Games for Inclusion Logic and Fixed-Point Logic

Erich Grädel

Abstract One of the most intriguing results on logics of dependence and independence is the tight connection between inclusion logic and the least fixed-point logic LFP. Here we re-examine this connection from a game-theoretic point of view. We study the model-checking games for inclusion logic and for posGFP, the fragment of LFP that uses only (non-negated) greatest fixed points. We show that the evaluation problems for both logics can be represented by a special kind of trap condition in safety games. We then study interpretation arguments for games. In combination with our study of traps for inclusion logic and posGFP, game interpretations will give us a model-theoretic construction of translations between the two logics

1 Introduction

Modern logics of dependence and independence come with a semantics that, unlike Tarski semantics, is not based on single assignments (mapping variables to elements of a structure) but on sets of such assignments. Sets of assignments with a common domain of variables are called teams. Team semantics was originallly introduced by Hodges [10, 11] as a compositional, model-theoretic semantics for the independence-friendly logic IF. In 2007, Väänänen [14] proposed a new approach to logics of dependence and independence. Rather than stating dependencies or independencies as annotations of quantifiers, he suggested to express dependencies as atomic formulae, of the form $=(x_1,\ldots,x_m,y)$, saying that the variable y is functionally dependent on (i.e. completely determined by) the variables $x_1,\ldots,x_m$. Dependence logic is first-order logic together with such dependency atoms. Notice that such dependency statements do not not even make sense on a single assignment, but only on larger collection of data, given either by sets of assignments, i.e., teams, or by a table or relation. Besides the functional dependency atoms proposed

Erich Grädel
RWTH Aachen University, e-mail: graedel@logic.rwth-aachen.de

by Väänänen, there are many other atomic dependence properties that give rise to interesting logics based on team semantics. In [8] we have discussed the notion of independence (which is a much more delicate but also more powerful notion than dependence) and introduced independence logics, and Galliani [5] and Engström [4] have studied several logics with team properties based on notions originating in database dependency theory,

Most of the logics of dependence and independence studied so far, including more traditional formalisms such as first-order logic with Henkin quantifiers and IF-logic, have an expressive power that is, at least for sentences, equivalent to the one of existential second-order logic [2, 5, 12, 13, 14], and it is rather easy to formalize NP-complete properties of, say, finite graphs in these logics. However, one of the most surprising results on logics with team semantics is the tight connection, established by Galliani and Hella [6], between inclusion logic and the least fixed-point logic LFP. Inclusion logic extends first-order logic (with team semantics) by atomic inclusion dependencies of the form $(\overline{x} \subseteq \overline{y})$, which are true in a team X if every value for $\overline{x}$ in X also occurs as a value for $\overline{y}$ in X. Inclusion logic has been introduced and studied in [5].

The least fixed-point logic LFP, on the other side, is a logic with classical semantics in the sense of Tarski, which extends first-order logic by least and greatest fixed points of definable relational operators. The logic LFP is of fundamental importance in finite model theory and descriptive complexity, for the study of inductive definability, and for the study of logic and games. Fragments of LFP, such as Datalog or the modal μ-calculus are very important in many areas of computer science, including databases, knowledge representation, and verification. See for instance [9] for background on least fixed-point logic.

Galliani and Hella showed, by a direct translation via structural induction, that sentences of inclusion logic have the same expressive power as sentences from the fragment of LFP that uses only (non-negated) greatest fixed points, denoted posGFP. It is known that, on finite structures, the full logic LFP collapses to its posGFP-fragment. Hence every property of finite structures that is LFP-definable is also definable in inclusion logic, and vice versa. It follows by the Immerman-Vardi-Theorem that, on *ordered* finite structures, inclusion logic captures polynomial time. For formulae with free variables, the connection between inclusion logic and posGFP is more complicated, due to the different semantics of the two logics. We will discuss this issue in detail in Section 7.

In this article, we re-examine the connection between LFP and inclusion logic from a game-theoretic point of view. We study the model-checking games for inclusion logic and for the posGFP-fragment of least fixed-point logic and use interpretation arguments for games to translate between the two logics.

It is well-known that the appropriate games for greatest fixed-point formulae are *safety games*, i.e., games with potentially infinite plays, where Player 0 has just the objective to keep the play inside a safe region or, equivalently, to avoid a given set of losing positions. In our case the positions to avoid are the literals that evaluate to false.

Model-checking games for logics with team semantics are *a priori* quite different. A uniform construction of such games has been presented in [7] in terms of *second-order reachability games* played on trees or forests. Whereas in classical reachability (or safety) games, the winning condition is specified by a set of positions that should be reached (or avoided), a second-order reachability condition is given by a *collection of sets of terminal positions*. Furthermore, a second-order reachability condition does not apply to single plays, but to strategies, and considers the set of all plays that are compatible with the strategy. To be winning, a strategy has to ensure that the set of all terminal positions that are reachable by a play following the strategy forms a winning set. We have shown in [7] that for any logic with team semantics, satisfying some natural basic conditions, the associated model-checking problem can be captured by appropriate second-order reachability games. In particular, this is the case for inclusion logic.

Although second-order reachability games are quite different from safety games, and in general algorithmically more complicated, we shall prove that games played on forests, with a second-order reachability condition of a special form given by a universal-existential statement, can be translated into equivalent safety games, on a transformed game graph that is no longer acyclic. This applies in particular to games for inclusion logic and thus provides safety games for this logic, as an alternative to the second-order reachability games obtained by the generic construction. Further we introduce *I-traps*, a special notion of traps for initial positions in safety games, and prove that this notion faithfully captures evaluation problems with respect to *teams*, for formulae of inclusion logic and also for posGFP-formulae of a special form. On the other side, such traps are definable in both logics in a quite simple way.

We shall then study game interpretations. It is a general observation that the model checking games for a given formula (from almost any reasonable logic) are uniformly interpretable inside the structure on which the formula is evaluated. In combination with our study of traps for inclusion logic and posGFP, interpretations will give us translations between the two logics. The argument roughly is the following. Given any formula ψ in, say, inclusion logic, we consider the interpretation $J(\psi)$ which, for any structure $\mathfrak{A}$, interprets the safety game for $\mathfrak{A}$ and ψ, denoted $\mathscr{G}_{\text{safe}}(\mathfrak{A},\psi)$, inside $\mathfrak{A}$. This game has the properties that the teams X that satisfy ψ in $\mathfrak{A}$ coincide with the I-traps in $\mathscr{G}_{\text{safe}}(\mathfrak{A},\psi)$. On the other side, I-traps in safety games are definable by a formula itrap of the target logic, in this case posGFP. The interpretation $J(\psi)$ maps this formula to another formula $\text{itrap}^{J(\psi)}$, which is also in posGFP, and which essentially expresses in $\mathfrak{A}$, what itrap expresses in the game. From $\text{itrap}^{J(\psi)}$ we then easily get a posGFP-formula that is equivalent to ψ. An analogous translation works in the other direction. We thus obtain a high-level model-theoretic technique for obtaining translations between the two logics, without the need to go through cumbersome structural inductions on the syntax of the formulae.

2 Safety games and traps

There are many models for path-forming games played on graphs. Here we work with a model of turn-based games with two players, called Player 0 and Player 1, that makes explicit not only the sets of positions associated to the two players, but also the initial and terminal positions.

A game graph is a structure $\mathscr{G} = (V, V_0, V_1, T, I, E)$, where $V = V_0 \cup V_1 \cup T$ is the set of positions, partitioned into the sets V_0, V_1 of the two players and the set T of terminal positions, where I is the set of initial positions, and where $E \subseteq V \times V$ is the set of moves. We denote the set of immediate successors of a position v by $vE := \{w : (v,w) \in E\}$ and require that $vE = \emptyset$ if, and only if, $v \in T$. A play from an initial position v_0 is a finite or infinite path $v_0 v_1 v_2 \ldots$ through $\mathscr{G}$ where the successor $v_{i+1} \in v_i E$ is chosen by Player 0 if $v_i \in V_0$ and by Player 1 if $v_1 \in V_1$. A play ends when it reaches a terminal node $v_m \in T$. A subgraph of a graph (V,E) is a pair (W,F) with $W \subseteq V$ and $F \subseteq E \cap (W \times W)$.

Definition 1. A *(nondeterministic strategy)* of Player σ in such a game $\mathscr{G}$ is a subgraph $\mathscr{S} = (W,F) \subseteq (V,E)$ satisfying the following conditions:

(1) If $v \in W \cap V_\sigma$, then vF is non-empty.
(2) If $v \in W \cap V_{1-\sigma}$ then $vF = vE$.

Here W is the region of $\mathscr{G}$ on which the strategy is defined, and F is the set of moves that are admitted by the strategy. A strategy $\mathscr{S} = (W,F)$ for Player σ is *deterministic* if $|vF| = 1$ for all $v \in W \cap V_\sigma$. A strategy $\mathscr{S}$ induces the set of those plays from the initial positions in $I \cap W$ whose moves are consistent with F. We call $\mathscr{S}$ well-founded if it does not admit any infinite plays; this is always the case on finite acyclic game graphs, but need not be the case otherwise. We are interested in *winning strategies* according to different winning conditions. Here, we shall mainly consider classical (first-order) safety and reachability conditions, and the second-order reachability conditions introduced in [7].

A *safety condition* for Player 0 is given by a set $L \subseteq V$ of 'losing' positions that Player 0 has to avoid, or dually, by its complement $S = V \setminus L$, the region of safe positions inside of which Player 0 has to keep the play. For convenience in game constructions we do not require that losing positions are terminal (but we could do so since deleting all outgoing edges from losing positions does not change anything relevant in the game). A play in a safety game is won by Player 0 if she can guarantee that the play never reaches a position $v \in L$. If Player 0 can, moreover, ensure, that the play reaches, after a finite number of steps a terminal position $v \in T \setminus L$, then she also wins the associated reachability game. The difference between reachability and safety conditions is relevant only in cases where infinite plays are possible. In first-order games, a winning strategy for a player is a strategy that guarantees that all plays consistent with it are won by that player. For a safety game this amounts to the following:

Definition 2. For a safety game $\mathscr{G}$, with safety condition $S \subseteq V$ and a set $X \subseteq I$ of initial positions, a strategy $\mathscr{S} = (W,F)$ for Player 0 is winning from X if $X \subseteq W \subseteq S$.

Traps. Two notions of fundamental importance for the algorithmic analysis of graph games, are *attractors* and *traps*. Intuitively the attractor (for Player 0) of a set $Y \subseteq V$ is the set of all positions from which Player σ has a strategy to ensure that the play reaches Y in a finite number of steps. The dual notion of a *trap* (for Player 1) encompasses those sets $Z \subseteq V$ for which Player 0 has a strategy to guarantee that every play starting at a position in Z remains inside Z.

Notice that in a game $\mathscr{G}$ with $I = V$ and a safety condition $S \subseteq V$ for Player 0, the winning region for Player 0 (i.e. the set of those positions from which she has a winning strategy) is precisely the maximal trap $Z \subseteq S$. For our analysis of games for inclusion logic and fixed-point logic, a specific variant of a trap will be relevant, which we call an I-trap.

Definition 3. For a game $\mathscr{G}$ with a set I of initial positions and a safety condition S for Player 0, an *I-trap* is a set $X \subseteq I$ of initial positions such that Player 0 has a strategy to ensure that every play starting in X remains inside S and avoids $I \setminus X$.

To put it differently, X is an I-trap in $(\mathscr{G}, S)$ if, and only if, Player 0 has a winning strategy from all positions in X for the safety game on $\mathscr{G}$ with losing positions $(V \setminus S) \cup (I \setminus X)$. Notice that an I-trap X is not a trap in $\mathscr{G}$, but it can be viewed as a kind of trap restricted to the set I, in the sense that Player 0 ensures that starting from X the only positions in I that are ever met in the play are those in X.

Clearly, the empty set is a trivial I-trap, and the union of two I-traps is again an I-trap, so there is a uniquely defined maximal I-trap for every safety game $(\mathscr{G}, S)$.

It is well-known that winning regions and winning strategies for reachability and safety games are computable in linear time in the size of the game graph (see e.g. [1, Chapt. 4]. Further one can, without loss of generality, restrict attention to deterministic strategies.

3 Second-order reachability games

While reachability and safety games are sufficient for many important applications, and in particular for evaluation games of first-order logic and the posGFP and posLFP-fragments of the least fixed-point logic LFP, they are, in general, not adequate for more complicated logics, such as full LFP and logics with team semantics. Model-checking games for the latter can be defined in terms of second-order reachability games.

Definition 4. A *second-order reachability condition* is a collection $\mathrm{Win} \subseteq \mathscr{P}(T)$ defining for each set $U \subseteq T$ of terminal positions whether it is a winning set for Player 0. A *consistent winning strategy* from $X \subseteq I$ for Player 0 for a second-order reachability game $\mathscr{G} = (V, V_0, V_1, T, I, E)$ with winning condition Win is a strategy $\mathscr{S} = (W, F)$ such that

(1) W is the set of nodes that are reachable from X via edges in F.
(2) $W \cap T \in \mathrm{Win}$.

Remark. The condition that W contains only nodes that are reachable from X by edges in the strategy is not needed for winning conditions that are downwards closed, and in particular for classical safety games. However, if Win is not downwards closed, then this condition is necessary to avoid the inclusion of unreachable nodes which could change a losing set of terminal nodes to a winning one.

As shown in [7], the problem whether a second-order reachability game, given by a finite game graph $\mathscr{G}$ with an oracle for Win, admits a consistent winning strategy for Player 0, is NP-complete. However, there are special cases of second-order reachability conditions for which the associated reachability games are efficiently solvable.

Universal-existential reachability conditions and the translation to safety games

Definition 5. We call a second-order reachability condition $\mathrm{Win} \subseteq \mathscr{P}(T)$ *universal-existential* if there exists a relation $R \subseteq T \times T$ such that

$$\mathrm{Win} = \{U \subseteq T : (\forall x \in U)(\exists y \in U)(x,y) \in R\}.$$

We shall see below that, for instance, the model-checking games for inclusion logic are second-order reachability games with universal-existential winning conditions. Further, the game graphs of such model-checking games are forests. Let now $\mathscr{G} = (V, V_0, V_1, T, I, E)$ be a second-order reachability game, played on a forest, where I is the set of roots of the forest, with a universal-existential winning condition $\mathrm{Win} = \{U \subseteq T : (\forall x \in U)(\exists y \in U)(x,y) \in R\}$.

We want to associate with $(\mathscr{G}, \mathrm{Win})$ a safety game $\mathscr{G}_{\mathrm{safe}}$ such that winning strategies for $(\mathscr{G}, \mathrm{Win})$ from $X \subseteq I$ correspond to winning strategies for $\mathscr{G}_{\mathrm{safe}}$ that ensure that X is an I-trap. The idea is to add to $\mathscr{G}$ moves of Player 0 for pairs $(s,t) \in R$ and moves of Player 1 taking the play back from t to ancestors in the forest. The nodes that Player 0 has to avoid in the safety game $\mathscr{G}_{\mathrm{safe}}$ are the nodes $s \in T$ without outgoing R-edges and the roots in $I \setminus X$.

To make this precise, we duplicate the nodes of $\mathscr{G}$, i.e. we add to $\mathscr{G}$ the set of vertices $V^* := \{v^* : v \in V\}$. In $\mathscr{G}_{\mathrm{safe}}$, we consider nodes $s \in T$ with $sR \neq \emptyset$ as positions of Player 0 and nodes $v^* \in V^*$ as positions of Player 1. We add moves so that Player 0 can move from s to t^* for any $(s,t) \in R$, and Player 1 can move from $v^* \in V^*$ either to v, or to the unique node u^* such that $(u,v) \in E$. We obtain the game graph

$$\mathscr{G}_{\mathrm{safe}} = (V \cup V^*, V_0 \cup \{s \in T : sR \neq \emptyset\}, V_1 \cup V^*, \tilde{T}, I, \tilde{E})$$

with $\tilde{T} := \{s \in T : sR = \emptyset\}$ and

$$\tilde{E} := E \cup \{(s,t^*) : (s,t) \in R\} \cup \{(v^*,u^*) : (u,v) \in E\} \cup \{(v^*,v), v \in V\}.$$

We obtain a safety game $\mathscr{G}_{\mathrm{safe}}$ where Player 0 has to avoid the positions in $\tilde{T}$.

Proposition 6 *Player 0 has a consistent winning strategy for the second-order reachability game $(\mathscr{G}, \mathrm{Win})$ from $X \subseteq I$ if, and only if, X is an I-trap in $\mathscr{G}_{safe}$.*

Proof. Let $\mathscr{S} = (W,F)$ be a consistent winning strategy for Player 0 from X for the game $(\mathscr{G},\mathrm{Win})$. For $U := W \cap T$ it follows that $U \in \mathrm{Win}$ and hence that $(\forall x \in U)(\exists y \in U)(x,y) \in R$.

We transform $\mathscr{S}$ into a strategy $\tilde{\mathscr{S}} = (\tilde{W},\tilde{F})$ for $\mathscr{G}_{\mathrm{safe}}$ with

$$\begin{aligned}\tilde{W} :=& W \cup \{v^* : v \in W\}\\ \tilde{F} :=& F \cup \{(s,t^*) : s,t \in U, (s,t) \in R\} \cup \{(v^*,v) : v \in W\} \cup \{(v^*,u^*) : (u,v) \in F\}.\end{aligned}$$

We claim that $\tilde{\mathscr{S}}$ is a winning strategy for $\mathscr{G}_{\mathrm{safe}}$ which moreover shows that X is an I-trap. We have to prove that $\tilde{\mathscr{S}}$ is indeed a strategy according to Definition 1 and additionally, avoids the positions in $\tilde{T} \cup (I \setminus X)$. Clearly, $\tilde{F} \subseteq \tilde{E} \cap \tilde{W} \times \tilde{W}$ and $X \subseteq W \subseteq \tilde{W}$.

It remains to verify the following conditions.

(1) If $w \in \tilde{W}$ is a node of Player 0, then $w\tilde{F}$ is non-empty.
For $w \in W \cap V_0$ this is clear since $wF \neq \emptyset$. Otherwise $w = s$ is a node in $W \cap T = U$ so there exists a node $t \in U$ with $(s,t) \in R$ and hence an edge $(s,t^*) \in \tilde{F}$.

(2) If $w \in \tilde{W}$ is a node of Player 1, then $w\tilde{F} = w\tilde{E}$.
For $w \in W \cap V_1$ we have that $w\tilde{F} = wF = wE = w\tilde{E}$. Otherwise $w = v^*$ for some $v \in W$. Then $w\tilde{E}$ consists of v and, unless v is a root, of the unique node u^* such that $(u,v) \in E$. Since $v \in W$ is reachable from its root by F-edges, also $u \in W$ and $(u,v) \in F$. It follows that edges (v^*,v) and (v^*,u^*) also belong to $\tilde{F}$. Hence in all cases $w\tilde{F} = w\tilde{E}$.

(3) $\tilde{W} \cap (\tilde{T} \cup (I \setminus X)) = \emptyset$.
Assume that there is a node $w \in \tilde{W} \cap \tilde{T}$. Then $w \in W \cap T = U$ which implies that there exists a node $t \in U$ with $(w,t) \in R$. But then $w \notin \tilde{T}$, contradicting our assumption. Finally, if $w \in W$, then $w \notin I \setminus X$ because all nodes in W are reachable from X by F-edges.

For the converse, consider any winning strategy $\tilde{\mathscr{S}} = (\tilde{W},\tilde{F})$ from X for Player 0 in the game $\mathscr{G}_{\mathrm{safe}}$ which avoids $\tilde{T}$ and $I \setminus X$, and its restriction $\mathscr{S} = (W,F)$ to $\mathscr{G}$, with $W = \tilde{W} \cap V$ and $F = \tilde{F} \cap E$.

Clearly $\tilde{W}$ cannot contain any position v^* such that v belongs to a tree whose root is in $I \setminus X$ because from such a position, Player 1 can move upwards to that root and win. Hence W only contains nodes that are reachable from a root in X and $\mathscr{S}$ is a strategy from X. To see that $\mathscr{S}$ is winning for the second-order reachability game $\mathscr{G}$, consider the set $U := W \cap T$ of terminal nodes in W. Since $\tilde{\mathscr{S}}$ is winning for the safety game, no node in U is terminal in $\mathscr{G}_{\mathrm{safe}}$. Hence $(\forall x \in U)(\exists y \in U)(x,y) \in R$ which means that $U \in \mathrm{Win}$. □

4 Logics and Their Games

4.1 First-order logic

We assume familiarity with first-order logic (FO) and briefly recall the construction of model-checking games for FO. We consider formulae with relational vocabulary $\tau = \{R_1, \ldots, R_m\}$ and assume that they are presented in negation normal form, i.e. built from literals (atomic formulae and their negations) by means of the propositional connectives $\vee$ and $\wedge$ and quantifiers $\exists$ and $\forall$.

For any such formula $\psi(\bar{x})$, let $\mathscr{T}(\psi)$ be its syntax tree whose nodes are the *occurrences* of the subformulae of ψ, with edges leading from any formula to its immediate subformulae, i.e. from $\varphi \vee \vartheta$ and $\varphi \wedge \vartheta$ to both φ and ϑ and from $\exists y\varphi$ and $\forall y\varphi$ to φ. The leaves of the tree are the nodes associated to literals.

For a formula $\psi(\bar{x})$ and a τ-structure $\mathfrak{A} = (A, R_1, \ldots, R_m)$, the model-checking game $\mathscr{G}(\mathfrak{A}, \psi)$ is obtained by taking an appropriate product of $\mathscr{T}(\psi)$ with the set of assignments mapping variables to elements of $\mathfrak{A}$. More precisely, the positions of the game are the pairs (φ, s) consisting of a node $\varphi \in \mathscr{T}(\psi)$ and an assignment $s : \mathrm{free}(\varphi) \to A$. Verifier (Player 0) moves from positions associated with disjunctions and with formulae starting with an existential quantifier. From a position $(\varphi \vee \vartheta, s)$, she moves to either (φ, s') or (ϑ, s'') where s', s'' are the restrictions of s to the free variables of φ and ϑ, respectively. From a position $(\exists y\varphi, s)$, Verifier can move to any position $(\varphi, s[y \mapsto a])$, where a is an arbitrary element of A. Dually, Falsifier (Player 1) makes corresponding moves for conjunctions and universal quantifications. If φ is a literal then the positions (φ, s) are terminal. The terminal positions are partitioned into the target sets T_0, T_1 of the two players, with $T_0 = \{(\varphi, s) \in T : \mathfrak{A} \models_s \varphi\}$ and $T_1 = \{(\varphi, s) \in T : \mathfrak{A} \models_s \neg\varphi\}$. Notice that since model-checking games for first-order logic are played on forests, it does not matter whether we consider them as reachability games, where Player σ has the objective to reach T_σ, or as safety games, where Player σ seeks to avoid $T_{1-\sigma}$.

4.2 Least fixed-point logic

Least fixed-point logic, denoted LFP, extends first order logic by least and greatest fixed points of definable relational operators. We will briefly recall some basic definitions here. For a more extensive introduction to LFP, we refer to [9].

Every formula $\psi(R, \bar{x})$, where R is a relation symbol of arity k and $\bar{x}$ is a tuple of k variables, defines, for any structure $\mathfrak{A}$ of matching vocabulary, an update operator $F^{\mathfrak{A}}_{\psi} : \mathscr{P}(A^k) \to \mathscr{P}(A^k)$ on the class of k-ary relations over the universe A of $\mathfrak{A}$, namely $F^{\mathfrak{A}}_{\psi} : R \mapsto \{\bar{a} : (\mathfrak{A}, R) \models \psi(R, \bar{a})\}$. If all occurrences of R in ψ are positive, then this operator is monotone in the sense that $R \subseteq R'$ implies $F^{\mathfrak{A}}_{\psi}(R) \subseteq F^{\mathfrak{A}}_{\psi}(R')$. It is well known that every monotone operator F has a least fixed point $\mathbf{lfp}(F)$ and a greatest fixed point $\mathbf{gfp}(F)$, with

$$\mathbf{lfp}(F) = \bigcap\{X : F(X) = X\} = \bigcap\{X : F(X) \subseteq X\}$$
$$\mathbf{gfp}(F) = \bigcup\{X : F(X) = X\} = \bigcup\{X : F(X) \supseteq X\},$$

which can also be constructed by transfinite induction.

LFP is defined by adding to the syntax of first order logic the following *fixed point formation rule:* If $\psi(R,\bar{x})$ is a formula of vocabulary $\tau \cup \{R\}$, in which the relational variable R occurs only positively, and if $\bar{x}$ is a tuple of variables such that the length of $\bar{x}$ matches the arity of R, then $[\mathbf{lfp}\, R\bar{x}.\, \psi](\bar{x})$ and $[\mathbf{gfp}\, R\bar{x}.\, \psi](\bar{x})$ are also formulae (of vocabulary τ).

The semantics of least fixed-point formulae in a structure $\mathfrak{A}$, providing interpretations for all free variables in the formula, is the following: $\mathfrak{A} \models [\mathbf{lfp}\, R\bar{x}.\, \psi](\bar{a})$ if $\bar{a}$ belongs to the least fixed point of the update operator defined by ψ on $\mathfrak{A}$. Similarly for greatest fixed points.

Note that in formulae $[\mathbf{lfp}\, R\bar{x}.\, \psi](\bar{x})$ one may allow ψ to have other free variables besides $\bar{x}$; these are called parameters of the fixed-point formula. However, at the expense of increasing the arity of the fixed-point predicates and the number of variables one can always eliminate parameters. For the construction of model-checking games it is convenient to assume that formulae are parameter-free.

The duality between least and greatest fixed point implies that for any ψ,

$$[\mathbf{gfp}\, R\bar{x}.\, \psi](\bar{x}) \equiv \neg[\mathbf{lfp}\, R\bar{x}.\, \neg\psi[R/\neg R]](\bar{x}).$$

Using this duality together with de Morgan's laws, every LFP-formula can be brought into *negation normal form*, where negation applies to atoms only.

Example 1 (Definability in safety games). Winning regions of reachability and safety games are definable by LFP-formulae of rather simple form. On game graphs $\mathscr{G}$ where the objective of Player 0 is to keep the play inside a given safe region $S \subseteq V$, and Player 1 wants to reach the set $L = V \setminus S$, the winning regions of the two players are uniformly definable by

$$\mathrm{win}(x) := [\mathbf{gfp}\, Wx.\, Sx \wedge (V_0x \to \exists y(Exy \wedge Wy)) \wedge (V_1x \to \forall y(Exy \to Wy))](x)$$
$$\mathrm{lose}(x) := [\mathbf{lfp}\, Wx.\, Lx \vee (V_1x \wedge \exists y(Exy \wedge Wy)) \vee (V_0x \wedge \forall y(Exy \to Wy))](x)$$

A simple modification of this construction gives a definition of I-traps in safety games. Let

$$\mathrm{itrap}(X,x) := [\mathbf{gfp}\, Yx.\, Sx \wedge (Ix \to Xx) \wedge (V_0x \to \exists y(Exy \wedge Yy)) \wedge (V_1x \to \forall y(Exy \to Yy))](x).$$

Then, for every safety game $\mathscr{G}$ and every set $X \subseteq I$ of initial positions we have that

$$(\mathscr{G}, X) \models \forall x(Xx \to \mathrm{itrap}(X,x)) \;\Leftrightarrow\; X \text{ is an } I\text{-trap in } \mathscr{G}.$$

The model-checking games for general LFP-formulae are *parity games*. These are games of possibly infinite duration, where each position is assigned a natural

number, called its priority, and an infinite play is won by Player 0 if the least priority seen infinitely often in the play is even.

Let ψ be an LFP-formula, which is assumed to be parameter-free, in negation normal form and in which distinct occurrences of fixed-point operators use distinct fixed-point variables. To construct the parity game $\mathscr{G}(\mathfrak{A},\psi)$, one extends the construction of the first-order model-checking game as follows: For every subformula of ψ of form $\vartheta := [\mathbf{fp}\, R\bar{x}.\, \varphi(R,\bar{x})](\bar{x})$ (where **fp** is either **lfp** or **gfp**) we add moves from positions (ϑ,s) to (φ,s), and from positions $(R\bar{y},s)$ to (φ,t) for the assignment t with $t(\bar{x}) = s(\bar{y})$. Since these moves are unique it makes no difference to which of the two players we assign the positions (ϑ,s) and $(R\bar{y},t)$. Priorities are assigned in such a way that positions $(R\bar{y},s)$, associated with fixed-point variables, get an even priority if R is a **gfp**-variable, and an odd priority if R is an **lfp**-variable. Further R gets a smaller (i.e. more significant) priority than R' if R' depends on R, i.e., if R occurs free in the formula defining R'. All other positions, associated with formulae that are not fixed-point atoms, get maximal (the least significant) priority. Thus the number of priorities needed in a parity game for a fixed-point formula ψ coincides with the alternation depth of least and greatest fixed points in ψ. For details, the proof of correctness, and for algorithmic and model-theoretic results based on parity games, see [9, Chapter 3.3].

4.3 The fragment of positive greatest fixed points

We denote by posGFP the fragment of LFP of formulae in negation normal form such that all its fixed-point operators are greatest fixed-points. Since all fixed-points are of the same kind, the priority assignment is trivial (all positions get priority 0), and the model-checking game $\mathscr{G}(\mathfrak{A},\psi)$ for a τ-structure $\mathfrak{A}$ and a formula $\psi \in$ posGFP is a safety game. The positions that Player 0 has to avoid are those of the form (α,s) where α is a τ-atom or a negated τ-atom such that $\mathfrak{A} \models_s \neg\alpha$. From positions associated with a **gfp**-variable R, the play is taken back to the fixed-point formula that defines R, and infinite plays correspond to successful infinite regeneration sequences of greatest fixed-points.

Proposition 7 *For every structure $\mathfrak{A}$, every formula $\psi(\bar{x})$ of* posGFP *and every assignment $s : \mathrm{free}(\psi) \to A$ we have that $\mathfrak{A} \models_s \psi(\bar{x})$ if, and only if, Player 0 has a winning strategy for the safety game $\mathscr{G}(\mathfrak{A},\psi)$ from the initial position (ψ,s).*

For the relationship between inclusion logic and fixed-point logic, we shall consider sentences in posGFP of vocabulary $\tau \cup \{X\}$ of the form

$$\vartheta := \forall\bar{x}(X\bar{x} \to \varphi(\bar{x}))$$

such that X occurs only positively in φ. The model checking game $\mathscr{G}((\mathfrak{A},X),\vartheta)$ is a safety game with initial position $(\vartheta,\emptyset)$ and safety condition

$$S(X) = \{(\eta, s) : \text{if } \eta \text{ is a } \tau\text{-literal, then } \mathfrak{A} \models_s \eta \text{ and}$$
$$\text{if } \eta = X\bar{y} \text{ then } s(\bar{y}) \in X \text{ and}$$
$$\text{if } \eta = \neg X\bar{x} \text{ then } s(\bar{x}) \notin X\}.$$

We modify these model-checking games by eliminating every explicit reference to the relation X and associate the model checking problem of whether $(\mathfrak{A}, X) \models \vartheta$ with a trap condition for a modified game $\mathscr{G}^*(\mathfrak{A}, \varphi)$. To do this, we identify every position of form $(X\bar{y}, t)$ with the position $(\varphi(\bar{x}), s)$ such that $s(\bar{x}) = t(\bar{y})$; this means that every edge in the game graph to a position $(X\bar{y}, t)$ is replaced by an edge to $(\varphi(\bar{x}), s)$, and the node $(X\bar{y}, t)$ is deleted. The set I of initial positions now consists of all pairs of form $(\varphi(\bar{x}), s)$ and the safety condition is simplified to

$$S^* := \{(\eta, s) : \text{ if } \eta \text{ is a } \tau\text{-literal, then } \mathfrak{A} \models_s \eta\}.$$

Given any interpretation for the relation X, let $X^* \subseteq I$ be the set of positions (φ, s) where $s(\bar{x}) \in X$.

Proposition 8 *The resulting game $\mathscr{G}^*(\mathfrak{A}, \varphi)$ has the property that*

$$(\mathfrak{A}, X) \models \forall \bar{x}(X\bar{x} \to \varphi(\bar{x})) \;\Leftrightarrow\; X^* \textit{ is an } I\textit{-trap in } \mathscr{G}^*(\mathfrak{A}, \varphi).$$

4.4 Logics with team semantics

Let $\mathfrak{A}$ be a structure of vocabulary τ with universe A. An *assignment* (into $\mathfrak{A}$) is a map $s : \mathscr{V} \to A$ whose domain $\mathscr{V}$ is a set of variables. Given such an assignment s, a variable y, and an element $a \in A$ we write $s[y \mapsto a]$ for the assignment with domain $\mathscr{V} \cup \{y\}$ that updates s by mapping y to a. A *team* is a set of assignments with the same domain. For a team X, a variable y, and a function $F : X \to \mathscr{P}(A)$, we write $X[y \mapsto F]$ for the set of all assignments $s[y \mapsto a]$ with $s \in X$ and $a \in F(s)$. Further we write $X[y \mapsto A]$ for the set of all assignments $s[y \mapsto a]$ with $s \in X$ and $a \in A$.

Team semantics, for a logic L, defines whether a formula $\psi \in L$ is satisfied by a team X in a structure $\mathfrak{A}$, written $\mathfrak{A} \models_X \psi$. We always assume formulae to be in negation normal form and require that the domain of X contains all free variables of ψ. Further we shall always make sure that the *locality principle* holds, saying that the meaning of a formula can only depend on the variables actually occurring in it. More precisely, if $Y = X \restriction \text{free}(\psi)$ is the restriction of the team X to the free variables of ψ then $\mathfrak{A} \models_X \psi$ if, and only if, $\mathfrak{A} \models_Y \psi$. A special case is the empty team which satisfies all formulae: $\mathfrak{A} \models_\emptyset \psi$ for all $\mathfrak{A}$ and all ψ. The locality principle implies that a sentence ψ (i.e. a formula without free variables) is true for a non-empty team X if, and only if, it is true for the team $\{\emptyset\}$ consisting only of the empty assignment. Thus, as it should be and as in logics with Tarski semantics, the truth of a sentence just depends on the structure in which it is evaluated. For sentences we then write $\mathfrak{A} \models \psi$ to denote that $\mathfrak{A} \models_{\{\emptyset\}} \psi$. This allows us to directly compare the expressive power of sentences between logics with team semantics and logics with

Tarski semantics. For open formulae, the situation is different and will be discussed later

For the operators of first-order logic (FO) the semantic rules are the following.

(1) If ψ is an atom $x = y$ or $Rx_1 \dots x_m$ or the negation of such an atom, then $\mathfrak{A} \models_X \psi$ if, and only if, $\mathfrak{A} \models_s \psi$ (in the sense of Tarski semantics) for all $s \in X$.
(2) $\mathfrak{A} \models_X (\varphi \wedge \vartheta)$ if, and only if, $\mathfrak{A} \models_X \varphi$ and $\mathfrak{A} \models_X \vartheta$.
(3) $\mathfrak{A} \models_X (\varphi \vee \vartheta)$ if, and only if, there exist teams Y, Z with $X = Y \cup Z$ such that $\mathfrak{A} \models_Y \varphi$ and $\mathfrak{A} \models_Z \vartheta$.
(4) $\mathfrak{A} \models_X \forall y \varphi$ if, and only if, $\mathfrak{A} \models_{X[y \mapsto A]} \varphi$.
(5) $\mathfrak{A} \models_X \exists y \varphi$ if, and only if, there is a map $F : X \to (\mathscr{P}(A) \setminus \{\emptyset\})$ such that $\mathfrak{A} \models_{X[y \mapsto F]} \varphi$.

Remark. Clause (5) giving semantics to existential quantifiers might seem surprising at first sight since it permits the choice of an arbitrary non-empty set of witnesses for an existentially quantified variable rather than a single witness (for each $s \in X$). What we use here has been called *lax semantics* in [5], as opposed to the more common *strict semantics*. For disjunctions (clause (3)) there is also a strict variant, requiring that the team X is split into *disjoint* subteams Y and Z. For first-order logic, and also for dependence logic, the difference is immaterial since the two semantics are equivalent. However, this is not the case for other logics of dependence and independence, in particular for independence logic and inclusion logic. In these cases, only the lax semantics is appropriate since it preserves the locality principle whereas the strict semantics violates this principle. In game-theoretic terms the difference between strict and lax semantics corresponds to the difference between deterministic and nondeterministic strategies, and it turns out that model-checking games for inclusion logic and independence logic do not admit deterministic winning strategies.

For first-order logic itself, team semantics does not provide anything new since a first order formula is true for a team X if, and only if, it is true in the sense of Tarski semantics, for all individual assignments $s \in X$:

$$\mathfrak{A} \models_X \psi \;\Leftrightarrow\; \mathfrak{A} \models_{\{s\}} \psi \;\Leftrightarrow\; \mathfrak{A} \models_s \psi.$$

This changes radically, when atomic statements on teams which express properties of dependence or independence are added to the logic. The most common examples of such properties are the following.

Dependence: A dependence atom has the form $=(x_1 \dots, x_m, y)$. It is true in a team X if all assignments s, s' in X that agree on the variables $x_1, \dots, x_m$ also have the same value for y. Dependence logic is first-order logic with dependence atoms. An important property of dependence logic is downwards closure: If a formula is satisfied by a team X then it is also satisfied by all subteams $Y \subseteq X$. Formulae of dependence logic are equivalent to sentences of existential second-order logic,

with an additional predicate for the team that may occur only negatively. See [14, 12] for further results.

Independence: Independence atoms come in several variants. Intuitively two variables x and y are independent, denoted $x \perp y$, if acquiring more knowledge about one does not provide any additional knowledge about the other, which means that values for (x,y) appear in all conceivable combinations: if values (a,b) and (a',b') occur for (x,y), then so do (a,b') and (a',b). To make this sufficiently general, we proposed in [8] the general conditional independence atom $\bar{y} \perp_{\bar{x}} \bar{z}$, for arbitrary tuples $\bar{x},\bar{y},\bar{z}$ of variables, which is true in team X if, and only if, for all assignments $s,s' \in X$ such that $s(\bar{x}) = s'(\bar{x})$ there is an assignment $s'' \in X$ with $s''(\bar{x}) = s(\bar{x})$, $s''(\bar{y}) = s(\bar{y})$ and $s''(\bar{z}) = s'(\bar{z})$. Independence logic is strictly more powerful than dependence logic and it is not downwards closed for teams. Galliani [5] has shown that independence logic is equivalent with existential second-order logic. Furthermore the conditional independence atoms can be eliminated in favour of pure independence atoms $\bar{x} \perp \bar{y}$.

Exclusion and inclusion: An exclusion atom $(\bar{x} \mid \bar{y})$ expresses that the values of $\bar{x}$ in the given team are disjoint from the values of $\bar{y}$. Inclusion atoms $(\bar{x} \subseteq \bar{y})$ state that all values for $\bar{x}$ in the given team occur also as values for $\bar{y}$ in X. It has been proved by Galliani [5] that first-order logic with both inclusion and exclusion atoms is equivalent with independence logic.

There are many other variants of atomic dependence or independence properties. In [7] we have shown that there is a uniform construction of model-checking games for logics with team semantics, based on the notion of a second-order reachability game. For every formula $\psi(\bar{x})$ (which we always assume to be in negation normal form) and every structure $\mathfrak{A}$ we define the game $\mathscr{G}(\mathfrak{A},\psi)$ as follows. The game graph is defined in precisely the same way as in the case of first-order logic. In particular, $\mathscr{G}(\mathfrak{A},\psi)$ is a forest, consisting of trees with roots (ψ,s) for all assignments $s : \mathrm{free}(\psi) \to A$. In the case that ψ is a sentence, we only have the empty assignment to consider, and the game graph is a tree. Given a team X of assignments $s : \mathrm{free}(\psi) \to A$, the relevant set of initial positions is $I(X) := \{(\psi,s) : s \in X\}$.

Although the game graphs for logics with team semantics are defined as for first-order logic, the winning conditions are very different. Indeed, model checking games for logics with team semantics are special cases of second-order reachability games.

To describe the second-order winning condition, we observe that any set W of nodes in a model-checking game for ψ associates to a formula $\varphi \in \mathscr{T}(\psi)$ a team

$$\mathrm{Team}(W,\varphi) := \{s : \mathrm{free}(\varphi) \to A : (\varphi,s) \in W\}.$$

We now say that a set U of terminal positions is a winning set if, for every literal α,

$$\mathfrak{A} \models_{\mathrm{Team}(U,\alpha)} \alpha.$$

Notice that for literals α for which no pair (α, s) appears in U, this is trivially satisfied because for the logics that we consider here, the empty team satisfies all formulae.

Described more abstractly, the model-checking game for ψ on $\mathfrak{A}$ consists of the game graph $\mathcal{G}(\mathfrak{A}, \psi) = (V, V_0, V_1, T, I, E)$ and the second-order reachability condition Win consisting of all sets $U \subseteq T$ such that

$$\mathfrak{A} \models_{\mathrm{Team}(U,\alpha)} \alpha, \text{ for all literals } \alpha.$$

Thus, a consistent winning strategy $S = (W, F)$ of Player 0 for $\mathcal{G}(\mathfrak{A}, \psi)$, from the set $I(X) \subseteq I$ of those initial positions that are associated with a team X, has the property that, for every literal φ, the team $\mathrm{Team}(S, \varphi) := \mathrm{Team}(W, \varphi) = \{s : (\varphi, s) \in W\}$ satisfies φ. As proved in [7] this then extends beyond the literals to all formulae in $\mathcal{T}(\psi)$ and in particular to the formula ψ itself. Let L be a logic with team semantics.

Theorem 9 *For every structure $\mathfrak{A}$, every formula $\psi(\bar{x}) \in L$ and every team X with domain $\mathrm{free}(\psi)$ we have that $\mathfrak{A} \models_X \psi$ if, and only if, Player 0 has a consistent winning strategy $S = (W, F)$ for $\mathcal{G}(\mathfrak{A}, \psi)$ from $I(X)$, with $\mathrm{Team}(S, \psi) = X$.*

Proof. We proceed by induction on ψ. First, let ψ be a literal. The game $\mathcal{G}(\mathfrak{A}, \psi)$ is just the set of isolated nodes (ψ, s) for all possible assignments s. If $\mathfrak{A} \models_X \psi$ then let $W_\psi = \{(\psi, s) : s \in X\}$ and $F_\psi = \emptyset$. Clearly $S_\psi = (W_\psi, F_\psi)$ is a consistent winning strategy in $\mathcal{G}(\mathfrak{A}, \psi)$ with $\mathrm{Team}(S_\psi, \psi) = X$. If $\mathfrak{A} \not\models_X \psi$ then for any consistent winning strategy S with $\mathfrak{A} \models_{\mathrm{Team}(S,\psi)} \psi$ it must be the case that $\mathrm{Team}(S, \psi) \neq X$.

Next suppose that $\psi = \eta \vee \vartheta$. If $\mathfrak{A} \models_X \eta \vee \vartheta$ then there exist teams Y, Z with $X = Y \cup Z$ such that $\mathfrak{A} \models_Y \eta$ and $\mathfrak{A} \models_Z \vartheta$. By induction hypothesis there are consistent winning strategies $S_\eta = (W_\eta, F_\eta)$ in $\mathcal{G}(\mathfrak{A}, \eta)$ and $S_\vartheta = (W_\vartheta, F_\vartheta)$ in $\mathcal{G}(\mathfrak{A}, \vartheta)$ with $\mathrm{Team}(S_\eta, \eta) = Y$ and $\mathrm{Team}(S_\vartheta, \vartheta) = Z$. We obtain a consistent winning strategy $S_\psi = (W_\psi, F_\psi)$ in $\mathcal{G}(\mathfrak{A}, \psi)$ by setting $W_\psi := W_\eta \cup W_\vartheta \cup \{(\psi, s) : s \in X\}$ and $F_\psi := F_\eta \cup F_\vartheta \cup \{((\psi, s), (\eta, s')) : s \in Y, s' = s \restriction_{\mathrm{free}(\eta)}\} \cup \{((\psi, s), (\vartheta, s')) : s \in Z, s' = s \restriction_{\mathrm{free}(\vartheta)}\}$. Obviously $\mathrm{Team}(S_\psi, \psi) = X$ and since $X = Y \cup Z$ the strategy S_ψ admits, from every point $(\psi, s) \in W_\psi$ at least one edge to either (η, s') or (ϑ, s'). Conversely, every consistent winning strategy $S_\psi = (W_\psi, F_\psi)$ for Player 0 with $\mathrm{Team}(S_\psi, \psi) = X$ induces a decomposition $X = Y \cup Z$ where Y contains those $s \in X$ such that F_ψ admits a move from (ψ, s) to $(\eta, s \restriction_{\mathrm{free}(\eta)})$ and analogously for Z and ϑ. By induction hypothesis it follows that $\mathfrak{A} \models_Y \eta$ and $\mathfrak{A} \models_Z \vartheta$ and therefore $\mathfrak{A} \models_X \psi$.

The arguments for $\psi = \eta \wedge \vartheta$ are analogous (and in fact even simpler).

Let us now consider formulae $\psi = \exists y \varphi$. If $\mathfrak{A} \models_X \psi$ then there is a function $F : X \to (\mathcal{P}(A) \setminus \{\emptyset\})$ such that $\mathfrak{A} \models_{X[y \mapsto F]} \varphi$. By induction hypothesis, Player 0 has a consistent winning strategy $S_\varphi = (W_\varphi, F_\varphi)$ with $\mathrm{Team}(S_\varphi, \varphi) = X[y \mapsto F]$. We obtain a consistent winning strategy $S_\psi = (W_\psi, F_\psi)$ by setting $W_\psi := W_\varphi \cup \{(\psi, s) : s \in X\}$ and $F_\psi = F_\varphi \cup \{((\psi, s), (\varphi, s[y \mapsto a])) : s \in X, a \in F(s)\}$. Obviously, $\mathrm{Team}(S_\psi, \psi) = X$. Conversely, a consistent winning strategy $S_\psi = (W_\psi, F_\psi)$ with $\mathrm{Team}(S_\psi, \psi) = X$ requires that from every node (ψ, s) with $s \in X$ the set $(\psi, s)F_\psi$ of admissible

successor nodes is non-empty. Let $F(s) := \{a \in A : (\varphi, s[y \mapsto a]) \in (\psi, s)F_\psi\}$. By induction hypothesis $\mathfrak{A} \models_{X[y \mapsto F]} \varphi$ and hence $\mathfrak{A} \models_X \psi$.

Again the arguments for formulae $\forall y\varphi$ are analogous. □

4.5 Inclusion logic

We now turn to inclusion logic, which is a specific case of a logic with team semantics. We recall the definition.

Definition 10. A team X satisfies an *inclusion atom* $\bar{x} \subseteq \bar{y}$ if for all $s \in X$ there is an $s' \in X$ with $s(\bar{x}) = s'(\bar{y})$. Inclusion logic is the extension of first-order logic with team semantics by inclusion atoms.

By structural induction, it is easy to verify that formulae $\varphi(\bar{x})$ of inclusion logic are closed under union of teams. For every structure $\mathfrak{A}$ and any collection $\{X_i : i \in I\}$ of teams such that $\mathfrak{A} \models_{X_i} \varphi$ for all $i \in I$ we also have that $\mathfrak{A} \models_X \varphi$ for $X = \bigcup\{X_i : i \in I\}$. Thus, there exists, for every structure $\mathfrak{A}$ a unique maximal team $X_{\max}$ with $\mathfrak{A} \models_{X_{\max}} \varphi$.

We next exhibit important examples for the power of inclusion logic, showing that winning regions, traps, and I-traps of safety games are definable in this logic. Further the trap-formula also reveals the technique of copying values from one variable to another one which is often necessary for dealing with disjunctions in the intended way. To simplify notation we identify a relation $Y \subseteq V^k$ with the team of all those assignments $s : \{x_1, \ldots, x_k\} \to V$ such that $s(\bar{x}) := (s(x_1), \ldots, s(x_k)) \in Y$.

For safety games $\mathscr{G} = (V, V_0, V_1, T, I, E)$ with safety condition $S \subseteq V$, we construct the formulae

$$\mathrm{trap}(x) := Sx \wedge \exists z(z \subseteq x \wedge (V_0 x \to \exists y(Exy \wedge y \subseteq z)) \wedge (V_1 x \to \forall y(Exy \to y \subseteq z))),$$
$$\mathrm{itrap}(x) := \exists y(x \subseteq y \wedge \mathrm{trap}(y) \wedge (Iy \to y \subseteq x)).$$

Here (and elsewhere) implications $(\alpha \to \varphi)$, for first-order literals α, are just meant as a different notation for $(\neg\alpha \vee \varphi)$.

Proposition 11 *For every game graph $\mathscr{G}$, every safety condition $S \subseteq V$ and every set $X \subseteq V$, we have that $(\mathscr{G}, S) \models_X \mathrm{trap}(x)$ if, and only if, X is a trap for Player 1, i.e., Player 0 has a winning strategy that keeps every play from X inside X. Further $X \subseteq I$ is an I-trap in $(\mathscr{G}, S)$ if, and only if, $(\mathscr{G}, S) \models_X \mathrm{itrap}(x)$.*

Proof. A set $X \subseteq V$ is a trap for Player 1 in $\mathscr{G}$ if, and only if, $X \subseteq S$ and there exists a set of edges $F \subseteq (X \times X) \cap E$ such that, for all $v \in V_0 \cap X$ there exists a node $w \in X$ with $(v, w) \in F$, for all $v \in V_1 \cap X$ and all edges $(v, w) \in E$ it holds that also $w \in X$. We claim that given such an X and F we can show that $(\mathscr{G}, S) \models_X \mathrm{trap}(x)$. Let XX be the team of assignments $s : (x, z) \to (v, v')$ such that $v, v' \in X$. It suffices to prove that $\mathscr{G} \models_{XX} (V_0 x \to \exists y(Exy \wedge y \subseteq z))$ and $\mathscr{G} \models_{XX} V_1 x \to \forall y(Exy \to y \subseteq z))$. For the first claim we split the team XX into the subteams $\overline{V_0}X = \{s \in XX : s(x) \notin V_0\}$ and

$V_0X = \{s \in XX : s(x) \in V_0\}$. Trivially, $\overline{V_0}X$ satisfies $\neg V_0x$. To prove that V_0X satisfies $\exists y(Exy \wedge y \subseteq z)$, the team V_0X is expanded to $V_0XY = \{s : (x,z,y) \mapsto (v,v',w) : v \in V_0, v' \in X, (v,w) \in F\}$ and we claim that $\mathscr{G} \models_{V_0XY} Exy \wedge y \subseteq z$. Since $F \subseteq E$, the atom Exy is clearly satisfied, and for the inclusion atom we find, for each $s : (x,z,y) \mapsto (v,v',w)$ the assignment $s' : (x,z,y) \mapsto (v,w,w)$ so that $s(y) = s'(z)$ and s' is in V_0XY as well (because $w \in X$). The reasoning for $\mathscr{G} \models_{XX} V_1x \to \forall y(Exy \to y \subseteq z))$ is analogous. We get a team V_1XX which has to be universally expanded, by values for y, to a team $V_1XY = \{s : (x,z,y) \mapsto (v,v',w) : v \in V_1, v' \in X, w \in V\}$. This team is then split into, on one hand, the subteam of those assignments with $(v,w) \notin E$, to satisfy the literal $\neg Exy$, and, on the other hand, the remaining set of assignments. But in the remaining team all assignments $s : (x,z,y) \mapsto (v,v',w)$ satisfy $(v,w) \in F$ and hence $w \in X$. Thus we can again map s to $s' : (x,z,y) \mapsto (v,w,w)$ to make sure that $y \subseteq z$ is satisfied. Notice that without copying all values for x in X also as values for z this reasoning would not work.

For the converse, assume that $(\mathscr{G},S) \models_X \mathrm{trap}(x)$. Clearly $X \subseteq S$. If X were not a trap, then there would be a node $v \in X$ such that either $v \in V_0$ and no edge from v leads to a node in X, or $v \in V_1$ and at least one edge from v leaves X. In both cases the formula is false for X, hence we would have a contradiction.

Finally $(\mathscr{G},S) \models_X \mathrm{itrap}(x)$ if, and only if, there is a trap Y in $(\mathscr{G},S)$ such that $X \subseteq Y$ and $Y \cap I \subseteq X$. This is the case if, and only if X is an I-trap in $(\mathscr{G},S)$. □

Let us now look at games for inclusion logic. In the second-order reachability games for formulae of inclusion logic, the terminal positions are associated either with first-order literals or with inclusion atoms of form $\bar{x} \subseteq \bar{y}$.

Proposition 12 *The winning conditions of second-order reachability model checking games for inclusion logic are universal-existential.*

Proof. The winning condition in a game $\mathscr{G}(\mathfrak{A}, \psi)$ consists of those sets $U \subseteq T$ such that, for every literal φ,

$$\mathfrak{A} \models_{\mathrm{Team}(U,\varphi)} \varphi.$$

We have to find a relation $R \subseteq T \times T$ such that a set $U \subseteq T$ is winning if, and only if, $(\forall x \in U)(\exists y \in U)R(x,y)$. Positions in T are either of the form (φ,s) where φ is a first-order literal, or of the form $(\bar{x} \subseteq \bar{y}, s)$. We define R to contain all loops $((\varphi,s)(\varphi,s))$ for first-order literals φ such that $\mathfrak{A} \models_s \varphi$, and all edges $((\bar{x} \subseteq \bar{y}, s), (\bar{x} \subseteq \bar{y}, t))$ such that $t(\bar{y}) = s(\bar{x})$. Then clearly, for all $U \subseteq T$ it holds that

$$\mathfrak{A} \models_{\mathrm{Team}(U,\varphi)} \varphi \text{ for all literals } \varphi \quad \Leftrightarrow \quad (\forall x \in U)(\exists y \in U)R(x,y).$$

□

By the construction in Sect. 3 we thus obtain safety games for inclusion logic and associate the teams satisfying the formula with the I-traps in the game. For further reference, let us describe this in a more detailed way. Given a structure $\mathfrak{A}$ and a formula $\psi(\bar{x})$ of inclusion logic we obtain a safety game $\mathscr{G}_{\mathrm{safe}}(\mathfrak{A}, \psi)$, with the set I of initial positions consisting of all pairs (ψ,s) with assignments $s : \mathrm{free}(\psi) \to A$,

and the safety winning condition S excluding the pairs (φ, s) where φ is a first-order literal with $\mathfrak{A} \models_s \neg\varphi$. The game consists of

- the forest $\mathscr{G} = \mathscr{G}(\mathfrak{A}, \psi)$ defined as for first-order logic, with positions of form (φ, s),
- a copy $\mathscr{G}^*$ of this forest, with positions $(\varphi, s)^*$, in which Player 1 either moves upwards the forest, or from $(\varphi, s)^*$ to the corresponding position (φ, s) in $\mathscr{G}$, and
- moves of Player 0 from positions in $\mathscr{G}$ associated to inclusion atoms, into $\mathscr{G}^*$. Such moves go from $(\bar{x} \subseteq \bar{y}, s)$ to positions $(\bar{x} \subseteq \bar{y}, t)^*$ subject to the condition that $t(\bar{y}) = s(\bar{x})$.

The objective of Player 0 in this game is to keep the play inside S and to avoid the initial positions (ψ, s) with $s \notin X$. We formulate this in terms of I-traps.

Proposition 13 *For every structure $\mathfrak{A}$, every formula $\psi(\bar{x})$ of inclusion logic, and every team X with domain* free(ψ)*, we have that $\mathfrak{A} \models_X \psi$ if, and only if $X^* := \{(\psi, s) \in I : s \in X\}$ is an I-trap in $\mathscr{G}_{safe}(\mathfrak{A}, \psi)$.*

5 Interpretations

The notion of an interpretation is fundamental in mathematical logic. Interpretations are used to define a copy of a structure inside another one, and thus permit us to transfer definability, decidability, and complexity results between theories. Here we shall use interpretations of model-checking games as a method to embed one logic inside another and to provide normal forms and complete problems for logics. A bit of care is necessary for the application of interpretations in the context of team semantics.

The interpretations that we consider are classical first-order interpretations (with Tarski semantics); in fact we are interested in interpretations that are given by very simple formulae, namely quantifier-free ones. However, we shall apply these simple interpretations as translations among formulae of more powerful logics, such as greatest fixed-point logic and inclusion logic.

For every first-order formula $\varphi(x_1, \ldots x_k)$ and every structure $\mathfrak{A}$, we write $\varphi^{\mathfrak{A}}$ for the relation defined by $\varphi(\bar{x})$ on $\mathfrak{A}$, i.e., $\varphi^{\mathfrak{A}} := \{\bar{a} \in A^k : \mathfrak{A} \models \varphi(\bar{a})\}$.

Definition 14. Let L be a fragment of first-order logic, let σ, τ be vocabularies, where $\tau = \{R_1, \ldots, R_m\}$ is relational, and let r_i be the arity of R_i. A *(k-dimensional) $L[\sigma, \tau]$-interpretation* is given by a sequence I of formulae in $L(\sigma)$ consisting of

- $\delta(x_1, \ldots, x_k)$, called the domain formula,
- $\varepsilon(x_1, \ldots, x_k, y_1, \ldots, y_k)$, called the equality formula, and,
- for every relation symbol $R \in \tau$ (of arity r), a formula $\psi_R(\bar{x}_1, \ldots, \bar{x}_r)$ (of arity kr).

An $L[\sigma, \tau]$-interpretation induces two mappings, one between structures, and the other one between formulae. For a τ-structure $\mathfrak{B}$ and a σ-structure $\mathfrak{A}$, we say that I

interprets $\mathfrak{B}$ in $\mathfrak{A}$ (in short, $I(\mathfrak{A}) = \mathfrak{B}$) if there exists a surjective map $h : \delta^{\mathfrak{A}} \to B$, called the *coordinate map*, such that

- for all $\overline{a}, \overline{a}' \in \delta^{\mathfrak{A}}$,
$$\mathfrak{A} \models \varepsilon(\overline{a}, \overline{a}') \iff h(\overline{a}) = h(\overline{a}');$$
- for every relation R of $\mathfrak{B}$ and all $\overline{a}_1, \ldots, \overline{a}_r \in \delta^{\mathfrak{A}}$,
$$\mathfrak{A} \models \psi_R(\overline{a}_1, \ldots, \overline{a}_r) \iff (h(\overline{a}_1), \ldots, h(\overline{a}_r)) \in R,$$
i.e. $h^{-1}(R) = (\delta^{\mathfrak{A}})^r \cap \psi_R^{\mathfrak{A}}$.

Hence $I = \langle \delta, \varepsilon, \psi_{R_1}, \ldots, \psi_{R_m} \rangle$ defines (together with the function $h : \delta^{\mathfrak{A}} \to B$) an interpretation of $\mathfrak{B} = (B, R_1, \ldots, R_m)$ in $\mathfrak{A}$ if, and only if, $\varepsilon(\overline{x}, \overline{y})$ defines a congruence on the structure $(\delta^{\mathfrak{A}}, \psi_{R_1}^{\mathfrak{A}}, \ldots, \psi_{R_m}^{\mathfrak{A}})$ and h is an isomorphism between the quotient structure $(\delta^{\mathfrak{A}}, \psi_{R_1}^{\mathfrak{A}}, \ldots, \psi_{R_m}^{\mathfrak{A}})/\varepsilon^{\mathfrak{A}}$ and $\mathfrak{B}$.

Besides the mapping $\mathfrak{A} \mapsto I(\mathfrak{A})$ from σ-structures to τ-structures, I also defines a mapping from τ-formulae to σ-formulae. With every τ-formula φ it associates a σ-formula φ^I, which is obtained by replacing every variable x by a k-tuple $\overline{x}$ of variables, by replacing every quantifier Qx by a quantifier $Q\overline{x}$ over k-tuples, relativized to $\delta(\overline{x})$, by replacing equalities $u = v$ by $\varepsilon(\overline{u}, \overline{v})$, and by replacing every atom $Ru_1 \ldots u_r$ by the corresponding formula $\psi_R(\overline{u}_1, \ldots, \overline{u}_r)$. In the case of fixed-point formulae or second-order-formulae, we may have relation variables Y (of some arity r) which we have to translate into corresponding relation variables Y^* of arity kr.

Most of the common logics (with Tarski semantics), including FO, LFP and its gfp-fragment, second-order logic etc., are closed under interpretations, that is, for every formula φ and every (first-order)-interpretation I, also φ^I is a formula of the same logic.

The semantics of these transformations is described by the Interpretation Lemma, which we formulate for formulae $\varphi(Y_1, \ldots, Y_m, x_1, \ldots, x_n)$ which may contain free relation variables Y_i and free element variables x_j. A k-dimensional interpretation I with coordinate map $h : \delta^{\mathfrak{A}} \to B$ induces for every relation $Y_i \subseteq B^r$ the relation $Y_i^* := h^{-1}(Y_i) \subseteq A^{kr}$ and for every assignment $s : \{x_1, \ldots, x_n\} \to B$, the set of assignments $h^{-1}s$ consisting of all $t : \{x_{ij} : 1 \leq i \leq n, 1 \leq j \leq k\} \to A$ such that, for all $i \leq n$, $t(x_{i1}, \ldots x_{ik}) \in h^{-1}(s(x_i))$.

Lemma 15 (Interpretation Lemma) *Let I be an $L[\sigma, \tau]$-interpretation with coordinate map h, let $\mathfrak{A}$ be a σ-structure, and let $\varphi = \varphi(Y_1, \ldots, Y_m, x_1, \ldots, x_n)$ be a formula of vocabulary τ with free relation variables $Y_1, \ldots, Y_m$ and free element variables $x_1, \ldots, x_n$. Then for every tuple $Y_1, \ldots Y_m$ of relations over $I(\mathfrak{A})$, every assignment $s : \{x_1, \ldots, x_n\} \to B$ and every assignment $t \in h^{-1}(s)$ we have that*

$$(\mathfrak{A}, Y_1^*, \ldots, Y_m^*) \models_t \varphi^I \iff (I(\mathfrak{A}), Y_1, \ldots, Y_m) \models_s \varphi.$$

In particular, for every τ-sentence φ, we have that $\mathfrak{A} \models \varphi^I \iff I(\mathfrak{A}) \models \varphi$.

For formulae with team semantics, the translation is a little bit more delicate, since we need to consider the transformations of teams under interpretations and

make sure that the atomic properties of teams are compatible with these transformations. In the presence of congruences, this may require to change the atomic formulae.

For inclusion logic, however, such complications do not arise. The interpretation I translates an inclusion statement $\eta := (x_1,\dots,x_m \subseteq y_1,\dots,y_m)$ on m-tuples of variables into an inclusion statement $\eta^I := (\bar{x}_1,\dots \bar{x}_m \subseteq \bar{y}_1,\dots,\bar{y}_m)$ on mk-tuples. For a team X of assignments s mapping variables x_i into the interpreted structure $I(\mathfrak{A})$, we get the team $h^{-1}(X) = \bigcup\{h^{-1}(s) : s \in X\}$ taking values in $\mathfrak{A}$.

Lemma 16 *An inclusion atom $\eta := (\bar{x} \subseteq \bar{y})$ holds in a team X with values in $I(\mathfrak{A})$ if, and only if, η^I holds in the team $h^{-1}(X)$.*

Proof. For simplicity of notation we just consider inclusion atoms of form $\eta := (x \subseteq y)$. Suppose that the translated inclusion statement $(\bar{x} \subseteq \bar{y})$ holds in $h^{-1}(X)$. Take any $s \in X$. We have to prove that there exists a $s' \in X$ with $s'(y) = s(x)$. For every $t \in h^{-1}(s)$ there exists a $t' \in h^{-1}(X)$ with $t'(\bar{y}) = t(\bar{x})$. For $s' = h(t') \in X$ we have that $s'(y) = h(t'(\bar{y})) = h(t(\bar{x})) = s(x)$.

For the converse, assume that $(x \subseteq y)$ holds in X. For every $t \in h^{-1}(X)$ we can choose some $s = h(t) \in X$. By assumption there exists an $s' \in X$ with $s'(y) = s(x)$. Since $h(t(\bar{x})) = s(x) = s'(y)$ there exists a $t' \in h^{-1}(s') \subseteq h^{-1}(X)$ with $t'(\bar{y}) = t(\bar{x})$. Thus $(\bar{x} \subseteq \bar{y})$ holds in $h^{-1}(X)$.

As above this extends to a translation from arbitrary formulae φ of inclusion logic on $I(\mathfrak{A})$ into formulae φ^I of inclusion logic on $\mathfrak{A}$.

Lemma 17 (Interpretation Lemma for Inclusion Logic) *Let I be a quantifier-free first-order interpretation, mapping a structure $\mathfrak{A}$ to $I(\mathfrak{A})$ with coordinate map h, and let $\varphi(\bar{x})$ be a formula of inclusion logic. Then for any team X with values in $I(\mathfrak{A})$ we have that*

$$\mathfrak{A} \models_{h^{-1}(X)} \varphi^I \;\Leftrightarrow\; I(\mathfrak{A}) \models_X \varphi.$$

6 Interpretability of game graphs

We now prove that for every formula there is a uniform interpretation of the model checking games for that formula in the structure in which the formula is evaluated. We first explain the construction for first-order formulae, but it immediately carries over to stronger logics such as the least fixed-point logic LFP and its fragments, and to logics with team semantics.

Proposition 18 *For every formula $\psi(\bar{x}) \in \mathrm{FO}(\tau)$ there exists a quantifier-free interpretation I_ψ which, for every τ-structure $\mathfrak{A}$ with at least two elements, interprets the game graph $\mathscr{G}(\mathfrak{A},\psi)$ in $\mathfrak{A}$.*

Proof. Let $\mathrm{Sf}(\psi)$ be the set of subformulae of $\psi(\bar{x})$ and let $\mathscr{T}(\psi) = (\mathrm{Sf}(\psi), E_\psi)$ be its syntax-tree. Let $x_1 \dots, x_k$ be the variables occurring in ψ. Recall that an *equality*

type in m-variables $u_1, \dots, u_m$ is a maximally consistent conjunction of equalities $u_i = u_j$ and inequalities $u_i \neq u_j$. Let E_m be the set of equality types in $u_1 \dots u_m$ (up to equivalence). Choose m sufficiently large so that $|E_m| \geq |\mathrm{Sf}(\psi)|$ and fix for every formula $\varphi \in \mathrm{Sf}(\psi)$ a separate equality type $e_\varphi \in E_m$.

A node (φ, s) of the game graph $\mathscr{G}(\mathfrak{A}, \psi)$ is represented in $\mathfrak{A}$ by the class of all $(m+k)$-tuples $(\bar{c}, \bar{a})$ such that $\bar{c}$ has equality type e_φ and $a_i = s(x_i)$ for all $x_i \in \mathrm{free}(\varphi)$. Thus, the domain and equality formulae of the interpretation I_ψ are

$$\delta(\bar{u}, \bar{x}) := \bigvee_{\varphi \in \mathrm{Sf}(\psi)} e_\varphi(\bar{u}),$$

$$\varepsilon(\bar{u}, \bar{x}; \bar{v}, \bar{y}) := \bigvee_{\varphi \in \mathrm{Sf}(\psi)} \Big(e_\varphi(\bar{u}) \wedge e_\varphi(\bar{v}) \wedge \bigwedge_{x_i \in \mathrm{free}(\varphi)} x_i = y_i \Big).$$

The relations V_0, V_1, E, T_0, T_1 of the game graph $\mathscr{G}(\mathfrak{A}, \psi)$ are clearly representable by quantifier-free formulae in $\mathfrak{A}$. Actually the formulae for V_0, V_1, E are pure equality formulae. Explicitly,

$$\psi_{V_\sigma}(\bar{u}, \bar{x}) := \bigvee_{\varphi \text{ belongs to Player } \sigma} e_\varphi(\bar{u}),$$

$$\psi_E(\bar{u}, \bar{x}; \bar{v}, \bar{y}) := \bigvee_{(\varphi, \vartheta) \in E_\psi} \Big(e_\varphi(\bar{u}) \wedge e_\vartheta(\bar{v}) \wedge \bigwedge_{x_i \in \mathrm{free}(\varphi) \cap \mathrm{free}(\vartheta)} x_i = y_i \Big).$$

Finally, the formulae defining the target sets T_0 and T_1 of the two players are

$$\psi_{T_0}(\bar{u}, \bar{x}) := \bigvee_{\varphi \text{ is a literal}} e_\varphi(\bar{u}) \wedge \varphi(\bar{x}),$$

$$\psi_{T_1}(\bar{u}, \bar{x}) := \bigvee_{\varphi \text{ is a literal}} e_\varphi(\bar{u}) \wedge \neg\varphi(\bar{x}).$$

This completes the definition of I_ψ. Clearly, for every structure $\mathfrak{A}$ with more than one element, we have the coordinate map $h : \delta^{\mathfrak{A}} \to \mathscr{G}(\mathfrak{A}, \psi)$ that maps every tuple $(\bar{c}, \bar{a}) \in \delta^{\mathfrak{A}}$ to the unique node (φ, s) such that $\bar{c}$ has equality type e_φ and $s(x_i) = a_i$ for all free variables x_i of φ. □

Obviously this construction is not limited to model-checking games for first-order logic. Indeed, with only trivial modifications it also works for many other logics, including the following cases.

(1) Least fixed point logic LFP and the associated parity games.
(2) The posGFP-fragment of LFP and the associated safety games.
(3) Logics with team semantics and the associated second-order reachability games.
(4) The safety games for inclusion logic (and other logics with universal-existential dependencies)

We briefly describe the modifications. For a parity game $\mathscr{G}(\mathfrak{A},\psi)$ for an LFP-formula ψ one has to take into account the additional edges from fixed-point formulae and fixed-point atoms to the formulae defining the fixed-point (as explained in Sect. 4.2), which changes the syntax tree to a syntax graph. In addition, the model checking game $\mathscr{G}(\mathfrak{A},\psi)$ has unary relations P_i associated to the priorities. These are defined by formulae $\psi_{P_i}(\overline{u},\overline{x})$ which are just the disjunctions over all equality types $e_\varphi(\overline{u})$ for the fixed-point atoms $\varphi = R\overline{x}$ which have priority i. For the safety games associated to posGFP-formulae the construction is similar, but instead of priorities we need a formula for the safety condition, of the form

$$\psi_S(\overline{u},\overline{x}) := \bigvee_{\varphi \text{ is a literal}} e_\varphi(\overline{u}) \to \varphi(\overline{x}),$$

saying that Player 0 has to avoid positions (φ,s) where φ is a literal with $\mathfrak{A} \models_s \neg\varphi$.

In what follows we shall need such game interpretations for the safety games associated with formulae of posGFP and for formulae of inclusion logic. The following proposition is an immediate consequence of the arguments given above and the constructions given in Sect. 3 and Sect. 4.5.

Proposition 19 *For every formula $\psi(\overline{x})$ of inclusion logic, there is a quantifier-free first-order interpretation $I(\psi)$ which, for every structure $\mathfrak{A}$, interprets the safety game $\mathscr{G}_{safe}(\mathfrak{A},\psi)$ in $\mathfrak{A}$.*

An analogous statement holds for the safety games associated with posGFP, based on the construction of Sect 4.3.

Game interpretations can be very useful to give high-level arguments for transformations of formulae among different logics (as we are going to show in the next section) and for establishing normal forms, without the need to go through a cumbersome structural induction over formulae. Let us illustrate this for the posGFP-fragment of least fixed-point logic.

Consider the formula $\mathrm{win}(x)$ that defines the winning region of Player 0 in safety games $\mathscr{G}$. For any formula ψ of posGFP, possibly with deeply nested gfp-operators, the interpretation $I = I(\psi)$ induces a translation of $\mathrm{win}(x)$ into a formula $\mathrm{win}^I(\overline{u},\overline{x})$ such that, for every structure $\mathfrak{A}$, every subformula $\varphi(\overline{y})$ of ψ, and every assignment $s : \mathrm{free}(\varphi) \to A$ we have that

$$\begin{aligned}
\mathfrak{A} \models_s \varphi(\overline{x}) &\Leftrightarrow \text{Player 0 wins } \mathscr{G}(\mathfrak{A},\psi) \text{ from position } (\varphi,s)\\
&\Leftrightarrow \mathscr{G}(\mathfrak{A},\psi) \models \mathrm{win}((\varphi,s))\\
&\Leftrightarrow \mathfrak{A} \models \mathrm{win}^I(\overline{c},\overline{a}) \text{ for one, and hence all, tuples } (\overline{c},\overline{a}) \in h^{-1}((\varphi,s))\\
&\Leftrightarrow \mathfrak{A} \models_s \exists\overline{u}\exists\overline{y}\Big(e_\varphi(\overline{u}) \wedge \bigwedge_{x_i \in \mathrm{free}(\varphi)} y_i = x_i \wedge \mathrm{win}^I(\overline{u},\overline{y})\Big).
\end{aligned}$$

Since the interpretation I is quantifier-free, the transformation $\mathrm{win}(x) \mapsto \mathrm{win}^I(\overline{u},\overline{x})$ does not change the structure very much; in particular it does not change the number of **gfp**-operators and the alternation depth of existential and universal quantifiers. Since the winning region of safety games is definable in posGFP by a formula

with just one application of a **gfp**-operator to a disjunction (or conjunction) of an existential and a universal formula, the same is true for the translated formula.

Proposition 20 *Every formula in* posGFP *is equivalent to a formula with a single application of a gfp-operator to a positive Boolean combination of purely existential and purely universal first-order formulae.*

It is known that, on finite structures, every LFP-formula is equivalent to a posGFP-formula. Thus, on finite structures, the normal form given by this proposition is in fact a normal form for LFP, a result first observed by Dahlhaus [3]. On arbitrary structures, however, this is not true since least fixed points are not expressible by greatest ones without negation, and the alternation hierarchy of least and greatest fixed points is strict.

7 Least fixed-point logic versus inclusion logic

We now are ready to explain in what sense inclusion logic and positive greatest fixed-point logic are 'equivalent'. For sentences, this is an unproblematic statement: for every sentence ψ of inclusion logic there is a sentence φ of posGFP, and vice versa, such that, for every structure $\mathfrak{A}$ we have that $\mathfrak{A} \models \psi$ if, and only if, $\mathfrak{A} \models \varphi$. An informal proof of this goes as follows. The model-checking games for a sentence ψ in any of these two logics are safety games, and these are interpretable by an interpretation $I(\psi)$ in the structure in which the sentence ψ is evaluated. Now take a formula of the other of the two logics, saying that Player 0 wins the given safety game (from the root), and transform it, by means of $I(\psi)$ back into a sentence on $\mathfrak{A}$, which is equivalent to ψ. (We shall give a precise argument for a more general statement below.)

For formulae with free variables, the situation is more complicated since formulae of logics with team semantics, such as inclusion logic, define different semantic objects than formulae of logics with Tarski semantics, such as posGFP. A formula $\varphi(\bar{x})$ (of vocabulary τ and arity k) with Tarski semantics defines a *query* Q_φ, a function that associates with every τ-structure $\mathfrak{A}$ the k-ary relation $\varphi^{\mathfrak{A}} := \{\bar{a} : \mathfrak{A} \models \varphi(\bar{a})\}$. Of course, the relation $\varphi^{\mathfrak{A}}$ can be identified with a team, namely the team

$$X_\varphi^{\mathfrak{A}} := \{s : \bar{x} \mapsto \bar{a} : \bar{a} \in \varphi^{\mathfrak{A}}\} = \{s : \mathfrak{A} \models_s \varphi\}.$$

For the general definition of a query, one has to require invariance under isomorphism.

Definition 21. A query (of vocabulary τ and arity k) is a function Q, that associates with every τ-structure a team $Q(\mathfrak{A})$ of assignments $s : \{x_1, \ldots, x_k\} \to A$ such that, for any isomorphism $h : \mathfrak{A} \to \mathfrak{B}$ between two τ-structures, we also have that $hQ(\mathfrak{A}) = Q(\mathfrak{B})$, which means that for any assignment $s : \{x_1, \ldots, x_k\} \to A$ we have that $s \in Q(\mathfrak{A})$ if, and only if, $h \circ s \in Q(\mathfrak{B})$.

On the other side, a formula $\psi(\bar{x})$ (again of vocabulary τ and arity k) with team semantics defines what we call a *team query*. It associates with every structure the set $[\![\psi]\!]^{\mathfrak{A}}$ of all teams X such that $\mathfrak{A} \models_X \varphi$. For a general definition, we again have to impose isomorphism invariance.

Definition 22. A *team query* (of vocabulary τ and arity k) is a function TQ that associates with every τ structure $\mathfrak{A}$ a set $\mathrm{TQ}(\mathfrak{A})$ of teams with domain $\{x_1,\dots,x_k\}$ and values in A, such that, for every isomorphism $h : \mathfrak{A} \to \mathfrak{B}$ between two τ-structures and every team X with values in A, we have that $X \in \mathrm{TQ}(\mathfrak{A})$ if, and only if $hX \in \mathrm{TQ}(\mathfrak{B})$. We say that a team query TQ is L-definable (for a logic L with team semantics) if there exists a formula $\psi \in L$ such that $[\![\psi]\!]^{\mathfrak{A}} = \mathrm{TQ}(\mathfrak{A})$ for all structures $\mathfrak{A}$.

Definability of team queries can also be considered in logics with Tarski semantics, by means of *sentences* of a vocabulary that is expanded by a relation representing the team.

Definition 23. A team query TQ of vocabulary τ is defined by a sentence ψ of vocabulary $\tau \cup \{X\}$ if, for every τ-structure $\mathfrak{A}$ we have that $\mathrm{TQ}(\mathfrak{A}) = \{X : (\mathfrak{A}, X) \models \psi\}$.

In this sense it has been shown in [12, 5] that

- the team queries definable in dependence logic are precisely those that are definable by an existential second-order sentence in which the predicate for the team occurs only negatively, and
- the team queries definable in independence logic (or inclusion-exclusion logic) are precisely those definable by existential second-order sentences.

Notice that, by the closure under unions of teams, there exists for every formula $\varphi(\bar{x})$ of inclusion logic and every structure $\mathfrak{A}$ a unique maximal team $X_{\max} := \max\{X : \mathfrak{A} \models_X \varphi\}$. It follows that every team query TQ that is definable in inclusion logic induces a query $\max \mathrm{TQ}$ associating with $\mathfrak{A}$ the maximal team $X \in \mathrm{TQ}(\mathfrak{A})$. For the same reason it follows that inclusion logic cannot be equivalent with greatest fixed point logic in the same sense in which independence logic is equivalent with existential second-order logic. Indeed sentences $\psi(X)$ of posGFP need not be closed under union (of relations for X), and there need not be a well-defined maximal relation X satisfying ψ.

The relationship between inclusion logic and positive greatest fixed-point logic can be made precise in several ways:

(1) The queries definable in posGFP are precisely those that occur as the maximum of a team query that is definable in inclusion logic.
(2) The team queries definable in inclusion logic are those definable by sentences in posGFP (of expanded vocabulary) of the form $\forall\bar{x}(X\bar{x} \to \psi(X,\bar{x}))$ where the team predicate X occurs only positively in ψ.
(3) The post-fixed points of posGFP-definable relational operators coincide with the team queries definable in inclusion logic.

Theorem 24 *For every formula $\psi(\bar{x})$ of inclusion logic there is a formula $\varphi(X,\bar{x})$ in* posGFP*, with only positive occurrences of X, such that, for every structure $\mathfrak{A}$ and every team X we have that*

$$\mathfrak{A} \models_X \psi(\bar{x}) \;\Leftrightarrow\; (\mathfrak{A},X) \models \forall\bar{x}(X\bar{x} \to \varphi(X,\bar{x})) \;\Leftrightarrow\; F^{\mathfrak{A}}_{\varphi}(X) \supseteq X.$$

In particular, for all assignments s, we have that $\mathfrak{A} \models_s [\mathbf{gfp}\, X\bar{x}.\, \varphi(X,\bar{x})](\bar{x})$ if, and only if, $s \in \max\{X : \mathfrak{A} \models_X \psi(\bar{x})\}$.

Proof. Let $I(\psi)$ be the quantifier-free first-order interpretation which, for every structure $\mathfrak{A}$ and every team X, interprets the safety game $\mathscr{G}_{\mathrm{safe}}(\mathfrak{A},\psi)$ in $\mathfrak{A}$, and let h be the associated coordinate map. Recall that $\mathfrak{A} \models_X \psi(\bar{x})$ if, and only if, $X^* = \{(\psi,s) : s \in X\}$ is an I-trap in $\mathscr{G}_{\mathrm{safe}}(\mathfrak{A},\psi)$.

Further, let $\mathrm{itrap}(X^*,z)$ be the posGFP-formula, from Example 1, defining I-traps in safety games, in the sense that $(\mathscr{G},X^*) \models \forall z(X^*z \to \mathrm{itrap}(X^*,z))$ if, and only if X^* is an I-trap in $\mathscr{G}$.

By the Interpretation Lemma we get a formula $\mathrm{itrap}^{I(\psi)}(Y,\bar{z})$, which is also in posGFP such that, for $Y = h^{-1}(X^*)$,

$$(\mathfrak{A},Y) \models \forall\bar{z}(Y\bar{z} \to \mathrm{itrap}^{I(\psi)}(Y,\bar{z})) \;\Leftrightarrow\; (\mathscr{G}_{\mathrm{safe}}(\mathfrak{A},\psi),X^*) \models \forall z(X^*z \to \mathrm{itrap}(X^*,z)).$$

For an assignment s with domain $\mathrm{free}(\psi) = \{x_1,\dots x_k\}$ the tuples $(\bar{c},\bar{a})$ in $\mathfrak{A}$ that interpret the position (ψ,s) are those satisfying $e_\psi(\bar{c}) \wedge \bigwedge_{i=1}^k a_i = s(x_i)$. (Notice that the length of $\bar{a}$ is in general, greater than the length of $\bar{x}$.) Thus we may write tuples $\bar{z}$ representing game positions as $\bar{z} = (\bar{u},\overline{xx'})$ and observe that $Y = h^{-1}(X^*) = \{(\bar{u},\overline{xx'}) : (\mathfrak{A},X) \models e_\psi(\bar{u}) \wedge X\bar{x}\}$. Let now $\mathrm{itrap}^*(X,\bar{u},\bar{x},\bar{x}')$ be the formula that is obtained from $\mathrm{itrap}^{I(\psi)}(Y,\overline{uxx'})$ by replacing all atoms $Y\overline{vyy'}$ by the formula $(e_\psi(\bar{v}) \wedge X\bar{y})$. Finally, we put

$$\varphi(X,\bar{x}) := \forall\bar{u}\forall\bar{x}'(e_\psi(\bar{u}) \to \mathrm{itrap}^*(X,\overline{uxx'})).$$

Notice that all occurrences of X in $\varphi(X,\bar{x})$ are positive. Putting everything together, we have that

$$\mathfrak{A} \models_X \psi(\bar{x}) \;\Leftrightarrow\; (\mathfrak{A},X) \models \forall\bar{x}(X\bar{x} \to \varphi(X,\bar{x})).$$

□

Corollary 25 *If* TQ *is a team query that is definable in inclusion logic, then the query* max TQ *is definable in* posGFP*. Moreover,* TQ *is definable by a sentence of* posGFP *(of an expanded vocabulary by a relation for the team).*

For the converse relationship we establish the following result.

Theorem 26 *For every formula $\varphi(X,\bar{x})$ in posGFP, with only positive occurrences of X, there is a formula $\psi(\bar{x})$ in inclusion logic such that, for every structure $\mathfrak{A}$ and every team X we have that*

$$(\mathfrak{A},X) \models \forall\bar{x}(X\bar{x} \to \varphi(X,\bar{x})) \Leftrightarrow\; \mathfrak{A} \models_X \psi(\bar{x}).$$

Proof. Let $\mathscr{G}^*(\mathfrak{A},\varphi)$ be the game from Proposition 8 such that $(\mathfrak{A},X) \models \forall\bar{x}(X\bar{x} \to \varphi(X,\bar{x}))$ if, and only if, $X^* = \{(\varphi,s) : s(\bar{x}) \in X\}$ is an I-trap in $\mathscr{G}^*(\mathfrak{A},\varphi)$. Further, let $J(\varphi)$ be the quantifier-free interpretation with coordinate map h which, for every structure $\mathfrak{A}$, interprets $\mathscr{G}^*(\mathfrak{A},\varphi)$ in $\mathfrak{A}$.

Let $\mathrm{itrap}(x)$ be the formula of inclusion logic, constructed in Sect. 4.5, such that for all games $\mathscr{G}$, $[\![\mathrm{itrap}(x)]\!]^{\mathscr{G}}$ is the set of teams that define an I-trap in $\mathscr{G}$. By the Interpretation Lemma we get a formula $\mathrm{itrap}^{J(\varphi)}(\bar{u},\bar{z})$, which is also in inclusion logic, such that $[\![\mathrm{itrap}^{J(\varphi)}(\bar{u},\bar{z})]\!]^{\mathfrak{A}}$ is the set of all teams $Z = h^{-1}(Y)$ where Y defines an I-trap of $\mathscr{G}^*(\mathfrak{A},\varphi)$. Notice that such a Z consists of all assignments $t' : (\bar{u},\bar{z}) \mapsto (\bar{c},\bar{a})$ such that $h(\bar{c},\bar{a}) = t(x)$ for some $t \in Y$.

For the specific team that defines X^* we can write $\bar{z} = \overline{yy'}$ and get, that $Z(X) = h^{-1}(X^*)$ is the set of all assignments $(\bar{u},\overline{xx'}) \mapsto (\bar{c},\overline{aa'})$ such that $e_\psi(\bar{c})$ and $\bar{a} \in X$. We now set

$$\psi(\bar{x}) := \forall\bar{u}\forall\bar{x}'(e_\psi(\bar{u}) \to \mathrm{itrap}^{J(\varphi)}(\bar{u},\overline{xx'}))$$

so that $\mathfrak{A} \models_X \psi(\bar{x})$ if,and only if, $\mathfrak{A} \models_{Z(X)} \mathrm{itrap}^{J(\varphi)}(\bar{u},\overline{xx'})$. Putting everything together, we have

$$\begin{aligned}(\mathfrak{A},X) \models \forall X(X\bar{x} \to \varphi(\bar{x})) &\Leftrightarrow X^* \text{ is an } I\text{-trap in } \mathscr{G}^*(\mathfrak{A},\varphi)\\ &\Leftrightarrow \mathscr{G}^*(\mathfrak{A},\varphi) \models_{X^*} \mathrm{itrap}(x)\\ &\Leftrightarrow \mathfrak{A} \models_{Z(X)} \mathrm{itrap}^{J(\varphi)}(\bar{u},\overline{xx'})\\ &\Leftrightarrow \mathfrak{A} \models_X \psi(\bar{x}).\end{aligned}$$

□

Corollary 27 *For every formula $\varphi(\bar{x})$ in posGFP there is a formula $\psi(\bar{x})$ in inclusion logic such that, for every structure $\mathfrak{A}$ and every team X we have that*

$$\mathfrak{A} \models_s \varphi(\bar{x}) \text{ for all } s \in X \;\Leftrightarrow\; \mathfrak{A} \models_X \psi(\bar{x}).$$

Corollary 28 *For every query Q that is definable in* posGFP*, the team query $\mathscr{P}(Q)$, which associates with every structure $\mathfrak{A}$ the power-set of the team $Q(\mathfrak{A})$, is definable in inclusion logic.*

Corollary 29 *The* posGFP*-definable queries Q are precisely those, for which there exists a team query* TQ *that is definable in inclusion logic with* $\max \mathrm{TQ} = Q$.

References

1. K. Apt and E. Grädel, editors. *Lectures in Game Theory for Computer Scientists*. Cambridge University Press, 2011.
2. A. Blass and Y. Gurevich. Henkin quantifiers and complete problems. *Annals of Pure and Applied Logic*, 32:1–16, 1986.

3. E. Dahlhaus. Skolem normal forms concerning the least fixed point. In E. Börger, editor, *Computation Theory and Logic*, number 270 in Lecture Notes in Computer Science, pages 101–106. Springer Verlag, 1987.
4. F. Engström. Generalized quantifiers in dependence logic. *Journal of Logic, Language, and Information*, 2012.
5. P. Galliani. Inclusion and exclusion in team semantics — on some logics of imperfect information. *Annals of Pure and Applied Logic*, 163:68–84, 2012.
6. P. Galliani and L. Hella. Inclusion logic and fixed-point logic. In *Computer Science Logic 2013*, volume 23 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 281–295, 2013.
7. E. Grädel. Model-checking games for logics of incomplete information. *Theoretical Computer Science*, 493:2–14, 2013.
8. E. Grädel and J.Väänänen. Dependence and independence. *Studia Logica*, 101(2):399–410, 2013.
9. E. Grädel et al. *Finite Model Theory and Its Applications*. Springer-Verlag, 2007.
10. W. Hodges. *A shorter model theory*. Cambridge University Press, 1997.
11. W. Hodges. Logics of imperfect information: Why sets of assignments? In Johan van Benthem, Benedikt Löwe, and Dov Gabbay, editors, *Interactive Logic*, volume 1 of *Texts in Logic and Games*, pages 117–134. Amsterdam University Press, 2007.
12. J. Kontinen and J. Väänänen. On definability in dependence logic. *Journal of Logic, Language, and Information*, 18:317–241, 2009.
13. A. Mann, G. Sandu, and M. Sevenster. *Independence-Friendly Logic. A Game-Theoretic Approach*, volume 386 of *London Mathematical Societay Lecture Notes Series*. Cambridge University Press, 2012.
14. J. Väänänen. *Dependence Logic*. Cambridge University Press, 2007.