Observation Predicates in Flow Logic

Flemming Nielson, Hanne Riis Nielson and Hongyan Sun

Informatics and Mathematical Modelling
Technical University of Denmark
Building 321, 2800 Kgs. Lyngby, Denmark
E-mail: {nielson, riis, sun}@imm.dtu.dk

Abstract. Motivated by the connection between strong and soft type systems we explore flow analyses with hard constraints on the admissible solutions. We show how to use observation predicates and formula rearrangements to map flow analyses with hard constraints into more traditional flow analyses in such a way that the hard constraints are satisfied exactly when the observation predicates report no violations. The development is carried out in a large fragment of a first order logic with negation and also takes care of the transformations necessary in order to adhere to the stratification restrictions inherent in Alternation-free Least Fixed Point Logic and similar formalisms such as Datalog.

1 Introduction

In the world of type systems one frequently distinguishes between soft typing, where all syntactically correct programs can be typed (possibly with an uninformative top type), and strong typing, where only some of the syntactically correct programs are well typed. Only strong typing offers the traditional strong point of type systems, that "well typed programs cannot go wrong", and hence only strong typing can be seen as a program development methodology. Soft typing on the other hand has much more the flavour of data flow and control flow analyses in accepting all (or a much larger class of) programs and merely reporting on their behaviour, irrespective of whether the behaviour is considered desirable or not.

Given the usefulness of program development methodologies for enhancing the quality of software, it is natural to consider the possibilities for extending flow analyses so that they not only deal with the "soft" properties but also the "hard" constraints. This is not unproblematic because a number of key theorems, in particular the Moore Family result [1, 2] saying that there always exists a least and acceptable solution (somewhat in the manner of principal typing), no longer need to hold. We study the problems in the context of Alternation-free Least Fixed Point Logic (ALFP), a fragment of first order logic slightly more general than Datalog [3], which has proved its usefulness for expressing flow analyses for a variety of programming languages and process algebras, and that may be implemented in systems such as our own Succinct Solver [4] or XSB Prolog [5] with tabled resolution.

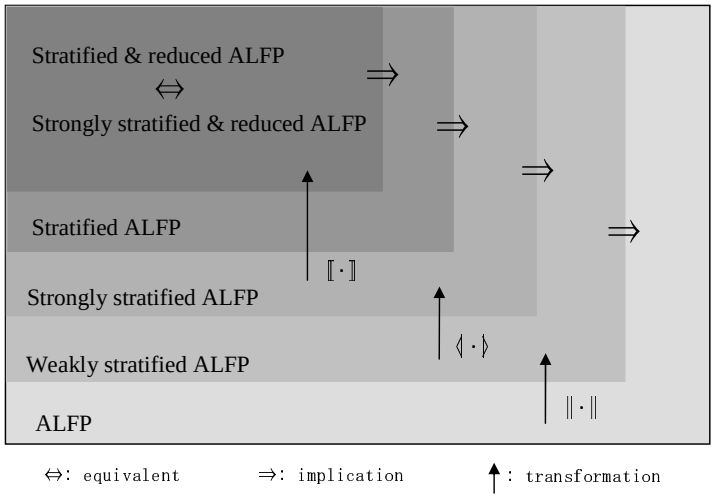


Fig. 1. The relations among various fragments of ALFP; the Succinct Solver accepts *stratified* ALFP whereas XSB Prolog accepts a slightly smaller set of clauses.

The development of the paper is summarised in Figure 1. We start from the most liberal ALFP clauses, and progress upward towards the generation of the stratified ALFP clauses that are accepted by a system such as the Succinct Solver or (with minor limitations) XSB Prolog. We explain the basic notions of ALFP and stratified ALFP in Section 2 and summarise the Moore Family result. We then introduce in Section 3 a useful notion of weak stratification and develop a transformation from ALFP clauses to weakly stratified clauses; the view is that violations of weak stratification correspond to "hard" constraints that must be enforced. We do so by translating each hard constraint into the update of an observation predicate in such a way that the hard constraints are fulfilled exactly when the observation predicates report no violations. We proceed one step further in Section 4 by introducing the concept of strong stratification and show how to transform weakly stratified formulae into strongly stratified ones. Finally in Section 5 we show how the stratification is captured by the strong stratification if we reduce clauses (meaning that certain tautologies have been evaluated) and we show also that reducing a clause preserves strong stratification.

The import of this development is to provide a systematic methodology for dealing with "hard" constraints in the form of Flow Logic [6] specifications. From the point of view of the user this means that analyses can be expressed in full ALFP at the expense of the automatic introduction of observation predicates where necessary in order to enforce stratification. To show the usefulness of this approach we use a small flow analysis for the λ-calculus as a running example. We give also in Section 6 a larger and more detailed example in the context of Discretionary Ambients [7] where we show in more detail how the observation

predicates can be automatically introduced to report violations of the mandatory access control policies [8] considered. The transformations reported here have been implemented as a front-end to our Succinct Solver and are available on the web page [9].

2 ALFP

ALFP clauses basically extend Horn clauses by allowing: both existential and universal quantifications in preconditions; negative queries (subject to the notion of stratification); disjunctions of preconditions; and conjunctions of conclusions.

2.1 Syntax

The set of ALFP clauses, $ALFP[\mathcal{X}, \mathcal{R}]$, is defined relative to a countably infinite set $\mathcal{X}$ of variables and a non-empty and finite set $\mathcal{R}$ of predicate symbols. A typical clause, cl, is generated by the following grammar

$$\begin{array}{lcl} pre & ::= & R\,(x_1, \cdots, x_k) \quad | \quad \neg R\,(x_1, \cdots, x_k) \quad | \quad pre_1 \wedge pre_2 \\ & | & pre_1 \vee pre_2 \quad | \quad \exists x : pre \quad | \quad \forall x : pre \\ cl & ::= & R\,(x_1, \cdots, x_k) \quad | \quad \mathbf{1} \quad | \quad cl_1 \wedge\ cl_2 \\ & | & pre \Rightarrow cl \quad | \quad \forall x : cl \end{array}$$

where $R \in \mathcal{R}$ is a k-ary predicate symbol for $k \geq 1$, $x, x_1, \cdots \in \mathcal{X}$ denote arbitrary variables, and $\mathbf{1}$ is the always true clause. Occurrences of $R(\cdots)$ and $\neg R(\cdots)$ in preconditions are also called *queries* and *negative queries*, respectively, whereas the other occurrences are called *assertions* of the predicate R.

2.2 Semantics

Given a non-empty and countable universe $\mathcal{U}$ of atomic values (also known as atoms) together with interpretations ϱ and σ for predicate symbols and free variables, respectively, the satisfaction relations

$$(\varrho, \sigma) \models pre \quad \text{and} \quad (\varrho, \sigma) \models cl$$

for preconditions and clauses are defined in Table 1. Here $\varrho(R)$ stands for the set of k-tuples $(a_1, \cdots, a_k)$ from $\mathcal{U}^k$ associated with the k-ary predicate R, $\sigma(x)$ stands for the atom of $\mathcal{U}$ bound to x, and $\sigma[x \mapsto a]$ stands for the mapping that is as σ except that x is mapped to a. The free variables occurring in a formula are considered as constant symbols or atoms from the universe $\mathcal{U}$. Thus, given an interpretation σ_0 of the constant symbols, in the clause cl, an interpretation ϱ of the predicate symbols $\mathcal{R}$ is called a *solution* to the clause provided $(\varrho, \sigma_0) \models cl$.

Example 1. ALFP formulae are useful and convenient in specifying flow analyses with hard constraints in the form of Flow Logic. To give a feeling of that, we

$(\varrho,\sigma) \models R\,(x_1,\cdots,x_k)$	iff	$(\sigma(x_1),\cdots,\sigma(x_k)) \in \varrho(R)$
$(\varrho,\sigma) \models \neg R\,(x_1,\cdots,x_k)$	iff	$(\sigma(x_1),\cdots,\sigma(x_k)) \notin \varrho(R)$
$(\varrho,\sigma) \models pre_1 \wedge pre_2$	iff	$(\varrho,\sigma) \models pre_1$ and $(\varrho,\sigma) \models pre_2$
$(\varrho,\sigma) \models pre_1 \vee pre_2$	iff	$(\varrho,\sigma) \models pre_1$ or $(\varrho,\sigma) \models pre_2$
$(\varrho,\sigma) \models \exists x : pre$	iff	$(\varrho,\sigma[x \mapsto a]) \models pre$ for some $a \in \mathcal{U}$
$(\varrho,\sigma) \models \forall x : pre$	iff	$(\varrho,\sigma[x \mapsto a]) \models pre$ for all $a \in \mathcal{U}$
$(\varrho,\sigma) \models R\,(x_1,\cdots,x_k)$	iff	$(\sigma(x_1),\cdots,\sigma(x_k)) \in \varrho(R)$
$(\varrho,\sigma) \models \mathbf{1}$	iff	true
$(\varrho,\sigma) \models cl_1 \wedge cl_2$	iff	$(\varrho,\sigma) \models cl_1$ and $(\varrho,\sigma) \models cl_2$
$(\varrho,\sigma) \models pre \Rightarrow cl$	iff	$(\varrho,\sigma) \models cl$ whenever $(\varrho,\sigma) \models pre$
$(\varrho,\sigma) \models \forall x : cl$	iff	$(\varrho,\sigma[x \mapsto a]) \models cl$ for all $a \in \mathcal{U}$

Table 1. Semantics of preconditions and clauses

shall consider the λ-calculus extended with labels [6], in which an λ-expression $e \in \mathsf{Exp}$ is given by

$$e^\ell ::= c^\ell \mid x^\ell \mid (\lambda x_0.e_0^{\ell_0})^\ell \mid (e_1^{\ell_1} e_2^{\ell_2})^\ell$$

where $c \in \mathsf{Const}$ denotes a constant and $x \in \mathsf{Var}$ denotes a variable. The labels $\ell \in \mathsf{Lab}$ allow us to explicitly refer to specific program points.

We consider a flow analysis that statically predicts which values an expression may evaluate to. Let Val be the set of values, we then define two binary predicates $\mathcal{C}$ and $\mathcal{E}$ such that $\mathcal{C} \subseteq \mathsf{Lab} \times \mathsf{Val}$ and $\mathcal{E} \subseteq \mathsf{Var} \times \mathsf{Val}$ where $\mathcal{C}$ plays a role as a cache to record the analysis estimates of all the subexpressions, hence $\mathcal{C}(\ell, v)$ says that the expression occurring at label ℓ may evaluate to the value v; while $\mathcal{E}$ plays a role as a global environment, and $\mathcal{E}(x, v)$ says that variable x may be bound to the value v.

The analysis is specified with a judgment of the form $(\mathcal{C}, \mathcal{E}) \models e^\ell$ expressing that $\mathcal{C}$ is an acceptable analysis estimate for the expression e referred by label ℓ under the environment $\mathcal{E}$. The analysis is defined by the following ALFP clauses:

$$\begin{array}{l}
\forall x : \forall \ell : \mathsf{ABS}(\mathsf{abst}(x, \ell_0)) \wedge \\
\forall \ell : \forall c : \mathsf{PRG}(\ell, \mathsf{const}(c)) \Rightarrow \mathcal{C}(\ell, \diamond) \wedge \\
\forall \ell : \forall x : \mathsf{PRG}(\ell, \mathsf{var}(x)) \Rightarrow \forall v : \mathcal{E}(x, v) \Rightarrow \mathcal{C}(\ell, v) \wedge \\
\forall \ell : \forall x : \forall \ell_0 : \mathsf{PRG}(\ell, \mathsf{lambda}(x, \ell_0)) \Rightarrow \mathcal{C}(\ell, \mathsf{abst}(x, \ell_0)) \wedge \\
\forall \ell : \forall \ell_1 : \forall \ell_2 : \mathsf{PRG}(\ell, \mathsf{apply}(\ell_1, \ell_2)) \Rightarrow \left[\begin{array}{l}
\forall v : \mathcal{C}(\ell_1, v) \Rightarrow \mathsf{ABS}(v) \wedge \\
\forall x : \forall \ell_0 : \mathcal{C}(\ell_1, \mathsf{abst}(x, \ell_0)) \Rightarrow \\
\quad \left[\begin{array}{l} \forall v : \mathcal{C}(\ell_2, v) \Rightarrow \mathcal{E}(x, v) \wedge \\ \forall v : \mathcal{C}(\ell_0, v) \Rightarrow \mathcal{C}(\ell, v) \end{array}\right]
\end{array}\right]
\end{array}$$

where the binary predicate PRG describes the syntax of the program being analysed. To enhance the readability, we use structured terms[1] by introducing function symbols const, var, lambda and apply that correspond respectively to the constant c, the variable x, the λ-abstraction and the λ-application of the syntax, while the function symbol abst stands for the *abstract value* of the λ-abstraction. These structured terms can be removed by introducing the corresponding predicates and variables (c.f. [4]). Note that $\diamond$ stands for the values of all the constants.

The first clause defines a priori fixed relation ABS which is used to impose the "hard" constraints on the analysis for the λ-application that the first expression (i.e. the operator) must evaluate to an *abstraction*. The last four clauses correspond respectively to the four kinds of λ-expressions. □

2.3 Stratified ALFP

In order to ensure desirable theoretical and pragmatic properties in the presence of negation, the ALFP formulae [4] introduce a notion of stratification similar to the one which is known from *Datalog* [11, 12]. To express this we make use of a mapping $\rho : \mathcal{R} \to \mathbb{N}$ that maps predicate symbols to ranks in $\mathbb{N} = \{0, 1, \cdots\}$.

Definition 1. *A clause cl is stratified (w.r.t. ρ, N) if it has the form $cl = cl_0 \wedge \cdots \wedge cl_k$, and the mapping $\rho : \mathcal{R} \to \{0, 1, \cdots, N\}$ satisfies the following properties for all $i = 0, \cdots, k$ and some $j_i \in \{0, 1, \cdots, N\}$ such that $j_0 < \cdots < j_k$:*

1. *$\rho(R) = j_i$ for every predicate R of assertions in cl_i;*
2. *$\rho(R) \leq j_i$ for every predicate R of queries in cl_i; and*
3. *$\rho(R) < j_i$ for every predicate R of negative queries in cl_i.*

We say that cl_i is in stratum j_i whenever properties (1) to (3) are satisfied.

We shall view stratum 0 as corresponding to priori fixed relations, stratum 1 as corresponding to the representation of the syntax, and stratum N as observation predicates (if any). Our definition is slightly more liberal than the one used in the Succinct Solver [4] where it is required that $j_i = i$; both notions of stratification suffice for the purposes of the Succinct Solver[2].

Example 2. Following the above principle, if we take $\rho(\mathsf{ABS}) = 0$, $\rho(\mathsf{PRG}) = 1$, and $\rho(\mathcal{C}) = \rho(\mathcal{E}) = 2$ in the previous example, then the last clause of the analysis in the example is *not* stratified. □

[1] Structured terms are supported in Succinct Solver V2.0 [10]. In the theoretical development of this paper, we use V1.0 [4] to avoid the complications arising from a structured universe $\mathcal{U}$.

[2] Indeed a clause is stratified (w.r.t. some ρ, N) with respect to one definition if and only if it is stratified (w.r.t. some ρ', N') with respect to the other (and such that $\rho(R) \leq \rho(R')$ is equivalent to $\rho'(R) \leq \rho'(R')$).

The least solution exists. Let Δ be the set of interpretations ϱ of predicate symbols in $\mathcal{R}$ over $\mathcal{U}$, then $\Delta = (\Delta, \sqsubseteq)$ forms a complete lattice, where the lexicographical ordering $\sqsubseteq$ is defined by $\varrho_1 \sqsubseteq \varrho_2$ if and only if there is some $0 \leq j \leq N$ such that the following properties hold:

- $\varrho_1(R) = \varrho_2(R)$ for all $R \in \mathcal{R}$ with $\rho(R) < j$
- $\varrho_1(R) \subseteq \varrho_2(R)$ for all $R \in \mathcal{R}$ with $\rho(R) = j$
- either $j = N$ or $\varrho_1(R) \subset \varrho_2(R)$ for at least one $R \in \mathcal{R}$ with $\rho(R) = j$

Proposition 1. *(Nielson, Seidl and Riis Nielson [4]) Assume cl is a stratified ALFP formula and σ_0 is an interpretation of the free variables in cl. Then the set $\Delta' = \{\varrho \in \Delta \mid (\varrho, \sigma_0) \models cl\}$ forms a Moore family, i.e. it is closed under greatest lower bounds.*

The proof can be found in [4].

3 Weak Stratification

Throughout this paper we study how to "live with" the restrictions imposed by stratification. We begin by introducing the concept of *weak* stratification by means of an inference system and we show how to transform general ALFP clauses to the corresponding weakly stratified clauses by introducing so called *observation predicates*.

3.1 Weakly Stratified Clauses

Definition 2. *A clause cl is weakly stratified (w.r.t. ρ, N)* iff $\exists \mathsf{s} \subseteq \{0, 1, \cdots, N\}$ *such that $\vdash_\rho cl : \mathsf{s}$ and $\rho(R) \in \{0, 1, \cdots, N\}$ for all $R \in \mathcal{R}$. The judgment $\vdash_\rho$ is defined by the rules in Table 2 using the judgment $\vdash^{\cdot}_\rho$ defined in Table 3.*

$[w1]$	$\vdash_\rho \mathbf{1} : \emptyset$	
$[w2]$	$\vdash_\rho R(\vec{x}) : \mathsf{s}$	$\texttt{if } \mathsf{s} = \{\rho(R)\}$
$[w3]$	$\dfrac{\vdash_\rho cl : \mathsf{s}}{\vdash_\rho \forall x : cl : \mathsf{s}}$	
$[w4]$	$\dfrac{\vdash_\rho cl_1 : \mathsf{s}_1 \quad \vdash_\rho cl_2 : \mathsf{s}_2}{\vdash_\rho cl_1 \wedge cl_2 : \mathsf{s}_1 \cup \mathsf{s}_2}$	
$[w5]$	$\dfrac{\vdash^{\cdot}_\rho pre : n \quad \vdash_\rho cl : \mathsf{s}}{\vdash_\rho pre \Rightarrow cl : \mathsf{s}}$	$\texttt{if } n \leq min(\mathsf{s})$

Table 2. Rules for weakly stratified clauses

Intuitively, $\vdash_\rho cl : \mathsf{s}$ says that s is the set of ranks $\rho(R)$ such that $R \in \mathcal{R}$ occurs as an assertion in cl, and that cl may be rearranged so as to be stratified in the sense of Definition 1. The stratification condition is enforced by the rule $[w5]$, where $min(\mathsf{s})$ denotes the minimal rank of an assertion occurring in s, taking $min(\emptyset) = N$. The rules for preconditions are given in Table 3, where $\vdash'_\rho pre : n$ says that n is the maximal rank of a query occurring in precondition pre, and we add 1 for each negative query. The following fact says that the set

$[p1]$	$\vdash'_\rho R(\vec{x}) : n$	$n = \rho(R)$
$[p2]$	$\vdash'_\rho \neg R(\vec{x}) : n$	$n = \rho(R) + 1$
$[p3]$	$\dfrac{\vdash'_\rho pre : n}{\vdash'_\rho \exists x : pre : n}$	
$[p4]$	$\dfrac{\vdash'_\rho pre : n}{\vdash'_\rho \forall x : pre : n}$	
$[p5]$	$\dfrac{\vdash'_\rho pre_1 : n_1 \quad \vdash'_\rho pre_2 : n_2}{\vdash'_\rho pre_1 \wedge pre_2 : n}$	$n = max(n_1, n_2)$
$[p6]$	$\dfrac{\vdash'_\rho pre_1 : n_1 \quad \vdash'_\rho pre_2 : n_2}{\vdash'_\rho pre_1 \vee pre_2 : n}$	$n = max(n_1, n_2)$

Table 3. Rules for preconditions

s is uniquely determined by ρ in $\vdash_\rho cl : \mathsf{s}$.

Fact 1 *If* $\vdash_\rho cl : \mathsf{s}_1$ *and* $\vdash_\rho cl : \mathsf{s}_2$ *then* $\mathsf{s}_1 = \mathsf{s}_2$*; similarly if* $\vdash'_\rho pre : n_1$ *and* $\vdash'_\rho pre : n_2$ *then* $n_1 = n_2$*.*

Remark 1. A clause cl in $ALFP[\mathcal{X}, \mathcal{R}]$ is by definition weakly stratified iff there exist ρ, N and $\mathsf{s} \subseteq \{0, 1, \cdots, N\}$ such that $\vdash_\rho cl : \mathsf{s}$ (and $\rho(R) \in \{0, 1, \cdots, N\}$ for all $R \in \mathcal{R}$). This condition is easily checked by the following procedure.

Construct a labelled graph where nodes are all the relations in $\mathcal{R}$. For each occurrence of relations R and S such that R occurs to the left of some $\Rightarrow$ and S to the right, construct an edge $R \xrightarrow{\varepsilon} S$. For each occurrence of relation $\neg R$ and S such that $\neg R$ occurs to the left of some $\Rightarrow$ and S to the right, construct an edge $R \xrightarrow{\neg} S$.

We state without proof that the clause cl is weakly stratified iff there exists no loop involving any $\xrightarrow{\neg}$ edges. □

3.2 Observation Predicates

To transform an ALFP clause to an equivalent weakly stratified clause, the basic idea is to transform a non-stratified clause[3], e.g. $R \Rightarrow S$ with $\rho(R) > \rho(S)$ into

[3] We say a clause is non-stratified if it is not weakly stratified.

a weakly stratified clause $R \Rightarrow \neg S \Rightarrow E_S$ by introducing a so called *observation predicate* E_S such that $\rho(E_S) > \rho(R)$.

Definition 3. *We define in Table 4 the function* $\|.\|^{\mathsf{s}}_{\rho}$ *mapping a clause* $cl \in ALFP[\mathcal{X}, \mathcal{R}_0]$ *into the clause* $\|cl\|^{\mathsf{s}}_{\rho} \in ALFP[\mathcal{X}, \mathcal{R}_0 \cup \mathcal{R}_E]$. *Here* $\mathsf{s} \subseteq \{0, 1, \cdots, N\}$ *and* $\rho(R) \in \{0, 1, \cdots, N\}$ *for all* $R \in \mathcal{R}_0$. *The function introduces new predicates* $E_R \in \mathcal{R}_E$, *which are called observation predicates. We assume that* $\mathcal{R}_E = \{E_R \mid R \in \mathcal{R}_0\}$ *is disjoint from* $\mathcal{R}_0$, *and* $\rho(E_R) = N + 1$ *(or more precisely,* $\rho(R) = N + 1 \Leftrightarrow R \in \mathcal{R}_E$*).*

$$
\begin{array}{lll}
\|\mathbf{1}\|^{\mathsf{s}}_{\rho} & = \mathbf{1} & \\
\|R(\vec{x})\|^{\mathsf{s}}_{\rho} & = \begin{cases} R(\vec{x}) & \text{if } \mathsf{s} \supseteq \{\rho(R)\} \\ \neg R(\vec{x}) \Rightarrow E_R(\vec{x}) & \text{otherwise} \end{cases} & \\
\|\forall x : cl\|^{\mathsf{s}}_{\rho} & = \forall x : \|cl\|^{\mathsf{s}}_{\rho} & \\
\|cl_1 \wedge cl_2\|^{\mathsf{s}}_{\rho} & = \|cl_1\|^{\mathsf{s}}_{\rho} \wedge \|cl_2\|^{\mathsf{s}}_{\rho} & \\
\|pre \Rightarrow cl\|^{\mathsf{s}}_{\rho} & = pre \Rightarrow \|cl\|^{\delta(\mathsf{s},n)}_{\rho} & \text{if } \vdash^{:}_{\rho} \text{ pre: n} \\
\text{where } \delta(\mathsf{s}, n) & = \{a \in \mathsf{s} \mid a \geq n\} &
\end{array}
$$

Table 4. The transformation function $\|.\|^{\mathsf{s}}_{\rho}$

Intuitively, s contains the ranks of assertions that are permissible at the stage where the function is called. In the second transformation rule, when $\rho(R)$ is not in s (which implies that R is not allowed to occur as an assertion), then a new predicate E_R of rank $N + 1$ is introduced.

The auxiliary function δ is defined by $\delta(\mathsf{s}, n) = \{a \in \mathsf{s} \mid a \geq n\}$. It filters the set s by n to get rid of the ranks which are less than n, thus the predicates having lower ranks than n and occurring as assertions (which violates rule $[w5]$) will be eventually transformed as negative queries (by the second transformation rule in Table 4) so that the violation of rule $[w5]$ is resolved.

Example 3. The last clause of the analysis in *Example 1* can be transformed into a weakly stratified clause by introducing the observation predicate $\mathsf{E_{ABS}}$ (taking $\rho(\mathsf{E_{ABS}}) = 3$) as follows:

$$
\forall \ell : \forall \ell_1 : \forall \ell_2 : \mathsf{PRG}(\ell, \mathsf{apply}(\ell_1, \ell_2)) \Rightarrow \begin{bmatrix} \forall v : \mathcal{C}(\ell_1, v) \Rightarrow \neg \mathsf{ABS}(v) \Rightarrow \mathsf{E_{ABS}}(v) \wedge \\ \forall x : \forall \ell_0 : \mathcal{C}(\ell_1, \mathsf{abst}(x, \ell_0)) \Rightarrow \\ \begin{bmatrix} \forall v : \mathcal{C}(\ell_2, v) \Rightarrow \mathcal{E}(x, v) \wedge \\ \forall v : \mathcal{C}(\ell_0, v) \Rightarrow \mathcal{C}(\ell, v) \end{bmatrix} \end{bmatrix}
$$

Note that this clause is not stratified in the sense of Definition 1. □

The following proposition says that indeed the function $\|\bullet\|^{\mathsf{s}}_{\rho}$ transforms a clause cl into a weakly stratified clause $\|cl\|^{\mathsf{s}}_{\rho}$ satisfying $\vdash_{\rho} \|cl\|^{\mathsf{s}}_{\rho} : \mathsf{s}'$ for some s' such that $\mathsf{s}' \subseteq \mathsf{s} \cup \{N+1\}$.

Proposition 2. *Given a clause cl together with $\rho : \mathcal{R} \to \{0, 1, \cdots, N\}$ and $\mathsf{s} \subseteq \{0, 1, \cdots, N\}$, then $\exists \mathsf{s}' \subseteq \mathsf{s} \cup \{N+1\}$ such that $\vdash_{\rho} \|cl\|^{\mathsf{s}}_{\rho} : \mathsf{s}'$. If $\exists \mathsf{s}' \subseteq \mathsf{s} : \vdash_{\rho} cl : \mathsf{s}'$ then $\|cl\|^{\mathsf{s}}_{\rho} = cl$.*

A proof by structural induction on cl is given in Appendix A.

Let ϱ_0 stand for an interpretation for all predicate symbols $R \in \mathcal{R}_0$, and ϱ_E stand for an interpretation for all observation predicate symbols $E_R \in \mathcal{R}_E$. We write $\varrho_0 \sqcup \varrho_E$ for the map ϱ defined by

$$\varrho(R) = \begin{cases} \varrho_0(R) & \text{if } R \in \mathcal{R}_0 \\ \varrho_E(R) & \text{if } R \in \mathcal{R}_E \end{cases}$$

and we write $\varrho_E = \bot$ to mean $\forall R : \varrho_E(R) = \emptyset$.

The following proposition says that a solution to clause cl is also a solution to $\|cl\|^{\mathsf{s}}_{\rho}$ provided that the solution maps observation predicates to the empty set.

Proposition 3. $(\varrho_0, \sigma) \models cl \Leftrightarrow (\varrho_0 \sqcup \varrho_E, \sigma) \models \|cl\|^{\mathsf{s}}_{\rho}$ *if* $\varrho_E = \bot$.

It is sufficient to prove (i) $(\varrho, \sigma) \models cl$ implies $(\varrho, \sigma) \models \|cl\|^{\mathsf{s}}_{\rho}$ if $\varrho_E = \bot$, and (ii) $(\varrho, \sigma) \models \|cl\|^{\mathsf{s}}_{\rho}$ implies $(\varrho, \sigma) \models cl$ if $\varrho_E = \bot$. For (i), a proof by structural induction on cl is given in Appendix A, and for (ii) it is analogous.

Sometimes we are interested in fixing a set $\mathcal{R}_b \subseteq \mathcal{R}_0$ of base predicates. These may depend on each other but not on any other predicates and hence may be used to express hard constraints on the other predicates. We shall say that a ranking $\rho : \mathcal{R} \to \{0, 1, \cdots, N+1\}$ for $\mathcal{R} = \mathcal{R}_0 \cup \mathcal{R}_E$ respects $\mathcal{R}_b$ whenever there exists a number $N' \in \{0, 1, \cdots, N+1\}$ such that $R \in \mathcal{R}_0 \Leftrightarrow \rho(R) \leq N'$; a typical choice of N' might be 0 as suggested already in Subsection 2.3.

Remark 2. One weakness of the above transformation is that it presupposes that a suitable ρ has been found and this is not always appropriate. In the manner of Remark 1 we therefore suggest the following alternative transformation strategy.

Given a clause cl construct the graph as in Remark 1 and write $\overset{\varepsilon}{\Rightarrow}$ for $\xrightarrow{\varepsilon}{}^*$ and $\overset{\neg}{\Rightarrow}$ for $(\xrightarrow{\varepsilon}{}^* \xrightarrow{\neg} \xrightarrow{\varepsilon}{}^*)^+$. Whenever we have an assertion $R(\vec{x})$ occurring in a context

$$pre \Rightarrow \cdots R(\vec{x}) \cdots$$

such that pre contains some relation S such that $R \overset{\neg}{\Rightarrow} S$ or some relation $\neg S$ such that $R \overset{\varepsilon}{\Rightarrow} S$, we replace it by

$$pre \Rightarrow \cdots (\neg R(\vec{x}) \Rightarrow E_R(\vec{x})) \cdots$$

This transformation is repeated as often as possible. The intention is that observation predicates should not arise for base predicates in $\mathcal{R}_b$. This is possible exactly when the graph constructed in Remark 1, restricted to the nodes in $\mathcal{R}_b$, does not contain any cycles involving any $\xrightarrow{\neg}$ edges. □

Sometimes it may be more informative to enlarge the set of variables included with the observation predicates. In the transformations considered so far we replace

$$\forall \vec{x}, \vec{y} : pre \Rightarrow \cdots R(\vec{x}) \cdots \qquad (*)$$

with

$$\forall \vec{x}, \vec{y} : pre \Rightarrow \cdots (\neg R(\vec{x}) \Rightarrow E_R(\vec{x})) \cdots$$

An interesting alternative is to replace $(*)$ with

$$\forall \vec{x}, \vec{y} : pre \Rightarrow \cdots (\neg R(\vec{x}) \Rightarrow E_R(\vec{x}, \vec{y})) \cdots$$

since this gives more information about the context in which the violation takes place. To be specific the idea is to incorporate with $\vec{x}$ all other names bound at the occurrence of $R(\vec{x})$ and in the clause $(*)$ it would seem that $\vec{y}$ is the appropriate set of variables.

When the transformations are performed manually rather than automatically it may be wise to explore intermediate possibilities among these extremes.

4 Strong Stratification

We now introduce *strong* stratification by placing further demands on the inference system introduced above; as we shall see in Section 5 this brings us very close to stratification in the sense of Definition 1. We then establish a transformation from a weakly stratified clause into an equivalent strongly stratified clause and we give an upper bound on the amount of syntactic expansion that may take place.

4.1 Strongly Stratified Clauses

Definition 4. *A clause cl is strongly stratified (w.r.t. ρ, N)* iff $\exists \mathsf{s} \subseteq \{0, 1, \cdots, N\}$ *such that* $\Vdash_\rho cl : \mathsf{s}$ *and* $\rho(R) \in \{0, 1, \cdots, N\}$ *for all* $R \in \mathcal{R}$. *The judgment* $\Vdash_\rho$ *is defined by the rules in Table 5 using the judgment* $\vdash^{\cdot}_\rho$ *defined in Table 3.*

Intuitively, $\Vdash_\rho cl : \mathsf{s}$ says that s is the set of ranks $\rho(R)$ such that $R \in \mathcal{R}$ occurs as an assertion in cl, and that cl is stratified. In the definition of $\Vdash_\rho$ in Table 5 we use $\mid \mathsf{s} \mid$ to denote the number of elements (i.e. ranks of assertions) in s. Again, $min(\mathsf{s})$ denotes the minimal rank of an assertion occurring in s, taking $min(\emptyset) = N$, while $max(\mathsf{s})$ denotes the maximal rank of an assertion occurring in s, taking $max(\emptyset) = 0$. The next fact states that the set s is uniquely determined by ρ in $\Vdash_\rho cl : \mathsf{s}$.

Fact 2 *If* $\Vdash_\rho cl : \mathsf{s}_1$ *and* $\Vdash_\rho cl : \mathsf{s}_2$ *then* $\mathsf{s}_1 = \mathsf{s}_2$*; similarly if* $\vdash^{\cdot}_\rho pre : n_1$ *and* $\vdash^{\cdot}_\rho pre : n_2$ *then* $n_1 = n_2$.

Proposition 4. *If cl is strongly stratified then cl is weakly stratified (w.r.t. the same ρ, N).*

$[s1]$	$\Vdash_\rho \mathbf{1} : \emptyset$	
$[s2]$	$\Vdash_\rho R(\vec{x}) : \mathsf{s}$	$\mathtt{if}\ \mathsf{s} = \{\rho(R)\}$
$[s3]$	$\dfrac{\Vdash_\rho cl : \mathsf{s}}{\Vdash_\rho \forall x : cl : \mathsf{s}}$	$\mathtt{if}\ \mid \mathsf{s} \mid \leq 1$
$[s4]$	$\dfrac{\Vdash_\rho cl_1 : \mathsf{s}_1 \quad \Vdash_\rho cl_2 : \mathsf{s}_2}{\Vdash_\rho cl_1 \wedge cl_2 : \mathsf{s}_1 \cup \mathsf{s}_2}$	$\mathtt{if}\ max(\mathsf{s}_1) \leq min(\mathsf{s}_2)$
$[s5]$	$\dfrac{\vdash^{\cdot}_\rho pre : n \quad \Vdash_\rho cl : \mathsf{s}}{\Vdash_\rho pre \Rightarrow cl : \mathsf{s}}$	$\mathtt{if}\ \mid \mathsf{s} \mid \leq 1 \wedge n \leq min(\mathsf{s})$

Table 5. Rules for strongly stratified clauses

It is sufficient to prove that $\Vdash_\rho cl : \mathsf{s}$ implies $\vdash_\rho cl : \mathsf{s}$. A proof by structural induction on the derivation tree for $\Vdash_\rho cl : \mathsf{s}$ is given in Appendix B.

The following example shows that there are weakly stratified formulae that are not strongly stratified.

Example 4. The weakly stratified clause obtained from *Example 3* is not strongly stratified, since the conclusion clause for the precondition $\mathsf{PRG}(\ell, \mathsf{apply}(\ell_1, \ell_2))$ is not strongly stratified. In this case the conclusion clause is a conjunction of two clauses with $\mathsf{s}_1 = \{3\}$ and $\mathsf{s}_2 = \{2\}$ since $\rho(\mathsf{E_{ABS}}) = 3$ and $\rho(\mathcal{C}) = \rho(\mathcal{E}) = 2$. Clearly it is not strongly stratified because that the side condition of $[s4]$ is not satisfied. Therefore the overall clause is not strongly stratified according to $[s3]$ and $[s5]$ respectively. □

4.2 From Weak Stratification to Strong Stratification

Definition 5. *We define in Table 6 the function* $\langle\!|\mathbf{.}|\!\rangle^{\mathsf{s}}_\rho$ *mapping a clause* $cl \in ALFP[\mathcal{X}, \mathcal{R}]$ *into the clause* $\langle\!|cl|\!\rangle^{\mathsf{s}}_\rho \in ALFP[\mathcal{X}, \mathcal{R}]$ *where we demand that* $\mathsf{s} \subseteq \{0, 1, \cdots, N\}$ *and* $\rho(R) \in \{0, 1, \cdots, N\}$ *for all* $R \in \mathcal{R}$.

The auxiliary functions h and t are defined as:

$$h(\mathsf{s}) = \begin{cases} \emptyset & \text{if } \mid \mathsf{s} \mid = 0 \\ \{min(\mathsf{s})\} & \text{if } \mid \mathsf{s} \mid \geq 1 \end{cases} \qquad\qquad t(\mathsf{s}) = \mathsf{s} \setminus h(\mathsf{s})$$

When $\mathsf{s} = \emptyset$, $h(\mathsf{s}) = t(\mathsf{s}) = \emptyset$. Otherwise, $h(\mathsf{s})$ contains the minimal element of s, and $t(\mathsf{s})$ contains the remaining elements of s. Hence $h(\mathsf{s}), h(t(\mathsf{s})), h(t(t(\mathsf{s}))), \cdots$ partitions s into singleton sets in ascending order, thus the side condition for $[s4]$ is likely to be satisfied after the transformation $\langle\!|cl|\!\rangle^{\mathsf{s}}_\rho$ has been performed.

In Table 6, the transformation for clause $\mathbf{1}$ in fact considers two cases: $\mathsf{s} = \emptyset$ and $\mathsf{s} \neq \emptyset$. Since in both cases $\langle\!|\mathbf{1}|\!\rangle^{\mathsf{s}}_\rho = \mathbf{1}$, the distinction between the two cases is not highlighted in the definition.

$$
\begin{array}{lll}
\langle\!|\mathbf{1}|\!\rangle^{\mathsf{s}}_{\rho} = \mathbf{1} & & \\
\langle\!|R(\vec{x})|\!\rangle^{\mathsf{s}}_{\rho} & = \begin{cases} R(\vec{x}) & \text{if } \mathsf{s} = \{\rho(R)\} \\ \mathbf{1} & \text{otherwise} \end{cases} & \\
\langle\!|\forall x : cl|\!\rangle^{\mathsf{s}}_{\rho} & = \begin{cases} \forall x : \langle\!|cl|\!\rangle^{\mathsf{s}}_{\rho} & \text{if } |\ \mathsf{s}\ | = 1 \\ \langle\!|\forall x : cl|\!\rangle^{h(\mathsf{s})}_{\rho} \wedge \langle\!|\forall x : cl|\!\rangle^{t(\mathsf{s})}_{\rho} & \text{if } |\ \mathsf{s}\ | > 1 \\ \mathbf{1} & \text{otherwise} \end{cases} & \\
\langle\!|cl_1 \wedge cl_2|\!\rangle^{\mathsf{s}}_{\rho} & = \begin{cases} \langle\!|cl_1|\!\rangle^{\mathsf{s}}_{\rho} \wedge \langle\!|cl_2|\!\rangle^{\mathsf{s}}_{\rho} & \text{if } |\ \mathsf{s}\ | = 1 \\ \langle\!|cl_1 \wedge cl_2|\!\rangle^{h(\mathsf{s})}_{\rho} \wedge \langle\!|cl_1 \wedge cl_2|\!\rangle^{t(\mathsf{s})}_{\rho} & \text{if } |\ \mathsf{s}\ | > 1 \\ \mathbf{1} & \text{otherwise} \end{cases} & \\
\langle\!|pre \Rightarrow cl|\!\rangle^{\mathsf{s}}_{\rho} & = \begin{cases} pre \Rightarrow \langle\!|cl|\!\rangle^{\mathsf{s}}_{\rho} & \text{if } \begin{array}{l} |\ \mathsf{s}\ | = 1 \wedge \\ \exists n \leq min(\mathsf{s}) : \vdash^{\cdot}_{\rho} pre : n \end{array} \\ \langle\!|pre \Rightarrow cl|\!\rangle^{h(\mathsf{s})}_{\rho} \wedge \langle\!|pre \Rightarrow cl|\!\rangle^{t(\mathsf{s})}_{\rho} & \text{if } |\ \mathsf{s}\ | > 1 \\ \mathbf{1} & \text{otherwise} \end{cases} &
\end{array}
$$

Table 6. The transformation function $\langle\!|\bullet|\!\rangle^{\mathsf{s}}_{\rho}$

Example 5. Applying the transformation function defined in Table 6, the weakly stratified clause obtained from *Example 3* can be transformed into the following strongly stratified clause:

$$\forall \ell : \forall \ell_1 : \forall \ell_2 : \mathsf{PRG}(\ell, \mathsf{apply}(\ell_1, \ell_2)) \Rightarrow \left[\begin{array}{l} \mathbf{1} \wedge \\ \forall x : \forall \ell_0 : \mathcal{C}(\ell_1, \mathsf{abst}(x, \ell_0)) \Rightarrow \\ \left[\begin{array}{l} \forall v : \mathcal{C}(\ell_2, v) \Rightarrow \mathcal{E}(x, v) \wedge \\ \forall v : \mathcal{C}(\ell_0, v) \Rightarrow \mathcal{C}(\ell, v) \end{array}\right] \end{array}\right] \wedge$$

$$\forall \ell : \forall \ell_1 : \forall \ell_2 : \mathsf{PRG}(\ell, \mathsf{apply}(\ell_1, \ell_2)) \Rightarrow [\forall v : \mathcal{C}(\ell_1, v) \Rightarrow \neg \mathsf{ABS}(v) \Rightarrow \mathsf{E}_{\mathsf{ABS}}(v)] \wedge \mathbf{1}$$

Clearly this clause and the clause of *Example 3* are logically equivalent. □

We show here some more examples of transformations from weakly stratified clauses into strongly stratified clauses.

Example 6. Given $R, S, T \in \mathcal{R}$ such that $\rho(R) = 1$, $\rho(S) = 2$ and $\rho(T) = 3$, then:

(1) $\langle\!|\forall x : R(x) \wedge T(x)|\!\rangle^{\{1,3\}}_{\rho} = (\forall x : R(x) \wedge \mathbf{1}) \wedge (\forall x : \mathbf{1} \wedge T(x))$
(2) $\langle\!|T \wedge S|\!\rangle^{\{3,2\}}_{\rho} = (\mathbf{1} \wedge S) \wedge (T \wedge \mathbf{1})$
(3) $\langle\!|R \Rightarrow (S \wedge T)|\!\rangle^{\{2,3\}}_{\rho} = (R \Rightarrow S \wedge \mathbf{1}) \wedge (R \Rightarrow \mathbf{1} \wedge T)$ □

The following proposition says that indeed the function $\langle\!|\bullet|\!\rangle^{\mathsf{s}}_{\rho}$ transforms a clause *cl* into a strongly stratified clause $\langle\!|cl|\!\rangle^{\mathsf{s}}_{\rho}$.

Proposition 5. *Given a clause cl then* $\forall \mathsf{s} : \exists \mathsf{s}' \subseteq \mathsf{s}$ *such that* $\Vdash_{\rho} \langle\!|cl|\!\rangle^{\mathsf{s}}_{\rho} : \mathsf{s}'$.

A proof of Proposition 5 by structural induction on cl is given in Appendix B. The proof is facilitated by Lemma 1, which is also proved in Appendix B.

Lemma 1. *Whenever cl is different from* $\mathbf{1}$ *and* $R(\vec{x})$*, and* $\mathsf{s} = \{j_1, \cdots, j_k\}$ *with* $|\ \mathsf{s}\ | > 0$ *and* $j_1 < \cdots < j_k$*, then* $\langle\!|cl|\!\rangle_\rho^{\mathsf{s}} = \langle\!|cl|\!\rangle_\rho^{\{j_1\}} \wedge \cdots \wedge \langle\!|cl|\!\rangle_\rho^{\{j_k\}}$.

Moreover, if the clause cl is weakly stratified, i.e. satisfying $\vdash_\rho cl : \mathsf{s}$, then it is equivalent[4] to the strongly stratified clause $\langle\!|cl|\!\rangle_\rho^{\mathsf{s}}$, as expressed below.

Proposition 6. *If* $\vdash_\rho cl : \mathsf{s}$*, then* $cl \Leftrightarrow \langle\!|cl|\!\rangle_\rho^{\mathsf{s}}$.

A proof by structural induction on cl is given in Appendix B. The proof is facilitated by Lemma 2, which is also proved in Appendix B.

Lemma 2. *If* $\vdash_\rho cl : \emptyset$*, then* $cl \Leftrightarrow \mathbf{1}$.

We state without proof that $cl \Rightarrow \langle\!|cl|\!\rangle_\rho^{\mathsf{s}}$ holds for all choices of s; hence the assumption of $\vdash_\rho cl : \mathsf{s}$ is only needed for the converse implication.

Bounded blow-up. We conclude by stating an upper bound on how much $\langle\!|cl|\!\rangle_\rho^{\mathsf{s}}$ may be syntactically larger than cl:

Proposition 7. *$cl' = \langle\!|cl|\!\rangle_\rho^{\mathsf{s}}$ is bounded in the sense that:*

1. *if* $|\ \mathsf{s}\ | \leq 1$*, then* $|\ cl'\ | \leq |\ cl\ |$*;*
2. *if* $|\ \mathsf{s}\ | > 1$*, then* $|\ cl'\ | \leq |\ \mathsf{s}\ |\ (|\ cl\ | + 1)$*;*

where $|\ cl\ |$ *denotes the size of* cl*. Hence in all cases* $|\ cl'\ | \leq (|\ \mathsf{s}\ | + 1)(|\ cl\ | + 1)$.

The proof of this proposition can be found in Appendix B.

It follows from Propositions 5 and 7 that $\langle\!|.|\!\rangle_\rho^{\mathsf{s}}$ transforms a weakly stratified clause into a strongly stratified clause and that the maximal blow-up is proportional to the number of strata.

5 Stratification Reconsidered

Strong stratification essentially captures stratification in the sense of Definition 1. In this section we consider the fine differences and introduce the notion of *reduced* clauses in order to state the precise relationship.

5.1 Reduced Clauses

Intuitively a reduced clause is one where $\mathbf{1}$ has been removed whenever possible.

Definition 6. *A reduced clause cl is a clause that has no redexes with respect to the* $\mathbf{1}$*-reduction rules defined in Table 7.*

$[r1]$	$\forall x : \mathbf{1} \to \mathbf{1}$	$[r2]$	$pre \Rightarrow \mathbf{1} \to \mathbf{1}$
$[r3]$	$\mathbf{1} \wedge cl \to cl$	$[r4]$	$cl \wedge \mathbf{1} \to cl$
$[r5]$	$\dfrac{cl \to cl'}{\forall x : cl \to \forall x : cl'}$	$[r6]$	$\dfrac{cl \to cl'}{pre \Rightarrow cl \to pre \Rightarrow cl'}$
$[r7]$	$\dfrac{cl_1 \to cl_1'}{cl_1 \wedge cl_2 \to cl_1' \wedge cl_2}$	$[r8]$	$\dfrac{cl_2 \to cl_2'}{cl_1 \wedge cl_2 \to cl_1 \wedge cl_2'}$

Table 7. The **1**-reduction rules

A rule in Table 7 says that whenever a non-reduced clause has the form as that on the left side of $\to$, it will be rewritten to the clause as that on the right side of $\to$. More specifically, $[r1]$ states that the universally quantified clause $\mathbf{1}$ is reduced to $\mathbf{1}$, and $[r2]$ that if $\mathbf{1}$ is the conclusion of an implication clause, then the implication clause is reduced to $\mathbf{1}$, while $[r3]$ and $[r4]$ state that the conjunction of $\mathbf{1}$ and cl is reduced to cl. The other rules deal with components of composite clauses. The next fact is an immediate observation.

Fact 3 *A reduced clause other than* $\mathbf{1}$ *has at least one assertion.*

5.2 The Reduction Algorithm

Definition 7. *We define in Table 8 the function* $[\![\mathbf{.}]\!]$ *mapping a clause* $cl \in ALFP[\mathcal{X}, \mathcal{R}]$ *into the clause* $[\![cl]\!] \in ALFP[\mathcal{X}, \mathcal{R}]$. *An auxiliary function* ϑ *is also defined in Table 8.*

$$
\begin{array}{ll}
[\![\mathbf{1}]\!] & = \mathbf{1} \\
[\![R(\vec{x})]\!] & = R(\vec{x}) \\
[\![\forall x : cl]\!] & = \vartheta(\forall x : [\![cl]\!]) \\
[\![cl_1 \wedge cl_2]\!] & = \vartheta([\![cl_1]\!] \wedge [\![cl_2]\!]) \\
[\![pre \Rightarrow cl]\!] & = \vartheta(pre \Rightarrow [\![cl]\!])
\end{array}
\qquad
\begin{array}{ll}
\vartheta(\forall x : cl) & = \begin{cases} \mathbf{1} & \text{if } cl = \mathbf{1} \\ \forall x : cl & \text{otherwise} \end{cases} \\
\vartheta(pre \Rightarrow cl) & = \begin{cases} \mathbf{1} & \text{if } cl = \mathbf{1} \\ pre \Rightarrow cl & \text{otherwise} \end{cases} \\
\vartheta(cl_1 \wedge cl_2) & = \begin{cases} cl_1 & \text{if } cl_2 = \mathbf{1} \\ cl_2 & \text{if } cl_1 = \mathbf{1} \\ cl_1 \wedge cl_2 & \text{otherwise} \end{cases}
\end{array}
$$

Table 8. The transformation function $[\![\mathbf{.}]\!]$

[4] In the paper, whenever we say $cl_1 \Leftrightarrow cl_2$, we mean $(\varrho, \sigma) \models cl_1$ iff $(\varrho, \sigma) \models cl_2$.

Example 7. The clause obtained from *Example 5* can be reduced to give

$$\forall \ell : \forall \ell_1 : \forall \ell_2 : \mathsf{PRG}(\ell, \mathsf{apply}(\ell_1, \ell_2)) \Rightarrow \begin{bmatrix} \forall x : \forall \ell_0 : \mathcal{C}(\ell_1, \mathsf{abst}(x, \ell_0)) \Rightarrow \\ \begin{bmatrix} \forall v : \mathcal{C}(\ell_2, v) \Rightarrow \mathcal{E}(x, v) \wedge \\ \forall v : \mathcal{C}(\ell_0, v) \Rightarrow \mathcal{C}(\ell, v) \end{bmatrix} \end{bmatrix} \wedge$$

$$\forall \ell : \forall \ell_1 : \forall \ell_2 : \mathsf{PRG}(\ell, \mathsf{apply}(\ell_1, \ell_2)) \Rightarrow [\forall v : \mathcal{C}(\ell_1, v) \Rightarrow \neg \mathsf{ABS}(v) \Rightarrow \mathsf{E}_{\mathsf{ABS}}(v)]$$

□

Proposition 8. *$[\![cl]\!]$ is the unique normal form of cl w.r.t. $\rightarrow$ and $[\![cl]\!] \Leftrightarrow cl$.*

The proof will be an immediate consequence of Lemmas 3 and 4 below.

Lemma 3. *$cl \rightarrow cl'$ implies $[\![cl]\!] = [\![cl']\!]$.*

A proof by structural induction on the derivation tree for $cl \rightarrow cl'$ is given in Appendix C.

Lemma 4. *Given a clause cl and $cl' = [\![cl]\!]$, then the following hold:*

1. *cl' is a reduced clause;*
2. *$cl' \Leftrightarrow cl$;*
3. *if cl is reduced then $cl' = cl$;*
4. *$cl \rightarrow^* cl'$.*

A proof of Lemma 4 by structural induction on *cl* is given in Appendix C.

Corollary 1. *The relation $\rightarrow$ is terminating and confluent.*

A proof is given in Appendix C.

5.3 Strong Stratification versus Stratification

It is fairly straightforward to show that stratified clauses are strongly stratified (c.f. the proof for Proposition 9). However the converse implication is more problematic as shown in the following example.

Example 8. Given R, $S \in \mathcal{R}$ such that $\rho(R) = 1$ and $\rho(S) = 2$, then the clause $(S \Rightarrow \mathbf{1}) \wedge R$ is strongly stratified, but not stratified. If we reduce it according to Definition 6, then we obtain R that is both stratified and strongly stratified. □

In the context of reduced clauses, the notion of *strong* stratification and the notion of stratification are equivalent as expressed by the following proposition.

Proposition 9. *A reduced clause cl is stratified* iff *cl is strongly stratified.*

A proof of Proposition 9 proceeds, as given in Appendix C, by showing (i) if *cl* is stratified, then it is stratified, i.e. $\Vdash_\rho cl : \mathsf{s}$ for $\mathsf{s} \subseteq \{0, 1, \cdots, N\}$; and (ii) if $\Vdash_\rho cl : \mathsf{s}$ and *cl* is reduced then *cl* is stratified, i.e., *cl* has the form $cl_0 \wedge \cdots \wedge cl_k$ and the properties in Definition 1 are satisfied.

Furthermore, the reduction algorithm preserves the *strong* stratification property as the following proposition says.

Proposition 10. *If cl is strongly stratified then* $[\![cl]\!]$ *is both strongly stratified (w.r.t. the same* ρ*,* N*) and stratified.*

A proof by structural induction on the derivation tree for $\Vdash_\rho cl : \mathsf{s}$ showing $\Vdash_\rho [\![cl]\!] : \mathsf{s}$ is given in Appendix C; that $[\![cl]\!]$ is stratified then follows from Proposition 9 and Lemma 4.

Example 9. Thus, if we replace the last clause in *Example 1* with the clause obtained from *Example 7*, all individual clauses are stratified. The overall clause is also a stratified clause, and hence is acceptable by the Succinct Solver.

For example, if the program is given by $((\lambda x.x)^{\ell_1} 5^{\ell_2})^{\ell}$, then we obtain from the solver that $\mathsf{E}_{\mathsf{ABS}} = \emptyset$ meaning that the hard constraint is fulfilled. If the program is given by $(5^{\ell_1}(\lambda x.x)^{\ell_2})^{\ell}$, then we obtain that $\mathsf{E}_{\mathsf{ABS}} = \{\diamond\}$ reporting that the hard constraint is violated. Clearly it would be more useful to have $\mathsf{E}_{\mathsf{ABS}} = \{(\ell, \ell_1, \ell_2, \diamond)\}$ as discussed at the end of Section 3. □

6 Worked Example

In this section we illustrate the usefulness of the transformations developed in the previous sections. The example is based on the control flow analysis developed in [7] for analyzing the Discretionary Ambients and for checking the adherence to the Bell-LaPadula mandatory access control policy [13, 8].

6.1 Discretionary Ambients

Mobile Ambients [14, 15] were introduced by Cardelli and Gordon as an abstract formalism for dealing with mobility in hierarchically structured systems. They were further developed into Safe Ambients [16, 17] by adding so-called co-capabilities for expressing discretionary access control [8]. This notion of access control was further refined in the Discretionary Ambients [7] in order to give a more faithful account of discretionary access control and in order to deal with mandatory access control [8]. In this subsection we briefly review the syntax and semantics of Discretionary Ambients.

Syntax. Let n range over names and μ range over groups, then processes P, and capabilities M in Discretionary Ambients are defined by the following syntax:

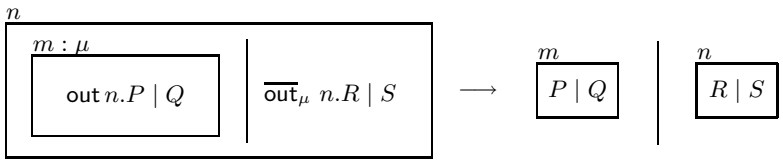
$$\begin{array}{lcl} P & ::= & (\nu\mu)P \mid (\nu n : \mu)P \mid \mathbf{0} \mid P_1 \mid P_2 \mid !P \mid n[P] \mid M.P \\ M & ::= & \mathsf{in}\ n \mid \mathsf{out}\ n \mid \mathsf{open}\ n \mid \overline{\mathsf{in}}_\mu n \mid \overline{\mathsf{out}}_\mu n \mid \overline{\mathsf{open}}_\mu n \end{array}$$

Here the construct $(\nu\mu)P$ introduces a new group μ with its scope P, and $(\nu n : \mu)P$ introduces a new ambient name n in group μ. An inactive process is denoted by $\mathbf{0}$, and the parallel composition of processes P_1 and P_2 is expressed by $P_1 \mid P_2$. The construct $!P$ replicates the process P, and $n[P]$ indicates that the process P is enclosed in the ambient named n, while $M.P$ expresses a capability followed by a process P.

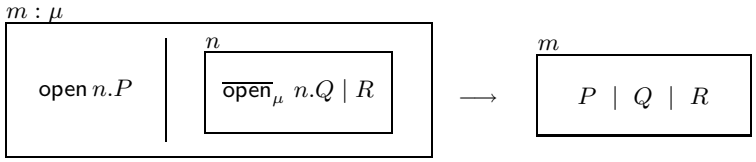
Semantics. Using boxes to represent ambients, we illustrate here informally the semantics of the capabilities (and co-capabilities) with the evolution of the configurations:

- The in-capability: the ambient named m (called the subject) of group μ has an in-capability that intends to move m into a sibling ambient named n (called the object), and n allows ambients of group μ to enter. Therefore the configuration on the left side of the arrow changes to the one on the right side after the operation.

- The out-capability: the ambient named m (called the subject) of group μ has an out-capability that intends to move m out of its parent ambient n, and n allows ambients of group μ to leave. Therefore the configuration on the left side of the arrow changes to the one on the right side after the operation.



- The open-capability: the ambient m (called the subject) of group μ has an open-capability that intends to dissolve the ambient named n (called the object), and n allows ambients of group μ to do so. Therefore the configuration on the left side of the arrow changes to the one on the right side after the operation.



In the terminology of security the semantics encodes the *reference monitor* that enforces discretionary access control; the traditional access control matrix is dispensed with and instead co-capabilities have been distributed to the appropriate places.

6.2 Control Flow Analysis

We base our work on the 1CFA analysis for Discretionary Ambients from [7]. The analysis over-approximates the behaviour of the processes by a single abstract configuration which describes both the grandparent-parent-child relations among ambients and the capabilities an ambient group may have in all the possible derivations during the process execution.

Specification. Letting Group be the set of groups, and Cap be the set of capabilities, we define a *ternary* predicate

$$\mathcal{I} \subseteq \mathsf{Group} \times \mathsf{Group} \times (\mathsf{Group} \cup \mathsf{Cap})$$

Here $\mathcal{I}(\mu_g, \mu_p, \mu)$ says that μ_g-μ_p-μ has the grandparent-parent-child relation i.e. μ_g is the grandparent, and μ_p is the parent of the ambient μ. Similarly $\mathcal{I}(\mu_p, \mu, m)$ for a capability m says that the ambient of group μ has a capability m and μ_p is the parent of μ.

The analysis is specified with judgments of the form

$$\mathcal{I} \models_{\Gamma}^{\langle \tau, \star \rangle} P$$

which expresses that $\mathcal{I}$ is an acceptable estimate for the process P when it occurs inside an ambient whose parent group is $\star$ and grandparent group is τ, and the ambient names are mapped to groups as specified by Γ.

The judgment is defined by structural induction on the syntactic constructs of the processes and is governed by following two principles:

- Each acceptable analysis estimate for a compositional process must also be an acceptable analysis estimate for its sub-processes;
- Each acceptable analysis estimate must mimic the semantics: if the semantics allows one configuration to evolve into another then it must be reflected in the analysis estimate.

The definition may be found in Table 9 where the *ternary* predicate PRG is used to describe the syntactic constructs of the process that is going to be analyzed. The function symbols in, out, open, $\overline{\mathsf{in}}$, $\overline{\mathsf{out}}$, and $\overline{\mathsf{open}}$ correspond to the capabilities, and amb that corresponds to the process construct $n[P]$.

The analysis contains three auxiliary clauses ($\mathsf{BLP}_{\mathsf{in}}, \mathsf{BLP}_{\mathsf{out}}, \mathsf{BLP}_{\mathsf{open}}$) that will be defined in Section 6.3. For the purposes of this subsection they should be considered to be $\mathbf{1}$; in [7] they are actually used to define a ternary relation $\mathcal{D} \subseteq \mathcal{I}$ of those capabilities that may indeed executed but we should not need this component here. For a more detailed description of the analysis as well as semantic soundness, we refer to [7].

The specification of Table 9 adheres to the syntax of ALFP, is stratified and hence can be implemented using the Succinct Solver.

6.3 Confidentiality Verification

Next we consider mandatory access control policy in the form of the Bell-LaPadula model (c.f. [7], Section 4.1). Our view here is that this is not part of the reference monitor; this means that it is not part of the semantics of Section 6.1 nor of the analysis of Section 6.2. Rather our view is that it is a "hard" constraint on the least solution $\mathcal{I}$.

The Bell-LaPadula model involves an assignment of security levels to each subject and object to enforce the confidentiality by *preventing information from flowing downwards from a high security level to a low security level*. The security levels form a lattice $(\mathcal{L}, \leq)$ such that $(l_1 \leq l_2)$ means that l_1 has a lower security level than l_2 for $l_1, l_2 \in \mathcal{L}$. In ALFP we formalise $(\mathcal{L}, \leq)$ by a binary relation L. In the sequel we take $\mathcal{L} = \{sec, pub\}$ with $pub \leq sec$ and hence assert $\mathsf{L}(pub, pub)$, $\mathsf{L}(pub, sec)$ and $\mathsf{L}(sec, sec)$.

$$\forall\tau:\forall *:\forall\mu:\mathsf{PRG}(\tau,*,\mathsf{amb}(\mu))\Rightarrow\mathcal{I}(\tau,*,\mu)$$

$$\forall\tau:\forall *:\forall\mu:\mathsf{PRG}(\tau,*,\mathsf{in}(\mu))\Rightarrow\left[\begin{array}{l}\mathcal{I}(\tau,*,\mathsf{in}(\mu))\wedge\\ \forall\mu^a:\forall\mu^p:\forall\mu^q:\left[\begin{array}{l}\mathcal{I}(\mu^p,\mu^a,\mathsf{in}(\mu))\wedge\\ \mathcal{I}(\mu^q,\mu^p,\mu^a)\wedge\\ \mathcal{I}(\mu^q,\mu^p,\mu)\wedge\\ \mathcal{I}(\mu^p,\mu,\overline{\mathsf{in}}(\mu^a,\mu))\end{array}\right]\\ \Rightarrow\left[\begin{array}{l}\mathcal{I}(\mu^p,\mu,\mu^a)\wedge\\ \forall u_1:\mathcal{I}(\mu^p,\mu^a,u_1)\Rightarrow\mathcal{I}(\mu,\mu^a,u_1)\wedge\\ \mathsf{BLP}_{\mathsf{in}}(\mu,\mu^a,\mu^p)\end{array}\right]\end{array}\right]$$

$$\forall\tau:\forall *:\forall\mu:\mathsf{PRG}(\tau,*,\mathsf{out}(\mu))\Rightarrow\left[\begin{array}{l}\mathcal{I}(\tau,*,\mathsf{out}(\mu))\wedge\\ \forall\mu^a:\forall\mu^g:\forall\mu^q:\left[\begin{array}{l}\mathcal{I}(\mu,\mu^a,\mathsf{out}(\mu))\wedge\\ \mathcal{I}(\mu^g,\mu,\mu^a)\wedge\\ \mathcal{I}(\mu^q,\mu^g,\mu)\wedge\\ \mathcal{I}(\mu^g,\mu,\overline{\mathsf{out}}(\mu^a,\mu))\end{array}\right]\\ \Rightarrow\left[\begin{array}{l}\mathcal{I}(\mu^q,\mu^g,\mu^a)\wedge\\ \forall u_1:\mathcal{I}(\mu,\mu^a,u_1)\Rightarrow\mathcal{I}(\mu^g,\mu^a,u_1)\wedge\\ \mathsf{BLP}_{\mathsf{out}}(\mu,\mu^a,\ \mu^g)\end{array}\right]\end{array}\right]$$

$$\forall\tau:\forall *:\forall\mu:\mathsf{PRG}(\tau,*,\mathsf{open}(\mu))\Rightarrow\left[\begin{array}{l}\mathcal{I}(\tau,*,\mathsf{open}(\mu))\wedge\\ \forall\mu^p:\forall\mu^q:\left[\begin{array}{l}\mathcal{I}(\mu^q,\mu^p,\mathsf{open}(\mu))\wedge\\ \mathcal{I}(\mu^q,\mu^p,\mu)\wedge\\ \mathcal{I}(\mu^p,\mu,\overline{\mathsf{open}}(\mu^p,\mu))\end{array}\right]\\ \Rightarrow\left[\begin{array}{l}\forall u_1:\mathcal{I}(\mu^p,\mu,u_1)\Rightarrow\mathcal{I}(\mu^q,\mu^p,u_1)\wedge\\ \mathsf{BLP}_{\mathsf{open}}(\mu,\mu^p,\ \mu^q)\end{array}\right]\end{array}\right]$$

$$\forall\tau:\forall *:\forall\mu:\forall\mu':\mathsf{PRG}(\tau,*,\overline{\mathsf{in}}(\mu,\mu'))\Rightarrow\mathcal{I}(\tau,*,\overline{\mathsf{in}}(\mu,\mu'))$$

$$\forall\tau:\forall *:\forall\mu:\forall\mu':\mathsf{PRG}(\tau,*,\overline{\mathsf{out}}(\mu,\mu'))\Rightarrow\mathcal{I}(\tau,*,\overline{\mathsf{out}}(\mu,\mu'))$$

$$\forall\tau:\forall *:\forall\mu:\forall\mu':\mathsf{PRG}(\tau,*,\overline{\mathsf{open}}(\mu,\mu'))\Rightarrow\mathcal{I}(\tau,*,\overline{\mathsf{open}}(\mu,\mu'))$$

Table 9. The 1CFA analysis for Discretionary Ambients [7] suitably simplified

In the case of our Discretionary Ambients, we consider both subjects and objects as ambients. A subject is thus an ambient possessing a capability (excluding co-capability), and an object is then an ambient that is operated upon by a subject. The access operations include all the capabilities. We assign each ambient a security level, and we consider a *secret* ambient as a protected boundary from which no information is allowed to flow out of the boundary to a *public* ambient. Thus anything is allowed to happen inside or outside the boundary but restrictions are imposed on which ambients can leave the boundary. This can be effectuated by making a number of restrictions on when operations (i.e. in, out, open) on ambients are allowed. Informally, we state these restrictions as follows:

- in: any ambient can enter any other ambient
- out: an ambient can only leave a secret ambient in a secret ambience
- open: a secret ambient can be dissolved in a secret ambience

The first item reflects that since nothing moves outwards when an in-capability is executed then the confidentiality cannot be effected. This is in contrast to the situation for an out-capability where we must prevent movement from a *secret* ambient out to a *public* ambience. Correspondingly, the third condition expresses that *secret* information inside a *secret* ambient is not allowed to flow into a *public* ambience when the *secret* ambient is opened.

These restrictions can be formalized as hard constraints on a solution $\mathcal{I}$. More precisely it will be a hard constraint on a solution ($\mathcal{I}$, PRG, L, SL) where L was specified above and $\mathsf{SL} \subseteq \mathsf{Group} \times \mathcal{L}$ records the assignment of security levels. It needs to mention that SL can be manually specified for each ambient group, or generated by a clause like

$$\forall \tau, \star, \mu, l : \mathsf{PRG}(\tau, \star, \mathsf{amb}(\mu, l)) \Rightarrow \mathsf{SL}(\mu, l)$$

provided we slightly modify the process construct $(\nu\mu)P$ to incorporate the security level information, e.g. $(\nu\mu : l)P$, and modify the function amb accordingly. It will be appropriate to consider L and SL as base predicates; in other words, L and SL will be assigned rank 0, PRG and $\mathcal{I}$ rank 1.

We then express the Bell-LaPadula conditions by defining the auxiliary clauses as follows:

$$\begin{array}{l}
\mathsf{BLP}_{\mathsf{in}}(\mu, \mu^a, \mu^p) = \mathbf{1} \\
\mathsf{BLP}_{\mathsf{out}}(\mu, \mu^a, \mu^g) = \forall s^a : \forall s^g : \mathsf{SL}(\mu^a, s^a) \wedge \mathsf{SL}(\mu^g, s^g) \Rightarrow \mathsf{L}(s^a, s^g) \\
\mathsf{BLP}_{\mathsf{open}}(\mu, \mu^p, \mu^q) = \forall s : \forall s^p : \mathsf{SL}(\mu, s) \wedge \mathsf{SL}(\mu^p, s^p) \Rightarrow \mathsf{L}(s, s^p)
\end{array}$$

The first definition corresponds to the item (in), where no constraints need to be imposed. The second definition corresponds to the item (out), where the precondition identifies a potential out-redex and the conclusion then requires that the security level of the subject μ_a is less than that of the ambience. The third definition corresponds to the item (open) and expresses the corresponding condition for the open-redex. In [7], it has been proved that if Bell-LaPadula conditions are satisfied by the analysis, then the confidentiality is ensured for the analyzed process.

Since L is a base predicate and $\mathcal{I}$ is not we have $\rho(\mathsf{L}) < \rho(\mathcal{I})$. It is then immediate that the clauses containing $\mathsf{BLP}_{\mathsf{out}}$ and $\mathsf{BLP}_{\mathsf{open}}$ in Table 9 are not (weakly) stratified. On the one hand this illustrates the naturalness of allowing non-stratified clauses to express "hard" constraints. On the other hand it shows the usefulness of the introduction of observation predicates.

The transformation then results in the following definitions

$$\begin{array}{rl}
\mathsf{BLP}_{\mathsf{out}}(\mu, \mu^a, \mu^g) = & \forall s^a : \forall s^g : \mathsf{SL}(\mu^a, s^a) \wedge \mathsf{SL}(\mu^g, s^g) \Rightarrow \\
& \neg \mathsf{L}(s^a, s^g) \Rightarrow \mathsf{E}_{\mathsf{L}}(\mu^a, \mu^g, s^a, s^g) \\
\mathsf{BLP}_{\mathsf{open}}(\mu, \mu^p, \mu^q) = & \forall s : \forall s^p : \mathsf{SL}(\mu, s) \wedge \mathsf{SL}(\mu^p, s^p) \Rightarrow \\
& \neg \mathsf{L}(s, s^p) \Rightarrow \mathsf{E}_{\mathsf{L}}(\mu, \mu^p, s, s^p)
\end{array}$$

which are weakly stratified. We say that the program satisfies the Bell-LaPadula conditions exactly when E_{L} is empty.

Compared to the development of [7] we could dispense with the auxiliary ternary relation $\mathcal{D}$ and the need to consider additional clauses, evaluated over the final solution, in order to express the desired mandatory access control policy.

Example 10. Let an ambient named a be an Internet site, and b be another site. An ambient named p is a packet. We assume that a and b are of the same groups $\hat{\mathsf{s}}$, and that p is of the group $\hat{\mathsf{p}}$, i.e. $\Gamma(a) = \hat{\mathsf{s}}$, $\Gamma(b) = \hat{\mathsf{s}}$ and $\Gamma(p) = \hat{\mathsf{p}}$. Given a process

$$a[p[\mathsf{out}\ a.\mathsf{in}\ b.\overline{\mathsf{open}}_{\hat{\mathsf{s}}}\ p] \mid \overline{\mathsf{out}}_{\hat{\mathsf{p}}}\ a] \mid b[\overline{\mathsf{in}}_{\hat{\mathsf{p}}}\ b.\mathsf{open}\ p]$$

we obtain the solution to the analysis of this process including relations associated with $\mathcal{I}$ as follows:

$$\begin{array}{l}\mathcal{I} : (\tau, \star, \hat{\mathsf{s}}), (\tau, \star, \hat{\mathsf{p}}) \\ \quad (\star, \hat{\mathsf{p}}, \mathsf{out}(\hat{\mathsf{s}})), (\star, \hat{\mathsf{p}}, \mathsf{in}(\hat{\mathsf{s}})), (\star, \hat{\mathsf{p}}, \overline{\mathsf{open}}(\hat{\mathsf{s}}, \hat{\mathsf{p}}) \\ \quad (\star, \hat{\mathsf{s}}, \mathsf{out}(\hat{\mathsf{s}})), (\star, \hat{\mathsf{s}}, \mathsf{in}(\hat{\mathsf{s}})), (\star, \hat{\mathsf{s}}, \mathsf{open}(\hat{\mathsf{s}})), \\ \quad (\star, \hat{\mathsf{s}}, \overline{\mathsf{open}}(\hat{\mathsf{s}}, \hat{\mathsf{p}})), (\star, \hat{\mathsf{s}}, \overline{\mathsf{out}}(\hat{\mathsf{s}}, \hat{\mathsf{p}})), (\star, \hat{\mathsf{s}}, \overline{\mathsf{in}}(\hat{\mathsf{s}}, \hat{\mathsf{p}})), (\star, \hat{\mathsf{s}}, \hat{\mathsf{p}}) \\ \quad (\hat{\mathsf{s}}, \hat{\mathsf{p}}, \mathsf{out}(\hat{\mathsf{s}})), (\hat{\mathsf{s}}, \hat{\mathsf{p}}, \mathsf{in}(\hat{\mathsf{s}})), (\hat{\mathsf{s}}, \hat{\mathsf{p}}, \overline{\mathsf{open}}(\hat{\mathsf{s}}, \hat{\mathsf{p}}))\end{array}$$

Now, if we give $\mathsf{SL}(\hat{\mathsf{p}}, pub)$, $\mathsf{SL}(\hat{\mathsf{s}}, sec)$, $\mathsf{SL}(\tau, sec)$ and $\mathsf{SL}(\star, sec)$, then after the test of the Bell-LaPadula conditions, it gives: $\varrho(\mathsf{E}_\mathsf{L}) = \emptyset$. That means the Bell-LaPadula conditions are fulfilled. Indeed, a public packet is allowed to pass over any site.

If we give that $\mathsf{SL}(\hat{\mathsf{p}}, sec)$, $\mathsf{SL}(\hat{\mathsf{s}}, pub)$, $\mathsf{SL}(\tau, pub)$ and $\mathsf{SL}(\star, pub)$, then $\varrho(\mathsf{E}_\mathsf{L}) = \{(\hat{\mathsf{p}}, \star, sec, pub), (\hat{\mathsf{p}}, \hat{\mathsf{s}}, sec, pub)\}$, meaning that the Bell-LaPadula conditions are failed. Indeed a secure packet is not allowed to be out of a secure boundary or to be opened in a public site. □

7 Conclusion

In this paper we have shown how to make use of non-stratified formulae in ALFP in order to express "hard" constraints in the manner familiar from strong typing. The worked example in Section 6 shows that this may lead to a much more natural formulation than is possible without this feature. By introducing suitable observation predicates we obtained a weakly stratified formula where the "hard" constraints are fulfilled exactly when the observation predicates report no violations. Further transformations, with a bounded blow-up, then produced a stratified formula acceptable to solvers such as Succinct Solver or XSB Prolog.

Acknowledgement. This work has been supported in part by the EU-project SecSafe (IST-1999-29075) and by the Danish Natural Science Research Council project LoST (21-02-0507).

References

1. P. Cousot and R. Cousot. Abstract Interpretation: A Unified Lattice Model for Static Analysis of Programs by Construction or Approximation of Fixpoint. *Proc. of 4th ACM Symposium on Principles of Programming Languages (POPL)*, pages 238-252, ACM Press, 1977.
2. F. Nielson, H. Riis Nielson, and C. Hankin. *Principles of Program Analysis*. Springer Verlag, 1999.
3. J. D. Ullman. *Principles of Database and Knowledge Base Systems*, Vol. 1. Computer Science Press, 1988.
4. F. Nielson, H. Seidl, and H. Riis Nielson. A Succinct Solver for ALFP. *Nordic Journal of Computing*, 9(4): 335-372, 2002.
5. B. Cui and D. S. Warren. A system for Tabled Constraint Logic Programming. *Computational Logic 2000*, LNAI 1861, pages 478-492, Springer-Verlag, 2000.
6. H. Riis Nielson and F. Nielson. Flow Logic: a multi-paradigmatic approach to static analysis. In the book *The Essence of Computation: Complexity, Analysis, Transformation*, LNCS 2566, pages 223-244, Springer Verlag, 2002.
7. H. Riis Nielson, F. Nielson, and M. Buchholtz. Security for Mobility. Technical Report IMM-TR-2002-20, 2002.
8. D. Gollmann. *Computer Security*. Wiley, 1999.
9. Succinct Solver web page: http://www.imm.dtu.dk/cs/Secure/SuccinctSolver.
10. H. Sun, H. Riis Nielson and F. Nielson. Extended Features in the Succinct Solver (V2.0), Technical Report SECSAFE-IMM-009, 2003.
11. A. Chandra and D. Harel. Computable Queries for Relational Data Bases. *Journal of Computer and System Sciences*, 21(2): 156-178 , 1980.
12. K. Apt, H. Blair, and A. Walker. Towards A Theory of Declarative Programming. In J. Minsker, editor, *Foundations of Deductive Databases and Logic Programming*, pages 89-148, Morgan-Kaufman, 1988.
13. D. Bell and L. LaPadula. Secure Computer Systems: Unified Exposition and Multics Interpretation. Technical Report ESDTR-75-306, MTR-2547, MITRE Corporation, 1975.
14. L. Cardelli and A. D. Gordon. Mobile Ambients. In *Foundations of Software Science and Computation Structures (FoSSaCS)*, LNCS 1378, pages 140-155, Springer Verlag, 1998.
15. L. Cardelli and A. D. Gordon. Mobile Ambients. *Theoretical Computer Science*. 240(1): 177-213, 2000.
16. F. Levi and D. Sangiorgi. Controlling Interference in Ambients. In *Proc. of the 27th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL 2000)*, pages 352-364, ACM Press, 2000.
17. F. Levi and D. Sangiorgi. Mobile Safe Ambients. *ACM Transactions on Programming Languages and Systems - TOPLAS*, 25(1): 1-69, 2003.

Appendix A: Proofs for Section 3

Proof of Proposition 2.

Proof. We prove (i) Given a clause cl then $\forall \mathsf{s} : \exists \mathsf{s}' \subseteq \mathsf{s} \cup \{N+1\}$ such that $\vdash_\rho \|cl\|^{\mathsf{s}}_\rho : \mathsf{s}'$ and (ii) If $\exists \mathsf{s}' \subseteq \mathsf{s} : \vdash_\rho cl : \mathsf{s}'$ then $\|cl\|^{\mathsf{s}}_\rho = cl$. For (i) we proceed by structural induction on cl.

Case 1. It is clear that $\vdash_\rho \|\mathbf{1}\|^{\mathsf{s}}_\rho : \mathsf{s}'$ for $\mathsf{s}' = \emptyset$ since $\|\mathbf{1}\|^{\mathsf{s}}_\rho = \mathbf{1}$, and we know $\vdash_\rho \mathbf{1} : \emptyset$ by [w1].

Case $R(\vec{x})$. In the case that $\mathsf{s} \supseteq \{\rho(R)\}$, we have $\|R(\vec{x})\|^{\mathsf{s}}_\rho = R(\vec{x})$, and we obtain $\vdash_\rho \|R(\vec{x})\|^{\mathsf{s}}_\rho : \mathsf{s}'$ for $\mathsf{s}' = \{\rho(R)\}$ according to [w2]. In the other case, we have $\|R(\vec{x})\|^{\mathsf{s}}_\rho = \neg R(\vec{x}) \Rightarrow E_R(\vec{x})$. We have already assumed that $\rho(E_R) = N+1$, then have $\vdash_\rho E_R(\vec{x}) : \mathsf{s}'$ for $\mathsf{s}' = \{\rho(E_R)\}$ by [w2]. Therefore, $\vdash_\rho \neg R(\vec{x}) \Rightarrow E_R(\vec{x}) : \mathsf{s}'$ by [w5]. Clearly $\vdash_\rho \|R(\vec{x})\|^{\mathsf{s}}_\rho : \mathsf{s}'$.

Case $\forall x : cl$. We know $\|\forall x : cl\|^{\mathsf{s}}_\rho = \forall x : \|cl\|^{\mathsf{s}}_\rho$. Applying induction hypothesis to cl we have $\vdash_\rho \|cl\|^{\mathsf{s}}_\rho : \mathsf{s}'$ for $\mathsf{s}' \subseteq \mathsf{s}$. We then obtain that $\vdash_\rho \forall x : \|cl\|^{\mathsf{s}}_\rho : \mathsf{s}'$ for $\mathsf{s}' \subseteq \mathsf{s}$ by [w3], i.e. $\vdash_\rho \|\forall x : cl\|^{\mathsf{s}}_\rho : \mathsf{s}'$.

Case $cl_1 \wedge cl_2$. It is known that $\|cl_1 \wedge cl_2\|^{\mathsf{s}}_\rho = \|cl_1\|^{\mathsf{s}}_\rho \wedge \|cl_2\|^{\mathsf{s}}_\rho$. Applying induction hypothesis to both cl_1 and cl_2 respectively, we have that $\vdash_\rho \|cl_1\|^{\mathsf{s}}_\rho : \mathsf{s}'_1$ and $\vdash_\rho \|cl_2\|^{\mathsf{s}}_\rho : \mathsf{s}'_2$ for $\mathsf{s}'_1, \mathsf{s}'_2 \subseteq \mathsf{s}$. We then have that $\vdash_\rho \|cl_1\|^{\mathsf{s}}_\rho \wedge \|cl_2\|^{\mathsf{s}}_\rho : \mathsf{s}'$ for $\mathsf{s}'_1 \cup \mathsf{s}'_2 = \mathsf{s}' \subseteq \mathsf{s}$ using [w4], and clearly $\vdash_\rho \|cl_1 \wedge cl_2\|^{\mathsf{s}}_\rho : \mathsf{s}'$.

Case $pre \Rightarrow cl$. It is known that $\|pre \Rightarrow cl\|^{\mathsf{s}}_\rho = pre \Rightarrow \|cl\|^{\delta(\mathsf{s},n)}_\rho$ for $\vdash^{\cdot}_\rho pre : n$. Applying induction hypothesis to cl, we have $\vdash_\rho \|cl\|^{\delta(\mathsf{s},n)}_\rho : \mathsf{s}'$ for $\mathsf{s}' \subseteq \delta(\mathsf{s}, n)$. According to the definition of function δ, we know $n \leq min(\delta(\mathsf{s}, n))$, and therefore, $n \leq min(\mathsf{s}')$. We then have $\vdash_\rho pre \Rightarrow \|cl\|^{\delta(\mathsf{s},n)}_\rho : \mathsf{s}'$ by [w5], and clearly, $\vdash_\rho \|pre \Rightarrow cl\|^{\mathsf{s}}_\rho : \mathsf{s}'$. □

Proof. For (ii) we also proceed by structural induction on the derivation tree for $\vdash_\rho cl : \mathsf{s}'$.

Case [w1]. Whenever $\vdash_\rho cl : \mathsf{s}'$ for $\mathsf{s}' = \emptyset$ by [w1], we $\|\mathbf{1}\|^{\mathsf{s}}_\rho = \mathbf{1}$ for $\mathsf{s}' \subseteq \mathsf{s}$.

Case [w2]. Whenever $\vdash_\rho R(\vec{x}) : \mathsf{s}'$ for $\mathsf{s}' = \{\rho(R)\}$ by [w2], we have $\|R(\vec{x})\|^{\mathsf{s}}_\rho = R(\vec{x})$ for $\mathsf{s}' \subseteq \mathsf{s}$.

Case [w3]. Assume that $\vdash_\rho \forall x : cl : \mathsf{s}'$ holds as $\vdash_\rho cl : \mathsf{s}'$ for $\mathsf{s}' \subseteq \mathsf{s}$, applying induction hypothesis to $\vdash_\rho cl : \mathsf{s}'$, we have $\|cl\|^{\mathsf{s}}_\rho = cl$. We then have $\|\forall x : cl\|^{\mathsf{s}}_\rho = \forall x : \|cl\|^{\mathsf{s}}_\rho = \forall x : cl$.

Case [w4]. Assume that $\vdash_\rho cl_1 \wedge cl_2 : \mathsf{s}'$ holds as $\vdash_\rho cl_1 : \mathsf{s}'_1$ and $\vdash_\rho cl_1 \wedge cl_2 : \mathsf{s}'$ holds as $\vdash_\rho cl_2 : \mathsf{s}'_2$ for $\mathsf{s}'_1, \mathsf{s}'_2 \subseteq \mathsf{s}$. Applying induction hypothesis to both $\vdash_\rho cl_1 : \mathsf{s}'_1$ and $\vdash_\rho cl_2 : \mathsf{s}'_2$, we have $\|cl_1\|^{\mathsf{s}}_\rho = cl_1$, and $\|cl_2\|^{\mathsf{s}}_\rho = cl_2$. We then have $\|cl_1 \wedge cl_2\|^{\mathsf{s}}_\rho = \|cl_1\|^{\mathsf{s}}_\rho \wedge \|cl_2\|^{\mathsf{s}}_\rho = cl_1 \wedge cl_2$.

Case [w5]. Assume that $\vdash_\rho pre \Rightarrow cl : \mathsf{s}'$ holds as $\vdash_\rho cl : \mathsf{s}'$ for $\mathsf{s}' \subseteq \mathsf{s}$ and $\vdash^{\cdot}_\rho pre : n$ for $n \leq min(\mathsf{s}')$. Applying induction hypothesis to $\vdash_\rho cl : \mathsf{s}'$, we have $\|cl\|^{\mathsf{s}}_\rho = cl$. We then have $\|pre \Rightarrow cl\|^{\mathsf{s}}_\rho = pre \Rightarrow x : \|cl\|^{\delta(\mathsf{s},n)}_\rho = pre \Rightarrow cl$. □

Proof of Proposition 3.

Proof. For (i) we proceed by the following structural induction on cl, and for (ii) it is analogous.

Case 1. It is clear that $(\varrho_0, \sigma) \models \mathbf{1}$ implies $(\varrho_0 \sqcup \varrho_E, \sigma) \models \|\mathbf{1}\|^{\mathsf{s}}_{\rho}$ whenever $\rho_E = \bot$ since $\|\mathbf{1}\|^{s}_{\rho} = \mathbf{1}$.

Case $R(\vec{x})$. In the case that $\mathsf{s} \supseteq \{\rho(R)\}$, we have $\|R(\vec{x})\|^{\mathsf{s}}_{\rho} = R(\vec{x})$. It is then clear that $(\varrho_0, \sigma) \models R(\vec{x})$ implies $(\varrho_0 \sqcup \varrho_E, \sigma) \models \|R(\vec{x})\|^{s}_{\rho}$ whenever $\varrho_E = \bot$. In the other case, we have $\|R(\vec{x})\|^{\mathsf{s}}_{\rho} = \neg R(\vec{x}) \Rightarrow E_R(\vec{x})$. If $\varrho_E = \bot$, then $\varrho(E_R) = \emptyset$. According to Table 1, it is the case that $(\varrho_0, \sigma) \models \neg(\neg R(\vec{x}))$, i.e. $(\varrho_0, \sigma) \models R(\vec{x})$. It is clear then $(\varrho_0, \sigma) \models R(\vec{x})$ implies $(\varrho_0 \sqcup \varrho_E, \sigma) \models \|R(\vec{x})\|^{s}_{\rho}$.

Case $\forall x : cl$. We know $\|\forall x : cl\|^{\mathsf{s}}_{\rho} = \forall x : \|cl\|^{\mathsf{s}}_{\rho}$. In the case that $\sigma[x \mapsto a] = \sigma(x)$, i.e. x does not occur free in cl, applying induction hypothesis to cl we have $(\varrho_0, \sigma) \models cl$ implies $(\varrho_0 \sqcup \varrho_E, \sigma) \models \|cl\|^{s}_{\rho}$ for $\varrho_E = \bot$. We then obtain that $(\varrho_0, \sigma) \models \forall x : cl$ implies $(\varrho_0 \sqcup \varrho_E, \sigma) \models \forall x : \|cl\|^{s}_{\rho}$ according to Table 1. In the case that $\sigma[x \mapsto a] \neq \sigma(x)$, then for each $a \in \mathcal{U}$, we have $(\varrho_0, \sigma[x \mapsto a]) \models cl$ implies $(\varrho_0 \sqcup \varrho_E, \sigma[x \mapsto a]) \models \|cl\|^{s}_{\rho}$ since each corresponds then to the previous case where no variable x occurs free. Clearly we can obtain that $(\varrho_0, \sigma) \models \forall x : cl$ implies $(\varrho_0 \sqcup \varrho_E, \sigma) \models \forall x : \|cl\|^{s}_{\rho}$, i.e. implies $(\varrho_0 \sqcup \varrho_E, \sigma) \models \|\forall x : cl\|^{s}_{\rho}$.

Case $cl_1 \wedge cl_2$. It is known that $\|cl_1 \wedge cl_2\|^{\mathsf{s}}_{\rho} = \|cl_1\|^{\mathsf{s}}_{\rho} \wedge \|cl_2\|^{\mathsf{s}}_{\rho}$. Applying induction hypothesis to both cl_1 and cl_2 respectively, we have that $(\varrho_0, \sigma) \models cl_1$ implies $(\varrho_0 \sqcup \varrho_E, \sigma) \models \|cl_1\|^{\mathsf{s}}_{\rho}$ and $(\varrho_0, \sigma) \models cl_2$ implies $(\varrho_0 \sqcup \varrho_E, \sigma) \models \|cl_2\|^{\mathsf{s}}_{\rho}$ for $\varrho_E = \bot$. We then have $(\varrho_0, \sigma) \models cl_1 \wedge cl_2$ implies $(\varrho_0 \sqcup \varrho_E, \sigma) \models \|cl_1\|^{\mathsf{s}}_{\rho} \wedge \|cl_2\|^{\mathsf{s}}_{\rho}$ according to Table 1, i.e. $(\varrho_0, \sigma) \models cl_1 \wedge cl_2$ implies $(\varrho_0 \sqcup \varrho_E, \sigma) \models \|cl_1 \wedge cl_2\|^{\mathsf{s}}_{\rho}$.

Case $pre \Rightarrow cl$. It is known that $\|pre \Rightarrow cl\|^{\mathsf{s}}_{\rho} = pre \Rightarrow \|cl\|^{\delta(\mathsf{s},n)}_{\rho}$. Applying induction hypothesis to cl, we have $(\varrho_0, \sigma) \models cl$ implies $(\varrho_0 \sqcup \varrho_E, \sigma) \models \|cl\|^{s}_{\rho}$ for $\varrho_E = \bot$. It is clear then that $(\varrho_0, \sigma) \models pre \Rightarrow cl$ implies $(\varrho_0 \sqcup \varrho_E, \sigma) \models pre \Rightarrow \|cl\|^{s}_{\rho}$ according to Table 1, and clearly $(\varrho_0, \sigma) \models pre \Rightarrow cl$ implies $(\varrho_0 \sqcup \varrho_E, \sigma) \models \|pre \Rightarrow cl\|^{s}_{\rho}$. □

Appendix B: Proofs for Section 4

Proof of Proposition 4.

Proof. It is sufficient to prove that $\Vdash_{\rho} cl : \mathsf{s}$ implies $\vdash_{\rho} cl : \mathsf{s}$. We proceed by induction on the derivation tree for $\Vdash_{\rho} cl : \mathsf{s}$.

Case [s1]. Whenever $\Vdash_{\rho} \mathbf{1} : \mathsf{s}$ holds for $\mathsf{s} = \emptyset$ by [s1], we can deduce that $\vdash_{\rho} \mathbf{1} : \mathsf{s}$ holds by [w1].

Case [s2]. Whenever $\Vdash_{\rho} R(\vec{x}) : \mathsf{s}$ holds for $\mathsf{s} = \{\rho(R)\}$, we can deduce that $\vdash_{\rho} R(\vec{x}) : \mathsf{s}$ holds by [w2].

Case [s3]. Assume that $\Vdash_{\rho} \forall x : cl : \mathsf{s}$ holds as the premise $\Vdash_{\rho} cl : \mathsf{s}$ holds for $|\ \mathsf{s}\ | \leq 1$. Applying induction hypothesis to $\Vdash_{\rho} cl : \mathsf{s}$, we have $\vdash_{\rho} cl : \mathsf{s}$. Using [w3], we obtain $\vdash_{\rho} \forall x : cl : \mathsf{s}$.

Case [s4]. Assume that $\Vdash_\rho cl_1 \wedge cl_2 : \mathsf{s}$ holds as both $\Vdash_\rho cl_1 : \mathsf{s}_1$ and $\Vdash_\rho cl_2 : \mathsf{s}_2$ hold for $\mathsf{s} = \mathsf{s}_1 \cup \mathsf{s}_2$. Applying induction hypothesis respectively to $\Vdash_\rho cl_1 : \mathsf{s}_1$ and $\Vdash_\rho cl_2 : \mathsf{s}_2$, we have $\vdash_\rho cl_1 : \mathsf{s}_1$ and $\vdash_\rho cl_2 : \mathsf{s}_2$. Using [w4], we obtain $\vdash_\rho cl_1 \wedge cl_2 : \mathsf{s}$.

Case [s5]. Assume that $\Vdash_\rho pre \Rightarrow cl : \mathsf{s}$ holds as both $\vdash_\rho^{\cdot} pre : n$ and $\Vdash_\rho cl : \mathsf{s}$ hold for $\mid \mathsf{s} \mid \leq 1 \wedge n \leq min(\mathsf{s})$. Applying induction hypothesis to $\Vdash_\rho cl : \mathsf{s}$, we obtain $\vdash_\rho cl : \mathsf{s}$. Following [w5], we have $\vdash_\rho pre \Rightarrow cl : \mathsf{s}$ for $n \leq min(\mathsf{s})$. □

Proof of Lemma 1.

Proof. In the case that $\mid \mathsf{s} \mid = 1$, it is clear that $\langle\!| cl |\!\rangle_\rho^{\mathsf{s}} = \langle\!| cl |\!\rangle_\rho^{\{j_1\}}$ for $\mathsf{s} = \{j_1\}$. In the case that $\mid \mathsf{s} \mid > 1$, $\langle\!| cl |\!\rangle_\rho^{\mathsf{s}} = \langle\!| cl |\!\rangle_\rho^{h(\mathsf{s})} \wedge \langle\!| cl |\!\rangle_\rho^{t(\mathsf{s})}$, we proceed by induction on $\mid \mathsf{s} \mid$.

Basis $\mid \mathsf{s} \mid = 2$. Then $h(\mathsf{s}) = \{j_1\}$ and $t(\mathsf{s}) = \{j_2\}$ for $\mathsf{s} = \{j_1, j_2\}$ and $j_1 < j_2$. It is clear that $\langle\!| cl |\!\rangle_\rho^{\mathsf{s}} = \langle\!| cl |\!\rangle_\rho^{\{j_1\}} \wedge \langle\!| cl |\!\rangle_\rho^{\{j_2\}}$.

Induction step $\mid \mathsf{s} \mid > 2$. Assume $\langle\!| cl |\!\rangle_\rho^{\mathsf{s}} = \langle\!| cl |\!\rangle_\rho^{\{j_1\}} \wedge \cdots \wedge \langle\!| cl |\!\rangle_\rho^{\{j_k\}}$ holds for all s such that $\mid \mathsf{s} \mid \leq k$, we prove that it also holds for $\mid \mathsf{s} \mid = k+1$. We know that $\langle\!| cl |\!\rangle_\rho^{\mathsf{s}} = \langle\!| cl |\!\rangle_\rho^{h(\mathsf{s})} \wedge \langle\!| cl |\!\rangle_\rho^{t(\mathsf{s})}$ for $\mid \mathsf{s} \mid = k+1$, and $\langle\!| cl |\!\rangle_\rho^{h(\mathsf{s})} = \langle\!| cl |\!\rangle_\rho^{\{j_1\}}$. By induction hypothesis we have $\langle\!| cl |\!\rangle_\rho^{t(\mathsf{s})} = \langle\!| cl |\!\rangle_\rho^{\{j_2\}} \wedge \cdots \wedge \langle\!| cl |\!\rangle_\rho^{\{j_{k+1}\}}$. Therefore we have $\langle\!| cl |\!\rangle_\rho^{\mathsf{s}} = \langle\!| cl |\!\rangle_\rho^{\{j_1\}} \wedge \cdots \wedge \langle\!| cl |\!\rangle_\rho^{\{j_{k+1}\}}$ holds for $\mid \mathsf{s} \mid = k+1$. □

Proof of Proposition 5.

Proof. We proceed by structural induction on cl. It needs to be mentioned that $\langle\!| cl |\!\rangle_\rho^{\mathsf{s}} = \mathbf{1}$ when $\mathsf{s} = \emptyset$, and we then obtain $\Vdash_\rho \langle\!| cl |\!\rangle_\rho^{\mathsf{s}} : \mathsf{s}'$ for $\mathsf{s}' = \emptyset$ since $\Vdash_\rho \mathbf{1} : \emptyset$ by [s1]. Therefore we will not enumerate the case of $\mathsf{s} = \emptyset$ in the proof.

Case 1. It is clear that $\Vdash_\rho \langle\!| \mathbf{1} |\!\rangle_\rho^{\mathsf{s}} : \mathsf{s}'$ for $\mathsf{s}' = \emptyset$ since $\langle\!| \mathbf{1} |\!\rangle_\rho^{\mathsf{s}} = \mathbf{1}$, and we know $\Vdash_\rho \mathbf{1} : \emptyset$ by [s1].

Case $R(\vec{x})$. In the case that $\mathsf{s} = \{\rho(R)\}$, we have $\langle\!| R(\vec{x}) |\!\rangle_\rho^{\mathsf{s}} = R(\vec{x})$, and we obtain $\Vdash_\rho \langle\!| R(\vec{x}) |\!\rangle_\rho^{\mathsf{s}} : \mathsf{s}'$ for $\mathsf{s}' = \mathsf{s}$ according to [s2]. In the other case, we have $\langle\!| R(\vec{x}) |\!\rangle_\rho^{\mathsf{s}} = \mathbf{1}$. We then have $\Vdash_\rho \langle\!| R(\vec{x}) |\!\rangle_\rho^{\mathsf{s}} : \mathsf{s}'$ for $\mathsf{s}' = \emptyset$ according to [s1].

Case $\forall x : cl$. In the case that $\mid \mathsf{s} \mid = 1$ it is known that $\langle\!| \forall x : cl |\!\rangle_\rho^{\mathsf{s}} = \forall x : \langle\!| cl |\!\rangle_\rho^{\mathsf{s}}$. Applying induction hypothesis to cl, we have $\Vdash_\rho \langle\!| cl |\!\rangle_\rho^{\mathsf{s}} : \mathsf{s}'$ for $\mathsf{s}' \subseteq \mathsf{s}$. We then obtain $\Vdash_\rho \forall x : \langle\!| cl |\!\rangle_\rho^{\mathsf{s}} : \mathsf{s}'$ using [s3] and clearly $\Vdash_\rho \langle\!| \forall x : cl |\!\rangle_\rho^{\mathsf{s}} : \mathsf{s}'$. In the case that $\mid \mathsf{s} \mid > 1$, we know $\langle\!| \forall x : cl |\!\rangle_\rho^{\mathsf{s}} = \langle\!| \forall x : cl |\!\rangle_\rho^{\{j_1\}} \wedge \cdots \wedge \langle\!| \forall x : cl |\!\rangle_\rho^{\{j_k\}}$ for $\mathsf{s} = \{j_1, \cdots, j_k\}$ and $j_1 < \cdots < j_k$ by Lemma 1. Each of the conjuncts corresponds to the case that $\mid \mathsf{s} \mid = 1$, i.e. $\Vdash_\rho \langle\!| \forall x : cl |\!\rangle_\rho^{\{j_i\}} : \mathsf{s}'_i$ such that $\mathsf{s}'_i \subseteq \{j_i\}$ for $i = 1, \cdots, k$. Using [s4], we obtain that $\Vdash_\rho \langle\!| \forall x : cl |\!\rangle_\rho^{\mathsf{s}} : \mathsf{s}'$ for $\mathsf{s}' = \mathsf{s}'_1 \cup \cdots \cup \mathsf{s}'_k$, and clearly $\mathsf{s}' \subseteq \mathsf{s}$.

Case $cl_1 \wedge cl_2$. In the case that $\mid \mathsf{s} \mid = 1$ it is known that $\langle\!| cl_1 \wedge cl_2 |\!\rangle_\rho^{\mathsf{s}} = \langle\!| cl_1 |\!\rangle_\rho^{\mathsf{s}} \wedge \langle\!| cl_2 |\!\rangle_\rho^{\mathsf{s}}$. Applying induction hypothesis to both cl_1 and cl_2 respectively, we have that $\Vdash_\rho \langle\!| cl_1 |\!\rangle_\rho^{\mathsf{s}} : \mathsf{s}'_1$ and $\Vdash_\rho \langle\!| cl_2 |\!\rangle_\rho^{\mathsf{s}} : \mathsf{s}'_2$ for $\mathsf{s}'_1, \mathsf{s}'_2 \subseteq \mathsf{s}$. We then have that $\Vdash_\rho \langle\!| cl_1 \wedge cl_2 |\!\rangle_\rho^{\mathsf{s}} : \mathsf{s}'$ for $\mathsf{s}'_1 \cup \mathsf{s}'_2 = \mathsf{s}' \subseteq \mathsf{s}$ using [s4]. In the case that $\mid \mathsf{s} \mid > 1$, we know $\langle\!| cl_1 \wedge cl_2 |\!\rangle_\rho^{\mathsf{s}} = \langle\!| cl_1 \wedge cl_2 |\!\rangle_\rho^{\{j_1\}} \wedge \cdots \wedge \langle\!| cl_1 \wedge cl_2 |\!\rangle_\rho^{\{j_k\}}$ for $\mathsf{s} = \{j_1, \cdots, j_k\}$

and $j_1 < \cdots < j_k$ by Lemma 1. Each of the conjuncts corresponds to the case that $|\ \mathsf{s}\ | = 1$, i.e. $\Vdash_\rho \langle\!| cl_1 \wedge cl_2 |\!\rangle_\rho^{\{j_i\}} : \mathsf{s}'_i$ for $i = 1, \cdots, k$. Using [s4], we obtain that $\Vdash_\rho \langle\!| cl_1 \wedge cl_2 |\!\rangle_\rho^{\mathsf{s}} : \mathsf{s}'$ for $\mathsf{s}' = \mathsf{s}'_1 \cup \cdots \cup \mathsf{s}'_k$, and clearly $\mathsf{s}' \subseteq \mathsf{s}$.

Case $pre \Rightarrow cl$. In the case that $|\ \mathsf{s}\ | = 1 \wedge \exists n \leq \min(\mathsf{s}) : \vdash_\rho^{\cdot} pre : n$ it is known that $\langle\!| pre \Rightarrow cl |\!\rangle_\rho^{\mathsf{s}} = pre \Rightarrow \langle\!| cl |\!\rangle_\rho^{\mathsf{s}}$. Applying induction hypothesis to cl, we have $\Vdash_\rho \langle\!| cl |\!\rangle_\rho^{\mathsf{s}} : \mathsf{s}'$ for $\mathsf{s}' \subseteq \mathsf{s}$. We then have $\Vdash_\rho \langle\!| pre \Rightarrow cl |\!\rangle_\rho^{\mathsf{s}} : \mathsf{s}'$ for $n \leq min(\mathsf{s}')$ from [s5]. In the other case of $|\ \mathsf{s}\ | = 1$, we have $\langle\!| pre \Rightarrow cl |\!\rangle_\rho^{\mathsf{s}} = \mathbf{1}$. We then obtain that $\Vdash_\rho \langle\!| pre \Rightarrow cl |\!\rangle_\rho^{\mathsf{s}} : \mathsf{s}'$ for $\mathsf{s}' = \emptyset$ by [s1]. In the case that $|\ \mathsf{s}\ | > 1$, we know $\langle\!| pre \Rightarrow cl |\!\rangle_\rho^{\mathsf{s}} = \langle\!| pre \Rightarrow cl |\!\rangle_\rho^{\{j_1\}} \wedge \cdots \wedge \langle\!| pre \Rightarrow cl |\!\rangle_\rho^{\{j_k\}}$ for $\mathsf{s} = \{j_1, \cdots, j_k\}$ and $j_1 < \cdots < j_k$ by Lemma 1. Each of the conjuncts corresponds to the case that $|\ \mathsf{s}\ | = 1$, i.e. $\Vdash_\rho \langle\!| pre \Rightarrow cl |\!\rangle_\rho^{\{j_i\}} : \mathsf{s}'_i$ such that $\mathsf{s}'_i \subseteq \{j_i\}$ and $n \leq min(\mathsf{s}'_i)$ for $i = 1, \cdots, k$. Using [s4], we obtain that $\Vdash_\rho \langle\!| pre \Rightarrow cl |\!\rangle_\rho^{\mathsf{s}} : \mathsf{s}'$ for $\mathsf{s}' = \mathsf{s}'_1 \cup \cdots \cup \mathsf{s}'_k$, and clearly $\mathsf{s}' \subseteq \mathsf{s}$ and $n \leq min(\mathsf{s}')$. □

Proof of Lemma 2.

Proof. We proceed by structural induction on the derivation tree for $\vdash_\rho cl : \emptyset$.

Case [w1]. Whenever $\vdash_\rho \mathbf{1} : \emptyset$, we have $\mathbf{1} \Leftrightarrow \mathbf{1}$.

Case [w2]. This case is not possible.

Case [w3]. Assume that $\vdash_\rho \forall x : cl : \emptyset$ as $\vdash_\rho cl : \emptyset$. Applying induction hypothesis to $\vdash_\rho cl : \emptyset$, we have $cl \Leftrightarrow \mathbf{1}$. It is then clear that $\forall x : cl \Leftrightarrow \mathbf{1}$.

Case [w4]. Assume that $\vdash_\rho cl_1 \wedge cl_2 : \emptyset$ as both $\vdash_\rho cl_1 : \emptyset$ and $\vdash_\rho cl_2 : \emptyset$. Applying induction hypothesis to both $\vdash_\rho cl_1 : \emptyset$ and $\vdash_\rho cl_2 : \emptyset$ respectively, we have $cl_1 \Leftrightarrow \mathbf{1}$, and $cl_2 \Leftrightarrow \mathbf{1}$. It is then clear that $cl_1 \wedge cl_2 \Leftrightarrow \mathbf{1}$.

Case [w5]. Assume that $\vdash_\rho pre \Rightarrow cl : \emptyset$ as $\vdash_\rho^{\cdot} pre : n$ and $\vdash_\rho cl : \mathsf{s}$ hold for $n \leq min(\mathsf{s})$. Applying induction hypothesis to $\vdash_\rho cl : \mathsf{s}$, we have $cl \Leftrightarrow \mathbf{1}$. It is then clear that $(pre \Rightarrow cl) \Leftrightarrow \mathbf{1}$. □

Proof of Proposition 6.

Proof. We proceed by structural induction on cl. Again $\langle\!| cl |\!\rangle_\rho^{\mathsf{s}} = \mathbf{1}$ when $\mathsf{s} = \emptyset$, thus $\langle\!| cl |\!\rangle_\rho^{\mathsf{s}} \Leftrightarrow cl \Leftrightarrow \mathbf{1}$ by Lemma 2. Therefore we will not enumerate the case of $\mathsf{s} = \emptyset$ in the proof.

Case $\mathbf{1}$. It is clear that $\mathbf{1} \Leftrightarrow \langle\!| \mathbf{1} |\!\rangle_\rho^{\mathsf{s}}$ since $\langle\!| \mathbf{1} |\!\rangle_\rho^{\mathsf{s}} = \mathbf{1}$.

Case $R(\vec{x})$. Whenever $\vdash_\rho R(\vec{x}) : \mathsf{s}$ for $\mathsf{s} = \{\rho(R)\}$, it is clear that $R(\vec{x}) \Leftrightarrow \langle\!| R(\vec{x}) |\!\rangle_\rho^{\mathsf{s}}$ since $\langle\!| R(\vec{x}) |\!\rangle_\rho^{\mathsf{s}} = R(\vec{x})$.

Case $\forall x : cl$. In the case that $|\ \mathsf{s}\ | = 1$, $\langle\!| \forall x : cl |\!\rangle_\rho^{\mathsf{s}} = \forall x : \langle\!| cl |\!\rangle_\rho^{\mathsf{s}}$. Applying induction hypothesis to cl, we have that $cl \Leftrightarrow \langle\!| cl |\!\rangle_\rho^{\mathsf{s}}$. It is then clear that $\forall x : cl \Leftrightarrow \forall x : \langle\!| cl |\!\rangle_\rho^{\mathsf{s}} \Leftrightarrow \langle\!| \forall x : cl |\!\rangle_\rho^{\mathsf{s}}$. In the case that $|\ \mathsf{s}\ | > 1$, we know that $\langle\!| \forall x : cl |\!\rangle_\rho^{\mathsf{s}} = \langle\!| \forall x : cl |\!\rangle_\rho^{\{j_1\}} \wedge \cdots \wedge \langle\!| \forall x : cl |\!\rangle_\rho^{\{j_k\}}$ for $\mathsf{s} = \{j_1, \cdots, j_k\}$ and $j_1 < \cdots < j_k$ by Lemma 1. We have further $\langle\!| \forall x : cl |\!\rangle_\rho^{\mathsf{s}} \Leftrightarrow \forall x : (\langle\!| cl |\!\rangle_\rho^{\{j_1\}} \wedge \cdots \wedge \langle\!| cl |\!\rangle_\rho^{\{j_k\}}) \Leftrightarrow \forall x : \langle\!| cl |\!\rangle_\rho^{\mathsf{s}}$ by Lemma 1. It is then clear that $\forall x : cl \Leftrightarrow \langle\!| \forall x : cl |\!\rangle_\rho^{\mathsf{s}}$.

Case $cl_1 \wedge cl_2$. In the case that $|\ \mathsf{s}\ | = 1$, $\langle\!|cl_1 \wedge cl_2|\!\rangle^{\mathsf{s}}_\rho = \langle\!|cl_1|\!\rangle^{\mathsf{s}}_\rho \wedge \langle\!|cl_2|\!\rangle^{\mathsf{s}}_\rho$. Applying induction hypothesis to both cl_1 and cl_2 respectively, we have that $cl_1 \Leftrightarrow \langle\!|cl_1|\!\rangle^{\mathsf{s}}_\rho$ and $cl_2 \Leftrightarrow \langle\!|cl_2|\!\rangle^{\mathsf{s}}_\rho$. It is then clear that $cl_1 \wedge cl_2 \Leftrightarrow \langle\!|cl_1 \wedge cl_2|\!\rangle^{\mathsf{s}}_\rho$. In the case that $|\ \mathsf{s}\ | > 1$, we know that $\langle\!|cl_1 \wedge cl_2|\!\rangle^{\mathsf{s}}_\rho = \langle\!|cl_1 \wedge cl_2|\!\rangle^{\{j_1\}}_\rho \wedge \cdots \wedge \langle\!|cl_1 \wedge cl_2|\!\rangle^{\{j_k\}}_\rho$ for $\mathsf{s} = \{j_1, \cdots, j_k\}$ and $j_1 < \cdots < j_k$ by Lemma 1. We have further $\langle\!|cl_1 \wedge cl_2|\!\rangle^{\mathsf{s}}_\rho = \langle\!|cl_1|\!\rangle^{\{j_1\}}_\rho \wedge \cdots \wedge \langle\!|cl_1|\!\rangle^{\{j_k\}}_\rho \wedge \langle\!|cl_2|\!\rangle^{\{j_1\}}_\rho \wedge \cdots \wedge \langle\!|cl_2|\!\rangle^{\{j_k\}}_\rho \Leftrightarrow \langle\!|cl_1|\!\rangle^{\mathsf{s}}_\rho \wedge \langle\!|cl_2|\!\rangle^{\mathsf{s}}_\rho$ by Lemma 1. It is then clear that $cl_1 \wedge cl_2 \Leftrightarrow \langle\!|cl_1 \wedge cl_2|\!\rangle^{\mathsf{s}}_\rho$.

Case $pre \Rightarrow cl$. Consider the case $|\ \mathsf{s}\ | = 1$. Whenever $\vdash_\rho pre \Rightarrow cl : \mathsf{s}$, there exists n such that $\vdash^{\cdot}_\rho pre : n$ and $n \leq min(\mathsf{s})$, and thus $\langle\!|pre \Rightarrow cl|\!\rangle^{\mathsf{s}}_\rho = pre \Rightarrow \langle\!|cl|\!\rangle^{\mathsf{s}}_\rho$. Applying induction hypothesis to cl, we have $cl \Leftrightarrow \langle\!|cl|\!\rangle^{\mathsf{s}}_\rho$. It is then clear that $(pre \Rightarrow cl) \Leftrightarrow (pre \Rightarrow \langle\!|cl|\!\rangle^{\mathsf{s}}_\rho)$. In the case that $|\ \mathsf{s}\ | > 1$, we know that $\langle\!|pre \Rightarrow cl|\!\rangle^{\mathsf{s}}_\rho = \langle\!|pre \Rightarrow cl|\!\rangle^{\{j_1\}}_\rho \wedge \cdots \wedge \langle\!|pre \Rightarrow cl|\!\rangle^{\{j_k\}}_\rho$ for $\mathsf{s} = \{j_1, \cdots, j_k\}$ and $j_1 < \cdots < j_k$ by Lemma 1. We have then $\langle\!|pre \Rightarrow cl|\!\rangle^{\mathsf{s}}_\rho \Leftrightarrow (pre \Rightarrow (\langle\!|cl|\!\rangle^{\{j_1\}}_\rho \wedge \cdots \wedge \langle\!|cl|\!\rangle^{\{j_k\}}_\rho)) \Leftrightarrow (pre \Rightarrow \langle\!|cl|\!\rangle^{\mathsf{s}}_\rho)$ by Lemma 1. It is clear then that $(pre \Rightarrow cl) \Leftrightarrow \langle\!|pre \Rightarrow cl|\!\rangle^{\mathsf{s}}_\rho$. □

Proofs of Proposition 7.

Proof. For the first statement: in the the case of $|\ \mathsf{s}\ | = 0$ it is clear that $|\ cl'\ | = 1$. In the case of $|\ \mathsf{s}\ | = 1$, we proceed by structural induction on cl.

Case 1. $|\ cl'\ | = |\ \mathbf{1}\ | = 1 = |\ cl\ |$.

Case $R(\vec{x})$. $|\ cl'\ | = |\ R(\vec{x})\ | = |\ cl\ |$.

Case $\forall x : cl$. We know that $\langle\!|\forall x : cl|\!\rangle^{\mathsf{s}}_\rho = \forall x : \langle\!|cl|\!\rangle^{\mathsf{s}}_\rho$. By induction hypothesis, we have $|\ \langle\!|cl|\!\rangle^{\mathsf{s}}_\rho\ | \leq |\ cl\ |$. It is then clear that $|\ \forall x : \langle\!|cl|\!\rangle^{\mathsf{s}}_\rho\ | \leq |\ \forall x : cl\ |$, and clearly $|\ \langle\!|\forall x : cl|\!\rangle^{\mathsf{s}}_\rho\ | \leq |\ \forall x : cl\ |$.

Case $cl_1 \wedge cl_2$. We know that $\langle\!|cl_1 \wedge cl_2|\!\rangle^{\mathsf{s}}_\rho = \langle\!|cl_1|\!\rangle^{\mathsf{s}}_\rho \wedge \langle\!|cl_2|\!\rangle^{\mathsf{s}}_\rho$. By induction hypothesis on both cl_1 and cl_2 respectively, we have $|\ \langle\!|cl_1|\!\rangle^{\mathsf{s}}_\rho\ | \leq |\ cl_1\ |$ and $|\ \langle\!|cl_2|\!\rangle^{\mathsf{s}}_\rho\ | \leq |\ cl_2\ |$. It is then clear that $|\ \langle\!|cl_1 \wedge cl_2|\!\rangle^{\mathsf{s}}_\rho\ | \leq |\ cl_1 \wedge cl_2\ |$.

Case $pre \Rightarrow cl$. In the case that $\exists n \leq min(\mathsf{s}) : \vdash^{\cdot}_\rho pre : n$, we know that $\langle\!|pre \Rightarrow cl|\!\rangle^{\mathsf{s}}_\rho = (pre \Rightarrow \langle\!|cl|\!\rangle^{\mathsf{s}}_\rho)$. By induction hypothesis, we have $|\ \langle\!|cl|\!\rangle^{\mathsf{s}}_\rho\ | \leq |\ cl\ |$. It is then clear that $|\ pre \Rightarrow \langle\!|cl|\!\rangle^{\mathsf{s}}_\rho\ | \leq |\ pre \Rightarrow cl\ |$, and clearly $|\ \langle\!|pre \Rightarrow cl|\!\rangle^{\mathsf{s}}_\rho\ | \leq |\ pre \Rightarrow cl\ |$. In the other case, we have $\langle\!|pre \Rightarrow cl|\!\rangle^{\mathsf{s}}_\rho = \mathbf{1}$. It is also clear that $|\ \langle\!|pre \Rightarrow cl|\!\rangle^{\mathsf{s}}_\rho\ | \leq |\ pre \Rightarrow cl\ |$. □

Proof. For the second statement, by Lemma 1 we know when cl is different from $\mathbf{1}$ and $R(\vec{x})$ and $|\ \mathsf{s}\ | > 1$, then $\langle\!|cl|\!\rangle^{\mathsf{s}}_\rho = \langle\!|cl|\!\rangle^{\{j_1\}}_\rho \wedge \cdots \wedge \langle\!|cl|\!\rangle^{\{j_k\}}_\rho$ for $\mathsf{s} = \{j_1, \cdots j_k\}$. Each of the conjuncts corresponds to the case that $|\ \mathsf{s}\ | = 1$, therefore we have that $|\ \langle\!|cl|\!\rangle^{\mathsf{s}}_\rho\ | \leq (|\ \langle\!|cl|\!\rangle^{\{j_1\}}_\rho\ | +1) + \cdots + (|\ \langle\!|cl|\!\rangle^{\{j_k\}}_\rho\ | +1)$ according to the first statement in this proposition. In the case that $cl = \mathbf{1}$, $\langle\!|cl|\!\rangle^{\mathsf{s}}_\rho = \mathbf{1}$, it is then obvious. Thus, we have $|\ \langle\!|cl|\!\rangle^{\mathsf{s}}_\rho\ | \leq |\ \mathsf{s}\ | (|\ cl\ | +1)$. □

Appendix C: Proofs for Section 5

Proof of Lemma 3.

Proof. We proceed by structural induction on the derivation tree for $cl \to cl'$.

Case [r1]. Whenever $\forall x : \mathbf{1} \to \mathbf{1}$, we have $[\![\forall x : \mathbf{1}]\!] = [\![\mathbf{1}]\!]$.
Case [r2]. Whenever $pre \Rightarrow \mathbf{1} \to \mathbf{1}$, we have $[\![pre \Rightarrow \mathbf{1}]\!] = [\![\mathbf{1}]\!]$.
Case [r3]. Whenever $\mathbf{1} \wedge cl \to cl$, we have $[\![\mathbf{1} \wedge cl]\!] = \vartheta([\![\mathbf{1}]\!] \wedge [\![cl]\!]) = [\![cl]\!]$.
Case [r4]. Whenever $cl \wedge \mathbf{1} \to cl$, we have $[\![cl \wedge \mathbf{1}]\!] = \vartheta([\![cl]\!] \wedge [\![\mathbf{1}]\!]) = [\![cl]\!]$.
Case [r5]. Assume $\forall x : cl \to \forall x : cl'$ as $cl \to cl'$. Applying induction hypothesis to $cl \to cl'$, we have $[\![cl]\!] = [\![cl']\!]$. We know $[\![\forall x : cl]\!] = \vartheta(\forall x : [\![cl]\!])$. In the case that $[\![cl]\!] = \mathbf{1}$, so is $[\![cl']\!] = \mathbf{1}$, and we have that $[\![\forall x : cl]\!] = [\![\forall x : cl']\!] = \mathbf{1}$. In the other case, $[\![\forall x : cl]\!] = \forall x : [\![cl]\!]$. It is clear then that $[\![\forall x : cl]\!] = [\![\forall x : cl']\!]$.
Case [r6]. Assume $pre \Rightarrow cl \to pre \Rightarrow cl'$ as $cl \to cl'$. Applying induction hypothesis to $cl \to cl'$, we have $[\![cl]\!] = [\![cl']\!]$. We know $[\![pre \Rightarrow cl]\!] = \vartheta(pre \Rightarrow [\![cl]\!])$. In the case that $[\![cl]\!] = \mathbf{1}$, so is $[\![cl']\!] = \mathbf{1}$, and we have that $[\![pre \Rightarrow cl]\!] = [\![pre \Rightarrow cl']\!] = \mathbf{1}$. In the other case, $[\![pre \Rightarrow cl]\!] = pre \Rightarrow [\![cl]\!]$. It is clear then that $[\![pre \Rightarrow cl]\!] = [\![pre \Rightarrow cl']\!]$.
Case [r7]. Assume $cl_1 \wedge cl_2 \to cl_1' \wedge cl_2$ as $cl_1 \to cl_1'$. Applying induction hypothesis to $cl_1 \to cl_1'$, we have $[\![cl_1]\!] = [\![cl_1']\!]$. We know $[\![cl_1 \wedge cl_2]\!] = \vartheta([\![cl_1]\!] \wedge [\![cl_2]\!])$. In the case that $[\![cl_1]\!] = \mathbf{1}$, so is $[\![cl_1']\!] = \mathbf{1}$, and we have that $[\![cl_1 \wedge cl_2]\!] = [\![cl_2]\!] = [\![cl_1' \wedge cl_2]\!]$. In the case that $[\![cl_2]\!] = \mathbf{1}$, we have that $[\![cl_1 \wedge cl_2]\!] = [\![cl_1]\!] = [\![cl_1']\!] = [\![cl_1' \wedge cl_2]\!]$. In the other case, $[\![cl_1 \wedge cl_2]\!] = [\![cl_1]\!] \wedge [\![cl_2]\!]$. It is clear then that $[\![cl_1 \wedge cl_2]\!] = [\![cl_1' \wedge cl_2]\!]$.
Case [r8]. Analogous to case [r7]. □

Proofs of Lemma 4.

Proof. For the statement (1), we proceed by structural induction on cl.

Case $\mathbf{1}$. It is clear that $\mathbf{1}$ is a reduced clause by Definition 6, therefore $[\![\mathbf{1}]\!]$ is a reduced clause, since $[\![\mathbf{1}]\!] = \mathbf{1}$.
Case $R(\vec{x})$. It is clear that $R(\vec{x})$ is a reduced clause by Definition 6, and then $[\![R(\vec{x})]\!]$ is a reduced clause, since $[\![R(\vec{x})]\!] = R(\vec{x})$.
Case $\forall x : cl$. Applying induction hypothesis to cl, we have that $[\![cl]\!]$ is a reduced clause. In the case that $[\![cl]\!] = \mathbf{1}$, $[\![\forall x : cl]\!] = \mathbf{1}$. In the other case, $[\![\forall x : cl]\!] = \forall x : [\![cl]\!]$. It is then clear that $[\![\forall x : cl]\!]$ is a reduced clause.
Case $cl_1 \wedge cl_2$. Applying induction hypothesis to cl_1 and cl_2 respectively, we have that both $[\![cl_1]\!]$ and $[\![cl_2]\!]$ are reduced clauses. In the case that $[\![cl_1]\!] = \mathbf{1}$, $[\![cl_1 \wedge cl_2]\!] = cl_2$. In the case that $[\![cl_2]\!] = \mathbf{1}$, $[\![cl_1 \wedge cl_2]\!] = cl_1$. In the other case, $[\![cl_1 \wedge cl_2]\!] = [\![cl_1]\!] \wedge [\![cl_2]\!]$. It is then clear that $[\![cl_1 \wedge cl_2]\!]$ is a reduced clause.
Case $pre \Rightarrow cl$. Applying induction hypothesis to cl, we have that $[\![cl]\!]$ is a reduced clause. In the case that $[\![cl]\!] = \mathbf{1}$, $[\![pre \Rightarrow cl]\!] = \mathbf{1}$. In the other case $[\![pre \Rightarrow cl]\!] = pre \Rightarrow [\![cl]\!]$. It is then clear that $[\![pre \Rightarrow cl]\!]$ is a reduced clause. □

Proof. For the statement (2), we proceed by structural induction on cl.

Case $\mathbf{1}$. It is clear that $\mathbf{1} \Leftrightarrow [\![\mathbf{1}]\!]$ since $[\![\mathbf{1}]\!] = \mathbf{1}$.
Case $R(\vec{x})$. It is clear that $R(\vec{x}) \Leftrightarrow [\![R(\vec{x})]\!]$ since $[\![R(\vec{x})]\!] = R(\vec{x})$.
Case $\forall x : cl$. Applying induction hypothesis to cl, we have $cl \Leftrightarrow [\![cl]\!]$. In the case that $[\![cl]\!] = \mathbf{1}$, $[\![\forall x : cl]\!] = \mathbf{1}$. In the other case, $[\![\forall x : cl]\!] = \forall x : [\![cl]\!]$. It is then clear that $\forall x : cl \Leftrightarrow [\![\forall x : cl]\!]$.
Case $cl_1 \wedge cl_2$. Applying induction hypothesis to both cl_1 and cl_2 respectively, we have $cl_1 \Leftrightarrow [\![cl_1]\!]$ and $cl_2 \Leftrightarrow [\![cl_2]\!]$. In the case that $[\![cl_1]\!] = \mathbf{1}$, $[\![cl_1 \wedge cl_2]\!] = [\![cl_2]\!]$, while $[\![cl_2]\!] = 1$, $[\![cl_1 \wedge cl_2]\!] = [\![cl_1]\!]$. In the other case, $[\![cl_1 \wedge cl_2]\!] = [\![cl_1]\!] \wedge [\![cl_2]\!]$. It is then clear that $cl_1 \wedge cl_2 \Leftrightarrow [\![cl_1 \wedge cl_2]\!]$.
Case $pre \Rightarrow cl$. Applying induction hypothesis to cl, we have $cl \Leftrightarrow [\![cl]\!]$. In the case that $[\![cl]\!] = \mathbf{1}$, $[\![pre \Rightarrow cl]\!] = \mathbf{1}$. In the other case, $[\![pre \Rightarrow cl]\!] = pre \Rightarrow [\![cl]\!]$. It is then clear that $pre \Rightarrow cl \Leftrightarrow [\![pre \Rightarrow cl]\!]$. □

Proof. For the statement (3), we proceed by structural induction on cl.

Case $\mathbf{1}$. It is clear that $\mathbf{1}$ is reduced and $[\![\mathbf{1}]\!] = \mathbf{1}$.
Case $R(\vec{x})$. It is clear $R(\vec{x})$ is reduced and $[\![R(\vec{x})]\!] = R(\vec{x})$.
Case $\forall x : cl$. Applying induction hypothesis to cl, we have $cl = [\![cl]\!]$ for cl is reduced. We know that $[\![\forall x : cl]\!] = \forall x : [\![cl]\!]$ if $\forall x : cl$ is reduced, i.e. $[\![cl]\!] \neq \mathbf{1}$. We then obtain that $[\![\forall x : cl]\!] = \forall x : cl$.
Case $cl_1 \wedge cl_2$. Applying induction hypothesis to both cl_1 and cl_2 respectively, we have $[\![cl_1]\!] = cl_1$ and $[\![cl_2]\!] = cl_2$ for both cl_1 and cl_2 are reduced. In the case that $cl_1 \wedge cl_2$ is reduced, neither $[\![cl_1]\!] = \mathbf{1}$, nor $[\![cl_2]\!] = \mathbf{1}$, therefore $[\![cl_1 \wedge cl_2]\!] = [\![cl_1]\!] \wedge [\![cl_2]\!]$. It is then clear that $[\![cl_1 \wedge cl_2]\!] = cl_1 \wedge cl_2$.
Case $pre \Rightarrow cl$. Applying induction hypothesis to cl, we have $[\![cl]\!] = cl$ for cl is reduced. In the case that $pre \Rightarrow cl$ is reduced, $[\![cl]\!] \neq \mathbf{1}$, thus $[\![pre \Rightarrow cl]\!] = pre \Rightarrow [\![cl]\!]$. It is then clear that $[\![pre \Rightarrow cl]\!] = pre \Rightarrow cl$. □

Proof. For the statement (4), it is sufficient to prove that $\exists k \in \mathbb{N} : cl \rightarrow^k [\![cl]\!]$ where k denotes the number of $\mathbf{1}$-deduction steps. We proceed by structural induction on cl.

Case $\mathbf{1}$. It is clear that $\mathbf{1} \rightarrow^k [\![\mathbf{1}]\!]$ for $k = 0$ according to the statement (3) above.
Case $R(\vec{x})$. It is clear that $R(\vec{x}) \rightarrow^k [\![R(\vec{x})]\!]$ for $k = 0$ according to the statement (3) above.
Case $\forall x : cl$. Applying induction hypothesis to cl, we have $cl \rightarrow^{k_1} [\![cl]\!]$, which can be written as $cl \rightarrow cl'' \underbrace{\rightarrow \cdots \rightarrow}_{k_1 - 1} [\![cl]\!]$ by Lemma 3. We then have $(\forall x : cl)$ $\rightarrow (\forall x : cl'') \underbrace{\rightarrow \cdots \rightarrow}_{k_1 - 1} (\forall x : [\![cl]\!])$ by repeatly applying [r5], and clearly $(\forall x : cl)$ $\rightarrow^{k_1} (\forall x : [\![cl]\!])$. In the case $[\![cl]\!] = \mathbf{1}$, we know that $[\![\forall x : cl]\!] = \mathbf{1}$. It is clear then that $(\forall x : cl) \rightarrow^k [\![\forall x : cl]\!]$ for $k = k_1 + 1$ since $(\forall x : [\![\mathbf{1}]\!]) \rightarrow \mathbf{1}$ by [r1]. In the other case, we have $[\![\forall x : cl]\!] = \forall x : [\![cl]\!]$, and it is immediate that $(\forall x : cl) \rightarrow^k [\![\forall x : cl]\!]$ for $k = k_1$.

Case $cl_1 \wedge cl_2$. Applying induction hypothesis to cl_1 and cl_2 respectively, we have that $cl_1 \rightarrow^{k_1} [\![cl_1]\!]$, and $cl_2 \rightarrow^{k_2} [\![cl_2]\!]$, which can also be written as $cl_1 \underbrace{\rightarrow \cdots \rightarrow}_{k_1} [\![cl_1]\!]$ and $cl_2 \underbrace{\rightarrow \cdots \rightarrow}_{k_2} [\![cl_2]\!]$. By repeatly applying rule [r7] k_1 times and [r8] k_2 times, we can obtain that $(cl_1 \wedge cl_2) \underbrace{\rightarrow \cdots \rightarrow}_{k_1+k_2} ([\![cl_1]\!] \wedge [\![cl_2]\!])$. In the case that $[\![cl_1]\!] = \mathbf{1}$, we can apply rule [r3] once so that $([\![cl_1]\!] \wedge [\![cl_2]\!]) \rightarrow [\![cl_2]\!]$. On the other hand we know $[\![cl_1 \wedge cl_2]\!] = [\![cl_2]\!]$ when $[\![cl_1]\!] = \mathbf{1}$. In the case that $[\![cl_2]\!] = \mathbf{1}$, we can apply rule [r4] once so that $([\![cl_1]\!] \wedge [\![cl_2]\!]) \rightarrow [\![cl_1]\!]$. We know also $[\![cl_1 \wedge cl_2]\!] = [\![cl_1]\!]$ when $[\![cl_2]\!] = \mathbf{1}$. In both cases we have $k = k_1 + k_2 + 1$ such that $(cl_1 \wedge cl_2) \rightarrow^k [\![cl_1 \wedge cl_2]\!]$. In the other case, we have $[\![cl_1 \wedge cl_2]\!] = [\![cl_1]\!] \wedge [\![cl_2]\!]$, therefore there is a $k = k_1 + k_2$ such that $(cl_1 \wedge cl_2) \rightarrow^k [\![cl_1 \wedge cl_2]\!]$.

Case $pre \Rightarrow cl$. Applying induction hypothesis to cl, we have $cl \rightarrow^{k_1} [\![cl]\!]$, which can be written as $cl \underbrace{\rightarrow \cdots \rightarrow}_{k_1} [\![cl]\!]$. We then have $(pre \Rightarrow cl) \underbrace{\rightarrow \cdots \rightarrow}_{k_1}$ $(pre \Rightarrow [\![cl]\!])$ by repeatly applying [r6], and clearly $(pre \Rightarrow cl) \rightarrow^{k_1} (pre \Rightarrow [\![cl]\!])$. In the case that $[\![cl]\!] = \mathbf{1}$ we know that $[\![pre \Rightarrow cl]\!] = \mathbf{1}$. It is clear then that $(pre \Rightarrow cl) \rightarrow^k [\![pre \Rightarrow cl]\!]$ for $k = k_1 + 1$ since $(pre \Rightarrow [\![\mathbf{1}]\!]) \rightarrow \mathbf{1}$ by [r2]. In the other case, we have $[\![pre \Rightarrow cl]\!] = (pre \Rightarrow [\![cl]\!])$, and it is immediate that $(pre \Rightarrow cl) \rightarrow^k [\![pre \Rightarrow cl]\!]$ for $k = k_1$. □

Proof of Corollary 1.

Proof. The relation $\rightarrow$ is terminating since each step of the reduction decreases the length of the clause. We can also show this fact by embedding the reduction system $(ALFP[\mathcal{X}, \mathcal{R}], \rightarrow)$ into the system $(\mathbb{N}, >)$ which is known to be terminating. For this we use the mapping $\phi : ALFP[\mathcal{X}, \mathcal{R}] \rightarrow \mathbb{N}$ given by $\phi(cl) = \mid cl \mid$ which denotes the length of cl. We proceed by contradiction. If $\rightarrow$ is not terminating, there is an infinite chain $cl_1 \rightarrow cl_2 \rightarrow \cdots$, therefore there is an infinite chain $\phi(cl_1) > \phi(cl_2) > \cdots$. This contradicts that $>$ is terminating.

To show that the relation $\rightarrow$ is also confluent, we assume that $cl \rightarrow^* cl_1$ and $cl \rightarrow^* cl_2$. Since $\rightarrow$ is terminating, there is a point during the reduction where both cl_1 and cl_2 have no further $\mathbf{1}$-reduction rule to apply, i.e. $cl_1 \rightarrow^* [\![cl_1]\!]$ and $cl_2 \rightarrow^* [\![cl_2]\!]$. Both $[\![cl_1]\!]$ and $[\![cl_2]\!]$ are reduced clauses w.r.t. $\rightarrow$ according to Lemma 4. By Lemma 3 we have $[\![cl]\!] = [\![cl_1]\!]$, and $[\![cl]\!] = [\![cl_2]\!]$. Therefore we have $[\![cl_1]\!] = [\![cl_2]\!]$. □

Proof of Proposition 9.

Proof. We shall show that (i) if cl is stratified, then $\Vdash_\rho cl : \mathsf{s}$ for $\mathsf{s} \subseteq \{0, 1, \cdots, N\}$; (ii) if $\Vdash_\rho cl : \mathsf{s}$ and cl is reduced then cl is stratified, i.e., cl has the form $cl_0 \wedge \cdots \wedge cl_k$ and the properties in Definition 1 are satisfied.

We prove (i) first. Since cl is stratified it has the form $cl_0 \wedge \cdots \wedge cl_k$, and we obtain the sequence $j_0, \cdots, j_k$ as defined in Definition 1. By repeated use of [s4]

it is sufficient to prove $\Vdash_\rho cl_i : \mathsf{s}_i$ for $\mathsf{s}_i \subseteq \{j_i\}$ and $j_i \in \{0, 1, \cdots N\}$. We proceed by structural induction on cl_i being in the j_ith stratum.

Case 1. We obtain $\Vdash_\rho \mathbf{1} : \mathsf{s}_i$ for $\mathsf{s}_i = \emptyset$ from [s1], and $\mathsf{s}_i \subseteq \{j_i\}$.

Case $R(\vec{x})$. Whenever $\rho(R) = j_i$ holds by Definition 1, we can deduce that $\Vdash_\rho R(\vec{x}) : \mathsf{s}_i$ holds for $\mathsf{s}_i = \{j_i\}$ from [s2].

Case $\forall x : cl$. Since $\forall x : cl_i$ is in stratum j_i also cl_i is in stratum j_i. Applying induction hypothesis to cl_i, we have $\Vdash_\rho cl_i : \mathsf{s}_i$ for $\mathsf{s}_i \subseteq \{j_i\}$. We then obtain $\Vdash_\rho \forall x : cl_i : \mathsf{s}_i$ from [s3].

Case $cl_1 \wedge cl_2$. Since $cl_{i_1} \wedge cl_{i_2}$ is in stratum j_i also both cl_{i_1} and cl_{i_2} are in stratum j_i. Applying induction hypothesis to both cl_{i_1} and cl_{i_2} respectively we obtain $\Vdash_\rho cl_{i_1} : \mathsf{s}_{i_1}$ and $\Vdash_\rho cl_{i_2} : \mathsf{s}_{i_2}$ for $\mathsf{s}_{i_1}, \mathsf{s}_{i_2} \subseteq \{j_i\}$. We then obtain $\Vdash_\rho cl_{i_1} \wedge cl_{i_2} : \mathsf{s}_i$ for $\mathsf{s}_i = \mathsf{s}_{i_1} \cup \mathsf{s}_{i_2}$ from [s4], and clearly $\mathsf{s}_i \subseteq \{j_i\}$.

Case $pre \Rightarrow cl$. Since $pre \Rightarrow cl_i$ is in stratum j_i also cl_i is in stratum j_i. Applying induction hypothesis to cl_i we obtain $\Vdash_\rho cl_i : \mathsf{s}_i$ for $\mathsf{s}_i \subseteq \{j_i\}$. We then obtain $\Vdash_\rho pre \Rightarrow cl_i : \mathsf{s}_i$ from [s5].

We now prove (ii). Whenever $\Vdash_\rho cl : \mathsf{s}$ and $\mid \mathsf{s} \mid > 1$, it would be the case that [s4] is the last inference rule to have been used; hence cl can be decomposed to $\Vdash_\rho cl_1 : \mathsf{s}_1$ and $\Vdash_\rho cl_2 : \mathsf{s}_2$ for $\mathsf{s}_1 \subseteq \mathsf{s}$ and $\mathsf{s}_2 \subseteq \mathsf{s}$ with $max(\mathsf{s}_1) \leq min(\mathsf{s}_2)$. When no further decomposition is possible we have $cl = cl_0 \wedge \cdots \wedge cl_k$ with $\Vdash_\rho cl_i : \mathsf{s}_i$ and $\mathsf{s}_i \subseteq \{j_i\}$ for some $j_i \in \{0, 1, \cdots, N\}$ such that $j_0 < \cdots < j_k$, and where each cl_i is reduced (as is cl). To see that cl is stratified it suffices to show that each cl_i is in stratum j_i. There are two cases to consider. If $\mathsf{s}_i = \emptyset$ it is immediate that cl_i has to be $\mathbf{1}$ (according to Fact 3). If $\mathsf{s}_i = \{j_i\}$ we proceed by structural induction on the derivation tree for $\Vdash_\rho cl_i : \mathsf{s}_i$.

Case [s1]. This case is not possible.

Case [s2]. Whenever $\Vdash_\rho R(\vec{x}) : \mathsf{s}_i$ holds for $\mathsf{s}_i = \{\rho(R)\}$ by [s2], it is clear that $R(\vec{x})$ is in stratum j_i for $j_i = \rho(R)$.

Case [s3]. Assume that $\Vdash_\rho \forall x : cl_i : \mathsf{s}_i$ holds as $\Vdash_\rho cl_i : \mathsf{s}_i$ holds for $\mathsf{s}_i = \{j_i\}$. Applying induction hypothesis to $\Vdash_\rho cl_i : \mathsf{s}_i$, we obtain that cl_i is in stratum j_i. We can then derive that $\forall x : cl_i$ is in stratum j_i.

Case [s4]. Assume that $\Vdash_\rho cl_{i_1} \wedge cl_{i_2} : \mathsf{s}_i$ holds as both $\Vdash_\rho cl_{i_1} : \mathsf{s}_{i_1}$ and $\Vdash_\rho cl_{i_2} : \mathsf{s}_{i_2}$ hold for $\mathsf{s}_i = \mathsf{s}_{i_1} \cup \mathsf{s}_{i_2} = \{j_i\}$. Applying induction hypothesis to both $\Vdash_\rho cl_{i_1} : \mathsf{s}_{i_1}$ and $\Vdash_\rho cl_{i_2} : \mathsf{s}_{i_2}$, we have that both cl_{i_1} and cl_{i_2} are in stratum j_i, hence $cl_{i_1} \wedge cl_{i_2}$ is in stratum j_i.

Case [s5]. Assume that $\Vdash_\rho pre \Rightarrow cl_i : \mathsf{s}_i$ holds as $\vdash_\rho pre : n$ and $\Vdash_\rho cl_i : \mathsf{s}_i$ hold for $n \leq min(\mathsf{s}_i)$ and $\mathsf{s}_i = \{j_i\}$. Applying induction hypothesis on $\Vdash_\rho cl_i : \mathsf{s}_i$, we have that cl_i is in stratum j_i, hence $pre \Rightarrow cl_i$ is in stratum j_i. □

Proof of Proposition 10.

Proof. We proceed by structural induction on the derivation tree for $\Vdash_\rho cl : \mathsf{s}$.

Case [s1]. Whenever $\Vdash_\rho \mathbf{1} : \mathsf{s}$ holds for $\mathsf{s} = \emptyset$ by [s1], it is clear that $\Vdash_\rho [\![\mathbf{1}]\!] : \mathsf{s}$ holds since $[\![\mathbf{1}]\!] = \mathbf{1}$.

Case [s2]. Whenever $\Vdash_\rho R(\vec{x}) : \mathsf{s}$ holds for $\mathsf{s} = \{\rho(R)\}$, we have $\Vdash_\rho [\![R(\vec{x})]\!] : \mathsf{s}$ holds since $[\![R(\vec{x})]\!] = R(\vec{x})$.

Case [s3]. Assume that $\Vdash_\rho \forall x : cl : \mathsf{s}$ holds as the premise $\Vdash_\rho cl : \mathsf{s}$ holds for $\mid \mathsf{s} \mid \leq 1$. Applying induction hypothesis to $\Vdash_\rho cl : \mathsf{s}$, we have $\Vdash_\rho [\![cl]\!] : \mathsf{s}$. In the case that $[\![cl]\!] = \mathbf{1}$, we have $[\![\forall x : cl]\!] = \mathbf{1}$ for $\mathsf{s} = \emptyset$; then we have $\Vdash_\rho [\![\forall x : cl]\!] : \mathsf{s}$. In the other case, we know that $[\![\forall x : cl]\!] = \forall x : [\![cl]\!]$. We then obtain $\Vdash_\rho [\![\forall x : cl]\!] : \mathsf{s}$ for $\mid \mathsf{s} \mid = 1$ by [s3].

Case [s4]. Assume that $\Vdash_\rho cl_1 \wedge cl_2 : \mathsf{s}$ holds as both $\Vdash_\rho cl_1 : \mathsf{s}_1$ and $\Vdash_\rho cl_2 : \mathsf{s}_2$ hold for $\mathsf{s} = \mathsf{s}_1 \cup \mathsf{s}_2$ and $max(\mathsf{s}_1) \leq min(\mathsf{s}_2)$. Applying induction hypothesis respectively to $\Vdash_\rho cl_1 : \mathsf{s}_1$ and $\Vdash_\rho cl_2 : \mathsf{s}_2$, we have $\Vdash_\rho [\![cl_1]\!] : \mathsf{s}_1$ and $\Vdash_\rho [\![cl_2]\!] : \mathsf{s}_2$. In the case that $[\![cl_1]\!] = \mathbf{1}$, we have $\mathsf{s}_1 = \emptyset$ and $[\![cl_1 \wedge cl_2]\!] = [\![cl_2]\!]$. In the case that $[\![cl_2]\!] = \mathbf{1}$, we have $\mathsf{s}_2 = \emptyset$ and $[\![cl_1 \wedge cl_2]\!] = [\![cl_1]\!]$. It is clear that $\Vdash_\rho [\![cl_1 \wedge cl_2]\!] : \mathsf{s}$ in both cases. In the other case $[\![cl_1 \wedge cl_2]\!] = [\![cl_1]\!] \wedge [\![cl_2]\!]$. We then obtain $\Vdash_\rho [\![cl_1 \wedge cl_2]\!] : \mathsf{s}$ by [s4].

Case [s5]. Assume that $\Vdash_\rho pre \Rightarrow cl : \mathsf{s}$ holds as both $\vdash^{\cdot}_\rho pre : n$ and $\Vdash_\rho cl : \mathsf{s}$ hold for $\mid \mathsf{s} \mid \leq 1 \wedge n \leq min(\mathsf{s})$. Applying induction hypothesis to $\Vdash_\rho cl : \mathsf{s}$, we obtain $\Vdash_\rho [\![cl]\!] : \mathsf{s}$. In the case that $[\![cl]\!] = \mathbf{1}$, we have $[\![pre \Rightarrow cl]\!] = \mathbf{1}$ and $\mathsf{s} = \emptyset$; then we have $\Vdash_\rho [\![pre \Rightarrow cl]\!] : \mathsf{s}$. In the other case, we know that $[\![pre \Rightarrow cl]\!] = pre \Rightarrow [\![cl]\!]$. We then obtain $\Vdash_\rho [\![pre \Rightarrow cl]\!] : \mathsf{s}$ for $\mid \mathsf{s} \mid = 1$ by [s5]. □