Dynamic Dependence Logic

Pietro Galliani
Universiteit van Amsterdam
ILLC
LINT (Logic for Interaction) LogiCCC project
Dynamic Dependence Logic

1) Dependence Logic
2) Dynamic Predicate Logic
3) Dynamic Dependence Logic: Hodges semantics
4) Properties
5) Game theoretic semantics for Dynamic Dependence Logic
1) Dependence Logic
2) Dynamic Predicate Logic
3) Dynamic Dependence Logic: Hodges semantics
4) Properties
5) Game theoretic semantics for Dynamic Dependence Logic
Dynamic Dependence Logic

(Henkin, 1961): Branching Quantifiers

\[
\left( \forall x \exists y \right) \left( \forall z \exists w \right) \phi(x,y,z,w)
\]
Dynamic Dependence Logic

(Henkin, 1961): Branching Quantifiers

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

where $y$ may depend on $x$, but not on $z$ or $w$;
and $w$ may depend on $z$, but not on $x$ or $y$.
Dynamic Dependence Logic

(Enkin, 1961): Branching Quantifiers

\[
\left( \forall x \exists y \left( \forall z \exists w \phi(x,y,z,w) \right) \right)
\]

y may depend on x, but not on z or w;
w may depend on z, but not on x or y;
but the value of \( \phi \) may depend on x, y, z, and w.
Dynamic Dependence Logic

$$\exists u \left( \forall x \exists y \left( \forall z \exists w \ (x = z \leftrightarrow y = w \land y \neq u) \right) \right)$$
Dynamic Dependence Logic

$$\exists u \left( \forall x \exists y \left( \forall z \exists w \left( x = z \leftrightarrow y = w \land y \neq u \right) \right) \right)$$

$$\exists u \exists f_u \exists g_u \forall x \forall z \left( x = z \leftrightarrow f_u(x) = g_u(z) \land f_u(x) \neq u \right)$$
Dynamic Dependence Logic

\[\exists u \left( \forall x \exists y \left( \forall z \exists w \left( x = z \leftrightarrow y = w \land y \neq u \right) \right) \right)\]

\[\exists u \exists f_u \exists g_u \forall x \forall z \left( x = z \leftrightarrow f_u(x) = g_u(z) \land f_u(x) \neq u \right)\]
Dynamic Dependence Logic

\[ \exists u \left( \forall x \exists y \left( \forall z \exists w \left( x = z \iff y = w \land y \neq u \right) \right) \right) \]

\[ \exists u \left( \exists f_u \exists g_u \left( \forall x \forall z \left( x = z \iff f_u(x) = g_u(z) \land f_u(x) \neq u \right) \right) \right) \]
Dynamic Dependence Logic

\[ \exists u \left( \forall x \exists y \left( \forall z \exists w \left( \begin{array}{c} x = z \\ y = w \end{array} \right) \wedge y \neq u \right) \right) \]

\[ \exists u \exists f_u \exists g_u \left( \forall x \forall z \left( x = z \iff f_u(x) = g_u(z) \wedge f_u(x) \neq u \right) \right) \]

\[ f_u, g_u : M \to M; \]
Dynamic Dependence Logic

$$\exists u \left( \forall x \exists y \left( \forall z \exists w \left( x = z \iff y = w \land y \neq u \right) \right) \right)$$

$$\exists u \exists f_u \exists g_u \forall x \forall z \left( x = z \iff f_u(x) = g_u(z) \land f_u(x) \neq u \right)$$

- $f_u, g_u : M \rightarrow M$
- $f_u = g_u$
- $f_u$ is 1-1;
\[ \exists u \left( \forall x \exists y \left( \forall z \exists w \right) \right) (x=z \iff y=w \land y \neq u) \]

\[ \exists u \exists f_u \exists g_u \forall x \forall z (x=z \iff f_u(x)=g_u(z) \land f_u(x) \neq u) \]

- \( f_u, g_u: M \rightarrow M \);
- \( f_u = g_u \);
- \( f_u \) is 1-1;
- \( u \not\in \text{Range}(f_u) \).
Dynamic Dependence Logic

\[
\exists u \left( \forall x \exists y \forall z \exists w \right) (x = z \leftrightarrow y = w \land y \neq u)
\]

\[
\exists u \exists f_u \exists g_u \forall x \forall z (x = z \leftrightarrow f_u(x) = g_u(z) \land f_u(x) \neq u)
\]

\[
\begin{align*}
&f_u, g_u : M \to M; \\
&f_u = g_u; \\
&f_u \text{ is 1-1;}
\end{align*}
\]

\[
u \not\in \text{Range}(f_u).
\]

Therefore, \( M \) is infinite.
But what does “dependence between connectives” mean?

1. Explanation in terms of *Skolem functions*: 

Dynamic Dependence Logic
But what does “dependence between connectives” mean?

1. Explanation in terms of Skolem functions:

\[
\left( \forall x \exists y \right) \phi(x,y,z,w) \equiv \exists f \exists g \ \forall x \forall z \ \phi(x,f(x),z,g(z))
\]
But what does “dependence between connectives” mean?

1. Explanation in terms of Skolem functions:

\[
\left( \forall x \exists y \right) \phi(x,y,z,w) \equiv \exists f \exists g \ \forall x \forall z \ \phi(x,f(x),z,g(z))
\]

Gives a translation from branching quantifiers to $\Sigma^1_1$.
But what does “dependence between connectives” mean?

1. Explanation in terms of Skolem functions:

\[
\left( \forall x \exists y \left( \forall z \exists w \right) \right) \phi(x,y,z,w) \equiv \exists f \exists g \ \forall x \forall z \ \phi(x,f(x),z,g(z))
\]

Gives a translation from branching quantifiers to $\Sigma^1_1$

Not a semantics!
Dynamic Dependence Logic

But what does “dependence between connectives” mean?

1. Explanation in terms of Skolem functions:

\[
\left(\forall x \exists y \quad \forall z \exists w\right) \phi(x,y,z,w) \equiv \exists f \exists g \quad \forall x \forall z \ \phi(x,f(x),z,g(z))
\]

Gives a translation from branching quantifiers to $\Sigma^1_1$

Not a semantics!

Dependencies between existential connectives are not expressed.
Dynamic Dependence Logic

But what does “dependence between connectives” mean?

2. Game Theoretic Semantics + Imperfect Information:
Dynamic Dependence Logic

But what does “dependence between connectives” mean?

2. Game Theoretic Semantics + Imperfect Information:

$$\forall x \exists y \forall z \exists w (x=w \lor y=z)$$
But what does “dependence between connectives” mean?

2. Game Theoretic Semantics + Imperfect Information:

\[ \forall x \exists y \forall z \exists w (x=w \lor y=z) \]
But what does “dependence between connectives” mean?

2. *Game Theoretic Semantics + Imperfect Information:*

\[
\forall x \exists y \forall z \exists w (x = w \lor y = z)
\]
Dynamic Dependence Logic

But what does “dependence between connectives” mean?

2. Game Theoretic Semantics + Imperfect Information:

\[ \forall x \exists y \forall z \exists w (x = w \lor y = z) \]

This is x.
But what does “dependence between connectives” mean?

2. Game Theoretic Semantics + Imperfect Information:

\[ \forall x \exists y \forall z \exists w (x = w \lor y = z) \]

This is y.
But what does “dependence between connectives” mean?

2. Game Theoretic Semantics + Imperfect Information:

\[ \forall x \exists y \forall z \exists w (x=w \lor y=z) \]
But what does “dependence between connectives” mean?

2. Game Theoretic Semantics + Imperfect Information:

$$\forall x \exists y \forall z \exists w (x=w \lor y=z)$$

This is w.
But what does “dependence between connectives” mean?

2. Game Theoretic Semantics + Imperfect Information:

$$\forall x \exists y \forall z \exists w \ (x = w \lor y = z)$$

Consider the left disjunct
Dynamic Dependence Logic

But what does “dependence between connectives” mean?

2. Game Theoretic Semantics + Imperfect Information:

$$\forall x \exists y \forall z \exists w (x = w \lor y = z)$$
Dynamic Dependence Logic

But what does “dependence between connectives” mean?

2. Game Theoretic Semantics + Imperfect Information:

\[ \forall x \exists y \forall z \exists w (x = w \lor y = z) \]

Yes!
But what does “dependence between connectives” mean?

2. Game Theoretic Semantics + Imperfect Information:

\[ \forall x \exists y \forall z \exists w (x = w \lor y = z) \]

I win!
Dynamic Dependence Logic

But what does “dependence between connectives” mean?

2. Game Theoretic Semantics + Imperfect Information:

\[ \forall x \exists y \forall z \exists w (x=w \lor y=z) \]

A FOL sentence is true in a model iff Eloise has a winning strategy for the corresponding game.
But what does “dependence between connectives” mean?

2. Game Theoretic Semantics + Imperfect Information:

\[
\left( \forall x \exists y \right) \left( \forall z \exists w \right) (x=w \lor y=z)
\]
Dynamic Dependence Logic

But what does “dependence between connectives” mean?

2. Game Theoretic Semantics + Imperfect Information:

\[ (\forall x \exists y \left( \forall z \exists w \left( x = w \lor y = z \right) \right) \]

This is \( x \).
But what does “dependence between connectives” mean?

2. *Game Theoretic Semantics + Imperfect Information*:

\[
\left( \forall x \exists y \right) \left( \forall z \exists w \right) (x = w \lor y = z)
\]

This is y.
Dynamic Dependence Logic

But what does “dependence between connectives” mean?

2. Game Theoretic Semantics + Imperfect Information:

\[
\left( \forall x \exists y \right) \left( \forall z \exists w \right) (x = w \lor y = z)
\]
Dynamic Dependence Logic

But what does “dependence between connectives” mean?

2. Game Theoretic Semantics + Imperfect Information:

\[
\left( \forall x \exists y \left( \forall z \exists w \left( x = w \lor y = z \right) \right) \right)
\]

This is z.
Dynamic Dependence Logic

But what does “dependence between connectives” mean?

2. Game Theoretic Semantics + Imperfect Information:

$$\left( \forall x \exists y \right) \left( \forall z \exists w \right) \left( x = w \lor y = z \right)$$
Dynamic Dependence Logic

But what does “dependence between connectives” mean?

2. Game Theoretic Semantics + Imperfect Information:

\[
\left( \forall x \exists y \left( \forall z \exists w \right) (x=w \lor y=z) \right)
\]

I don't know x or y...
But what does “dependence between connectives” mean?

2. Game Theoretic Semantics + Imperfect Information:

\[
(\forall x \exists y) (\forall z \exists w) (x = w \lor y = z)
\]

This is w.
But what does “dependence between connectives” mean?

2. Game Theoretic Semantics + Imperfect Information:

\[
\left( \forall x \exists y \right) \quad \left( \forall z \exists w \right) \quad (x=w \lor y=z)
\]

Consider the left disjunct.
But what does “dependence between connectives” mean?

2. Game Theoretic Semantics + Imperfect Information:

\[
\left( \forall x \exists y \left( \forall z \exists w \left( x=w \lor y=z \right) \right) \right)
\]

M

\( x \neq w, \text{ I lose...} \)
But what does “dependence between connectives” mean?

2. Game Theoretic Semantics + Imperfect Information:

\[
\left( \forall x \exists y \right) \left( \forall z \exists w \right) (x = w \lor y = z)
\]
Dynamic Dependence Logic

But what does “dependence between connectives” mean?

2. Game Theoretic Semantics + Imperfect Information:

\[
\left( \forall x \exists y \left( \forall z \exists w \left( (x=w \lor y=z) \right) \right) \right)
\]

I lose...

\[y \neq z, \text{ I lose...}\]
But what does “dependence between connectives” mean?

2. Game Theoretic Semantics + Imperfect Information:

\[
\left( \forall x \exists y \left( \forall z \exists w \right) (x=w \lor y=z) \right)
\]

This is w.
Dynamic Dependence Logic

But what does “dependence between connectives” mean?

2. Game Theoretic Semantics + Imperfect Information:

\[
\left( \forall x \exists y \right) \left( \forall z \exists w \right) (x=w \lor y=z)
\]

Consider the left disjunct.
But what does “dependence between connectives” mean?

2. Game Theoretic Semantics + Imperfect Information:

\[
\left( \forall x \exists y \left( \forall z \exists w \left( x = w \lor y = z \right) \right) \right)
\]

M

...
Dynamic Dependence Logic

But what does “dependence between connectives” mean?

2. **Game Theoretic Semantics + Imperfect Information:**

\[
(\forall x \exists y) (\forall z \exists w) (x = w \lor y = z)
\]

Neither player has a winning strategy
Dynamic Dependence Logic

But what does “dependence between connectives” mean?

2. Game Theoretic Semantics + Imperfect Information:

\[ M \models \phi \text{ iff Eloise has a winning strategy for } G_M(\phi) \]

This gives us a semantics for Branching quantifiers
But what does “dependence between connectives” mean?

2. Game Theoretic Semantics + Imperfect Information:

\[ M \models \phi \text{ iff Eloise has a winning strategy for } G_M(\phi) \]

This gives us a semantics for Branching quantifiers.

But not compositional: no way to deal with open formulas!
Dynamic Dependence Logic

Simpler notations for dependence and independence:

IF Logic (Hintikka and Sandu, 1989)

\[
\left( \forall x \exists y \left( \forall z \exists w \right) \phi(x,y,z,w) \right) \equiv \forall x \exists y \forall z \left( \exists w/x,y \right) \phi(x,y,z,w)
\]
Dynamic Dependence Logic

Simpler notations for dependence and independence:

IF Logic (Hintikka and Sandu, 1989)

$$\left( \forall x \exists y \quad \forall z \exists w \right) \phi(x,y,z,w) \equiv \forall x \exists y \forall z (\exists w/x,y) \phi(x,y,z,w)$$

w is independent from x and y
Dynamic Dependence Logic

Simpler notations for dependence and independence:

IF Logic (Hintikka and Sandu, 1989)

\[
\left( \forall x \exists y \right) \phi(x,y,z,w) \equiv \forall x \exists y \quad \forall z \quad (\exists w/\!\!x,y) \phi(x,y,z,w)
\]

Dependence Logic (Väänänen, 2007)

\[
\left( \forall x \exists y \right) \phi(x,y,z,w) \equiv \forall x \exists y \quad \forall z \exists w (= (z,w) \land \phi(x,y,z,w))
\]
Dynamic Dependence Logic

Simpler notations for dependence and independence:

**IF Logic** (Hintikka and Sandu, 1989)

\[
\left( \forall x \exists y \right) \phi(x,y,z,w) \equiv \forall x \exists y \forall z (\exists w/x,y)\phi(x,y,z,w)
\]

**Dependence Logic** (Väänänen, 2007)

\[
\left( \forall x \exists y \right) \phi(x,y,z,w) \equiv \forall x \exists y \forall z \exists w (= (z,w) \land \phi(x,y,z,w))
\]

- \( w \) depends only on \( z \)
Dynamic Dependence Logic

Hodges semantics (1997)

In order to reason about dependence, one needs to consider sets of assignments!
Hodges semantics

A team $X$ is a set of assignments:
Dynamic Dependence Logic

Hodges semantics

A team $X$ is a set of assignments:

<table>
<thead>
<tr>
<th></th>
<th>x</th>
<th>y</th>
<th>z</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_0$</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>$s_1$</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>...</td>
</tr>
<tr>
<td>$s_2$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>...</td>
</tr>
<tr>
<td>$s_3$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
**Dynamic Dependence Logic**

Hodges semantics

A *team* $X$ is a set of assignments:

<table>
<thead>
<tr>
<th></th>
<th>$x$</th>
<th>$y$</th>
<th>$z$</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_0$</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>$s_1$</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>...</td>
</tr>
<tr>
<td>$s_2$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>...</td>
</tr>
<tr>
<td>$s_3$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

An assignment (row) represents a *state of things* (e.g., partial play). A *team* represents a state of *knowledge*: “the actual state is in $X$”.
A team $X$ is a set of assignments:

<table>
<thead>
<tr>
<th>$s_0$</th>
<th>$x$</th>
<th>$y$</th>
<th>$z$</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>$s_1$</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>...</td>
</tr>
<tr>
<td>$s_2$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>...</td>
</tr>
<tr>
<td>$s_3$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>...</td>
</tr>
</tbody>
</table>

...     ...     ...     ...     ...

An assignment (row) represents a state of things (e.g., partial play). A team $X$ represents a state of knowledge: “the actual state is in $X$”.

**Closure Principle:** $X \models \emptyset$, $X' \subseteq X \Rightarrow X' \models \emptyset$
Dynamic Dependence Logic

Hodges semantics

A *team* $X$ is a set of assignments:

<table>
<thead>
<tr>
<th></th>
<th>$x$</th>
<th>$y$</th>
<th>$z$</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_0$</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>$s_1$</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>...</td>
</tr>
<tr>
<td>$s_2$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>...</td>
</tr>
<tr>
<td>$s_3$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Operations over teams
Dynamic Dependence Logic

Hodges semantics

A team $X$ is a set of assignments:

<table>
<thead>
<tr>
<th></th>
<th>x</th>
<th>y</th>
<th>z</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_0$</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>$s_1$</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>$s_2$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>$s_3$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Operations over teams

Supplementation: $F : X \rightarrow \text{Dom}(M), X[F/x] = \{s[F(s)/x] : s \in X\}$
Dynamic Dependence Logic

Hodges semantics

A team $X$ is a set of assignments:

<table>
<thead>
<tr>
<th></th>
<th>$x$</th>
<th>$y$</th>
<th>$z$</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_0$</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>$s_1$</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>...</td>
</tr>
<tr>
<td>$s_2$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>...</td>
</tr>
<tr>
<td>$s_3$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

$F(s_0) = F(s_3) = 1$
$F(s_1) = F(s_2) = 0$

Operations over teams

Supplementation: $F : X \rightarrow \text{Dom}(M)$, $X[F/x] = \{s[F(s)/x] : s \in X\}$
Dynamic Dependence Logic

Hodges semantics

A team $X$ is a set of assignments:

<table>
<thead>
<tr>
<th></th>
<th>$x$</th>
<th>$y$</th>
<th>$z$</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_0$</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>$s_1$</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>...</td>
</tr>
<tr>
<td>$s_2$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>...</td>
</tr>
<tr>
<td>$s_3$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

$F(s_0) = F(s_3) = 1$
$F(s_1) = F(s_2) = 0$

$X[F/x]$: 

Operations over teams

Supplementation: $F : X \rightarrow \text{Dom}(M)$, $X[F/x] = \{s[F(s)/x] : s \in X\}$
Dynamic Dependence Logic

Hodges semantics

A *team* $X$ is a set of assignments:

<table>
<thead>
<tr>
<th></th>
<th>$s'_0$</th>
<th>$s_1$</th>
<th>$s_2$</th>
<th>$s_3$</th>
<th>$...$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x$</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>$...$</td>
</tr>
<tr>
<td>$y$</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$...$</td>
</tr>
<tr>
<td>$z$</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>$...$</td>
</tr>
<tr>
<td>$...$</td>
<td>$...$</td>
<td>$...$</td>
<td>$...$</td>
<td>$...$</td>
<td>$...$</td>
</tr>
</tbody>
</table>

Operations over teams

**Supplementation**: $F : X \rightarrow \text{Dom}(M)$, $X[F/x] = \{s[F(s)/x] : s \in X\}$
Hodges semantics

A team $X$ is a set of assignments:

<table>
<thead>
<tr>
<th></th>
<th>x</th>
<th>y</th>
<th>z</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_0'$</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>$s_1'$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>...</td>
</tr>
<tr>
<td>$s_2$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>...</td>
</tr>
<tr>
<td>$s_3$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

$F(s_0) = F(s_3) = 1$
$F(s_1) = F(s_2) = 0$

$X[F/x]$:

Operations over teams

Supplementation: $F : X \rightarrow \text{Dom}(M)$, $X[F/x] = \{s[F(s)/x] : s \in X\}$
Dynamic Dependence Logic

Hodges semantics

A *team* $X$ is a set of assignments:

<table>
<thead>
<tr>
<th></th>
<th>$x$</th>
<th>$y$</th>
<th>$z$</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s'_0$</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>$s'_1$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>...</td>
</tr>
<tr>
<td>$s'_2$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>...</td>
</tr>
<tr>
<td>$s_3$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>...</td>
</tr>
</tbody>
</table>

$F(s_0) = F(s_3) = 1$
$F(s_1) = F(s_2) = 0$

$X[F/x]$:

Operations over teams

*Supplementation*: $F : X \rightarrow \text{Dom}(M)$, $X[F/x] = \{ s[F(s)/x] : s \in X \}$
Hodges semantics

A team \( X \) is a set of assignments:

\[
\begin{array}{|c|c|c|c|c|}
\hline
& x & y & z & \ldots \\
\hline
s'_{0} & 1 & 1 & 0 & \ldots \\
\hline
s'_{1} & 0 & 0 & 1 & \ldots \\
\hline
s'_{2} & 0 & 0 & 1 & \ldots \\
\hline
s'_{3} & 1 & 0 & 1 & \ldots \\
\hline
\ldots & \ldots & \ldots & \ldots & \ldots \\
\hline
\end{array}
\]

\( F(s_{0}) = F(s_{3}) = 1 \)
\( F(s_{1}) = F(s_{2}) = 0 \)

\( X[F/x] \):

Operations over teams

Supplementation: \( F : X \rightarrow \text{Dom}(M), X[F/x] = \{s[F(s)/x] : s \in X\} \)
Dynamic Dependence Logic

Hodges semantics

A team $X$ is a set of assignments:

<table>
<thead>
<tr>
<th></th>
<th>x</th>
<th>y</th>
<th>z</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s'_0$</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>$s'_1$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>...</td>
</tr>
<tr>
<td>$s'_3$</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

$F(s_0) = F(s_3) = 1$
$F(s_1) = F(s_2) = 0$

$X[F/x]$:

Operations over teams

*Supplementation:* $F : X \rightarrow \text{Dom}(M)$, $X[F/x] = \{s[F(s)/x] : s \in X\}$
Hodges semantics

A *team* $X$ is a set of assignments:

<table>
<thead>
<tr>
<th></th>
<th>$x$</th>
<th>$y$</th>
<th>$z$</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s'_0$</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>$s'_1$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>...</td>
</tr>
<tr>
<td>$s'_3$</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

**Operations over teams**

*Supplementation:* $F : X \rightarrow \text{Dom}(M)$, $X[F/x] = \{s[F(s)/x] : s \in X\}$

*Duplication:* $X[M/x] = \{s[m/x] : s \in X, m \in \text{Dom}(M)\}$
Dynamic Dependence Logic

Hodges semantics

<table>
<thead>
<tr>
<th></th>
<th>x</th>
<th>y</th>
<th>z</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>s'_{00}</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>s'_{01}</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>s'_{10}</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>s'_{11}</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>s'_{30}</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>s'_{30}</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Operations over teams

Supplementation: $F : X \to \text{Dom}(M)$, $X[F/x] = \{s[F(s)/x] : s \in X\}$

Duplication: $X[M/x] = \{s[m/x] : s \in X, m \in \text{Dom}(M)\}$
Dynamic Dependence Logic

Hodges semantics

<table>
<thead>
<tr>
<th></th>
<th>x</th>
<th>y</th>
<th>z</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>s'_{00}</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>s'_{01}</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>s'_{10}</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>...</td>
</tr>
<tr>
<td>s'_{11}</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Operations over teams

Supplementation: $F : X \rightarrow \text{Dom}(M)$, $X[F/x] = \{s[F(s)/x] : s \in X\}$

Duplication: $X[M/x] = \{s[m/x] : s \in X, m \in \text{Dom}(M)\}$
Dynamic Dependence Logic

Hodges semantics

Two satisfiability relations, $X \models^{+} \phi$ and $X \models^{-} \phi$;

$X \models^{+} R_{t_1...t_n} \iff \forall s \in X, s \models_{FO} R_{t_1...t_n}$;

$X \models^{-} R_{t_1...t_n} \iff \forall s \in X, s \models_{FO} \neg R_{t_1...t_n}$;
Dynamic Dependence Logic

Hodges semantics

Two satisfiability relations, $X \models^+ \phi$ and $X \models^- \phi$:

$X \models^+ \text{R}_1\ldots \text{R}_n \iff \forall s \in X, s \models_{\text{FO}} \text{R}_1\ldots \text{R}_n$;

$X \models^- \text{R}_1\ldots \text{R}_n \iff \forall s \in X, s \models_{\text{FO}} \neg \text{R}_1\ldots \text{R}_n$;

$X \models^+ t = t' \iff \forall s \in X, s \models_{\text{FO}} t = t'$;

$X \models^- t = t' \iff \forall s \in X, s \models_{\text{FO}} \neg t = t'$;
Dynamic Dependence Logic

Hodges semantics

Two satisfiability relations, \( X |=^+ \phi \) and \( X |=^- \phi \):

\[
X |=^+ R_{t_1...t_n} \iff \forall s \in X, s |=-_{FO} R_{t_1...t_n}; \\
X |=^- t = t' \iff \forall s \in X, s |=-_{FO} t = t'; \\
X |=^+ \neg \phi \iff X |=- \phi; \\
X |=^- \neg \phi \iff X |=^+ \phi;
\]
Hodges semantics

Two satisfiability relations, $X \models^+ \phi$ and $X \models^- \phi$;

$X \models^+ R_{t_1 \ldots t_n} \iff \forall s \in X, s \models_{FO} R_{t_1 \ldots t_n}$;

$X \models^- t = t' \iff \forall s \in X, s \models_{FO} t = t'$;

$X \models^+ \neg \phi \iff X \models^- \phi$;

$X \models^+ \phi \lor \psi \iff X = Y \cup Z, Y \models^+ \phi, Z \models^+ \psi$;

$X \models^- \phi \lor \psi \iff X \models^- \phi$ and $X \models^- \psi$;
Dynamic Dependence Logic

Hodges semantics

Two satisfiability relations, $X \models^+ \phi$ and $X \models^- \phi$:

- $X \models^+ R t_1...t_n \iff \forall s \in X, s \models_{FO} R t_1...t_n$;
- $X \models^+ \top t = t' \iff \forall s \in X, s \models_{FO} \top t = t'$;
- $X \models^+ \top \neg \phi \iff X \models \neg \phi$;
- $X \models^+ \top \phi \lor \psi \iff X = Y \cup Z, Y \models \phi, Z \models \psi$;
- $X \models^+ \top \exists x \phi \iff \exists F \text{ s.t. } X[F/x] \models ^+ \phi$. 

- $X \models^- R t_1...t_n \iff \forall s \in X, s \models_{FO} \neg R t_1...t_n$;
- $X \models^- \top \neg t = t' \iff \forall s \in X, s \models_{FO} \neg t = t'$;
- $X \models^- \top \neg \neg \phi \iff X \models \neg \phi$;
- $X \models^- \top \neg \phi \lor \psi \iff X \models \neg \phi \text{ and } X \models \neg \psi$;
- $X \models^- \top \exists x \phi \iff X[M/x] \models \neg \phi$. 

Dynamic Dependence Logic

Hodges semantics

Two satisfiability relations, $X \models ^+ \phi$ and $X \models ^- \phi$:

$X \models ^+ R_{t_1...t_n} \iff \forall s \in X, s \models_{FO} R_{t_1...t_n}$;

$X \models ^+ t = t' \iff \forall s \in X, s \models_{FO} t = t'$;

$X \models ^+ \neg \phi \iff X \models ^- \phi$;

$X \models ^+ \phi \lor \psi \iff X = Y \cup Z, Y \models ^+ \phi, Z \models ^+ \psi$;

$X \models ^+ \exists x \phi \iff \exists F \text{ s.t. } X[F/x] \models ^+ \phi$.

$X \models ^+ = (t_1...t_n) \iff \forall s, s' \in X, t_i(s) = t_i(s') \text{ for } i=1...n-1 \Rightarrow t_n(s) = t_n(s')$;

$X \models ^- = (t_1...t_n) \iff X = \emptyset$.

A sentence $\phi$ of Dependence Logic is true in Game Theoretic Semantics iff $X \models ^+ \phi$ for any nonempty team $X$. 


Dynamic Dependence Logic

Meaning is carried by relations (tables)
Connectives are *transitions* between relations

\[ X |=^+ \phi \iff \forall s \in X, s |=_{FO} \phi \]
\[ X |=^+ \phi \iff \exists F \text{ s.t. } X[F/x] |=^+ \phi \]
\[ X |=^+ \phi \iff \forall s, s' \in X, t_i(s) = t_i(s') \text{ for } i=1...n-1 \implies t_n(s) = t_n(s') \]

A sentence \( \phi \) of Dependence Logic is true in Game Theoretic Semantics iff \( X |=^+ \phi \) for any nonempty team \( X \).
Dynamic Dependence Logic

1) Dependence Logic
2) **Dynamic Predicate Logic**
3) Dynamic Dependence Logic: Hodges semantics
4) Properties
5) Game theoretic semantics for Dynamic Dependence Logic
6) Equivalence of Hodges semantics and Game Theoretic Semantics
Dynamic Dependence Logic

“Dynamic Semantics” is the name of a family of theoretical linguistics frameworks with subscribe to the following motto:

The meaning of a sentence does not lie in its truth conditions, but rather in the way it changes (the representation of) the information of the interpreter.

(Groenendijk and Stokhof, 1991)
Dynamic Dependence Logic

Introduced to deal with anaphora binding:
Dynamic Dependence Logic

Introduced to deal with anaphora binding:

A man_1 walks in the park.

\[ \exists x_1 (\text{MAN}(x_1) \land \text{WALK\_IN\_PARK}(x_1)) \]
Dynamic Dependence Logic

Introduced to deal with anaphora binding:

A man$_1$ walks in the park. He$_1$ whistles.

$\exists x_1 (\text{MAN}(x_1) \land \text{WALK\_IN\_PARK}(x_1)) \land \text{WHISTLES}(x_1)$
Dynamic Dependence Logic

Introduced to deal with anaphora binding:

A man \(_1\) walks in the park. He \(_1\) whistles.

\[ \exists x_1 (\text{MAN}(x_1) \land \text{WALK\_IN\_PARK}(x_1)) \land \text{WHISTLES}(x_1) \]

\[ \exists x_1 (\text{MAN}(x_1) \land \text{WALK\_IN\_PARK}(x_1) \land \text{WHISTLES}(x_1)) \]
Dynamic Dependence Logic

Introduced to deal with anaphora binding:

A man$_1$ walks in the park. He$_1$ whistles.

$$\exists x_1 (\text{MAN}(x_1) \land \text{WALK\_IN\_PARK}(x_1)) \land \text{WHISTLES}(x_1)$$

$$\exists x_1 (\text{MAN}(x_1) \land \text{WALK\_IN\_PARK}(x_1) \land \text{WHISTLES}(x_1))$$

But:
Dynamic Dependence Logic

Introduced to deal with anaphora binding:

A man_1 walks in the park. He_1 whistles.

\[ \exists x_1 (\text{MAN}(x_1) \land \text{WALK\_IN\_PARK}(x_1)) \land \text{WHISTLES}(x_1) \]

\[ \exists x_1 (\text{MAN}(x_1) \land \text{WALK\_IN\_PARK}(x_1) \land \text{WHISTLES}(x_1)) \]

But:

Not every man_1 does not walk in the park.

\[ \neg \forall x_1 (\text{MAN}(x_1) \rightarrow \neg \text{WALK\_IN\_PARK}(x_1)) \]
Dynamic Dependence Logic

Introduced to deal with anaphora binding:

A man\(_1\) walks in the park. He\(_1\) whistles.

\(\exists x_1(MAN(x_1) \land WALK\_IN\_PARK(x_1)) \land WHISTLES(x_1)\)

\(\exists x_1(MAN(x_1) \land WALK\_IN\_PARK(x_1) \land WHISTLES(x_1))\)

But:

Not every man\(_1\) does not walk in the park. *He\(_1\) whistles.

\(\neg \forall x_1(MAN(x_1) \rightarrow \neg WALK\_IN\_PARK(x_1)) \land WHISTLES(x_1)\)
Dynamic Dependence Logic

Introduced to deal with anaphora binding:

A man₁ walks in the park. He₁ whistles.

\[ \exists x_1 (\text{MAN}(x_1) \land \text{WALK\_IN\_PARK}(x_1)) \land \text{WHISTLES}(x_1) \]
\[ \exists x_1 (\text{MAN}(x_1) \land \text{WALK\_IN\_PARK}(x_1) \land \text{WHISTLES}(x_1)) \]

But:

Not every man₁ does not walk in the park. *He₁ whistles.

\[ \neg \forall x_1 (\text{MAN}(x_1) \rightarrow \neg \text{WALK\_IN\_PARK}(x_1)) \land \text{WHISTLES}(x_1) \]
\[ \neg \forall x_1 ((\text{MAN}(x_1) \rightarrow \neg \text{WALK\_IN\_PARK}(x_1)) \land \text{WHISTLES}(x_1)) \]
Dynamic Dependence Logic

Introduced to deal with anaphora binding:

A man₁ walks in the park. He₁ whistles.

$$\exists x_1 (\text{MAN}(x_1) \land \text{WALK\_IN\_PARK}(x_1)) \land \text{WHISTLES}(x_1)$$

$$\exists x_1 (\text{MAN}(x_1) \land \text{WALK\_IN\_PARK}(x_1) \land \text{WHISTLES}(x_1))$$

But:  

**UNGRAMMATICAL**

Not every man₁ does not walk in the park. *He₁ whistles.

$$\neg \forall x_1 (\text{MAN}(x_1) \rightarrow \neg \text{WALK\_IN\_PARK}(x_1)) \land \text{WHISTLES}(x_1)$$

$$\neg \forall x_1 ((\text{MAN}(x_1) \rightarrow \neg \text{WALK\_IN\_PARK}(x_1)) \land \text{WHISTLES}(x_1))$$
A sentence does not just correspond to a set of possible states of things, but also to a *transition* between information states of the listener.
Dynamic Dependence Logic

A sentence does not just correspond to a set of possible states of things, but also to a *transition* between information states of the listener.

\[(s,s') |= \phi : \text{starting from the information state } s, \ \phi \text{ can be successfully interpreted and the final information state can be } s'.\]
Dynamic Dependence Logic

A sentence does not just correspond to a set of possible states of things, but also to a transition between information states of the listener.

\[(s,s') |= \phi :\text{ starting from the information state } s, \phi \text{ can be successfully interpreted and the final information state can be } s'.\]

*Dynamic Predicate Logic* (Groenendijk and Stokhof 1991):
- information states = assignments
- The only dynamic aspect is anaphora bindings (see van Benthem 1996 for other forms of dynamics)
A sentence does not just correspond to a set of possible states of things, but also to a transition between information states of the listener.

\[(s,s') \models \phi \; : \; \text{starting from the information state } s, \phi \text{ can be successfully interpreted and the final information state can be } s'.\]

*Dynamic Predicate Logic* (Groenendijk and Stokhof 1991):
- information states = assignments
- The only dynamic aspect is anaphora bindings (*see van Benthem 1996 for other forms of dynamics*)

\[(s,s') \models R_{t_1 \ldots t_n} \iff s = s' \text{ and } s \models_{FO} R_{t_1 \ldots t_n};\]
Dynamic Dependence Logic

A sentence does not just correspond to a set of possible states of things, but also to a \textit{transition} between information states of the listener.

\[(s,s') \models \phi : \text{starting from the information state } s, \ \phi \text{ can be successfully interpreted and the final information state can be } s'.\]

\textit{Dynamic Predicate Logic} (Groenendijk and Stokhof 1991):

- information states = assignments
- The only dynamic aspect is anaphora bindings (see van Benthem 1996 for other forms of dynamics)

\[(s,s') \models R_{t_1..t_n} \iff s = s' \text{ and } s \models_{\text{FO}} R_{t_1..t_n};\]

\[(s,s') \models t = t' \iff s = s' \text{ and } s \models_{\text{FO}} t = t';\]
Dynamic Dependence Logic

A sentence does not just correspond to a set of possible states of things, but also to a transition between information states of the listener.

(s, s') |= \phi : starting from the information state s, \phi can be successfully interpreted and the final information state can be s'.

Dynamic Predicate Logic (Groenendijk and Stokhof 1991):

- information states = assignments
- The only dynamic aspect is anaphora bindings (see van Benthem 1996 for other forms of dynamics)

(s, s') |= R_{t_1..t_n} \iff s = s' and s |=_{FO} R_{t_1..t_n};

(s, s') |= t=t' \iff s = s' and s |=_{FO} t=t';

(s, s') |= \neg \phi \iff s = s' and for no s'', (s, s'') |= \phi;
Dynamic Dependence Logic

A sentence does not just correspond to a set of possible states of things, but also to a *transition* between information states of the listener.

\[(s,s') \models \phi : \text{starting from the information state } s, \phi \text{ can be successfully interpreted and the final information state can be } s'.\]

*Dynamic Predicate Logic* (Groenendijk and Stokhof 1991):

- information states = assignments
- The only dynamic aspect is anaphora bindings (see van Benthem 1996 for other forms of dynamics)

\[(s,s') \models R_{t_1..t_n} \iff s = s' \text{ and } s \models_{\text{FO}} R_{t_1..t_n};\]

\[(s,s') \models t=t' \iff s=s' \text{ and } s \models_{\text{FO}} t=t';\]

\[(s,s') \models \neg \phi \iff s=s' \text{ and for no } s'', (s,s'') \models \phi;\]

\[(s,s') \models \phi \land \psi \iff \exists s'', (s,s'') \models \phi \text{ and } (s'', s') \models \psi;\]
Dynamic Dependence Logic

A sentence does not just correspond to a set of possible states of things, but also to a *transition* between information states of the listener.

\[(s,s') \models \phi : \text{starting from the information state } s, \phi \text{ can be successfully interpreted and the final information state can be } s'.\]

*Dynamic Predicate Logic* (Groenendijk and Stokhof 1991):

- Information states = assignments
- The only dynamic aspect is anaphora bindings (see van Benthem 1996 for other forms of dynamics)

\[(s,s') \models R_{t_1..t_n} \iff s = s' \text{ and } s \models_{FO} R_{t_1..t_n};\]

\[(s,s') \models t = t' \iff s = s' \text{ and } s \models_{FO} t = t';\]

\[(s,s') \models \neg \phi \iff s = s' \text{ and for no } s'', (s,s'') \models \phi;\]

\[(s,s') \models \phi \land \psi \iff \exists s'', (s,s'') \models \phi \text{ and } (s'', s') \models \psi;\]

\[(s,s') \models \exists x \phi \iff \exists m \in \text{dom}(M) \text{ s.t. } (s[m/x], s') \models \phi.\]
Dynamic Dependence Logic

A sentence does not just correspond to a set of possible states of things, but also to a *transition* between information states of the listener.

\[(s,s') \models \phi : \text{starting from the information state } s, \phi \text{ can be successfully interpreted and the final information state can be } s'.\]

*Dynamic Predicate Logic* (Groenendijk and Stokhof 1991):

information states = assignments

The only dynamic aspect is anaphora bindings (see van Benthem 1996 for other forms of dynamics)

\[(s,s') \models R_{t_1..t_n} \iff s = s' \text{ and } s \models_{FO} R_{t_1..t_n};\]

\[(s,s') \models t=t' \iff s=s' \text{ and } s \models_{FO} t=t';\]

\[(s,s') \models \neg \phi \iff s = s' \text{ and for no } s'', (s,s'') \models \phi;\]

\[(s,s') \models \phi \land \psi \iff \exists s'', (s,s'') \models \phi \text{ and } (s'', s') \models \psi;\]

\[(s,s') \models \exists x \phi \iff \exists m \in \text{dom}(M) \text{ s.t. } (s[m/x], s') \models \phi.\]
Dynamic Dependence Logic

A sentence does not just correspond to a set of possible states of things, but also to a *transition* between information states of the listener.

\[(s,s') \models \phi : \text{starting from the information state } s, \phi \text{ can be successfully interpreted and the final information state can be } s'.\]

*Dynamic Predicate Logic* (Groenendijk and Stokhof 1991):

- information states = assignments
- The only dynamic aspect is anaphora bindings (see van Benthem 1996 for other forms of dynamics)

\[(s,s') \models R_{t_1\ldots t_n} \iff s = s' \text{ and } s \models_{FO} R_{t_1\ldots t_n};\]

\[(s,s') \models t=t' \iff s=s' \text{ and } s \models_{FO} t=t';\]

\[(s,s') \models \neg \phi \iff s=s' \text{ and for no } s'', (s,s'') \models \phi;\]

\[(s,s') \models \phi \land \psi \iff \exists s'', (s,s'') \models \phi \text{ and } (s'', s') \models \psi;\]

\[(s,s') \models \exists x \phi \iff \exists m \in \text{dom}(M) \text{ s.t. } (s[m/x], s') \models \phi.\]
A sentence does not just correspond to a set of possible states of things, but also to a transition between information states of the listener.

\[(s,s') \models \phi : \text{starting from the information state } s, \phi \text{ can be successfully interpreted and the final information state can be } s'.\]

*Dynamic Predicate Logic* (Groenendijk and Stokhof 1991):

- information states = assignments
- The only dynamic aspect is anaphora bindings (see van Benthem 1996 for other forms of dynamics)

\[(s,s') \models R_{t_1..t_n} \iff s = s' \text{ and } s \models_{FO} R_{t_1..t_n};\]

\[(s,s') \models t = t' \iff s = s' \text{ and } s \models_{FO} t = t';\]

\[(s,s') \models \neg \phi \iff s = s' \text{ and for no } s'', (s,s'') \models \phi;\]

\[(s,s') \models \phi \land \psi \iff \exists s'', (s,s'') \models \phi \text{ and } (s'', s') \models \psi;\]

\[(s,s') \models \exists x \phi \iff \exists m \in \text{dom}(M) \text{ s.t. } (s[m/x], s') \models \phi.\]

\[\exists x \phi := \exists x \land \phi\]
Dynamic Dependence Logic

1) Dependence Logic
2) Dynamic Predicate Logic
3) **Dynamic Dependence Logic: Hodges semantics**
4) Properties
5) Game theoretic semantics for Dynamic Dependence Logic
6) Equivalence of Hodges semantics and Game Theoretic Semantics
Dynamic Dependence Logic

Can we make a dynamic version of dependence logic?
Dynamic Dependence Logic

Can we make a dynamic version of dependence logic?

Formulas = transitions between teams
Can we make a dynamic version of dependence logic?

Formulas = transitions between teams

\[ \phi ::= Rt_1...t_n | t=t' | =(t_1...t_n) | \exists x | \neg(\phi) | \phi \lor \psi | \phi \land \psi \]

As the excluded middle does not hold in DL, we must keep track of positive and negative transitions:

\[(X, Y) |=^+ \phi \text{ means } "If the current state is in } X, \text{ after executing } \phi \text{ the current state will be in } Y".\]
Dynamic Dependence Logic

Can we make a dynamic version of dependence logic?

Formulas = transitions between teams

\[ \phi ::= Rt_1...t_n \mid t=t' \mid =(t_1...t_n) \mid \exists x \mid \neg(\phi) \mid \phi \lor \psi \mid \phi \cdot \psi \]

As the excluded middle does not hold in DL, we must keep track of positive and negative transitions:

\((X, Y) \models^+ \phi\) means "If the current state is in X, after executing \(\phi\) the current state will be in Y".

\((X, Y) \models^- \phi\) means "If the current state is in X, after executing \(\neg\phi\) the current state will be in Y".
Can we make a dynamic version of dependence logic?

Formulas = transitions between teams

\[ \phi ::= \text{Rt}_1\ldots\text{t}_n \mid \text{t}=\text{t}' \mid =\left(\text{t}_1\ldots\text{t}_n\right) \mid \exists\text{x} \mid \neg(\phi) \mid \phi \lor \psi \mid \phi \cdot \psi \]

As the excluded middle does not hold in DL, we must keep track of *positive* and *negative* transitions:

\[(X, Y) \models^+ \text{Rt}_1\ldots\text{t}_n \iff X \subseteq Y \text{ and } \forall s \in X, s \models_{\text{FO}} \text{Rt}_1\ldots\text{t}_n;\]

\[(X, Y) \models^- \text{Rt}_1\ldots\text{t}_n \iff X \subseteq Y \text{ and } \forall s \in X, s \models_{\text{FO}} \neg\text{Rt}_1\ldots\text{t}_n;\]
Dynamic Dependence Logic

Can we make a dynamic version of dependence logic?

Formulas = transitions between teams

\( \phi ::= R_{t_1...t_n} | t=t' | (t_1...t_n) | \exists x | \neg(\phi) | \phi \lor \psi | \phi \cdot \psi \)

As the excluded middle does not hold in DL, we must keep track of positive and negative transitions:

\((X, Y) \models^+ R_{t_1...t_n} \iff X \subseteq Y \text{ and } \forall s \in X, s \models_{\text{FO}} R_{t_1...t_n} ;\)

\((X, Y) \models^- R_{t_1...t_n} \iff X \subseteq Y \text{ and } \forall s \in X, s \models_{\text{FO}} \neg R_{t_1...t_n} ;\)

\((X, Y) \models^+ t=t' \iff X \subseteq Y \text{ and } \forall s \in X, s \models_{\text{FO}} t=t' ;\)

\((X, Y) \models^- t=t' \iff X \subseteq Y \text{ and } \forall s \in X, s \models_{\text{FO}} t\neq t' ;\)
Dynamic Dependence Logic

Can we make a dynamic version of dependence logic?

Formulas = transitions between teams

\[ \phi ::= R t_1 \ldots t_n \mid t=t' \mid =(t_1 \ldots t_n) \mid \exists x \mid \neg(\phi) \mid \phi \vee \psi \mid \phi \cdot \psi \]

As the excluded middle does not hold in DL, we must keep track of *positive* and *negative* transitions:

\[(X, Y) \models^+ =(t_1 \ldots t_n) \iff X \subseteq Y \text{ and } \forall s, s' \in X, \]
\[ t_1(s) = t_1(s') \ldots t_{n-1}(s) = t_{n-1}(s') \Rightarrow t_n(s) = t_n(s') \; ; \]

\[(X, Y) \models^- =(t_1 \ldots t_n) \iff X = \emptyset ; \]
Dynamic Dependence Logic

Can we make a dynamic version of dependence logic?

Formulas = transitions between teams

\[ \phi ::= R t_1 \ldots t_n \mid t = t' \mid=(t_1 \ldots t_n) \mid \exists x \mid \neg(\phi) \mid \phi \lor \psi \mid \phi \cdot \psi \]

As the excluded middle does not hold in DL, we must keep track of *positive* and *negative* transitions:

\[(X, Y) |=^+ (t_1 \ldots t_n) \iff X \subseteq Y \text{ and } \forall s, s' \in X, \quad t_1(s) = t_1(s') \ldots t_{n-1}(s) = t_{n-1}(s') \Rightarrow t_n(s) = t_n(s') ; \]
\[(X, Y) |=^- (t_1 \ldots t_n) \iff X = \emptyset ; \]
\[(X, Y) |=^+ \exists x \iff \exists F \text{ s.t. } X[F/x] \subseteq Y ; \]
\[(X, Y) |=^- \exists x \iff X[M/x] \subseteq Y ; \]
Dynamic Dependence Logic

Can we make a dynamic version of dependence logic?

Formulas = transitions between teams

\( \phi ::= R_{t_1 \ldots t_n} \mid t=t' \mid =_{(t_1 \ldots t_n)} \mid \exists x \mid \neg(\phi) \mid \phi \lor \psi \mid \phi \cdot \psi \)

As the excluded middle does not hold in DL, we must keep track of \textit{positive} and \textit{negative} transitions:

\((X, Y) |\equiv^+ \neg(\phi) \iff (X, Y) |\equiv^- \phi;\)
\((X, Y) |\equiv^- \neg(\phi) \iff (X, Y) |\equiv^+ \phi;\)
Can we make a dynamic version of dependence logic?

Formulas = transitions between teams

\[ \phi ::= R t_1 \ldots t_n \mid t=t' \mid =(t_1 \ldots t_n) \mid \exists x \mid \neg(\phi) \mid \phi \lor \psi \mid \phi . \psi \]

As the excluded middle does not hold in DL, we must keep track of *positive* and *negative* transitions:

\[(X, Y) \models^+ \neg(\phi) \iff (X, Y) \models^− \phi; \]
\[(X, Y) \models^− \neg(\phi) \iff (X, Y) \models^+ \phi; \]
\[(X, Y) \models^+ \phi \lor \psi \iff X = X_1 \cup X_2, (X_1, Y) \models^+ \phi \text{ and } (X_2, Y) \models^+ \psi; \]
\[(X, Y) \models^− \phi \lor \psi \iff (X, Y) \models^− \phi \text{ and } (X, Y) \models^− \psi. \]
Dynamic Dependence Logic

Can we make a dynamic version of dependence logic?

Formulas = transitions between teams

\[ \phi ::= Rt_1...t_n | t=t' | =(t_1...t_n) | \exists x | \neg(\phi) | \phi \lor \psi | \phi \cdot \psi \]

As the excluded middle does not hold in DL, we must keep track of \textit{positive} and \textit{negative} transitions:

\[(X, Y) \models^+ \phi \cdot \psi \iff \exists Z \text{ s.t. } (X, Z) \models^+ \phi \text{ and } (Z, Y) \models^+ \psi;\]
\[(X, Y) \models^- \phi \cdot \psi \iff \exists Z \text{ s.t. } (X, Z) \models^- \phi \text{ and } (Z, Y) \models^- \psi.\]
Can we make a dynamic version of dependence logic?

Formulas = transitions between teams

\[ \phi ::= Rt_1...t_n \mid t=t' \mid =(t_1...t_n) \mid \exists x \mid \neg(\phi) \mid \phi \lor \psi \mid \phi \cdot \psi \]

As the excluded middle does not hold in DL, we must keep track of positive and negative transitions:

\[(X, Y) |=^+ \phi \cdot \psi \iff \exists Z \text{ s.t. } (X, Z) |=^+ \phi \text{ and } (Z, Y) |=^+ \psi; \]
\[(X, Y) |=^- \phi \cdot \psi \iff \exists Z \text{ s.t. } (X, Z) |=^- \phi \text{ and } (Z, Y) |=^- \psi. \]

\[X |=^+ \phi \text{ iff } \exists Y \text{ s.t. } (X, Y) |=^+ \phi\]

If \( \psi \) sentence, \( |=^+ \psi \) iff \( X |=^+ \psi \) for any \( X \).
Dynamic Dependence Logic

1) Dependence Logic
2) Dynamic Predicate Logic
3) Dynamic Dependence Logic: Hodges semantics

4) **Properties**

5) Game theoretic semantics for Dynamic Dependence Logic
Dynamic Dependence Logic is a conservative extension of Dependence Logic:
Dynamic Dependence Logic is a conservative extension of Dependence Logic:

\[ X |=^+_{DL} \phi \iff \exists Y, (X,Y) |=^+_{DDL} \phi \]
Dynamic Dependence Logic is a conservative extension of Dependence Logic:

\[ X \models^+_{DL} \phi \iff \exists Y, (X,Y) \models^+_{DDL} \phi \]

Dynamic Dependence Logic is paraconsistent:
Dynamic Dependence Logic is a conservative extension of Dependence Logic:

\[ X \models^+_{DL} \phi \iff \exists Y, (X,Y) \models^+_{DDL} \phi \]

Dynamic Dependence Logic is paraconsistent:

For all teams \( X \), \( X \models^+_{DDL} \exists x \) and \( X \models^-_{DDL} \exists x \)
Dynamic Dependence Logic is a conservative extension of Dependence Logic:

\[ X \models^{+}_{DL} \phi \iff \exists Y, (X, Y) \models^{+}_{DDL} \phi \]

Dynamic Dependence Logic is paraconsistent:

For all teams X, \( X \models^{+}_{DDL} \exists x \) and \( X \models^{-}_{DDL} \exists x \)

Dynamic Dependence Logic is as expressive as Dependence Logic:
Dynamic Dependence Logic

*Dynamic Dependence Logic is a conservative extension of Dependence Logic:*

\[ X \models^{+}_{DL} \phi \iff \exists Y, (X,Y) \models^{+}_{DDL} \phi \]

*Dynamic Dependence Logic is paraconsistent:*

For all teams \( X \), \( X \models^{+}_{DDL} \exists x \) and \( X \models^{-}_{DDL} \exists x \)

*Dynamic Dependence Logic is as expressive as Dependence Logic:*

\[ \forall \phi \in DDL \ \exists \phi' \in DL \ s.t., \ for \ all \ X, \ X \models^{+}_{DDL} \phi \iff X \models^{+}_{DL} \phi' \]
Dynamic Dependence Logic

Closure Property:
Dynamic Dependence Logic

Closure Property:

If \((X,Y) |= \phi\), \(X' \subseteq X\), \(Y \subseteq Y'\) then \((X', Y') |= \phi\)
Closure Property:

If \((X,Y) \models \phi, X' \subseteq X, Y \subseteq Y'\) then \((X', Y') \models \phi\)
Dynamic Dependence Logic

Closure Property:

If \((X,Y) |= \phi\), \(X' \subseteq X\), \(Y \subseteq Y'\) then \((X', Y') |= \phi\)
**Dynamic Dependence Logic**

**Closure Property:**

If \((X,Y) |= \phi\), \(X' \subseteq X\), \(Y \subseteq Y'\) then \((X', Y') |= \phi\)

<table>
<thead>
<tr>
<th></th>
<th>x</th>
<th>y</th>
<th>z</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>s₀</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>s₁</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>...</td>
</tr>
<tr>
<td>s₂</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>x</th>
<th>y</th>
<th>z</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>s₀</td>
<td>F(s₀)</td>
<td>1</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>s₁</td>
<td>F(s₁)</td>
<td>1</td>
<td>1</td>
<td>...</td>
</tr>
<tr>
<td>s₂</td>
<td>F(s₂)</td>
<td>0</td>
<td>0</td>
<td>...</td>
</tr>
</tbody>
</table>
Dynamic Dependence Logic

Closure Property:

If \((X, Y) \models \phi\), \(X' \subseteq X\), \(Y' \subseteq Y'\) then \((X', Y') \models \phi\)

| \(s_0\) | \(0\) | \(1\) | \(0\) | ... |
| \(s_1\) | \(1\) | \(1\) | \(1\) | ... |
| \(s_2\) | \(1\) | \(0\) | \(0\) | ... |

\(F(s_0) = 0\)
\(F(s_1) = 1\)
\(F(s_2) = 1\)
Dynamic Dependence Logic

Closure Property:

If \((X,Y) \models \phi\), \(X' \subseteq X\), \(Y \subseteq Y'\) then \((X', Y') \models \phi\)

\[
\begin{array}{c|c|c|c|c|c|c}
\hline
x & y & z & \ldots \\
\hline
s_0 & 0 & 1 & 0 & \ldots \\
\hline
s_1 & 1 & 1 & 1 & \ldots \\
\hline
s_2 & 1 & 0 & 0 & \ldots \\
\hline
\end{array}
\]

\[
\begin{array}{c|c|c|c|c|c|c}
\hline
x & y & z & \ldots \\
\hline
s_0 & 0 & 1 & 0 & \ldots \\
\hline
s_1 & 1 & 1 & 1 & \ldots \\
\hline
s_2 & 1 & 0 & 0 & \ldots \\
\hline
s_3 & 0 & 0 & 0 & \ldots \\
\hline
\end{array}
\]

\(\exists x \quad F(s_0) = 0\)
\(F(s_1) = 1\)
\(F(s_2) = 1\)

\(F(s_3) = 1\)
Dynamic Dependence Logic

Closure Property:

If \((X, Y) \models \phi\), \(X' \subseteq X\), \(Y \subseteq Y'\) then \((X', Y') \models \phi\)

<table>
<thead>
<tr>
<th></th>
<th>x</th>
<th>y</th>
<th>z</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>s_0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>s_2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>...</td>
</tr>
</tbody>
</table>

\(F(s_0) = 0\)
\(F(s_1) = 1\)
\(F(s_2) = 1\)

\(\exists x\)

<table>
<thead>
<tr>
<th></th>
<th>x</th>
<th>y</th>
<th>z</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>s_0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>s_1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>...</td>
</tr>
<tr>
<td>s_2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>s_3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Closure Property:

If \((X,Y) \models \phi\), \(X' \subseteq X\), \(Y \subseteq Y'\) then \((X', Y') \models \phi\)

Every row of the antecedent is sent into some row of the consequent
Dynamic Dependence Logic

1) Dependence Logic
2) Dynamic Predicate Logic
3) Dynamic Dependence Logic: Hodges semantics
4) Properties
5) Game theoretic semantics for Dynamic Dependence Logic
Dynamic Dependence Logic

\[ \phi ::= R t_1 \ldots t_n \mid t=t' \mid (t_1 \ldots t_n) \mid \exists x \mid \neg \mid \phi \lor \psi \mid \phi \cdot \psi \]
Dynamic Dependence Logic

\[ \phi ::= Rt_1...t_n \mid t=t' \mid =(t_1...t_n) \mid \exists x \mid \neg \mid \phi \lor \psi \mid \phi \cdot \psi \]

\[ \neg(\phi) ::= \neg \cdot \phi \cdot \neg \]
Dynamic Dependence Logic

\[ \phi ::= R t_1 \ldots t_n \mid t = t' \mid (t_1 \ldots t_n) \mid \exists x \mid \neg \mid \phi \lor \psi \mid \phi \cdot \psi \]

\[ \neg(\phi) ::= \neg . \phi . \neg \]

For every \( \phi \), \( X \subseteq M \), define the game \( G^M_X(\phi) \):
Dynamic Dependence Logic

\[ \phi ::= \text{Rt}_1...t_n \mid t=t' \mid =(t_1...t_n) \mid \exists x \mid \neg \mid \phi \lor \psi \mid \phi \cdot \psi \]

\[ \neg(\phi) ::= \neg \cdot \phi \cdot \neg \]

For every \( \phi \), \( X \) \( M \), define the game \( G^M_X(\phi) \):

Two players, \( \bigcirc \) and \( \blacklozenge \);
Dynamic Dependence Logic

\[ \phi ::= Rt_{1 \ldots n} \mid t=t' \mid =t_{1 \ldots n} \mid \exists x \mid \neg \mid \phi \lor \psi \mid \phi . \psi \]

\[ \neg(\phi) ::= \neg . \phi . \neg \]

For every \( \phi \), \( X \) \( M \), define the game \( G^M_X(\phi) \):

Two players, ⬤ and ⬦;

Positions of the form \((\phi_0, \phi_1, \ldots \phi_n, s, \alpha)\),
\( \phi_0 \ldots \phi_n \in \text{DDL} \);
\( s \) assignment;
\( \alpha \in \{I, II\} \);
Dynamic Dependence Logic

\[ \phi ::= R_{t_1...t_n} | t=t' | (t_1...t_n) | \exists x | \neg | \phi \lor \psi | \phi \cdot \psi \]

\[ \neg(\phi) ::= \neg \cdot \phi \cdot \neg \]

For every \( \phi \), \( X \in \mathbb{M} \), define the game \( G^M_X(\phi) \):

Two players, \( \heartsuit \) and \( \diamondsuit \);

Positions of the form \((\phi_0, \phi_1, \ldots, \phi_n, s, \alpha)\),
\[ \phi_0 \ldots \phi_n \in \text{DDL}; \]
\[ s \text{ assignment}; \]
\[ \alpha \in \{I, II\}; \quad (\alpha = I: \heartsuit \text{ moves}, \alpha = II: \diamondsuit \text{ moves}) \]
Dynamic Dependence Logic

\[ \phi ::= Rt_1 \ldots t_n \mid \text{t=t'} \mid \text{=(t_1\ldots t_n)} \mid \exists x \mid \neg \mid \phi \lor \psi \mid \phi \cdot \psi \]

\[ \neg(\phi) ::= \neg . \phi . \neg \]

For every \( \phi, X \ M \), define the game \( G^M_X(\phi) \):

Two players, \( \blacklozenge \) and \( \blacksquare \);  

Positions of the form \( (\phi_0, \phi_1, \ldots \phi_n, s, \alpha) \),  
\( \phi_0 \ldots \phi_n \in \text{DDL} \);  
\( s \) assignment;  
\( \alpha \in \{I, II\}; \ (\alpha = I: \blacklozenge \text{ moves, } \alpha = II: \blacksquare \text{ moves}) \)

Starting positions \( (\phi, s, II), s \in X \).
Dynamic Dependence Logic

Position $p$  Successors $S(p)$
Dynamic Dependence Logic

**Position** $p$

$$(R_{t_1...t_n}, \phi_1 ... \phi_n, s, \alpha)$$

**Successors** $S(p)$

$$\{(\phi_1 ... \phi_n, s, \alpha)\} \text{ if } s \models_{\text{FO}} R_{t_1...t_n}, \\emptyset \text{ otherwise;}$$
Dynamic Dependence Logic

**Position** $p$

$(R_{t_1...t_n}, \phi_1 ... \phi_n, s, \alpha)$

$(t=t', \phi_1 ... \phi_n, s, \alpha)$

**Successors** $S(p)$

$\{(\phi_1 ... \phi_n, s, \alpha)\}$ if $s \models_{FO} R_{t_1...t_n}$, \emptyset otherwise;

$\{(\phi_1 ... \phi_n, s, \alpha)\}$ if $s \models_{FO} t=t'$, \emptyset otherwise;
Dynamic Dependence Logic

Position $p$

$(Rt_1...t_n, \phi_1 ... \phi_n, s, \alpha)$

$(t=t', \phi_1 ... \phi_n, s, \alpha)$

$(=(t_1...t_n), \phi_1 ... \phi_n, s, \alpha)$

Successors $S(p)$

$\{\langle \phi_1 ... \phi_n, s, \alpha \rangle \}$ if $s \models_{FO} Rt_1...t_n$

$\emptyset$ otherwise;

$\{\langle \phi_1 ... \phi_n, s, \alpha \rangle \}$ if $s \models_{FO} t=t'$

$\emptyset$ otherwise;

$\{\langle \phi_1 ... \phi_n, s, \alpha \rangle\}$;
Dynamic Dependence Logic

**Position $p$**

$(Rt_1...t_n, \phi_1 ... \phi_n, s, \alpha)$

$(t=t', \phi_1 ... \phi_n, s, \alpha)$

$(= (t_1...t_n), \phi_1 ... \phi_n, s, \alpha)$

$(\exists x, \phi_1 ... \phi_n, s, \alpha)$

**Successors $S(p)$**

$
\{(\phi_1 ... \phi_n, s, \alpha)\}$ if $s \models_{\text{FO}} Rt_1...t_n \land$
\[\emptyset \text{ otherwise;}
\]

$
\{(\phi_1 ... \phi_n, s, \alpha)\}$ if $s \models_{\text{FO}} t=t'$,
\[\emptyset \text{ otherwise;}
\]

$
\{(\phi_1 ... \phi_n, s, \alpha)\};
\]

$
\{(\phi_1 ... \phi_n, s[m/x], \alpha), m \in \text{Dom}(M)\};
\]
Dynamic Dependence Logic

**Position** $p$

\[(R_{t_1...t_n}, \phi_1 ... \phi_n, s, \alpha)\]

\[(t=t', \phi_1 ... \phi_n, s, \alpha)\]

\[=(t_1...t_n), \phi_1 ... \phi_n, s, \alpha)\]

\[\exists x, \phi_1 ... \phi_n, s, \alpha)\]

\[\neg, \phi_1 ... \phi_n, s, \alpha)\]

**Successors** $S(p)$

\[
\{ (\phi_1 ... \phi_n, s, \alpha) \} \text{ if } s \models_{FO} R_{t_1...t_n} ,
\]
\[
\emptyset \text{ otherwise};
\]

\[
\{ (\phi_1 ... \phi_n, s, \alpha) \} \text{ if } s \models_{FO} t=t',
\]
\[
\emptyset \text{ otherwise};
\]

\[
\{ (\phi_1 ... \phi_n, s, \alpha) \} ;
\]

\[
\{ (\phi_1 ... \phi_n, s[m/x], \alpha), m \in \text{Dom}(M) \} ;
\]

\[
\{ (\phi_1 ... \phi_n, s, 1-\alpha) \} ;
\]
<table>
<thead>
<tr>
<th>Position $p$</th>
<th>Successors $S(p)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(Rt_1...t_n, \phi_1 ... \phi_n, s, \alpha)$</td>
<td>${(\phi_1 ... \phi_n, s, \alpha)} \text{ if } s \models_{\text{FO}} Rt_1...t_n$</td>
</tr>
<tr>
<td>$(t=t', \phi_1 ... \phi_n, s, \alpha)$</td>
<td>${(\phi_1 ... \phi_n, s, \alpha)} \text{ if } s \models_{\text{FO}} t=t'$,</td>
</tr>
<tr>
<td>$(\exists x, \phi_1 ... \phi_n, s, \alpha)$</td>
<td>$\emptyset \text{ otherwise}$</td>
</tr>
<tr>
<td>$(\neg, \phi_1 ... \phi_n, s, \alpha)$</td>
<td>${(\phi_1 ... \phi_n, s[m/x], \alpha), m \in \text{Dom}(M)}$</td>
</tr>
<tr>
<td>$(\psi \lor \theta, \phi_1 ... \phi_n, s, \alpha)$</td>
<td>${(\phi_1 ... \phi_n, s, 1-\alpha)}$</td>
</tr>
<tr>
<td></td>
<td>${(\psi, \phi_1 ... \phi_n, s, \alpha), (\theta, \phi_1 ... \phi_n, s, \alpha)}$</td>
</tr>
</tbody>
</table>
Dynamic Dependence Logic

**Position $p$**

(Rt$_1$...t$_n$, $\phi_1$ ... $\phi_n$, s, $\alpha$)

(t=t', $\phi_1$ ... $\phi_n$, s, $\alpha$)

(=t$_1$...t$_n$, $\phi_1$ ... $\phi_n$, s, $\alpha$)

($\exists x$, $\phi_1$ ... $\phi_n$, s, $\alpha$)

(¬ $\phi_1$ ... $\phi_n$, s, $\alpha$)

($\psi \lor \theta$, $\phi_1$ ... $\phi_n$, s, $\alpha$)

($\psi \cdot \theta$, $\phi_1$ ... $\phi_n$, s, $\alpha$)

**Successors $S(p)$**

\{($\phi_1$ ... $\phi_n$, s, $\alpha$)\} if $s \models_{\text{FO}} \text{Rt}_1$...t$_n$,

$\emptyset$ otherwise;

\{($\phi_1$ ... $\phi_n$, s, $\alpha$)\} if $s \models_{\text{FO}} \text{t}=\text{t}'$,

$\emptyset$ otherwise;

\{($\phi_1$ ... $\phi_n$, s, $\alpha$)\};

\{($\phi_1$ ... $\phi_n$, s[$m/x$], $\alpha$), $m \in \text{Dom}(M)$\};

\{($\phi_1$ ... $\phi_n$, s, 1−$\alpha$)\};

\{($\psi$, $\phi_1$ ... $\phi_n$, s, $\alpha$), ($\theta$, $\phi_1$ ... $\phi_n$, s, $\alpha$)\};

\{($\psi$, $\theta$, $\phi_1$ ... $\phi_n$, s, $\alpha$)\}. 
A strategy $\sigma$ for Player $\alpha$ in $\{I, \ II\}$ is a function s.t.

$$\sigma(p) \in S(p) \text{ for all } p = (\phi_0, \phi_1, \ldots, \phi_n, s, \alpha).$$
Dynamic Dependence Logic

A strategy $\sigma$ for Player $\alpha$ in $\{I, II\}$ is a function s.t.

$$\sigma(p) \in S(p) \text{ for all } p = (\phi_0, \phi_1, \ldots, \phi_n, s, \alpha).$$

Given two strategies $\sigma$, $\tau$, for I, II in $G^M_X(\phi)$,

$$\text{Plays}(\sigma, \tau, G^M_X(\phi)) = \{ \text{LFP}(\overline{ld} + (\sigma \cup \tau) \circ \text{lst}, (\phi, s, \text{II})) : s \in X\}.$$
Dynamic Dependence Logic

A strategy $\sigma$ for Player $\alpha$ in $\{I, II\}$ is a function s.t.

$$\sigma(p) \in S(p) \text{ for all } p = (\phi_0, \phi_1, \ldots, \phi_n, s, \alpha).$$

Given two strategies $\sigma$, $\tau$, for I, II in $G^M_X(\phi)$,

$$\text{Plays}(\sigma, \tau, G^M_X(\phi)) = \{ \text{LFP}((\overline{id} + (\sigma \cup \tau) \circ \text{lst}, (\phi, s, II)) : s \in X) \}.$$  

A strategy $\tau$ for II in $G^M_X(\phi)$ is uniform iff whenever
A strategy \( \sigma \) for Player \( \alpha \) in \{I, II\} is a function s.t.

\[
\sigma(p) \in S(p) \text{ for all } p = (\phi_0, \phi_1, \ldots, \phi_n, s, \alpha).
\]

Given two strategies \( \sigma, \tau \), for I, II in \( G_X^M(\phi) \),

\[
\text{Plays}(\sigma, \tau, G_X^M(\phi)) = \{ \text{LFP}(\text{id} + (\sigma \cup \tau) \circ \text{lst}, (\phi, s, \text{II})) : s \in X \}.
\]

A strategy \( \tau \) for II in \( G_X^M(\phi) \) is \textit{uniform} iff whenever

- \( \overline{p} \in \text{Plays}(\sigma, \tau, G_X^M(\phi)) \), \( \overline{p}' \in \text{Plays}(\sigma', \tau, G_X^M(\phi)) \), and
A strategy $\sigma$ for Player $\alpha$ in $\{I, II\}$ is a function s.t.

$$\sigma(p) \in S(p) \text{ for all } p = (\phi_0, \phi_1, \ldots, \phi_n, s, \alpha).$$

Given two strategies $\sigma$, $\tau$, for I, II in $G^M_X(\phi)$,

$$\text{Plays}(\sigma, \tau, G^M_X(\phi)) = \{ \text{LFP}(\text{id} + (\sigma \cup \tau) \circ \text{lst}, (\phi, s, \text{II})) : s \in X \}.$$

A strategy $\tau$ for II in $G^M_X(\phi)$ is uniform iff whenever

- $p \in \text{Plays}(\sigma, \tau, G^M_X(\phi))$, $p' \in \text{Plays}(\sigma', \tau, G^M_X(\phi))$, and
- $(=(t_1\ldots t_n), \phi_1 \ldots \phi_n, s, \text{II}), (=(t_1\ldots t_n), \phi_1 \ldots \phi_n, s', \text{II})$ occurs in $p$, $p'$ for the same instance of $=(t_1\ldots t_n)$, and $t_i(s) = t_i(s')$ for $i=1..n-1$
A strategy $\sigma$ for Player $\alpha$ in \{I, II\} is a function s.t.

$$\sigma(p) \in S(p) \text{ for all } p = (\phi_0, \phi_1, \ldots \phi_n, s, \alpha).$$

Given two strategies $\sigma, \tau$, for I, II in $G^M_X(\phi)$,

$$\text{Plays}(\sigma, \tau, G^M_X(\phi)) = \{ \text{LFP}(\text{Id} + (\sigma \cup \tau) \circ \text{lst}, (\phi, s, II)) : s \in X\}.$$  

A strategy $\tau$ for II in $G^M_X(\phi)$ is uniform iff whenever

- $\bar{p} \in \text{Plays}(\sigma, \tau, G^M_X(\phi))$, $\bar{p}' \in \text{Plays}(\sigma', \tau, G^M_X(\phi))$, and
- $=(t_1 \ldots t_n), \phi_1 \ldots \phi_n, s, II), (=(t_1 \ldots t_n), \phi_1 \ldots \phi_n, s', II)$ occurs in $\bar{p}$, $\bar{p}'$ for the same instance of $=(t_1 \ldots t_n)$, and $t_i(s) = t_i(s')$ for $i=1..n-1$

then $t_n(s) = t_n(s')$
Dynamic Dependence Logic

We say that $\phi: X \rightarrow Y$ if there exists a uniform strategy $\tau$ for $\Pi$ in $G^M_X(\phi)$ s.t., for all strategies $\sigma$ of $I$, 
We say that $\phi: X \rightarrow Y$ if there exists a uniform strategy $\tau$ for II in $G^M_X(\phi)$ s.t., for all strategies $\sigma$ of I,

$$\overline{p} = p_1...p_n \in \text{Plays}(\sigma, \tau, G^M_X(\phi)) \Rightarrow p_n = (\varepsilon, s, \text{II}) \text{ for some } s \in Y$$
Dynamic Dependence Logic

We say that $\phi: X \rightarrow Y$ if there exists a uniform strategy $\tau$ for II in $G_M^{X}(\phi)$ s.t., for all strategies $\sigma$ of I,

$$p = p_1...p_n \in \text{Plays}(\sigma, \tau, G_M^{X}(\phi)) \Rightarrow p_n = (\varepsilon, s, II) \text{ for some } s \in Y$$

**Theorem**

$$(X,Y) \models^+ \phi \text{ iff } \phi: X \rightarrow Y$$

$$(X,Y) \models^- \phi \text{ iff } \neg(\phi): X \rightarrow Y$$
Dynamic Dependence Logic

M = (\{0,1\})
Dynamic Dependence Logic

\[ M = (\{0,1\}) \quad (\exists x \lor \exists y) \cdot (x = y) \]
Dynamic Dependence Logic

\[ M = \{0, 1\} \]

\[ (\exists x \lor \exists y) \land (x = y) \]

\[ X = \]

<table>
<thead>
<tr>
<th></th>
<th>x</th>
<th>y</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>s_0</td>
<td>0</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>s_1</td>
<td>0</td>
<td>1</td>
<td>...</td>
</tr>
</tbody>
</table>
Dynamic Dependence Logic

\[ M = \{0,1\} \]

\[ (\exists x \lor \exists y) \cdot (x=y), s, II) (s \in X) \]

<table>
<thead>
<tr>
<th></th>
<th>x</th>
<th>y</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>s_0</td>
<td>0</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>s_1</td>
<td>0</td>
<td>1</td>
<td>...</td>
</tr>
</tbody>
</table>
M = \{0, 1\}

\[
\begin{array}{c|cc|}
 & x & y \\
\hline
s_0 & 0 & 0 \\
 s_1 & 0 & 1 \\
\end{array}
\]

\[(\exists x \lor \exists y) \cdot (x = y), s, \text{II} \] (s \in X)

\[(\exists x \lor \exists y), (x = y), s, \text{II} \]
Dynamic Dependence Logic

\[ M = \{0,1\} \]

\[ X = \]

\[
\begin{array}{ccc}
  & x & y & \ldots \\
 s_0 & 0 & 0 & \ldots \\
 s_1 & 0 & 1 & \ldots \\
\end{array}
\]

\[
((\exists x \lor \exists y) \cdot (x=y), s, \text{II}) \ (s \in X)
\]

\[
((\exists x \lor \exists y), (x=y), s, \text{II})
\]

II chooses L

II chooses R

\[
(\exists x, (x=y), s, \text{II})
\]

\[
(\exists y, (x=y), s, \text{II})
\]
Dynamic Dependence Logic

\[ M = (\{0,1\}) \]

\[ X = \]

<table>
<thead>
<tr>
<th></th>
<th>x</th>
<th>y</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>s₀</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>s₁</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

\[(\exists x \lor \exists y). (x=y), s, \text{II}) (s \in X)\]

(\[(\exists x \lor \exists y), (x=y), s, \text{II}\])

\[ \text{II chooses } L \]

(\[\exists x, (x=y), s, \text{II}\])

\[ \text{II picks } m \]

\[ s' = s[m/x] \]

(\[(x=y), s', \text{II}\])

\[ \text{II chooses } R \]

(\[\exists y, (x=y), s, \text{II}\])
Dynamic Dependence Logic

\[ M = \{0,1\} \]

\[ X = \]

<table>
<thead>
<tr>
<th></th>
<th>x</th>
<th>y</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>s_0</td>
<td>0</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>s_1</td>
<td>0</td>
<td>1</td>
<td>...</td>
</tr>
</tbody>
</table>

\[ ((\exists x \land \exists y). (x=y), s, \text{II}) \quad (s \in X) \]

\[ ((\exists x \land \exists y), (x=y), s, \text{II}) \]

II chooses L

(\exists x, (x=y), s, \text{II})

II picks \( m \)

\( s' = s[m/x] \)

(\( x=y \), \( s' \), II)

II chooses R

(\( \exists y, (x=y), s, \text{II} \))

II picks \( n \)

\( s' = s[n/x] \)
Dynamic Dependence Logic

\[ M = (\{0, 1\}) \]

<table>
<thead>
<tr>
<th></th>
<th>x</th>
<th>y</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>(s_0)</td>
<td>0</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>(s_1)</td>
<td>0</td>
<td>1</td>
<td>...</td>
</tr>
</tbody>
</table>

\[ X = \]

\((\exists x \lor \exists y). (x=y), s, \text{II}) (s \in X)\]

\((\exists x \lor \exists y), (x=y), s, \text{II})\]

II chooses L

\((\exists x, (x=y), s, \text{II})\)

II picks m

\(s' = s[m/x]\)

\((x=y), s', \text{II})\)

If \(s(x) = s(y)\)

\((\varepsilon, s', \text{II})\)

II picks n

\(s' = s[n/x]\)

\((\exists y, (x=y), s, \text{II})\)

II chooses R
Dynamic Dependence Logic

\[ M = (\{0,1\}) \]

\[ X = \]

<table>
<thead>
<tr>
<th></th>
<th>x</th>
<th>y</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>(s_0)</td>
<td>0</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>(s_1)</td>
<td>0</td>
<td>1</td>
<td>...</td>
</tr>
</tbody>
</table>

\[ ((\exists x \lor \exists y) \cdot (x=y), s, II) (s \in X) \]

- \(II\) chooses \(L\)
- \(II\) chooses \(R\)

\[ ((\exists x \lor \exists y), (x=y), s, II) \]

- \(II\) picks \(m\)
  - \(s' = s[m/x]\)

- \(II\) picks \(n\)
  - \(s' = s[n/x]\)

\[ ((x=y), s', II) \]

- If \(s(x) = s(y)\)
  - \((\varepsilon, s', II)\)
\[ M = \{0, 1\} \]

\[
\begin{array}{c|ccc}
  & x & y & \ldots \\
\hline
s_0 & 0 & 0 & \ldots \\
\hline
s_1 & 0 & 1 & \ldots \\
\end{array}
\]

\[ X = \]

\[ ((\exists x \lor \exists y) \cdot (x=y), s, \Pi) (s \in X) \]

\[ ((\exists x \lor \exists y), (x=y), s, \Pi) \]

\[ (\exists x, (x=y), s, \Pi) \]

\[ (\forall y, (x=y), s, \Pi) \]

\[ \Pi \text{ chooses } L \]

\[ \Pi \text{ chooses } R \]

\[ \Pi \text{ picks } m \]

\[ \Pi \text{ picks } n \]

\[ s' = s[m/x] \]

\[ s' = s[n/x] \]

\[ (x=y), s', \Pi \]

\[ \tau((\exists x \lor \exists y) \cdot (x=y), s, \Pi) = ((\exists x \lor \exists y), (x=y), s, \Pi); \]

\[ If \ s(x) = s(y) \]

\[ (\epsilon, s', \Pi) \]
Dynamic Dependence Logic

\[ M = \{0,1\} \]

\[ X = \begin{array}{c|c|c|c|c|c}
 x & y & \ldots \\
\hline
 s_0 & 0 & 0 & \ldots \\
 s_1 & 0 & 1 & \ldots \\
\end{array} \]

\[ \tau((\exists x \lor \exists y) \times (x=y), s, \Pi) = ((\exists x \lor \exists y), (x=y), s, \Pi); \]

\[ \tau((\exists x \lor \exists y), (x=y), s, \Pi) = (\exists x, (x=y), s, \Pi); \]
\[ M = (\{0,1\}) \]

\[ X = \]

\[
\begin{array}{ccc}
\hline
& x & y & \ldots \\
\hline
s_0 & 0 & 0 & \ldots \\
\hline
s_1 & 0 & 1 & \ldots \\
\hline
\end{array}
\]

\[
\tau((\exists x \lor \exists y) \land (x=y), s, \Pi) = ((\exists x \lor \exists y), (x=y), s, \Pi); \\
\tau((\exists x \lor \exists y), (x=y), s, \Pi) = (\exists x, (x=y), s, \Pi); \\
\tau(\exists x, (x=y), s, \Pi) = (x=y, s[s(y)/x], \Pi); \\
\tau((x=y), s', \Pi) = (x=y, s'[n/x], \Pi); \\
\text{If } s(x) = s(y) \\
(\varepsilon, s', \Pi)
\]
$M = \{0, 1\}$

$X' =$

<table>
<thead>
<tr>
<th></th>
<th>x</th>
<th>y</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_0$</td>
<td>0</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>$s'_1$</td>
<td>1</td>
<td>1</td>
<td>...</td>
</tr>
</tbody>
</table>

$\tau((\exists x \lor \exists y) \cdot (x=y), s, II) = ((\exists x \lor \exists y), (x=y), s, II)$;

$\tau((\exists x \lor \exists y), (x=y), s, II) = (\exists x, (x=y), s, II)$;

$\tau(\exists x, (x=y), s, II) = (x=y, s[s(y)/x], II)$;

If $s(x) = s(y)$

$(\varepsilon, s', II)$
Dynamic Dependence Logic

\[ M = \{0,1\} \]

\[ X' = \]

\begin{array}{cccc}
\hline
 & x & y & \ldots \\
\hline
s_0 & 0 & 0 & \ldots \\
 s'_1 & 1 & 1 & \ldots \\
\hline
\end{array}

\[ ((\exists x \lor \exists y) . \ (x=y), \ s, \ II) \ (s \in X) \]

\[ ((\exists x \lor \exists y), \ (x=y), \ s, \ II) \]

\[ (\exists x, \ (x=y), \ s, \ II) \]

\[ s' = s[s(y)/x] \]

\[ ((x=y), \ s', \ II) \]

\[ ((x=y), \ s', \ II) = (x=y, \ s[s(y)/x], \ II); \]
\[ \tau(\exists x \lor \exists y) \times (x=y), \ s, \ II) = ((\exists x \lor \exists y), \ (x=y), \ s, \ II); \]
\[ \tau((\exists x \lor \exists y), \ (x=y), \ s, \ II) = (\exists x, \ (x=y), \ s, \ II); \]
\[ \tau(\exists x, \ (x=y), \ s, \ II) = (x=y, \ s[s(y)/x], \ II); \]
\[ \tau(x=y, \ s[s(y)/x], \ II) = (\varepsilon, \ s[s(y)/x], \ II). \]
Dynamic Dependence Logic

\[ M = \{0, 1\} \]

\[ X = \]

<table>
<thead>
<tr>
<th></th>
<th>x</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>s_0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>s_1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

\[ X' = \]

<table>
<thead>
<tr>
<th></th>
<th>x</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>s'_0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>s'_1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

\[ \mathcal{L}((\exists x \lor \exists y) \cdot (x=y), s, \Pi) (s \in X) \]

\[ (\exists x \lor \exists y), (x=y), s, \Pi \]

\[ (x=y), s, \Pi \]

\[ (\exists x, (x=y), s, \Pi) \]

\[ (\exists y, (x=y), s, \Pi) \]

\[ \Pi \text{ chooses } L \]

\[ \Pi \text{ chooses } R \]

\[ s' = s[y/x] \]

\[ (x=y), s', \Pi \]

\[ (\varepsilon, s', \Pi) \]

\[ s(x) = s(y) \]

\[ (\exists x \lor \exists y) \cdot (x=y): X \to X' \]
Dynamic Dependence Logic

$M = \{0,1\}$

$X =$

<table>
<thead>
<tr>
<th></th>
<th>x</th>
<th>y</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_0$</td>
<td>0</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>$s_1$</td>
<td>0</td>
<td>1</td>
<td>...</td>
</tr>
</tbody>
</table>
Dynamic Dependence Logic

\[ M = \{0,1\} \]

\[ X = \]

\[
\begin{array}{ccc}
\hline
& x & y & \cdots \\
\hline
s_0 & 0 & 0 & \cdots \\
\hline
s_1 & 0 & 1 & \cdots \\
\hline
\end{array}
\]

\[ (\exists x \lor \exists y) \cdot (x=y) \cdot \neg(x) \]
Dynamic Dependence Logic

\[ M = \{0, 1\} \]

\[ X = \]

<table>
<thead>
<tr>
<th></th>
<th>x</th>
<th>y</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>s_0</td>
<td>0</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>s_1</td>
<td>0</td>
<td>1</td>
<td>...</td>
</tr>
</tbody>
</table>

\[ ((\exists x \lor \exists y) \cdot (x=y) \cdot =(x), s, \text{II}) (s \in X) \]

\[ ((\exists x \lor \exists y) \cdot (x=y), =x, s, \text{II}) (s \in X) \]

\[ ((\exists x \lor \exists y), (x=y), =x, s, \text{II}) \]

\[ \text{II chooses L} \]

\[ (\exists x, (x=y), =x, s, \text{II}) \]

\[ \text{II chooses R} \]

\[ (\exists y, (x=y), =x, s, \text{II}) \]

\[ \text{II picks } m \]

\[ s' = s[m/x] \]

\[ ((x=y), s', \text{II}) \]

\[ \text{If } s(x) = s(y) \]

\[ (=x, s', \text{II}) \]

\[ (\varepsilon, s', \text{II}) \]
Dynamic Dependence Logic

\[M = \{0,1\}\]

\[X = \begin{array}{c|c|c|c}
       & x & y & \ldots \\
\hline
s_0 & 0 & 0 & \ldots \\
\hline
s_1 & 0 & 1 & \ldots \\
\end{array}\]

\[((\exists x \lor \exists y) \cdot (x=y) \cdot =)(x), s, \text{II} (s \in X)\]
\[\downarrow\]
\[((\exists x \lor \exists y) \cdot (x=y), =)(x), s, \text{II} (s \in X)\]
\[\downarrow\]
\[((\exists x \lor \exists y), (x=y), =)(x), s, \text{II}\]

\[\text{II chooses L}\]
\[\downarrow\]
\[\exists x, (x=y), =)(x), s, \text{II}\]

\[\text{II chooses R}\]
\[\downarrow\]
\[\exists y, (x=y), =)(x), s, \text{II}\]

\[\text{II picks n}\]
\[s' = s[n/x]\]

\[s(x) = s(y)\]
\[\downarrow\]
\[=)(x), s', \text{II}\]
\[\downarrow\]
\[(\varepsilon, s', \text{II})\]
Dynamic Dependence Logic

\[ M = \{0,1\} \]

\[ X = \]

\[
\begin{array}{ccc}
\text{x} & \text{y} & \ldots \\
\text{s}_0 & 0 & 0 & \ldots \\
\text{s}_1 & 0 & 1 & \ldots \\
\end{array}
\]

\[ X' = \]

\[
\begin{array}{ccc}
\text{x} & \text{y} & \ldots \\
\text{s}'_0 & 0 & 0 & \ldots \\
\text{s}'_1 & 1 & 1 & \ldots \\
\end{array}
\]

\[ (\exists x \vee \exists y) \cdot (x=y) \cdot =(x), s, \text{II} \quad (s \in X) \]

\[ (\exists x \vee \exists y) \cdot (x=y), =x, s, \text{II} \quad (s \in X) \]

\[ (\exists x \vee \exists y), (x=y), =x, s, \text{II} \]

II chooses L

II chooses R

II picks n

\[ s' = s[n/x] \]

\[ =x, s', \text{II} \]

\[ s(x) = s(y) \]

\[ (\text{II}, s', \text{II}) \]

\[ (\varepsilon, s', \text{II}) \]

\[ ((\exists x \vee \exists y) \cdot (x=y) \cdot =x) : X \to X' \]
Dynamic Dependence Logic

\[ M = \{0, 1\} \]

\[ X = \]

\[
\begin{array}{ccc}
\hline
x & y & \ldots \\
\hline
s_0 & 0 & 0 & \ldots \\
\hline
s_1 & 0 & 1 & \ldots \\
\hline
\end{array}
\]

\[ X' = \]

\[
\begin{array}{ccc}
\hline
x & y & \ldots \\
\hline
s'_0 & 0 & 0 & \ldots \\
\hline
s'_1 & 1 & 1 & \ldots \\
\hline
\end{array}
\]

\[(\exists x \lor \exists y). (x=y). = (x), s, \text{II} (s \in X)\]

\[(\exists x \lor \exists y). (x=y), = (x), s, \text{II} (s \in X)\]

\[(\exists x \lor \exists y), (x=y), = (x), s, \text{II}\]

II chooses L

\[(\exists x, (x=y), = (x), s, \text{II})\]

II chooses R

\[(\exists y, (x=y), = (x), s, \text{II})\]

II picks n

\[s' = s[n/x]\]

\[(x=y), s', \text{II}\]

\[s(x) = s(y)\]

\[(= (x), s', \text{II})\]

\[(\varepsilon, s', \text{II})\]

\[(\exists x \lor \exists y). (x=y). = (x): X \rightarrow X'\]

No: not uniform!
Dynamic Dependence Logic

\[ M = (\{0,1\}) \]

\[ X = \]

\[
\begin{array}{ccc}
X & y & \ldots \\
\hline
s_0 & 0 & 0 & \ldots \\
\hline
s_1 & 0 & 1 & \ldots \\
\end{array}
\]

\[ X' = \]

\[
\begin{array}{ccc}
X' & y & \ldots \\
\hline
s'_0 & 0 & 0 & \ldots \\
\hline
s'_1 & 1 & 1 & \ldots \\
\end{array}
\]

\[(\exists x \lor \exists y). (x=y). =(x): X \to X'?\]

No: not uniform!
Dynamic Dependence Logic

\[ M = (\{0,1\}) \]

\[
\begin{array}{ccc}
   & x & y & \ldots \\
\hline
s_0 & 0 & 0 & \ldots \\
s_1 & 0 & 1 & \ldots \\
\end{array}
\]

\[ X' = \]

\[
\begin{array}{ccc}
   & x & y & \ldots \\
\hline
s'_0 & 0 & 0 & \ldots \\
s'_1 & 1 & 1 & \ldots \\
\end{array}
\]

\[(\exists x \lor \exists y) \cdot (x=y) \cdot =x : X \rightarrow X'\]

No: not uniform!
(\exists x \lor \exists y) \cdot (x=y) \cdot = (x), s, II \cdot s \in X

II chooses L

(\exists x, (x=y), = (x), s, II)

s' = s[y/x]

II picks n

s' = s[n/x]

II chooses R

((\exists x, (x=y), = (x), s, II)

(\exists y, (x=y), = (x), s, II)

No: not uniform!
Dynamic Dependence Logic

$M = \{0, 1\}$

$X = \begin{array}{ccc} s_0 & 0 & 0 & \ldots \\ s_1 & 0 & 1 & \ldots \end{array}$

$X' = \begin{array}{ccc} s'_0 & 0 & 0 & \ldots \\ s'_1 & 1 & 1 & \ldots \end{array}$

$(\exists x \lor \exists y). (x=y) . =(x): X \rightarrow X'\ ?$

No: not uniform!
Dynamic Dependence Logic

\( M = \{0,1\} \)

\[ \begin{array}{ccc}
   & x & y & \ldots \\
\hline
s_0 & 0 & 0 & \ldots \\
s_1 & 0 & 1 & \ldots \\
\end{array} \]

\[ \begin{array}{ccc}
   & x & y & \ldots \\
\hline
s_0' & 0 & 0 & \ldots \\
s_1' & 1 & 1 & \ldots \\
\end{array} \]

\((\exists x \lor \exists y) \cdot (x=y) \cdot =x: X \rightarrow X'\)?

No: not uniform!
Dynamic Dependence Logic

\[ M = \{0,1\} \]

<table>
<thead>
<tr>
<th></th>
<th>x</th>
<th>y</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>(s_0)</td>
<td>0</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>(s_1)</td>
<td>0</td>
<td>1</td>
<td>...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>x</th>
<th>y</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>(s'_0)</td>
<td>0</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>(s'_1)</td>
<td>1</td>
<td>1</td>
<td>...</td>
</tr>
</tbody>
</table>

\[(\exists x \lor \exists y) \cdot (x=y) \cdot = (x): X \rightarrow X'\]

No: not uniform!
Dynamic Dependence Logic

\[ M = \{0, 1\} \]

\[ X = \]

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

\[ X' = \]

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

\[(\exists x \lor \exists y). (x=y). (x)\]: \(X \rightarrow X'\)

No: not uniform!
Dynamic Dependence Logic

\[ M = \{0, 1\} \]

\[ X = \]
\[
\begin{array}{ccc}
 & x & y & \ldots \\
 s_0 & 0 & 0 & \ldots \\
s_1 & 0 & 1 & \ldots \\
\end{array}
\]

\[ X' = \]
\[
\begin{array}{ccc}
 & x & y & \ldots \\
 s'_0 & 0 & 0 & \ldots \\
s'_1 & 1 & 1 & \ldots \\
\end{array}
\]

\((\exists x \lor \exists y) \cdot (x=y) \cdot (=x), s_0, II) (s \in X)\]

\((\exists x \lor \exists y) \cdot (x=y), (=x), s_0, II) (s \in X)\]

\((\exists x \lor \exists y), (x=y), (=x), s_0, II)\]

II chooses \( L \)

II chooses \( R \)

\((\exists x, (x=y), (=x), s_0, II)\]

\((\exists y, (x=y), (=x), s, II)\]

II picks \( n \)

\[ s' = s_0 \]

\[ s' = s[n/x] \]

\((x=y), s_0, II)\]

\[ s_0(x) = s_0(y) \]

\((=x), s_0, II)\]

\((\varepsilon, s', II)\]

\((\exists x \lor \exists y) \cdot (x=y) \cdot (=x): X \rightarrow X'?\]

No: not uniform!
**Dynamic Dependence Logic**

\[ M = \{0, 1\} \]

\[ \begin{array}{ccc}
  & x & y & \ldots \\
 s_0 & 0 & 0 & \ldots \\
 s_1 & 0 & 1 & \ldots 
\end{array} \]

\[ \begin{array}{ccc}
  & x & y & \ldots \\
 s_0' & 0 & 0 & \ldots \\
 s_1' & 1 & 1 & \ldots 
\end{array} \]

\[(\exists x \vee \exists y) \cdot (x=y) \cdot = (x) : X \rightarrow X' ? \]

No: not uniform!  \[ (= x, s_0, \mathrm{II}) \]
Dynamic Dependence Logic

\[
M = (\{0,1\})
\]

\[
X = \begin{array}{ccc}
  & x & y & \ldots \\
 s_0 & 0 & 0 & \ldots \\
s_1 & 0 & 1 & \ldots \\
\end{array}
\]

\[
X' = \begin{array}{ccc}
  & x & y & \ldots \\
 s'_0 & 0 & 0 & \ldots \\
s'_1 & 1 & 1 & \ldots \\
\end{array}
\]

\[(\exists x \lor \exists y) \cdot (x=y) \cdot =(x) : X \rightarrow X'?
\]

No: not uniform! \[(=(x), s_0, \text{II})\]
Dynamic Dependence Logic

$$M = \{0,1\}$$

$$X = \begin{array}{ccc}
  & x & y & \ldots \\
 s_0 & 0 & 0 & \ldots \\
 s_1 & 0 & 1 & \ldots 
\end{array}$$

$$X' = \begin{array}{ccc}
  & x & y & \ldots \\
 s'_0 & 0 & 0 & \ldots \\
 s'_1 & 1 & 1 & \ldots 
\end{array}$$

$$(\exists x \lor \exists y) \cdot (x=y) \cdot =x : X \rightarrow X'$$

No: not uniform! $$(=x, s_0, \text{II})$$
Dynamic Dependence Logic

\( M = \{0, 1\} \)

<table>
<thead>
<tr>
<th></th>
<th>x</th>
<th>y</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>( s_0 )</td>
<td>0</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>( s_1 )</td>
<td>0</td>
<td>1</td>
<td>...</td>
</tr>
</tbody>
</table>

\( X = \)

<table>
<thead>
<tr>
<th></th>
<th>x</th>
<th>y</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>( s'_0 )</td>
<td>0</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>( s'_1 )</td>
<td>1</td>
<td>1</td>
<td>...</td>
</tr>
</tbody>
</table>

\( X' = \)

\((\exists x \lor \exists y) \cdot (x=y) \cdot (=x), s_1, \II) (s \in X)\)

\((\exists x \lor \exists y) \cdot (x=y), (=x), s_1, \II) (s \in X)\)

\((\exists x \lor \exists y), (x=y), (=x), s, \II)\)

II chooses L

\((\exists x, (x=y), (=x), s, \II)\)

II chooses R

\((\exists y, (x=y), (=x), s, \II)\)

II picks n

\( s' = s[n/x] \)

\((x=y), s', \II)\)

\( s(x) = s(y)\)

\( (=x), s', \II)\)

\( (\epsilon, s', \II)\)

\((\exists x \lor \exists y) \cdot (x=y) \cdot (=x): X \to X'\)?
Dynamic Dependence Logic

\[ \mathcal{M} = \{0, 1\} \]

\[ \mathcal{X} = \]

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>x</td>
<td>y</td>
</tr>
<tr>
<td>s_0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>s_1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

\[ \mathcal{X}' = \]

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>x</td>
<td>y</td>
</tr>
<tr>
<td>s'_0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>s'_1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

\[ (\exists x \vee \exists y) \cdot (x=y) \cdot = (x), s, \ II \]

(\exists x \vee \exists y) \cdot (x=y), = (x), s_1, \ II \ (s \in \mathcal{X})

II chooses L

(\exists x \cdot (x=y), = (x), s, \ II)

II chooses R

(\exists y \cdot (x=y), = (x), s, \ II)

II picks \( n \)

\[ s' = s[s(y)/x] \]

\[ s(x) = s(y) \]

(\exists x \cdot (x=y), s, \ II)

(= (x), s_0, \ II)

No: not uniform!
Dynamic Dependence Logic

\[ M = \{0, 1\} \]

\[ X = \]
\[
\begin{array}{c|cc|c|c|c}
  & x & y & \ldots \\
\hline
s_0 & 0 & 0 & \ldots \\
\hline
s_1 & 0 & 1 & \ldots \\
\end{array}
\]

\[ X' = \]
\[
\begin{array}{c|cc|c|c|c}
  & x & y & \ldots \\
\hline
s'_0 & 0 & 0 & \ldots \\
\hline
s'_1 & 1 & 1 & \ldots \\
\end{array}
\]

\[(\exists x \vee \exists y). (x=y) \cdot =(x): X \rightarrow X'\]

No: not uniform!  \[(=(x), s_0, \text{II})\]
### Dynamic Dependence Logic

**M** = ({0,1})

<table>
<thead>
<tr>
<th></th>
<th>x</th>
<th>y</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>s₀</td>
<td>0</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>s₁</td>
<td>0</td>
<td>1</td>
<td>...</td>
</tr>
</tbody>
</table>

**X** =

<table>
<thead>
<tr>
<th></th>
<th>x</th>
<th>y</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>s₀</td>
<td>0</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>s₁</td>
<td>0</td>
<td>1</td>
<td>...</td>
</tr>
</tbody>
</table>

**X’** =

<table>
<thead>
<tr>
<th></th>
<th>x</th>
<th>y</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>s₀’</td>
<td>0</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>s₁’</td>
<td>1</td>
<td>1</td>
<td>...</td>
</tr>
</tbody>
</table>

(∃x ∨ ∃y) · (x=y) · =(x): X → X’?

No: not uniform!  

(=x), s₀, II

((∃x ∨ ∃y) · (x=y) · =(x), s₁, II) (s ∈ X)

II chooses L

((∃x ∨ ∃y) · (x=y), =(x), s₁, II) (s ∈ X)

II chooses R

((∃x ∨ ∃y), (x=y), =(x), s₁, II)

II picks n  
s’ = s[n/x]

s’ = s₁[s₁(y)/x]

((x=y), s’, II)

s(x) = s(y)

(=x), s’, II

(ε, s’, II)
**Dynamic Dependence Logic**

\[ M = \{0,1\} \]

\[ X = \]

\[
\begin{array}{ccc}
\hline
 & x & y & \ldots \\
\hline
s_0 & 0 & 0 & \ldots \\
\hline
s_1 & 0 & 1 & \ldots \\
\hline
\end{array}
\]

\[ X' = \]

\[
\begin{array}{ccc}
\hline
 & x & y & \ldots \\
\hline
s'_0 & 0 & 0 & \ldots \\
\hline
s'_1 & 1 & 1 & \ldots \\
\hline
\end{array}
\]

\[(\exists x \lor \exists y) \cdot (x=y) \cdot = (x), s_1, \text{II} \quad (s \in X)\]

\[(\exists x \lor \exists y) \cdot (x=y), = (x), s_1, \text{II} \quad (s \in X)\]

\[(\exists x \lor \exists y), (x=y), = (x), s_1, \text{II} \]

\[ s' = s_1[1/x] \]

\[ (\exists x, (x=y), = (x), s_1, \text{II}) \]

\[ (\exists y, (x=y), = (x), s, \text{II}) \]

\[ \text{II chooses L} \]

\[ \text{II chooses R} \]

\[ \text{II picks n} \]

\[ s' = s[n/x] \]

\[ ((x=y), s', \text{II}) \]

\[ s(x) = s(y) \]

\[ (= (x), s', \text{II}) \]

\[ (= (x), s_0, \text{II}) \]

\[ (\varepsilon, s', \text{II}) \]
Dynamic Dependence Logic

\[ M = \{0,1\} \]

\[
\begin{array}{ccc}
X = & x & y & \ldots \\
  s_0 & 0 & 0 & \ldots \\
  s_1 & 0 & 1 & \ldots \\
\end{array}
\]

\[
\begin{array}{ccc}
X' = & x & y & \ldots \\
 s'_0 & 0 & 0 & \ldots \\
 s'_1 & 1 & 1 & \ldots \\
\end{array}
\]

\[(\exists x \lor \exists y) \cdot (x=y) \cdot =(x), s_1, \text{II} (s \in X)\]

\[
(\exists x \lor \exists y) \cdot (x=y), =\langle x \rangle, (x=y), =\langle x \rangle, (x=y), =\langle x \rangle
\]

\[
(\exists x \lor \exists y) \cdot (x=y), =\langle x \rangle, (x=y), =\langle x \rangle, (x=y), =\langle x \rangle
\]

\[
(\exists x \lor \exists y) \cdot (x=y), =\langle x \rangle, (x=y), =\langle x \rangle, (x=y), =\langle x \rangle
\]

\[(\exists x \lor \exists y) \cdot (x=y), =\langle x \rangle, (x=y), =\langle x \rangle, (x=y), =\langle x \rangle
\]

\[
(\exists x \lor \exists y) \cdot (x=y), =\langle x \rangle, (x=y), =\langle x \rangle, (x=y), =\langle x \rangle
\]

\[
(\exists x \lor \exists y) \cdot (x=y), =\langle x \rangle, (x=y), =\langle x \rangle, (x=y), =\langle x \rangle
\]

(\exists x \lor \exists y) \cdot (x=y) \cdot =(x): X \to X'?

No: not uniform!  
\[
(=\langle x \rangle, s_0, \text{II})
\]

\[
(\forall x \lor \forall y) \cdot (x=y) \cdot =\langle x \rangle, s_1, \text{II} (s \in X)
\]

\[
(\forall x \lor \forall y) \cdot (x=y), =\langle x \rangle, (x=y), =\langle x \rangle, (x=y), =\langle x \rangle
\]

\[
(\forall x \lor \forall y) \cdot (x=y), =\langle x \rangle, (x=y), =\langle x \rangle, (x=y), =\langle x \rangle
\]

\[
(\forall x \lor \forall y) \cdot (x=y), =\langle x \rangle, (x=y), =\langle x \rangle, (x=y), =\langle x \rangle
\]

\[
(\forall x \lor \forall y) \cdot (x=y), =\langle x \rangle, (x=y), =\langle x \rangle, (x=y), =\langle x \rangle
\]

(\exists x \lor \exists y) \cdot (x=y) \cdot =(x): X \to X'?
**Dynamic Dependence Logic**

\[
M = \{0, 1\}
\]

\[
X =
\begin{array}{ccc}
\hline
x & y & \ldots \\
\hline
s_0 & 0 & 0 & \ldots \\
\hline
s_1 & 0 & 1 & \ldots \\
\hline
\end{array}
\]

\[
X' =
\begin{array}{ccc}
\hline
x & y & \ldots \\
\hline
s'_0 & 0 & 0 & \ldots \\
\hline
s'_1 & 1 & 1 & \ldots \\
\hline
\end{array}
\]

\[(\exists x \lor \exists y) \cdot (x = y) \cdot = (x), s_1, \text{II} (s \in X)\]

\[(\exists x \lor \exists y) \cdot (x = y), = (x), s_1, \text{II} (s \in X)\]

\[(\exists x \lor \exists y), (x = y), = (x), s_1, \text{II}\]

\[\text{II chooses L}\]

\[\text{II chooses R}\]

\[s' = s'_1\]

\[(\exists x, (x = y), = (x), s_1, \text{II})\]

\[(\exists y, (x = y), = (x), s, \text{II})\]

\[(\exists x \lor \exists y), (x = y), = (x), s_1, \text{II}\]

\[s'_1(x) = s'_1(y)\]

\[= (x), s'_1, \text{II}\]

\[
(\exists x \lor \exists y). (x = y). = (x): X \rightarrow X'?
\]

No: not uniform! \[= (x), s_0, \text{II}\]
Dynamic Dependence Logic

\[ M = \{0,1\} \]

<table>
<thead>
<tr>
<th>(x)</th>
<th>(y)</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>...</td>
</tr>
</tbody>
</table>

\[ X = \]

\[ X' = \]

\[ (\exists x \lor \exists y). (x=y). = (x), s_1, \text{II} \quad (s \in X) \]

II chooses L

\[ ((\exists x \lor \exists y). (x=y), = (x), s_1, \text{II}) \quad (s \in X) \]

II chooses R

\[ (\exists y, (x=y), = (x), s_1, \text{II}) \]

II picks \(n\)

\[ s' = s'_1 \]

\[ s'_1(x) = s'_1(y) \]

\[ (x=y), s'_1, \text{II} \]

\[ (= (x), s_0, \text{II}) \]

\[ (= (x), s'_1, \text{II}) \]

\[ (\varepsilon, s', \text{II}) \]

No: not uniform!
Dynamic Dependence Logic

\[ M = \{0, 1\} \]

\[ X = \]

\[
\begin{array}{ccc}
\hline
x & y & \ldots \\
\hline
s_0 & 0 & 0 & \ldots \\
s_1 & 0 & 1 & \ldots \\
\hline
\end{array}
\]

\[ X' = \]

\[
\begin{array}{ccc}
\hline
x & y & \ldots \\
\hline
s'_0 & 0 & 0 & \ldots \\
s'_1 & 1 & 1 & \ldots \\
\hline
\end{array}
\]

\[ (\exists x \lor \exists y) \cdot (x=y) \cdot = (x) : X \rightarrow X' ? \]

No: not uniform!

\[ (= (x), s_0, \text{II}) \]

\[ (= (x), s'_1, \text{II}) \]

\[ s'_1 (x) = s'_1 (y) \]

\[ (= (x), s'_1, \text{II}) \]

\[ (\varepsilon, s'_1, \text{II}) \]
Dynamic Dependence Logic

\[ M = \{0,1\} \]

\[ X = \]

\[
\begin{array}{cccc}
  & x & y & \ldots \\
 s_0 & 0 & 0 & \ldots \\
 s_1 & 0 & 1 & \ldots \\
\end{array}
\]

\[ X' = \]

\[
\begin{array}{cccc}
  & x & y & \ldots \\
 s'_0 & 0 & 0 & \ldots \\
 s'_1 & 1 & 1 & \ldots \\
\end{array}
\]

\((\exists x \lor \exists y) \cdot (x=y) \cdot =(x), s_1, \ II \) \((s \in X)\)

II chooses L

\((\exists x \lor \exists y), (x=y), =(x), s_1, \ II\)

II chooses R

\((\exists y, (x=y), =(x), s, \ II)\)

II picks \(n\)

\(s' = s'_1\)

\((x=y), s'_1, \ II\)

\((=x), s'_1, \ II\)

\(s'_1(x) = s'_1(y)\)

\((=(x), s'_1, \ II)\)

\((\varepsilon, s'_1, \ II)\)

\(s_0(x) \neq s'_1(x)!\)

\((=(x), s_0, \ II)\)

\((=(x), s'_1, \ II)\)
Dynamic Dependence Logic

Thank you!