A signature $\sigma$ is a set of symbols.

A model $\mathcal{M}$ over $\sigma$ is a set $M$ and an interpretation of the signature’s symbols.

A set of variables $V$.

If $R$ is a symbol of $\sigma$ with arity $k$ and $x_1, \ldots, x_k \in V$, then $R(x_1, \ldots, x_k)$ is a term.

A first order formula is built on terms and symbols $\vee, \wedge, \exists, \forall$.

An assignment is application $s : V \rightarrow M$. 
- A signature $\sigma$ is a set of symbols. $\sigma = \{E\}$
- A model $\mathcal{M}$ over $\sigma$ is a set $M$ and an interpretation of the signature’s symbols.
- A set of variables $V$.
- If $R$ is a symbol of $\sigma$ with arity $k$ and $x_1, \ldots, x_k \in V$, then $R(x_1, \ldots, x_k)$ is a term.
- A first order formula is built on terms and symbols $\lor, \land, \exists, \forall$.
- An assignment is application $s : V \rightarrow M$. 

Introduction to dependency logic

Definition
A signature $\sigma$ is a set of symbols. $\sigma = \{E\}$

A model $\mathcal{M}$ over $\sigma$ is a set $M$ and an interpretation of the signature's symbols.

$M = \{1, 2, 3, 4\}$, $E^\mathcal{M} = \{(1, 2), (2, 3), (3, 4), (4, 1), (1, 3)\}$

A set of variables $V$.

If $R$ is a symbol of $\sigma$ with arity $k$ and $x_1, \ldots, x_k \in V$, then $R(x_1, \ldots, x_k)$ is a term.

A first order formula is built on terms and symbols $\lor, \land, \exists, \forall$.

An assignment is application $s : V \rightarrow M$. 
A signature \( \sigma \) is a set of symbols. \( \sigma = \{ E \} \)

A model \( M \) over \( \sigma \) is a set \( M \) and an interpretation of the signature’s symbols.
\[
M = \{ 1, 2, 3, 4 \}, \quad E^M = \{ (1, 2), (2, 3), (3, 4), (4, 1), (1, 3) \}
\]

A set of variables \( V \). \( V = \{ x, y \} \)

If \( R \) is a symbol of \( \sigma \) with arity \( k \) and \( x_1, \ldots, x_k \in V \), then \( R(x_1, \ldots, x_k) \) is a term.

A first order formula is built on terms and symbols \( \lor, \land, \exists, \forall \).

An assignment is application \( s : V \rightarrow M \).
A signature $\sigma$ is a set of symbols. $\sigma = \{E\}$

A model $M$ over $\sigma$ is a set $M$ and an interpretation of the signature's symbols.

$M = \{1, 2, 3, 4\}$, $E^M = \{(1, 2), (2, 3), (3, 4), (4, 1), (1, 3)\}$

A set of variables $V$. $V = \{x, y\}$

If $R$ is a symbol of $\sigma$ with arity $k$ and $x_1, \ldots, x_k \in V$, then $R(x_1, \ldots, x_k)$ is a term. $E(x, y)$

A first order formula is built on terms and symbols $\lor, \land, \exists, \forall$.

An assignment is application $s : V \rightarrow M$. 
A signature $\sigma$ is a set of symbols. $\sigma = \{E\}$

A model $\mathcal{M}$ over $\sigma$ is a set $M$ and an interpretation of the signature’s symbols.

$M = \{1, 2, 3, 4\}$, $E^\mathcal{M} = \{(1, 2), (2, 3), (3, 4), (4, 1), (1, 3)\}$

A set of variables $V$. $V = \{x, y\}$

If $R$ is a symbol of $\sigma$ with arity $k$ and $x_1, \ldots, x_k \in V$, then $R(x_1, \ldots, x_k)$ is a term. $E(x, y)$

A first order formula is built on terms and symbols $\lor, \land, \exists, \forall$.

$E(x, y) \land \exists z (E(x, z) \land E(z, y))$

An assignment is application $s : V \rightarrow M$. 
A signature $\sigma$ is a set of symbols. $\sigma = \{ E \}$

A model $\mathcal{M}$ over $\sigma$ is a set $M$ and an interpretation of the signature’s symbols.

$M = \{ 1, 2, 3, 4 \}$, $E^\mathcal{M} = \{ (1, 2), (2, 3), (3, 4), (4, 1), (1, 3) \}$

A set of variables $V$. $V = \{ x, y \}$

If $R$ is a symbol of $\sigma$ with arity $k$ and $x_1, \ldots, x_k \in V$, then $R(x_1, \ldots, x_k)$ is a term. $E(x, y)$

A first order formula is built on terms and symbols $\lor, \land, \exists, \forall$.

$E(x, y) \land \exists z \left( E(x, z) \land E(z, y) \right)$

An assignment is application $s : V \rightarrow M$. $s : x \mapsto 1, y \mapsto 3$
A signature $\sigma$ is a set of symbols. $\sigma = \{E\}$

A model $\mathcal{M}$ over $\sigma$ is a set $M$ and an interpretation of the signature's symbols.

$M = \{1, 2, 3, 4\}$, $E^M = \{(1, 2), (2, 3), (3, 4), (4, 1), (1, 3)\}$

A set of variables $V$. $V = \{x, y\}$

If $R$ is a symbol of $\sigma$ with arity $k$ and $x_1, \ldots, x_k \in V$, then $R(x_1, \ldots, x_k)$ is a term. $E(x, y)$

A first order formula is built on terms and symbols $\lor, \land, \exists, \forall$.

$E(x, y) \land \exists z (E(x, z) \land E(z, y))$

An assignment is application $s : V \rightarrow M$. $s : x \mapsto 1, y \mapsto 3$

**Remark**

In all our talk, every model $\mathcal{M}$ will be finite (i.e., $|\mathcal{M}| \in \mathbb{N}$).
Let $x, y$ be some variables, $\phi$ and $\psi$ some first order formulas and $s$ an assignment. A formula is satisfiable by $\mathcal{M}$ and $s$ if:

- $(\mathcal{M}, s) \models R(x_1, \ldots, x_k)$ iff $(s(x_1), \ldots, s(x_k)) \in R^\mathcal{M}$
- $(\mathcal{M}, s) \models \phi \land \psi$ iff $(\mathcal{M}, s) \models \phi$ and $(\mathcal{M}, s) \models \psi$
- $(\mathcal{M}, s) \models \phi \lor \psi$ iff $(\mathcal{M}, s) \models \phi$ or $(\mathcal{M}, s) \models \psi$
- $(\mathcal{M}, s) \models \exists x \phi$ iff there exists at least one element $a \in M$ such that if we write $s[a/x]$ the prolongation of $s$ to $x$ such that $s[a/x](x) = a$, $\mathcal{M} \models x[a/x] \phi$
- $(\mathcal{M}, s) \models \forall x \phi$ iff for any $a \in M$, $\mathcal{M} \models x[a/x] \phi
$E(x, y) \land \exists z (E(x, z) \lor E(y, z))$

$E(x, y)$

$\exists z (E(x, z) \lor E(y, z))$

$E(x, z) \lor E(y, z)$

$E(x, z)$

$E(y, z)$
Introduction to dependency logic

Definition

\[ E(x, y) \land \exists z (E(x, z) \lor E(y, z)) \]

\[
\begin{array}{c|c}
 x & y \\
- & - \\
1 & 2 \\
\end{array}
\]
**Introduction to dependency logic**

**Definition**

\[ E(x, y) \land \exists z (E(x, z) \lor E(y, z)) \]

\[
\begin{array}{c|c|c}
 x & y & z \\
1 & 2 & 3 \\
\end{array}
\]

\[
\begin{array}{c|c|c|c}
 x & y & z & \ \\
1 & 2 & 3 & \\
\end{array}
\]
Introduction to dependency logic

Definition

\[ E^M = \{ (1, 2), (2, 3), (3, 4), (4, 1), (1, 3) \} \]

\[ E(x, y) \land \exists z (E(x, z) \lor E(y, z)) \]
Definition

Let $V$ be a set of variables and $M$ a model. A team $X$ over $V$ and $M$ is a set of assignments $X = \{ s : V \rightarrow M \}$

Let $x, y$ be some variables, $\phi$ and $\psi$ some first order formulas and $X$ a team. $M$ and $X$ satisfy a formula if:

- $M \models_X R(x_1, \ldots, x_k)$ iff for all $s \in X$, $(s(x_1), \ldots, s(x_k)) \in R^M$
- $M \models_X \phi \land \psi$ iff $M \models_X \phi$ and $M \models_X \psi$
- $M \models_X \phi \lor \psi$ iff $X = Y \cup Z$ and $M \models_Y \phi$ and $M \models_Z \psi$
- $M \models_X \exists x \phi$ iff there exists some function $F : X \rightarrow \mathcal{P}(M)$ such that $M \models_X [F/x] \phi$
- $M \models_X \forall x \phi$ iff $M \models_X [M/x] \phi$
Let $x, y$ be some variables, $\phi$ and $\psi$ some first order formulas and $X$ a team. $\mathcal{M}$ and $X$ satisfy a formula if:

- $\mathcal{M} \models_X R(x_1, \ldots, x_k)$ iff for all $s \in X$, $(s(x_1), \ldots, s(x_k)) \in R^\mathcal{M}$
- $\mathcal{M} \models_X \phi \land \psi$ iff $\mathcal{M} \models_X \phi$ and $\mathcal{M} \models_X \psi$
- $\mathcal{M} \models_X \phi \lor \psi$ iff $X = Y \cup Z$ and $\mathcal{M} \models_Y \phi$ and $\mathcal{M} \models_Z \psi$
- $\mathcal{M} \models_X \exists x \phi$ iff there exists some function $F : X \to \mathcal{P}(\mathcal{M})$ such that $\mathcal{M} \models_X [F/x] \phi$
- $\mathcal{M} \models_X \forall x \phi$ iff $\mathcal{M} \models_X [M/x] \phi$

**Theorem**

*If $\phi$ is a first order formula and $X$ is a team, then,*

$$\mathcal{M} \models_X \phi \text{ if and only if for all } s \in X, \mathcal{M} \models_s \phi$$
Definition

\[ E(x, y) \land \exists z (E(x, z) \lor E(y, z)) \]
Introduction to dependency logic

Definition

\[ E(x, y) \land \exists z (E(x, z) \lor E(y, z)) \]

\[ E(x, y) \]

\[ \exists z (E(x, z) \lor E(y, z)) \]

\[ E(x, z) \lor E(y, z) \]

\[ E(x, z) \]

\[ E(y, z) \]

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>
Introduction to dependency logic

Definition

\[ E(x, y) \land \exists z (E(x, z) \lor E(y, z)) \]

\[ \exists z (E(x, z) \lor E(y, z)) \]

\[ E(x, z) \lor E(y, z) \]

\[ E(x, z) \]

\[ E(y, z) \]
**Introduction to dependency logic**

**Definition**

\[ E^M = \{ (1, 2), (2, 3), (3, 4), (4, 1), (1, 3) \} \]
question

In the formula

\[ \forall x_1 \exists y_1 \forall x_2 \exists y_2 \phi(x_1, x_2, y_1, y_2) \]

\( y_2 \) depends on \( x_1, y_1 \) and \( x_2 \). How can we express that \( y_1 \) depends on \( x_1 \) and \( y_2 \) on \( x_2 \) only?
In the formula
\[ \forall x_1 \exists y_1 \forall x_2 \exists y_2 \phi(x_1, x_2, y_1, y_2) \]
y_2 depends on \( x_1 \), \( y_1 \) and \( x_2 \). How can we express that \( y_1 \) depends on \( x_1 \) and \( y_2 \) on \( x_2 \) only?

Henkin partially ordered quantifiers are formulas of the form:

\[
\left( \forall x_1 \exists y_1 \right) \left( \forall x_2 \exists y_2 \right) \phi(x_1, x_2, y_1, y_2)
\]
In the formula
\[ \forall x_1 \exists y_1 \forall x_2 \exists y_2 \phi(x_1, x_2, y_1, y_2) \]
y_2 depends on \( x_1, y_1 \) and \( x_2 \). How can we express that \( y_1 \) depends on \( x_1 \) and \( y_2 \) on \( x_2 \) only?

Hintikka and Sandu express the independence between variables by quantifiers of the form \( \exists y / \forall x \) in the so call Independence Friendly Logic:
\[ \forall x_1 \exists y_1 \forall x_2 \exists y_2 / \forall x_1 \phi(x_1, x_2, y_1, y_2) \]
In the formula
\[ \forall x_1 \exists y_1 \forall x_2 \exists y_2 \phi(x_1, x_2, y_1, y_2) \]
y_2 depends on x_1, y_1 and x_2. How can we express that y_1 depends on x_1 and y_2 on x_2 only?

Väänänen introduced the independence logic:
\[ \forall x_1 \forall x_2 \exists y_1 \exists y_2 (\phi(x_1, x_2, y_1, y_2) \land y_2 \perp x_1) \]
Let $\mathcal{M}$ be a model, $\bar{x}, \bar{y}$ some variables, $X$ a team.

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

The first order logic augmented by this atom $\exists (., .)$ is called dependence logic and written $\text{FO}(\mathcal{D})$. 

The first order logic augmented by this atom $\exists (., .)$ is called independence logic and written $\text{FO}(\perp)$.

The first order logic augmented by this atom $\exists (., .)$ is called inclusion logic and written $\text{FO}(\subseteq)$. 

The first order logic augmented by this atom $\exists (., .)$ is called dependence logic and written $\text{FO}(\mathcal{D})$. 

The first order logic augmented by this atom $\exists (., .)$ is called independence logic and written $\text{FO}(\perp)$.

The first order logic augmented by this atom $\exists (., .)$ is called inclusion logic and written $\text{FO}(\subseteq)$. 

The first order logic augmented by this atom $\exists (., .)$ is called dependence logic and written $\text{FO}(\mathcal{D})$. 

The first order logic augmented by this atom $\exists (., .)$ is called independence logic and written $\text{FO}(\perp)$.

The first order logic augmented by this atom $\exists (., .)$ is called inclusion logic and written $\text{FO}(\subseteq)$. 

The first order logic augmented by this atom $\exists (., .)$ is called dependence logic and written $\text{FO}(\mathcal{D})$. 

The first order logic augmented by this atom $\exists (., .)$ is called independence logic and written $\text{FO}(\perp)$.

The first order logic augmented by this atom $\exists (., .)$ is called inclusion logic and written $\text{FO}(\subseteq)$. 

The first order logic augmented by this atom $\exists (., .)$ is called dependence logic and written $\text{FO}(\mathcal{D})$. 

The first order logic augmented by this atom $\exists (., .)$ is called independence logic and written $\text{FO}(\perp)$.

The first order logic augmented by this atom $\exists (., .)$ is called inclusion logic and written $\text{FO}(\subseteq)$. 

The first order logic augmented by this atom $\exists (., .)$ is called dependence logic and written $\text{FO}(\mathcal{D})$. 

The first order logic augmented by this atom $\exists (., .)$ is called independence logic and written $\text{FO}(\perp)$.
Definition

Let $\mathcal{M}$ be a model, $\bar{x}, \bar{y}$ some variables, $X$ a team.

- $\mathcal{M} \models X = (\bar{x}, y)$ iff for all $s, s' \in X$, if $s(\bar{x}) = s'($\bar{x}$)$, then $s(y) = s'(y)$.
  The first order logic augmented by this atom $= (., .)$ is called dependence logic and written $\text{FO}(\mathcal{D})$.

- $\mathcal{M} \models X \perp \bar{y}$ iff $\forall s, s' \in X$, there exists $s'' \in X$ such that $s(\bar{x}) = s''(\bar{x})$ and $s'(\bar{y}) = s''(\bar{y})$.
  The first order logic augmented by this atom $\perp$ is called independence logic and written $\text{FO}(\perp)$.
Definition

Let $\mathcal{M}$ be a model, $\bar{x}, \bar{y}$ some variables, $X$ a team.

- $\mathcal{M} \models_{X}(\bar{x}, \bar{y})$ iff for all $s, s' \in X$, if $s(\bar{x}) = s'(\bar{x})$, then $s(\bar{y}) = s'(\bar{y})$.
  The first order logic augmented by this atom $= (., .)$ is called dependence logic and written $\text{FO}(\mathcal{D})$.

- $\mathcal{M} \models_{X} \bar{x} \perp \bar{y}$ iff $\forall s, s' \in X$, there exists $s'' \in X$ such that $s(\bar{x}) = s''(\bar{x})$ and $s'(\bar{y}) = s''(\bar{y})$.
  The first order logic augmented by this atom $\perp$ is called independence logic and written $\text{FO}(\perp)$.

- $\mathcal{M} \models_{X} \bar{x} \subseteq \bar{y}$ iff $X(\bar{x}) \subseteq X(\bar{y})$. Or for all $s \in X$, there exists $s' \in X$ such that $s(\bar{x}) = s(\bar{y})$.
  The first order logic augmented by this atom $\subseteq$ is called inclusion logic and written $\text{FO}(\subseteq)$.
Introduction to dependency logic

Definition

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>z</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Theorem

Let $\mathcal{M} = \{0, 1\}$ a model on an empty signature, $X$ the above team. Then

$\mathcal{M} \models x = (x, y)$
Introduction to dependency logic

Definition

Let $\mathcal{M} = \{0, 1\}$ a model on an empty signature, $X$ the above team. Then

$$\mathcal{M} \not\models (x, z)$$
Theorem

Let $\mathcal{M} = \{0, 1\}$ a model on an empty signature, $X$ the above team. Then

$$\mathcal{M} \not\vdash x \perp y$$
Theorem

Let $\mathcal{M} = \{0, 1\}$ a model on an empty signature, $X$ the above team. Then

$$\mathcal{M} \not\models_X x \perp y$$

$(1, 0)$ and $(0, 1)$ are in $X(x, y)$ but not $(1, 1)$
Introduction to dependency logic

Definition

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>z</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Theorem

Let $\mathcal{M} = \{0, 1\}$ a model on an empty signature, $X$ the above team. Then

$\mathcal{M} \models x \perp z$
Theorem (Jarmo Kontinen 2010)

For every 3-SAT instance $\theta$, there is a model $\mathcal{M}$ and a team $X$ such that

$$\mathcal{M} \models X = (x, y) \lor (z, v) \lor (z, v) \iff \theta(p_1, \ldots, p_m) \text{ is satisfiable.}$$

$\theta = \bigwedge_{i=1}^{k} C_i$ where $C_i$ are 3-clause, disjunction of three variables $p_i$ or $\neg p_i$. The sub-team associated to a clause $C_i = p_{i_1} \lor \neg p_{i_2} \lor p_{i_3}$ is:

<table>
<thead>
<tr>
<th>clause</th>
<th>var</th>
<th>parity</th>
<th>position</th>
</tr>
</thead>
<tbody>
<tr>
<td>$z$</td>
<td>$x$</td>
<td>$y$</td>
<td>$v$</td>
</tr>
<tr>
<td>$i$</td>
<td>$p_{i_1}$</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>$i$</td>
<td>$p_{i_2}$</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$i$</td>
<td>$p_{i_3}$</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>
Definition

The model checking in data complexity (i.e. for a fixed formula $\phi$) is the following problem:

- **Input:** a model $\mathcal{M}$ and a team $X$.
- **Question:** Is $\mathcal{M} \models X \phi$?

Corollary (Jarmo Kontinen 2010)

*The model checking in data complexity for formula in $\text{FO}(\mathcal{D})$ is $\text{NP}$-complete.*
Definition

Let $L$ be a logic and $C$ a complexity class. We say that the logic $L$ characterizes $C$ if for any signature $\sigma$, any class of models $\mathcal{K}$ on $\sigma$, the two following conditions are equivalent:

- $\mathcal{K}$ is definable in $L$ i.e. there exists a formula $\phi \in L$ such that for each model $\mathcal{M} \in \mathcal{M}^k$, $\mathcal{M} \in \mathcal{K}$ iff $\mathcal{M} \models \phi$.
- $\mathcal{K} \in C$ i.e., the problem: given a model $\mathcal{M} \in \mathcal{M}^r$, is $\mathcal{M} \in \mathcal{K}$? is in $C$.

Theorem (Fagin 1973)

Existential second order logic (written $\text{ESO}$ or $\Sigma^1_1$) characterize $\text{NP}$.
**Definition**

Let $L$ be a logic and $C$ a complexity class. We say that the logic $L$ characterizes $C$ if for any signature $\sigma$, any class of models $\mathcal{K}$ on $\sigma$, the two following conditions are equivalent:

- $\mathcal{K}$ is definable in $L$ i.e. there exists a formula $\phi \in L$ such that for each model $\mathcal{M} \in \mathcal{M}^k$, $\mathcal{M} \in \mathcal{K}$ iff $\mathcal{M} \models \phi$.
- $\mathcal{K} \in C$ i.e., the problem : given a model $\mathcal{M} \in \mathcal{M}^r$, is $\mathcal{M} \in \mathcal{K}$? is in $C$.

**Theorem (Väänänen 2007)**

$FO(\mathcal{D})$ characterize $NP$. 
Definition

Let $L$ be a logic and $C$ a complexity class. We say that the logic $L$ characterizes $C$ if for any signature $\sigma$, any class of models $\mathcal{K}$ on $\sigma$, the two following conditions are equivalent:

- $\mathcal{K}$ is definable in $L$ i.e. there exists a formula $\phi \in L$ such that for each model $M \in M^k$, $M \in \mathcal{K}$ iff $M \models \phi$.
- $\mathcal{K} \in C$ i.e., the problem: given a model $M \in M^r$, is $M \in \mathcal{K}$? is in $C$.

Theorem (Väänänen 2007)

$\text{FO}(\bot)$ characterize NP.
Definition

Let $L$ be a logic and $C$ a complexity class. We say that the logic $L$ characterizes $C$ if for any signature $\sigma$, any class of models $\mathcal{K}$ on $\sigma$, the two following conditions are equivalent:

- $\mathcal{K}$ is definable in $L$ i.e. there exists a formula $\phi \in L$ such that for each model $M \in M^k$, $M \in \mathcal{K}$ iff $M \models \phi$.
- $\mathcal{K} \in C$ i.e., the problem : given a model $M \in M^r$, is $M \in \mathcal{K}$? is in $C$.

Theorem (Galliani 2013)

$\text{FO}(\subseteq)$ characterize $\mathbb{P}$ on ordered structures.
The model-checking (with fixed formula) for formula of the form
\[ \phi \equiv (x \perp y) \lor (x \perp y) \lor (x \perp y) \lor x \neq y \]
is NP-complete
Let $G$ be a graph which vertices are $\{v_1, \ldots, v_n\}$, $\mathcal{M} = \{V_G\}$ and $X$ be the team

$\{ (v, v) \mid v \in V_G \} \cup \{ (v_1, v_2), (v_2, v_1) \mid (v_1, v_2) \in E_G \}$.
**Proposition**

*Then G has a 3-clique cover if and only if \( \mathcal{M} \models \exists x (x \perp y) \lor (x \perp y) \lor (x \perp y) \lor x \neq y. \)*
Proposition

If $G$ has a 3-clique cover then
\[ M \models x \ (x \perp y) \lor (x \perp y) \lor (x \perp y) \lor x \neq y. \]
Proposition

If $G$ has a 3-clique cover then
$\mathcal{M} \models x (x \perp y) \lor (x \perp y) \lor (x \perp y) \lor x \neq y.$
**Proposition**

If $G$ has a 3-clique cover then

$$\mathcal{M} \models x (x \perp y) \lor (x \perp y) \lor (x \perp y) \lor x \neq y.$$
**Proposition**

*If G has a 3-clique cover then*

\[ M \models \forall x (x \perp y) \lor (x \perp y) \lor (x \perp y) \lor x \neq y. \]
Introduction to dependency logic
Descriptive complexity

**Proposition**

If $G$ has a 3-clique cover then
\[ \mathcal{M} \models x (x \perp y) \lor (x \perp y) \lor (x \perp y) \lor x \neq y. \]
Introduction to dependency logic

Descriptive complexity

\[
\begin{array}{|c|c|}
\hline
x & y \\
\hline
1 & 1 \\
2 & 2 \\
3 & 3 \\
4 & 4 \\
5 & 5 \\
\hline
\end{array}
\bigcup
\begin{array}{|c|c|}
\hline
x & y \\
\hline
1 & 2 \\
2 & 1 \\
2 & 3 \\
3 & 2 \\
3 & 1 \\
1 & 3 \\
3 & 4 \\
4 & 3 \\
4 & 5 \\
5 & 4 \\
\hline
\end{array}
\]

Proposition

If \( \mathcal{M} \models x (x \perp y) \lor (x \perp y) \lor x \perp y \lor x \neq y \) then \( G \) has a 3-clique cover.
Proposition

If $\mathcal{M} \models x (x \perp y) \lor (x \perp y) \lor (x \perp y) \lor x \neq y$ then $G$ has a 3-clique cover.
Proposition

If $M \models (x \perp y) \lor (x \perp y) \lor (x \perp y) \lor x \neq y$ then $G$ has a 3-clique cover.
Proposition

If $\mathcal{M} \models x \land (x \perp y) \lor (x \perp y) \lor (x \perp y) \lor x \neq y$ then $G$ has a 3-clique cover.
Introduction to dependency logic

Descriptive complexity

\[
\begin{array}{c|c|c}
\hline
x & y & \cup \\
\hline
1 & 1 & \\
2 & 2 & \\
3 & 3 & \\
4 & 4 & \\
5 & 5 & \\
\hline
\end{array}
\]

\[
\begin{array}{c|c|c}
\hline
x & y & \\
\hline
1 & 2 & \\
2 & 1 & \\
2 & 3 & \\
3 & 2 & \\
3 & 1 & \\
1 & 3 & \\
3 & 4 & \\
4 & 3 & \\
4 & 5 & \\
5 & 4 & \\
\hline
\end{array}
\]

Proposition

\[\text{If } M \models x (x \perp y) \lor (x \perp y) \lor (x \perp y) \lor x \neq y \text{ then } G \text{ has a 3-clique cover.}\]