Team semantics and substitutionality

Fredrik Engström,
University of Gothenburg
joint work with Orvar Lorimer Olsson

January 27, 2025

Nordic Online Logic seminar

The plan

- Background and motivation
  - Standard team semantics and flatness
  - Non-standard team semantics: Generalized quantifiers
  - Substitutionality
- A substitutional propositional logic of teams
  - Semantics and adequacy
  - Deductions and completeness
  - Axiomatizing standard team semantics
- Further work

Background and motivation

Historical outline of the development of team semantics

- Henkin -61: Branching by skolemization: $\begin{pmatrix} \forall x \exists y \\ \forall z \exists w \end{pmatrix}$

Historical outline of the development of team semantics

- Henkin -61: Branching by skolemization: $\begin{pmatrix} \forall x \exists y \\ \forall z \exists w \end{pmatrix}$
- Hintikka -73: Branching as a contruction in natural languages

Historical outline of the development of team semantics

- Henkin -61: Branching by skolemization: $\begin{pmatrix} \forall x \exists y \\ \forall z \exists w \end{pmatrix}$
- Hintikka -73: Branching as a contruction in natural languages
- Barwise -79: Branching of generalized quantifiers

Historical outline of the development of team semantics

- Henkin -61: Branching by skolemization: $\begin{pmatrix} \forall x \exists y \\ \forall z \exists w \end{pmatrix}$
- Hintikka -73: Branching as a contruction in natural languages
- Barwise -79: Branching of generalized quantifiers
- Hintikka/Sandu -89: IF-logic, game theoretic semantics

Historical outline of the development of team semantics

- Henkin -61: Branching by skolemization: $\begin{pmatrix} \forall x \exists y \\ \forall z \exists w \end{pmatrix}$
- Hintikka -73: Branching as a contruction in natural languages
- Barwise -79: Branching of generalized quantifiers
- Hintikka/Sandu -89: IF-logic, game theoretic semantics
- Hodges -97: Compositional semantics for IF-logic

Historical outline of the development of team semantics

- Henkin -61: Branching by skolemization: $\begin{pmatrix} \forall x \exists y \\ \forall z \exists w \end{pmatrix}$
- Hintikka -73: Branching as a contruction in natural languages
- Barwise -79: Branching of generalized quantifiers
- Hintikka/Sandu -89: IF-logic, game theoretic semantics
- Hodges -97: Compositional semantics for IF-logic
- Väänänen -07: Dependence Logic

Historical outline of the development of team semantics

- Henkin -61: Branching by skolemization: $\begin{pmatrix} \forall x \exists y \\ \forall z \exists w \end{pmatrix}$
- Hintikka -73: Branching as a contruction in natural languages
- Barwise -79: Branching of generalized quantifiers
- Hintikka/Sandu -89: IF-logic, game theoretic semantics
- Hodges -97: Compositional semantics for IF-logic
- Väänänen -07: Dependence Logic

- Express non first-order properties within a first-order syntax.

Historical outline of the development of team semantics

- Henkin -61: Branching by skolemization: $\begin{pmatrix} \forall x \exists y \\ \forall z \exists w \end{pmatrix}$
- Hintikka -73: Branching as a contruction in natural languages
- Barwise -79: Branching of generalized quantifiers
- Hintikka/Sandu -89: IF-logic, game theoretic semantics
- Hodges -97: Compositional semantics for IF-logic
- Väänänen -07: Dependence Logic

- Express non first-order properties within a first-order syntax.
- Evaluating formulas w.r.t. sets of assignments, teams.

Historical outline of the development of team semantics

- Henkin -61: Branching by skolemization: $\begin{pmatrix} \forall x \exists y \\ \forall z \exists w \end{pmatrix}$
- Hintikka -73: Branching as a contruction in natural languages
- Barwise -79: Branching of generalized quantifiers
- Hintikka/Sandu -89: IF-logic, game theoretic semantics
- Hodges -97: Compositional semantics for IF-logic
- Väänänen -07: Dependence Logic

- Express non first-order properties within a first-order syntax.
- Evaluating formulas w.r.t. sets of assignments, teams.
- Can directly express independence, dependence, etc, via new atoms.

Historical outline of the development of team semantics

- Henkin -61: Branching by skolemization: $\begin{pmatrix} \forall x \exists y \\ \forall z \exists w \end{pmatrix}$
- Hintikka -73: Branching as a contruction in natural languages
- Barwise -79: Branching of generalized quantifiers
- Hintikka/Sandu -89: IF-logic, game theoretic semantics
- Hodges -97: Compositional semantics for IF-logic
- Väänänen -07: Dependence Logic

- Express non first-order properties within a first-order syntax.
- Evaluating formulas w.r.t. sets of assignments, teams.
- Can directly express independence, dependence, etc, via new atoms.
- Dependence logic = First-order logic + dependence atom.

Dependence atom

	x	y	z	w
s_0 :	3	2	0	3
s_1 :	1	2	1	0
s_2 :	0	1	0	0
s_3 :	2	1	1	0

Dependence atom

	x	y	z	w
s_0 :	3	2	0	3
s_1 :	1	2	1	0
s_2 :	0	1	0	0
s_3 :	2	1	1	0

$$\models \mathsf{dep}(yz, w)$$

Dependence atom

	x	y	z	w
s_0 :	3	2	0	3
s_1 :	1	2	1	0
s_2 :	0	1	0	0
s_3 :	2	1	1	0

$$\models \mathsf{dep}(yz, w)$$
$$\not\models \mathsf{dep}(z, w)$$

Dependence atom

	x	y	z	w
s_0 :	3	2	0	3
s_1 :	1	2	1	0
s_2 :	0	1	0	0
s_3 :	2	1	1	0

$$\models \mathrm{dep}(yz, w)$$
$$\not\models \mathrm{dep}(z, w)$$

X is a **team**, i.e., a set of assignments.

Dependence atom

	x	y	z	w
s_0 :	3	2	0	3
s_1 :	1	2	1	0
s_2 :	0	1	0	0
s_3 :	2	1	1	0

$$\models \mathsf{dep}(yz, w)$$
$$\not\models \mathsf{dep}(z, w)$$

X is a **team**, i.e., a set of assignments.

Definition

$M, X \models \mathsf{dep}(\bar{x}, y)$ iff for all $s, s' \in X$ if $s(\bar{x}) = s'(\bar{x})$ then $s(y) = s'(y)$.

Standard team semantics and flatness

- A formula is satisfied by a set of assignments, a team.

Standard team semantics and flatness

- A formula is satisfied by a set of assignments, a **team**.
- **Flatness:** A first-order formula is satisfied by a team iff all assignments satisfy the formula, i.e.,

$$X \models \varphi \text{ iff } \forall s \in X, s \models \varphi.$$

Standard team semantics and flatness

- A formula is satisfied by a set of assignments, a **team**.
- **Flatness:** A first-order formula is satisfied by a team iff all assignments satisfy the formula, i.e.,

$$X \models \varphi \text{ iff } \forall s \in X, s \models \varphi.$$

$X \models \varphi \wedge \psi$

Standard team semantics and flatness

- A formula is satisfied by a set of assignments, a **team**.
- **Flatness:** A first-order formula is satisfied by a team iff all assignments satisfy the formula, i.e.,

$$X \models \varphi \text{ iff } \forall s \in X, s \models \varphi.$$

$X \models \varphi \wedge \psi$ iff $\forall s \in X, s \models \varphi \wedge \psi$

Standard team semantics and flatness

- A formula is satisfied by a set of assignments, a **team**.
- **Flatness:** A first-order formula is satisfied by a team iff all assignments satisfy the formula, i.e.,

$$X \models \varphi \text{ iff } \forall s \in X, s \models \varphi.$$

$X \models \varphi \wedge \psi$ iff $\forall s \in X, s \models \varphi \wedge \psi$ iff $\forall s \in X, s \models \varphi$ and $s \models \psi$

Standard team semantics and flatness

- A formula is satisfied by a set of assignments, a **team**.
- **Flatness:** A first-order formula is satisfied by a team iff all assignments satisfy the formula, i.e.,

$$X \models \varphi \text{ iff } \forall s \in X, s \models \varphi.$$

$X \models \varphi \wedge \psi$ iff $\forall s \in X, s \models \varphi \wedge \psi$ iff $\forall s \in X, s \models \varphi$ and $s \models \psi$ iff $X \models \varphi$ and $X \models \psi$

Standard team semantics and flatness

- A formula is satisfied by a set of assignments, a **team**.
- **Flatness:** A first-order formula is satisfied by a team iff all assignments satisfy the formula, i.e.,

$$X \models \varphi \text{ iff } \forall s \in X, s \models \varphi.$$

$X \models \varphi \wedge \psi$ iff $X \models \varphi$ and $X \models \psi$

Standard team semantics and flatness

- A formula is satisfied by a set of assignments, a **team**.
- **Flatness:** A first-order formula is satisfied by a team iff all assignments satisfy the formula, i.e.,

$$X \models \varphi \text{ iff } \forall s \in X, s \models \varphi.$$

$X \models \varphi \wedge \psi$ iff $X \models \varphi$ and $X \models \psi$

$X \models \varphi \vee \psi$

Standard team semantics and flatness

- A formula is satisfied by a set of assignments, a **team**.
- **Flatness:** A first-order formula is satisfied by a team iff all assignments satisfy the formula, i.e.,

$$X \models \varphi \text{ iff } \forall s \in X, s \models \varphi.$$

$X \models \varphi \wedge \psi$ iff $X \models \varphi$ and $X \models \psi$

$X \models \varphi \vee \psi$ iff $\forall s \in X, s \models \varphi \vee \psi$

Standard team semantics and flatness

- A formula is satisfied by a set of assignments, a **team**.
- **Flatness:** A first-order formula is satisfied by a team iff all assignments satisfy the formula, i.e.,

$$X \models \varphi \text{ iff } \forall s \in X, s \models \varphi.$$

$X \models \varphi \wedge \psi$ iff $X \models \varphi$ and $X \models \psi$

$X \models \varphi \vee \psi$ iff $\forall s \in X, s \models \varphi \vee \psi$ iff $\forall s \in X, s \models \varphi$ or $s \models \psi$

Standard team semantics and flatness

- A formula is satisfied by a set of assignments, a **team**.
- **Flatness:** A first-order formula is satisfied by a team iff all assignments satisfy the formula, i.e.,

$$X \models \varphi \text{ iff } \forall s \in X, s \models \varphi.$$

$X \models \varphi \wedge \psi$ iff $X \models \varphi$ and $X \models \psi$

$X \models \varphi \vee \psi$ iff $\forall s \in X, s \models \varphi \vee \psi$ iff $\forall s \in X, s \models \varphi$ or $s \models \psi$ iff $\exists Y \cup Z = X, Y \models \varphi$ and $Z \models \psi$

Standard team semantics and flatness

- A formula is satisfied by a set of assignments, a **team**.
- **Flatness:** A first-order formula is satisfied by a team iff all assignments satisfy the formula, i.e.,

$$X \models \varphi \text{ iff } \forall s \in X, s \models \varphi.$$

$X \models \varphi \wedge \psi$ iff $X \models \varphi$ and $X \models \psi$

$X \models \varphi \vee \psi$ iff $\exists Y \cup Z = X, Y \models \varphi$ and $Z \models \psi$

Dependence Logic

$\varphi ::= P(\bar{x}) \mid \neg P(\bar{x}) \mid \mathbf{dep}(\bar{x}, y) \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \exists x\, \varphi \mid \forall x\, \varphi$

Dependence Logic

$\varphi ::= P(\bar{x}) \mid \neg P(\bar{x}) \mid \mathbf{dep}(\bar{x}, y) \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \exists x\, \varphi \mid \forall x\, \varphi$

- $M, X \models \gamma$ if for all $s \in X$: $M, s \models \gamma$, where γ is a literal.

Dependence Logic

$\varphi ::= P(\bar{x}) \mid \neg P(\bar{x}) \mid \mathbf{dep}(\bar{x}, y) \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \exists x \varphi \mid \forall x \varphi$

- $M, X \models \gamma$ if for all $s \in X$: $M, s \models \gamma$, where γ is a literal.
- $M, X \models \mathbf{dep}(\bar{x}, y)$ if for all $s, s' \in X$ if $s(\bar{x}) = s'(\bar{x})$ then $s(y) = s'(y)$.

Dependence Logic

$\varphi ::= P(\bar{x}) \mid \neg P(\bar{x}) \mid \mathbf{dep}(\bar{x}, y) \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \exists x\, \varphi \mid \forall x\, \varphi$

- $M, X \models \gamma$ if for all $s \in X$: $M, s \models \gamma$, where γ is a literal.
- $M, X \models \mathbf{dep}(\bar{x}, y)$ if for all $s, s' \in X$ if $s(\bar{x}) = s'(\bar{x})$ then $s(y) = s'(y)$.
- $M, X \models \varphi \wedge \psi$ if $M, X \models \varphi$ and $M, X \models \psi$.

Dependence Logic

$\varphi ::= P(\bar{x}) \mid \neg P(\bar{x}) \mid \mathbf{dep}(\bar{x}, y) \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \exists x\, \varphi \mid \forall x\, \varphi$

- $M, X \models \gamma$ if for all $s \in X$: $M, s \models \gamma$, where γ is a literal.
- $M, X \models \mathbf{dep}(\bar{x}, y)$ if for all $s, s' \in X$ if $s(\bar{x}) = s'(\bar{x})$ then $s(y) = s'(y)$.
- $M, X \models \varphi \wedge \psi$ if $M, X \models \varphi$ and $M, X \models \psi$.
- $M, X \models \varphi \vee \psi$ if $\exists Y, Z$ such that $X = Y \cup Z$, $M, Y \models \varphi$ and $M, Z \models \psi$.

Dependence Logic

$\varphi ::= P(\bar{x}) \mid \neg P(\bar{x}) \mid \mathbf{dep}(\bar{x}, y) \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \exists x\, \varphi \mid \forall x\, \varphi$

- $M, X \models \gamma$ if for all $s \in X$: $M, s \models \gamma$, where γ is a literal.
- $M, X \models \mathbf{dep}(\bar{x}, y)$ if for all $s, s' \in X$ if $s(\bar{x}) = s'(\bar{x})$ then $s(y) = s'(y)$.
- $M, X \models \varphi \wedge \psi$ if $M, X \models \varphi$ and $M, X \models \psi$.
- $M, X \models \varphi \vee \psi$ if $\exists Y, Z$ such that $X = Y \cup Z$, $M, Y \models \varphi$ and $M, Z \models \psi$.
- $M, X \models Qx\, \varphi$ if ..

Quantifiers in dependence logic

- $M, X \models Qx\varphi$ iff there is $F : X \to Q_M$ such that $M, X[F/x] \models \varphi$.

Quantifiers in dependence logic

- $M, X \models Qx\varphi$ iff there is $F : X \to Q_M$ such that $M, X[F/x] \models \varphi$.

$\forall_M = \{ M \}$,

Quantifiers in dependence logic

- $M, X \models Qx\varphi$ iff there is $F : X \to Q_M$ such that $M, X[F/x] \models \varphi$.

$\forall_M = \{ M \}$, $\exists_M = \{ A \subseteq M \mid A \neq \emptyset \}$,

Quantifiers in dependence logic

- $M, X \models Qx\varphi$ iff there is $F : X \to Q_M$ such that $M, X[F/x] \models \varphi$.

$\forall_M = \{ M \}$, $\exists_M = \{ A \subseteq M \mid A \neq \emptyset \}$, $X[F/x] = \{ s[a/x] \mid s \in X, a \in F(s) \}$

Quantifiers in dependence logic

- $M, X \models Qx\varphi$ iff there is $F : X \to Q_M$ such that $M, X[F/x] \models \varphi$.

$\forall_M = \{ M \}$, $\exists_M = \{ A \subseteq M \mid A \neq \emptyset \}$, $X[F/x] = \{ s[a/x] \mid s \in X, a \in F(s) \}$

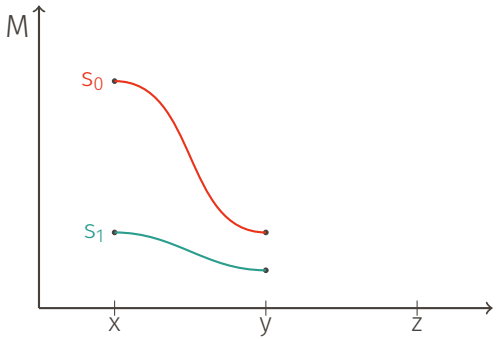
Example: $M, \{s_0, s_1\} \models \exists z\, R(x, y, z)$

Quantifiers in dependence logic

- $M, X \models Qx\varphi$ iff there is $F : X \to Q_M$ such that $M, X[F/x] \models \varphi$.

$\forall_M = \{ M \}$, $\exists_M = \{ A \subseteq M \mid A \neq \emptyset \}$, $X[F/x] = \{ s[a/x] \mid s \in X, a \in F(s) \}$

Example: $M, \{s_0, s_1\} \models \exists z\, R(x, y, z)$

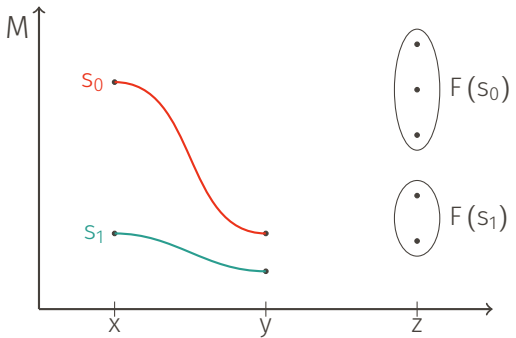


Quantifiers in dependence logic

- $M, X \models Qx\varphi$ iff there is $F : X \to Q_M$ such that $M, X[F/x] \models \varphi$.

$\forall_M = \{ M \}$, $\exists_M = \{ A \subseteq M \mid A \neq \emptyset \}$, $X[F/x] = \{ s[a/x] \mid s \in X, a \in F(s) \}$

Example: $M, \{s_0, s_1\} \models \exists z\, R(x, y, z)$

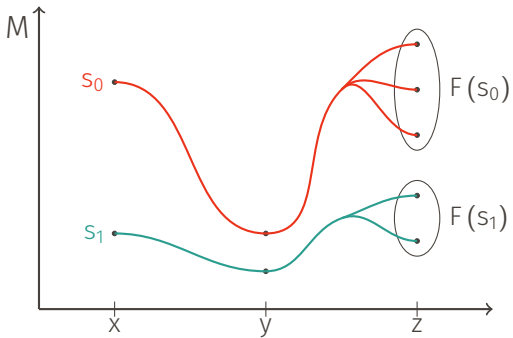


Quantifiers in dependence logic

- $M, X \models Qx\varphi$ iff there is $F : X \to Q_M$ such that $M, X[F/x] \models \varphi$.

$\forall_M = \{ M \}$, $\exists_M = \{ A \subseteq M \mid A \neq \emptyset \}$, $X[F/x] = \{ s[a/x] \mid s \in X, a \in F(s) \}$

Example: $M, \{s_0, s_1\} \models \exists z\, R(x, y, z)$



Some properties of Dependence logic

- Empty team property: $\mathsf{M}, \emptyset \models \varphi$

Some properties of Dependence logic

- **Empty team property:** $M, \emptyset \models \varphi$
- **Downwards closure:** If $Y \subseteq X$ and $M, X \models \varphi$ then $M, Y \models \varphi$.

Some properties of Dependence logic

- **Empty team property:** $M, \emptyset \models \varphi$
- **Downwards closure:** If $Y \subseteq X$ and $M, X \models \varphi$ then $M, Y \models \varphi$.
- **Flatness:** For FO-formulas: $M, X \models \varphi$ iff $M, s \models \varphi$ for all $s \in X$.

Some properties of Dependence logic

- **Empty team property:** $M, \emptyset \models \varphi$
- **Downwards closure:** If $Y \subseteq X$ and $M, X \models \varphi$ then $M, Y \models \varphi$.
- **Flatness:** For FO-formulas: $M, X \models \varphi$ iff $M, s \models \varphi$ for all $s \in X$.
- Dependence logic and Existential second-order logic (ESO/Σ^1_1) are equivalent.

Denotations and lifts

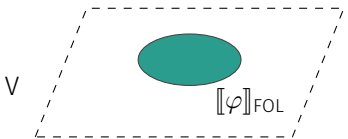
- $M, s \vDash \varphi$

Denotations and lifts

- $M, s \vDash \varphi$
- V is the set of all assignments.

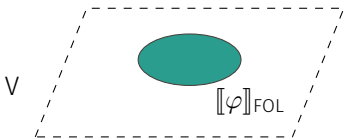
Denotations and lifts

- $M, s \vDash \varphi$
- V is the set of all assignments.
- $[\![\varphi]\!]_{FOL} = \{ s \in V \mid M, s \vDash \varphi \}$



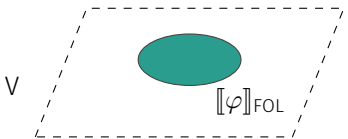
Denotations and lifts

- $M, s \vDash \varphi$
- V is the set of all assignments.
- $[\![\varphi]\!]_{FOL} = \{ s \in V \mid M, s \vDash \varphi \}$
- $[\![\varphi]\!]_{FOL} \in \mathcal{P}V$



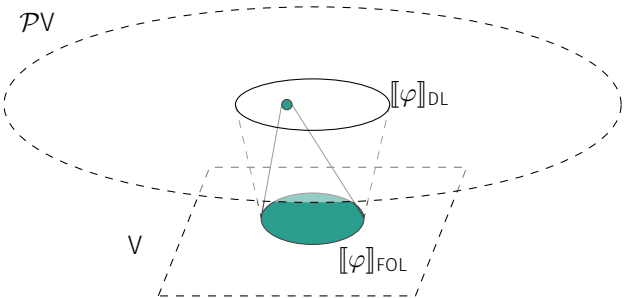
Denotations and lifts

- $M, s \vDash \varphi$
- V is the set of all assignments.
- $[\![\varphi]\!]_{FOL} = \{ s \in V \mid M, s \vDash \varphi \}$
- $[\![\varphi]\!]_{FOL} \in \mathcal{P}V$
- $M, X \vDash \varphi$



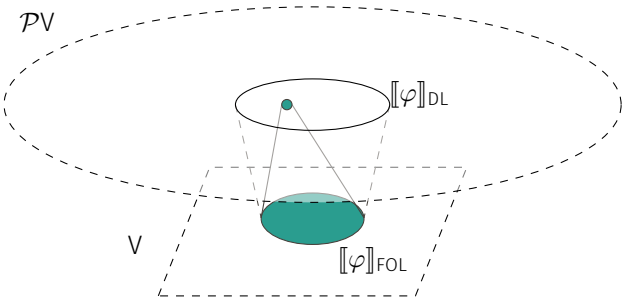
Denotations and lifts

- $M, s \vDash \varphi$
- V is the set of all assignments.
- $[\![\varphi]\!]_{FOL} = \{ s \in V \mid M, s \vDash \varphi \}$
- $[\![\varphi]\!]_{FOL} \in \mathcal{P}V$
- $M, X \vDash \varphi$
- $[\![\varphi]\!]_{DL} = \{ X \subseteq V \mid M, X \vDash \varphi \}$



Denotations and lifts

- $M, s \vDash \varphi$
- V is the set of all assignments.
- $[\![\varphi]\!]_{FOL} = \{ s \in V \mid M, s \vDash \varphi \}$
- $[\![\varphi]\!]_{FOL} \in \mathcal{P}V$
- $M, X \vDash \varphi$
- $[\![\varphi]\!]_{DL} = \{ X \subseteq V \mid M, X \vDash \varphi \}$
- $[\![\varphi]\!]_{DL} \in \mathcal{P}\mathcal{P}V$



Flatness

$$[\![\varphi]\!]_{DL} = \mathcal{P}([\![\varphi]\!]_{FOL})$$

Generalized quantifiers

- A generalized quantifier is a class of structures in a finite relational signature.

Generalized quantifiers

- A generalized quantifier is a class of structures in a finite relational signature.
- Example: Q_0, "There exists infinitely many", is the class of structures (M, P) where $P \subseteq M$ is infinite.

Generalized quantifiers

- A generalized quantifier is a class of structures in a finite relational signature.
- Example: Q_0, "There exists infinitely many", is the class of structures (M, P) where $P \subseteq M$ is infinite.

Barwise (-79)

Branching of generalized quantifiers occurs in natural languages.

Generalized quantifiers

- A **generalized quantifier** is a class of structures in a finite relational signature.
- Example: Q_0, "There exists infinitely many", is the class of structures (M, P) where $P \subseteq M$ is infinite.

Barwise (-79)

Branching of generalized quantifiers occurs in natural languages.

- Adding **monotone increasing** generalized quantifiers to Dependence logic can be done conservatively:

Generalized quantifiers

- A **generalized quantifier** is a class of structures in a finite relational signature.
- Example: Q_0, "There exists infinitely many", is the class of structures (M, P) where $P \subseteq M$ is infinite.

Barwise (-79)

Branching of generalized quantifiers occurs in natural languages.

- Adding **monotone increasing** generalized quantifiers to Dependence logic can be done conservatively:

$$M, X \models Qx\varphi \text{ iff there is } F : X \to Q_M \text{ such that } M, X[F/x] \models \varphi.$$

- $Q_M = \{ P \mid (M, P) \in Q \}$

Generalized quantifiers, continued

Observation

A generalized quantifier and its monotone closure gets the same truth condition.

Generalized quantifiers, continued

Observation

A generalized quantifier and its monotone closure gets the same truth condition.

Example: $\exists^{=\aleph_0}$ gets the same truth condition in Dependence logic as Q_0.

Generalized quantifiers, continued

Observation

A generalized quantifier and its monotone closure gets the same truth condition.

Example: $\exists^{=\aleph_0}$ gets the same truth condition in Dependence logic as Q_0.

- Alternative guiding principle: $[\![\varphi]\!]_{\mathrm{ALT}} = \{\, [\![\varphi]\!]_{\mathrm{FOL}} \,\}$.

Generalized quantifiers, continued

Observation

A generalized quantifier and its monotone closure gets the same truth condition.

Example: $\exists^{=\aleph_0}$ gets the same truth condition in Dependence logic as Q_0.

- Alternative guiding principle: $[\![\varphi]\!]_{\mathrm{ALT}} = \{\, [\![\varphi]\!]_{\mathrm{FOL}} \,\}$.

Closely related to 1-semantics of Nurmi (2009).

Generalized quantifiers, continued

Observation

A generalized quantifier and its monotone closure gets the same truth condition.

Example: $\exists^{=\aleph_0}$ gets the same truth condition in Dependence logic as Q_0.

- Alternative guiding principle: $[\![\varphi]\!]_{\mathrm{ALT}} = \{\, [\![\varphi]\!]_{\mathrm{FOL}} \,\}$.

Closely related to 1-semantics of Nurmi (2009).

Theorem (Engström (2024))

Using this alternative principle we can define a non-standard team semantics that can handle any generalized quantifiers.

Substitutionality

Substitutionality as a basic property of logic:

Bolzano (1837)

A proposition is universally valid if all variants are true.

Substitutionality

Substitutionality as a basic property of logic:

Bolzano (1837)

A proposition is universally valid if all variants are true.

Tarski (1936), property (F)

If, in the sentences of the class K and in the sentence X, the constants— apart from purely logical constants—are replaced by any other constants (like signs being everywhere replaced by like signs), and if we denote the class of sentences thus obtained from K by 'K'', and the sentence obtained from X by 'X'', then the sentence X′ must be true provided only that all sentences of the class K′ are true.

Dependence logic and substitutionality

Dependence logic and substitutionality

Substitutionality:

$$\varphi \vDash \psi \;\Rightarrow\; \mathsf{h}(\varphi) \vDash \mathsf{h}(\psi)$$

for all homomorphisms h from the absolutely free (term) algebra of formulas to itself.

Dependence logic and substitutionality

Substitutionality:

$$\varphi \vDash \psi \;\Rightarrow\; \mathsf{h}(\varphi) \vDash \mathsf{h}(\psi)$$

for all homomorphisms h from the absolutely free (term) algebra of formulas to itself.

- helps when constructing proof systems.

Dependence logic and substitutionality

Substitutionality:

$$\varphi \vDash \psi \;\Rightarrow\; \mathsf{h}(\varphi) \vDash \mathsf{h}(\psi)$$

for all homomorphisms h from the absolutely free (term) algebra of formulas to itself.

- helps when constructing proof systems.
- needed to apply (standard) techniques from algebraic logic.

Dependence logic and substitutionality

Substitutionality:

$$\varphi \vDash \psi \;\Rightarrow\; \mathsf{h}(\varphi) \vDash \mathsf{h}(\psi)$$

for all homomorphisms h from the absolutely free (term) algebra of formulas to itself.

- helps when constructing proof systems.
- needed to apply (standard) techniques from algebraic logic.

Dependence logic and its siblings are not substitutional:
$\mathsf{P}(x) \vee \mathsf{P}(x) \vDash \mathsf{P}(x)$, but $\mathsf{dep}(x) \vee \mathsf{dep}(x) \nvDash \mathsf{dep}(x)$

Dependence logic and substitutionality

Substitutionality:

$$\varphi \vDash \psi \;\Rightarrow\; \mathsf{h}(\varphi) \vDash \mathsf{h}(\psi)$$

for all homomorphisms h from the absolutely free (term) algebra of formulas to itself.

- helps when constructing proof systems.
- needed to apply (standard) techniques from algebraic logic.

Dependence logic and its siblings are not substitutional:
$\mathsf{P}(x) \vee \mathsf{P}(x) \vDash \mathsf{P}(x)$, but $\mathbf{dep}(x) \vee \mathbf{dep}(x) \nvDash \mathbf{dep}(x)$

Due to that the possible denotations of atoms is a proper subset of the possible denotations of formulas.

The project

Basic question

Is there a substitutional logic that can express dependence and independence?

The project

Basic question

Is there a substitutional logic that can express dependence and independence?

Guiding ideas for the project:

- Start with propositional logic, for simplicty and its strong connection to Boolean algebras.

The project

Basic question

Is there a substitutional logic that can express dependence and independence?

Guiding ideas for the project:

- Start with propositional logic, for simplicty and its strong connection to Boolean algebras.
- Make sure that the denotations of propositional variables are as general as possible.

The project

Basic question

Is there a substitutional logic that can express dependence and independence?

Guiding ideas for the project:

- Start with propositional logic, for simplicty and its strong connection to Boolean algebras.
- Make sure that the denotations of propositional variables are as general as possible.
- Include “enough” connectives to be able to express standard team semantics.

Propositional logic of teams

Boolean algebras and teams

Propositional logic is the logic of Boolean algebras

$$\varphi \models \psi \text{ iff } \mathsf{h}(\varphi) \leq \mathsf{h}(\psi).$$

for all Boolean algebras B and homomorphisms $\mathsf{h} : \mathtt{Fm} \to \mathsf{B}$.

Boolean algebras and teams

Propositional logic is the logic of Boolean algebras

$$\varphi \models \psi \text{ iff } \mathsf{h}(\varphi) \leq \mathsf{h}(\psi).$$

for all Boolean algebras B and homomorphisms $\mathsf{h} : \mathtt{Fm} \to \mathsf{B}$.

- Teams are elements of $\mathcal{P}(\mathsf{V})$

Boolean algebras and teams

Propositional logic is the logic of Boolean algebras

$$\varphi \models \psi \text{ iff } \mathsf{h}(\varphi) \leq \mathsf{h}(\psi).$$

for all Boolean algebras B and homomorphisms $\mathsf{h} : \mathtt{Fm} \to \mathsf{B}$.

- Teams are elements of $\mathcal{P}(\mathsf{V})$
- $\mathsf{B} = \mathcal{P}(\mathsf{V})$ is a Boolean algebra.

Boolean algebras and teams

Propositional logic is the logic of Boolean algebras

$$\varphi \models \psi \text{ iff } \mathsf{h}(\varphi) \leq \mathsf{h}(\psi).$$

for all Boolean algebras B and homomorphisms $\mathsf{h} : \mathtt{Fm} \to \mathsf{B}$.

- Teams are elements of $\mathcal{P}(\mathsf{V})$
- $\mathsf{B} = \mathcal{P}(\mathsf{V})$ is a Boolean algebra.
- Denotations of formulas are elements of $\mathcal{P}(\mathsf{B}) = \mathcal{P}(\mathcal{P}(\mathsf{V}))$.

Boolean algebras and teams

Propositional logic is the logic of Boolean algebras

$$\varphi \models \psi \text{ iff } \mathsf{h}(\varphi) \leq \mathsf{h}(\psi).$$

for all Boolean algebras B and homomorphisms $\mathsf{h} : \mathtt{Fm} \to \mathsf{B}$.

- Teams are elements of $\mathcal{P}(\mathsf{V})$
- $\mathsf{B} = \mathcal{P}(\mathsf{V})$ is a Boolean algebra.
- Denotations of formulas are elements of $\mathcal{P}(\mathsf{B}) = \mathcal{P}(\mathcal{P}(\mathsf{V}))$.
- Logical connectives correspond to operators on $\mathcal{P}(\mathsf{B})$.

Boolean algebras and teams

Propositional logic is the logic of Boolean algebras

$$\varphi \models \psi \text{ iff } \mathsf{h}(\varphi) \leq \mathsf{h}(\psi).$$

for all Boolean algebras B and homomorphisms $\mathsf{h} : \mathtt{Fm} \to \mathsf{B}$.

- Teams are elements of $\mathcal{P}(\mathsf{V})$
- $\mathsf{B} = \mathcal{P}(\mathsf{V})$ is a Boolean algebra.
- Denotations of formulas are elements of $\mathcal{P}(\mathsf{B}) = \mathcal{P}(\mathcal{P}(\mathsf{V}))$.
- Logical connectives correspond to operators on $\mathcal{P}(\mathsf{B})$.

Structures $\mathcal{P}(\mathsf{B})$ with point-wise operators have been studied under different names:

- Second-order Boolean algebras (Brink, 1984)
- Complex algebras (Goldblatt, 1989)
- Hyperboolean algebras (Goranko and Vakarelov, 1999)

Algebras for teams

$\mathcal{P}(B)$ is a Boolean algebra with additional point-wise operators.

- Standard (set-theoretic) Boolean operators: $\bot$, $\neg$, $\vee$, $\wedge$

Algebras for teams

$\mathcal{P}(B)$ is a Boolean algebra with additional point-wise operators.

- Standard (set-theoretic) Boolean operators: $\bot$, $\neg$, $\vee$, $\wedge$
- Internal (point-wise) Boolean operators: $\mathbin{⫫}$, $\multimap$, $\mathbin{⅋}$, $\mathbin{⫽}$

Algebras for teams

$\mathcal{P}(B)$ is a Boolean algebra with additional point-wise operators.

- Standard (set-theoretic) Boolean operators: $\bot$, $\neg$, $\vee$, $\wedge$
- Internal (point-wise) Boolean operators: $\mathbin{⫫}$, $\dashv$, $\mathbin{\mathbb{V}}$, $\mathbin{⩚}$

$$
\begin{aligned}
⫫ &= \{ \bot \}, \\
\dashv X &= \{ \neg a \mid a \in X \}, \\
X \mathbin{\mathbb{V}} Y &= \{ a \vee b \mid a \in X, b \in Y \}, \text{ and} \\
X \mathbin{⩚} Y &= \{ a \wedge b \mid a \in X, b \in Y \},
\end{aligned}
$$

where $X, Y \in \mathcal{P}(B)$.

Propositional Logic of Teams (LT)

Formulas of LT (**Fm**):

$$\varphi ::= \bot \mid P_i \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \perp\!\!\!\perp \mid \Box\varphi \mid \varphi \mathbin{\mathbb{V}} \varphi \mid \varphi \mathbin{/\!\!/\!\wedge} \varphi$$

Propositional Logic of Teams (LT)

Formulas of LT ($\mathtt{Fm}$):

$$\varphi ::= \bot \mid \mathrm{P}_i \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \perp\!\!\!\perp \mid \dashv\varphi \mid \varphi \mathbin{\curlyvee} \varphi \mid \varphi \mathbin{\curlywedge} \varphi$$

Definition

$$\varphi \vDash \psi \text{ iff}$$

for all Boolean algebras B and all homomorphisms $\mathrm{H} : \mathtt{Fm} \to \mathcal{P}(\mathrm{B})$: $\mathrm{H}(\varphi) \subseteq \mathrm{H}(\psi)$.

Propositional Logic of Teams (LT)

Formulas of LT ($\mathtt{Fm}$):

$$\varphi ::= \bot \mid \mathrm{P}_i \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \perp\!\!\!\perp \mid \dashv\varphi \mid \varphi \mathbin{\curlyvee} \varphi \mid \varphi \mathbin{/\!\!\wedge} \varphi$$

Definition

$$\varphi \vDash \psi \text{ iff}$$

for all Boolean algebras B and all homomorphisms $\mathrm{H} : \mathtt{Fm} \to \mathcal{P}(\mathrm{B})$: $\mathrm{H}(\varphi) \subseteq \mathrm{H}(\psi)$.

$\vDash \psi$ iff $\mathrm{H}(\psi) = \mathrm{B}$ for all $\mathrm{H} : \mathtt{Fm} \to \mathcal{P}(\mathrm{B})$.

Some results

- Substitutional.

Some results

- Substitutional. [Composition of two homomorphisms is a homomorphism.]

Some results

- Substitutional. [Composition of two homomorphisms is a homomorphism.]
- Conservative over propositional logic.

Some results

- Substitutional. [Composition of two homomorphisms is a homomorphism.]
- Conservative over propositional logic. [External connectives are defined over a Boolean algebra.]

Some results

- Substitutional. [Composition of two homomorphisms is a homomorphism.]
- Conservative over propositional logic. [External connectives are defined over a Boolean algebra.]
- $\not\models P \mathbin{\mathbb{V}} \neg P$ and also $\not\models P \mathbin{\mathbb{V}} \dashv P$

Some results

- Substitutional. [Composition of two homomorphisms is a homomorphism.]
- Conservative over propositional logic. [External connectives are defined over a Boolean algebra.]
- $\not\models P \mathbin{\mathbb{V}} \neg P$ and also $\not\models P \mathbin{\mathbb{V}} \dashv P$ $[H(P) = \{\bot\}]$

Some results

- Substitutional. [Composition of two homomorphisms is a homomorphism.]
- Conservative over propositional logic. [External connectives are defined over a Boolean algebra.]
- $\not\models P \mathbin{\mathbb{V}} \neg P$ and also $\not\models P \mathbin{\mathbb{V}} \dashv P$ $[H(P) = \{\bot\}]$

Theorem

- The set of countable and finite Boolean algebras is adequate for LT.

Some results

- Substitutional. [Composition of two homomorphisms is a homomorphism.]
- Conservative over propositional logic. [External connectives are defined over a Boolean algebra.]
- $\not\models P \mathbin{\mathbb{V}} \neg P$ and also $\not\models P \mathbin{\mathbb{V}} \exists P$ $[H(P) = \{\bot\}]$

Theorem

- The set of countable and finite Boolean algebras is adequate for LT.
- The set of finite Boolean algebras is not adequate.

Some results

- Substitutional. [Composition of two homomorphisms is a homomorphism.]
- Conservative over propositional logic. [External connectives are defined over a Boolean algebra.]
- $\not\models P \mathbin{\mathbb{V}} \neg P$ and also $\not\models P \mathbin{\mathbb{V}} \dashv P$ $[H(P) = \{\bot\}]$

Theorem

- The set of countable and finite Boolean algebras is adequate for LT.
- The set of finite Boolean algebras is not adequate.
- LT is undecidable (Knudstorp, unpublished).

Some results

- Substitutional. [Composition of two homomorphisms is a homomorphism.]
- Conservative over propositional logic. [External connectives are defined over a Boolean algebra.]
- $\not\models P \mathbin{\mathbb{V}} \neg P$ and also $\not\models P \mathbin{\mathbb{V}} \dashv P$ $[H(P) = \{\bot\}]$

Theorem

- The set of countable and finite Boolean algebras is adequate for LT.
- The set of finite Boolean algebras is not adequate.
- LT is undecidable (Knudstorp, unpublished).

LT is closely related to "Hyperboolean modal logic" (HBML) of Goranko and Vakarelov (1999).

Labelled formulas

Labels ($\mathbf{Lb}$) are $a ::= \bot \mid p_i \mid \neg a \mid a \vee a \mid a \wedge a$

Labelled formulas

Labels ($\mathtt{Lb}$) are $a ::= \bot \mid p_i \mid \neg a \mid a \vee a \mid a \wedge a$ $\qquad \boxed{a : \varphi} \qquad a \in \mathtt{Lb}, \varphi \in \mathtt{Fm}$

Labelled formulas

Labels ($\mathtt{Lb}$) are $a ::= \bot \mid p_i \mid \neg a \mid a \vee a \mid a \wedge a$ $\qquad \boxed{a : \varphi} \qquad a \in \mathtt{Lb}, \varphi \in \mathtt{Fm}$

Definition

$$a : \varphi \vDash b : \psi \text{ iff}$$

for all B, all $h : \mathtt{Lb} \to B$ and all $H : \mathtt{Fm} \to \mathcal{P}B$: if $h(a) \in H(\varphi)$ then $h(b) \in H(\psi)$.

Observation

$$\varphi \vDash \psi \quad \text{iff} \quad p : \varphi \vDash p : \psi$$

Labelled formulas

Labels ($\mathtt{Lb}$) are $a ::= \bot \mid p_i \mid \neg a \mid a \vee a \mid a \wedge a$ $\qquad \boxed{a : \varphi} \qquad a \in \mathtt{Lb}, \varphi \in \mathtt{Fm}$

Definition

$$a : \varphi \vDash b : \psi \text{ iff}$$

for all B, all $h : \mathtt{Lb} \to B$ and all $H : \mathtt{Fm} \to \mathcal{P}B$: if $h(a) \in H(\varphi)$ then $h(b) \in H(\psi)$.

Observation

$$\varphi \vDash \psi \quad \text{iff} \quad p : \varphi \vDash p : \psi$$

Rules for external connectives

$$\frac{\mathsf{a} : \varphi \quad \mathsf{a} : \psi}{\mathsf{a} : \varphi \wedge \psi}\ \wedge\mathsf{I} \qquad \frac{\mathsf{a} : \varphi \wedge \psi}{\mathsf{a} : \varphi}\ \wedge\mathsf{E} \qquad \frac{\mathsf{a} : \varphi \wedge \psi}{\mathsf{a} : \psi}\ \wedge\mathsf{E}$$

$$\frac{\mathsf{a} : \varphi}{\mathsf{a} : \varphi \vee \psi}\ \vee\mathsf{I} \qquad \frac{\mathsf{a} : \psi}{\mathsf{a} : \varphi \vee \psi}\ \vee\mathsf{I} \qquad \frac{\mathsf{a} : \varphi \vee \psi \qquad \begin{array}{c}[\mathsf{a} : \varphi] \\ \vdots \\ \mathsf{b} : \sigma\end{array} \qquad \begin{array}{c}[\mathsf{a} : \psi] \\ \vdots \\ \mathsf{b} : \sigma\end{array}}{\mathsf{b} : \sigma}\ \vee\mathsf{E}$$

$$\frac{\begin{array}{c}[\mathsf{a} : \varphi] \\ \vdots \\ \mathsf{b} : \bot\end{array}}{\mathsf{a} : \neg\varphi}\ \neg\mathsf{I} \qquad \frac{\mathsf{a} : \varphi \quad \mathsf{a} : \neg\varphi}{\mathsf{a} : \bot}\ \neg\mathsf{E} \qquad \frac{\begin{array}{c}[\mathsf{a} : \neg\varphi] \\ \vdots \\ \mathsf{b} : \bot\end{array}}{\mathsf{a} : \varphi}\ \mathsf{RAA} \qquad \frac{\mathsf{a} : \bot}{\mathsf{b} : \varphi}\ \bot\mathsf{E}$$

Rules for internal connectives

$a \equiv b$ is shorthand for $a \leftrightarrow b : \top$, and $\top = \neg\bot$

Rules for internal connectives

$a \equiv b$ is shorthand for $a \leftrightarrow b : \top$, and $\top = \neg\bot$

$$\frac{a : \varphi \quad b : \psi}{a \wedge b : \varphi \mathbin{/\!\!\wedge} \psi} \mathbin{/\!\!\wedge}\mathtt{I}$$

$$\frac{a : \varphi \quad b : \psi}{a \vee b : \varphi \mathbin{\backslash\!\!\vee} \psi} \mathbin{\backslash\!\!\vee}\mathtt{I}$$

Rules for internal connectives

$a \equiv b$ is shorthand for $a \leftrightarrow b : \top$, and $\top = \neg\bot$

$$\frac{a : \varphi \quad b : \psi}{a \wedge b : \varphi \mathbin{/\!\!\wedge} \psi}\ /\!\!\wedge\mathtt{I}$$

$$\frac{a : \varphi \quad b : \psi}{a \vee b : \varphi \mathbin{\backslash\!\!\vee} \psi}\ \backslash\!\!\vee\mathtt{I}$$

$$\frac{\begin{array}{c}[p : \varphi]\\ [q : \psi]\\ [a \equiv p \wedge q]\\ \vdots\\ a : \varphi \mathbin{/\!\!\wedge} \psi \qquad b : \sigma\end{array}}{b : \sigma}\ /\!\!\wedge\mathtt{E}$$

$$\frac{\begin{array}{c}[p : \varphi]\\ [q : \psi]\\ [a \equiv p \vee q]\\ \vdots\\ a : \varphi \mathbin{\backslash\!\!\vee} \psi \qquad b : \sigma\end{array}}{b : \sigma}\ \backslash\!\!\vee\mathtt{E}$$

Rules for internal connectives

$a \equiv b$ is shorthand for $a \leftrightarrow b : \top$, and $\top = \dashv\bot$

$$\frac{a : \varphi \quad b : \psi}{a \wedge b : \varphi \mathbin{/\!\!\wedge} \psi}\ /\!\!\wedge\mathtt{I}$$

$$\frac{a : \varphi \quad b : \psi}{a \vee b : \varphi \mathbin{\backslash\!\!\vee} \psi}\ \backslash\!\!\vee\mathtt{I}$$

$$\frac{\begin{array}{c}[p : \varphi]\\ [q : \psi]\\ [a \equiv p \wedge q]\\ \vdots\\ a : \varphi \mathbin{/\!\!\wedge} \psi \qquad b : \sigma\end{array}}{b : \sigma}\ /\!\!\wedge\mathtt{E}$$

$$\frac{\begin{array}{c}[p : \varphi]\\ [q : \psi]\\ [a \equiv p \vee q]\\ \vdots\\ a : \varphi \mathbin{\backslash\!\!\vee} \psi \qquad b : \sigma\end{array}}{b : \sigma}\ \backslash\!\!\vee\mathtt{E}$$

$$\frac{a : \varphi}{\neg a : \dashv\varphi}\ \dashv\mathtt{I} \qquad \frac{a : \dashv\varphi}{\neg a : \varphi}\ \dashv\mathtt{E}$$

Rules for labels

$$\frac{a_1 : \mathbb{T} \quad \cdots \quad a_k : \mathbb{T}}{b : \mathbb{T}}\ \texttt{taut}$$

Assuming $a_1, \ldots, a_k \vdash b$ in propositional logic, i.e., that $a_1 \wedge \ldots \wedge a_k \to b$ is a tautology.

Rules for labels

$$\frac{a_1 : \top \quad \cdots \quad a_k : \top}{b : \top}\ \texttt{taut}$$

Assuming $a_1, \ldots, a_k \vdash b$ in propositional logic, i.e., that $a_1 \wedge \ldots \wedge a_k \to b$ is a tautology.

$$\frac{a \equiv b \quad b : \varphi}{a : \varphi}\ \texttt{sub}$$

Soundness and completeness

$$a : \varphi \vDash b : \psi \quad \text{iff} \quad a : \varphi \vdash b : \psi$$

Strong propositional team logic: PT^+

Yang and Väänänen (2017)

$$\varphi ::= \mathrm{P}_i \mid {\sim}\mathrm{P}_i \mid \perp\!\!\!\perp \mid \mathrm{NB} \mid \varphi \mathbin{\mathbb{V}} \varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi$$

Strong propositional team logic: PT^+

Yang and Väänänen (2017)

$$\varphi ::= \mathrm{P}_i \mid {\sim}\mathrm{P}_i \mid \perp\!\!\!\perp \mid \mathrm{NB} \mid \varphi \mathbin{\mathbb{V}} \varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi$$

Valuations are elements of $2^{\mathbb{N}}$.

Strong propositional team logic: PT^+

Yang and Väänänen (2017)

$$\varphi ::= \mathrm{P}_i \mid {\sim}\mathrm{P}_i \mid \perp\!\!\!\perp \mid \mathrm{NB} \mid \varphi \mathbin{\mathbb{V}} \varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi$$

Valuations are elements of $2^{\mathbb{N}}$. Teams are elements of $\mathcal{P}(2^{\mathbb{N}})$.

Strong propositional team logic: PT^+

Yang and Väänänen (2017)

$$\varphi ::= \mathrm{P}_i \mid {\sim}\mathrm{P}_i \mid \perp\!\!\!\perp \mid \mathrm{NB} \mid \varphi \mathbin{\mathbb{V}} \varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi$$

Valuations are elements of $2^{\mathbb{N}}$. Teams are elements of $\mathcal{P}(2^{\mathbb{N}})$.

Define $\mathrm{H}_V : \mathtt{Fm}_{\mathrm{PT}^+} \to \mathcal{P}\mathcal{P}(2^{\mathbb{N}})$ by

Strong propositional team logic: PT^+

Yang and Väänänen (2017)

$$\varphi ::= P_i \mid {\sim}P_i \mid \bot\!\!\!\bot \mid \mathrm{NB} \mid \varphi \mathbin{\mathbb{V}} \varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi$$

Valuations are elements of $2^{\mathbb{N}}$. Teams are elements of $\mathcal{P}(2^{\mathbb{N}})$.

Define $H_V : \mathtt{Fm}_{\mathrm{PT}^+} \to \mathcal{P}\mathcal{P}(2^{\mathbb{N}})$ by

$$H_V(P_i) = \{\, a \in \mathcal{P}(2^{\mathbb{N}}) \mid \forall v \in a, v(i) = 1 \,\}$$
$$H_V({\sim}P_i) = \{\, a \in \mathcal{P}(2^{\mathbb{N}}) \mid \forall v \in a, v(i) = 0 \,\}$$

Strong propositional team logic: PT^+

Yang and Väänänen (2017)

$$\varphi ::= \mathrm{P}_i \mid {\sim}\mathrm{P}_i \mid \perp\!\!\!\perp \mid \mathrm{NB} \mid \varphi \mathbin{\mathbb{V}} \varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi$$

Valuations are elements of $2^{\mathbb{N}}$. Teams are elements of $\mathcal{P}(2^{\mathbb{N}})$.

Define $\mathrm{H}_V : \mathtt{Fm}_{\mathrm{PT}^+} \to \mathcal{P}\mathcal{P}(2^{\mathbb{N}})$ by

$$\mathrm{H}_V(\mathrm{P}_i) = \{\, a \in \mathcal{P}(2^{\mathbb{N}}) \mid \forall v \in a, v(i) = 1 \,\}$$
$$\mathrm{H}_V({\sim}\mathrm{P}_i) = \{\, a \in \mathcal{P}(2^{\mathbb{N}}) \mid \forall v \in a, v(i) = 0 \,\}$$

$\mathrm{PT}^+ : \varphi \vDash \psi \quad$ iff $\quad \mathrm{H}_V(\varphi) \subseteq \mathrm{H}_V(\psi)$

Strong propositional team logic: PT^+

Yang and Väänänen (2017)

$$\varphi ::= \mathrm{P}_i \mid {\sim}\mathrm{P}_i \mid \perp\!\!\!\perp \mid \mathrm{NB} \mid \varphi \mathbin{\mathbb{V}} \varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi$$

Valuations are elements of $2^{\mathbb{N}}$. Teams are elements of $\mathcal{P}(2^{\mathbb{N}})$.

Define $\mathrm{H}_V : \mathsf{Fm}_{\mathrm{PT}^+} \to \mathcal{P}\mathcal{P}(2^{\mathbb{N}})$ by

$$\mathrm{H}_V(\mathrm{P}_i) = \{\, a \in \mathcal{P}(2^{\mathbb{N}}) \mid \forall v \in a, v(i) = 1 \,\}$$
$$\mathrm{H}_V({\sim}\mathrm{P}_i) = \{\, a \in \mathcal{P}(2^{\mathbb{N}}) \mid \forall v \in a, v(i) = 0 \,\}$$

$\mathrm{PT}^+ : \varphi \vDash \psi \quad$ iff $\quad \mathrm{H}_V(\varphi) \subseteq \mathrm{H}_V(\psi)$

Note that $\mathrm{H}_V(\mathrm{P}_i)$ are principal ideals.

Some definable operators in LT

- $\mathsf{NB} = \neg\perp\!\!\!\perp$ $\qquad B \setminus \{\perp\}$

Some definable operators in LT

- $\mathsf{NB} = \neg\!\perp\!\!\!\perp$ $\qquad B \setminus \{\bot\}$
- $\top\!\!\top = \dashv\!\perp\!\!\!\perp$ $\qquad \{\top\}$

Some definable operators in LT

- $\mathrm{NB} = \neg\perp\!\!\!\perp$ $\quad B \setminus \{ \bot \}$
- $\mathbb{T} = \neg\perp\!\!\!\perp$ $\quad \{ \top \}$
- $\downarrow\varphi = \varphi \mathbin{/\!\!/\!\wedge} \top$, $\top = \neg\bot$ $\quad \downarrow X = \{ b \in B \mid \exists a \in X, b \leq a \}$

Some definable operators in LT

- $\mathrm{NB} = \neg\perp\!\!\!\perp$
- $\mathbb{T} = \dashv\perp\!\!\!\perp$
- $\downarrow\varphi = \varphi \mathbin{/\!\!/\!\!\wedge} \top$, $\top = \neg\bot$
- $\uparrow\varphi = \varphi \mathbin{\backslash\!\!/\!\!\vee} \top$

$B \setminus \{ \bot \}$

$\{ \top \}$

$\downarrow X = \{ b \in B \mid \exists a \in X, b \leq a \}$

$\uparrow X = \{ b \in B \mid \exists a \in X, a \leq b \}$

Some definable operators in LT

- $\mathrm{NB} = \neg\perp\!\!\!\perp$ — $B \setminus \{\bot\}$
- $\top\!\!\!\top = \dashv\perp\!\!\!\perp$ — $\{\top\}$
- $\downarrow\varphi = \varphi \mathbin{/\!\!/\!\!\wedge} \top$, $\top = \neg\bot$ — $\downarrow X = \{\, b \in B \mid \exists a \in X, b \leq a \,\}$
- $\uparrow\varphi = \varphi \mathbin{\backslash\!\!/\!\!\vee} \top$ — $\uparrow X = \{\, b \in B \mid \exists a \in X, a \leq b \,\}$
- $\Diamond\varphi = \uparrow\downarrow\varphi$ — $\Diamond X = B$ if $X \neq \emptyset$; and $\Diamond\emptyset = \emptyset$

Some definable operators in LT

- $\mathrm{NB} = \neg\!\perp\!\!\!\perp$
- $\mathbin{\top\!\!\!\top} = \dashv\!\perp\!\!\!\perp$
- $\downarrow\varphi = \varphi \mathbin{/\!\!/\!\!\wedge} \top$, $\top = \neg\bot$
- $\uparrow\varphi = \varphi \mathbin{\backslash\!\!\backslash\!\!\vee} \top$
- $\Diamond\varphi = \uparrow\downarrow\varphi$
- $\Box\varphi = \neg\Diamond\neg\varphi$

$B \setminus \{ \bot \}$

$\{ \top \}$

$\downarrow X = \{ b \in B \mid \exists a \in X, b \leq a \}$

$\uparrow X = \{ b \in B \mid \exists a \in X, a \leq b \}$

$\Diamond X = B$ if $X \neq \emptyset$; and $\Diamond\emptyset = \emptyset$

$\Box X = \emptyset$ if $X \neq B$; and $\Box B = B$

Some definable operators in LT

- $\mathsf{NB} = \neg\!\perp\!\!\!\perp$ — $B \setminus \{\bot\}$
- $\mathbb{T} = \dashv\!\perp\!\!\!\perp$ — $\{\top\}$
- $\downarrow\varphi = \varphi \mathbin{/\!\!/} \top$, $\top = \neg\bot$ — $\downarrow X = \{\, b \in B \mid \exists a \in X, b \leq a \,\}$
- $\uparrow\varphi = \varphi \mathbin{\backslash\!\!\backslash} \top$ — $\uparrow X = \{\, b \in B \mid \exists a \in X, a \leq b \,\}$
- $\Diamond\varphi = \uparrow\downarrow\varphi$ — $\Diamond X = B$ if $X \neq \emptyset$; and $\Diamond\emptyset = \emptyset$
- $\Box\varphi = \neg\Diamond\neg\varphi$ — $\Box X = \emptyset$ if $X \neq B$; and $\Box B = B$

Observation

$\Box\varphi \vDash \psi$ iff for all $H : \mathsf{Fm} \to \mathcal{P}(B)$, if $H(\varphi) = B$ then $H(\psi) = B$.

- $\sim\varphi = \neg\uparrow(\downarrow\varphi \wedge \neg\!\perp\!\!\!\perp)$ — $\sim X = \{\, b \in B \mid \forall a \in X, b \wedge a = \bot \,\}$

Note that $\not\vDash P \mathbin{\backslash\!\!\backslash} \sim P$.

Axiomatizing PT^+ in LT

Theorem

$X \mathbin{\mathbb{V}} {\sim}X = B$ iff X is a principal ideal.

Axiomatizing PT^+ in LT

Theorem

$X \mathbin{\mathbb{V}} {\sim}X = B$ iff X is a principal ideal.

Let $\mathrm{PVA} = \{\, \Box(P_i \mathbin{\mathbb{V}} {\sim}P_i) \mid i \in \mathbb{N} \,\}$.

Axiomatizing PT^+ in LT

Theorem

$X \mathbin{\mathbb{V}} {\sim}X = B$ iff X is a principal ideal.

Let $\mathrm{PVA} = \{ \Box(P_i \mathbin{\mathbb{V}} {\sim}P_i) \mid i \in \mathbb{N} \}$.

Theorem

$\mathrm{PT}^+ : \varphi \vDash \psi \quad$ iff $\quad \mathrm{LT} : \mathrm{PVA}, \varphi \vDash \psi.$

Conclusion

There is a substitutional propositional logic with a complete labelled natural deduction system that can express everything that standard propositional dependence logics can.

Further work

Further work

- What are the algebras of LT? In particular, are there algebras that satisfy all LT-equations but is not isomorphic to a substructure of a $\mathcal{P}(B)$?

Further work

- What are the algebras of LT? In particular, are there algebras that satisfy all LT-equations but is not isomorphic to a substructure of a $\mathcal{P}(\mathrm{B})$?
- The proof system can be seen as propositional logic “on top” of propositional logic. Are similar constructions applicable to other logics?

Further work

- What are the algebras of LT? In particular, are there algebras that satisfy all LT-equations but is not isomorphic to a substructure of a $\mathcal{P}(\mathrm{B})$?
- The proof system can be seen as propositional logic “on top” of propositional logic. Are similar constructions applicable to other logics?
- Specifically, what is propositional logic “on top” of first-order logic?

Further work: First-order logic

FOL is not as easily algebraisable as propositional logic is.

Further work: First-order logic

FOL is not as easily algebraisable as propositional logic is.

$$\varphi ::= P(\bar{x}) \mid x = y \mid x \neq y \mid \neg\varphi \mid \ni\varphi \mid \varphi \between \varphi \mid \varphi \barwedge \varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \exists x\, \varphi \mid \forall x\, \varphi \mid \exists^1 x\, \varphi \mid \forall^1 x\, \varphi,$$

Further work: First-order logic

FOL is not as easily algebraisable as propositional logic is.

$$\varphi ::= P(\bar{x}) \mid x = y \mid x \neq y \mid \neg\varphi \mid \ni\varphi \mid \varphi \mathbin{\mathscr{\wedge}} \varphi \mid \varphi \mathbin{\mathscr{\vee}} \varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \exists x\, \varphi \mid \forall x\, \varphi \mid \exists^1 x\, \varphi \mid \forall^1 x\, \varphi,$$

Structures

A structure is a pair of a non-empty domain M together with an interpretation I such that $I(P) \subseteq \mathcal{P}(M^k)$, where P is a k-ary predicate symbol.

Further work: First-order logic

FOL is not as easily algebraisable as propositional logic is.

$$\varphi ::= P(\bar{x}) \mid x = y \mid x \neq y \mid \neg\varphi \mid \dashv\varphi \mid \varphi \mathbin{/\!\!/} \varphi \mid \varphi \mathbin{\backslash\!\!\backslash} \varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \exists x\, \varphi \mid \forall x\, \varphi \mid \exists^1 x\, \varphi \mid \forall^1 x\, \varphi,$$

Structures

A structure is a pair of a non-empty domain M together with an interpretation I such that $I(P) \subseteq \mathcal{P}(M^k)$, where P is a k-ary predicate symbol.

It seems that $\mathrm{PVA} = \{\, \Box(P_i \mathbin{\backslash\!\!\backslash} {\sim}P_i) \mid i \in \mathbb{N} \,\}$ axiomatizes standard team semantics in this setting as well.

THANKS!

Chris Brink. Second-order boolean algebras. Quaestiones Mathematicae, 7(2):93–100, 1984.

Fredrik Engström and Orvar Lorimer Olsson. The propositional logic of teams. preprint arXiv:2303.14022, 2023.

Fredrik Engström. Generalized quantifiers using team semantics. 2024.

Robert Goldblatt. Varieties of complex algebras. Annals of pure and applied logic, 44(3): 173–242, 1989.

Valentin Goranko and Dimiter Vakarelov. Hyperboolean algebras and hyperboolean modal logic. Journal of Applied Non-classical Logics, 9(2-3):345–368, 1999.

Ville Nurmi. Dependence logic: Investigations into higher-order semantics defined on teams. PhD thesis, 2009.

Davide Emilio Quadrellaro. Algebraic semantics of intuitionistic inquisitive and dependence logic. Short Papers Advances in Modal Logic AiML 2020, page 75, 2020.

Fan Yang and Jouko Väänänen. Propositional team logics. Annals of Pure and Applied Logic, 168(7):1406–1441, 2017.