Team Semantics, Games, and Negation

Erich Grädel

Academy Colloquium on Dependence Logic, 3rd March 2014

Outline

- A short introduction into logics with team semantics
- Classical model-checking games: reachability games
- Second-order reachability games
- Model-checking games for logics with team semantics
- Inclusion logic and greatest fixed-points: a game theoretic view
- Negation

Logics of dependence and independence

Prehistory:

Henkin, Enderton, Walkoe, …: partially ordered (or Henkin-) quantifiers

Blass and Gurevich: correspondence to Σ^1_1 (and thus NP)

Hintikka and Sandu: Independence-friendly (IF) logic with explicit dependencies of quantifiers on each other

Semantics in terms of games with imperfect information

Logics of dependence and independence

Prehistory:

Henkin, Enderton, Walkoe, …: partially ordered (or Henkin-) quantifiers

Blass and Gurevich: correspondence to Σ^1_1 (and thus NP)

Hintikka and Sandu: Independence-friendly (IF) logic with explicit dependencies of quantifiers on each other

Semantics in terms of games with imperfect information

Hodges: model-theoretic semantics for IF-logic

Difference to Tarski semantics: a formula is not evaluated against a single assignment but against a set of assignments, now called a team

This kind of semantics is an important achievement of independent interest.

Logics of dependence and independence

Modern framework: Rather than stating dependencies as annotations of quantifiers, treat dependence or independence as atomic statements

Logics of dependence and independence

Modern framework: Rather than stating dependencies as annotations of quantifiers, treat dependence or independence as atomic statements

These express atomic properties not of single assignments, but of sets of assignments, called teams.

Thus team semantics starts right at the atomic level.

Axiomatizations of atomic dependence properties is an interesting issue

Logics of dependence and independence

Modern framework: Rather than stating dependencies as annotations of quantifiers, treat dependence or independence as atomic statements

These express atomic properties not of single assignments, but of sets of assignments, called teams.

Thus team semantics starts right at the atomic level.

Axiomatizations of atomic dependence properties is an interesting issue

Combine these atoms stating dependencies and/or independencies with common logical operators, such as connectives and quantifiers, to obtain full-fledged logics for reasoning about dependence and independence.

Dependence atoms

Dependence atoms are expressions $=(\overline{x}, y)$.

Semantics: Let $\mathfrak{A}$ be a structure and X a team of assignments $s : V \to A$.

$\mathfrak{A} \models_X =(\overline{x}, y)$ if y depends on $\overline{x}$ in $\mathfrak{A}$ and X.

This means that for all $s, s' \in X$,

$$\bigwedge_{i=1}^{n} s(x_i) = s'(x_i) \implies s(y) = s'(y)$$

Dependence atoms

Dependence atoms are expressions $=(\overline{x}, y)$.

Semantics: Let $\mathfrak{A}$ be a structure and X a team of assignments $s : V \to A$.

$\mathfrak{A} \models_X =(\overline{x}, y)$ if y depends on $\overline{x}$ in $\mathfrak{A}$ and X.

This means that for all $s, s' \in X$,

$$\bigwedge_{i=1}^{n} s(x_i) = s'(x_i) \implies s(y) = s'(y)$$

Expanded form of dependence atoms $=(\overline{x}, \overline{y})$ saying that all variables of $\overline{y}$ depend on $\overline{x}$.

Independence atoms: simple case

Definition. A team X satisfies the atom $x \perp y$ if

$$(\forall s, s' \in X)(\exists s'' \in X)(s''(x) = s(x) \wedge s''(y) = s'(y))$$

Independence atoms: simple case

Definition. A team X satisfies the atom $x \perp y$ if

$$(\forall s, s' \in X)(\exists s'' \in X)(s''(x) = s(x) \wedge s''(y) = s'(y))$$

Suppose you know X and you know that s is some assignment in X. You want to gather information about $s(y)$. What you know is that $s(y) \in \{a : a = s'(y) \text{ for some } s' \in X\}$.

Independence atoms: simple case

Definition. A team X satisfies the atom $x \perp y$ if

$$(\forall s, s' \in X)(\exists s'' \in X)(s''(x) = s(x) \wedge s''(y) = s'(y))$$

Suppose you know X and you know that s is some assignment in X.
You want to gather information about $s(y)$. What you know is that
$s(y) \in \{a : a = s'(y) \text{ for some } s' \in X\}$.

Suppose that I tell you $s(x)$ where $x \perp y$.
You cannot infer anything new about $s(y)$. Indeed, for all potential values a for $s(y)$ there is an assignment $s'' \in X$ with $s''(x) = s(x)$ and $s''(y) = a$.

Independence atoms: general case

The independence atom $x \perp y$, and also its extension to independency atoms $\overline{x} \perp \overline{y}$ on tuples of variables, are special forms of a more general atom

$$\overline{x} \perp_{\overline{z}} \overline{y}$$

saying that the variables $\overline{x}$ are completely independent from $\overline{y}$ for any constant value of $\overline{z}$.

Definition. A team X satisfies the atom $\overline{x} \perp_{\overline{z}} \overline{y}$ if for any pair of assignments $s, s' \in X$ with $s(\overline{z}) = s'(\overline{z})$ there is a third assignment $s'' \in X$ with

- $s''(\overline{z}) = s(\overline{z}) = s'(\overline{z})$
- $s''(\overline{x}) = s(\overline{x})$
- $s''(\overline{y}) = s'(\overline{y})$.

Inclusion, exclusion, and all that

Besides dependence and independence, there are other interesting atomic properties of teams. One source of such properties is database dependency theory.

Inclusion, exclusion, and all that

Besides dependence and independence, there are other interesting atomic properties of teams. One source of such properties is database dependency theory.

Inclusion dependencies:

$$\vDash_X (\bar{x} \subseteq \bar{y}) \quad :\Longleftrightarrow \quad (\forall s \in X)(\exists s' \in X)(s(\bar{x}) = s'(\bar{y}))$$

Inclusion, exclusion, and all that

Besides dependence and independence, there are other interesting atomic properties of teams. One source of such properties is database dependency theory.

Inclusion dependencies:

$$\models_X (\overline{x} \subseteq \overline{y}) \quad :\Longleftrightarrow \quad (\forall s \in X)(\exists s' \in X)(s(\overline{x}) = s'(\overline{y}))$$

Exclusion dependencies:

$$\models_X (\overline{x} \mid \overline{y}) \quad :\Longleftrightarrow \quad (\forall s \in X)(\forall s' \in X)(s(\overline{x}) \neq s'(\overline{y}))$$

Inclusion, exclusion, and all that

Besides dependence and independence, there are other interesting atomic properties of teams. One source of such properties is database dependency theory.

Inclusion dependencies:

$$\models_X (\overline{x} \subseteq \overline{y}) \quad :\Longleftrightarrow \quad (\forall s \in X)(\exists s' \in X)(s(\overline{x}) = s'(\overline{y}))$$

Exclusion dependencies:

$$\models_X (\overline{x} \mid \overline{y}) \quad :\Longleftrightarrow \quad (\forall s \in X)(\forall s' \in X)(s(\overline{x}) \neq s'(\overline{y}))$$

Equiextension:

$$\models_X (\overline{x} \bowtie \overline{y}) \quad :\Longleftrightarrow \quad \{s(\overline{x}) : s \in X\} = \{s(\overline{y}) : s \in X\}$$

Logics of dependence and independence

Dependence logic: FO + dependence atoms $=(\overline{x}, y)$

Independence logic: FO + independence atoms $\overline{x} \perp_{\overline{z}} \overline{y}$

and so on.

Logics of dependence and independence

Dependence logic: FO + dependence atoms $=(\overline{x}, y)$

Independence logic: FO + independence atoms $\overline{x} \perp_{\overline{z}} \overline{y}$

and so on.

All these logics require team semantics.

What precisely does it mean that, $\mathfrak{A} \models_X \psi(\overline{x})$?

Team semantics for first-order logic

$$\mathfrak{A} \models_X \psi(\overline{y}) \quad \Longleftrightarrow \quad \mathfrak{A} \models_s \psi(\overline{y}) \text{ for all } s \in X$$

Team semantics for first-order logic

$$\mathfrak{A} \models_X \psi(\overline{y}) \quad \Longleftrightarrow \quad \mathfrak{A} \models_s \psi(\overline{y}) \text{ for all } s \in X$$

- $\mathfrak{A} \models_X \psi \wedge \varphi \quad :\Longleftrightarrow \quad$ for all $s \in X$, $\mathfrak{A} \models_s \psi$ and $\mathfrak{A} \models_s \varphi$
- $\mathfrak{A} \models_X \psi \vee \varphi \quad :\Longleftrightarrow \quad$ for all $s \in X$, $\mathfrak{A} \models_s \psi$ or $\mathfrak{A} \models_s \varphi$
- $\mathfrak{A} \models_X \exists y \psi \quad :\Longleftrightarrow \quad$ for all $s \in X$ there exist $a \in A$ with $\mathfrak{A} \models_{s[y \mapsto a]} \psi$
- $\mathfrak{A} \models_X \forall y \psi \quad :\Longleftrightarrow \quad$ for all $s \in X$ and all $a \in A$, $\mathfrak{A} \models_{s[y \mapsto a]} \psi$

Team semantics: inductive definition

- For $\psi(\overline{y}) \in \text{FO}: \quad \mathfrak{A} \models_X \psi(\overline{y}) \iff \mathfrak{A} \models_s \psi(\overline{y})$ for all $s \in X$
- $\mathfrak{A} \models_X \psi \wedge \varphi \quad :\iff \quad$ for all $s \in X$, $\mathfrak{A} \models_s \psi$ and $\mathfrak{A} \models_s \varphi$
- $\mathfrak{A} \models_X \psi \vee \varphi \quad :\iff \quad$ for all $s \in X$, $\mathfrak{A} \models_s \psi$ or $\mathfrak{A} \models_s \varphi$
- $\mathfrak{A} \models_X \exists y \psi \quad :\iff \quad$ for all $s \in X$ there exist $a \in A$ with $\mathfrak{A} \models_{s[y \mapsto a]} \psi$
- $\mathfrak{A} \models_X \forall y \psi \quad :\iff \quad$ for all $s \in X$ and all $a \in A$, $\mathfrak{A} \models_{s[y \mapsto a]} \psi$

Team semantics: inductive definition

- For $\psi(\overline{y}) \in \mathrm{FO}$: $\mathfrak{A} \models_X \psi(\overline{y}) \iff \mathfrak{A} \models_s \psi(\overline{y})$ for all $s \in X$
- $\mathfrak{A} \models_X \psi \land \varphi \quad :\iff \quad \mathfrak{A} \models_X \psi$ and $\mathfrak{A} \models_X \varphi$
- $\mathfrak{A} \models_X \psi \lor \varphi \quad :\iff \quad$ for all $s \in X$, $\mathfrak{A} \models_s \psi$ or $\mathfrak{A} \models_s \varphi$
- $\mathfrak{A} \models_X \exists y \psi \quad :\iff \quad$ for all $s \in X$ there exist $a \in A$ with $\mathfrak{A} \models_{s[y \mapsto a]} \psi$
- $\mathfrak{A} \models_X \forall y \psi \quad :\iff \quad$ for all $s \in X$ and all $a \in A$, $\mathfrak{A} \models_{s[y \mapsto a]} \psi$

Team semantics: inductive definition

- For $\psi(\overline{y}) \in \text{FO}$: $\mathfrak{A} \models_X \psi(\overline{y}) \iff \mathfrak{A} \models_s \psi(\overline{y})$ for all $s \in X$
- $\mathfrak{A} \models_X \psi \wedge \varphi \quad :\iff \quad \mathfrak{A} \models_X \psi$ and $\mathfrak{A} \models_X \varphi$
- $\mathfrak{A} \models_X \psi \vee \varphi \quad :\iff \quad X = Y \cup Z$ such that $\mathfrak{A} \models_Y \psi$ and $\mathfrak{A} \models_Z \varphi$
- $\mathfrak{A} \models_X \exists y\psi \quad :\iff \quad$ for all $s \in X$ there exist $a \in A$ with $\mathfrak{A} \models_{s[y\mapsto a]} \psi$
- $\mathfrak{A} \models_X \forall y\psi \quad :\iff \quad$ for all $s \in X$ and all $a \in A$, $\mathfrak{A} \models_{s[y\mapsto a]} \psi$

Team semantics: inductive definition

- For $\psi(\overline{y}) \in \mathrm{FO}$: $\quad \mathfrak{A} \models_X \psi(\overline{y}) \quad \Longleftrightarrow \quad \mathfrak{A} \models_s \psi(\overline{y})$ for all $s \in X$
- $\mathfrak{A} \models_X \psi \wedge \varphi \quad :\Longleftrightarrow \quad \mathfrak{A} \models_X \psi$ and $\mathfrak{A} \models_X \varphi$
- $\mathfrak{A} \models_X \psi \vee \varphi \quad :\Longleftrightarrow \quad X = Y \cup Z$ such that $\mathfrak{A} \models_Y \psi$ and $\mathfrak{A} \models_Z \varphi$
- $\mathfrak{A} \models_X \exists y \psi \quad :\Longleftrightarrow \quad$ for all $s \in X$ there exist $a \in A$ with $\mathfrak{A} \models_{s[y \mapsto a]} \psi$
- $\mathfrak{A} \models_X \forall y \psi \quad :\Longleftrightarrow \quad$ for all $s \in X$ and all $a \in A$, $\mathfrak{A} \models_{s[y \mapsto a]} \psi$

Notice that $(\varphi \vee \varphi)$ is, in general, not equivalent to φ

Team semantics: inductive definition

- For $\psi(\overline{y}) \in \mathrm{FO}$: $\mathfrak{A} \models_X \psi(\overline{y}) \iff \mathfrak{A} \models_s \psi(\overline{y})$ for all $s \in X$
- $\mathfrak{A} \models_X \psi \wedge \varphi \quad :\Longleftrightarrow \quad \mathfrak{A} \models_X \psi$ and $\mathfrak{A} \models_X \varphi$
- $\mathfrak{A} \models_X \psi \vee \varphi \quad :\Longleftrightarrow \quad X = Y \cup Z$ such that $\mathfrak{A} \models_Y \psi$ and $\mathfrak{A} \models_Z \varphi$
- $\mathfrak{A} \models_X \exists y \psi \quad :\Longleftrightarrow \quad$ for all $s \in X$ there exist $a \in A$ with $\mathfrak{A} \models_{s[y \mapsto a]} \psi$
- $\mathfrak{A} \models_X \forall y \psi \quad :\Longleftrightarrow \quad$ for all $s \in X$ and all $a \in A$, $\mathfrak{A} \models_{s[y \mapsto a]} \psi$

Notice that $(\varphi \vee \varphi)$ is, in general, not equivalent to φ

An example from dependence logic:

$=(y)$ means that the value of y is constant in the given team

$=(y) \vee =(y)$ means that y takes at most two values in the given team

Team semantics: inductive definition

- For $\psi(\overline{y}) \in \mathrm{FO}$: $\quad \mathfrak{A} \models_X \psi(\overline{y}) \iff \mathfrak{A} \models_s \psi(\overline{y})$ for all $s \in X$
- $\mathfrak{A} \models_X \psi \land \varphi \quad :\iff \quad \mathfrak{A} \models_X \psi$ and $\mathfrak{A} \models_X \varphi$
- $\mathfrak{A} \models_X \psi \lor \varphi \quad :\iff \quad X = Y \cup Z$ such that $\mathfrak{A} \models_Y \psi$ and $\mathfrak{A} \models_Z \varphi$
- $\mathfrak{A} \models_X \exists y \psi \quad :\iff \quad \exists F : X \to \mathcal{P}(A) \setminus \{\emptyset\}$ such that $\mathfrak{A} \models_{X[y \mapsto F]} \psi$
- $\mathfrak{A} \models_X \forall y \psi \quad :\iff \quad$ for all $s \in X$ and all $a \in A$, $\mathfrak{A} \models_{s[y \mapsto a]} \psi$

Team semantics: inductive definition

- For $\psi(\overline{y}) \in \mathrm{FO}$: $\quad \mathfrak{A} \models_X \psi(\overline{y}) \quad \Longleftrightarrow \quad \mathfrak{A} \models_s \psi(\overline{y})$ for all $s \in X$
- $\mathfrak{A} \models_X \psi \wedge \varphi \quad :\Longleftrightarrow \quad \mathfrak{A} \models_X \psi$ and $\mathfrak{A} \models_X \varphi$
- $\mathfrak{A} \models_X \psi \vee \varphi \quad :\Longleftrightarrow \quad X = Y \cup Z$ such that $\mathfrak{A} \models_Y \psi$ and $\mathfrak{A} \models_Z \varphi$
- $\mathfrak{A} \models_X \exists y \psi \quad :\Longleftrightarrow \quad \exists F : X \to \mathcal{P}(A) \setminus \{\varnothing\}$ such that $\mathfrak{A} \models_{X[y \mapsto F]} \psi$
- $\mathfrak{A} \models_X \forall y \psi \quad :\Longleftrightarrow \quad$ for all $s \in X$ and all $a \in A$, $\mathfrak{A} \models_{s[y \mapsto a]} \psi$

Choose (for every $s \in X$) an arbitrary non-empty set of witnesses for $\exists x \dots$
rather than just a single witness: lax semantics as opposed to strict semantics.

For FO and dependence logic the difference is immaterial
For stronger logics, only lax semantics guarantees the locality principle:

$$\mathfrak{A} \models_X \varphi \quad \Longleftrightarrow \quad \mathfrak{A} \models_{X \upharpoonright \mathrm{free}(\varphi)} \varphi$$

Team semantics: inductive definition

- For $\psi(\overline{y}) \in \mathrm{FO}:\quad \mathfrak{A} \models_X \psi(\overline{y}) \iff \mathfrak{A} \models_s \psi(\overline{y})$ for all $s \in X$
- $\mathfrak{A} \models_X \psi \wedge \varphi \quad :\iff \quad \mathfrak{A} \models_X \psi$ and $\mathfrak{A} \models_X \varphi$
- $\mathfrak{A} \models_X \psi \vee \varphi \quad :\iff \quad X = Y \cup Z$ such that $\mathfrak{A} \models_Y \psi$ and $\mathfrak{A} \models_Z \varphi$
- $\mathfrak{A} \models_X \exists y \psi \quad :\iff \quad \exists F : X \to \mathcal{P}(A) \setminus \{\varnothing\}$ such that $\mathfrak{A} \models_{X[y \mapsto F]} \psi$
- $\mathfrak{A} \models_X \forall y \psi \quad :\iff \quad \mathfrak{A} \models_{X[y \mapsto A]} \psi$

Team semantics: inductive definition

- For $\psi(\overline{y}) \in \mathrm{FO}:\quad \mathfrak{A} \models_X \psi(\overline{y}) \quad\Longleftrightarrow\quad \mathfrak{A} \models_s \psi(\overline{y})$ for all $s \in X$
- $\mathfrak{A} \models_X \psi \wedge \varphi \quad :\Longleftrightarrow \quad \mathfrak{A} \models_X \psi$ and $\mathfrak{A} \models_X \varphi$
- $\mathfrak{A} \models_X \psi \vee \varphi \quad :\Longleftrightarrow \quad X = Y \cup Z$ such that $\mathfrak{A} \models_Y \psi$ and $\mathfrak{A} \models_Z \varphi$
- $\mathfrak{A} \models_X \exists y \psi \quad :\Longleftrightarrow \quad \exists F : X \to \mathcal{P}(A) \setminus \{\varnothing\}$ such that $\mathfrak{A} \models_{X[y \mapsto F]} \psi$
- $\mathfrak{A} \models_X \forall y \psi \quad :\Longleftrightarrow \quad \mathfrak{A} \models_{X[y \mapsto A]} \psi$

For sentences we define: $\mathfrak{A} \models \psi \ :\Longleftrightarrow \mathfrak{A} \models_{\{\varnothing\}} \psi$

Notice that we cannot reasonably replace $\{\varnothing\}$ by $\varnothing$ since the empty team satisfies all formulae: $\mathfrak{A} \models_\varnothing \psi$ for all ψ

Example: Defining 3-SAT in dependence logic

Represent an instance $\varphi = \bigwedge_{i=1}^{m}(X_{i_1} \vee X_{i_2} \vee X_{i_3})$ of 3-SAT by a team

$Z_\varphi = \{(i, j, X, \sigma)$: in clause i at position j, the variable X appears with parity $\sigma\}$

Example: The formula $\varphi = (X_1 \vee \neg X_2 \vee X_3) \wedge (X_2 \vee X_4 \vee \neg X_5)$, is described by the team

clause	position	variable	parity
1	1	X_1	+
1	2	X_2	-
1	3	X_3	+
2	1	X_2	+
2	2	X_4	+
2	3	X_5	-

Example: Defining 3-SAT in dependence logic

Represent an instance $\varphi = \bigwedge_{i=1}^{m}(X_{i_1} \vee X_{i_2} \vee X_{i_3})$ of 3-SAT by a team

$Z_\varphi = \{(i, j, X, \sigma)$: in clause i at position j, the variable X appears with parity $\sigma\}$

Example: The formula $\varphi = (X_1 \vee \neg X_2 \vee X_3) \wedge (X_2 \vee X_4 \vee \neg X_5)$, is described by the team

clause	position	variable	parity
1	1	X_1	+
1	2	X_2	-
1	3	X_3	+
2	1	X_2	+
2	2	X_4	+
2	3	X_5	-

Proposition. φ is satisfiable if, and only if, the team Z_φ is a model of

$$=(\text{clause},\text{position}) \vee =(\text{clause}, \text{position}) \vee =(\text{variable},\text{parity})$$

Model-Checking Games

The model checking problem for a logic L (with classical Tarski-semantics)

Given: structure $\mathfrak{A}$

formula $\psi(\overline{x}) \in L$

assignment $s : \text{free}(\psi) \to A$

Question: $\mathfrak{A} \models_s \psi$?

Model-Checking Games

The model checking problem for a logic L (with classical Tarski-semantics)

Given:	structure $\mathfrak{A}$
	formula $\psi(\overline{x}) \in L$
	assignment $s : \text{free}(\psi) \to A$
Question:	$\mathfrak{A} \models_s \psi$?

Reduce model checking problem $\mathfrak{A} \models_s \psi$ to strategy problem for model checking game $G(\mathfrak{A}, \psi, s)$, played by

- Falsifier (also called Player 1), and
- Verifier (also called Player 0), such that

$$\mathfrak{A} \models_s \psi \iff \text{Verifier has winning strategy for } G(\mathfrak{A}, \psi, s)$$

Model-Checking Games

The model checking problem for a logic L (with classical Tarski-semantics)

Given: structure $\mathfrak{A}$

formula $\psi(\overline{x}) \in L$

assignment $s : \text{free}(\psi) \to A$

Question: $\mathfrak{A} \models_s \psi$?

Reduce model checking problem $\mathfrak{A} \models_s \psi$ to strategy problem for model checking game $G(\mathfrak{A}, \psi, s)$, played by

- Falsifier (also called Player 1), and
- Verifier (also called Player 0), such that

$$\mathfrak{A} \models_s \psi \iff \text{Verifier has winning strategy for } G(\mathfrak{A}, \psi, s)$$

$\Longrightarrow$ Model checking via construction of winning strategies

Model-checking game for first-order logic

The game $\mathcal{G}(\mathfrak{A}, \psi$ for a structure $\mathfrak{A}$ and $\psi(\overline{x}) \in$ FO.

Positions: (φ, s) φ is a subformula of ψ and $s : \text{free}(\varphi) \to A$

Verifier moves:

$$(\varphi_1 \vee \varphi_2, s) \to (\varphi_i, s \upharpoonright \text{free}(\varphi_i)) \qquad (i = 1, 2)$$
$$(\exists x \varphi, s) \to (\varphi, s[x \mapsto a]) \qquad (a \in A)$$

Falsifier moves

$$(\varphi_1 \wedge \varphi_2, s) \to (\varphi_i, s \upharpoonright \text{free}(\varphi_i)) \qquad (i = 1, 2)$$
$$(\forall x \varphi, s) \to (\varphi, s[x \mapsto a]) \qquad (a \in A)$$

Terminal positions: φ atomic / negated atomic

$$\begin{matrix}\text{Verifier} \\ \text{Falsifier}\end{matrix} \text{ wins at } (\varphi, s) \iff \mathfrak{A} \begin{matrix}\models_s \\ \not\models_s\end{matrix} \varphi$$

Reachability games

Two-player games with perfect information given by a game graph

$\mathcal{G} = (V, E), \quad V = V_0 \cup V_1 \cup T, \quad E \subseteq V \times V$

- Player 0 moves from positions $v \in V_0$,
  Player 1 moves from $v \in V_1$,
  T is the set of terminal nodes.
- Moves are along edges.
  Hence plays are finite or infinite paths through the graph
- Winning condition $\text{Win} \subseteq T$: at a terminal position $v \in T$,
  Player 0 has won if $v \in \text{Win}$ and Player 1 has won if $v \in T \setminus \text{Win}$.
  Infinite plays are draws.

Reachability games

Two-player games with perfect information given by a game graph
$\mathcal{G} = (V, E), \quad V = V_0 \cup V_1 \cup T, \quad E \subseteq V \times V$

- Player 0 moves from positions $v \in V_0$,
  Player 1 moves from $v \in V_1$,
  T is the set of terminal nodes.
- Moves are along edges.
  Hence plays are finite or infinite paths through the graph
- Winning condition $\text{Win} \subseteq T$: at a terminal position $v \in T$,
  Player 0 has won if $v \in \text{Win}$ and Player 1 has won if $v \in T \setminus \text{Win}$.
  Infinite plays are draws.

Let us assume that game graphs are acyclic and of finite depth.
Hence all plays are finite.

Winning regions and winning strategies

Winning regions

$$W_\sigma := \{v \in V : \text{Player } \sigma \text{ has a winning strategy from position } v\}$$

A (nondeterministic) winning strategy of Player 0 for $\mathcal{G}$ and Win with winning region W is a subgraph $S = (W, F) \subseteq (V, E)$ such that

(1) if $v \in V_0 \cap W$ then $vF \neq \emptyset$,

(2) if $v \in V_1 \cap W$ then $vF = vE$

(3) $W \cap T \subseteq \text{Win}$.

Winning regions and winning strategies

Winning regions

$$W_\sigma := \{v \in V : \text{Player } \sigma \text{ has a winning strategy from position } v\}$$

A (nondeterministic) winning strategy of Player 0 for $\mathcal{G}$ and Win with winning region W is a subgraph $S = (W, F) \subseteq (V, E)$ such that

(1) if $v \in V_0 \cap W$ then $vF \neq \emptyset$,

(2) if $v \in V_1 \cap W$ then $vF = vE$

(3) $W \cap T \subseteq \text{Win}$.

All plays that start at some $v \in W$ and are consistent with S reach a winning terminal position $w \in \text{Win}$.

Complexity of reachability games

Given a reachability game game $(\mathcal{G}, \text{Win})$ on a finite game graph, we can compute in linear time $O(|V| + |E|)$

- winning regions W_0, W_1
- winning strategies for both players on W_0 and W_1

Associated decision problem:

$$\text{GAME} := \{(\mathcal{G}, v) : \text{Player 0 has winning strategy for } \mathcal{G} \text{ from position } v\}$$

Theorem. GAME is PTIME-complete.

Reachability games for logics with team semantics

Reduce model checking problem $\mathfrak{A} \models_X \psi$ to a reachability game $G(\mathfrak{A}, X, \psi)$.

Positions: (φ, Y) Y is now a team of assignments $s : \text{free}(\varphi) \to A$

Reachability games for logics with team semantics

Reduce model checking problem $\mathfrak{A} \models_X \psi$ to a reachability game $G(\mathfrak{A}, X, \psi)$.

Positions: (φ, Y) Y is now a team of assignments $s : \text{free}(\varphi) \to A$

Falsifier moves:

$(\varphi_1 \land \varphi_2, Y) \to (\varphi_i, Y \upharpoonright \text{free}(\varphi_i))$ $(i = 1, 2)$

$(\forall x\varphi, Y) \to (\varphi, Y[x \mapsto A])$

Verifier moves:

$(\exists x\varphi, Y) \to (\varphi, Y[x \mapsto F])$ for some $F : Y \to (\mathcal{P}(A) \setminus \{\varnothing\})$

Protocol for disjunctions:

At $(\varphi_1 \lor \varphi_2, Y)$ Verifier selects Y_1, Y_2 such that $Y_1 \cup Y_2 = Y$.
Falsifier selects $i \in \{1, 2\}$ and moves to $(\varphi_i, Y_i \upharpoonright \text{free}(\varphi_i))$

Terminal positions: φ atomic / negated atomic

Do reachability games provide a satisfactory game-theoretic analysis for logics with team semantics?

The constructed reachability game really is the model-checking game for the translated formula into existential second-order logic.

Do reachability games provide a satisfactory game-theoretic analysis for logics with team semantics?

The constructed reachability game really is the model-checking game for the translated formula into existential second-order logic.

The size of this model is exponentially larger than the first-order game.

Do reachability games provide a satisfactory game-theoretic analysis for logics with team semantics?

The constructed reachability game really is the model-checking game for the translated formula into existential second-order logic.

The size of this model is exponentially larger than the first-order game.

It is not clear how to extract complexity results for model-checking problems out of these games

Do reachability games provide a satisfactory game-theoretic analysis for logics with team semantics?

The constructed reachability game really is the model-checking game for the translated formula into existential second-order logic.

The size of this model is exponentially larger than the first-order game.

It is not clear how to extract complexity results for model-checking problems out of these games

The game for the negated formula is not obtained by a simple dualization operation or a role switch of players

Second-order reachability games

Game graph: $\mathcal{G} = (V, V_0, V_1, T, I, E)$

I: set of initial positions

T: set of terminal positions

Winning condition for Player 0: $\text{Win} \subseteq \mathcal{P}(T)$.

For algorithmic purposes, we assume that Win is given by a compact description and that it can be decided in Ptime whether a given set $U \subseteq T$ belongs to Win.

Second-order reachability games

Game graph: $\mathcal{G} = (V, V_0, V_1, T, I, E)$

I: set of initial positions

T: set of terminal positions

Winning condition for Player 0: Win $\subseteq \mathcal{P}(T)$.

For algorithmic purposes, we assume that Win is given by a compact description and that it can be decided in PTIME whether a given set $U \subseteq T$ belongs to Win.

Does Player 0 have a (nondeterministic) strategy such that the set of terminal positions that are reachable by a play from I that is consistent with the strategy belongs to Win ?

Consistent winning strategies

Game graph: $\mathcal{G} = (V, V_0, V_1, T, I, E)$

Winning condition for Player 0: $\text{Win} \subseteq \mathcal{P}(T)$.

A consistent winning strategy of Player 0 for $\mathcal{G}$ and Win is a pair $S = (W, F) \subseteq (V, E)$ with $F \subseteq (W \times W) \cap E$ such that

(1) W is the set of nodes that are reachable from I via edges in F

(2) if $v \in V_0 \cap W$ then $vF \neq \emptyset$

(3) if $v \in V_1 \cap W$ then $vF = vE$

(4) $W \cap T \in \text{Win}$.

Notice that (1) implies $I \subseteq W$.

Complexity

Theorem. The problem whether a given game graph $\mathcal{G}$ with a compact decription for Win admits a consistent winning strategy for Player 0, is NP-complete.

Complexity

Theorem. The problem whether a given game graph $\mathcal{G}$ with a compact decription for Win admits a consistent winning strategy for Player 0, is NP-complete.

The problem is obviously in NP: guess a subgraph, and verify that it is a consistent winning strategy.

Complexity

Theorem. The problem whether a given game graph $\mathcal{G}$ with a compact decription for Win admits a consistent winning strategy for Player 0, is NP-complete.

The problem is obviously in NP: guess a subgraph, and verify that it is a consistent winning strategy.

NP-hardness by reduction from SAT. Given a CNF-formula ψ consider the obvious game $G(\psi)$, where Player 1 can move from the initial position ψ to clauses, and Player 0 from clauses to their literals, with

$$T := \{(C, Y) : C \text{ is a clause of } \psi, Y \in C\}$$
$$\text{Win} := \{U \subseteq T : U \text{ contains no conflicting pair } (C, Y), (C', \overline{Y})\}$$

Player 0 has a consistent winning strategy for $G(\psi) \iff \psi$ is satisfiable

Model-checking game for logics with team semantics

Consider FO together with a collection of atomic properties of teams.

Appropriate model checking games are obtained as follows:

- Take precisely the same model-checking game as for FO with Tarski-semantics but insist that distinct occurrences of the same subformula are represented by distinct nodes.
- Impose consistency conditions on the admissible strategies .

Model-checking game for logics with team semantics

Consider FO together with a collection of atomic properties of teams.

Appropriate model checking games are obtained as follows:

- Take precisely the same model-checking game as for FO with Tarski-semantics but insist that distinct occurrences of the same subformula are represented by distinct nodes.
- Impose consistency conditions on the admissible strategies .

The resulting games $\mathcal{G}(\mathfrak{A}, \psi)$, where Verifier may use only consistent strategies, can be viewed as games of imperfect information.

Second-order reachability games for model checking

Let (V, V_0, V_1, T, E) be the game graph of a model-checking game $\mathcal{G}(\mathfrak{A}, \psi)$.

Recall that $V = \{(\varphi, s) : \varphi \text{ is a subformula of } \psi,\ s : \text{free}(\varphi) \to A\}$

Second-order reachability games for model checking

Let (V, V_0, V_1, T, E) be the game graph of a model-checking game $\mathcal{G}(\mathfrak{A}, \psi)$.

Recall that $V = \{(\varphi, s) : \varphi \text{ is a subformula of } \psi,\ s : \text{free}(\varphi) \to A\}$

Teams defined by strategies: For any subset $W \subseteq V$ and any subformula φ of ψ

$$\text{Team}(W, \varphi) := \{s : (\varphi, s) \in W\}$$

For a strategy $S = (W, F)$, let $\text{Team}(S, \varphi) := \text{Team}(W, \varphi)$

Second-order reachability games for model checking

Let (V, V_0, V_1, T, E) be the game graph of a model-checking game $\mathcal{G}(\mathfrak{A}, \psi)$.

Recall that $V = \{(\varphi, s) : \varphi \text{ is a subformula of } \psi,\ s : \text{free}(\varphi) \to A\}$

Teams defined by strategies: For any subset $W \subseteq V$ and any subformula φ of ψ

$$\text{Team}(W, \varphi) := \{s : (\varphi, s) \in W\}$$

For a strategy $S = (W, F)$, let $\text{Team}(S, \varphi) := \text{Team}(W, \varphi)$

Second-order reachability condition $\text{Win} \subseteq \mathcal{P}(T)$:

$T = \{(\varphi, s) \in V : \varphi \text{ is an atomic or negated atomic subformula of } \psi\}$

$\text{Win} := \{U \subseteq T : \mathfrak{A} \vDash_{\text{Team}(U,\varphi)} \varphi \text{ for every atomic/negated atomic } \varphi\}$.

Second-order reachability games for model checking

For a model-checking game $\mathcal{G}(\mathfrak{A}, \psi) = (V, V_0, V_1, T, E)$ with the second-order reachability condition Win we thus have:

A consistent winning strategy for Player 0 (Verifier) with winning region W is a pair $S = (W, F) \subseteq (V, E)$ with $F \subseteq (W \times W) \cap E$ such that

(1) if $v \in V_0 \cap W$, then $vF \neq \emptyset$

(2) if $v \in V_1 \cap W$ then $vF = vE$

(3) for every atomic or negated atomic formula φ, $\mathfrak{A} \models_{\text{Team}(S,\varphi)} \varphi$ where $\text{Team}(S, \varphi) = \{s : (\varphi, s) \in W\}$

Notice that condition (3) refers to the entire set of terminal positions that are reachable by plays that are consistent with the strategy.

Correctness of the model checking games

The consistency condition for winning strategies translates from the atomic formulae to all positions of the game.

If $S = (W, F)$ is a consistent winning strategy for $\mathcal{G}(\mathfrak{A}, \psi)$ then, for all subformulae φ of ψ we have that $\mathfrak{A} \models_{\text{Team}(S,\varphi)} \varphi$.

Correctness of the model checking games

The consistency condition for winning strategies translates from the atomic formulae to all positions of the game.

If $S = (W, F)$ is a consistent winning strategy for $\mathcal{G}(\mathfrak{A}, \psi)$ then, for all subformulae φ of ψ we have that $\mathfrak{A} \models_{\text{Team}(S,\varphi)} \varphi$.

Theorem.

$\mathfrak{A} \models_X \psi \iff$ Verifier has a consistent winning strategy S for $\mathcal{G}(\mathfrak{A}, \psi)$ with $\text{Team}(S, \psi) = X$.

The other player

A consistent winning strategy for Falsifier is defined dually, with
$\text{Win}' := \{U \subseteq T : \mathfrak{A} \models_{\text{Team}(U,\varphi)} \neg\varphi$ for every atomic/negated atomic $\varphi\}$.

Notice that Win' need not be the complement of Win.

Theorem.
$\mathfrak{A} \models_Y \psi^{\neg} \iff$ Falsifier has a consistent winning strategy S' for $\mathcal{G}(\mathfrak{A}, \psi)$ with $\text{Team}(S', \psi) = Y$.

Here $\psi^{\neg}$ is the formula in negation normal form, corresponding to the negation of ψ.

Notice that logics with team semantics do not have the tertium non datur.

Complexity

Recall that the problem whether a given game graph $\mathcal{G}$ with a compact description for Win admits a consistent winning strategy for Player 0, is NP-complete.

Complexity

Recall that the problem whether a given game graph $\mathcal{G}$ with a compact description for Win admits a consistent winning strategy for Player 0, is NP-complete.

The size of a model checking game $\mathcal{G}(\mathfrak{A}, \psi)$ on a finite structure $\mathfrak{A}$ is bounded by $|\psi| \cdot |A|^{\text{width}(\psi)}$.

Theorem. Let L be any extension of first-order logic by atomic formulae on teams that can be evaluated in polynomial time. Then the model-checking problem for L on finite structures is in NEXPTIME. For formulae of bounded width, the model-checking problem is in NP.

Complexity

Theorem. The problem to decide, given a finite structure $\mathfrak{A}$, a team X and a formula ψ in dependence logic, whether $\mathfrak{A} \models_X \psi$, is NEXPTIME-complete. This also holds when $\mathfrak{A}$ and X are fixed, in fact even in the case where $\mathfrak{A}$ is just the set $\{0,1\}$ and $X = \{\varnothing\}$.

Complexity

Theorem. The problem to decide, given a finite structure $\mathfrak{A}$, a team X and a formula ψ in dependence logic, whether $\mathfrak{A} \models_X \psi$, is NEXPTIME-complete. This also holds when $\mathfrak{A}$ and X are fixed, in fact even in the case where $\mathfrak{A}$ is just the set $\{0,1\}$ and $X = \{\varnothing\}$.

The same complexity results hold for independence logic, and logics using inclusion, exclusion, and/or equiextension atoms.

Constancy logic. Fragment of dependence logic, using only dependence atoms of form $=(y)$.

The model checking problem for constancy logic is PSPACE-complete.

Example: 3-colourability

$X(G) = \{s : (\text{edge,node}) \mapsto (e, u) : e \text{ is an edge of } G \text{ and } u \in e\}$

$$\psi(\text{edge,node}) := \exists\text{colour}\Big(\big(=\!(\text{colour}) \vee =\!(\text{colour}) \vee =\!(\text{colour})\big)\wedge$$
$$=\!(\text{node,colour}) \wedge =\!(\text{edge,colour,node})\Big).$$

Claim. G is 3-colourable $\iff$ $V \cup E \models_{X(G)} \psi(\text{edge,node})$

Example: 3-colourability

$X(G) = \{s : (\text{edge,node}) \mapsto (e, u) : e \text{ is an edge of } G \text{ and } u \in e\}$

$$\psi(\text{edge,node}) := \exists\text{colour}\Big(\big(=(\text{colour}) \lor =(\text{colour}) \lor =(\text{colour})\big)\land$$
$$=(\text{node,colour}) \land =(\text{edge,colour,node})\Big).$$

Claim. G is 3-colourable $\Longleftrightarrow$ $V \cup E \models_{X(G)} \psi(\text{edge,node})$

A consistent winning strategy $S = (W, F)$ with $\text{Team}(S, \psi) = X(G)$ selects, for every assignment $s : (\text{edge,node}) \mapsto (e, u)$ at least one colour $c(s)$.

First conjunct: at most three colours are used

Second conjunct: the colour of (e, u) only depends on the node u

Final conjunct: the edge and the colour determine the node, i.e. a different colour is assigned to (e, u) and (e, v) for every edge $e = \{u, v\}$.

Example: Infinite descending chains

$$(A, <) \models \exists x \exists y (y < x \wedge y \subseteq x) \iff (A, <) \text{ is not well-founded}$$

Example: Infinite descending chains

$$(A, <) \models \exists x \exists y (y < x \wedge y \subseteq x) \quad \Longleftrightarrow \quad (A, <) \text{ is not well-founded}$$

A consistent winning strategy $S = (W, F)$ for this sentence just amounts to a selection of a team X of assignments $s : (x, y) \mapsto (a, b)$.

Example: Infinite descending chains

$$(A,<) \models \exists x \exists y (y < x \wedge y \subseteq x) \quad \Longleftrightarrow \quad (A,<) \text{ is not well-founded}$$

A consistent winning strategy $S = (W, F)$ for this sentence just amounts to a selection of a team X of assignments $s : (x, y) \mapsto (a, b)$.

All atoms are reachable by the opponent. Hence $\text{Team}(S, y < x) = \text{Team}(S, y \subseteq x) = X$. Thus, consistency of S means:

Example: Infinite descending chains

$$(A, <) \models \exists x \exists y (y < x \land y \subseteq x) \quad \Longleftrightarrow \quad (A, <) \text{ is not well-founded}$$

A consistent winning strategy $S = (W, F)$ for this sentence just amounts to a selection of a team X of assignments $s : (x, y) \mapsto (a, b)$.

All atoms are reachable by the opponent. Hence $\text{Team}(S, y < x) = \text{Team}(S, y \subseteq x) = X$. Thus, consistency of S means:

$(A, <) \models_X (y < x)$: For all $s : (x, y) \mapsto (a, b)$ in X, we have that $b < a$

Example: Infinite descending chains

$$(A, <) \models \exists x \exists y (y < x \wedge y \subseteq x) \quad \Longleftrightarrow \quad (A, <) \text{ is not well-founded}$$

A consistent winning strategy $S = (W, F)$ for this sentence just amounts to a selection of a team X of assignments $s : (x, y) \mapsto (a, b)$.

All atoms are reachable by the opponent. Hence $\text{Team}(S, y < x) = \text{Team}(S, y \subseteq x) = X$. Thus, consistency of S means:

$(A, <) \models_X (y < x)$: For all $s : (x, y) \mapsto (a, b)$ in X, we have that $b < a$

$(A, <) \models_X (y \subseteq x)$: For all $s : (x, y) \mapsto (a, b)$ in X, we have another assignment $t : (x, y) \mapsto (b, c)$ in X.

Example: Infinite descending chains

$$(A, <) \models \exists x \exists y (y < x \wedge y \subseteq x) \iff (A, <) \text{ is not well-founded}$$

A consistent winning strategy $S = (W, F)$ for this sentence just amounts to a selection of a team X of assignments $s : (x, y) \mapsto (a, b)$.

All atoms are reachable by the opponent. Hence $\mathrm{Team}(S, y < x) = \mathrm{Team}(S, y \subseteq x) = X$. Thus, consistency of S means:

$(A, <) \models_X (y < x)$: For all $s : (x, y) \mapsto (a, b)$ in X, we have that $b < a$

$(A, <) \models_X (y \subseteq x)$: For all $s : (x, y) \mapsto (a, b)$ in X, we have another assignment $t : (x, y) \mapsto (b, c)$ in X.

Hence we have a consistent winning strategy in the model-checking game if, and only if, $(A, <)$ has an infinite descending chain.

Inclusion logic and least fixed-point logic

Theorem. (Galliani and Hella) For every formula $\psi(X, \overline{z})$ in posGFP one can construct a formula $\varphi(\overline{z}) \in$ Inc, and vice versa, such that, for all $\mathfrak{A}$ and all X

$$\mathfrak{A} \models_X \varphi \quad \Longleftrightarrow \quad F_\psi^{\mathfrak{A}}(X) \supseteq X \quad \Longleftrightarrow \quad (\mathfrak{A}, X) \models_s \psi \text{ for all } s \in X$$

Here $F_\psi^{\mathfrak{A}} : X \mapsto \{\overline{a} : (\mathfrak{A}, X) \models \psi(X, \overline{a})\}$

For the case of sentences, ψ and φ are equivalent.

Corollary. For sentences, Inc and posGFP have the same expressive power.

Corollary. On finite structures, inclusion logic and LFP have the same expressive power. In particular, on ordered finite structures, inclusion logic captures PTIME.

We want to explain this result, with a game-theoretic view.

The gfp-fragment of LFP

The fragment posGFP of least fixed-point logic can be defined in two equivalent ways:

(1) posGFP is the closure of the set of formulae of form $[\mathbf{gfp}\, R.\overline{x}\, .\, \varphi(R, \overline{x})]\overline{(}x)$, where $\varphi(R\overline{x})$ is in FO, under disjunction, conjunction, quantifiers, and applications of gfp.

(2) posGFP is the set of relations definable by simultaneous greatest fixed points of systems of first-order formulae.

The gfp-fragment of LFP

The fragment posGFP of least fixed-point logic can be defined in two equivalent ways:

(1) posGFP is the closure of the set of formulae of form $[\mathbf{gfp}\, R.\overline{x}\,.\, \varphi(R, \overline{x})]\overline{(}x)$, where $\varphi(R\overline{x})$ is in FO, under disjunction, conjunction, quantifiers, and applications of gfp.

(2) posGFP is the set of relations definable by simultaneous greatest fixed points of systems of first-order formulae.

It is well-known that these two formulations are equivalent.

The alternation hierarchy

Fixed-point formulae become difficult to read and evaluate if they involve more than (very) few alternations of least and greatest fixed points.

Alternation between lfp- and gfp- operations define a hierarchy analogous to the Σ/Π hierarchies in first-order and second-order logic.

The fragment posGFP is at the bottom level of the alternation hierarchy of least fixed-point logic.

The alternation hierarchy

Fixed-point formulae become difficult to read and evaluate if they involve more than (very) few alternations of least and greatest fixed points.

Alternation between lfp- and gfp- operations define a hierarchy analogous to the Σ/Π hierarchies in first-order and second-order logic.

The fragment posGFP is at the bottom level of the alternation hierarchy of least fixed-point logic.

Theorem. (Immerman)
On finite structures the alternation hierarchy collapses: LFP ≡ posGFP.

However, this is not the case in general. For instance on $(\mathbb{N}, +, \cdot)$ the alternation hierarchy of LFP is strict.

Model-checking games for LFP and posGFP

The model-checking games for LFP are parity games. It is, in the general case, not known whether these can be solved in polynomial time.

The model-checking games for posGFP are much simpler. They are safety games in which all infinite plays are won by the Verifier.

Model-checking games for LFP and posGFP

The model-checking games for LFP are parity games. It is, in the general case, not known whether these can be solved in polynomial time.

The model-checking games for posGFP are much simpler. They are safety games in which all infinite plays are won by the Verifier.

Conversely, winning regions of safety games are definable in pos GFP:

Safety: Player 0 can avoid losing $\mathcal{G}$ from position v

$$\iff$$

$$\mathcal{G} = (V, V_0, V_1, E) \vDash [\mathbf{gfp}\, Wx\,.\,(V_0 x \wedge \exists y(Exy \wedge Wy)) \vee (V_1 x \wedge \forall y(Exy \rightarrow Wy))](v)$$

Inclusion logic is closed under union of teams

Dependence logic is closed downwards: If $\mathfrak{A} \models_X \psi$ and $Y \subseteq X$ then $\mathfrak{A} \models_Y \psi$.
This is not true for inclusion statements $x \subseteq y$.

Inclusion logic is closed under union of teams

Dependence logic is closed downwards: If $\mathfrak{A} \models_X \psi$ and $Y \subseteq X$ then $\mathfrak{A} \models_Y \psi$.
This is not true for inclusion statements $x \subseteq y$.

However, inclusion logic also has an interesting closure property.

Proposition. If $\mathfrak{A} \models_X \psi$ and $\mathfrak{A} \models_Y \psi$, then $\mathfrak{A} \models_{X \cup Y} \psi$

Remark. Notice that dependence logic, independence logic and in fact even constancy formulae $=(x)$ are not closed under unions of teams.

Corollary. For every structure $\mathfrak{A}$, every team X and every formula $\psi \in \text{Inc}$ there is the unique maximal subteam $X_{\max} \subseteq X$ with $\mathfrak{A} \models_{X_{\max}} \psi$.

Inclusion statements and greatest fixed points

Given a team X, the maximal subteam $X_{\max} \subseteq X$ satisfying inclusion statement $(x_i \subseteq x_j)$ is naturally definable by a gfp-induction:

$X^1 := X,$
$X^{\alpha+1} := \{s \in X^\alpha : (\exists s' \in X^\alpha) s'(x_j) = s(x_i)\}$
$X^\lambda = \bigcap_{\alpha<\lambda} X^\alpha$ for limit ordinals λ.

Hence $X_{\max}$ is uniformly definable by the posGFP-formula

$$\psi(X, \overline{z}) := [\mathbf{gfp}\, Y\overline{x} \,.\, X\overline{x} \wedge \exists \overline{y}(Y\overline{y} \wedge y_j = x_i)](\overline{z})$$

Further $\mathfrak{A} \models_X (x_i \subseteq x_j)$ if, and only if $(\mathfrak{A}, X) \models \forall \overline{z}(X\overline{z} \rightarrow \psi(X, \overline{z}))$

Inclusion statements and greatest fixed points

Given a team X, the maximal subteam $X_{\max} \subseteq X$ satisfying inclusion statement $(x_i \subseteq x_j)$ is naturally definable by a gfp-induction:

$X^1 := X,$
$X^{\alpha+1} := \{s \in X^\alpha : (\exists s' \in X^\alpha) s'(x_j) = s(x_i)\}$
$X^\lambda = \bigcap_{\alpha<\lambda} X^\alpha$ for limit ordinals λ.

Hence $X_{\max}$ is uniformly definable by the posGFP-formula

$$\psi(X, \overline{z}) := [\mathbf{gfp}\, Y\overline{x} \,.\, X\overline{x} \wedge \exists \overline{y}(Y\overline{y} \wedge y_j = x_i)](\overline{z})$$

Further $\mathfrak{A} \models_X (x_i \subseteq x_j)$ if, and only if $(\mathfrak{A}, X) \models \forall \overline{z}(X\overline{z} \rightarrow \psi(X, \overline{z}))$

This generalizes to all formulae of inclusion logic.

Translating posGFP to inclusion logic

For every formula $\psi(X, \overline{z})$ in posGFP one can construct a formula $\varphi(\overline{z}) \in \text{Inc}$ such that, for all $\mathfrak{A}$ and all X

$$\mathfrak{A} \vDash_X \varphi \quad \Longleftrightarrow \quad F_\psi^{\mathfrak{A}}(X) \supseteq X \quad \Longleftrightarrow \quad (\mathfrak{A}, X) \vDash_s \psi \text{ for all } s \in X$$

Translating posGFP to inclusion logic

For every formula $\psi(X, \overline{z})$ in posGFP one can construct a formula $\varphi(\overline{z}) \in \text{Inc}$ such that, for all $\mathfrak{A}$ and all X

$$\mathfrak{A} \models_X \varphi \quad \Longleftrightarrow \quad F_\psi^{\mathfrak{A}}(X) \supseteq X \quad \Longleftrightarrow \quad (\mathfrak{A}, X) \models_s \psi \text{ for all } s \in X$$

Intuition. In a gfp-induction, we justify that a tuple in $\overline{z} \in X^\alpha$ survives the next iteration, i.e. that $\overline{z} \in X^{\alpha+1}$, by means of statements of form $\overline{y} \in X^\alpha$ where $\overline{y}$ is related to $\overline{z}$ by first-order operations (or, equivalently, moves in a first-order game). By evaluating the corresponding Inc-formula on a team (with variables $\overline{x}$) that represents X^α we can use inclusion statements $\overline{y} \subseteq \overline{x}$ for expressing that $\overline{y} \in X^\alpha$.

Illustration via safety games

$\psi(W, x) = (V_0 x \wedge \exists y(Exy \wedge Wy)) \vee (V_1 x \wedge \forall y(Exy \rightarrow Wy))$

$\mathcal{G} = (V, V_0, V_1, E) \vDash [\mathbf{gfp}\, Wx\,.\, \psi(W, x)](v)$ if Player 0 can avoid losing from v. Further $F_\psi^{\mathcal{G}}(W) \supseteq W$ if Player 0 has a strategy not to leave W.

Translation to inclusion logic (first attempt):

$\varphi(x) = (V_0 x \wedge \exists y(Exy \wedge y \subseteq x)) \vee (V_1 x \wedge \forall y(\neg Exy \vee y \subseteq x))$

We want to prove that $F_\psi^{\mathcal{G}}(W) \supseteq W \iff \mathcal{G} \vDash_W \varphi(x)$.

Problem: φ is a disjunction and we have to split the team W into the subteams $W \cap V_0$ and $W \cap V_1$ and check that $\mathcal{G} \vDash_{W \cap V_0} \exists y(Exy \wedge y \subseteq x)$ and $\mathcal{G} \vDash_{W \cap V_1} \forall y(\neg Exy \vee y \subseteq x)$. But now only the values for x in the subteams are available for the inclusion statements. The formula is not correct.

Illustration via safety games

$\psi(W,x) = (V_0 x \wedge \exists y(Exy \wedge Wy)) \vee (V_1 x \wedge \forall y(Exy \rightarrow Wy))$

$\mathcal{G} = (V, V_0, V_1, E) \models [\mathbf{gfp}\, Wx\,.\,\psi(W,x)](v)$ if Player 0 can avoid losing from v. Further $F_\psi^{\mathcal{G}}(W) \supseteq W$ if Player 0 has a strategy not to leave W.

Translation to inclusion logic (corrected):

$\varphi(x) = \exists z(z \subseteq x \wedge ((V_0 x \wedge \exists y(Exy \wedge y \subseteq z)) \vee (V_1 x \wedge \forall y(\neg Exy \vee y \subseteq z))$

Claim. $F_\psi^{\mathcal{G}}(W) \supseteq W \iff \mathcal{G} \models_W \varphi(x)$.

Proof. Choose for z at each assignment all values that x takes in W. φ is a disjunction and we have to split the team W into the subteams $W \cap V_0$ and $W \cap V_1$ and check that $\mathcal{G} \models_{W \cap V_0} \exists y(Exy \wedge y \subseteq z)$ and $\mathcal{G} \models_{W \cap V_1} \forall y(\neg Exy \vee y \subseteq z)$. Also in the subteams z takes all values in W, hence $y \subseteq z$ correctly expresses that $y \in W$.

Safety games for inclusion logic

Let $\mathcal{G}(\mathfrak{A}, \psi)$ be a second-order reachability game for a formula $\psi \in \text{Inc}$.
We define an associated safety game $\mathcal{H}(\mathfrak{A}, \psi)$ as follows:

From positions $(\overline{x} \subseteq \overline{y}, s)$, Verifier can move to any position $(\overline{x} \subseteq \overline{y}, t)$ such that $t(\overline{y}) = s(\overline{x})$. From there, Falsifier can move upwards in the game tree to any position (φ, t') with $t'(\overline{x}, \overline{y}) = t(\overline{x}, \overline{y})$.

Safety games for inclusion logic

Let $\mathcal{G}(\mathfrak{A}, \psi)$ be a second-order reachability game for a formula $\psi \in$ Inc.
We define an associated safety game $\mathcal{H}(\mathfrak{A}, \psi)$ as follows:

From positions $(\overline{x} \subseteq \overline{y}, s)$, Verifier can move to any position $(\overline{x} \subseteq \overline{y}, t)$ such that $t(\overline{y}) = s(\overline{x})$. From there, Falsifier can move upwards in the game tree to any position (φ, t') with $t'(\overline{x}, \overline{y}) = t(\overline{x}, \overline{y})$.

Proposition. Every consistent winning strategy $S = (W, F)$ for $\mathcal{G}(\mathfrak{A}, \psi)$ is also a winning strategy for $\mathcal{H}(\mathfrak{A}, \psi)$, and vice versa.

$\mathfrak{A} \models_X \psi$ if, and only if, Verifier has a winning strategy in $\mathcal{H}(\mathfrak{A}, \psi)$ from all positions (ψ, s) with $s \in X$.

Safety games for inclusion logic

Let $\mathcal{G}(\mathfrak{A}, \psi)$ be a second-order reachability game for a formula $\psi \in$ Inc.
We define an associated safety game $\mathcal{H}(\mathfrak{A}, \psi)$ as follows:

From positions $(\overline{x} \subseteq \overline{y}, s)$, Verifier can move to any position $(\overline{x} \subseteq \overline{y}, t)$ such that $t(\overline{y}) = s(\overline{x})$. From there, Falsifier can move upwards in the game tree to any position (φ, t') with $t'(\overline{x}, \overline{y}) = t(\overline{x}, \overline{y})$.

Proposition. Every consistent winning strategy $S = (W, F)$ for $\mathcal{G}(\mathfrak{A}, \psi)$ is also a winning strategy for $\mathcal{H}(\mathfrak{A}, \psi)$, and vice versa.

$\mathfrak{A} \models_X \psi$ if, and only if, Verifier has a winning strategy in $\mathcal{H}(\mathfrak{A}, \psi)$ from all positions (ψ, s) with $s \in X$.

Moreover the safety game $\mathcal{H}(\mathfrak{A}, \psi)$ in $\mathfrak{A}$ is interpretable in $\mathfrak{A}$ by a first-order interpretation I_ψ.

Game-based translations between Inc and posGFP

For every formula $\psi(\overline{x}) \in \mathrm{Inc}$ there is a first-order interpretation I_ψ that interprets the safety game $\mathcal{H}(\mathfrak{A}, \psi)$ in $\mathfrak{A}$. That is: $I_\psi : \mathfrak{A} \mapsto \mathcal{H}(\mathfrak{A}, \psi)$.

Game-based translations between Inc and posGFP

For every formula $\psi(\overline{x}) \in \text{Inc}$ there is a first-order interpretation I_ψ that interprets the safety game $\mathcal{H}(\mathfrak{A}, \psi)$ in $\mathfrak{A}$. That is: $I_\psi : \mathfrak{A} \mapsto \mathcal{H}(\mathfrak{A}, \psi)$.

Let $\text{Win}(x) \in \text{posGFP}$ define the winning regions of safety games:
$\mathcal{H}(\mathfrak{A}, \psi) \models \text{Win}(v)$ if, and only if, Verifier wins $\mathcal{H}(\mathfrak{A}, \psi)$ from v.

Game-based translations between Inc and posGFP

For every formula $\psi(\overline{x}) \in \text{Inc}$ there is a first-order interpretation I_ψ that interprets the safety game $\mathcal{H}(\mathfrak{A}, \psi)$ in $\mathfrak{A}$. That is: $I_\psi : \mathfrak{A} \mapsto \mathcal{H}(\mathfrak{A}, \psi)$.

Let $\text{Win}(x) \in \text{posGFP}$ define the winning regions of safety games:
$\mathcal{H}(\mathfrak{A}, \psi) \vDash \text{Win}(v)$ if, and only if, Verifier wins $\mathcal{H}(\mathfrak{A}, \psi)$ from v.

Using the interpretation I_ψ the formula $\text{Win}(x)$ translates in to a posGFP-formula $\text{Win}^*(\overline{x})$ on $\mathfrak{A}$ such that
$(\mathfrak{A}, X) \vDash_s \text{Win}^*$ if and only if Verifier wins $\mathcal{H}(\mathfrak{A}, \psi)$ from (ψ, s).

Game-based translations between Inc and posGFP

For every formula $\psi(\overline{x}) \in \mathrm{Inc}$ there is a first-order interpretation I_ψ that interprets the safety game $\mathcal{H}(\mathfrak{A}, \psi)$ in $\mathfrak{A}$. That is: $I_\psi : \mathfrak{A} \mapsto \mathcal{H}(\mathfrak{A}, \psi)$.

Let $\mathrm{Win}(x) \in \mathrm{posGFP}$ define the winning regions of safety games:
$\mathcal{H}(\mathfrak{A}, \psi) \models \mathrm{Win}(v)$ if, and only if, Verifier wins $\mathcal{H}(\mathfrak{A}, \psi)$ from v.

Using the interpretation I_ψ the formula $\mathrm{Win}(x)$ translates in to a posGFP-formula $\mathrm{Win}^*(\overline{x})$ on $\mathfrak{A}$ such that
$(\mathfrak{A}, X) \models_s \mathrm{Win}^*$ if and only if Verifier wins $\mathcal{H}(\mathfrak{A}, \psi)$ from (ψ, s).

Together we have
$\mathfrak{A} \models_X \psi \quad \Longleftrightarrow \quad (\mathfrak{A}, X) \models_s \mathrm{Win}^*$ for all $s \in X$

Game-based translations between Inc and posGFP

For every formula $\psi(\overline{x}) \in \text{Inc}$ there is a first-order interpretation I_ψ that interprets the safety game $\mathcal{H}(\mathfrak{A}, \psi)$ in $\mathfrak{A}$. That is: $I_\psi : \mathfrak{A} \mapsto \mathcal{H}(\mathfrak{A}, \psi)$.

Let $\text{Win}(x) \in \text{posGFP}$ define the winning regions of safety games:
$\mathcal{H}(\mathfrak{A}, \psi) \models \text{Win}(v)$ if, and only if, Verifier wins $\mathcal{H}(\mathfrak{A}, \psi)$ from v.

Using the interpretation I_ψ the formula $\text{Win}(x)$ translates in to a posGFP-formula $\text{Win}^*(\overline{x})$ on $\mathfrak{A}$ such that
$(\mathfrak{A}, X) \models_s \text{Win}^*$ if and only if Verifier wins $\mathcal{H}(\mathfrak{A}, \psi)$ from (ψ, s).

Together we have
$\mathfrak{A} \models_X \psi \quad \Longleftrightarrow \quad (\mathfrak{A}, X) \models_s \text{Win}^* \text{ for all } s \in X$

An analogous argument works for the translation of posGFP into Inc.

Negation

Define the semantic extension of ψ on $\mathfrak{A}$ as $[\![\psi]\!]^{\mathfrak{A}} := \{X : \mathfrak{A} \models_X \psi\}$.

Negation

Define the semantic extension of ψ on $\mathfrak{A}$ as $[[\psi]]^{\mathfrak{A}} := \{X : \mathfrak{A} \models_X \psi\}$.

Negation is not a semantic operation, contrary to disjunction, conjunction, and quantifiers.

Negation

Define the semantic extension of ψ on $\mathfrak{A}$ as $[\![\psi]\!]^{\mathfrak{A}} := \{X : \mathfrak{A} \models_X \psi\}$.

Negation is not a semantic operation, contrary to disjunction, conjunction, and quantifiers.

When we know $[\![\psi]\!]^{\mathfrak{A}}$ and $[\![\varphi]\!]^{\mathfrak{A}}$ we can easily compute $[\![\varphi \wedge \psi]\!]^{\mathfrak{A}}$ and $[\![\varphi \vee \psi]\!]^{\mathfrak{A}}$ (without even knowing the syntax of ψ and φ). Analogous observations for quantifiers.

However, knowing $[\![\psi]\!]^{\mathfrak{A}}$ does not provide much knowledge about $[\![\psi^{\neg}]\!]^{\mathfrak{A}}$.

Negation and interpolation

We only consider structures with at least wo elements. Two formulae ψ and φ are contradictory if $[\![\psi]\!]^{\mathfrak{A}} \cap [\![\varphi]\!]^{\mathfrak{A}} = \{\varnothing\}$ for any structure $\mathfrak{A}$.

Proposition. (Kontinen and Väänänen) For any two contradictory formulae ψ and φ of dependence logic there is a formula ϑ such that $[\![\vartheta]\!]^{\mathfrak{A}} = [\![\psi]\!]^{\mathfrak{A}}$ and $[\![\vartheta^{\neg}]\!]^{\mathfrak{A}} = [\![\varphi]\!]^{\mathfrak{A}}$ for all $\mathfrak{A}$.

Negation and interpolation

We only consider structures with at least wo elements. Two formulae ψ and φ are contradictory if $[\![\psi]\!]^{\mathfrak{A}} \cap [\![\varphi]\!]^{\mathfrak{A}} = \{\varnothing\}$ for any structure $\mathfrak{A}$.

Proposition. (Kontinen and Väänänen) For any two contradictory formulae ψ and φ of dependence logic there is a formula ϑ such that $[\![\vartheta]\!]^{\mathfrak{A}} = [\![\psi]\!]^{\mathfrak{A}}$ and $[\![\vartheta^{\neg}]\!]^{\mathfrak{A}} = [\![\varphi]\!]^{\mathfrak{A}}$ for all $\mathfrak{A}$.

In this form, this is not true in general for logics with team semantics. For instance independence logic contains $\forall x(x \subseteq y)$ and $=(y)$ which are contradictory but we will see that they have no such interpolant.

Strongly contradictory formulae

Two formulae ψ and φ are strongly contradictory if $X \cap Y = \varnothing$ for all teams X, Y and all structures $\mathfrak{A}$ such that $\mathfrak{A} \models_X \psi$ and $\mathfrak{A} \models_Y \varphi$.

Lemma. Every formula is strongly contradictory to its negation.

By induction. For atomic formulae, this is true by definition. Consider $\psi = \varphi \vee \vartheta$ and $\psi^{\neg} = \varphi^{\neg} \wedge \vartheta^{\neg}$. If $\mathfrak{A} \models_X \psi$ then $X = X' \cup X''$ with $\mathfrak{A} \models_{X'} \varphi$ and $\mathfrak{A} \models_{X''} \vartheta$. If $\mathfrak{A} \models_Y \psi^{\neg}$ then $\mathfrak{A} \models_Y \varphi^{\neg}$ and $\mathfrak{A} \models_Y \vartheta^{\neg}$. Hence $X' \cap Y = X'' \cap Y = \varnothing$ and thus also $X \cap Y = \varnothing$.

Finally let $\psi = \exists y \, \varphi$ and $\psi^{\neg} = \forall y \, \varphi^{\neg}$. If $\mathfrak{A} \models_X \psi$ then $\mathfrak{A} \models_{X[y \mapsto F]} \varphi$ for some $F : X \to \mathcal{P}(A) \setminus \{\varnothing\}$. If $\mathfrak{A} \models_Y \psi^{\neg}$ then $\mathfrak{A} \models_{Y[y \mapsto A]} \varphi$. Hence $X[y \mapsto F] \cap Y[y \mapsto A] = \varnothing$ and thus $X \cap Y = \varnothing$.

Strongly contradictory formulae

Two formulae ψ and φ are strongly contradictory if $X \cap Y = \varnothing$ for all teams X, Y and all structures $\mathfrak{A}$ such that $\mathfrak{A} \models_X \psi$ and $\mathfrak{A} \models_Y \varphi$.

Lemma. Every formula is strongly contradictory to its negation.

By induction. For atomic formulae, this is true by definition. Consider $\psi = \varphi \vee \vartheta$ and $\psi^{\neg} = \varphi^{\neg} \wedge \vartheta^{\neg}$. If $\mathfrak{A} \models_X \psi$ then $X = X' \cup X''$ with $\mathfrak{A} \models_{X'} \varphi$ and $\mathfrak{A} \models_{X''} \vartheta$. If $\mathfrak{A} \models_Y \psi^{\neg}$ then $\mathfrak{A} \models_Y \varphi^{\neg}$ and $\mathfrak{A} \models_Y \vartheta^{\neg}$. Hence $X' \cap Y = X'' \cap Y = \varnothing$ and thus also $X \cap Y = \varnothing$.

Finally let $\psi = \exists y\, \varphi$ and $\psi^{\neg} = \forall y\, \varphi^{\neg}$. If $\mathfrak{A} \models_X \psi$ then $\mathfrak{A} \models_{X[y \mapsto F]} \varphi$ for some $F : X \to \mathcal{P}(A) \setminus \{\varnothing\}$. If $\mathfrak{A} \models_Y \psi^{\neg}$ then $\mathfrak{A} \models_{Y[y \mapsto A]} \varphi$. Hence $X[y \mapsto F] \cap Y[y \mapsto A] = \varnothing$ and thus $X \cap Y = \varnothing$.

Remark. Contradictory formulae of dependence logic are in fact strongly contradictory. Indeed, suppose that $\mathfrak{A} \models_X \psi$ and $\mathfrak{A} \models_Y \varphi$. Then, by downwards closure, $\mathfrak{A} \models_{X \cap Y} \psi \wedge \varphi$, so $X \cap Y = \varnothing$.

Contradictory formulae without interpolants

Corollary. Only a pair of strongly contradictory formulae ψ, φ can have an interpolant ϑ with $\psi \equiv \vartheta$ and $\varphi \equiv \vartheta^{\neg}$.

The formulae $\forall x(x \subseteq y)$ and $=(y)$ are contradictory (on structures with at least two elements) but have no interpolant, since they are not strongly contradictory.

Indeed, let s_a be the assignment $y \mapsto a$. Then $[\![\forall x(x \subseteq y]\!]^{\mathfrak{A}} = \{X\}$ with $X = \{s_a : a \in A\}$, whereas $[\![=(y)]\!]^{\mathfrak{A}} = \{\{s_a\} : a \in A\}$, and we have $X \cap \{s_a\} = \{s_a\} \neq \varnothing$.

Contradictory formulae without interpolants

Corollary. Only a pair of strongly contradictory formulae ψ, φ can have an interpolant ϑ with $\psi \equiv \vartheta$ and $\varphi \equiv \vartheta^{\neg}$.

The formulae $\forall x(x \subseteq y)$ and $=(y)$ are contradictory (on structures with at least two elements) but have no interpolant, since they are not strongly contradictory.

Indeed, let s_a be the assignment $y \mapsto a$. Then $[\![\forall x(x \subseteq y]\!]^{\mathfrak{A}} = \{X\}$ with $X = \{s_a : a \in A\}$, whereas $[\![=(y)]\!]^{\mathfrak{A}} = \{\{s_a\} : a \in A\}$, and we have $X \cap \{s_a\} = \{s_a\} \neq \emptyset$.

Being strongly contradictory is a necessary condition for having an interpolant. We will see that it is also sufficient.

Completely undetermined sentences

A sentence is completely undetermined if neither the sentence itself, nor its negation is true on any structure with more than one element.

Examples: $\forall x \; {=}(x)$ or $\forall x \exists y (x \bot y \wedge x = y)$

Proposition. Let L be any logic with team semantics, closed under first-order operations, containing a completely undetermined sentence $\bot_\bot$. Then for any $\psi, \varphi \in L$, the following are equivalent

(1) There is a formula η such that $\psi \models \eta$ and $\varphi \models \eta^\neg$

(2) There is a formula ϑ such that $\psi \equiv \vartheta$ and $\varphi \equiv \vartheta^\neg$

Proof. (2) $\Rightarrow$ (1): Take $\eta = \vartheta$.

(1) $\Rightarrow$ (2): Take $\vartheta := (\psi \vee \bot_\bot) \wedge ((\varphi \vee \bot_\bot)^\neg \vee \eta)$. Then

$$\vartheta \equiv \psi \wedge (\bot \vee \eta) \equiv \psi \wedge \eta \equiv \psi$$

$$\vartheta^\neg = (\psi \vee \bot_\bot)^\neg \vee ((\varphi \vee \bot_\bot) \wedge \eta^\neg) \equiv \bot \vee (\varphi \wedge \eta^\neg) \equiv \varphi \wedge \eta^\neg \equiv \varphi.$$

1-closed formulae

A formula ψ 1-closed if whenever $\mathfrak{A} \models_X \psi$ then also $\mathfrak{A} \models_{\{s\}} \psi$ for all $s \in X$.

All formulae of dependence logic are 1-closed. Further independence atoms (and in fact all purely existential formulae of indepence logic) are 1-closed.

Lemma. Let L be a logic with team semantics, that is closed under first.order operations and translatable into Σ_1^1. Then for every formula $\psi \in L$ there exists a formula $\psi^\downarrow$ in dependence logic such that the teams satisfying $\psi^\downarrow$ are exactly the subteams of the teams satisfying ψ.

Proof. Translate ψ into $\psi^*(Y) \in \Sigma_1^1$ and let
$\varphi(X) = \exists Y(\forall \overline{x}(X\overline{x} \to Y\overline{x}) \wedge \psi^*(Y))$. Since X appears only negatively in $\varphi(X)$, it can be translated into an equivalent formula $\psi^\downarrow$.

Notice that $\psi \models \psi^\downarrow$ and that $\psi^\downarrow$ is 1-closed. Further ψ and φ are strongly contradictory if, and only if, $\psi^\downarrow$ and $\varphi^\downarrow$ are contradictory (and hence strongly contradictory).

The Interpolation Theorem

Let L be a logic with team semantics, which contains FO and can be embedded into Σ_1^1, and which has a totally undetermined sentence.

Theorem. For any two strongly contradictory formulae ψ, φ from L, there exists a formula ϑ in L such that $\psi \equiv \vartheta$ and $\varphi \equiv \vartheta^{\neg}$.

Let $\exists \overline{R} \tilde{\psi}(\overline{R}, X)$ and $\exists \overline{S} \tilde{\varphi}(\overline{S}, X)$ be Σ_1^1-translations of $\psi^{\downarrow}$ and $\varphi^{\downarrow}$. Then $\tilde{\psi}(\overline{R}, X) \vDash (\neg\tilde{\varphi}(\overline{S}, X) \vee X = \varnothing)$ is a valid first-order implication.

By Craig's Interpolation Theorem there is a first-order sentence $\tilde{\eta}(X)$ such that $\tilde{\psi}(\overline{R}, X) \vDash \tilde{\eta}(X)$ and $(\tilde{\varphi}(\overline{S}, X) \wedge X \neq \varnothing) \vDash \neg\tilde{\eta}(X)$.

Let $\eta(\overline{x}) := \tilde{\eta}[X\overline{z}/\overline{z} = \overline{x}]$. For any team $\{s\}$ of size one, we have $(\mathfrak{A}, \{s\}) \vDash \tilde{\eta}$ if, and only if, $\mathfrak{A} \vDash_{\{s\}} \eta$.

Claim. $\psi \vDash \eta$ and $\varphi \vDash \neg\eta$ (and hence an interpolant ϑ exists)

The Interpolation Theorem

Claim. $\psi \models \eta$ and $\varphi \models \neg\eta$.

Let $\mathfrak{A} \models_X \psi$. Then also $\mathfrak{A} \models_X \psi^{\downarrow}$ and, since $\psi^{\downarrow}$ is 1-closed, also $\mathfrak{A} \models_{\{s\}} \psi^{\downarrow}$ for all $s \in X$. This implies that $\mathfrak{A} \models \exists \overline{R} \tilde{\psi}(\overline{R}, \{s\})$ and therefore $\mathfrak{A} \models \tilde{\eta}(\{s\})$ and thus $\mathfrak{A} \models_{\{s\}} \eta$ for all $s \in X$. But since $\eta \in \text{FO}$ this implies that $\mathfrak{A} \models_X \eta$.

The Interpolation Theorem

Claim. $\psi \models \eta$ and $\varphi \models \neg\eta$.

Let $\mathfrak{A} \models_X \psi$. Then also $\mathfrak{A} \models_X \psi^{\downarrow}$ and, since $\psi^{\downarrow}$ is 1-closed, also $\mathfrak{A} \models_{\{s\}} \psi^{\downarrow}$ for all $s \in X$. This implies that $\mathfrak{A} \models \exists \overline{R} \tilde{\psi}(\overline{R}, \{s\})$ and therefore $\mathfrak{A} \models \tilde{\eta}(\{s\})$ and thus $\mathfrak{A} \models_{\{s\}} \eta$ for all $s \in X$. But since $\eta \in \mathrm{FO}$ this implies that $\mathfrak{A} \models_X \eta$.

Suppose $\mathfrak{A} \models_X \varphi$. Again we get $\mathfrak{A} \models_{\{s\}} \varphi^{\downarrow}$ for all $s \in X$. Therefore $\mathfrak{A} \models \exists \overline{S} \tilde{\varphi}(\overline{S}, \{s\})$ and hence $\mathfrak{A} \models \neg\tilde{\eta}(\{s\})$. This implies $\mathfrak{A} \models_{\{s\}} \neg\eta$ for all $s \in X$, and since $\eta \in \mathrm{FO}$, we have $\mathfrak{A} \models_X \neg\eta$.

Deterministic versus nondeterministic strategies

Our notion of consistent winning strategies is nondeteministic:

A consistent winning strategy of Player 0 for $\mathcal{G}$ and Win is a pair $S = (W, F) \subseteq (V, E)$ with $F \subseteq (W \times W) \cap E$ such that

(1) W is the set of nodes that are reachable from I via edges in F

(2) if $v \in V_0 \cap W$ then $vF \neq \varnothing$

(3) if $v \in V_1 \cap W$ then $vF = vE$

(4) $W \cap T \in \text{Win}$.

Deterministic versus nondeterministic strategies

The deterministic variant:

A deterministic consistent winning strategy of Player 0 for $\mathcal{G}$ and Win is a pair $S = (W, F) \subseteq (V, E)$ with $F \subseteq (W \times W) \cap E$ such that

(1) W is the set of nodes that are reachable from I via edges in F

(2) if $v \in V_0 \cap W$ then $|vF| = 1$

(3) if $v \in V_1 \cap W$ then $vF = vE$

(4) $W \cap T \in \text{Win}$.

In most classical games, deterministic strategies are no less powerful than nondeterministic ones. Is this also the case for second-order reachability games?

Why is the nondeterministic semantics the right one ?

Consider the formula $\exists x(y \subseteq x \wedge z \subseteq x)$ which says

> *The team under consideration can be extended by values for x such that all values for y and z in the team occur als values for x*

Why is the nondeterministic semantics the right one ?

Consider the formula $\exists x(y \subseteq x \wedge z \subseteq x)$ which says

> *The team under consideration can be extended by values for x such that all values for y and z in the team occur als values for x*

True for any team X: assign to x all values occurring for y and z in X.

Why is the nondeterministic semantics the right one ?

Consider the formula $\exists x(y \subseteq x \wedge z \subseteq x)$ which says

> *The team under consideration can be extended by values for x such that all values for y and z in the team occur als values for x*

True for any team X: assign to x all values occurring for y and z in X.

But under deterministic (strict) semantics for $(\exists x)$ this not the case. Let $X = \{s\}$ with $s(y) \neq (z)$. By chosing just one witness for x, we cannot make the formula true for this team.

So under deterministic semantics, this formula says something different.

Why is the nondeterministic semantics the right one ?

Consider the formula $\exists x(y \subseteq x \wedge z \subseteq x)$ which says

> *The team under consideration can be extended by values for x such that all values for y and z in the team occur als values for x*

True for any team X: assign to x all values occurring for y and z in X.

But under deterministic (strict) semantics for $(\exists x)$ this not the case. Let $X = \{s\}$ with $s(y) \neq (z)$. By chosing just one witness for x, we cannot make the formula true for this team.

So under deterministic semantics, this formula says something different.

So what?

Why is the nondeterministic semantics the right one ?

Consider the formula $\exists x(y \subseteq x \wedge z \subseteq x)$ and the team

$X =$

y	z	u
1	2	3
1	2	4

Clearly $X \vDash \exists x(y \subseteq x \wedge z \subseteq x)$ even under deterministic semantics.

Why is the nondeterministic semantics the right one ?

Consider the formula $\exists x(y \subseteq x \land z \subseteq x)$ and the team

$X =$

y	z	u
1	2	3
1	2	4

Clearly $X \models \exists x(y \subseteq x \land z \subseteq x)$ even under deterministic semantics.

But under deterministic semantics $X \upharpoonright \{y, z\} \not\models \exists x(y \subseteq x \land z \subseteq x)$

Why is the nondeterministic semantics the right one ?

Consider the formula $\exists x(y \subseteq x \wedge z \subseteq x)$ and the team

$X =$

y	z	u
1	2	3
1	2	4

Clearly $X \models \exists x(y \subseteq x \wedge z \subseteq x)$ even under deterministic semantics.

But under deterministic semantics $X \upharpoonright \{y, z\} \not\models \exists x(y \subseteq x \wedge z \subseteq x)$

Hence deterministic semantics violates the locality principle !

Downwards Closure

In model-checking games for dependence logic, deterministic strategies suffice. The reason is that dependence logic is downwards closed for teams.

A winning condition $\text{Win} \subseteq \mathcal{P}(T)$ is downwards closed if $U \in \text{Win}$ and $Z \subseteq U$ imply $Z \in \text{Win}$.

Proposition. Let Win be downwards closed. Then Player 0 has a consistent winning strategy for G and Win if, and only if, she has a deterministic one.

Downwards Closure

In model-checking games for dependence logic, deterministic strategies suffice. The reason is that dependence logic is downwards closed for teams.

A winning condition $\text{Win} \subseteq \mathcal{P}(T)$ is downwards closed if $U \in \text{Win}$ and $Z \subseteq U$ imply $Z \in \text{Win}$.

Proposition. Let Win be downwards closed. Then Player 0 has a consistent winning strategy for G and Win if, and only if, she has a deterministic one.

Is this sufficient condition also necessary?

Downwards Closure

In model-checking games for dependence logic, deterministic strategies suffice. The reason is that dependence logic is downwards closed for teams.

A winning condition $\text{Win} \subseteq \mathcal{P}(T)$ is downwards closed if $U \in \text{Win}$ and $Z \subseteq U$ imply $Z \in \text{Win}$.

Proposition. Let Win be downwards closed. Then Player 0 has a consistent winning strategy for G and Win if, and only if, she has a deterministic one.

Is this sufficient condition also necessary?

No, but we can find a weaker condition, that is both necessary and sufficient for guaranteeing the possibility to elminate nondeterministic strategies.

Splits

A collection $\mathcal{F} \subseteq \mathcal{P}(T)$ has a split if there exist $U_1, U_2 \notin \mathcal{F}$ such that $U_1 \cup U_2 \in \mathcal{F}$.

Splits

A collection $\mathcal{F} \subseteq \mathcal{P}(T)$ has a split if there exist $U_1, U_2 \notin \mathcal{F}$ such that $U_1 \cup U_2 \in \mathcal{F}$.

If $\mathcal{F}$ is downwards closed, then it has no splits. The converse is not true.

Splits

A collection $\mathcal{F} \subseteq \mathcal{P}(T)$ has a split if there exist $U_1, U_2 \notin \mathcal{F}$ such that $U_1 \cup U_2 \in \mathcal{F}$.

If $\mathcal{F}$ is downwards closed, then it has no splits. The converse is not true.

Theorem. If $\text{Win} \subseteq \mathcal{P}(T)$ has no splits, then for any second-order reachability game (G, Win), Player 0 has a consistent winning strategy, if and only if, she has a deterministic one. Conversely, if $\text{Win} \subseteq \mathcal{P}(T)$ has a split, then there exists a game graph G with T as its set of terminal nodes such that Player 0 has a consistent winning strategy for (G, Win) but not a deterministic one.