Logics for dependence and independence

Erich Grädel

Nordic Spring School in Logic, Nordfjordeid, 27 - 31 May 2013

Part I: Dependence. Independence, Team Semantics

- Motivation and historical remarks
- Dependence and independence as atomic properties
- Logics of dependence and independence. Team semantics
- Expressive power

Logics of dependence and independence

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

Blass and Gurevich: correspondence to Σ_1^1 (and thus NP)

Logics of dependence and independence

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

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

Claim: Impossibility of model-theoretic (compositional) semantics
never made precise, never proved

Logics of dependence and independence

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

Claim: Impossibility of model-theoretic (compositional) semantics
never made precise, never proved

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

This kind of semantics is an achievement of independent interest. The full potential of this innovation has not yet been fully appreciated.

Henkin quantifiers

$$\varphi := \begin{pmatrix} \forall x & \exists y \\ \forall u & \exists v \end{pmatrix} Pxyuv$$

Intuitively, this says that for all x, u there exist y, v such that $Pxyuv$ holds, and moreover, the choice of y only depends on the value of x and the choice of v only depends on the value of u.

Henkin quantifiers

$$\varphi := \begin{pmatrix} \forall x & \exists y \\ \forall u & \exists v \end{pmatrix} Pxyuv$$

Intuitively, this says that for all x, u there exist y, v such that $Pxyuv$ holds, and moreover, the choice of y only depends on the value of x and the choice of v only depends on the value of u.

Precise semantics can be given either in terms of Skolem functions, or in terms of games of imperfect information.

Henkin quantifiers

$$\varphi := \begin{pmatrix} \forall x & \exists y \\ \forall u & \exists v \end{pmatrix} Pxyuv$$

Intuitively, this says that for all x, u there exist y, v such that $Pxyuv$ holds, and moreover, the choice of y only depends on the value of x and the choice of v only depends on the value of u.

Precise semantics can be given either in terms of Skolem functions, or in terms of games of imperfect information.

$(A, P) \models \varphi$ if there exist functions $f, g : A \to A$ such that $P(a, fa, c, gc)$ holds for all $a, c \in A$.

Henkin quantifiers and NP

Simple formulae with Henkin quantifiers express NP-complete problems

A graph $G = (V, E)$ is 3-colourable if, and only if,

$$G \models \begin{pmatrix} \forall x & \exists y \\ \forall u & \exists v \end{pmatrix} \big(y, v \in \{0,1,2\} \wedge (x = u \to y = v) \wedge (Exu \to y \neq v) \big)$$

Theorem (Blass, Gurevich)
Henkin quantfiers over first-order formulae capture NP

Independence-friendly logic

IF-logic is first-order logic where quantifiers are annoted by independencies

Quantification rule: If φ is a formula, x is a variable, and W is a finite set of variables, then the expressions $(\exists x/W)\varphi$ and $(\forall x/W)\varphi$ are also formulae.

Game-theoretic semantics: In the evaluation game for $(Qx/W)\varphi$, the value for x must be chosen independently from the the values of the variables in W.

At two positions $((Qx/W)\varphi, s)$ and $(Qx/W)\varphi, s')$ such that $s(y) \neq s'(y)$ only for variables in W, the same value for x must be chosen.

IF-logic and existential second-order logic

A graph $G = (V, E)$ admits a perfect matching if, and only if,

$$G \models \forall x \forall y (\exists u / \{y\})(\exists v / \{x, u\})\Big((x = y \rightarrow u = v) \land (u = y \rightarrow v = x) \land Exu\Big)$$

IF-logic and existential second-order logic

A graph $G = (V, E)$ admits a perfect matching if, and only if,

$$G \models \forall x \forall y (\exists u / \{y\})(\exists v / \{x, u\})\Big((x = y \rightarrow u = v) \wedge (u = y \rightarrow v = x) \wedge Exu\Big)$$

Skolem semantics. Replace existential quantifiers $(\exists x / W)$ by Skolem functions whose arguments are the variables outside W.

IF-logic and existential second-order logic

A graph $G = (V, E)$ admits a perfect matching if, and only if,

$$G \models \forall x \forall y (\exists u / \{y\})(\exists v / \{x, u\})\big((x = y \rightarrow u = v) \wedge (u = y \rightarrow v = x) \wedge Exu\big)$$

Skolem semantics. Replace existential quantifiers $(\exists x / W)$ by Skolem functions whose arguments are the variables outside W.

The formula for the perfect matching becomes

$$(\exists f)(\exists g) \forall x \forall y \big((x = y \rightarrow fx = gy) \wedge (fx = y \rightarrow gy = x) \wedge Exfx\big)$$

which is equivalent to $(\exists f) \forall x (ffx = x \wedge Exfx)$

IF-logic and existential second-order logic

A graph $G = (V, E)$ admits a perfect matching if, and only if,

$$G \models \forall x \forall y (\exists u / \{y\})(\exists v / \{x, u\})\Big((x = y \rightarrow u = v) \wedge (u = y \rightarrow v = x) \wedge Exu\Big)$$

Skolem semantics. Replace existential quantifiers $(\exists x / W)$ by Skolem functions whose arguments are the variables outside W.

The formula for the perfect matching becomes

$$(\exists f)(\exists g)\forall x \forall y\Big((x = y \rightarrow fx = gy) \wedge (fx = y \rightarrow gy = x) \wedge Exfx\Big)$$

which is equivalent to $(\exists f)\forall x(ffx = x \wedge Exfx)$

Theorem. In expressive power, IF-logic is equivalent to existential second-order logic and thus captures NP on finite structures.

Dependence and independence

What does it mean that "x depends on y" or that "x and y are independent" ?

Dependence and independence

What does it mean that “x depends on y” or that “x and y are independent” ?

Compare this to statements of a more common form such as “x divides y”.

Dependence and independence

What does it mean that "x depends on y" or that "x and y are independent" ?

Compare this to statements of a more common form such as "x divides y".

To make sense of this we fix a structure $\mathfrak{A}$ in which the notion of divisibility is well-defined and an assignment $s : \{x, y\} \to A$ and use Tarski-style semantics to determine whether $\mathfrak{A} \models_s$ "x divides y".

Dependence and independence

What does it mean that "x depends on y" or that "x and y are independent" ?

Compare this to statements of a more common form such as "x divides y".

To make sense of this we fix a structure $\mathfrak{A}$ in which the notion of divisibility is well-defined and an assignment $s : \{x, y\} \to A$ and use Tarski-style semantics to determine whether $\mathfrak{A} \models_s$ "x divides y".

Dependence and independence are notions of a different kind! They manifest themselves not in single assignments, but only in larger amounts of data:

- sets of plays in a game (e.g. strategies)
- tables or relations
- sets of assignments.

Dependence and independence

What does it mean that "x depends on y" or that "x and y are independent" ?

Compare this to statements of a more common form such as "x divides y".

To make sense of this we fix a structure $\mathfrak{A}$ in which the notion of divisibility is well-defined and an assignment $s : \{x, y\} \to A$ and use Tarski-style semantics to determine whether $\mathfrak{A} \models_s$ "x divides y".

Dependence and independence are notions of a different kind! They manifest themselves not in single assignments, but only in larger amounts of data:

- sets of plays in a game (e.g. strategies)
- tables or relations
- sets of assignments.

A set of assigments (all with the same domain of variables) is called a team.

Logics of dependence and independence

Väänänen: Dependence logic

rather than stating dependencies or independencies as annotations of quantifiers, treat dependence as an atomic statement

model-theoretic semantics in terms of sets of assignments (team semantics)

Logics of dependence and independence

Väänänen: Dependence logic

rather than stating dependencies or independencies as annotations of quantifiers, treat dependence as an atomic statement

model-theoretic semantics in terms of sets of assignments (team semantics)

Grädel, Väänänen: Independence logic

based on independence rather than dependence
dependence as a special case of a general form of independence

Logics of dependence and independence

Väänänen: Dependence logic

rather than stating dependencies or independencies as annotations of quantifiers, treat dependence as an atomic statement

model-theoretic semantics in terms of sets of assignments (team semantics)

Grädel, Väänänen: Independence logic

based on independence rather than dependence
dependence as a special case of a general form of independence

Galliani: Logics based on other dependency properties

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 $\overline{x}$.

Thus dependence is understood as functional dependence, which is the strongest form of dependence. The values of $x_1, \ldots x_n$ in X determine the value of y as surely as x and y determine $x + y$ and $x \cdot y$ in elementary arithmetic. Weaker forms of dependence are definable from that.

Armstrong's axioms for dependence

(1) $=(\overline{x}, \overline{x})$
Anything is functionally dependent of itself.

(2) If $=(\overline{x}, \overline{y})$, $\overline{x} \subseteq \overline{x}'$ and $\overline{y}' \subseteq \overline{y}$, then $=(\overline{x}', \overline{y}')$.
Functional dependence is preserved by increasing input data and decreasing output data.

(3) If $=(\overline{x}, \overline{y})$ and $\overline{x}'$ and $\overline{y}'$ are permutations of $\overline{x}$ and $\overline{y}$, then $=(\overline{x}', \overline{y}')$.
Functional dependence does not look at the order of the variables.

(4) If $=(\overline{x}, \overline{y})$ and $=(\overline{y}, \overline{z})$, then $=(\overline{x}, \overline{z})$.
Functional dependences can be transitively composed.

Completeness: If T is a finite set of dependence atoms of the form $=(\overline{x}, \overline{y})$ for various $\overline{x}$ and $\overline{y}$, then $=(\overline{x}, \overline{y})$ follows from T according to Armstrong's rules if, and only if, every team that satisfies T also satisfies $=(\overline{x}, \overline{y})$.

Independence: informal discussion

Independence is a more complicated notion than dependence.

Strongest form of logical independence of x and y:

- Every conceivable pattern of values for (x, y) occurs.
- Knowing one of x and y gives no information about the other.

Caution. Probability theory has its own concept of independence: two random variables are independent if observing one does not affect the probabilities of the other. Logical independence is in harmony with probabilistic independence, but does not pay attention to how often a pattern occurs.

Example for independence

Suppose you want gather experimental data by throwing balls of various size and masses from the Leaning Tower of Pisa to observe how size and mass influence the time of descent. Setting up the experiment you may want to make sure that

The size of the ball is independent from its mass

Example for independence

Suppose you want gather experimental data by throwing balls of various size and masses from the Leaning Tower of Pisa to observe how size and mass influence the time of descent. Setting up the experiment you may want to make sure that

The size of the ball is independent from its mass

How to make sure of this? Vary sizes and masses so that if one size is chosen for one mass it is also chosen for all other masses (and vice versa). This would eliminate any dependence between size and mass and the experiment would genuinely tell us something relevant about the time of descent.

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 $s''(y) = a$.

Constants

Consider a team of data obtained by throwing objects of the same size but different masses from the leaning tower of Pisa.

Galileo: The time of descent is independent of the mass.

Constants

Consider a team of data obtained by throwing objects of the same size but different masses from the leaning tower of Pisa.

Galileo: The time of descent is independent of the mass.

Special theory of relativity:

Einstein: The speed of light is independent of the observer's state of motion.

Constants

Consider a team of data obtained by throwing objects of the same size but different masses from the leaning tower of Pisa.

Galileo: The time of descent is independent of the mass.

Special theory of relativity:

Einstein: The speed of light is independent of the observer's state of motion.

A special case of total independence is when one of the variables is a constant !

Constants

Consider a team of data obtained by throwing objects of the same size but different masses from the leaning tower of Pisa.

Galileo: The time of descent is independent of the mass.

Special theory of relativity:

Einstein: The speed of light is independent of the observer's state of motion.

A special case of total independence is when one of the variables is a constant !

Proposition. A constant variable is independent from every other variable including itself: $=(x) \iff x \perp x$

Axioms for independence

(1) If $x \perp y$ then $y \perp x$ (Symmetry Rule)

(2) If $x \perp x$ then $x \perp y$ (Constancy Rule)

Axioms for independence

(1) If $x \perp y$ then $y \perp x$ (Symmetry Rule)

(2) If $x \perp x$ then $x \perp y$ (Constancy Rule)

Completeness: If T is a finite set of independence atoms of the form $u \perp v$ then $x \perp y$ follows from T via these two rules if, and only if, every team that satisfies T also satisfies $x \perp y$.

Axioms for independence

(1) If $x \perp y$ then $y \perp x$ (Symmetry Rule)

(2) If $x \perp x$ then $x \perp y$ (Constancy Rule)

Completeness: If T is a finite set of independence atoms of the form $u \perp v$ then $x \perp y$ follows from T via these two rules if, and only if, every team that satisfies T also satisfies $x \perp y$.

Proof. Suppose $T \models x \perp y$. Derive $x \perp y$ from T.

Let V be the variables in T and $U = \{z \in V : z \perp z \in T\}$.
If T contains $x \perp y$ or $y \perp x$, or if U contains x or y, we are done.

If not we construct a team $X = X_0 \cup X_1$ that satisfies T, but not $x \perp y$. Set

$$X_\sigma = \left\{ s : V \to (U \cup \{0,1\}) : s(z) = z \text{ for } z \in U, s(x) = s(y) = \sigma \right\}$$

Completeness of simple independence axioms

$X \models T$: Let $(u \perp v) \in T$.

- if u or v is in U then it is constant in X and hence $X \models u \perp v$
- if $\{u, v\} \cap (U \cup \{x, y\}) = \varnothing$, then all values occur for u, v in X and hence $X \models u \perp v$
- otherwise we can assume that $u = x$ and $v \in V \setminus (U \cap \{x, y\})$. For any $s, s' \in X$, let $s''(z) := s(z)$ for $z \neq v$ and $s''(v) := s'(v)$. Then $s'' \in X$ and hence $X \models u \perp v$.

$X \not\models x \perp y$:
There exist $s, s \in X$ with $s(x) = 0$ and $s'(y) = 1$, but there is no $s'' \in X$ with $s''(x) = 0$ and $s''(y) = 1$.

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

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

Dependence versus independence

How do dependence atoms and independence atoms compare with respect to expressive power?

Dependence versus independence

How do dependence atoms and independence atoms compare with respect to expressive power?

Lemma $=(\overline{z}, \overline{x})$ logically implies $\overline{x} \perp_{\overline{z}} \overline{y}$

Dependence versus independence

How do dependence atoms and independence atoms compare with respect to expressive power?

Lemma $=(\overline{z}, \overline{x})$ logically implies $\overline{x} \perp_{\overline{z}} \overline{y}$

Lemma $\overline{x} \perp_{\overline{z}} \overline{y}$ logically implies $=(\overline{z}, \overline{x} \cap \overline{y})$

Dependence versus independence

How do dependence atoms and independence atoms compare with respect to expressive power?

Lemma $=(\overline{z}, \overline{x})$ logically implies $\overline{x} \perp_{\overline{z}} \overline{y}$

Lemma $\overline{x} \perp_{\overline{z}} \overline{y}$ logically implies $=(\overline{z}, \overline{x} \cap \overline{y})$

Corollary $=(\overline{z}, \overline{x}) \iff \overline{x} \perp_{\overline{z}} \overline{x}$

Dependence versus independence

How do dependence atoms and independence atoms compare with respect to expressive power?

Lemma $=(\overline{z}, \overline{x})$ logically implies $\overline{x} \perp_{\overline{z}} \overline{y}$

Lemma $\overline{x} \perp_{\overline{z}} \overline{y}$ logically implies $=(\overline{z}, \overline{x} \cap \overline{y})$

Corollary $=(\overline{z}, \overline{x}) \iff \overline{x} \perp_{\overline{z}} \overline{x}$

So dependence is a special case of (the general form of) independence

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:

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

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:

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

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

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

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

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

Inclusion logic: FO + inclusion atoms $\overline{x} \subseteq \overline{y}$

and so on.

Logics of dependence and independence

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

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

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

Inclusion logic: FO + inclusion atoms $\overline{x} \subseteq \overline{y}$

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

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

- $\mathfrak{A} \models_X \psi \wedge \varphi \quad :\Longleftrightarrow \quad$ 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 $\psi(\overline{y}) \in \mathrm{FO}:\quad \mathfrak{A} \models_X \psi(\overline{y}) \quad\Longleftrightarrow\quad \mathfrak{A} \models_s \psi(\overline{y})$ for all $s \in X$
- $\mathfrak{A} \models_X \psi \wedge \varphi \quad :\Longleftrightarrow\quad$ 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 $\psi(\overline{y}) \in \mathrm{FO}$: $\mathfrak{A} \models_X \psi(\overline{y}) \iff \mathfrak{A} \models_s \psi(\overline{y})$ for all $s \in X$
- $\mathfrak{A} \models_X \psi \wedge \varphi \quad :\iff \quad \mathfrak{A} \models_X \psi$ and $\mathfrak{A} \models_X \varphi$
- $\mathfrak{A} \models_X \psi \vee \varphi \quad :\iff \quad$ 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 $\psi(\overline{y}) \in \mathrm{FO}$: $\quad \mathfrak{A} \models_X \psi(\overline{y}) \quad \Longleftrightarrow \quad \mathfrak{A} \models_s \psi(\overline{y})$ 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 $a \in A$, $\mathfrak{A} \models_{s[y \mapsto a]} \psi$

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 $\psi(\overline{y}) \in \mathrm{FO}$: $\quad \mathfrak{A} \models_X \psi(\overline{y}) \quad \Longleftrightarrow \quad \mathfrak{A} \models_s \psi(\overline{y})$ 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

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

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

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

From team semantics to Tarski semantics

A team X of assignments $s : V \to A$ can be represented as a relation $\mathrm{rel}(X) \subseteq A^{|V|}$.

The translation to Tarski semantics requires that we go to existential second-order logic Σ^1_1.

Proposition. Every formula $\psi(x_1, \dots, x_n)$ in dependence or independence logic, with vocabulary τ, can be translated into a Σ^1_1-sentence ψ^* of vocabulary $\tau \cup \{R\}$ such that

$$\mathfrak{A} \models_X \psi(\overline{x}) \quad \Longleftrightarrow \quad (\mathfrak{A}, \mathrm{rel}(X)) \models \psi^*$$

Indeed this holds for any extension of FO by atomic properties of teams that are first-order expressible (on the relation describing the team).

From team semantics to Tarski semantics

A team X of assignments $s : V \to A$ can be represented as a relation $\mathrm{rel}(X) \subseteq A^{|V|}$.

The translation to Tarski semantics requires that we go to existential second-order logic Σ^1_1.

Proposition. Every formula $\psi(x_1, \ldots, x_n)$ in dependence or independence logic, with vocabulary τ, can be translated into a Σ^1_1-sentence ψ^* of vocabulary $\tau \cup \{R\}$ such that

$$\mathfrak{A} \models_X \psi(\overline{x}) \quad \Longleftrightarrow \quad (\mathfrak{A}, \mathrm{rel}(X)) \models \psi^*$$

Indeed this holds for any extension of FO by atomic properties of teams that are first-order expressible (on the relation describing the team).

For sentences, dependence logic, and thus also independence logic coincides in expressive power with Σ^1_1.

The translation from dependence logic into Σ_1^1

Construct translation $\psi(x_1, \dots, x_n) \quad \mapsto \quad \psi^*(R)$ such that

$$\mathfrak{A} \models_X \psi(\overline{x}) \quad \Longleftrightarrow \quad (\mathfrak{A}, \mathrm{rel}(X)) \models \psi^*$$

The translation from dependence logic into Σ^1_1

Construct translation $\psi(x_1, \dots, x_n) \quad \mapsto \quad \psi^*(R)$ such that

$$\mathfrak{A} \models_X \psi(\overline{x}) \quad \Longleftrightarrow \quad (\mathfrak{A}, \mathrm{rel}(X)) \models \psi^*$$

- first-order literals $\alpha(\overline{x})$ are translated into $\forall \overline{x}(R\overline{x} \to \alpha(\overline{x}))$

The translation from dependence logic into Σ_1^1

Construct translation $\psi(x_1, \ldots, x_n) \quad \mapsto \quad \psi^*(R)$ such that

$$\mathfrak{A} \models_X \psi(\overline{x}) \quad \Longleftrightarrow \quad (\mathfrak{A}, \mathrm{rel}(X)) \models \psi^*$$

- first-order literals $\alpha(\overline{x})$ are translated into $\forall \overline{x}(R\overline{x} \to \alpha(\overline{x}))$
- dependency atoms $=(x_{i_1}, \ldots, x_{i_k}, x_i)$ are translated into

$$\forall \overline{x} \forall \overline{y} \Big(R\overline{x} \land R\overline{y} \land \bigwedge_{j=1}^{k} (x_{i_j} = y_{i_j}) \to (x_i = y_i) \Big)$$

The translation from dependence logic into Σ^1_1

Construct translation $\psi(x_1, \dots, x_n) \quad \mapsto \quad \psi^*(R)$ such that

$$\mathfrak{A} \models_X \psi(\overline{x}) \quad \Longleftrightarrow \quad (\mathfrak{A}, \mathrm{rel}(X)) \models \psi^*$$

- first-order literals $\alpha(\overline{x})$ are translated into $\forall \overline{x}(R\overline{x} \rightarrow \alpha(\overline{x}))$
- dependency atoms $=(x_{i_1}, \dots, x_{i_k}, x_i)$ are translated into

$$\forall \overline{x} \forall \overline{y} \Big(R\overline{x} \wedge R\overline{y} \wedge \bigwedge_{j=1}^{k} (x_{i_j} = y_{i_j}) \rightarrow (x_i = y_i) \Big)$$

- $(\psi(\overline{x}) \wedge \varphi(\overline{x}))^* := \psi^*(R) \wedge \varphi^*(R)$

The translation from dependence logic into Σ_1^1

Construct translation $\psi(x_1, \dots, x_n) \quad \mapsto \quad \psi^*(R)$ such that

$$\mathfrak{A} \models_X \psi(\overline{x}) \quad \Longleftrightarrow \quad (\mathfrak{A}, \mathrm{rel}(X)) \models \psi^*$$

- first-order literals $\alpha(\overline{x})$ are translated into $\forall \overline{x}(R\overline{x} \rightarrow \alpha(\overline{x}))$
- dependency atoms $=\!(x_{i_1}, \dots, x_{i_k}, x_i)$ are translated into

$$\forall \overline{x} \forall \overline{y} \Big(R\overline{x} \wedge R\overline{y} \wedge \bigwedge_{j=1}^{k} (x_{i_j} = y_{i_j}) \rightarrow (x_i = y_i) \Big)$$

- $(\psi(\overline{x}) \wedge \varphi(\overline{x}))^* := \psi^*(R) \wedge \varphi^*(R)$
- $(\psi(\overline{x}) \vee \varphi(\overline{x}))^* := \exists S \exists T (\forall \overline{x}(R\overline{x} \rightarrow (S\overline{x} \vee T\overline{x})) \wedge \psi^*(S) \wedge \varphi^*(T))$

The translation from dependence logic into Σ_1^1

Construct translation $\psi(x_1, \dots, x_n) \quad \mapsto \quad \psi^*(R)$ such that

$$\mathfrak{A} \models_X \psi(\overline{x}) \quad \Longleftrightarrow \quad (\mathfrak{A}, \mathrm{rel}(X)) \models \psi^*$$

- first-order literals $\alpha(\overline{x})$ are translated into $\forall \overline{x}(R\overline{x} \to \alpha(\overline{x}))$
- dependency atoms $=(x_{i_1}, \dots, x_{i_k}, x_i)$ are translated into

$$\forall \overline{x} \forall \overline{y} \Big(R\overline{x} \land R\overline{y} \land \bigwedge_{j=1}^{k} (x_{i_j} = y_{i_j}) \to (x_i = y_i) \Big)$$

- $(\psi(\overline{x}) \land \varphi(\overline{x}))^* := \psi^*(R) \land \varphi^*(R)$
- $(\psi(\overline{x}) \lor \varphi(\overline{x}))^* := \exists S \exists T (\forall \overline{x}(R\overline{x} \to (S\overline{x} \lor T\overline{x})) \land \psi^*(S) \land \varphi^*(T))$
- $(\forall y \, \psi)^* := \exists S (\forall \overline{x} \forall y (R\overline{x} \to S\overline{x}y) \land \psi^*(S))$

The translation from dependence logic into Σ_1^1

Construct translation $\psi(x_1, \dots, x_n) \quad \mapsto \quad \psi^*(R)$ such that

$$\mathfrak{A} \models_X \psi(\overline{x}) \quad \Longleftrightarrow \quad (\mathfrak{A}, \mathrm{rel}(X)) \models \psi^*$$

- first-order literals $\alpha(\overline{x})$ are translated into $\forall \overline{x}(R\overline{x} \to \alpha(\overline{x}))$
- dependency atoms $=(x_{i_1}, \dots, x_{i_k}, x_i)$ are translated into

$$\forall \overline{x} \forall \overline{y} \Big(R\overline{x} \wedge R\overline{y} \wedge \bigwedge_{j=1}^{k} (x_{i_j} = y_{i_j}) \to (x_i = y_i) \Big)$$

- $(\psi(\overline{x}) \wedge \varphi(\overline{x}))^* := \psi^*(R) \wedge \varphi^*(R)$
- $(\psi(\overline{x}) \vee \varphi(\overline{x}))^* := \exists S \exists T (\forall \overline{x}(R\overline{x} \to (S\overline{x} \vee T\overline{x})) \wedge \psi^*(S) \wedge \varphi^*(T))$
- $(\forall y\, \psi)^* := \exists S(\forall \overline{x} \forall y(R\overline{x} \to S\overline{x}y) \wedge \psi^*(S))$
- $(\exists y\, \psi)^* := \exists S(\forall \overline{x} \exists y(R\overline{x} \to S\overline{x}y) \wedge \psi^*(S))$

Expressive power of dependence logic: formulae

Downwards Closure: If $\mathfrak{A} \models_X \psi$ and $Y \subseteq X$, then $\mathfrak{A} \models_Y \psi$

Hence, the sentences ψ^* have to be downwards monotone:
If $(\mathfrak{A}, R) \models \psi^*$ and $S \subseteq R$ then $(\mathfrak{A}, S) \models \psi^*$.

Expressive power of dependence logic: formulae

Downwards Closure: If $\mathfrak{A} \models_X \psi$ and $Y \subseteq X$, then $\mathfrak{A} \models_Y \psi$

Hence, the sentences ψ^* have to be downwards monotone:
If $(\mathfrak{A}, R) \models \psi^*$ and $S \subseteq R$ then $(\mathfrak{A}, S) \models \psi^*$.

Thus dependence logic is a strict fragment of existential second-order logic.

Expressive power of dependence logic: formulae

Downwards Closure: If $\mathfrak{A} \models_X \psi$ and $Y \subseteq X$, then $\mathfrak{A} \models_Y \psi$

Hence, the sentences ψ^* have to be downwards monotone:
If $(\mathfrak{A}, R) \models \psi^*$ and $S \subseteq R$ then $(\mathfrak{A}, S) \models \psi^*$.

Thus dependence logic is a strict fragment of existential second-order logic.

Theorem (Kontinen and Väänänen 2009)
The expressive power of formulae $\psi(x_1, \ldots, x_n)$ of dependence logic is precisely that of existential second-order sentences with the predicate for the team occurring only negatively.

Expressive power of dependence logic: formulae

Downwards Closure: If $\mathfrak{A} \models_X \psi$ and $Y \subseteq X$, then $\mathfrak{A} \models_Y \psi$

Hence, the sentences ψ^* have to be downwards monotone:
If $(\mathfrak{A}, R) \models \psi^*$ and $S \subseteq R$ then $(\mathfrak{A}, S) \models \psi^*$.

Thus dependence logic is a strict fragment of existential second-order logic.

Theorem (Kontinen and Väänänen 2009)
The expressive power of formulae $\psi(x_1, \ldots, x_n)$ of dependence logic is precisely that of existential second-order sentences with the predicate for the team occurring only negatively.

While sentences of dependence logic can express all NP-properties of finite structures, only the downwards closed NP-properties of teams are definable by open formulae of dependence logic.

The translation from independence logic into Σ^1_1

Construct translation $\psi(x_1, \ldots, x_n) \quad \mapsto \quad \psi^*(R)$ such that

$$\mathfrak{A} \models_X \psi(\overline{x}) \quad \Longleftrightarrow \quad (\mathfrak{A}, \mathrm{rel}(X)) \models \psi^*$$

- first-order literals $\alpha(\overline{x})$ are translated into $\forall \overline{x}(R\overline{x} \rightarrow \alpha(\overline{x}))$
- independence atoms $x_i \perp_{x_k} x_j$ are translated into

$$\forall \overline{x} \forall \overline{y}\Big(R\overline{x} \wedge R\overline{y} \wedge (x_k = y_k) \rightarrow \exists \overline{z}\Big(R\overline{z} \wedge (z_k = x_k) \wedge (z_i = x_i) \wedge (z_j = y_j)\Big)\Big)$$

- $(\psi(\overline{x}) \wedge \varphi(\overline{x}))^* := \psi^*(R) \wedge \varphi^*(R)$
- $(\psi(\overline{x}) \vee \varphi(\overline{x}))^* := \exists S \exists T(\forall \overline{x}(R\overline{x} \leftrightarrow (S\overline{x} \vee T\overline{x})) \wedge \psi^*(S) \wedge \varphi^*(T))$
- $(\forall y\, \psi)^* := \exists S(\forall \overline{x} \forall y(R\overline{x} \leftrightarrow S\overline{x}y) \wedge \psi^*(S))$
- $(\exists y\, \psi)^* := \exists S(\forall \overline{x} \exists y(R\overline{x} \leftrightarrow S\overline{x}y) \wedge \psi^*(S))$

Expressive power of independence logic

Contrary to dependence logic, inclusion logic and independence logic are not downwards closed.

For instance $x \perp y$ is not preserved under subteams.

Inclusion logic is closed under union. Independence logic is not.

Expressive power of independence logic

Contrary to dependence logic, inclusion logic and independence logic are not downwards closed.

For instance $x \perp y$ is not preserved under subteams.

Inclusion logic is closed under union. Independence logic is not.

- Exclusion logic $\equiv$ dependence logic.
- Inclusion logic and dependence logic are incomparable.
- FO + inclusion + exclusion $\equiv$ independence logic.
- Independence logic coincides with Σ^1_1 also wrt the expressive power of open formulae.

Corollary. All NP-properties of teams are definable in independence logic.

Part II: Model-Checking Games

- Reachability games and model-checking games for first-order logic
- Second-order reachability games
- Model-checking games for logics with team semantics
- Examples
- Complexity
- Deterministic versus nondeterministic strategies

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 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.

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 $\psi(\overline{x}) \in \text{FO}$.

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

Model-checking game for first-order logic

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

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

Verifier moves:

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

Falsifier moves

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

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:

$(\varphi_1 \vee \varphi_2, s) \to (\varphi_i, s \restriction \text{free}(\varphi_i))$ $(i = 1, 2)$

$(\exists x \varphi, s) \to (\varphi, s[x \mapsto a])$ $(a \in A)$

Falsifier moves

$(\varphi_1 \wedge \varphi_2, s) \to (\varphi_i, s \restriction \text{free}(\varphi_i))$ $(i = 1, 2)$

$(\forall x \varphi, s) \to (\varphi, s[x \mapsto a])$ $(a \in A)$

Terminal positions: φ atomic / negated atomic

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

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: $\text{Win} \subseteq \mathcal{P}(T)$.

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.

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.

Games with imperfect information

There are different views of games with imperfect information.

Explicitly given information sets: Players have only partial information $I(v)$ of the current position v, and may thus, according to their knowledge, be in any position in the information set $I = \{w : I(w) = I(v)\}$. As a consequence, strategies may depend only on current information $I(v)$ and must assign the same action to all nodes in the same information set.

Games with imperfect information

There are different views of games with imperfect information.

Explicitly given information sets: Players have only partial information $I(v)$ of the current position v, and may thus, according to their knowledge, be in any position in the information set $I = \{w : I(w) = I(v)\}$. As a consequence, strategies may depend only on current information $I(v)$ and must assign the same action to all nodes in the same information set.

Explicit restrictions on strategies: Do not make the information explicit, but directly define restrictions on the strategies available for the players. For this purpose it is often useful to identify strategies with the sets of plays that are consistent with them.

Games with imperfect information

There are different views of games with imperfect information.

Explicitly given information sets: Players have only partial information $I(v)$ of the current position v, and may thus, according to their knowledge, be in any position in the information set $I = \{w : I(w) = I(v)\}$. As a consequence, strategies may depend only on current information $I(v)$ and must assign the same action to all nodes in the same information set.

Explicit restrictions on strategies: Do not make the information explicit, but directly define restrictions on the strategies available for the players. For this purpose it is often useful to identify strategies with the sets of plays that are consistent with them.

Here, we use the second approach.

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 $V = \{(\varphi, s) : \varphi$ 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 $\text{Win} \subseteq \mathcal{P}(T)$:

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

$\text{Win} := \{U \subseteq T : \mathfrak{A} \models_{\text{Team}(U,\varphi)} \varphi$ 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 $vF \neq \varnothing$

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

Theorem.

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

The other player

A consistent winning strategy for Falsifier is defined dually, with
$\mathrm{Win}' := \{U \subseteq T : \mathfrak{A} \models_{\mathrm{Team}(U,\varphi)} \neg\varphi$ 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 $\mathrm{Team}(S', \psi) = Y$.

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 $\mathfrak{A} \vDash_X \psi$, 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

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

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

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 $\iff$ $V \cup E \models_{X(G)} \psi(\text{edge,node})$

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 $e = \{u, v\}$.

Independence versus Henkin quantifiers

Independence atoms can be used to describe semantics of Henkin quantifiers.

$$\varphi := \begin{pmatrix} \forall x & \exists y \\ \forall u & \exists v \end{pmatrix} Pxyuv$$

$(A, P) \models \varphi$ if there exist functions $f, g : A \to A$ such that $\mathfrak{A} \models P(a, fa, c, gc)$ for all $a, c \in A$.

Independence versus Henkin quantifiers

Independence atoms can be used to describe semantics of Henkin quantifiers.

$$\varphi := \begin{pmatrix} \forall x & \exists y \\ \forall u & \exists v \end{pmatrix} Pxyuv$$

$(A, P) \models \varphi$ if there exist functions $f, g : A \to A$ such that $\mathfrak{A} \models P(a, fa, c, gc)$ for all $a, c \in A$.

$$\psi := \forall x \exists y \forall u \exists v (xy \perp_u v \wedge Pxyuv)$$

Game-theoretic semantics: $(A, P) \models \psi$ if there exist functions $F : A \to \mathcal{P}(A) \setminus \{\varnothing\}$ and $G : A \times A \times A \to \mathcal{P}(A) \setminus \{\varnothing\}$ such that the team

$$X_{FG} = \{s : (x, y, u, v) \mapsto (a, b, c, d) : \quad b \in F(a) \text{ and } d \in G(a, b, c)\}$$

satisfies $xy \perp_u v$ and $Pxyuv$

Claim. The two formulae are equivalent.

Independence versus Henkin quantifiers

$$\varphi := \begin{pmatrix} \forall x & \exists y \\ \forall u & \exists v \end{pmatrix} Pxyuv \quad \vDash \quad \psi := \forall x \exists y \forall u \exists v (xy \perp_u v \wedge Pxyuv)$$

From Skolem functions f, g for φ we immediately get a consistent winning strategy for ψ, witnessed by $F(a) := \{fa\}$ and $G(a, b, c) := \{gc\}$.

The team X_{FG} then consists of all assignments (a, fa, c, gc) which clearly satisfies $Pxyuv$ and also satisfies the independence atom $xy \perp_u v$ since v is constant for any fixed value for u.

Independence versus Henkin quantifiers

$$\psi := \forall x \exists y \forall u \exists v (xy \perp_u v \land Pxyuv) \quad \vDash \quad \varphi := \begin{pmatrix} \forall x & \exists y \\ \forall u & \exists v \end{pmatrix} Pxyuv$$

Given a consistent winning strategy for ψ on $\mathfrak{A}$, witnessed by F and G, define Skolem functions for φ via a choice function ε on $\mathcal{P}(A)$.
Set $fa := \varepsilon F(a)$ and $gc := \varepsilon(\bigcup_{a \in A} G(a, fa, c))$.

Claim. $P(a, fa, c, gc)$ for all $a, c \in A$.

Prove that, for all a, c, the assignment (a, fa, c, gc) belongs to X_{FG}.

(1) Given a, c, there exists $a' \in A$ such that $gc \in G(a', fa', c)$; hence $(a', fa', c, gc) \in X_{FG}$.

(2) $(a, fa, c, d) \in X_{FG}$ for all $d \in G(a, fa, c)$.

(3) By the independence atom we infer that X_{FG} also contains (a, fa, c, gc).

Kernels in directed graphs

Represent a directed graph by a team E with attributes (source,target).

Kernel of G: set K of nodes such that no edges go from K to K and every node outside K is dominated by a node in K.

Existence of a kernel is an NP-complete problem.
Kernel is not downwards closed, hence not expressible in dependence logic.

A formula with dependence, exclusion, and inclusion:

$$\psi(\text{source,target}) = \exists c \forall x \exists y \forall x' \exists y' \big({=}(c) \wedge {=}(x,y) \wedge {=}(x'y') \wedge$$
$$(x = x' \to y = y') \wedge (\varphi \vee \vartheta), \quad \text{where}$$
$$\varphi := (x = \text{source} \to (y = c \wedge (\text{source} \mid \text{target}) \wedge (y' = c \vee x' \subseteq \text{target})))$$
$$\vartheta := (x = \text{source} \to y \neq c)$$

Claim. $V \models_E \psi(\text{source,target}) \iff (V, E)$ has a kernel

Kernels in directed graphs

$$\psi(\text{source,target}) = \exists c \forall x \exists y \forall x' \exists y' \big(=(c) \wedge =(x,y) \wedge =(x'y') \wedge$$
$$(x = x' \rightarrow y = y') \wedge (\varphi \vee \vartheta) \big)$$

There is constant c and two functions $f : x \mapsto y$ and $f' : x' \mapsto y'$, which are actually the same, such that the resulting team can be split into two subsets one of which satisfies φ and the other satisfies ϑ.

Idea: $K = f^{-1}(c)$, and E split into the team E_1 of edges originating in K and the team E_2 of edges originating in its complement.

$$\varphi := (x = \text{source} \rightarrow (y = c \wedge (\text{source} \mid \text{target}) \wedge (y' = c \vee x' \subset \text{target})))$$
$$\vartheta := (x = \text{source} \rightarrow y \neq c)$$

No edge in E_1 has its target in K, and that all nodes outside K are the target of an edge in E_1.

Exclusion is expressible with simple independence

Lemma. (Galliani) $x \mid y \equiv \exists z (x \subseteq z \wedge y \neq z \wedge y \perp z)$

Suppose that $A \models_X x \mid y$. Then $B := \{s(y) : s \in X\} \subsetneq A$. We define a consistent winning strategy S showing that $A \models_X \exists z (x \subseteq z \wedge y \neq x \wedge y \perp z)$ by permiting, for all $s \in X$, all values in $A \setminus B$ for z.

Exclusion is expressible with simple independence

Lemma. (Galliani) $x \mid y \equiv \exists z(x \subseteq z \wedge y \neq z \wedge y \perp z)$

Suppose that $A \vDash_X x \mid y$. Then $B := \{s(y) : s \in X\} \subsetneq A$. We define a consistent winning strategy S showing that $A \vDash_X \exists z(x \subseteq z \wedge y \neq x \wedge y \perp z)$ by permiting, for all $s \in X$, all values in $A \setminus B$ for z.

- $\text{Team}(S, x \subseteq z) = C \times (A \setminus B)$ for some C with $C \cap B = \emptyset$. This obviously satisfies $x \subseteq z$.
- $\text{Team}(S, y \neq z) = \text{Team}(S, y \perp z) = B \times (A \setminus B)$.
  This obviously satisfies both $y \neq z$ and $y \perp z$.

Exclusion is expressible with simple independence

Lemma. (Galliani) $x \mid y \equiv \exists z(x \subseteq z \wedge y \neq z \wedge y \perp z)$

Conversely suppose that S is a consistent winning strategy witnessing that $A \models_X \exists z(x \subseteq z \wedge y \neq z \wedge y \perp z)$. Towards a contradiction, assume that some $a \in A$ occurs as value for both x and y in X.

Exclusion is expressible with simple independence

Lemma. (Galliani) $\quad x \mid y \equiv \exists z(x \subseteq z \wedge y \neq z \wedge y \perp z)$

Conversely suppose that S is a consistent winning strategy witnessing that $A \models_X \exists z(x \subseteq z \wedge y \neq z \wedge y \perp z)$. Towards a contradiction, assume that some $a \in A$ occurs as value for both x and y in X.

Let $T = \text{Team}(S, (x \subseteq z \wedge y \neq z \wedge y \perp z))$.

- By consistency with $x \subseteq z$, T contains an assignment s with $s(z) = a$.
- Since a occurs as a value for y, T has an assignment s' with $s'(y) = a$.
- By consistency with $y \perp z$ there must be a further assignent s'' with $s''(y) = a$ and $s''(z) = a$.
- This contradicts consistency with $y \neq z$.

Inclusion is expressible with simple independence

Lemma. (Galliani) $x \subseteq y \quad \equiv \quad \forall z \forall u(\varphi_1 \vee \varphi_2 \vee \varphi_3)$ where

- $\varphi_1 := (z \neq x \wedge z \neq y) \vee (u \neq 0 \wedge u \neq 1)$
- $\varphi_2 := (u = 0 \wedge z \neq y)$
- $\varphi_3 := ((u = 0 \wedge z = y) \vee u = 1) \wedge z \perp u)$

Let X be a team on A (containing two distinct elements 0,1) and let $Y = X[z \mapsto A][u \mapsto A]$. We have to show that $A \vDash_X x \subseteq y$ if, and only if, Player 0 can find a decomposition $Y = Y_1 \cup Y_2 \cup Y_3$ such that $A \vDash_{Y_i} \varphi_i$ (for $i = 1, 2, 3$).

Suppose that $A \vDash_X (x \subseteq y)$. We can satisfy the first two formulae by

$Y_1 := \{s \in Y : s(u) \in A \setminus \{0,1\}\} \cup \{s \in Y : s(z) \notin \{s(x), s(y)\}\}$

$Y_2 := \{s \in Y : s(u) = 0 \wedge s(z) \neq s(y)\}$

This leaves us with

$Y_3 = \{s \in Y : s(z) = s(y) \wedge s(u) \in \{0,1\}\} \cup \{s \in Y : s(u) = 1 \wedge s(z) = s(x)\}$

Inclusion is expressible with simple independence

For $Y_3 = \{s \in Y : s(z) = s(y) \land s(u) \in \{0,1\}\} \cup \{s \in Y : s(u) = 1 \land s(z) = s(x)\}$ we have to show that

$$A \models_{Y_3} ((u = 0 \land z = y) \lor u = 1) \land z \perp u)$$

Inclusion is expressible with simple independence

For $Y_3 = \{s \in Y : s(z) = s(y) \wedge s(u) \in \{0,1\}\} \cup \{s \in Y : s(u) = 1 \wedge s(z) = s(x)\}$ we have to show that

$$A \models_{Y_3} ((u = 0 \wedge z = y) \vee u = 1) \wedge z \perp u)$$

Clearly the first conjunct is true. It remains to show that $Y_3 \models (z \perp u)$, i.e.,

$$(\forall s, s' \in Y_3)(\exists s'' \in Y_3)(s''(z) = s(z) \wedge s''(u) = s'(u))$$

Inclusion is expressible with simple independence

For $Y_3 = \{s \in Y : s(z) = s(y) \wedge s(u) \in \{0,1\}\} \cup \{s \in Y : s(u) = 1 \wedge s(z) = s(x)\}$
we have to show that

$$A \models_{Y_3} ((u = 0 \wedge z = y) \vee u = 1) \wedge z \perp u)$$

Clearly the first conjunct is true. It remains to show that $Y_3 \models (z \perp u)$, i.e.,

$$(\forall s, s' \in Y_3)(\exists s'' \in Y_3)(s''(z) = s(z) \wedge s''(u) = s'(u))$$

- if $s(z) = s(y)$ let $s'' := s[u \mapsto s'(u)]$.
  $s'' \in Y_3$ since $s''(z) = s(z) = s(y) = s''(y)$

Inclusion is expressible with simple independence

For $Y_3 = \{s \in Y : s(z) = s(y) \wedge s(u) \in \{0,1\}\} \cup \{s \in Y : s(u) = 1 \wedge s(z) = s(x)\}$
we have to show that

$$A \models_{Y_3} ((u = 0 \wedge z = y) \vee u = 1) \wedge z \perp u)$$

Clearly the first conjunct is true. It remains to show that $Y_3 \models (z \perp u)$, i.e.,

$$(\forall s, s' \in Y_3)(\exists s'' \in Y_3)(s''(z) = s(z) \wedge s''(u) = s'(u))$$

- if $s(z) = s(y)$ let $s'' := s[u \mapsto s'(u)]$.
  $s'' \in Y_3$ since $s''(z) = s(z) = s(y) = s''(y)$
- if $s(z) \neq s(y)$, then $s(u) = 1$ and $s(z) = s(x)$.
  Since $X \models (x \subseteq y)$ there exists $t \in X$ with $t(y) = s(x)$.
  Set $s'' := t[z \mapsto s(z)][u \mapsto s'(u)]$.
  Again $s'' \in Y_3$ since $s''(z) = s(z) = s(x) = t(y) = s''(y)$.

Inclusion is expressible with simple independence

Lemma. (Galliani) $x \subseteq y \quad \equiv \quad \forall z \forall u(\varphi_1 \vee \varphi_2 \vee \varphi_3)$ where

- $\varphi_1 := (z \neq x \wedge z \neq y) \vee (u \neq 0 \wedge u \neq 1)$
- $\varphi_2 := (u = 0 \wedge z \neq y)$
- $\varphi_3 := ((u = 0 \wedge z = y) \vee u = 1) \wedge z \perp u)$

Inclusion is expressible with simple independence

Lemma. (Galliani) $x \subseteq y \quad \equiv \quad \forall z \forall u(\varphi_1 \vee \varphi_2 \vee \varphi_3)$ where

- $\varphi_1 := (z \neq x \wedge z \neq y) \vee (u \neq 0 \wedge u \neq 1)$
- $\varphi_2 := (u = 0 \wedge z \neq y)$
- $\varphi_3 := ((u = 0 \wedge z = y) \vee u = 1) \wedge z \perp u)$

For the converse, assume Player 0 can find a decomposition $Y = Y_1 \cup Y_2 \cup Y_3$ of $Y = X[z \mapsto A][u \mapsto A]$ such that $A \vDash_{Y_i} \varphi_i$ (for $i = 1, 2, 3$).

Inclusion is expressible with simple independence

Lemma. (Galliani) $\quad x \subseteq y \quad \equiv \quad \forall z \forall u(\varphi_1 \vee \varphi_2 \vee \varphi_3)$ where

- $\varphi_1 := (z \neq x \wedge z \neq y) \vee (u \neq 0 \wedge u \neq 1)$
- $\varphi_2 := (u = 0 \wedge z \neq y)$
- $\varphi_3 := ((u = 0 \wedge z = y) \vee u = 1) \wedge z \perp u)$

For the converse, assume Player 0 can find a decomposition $Y = Y_1 \cup Y_2 \cup Y_3$ of $Y = X[z \mapsto A][u \mapsto A]$ such that $A \vDash_{Y_i} \varphi_i$ (for $i = 1, 2, 3$).

For any $t \in X$, let $s = t[z \mapsto t(x)][u \mapsto 1]$ and $s' = t[z \mapsto t(y)][u \mapsto 0]$. Any team containing s or s' violates φ_1 and φ_2, so $s, s' \in Y_3$.

Since $Y_3 \vDash z \perp u$ there exists some $s'' \in Y_3$ with $s''(z) = s(z) = t(x)$ and $s''(u) = s'(u) = 0$. Since $s''(u) = 0$ it follows that $s''(y) = s''(z) = t(x)$. But this means that $t(x)$ occurs in X as a value for y.

Inclusion is expressible with simple independence

Lemma. (Galliani) $x \subseteq y \quad \equiv \quad \forall z \forall u(\varphi_1 \vee \varphi_2 \vee \varphi_3)$ where

- $\varphi_1 := (z \neq x \wedge z \neq y) \vee (u \neq 0 \wedge u \neq 1)$
- $\varphi_2 := (u = 0 \wedge z \neq y)$
- $\varphi_3 := ((u = 0 \wedge z = y) \vee u = 1) \wedge z \perp u)$

For the converse, assume Player 0 can find a decomposition $Y = Y_1 \cup Y_2 \cup Y_3$ of $Y = X[z \mapsto A][u \mapsto A]$ such that $A \models_{Y_i} \varphi_i$ (for $i = 1, 2, 3$).

For any $t \in X$, let $s = t[z \mapsto t(x)][u \mapsto 1]$ and $s' = t[z \mapsto t(y)][u \mapsto 0]$. Any team containing s or s' violates φ_1 and φ_2, so $s, s' \in Y_3$.

Since $Y_3 \models z \perp u$ there exists some $s'' \in Y_3$ with $s''(z) = s(z) = t(x)$ and $s''(u) = s'(u) = 0$. Since $s''(u) = 0$ it follows that $s''(y) = s''(z) = t(x)$. But this means that $t(x)$ occurs in X as a value for y.

We have proved that $X \models (x \subseteq y)$.

Simple independence atoms suffice

Recall that Galliani proved that independence logic has the same expressive power as FO with inclusion and exclusion atoms. Since these are expressible via simple independence atoms, the generalized independence atoms are not really necessary.

Corollary (Galliani)

$$\text{Independence logic} \quad \equiv \quad (\text{FO} + \text{simple independence atoms})$$

Deterministic versus nondeterministic strategies

Our notion of constent 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 $\exists x(y \subseteq x \land z \subseteq x)$ 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.

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

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 eliminate 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 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 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.

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 $[\![\varphi \land \psi]\!]^{\mathfrak{A}}$ and $[\![\varphi \lor \psi]\!]^{\mathfrak{A}}$ (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 Väänňanen) 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 Väänñanen) 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 $X \cap \{s_a\} = \{s_a\} \neq \varnothing$.

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. $\psi \vDash \eta$ and $\varphi \vDash \neg\eta$.

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

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 $\mathfrak{A} \models_X \neg\eta$.

Part III: Least Fixed-Point Logic, Inclusion Logic, and the Quest for a Logic for PTIME

- The quest for a logic for polynomial time
- State of the art and challenges for current research
- Structure of least fixed point logic
- Inclusion logic versus least fixed-point logic

The most important problem of Finite Model Theory

Is there a logic that captures PTIME?

The most important problem of Finite Model Theory

Is there a logic that captures PTIME?

Informal definition: A logic L captures PTIME if it defines precisely those properties of finite structures that are decidable in polynomial time:

(1) For every sentence $\psi \in L$, the set of finite models of ψ is decidable in polynomial time.

(2) For every PTIME-property S of finite τ-structures, there is a sentence $\psi \in L$ such that $S = \{\mathfrak{A} \in \mathrm{Fin}(\tau) : \mathfrak{A} \models \psi\}$.

The most important problem of Finite Model Theory

Is there a logic that captures PTIME?

Informal definition: A logic L captures PTIME if it defines precisely those properties of finite structures that are decidable in polynomial time:

(1) For every sentence $\psi \in L$, the set of finite models of ψ is decidable in polynomial time.

(2) For every PTIME-property S of finite τ-structures, there is a sentence $\psi \in L$ such that $S = \{\mathfrak{A} \in \mathrm{Fin}(\tau) : \mathfrak{A} \models \psi\}$.

The precise definition is more subtle. It includes certain effectiveness requirements to exclude pathological 'solutions'.

First-Order Logic

First-order logic (FO) is far too weak to capture PTIME.

- FO can express only local properties of finite structures
  Theorems of Gaifman and Hanf

  Global properties (e.g. planarity of graphs) are not expressible.

- FO has no mechanism for recursion or unbounded iteration.

  Transitive closures, reachability or termination properties, winning regions in games, etc. are not FO-definable.

- FO can only express properties in AC^0

  AC^0 is constant parallel time with polynomial hardware. In particular, FO $\subseteq$ LOGSPACE.

Second-Order Logic

Second-order logic (SO) is (probably) too strong to capture PTIME.

Fagin's Theorem. Existential SO captures NP.

Corollary. SO captures the polynomial hierarchy.

Thus SO captures polynomial time if, and only if, P = NP.

Second-Order Logic

Second-order logic (SO) is (probably) too strong to capture PTIME.

Fagin's Theorem. Existential SO captures NP.

Corollary. SO captures the polynomial hierarchy.

Thus SO captures polynomial time if, and only if, P = NP.

Monadic second-order logic is orthogonal to PTIME:

On words, MSO captures the regular languages, and not all PTIME-languages are regular.

On graphs, MSO can express NP-complete properties, such as 3-colourability.

Least fixed point logic LFP

Syntax. LFP extends FO by fixed point rule:

- For every formula $\psi(T, x_1 \dots x_k) \in \text{LFP}[\tau \cup \{T\}]$,
  T k-ary relation variable, occuring only positive in ψ,
  build formulae $[\mathbf{lfp}\, T\overline{x}\,.\,\psi](\overline{x})$ and $[\mathbf{gfp}\, T\overline{x}\,.\,\psi](\overline{x})$

Semantics. On τ-structure $\mathfrak{A}$, $\psi(T, \overline{x})$ defines monotone operator

$$\begin{aligned} \psi^{\mathfrak{A}} : \mathcal{P}(A^k) &\longrightarrow \mathcal{P}(A^k) \\ T &\longmapsto \{\overline{a} : (\mathfrak{A}, T) \models \psi(T, \overline{a})\} \end{aligned}$$

- $\mathfrak{A} \models [\mathbf{lfp}\, T\overline{x}\,.\,\psi(T, \overline{x})](\overline{a}) \;\;:\Longleftrightarrow\; \overline{a} \in \mathbf{lfp}(\psi^{\mathfrak{A}})$
  $\mathfrak{A} \models [\mathbf{gfp}\, T\overline{x}\,.\,\psi(T, \overline{x})](\overline{a}) \;\;:\Longleftrightarrow\; \overline{a} \in \mathbf{gfp}(\psi^{\mathfrak{A}})$

LFP and polynomial time

Theorem (Immerman, Vardi)
On ordered finite structures, LFP captures PTIME.

On arbitrary finite structures, LFP can express certain PTIME-complete problems (such as winning regions of reachability games), but fails to express all of PTIME.

LFP and polynomial time

Theorem (Immerman, Vardi)
On ordered finite structures, LFP captures PTIME.

On arbitrary finite structures, LFP can express certain PTIME-complete problems (such as winning regions of reachability games), but fails to express all of PTIME.

LFP is unable to count.

For instance the class of graphs with an even number of vertices is not LFP-definable.

LFP and polynomial time

Theorem (Immerman, Vardi)
On ordered finite structures, LFP captures PTIME.

On arbitrary finite structures, LFP can express certain PTIME-complete problems (such as winning regions of reachability games), but fails to express all of PTIME.

LFP is unable to count.

For instance the class of graphs with an even number of vertices is not LFP-definable.

Immerman suggested to extend fixed-point logics by a counting mechanism.

Fixed-point logic with counting

(FP + C): A two-sorted fixed-point logic with counting terms.

Two sorts of variables:

- $x, y, z, \dots$ ranging over the domain of the given finite structure
- $\mu, \nu, \dots$ ranging over natural numbers

On natural numbers, standard arithmetic operations $+$, $\cdot$ and $<$ are available, but variables must be explicitely bounded and only take polynomially bounded values.

Counting terms: For a formula $\varphi(x)$, the term $\#_x\varphi(x)$ denotes the number of elements a of the structure that satisfy $\varphi(a)$.

Least or inflationary fixed point operator defining formulae of the form

$$[\mathbf{fp}\, R\overline{x}\overline{\mu}_{\leq\bar{t}}\,.\,\psi(R,\overline{x},\overline{\mu})](\overline{y},\overline{\nu}).$$

Infinitary logic with counting

(FP + C) can be embedded into the infinitary logic $C^{\omega}_{\infty\omega}$, which extends first-order logic by allowing

- counting quantifiers $\exists^i x$: there exist at least i elements x such that…
- infinitary conjunctions and disjunctions:
  $\bigvee \Phi$ and $\bigwedge \Phi$ for any set Φ of formulae
- but only finitely many distinct variables in each formula.

Infinitary logic with counting

(FP + C) can be embedded into the infinitary logic $C^{\omega}_{\infty\omega}$, which extends first-order logic by allowing

- counting quantifiers $\exists^i x$: there exist at least i elements x such that…
- infinitary conjunctions and disjunctions:
  $\bigvee \Phi$ and $\bigwedge \Phi$ for any set Φ of formulae
- but only finitely many distinct variables in each formula.

$C^{k}_{\infty\omega}$ is the k-variable-fragment of $C^{\omega}_{\infty\omega}$

Infinitary logic with counting

(FP + C) can be embedded into the infinitary logic $C^{\omega}_{\infty\omega}$, which extends first-order logic by allowing

- counting quantifiers $\exists^{i} x$: there exist at least i elements x such that…
- infinitary conjunctions and disjunctions:
  $\bigvee \Phi$ and $\bigwedge \Phi$ for any set Φ of formulae
- but only finitely many distinct variables in each formula.

$C^{k}_{\infty\omega}$ is the k-variable-fragment of $C^{\omega}_{\infty\omega}$

Why are we interested in this infinitary logic?

Infinitary logic with counting

(FP + C) can be embedded into the infinitary logic $C^{\omega}_{\infty\omega}$, which extends first-order logic by allowing

- counting quantifiers $\exists^{i} x$: there exist at least i elements x such that…
- infinitary conjunctions and disjunctions:
  $\bigvee \Phi$ and $\bigwedge \Phi$ for any set Φ of formulae
- but only finitely many distinct variables in each formula.

$C^{k}_{\infty\omega}$ is the k-variable-fragment of $C^{\omega}_{\infty\omega}$

Why are we interested in this infinitary logic?

$C^{k}_{\infty\omega}$-equivalence of finite structures, and hence non-expressibility results for (FP+C) can be proved via appropriate variants of Ehrenfeucht-Fraïssé games, as for instance Hella's k-pebble bijection games.

Fixed-point logic with counting versus polynomial time

Theorem. (FP+C) $\subsetneq$ Ptime (Cai, Fürer, Immerman 1992)

Fixed-point logic with counting versus polynomial time

Theorem. (FP+C) $\subsetneq$ PTIME (Cai, Fürer, Immerman 1992)

It is easy to see that every (FP+C)-definable property of finite structures is decidable in polynomial time.

On the other side, Cai, Fürer, and Immerman constructed sequences $(G_n)_{n\in\omega}$ and $(H_n)_{n\in\omega}$ of graphs such that

(1) There is a class of graphs, that is decidable in polynomial time, which includes all G_n and excludes all H_n

(2) $G_n \equiv^{C^n_{\infty\omega}} H_n$, for all n.

Fixed-point logic with counting versus polynomial time

Theorem. (FP+C) $\subsetneq$ PTIME (Cai, Fürer, Immerman 1992)

It is easy to see that every (FP+C)-definable property of finite structures is decidable in polynomial time.

On the other side, Cai, Fürer, and Immerman constructed sequences $(G_n)_{n\in\omega}$ and $(H_n)_{n\in\omega}$ of graphs such that

(1) There is a class of graphs, that is decidable in polynomial time, which includes all G_n and excludes all H_n

(2) $G_n \equiv^{C^n_{\infty\omega}} H_n$, for all n.

Although the Cai-Fürer-Immerman construction is sophisticated, the property separating the G_n from the H_n seemed somewhat artificial.
Might it be the case that (FP+C) captures all "natural" PTIME-properties of finite structures?

Capturing polynomial time sometimes

Fixed-point logic with counting captures PTIME on certain interesting classes of structures, such as

- trees (Immerman, Lander)
- planar graphs and graphs of bounded genus (Grohe)
- structures of bounded tree-width (Grohe, Marino)
- chordal line graphs (Grohe)
- interval graphs (Laubner)
- all classes of graphs that exclude a minor (Grohe)

Further (FP+C) captures PTIME almost everywhere, i.e. on a class of random structures with asymptotic probability one. (Hella, Kolaitis, Luosto)

Formulae that define matrices

A formula $\varphi(x, y)$ defines, when evaluated on a finite structure $\mathfrak{A}$, a square matrix $M_\varphi^{\mathfrak{A}} = (m_{a,b})_{a,b\in\mathfrak{A}}$, with entries

$$m_{a,b} := \begin{cases} 1 & \text{if } \mathfrak{A} \models \varphi(a,b) \\ 0 & \text{if } \mathfrak{A} \not\models \varphi(a,b) \end{cases}$$

Also more general matrices, for instance with entries in an arbitrary finite commutative ring, can be defined by appropriate (collections of) formulae.

Since we assume our structures to be unordered, these matrices are defined only up to permutations of the rows and columns.

The next frontier: linear algebra

Most concepts in linear algebra can be formulated in terms of matrices.

We are interested in matrix properties and functions on matrices that are

(1) invariant under permutations of the rows and columns

(2) computable in polynomial time

This includes arithmetic operations on matrices, singularity, rank and determinant, characteristic polynomials, minimal polynomials, solvablity of linear equation systems, normal forms, …

Question: Which of these properties and operations are definable in (FP+C)?

The answer may depend on the underlying ring.
One may consider $\mathbb{Q}$, $\mathbb{Z}$, finite fields, or arbitrary finite commutative rings.

Linear algebra in (FP+C)

Actually a fair amount of linear algebra can be defined in (FP + C):

- matrix addition and matrix multiplication
- matrix exponentiation M^k (with k given in binary notation)
- (non-)singularity of matrices
- determinant (over finite fields, $\mathbb{Q}$, and $\mathbb{Z}$)
- characteristic polynomials
- minimal polynomials (on fields)
- matrix rank over $\mathbb{Q}$

(Blass, Gurevich, Shelah), (Dawar, Grohe, Holm, Laubner), (Grädel, Pakusa)

Linear algebra outside (FP+C)

However, there are some fundamental polynomial-time properties in linear algebra that are not definable in (FP+C)

- solvability of linear equation systems, over any finite commutative ring
- the rank of a matrix (over a finite field)
- similarity of matrices (over a finite field)
- rank equality (over a finite field)

Linear algebra outside (FP+C)

However, there are some fundamental polynomial-time properties in linear algebra that are not definable in (FP+C)

- solvability of linear equation systems, over any finite commutative ring
- the rank of a matrix (over a finite field)
- similarity of matrices (over a finite field)
- rank equality (over a finite field)

These are also sources of new operators to extend the logics.

Current research: Fixed-point logics with rank operators, or with solvability operators for linear equation systems. Expressive power, separation from PTIME. Choiceless computation.

Least and greatest fixed points

$[\mathbf{lfp}\, T\overline{x}\,.\,\varphi(T,\overline{x})](\overline{a})$: $\overline{a}$ contained in smallest T with $T = \{\overline{x} : \varphi(T,\overline{x})\}$

$[\mathbf{gfp}\, T\overline{x}\,.\,\varphi(S,\overline{x})](\overline{a})$: $\overline{a}$ contained in greatest T with $T = \{\overline{x} : \varphi(T,\overline{x})\}$

Least and greatest fixed points

$[\mathbf{lfp}\, T\overline{x}\,.\,\varphi(T,\overline{x})](\overline{a})$: $\overline{a}$ contained in smallest T with $T = \{\overline{x} : \varphi(T,\overline{x})\}$

$[\mathbf{gfp}\, T\overline{x}\,.\,\varphi(S,\overline{x})](\overline{a})$: $\overline{a}$ contained in greatest T with $T = \{\overline{x} : \varphi(T,\overline{x})\}$

these T exist if $F_\varphi : T \mapsto \{\overline{x} : \varphi(T,\overline{x})\}$ is monotone (preserves $\subseteq$)

to guarantee monotonicity: require that T positive in φ

Least and greatest fixed points

$[\mathbf{lfp}\, T\bar{x}\,.\,\varphi(T,\bar{x})](\bar{a})$: $\bar{a}$ contained in smallest T with $T = \{\bar{x} : \varphi(T,\bar{x})\}$
$[\mathbf{gfp}\, T\bar{x}\,.\,\varphi(S,\bar{x})](\bar{a})$: $\bar{a}$ contained in greatest T with $T = \{\bar{x} : \varphi(T,\bar{x})\}$

these T exist if $F_\varphi : T \mapsto \{\bar{x} : \varphi(T,\bar{x})\}$ is monotone (preserves $\subseteq$)
to guarantee monotonicity: require that T positive in φ

Inductive construction of the least and greatest fixed points on a structure $\mathfrak{A}$:

$$T^0 := \varnothing \qquad S^0 := A^k$$
$$T^{\alpha+1} := F_\varphi(T^\alpha) \qquad S^{\alpha+1} := F_\varphi(S^\alpha)$$
$$T^\lambda := \bigcup_{\alpha<\lambda} T^\alpha \qquad S^\lambda := \bigcap_{\alpha<\lambda} S^\alpha \qquad (\lambda \text{ limit ordinal})$$

$\Longrightarrow$ increasing/decreasing sequence of stages $(T^\alpha \subseteq T^{\alpha+1}, S^\alpha \supseteq S^{\alpha+1})$,
converges to fixed points T^∞ and S^∞ of F_φ

Least and greatest fixed points

$[\mathbf{lfp}\, T\overline{x}\,.\,\varphi(T,\overline{x})](\overline{a})$: $\overline{a}$ contained in smallest T with $T = \{\overline{x} : \varphi(T,\overline{x})\}$
$[\mathbf{gfp}\, T\overline{x}\,.\,\varphi(S,\overline{x})](\overline{a})$: $\overline{a}$ contained in greatest T with $T = \{\overline{x} : \varphi(T,\overline{x})\}$

these T exist if $F_\varphi : T \mapsto \{\overline{x} : \varphi(T,\overline{x})\}$ is monotone (preserves $\subseteq$)
to guarantee monotonicity: require that T positive in φ

Inductive construction of the least and greatest fixed points on a structure $\mathfrak{A}$:

$$T^0 := \varnothing \qquad S^0 := A^k$$
$$T^{\alpha+1} := F_\varphi(T^\alpha) \qquad S^{\alpha+1} := F_\varphi(S^\alpha)$$
$$T^\lambda := \bigcup_{\alpha<\lambda} T^\alpha \qquad S^\lambda := \bigcap_{\alpha<\lambda} S^\alpha \qquad (\lambda \text{ limit ordinal})$$

$\Longrightarrow$ increasing/decreasing sequence of stages $(T^\alpha \subseteq T^{\alpha+1}, S^\alpha \supseteq S^{\alpha+1})$,
converges to fixed points T^∞ and S^∞ of F_φ

Theorem: $T^\infty = \mathbf{lfp}(F_\varphi)$ and $S^\infty = \mathbf{gfp}(F_\varphi)$ (Knaster, Tarski)

LFP-definability for reachability and safety games

GAME is definable in LFP

Reachability: Player 0 has winning strategy for game $\mathcal{G}$ from position v

$$\iff$$

$$\mathcal{G} = (V, V_0, V_1, E) \models [\mathbf{lfp}\, Wx\,.\,(V_0x \land \exists y(Exy \land Wy)) \lor (V_1x \land \forall y(Exy \to Wy)](v)$$

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

$$\iff$$

$$\mathcal{G} = (V, V_0, V_1, E) \models [\mathbf{gfp}\, Wx\,.\,(V_0x \land \exists y(Exy \land Wy)) \lor (V_1x \land \forall y(Exy \to Wy)](v)$$

LFP-definability for reachability and safety games

GAME is definable in LFP

Reachability: Player 0 has winning strategy for game $\mathcal{G}$ from position v

$$\Longleftrightarrow$$

$$\mathcal{G} = (V, V_0, V_1, E) \models [\mathbf{lfp}\, Wx\,.\,(V_0 x \land \exists y(Exy \land Wy)) \lor (V_1 x \land \forall y(Exy \to Wy)](v)$$

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

$$\Longleftrightarrow$$

$$\mathcal{G} = (V, V_0, V_1, E) \models [\mathbf{gfp}\, Wx\,.\,(V_0 x \land \exists y(Exy \land Wy)) \lor (V_1 x \land \forall y(Exy \to Wy)](v)$$

GAME is complete for LFP (via quantifier-free reductions on finite structures)

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 point.

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 point.

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 $\equiv$ posGFP.

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

posGFP versus existential second-order logic

Proposition. (Knaster, Tarski) The greastet fixed point of a monotone operator is the union of all its post-fixed-points:
$\mathbf{gfp}(F) = \bigcup\{R : F(R) = R\} = \bigcup\{R : F(R) \supseteq R\}$

As a consequence, it immediately follows that posGFP is a fragment of Σ^1_1:

$$[\mathbf{gfp}\, R\overline{x} \,.\, \varphi(R\overline{x})](\overline{z}) \equiv \exists R(R\overline{z} \land \forall \overline{x}(R\overline{x} \to \varphi(R, \overline{x})))$$

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 $\mathfrak{A} \models_{X \cup Y} \psi$

For formulae $\overline{x} \subseteq \overline{y}$ and for first-order formulae this is obvious.

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 $\mathfrak{A} \models_{X \cup Y} \psi$

For formulae $\overline{x} \subseteq \overline{y}$ and for first-order formulae this is obvious.

Closure under union is obviously preserved by $\wedge$ and $\forall$.

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 $\mathfrak{A} \models_{X \cup Y} \psi$

For formulae $\overline{x} \subseteq \overline{y}$ and for first-order formulae this is obvious.

Closure under union is obviously preserved by $\wedge$ and $\forall$.

Disjunction: assume that $\mathfrak{A} \models_X \psi \vee \varphi$ and $\mathfrak{A} \models_Y \psi \vee \varphi$.
Then $X = X_\psi \cup X_\varphi$ such that $\mathfrak{A} \models_{X_\psi} \psi$ and $\mathfrak{A} \models_{X_\varphi} \varphi$.
Similarly $Y = Y_\psi \cup Y_\varphi$ such that $\mathfrak{A} \models_{Y_\psi} \psi$ and $\mathfrak{A} \models_{Y_\varphi} \varphi$.

By induction hypothesis $\mathfrak{A} \models_{X_\psi \cup Y_\psi} \psi$ and $\mathfrak{A} \models_{X_\varphi \cup Y_\varphi} \varphi$, and therefore $\mathfrak{A} \models_{X \cup Y} \psi \vee \varphi$.

Inclusion logic is closed under union of teams

Existential quantification: assume that $\mathfrak{A} \models_X \exists y\, \psi$ and $\mathfrak{A} \models_Y \exists y\, \psi$.
There exist functions $F_X : X \to \mathcal{P}(A)$ and $F_Y : Y \to \mathcal{P}(A)$ such that $\mathfrak{A} \models_{X[y \mapsto F_X]} \psi$ and $\mathfrak{A} \models_{X[y \mapsto F_X]} \psi$. Define $F : X \cup Y \to \mathcal{P}(A)$ by

$$F(s) := \begin{cases} F_X(s) & \text{if } s \in X \setminus Y \\ F_Y(s) & \text{if } s \in Y \setminus X \\ F_X(s) \cup F_Y(s) & \text{if } s \in X \cap Y \end{cases}$$

Then $(X \cup Y)[y \mapsto F] = X[y \mapsto F_X] \cup Y[y \mapsto F_Y]$. Hence $\mathfrak{A}_{(X \cup Y)[y \mapsto F]} \models \psi$ and therefore $\mathfrak{A} \models_{X \cup Y} \exists y\, \psi$.

Inclusion logic is closed under union of teams

Existential quantification: assume that $\mathfrak{A} \models_X \exists y\, \psi$ and $\mathfrak{A} \models_Y \exists y\, \psi$.
There exist functions $F_X : X \to \mathcal{P}(A)$ and $F_Y : Y \to \mathcal{P}(A)$ such that $\mathfrak{A} \models_{X[y \mapsto F_X]} \psi$ and $\mathfrak{A} \models_{X[y \mapsto F_X]} \psi$. Define $F : X \cup Y \to \mathcal{P}(A)$ by

$$F(s) := \begin{cases} F_X(s) & \text{if } s \in X \setminus Y \\ F_Y(s) & \text{if } s \in Y \setminus X \\ F_X(s) \cup F_Y(s) & \text{if } s \in X \cap Y \end{cases}$$

Then $(X \cup Y)[y \mapsto F] = X[y \mapsto F_X] \cup Y[y \mapsto F_Y]$. Hence $\mathfrak{A}_{(X \cup Y)[y \mapsto F]} \models \psi$ and therefore $\mathfrak{A} \models_{X \cup Y} \exists y\, \psi$.

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

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

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

This generalizes to all formulae of inclusion logic.

Translating inclusion logic into fixed-point logic

For every formula $\psi(\overline{y})$ of inclusion logic there is a formula $\varphi(X, \overline{y})$ in posGFP, such that, for all structures $\mathfrak{A}$ and all teams X,

$$\mathfrak{A} \models_X \psi(\overline{y}) \quad \Longleftrightarrow \quad F_\varphi^{\mathfrak{A}}(X) \supseteq X \quad \Longleftrightarrow \quad (\mathfrak{A}, X) \models \forall X(X\overline{y} \rightarrow \varphi(X, \overline{y}))$$

For a sentence ψ of Inc, also $\varphi \in$ posGFP is a sentence, which is equivalent to ψ.

Corollary. Every class of finite structures definable in Inc is also definable in posGFP, and hence in PTIME.

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

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

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

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

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 twe can use inclusion statements $\overline{y} \subseteq \overline{x}$ for expressing that $\overline{y} \in X^\alpha$.

Illustration via safety games

$\psi(W,x) = (V_0x \wedge \exists y(Exy \wedge Wy)) \vee (V_1x \wedge \forall y(Exy \rightarrow Wy))$

$\mathcal{G} = (V, V_0, V_1, E) \vDash [\mathbf{gfp}\, Wx \,.\, \psi(W,x)](v)$ 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):

$\varphi(x) = (V_0x \wedge \exists y(Exy \wedge y \subseteq x)) \vee (V_1x \wedge \forall y(\neg Exy \vee y \subseteq x))$

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

$\psi(W,x) = (V_0x \wedge \exists y(Exy \wedge Wy)) \vee (V_1x \wedge \forall y(Exy \to Wy))$

$\mathcal{G} = (V, V_0, V_1, E) \models [\mathbf{gfp}\, Wx\,.\,\psi(W,x)](v)$ 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):

$\varphi(x) = \exists z(z \subseteq x \wedge ((V_0x \wedge \exists y(Exy \wedge y \subseteq z)) \vee (V_1x \wedge \forall y(\neg Exy \vee y \subseteq z))$

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 $y \in W$.

Inclusion logic and least fixed-point logic

To summarize

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

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

Inclusion logic and least fixed-point logic

To summarize

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

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

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

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

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