Dependence logic

Miika Hannula

University of Tartu

January 21, 2025

"x is the square root of y"

"x is a prime"

"node x is connected to node y"

"x depends on y"

"x is the square root of y"

"x is a prime"

"node x is connected to node y"

"x depends on y"

Dependence logic

Single assignment $\mapsto$ Set of assignments

Employee	Department	Salary
Alice	Math	50k
Bob	CS	40k
Carol	Physics	60k
David	Math	80k

Salary depends on Employee but not on Department.

Team semantics provides a framework for analyzing more complex statements involving dependencies.

Basic semantical unit is a set of entities (assignments, possible worlds, traces, ...)

Dependence logic

Single assignment $\mapsto$ Set of assignments

Employee	Department	Salary
Alice	Math	50k
Bob	CS	40k
Carol	Physics	60k
David	Math	80k

Salary depends on Employee but not on Department.

Team semantics provides a framework for analyzing more complex statements involving dependencies.

Basic semantical unit is a set of entities (assignments, possible worlds, traces, ...)

Dependence logic

Single assignment $\mapsto$ Set of assignments

Employee	Department	Salary
Alice	Math	50k
Bob	CS	40k
Carol	Physics	60k
David	Math	80k

`Salary` depends on `Employee` but not on `Department`.

Team semantics provides a framework for analyzing more complex statements involving dependencies.

Basic semantical unit is a set of entities (assignments, possible worlds, traces, ...)

Dependence logic

Single assignment $\mapsto$ Set of assignments

Employee	Department	Salary
Alice	Math	50k
Bob	CS	40k
Carol	Physics	60k
David	Math	80k

`Salary` depends on `Employee` but not on `Department`.

Team semantics provides a framework for analyzing more complex statements involving dependencies.

Basic semantical unit is a set of entities (assignments, possible worlds, traces, ...)

Concrete notions of dependence and independence

Dependence and independence occur in contexts such as:

- dependence of a move of a player in a game on some previous moves;
- dependence of an attribute of a database on other attributes;
- dependence/independence of a choice of an agent on choices of other agents;
- linear dependence/independence of a vector v of vectors $v_1, \ldots, v_n$;
- Independence of random variables X and Y;
- dependence of an outcome an experiment e_0 on the outcomes of $e_1, \ldots, e_n$.

Expressing complex variable dependencies not possible in first-order logic

Concrete notions of dependence and independence

Dependence and independence occur in contexts such as:

- dependence of a move of a player in a game on some previous moves;
- dependence of an attribute of a database on other attributes;
- dependence/independence of a choice of an agent on choices of other agents;
- linear dependence/independence of a vector v of vectors $v_1, \ldots, v_n$;
- Independence of random variables X and Y;
- dependence of an outcome an experiment e_0 on the outcomes of $e_1, \ldots, e_n$.

Expressing complex variable dependencies not possible in first-order logic

Concrete notions of dependence and independence

Dependence and independence occur in contexts such as:

- dependence of a move of a player in a game on some previous moves;
- dependence of an attribute of a database on other attributes;
- dependence/independence of a choice of an agent on choices of other agents;
- linear dependence/independence of a vector v of vectors $v_1, \ldots, v_n$;
- Independence of random variables X and Y;
- dependence of an outcome an experiment e_0 on the outcomes of $e_1, \ldots, e_n$.

Expressing complex variable dependencies not possible in first-order logic

Concrete notions of dependence and independence

Dependence and independence occur in contexts such as:

- dependence of a move of a player in a game on some previous moves;
- dependence of an attribute of a database on other attributes;
- dependence/independence of a choice of an agent on choices of other agents;
- linear dependence/independence of a vector v of vectors $v_1, \ldots, v_n$;
- Independence of random variables X and Y;
- dependence of an outcome an experiment e_0 on the outcomes of $e_1, \ldots, e_n$.

Expressing complex variable dependencies not possible in first-order logic

Concrete notions of dependence and independence

Dependence and independence occur in contexts such as:

- dependence of a move of a player in a game on some previous moves;
- dependence of an attribute of a database on other attributes;
- dependence/independence of a choice of an agent on choices of other agents;
- linear dependence/independence of a vector v of vectors $v_1, \dots, v_n$;
- Independence of random variables X and Y;
- dependence of an outcome an experiment e_0 on the outcomes of $e_1, \dots, e_n$.

Expressing complex variable dependencies not possible in first-order logic

Concrete notions of dependence and independence

Dependence and independence occur in contexts such as:

- dependence of a move of a player in a game on some previous moves;
- dependence of an attribute of a database on other attributes;
- dependence/independence of a choice of an agent on choices of other agents;
- linear dependence/independence of a vector v of vectors $v_1, \ldots, v_n$;
- Independence of random variables X and Y;
- dependence of an outcome an experiment e_0 on the outcomes of $e_1, \ldots, e_n$.

Expressing complex variable dependencies not possible in first-order logic

Concrete notions of dependence and independence

Dependence and independence occur in contexts such as:

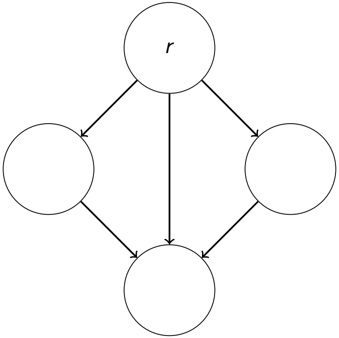
- dependence of a move of a player in a game on some previous moves;
- dependence of an attribute of a database on other attributes;
- dependence/independence of a choice of an agent on choices of other agents;
- linear dependence/independence of a vector v of vectors $v_1, \ldots, v_n$;
- Independence of random variables X and Y;
- dependence of an outcome an experiment e_0 on the outcomes of $e_1, \ldots, e_n$.

Expressing complex variable dependencies not possible in first-order logic

First-order logic (FO)

First-order logic (FO) formed by closing atomic formulas ($t = u, R(\vec{t})$) in terms of connectives $\neg, \vee, \wedge$ and quantifiers $\exists, \forall$.

Considering a directed graph G with edge relation E and root r



1. $G \models \forall x \neg E(x, r)$
2. $G \not\models \forall x \exists y E(x, y)$
3. $G \models \forall x \forall y \forall z($
   $E(x, y) \wedge E(y, z) \rightarrow E(x, z)$
   $)$

FO: Limits of expressiveness

Consider the FO-formula

$$\forall u \exists v \forall x \exists y \phi$$

Variable dependence is transitive:

- v is in the scope of u
- y is in the scope of v
- $\implies y$ is in the scope of u

Dependence relations between variables arise from quantification order!

Other limits: no counting, no recursion, captures only local properties

FO: Limits of expressiveness

Consider the FO-formula

$$\forall u \exists v \forall x \exists y \phi$$

Variable dependence is transitive:

- v is in the scope of u
- y is in the scope of v
- $\implies y$ is in the scope of u

Dependence relations between variables arise from quantification order!

Other limits: no counting, no recursion, captures only local properties

Solutions I

Branching quantifiers (also called Henkin quantifiers) [Hen61]:

$$\begin{pmatrix} \forall x & \exists y \\ \forall u & \exists v \end{pmatrix} \phi \tag{1}$$

read as:

for all x there is y that depends only on x,
and for all u there is v that depends only on u,
s.t. ϕ

Solutions II

Independence-friendly logic [HS89]:

$$\forall x \exists y \forall u (\exists v / x) \phi$$

where $(\exists y / u)$ is read as:

there exists v independently of x

Solutions III

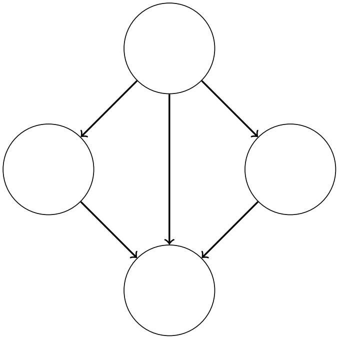
Dependence logic [V07]:

$$\forall x \exists y \forall u \exists v (\mathrm{dep}(u, v) \wedge \phi)$$

where $\mathrm{dep}(x, y)$ is a dependence atom stating that v depends only on u

Example cont.

Consider a directed graph G with edge relation E. Assume additional distinct constants b, r. Then:



$$
\begin{aligned}
\mathcal{M} \text{ is bipartite } &\iff \mathcal{M} \models \begin{pmatrix} \forall x & \exists y \\ \forall u & \exists v \end{pmatrix} \phi \\
&\iff G \models \forall x \exists y \forall u (\exists v / x) \phi \\
&\iff G \models \forall x \exists y \forall u \exists v (\mathrm{dep}(u, v) \wedge \phi)
\end{aligned}
$$

where:

$$\phi = (y = b \vee y = r) \wedge (x = u \rightarrow y = v) \wedge (E(x, u) \rightarrow \neg y = v)$$

From syntax to semantics?

First-order logic has both model-theoretic and game-theoretic semantics:

Model-theoretic semantics (Tarski, 1930s):

- Recursive definition of the satisfaction relation $\mathcal{M} \models \phi$ (model $\mathcal{M}$ satisfies formula ϕ)
- E.g., $\mathcal{M} \models \psi \wedge \theta$ iff $\mathcal{M} \models \psi$ and $\mathcal{M} \models \theta$.

Game-theoretic semantics (Lorenzen, Hintikka, 1950s):

- Two players: Verifier and Falsifier
- Consider: $\phi = \forall x \exists y E(x, y)$
  - Falsifier picks x
  - Verifier picks y
  - If $E(x, y)$ holds, Verifier wins; otherwise Falsifier wins.
  - ϕ is true iff Verifier has a winning strategy

From syntax to semantics with dependencies

Dependence/Independence-Friendly logic

Game-theoretic semantics:

- Consider $\forall x \exists y \forall u (\exists v / x) \phi$
  - Imperfect information game
  - Verifier should choose v independently x
  - Formula is true if Verifiers has a winning strategy

Model-theoretic semantics:

- Found by Hodges (1990s)
- Next few slides: model-theoretic semantics of dependence logic

Dependence logic

Dependence logic, $\mathrm{FO}(\mathrm{dep}(\cdots))$, defined via grammar:

$$\phi ::= \theta \mid \mathrm{dep}(\vec{x}, y) \mid \phi \wedge \phi \mid \phi \vee \phi \mid \exists x \phi \mid \forall x \phi,$$

where θ is a literal ($=$ atom or its negation).

NB. Negation pushed in front of first-order atoms

Next: Model-theoretic semantics for dependence logic

Team

Let A be a set and $V = \{x_1, \ldots, x_k\}$ a finite set of variables. A team X with domain V is a set of assignments

$$s \colon V \to A.$$

Intuition: data table with columns named as $x_1 \ldots x_n$

Dependence atom

Let $\vec{x}$ be a tuple of variables, and y an variable. An expression $\mathrm{dep}(\vec{x}, y)$ is called a dependence atom. For a model $\mathcal{M}$ and X its team, we define:

$$\mathcal{M} \models_X \mathrm{dep}(\vec{x}, y) \text{ if for all } s, s' \in X : s(\vec{x}) = s'(\vec{x}) \text{ implies } s(y) = s'(y).$$

Dependence atom

Let $\vec{x}$ be a tuple of variables, and y an variable. An expression $\mathrm{dep}(\vec{x}, y)$ is called a dependence atom. For a model $\mathcal{M}$ and X its team, we define:

$$\mathcal{M} \models_X \mathrm{dep}(\vec{x}, y) \text{ if for all } s, s' \in X : s(\vec{x}) = s'(\vec{x}) \text{ implies } s(y) = s'(y).$$

$\mathrm{dep}(x, y)$

x	y
1	2
2	3
2	4
3	3
3	4
4	1

Dependence atom

Let $\vec{x}$ be a tuple of variables, and y an variable. An expression $\mathrm{dep}(\vec{x}, y)$ is called a dependence atom. For a model $\mathcal{M}$ and X its team, we define:

$$\mathcal{M} \models_X \mathrm{dep}(\vec{x}, y) \text{ if for all } s, s' \in X : s(\vec{x}) = s'(\vec{x}) \text{ implies } s(y) = s'(y).$$

$\mathrm{dep}(x, y)$

x	y
1	2
2	3
2	4
3	3
3	4
4	1

Team Semantics: From Tarski to Hodges

Definition (Team Semantics [Hod97])

Let $\mathcal{M}$ be a τ-model, and X its team. The satisfaction relation $\mathcal{M} \models_X \phi$ for first-order literals and compound formulas is defined inductively as follows:

- For a literal ϕ, $\mathcal{M} \models_X \phi$ iff $\mathcal{M} \models_s \phi$ for all $s \in X$;
- $\mathcal{M} \models_s \phi \wedge \psi$ iff $\mathcal{M} \models_s \phi$ and $\mathcal{M} \models_s \psi$;
- $\mathcal{M} \models_s \phi \vee \psi$ iff $\mathcal{M} \models_s \phi$ or $\mathcal{M} \models_s \psi$;
- $\mathcal{M} \models_s \exists x \phi$ iff there exists $a \in \mathrm{Dom}(\mathcal{M})$ such that $\mathcal{M} \models_{s(a/x)} \phi$;
- $\mathcal{M} \models_s \forall x \phi$ iff for all $a \in \mathrm{Dom}(\mathcal{M})$, $\mathcal{M} \models_{s(a/x)} \phi$.

From Tarski to Hodges

Definition (Team Semantics [Hod97])

Let $\mathcal{M}$ be a τ-model, and X its team. The satisfaction relation $\mathcal{M} \models_X \phi$ for any negation-normal form first-order formula ϕ is defined inductively as follows:

- For a literal ϕ, $\mathcal{M} \models_X \phi$ iff $\mathcal{M} \models_s \phi$ for all $s \in X$;
- $\mathcal{M} \models_X \phi \wedge \psi$ iff $\mathcal{M} \models_X \phi$ and $\mathcal{M} \models_X \psi$;
- $\mathcal{M} \models_s \phi \vee \psi$ iff $\mathcal{M} \models_s \phi$ or $\mathcal{M} \models_s \psi$;
- $\mathcal{M} \models_s \exists x \phi$ iff there exists $a \in \mathrm{Dom}(\mathcal{M})$ such that $\mathcal{M} \models_{s(a/x)} \phi$;
- $\mathcal{M} \models_s \forall x \phi$ iff for all $a \in \mathrm{Dom}(\mathcal{M})$, $\mathcal{M} \models_{s(a/x)} \phi$.

From Tarski to Hodges

Definition (Team Semantics [Hod97])

Let $\mathcal{M}$ be a τ-model, and X its team. The satisfaction relation $\mathcal{M} \models_X \phi$ for any negation-normal form first-order formula ϕ is defined inductively as follows:

- For a literal ϕ, $\mathcal{M} \models_X \phi$ iff $\mathcal{M} \models_s \phi$ for all $s \in X$;
- $\mathcal{M} \models_X \phi \wedge \psi$ iff $\mathcal{M} \models_X \phi$ and $\mathcal{M} \models_X \psi$;
- $\mathcal{M} \models_X \phi \vee \psi$ iff there are $Y, Z \subseteq X$, $Y \cup Z = X$, s.t. $\mathcal{M} \models_Y \phi$ and $\mathcal{M} \models_Z \psi$;
- $\mathcal{M} \models_s \exists x \phi$ iff there exists $a \in \mathrm{Dom}(\mathcal{M})$ such that $\mathcal{M} \models_{s(a/x)} \phi$;
- $\mathcal{M} \models_s \forall x \phi$ iff for all $a \in \mathrm{Dom}(\mathcal{M})$, $\mathcal{M} \models_{s(a/x)} \phi$.

number	position	nationality	name
1	GK	BEL	Courtois
2	DF	SPA	Carvajal
22	DF	GER	Rüdiger
6	DF	SPA	Nacho
23	DF	FRA	Mendy
15	MF	URU	Valverde
12	MF	FRA	Camavinga
8	MF	GER	Kroos
5	MF	ENG	Bellingham
11	FW	BRA	Rodrygo
7	FW	BRA	Vini Jr.

$$\mathcal{M} \models_X \texttt{position} = \mathit{FW} \rightarrow \texttt{nationality} = \mathit{BRA}$$

number	position	nationality	name
1	GK	BEL	Courtois
2	DF	SPA	Carvajal
22	DF	GER	Rüdiger
6	DF	SPA	Nacho
23	DF	FRA	Mendy
15	MF	URU	Valverde
12	MF	FRA	Camavinga
8	MF	GER	Kroos
5	MF	ENG	Bellingham
11	FW	BRA	Rodrygo
7	FW	BRA	Vini Jr.

$$\mathcal{M} \models_X \neg \texttt{position} = \mathit{FW} \vee \texttt{nationality} = \mathit{BRA}$$

number	position	nationality	name
1	GK	BEL	Courtois
2	DF	SPA	Carvajal
22	DF	GER	Rüdiger
6	DF	SPA	Nacho
23	DF	FRA	Mendy
15	MF	URU	Valverde
12	MF	FRA	Camavinga
8	MF	GER	Kroos
5	MF	ENG	Bellingham
11	FW	BRA	Rodrygo
7	FW	BRA	Vini Jr.

$$\mathcal{M} \models_x \neg \texttt{position} = \mathit{FW} \vee \texttt{nationality} = \mathit{BRA}$$

number	position	nationality	name
1	GK	BEL	Courtois
2	DF	SPA	Carvajal
22	DF	GER	Rüdiger
6	DF	SPA	Nacho
23	DF	FRA	Mendy
15	MF	URU	Valverde
12	MF	FRA	Camavinga
8	MF	GER	Kroos
5	MF	ENG	Bellingham
11	FW	BRA	Rodrygo
7	FW	BRA	Vini Jr.

$$\mathcal{M} \models_X \texttt{position} = GK \rightarrow \mathrm{dep}(\emptyset, (\texttt{number, nationality, name}))$$

1

[1] $\mathrm{dep}(\emptyset, (\texttt{number, nationality, name}))$: number, nationality, name depend on the empty sequence (i.e., are constant)

number	position	nationality	name
1	GK	BEL	Courtois
2	DF	SPA	Carvajal
22	DF	GER	Rüdiger
6	DF	SPA	Nacho
23	DF	FRA	Mendy
15	MF	URU	Valverde
12	MF	FRA	Camavinga
8	MF	GER	Kroos
5	MF	ENG	Bellingham
11	FW	BRA	Rodrygo
7	FW	BRA	Vini Jr.

$$\mathcal{M} \models_X \neg \texttt{position} = GK \vee \mathrm{dep}(\emptyset, (\texttt{number, nationality, name}))$$

1

[1] $\mathrm{dep}(\emptyset, (\texttt{number, nationality, name}))$: number, nationality, name depend on the empty sequence (i.e., are constant)

number	position	nationality	name
1	GK	BEL	Courtois
2	DF	SPA	Carvajal
22	DF	GER	Rüdiger
6	DF	SPA	Nacho
23	DF	FRA	Mendy
15	MF	URU	Valverde
12	MF	FRA	Camavinga
8	MF	GER	Kroos
5	MF	ENG	Bellingham
11	FW	BRA	Rodrygo
7	FW	BRA	Vini Jr.

$$\mathcal{M} \models_X \neg \texttt{position} = GK \vee \mathrm{dep}(\emptyset, (\texttt{number}, \texttt{nationality}, \texttt{name}))$$

1

[1] $\mathrm{dep}(\emptyset, (\texttt{number}, \texttt{nationality}, \texttt{name}))$: number, nationality, name depend on the empty sequence (i.e., are constant)

number	position	nationality	name
1	GK	BEL	Courtois
2	DF	SPA	Carvajal
22	DF	GER	Rüdiger
6	DF	SPA	Nacho
23	DF	FRA	Mendy
15	MF	URU	Valverde
12	MF	FRA	Camavinga
8	MF	GER	Kroos
5	MF	ENG	Bellingham
11	FW	BRA	Rodrygo
7	FW	BRA	Vini Jr.

$$\mathcal{M} \not\models_X (\texttt{nationality} = \mathit{SPA} \to \texttt{position} = \mathit{DF}) \wedge \neg(\texttt{position} = \mathit{DF} \to \texttt{nationality} = \mathit{SPA})$$

number	position	nationality	name
1	GK	BEL	Courtois
2	DF	SPA	Carvajal
22	DF	GER	Rüdiger
6	DF	SPA	Nacho
23	DF	FRA	Mendy
15	MF	URU	Valverde
12	MF	FRA	Camavinga
8	MF	GER	Kroos
5	MF	ENG	Bellingham
11	FW	BRA	Rodrygo
7	FW	BRA	Vini Jr.

$$\mathcal{M} \not\models_X (\neg \texttt{nationality} = \mathit{SPA} \vee \texttt{position} = \mathit{DF}) \wedge \neg(\neg \texttt{position} = \mathit{DF} \vee \texttt{nationality} = \mathit{SPA})$$

number	position	nationality	name
1	GK	BEL	Courtois
2	DF	SPA	Carvajal
22	DF	GER	Rüdiger
6	DF	SPA	Nacho
23	DF	FRA	Mendy
15	MF	URU	Valverde
12	MF	FRA	Camavinga
8	MF	GER	Kroos
5	MF	ENG	Bellingham
11	FW	BRA	Rodrygo
7	FW	BRA	Vini Jr.

$$\mathcal{M} \not\models_X (\neg \texttt{nationality} = SPA \vee \texttt{position} = DF) \wedge (\texttt{position} = DF \wedge \neg \texttt{nationality} = SPA)$$

number	position	nationality	name
1	GK	BEL	Courtois
2	DF	SPA	Carvajal
22	DF	GER	Rüdiger
6	DF	SPA	Nacho
23	DF	FRA	Mendy
15	MF	URU	Valverde
12	MF	FRA	Camavinga
8	MF	GER	Kroos
5	MF	ENG	Bellingham
11	FW	BRA	Rodrygo
7	FW	BRA	Vini Jr.

$$\mathcal{M} \not\models_X (\neg \texttt{nationality} = \textit{SPA} \vee \texttt{position} = \textit{DF}) \wedge \texttt{position} = \textit{DF} \wedge \neg \texttt{nationality} = \textit{SPA}$$

number	position	nationality	name
1	GK	BEL	Courtois
2	DF	SPA	Carvajal
22	DF	GER	Rüdiger
6	DF	SPA	Nacho
23	DF	FRA	Mendy
15	MF	URU	Valverde
12	MF	FRA	Camavinga
8	MF	GER	Kroos
5	MF	ENG	Bellingham
11	FW	BRA	Rodrygo
7	FW	BRA	Vini Jr.

$$\mathcal{M} \models_X (\texttt{nationality} = \mathit{SPA} \to \texttt{position} = \mathit{DF}) \wedge \\ \sim(\texttt{position} = \mathit{DF} \to \texttt{nationality} = \mathit{SPA})$$

NB. $\neg$ is not the classical negation (which is usually denoted by $\sim$ in team semantics)

From Tarski to Hodges

Definition (Team Semantics [Hod97])

Let $\mathcal{M}$ be a τ-model, and X its team. The satisfaction relation $\mathcal{M} \models_X \phi$ for any negation-normal form first-order formula ϕ is defined inductively as follows:

- For a literal ϕ, $\mathcal{M} \models_X \phi$ iff $\mathcal{M} \models_s \phi$ for all $s \in X$;
- $\mathcal{M} \models_X \phi \wedge \psi$ iff $\mathcal{M} \models_X \phi$ and $\mathcal{M} \models_X \psi$;
- $\mathcal{M} \models_X \phi \vee \psi$ iff there are $Y, Z \subseteq X$, $Y \cup Z = X$, s.t. $\mathcal{M} \models_Y \phi$ and $\mathcal{M} \models_Z \psi$;
- $\mathcal{M} \models_s \exists x \phi$ iff there exists $a \in \mathrm{Dom}(\mathcal{M})$ such that $\mathcal{M} \models_{s(a/x)} \phi$;
- $\mathcal{M} \models_s \forall x \phi$ iff for all $a \in \mathrm{Dom}(\mathcal{M})$, $\mathcal{M} \models_{s(a/x)} \phi$.

From Tarski to Hodges

Definition (Team Semantics [Hod97])

Let $\mathcal{M}$ be a τ-model, and X its team. The satisfaction relation $\mathcal{M} \models_X \phi$ for any negation-normal form first-order formula ϕ is defined inductively as follows:

- For a literal ϕ, $\mathcal{M} \models_X \phi$ iff $\mathcal{M} \models_s \phi$ for all $s \in X$;
- $\mathcal{M} \models_X \phi \wedge \psi$ iff $\mathcal{M} \models_X \phi$ and $\mathcal{M} \models_X \psi$;
- $\mathcal{M} \models_X \phi \vee \psi$ iff there are $Y, Z \subseteq X$, $Y \cup Z = X$, s.t. $\mathcal{M} \models_Y \phi$ and $\mathcal{M} \models_Z \psi$;
- $\mathcal{M} \models_X \exists x \phi$ iff there exists $F : X \to \mathcal{P}(\mathrm{Dom}(\mathcal{M})) \setminus \{\emptyset\}$ such that $\mathcal{M} \models_{X[F/x]} \phi$;
- $\mathcal{M} \models_s \forall x \phi$ iff for all $a \in \mathrm{Dom}(\mathcal{M})$, $\mathcal{M} \models_{s(a/x)} \phi$.

Team Semantics: existential quantification

$$\mathcal{M} \models_X \exists x \phi \text{ if } \mathcal{M} \models_{X[F/x]} \phi \text{ for some } F : X \to \mathcal{P}(\mathrm{Dom}(\mathcal{M})) \setminus \{\emptyset\},$$

where $X[F/x] := \{s(a/x) \mid a \in F(s)\}$.

Team Semantics: existential quantification

$$\mathcal{M} \models_X \exists x \phi \text{ if } \mathcal{M} \models_{X[F/x]} \phi \text{ for some } F : X \to \mathcal{P}(\mathrm{Dom}(\mathcal{M})) \setminus \{\emptyset\},$$

where $X[F/x] := \{s(a/x) \mid a \in F(s)\}$.

$\exists x \phi(y, z)$

y	z
1	2
2	3

Team Semantics: existential quantification

$$\mathcal{M} \models_X \exists x \phi \text{ if } \mathcal{M} \models_{X[F/x]} \phi \text{ for some } F : X \to \mathcal{P}(\mathrm{Dom}(\mathcal{M})) \setminus \{\emptyset\},$$

where $X[F/x] := \{s(a/x) \mid a \in F(s)\}$.

$\exists x \phi(y, z)$

y	z	x
1	2	1
1	2	2
1	2	3
2	3	1
2	3	2
2	3	3

Team Semantics: existential quantification

$$\mathcal{M} \models_X \exists x \phi \text{ if } \mathcal{M} \models_{X[F/x]} \phi \text{ for some } F : X \to \mathcal{P}(\mathrm{Dom}(\mathcal{M})) \setminus \{\emptyset\},$$

where $X[F/x] := \{s(a/x) \mid a \in F(s)\}$.

$\exists x \phi(y, z)$

y	z	x
1	2	1
1	2	2
1	2	3
2	3	1
2	3	2
2	3	3

From Tarski to Hodges

Definition (Team Semantics [Hod97])

Let $\mathcal{M}$ be a τ-model, and X its team. The satisfaction relation $\mathcal{M} \models_X \phi$ for any negation-normal form first-order formula ϕ is defined inductively as follows:

- For a literal ϕ, $\mathcal{M} \models_X \phi \quad \text{iff} \quad \mathcal{M} \models_s \phi$ for all $s \in X$;
- $\mathcal{M} \models_X \phi \wedge \psi \quad \text{iff} \quad \mathcal{M} \models_X \phi$ and $\mathcal{M} \models_X \psi$;
- $\mathcal{M} \models_X \phi \vee \psi \quad \text{iff} \quad$ there are $Y, Z \subseteq X$, $Y \cup Z = X$, s.t. $\mathcal{M} \models_Y \phi$ and $\mathcal{M} \models_Z \psi$;
- $\mathcal{M} \models_X \exists x \phi \quad \text{iff} \quad$ there exists $F : X \to \mathcal{P}(\mathrm{Dom}(\mathcal{M})) \setminus \{\emptyset\}$ such that $\mathcal{M} \models_{X[F/x]} \phi$;
- $\mathcal{M} \models_s \forall x \phi \quad \text{iff} \quad$ for all $a \in \mathrm{Dom}(\mathcal{M})$, $\mathcal{M} \models_{s(a/x)} \phi$.

From Tarski to Hodges

Definition (Team Semantics [Hod97])

Let $\mathcal{M}$ be a τ-model, and X its team. The satisfaction relation $\mathcal{M} \models_X \phi$ for any negation-normal form first-order formula ϕ is defined inductively as follows:

- For a literal ϕ, $\mathcal{M} \models_X \phi$ iff $\mathcal{M} \models_s \phi$ for all $s \in X$;
- $\mathcal{M} \models_X \phi \wedge \psi$ iff $\mathcal{M} \models_X \phi$ and $\mathcal{M} \models_X \psi$;
- $\mathcal{M} \models_X \phi \vee \psi$ iff there are $Y, Z \subseteq X$, $Y \cup Z = X$, s.t. $\mathcal{M} \models_Y \phi$ and $\mathcal{M} \models_Z \psi$;
- $\mathcal{M} \models_X \exists x \phi$ iff there exists $F : X \to \mathcal{P}(\mathrm{Dom}(\mathcal{M})) \setminus \{\emptyset\}$ such that $\mathcal{M} \models_{X[F/x]} \phi$;
- $\mathcal{M} \models_X \forall x \phi$ iff $\mathcal{M} \models_{X[\mathrm{Dom}(\mathcal{M})/x]} \phi$.

Team Semantics: universal quantification

$$\mathcal{M} \models_X \forall x \phi \text{ if } \mathcal{M} \models_{X[\mathrm{Dom}(\mathcal{M})/x]} \phi,$$

where $X[A/x] := \{s(a/x) \mid a \in A\}$.

Team Semantics: universal quantification

$$\mathcal{M} \models_X \forall x \phi \text{ if } \mathcal{M} \models_{X[\mathrm{Dom}(\mathcal{M})/x]} \phi,$$

where $X[A/x] := \{s(a/x) \mid a \in A\}$.

$\forall x \phi(y, z)$

y	z
1	2
2	3

Team Semantics: universal quantification

$$\mathcal{M} \models_X \forall x\phi \text{ if } \mathcal{M} \models_{X[\mathrm{Dom}(\mathcal{M})/x]} \phi,$$

where $X[A/x] := \{s(a/x) \mid a \in A\}$.

$\forall x\phi(y, z)$

y	z	x
1	2	1
		2
		3
2	3	1
		2
		3

Team Semantics: universal quantification

$$\mathcal{M} \models_X \forall x \phi \text{ if } \mathcal{M} \models_{X[\mathrm{Dom}(\mathcal{M})/x]} \phi,$$

where $X[A/x] := \{s(a/x) \mid a \in A\}$.

$\forall x \phi(y, z)$

<table>
<tr><th>y</th><th>z</th><th>x</th></tr>
<tr><td rowspan="3">1</td><td rowspan="3">2</td><td>1</td></tr>
<tr><td>2</td></tr>
<tr><td>3</td></tr>
<tr><td rowspan="3">2</td><td rowspan="3">3</td><td>1</td></tr>
<tr><td>2</td></tr>
<tr><td>3</td></tr>
</table>

Connections

Dependence logic can be used to formalize reasoning about statements related to data (outcomes of experiments, voting profiles, databases):

- No-Go theorems (e.g., Bell's theorem) in quantum mechanics [APV24, AG22]
- Arrow's theorem in social choice [PY16]
- Logical implication for database dependencies [H. and Kontinen, 2016]

Limits of formal reasoning

First-order logic has a tight connection between syntax and semantics

Gödel's completeness theorem (1929). The following are equivalent for a first-order logic formula ϕ:

1. ϕ is valid (true in all structures)
2. ϕ has a finite formal deduction

Transferring rules

The following rules **hold true** in first-order logic:

- If $\phi \vee \phi$, then ϕ
- If $(\phi \wedge \psi) \vee (\phi \wedge \theta)$, then $\phi \wedge (\psi \vee \theta)$
- If $(\phi \vee \psi) \wedge (\phi \vee \theta)$, then $\phi \vee (\psi \wedge \theta)$

Transferring rules

The following rules **hold true** in first-order logic:

- If $\phi \vee \phi$, then ϕ
- If $(\phi \wedge \psi) \vee (\phi \wedge \theta)$, then $\phi \wedge (\psi \vee \theta)$
- If $(\phi \vee \psi) \wedge (\phi \vee \theta)$, then $\phi \vee (\psi \wedge \theta)$

These rules do not hold in dependence logic

Limits of formal reasoning cont.

Dependence logic does not have as tight a connection between syntax and semantics

The following are not equivalent for a dependence logic formula ϕ:

1. ϕ is valid (true in all structures)
2. ϕ has a finite formal deduction

Reason. Not possible to enumerate all valid formulas ϕ by a computer program. In contrast, a formal deduction system gives rise to such a program.

Limited formal reasoning

It is possible to create a formal deduction system that is sound and complete w.r.t. the following logical consequence relation:

$$T \models \psi \qquad\qquad / \; T \textit{ implies } \psi$$

where

- T is a set of dependence logic formulas
- ψ is a first-order logic formula

Such systems created in [KV13], [H. 15]

What can be expressed in dependence logic?

Those properties of teams (=data) that are closed downwards and definable in existential second-order logic (ESO):

- obtained by adding to a first-order logic formula ϕ a prefix of existentially quantified relations: $\exists R_1 \ldots \exists R_n \phi$

Example (Bipartiteness revisited)

Graph G with edge relation E is bipartite iff $G \models \exists P \exists Q \forall x \forall y \theta$, where θ is a conjunction of

- $P(x) \vee Q(x)$
- $\neg P(x) \vee \neg Q(x)$
- $(P(x) \wedge Q(y)) \rightarrow \neg E(x, y)$

Fagin's theorem

Non-deterministic polynomial time (NP) consists of all those problems that are solvable in polynomial-time by non-deterministic computation.

Theorem ([Fag74])

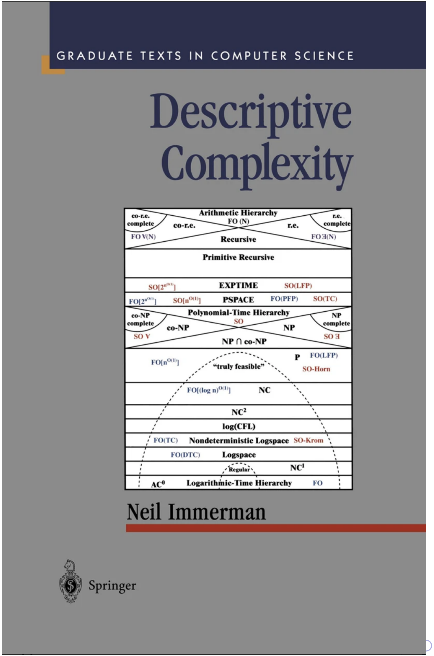
A class of finite structures can be recognized in NP *if and only if it can be described in existential second-order logic.*

Consequence: descriptive power of dependence logic $\approx$ computational power of NP

Side note: descriptive complexity

- Fagin's theorem led to development of descriptive complexity
- Idea is to find correspondencies between:
  - computational complexity classes
  - descriptive logical languages
- Examples:
  - ESO $=$ NP
  - full second-order logic $=$ polynomial-time hierarchy
- **Open question.** Is there a logic L such that

$$\mathrm{L} = \mathrm{PTIME}$$



Descriptive complexity of dependence logic(s)

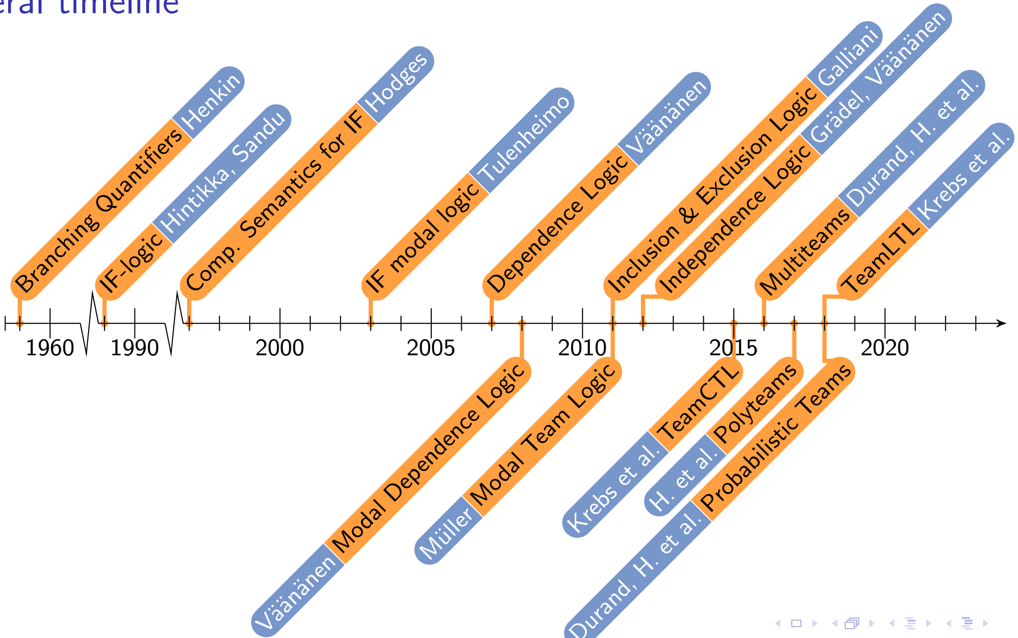
Different variants of dependence logic obtained by combining:

- Logical connectives and quantifiers
- Notions of dependence and independence

Recall: FO formed using $\neg, \wedge, \vee, \exists, \forall$

- Dependence logic: $\mathrm{FO} + \mathrm{dep}(x, y) \leq$ NP
- Independence logic: $\mathrm{FO} + y \perp_x z =$ NP [GV13]
- Inclusion logic: $\mathrm{FO} + x \subseteq y \leq$ PTIME [GH13]
- ...

General timeline



Complexity of Modal Logics in Team Semantics

Logic	SAT	VAL	MC
ML	PSPACE (Lad77)	PSPACE (Lad77)	P (CES86, Sch02)
$\mathrm{ML}(\subseteq)$	EXPTIME (HKMV15)	coNEXPTIME-hard (HKMV17)	P (HKMV17)
$\mathrm{ML}(\mathrm{dep}(\cdot))$	NEXPTIME (Sev09)	NEXPTIME (Virtema 14, H. 17)	NP (EbLo12)
$\mathrm{ML}(\perp)$	NEXPTIME (KMSV17)	$\mathrm{coNEXPTIME}^{\mathrm{NP}}$-hard (H. 19)	NP (KMSV17)
$\mathrm{ML}(\sim)$	TOWER(poly) (Lüc18)	TOWER(poly) (Lüc18)	PSPACE (Mül14)

TOWER(poly): computation time $2^{\underbrace{n^{n^{n^{\cdots^{n}}}}}_{poly}}$ with a polynomial upper bound for the exponent tower height

Complexity of Modal Logics in Team Semantics

Logic	SAT	VAL	MC
ML	PSPACE (Lad77)	PSPACE (Lad77)	P (CES86, Sch02)
$\mathrm{ML}(\subseteq)$	EXPTIME (HKMV15)	coNEXPTIME-hard (HKMV17)	P (HKMV17)
$\mathrm{ML}(\mathrm{dep}(\cdot))$	NEXPTIME (Sev09)	NEXPTIME (Virtema 14, H. 17)	NP (EbLo12)
$\mathrm{ML}(\perp)$	NEXPTIME (KMSV17)	$\mathrm{coNEXPTIME}^{\mathrm{NP}}$-hard (H. 19)	NP (KMSV17)
$\mathrm{ML}(\sim)$	TOWER(poly) (Lüc18)	TOWER(poly) (Lüc18)	PSPACE (Mül14)
$\mathrm{PL}(\sim)$	AEXPTIME(poly) (HKVV18)	AEXPTIME(poly) (HKVV18)	PSPACE (Mül14)

Interestingly, in many cases hardness holds already for the propositional frangment.

TOWER(poly): computation time $2^{\overbrace{n^{n^{n^{\cdots^{n}}}}}^{\text{poly}}}$ with a polynomial upper bound for the exponent tower height

Quantitative versions I

Dependence logic models relational but not quantitative notions of dependence

Logics of quantitative dependence (such as probabilistic independence) require extension of semantics:

x	y	z	#
a	b	c	2
a	b	d	1
b	a	c	1
b	a	d	1

$\longrightarrow$ Multiteam semantics [Durand, H. et al. 18]

Quantitative versions II

Dependence logic models relational but not quantitative notions of dependence

Logics of quantitative dependence (such as probabilistic independence) require extension of semantics:

x	y	z	Prob
a	b	c	$\frac{2}{5}$
a	b	d	$\frac{1}{5}$
b	a	c	$\frac{1}{5}$
b	a	d	$\frac{1}{5}$

$\longrightarrow$ Probabilistic team semantics [H. et al., 18, 20, 22]

Quantitative versions III

All versions of team semantics unified by assuming a generic number domain K, such as a positive semiring $(\mathbb{R}_{\geq 0}, \mathbb{N}, \mathbb{B}, \dots)$

x	y	z	K
a	b	c	k_1
a	b	d	k_2
b	a	c	k_3
b	a	d	k_4

$\longrightarrow$ Semiring team semantics [Barlag, H. et al., 23]

Example: the so-called *semi-graphoid axioms* of conditional independence are sound in most semirings [H. 2024]

Conclusion

Dependence logic:

- How dependencies and logic interact?
- Tool for the study of more complex dependence relations
- A vehicle for uncovering and unifying the mathematics of dependence in a variety of contexts:
  - Databases
  - Probability theory
  - Social choice theory
  - Quantum physics
  - ...

References I

- Rafael Albert and Erich Grädel, *Unifying hidden-variable problems from quantum mechanics by logics of dependence and independence*, Ann. Pure Appl. Log. **173** (2022), no. 10, 103088.
- Samson Abramsky, Joni Puljujärvi, and Jouko Väänänen, *Team semantics and independence notions in quantum physics*, 2024.
- Timon Barlag, Miika Hannula, Juha Kontinen, Nina Pardal, and Jonni Virtema, *Unified foundations of team semantics via semirings*, KR, 2023, pp. 75–85.
- Arnaud Durand, Miika Hannula, Juha Kontinen, Arne Meier, and Jonni Virtema, *Approximation and dependence via multiteam semantics*, Ann. Math. Artif. Intell. **83** (2018), no. 3-4, 297–320.
- ———, *Probabilistic team semantics*, Foundations of Information and Knowledge Systems - 10th International Symposium, FoIKS 2018, Budapest, Hungary, May 14-18, 2018, Proceedings, 2018, pp. 186–206.

References II

- Ronald Fagin, *Generalized first-order spectra and polynomial-time recognizable sets*, Complexity of Computation, SIAM-AMS Proceedings,, vol. Vol. 7, 1974, pp. 43–73.
- Pietro Galliani and Lauri Hella, *Inclusion Logic and Fixed Point Logic*, CSL 2013, 2013, pp. 281–295.
- Erich Grädel and Jouko Väänänen, *Dependence and independence*, Studia Logica **101** (2013), no. 2, 399–410 (English).
- Miika Hannula, *Axiomatizing first-order consequences in independence logic*, Ann. Pure Appl. Logic **166** (2015), no. 1, 61–91.
- Miika Hannula, *Conditional independence on semiring relations*, ICDT, LIPIcs, vol. 290, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024, pp. 20:1–20:20.
- Leon Henkin, *Some Remarks on Infinitely Long Formulas*, Infinitistic Methods. Proc. Symposium on Foundations of Mathematics, Pergamon Press, 1961, pp. 167–183.

References III

- Miika Hannula and Juha Kontinen, *A finite axiomatization of conditional independence and inclusion dependencies*, Inf. Comput. **249** (2016), 121–137.

- Miika Hannula, Juha Kontinen, Jan Van den Bussche, and Jonni Virtema, *Descriptive complexity of real computation and probabilistic independence logic*, LICS, ACM, 2020, pp. 550–563.

- Wilfrid Hodges, *Compositional Semantics for a Language of Imperfect Information*, Journal of the Interest Group in Pure and Applied Logics **5 (4)** (1997), 539–563.

- J. Hintikka and G. Sandu, *Informational independence as a semantical phenomenon*, Logic, Methodology and Philosophy of Science (R. Hilpinen J. E. Fenstad, I. T. Frolov, ed.), Amsterdam: Elsevier, 1989, pp. 571–589.

- Miika Hannula and Jonni Virtema, *Tractability frontiers in probabilistic team semantics and existential second-order logic over the reals*, Ann. Pure Appl. Log. **173** (2022), no. 10, 103108.

References IV

Juha Kontinen and Jouko Väänänen, *Axiomatizing first-order consequences in dependence logic*, Ann. Pure Appl. Logic **164** (2013), no. 11, 1101–1117.

Eric Pacuit and Fan Yang, *Dependence and independence in social choice: Arrow's theorem*, Dependence Logic, Springer, 2016, pp. 235–260.

Jouko Väänänen, *Dependence logic*, Cambridge University Press, 2007.