eta(b_\mu) \rangle$ and

$$\gamma'(\eta(\vec{a})) \;=\; \gamma(\vec{a}) \;=\; \mathfrak{h}(\gamma(\ell_1)) \;=\; \mathfrak{h}(\gamma'(\ell)).$$

4. Let $\ell \in \mathrm{dom}(\eta(\mathrm{aux}(\mathfrak{h}')))$. Then there exists $\ell_2 \in \mathrm{dom}(\mathrm{aux}(\mathfrak{h}'))$ such that $\ell = \eta(\ell_2)$, and since $\eta(\bot) = \bot$, we have $\eta(\mathrm{aux}(\mathfrak{h}'))(\ell) = \eta(\mathrm{aux}(\mathfrak{h}')(\ell_2)) = \eta(\vec{\bot}) = \vec{\bot}$. By hypothesis ℓ_2 admits a connection ℓ' in $\mathfrak{h}'$, and it is straightforward to check that $\eta(\ell')$ is a connection of ℓ in $\eta(\mathfrak{h}')$.

□

Lemma 19. *Assume* $(\mathfrak{s}, \mathfrak{h}') \triangleright_{id} (\mathfrak{s}, \mathfrak{h})$*, let* $D \subseteq \mathrm{dom}(\mathfrak{h}')$ *and consider* $\mathfrak{h}'_1$ *(resp.* $\mathfrak{h}_1$*), the restriction of* $\mathfrak{h}'$ *(resp.* $\mathfrak{h}$*) to* D*. If every location in* $\mathrm{dom}(\mathrm{aux}(\mathfrak{h}')) \cap D$ *has a connection in* $\mathfrak{h}'_1$*, then* $(\mathfrak{s}, \mathfrak{h}'_1) \triangleright_{id} (\mathfrak{s}, \mathfrak{h}_1)$*.*

Proof. We check the conditions of Definition 14:

1. Trivial.
2. Since $\mathrm{dom}(\mathrm{main}(\mathfrak{h}')) = \mathrm{dom}(\mathfrak{h})$, we obtain $\mathrm{dom}(\mathrm{main}(\mathfrak{h}'_1)) = \mathrm{dom}(\mathrm{main}(\mathfrak{h}')) \cap D = \mathrm{dom}(\mathfrak{h}) \cap D = \mathrm{dom}(\mathfrak{h}_1)$.
3. Since Point (3) holds for $\mathrm{main}(\mathfrak{h}')$, it also holds for $\mathrm{main}(\mathfrak{h}'_1)$.
4. Let $\ell \in \mathrm{dom}(\mathrm{aux}(\mathfrak{h}'_1)) = \mathrm{dom}(\mathrm{aux}(\mathfrak{h}')) \cap D$. We have $\mathfrak{h}'_1(\ell) = \mathfrak{h}'(\ell) = \vec{\bot}$. Further, ℓ has a connection in $\mathfrak{h}'_1$ by hypothesis.

□

Definition 20. *Given a heap* $\mathfrak{h}'$*, we denote by* $\mathrm{trunc}(\mathfrak{h}')$ *the heap* $\mathfrak{h}$ *defined as follows:* $\mathrm{dom}(\mathfrak{h}) \stackrel{\text{def}}{=} \mathrm{dom}(\mathfrak{h}') \setminus \{\ell \in \mathrm{dom}(\mathfrak{h}') \mid \mathfrak{h}'(\ell) = \vec{\bot}\}$ *and for all* $\ell \in \mathrm{dom}(\mathfrak{h})$*, if* $\mathfrak{h}'(\ell) = (\ell_1, \ldots, \ell_{\kappa+\nu+\mu})$*, then* $\mathfrak{h}(\ell) \stackrel{\text{def}}{=} (\ell_1, \ldots, \ell_\kappa)$*.*

Example 21. *Assume that* $\mathcal{L} = \mathbb{N}$*,* $\nu = \mu = 1$*. Let* $\mathfrak{s}$ *be a store such that* $\mathfrak{s}(w_1) = 0$*. We consider:*

$$\begin{array}{rcl} \mathfrak{h} & \stackrel{\text{def}}{=} & \{\langle 1, 2\rangle, \langle 2, 2\rangle\}, \\ \mathfrak{h}'_1 & \stackrel{\text{def}}{=} & \{\langle 1, (2, 0, 1)\rangle, \langle 2, (2, 0, 3)\rangle, \langle 3, (\bot, \bot, \bot)\rangle\}, \\ \mathfrak{h}'_2 & \stackrel{\text{def}}{=} & \{\langle 1, (3, 0, 1)\rangle, \langle 2, (4, 0, 3)\rangle, \langle 3, (\bot, \bot, \bot)\rangle\}. \end{array}$$

We have $(\mathfrak{s}, \mathfrak{h}'_1) \triangleright_{id} (\mathfrak{s}, \mathfrak{h})$ *and* $(\mathfrak{s}, \mathfrak{h}'_2) \triangleright_{\gamma} (\mathfrak{s}, \mathfrak{h})$*, with* $\gamma \stackrel{\text{def}}{=} \{\langle 1, 1\rangle, \langle 2, 2\rangle, \langle 3, 2\rangle, \langle 4, 2\rangle\}$*. Also,* $\mathrm{trunc}(\mathfrak{h}'_1) = \{\langle 1, 2\rangle, \langle 2, 2\rangle\} = \mathfrak{h}$ *and* $\mathrm{trunc}(\mathfrak{h}'_2) = \{\langle 1, 3\rangle, \langle 2, 4\rangle\}$*.* ■

Lemma 22. *If* $(\mathfrak{s}, \mathfrak{h}') \triangleright_{\gamma} (\mathfrak{s}, \mathfrak{h})$ *then* $\mathfrak{h} = \gamma(\mathrm{trunc}(\mathfrak{h}'))$*.*

Proof. Since $(\mathfrak{s}, \mathfrak{h}') \rhd_\gamma (\mathfrak{s}, \mathfrak{h})$, the restriction of γ to $\mathrm{dom}(\mathrm{main}(\mathfrak{h}'))$ is injective and $\gamma(\mathrm{dom}(\mathrm{main}(\mathfrak{h}'))) = \mathrm{dom}(\mathfrak{h})$, by Point (2) of Definition 14. Furthermore, by Definition 20, $\mathrm{dom}(\mathrm{trunc}(\mathfrak{h}')) = \mathrm{dom}(\mathrm{main}(\mathfrak{h}'))$. Thus $\mathrm{dom}(\gamma(\mathrm{trunc}(\mathfrak{h}'))) = \gamma(\mathrm{dom}(\mathrm{trunc}(\mathfrak{h}'))) = \gamma(\mathrm{dom}(\mathrm{main}(\mathfrak{h}'))) = \mathrm{dom}(\mathfrak{h})$. Moreover, for any $\ell \in \mathrm{dom}(\mathrm{main}(\mathfrak{h}'))$, we have $\mathfrak{h}'(\ell) = \langle \vec{a}, \mathfrak{s}(\vec{w}), b_1^\ell, \ldots, b_\mu^\ell \rangle$ and $\mathfrak{h}(\gamma(\ell)) = \gamma(\vec{a})$, for some $\vec{a} \in \mathcal{L}^\kappa$ and $b_1^\ell, \ldots, b_\mu^\ell \in \mathcal{L}$, thus by Definition 20, $\gamma(\mathrm{trunc}(\mathfrak{h}')(\ell)) = \gamma(\vec{a}) = \gamma(\mathfrak{h}(\ell))$. □

Corollary 23. *If* $(\mathfrak{s}, \mathfrak{h}') \rhd_\gamma (\mathfrak{s}, \mathfrak{h})$, *then* $(\mathfrak{s}, \mathfrak{h}') \rhd_{id} (\mathfrak{s}, \mathrm{trunc}(\mathfrak{h}'))$.

The converse of Lemma 22 does not hold in general but it holds under some additional conditions:

Lemma 24. *Consider a store* $\mathfrak{s}$, *let* $\mathfrak{h}'$ *be a heap and let* $\mathfrak{h} \stackrel{\text{def}}{=} \mathrm{trunc}(\mathfrak{h}')$. *Let* $D_2 \stackrel{\text{def}}{=} \{\ell \in \mathrm{dom}(\mathfrak{h}') \mid \mathfrak{h}'(\ell) = \vec{\bot}\}$ *and* $D_1 \stackrel{\text{def}}{=} \mathrm{dom}(\mathfrak{h}') \setminus D_2$. *Assume that:*

1. *for every location* $\ell \in D_1$, $\mathfrak{h}(\ell)$ *is of the form* $(\ell_1, \ldots, \ell_\kappa)$ *and* $\mathfrak{h}'(\ell)$ *is of the form* $(\ell_1, \ldots, \ell_\kappa, \mathfrak{s}(\vec{w}), \ell'_1, \ldots, \ell'_\mu)$;
2. *every location* $\ell \in D_2$ *has a connection in* $\mathfrak{h}'$.

Then $(\mathfrak{s}, \mathfrak{h}') \rhd_{id} (\mathfrak{s}, \mathfrak{h})$.

Proof. It is straightforward to check that Conditions 1 and 2 of Definition 14 hold, with $\mathrm{main}(\mathfrak{h}')$ (resp. $\mathrm{aux}(\mathfrak{h}')$) defined as the restriction of $\mathfrak{h}'$ to D_1 (resp. D_2). Condition 3 follows immediately from Point 1 and from the definition of $\mathrm{trunc}(\mathfrak{h}')$. Condition 4 holds by Point 2. □

Lemma 25. *Let* $\mathfrak{h}'_1, \mathfrak{h}'_2$ *be disjoint heaps and let* $\mathfrak{h}_i \stackrel{\text{def}}{=} \mathrm{trunc}(\mathfrak{h}'_i)$, *for* $i = 1, 2$. *Then* $\mathfrak{h}_1$ *and* $\mathfrak{h}_2$ *are disjoint and* $\mathrm{trunc}(\mathfrak{h}'_1 \uplus \mathfrak{h}'_2) = \mathfrak{h}_1 \uplus \mathfrak{h}_2$.

Proof. By definition, $\mathrm{dom}(\mathfrak{h}_i) \subseteq \mathrm{dom}(\mathfrak{h}'_i)$ hence since $\mathfrak{h}'_1, \mathfrak{h}'_2$ are disjoint, $\mathfrak{h}_1$ and $\mathfrak{h}_2$ are also disjoint. and $\mathfrak{h} \stackrel{\text{def}}{=} \mathfrak{h}_1 \uplus \mathfrak{h}_2$. Let D be the set of locations ℓ such that $\mathfrak{h}'_1(\ell) = \vec{\bot}$ or $\mathfrak{h}'_2(\ell) = \vec{\bot}$. By Definition 20, we have $\mathrm{dom}(\mathfrak{h}_i) = \mathrm{dom}(\mathfrak{h}'_i) \setminus D$, hence $\mathrm{dom}(\mathfrak{h}_1 \uplus \mathfrak{h}_2) = \mathrm{dom}(\mathfrak{h}'_1 \uplus \mathfrak{h}'_2) \setminus D = \mathrm{dom}(\mathrm{trunc}(\mathfrak{h}'_1 \uplus \mathfrak{h}'_2))$. It is clear that this entails that $\mathrm{trunc}(\mathfrak{h}'_1 \uplus \mathfrak{h}'_2) = \mathfrak{h}_1 \uplus \mathfrak{h}_2$. □

4.2 Transforming The Consequent

We first describe the transformation for the right-hand side of the entailment problem, as this transformation is simpler. We recall that $\vec{w} = (w_1, \ldots, w_\nu)$ denotes the vector of free variables occurring in the problem, which is assumed to be fixed throughout this section and that $\{w_1, \ldots, w_\nu, \bot\} \subseteq \mathrm{dom}(\mathfrak{s})$, for every store $\mathfrak{s}$ considered in this section. Moreover, we assume w.l.o.g. that $w_1, \ldots, w_\nu$ do not occur in the considered SID $\mathcal{R}$.

Definition 26. *We associate each* n*-ary predicate* $p \in \mathcal{P}_r$ *with a new predicate* $\widehat{p}$ *of arity* $n+\nu$. *We denote by* $\widehat{\alpha}$ *the formula obtained from* α *by replacing every predicate atom* $p(x_1, \ldots, x_n)$ *by* $\widehat{p}(x_1, \ldots, x_n, \vec{w})$.

Definition 27. *We denote by* $\widehat{\mathcal{R}}$ *the set of rules of the form:*

$$\widehat{p}(x_1, \ldots, x_n, \vec{w}) \Leftarrow x_1 \mapsto (y_1, \ldots, y_\kappa, \vec{w}, z_1, \ldots, z_\mu)\sigma * \widehat{\rho}\sigma * \xi_I * \chi_\sigma$$

where:

- $p(x_1, \ldots, x_n) \Leftarrow x_1 \mapsto (y_1, \ldots, y_\kappa) * \rho$ *is a rule in* $\mathcal{R}$ *with* $p \in \mathcal{P}_r$,
- $\{z_1, \ldots, z_\mu\}$ *is a set of variables not occurring in* $\mathrm{fv}(\rho) \cup \{x_1, \ldots, x_n, y_1, \ldots, y_\kappa, w_1, \ldots, w_\nu\}$,
- σ *is a substitution with* $\mathrm{dom}(\sigma) \subseteq \mathrm{fv}(\rho) \setminus \{x_1\}$ *and* $\mathrm{rng}(\sigma) \subseteq \{w_1, \ldots, w_\nu\}$,
- $\xi_I \stackrel{\text{def}}{=} *_{i \in I} \underline{\bot}(z_i)$, *with* $I \subseteq \{1, \ldots, \mu\}$,
- $\chi_\sigma \stackrel{\text{def}}{=} *_{x \in \mathrm{dom}(\sigma)} x \approx x\sigma$.

We denote by $\mathcal{R}_r$ *the set of rules in* $\widehat{\mathcal{R}}$ *that are connected*[3].

Note that the free variables $\vec{w}$ are added as parameters in the rules above (instead of some vector of fresh variables $\vec{\omega}$). This is for the sake of clarity, since these parameters $\vec{\omega}$ will be systematically mapped to $\vec{w}$.

Example 28. *Assume that* $\psi = \exists x \,.\, p(x, w_1)$, *with* $\nu = 1$, $\mu = 1$ *and* $\lambda(p) = \{2\}$. *Assume also that* p *is associated with the rule:*

$$p(u_1, u_2) \quad \Leftarrow \quad u_1 \mapsto u_1 * q(u_2).$$

Observe that the rule is λ*-connected, but not connected. Then* $\mathrm{dom}(\sigma) \subseteq \{u_2\}$, $\mathrm{rng}(\sigma) \subseteq \{w_1\}$ *and* $I \subseteq \{1\}$, *so that* $\widehat{\mathcal{R}}$ *contains the following rules:*

$$\begin{array}{lllll}
(1) & p(u_1, u_2, w_1) & \Leftarrow & u_1 \mapsto (u_1, w_1, z_1) * q(u_2) \\
(2) & p(u_1, u_2, w_1) & \Leftarrow & u_1 \mapsto (u_1, w_1, z_1) * q(u_2) * \underline{\bot}(z_1) \\
(3) & p(u_1, u_2, w_1) & \Leftarrow & u_1 \mapsto (u_1, w_1, z_1) * q(w_1) * u_2 \approx w_1 \\
(4) & p(u_1, u_2, w_1) & \Leftarrow & u_1 \mapsto (u_1, w_1, z_1) * q(w_1) * \underline{\bot}(z_1) * u_2 \approx w_1
\end{array}$$

Rules (1) and (2) are not connected, hence do not occur in $\mathcal{R}_r$. *Rules (3) and (4) are both connected, hence occur in* $\mathcal{R}_r$. *Note that (4) is established, but (3) is not.*

We now relate $\mathcal{R}$ and $\mathcal{R}_r$.

Lemma 29. *If* $(\mathfrak{s}, \mathfrak{h}') \models_{\mathcal{R}_r} \widehat{\alpha}$ *then for all* $\ell \in \mathrm{dom}(\mathrm{aux}(\mathfrak{h}'))$, ℓ *has a connection in* $\mathfrak{h}'$.

Proof. Let $\ell \in \mathrm{dom}(\mathrm{aux}(\mathfrak{h}'))$. By definition $\mathfrak{h}'(\ell) = \vec{\bot}$. Observe that ℓ cannot be allocated by the points-to atom of a rule in $\mathcal{R}_r$, since otherwise, by Definition 27, it would be mapped to a tuple containing $\mathfrak{s}(w_1), \ldots, \mathfrak{s}(w_\nu)$, hence to a tuple distinct from $\vec{\bot}$ since $\nu > 0$ and $\mathfrak{s}(w_i) \neq \bot$ for $i = 1, \ldots, \nu$. Consequently, ℓ must be allocated by a predicate $\underline{\bot}(z_i)$ invoked in a rule in Definition 27. Since z_i also occurs as one of the last μ components on the right-hand side of the points-to atom of the considered rule, necessarily ℓ has a connection in $\mathfrak{h}'$. □

[3] Note that all the rules in $\widehat{\mathcal{R}}$ are progressing.

Lemma 30. *Let α be a formula that is λ-restricted w.r.t. $\{w_1, \ldots, w_\nu\}$ and contains no points-to atom, with $\mathcal{P}(\alpha) \subseteq \mathcal{P}_r$. Given a store $\mathfrak{s}$ and two heaps $\mathfrak{h}$ and $\mathfrak{h}'$, such that $(\mathfrak{s}, \mathfrak{h}') \triangleright_{id} (\mathfrak{s}, \mathfrak{h})$, we have $(\mathfrak{s}, \mathfrak{h}') \models_{\mathcal{R}_r} \widehat{\alpha}$ if and only if $(\mathfrak{s}, \mathfrak{h}) \models_{\mathcal{R}} \alpha$.*

Proof. The proof is by induction on the pair $(|\mathfrak{h}|, |\alpha|)$, using the lexicographic order. We distinguish several cases, depending on the form of α.

- If α is of the form $x \approx y$ then by Definition 26, $\widehat{\alpha} = \alpha$. We have $(\mathfrak{s}, \mathfrak{h}') \models_{\mathcal{R}_r} \widehat{\alpha}$ iff $\mathfrak{h}' = \emptyset$ and $\mathfrak{s}(x) = \mathfrak{s}(y)$, and $(\mathfrak{s}, \mathfrak{h}) \models_{\mathcal{R}} \alpha$ iff $\mathfrak{h} = \emptyset$ and $\mathfrak{s}(x) = \mathfrak{s}(y)$. By Lemma 16, $\mathfrak{h}' = \emptyset$ iff $\mathfrak{h} = \emptyset$, hence the result.
- The proof is similar if α is of the form $x \not\approx y$.
- Assume that $\alpha = \alpha_1 \vee \alpha_2$. By construction we have $\widehat{\alpha} = \widehat{\alpha}_1 \vee \widehat{\alpha}_2$. Now, $(\mathfrak{s}, \mathfrak{h}') \models_{\mathcal{R}_r} \widehat{\alpha}$ if and only if $(\mathfrak{s}, \mathfrak{h}') \models_{\mathcal{R}_r} \widehat{\alpha_i}$, for some $i \in \{1, 2\}$. By the induction hypothesis, this is equivalent to $(\mathfrak{s}, \mathfrak{h}) \models_{\mathcal{R}_r} \alpha_i$, for some $i \in \{1, 2\}$, i.e. equivalent to $(\mathfrak{s}, \mathfrak{h}) \models_{\mathcal{R}_r} \alpha$.
- Assume that $\alpha = \alpha_1 * \alpha_2$. Then it is straightforward to check that $\widehat{\alpha} = \widehat{\alpha}_1 * \widehat{\alpha}_2$. If $(\mathfrak{s}, \mathfrak{h}') \models_{\mathcal{R}_r} \widehat{\alpha}$ then there exists $\mathfrak{h}'_1, \mathfrak{h}'_2$ such that $\mathfrak{h}' = \mathfrak{h}'_1 \uplus \mathfrak{h}'_2$ and $(\mathfrak{s}, \mathfrak{h}'_i) \models_{\mathcal{R}_r} \widehat{\alpha_i}$ for $i = 1, 2$. Let $\mathfrak{h}_i = \mathrm{trunc}(\mathfrak{h}'_i)$. By Lemma 29, every location in $\mathrm{aux}(\mathfrak{h}'_i)$ has a connection in $\mathfrak{h}'_i$, thus, by Lemma 19 (applied with $D = \mathrm{dom}(\mathfrak{h}'_i)$), we deduce that $(\mathfrak{s}, \mathfrak{h}'_i) \triangleright_{id} (\mathfrak{s}, \mathfrak{h}_i)$. By the induction hypothesis, we deduce that $(\mathfrak{s}, \mathfrak{h}_i) \models_{\mathcal{R}} \alpha_i$, and by Lemma 25, $\mathfrak{h} = \mathfrak{h}_1 \uplus \mathfrak{h}_2$. Thus $(\mathfrak{s}, \mathfrak{h}) \models_{\mathcal{R}} \alpha$.
  Conversely, assume that $(\mathfrak{s}, \mathfrak{h}) \models_{\mathcal{R}} \alpha$. Then there exists $\mathfrak{h}_1, \mathfrak{h}_2$ such that $\mathfrak{h} = \mathfrak{h}_1 \uplus \mathfrak{h}_2$ and $(\mathfrak{s}, \mathfrak{h}_i) \models_{\mathcal{R}} \alpha_i$ for $i = 1, 2$. Let $D \stackrel{\text{def}}{=} \mathrm{dom}(\mathfrak{h}') \setminus \mathrm{dom}(\mathfrak{h})$, $D_1 \stackrel{\text{def}}{=} \{d \in D \mid \mathrm{C}_{\mathfrak{h}'}(d) \in \mathrm{dom}(\mathfrak{h}_1)\}$ and $D_2 \stackrel{\text{def}}{=} D \setminus D_1$. For $i = 1, 2$, let $\mathfrak{h}'_i$ be the restriction of $\mathfrak{h}'$ to $\mathrm{dom}(\mathfrak{h}_i) \cup D_i$. Since $(\mathfrak{s}, \mathfrak{h}') \triangleright_{id} (\mathfrak{s}, \mathfrak{h})$ by hypothesis, it is straightforward to verify that $(\mathfrak{s}, \mathfrak{h}'_i) \triangleright_{id} (\mathfrak{s}, \mathfrak{h}_i)$, and by the induction hypothesis, we deduce that $(\mathfrak{s}, \mathfrak{h}'_i) \models_{\mathcal{R}_r} \widehat{\alpha_i}$ for $i = 1, 2$. By construction, D_1, D_2, $\mathrm{dom}(\mathfrak{h}_1)$ and $\mathrm{dom}(\mathfrak{h}_2)$ are pairwise disjoint and $\mathrm{dom}(\mathfrak{h}_1) \cup D_1 \cup \mathrm{dom}(\mathfrak{h}_2) \cup D_2 = \mathrm{dom}(\mathfrak{h}) \cup D = \mathrm{dom}(\mathfrak{h}')$, hence $\mathfrak{h}' = \mathfrak{h}'_1 \uplus \mathfrak{h}'_2$. We conclude that $(\mathfrak{s}, \mathfrak{h}') \models_{\mathcal{R}_r} \widehat{\alpha}$.
- Assume that $\alpha = p(u_1, \ldots, u_n)$, so that $\widehat{\alpha} = \widehat{p}(u_1, \ldots, u_n, \vec{w})$. If $(\mathfrak{s}, \mathfrak{h}') \models_{\mathcal{R}_r} \widehat{\alpha}$, then $\mathcal{R}_r$ contains a rule of the form

$$\widehat{p}(x_1, \ldots, x_n, \vec{w}) \Leftarrow x_1 \mapsto (y_1, \ldots, y_\kappa, \vec{w}, z_1, \ldots, z_\mu) * \widehat{\rho}\sigma * \xi_I * \chi_\sigma,$$

  satisfying the conditions of Definition 27, and there exists an extension $\mathfrak{s}_e$ of $\mathfrak{s}$ such that $(\mathfrak{s}_e, \mathfrak{h}') \models_{\mathcal{R}_r} u_1 \mapsto (y_1, \ldots, y_\kappa, \vec{w}, z_1, \ldots, z_\mu)\theta * \widehat{\rho}\sigma\theta * \xi_I\theta * \chi_\sigma\theta$, with $\theta \stackrel{\text{def}}{=} \{\langle x_i, u_i \rangle \mid i \in [\![1, n]\!]\}$. Then, necessarily, $\mathcal{R}$ contains a rule:

$$p(x_1, \ldots, x_n) \Leftarrow x_1 \mapsto (y_1, \ldots, y_\kappa) * \rho \quad (\ddagger)$$

  Let $\mathfrak{h}'_1$ be the restriction of $\mathfrak{h}'$ to $\mathrm{dom}(\mathfrak{h}) \setminus (\{\mathfrak{s}(u_1)\} \cup \{\mathfrak{s}_e(z_i) \mid i \in I\})$ and let $\mathfrak{h}_1 \stackrel{\text{def}}{=} \mathrm{trunc}(\mathfrak{h}'_1)$. By definition of $\mathfrak{h}'_1$, we have $(\mathfrak{s}_e, \mathfrak{h}'_1) \models_{\mathcal{R}_r} \widehat{\rho}\sigma\theta$. By hypothesis α is λ-restricted w.r.t. $\{w_1, \ldots, w_\nu\}$, thus $\mathsf{V}_\lambda(p(u_1, \ldots, u_n)) \subseteq \{w_1, \ldots, w_\nu\}$ and, since the entailment problem under consideration is safe, rule $(\ddagger)$ is necessarily λ-restricted by Point (3) of Definition 9. Thus,

$\rho\theta$ must be λ-restricted w.r.t. $\{w_1, \ldots, w_\nu\}$, by Lemma 11. Since the image of σ is contained in $\{w_1, \ldots, w_\nu\}$, we deduce that $\rho\sigma\theta$ is also λ-restricted w.r.t. $\{w_1, \ldots, w_\nu\}$. By the induction hypothesis, this entails that $(\mathfrak{s}_e, \mathfrak{h}_1) \models_{\mathcal{R}} \rho\sigma\theta$. Since $(\mathfrak{s}_e, \emptyset) \models_{\mathcal{R}} \chi_\sigma\theta$ by definition of χ_σ — see Definition 27, we have $\mathfrak{s}_e(x\theta) = \mathfrak{s}_e(x\sigma\theta)$ for every variable $x \in \mathrm{dom}(\sigma)$. But the latter equality trivially holds for every variable $x \notin \mathrm{dom}(\sigma)$, hence replacing all variables $x\sigma\theta$ occurring in $\rho\sigma\theta$ by $x\theta$ preserves the truth value of the formula in $(\mathfrak{s}_e, \mathfrak{h}_1)$. Consequently $\rho\sigma\theta$ and $\rho\theta$ have the same truth value in $(\mathfrak{s}_e, \mathfrak{h}_1)$, and thus $(\mathfrak{s}_e, \mathfrak{h}_1) \models_{\mathcal{R}} \rho\theta$.
Let $\mathfrak{h}'_u$ denote the restriction of $\mathfrak{h}'$ to $\{\mathfrak{s}(u_1)\}$. By construction $\mathfrak{h}'(\mathfrak{s}_e(z_i)) = \vec{\bot}$ for every $i \in I$, hence by Lemmas 22 and 25 we have

$$\mathfrak{h} = \mathrm{trunc}(\mathfrak{h}') = \mathrm{trunc}(\mathfrak{h}'_1 \uplus \mathfrak{h}'_u) = \mathfrak{h}_1 \uplus \mathrm{trunc}(\mathfrak{h}'_u).$$

Since $\mathrm{trunc}(\mathfrak{h}'_u) = \{(\mathfrak{s}_e(u_1), \langle \mathfrak{s}_e(y_1\theta), \ldots, \mathfrak{s}_e(y_\kappa\theta)\rangle)\}$, we deduce that $(\mathfrak{s}_e, \mathfrak{h}) \models_{\mathcal{R}} u_1 \mapsto (y_1, \ldots, y_\kappa)\theta * \rho\theta$, and therefore that $(\mathfrak{s}, \mathfrak{h}) \models_{\mathcal{R}} p(u_1, \ldots, u_n)$.
Conversely, assuming that $(\mathfrak{s}, \mathfrak{h}) \models_{\mathcal{R}} p(u_1, \ldots, u_n)$, $\mathcal{R}$ contains a rule:

$$p(x_1, \ldots, x_n) \Leftarrow x_1 \mapsto (y_1, \ldots, y_\kappa) * \rho \ (\dagger)$$

and there exists an extension $\mathfrak{s}_e$ of $\mathfrak{s}$ such that $(\mathfrak{s}_e, \mathfrak{h}) \models_{\mathcal{R}} u_1 \mapsto (y_1, \ldots, y_\kappa)\theta * \rho\theta$, where $\theta \stackrel{\text{def}}{=} \{\langle x_i, u_i\rangle \mid i \in [\![1, n]\!]\}$. Thus we must have:

$$\mathfrak{h}'(\mathfrak{s}_e(u_1)) \ = \ \langle \mathfrak{s}_e(y_1\theta), \ldots, \mathfrak{s}_e(y_\kappa\theta), \mathfrak{s}_e(\vec{w}), \ell_1, \ldots, \ell_\mu\rangle$$

for some locations $\ell_1, \ldots, \ell_\mu$. Since the variables $z_1, \ldots, z_\mu$ do not occur in $\{x_1, \ldots, x_n, y_1, \ldots, y_\kappa, w_1, \ldots, w_\nu\} \cup \mathrm{fv}(\rho)$ by hypothesis (see Definition 27), we may assume, w.l.o.g., that $\mathfrak{s}_e(z_i) = \ell_i$. Let $I \stackrel{\text{def}}{=} \{i \in [\![1, \mu]\!] \mid \mathfrak{h}'(\ell_i) = \vec{\bot}\}$ and let σ be the substitution defined as follows:

- $\mathrm{dom}(\sigma) = (\mathrm{fv}(\rho) \setminus \{x_1\}) \cap \{x \mid \mathfrak{s}_e(x\theta) \in \{\mathfrak{s}_e(w_1), \ldots, \mathfrak{s}_e(w_\mu)\}\}$, and
- for every variable $x \in \mathrm{fv}(\rho) \setminus \{x_1\}$, such that $\mathfrak{s}_e(x\theta) = \mathfrak{s}_e(w_i)$ for some $i \in [\![1, \nu]\!]$, we let $\sigma(x) \stackrel{\text{def}}{=} w_i$; if several such values of i are possible, then one is chosen arbitrarily.

By construction (see again Definition 27), $\widehat{\mathcal{R}}$ contains the rule:

$$\widehat{p}(x_1, \ldots, x_n, \vec{w}) \Leftarrow x_1 \mapsto (y_1, \ldots, y_\kappa, \vec{w}, z_1, \ldots, z_\mu) * \widehat{\rho}\sigma * \xi_I * \chi_\sigma.$$

Let $\mathfrak{h}_1$ be the restriction of $\mathfrak{h}$ to $\mathrm{dom}(\mathfrak{h}) \setminus \{\mathfrak{s}(u_1)\}$, and let $\mathfrak{h}'_1$ be the restriction of $\mathfrak{h}'$ to $\mathrm{dom}(\mathfrak{h}') \setminus \{\mathfrak{s}(u_1), \ell_i \mid i \in I\}$. Since $(\mathfrak{s}, \mathfrak{h}') \triangleright_{id} (\mathfrak{s}, \mathfrak{h})$, no element ℓ_i with $i \in I$ can be in $\mathrm{dom}(\mathfrak{h})$. By Definition, every element of $\mathrm{dom}(\mathrm{aux}(\mathfrak{h}'_1))$ has a connection in $\mathfrak{h}'_1$ (since the locations ℓ_i with $i \in I$ are the only elements of $\mathrm{dom}(\mathrm{aux}(\mathfrak{h}'))$ whose connection is $\mathfrak{s}(u_1)$), and by Lemma 19 we deduce that $(\mathfrak{s}, \mathfrak{h}'_1) \triangleright_{id} (\mathfrak{s}, \mathfrak{h}_1)$.
Note that by hypothesis α is λ-restricted w.r.t. $\{w_1, \ldots, w_\nu\}$ and the entailment problem $\mathfrak{P}$ is safe, thus $\rho\theta$ is λ-restricted w.r.t. $\{w_1, \ldots, w_\nu\}$ by Lemma 11. By construction $\mathrm{rng}(\sigma) \subseteq \{w_1, \ldots, w_\nu\}$ thus necessarily, $\rho\sigma\theta$ must also be λ-restricted w.r.t. $\{w_1, \ldots, w_\nu\}$. Furthermore, since the

rule (†) is λ-connected, for every $q(x'_1, \ldots, x'_m)$ occurring in ρ, if $x'_1 \notin \{y_1, \ldots, y_\kappa\}$ then necessarily $x'_1\theta \in \mathsf{V}_\lambda(p(u_1, \ldots, u_n)) \subseteq \{w_1, \ldots, w_\nu\}$. By the definition of σ we deduce that $x'_1 \in \mathrm{dom}(\sigma)$, and the rule above must be connected and occur in $\mathcal{R}_r$ (note that we cannot have $x'_1 = x_1$, because all the rules are progressing, hence $\mathfrak{s}_e(x'_1\theta) \in \mathrm{dom}(\mathfrak{h}_1)$ and by definition $\mathfrak{s}(x_1\theta) \notin \mathrm{dom}(\mathfrak{h}_1)$).
Now $\rho\sigma\theta$ is λ-restricted w.r.t. $\{w_1, \ldots, w_\nu\}$, and since $(\mathfrak{s}_e, \mathfrak{h}_1) \models_{\mathcal{R}} \rho\theta$, by definition of σ we have $(\mathfrak{s}_e, \mathfrak{h}_1) \models_{\mathcal{R}} \rho\sigma\theta$. By the induction hypothesis, $(\mathfrak{s}_e, \mathfrak{h}'_1) \models_{\mathcal{R}_r} \widehat{\rho\sigma\theta}$. For every $i \in I$, we have $\mathfrak{h}'(\mathfrak{s}_e(z_i)) = \vec{\bot}$, and by definition of σ we have $\mathfrak{s}_e(x\theta) = \mathfrak{s}_e(x\sigma\theta)$ for every $x \in \mathrm{dom}(\sigma)$, thus $(\mathfrak{s}_e, \mathfrak{h}') \models_{\mathcal{R}_r} x_1 \mapsto (y_1, \ldots, y_\kappa, \vec{w}, z_1, \ldots, z_\mu)\theta * \rho\sigma\theta * \xi_I\theta * \chi_\sigma\theta$, hence $(\mathfrak{s}, \mathfrak{h}') \models_{\mathcal{R}_r} \widehat{p}(u_1, \ldots, u_n, \vec{w})$.

□

4.3 Transforming The Antecedent

We now describe the transformation operating on the left-hand side of the entailment problem. We recall that $\vec{w} = (w_1, \ldots, w_\nu)$ are the free variables occurring in the entailment problem and that μ is the number of existentially quantified variables in each rule of the considered SID. For technical convenience, we make the following assumption:

Assumption 31. *We assume that for every predicate $p \in \mathcal{P}_l$, every rule $p(x_1, \ldots, x_n) \Leftarrow \pi$ in $\mathcal{R}$ and every atom $q(x'_1, \ldots, x'_m)$ occurring in π, $x'_1 \notin \{x_1, \ldots, x_n\}$.*

This is without loss of generality since every variable $x'_1 \in \{x_1, \ldots, x_n\}$ can be replaced by a fresh variable z while adding the equation $z \approx x'_1$ to π. Note that the obtained SID may no longer be connected, but this is not problematic.

Definition 32. *We associate each pair (p, X) where p is a predicate symbol in $\mathcal{P}_l$ of arity n, and $X \subseteq [\![1, n]\!]$, with a new predicate p_X of arity $n + \nu$. A* decoration *of a formula α containing no points-to atom and such that $\mathcal{P}(\alpha) \subseteq \mathcal{P}_l$ is a formula obtained from α by replacing each predicate atom $\beta \stackrel{\mathsf{def}}{=} q(y_1, \ldots, y_m)$ in α by an atom of the form $q_{X_\beta}(y_1, \ldots, y_m, \vec{w})$, with $X_\beta \subseteq [\![1, m]\!]$. The set of decorations of a formula α is denoted by $D(\alpha)$.*

Note that the set of decorations of an atom α is always finite.

Proposition 33. *If an atom α' is a decoration of α and is of the form $\alpha'_1 * \alpha'_2$, then α is of the form $\alpha_1 * \alpha_2$ and for $i = 1, 2$, α'_i is a decoration of α_i.*

Definition 34. *We denote by $D(\mathcal{R})$ the set of rules of the form*

$$p_X(x_1, \ldots, x_n, \vec{w}) \Leftarrow x_1 \mapsto (y_1, \ldots, y_\kappa, \vec{w}, z_1, \ldots, z_\mu)\sigma * \rho' * \ast_{i \in I} \underline{\bot}(z_i),$$

where:

- $p(x_1, \ldots, x_n) \Leftarrow x_1 \mapsto (y_1, \ldots, y_\kappa) * \rho$ *is a rule in $\mathcal{R}$ and $X \subseteq [\![1, n]\!]$;*
- $\{z_1, \ldots, z_\mu\} = (\mathrm{fv}(\rho) \cup \{y_1, \ldots, y_\kappa\}) \setminus \{x_1, \ldots, x_n\}$,

- σ *is a substitution with domain* $\mathrm{dom}(\sigma) \subseteq \{z_1, \ldots, z_\mu\}$ *and range* $\mathrm{rng}(\sigma) \subseteq \{x_1, \ldots, x_n, w_1, \ldots, w_\nu, z_1, \ldots, z_\mu\}$*;*
- ρ' *is a decoration of* $\rho\sigma$*;*
- $I \subseteq \{1, \ldots, \mu\}$ *and* $\forall i \in I,\ z_i \notin \mathrm{dom}(\sigma)$.

Lemma 35. *If* $(\mathfrak{s}, \mathfrak{h}') \models_{\mathcal{R}_l} \widehat{\alpha}$ *then for all* $\ell \in \mathrm{dom}(\mathrm{aux}(\mathfrak{h}'))$, ℓ *has a connection in* $\mathfrak{h}'$.

Proof. The proof is similar to that of Lemma 29. □

Lemma 36. *Let* α *be a formula containing no points-to atom, with* $\mathcal{P}(\alpha) \subseteq \mathcal{P}_l$, *and let* α' *be a decoration of* α. *If* $(\mathfrak{s}, \mathfrak{h}') \models_{D(\mathcal{R})} \alpha'$ *and* $(\mathfrak{s}, \mathfrak{h}') \rhd_{id} (\mathfrak{s}, \mathfrak{h})$, *then* $(\mathfrak{s}, \mathfrak{h}) \models_{\mathcal{R}} \alpha$.

Proof. The proof is by induction on the pair $(|\mathfrak{h}|, |\alpha'|)$, using the lexicographic order. We distinguish several cases.

- If α' is of the form $x \approx y$ or $x \not\approx y$, then necessarily $\alpha = \alpha'$ and $\mathfrak{h}' = \emptyset$. Thus $\mathfrak{h} = \emptyset$ by Lemma 16 and $(\mathfrak{s}, \mathfrak{h}) \models_{\mathcal{R}} \alpha$.
- If α' is of the form $\alpha'_1 \vee \alpha'_2$ then α is of the form $\alpha_1 \vee \alpha_2$ where α'_i is a decoration of α_i. If $(\mathfrak{s}, \mathfrak{h}') \models_{D(\mathcal{R})} \alpha'$ then $(\mathfrak{s}, \mathfrak{h}') \models_{D(\mathcal{R})} \alpha'_i$ for some $i = 1, 2$, and by the induction hypothesis we deduce that $(\mathfrak{s}, \mathfrak{h}) \models_{\mathcal{R}} \alpha_i$, thus $(\mathfrak{s}, \mathfrak{h}) \models_{\mathcal{R}} \alpha$.
- If α' is of the form $\alpha'_1 * \alpha'_2$ then $\mathfrak{h}' = \mathfrak{h}'_1 \uplus \mathfrak{h}'_2$, with $(\mathfrak{s}, \mathfrak{h}'_i) \models_{D(\mathcal{R})} \alpha'_i$ for $i = 1, 2$, and by Lemma 33, α is of the form $\alpha_1 * \alpha_2$ where α'_i is a decoration of α_i. Let $\mathfrak{h}_i$ be the restriction of $\mathfrak{h}$ to the locations occurring in $\mathrm{dom}(\mathfrak{h}'_i)$. By Lemma 35, every element of $\mathrm{dom}(\mathrm{aux}(\mathfrak{h}'_i))$ has a connection in $\mathfrak{h}'_i$. Therefore, by Lemma 19, we deduce that $(\mathfrak{s}, \mathfrak{h}'_i) \rhd_{id} (\mathfrak{s}, \mathfrak{h}_i)$ and by the induction hypothesis we deduce that $(\mathfrak{s}, \mathfrak{h}_i) \models_{\mathcal{R}} \alpha_i$. By definition of $\rhd_{id}$, we have $\mathrm{dom}(\mathfrak{h}_i) = \mathrm{dom}(\mathfrak{h}'_i) \setminus \{\ell \mid \mathfrak{h}'(\ell) = \vec{\bot}\}$, and since $\mathrm{dom}(\mathfrak{h}'_1) \cap \mathrm{dom}(\mathfrak{h}'_2) = \emptyset$, $\mathfrak{h}_1$ and $\mathfrak{h}_2$ are disjoint. Furthermore, we have:

$$\begin{array}{rcl} \mathrm{dom}(\mathfrak{h}) & = & \mathrm{dom}(\mathfrak{h}') \setminus \{\ell \mid \mathfrak{h}'(\ell) = \vec{\bot}\} \\ & = & (\mathrm{dom}(\mathfrak{h}'_1) \cup \mathrm{dom}(\mathfrak{h}'_2)) \setminus \{\ell \mid \mathfrak{h}'(\ell) = \vec{\bot}\} \\ & = & (\mathrm{dom}(\mathfrak{h}'_1) \setminus \{\ell \mid \mathfrak{h}'(\ell) = \vec{\bot}\}) \cup (\mathrm{dom}(\mathfrak{h}'_2) \setminus \{\ell \mid \mathfrak{h}'(\ell) = \vec{\bot}\}) \\ & = & \mathrm{dom}(\mathfrak{h}_1) \cup \mathrm{dom}(\mathfrak{h}_2), \end{array}$$

  therefore $\mathfrak{h} = \mathfrak{h}_1 \uplus \mathfrak{h}_2$ and $(\mathfrak{s}, \mathfrak{h}) \models_{\mathcal{R}} \alpha_1 * \alpha_2 = \alpha$.
- If α' is of the form $p_X(u_1, \ldots, u_n, \vec{w})$, then $\alpha = p(u_1, \ldots, u_n)$. By definition $D(\mathcal{R})$ contains a rule

$$p_X(x_1, \ldots, x_n, \vec{w}) \Leftarrow x_1 \mapsto (y_1, \ldots, y_\kappa, \vec{w}, z_1, \ldots, z_\mu)\sigma * \rho' * \mathop{\ast}_{i \in I} \underline{\bot}(z_i)$$

  satisfying the conditions of Definition 34, and there exists an extension $\mathfrak{s}_e$ of $\mathfrak{s}$ with $(\mathfrak{s}_e, \mathfrak{h}') \models_{D(\mathcal{R})} u_1 \mapsto (y_1, \ldots, y_\kappa, \vec{w}, z_1, \ldots, z_\mu)\sigma\theta * \rho'\theta * \mathop{\ast}_{i \in I} \underline{\bot}(z_i)$, where $\theta \stackrel{\mathsf{def}}{=} \{\langle x_i, u_i \rangle \mid i \in [\![1, n]\!]\}$. In particular, $\mathcal{R}$ contains a rule $p(x_1, \ldots, x_n) \Leftarrow x_1 \mapsto (y_1, \ldots, y_\kappa) * \rho$, where ρ' is a decoration of $\rho\sigma$; note that $\rho'\theta$ is a decoration of $\rho\sigma\theta$.

Let $\mathfrak{h}'_1$ be the restriction of $\mathfrak{h}'$ to $\mathrm{dom}(\mathfrak{h}') \setminus (\{\mathfrak{s}(u_1)\} \cup \{\mathfrak{s}_e(z_i) \mid i \in I\})$ and let $\mathfrak{h}_1$ be the restriction of $\mathfrak{h}$ to $\mathrm{dom}(\mathfrak{h}) \setminus \{\mathfrak{s}(u_1)\}$. We have $(\mathfrak{s}_e, \mathfrak{h}'_1) \models_{D(\mathcal{R})} \rho'\theta$, with $|\mathfrak{h}'_1| < |\mathfrak{h}'|$. By Lemma 35, every element of $\mathrm{dom}(\mathrm{aux}(\mathfrak{h}'_1))$ has a connection in $\mathfrak{h}'_1$. Since $(\mathfrak{s}, \mathfrak{h}') \triangleright_{id} (\mathfrak{s}, \mathfrak{h})$, by Lemma 19 we have $(\mathfrak{s}_e, \mathfrak{h}'_1) \triangleright_{id} (\mathfrak{s}_e, \mathfrak{h}_1)$. Hence, by the induction hypothesis, we deduce that $(\mathfrak{s}_e, \mathfrak{h}_1) \models_{\mathcal{R}} \rho\sigma\theta$. Moreover we have

$$\mathfrak{h}'(u_1) \;=\; (\mathfrak{s}_e(y_1\sigma\theta), \ldots, \mathfrak{s}(y_\kappa\sigma\theta), \mathfrak{s}(\vec{w}), \mathfrak{s}(z_1\sigma\theta), \ldots, \mathfrak{s}(z_\mu\sigma\theta)),$$

and since $(\mathfrak{s}, \mathfrak{h}') \triangleright_{id} (\mathfrak{s}, \mathfrak{h})$, we deduce that $\mathfrak{h}(u_1) = (\mathfrak{s}_e(y_1\sigma\theta), \ldots, \mathfrak{s}(y_\kappa\sigma\theta))$. Consequently, $(\mathfrak{s}_e, \mathfrak{h}) \models_{\mathcal{R}} (x_1 \mapsto (y_1, \ldots, y_\kappa) * \rho)\theta$, and therefore $(\mathfrak{s}, \mathfrak{h}) \models_{\mathcal{R}} p(x_1\theta, \ldots, x_n\theta) = p(u_1, \ldots, u_n)$.

□

At this point, the set X for predicate symbol p_X is of little interest: atoms are simply decorated with arbitrary sets. However, we shall restrict the considered rules in such a way that for every model $(\mathfrak{s}, \mathfrak{h})$ of an atom $p_X(x_1, \ldots, x_{n+\nu})$ (with $n = ar(p)$), the set X denotes a set of indices $i \in [\![1, n]\!]$ such that $\mathfrak{s}(x_i) \in \mathrm{dom}(\mathfrak{h})$. In other words, X will denote a set of formal parameters of p_X that are allocated in every model of p_X.

Definition 37. *Given a formula α, we define the set $Alloc(\alpha)$ as follows: $x \in Alloc(\alpha)$ iff either α contains a points-to atom of the form $x \mapsto (\ldots)$, or a predicate atom $q_X(x'_1, \ldots, x'_{m+\nu})$ with $x'_i = x$ for some $i \in X$.*

Definition 38. *A rule $p_X(x_1, \ldots, x_{n+\nu}) \Leftarrow \pi$ in $D(\mathcal{R})$ with $n = ar(p)$ with $\rho = x_1 \mapsto (y_1, \ldots, y_k, \vec{w}, z_1, \ldots, z_\mu) * \rho'$ is* well-defined *if the following conditions hold:*

1. $\{x_1\} \subseteq Alloc(p_X(x_1, \ldots, x_{n+\nu})) \subseteq Alloc(\pi)$;
2. $\mathrm{fv}(\pi) \subseteq Alloc(\pi) \cup \{x_1, \ldots, x_{n+\nu}\}$.

We denote by $\mathcal{R}_l$ the set of well-defined rules in $D(\mathcal{R})$.

We first establish some important properties of $\mathcal{R}_l$.

Lemma 39. *If $i \in X$ then x_i is allocated in every predicate-less unfolding of $p_X(x_1, \ldots, x_{n+\nu})$.*

Proof. Let ϕ be a predicate-less unfolding of $p_X(x_1, \ldots, x_{n+\nu})$. The proof is by induction on the length of derivation from $p_X(x_1, \ldots, x_{n+\nu})$ to ϕ. Assume that $i \in X$. Then $\mathcal{R}_l$ contains a rule

$$p_X(x_1, \ldots, x_n, \vec{w}) \Leftarrow x_1 \mapsto (y_1, \ldots, y_\kappa, \vec{w}, z_1, \ldots, z_\mu)\sigma * \rho' * \mathop{\ast}_{i \in I} \underline{\perp}(z_i)$$

and $x_1 \mapsto (y_1, \ldots, y_\kappa, \vec{w}, z_1, \ldots, z_\mu)\sigma * \rho' * \mathop{\ast}_{i \in I} \underline{\perp}(z_i) \Leftarrow^*_{\mathcal{R}_l} \psi$. If $i = 1$ then it is clear that x_i is allocated in ψ. Otherwise by Condition 1 of Definition 38 we have

$$\begin{array}{rcl} x_i & \in & Alloc(p_X(x_1, \ldots, x_n, \vec{w})) \\ & \subseteq & Alloc(x_1 \mapsto (y_1, \ldots, y_\kappa, \vec{w}, z_1, \ldots, z_\mu)\sigma * \rho' * \mathop{\ast}_{i \in I} \underline{\perp}(z_i)) \\ & = & \{x_1\} \cup Alloc(\rho' * \mathop{\ast}_{i \in I} \underline{\perp}(z_i)), \end{array}$$

thus (since $\{z_1, \ldots, z_\mu\} \cap \{x_1, \ldots, x_n\} = \emptyset$) there exists an atom $q_Y(x'_1, \ldots, x'_m)$ occurring in ρ' and an index $j \in Y$ such that $x_i = x'_j$. Then ψ is of the form (modulo AC) $\psi' * \psi''$, with $q_Y(x'_1, \ldots, x'_m) \Leftarrow^*_{\mathcal{R}} \psi'$, and by the induction hypothesis, x'_j is allocated in ψ', hence x_i is allocated in ψ. □

Corollary 40. *Every rule in $\mathcal{R}_l$ is progressing, connected and established.*

Proof. Since $\mathcal{R}$ is progressing by hypothesis, it is straightforward to verify that $\mathcal{R}_l$ is also progressing. Consider a rule $p_X(x_1, \ldots, x_n, \vec{w}) \Leftarrow \pi$, with

$$\pi \stackrel{\text{def}}{=} x_1 \mapsto (y_1, \ldots, y_\kappa, \vec{w}, z_1, \ldots, z_\mu)\sigma * \rho' * *_{i \in I} \underline{\bot}(z_i),$$

that occurs in $\mathcal{R}_l$, and a predicate atom α occurring in $\rho' * *_{i \in I} \underline{\bot}(z_i)$. This rule is obtained from a rule $p(x_1, \ldots, x_n) \Leftarrow x_1 \mapsto (y_1, \ldots, y_\kappa) * \rho$ in $\mathcal{R}$. The atom α is either of the form $\underline{\bot}(z_i)$ for some $i \in I$ (so that $z_i \notin \mathrm{dom}(\sigma)$), or a decoration $q_Y(x'_1, \ldots, x'_m, \vec{w})$ of some atom $q(x'_1, \ldots, x'_m)$ occurring in ρ. By Assumption 31 $x'_1 \notin \{x_1, \ldots, x_n\}$, hence by definition of $z_1, \ldots z_\mu$ we have $x'_1 \in \{z_1, \ldots, z_\mu\}\sigma$ and the rule is connected. Let x be a variable occurring in $\mathrm{fv}(\pi) \setminus \{x_1, \ldots, x_{n+\nu}\}$ and assume $(\mathfrak{s}, \mathfrak{h}) \models_{\mathcal{R}_l} \pi$. Then by Condition 2 of Definition 38, $x \in \mathit{Alloc}(\pi)$. Thus, π contains either a points-to atom of the form $x \mapsto (\ldots)$, or a predicate atom $q_X(x'_1, \ldots, x'_{m+\nu})$ where $x'_i = x$ for some $i \in X$. In the first case, it is clear that x is allocated in any predicate-free unfolding of π. In the second case, by Lemma 39, x'_i is allocated in any predicate-free unfolding of $q_X(x'_1, \ldots, x'_{m+\nu})$, hence x is allocated in any predicate-free unfolding of π. This proves that the rule is established. □

We now relate the systems $\mathcal{R}$ and $\mathcal{R}_l$.

Definition 41. *A store $\mathfrak{s}$ is* quasi-injective *if, for all $x, y \in \mathrm{dom}(\mathfrak{s})$, the implication $\mathfrak{s}(x) = \mathfrak{s}(y) \Rightarrow x = y$ holds whenever $\{x, y\} \not\subseteq \{w_1, \ldots, w_\nu\}$.*

Lemma 42. *Let L be an infinite subset of $\mathcal{L}$. Consider a formula α containing no points-to atom, with $\mathcal{P}(\alpha) \subseteq \mathcal{P}_l$, and let $(\mathfrak{s}, \mathfrak{h})$ be an $\mathcal{R}$-model of α, where $\mathfrak{s}$ is quasi-injective, and $(\mathrm{rng}(\mathfrak{s}) \cup \mathrm{loc}(\mathfrak{h})) \cap L = \emptyset$. There exists a decoration α' of α, a heap $\mathfrak{h}'$ and a mapping $\gamma : \mathcal{L} \to \mathcal{L}$ such that:*

- $(\mathfrak{s}, \mathfrak{h}') \rhd_\gamma (\mathfrak{s}, \mathfrak{h})$,
- *if $\ell \notin L$ then $\gamma(\ell) = \ell$,*
- $\mathrm{loc}(\mathfrak{h}') \setminus \mathrm{rng}(\mathfrak{s}) \subseteq L$,
- $\mathrm{dom}(\mathrm{aux}(\mathfrak{h}')) \subseteq L$ *and*
- $(\mathfrak{s}, \mathfrak{h}') \models_{\mathcal{R}_l} \alpha'$.

Furthermore, if $\mathfrak{s}(u) \in \mathrm{dom}(\mathfrak{h}') \setminus \{\mathfrak{s}(w_i) \mid 1 \leq i \leq \nu\}$ then $u \in \mathit{Alloc}(\alpha')$.

Proof. Note that by hypothesis, L cannot contain $\bot$, since $\mathfrak{s}(\bot) = \bot$ and $\mathrm{rng}(\mathfrak{s}) \cap L = \emptyset$. The proof is by induction on the pair $(|\mathfrak{h}|, |\alpha|)$, using the lexicographic order. We distinguish several cases:

- If α is of the form $x \approx y$ or $x \not\approx y$, then $\mathfrak{h} = \emptyset$, and α is a decoration of itself, since it contains no predicate symbol. Since $(\mathfrak{s}, \mathfrak{h}) \rhd_{id} (\mathfrak{s}, \mathfrak{h})$, we may thus set $\alpha' \stackrel{\text{def}}{=} \alpha$, $\gamma \stackrel{\text{def}}{=} id$ and $\mathfrak{h}' \stackrel{\text{def}}{=} \mathfrak{h}$.

- If α' is of the form $\alpha'_1 \vee \alpha'_2$ then the proof follows immediately from the induction hypothesis.
- If α is of the form $\alpha_1 * \alpha_2$, then let L_1, L_2 be two disjoint infinite subsets of L. Since $(\mathfrak{s}, \mathfrak{h}) \models_{\mathcal{R}} \alpha_1 * \alpha_2$, there exist disjoint heaps $\mathfrak{h}_1$, $\mathfrak{h}_2$ such that $\mathfrak{h} = \mathfrak{h}_1 \uplus \mathfrak{h}_2$ and $(\mathfrak{s}, \mathfrak{h}_i) \models_{\mathcal{R}} \alpha_i$. By the induction hypothesis, for $i = 1, 2$, there exists a decoration α'_i of α_i, a heap $\mathfrak{h}'_i$ and a mapping $\gamma_i : \mathcal{L} \to \mathcal{L}$ such that: $(\mathfrak{s}, \mathfrak{h}'_i) \rhd_{\gamma_i} (\mathfrak{s}, \mathfrak{h}_i)$; $\ell \notin L_i \Rightarrow \gamma_i(\ell) = \ell$; $\mathrm{loc}(\mathfrak{h}'_i) \setminus \mathrm{rng}(\mathfrak{s}) \subseteq L_i$; $\mathrm{dom}(\mathrm{aux}(\mathfrak{h}')) \subseteq L_i$; $(\mathfrak{s}, \mathfrak{h}'_i) \models_{\mathcal{R}_l} \alpha'_i$ and if $\mathfrak{s}(u) \in \mathrm{dom}(\mathfrak{h}'_i) \setminus \{\mathfrak{s}(w_j) \mid 1 \leq j \leq \nu\}$ then $u \in \mathit{Alloc}(\alpha'_i)$. Let $\alpha' \stackrel{\text{def}}{=} \alpha'_1 * \alpha'_2$ and consider the function

$$\gamma : \ell \mapsto \begin{cases} \gamma_1(\ell) & \text{if } \ell \in L_1 \\ \gamma_2(\ell) & \text{if } \ell \in L_2 \\ \ell & \text{otherwise} \end{cases}$$

  Since $L_1 \cap L_2 = \emptyset$, this function is well-defined, and since $L_1 \cup L_2 \subseteq L$, if $\ell \notin L$ then $\gamma(\ell) = \ell$. Assume that $\mathrm{dom}(\mathfrak{h}'_1) \cap \mathrm{dom}(\mathfrak{h}'_2)$ contains an element ℓ. Then by the induction hypothesis, for $i = 1, 2$, $\ell \in \mathrm{rng}(\mathfrak{s}) \cup L_i$; and since $L_1 \cap L_2 = \emptyset$, we deduce that $\ell \in \mathrm{rng}(\mathfrak{s})$, so that $\ell \notin L$. Since $\mathrm{dom}(\mathrm{aux}(\mathfrak{h}'_i)) \subseteq L_i$, necessarily $\ell \in \mathrm{dom}(\mathrm{main}(\mathfrak{h}'_i))$ and $(\mathfrak{s}, \mathfrak{h}'_i) \rhd_{\gamma_i} (\mathfrak{s}, \mathfrak{h}_i)$, by Condition 2 of Definition 14, $\gamma_i(\ell) \in \mathrm{dom}(\mathfrak{h}_i)$. Since $\ell \notin L$ we have $\gamma_1(\ell) = \gamma_2(\ell) = \ell$, and we deduce that $\ell \in \mathrm{dom}(\mathfrak{h}_1) \cap \mathrm{dom}(\mathfrak{h}_2)$, which contradicts the fact that $\mathfrak{h}_1$ and $\mathfrak{h}_2$ are disjoint. Consequently, $\mathfrak{h}'_1$ and $\mathfrak{h}'_2$ are disjoint and we may define $\mathfrak{h}' \stackrel{\text{def}}{=} \mathfrak{h}'_1 \uplus \mathfrak{h}'_2$. Since $(\mathfrak{s}, \mathfrak{h}'_i) \models_{\mathcal{R}_l} \alpha'_i$, for both $i = 1, 2$, we have $(\mathfrak{s}, \mathfrak{h}') \models_{\mathcal{R}_l} \alpha'$.
  Let $\ell \in \mathrm{loc}(\mathfrak{h}'_i)$ for $i = 1, 2$. If $\ell \in \mathrm{rng}(\mathfrak{s})$, then by hypothesis $\ell \notin L$ and by construction, $\gamma(\ell) = \ell = \gamma_i(\ell)$. Otherwise, $\ell \in \mathrm{loc}(\mathfrak{h}'_i) \setminus \mathrm{rng}(\mathfrak{s})$, thus by the induction hypothesis $\ell \in L_i$ and by construction, $\gamma(\ell) = \gamma_i(\ell)$. We deduce by Lemma 17 that $(\mathfrak{s}, \mathfrak{h}') \rhd_{\gamma} (\mathfrak{s}, \mathfrak{h})$. Furthermore, still by the induction hypothesis we have:

$$\mathrm{loc}(\mathfrak{h}') \setminus \mathrm{rng}(\mathfrak{s}) \subseteq (\mathrm{loc}(\mathfrak{h}'_1) \setminus \mathrm{rng}(\mathfrak{s})) \cup (\mathrm{loc}(\mathfrak{h}'_2) \setminus \mathrm{rng}(\mathfrak{s})) \subseteq L_1 \cup L_2 \subseteq L.$$

  We also have $\mathrm{aux}(\mathfrak{h}') = \mathrm{aux}(\mathfrak{h}'_1) \uplus \mathrm{aux}(\mathfrak{h}'_2)$, thus $\mathrm{dom}(\mathrm{aux}(\mathfrak{h}')) = \mathrm{dom}(\mathrm{aux}(\mathfrak{h}'_1)) \cup \mathrm{dom}(\mathrm{aux}(\mathfrak{h}'_2)) \subseteq L_1 \cup L_2 \subseteq L$. Finally, if $\mathfrak{s}(x) \in \mathrm{dom}(\mathfrak{h}') \setminus \{\mathfrak{s}(w_i) \mid 1 \leq i \leq \nu\}$ then necessarily $\mathfrak{s}(x) \in \mathrm{dom}(\mathfrak{h}'_i)$ for some $i = 1, 2$, so that $x \in \mathit{Alloc}(\alpha'_i)$ and therefore $x \in \mathit{Alloc}(\alpha')$.
- Assume that α is of the form $p(u_1, \ldots, u_n)$ and that $\mathrm{dom}(\mathfrak{s}) = \mathrm{fv}(\alpha) \cup \{w_1, \ldots, w_\nu, \bot\}$. Then, $\mathcal{R}$ contains a rule $p(x_1, \ldots, x_n) \Leftarrow x_1 \mapsto (y_1, \ldots, y_\kappa) * \rho$ such that $(\mathfrak{s}_e, \mathfrak{h}) \models_{\mathcal{R}} u_1 \mapsto (y_1, \ldots, y_\kappa)\theta * \rho\theta$, where $\theta = \{\langle x_i, u_i \rangle \mid i \in [\![1, n]\!]\}$ and $\mathfrak{s}_e$ is an extension of $\mathfrak{s}$. Let $\{z_1, \ldots, z_\mu\} \stackrel{\text{def}}{=} (\mathrm{fv}(\rho) \cup \{y_1, \ldots, y_\kappa\}) \setminus \{x_1, \ldots, x_n\}$, be the set of existential variables of the above rule. We have $\mathrm{dom}(\mathfrak{s}_e) = \mathrm{dom}(\mathfrak{s}) \cup \{z_1, \ldots, z_\mu\}$. Consider the substitution σ such that: $\mathrm{dom}(\sigma) \subseteq \{z_1, \ldots, z_\mu\}$ and $\sigma(z_i) = z$ iff z is the first variable in $u_1, \ldots, u_n, w_1, \ldots, w_\nu, z_1, \ldots, z_{i-1}$ such that $\mathfrak{s}_e(z_i) = \mathfrak{s}_e(z)$ ($\sigma(z_i)$ is undefined in there is no such variable). By construction, if x is a variable occurring in $\rho\sigma$, then $x \notin \mathrm{dom}(\sigma)$. Let $\hat{\mathfrak{s}}$ be the restriction of $\mathfrak{s}_e$ to

$\mathrm{dom}(\mathfrak{s}_e) \setminus \mathrm{dom}(\sigma)$ and we show that $\hat{\mathfrak{s}}$ is quasi-injective. Assume that $\hat{\mathfrak{s}}(x) = \hat{\mathfrak{s}}(x')$ for distinct variables $x, x' \in \mathrm{dom}(\hat{\mathfrak{s}})$ with $\{x, x'\} \not\subseteq \{w_i \mid i \in [\![1, \nu]\!]\}$. Since $\hat{\mathfrak{s}}$ is a restriction of $\mathfrak{s}_e$, we have $\mathfrak{s}_e(x) = \mathfrak{s}_e(x')$. We deduce that x and x' both occur in the sequence $u_1, \ldots, u_n, w_1, \ldots, w_\nu, z_1, \ldots, z_\mu$, and we assume w.l.o.g. that x occurs before x' in this sequence. If $x, x' \in \{u_1, \ldots, u_n, w_1, \ldots, w_\nu\}$, then since $\mathfrak{s}_e$ is an extension of $\mathfrak{s}$, we would have $\mathfrak{s}(x) = \mathfrak{s}_e(x) = \mathfrak{s}_e(x') = \mathfrak{s}(x')$, so that $x = x'$, because by hypothesis $\mathfrak{s}$ is quasi-injective. Thus, one of the variables x, x' is in $\{z_1, \ldots, z_\mu\}$. Since x occurs before x' in the sequence $u_1, \ldots, u_n, w_1, \ldots, w_\nu, z_1, \ldots, z_\mu$, we deduce that $x' \in \{z_1, \ldots, z_\mu\}$. By definition of σ, this entails that $x'\sigma = x\sigma \neq x'$, hence $x' \in \mathrm{dom}(\sigma)$ and $x' \notin \mathrm{dom}(\hat{\mathfrak{s}})$, which contradicts our assumption.

Let $\mathfrak{h}_1$ be the restriction of $\mathfrak{h}$ to $\mathrm{dom}(\mathfrak{h}) \setminus \{\mathfrak{s}(u_1)\}$, so that $(\mathfrak{s}_e, \mathfrak{h}_1) \models_{\mathcal{R}} \rho\theta$. Then by construction, $(\hat{\mathfrak{s}}, \mathfrak{h}_1) \models_{\mathcal{R}} \rho\sigma\theta$. Let $L_1 \stackrel{\text{def}}{=} L \setminus \mathrm{rng}(\hat{\mathfrak{s}})$. Since $\mathrm{loc}(\mathfrak{h}_1) \subseteq \mathrm{loc}(\mathfrak{h})$ and $L_1 \subseteq L$, we have $(\mathrm{rng}(\hat{\mathfrak{s}}) \cup \mathrm{loc}(\mathfrak{h}_1)) \cap L_1 = \emptyset$. Thus, by the induction hypothesis, there exists a decoration ρ' of $\rho\sigma\theta$, a mapping $\gamma_1 : \mathcal{L} \to \mathcal{L}$ satisfying $\ell \notin L_1 \Rightarrow \gamma_1(\ell) = \ell$ and a heap $\mathfrak{h}'_1$ satisfying $\mathrm{loc}(\mathfrak{h}'_1) \setminus \mathrm{rng}(\hat{\mathfrak{s}}) \subseteq L_1$ and $\mathrm{dom}(\mathrm{aux}(\mathfrak{h}'_1)) \subseteq L_1$, such that $(\hat{\mathfrak{s}}, \mathfrak{h}'_1) \rhd_{\gamma_1} (\hat{\mathfrak{s}}, \mathfrak{h}_1)$, $(\hat{\mathfrak{s}}, \mathfrak{h}'_1) \models_{\mathcal{R}_l} \rho'$ and for all variables u, if $\hat{\mathfrak{s}}(u) \in \mathrm{dom}(\mathfrak{h}'_1) \setminus \{\hat{\mathfrak{s}}(w_i) \mid 1 \leq i \leq \nu\}$, then $u \in \mathit{Alloc}(\rho')$.

Let $E \stackrel{\text{def}}{=} (\mathrm{rng}(\hat{\mathfrak{s}}) \cup \mathrm{loc}(\mathfrak{h}'_1)) \setminus \mathrm{rng}(\mathfrak{s})$ and consider a bijection $\eta : \mathcal{L} \to \mathcal{L}$ such that:

- if $\ell \in E$ then $\eta(\ell) \in L \setminus E$ and $\eta(\eta(\ell)) = \ell$;
- if $\ell \in \mathcal{L} \setminus (E \cup \eta(E))$ then $\eta(\ell) = \ell$.

Such a bijection necessarily exists because E is finite and L is infinite. Let $\ell \in \mathrm{rng}(\mathfrak{s})$, so that $\ell \notin E$, and assume $\ell \in \eta(E)$. Then $\eta^{-1}(\ell) \in E$, hence $\ell \in L \setminus E$. But by the hypotheses of the lemma, $(\mathrm{rng}(\mathfrak{s}) \cup \mathrm{loc}(\mathfrak{h})) \cap L = \emptyset$, so this case is impossible. We deduce that $\ell \in \mathcal{L} \setminus (E \cup \eta(E))$ and that $\eta(\ell) = \ell$. Thus, in particular, $\eta(\bot) = \bot$. Consider the mapping $\gamma \stackrel{\text{def}}{=} \gamma_1 \circ \eta^{-1}$, the heap $\mathfrak{h}''_1 \stackrel{\text{def}}{=} \eta(\mathfrak{h}'_1)$ and the store $\hat{\mathfrak{s}}' \stackrel{\text{def}}{=} \eta(\hat{\mathfrak{s}})$. By Lemma 18 $(\hat{\mathfrak{s}}', \mathfrak{h}''_1) \rhd_{\gamma} (\hat{\mathfrak{s}}', \mathfrak{h}_1)$. By Lemma 13, since $(\hat{\mathfrak{s}}, \mathfrak{h}'_1) \models_{\mathcal{R}_l} \rho'$, we deduce that $(\hat{\mathfrak{s}}', \mathfrak{h}''_1) \models_{\mathcal{R}_l} \rho'$. We have $\mathrm{dom}(\mathfrak{s}) \subseteq \mathrm{dom}(\mathfrak{s}_e) \setminus \mathrm{dom}(\sigma) = \mathrm{dom}(\hat{\mathfrak{s}}) = \mathrm{dom}(\hat{\mathfrak{s}}')$, hence the restriction of $\hat{\mathfrak{s}}'$ to $\mathrm{dom}(\mathfrak{s})$ is well-defined, and if $x \in \mathrm{dom}(\mathfrak{s})$, then

$$\hat{\mathfrak{s}}'(x) = \eta(\hat{\mathfrak{s}}(x)) = \eta(\mathfrak{s}_e(x)) = \eta(\mathfrak{s}(x)) = \mathfrak{s}(x).$$

This shows that the restriction of $\hat{\mathfrak{s}}'$ to $\mathrm{dom}(\mathfrak{s})$ coincides with $\mathfrak{s}$.

Let $j \in [\![1, \mu]\!]$ such that $z_j \notin \mathrm{dom}(\sigma)$. By definition we have $\hat{\mathfrak{s}}(z_j) \notin \mathrm{rng}(\mathfrak{s})$, thus $\hat{\mathfrak{s}}(z_j) \in E$ and $\hat{\mathfrak{s}}'(z_j) = \eta(\hat{\mathfrak{s}}(z_j)) \in L$. Let I be the set of indices $j \in [\![1, \mu]\!]$ such that $z_j \notin \mathrm{dom}(\sigma)$ and $\hat{\mathfrak{s}}(z_j) \notin \mathrm{dom}(\mathfrak{h}'_1)$; for all $j \in I$, we therefore have $\hat{\mathfrak{s}}'(z_j) \in L$. Consider the set:

$$X \stackrel{\text{def}}{=} \{1\} \cup \{i \in [\![1, n]\!] \mid u_i \in \mathit{Alloc}(\rho') \wedge \forall j \in [\![1, \nu]\!],\, \mathfrak{s}(u_i) \neq \mathfrak{s}(w_j)\},$$

and let $\alpha' \stackrel{\text{def}}{=} p_X(u_1, \ldots, u_n, \vec{w})$. By definition, $D(\mathcal{R})$ contains a rule $(\mp)$

of the form

$$p_X(x_1,\ldots,x_n,\vec{w}) \Leftarrow x_1 \mapsto (y_1,\ldots,y_\kappa,\vec{w},z_1,\ldots,z_\mu)\sigma * \rho'' * *_{i\in I}\underline{\bot}(z_i),$$

where $\rho''\theta = \rho'$. We define the following heaps:

$$\begin{array}{lcl} \mathfrak{h}'_2 & \stackrel{\text{def}}{=} & \{\langle \hat{\mathfrak{s}}'(u_1), (\hat{\mathfrak{s}}'(y_1\sigma\theta),\ldots,\hat{\mathfrak{s}}'(y_\kappa\sigma\theta),\hat{\mathfrak{s}}'(\vec{w}),\hat{\mathfrak{s}}'(z_1\sigma\theta),\ldots,\hat{\mathfrak{s}}'(z_\mu\sigma\theta))\rangle\}, \\ \mathfrak{h}'_3 & \stackrel{\text{def}}{=} & \{\langle \hat{\mathfrak{s}}'(z_j), \vec{\bot}\rangle \mid j \in I\}. \end{array}$$

By definition of I, it cannot be the case that $\hat{\mathfrak{s}}'(u_1) = \hat{\mathfrak{s}}'(z_j)$ for $j \in I$, because otherwise we would have $\mathfrak{s}(u_1) = \hat{\mathfrak{s}}(z_j)$ and $z_j \in \mathrm{dom}(\sigma)$, hence $\mathrm{dom}(\mathfrak{h}'_2)\cap\mathrm{dom}(\mathfrak{h}'_3) = \emptyset$. We show that $(\mathrm{dom}(\mathfrak{h}'_2)\cup\mathrm{dom}(\mathfrak{h}'_3))\cap\mathrm{dom}(\mathfrak{h}''_1) = \emptyset$. First let $j \in I$, and assume $\hat{\mathfrak{s}}'(z_j) \in \mathrm{dom}(\mathfrak{h}''_1)$. Then since η is a bijection, necessarily, $\hat{\mathfrak{s}}(z_j) \in \mathrm{dom}(\mathfrak{h}'_1)$, which is impossible by definition of I. Now assume that $\hat{\mathfrak{s}}'(u_1) \in \mathrm{dom}(\mathfrak{h}''_1)$, so that $\hat{\mathfrak{s}}(u_1) \in \mathrm{dom}(\mathfrak{h}'_1)$. Then by definition of L_1 we have $\hat{\mathfrak{s}}(u_1) \notin L_1$, and $\gamma_1(\hat{\mathfrak{s}}(u_1)) = \hat{\mathfrak{s}}(u_1)$. Since $\mathrm{dom}(\mathrm{aux}(\mathfrak{h}'_1)) \subseteq L_1$, necessarily $\hat{\mathfrak{s}}(u_1) \in \mathrm{dom}(\mathrm{main}(\mathfrak{h}'_1))$, hence $\gamma_1(\hat{\mathfrak{s}}(u_1)) = \hat{\mathfrak{s}}(u_1) = \mathfrak{s}(u_1) \in \mathrm{dom}(\mathfrak{h}_1)$, which is impossible by definition of $\mathfrak{h}_1$. This shows that the domains of $\mathfrak{h}''_1$, $\mathfrak{h}'_2$ and $\mathfrak{h}'_3$ are pairwise disjoint, that $\mathfrak{h}' \stackrel{\text{def}}{=} \mathfrak{h}''_1 \uplus \mathfrak{h}'_2 \uplus \mathfrak{h}'_3$ is well-defined, and by construction,

$$(\hat{\mathfrak{s}}',\mathfrak{h}') \models_{\mathcal{R}_l} u_1 \mapsto (y_1,\ldots,y_\kappa,\vec{w},z_1,\ldots,z_\mu)\sigma\theta * \rho''\theta * *_{i\in I}\underline{\bot}(z_i)\theta.$$

We show that $(\mathfrak{s},\mathfrak{h}')\rhd_\gamma(\mathfrak{s},\mathfrak{h})$, with $\mathrm{main}(\mathfrak{h}') \stackrel{\text{def}}{=} \mathrm{main}(\mathfrak{h}''_1)\uplus\mathfrak{h}'_2$ and $\mathrm{aux}(\mathfrak{h}') \stackrel{\text{def}}{=} \mathrm{aux}(\mathfrak{h}''_1) \uplus \mathfrak{h}'_3$. Note that if $\ell \in \mathrm{dom}(\mathfrak{h}'_2)$ then necessarily $\ell = \hat{\mathfrak{s}}'(u_1) = \mathfrak{s}(u_1) \in \mathrm{rng}(\mathfrak{s})$, so that $\ell \notin L_1$ and by definition of η and γ_1, $\gamma(\ell) = \gamma_1 \circ \eta^{-1}(\ell) = \gamma_1(\ell) = \ell$. We check the four points of Definition 14 below:

1. Let ℓ_1,ℓ_2 be locations in $\mathrm{dom}(\mathrm{main}(\mathfrak{h}'))$ such that $\gamma(\ell_1) = \gamma(\ell_2)$, and assume that $\ell_1 \in \mathrm{dom}(\mathrm{main}(\mathfrak{h}''_1))$. Then $\gamma(\ell_1) \in \mathrm{dom}(\mathfrak{h}_1)$ because $(\hat{\mathfrak{s}}',\mathfrak{h}''_1)\rhd_\gamma(\hat{\mathfrak{s}}',\mathfrak{h}_1)$. If $\ell_2 \in \mathrm{dom}(\mathfrak{h}'_2)$ then $\gamma(\ell_2) = \ell_2$. By definition of $\mathfrak{h}_1$ we cannot have $\ell_2 \in \mathrm{dom}(\mathfrak{h}_1)$, and therefore, $\ell_1 \neq \ell_2$. Otherwise, $\ell_2 \in \mathrm{dom}(\mathrm{main}(\mathfrak{h}''_1))$ and for $i = 1,2$ we have $\gamma(\ell_i) = \gamma_1(\eta^{-1}(\ell_i))$ and $\eta^{-1}(\ell_i) \in \mathrm{dom}(\mathfrak{h}'_1)$. Since $(\hat{\mathfrak{s}},\mathfrak{h}'_1)\rhd_{\gamma_1}(\hat{\mathfrak{s}},\mathfrak{h}_1)$, we deduce that $\eta^{-1}(\ell_1) = \eta^{-1}(\ell_2)$ and because η is a bijection, $\ell_1 = \ell_2$. The proof is symmetric if $\ell_2 \in \mathrm{dom}(\mathrm{main}(\mathfrak{h}''_1))$. Finally, if $\ell_1,\ell_2 \in \mathrm{dom}(\mathfrak{h}'_2)$, then $\ell_1 = \ell_2$ since $\mathrm{dom}(\mathfrak{h}'_2)$ is a singleton.
2. We have $\gamma(\mathrm{dom}(\mathrm{main}(\mathfrak{h}''_1))) = \mathrm{dom}(\mathfrak{h}_1)$ and since $\gamma(\hat{\mathfrak{s}}'(u_1)) = \hat{\mathfrak{s}}'(u_1) = \mathfrak{s}(u_1)$, we deduce that $\gamma(\mathrm{dom}(\eta(\mathrm{main}(\mathfrak{h}''_1))) \cup \mathrm{dom}(\mathfrak{h}'_2)) = \mathrm{dom}(\mathfrak{h})$.
3. Let $\ell \in \mathrm{dom}(\mathrm{main}(\mathfrak{h}''_1))\cup\mathrm{dom}(\mathfrak{h}'_2)$. If $\ell = \hat{\mathfrak{s}}'(u_1)$, then by construction we have

$$\begin{array}{rcl} \mathfrak{h}'(\hat{\mathfrak{s}}'(u_1)) & = & (\hat{\mathfrak{s}}'(y_1\sigma\theta),\ldots,\hat{\mathfrak{s}}'(y_\kappa\sigma\theta),\hat{\mathfrak{s}}'(\vec{w}),\hat{\mathfrak{s}}'(z_1\sigma\theta),\ldots,\hat{\mathfrak{s}}'(z_\mu\sigma\theta)) \\ & = & (\hat{\mathfrak{s}}'(y_1\sigma\theta),\ldots,\hat{\mathfrak{s}}'(y_\kappa\sigma\theta),\mathfrak{s}(\vec{w}),\hat{\mathfrak{s}}'(z_1\sigma\theta),\ldots,\hat{\mathfrak{s}}'(z_\mu\sigma\theta)), \end{array}$$

   where the second line follows from the fact that $\hat{\mathfrak{s}}'$ coincides with $\mathfrak{s}$ on $\mathrm{dom}(\mathfrak{s})$. Note that, using the fact that $\mathrm{rng}(\hat{\mathfrak{s}}) \cap L_1 = \emptyset$ and by

definition of γ and $\hat{\mathfrak{s}}'$, the following equalities hold:

$$\begin{array}{rcl}\gamma\left(\hat{\mathfrak{s}}'(y_1\sigma\theta),\ldots,\hat{\mathfrak{s}}'(y_\kappa\sigma\theta)\right) & = & (\gamma_1(\hat{\mathfrak{s}}(y_1\sigma\theta)),\ldots,\gamma_1(\hat{\mathfrak{s}}(y_\kappa\sigma\theta))) \\ & = & (\hat{\mathfrak{s}}(y_1\sigma\theta),\ldots,\hat{\mathfrak{s}}(y_\kappa\sigma\theta)).\end{array}$$

We also have:

$$\begin{array}{rcl}\mathfrak{h}(\mathfrak{s}_e(u_1)) & = & (\mathfrak{s}_e(y_1\theta),\ldots,\mathfrak{s}_e(y_\kappa\theta)) \text{ and} \\ \mathfrak{h}(\hat{\mathfrak{s}}(u_1)) & = & (\hat{\mathfrak{s}}(y_1\sigma\theta),\ldots,\hat{\mathfrak{s}}(y_\kappa\sigma\theta)),\end{array}$$

where the second equation is a consequence of the definitions of σ and $\hat{\mathfrak{s}}$ respectively. This proves that:

$$\begin{array}{rcl}\mathfrak{h}(\gamma(\hat{\mathfrak{s}}'(u_1))) = \mathfrak{h}(\hat{\mathfrak{s}}(u_1)) & = & (\hat{\mathfrak{s}}(y_1\sigma\theta),\ldots,\hat{\mathfrak{s}}(y_\kappa)\sigma\theta) \\ & = & \gamma\left(\hat{\mathfrak{s}}'(y_1\sigma\theta),\ldots,\hat{\mathfrak{s}}'(y_\kappa\sigma\theta)\right),\end{array}$$

hence that $\hat{\mathfrak{s}}'(u_1)$ satisfies Condition 3 of Definition 14. Now if $\ell \in \mathrm{dom}(\mathrm{main}(\mathfrak{h}_1''))$, then it is straightforward to verify that ℓ satisfies Condition 3 of Definition 14, using the fact that $(\hat{\mathfrak{s}}', \mathfrak{h}_1'') \triangleright_\gamma (\hat{\mathfrak{s}}', \mathfrak{h}_1)$.

4. If $\ell \in \mathrm{dom}(\mathfrak{h}_3')$, then $\ell = \hat{\mathfrak{s}}'(z_j\theta)$ for some $j \in I$ and it is simple to verify that Condition 4 of Definition 14 is verified, setting $\mathrm{C}_{\mathfrak{h}'}(\ell) \stackrel{\mathsf{def}}{=} \hat{\mathfrak{s}}'(u_1)$. If $\ell \in \mathrm{dom}(\mathrm{aux}(\mathfrak{h}_1''))$ then, using the fact that $(\hat{\mathfrak{s}}', \mathfrak{h}_1'') \triangleright_\gamma (\hat{\mathfrak{s}}', \mathfrak{h}_1)$, we deduce that Condition 4 of Definition 14 is verified.

We prove that $\mathrm{dom}(\mathrm{aux}(\mathfrak{h}')) \subseteq L$. Let $\ell \in \mathrm{dom}(\mathrm{aux}(\mathfrak{h}'))$, and first assume that $\ell \in \mathrm{dom}(\mathrm{aux}(\mathfrak{h}_1''))$, so that $\ell = \eta(\ell')$ for $\ell' \in \mathrm{dom}(\mathrm{aux}(\mathfrak{h}_1'))$. By the induction hypothesis we have $\mathrm{dom}(\mathrm{aux}(\mathfrak{h}_1')) \subseteq L_1$, thus $\ell' \in L_1$. If $\ell' \in \mathrm{rng}(\mathfrak{s})$, then $\ell' \notin E$ (by definition of E), and $\ell' \notin L$ (by the hypothesis of the lemma), hence by definition of η we have $\eta(\ell') = \ell' = \ell \in L_1 \subseteq L$. Otherwise $\ell' \in E$ and by construction, $\eta(\ell') \in L \setminus E \subseteq L$. Now assume that $\ell \in \mathrm{dom}(\mathfrak{h}_3')$. Then $\ell = \hat{\mathfrak{s}}'(z_j)$ for some $j \in I$, and since we have shown that $\hat{\mathfrak{s}}'(z_j) \in L$, for every $j \in I$, we have $\ell \in L$.

We now show that $\mathrm{loc}(\mathfrak{h}') \setminus \mathrm{rng}(\mathfrak{s}) \subseteq L$. Since $\mathrm{dom}(\mathrm{aux}(\mathfrak{h}')) \subseteq L$ and $\mathrm{loc}(\mathrm{aux}(\mathfrak{h}')) = \mathrm{dom}(\mathrm{aux}(\mathfrak{h}')) \cup \{\bot\}$, we deduce that $\mathrm{loc}(\mathrm{aux}(\mathfrak{h}')) \setminus \mathrm{rng}(\mathfrak{s}) \subseteq L$, because $\bot \in \mathrm{rng}(\mathfrak{s})$. Now let $\ell \in \mathrm{loc}(\mathrm{main}(\mathfrak{h}')) \setminus \mathrm{rng}(\mathfrak{s})$. If $\ell \in \mathrm{loc}(\mathfrak{h}_2')$ then by definition of $\mathfrak{h}_2'$ and since $\ell \notin \mathrm{rng}(\mathfrak{s})$, we must have $\ell = \hat{\mathfrak{s}}'(z_j\sigma\theta) = \hat{\mathfrak{s}}'(z_j)$ for some $j \in I$, hence $\ell \in L$. Otherwise $\ell \in \mathrm{loc}(\mathrm{main}(\mathfrak{h}_1''))$, and $\ell = \eta(\ell')$ for some $\ell' \in \mathrm{loc}(\mathrm{main}(\mathfrak{h}_1'))$. If $\ell' \in \mathrm{rng}(\mathfrak{s})$ then $\eta(\ell') = \ell \in \mathrm{rng}(\mathfrak{s})$, which contradicts our assumption. Thus $\ell' \notin \mathrm{rng}(\mathfrak{s})$, hence $\ell' \in E$, and by definition of η, $\eta(\ell') = \ell \in L$.

There remains to prove that Rule $(\mp)$ is well-defined; this entails that it occurs in $\mathcal{R}_l$, hence that $(\mathfrak{s}, \mathfrak{h}') \models_{\mathcal{R}_l} \alpha'$. We first check that Condition 1 of Definition 38 holds. By construction we have $1 \in X$, hence $x_1 \in \mathit{Alloc}(p_X(x_1,\ldots,x_n))$, and we also have $x_1 \in \mathit{Alloc}(x_1 \mapsto (y_1,\ldots,y_\kappa,\vec{w},z_1,\ldots,z_\mu)\sigma)$. Now assume that $x_i \in \mathit{Alloc}(p_X(x_1,\ldots,x_n))$ and that $i \neq 1$. By definition of X, this entails that $x_i\theta \in \mathit{Alloc}(\rho')$. Since $\rho' = \rho''\theta$, we deduce that $x_i \in \mathit{Alloc}(\rho'')$. Next, we check that Condition 2 of Definition 38 holds. Let z be a variable occurring on the right-hand side of

rule ($\mp$) but not on its left-hand side. Then $z = z_j$, for some j with $z_j \notin \mathrm{dom}(\sigma)$ (indeed, $z_1, \ldots, z_\mu$ are the only existential variables, and if $z \in \mathrm{dom}(\sigma)$ then by definition of σ, we have $\sigma(z') \neq z$ for every variable z', thus z cannot occur in $\rho\sigma$, hence in ρ') and by definition of σ, we have $\hat{\mathfrak{s}}(z_j) = \mathfrak{s}_e(z_j) \notin \{\mathfrak{s}(w_1), \ldots, \mathfrak{s}(w_\nu)\}$. If $j \in I$ then $z_j \in \mathit{Alloc}(*_{i \in I} \underline{\perp}(z_i))$. Otherwise, by definition of I, we must have $\hat{\mathfrak{s}}(z_j) \in \mathrm{dom}(\mathfrak{h}'_1)$, and since $\mathrm{rng}(\hat{\mathfrak{s}}) \cap L_1 = \emptyset$, necessarily, $\hat{\mathfrak{s}}(z_j) = \gamma_1(\hat{\mathfrak{s}}(z_j))$. By the induction hypothesis, since $\hat{\mathfrak{s}}(z_j) \notin \{\mathfrak{s}(w_1), \ldots, \mathfrak{s}(w_\nu)\}$ and $\hat{\mathfrak{s}}(w_i) = \mathfrak{s}(w_i)$ for $i \in [\![1, \nu]\!]$, we deduce that $z_j \in \mathit{Alloc}(\rho')$; and since $z_j \notin \mathrm{dom}(\theta)$, we must have $z_j \in \mathit{Alloc}(\rho'')$.
We finally show that if $\mathfrak{s}(u) \in \mathrm{dom}(\mathfrak{h}') \setminus \{\mathfrak{s}(w_i) \mid 1 \leq i \leq \nu\}$ then $u \in \mathit{Alloc}(\alpha')$. Consider such a variable u. Assume $\mathfrak{s}(u) \in \mathrm{dom}(\mathfrak{h}'_3)$. Then $\mathfrak{s}(u)$ is of the form $\hat{\mathfrak{s}}'(z_j)$ for some $j \in I$, hence $\mathfrak{s}(u) \in L$, since we have shown that $\hat{\mathfrak{s}}'(z_j) \in L$, for every $j \in I$. But $L \cap \mathrm{rng}(\mathfrak{s}) = \emptyset$, so this case is impossible. We deduce that $\mathfrak{s}(u) \in \mathrm{dom}(\mathfrak{h}''_1) \cup \{\mathfrak{s}(u_1)\}$. If $\mathfrak{s}(u) = \mathfrak{s}(u_1)$, then since $\mathfrak{s}$ is quasi-injective we deduce that $u = u_1$ thus $u \in \mathit{Alloc}(\alpha')$, because by construction, $1 \in X$. Otherwise, we have $\mathfrak{s}(u) \in \mathrm{dom}(\mathfrak{h}''_1)$ and since $\eta(\ell) = \ell$ for all $\ell \in \mathrm{rng}(\mathfrak{s})$, necessarily $\mathfrak{s}(u) \in \mathrm{dom}(\mathfrak{h}'_1)$. By the induction hypothesis, we deduce that $u \in \mathit{Alloc}(\rho')$, hence there exists $x \in \mathit{Alloc}(\rho'')$ $(1 \leq j \leq n)$ such that $u = x\theta$. By definition of θ (and assuming by renaming that $\mathrm{fv}(\rho) \cap \mathrm{fv}(\alpha) = \emptyset$), necessarily, $x = x_j$, for some $j \in [\![1, n]\!]$. Then by definition of X we have $j \in X$, thus $u = x_j\theta \in \mathit{Alloc}(\alpha')$.

- Assume that α is of the form $p(u_1, \ldots, u_n)$ and that $\mathrm{dom}(\mathfrak{s}) \neq \mathrm{fv}(\alpha) \cup \{w_1, \ldots, w_\nu, \perp\}$. Note that we have necessarily $\mathrm{dom}(\mathfrak{s}) \supseteq \mathrm{fv}(\alpha) \cup \{w_1, \ldots, w_\nu, \perp\}$. Let $\mathfrak{s}'$ be the restriction of $\mathfrak{s}$ to $\mathrm{fv}(\alpha) \cup \{w_1, \ldots, w_\nu, \perp\}$. It is clear that $(\mathfrak{s}', \mathfrak{h}) \models \alpha$ and that $\mathfrak{s}'$ fulfills all the hypotheses of the lemma. Thus, by the previous item, there exists α', $\mathfrak{h}'$ and $\gamma : \mathcal{L} \to \mathcal{L}$ such that $(\mathfrak{s}', \mathfrak{h}') \triangleright_\gamma (\mathfrak{s}', \mathfrak{h})$, if $\ell \notin L$ then $\gamma(\ell) = \ell$, $\mathrm{loc}(\mathfrak{h}') \setminus \mathrm{rng}(\mathfrak{s}') \subseteq L$, $(\mathfrak{s}', \mathfrak{h}') \models_{\mathcal{R}_l} \alpha'$ and $\mathfrak{s}(u) \in \mathrm{dom}(\mathfrak{h}') \setminus \{\mathfrak{s}(w_i) \mid 1 \leq i \leq \nu\} \Rightarrow u \in \mathit{Alloc}(\alpha')$. It is clear that we have $(\mathfrak{s}, \mathfrak{h}') \triangleright_\gamma (\mathfrak{s}, \mathfrak{h})$, and $(\mathfrak{s}, \mathfrak{h}') \models_{\mathcal{R}_l} \alpha'$. Furthermore, since $\mathrm{rng}(\mathfrak{s}') \subseteq \mathrm{rng}(\mathfrak{s})$, we also have $\mathrm{loc}(\mathfrak{h}') \setminus \mathrm{rng}(\mathfrak{s}) \subseteq L$. Finally, if $\mathfrak{s}(u) \in \mathrm{dom}(\mathfrak{h}') \setminus \{\mathfrak{s}(w_i) \mid 1 \leq i \leq \nu\}$, then, since $\mathfrak{s}(u) \notin L$, and $\mathrm{ref}(\mathfrak{h}') \setminus \mathrm{rng}(\mathfrak{s}') \subseteq L$ we must have $\mathfrak{s}(u) \in \mathrm{rng}(\mathfrak{s}')$, thus $u \in \mathit{Alloc}(\alpha')$ by the previous item.

□

4.4 Transforming Entailments

We define $\widehat{\mathcal{R}} \stackrel{\text{def}}{=} \mathcal{R}_l \cup \mathcal{R}_r$. We show that the safe entailment problem $\phi \models_{\mathcal{R}} \psi$ can be solved by considering an entailment problem on $\widehat{\mathcal{R}}$ involving the elements of $D(\phi)$ (see Definition 32).

Corollary 43. *The entailment $\phi \models_{\mathcal{R}} \psi$ holds iff $\bigvee_{\phi' \in D(\phi)} \phi' \models_{\widehat{\mathcal{R}}} \widehat{\psi}$ holds.*

Proof. First assume that $\phi \models_{\mathcal{R}} \psi$. Consider a formula $\phi' \in D(\phi)$, let $(\mathfrak{s}, \mathfrak{h}')$ be an $\widehat{\mathcal{R}}$-model of ϕ' and let $\mathfrak{h} \stackrel{\text{def}}{=} \mathrm{trunc}(\mathfrak{h}')$. Note that by construction, $(\mathfrak{s}, \mathfrak{h}')$

is an $\mathcal{R}_l$-model of ϕ'. By definition of $D(\phi)$, ϕ' is a decoration of ϕ. Let $D_2 \stackrel{\text{def}}{=} \{\ell \in \mathrm{dom}(\mathfrak{h}') \mid \mathfrak{h}'(\ell) = \vec{\bot}\}$, $D_1 \stackrel{\text{def}}{=} \mathrm{dom}(\mathfrak{h}') \setminus D_2$, and consider a location $\ell \in \mathrm{dom}(\mathfrak{h}')$. By definition, ℓ must be allocated by some rule in $\mathcal{R}_l$. If ℓ is allocated by a rule of the form given in Definition 34, then necessarily $\mathfrak{h}'(\ell)$ is of the form $(\ell_1, \ldots, \ell_\kappa, \mathfrak{s}(w), \ell'_1, \ldots, \ell'_\mu)$ and $\ell \in D_1$. Otherwise, ℓ is allocated by the predicate $\underline{\bot}$ and we must have $\ell \in D_2$ by definition of the only rule for $\underline{\bot}$. Since this predicate must occur within a rule of the form given in Definition 34, ℓ necessarily occurs in the μ last components of the image of a location in D_1, hence admits a connection in $\mathfrak{h}'$. Consequently, by Lemma 24 $(\mathfrak{s}, \mathfrak{h}') \rhd_{id} (\mathfrak{s}, \mathfrak{h})$, and by Lemma 36, $(\mathfrak{s}, \mathfrak{h}) \models_{\mathcal{R}} \phi$. Thus $(\mathfrak{s}, \mathfrak{h}) \models_{\mathcal{R}} \psi$, and by Lemma 30, $(\mathfrak{s}, \mathfrak{h}') \models_{\mathcal{R}_r} \widehat{\psi}$, so that $(\mathfrak{s}, \mathfrak{h}') \models_{\widehat{\mathcal{R}}} \widehat{\psi}$.

Now assume that $\bigvee_{\phi' \in D(\phi)} \phi' \models_{\widehat{\mathcal{R}}} \widehat{\psi}$ and let $(\mathfrak{s}, \mathfrak{h})$ be a model of ϕ. Since the truth values of ϕ and ψ depend only on the variables in $\mathrm{fv}(\phi) \cup \mathrm{fv}(\psi)$, we may assume, w.l.o.g., that $\mathfrak{s}$ is quasi-injective. Consider an infinite set $L \subseteq \mathcal{L}$ such that $(\mathrm{rng}(\mathfrak{s}) \cup \mathrm{loc}(\mathfrak{h})) \cap L = \emptyset$. By Lemma 42, there exist a heap $\mathfrak{h}'$, a mapping $\gamma : \mathcal{L} \to \mathcal{L}$ and a decoration ϕ' of ϕ such that $\gamma(\ell) = \ell$ for all $\ell \notin L$, $(\mathfrak{s}, \mathfrak{h}') \rhd_\gamma (\mathfrak{s}, \mathfrak{h})$ and $(\mathfrak{s}, \mathfrak{h}') \models \phi'$. Since $\mathrm{rng}(\mathfrak{s}) \cap L = \emptyset$, we also have $\gamma(\mathfrak{s}) = \mathfrak{s}$. Then $(\mathfrak{s}, \mathfrak{h}') \models \widehat{\psi}$. Let $\mathfrak{h}_1 \stackrel{\text{def}}{=} \mathrm{trunc}(\mathfrak{h}')$. Since $(\mathfrak{s}, \mathfrak{h}') \rhd_\gamma (\mathfrak{s}, \mathfrak{h})$, by Corollary 23 we have $(\mathfrak{s}, \mathfrak{h}') \rhd_{id} (\mathfrak{s}, \mathfrak{h}_1)$, and by Lemma 30, $(\mathfrak{s}, \mathfrak{h}_1) \models \psi$. By Lemma 22 we have $\mathfrak{h} = \gamma(\mathfrak{h}_1)$; thus, since ψ is λ-restricted w.r.t. $\{w_1, \ldots, w_n\}$, we deduce by Lemma 12 that $(\mathfrak{s}, \mathfrak{h}) \models \psi$. □ □

This leads to the main result of this paper:

Theorem 44. *The safe entailment problem is 2EXPTIME-complete.*

Proof. The 2EXPTIME-hard lower bound follows from [8, Theorem 32], as the class of progressing, λ-connected and λ-restricted entailment problems is a subset of the safe entailment class. For the 2EXPTIME membership, Corollary 43 describes a many-one reduction to the progressing, connected and established class, shown to be in 2EXPTIME, by Theorem 5. Considering an instance $\mathfrak{P} = \phi \models_{\mathcal{R}} \psi$ of the safe class, Corollary 43 reduces this to checking the validity of $|D(\phi)|$ instances of the form $\phi' \models_{\widehat{\mathcal{R}}} \widehat{\psi}$, that are all progressing, connected and established, by Corollary 40. Since a formula $\phi' \in D(\phi)$ is obtained by replacing each predicate atom $p(x_1, \ldots, x_n)$ of ϕ by $p_X(x_1, \ldots, x_n, \vec{w})$ and there are at most 2^n such predicate atoms, it follows that $|D(\phi)| = 2^{\mathcal{O}(\mathrm{w}(\mathfrak{P}))}$. To obtain 2EXPTIME-membership of the problem, it is sufficient to show that each of the progressing, connected and established instances $\phi' \models_{\widehat{\mathcal{R}}} \widehat{\psi}$ can be built in time $|\mathfrak{P}| \cdot 2^{\mathcal{O}(\mathrm{w}(\mathfrak{P}) \cdot \log \mathrm{w}(\mathfrak{P}))}$. First, for each $\phi' \in D(\phi)$, by Definition 32, we have $|\phi'| \leq |\phi| \cdot (1 + \nu) \leq |\phi| \cdot (1 + \mathrm{w}(\mathfrak{P})) = |\phi| \cdot 2^{\mathcal{O}(\log \mathrm{w}(\mathfrak{P}))}$. By Definition 26, we have $|\widehat{\phi}| \leq |\phi| \cdot (1 + \nu) = |\phi| \cdot 2^{\mathcal{O}(\log \mathrm{w}(\mathfrak{P}))}$. By Definition 34, $D(\mathcal{R})$ can be obtained by enumeration in time that depends linearly of:

$$\begin{array}{rcl} |D(\mathcal{R})| & \leq & |\mathcal{R}| \cdot 2^{\mu} \cdot (n + \nu + \mu)^{\nu} \\ & \leq & |\mathcal{R}| \cdot 2^{\mathrm{w}(\mathfrak{P}) + \mathrm{w}(\mathfrak{P}) \cdot \log \mathrm{w}(\mathfrak{P})} \\ & = & |\mathfrak{P}| \cdot 2^{\mathcal{O}(\mathrm{w}(\mathfrak{P}))} \end{array}$$

This is because the number of intervals I is bounded by 2^{μ} and the number of substitutions σ by $(n+\nu+\mu)^{\nu}$, in Definition 34. By Definition 37, checking whether a rule is well-defined can be done in polynomial time in the size of the rule, hence in $2^{\mathcal{O}(\mathrm{w}(\mathfrak{P}))}$, so the construction of $\mathcal{R}_l$ takes time $|\mathfrak{P}|\cdot 2^{\mathcal{O}(\mathrm{w}(\mathfrak{P})\log \mathrm{w}(\mathfrak{P}))}$. Similarly, by Definition 34, the set $\widehat{\mathcal{R}}$ is constructed in time:

$$\begin{array}{rcl} |\widehat{\mathcal{R}}| & \leq & |\mathcal{R}| \cdot 2^{\mu} \cdot \mathrm{w}(\mathfrak{P})^{\nu} \\ & \leq & |\mathcal{R}| \cdot 2^{\mathrm{w}}(\mathfrak{P}) \cdot 2^{\mathrm{w}(\mathfrak{P})\cdot \log \mathrm{w}(\mathfrak{P})} \\ & = & |\mathfrak{P}| \cdot 2^{\mathcal{O}(\mathrm{w}(\mathfrak{P}))} \end{array}$$

Moreover, checking that a rule in $\widehat{\mathcal{R}}$ is connected can be done in time polynomial in the size of the rule, hence the construction of $\mathcal{R}_r$ takes time $2^{\mathcal{O}(\mathrm{w}(\mathfrak{P})\log \mathrm{w}(\mathfrak{P}))}$. Then the entire reduction takes time $2^{\mathcal{O}(\mathrm{w}(\mathfrak{P})\log \mathrm{w}(\mathfrak{P}))}$, which proves the 2EXPTIME upper bound for the safe class of entailments. □

References

[1] Timos Antonopoulos, Nikos Gorogiannis, Christoph Haase, Max I. Kanovich, and Joël Ouaknine. Foundations for decision problems in separation logic with general inductive predicates. In Anca Muscholl, editor, *FOSSACS 2014, ETAPS 2014, Proceedings*, volume 8412 of *Lecture Notes in Computer Science*, pages 411–425, 2014.

[2] Yehoshua Bar-Hillel, Micha Perles, and Eli Shamir. On formal properties of simple phrase structure grammars. *Sprachtypologie und Universalienforschung*, 14:143–172, 1961.

[3] Josh Berdine, Byron Cook, and Samin Ishtiaq. Slayer: Memory safety for systems-level code. In Ganesh Gopalakrishnan andShaz Qadeer, editor, *Computer Aided Verification - 23rd International Conference, CAV 2011, Snowbird, UT, USA, July 14-20, 2011. Proceedings*, volume 6806 of *LNCS*, pages 178–183. Springer, 2011.

[4] Cristiano Calcagno, Dino Distefano, Jérémy Dubreil, Dominik Gabi, Pieter Hooimeijer, Martino Luca, Peter W. O'Hearn, Irene Papakonstantinou, Jim Purbrick, and Dulma Rodriguez. Moving fast with software verification. In Klaus Havelund, Gerard J. Holzmann, and Rajeev Joshi, editors, *NASA Formal Methods - 7th International Symposium, NFM 2015, Pasadena, CA, USA, April 27-29, 2015, Proceedings*, volume 9058 of *LNCS*, pages 3–11. Springer, 2015.

[5] Kamil Dudka, Petr Peringer, and Tomás Vojnar. Predator: A practical tool for checking manipulation of dynamic data structures using separation logic. In Ganesh Gopalakrishnan and Shaz Qadeer, editors, *Computer Aided Verification - 23rd International Conference, CAV 2011, Snowbird, UT, USA, July 14-20, 2011. Proceedings*, volume 6806 of *LNCS*, pages 372–378. Springer, 2011.

[6] Mnacho Echenim, Radu Iosif, and Nicolas Peltier. Entailment checking in separation logic with inductive definitions is 2-exptime hard. In *LPAR 2020: 23rd International Conference on Logic for Programming, Artificial Intelligence and Reasoning, Alicante, Spain, May 22-27, 2020*, volume 73 of *EPiC Series in Computing*, pages 191–211. EasyChair, 2020.

[7] Mnacho Echenim, Radu Iosif, and Nicolas Peltier. Entailment is Undecidable for Symbolic Heap Separation Logic Formulae with Non-Established Inductive Rules. working paper or preprint, September 2020. URL: https://hal.archives-ouvertes.fr/hal-02951630.

[8] Mnacho Echenim, Radu Iosif, and Nicolas Peltier. Decidable entailments in separation logic with inductive definitions: Beyond establishment. In *CSL 2021: 29th International Conference on Computer Science Logic*, EPiC Series in Computing. EasyChair, 2021.

[9] Radu Iosif, Adam Rogalewicz, and Jiri Simacek. The tree width of separation logic with recursive definitions. In *Proc. of CADE-24*, volume 7898 of *LNCS*, 2013.

[10] Radu Iosif, Adam Rogalewicz, and Tomás Vojnar. Deciding entailments in inductive separation logic with tree automata. In Franck Cassez and Jean-François Raskin, editors, *ATVA 2014, Proceedings*, volume 8837 of *Lecture Notes in Computer Science*, pages 201–218. Springer, 2014.

[11] Samin S Ishtiaq and Peter W O'Hearn. Bi as an assertion language for mutable data structures. In *ACM SIGPLAN Notices*, volume 36, pages 14–26, 2001.

[12] Jens Katelaan, Christoph Matheja, and Florian Zuleger. Effective entailment checking for separation logic with inductive definitions. In Tomás Vojnar and Lijun Zhang, editors, *TACAS 2019, Proceedings, Part II*, volume 11428 of *Lecture Notes in Computer Science*, pages 319–336. Springer, 2019.

[13] Jens Pagel and Florian Zuleger. Beyond symbolic heaps: Deciding separation logic with inductive definitions. In *LPAR-23*, volume 73 of *EPiC Series in Computing*, pages 390–408. EasyChair, 2020.

[14] J.C. Reynolds. Separation Logic: A Logic for Shared Mutable Data Structures. In *Proc. of LICS'02*, 2002.