Local Reasoning about Parameterized Reconfigurable Distributed Systems

Emma Ahrens, Marius Bozga, Radu Iosif, Joost-Pieter Katoen

Local Reasoning about Parameterized Reconfigurable Distributed Systems

Emma Ahrens[*] Marius Bozga[†] Radu Iosif[‡]
Joost-Pieter Katoen[§]

November 8, 2021

Abstract

This paper presents a Hoare-style calculus for formal reasoning about reconfiguration programs of distributed systems. Such programs delete or create interactions or components while the system components change state according to their local behaviour. Our proof calculus uses a configuration logic that supports local reasoning and that relies on inductive predicates to describe distributed systems with an unbounded number of components. The validity of reconfiguration programs relies on havoc invariants, assertions about the ongoing interactions in the system. We present a proof system for such invariants in an assume/rely-guarantee style. We illustrate the feasibility of our approach by proving the correctness of self-adjustable tree architectures and provide tight complexity bounds for entailment checking in the configuration logic.

1 Introduction

The relevance of dynamic reconfiguration. Dynamic reconfigurable distributed systems are en vogue. For instance, distributed architectures of modern data centers have the possibility to change their communication topology at runtime. This enables demand-aware data center networks which (self-)adjust to their workload. We refer the interested reader to the recent survey [**?**] for more details. This development provides e.g., new impulses to distributed algorithm design [**?**] and has given rise to self-adjustable network architectures whose topology reconfigurations are akin to amendments of dynamic data structures such as splay trees [**?**]. This paper focuses on a formal framework to reason about elementary properties of such systems.

[*]Emma.Ahrens@rwth-aachen.de, RWTH Aachen University, D-52056, Germany
[†]Marius.Bozga@univ-grenoble-alpes.fr, Univ. Grenoble Alpes, CNRS, Grenoble INP, VERIMAG, 38000, France
[‡]Radu.Iosif@univ-grenoble-alpes.fr, Univ. Grenoble Alpes, CNRS, Grenoble INP, VERIMAG, 38000, France
[§]katoen@cs.rwth-aachen.de, RWTH Aachen University, D-52056, Germany

Distributed systems. The distributed systems with dynamic reconfigurations considered in this paper consist of an unbounded number of components and have a flexible, i.e., not a priori fixed, topology. Communication is assumed to be correct, i.e., packet losses and corruptions are abstracted from. We also abstract from low-level coordination mechanisms between processes such as semaphores, compare-and-swap operations and the like. Components are finite-state abstractions of sequential programs, whose transitions are labelled with events. They communicate via interactions — a form of handshaking — modelled as sets of events that occur simultaneously in multiple components. We thus distinguish between *behaviour* (encapsulated by components) and *coordination* (described by interactions). System architectures are hypergraphs of components and interactions defining the coordination within the distributed system.

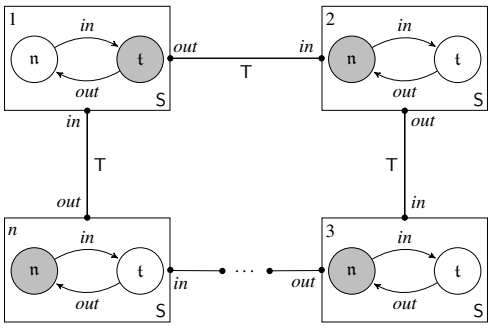
An illustrative example. We illustrate the setting by a token ring example, consisting of a finite but unbounded number of components, indexed from 1 to n, connected via an unidirectional ring (Fig. 1). A token may be passed from a component i in state $\mathfrak{t}$ (it has a token) to its neighbour, with index $(i \bmod n)+1$, which must be in state $\mathfrak{n}$ (it has no token). As result of this interaction, the i-th component moves to state $\mathfrak{n}$ while the $(i \bmod n)+1$ component moves to state $\mathfrak{t}$. Note that token passing interactions are possible as long as at least two components are in different states; if all the components are in the same state at the same time, the ring is in a *deadlock* configuration.

During operation, components can be added to, or removed from the ring. On removing the component with index i, its incoming (from $i-1$, if $i>1$, or from n, if $i=1$) and outgoing (to $(i \bmod n)+1$) connectors are deleted before the component is deleted, and its left and right neighbours are reconnected in order to re-establish the ring-shaped topology. Consider the program in Listing 1, where the variables x, y and z are assigned the indices 1, 2 and 3, respectively. The program removes first the right connector between y and z (line 3), then removes the left connector between x and y (line 4), before removing the component indexed by y (line 5) and reconnecting the x and y components (line 6). Note that the order of the commands is crucial: assume that component x is the only one in state $\mathfrak{t}$ in the entire system. Then the token may move from x to y and is deleted together with the component (line 5). In this case, the resulting ring has no token and the system is in a deadlock configuration. The reconfiguration program in Listing 2 is obtained by swapping lines 3 and 4 from Listing 1. In this case, the deleted component is in state $\mathfrak{n}$ before the reconfiguration and its left connector is removed before its right one, thus ensuring that the token does not move to the y component (deleted at line 5).

The framework developed in this paper allows to prove that, when applied to a token ring of size $n \geq 2$, with at least two components in state $\mathfrak{n}$ and at least one component in state $\mathfrak{t}$, the program in Listing 2 yields a system with at least two components in different states, *for any* $n \geq 2$. Using, e.g. invariant synthesis methods similar to those in [**?**], token ring systems can be automatically proved to be deadlock-free, under this assumption.

The contributions of this paper. Whereas various formalisms for modelling distributed systems support dynamic reconfiguration, see e.g., [**?**, **?**], the formal verifi-

Figure 1: Reconfiguration of a Parametric Token Ring System



Listing 1: Delete Component (wrong version)

```
1 with x,y,z : T(x,y) * S(y) *
2        T(y,z) ∧ state(y,S,n) do
3   disconnect(T,y,z);
4   disconnect(T,x,y);
5   delete(S,y);
6   connect(T,x,z); od
```

Listing 2: Delete Component (correct version)

```
1 with x,y,z : T(x,y) * S(y) *
2        T(y,z) ∧ state(y,S,n) do
3   disconnect(T,x,y);
4   disconnect(T,y,z);
5   delete(S,y);
6   connect(T,x,z); od
```

cation of system properties under reconfigurations has received scant attention. We provide a Hoare-style framework to formally reason about properties of reconfigurable distributed systems. Our approach annotates reconfiguration programs (i.e. programs that delete and create interactions or components) with assertions written in a *configuration logic*, that describes both the topology of the system (i.e. the components and connectors that form its coordinating architecture) and the local states of the components. The annotations of the reconfiguration program are proved to be valid under so-called *havoc invariants*, expressing global properties about the states of the components, that remain, moreover, unchanged under the ongoing interactions in the system. In order to prove these havoc invariants for networks of any size, we develop an induction-based proof system, that uses a parallel composition rule in the style of assume/rely-guarantee reasoning. In contrast with existing formal verification techniques, we do not consider the network topology to be fixed in advance, and allow it to change dynamically, as described by the reconfiguration program. This paper provides the details of our proof systems and the semantics of reconfiguration programs. We illustrate the usability of our approach by proving the correctness of self-adjustable tree architectures and provide tight complexity bounds for the problem of entailment between formulæ in the configuration logic, relevant for the automation of our approach.

Main challenges. Formal reasoning about reconfigurable distributed systems faces two technical challenges. The first issue is the huge complexity of nowadays distributed systems, that requires highly scalable proof techniques, which can only be achieved by *local reasoning*, a key ingredient of other successful proof techniques, based on Separation Logic [**?**]. To this end, atomic reconfiguration commands in our proof system are specified by axioms that only refer to the components directly involved in the action, while framing out the rest of the distributed system. This principle sounds appealing, but is technically challenging, as components from the local specification interfere with components from the frame [1]. To tackle this issue, we provide a compositional proof rule in the spirit of *rely/assume-guarantee* reasoning [**?**, **?**], whose assumptions are automatically synthesized from the formulæ describing the split of the system.

The second issue is dealing with the non-trivial interplay between reconfigurations and interactions. Reconfigurations change the system by adding/removing components/interactions *while the system is running*, i.e. while state changes occur within components by firing interactions. Although changes to the structure of the distributed system at first sight seem orthogonal to the component's state changes, the impact of a reconfiguration can be immense. For instance, deleting a component holding the token in a token-ring network yields a deadlocked system, while adding a component with a token could lead to a race scenario, where two components access a shared resource simultaneously. Technically, this means that a frame rule cannot be directly applied to sequentially composed reconfigurations, as e.g. an unbounded number of interactions may fire between two atomic reconfiguration actions.

2 Models of Distributed Systems

Given integers i and j, we write $[i,j]$ for the set $\{i, i+1, \ldots, j\}$, assumed to be empty if $i > j$. For a tuple $\mathbf{a} = \langle a_1, \ldots, a_k \rangle$, we write $\langle \mathbf{a} \rangle_i \stackrel{\text{def}}{=} a_i$ for its i-th element and $\langle \mathbf{a} \rangle_{[i,j]} \stackrel{\text{def}}{=} \langle a_i, \ldots, a_j \rangle$ for the subsequence of $\mathbf{a}$ from i to j included. We sometimes abuse notation and write $a \in \mathbf{a}$ instead of $a = \langle \mathbf{a} \rangle_i$, for some $i \in [1,k]$. Function applications are lifted to sets $f(A) \stackrel{\text{def}}{=} \{f(a) \mid a \in A\}$ and tuples $f(\langle a_1, \ldots, a_k \rangle) \stackrel{\text{def}}{=} \langle f(a_1), \ldots, f(a_k) \rangle$. When no confusion arises, we write $f(a_1, \ldots, a_k)$ instead of $f(\langle a_1, \ldots, a_k \rangle)$. By $\mathcal{P}(A)$ we denote the powerset of a set A. We denote the domain of a function f by $\mathrm{dom}(f)$. By $f[a \leftarrow b]$ we denote the function that maps a into b and behaves like f for all elements of $\mathrm{dom}(f) \setminus \{a\}$. The cardinality of a finite set A is denoted as $\|A\|$.

A *signature* is a tuple $\mathfrak{S} = \langle C_1, \ldots, C_n, I_1, \ldots, I_m \rangle$ of relation symbols. The relation symbols $\mathfrak{C} \stackrel{\text{def}}{=} \{C_1, \ldots, C_n\}$ of arity $\#(C_i) = 1$ are called *component types* and the symbols $\mathfrak{I} \stackrel{\text{def}}{=} \{I_1, \ldots, I_m\}$ of arity $\#(I_j) \geq 2$ are called *interaction types*. Let $\mathfrak{S}$, $\mathfrak{C}$ and $\mathfrak{I}$ be fixed in the rest of the paper.

Let $\mathbb{Q}$ and $\mathbb{P}$ be finite sets of *states* and *ports*, respectively. Each *component type* C_i is associated with a finite-state machine $\mathcal{B}(C_i) \stackrel{\text{def}}{=} (Q_i, P_i, q_i^0, \rightarrow_i)$, called the *behavior* of C_i, where $Q_i \subseteq \mathbb{Q}$ is a set of states, $P_i \subseteq \mathbb{P}$ is a set of ports, $q_i^0 \in Q_i$ is the initial state, $\rightarrow_i \subseteq Q_i \times P_i \times Q_i$ is a transition relation, the elements of which are called transitions,

[1] Essentially the equivalent of the environment in a parallel compositional proof system.

denoted as $q \xrightarrow{p}_i q'$. The states q and q' are the *pre-* and *post-state* and the port p is the *label* of the transition. For each component type C_i, we denote $ports(C_i) \stackrel{\text{def}}{=} P_i$ and require that $ports(C_i) \cap ports(C_j) = \emptyset$, for all $i, j \in [1,n]$. Each interaction type I is associated a distinct tuple $ports(I) = \langle p_1, \ldots, p_{\#(I)} \rangle \in \mathbb{P}^{\#(I)}$. Intuitively, an interaction of type I fires transitions labeled by $p_1, \ldots, p_{\#(I)}$, from several behaviors, all at once.

Example 1 *We model token rings using the signature $\mathfrak{S} \stackrel{\text{def}}{=} \langle \mathsf{S}, \mathsf{T} \rangle$, where the component type S (station) has arity 1 and the interaction type T (transfer) has arity 2. The set of states is $\mathbb{Q} \stackrel{\text{def}}{=} \{\mathfrak{t}, \mathfrak{n}\}$, where $\mathfrak{t}$ ($\mathfrak{n}$) means that the component has (doesn't have) the token, and the set of ports is $\mathbb{P} \stackrel{\text{def}}{=} \{in, out\}$. The behavior $\mathcal{B}(\mathsf{S}) \stackrel{\text{def}}{=} (\mathbb{Q}, \mathbb{P}, q^0, \rightarrow)$ has initial state $\mathfrak{n}$ and transitions $\mathfrak{t} \xrightarrow{out} \mathfrak{n}$, $\mathfrak{n} \xrightarrow{in} \mathfrak{t}$ (Fig. 1). The interaction type T is associated to the tuple $ports(\mathsf{T}) = \langle out, in \rangle \in \mathbb{P}^2$.* ■

Definition 1 (Configurations) *Let $\mathbb{V}$ and $\mathbb{U}$ be countably infinite sets of* variables *and* indices*, respectively. A* configuration *is a tuple (σ, ν, ρ), where:*

- $\sigma \stackrel{\text{def}}{=} \langle C_1^\sigma, \ldots, C_n^\sigma, I_1^\sigma, \ldots, I_m^\sigma \rangle$ *is a* structure *that interprets the relation symbols in the signature $\mathfrak{S} = \langle C_1, \ldots, C_n, I_1, \ldots, I_m \rangle$; each component type C_i is interpreted by a set $C_i^\sigma \subseteq \mathbb{U}$ and each interaction type I_j is interpreted by a relation $I_j^\sigma \subseteq \mathbb{U}^{\#(I_j)}$. A tuple $\mathbf{u} \in I_j^\sigma$ is called an* interaction of type I_j*, sometimes denoted as the pair $[I_j, \mathbf{u}]$. We denote by* $\text{nodes}(\sigma) \stackrel{\text{def}}{=} \bigcup_{i=1}^n C_i^\sigma \cup \bigcup_{j=1}^m \{\langle \mathbf{u} \rangle_k \mid \mathbf{u} \in I_j^\sigma,\ k \in [1, \#(I_j)]\}$ *the set of indices that occur in σ.*
- $\nu : \mathbb{V} \rightarrow \mathbb{U}$ *is a* store*, i.e. a total function associating variables to indices, and*
- $\rho : \mathbb{U} \times \mathfrak{C} \rightarrow \mathbb{Q}$ *is a* state map*, i.e. a total function associating an index u and a component type C_i a state in Q_i, where $\mathcal{B}(C_i) = (Q_i, P_i, q_i^0, \rightarrow_i)$ is the behavior of C_i.*

We denote by Γ the set of configurations.

Example 2 *Consider the token ring system in Fig. 1. A configuration (σ, ν, ρ) of this system has structure $\sigma \stackrel{\text{def}}{=} \langle \mathsf{S}^\sigma = \{1, 2, 3, \ldots, n\}, \mathsf{T}^\sigma = \{\langle 1,2 \rangle, \langle 2,3 \rangle, \ldots, \langle n, 1 \rangle\} \rangle$, where the set S^σ contains the indices of n components and the interactions T^σ relate each component of index i to its successor, of index $(i \bmod n) + 1$. The state map is $\rho(1, \mathsf{S}) = \mathfrak{t}$ and $\rho(i, \mathsf{S}) = \mathfrak{n}$ for $i \in \{2, \ldots, n\}$.* ■

An *action* is a function $f : \Gamma \rightarrow \mathcal{P}(\Gamma)^\top$, where the complete lattice $(\mathcal{P}(\Gamma), \subseteq, \cup, \cap)$ is extended with a greatest element $\top$, with the conventions $S \cup \top \stackrel{\text{def}}{=} \top$ and $S \cap \top \stackrel{\text{def}}{=} S$, for each $S \in \mathcal{P}(\Gamma)$ and let $\mathcal{P}(\Gamma)^\top \stackrel{\text{def}}{=} \mathcal{P}(\Gamma) \cup \{\top\}$. We say that an action f is *disabled* in (σ, ν, ρ) iff $f(\sigma, \nu, \rho) = \emptyset$ and that it *faults* in (σ, ν, ρ) iff $f(\sigma, \nu, \rho) = \top$. The composition of actions f and g is $(f \circ g)(\sigma, \nu, \rho) \stackrel{\text{def}}{=} \bigcup_{(\sigma', \nu', \rho') \in g(\sigma, \nu, \rho)} f(\sigma', \nu', \rho')$ and the *iteration* of an action f is $f^* \stackrel{\text{def}}{=} \bigcup_{i=0}^\infty f^i$, where $f^0(\sigma, \nu, \rho) \stackrel{\text{def}}{=} \{(\sigma, \nu, \rho)\}$ and $f^{i+1} \stackrel{\text{def}}{=} f^i \circ f$, for all $(\sigma, \nu, \rho) \in \Gamma$.

A *state change* is an action f such that, for all $(\sigma',\nu',\rho') \in f(\sigma,\nu,\rho)$, we have $\sigma' = \sigma$, $\nu' = \nu$ and $\rho'(u,C) = \rho(u,C)$, for all u,C such that $u \in \mathbb{U} \setminus C^\sigma$; these actions only change the states of the components from the structure, but not the structure, the store, or the states of the components outside of the structure. Given state changes f and g, the composition $f \circ g$ and iteration f^* are also state changes. In the following, we shall also consider actions that change the structure, the store and the state map for indices outside of the structure (§4).

Definition 2 (Havoc) *Given an interaction type I and a tuple $\mathbf{u} = \langle u_1, \ldots, u_{\#(I)} \rangle$, such that $ports(I) = \langle p_1, \ldots, p_{\#(I)} \rangle$ and $p_j \in P_{i_j}$, where C_{i_j} is a component type with behavior $\mathcal{B}(C_{i_j}) \stackrel{\text{def}}{=} (Q_{i_j}, P_{i_j}, q^0_{i_j}, \rightarrow_{i_j})$, for all $j \in [1,\#(I)]$, the* atomic state change $\mathfrak{c}[I,\mathbf{u}]$ *maps a configuration (σ,ν,ρ) onto the set of configurations (σ,ν,ρ'), such that:*

1. *for all $j \in [1,\#(I)]$, we have $u_j \in C^\sigma_{i_j}$ and $\rho(u_j,C_{i_j}) \xrightarrow{p_j}_{i_j} \rho'(u_j,C_{i_j})$, and*
2. *$\rho(u,C) = \rho'(u,C)$, for all $(u,C) \in (\mathbb{U} \times \mathfrak{C}) \setminus \{(u_j,C_{i_j}) \mid j \in [1,\#(I)]\}$,*

if $\mathbf{u} \in I^\sigma$ and the empty set, otherwise. The havoc *action is the iteration $\mathfrak{h} \stackrel{\text{def}}{=} \mathfrak{c}^*$ of the action $\mathfrak{c} \stackrel{\text{def}}{=} \bigcup \left\{ \mathfrak{c}[I,\mathbf{u}] \mid I \in \mathfrak{I}, \mathbf{u} \in \mathbb{U}^{\#(I)} \right\}$.*

Intuitively, an atomic state change $\mathfrak{c}[I,\mathbf{u}]$ corresponds to firing an interaction $[I,\mathbf{u}]$, which happens only if all indices from the interaction denote components from the structure and, moreover, each such component is in a state from which a transition (labeled by the port specified by the type of the interaction) is enabled (1). The states of the components not involved in the interaction are not changed (2). Note that the action is disabled if either the tuple of indices is not an interaction ($\mathbf{u} \notin I^\sigma$), or at least one element of the tuple $\mathbf{u}$ is not the index of a component from the structure.

Example 3 *Consider the atomic state changes $\mathfrak{c}[\mathsf{T},\langle i,(i \bmod n)+1\rangle]$, for all $i \in [1,n]$, applied to the configuration in Example 2. Since $\mathsf{T}^\sigma = \{\langle 1,2\rangle, \langle 2,3\rangle, \ldots, \langle n,1\rangle\}$, the action $\mathfrak{c}[\mathsf{T},\langle i,j\rangle]$ is disabled, for all $j \neq (i \bmod n)+1$. The interaction type T has $ports(\mathsf{T}) = \langle out, in\rangle$ and the transitions of $\mathcal{B}(\mathsf{S})$ are $\mathfrak{t} \xrightarrow{out} \mathfrak{n}$ and $\mathfrak{n} \xrightarrow{in} \mathfrak{t}$. The first transition is only possible if the component is in state $\mathfrak{t}$, which is the case only for the component with index 1, because $\rho(1,\mathsf{S}) = \mathfrak{t}$ and $\rho(i,\mathsf{S}) = \mathfrak{n}$ for $i \in [2,n]$. In this configuration, only the atomic state change $\mathfrak{c}[\mathsf{T},\langle 1,2\rangle]$ is enabled, the outcome of which is the configuration (σ,ν,ρ'), where $\rho'(1,\mathsf{S}) \stackrel{\text{def}}{=} \mathfrak{n}$, $\rho'(2,\mathsf{S}) \stackrel{\text{def}}{=} \mathfrak{t}$ and $\rho'(j,\mathsf{S}) \stackrel{\text{def}}{=} \rho(j,\mathsf{S})$, for any $j \in \mathbb{U} \setminus \{1,2\}$. In this new configuration, only $\mathfrak{c}[\mathsf{T},\langle 2,3\rangle]$ is enabled. Continuing this way, the token is moved around the ring and the system gets back to the starting configuration (Example 2) after n steps. Consequently, the havoc action $\mathfrak{h}$ applied to (σ,ν,ρ) yields the set of configurations in which exactly one component is in state $\mathfrak{t}$ and the other are in state $\mathfrak{n}$.* ■

3 Configuration Logic

We define a *configuration logic* (CL) that is, an assertion language describing sets of configurations. Let $\mathbb{A}$ be a countably infinite set of predicate symbols, where $\#(\mathsf{A}) \geq$

1 denotes the arity of a predicate symbol $\mathsf{A} \in \mathbb{A}$. The CL formulæ are inductively described by the following syntax:

$$\begin{array}{rcl} \phi & ::= & \mathsf{emp} \mid x = y \mid C(x) \mid I(x_1, \ldots, x_{\#(I)}) \mid \mathsf{state}(x, C, q) \mid \\ & & \mathsf{true} \mid \mathsf{A}(x_1, \ldots, x_{\#(\mathsf{A})}) \mid \phi_1 * \phi_2 \mid \phi_1 \wedge \phi_2 \mid \neg\phi_1 \mid \exists x \,.\, \phi_1 \end{array}$$

where $C \in \mathfrak{C}$ and $I \in \mathfrak{I}$ are component and interaction types, respectively, $q \in \mathbb{Q}$ are states, $\mathsf{A} \in \mathbb{A}$ are predicate symbols and $x, y, x_1, \ldots \in \mathbb{V}$ are variables. The atomic formulæ $C(x)$, $I(x_1, \ldots, x_{\#(I)})$, $\mathsf{state}(x, C, q)$ and $\mathsf{A}(x_1, \ldots, x_{\#(\mathsf{A})})$ are called component, interaction, state and predicate atoms, respectively. We use the shorthands $\mathsf{false} \stackrel{\mathsf{def}}{=} \neg\mathsf{true}$, $\phi_1 \vee \phi_2 \stackrel{\mathsf{def}}{=} \neg(\neg\phi_1 \wedge \neg\phi_2)$, $\forall x \,.\, \phi_1 \stackrel{\mathsf{def}}{=} \neg(\exists x \,.\, \neg\phi_1)$ and $C^q(x) \stackrel{\mathsf{def}}{=} C(x) \wedge \mathsf{state}(x, C, q)$. Let $\mathrm{size}(\phi)$ denote the number of occurrences of symbols in ϕ.

Intuitively, a formula emp describes configurations with empty structure, $C(x)$ describes configurations with structures consisting of a single instance of the component type C, indexed by x, and $I(x_1, \ldots, x_k)$ describes a single interaction of type I, between components indexed by $x_1, \ldots, x_k$, respectively. The formula $C^{q_1}(x_1) * \ldots * C^{q_k}(x_k) * I(x_1, \ldots, x_k)$ describes a structure consisting of k *pairwise distinct* instances of the component type C, in states $q_1, \ldots, q_k$, respectively, joined by an interaction of type I. The formula $I(x_1, \ldots, x_k) * I(x'_1, \ldots, x'_k)$ states the existence of two interactions of type I, with *distinct* tuples of indices, given by the values of $\langle x_1, \ldots, x_k \rangle$ and $\langle x'_1, \ldots, x'_k \rangle$, respectively, i.e. the values of x_i and x'_i must differ for at least one $i \in [1, k]$.

A formula is said to be *predicateless* if it has no occurrences of predicate atoms. By $\mathrm{fv}(\phi)$ we denote the set of free variables that do not occur within the scope of an existential quantifier. A formula is *quantifier-free* if it has no occurrence of existential quantifiers. By convention, the formulæ $\divideontimes_{\phi \in F} \phi$ and $\bigwedge_{\phi \in F} \phi$ are considered to be the same as emp and true, respectively, when F is the empty set of formulæ. A *substitution* is a partial mapping $\theta : \mathbb{V} \to \mathbb{V}$ and the formula $\phi\theta$ is the result of replacing each free variable $x \in \mathrm{fv}(\phi) \cap \mathrm{dom}(\theta)$ by $\theta(x)$ in ϕ. We denote by $[x_1/y_1, \ldots, x_k/y_k]$ the substitution that replaces x_i with y_i, for all $i \in [1, k]$.

The semantics of predicateless CL formulæ is given in terms of configurations (σ, ν, ρ), with structure $\sigma = \langle C_1^\sigma, \ldots, C_n^\sigma, I_1^\sigma, \ldots, I_m^\sigma \rangle$, by a satisfaction relation $(\sigma, \nu, \rho) \models \phi$ defined inductively on the structure of the formula ϕ. In particular, the interpretation of the separated conjunction $*$ relies on the following notion of *composition* of structures and configurations:

Definition 3 (Composition) *Two structures σ_1 and σ_2 are* disjoint*, denoted $\sigma_1 \perp \sigma_2$, if $C^{\sigma_1} \cap C^{\sigma_2} = \emptyset$, for all $C \in \mathfrak{C}$ and $I^{\sigma_1} \cap I^{\sigma_2} = \emptyset$, for all $I \in \mathfrak{I}$. Their* disjoint union *is the structure $\sigma_1 \uplus \sigma_2 \stackrel{\mathsf{def}}{=} \langle C_1^{\sigma_1} \cup C_1^{\sigma_2}, \ldots, C_n^{\sigma_1} \cup C_n^{\sigma_2}, I_1^{\sigma_1} \cup I_1^{\sigma_2}, \ldots, I_m^{\sigma_1} \cup I_m^{\sigma_2} \rangle$, which is undefined if $\sigma_1 \not\perp \sigma_2$. The* composition *of configurations is $(\sigma_1, \nu_1, \rho_1) \bullet (\sigma_2, \nu_2, \rho_2) \stackrel{\mathsf{def}}{=} (\sigma_1 \uplus \sigma_2, \nu_1, \rho_1)$ and is defined iff $\sigma_1 \perp \sigma_2$, $\nu_1 = \nu_2$ and $\rho_1 = \rho_2$. Composition is lifted to $\mathcal{P}(\Gamma)^\top$ as $\Gamma_1 \bullet \Gamma_2 \stackrel{\mathsf{def}}{=} \{\gamma_1 \bullet \gamma_2 \mid \gamma_i \in \Gamma_i,\ i = 1, 2\}$ and $\Gamma_1 \bullet \top = \top \bullet \Gamma_1 = \top$, for each $\Gamma_1, \Gamma_2 \in \mathcal{P}(\Gamma)^\top$. We consider the partial order $\gamma_1 \sqsubseteq \gamma_2$ iff $\gamma_2 = \gamma_1 \bullet \gamma_0$, for some $\gamma_0 \in \Gamma$.*

It is worth mentioning that $(\Gamma, \bullet)$ is a partial commutative and cancellative multi-unit monoid, i.e. if $\gamma_1 \bullet \gamma_2$ and $\gamma_1 \bullet \gamma_3$ are both defined and equal, then $\gamma_2 = \gamma_3$ (cancellativity), and for each $\gamma \in \Gamma$, there exists a unique unit element $\gamma_0 \in \Gamma$, namely the configuration

with empty structure and same store and state map as γ, such that $\gamma \bullet \gamma_0 = \gamma_0 \bullet \gamma = \gamma$. In other words, $(\Gamma, \bullet)$ is a *multi-unit separation algebra* [**?**, **?**]. Satisfaction of predicateless formulæ is defined inductively on the structure of formulæ:

$$
\begin{array}{llll}
(\sigma,\nu,\rho) & \models & \mathsf{emp} & \text{iff } C_i^\sigma = \emptyset, \text{ for all } i \in [1,n] \text{ and } I_j^\sigma = \emptyset, \text{ for all } j \in [1,m] \\
(\sigma,\nu,\rho) & \models & x = y & \text{iff } \nu(x) = \nu(y) \\
(\sigma,\nu,\rho) & \models & C_k(x) & \text{iff } C_k^\sigma = \{\nu(x)\},\ C_i^\sigma = \emptyset, \text{ for all } i \in [1,n]\setminus\{k\} \\
 & & & \quad \text{and } I_j^\sigma = \emptyset, \text{ for all } j \in [1,m] \\
(\sigma,\nu,\rho) & \models & I_k(x_1,\ldots,x_{\#(I_k)}) & \text{iff } C_i^\sigma = \emptyset, \text{ for all } i \in [1,n],\ I_k^\sigma = \{\langle \nu(x_1),\ldots,\nu(x_{\#(I_k)})\rangle\} \\
 & & & \quad \text{and } I_j^\sigma = \emptyset, \text{ for all } j \in [1,m]\setminus\{k\} \\
(\sigma,\nu,\rho) & \models & \mathsf{state}(x,C,q) & \text{iff } \rho(\nu(x),C) = q \\
(\sigma,\nu,\rho) & \models & \phi_1 * \phi_2 & \text{iff there exist configurations } \gamma_1 \text{ and } \gamma_2, \text{ such that} \\
 & & & \quad (\sigma,\nu,\rho) = \gamma_1 \bullet \gamma_2 \text{ and } \gamma_i \models \phi_i, \text{ for both } i = 1,2.
\end{array}
$$

The semantics of the boolean formulæ true, $\phi_1 \wedge \phi_2$ and $\exists x \,.\, \phi_1$ is defined as usual, with existential quantifiers ranging over the set $\mathbb{U}$ of indices. We do not consider magic wand[2] ($-\!\!*$) in CL, as we make no explicit use of this logical connective in the rest of this paper.

3.1 Symbolic Configurations and Inductive Definitions

The CL logic is used to describe configurations of distributed systems of unbounded size, by means of predicate symbols, defined inductively by a given set of rules. This style of specification recalls the usual definitions of Algebraic Datatypes [**?**] or heap memory shapes in Separation Logic (SL) [**?**]. For reasons related to the existence of (least) fixed points, the definitions of predicates are given in a restricted fragment of the logic.

A *symbolic configuration* is a quantifier-free CL formula of the form $\xi \wedge \pi$, where the formulæ ξ and π, called *spatial* and *pure*, respectively, are generated inductively by the following syntax:

$$
\begin{array}{llll}
\xi & ::= & \mathsf{emp} \mid C(x) \mid I(x_1,\ldots,x_{\#(I)}) \mid \mathsf{A}(x_1,\ldots,x_{\#(\mathsf{A})}) \mid \xi_1 * \xi_2 & \text{(spatial formulæ)} \\
\pi & ::= & \mathsf{true} \mid x = y \mid \mathsf{state}(x,C,q) \mid \pi_1 \wedge \pi_2 & \text{(pure formulæ)}
\end{array}
$$

Intuitively, pure formulæ are independent of the structure σ in a configuration (σ,ν,ρ). Their truth value is determined exclusively by the store ν and the state map ρ, i.e. π and $\pi * \mathsf{true}$ are equivalent, if π is pure. We write $x \simeq_\phi y$ ($x \not\simeq_\phi y$) if and only if the equality (disequality) between x and y is asserted by the symbolic configuration ϕ. Note that $x \not\simeq_\phi y$ is not the negation of $x \simeq_\phi y$, as we can have, e.g. $x \simeq_{\mathsf{emp} * x=z * z=y} y$ and $x \not\simeq_{C(x)*C(y)} y$. We denote by $\mathbb{S}$ the set of symbolic configurations.

Definition 4 (Systems of Inductive Definitions) *A* system of inductive definitions *(SID) is a set $\mathcal{D}$ of rules of the form $\mathsf{A}(x_1,\ldots,x_{\#(\mathsf{A})}) \leftarrow \exists y_1 \ldots \exists y_k \,.\, \phi$, where ϕ is a symbolic configuration, $\mathrm{fv}(\phi) \subseteq \{x_1,\ldots,x_{\#(\mathsf{A})}\} \cup \{y_1,\ldots,y_k\}$ and each variable in $x_1,\ldots,x_{\#(\mathsf{A})}$ occurs in ϕ. Given formulæ ψ and φ, the* unfolding step $\psi \Leftarrow_{\mathcal{D}} \varphi$ *replaces a predicate atom $\mathsf{A}(z_1,\ldots,z_{\#(A)})$ in ψ with the formula $\exists y_1 \ldots \exists y_k \,.\, \phi[x_1/z_1,\ldots,x_{\#(\mathsf{A})}/z_{\#(\mathsf{A})}]$, where $\mathsf{A}(x_1,\ldots,x_{\#(\mathsf{A})}) \leftarrow \exists y_1 \ldots \exists y_k \,.\, \phi$ is a rule in $\mathcal{D}$.*

[2] A formula $\phi -\!\!* \psi$ holds in γ if for any $\gamma' \perp \gamma$, such that $\gamma' \models \phi$, we have $\gamma \bullet \gamma' \models \psi$.

The *size* and *width* of a SID are defined as $\mathrm{size}(\mathcal{D}) \stackrel{\text{def}}{=} \sum_{\mathsf{A}(x_1,\ldots,x_{\#(\mathsf{A})}) \leftarrow_{\mathcal{D}} \varphi} \mathrm{size}(\varphi)$ and $\mathrm{width}(\mathcal{D}) \stackrel{\text{def}}{=} \max_{\mathsf{A}(x_1,\ldots,x_{\#(\mathsf{A})}) \leftarrow_{\mathcal{D}} \varphi} \mathrm{size}(\varphi)$. By $\mathsf{A}(x_1,\ldots,x_{\#(\mathsf{A})}) \leftarrow_{\mathcal{D}} \varphi$, we mean that $\mathsf{A}(x_1,\ldots,x_{\#(\mathsf{A})}) \leftarrow \varphi \in \mathcal{D}$. We write $\psi \Leftarrow^{*}_{\mathcal{D}} \varphi$ for the reflexive and transitive closure of the $\Leftarrow_{\mathcal{D}}$ relation and say that φ is a *complete unfolding* of ψ, written $\psi \Leftarrow^{\circ}_{\mathcal{D}} \varphi$, if and only if $\psi \Leftarrow^{*}_{\mathcal{D}} \varphi$ and φ is a predicateless formula. We denote by $\mathrm{def}(\mathcal{D})$ the set of predicate atoms having a complete unfolding using the rules in $\mathcal{D}$. Note that $\mathrm{def}(\mathcal{D})$ is closed under renaming of free variables and the set of representatives[3] is computable by a least fixed point iteration, in time $O(\mathrm{size}(\mathcal{D}))$. Complete unfoldings define the semantics of predicate atoms, as follows:

$$(\sigma,\nu,\rho) \models_{\mathcal{D}} \mathsf{A}(x_1,\ldots,x_{\#(\mathsf{A})}) \text{ iff } (\sigma,\nu,\rho) \models \varphi, \text{ for some complete unfolding } \mathsf{A}(x_1,\ldots,x_{\#(\mathsf{A})}) \Leftarrow^{\circ}_{\mathcal{D}} \varphi$$

Example 4 *The rules below, written using the signature from the token ring Example 1, define the chains of* S *and* T *components, with at least* $h,t \in \mathbb{N}$ *components in state* $\mathfrak{n}$ *and* $\mathfrak{t}$*, respectively:*

$$\begin{array}{lll} \mathsf{chain}_{0,0}(x,x) \leftarrow \mathsf{S}(x) & \mathsf{chain}_{0,1}(x,x) \leftarrow \mathsf{S}^{\mathfrak{t}}(x) & \mathsf{chain}_{h,t}(x,z) \leftarrow \exists y.\ \mathsf{S}^{\mathfrak{t}}(x) * \mathsf{T}(x,y) * \mathsf{chain}_{h,t\dot{-}1}(y,z) \\ & \mathsf{chain}_{1,0}(x,x) \leftarrow \mathsf{S}^{\mathfrak{n}}(x) & \mathsf{chain}_{h,t}(x,z) \leftarrow \exists y.\ \mathsf{S}^{\mathfrak{n}}(x) * \mathsf{T}(x,y) * \mathsf{chain}_{h\dot{-}1,t}(y,z) \end{array}$$

where $k\dot{-}1 \stackrel{\text{def}}{=} \max(k-1,0)$*,* $k \in \mathbb{N}$*. The complete unfoldings of the predicate atom* $\mathsf{chain}_{1,1}(x,y)$ *are of the form* $\exists x_1 \ldots \exists x_{n-2}\ .\ \mathsf{S}^{\mathfrak{t}/\mathfrak{n}}(x) * \mathsf{T}(x,x_1) * \mathsf{S}^{\mathfrak{t}/\mathfrak{n}}(x_1) * \ldots * \mathsf{S}^{\mathfrak{t}/\mathfrak{n}}(x_{n-2}) * \mathsf{T}(x_{n-2},y) * \mathsf{S}^{\mathfrak{t}/\mathfrak{n}}(y)$ *with at least one* S *component in state* $\mathfrak{t}$ *and one in state* $\mathfrak{n}$*. Consequently, the configurations from Example 2 are models of the formula* $\exists x \exists y\ .\ \mathsf{chain}_{1,1}(x,y) * \mathsf{T}(y,x)$*, for all* $n \in \mathbb{N}$*.* ■

In the following, we extend the $\models_{\mathcal{D}}$ relation homomorphically to all CL formulæ. If $\gamma \models_{\mathcal{D}} \phi$, we say that γ is an $\mathcal{D}$-model of ϕ and define $[\![\phi]\!]_{\mathcal{D}} \stackrel{\text{def}}{=} \{\gamma \mid \gamma \models_{\mathcal{D}} \phi\}$. A formula ϕ is *$\mathcal{D}$-satisfiable* if $[\![\phi]\!]_{\mathcal{D}} \neq \emptyset$ and *consistent* if it is $\mathcal{D}$-satisfiable, for some SID $\mathcal{D}$. Given formulæ ϕ and ψ, we say that ϕ *$\mathcal{D}$-entails* ψ, written $\phi \models_{\mathcal{D}} \psi$, if and only if $[\![\phi]\!]_{\mathcal{D}} \subseteq [\![\psi]\!]_{\mathcal{D}}$.

3.2 Precise and Tight Formulæ

We define two restrictions on CL formulæ, for later use (§5). First, we adapt the notion of *precision*, originally introduced for SL [**?**, **?**] to CL.

Definition 5 (Precision) *A formula* φ *is* precise on a set of configurations $\mathcal{C}$ *if and only if, for every configuration* $\gamma \in \mathcal{C}$*, there exists at most one configuration* γ'*, such that* $\gamma' \sqsubseteq \gamma$ *and* $\gamma' \models_{\mathcal{D}} \varphi$*. A set* Φ *of formulæ is* precisely closed *if* φ *is precise on* $[\![\phi]\!]_{\mathcal{D}}$*, for any two formulæ* $\varphi,\phi \in \Phi$*.*

Symbolic configurations using predicate atoms are not precise for Γ, in general[4]. To understand this point, consider a structure consisting of two overlapping models of

[3] Predicate atoms $\mathsf{A}(x_1,\ldots,x_{\#(\mathsf{A})})$, where the tuple of variable names $\langle x_1,\ldots,x_{\#(\mathsf{A})}\rangle$ depends canonically on A.

[4] Unlike the predicates that define acyclic data structures (lists, trees) in SL, which are typically precise.

$\mathsf{chain}_{h,t}(x,y)$, starting and ending in x and y, respectively, with a component that branches on two interactions after x and another component that joins the two branches before y. Then $\mathsf{chain}_{h,t}(x,y)$ is not precise on configurations with such structures. On the positive side, we can state the following:

Proposition 1 *The set of symbolic configurations built using predicate atoms* $\mathsf{chain}_{h,t}(x,y)$, *for* $h,t \geq 0$ *(Example 4) is precisely closed.*

The existence of a decision procedure for the problem *given a formula* φ *and a set of configurations* $\mathcal{C}$, *is* φ *precise on* $\mathcal{C}$*?* is an open problem, considered for future work. Moreover, we are not aware of the decidability status of this problem for SL [**?**, **?**] either.

The second restriction on CL formulae forbids formulæ describing configurations with loosely dangling interactions that do not connect to components from the structure:

Definition 6 *A configuration* (σ,ν,ρ) *is* tight *if and only if, for each interaction* $\mathbf{u} \in I_j^\sigma$ *and each* $k \in [1,\#(I_j)]$, *we have* $\langle \mathbf{u} \rangle_k \in C_i^\sigma$, *where* C_i *is the unique component type such that* $\langle ports(I_j) \rangle_k \in ports(C_i)$. *A formula* φ *is* tight *if and only if every* $\mathcal{D}$*-model of* φ *is tight. A set* Φ *of formulæ is* tight *if and only if each formula* $\varphi \in \Phi$ *is tight.*

For instance, a predicate atom $\mathsf{chain}_{h,t}(x,y)$, for $h,t \geq 0$ (Example 4) is tight, because, in each configuration $(\sigma,\nu,\rho) \in [\![\mathsf{chain}_{h,t}(x,y)]\!]_{\mathcal{D}}$, the interactions $\langle i, (i \bmod n)+1 \rangle \in \mathsf{T}^\sigma$ are between the $ports(\mathsf{T}) = \langle out, in \rangle$ of the components indexed by $i, (i \bmod n)+1 \in \mathsf{S}^\sigma$. In the rest of this paper, we proceed under the following assumptions:

Assumption 1 *The set of symbolic configurations built using predicate atoms from* $\mathrm{def}(\mathcal{D})$ *is precisely closed and, moreover,* $\mathrm{def}(\mathcal{D})$ *is tight.*

As our examples show, most useful sets of predicate symbols are precisely closed (Prop. 1 and 3) whereas tightness can, moreover, be effectively decided (see Prop. 6 in §7).

4 A Language for Programming Reconfigurations

This section defines *reconfiguration* actions, that change the structure of a configuration. We distinguish between reconfigurations and atomic state changes (Def. 2), that change configurations in orthogonal ways. The interplay between the two types of actions is captured by the semantics of the sequential composition rule.

4.1 Syntax and Operational Semantics

Reconfiguration programs, ranged over by R, are inductively defined by the following syntax:

$$\begin{array}{rcl} \mathsf{R} & ::= & \mathsf{new}(C,x) \mid \mathsf{delete}(C,x) \mid \mathsf{connect}(I,x_1,\ldots,x_{\#(I)}) \mid \mathsf{disconnect}(I,x_1,\ldots,x_{\#(I)}) \\ & & \mid \mathsf{skip} \mid \mathsf{with}\ x_1,\ldots,x_k : \varphi\ \mathsf{do}\ \mathsf{R}_1\ \mathsf{od} \mid \mathsf{R}_1;\mathsf{R}_2 \mid \mathsf{R}_1+\mathsf{R}_2 \mid \mathsf{R}_1^* \end{array}$$

where $C \in \mathfrak{C}$ is a component type, $I \in \mathfrak{I}$ is an interaction type, $x, x_1, \ldots \in \mathbb{V}$ are program variables and φ is a predicateless formula of the CL logic (§3). For instance, Fig. 1 shows two reconfiguration programs, written using the component and interaction types from Example 1.

The *primitive commands* are $\mathsf{new}(C,x)$ and $\mathsf{delete}(C,x)$, that create and delete a component of type C, indexed by the store value of x, $\mathsf{connect}(I,x_1,\ \ldots,x_{\#(I)})$ and $\mathsf{disconnect}(I,x_1,\ldots,x_{\#(I)})$, that create and delete an interaction of type I, between components indexed by the store values of $x_1,\ldots,x_{\#(I)}$, respectively. Note that, since each type I interaction is associated a distinct tuple $\mathit{ports}(I) \in \mathbb{P}^{\#(I)}$ and each port belongs to at most one component type C, the component types of the participants to the interaction are uniquely identified by I. As usual, the skip command does nothing, but becomes useful in combination with the following conditional construct. We denote by $\mathfrak{P}$ the set of primitive commands, in the following.

A *conditional* is a program of the form $(\mathsf{with}\ x_1,\ldots,x_k : \varphi\ \mathsf{do}\ \mathsf{R}\ \mathsf{od})$ that performs the following steps, with no state changes in between:

1. assigns the so-called *bound variables* $x_1,\ldots,x_k$ to some indices $u_1,\ldots,u_k \in \mathbb{U}$ such that the configuration after the assignment contains a model of the predicateless formula φ, and
2. launches the first command of the program R on this configuration; after this, the remainder of R proceeds normally, in interleaving with havoc state changes.

Upon completion of R, the values of $x_1,\ldots,x_k$ are forgotten. The action is disabled if the current configuration is not a model of $\exists x_1 \ldots \exists x_k \,.\, \varphi * \mathsf{true}$. For instance, the conditional $(\mathsf{with}\ x,y,z : \mathsf{T}(x,y) * \mathsf{S}^{\mathfrak{n}}(y) * \mathsf{T}(y,z)\ \mathsf{do}\ \mathsf{R}\ \mathsf{od})$ applies the reconfiguration program R to any part of a token ring configuration (Example 2) consisting of two adjacent T interactions that share an S component in state $\mathfrak{n}$. To avoid technical complications, we assume that nested conditionals use pairwise disjoint tuples of bound variables — every program can be statically changed to meet this condition, by renaming bound variables.

The sequential composition $\mathsf{R}_1;\mathsf{R}_2$ executes R_1 followed by R_2, with an arbitrary sequence of atomic state changes (Def. 2) in between. This is because, even though being sequential, a reconfiguration program runs in parallel with the state changes that occur as a result of firing the interactions. Last, $\mathsf{R}_1 + \mathsf{R}_2$ executes either R_1 or R_2, and R^* executes R zero or more times in sequence, nondeterministically.

Example 5 *The reconfiguration program* $(\mathsf{with}\ x : \mathsf{S}^{\mathfrak{t}}(x)\ \mathsf{do}\ \mathsf{delete}(\mathsf{S},x)\ \mathsf{od})^*$ *deletes all* S *components in state* $\mathfrak{t}$ *from a token ring configuration (Example 2).* ■

It is worth noticing that the reconfiguration language does not have explicit assignments between variables. As a matter of fact, the conditionals are the only constructs that nondeterministically bind variables to indices that satisfy a given logical condition. This design choice sustains the view of a distributed system as a cloud of components and interactions in which reconfigurations can occur anywhere a local condition is met. In other words, we do not need variable assignments to traverse the architecture — the program works rather by identifying a part of the system that matches a small pattern, and applying the reconfiguration locally to that

Figure 2: Operational Semantics of the Reconfiguration Language

$$\frac{u \in \mathbb{U} \setminus C_i^\sigma \qquad \sigma' = \langle C_1^\sigma, \ldots, C_i^\sigma \cup \{u\}, \ldots, C_n^\sigma, I_1^\sigma, \ldots, I_m^\sigma \rangle}{\mathsf{new}(C_i, x) : (\sigma, \nu, \rho) \rightsquigarrow (\sigma', \nu[x \leftarrow u], \rho[(u, C_i) \leftarrow q_i^0])}$$

$$\frac{\nu(x) \in C_i^\sigma \qquad \sigma' = \langle C_1^\sigma, \ldots, C_i^\sigma \setminus \{\nu(x)\}, \ldots, C_n^\sigma, I_1^\sigma, \ldots, I_m^\sigma \rangle}{\mathsf{delete}(C_i, x) : (\sigma, \nu, \rho) \rightsquigarrow (\sigma', \nu, \rho)} \qquad \frac{\nu(x) \notin C_i^\sigma}{\mathsf{delete}(C_i, x) : (\sigma, \nu, \rho) \lightning}$$

$$\frac{\sigma' = \langle C_1^\sigma, \ldots, C_n^\sigma, I_1^\sigma, \ldots, I_j^\sigma \cup \{(\nu(x_1), \ldots, \nu(x_{\#(I_j)}))\}, \ldots, I_m^\sigma \rangle}{\mathsf{connect}(I_j, x_1, \ldots, x_{\#(I_j)}) : (\sigma, \nu, \rho) \rightsquigarrow (\sigma', \nu, \rho)}$$

$$\frac{\begin{array}{c}\mathbf{u} = \langle \nu(x_1), \ldots, \nu(x_{\#(I_j)}) \rangle \in I_j^\sigma \\ \sigma' = \langle C_1^\sigma, \ldots, C_n^\sigma, I_1^\sigma, \ldots, I_j^\sigma \setminus \{\mathbf{u}\}, \ldots, I_m^\sigma \rangle\end{array}}{\mathsf{disconnect}(I_j, x_1, \ldots, x_{\#(I_j)}) : (\sigma, \nu, \rho) \rightsquigarrow (\sigma', \nu, \rho)} \qquad \frac{\langle \nu(x_1), \ldots, \nu(x_{\#(I_j)}) \rangle \notin I_j^\sigma}{\mathsf{disconnect}(I_j, x_1, \ldots, x_{\#(I_j)}) : (\sigma, \nu, \rho) \lightning}$$

$$\frac{\begin{array}{c}u_1, u_1', \ldots, u_k, u_k' \in \mathbb{U} \qquad (\sigma, \nu[x_1 \leftarrow u_1, \ldots, x_k \leftarrow u_k], \rho) \models \varphi * \mathsf{true} \\ \mathsf{R} : (\sigma, \nu[x_1 \leftarrow u_1, \ldots, x_k \leftarrow u_k], \rho) \rightsquigarrow (\sigma', \nu', \rho')\end{array}}{\mathsf{with}\ x_1, \ldots, x_k : \varphi\ \mathsf{do}\ \mathsf{R}\ \mathsf{od} : (\sigma, \nu, \rho) \rightsquigarrow (\sigma', \nu'[x_1 \leftarrow u_1', \ldots, x_k \leftarrow u_k'], \rho')} \qquad \frac{}{\mathsf{skip} : \gamma \rightsquigarrow \gamma}$$

$$\frac{\mathsf{R}_1 : \gamma \rightsquigarrow \gamma_0 \quad \gamma_1 \in \mathfrak{h}(\gamma_0) \qquad \mathsf{R}_2 : \gamma_1 \rightsquigarrow \gamma'}{\mathsf{R}_1; \mathsf{R}_2 : \gamma \rightsquigarrow \gamma'} \qquad \frac{\mathsf{R}_1 : \gamma \rightsquigarrow \gamma'}{\mathsf{R}_1 + \mathsf{R}_2 : \gamma \rightsquigarrow \gamma'}$$

$$\frac{\mathsf{R}^n : \gamma \rightsquigarrow \gamma'}{\mathsf{R}^* : \gamma \rightsquigarrow \gamma'}\ , \mathsf{R}^n = \begin{cases} \mathsf{R}^{n-1}; \mathsf{R} & \text{if } n \geq 1 \\ \mathsf{skip} & \text{if } n = 0 \end{cases}$$

subsystem. For instance, a typical pattern for writing reconfiguration programs is $(\mathsf{with}\ \mathbf{x}_1 : \varphi_1\ \mathsf{do}\ \mathsf{R}_1\ \mathsf{od} + \ldots + \mathsf{with}\ \mathbf{x}_k : \varphi_k\ \mathsf{do}\ \mathsf{R}_k\ \mathsf{od})^*$, where $\mathsf{R}_1, \ldots, \mathsf{R}_k$ are star-free sequential compositions of primitive commands. This program continuously choses a reconfiguration sequence R_i nondeterministically and either applies it on a small part of the configuration that satisfies φ_i, or does nothing, if no such subconfiguration exists.

The operational semantics of reconfiguration programs is given by the structural rules in Fig. 2, that define the judgements $\mathsf{R} : \gamma \rightsquigarrow \gamma'$ and $\mathsf{R} : \gamma \lightning$, where γ and γ' are configurations and R is a program. Intuitively, $\mathsf{R} : \gamma \rightsquigarrow \gamma'$ means that γ' is a successor of γ following the execution of R and $\mathsf{R} : \gamma \lightning$ means that R faults in γ. The semantics of a program R is the action $\langle\langle \mathsf{R} \rangle\rangle : \Gamma \to \mathcal{P}(\Gamma)^\top$, defined as:

$$\langle\langle \mathsf{R} \rangle\rangle(\gamma) \stackrel{\mathsf{def}}{=} \begin{cases} \top & \text{if } \mathsf{R} : \gamma \lightning \\ \{\gamma' \mid \mathsf{R} : \gamma \rightsquigarrow \gamma'\} & \text{otherwise} \end{cases}$$

The only primitive commands that may fault are $\mathsf{delete}(C, x)$ and $\mathsf{disconnect}(I, x_1, \ldots, x_{\#(I)})$; for both, the premisses of the faulty rules are disjoint from the ones for normal termination, thus the action $\langle\langle \mathsf{R} \rangle\rangle$ is properly defined for all programs R. Notice the rule for sequential composition, that uses the havoc action $\mathfrak{h}$ in the premiss to capture the interleaving of state changes and reconfiguration actions.

4.2 Reconfiguration Proof System

To reason about the correctness properties of reconfiguration programs, we introduce a Hoare-style proof system consisting of a set of axioms that formalize the primitive

Figure 3: Proof System for the Reconfiguration Language

$$\frac{}{\{\mathsf{emp}\}\ \mathsf{new}(C,x)\ \{C^q(x)\}} \qquad \frac{\mathcal{B}(C) = (Q,P,q,\rightarrow)}{\dfrac{}{\{C(x)\}\ \mathsf{delete}(C,x)\ \{\mathsf{emp}\}}\quad \dfrac{}{\{\mathsf{emp}\}\ \mathsf{skip}\ \{\mathsf{emp}\}}}$$

$$\frac{}{\{\mathsf{emp}\}\ \mathsf{connect}(I,x_1,\ldots,x_{\#(I)})\ \{I(x_1,\ldots,x_{\#(I)})\}}$$

$$\frac{}{\{I(x_1,\ldots,x_{\#(I)})\}\ \mathsf{disconnect}(I,x_1,\ldots,x_{\#(I)})\ \{\mathsf{emp}\}}$$

a. Axioms for Atomic Programs

$$\frac{\{\phi \wedge (\varphi * \mathsf{true})\}\ \mathsf{R}\ \{\psi\}}{\{\forall \mathbf{x}\ .\ \neg(\varphi * \mathsf{true}) \vee \phi\}\ \mathsf{with}\ \mathbf{x} : \varphi\ \mathsf{do}\ \mathsf{R}\ \mathsf{od}\ \{\exists \mathbf{x}\ .\ \psi\}}\ \mathrm{fv}(\phi) \cap \mathbf{x} = \emptyset$$

$$\frac{\{\phi\}\ \mathsf{R}_1\ \{\phi_0\} \qquad \{\phi_1\}\ \mathsf{R}_2\ \{\psi\}}{\{\phi\}\ \mathsf{R}_1;\mathsf{R}_2\ \{\psi\}}\ \mathfrak{h}(\llbracket\phi_0\rrbracket_{\mathcal{D}}) \subseteq \llbracket\phi_1\rrbracket_{\mathcal{D}}$$

$$\frac{\{\phi\}\ \mathsf{R}_1\ \{\psi\} \qquad \{\phi\}\ \mathsf{R}_2\ \{\psi\}}{\{\phi\}\ \mathsf{R}_1 + \mathsf{R}_2\ \{\psi\}} \qquad \frac{\{\phi\}\ \mathsf{R}\ \{\phi\}}{\{\phi\}\ \mathsf{R}^*\ \{\phi\}}\ \mathfrak{h}(\llbracket\phi\rrbracket_{\mathcal{D}}) \subseteq \llbracket\phi\rrbracket_{\mathcal{D}}$$

b. Inference Rules for Composite Programs

$$\frac{\{\phi_i\}\ \mathsf{R}\ \{\psi_i\} \mid i \in [1,k]}{\{\bigvee_{i=1}^k \phi_i\}\ \mathsf{R}\ \{\bigvee_{i=1}^k \psi_i\}} \qquad \frac{\{\phi_i\}\ \mathsf{R}\ \{\psi_i\} \mid i \in [1,k]}{\{\bigwedge_{i=1}^k \phi_i\}\ \mathsf{R}\ \{\bigwedge_{i=1}^k \psi_i\}}$$

$$\frac{\{\phi'\}\ \mathsf{R}\ \{\psi'\}}{\{\phi\}\ \mathsf{R}\ \{\psi\}}\ \begin{array}{l}\phi \models_{\mathcal{D}} \phi' \\ \psi' \models_{\mathcal{D}} \psi\end{array} \qquad \frac{\{\phi\}\ \mathsf{R}\ \{\psi\}}{\{\phi * \varphi\}\ \mathsf{R}\ \{\psi * \varphi\}}\ \begin{array}{l}\mathsf{R} \in \mathfrak{L} \\ \mathsf{modif}(\mathsf{R}) \cap \mathrm{fv}(\varphi) = \emptyset\end{array}$$

c. Structural Inference Rules

commands (Fig. 3a), a set of inference rules for the composite programs (Fig. 3b) and a set of structural rules (Fig. 3c). The judgements are Hoare triples $\{\phi\}\ \mathsf{R}\ \{\psi\}$, where ϕ and ψ (called pre- and postcondition, respectively) are CL formulæ with predicate symbols interpreted by a given SID $\mathcal{D}$. The triple $\{\phi\}\ \mathsf{R}\ \{\psi\}$ is *valid*, written $\models_{\mathcal{D}} \{\phi\}\ \mathsf{R}\ \{\psi\}$, if $\langle\langle \mathsf{R} \rangle\rangle(\llbracket\phi\rrbracket_{\mathcal{D}}) \subseteq \llbracket\psi\rrbracket_{\mathcal{D}}$, where $\langle\langle \mathsf{R} \rangle\rangle(\llbracket\phi\rrbracket_{\mathcal{D}}) \stackrel{\mathsf{def}}{=} \bigcup_{\gamma \in \llbracket\phi\rrbracket_{\mathcal{D}}} \langle\langle \mathsf{R} \rangle\rangle(\gamma)$ is the result of the action $\langle\langle \mathsf{R} \rangle\rangle$ lifted to sets of configurations. Note that a triple is valid only if the program does not fault on any model of the precondition. In other words, an invalid Hoare triple $\{\phi\}\ \mathsf{R}\ \{\psi\}$ cannot distinguish between $\langle\langle \mathsf{R} \rangle\rangle(\llbracket\phi\rrbracket_{\mathcal{D}}) \not\subseteq \llbracket\psi\rrbracket_{\mathcal{D}}$ (non-faulting incorrectness) and $\langle\langle \mathsf{R} \rangle\rangle(\llbracket\phi\rrbracket_{\mathcal{D}}) = \top$ (faulting).

The axioms (Fig. 3a) give the *local specifications* of the primitive commands in the language by Hoare triples whose preconditions describe only those resources (components and interactions) necessary to avoid faulting. In particular, $\mathsf{delete}(C,x)$ and $\mathsf{disconnect}(I,x_1,\ldots,x_{\#(I)})$ require a single component $C(x)$ and an interaction $I(x_1,\ldots,x_{\#(I)})$ to complete, respectively. The rules for sequential composition and iteration (Fig 3b) use a semantic side condition based on the havoc action (Def. 2). In particular, formula ϕ is said to be *havoc invariant* (for the given SID $\mathcal{D}$) iff $\mathfrak{h}(\llbracket\phi\rrbracket_{\mathcal{D}}) \subseteq \llbracket\phi\rrbracket_{\mathcal{D}}$. For the moment, we assume the existence of an external procedure able to prove havoc conditions of the form $\mathfrak{h}(\llbracket\phi\rrbracket_{\mathcal{D}}) \subseteq \llbracket\psi\rrbracket_{\mathcal{D}}$ and defer its description to §5. Similarly, the side condition of the *consequence rule* (Fig. 3c left) consists of two semantic entailments; we give a decision procedure for checking such entailment conditions in §7.

The *frame rule* (Fig. 3c bottom-right) allows to apply the specification of a *local*

program, defined below, to a set of configurations that may contain more resources (components and interactions) than the ones asserted by the precondition. Formally, the set of local programs $\mathfrak{L}$ is the least set that contains the primitive commands $\mathfrak{P}$ and is closed under the application of the following rules:

$$\mathsf{R} \in \mathfrak{L} \Rightarrow \mathsf{with}\ \mathbf{x} : \pi\ \mathsf{do}\ \mathsf{R}\ \mathsf{od} \in \mathfrak{L}, \text{ if } \pi \text{ is pure} \qquad \mathsf{R}_1, \mathsf{R}_2 \in \mathfrak{L} \Rightarrow \mathsf{R}_1 + \mathsf{R}_2 \in \mathfrak{L}$$

The extra resources, not required to execute a local program, are specified by a frame φ, whose free variables are not modified by the program. Formally, the set of variables modified by a local program $\mathsf{R} \in \mathfrak{L}$ is defined inductively on its structure:

$$\mathsf{modif}(\mathsf{new}(C,x)) \stackrel{\mathsf{def}}{=} \{x\} \qquad \mathsf{modif}(\mathsf{R}) \stackrel{\mathsf{def}}{=} \emptyset, \text{ for all } \mathsf{R} \in \mathfrak{P} \setminus \{\mathsf{new}(C,x) \mid C \in \mathfrak{C}, x \in \mathbb{V}\}$$
$$\mathsf{modif}(\mathsf{with}\ \mathbf{x} : \varphi\ \mathsf{do}\ \mathsf{R}\ \mathsf{od}) \stackrel{\mathsf{def}}{=} \mathbf{x} \cup \mathsf{modif}(\mathsf{R}) \qquad \mathsf{modif}(\mathsf{R}_1 + \mathsf{R}_2) \stackrel{\mathsf{def}}{=} \mathsf{modif}(\mathsf{R}_1) \cup \mathsf{modif}(\mathsf{R}_2)$$

We write $\vdash \{\phi\}\ \mathsf{R}\ \{\psi\}$ if and only if $\{\phi\}\ \mathsf{R}\ \{\psi\}$ can be derived from the axioms using the inference rules from Fig. 3 and show the soundness of the proof system in the following. For a set Γ of configurations, we denote by $\min_{\sqsubseteq}\Gamma$ the set of $\sqsubseteq$-minimal elements of Γ (Def. 3). The next lemma gives sufficient conditions for the soundness of the axioms (Fig. 3a):

Lemma 1 *For each axiom* $\{\phi\}\ \mathsf{R}\ \{\psi\}$*, where* $\mathsf{R} \in \mathfrak{P}$*, the following hold:*

1. $[\![\phi]\!]_{\mathcal{D}} = \min_{\sqsubseteq}\{\gamma \in \Gamma \mid \langle\langle \mathsf{R} \rangle\rangle(\gamma) \neq \top\}$,
2. $\langle\langle \mathsf{R} \rangle\rangle([\![\phi]\!]_{\mathcal{D}}) = [\![\psi]\!]_{\mathcal{D}}$.

The frame rule is sound only for programs whose semantics are local actions, defined below:

Definition 7 (Locality) *Given a set of variables* $X \subseteq \mathbb{V}$*, an action* $f : \Gamma \to \mathcal{P}(\Gamma)^{\top}$ *is* local for X *if* $f(\gamma_1 \bullet \gamma_2) \subseteq f(\gamma_1) \bullet \{\gamma_2\}\!\uparrow^X$ *for all* $\gamma_1, \gamma_2 \in \Gamma$*, where, for any set* $\mathcal{C}$ *of configurations, we define:*

$$\mathcal{C}\!\uparrow^X \stackrel{\mathsf{def}}{=} \{(\sigma, \nu', \rho') \mid (\sigma, \nu, \rho) \in \mathcal{C}, \forall x \in \mathbb{V} \setminus X \,.\, \nu'(x) = \nu(x),\ \forall u \in \mathbb{U} \setminus \nu(X)\ \forall C \in \mathfrak{C} \,.\, \rho'(u,C) = \rho(u,C)\}.$$

Intuitively, an action local for X allows for the change of the store values of the variables in X and the states of the components indexed by these values, only. Essentially, the $\mathsf{new}(C,x)$ commands are local for $\{x\}$, because the fresh index associated to x is nondeterministically chosen and the state of the new component of type C is the initial state of the behavior $\mathcal{B}(C)$, whereas the other primitive commands are local for the empty set.

We show that the semantics of every local program is a local action. Moreover, $\mathfrak{L}$ is precisely the set of programs with local semantics, as it can be easily seen that $\mathsf{with}\ \mathbf{x} : \psi\ \mathsf{do}\ \mathsf{R}\ \mathsf{od}$ conditionals with non-pure conditions and sequential compositions (thus iterations) are not local.

Lemma 2 *For every program* $\mathsf{R} \in \mathfrak{L}$*, the action* $\langle\langle \mathsf{R} \rangle\rangle$ *is local for* $\mathsf{modif}(\mathsf{R})$.

Example 6 *To understand why* $\mathfrak{L}$ *defines the precise set of local commands, consider the following programs:*

- with $x : C(x)$ do delete(C,x) od *is not local because, if we consider* γ_1 *to be a configuration containing only one component of type* C *and* γ_2 *to be an empty configuration, such that* $\gamma_1 \bullet \gamma_2$ *is defined, we have:*

$$\langle\langle \text{with } x : C(x) \text{ do delete}(C,x) \text{ od} \rangle\rangle(\gamma_2 \bullet \gamma_1) = \langle\langle \text{with } x : C(x) \text{ do delete}(C,x) \text{ od} \rangle\rangle(\gamma_1) = \{\gamma_2\}$$
$$\langle\langle \text{with } x : C(x) \text{ do delete}(C,x) \text{ od} \rangle\rangle(\gamma_2) = \langle\langle \text{with } x : C(x) \text{ do delete}(C,x) \text{ od} \rangle\rangle(\gamma_2) \bullet \{\gamma_1\}\uparrow^x = \emptyset \ .$$

- skip;skip *is not local because, if we take* γ_1 *and* γ_2*, such that* $\gamma_1 \models \mathsf{S}^{\mathsf{t}}(x)$ *and* $\gamma_2 \models \mathsf{T}(x,y) * \mathsf{S}^{\mathsf{n}}(y)$*, with the signature from Example 1, we have:*

$$\begin{array}{rcl} \langle\langle \text{skip;skip} \rangle\rangle(\gamma_1 \bullet \gamma_2) & = & [\![\mathsf{S}^{\mathsf{t}}(x) * \mathsf{T}(x,y) * \mathsf{S}^{\mathsf{n}}(y)]\!] \cup \\ & & [\![\mathsf{S}^{\mathsf{n}}(x) * \mathsf{T}(x,y) * \mathsf{S}^{\mathsf{t}}(y)]\!] \\ \langle\langle \text{skip;skip} \rangle\rangle(\{\gamma_2\}) & = & [\![\mathsf{S}^{\mathsf{t}}(x) * \mathsf{T}(x,y) * \mathsf{S}^{\mathsf{n}}(y)]\!] \ \blacksquare \end{array}$$

The soundness of the proof system in Fig. 3 is a consequence of the soundness of each axiom and inference rule, stated below:

Theorem 1 *Given a SID* $\mathcal{D}$*, for any triple* $\{\phi\}\ \mathsf{R}\ \{\psi\}$*, if* $\vdash \{\phi\}\ \mathsf{R}\ \{\psi\}$ *then* $\models_{\mathcal{D}} \{\phi\}\ \mathsf{R}\ \{\psi\}$.

Reconfiguration proofs can often be simplified, by safely skipping the check of one or more havoc side conditions of sequential compositions, as explained below. A sequential composition of reconfiguration commands of the form

$$\text{disconnect}(I_1,\mathbf{x}_1);\ldots\text{disconnect}(I_k,\mathbf{x}_k);\text{connect}(I_{k+1},\mathbf{x}_{k+1});\ldots\text{connect}(I_\ell,\mathbf{x}_\ell)$$

is said to be a *single reversal sequence*. Such reconfiguration programs first disconnect components and then reconnect them in a different way (see Fig. 7 for an example). For such programs, only the first and last application of the sequential composition rule require havoc invariance proofs. For space reasons, we only sketch the explanation below:

Remark 1 *Given the following annotation of a single reversal reconfiguration sequence:*

$$\begin{array}{lllll} \{\phi_0\}\ \text{disconnect}(I_1,\mathbf{x}_1); & \{\phi_1\} & \ldots & \{\phi_{k-1}\}\ \text{disconnect}(I_k,\mathbf{x}_k); & \{\phi_k\} \\ \quad\ \text{connect}(I_{k+1},\mathbf{x}_{k+1}); & \{\phi_{k+1}\} & \ldots & \{\phi_{\ell-1}\}\ \text{connect}(I_\ell,\mathbf{x}_\ell) & \{\phi_\ell\} \end{array}$$

such that $\{\phi_{i-1}\}\ \text{disconnect}(I_i,\mathbf{x}_i)\ \{\phi_i\}$*,* $i \in [1,k]$ *and* $\{\phi_{j-1}\}\ \text{connect}(I_j,\mathbf{x}_j)\ \{\phi_j\}$*,* $j \in [k+1,\ell]$ *are valid triples and the formulæ* ϕ_1 *and* $\phi_{\ell-1}$ *are havoc invariant, we show that the Hoare triple:*
$\{\phi_0\}\ \text{disconnect}(I_1,\mathbf{x}_1);\ldots\text{disconnect}(I_k,\mathbf{x}_k);\text{connect}(I_{k+1},\mathbf{x}_{k+1});\ldots\text{connect}(I_\ell,\mathbf{x}_\ell)\ \{\phi_\ell\}$

is valid. To this end, it suffices to prove the following points:

- $\phi_1,\ldots,\phi_k$ *are havoc invariants, by induction on k, as havoc invariance is preserved by interaction removal, and*

- $\phi_{\ell-1}$ *is havoc invariant w.r.t. the state changes in each* $\mathcal{D}$*-model of* $\phi_{\ell-2},\ldots,\phi_k$*, respectively. Intuitively, this is because every interaction from a configuration* $\gamma_j \in [\![\phi_j]\!]_{\mathcal{D}}$ *occurs also in every configuration* $\gamma_{\ell-1} \in [\![\phi_{\ell-1}]\!]_{\mathcal{D}}$*, hence each state change has the same effect in* γ_j*,* $j \in [k,\ell-2]$ *and* $\gamma_{\ell-1}$.

4.3 A Reconfiguration Proof Example

Example 7 *We prove that the outcome of the reconfiguration program from Fig. 1 (Listing 2), started in a token ring configuration with at least two* S *components in state* $\mathfrak{n}$ *and at least one in state* $\mathfrak{t}$*, is a token ring with at least one component in each state. The pre- and postcondition are* $\exists a,b\,.\,\mathsf{chain}_{2,1}(a,b)*\mathsf{T}(b,a)$ *and* $\exists a,b\,.\,\mathsf{chain}_{1,1}(a,b)*\mathsf{T}(b,a)$*, respectively, with the definitions of* $\mathsf{chain}_{h,t}(x,y)$ *given in Example 4.*

$\{\exists a,b\ .\ \mathsf{chain}_{2,1}(a,b)*\mathsf{T}(b,a)\}$
$\left\{\begin{array}{l}\forall x,y,z\ .\ \neg(\mathsf{T}(x,y)*\mathsf{S}^{\mathfrak{n}}(y)*\mathsf{T}(y,z)*\mathsf{true})\ \vee\\ \exists a,b.\ \mathsf{chain}_{2,1}(a,b)*\mathsf{T}(b,a)\end{array}\right\}$
with $x,y,z:\mathsf{T}(x,y)*\mathsf{S}^{\mathfrak{n}}(y)*\mathsf{T}(y,z)$ **do**
$\left\{\exists a\exists b\ .\ \mathsf{chain}_{2,1}(a,b)*\mathsf{T}(b,a)\wedge\left(\mathsf{T}(x,y)*\mathsf{S}^{\mathfrak{n}}(y)*\mathsf{T}(y,z)*\mathsf{true}\right)\right\}$ $(\star)$
$\left\{\mathsf{T}(x,y)*\boxed{\mathsf{S}^{\mathfrak{n}}(y)*\mathsf{T}(y,z)*\mathsf{chain}_{1,1}(z,x)}\right\}$
`disconnect(T,x,y);`
$\left\{\boxed{\mathsf{S}^{\mathfrak{n}}(y)}*\mathsf{T}(y,z)*\boxed{\mathsf{chain}_{1,1}(z,x)}\right\}$ (hinv)
`disconnect(T,y,z);`
$\left\{\mathsf{S}^{\mathfrak{n}}(y)*\boxed{\mathsf{chain}_{1,1}(z,x)}\right\}$ (hinv)
`delete(S,y);`
$\left\{\boxed{\mathsf{chain}_{1,1}(z,x)}\right\}$ (hinv)
`connect(T,x,z)`
$\{\mathsf{chain}_{1,1}(z,x)*\mathsf{T}(x,z)\}$
od
$\{\exists a\exists b\ .\ \mathsf{chain}_{1,1}(a,b)*\mathsf{T}(b,a)\}$

The inference rule for conditional programs sets up the precondition $(\star)$ *for the body of the conditional. This formula is equivalent to* $\mathsf{T}(x,y)*\mathsf{S}^{\mathfrak{n}}(y)*\mathsf{T}(y,z)*\mathsf{chain}_{1,1}(z,x)$*. To understand this, we derive from* $(\star)$*:*

$$\begin{array}{l}\exists a\exists b\,.\,\mathsf{chain}_{2,1}(a,b)*\mathsf{T}(b,a)\wedge(\mathsf{T}(x,y)*\mathsf{S}^{\mathfrak{n}}(y)*\mathsf{T}(y,z)*\mathsf{true})\\ \equiv\exists\bar{x}\exists\bar{y}\exists\bar{z}\,.\,\mathsf{T}(\bar{x},\bar{y})*\mathsf{S}^{\mathfrak{n}}(\bar{y})*\mathsf{T}(\bar{y},\bar{z})*\mathsf{chain}_{1,1}(\bar{z},\bar{x})\wedge(\mathsf{T}(x,y)*\mathsf{S}^{\mathfrak{n}}(y)*\mathsf{T}(y,z)*\mathsf{true})\\ \equiv\mathsf{T}(x,y)*\mathsf{S}^{\mathfrak{n}}(y)*\mathsf{T}(y,z)*\mathsf{chain}_{1,1}(z,x),\end{array}$$

where the equivalence of the formulæ $\exists a\exists b\,.\,\mathsf{chain}_{2,1}(a,b)*\mathsf{T}(b,a)$ *and* $\exists\bar{x}\exists\bar{y}\exists\bar{z}\,.\,\mathsf{T}(\bar{x},\bar{y})*\mathsf{S}^{\mathfrak{n}}(\bar{y})*\mathsf{T}(\bar{y},\bar{z})*\mathsf{chain}_{1,1}(\bar{z},\bar{x})$ *can be proven by the decision procedure from §7 (see Examples 9 and 11).*

The following four annotations above are obtained by applications of the axioms and the frame rule (the frame formulæ are displayed within boxes). The sequential composition rule is applied by proving first that the annotations marked as (hinv) *are havoc invariant.* ■

4.4 Another Example of a Reconfiguration Proof

Example 8 *In previous examples, we have looked at the reconfiguration program from Fig. 1 (Listing 2) which allows us to delete components from a token ring. An orthogonal operation is the addition of new components into a ring such that the resulting system remains a valid token ring. Figure 4 contains such a reconfiguration program.*

Figure 4: Another Reconfiguration Program for the Parametric Token Ring System

Listing 3: New Component

```
1    with x,z : T(x,z) do
2      disconnect(T,x,z);
3      new(S,y);
4      connect(T,y,z);
5      connect(T,x,y) od
```

If the precondition states that the system is a valid token ring (with at least one component in state $\mathfrak{n}$ *and at least another one in state* $\mathfrak{t}$*), then the execution of the program yields again a valid token ring (with at least two component in state* $\mathfrak{n}$ *- the new component is added without a token).*

$\{\exists a,b \;.\; \mathsf{chain}_{1,1}(a,b) * \mathsf{T}(b,a)\}$
$\left\{ \begin{array}{l} \forall x,z \;.\; \neg(\mathsf{T}(x,z) * \mathsf{true}) \;\vee \\ \exists a,b.\; \mathsf{chain}_{1,1}(a,b) * \mathsf{T}(b,a) \end{array} \right\}$
with $x,z : \mathsf{T}(x,z)$ **do**
$\left\{\exists a \exists b \;.\; \mathsf{chain}_{1,1}(a,b) * \mathsf{T}(b,a) \wedge \left(\mathsf{T}(x,z) * \mathsf{true}\right)\right\}$ $(\star)$
$\left\{\mathsf{T}(x,z) * \boxed{\mathsf{chain}_{1,1}(z,x)}\right\}$
`disconnect(T,x,z);`
$\{\mathsf{chain}_{1,1}(z,x)\}$ *(hinv)*
`new(S,y);`
$\left\{\mathsf{S}^{\mathfrak{n}}(y) * \boxed{\mathsf{chain}_{1,1}(z,x)}\right\}$ *(hinv)*
`connect(T,y,z);`
$\left\{\boxed{\mathsf{S}^{\mathfrak{n}}(y)} * \mathsf{T}(y,z) * \boxed{\mathsf{chain}_{1,1}(z,x)}\right\}$
$\{\mathsf{chain}_{2,1}(y,x)\}$ *(hinv)*
`connect(T,x,y)`
$\{\mathsf{chain}_{2,1}(y,x) * \mathsf{T}(x,y)\}$
od
$\{\exists a \exists b \;.\; \mathsf{chain}_{2,1}(a,b) * \mathsf{T}(b,a)\}$

Again, the precondition $(\star)$ *is given by the inference rule for conditional programs. Then we can derive that*

$$\exists a \exists b \;.\; \mathsf{chain}_{1,1}(a,b) * \mathsf{T}(b,a) \wedge \left(\mathsf{T}(x,z) * \mathsf{true}\right) \equiv \mathsf{T}(x,z) * \mathsf{chain}_{1,1}(z,x).$$

In the subsequent lines, some axioms and the frame rule are applied (here, the frame is displayed in the postconditions of the commands within the boxes). The annotations marked as (hinv) *must be shown to be havoc invariant and then the sequential composition rule is applied to complete the proof.* ■

5 The Havoc Proof System

This section describes a set of axioms and inference rules for proving the validity of *havoc queries* of the form $\mathfrak{h}([\![\phi]\!]_{\mathcal{D}}) \subseteq [\![\psi]\!]_{\mathcal{D}}$, where ϕ and ψ are CL formulæ interpreted over a SID $\mathcal{D}$ and $\mathfrak{h}$ is the havoc action (Def. 2). Let $\mathcal{D}$ be a fixed SID for the rest of this section. A havoc query is valid if and only if, in each $\mathcal{D}$-model (σ,ν,ρ) of ϕ, by firing a sequence of enabled interactions from σ, we obtain a configuration[5] (σ,ν,ρ') that is a $\mathcal{D}$-model of ψ. Such queries are used as side conditions in the rules for sequential composition and iteration (Fig. 3b) of reconfiguration programs. Thus, having a proof system for the validity of havoc queries is crucial for the applicability of the rules in Fig. 3 to obtain proofs of reconfiguration programs.

For reasons of conciseness and scalability, the havoc proof system uses a compositional rule, able to split a query of the form $\mathfrak{h}([\![\phi_1 * \phi_2]\!]_{\mathcal{D}}) \subseteq [\![\psi_1 * \psi_2]\!]_{\mathcal{D}}$ into two queries of the form $\mathfrak{h}([\![\phi_i * \varphi_i]\!]_{\mathcal{D}} \subseteq [\![\psi_i * \varphi_i]\!]_{\mathcal{D}})$, where each φ_i defines a simple abstraction of the effect of executing the system described by ϕ_{3-i} over the one described by ϕ_i, for $i = 1,2$. The formulæ φ_1 and φ_2 can be viewed as the environment assumptions of a parallel composition proof rule [**?**, **?**]. But first, reasoning about havoc actions compositionally requires a relaxation of the definition of atomic state changes (Def. 2):

Definition 8 *Given an interaction type I and a tuple $\mathbf{u} = \langle u_1, \ldots, u_{\#(I)} \rangle$, such that $ports(I) = \langle p_1, \ldots, p_{\#(I)} \rangle$ and $p_j \in P_{i_j}$, where C_{i_j} is a component type of behavior $\mathcal{B}(C_{i_j}) \stackrel{\text{def}}{=} (Q_{i_j}, P_{i_j}, q^0_{i_j}, \rightarrow_{i_j})$, for all $j \in [1,\#(I)]$, the* open state change $\mathfrak{o}[I,\mathbf{u}]$ *maps a configuration (σ,ν,ρ) into the set of configurations (σ,ν,ρ'), such that:*

1. *for all $j \in [1,\#(I)]$, if $u_j \in C^{\sigma}_{i_j}$ then $\rho(u_j,C_{i_j}) \xrightarrow{p_j}_{i_j} \rho'(u_j,C_{i_j})$, and*
2. *$\rho(u,C) = \rho'(u,C)$, for all $(u,C) \in (\mathbb{U} \times \mathfrak{C}) \setminus \{(u_j,C_{i_j}) \mid j \in [1,\#(I)]\}$*

if $\mathbf{u} \in I^{\sigma}$ and the empty set, otherwise.

The only difference with Def. 2 is point (1), i.e. instead of requiring all components involved in an interaction to be part of the structure, now the interaction can fire even if some components are not present, hence the name *open* state change.

5.1 Regular Expressions and Havoc Triples

Proving the validity of a statement $\mathfrak{h}([\![\phi]\!]_{\mathcal{D}}) \subseteq [\![\psi]\!]_{\mathcal{D}}$ involves reasoning about the sequences of atomic state changes that define the outcome of the havoc action. We specify languages of such sequences using extended regular expressions, defined inductively by the following syntax:

$$\mathsf{L} ::= \varepsilon \mid \Sigma[\alpha] \mid \mathsf{L}_1 \cdot \mathsf{L}_2 \mid \mathsf{L}_1 \cup \mathsf{L}_2 \mid \mathsf{L}_1^* \mid \mathsf{L}_1 \bowtie_{\eta_1,\eta_2} \mathsf{L}_2$$

where ε denotes the empty string, $\Sigma[\alpha]$ is an *alphabet symbol* associated with either an interaction atom or a *precise* predicate atom α (Def. 5) and $\cdot$, $\cup$ and * are the usual

[5] We recall that an atomic state change can only change the state map, not the structure nor the store.

concatenation, union and Kleene star. By $\mathsf{L}_1 \bowtie_{\eta_1,\eta_2} \mathsf{L}_2$ we denote the interleaving (zip) product of the languages described by L_1 and L_2 with respect to the sets η_1 and η_2 of alphabet symbols of the form $\Sigma[\alpha]$, respectively.

The language of a regular expression L in a configuration $\gamma = (\sigma, \nu, \rho)$ is formally defined below:

$$\langle\langle \varepsilon \rangle\rangle^\gamma \stackrel{\text{def}}{=} \{\varepsilon\} \qquad \langle\langle \Sigma[\alpha] \rangle\rangle^\gamma \stackrel{\text{def}}{=} \{[I, \mathbf{u}] \mid \mathbf{u} \in I^{\sigma'}, \sigma' \sqsubseteq \sigma, (\sigma', \nu, \rho) \models_{\mathcal{D}} \alpha\}$$

$$\langle\langle \mathsf{L}_1 \cdot \mathsf{L}_2 \rangle\rangle^\gamma \stackrel{\text{def}}{=} \{w_1 w_2 \mid w_i \in \langle\langle \mathsf{L}_i \rangle\rangle^\gamma,\ i = 1,2\} \qquad \langle\langle \mathsf{L}_1 \cup \mathsf{L}_2 \rangle\rangle^\gamma \stackrel{\text{def}}{=} \langle\langle \mathsf{L}_1 \rangle\rangle^\gamma \cup \langle\langle \mathsf{L}_2 \rangle\rangle^\gamma$$

$$\langle\langle \mathsf{L}^* \rangle\rangle^\gamma \stackrel{\text{def}}{=} \bigcup_{i \geq 0} \langle\langle \mathsf{L}^i \rangle\rangle^\gamma \qquad \langle\langle \mathsf{L}_1 \bowtie_{\eta_1,\eta_2} \mathsf{L}_2 \rangle\rangle^\gamma \stackrel{\text{def}}{=} \{w \mid w{\downarrow_{\langle\langle \eta_i \rangle\rangle^\gamma}} \in \langle\langle \mathsf{L}_i \rangle\rangle^\gamma,\ i = 1,2\}$$

where $\langle\langle \eta \rangle\rangle^\gamma \stackrel{\text{def}}{=} \{\langle\langle \Sigma[\alpha] \rangle\rangle^\gamma \mid \Sigma[\alpha] \in \eta\}$ and $w{\downarrow_{\langle\langle \eta \rangle\rangle^\gamma}}$ is the word obtained from w by deleting each symbol not in $\langle\langle \eta \rangle\rangle^\gamma$ from it. The i-th composition of L with itself is defined, as usual, by $\mathsf{L}^0 \stackrel{\text{def}}{=} \varepsilon$ and $\mathsf{L}^{i+1} = \mathsf{L}^i \cdot \mathsf{L}$, for $i \geq 0$. We stress the role of precision (Def. 5) in the definition of languages: if α is not precise, then $\langle\langle \Sigma[\alpha] \rangle\rangle^\gamma$ may mix interactions from different substructures of γ, that are $\mathcal{D}$-models of α, which clutters the meaning of these symbols in a regular expression. Note that, since interaction atoms are always precise on Γ, only predicate atoms may raise problems.

The proof rules infer judgements of the form $\eta \triangleright \{\{\phi\}\}\ \mathsf{L}\ \{\{\psi\}\}$, called *havoc triples*, where ϕ and ψ are CL formulæ, L is a regular expression, and η is an *environment* (a set of alphabet symbols), whose role will be made clear below (Def. 12 and Lemma 3). Intuitively, a havoc triple requires that each finite sequence of atomic state changes in L, when applied to a model of the precondition ϕ, yields a model of the postcondition ψ.

Definition 9 *A havoc triple* $\eta \triangleright \{\{\phi\}\}\ \mathsf{L}\ \{\{\psi\}\}$ *is* valid*, written* $\models_{\mathcal{D}} \eta \triangleright \{\{\phi\}\}\ \mathsf{L}\ \{\{\psi\}\}$*, if and only if, for each* $\gamma \in [\![\phi]\!]_{\mathcal{D}}$ *and each* $w \in \langle\langle \mathsf{L} \rangle\rangle^\gamma$*, we have* $\mathfrak{o}[w](\gamma) \subseteq [\![\psi]\!]_{\mathcal{D}}$*, where* $\mathfrak{o}[\varepsilon](\gamma) \stackrel{\text{def}}{=} \{\gamma\}$ *and* $\mathfrak{o}[w \cdot [I, \mathbf{u}]] \stackrel{\text{def}}{=} \mathfrak{o}[I, \mathbf{u}] \circ \mathfrak{o}[w]$*, for each sequence* w *of interactions.*

For a symbolic configuration ϕ, we denote by $\text{inter}(\phi)$ and $\text{preds}(\phi)$ the sets of interaction and predicate atoms from ϕ, respectively and let $\text{atoms}(\phi) \stackrel{\text{def}}{=} \text{inter}(\phi) \cup \text{preds}(\phi)$. We show that the validity of a havoc triple is a sufficient argument for the validity of a query $\mathfrak{h}([\![\phi]\!]_{\mathcal{D}}) \subseteq [\![\psi]\!]_{\mathcal{D}}$. Because havoc triples are evaluated via open state changes (Def. 9), the dual implication is not true, in general. Defining $\Sigma[\phi] \stackrel{\text{def}}{=} \bigcup_{\alpha \in \text{atoms}(\phi)} \Sigma[\alpha]$, we have:

Proposition 2 *If* $\models_{\mathcal{D}} \eta \triangleright \{\{\phi\}\}\ \Sigma[\phi]^*\ \{\{\psi\}\}$ *then* $\mathfrak{h}([\![\phi]\!]_{\mathcal{D}}) \subseteq [\![\psi]\!]_{\mathcal{D}}$.

5.2 Havoc Axioms and Inference Rules

We describe next a set of axioms and inference rules used to prove the validity of havoc triples. The side conditions of some of these rules use the following shorthands:

Definition 10 *For a symbolic configuration* ϕ *and an interaction atom* $I(x_1, \ldots, x_{\#(I)})$*, we write:*

- $\phi \dagger I(x_1, \ldots, x_{\#(I)})$ *if and only if* ϕ *contains a subformula* $C^q(y)$*, such that* $y \simeq_\phi x_i$ *and* q *is not the pre-state of some transition with label* $\langle ports(I) \rangle_i$ *in* $\mathcal{B}(C)$*, for some* $i \in [1, \#(I)]$*; intuitively, any interaction described by* $I(x_1, \ldots, x_{\#(I)})$ *must be disabled in any model of* ϕ*,*

- $\phi \ddagger I(x_1, \ldots, x_{\#(I)})$ *if and only if, for each* $I(y_1, \ldots, y_{\#(I)}) \in \mathrm{inter}(\phi)$*, there exists* $i \in [1, \#(I)]$*, such that* $x_i \not\simeq_\phi y_i$*; intuitively, an interaction described by* $I(x_1, \ldots, x_{\#(I)})$ *cannot be part of a model of* ϕ*.*

The axioms (Fig. 5a) introduce havoc triples for the empty sequence (ε), that changes nothing and the sequence consisting of a single interaction atom, that can be either disabled in every model ($\dagger$), or enabled in some model (Σ) of the precondition, respectively; in particular, the (Σ) axiom describes the open state change produced by an interaction (Def. 8), firing on a (possibly empty) set of components, whose states match the pre-states of transitions for the associated behaviors. The ($\bot$) axiom introduces trivially valid triples with unsatisfiable (false) preconditions.

The redundancy rule (I) in Fig. 5b adds an interaction atom to the precondition of a havoc triple, provided that the atom is never interpreted as an interaction from the language denoted by the regular expression from the triple, where $\mathrm{supp}(\mathsf{L})$ denotes the set of alphabet symbols of the form $\Sigma[\alpha]$ from a regular expression L. Conversely, the rule (E) removes an interaction from the precondition, provided that the precondition (with that interaction atom) is consistent[6].

The composition rule ($\bowtie$) splits a proof obligation into two simpler havoc triples (Fig. 5c). The pre- and postconditions of the premisses are subformulæ of the pre- and postcondition of the conclusion, joined by separating conjunction and extended by so-called *frontier* formulæ, describing those sets of interaction atoms that may cross the boundary between the two separated conjuncts.

Definition 11 (Frontier) *Given symbolic configurations* ϕ_1 *and* ϕ_2*, the* frontier of ϕ_i and ϕ_{3-i} *is the formula* $\mathcal{F}(\phi_i, \phi_{3-i}) \stackrel{\mathsf{def}}{=} \ast_{\alpha \in \mathrm{inter}(\phi_{3-i}) \setminus (\mathrm{inter}(\overline{\phi}_{3-i}) \cup \mathrm{inter}(\phi_i))} \alpha$*, where* $\overline{\phi}_i$ *is the largest tight subformula of* ϕ_i*, for* $i = 1, 2$*.*

As a remark, the largest tight subformula (Def. 6) of a symbolic configuration can be effectively computed using the result of Prop. 6. The frontier formulæ play the role of assumptions in a rely/assume-guarantee style of reasoning [**?**, **?**]. They are required for the soundness of the ($\bowtie$) rule, under the semantics of open state changes (Def. 8), which considers that the interactions from $\mathcal{F}(\phi_i, \phi_{3-i})$ can fire anytime, unless they are explicitly disabled by some component from ϕ_i, for $i = 1, 2$. Moreover, provided that the predicate atoms are tight (Assumption 1), no interaction arising from an unfolding of a predicate atom in ϕ_i, can impact a component from ϕ_{3-i}, hence it is sound to consider only the finite set of interactions $\mathcal{F}(\phi_i, \phi_{3-i})$. The regular expression of the conclusion is the interleaving of the regular expressions from the premisses, with respect to the environments η_i, which are the sets of predicate and interaction atoms from both the precondition ϕ_i and the frontier $\mathcal{F}(\phi_i, \phi_{3-i})$, for $i = 1, 2$.

[6] Without the $\phi \ddagger \alpha$ side condition, we would obtain a trivial proof for any triple, by adding an interaction atom twice to the precondition, i.e. using the rule (E), followed by ($\bot$).

Figure 5: Proof System for Havoc Triples

$$(\varepsilon)\ \frac{}{\eta \triangleright \{\{\phi\}\}\ \varepsilon\ \{\{\phi\}\}} \qquad (\dagger)\ \frac{}{\eta \triangleright \{\{\phi\}\}\ \Sigma[\alpha]\ \{\{\mathsf{false}\}\}}\ \begin{array}{l}\alpha{=}I(x_1,\ldots,x_{\#(I)})\\ \phi\dagger\alpha\end{array}$$

$$(\bot)\ \frac{}{\eta \triangleright \{\{\mathsf{false}\}\}\ \mathsf{L}\ \{\{\psi\}\}}$$

$$(\Sigma)\ \frac{}{\eta \triangleright \{\{\alpha * \divideontimes_{j=1}^{k} C_j^{q_j}(x_{i_j})\}\}\ \Sigma[\alpha]\ \{\{\alpha * \divideontimes_{j=1}^{k} \bigvee_{q_j \xrightarrow{\langle ports(I)\rangle_{i_j}}_j r_j} C_j^{r_j}(x_{i_j})\}\}}\ \begin{array}{l}\alpha{=}I(x_1,\ldots,x_{\#(I)})\\ i_1,\ldots,i_k \in [1,\#(I)],\ k{\geq}0\\ \mathcal{B}(C_j){=}\langle Q_j, P_j, q_j^0, \rightarrow_j\rangle\end{array}$$

a. Axioms

$$(\mathsf{I})\ \frac{\eta \setminus \{\Sigma[\alpha]\} \triangleright \{\{\phi\}\}\ \mathsf{L}\ \{\{\psi\}\}}{\eta \triangleright \{\{\phi * \alpha\}\}\ \mathsf{L}\ \{\{\psi * \alpha\}\}}\ \begin{array}{l}\alpha{=}I(x_1,\ldots,x_{\#(I)})\\ \Sigma[\alpha]{\in}\eta\setminus\mathrm{supp}(\mathsf{L})\end{array} \qquad (\mathsf{E})\ \frac{\eta \cup \{\Sigma[\alpha]\} \triangleright \{\{\phi * \alpha\}\}\ \mathsf{L}\ \{\{\psi * \alpha\}\}}{\eta \triangleright \{\{\phi\}\}\ \mathsf{L}\ \{\{\psi\}\}}\ \begin{array}{l}\alpha{=}I(x_1,\ldots,x_{\#(I)})\\ \phi\ddagger\alpha\end{array}$$

b. Redundancy Rules

$$(\bowtie)\ \frac{\eta_i \triangleright \{\{\phi_i * \mathcal{F}(\phi_i, \phi_{3-i})\}\}\ \mathsf{L}_i\ \{\{\psi_i * \mathcal{F}(\phi_i, \phi_{3-i})\}\} \mid i = 1,2}{\eta_1 \cup \eta_2 \triangleright \{\{\phi_1 * \phi_2\}\}\ \mathsf{L}_1 \bowtie_{\eta_1,\eta_2} \mathsf{L}_2\ \{\{\psi_1 * \psi_2\}\}}\ \eta_i{=}\Sigma[\phi_i * \mathcal{F}(\phi_i,\phi_{3-i})],\ i{=}1,2$$

c. Composition Rule

$$(\cdot)\ \frac{\eta \triangleright \{\{\bigvee_{i=1}^{k} \phi \wedge \delta_i\}\}\ \mathsf{L}_1\ \{\{\bigvee_{i=1}^{k} \phi \wedge \delta_i'\}\} \qquad \eta \triangleright \{\{\bigvee_{i=1}^{k} \phi \wedge \delta_i'\}\}\ \mathsf{L}_2\ \{\{\psi\}\}}{\eta \triangleright \{\{\bigvee_{i=1}^{k} \phi \wedge \delta_i\}\}\ \mathsf{L}_1 \cdot \mathsf{L}_2\ \{\{\psi\}\}}\ \begin{array}{l}\delta_i,\delta_i',\ i{\in}[1,k]\\ \text{state atoms}\end{array}$$

$$(*)\ \frac{\eta \triangleright \{\{\phi\}\}\ \mathsf{L}\ \{\{\phi\}\}}{\eta \triangleright \{\{\phi\}\}\ \mathsf{L}^*\ \{\{\phi\}\}}$$

$$(\cup)\ \frac{\eta \triangleright \{\{\phi\}\}\ \mathsf{L}_1\ \{\{\psi\}\} \qquad \eta \triangleright \{\{\phi\}\}\ \mathsf{L}_2\ \{\{\psi\}\}}{\eta \triangleright \{\{\phi\}\}\ \mathsf{L}_1 \cup \mathsf{L}_2\ \{\{\psi\}\}} \qquad (\subset)\ \frac{\eta \triangleright \{\{\phi\}\}\ \mathsf{L}_1 \cup \mathsf{L}_2\ \{\{\psi\}\}}{\eta \triangleright \{\{\phi\}\}\ \mathsf{L}_1\ \{\{\psi\}\}}$$

d. Regular Expression Rules

$$(\mathsf{C})\ \frac{\eta \triangleright \{\{\phi\}\}\ \mathsf{L}\ \{\{\psi'\}\}}{\eta \triangleright \{\{\phi\}\}\ \mathsf{L}\ \{\{\psi\}\}}\ \psi' \models_{\mathcal{D}} \psi \qquad (\mathsf{LU})\ \frac{\eta' \triangleright \{\{\phi * \varphi\}\}\ \mathsf{L}'\ \{\{\psi\}\}}{\eta \triangleright \{\{\phi * \mathsf{A}(x_1,\ldots,x_{\#(\mathsf{A})})\}\}\ \mathsf{L}\ \{\{\psi\}\}}\ \left|\begin{array}{l}\mathsf{A}(x_1,\ldots,x_{\#(\mathsf{A})}) \Leftarrow_{\mathcal{D}} \exists \mathbf{z}\ .\ \varphi,\ \varphi{\in}\mathbb{S},\ \mathbf{z} \cap \mathrm{fv}(\phi){=}\emptyset\\ \eta' = \big(\eta \setminus \{\Sigma[\mathsf{A}(x_1,\ldots,x_{\#(\mathsf{A})})]\}\big) \cup \Sigma[\varphi]\\ \mathsf{L}' = \mathsf{L}\big[\Sigma[\mathsf{A}(x_1,\ldots,x_{\#(\mathsf{A})})] \leftarrow \big(\cup\Sigma[\varphi]\big)\big]\end{array}\right.$$

$$(\vee)\ \frac{\eta \triangleright \{\{\phi \wedge \delta_i\}\}\ \mathsf{L}\ \{\{\psi_i\}\} \mid i \in [1,k]}{\eta \triangleright \{\{\bigvee_{i=1}^{k} \phi \wedge \delta_i\}\}\ \mathsf{L}\ \{\{\bigvee_{i=1}^{k} \psi_i\}\}}\ \begin{array}{l}\delta_i,\ i{\in}[1,k]\\ \text{state atoms}\end{array} \qquad (\wedge)\ \frac{\eta \triangleright \{\{\phi \wedge \delta_i\}\}\ \mathsf{L}\ \{\{\psi_i\}\} \mid i \in [1,k]}{\eta \triangleright \{\{\bigwedge_{i=1}^{k} \phi \wedge \delta_i\}\}\ \mathsf{L}\ \{\{\bigwedge_{i=1}^{k} \psi_i\}\}}\ \begin{array}{l}\delta_i,\ i{\in}[1,k]\\ \text{state atoms}\end{array}$$

e. Structural Rules

The rules in Fig. 5d introduce regular expressions built using concatenation, Kleene star and union. In particular, for reasons related to the soundness of the proof system, the concatenation rule $(\cdot)$ applies to havoc triples whose preconditions are finite disjunctions of symbolic configurations, sharing the same set of component, interaction and predicate atoms and different conjunctions of state atoms, whereas the cut formulæ (postcondition of the left and precondition of the right premiss) share the same structure as the precondition. The $(\subset)$ rule is the dual of $(\cup)$, that restricts the language from the conclusion to a subset of the one from the premiss. As a remark, by applying the (E) and $(\subset)$ rules in any order, one can derive the havoc invariance of the intermediate assertions in a single-reversal reconfiguration sequence (Remark 1).

Finally, the rules in Fig. 5e modify the structure of the pre- and postconditions.

In particular, the left unfolding rule (LU) has a premiss for each unfolding step of a predicate atom from the conclusion's precondition. The environment and the regular expression in each premiss are obtained by replacing the alphabet symbol of the unfolded predicate symbol by the set of alphabet symbols from the unfolding step, where $\mathsf{L}[\Sigma[\alpha] \leftarrow \mathsf{L}']$ denotes the regular expression obtained by replacing each occurrence of the alphabet symbol $\Sigma[\alpha]$ in L with the regular expression L'.

5.3 Havoc Proofs

A *proof tree* is a finite tree T whose nodes are labeled by havoc triples and, for each node n not on the frontier of T, the children of n are the premisses of the application of a rule from Fig. 5, whose conclusion is the label of n. For the purposes of this paper, we consider only proof trees that meet the following condition:

Assumption 2 *The root of the proof tree is labeled by a havoc triple* $\eta \triangleright \{\{\phi\}\} \; \mathsf{L} \; \{\{\psi\}\}$, *such that* ϕ *is a symbolic configuration and* $\eta = \{\Sigma[\alpha] \mid \alpha \in \mathrm{atoms}(\phi)\}$.

It is easy to check that the above condition on the shape of the precondition and the relation between the precondition and the environment holds recursively, for the labels of all nodes in a proof tree that meets assumption 2. Moreover, havoc triples of the form $\Sigma[\phi] \triangleright \{\{\bigvee_{i=1}^{k} \phi \wedge \delta_i\}\} \; \mathsf{L} \; \{\{\psi\}\}$, where ϕ is a symbolic configuration and each δ_i is a conjunction of state atoms, can be handled as well, by proving each triple $\Sigma[\phi] \triangleright \{\{\phi \wedge \delta_i\}\} \; \mathsf{L} \; \{\{\psi\}\}$ individually. Before tackling the soundness of the havoc proof system (Fig. 5), we state an invariance property of the environments of havoc triples that occur in a proof tree:

Definition 12 *A havoc triple* $\eta \triangleright \{\{\phi\}\} \; \mathsf{L} \; \{\{\psi\}\}$ *is* distinctive *if and only if* $\langle\langle \Sigma[\alpha_1] \rangle\rangle^{\gamma} \cap \langle\langle \Sigma[\alpha_2] \rangle\rangle^{\gamma} = \emptyset$, *for all* $\Sigma[\alpha_1], \Sigma[\alpha_2] \in \eta$ *and all* $\gamma \in [\![\phi]\!]_{\mathcal{D}}$.

The next lemma is proved inductively on the structure of the proof tree, using Assumption 2.

Lemma 3 *Given a proof tree T, each node in T is labeled with a distinctive havoc triple.*

In order to deal with inductively defined predicates that occur within the pre- and postconditions of the havoc triples, we use cyclic proofs [**?**]. A *cyclic proof tree T* is a proof tree such that every node on the frontier is either the conclusion of an axiom in Fig. 5a, or there is another node m whose label matches to the label of n via a substitution of variables; we say that n is a *bud* and m is its *companion*. A cyclic proof tree is a *cyclic proof* if and only if every infinite path through the proof tree extended with bud-companion edges, goes through the conclusion of a (LU) rule infinitely often[7]. We denote by $\Vdash \eta \triangleright \{\{\phi\}\} \; \mathsf{L} \; \{\{\psi\}\}$ the fact that $\eta \triangleright \{\{\phi\}\} \; \mathsf{L} \; \{\{\psi\}\}$ labels the root of a cyclic proof and state the following soundness theorem:

Theorem 2 *If* $\Vdash \eta \triangleright \{\{\phi\}\} \; \mathsf{L} \; \{\{\psi\}\}$ *then* $\models_{\mathcal{D}} \eta \triangleright \{\{\phi\}\} \; \mathsf{L} \; \{\{\psi\}\}$.

[7] This condition can be effectively decided by checking the emptiness of a Büchi automaton [**?**].

The proof is by induction on the structure of the proof tree, using Lemma 3 for (I) rules. We conclude the presentation of the havoc proof system with a remark concerning the equivalence between regular expressions, needed to apply the rules in Fig. 5c-d. Given a symbolic configuration ϕ, two regular expressions are *congruent*, denoted $\mathsf{L}_1 \cong_\phi \mathsf{L}_2$, if and only if $\langle\langle \mathsf{L}_1 \rangle\rangle^\gamma = \langle\langle \mathsf{L}_2 \rangle\rangle^\gamma$, for all configurations $\gamma \in [\![\phi]\!]_{\mathcal{D}}$. Despite the universal condition that ranges over a possibly infinite set of configurations, congruence of regular expressions with alphabet symbols of the form $\Sigma[\alpha]$, where α is an interaction or a predicate atom, is decidable by an argument similar to the one used to prove equivalence of symbolic automata [**?**]. For space reasons, we only sketch the justification of this point below:

Remark 2 *We build finite automata that recognize the regular languages $\mathcal{L}(\mathsf{L}_i)$, $i = 1,2$ with the alphabet symbols $\Sigma[\alpha]$ taken as such and check the equivalence $\mathcal{L}(\mathsf{L}_1) = \mathcal{L}(\mathsf{L}_2)$ using these automata. Since the alphabet symbols are interpreted as disjoint sets in every model of the precondition of a distinctive havoc triple, we can assume w.l.o.g. that the sets $\langle\langle \Sigma[\alpha] \rangle\rangle^\gamma$ are pairwise disjoint, in each configuration $\gamma \in [\![\phi]\!]_{\mathcal{D}}$, and define a language morphism F, that maps each element of $\langle\langle \Sigma[\alpha] \rangle\rangle^\gamma$ onto the symbol $\Sigma[\alpha]$. Then $F(\langle\langle \mathsf{L}_1 \rangle\rangle^\gamma) = F(\langle\langle \mathsf{L}_2 \rangle\rangle^\gamma)$ if and only if $\mathcal{L}(\mathsf{L}_1) = \mathcal{L}(\mathsf{L}_2)$, where the choice of $\gamma \in [\![\phi]\!]_{\mathcal{D}}$ is not important, provided that the sets $\langle\langle \Sigma[\alpha] \rangle\rangle^\gamma$ are pairwise disjoint.*

5.4 A Havoc Proof Example

We demonstrate the use of the havoc proof system (Fig. 5) for the havoc invariance side conditions from Example 7. In fact, we prove a more general statement, namely that $\mathsf{chain}_{h,t}(x,y)$ is havoc invariant, i.e. that

$$\{\mathsf{chain}_{h,t}(x,y)\} \rhd \{\{\mathsf{chain}_{h,t}(x,y)\}\}\ \Sigma[\mathsf{chain}_{h,t}(x,y)]^*\ \{\{\mathsf{chain}_{h,t}(x,y)\}\}$$

is valid, for all $h,t \geq 0$. An immediate consequence is that $\mathsf{chain}_{1,1}(z,x)$ is havoc invariant. The havoc invariance proof for $\mathsf{S}^{\mathbf{n}}(y) * \mathsf{T}(y,z) * \mathsf{chain}_{1,1}(z,x)$ is an instance of the subgoal (**A**) below, whereas the proof for $\mathsf{S}^{\mathbf{n}}(y) * \mathsf{chain}_{1,1}(z,x)$ can be obtained by applying rules (E) and ($\subset$) to (**A**), for $h = t = 1$.

For space reasons, we introduce backlinks from buds to companions whose labels differ by a renaming of free variables and of the h and t indices in $\mathsf{chain}_{h,t}$, such that each pair (h',t') in the label of a companion is lexicographically smaller or equal to a pair (h,t) in the bud. This is a compact (folded) representation of a proof tree, obtained by repeatedly appending the subtree rooted at the companion to the bud, until all buds are labeled with triples that differ from their companion's only by a renaming of free variables[8]. Note that such folding is only possible because the definitions of $\mathsf{chain}_{h,t}(x,y)$ and $\mathsf{chain}_{h',t'}(x,y)$, for $h,t,h',t' \geq 1$ are the same, up to the indices of the predicate symbols (Example 4).

$$\dfrac{\dfrac{}{\emptyset \rhd \{\{\mathsf{S}(x)\}\} \in \{\{\mathsf{S}(x)\}\}}\ (\varepsilon) \quad \dfrac{}{\emptyset \rhd \{\{\mathsf{S}^{\mathbf{n}}(x)\}\} \in \{\{\mathsf{S}^{\mathbf{n}}(x)\}\}}\ (\varepsilon) \quad \dfrac{}{\emptyset \rhd \{\{\mathsf{S}^{\mathbf{t}}(x)\}\} \in \{\{\mathsf{S}^{\mathbf{t}}(x)\}\}}\ (\varepsilon) \quad \textbf{(A)} \quad \textbf{(B)}}{\{\Sigma[\mathsf{chain}_{h,t}(z,x)]\} \rhd \{\{\mathsf{chain}_{h,t}(z,x)\}\}\ \Sigma[\mathsf{chain}_{h,t}(z,x)]^*\ \{\{\mathsf{chain}_{h,t}(z,x)\}\}}\ (\mathrm{LU})\ \textbf{(1)}$$

[8]This is bound to happen, because a pair (h,t) of positive integers cannot be decreased indefinitely.

In the proof of the subgoal **(A)** below, alphabet symbols are abbreviated as $\Sigma_{z,y} \stackrel{\mathsf{def}}{=} \Sigma[\mathsf{T}(z,y)]$ and $\Sigma^1_{y,x} \stackrel{\mathsf{def}}{=} \Sigma[\mathsf{chain}_{h\dot{-}1,t}(y,x)]$. The rule $(\cup)$ uses the congruence $(\Sigma_{z,y} \cup \Sigma^1_{y,x})^* \cong_{\mathsf{S^n}(z)*\mathsf{T}(z,y)*\mathsf{chain}_{h\dot{-}1,t}(y,x)} \Sigma^1_{y,x}{}^* \cup [\Sigma^1_{y,x}{}^* \cdot \Sigma_{z,y} \cdot (\Sigma_{z,y} \cup \Sigma^1_{y,x})^*]$, which can be checked using finite automata (Remark 2). The rule (C) strenghtens the postcondition $\mathsf{chain}_{h,t}(z,x)$ to an unfolding $\mathsf{chain}_{h,t}(z,x) \Leftarrow_{\mathcal{D}} \exists y \,.\, \mathsf{S^n}(z) * \mathsf{T}(z,y) * \mathsf{chain}_{h\dot{-}1,t}(y,x)$, whose existentially quantified variable is, moreover, bound to the free variable y from the precondition. The frontier formulæ in the application of rule $(\bowtie)$ are just emp.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\emptyset \rhd \{\{\mathsf{S^n}(z)\}\} \; \varepsilon \; \{\{\mathsf{S^n}(z)\}\}}\,(\varepsilon) \quad \dfrac{}{\{\Sigma^1_{y,x}\} \rhd \{\{\mathsf{chain}_{h\dot{-}1,t}(y,x)\}\} \; \Sigma^1_{y,x}{}^* \; \{\{\mathsf{chain}_{h\dot{-}1,t}(y,x)\}\}}\,\text{backlink to (1)}}{\{\Sigma^1_{y,x}\} \rhd \{\{\mathsf{S^n}(z) * \mathsf{chain}_{h\dot{-}1,t}(y,x)\}\} \; \Sigma^1_{y,x}{}^* \; \{\{\mathsf{S^n}(z) * \mathsf{chain}_{h\dot{-}1,t}(y,x)\}\}}\,(\bowtie)}{\textbf{(A1)} \; \{\Sigma_{z,y}, \Sigma^1_{y,x}\} \rhd \{\{\mathsf{S^n}(z) * \mathsf{T}(z,y) * \mathsf{chain}_{h\dot{-}1,t}(y,x)\}\} \; \Sigma^1_{y,x}{}^* \; \{\{\mathsf{S^n}(z) * \mathsf{T}(z,y) * \mathsf{chain}_{h\dot{-}1,t}(y,x)\}\}}\,(\mathsf{I})}{\{\Sigma_{z,y}, \Sigma^1_{y,x}\} \rhd \{\{\mathsf{S^n}(z) * \mathsf{T}(z,y) * \mathsf{chain}_{h\dot{-}1,t}(y,x)\}\} \; \Sigma^1_{y,x}{}^* \; \{\{\mathsf{chain}_{h,t}(z,x)\}\}}\,(\mathsf{C}) \qquad \textbf{(A2)}}{\textbf{(A)} \; \{\Sigma_{z,y}, \Sigma^1_{y,x}\} \rhd \{\{\mathsf{S^n}(z) * \mathsf{T}(z,y) * \mathsf{chain}_{h\dot{-}1,t}(y,x)\}\} \; (\Sigma_{z,y} \cup \Sigma^1_{y,x})^* \; \{\{\mathsf{chain}_{h,t}(z,x)\}\}}\,(\cup)$$

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\{\Sigma_{z,y}, \Sigma^1_{y,x}\} \rhd \{\{\mathsf{S^n}(z) * \mathsf{T}(z,y) * \mathsf{chain}_{h\dot{-}1,t}(y,x)\}\}}\,\text{backlink to (A1)} \quad \dfrac{}{\{\Sigma_{z,y}, \Sigma^1_{y,x}\} \rhd \{\{\mathsf{S^n}(z) * \mathsf{T}(z,y) * \mathsf{chain}_{h\dot{-}1,t}(y,x)\}\}}\,(\dagger) \quad \dfrac{}{\{\Sigma_{z,y}, \Sigma^1_{y,x}\} \rhd \{\{\mathsf{false}\}\}}\,(\bot)}{\Sigma^1_{y,x}{}^* \; \{\{\mathsf{S^n}(z) * \mathsf{T}(z,y) * \mathsf{chain}_{h\dot{-}1,t}(y,x)\}\} \qquad \Sigma_{z,y} \; \{\{\mathsf{false}\}\} \qquad (\Sigma_{z,y} \cup \Sigma^1_{y,x})^* \; \{\{\mathsf{false}\}\}}\,(\cdot)}{\{\Sigma_{z,y}, \Sigma^1_{y,x}\} \rhd \{\{\mathsf{S^n}(z) * \mathsf{T}(z,y) * \mathsf{chain}_{h\dot{-}1,t}(y,x)\}\} \; \Sigma^1_{y,x}{}^* \cdot \Sigma_{z,y} \cdot (\Sigma_{z,y} \cup \Sigma^1_{y,x})^* \; \{\{\mathsf{false}\}\}}\,(\mathsf{C})}{\textbf{(A2)} \; \{\Sigma_{z,y}, \Sigma^1_{y,x}\} \rhd \{\{\mathsf{S^n}(z) * \mathsf{T}(z,y) * \mathsf{chain}_{h\dot{-}1,t}(y,x)\}\} \; \Sigma^1_{y,x}{}^* \cdot \Sigma_{z,y} \cdot (\Sigma_{z,y} \cup \Sigma^1_{y,x})^* \; \{\{\mathsf{chain}_{h,t}(z,x)\}\}}$$

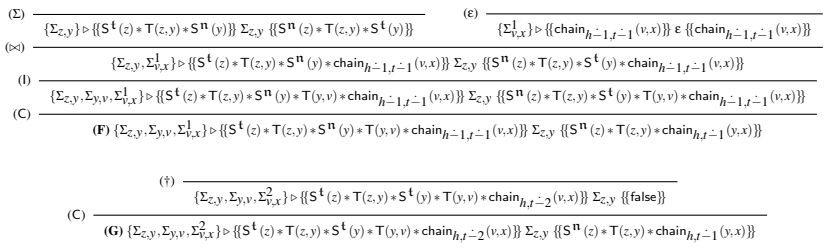
In the proof of the subgoal **(B)**, we use the shorthand $\Sigma^2_{y,x} \stackrel{\mathsf{def}}{=} \Sigma[\mathsf{chain}_{h,t\dot{-}1}(y,x)]$ and the congruence $(\Sigma_{z,y} \cup \Sigma^2_{y,x})^* \cong_{\mathsf{S^t}(z)*\mathsf{T}(z,y)*\mathsf{chain}_{h,t\dot{-}1}(y,x)} \Sigma^2_{y,x}{}^* \cup [\Sigma^2_{y,x}{}^* \cdot \Sigma_{z,y} \cdot (\Sigma_{z,y} \cup \Sigma^2_{y,x})^*]$.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\textbf{(B1)} \; \{\Sigma_{z,y}, \Sigma^2_{y,x}\} \rhd \{\{\mathsf{S^t}(z) * \mathsf{T}(z,y) * \mathsf{chain}_{h,t\dot{-}1}(y,x)\}\} \; \Sigma^2_{y,x}{}^* \; \{\{\mathsf{S^t}(z) * \mathsf{T}(z,y) * \mathsf{chain}_{h,t\dot{-}1}(y,x)\}\}}\,\text{similar to (A1)}}{\{\Sigma_{z,y}, \Sigma^2_{y,x}\} \rhd \{\{\mathsf{S^t}(z) * \mathsf{T}(z,y) * \mathsf{chain}_{h,t\dot{-}1}(y,x)\}\} \; \Sigma^2_{y,x}{}^* \; \{\{\mathsf{chain}_{h,t}(z,x)\}\}}\,(\mathsf{C}) \qquad \textbf{(B2)}}{\textbf{(B)} \; \{\Sigma_{z,y}, \Sigma^2_{y,x}\} \rhd \{\{\mathsf{S^t}(z) * \mathsf{T}(z,y) * \mathsf{chain}_{h,t\dot{-}1}(y,x)\}\} \; (\Sigma_{z,y} \cup \Sigma^2_{y,x})^* \; \{\{\mathsf{chain}_{h,t}(z,x)\}\}}\,(\cup)$$

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\{\Sigma_{z,y}, \Sigma^2_{y,x}\} \rhd \{\{\mathsf{S^t}(z) * \mathsf{T}(z,y) * \mathsf{chain}_{h,t\dot{-}1}(y,x)\}\}}\,\text{backlink to (B1)} \quad \dfrac{}{\{\Sigma_{z,y}, \Sigma^2_{y,x}\} \rhd \{\{\mathsf{S^t}(z) * \mathsf{T}(z,y) * \mathsf{chain}_{h,t\dot{-}1}(y,x)\}\}}\,\textbf{(C)} \quad \dfrac{}{\{\Sigma_{z,y}, \Sigma^2_{y,x}\} \rhd \{\{\mathsf{S^n}(z) * \mathsf{T}(z,y) * \mathsf{chain}_{h,t\dot{-}1}(y,x)\}\}}\,\text{similar to (A)}}{\Sigma^2_{y,x}{}^* \; \{\{\mathsf{S^t}(z) * \mathsf{T}(z,y) * \mathsf{chain}_{h,t\dot{-}1}(y,x)\}\} \qquad \Sigma_{z,y} \; \{\{\mathsf{S^n}(z) * \mathsf{T}(z,y) * \mathsf{chain}_{h,t\dot{-}1}(y,x)\}\} \qquad (\Sigma_{z,y} \cup \Sigma^2_{y,x})^* \; \{\{\mathsf{chain}_{h,t}(z,x)\}\}}\,(\cdot)}{\textbf{(B2)} \; \{\Sigma_{z,y}, \Sigma^2_{y,x}\} \rhd \{\{\mathsf{S^t}(z) * \mathsf{T}(z,y) * \mathsf{chain}_{h,t\dot{-}1}(y,x)\}\} \; \Sigma^2_{y,x}{}^* \cdot \Sigma_{z,y} \cdot (\Sigma_{z,y} \cup \Sigma^2_{y,x})^* \; \{\{\mathsf{chain}_{h,t}(z,x)\}\}}$$

The proof of **(B1)** is similar to the proof of **(A1)**, with $\Sigma^1_{y,x}$ and $\mathsf{chain}_{h\dot{-}1,t}(y,x)$ replaced by $\Sigma^2_{y,x}$ and $\mathsf{chain}_{h,t\dot{-}1}(y,x)$, respectively. The proof of the right-most subtree is similar to the proof of **(A)**, with $\mathsf{chain}_{h\dot{-}1,t}(y,x)$ replaced by $\mathsf{chain}_{h,t\dot{-}1}(y,x)$. In the proof of **(C)** below, we use the shorthands $\Sigma_{y,v} \stackrel{\mathsf{def}}{=} \Sigma[\mathsf{T}(y,v)]$, $\Sigma^1_{v,x} \stackrel{\mathsf{def}}{=} \Sigma[\mathsf{chain}_{h\dot{-}1,t\dot{-}1}(v,x)]$ and $\Sigma^2_{v,x} \stackrel{\mathsf{def}}{=} \Sigma[\mathsf{chain}_{h,t\dot{-}2}(v,x)]$. The frontier formulæ are empty for the application of rule $(\bowtie)$ in the proof of the subgoal **(G)** because both $\mathsf{S^t}(z) * \mathsf{T}(z,y) * \mathsf{S^n}(y)$ and $\mathsf{chain}_{h\dot{-}1,t\dot{-}1}(v,x)$ are provably tight formulæ (Prop. 6).

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\{\Sigma_{z,y}\} \rhd \{\{\mathsf{S^t}(z) * \mathsf{T}(z,y) * \mathsf{S^n}(x)\}\}}\,(\Sigma) \quad \dfrac{}{\{\Sigma_{z,y}\} \rhd \{\{\mathsf{S^t}(z) * \mathsf{T}(z,y) * \mathsf{S^t}(x)\}\}}\,(\dagger)}{\{\Sigma_{z,y}\} \rhd \{\{\mathsf{S^t}(z) * \mathsf{T}(z,y) * \mathsf{S}(x)\}\}}\,(\vee)\;\text{backlink to (D)}\;\text{backlink to (E)} \quad \dfrac{\Sigma_{z,y} \; \{\{\mathsf{S^n}(z) * \mathsf{T}(z,y) * \mathsf{S^t}(x)\}\}}{\textbf{(D)} \; \{\Sigma_{z,y}\} \rhd \{\{\mathsf{S^t}(z) * \mathsf{T}(z,y) * \mathsf{S^n}(x)\}\}}\,(\mathsf{C}) \quad \dfrac{\Sigma_{z,y} \; \{\{\mathsf{false}\}\}}{\textbf{(E)} \; \{\Sigma_{z,y}\} \rhd \{\{\mathsf{S^t}(z) * \mathsf{T}(z,y) * \mathsf{S^t}(x)\}\}}\,(\mathsf{C})}{\Sigma_{z,y} \; \{\{\mathsf{S^n}(z) * \mathsf{T}(z,y) * \mathsf{chain}_{h,t\dot{-}1}(y,x)\}\} \qquad \Sigma_{z,y} \; \{\{\mathsf{S^n}(z) * \mathsf{T}(z,y) * \mathsf{chain}_{h,t\dot{-}1}(y,x)\}\} \qquad \Sigma_{z,y} \; \{\{\mathsf{S^n}(z) * \mathsf{T}(z,y) * \mathsf{chain}_{h,t\dot{-}1}(y,x)\}\} \quad \textbf{(F)} \quad \textbf{(G)}}\,(\mathsf{LU})$$
$$\textbf{(C)} \; \{\Sigma_{z,y}, \Sigma^2_{y,x}\} \rhd \{\{\mathsf{S^t}(z) * \mathsf{T}(z,y) * \mathsf{chain}_{h,t\dot{-}1}(y,x)\}\} \; \Sigma_{z,y} \; \{\{\mathsf{S^n}(z) * \mathsf{T}(z,y) * \mathsf{chain}_{h,t\dot{-}1}(y,x)\}\}$$



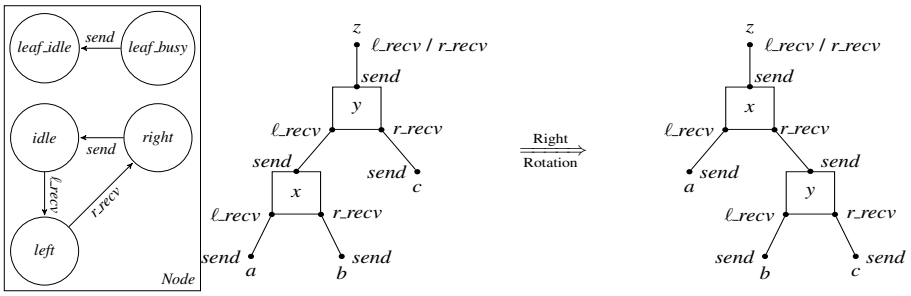
6 A Worked-out Example: Reconfigurable Trees

In addition to token rings (Fig. 1), we apply our method to reconfiguration scenarios of distributed systems with tree-shaped architectures. Such (virtual) architectures are e.g. used in flooding and leader election algorithms. They are applicable, for instance, when every component in the system must notify a designated controller, placed in the root of the tree, about an event that involves each component from the frontier of the tree. Conversely, the root component may need to notify the rest of the components. The tree architecture guarantees that the notification phase takes time $O(\log n)$ in the number n of components in the tree, when the tree is balanced, i.e. the lengths of the longest and shortest paths between the root and the frontier differ by at most a constant factor. A reconfiguration of a tree places a designated component (whose priority has increased dynamically) closer to the frontier (dually, closer to the root) in order to receive the notification faster. In balanced trees, reconfigurations involve structure-preserving rotations. For instance, *self-adjustable splay-tree networks* [**?**] use the *zig* (left rotation), *zig-zig* (left-left rotation) and *zig-zag* (left-right rotation) operations [**?**] to move nodes in the tree, while keeping the balance between the shortest and longest paths.

Fig. 6 shows a model of reconfigurable tree architectures, in which each leaf component sends a notification to its parent before entering the *leaf_idle* state. An inner component waits for notifications from both its left (ℓ*_recv*) and right (*r_recv*) children before sending a notification to its parent (*send*), unless this component is the root (Fig. 6a). We model notifications by the interaction types I_ℓ and I_r, with $ports(I_\ell) = \langle send, \ell\_recv \rangle$ and $ports(I_r) = \langle send, r\_recv \rangle$. The notification phase is completed when the root is in state *right*, every inner component is in the *idle* state and every leaf is in the *leaf_idle* state. Fig. 6b shows a right rotation that reverses the positions of components with identifiers x and y, implemented by the reconfiguration program from Fig. 7. The rotation applies only to configurations in which both x and y are in state *idle*, by distinguishing the case when y is a left or a right child of z. For simplicity, Fig. 7 shows the program in case y is a left child, the other case being symmetric.

Note that, applying the rotation in a configuration where the *Node* component indexed by x is in state *right* (both a and b have sent their notifications to x) and the one indexed by y is in state *idle* (c has not yet sent its notification to y) yields a configuration from which c cannot send its notification further, because x has now become the root of the subtree changed by the rotation (a similar scenario is when y is in state *right*, x

Figure 6: Reconfiguration of a Tree Architecture



(a) (b)

$$\mathsf{tree}_{idle}(x) \leftarrow Node^{leaf\_idle}(x)$$
$$\mathsf{tree}_{idle}(x) \leftarrow \exists y \exists z \,.\, Node^{idle}(x) * I_\ell(y,x) * I_r(z,x) * \mathsf{tree}_{idle}(y) * \mathsf{tree}_{idle}(z)$$
$$\mathsf{tree}_{\neg idle}(x) \leftarrow Node^{leaf\_busy}(x)$$
$$\mathsf{tree}_{\neg idle}(x) \leftarrow \exists y \exists z \,.\, Node^{left}(x) * I_\ell(y,x) * I_r(z,x) * \mathsf{tree}_{idle}(y) * \mathsf{tree}_{\neg idle}(z)$$
$$\mathsf{tree}_{\neg idle}(x) \leftarrow \exists y \exists z \,.\, Node^{right}(x) * I_\ell(y,x) * I_r(z,x) * \mathsf{tree}_{idle}(y) * \mathsf{tree}_{idle}(z)$$
$$\mathsf{tree}_{\neg idle}(x) \leftarrow \exists y \exists z \,.\, Node^{idle}(x) * I_\ell(y,x) * I_r(z,x) * \mathsf{tree}_{\neg idle}(y) * \mathsf{tree}_{\neg idle}(z)$$

$$\mathsf{tree}(x) \leftarrow Node^{leaf\_idle}(x)$$
$$\mathsf{tree}(x) \leftarrow Node^{leaf\_busy}(x)$$
$$\mathsf{tree}(x) \leftarrow \exists y \exists z \,.\, Node(x) * I_\ell(y,x) * I_r(z,x) * \mathsf{tree}(y) * \mathsf{tree}(z)$$
$$\mathsf{tseg}(x,x) \leftarrow Node(x)$$
$$\mathsf{tseg}(x,u) \leftarrow \exists y \exists z \,.\, Node(x) * I_\ell(y,x) * I_r(z,x) * \mathsf{tseg}(y,u) * \mathsf{tree}(z)$$
$$\mathsf{tseg}(x,u) \leftarrow \exists y \exists z \,.\, Node(x) * I_\ell(y,x) * I_r(z,x) * \mathsf{tree}(y) * \mathsf{tseg}(z,u)$$

(c)

is in state *idle* and a, b and c have sent their notifications to their parents). We prove that, whenever a right rotation is applied to a tree, such that the subtrees rooted at a, b and c have not sent their notifications yet, the result is another tree in which the subtrees rooted at a, b and c are still waiting to submit their notifications. This guarantees that the notification phase will terminate properly with every inner component (except for the root) in state *idle* and every leaf component in state *leaf_idle*, even if one or more reconfigurations take place in between. In particular, this proves the correctness of more complex reconfigurations of splay tree architectures, using e.g. the zig-zig and zig-zag operations [**?**].

The proof in Fig. 7 uses the inductive definitions from Fig. 6c. The predicates $\mathsf{tree}_{idle}(x)$, $\mathsf{tree}_{\neg idle}(x)$ define trees where all components are idle, and where some notification are still being propagated, respectively. The predicate $\mathsf{tree}(x)$ conveys no information about the states of the components and the predicate $\mathsf{tseg}(x,u)$ defines a tree segment, from component x to component u. To use the havoc proof system from Fig. 5, we need the following statement[9]:

Proposition 3 *The set of symbolic configurations using predicate atoms* $\mathsf{tree}_{idle}(x)$, $\mathsf{tree}_{\neg idle}(x)$, $\mathsf{tree}(x)$ *and* $\mathsf{tseg}(x,y)$ *is precisely closed.*

Moreover, each predicate atom $\mathsf{tree}_{idle}(x)$, $\mathsf{tree}_{\neg idle}(x)$, $\mathsf{tree}(x)$ and $\mathsf{tseg}(x,y)$ is tight, because, in each $\mathcal{D}$-model (σ, ν, ρ), the interactions $\langle u, v \rangle \in I_\ell^\sigma \cup I_r^\sigma$ are between

[9]This is similar to Prop. 1.

Figure 7: Proof of a Tree Rotation

the $ports(I_\ell) = \langle send, \ell\_recv \rangle$ and $ports(I_r) = \langle send, r\_recv \rangle$ of the components $u, v \in Node^{\sigma}$ in the structure. Together with Prop. 3, this shows that Assumption 1 loses no generality for the SID from Fig. 6c.

The precondition of the reconfiguration program in Fig. 7 states that x and y are idle components, and the a, b and c subtrees are not idle, whereas the postcondition states that the x subtree is not idle. As mentioned, this is sufficient to guarantee the correct termination of the notification phase after the right rotation. As in the proof from Example 7, proving the correctness of the sequential composition of primitive commands requires proving the havoc invariance of the annotations. However, since in this case, the reconfiguration sequence is single-reversal (Remark 1), we are left with proving havoc invariance[10] only for the second and second-last annotations, marked with (*hinv*) in Fig. 7.

[10]These proofs rely on the havoc invariance proofs for $\mathsf{tree}(x)$, $\mathsf{tree}_{idle}x$, $\mathsf{tree}_{\neg idle}x$ and $\mathsf{tseg}(x,u)$ given in Appendix F.

7 Entailment Problems

We describe a decision procedure for the entailment problem of SL, between symbolic configurations with predicate symbols defined by SIDs. Together with proving havoc invariants, deciding entailments is a key ingredient for mechanising the correctness proofs of reconfiguration programs (§4). We fix a SID $\mathcal{D}$, for the rest of this section. An instance of the *entailment problem* consists of SL formulæ ϕ and $\exists y_1 \ldots \exists y_r \,.\, \bigvee_{\ell=1}^{h} \psi_\ell$, where $\phi, \psi_1, \ldots, \psi_h$ are symbolic configurations, such that $\mathrm{fv}(\exists y_1 \ldots \exists y_r \,.\, \bigvee_{\ell=1}^{h} \psi_\ell) \subseteq \mathrm{fv}(\phi)$, and asks whether every $\mathcal{D}$-model of ϕ is an $\mathcal{D}$-model of $\exists y_1 \ldots \exists y_r \,.\, \psi_\ell$, for some $\ell \in [1,h]$.

Example 9 *The reconfiguration proof from Example 7 relies on the following entailments:*

$$\begin{array}{rcl} \mathsf{chain}_{2,1}(a,b) * \mathsf{T}(b,a) & \models_{\mathcal{D}} & \exists x \exists y \exists z \,.\, \mathsf{T}(x,y) * \mathsf{S}^{\mathsf{n}}(y) * \mathsf{T}(y,z) * \mathsf{chain}_{1,1}(z,x) \\ \mathsf{T}(x,y) * \mathsf{S}^{\mathsf{n}}(y) * \mathsf{T}(y,z) * \mathsf{chain}_{1,1}(z,x) & \models_{\mathcal{D}} & \exists a \exists b \,.\, \mathsf{chain}_{2,1}(a,b) * \mathsf{T}(b,a) \end{array}$$

■

We define a decidable class of entailment problems by two fairly natural conditions (Def. 13), typically met in our examples. These conditions rely on the following notion. The *profile* of a formula $\varphi = \exists x_1 \ldots \exists x_k \,.\, \phi$, where ϕ is a symbolic configuration, is the pointwise greatest function $\lambda_\varphi : \mathbb{A} \to 2^{\mathbb{N}}$ mapping each predicate symbol A onto a set of positions $\lambda_\varphi(\mathsf{A}) \subseteq [1, \#(\mathsf{A})]$, such that:

- $\{y_i \mid \mathsf{A}(y_1, \ldots, y_{\#(\mathsf{A})}) \in \mathrm{preds}(\varphi),\ i \in \lambda_\varphi(\mathsf{A})\} \subseteq \mathrm{fv}(\varphi)$, and
- for all rules $\mathsf{A}(x_1, \ldots, x_{\#(\mathsf{A})}) \leftarrow_{\mathcal{D}} \phi$, all predicate symbols $\mathsf{B}(y_1, \ldots, y_{\#(\mathsf{B})})$ that occur in ϕ and all $j \in \lambda_\varphi(\mathsf{B})$, we have $y_j = x_i$, for some $i \in \lambda_\varphi(\mathsf{A})$.

Intuitively, the profile of φ identifies those parameters of a predicate symbol that are always replaced by a top-level free variable in each unfolding of φ, according to the rules in $\mathcal{D}$; it can be computed by a greatest fixed point iteration over the rules in $\mathcal{D}$, in time $O(\mathrm{size}(\mathcal{D}) \cdot \mathrm{width}(\mathcal{D}))$.

Definition 13 *Given a profile λ, a rule $\mathsf{A}(x_1, \ldots, x_{\#(\mathsf{A})}) \leftarrow_{\mathcal{D}} \exists z_1 \ldots \exists z_k \,.\, \phi * \mathbin{\ast}_{\ell=1}^{h} \mathsf{B}^\ell(y_1^\ell, \ldots, y_{\#(\mathsf{B}^\ell)}^\ell)$, where ϕ is a predicateless symbolic configuration without occurrences of equality atoms, is called:*

1. progressing *if and only if each component (resp. interaction) atom of ϕ is of the form $C_i(x_1)$ (resp. $I_j(\mathbf{y})$, $x_1 \in \mathbf{y}$) and the sets $\{y_1^\ell, \ldots, y_{\#(\mathsf{B}^\ell)}^\ell\}$, $\ell \in [1,h]$, partition $\{x_2, \ldots, x_{\#(\mathsf{A})}, z_1, \ldots, z_k\}$,*
2. λ-connected *if and only if, for each $\ell \in [1,h]$, there exists an interaction atom in ϕ that contains both y_1^ℓ and a variable in $\{x_i \mid i \in \lambda(\mathsf{A})\} \cup \{x_1\}$.*

A SID is progressing *(*λ-connected*) if and only if all its rules are progressing (*λ-*connected).*

For instance, the SIDs from Example 4 and Fig. 6c are both progressing and connected. For two predicate symbols A and B, we write $\mathsf{A} \preceq_{\mathcal{D}} \mathsf{B}$ if and only if B occurs in the body of a rule from $\mathcal{D}$ that defines A. For a symbolic configuration ϕ, let

$\mathrm{dep}_{\mathcal{D}}(\phi) \stackrel{\mathsf{def}}{=} \{\mathsf{B} \mid \mathsf{A} \preceq^*_{\mathcal{D}} \mathsf{B},\ \mathsf{A}(x_1,\ldots,x_{\#(\mathsf{A})}) \in \mathrm{preds}(\phi)\}$, where $\preceq^*_{\mathcal{D}}$ is the reflexive and transitive closure of the $\preceq_{\mathcal{D}}$ relation. The following shows that the entailment problem becomes undecidable when the conditions of Def. 13 are even slightly lifted:

Proposition 4 *The entailment* $\mathsf{A}(x_1,\ldots,x_k) \models_{\mathcal{D}} \mathsf{B}(x_1,\ldots,x_k)$ *is undecidable, even when* $\mathcal{D}$ *is progressing and only the rules defining the predicate symbols from* $\mathrm{dep}_{\mathcal{D}}(\phi)$ *are* λ_ϕ*-connected.*

We prove the decidability of entailment problems for symbolic configurations interpreted over progressing and connected SIDs (Def. 13), via a reduction to a decidable entailment problem for Separation Logic (SL), interpreted over heaps. Let $\mathfrak{K} \geq 1$ be a fixed integer in the rest of this section. A *heap* is a finite partial function $\mathsf{h} : \mathbb{U} \rightharpoonup_{fin} \mathbb{U}^{\mathfrak{K}}$. The composition of two heaps h_1 and h_2 is their disjoint union $\mathsf{h}_1 \uplus \mathsf{h}_2$, defined if and only if $\mathrm{dom}(\mathsf{h}_1) \cap \mathrm{dom}(\mathsf{h}_2) = \emptyset$. The SL formulæ are:

$$\phi ::= \mathsf{emp} \mid x \mapsto (y_1,\ldots,y_{\mathfrak{K}}) \mid x \doteq y \mid \overline{\mathsf{A}}(x_1,\ldots,x_{\#(\overline{\mathsf{A}})}) \mid \phi_1 * \phi_2 \mid \exists x \,.\, \phi_1 \ .$$

Predicateless SL formulæ are interpreted by the relation:

$$(\nu,\mathsf{h}) \models^{\mathsf{SL}} \mathsf{emp} \text{ iff } \mathrm{dom}(\mathsf{h}) = \emptyset \qquad (\nu,\mathsf{h}) \models^{\mathsf{SL}} x \doteq y \text{ iff } \nu(x) = \nu(y) \text{ and } \mathrm{dom}(\mathsf{h}) = \emptyset$$
$$(\nu,\mathsf{h}) \models^{\mathsf{SL}} x \mapsto (y_1,\ldots,y_{\mathfrak{K}}) \text{ iff } \mathrm{dom}(\mathsf{h}) = \{\nu(x)\} \text{ and } \mathsf{h}(\nu(x)) = \langle \nu(y_1),\ldots,\nu(y_{\mathfrak{K}})\rangle \ .$$

As usual, the separating conjunction is interpreted using the composition of heaps and the predicate symbols are interpreted inductively over a set $\overline{\mathcal{D}}$ of rules $\overline{\mathsf{A}}(x_1,\ldots,x_{\#(\overline{\mathsf{A}})}) \leftarrow \phi$, where ϕ is a SL formula, such that $\mathrm{fv}(\phi) \subseteq \{x_1,\ldots,x_{\#(\overline{\mathsf{A}})}\}$. An entailment between SL formulæ, denoted as $\phi \models^{\mathsf{SL}}_{\overline{\mathcal{D}}} \psi$ is valid if and only if, for each store-heap pair (ν,h), if $(\nu,\mathsf{h}) \models^{\mathsf{SL}}_{\overline{\mathcal{D}}} \phi$ then $(\nu,\mathsf{h}) \models^{\mathsf{SL}}_{\overline{\mathcal{D}}} \psi$, where $\models^{\mathsf{SL}}_{\overline{\mathcal{D}}}$ is the homomorphic extension of $\models^{\mathsf{SL}}$ to SL formulæ with predicate atoms. The profile of a SL SID is defined similar to the one for CL.

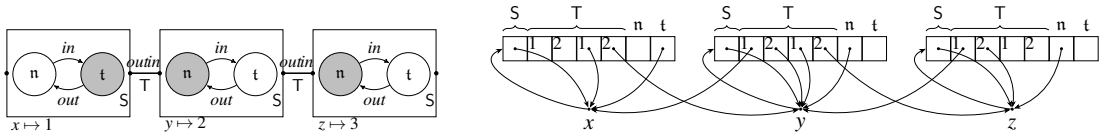
Definition 14 *Given a profile* λ*, a SL rule* $\overline{\mathsf{A}}(x_1,\ldots,x_{\#(\overline{\mathsf{A}})}) \leftarrow \phi$ *is said to be:*

1. progressing *if and only if* $\phi = \exists t_1 \ldots \exists t_k \,.\, x_1 \mapsto (y_1,\ldots,y_{\mathfrak{K}}) * \psi$*, where* ψ *contains only predicate and equality atoms, and*
2. λ-connected *if and only if* $\phi = \exists t_1 \ldots \exists t_k \,.\, x_1 \mapsto (y_1,\ldots,y_{\mathfrak{K}}) * \psi$ *and, for every predicate atom* $\overline{\mathsf{B}}(z_1,\ldots,z_{\#(\overline{\mathsf{B}})})$ *in* ψ*, we have* $z_1 \in \{x_i \mid i \in \lambda(\overline{\mathsf{A}})\} \cup \{y_1,\ldots,y_{\mathfrak{K}}\}$.

Note that the definitions of progressing and connected rules are different for SL, compared to CL (Def. 13); in the rest of this section, we rely on the context to distinguish progressing (connected) SL rules from progressing (connected) CL rules. The tight complexity class for the entailment problem between SL formulæ interpreted by progressing and connected SIDs is given below:

Theorem 3 ([?]) *The SL entailment problem* $\phi \models^{\mathsf{SL}}_{\overline{\mathcal{D}}} \psi$ *is* 2EXP*-complete, if* $\overline{\mathcal{D}}$ *is progressing and* λ_ϕ*-connected.*

Figure 8: Gaifman Heap for a Chain Configuration



(a) (b)

The reduction of CL to SL entailments is based on the idea of representing a logical structure σ, over a signature $\mathfrak{S} = \langle C_1, \ldots, C_n, I_1, \ldots, I_m \rangle$, by an undirected *Gaifman graph*, in which every k-tuple from the interpretation of a relation symbol becomes a k-clique. Let the *degree* of an index $u \in \mathrm{nodes}(\sigma)$ be the maximum number of interactions, of a given type, involving u and the degree of σ be the maximum degree among all $u \in \mathrm{nodes}(\sigma)$, denoted as $\delta(\sigma)$. We encode a configuration $(\sigma, \nu, \rho) \in \Gamma$ by a store-heap pair (ν, h), where h is defined below, using the integer function $\mathrm{pos}_b(j,i,k,\ell) \stackrel{\mathsf{def}}{=} n + b \cdot \sum_{h=1}^{j} \#(I_h) + i \cdot \#(I_{j+1}) + k \cdot \|\mathbb{Q}\| + \ell$, where $\mathbb{Q}$ is the finite set of states used to describe the behaviors of component types (§2):

Definition 15 *Given a state map ρ, a structure σ and an integer $b \geq \delta(\sigma)$, a* Gaifman heap *of (σ, ρ) is a heap $\mathsf{h} : \mathbb{U} \rightharpoonup_{fin} \mathbb{U}^{\mathfrak{K}}$, where $\mathfrak{K} = \mathrm{pos}_b(m, 0, n, 0)$, such that $\mathrm{dom}(\mathsf{h}) = \mathrm{nodes}(\sigma)$ and, for all $u \in \mathrm{dom}(\mathsf{h})$, such that $\mathsf{h}(u) = \langle u_1, \ldots, u_{\mathfrak{K}} \rangle$, the following hold:*

1. *for all $i \in [1,n]$, we have $u_i = u$ if and only if $u \in C_i^{\sigma}$,*
2. *for all $j \in [1,m]$, if $\mathbf{u}_1, \ldots, \mathbf{u}_h$ are the tuples from I_j^{σ} containing u, then there exist integers $0 \leq k_1 < \ldots < k_h < b$, such that $\langle \mathsf{h}(u) \rangle_{\mathrm{ipos}(j,k_i)} = \mathbf{u}_i$, for each $i \in [1,h]$, where the entries from $\mathrm{ipos}(j,k) \stackrel{\mathsf{def}}{=} [\mathrm{pos}_b(j-1,k,0,0), \mathrm{pos}_b(j-1,k+1,0,0)]$ encode the i-th tuple from I_j^{σ},*
3. *for all $i \in [1,n]$ and $k \in [1, \|\mathbb{Q}\|]$, we have $\langle \mathsf{h}(u) \rangle_{\mathrm{spos}(i,k)} = u$ if and only if $\rho(u, C_i) = q_k$, where $\mathrm{spos}(i,k) \stackrel{\mathsf{def}}{=} \mathrm{pos}_b(m, 0, i-1, k)$ is the position of the k-th state in C_i^{σ} and $q_1, \ldots, q_{\|\mathbb{Q}\|}$ is the enumeration of $\mathbb{Q}$, in some fixed predefined order.*

We denote by $\mathcal{G}_b(\sigma, \rho)$ the set of Gaifman heaps for (σ, ρ) and b.

Intuitively, if $\mathsf{h} \in \mathcal{G}_b(\sigma, \rho)$ and $u \in \mathrm{dom}(\mathsf{h})$ is an index, then the first n entries of $\mathsf{h}(u)$ represent the types C_i of the components indexed by u (i.e. $u \in C_i^{\sigma}$), the next $b \cdot \sum_{j=1}^{m} \#(I_j)$ entries are used to encode the interactions of each type I_j, whereas the last $n \cdot \|Q\|$ entries are used to represent the state map (i.e. the state $\rho(u, C_i)$, for each component type C_i).

Example 10 *Fig. 8b shows a Gaifman heap for the configuration in Fig. 8a, over the signature from Example 2, where each index belongs to at most $b = 2$ tuples of the interaction type* T. ■

Note that the Gaifman heap encoding is not unique: two Gaifman heaps for the same structure and state map may actually differ by the order of tuples from the encoding of an interaction type I_j^{σ} (point 2 of Def. 15) and by the choice of the unconstrained locations in $\mathsf{h}(u)$, for each $u \in \mathrm{dom}(\mathsf{h})$.

In the following, we build an SID $\overline{\mathcal{D}}$ defining the Gaifman heaps of the $\mathcal{D}$-models of the predicate atoms from the CL entailment. Because the degree of heaps is always fixed by $\mathfrak{K}$, which is a parameter of SL, this construction is only possible under the following assumption:

Assumption 3 *There exists an integer $\mathfrak{B} \geq 1$, such that $\delta(\sigma) \leq \mathfrak{B}$, for every configuration $(\sigma, \nu, \rho) \in [\![\alpha]\!]_{\mathcal{D}}$, where $\alpha \in \mathrm{def}(\mathcal{D})$ is a predicate atom and fix $\mathfrak{K} \stackrel{\mathsf{def}}{=} \mathrm{pos}_{\mathfrak{B}}(m, 0, n, 0)$.*

In our examples, it is sufficient to take $\mathfrak{B} = 2$, for the predicate atoms $\mathsf{chain}_{h,t}(x,y)$ (Example 4) and $\mathsf{tree}(x)$, $\mathsf{tree}_{idle}(x)$, $\mathsf{tree}_{\neg idle}(x)$ and $\mathsf{tseg}(x,y)$ (Fig. 6c). Moreover, the existence of a bound on the degree of a model of a predicate atom is subject to the following cut-off result:

Proposition 5 *There exists an integer $\mathfrak{B} \geq 1$, such that $\delta(\sigma) \leq \mathfrak{B}$, for each $(\sigma, \nu, \rho) \in [\![\mathsf{A}(x_1, \ldots, x_{\#(\mathsf{A})})]\!]_{\mathcal{D}}$ only if $\mathfrak{B} = O(\mathrm{size}(\mathcal{D})^c)$, for a constant $c \geq 1$.*

However, there are distributed systems with coordinating architectures described by structures of unbounded degree, such as star topologies, having a designated controller node, with which every other node communicates. We conjecture that it is possible to lift Assumption 3 and tackle these more general cases by means of domain-specific results, considered for future work.

Back to the definition of $\overline{\mathcal{D}}$, we associate to each variable x, that occurs free or existentially quantified in $\mathcal{D}$, a unique $\mathfrak{K}$-tuple of variables $\eta(x) \in \mathbb{V}^{\mathfrak{K}}$, that represents the image of the store value $\nu(x)$ in a Gaifman heap h, namely $\mathsf{h}(\nu(x)) = \nu(\eta(x))$. Moreover, we consider, for each predicate symbol A, that is defined by $\mathcal{D}$, an annotated predicate symbol $\overline{\mathsf{A}}_{\iota}$ of arity $\#(\overline{\mathsf{A}}_{\iota}) = (\mathfrak{K}+1) \cdot \#(\mathsf{A})$, where $\iota : [1, \#(\mathsf{A})] \times [1, m] \to 2^{[0, \mathfrak{B}-1]}$ is a mapping associating a parameter position $k \in [1, \#(\mathsf{A})]$ and an interaction symbol I_j, for $j \in [1, m]$, a set of integers $\iota(k, j)$ denoting the positions of the encodings of the interactions of type I_j, involving the value of x_k, in the $\overline{\mathcal{D}}$-models of $\overline{\mathsf{A}}_{\iota}(x_1, \ldots, x_{\#(\mathsf{A})}, \eta(x_1), \ldots, \eta(x_{\#(\mathsf{A})}))$ (point 2 of Def. 15). Then $\overline{\mathcal{D}}$ consists of rules of the form:

$$\overline{\mathsf{A}}_{\iota}(x_1, \ldots, x_{\#(\mathsf{A})}, \eta(x_1), \ldots, \eta(x_{\#(\mathsf{A})})) \leftarrow \exists z_1 \ldots \exists z_p \exists \eta(z_1) \ldots \exists \eta(z_p) \,.\, \overline{\phi} * \circledast_{\ell=1}^{h} \overline{\mathsf{B}}^{\ell}_{\iota^{\ell}}(y^{\ell}_1, \ldots, y^{\ell}_{\#(\mathsf{B}^{\ell})}, \eta(y^{\ell}_1), \ldots, \eta(y^{\ell}_{\#(\mathsf{B}^{\ell})})) \quad (1)$$

for which there exist a *stem rule* $\mathsf{A}(x_1, \ldots, x_{\#(\mathsf{A})}) \leftarrow \exists z_1 \ldots \exists z_p \,.\, \phi * \circledast_{\ell=1}^{h} \mathsf{B}^{\ell}(y^{\ell}_1, \ldots, y^{\ell}_{\#(\mathsf{B}^{\ell})})$ in $\mathcal{D}$. A rule of the form (1) is *well-formed* if and only if, for each $i \in [1, \#(\mathsf{A})]$ and $j \in [1, m]$, there exists a set of integers $Y_{i,j} \subseteq [0, \mathfrak{B}-1]$, such that:

- $\|Y_{i,j}\| = \| I^j_{\phi}(x_i) \|$, where $I^j_{\phi}(x) \stackrel{\mathsf{def}}{=} \{ I_j(\mathbf{y}) \in \mathrm{inter}(\phi) \mid \langle y \rangle_k \simeq_{\phi} x,\ k \in [1, \#(I_j)] \}$, and
- $Y_{i,j} \subseteq \iota(i,j)$ and $\iota(i,j) \setminus Y_{i,j} = \mathcal{Z}_j(x_i)$, where $\mathcal{Z}_j(x) \stackrel{\mathsf{def}}{=} \bigcup_{\ell=1}^{h} \bigcup_{k=1}^{\#(\mathsf{B}^{\ell})} \{ \iota^{\ell}(k, j) \mid x \simeq_{\phi} y^{\ell}_k \}$ denotes the set of positions used to encode the interactions of type I_j involving the store value of the variable x, in a $\mathcal{D}$-model of $\circledast_{\ell=1}^{h} \mathsf{B}^{\ell}(y^{\ell}_1, \ldots, y^{\ell}_{\#(\mathsf{B}^{\ell})})$.

Let $\overline{\mathcal{D}}$ be the set of well-formed rules (1), whose predicateless subformulæ $\overline{\phi}$ are defined below:

$\overline{\phi} \stackrel{\text{def}}{=} x_1 \mapsto \eta(x_1) \ * \ \ast_{x \in \text{fv}(\phi)} \mathsf{CompState}_\phi(x) \ * \ \ast_{i=1}^{\#(\mathsf{A})} \mathsf{Inter}_\phi(x_i)$, where we define:

$\mathsf{CompState}_\phi(x) \stackrel{\text{def}}{=} \ast_{C_i(x) \text{ occurs in } \phi} \langle \eta(x) \rangle_i \doteq x \ * \ \ast_{\mathsf{state}(x,C_i,q_k) \text{ occurs in } \phi} \langle \eta(x) \rangle_{\text{spos}(i,k)} \doteq x$

$\mathsf{Inter}_\phi(x_i) \stackrel{\text{def}}{=} \ast_{j=1}^{m} \ast_{q=1}^{r_j} \langle \eta(x_i) \rangle_{\text{ipos}(j,k_q^j)} \doteq \mathbf{x}_q^j$, for the following sets:

$\{I_j(\mathbf{x}_1^j), \ldots, I_j(\mathbf{x}_{r_j}^j)\} \stackrel{\text{def}}{=} I_\phi^j(x_i)$, the interaction atoms involving x_i in ϕ, and

$\{k_1^j, \ldots, k_{r_j}^j\} \stackrel{\text{def}}{=} \iota(i,j) \setminus \mathcal{Z}_j(x_i)$, the encoding positions of their corresponding interactions.

We write $\mathbf{x} \doteq \mathbf{y}$ for $\ast_{i=1}^{k} \langle \mathbf{x} \rangle_i \doteq \langle \mathbf{y} \rangle_i$, where $\mathbf{x}$ and $\mathbf{y}$ are tuples of variables of length k. Intuitively, the SL formula $\mathsf{CompState}_\phi(x)$ realizes the encoding of the component and state atoms from ϕ, in the sense of points (1) and (3) from Def. 15, whereas the formula $\mathsf{Inter}_\phi(x_i)$ realizes the encodings of the interactions involving a parameter x_i in the stem rule of (1)[11] (point 2 of Def. 15). The main result of this section relies on the two technical lemmas below:

Lemma 4 *If $\mathcal{D}$ is progressing, for each $\mathcal{D}$-model (σ, ν, ρ) of $\mathsf{A}(x_1, \ldots, x_{\#(\mathsf{A})}) \in \text{def}(\mathcal{D})$ and each heap $\mathsf{h} \in \mathcal{G}_{\mathfrak{B}}(\sigma, \rho)$, there exists a mapping $\iota : [1, \#(\mathsf{A})] \times [1, m] \to 2^{[0, \mathfrak{B}-1]}$ and a store $\overline{\nu}$, such that the following hold:*

1. *$\overline{\nu}(x_i) = \nu(x_i) \in \text{dom}(\mathsf{h})$ and $\overline{\nu}(\eta(x_i)) = \mathsf{h}(\nu(x_i))$, for each $i \in [1, \#(\mathsf{A})]$,*
2. *$\{\mathbf{u} \in I_j^\sigma \mid \overline{\nu}(x_i) \in \mathbf{u}\} = \{\langle \mathsf{h}(\overline{\nu}(x_i)) \rangle_{\text{ipos}(j,k)} \mid k \in \iota(i,j)\}$, for all $i \in [1, \#(\mathsf{A})]$ and $j \in [1, m]$,*
3. *$(\overline{\nu}, \mathsf{h}) \models^{\mathsf{SL}}_{\overline{\mathcal{D}}} \overline{\mathsf{A}}_\iota(x_1, \ldots, x_{\#(\mathsf{A})}, \eta(x_1), \ldots, \eta(x_{\#(\mathsf{A})}))$.*

The following lemma states the dual of Lemma 4:

Lemma 5 *If $\mathcal{D}$ is progressing, for a predicate atom $\mathsf{A}(x_1, \ldots, x_{\#(\mathsf{A})}) \in \text{def}(\mathcal{D})$, each mapping $\iota : [1, \#(\mathsf{A})] \times [1, m] \to 2^{[0, \mathfrak{B}-1]}$ and each $\overline{\mathcal{D}}$-model $(\overline{\nu}, \mathsf{h})$ of $\overline{\mathsf{A}}_\iota(x_1, \ldots, x_{\#(\mathsf{A})}, \eta(x_1), \ldots, \eta(x_{\#(\mathsf{A})}))$, the following hold:*

1. *for each $i \in [1, \#(\mathsf{A})]$, we have $\overline{\nu}(x_i) \in \text{dom}(\mathsf{h})$ and $\mathsf{h}(\overline{\nu}(x_i)) = \overline{\nu}(\eta(x_i))$, and*
2. *there is a structure σ and a state map ρ, such that $\mathsf{h} \in \mathcal{G}_{\mathfrak{B}}(\sigma, \rho)$ and $(\sigma, \overline{\nu}, \rho) \models_{\mathcal{D}} \mathsf{A}(x_1, \ldots, x_{\#(\mathsf{A})})$.*

Theorem 4 *If $\mathcal{D}$ is progressing and $\lambda_{\mathsf{A}(x_1, \ldots, x_{\#(\mathsf{A})})}$-connected, then the entailment $\mathsf{A}(x_1, \ldots, x_{\#(\mathsf{A})}) \models_{\mathcal{D}} \exists y_1 \ldots \exists y_r \cdot \bigvee_{\ell=1}^{h} \mathsf{B}^\ell(z_1^\ell, \ldots, z_{\#(\mathsf{B}^\ell)}^\ell)$ is 2EXP-complete.*

The proof of the upper bound uses a result on SL entailments (Theorem 3), that relies on an algorithm simply exponential in the size and doubly exponential in the

[11] The definition of $\mathsf{Inter}_\phi(x_i)$ uses the fact that the rule is well-formed, which implies $\|I_\phi^j(x_i)\| = \|\iota(i,j) \setminus \mathcal{Z}_j(x_i)\|$.

width of $\overline{\mathcal{D}}$ (a generalization of the method described in [**?**, **?**]). Since our reduction of CL to SL entailments uses annotated predicate symbols $\overline{\mathsf{A}}_{\iota}$, the number of rules in $\overline{\mathcal{D}}$ is increased by at most one exponential, yielding the 2EXP upper bound. Moreover, this matches the 2EXP-hard lower bound, obtained by reduction from SL entailments [**?**]. The undecidability result of Prop. 4 shows the importance of the fact that $\mathcal{D}$ is $\lambda_{\mathsf{A}(x_1,\ldots,x_{\#(\mathsf{A})})}$-connected, as entailments between predicate atoms become undecidable whenever this condition is lifted.

Note that, even if restricted, the form of entailments from Theorem 4 suffices in our examples, that can be easily pre-processed to fit within the decidable class:

Example 11 *Consider the entailments from Example 9, for instance:*

$$\mathsf{chain}_{2,1}(a,b) * \mathsf{T}(b,a) \models_{\mathcal{D}} \exists x \exists y \exists z \,.\, \mathsf{T}(x,y) * \mathsf{S}^{\mathsf{n}}(y) * \mathsf{T}(y,z) * \mathsf{chain}_{1,1}(z,x)$$

By unfolding the left hand side, we obtain two entailments, corresponding to the two rules defining $\mathsf{chain}_{2,1}(a,b)$ *(Example 4):*

$$\begin{array}{lcl} \mathsf{S}^{\mathsf{t}}(a) * \mathsf{T}(a,c) * \mathsf{chain}_{2,0}(c,b) * \mathsf{T}(b,a) & \models_{\mathcal{D}} & \exists x \exists y \exists z \,.\, \mathsf{T}(x,y) * \mathsf{S}^{\mathsf{n}}(y) * \mathsf{T}(y,z) * \mathsf{chain}_{1,1}(z,x) \\ \mathsf{S}^{\mathsf{n}}(a) * \mathsf{T}(a,c) * \mathsf{chain}_{1,1}(c,b) * \mathsf{T}(b,a) & \models_{\mathcal{D}} & \exists x \exists y \exists z \,.\, \mathsf{T}(x,y) * \mathsf{S}^{\mathsf{n}}(y) * \mathsf{T}(y,z) * \mathsf{chain}_{1,1}(z,x) \end{array}$$

Next, we introduce the following progressing and connected rules:

$$\begin{array}{rcl} \mathsf{A}_1(x) & \leftarrow & \exists y \exists z \,.\, \mathsf{S}^{\mathsf{t}}(x) * \mathsf{T}(x,y) * \mathsf{chain}_{2,0}(y,z) * \mathsf{T}(z,x) \\ \mathsf{A}_2(x) & \leftarrow & \exists y \exists z \,.\, \mathsf{S}^{\mathsf{n}}(x) * \mathsf{T}(x,y) * \mathsf{chain}_{1,1}(y,z) * \mathsf{T}(z,x) \\ \mathsf{B}(x) & \leftarrow & \exists y \exists z \,.\, \mathsf{T}(y,x) * \mathsf{S}^{\mathsf{n}}(x) * \mathsf{T}(x,z) * \mathsf{chain}_{1,1}(z,y) \end{array}$$

The previous entailment is valid if and only if $\mathsf{A}_1(x) \models_{\mathcal{D}} \exists y \,.\, \mathsf{B}(y)$ *and* $\mathsf{A}_2(x) \models_{\mathcal{D}} \exists y \,.\, \mathsf{B}(y)$. ■

Finally, we apply the result of Theorem 4 to decide the following *tightness problem* for symbolic configurations (Def. 6):

Proposition 6 *The problem* is a given symbolic configuration tight? *is in* 3EXP.

The proof uses a reduction to an entailment problem between predicate symbols annotated with tuples of sets of ports that are provided in each unfolding of a predicate atom. The blowup from 2EXP (Theorem 4) to 3EXP is caused by the annotation of predicate symbols, that increases the number of rules by an exponential factor.

8 Related Work

The ability of reconfiguring coordinating architectures of software systems has recently received much interest in the Software Engineering community [**?**, **?**]. We consider *programmed reconfiguration*, in which the in the architecture changes occur according to a sequential program, executed in parallel with the system to which reconfiguration applies. The languages used to write such programs are classified according to the underlying formalism used to define their operational semantics: *process algebras*, e.g. π-ADL [**?**], DARWIN [**?**], *graph rewriting* [**?**, **?**], *chemical reactions* [**?**], etc. We

separate architectures from behaviors, thus relating to the BIP framework [**?**] and its extensions for dynamic reconfigurable systems DR-BIP [**?**]. In a similar vein, the REO language [**?**] supports reconfiguration by changing the structure of connectors [**?**].

Checking the correctness of a dynamically reconfigurable system considers mainly *runtime verification* methods, i.e. checking a given finite trace of observed configurations against a logical specification. For instance, in [**?**], configurations are described by annotated hyper-graphs and configuration invariants of finite traces, given first-order logic, are checked using ALLOY [**?**]. More recently, [**?**, **?**, **?**] apply temporal logic for runtime verification of reconfigurable systems. Model checking of temporal specification is also applied to REO programs, under simplifying assumption that render the system finite-state [**?**]. In contrast, we use induction to deal with parameterized systems of unbounded sizes.

To the best of our knowledge, our work is the first to tackle the *verification* of reconfiguration programs, by formally proving the absence of bugs, using a Hoare-style annotation of a reconfiguration program with assertions that describe infinite sets of configurations, with unboundedly many components. Reasoning about the correctness of unbounded networks of parallel processes uses mostly hard-coded architectures (see [**?**] for a survey), whereas more recent architecture description logics [**?**, **?**] do not deal with the reconfigurability aspect of distributed systems.

Our assertion language is a resource logic that supports local reasoning [**?**]. Local reasoning about parallel programs has been traditionally within the scope of Concurrent Separation Logic (CSL), that introduced a parallel composition rule [**?**], with a non-interfering (race-free) semantics of shared-memory parallelism [**?**]. Considering interference in CSL requires more general proof rules, combining ideas of assume-and rely-guarantee [**?**, **?**] with local reasoning [**?**, **?**] and abstract notions of framing [**?**, **?**, **?**]. These rules generalize from both standard CSL parallel composition and rely-guarantee rules, allowing even to reason about properties of concurrent objects, such as (non-)linearizability [**?**]. However, the body of work on CSL deals almost entirely with shared-memory multithreading programs, instead of distributed systems, which is the aim of our work.

9 Conclusions and Future Work

We present a framework for deductive verification of reconfiguration programs, based on a configuration logic that supports local reasoning. We prove the absence of design bugs in ideal networks, without packet loss and communication delays, using a discrete event-based model of behavior, the usual level of abstraction in formal verification of parameterized distributed systems. Our configuration logic relies on inductive predicates to describe systems with unbounded number of components. It is used to annotate reconfiguration programs with Hoare triples, whose validity relies on havoc invariants about the ongoing interactions in the system. These invariants are tackled with a specific proof system, that uses a parallel composition rule in the style of assume/rely-guarantee reasoning. Finally, we give the tight complexity of entailments between predicate atoms in the configuration logic, as a step towards automating the search for proofs of reconfiguration programs.

As future work, we consider push-button techniques for havoc invariant synthesis, allowing broadcast interactions between all the components, and extensions of the finite-state model of behavior, using timed and hybrid automata.

References

A Proofs from Section 3

Proposition 1 *The set of symbolic configurations built using predicate atoms* $\mathsf{chain}_{h,t}(x,y)$*, for* $h,t \geq 0$ *(Example 4) is precisely closed.*

Proof. Let $\varphi_i \stackrel{\mathsf{def}}{=} \phi_i * \circledast_{j=1}^{k_i} \mathsf{chain}_{h_{i,j},t_{i,j}}(x_{i,j},y_{i,j})$ be symbolic configurations, where ϕ_i is a predicateless symbolic configuration and $h_{i,j},t_{i,j} \geq 0$ are integers, for all $j \in [1,k_i]$ and $i = 1,2$. We prove that φ_1 is precise on $[\![\varphi_2]\!]_{\mathcal{D}}$. Let $\gamma = (\sigma,\nu,\rho) \in [\![\varphi_2]\!]_{\mathcal{D}}$ be a configuration and suppose that there exist configurations $\gamma' = (\sigma',\nu,\rho), \gamma'' = (\sigma'',\nu,\rho)$, such that $\gamma' \sqsubseteq \gamma$, $\gamma'' \sqsubseteq \gamma$, $\gamma' \models_{\mathcal{D}} \varphi_1$ and $\gamma'' \models_{\mathcal{D}} \varphi_1$. Then there exist configurations $\gamma'_0 \stackrel{\mathsf{def}}{=} (\sigma'_0,\nu,\rho),\ldots,\gamma'_{k_1} \stackrel{\mathsf{def}}{=} (\sigma'_{k_1},\nu,\rho)$ and $\gamma''_0 \stackrel{\mathsf{def}}{=} (\sigma''_0,\nu,\rho),\ldots,\gamma''_{k_1} \stackrel{\mathsf{def}}{=} (\sigma''_{k_1},\nu,\rho)$, such that:

- $\gamma' = \circledast_{j=0}^{k_1} \gamma'_j$ and $\gamma'' = \circledast_{j=0}^{k_1} \gamma''_j$,
- $\gamma'_0 \models \phi_1$ and $\gamma''_0 \models \phi_1$, and
- $\gamma'_j \models_{\mathcal{D}} \mathsf{chain}_{h_{1,j},t_{1,j}}(x_{1,j},y_{1,j})$ and $\gamma''_j \models_{\mathcal{D}} \mathsf{chain}_{h_{1,j},t_{1,j}}(x_{1,j},y_{1,j})$, for all $j \in [1,k_1]$.

Since ϕ_1 is a predicateless symbolic heap, we have $\mathsf{S}^{\sigma'_0} = \mathsf{S}^{\sigma''_0}$ and $\mathsf{T}^{\sigma'_0} = \mathsf{T}^{\sigma''_0}$, thus $\gamma'_0 = \gamma''_0$. Moreover, for each $j \in [1,k_1]$, we have $\mathsf{S}^{\sigma'_j} = \mathsf{S}^{\sigma''_j}$ and $\mathsf{T}^{\sigma'_j} = \mathsf{T}^{\sigma''_j}$, because both relations consist of all the interactions $\langle u_1,u_2\rangle, \langle u_2,u_3\rangle, \ldots, \langle u_{\ell-1},u_\ell\rangle \in \mathsf{T}^{\sigma}$, such that $\nu(x_{1,j}) = u_1$ and $\nu(y_{1,j}) = u_\ell$. Thus, we obtain $\gamma'_j = \gamma''_j$, leading to $\gamma' = \gamma''$. $\square$

B Proofs from Section 4

Lemma 1 *For each axiom* $\{\phi\}\ \mathsf{R}\ \{\psi\}$*, where* $\mathsf{R} \in \mathfrak{P}$*, the following hold:*

1. $[\![\phi]\!]_{\mathcal{D}} = \min_{\sqsubseteq}\{\gamma \in \Gamma \mid \langle\!\langle \mathsf{R} \rangle\!\rangle(\gamma) \neq \top\}$,
2. $\langle\!\langle \mathsf{R} \rangle\!\rangle([\![\phi]\!]_{\mathcal{D}}) = [\![\psi]\!]_{\mathcal{D}}$.

Proof. (1) Let $\sigma_\emptyset$ be the structure with $C_i^{\sigma_\emptyset} = I_j^{\sigma_\emptyset} = \emptyset$, for all $i \in [1,n]$ and all $j \in [1,m]$. The proof goes by case split on the type of the primitive command R, which determines the precondition ϕ of the axiom:

- $\mathsf{R} = \mathsf{new}(C_i,x)$ and $\phi = \mathsf{emp}$: "$\subseteq$" Any configuration with empty structure is $\sqsubseteq$-minimal and, moreover, $\mathsf{new}(C_i,x)$ never faults. "$\supseteq$" Suppose, for a contradiction, the existence of a configuration $(\sigma,\nu,\rho) \in \min_{\sqsubseteq}\{\gamma \in \Gamma \mid \langle\!\langle \mathsf{new}(C_i,x) \rangle\!\rangle(\gamma) \neq \top\}$ such that $(\sigma,\nu,\rho) \not\models \mathsf{emp}$. Then $(\sigma_\emptyset,\nu,\rho) \sqsubset (\sigma,\nu,\rho)$, thus $\langle\!\langle \mathsf{new}(C_i,x) \rangle\!\rangle(\sigma_\emptyset,\nu,\rho) = \top$, contradiction.
- $\mathsf{R} = \mathsf{delete}(C_i,x)$ and $\phi = C_i(x)$: "$\subseteq$" Let (σ,ν,ρ) be a configuration, such that $C_i^{\sigma} = \{\nu(x)\}$, $C_j^{\sigma} = \emptyset$, for all $j \in [1,n] \setminus \{i\}$ and $I_j^{\sigma} = \emptyset$, for all $j \in [1,m]$. We have that $\langle\!\langle \mathsf{delete}(C_i,x) \rangle\!\rangle(\sigma,\nu,\rho) \neq \top$ and show that $\langle\!\langle \mathsf{delete}(C_i,x) \rangle\!\rangle(\gamma) = \top$, for all configurations $\gamma \sqsubset (\sigma,\nu,\rho)$. Let γ be any such configuration. Then there exists a configuration (σ',ν,ρ) such that $\sigma' \neq \sigma_\emptyset$ and $\gamma \bullet (\sigma',\nu,\rho) = (\sigma,\nu,\rho)$. Then $\gamma = (\sigma_\emptyset,\nu,\rho)$ is the only possibility and we have $\langle\!\langle \mathsf{delete}(C_i,x) \rangle\!\rangle(\gamma) = \top$. "$\supseteq$"

Let $(\sigma,\nu,\rho) \in \min_{\sqsubseteq}\{\gamma \in \Gamma \mid \langle\langle \mathsf{delete}(C_i,x)\rangle\rangle(\gamma) \neq \top\}$ be a configuration. Then $\nu(x) \in C_i^\sigma$ and, since (σ,ν,ρ) is $\sqsubseteq$-minimal, we have $C_i^\sigma = \{\nu(x)\}$, $C_j^\sigma = \emptyset$, for all $j \in [1,n] \setminus \{i\}$ and $I_j^\sigma = \emptyset$, for all $j \in [1,m]$, thus $(\sigma,\nu,\rho) \models C_i(x)$.

- $\mathsf{R} = \mathsf{connect}(I_j,x_1,\ldots,x_{\#(I_j)})$ and $\phi = \mathsf{emp}$: similar to the case $\mathsf{R} = \mathsf{new}(C_i,x)$ and $\phi = \mathsf{emp}$.

- $\mathsf{R} = \mathsf{disconnect}(I_j,x_1,\ldots,x_{\#(I_j)})$ and $\phi = I_j(x_1,\ldots,x_{\#(I_j)})$: similar to the case $\mathsf{R} = \mathsf{delete}(C_i,x)$ and $\phi = C_i(x)$.

- $\mathsf{R} = \mathsf{skip}$ and $\phi = \mathsf{emp}$: trivial.

(2) The proof goes by case split on the type of the primitive command R, which determines the pre- and post-condition ϕ and ψ of the axiom, respectively:

- $\mathsf{R} = \mathsf{new}(C_i,x)$, $\phi = \mathsf{emp}$ and $\psi = C_i(x) \wedge \mathsf{state}(x,C_i,q_i^0)$, where $\mathcal{B}(C_i) = (Q_i,P_i,q_i^0,\rightarrow_i)$:

  $$\begin{array}{l}\langle\langle \mathsf{new}(C_i,x)\rangle\rangle([\![\mathsf{emp}]\!]) = \langle\langle \mathsf{new}(C_i,x)\rangle\rangle(\{(\sigma,\nu,\rho) \in \Gamma \mid \sigma = \sigma_\emptyset\}) \\ = \{(\langle C_1^{\sigma_\emptyset},\ldots,C_i^{\sigma_\emptyset} \cup \{u\},\ldots,C_n^{\sigma_\emptyset},I_1^{\sigma_\emptyset},\ldots,I_m^{\sigma_\emptyset}\rangle,\nu[x \leftarrow u],\rho[(u,C_i) \leftarrow q_i^0]) \mid u \in \mathbb{U}\} \\ = [\![C_i(x) \wedge \mathsf{state}(x,C_i,q_i^0)]\!]\end{array}$$

  The third step applies the definition $\langle\langle \mathsf{R}\rangle\rangle([\![\phi]\!]) = \{\gamma' \mid \exists \gamma \in [\![\phi]\!] \,.\, \mathsf{R} : \gamma \rightsquigarrow \gamma'\}$ to the case $\mathsf{R} = \mathsf{new}(C_i,x)$, where the judgement $\mathsf{new}(C_i,x) : \gamma \rightsquigarrow \gamma'$ is defined in Fig. 2. The rest is by the semantics of CL.

- $\mathsf{R} = \mathsf{delete}(C_i,x)$, $\phi = C_i(x)$ and $\psi = \mathsf{emp}$:

  $$\begin{array}{l}\langle\langle \mathsf{delete}(\mathsf{R}_i,x)\rangle\rangle([\![C_i(x)]\!]) = \\ \langle\langle \mathsf{delete}(\mathsf{R}_i,x)\rangle\rangle(\{(\sigma,\nu,\rho) \in \Gamma \mid C_i^\sigma = \{\nu(x)\}, C_j^\sigma = \emptyset, j \in [1,n] \setminus \{i\}, I_k^\sigma = \emptyset, k \in [1,m]\}) = \\ \{(\sigma_\emptyset,\nu,\rho) \mid (\sigma,\nu,\rho) \in \Gamma\} = [\![\mathsf{emp}]\!]\end{array}$$

  The third step applies the definition $\langle\langle \mathsf{R}\rangle\rangle([\![\phi]\!]) = \{\gamma' \mid \exists \gamma \in [\![\phi]\!] \,.\, \mathsf{R} : \gamma \rightsquigarrow \gamma'\}$ to the case $\mathsf{R} = \mathsf{delete}(C_i,x)$, where the judgement $\mathsf{delete}(C_i,x) : \gamma \rightsquigarrow \gamma'$ is defined in Fig. 2. The rest is by the semantics of CL.

- $\mathsf{R} = \mathsf{connect}(I_j,x_1,\ldots,x_{\#(I_j)})$, $\phi = \mathsf{emp}$ and $\psi = I_j(x_1,\ldots,x_{\#(I_j)})$: similar to the case $\mathsf{R} = \mathsf{new}(C_i,x)$.

- $\mathsf{R} = \mathsf{disconnect}(I_j,x_1,\ldots,x_{\#(I_j)})$, $\phi = I_j(x_1,\ldots,x_{\#(I_j)})$ and $\psi = \mathsf{emp}$: similar to the case $\mathsf{R} = \mathsf{delete}(C_i,x)$.

- $\mathsf{R} = \mathsf{skip}$ and $\phi = \mathsf{emp}$: trivial. $\square$

Lemma 2 *For every program* $\mathsf{R} \in \mathfrak{L}$*, the action* $\langle\langle \mathsf{R}\rangle\rangle$ *is local for* $\mathsf{modif}(\mathsf{R})$.

Proof. By induction on the structure of the local program R. For the base case $\mathsf{R} \in \mathfrak{P}$, we check the following points, for all $\gamma_i = (\sigma_i,\nu,\rho) \in \Gamma$, for $i = 1,2$, such that $\sigma_1 \perp \sigma_2$:

- $\mathsf{R} = \mathsf{new}(C_i, x)$: we compute $\langle\langle \mathsf{new}(C_i,x)\rangle\rangle(\gamma_1 \bullet \gamma_2) =$

  $\{(\langle C_1^{\sigma_1 \uplus \sigma_2}, \ldots, C_i^{\sigma_1 \uplus \sigma_2} \cup \{u\}, \ldots, I_1^{\sigma_1 \uplus \sigma_2}, \ldots, I_m^{\sigma_1 \uplus \sigma_2}\rangle, \nu[x \leftarrow u], \rho[(u, C_i) \leftarrow q_i^0]) \mid u \in \mathbb{U} \setminus C_i^{\sigma_1 \uplus \sigma_2}\} \subseteq$
  $\{(\langle C_1^{\sigma_1}, \ldots, C_i^{\sigma_1} \cup \{u\}, \ldots, I_1^{\sigma_1}, \ldots, I_m^{\sigma_1}\rangle), \nu[x \leftarrow u], \rho[(u, C_i) \leftarrow q_i^0]) \mid u \in \mathbb{U} \setminus C_i^{\sigma_1}\} \bullet \{\gamma_2\}\uparrow^{\{x\}} =$
  $\langle\langle \mathsf{new}(C_i,x)\rangle\rangle(\gamma_1) \bullet \{\gamma_2\}\uparrow^{\{x\}}$, as required, since $\mathsf{modif}(\mathsf{new}(C_i,x)) = \{x\}$.

- $\mathsf{R} = \mathsf{delete}(C_i, x)$: we distinguish the following cases:

  - if $\nu(x) \in C_i^{\sigma_1}$, we compute $\langle\langle \mathsf{delete}(C_i,x)\rangle\rangle(\gamma_1 \bullet \gamma_2) =$

    $\{(\langle C_1^{\sigma_1 \uplus \sigma_2}, \ldots, C_i^{\sigma_1 \uplus \sigma_2} \setminus \{\nu(x)\}, \ldots, I_1^{\sigma_1 \uplus \sigma_2}, \ldots, I_m^{\sigma_1 \uplus \sigma_2}\rangle, \nu, \rho)\} =$
    $\{(\langle C_1^{\sigma_1}, \ldots, C_i^{\sigma_1} \setminus \{\nu(x)\}, \ldots, I_1^{\sigma_1}, \ldots, I_m^{\sigma_1}\rangle, \nu, \rho)\} \bullet \{\gamma_2\} =$
    $\langle\langle \mathsf{delete}(C_i,x)\rangle\rangle(\gamma_1) \bullet \{\gamma_2\}\uparrow^{\emptyset}$, as required, since $\mathsf{modif}(\mathsf{delete}(C_i,x)) = \emptyset$.

  - else $\nu(x) \notin C_i^{\sigma_1}$ and $\langle\langle \mathsf{delete}(C_i,x)\rangle\rangle(\gamma_1) = \top$, thus we obtain:

    $\langle\langle \mathsf{delete}(C_i,x)\rangle\rangle(\gamma_1 \bullet \gamma_2) \subseteq \top = \top \bullet \{\gamma_2\}\uparrow^{\emptyset} = \langle\langle \mathsf{delete}(C_i,x)\rangle\rangle(\gamma_1) \bullet \{\gamma_2\}\uparrow^{\emptyset}$

    as required, since $\mathsf{modif}(\mathsf{delete}(C_i,x)) = \emptyset$.

- $\mathsf{R} = \mathsf{connect}(I_j, x_1, \ldots, x_{\#(I_j)})$: we compute $\langle\langle \mathsf{connect}(I_j,x_1,\ldots,x_{\#(I_j)})\rangle\rangle(\gamma_1 \bullet \gamma_2) =$

  $\{(\langle C_1^{\sigma_1 \uplus \sigma_2}, \ldots, C_n^{\sigma_1 \uplus \sigma_2}, I_1^{\sigma_1 \uplus \sigma_2}, \ldots, I_j^{\sigma_1 \uplus \sigma_2} \cup \{\langle \nu(x_1), \ldots, \nu(x_{\#(I_j)})\rangle\}, \ldots, I_m^{\sigma_1 \uplus \sigma_2}\rangle\} =$
  $\{(\langle C_1^{\sigma_1}, \ldots, C_n^{\sigma_1}, I_1^{\sigma_1}, \ldots, I_j^{\sigma_1} \cup \{\langle \nu(x_1), \ldots, \nu(x_{\#(I_j)})\rangle\}, \ldots, I_m^{\sigma_1}\rangle\} \bullet \{\gamma_2\} =$
  $\langle\langle \mathsf{connect}(I_j,x_1,\ldots,x_{\#(I_j)})\rangle\rangle(\gamma_1) \bullet \{\gamma_2\}\uparrow^{\emptyset}$, as required, since $\mathsf{modif}(\mathsf{connect}(I_j,x_1,\ldots,x_{\#(I_j)})) = \emptyset$.

- $\mathsf{R} = \mathsf{disconnect}(I_j, x_1, \ldots, x_{\#(I_j)})$: we distinguish the following cases:

  - if $\langle \nu(x_1), \ldots, \nu(x_{\#(I_j)})\rangle \in I_j^{\sigma_1}$, we compute $\langle\langle \mathsf{disconnect}(I_j,x_1,\ldots,x_{\#(I_j)})\rangle\rangle(\gamma_1 \bullet \gamma_2) =$

    $\{(\langle C_1^{\sigma_1 \uplus \sigma_2}, \ldots, C_n^{\sigma_1 \uplus \sigma_2}, I_1^{\sigma_1 \uplus \sigma_2}, \ldots, I_j^{\sigma_1 \uplus \sigma_2} \cup \{\langle \nu(x_1), \ldots, \nu(x_{\#(I_j)})\rangle\}, \ldots, I_m^{\sigma_1 \uplus \sigma_2}\rangle, \nu, \rho)\} =$
    $\{(\langle C_1^{\sigma_1}, \ldots, C_n^{\sigma_1}, I_1^{\sigma_1}, \ldots, I_j^{\sigma_1} \cup \{\langle \nu(x_1), \ldots, \nu(x_{\#(I_j)})\rangle\}, \ldots, I_m^{\sigma_1}\rangle, \nu, \rho)\} \bullet \{\gamma_2\} =$
    $\{(\langle C_1^{\sigma_1}, \ldots, C_n^{\sigma_1}, I_1^{\sigma_1}, \ldots, I_j^{\sigma_1} \cup \{\langle \nu(x_1), \ldots, \nu(x_{\#(I_j)})\rangle\}, \ldots, I_m^{\sigma_1}\rangle, \nu, \rho)\} \bullet \{\gamma_2\}\uparrow^{\emptyset}$
    as required, since $\mathsf{modif}(\mathsf{disconnect}(I_j,x_1,\ldots,x_{\#(I_j)})) = \emptyset$.

  - else $\langle \nu(x_1), \ldots, \nu(x_{\#(I_j)})\rangle \notin I_j^{\sigma_1}$ and $\langle\langle \mathsf{disconnect}(I_j,x_1,\ldots,x_{\#(I_j)})\rangle\rangle(\gamma_1) = \top$, thus:

    $\langle\langle \mathsf{disconnect}(I_j,x_1,\ldots,x_{\#(I_j)})\rangle\rangle(\gamma_1 \bullet \gamma_2) \subseteq \top = \top \bullet \{\gamma_2\}\uparrow^{\emptyset} =$
    $\langle\langle \mathsf{disconnect}(I_j,x_1,\ldots,x_{\#(I_j)})\rangle\rangle(\gamma_1) \bullet \{\gamma_2\}\uparrow^{\emptyset}$

    as required, since $\mathsf{modif}(\mathsf{disconnect}(I_j,x_1,\ldots,x_{\#(I_j)})) = \emptyset$.

- $\mathsf{R} = \mathsf{skip}$: this case is a trivial check.

For the inductive step, we check the following points:

- $\mathsf{R} = \mathsf{R}_1 + \mathsf{R}_2$: we compute $\langle\langle \mathsf{R}_1 + \mathsf{R}_2 \rangle\rangle(\gamma_1 \bullet \gamma_2) =$

  $\langle\langle \mathsf{R}_1 \rangle\rangle(\gamma_1 \bullet \gamma_2) \cup \langle\langle \mathsf{R}_2 \rangle\rangle(\gamma_1 \bullet \gamma_2) \subseteq$ [by the inductive hypothesis]
  $\langle\langle \mathsf{R}_1 \rangle\rangle(\gamma_1) \bullet \{\gamma_2\}\uparrow^{\mathsf{modif}(\mathsf{R}_1)} \cup \langle\langle \mathsf{R}_2 \rangle\rangle(\gamma_1) \bullet \{\gamma_2\}\uparrow^{\mathsf{modif}(\mathsf{R}_2)} \subseteq$ [$\mathsf{modif}(\mathsf{R}_1 + \mathsf{R}_2) = \mathsf{modif}(\mathsf{R}_1) \cup \mathsf{modif}(\mathsf{R}_2)$]
  $\langle\langle \mathsf{R}_1 \rangle\rangle(\gamma_1) \bullet \{\gamma_2\}\uparrow^{\mathsf{modif}(\mathsf{R}_1+\mathsf{R}_2)} \cup \langle\langle \mathsf{R}_2 \rangle\rangle(\gamma_1) \bullet \{\gamma_2\}\uparrow^{\mathsf{modif}(\mathsf{R}_1+\mathsf{R}_2)} =$
  $\big(\langle\langle \mathsf{R}_1 \rangle\rangle(\gamma_1) \cup \langle\langle \mathsf{R}_2 \rangle\rangle(\gamma_1)\big) \bullet \{\gamma_2\}\uparrow^{\mathsf{modif}(\mathsf{R}_1+\mathsf{R}_2)} = \langle\langle \mathsf{R}_1 + \mathsf{R}_2 \rangle\rangle(\gamma_1) \bullet \{\gamma_2\}\uparrow^{\mathsf{modif}(\mathsf{R}_1+\mathsf{R}_2)}$

- $\mathsf{R} = (\mathsf{with}\ \mathbf{x} : \pi\ \mathsf{do}\ \mathsf{R}_1\ \mathsf{od})$, where π is a pure formula: we distinguish the cases

  - if $\gamma_1 \bullet \gamma_2 \models \pi$, we compute

    $$\begin{array}{lll} \langle\langle \mathsf{with}\ \mathbf{x} : \pi\ \mathsf{do}\ \mathsf{R}_1\ \mathsf{od} \rangle\rangle(\gamma_1 \bullet \gamma_2) & \subseteq & \langle\langle \mathsf{R}_1 \rangle\rangle(\{\gamma_1 \bullet \gamma_2\}\uparrow^{\mathbf{x}}) \\ & \subseteq & \langle\langle \mathsf{R}_1 \rangle\rangle(\gamma_1) \bullet \{\gamma_2\}\uparrow^{\mathbf{x} \cup \mathsf{modif}(\mathsf{R}_1)} \\ & = & \langle\langle \mathsf{R}_1 \rangle\rangle(\gamma_1) \bullet \{\gamma_2\}\uparrow^{\mathsf{modif}(\mathsf{when}\ \pi\ \mathsf{do}\ \mathsf{R}_1)} \end{array}$$

  - else $\gamma_1 \bullet \gamma_2 \not\models \pi$ and

    $$\langle\langle \mathsf{with}\ \mathbf{x} : \pi\ \mathsf{do}\ \mathsf{R}_1\ \mathsf{od} \rangle\rangle(\gamma_1 \bullet \gamma_2) = \emptyset \subseteq \langle\langle \mathsf{R}_1 \rangle\rangle(\gamma_1) \bullet \{\gamma_2\}\uparrow^{\mathsf{modif}(\mathsf{when}\ \pi\ \mathsf{do}\ \mathsf{R}_1)}$$

□

Theorem 1 *Given a SID $\mathcal{D}$, for any triple $\{\phi\}\ \mathsf{R}\ \{\psi\}$, if $\vdash \{\phi\}\ \mathsf{R}\ \{\psi\}$ then $\models_{\mathcal{D}} \{\phi\}\ \mathsf{R}\ \{\psi\}$.*

Proof. We prove that the inference rules in Fig. 3 are sound. For the axioms, soundness follows from Lemma 1. The rules for the composite programs are proved below by a case split on the syntax of the program from the conclusion $\{\phi\}\ \mathsf{R}\ \{\psi\}$, assuming that $\models_{\mathcal{D}} \{\phi_i\}\ \mathsf{R}_i\ \{\psi_i\}$, for each premiss $\{\phi_i\}\ \mathsf{R}_i\ \{\psi_i\}$ of the rule:

- $\mathsf{R} = \mathsf{with}\ x_1, \ldots, x_k : \varphi\ \mathsf{do}\ \mathsf{R}\ \mathsf{od}$: let $(\sigma, \nu, \rho) \in [\![\phi]\!]_{\mathcal{D}}$ be a configuration and distinguish the cases

  - if $(\sigma, \nu[x_1 \leftarrow u_1, \ldots, x_k \leftarrow u_k], \rho) \models \varphi * \mathsf{true}$, for some $u_1, \ldots, u_k \in \mathbb{U}$, then we obtain $(\sigma, \nu[x_1 \leftarrow u_1, \ldots, x_k \leftarrow u_k], \rho) \models_{\mathcal{D}} \phi \wedge (\varphi * \mathsf{true})$, because $\mathrm{fv}(\phi) \cap \{x_1, \ldots, x_k\} = \emptyset$. Then $\langle\langle \mathsf{with}\ x_1, \ldots, x_k : \varphi\ \mathsf{do}\ \mathsf{R}\ \mathsf{od} \rangle\rangle(\sigma, \nu, \rho) = \langle\langle \mathsf{R} \rangle\rangle(\sigma, \nu[x_1 \leftarrow u_1, \ldots, x_k \leftarrow u_k], \rho) \subseteq [\![\psi]\!]_{\mathcal{D}} \subseteq [\![\exists \mathbf{x}\ .\ \psi]\!]_{\mathcal{D}}$ follows from the premiss of the rule.
  - othewise, we have $(\sigma, \nu, \rho) \models \forall x_1 \ldots \forall x_k\ .\ \neg(\varphi * \mathsf{true})$ and $\langle\langle \mathsf{with}\ \mathbf{x} : \varphi\ \mathsf{do}\ \mathsf{R}\ \mathsf{od} \rangle\rangle(\sigma, \nu, \rho) = \emptyset \subseteq [\![\exists \mathbf{x}\ .\ \psi]\!]_{\mathcal{D}}$ follows.

- the cases $\mathsf{R} = \mathsf{R}_1; \mathsf{R}_2$, $\mathsf{R} = \mathsf{R}_1 + \mathsf{R}_2$ and $\mathsf{R} = \mathsf{R}_1^*$ are simple checks using the operational semantics rules from Fig. 2.

Concerning the structural rules, we show only the soundness of the frame rule below; the other rules are simple checks, left to the reader. Let $\gamma \in [\![\phi * \varphi]\!]_{\mathcal{D}}$ be a configuration. By the semantics of $*$, there exists $\gamma_1 \in [\![\phi]\!]_{\mathcal{D}}$ and $\gamma_2 \in [\![\varphi]\!]_{\mathcal{D}}$, such that $\gamma = \gamma_1 \bullet \gamma_2$. Since $\mathsf{R} \in \mathfrak{L}$, by Lemma 2, we obtain $\langle\langle \mathsf{R} \rangle\rangle(\gamma_1 \bullet \gamma_2) \subseteq \langle\langle \mathsf{R} \rangle\rangle(\gamma_1) \bullet \{\gamma_2\}\uparrow^{\mathsf{modif}(\mathsf{R})}$. Since $\gamma_1 \in [\![\phi]\!]_{\mathcal{D}}$, by the hypothesis on the premiss we obtain $\langle\langle \mathsf{R} \rangle\rangle(\gamma_1) \subseteq [\![\psi]\!]_{\mathcal{D}}$. Moreover, since $\gamma_2 \in [\![\varphi]\!]_{\mathcal{D}}$ and $\mathsf{modif}(\mathsf{R}) \cap \mathrm{fv}(\varphi)$, we obtain $\{\gamma_2\}\uparrow^{\mathsf{modif}(\mathsf{R})} \subseteq [\![\varphi]\!]_{\mathcal{D}}$, leading to $\langle\langle \mathsf{R} \rangle\rangle(\gamma) \subseteq [\![\psi * \varphi]\!]_{\mathcal{D}}$, as required. □

C Proof from Section 5

Proposition 2 *If* $\models_{\mathcal{D}} \eta \rhd \{\{\phi\}\}\ \Sigma[\phi]^*\ \{\{\psi\}\}$ *then* $\mathfrak{h}([\![\phi]\!]_{\mathcal{D}}) \subseteq [\![\psi]\!]_{\mathcal{D}}$.

Proof. Let $\gamma = (\sigma, \nu, \rho)$ be a $\mathcal{D}$-model of ϕ, i.e. we have $\gamma \in [\![\phi]\!]_{\mathcal{D}}$. It is sufficient to prove that $\mathfrak{h}(\gamma) \subseteq \bigcup\{\mathfrak{o}[w](\gamma) \mid w \in \langle\langle \Sigma[\phi]^* \rangle\rangle^{\gamma}\}$, because $\mathfrak{o}[w](\gamma) \subseteq [\![\psi]\!]_{\mathcal{D}}$ for each $w \in \langle\langle \Sigma[\phi]^* \rangle\rangle^{\gamma}$, by the hypothesis $\models_{\mathcal{D}} \eta \rhd \{\{\phi\}\}\ \Sigma[\phi]^*\ \{\{\psi\}\}$ (Def. 9). Let $\gamma' = (\sigma, \nu, \rho') \in \mathfrak{h}(\gamma)$ be a configuration. Then there exists a finite sequence $w = (I_1, \mathbf{u}_1), \ldots, (I_k, \mathbf{u}_k)$, such that $\gamma' \in \mathfrak{c}[w](\gamma)$, where $\mathfrak{c}[w] \stackrel{\text{def}}{=} \mathfrak{c}[I_k, \mathbf{u}_k] \circ \ldots \circ \mathfrak{c}[I_1, \mathbf{u}_1]$, by Def. 2. Note that $\mathfrak{c}[I, \mathbf{u}] \subseteq \mathfrak{o}[I, \mathbf{u}]$ pointwise, for each interaction type I and tuple $\mathbf{u} \in \mathbb{U}^{\#(I)}$, by Def. 2 and 8; indeed the two definitions are identical, except for point (1) of Def. 2, which is stronger than point (1) of Def. 8. We are thus left with proving $w \in \langle\langle \Sigma[\phi]^* \rangle\rangle^{\gamma}$, or equivalently, $(I_j, \mathbf{u}_j) \in \langle\langle \Sigma[\phi] \rangle\rangle^{\gamma}$, for each $j \in [1, k]$. Since $\gamma' \in \mathfrak{c}[w](\gamma)$, by Def. 2, we have $\mathbf{u}_j \in I_j^{\sigma}$, for each $j \in [1, k]$, because the structure σ cannot be changed by a state change $\mathfrak{c}[I_j, \mathbf{u}_j]$. Since, moreover, $\gamma \models_{\mathcal{D}} \phi$ by the choice of γ, we obtain $(I_j, \mathbf{u}_j) \in \langle\langle \Sigma[\phi] \rangle\rangle^{\gamma}$, for each $j \in [1, k]$. $\square$

Lemma 3 *Given a proof tree T, each node in T is labeled with a distinctive havoc triple.*

Proof. The proof goes by induction on the structure of the proof tree. For the base case, the tree consists of a single root node and let $\eta \rhd \{\{\phi\}\}\ \mathsf{L}\ \{\{\psi\}\}$ be the label of the root node. By Assumption 2, ϕ is a symbolic configuration and $\eta = \{\Sigma[\alpha_1], \ldots, \Sigma[\alpha_k]\}$, where $\mathrm{atoms}(\phi) = \{\alpha_1, \ldots, \alpha_k\}$ is the set of interaction and predicate atoms from ϕ. Let γ be a $\mathcal{D}$-model of ϕ, hence there exist configurations $\gamma_0, \gamma_1, \ldots, \gamma_k$, such that $\gamma = \bullet_{i=0}^{k} \gamma_i$ and $\gamma_i \models_{\mathcal{D}} \alpha_i$, for all $i \in [1, k]$. Because the composition $\gamma_i \bullet \gamma_j$ is defined, we obtain that $\langle\langle \Sigma[\alpha_i] \rangle\rangle^{\gamma_i} \cap \langle\langle \Sigma[\alpha_j] \rangle\rangle^{\gamma_j} = \emptyset$, for all $i \neq j \in [1, k]$. Moreover, since each formula $\alpha_i \in \mathrm{atoms}(\phi)$ is precise on $[\![\phi]\!]_{\mathcal{D}}$, by Assumption 1, we have $\langle\langle \Sigma[\alpha_i] \rangle\rangle^{\gamma_i} = \langle\langle \Sigma[\alpha_i] \rangle\rangle^{\gamma}$, hence $\langle\langle \Sigma[\alpha_i] \rangle\rangle^{\gamma} \cap \langle\langle \Sigma[\alpha_j] \rangle\rangle^{\gamma} = \emptyset$, for all $i \neq j \in [1, k]$. For the inductive step, we distinguish the cases below, based on the type of the inference rule that expands the root:

- (I) Let $\eta \rhd \{\{\phi * I(x_1, \ldots, x_{\#(I)})\}\}\ \mathsf{L}\ \{\{\psi * I(x_1, \ldots, x_{\#(I)})\}\}$ be the label of the root and let $\gamma \in [\![\phi * I(x_1, \ldots, x_{\#(I)})]\!]_{\mathcal{D}}$ be a configuration. Then there exists configurations γ_0 and γ_1, such that $\gamma = \gamma_0 \bullet \gamma_1$, $\gamma_0 \models_{\mathcal{D}} \phi$ and $\gamma_1 \models I(x_1, \ldots, x_{\#(I)})$. By Assumption 2, we have $\eta = \Sigma[\phi] \cup \{\Sigma[I(x_1, \ldots, x_{\#(I)})]\}$. By the inductive hypothesis, the premiss $\eta \setminus \{I(x_1, \ldots, x_{\#(I)})\} \rhd \{\{\phi\}\}\ \mathsf{L}\ \{\{\psi\}\}$ of the rule is distinctive, hence the interpretations of the atoms in the environment $\{\langle\langle \Sigma[\alpha] \rangle\rangle^{\gamma_0} \mid \alpha \in \mathrm{atoms}(\phi)\}$ are pairwise disjoint. Since each predicate atom $\alpha \in \mathrm{atoms}(\phi)$ is precise on $[\![\phi]\!]_{\mathcal{D}}$, by Assumption 1, the sets $\{\langle\langle \Sigma[\alpha] \rangle\rangle^{\gamma} \mid \alpha \in \mathrm{atoms}(\phi)\}$ are pairwise disjoint. Since $I(x_1, \ldots, x_{\#(I)})$ is precise on Γ, we obtain that $\langle\langle I(x_1, \ldots, x_{\#(I)}) \rangle\rangle^{\gamma_1} = \langle\langle I(x_1, \ldots, x_{\#(I)}) \rangle\rangle^{\gamma}$ and, since $\gamma = \gamma_0 \bullet \gamma_1$, the set $\langle\langle I(x_1, \ldots, x_{\#(I)}) \rangle\rangle^{\gamma}$ is disjoint from the sets $\{\langle\langle \Sigma[\alpha] \rangle\rangle^{\gamma} \mid \alpha \in \mathrm{atoms}(\phi)\}$, thus $\eta \rhd \{\{\phi * I(x_1, \ldots, x_{\#(I)})\}\}\ \mathsf{L}\ \{\{\psi * I(x_1, \ldots, x_{\#(I)})\}\}$ is distinctive.

- (E) Let $\eta \rhd \{\{\phi\}\}\ \mathsf{L}\ \{\{\psi\}\}$ be the label of the root and let $\gamma \in [\![\phi]\!]_{\mathcal{D}}$ be a configuration. By Assumption 2, we have $\eta = \Sigma[\phi]$ and let $I(x_1, \ldots, x_{\#(I)})$ be an interaction atom, such that $\phi \ddagger I(x_1, \ldots, x_{\#(I)})$. Let γ' be any model of $I(x_1, \ldots, x_{\#(I)})$. By $\phi \ddagger$

$I(x_1,\ldots,x_{\#(I)})$, we have $I(x_1,\ldots,x_{\#(I)}) \notin \eta$ and, moreover, the composition $\gamma \bullet \gamma'$ is defined, thus $\gamma \bullet \gamma' \in [\![\phi * I(x_1,\ldots,x_{\#(I)})]\!]_{\mathcal{D}}$. By the inductive hypothesis, the havoc triple $\eta \cup \{\Sigma[I(x_1,\ldots,x_{\#(I)})]\} \triangleright \{\{\phi * I(x_1,\ldots,x_{\#(I)})\}\} \mathsf{L} \{\{\psi * I(x_1,\ldots,x_{\#(I)})\}\}$ is distinctive, hence $\langle\langle \Sigma[\alpha_1] \rangle\rangle^{\gamma \bullet \gamma'} \cap \langle\langle \Sigma[\alpha_2] \rangle\rangle^{\gamma \bullet \gamma'} = \emptyset$, for all $\Sigma[\alpha_1], \Sigma[\alpha_2] \in \eta$. Since $\gamma' \models I(x_1,\ldots,x_{\#(I)})$, $\eta = \Sigma[\phi]$ and $\phi \ddagger I$, we obtain $\langle\langle \Sigma[\alpha] \rangle\rangle^{\gamma \bullet \gamma'} = \langle\langle \Sigma[\alpha] \rangle\rangle^{\gamma}$, for all $\Sigma[\alpha] \in \eta$, thus $\eta \triangleright \{\{\phi\}\} \mathsf{L} \{\{\psi\}\}$ is distinctive.

- ($\bowtie$) Let $\eta_1 \cup \eta_2 \triangleright \{\{\phi_1 * \phi_2\}\} \mathsf{L}_1 \bowtie_{\eta_1,\eta_2} \mathsf{L}_2 \{\{\psi_1 * \psi_2\}\}$ be the label of the root, $\eta_i = \Sigma[\phi_i * \mathcal{F}(\phi_i, \phi_{3-i})]$, for $i = 1,2$, and let γ be a $\mathcal{D}$-model of the precondition of this havoc triple. Then there exists two configurations γ_1, γ_2, such that $\gamma = \gamma_1 \bullet \gamma_2$ and $\gamma_i \models_{\mathcal{D}} \phi_i$, for $i = 1,2$. By Assumption 2, we have $\eta_1 \cup \eta_2 = \Sigma[\phi_1] \cup \Sigma[\phi_2]$. Let γ'_i be a structure, such that $\gamma'_i \sqsubseteq \gamma_{3-i}$ and $\gamma'_i \models \mathcal{F}(\phi_i, \phi_{3-i})$, for $i = 1,2$. By the definition of $\mathcal{F}(\phi_i, \phi_{3-i})$, as separated conjunction of interaction atoms from ϕ_{3-i}, these substructures exist, and moreover, because each interaction atom is precise on Γ, they are unique. Then we have $\gamma_i \bullet \gamma'_i \models_{\mathcal{D}} \phi_i * \mathcal{F}(\phi_i, \phi_{3-i})$, for $i = 1,2$. By the inductive hypothesis, since each havoc triple $\eta_i \triangleright \{\{\phi_i * \mathcal{F}(\phi_i, \phi_{3-i})\}\} \mathsf{L}_i \{\{\psi_i * \mathcal{F}(\phi_i, \phi_{3-i})\}\}$ is distinctive, the sets $\{\langle\langle \Sigma[\alpha] \rangle\rangle^{\gamma_i \bullet \gamma'_i} \mid \alpha \in \mathrm{atoms}(\phi_i)\}$ are pairwise disjoint, for $i = 1,2$. By Assumption 1, each predicate atom $\alpha \in \mathrm{atoms}(\phi_1 * \phi_2)$ is precise on $[\![\phi_1 * \phi_2]\!]_{\mathcal{D}}$, hence the sets $\{\langle\langle \Sigma[\alpha] \rangle\rangle^{\gamma} \mid \alpha \in \mathrm{atoms}(\phi_i)\}$ are pairwise disjoint as well, for $i = 1,2$. Moreover, since the configurations γ_1 and γ_2 share no interactions, the havoc triple $\eta_1 \cup \eta_2 \triangleright \{\{\phi_1 * \phi_2\}\} \mathsf{L}_1 \bowtie_{\eta_1,\eta_2} \mathsf{L}_2 \{\{\psi_1 * \psi_2\}\}$ is distinctive.

- (LU) Let $\eta \triangleright \{\{\phi * \mathsf{A}(x_1,\ldots,x_{\#(\mathsf{A})})\}\} \mathsf{L} \{\{\psi\}\}$ be the label of the root let γ be a $\mathcal{D}$-model of the precondition of this havoc triple. Then there exist configurations $\gamma_0 = (\sigma_0, \nu, \rho)$ and $\gamma_1 = (\sigma_1, \nu, \rho)$, such that $\gamma = \gamma_0 \bullet \gamma_1$, $\gamma_0 \models_{\mathcal{D}} \phi$ and $\gamma_1 \models_{\mathcal{D}} \mathsf{A}(x_1,\ldots,x_{\#(\mathsf{A})})$. By Assumption 2, we have $\eta = \Sigma[\phi] \cup \{\Sigma[\mathsf{A}(x_1,\ldots,x_{\#(\mathsf{A})})]\}$. By the inductive hypothesis, each of the premisses $\eta' \triangleright \{\{\phi * \varphi\}\} \mathsf{L}' \{\{\psi\}\}$ is distinctive, where:

  - $\mathsf{A}(x_1,\ldots,x_{\#(\mathsf{A})}) \Leftarrow_{\mathcal{D}} \exists z_1 \ldots \exists z_h \,.\, \varphi$ is an unfolding step and φ is a symbolic configuration, such that $(\sigma_1, \nu[z_1 \leftarrow u_1, \ldots, z_h \leftarrow u_h], \rho) \models_{\mathcal{D}} \varphi$, for some indices $u_1,\ldots,u_h \in \mathbb{U}$,
  - $\eta' = (\eta \setminus \{\Sigma[\mathsf{A}(x_1,\ldots,x_{\#(\mathsf{A})})]\}) \cup \Sigma[\varphi]$.

  Because we assumed that $\{z_1,\ldots,z_h\} \cap \mathrm{fv}(\phi) = \emptyset$ (if necessary, by an α-renaming of existentially quantified variables), we have $\gamma'_0 \models_{\mathcal{D}} \phi$ and $\gamma'_1 \models_{\mathcal{D}} \mathsf{A}(x_1,\ldots,x_{\#(\mathsf{A})})$, where $\gamma'_0 \stackrel{\mathsf{def}}{=} (\sigma_0, \nu[z_1 \leftarrow u_1, \ldots, z_h \leftarrow u_h], \rho)$ and $\gamma'_1 \stackrel{\mathsf{def}}{=} (\sigma_1, \nu[z_1 \leftarrow u_1, \ldots, z_h \leftarrow u_h], \rho)$, thus $\gamma' \models_{\mathcal{D}} \phi * \mathsf{A}(x_1,\ldots,x_{\#(\mathsf{A})})$, where $\gamma' = \gamma'_0 \bullet \gamma'_1$. Because $\eta' \triangleright \{\{\phi * \varphi\}\} \mathsf{L}' \{\{\psi\}\}$ is distinctive and $\gamma' \models_{\mathcal{D}} \phi * \varphi$, we obtain $\langle\langle \Sigma[\alpha_1] \rangle\rangle^{\gamma'} \cap \langle\langle \Sigma[\alpha_2] \rangle\rangle^{\gamma'} = \emptyset$, for all $\alpha_1 \in \mathrm{atoms}(\phi)$ and $\alpha_2 \in \mathrm{atoms}(\phi) \cup \mathrm{atoms}(\varphi)$. By Assumption 1, $\mathsf{A}(x_1,\ldots,x_{\#(\mathsf{A})})$ is precise on $[\![\phi * \mathsf{A}(x_1,\ldots,x_{\#(\mathsf{A})})]\!]_{\mathcal{D}}$, hence $\langle\langle \Sigma[\mathsf{A}(x_1,\ldots,x_{\#(\mathsf{A})})] \rangle\rangle^{\gamma'} = \bigcup_{\alpha \in \mathrm{atoms}(\varphi)} \langle\langle \Sigma[\alpha] \rangle\rangle^{\gamma'}$. Since $\langle\langle \Sigma[\alpha] \rangle\rangle^{\gamma'} = \langle\langle \Sigma[\alpha] \rangle\rangle^{\gamma}$, for each $\alpha \in \mathrm{atoms}(\phi) \cup \{\mathsf{A}(x_1,\ldots,x_{\#(\mathsf{A})})\}$, we obtain that $\langle\langle \Sigma[\alpha_1] \rangle\rangle^{\gamma} \cap \langle\langle \Sigma[\alpha_2] \rangle\rangle^{\gamma} = \emptyset$, for all $\alpha_1, \alpha_2 \in \mathrm{atoms}(\phi) \cup \{\mathsf{A}(x_1,\ldots,x_{\#(\mathsf{A})})\}$, i.e. $\eta \triangleright \{\{\phi * \mathsf{A}(x_1,\ldots,x_{\#(\mathsf{A})})\}\} \mathsf{L} \{\{\psi\}\}$ is distinctive.

- ($\vee$) Let $\eta \rhd \{\{\bigvee_{i=1}^{k} \phi \wedge \delta_i\}\} \mathsf{L} \{\{\bigvee_{i=1}^{k} \psi_i\}\}$ be the label of the root and let γ be a $\mathcal{D}$-model of the precondition of this triple. Then $\gamma \models_{\mathcal{D}} \phi \wedge \delta_i$, for some $i \in [1,k]$. By the inductive hypothesis, the triple $\eta \rhd \{\{\phi \wedge \delta_i\}\} \mathsf{L} \{\{\psi_i\}\}$ is distinctive, hence $\langle\langle \Sigma[\alpha_1] \rangle\rangle^{\gamma} \cap \langle\langle \Sigma[\alpha_1] \rangle\rangle^{\gamma} = \emptyset$, for all $\alpha_1, \alpha_2 \in \mathrm{atoms}(\phi)$. Since $\mathrm{atoms}(\phi) = \mathrm{atoms}(\bigvee_{i=1}^{k} \phi \wedge \delta_i)$, we obtain that $\eta \rhd \{\{\bigvee_{i=1}^{k} \phi \wedge \delta_i\}\} \mathsf{L} \{\{\bigvee_{i=1}^{k} \psi_i\}\}$ is distinctive.
- ($\wedge$) and ($\cdot$): these cases are similar to ($\vee$).
- (C), ($*$), ($\cup$) and ($\subset$): these cases are trivial, because the precondition and the environment does not change between the conclusion and the premisses of these rules. $\square$

Theorem 2 *If* $\Vdash \eta \rhd \{\{\phi\}\} \mathsf{L} \{\{\psi\}\}$ *then* $\models_{\mathcal{D}} \eta \rhd \{\{\phi\}\} \mathsf{L} \{\{\psi\}\}$.

Proof. For each axiom and inference rule in Fig. 5, with premisses $\eta_i \rhd \{\{\phi_i\}\} \mathsf{L}_i \{\{\psi_i\}\}$, for $i = 1, \ldots, k$, $k \geq 0$, and conclusion $\eta \rhd \{\{\phi\}\} \mathsf{L} \{\{\psi\}\}$, we prove that:

$$(\star) \ \models_{\mathcal{D}} \eta \rhd \{\{\phi\}\} \mathsf{L} \{\{\psi\}\}, \text{ if } \models_{\mathcal{D}} \eta_i \rhd \{\{\phi_i\}\} \mathsf{L}_i \{\{\psi_i\}\}, \text{ for all } i \in [1,k]$$

Let us show first that ($\star$) is a sufficient condition. If $\Vdash \eta \rhd \{\{\phi\}\} \mathsf{L} \{\{\psi\}\}$ then there exists a cyclic proof whose root is labeled by $\eta \rhd \{\{\phi\}\} \mathsf{L} \{\{\psi\}\}$ and we apply the principle of infinite descent to prove that $\models_{\mathcal{D}} \eta \rhd \{\{\phi\}\} \mathsf{L} \{\{\psi\}\}$. Suppose, for a contradiction, that this is not the case and there exists a configuration $\gamma_0 \in [\![\phi]\!]_{\mathcal{D}}$ and a word $w_0 \in \langle\langle \mathsf{L} \rangle\rangle^{\gamma}$, such that $\mathfrak{o}[w_0](\gamma_0) \not\subseteq [\![\psi]\!]_{\mathcal{D}}$. Assuming that ($\star$) holds, each invalid node, with label $\eta_i \rhd \{\{\phi_i\}\} \mathsf{L}_i \{\{\psi_i\}\}$ and counterexample $\gamma_i = (\sigma_i, \nu_i, \rho_i)$, not on the frontier of the proof tree, has a successor, whose label is invalid, for all $i \geq 0$. Let $\mathrm{preds}(\phi_i) = \{\mathsf{A}_1^i(\mathbf{x}_1^i), \ldots, \mathsf{A}_{k_i}^i(\mathbf{x}_{k_i}^i)\}$ be the set of predicate atoms from ϕ_i, for each $i \geq 0$. Consequently, there exists a set of configurations $\Gamma_i = \{\gamma_0^i, \ldots, \gamma_{k_i}^i\}$, such that $\gamma_i = \gamma_0^i \bullet \ldots \bullet \gamma_{k_i}^i$ and $\gamma_j^i \models_{\mathcal{D}} \mathsf{A}_j^i(\mathbf{x}_j^i)$, for all $j \in [1,k_i]$ and all $i \geq 0$.

Fact 1 *For each* $i \geq 0$, *either* $\Gamma_{i+1} \subseteq \Gamma_i$ *or there exists* $j \in [1,k_i]$, *such that* $\Gamma_{i+1} = \big(\Gamma_i \setminus \{\gamma_j^i\}\big) \cup \{\gamma' \in [\![\mathsf{A}(x_1, \ldots, x_{\#(\mathsf{A})})]\!]_{\mathcal{D}} \mid \gamma' \sqsubseteq \gamma_j^i, \ \mathsf{A}(x_1, \ldots, x_{\#(\mathsf{A})}) \in \mathrm{preds}(\varphi_j^i)\}$, *where* $\mathsf{A}_j^i(\mathbf{x}_j^i) \Leftarrow_{\mathcal{D}} \exists \mathbf{z}_j^i \ . \ \varphi_j^i$ *is an unfolding step and* $\varphi_j^i \in \mathbb{S}$ *is a symbolic configuration.*

Proof. By inspection of the inference rules in Fig. 5b-e. The only interesting cases are:

- ($\bowtie$) in this case Γ_{i+1} is a possibly strict subset of Γ_i, because the models of the preconditions from the premises are subconfigurations of the model of the precondition in the conclusion,
- (LU) in this case Γ_{i+1} is obtained by replacing an element γ_j^i from Γ_i with a set of configurations γ', such that $\gamma' \sqsubseteq \gamma_j^i$ and γ' is a model of a predicate atom from an unfolding of the predicate atom for which γ_j^i is a model. $\square$

For a configuration γ_j^i, we denote by $\mathfrak{n}(i,j)$ the minimum length of a complete unfolding $\mathsf{A}_j^i(\mathbf{x}_j^i) \Leftarrow_{\mathcal{D}}^{\circ} \varphi_j^i$, such that $\gamma_j^i \models \varphi_j^i$, taken among all such complete unfoldings. Let $\mathfrak{m}_i$ be the multiset of numbers $\mathfrak{n}(i,j)$, for all $j \in [1,k_i]$ and $i \geq 0$. By Fact 1, the

sequence of multisets $\mathfrak{m}_0, \mathfrak{m}_1, \ldots$ is such that either $\mathfrak{m}_i = \mathfrak{m}_{i+1}$ or $\mathfrak{m}_i \succ \mathfrak{m}_{i+1}$, where the Dershowitz-Manna multiset ordering $\prec$ is defined as $\mathfrak{m} \prec \mathfrak{m}'$ if and only if there exist two multisets X and Y, such that $X \neq \emptyset$, $X \subseteq \mathfrak{m}'$, $\mathfrak{m} = (\mathfrak{m}' \setminus X) \cup Y$, and for all $y \in Y$ there exists some $x \in X$, such that $y < x$. By the fact that the cyclic proof tree is a cyclic proof, the infinite path goes infinitely often via a node whose label is the conclusion of the application of (LU). Then the infinite sequence of multisets $\mathfrak{m}_0, \mathfrak{m}_1, \ldots$ contains a strictly decreasing subsequence in the multiset order, which contradicts the fact that $\prec$ is well-founded. We are left with proving ($\star$) for each type of axiom and inference rule in Fig. 5:

- (ε) For each configuration γ, we have $\langle\langle \varepsilon \rangle\rangle^{\gamma} = \{\varepsilon\}$ and $\mathfrak{o}[\varepsilon]$ is the identity (Def. 9).
- ($\dagger$) In any $\mathcal{D}$-model (σ, ν, ρ) of ϕ, the action $\mathfrak{o}[I, \langle \nu(x_1), \ldots, \nu(x_{\#(I)}) \rangle]$ is disabled, by the side condition $\phi \dagger I(x_1, \ldots, x_{\#(I)})$ (Def. 10).
- ($\bot$) Because the precondition has no models.
- (Σ) By Def. 8.
- (I) Let $\gamma = (\sigma, \nu, \rho) \in [\![\phi * I(x_1, \ldots, x_{\#(I)})]\!]_{\mathcal{D}}$ be a configuration. By Lemma 3, we have that $\eta \triangleright \{\{\phi * I(x_1, \ldots, x_{\#(I)})\}\} \sqsubset \{\{\psi * I(x_1, \ldots, x_{\#(I)})\}\}$ is distinctive. By the side condition $\Sigma[I(x_1, \ldots, x_{\#(I)})] \in \eta \setminus \mathrm{supp}(\sqsubset)$, it follows that $\langle\langle I(x_1, \ldots, x_{\#(I)}) \rangle\rangle^{\gamma}$ is disjoint from the interpretation $\langle\langle \Sigma[\alpha] \rangle\rangle^{\gamma}$ of any alphabet symbol $\Sigma[\alpha] \in \mathrm{supp}(\sqsubset)$, hence the interaction $(I, \langle \nu(x_1), \ldots, \nu(x_{\#(I)}) \rangle)$ does not occur in $\langle\langle \sqsubset \rangle\rangle^{\gamma}$. By the inductive hypothesis, we have $\models_{\mathcal{D}} \eta \triangleright \{\{\phi\}\} \sqsubset \{\{\psi\}\}$, which leads to the required $\models_{\mathcal{D}} \eta \triangleright \{\{\phi * I(x_1, \ldots, x_{\#(I)})\}\} \sqsubset \{\{\psi * I(x_1, \ldots, x_{\#(I)})\}\}$.
- (E) Let $\gamma = (\sigma, \nu, \rho) \in [\![\phi]\!]_{\mathcal{D}}$ be a configuration and $\omega \stackrel{\mathsf{def}}{=} \Sigma[\alpha_1] \cdot \ldots \cdot \Sigma[\alpha_k]$ be a finite concatenation of alphabet symbols from $\mathrm{supp}(\sqsubset)$. If $\alpha_i = I(x_1, \ldots, x_{\#(I)})$, for some $i \in [1, k]$, then we have $\langle\langle \Sigma[\alpha_i] \rangle\rangle^{\gamma} = \emptyset$, because of the side condition $\phi \ddagger I(x_1, \ldots, x_{\#(I)})$. Then $\mathfrak{o}[w](\gamma) = \emptyset \subseteq [\![\psi]\!]_{\mathcal{D}}$, for each $w \in \langle\langle \omega \rangle\rangle^{\gamma}$. Otherwise, if $I(x_1, \ldots, x_{\#(I)})$ does not occur on ω, then $\mathfrak{o}[w](\gamma) \subseteq [\![\psi]\!]_{\mathcal{D}}$, for each $w \in \langle\langle \sqsubset \rangle\rangle^{\gamma} \cap \langle\langle \omega \rangle\rangle^{\gamma}$, by the inductive hypothesis.
- ($\bowtie$) Let $\gamma = (\sigma, \nu, \rho) \in [\![\phi_1 * \phi_2]\!]_{\mathcal{D}}$ be a configuration. Then there exist configurations $\gamma_i = (\sigma_i, \nu, \rho) \in [\![\phi_i]\!]_{\mathcal{D}}$, for $i = 1, 2$, such that $\gamma = \gamma_1 \bullet \gamma_2$. Let γ_i' be configurations such that $\gamma_i' \sqsubseteq \gamma_{3-i}$ and $\gamma_i' \models \mathcal{F}(\phi_i, \phi_{3-i})$, for $i = 1, 2$. Because $\mathcal{F}(\phi_i, \phi_{3-i})$ is a separated conjunction of interaction atoms, each of which is precise on Γ, it follows that $\mathcal{F}(\phi_i, \phi_{3-i})$ is precise on Γ, thus γ_i' are unique, for $i = 1, 2$. Let $\gamma_i'' \stackrel{\mathsf{def}}{=} \gamma_i \bullet \gamma_{3-i}'$, for $i = 1, 2$. Moreover, since $\eta_i = \Sigma[\phi_i * \mathcal{F}(\phi_i, \phi_{3-i})]$, the only interactions in $\langle\langle \eta_i \rangle\rangle^{\gamma}$ are the ones in $\langle\langle \eta_i \rangle\rangle^{\gamma_i''}$, hence $\langle\langle \eta_i \rangle\rangle^{\gamma} = \langle\langle \eta_i \rangle\rangle^{\gamma_i''}$, for $i = 1, 2$. Let $w \in \langle\langle \sqsubset_1 \bowtie_{\eta_1, \eta_2} \sqsubset_2 \rangle\rangle^{\gamma}$ be a word. Then $w \downarrow_{\langle\langle \eta_i \rangle\rangle^{\gamma}} \in \langle\langle \sqsubset_i \rangle\rangle^{\gamma}$, for $i = 1, 2$. Because $\langle\langle \eta_i \rangle\rangle^{\gamma} = \langle\langle \eta_i \rangle\rangle^{\gamma_i''}$, we obtain $w \downarrow_{\langle\langle \eta_i \rangle\rangle^{\gamma}} = w \downarrow_{\langle\langle \eta_i \rangle\rangle^{\gamma_i''}} \in \langle\langle \sqsubset_i \rangle\rangle^{\gamma_i''}$, for $i = 1, 2$. Since, moreover, $\gamma_i'' \models_{\mathcal{D}} \phi_i * \mathcal{F}(\phi_i, \phi_{3-i})$, by the inductive hypothesis we obtain that $\mathfrak{o}[w \downarrow_{\langle\langle \eta_i \rangle\rangle^{\gamma_i''}}](\gamma_i'') \subseteq [\![\psi_i * \mathcal{F}(\phi_i, \phi_{3-i})]\!]_{\mathcal{D}}$, for $i = 1, 2$. We partition $w = w_1 w_1' w_1'' w_1''' \ldots w_k w_k' w_k'' w_k'''$, for some $k \geq 1$, in three types of (possibly empty) blocks, such that, for all $j \in [1, k]$, we have:

– $w_j \in \big(\langle\langle \eta_1 \rangle\rangle^{\gamma_1''} \setminus \langle\langle \eta_2 \rangle\rangle^{\gamma_2''}\big)^*$,

– $w_j', w_j''' \in \big(\langle\langle \eta_1 \rangle\rangle^{\gamma_1''} \cap \langle\langle \eta_2 \rangle\rangle^{\gamma_2''}\big)^*$, and

– $w_j'' \in \big(\langle\langle \eta_2 \rangle\rangle^{\gamma_2''} \setminus \langle\langle \eta_1 \rangle\rangle^{\gamma_1''}\big)^*$.

If $\mathfrak{o}[w](\gamma) = \emptyset$, there is nothing to prove. Otherwise, let $\gamma' = (\sigma, \nu, \rho') \in \mathfrak{o}[w](\gamma)$ and $\rho_1 \stackrel{\text{def}}{=} \rho, \rho_1', \rho_1'', \rho_1''', \ldots, \rho_k, \rho_k', \rho_k'', \rho_k''' \stackrel{\text{def}}{=} \rho'$ be an arbitrary sequence of state maps such that, for all $j \in [1,k]$, we have $(\sigma, \nu, \rho_j') \in \mathfrak{o}[w_j](\sigma, \nu, \rho_j)$, $(\sigma, \nu, \rho_j'') \in \mathfrak{o}[w_j'](\sigma, \nu, \rho_j')$, $(\sigma, \nu, \rho_j''') \in \mathfrak{o}[w_j''](\sigma, \nu, \rho_j'')$, and $(\sigma, \nu, \rho_{j+1}) \in \mathfrak{o}[w_j'''](\sigma, \nu, \rho_j''')$, if $j < k$. Consider the sets of component indices $\mathfrak{C}_i \stackrel{\text{def}}{=} \bigcup_{j=1}^{n} C_j^{\sigma_i}$ from σ_i and denote by $\rho_{i,j}$, $\rho_{i,j}'$, $\rho_{i,j}''$ and $\rho_{i,j}'''$ the finite restrictions of ρ_j, ρ_j', ρ_j'' and ρ_j''' to C_i, for $i = 1, 2$, respectively. We prove the following:

1. $\rho_{2,j} = \rho_{2,j}'$, for all $j \in [1,k]$, and
2. $\rho_{1,j}'' = \rho_{1,j}'''$, for all $j \in [1,k]$.

We prove the first point, the argument for the second point being symmetric. It is sufficient to prove that the state of the components with indices in $\mathfrak{C}_2$, which are the only ones $\rho_{2,j}$ and $\rho_{2,j}'$ account for, is not changed by w_j, for all $j \in [1,k]$. Since $w_j \in \big(\langle\langle \eta_1 \rangle\rangle^{\gamma_1''} \setminus \langle\langle \eta_2 \rangle\rangle^{\gamma_2''}\big)^*$, the only interactions on w_j are the ones from $\gamma_1'' = \gamma_1 \bullet \gamma_1'$ that do not occur in $\gamma_2'' = \gamma_2 \bullet \gamma_2'$, where $\gamma_1' \sqsubseteq \gamma_2$ and $\gamma_2' \sqsubseteq \gamma_1$. It follows that the interactions occurring on w_j are the ones from γ_1 that do not occur in γ_2''. Since $\gamma_2'' \models \mathop{\ast}_{\alpha \in \mathrm{inter}(\phi_1) \setminus (\mathrm{inter}(\overline{\phi}_1) \cup \mathrm{inter}(\phi_2))} \alpha \; [= \mathcal{F}(\phi_2, \phi_1)]$, the interactions occurring on w_j must occur in some model $\overline{\gamma}$ of a tight subformula of ϕ_1. Hence, the interactions from $\overline{\gamma}$ can only change the state of a component from $\overline{\gamma}$. Since $\overline{\gamma} \sqsubseteq \gamma_1$ and $\gamma_1 \bullet \gamma_2$ is defined, there can be no component indexed by some element of $\mathfrak{C}_2$, whose state is changed by an interaction from $\overline{\gamma}$, thus $\rho_{2,j} = \rho_{2,j}'$ (1). Consequently, we obtain two sequences of words and finite state maps:

– $w_1, w_1', w_1''', \ldots, w_k, w_k', w_k'''$ and $\rho_{1,1}, \rho_{1,1}', \rho_{1,1}'', \ldots, \rho_{1,k}, \rho_{1,k}', \rho_{1,k}''$, where $(\sigma_1, \nu_1, \rho_{1,j}') \in \mathfrak{o}[w_1](\sigma_1, \nu_1, \rho_{1,j})$, $(\sigma_1, \nu_1, \rho_{1,j}'') \in \mathfrak{o}[w_1'](\sigma_1, \nu_1, \rho_{1,j}')$ and $(\sigma_1, \nu_1, \rho_{1,j+1}) \in \mathfrak{o}[w_1'''](\sigma_1, \nu_1, \rho_{1,j}'')$, for all $j \in [1,k-1]$, and

– $w_1', w_1'', w_1''', \ldots, w_k', w_k'', w_k'''$ and $\rho_{2,1}', \rho_{2,1}'', \rho_{2,1}''', \ldots, \rho_{2,k}', \rho_{2,k}'', \rho_{2,k}'''$, where $(\sigma_2, \nu_2, \rho_{2,j}'') \in \mathfrak{o}[w_1'](\sigma_2, \nu_2, \rho_{2,j}')$, $(\sigma_2, \nu_2, \rho_{2,j}''') \in \mathfrak{o}[w_1''](\sigma_2, \nu_2, \rho_{2,j}'')$ and $(\sigma_2, \nu_2, \rho_{2,j+1}') \in \mathfrak{o}[w_1'''](\sigma_2, \nu_2, \rho_{2,j}''')$, for all $j \in [1,k-1]$.

Note that $w{\downarrow}_{\langle\langle \eta_1 \rangle\rangle^{\gamma_1''}} = w_1 w_1' w_1''' \ldots w_k w_k' w_k'''$ and $w{\downarrow}_{\langle\langle \eta_2 \rangle\rangle^{\gamma_2''}} = w_1' w_1'' w_1''' \ldots w_k' w_k'' w_k'''$. By the inductive hypothesis, we have $\mathfrak{o}[w{\downarrow}_{\langle\langle \eta_i \rangle\rangle^{\gamma_i''}}](\gamma_i'') \subseteq [\![\psi_i * \mathcal{F}(\phi_i, \phi_{3-i})]\!]_{\mathcal{D}}$, hence $(\sigma_i, \nu, \rho_{i,k}''') \in \mathfrak{o}[w{\downarrow}_{\langle\langle \eta_i \rangle\rangle^{\gamma_i''}}](\gamma_i'')$, for $i = 1, 2$. Moreover, $\sigma_1 \bullet \sigma_2 = \sigma$ and the state maps ρ' and $\rho_{1,k}''' \cup \rho_{2,k}'''$ agree on all pairs (u, C_i), such that $u \in C_i^{\sigma}$, hence $\gamma' = (\sigma, \nu, \rho') \in [\![\psi_1 * \psi_2]\!]_{\mathcal{D}}$.

Proving $(\star)$ for the rest of the rules is a standard check, left to the reader. $\square$

D Proofs from Section 6

Proposition 3 *The set of symbolic configurations using predicate atoms* $\mathsf{tree}_{idle}(x)$, $\mathsf{tree}_{\neg idle}(x)$, $\mathsf{tree}(x)$ *and* $\mathsf{tseg}(x,y)$ *is precisely closed.*

Proof. Let $\varphi_i \stackrel{\text{def}}{=} \phi_i * \circledast_{j=1}^{k_i} \mathsf{tree}_\star(x_{i,j}) * \circledast_{j=k_i+1}^{\ell_i} \mathsf{tseg}(x_{i,j}, y_{i,j})$ be symbolic configurations, where ϕ_i is a predicateless symbolic configuration and $\mathsf{tree}_\star(x)$ is either $\mathsf{tree}_{idle}(x)$, $\mathsf{tree}_{\neg idle}(x)$ or $\mathsf{tree}(x)$, for all $j \in [1,\ell_i]$ and $i = 1,2$. We prove that φ_1 is precise on $[\![\varphi_2]\!]_{\mathcal{D}}$. Let $\gamma = (\sigma,\nu,\rho) \in [\![\varphi_2]\!]_{\mathcal{D}}$ be a configuration and suppose that there exist configurations $\gamma' = (\sigma',\nu,\rho), \gamma'' = (\sigma'',\nu,\rho)$, such that $\gamma' \sqsubseteq \gamma$, $\gamma'' \sqsubseteq \gamma$, $\gamma' \models_{\mathcal{D}} \varphi_1$ and $\gamma'' \models_{\mathcal{D}} \varphi_1$. Then there exist configurations $\gamma'_0 \stackrel{\text{def}}{=} (\sigma'_0,\nu,\rho), \ldots, \gamma'_{\ell_1} \stackrel{\text{def}}{=} (\sigma'_{\ell_1},\nu,\rho)$ and $\gamma''_0 \stackrel{\text{def}}{=} (\sigma''_0,\nu,\rho), \ldots, \gamma''_{\ell_1} \stackrel{\text{def}}{=} (\sigma''_{\ell_1},\nu,\rho)$, such that:

- $\gamma' = \circledast_{j=0}^{\ell_1} \gamma'_j$ and $\gamma'' = \circledast_{j=0}^{\ell_1} \gamma''_j$,
- $\gamma'_0 \models \phi_1$ and $\gamma''_0 \models \phi_1$,
- $\gamma'_j \models_{\mathcal{D}} \mathsf{tree}_\star(x_{1,j})$ and $\gamma''_j \models_{\mathcal{D}} \mathsf{tree}_\star(x_{2,j})$, for all $j \in [1,k_1]$, and
- $\gamma'_j \models_{\mathcal{D}} \mathsf{tseg}(x_{1,j},y_{1,j})$ and $\gamma''_j \models_{\mathcal{D}} \mathsf{tseg}(x_{2,j},y_{2,j})$, for all $j \in [k_1+1,\ell_1]$.

Since ϕ_1 is a predicateless symbolic heap, we have $Node^{\sigma'_0} = Node^{\sigma''_0}$, $Leaf^{\sigma'_0} = Leaf^{\sigma''_0}$, $I_\ell^{\sigma'_0} = I_\ell^{\sigma''_0}$ and $I_r^{\sigma'_0} = I_r^{\sigma''_0}$, thus $\gamma'_0 = \gamma''_0$. Next, for each $j \in [1,k_1]$, we have $I_\ell^{\sigma'_j} = I_\ell^{\sigma''_j}$ and $I_r^{\sigma'_j} = I_r^{\sigma''_j}$, because these relations correspond to the same tree whose root is $\nu(x_{1,j})$, whose frontier contains only indices $u \in Node^{\sigma'_j} \cap Node^{\sigma''_j}$, such that $\rho(u,Node) \in \{leaf\_idle, leaf\_busy\}$. The interpretation of I_ℓ and I_r in both cases is uniquely determined by the fact that the base cases of the inductive definitions of $\mathsf{tree}_\star(x)$ declare components in states *leaf_idle* and *leaf_busy*, which, moreover, do not occur any other rules in $\mathcal{D}$. Finally, for each $j \in [k_1+1,\ell_1]$, we have $I_\ell^{\sigma'_j} = I_\ell^{\sigma''_j}$ and $I_r^{\sigma'_j} = I_r^{\sigma''_j}$, because the structures in which these relations are interpreted correspond to the same tree whose root is $\nu(x_{1,j})$ and frontier contains $\nu(y_{1,j})$ together with indices $u \in Node^{\sigma'_j} \cap Node^{\sigma''_j}$, such that $\rho(u,Node) \in \{leaf\_idle, leaf\_busy\}$. We obtain, consequently, that $\gamma'_j = \gamma''_j$, for all $j \in [1,\ell_1]$, leading to $\gamma' = \gamma''$. $\square$

E Proofs from Section 7

Proposition 4 *The entailment* $\mathsf{A}(x_1,\ldots,x_k) \models_{\mathcal{D}} \mathsf{B}(x_1,\ldots,x_k)$ *is undecidable, even when* $\mathcal{D}$ *is progressing and only the rules defining the predicate symbols from* $\mathrm{dep}_{\mathcal{D}}(\phi)$ *are* λ_ϕ*-connected.*

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 [**?**]. Moreover, we assume w.l.o.g. 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 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, i.e. it contains only production rules of the form $\mathsf{A}_0 \to a\mathsf{A}_1 \dots \mathsf{A}_n$, where $\mathsf{A}_0, \dots \mathsf{A}_n \in N$, for some $n \geq 0$ and $a \in T$.

We consider the signature $\mathfrak{S} = \langle I_0, I_1 \rangle$, where $\#(I_0) = \#(I_1) = 2$. For each nonterminal $\mathsf{A}_0 \in N$, we have a predicate $\mathsf{A}_0(x,y)$ and a rule $\mathsf{A}_0(x,y) \leftarrow \exists x_1 \dots \exists x_n \,.\, I_a(x,x_1) * \mathsf{A}_1(x_1,x_2) * \dots * \mathsf{A}_n(x_n,y)$, for each rule $\mathsf{A}_0 \to a\mathsf{A}_1 \dots \mathsf{A}_n$ of G. Moreover, we consider the rules $\mathsf{B}(x,y) \leftarrow \exists z \,.\, I_a(x,z) * \mathsf{B}(z,y)$ and $\mathsf{B}(x,y) \leftarrow I_a(x,y)$, for all $a \in \{0,1\}$ and let $\mathcal{D}$ be the resulting SID. It is easy to check that the SID is progressing and established and that, moreover, the rules for $\mathsf{B}(x,y)$ are connected. Finally, the entailment $\mathsf{B}(x,y) \models_{\mathcal{D}} \mathsf{A}(x,y)$ is valid if and only if $T^+ \subseteq \mathcal{L}(G)$. $\square$

Proposition 5 *There exists an integer $\mathfrak{B} \geq 1$, such that $\delta(\sigma) \leq \mathfrak{B}$, for each $(\sigma, \nu, \rho) \in [\![\mathsf{A}(x_1, \dots, x_{\#(\mathsf{A})})]\!]_{\mathcal{D}}$ only if $\mathfrak{B} = O(\mathrm{size}(\mathcal{D})^c)$, for a constant $c \geq 1$.*

Proof. Given rules $\rho_1, \rho_2 \in \mathcal{D}$, we write $(\rho_1, i_1) \rightsquigarrow (\rho_2, i_2)$ for the conjunction of the following:

- ρ_1 is a rule $\mathsf{A}(x_1, \dots, x_{\#(\mathsf{A})}) \leftarrow_{\mathcal{D}} \exists y_1 \dots \exists y_r \,.\, \phi * \mathbin{\ast\!\!\!\ast}_{\ell=1}^{h} \mathsf{B}^\ell(z^\ell_1, \dots, z^\ell_{\#(\mathsf{B}^\ell)})$, where ϕ is a symbolic configuration and $i_1 \in [1, \#(\mathsf{A})]$,
- ρ_2 defines B, where $i_2 \in [1, \#(\mathsf{B})]$, and
- $\mathsf{B} = \mathsf{B}^\ell$ and $x_i \simeq_\phi z^\ell_j$, for some $\ell \in [1,h]$.

If, moreover, there exists a variable $y \in \mathrm{fv}(\phi)$, such that $x_i \simeq_\phi y$ and y occurs in an interaction atom in ϕ, we write $(\rho_1, i_1) \dot{\rightsquigarrow} (\rho_2, i_2)$ instead of $(\rho_1, i_1) \rightsquigarrow (\rho_2, i_2)$. If the rule ρ_1 defines A, the sequence $(\rho_1, i_1) \rightsquigarrow (\rho_2, i_2) \rightsquigarrow \dots \rightsquigarrow (\rho_n, i_n)$ corresponds to one or more unfoldings $\mathsf{A}(x_1, \dots, x_{\#(\mathsf{A})}) \Leftarrow_{\mathcal{D}} \varphi_2 \Leftarrow_{\mathcal{D}} \dots \Leftarrow_{\mathcal{D}} \varphi_n$ that differ only by the choices of the rules and predicate atoms along the way. The following fact gives an equivalent condition for degree boundedness:

Fact 2 *There exists an integer $b \geq 1$, such that $\delta(\sigma) \leq b$, for every configuration $(\sigma, \nu, \rho) \in [\![\mathsf{A}(x_1, \dots, x_{\#(\mathsf{A})})]\!]_{\mathcal{D}}$ if and only if there exists an integer $K \geq 1$, such that every unfolding corresponding to the K-th iteration of an elementary cycle $(\rho_1, i_1) \rightsquigarrow (\rho_2, i_2) \rightsquigarrow \dots \rightsquigarrow (\rho_r, i_r) \rightsquigarrow (\rho_1, i_1)$, where $(\rho_\ell, i_\ell) \dot{\rightsquigarrow} (\rho_{\ell+1}, i_{\ell+1})$, for some $\ell \in [1, r-1]$, is $\mathcal{D}$-unsatisfiable.*

Proof. "$\Rightarrow$" Suppose, for a contradiction, that there exists an elementary cycle $\gamma = (\rho_1, i_1) \rightsquigarrow (\rho_2, i_2) \rightsquigarrow \ldots \rightsquigarrow (\rho_r, i_r) \rightsquigarrow (\rho_1, i_1)$, such that $(\rho_\ell, i_\ell) \dot{\rightsquigarrow} (\rho_{\ell+1}, i_{\ell+1})$, for some $\ell \in [1, r-1]$ and for each $K \geq 1$, there exists an unfolding $\mathsf{A}(x_1, \ldots, x_{\#(\mathsf{A})}) \Leftarrow_{\mathcal{D}} \exists \mathbf{z} \,.\, \psi$ corresponding to γ^K, such that ψ is a symbolic configuration and $\exists \mathbf{z} \,.\, \psi$ is $\mathcal{D}$-satisfiable. Let (σ, ν, ρ) be a $\mathcal{D}$-model of ψ. Then there are at least K interaction atoms $I(\mathbf{x})$ in ψ and variables $y \in \mathbf{x}$, such that $x_{i_1} \simeq_\psi y$. Clearly the interactions $(I, \nu(\mathbf{x}))$ must be pairwise distinct, otherwise ψ, and consequently φ^K could not be satisfiable (all these interaction atoms are connected by separating conjunctions). Then $\nu(x_{i_1})$ occurs in at least K distinct interactions. Since this happens for each $K \geq 1$, we obtain a contradiction with the hypothesis.

"$\Leftarrow$" Since no k-unfolding corresponding to an elementary cycle $(\rho_1, i_1) \rightsquigarrow (\rho_2, i_2) \rightsquigarrow \ldots \rightsquigarrow (\rho_r, i_r) \rightsquigarrow (\rho_1, i_1)$ yields a satisfiable formula, and the length of each elementary cycle is bounded by $\|\mathcal{D}\| \cdot \mathrm{width}(\mathcal{D})$. $\square$

We are left with proving that the bound b on the maximum number of tuples to which an index may belong in a $\mathcal{D}$-model of $\mathsf{A}(x_1, \ldots, x_{\#(\mathsf{A})})$ depends only on $\mathcal{D}$. Let $\mathsf{A}(x_1, \ldots, x_{\#(\mathsf{A})}) \Leftarrow_{\mathcal{D}} \exists \mathbf{z} \,.\, \psi$ be an unfolding corresponding to γ^K, such that ψ is $\mathcal{D}$-unsatisfiable. By Fact 2, we know that for each elementary cycle γ as above, such an integer K exists. We define the undirected graph $(\mathcal{V}, \mathcal{E})$:

- $\mathcal{V}$ is the set of pairs (ρ, x), where:

  $$\rho : \mathsf{A}(x_1, \ldots, x_{\#(\mathsf{A})}) \leftarrow_{\mathcal{D}} \exists y_1 \ldots \exists y_r \,.\, \phi * \mathop{\raisebox{-.5ex}{\Large$\ast$}}_{\ell=1}^{h} \mathsf{B}^\ell(z_1^\ell, \ldots, z_{\#(\mathsf{B}^\ell)}^\ell)$$

  is a rule of $\mathcal{D}$, ϕ is a symbolic configuration, such that $x \in \mathrm{fv}(\phi)$, and

- $\mathcal{E}$ is the set of undirected edges between (ρ, x) and (ρ', y), where ρ is a rule as above and either:
  - $\rho = \rho'$, $y \in \mathrm{fv}(\phi)$ and $x \simeq_\phi y$, or
  - ρ' defines the predicate atom $\mathsf{B}^\ell(y_1, \ldots, y_{\#(\mathsf{B}^\ell)})$, $y = y_j$ and $x \simeq_\phi z_j^\ell$.

Clearly, we have $\|\mathcal{V}\| \leq \mathrm{size}(\mathcal{D}) \cdot \mathrm{width}(\mathcal{D})$, because $\|\mathcal{D}\| \leq \mathrm{size}(\mathcal{D})$ and there are at most $\mathrm{width}(\mathcal{D})$ variables in each rule of $\mathcal{D}$. Since ψ is unsatisfiable, at least one of the following holds:

- ψ contains two component atoms $C_i(x)$ and $C_i(y)$, such that $x \simeq_\psi y$. In this case, there exists an elementary path from (ρ, x) to (ρ', y), for some rules $\rho, \rho' \in \mathcal{D}$, in $(\mathcal{V}, \mathcal{E})$, of length at most $\|\mathcal{V}\|$, hence $b \leq \mathrm{size}(\mathcal{D}) \cdot \mathrm{width}(\mathcal{D})$.
- ψ contains two interaction atoms $I_j(x_1, \ldots, x_{\#(I_j)})$ and $I_j(y_1, \ldots, y_{\#(I_j)})$, such that $x_i \simeq_\psi y_i$, for each $i \in [1, \#(I_j)]$. In this case, consider the product graph $(\mathcal{V}^{\#(I_j)}, \mathcal{E}_{I_j})$, where $\mathcal{E}_{I_j}$ has an edge between $\langle (\rho_1, x_1), \ldots, (\rho_{\#(I_j)}, x_{\#(I_j)}) \rangle$ and $\langle (\rho'_1, x'_1), \ldots, (\rho'_{\#(I_j)}, x'_{\#(I_j)}) \rangle$ if and only if there is an edge between (ρ_i, x_i) and (ρ'_i, x'_i) in $\mathcal{E}$, for all $i \in [1, \#(I_j)]$. Then there exists a path between $\langle (\rho_1, x_1), \ldots, (\rho_{\#(I_j)}, x_{\#(I_j)}) \rangle$ and $\langle (\rho'_1, y_1), \ldots, (\rho'_{\#(I_j)}, y_{\#(I_j)}) \rangle$, for some rules $\rho_1, \rho'_1, \ldots, \rho_{\#(I_j)}, \rho'_{\#(I_j)} \in \mathcal{D}$, of length at most $\|\mathcal{V}\|^{\#(I_j)}$. Hence $\mathfrak{B} \leq \|\mathcal{V}\|^{\#(I_j)} \leq (\mathrm{size}(\mathcal{D}) \cdot \mathrm{width}(\mathcal{D}))^{\#(I_j)}$.

- ψ contains two state atoms $\mathsf{state}(x, C_i, q)$ and $\mathsf{state}(y, C_i, q')$, such that $x \simeq_\psi y$ and $q \neq q'$. Similar to the first point, we obtain $b \leq \|\mathcal{V}\|$, in this case.

In all cases, we have $\mathfrak{B} = \mathcal{O}\left((\mathrm{size}(\mathcal{D}) \cdot \mathrm{width}(\mathcal{D}))^{2\max_{j=1}^{m} \#(I_j)}\right) = \mathcal{O}\left(\mathrm{size}(\mathcal{D})^{2\max_{j=1}^{m} \#(I_j)}\right)$. $\square$

Lemma 4 *If $\mathcal{D}$ is progressing, for each $\mathcal{D}$-model (σ, ν, ρ) of $\mathsf{A}(x_1, \ldots, x_{\#(\mathsf{A})}) \in \mathrm{def}(\mathcal{D})$ and each heap $\mathsf{h} \in \mathcal{G}_{\mathfrak{B}}(\sigma, \rho)$, there exists a mapping $\iota : [1, \#(\mathsf{A})] \times [1, m] \to 2^{[0, \mathfrak{B}-1]}$ and a store $\overline{\nu}$, such that the following hold:*

1. $\overline{\nu}(x_i) = \nu(x_i) \in \mathrm{dom}(\mathsf{h})$ *and* $\overline{\nu}(\eta(x_i)) = \mathsf{h}(\nu(x_i))$, *for each* $i \in [1, \#(\mathsf{A})]$,
2. $\{\mathbf{u} \in I_j^{\sigma} \mid \overline{\nu}(x_i) \in \mathbf{u}\} = \{\langle \mathsf{h}(\overline{\nu}(x_i)) \rangle_{\mathrm{ipos}(j,k)} \mid k \in \iota(i, j)\}$, *for all* $i \in [1, \#(\mathsf{A})]$ *and* $j \in [1, m]$,
3. $(\overline{\nu}, \mathsf{h}) \models^{\mathsf{SL}}_{\mathcal{D}} \overline{\mathsf{A}}_{\iota}(x_1, \ldots, x_{\#(\mathsf{A})}, \eta(x_1), \ldots, \eta(x_{\#(\mathsf{A})}))$.

Proof. Since $(\sigma, \nu, \rho) \models_{\mathcal{D}} \mathsf{A}(x_1, \ldots, x_{\#(\mathsf{A})})$, there exists a complete unfolding $\mathsf{A}(x_1, \ldots, x_{\#(\mathsf{A})}) \Leftarrow^{\circ}_{\mathcal{D}} \exists t_1 \ldots \exists t_k \,.\, \psi$, where ψ is a predicateless symbolic configuration, and indices $u_1, \ldots, u_k \in \mathbb{U}$, such that $(\sigma, \nu[t_1 \leftarrow u_1, \ldots, t_k \leftarrow u_k], \rho) \models \psi$. By an α-renaming of the existentially quantified variables, if necessary, we can assume that $t_1, \ldots, t_k$ are pairwise distinct and $\{x_1, \ldots, x_{\#(\mathsf{A})}\} \cap \{t_1, \ldots, t_k\} = \emptyset$. Moreover, because $\mathcal{D}$ is progressing, there are no equality atoms in ψ, hence we can assume w.l.o.g. that $\nu' \stackrel{\mathsf{def}}{=} \nu[t_1 \leftarrow u_1, \ldots, t_k \leftarrow u_k]$ is injective. The proof goes by induction on the length of the complete unfolding above.

For the base case, the length is one and there exists a rule $\mathsf{A}(x_1, \ldots, x_{\#(\mathsf{A})}) \leftarrow_{\mathcal{D}} \exists t_1 \ldots \exists t_k \,.\, \psi$. Since $\mathcal{D}$ is progressing, by point (1) of Def. 13, we have $\#(\mathsf{A}) = 1$, $k = 0$ and x_1 is the only variable that occurs in ψ. Since $(\sigma, \nu', \rho) \models \psi$, we obtain that $\mathrm{nodes}(\sigma) = \{\nu(x_1)\}$, because $\nu'(x_1) = \nu(x_1)$, by the definition of ν'. Because $\mathsf{h} \in \mathcal{G}_{\mathfrak{B}}(\sigma, \rho)$, we have $\mathrm{dom}(\mathsf{h}) = \{\nu(x_1)\}$, by Def. 15. We define the store $\overline{\nu}$ as $\overline{\nu}(\eta(x_1)) \stackrel{\mathsf{def}}{=} \mathsf{h}(\nu(x_1))$ and $\overline{\nu}(x) = \nu'(x)$, for each x that does not occur in $\eta(x_1)$, thus taking care of point (1) of the statement. Let $\iota(1, j) \stackrel{\mathsf{def}}{=} \{k_1, \ldots, k_h\}$ be the set of integers whose existence is stated by point (2) of Def. 15, relative to $\nu(x_1)$, thus taking care of point (2) of the statement. The proof for point (3) relies on the following:

- $(\overline{\nu}, \mathsf{h}) \models^{\mathsf{SL}} x_1 \mapsto \eta(x_1)$: $\mathrm{dom}(\mathsf{h}) = \{\overline{\nu}(x_1)\}$ and $\overline{\nu}(\eta(x_1)) = \mathsf{h}(\overline{\nu}(x_1))$, by definition of $\overline{\nu}$.
- $(\overline{\nu}, \emptyset) \models^{\mathsf{SL}} \mathsf{CompState}_{\psi}(x_1)$: $\mathsf{h} \in \mathcal{G}_{\mathfrak{B}}(\sigma, \rho)$, by points 1 and 3 of Def. 15.
- $(\overline{\nu}, \emptyset) \models^{\mathsf{SL}} \mathsf{Inter}_{\psi}(x_1)$: $\mathsf{h} \in \mathcal{G}_{\mathfrak{B}}(\sigma, \rho)$, by the definition of $\iota(1, j)$, for all $j \in [1, m]$ and by point 2 of Def. 15.

We obtain $(\overline{\nu}, \mathsf{h}) \models^{\mathsf{SL}} x_1 \mapsto \eta(x_1) * \mathsf{CompState}_{\psi}(x_1) * \mathsf{Inter}_{\psi}(x_1)$, thus $(\overline{\nu}, \mathsf{h}) \models^{\mathsf{SL}}_{\mathcal{D}} \overline{\mathsf{A}}_{\iota}(x_1, \eta(x_1))$, because $\overline{\mathsf{A}}_{\iota}(x_1, \eta(x_1)) \leftarrow x_1 \mapsto \eta(x_1) * \mathsf{CompState}_{\psi}(x_1) * \mathsf{Inter}_{\psi}(x_1)$ is a rule of $\overline{\mathcal{D}}$.

For the inductive step, let $\mathsf{A}(x_1, \ldots, x_{\#(\mathsf{A})}) \Leftarrow_{\mathcal{D}} \exists z_1 \ldots \exists z_p \,.\, \phi * \mathop{\ast}_{\ell=1}^{h} \mathsf{B}^{\ell}(y_1^{\ell}, \ldots, y_{\#(\mathsf{B}^{\ell})}^{\ell})$ be the first step of the unfolding, hence $(\sigma, \nu', \rho) \models_{\mathcal{D}} \phi * \mathop{\ast}_{\ell=1}^{h} \mathsf{B}^{\ell}(y_1^{\ell}, \ldots, y_{\#(\mathsf{B}^{\ell})}^{\ell})$ and

there exist structures $\sigma_0, \ldots, \sigma_h$, such that $\sigma = \sigma_0 \bullet \ldots \bullet \sigma_h$, $(\sigma_0, \nu', \rho) \models \phi$ and $(\sigma_\ell, \nu', \rho) \models_{\mathcal{D}} \mathsf{B}^\ell(y_1^\ell, \ldots, y_{\#(\mathsf{B}^\ell)}^\ell)$, for all $\ell \in [1,h]$. We define the heaps $\mathsf{h}_1, \ldots, \mathsf{h}_h$ as the restrictions of h to $\mathrm{nodes}(\sigma_1), \ldots, \mathrm{nodes}(\sigma_h)$, respectively, and infer that $\mathsf{h}_\ell \in \mathcal{G}_{\mathfrak{B}}(\sigma_\ell, \rho)$, for all $\ell \in [1,h]$, by proving the following fact:

Fact 3 *Given a configuration* (σ, ν, ρ) *and* $\mathsf{h} \in \mathcal{G}_\rho(\sigma, \nu)$*, for any subconfiguration* $(\sigma', \nu, \rho) \sqsubseteq (\sigma, \nu, \rho)$*, we have* $\mathsf{h}' \in \mathcal{G}_\rho(\sigma', \nu)$*, where* h' *is the restriction of* h *to* $\mathrm{nodes}(\sigma')$.

Proof. We have $\mathrm{dom}(\mathsf{h}') = \mathrm{dom}(\mathsf{h}) \cap \mathrm{nodes}(\sigma') = \mathrm{nodes}(\sigma) \cap \mathrm{nodes}(\sigma') = \mathrm{nodes}(\sigma')$, because $\mathrm{dom}(\mathsf{h}) = \mathrm{nodes}(\sigma) \supseteq \mathrm{nodes}(\sigma')$. The points (1-3) of Def. 15 are by easy inspection. $\square$

Next, we prove that $\mathrm{dom}(\mathsf{h}_i) \cap \mathrm{dom}(\mathsf{h}_j) = \emptyset$, for all $i \neq j \in [1,h]$. Suppose, for a contradiction, that there exists $u \in \mathrm{dom}(\mathsf{h}_i) \cap \mathrm{dom}(\mathsf{h}_j)$, for some $i \neq j \in [1,h]$. Since $\mathsf{h}_\ell \in \mathcal{G}_{\mathfrak{B}}(\sigma_\ell, \rho)$, we obtain $u \in \mathrm{nodes}(\sigma_i) \cap \mathrm{nodes}(\sigma_j)$, by Def. 15. Let $\mathsf{B}^i(y_1^i, \ldots, y_{\#(\mathsf{B}^i)}^i) \Leftarrow^\circ_{\mathcal{D}} \exists \mathbf{t}_i \ . \ \psi_i$ and $\mathsf{B}^j(y_1^j, \ldots, y_{\#(\mathsf{B}^i)}^j) \Leftarrow^\circ_{\mathcal{D}} \exists \mathbf{t}_j \ . \ \psi_j$ be the complete unfoldings of the predicate atoms $\mathsf{B}^i(y_1^i, \ldots, y_{\#(\mathsf{B}^i)}^i)$ and $\mathsf{B}^j(y_1^j, \ldots, y_{\#(\mathsf{B}^i)}^j)$ in the above unfolding, respectively. Because we have assumed that ν' is injective, there exists a variable $x \in (\mathbf{t}_i \cup \{y_1^i, \ldots, y_{\#(\mathsf{B}^i)}^i\}) \cap (\mathbf{t}_j \cup \{y_1^j, \ldots, y_{\#(\mathsf{B}^j)}^j\})$, such that $\nu'(x) = u$. Since $t_1, \ldots, t_k$ are pairwise distinct, by the above assumption, we obtain that $\mathbf{t}_i \cap \mathbf{t}_j = \emptyset$. Since, moreover, $\{y_1^i, \ldots, y_{\#(\mathsf{B}^i)}^i\} \subseteq \mathbf{t}_i \cup \{x_1, \ldots, x_{\#(\mathsf{A})}\}$ and $\{y_1^j, \ldots, y_{\#(\mathsf{B}^j)}^j\} \subseteq \mathbf{t}_j \cup \{x_1, \ldots, x_{\#(\mathsf{A})}\}$, we must have $x \in \{y_1^i, \ldots, y_{\#(\mathsf{B}^i)}^i\} \cap \{y_1^j, \ldots, y_{\#(\mathsf{B}^j)}^j\}$, which contradicts the fact that $\mathcal{D}$ is progressing (point 1 of Def. 13). Let $\mathsf{h}_0 \stackrel{\mathsf{def}}{=} \mathsf{h} \setminus \left(\bigcup_{\ell=1}^h \mathsf{h}_\ell\right)$. Since $\mathrm{dom}(\mathsf{h}_i) \cap \mathrm{dom}(\mathsf{h}_j) = \emptyset$, for all $i \neq j \in [1,h]$, the composition $\mathsf{h} = \mathsf{h}_0 \uplus \mathsf{h}_1 \uplus \ldots \uplus \mathsf{h}_h$ is properly defined. Since $(\sigma_\ell, \nu', \rho) \models_{\mathcal{D}} \mathsf{B}^\ell(y_1^\ell, \ldots, y_{\#(\mathsf{B}^\ell)}^\ell)$ and $\mathsf{h}_\ell \in \mathcal{G}_{\mathfrak{B}}(\sigma_\ell, \rho)$, by the induction hypothesis, there exist stores $\overline{\nu}_\ell$ and mappings ι^ℓ, such that:

- $\overline{\nu}_\ell(y_i^\ell) = \nu'(y_i^\ell) \in \mathrm{dom}(\mathsf{h}_\ell)$, for all $i \in [1, \#(\mathsf{B}^\ell)]$,
- $\mathsf{h}_\ell(\overline{\nu}_\ell(y_i^\ell)) = \overline{\nu}_\ell(\eta(y_i^\ell))$,
- $\{\mathbf{u} \in I_j^{\sigma_\ell} \mid \overline{\nu}_\ell(x_i) \in \mathbf{u}\} = \{\langle \mathsf{h}_\ell(\overline{\nu}_\ell(x_i)) \rangle_{\mathrm{ipos}(j,k)} \mid k \in \iota(i,j)\}$, for all $i \in [1, \#(\mathsf{B}^\ell)]$, $j \in [1,m]$,
- $(\overline{\nu}_\ell, \mathsf{h}_\ell) \models^{\mathsf{SL}}_{\mathcal{D}} \overline{\mathsf{B}}^\ell_{\iota^\ell}(y_1^\ell, \ldots, y_{\#(\mathsf{B}^\ell)}^\ell)$.

for all $\ell \in [1,h]$. We define the store $\overline{\nu}$ as follows:

- $\overline{\nu}(\eta(x_i)) \stackrel{\mathsf{def}}{=} \mathsf{h}(\overline{\nu}'(x_i))$, for each $i \in [1, \#(\mathsf{A})]$,
- $\overline{\nu}(\eta(y_i^\ell)) \stackrel{\mathsf{def}}{=} \overline{\nu}_\ell(\eta(y_i^\ell))$, for each $\ell \in [1,h]$ and $i \in [1, \#(\mathsf{B}^\ell)]$.
- $\overline{\nu}$ agrees with ν' everywhere else.

Point (1) follows directly from the definition of $\overline{\nu}$. For each $i \in [1, \#(\mathsf{A})]$ and each $j \in [1,m]$, let $\{I_j(\mathbf{y}_1), \ldots, I_j(\mathbf{y}_{h_i})\} \stackrel{\mathsf{def}}{=} I_\phi^j(x_i)$ and let $k_1, \ldots, k_{h_i} \in [0, \mathfrak{B}-1]$ be integers,

such that $\langle \mathsf{h}(\nu(x_i))\rangle_{\mathrm{ipos}(j,k_\ell)} = \nu'(\mathbf{y}_\ell)$, for all $\ell \in [1,h_i]$. The existence of these integers is stated by point (2) of Def. 15, relative to $\nu(x_i)$. We define $\iota(i,j) \stackrel{\mathsf{def}}{=} \{k_1,\ldots,k_{h_i}\} \cup \mathcal{Z}_j(x_i)$, for all $i \in [1,\#(\mathsf{A})]$ and $j \in [1,m]$, where $\mathcal{Z}_j(x_i) = \bigcup_{\ell=1}^{h} \bigcup_{k=1}^{\#(\mathsf{B}^\ell)} \{\iota^\ell(k,j) \mid x_i \simeq_\phi y_k^\ell\}$. This takes care of point (2) of the statement. Suppose, for a contradiction, that $\{k_1,\ldots,k_{h_i}\} \cap \mathcal{Z}_j(x_i) \neq \emptyset$, for some $i \in [1,\#(\mathsf{A})]$ and $j \in [1,m]$. By the condition on ι_ℓ from the inductive hypothesis, there exists a tuple $\mathbf{u} \in I_j^{\sigma_0} \cap I_j^{\sigma_\ell}$, for some $\ell \in [1,h]$, which contradicts the fact that the composition $\sigma_0 \bullet \sigma_\ell$ is defined. Hence $\{k_1,\ldots,k_{h_i}\} \cap \mathcal{Z}_j(x_i) = \emptyset$ and the annotated rule below:

$$\overline{\mathsf{A}}_\iota(x_1,\ldots,x_{\#(\mathsf{A})},\eta(x_1),\ldots,\eta(x_{\#(\mathsf{A})})) \leftarrow \exists z_1 \ldots \exists z_p \exists \eta(z_1) \ldots \exists \eta(z_p) \,.\, \overline{\phi} * \mathbin{\ast}_{\ell=1}^{h} \overline{\mathsf{B}}_{\iota^\ell}^{\ell}(y_1^\ell,\ldots,y_{\#(\mathsf{B}^\ell)}^\ell,\eta(y_1^\ell),\ldots,\eta(y_{\#(\mathsf{B}^\ell)}^\ell))$$

is well-formed and thus belongs to $\overline{\mathcal{D}}$. To prove point (3) of the statement, namely that $(\overline{\nu},\mathsf{h}) \models^{\mathsf{SL}}_{\overline{\mathcal{D}}} \overline{\mathsf{A}}_\iota(x_1,\ldots,x_{\#(\mathsf{A})},\eta(x_1),\ldots,\eta(x_{\#(\mathsf{A})}))$, we are left with proving that $(\overline{\nu},\mathsf{h}_0) \models \overline{\phi}$, where:

$$\overline{\phi} = x_1 \mapsto \eta(x_1) \;*\; \mathbin{\ast}_{x \in \mathrm{fv}(\phi)} \mathsf{CompState}_\phi(x) \;*\; \mathbin{\ast}_{i=1}^{\#(\mathsf{A})} \mathsf{Inter}_\phi(x_i)$$

To this end, we show the following points:

- $(\overline{\nu},\mathsf{h}_0) \models x_1 \mapsto \eta(x_1)$: by the definition of $\overline{\nu}$, it suffices to prove that $\mathrm{dom}(\mathsf{h}_0) = \{\overline{\nu}(x_1)\}$, or equivalently, $\mathrm{nodes}(\sigma) \setminus \bigcup_{\ell=1}^{h} \mathrm{nodes}(\sigma_\ell) = \{\overline{\nu}(x_1)\}$. "$\subseteq$" Let $u \in \mathrm{dom}(\mathsf{h}_0)$ be an index. Then there exists a variable $y \in \mathrm{fv}(\phi * \mathbin{\ast}_{\ell=1}^{h} \psi_\ell)$, such that $\nu'(y) = u$. If y occurs in a component or interaction atom from some ψ_ℓ, $\ell \in [1,h]$, then $u \in \mathrm{nodes}(\sigma_\ell)$, contradiction. Hence y must occur in a component or interaction atom from ϕ and $y \in \mathrm{fv}(\phi) \setminus \bigcup_{\ell=1}^{h} \{y_1^\ell,\ldots,y_{\#(\mathsf{B}^\ell)}^\ell\}$. Since $\mathcal{D}$ is progressing, we obtain $y = x_1$. "$\supseteq$" Because $\mathcal{D}$ is progressing, x_1 occurs in a component or interaction atom from ϕ, thus $\overline{\nu}(x_1) = \nu'(x_1) \in \mathrm{nodes}(\sigma)$. Suppose, for a contradiction, that $\nu'(x_1) \in \mathrm{nodes}(\sigma_\ell)$, for some $\ell \in [1,h]$. Then $\nu'(x_1) = \nu(y)$, for some $y \in \mathrm{fv}(\psi_\ell) = \{y_1^\ell,\ldots,y_{\#(\mathsf{B}^\ell)}^\ell\} \cup \mathbf{t}_\ell$, which contradicts the assumption that ν' is injective.
- $(\overline{\nu},\emptyset) \models \mathsf{CompState}_\phi(x)$, for each $x \in \mathrm{fv}(\phi)$: $\mathsf{h} \in \mathcal{G}_{\mathfrak{B}}(\sigma,\rho)$, by points 1 and 3 of Def. 15.
- $(\overline{\nu},\emptyset) \models \mathsf{Inter}_\phi(x_i)$, for each $i \in [1,\#(\mathsf{A})]$: $\mathsf{h} \in \mathcal{G}_{\mathfrak{B}}(\sigma,\rho)$, by the definition of $\iota(1,j)$, for all $j \in [1,m]$ and by point 2 of Def. 15.

Lemma 5 *If $\mathcal{D}$ is progressing, for a predicate atom $\mathsf{A}(x_1,\ldots,x_{\#(\mathsf{A})}) \in \mathrm{def}(\mathcal{D})$, each mapping $\iota : [1,\#(\mathsf{A})] \times [1,m] \to 2^{[0,\mathfrak{B}-1]}$ and each $\overline{\mathcal{D}}$-model $(\overline{\nu},\mathsf{h})$ of $\overline{\mathsf{A}}_\iota(x_1,\ldots,x_{\#(\mathsf{A})},\eta(x_1),\ldots,\eta(x_{\#(\mathsf{A})}))$, the following hold:*

1. *for each $i \in [1,\#(\mathsf{A})]$, we have $\overline{\nu}(x_i) \in \mathrm{dom}(\mathsf{h})$ and $\mathsf{h}(\overline{\nu}(x_i)) = \overline{\nu}(\eta(x_i))$, and*
2. *there is a structure σ and a state map ρ, such that $\mathsf{h} \in \mathcal{G}_{\mathfrak{B}}(\sigma,\rho)$ and $(\sigma,\overline{\nu},\rho) \models_{\mathcal{D}} \mathsf{A}(x_1,\ldots,x_{\#(\mathsf{A})})$.*

Proof. Since $(\overline{\nu},\mathsf{h}) \models^{\mathsf{SL}}_{\overline{\mathcal{D}}} \overline{\mathsf{A}}_\iota(x_1,\ldots,x_{\#(\mathsf{A})},\eta(x_1),\ldots,\eta(x_{\#(\mathsf{A})}))$, there exists a complete unfolding $\overline{\mathsf{A}}_\iota(x_1,\ldots,x_{\#(\mathsf{A})},\eta(x_1),\ldots,\eta(x_{\#(\mathsf{A})})) \Leftarrow^{\circ}_{\overline{\mathcal{D}}} \exists t_1\ldots\exists t_k \exists \eta(t_1)\ldots\exists\eta(t_k)\,.\,\overline{\psi}$, such that $(\overline{\nu}[t_1\leftarrow u_1,\ldots,t_k\leftarrow u_k,\eta(t_1)\leftarrow \mathbf{v}_1,\ldots,\eta(t_k)\leftarrow \mathbf{v}_k],\mathsf{h}) \models \overline{\psi}$, for some indices $u_1,\ldots,u_k \in \mathbb{U}$ and tuples of indices $\mathbf{v}_1,\ldots,\mathbf{v}_k \in \mathbb{U}^{\mathfrak{K}}$. Since each rule in $\overline{\mathcal{D}}$ has a stem in $\mathcal{D}$, we consider the complete unfolding $\mathsf{A}(x_1,\ldots,x_{\#(\mathsf{A})}) \Leftarrow^{\circ}_{\mathcal{D}} \exists t_1\ldots\exists t_k\,.\,\psi$. We assume in the following that:

- $t_1,\ldots,t_k$ are pairwise distinct and $\eta(t_1),\ldots,\eta(t_k)$ are pairwise disjoint tuples. This assumption is w.l.o.g. because existentially quantified variables can be α-renamed, if necessary.

- the store $\overline{\nu}' \stackrel{\mathsf{def}}{=} \overline{\nu}[t_1\leftarrow u_1,\ldots,t_k\leftarrow u_k,\eta(t_1)\leftarrow \mathbf{v}_1,\ldots,\eta(t_k)\leftarrow \mathbf{v}_k]$ is injective over $\mathrm{fv}(\psi)$. This assumption is w.l.o.g. because the only equalities in $\overline{\psi}$ are of the form $x\doteq y$, where $x\in\mathrm{fv}(\phi)$ and $y\in\eta(x)$.

The two points of the statement are proved by induction on the length of the complete unfolding:
(1) For the base case, because $\mathcal{D}$ is progressing, we have $\#(\mathsf{A})=1$, $k=0$ and there exists a rule $\overline{\mathsf{A}}_\iota(x_1,\eta(x_1)) \leftarrow_{\overline{\mathcal{D}}} x_1\mapsto(\eta(x_1)) * \mathsf{CompState}_\phi(x_1) * \mathsf{Inter}_\phi(x_1)$, such that $(\overline{\nu},\mathsf{h}) \models x_1\mapsto(\eta(x_1))$, thus $\overline{\nu}(x_1)\in\mathrm{dom}(\mathsf{h})$ and $\mathsf{h}(\overline{\nu}(x_1)) = \overline{\nu}(\eta(x_1))$. For the inductive case, let

$$\overline{\mathsf{A}}_\iota(x_1,\ldots,x_{\#(\mathsf{A})},\eta(x_1),\ldots,\eta(x_{\#(\mathsf{A})})) \quad \Leftarrow_{\overline{\mathcal{D}}} \quad \begin{array}{l} \exists z_1\ldots\exists z_p\exists\eta(z_1)\ldots\exists\eta(z_p)\,.\,\overline{\phi} * \\ \quad * \mathop{*}_{\ell=1}^{h} \overline{\mathsf{B}}^\ell_{\iota^\ell}(y^\ell_1,\ldots,y^\ell_{\#(\mathsf{B}^\ell)},\eta(y^\ell_1),\ldots,\eta(y^\ell_{\#(\mathsf{B}^\ell)})) \end{array}$$

be the first step of the complete unfolding. Let $i\in[1,\#(\mathsf{A})]$ and prove that $\overline{\nu}(x_i)\in\mathrm{dom}(\mathsf{h})$ and $\mathsf{h}(\overline{\nu}(x_i)) = \overline{\nu}(\eta(x_i))$. If $i=1$ then $x_1\mapsto(\eta(x_1))$ is a subformula of $\overline{\phi}$ and the result follows from the fact that $(\overline{\nu}',\mathsf{h}) \models^{\mathsf{SL}}_{\overline{\mathcal{D}}} \overline{\phi} * \mathop{*}_{\ell=1}^{h} \overline{\mathsf{B}}^\ell_{\iota^\ell}(y^\ell_1,\ldots,y^\ell_{\#(\mathsf{B}^\ell)},\eta(y^\ell_1),\ldots,\eta(y^\ell_{\#(\mathsf{B}^\ell)}))$. Otherwise, $i\in[2,\#(\mathsf{A})]$ and, because $\mathcal{D}$ is progressing, it must be that $x_i = y^\ell_j$, for some $\ell\in[1,h]$ and $j\in[1,\#(\mathsf{B}^\ell)]$. In this case, the result follows by an application of the inductive hypothesis.

(2) For the base case, we have $\mathrm{dom}(\mathsf{h}) = \overline{\nu}(x_1)$ and define the structure σ and the state map ρ below:

- $\sigma \stackrel{\mathsf{def}}{=} \langle C^\sigma_1,\ldots,C^\sigma_n,I^\sigma_1,\ldots,I^\sigma_m\rangle$ is such that, for all $i\in[1,n]$ and $j\in[1,m]$:

$$C^\sigma_i \;=\; \begin{cases} \{\overline{\nu}(x_1)\} & \text{, if } \langle \mathsf{h}(\overline{\nu}(x_1))\rangle_i = \overline{\nu}(x_1) \\ \emptyset & \text{, otherwise} \end{cases}$$

$$I^\sigma_j \;=\; \begin{cases} \{\overline{\nu}(\eta(x_1))\} & \text{, if } \langle \mathsf{h}(\overline{\nu}(x_1))\rangle_{\mathrm{ipos}(j,k)} = \overline{\nu}(\eta(x_1)), \text{ for some } k\in[0,\mathfrak{B}-1] \\ \emptyset & \text{, otherwise} \end{cases}$$

- for all $i\in[1,n]$ and $s\in[1,\|\mathbb{Q}\|]$, we have $\rho(\overline{\nu}(x_1),C_i) = q_s$ if $\langle \mathsf{h}(\overline{\nu}(x_1))\rangle_{\mathrm{spos}(i,s)} = \overline{\nu}(x_1)$ and $\rho(\overline{\nu}(x_1),C_i)$ is random, otherwise.

By the definition of σ and ρ, we have $\mathsf{h} \in \mathcal{G}_{\mathfrak{B}}(\sigma,\rho)$. To prove that $(\sigma,\overline{\nu},\rho) \models_{\mathcal{D}} \mathsf{A}(x_1,\ldots,x_{\#(\mathsf{A})})$, let $\overline{\mathsf{A}}_\iota(x_1,\eta(x_1)) \leftarrow_{\overline{\mathcal{D}}} \overline{\phi}$ be the single rule applied on the complete unfolding of $\overline{\mathsf{A}}_\iota(x_1,\eta(x_1))$ and let $\mathsf{A}(x_1) \leftarrow_{\mathcal{D}} \phi$ be its stem. We infer that $(\sigma,\overline{\nu},\rho) \models \phi$ using the fact that $(\overline{\nu},\mathsf{h}) \models^{\mathsf{SL}} \overline{\phi}$, by a case split on the type of component and interaction atoms in ϕ.

For the inductive case, let the first step of the complete unfolding of $\overline{\mathsf{A}}_\iota(x_1,\ldots,x_{\#(\mathsf{A})},\eta(x_1),\ldots,\eta(x_{\#(\mathsf{A})}))$ be:

$$\overline{\mathsf{A}}_\iota(x_1,\ldots,x_{\#(\mathsf{A})},\eta(x_1),\ldots,\eta(x_{\#(\mathsf{A})})) \quad \Leftarrow_{\overline{\mathcal{D}}} \quad \begin{array}{l} \exists z_1 \ldots \exists z_p \exists \eta(z_1) \ldots \exists \eta(z_p) \,.\, \overline{\phi} * \\ \quad * \, {}^{h}_{\ell=1} \, \overline{\mathsf{B}}^{\ell}_{\iota\ell}(y^\ell_1,\ldots,y^\ell_{\#(\mathsf{B}^\ell)},\eta(y^\ell_1),\ldots,\eta(y^\ell_{\#(\mathsf{B}^\ell)})) \end{array}$$

where the stem of the rule applied at this step is:

$$\mathsf{A}(x_1,\ldots,x_{\#(\mathsf{A})}) \leftarrow_{\mathcal{D}} \exists z_1 \ldots \exists z_p \,.\, \phi \, * \, {}^{*}_{\ell=1}{}^{h} \mathsf{B}^\ell(y^\ell_1,\ldots,y^\ell_{\#(\mathsf{B}^\ell)})$$

Since $(\overline{\nu}',\mathsf{h}) \models^{\mathsf{SL}}_{\overline{\mathcal{D}}} \overline{\phi} * {}^{*}_{\ell=1}{}^{h} \, \overline{\mathsf{B}}^{\ell}_{\iota\ell}(y^\ell_1,\ldots,y^\ell_{\#(\mathsf{B}^\ell)},\eta(y^\ell_1),\ldots,\eta(y^\ell_{\#(\mathsf{B}^\ell)}))$, there exist heaps $\mathsf{h}_0,\ldots,\mathsf{h}_h$, such that the following hold:

- $\mathsf{h} = \mathsf{h}_0 \uplus \ldots \uplus \mathsf{h}_h$,
- $(\overline{\nu}',\mathsf{h}_0) \models^{\mathsf{SL}} \overline{\phi}$, and
- $(\overline{\nu}',\mathsf{h}_\ell) \models^{\mathsf{SL}}_{\overline{\mathcal{D}}} \overline{\mathsf{B}}^{\ell}_{\iota\ell}(y^\ell_1,\ldots,y^\ell_{\#(\mathsf{B}^\ell)},\eta(y^\ell_1),\ldots,\eta(y^\ell_{\#(\mathsf{B}^\ell)}))$, for all $\ell \in [1,h]$.

By the inductive hypothesis, there exist structures $\sigma_1,\ldots,\sigma_h$ and state maps $\rho_1,\ldots,\rho_h$, such that $\mathsf{h}_\ell \in \mathcal{G}_{\mathfrak{B}}(\sigma_\ell,\rho_\ell)$ and $(\sigma_\ell,\overline{\nu}',\rho_\ell) \models_{\mathcal{D}} \mathsf{B}^\ell(y^\ell_1,\ldots,y^\ell_{\#(\mathsf{B}^\ell)})$, for all $\ell \in [1,h]$. We define a structure σ_0 and a statemap ρ_0 as follows:

- $\sigma_0 \stackrel{\text{def}}{=} \langle C^{\sigma_0}_1,\ldots,C^{\sigma_0}_n,I^{\sigma_0}_1,\ldots,I^{\sigma_0}_m \rangle$ is such that, for all $i \in [1,n]$ and $j \in [1,m]$:

$$\begin{array}{rcl} C^{\sigma_0}_i & = & \begin{cases} \{\overline{\nu}(x_1)\} & \text{, if } \langle \mathsf{h}(\overline{\nu}(x_1)) \rangle_i = \overline{\nu}(x_1) \\ \emptyset & \text{, otherwise} \end{cases} \\ I^{\sigma_0}_j & = & \{\overline{\nu}'(\mathbf{y}) \mid I_j(\mathbf{y}) \in I^j_\phi(x_1)\} \end{array}$$

- $\rho_0(u,C_i) = q_s$ if $\langle \mathsf{h}_0(\overline{\nu}(x_1)) \rangle_{\mathrm{spos}(i,s)} = \overline{\nu}(x_1)$, for each $u \in \mathrm{dom}(\mathsf{h}_0)$ and $s \in [1,\|\mathbb{Q}\|]$, otherwise $\rho_0(u,C_i)$ is random.

Next, we prove that the composition $\sigma_q \bullet \sigma_r$ is defined, for all $q,r \in [0,h]$. To this end, we show:

- $C^{\sigma_q}_i \cap C^{\sigma_r}_i = \emptyset$, for all $i \in [1,n]$, by the following case split:
  - $q = 0$ and $r \in [1,h]$: by the definition of σ_0, we have $C^{\sigma_0}_i = \{\overline{\nu}(x_1)\}$ and $\overline{\nu}(x_1) \notin \mathrm{dom}(\mathsf{h}_r) = \mathrm{nodes}(\sigma_r)$, since $\mathsf{h}_r \in \mathcal{G}_{\mathfrak{B}}(\sigma_r,\rho)$. Moreover, $C^{\sigma_r}_i \subseteq \mathrm{nodes}(\sigma_r)$.
  - $q,r \in [1,h]$: because $\mathsf{h}_q \in \mathcal{G}_{\mathfrak{B}}(\sigma_q,\rho)$, $\mathsf{h}_r \in \mathcal{G}_{\mathfrak{B}}(\sigma_r,\rho)$ and $\mathrm{dom}(\mathsf{h}_q) \cap \mathrm{dom}(\mathsf{h}_r) = \emptyset$, we obtain $\mathrm{nodes}(\sigma_q) \cap \mathrm{nodes}(\sigma_r) = \emptyset$. Moreover, $C^{\sigma_q}_i \subseteq \mathrm{nodes}(\sigma_q)$ and $C^{\sigma_r}_i \subseteq \mathrm{nodes}(\sigma_r)$.

- $I_j^{\sigma_q} \cap I_j^{\sigma_r} = \emptyset$, for all $j \in [1,m]$, by the following case split:
  - $q = 0$ and $r \in [1,h]$: let $\mathbf{u} \in I_j^{\sigma_0}$ be an arbitrary interaction. By the definition of σ_0, we have $\overline{\nu}(x_1) \in \mathbf{u}$. Since $\mathcal{D}$ is progressing, $x_1 \notin \{y_1^r, \ldots, y_{\#(\mathsf{B}^r)}^r\}$ and, moreover, because we assumed $\overline{\nu}$ to be injective over $\mathrm{fv}(\psi)$, $\overline{\nu}(x_1)$ cannot occur in an interaction from σ_r.
  - $q, r \in [1,h]$: because $I_j^{\sigma_q} \subseteq \mathrm{nodes}(\sigma_q) = \mathrm{dom}(\mathsf{h}_q)$, $I_j^{\sigma_r} \subseteq \mathrm{nodes}(\sigma_r) = \mathrm{dom}(\mathsf{h}_r)$ and $\mathrm{dom}(\mathsf{h}_q) \cap \mathrm{dom}(\mathsf{h}_r) = \emptyset$.

Consequently, the composition $\sigma \stackrel{\mathsf{def}}{=} \sigma_0 \bullet \ldots \bullet \sigma_h$ is defined. Moreover, we define $\rho(u, C_i) \stackrel{\mathsf{def}}{=} \rho_\ell(u, C_i)$ if $u \in \mathrm{dom}(\mathsf{h}_\ell)$, for $\ell \in [1,h]$ and $\rho(u,C_i)$ is random, for all $u \notin \mathrm{dom}(\mathsf{h})$ and all $i \in [1,n]$. We conclude by proving the following points:

- $\mathsf{h} \in \mathcal{G}_{\mathfrak{B}}(\sigma, \rho)$: we show that $\mathrm{dom}(\mathsf{h}) = \bigcup_{\ell=0}^{h} \mathrm{dom}(\mathsf{h}_\ell) = \mathrm{nodes}(\sigma)$, as required by Def. 15. The points (1-3) are an easy check, by the definition of σ_0 and the fact that $\mathsf{h}_\ell \in \mathcal{G}_{\mathfrak{B}}(\sigma_\ell, \rho_\ell)$, for all $\ell \in [1,h]$. "$\subseteq$" Let $u \in \mathrm{dom}(\mathsf{h})$ be an index. If $u \in \mathrm{dom}(\mathsf{h}_0)$, then $u = \overline{\nu}(x_1)$. Since $\mathcal{D}$ is progressing, x_1 occurs in a component or interaction atom in ϕ, hence $u \in \mathrm{nodes}(\sigma_0)$, by the definition of σ_0. Else, $u \in \mathrm{dom}(\mathsf{h}_\ell)$, for some $\ell \in [1,h]$, thus $u \in \mathrm{nodes}(\sigma_\ell)$, because $\mathsf{h}_\ell \in \mathcal{G}_{\mathfrak{B}}(\sigma_\ell, \rho_\ell)$. "$\supseteq$" Let $u \in \mathrm{nodes}(\sigma) = \bigcup_{\ell=0}^{h} \mathrm{nodes}(\sigma_\ell)$ be an index. If $u \in \mathrm{nodes}(\sigma_\ell)$, for some $\ell \in [1,h]$, we obtain $u \in \mathrm{dom}(\mathsf{h}_\ell) \subseteq \mathrm{dom}(\mathsf{h})$, because $\mathsf{h}_\ell \in \mathcal{G}_{\mathfrak{B}}(\sigma_\ell, \rho_\ell)$. Otherwise, $u \in \mathrm{nodes}(\sigma_0) \setminus \left(\bigcup_{\ell=1}^{h} \mathrm{nodes}(\sigma_\ell)\right)$. If $u \in C_i^{\sigma_0}$, for some $i \in [1,n]$, then $u = \overline{\nu}(x_1) \in \mathrm{dom}(\mathsf{h}_0) \subseteq \mathrm{dom}(\mathsf{h})$, by the definition of σ_0. Else, u occurs in an interaction $\mathbf{u} \in I_j^{\sigma_0}$, for some $j \in [1,m]$. By the definition of σ_0, there exists an interaction atom $I_j(\mathbf{y}) \in I_\phi^j(x_1)$, such that $u = \overline{\nu}'(y)$, for some variable in $\mathbf{y}$, different than x_1. Because $\mathcal{D}$ is progressing, we have $y = y_s^\ell$, for some $\ell \in [1,h]$ and $s \in [1, \#(\mathsf{B}^\ell)]$ and y_s^ℓ occurs in a component or interaction atom from the complete unfolding of $\mathsf{B}^\ell(y_1^\ell, \ldots, y_{\#(\mathsf{B}^\ell)}^\ell)$ using the rules in $\mathcal{D}$. Hence $u = \overline{\nu}'(y_s^\ell) \in \mathrm{nodes}(\sigma_\ell)$, which contradicts with the choice of u.
- $(\sigma, \overline{\nu}, \rho) \models^{\mathsf{SL}}_{\mathcal{D}} \mathsf{A}(x_1, \ldots, x_{\#(\mathsf{A})})$: by the inductive hypothesis, $(\overline{\nu}', \sigma_\ell) \models_{\mathcal{D}} \mathsf{B}^\ell(y_1^\ell, \ldots, y_{\#(\mathsf{B}^\ell)}^\ell)$, for all $\ell \in [1,h]$ and we are left with proving that $(\overline{\nu}', \sigma_0) \models \phi$. Because $\mathcal{D}$ is progressing the only component atoms in ϕ are of the form $C_i(x_1)$, $i \in [1,n]$ and the only interaction atoms in ϕ are of the form $I_j(\mathbf{y})$, with $x \in \mathbf{y}$. The conclusion follows from the definition of σ_0. $\square$

Theorem 4 *If $\mathcal{D}$ is progressing and $\lambda_{\mathsf{A}(x_1,\ldots,x_{\#(\mathsf{A})})}$-connected, then the entailment $\mathsf{A}(x_1, \ldots, x_{\#(\mathsf{A})}) \models_{\mathcal{D}} \exists y_1 \ldots \exists y_r \cdot \bigvee_{\ell=1}^{h} \mathsf{B}^\ell(z_1^\ell, \ldots, z_{\#(\mathsf{B}^\ell)}^\ell)$ is 2EXP-complete.*

Proof. We prove a many-one reduction from CL to SL entailments:

$$\begin{array}{rcl} \mathsf{A}(x_1, \ldots, x_{\#(\mathsf{A})}) & \models_{\mathcal{D}} & \exists y_1 \ldots \exists y_r \cdot \bigvee_{\ell=1}^{h} \mathsf{B}^\ell(z_1^\ell, \ldots, z_{\#(\mathsf{B}^\ell)}^\ell) \iff \\ \overline{\mathsf{A}}_\iota(x_1, \ldots, x_k, \eta(x_1), \ldots, \eta(x_k)) & \models^{\mathsf{SL}}_{\mathcal{D}} & \exists y_1 \ldots \exists y_r \cdot \bigvee_{\ell=1}^{h} \bigvee_{\iota' : [1,\#(\mathsf{B}^\ell)] \times [1,m] \to 2^{[0,\mathfrak{B}-1]}} \\ & & \qquad \overline{\mathsf{B}}^\ell_{\iota'}(z_1^\ell, \ldots, z_{\#(\mathsf{B}^\ell)}^\ell, \eta(z_1^\ell), \ldots, \eta(z_{\#(\mathsf{B}^\ell)}^\ell)) \end{array}$$

for each mapping $\iota : [1,\#(\mathsf{A})] \times [1,m] \to 2^{[0,\mathfrak{B}-1]}$.

"$\Rightarrow$" Let $(\overline{\nu},\mathsf{h})$ be a $\overline{\mathcal{D}}$-model of $\mathsf{A}_\iota(x_1,\ldots,x_{\#(\mathsf{A})},\eta(x_1),\ldots,\eta(x_{\#(\mathsf{A})}))$, for some mapping ι. By Lemma 5, we have $\mathsf{h}(\overline{\nu}(x_i)) = \overline{\nu}(\eta(x_i))$, for all $i \in [1,\#(\mathsf{A})]$ and, moreover, there exists a structure σ and a state map ρ, such that $\mathsf{h} \in \mathcal{G}_{\mathfrak{B}}(\sigma,\rho)$ and $(\sigma,\overline{\nu},\rho) \models_{\mathcal{D}} \mathsf{A}(x_1,\ldots,x_{\#(\mathsf{A})})$. By the hypothesis, we obtain $(\sigma,\overline{\nu},\rho) \models_{\mathcal{D}} \exists y_1 \ldots \exists y_r \,.\, \bigvee_{\ell=1}^{h} \mathsf{B}^\ell(z_1^\ell,\ldots,z^\ell_{\#(\mathsf{B}^\ell)})$, hence there exist indices $u_1,\ldots,u_r \in \mathbb{U}$, such that $(\sigma,\overline{\nu}',\rho) \models \mathsf{B}^\ell(z_1^\ell,\ldots,z^\ell_{\#(\mathsf{B}^\ell)})$, for some $\ell \in [1,h]$, where $\overline{\nu}' \stackrel{\mathsf{def}}{=} \overline{\nu}[y_1 \leftarrow u_1,\ldots,y_r \leftarrow u_r]$. By Lemma 4, there exists a mapping ι' and a store $\overline{\nu}''$, such that $\overline{\nu}''(z_i^\ell) = \overline{\nu}'(z_i^\ell)$ and $\mathsf{h}(\overline{\nu}'(z_i^\ell)) = \overline{\nu}''(\eta(z_i^\ell))$, for all $i \in [1,\#(\mathsf{B}^\ell)]$ and, moreover, $(\overline{\nu}'',\mathsf{h}) \models^{\mathsf{SL}}_{\overline{\mathcal{D}}} \overline{\mathsf{B}}^\ell_{\iota'}(z_1^\ell,\ldots,z^\ell_{\#(\mathsf{B}^\ell)},\eta(z_1^\ell),\ldots,\eta(z^\ell_{\#(\mathsf{B}^\ell)}))$. We conclude this direction by observing that $\overline{\nu}''$ agrees with $\overline{\nu}$ over $x_1,\ldots,x_k$, because $\overline{\nu}''(\eta(x_i)) = \mathsf{h}(\overline{\nu}''(x_i)) = \mathsf{h}(\overline{\nu}(x_i)) = \overline{\nu}(\eta(x_i))$, for all $i \in [1,\#(\mathsf{A})]$, leading to the required $(\overline{\nu},\mathsf{h}) \models_{\mathcal{D}} \exists y_1 \ldots \exists y_r \,.\overline{\mathsf{B}}^\ell_{\iota'}(z_1^\ell,\ldots,z^\ell_{\#(\mathsf{B}^\ell)},\eta(z_1^\ell),\ldots,\eta(z^\ell_{\#(\mathsf{B}^\ell)}))$.

"$\Leftarrow$" Let (σ,ν,ρ) be a $\mathcal{D}$-model of $\mathsf{A}(x_1,\ldots,x_{\#(\mathsf{A})})$ and let $\mathsf{h} \in \mathcal{G}_{\mathfrak{B}}(\sigma,\rho)$. Clearly, such a heap always exists, an effective construction is possible following Def. 15. By Lemma 4, there exists a store $\overline{\nu}$, such that $\overline{\nu}(x_i) = \nu(x_i)$ and $\overline{\nu}(\eta(x_i)) = \mathsf{h}(\overline{\nu}(x_i))$, for all $i \in [1,\#(\mathsf{A})]$ and a mapping ι, such that $(\overline{\nu},\mathsf{h}) \models^{\mathsf{SL}}_{\overline{\mathcal{D}}} \overline{\mathsf{A}}_\iota(x_1,\ldots,x_{\#(\mathsf{A})},\eta(x_1),\ldots,\eta(x_{\#(\mathsf{A})}))$. By the hypothesis, we obtain:

$$(\overline{\nu},\mathsf{h}) \models^{\mathsf{SL}}_{\overline{\mathcal{D}}} \exists y_1 \ldots \exists y_r \,.\, \bigvee_{\ell=1}^{h} \bigvee_{\iota':[1,m]\times[1,\#(\mathsf{B}^\ell)]\to 2^{[0,\mathfrak{B}-1]}} \overline{\mathsf{B}}^\ell_{\iota'}(z_1^\ell,\ldots,z^\ell_{\#(\mathsf{B}^\ell)},\eta(z_1^\ell),\ldots,\eta(z^\ell_{\#(\mathsf{B}^\ell)}))$$

hence there exist indices $u_1,\ldots,u_r \in \mathbb{U}$, such that $(\overline{\nu}',\mathsf{h}) \models^{\mathsf{SL}}_{\overline{\mathcal{D}}} \overline{\mathsf{B}}^\ell_{\iota'}(z_1^\ell,\ldots,z^\ell_{\#(\mathsf{B}^\ell)},\eta(z_1^\ell),\ldots,\eta(z^\ell_{\#(\mathsf{B}^\ell)}))$, where $\overline{\nu}' = \overline{\nu}[y_1 \leftarrow u_1,\ldots,y_\ell \leftarrow u_\ell]$, for some $\ell \in [1,h]$ and some mapping ι'. By Lemma 5, there exists a structure σ' and a state map ρ', such that $\mathsf{h} \in \mathcal{G}_{\mathfrak{B}}(\sigma',\rho')$ and $(\sigma',\overline{\nu}',\rho') \models_{\mathcal{D}} \mathsf{B}^\ell(z_1^\ell,\ldots,z^\ell_{\#(\mathsf{B}^\ell)})$. Since $\mathsf{h} \in \mathcal{G}_{\mathfrak{B}}(\sigma,\rho) \cap \mathcal{G}_{\mathfrak{B}}(\sigma',\rho')$, we conclude $\sigma = \sigma'$ and $\rho = \rho'$, by Def. 15, thus $(\sigma,\overline{\nu}',\rho) \models_{\mathcal{D}} \mathsf{B}^\ell(z_1^\ell,\ldots,z^\ell_{\#(\mathsf{B}^\ell)})$, leading to $(\sigma,\overline{\nu},\rho) \models_{\mathcal{D}} \exists y_1 \ldots \exists y_r \,.\, \bigvee_{\ell=1}^{h} \mathsf{B}^\ell(z_1^\ell,\ldots,z^\ell_{\#(\mathsf{B}^\ell)})$.

Moreover, $\overline{\mathcal{D}}$ is $\lambda_{\overline{\mathsf{A}}_\iota(x_1,\ldots,x_{\#(\overline{\mathsf{A}}_\iota)},\eta(x_1),\ldots,\eta(x_{\#(\overline{\mathsf{A}}_\iota)}))}$-connected if $\mathcal{D}$ is $\lambda_{\mathsf{A}(x_1,\ldots,x_{\#(\mathsf{A})})}$-connected. We compute the upper bound given by the above reduction. We use the result of [**?**, Theorem 32], that gives a $2^{2^{\mathsf{poly}(\mathrm{width}(\overline{\mathcal{D}})\cdot\log \mathrm{size}(\overline{\mathcal{D}}))}}$ upper bound for SL entailments of the form $\phi \models^{\mathsf{SL}}_{\overline{\mathcal{D}}} \exists y_1 \ldots \exists y_r \,.\, \bigvee_{\ell=1}^{h} \psi^\ell$, where $\mathsf{poly}(x)$ is a polynomial function and:

- $\phi,\psi^1,\ldots,\psi^h$ are quantifier-free formulæ, and assume w.l.o.g. that $\max(\mathrm{size}(\phi),\mathrm{size}(\psi^1),\ldots,\mathrm{size}(\psi^h)) = O(\mathrm{width}(\overline{\mathcal{D}}))$ and $\mathrm{size}(\phi) + \sum_{\ell=1}^{h} \mathrm{size}(\psi^\ell) = O(\mathrm{size}(\overline{\mathcal{D}}))$,
- $\overline{\mathcal{D}}$ is progressing and λ_ϕ-connected, and
- each equational atom in a rule $\overline{\mathsf{A}}(x_1,\ldots,x_{\#(\overline{\mathsf{A}})}) \leftarrow_{\overline{\mathcal{D}}} \phi$ is of the form $x \doteq y$ or $x \not\doteq y$, where $\{x,y\} \cap \{x_i \mid i \in \lambda_\phi(\mathsf{A})\} \neq \emptyset$. The reduction from [**?**, Theorem 31] is used to remove equalities and, moreover, our reduction introduces no disequalities.

By Prop. 5, we have $\mathfrak{B} = \mathcal{O}(\text{size}(\mathcal{D})^c)$, for a constant $c \geq 1$ and $\mathfrak{K} = \text{pos}_{\mathfrak{B}}(m,0,n,0) = \mathcal{O}(\text{size}(\mathcal{D})^c)$, by Assumption 3. We compute:

$$\text{width}(\overline{\mathcal{D}}) \leq (\mathfrak{K}+1)\cdot\text{width}(\mathcal{D}) = \mathcal{O}(\text{size}(\mathcal{D})^c \cdot \text{width}(\mathcal{D})) = \mathcal{O}(\text{size}(\mathcal{D})^{c+1})$$

Note that there are $2^{\mathfrak{B}\cdot m\cdot\text{width}(\mathcal{D})} = 2^{\mathcal{O}(\text{size}(\mathcal{D})^{c+1}\cdot\text{width}(\mathcal{D}))} = 2^{\mathcal{O}(\text{size}(\mathcal{D})^{c+2})}$ mappings $\iota : [1,m]\times[1,\#(\mathsf{A})] \rightarrow 2^{[0,\mathfrak{B}-1]}$. We compute:

$$\begin{array}{rcl} \text{size}(\overline{\mathcal{D}}) & \leq & \|\overline{\mathcal{D}}\|\cdot\text{width}(\overline{\mathcal{D}}) \\ & = & \|\mathcal{D}\|\cdot\text{width}(\overline{\mathcal{D}})\cdot 2^{\mathcal{O}(\text{size}(\mathcal{D})^{c+2})}\cdot\text{width}(\overline{\mathcal{D}}) \\ & = & \text{size}(\mathcal{D})^{2c+3}\cdot 2^{\mathcal{O}(\text{size}(\mathcal{D})^{c+2})} \end{array}$$

Using [**?**, Theorem 32], we obtain a $2^{2^{\mathsf{poly}(\text{size}(\mathcal{D}))}}$ upper bound.

The lower bound is obtained by reduction from the SL entailment problem $\overline{\mathsf{A}}(x_1,\ldots,x_k) \models^{\mathsf{SL}}_{\overline{\mathcal{D}}} \overline{\mathsf{B}}(x_1,\ldots,x_k)$, where $\overline{\mathcal{D}}$ is progressing and $\lambda_{\overline{\mathsf{A}}(x_1,\ldots,x_k)}$-connected [**?**, Theorem 18]. The idea of the reduction is to encode each SL atomic proposition of the form $x \mapsto (y_1,\ldots,y_{\mathfrak{K}})$ by the CL formula $C(x) * I(x,y_1,\ldots,y_{\mathfrak{K}})$, where C is a component type and I is an interaction type of arity $\mathfrak{K}+1$. Then each $\overline{\mathcal{D}}$-model (ν,h) of a SL predicate atom $\overline{\mathsf{A}}(x_1,\ldots,x_{\#(\overline{\mathsf{A}})})$ is represented by a set of configurations (σ,ν,ρ), sharing the same structure σ over the signature $\mathfrak{S} = \langle C, I\rangle$, such that $C^\sigma = \text{dom}(\mathsf{h})$ and $I^\sigma = \{\mathsf{h}(u) \mid u \in \text{dom}(\mathsf{h})\}$ (the state map in these configurations is random). Moreover, if $\overline{\mathcal{D}}$ is progressing and $\lambda_{\overline{\mathsf{A}}(x_1,\ldots,x_k)}$-connected, then the CL SID $\mathcal{D}$ obtained from the reduction is progressing and $\lambda_{\overline{\mathsf{A}}(x_1,\ldots,x_k)}$-connected. Since the reduction takes polynomial time, we obtain a $\mathsf{2EXP}$-hard lower bound. □

Proposition 6 *The problem* is a given symbolic configuration tight? *is in* $\mathsf{3EXP}$.

Proof. For a given a predicateless symbolic configuration ϕ and $x \in \text{fv}(\phi)$, we consider the following sets of ports:

- $prov(x,\phi) \stackrel{\text{def}}{=} \bigcup\{ports(C_j) \mid C_j(y) \text{ occurs in } \phi,\ y \simeq_\phi x\}$,
- $req(x,\phi) \stackrel{\text{def}}{=} \{\langle ports(I_j)\rangle_k \mid I_j(\mathbf{y}) \text{ occurs in } \varphi,\ \langle\mathbf{y}\rangle_k \simeq_\phi x\}$.

We build a new SID $\mathcal{D}^\sharp$, by annotating each predicate symbol $\mathsf{A} \in \mathbb{A}$ with a tuple of sets of ports $\mathbf{p}$ provided by a component, in each complete unfolding of an annotated atom $\mathsf{A}_{\mathbf{p}}(x_1,\ldots,x_{\#(\mathsf{A})})$, i.e. $\langle\mathbf{p}\rangle_i = prov(x_i,\phi)$, for each $\mathsf{A}_{\mathbf{p}}(x_1,\ldots,x_{\#(\mathsf{A})}) \Leftarrow^{\circ}_{\mathcal{D}^\sharp} \exists\mathbf{z}\ .\ \phi$, where ϕ is a predicateless symbolic configuration, for each $i \in [1,\#(\mathsf{A})]$. Concretely, for each rule $\mathsf{A}(x_1,\ldots,x_{\#(\mathsf{A})}) \leftarrow_{\mathcal{D}} \exists z_1\ldots\exists z_k\ .\ \phi * \bigstar_{\ell=1}^{h} \mathsf{B}^\ell(\mathbf{y}^\ell)$ in $\mathcal{D}^\sharp$, where ϕ is a predicateless symbolic configuration, there are zero or more annotated rules $\mathsf{A}_{\mathbf{p}}(x_1,\ldots,x_{\#(\mathsf{A})}) \leftarrow_{\mathcal{D}^\sharp} \exists z_1\ldots\exists z_k\ .\ \phi * \bigstar_{\ell=1}^{h} \mathsf{B}^\ell_{\mathbf{p}^\ell}(\mathbf{y}^\ell)$, such that:

1. $\langle\mathbf{p}\rangle_i = prov(x_i,\phi) \cup \bigcup_{\ell=1}^{h}\{\langle\mathbf{p}^\ell\rangle_j \mid \langle\mathbf{y}^\ell\rangle_j \simeq_\phi x_i\}$, the ports provided by a component indexed by the value of the x_i parameter are the ones in $\langle\mathbf{p}\rangle_i$, in each rule that defines $\mathsf{A}_{\mathbf{p}}(x_1,\ldots,x_{\#(\mathsf{A})})$,

2. $req(x,\phi) \subseteq prov(x,\phi) \cup \bigcup_{\ell=1}^{h} \{\langle \mathbf{p}^\ell \rangle_j \mid \langle \mathbf{y}^\ell \rangle_j \simeq_\phi x\}$, for each $x \in \mathrm{fv}(\phi)$, i.e. each port required by an interaction in ϕ is provided either by ϕ or by a subsequent unfolding of a predicate atom; we say that $\phi * \mathbin{\circledast}_{\ell=1}^{h} \mathsf{B}^\ell_{\mathbf{p}^\ell}(\mathbf{y}^\ell)$ is *tightly annotated* in this case.

We prove the following facts:

Fact 4 *Each configuration* $\gamma \in [\![\mathsf{A}_{\mathbf{p}}(x_1,\ldots,x_{\#(\mathsf{A})})]\!]_{\mathcal{D}^\sharp}$ *is tight.*

Proof. Let γ be a $\mathcal{D}^\sharp$-model of $\mathsf{A}_{\mathbf{p}}(x_1,\ldots,x_{\#(\mathsf{A})})$, i.e. there exists a complete unfolding $\mathsf{A}_{\mathbf{p}}(x_1,\ldots,x_{\#(\mathsf{A})}) \Leftarrow^\circ_{\mathcal{D}^\sharp} \varphi$, such that $\gamma \models \varphi$. Proving that γ is tight is by induction on the length of this unfolding. $\square$

Fact 5 *For each complete unfolding of an annotated predicate atom* $\mathsf{A}_{\mathbf{p}}(x_1,\ldots,x_{\#(\mathsf{A})}) \Leftarrow^\circ_{\mathcal{D}^\sharp} \exists \mathbf{z} \,.\, \phi$*, where* ϕ *is a predicateless symbolic configuration, we have* $\langle \mathbf{p} \rangle_i = prov(x_i,\phi)$*, for each* $i \in [1,\#(\mathsf{A})]$.

Proof. The proof of this fact goes by induction on the length of the unfolding. $\square$

Fact 6 *Given a SID* $\mathcal{D}$ *and a predicateless symbolic configuration* ϕ*, the following hold:*

1. *The set* $\mathrm{def}(\mathcal{D})$ *is tight if and only if* $\mathsf{A}(x_1,\ldots,x_{\#(\mathsf{A})}) \models_{\mathcal{D}\cup\mathcal{D}^\sharp} \bigvee_{\mathbf{p}\in\mathbb{P}^{\#(\mathsf{A})}} \mathsf{A}_{\mathbf{p}}(x_1,\ldots,x_{\#(\mathsf{A})})$*, for each predicate atom* $\mathsf{A}(x_1,\ldots,x_{\#(\mathsf{A})}) \in \mathrm{def}(\mathcal{D})$.
2. *Provided that* $\mathrm{def}(\mathcal{D})$ *is tight, the formula* $\phi * \mathbin{\circledast}_{\ell=1}^{h} \mathsf{A}^\ell(\mathbf{x}^\ell)$ *is tight if and only if* $\phi * \mathbin{\circledast}_{\ell=1}^{h} \mathsf{A}^\ell_{\mathbf{p}^\ell}(\mathbf{x}^\ell)$ *is tightly annotated, for all* $\mathsf{A}^1_{\mathbf{p}^1}(\mathbf{x}^1),\ldots,\mathsf{A}^h_{\mathbf{p}^h}(\mathbf{x}^h) \in \mathrm{def}(\mathcal{D}^\sharp)$.

Proof. (1) We prove the following equivalent condition, for each $\mathsf{A}(x_1,\ldots,x_{\#(\mathsf{A})}) \in \mathrm{def}(\mathcal{D})$:

$$(\dagger)\ \{\gamma \in [\![\mathsf{A}(x_1,\ldots,x_{\#(\mathsf{A})})]\!]_{\mathcal{D}} \mid \gamma \text{ is tight}\} = \bigcup_{\mathbf{p}\in\mathbb{P}^{\#(\mathsf{A})}} [\![\mathsf{A}_{\mathbf{p}}(x_1,\ldots,x_{\#(\mathsf{A})})]\!]_{\mathcal{D}^\sharp}$$

The equivalence of (1) with (†) follows from Fact 4. We prove (†) below:
"$\subseteq$" Let $\gamma = (\sigma,\nu,\rho)$ be a tight $\mathcal{D}$-model of $\mathsf{A}(x_1,\ldots,x_{\#(\mathsf{A})})$. Then there exists a complete unfolding $\mathsf{A}(x_1,\ldots,x_{\#(\mathsf{A})}) \Leftarrow^\circ_{\mathcal{D}} \varphi$, such that $(\sigma,\nu,\rho) \models \varphi$ and prove that there exists a complete unfolding $\mathsf{A}_{\mathbf{p}}(x_1,\ldots,x_{\#(\mathsf{A})}) \Leftarrow^\circ_{\mathcal{D}^\sharp} \varphi$. First, we annotate the unfolding $\mathsf{A}(x_1,\ldots,x_{\#(\mathsf{A})}) \Leftarrow^\circ_{\mathcal{D}} \varphi$ with tuples of sets of ports, backwards, starting with the last step of the unfolding, say $\mathsf{B}(y_1,\ldots,y_{\#(\mathsf{B})}) \Leftarrow \exists \mathbf{z} \,.\, \psi$, where ψ is a predicateless symbolic configuration. In this case, we annotate B with the tuple $\mathbf{p}$, where $\langle \mathbf{p} \rangle_i \stackrel{\mathsf{def}}{=} prov(y_i,\psi)$, for all $i \in [1,\#(\mathsf{B})]$. Next, assume that $\mathsf{B}^1,\ldots,\mathsf{B}^h$ have been already annotated with tuples $\mathbf{p}^1,\ldots,\mathbf{p}^h$ in a step $\mathsf{A}(x_1,\ldots,x_{\#(\mathsf{A})}) \Leftarrow \exists z_1 \ldots \exists z_k \,.\, \phi * \mathbin{\circledast}_{\ell=1}^{h} \mathsf{B}^\ell(\mathbf{y}^\ell)$, where ϕ is a predicateless symbolic configuration and annotate A with the tuple $\mathbf{p}$, defined as $\langle \mathbf{p} \rangle_i \stackrel{\mathsf{def}}{=} prov(x_i,\phi) \cup \bigcup_{\ell=1}^{h} \{\langle \mathbf{p} \rangle_j \mid \langle \mathbf{y}^\ell \rangle_j \simeq_\phi x_i\}$, for all $i \in [1,\#(\mathsf{A})]$. The result is an annotated unfolding of $\mathsf{A}_{\mathbf{p}}(x_1,\ldots,x_{\#(\mathsf{A})})$ and we are left with proving that this is indeed an unfolding of $\mathcal{D}^\sharp$. This proof goes by induction on the length of the unfolding:

- In the base case, the unfolding consists of a single step $\mathsf{A}_{\mathbf{p}}(x_1,\ldots,x_{\#(\mathsf{A})}) \Leftarrow \exists \mathbf{z} \,.\, \psi$, where ψ is a predicateless symbolic configuration. Since $(\sigma,\nu,\rho) \models \exists \mathbf{z} \,.\, \psi$ and (σ,ν,ρ) is tight, we obtain that $\mathit{req}(x,\psi) \subseteq \mathit{prov}(x,\psi)$, for each $x \in \mathrm{fv}(\psi)$, thus $\mathsf{A}_{\mathbf{p}}(x_1,\ldots,x_{\#(\mathsf{A})}) \Leftarrow_{\mathcal{D}^\sharp} \exists \mathbf{z} \,.\, \psi$.

- For the inductive case, let $\mathsf{A}_{\mathbf{p}}(x_1,\ldots,x_{\#(\mathsf{A})}) \Leftarrow \exists z_1 \ldots \exists z_k \,.\, \phi * \mathbin{\text{⋇}}_{\ell=1}^{h} \mathsf{B}^{\ell}_{\mathbf{p}^\ell}(\mathbf{y}^\ell)$ be the first step of the unfolding $\mathsf{A}_{\mathbf{p}}(x_1,\ldots,x_{\#(\mathsf{A})}) \Leftarrow \varphi$, where ϕ is a predicateless symbolic configuration. Then there exists a step $\mathsf{A}(x_1,\ldots,x_{\#(\mathsf{A})}) \Leftarrow_{\mathcal{D}} \exists z_1 \ldots \exists z_k \,.\, \phi * \mathbin{\text{⋇}}_{\ell=1}^{h} \mathsf{B}^{\ell}(\mathbf{y}^\ell)$ and complete unfoldings $\mathsf{B}^{\ell}(\mathbf{y}^\ell) \Leftarrow^{\circ}_{\mathcal{D}} \varphi^\ell$, for all $\ell \in [1,h]$, such that $\varphi = \exists z_1 \ldots \exists z_k \,.\, \phi * \mathbin{\text{⋇}}_{\ell=1}^{h} \varphi^\ell$. Since $(\sigma,\nu,\rho) \models \varphi$, there exist indices $u_1,\ldots,u_k$ and structures $\sigma_0,\ldots,\sigma_k$, such that:

  - $(\sigma_0, \nu[x_1 \leftarrow u_1,\ldots,x_k \leftarrow u_k], \rho) \models \phi$,
  - $(\sigma_\ell, \nu[x_1 \leftarrow u_1,\ldots,x_k \leftarrow u_k], \rho) \models \varphi^\ell$, for all $\ell \in [1,h]$, and
  - $\sigma = \sigma_0 \bullet \ldots \bullet \sigma_\ell$.

  By the inductive hypothesis, there exist complete unfoldings $\mathsf{B}^{\ell}_{\mathbf{p}^\ell}(\mathbf{y}^\ell) \Leftarrow^{\circ}_{\mathcal{D}^\sharp} \varphi^\ell$, hence $(\sigma_\ell, \nu[x_1 \leftarrow u_1,\ldots,x_k \leftarrow u_k], \rho) \models_{\mathcal{D}^\sharp} \mathsf{B}^{\ell}_{\mathbf{p}^\ell}(\mathbf{y}^\ell)$, for all $\ell \in [1,h]$. By fact (2) above, the configuration $(\sigma_\ell, \nu[x_1 \leftarrow u_1,\ldots,x_k \leftarrow u_k], \rho)$ is tight, for each $\ell \in [1,h]$. Since (σ,ν,ρ) is tight, also $(\sigma, \nu[x_1 \leftarrow u_1,\ldots,x_k \leftarrow u_k], \rho)$ is tight. Assuming w.l.o.g. that $\varphi^\ell = \exists \mathbf{z}^\ell \,.\, \phi^\ell$, for all $\ell \in [1,h]$, we obtain:

  $$\begin{array}{lcl} \mathit{req}(x,\phi) & \subseteq & \mathit{prov}(x,\phi) \cup \bigcup_{\ell=1}^{h} \{\mathit{prov}(\langle \mathbf{y}^\ell \rangle_j, \phi^\ell) \mid \langle \mathbf{y}^\ell \rangle_j \simeq_\phi x\}, \text{ by Fact 5} \\ & = & \mathit{prov}(x,\phi) \cup \bigcup_{\ell=1}^{h} \{\langle \mathbf{p}^\ell \rangle_j \mid \langle \mathbf{y}^\ell \rangle_j \simeq_\phi x\} \end{array}$$

  for each $x \in \mathrm{fv}(\phi)$, thus $\mathsf{A}_{\mathbf{p}}(x_1,\ldots,x_{\#(\mathsf{A})}) \Leftarrow_{\mathcal{D}^\sharp} \exists z_1 \ldots \exists z_k \,.\, \phi * \mathbin{\text{⋇}}_{\ell=1}^{h} \mathsf{B}^{\ell}_{\mathbf{p}^\ell}(\mathbf{y}^\ell)$.

"$\supseteq$" Let $\gamma \in [\![\mathsf{A}_{\mathbf{p}}(x_1,\ldots,x_{\#(\mathsf{A})})]\!]_{\mathcal{D}^\sharp}$ be a configuration, for some $\mathbf{p} \in \mathbb{P}^{\#(\mathsf{A})}$. By fact (2) above, γ is tight. Moreover, there exists a complete unfolding $\mathsf{A}_{\mathbf{p}}(x_1,\ldots,x_{\#(\mathsf{A})}) \Leftarrow^{\circ}_{\mathcal{D}^\sharp} \varphi$, such that $\gamma \models \varphi$. By erasing the annotations from each predicate symbol in the unfolding, we obtain a complete unfolding $\mathsf{A}(x_1,\ldots,x_{\#(\mathsf{A})}) \Leftarrow^{\circ}_{\mathcal{D}^\sharp} \varphi$, hence $\gamma \in [\![\mathsf{A}(x_1,\ldots,x_{\#(\mathsf{A})})]\!]_{\mathcal{D}}$.

(2) "$\Rightarrow$" Let $\gamma = (\sigma,\nu,\rho)$ be a $\mathcal{D}$-model of $\phi * \mathbin{\text{⋇}}_{\ell=1}^{h} \mathsf{A}^{\ell}(\mathbf{x}^\ell)$. Suppose, for a contradiction, that there exists a free variable $x \in \mathrm{fv}(\phi)$ and a port $p \in \mathit{req}(x,\phi)$, such that $p \notin \mathit{prov}(x,\phi) \cup \bigcup_{\ell=1}^{h} \{\langle \mathbf{p}^\ell \rangle_j \mid \langle \mathbf{y}^\ell \rangle_j \simeq_\phi x\}$, for some annotated predicate atoms $\mathsf{A}^1_{\mathbf{p}^1}(\mathbf{x}^1),\ldots,\mathsf{A}^h_{\mathbf{p}^h}(\mathbf{x}^h) \in \mathrm{def}(\mathcal{D}^\sharp)$. By Fact 5, we reach a contradiction with the fact that γ is tight.
"$\Leftarrow$" Let $\gamma = (\sigma,\nu,\rho)$ be a $\mathcal{D}$-model of $\phi * \mathbin{\text{⋇}}_{\ell=1}^{h} \mathsf{A}^{\ell}(\mathbf{x}^\ell)$. Then there exist configurations $\gamma_0 = (\sigma_0,\nu,\rho),\ldots,\gamma_h = (\sigma_h,\nu,\rho)$, such that $\gamma = \gamma_0 \bullet \ldots \bullet \gamma_h$, $\gamma_0 \models \phi$ and $\gamma_\ell \models_{\mathcal{D}} \mathsf{A}^{\ell}(\mathbf{x}^\ell)$, for all $\ell \in [1,h]$. Let $\mathbf{u} \in I_j^\sigma$ be an interaction. If $\mathbf{u} \in I_j^{\sigma_\ell}$, for some $\ell \in [1,h]$, because $\mathsf{A}^{\ell}(\mathbf{x}^\ell)$ is tight, for each $k \in [1,\#(I_j)]$, there exists a component type C_i, such that $\langle \mathit{ports}(I_j) \rangle_k \in \mathit{ports}(C_i)$ and $\langle \mathbf{u} \rangle_k \in C_i^{\sigma_\ell} \subseteq C_i^\sigma$. Otherwise, we have $\mathbf{u} \in I_j^{\sigma_0}$ and let $\mathbf{p}^1,\ldots,\mathbf{p}^\ell \in \mathbb{P}^{\#(\mathsf{A})}$ be tuples of ports, such that $\gamma \models_{\mathcal{D}^\sharp} \phi * \mathbin{\text{⋇}}_{\ell=1}^{h} \mathsf{A}^{\ell}_{\mathbf{p}^\ell}(\mathbf{x}^\ell)$. Such tuples exist, by point (1) and the fact that all predicate atoms are tight. Since the formula $\phi * \mathbin{\text{⋇}}_{\ell=1}^{h} \mathsf{A}^{\ell}_{\mathbf{p}^\ell}(\mathbf{x}^\ell)$ is tighly annotated, by Fact 5, we obtain that, for each $k \in [1,\#(I_j)]$, there exists a component type C_i, such that $\langle \mathit{ports}(I_j) \rangle_k \in \mathit{ports}(C_i)$ and $\langle \mathbf{u} \rangle_k \in C_i^\sigma$. $\square$

By Fact 6, in order to decide whether a given symbolic configuration is tight, one must check the entailment $\mathsf{A}(x_1,\ldots,x_{\#(\mathsf{A})}) \models_{\mathcal{D}\cup\mathcal{D}^\sharp} \bigvee_{\mathbf{p}\in\mathbb{P}^{\#(\mathsf{A})}} \mathsf{A}_{\mathbf{p}}(x_1,\ldots,x_{\#(\mathsf{A})})$, for each predicate atom $\mathsf{A}(x_1,\ldots,x_{\#(\mathsf{A})}) \in \mathrm{def}(\mathcal{D})$. We have $\mathrm{width}(\mathcal{D}^\sharp) = \mathrm{width}(\mathcal{D})$ and, since $\|\mathbb{P}\|$ is a constant, we obtain $\mathrm{size}(\mathcal{D}^\sharp) \leq \mathrm{size}(\mathcal{D}) \cdot \mathrm{width}(\mathcal{D}) \cdot 2^{\mathrm{width}(\mathcal{D})} = \mathrm{size}(\mathcal{D}) \cdot 2^{\mathcal{O}(\mathrm{width}(\mathcal{D}))}$. Using the upper bound from the proof of Theorem 4, we obtain the upper bound $2^{2^{\mathsf{poly}\left(\mathrm{size}(\mathcal{D})\cdot 2^{\mathrm{width}(\mathcal{D})}\right)}} = 2^{2^{2^{\mathsf{poly}(\mathrm{width}(\mathcal{D})\cdot\log \mathrm{size}(\mathcal{D}))}}}$. □

F Havoc Invariance Proofs for Trees

In order to shorten the following proofs, we introduce the rule (I†) that allows us to remove a disabled interaction atom α from the pre- and postcondition, the environment and the language if certain conditions hold.

Lemma 6 *Using the notation in §5, the following rule is sound:*

$$(\mathsf{I}\dagger)\ \frac{\eta \setminus \{\Sigma[\alpha]\} \triangleright \{\{\phi\}\}\ \mathsf{L}\ \{\{\psi\}\}}{\eta \triangleright \{\{\phi * \alpha\}\}\ \Sigma[\alpha] \cup \mathsf{L}\ \{\{\psi * \alpha\}\}}\quad \begin{array}{l}\alpha = I(x_1,\ldots,x_{\#(I)})\\ \Sigma[\alpha] \in \eta \setminus \mathrm{supp}(\mathsf{L})\\ \phi \dagger \alpha.\end{array}$$

Proof. We assume that ϕ and ψ are two symbolic configurations, η is an environment and $\alpha = I(x_1,\ldots,x_{\#(I)})$ an interaction atom. Furthermore, $\Sigma[\alpha] \in \eta \setminus \mathrm{supp}(\mathsf{L})$ and $\phi \dagger \alpha$. Then we can apply the rule (∪) first and the rules (C), (Σ) and (I) on the subtrees and obtain:

$$(\cup)\ \frac{(\mathsf{C})\ \dfrac{(\dagger)\ \dfrac{}{\eta \triangleright \{\{\phi * \alpha\}\}\ \Sigma[\alpha]\ \{\{\mathsf{false}\}\}}}{\eta \triangleright \{\{\phi * \alpha\}\}\ \Sigma[\alpha]\ \{\{\psi * \alpha\}\}} \qquad (\mathsf{I})\ \dfrac{\eta \setminus \{\Sigma[\alpha]\} \triangleright \{\{\phi\}\}\ \mathsf{L}\ \{\{\psi\}\}}{\eta \triangleright \{\{\phi * \alpha\}\}\ \mathsf{L}\ \{\{\psi * \alpha\}\}}}{\eta \triangleright \{\{\phi * \alpha\}\}\ \Sigma[\alpha] \cup \mathsf{L}\ \{\{\psi * \alpha\}\}.}$$

Hence the rule can by derived from the rules in Fig. 5.

F.1 Havoc Invariance of the Predicate Atom $\mathsf{tree}(x)$

The invariance of the predicate $\mathsf{tree}(x)$ is proven via the rules in Fig. 5. The proof is divided into subtrees labeled by letters. Backlinks are indicated by numbers and in each cycle in the proof tree the rule (LU) is applied at least once.

$$(*)\ \frac{(\mathbf{1})\ \dfrac{(\mathsf{LU})\ \dfrac{(\varepsilon)\ \dfrac{}{\emptyset \triangleright \{\{Node^{leaf\_idle}(x)\}\}}\ \varepsilon\ \{\{\mathsf{tree}(x)\}\} \qquad (\varepsilon)\ \dfrac{}{\emptyset \triangleright \{\{Node^{leaf\_busy}(x)\}\}}\ \varepsilon\ \{\{\mathsf{tree}(x)\}\} \qquad \dfrac{(\mathbf{A})}{\begin{array}{l}\{\Sigma[I_\ell(y,x)],\Sigma[I_r(z,x)],\Sigma[\mathsf{tree}(y)],\Sigma[\mathsf{tree}(z)]\}\\ \triangleright \{\{Node(x) * I_\ell(y,x) * I_r(z,x) * \mathsf{tree}(y) * \mathsf{tree}(z)\}\}\\ \Sigma[I_\ell(y,x)] \cup \Sigma[I_r(z,x)] \cup \Sigma[\mathsf{tree}(y)] \cup \Sigma[\mathsf{tree}(z)]\end{array}}\ \{\{\mathsf{tree}(x)\}\}}{\{\Sigma[\mathsf{tree}(x)]\} \triangleright \{\{\mathsf{tree}(x)\}\}\ \Sigma[\mathsf{tree}(x)]\ \{\{\mathsf{tree}(x)\}\}}}{\{\Sigma[\mathsf{tree}(x)]\} \triangleright \{\{\mathsf{tree}(x)\}\}\ \Sigma[\mathsf{tree}(x)]^*\ \{\{\mathsf{tree}(x)\}\}}$$

$$(\mathsf{C})\ \frac{(\mathsf{C})\ \dfrac{(\cup)\ \dfrac{\dfrac{}{\begin{array}{l}(\mathbf{B})\ \{\Sigma[I_\ell(y,x)],\Sigma[I_r(z,x)],\Sigma[\mathsf{tree}(y)],\Sigma[\mathsf{tree}(z)]\}\\ \triangleright \{\{Node(x) * I_\ell(y,x) * I_r(z,x) * \mathsf{tree}(y) * \mathsf{tree}(z)\}\}\\ \Sigma[I_\ell(y,x)]\end{array}}\ \{\{Node(x) * I_\ell(y,x) * I_r(z,x) * \mathsf{tree}(y) * \mathsf{tree}(z)\}\}\ \text{similar to } (\mathbf{B})}{\{\Sigma[I_\ell(y,x)],\Sigma[I_r(z,x)],\Sigma[\mathsf{tree}(y)],\Sigma[\mathsf{tree}(z)]\} \triangleright \{\{Node(x) * I_\ell(y,x) * I_r(z,x) * \mathsf{tree}(y) * \mathsf{tree}(z)\}\}} \qquad \dfrac{}{\begin{array}{l}(\mathbf{C})\ \{\Sigma[I_\ell(y,x)],\Sigma[I_r(z,x)],\Sigma[\mathsf{tree}(y)],\Sigma[\mathsf{tree}(z)]\}\\ \triangleright \{\{Node(x) * I_\ell(y,x) * I_r(z,x) * \mathsf{tree}(y) * \mathsf{tree}(z)\}\}\\ \Sigma[\mathsf{tree}(y)]\end{array}}\ \{\{Node(x) * I_\ell(y,x) * I_r(z,x) * \mathsf{tree}(y) * \mathsf{tree}(z)\}\}\ \text{similar to } (\mathbf{C})}{\Sigma[I_\ell(y,x)] \cup \Sigma[I_r(z,x)] \cup \Sigma[\mathsf{tree}(y)] \cup \Sigma[\mathsf{tree}(z)]\ \{\{Node(x) * I_\ell(y,x) * I_r(z,x) * \mathsf{tree}(y) * \mathsf{tree}(z)\}\}}}{(\mathbf{A})\ \{\Sigma[I_\ell(y,x)],\Sigma[I_r(z,x)],\Sigma[\mathsf{tree}(y)],\Sigma[\mathsf{tree}(z)]\} \triangleright \{\{Node(x) * I_\ell(y,x) * I_r(z,x) * \mathsf{tree}(y) * \mathsf{tree}(z)\}\}}}{\Sigma[I_\ell(y,x)] \cup \Sigma[I_r(z,x)] \cup \Sigma[\mathsf{tree}(y)] \cup \Sigma[\mathsf{tree}(z)]\ \{\{\mathsf{tree}(x)\}\}}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\{\Sigma[l_\ell(y,x)],\Sigma[\mathsf{tree}(y)]\}\triangleright\ \{\{Node(x)*l_\ell(y,x)*\mathsf{tree}(y)\}\}}\ \textbf{(D)}\quad \textbf{(E)}\quad \textbf{(F)}}{\Sigma[l_\ell(y,x)]\ \{\{Node(x)*l_\ell(y,x)*\mathsf{tree}(y)\}\}}\ (\mathrm{LU}) \qquad \dfrac{}{\{\Sigma[\mathsf{tree}(z)]\}\triangleright\ \{\{\mathsf{tree}(z)\}\}}\ (\varepsilon)}{\{\Sigma[l_\ell(y,x)],\Sigma[\mathsf{tree}(y)],\Sigma[\mathsf{tree}(z)]\}\triangleright\ \{\{Node(x)*l_\ell(y,x)*\mathsf{tree}(y)*\mathsf{tree}(z)\}\}\ \Sigma[l_\ell(y,x)]\ \{\{Node(x)*l_\ell(y,x)*\mathsf{tree}(y)*\mathsf{tree}(z)\}\}}\ (\triangleright\!\triangleleft)}{\textbf{(B)}\ \{\Sigma[l_\ell(y,x)],\Sigma[l_r(z,x)],\Sigma[\mathsf{tree}(y)],\Sigma[\mathsf{tree}(z)]\}\triangleright\ \{\{Node(x)*l_\ell(y,x)*l_r(z,x)*\mathsf{tree}(y)*\mathsf{tree}(z)\}\}}\ (\mathrm{I})$$
$$\Sigma[l_\ell(y,x)]\ \{\{Node(x)*l_\ell(y,x)*l_r(z,x)*\mathsf{tree}(y)*\mathsf{tree}(z)\}\}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\text{backlink to }\textbf{(1)}}{\{\Sigma[\mathsf{tree}(y)]\}\triangleright\ \{\{\mathsf{tree}(y)\}\}\ \Sigma[\mathsf{tree}(y)]\ \{\{\mathsf{tree}(y)\}\}}\qquad \dfrac{}{\emptyset\triangleright\ \{\{Node(x)*\mathsf{tree}(z)\}\}\ \varepsilon\ \{\{Node(x)*\mathsf{tree}(z)\}\}}\ (\varepsilon)}{\{\Sigma[\mathsf{tree}(y)]\}\triangleright\ \{\{Node(x)*\mathsf{tree}(y)*\mathsf{tree}(z)\}\}\ \Sigma[\mathsf{tree}(y)]\ \{\{Node(x)*\mathsf{tree}(y)*\mathsf{tree}(z)\}\}}\ (\triangleright\!\triangleleft)}{\textbf{(C)}\ \{\Sigma[l_\ell(y,x)],\Sigma[l_r(z,x)],\Sigma[\mathsf{tree}(y)],\Sigma[\mathsf{tree}(z)]\}\triangleright\ \{\{Node(x)*l_\ell(y,x)*l_r(z,x)*\mathsf{tree}(y)*\mathsf{tree}(z)\}\}}\ (\mathrm{I})$$
$$\Sigma[\mathsf{tree}(y)]\ \{\{Node(x)*l_\ell(y,x)*l_r(z,x)*\mathsf{tree}(y)*\mathsf{tree}(z)\}\}$$

$$\dfrac{\dfrac{}{\{\Sigma[l_\ell(y,x)]\}\triangleright\ \{\{Node(x)*l_\ell(y,x)*Node^{leaf\_idle}(y)\}\}\ \Sigma[l_\ell(y,x)]\ \{\{\mathsf{false}\}\}}\ (\dagger)}{\textbf{(D)}\ \{\Sigma[l_\ell(y,x)]\}\triangleright\ \{\{Node(x)*l_\ell(y,x)*Node^{leaf\_idle}(y)\}\}\ \Sigma[l_\ell(y,x)]\ \{\{Node(x)*l_\ell(y,x)*\mathsf{tree}(y)\}\}}\ (\mathrm{C})$$

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\{\Sigma[l_\ell(y,x)]\}\triangleright\ \{\{Node^{idle}(x)*l_\ell(y,x)*Node^{leaf\_busy}(y)\}\}}\ (\Sigma)}{\{\Sigma[l_\ell(y,x)]\}\triangleright\ \{\{Node^{idle}(x)*l_\ell(y,x)*Node^{leaf\_busy}(y)\}\}}\ (\mathrm{C}) \qquad \dfrac{\dfrac{}{\{\Sigma[l_\ell(y,x)]\}\triangleright\ \{\{Node^{q}(x)*l_\ell(y,x)*Node^{leaf\_busy}(y)\}\}}\ (\dagger)}{\{\Sigma[l_\ell(y,x)]\}\triangleright\ \{\{Node^{q}(x)*l_\ell(y,x)*Node^{leaf\_busy}(y)\}\}}\ (\mathrm{C})\ \text{for } q\neq idle}{\dfrac{\Sigma[l_\ell(y,x)]\ \{\{Node^{left}(x)*l_\ell(y,x)*Node^{leaf\_idle}(y)\}\}}{\Sigma[l_\ell(y,x)]\ \{\{Node(x)*l_\ell(y,x)*\mathsf{tree}(y)\}\}}\qquad \dfrac{\Sigma[l_\ell(y,x)]\ \{\{Node(x)*l_\ell(y,x)*\mathsf{tree}(y)\}\}}{\Sigma[l_\ell(y,x)]\ \{\{Node(x)*l_\ell(y,x)*\mathsf{tree}(y)\}\}}}\ (\vee)$$
$$\textbf{(E)}\ \{\Sigma[l_\ell(y,x)]\}\triangleright\ \{\{Node(x)*l_\ell(y,x)*Node^{leaf\_busy}(y)\}\}\ \Sigma[l_\ell(y,x)]\ \{\{Node(x)*l_\ell(y,x)*\mathsf{tree}(y)\}\}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\text{backlink to }\textbf{(D)}\qquad \text{backlink to }\textbf{(E)}}{\{\Sigma[l_\ell(y,x)]\}\triangleright\ \{\{Node(x)*l_\ell(y,x)*Node(y)\}\}}\ (\vee) \qquad \dfrac{}{\{\Sigma[\mathsf{tree}(v)],\Sigma[\mathsf{tree}(w)]\}\triangleright\ \{\{\mathsf{tree}(v)*\mathsf{tree}(w)\}\}}\ (\varepsilon)}{\Sigma[l_\ell(y,x)]\ \{\{Node(x)*l_\ell(y,x)*Node(y)\}\}\qquad \varepsilon\ \{\{\mathsf{tree}(v)*\mathsf{tree}(w)\}\}}\ (\triangleright\!\triangleleft)}{\{\Sigma[l_\ell(y,x)],\Sigma[\mathsf{tree}(v)],\Sigma[\mathsf{tree}(w)]\}\triangleright\ \{\{Node(x)*l_\ell(y,x)*Node(y)*\mathsf{tree}(v)*\mathsf{tree}(w)\}\}}$$
$$\dfrac{\Sigma[l_\ell(y,x)]\ \{\{Node(x)*l_\ell(y,x)*Node(y)*l_\ell(v,y)*l_r(w,y)*\mathsf{tree}(v)*\mathsf{tree}(w)\}\}}{\{\Sigma[l_\ell(y,x)],\Sigma[l_\ell(v,y)],\Sigma[l_r(w,y)],\Sigma[\mathsf{tree}(v)],\Sigma[\mathsf{tree}(w)]\}\triangleright\ \{\{Node(x)*l_\ell(y,x)*Node(y)*l_\ell(v,y)*l_r(w,y)*\mathsf{tree}(v)*\mathsf{tree}(w)\}\}}\ (\mathrm{I})$$
$$\dfrac{\Sigma[l_\ell(y,x)]\ \{\{Node(x)*l_\ell(y,x)*Node(y)*l_\ell(v,y)*l_r(w,y)*\mathsf{tree}(v)*\mathsf{tree}(w)\}\}}{\textbf{(F)}\ \{\Sigma[l_\ell(y,x)],\Sigma[l_\ell(v,y)],\Sigma[l_r(w,y)],\Sigma[\mathsf{tree}(v)],\Sigma[\mathsf{tree}(w)]\}\triangleright\ \{\{Node(x)*l_\ell(y,x)*Node(y)*l_\ell(v,y)*l_r(w,y)*\mathsf{tree}(v)*\mathsf{tree}(w)\}\}}\ (\mathrm{C})$$
$$\Sigma[l_\ell(y,x)]\ \{\{Node(x)*l_\ell(y,x)*\mathsf{tree}(y)\}\}$$

F.2 Havoc Invariance of the Predicate Atom $\mathsf{tree}_{idle}(x)$

The invariance of the predicate $\mathsf{tree}_{idle}(x)$ is proven via the rules in Fig. 5 and the proof is structured similar to the previous invariance proof.

$$\dfrac{\dfrac{\dfrac{}{\emptyset\triangleright\ \{\{Node^{leaf\_idle}(x)\}\}}\ (\varepsilon) \qquad \dfrac{\textbf{(A)}}{\{\Sigma[l_\ell(y,x)],\Sigma[l_r(z,x)],\Sigma[\mathsf{tree}_{idle}(y)],\Sigma[\mathsf{tree}_{idle}(z)]\}\triangleright\ \{\{Node^{idle}(x)*l_\ell(y,x)*l_r(z,x)*\mathsf{tree}_{idle}(y)*\mathsf{tree}_{idle}(z)\}\}}}{\varepsilon\ \{\{\mathsf{tree}_{idle}(x)\}\}\qquad \Sigma[l_\ell(y,x)]\cup\Sigma[l_r(z,x)]\cup\Sigma[\mathsf{tree}_{idle}(y)]\cup\Sigma[\mathsf{tree}_{idle}(z)]\ \{\{\mathsf{tree}_{idle}(x)\}\}}\ (\mathrm{LU})}{\dfrac{\textbf{(2)}\ \{\Sigma[\mathsf{tree}_{idle}(x)]\}\triangleright\ \{\{\mathsf{tree}_{idle}(x)\}\}\ \Sigma[\mathsf{tree}_{idle}(x)]\ \{\{\mathsf{tree}_{idle}(x)\}\}}{\{\Sigma[\mathsf{tree}_{idle}(x)]\}\triangleright\ \{\{\mathsf{tree}_{idle}(x)\}\}\ \Sigma[\mathsf{tree}_{idle}(x)]^*\ \{\{\mathsf{tree}_{idle}(x)\}\}}\ (*)$$

$$\dfrac{\dfrac{}{\textbf{(B)}\ \{\Sigma[I_\ell(y,x)],\Sigma[\mathsf{tree}_{idle}(y)],\Sigma[\mathsf{tree}_{idle}(z)]\}\ \triangleright\ \{\{Node^{idle}(x)*I_\ell(y,x)*\mathsf{tree}_{idle}(y)*\mathsf{tree}_{idle}(z)\}\}\ \Sigma[I_\ell(y,x)]}\quad \dfrac{}{\textbf{(C)}\ \{\Sigma[I_\ell(y,x)],\Sigma[\mathsf{tree}_{idle}(y)],\Sigma[\mathsf{tree}_{idle}(z)]\}\ \triangleright\ \{\{Node^{idle}(x)*I_\ell(y,x)*\mathsf{tree}_{idle}(y)*\mathsf{tree}_{idle}(z)\}\}\ \Sigma[\mathsf{tree}_{idle}(y)]}}{(\cup)\ \dfrac{\{\{Node^{idle}(x)*I_\ell(y,x)*\mathsf{tree}_{idle}(y)*\mathsf{tree}_{idle}(z)\}\}\qquad \{\{Node^{idle}(x)*I_\ell(y,x)*\mathsf{tree}_{idle}(y)*\mathsf{tree}_{idle}(z)\}\}\qquad \text{similar to } \textbf{(C)}}{\{\Sigma[I_\ell(y,x)],\Sigma[\mathsf{tree}_{idle}(y)],\Sigma[\mathsf{tree}_{idle}(z)]\}\ \triangleright\ \{\{Node^{idle}(x)*I_\ell(y,x)*\mathsf{tree}_{idle}(y)*\mathsf{tree}_{idle}(z)\}\}}$$

$$(\mathsf{I}\dagger)\ \dfrac{\Sigma[I_\ell(y,x)]\cup\Sigma[\mathsf{tree}_{idle}(y)]\cup\Sigma[\mathsf{tree}_{idle}(z)]\ \{\{Node^{idle}(x)*I_\ell(y,x)*\mathsf{tree}_{idle}(y)*\mathsf{tree}_{idle}(z)\}\}}{\{\Sigma[I_\ell(y,x)],\Sigma[I_r(z,x)],\Sigma[\mathsf{tree}_{idle}(y)],\Sigma[\mathsf{tree}_{idle}(z)]\}\ \triangleright\ \{\{Node^{idle}(x)*I_\ell(y,x)*I_r(z,x)*\mathsf{tree}_{idle}(y)*\mathsf{tree}_{idle}(z)\}\}}$$

$$(\mathsf{C})\ \dfrac{\Sigma[I_\ell(y,x)]\cup\Sigma[I_r(z,x)]\cup\Sigma[\mathsf{tree}_{idle}(y)]\cup\Sigma[\mathsf{tree}_{idle}(z)]\ \{\{Node^{idle}(x)*I_\ell(y,x)*I_r(z,x)*\mathsf{tree}_{idle}(y)*\mathsf{tree}_{idle}(z)\}\}}{\textbf{(A)}\ \{\Sigma[I_\ell(y,x)],\Sigma[I_r(z,x)],\Sigma[\mathsf{tree}_{idle}(y)],\Sigma[\mathsf{tree}_{idle}(z)]\}\ \triangleright\ \{\{Node^{idle}(x)*I_\ell(y,x)*I_r(z,x)*\mathsf{tree}_{idle}(y)*\mathsf{tree}_{idle}(z)\}\}}$$

$$\Sigma[I_\ell(y,x)]\cup\Sigma[I_r(z,x)]\cup\Sigma[\mathsf{tree}_{idle}(y)]\cup\Sigma[\mathsf{tree}_{idle}(z)]\ \{\{\mathsf{tree}_{idle}(x)\}\}$$

$$(\mathsf{LU})\ \dfrac{\textbf{(D)}\qquad\textbf{(E)}}{\{\Sigma[I_\ell(y,x)],\Sigma[\mathsf{tree}_{idle}(y)]\}\ \triangleright\ \{\{Node^{idle}(x)*I_\ell(y,x)*\mathsf{tree}_{idle}(y)\}\}}$$

$$(\bowtie)\ \dfrac{\Sigma[I_\ell(y,x)]\ \{\{Node^{idle}(x)*I_\ell(y,x)*\mathsf{tree}_{idle}(y)\}\}\qquad (\varepsilon)\ \dfrac{}{\{\Sigma[\mathsf{tree}_{idle}(z)]\}\ \triangleright\ \{\{\mathsf{tree}_{idle}(z)\}\}\ \varepsilon\ \{\{\mathsf{tree}_{idle}(z)\}\}}}{\textbf{(B)}\ \{\Sigma[I_\ell(y,x)],\Sigma[\mathsf{tree}_{idle}(y)],\Sigma[\mathsf{tree}_{idle}(z)]\}\ \triangleright\ \{\{Node^{idle}(x)*I_\ell(y,x)*\mathsf{tree}_{idle}(y)*\mathsf{tree}_{idle}(z)\}\}}$$

$$\Sigma[I_\ell(y,x)]\ \{\{Node^{idle}(x)*I_\ell(y,x)*\mathsf{tree}_{idle}(y)*\mathsf{tree}_{idle}(z)\}\}$$

$$(\varepsilon)\ \dfrac{}{\{\Sigma[\mathsf{tree}_{idle}(z)]\}\ \triangleright\ \{\{Node^{idle}(x)*\mathsf{tree}_{idle}(z)\}\}\ \varepsilon\ \{\{Node^{idle}(x)*\mathsf{tree}_{idle}(z)\}\}}\qquad \dfrac{\text{backlink to } \textbf{(2)}}{\{\Sigma[\mathsf{tree}_{idle}(y)]\}\ \triangleright\ \{\{\mathsf{tree}_{idle}(y)\}\}\ \Sigma[\mathsf{tree}_{idle}(y)]\ \{\{\mathsf{tree}_{idle}(y)\}\}}$$

$$(\bowtie)\ \dfrac{}{(\mathsf{I})\ \dfrac{\{\Sigma[\mathsf{tree}_{idle}(y)],\Sigma[\mathsf{tree}_{idle}(z)]\}\ \triangleright\ \{\{Node^{idle}(x)*\mathsf{tree}_{idle}(y)*\mathsf{tree}_{idle}(z)\}\}\ \Sigma[\mathsf{tree}_{idle}(y)]\ \{\{Node^{idle}(x)*\mathsf{tree}_{idle}(y)*\mathsf{tree}_{idle}(z)\}\}}{\textbf{(C)}\ \{\Sigma[I_\ell(y,x)],\Sigma[\mathsf{tree}_{idle}(y)],\Sigma[\mathsf{tree}_{idle}(z)]\}\ \triangleright\ \{\{Node^{idle}(x)*I_\ell(y,x)*\mathsf{tree}_{idle}(y)*\mathsf{tree}_{idle}(z)\}\}}}$$

$$\Sigma[\mathsf{tree}_{idle}(y)]\ \{\{Node^{idle}(x)*I_\ell(y,x)*\mathsf{tree}_{idle}(y)*\mathsf{tree}_{idle}(z)\}\}$$

$$(\mathsf{C})\ \dfrac{(\dagger)\ \dfrac{}{\{\Sigma[I_\ell(y,x)]\}\ \triangleright\ \{\{Node^{idle}(x)*I_\ell(y,x)*Node^{leaf\_idle}(y)\}\}\ \Sigma[I_\ell(y,x)]\ \{\{\mathsf{false}\}\}}}{\textbf{(D)}\ \{\Sigma[I_\ell(y,x)]\}\ \triangleright\ \{\{Node^{idle}(x)*I_\ell(y,x)*Node^{leaf\_idle}(y)\}\}\ \Sigma[I_\ell(y,x)]\ \{\{Node^{idle}(x)*I_\ell(y,x)*\mathsf{tree}_{idle}(y)\}\}}$$

$$(\dagger)\ \dfrac{}{\{\Sigma[I_\ell(y,x)]\}\ \triangleright\ \{\{Node^{idle}(x)*I_\ell(y,x)*Node^{idle}(y)\}\}}\qquad (\varepsilon)\ \dfrac{}{\{\Sigma[\mathsf{tree}_{idle}(v)],\Sigma[\mathsf{tree}_{idle}(w)]\}\ \triangleright\ \{\{\mathsf{tree}_{idle}(v)*\mathsf{tree}_{idle}(w)\}\}}$$

$$(\bowtie)\ \dfrac{\Sigma[I_\ell(y,x)]\ \{\{Node^{idle}(x)*I_\ell(y,x)*Node^{idle}(y)\}\}\qquad \varepsilon\ \{\{\mathsf{tree}_{idle}(v)*\mathsf{tree}_{idle}(w)\}\}}{\{\Sigma[I_\ell(y,x)],\Sigma[\mathsf{tree}_{idle}(v)],\Sigma[\mathsf{tree}_{idle}(w)]\}\ \triangleright\ \{\{Node^{idle}(x)*I_\ell(y,x)*Node^{idle}(y)*\mathsf{tree}_{idle}(v)*\mathsf{tree}_{idle}(w)\}\}}$$

$$(\mathsf{I})\ \dfrac{\Sigma[I_\ell(y,x)]\ \{\{Node^{idle}(x)*I_\ell(y,x)*Node^{idle}(y)*\mathsf{tree}_{idle}(v)*\mathsf{tree}_{idle}(w)\}\}}{\{\Sigma[I_\ell(y,x)],\Sigma[I_\ell(v,y)],\Sigma[I_r(w,y)],\Sigma[\mathsf{tree}_{idle}(v)],\Sigma[\mathsf{tree}_{idle}(w)]\}}$$

$$\triangleright\ \{\{Node^{idle}(x)*I_\ell(y,x)*Node^{idle}(y)*I_\ell(v,y)*I_r(w,y)*\mathsf{tree}_{idle}(v)*\mathsf{tree}_{idle}(w)\}\}$$

$$\Sigma[I_\ell(y,x)]$$

$$(\mathsf{C})\ \dfrac{\{\{Node^{idle}(x)*I_\ell(y,x)*Node^{idle}(y)*I_\ell(v,y)*I_r(w,y)*\mathsf{tree}_{idle}(v)*\mathsf{tree}_{idle}(w)\}\}}{\textbf{(E)}\ \{\Sigma[I_\ell(y,x)],\Sigma[I_\ell(v,y)],\Sigma[I_r(w,y)],\Sigma[\mathsf{tree}_{idle}(v)],\Sigma[\mathsf{tree}_{idle}(w)]\}}$$

$$\triangleright\ \{\{Node^{idle}(x)*I_\ell(y,x)*Node^{idle}(y)*I_\ell(v,y)*I_r(w,y)*\mathsf{tree}_{idle}(v)*\mathsf{tree}_{idle}(w)\}\}$$

$$\Sigma[I_\ell(y,x)]$$

$$\{\{Node^{idle}(x)*I_\ell(y,x)*\mathsf{tree}_{idle}(y)\}\}$$

F.3 Havoc Invariance of the Predicate Atom $\mathsf{tree}_{\neg idle}(x)$

The proof of the invariance of the predicate $\mathsf{tree}_{\neg idle}(x)$ is similar to the previous proofs.

$$\dfrac{\dfrac{(\varepsilon)\ \dfrac{}{\emptyset \triangleright \{\{Node^{leaf\_busy}(x)\}\}\ \varepsilon\ \{\{\mathsf{tree}_{\neg idle}(x)\}\}}}{(\mathrm{LU})\ \dfrac{\textbf{(A)}\quad\textbf{(B)}\quad \text{similar to } \textbf{(A)}}{\textbf{(3)}\ \{\Sigma[\mathsf{tree}_{\neg idle}(x)]\} \triangleright \{\{\mathsf{tree}_{\neg idle}(x)\}\}\ \Sigma[\mathsf{tree}_{\neg idle}(x)]\ \{\{\mathsf{tree}_{\neg idle}(x)\}\}}}}{(*)\ \dfrac{}{\{\Sigma[\mathsf{tree}_{\neg idle}(x)]\} \triangleright \{\{\mathsf{tree}_{\neg idle}(x)\}\}\ \Sigma[\mathsf{tree}_{\neg idle}(x)]^*\ \{\{\mathsf{tree}_{\neg idle}(x)\}\}}}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\textbf{(C)}}{\{\Sigma[I_r(z,x)], \Sigma[\mathsf{tree}_{idle}(y)], \Sigma[\mathsf{tree}_{\neg idle}(z)]\}}}{\triangleright \{\{Node^{left}(x) * I_r(z,x) * \mathsf{tree}_{idle}(y) * \mathsf{tree}_{\neg idle}(z)\}}}{\Sigma[I_r(z,x)]}}{(\cup)\ \dfrac{\{\{Node^{left}(x) * I_r(z,x) * \mathsf{tree}_{idle}(y) * \mathsf{tree}_{\neg idle}(z)\}}{\{\Sigma[I_r(z,x)], \Sigma[\mathsf{tree}_{idle}(y)], \Sigma[\mathsf{tree}_{\neg idle}(z)]\} \triangleright \{\{Node^{left}(x) * I_r(z,x) * \mathsf{tree}_{idle}(y) * \mathsf{tree}_{\neg idle}(z)\}}}{\Sigma[I_r(z,x)] \cup \Sigma[\mathsf{tree}_{idle}(y)] \cup \Sigma[\mathsf{tree}_{\neg idle}(z)]\ \{\{Node^{left}(x) * I_r(z,x) * \mathsf{tree}_{idle}(y) * \mathsf{tree}_{\neg idle}(z)\}}}$$

$$(\mathrm{I}\dagger)\ \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\textbf{(D)}}{\{\Sigma[I_r(z,x)], \Sigma[\mathsf{tree}_{idle}(y)], \Sigma[\mathsf{tree}_{\neg idle}(z)]\}}}{\triangleright \{\{Node^{left}(x) * I_r(z,x) * \mathsf{tree}_{idle}(y) * \mathsf{tree}_{\neg idle}(z)\}}}{\Sigma[\mathsf{tree}_{idle}(y)]}}{\Sigma[I_r(z,x)] \cup \Sigma[\mathsf{tree}_{idle}(y)] \cup \Sigma[\mathsf{tree}_{\neg idle}(z)]\ \{\{Node^{left}(x) * I_r(z,x) * \mathsf{tree}_{idle}(y) * \mathsf{tree}_{\neg idle}(z)\}}}{\{\Sigma[I_\ell(y,x)], \Sigma[I_r(z,x)], \Sigma[\mathsf{tree}_{idle}(y)], \Sigma[\mathsf{tree}_{\neg idle}(z)]\} \triangleright \{\{Node^{left}(x) * I_\ell(y,x) * I_r(z,x) * \mathsf{tree}_{idle}(y) * \mathsf{tree}_{\neg idle}(z)\}}$$

$$(\mathrm{C})\ \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\textbf{(E)}}{\{\Sigma[I_r(z,x)], \Sigma[\mathsf{tree}_{idle}(y)], \Sigma[\mathsf{tree}_{\neg idle}(z)]\}}}{\triangleright \{\{Node^{left}(x) * I_r(z,x) * \mathsf{tree}_{idle}(y) * \mathsf{tree}_{\neg idle}(z)\}}}{\Sigma[\mathsf{tree}_{\neg idle}(z)]}}{\Sigma[I_\ell(y,x)] \cup \Sigma[I_r(z,x)] \cup \Sigma[\mathsf{tree}_{idle}(y)] \cup \Sigma[\mathsf{tree}_{\neg idle}(z)]\ \{\{Node^{left}(x) * I_r(z,x) * \mathsf{tree}_{idle}(y) * \mathsf{tree}_{\neg idle}(z)\}}}{\textbf{(A)}\ \{\Sigma[I_\ell(y,x)], \Sigma[I_r(z,x)], \Sigma[\mathsf{tree}_{idle}(y)], \Sigma[\mathsf{tree}_{\neg idle}(z)]\} \triangleright \{\{Node^{left}(x) * I_\ell(y,x) * I_r(z,x) * \mathsf{tree}_{idle}(y) * \mathsf{tree}_{\neg idle}(z)\}}$$

$$\Sigma[I_\ell(y,x)] \cup \Sigma[I_r(z,x)] \cup \Sigma[\mathsf{tree}_{idle}(y)] \cup \Sigma[\mathsf{tree}_{\neg idle}(z)]\ \{\{\mathsf{tree}_{\neg idle}(x)\}\}$$

$$(\triangleright\triangleleft)\ \dfrac{\dfrac{}{\emptyset \triangleright \{\{Node^{right}(x)\}\}}\quad \dfrac{\text{backlink to } \textbf{(2)}}{\{\Sigma[\mathsf{tree}_{idle}(y)]\} \triangleright \{\{\mathsf{tree}_{idle}(y)\}\}}\quad \dfrac{\text{backlink to } \textbf{(2)}}{\{\Sigma[\mathsf{tree}_{idle}(z)]\} \triangleright \{\{\mathsf{tree}_{idle}(z)\}\}}}{\varepsilon\ \{\{Node^{right}(x)\}\}\quad \Sigma[\mathsf{tree}_{idle}(y)]\ \{\{\mathsf{tree}_{idle}(y)\}\}\quad \Sigma[\mathsf{tree}_{idle}(z)]\ \{\{\mathsf{tree}_{idle}(z)\}\}}$$

$$(\mathrm{I}\dagger)\ \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\textbf{(F)}\quad\textbf{(G)}\quad\textbf{(H)}\quad\textbf{(I)}}{\{\Sigma[I_r(z,x)], \Sigma[\mathsf{tree}_{\neg idle}(z)]\} \triangleright \{\{Node^{left}(x) * I_r(z,x) * \mathsf{tree}_{\neg idle}(z)\}}}{(\mathrm{LU})}\quad \dfrac{(\varepsilon)\ \dfrac{}{\{\Sigma[\mathsf{tree}_{idle}(y)]\} \triangleright \{\{\mathsf{tree}_{idle}(y)\}\}}}{\varepsilon\ \{\{\mathsf{tree}_{idle}(y)\}\}}}{(\triangleright\triangleleft)\ \dfrac{\Sigma[I_r(z,x)]\ \{\{Node^{left}(x) * I_r(z,x) * \mathsf{tree}_{\neg idle}(z)\}\}}{\textbf{(C)}\ \{\Sigma[I_r(z,x)], \Sigma[\mathsf{tree}_{idle}(y)], \Sigma[\mathsf{tree}_{\neg idle}(z)]\} \triangleright \{\{Node^{left}(x) * I_r(z,x) * \mathsf{tree}_{idle}(y) * \mathsf{tree}_{\neg idle}(z)\}}}{\Sigma[I_r(z,x)]\ \{\{Node^{left}(x) * I_r(z,x) * \mathsf{tree}_{idle}(y) * \mathsf{tree}_{\neg idle}(z)\}}$$

$$(\varepsilon)\ \dfrac{\dfrac{\dfrac{}{\{\Sigma[I_r(z,x)], \Sigma[\mathsf{tree}_{\neg idle}(z)]\} \triangleright \{\{Node^{left}(x) * I_r(z,x) * \mathsf{tree}_{\neg idle}(z)\}}\quad \dfrac{\text{backlink to } \textbf{(2)}}{\{\Sigma[\mathsf{tree}_{idle}(y)]\} \triangleright \{\{\mathsf{tree}_{idle}(y)\}\}}}{(\triangleright\triangleleft)\ \dfrac{\varepsilon\ \{\{Node^{left}(x) * I_r(z,x) * \mathsf{tree}_{\neg idle}(z)\}\}\quad \Sigma[\mathsf{tree}_{idle}(y)]\ \{\{\mathsf{tree}_{idle}(y)\}\}}{\textbf{(D)}\ \{\Sigma[I_r(z,x)], \Sigma[\mathsf{tree}_{idle}(y)], \Sigma[\mathsf{tree}_{\neg idle}(z)]\} \triangleright \{\{Node^{left}(x) * I_r(z,x) * \mathsf{tree}_{idle}(y) * \mathsf{tree}_{\neg idle}(z)\}}}{\Sigma[\mathsf{tree}_{idle}(y)]\ \{\{Node^{left}(x) * I_r(z,x) * \mathsf{tree}_{idle}(y) * \mathsf{tree}_{\neg idle}(z)\}}$$

$$(\varepsilon)\ \dfrac{\dfrac{\dfrac{}{\{\Sigma[\mathsf{tree}_{idle}(y)]\} \triangleright \{\{Node^{left}(x) * \mathsf{tree}_{idle}(y)\}\}\ \varepsilon\ \{\{Node^{left}(x) * \mathsf{tree}_{idle}(y)\}\}}\quad \dfrac{\text{backlink to } \textbf{(3)}}{\{\Sigma[\mathsf{tree}_{\neg idle}(z)]\} \triangleright \{\{\mathsf{tree}_{\neg idle}(z)\}\}\ \Sigma[\mathsf{tree}_{\neg idle}(z)]\ \{\{\mathsf{tree}_{\neg idle}(z)\}\}}}{(\triangleright\triangleleft)\ \dfrac{\{\Sigma[\mathsf{tree}_{idle}(y)], \Sigma[\mathsf{tree}_{\neg idle}(z)]\} \triangleright \{\{Node^{left}(x) * \mathsf{tree}_{idle}(y) * \mathsf{tree}_{\neg idle}(z)\}\}\ \Sigma[\mathsf{tree}_{\neg idle}(z)]\ \{\{Node^{left}(x) * \mathsf{tree}_{idle}(y) * \mathsf{tree}_{\neg idle}(z)\}\}}{(\mathrm{I})\ \dfrac{}{\textbf{(E)}\ \{\Sigma[I_r(z,x)], \Sigma[\mathsf{tree}_{idle}(y)], \Sigma[\mathsf{tree}_{\neg idle}(z)]\} \triangleright \{\{Node^{left}(x) * I_r(z,x) * \mathsf{tree}_{idle}(y) * \mathsf{tree}_{\neg idle}(z)\}}}$$

$$\Sigma[\mathsf{tree}_{\neg idle}(z)]\ \{\{Node^{left}(x) * I_r(z,x) * \mathsf{tree}_{idle}(y) * \mathsf{tree}_{\neg idle}(z)\}\}$$

$$\dfrac{(\Sigma)\ \dfrac{}{\{\Sigma[I_r(z,x)]\} \triangleright \{\{Node^{left}(x) * I_r(z,x) * Node^{leaf\_busy}(z)\}\}\ \Sigma[I_r(z,x)]\ \{\{Node^{right}(x) * I_r(z,x) * Node^{leaf\_idle}(z)\}\}}}{(C)\ \dfrac{}{\textbf{(F)}\ \{\Sigma[I_r(z,x)]\} \triangleright \{\{Node^{left}(x) * I_r(z,x) * Node^{leaf\_busy}(z)\}\}\ \Sigma[I_r(z,x)]\ \{\{Node^{left}(x) * I_r(z,x) * \mathsf{tree}_{\neg idle}(z)\}\}}}$$

$$\dfrac{(\dagger)\ \dfrac{}{\{\Sigma[I_r(z,x)]\} \triangleright \{\{Node^{left}(x) * I_r(z,x) * Node^{left}(z)\}\}}}{(C)\ \dfrac{\Sigma[I_r(z,x)]\ \{\{\mathsf{false}\}\}}{\{\Sigma[I_r(z,x)]\} \triangleright \{\{Node^{left}(x) * I_r(z,x) * Node^{left}(z)\}\}}}\qquad (\varepsilon)\ \dfrac{}{\{\Sigma[\mathsf{tree}_{idle}(v)], \Sigma[\mathsf{tree}_{\neg idle}(w)]\} \triangleright \{\{\mathsf{tree}_{idle}(v) * \mathsf{tree}_{\neg idle}(w)\}\}}$$

$$(\bowtie)\ \dfrac{\Sigma[I_r(z,x)]\ \{\{Node^{left}(x) * I_r(z,x) * Node^{left}(z)\}\} \qquad \varepsilon\ \{\{\mathsf{tree}_{idle}(v) * \mathsf{tree}_{\neg idle}(w)\}\}}{\{\Sigma[I_r(z,x)], \Sigma[\mathsf{tree}_{idle}(v)], \Sigma[\mathsf{tree}_{\neg idle}(w)]\} \triangleright \{\{Node^{left}(x) * I_r(z,x) * Node^{left}(z) * \mathsf{tree}_{idle}(v) * \mathsf{tree}_{\neg idle}(w)\}\}}$$

$$(I)\ \dfrac{\Sigma[I_r(z,x)]\ \{\{Node^{left}(x) * I_r(z,x) * Node^{left}(z) * \mathsf{tree}_{idle}(v) * \mathsf{tree}_{\neg idle}(w)\}\}}{\{\Sigma[I_r(z,x)], \Sigma[I_\ell(v,z)], \Sigma[I_r(w,z)], \Sigma[\mathsf{tree}_{idle}(v)], \Sigma[\mathsf{tree}_{\neg idle}(w)]\}}$$

$$\triangleright \{\{Node^{left}(x) * I_r(z,x) * Node^{left}(z) * I_\ell(v,z) * I_r(w,z) * \mathsf{tree}_{idle}(v) * \mathsf{tree}_{\neg idle}(w)\}\}$$

$$\Sigma[I_r(z,x)]$$

$$(C)\ \dfrac{\{\{Node^{left}(x) * I_r(z,x) * Node^{left}(z) * I_\ell(v,z) * I_r(w,z) * \mathsf{tree}_{idle}(v) * \mathsf{tree}_{\neg idle}(w)\}\}}{\textbf{(G)}\ \{\Sigma[I_r(z,x)], \Sigma[I_\ell(v,z)], \Sigma[I_r(w,z)], \Sigma[\mathsf{tree}_{idle}(v)], \Sigma[\mathsf{tree}_{\neg idle}(w)]\}}$$

$$\triangleright \{\{Node^{left}(x) * I_r(z,x) * Node^{left}(z) * I_\ell(v,z) * I_r(w,z) * \mathsf{tree}_{idle}(v) * \mathsf{tree}_{\neg idle}(w)\}\}$$

$$\Sigma[I_r(z,x)]$$

$$\{\{Node^{left}(x) * I_r(z,x) * \mathsf{tree}_{\neg idle}(z)\}\}$$

$$\dfrac{(\Sigma)\ \dfrac{}{\{\Sigma[I_r(z,x)]\} \triangleright \{\{Node^{left}(x) * I_r(z,x) * Node^{right}(z)\}\}}}{(C)\ \dfrac{\Sigma[I_r(z,x)]\ \{\{Node^{right}(x) * I_r(z,x) * Node^{idle}(z)\}\}}{\{\Sigma[I_r(z,x)]\} \triangleright \{\{Node^{left}(x) * I_r(z,x) * Node^{right}(z)\}\}}}\qquad (\varepsilon)\ \dfrac{}{\{\Sigma[\mathsf{tree}_{idle}(v)], \Sigma[\mathsf{tree}_{idle}(w)]\} \triangleright \{\{\mathsf{tree}_{idle}(v) * \mathsf{tree}_{idle}(w)\}\}}$$

$$(\bowtie)\ \dfrac{\Sigma[I_r(z,x)]\ \{\{Node^{right}(x) * I_r(z,x) * Node^{right}(z)\}\} \qquad \varepsilon\ \{\{\mathsf{tree}_{idle}(v) * \mathsf{tree}_{idle}(w)\}\}}{\{\Sigma[I_r(z,x)], \Sigma[\mathsf{tree}_{idle}(v)], \Sigma[\mathsf{tree}_{idle}(w)]\} \triangleright \{\{Node^{left}(x) * I_r(z,x) * Node^{right}(z) * \mathsf{tree}_{idle}(v) * \mathsf{tree}_{idle}(w)\}\}}$$

$$(I)\ \dfrac{\Sigma[I_r(z,x)]\ \{\{Node^{left}(x) * I_r(z,x) * Node^{right}(z) * \mathsf{tree}_{idle}(v) * \mathsf{tree}_{idle}(w)\}\}}{\{\Sigma[I_r(z,x)], \Sigma[I_\ell(v,z)], \Sigma[I_r(w,z)], \Sigma[\mathsf{tree}_{idle}(v)], \Sigma[\mathsf{tree}_{idle}(w)]\}}$$

$$\triangleright \{\{Node^{left}(x) * I_r(z,x) * Node^{right}(z) * I_\ell(v,z) * I_r(w,z) * \mathsf{tree}_{idle}(v) * \mathsf{tree}_{idle}(w)\}\}$$

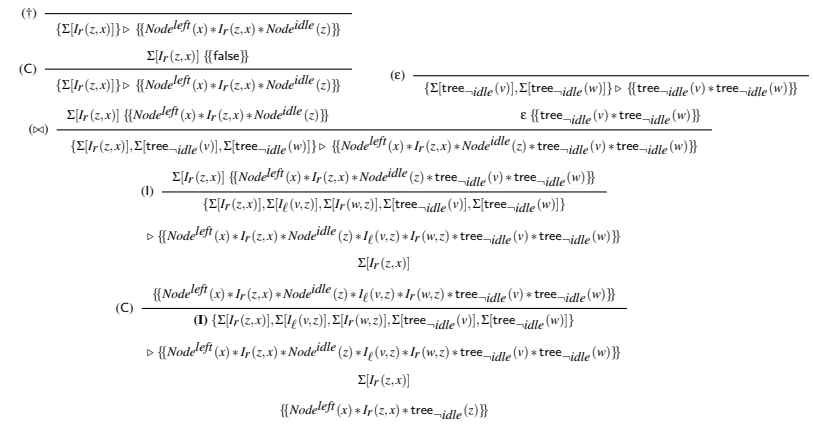
$$\Sigma[I_r(z,x)]$$

$$(C)\ \dfrac{\{\{Node^{left}(x) * I_r(z,x) * Node^{right}(z) * I_\ell(v,z) * I_r(w,z) * \mathsf{tree}_{idle}(v) * \mathsf{tree}_{idle}(w)\}\}}{\textbf{(H)}\ \{\Sigma[I_r(z,x)], \Sigma[I_\ell(v,z)], \Sigma[I_r(w,z)], \Sigma[\mathsf{tree}_{idle}(v)], \Sigma[\mathsf{tree}_{idle}(w)]\}}$$

$$\triangleright \{\{Node^{left}(x) * I_r(z,x) * Node^{right}(z) * I_\ell(v,z) * I_r(w,z) * \mathsf{tree}_{idle}(v) * \mathsf{tree}_{idle}(w)\}\}$$

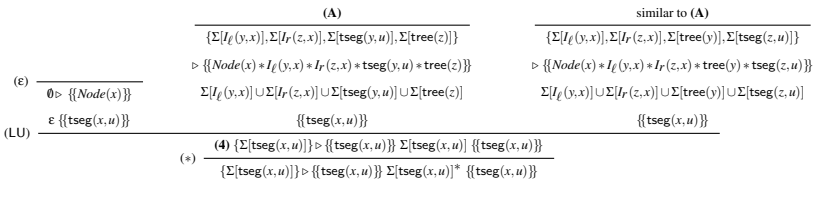
$$\Sigma[I_r(z,x)]$$

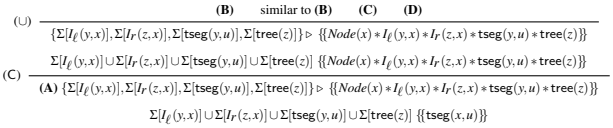
$$\{\{Node^{left}(x) * I_r(z,x) * \mathsf{tree}_{idle}(z)\}\}$$

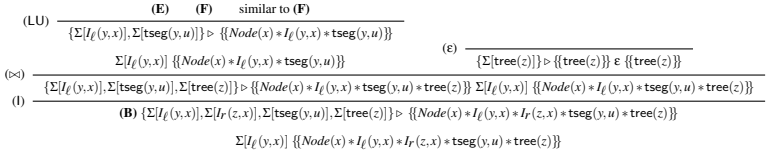


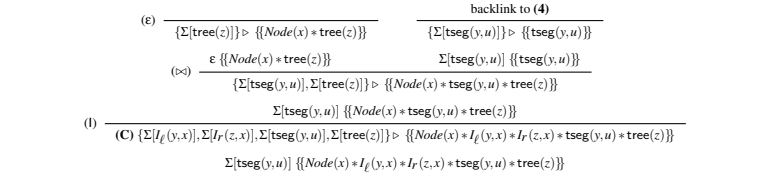
F.4 Havoc Invariance of the Predicate Atom $\mathsf{tseg}(x,u)$

Lastly, we prove the invariance of the predicate $\mathsf{tseg}(x,u)$ via the rules in Fig. 5.









$$\dfrac{\dfrac{\dfrac{(\varepsilon)\ \dfrac{}{\{\Sigma[\mathsf{tseg}(y,u)]\} \triangleright \{\{Node(x) * \mathsf{tseg}(y,u)\}\}\ \varepsilon\ \{\{Node(x) * \mathsf{tseg}(y,u)\}\}} \qquad \dfrac{\text{backlink to } \mathbf{(1)}}{\{\Sigma[\mathsf{tree}(z)]\} \triangleright \{\{\mathsf{tree}(z)\}\}\ \Sigma[\mathsf{tree}(z)]\ \{\{\mathsf{tree}(z)\}\}}}{(\bowtie)\ \{\Sigma[\mathsf{tseg}(y,u)], \Sigma[\mathsf{tree}(z)]\} \triangleright \{\{Node(x) * \mathsf{tseg}(y,u) * \mathsf{tree}(z)\}\}\ \Sigma[\mathsf{tree}(z)]\ \{\{Node(x) * \mathsf{tseg}(y,u) * \mathsf{tree}(z)\}\}}}{(\mathsf{I})\ \mathbf{(D)}\ \{\Sigma[I_\ell(y,x)], \Sigma[I_r(z,x)], \Sigma[\mathsf{tseg}(y,u)], \Sigma[\mathsf{tree}(z)]\} \triangleright \{\{Node(x) * I_\ell(y,x) * I_r(z,x) * \mathsf{tseg}(y,u) * \mathsf{tree}(z)\}\}}$$

$$\Sigma[\mathsf{tree}(z)]\ \{\{Node(x) * I_\ell(y,x) * I_r(z,x) * \mathsf{tseg}(y,u) * \mathsf{tree}(z)\}\}$$

$$\dfrac{\dfrac{(\Sigma)\ \dfrac{}{\{\Sigma[I_\ell(y,x)]\} \triangleright \{\{Node^{idle}(x) * I_\ell(y,x) * Node^{p}(y)\}\}}\ \text{for } (p = right,\ p' = idle) \text{ or } (p = leaf\_busy,\ p' = leaf\_idle)}{(\mathsf{C})\ \dfrac{\Sigma[I_\ell(y,x)]\ \{\{Node^{left}(x) * I_\ell(y,x) * Node^{p'}(y)\}\}}{\{\Sigma[I_\ell(y,x)]\} \triangleright \{\{Node^{idle}(x) * I_\ell(y,x) * Node^{p}(y)\}\}}} \qquad (\dagger)\ \dfrac{}{\{\Sigma[I_\ell(y,x)]\} \triangleright \{\{Node^{q}(x) * I_\ell(y,x) * Node^{p}(y)\}\}}}{(\vee)\ \dfrac{\Sigma[I_\ell(y,x)]\ \{\{Node(x) * I_\ell(y,x) * \mathsf{tseg}(y,u)\}\} \qquad (\mathsf{C})\ \dfrac{\Sigma[I_\ell(y,x)]\ \{\{\mathsf{false}\}\}}{\{\Sigma[I_\ell(y,x)]\} \triangleright \{\{Node(x) * I_\ell(y,x) * Node(y)\}\}}\ \text{for } (q,p) \neq (idle, right),\ (idle, leaf\_busy)}{\mathbf{(E)}\ \{\Sigma[I_\ell(y,x)]\} \triangleright \{\{Node(x) * I_\ell(y,x) * Node(y)\}\}}}$$

$$\Sigma[I_\ell(y,x)]\ \{\{Node(x) * I_\ell(y,x) * \mathsf{tseg}(y,u)\}\}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\text{backlink to } \mathbf{(E)}}{\{\Sigma[I_\ell(y,x)]\} \triangleright \{\{Node(x) * I_\ell(y,x) * Node(y)\}\}} \qquad (\varepsilon)\ \dfrac{}{\{\Sigma[\mathsf{tseg}(v,u)], \Sigma[\mathsf{tree}(w)]\} \triangleright \{\{\mathsf{tseg}(v,u) * \mathsf{tree}(w)\}\}}}{(\bowtie)\ \dfrac{\Sigma[I_\ell(y,x)]\ \{\{Node(x) * I_\ell(y,x) * Node(y)\}\} \qquad \varepsilon\ \{\{\mathsf{tseg}(v,u) * \mathsf{tree}(w)\}\}}{\{\Sigma[I_\ell(y,x)], \Sigma[\mathsf{tseg}(v,u)], \Sigma[\mathsf{tree}(w)]\} \triangleright \{\{Node(x) * I_\ell(y,x) * Node(y) * \mathsf{tseg}(v,u) * \mathsf{tree}(w)\}\}}}{(\mathsf{I})\ \dfrac{\Sigma[I_\ell(y,x)]\ \{\{Node(x) * I_\ell(y,x) * Node(y) * \mathsf{tseg}(v,u) * \mathsf{tree}(w)\}\}}{\{\Sigma[I_\ell(y,x)], \Sigma[I_\ell(v,y)], \Sigma[I_r(w,y)], \Sigma[\mathsf{tseg}(v,u)], \Sigma[\mathsf{tree}(w)]\} \triangleright \{\{Node(x) * I_\ell(y,x) * Node(y) * I_\ell(v,y) * I_r(w,y) * \mathsf{tseg}(v,u) * \mathsf{tree}(w)\}\}}}{(\mathsf{C})\ \dfrac{\Sigma[I_\ell(y,x)]\ \{\{Node(x) * I_\ell(y,x) * Node(y) * I_\ell(v,y) * I_r(w,y) * \mathsf{tseg}(v,u) * \mathsf{tree}(w)\}\}}{\mathbf{(F)}\ \{\Sigma[I_\ell(y,x)], \Sigma[I_\ell(v,y)], \Sigma[I_r(w,y)], \Sigma[\mathsf{tseg}(v,u)], \Sigma[\mathsf{tree}(w)]\} \triangleright \{\{Node(x) * I_\ell(y,x) * Node(y) * I_\ell(v,y) * I_r(w,y) * \mathsf{tseg}(v,u) * \mathsf{tree}(w)\}\}}}$$

$$\Sigma[I_\ell(y,x)]\ \{\{Node(x) * I_\ell(y,x) * \mathsf{tseg}(y,u)\}\}$$