Formal Methods in System Design
https://doi.org/10.1007/s10703-022-00404-9

Stratified guarded first-order transition systems

Christian Müller[1] · Helmut Seidl[1]

Received: 16 October 2021 / Accepted: 17 October 2022
© The Author(s) 2022

Abstract
First-order transition systems are a convenient formalism to specify parametric systems such as multi-agent workflows or distributed algorithms. In general, any nontrivial question about such systems is undecidable. Here, we present three subclasses of first-order transition systems where every universal invariant can effectively be decided via fixpoint iteration. These subclasses are defined in terms of syntactical restrictions: negation, stratification and guardedness. While guardedness represents a particular pattern how input predicates control existential quantifiers, stratification limits the information flow between predicates. Guardedness implies that the weakest precondition for every universal invariant is again universal, while the remaining sufficient criteria enforce that either the number of occurring negated literals decreases in every iteration, or the number of required instances of input predicates or the number of first-order variables remains bounded. We argue for each of these three cases that termination of the fixpoint iteration can be guaranteed. We apply these results to identify classes of multi-agent systems, when formalized as first-order transition systems, where noninterference in presence of declassification is decidable for coalitions of attackers of bounded size.

Keywords First-order transition systems · Multi-agent systems · Universal invariants · Noninterference · Decidability

1 Introduction

FO transition systems (FO for First-order) are a convenient tool for specifying systems where the number of agents is not known in advance. This is very useful for modeling systems like network protocols [1] or web-based workflows like conference management, banking or commerce platforms. Consider, e.g., the specification from Fig. 1 modeling parts of the review process of a conference management system as a FO transition system.

Christian Müller and Helmut Seidl have contributed equally to this work.

✉ Helmut Seidl
seidl@in.tum.de

Christian Müller
christian.mueller@in.tum.de

[1] Fakultät für Informatik, TU München, Boltzmannstraße 3, 45748 Garching, Germany

Published online: 22 November 2022

Assume that initially, all predicates with the exception of auth are false, i.e., the property $\mathcal{H}$ given by

$$\forall x_1, x_2, p, r, d. \neg \mathsf{conf}(x_1, p) \wedge \neg \mathsf{assign}(x_1, p) \wedge \\ \neg \mathsf{report}(x_1, p, r) \wedge \neg \mathsf{discuss}(x_1, x_2, p, d) \tag{1}$$

holds. The predicates $A_1, \ldots, A_4$ are *input predicates* whose values either represent agents' decisions or input from the environment. Intuitively, the transition system works as follows: First, each PC member x possibly declares her conflict with each paper p. Then the assignment relation of papers p to PC members x is extended in such a way that the conf relation is respected. Repeatedly, reports for PC members x about papers p arrive, where a subsequent discussion between PC members x_1, x_2 on some paper p is only possible if both have received a report on that paper and may update their reviews based on the discussions. Variants of this example have already been studied in [2, 3].

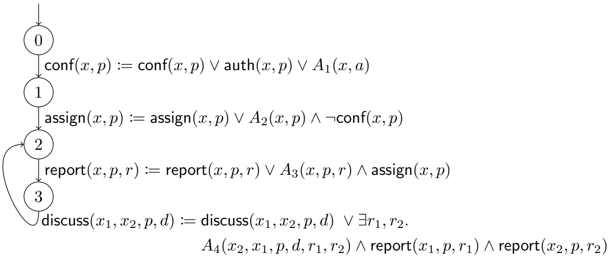
A useful property to ensure in this example is that a discussion between x_1 and x_2 on some paper p is only possible if neither x_1 nor x_2 are authors of p:

$$\forall x_1, x_2, p, d. \neg \mathsf{discuss}(x_1, x_2, p, d) \vee \neg \mathsf{auth}(x_1, p) \wedge \neg \mathsf{auth}(x_2, p) \tag{2}$$

As FO predicate logic is undecidable, we cannot hope to find an effective algorithm for proving an invariant such as (2) for arbitrary FO transition systems. That does not exclude, though, that at least some invariants can be proven *inductive* and thus, to be valid. Also, approximation techniques may be conceived to construct *strengthenings* of given invariants which, hopefully turn out to be inductive and thus may serve as certificates for the invariants in question.

The idea of using FO predicate logic for specifying the semantics of systems has perhaps been pioneered by abstract state machines (ASMs) [4–6]. Recently, it has successfully been applied for the specification and verification of software-defined networks [7, 8], of network protocols [9], of distributed algorithms [1]. The corresponding approach is built into the tool Ivy [9, 10]. Ivy is a proof assistant for systems specified in FO logic which is carefully designed around a decidable many-sorted extension of EPR (Effectively Propositional Logic, or $\exists^*\forall^*$FO logic). In the base setting, invariants are provided manually and then checked for inductiveness by the theorem prover Z3 [11]. Some effort, though, has been invested to come up with more automatic techniques for specific settings such as threshold algorithms [12] or more general FO invariant inference [13, 14]. The fundamental problem thereby is that repeated application of the weakest precondition operator may introduce additional first-order variables, new instances of input predicates or existential quantifiers and thus result in formulas outside the decidable fragment of FO logic.

Fig. 1 A conference management system


0
$\mathsf{conf}(x,p) \coloneqq \mathsf{conf}(x,p) \vee \mathsf{auth}(x,p) \vee A_1(x,a)$
1
$\mathsf{assign}(x,p) \coloneqq \mathsf{assign}(x,p) \vee A_2(x,p) \wedge \neg\mathsf{conf}(x,p)$
2
$\mathsf{report}(x,p,r) \coloneqq \mathsf{report}(x,p,r) \vee A_3(x,p,r) \wedge \mathsf{assign}(x,p)$
3
$\mathsf{discuss}(x_1,x_2,p,d) \coloneqq \mathsf{discuss}(x_1,x_2,p,d) \ \vee \exists r_1, r_2.$
$A_4(x_2,x_1,p,d,r_1,r_2) \wedge \mathsf{report}(x_1,p,r_1) \wedge \mathsf{report}(x_2,p,r_2)$

This problem also has been encountered in [3, 15, 16] where noninterference [17] is investigated for multi-agent workflows in the spirit of the conference management system from Fig. 1. In [3], the authors present a a symbolic verification approach where the agent capabilities as well as declassification and self-composition of the original system $\mathcal{T}$ is encoded into a FO transition system $\mathcal{T}^2$. Noninterference of the original system is thus reduced to a universal invariant of the resulting system $\mathcal{T}^2$. Further abstraction (i.e., strengthening of the encountered formulas) is applied in order to arrive at a practical algorithm which iteratively strengthens the initial invariant.

Only for rare cases, so far, decidability could be shown. In [18], Sagiv et al. show that inferring universal inductive invariants is decidable when the transition relation is expressed by formulas with unary predicates and a single binary predicate restricted by the background theory of singly-linked-lists. The same problem becomes undecidable when the binary symbol is not restricted by a background theory. In [3] on the other hand, syntactic restrictions are introduced under which termination at least of an *abstract* fixpoint iteration can be guaranteed. The abstraction thereby, consists in strengthening each occurring existential quantifier via appropriate instantiations (see also [19]). The syntactic restrictions proposed in [3] essentially amount to introducing a *stratification* on the predicates and restricting substitutions to be *stratified guarded updates*. It is argued that these restrictions are not unrealistic in specifications of multi-agent systems where the computation proceeds in stages each of which accumulates information based on the results obtained in earlier stages. The example transition system from Fig. 1, e.g., is stratified: there is a mapping λ assigning a *level* $\lambda(R)$ to each predicate R so that the predicates occurring in right-hand sides which are distinct from the left-hand side have lower levels. In the example, λ could be given by

$$\{\mathsf{auth} \mapsto 0, \mathsf{conf} \mapsto 1, \mathsf{assign} \mapsto 2, \mathsf{report} \mapsto 3, \mathsf{discuss} \mapsto 4\}$$

Intuitively, stratification limits dependencies between predicates to be acyclic. Examples of *stratified guarded updates* on the other hand, are the two statements in the loop body of Fig. 1. *Guarded updates* only allow to extend predicates where the extensions constrain the use of existential quantifiers to the format $\varphi \vee \exists \bar{z}.A\bar{y}\bar{z} \wedge \psi$ for some input predicate A and quantifier-free subformulas φ, ψ.

The loop of the example thus satisfies the requirements of [3], implying that an *abstract* fixpoint iteration is guaranteed to terminate for every universal invariant. Here, we show that under the given assumptions, *no abstraction* is required: the *concrete* fixpoint iteration in question already terminates and returns the weakest inductive invariant, which happens to consist of universal formulas only. We conclude that universal invariants for the given class of FO transition systems are decidable.

Beyond that, we extend this class of FO transition systems by additionally allowing stratified guarded *resets*. Resets are seemingly *easier* than updates, as they define their left-hand sides solely in terms of predicates of lower levels without resorting to input predicates. In full generality, though, when there are both updates and resets, we *failed* to prove that universal invariants are decidable. We only succeed so — provided further (mild) restrictions are satisfied. Our results are that jointly, stratified guarded updates and resets can be allowed

- when all updates are not only guarded, but *strictly* guarded; or
- when all substitutions are single, and all predicates of level at least 1, occur in right-hand sides only positively; or

- when resets refer to predicates at the highest and at the lowest level of the stratification only.

Data sharing is not applicable to this article as no new data were created or analyzed in this study. A preliminary version of this paper has appeared in [20]. There just *single* updates of predicates at a time have been considered — while *simultaneous* updates are crucial for the application to noninterference in multi-agent systems with declassification where agent capabilities are taken into account (NDA). A detailed study of that application makes up the second part of the paper. For this, we review the constructions for self-composition from [3] and present their adaptations to FO transition systems. Based on these constructions, we provide general classes of multi-agent systems, when formalized as FO transition systems, where NDA is in fact *decidable*. In this way, we strengthen the results from [3] in two ways: on the one hand, our systems are more general in that (some forms of) *resets* are additionally allowed; on the other, we obtain *decidability* of NDA where termination could only be proven for an *abstraction* of the system.

2 Basic definitions

Assume that we are given a finite set of predicate names $\mathcal{R}$ together with a finite set of constant names $\mathcal{C}$. A *FO structure* $s = \langle I, \rho \rangle$ over a given universe $\mathcal{U}$ consists of an *interpretation* I of the predicates in $\mathcal{R}$, i.e., a mapping which assigns to each predicate $R \in \mathcal{R}$ of arity $k \geq 0$, a k-ary relation over $\mathcal{U}$, together with a valuation $\rho : \mathcal{C} \to \mathcal{U}$ which assigns to each constant name an element in $\mathcal{U}$. The *semantics* of FO (first-order) formulas as well as SO (second-order) formulas with free occurrences of predicates and variables in $\mathcal{R}$ and $\mathcal{C}$, respectively, is defined as usual. We write $s \models \varphi$ or $I, \rho \models \varphi$ to denote that φ is valid for the given interpretation I and valuation ρ as provided by s. For FO transition systems, we distinguish between the set $\mathcal{R}_{state}$ of *state predicates* and the disjoint set $\mathcal{A}$ of *input predicates*. While the values of constants as well as the interpretation of the state predicates constitute the state attained by the system, the input predicates are used to model (unknown) input from the environment or decisions of participating agents.

At each transition of a FO transition system, the system state s' after the transition is determined in terms of the system state s before the transition via an *assumption* formula g and a *substitution* θ. The transition can only take place when the assumption formula g is satisfied. For each state predicate $R \in \mathcal{R}_{state}$, θ then provides a FO formula to specify the interpretation of R after the transition in terms of the interpretation and valuation in s.

Technically, we introduce a set $\mathcal{Y} = \{y_i \mid i \in \mathbb{N}\}$ of distinct formal parameters where $\mathcal{C} \cap \mathcal{Y} = \emptyset$. For a predicate R of arity $k \geq 0$, we write $R\bar{y}_R$ for the literal $R(y_1, \dots, y_k)$ and assume that each substitution θ maps each literal $R\bar{y}_R$, $R \in \mathcal{R}_{state}$, to some FO formula $\theta(R\bar{y}_R)$ with predicates in $\mathcal{R}_{state} \cup \mathcal{A}$ and free variables either from $\mathcal{C}$ or occurring among the variables in $\bar{y}$. In case that $\theta(R_i\bar{y}_{R_i}) = \psi_i$ for $i = 1, \dots, r$, and $\theta(R'\bar{y}_{R'}) = R'\bar{y}_{R'}$ for all $R' \in \mathcal{R}_{state} \setminus \{R_1, \dots, R_r\}$, we also denote θ by

$$\{\, R_1\bar{y}_{R_1} := \psi_1; \dots ; R_r\bar{y}_{R_r} := \psi_r \,\}$$

or just

$$R_1\bar{y}_{R_1} := \psi_1$$

if $r = 1$. In this case, we call θ *single*. Finally, the assumption formula g at edges is omitted if it equals $\top$.

Example 1 In the example from Fig. 1, $\mathcal{R}_{state}$ consists of the predicates conf, auth, assign, report and $\mathsf{discuss}$ while $\mathcal{A}$ consists of the predicates $A_1 \dots A_4$. No constants are needed, so $\mathcal{C} = \emptyset$. The edge from node 1 to 2 has no assumption and specifies a single substitution θ that updates assign with

$$\theta(\mathsf{assign}(x,p)) = \mathsf{assign}(x,p) \vee A_2(x,p) \wedge \neg\mathsf{conf}(x,p)$$

while leaving literals of predicates conf, auth, report or $\mathsf{discuss}$ unchanged. □

Applying the substitution θ to a FO formula φ results in the FO formula $\theta(\varphi)$ which is obtained from φ by replacing each literal $R\bar{z}$ with the FO formula $\theta(R\bar{y}_R)[\bar{z}/\bar{y}_R]$. Here, $[\bar{z}/\bar{y}_R]$ represents the simultaneous substitution of the variables in $\bar{y}_R$ by the corresponding variables in $\bar{z}$ (perhaps, with appropriate renaming of bound variables in the formula $\theta(R\bar{y}_R)$).

Example 2 Consider formula φ that specifies that the author of a paper p should never be assigned to provide a review for p:

$$\varphi = \forall x, p.\neg\mathsf{assign}(x,p) \vee \neg\mathsf{auth}(x,p)$$

Applying the substitution θ from Example 1 results in

$$\begin{aligned}\theta(\varphi) &= \forall x, p.\neg(\mathsf{assign}(x,p) \vee A_2(x,p) \wedge \neg\mathsf{conf}(x,p)) \vee \neg\mathsf{auth}(x,p)\\ &\leftrightarrow \forall x, p.\neg\mathsf{assign}(x,p) \wedge (\neg A_2(x,p) \vee \mathsf{conf}(x,p)) \vee \neg\mathsf{auth}(x,p)\end{aligned}$$

□

A FO transition system $\mathcal{T}$ (over the given sets $\mathcal{R}_{state}$ of predicates, $\mathcal{A}$ of input predicates and $\mathcal{C}$ of constant names) consists of a finite set of nodes V together with a finite set E of edges of the form $e = (u, (g;\theta), v)$ where $u, v \in V$, and g, θ specify an assumption and substitution of the predicates in $\mathcal{R}_{state}$, respectively. W.l.o.g., we assume that the assumption g and substitution θ at each edge e always has occurrences of at most one input predicate, which we denote by A_e. For a given universe $\mathcal{U}$, a program state s attained at a program point is a FO structure for the predicates in $\mathcal{R}_{state}$ and the constants in $\mathcal{C}$ over the universe $\mathcal{U}$. Let S denote the set of all program states. A *configuration* of $\mathcal{T}$ is a pair $(v, s) \in V \times S$. A (finite) *run* τ of $\mathcal{T}$ starting in configuration (v_0, s_0) and ending at node v in state s, i.e., in configuration (v, s) is a sequence of configurations (v_i, s_i), $i = 0, \dots, n$ where $(v_n, s_n) = (v, s)$ and for all $i = 1, \dots, n$, there is some edge $e_i = (v_{i-1}, (g_i;\theta_i), v_i) \in E$ such that for $s_{i-1} = \langle I, \rho\rangle$ and some interpretation R_i of the input predicate A_{e_i}, the following holds:

- $I \oplus \{A_{e_i} \mapsto R_i\}, \rho \models g$, and
- $s_i = \langle I', \rho\rangle$ where for every state predicate R and valuation ρ_R of the formals y_i occurring in $\bar{y}_R$,

$$I', \rho \oplus \rho_R \models R\bar{y}_R \text{ iff } I \oplus \{A_{e_i} \mapsto R_i\}, \rho \oplus \rho_R \models \theta(R\bar{y}_R)$$

Here, the operator $\oplus$ is meant to extend the assignment in the left argument with the assignment to the right.

Subsequently, we assume that we are given an *initial node* $v_0 \in V$ together with an *initial hypothesis* $\mathcal{H}$, i.e., a FO formula (with predicates in $\mathcal{R}_{state}$ and free variables only in $\mathcal{C}$) characterizing all possible initial states attained at v_0.

Example 3 According to the specification in eq. (1) for the example transition system in Fig. 1, the single initial state (w.r.t. any given universe) is the pair of node 0 and the FO structure which interprets the relations auth, assign, report and discuss with empty relations each. □

Input predicates may take fresh interpretations whenever the substitution of the corresponding edge is executed. This should be contrasted to state predicates whose interpretations stay the same if they are not explicitly updated by the transition system. State predicates which are never updated, thus have *constant* interpretations. These still may be constrained by some background theory provided via conjuncts of the initial hypothesis.

Assume that Ψ assigns to each program point $v \in V$, a FO formula $\Psi[v]$. Then Ψ is a *valid invariant* (relative to the initial hypothesis $\mathcal{H}$), if every run τ of the system starting in a configuration (v_0, s_0) with $s_0 \models \mathcal{H}$ and visiting some configuration (v, s), it holds that $s \models \Psi[v]$. Ψ is *inductive* if

$$\Psi[u] \to \theta(\Psi[v]) \qquad \text{forall } (u, (g;\theta), v) \in E \tag{3}$$

If Ψ is inductive, then Ψ is a valid whenever

$$\mathcal{H} \to \Psi[v_0] \tag{4}$$

Indeed, it is this observation which is used in the Ivy project to verify distributed algorithms such as the Paxos protocol, essentially, by manually providing the invariant Ψ and verifying properties (3) and(4) via the theorem prover Z3 [11].

Not every valid invariant Ψ, though, is by itself inductive. If this is not yet the case, iterative *strengthenings* $\Psi^{(h)}, h \geq 0$, of Ψ may be computed as follows:

$$\begin{aligned} \Psi^{(0)}[u] =& \Psi[u] \qquad \text{and for } \ h > 0, \\ \Psi^{(h)}[u] =& \Psi^{(h-1)}[u] \wedge \bigwedge_{e=(u,(g;\theta),v)\in E} \forall A_e.\, (\neg g \vee \theta(\Psi^{(h-1)}[v]) \end{aligned} \tag{5}$$

For computing the next iterate in (5), universal SO quantification over the input predicate A_e is required in order to account for *every* input possibly occurring during a run when executing the edge e. As, e.g., noted in [2], $s \models \Psi^{(h)}[u]$ iff every run of length at most h starting in (u, s), ends in some configuration (u', s') with $s' \models \Psi[u']$. In particular, the assignment Ψ is a valid invariant iff $\mathcal{H} \to \Psi^{(h)}[v_0]$ for all $h \geq 0$. The iteration thus can be considered as computing the *weakest pre-condition* of the given invariant Ψ – as opposed to the *collecting semantics* of the FO transition system, which corresponds to the set of all configurations reachable from the set of initial configurations $(v_0, s), s \models \mathcal{H}$. Whenever the fixpoint iteration (5) terminates, we obtain the *weakest* strengthening of the given invariant Ψ which is inductive. We have:

Lemma 1 *Let $\mathcal{T}$ be a FO transition system and let Ψ an invariant. Assume that for some $h \geq 0, \Psi^{(h)} = \Psi^{(h+1)}$ holds. Then $\Psi^{(h)}$ is the weakest inductive invariant implying Ψ. Moreover, Ψ is valid iff $\mathcal{H} \to \Psi^{(h)}[v_0]$.* □

In general, the required SO quantifier elimination may not always be possible, i.e., there need not always exist an equivalent FO formula [21], and even if SO quantifier elimination is always possible, the fixpoint iteration need not terminate. Non-termination may already occur when all involved predicates either have no arguments or are *monadic* [2]. Termination as well as effective computability can be enforced by applying *abstraction* (see, e.g., [22] for a general discussion). Applying an abstraction α amounts to computing a *sufficient* condition for the invariant Ψ to hold. Technically, an abstraction maps each occurring formula ψ to a formula $\alpha[\psi]$ (hopefully of a simpler form) so that $\alpha[\psi] \to \psi$. Subsequently, we list three examples for such strengthenings.

Example 4 In [3], formulas with universal SO quantifiers and universal as well as existential quantifiers are strengthened to formulas with universal quantifiers only. The idea is to replace an existentially quantified subformula $\exists x.\varphi$ with a disjunction $\bigvee_{y \in Y} \varphi[y/x]$ where Y is the subset of constants and those universally quantified variables in whose scope φ occurs. So, the formula $\forall y_1, y_2.\exists x.R(x)$ is abstracted by $\forall y_1, y_2.R(y_1) \vee R(y_2)$. This abstraction is particularly useful, since SO universal quantifiers can be eliminated from universally quantified formulas. □

Example 5 Fixpoint iteration for universally quantified formulas still may not terminate due to an ever increasing number of quantified variables. The universally quantified variable x in an otherwise quantifier-free formula ψ in negation normal form can be removed by replacing each literal containing x with `false`. In this way, the formula $\forall x.\ (Rx \vee \neg Sy \vee Tz) \wedge (\neg Rx \vee \neg Ty)$ is strengthened to $(\neg Sy \vee Tz) \wedge \neg Ty$. □

Example 6 Assume that the quantifier-free formula ψ is a conjunction of clauses. Then ψ is implied by the single clause c consisting of all literals which all clauses in ψ have in common. The formula $(Rx \vee \neg Sy \vee Tz) \wedge (Rx \vee Tz \vee \neg Tx)$, e.g., can be strengthened to $Rx \vee Tz$. □

In this article, rather than focusing on abstractions, we identify sufficient criteria for the concrete iteration (5) to terminate without any abstraction.

3 Stratification and guardedness

Subsequently, we concentrate on initial conditions in the $\exists^*\forall^*$ fragment and *universal* invariants, i.e., where the invariant Ψ consists of *universal* FO formulas only. Already for this setting, non-termination of the inference algorithm may occur even without SO quantification when a single binary predicate is involved.

Example 7 Consider the FO transition system $\mathcal{T}$ over a monadic state predicate R, a binary state predicate E and a constant a. $\mathcal{T}$ consists of a single node u with a single transition (u, θ, u) where θ is given by

$$R(y) := R(y) \vee \exists z.\, E(y,z) \wedge R(z)$$

Consider the invariant $\Psi[u] = \neg R(a)$. Then for $h \geq 0$,

$$\Psi^{(h)}[u] = \neg R(a) \wedge \bigwedge_{k=1}^{h} \forall z_1, \dots, z_k.\, \neg E(a, z_1) \vee \bigvee_{i=1}^{k-1} \neg E(z_i, z_{i+1}) \vee \neg R(z_k)$$

The weakest inductive invariant thus states that the unary predicate R only holds for elements which are not reachable from a via the edge predicate E. This property is not expressible in FO predicate logic. Accordingly, $\Psi^{(h)}[u] \neq \Psi^{(h+1)}[u]$ must hold for all $h \geq 0$. □

Our goal is to identify useful non-trivial classes of FO transition systems where the fixpoint iteration is guaranteed to terminate. One ingredient for this definition is a stratification mapping $\lambda : \mathcal{R}_{state} \to \mathbb{N}$ which assigns to each state predicate R a *level* $\lambda(R)$. Intuitively, this mapping is intended to describe how the information flows between predicates. Thereby, we use the convention that $\lambda(R) = 0$ only for predicates R which are never substituted, i.e., whose values are constant throughout each run of the transition system.

We only consider substitutions which are either *guarded updates* or *resets*. Thereby, a substitution θ is called a (simultaneous) *guarded update* if it simultaneously modifies predicates $R_1, \dots, R_r \in \mathcal{R}_{state}$ by

$$R_j \bar{y}_{R_j} := \quad R_j \bar{y}_{R_j} \vee \varphi_j \vee \exists \bar{z}_j.\, A \bar{y}_{R_j}^{\sigma_j} \bar{z}_j \wedge \psi_j \tag{6}$$

where $A \in \mathcal{A}$ is the input predicate of θ, $\bar{y}_{R_j}^{\sigma_j}$ is the sequence of variables obtained by applying the permutation σ_j to the sequence of FO variables $\bar{y}_{R_j}$, $\bar{z}_j$ is a sequence of FO variables, dedicatedly used for the right-hand side of R_j in θ, and φ_j, ψ_j are quantifier-free FO formulas without occurrences of predicate A which do not contain occurrences of left-hand sides $R_1, \dots, R_r$ either. We remark here that, technically speaking, allowing permutations σ_j does not introduce extra technical difficulties, but conveniently supports specifications, e.g., as in Fig. 1 where a non-trivial permutation is used for the predicate discuss (see section 8 for a usage of such reordering).

The guarded update is called *strict* if additionally, all formulas φ_i are $\perp$ (*false*), i.e., are missing. Note that we do *not* assume here that all left-hand sides in updates agree in their arities. Instead, for each j, the sum of the lengthes of $\bar{y}_{R_j}$ and $\bar{z}_j$ should all equal the arity of the input predicate A of the update. Accordingly, also the sequences of existentially quantified variables in right-hand sides of guarded updates need not agree in their lengthes. Guarded updates according to this definition, are specific instances of the basic constituents of *workflows* as considered in [3, 15]. There, simultaneous updates are called *blocks*, where the left-hand sides R_i are allowed to be extended (or filtered) by means of arbitrary FO formulas. Here, we only allow extensions and only by means of formulas of the restricted form (6).

Additionally, we now also allow substitutions to be *resets*. The substitution θ is called a *reset*, if it substitutes just a single predicate R by means of

$$R \bar{y}_R := \quad \varphi \tag{7}$$

where φ is a quantifier-free FO formula without occurrences of the left-hand side R or any input predicate. The update θ is called *stratified*, if additionally all substituted predicates R_j have identical level, and each predicate R' occurring in the φ_j, ψ_j has level less than $\lambda(R_j)$.

Likewise, a reset is called *stratified*, if all predicates occurring in the right-hand side for the substituted predicate R have levels less than $\lambda(R)$.

According to our definition, a stratified guarded update, may simultaneously substitute predicates at the same level only. We thus might wonder whether this restriction could be lifted. For FO transition systems with this extension, however, termination can no longer be guaranteed.

Lemma 2 *There exists a FO transition system $\mathcal{T}$ without assumptions, using stratified strictly guarded updates and resets, with simultaneous substitutions of predicates at different levels, together with some universal invariant Ψ such that for each $h \geq 0$, $\Psi^{(h)}$ is universal FO definable, but $\Psi^{(h)}[u] \not\Rightarrow \Psi^{(h+1)}[u]$ for some program point u.*

Proof Consider the FO transition system $\mathcal{T}$ as shown in Fig. 2 for some binary predicate E, together with the invariant $\Psi = \{0 \mapsto \top, 1 \mapsto \top, 2 \mapsto \mathsf{error} \vee \neg\mathsf{hull}(a,b), 3 \mapsto \top\}$ for constants a, b. Initially, the predicate hull is set to E. By executing the loop k times, either the error flag error is set to $\top$, or hull receives at most $(k+1)$fold compositions of E. Still, we can assign levels to the predicates used by $\mathcal{T}$ which meet the requirements of a stratification, namely,

$$\lambda = \{E \mapsto 0, \mathsf{add} \mapsto 0, \mathsf{hull} \mapsto 1, \mathsf{error} \mapsto 2\}$$

For the required SO quantifier elimination of A_1, A_2, we note that in order to avoid error to be set to $\top$, $\mathsf{add}(x,y,z)$ must imply $\mathsf{hull}(x,y) \wedge E(y,z)$. In order to falsify the invariant at program point 1 whenever possible, thus, $A_1(x,y,z)$ should be set to $\mathsf{hull}(x,y) \wedge E(y,z)$, and $A_2(x,z,y)$ at least to $\mathsf{add}(x,y,z)$. For the iterates $\Psi^{(h)}$ at program point 2, we therefore obtain

$$\begin{array}{rcl}
\Psi^{(0)}[2] & = & \mathsf{error} \vee \neg\mathsf{hull}(a,b) \\
\Psi^{(1)}[2] & = & \Psi^{(0)}[2] \wedge \forall y_1. \\
 & & (\mathsf{error} \vee \neg\mathsf{hull}(a,y_1) \vee \neg E(y_1,b)) \wedge \\
 & & (\mathsf{error} \vee \neg\mathsf{add}(a,y_1,b)) \\
\Psi^{(h)}[2] & = & \Psi^{(h-1)}[2] \;\wedge \forall y_h \dots y_1. \\
 & & (\mathsf{error} \vee \neg\mathsf{hull}(a,y_h) \vee \bigvee_{j=1}^{h-1} \neg E(y_{j+1},y_j) \vee \neg E(y_1,b)) \wedge \\
 & & (\mathsf{error} \vee \neg\mathsf{add}(a,y_h,y_{h-1}) \vee \bigvee_{j=1}^{h-2} \neg E(y_{j+1},y_j) \vee \neg E(y_1,b)) \quad (h \geq 2)
\end{array}$$

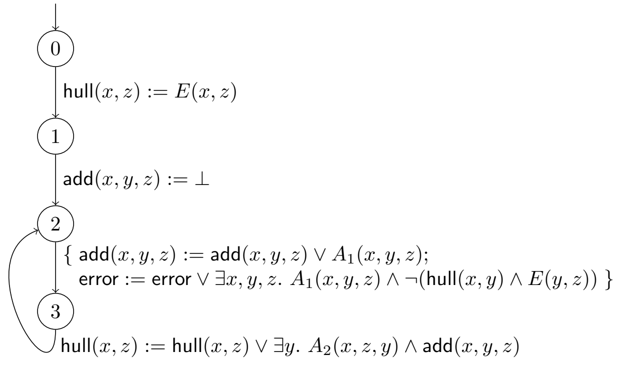

0
hull(x, z) := E(x, z)
1
add(x, y, z) := ⊥
2
{ add(x, y, z) := add(x, y, z) ∨ A₁(x, y, z);
error := error ∨ ∃x, y, z. A₁(x, y, z) ∧ ¬(hull(x, y) ∧ E(y, z)) }
3
hull(x, z) := hull(x, z) ∨ ∃y. A₂(x, z, y) ∧ add(x, y, z)

Fig. 2 FO transition system capturing transitive closure

Altogether, the weakest inductive invariant for program point 0 is given by $\mathsf{error} \vee \neg E^+(a, b)$ where E^+ is the transitive closure of E. As the transitive closure of a binary relation is not FO definable, we conclude that the fixpoint iteration cannot terminate. □

Thus, the crucial issue which results in inexpressible weakest inductive invariants, is the use of the *same* input predicate in the simultaneous update of predicates at different level. In the next section, we indicate how to generally deal with SO quantifiers, once a guarded update has been applied.

4 Universal SO quantifier elimination

It is well-known that universal SO quantifiers can be removed from otherwise quantifier-free formulas [3, 23]. For example,

$$\forall A.\, R\bar{x} \vee A\bar{y} \vee \neg A\bar{z} \quad \longleftrightarrow \quad R\bar{x} \vee (\bar{y} = \bar{z})$$

where for $\bar{y} = (y_1, \dots, y_k)$ and $\bar{z} = (z_1, \dots, z_k)$, $\bar{y} = \bar{z}$ is a shortcut for the formula $(y_1 = z_1) \wedge \dots \wedge (y_k = z_k)$. Interestingly, there are also cases where SO quantifier elimination is possible even in presence of FO existential quantifiers.

Example 8 Consider the substitution θ

$$R(y) \;:= R(y) \vee \exists z.\, A(y, z) \wedge S(y, z)$$

Then $\theta(R(a) \vee \neg R(b))$ is given by

$$\begin{array}{ll} & \forall z_1.\, R(a) \vee \exists z.\, A(a, z) \wedge S(a, z) \vee \neg R(b) \wedge (\neg A(b, z_1) \vee \neg S(b, z_1)) \\ \longleftrightarrow & \forall z_1.\, (R(a) \vee \exists z.\, A(a, z) \wedge S(a, z) \vee \neg R(b)) \wedge \\ & (R(a) \vee (\exists z.\, A(a, z) \wedge S(a, z)) \vee \neg A(b, z_1) \vee \neg S(b, z_1)) \end{array}$$

A closer inspection reveals that here SO quantifier elimination of A is possible where $\forall A.\, \theta(R(a) \vee \neg R(b))$ is equivalent to

$$\begin{array}{ll} & \forall z_1.\, (R(a) \vee \neg R(b)) \wedge (R(a) \vee (a = b) \wedge S(a, z_1) \vee \neg S(b, z_1)) \\ \longleftrightarrow & \forall z_1.\, (R(a) \vee \neg R(b)) \wedge (R(a) \vee (a = b) \wedge S(b, z_1) \vee \neg S(b, z_1)) \\ \longleftrightarrow & \forall z_1.\, (R(a) \vee \neg R(b)) \wedge (R(a) \vee (a = b) \vee \neg S(b, z_1)) \\ \longleftrightarrow & \forall z_1.\, R(a) \vee \neg R(b) \wedge ((a = b) \vee \neg S(b, z_1)) \end{array}$$

In particular, the resulting FO formula has universal FO quantifiers only. □

The observation in example 8 can be generalized.

Lemma 3 *Assume that A is a predicate of arity r and $\bar{y} = (q_1, \dots, q_r)$ is a sequence of distinct variables from $\mathcal{Y}$.*

1. *Assume that FO formula Ψ is of the form*

$$\bigvee_{i=1}^{m} (\exists \bar{z}_i .\, A\bar{a}_i\bar{z}_i \wedge \varphi_i[\bar{a}_i\bar{z}_i/\bar{y}]) \vee \bigvee_{j=m+1}^{n} (\forall \bar{z}_j .\neg A\bar{b}_j\bar{z}_j \vee \varphi_j) \tag{8}$$

for $m \leq n \in \mathbb{N}$ and fresh sequences of FO variables $\bar{z}_i$ where φ_i are FO formulas without occurrences of A. Then $\forall A.\ \Psi$ is equivalent to

$$\bigvee_{j=m+1}^{n} \forall \bar{z}_j .\ \bigvee_{i=1}^{m} (\bar{a}_i = \bar{b}_j\bar{z}_j) \wedge \varphi_i[\bar{b}_j\bar{z}_j/\bar{y}] \vee \varphi_j \tag{9}$$

Here, juxtaposition is meant to denote the concatenation of the corresponding sequences of FO variables, while an equality $\bar{x}_i = \bar{x}'_j$ for sequences $\bar{x}_i, \bar{x}'_j$ with $|\bar{x}_i| \leq |\bar{x}'_j|$ denotes the conjunction of equalities between the variables in $\bar{x}_i$ and the variables from the prefix of $\bar{x}'_j$ of the length $|\bar{x}_i|$.

2. *If* Ψ *is of the form*

$$\varphi_0 \vee \bigvee_{i=1}^{m} (\exists \bar{z}_i .\, A\bar{a}_i\bar{z}_i \wedge \varphi_i[\bar{a}_i\bar{z}_i/\bar{y}]) \vee \bigvee_{j=m+1}^{n} \psi_j \wedge \forall \bar{z}_j .\neg A\bar{b}_j\bar{z}_j \vee \varphi_j \tag{10}$$

for $m \leq n \in \mathbb{N}$ where $\varphi, \varphi_i, \psi_j$ all are FO formulas without occurrences of A. Then $\forall A.\ \Psi$ is equivalent to

$$\varphi_0 \vee \bigvee_{j=m+1}^{n} \psi_j \wedge \forall \bar{z}_j .\ \bigvee_{i=1}^{m} (\bar{a}_i = \bar{b}_j\bar{z}_j \wedge \varphi_i[\bar{b}_j\bar{z}_j/\bar{y}] \vee \varphi_j \tag{11}$$

In case that all formulas $\varphi_i, i = 1, \ldots, m$ are all equivalent to φ' and and for $j = m+1, \ldots, n$, $\varphi_j = \neg\varphi'[\bar{b}_j\bar{z}_j/\bar{y}]$ where all lists $\bar{a}_i, \bar{b}_j$ have the same length, then the formulas (9) and (11) after second-order quantifier elimination of A, can be simplified to:

$$\bigvee_{j=m+1}^{n} \forall \bar{z}_j .\ \bigvee_{i=1}^{m} (\bar{a}_i = \bar{b}_j) \vee \neg\varphi'[\bar{b}_j\bar{z}_j/\bar{y}] \tag{12}$$

and
$$\varphi_0 \vee \bigvee_{j=m+1}^{n} \psi_j \wedge \forall \bar{z}_j .\ \bigvee_{i=1}^{m} (\bar{a}_i = \bar{b}_j) \vee \neg\varphi'[\bar{b}_j\bar{z}_j/\bar{y}] \tag{13}$$

Proof We apply Ackermann's lemma in negated form. According to the form provided in [24], it states that

$$\forall A.\ (\exists \bar{y}.\, A\bar{y} \wedge \varphi) \vee \psi \qquad \longleftrightarrow \qquad \psi[\neg\varphi/A] \tag{14}$$

if φ contains no occurrence of A, and ψ only contains negative occurrences of A.

Then the equivalence for (8) follows by choosing

$$\begin{array}{lcl} \bar{y} & \equiv & (y_1, \ldots, y_r) \\ \varphi & \equiv & \bigvee_{i=1}^{m} (\bar{a}_i = \bar{y}_i) \wedge \varphi_i[\bar{a}_i/\bar{y}_i] \\ \psi & \equiv & \bigvee_{j=m+1}^{n} (\forall \bar{z}_j .\neg A\bar{b}_j\bar{z}_j \vee \varphi_j) \end{array}$$

where for $i = 1, \ldots, m$, $\bar{y}_i$ is the prefix of $\bar{y}$ of length $|\bar{a}_i|$, while $\bar{z}_i$ is considered as the corresponding suffix, i.e., $\bar{y} = \bar{y}_i\bar{z}_i$. The equivalence for (10) is obtained by using the same definition for φ, but replacing the formula ψ with

$$\varphi_0 \vee \bigvee_{j=m+1}^{n} \psi_j \wedge (\forall \bar{z}_j.\neg A\bar{b}_j\bar{z}_j \vee \varphi_j)$$

□

Example 9 Consider the second update in the loop of the transition system from Fig. 1.

$\mathsf{discuss}(x_1, x_2, p, d)$: $=$ $\mathsf{discuss}(x_1, x_2, p, d)$ $\vee$ $\exists r_1, r_2. A_4(x_2, x_1, p, d, r_1, r_2)$ $\wedge$ $\mathsf{report}(x_1, p, r_1)$ $\wedge$ $\mathsf{report}(x_2, p, r_2)$

Let θ_4 denote this update, and consider the invariant (2) from the introduction. Application of θ_4 results in the formula

$$\begin{aligned}&\forall x_1, x_2, p, d, r_1, r_2.\neg\mathsf{discuss}(x_1, x_2, p, d) \wedge\\ &\quad(\neg A_4(x_1, x_2, p, d, r_1, r_2) \vee \neg\mathsf{report}(x_1, p, r_1) \vee \neg\mathsf{report}(x_2, p, r_2)) \vee\\ &\quad(\neg\mathsf{auth}(x_1, p) \wedge \neg\mathsf{auth}(x_2, p))\end{aligned}$$

Since A_4 only occurs negatively, universal SO quantifier elimination of A_4 yields

$$\begin{aligned}&\forall x_1, x_2, p, d, r_1, r_2.\neg\mathsf{discuss}(x_1, x_2, p, d) \wedge\\ &\quad(\neg\mathsf{report}(x_1, p, r_1) \vee \neg\mathsf{report}(x_2, p, r_2)) \vee\\ &\quad(\neg\mathsf{auth}(x_1, p) \wedge \neg\mathsf{auth}(x_2, p))\end{aligned}$$

□

As an important consequence of lemma 3, we obtain:

Theorem 4 *Assume that $\mathcal{T}$ is a FO transition systems with guarded updates and resets only, and assumptions g of the form*

$$\exists \bar{z}_1.\varphi_0 \wedge (\forall \bar{z}_2.\neg A\bar{z}_1\bar{z}_2 \vee \varphi_1) \wedge (\varphi_2 \vee \exists \bar{z}_3.A\bar{z}_1\bar{z}_3 \wedge \varphi_3)$$

where $\varphi_0, \varphi_1, \varphi_2, \varphi_3$ are quantifierfree formulas not containing occurrences of the SO variable A. Let Ψ a universal FO invariant. Then the following holds:

1. *The iterates $\Psi^{(h)}[u], h \geq 0$, in (5) all are effectively equivalent to universal FO formulas.*
2. *The iteration terminates, i.e., $\Psi^{(h)} = \Psi^{(h+1)}$ for some $h \geq 0$, iff for each program point u, the weakest strengthening of all iterates $\Psi^{(h)}[u]$ is FO-definable.*

Proof Due to lemma 3, for each universal FO formula ψ, each guard g of appropriate form, and each guarded update or reset θ with input predicate A, $\forall A. (\neg g \vee \theta\psi)$ is equivalent to a universal FO formula. That implies statement (1). Now assume for for each $h \geq 0$ and each $v \in V$, $\Phi^{(h)}[v]$ is FO definable. Then due to the compactness theorem for FO predicate logic [25], there is some $h \geq 0$ such that $\Psi^{(h)}[v] \leftrightarrow \Psi^{(h+j)}[v]$ holds for all $v \in V$ and $j \geq 0$, iff for each $v \in V$, the conjunction $\bigwedge_{h\geq 0} \Psi^{(h)}[v]$ is again FO definable. □

Example 10 Consider again the specification from Fig. 1, and let $\theta_1, \theta_2, \theta_3$, and θ_4 denote the substitutions occurring therein.

Assume that Ψ equals the universal formula in (2), and we are interested in its validity at program point 2 of the transition system. The formula $\forall A_3.\, \theta_3(\forall A_4.\, \theta_4(\Psi))$ is given by

$$\begin{array}{ll} & \forall A_3.\, \theta_3(\forall x_1, x_2, p, d, r_1, r_2.\neg\mathsf{discuss}(x_1, x_2, p, d) \wedge \\ & \quad (\neg\mathsf{report}(x_1, p, r_1) \vee \neg\mathsf{report}(x_2, p, r_2)) \vee (\neg\mathsf{auth}(x_1, p) \wedge \neg\mathsf{auth}(x_2, p)) \\ \longleftrightarrow & \forall x_1, x_2, p, d, r_1, r_2.\neg\mathsf{discuss}(x_1, x_2, p, d) \wedge \\ & \quad (\neg\mathsf{report}(x_1, p, r_1) \wedge \neg\mathsf{assign}(x_1, p) \vee \neg\mathsf{report}(x_2, p, r_2) \wedge \neg\mathsf{assign}(x_2, p)) \vee \\ & \quad (\neg\mathsf{auth}(x_1, p) \wedge \neg\mathsf{auth}(x_2, p)) \end{array}$$

The resulting formula Ψ' already equals the fixpoint for the loop.

Since the predicate assign only occurs negatively in Ψ' and conf only negatively in the right-hand side for assign, the formula $\forall A_1.\theta_1(\forall A_2.\theta_2(\Phi'))$ is constructed from Ψ' via the substitution θ_{assign} defined by

$$\mathsf{assign}(y_1, y_2) := \mathsf{assign}(y_1, y_2) \vee \neg(\mathsf{conf}(y_1, y_2) \vee \mathsf{auth}(y_1, y_2))$$

This means the formula Ψ'' for the initial node 0 of the transition system is given by

$$\begin{array}{l} \forall x_1, x_2, p, d, r_1, r_2.\neg\mathsf{discuss}(x_1, x_2, p, d) \wedge \\ \quad (\neg\mathsf{report}(x_1, p, r_1) \wedge \neg\mathsf{assign}(x_1, p) \wedge (\mathsf{conf}(x_1, p) \vee \mathsf{auth}(x_1, p)) \vee \\ \quad \neg\mathsf{report}(x_2, p, r_2) \wedge \neg\mathsf{assign}(x_2, p) \wedge (\mathsf{conf}(x_2, p) \vee \mathsf{auth}(x_2, p))) \vee \\ \quad (\neg\mathsf{auth}(x_1, p) \wedge \neg\mathsf{auth}(x_2, p)) \end{array}$$

By the initial condition $\mathcal{H}$ from the introduction, $\neg\mathsf{discuss}(x_1, x_2, p, d)$ generally holds at node 0 of the transition system, as well as $\neg\mathsf{report}(x_i, p, r_i)$, $\neg\mathsf{assign}(x_i, p)$, and $\neg\mathsf{conf}(x_i, p)$ for $i = 1, 2$. Therefore, $\mathcal{H}$ implies Ψ'', and the property Ψ at node 3 of the transition system is valid. □

In this section we have shown comprehensively how to eliminate universal SO quantifiers introduced by guarded updates or resets in a FO transition system. In the next two sections, we will apply these results to FO transition systems which additionally are stratified.

5 Strictly guarded stratified updates

In this section, we consider FO transition systems without guards with stratified guarded updates and resets, where all guarded updates are *strict*. This means they all are of the form

$$\{\ R_j \bar{y}_{R_j} := \quad R_j \bar{y}_{R_j} \vee \exists \bar{z}_j.\, A\bar{y}_{R_j}^{\sigma_j} b z_j \wedge \psi_i \mid i = 1, \dots, r\} \tag{15}$$

for predicates $R_1, \dots, R_r$ all of the same level. Let us call such a FO transition system *strictly guarded and stratified.*

Theorem 5 *Assume we are given a strictly guarded and stratified FO transition system. Then for every universal invariant Ψ, the weakest inductive invariant strengthening Ψ can be represented by universal FO formulas, and can effectively be computed.*

Proof For this proof, it is convenient to use the notation $\Phi \ni \forall \bar{x}.\, c$ for a universal FO formula Φ, a clause c, and a list $\bar{x}$ of distinct variables so that for the prenex CNF

$\forall\bar{z}.\, c_1 \wedge \ldots \wedge c_m$ of Φ, c occurs among the c_j, and $\bar{x}$ is the subsequence of variables in $\bar{z}$ which occur in c. We rely on the following technical lemma.

Lemma 6 *Assume that c is a clause and θ a stratified reset or a stratified strictly guarded update with input predicate A which substitutes predicates of level s.*

Let c' be a clause with $\forall A.\, \theta(c) \ni \forall\bar{z}.\, c'$ where $\bar{z}$ is the list of newly introduced variables in c'. Then either $c = c'$ and $\bar{z}$ is empty, or the number of literals at level s of c' is less than the corresponding number of literals in c, while the set of literals at levels exceeding s stays the same.

Proof Assume that the clause c is of the form

$$c_0 \vee \bigvee_{i=1}^{m} R_i\bar{y}_i \vee \bigvee_{j=m+1}^{n} \neg R_j\bar{y}_j$$

where $R_i\bar{y}_i, R_j\bar{y}_j$ are the literals of c substituted by θ, while c_0 does not contain occurrences of left-hand sides of θ.

If θ is a reset, all literals containing R_i are eliminated from c.

Therefore, the assertion of the lemma holds. Now assume that θ is a strictly guarded update where $\theta(R_i\bar{y}_{R_i}) = R_i\bar{y}_{R_i} \vee \exists\bar{z}_i.A\bar{y}_{R_i}^{\sigma_i}\bar{z}_i \wedge \psi_i$ for some permutation σ_i. Then by lemma 3,

$$\begin{aligned}
\forall A.\, \theta(c) \;&\longleftrightarrow\; c_0 \vee \textstyle\bigvee_{i=1}^{m} R_i\bar{y}_i \vee \bigvee_{j=m+1}^{n} \neg R_j\bar{y}_j \wedge \\
&\;\; \forall\bar{z}_j.(\textstyle\bigvee_{i=1}^{m}(\bar{y}_i^{\sigma_i} = \bar{y}_j^{\sigma_j}\bar{z}_j) \wedge \psi_i[\bar{y}_j^{\sigma_j}\bar{z}_j/\bar{y}_{R_i}^{\sigma_i}\bar{z}_i] \vee \neg\psi_j[\bar{y}_j/\bar{y}_{R_j}]) \\
&\longleftrightarrow\; \textstyle\bigwedge_{J\subseteq[m+1,n]} \forall\bar{z}_J.\, (c_0 \vee \bigvee_{i=1}^{m} R_i\bar{y}_i \vee \bigvee_{j\notin J} \neg R_j\bar{y}_j \vee \\
&\;\; \textstyle\bigvee_{j\in J} \bigvee_{i=1}^{m}(\bar{y}_i^{\sigma_i} = \bar{y}_j^{\sigma_j}\bar{z}_j) \wedge \psi_i[\bar{y}_j^{\sigma_j}\bar{z}_j/\bar{y}_{R_i}^{\sigma_i}\bar{z}_i] \vee \neg\psi_j'[\bar{y}_j/\bar{y}_{R_j}])
\end{aligned}$$

where $\bar{z}_j$ is a fresh list of FO variables of the same length as $\bar{z}$, and $\bar{z}_J$ is the concatenation of all lists $\bar{z}_j, j \in J$.

In particular for $J = \emptyset$, $\bar{z}_J$ is empty and the corresponding clause equals c. If on the other hand $J \neq \emptyset$, the number of negated literals occurring in the clause has decreased. □

By lemma 6, the number of literals at level s therefore either decreases, or the clause stays the same.

Let Θ denote a finite set of stratified guarded substitutions where all updates in Θ are strictly guarded, and let c_0 denote any clause.

Consider a sequence $(\theta_t, \forall\bar{x}_t.c_t), t \geq 1$, where for all $t \geq 1$, $\theta_t \in \Theta$ with some input predicate A_t, and $\forall A_t.\, (\theta_t c_{t-1}) \ni \forall\bar{x}_t.\, c_t$ holds.

We claim that then there is some $t' \geq 1$ so that $c_{t'} = c_{t''}$ and $\bar{x}_{t''}$ is empty for all $t'' > t'$.

In order to prove that claim, we introduce for $t \geq 1$, the vector $v_t = (v_{t,L}, \ldots, v_{t,1}) \in \mathbb{N}^L$ where L is the maximal level of a predicate in $\mathcal{R}_{state}$, and $v_{t,i}$ is the number of literals with predicates of level i.

By lemma 6, it holds for all $t \geq 0$, that either $c_t = c_{t+1}$ and $\bar{z}_t$ is empty, or $v_t > v_{t+1}$ w.r.t. the lexicographic order on $\mathbb{N}^L$. Since the lexicographical ordering on $\mathbb{N}^L$ is well-founded, the claim follows.

We conclude that the set of quantified clauses $\forall\bar{z}.c$ with $\Psi^{(h)}[u] \ni \forall\bar{z}.c$ for any u and h, is finite. From that, the statement of the theorem follows. □

Theorem 5 leaves open the case of transition systems with stratified resets and stratified guarded updates of which some are not strictly guarded. To these, the presented proof technique cannot be easily extended. The reason is that a non-strictly guarded update θ for some predicate R, when applied to some clause c, may result in a quantified clause $\forall\bar{z}.\, c'$ with $\forall A.\theta(c) \ni \forall\bar{z}.\, c'$ so that neither $c = c'$ holds nor does the number of literals $\neg R\bar{b}$ decrease.

6 Positive guarded and stratified updates

Let us consider another case where termination can be guaranteed in presence of updates as well as resets. We call an update or reset θ *positive* if all predicates only occur positively in the right-hand sides of θ. Let us call a FO transition system *positive* if it uses no assumptions and only positive stratified guarded updates and resets at all levels at least 1.

Theorem 7 *Assume that $\mathcal{T}$ is a positive FO transition system where all updates are single. Then for every universal invariant Ψ, the weakest inductive invariant implying Ψ is again universal and can effectively be computed.*

Proof Let Θ denote the substitutions occurring at edges in $\mathcal{T}$. For $\theta \in \Theta$, let $[\![\theta]\!]$ denote the transformation of universal FO formulas which maps every universal FO formula ψ to a universal FO formula equivalent to $\forall A.\theta(\psi)$, if A is the input predicate corresponding to θ. Let $\pi = \theta_L \dots \theta_1$ denote a sequence of substitutions from Θ. Then $[\![\pi]\!] = [\![\theta_N]\!] \circ \dots \circ [\![\theta_1]\!]$ is the composition of the transformations corresponding to the substitutions occurring in π. Now consider a quantifierfree formula ψ in conjunctive normal form with predicates from $\mathcal{R}_{state}$ and FO variables from X. Consider some $\theta \in \Theta$. Since θ is a stratified reset or a stratified guarded update which is single and positive, $[\![\theta]\!](\psi)$ can be chosen in such a way that all occurring argument lists $\bar{b}$ of positive literals $R\bar{b}$, $R \in \mathcal{R}$, only use variables from X. Since θ is single, SO quantifier elimination of the input predicate A of θ can be realized by a single substitution of the form

$$A\bar{y}\bar{z} := \bigwedge_{i=1}^{n} (\bar{y} \neq \bar{a}_i)$$

for suitable $n \geq 0$ and sequences $\bar{a}_i$ of FO variables from X. Let us call the right-hand side here a *worst attacker*. This means that, for all levels $t = L, \dots, 1$, there are only finitely many possibilities of *worst attackers* to substitute the occurring input predicates. Let L denote the maximal level of predicates from $\mathcal{R}_{state}$. Moreover for $t = L, \dots, 1$, let $\bar{z}_t$ denote one distinct enumeration of variables occurring in substitutions in Θ whose right-hand sides are of level t. The formulas $[\![\pi_N]\!](\psi)$ thus can all be represented as finite conjunctions g of generalized clauses $c^{(L)}$ according to the following grammar:

$$\begin{array}{lcl} g & ::= & c_1 \wedge \dots \wedge c_r \\ c & ::= & c_0 \vee c^{(L)} \\ c^{(t)} & ::= & \bot \mid c^{(t)} \vee \neg R\bar{b} \mid c^{(t)} \vee e_t \mid c^{(t)} \vee o^{(t)} \\ o^{(t)} & ::= & (\bigvee_{i=1}^{n} (\bar{b} = \bar{a}_i) \vee \forall\bar{z}_t.c^{(t-1)}) \end{array}$$

where c_0 is a disjunction of literals without negative literals $\neg R\bar{b}$ using FO variables from X; while for $0 \leq t \leq L$, the clause $c^{(t)}$ satisfies the following side constraints:

- For every occurring negative literal $\neg R\bar{b}$, $\lambda(R) \leq t$ holds and all variables in $\bar{b}$ are contained in X or occur in $\bar{z}_L \ldots \bar{z}_{t+1}$;
- e_t ranges over equalities or disequalities between FO variables from X or $\bar{z}_L \ldots \bar{z}_{t+1}$.

Moreover, for $o^{(t)}$, all FO variables occurring in the lists $\bar{a}_i$ are from X only, while those occurring in $\bar{b}$ are either from X or occur in $\bar{z}_L \ldots \bar{z}_{t-1}$. By induction on t, we verify that for each level $t \geq 0$, the number of non-equivalent formulas $o^{(t)}$ and thus also the number of non-equivalent formulas $c^{(t)}$ is finite. We conclude that also the number on non-equivalent formulas c as well as the number of non-equivalent formulas g is finite.

Accordingly, the number of non-equivalent formulas $\forall A_1 \ldots A_N.\, \pi(\psi)$ is finite — implying that for every universal invariant Ψ, $\Psi^{(h+1)} = \Psi^{(h)}$ for some $h \geq 0$. From that, the statement of the theorem follows. □

The proof argument for theorem 7 cannot easily be extended to unrestricted stratified guarded substitutions. In presence of *negated* literals in substitutions or updates which are not single, the arguments of positive literals $R\bar{a}$ occurring in $[\![\pi]\!](\psi)$ need not necessariliy have already occurred in ψ. For the next result, we therefore have to rely on a different proof strategy.

7 Stratified guarded updates

Let us finally consider FO transition systems without assumptions where all occurring substitituions are resets or guarded updates which are stratified. Let us call such a FO transition system *stratified guarded*. In [3], termination was announced for stratified guarded FO transition systems without resets, where additionally instantiation of existential quantifiers was applied as an *abstraction* to enforce all occurring formulas to be universal. Here, we improve on that result in two respects. First, we present a proof that termination can also be guaranteed without any abstraction. Second, we generalize the setting by allowing stratified resets — at least at the maximal and minimal levels.

Theorem 8 *Assume that $\mathcal{T}$ is a stratified guarded FO transition system where resets only occur for predicates of level* 1 *and the maximal level L. Then for every universal invariant Ψ, the weakest inductive invariant is again universal and can effectively be computed.*

Proof We show that there is some $h \geq 0$, so that $\Psi^{(h+1)} = \Psi^{(h)}$. Since by lemma 3, $\Psi^{(h)}[u]$ is a universal formula for all $h \geq 0$ and program points u, the statement of the theorem follows.

Let Θ denote the finite set of stratified guarded substitutions occurring in $\mathcal{T}$. W.l.o.g., we assume that each substitution in Θ simultaneously substitutes all predicates at a given level t. Let ψ a quantifierfree FO formula in negation normal form. Let $\pi = \theta_N, \ldots, \theta_1$ be any sequence of substitutions where for each $i = 1, \ldots, N$, $\theta_i = \theta'[A_i/A_{e_i}]$ holds for a fresh input predicate A_i, and some substitution $\theta' \in \Theta$. Thereby, let $\lambda(A_i)$ denote the level of left-hand sides of θ'. Termination of fixpoint iteration is based on the following lemma. □

Lemma 9 *There is a fixed finite sequence $\mathbf{z}$ of variables only depending on ψ and the substitutions from $\mathcal{T}$ so that $\forall A_N \ldots A_1.\pi(\psi) = \forall \mathbf{z}.\psi'$ for some quantifierfree FO formula ψ'. In particular, $\mathbf{z}$ can be chosen independently of the length N of π.*

Since the number of quantifierfree FO formulas with a given fixed finite set of FO variables is finite, lemma 9 implies that the number of non-equivalent universal FO formulas possibly occurring as $\forall A_N \dots A_1.\pi(\psi)$ for every universally quantified FO formula ψ, and thus also the number of non-equivalent conjunctions of these formulas is finite. Accordingly, there must be some $h \geq 0$ so that in (5), $\Psi^{(h+1)} = \Psi^{(h)}$, and the theorem follows.

□

It therefore remains to prove lemma 9.

Proof of lemma 9 Let us first consider the case where there is no reset of predicates. Let $\mathcal{R}_{state}$ denote the finite set of predicate symbols used by $\mathcal{T}$. Let X and Z denote the finite sets of variables occurring in ψ and introduced by substitutions from Θ, respectively. Let L denote the maximal level of a predicate in $\mathcal{R}$. For $1 \leq t \leq L$, let V_t denote the set of all variables

$$z_{R,\bar{b}} \tag{16}$$

with $z \in Z$, $\lambda(R) = t$, and $\bar{b}$ a sequence of variables from $X \cup V_L \cup \ldots \cup V_{t+1}$ whose length equals the arity of R. We then will use the set $V = V_1 \cup \ldots \cup V_L$ as our set of bound variables. According to the definition of the V_t, the set V is finite. By induction on the length N of π, we construct a formula in a particular normal form ψ_N which is equivalent to $\pi(\psi)$. In this construction, we make sure that all variables bound by existential or universal quantifiers are always taken from V. The normal form g of formulas we rely on, consists of a finite conjunction of *generalized* clauses c which are built up according to the following abstract grammar

$$\begin{array}{lclclcl} g & ::= & \top & | & c \wedge g & & \\ c & ::= & c_0 & | & c \vee \exists \bar{z}.A'\bar{a}\bar{z} \wedge g' & | & c \vee f_{R,\bar{b}} \\ f_{R,\bar{b}} & ::= & \neg R\bar{b} \wedge \forall \bar{z}_R.\ \bigwedge_{n=1}^{r} (\neg A_n \bar{b}^{\sigma_n} \bar{z}_n \vee c_n) & & & & \end{array}$$

In the second line, c_0 is an ordinary clause without occurrences of input predicates, A' is an input predicate where all predicates occurring in g' have levels less than $\lambda(A')$. In the third line, R is a predicate, $\bar{b}$ are sequences of arguments, $\bar{z}_R$ is a sequence of FO variables whose length only depends on R and contains all FO variables required by substitutions of R, A_n are input predicates of levels $\lambda(R)$, and all predicates occurring in any of the c_n have levels less than $\lambda(R)$. A formula $f_{R,\bar{b}}$ is also called *negation* tree with head $\neg R\bar{b}$. $f_{R,\bar{b}}$ is called *non-trivial*, if it contains occurrences of input predicates, i.e., the conjunction in the second conjunct is non-empty. Thereby, we maintain the invariant that on the toplevel, i.e., outside the scopes of all quantifiers, the heads of all negation trees are distinct.

This normal form can be obtained for ψ by constructing the conjunctive normal form of ψ and thereby, removing all duplicates of literals in clauses. Let ψ_0 denote the resulting conjunction for ψ. Now assume that we have already constructed the normal form ψ_{N-1}. By using distributivity of $\wedge$, the corresponding normal form for $\psi_N = \theta_N(\psi_{N-1})$ can be readily computed. For maintaining the invariant on negation trees, we recall that $\neg R\bar{b} \vee \neg R\bar{b} \wedge g \longleftrightarrow \neg R\bar{b}$, i.e., newly created literals $\neg R\bar{b}$ in clauses *kill* already existing negation trees with this head.

Given the normal form ψ_N for $\pi(\psi)$, we now successively apply SO quantifier elimination. We proceed from input predicates A_i of higher levels downwards towards the input predicates of smaller levels. For each clause, we thereby collect the lists of FO variables $\bar{z}_t$ required to be universally quantified and prove that these lists possibly introduced for level t, consist of variables from V_t only. In the end, we thus arrive at a formula $\forall \mathbf{z}_1 \dots \mathbf{z}_L.\psi'$

where $\mathbf{z}_t$ is the sequence of variables in V_t, and ψ' is a quantifierfree conjunction of (plain) clauses with variables from $X \cup V$ only. Since there is only a finite set of of such clauses, the theorem follows.

So, let τ denote a permutation of $\{1, \ldots, N\}$ so that $\lambda(A_{\tau(N)}) \leq \ldots \leq \lambda(A_{\tau(1)})$ holds. In particular, $\lambda(A_{\tau(1)})$ and $\lambda(A_{\pi(N)})$ denote input predicates of highest and lowest occurring levels, respectively. Then we successively remove the occurrences of $A_{\tau(1)}, A_{\tau(2)}, \ldots$ by maintaining the normal form of the resulting formulas — while introducing fresh bound variable names just from the set V. Let $\psi'_{L+1} = \psi_N$. Now we proceed level by level. for $L \geq t > 0$, assume that $\forall \mathbf{z}_{t+1} \ldots \mathbf{z}_L . \psi'_{t+1}$ has already been constructed such that ψ'_{t+1} is in our normal form where the heads of all non-trivial negation trees on top-level are distinct and have levels at most t. Let us consider a single generalized clause c of ψ'_{t+1}. Let $f_{R_1, \bar{b}_1}, \ldots, f_{R_m \bar{b}_m}$ denote the sequence of top-level negation trees in ψ'_{t+1} of level t in c.

If this sequence is empty, then we set $\mathbf{z}_t = \epsilon$ (the empty sequence) and ψ'_t as the formula obtained from ψ'_{t+1} by removing all occurrences of subformulas $\exists \bar{z}_R . A' \bar{b} \bar{z} \wedge \psi'$ with $\lambda(A') = t$.

Otherwise, let $A_{\tau(j_1)} \ldots A_{\tau(j_2)}$ denote the subsequence of input predicates of level t occurring in c. Then for each $j = 1, \ldots, m$, the negation tree $f_{R_j, \bar{b}_j}$ is of the form

$$\neg R_j \bar{b}_j \wedge \forall \bar{z}_{R_j} . \bigwedge_{k=1}^{n_j} \neg A'_k \bar{b}_j^{\sigma_k} \bar{z}_{jk} \vee c_{jk} \tag{17}$$

for suitable input predicates A'_k of level t, subsequences $\bar{z}_{jk}$ of $\bar{z}_{R_j}$ and (generalized) clauses c_{jk} containing predicates only of levels less than t. Since the heads of the negation trees $f_{R_j, \bar{b}_j}$ are all distinct, we can rename the corresponding universally quantified variables to distinct sequences $\bar{z}_{R_j, \bar{b}_j} = z_{(R_j, \bar{b}_j),1} \cdots z_{(R_j, \bar{b}_j), l_j}$ if l_j is the length of the corresponding list of bound variables and let the sequences $\bar{z}_{\bar{b}_j, jk}$ be appropriate subsequences of this list for the $\bar{z}_{jk}$. By inductive hypothesis for variables in $\bar{b}_j$ and the definition of the set V_t, all these variables are contained in V_t. Therefore, c is equivalent to the formula $\forall \mathbf{z}_t . c'$ where c' coincides with c up the negation trees of level t, which now take the form

$$\neg R_j \bar{b}_j \wedge \bigwedge_{k=1}^{n_j} \neg A'_k \bar{b}_j^{\sigma_k} \bar{z}_{\bar{b}_j, jk} \vee c_{jk}[\bar{z}_{R_j, \bar{b}_j} / \bar{z}_{R_j}] \tag{18}$$

($j = 1, \ldots, m$). Performing SO quantifier elimination of all input predicates $A_{\tau(j_1)}, \ldots, A_{\tau(j_2)}$ in a row removes all subformulas $\exists \bar{z}_{R'} . A' \bar{a} \bar{z}_{R'} \wedge \psi'$ where $\lambda(A') = t$ from c', and additionally replaces the subformulas (18) with formulas

$$\neg R_j \bar{b}_j \wedge \bigwedge_{k=1}^{n_j} \bigvee_{i=1}^{n_j} (\bar{a}_{ji}^{\sigma_{ji}} = \bar{b}_j^{\sigma_k}) \wedge g_{ji}[\bar{b}_j^{\sigma_k} \bar{z}_{\bar{b}_j, jk} / \bar{y}_{ji}^{\sigma_{ji}} \bar{z}_{ji}] \vee c_{jk}[\bar{z}_{R_j, \bar{b}_j} / \bar{z}_{R_j}] \tag{19}$$

for suitable conjunctions of (generalized) clauses g_{ji} in with constants from X and free variables from $\bar{z}_R$ and some list of parameters $\bar{y}$ for which a subformula $\exists \bar{z}_{ji} . A'_k \bar{a}_{ji}^{\sigma_{ji}} \bar{z}_{ji} \wedge g_{ji}[\bar{a}_{ji} / \bar{y}]$ has occurred in c'. In particular, each predicate occurring in any of the g_{ji} is of level less than t. By distributivity, the resulting formula is equivalent to a conjunction g' of (generalized) clauses c'' each of which is obtained from c' by replacing each of the subformulas (18) with $\neg R_j \bar{b}_j$, or, for some k and some subset $I \subseteq \{1, \ldots, n_j\}$, with the clause

$$\bigvee_{i\in I}(\bar{a}_{ji}^{\sigma_{ji}} = \bar{b}_j^{\sigma_k}) \vee \bigvee_{i\notin I} c'_{ji}[\bar{b}_j^{\sigma_k}\bar{z}_{\bar{b}_j,ik}/\bar{y}_{ji}^{\sigma_{ji}}\bar{z}_{ji}] \vee c_{jk}[\bar{z}_{R_j,\bar{b}_j}/\bar{z}_{R_j}] \tag{20}$$

where c'_{ji} is any clause of g_{ji}. The clauses from g' constructed in this way, may contain negation trees which agree in their heads. Consider two such negation trees both with head $\neg R\bar{b}$ occurring in the same clause c'' of g' on top-level. If both originate from ψ or have been introduced by means of the same substitution, say, θ_j, the same sequence of substitutions has been applied to both, and subsequently also to their negation trees. Accordingly, the two negation trees are equivalent, meaning that one of them can be removed from c''. Therefore, now assume that one literal $\neg R\bar{b}$ has been introduced by substitution θ_j while the the other did already occur in ψ_{j-1}. That means that the sequence of substitutions applied to the later one and its negation tree, also is applied to the earlier one and its negation tree. In their disjunction (now introduced due to SO quantifier elimination), therefore, the negation tree of the earlier literal implies the negation tree of the later, and therefore can be omitted. Performing this normalization on the clauses of g', we achieve that in the resulting conjunction g'' of (generalized) clausesall heads of top-level negation trees are distinct. Then we set ψ'_t to g''.

In order to compute an explicit bound on the number of possible FO variables occurring as arguments to predicates of level $t = L, \dots, 0$, let us introduce the following structural parameters:

v — the number of variables occurring in ψ
L — the number of levels of predicates
r — maximal arity of a predicate
m — maximal number predicates at some level i
l — maximal length of $\bar{z}$in subformulas $\exists\bar{z}.\,\psi$occurring in the substitutions from Θ

For $t = L, \dots, 0$, we inductively determine a bound B_t to the number of distinct FO variables possibly occurring as arguments of literals at level t. Thereby, we set $B_L = v$, since the only literals at level L occurring in c' already must have occurred in ψ. Therefore, assume that $t < L$ and a bound B_{t+1} has already been found. Given the number B_{t+1}, the number of negated literals of predicates at level $t + 1$ can be bound by $m \cdot B^r_{t+1}$. For each of these literals, a fresh list of variables of length at most l may be provided. Accordingly, we set

$$B_t = B_{t+1} + l \cdot m \cdot B^r_{t+1} \leq (1 + l \cdot m) \cdot B^r_{t+1}$$

Altogether, this means that the total number B of variables possibly occurring in literals of c' at level at least 0 is bounded by

$$B \leq \begin{cases} (1 + l \cdot m)^L \cdot v & \text{if } r = 1 \\ (1 + l \cdot m)^{\frac{r^L-1}{r-1}} \cdot v^{r^L} & \text{if } r > 1 \end{cases} \tag{21}$$

It remains to consider the case when resets occur either on the highest level L or at level 1. We claim that with these kinds of resets, still the same set V for bound variables suffices to construct a FO universally quantified formula for $\forall A_1 \dots A_N.\pi(\psi)$. Let us first consider resets of predicates at top-level L. Each of these, either has no effect or replaces all occurrences of one predicate R with $\lambda(R) = L$ with a quantifierfree formula with occurrences of state predicates of lower levels and no input predicates. The only FO variables occurring in literals $R\bar{b}$ or $\neg R\bar{b}$ for predicates R of level L are the variables from X occurring already in ψ. Accordingly, we modify the first phase of constructing $\psi_1, \dots, \psi_N$ as follows. We put the list $\mathbf{z}_L$ as universally prenex in front of all ψ_j in order to have a reservoir of FO variables for all univerally quantified bound variables introduced at level L. As soon as an update substitution θ_i of a predicate R at level L occurs, we rename in each clause the universally quantified variables from $\bar{z}_R$ introduced for $\neg R\bar{b}$ to $\bar{z}_{R,\bar{b}}$ from the reservoir and *immediately* perform SO quantifier elimination for A_i. Then we bring the resulting formula into the format g to obtain ψ_i. With this preparation, a reset θ_j at level L can be dealt with during the construction of the sequence $\psi_1, \dots, \psi_N$ just by replacing literals at level L with appropriate clauses of literals with predicates of lower level all using variables from X as arguments only. In particular, no new variables are introduced. For the second phase, it then remains to perform SO quantifier elimination just for the levels $L-1, \dots, 1$.

Now additionally, consider resets of predicates at level 1. In principle, we proceed as before. The resets at level 1, however, may additionally introduce subformulas of the form $o_{\bar{b}}$ in clauses where

$$o_{\bar{b}} ::= \forall \bar{z}_R. \bigwedge_{k=1}^{n} (\neg A_k \bar{b}\bar{z} \vee c_k) \tag{22}$$

where $\bar{b}$ is a sequence of variables, A_k are input variables of level 1, and c_n contains predicates of level 0 only with arguments possibly from X, $\bar{b}$, and $\bar{z}$. No uniqueless can be guaranteed for subformulas of the form $o_{\bar{b}}$. We note, however, that in the second phase when we perform SO quantifier elimination of SO predicates of level 1 for a (generalized) clause c, a formula is encountered of the form

$$c_0 \vee \bigvee_{l=1}^{r} \forall \bar{z}_R. \bigwedge_{j=1}^{m_l} (\bigvee_{i=1}^{n_{lj}} (\bar{b}_l = \bar{a}_{lji}) \wedge g_{lji} \vee c_{lj}) \tag{23}$$

where c_0 consists of literals using variables from V', the lists of variables $\bar{a}_{lji}$ are only from V', and the subformulas g_{lji} and c_{lj} use state predicates of level 0 only where all occurring FO variables are from $V' \cup Z$. The number of non-equivalent such formulas g_{lji} as well as c_{lj} with this restriction onto the occurring FO variables, however, is finite. Therefore, there is a fixed finite prefix of FO variables, independent of the length of the sequence of substitutions π to take all universal quantifications from (23) into account.

Altogether, therefore, the number of FO variables in quantified clauses $\forall \bar{z}'.c'$ contained in $\pi(\psi)$ remains bounded – even when we allow resets both on the maximal level and on level 1. This completes the proof of lemma 9.

We remark that theorem 8 remains true if there are predicates R' with stratified guarded updates as well as resets also at non-extremal levels — given that neither their updates nor their resets introduce FO variables, i.e., the variable lists $\bar{z}$ in (6) are empty. In general, though, the proof technique of theorem 8 cannot easily be extended to FO transition systems with arbitrary resets of the form (7), since then conjunctions of the form $o_{\bar{b}}$ with non-empty lists of quantified variables may also occur at higher levels — where it is no longer clear how to prove that their number is finite.

8 Noninterference for Multi-agent systems

In the following we would like to apply the termination results from the last section to prove non-interference for multi-agent systems represented as FO transition system. In this representation, we assume the FO transition system to be executed by *agents*. Thereby, an agent a can observe all tuples of state relations that mention her in the first component (i.e., all tuples of the form $a\bar{b}$ satisfying some predicate R). Accordingly, we assume all predicates to have arities at least 1. Subsequently, we assume that each substitution of the FO transition system $\mathcal{T}$ either is an update or a reset. In particular, there are no conditions at edges in the control-flow graph. Besides public input from the environment, we distinuish two dedicated further kinds of input predicates, namely, *oracles* and *choices*.

An oracle predicate O is used to represent *secret data input* to the FO transition system which is meant to be disclosed only to a subset of agents. Thereby, each oracle O comes with a FO formula $\delta_O x\bar{y}$ specifying for each agent x, which tuples $\bar{y}$ should possibly be visible to x.

We remark that an oracle predicate O with $\delta_O x\bar{y} = \top$ (*true*) for all $x, \bar{y}$, does not pose any restrictions on the visibility of tuples $\bar{y}$ and thus may serve as (public) input from the environment. It is for conceptual clarity only, that we consider environment input separately.

Choice predicates C formalize the behavior and individual decisions of agents. The literal $Cx\bar{y}$, when true, indicates that agent x offers tuple $x\bar{y}$ for the current update operation. Let $\mathcal{R}_{env}, \mathcal{R}_{high}, \mathcal{R}_{low} \subseteq \mathcal{A}$ denote the set of all environment input, oracle and choice predicates, respectively, that are used by the given FO transition system.

Consider, e.g., the FO transition system from Fig. 1. Then A_1,A_2 may be considered as input from the environment (providing information on conflicts and the assignment of papers to pc members), while A_3 is an oracle: it provides reports on papers which should not be disclosed to all members of the pc. The input predicate A_4, on the other hand, represents the *choices* of pc members what to contibute to the discussion on papers. According to our convention, it is agent x_2 who (by means of A_4) decides to add tuples to predicate $\mathsf{discuss}$ which in this way become visible to x_1.

Noninterference is best formulated as a 2-*hyperproperty* [26], that is, a property of pairs of traces. In our application, the sequence of edges traversed by an execution of the workflow is determined *externally*, i.e., independent of any oracle or choice predicate. For instance in case of a conference management system, it is up to the PC chair to decide when a particular stage is complete and which next stage to execute. This means that we are only interested in 2-hyperproperties where the considered two traces follow the *same* control flow path, but may differ in the sequences of attained states. This restriction has also been imposed in [15, 16]. In order to reason about the pairs of states attained by a pair of traces, we introduce a copy $\mathcal{R}' = \{R' \mid R \in \mathcal{R}\}$ of the predicates in $\mathcal{R}$ and assume that the states s'_i are expressed by means of the predicates in $\mathcal{R}_{state}{}'$, i.e., primed state predicates. Thus, we can combine each pair $\langle s_i, s'_i \rangle$ of first-order structures into a single structure $s_i \otimes s'_i$ over $\mathcal{R}_{state} \cup \mathcal{R}_{state}{}'$. Following [3, 15, 16], noninterference is expressed from the point of view of a single (but arbitrary) agent, and the notions of high/low security inputs/outputs from the standard definition of noninterference [17] are interpreted with respect to this agent. Furthermore, the property is parameterized by an assumption on the behavior of agents (called *agent model*) and by *declassification conditions*, which specify when and what information can be legitimately exposed. Therefore, *Noninterference with Declassification and Agent model* (NDA) is expressed by the FOLTL formula

$$G \text{ public\_env} \land \text{agent\_model} \rightarrow \\ \forall a.(G \text{ same\_high\_inputs}(a)) \rightarrow (G \text{ same\_observations}(a)) \tag{24}$$

where G is the LTL *globally* operator. The property states that for any two traces following the same control-flow path, a given agent model for every agent a, and the assumption that public input from the environment does not differ on both traces, that the noninterference property holds iff agent a is never able to observe a difference between two traces that differ only in the (non-declassified) inputs from the oracles:

$$\begin{aligned} \mathit{public\_env} &:= \bigwedge_{R \in \mathcal{R}_{env}} \left(\forall \bar{y}_R.\, R\bar{y}_R \leftrightarrow R'\bar{y}_R\right) \\ \mathit{same\_observations}(x) &:= \bigwedge_{R \in \mathcal{R}_{state}} \left(\forall \bar{z}_R.\, Rx\bar{z}_R \leftrightarrow R'x\bar{z}_R\right) \\ \mathit{same\_high\_inputs}(x) &:= \bigwedge_{O \in \mathcal{R}_{high}} \forall \bar{z}_R.\, \left(\delta_O x\bar{z}_O \rightarrow (O\bar{z}_O \leftrightarrow O'\bar{z}_O)\right) \end{aligned}$$

where $\bar{y}_R, \bar{z}_R, \bar{z}_O$ are sequences of distinct variables of appropriate lengthes. For each oracle predicate O and agent x, the formula $\delta_O x\bar{z}_O$ using predicates from $\mathcal{R}_{state}$ encodes a declassification condition that specifies which tuples $\bar{z}_O$ from O can be made visible without causing a security breach to x. For our running example, we use $\delta_O(x, y, p, r) := \neg\mathsf{conf}(x, p)$. This example declassification condition allows any agent x to safely read reports p by reviewer y on paper p, as lang as x neither is an author of p nor has declared conflict with p.

For any agent, we consider two kinds of possible behavior. One agent either *stubbornly* makes the same choices, independently of its observations; or its choices may depend on previous observations, i.e., it acts *causally*. Causal behavior characterizes the behavior of members of an *adversarial coalition* who in this way try to spread secret information. The two behaviors are captured by the following formulas, respectively.

$$\begin{aligned} \mathit{stubborn}(x) &:= \mathsf{G}\ \mathit{same\_low\_inputs}(x) \\ \mathit{causal}(x) &:= \mathit{same\_low\_inputs}(x)\ \mathsf{W}\ \neg \mathit{same\_observations}(x) \end{aligned}$$

where

$$\mathit{same\_low\_inputs}(x) := \bigwedge_{C \in \mathcal{R}_{low}} \left(\forall \bar{z}_C.\, Cx\bar{z}_C \leftrightarrow C'x\bar{z}_C\right)$$

and W denotes the *weak until* LTL operator. Note that any stubborn agent also satisfies the causality assumption which allows for more behaviors. Therefore, the most general agent model is when each agent is causal, while the most restrictive model is when each agent is stubborn.

We thus assume that the *agent_model* formula from the formalization of NDA can be instantiated with one of the following formulas:

$$\begin{aligned} \mathit{agent\_model}^{(c,t)} &:= \exists y_1, \dots, y_t.\, \left(\bigwedge_{i=1}^{t} \mathit{causal}(y_i)\right) \land \\ &\quad\ \left(\forall x.\, (\bigwedge_{i=1}^{t} x \neq y_i) \rightarrow \mathit{stubborn}(x)\right) \\ \mathit{agent\_model}^{(c)} &:= \forall x.\mathit{causal}(x) \end{aligned}$$

where $t \geq 0$. Note that $\mathit{agent\_model}^{(c,0)} \equiv \forall x.\mathit{stubborn}(x)$. We denote this formula by $\mathit{agent\_model}^{(s)}$.

9 Encoding agent models and declassification

The key idea for verifying NDA from [3] is to introduce a fresh agent constant a, and then encode property (24) for a, a particular agent model and a given FO transition system $\mathcal{T}$ as an *invariant* of a (suitably defined) synchronous *self-composition* of the FO transition system $\mathcal{T}$ in a way that the invariant consists of universal FO formulas only. For convenience, we recall these constructions in order to adapt these to the specific setting of FO transition systems with guarded stratified updates and resets. Let us first assume that all agents are *stubborn*. The new FO transition system $\mathcal{P}_a^{(s)}(\mathcal{T})$ then is constructed as follows. Let $\mathcal{R}'_{state}$ denote the set of primed predicates R' corresponding to the state predicates R used by $\mathcal{T}$. For a first-order formula φ with predicates from $\mathcal{R}_{state}$, let $[\varphi]'$ denote the formula obtained from φ by replacing each predicate $R \in \mathcal{R}_{state}$ with the corresponding predicate R' in $\mathcal{R}'_{state}$. Then each edge $e = (u, \theta, v)$ of $\mathcal{T}$ gives rise to a sequence of edges

$$(u, \theta_0, u_{e,1}), (u_{e,1}, \theta_1, u_{e,2}), \ldots, (u_{e,r-1}, \theta_{r-1}, u_{e,r}), (u_{e,r}, \theta_r, v)$$

for a sequence of fresh auxiliary nodes $u_{e,1}, \ldots, u_{e,r}$ and a sequence $\theta_0, \ldots, \theta_r = \mathcal{P}_a^{(s)}\theta$ defined as follows.

Reset. If θ is a reset of the form $R\bar{y} := \varphi$, then $\mathcal{P}_a^{(s)}(\theta)$ is given by the sequence of resets, first, of R, then of R':

$$\begin{aligned} &R\bar{y} := \varphi; \\ &R'\bar{y} := [\varphi]' \end{aligned}$$

Update. Now assume that θ is an update of the form

$$\{R_j\bar{y}_{R_j} := \varphi_j \vee \exists \bar{z}_j.A\bar{y}_{R_j}^{\sigma_j}\bar{z}_j \wedge \psi_j \mid j = 1, \ldots, r\}$$

If A is an input predicate from the environment, or a choice predicate, then A is used both for updating the state predicates and the primed state predicates. This means that $\mathcal{P}_a^{(s)}(\theta)$ is given by the single update

$$\begin{aligned} \{R_j\bar{y}_{R_j} &:= R_j\bar{y}_{R_j} \vee \varphi_j \vee \exists \bar{z}_j.A\bar{y}_{R_j}^{\sigma_j}\bar{z}_j \wedge \psi_j; \\ R'_j\bar{y}_{R_j} &:= R'_j\bar{y}_{R_j} \vee [\varphi_j]' \vee \exists \bar{z}_j.A\bar{y}_{R_j}^{\sigma_j}\bar{z}_j \wedge [\psi_j]' | j = 1, \ldots, r\} \end{aligned} \tag{25}$$

In particular, no auxiliary nodes are required here.

If the input predicate A equals some secret oracle O, then the construction must make sure that, generally, the value of O may differ on both traces – up to when declassification for O and a applies. Accordingly, we *split* the corresponding update into the sequence of updates $\theta_0, \theta_1, \theta_2$.

The update θ_0 takes care of argument tuples $x\bar{y}$ where declassification w.r.t. agent a applies, while θ_1, θ_2 deal with argument tuples where declassification does not apply.

The update θ_0 is as (25) only that the formulas ψ_j, $[\psi_j]'$ in the right-hand sides are now replaced with

$$\psi_j \wedge \delta_O a\bar{y}\bar{z} \qquad \text{and} \qquad [\psi_j]' \wedge \delta_O a\bar{y}\bar{z}$$

respectively.

Tuples then may independently be added to R_j and R'_j, when declassification does not apply. This independent addition is taken care of by the updates θ_1, θ_2:

$$\theta_1 = \{ \begin{array}{l} R_j\bar{y}_{R_j} := R_j\bar{y}_{R_j} \vee \exists \bar{z}_j . A\bar{y}^{\sigma_j}_{R_j}\bar{z}_j \wedge \psi_j \wedge \neg\delta_O a\bar{y}_{R_j}\bar{z}_j \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad | j = 1, \dots, r\} \end{array} \tag{26}$$

$$\theta_2 = \{ \begin{array}{l} R'_j\bar{y}_{R_j} := R'_j\bar{y}_{R_j} \vee \exists \bar{z}_j . A\bar{y}^{\sigma_j}_{R_j}\bar{z}_j \wedge [\psi_j]' \wedge \neg\delta_O a\bar{y}_{R_j}\bar{z}_j \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad | j = 1, \dots, r\} \end{array} \tag{27}$$

Causality. Let us now consider *causal* agents. The corresponding new FO transition system $\mathcal{P}^{(c)}_a(\mathcal{T})$ is constructed as the FO transition system $\mathcal{P}^{(s)}_a(\mathcal{T})$ – with the only difference that choices of agents now may depend on whether they previously have acquired knowledge about secrets or not.

In order to capture causality, a new unary predicate $I(x)$ is introduced which records all agents x that have already made observations depending on information secret to a. These agents' choices may diverge when updating predicates R, R', respectively. Initially, $\forall x.\neg I(x)$ holds.

Moreover, after each sequence of substitutions realizing the modification of unprimed or primed state predicates, the predicate I itself must be updated — depending on whether the substituted predicates $R_1, \dots, R_r$ differ from their primed versions. This update θ_I is given by

$$I(x) := I(x) \vee \exists \bar{z}.Ax\bar{z} \wedge \bigvee_{j=1}^{r} \neg(R_j x\bar{z}_j \leftrightarrow R'_j x\bar{z}_j) \tag{28}$$

where $\bar{z}_j$ are lists of distinct FO variables, and $\bar{z}$ is an enumeration of all FO variables occurring in the lists $\bar{z}_j$. Informally, update θ_I checks wether a difference in any relation R_j can be been observed by agent x and if so, adds x to the predicate I. The input predicate A guards this check, which allows us to later eliminate the existential quantifier during the fixpoint iteration.

Beyond the addition of θ_I, a second modification occurs for updates θ of the form

$$\{R_j\bar{y}_{R_j} := \varphi_j \vee \exists \bar{z}_j.\ C\bar{y}^{\sigma_j}_{R_j}\bar{z}_j \wedge \psi_j \mid j = 1, \dots, r\}$$

which query some choice predicate C. This update is simulated in the self-composition by the sequence of substitutions $\theta_0, \theta_1, \theta_2, \theta_I$. Thereby, θ_0 is defined as (25) (with $A = C$) — with the only exception that the subformulas ψ_j, $[\psi_j]'$ in right-hand sides are replaced with

$$\psi_j \wedge \neg I(x) \qquad \text{and} \qquad [\psi_j]' \wedge \neg I(x)$$

if x is the first variable from the sequence $\bar{y}^{\sigma_j}_{R_j}\bar{z}_j$. The substitutions θ_1, θ_2 perform *independent* updates — given that agent x already has been informed. This means that

$$\theta_1 = \{R_j\bar{y}_{R_j} := R_j\bar{y}_{R_j} \vee \exists \bar{z}_j.C_1\bar{y}^{\sigma_j}_{R_j}\bar{z}_j \wedge \psi_j \wedge I(x) \mid j = 1, \dots, r\} \tag{29}$$

$$\theta_2 = \{R'_j\bar{y}_{R_j} := R'_j\bar{y}_{R_j} \vee \exists \bar{z}_j.C_2\bar{y}^{\sigma_j}_{R_j}\bar{z}_j \wedge [\psi_j]' \wedge I(x) \mid j = 1, \dots, r\} \tag{30}$$

Note here that there is just one instance of the predicate I which is queried both for updating unprimed and primed versions of state predicates.

Example 11 Consider in Fig. 1 the update

$$\mathsf{report}(x,p,r) \ := \mathsf{report}(x,p,r) \vee A_3(x,p,r) \wedge \mathsf{assign}(x,p)$$

A_3 is a secret oracle with corresponding declassification formula $\neg\mathsf{conf}(a,p)$. Let us first consider *stubborn* agents only: Then the transformation results in the sequence of three updates

$$\begin{array}{l}
\{ \ \mathsf{report}(x,p,r) := \mathsf{report}(x,p,r) \vee A_3(x,p,r) \wedge \mathsf{assign}(x,p) \wedge \neg\mathsf{conf}(a,p); \\
\quad \mathsf{report}'(x,p,r) := \mathsf{report}'(x,p,r) \vee A_3(x,p,r) \wedge \mathsf{assign}'(x,p) \wedge \neg\mathsf{conf}(a,p) \\
\} \\
\quad \mathsf{report}(x,p,r) := \mathsf{report}(x.p,r) \vee A_3(x,p,r) \wedge \mathsf{assign}(x,p) \wedge \mathsf{conf}(a,p) \\
\quad \mathsf{report}'(x,p,r) := \mathsf{report}'(x.p,r) \vee A_3(x,p,r) \wedge \mathsf{assign}'(x,p) \wedge \mathsf{conf}(a,p)
\end{array}$$

In case of *causal* agents, subsequently also the informedness predicate I must be updated by

$$I(x) := I(x) \vee \exists p, r. A_3(x,p,r) \wedge \neg(\mathsf{report}(x,p,r) \leftrightarrow \mathsf{report}'(x,p,r))$$

Now consider the substitution

$$\begin{array}{l}
\mathsf{discuss}(x_1,x_2,p,d) \ := \mathsf{discuss}(x_1,x_2,p,d) \vee \\
\qquad \exists r_1, r_2.A_4(x_2,x_1,p,d,r_1,r_2) \wedge \mathsf{report}(x_1,p,r_1) \wedge \mathsf{report}(x_2,p,r_2)
\end{array}$$

where A_4 is a choice predicate, allowing agent x_2 to decide which tuples to add. In case of *stubborn* agents, agent choices are independent of acquired information about secrets. Therefore, the transformation results in just one (simultaneous) update of both $\mathsf{discuss}$ and $\mathsf{discuss}'$ together with an update of the predicate I. In case of *causal* agents, however, four updates are introduced in a row, namely,

$$\begin{array}{l}
\{ \ \mathsf{discuss}(x_1,x_2,p,d) \ := \mathsf{discuss}(x_1,x_2,p,d) \vee \\
\qquad \exists r_1, r_2.A_4(x_2,x_1,p,d,r_1,r_2) \wedge \mathsf{report}(x_1,p,r_1) \wedge \mathsf{report}(x_2,p,r_2) \wedge \neg I(x_2); \\
\quad \mathsf{discuss}'(x_1,x_2,p,d) \ := \mathsf{discuss}'(x_1,x_2,p,d) \vee \\
\qquad \exists r_1, r_2.A_4(x_2,x_1,p,d,r_1,r_2) \wedge \mathsf{report}'(x_1,p,r_1) \wedge \mathsf{report}'(x_2,p,r_2) \wedge \neg I(x_2) \\
\} \\
\quad \mathsf{discuss}(x_1,x_2,p,d) \ := \mathsf{discuss}(x_1,x_2,p,d) \vee \\
\qquad \exists r_1, r_2.A_4(x_2,x_1,p,d,r_1,r_2) \wedge \mathsf{report}(x_1,p,r_1) \wedge \mathsf{report}(x_2,p,r_2) \wedge I(x_2); \\
\quad \mathsf{discuss}'(x_1,x_2,p,d) \ := \mathsf{discuss}'(x_1,x_2,p,d) \vee \\
\qquad \exists r_1, r_2.A_4(x_2,x_1,p,d,r_1,r_2) \wedge \mathsf{report}'(x_1,p,r_1) \wedge \mathsf{report}'(x_2,p,r_2) \wedge I(x_2); \\
\quad I(x_1) \ := I(x_1) \vee \exists x_2, p, d. A_4(x_1,x_2,p,d) \wedge \\
\qquad\quad \neg(\mathsf{discuss}(x_1,x_2,p,d) \leftrightarrow \mathsf{discuss}'(x_1,x_2,p,d))
\end{array}$$

Splitting the updates to $\mathsf{discuss}$ and $\mathsf{discuss}'$ into three only takes effect for agents x_2 which are already informed, i.e., for which $I(x_2)$ holds true. For the others, just the first (simultaneous) update takes effect. □

The transformed workflow $\mathcal{P}^{(c)}_a(\mathcal{T})$ (or $\mathcal{P}^{(s)}_a(\mathcal{T})$) captures all pairs of traces of $\mathcal{T}$ that satisfy the causal (or stubborn) agent model together with declassification, relative to a. We recall the following theorem from [3].

Theorem 10 [[3]] *Let $\mathcal{T}$ without assumptions and stratified guarded updates and resets only.*

Let $\mathcal{P}_a^{(s)}(\mathcal{T})$ ($\mathcal{P}_a^{(c)}(\mathcal{T})$) denote FO transition systems obtained as the self-composition of $\mathcal{T}$ for stubborn (causal) agents. Let Φ_{NDA} denote the universal invariant mapping each node u to the formula

$$\forall \bar{y}. \bigwedge_{R \in \mathcal{R}_{state}} Ra\bar{y}'_R \leftrightarrow R'a\bar{y}'_R \tag{31}$$

where for each predicate $R \in \mathcal{R}_{state}$, $\bar{y}'_R$ is an appropriate list of distinct FO variables, and $\bar{y}$ is an enumeration of the FO variables occurring in any of the $\bar{y}'_R$.

Then the following holds for $\mathcal{T}$ with initial hypothesis $\mathcal{H}$.

1. 2-*hyperproperty (NDA) holds for stubborn agents iff invariant Φ_{NDA} holds for $\mathcal{P}_a^{(s)}(\mathcal{T})$ with initial hypothesis*

$$\mathcal{H} \wedge \bigwedge_{R \in \mathcal{R}_{state}} \forall \bar{y}_R . R\bar{y}_R \leftrightarrow R'\bar{y}_R \tag{32}$$

2. 2-*hyperproperty (NDA) holds for causal agents iff invariant Φ_{NDA} holds for $\mathcal{P}_a^{(c)}(\mathcal{T})$ with initial hypothesis*

$$\mathcal{H} \wedge \forall x. \neg I(x) \wedge \bigwedge_{R \in \mathcal{R}_{state}} \forall \bar{y}_R . R\bar{y}_R \leftrightarrow R'\bar{y}_R \tag{33}$$

□

10 Application to noninterference

Theorem 10 allows us to apply our methods for constructing weakest inductive invariants for proving NDA for certain FO transition systems. Let us first consider stubborn agents only. Assume that the FO transition system $\mathcal{T}$ is stratified and guarded without using assumptions at edges. Then the same is true for the transformed workflow $\mathcal{P}_a^{(s)}(\mathcal{T})$ which takes care of stubbornness of agents and declassification relative to a — at least, when for each oracle O, the declassification formula δ_O is sufficiently well-behaved — i.e. is quantifierfree and uses only predicates of level less than the left-hand sides of substitutions where O is queried. Thereby, we assign to primed predicates R' the same levels as the corresponding unprimed predicates R. Furthermore, if all resets of $\mathcal{T}$ occur either on level 1, or the maximal level, then this property also holds for $\mathcal{P}_a^{(s)}(\mathcal{T})$. Likewise, if $\mathcal{T}$ uses arbitrary resets, but strict updates only, then the same holds true also for $\mathcal{P}_a^{(s)}(\mathcal{T})$. As application of Theorems 8 and 5, we therefore obtain:

Theorem 11 *Consider a FO transition system $\mathcal{T}$ without assumptions, but with resets and stratified guarded updates Assume that for each oracle O queried in some update θ, the corresponding declassification formula δ_O is quantifierfree where each predicate occurring in δ_O has level less than the level of the left-hand sides in θ. Assume further that one of the two conditions are met:*

1. *All updates are strictly guarded*; *or*
2. *All resets either occur at maximal level or at level* 1.

Assume that the initial hypothesis $\mathcal{H}$ for $\mathcal{T}$ is given as a FO formula from the prenex class $\exists^*\forall^*$. Assume that all agents participating in $\mathcal{T}$ are stubborn. Then NDA for $\mathcal{T}$ with stubborn agents is effectively decidable.

Proof Under each of the two assumptions, the self-composition $\mathcal{P}_a^{(s)}(\mathcal{T})$ can be effectively constructed and satisfies either the assumptions of Theorem 5 or Theorem 8. Therefore for every universal invariant Φ_{NDA}, the weakest inductive invariant Ψ can be effectively computed so that $\Psi[u] \to \Phi[u]$ for every is again uniform and leveled. This invariant then holds iff the conjunction of the formula (32) and $\neg\Psi[v_0]$ is unsatisfiable. Since both formulas are effectively contained in the prenex class $\exists^*\forall^*$, the claim of the theorem follows. □

Example 12 Consider again the FO transition system $\mathcal{T}$ from Fig. 1 where all participating agents are stubborn. In order to prove NDA for the arbitrary agent a, it suffices to verify for every program point of the self-compostion $\mathcal{P}_a^{(s)}(\mathcal{T})$ the following invariant holds:

$$\begin{array}{l} \forall p.\ \mathsf{conf}(a,p) \leftrightarrow \mathsf{conf}'(a,p) \quad \wedge \\ \forall p.\ \mathsf{assign}(a,p) \leftrightarrow \mathsf{assign}'(a,p) \quad \wedge \\ \forall p,r.\ \mathsf{report}(a,p,r) \leftrightarrow \mathsf{report}'(a,p,r) \quad \wedge \\ \forall x,p,d.\ \mathsf{discuss}(a,x,p,d) \leftrightarrow \mathsf{discuss}'(a,x,p,d) \end{array} \tag{34}$$

Since the self-composition $\mathcal{P}_a^{(s)}(\mathcal{T})$ is stratified strictly guarded (here even without resets), the iterative strengthening of the invariant terminates with a weakest inductive invariant. This invariant is too complicated to be spelled out here. We argue, however, that throughout all program points 0, 1, 2, 3 of the FO transition system,

$$\forall x,p.\ (\mathsf{conf}(x,p) \leftrightarrow \mathsf{conf}'(x,p)) \wedge (\mathsf{assign}(x,p) \leftrightarrow \mathsf{assign}'(x,p))$$

holds. Moreover, we have at all program points,

$$\forall x,p,r.\ \mathsf{conf}(a,p) \vee (\mathsf{report}(x,p,r) \leftrightarrow \mathsf{report}'(x,p,r))$$

where

$$\forall p,r.\ \neg\mathsf{conf}(a,p) \vee \neg\mathsf{report}(a,p,r)$$

holds. The latter property can, e.g., be proven for the original FO transition system $\mathcal{T}$. Each of these invariants is already inductive. Together, they imply that

$$\forall x,p,d.\ \mathsf{discuss}(a,x,p,d) \leftrightarrow \mathsf{discuss}'(a,x,p,d)$$

holds for all program points as well — implying that the invariant (34) holds. □

We would like to apply the same strategy to certify NDA now for FO transition systems with *causal* agents. Again assume that the FO transition system $\mathcal{T}$ is stratified and guarded. The workflow $\mathcal{P}_a^{(c)}(\mathcal{T})$, however, is no longer stratified. This is due to the auxiliary predicate I introduced by $\mathcal{P}_a^{(c)}$. That predicate is queried at updates of all updated predicates R, R', $R \in \mathcal{R}_{state}$ where a choice predicate is queried. Likewise it is updated by means of formulas which depend on the same predicates. Still, we can apply fixpoint iteration on universal formulas — which, however, is no longer guaranteed to terminate. Thus, we obtain a possibly *incomplete* method for certifying NDA for agent model *agent_model*$^{(c)}$. The potential non-termination, however, does not come as a surprise as NDA in general is undecidable for an unbounded number of causal agents [15]. Interestingly, the situation is different when only a *fixed* bounded number of agents behaves causally, while all others behave stubbornly.

Assume that at most $t \geq 0$ agents behave causally, while all other agents are stubborn. Our goal is to provide a dedicated self-composition of the given FO transition system $\mathcal{T}$ which takes care of the particular agent model, while preserving stratification and guardedness of the transition system. In the case that there are at most t causal agents, the informedness predicate I may receive only finitely many values. These finitely many values, however, can be encoded into the program points of the transformed FO transition system itself. Updates to I now, however, show up as *assumptions* at certain control flow edges.

Let $y_1, \ldots, y_t$ denote a sequence of t distinct fresh variables. The initial program point of the self-composition $\mathcal{P}_a^{c,t}(T)$ then is given by $\langle v_0, \emptyset \rangle$, since no causal agent is informed in the beginning. Now consider the substitution θ at an edge $e = (u, \theta, v)$ of $\mathcal{T}$ possibly modifying predicates $R_1, \ldots, R_k$. Let $\theta_0, \ldots, \theta_r = \mathcal{P}_a^{(c)}\theta$ denote the sequence of substitutions of $\mathcal{P}_a^{(c)}(\mathcal{T})$ meant to simulate θ, where the last substitution θ_r is the update to the predicate I. Let $u_{e,1}, \ldots, u_{e,r}$ denote the auxiliary nodes introduced by $\mathcal{P}_a^{(c)}(\mathcal{T})$ for simulating the sequence of substitutions $\mathcal{P}_a^{(c)}\theta$. Let $Y \subset Y' \subseteq \{y_1, \ldots, y_t\}$ where Y is the set of informed causal agents before the execution of $\theta_0, \ldots, \theta_r$, and Y' a possible set of causal agents informed afterwards.

First assume that θ does not query a choice predicate, i.e., none of the substitutions θ_j mentions the predicate I. Then $\mathcal{P}_a^{c,t}(\mathcal{T})$ has edges

- $(\langle u, Y \rangle, \theta_0, \langle u_{e,1}, Y \rangle), (\langle u_{e,1}, Y \rangle, \theta_1, \langle u_{e,2}, Y \rangle), \ldots, (\langle u_{e,r-1}, Y \rangle, \theta_{e,r-1}, \langle u_{e,r}, Y \rangle)$; together with
- $(\langle u_{e,r}, Y \rangle, \mathsf{id}, \langle v, Y \rangle)$ where id is the identical substitution, and
- $(\langle u_{e,r}, Y \rangle, (g_{Y,Y'}; \mathsf{id}), \langle v, Y' \rangle)$ where the assumption $g_{Y,Y'}$ is given by

$$\bigwedge_{y' \in Y' \setminus Y} \bigvee_{j=1}^{k} \exists \bar{z}. \neg(R_j y' \bar{z}_j \leftrightarrow R'_j y' \bar{z}_j)$$

where $\bar{z}_j$ are lists of distinct FO variables of appropriate length and $\bar{z}$ is an enumeration of all variables occurring in any of the $\bar{z}_j$. Assuming that all agents in Y are already informed, the assumption $g_{Y,Y'}$ implies that at the new node $\langle v, Y' \rangle$, (at least) all agents from Y' are informed.

Finally, assume that θ is an update querying a choice predicate C. Then the same construction applies — only that in the substitutions $\theta_0, \ldots, \theta_{r-1}$ the literals $I(x)$ and $\neg I(x)$ must be replaced with the formulas

$$\bigvee_{y' \in Y} x = y' \qquad \text{and} \qquad \bigwedge_{y' \in Y} x \neq y'$$

respectively. The correctness of the transformation can be proven along the same lines as Theorem 10. Theorems 8 and 5 can still be applied, since no assumptions are introduced inside strongly connected components. Therefore, we finally obtain:

Theorem 12 *Consider a FO transition system $\mathcal{T}$ and declassification formulas satisfying the same assumptions as in theorem 11, and assume that at most $t \geq 0$ of the agents* *participating in $\mathcal{T}$ are causal, while all other agents are stubborn. Then NDA for $\mathcal{T}$ is effectively decidable.* □

Example 13 Consider again the FO transition system from Fig. 1, but now with causal agents and the self-composition $\mathcal{P}_a^{(c)}(\mathcal{T})$ where informedness of agents is maintained via the explicit auxiliary predicate I. Interestingly, the fixpoint iteration for proving the invariant (34) still terminates with a rather complicated inductive invariant. The latter, however, is not weak enough to prove NDA. In fact, this property cannot be guaranteed, as is indicated by the following counterexample:

$$\begin{array}{lll}
\mathcal{U} & = & \{a, a', p, p', r, r_1, r_2, d_1, d_2\} \\
I\,\mathsf{conf} & = & \{(a, p)\} \\
I\,\mathsf{assign} & = & \{(a, p'), (a', p), (a', p')\} \\
I\,\mathsf{report} & = & \{(a, p', r), (a', p', r), (a', p, r_1)\} \\
I\,\mathsf{report}' & = & \{(a, p', r), (a', p', r), (a', p, r_2)\} \\
I\,\mathsf{discuss} & = & \{(a, a', p', d_1)\} \\
I\,\mathsf{discuss}' & = & \{(a, a', p', d_2)\}
\end{array}$$

Intuitively, pc member a has declared conflict with paper p, but has been assigned paper p'. Pc member a' on the other hand, has declared no conflict and has been assigned both papers p and p'. Since pc members a, a' share a common paper p' for which both have received a report, they may start a discussion on paper p'. This discussion can be used by a' to leak secret information on paper p, i.e., depending on the received report on p, a' may contribute d_1 or d_2 to the discussion of p'. □

11 Conclusion

We have investigated FO transition systems where all substitutions are either guarded updates or guarded resets. For these, we observed that the exact weakest pre-condition of a universal FO formula is again a universal FO formula, thus allowing us to realize a fixpoint computation of iterated strengthening for proving the validity of universal invariants. In order to identify subclasses of FO transition systems where termination can be guaranteed, we relied on a natural notion of stratification. Here, we were able to prove termination (and thus decidability) for three interesting subclasses of stratified guarded FO transition systems. However, it remains as an open question whether termination can be proven for *all* FO transition systems with stratified guarded updates and resets.

In the second part, we applied the obtained termination results to multi-agent systems encoded as FO transition systems. This formalism subsumes the workflow language as considered in [3] for analyzing noninterference in presence of declassification and agent coalitions (NDA). We indicated how NDA can be naturally encoded as a universal invariant of suitably defined *self-compositions* of the given FO transition systems [27, 28]. At least for the case of *stubborn agents* [16], i.e., agents who do not participate in adversarial coalitions and for the case where the number of causal agents is bounded, our novel decidability results translate into decidability of NDA.

Acknowledgements This work was supported by Shota Rustaveli National Science Foundation of Georgia under the project FR-21-7973.

Funding Open Access funding enabled and organized by Projekt DEAL.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Padon O, Losa G, Sagiv M, Shoham S (2017) Paxos made EPR: decidable reasoning about distributed protocols. Proc ACM Program Lang. https://doi.org/10.1145/3140568
2. Seidl H, Müller C, Finkbeiner B (2020) How to win first-order safety games. In: Beyer, D., Zufferey, D. (eds.) Verification, Model Checking, and Abstract Interpretation - 21st International Conference (VMCAI) 2020), pp. 426–448. LNCS 11990, Springer, Heidelberg, Berlin. https://doi.org/10.1007/978-3-030-39322-9_20
3. Müller C, Seidl H, Zalinescu E (2018) Inductive invariants for noninterference in multi-agent workflows. In: 31st IEEE Computer Security Foundations Symposium, (CSF 2018), pp. 247–261. IEEE, New York. https://doi.org/10.1109/CSF.2018.00025
4. Börger E, Stärk R (2003) History and survey of ASM research. Abstract state machines: a method for High-level system design and analysis. Springer, Heidelberg, Berlin, pp 343–367
5. Börger E, Stärk R (2003) Tool support for ASMs. Abstract state machines: a method for High-level system design and analysis. Springer, Heidelberg, Berlin, pp 313–342
6. Gurevich Y (2018) Evolving algebras 1993: Lipari guide. arXiv preprint arXiv:1808.06255
7. Ball T, Bjørner N, Gember A, Itzhaky S, Karbyshev A, Sagiv M, Schapira M, Valadarsky A (2014) Vericon: towards verifying controller programs in software-defined networks. ACM Sigplan Notices 49(6):282–293
8. Padon O, Immerman N, Karbyshev A, Lahav O, Sagiv M, Shoham S (2015) Decentralizing SDN policies. ACM SIGPLAN Notices 50(1):663–676
9. Padon O, McMillan KL, Panda A, Sagiv M, Shoham S (2016) Ivy: safety verification by interactive generalization. ACM SIGPLAN Notices 51(6):614–630
10. McMillan KL, Padon O (2018) Deductive verification in decidable fragments with ivy. In: Podelski, A. (ed.) Static Analysis - 25th International symposium, SAS 2018, Freiburg, Germany, August 29-31, 2018, Proceedings, pp. 43–55. LNCS 11002, Springer, Heidelberg, Berlin. https://doi.org/10.1007/978-3-319-99725-4_4
11. De Moura L, Bjørner N (2008) Z3: An efficient SMT solver. In: Ramakrishnan CR, Rehof J (eds) International conference on tools and algorithms for the construction and analysis of systems. Springer, Heidelberg, Berlin, pp 337–340
12. Berkovits I, Lazic M, Losa G, Padon O, Shoham S (2019) Verification of threshold-based distributed algorithms by decomposition to decidable logics. In: Dillig, I., Tasiran, S. (eds.) Computer aided verification - 31st international conference, CAV 2019, New York City, NY, USA, July 15-18, 2019, Proceedings, Part II, pp. 245–266. LNCS 11562, Springer, Heidelberg, Berlin. https://doi.org/10.1007/978-3-030-25543-5_15
13. Karbyshev A, Bjørner N, Itzhaky S, Rinetzky N, Shoham S (2017) Property-directed inference of universal invariants or proving their absence. Journal of the ACM (JACM) 64(1):7
14. Koenig JR, Padon O, Immerman N, Aiken A (2020) First-order quantified separators. In: Donaldson, A.F., Torlak, E. (eds.) Proceedings of the 41st ACM SIGPLAN international conference on programming language design and implementation, PLDI 2020, London, UK, June 15-20, 2020, pp. 703–717. ACM, New York. https://doi.org/10.1145/3385412.3386018
15. Finkbeiner B, Müller C, Seidl H, Zalinescu E (2017) Verifying security policies in multi-agent workflows with loops. In: Proceedings of the 2017 ACM SIGSAC conference on computer and communications security (CCS 2017), pp. 633–645. IEEE, New York. https://doi.org/10.1145/3133956.3134080

16. Finkbeiner B, Seidl H, Müller C (2016) Specifying and verifying secrecy in workflows with arbitrarily many agents. In: Proceedings of the 14th international symposium on automated technology for verification and analysis (ATVA 2016), pp. 157–173. LNCS 9938, Springer, Heidelberg, Berlin
17. Goguen JA, Meseguer J (1982) Security policies and security models. In: 1982 IEEE symposium on security and privacy, Oakland, CA, USA, pp. 11–20. IEEE Computer Society, New York. https://doi.org/10.1109/SP.1982.10014
18. Padon O, Immerman N, Shoham S, Karbyshev A, Sagiv M (2016) Decidability of inferring inductive invariants. In: Proceeding of the 43rd Annual ACM SIGPLAN-SIGACT Symposium on Principles of programming languages, POPL 2016, pp. 217–231. ACM, New York. https://doi.org/10.1145/2837614.2837640
19. Feldman YMY, Padon O, Immerman N, Sagiv M, Shoham S (2019) Bounded quantifier instantiation for checking inductive invariants. Logic Method in Comput Sci **15**(3). https://doi.org/10.23638/LMCS-15(3:18)2019
20. Müller C, Seidl H (2020) Stratified guarded first-order transition systems. In: Pichardie, D., Sighireanu, M. (eds.) IN: static analysis - 27th international symposium, SAS 2020, virtual event, 2020, Proceedings, pp. 113–133. LNCS 12389, Springer, Heidelberg, Berlin. https://doi.org/10.1007/978-3-030-65474-0_6
21. Ackermann W (1935) Untersuchungen über das eliminationsproblem der mathematischen logik. Mathe Ann 110:390–413
22. Ranzato F (2020) Decidability and synthesis of abstract inductive invariants. In: Konnov, I., Kovács, L. (eds.) In: 31st international conference on concurrency theory, CONCUR 2020, September 1-4, 2020, Vienna, Austria (Virtual Conference). LIPIcs, vol. 171, pp. 30–13021. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, Dagstuhl, Germany. https://doi.org/10.4230/LIPIcs.CONCUR.2020.30
23. Gabbay DM, Schmidt R, Szalas A (2008) Second order quantifier elimination: foundations. Computational Aspects and Applications. College Publications, Oxford
24. Szalas A (1993) On the correspondence between modal and classical logic: an automated approach. J Log Comput 3(6):605–620
25. Börger E, Grädel E, Gurevich Y (1997) The classical decision problem. Perspectives in Mathematical Logic. Springer, Heidelberg, Berlin
26. Clarkson MR, Schneider FB (2010) Hyperproperties. J Comput Sec 18(6):1157–1210
27. Kovács M, Seidl H, Finkbeiner B (2013) Relational abstract interpretation for the verification of 2-hypersafety properties. In: Sadeghi, A.-R., Gligor, V.D., Yung, M. (eds.) 2013 ACM SIGSAC conference on computer and communications security, CCS'13, Berlin, Germany, pp. 211–222. ACM, New York. https://doi.org/10.1145/2508859.2516721
28. Barthe G, Crespo JM, Kunz C (2016) Product programs and relational program logics. J Log Algebra Method Program 85(5):847–859. https://doi.org/10.1016/j.jlamp.2016.05.004

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.