How to Win First-Order Safety Games *

Helmut Seidl[1], Christian Müller[1], and Bernd Finkbeiner[2]

[1] Technische Universität München
{seidl,christian.mueller}@in.tum.de
[2] CISPA, Saarland University
finkbeiner@cs.uni-saarland.de

Abstract. First-order (FO) transition systems have recently attracted attention for the verification of parametric systems such as network protocols, software-defined networks or multi-agent workflows like conference management systems. Functional correctness or noninterference of these systems have conveniently been formulated as safety or hypersafety properties, respectively. In this article, we take the step from verification to synthesis — tackling the question whether it is possible to automatically synthesize predicates to enforce safety or hypersafety properties like noninterference. For that, we generalize FO transition systems to FO safety games. For FO games with monadic predicates only, we provide a complete classification into decidable and undecidable cases. For games with non-monadic predicates, we concentrate on universal first-order invariants, since these are sufficient to express a large class of properties — for example noninterference. We identify a non-trivial sub-class where invariants can be proven inductive and FO winning strategies be effectively constructed. We also show how the extraction of weakest FO winning strategies can be reduced to SO quantifier elimination itself. We demonstrate the usefulness of our approach by automatically synthesizing nontrivial FO specifications of messages in a leader election protocol as well as for paper assignment in a conference management system to exclude unappreciated disclosure of reports.

Keywords: First Order Safety Games · Universal Invariants · First Order Logic · Second Order Quantifier Elimination

1 Introduction

Given a network of processes, can we synthesize the content of messages to be sent to elect a single leader? Given a conference management system, can we automatically synthesize a strategy for paper assignment so that no PC member is able to obtain illegitimate information about reports? Parametric systems like conference management systems can readily be formalized as first order (FO)

* The project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme under grant agreement No. 787367 (PaVeS).

transition systems where the attained states of agents are given as a FO structure, i.e., a finite set of relations. This approach was pioneered by abstract state machines (ASMs) [16], and has found many practical applications, for example in the verification of network protocols [28], software defined networks [3], and multi-agent workflows [13,14,24]. FO transition systems rely on *input* predicates to receive information from the environment such as network events, interconnection topologies, or decisions of agents. In addition to the externally provided inputs, there are also *internal* decisions that are made to ensure well-behaviour of the system. This separation of input predicates into these two groups turns the underlying transition system into a two-player *game*. In order to systematically explore possibilities of synthesizing message contents in protocols or strategies in workflows, we generalize FO transition systems to FO games.

Example 1. Figure 1 shows a slightly simplified version of the network leader election protocol from [28] turned into a FO game. The topology of the network,

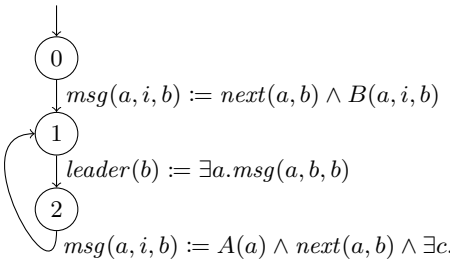

Fig. 1. FO safety game for the running leader election example

here a ring, is given by the predicates *next* and $\leq$, which are appropriately axiomatized. The participating agents communicate via messages through the predicate *msg* but are only allowed to send messages to the next agent in the ring topology. In the first step, agents can send any message (determined via the input predicate B) to their neighbor. Afterwards they check if they have received a message containing their own id. If so, they declare themselves leader and add themselves to the *leader* relation. Then, a subset of processes determined by the input predicate A decides to send any id to their next neighbor that they have received which is not exceeded by their own.

At no point more than one process should have declared itself leader — regardless of the size of the ring. This property is enforced, e.g., if the initial message to be sent is given by the id of the sending process itself, i.e., $B(a,i,b)$ is given by the literal $(i = a)$. □

Example 2. Consider the workflow of a conference management system as specified in fig. 2. The specification maintains the binary predicates *Conflict* and *Assign* together with the ternary predicates *Review* and *Read* to record conflicts of interest between PC members and papers, the paper assignment as well as the reports provided by PC members for papers. After the initial declaration of conflicts of interest, PC members write reviews for the papers they are assigned and

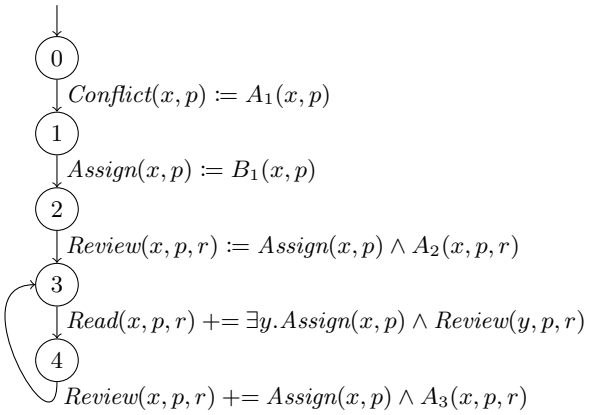

Fig. 2. FO safety game for the running conference management example

update them after reading the other reviews to the same paper. The predicates A_1, A_2, A_3 represent choices by PC members, while the predicate B_1 is under control of the PC chair. The operator $+=$ adds tuples to a relation instead of replacing all contents. Specifically, $R\bar{y} \mathrel{+}= \varphi$ abbreviates $R\bar{y} := R\bar{y} \vee \varphi$. □

One property to be checked in example 2 is that no PC member can learn anything about papers she has declared conflict with. *Noninterference* properties like this one can be formalized as *hyper-safety* properties, but can be reduced to *safety* properties of suitable *self-compositions* of the system in question [24]. This reduction is explained in appendix B. A plain safety property in this example would be, e.g., the more humble objective that no PC member x is going to read a report on a paper p which she herself has authored, i.e.,

$$\forall x, p, r. \neg(\mathit{Conflict}(x, p) \wedge \mathit{Read}(x, p, r))$$

Obvious choices for B_1 to enforce this property are

$$B_1(x, p) := \neg \mathit{Conflict}(x, p) \qquad \text{or}$$
$$B_1(x, p) := \mathit{false}$$

The second choice is rather trivial. The first choice, on the other hand, which happens to be the *weakest* possible, represents a meaningful strategy.

In this paper, we therefore investigate cases where *safety* is decidable and winning strategies for safety player are effectively computable and as weak as possible. For FO transition systems as specified by the Relational Modeling Language (RML) [28], typed update commands are restricted to preserve Bernays-Schönfinkel-Ramsey (also called *effectively propositional*) formulas. As a consequence, inductiveness of a universal invariant can be checked automatically. We show that this observation can be extended to FO safety *games* — given that appropriate winning strategies for safety player are either provided or can be effectively constructed (see section 5). We also provide sufficient conditions under which a *weakest* such strategy can be constructed (see section 6).

The question arises whether a similar transfer of the decidability of the logic to the decidability of the verification problem is possible for other decidable fragments of FO logic. A both natural and useful candidate is *monadic* logic. Interestingly, this transfer is only possible for specific *fragments* of monadic FO safety games, while in general safety is undecidable. For FO safety games using arbitrary predicates, we restrict ourselves to FO universal invariants only, since the safety properties, e.g., arising from noninterference can be expressed in this fragment. For universal invariants, we show how general methods for second order quantifier elimination can be instantiated to compute winning strategies. Existential SO quantifier elimination, though, is not always possible. Still, we provide a non-trivial class of universal invariants where optimal strategies can be synthesized. In the general case and, likewise, when existential FO quantifiers are introduced during game solving, we resort to *abstraction* as in [24]. This allows us to automatically construct strategies that guarantee safety or, in the case of information-flow, to enforce noninterference.

The paper is organized as follows. In sections 2 and 3, the notion of first-order safety games is introduced. We prove that safety player indeed has a positional winning strategy, whenever the game is safe. We also prove that safety of *finite* games is already inter-reducible to SO predicate logic. In section 4, we consider the important class of FO safety games where all predicates are either monadic or boolean flags. Despite the fact that this logic is decidable and admits SO quantifier elimination, safety for this class is undecidable. Nonetheless, we identify three subclasses of monadic games where decidability is retained. Section 5 proves that even when a universally quantified FO candidate for an inductive invariant of the safety game is already provided, checking whether or not the candidate invariant is inductive, can be reduced to SO existential quantifier elimination. Section 6 provides background techniques for SO universal as well as existential quantifier elimination. It proves that for universal FO formulas, the construction of a *weakest* SO Hilbert choice operator can be reduced to SO quantifier elimination itself. Moreover, it provides sufficient conditions when a universal invariant for a FO safety game can effectively proven inductive and a corresponding weakest strategy for safety player be extracted. Based on the candidates for the second-order Hilbert choice operator from section 6, and abstraction techniques from [24], a practical implementation is presented in section 7 which allows to infer inductive invariants and FO definable winning strategies for safety player. Finally, section 8 provides a more detailed comparison with related work while section 9 concludes.

2 First-Order Transition Systems

Assume that we are given finite sets $\mathcal{R}_{state}, \mathcal{R}_{input}, \mathcal{C}$ of relation symbols and constants, respectively. A first-order (FO) transition system $\mathcal{S}$ (over $\mathcal{R}_{state}$, $\mathcal{R}_{input}$ and $\mathcal{C}$) consists of a control-flow graph (V, E, v_0) underlying $\mathcal{S}$ where V is a finite set of program points, $v_0 \in V$ is the start point and E is a finite set of edges between vertices in V. Each edge thereby is of the form (v, θ, v') where θ sig-

nifies how the first-order structure for program point v' is determined in terms of a first-order structure at program point v. Thus, θ is defined as a mapping which provides for each predicate $R \in \mathcal{R}_{state}$ of arity r, a first-order formula $R\theta$ with free variables from $\mathcal{C}$ as well as a dedicated sequence of fresh FO variables $\bar{y} = y_1 \dots y_r$. Each formula $R\theta$ may use FO quantification, equality or disequality literals as well as predicates from $\mathcal{R}_{state}$. Additionally, we allow occurrences of dedicated *input* predicates from $\mathcal{R}_{input}$. For convenience, we denote a substitution θ of predicates $R_1, \dots, R_n$ with $\varphi_1, \dots, \varphi_n$ by

$$\theta = \{R_1\bar{y}_1 \coloneqq \varphi_1, \dots, R_n\bar{y}_n \coloneqq \varphi_n\}$$

where $\bar{y}_j = y_1 \dots y_{r_j}$ are the formal parameters of R_i and may occur free in φ_i.

Example 3. In the example from fig. 2, the state predicates in $\mathcal{R}_{state}$ are *Conflict*, *Assign*, *Review* and *Read*, while the input predicates $\mathcal{R}_{input}$ consist of A_1, A_2, A_3 and B_1. As there are no global constants, $\mathcal{C}$ is empty. For the edge from node 2 to node 3, θ maps *Review* to the formula $\mathit{Assign}(y_1, y_2) \wedge A_2(y_1, y_2, y_3)$ and each other predicate R from $\mathcal{R}_{state}$ to itself (applied to the appropriate list of formal parameters). Thus, θ maps, e.g., *Conflict* to $\mathit{Conflict}(y_1, y_2)$. □

Let U be some universe and $\rho : \mathcal{C} \to U$ be a valuation of the globally free variables. Let $\mathcal{R}^n_{state}$ denote the set of predicates with arity n. A *state* $s : \bigcup_{n\geq 0} \mathcal{R}^n_{state} \times U^n \to \mathbb{B}$ is an evaluation of the predicates $\mathcal{R}_{state}$ by means of relations over U. Let States_U denote the set of all states with universe U. For an edge (v, θ, v'), a valuation ω of the input predicates, and states s, s', there is a transition from (v, s) to (v', s') iff for each predicate $R \in \mathcal{R}_{state}$ of arity r together with a vector $\bar{y} = y_1 \dots y_r$ and an element $u \in U^r$

$$s', \rho \oplus \{y \mapsto u\} \models Ry \text{ iff } s \oplus \omega, \rho \oplus \{y \mapsto u\} \models (R\theta)$$

holds. Here, the operator "$\oplus$" is meant to update the assignment in the left argument with the variable/value pairs listed in the second argument. The set of all pairs $((v, s), (v', s'))$ constructed in this way, constitute the *transition relation* $\Delta_{U,\rho}$ of $\mathcal{S}$ (relative to universe U and valuation ρ). A finite *trace* from (v, s) to (v', s') is a finite sequence $(v_0, s_0), \dots, (v_n, s_n)$ with $(v, s) = (v_0, s_0)$ and $(v_n, s_n) = (v, s')$ such that for each $i = 0, \dots, n-1$, $((v_i, s_i), (v_{i+1}, s_{i+1})) \in \Delta_{U,\rho}$ holds. We denote the set of all finite traces of a transition system $\mathcal{S}$ as $\mathsf{Traces}(\mathcal{S})$.

Example 4. Let us instantiate the running example from fig. 2 for the universe $\{x_1, x_2, p_1, p_2, r_1\}$. A possible state attainable at node 2 could have $\mathit{Conflict} = \{(x_1, p_1)\}$, $\mathit{Assign} = \{(x_1, p_2), (x_2, p_1), (x_2, p_2)\}$ and all other relations empty. For the valuation $A_2 = \{(x_2, p_2, r_1)\}$ of the input predicate, there would be a transition to node 3 and a state where $\mathit{Review} = \{(x_2, p_2, r_1)\}$, with *Conflict* and *Assign* unchanged and *Read* still empty. □

3 First-Order Safety Games

For a first-order transition system, a FO *assertion* is a mapping I that assigns to each program point $v \in V$ a FO formula $I[v]$ with relation symbols from

$\mathcal{R}_{state}$ and free variables from $\mathcal{C}$. Assume that additionally we are given a FO formula Init (also with relation symbols from $\mathcal{R}_{state}$ and free variables from $\mathcal{C}$) describing the potential initial states. The assertion I *holds* if for all universes U, all valuations ρ, all states s with $s, \rho \models \mathsf{Init}$ and all finite traces τ from (v_0, s) to (v, s'), we have that $s', \rho \models I[v]$. In that case, we say that I is an *invariant* of the transition system (w.r.t. the initial condition Init).

Example 5. For our running example from fig. 2, the initial condition specifies that all relations R in $\mathcal{R}_{state}$ are empty, i.e., $\mathsf{Init} = \bigwedge_{R \in \mathcal{R}_{state}} \forall \bar{y}.\neg R\bar{y}$ where we assume that the length of the sequence of variables $\bar{y}$ matches the rank of the corresponding predicate R. Since the example assertion should hold everywhere, we have for every u, $I[u] = \forall x, p, r.\neg(\mathit{Conflict}(x, p) \wedge \mathit{Read}(x, p, r))$ □

We now generalize FO transition systems to *FO safety games*, i.e., 2-player games where reachability player $\mathcal{A}$ aims at violating the given assertion I while safety player $\mathcal{B}$ tries to establish I as an invariant. To do so, player $\mathcal{A}$ is able to choose the universe, which outgoing edges are chosen at a given node and all interpretations of relations under his control. Accordingly, we partition the set of input predicates $\mathcal{R}_{input}$ into subsets $\mathcal{R}_{\mathcal{A}}$ and $\mathcal{R}_{\mathcal{B}}$. While player $\mathcal{B}$ controls the valuation of the predicates in $\mathcal{R}_{\mathcal{B}}$, player $\mathcal{A}$ has control over the valuations of predicates in $\mathcal{R}_{\mathcal{A}}$ as well as over the universe and the valuation of the FO variables in $\mathcal{C}$. For notational convenience, we assume that each substitution θ in the control-flow graph contains at most one input predicate, and that all these are distinct [3]. Also we consider a partition of the set E of edges into the subsets $E_{\mathcal{A}}$ and $E_{\mathcal{B}}$ where the substitutions only at edges from $E_{\mathcal{B}}$ may use predicates from $\mathcal{R}_{\mathcal{B}}$. Edges in $E_{\mathcal{A}}$ or $E_{\mathcal{B}}$ will also be called $\mathcal{A}$-edges or $\mathcal{B}$-edges, respectively. For a particular universe U and valuation ρ, a trace τ starting in some (v_0, s) with $s, \rho \models \mathsf{Init}$ and ending in some pair (v, s') is considered a *play*. For a given play, player $\mathcal{A}$ *wins* iff $s', \rho \not\models I[v]$ and player $\mathcal{B}$ wins otherwise.

A *strategy* σ for player $\mathcal{B}$ is a mapping which for each $\mathcal{B}$-edge $e = (u, \theta, v)$ with input predicate B_e (of some arity r), each universe U, valuation ρ, each state s and each play τ reaching (u, s), returns a relation $B' \subseteq U^r$. Thus, σ provides for each universe, the history of the play and the next edge controlled by $\mathcal{B}$, a possible choice. σ is *positional* or *memoryless*, if it depends on the universe U, the valuation ρ, the state s and the $\mathcal{B}$-edge (u, θ, v) only.

A play τ *conforms* to a strategy σ for safety player $\mathcal{B}$, if all input relations at $\mathcal{B}$-edges occurring in τ are chosen according to σ. The strategy σ is *winning* for $\mathcal{B}$ if $\mathcal{B}$ wins all plays that conform to σ. An FO safety game can be won by $\mathcal{B}$ iff there exists a winning strategy for $\mathcal{B}$. In this case, the game is *safe*.

Example 6. In the running conference management example 2, player $\mathcal{A}$, who wants to reach a state where the invariant from example 5 is violated (a state

[3] In general, edges may use multiple input predicates of the same type. This can, however, always be simulated by a sequence of edges that stores the contents of the input relations in auxiliary predicates from $\mathcal{R}_{state}$ one by one, before realizing the substitution of the initial edge by means of the auxiliary predicates.

where someone reads a review to his own paper before the official release) has control over the predicates A_1, A_2, A_3 and thus provides the values for the predicates *Conflict* and *Review* and also determines how often the loop body is iterated. Player $\mathcal{B}$ only has control over predicate B_1 which is used to determine the value of predicate *Assign*. This particular game is *safe*, and player $\mathcal{B}$ has several winning strategies, e.g., $B_1(x,p) \coloneqq \neg \mathit{Conflict}(x,p)$. □

Lemma 1. *If there exists a winning strategy for player $\mathcal{B}$, then there also exists a winning strategy that is positional.*

Proof. Once a universe U is fixed, together with a valuation ρ of the globally free variables, the FO safety game G turns into a reachability game $G_{U,\rho}$ where the positions are given by all pairs $(v,s) \in V \times \mathsf{States}_U$ (controlled by reachability player $\mathcal{A}$) together with all pairs $(s,e) \in \mathsf{States}_U \times E$ controlled by safety player $\mathcal{B}$ if $e \in E_{\mathcal{B}}$ and by $\mathcal{A}$ otherwise. For an edge $e = (v,\theta,v')$ in G, $G_{U,\rho}$ contains all edges $(v,s) \to (s,e)$, together with all edges $(s,e) \to (v',s')$ where s' is a successor state of s w.r.t. e and ρ.

Let $\mathsf{Init}_{U,\rho}$ denote the set of all positions (v_0,s) where $s,\rho \models \mathsf{Init}$, and $I_{U,\rho}$ the set of all positions (v,s) where $s,\rho \models I[v]$ together with all positions (s,e) where $s,\rho \models I[v]$ for edges e starting in v. Then $G_{U,\rho}$ is safe iff safety player $\mathcal{B}$ has a strategy $\sigma_{U,\rho}$ to force each play started in some position $\mathsf{Init}_{U,\rho}$ to stay within the set $I_{U,\rho}$. Assuming the axiom of choice for set theory, the set of positions can be well-ordered. Therefore, the strategy $\sigma_{U,\rho}$ for safety player $\mathcal{B}$ can be chosen positionally, see, e.g., lemma 2.12 of [22]. Putting all positional strategies $\sigma_{U,\rho}$ for safety player $\mathcal{B}$ together, we obtain a single positional strategy for $\mathcal{B}$. □

In case the game is safe, we are interested in strategies that can be included into the FO transition system itself, i.e., are themselves first-order definable. Lemma 1 as is, gives no clue whether or not there is a winning strategy which is positional and can be expressed in FO logic, let alone be effectively computed.

Theorem 1. *There exist safe FO safety games where no winning strategy is expressible in FO logic.*

Proof. Consider a game with $\mathcal{R}_{state} = \{E, R_1, R_2\}$, $\mathcal{R}_{\mathcal{A}} = \{A_1, A_2\}$ and $\mathcal{R}_{\mathcal{B}} = \{B_1\}$, performing three steps in sequence:

$$E(x,y) \coloneqq A_1(x,y); \quad R_1(x,y) \coloneqq B_1(x,y); \quad R_2(x,y) \coloneqq A_2(x,y)$$

In this example, reachability player $\mathcal{A}$ chooses an arbitrary relation E, then safety player $\mathcal{B}$ chooses R_1 and player $\mathcal{A}$ chooses R_2. The assertion I ensures that at the endpoint R_1 is at least the transitive closure of E and R_1 is smaller or equal to R_2 (provided $\mathcal{A}$ chose R_2 to include the closure of E) i.e.,

$$\mathsf{closure}(R_2, E) \to (\mathsf{closure}(R_1, E) \wedge \forall x, y.(R_1(x,y) \to R_2(x,y)))$$

where $\mathsf{closure}(R,E)$ is given by $\forall x, y.R(x,y) \leftarrow (E(x,y) \ \vee \exists z.R(x,z) \wedge E(z,y))$. The only winning strategy for safety player $\mathcal{B}$ (choosing R_1) is to select the

smallest relation R_1 satisfying $closure(R_1, E)$, which is the transitive closure of E. In this case, no matter what reachability player $\mathcal{A}$ chooses for R_2, safety player $\mathcal{B}$ wins, but the winning strategy for $\mathcal{B}$ is not expressible in FO logic. □

Despite this negative result, effective means are sought for of computing FO definable strategies, whenever they exist. In order to do so, we rely on a *weakest precondition* operator $[\![e]\!]^\top$ corresponding to edge $e = (u, \theta, v)$ of the control-flow graph of a FO safety game $\mathcal{T}$ by

$$[\![e]\!]^\top \Psi = \begin{cases} \forall A_e.(\Psi\theta) \text{ if } e \ \mathcal{A}\text{-edge} \\ \exists B_e.(\Psi\theta) \text{ if } e \ \mathcal{B}\text{-edge} \end{cases}$$

The weakest pre-condition operator captures the minimal requirement at the start point of an edge to meet the post-condition Ψ at the end point. That operator allows to define the following iteration: Let $\mathcal{T}$ denote a game and I an assertion. For $h \geq 0$, let the assignment $\Psi^{(h)}$ of program points v to formulas be

$$\begin{array}{l} \Psi^{(0)}[v] = I[v] \\ \Psi^{(h)}[v] = \Psi^{(h-1)}[v] \ \wedge \ \bigwedge_{e \in out(v)} [\![e]\!]^\top \Psi^{(h-1)} \quad \text{for } h > 0 \end{array} \tag{1}$$

where $out(v)$ are the outgoing edges of node v. Then the following holds:

Theorem 2. *A FO safety game $\mathcal{T}$ is safe iff $\mathsf{Init} \to \Psi^{(h)}[v_0]$ holds for all $h \geq 0$.*

The proof can be found in appendix A. The characterization of safety due to theorem 2 is precise — but may require to construct infinitely many $\Psi^{(h)}$. Whenever, though, the safety game $\mathcal{T}$ is *finite*, i.e., the underlying control-flow graph of G is acyclic, then G is safe iff $\mathsf{Init} \to \Psi^{(h)}[v_0]$ where h equals the length of the longest path in the control-flow graph of G starting in v_0. As a result, we get that *finite* first order safety games are as powerful as second order logic.

Theorem 3. *Deciding a finite FO safety game with predicates from $\mathcal{R}_{state}$ is inter-reducible to satisfiability of SO formulas with predicates from $\mathcal{R}_{state}$.*

Proof. We already showed that solving a finite FO safety game can be achieved by solving the SO formula $\psi^{(h)}$ for some sufficiently large h. For the reverse implication, consider an arbitrary closed formula φ in SO Logic. W.l.o.g., assume that φ has no function symbols and is in prenex normal form where no SO Quantifier falls into the scope of a FO quantifier [20]. Thus, φ is of the form $Q_1 C_1 \ldots Q_n C_n.\ \psi$ where all Q_n are SO quantifiers and ψ is a relational formula in FO logic.

We then construct a FO safety game $\mathcal{T}$ as follows. The set $\mathcal{R}_{state}$ of predicates consists of all predicates that occur freely in φ together with copies R'_i of all quantified relations C_i. The control-flow graph consists of $n+1$ nodes $v_0, \ldots, v_n$, together with edges (v_{i-1}, θ_i, v_i) for $i = 1, \ldots, n$. Thus, the maximal length of any path is exactly n. An edge $e = (v_{i-1}, \theta_i, v_i)$ is used to simulate the quantifier $Q_i C_i$. The substitution θ_i is the identity on all predicates from $\mathcal{R}_{state}$ except R'_i which is mapped to C_i. If Q_i is a universal quantifier, C_i is included into $\mathcal{R}_{\mathcal{A}}$,

and e is an $\mathcal{A}$-edge. Similarly, if Q_i is existential, C_i is included into $\mathcal{R}_\mathcal{B}$ and e is a $\mathcal{B}$-edge. Assume that ψ' is obtained from ψ by replacing every relation R_i with R'_i. As FO assertion I, we then use $I[v_i] = \mathit{true}$ for $i = 0, \ldots, n-1$ and $I[v_n] = \psi'$. Then $\Psi^{(n)}[v_n] = \varphi$. Accordingly for $\mathsf{Init} = \mathit{true}$, player $\mathcal{B}$ can win the game iff φ is universally true. □

Theorem 3 implies that a FO definable winning strategy for safety player $\mathcal{B}$ (if it exists) can be constructed whenever the SO quantifiers introduced by the choices of the respective players can be eliminated. Theorem 3, though, gives no clue *how* to decide whether or not safety player $\mathcal{B}$ has a winning strategy and if so, whether it can be effectively represented.

4 Monadic FO Safety Games

Assuming that the universe is finite and bounded in size by some $h \geq 0$, then FO games reduce to finite games (of tremendous size, though). This means that, at least in principle, both checking of invariants as well as the construction of a winning strategy (in case that the game is safe) is effectively possible. A more complicated scenario arises when the universe consists of several disjoints *sorts* of which some are bounded in size and some are unbounded.

We will now consider the special case where each predicate has at most one argument which takes elements of an unbounded sort. In the conference management example, we could, e.g., assume that PC members, papers and reports constitute disjoint sorts of bounded cardinalities, while the number of (versions of) reviews is unbounded. By encoding the tuples of elements of finite sorts into predicate names, we obtain FO games where all predicates are either nullary or *monadic*. Monadic FO logic is remarkable since satisfiability of formulas in that logic is decidable, and monadic SO quantifiers can be effectively eliminated [4,35]. Due to theorem 3, we therefore conclude for *finite* monadic safety games that safety is decidable. Moreover, in case the game is safe, a positional winning strategy for safety player $\mathcal{B}$ can be effectively computed.

Monadic safety games which are not finite, turn out to be very close in expressive power to *multi-counter machines*, for which reachability is undecidable [18,30]. The first statement of the following theorem has been communicated to us by Igor Walukiewicz:

Theorem 4. *For monadic safety games, safety is undecidable when one of the following conditions is met:*

1. *there are both $\mathcal{A}$-edges as well as $\mathcal{B}$-edges;*
2. *there are $\mathcal{A}$-edges and substitutions with equalities or disequalities;*
3. *there are $\mathcal{B}$-edges and substitutions with equalities or disequalities.*

The proof of statement (1) is by using monadic predicates to simulate the counters of a multi-counter machine. Statements (2) and (3) follow from the observation that one player in this simulation can be replaced by substitutions using equality or disequality literals (see appendix C for details of the simulation).

There are, though, interesting cases that do not fall into the listed classes and can be effectively decided. Let us first consider monadic safety games where no predicate is under the control of either player, i.e., $\mathcal{R}_\mathcal{A} = \mathcal{R}_\mathcal{B} = \emptyset$, but both equalities and disequalities are allowed. Then, safety of the game collapses to the question if player $\mathcal{A}$ can pick universe and control-flow such that the assertion is violated at some point. For this case, we show that the conjunction of preconditions from section 3 necessarily stabilizes.

Theorem 5. *Assume that $\mathcal{T}$ is a monadic safety game, possibly containing equalities and/or disequalities with $\mathcal{R}_\mathcal{A} = \mathcal{R}_\mathcal{B} = \emptyset$. Then for some $h \geq 0$, $\Psi^{(h)} = \Psi^{(h+1)}$. Therefore, safety of $\mathcal{T}$ is decidable.*

Theorem 5 relies on the observation that when applying substitutions alone, i.e., without additional SO quantification, the number of equalities and disequalities involving FO variables, remains bounded. Our proof relies on variants of the *counting quantifier normal form* for monadic FO formulas [4] (see appendix D).

Interestingly, decidability is also retained for assertions I that only contain disequalities, if no equalities between bound variables are introduced during the weakest precondition computation. This can only be guaranteed if safety player $\mathcal{B}$ does not have control over any predicates. [4]

Theorem 6. *Assume that $\mathcal{T}$ is a monadic safety game without $\mathcal{B}$-edges (i.e. $\mathcal{R}_\mathcal{B} = \emptyset$) and*

1. *there are no disequalities between bound variables in I,*
2. *in all literals $x = y$ or $x \neq y$ in* Init *and substitutions θ, $x \in \mathcal{C}$ or $y \in \mathcal{C}$.*

Then it is decidable whether $\mathcal{T}$ is safe.

The proof is based on the following observation: Assume that $\mathcal{C}$ is a set of variables of cardinality d, and formulas φ_1, φ_2 have free variables only from $\mathcal{C}$. If φ_1, φ_2 contain no disequalities between bound variables, then φ_1, φ_2 are equivalent for all models and all valuations ρ iff they are equivalent for models and valuations with *multiplicity* exceeding d. Here, the *multiplicity* $\mu(s)$ of a model s is the minimal cardinality of a non-empty equivalence class of U w.r.t. *indistinguishability*. We call two elements u, u' of the universe U indistinguishable in a model s iff $(s, \{x \mapsto u\} \models Rx) \leftrightarrow (s, \{x \mapsto u'\} \models Rx)$ for all relations R. Then, when computing $\Psi^{(h)}$, we use an *abstraction* by formulas without equalities, which is shown to be a *weakest strengthening* (see appendix E).

Analogously, decidability is retained for assertions that only contain positive equalities if there are no disequalities introduced during the weakest precondition computation. This is only the case when $\mathcal{R}_\mathcal{A} = \emptyset$, i.e., reachability player $\mathcal{A}$ only selects universe and control-flow path. As a consequence, we obtain:

Theorem 7. *Assume that $\mathcal{T}$ is a monadic safety game without $\mathcal{A}$-edges where*

[4] The simulation in appendix C shows how predicates under the control of player $\mathcal{B}$ can be used to introduce equalities through SO existential quantifier elimination.

1. *there are no equalities between bound variables in* I,
2. *in all literals* $x = y, x \neq y$ *in* Init *and substitutions* θ, *either* $x \in \mathcal{C}$ *or* $y \in \mathcal{C}$.

Then it is decidable whether $\mathcal{T}$ *is safe.*

The proof is analogous to the proof of theorem 6 where the abstraction of equalities now is replaced with an abstraction of disequalities (see appendix F). In summary, we have shown that even though monadic logic is decidable, 2-player monadic FO safety games are undecidable in general. However, for games where one of the players does not choose interpretations for any relation, decidability can be salvaged if the safety condition has acceptable equality/disequality literals only and neither Init, nor the transition relation introduce further equality/disequality literals between bound variables.

5 Proving Invariants Inductive

Even though the general problem of verification is already hard for monadic FO games, there are useful incomplete algorithms to still prove general FO safety games safe. One approach for verifying infinite state systems is to come up with a candidate invariant which then is proven *inductive* (see, e.g., [28]). This idea can be extended to *safety games* where, additionally strategies must either be provided or extracted.

In the context of FO safety games, an invariant Ψ is called *inductive* iff for all edges $e = (u, \theta, v)$, $\Psi[u] \to [\![e]\!]^{\top}(\Psi[v])$ holds. We have:

Lemma 2. *Assume that* Ψ *is inductive, and* $\Psi[v] \to I[v]$ *for all nodes* v. *Then*

1. *For all* $h \geq 0$, $\Psi[v] \to \Psi^{(h)}[v]$;
2. *The game* G *is safe, whenever* $\mathsf{Init} \to \Psi[v_0]$ *holds.*

We remark that, under the assumptions of lemma 2, a *positional* winning strategy σ for safety player $\mathcal{B}$ *exists*. Checking an FO safety game $\mathcal{T}$ for safety thus boils down to the following tasks:

1. Come up with a candidate invariant Ψ so that
   - $\Psi[v] \to I[v]$ for all nodes v, and
   - $\mathsf{Init} \to \Psi[v_0]$ hold;
2. Come up with a strategy σ which assigns some FO formula to each predicate in $\mathcal{R}_{\mathcal{B}}$;
3. Prove that Ψ is inductive for the FO transition system $\mathcal{T}\sigma$ which is obtained from $\mathcal{T}$ by substituting each occurrence of B with $\sigma(B)$ for all $B \in \mathcal{R}_{\mathcal{B}}$.

For monadic FO safety games, we thereby obtain:

Theorem 8. *Assume that* $\mathcal{T}$ *is a monadic FO safety game with initial condition* Init *and assertion* I. *Assume further that* Ψ *is a monadic FO invariant, i.e., maps each program point to a monadic formula. Then the following holds:*

1. *It is decidable whether* $\mathsf{Init} \to \Psi[v_0]$ *as well as* $\Psi[v] \to I[v]$ *holds for each program point* v*;*
2. *It is decidable whether* Ψ *is inductive, and if so, an FO definable strategy* σ *can be constructed which upholds* Ψ*.*

The proof is by showing that all formulas fall into a decidable fragment — in this case Monadic Second Order logic. A *monadic* FO safety game can thus be proven safe by providing an appropriate monadic FO invariant Ψ: the winning strategy itself can be effectively computed.

Another important instance is when the candidate invariant Ψ as well as I consists of universal FO formulas only, while Init is in the *Bernays-Schönfinkel-Ramsey* (BSR) fragment[5].

Theorem 9. *Let* $\mathcal{T}$ *denote a safety game where each substitution* θ *occurring at edges of the control-flow graph uses non-nested FO quantifiers only. Let* Ψ *denote a universal FO invariant for* $\mathcal{T}$*, i.e.,* $\Psi[v]$ *is a universal FO formula for each node* v*.*

1. *It is decidable whether* $\mathsf{Init} \to \Psi[v_0]$ *as well as* $\Psi[v] \to I[v]$ *holds for each program point* v*;*
2. *Assume that no* $B \in \mathcal{R}_{\mathcal{B}}$ *occurs in the scope of an existential FO quantifier, and* σ *is a strategy which provides a universal FO formula for each* $B \in \mathcal{R}_{\mathcal{B}}$*. Then it is decidable whether or not* Ψ *is inductive for* $\mathcal{T}\sigma$*.*

The proof is by showing that all mentioned formulas can be solved by checking satisfiability of a formula in the decidable fragment $\exists^*\forall^*$FOL. Theorem 9 states that (under mild restrictions on the substitutions occurring at $\mathcal{B}$-edges), the candidate invariant Ψ can be checked for inductiveness — at least when a positional strategy of $\mathcal{B}$ is provided which is expressed by means of universal FO formulas. In particular, this implies decidability for the case when the set $E_{\mathcal{B}}$ is empty. The proof works by showing that all verification conditions fall into the BSR fragment of FO Logic. For the verification of inductive invariants for FO transition systems (no $\mathcal{B}$ edges), the Ivy system essentially relies on the observations summarized in theorem 9 [28].

Besides finding promising strategies σ, the question remains how for a given assertion I a suitable *inductive* invariant can be inferred. One option is to iteratively compute the sequence $\Psi^{(h)}, h \geq 0$ as in (1). In general that iteration may never reach a fixpoint. Here, however, FO definability implies termination:

Theorem 10. *Assume that for all program points* u *and* $h \geq 0$*,* $\Psi^{(h)}[u]$ *is FO definable as well as the infinite conjunction* $\bigwedge_{h\geq 0} \Psi^{(h)}[u]$*. Then there exists some* $m \geq 0$ *such that* $\Psi^{(m)} = \Psi^{(m+k)}$ *holds for each* $k \geq 0$*. Thus,* $\bigwedge_{h\geq 0} \Psi^{(h)}[u] = \Psi^{(m)}[u]$ *for all* u*.*

[5] The Bernays-Schönfinkel-Ramsey fragment contains all formulas of First Order Logic that have a quantifier prefix of $\exists^*\forall^*$ and do not contain function symbols. Satisfiability of formulas in BSR is known to be decidable [29].

Proof. Let φ_u denote the first order formula which is equivalent $\bigwedge_{h\geq 0} \Psi^{(h)}[u]$. In particular, this means that $\varphi_u \to \Psi^{(h)}[u]$ for each $h \geq 0$. On the other hand, we know that $\bigwedge_{h\geq 0} \Psi^{(h)}[u]$ implies φ_u. Since φ_u as well as each $\Psi^{(h)}[u]$ are assumed to be FO definable, it follows from Gödel's compactness theorem that there is a *finite* subset $J \subseteq \mathcal{N}$ such that $\bigwedge_{h\in J} \Psi^{(h)}[u]$ implies φ_u. Let m be the maximal element in J. Then, $\bigwedge_{h\in J} \Psi^{(h)}[u] = \Psi^{(m)}[u]$ since the $\Psi^{(h)}[u]$ form a decreasing sequence of formulas. Together, this proves that φ_u is equivalent to $\Psi^{(m)}[u]$. □

Theorem 10 proves that if there exists an inductive invariant proving a given FO game safe, then fixpoint iteration will definitely terminate and find it.

For the case of *monadic* FO safety games, this means that the corresponding infinite conjunction is not always FO definable — otherwise decidability would follow. In general, not every invariant I can be strengthened to an inductive Ψ, and universal strategies need not be sufficient to win a universal safety game. Nonetheless, there is a variety of non-trivial cases where existential SO quantifiers can be effectively eliminated, e.g., by Second Order quantifier elimination algorithms SCAN or DLS* (see the overview in [15]). In our case, in addition to plain elimination we need an explicit construction of the corresponding strategy, expressed as a FO formula. We remark that following theorem 9, it is not necessary to perform *exact* quantifier elimination: instead, a sufficiently weak *strengthening* may suffice. Techniques for such *approximate* SO existential quantifier elimination are provided in the next section.

6 Hilbert's Choice Operator for Second Order Quantifiers

In this section, we concentrate on formulas with universal FO quantifiers only. First, we recall the following observation:

Lemma 3 (see Fact 1, [24]). *Consider a disjunction c of the form*

$$F \vee \bigvee_{i=1}^{k} A\bar{z}_i \vee \bigvee_{j=1}^{l} \neg A\bar{z}'_i$$

for some formula F without occurrences of predicate A. Then $\forall A.c$ is equivalent to $F \vee \bigvee_{i,j}(\bar{z}_i = \bar{z}'_j)$ for sequences of variables $\bar{z}_i = z_{i1} \dots z_{ir}$, $\bar{z}'_j = z'_{j1} \dots z'_{jr}$, where $\bar{z}_i = \bar{z}_j$ is an abbreviation for $\bigwedge_{k=1}^{r} z_{ik} = z'_{jk}$. □

As a consequence, universal SO quantification can always be eliminated from universal formulas.

Example 7. Consider the assertion $I = \forall x, p, r.\neg(\mathit{Conflict}(x,p) \wedge \mathit{Review}(x,p,r))$ and substitution θ from the edge between program points 2 and 3 in fig. 2, given by $\mathit{Review}(x,p,r) \coloneqq \mathit{Assign}(x,p) \wedge A_3(x,p,r)$ and $\mathit{Conflict}(x,p) \coloneqq \mathit{Conflict}(x,p)$. Since $I\theta$ contains only negative occurrences of A_3, we obtain:

$$\begin{aligned} \forall A_3.(I\theta) &= \forall x, p, r. \forall A_3. \neg \mathit{Conflict}(x,p) \vee \neg \mathit{Assign}(x,p) \vee \neg A_3(x,p,r) \\ &= \forall x, p, r. \neg \mathit{Conflict}(x,p) \vee \neg \mathit{Assign}(x,p) \end{aligned}$$

□

As we have seen in section 5, checking whether a universal FO invariant is inductive can be reduced to SO existential quantifier elimination. While universal SO quantifiers can always be eliminated in formulas with universal FO quantifiers only, this is not necessarily the case for existential SO quantifiers. As already observed by Ackermann [1], the formula

$$\exists B.\, Ba \wedge \neg Bb \wedge \forall x, y. \neg Bx \vee \neg Rxy \vee By$$

expresses that b is not reachable from a via the edge relation R and thus cannot be expressed in FO logic. This negative result, though, does not exclude that in a variety of meaningful cases, equivalent FO formulas can be constructed. For formulas with universal FO quantifiers only, we provide a simplified algorithm for existential SO quantifier elimination. Moreover, we show that the construction of a *weakest* SO *Hilbert choice operator* can be reduced to existential SO quantifier elimination itself. In terms of FO safety games, the latter operator enables us to extract *weakest* winning strategies for safety player $\mathcal{B}$. For an in-depth treatment on SO existential quantifier elimination, we refer to [15].

Let φ denote some universally quantified formula, possibly containing a predicate B of arity r. Let $\bar{y} = y_1 \dots y_r$, and $\bar{y}' = y'_1 \dots y'_r$. We remark that for *any* formula ψ with free variables in y, $\varphi[\psi/B] \rightarrow \exists B.\varphi$ holds. Here, this SO substitution means that every literal $B\bar{z}$ and every literal $\neg B\bar{z}'$ is replaced with $\psi[\bar{z}/\bar{y}]$ and $\neg\psi[\bar{z}'/\bar{y}]$, respectively. Let $H_{B,\varphi}$ denote the set of *all* FO formulas ψ such that $\exists B.\varphi$ is equivalent to $\varphi[\psi/B]$. A general construction for B and φ (at least from some suitably restricted class of formulas) of some FO formula $\psi \in H_{B,\varphi}$ is an instance of Hilbert's (second-order) *choice* operator. If it exists, we write $\psi = \mathcal{H}_{\mathcal{B}}(\varphi)$. In order to better understand the construction of such operators, we prefer to consider universal FO formulas in *normal form*.

Lemma 4. *Every universal FO formula φ possibly containing occurrences of B is equivalent to a formula*

$$E \wedge (\forall \bar{y}.F \vee B\bar{y}) \wedge (\forall \bar{y}'.G \vee \neg B\bar{y}') \wedge (\forall \bar{y}\bar{y}'.H \vee B\bar{y} \vee \neg B\bar{y}') \tag{2}$$

where E, F, G, H are universal formulas without B.

The corresponding construction is provided in appendix G. For that, disequalities between variables, and fresh auxiliary variables $\bar{y}$ and $\bar{y}'$ are introduced, where the sequence $\bar{y}'$ is only required when both positive and negative B literals occur within the same clause. In case these are missing, the formula is said to be in *simple* normal form. For that case, Ackermann's lemma applies:

Lemma 5 (Ackermann's lemma [1]). *Assume that φ is in simple normal form $E \wedge (\forall \bar{y}.F \vee B\bar{y}) \wedge (\forall \bar{y}.G \vee \neg B\bar{y})$. Then we have:*

1. *$\exists B.\varphi = E \wedge (\forall \bar{y}.F \vee G)$;*
2. *For every FO formula ψ, $\exists B.\varphi = \varphi[\psi/B]$ iff $(E \wedge \neg F) \rightarrow \psi$ and $\psi \rightarrow (\neg E \vee G)$.* □

For formulas in simple normal form a Hilbert choice operator thus is given by:

$$\mathcal{H}_B\varphi = \neg E \vee G \tag{3}$$

— which is the *weakest* ψ for which $\exists B.\varphi$ is equivalent to $\varphi[\psi/B]$.

Example 8. For the invariant from example 5, the weakest precondition w.r.t. the second statement amounts to: $\exists B_1.\forall x, p.\neg \mathit{Conflict}(x,p) \vee \neg B_1(x,p)$ which is *true* for any formula Ψ for B_1 (with free x, p) implying $\neg \mathit{Conflict}(x,p)$. □

The *strongest* solution according to example 8 thus is that the PC chair decides to assign papers to *no* PC member. While guaranteeing safety, this choice is not very useful. The *weakest* choice on the other hand, provides us here with a decent strategy. In the following we therefore will aim at constructing as *weak* strategies as possible.

Ackermann's Lemma gives rise to a nontrivial class of safety games where existential SO quantifier elimination succeeds. We call $B \in \mathcal{R}_{\mathcal{B}}$ *ackermannian* in the substitution θ iff for every predicate $R \in \mathcal{R}_{state}$, if $\theta(R)$ contains B literals, $\theta(R)$ is quantifierfree and its CNF does not contain clauses with both positive and negative B literals.

Theorem 11. *Assume we are given a FO Safety Game $\mathcal{T}$ where all substitutions contain nonnested quantifiers only, and a universal inductive invariant Ψ. Assume further that the following holds:*

1. *All predicates B under the control of safety player $\mathcal{B}$ are ackermannian in all substitutions θ;*
2. *For every $\mathcal{B}$-edge $e = (u, \theta, v)$, every clause of $\Psi[v]$ contains at most one literal with a predicate R where $\theta(R)$ has a predicate from $\mathcal{R}_{\mathcal{B}}$.*

Then the weakest FO strategy for safety player $\mathcal{B}$ can be effectively computed for which Ψ is inductive.

Proof. Consider an edge (u, θ, v) where the predicate B under control of safety player occurs in θ. Assume that $\Psi[v] = \forall \bar{z}.\Psi'$ where Ψ' is quantifierfree and in conjunctive normal form. Since θ is ackermannian and due to the restrictions given for Ψ', Ψ' can be written as $\Psi' = \Psi_0 \wedge \Psi_1 \wedge \Psi_2$ where Ψ_0, Ψ_1, Ψ_2 are the conjunctions of clauses c of Ψ' where $\theta(c)$ contains none, only positive or only negative occurrences of B-literals, respectively. In particular, $\theta(\Psi_0)$ is a FO formula without nested quantifiers. The formula $\theta(\Psi_1)$ is equivalent to a conjunction of formulas of the form $F \vee B\bar{y}_1 \vee \ldots B\bar{y}_r$ where F has non-nested quantifiers only, which thus are equivalent to

$$\forall \bar{y}.F \vee (\bar{y}_1 \neq \bar{y}) \wedge \ldots \wedge (\bar{y}_r \neq \bar{y}) \vee B\bar{y}$$

Likewise, $\theta(\Psi_2)$ is equivalent to a conjunction of formulas of the form $G \vee \neg B\bar{y}_1 \vee \ldots \neg B\bar{y}_r$ where G has non-nested quantifiers only, which thus are equivalent to

$$\forall \bar{y}.G \vee (\bar{y}_1 \neq \bar{y}) \wedge \ldots \wedge (\bar{y}_r \neq \bar{y}) \vee \neg B\bar{y}$$

Therefore, we can apply Ackermann's lemma to obtain a formula $\bar{\Psi}$ in $\forall^*\exists^*$FOL equivalent to $\exists B.\, \theta(\Psi[v])$. Likewise, we obtain a weakest FO formula φ for B so that $\theta(\Psi[v])[\varphi/B] = \bar{\Psi}$. Since $\Psi[u]$ only contains universal quantifiers, $\Psi[u] \to \bar{\Psi}$ is effectively decidable. □

Example 9. Consider the leader election protocol from example 1, together with the invariant from [28]. Therein, the predicate B is ackermannian, and *msg* appears once in two different clauses of the invariant. Thus by theorem 11, the weakest safe strategy for player $\mathcal{B}$ can be effectively computed. Our solver, described in section 7 finds it to be

$$B(a,i,b) \coloneqq \neg E \vee \neg \mathit{next}(a,b) \vee \begin{pmatrix} \forall n.\ (i \geq n \vee b \neq i) \wedge \\ \forall n.\ (\neg \mathit{between}(b,i,n) \vee i > n) \end{pmatrix}$$

where E axiomatizes the ring architecture, i.e., the predicate *between* as the transitive closure of *next* together with the predicate $\leq$. The given strategy is weaker than the intuitive (and also safe) strategy of $(i = a)$, and allows for more behaviours — for example a can send messages that are greater than its own id in case they are not greater than the ids of nodes along the way from b back to a. □

In general, though, existential SO quantifier elimination must be applied to universally quantified formulas which cannot be brought into simple normal form. In particular, we provide a sequence of *candidates* for the Hilbert choice operator which provides the *weakest* Hilbert choice operator — whenever it is FO definable. Consider a formula φ in normal form (2). Therein, the sub-formula $\neg H$ can be understood as a binary predicate between the variables $\bar{y}'$ and $\bar{y}$ which may be composed, iterated, post-applied to predicates on $\bar{y}'$ and pre-applied to predicates on $\bar{y}$. We define H^k, $k \geq 0$, with free variables from $\bar{y}, \bar{y}'$ by

$$\begin{aligned} H^0 &= \bar{y} \neq \bar{y}' \\ H^k &= \forall \bar{y}_1.H^{k-1}[\bar{y}_1/\bar{y}'] \vee H[\bar{y}_1/\bar{y}] \qquad \text{for } k > 0 \end{aligned}$$

We remark that by this definition,

$$H^{k+l} = \forall y_1.H^k[\bar{y}_1/\bar{y}'] \vee H^l[\bar{y}_1/\bar{y}]$$

for all $k, l \geq 0$. Furthermore, we define the formulas:

$$\begin{aligned} G \circ H^k &= \forall \bar{y}.G[\bar{y}/\bar{y}'] \vee H^k \\ G \circ H^k \circ F &= \forall \bar{y}'.(G \circ H^k) \vee F[\bar{y}'/\bar{y}] \end{aligned}$$

Then, we have:

Lemma 6. *If $\exists B.\varphi$ is FO definable, then it is equivalent to $E \wedge \bigwedge_{i=0}^{k} G \circ H^i \circ F$ for some $k \geq 0$.*

Starting from G and iteratively composing with H, provides us with a sequence of candidate SO Hilbert choice operators. Let

$$\gamma_k = \neg E \vee \textstyle\bigwedge_{i=0}^{k} (G \circ H^i)[\bar{y}/\bar{y}'] \tag{4}$$

for $k \geq 0$. The candidate γ_k takes all $i$fold compositions of H with $i \leq k$ into account. Then the following holds:

Lemma 7. *For every $k \geq 0$,*

1. *$\varphi[\gamma_k/B]$ implies $\exists B.\varphi$;*
2. *If $\exists B.\varphi$ is equivalent to $\varphi[\psi/B]$ for some FO formula ψ, then $\psi \to \gamma_k$.*
3. *$\gamma_{k+1} \to \gamma_k$, and if $\gamma_k \to \gamma_{k+1}$, then $\varphi[\gamma_k/B] = \exists B.\varphi$.*

As a result, the γ_k form a decreasing sequence of candidate strategies for safety player $\mathcal{B}$. We remark that due to statement (2), the sequence γ_k results in the *weakest* Hilbert choice operator — whenever it becomes stable.
We close this section by noting that there is a SO Hilbert choice operator which can be expressed in SO logic itself. The following theorem is related to Corollary 6.20 of [15], but avoids the explicit use of fixpoint operators in the logic.

Theorem 12. *The weakest Hilbert choice operator $\mathcal{H}_B\varphi$ for the universal formula* (2) *is definable by the SO formula:*

$$\neg E \vee \exists B.B\bar{y} \wedge (\forall \bar{y}'.G \vee \neg B\bar{y}') \wedge (\forall \bar{y}\bar{y}'.H \vee B\bar{y} \vee \neg B\bar{y}')$$

The weakest Hilbert choice operator itself can thus be obtained by SO existential quantifier elimination. The proof is by rewriting the formula and can be found in appendix H.

7 Implementation

We have extended our solver NIWO for FO transition systems [24] to a solver for FO safety games which is able to verify inductive universal FO invariants and extract corresponding winning strategies for safety player $\mathcal{B}$. It has been packaged and published under [32]. Our solver supports *inference* of inductive invariants if the given candidate invariant is not yet inductive. For that it relies on the abstraction techniques from [24] to strengthen arbitrary FO formulas by means of FO formulas using universal FO quantifiers only. For the simplification of FO formulas as well as for satisfiability of BSR formulas, it relies on the EPR algorithms of the automated theorem prover Z3 [10].

We evaluate our solver on three kinds of benchmark problems. First, we consider FO games with safety properties such as the running example "Conference, Safety" from fig. 2 and example 5. For all of its variants, the fixpoint iteration (1) terminates in less than one second with a weakest winning strategy (w.r.t. the found inductive invariant) whenever possible. The second group "Leader Election" considers variants of the leader election protocol from example 1, initially taken from [28]. Since the inductive invariant implies some transitively closed property, it cannot be inferred automatically by our means. Yet, our solver succeeds in proving the invariant from [28] inductive, and moreover, infers a FO definition for the message to be forwarded to arrive at a single leader. The third group "Conference, NI" deals with noninterference for variants of the conference management example where the acyclic version has been obtained by unrolling the loop twice. The difference between the *stubborn* and *causal* settings is the

Name	Mode	Size	Invariant	#Str.	Max. inv.	Time
Conference, Safety	synthesis	6	inferred	4	50	736 ms
Leader Election	verification	4	inductive	0	42	351 ms
Leader Election	synthesis	4	inductive	0	42	346 ms
Conference, NI, stubborn	verification	6	inferred	4	850	6782 ms
Conference, NI, stubborn	synthesis	6	inferred	4	850	6817 ms
Conference acyclic, NI, causal	synthesis	8	inferred	4	137	1985 ms
Conference, NI, causal	verification	11	counterex.	7	-	2114 ms
Conference, NI, causal	synthesis	11	inferred	2	102	2460 ms
Conference, NI, causal, approx.	synthesis	11	inferred	8	5090	3359 ms

Fig. 3. Experimental Results

considered angle of attack (see [14] for an in-depth explanation). In the setting of stubborn agents, the attackers try to break the Noninterference property with no specific intent of working together. Here, the solver infers inductive invariants together with winning FO strategies (where possible) in $5-7$ seconds. The setting of causal agents is inherently more complex as it allows for groups of unbounded size that are working together to extract secrets from the system. This allows for elaborate attacks where multiple agents conspire to defeat noninterference [13]. The weakest strategy that is safe for stubborn agents ($\neg \mathit{Conflict}(x, p)$ as a strategy for B_1) can can no longer be proven correct — instead the solver finds a counterexample for universes of size ≥ 5. To infer an inductive invariant and a safe strategy for causal agents, multiple iterates of the fixpoint iteration from section 3 must be computed. Each iteration requires formulas to be brought into conjunctive normal form — possibly increasing formula size drastically. To cope with that increase, formula simplification turns out not to be sufficient. We try two different approaches to overcome this challenge: First, we provide the solver with parts of the inductive invariant, so fewer strengthening steps are needed. Given the initial direction, inference terminates much faster and provides us with a useful strategy. For the second approach, we do not supply an initial invariant, but accelerate fixpoint iteration by further strengthening of formulas. This enforces termination while still verifying safety. The extracted strategy, though, is much stronger and essentially rules out all intended behaviours of the system.

All benchmarks were run on a workstation running Debian Linux on an Intel i7-3820 clocked at 3.60 GHz with 15.7 GiB of RAM. The results are summarized in fig. 3. The table gives the group and type of experiment as well as the size of the transition system in the number of nodes of the graph. For the examples that regard Noninterference, the agent model is given. For verification benchmarks, the solver either proves the given invariant inductive or infers an inductive invariant if the property is not yet inductive. For synthesis benchmarks, it additionally extracts a universal formula for each $B \in \mathcal{R}_{\mathcal{B}}$ to be used as a strategy. The remaining columns give the results of the solver: Could the given invariant be proven inductive, could an inductive strengthening be found, or did the solver find a counterexample violating the invariant? We list the number of times any label of the invariant needed to be strengthened during the inference algorithm, the size of the largest label formula of the inferred invariant mea-

sured in the number of nodes of the syntax tree and the time the solver needed in milliseconds (averaged over 10 runs).

Altogether, the experiments confirm that verification of provided invariants as well as synthesis of inductive invariants and winning strategies is possible for nontrivial transition systems with safety as well as noninterference objectives.

8 Related Work

In AI, First Order Logic has a long tradition for representing potentially changing states of the environment [8]. First-order transition systems have then been used to model reachability problems that arise in robot planning (see, e.g., chapters 8-10 in [31]). The system GOLOG [21], for instance, is a programming language based on FO logic. A GOLOG program specifies the behavior of the agents in the system. The program is then evaluated with a theorem prover, and thus assertions about the program can be checked for validity. Automated synthesis of predicates to enforce safety of the resulting system has not yet been considered.

There is a rich body of work on *abstract state machines* (ASMs) [16], i.e., state machines whose states are first-order structures. ASMs have been used to give comprehensive specifications of programming languages such as Prolog, C, and Java, and design languages, like UML and SDL (cf. [6]). A number of tools for the verification and validation of ASMs are available [7]. Known decidability results for ASMs require, however, on strong restrictions such as sequential nullary ASMs [33].

In [25,3], it is shown that the semantics of switch controllers of software-defined networks as expressed by *Core SDN* can be nicely compiled into FO transition systems. The goal then is to use this translation to verify given invariants by proving them inductive. Inductivity of invariants is checked by means of the theorem prover Z3 [10]. The authors report that, if their invariants are not already inductive, a single strengthening, corresponding to the computation of $\Psi^{(1)}$ is often sufficient. In [26], the difficulty of inferring universal inductive invariants is investigated for classes of transition systems whose transition relation is expressed by FO logic formulas over theories of data structures. The authors 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 theory of linked lists and becomes undecidable as soon as the binary symbol is not restricted by background theory. By excluding the binary predicate, this result is related to our result for transition systems with monadic predicates, equality and disequality, but neither $\mathcal{A}$- nor $\mathcal{B}$-predicates. In [19], an inference method is provided for universal invariants as an extension of Bradley's PDR/IC3 algorithm for inference of propositional invariants [9]. The method is applied to variants of FO transition systems (no games) within a fragment of FO logic which enjoys the finite model property and is decidable. Whenever it terminates, it either returns a universal invariant which is inductive, or a counter-example. This line of research has led to the tool Ivy which generally applies FO predicate logic for the verification of parametric

systems [28,23]. Relying on a language similar to [25,3], it meanwhile has been used, e.g., for the verification of network protocols such as leader election in ring topologies and the PAXOS protocol [27].

In [13,14,24], hypersafety properties such as noninterference are studied for *multi-agent workflows*. These workflows are naturally generalized by our notion of FO transition systems. The transformation in appendix B for reducing noninterference to universal invariants originates from [24] where also an approximative approach for inferring inductive invariants is provided. When the attempt fails, a counter-example can be extracted — but might be spurious.

All works discussed so far are concerned with verification rather than synthesis. For synthesizing controllers for systems with an infinite state space, several approaches have been introduced that automatically construct, from a symbolic description of a given concrete game, a finite-state abstract game [17,2,11,12,34]. The main method to obtain the abstract state space is predicate abstraction, which partitions the states according to the truth values of a set of predicates. States that satisfy the same predicates are indistinguishable in the abstract game. The abstraction is iteratively refined by introducing new predicates. Applications include the control of real-time systems [12] and the synthesis of drivers for I/O devices [34]. In comparison, our approach provides a general modelling framework of First Order Safety Games to unify different applications of synthesis for infinite-state systems.

9 Conclusion

We have introduced *First Order Safety Games* as a model for reasoning about games on parametric systems where attained states are modeled as FO structures. We showed that this approach allows to model interesting real-world synthesis problems from the domains network protocols and informationflow in multiagent systems. We examined the important case where all occurring predicates are monadic or nullary and provided a complete classification into decidable and undecidable cases. For the non-monadic case, we concentrated on *universal* FO safety properties. We provided techniques for certifying safety and also designed methods for synthesizing FO definitions of predicates as strategies to enforce the given safety objective. We have implemented our approach and succeeded to infer contents of particular messages in the leader election protocol from [28] in order to prove the given invariant inductive. Our implementation also allowed us to synthesize predicates for parametric workflow systems as in [24], to enforce noninterference in presence of declassification. In this application, however, we additionally must take into account that the synthesized formulas only depend on predicates whose values are independent of the secret. Restricting the subset of predicates possibly used by strategies, turns FO safety games into *partial information* safety games. It remains for future work, to explore this connection in greater detail in order, e.g., to determine whether strategies can be automatically synthesized which only refer to specific *admissible* predicates and, perhaps, also take the *history* of plays into account.

References

1. Ackermann, W.: Untersuchungen über das Eliminationsproblem der mathematischen Logik. Mathematische Annalen **110**, 390–413 (1935)
2. de Alfaro, L., Roy, P.: Solving games via three-valued abstraction refinement. In: Proc. CONCUR. vol. 4703, pp. 74–89. Springer-Verlag (2007)
3. Ball, T., Bjørner, N., Gember, A., Itzhaky, S., Karbyshev, A., Sagiv, M., Schapira, M., Valadarsky, A.: Vericon: Towards verifying controller programs in software-defined networks. In: ACM Sigplan Notices. vol. 49, pp. 282–293. ACM (2014)
4. Behmann, H.: Beiträge zur Algebra der Logik, insbesondere zum Entscheidungsproblem. Mathematische Annalen **86**(3-4), 163–229 (1922)
5. Börger, E., Grädel, E., Gurevich, Y.: The classical decision problem. Springer Science & Business Media (2001)
6. Börger, E., Stärk, R.: History and Survey of ASM Research, pp. 343–367. Springer (2003)
7. Börger, E., Stärk, R.: Tool Support for ASMs, pp. 313–342. Springer (2003)
8. Brachman, R.J., Levesque, H.J., Reiter, R.: Knowledge representation. MIT press (1992)
9. Bradley, A.R.: Sat-based model checking without unrolling. In: International Workshop on Verification, Model Checking, and Abstract Interpretation. pp. 70–87. Springer (2011)
10. De Moura, L., Bjørner, N.: Z3: An efficient SMT solver. In: International conference on Tools and Algorithms for the Construction and Analysis of Systems. pp. 337–340. Springer (2008)
11. Dimitrova, R., Finkbeiner, B.: Abstraction refinement for games with incomplete information. In: IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2008. pp. 175–186 (2008)
12. Dimitrova, R., Finkbeiner, B.: Counterexample-guided synthesis of observation predicates. In: Jurdziński, M., Ničković, D. (eds.) Formal Modeling and Analysis of Timed Systems. pp. 107–122. Springer Berlin Heidelberg, Berlin, Heidelberg (2012)
13. Finkbeiner, B., Müller, C., Seidl, H., Zalinescu, E.: Verifying security policies in multi-agent workflows with loops. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017. pp. 633–645. IEEE (2017). https://doi.org/10.1145/3133956.3134080, `https://doi.org/10.1145/3133956.3134080`
14. Finkbeiner, B., Seidl, H., Müller, C.: 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). Lecture Notes in Computer Science, vol. 9938, pp. 157–173 (2016)
15. Gabbay, D.M., Schmidt, R., Szalas, A.: Second Order Quantifier Elimination: Foundations, Computational Aspects and Applications. College Publications (2008)
16. Gurevich, Y.: Evolving algebras 1993: Lipari guide. arXiv preprint arXiv:1808.06255 (2018)
17. Henzinger, T., Jhala, R., Majumdar, R.: Counterexample-guided control. In: Proc. ICALP'03, LNCS, vol. 2719, pp. 886–902. Springer (2003)
18. Holzer, M., Kutrib, M., Malcher, A.: Complexity of multi-head finite automata: Origins and directions. Theoretical Computer Science **412**(1-2), 83–96 (2011)

19. Karbyshev, A., Bjørner, N., Itzhaky, S., Rinetzky, N., Shoham, S.: Property-directed inference of universal invariants or proving their absence. Journal of the ACM (JACM) **64**(1), 7 (2017)
20. Leivant, D.: Higher order logic. In: Gabbay, D.M., Hogger, C.J., Robinson, J.A., Siekmann, J.H. (eds.) Handbook of Logic in Artificial Intelligence and Logic Programming, Volume2, Deduction Methodologies, pp. 229–322. Oxford University Press (1994)
21. Levesque, H.J., Reiter, R., Lespérance, Y., Lin, F., Scherl, R.B.: GOLOG: A logic programming language for dynamic domains. The Journal of Logic Programming **31**(1), 59 – 83 (1997). https://doi.org/https://doi.org/10.1016/S0743-1066(96)00121-5
22. Mazala, R.: Infinite games. In: Grädel, E., Thomas, W., Wilke, T. (eds.) Automata, Logics, and Infinite Games, pp. 23–38. LNCS 2500, Springer, Heidelberg
23. McMillan, K.L., Padon, O.: Deductive verification in decidable fragments with Ivy. In: International Static Analysis Symposium. pp. 43–55. Springer (2018)
24. Müller, C., Seidl, H., Zalinescu, E.: Inductive invariants for noninterference in multi-agent workflows. In: 31st IEEE Computer Security Foundations Symposium, CSF 2018, Oxford, United Kingdom, July 9-12, 2018. pp. 247–261. IEEE (2018). https://doi.org/10.1109/CSF.2018.00025, `https://doi.org/10.1109/CSF.2018.00025`
25. Padon, O., Immerman, N., Karbyshev, A., Lahav, O., Sagiv, M., Shoham, S.: Decentralizing SDN policies. In: ACM SIGPLAN Notices. vol. 50, pp. 663–676. ACM (2015)
26. Padon, O., Immerman, N., Shoham, S., Karbyshev, A., Sagiv, M.: Decidability of inferring inductive invariants. In: Proc. of the 43rd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2016. pp. 217–231. ACM (2016). https://doi.org/10.1145/2837614.2837640
27. Padon, O., Losa, G., Sagiv, M., Shoham, S.: Paxos made EPR: decidable reasoning about distributed protocols. Proceedings of the ACM on Programming Languages **1**(OOPSLA), 108 (2017)
28. Padon, O., McMillan, K.L., Panda, A., Sagiv, M., Shoham, S.: Ivy: safety verification by interactive generalization. ACM SIGPLAN Notices **51**(6), 614–630 (2016)
29. Ramsey, F.P.: On a problem of formal logic. In: Classic Papers in Combinatorics, pp. 1–24. Springer (2009)
30. Rosenberg, A.L.: On multi-head finite automata. IBM Journal of Research and Development **10**(5), 388–394 (1966)
31. Russell, S.J., Norvig, P.: Artificial intelligence: a modern approach. Malaysia; Pearson Education Limited, (2016)
32. Seidl, H., Müller, C., Finkbeiner, B.: How to Win First Order Safety Games - Software Artifact (Oct 2019). https://doi.org/10.5281/zenodo.3514277, `https://doi.org/10.5281/zenodo.3514277`
33. Spielmann, M.: Abstract state machines: verification problems and complexity. Ph.D. thesis, RWTH Aachen University, Germany (2000), `http://sylvester.bth.rwth-aachen.de/dissertationen/2001/008/01_008.pdf`
34. Walker, A., Ryzhyk, L.: Predicate abstraction for reactive synthesis. In: 2014 Formal Methods in Computer-Aided Design (FMCAD). pp. 219–226 (Oct 2014). https://doi.org/10.1109/FMCAD.2014.6987617
35. Wernhard, C.: Second-order quantifier elimination on relational monadic formulas–a basic method and some less expected applications. In: International Conference on Automated Reasoning with Analytic Tableaux and Related Methods. pp. 253–269. Springer (2015)

36. Wernhard, C.: Heinrich Behmann's contributions to second-order quantifier elimination from the view of computational logic. arXiv preprint arXiv:1712.06868 (2017)

A Proof of Theorem 2

Theorem 2. *A FO safety game $\mathcal{T}$ is safe iff* $\mathsf{Init} \to \Psi^{(h)}[v_0]$ *holds for all* $h \geq 0$.

Proof. First, we extend the *weakest precondition* operator to paths π in the control-flow graph of the FO safety game $\mathcal{T}$. Assume that π ends in program point v, and Ψ is a property for that point. The weakest precondition $[\![\pi]\!]^\top \Psi$ of Ψ is defined by induction on the length of π. If $\pi = \epsilon$, then $[\![\pi]\!]^\top \Psi = \Psi$.

Otherwise, $\pi = e\pi'$ for some edge $e = (v_1, \theta, v')$. Then

$$[\![\pi]\!]^\top \Psi = [\![e]\!]^\top ([\![\pi']\!]^\top \Psi)$$

For any path through a FO safety game $\mathcal{T}$ chosen by player $\mathcal{A}$, the weakest precondition can be used to decide if player $\mathcal{B}$ can enforce a given assertion.

Lemma 8. *Let π be a path and* Init *some initial condition and Ψ an assertion about the endpoint of π. Safety player $\mathcal{B}$ has a winning strategy for all plays on π iff* $\mathsf{Init} \to [\![\pi]\!]^\top \Psi$.

Proof. We proceed by induction on the length of π. If $\pi = \epsilon$, then safety player $\mathcal{B}$ wins all games in universes U with valuations ρ starting in states s with $s, \rho \models \mathsf{Init}$ iff $\mathsf{Init} \to \Psi$, and the assertion holds. Now assume that $\pi = e\pi'$ where $e = (u, \theta, v)$. Let $\Psi' = [\![\pi']\!]^\top \Psi$. By inductive hypothesis, safety player $\mathcal{B}$ has a winning strategy for π' with initial condition Ψ'. This means that she can force to arrive at the end point in some s such that $s, \rho \models \Psi$, given that she can start in some s' with $s', \rho \models \Psi'$. By case distinction on whether edge e is an $\mathcal{A}$- or a $\mathcal{B}$-edge, this holds whenever the play starts in some s with $s, \rho \models \mathsf{Init}$.

For the reverse direction, assume that for every universe U and valuation ρ chosen by reachability player $\mathcal{A}$, safety player $\mathcal{B}$ can force to arrive at the end point of π by means of the strategy $\sigma_{U,\rho}$ in a state s such that $s, \rho \models \Psi$ whenever the play starts in some state s_0 with $s_0, \rho \models \mathsf{Init}$. Assume that s_0 is an initial state with $s_0, \rho \models \mathsf{Init}$. Again, we perform a case distinction on the first edge e. First assume that e is a $\mathcal{B}$-edge. Let B denote the relation selected by strategy $\sigma_{U,\rho}$ for e. We construct the successor state s_1 corresponding to edge e and relation B. Since safety player $\mathcal{B}$ can win the game on π' when starting in s_1, we conclude by inductive hypothesis that $s_1, \rho \models \Psi'$. This means that $s_0 \oplus \{B_e \mapsto B_1\}, \rho \models \Psi'\theta$ and therefore $s_0, \rho \models \exists B_e.\Psi'\theta$. If e is an $\mathcal{A}$-edge, then for every choice A of reachability player $\mathcal{A}$, we obtain a successor state s_1 such that by inductive hypothesis, $s_1, \rho \models \Psi'$. This means that for all A, $s_0 \oplus \{A_e \mapsto A\}, \rho \models \Psi'\theta$ and therefore also $s_0, \rho \models \forall A_e.\Psi'\theta$. In both cases, $s_0, \rho \models [\![e]\!]^\top ([\![\pi']\!]^\top \Psi)$ and the claim follows. ◻

For any node in the graph of the game, we successively construct the conjunction of the weakest preconditions of longer and longer paths starting in this particular node. By induction, we verify that

$$\Psi^{(h)}[v] = \bigwedge\{[\![\pi]\!]^\top I \mid \pi \text{ path starting at } v, |\pi| \leq h\}$$

for all $h \geq 0$. Thus, safety player wins on all games of length at most h starting at v_0 iff $\mathsf{Init} \to \Psi^{(h)}[v_0]$ holds, and the assertion of the lemma follows. □

B Games for Noninterference

In a multi-agent application such as the conference management system in fig. 2, not only safety properties but also *noninterference* properties are of interest [13,14,24]. Assume, e.g., that no PC member should learn anything about the reports provided for papers for which she has declared conflict of interest. Our goal is to devise a predicate *Assign* for the edge between program points 1 and 2 which enforces this property.

In order to formalize noninterference for FO transition systems, we require a notion of *observations* of participating agents. Following the conventions in [13,14,24], we assume that from every predicate R of rank at least 1, agent a *observes* the set of all tuples $\bar{z}$ so that $Ra\bar{z}$ holds. Moreover, we assume that there is a set Ω of input predicates predicates whose values are meant to be *disclosed* only to privileged agents. In the example from fig. 2, we are interested in a particular agent a. The predicate A_2 which provides reports for papers, constitutes a predicate whose tuples are only disclosed to agent a if they speak about papers with which a has no conflict.

In general, we assume that for each input predicate $O \in \Omega$, we are given a FO declassification condition $\Delta_{O,a}$ which specifies the set of tuples $\bar{y}$ where the value of $O\bar{y}$ may be disclosed to a. In the example from fig. 2, we have:

$$\Delta_{A_2,a} \;=\; \neg \mathit{Conflict}(a, y_1)$$

Noninterference for agent a at a program point v then means that for each predicate $R \in \mathcal{R}_{state}$, the set of tuples observable by a does not contain information about the sets of nondisclosed tuples of the input predicates in Ω, i.e., stay the same when these sets are modified. Hereby, we assume that the control-flow is public and does not depend on secret information. In a conference management system, e.g., the control-flow does not depend on on the contents of specific reviews or posted opinions on papers.

While noninterference is best expressed as a *hypersafety* property φ_a [13,14], we will not introduce that logic here, but remark that the verification of φ_a for a FO transition system $\mathcal{T}$ can be reduced to the verification of an ordinary safety property φ_a^2 of the (appropriately defined) *self-composition* of $\mathcal{T}$ [24].

In the following we recall that construction for the case that all agents of the system $\mathcal{T}$ are *stubborn* [14]. Intuitively, this property means that all agents' decisions are *independent* of their respective knowledge about input predicates.

A more elaborate construction $\mathcal{T}_a^{(c)}$ works for *causal* agents whose decisions may take their acquired knowledge into account (see [24] for the details). *Self-composition* $\mathcal{T}_a^{(s)}$ of the FO transition system $\mathcal{T}$ for stubborn agents keeps track of two traces of $\mathcal{T}$. For that, a copy R' is introduced for each predicate R in $\mathcal{R}_{state} \cup \Omega$. For a formula φ, let $[\varphi]'$ denote the formula obtained from φ by replacing each occurrence of a predicate $R \in \mathcal{R}_{state} \cup \Omega$ with the corresponding primed version R'. Initially, predicates and their primed versions have identical values, but later-on may diverge due to differences in predicates from Ω. The initial condition Init^2 is obtained from the initial condition Init of $\mathcal{T}$ by setting

$$\mathsf{Init}^2 = \mathsf{Init} \wedge \bigwedge_{R \in \mathcal{R}_{state}} \forall \bar{z}. R\bar{z} \leftrightarrow R'\bar{z} \tag{5}$$

where we assume that the length of the sequence of variables $\bar{z}$ matches the rank of the corresponding predicate R. The first track of $\mathcal{T}_a^{(s)}$ operates on the original predicates from $\mathcal{R}_{state}$, while the second track executes the same operations, but on the primed predicates. Let (u, θ, v) denote a transition of the original system $\mathcal{T}$. First assume that θ does not query any of the predicates from Ω. Then the self-composed system has a transition (u, θ^2, v) where θ^2 agrees with θ on all predicates from $\mathcal{R}_{state}$ where the right-hand side of θ^2 for R' is obtained from the right-hand side of θ for R by replacing each predicate $R \in \mathcal{R}_{state}$ with its primed counterpart.

When all agents are assumed to be *stubborn*, their choices may not depend on their acquired knowledge about the input predicates from Ω. For that reason, the same predicates from $\mathcal{R}_{\mathcal{A}}$ are used on both tracks of the self-composition. Thus, e.g., the second substitution applied in the loop of fig. 2 is extended with

$$\mathit{Review}'(x, p, r) \mathrel{+}= \mathit{Assign}'(x, p) \wedge A_3(x, p, r)$$

Next, assume that θ has no occurrences of predicates in $\mathcal{A} \cup \mathcal{B}$, but accesses some predicate $O \in \Omega$. Then the value of the declassification predicate $\Delta_{O,a}$ for agent a is queried to determine for which arguments O is allowed to *differ* on the two tracks. Accordingly, the substitution θ in $\mathcal{T}$ is replaced with θ_a^2, followed by

$$\begin{aligned} O\bar{y} &\coloneqq Ay \\ O'\bar{y} &\coloneqq \Delta_{O,a} \wedge [\Delta_{O,a}]' \wedge A\bar{y} \vee (\neg\Delta_{O,a} \vee \neg[\Delta_{O,a}]') \wedge A'\bar{y} \end{aligned} \tag{6}$$

for fresh input predicates A, A' controlled by player $\mathcal{A}$. In fig. 4, this means that at the edge from program point 2 to 3, we have

$$\begin{aligned} \mathit{Review}(x, p, r) &:= \mathit{Assign}(x, p) \wedge A_2(x, p, r) \\ \mathit{Review}'(x, p, r) &:= \mathit{Assign}'(x, p) \wedge \\ &\qquad (\neg \mathit{Conflict}(a, p) \wedge \neg \mathit{Conflict}'(a, p) \wedge A_2(x, p, r) \vee \\ &\qquad (\mathit{Conflict}(a, p) \vee \mathit{Conflict}'(a, p)) \wedge A_2'(x, p, r)) \end{aligned}$$

Thus, classification on one of the tracks suffices to allow distinct results when querying the oracle.

The safety property φ_a^2 to be verified for $\mathcal{T}_a^{(s)}$ then amounts to:

$$\bigwedge_{R \in \mathcal{R}_{state}} \forall \bar{z}. Ra\bar{z} \leftrightarrow R'a\bar{z} \tag{7}$$

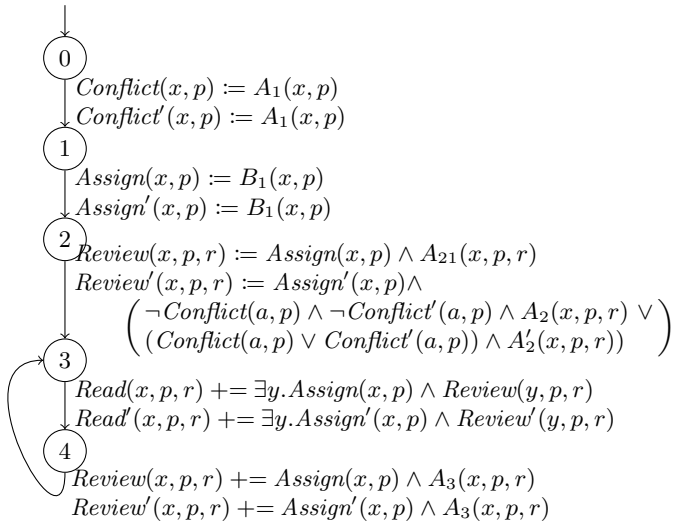

Fig. 4. Self-composition of the FO transition system from fig. 2

where we assume that the length of the sequence of variables $a\bar{z}$ matches the rank of the corresponding predicate R. Let $\mathcal{T}_a^{(s)}$ denote the FO transition system obtained from $\mathcal{T}$ for stubborn agents in this way. In case that player $\mathcal{B}$ has no choice, an adaptation of Theorem 1 from [24] implies that that $\mathcal{T}_a^{(s)}$ satisfies the invariant (7) for all program points u iff φ_a holds for $\mathcal{T}$. This correspondence can be extended to an FO safety game $\mathcal{T}$ where the set of $\mathcal{B}$ predicates $\mathcal{R}_{\mathcal{B}}$ is non-empty. However, we must ensure that the FO formulas describing the winning strategy σ of the self-composition can be translated back into a meaningful strategy for $\mathcal{T}$. A meaningful sufficient condition is that for each $B \in \mathcal{R}_{\mathcal{B}}$, $B\sigma$ depends only on predicates $R \in \mathcal{R}_{state}$ for which R and R' are equivalent. More generally, assume that we are given for each program point u, a set $\mathcal{R}_u$ where

$$\forall \bar{y}.R\bar{y} \leftrightarrow R'\bar{y} \qquad (R \in \mathcal{R}_u) \tag{8}$$

holds whenever program point u is reached. Then the strategy σ for $\mathcal{T}_a^{(s)}$ is *admissible* if for each edge (u, θ, v) containing some $B \in \mathcal{R}_{\mathcal{B}}$, $B\sigma$ contains predicates from $R \in \mathcal{R}_u$ only. Due to (8), the formulas $B\sigma$ and $[B\sigma]'$ then are equivalent. Therefore, we obtain:

Theorem 13. *Let $\mathcal{T}$ be an FO safety game with initial condition I and subset $\mathcal{R}_u \subseteq \mathcal{R}_{state}$ of predicates for each program point u of $\mathcal{T}$. Assume that σ is a strategy so that for each predicate B occurring at some edge (u, θ, v), the FO formula $B\sigma$ only uses predicates from $\mathcal{R}_u$. Let $\mathcal{T}_a^{(s)}$ denote the corresponding FO game with respect to stubborn agents and declassification predicates $\Delta_{O,a}$, and assume that for each program point u, property* (8) *holds whenever u is reached by $\mathcal{T}_a^{(s)}\sigma$. Then the following two statements are equivalent:*

1. *$\mathcal{T}\,\sigma$ satisfies the noninterference property φ_a;*

2. $\mathcal{T}_a^{(s)}\sigma$ satisfies the safety property φ_a^2.

In particular, each admissible winning strategy for the FO safety game $\mathcal{T}_a^{(s)}$ gives rise to a strategy for $\mathcal{T}$ that enforces noninterference.

Finding a strategy that enforces noninterference, thus turns into the synthesis problem for an FO safety game — with the extra obligation that potential winning strategies only access subsets of *admissible* predicates only. In the conference management workflow from fig. 2 with stubborn agents, a strategy is required at the edge from program point 1 to program point 2. At that point, no secret has yet been encountered. Therefore, *all* predicates are admissible — implying that *any* winning strategy for the FO safety game in fig. 4 can be translated back to a strategy which enforces noninterference in $\mathcal{T}$. In particular, we obtain (via $\mathcal{T}_a^{(s)}$) that any FO formula ψ_a guarantees noninterference for which $\psi_a \to \neg \mathit{Conflict}(y_1, y_2)$ holds.

C Proof of Theorem 4

Theorem 4. *For monadic safety games, safety is undecidable when one of the following conditions is met:*

1. *there are both $\mathcal{A}$-edges as well as $\mathcal{B}$-edges;*
2. *there are $\mathcal{A}$-edges and substitutions with equalities or disequalities;*
3. *there are $\mathcal{B}$-edges and substitutions with equalities or disequalities.*

We first consider the case where there are no equalities, but $\mathcal{A}$- as well as $\mathcal{B}$-edges. We show how to construct a safety game G such that an automaton M with multiple counters has a run, starting with empty counters and reaching some designated state term iff safety player $\mathcal{B}$ has no winning strategy in G. The states $q \in \{1, \ldots, n\}$ of M are encoded into flags $f_1, \ldots, f_n$ where f_1 and $f_n = \mathsf{term}$ correspond to the initial and final states, respectively. The invariant I is given by $\neg\mathsf{term}$. Each counter c_i of M is represented by a monadic predicate P_i. Incrementing the counter means to add exactly one element to P_i. In order to do so, we set all flags f_i to *false* whenever the simulation was faulty. Accordingly, we use as initial condition

$$f_1 \wedge \bigwedge_{j>1} \neg f_j \wedge \bigwedge_i \forall x. \neg P_i x$$

Consider a step of M which changes state f_l to $f_{l'}$ and increments counter c_i. The simulation is split into two steps, one A-step followed by one B-step. The A-step uses the substitution:

$$\begin{aligned} \theta_1 = \{\, & P_i y \mapsto P_i y \vee Ay, \\ & f_{l''} \mapsto \begin{cases} f_l \wedge (\exists x. Ax \wedge \neg P_i x) & \text{if } l'' = l' \\ \mathit{false} & \text{if } l'' \neq l', \end{cases} \\ & P' y \mapsto Py \,\} \end{aligned}$$

where P' is meant to record the values of the predicate P_i before the transition. By this transition, some flag $f_{l''}$ is set only when the new predicate P_i has received some new element. By the subsequent second transition, safety player $\mathcal{B}$ can achieve $\bigwedge_l \neg f_l$ whenever the predicate A chosen by reachability player $\mathcal{A}$ in the previous step, has more than one element outside P_i:

$$\theta_2 = \{\, P_{i'}y \mapsto P_{i'}y,\quad f_{l''} \mapsto f_{l''} \wedge \forall x_1 x_2.\begin{pmatrix} P_i x_1 \vee \neg P' x_1 \vee \\ P_i x_2 \vee \neg P' x_2 \vee \\ B x_1 \vee \neg B x_2 \end{pmatrix},\quad P'y \mapsto \mathit{false}\,\}$$

Decrement by 1 can be simulated analogously. Since counters can also be checked for 0, we find that safety player $\mathcal{B}$ wins a play iff either the simulation of the counters was erroneous or reachability player $\mathcal{A}$ is not able to reach term. Accordingly, statement (1) of the theorem follows.

In the given simulation, the B-predicates can be replaced by means of an equality in the substitution:

$$f_{l''} \mapsto f_{l''} \wedge \forall x_1 x_2 . P_i x_1 \vee \neg P' x_1 \vee P_i x_2 \vee \neg P' x_2 \vee x_1 = x_2$$

Therefore, also statement (2) follows. A disequality would have served the same purpose if deviation from the correct simulation would have been tracked by means of an error flag. This kind of simulation is exemplified for the proof of statement (3).

For statement (3), we introduce a dedicated error flag *error* and sharpen the invariant to

$$\neg \mathit{error} \wedge (\bigvee_{j=1}^{n-1} f_j \vee \bigwedge_{j=1}^{n} \neg f_j)$$

The error flag is initially assumed to be *false*, and used to force safety player $\mathcal{B}$ to choose sets B with appropriate properties. Thus, we use

$$\neg \mathit{error} \wedge f_1 \wedge \bigwedge_{j>1} \neg f_j \wedge \bigwedge_i \forall x . \neg P_i x$$

as initial condition. For the actual simulation, we use a single program point together with edges for each transition of the counter machine. Incrementing counter c_i (combined with state transition from q_l to $q_{l'}$), e.g., is simulated by an edge with the substitution

$$\begin{aligned} \theta' = \{\, P_i y &\mapsto P_i y \vee B y, \\ f_{l''} &\mapsto \begin{cases} f_l & \text{if } l'' = l' \\ \mathit{false} & \text{if } l'' \neq l' \end{cases}, \\ \mathit{error} &\mapsto \mathit{error} \vee (\forall x . \neg B x \vee P_i x) \vee \\ &\quad (\exists x_1 x_2 . \neg(B x_1 \wedge \neg P_i x_1) \vee \\ &\qquad \neg(B x_2 \wedge \neg P_i x_2) \vee x_1 \neq x_2)\,\} \end{aligned}$$

Due to $\neg \mathit{error}$ in the invariant, safety player $\mathcal{B}$ is forced to choose a set B which adds exactly one element to P_i, while the subformula $\bigwedge_j \neg f_j$ forces reachability player $\mathcal{A}$ is choose edges according to the state transitions of the counter machine. □

D Proof of Theorem 5

Theorem 5. *Assume that $\mathcal{T}$ is a monadic safety game, possibly containing equalities and/or disequalities with $\mathcal{R}_{\mathcal{A}} = \mathcal{R}_{\mathcal{B}} = \emptyset$. Then for some $h \geq 0$, $\Psi^{(h)} = \Psi^{(h+1)}$. Therefore, safety of $\mathcal{T}$ is decidable.*

For the proof of theorem 5, we rely on a technique similar to the *Counting Quantifier Normal Form* (CQNF) as introduced by Behmann in [4] and picked up in [36]. A *counting quantifier* $\exists^{\geq n} x.\varphi(x)$ expresses that at least n individuals exist for which φ holds, i.e.

$$\exists^{\geq n} x.\varphi \equiv \exists x_1 \ldots x_n . \textstyle\bigwedge_{1 \leq i \leq n} \varphi[x_i/x] \wedge \bigwedge_{i<j \leq n} x_i \neq x_j$$

The main theorem is: A monadic FO formula φ is said to be in *liberal counting quantifier normal form* (liberal CQNF) iff φ is a Boolean combination of *basic formulas* of the form:

- $\exists^{\geq n} x. \bigwedge\limits_{1 \leq i \leq m} L_i(x)$
  where $n \geq 1$, $m \geq 0$, and the $L_i(x)$ are pairwise different and pairwise non-complementary positive or negative literals with unary predicates applied to the individual variable x, and dis-equalities $x \neq a$ for free variables a,
- nullary predicates P,
- $P(x)$, where P is a unary predicate and x is a global variable,
- $x = x'$, where x, x' are global variables.

φ is in *strict counting quantifier normal form* (strict CQNF) if it is in liberal CQNF and additionally does not have dis-equalities of bound FO variables with free FO variables. We remark that the notion of strict CQNF has been called just CQNF in [36]. We have:

Theorem 14 (CQNF for Monadic FO Formulas [36,4]). *From each monadic FO formula φ equivalent FO formulas φ_1, φ_2 can be constructed such that*

1. *φ_1 is in liberal CQNF;*
2. *φ_2 is in strict CQNF;*
3. *all FO variables and predicates in φ_1, φ_2 also occur in φ.* □

We remark that the construction of φ_1 in liberal CQNF follows the same lines as the construction of φ_2 where only the step of eliminating dis-equalities between bound variables and free variables is omitted.

The transformation into strict CQNF is illustrated by the following example.

Example 10.

$$\begin{aligned}
&\exists y.\ \exists x.\ px \wedge x \neq y && \equiv \\
&\exists y.\ (\exists^{\geq 1} x.\ px) \wedge ((\exists^{\geq 2} x.\ px) \vee \neg py) && \equiv \\
&(\exists^{\geq 1} x.\ px) \wedge ((\exists^{\geq 2} x.\ px) \vee \exists y.\neg py) && \equiv \\
&(\exists^{\geq 1} x.\ px) \wedge ((\exists^{\geq 2} x.\ px) \vee \exists^{\geq 1} y.\neg py)
\end{aligned}$$

For a monadic FO formula φ in liberal or strict CQNF, the *quantifier rank* $qr(\varphi)$ equals the maximal k such that $\exists^{\geq k}$ occurs in φ. Likewise, for a substitution θ where all images of predicates are in strict CQNF, $qr(\theta)$ equals the maximal rank of a formula in the image of θ. For the rest of this subsection, we assume that for all substitutions, all formulas in their images are in strict CQNF. We now state our results for such substitutions on monadic FO formulas in CQNF.

Lemma 9. *Given a monadic FO formula φ in liberal CQNF and a substitution θ, the quantifier rank of $\varphi\theta$ in liberal CQNF is at most the maximum of $qr(\varphi)$ and $qr(\theta)$.*

Proof. Since φ is in liberal CQNF and $R\theta$ ($R \in \mathcal{R}_{state}$) are all in strict CQNF, none of them contain equalities between bound variables, and All quantifier scopes $\exists^{\geq k} x.$ contain only literals that mention x. While in φ these scopes may contain inequalities $x \neq a$ for free variables a, this is not allowed in the $R\theta$. In particular, there are no dis-equalities between y and a bound FO variable. Thus, we can write $R\theta$ in the form $\bigvee_{j=1}^{l_R} \psi_{R,j}(y) \wedge \psi'_{R,j} \vee \psi''_R$ where each $\psi_{R,j}(y)$ is a *quantifierfree* boolean combination of literals applied to y, equalities or dis-equalities of y with further free variables, and all $\psi'_{R,j}$, ψ''_R do not contain y. Applying θ to a literal $L(a)$, where a is a free variable, does not introduce new nested quantifiers. Now consider a quantified basic formula

$$\exists^{\geq k} x.\, (\textstyle\bigwedge_{i=1}^{l_1} L_i(x)) \wedge (\bigwedge_{i=1}^{l_2} \neg L'_i(x)) \wedge D(x)$$

of φ where $D(x)$ is a conjunction of disequalities with free variables of φ. Application of θ yields a formula which is a boolean combination of

- basic formulas from θ without occurrences of y since these can be extracted out of the scope of any quantifier of φ;
- basic formulas from φ without occurrences of predicates;
- basic formulas arising of the CQNF of a formula

$$\exists^{\geq k} (\bigwedge_{j=1}^{m} \psi_{R_j,i_j}[x/y]) \wedge \neg\psi_{R,i}[x/y] \wedge D(x)$$

  for some predicates R_j, R and indices i_j, i. By construction, each formula $\psi_{R_j,i_j}[x/y]$ as well as formula $\neg\psi_{R,i}[x/y]$ is quantifierfree. Therefore, it is equivalent to a boolean combination of basic formulas of rank at most k.

Altogether, the rank of $\varphi\theta$ is thus bounded by the maximum of the ranks of φ and θ.

Lemma 10. *For any monadic FO formula φ in liberal CQNF and a sequence of substitutions $\theta_0, \ldots, \theta_n$, in strict CQNF, it holds that*

$$qr(\varphi\theta_0 \ldots \theta_n) \leq \max(qr(\varphi), qr(\theta_0), \ldots, qr(\theta_n))$$

The proof follows from the repeated application of lemma 9.

Now that we proved the intermediate steps, we can prove the initial theorem 5.

Proof (Proof of theorem 5). For all $h \geq 0$ and nodes v, $\Psi^{(h)}[v]$ is a conjunction of sequences of substitutions θ from E applied to some FO formula $I[v']$. Thus, $qr(\Psi^{(h)}[v])$ is at most

$$\max(\{qr(\theta) \mid (v, \theta, v') \in E\} \cup \{qr(I[v']) \mid v' \in V\})$$

Let r be this maximum. For a given set of constants, there are only finitely many formulas of fixed quantifier rank (up to logical equivalence). Thus, fixpoint computation as given in section 3 necessarily terminates. According to the proof of theorem 2, a game G is safe iff for all $h \geq 0$, $\mathsf{Init} \rightarrow \Psi^{(h)}[v_0]$. Therefore, theorem 5 follows.

We remark that the given finite upper bound r to the ranks of all formulas $\Psi^{(h)}[v]$ together with the finite model property [5] implies that reachability player $\mathcal{A}$ can win iff $\mathcal{A}$ can win in a a universe of size at most $r'2^{|\mathcal{R}_{state}|}$ where r' is the maximum of r and the rank of Init.

E Proof of Theorem 6

Theorem 6. *Assume that $\mathcal{T}$ is a monadic safety game without $\mathcal{B}$-edges (i.e. $\mathcal{R}_{\mathcal{B}} = \emptyset$) and*

1. *there are no disequalities between bound variables in I,*
2. *in all literals $x = y$ or $x \neq y$ in Init and substitutions θ, $x \in \mathcal{C}$ or $y \in \mathcal{C}$.*

Then it is decidable whether $\mathcal{T}$ is safe.

We have:

Lemma 11. *Let φ be a FO formula with free variables from $\mathcal{C}$ possibly containing equalities or disequalities between bound variables. We construct a formula $\varphi^\sharp$ with free variables from $\mathcal{C}$ and neither positive nor negative equalities between bound variables such that the following holds:*

1. *$\varphi^\sharp \rightarrow \varphi$;*
2. *If $\psi \rightarrow \varphi$ holds for any other monadic formula ψ without (dis-)equalities between bound variables, then $\psi \rightarrow \varphi^\sharp$.*
3. *There exists some $d \geq 0$ such that for a model s of multiplicity at least d and a valuation ρ, $s, \rho \models \varphi^\sharp$ iff $s, \rho \models \varphi$.*

If the assumptions of lemma 11 are met, $\varphi^\sharp$ is called the *weakest strengthening* of φ by formulas without equalities.

Proof. Assume that φ is in prenex normal form and that the quantifierfree part φ' is in disjunctive normal form. By transitivity of equality, we may assume that in each monomial m of φ' for each occurring equality $x = y$ one of the following properties holds:

- both x, y are free variables; or
- x is free and y occurs in the quantifier prefix; or
- neither x nor y are free, x is different from y and the leftmost variable in the quantifier prefix which is transitively equal to y.

Next, m is rewritten in such a way that additionally no variable y on a right side of an equality is existentially quantified. As a result, each remaining right side of an equality literal is either free in φ' (in which case the left side is also free) or universally quantified. Now consider any model s such that $\mu(s) \geq d$ for some $d > 0$ exceeding the number of free variables plus the length of the quantifier prefix of φ. Then we verify for each universally quantified variable y (by induction on the number of universally quantified variables occurring in a quantifier prefix Qz), that $s, \rho \models \forall y\, Qz.\varphi'$ iff $s, \rho \models \forall y\, Qz.\varphi''$ where φ'' is obtained from φ' by replacing each occurrence of an equality $x = y$ with $x \sim_{\mathcal{C}} y$, defined as $\bigvee_{c \in \mathcal{C}} x = c \wedge y = c$.

Accordingly, we construct $\varphi^\sharp$ from φ by replacing all equalities $x = y$ where y is universally quantified with $x \sim_{\mathcal{C}} y$. Then $\varphi^\sharp$ satisfies statements (1) and (3). In order to prove statement (2), we first observe that $\psi \to \varphi$ also holds for all models s with $\mu(s) \geq d$ for all values of d exceeding the cardinality of $\mathcal{C}$. By property (3), we therefore have that $\psi \to \varphi^\sharp$ in all models s and all valuations ρ where $\mu(s)$ is sufficiently large. Since (dis-)equalities in $\varphi^\sharp$ and in ψ are not applied to pairs of bound variables, the assertion follows.

We conclude:

Corollary 1. *Assume that φ, φ' are monadic FO formulas with positive occurrences of equality only. Then*

1. $(\varphi \wedge \varphi')^\sharp = \varphi^\sharp \wedge (\varphi')^\sharp$, *and*
2. $(\forall A.\varphi)^\sharp = (\forall A.\varphi^\sharp)^\sharp$

With this, we can now prove the initial theorem 6.

Proof (Proof of theorem 6). Let $\Psi^{(h)}$ denote the hth iteration of the weakest precondition (1) as defined in section 3. Due to SO Quantifier Elimination as in [4], each formula $\Psi^{(h)}[v]$ is equivalent to a monadic FO formula. If neither I nor θ contain equalities, $\Psi^{(h)}[v]$ has positive occurrences of equalities only.
The sequence $\Psi^{(h)}[v]$ for $h \geq 0$ still need not stabilize as more and more FO variables may be introduced. Let $\Psi_0^{(h)}$ denote the hth iteration of the *abstraction* of the weakest precondition:

$$\begin{aligned}
\Psi_0^{(0)}[v] &= I[v] \\
\Psi_0^{(h)}[v] &= \Psi_0^{(h-1)} \wedge \\
&\qquad \bigwedge_{(v,\theta,v') \in E} (\forall A_e.(\Psi_0^{(h-1)}[v']\theta))^\sharp \quad \text{for } h > 0
\end{aligned}$$

where the abstraction operator $(\cdot)^\sharp$ returns the weakest strengthening by means of a monadic FO formula without equality. Recall from corollary 1 that the abstraction operator commutes with conjunctions. Also, we have that $(\forall A_e.\varphi)^\sharp = (\forall A_e.\varphi^\sharp)^\sharp$ for each monadic FO formula with positive occurrences of equality only. By induction on h, we find that $\Psi_0^{(h)}[v] = (\Psi^{(h)}[v])^\sharp$ holds for all $h \geq 0$. Since Init does not contain equalities, we therefore have for all $h \geq 0$, that $\mathsf{Init} \to \Psi^{(h)}[v_0]$ iff $\mathsf{Init} \to \Psi_0^{(h)}[v_0]$. Since (up to equivalence) the number of monadic formulas without equalities or disequalities is finite, the sequence $\Psi_0^{(h)}$ for $h \geq 0$ eventually stabilizes. This means that there is some $h' \geq 0$ such that for each program point v, $\Psi_0^{(h')}[v] = \Psi_0^{(h'+1)}[v]$. Thus, game G is safe iff $\mathsf{Init} \to \Psi_0^{(h')}[v_0]$. Since the implication is decidable, the theorem follows.

F Proof of Theorem 7

Theorem 7. *Assume that $\mathcal{T}$ is a monadic safety game without $\mathcal{A}$-edges where*

1. *there are no equalities between bound variables in I,*
2. *in all literals $x = y, x \neq y$ in Init and substitutions θ, either $x \in \mathcal{C}$ or $y \in \mathcal{C}$.*

Then it is decidable whether $\mathcal{T}$ is safe.

The proof is analogous to the proof of theorem 6 where the abstraction of equalities now is replaced with an abstraction of disequalities, and corollary 1 is replaced with a similar corollary 2 dealing with disequalities.

In analogy to safety games with invariants containing equalities, we provide a weakest strengthening of monadic FO formulas, now containing positive occurrences of disequalities only. Let φ denote a monadic formula in negation normal form with free variables from $\mathcal{C}$, and no positive occurrences of equalities between bound variables. We define $\varphi^\sharp$ now as the formula obtained from φ by replacing each literal $x \neq y$ (x, y bound variables) with

$$\begin{array}{l}(\bigvee_{R\in\mathcal{R}} Rx \wedge \neg Ry \vee \neg Rx \wedge Ry) \vee \\ (\bigvee_{c\in\mathcal{C}} x = c \wedge y \neq c \vee x \neq c \wedge y = c)\end{array} \tag{9}$$

Then, $\varphi^\sharp \to \varphi$ holds, and we claim:

Lemma 12. *Let ψ be any monadic FO formula without equalities or disequalities between bound variables such that $\psi \to \varphi$ holds. Then also $\psi \to \varphi^\sharp$ holds.*

Proof. We proceed by induction on the structure of φ. Clearly, the assertion holds whenever φ does not contain disequalities between bound variables. Assume that φ is the literal $x \neq y$ for bound variables x, y. Assume that $\psi \to (x \neq y)$, but ψ does not imply formula (9). This means that there is a model M and an assignment ρ such that $M, \rho \models \psi \wedge \bigwedge_{R\in\mathcal{R}} Rx \wedge Ry \vee \neg Rx \wedge \neg Ry \wedge \bigwedge_{c\in\mathcal{C}} (x \neq c \vee y = c) \wedge (x = c \vee y \neq c)$ holds. W.l.o.g., M is minimal, i.e., elements which cannot be distinguished by means of predicates in $\mathcal{R}$ or free variables in $\mathcal{C}$, are equal. But then $M, \rho \not\models (x \neq y)$ — in contradiction to the assumption.

Now assume that $\varphi = \varphi_1 \wedge \varphi_2$. Then $\varphi^\sharp = \varphi_1^\sharp \wedge \varphi_2^\sharp$. Let ψ imply φ. Then $\psi \to \varphi_i$ for each i. Therefore, by induction hypothesis, $\psi \to \psi_i^\sharp$ for all i. As a consequence, $\psi \to \varphi^\sharp$.

Now assume that $\varphi = \varphi_1 \vee \varphi_2$. Then $\varphi^\sharp = \varphi_1^\sharp \vee \varphi_2^\sharp$. If ψ implies φ, then for each model M variable assignment ρ, there is some i so that $M, \rho \models \psi \to \varphi_i$. Assume for a contradiction that $\psi \wedge \neg(\varphi_1^\sharp \vee \varphi_2^\sharp)$ is satisfiable. Then there is some model M, assignment ρ so that $M, \rho \models \psi \wedge \neg\varphi_1^\sharp \wedge \neg\varphi_2^\sharp$. In particular, there is some i so that $M, \rho \models \varphi_i \wedge \neg\varphi_1^\sharp \wedge \neg\varphi_2^\sharp$. By inductive hypothesis, $\varphi_i^\sharp \to \varphi_i$ holds. We conclude that $M, \rho \models \varphi_i \wedge \neg\varphi_1 \wedge \neg\varphi_2$ holds — contradiction. Similar arguments also apply to existential and universal quantification in φ. As a consequence, $\psi \to \varphi^\sharp$ holds.

Corollary 2. *Assume that φ, φ' are monadic FO formulas without positive occurrences of equalities between bound variables. Then*

1. $(\varphi \wedge \varphi')^\sharp = \varphi^\sharp \wedge (\varphi')^\sharp$, *and*
2. $(\exists B.\varphi)^\sharp = (\exists B.\varphi^\sharp)^\sharp$

G Proof of Lemma 4

Lemma 4. *Every universal FO formula φ possibly containing occurrences of B is equivalent to a formula*

$$E \wedge (\forall \bar{y}.F \vee B\bar{y}) \wedge (\forall \bar{y}'.G \vee \neg B\bar{y}') \wedge (\forall \bar{y}\bar{y}'.H \vee B\bar{y} \vee \neg B\bar{y}') \tag{2}$$

where E, F, G, H are universal formulas without B.

Proof. W.l.o.g., we assume that $\varphi = \forall x.\varphi'$ where φ' is quantifierfree and in conjunctive normal form. Let E, F', G', H' equal the conjunction of all clauses in φ' containing no occurrence of B, only positive, only negative and both positive and negative occurrences of B. Each clause of the form $c' \vee Bz_1 \vee \ldots \vee Bz_k$ (c' without B) is equivalent to

$$\forall y.c' \vee (\textstyle\bigwedge_{i=1}^k z_i \neq y) \vee By$$

where $z_i \neq y$ abbreviates the disjunction $\bigvee_{j=1}^r z_{ij} \neq y_j$ — given that $z_i = z_{i1} \ldots z_{ir}$. Likewise, each clause of the form $c' \vee \neg Bz_1 \vee \ldots \vee \neg Bz_k$ (c' without B) is equivalent to

$$\forall y'.c' \vee (\textstyle\bigwedge_{i=1}^k z_i \neq y') \vee \neg By'$$

Finally, each clause of the form $c' \vee \neg Bz_1 \vee \ldots \vee \neg Bz_k \vee \neg z_1' \vee \ldots \neg Bz_l'$ (c' without B) is equivalent to

$$\forall y y'.c' \vee (\textstyle\bigwedge_{i=1}^k z_i \neq y) \vee (\bigwedge_{i=1}^l z_i' \neq y') \vee By \vee \neg By'$$

Applying these equivalences to the clauses in the conjunctions in F', G', H', respectively, we arrive at conjunctions of clauses which all contain just the B-literal By, the B-literal $\neg By'$ or $By \vee \neg By'$, respectively. From these, the formulas F, G and H can be constructed by distributivity.

H Proof of theorem 12

Theorem 12. *The weakest Hilbert choice operator $\mathcal{H}_B\varphi$ for the universal formula* (2) *is definable by the SO formula:*

$$\neg E \vee \exists B.B\bar{y} \wedge (\forall \bar{y}'.G \vee \neg B\bar{y}') \wedge (\forall \bar{y}\bar{y}'.H \vee B\bar{y} \vee \neg B\bar{y}')$$

Proof. Our goal is to prove that $\exists B.\varphi$ implies the formula $\varphi[\mathcal{H}_B\varphi/B]$. We consider each conjunct of φ in turn.

$$\begin{aligned}
\forall \bar{y}.F \vee \mathcal{H}_B\varphi &= \forall \bar{y}.\exists B.F \vee (B\bar{y} \wedge \\
&\quad (\forall \bar{y}'.G \vee \neg B\bar{y}') \wedge (\forall \bar{y}\bar{y}'.H \vee B\bar{y} \vee \neg B\bar{y}')) \\
&\leftarrow \forall \bar{y}.\exists B.(F \vee B\bar{y}) \wedge \\
&\quad (\forall \bar{y}'.G \vee \neg B\bar{y}') \wedge (\forall \bar{y}\bar{y}'.H \vee B\bar{y} \vee \neg B\bar{y}') \\
&\leftarrow \exists B.\forall \bar{y}.(F \vee B\bar{y}) \wedge \\
&\quad (\forall \bar{y}'.G \vee \neg B\bar{y}') \wedge (\forall \bar{y}\bar{y}'.H \vee B\bar{y} \vee \neg B\bar{y}') \\
&= \exists B.\varphi
\end{aligned}$$

$$\begin{aligned}
\forall \bar{y}'.G \vee \neg\mathcal{H}_B\varphi &= \forall \bar{y}'.\forall B.G \vee \neg B\bar{y}' \vee \\
&\quad (\exists \bar{y}'.B\bar{y}' \wedge \neg G) \vee (\exists \bar{y}\bar{y}'.\neg H \wedge \neg B\bar{y} \wedge B\bar{y}') \\
&= \forall B.\forall \bar{y}'.G \vee \neg B\bar{y}' \vee \\
&\quad (\exists \bar{y}'.\neg G \wedge B\bar{y}') \vee (\exists \bar{y}\bar{y}'.\neg H \wedge \neg B\bar{y} \wedge B\bar{y}') \\
&= \mathit{true}
\end{aligned}$$

$$\begin{aligned}
&\forall \bar{y}\bar{y}'.H \vee \mathcal{H}_B\varphi \vee \neg\mathcal{H}_B\varphi[\bar{y}'/\bar{y}] \\
&= \forall \bar{y}\bar{y}'.H \vee \\
&\quad (\exists B.B\bar{y} \wedge (\forall \bar{y}'.G \vee \neg B\bar{y}') \wedge (\forall \bar{y}\bar{y}'.H \vee B\bar{y} \vee \neg B\bar{y}')) \vee \\
&\quad (\forall B.\neg B\bar{y}' \vee \neg(\forall \bar{y}'.G \vee \neg B\bar{y}') \vee \neg(\forall \bar{y}\bar{y}'.H \vee B\bar{y} \vee \neg B\bar{y}')) \\
&\leftarrow \forall \bar{y}\bar{y}'.\forall B.H \vee \\
&\quad B\bar{y} \wedge (\forall \bar{y}'.G \vee \neg B\bar{y}') \wedge (\forall \bar{y}\bar{y}'.H \vee B\bar{y} \vee \neg B\bar{y}') \vee \\
&\quad \neg B\bar{y}' \vee \neg(\forall \bar{y}'.G \vee \neg B\bar{y}') \vee \neg(\forall \bar{y}\bar{y}'.H \vee B\bar{y} \vee \neg B\bar{y}') \\
&= \forall \bar{y}\bar{y}'.\forall B.H \vee B\bar{y} \vee \neg B\bar{y}' \vee \\
&\quad \neg B\bar{y}' \vee \neg(\forall \bar{y}'.G \vee \neg B\bar{y}') \vee \neg(\forall \bar{y}\bar{y}'.H \vee B\bar{y} \vee \neg B\bar{y}') \\
&= \mathit{true}
\end{aligned}$$

Altogether therefore, $\exists B.\varphi \rightarrow \varphi[\mathcal{H}_B\varphi/B]$, and the assertion follows.