On dependence and logic

Swedish Congress of Philosophy, Stockholm

Fredrik Engström

June 16, 2013

$$\forall x\, \exists y\, Rxy$$

$$\begin{array}{c}\forall x \\ \downarrow \\ \exists y\end{array}$$

$$\forall x \exists y \forall z \exists w Rxyzw$$

$$\begin{array}{c} \forall x \\ \downarrow \\ \exists y \\ \downarrow \\ \forall z \\ \downarrow \\ \exists w \end{array}$$

$$\forall x \exists y \forall z \exists w \, Rxyzw$$

$$\downarrow$$

$$\forall z$$

$$\downarrow$$

$$\exists w$$

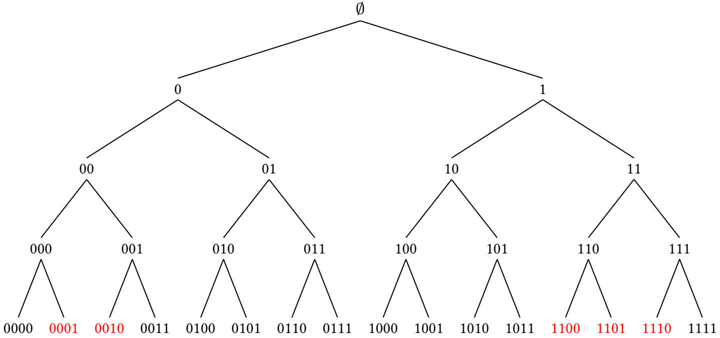
$$
\begin{array}{cc}
\forall x & \forall z \\
\downarrow & \downarrow \\
\exists y & \exists w
\end{array}
$$

$$\begin{pmatrix} \forall x \, \exists y \\ \forall z \, \exists w \end{pmatrix} \; Rxyzw$$

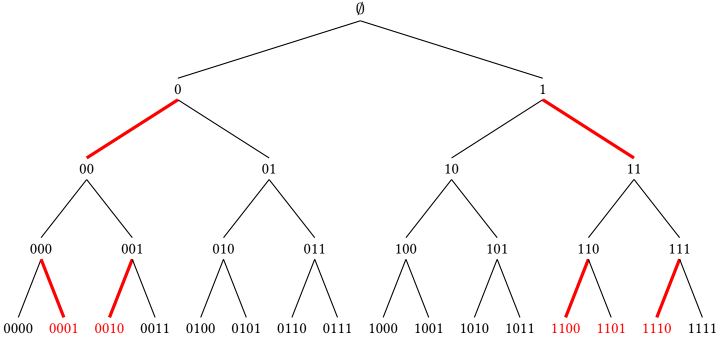
$$\begin{array}{cc} \forall x & \forall z \\ \downarrow & \downarrow \\ \exists y & \exists w \end{array}$$

Domain $\{0, 1\}$. $\forall x \exists y \forall z \exists w Rxyzw$

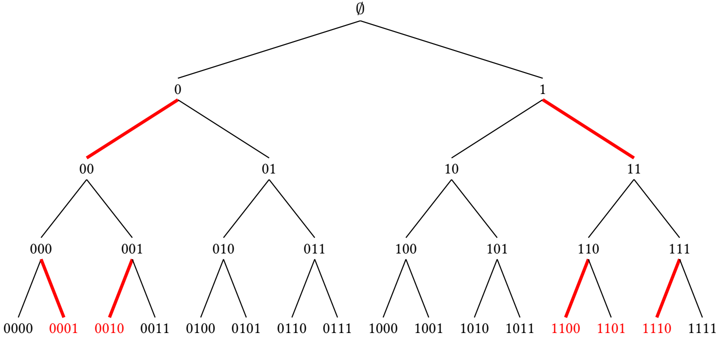
Domain $\{0, 1\}$. $\forall x \exists y \forall z \exists w Rxyzw$



Domain $\{ 0, 1 \}$. $\forall x \exists y \forall z \exists w\, Rxyzw$

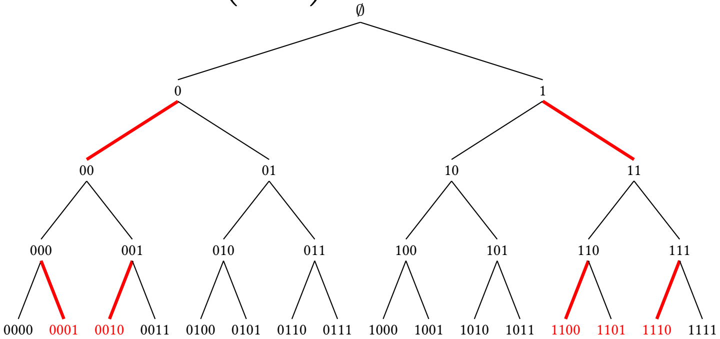


Domain $\{0, 1\}$. $\forall x \exists y \forall z \exists w Rxyzw$



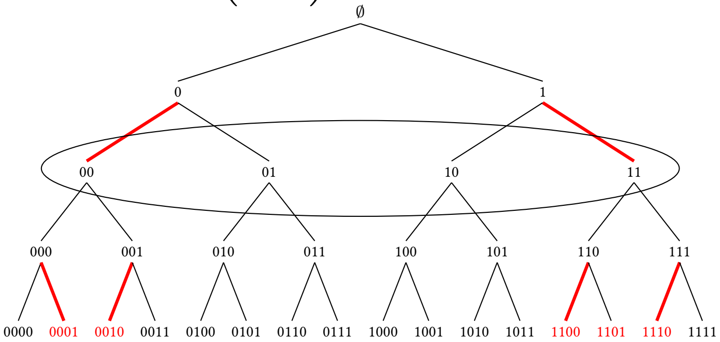
x	y	z	w
0	0	0	1
0	0	1	0
1	1	0	0
1	1	1	0

Domain $\{0, 1\}$. $\begin{pmatrix} \forall x \exists y \\ \forall z \exists w \end{pmatrix} Rxyzw$



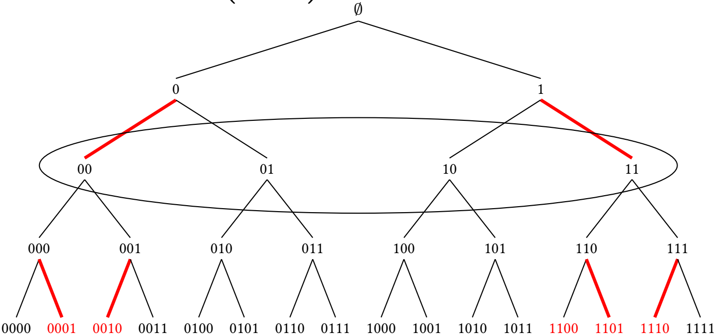
x	y	z	w
0	0	0	1
0	0	1	0
1	1	0	0
1	1	1	0

Domain $\{ 0, 1 \}$. $\begin{pmatrix} \forall x \exists y \\ \forall z \exists w \end{pmatrix} Rxyzw$



x	y	z	w
0	0	0	1
0	0	1	0
1	1	0	0
1	1	1	0

Domain $\{ 0, 1 \}$. $\begin{pmatrix} \forall x \exists y \\ \forall z \exists w \end{pmatrix} Rxyzw$



x	y	z	w
0	0	0	1
0	0	1	0
1	1	0	1
1	1	1	0

Dependence

x	y	z	w
0	0	0	1
0	0	1	0
1	1	0	0
1	1	1	0

$$\not\models D(z, w)$$

x	y	z	w
0	0	0	1
0	0	1	0
1	1	0	0
1	1	1	0

$$\not\models D(z, w)$$

x	y	z	w
0	0	0	1
0	0	1	0
1	1	0	1
1	1	1	0

$$\models D(z, w)$$

x	y	z	w
0	0	0	1
0	0	1	0
1	1	0	0
1	1	1	0

$$\not\models D(z, w)$$

x	y	z	w
0	0	0	1
0	0	1	0
1	1	0	1
1	1	1	0

$$\models D(z, w)$$

$$\begin{pmatrix} \forall x \exists y \\ \forall z \exists w \end{pmatrix} Rxyzw \equiv \forall x \exists y \forall z \exists w \left(D(z, w) \wedge Rxyzw\right)$$

Take home messages

Dependence is a property of strategies.

strategy $\approx$ set of assignments = **teams**.

Take home messages

Dependence is a property of strategies.

strategy ≈ set of assignments = **teams**.

Definition

X a **team**.

$M, X \vDash \mathrm{D}(\bar{x}, y)$ iff for all $s, s' \in X$ if $s(\bar{x}) = s'(\bar{x})$ then $s(y) = s'(y)$.

Dependence logic

- Syntax: FOL + $D(\bar{x}, y)$ in negation normal form
- X is a **team**, i.e., a set of assignments.
- $M, X \models (\neg)R\bar{x}$ iff for all $s \in X$: $M, s \models (\neg)R\bar{x}$.
- $M, X \models \varphi \wedge \psi$ iff $M, X \models \varphi$ and $M, X \models \psi$.

Dependence logic

- Syntax: FOL + $D(\bar{x}, y)$ in negation normal form
- X is a **team**, i.e., a set of assignments.
- $M, X \models (\neg)R\bar{x}$ iff for all $s \in X$: $M, s \models (\neg)R\bar{x}$.
- $M, X \models \varphi \wedge \psi$ iff $M, X \models \varphi$ and $M, X \models \psi$.
- $M, X \models \varphi \vee \psi$ iff there are Y and Z such that $M, Y \models \varphi$ and $M, Z \models \psi$ and $X = Y \cup Z$.

Dependence logic

- Syntax: FOL + $D(\bar{x}, y)$ in negation normal form
- X is a **team**, i.e., a set of assignments.
- $M, X \models (\neg)R\bar{x}$ iff for all $s \in X$: $M, s \models (\neg)R\bar{x}$.
- $M, X \models \varphi \wedge \psi$ iff $M, X \models \varphi$ and $M, X \models \psi$.
- $M, X \models \varphi \vee \psi$ iff there are Y and Z such that $M, Y \models \varphi$ and $M, Z \models \psi$ and $X = Y \cup Z$.
- $M, X \models \exists x \varphi$ iff there is $f : X \to M$ such that $M, X(x \mapsto f) \models \varphi$.

Dependence logic

- Syntax: FOL + $D(\bar{x}, y)$ in negation normal form
- X is a **team**, i.e., a set of assignments.
- $M, X \vDash (\neg)R\bar{x}$ iff for all $s \in X$: $M, s \vDash (\neg)R\bar{x}$.
- $M, X \vDash \varphi \wedge \psi$ iff $M, X \vDash \varphi$ and $M, X \vDash \psi$.
- $M, X \vDash \varphi \vee \psi$ iff there are Y and Z such that $M, Y \vDash \varphi$ and $M, Z \vDash \psi$ and $X = Y \cup Z$.
- $M, X \vDash \exists x \varphi$ iff there is $f : X \to M$ such that $M, X(x \mapsto f) \vDash \varphi$.

$$X(x \mapsto f) = \{\, s(x \mapsto f(s)) \mid s \in X \,\}.$$

Dependence logic

- Syntax: FOL + $D(\bar{x}, y)$ in negation normal form
- X is a **team**, i.e., a set of assignments.
- $M, X \models (\neg)R\bar{x}$ iff for all $s \in X$: $M, s \models (\neg)R\bar{x}$.
- $M, X \models \varphi \wedge \psi$ iff $M, X \models \varphi$ and $M, X \models \psi$.
- $M, X \models \varphi \vee \psi$ iff there are Y and Z such that $M, Y \models \varphi$ and $M, Z \models \psi$ and $X = Y \cup Z$.
- $M, X \models \exists x \varphi$ iff there is $f : X \to M$ such that $M, X(x \mapsto f) \models \varphi$.
- $M, X \models \forall x \varphi$ iff $M, X(x \mapsto M) \models \varphi$.

$$X(x \mapsto f) = \{ s(x \mapsto f(s)) \mid s \in X \}.$$

Dependence logic

- Syntax: FOL + $D(\bar{x}, y)$ in negation normal form
- X is a **team**, i.e., a set of assignments.
- $M, X \models (\neg)R\bar{x}$ iff for all $s \in X$: $M, s \models (\neg)R\bar{x}$.
- $M, X \models \varphi \wedge \psi$ iff $M, X \models \varphi$ and $M, X \models \psi$.
- $M, X \models \varphi \vee \psi$ iff there are Y and Z such that $M, Y \models \varphi$ and $M, Z \models \psi$ and $X = Y \cup Z$.
- $M, X \models \exists x \varphi$ iff there is $f : X \to M$ such that $M, X(x \mapsto f) \models \varphi$.
- $M, X \models \forall x \varphi$ iff $M, X(x \mapsto M) \models \varphi$.

$$X(x \mapsto f) = \{\, s(x \mapsto f(s)) \mid s \in X \,\}.$$

$$X(x \mapsto M) = \{\, s(x \mapsto a) \mid a \in M, s \in X \,\}.$$

Dependence logic

- Syntax: FOL + $D(\bar{x}, y)$ in negation normal form
- X is a **team**, i.e., a set of assignments.
- $M, X \models (\neg)R\bar{x}$ iff for all $s \in X$: $M, s \models (\neg)R\bar{x}$.
- $M, X \models \varphi \wedge \psi$ iff $M, X \models \varphi$ and $M, X \models \psi$.
- $M, X \models \varphi \vee \psi$ iff there are Y and Z such that $M, Y \models \varphi$ and $M, Z \models \psi$ and $X = Y \cup Z$.
- $M, X \models \exists x \varphi$ iff there is $f : X \to M$ such that $M, X(x \mapsto f) \models \varphi$.
- $M, X \models \forall x \varphi$ iff $M, X(x \mapsto M) \models \varphi$.

$$X(x \mapsto f) = \{ s(x \mapsto f(s)) \mid s \in X \}.$$

$$X(x \mapsto M) = \{ s(x \mapsto a) \mid a \in M, s \in X \}.$$

- $M \models \sigma$ iff $M, \{\emptyset\} \models \sigma$.

Properties of Dependence logic

Properties of Dependence logic

- $M, \emptyset \vDash \varphi$

Properties of Dependence logic

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

Properties of Dependence logic

- $M, \emptyset \vDash \varphi$
- **Downwards closure:** If $Y \subseteq X$ and $M, X \vDash \varphi$ then $M, Y \vDash \varphi$.
- Dependence logic $\equiv$ Existential Second Order logic (ESO or Σ_1^1)

Properties of Dependence logic

- $M, \emptyset \models \varphi$
- **Downwards closure:** If $Y \subseteq X$ and $M, X \models \varphi$ then $M, Y \models \varphi$.
- Dependence logic $\equiv$ Existential Second Order logic (ESO or Σ^1_1)
- For formulas the situation is slightly different: Dependence logic and the negative fragment of ESO are equivalent.

Dependence logic, again

- $M, X \models \exists x \varphi$ iff there is $f : X \to M$ such that $M, X(x \mapsto f) \models \varphi$.

- $M, X \models \forall x \varphi$ iff $M, X(x \mapsto M) \models \varphi$.

Dependence logic, again

- ~~$M, X \models \exists x \varphi$ iff there is $f : X \to M$ such that $M, X(x \mapsto f) \models \varphi$.~~
- $M, X \models \exists x \varphi$ iff there is $F : X \to \mathcal{P}(M) \setminus \{ \emptyset \}$ such that $M, X(x \mapsto F) \models \varphi$.
- $M, X \models \forall x \varphi$ iff $M, X(x \mapsto M) \models \varphi$.

The move to set-valued Fs corresponds to **non-deterministic** strategies.

Dependence logic, again

- ~~$M, X \vDash \exists x \varphi$ iff there is $f : X \to M$ such that $M, X(x \mapsto f) \vDash \varphi$.~~
- ~~$M, X \vDash \exists x \varphi$ iff there is $F : X \to \mathcal{P}(M) \setminus \{ \emptyset \}$ such that $M, X(x \mapsto F) \vDash \varphi$.~~
- ~~$M, X \vDash \forall x \varphi$ iff $M, X(x \mapsto M) \vDash \varphi$.~~
- M,X $\vDash Qx \varphi$ iff there is $F : X \to Q_M$ such that $M, X(x \mapsto F) \vDash \varphi$.

The move to set-valued *F*s corresponds to **non-deterministic** strategies.

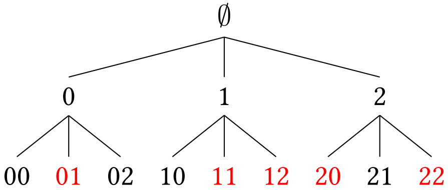
Dependence logic, again

- ~~$M, X \vDash \exists x \varphi$ iff there is $f : X \to M$ such that $M, X(x \mapsto f) \vDash \varphi$.~~
- ~~$M, X \vDash \exists x \varphi$ iff there is $F : X \to \mathcal{P}(M) \setminus \{ \emptyset \}$ such that $M, X(x \mapsto F) \vDash \varphi$.~~
- ~~$M, X \vDash \forall x \varphi$ iff $M, X(x \mapsto M) \vDash \varphi$.~~
- M,X $\vDash Qx \varphi$ iff there is $F : X \to Q_M$ such that $M, X(x \mapsto F) \vDash \varphi$.

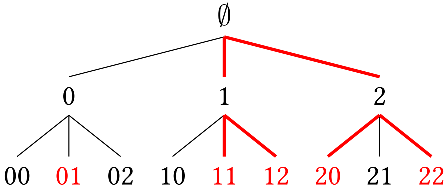
> - **Generalized quantifier**: Q a class of structures (one unary relation).
> - $M, s \vDash Qx \varphi$ iff $(M, \varphi^{M,s}) \in Q$.
> - $Q_M = \{ R \mid (M, R) \in Q \}$.

The move to set-valued Fs corresponds to **non-deterministic** strategies.

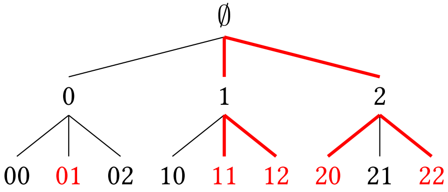
Domain $\{0, 1, 2\}$. $\exists^{\geq 2} x \exists^{\geq 2} y Rxy$



Domain $\{ 0, 1, 2 \}$. $\exists^{\geq 2} x \exists^{\geq 2} y\, Rxy$



Domain $\{ 0, 1, 2 \}$. $\exists^{\geq 2} x \exists^{\geq 2} y Rxy$



Theorem

For φ in $L(Q)$ (without dependence atoms) and Q **monotone increasing**:

$$M, X \vDash \varphi \text{ iff } M, s \vDash \varphi \text{ for all } s \in X.$$

Example I

$$\exists a \forall x \exists y \forall z \exists w \big(\mathrm{D}(z, w) \wedge (x = z \leftrightarrow y = w) \wedge y \neq a\big)$$

Example I

$$\exists a \forall x \exists y \forall z \exists w \left(\mathrm{D}(z, w) \wedge (x = z \leftrightarrow y = w) \wedge y \neq a\right)$$

$$\exists a \exists f, g \forall x, z \left((x = z \leftrightarrow f(x) = g(z)) \wedge f(x) \neq a\right)$$

Example I

$$\exists a \forall x \exists y \forall z \exists w \left(\mathrm{D}(z, w) \wedge (x = z \leftrightarrow y = w) \wedge y \neq a\right)$$

$$\exists a \exists f, g \forall x, z \left((x = z \leftrightarrow f(x) = g(z)) \wedge f(x) \neq a\right)$$

$$\exists a \exists f \forall x, z \left((f(x) = f(z) \rightarrow x = z) \wedge f(x) \neq a\right)$$

Example II

$$\begin{aligned}\exists w, w' \Big(\quad & \mathrm{D}(\bar{z}, w) \wedge \mathrm{D}(\bar{z}, w') \wedge \\ & Qx\, \exists y\, (y = w \wedge \mathrm{D}(\bar{z}, x, y) \wedge \\ & Qx'\, \exists y'\, (y' = w' \wedge \mathrm{D}(\bar{z}, x', y') \wedge \\ & \forall u\, \exists v\, (\mathrm{D}(\bar{z}, u, v) \wedge (x = u \to v = w) \wedge \\ & \forall u'\, \exists v'\, (\mathrm{D}(\bar{z}, u', v') \wedge (x' = u' \to v' = w') \wedge \\ & ((v = w \wedge v' = w') \to \varphi(u, u', \bar{z}))))) \Big)\end{aligned}$$

Example II

$$
\begin{aligned}
\exists w, w' \Big(\quad & D(\bar{z}, w) \wedge D(\bar{z}, w') \wedge \\
& Qx \, \exists y \, (y = w \wedge D(\bar{z}, x, y) \wedge \\
& Qx' \, \exists y' \, (y' = w' \wedge D(\bar{z}, x', y') \wedge \\
& \forall u \, \exists v \, (D(\bar{z}, u, v) \wedge (x = u \to v = w) \wedge \\
& \forall u' \, \exists v' \, (D(\bar{z}, u', v') \wedge (x' = u' \to v' = w') \wedge \\
& ((v = w \wedge v' = w') \to \varphi(u, u', \bar{z}))))) \Big)
\end{aligned}
$$

$$
\exists A \in Q \, \exists B \in Q \, \forall x \in A \, \forall y \in B \, \varphi(x, y, \bar{z})
$$

Example II

$$
\begin{aligned}
\exists w, w' \big(\quad & D(\bar{z}, w) \land D(\bar{z}, w') \land \\
& Qx \, \exists y \, (y = w \land D(\bar{z}, x, y) \land \\
& Qx' \, \exists y' \, (y' = w' \land D(\bar{z}, x', y') \land \\
& \forall u \, \exists v \, (D(\bar{z}, u, v) \land (x = u \to v = w) \land \\
& \forall u' \, \exists v' \, (D(\bar{z}, u', v') \land (x' = u' \to v' = w') \land \\
& ((v = w \land v' = w') \to \varphi(u, u', \bar{z}))))) \big)
\end{aligned}
$$

$$
\exists A \in Q \, \exists B \in Q \, \forall x \in A \, \forall y \in B \, \varphi(x, y, \bar{z})
$$

$$
\begin{pmatrix} Qx \\ Qy \end{pmatrix} \varphi(x, y, \bar{z}) \equiv \mathrm{Br}(Q, Q) xy \varphi(x, y, \bar{z})
$$

Example II

$$\exists w, w' \Big(\begin{array}{l} \mathrm{D}(\bar{z}, w) \wedge \mathrm{D}(\bar{z}, w') \wedge \\ Qx\, \exists y\, (y = w \wedge \mathrm{D}(\bar{z}, x, y) \wedge \\ Qx'\, \exists y'\, (y' = w' \wedge \mathrm{D}(\bar{z}, x', y') \wedge \\ \forall u\, \exists v\, (\mathrm{D}(\bar{z}, u, v) \wedge (x = u \to v = w) \wedge \\ \forall u'\, \exists v'\, (\mathrm{D}(\bar{z}, u', v') \wedge (x' = u' \to v' = w') \wedge \\ ((v = w \wedge v' = w') \to \varphi(u, u', \bar{z}))))) \Big) \end{array}$$

$$\exists A \in Q \exists B \in Q \forall x \in A \forall y \in B\, \varphi(x, y, \bar{z})$$

$$\begin{pmatrix} Qx \\ Qy \end{pmatrix} \varphi(x, y, \bar{z}) \equiv \mathrm{Br}(Q, Q)xy\varphi(x, y, \bar{z})$$

Remember that

$$\begin{pmatrix} \forall x \exists y \\ \forall z \exists w \end{pmatrix} \varphi(x, y, z, w, \bar{z}') \equiv \forall x \exists y \forall z \exists w \left(\mathrm{D}(z, w) \wedge \varphi(x, y, z, w, \bar{z}') \right)$$

Axiomatizations

Dependence: Armstrong's axioms

Let $D(\bar{x}; \bar{y})$ be a short-hand for the conjunction of $D(\bar{x}, y_1)$, ..., $D(\bar{x}, y_k)$.

Dependence: Armstrong's axioms

Let $D(\bar{x}; \bar{y})$ be a short-hand for the conjunction of $D(\bar{x}, y_1)$, ..., $D(\bar{x}, y_k)$.

If $D \cup \{ \varphi \}$ is a finite set of dependence atoms then $D \vDash \varphi$ iff φ is derivable from D with the following inference rules:

- Reflexivity: If $\bar{y} \subseteq \bar{x}$ then $D(\bar{x}; \bar{y})$.
- Augmentation: If $D(\bar{x}; \bar{y})$ then $D(\bar{x}, \bar{z}; \bar{y}, \bar{z})$.
- Transitivity: If $D(\bar{x}; \bar{y})$ and $D(\bar{y}; \bar{z})$ then $D(\bar{x}; \bar{z})$.

Dependence logic

- Dependence logic has the same strength as ESO.

Dependence logic

- Dependence logic has the same strength as ESO.
- The relation $\Gamma \vDash \varphi$ is not r.e.

Dependence logic

- Dependence logic has the same strength as ESO.
- The relation $\Gamma \models \varphi$ is not r.e.
- Restricting to φ's without dependence atoms.

Dependence logic

- Dependence logic has the same strength as ESO.
- The relation $\Gamma \vDash \varphi$ is not r.e.
- Restricting to φ's without dependence atoms.
- An easy exercise shows that $\Gamma \vDash \varphi$ **is** r.e.

Dependence logic

- Dependence logic has the same strength as ESO.
- The relation $\Gamma \vDash \varphi$ is not r.e.
- Restricting to φ's without dependence atoms.
- An easy exercise shows that $\Gamma \vDash \varphi$ **is** r.e.
- An explicit axiomatization has been given by Kontinen and Väänänen.

Dependence logic with generalized quantifiers

Dependence logic with generalized quantifiers

- For the $L(Q)$ consequences of $D(Q)$ to be axiomatizable we need the $L(Q)$ to be axiomatizable.

Dependence logic with generalized quantifiers

- For the $L(Q)$ consequences of $D(Q)$ to be axiomatizable we need the $L(Q)$ to be axiomatizable.
- Keisler showed that $L(Q_1)$ is axiomatizable, where Q_1 is ”there exists uncountable many.”

Dependence logic with generalized quantifiers

- For the $L(Q)$ consequences of $D(Q)$ to be axiomatizable we need the $L(Q)$ to be axiomatizable.
- Keisler showed that $L(Q_1)$ is axiomatizable, where Q_1 is ”there exists uncountable many.”
- An axiomatization of the $L(Q_1)$ consequences of $D(Q_1)$ was recently given.

Dependence logic with generalized quantifiers

- For the $L(Q)$ consequences of $D(Q)$ to be axiomatizable we need the $L(Q)$ to be axiomatizable.
- Keisler showed that $L(Q_1)$ is axiomatizable, where Q_1 is ”there exists uncountable many.”
- An axiomatization of the $L(Q_1)$ consequences of $D(Q_1)$ was recently given.
- Achieved by using a prenex normal form theorem.

Conclusion

Extending dependence logic with generalized quantifiers is a natural and **stable** extension:

- The satisfaction relation is naturally defined when moving to non-deterministic strategies.
- $D(Q)$ properly extends both $L(Q)$ and D.
- $D(Q)$ is in fact equivalent to $\mathrm{ESO}(Q)$.
- $D(Q)$ has a prenex normal form theorem.
- $L(Q)$ consequences of $D(Q)$ can be axiomatized in some cases.

Conclusion

Extending dependence logic with generalized quantifiers is a natural and **stable** extension:

- The satisfaction relation is naturally defined when moving to non-deterministic strategies.
- $D(Q)$ properly extends both $L(Q)$ and D.
- $D(Q)$ is in fact equivalent to $\mathrm{ESO}(Q)$.
- $D(Q)$ has a prenex normal form theorem.
- $L(Q)$ consequences of $D(Q)$ can be axiomatized in some cases.

What about the **non-monotone** case?

Thank you for your attention.

Bibliography

Fredrik Engström and Juha Kontinen. Characterizing quantifier extensions of dependence logic. **Journal of Symbolic Logic**, 78(1):307–316, 2013.

Fredrik Engström, Juha Kontinen, and Jouko Väänänen. Dependence logic with generalized quantifiers: Axiomatizations. In **Workshop on Logic, Language, Information and Computation**, 2013.

Fredrik Engström. Generalized quantifiers in dependence logic. **Journal of Logic, Language and Information**, pages 1–26, 2012. ISSN 0925-8531. 10.1007/s10849-012-9162-4.

Leon Henkin. Some remarks on infinitely long formulas. In **Infinitistic Methods (Proc. Sympos. Foundations of Math., Warsaw, 1959)**, pages 167–183. Pergamon, Oxford, 1961.

Wilfrid Hodges. Compositional semantics for a language of imperfect information. **Logic Journal of IGPL**, 5(4):539–563, 1997.

H Jerome Keisler. Logic with the quantifier "there exist uncountably many". **Annals of Mathematical Logic**, 1(1):1–93, 1970.

Juha Kontinen and Jouko Väänänen. Axiomatizing first order consequences in dependence logic. **arXiv preprint arXiv:1208.0176**, 2012.

Jouko Väänänen. **Dependence logic**, volume 70 of **London Mathematical Society Student Texts**. Cambridge University Press, Cambridge, 2007. ISBN 978-0-521-70015-3; 0-521-70015-9. A new approach to independence friendly logic.