Dynamic Dependence Logic

Pietro Galliani
Universiteit van Amsterdam
ILLC
LINT (**L**ogic for **INT**eraction) 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

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

$$\begin{pmatrix} \forall x \ \exists y \\ \forall z \ \exists w \end{pmatrix} \phi(x,y,z,w)$$

Dynamic Dependence Logic

(Henkin, 1961): Branching Quantifiers

$$\begin{pmatrix} \forall x \; \exists y \\ \forall z \; \exists w \end{pmatrix} \phi(x,y,z,w)$$

y may depend on x, but not on z or w;
w may depend on z, but not on x or y;

Dynamic Dependence Logic

(Henkin, 1961): Branching Quantifiers

$$\begin{pmatrix} \forall x \; \exists y \\ \forall z \; \exists w \end{pmatrix} \phi(x,y,z,w)$$

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 ϕ may depend on x, y, z, and w.

Dynamic Dependence Logic

$$\exists u \begin{pmatrix} \forall x \; \exists y \\ \forall z \; \exists w \end{pmatrix} (x=z \leftrightarrow y=w \wedge y \neq u)$$

Dynamic Dependence Logic

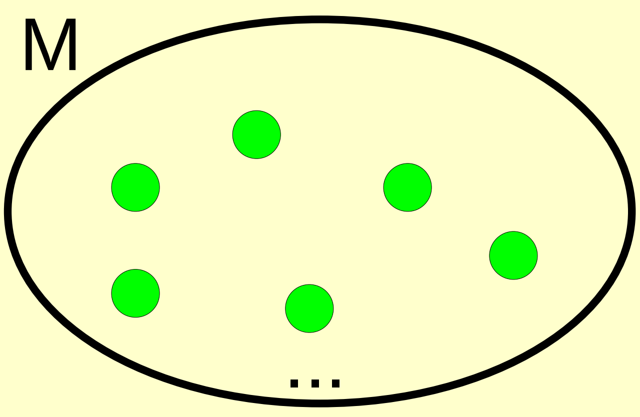
$$\exists u \begin{pmatrix} \forall x\ \exists y \\ \forall z\ \exists w \end{pmatrix} (x=z \leftrightarrow y=w \wedge y \neq u)$$

$$\exists u\ \exists f_u\ \exists g_u\ \forall x\ \forall z\ (x=z \leftrightarrow f_u(x)=g_u(z) \wedge f_u(x) \neq u)$$

Dynamic Dependence Logic

$$\exists u \begin{pmatrix} \forall x \; \exists y \\ \forall z \; \exists w \end{pmatrix} (x=z \leftrightarrow y=w \wedge y \neq u)$$

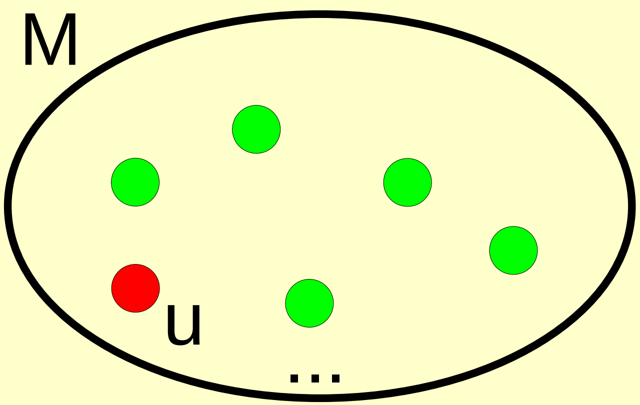
$$\exists u \; \exists f_u \; \exists g_u \; \forall x \; \forall z \; (x=z \leftrightarrow f_u(x)=g_u(z) \wedge f_u(x) \neq u)$$



Dynamic Dependence Logic

$$\exists u \begin{pmatrix} \forall x \; \exists y \\ \forall z \; \exists w \end{pmatrix} (x=z \leftrightarrow y=w \wedge y \neq u)$$

$$\color{red}{\exists u} \; \exists f_u \; \exists g_u \; \forall x \; \forall z \; (x=z \leftrightarrow f_u(x)=g_u(z) \wedge f_u(x) \neq u)$$

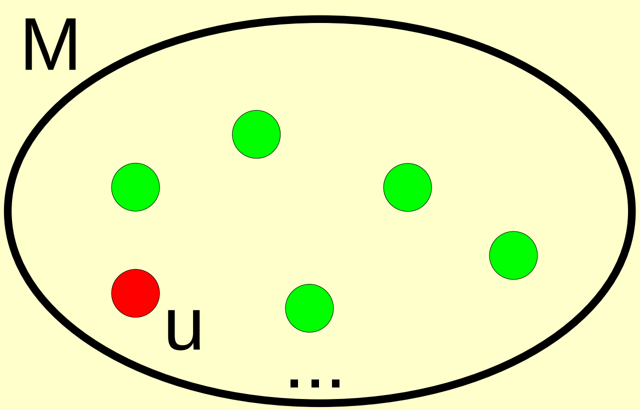


Dynamic Dependence Logic

$$\exists u \begin{pmatrix} \forall x\ \exists y \\ \forall z\ \exists w \end{pmatrix} (x=z \leftrightarrow y=w \wedge y \neq u)$$

$$\exists u\ \exists f_u\ \exists g_u\ \forall x\ \forall z\ (x=z \leftrightarrow f_u(x)=g_u(z) \wedge f_u(x) \neq u)$$

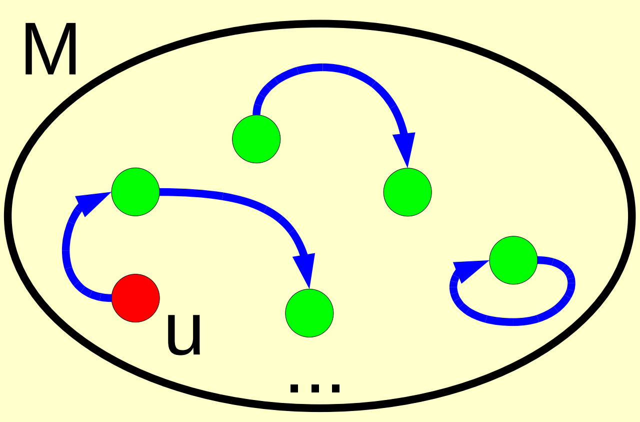
$f_u, g_u: M \to M;$



Dynamic Dependence Logic

$$\exists u \begin{pmatrix} \forall x \ \exists y \\ \forall z \ \exists w \end{pmatrix} (x=z \leftrightarrow y=w \wedge y \neq u)$$

$$\exists u \ \exists f_u \ \exists g_u \ \forall x \ \forall z \ (x=z \leftrightarrow f_u(x)=g_u(z) \wedge f_u(x) \neq u)$$



$f_u, g_u: M \rightarrow M$;

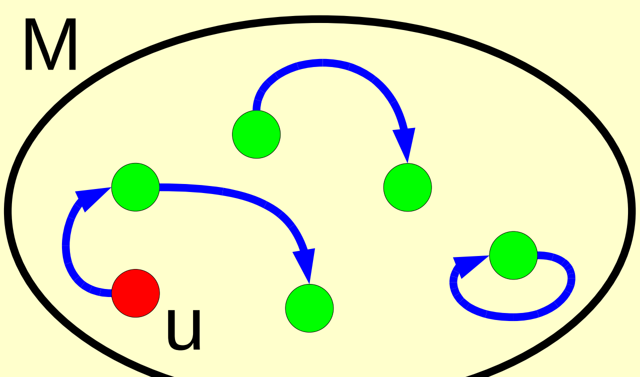
$f_u = g_u$;

f_u is 1-1;

Dynamic Dependence Logic

$$\exists u \begin{pmatrix} \forall x\ \exists y \\ \forall z\ \exists w \end{pmatrix} (x=z \leftrightarrow y=w \wedge y \neq u)$$

$$\exists u\ \exists f_u\ \exists g_u\ \forall x\ \forall z\ (x=z \leftrightarrow f_u(x)=g_u(z) \wedge f_u(x) \neq u)$$



$f_u, g_u: M \to M$;

$f_u = g_u$;

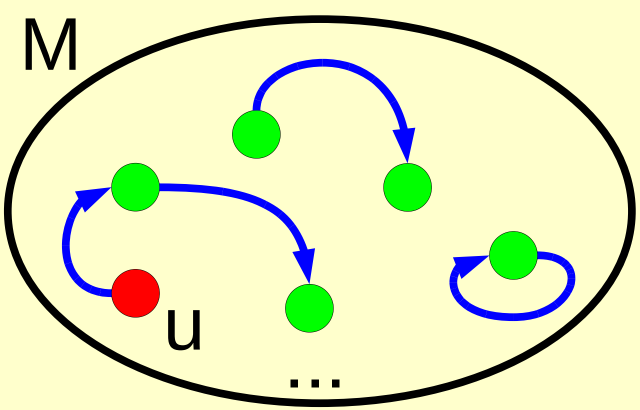
f_u is 1-1;

$u \notin \mathrm{Range}(f_u)$.

Dynamic Dependence Logic

$$\exists u \begin{pmatrix} \forall x\ \exists y \\ \forall z\ \exists w \end{pmatrix} (x=z \leftrightarrow y=w \wedge y \neq u)$$

$$\exists u\ \exists f_u\ \exists g_u\ \forall x\ \forall z\ (x=z \leftrightarrow f_u(x)=g_u(z) \wedge f_u(x) \neq u)$$



$f_u, g_u: M \rightarrow M$;

$f_u = g_u$;

f_u is 1-1;

$u \notin \text{Range}(f_u)$.

Therefore, M is infinite

Dynamic Dependence Logic

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*:

$$\begin{pmatrix} \forall x \; \exists y \\ \forall z \; \exists w \end{pmatrix} \phi(x,y,z,w) \equiv \exists f \exists g \; \forall x \forall z \; \phi(x,f(x),z,g(z))$$

Dynamic Dependence Logic

But what does “dependence between connectives” *mean*?

1. Explanation in terms of *Skolem functions*:

$$\begin{pmatrix} \forall x\ \exists y \\ \forall z\ \exists w \end{pmatrix} \phi(x,y,z,w) \equiv \exists f \exists g\ \forall x \forall z\ \phi(x,f(x),z,g(z))$$

Gives a translation from branching quantifiers to Σ^1_1

Dynamic Dependence Logic

But what does “dependence between connectives” *mean*?

1. Explanation in terms of *Skolem functions*:

$$\begin{pmatrix} \forall x\ \exists y \\ \forall z\ \exists w \end{pmatrix} \phi(x,y,z,w) \equiv \exists f \exists g\ \forall x \forall z\ \phi(x,f(x),z,g(z))$$

Gives a translation from branching quantifiers to Σ^1_1

Not a semantics!

Dynamic Dependence Logic

But what does “dependence between connectives” *mean*?

1. Explanation in terms of *Skolem functions*:

$$\begin{pmatrix}\forall x\ \exists y \\ \forall z\ \exists w\end{pmatrix} \phi(x,y,z,w) \equiv \exists f \exists g\ \forall x \forall z\ \phi(x,f(x),z,g(z))$$

Gives a translation from branching quantifiers to Σ^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:*

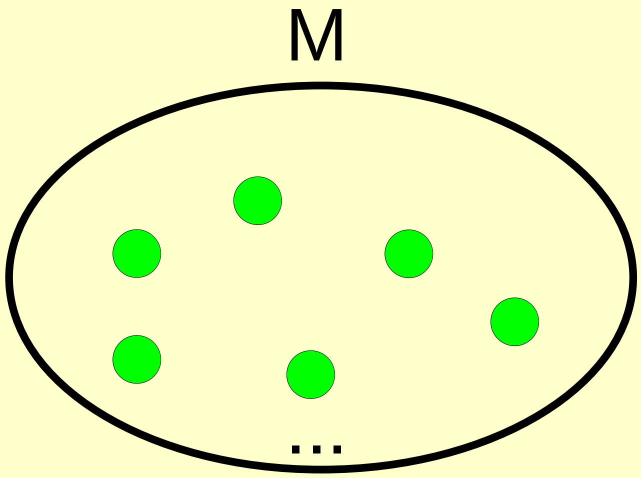
$$\forall x\ \exists y\ \forall z\ \exists w\ (x=w \vee y=z)$$

Dynamic Dependence Logic

But what does “dependence between connectives” *mean*?

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

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

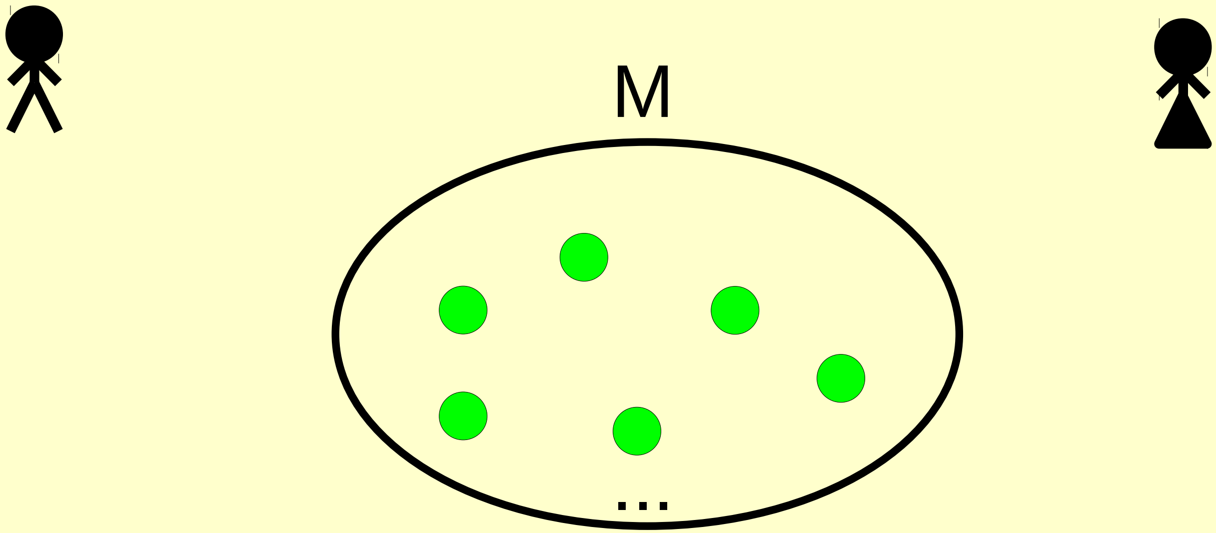


Dynamic Dependence Logic

But what does "dependence between connectives" *mean*?

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

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

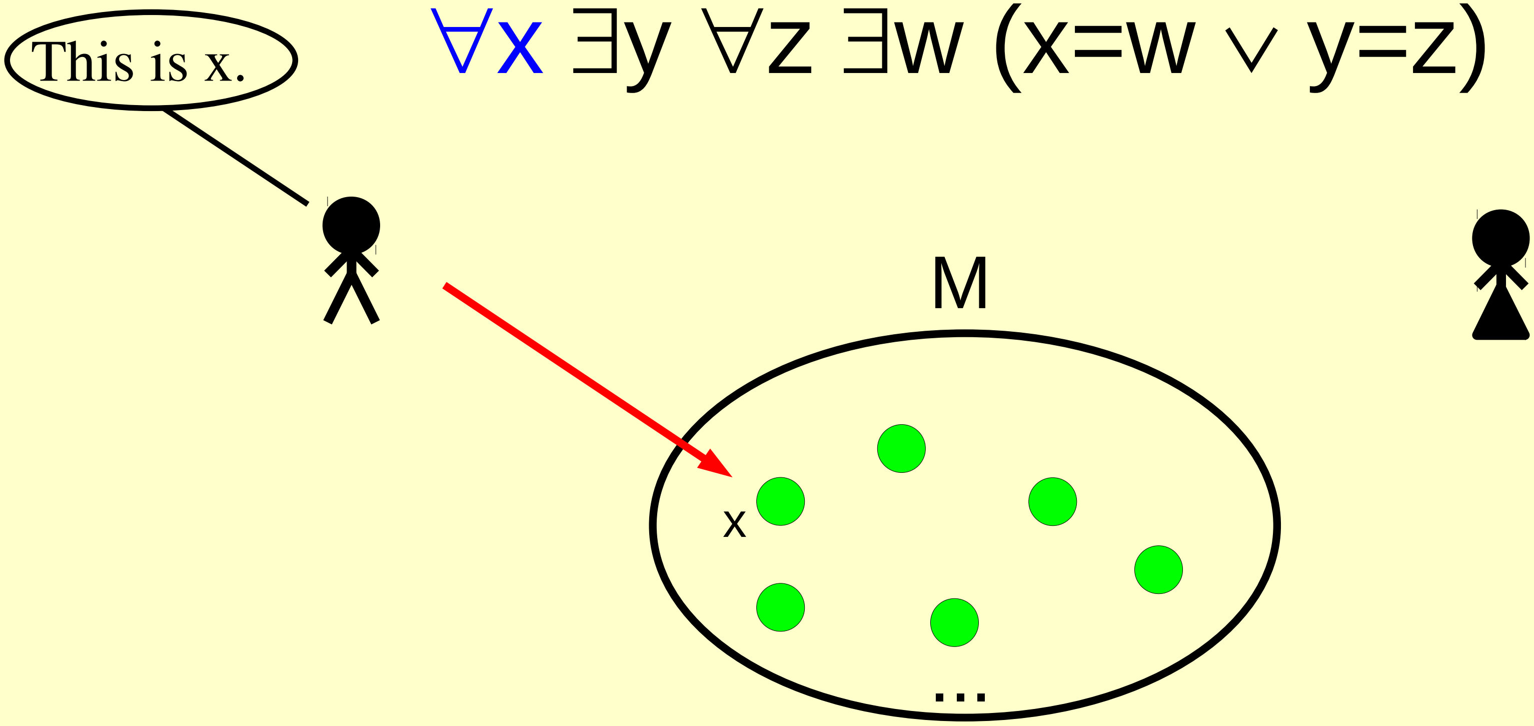


Dynamic Dependence Logic

But what does “dependence between connectives” *mean*?

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

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

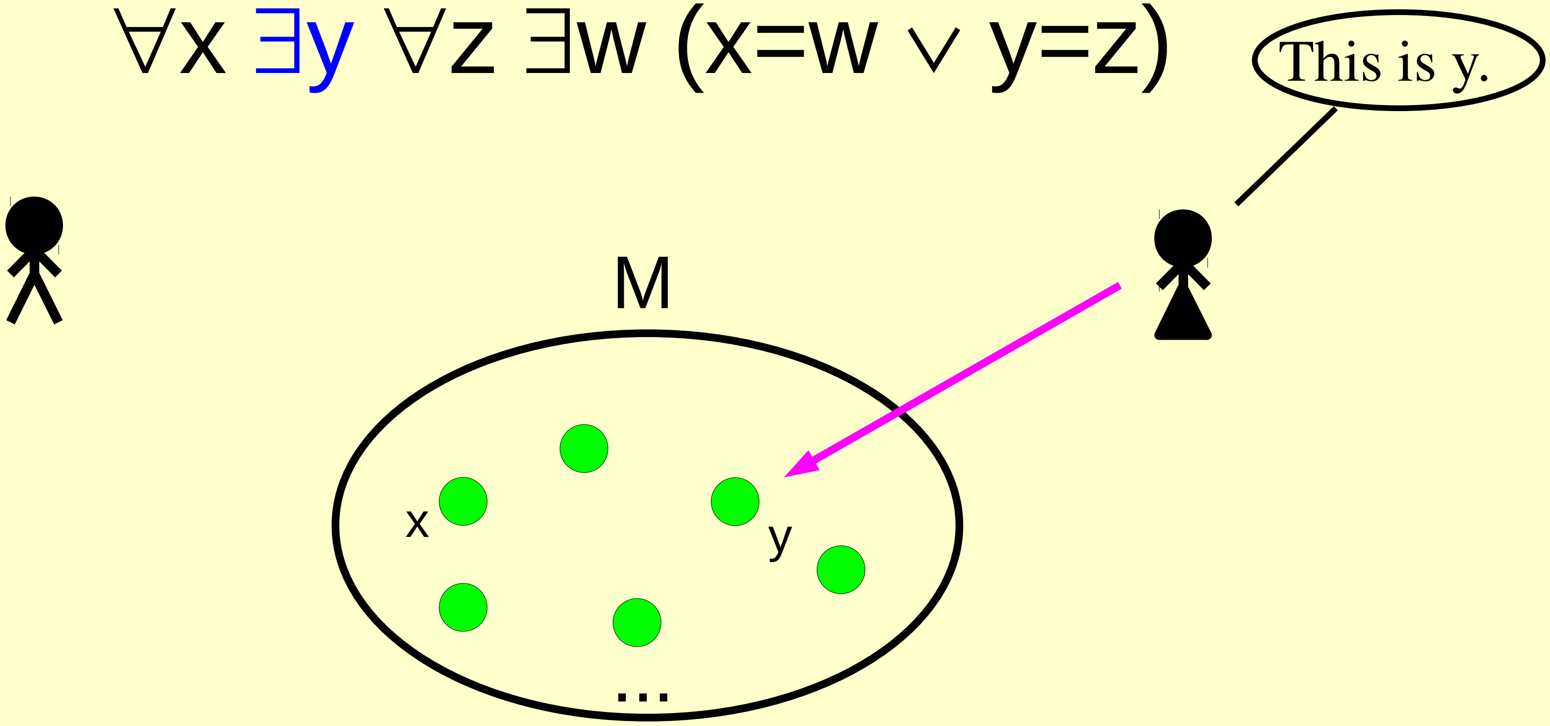


Dynamic Dependence Logic

But what does “dependence between connectives” *mean*?

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

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

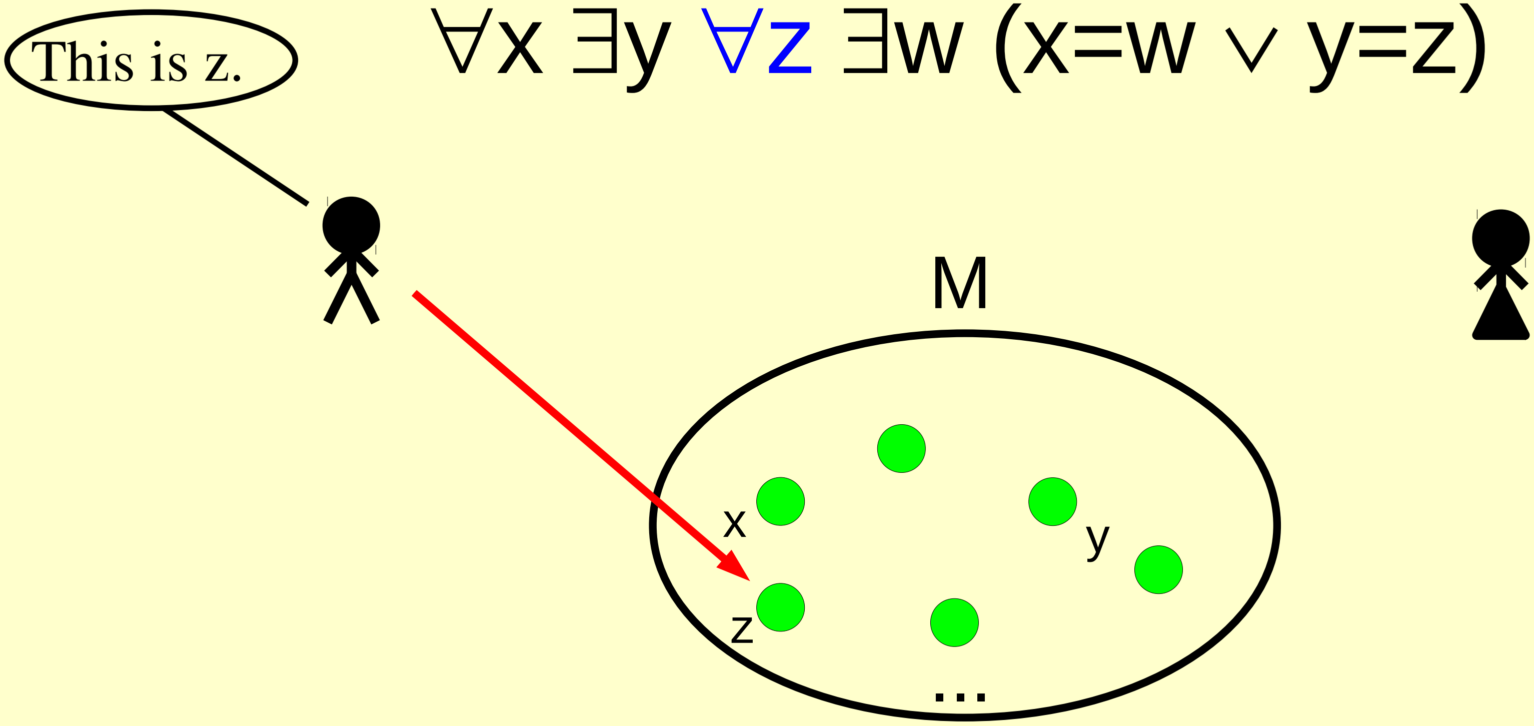


Dynamic Dependence Logic

But what does “dependence between connectives” *mean*?

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

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

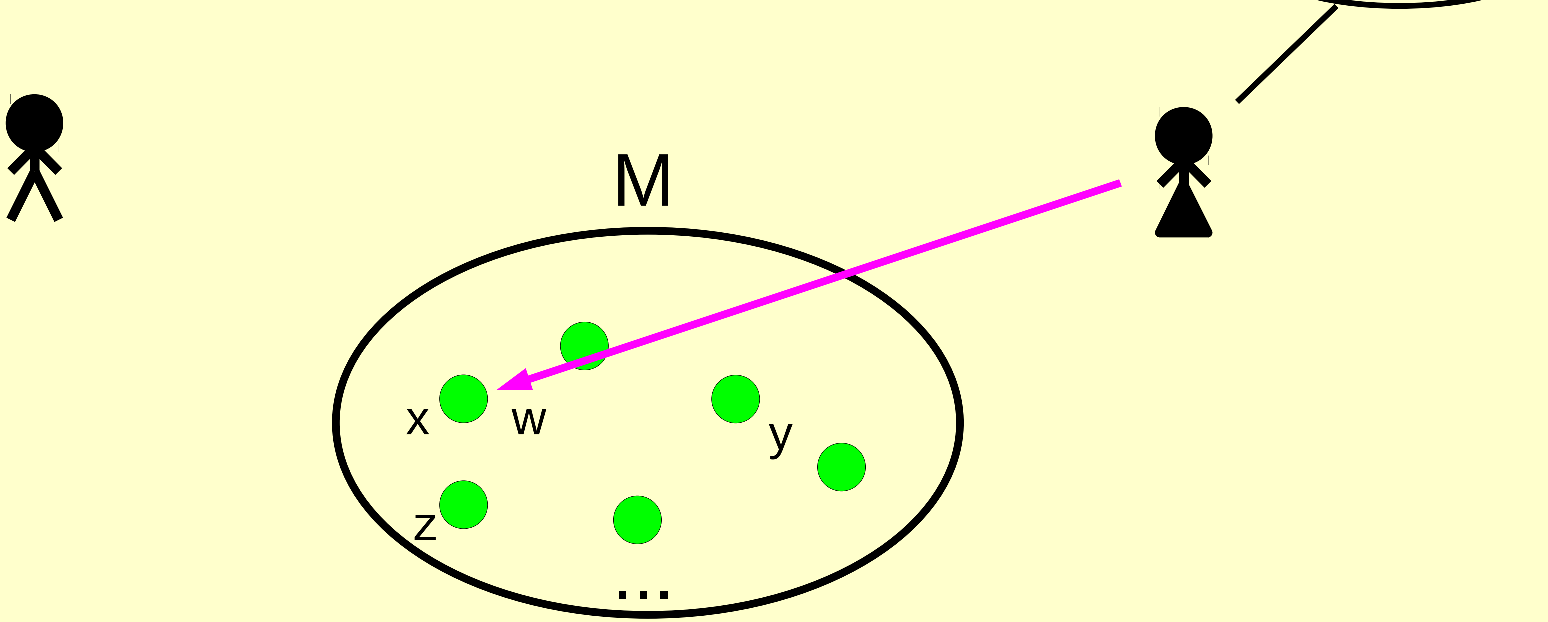


Dynamic Dependence Logic

But what does “dependence between connectives” *mean*?

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

$$\forall x\ \exists y\ \forall z\ \exists w\ (x{=}w \vee y{=}z)$$

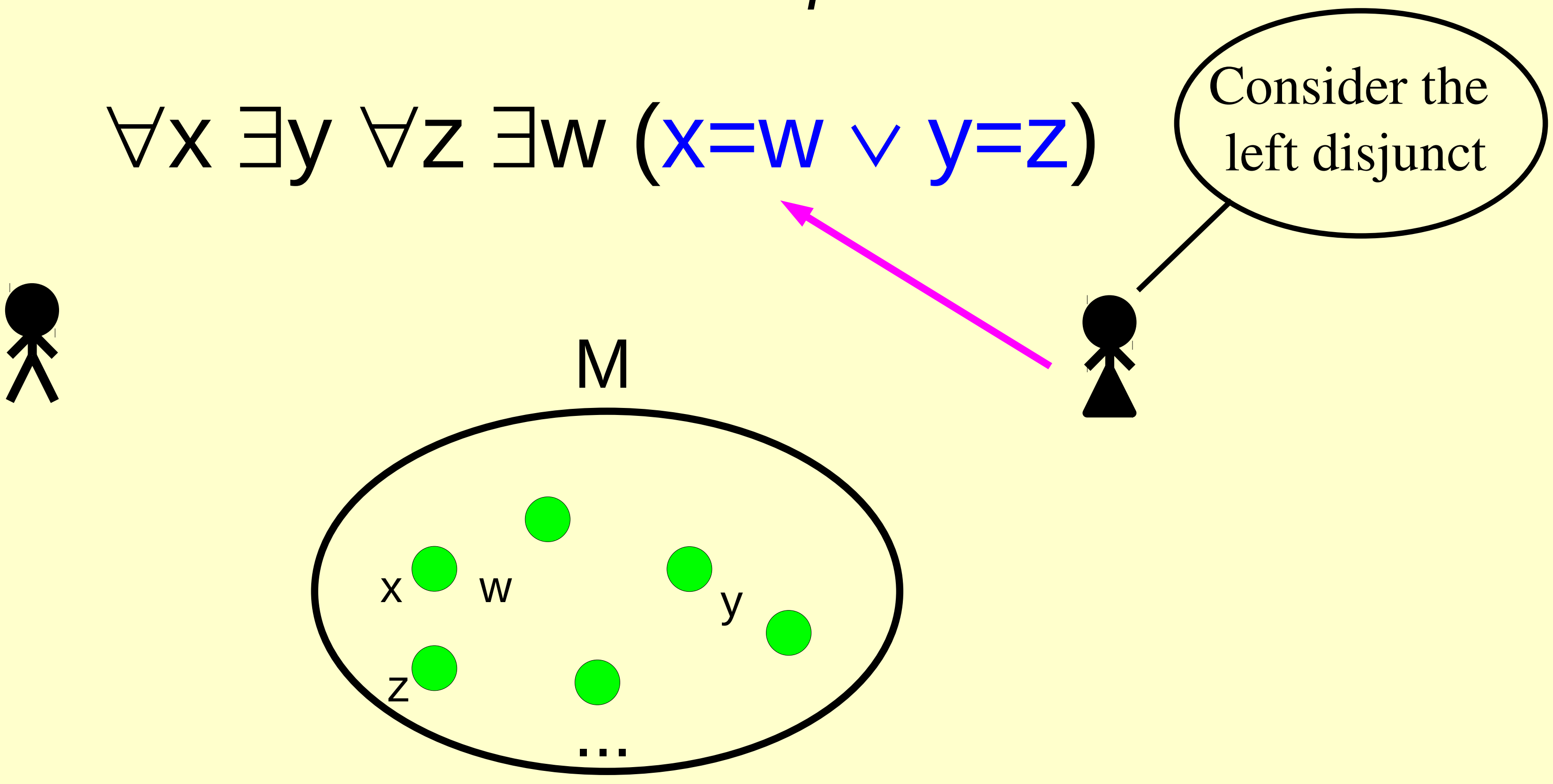


Dynamic Dependence Logic

But what does “dependence between connectives” *mean*?

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

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

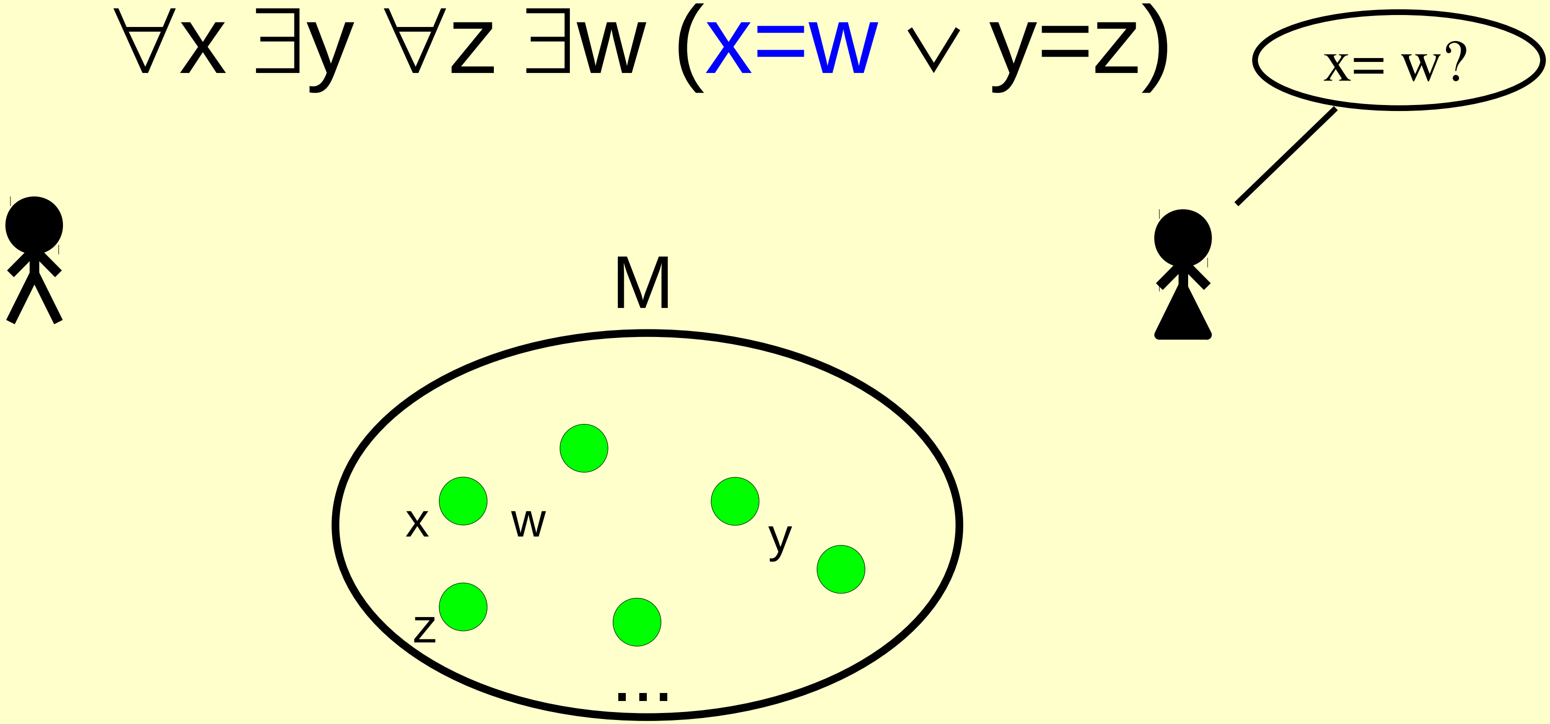


Dynamic Dependence Logic

But what does “dependence between connectives” *mean*?

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

$$\forall x\ \exists y\ \forall z\ \exists w\ ({\color{blue}x=w} \vee y=z)$$

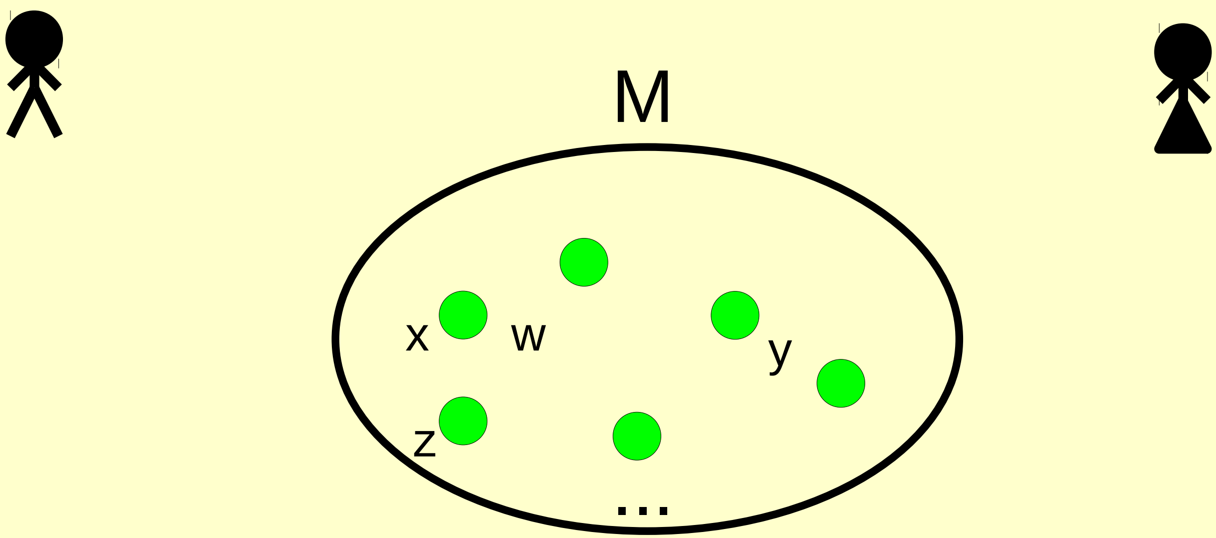


Dynamic Dependence Logic

But what does “dependence between connectives” *mean*?

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

$$\forall x\ \exists y\ \forall z\ \exists w\ ({\color{blue}x=w} \vee y=z)$$

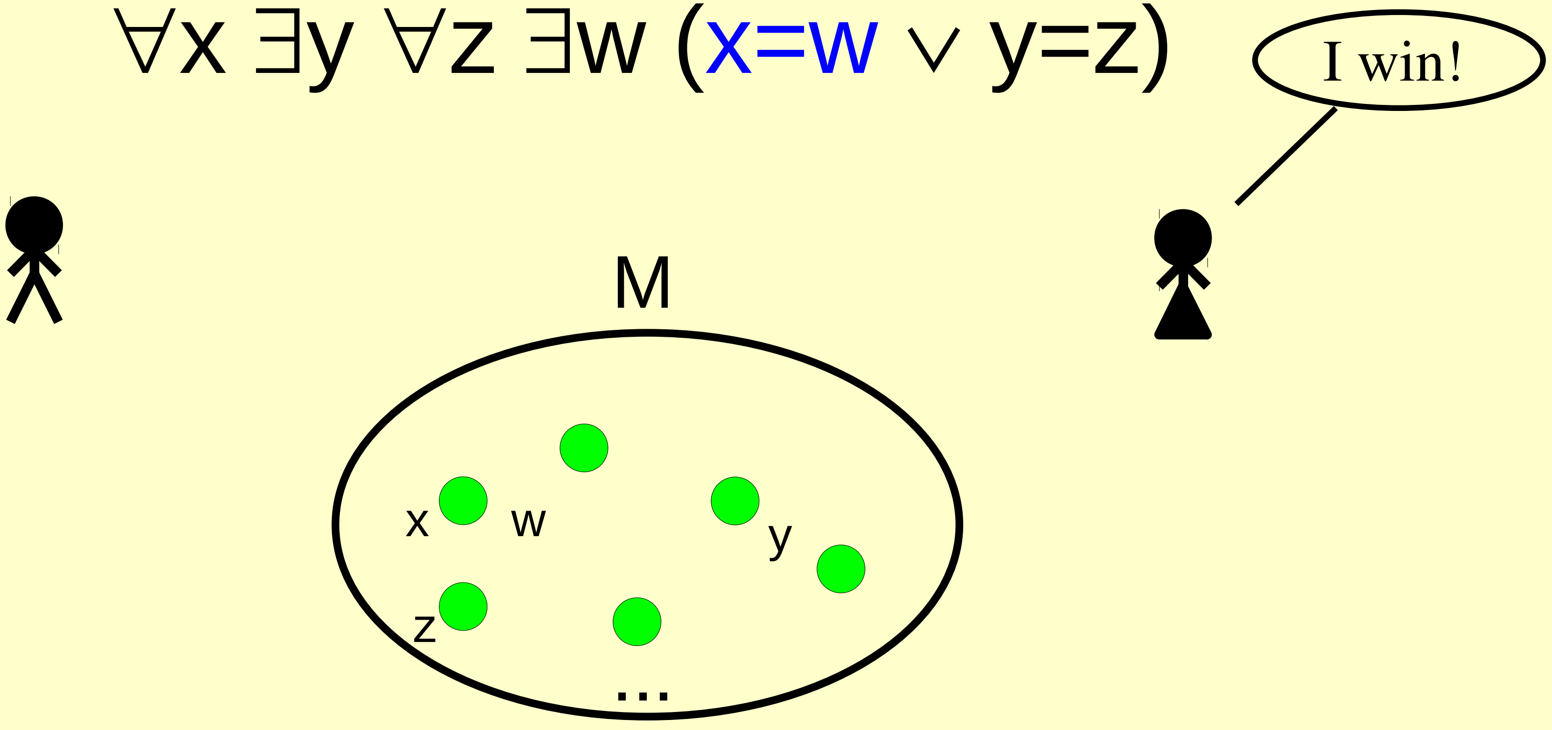


Dynamic Dependence Logic

But what does “dependence between connectives” *mean*?

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

$$\forall x\, \exists y\, \forall z\, \exists w\, ({\color{blue}x=w} \vee y=z)$$

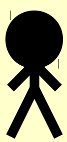


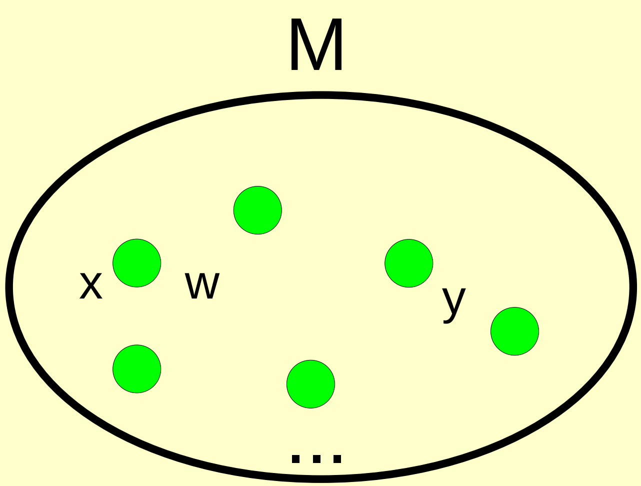
Dynamic Dependence Logic

But what does "dependence between connectives" *mean*?

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

$$\forall x \exists y \forall z \exists w \, (x=w \vee y=z)$$





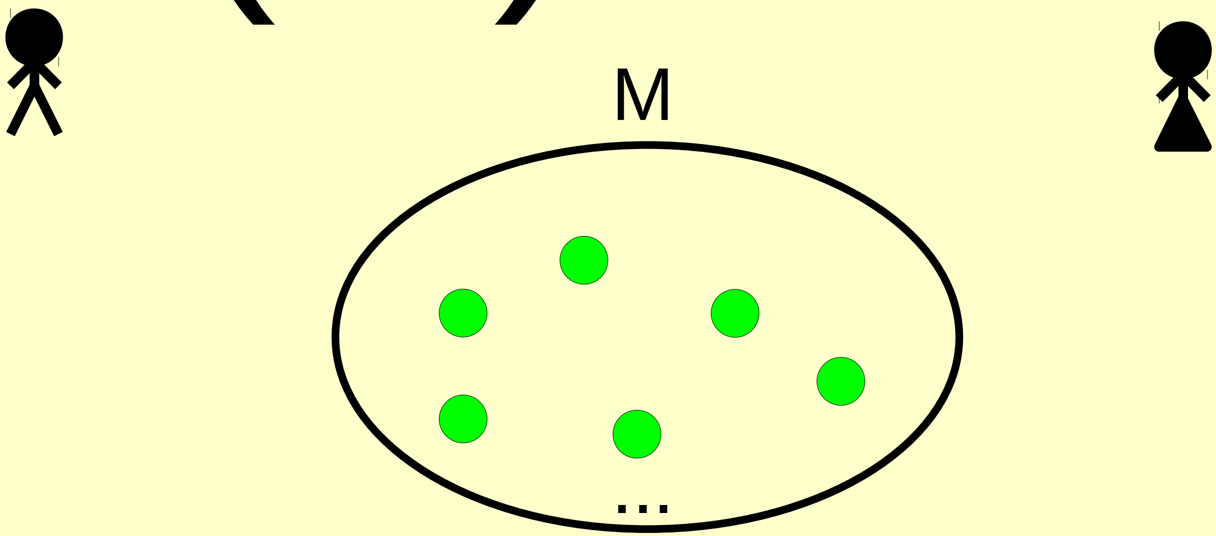
A FOL sentence is true in a model iff Eloise has a winning strategy for the corresponding game

Dynamic Dependence Logic

But what does “dependence between connectives” *mean*?

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

$$\begin{pmatrix}\forall x\ \exists y\\ \forall z\ \exists w\end{pmatrix}(x=w \vee y=z)$$

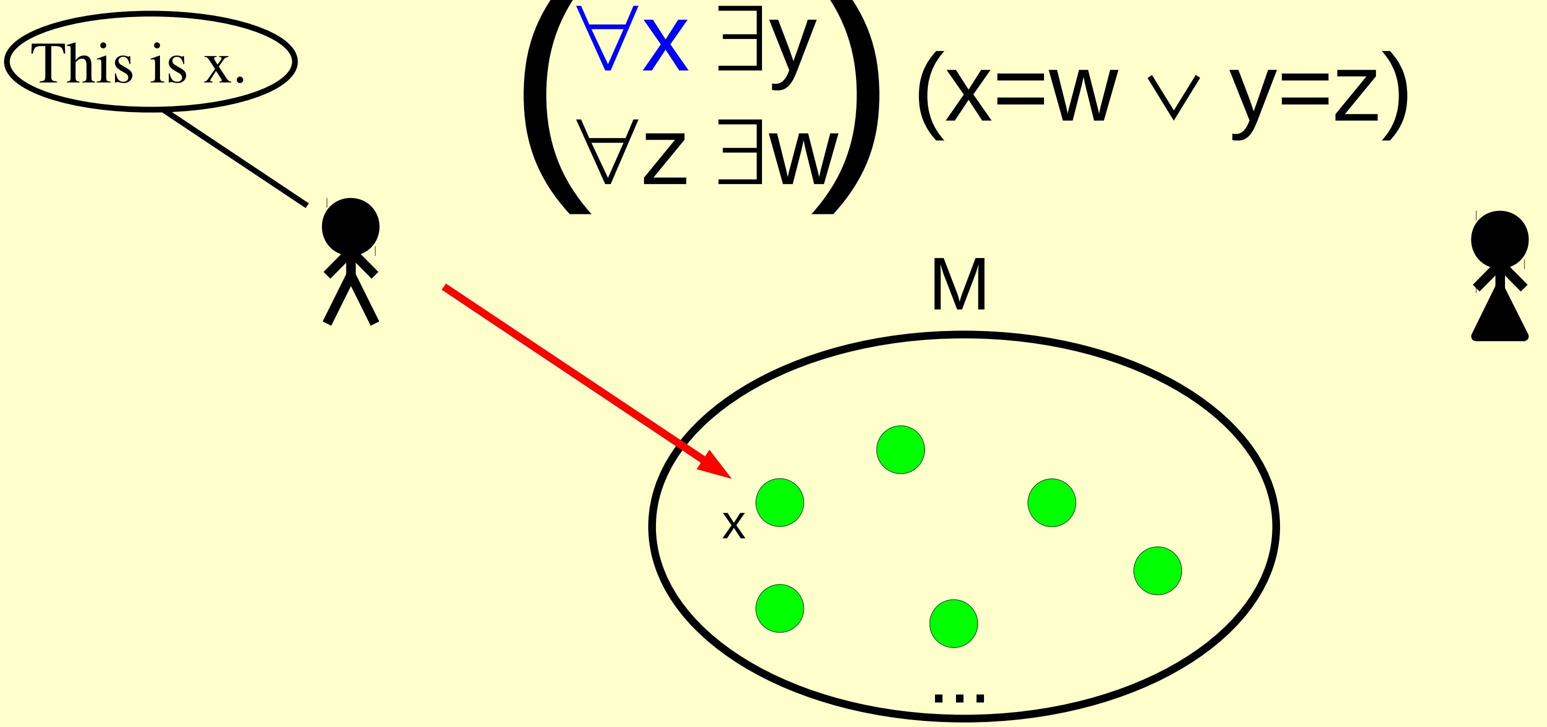


Dynamic Dependence Logic

But what does “dependence between connectives” *mean*?

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

$$\begin{pmatrix} \forall x & \exists y \\ \forall z & \exists w \end{pmatrix} (x=w \vee y=z)$$

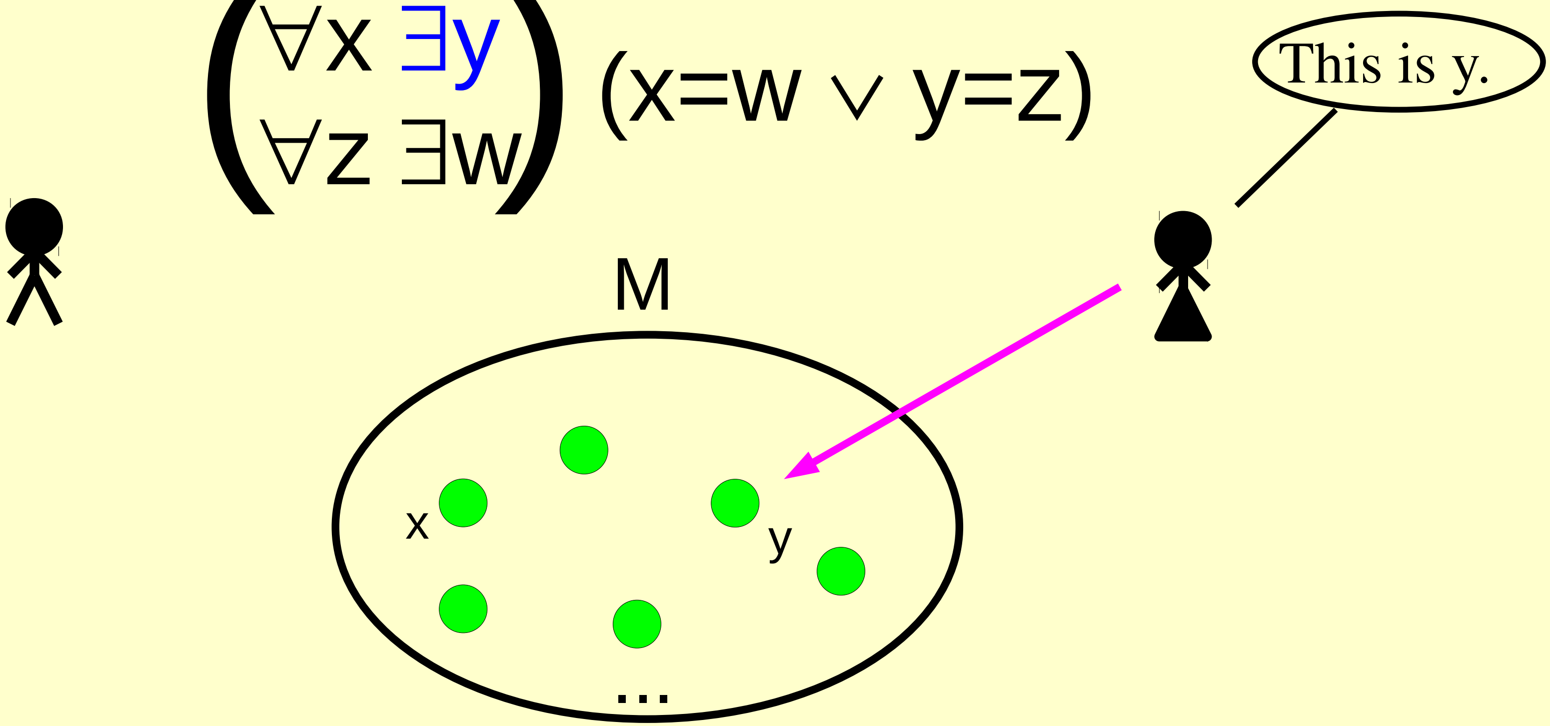


Dynamic Dependence Logic

But what does “dependence between connectives” *mean*?

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

$$\begin{pmatrix} \forall x \ {\color{blue}\exists y} \\ \forall z \ \exists w \end{pmatrix} (x=w \vee y=z)$$

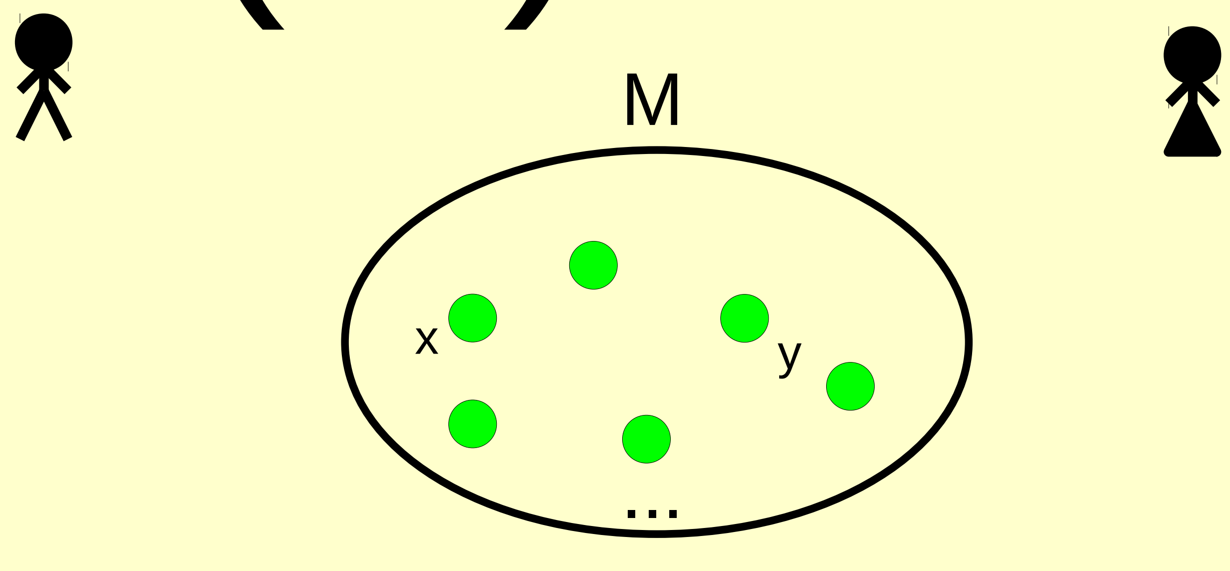


Dynamic Dependence Logic

But what does “dependence between connectives” *mean*?

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

$$\begin{pmatrix} \forall x \; \exists y \\ \forall z \; \exists w \end{pmatrix} (x=w \vee y=z)$$

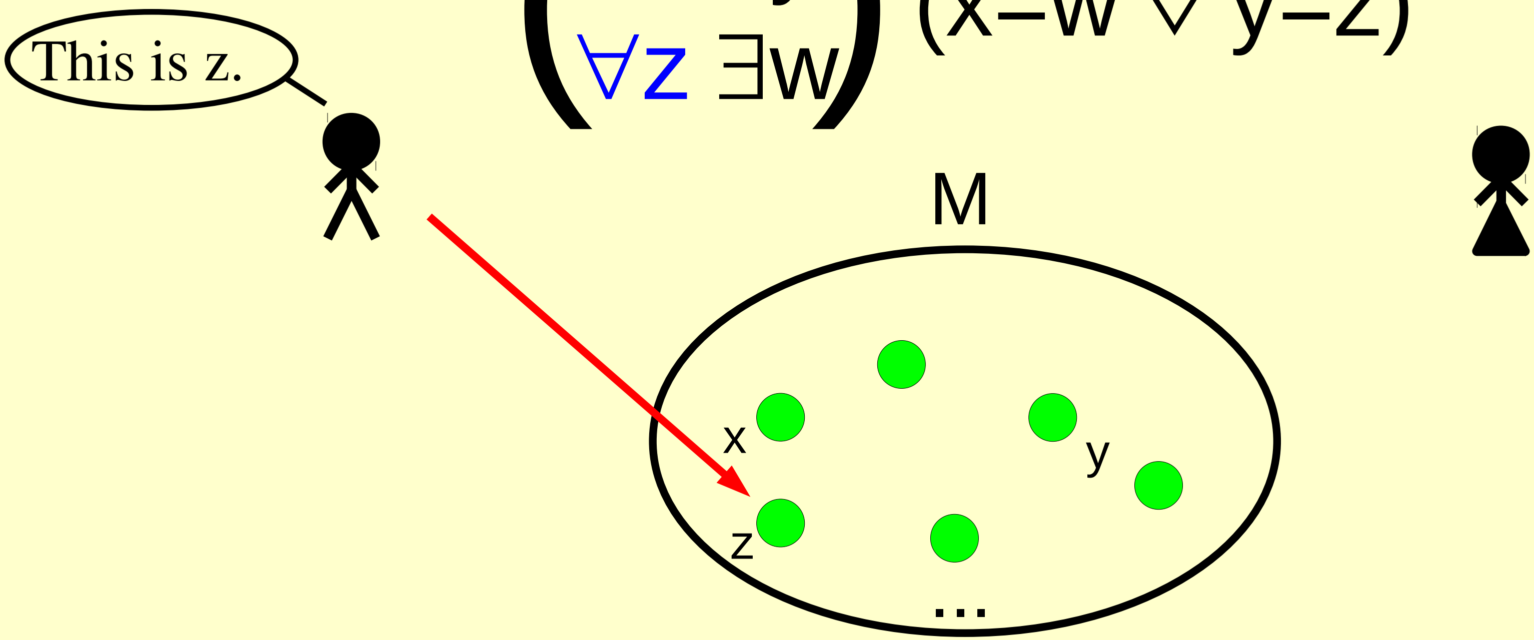


Dynamic Dependence Logic

But what does “dependence between connectives” *mean*?

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

$$\begin{pmatrix} \forall x \ \exists y \\ \forall z \ \exists w \end{pmatrix} (x=w \vee y=z)$$

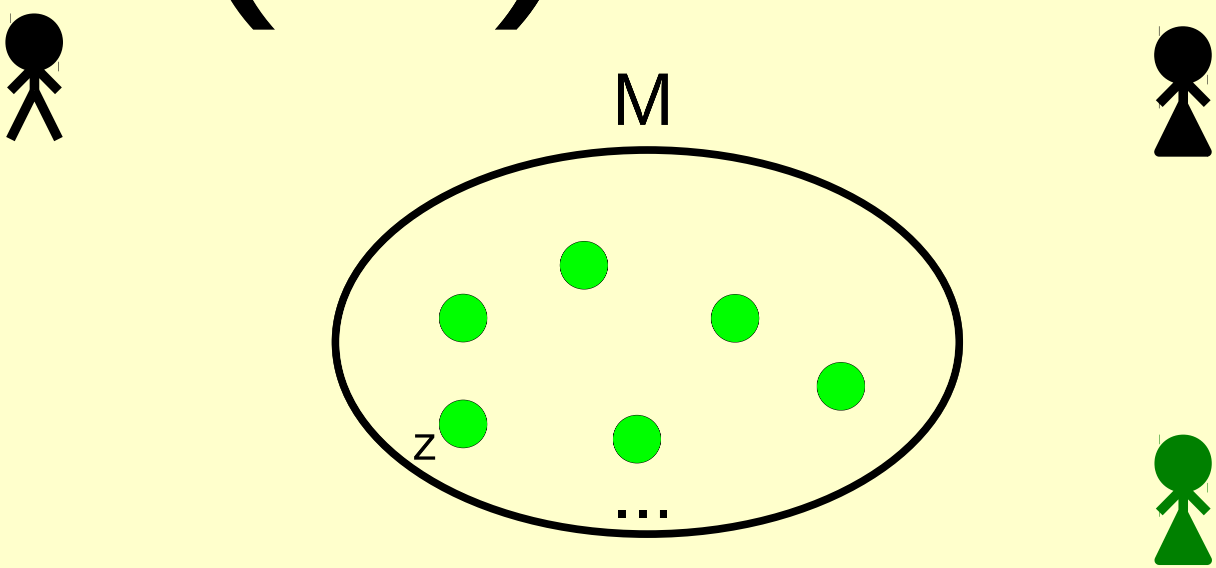


Dynamic Dependence Logic

But what does “dependence between connectives” *mean*?

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

$$\begin{pmatrix} \forall x \ \exists y \\ \forall z \ \exists w \end{pmatrix} (x=w \vee y=z)$$

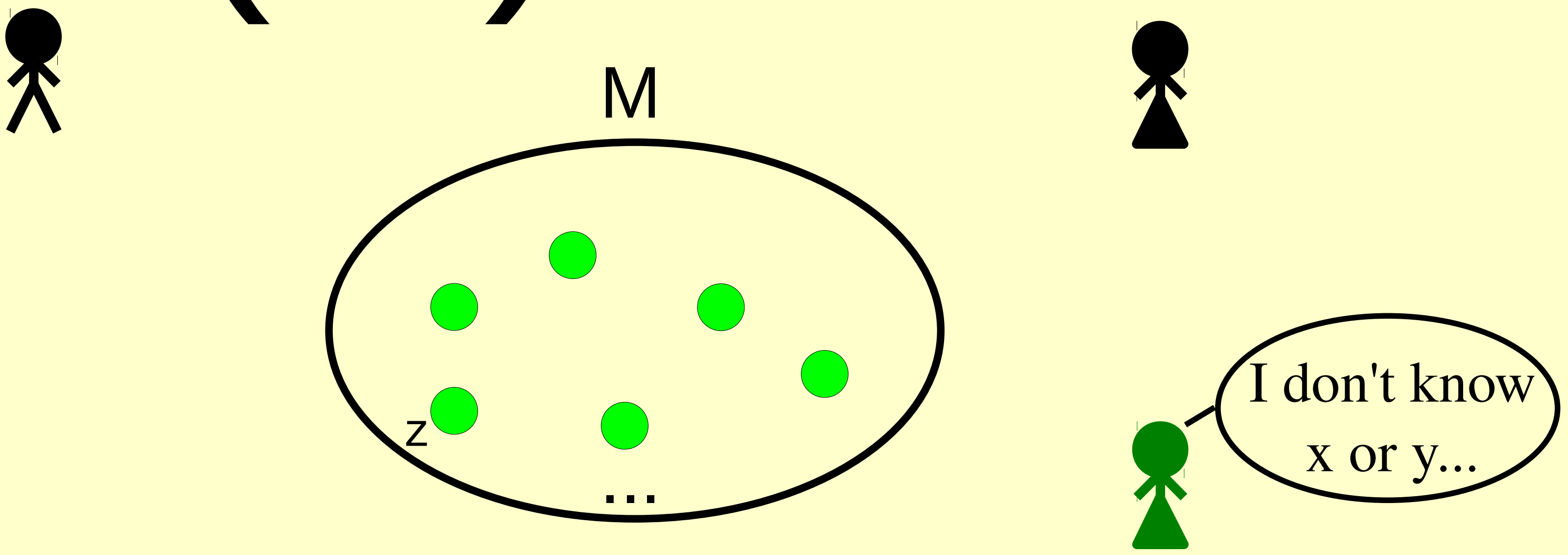


Dynamic Dependence Logic

But what does “dependence between connectives” *mean*?

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

$$\begin{pmatrix}\forall x\ \exists y \\ \forall z\ \exists w\end{pmatrix}(x=w \vee y=z)$$

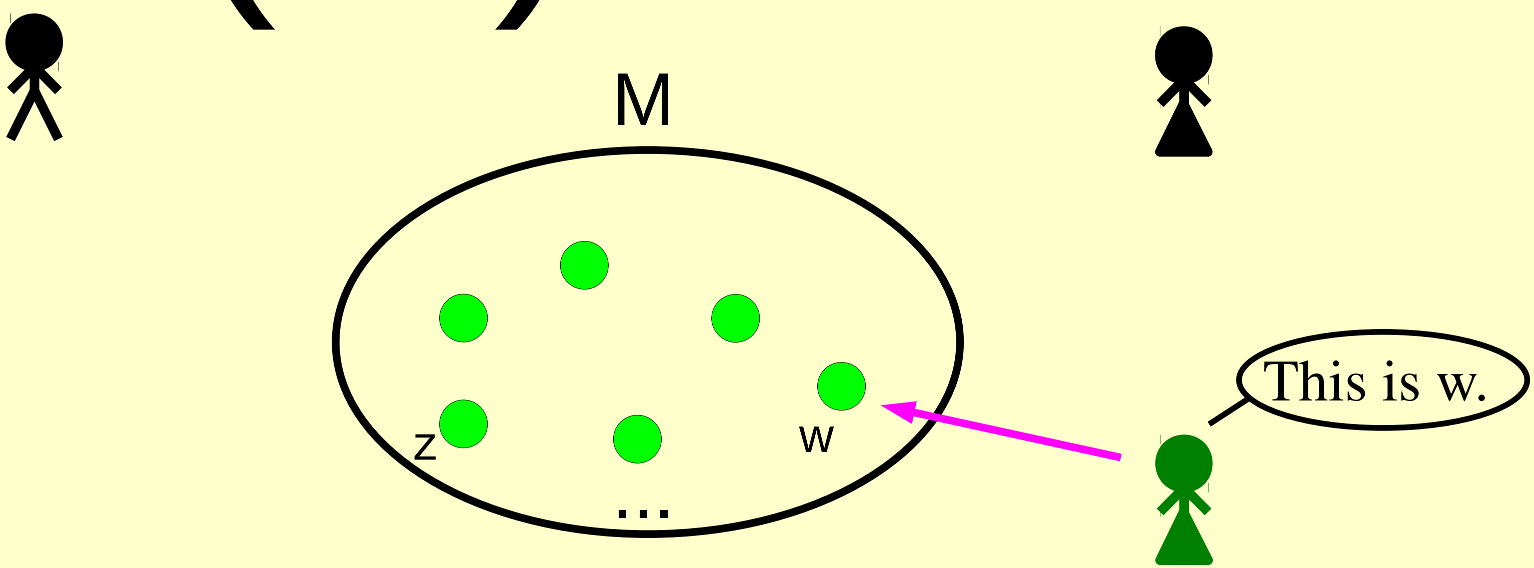


Dynamic Dependence Logic

But what does “dependence between connectives” *mean*?

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

$$\begin{pmatrix} \forall x \; \exists y \\ \forall z \; \exists w \end{pmatrix} (x=w \vee y=z)$$

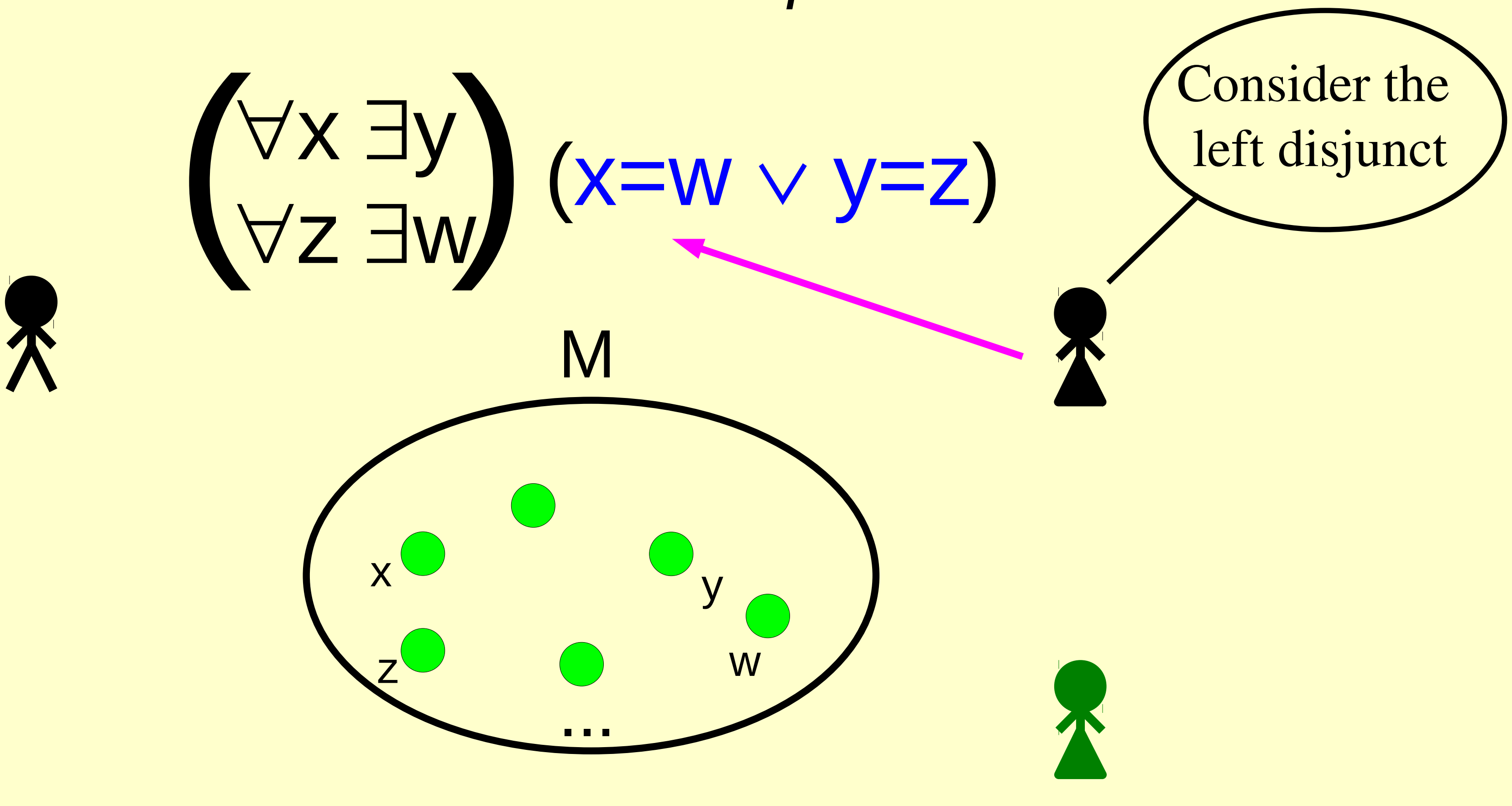


Dynamic Dependence Logic

But what does “dependence between connectives” *mean*?

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

$$\begin{pmatrix} \forall x \; \exists y \\ \forall z \; \exists w \end{pmatrix} (x=w \lor y=z)$$

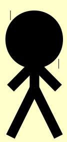


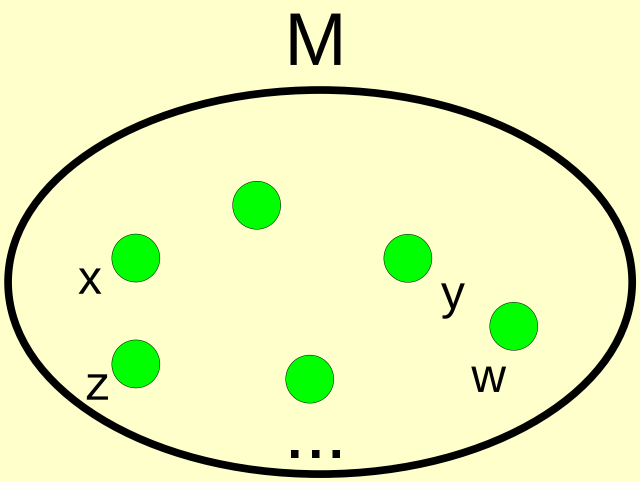
Dynamic Dependence Logic

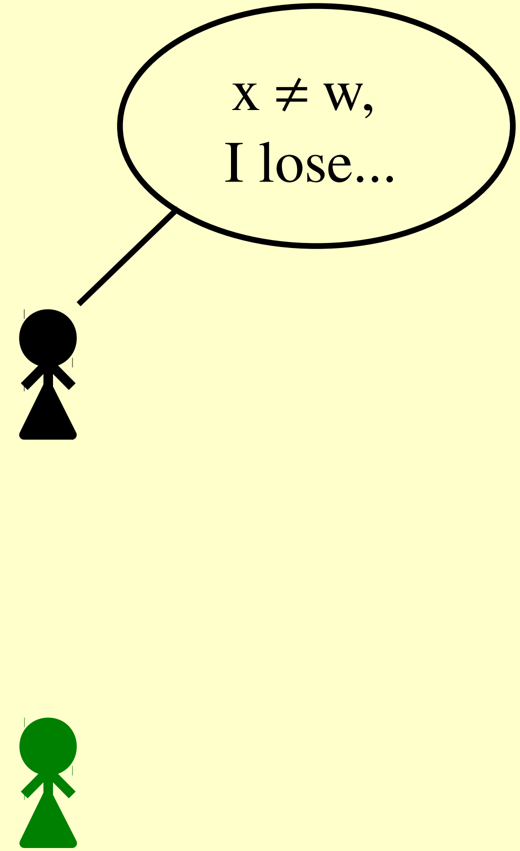
But what does “dependence between connectives” *mean*?

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

$$\begin{pmatrix} \forall x \; \exists y \\ \forall z \; \exists w \end{pmatrix} (x=w \vee y=z)$$





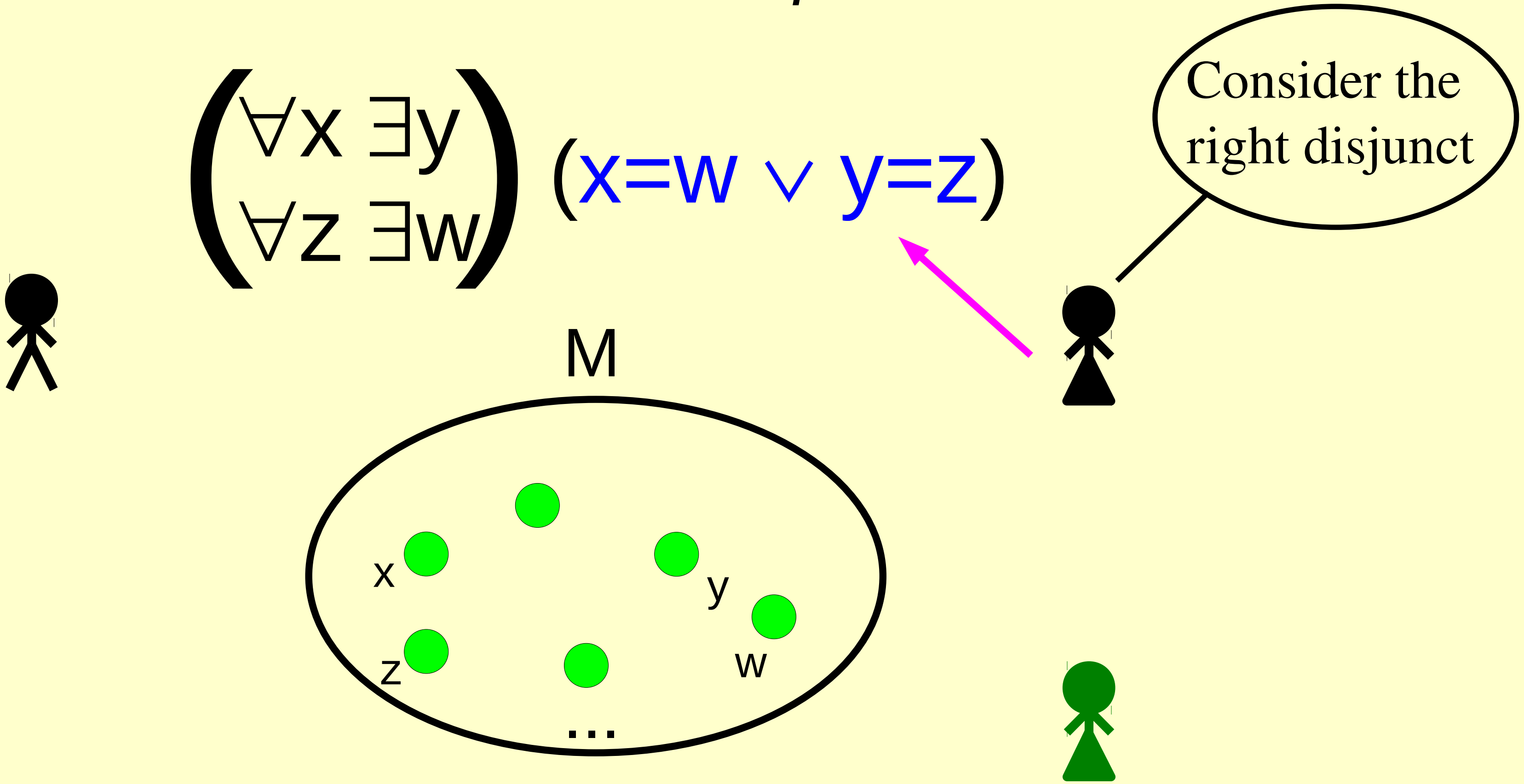


Dynamic Dependence Logic

But what does “dependence between connectives” *mean*?

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

$$\begin{pmatrix} \forall x\ \exists y \\ \forall z\ \exists w \end{pmatrix} (x=w \lor y=z)$$

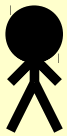


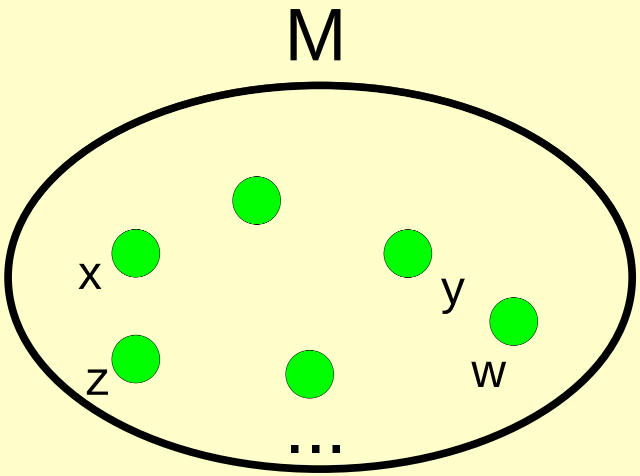
Dynamic Dependence Logic

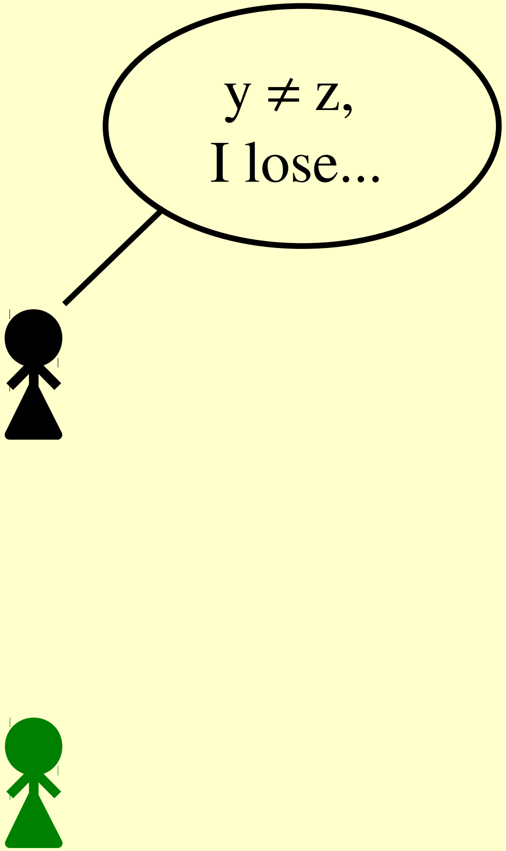
But what does “dependence between connectives” *mean*?

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

$$\begin{pmatrix}\forall x \; \exists y \\ \forall z \; \exists w\end{pmatrix} (x=w \lor y=z)$$





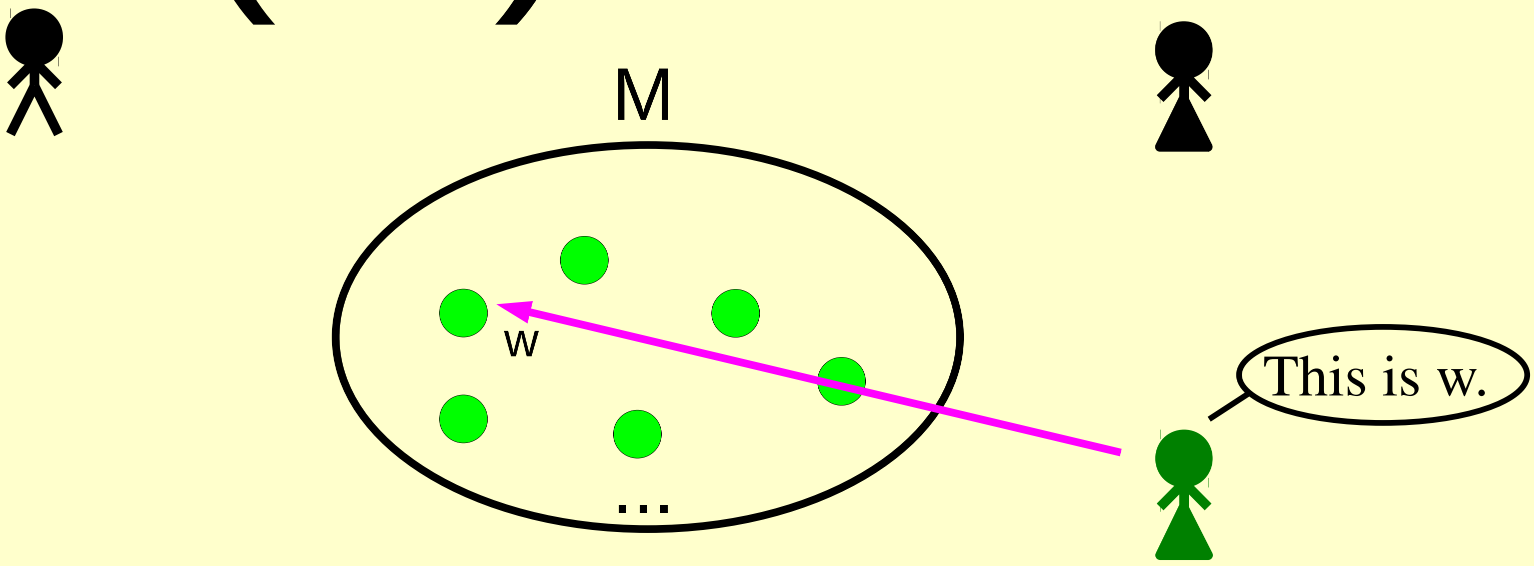


Dynamic Dependence Logic

But what does "dependence between connectives" *mean*?

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

$$\begin{pmatrix} \forall x \; \exists y \\ \forall z \; {\color{blue}\exists w} \end{pmatrix} (x=w \vee y=z)$$

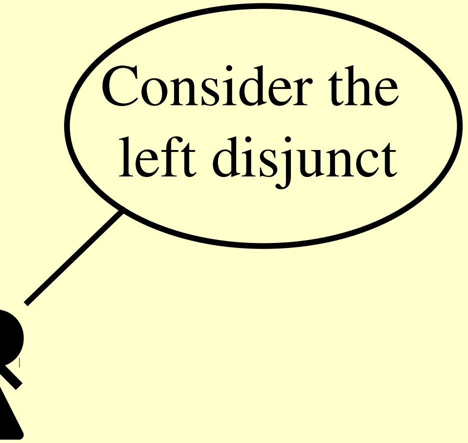


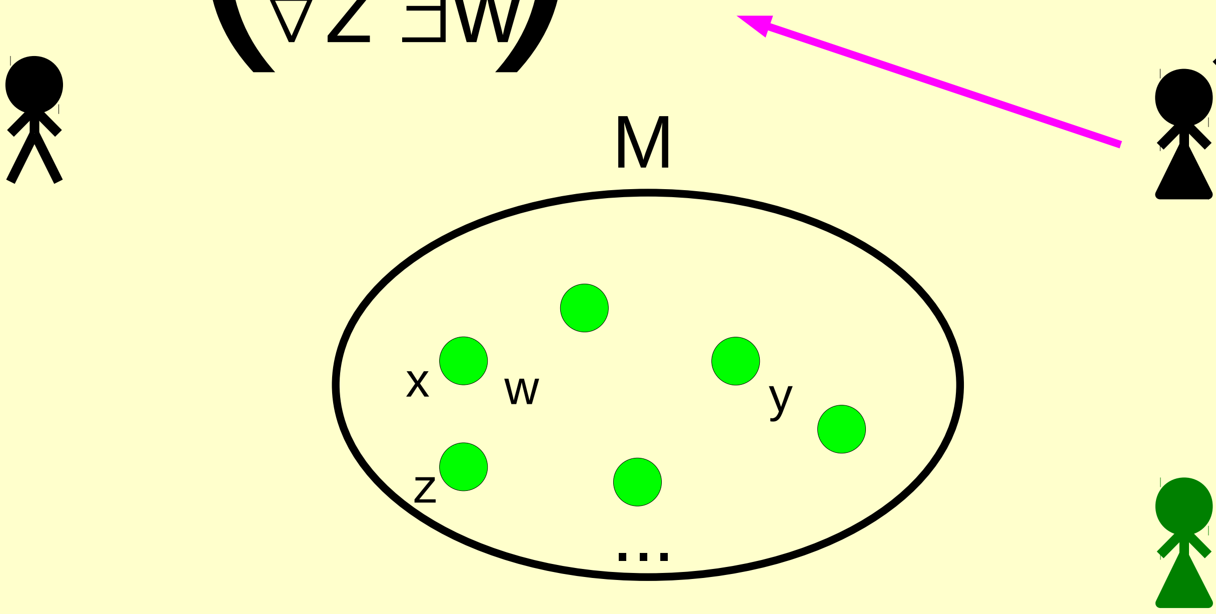
Dynamic Dependence Logic

But what does “dependence between connectives” *mean*?

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

$$\begin{pmatrix} \forall x \ \exists y \\ \forall z \ \exists w \end{pmatrix} (x=w \vee y=z)$$



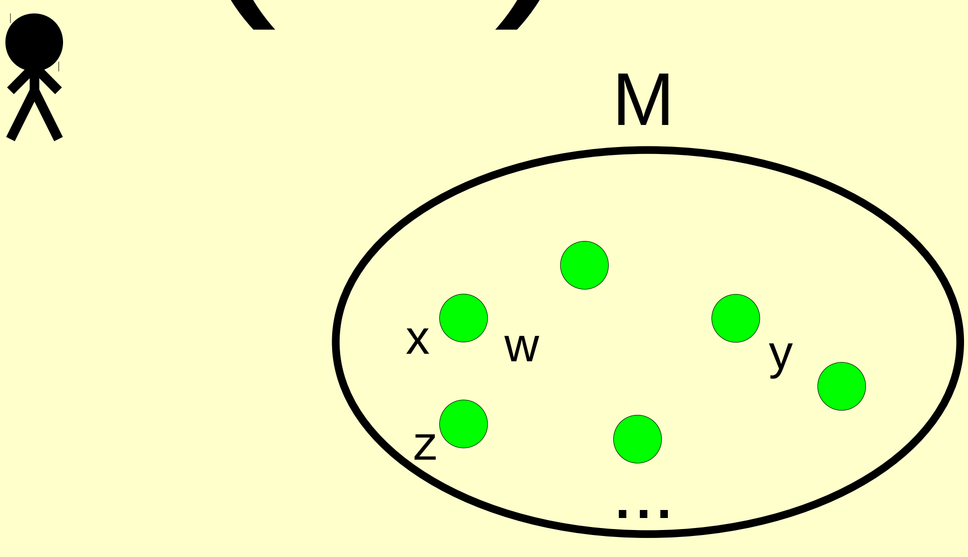


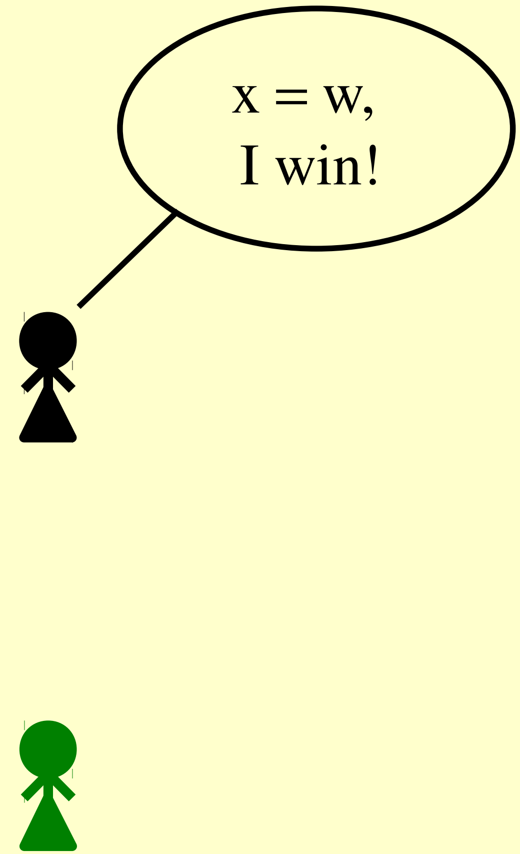
Dynamic Dependence Logic

But what does “dependence between connectives” *mean*?

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

$$\begin{pmatrix} \forall x\ \exists y \\ \forall z\ \exists w \end{pmatrix} ({\color{blue}x=w} \vee y=z)$$



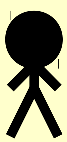


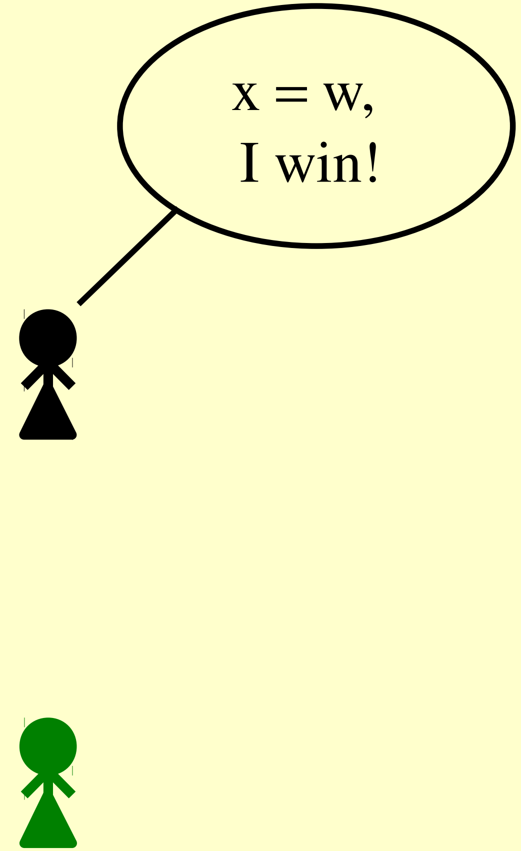
Dynamic Dependence Logic

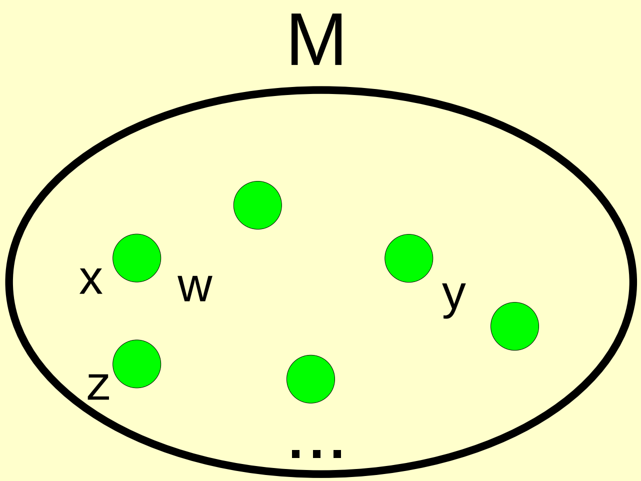
But what does “dependence between connectives” *mean*?

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

$$\begin{pmatrix}\forall x\ \exists y\\ \forall z\ \exists w\end{pmatrix}(x=w \vee y=z)$$







Neither player has a winning strategy

Dynamic Dependence Logic

But what does “dependence between connectives” *mean*?

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

$M \models \phi$ iff Eloise has a winning strategy for $G_M(\phi)$

This gives us a semantics for Branching quantifiers

Dynamic Dependence Logic

But what does “dependence between connectives” *mean*?

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

$M \models \phi$ 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)

$$\begin{pmatrix} \forall x \ \exists y \\ \forall z \ \exists w \end{pmatrix} \phi(x,y,z,w) \equiv \forall x \ \exists y \ \forall z \ (\exists w/x,y)\phi(x,y,z,w)$$

Dynamic Dependence Logic

Simpler notations for dependence and independence:

IF Logic (Hintikka and Sandu, 1989)

$$\begin{pmatrix} \forall x \ \exists y \\ \forall z \ \exists w \end{pmatrix} \phi(x,y,z,w) \equiv \forall x \ \exists y \ \forall z \ (\exists w/x,y)\phi(x,y,z,w)$$

w is *independent* from x and y

Dynamic Dependence Logic

Simpler notations for dependence and independence:

IF Logic (Hintikka and Sandu, 1989)

$$\begin{pmatrix} \forall x & \exists y \\ \forall z & \exists w \end{pmatrix} \phi(x,y,z,w) \equiv \forall x\ \exists y\ \forall z\ (\exists w/x,y)\phi(x,y,z,w)$$

Dependence Logic (Väänänen, 2007)

$$\begin{pmatrix} \forall x & \exists y \\ \forall z & \exists w \end{pmatrix} \phi(x,y,z,w) \equiv \forall x \exists y \forall z \exists w (=(z,w) \wedge \phi(x,y,z,w))$$

Dynamic Dependence Logic

Simpler notations for dependence and independence:

IF Logic (Hintikka and Sandu, 1989)

$$\begin{pmatrix}\forall x \ \exists y \\ \forall z \ \exists w\end{pmatrix} \phi(x,y,z,w) \equiv \forall x \ \exists y \ \forall z \ (\exists w/x,y)\phi(x,y,z,w)$$

Dependence Logic (Väänänen, 2007)

w *depends only* on z

$$\begin{pmatrix}\forall x \ \exists y \\ \forall z \ \exists w\end{pmatrix} \phi(x,y,z,w) \equiv \forall x \exists y \forall z \exists w (=(z,w) \wedge \phi(x,y,z,w))$$

Dynamic Dependence Logic

Hodges semantics (1997)

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

Dynamic Dependence Logic

Hodges semantics

A *team* X is a set of assignments:

Dynamic Dependence Logic

Hodges semantics

A *team* X is a set of assignments:

	x	**y**	**z**	**...**
$\mathbf{s_0}$	0	1	0	...
$\mathbf{s_1}$	1	0	1	...
$\mathbf{s_2}$	0	0	1	...
$\mathbf{s_3}$	0	0	1	...
...	...	...	...	...

Dynamic Dependence Logic

Hodges semantics

A *team* X is a set of assignments:

	x	**y**	**z**	**...**
s_0	0	1	0	...
s_1	1	0	1	...
s_2	0	0	1	...
s_3	0	0	1	...
...	...	...	...	...

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”.

Dynamic Dependence Logic

Hodges semantics

A *team* X is a set of assignments:

	x	y	z	...
s_0	0	1	0	...
s_1	1	0	1	...
s_2	0	0	1	...
s_3	0	0	1	...
...	...	...	...	...

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”.

Closure Principle: $X \models \phi,\ X' \subseteq X \Rightarrow X' \models \phi$

Dynamic Dependence Logic

Hodges semantics

A *team* X is a set of assignments:

	x	**y**	**z**	**...**
$\mathbf{s_0}$	0	1	0	...
$\mathbf{s_1}$	1	0	1	...
$\mathbf{s_2}$	0	0	1	...
$\mathbf{s_3}$	0	0	1	...
...	...	...	...	...

Operations over teams

Dynamic Dependence Logic

Hodges semantics

A *team* X is a set of assignments:

	x	y	z	...
s_0	0	1	0	...
s_1	1	0	1	...
s_2	0	0	1	...
s_3	0	0	1	...
...	...	...	...	...

Operations over teams

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

Dynamic Dependence Logic

Hodges semantics

A *team* X is a set of assignments:

	x	**y**	**z**	**...**
$\mathbf{s_0}$	0	1	0	...
$\mathbf{s_1}$	1	0	1	...
$\mathbf{s_2}$	0	0	1	...
$\mathbf{s_3}$	0	0	1	...
...	...	...	...	...

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

Operations over teams

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

Dynamic Dependence Logic

Hodges semantics

A *team* X is a set of assignments:

	x	y	z	...
s_0	0	1	0	...
s_1	1	0	1	...
s_2	0	0	1	...
s_3	0	0	1	...
...	...	...	...	...

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

X[F/x]:

Operations over teams

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

Dynamic Dependence Logic

Hodges semantics

A *team* X is a set of assignments:

	x	**y**	**z**	**...**
s'_0	**1**	1	0	...
s_1	1	0	1	...
s_2	0	0	1	...
s_3	0	0	1	...
...	...	...	...	...

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

X[F/x]:

Operations over teams

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

Dynamic Dependence Logic

Hodges semantics

A *team* X is a set of assignments:

	x	**y**	**z**	**...**
s'_0	1	1	0	...
s'_1	**0**	0	1	...
s_2	0	0	1	...
s_3	0	0	1	...
...	...	...	...	...

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

X[F/x]:

Operations over teams

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

Dynamic Dependence Logic

Hodges semantics

A *team* X is a set of assignments:

	x	**y**	**z**	**...**
s'_0	1	1	0	...
s'_1	0	0	1	...
s'_2	**0**	0	1	...
s_3	0	0	1	...
...	...	...	...	...

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

X[F/x]:

Operations over teams

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

Dynamic Dependence Logic

Hodges semantics

A *team* X is a set of assignments:

	x	**y**	**z**	**...**
s'$_0$	1	1	0	...
s'$_1$	0	0	1	...
s'$_2$	0	0	1	...
s'$_3$	**1**	0	1	...
...	...	...	...	...

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

X[F/x]:

Operations over teams

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

Dynamic Dependence Logic

Hodges semantics

A *team* X is a set of assignments:

	x	**y**	**z**	**...**
s'_0	1	1	0	...
s'_1	0	0	1	...
s'_3	1	0	1	...
...	...	...	...	...

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

X[F/x]:

Operations over teams

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

Dynamic Dependence Logic

Hodges semantics

A *team* X is a set of assignments:

	x	y	z	...
s'_0	1	1	0	...
s'_1	0	0	1	...
s'_3	1	0	1	...
...	...	...	...	...

Operations over teams

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

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

Dynamic Dependence Logic

Hodges semantics

	x	y	z	...
s'_{00}	0	1	0	...
s'_{01}	1	1	0	...
s'_{10}	0	0	1	...
s'_{11}	1	0	1	...
s'_{30}	0	0	1	...
s'_{30}	1	0	1	...
...	...	...	...	...

Operations over teams

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

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

Dynamic Dependence Logic

Hodges semantics

	x	y	z	...
s'_{00}	0	1	0	...
s'_{01}	1	1	0	...
s'_{10}	0	0	1	...
s'_{11}	1	0	1	...
...	...	...	...	...

Operations over teams

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

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

Dynamic Dependence Logic

Hodges semantics

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

$X \models^{+} Rt_1 \ldots t_n \Leftrightarrow \forall s \in X,\ s \models_{FO} Rt_1 \ldots t_n$; $X \models^{-} Rt_1 \ldots t_n \Leftrightarrow \forall s \in X,\ s \models_{FO} \neg Rt_1 \ldots t_n$;

Dynamic Dependence Logic

Hodges semantics

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

$X \models^{+} Rt_1...t_n \Leftrightarrow \forall s \in X, s \models_{FO} Rt_1...t_n$;

$X \models^{-} Rt_1...t_n \Leftrightarrow \forall s \in X, s \models_{FO} \neg Rt_1...t_n$;

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

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

Dynamic Dependence Logic

Hodges semantics

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

$X \models^{+} Rt_1...t_n \Leftrightarrow \forall s \in X, s \models_{FO} Rt_1...t_n$;

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

$X \models^{+} \neg\phi \Leftrightarrow X \models^{-} \phi$;

$X \models^{-} Rt_1...t_n \Leftrightarrow \forall s \in X, s \models_{FO} \neg Rt_1...t_n$;

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

$X \models^{-} \neg \phi \Leftrightarrow X \models^{+} \phi$;

Dynamic Dependence Logic

Hodges semantics

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

$X \models^{+} Rt_1 \ldots t_n \Leftrightarrow \forall s \in X,\ s \models_{FO} Rt_1 \ldots t_n$;

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

$X \models^{+} \neg\phi \Leftrightarrow X \models^{-} \phi$;

$X \models^{+} \phi \vee \psi \Leftrightarrow X = Y \cup Z,\ Y \models^{+} \phi,\ Z \models^{+} \psi$;

$X \models^{-} Rt_1 \ldots t_n \Leftrightarrow \forall s \in X,\ s \models_{FO} \neg Rt_1 \ldots t_n$;

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

$X \models^{-} \neg\, \phi \Leftrightarrow X \models^{+} \phi$;

$X \models^{-} \phi \vee y \Leftrightarrow X \models^{-} \phi$ and $X \models^{-} \psi$;

Dynamic Dependence Logic

Hodges semantics

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

$X \models^{+} Rt_1 \ldots t_n \Leftrightarrow \forall s \in X, s \models_{FO} Rt_1 \ldots t_n$;

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

$X \models^{+} \neg\phi \Leftrightarrow X \models^{-} \phi$;

$X \models^{+} \phi \vee \psi \Leftrightarrow X = Y \cup Z, Y \models^{+} \phi, Z \models^{+} \psi$;

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

$X \models^{-} Rt_1 \ldots t_n \Leftrightarrow \forall s \in X, s \models_{FO} \neg Rt_1 \ldots t_n$;

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

$X \models^{-} \neg\, \phi \Leftrightarrow X \models^{+} \phi$;

$X \models^{-} \phi \vee y \Leftrightarrow X \models^{-} \phi$ and $X \models^{-} \psi$;

$X \models^{-} \exists x \phi \Leftrightarrow X[M/x] \models^{-} \phi$;

Dynamic Dependence Logic

Hodges semantics

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

$X \models^+ Rt_1 \ldots t_n \Leftrightarrow \forall s \in X,\ s \models_{FO} Rt_1 \ldots t_n$;

$X \models^- Rt_1 \ldots t_n \Leftrightarrow \forall s \in X,\ s \models_{FO} \neg Rt_1 \ldots t_n$;

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

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

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

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

$X \models^+ \phi \vee \psi \Leftrightarrow X = Y \cup Z,\ Y \models^+ \phi,\ Z \models^+ \psi$;

$X \models^- \phi \vee y \Leftrightarrow X \models^- \phi$ and $X \models^- \psi$;

$X \models^+ \exists x\ \phi \Leftrightarrow \exists F$ s.t. $X[F/x] \models^+ \phi$.

$X \models^- \exists x \phi \Leftrightarrow X[M/x] \models^- \phi$;

$X \models^+ =(t_1 \ldots t_n) \Leftrightarrow \forall s, s' \in X$,

$t_i(s) = t_i(s')$ for $i=1 \ldots n-1 \Rightarrow t_n(s)=t_n(s')$;

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

A sentence ϕ 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

$X \models^{+} Rt_1 \ldots t_n \Leftrightarrow \forall s \in X,\ s \models_{FO} Rt_1 \ldots t_n;$

$X \models^{-} Rt_1 \ldots t_n \Leftrightarrow \forall s \in X,\ s \models_{FO} \neg Rt_1 \ldots t_n;$

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

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

$X \models^{+} \neg\phi \Leftrightarrow X \models^{-} \phi;$

$X \models^{-} \neg\, \phi \Leftrightarrow X \models^{+} \phi;$

$X \models^{+} \phi \vee \psi \Leftrightarrow X = Y \cup Z,\ Y \models^{+} \phi,\ Z \models^{+} \psi;$

$X \models^{-} \phi \vee y \Leftrightarrow X \models^{-} \phi$ and $X \models^{-} \psi;$

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

$X \models^{-} \exists x \phi \Leftrightarrow X[M/x] \models^{-} \phi;$

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

$X \models^{-} =(t_1 \ldots t_n) \Leftrightarrow X = \emptyset.$

A sentence ϕ of Dependence Logic is true in Game Theoretic Semantics iff $X \models^{+} \phi$ 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.

$\exists x_1(\mathrm{MAN}(x_1) \wedge \mathrm{WALK\_IN\_PARK}(x_1))$

Dynamic Dependence Logic

Introduced to deal with anaphora binding:

A man_1 walks in the park. He_1 whistles.

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

Dynamic Dependence Logic

Introduced to deal with anaphora binding:

A man_1 walks in the park. He_1 whistles.

$\exists x_1(MAN(x_1) \wedge WALK\_IN\_PARK(x_1)) \wedge WHISTLES(x_1)$

$\exists x_1(MAN(x_1) \wedge WALK\_IN\_PARK(x_1) \wedge WHISTLES(x_1))$

Dynamic Dependence Logic

Introduced to deal with anaphora binding:

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

$\exists x_1(\mathrm{MAN}(x_1) \wedge \mathrm{WALK\_IN\_PARK}(x_1)) \wedge \mathrm{WHISTLES}(x_1)$

$\exists x_1(\mathrm{MAN}(x_1) \wedge \mathrm{WALK\_IN\_PARK}(x_1) \wedge \mathrm{WHISTLES}(x_1))$

But:

Dynamic Dependence Logic

Introduced to deal with anaphora binding:

A man_1 walks in the park. He_1 whistles.

$\exists x_1(MAN(x_1) \wedge WALK\_IN\_PARK(x_1)) \wedge WHISTLES(x_1)$

$\exists x_1(MAN(x_1) \wedge WALK\_IN\_PARK(x_1) \wedge WHISTLES(x_1))$

But:

Not every man_1 does not walk in the park.

$\neg\forall x_1(MAN(x_1) \rightarrow \neg WALK\_IN\_PARK(x_1))$

Dynamic Dependence Logic

Introduced to deal with anaphora binding:

A man_1 walks in the park. He_1 whistles.

$\exists x_1(MAN(x_1) \wedge WALK\_IN\_PARK(x_1)) \wedge WHISTLES(x_1)$

$\exists x_1(MAN(x_1) \wedge WALK\_IN\_PARK(x_1) \wedge WHISTLES(x_1))$

But:

Not every man_1 does not walk in the park. $*He_1$ whistles.

$\neg\forall x_1(MAN(x_1) \rightarrow \neg WALK\_IN\_PARK(x_1)) \wedge WHISTLES(x_1)$

Dynamic Dependence Logic

Introduced to deal with anaphora binding:

A man_1 walks in the park. He_1 whistles.

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

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

But:

Not every man_1 does not walk in the park. $^*\text{He}_1$ whistles.

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

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

Dynamic Dependence Logic

Introduced to deal with anaphora binding:

A man_1 walks in the park. He_1 whistles.

$\exists x_1(MAN(x_1) \wedge WALK\_IN\_PARK(x_1)) \wedge WHISTLES(x_1)$

$\exists x_1(MAN(x_1) \wedge WALK\_IN\_PARK(x_1) \wedge WHISTLES(x_1))$

But:

UNGRAMMATICAL

Not every man_1 does not walk in the park. *He_1 whistles.

$\neg\forall x_1(MAN(x_1) \rightarrow \neg WALK\_IN\_PARK(x_1)) \wedge WHISTLES(x_1)$

$\neg\forall x_1((MAN(x_1) \rightarrow \neg WALK\_IN\_PARK(x_1)) \wedge WHISTLES(x_1))$

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.

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.

$(s,s') \models \phi$: starting from the information state s, ϕ 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') |= ϕ : starting from the information state s, ϕ 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)

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.

$(s,s') \models \phi$: starting from the information state s, ϕ 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)

$(s,s') \models Rt_1..t_n \Leftrightarrow s = s'$ and $s \models_{FO} Rt_1..t_n$;

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.

$(s,s') \models \phi$: starting from the information state s, ϕ 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)

$(s,s') \models Rt_1..t_n \Leftrightarrow s = s'$ and $s \models_{FO} Rt_1..t_n$;

$(s,s') \models t=t' \Leftrightarrow s=s'$ and $s \models_{FO} t=t'$;

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.

$(s,s') \models \phi$: starting from the information state s, ϕ 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)

$(s,s') \models Rt_1..t_n \Leftrightarrow s = s'$ and $s \models_{FO} Rt_1..t_n$;

$(s,s') \models t=t' \Leftrightarrow s=s'$ and $s \models_{FO} t=t'$;

$(s,s') \models \neg\phi \Leftrightarrow s=s'$ and for no s'', $(s,s'') \models \phi$;

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.

$(s,s') \models \phi$: starting from the information state s, ϕ 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)

$(s,s') \models Rt_1..t_n \Leftrightarrow s = s'$ and $s \models_{FO} Rt_1..t_n$;

$(s,s') \models t=t' \Leftrightarrow s=s'$ and $s \models_{FO} t=t'$;

$(s,s') \models \neg\phi \Leftrightarrow s=s'$ and for no s'', $(s,s'') \models \phi$;

$(s,s') \models \phi \wedge \psi \Leftrightarrow \exists s''$, $(s,s'') \models \phi$ 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') |= ϕ : starting from the information state s, ϕ 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)

$(s,s') \models Rt_1..t_n \Leftrightarrow s = s'$ and $s \models_{FO} Rt_1..t_n$;

$(s,s') \models t=t' \Leftrightarrow s=s'$ and $s \models_{FO} t=t'$;

$(s,s') \models \neg\phi \Leftrightarrow s=s'$ and for no s'', $(s,s'') \models \phi$;

$(s,s') \models \phi \wedge \psi \Leftrightarrow \exists s'', (s,s'') \models \phi$ and $(s'', s') \models \psi$;

$(s,s') \models \exists x\, \phi \Leftrightarrow \exists m \in dom(M)$ 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.

$(s,s') \models \phi$: starting from the information state s, ϕ 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)

$(s,s') \models Rt_1..t_n \Leftrightarrow s = s'$ and $s \models_{FO} Rt_1..t_n$;

$(s,s') \models t=t' \Leftrightarrow s=s'$ and $s \models_{FO} t=t'$;

$(s,s') \models \neg\phi \Leftrightarrow s=s'$ and for no s'', $(s,s'') \models \phi$;

$(s,s') \models \phi \wedge \psi \Leftrightarrow \exists s''$, $(s,s'') \models \phi$ and $(s'', s') \models \psi$;

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

$\exists x$ can be taken as an atomic sentence:

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.

$(s,s') \models \phi$: starting from the information state s, ϕ 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)

$(s,s') \models Rt_1..t_n \Leftrightarrow s = s'$ and $s \models_{FO} Rt_1..t_n$;

$(s,s') \models t=t' \Leftrightarrow s=s'$ and $s \models_{FO} t=t'$;

$(s,s') \models \neg\phi \Leftrightarrow s=s'$ and for no s'', $(s,s'') \models \phi$;

$(s,s') \models \phi \wedge \psi \Leftrightarrow \exists s''$, $(s,s'') \models \phi$ and $(s'', s') \models \psi$;

$(s,s') \models \exists x\, \phi \Leftrightarrow \exists m \in dom(M)$ s.t. $(s[m/x], s') \models \phi$.

$\exists$x can be taken as an atomic sentence:

$(s,s') \models \exists x \Leftrightarrow \exists m$, $s' = s[m/x]$

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.

$(s,s') \models \phi$: starting from the information state s, ϕ 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)

$(s,s') \models Rt_1..t_n \Leftrightarrow s = s'$ and $s \models_{FO} Rt_1..t_n$;

$(s,s') \models t=t' \Leftrightarrow s=s'$ and $s \models_{FO} t=t'$;

$(s,s') \models \neg\phi \Leftrightarrow s=s'$ and for no s'', $(s,s'') \models \phi$;

$(s,s') \models \phi \wedge \psi \Leftrightarrow \exists s''$, $(s,s'') \models \phi$ and $(s'', s') \models \psi$;

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

$\exists x$ can be taken as an atomic sentence:

$(s,s') \models \exists x \Leftrightarrow \exists m,\ s' = s[m/x]$

$$\exists x\, \phi := \exists x \wedge \phi$$

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

Dynamic Dependence Logic

Can we make a dynamic version of dependence logic?

Formulas = transitions between teams

$\phi ::= Rt_1...t_n \mid t=t' \mid =(t_1...t_n) \mid \exists x \mid \neg(\phi) \mid \phi \vee \psi \mid \phi \times \psi$

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 ϕ the current state will be in Y"*.

Dynamic Dependence Logic

Can we make a dynamic version of dependence logic?

Formulas = transitions between teams

$\phi ::= Rt_1...t_n \mid t=t' \mid =(t_1...t_n) \mid \exists x \mid \neg(\phi) \mid \phi \vee \psi \mid \phi \,.\, \psi$

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

$(X, Y) \models^{+} \phi$ means *"If the current state is in X, after executing* ϕ *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"*.

Dynamic Dependence Logic

Can we make a dynamic version of dependence logic?

Formulas = transitions between teams

$\phi ::= Rt_1...t_n \mid t=t' \mid =(t_1...t_n) \mid \exists x \mid \neg(\phi) \mid \phi \vee \psi \mid \phi \,.\, \psi$

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

$(X, Y) \models^{+} Rt_1..t_n \Leftrightarrow X \subseteq Y$ and $\forall s \in X$, $s \models_{FO} Rt_1..t_n$;

$(X, Y) \models^{-} Rt_1..t_n \Leftrightarrow X \subseteq Y$ and $\forall s \in X$, $s \models_{FO} \neg Rt_1..t_n$;

Dynamic Dependence Logic

Can we make a dynamic version of dependence logic?

Formulas = transitions between teams

$$\phi ::= Rt_1...t_n \mid t=t' \mid =(t_1...t_n) \mid \exists x \mid \neg(\phi) \mid \phi \vee \psi \mid \phi \,.\, \psi$$

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

$(X, Y) \models^+ Rt_1..t_n \Leftrightarrow X \subseteq Y$ and $\forall s \in X$, $s \models_{FO} Rt_1..t_n$;

$(X, Y) \models^- Rt_1..t_n \Leftrightarrow X \subseteq Y$ and $\forall s \in X$, $s \models_{FO} \neg Rt_1..t_n$;

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

$(X, Y) \models^- t=t' \Leftrightarrow X \subseteq Y$ and $\forall s \in X$, $s \models_{FO} t \neq t'$;

Dynamic Dependence Logic

Can we make a dynamic version of dependence logic?

Formulas = transitions between teams

$\phi ::= Rt_1...t_n \mid t=t' \mid =(t_1...t_n) \mid \exists x \mid \neg(\phi) \mid \phi \vee \psi \mid \phi \,.\, \psi$

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

$(X, Y) \models^+ =(t_1...t_n) \Leftrightarrow X \subseteq Y$ and $\forall s, s' \in X,$

$t_1(s) = t_1(s') \ldots t_{n-1}(s) = t_{n-1}(s') \Rightarrow t_n(s) = t_n(s')$;

$(X, Y) \models^- =(t_1...t_n) \Leftrightarrow X = \emptyset$;

Dynamic Dependence Logic

Can we make a dynamic version of dependence logic?

Formulas = transitions between teams

$\phi ::= Rt_1...t_n \mid t=t' \mid =(t_1...t_n) \mid \exists x \mid \neg(\phi) \mid \phi \vee \psi \mid \phi \,.\, \psi$

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

$(X, Y) \models^{+} =(t_1...t_n) \Leftrightarrow X \subseteq Y$ and $\forall s, s' \in X$,

$t_1(s) = t_1(s') \dots t_{n-1}(s) = t_{n-1}(s') \Rightarrow t_n(s) = t_n(s')$;

$(X, Y) \models^{-} =(t_1...t_n) \Leftrightarrow X = \emptyset$;

$(X, Y) \models^{+} \exists x \Leftrightarrow \exists F$ s.t. $X[F/x] \subseteq Y$;

$(X, Y) \models^{-} \exists x \Leftrightarrow X[M/x] \subseteq Y$;

Dynamic Dependence Logic

Can we make a dynamic version of dependence logic?

Formulas = transitions between teams

ϕ ::= $Rt_1 \ldots t_n$ | $t=t'$ | $=(t_1 \ldots t_n)$ | $\exists x$ | $\neg(\phi)$ | $\phi \vee \psi$ | $\phi \,.\, \psi$

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

$(X, Y) \models^{+} \neg(\phi) \Leftrightarrow (X,Y) \models^{-} \phi$;

$(X, Y) \models^{-} \neg(\phi) \Leftrightarrow (X,Y) \models^{+} \phi$;

Dynamic Dependence Logic

Can we make a dynamic version of dependence logic?

Formulas = transitions between teams

$\phi ::= Rt_1 \ldots t_n \mid t{=}t' \mid {=}(t_1 \ldots t_n) \mid \exists x \mid \neg(\phi) \mid \phi \vee \psi \mid \phi \, . \, \psi$

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

$(X, Y) \models^{+} \neg(\phi) \Leftrightarrow (X,Y) \models^{-} \phi$;
$(X, Y) \models^{-} \neg(\phi) \Leftrightarrow (X,Y) \models^{+} \phi$;

$(X, Y) \models^{+} \phi \vee \psi \Leftrightarrow X = X_1 \cup X_2$, $(X_1, Y) \models^{+} \phi$ and $(X_2, Y) \models^{+} \psi$;
$(X, Y) \models^{-} \phi \vee \psi \Leftrightarrow (X, Y) \models^{-} \phi$ and $(X, Y) \models^{-} \psi$.

Dynamic Dependence Logic

Can we make a dynamic version of dependence logic?

Formulas = transitions between teams

$\phi ::= Rt_1...t_n \mid t=t' \mid =(t_1...t_n) \mid \exists x \mid \neg(\phi) \mid \phi \vee \psi \mid \phi \,.\, \psi$

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

$(X, Y) \models^{+} \phi \,.\, \psi \Leftrightarrow \exists Z$ s.t. $(X, Z) \models^{+} \phi$ and $(Z, Y) \models^{+} \psi$;

$(X, Y) \models^{-} \phi \,.\, \psi \Leftrightarrow \exists Z$ s.t $(X, Z) \models^{-} \phi$ and $(Z, Y) \models^{-} \psi$.

Dynamic Dependence Logic

Can we make a dynamic version of dependence logic?

Formulas = transitions between teams

$\phi ::= Rt_1 \ldots t_n \mid t = t' \mid =(t_1 \ldots t_n) \mid \exists x \mid \neg(\phi) \mid \phi \vee \psi \mid \phi \,.\, \psi$

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

$(X, Y) \models^{+} \phi \,.\, \psi \Leftrightarrow \exists Z$ s.t. $(X, Z) \models^{+} \phi$ and $(Z, Y) \models^{+} \psi$;
$(X, Y) \models^{-} \phi \,.\, \psi \Leftrightarrow \exists Z$ s.t $(X, Z) \models^{-} \phi$ and $(Z, Y) \models^{-} \psi$.

$X \models^{+} \phi$ iff $\exists Y$ s.t. $(X, Y) \models^{+} \phi$
If ψ sentence, $\models^{+} \psi$ iff $X \models^{+} \psi$ 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

Dynamic Dependence Logic is a conservative extension of Dependence Logic:

Dynamic Dependence Logic

Dynamic Dependence Logic is a conservative extension of Dependence Logic:

$$X \models^{+}_{DL} \phi \Leftrightarrow \exists Y, (X,Y) \models^{+}_{DDL} \phi$$

Dynamic Dependence Logic

Dynamic Dependence Logic is a conservative extension of Dependence Logic:

$$X \models^{+}_{DL} \phi \Leftrightarrow \exists Y, (X,Y) \models^{+}_{DDL} \phi$$

Dynamic Dependence Logic is paraconsistent:

Dynamic Dependence Logic

Dynamic Dependence Logic is a conservative extension of Dependence Logic:

$$X \models^{+}_{DL} \phi \Leftrightarrow \exists Y, (X,Y) \models^{+}_{DDL} \phi$$

Dynamic Dependence Logic is paraconsistent:

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

Dynamic Dependence Logic

Dynamic Dependence Logic is a conservative extension of Dependence Logic:

$$X \models^{+}_{DL} \phi \Leftrightarrow \exists Y, (X,Y) \models^{+}_{DDL} \phi$$

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:

$$X \models^{+}_{DL} \phi \Leftrightarrow \exists Y, (X,Y) \models^{+}_{DDL} \phi$$

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, $X \models^{+}_{DDL} \phi \Leftrightarrow X \models^{+}_{DL} \phi'$

Dynamic Dependence Logic

Closure Property:

Dynamic Dependence Logic

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) |= ϕ, X' $\subseteq$ X, Y $\subseteq$ Y' then (X', Y') |= ϕ

	x	**y**	**z**	**...**
s_0	0	1	0	...
s_1	1	1	1	,,,
s_2	1	0	0	...

Dynamic Dependence Logic

Closure Property:

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

	x	**y**	**z**	**...**
s_0	0	1	0	...
s_1	1	1	1	,,,
s_2	1	0	0	...

$\exists x$ →

Dynamic Dependence Logic

Closure Property:

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

	x	**y**	**z**	**...**
s_0	0	1	0	...
s_1	1	1	1	,,,
s_2	1	0	0	...

$\exists x$ →

	x	**y**	**z**	**...**
s_0	$F(s_0)$	1	0	...
s_1	$F(s_1)$	1	1	,,,
s_2	$F(s_2)$	0	0	...

Dynamic Dependence Logic

Closure Property:

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

	x	**y**	**z**	**...**
s_0	0	1	0	...
s_1	1	1	1	,,,
s_2	1	0	0	...

$\exists x$

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

	x	**y**	**z**	**...**
s_0	0	1	0	...
s_1	1	1	1	,,,
s_2	1	0	0	...

Dynamic Dependence Logic

Closure Property:

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

	x	y	z	...
s_0	0	1	0	...
s_1	1	1	1	,,,
s_2	1	0	0	...

$\exists x$

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

	x	y	z	...
s_0	0	1	0	...
s_1	1	1	1	,,,
s_2	1	0	0	...
s_3	0	0	0	,,,

Dynamic Dependence Logic

Closure Property:

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

	x	y	z	...
s_0	0	1	0	...
s_2	1	0	0	...

$\exists x$ →

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

	x	y	z	...
s_0	0	1	0	...
s_1	1	1	1	,,,
s_2	1	0	0	...
s_3	0	0	0	,,,

Dynamic Dependence Logic

Closure Property:

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

	x	**y**	**z**	**...**
s_0	0	1	0	...
s_2	1	0	0	...

$\exists x$ →

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

	x	**y**	**z**	**...**
s_0	0	1	0	...
s_1	1	1	1	,,,
s_2	1	0	0	...
s_3	0	0	0	,,,

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

$\phi ::= Rt_1...t_n \mid t=t' \mid =(t_1...t_n) \mid \exists x \mid \neg \mid \phi \vee \psi \mid \phi \,.\, \psi$

Dynamic Dependence Logic

$\phi ::= Rt_1...t_n \mid t=t' \mid =(t_1...t_n) \mid \exists x \mid \neg \mid \phi \vee \psi \mid \phi \,.\, \psi$

$\neg(\phi) := \neg \,.\, \phi \,.\, \neg$

Dynamic Dependence Logic

$\phi ::= Rt_1...t_n \mid t=t' \mid =(t_1...t_n) \mid \exists x \mid \neg \mid \phi \vee \psi \mid \phi . \psi$

$\neg(\phi) := \neg . \phi . \neg$

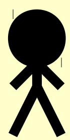
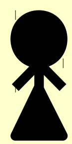
For every ϕ, X M, define the game $G^M_X(\phi)$:

Dynamic Dependence Logic

$\phi ::= Rt_1...t_n \mid t=t' \mid =(t_1...t_n) \mid \exists x \mid \neg \mid \phi \vee \psi \mid \phi \,.\, \psi$

$\neg(\phi) := \neg \,.\, \phi \,.\, \neg$

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

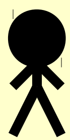
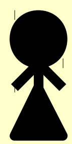
Two players,  and  ;

Dynamic Dependence Logic

$\phi ::= Rt_1...t_n \mid t=t' \mid =(t_1...t_n) \mid \exists x \mid \neg \mid \phi \vee \psi \mid \phi \,.\, \psi$

$\neg(\phi) := \neg \,.\, \phi \,.\, \neg$

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

Two players,  and  ;

Positions of the form $(\phi_0, \phi_1, \ldots \phi_n, s, \alpha)$,

$\phi_0 \ldots \phi_n \in$ DDL;

s assignment;

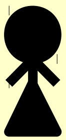
$\alpha \in \{I, II\}$;

Dynamic Dependence Logic

$\phi ::= Rt_1 \ldots t_n \mid t{=}t' \mid {=}(t_1 \ldots t_n) \mid \exists x \mid \neg \mid \phi \vee \psi \mid \phi \,.\, \psi$

$\neg(\phi) := \neg \,.\, \phi \,.\, \neg$

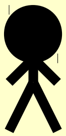
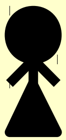
For every ϕ, X M, define the game $G^M_X(\phi)$:

Two players,  and  ;

Positions of the form $(\phi_0, \phi_1, \ldots \phi_n, s, \alpha)$,

$\phi_0 \ldots \phi_n \in$ DDL;

s assignment;

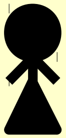
$\alpha \in \{I, II\}$; *($\alpha = I$:*  *moves, $\alpha = II$:*  *moves)*

Dynamic Dependence Logic

$\phi ::= Rt_1 \ldots t_n \mid t=t' \mid =(t_1 \ldots t_n) \mid \exists x \mid \neg \mid \phi \vee \psi \mid \phi \,.\, \psi$

$\neg(\phi) := \neg \,.\, \phi \,.\, \neg$

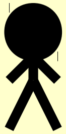
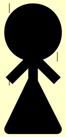
For every ϕ, X M, define the game $G^M{}_X(\phi)$:

Two players,  and  ;

Positions of the form $(\phi_0, \phi_1, \ldots \phi_n, s, \alpha)$,

$\phi_0 \ldots \phi_n \in$ DDL;

s assignment;

$\alpha \in \{I, II\}$; *($\alpha = I$:*  *moves, $\alpha = II$:*  *moves)*

Starting positions (ϕ, s, II), $s \in X$.

Dynamic Dependence Logic

Position *p* **Successors *S(p)***

Dynamic Dependence Logic

Position *p*	**Successors *S(p)***
$(Rt_1...t_n, \phi_1 \ldots \phi_n, s, \alpha)$	$\{(\phi_1 \ldots \phi_n, s, \alpha)\}$ if $s \models_{FO} Rt_1...t_n$, $\emptyset$ otherwise;

Dynamic Dependence Logic

Position *p*	**Successors *S(p)***
$(Rt_1...t_n, \phi_1 \dots \phi_n, s, \alpha)$	$\{(\phi_1 \dots \phi_n, s, \alpha)\}$ if $s \models_{FO} Rt_1...t_n$, $\emptyset$ otherwise;
$(t=t', \phi_1 \dots \phi_n, s, \alpha)$	$\{(\phi_1 \dots \phi_n, s, \alpha)\}$ if $s \models_{FO} t=t'$, $\emptyset$ otherwise;

Dynamic Dependence Logic

Position *p*	**Successors *S(p)***
$(Rt_1...t_n, \phi_1 \ldots \phi_n, s, \alpha)$	$\{(\phi_1 \ldots \phi_n, s, \alpha)\}$ if $s \models_{FO} Rt_1...t_n$, $\emptyset$ otherwise;
$(t=t', \phi_1 \ldots \phi_n, s, \alpha)$	$\{(\phi_1 \ldots \phi_n, s, \alpha)\}$ if $s \models_{FO} t=t'$, $\emptyset$ otherwise;
$(=(t_1...t_n), \phi_1 \ldots \phi_n, s, \alpha)$	$\{(\phi_1 \ldots \phi_n, s, \alpha)\}$;

Dynamic Dependence Logic

Position *p*	**Successors *S(p)***
$(Rt_1 \ldots t_n, \phi_1 \ldots \phi_n, s, \alpha)$	$\{(\phi_1 \ldots \phi_n, s, \alpha)\}$ if $s \models_{FO} Rt_1 \ldots t_n$, $\emptyset$ otherwise;
$(t=t', \phi_1 \ldots \phi_n, s, \alpha)$	$\{(\phi_1 \ldots \phi_n, s, \alpha)\}$ if $s \models_{FO} t=t'$, $\emptyset$ otherwise;
$(=(t_1 \ldots t_n), \phi_1 \ldots \phi_n, s, \alpha)$	$\{(\phi_1 \ldots \phi_n, s, \alpha)\}$;
$(\exists x, \phi_1 \ldots \phi_n, s, \alpha)$	$\{(\phi_1 \ldots \phi_n, s[m/x], \alpha), m \in Dom(M)\}$;

Dynamic Dependence Logic

Position *p*	**Successors *S(p)***
$(Rt_1...t_n, \phi_1 \dots \phi_n, s, \alpha)$	$\{(\phi_1 \dots \phi_n, s, \alpha)\}$ if $s \models_{FO} Rt_1...t_n$, $\varnothing$ otherwise;
$(t=t', \phi_1 \dots \phi_n, s, \alpha)$	$\{(\phi_1 \dots \phi_n, s, \alpha)\}$ if $s \models_{FO} t=t'$, $\varnothing$ otherwise;
$(=(t_1...t_n), \phi_1 \dots \phi_n, s, \alpha)$	$\{(\phi_1 \dots \phi_n, s, \alpha)\}$;
$(\exists x, \phi_1 \dots \phi_n, s, \alpha)$	$\{(\phi_1 \dots \phi_n, s[m/x], \alpha), m \in \mathrm{Dom}(M)\}$;
$(\neg, \phi_1 \dots \phi_n, s, \alpha)$	$\{(\phi_1 \dots \phi_n, s, 1-\alpha)\}$;

Dynamic Dependence Logic

Position *p*	**Successors *S(p)***
$(Rt_1...t_n, \phi_1 \ldots \phi_n, s, \alpha)$	$\{(\phi_1 \ldots \phi_n, s, \alpha)\}$ if $s \models_{FO} Rt_1...t_n$, $\emptyset$ otherwise;
$(t=t', \phi_1 \ldots \phi_n, s, \alpha)$	$\{(\phi_1 \ldots \phi_n, s, \alpha)\}$ if $s \models_{FO} t=t'$, $\emptyset$ otherwise;
$(=(t_1...t_n), \phi_1 \ldots \phi_n, s, \alpha)$	$\{(\phi_1 \ldots \phi_n, s, \alpha)\}$;
$(\exists x, \phi_1 \ldots \phi_n, s, \alpha)$	$\{(\phi_1 \ldots \phi_n, s[m/x], \alpha), m \in \mathrm{Dom}(M)\}$;
$(\neg, \phi_1 \ldots \phi_n, s, \alpha)$	$\{(\phi_1 \ldots \phi_n, s, 1-\alpha)\}$;
$(\psi \vee \theta, \phi_1 \ldots \phi_n, s, \alpha)$	$\{(\psi, \phi_1 \ldots \phi_n, s, \alpha), (\theta, \phi_1 \ldots \phi_n, s, \alpha)\}$;

Dynamic Dependence Logic

Position *p*	**Successors *S(p)***
$(Rt_1...t_n, \phi_1 \dots \phi_n, s, \alpha)$	$\{(\phi_1 \dots \phi_n, s, \alpha)\}$ if $s \models_{FO} Rt_1...t_n$, $\emptyset$ otherwise;
$(t=t', \phi_1 \dots \phi_n, s, \alpha)$	$\{(\phi_1 \dots \phi_n, s, \alpha)\}$ if $s \models_{FO} t=t'$, $\emptyset$ otherwise;
$(=(t_1...t_n), \phi_1 \dots \phi_n, s, \alpha)$	$\{(\phi_1 \dots \phi_n, s, \alpha)\}$;
$(\exists x, \phi_1 \dots \phi_n, s, \alpha)$	$\{(\phi_1 \dots \phi_n, s[m/x], \alpha), m \in \text{Dom}(M)\}$;
$(\neg, \phi_1 \dots \phi_n, s, \alpha)$	$\{(\phi_1 \dots \phi_n, s, 1-\alpha)\}$;
$(\psi \vee \theta, \phi_1 \dots \phi_n, s, \alpha)$	$\{(\psi, \phi_1 \dots \phi_n, s, \alpha), (\theta, \phi_1 \dots \phi_n, s, \alpha)\}$;
$(\psi . \theta, \phi_1 \dots \phi_n, s, \alpha)$	$\{(\psi, \theta, \phi_1 \dots \phi_n, s, \alpha\}$.

Dynamic Dependence Logic

A *strategy* σ for Player α in $\{\mathrm{I}, \mathrm{II}\}$ is a function s.t.

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

Dynamic Dependence Logic

A *strategy* σ for Player α in $\{I, II\}$ is a function s.t.

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

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

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

Dynamic Dependence Logic

A *strategy* σ for Player α in $\{\mathrm{I}, \mathrm{II}\}$ is a function s.t.

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

Given two strategies σ, τ, for I, II in $G^M{}_X(\phi)$,

$\mathrm{Plays}(\sigma, \tau, G^M{}_X(\phi)) = \{ \mathrm{LFP}(\overline{\mathit{Id}} + (\sigma \cup \tau) \circ \mathit{lst}, (\phi, s, \mathrm{II})) : s \in X\}$.

A strategy τ for II in $G^M{}_X(\phi)$ is *uniform* iff whenever

Dynamic Dependence Logic

A *strategy* σ for Player α in $\{\mathrm{I}, \mathrm{II}\}$ is a function s.t.

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

Given two strategies σ, τ, for I, II in $G^M{}_X(\phi)$,

$\mathrm{Plays}(\sigma, \tau, G^M{}_X(\phi)) = \{ \mathrm{LFP}(\overline{\mathit{Id}} + (\sigma \cup \tau) \circ \mathit{lst}, (\phi, s, \mathrm{II})) : s \in X\}$.

A strategy τ for II in $G^M{}_X(\phi)$ is *uniform* iff whenever

- $\overline{p} \in \mathrm{Plays}(\sigma, \tau, G^M{}_X(\phi))$, $\overline{p}' \in \mathrm{Plays}(\sigma', \tau, G^M{}_X(\phi))$, and

Dynamic Dependence Logic

A *strategy* σ for Player α in $\{\mathrm{I}, \mathrm{II}\}$ 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 σ, τ, for I, II in $G^M_X(\phi)$,

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

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

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

Dynamic Dependence Logic

A *strategy* σ for Player α in {I, II} is a function s.t.

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

Given two strategies σ, τ, for I, II in $G^M{}_X(\phi)$,

$\mathrm{Plays}(\sigma, \tau, G^M{}_X(\phi)) = \{ \mathrm{LFP}(\overline{Id} + (\sigma \cup \tau) \circ \mathit{lst}, (\phi, s, \mathrm{II})) : s \in X\}$.

A strategy τ for II in $G^M{}_X(\phi)$ is *uniform* iff whenever

- $\overline{p} \in \mathrm{Plays}(\sigma, \tau, G^M{}_X(\phi))$, $\overline{p}' \in \mathrm{Plays}(\sigma', \tau, G^M{}_X(\phi))$, and
- $(=(t_1...t_n), \phi_1 \ldots \phi_n, s, \mathrm{II})$, $(=(t_1...t_n), \phi_1 \ldots \phi_n, s', \mathrm{II})$ occurs in $\overline{p}$, $\overline{p}'$ for the same instance of $=(t_1...t_n)$, 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 τ for II in $G^M_X(\phi)$ s.t., for all strategies σ of I,

Dynamic Dependence Logic

We say that $\phi: X \to Y$ if there exists a uniform strategy τ for II in $G^M{}_X(\phi)$ s.t., for all strategies σ of I,

$$\overline{p} = p_1 \ldots p_n \in \text{Plays}(\sigma, \tau, G^M{}_X(\phi)) \Rightarrow p_n = (\varepsilon, s, \mathrm{II}) \text{ for some } s \in Y$$

Dynamic Dependence Logic

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

$\overline{p} = p_1 ... p_n \in$ Plays(σ, τ, $G^M_X(\phi)$) $\Rightarrow$ p_n= (ε, s, II) for some s $\in$ Y

Theorem

$(X,Y) \models^+ \phi$ iff ϕ: X $\rightarrow$ Y

$(X,Y) \models^- \phi$ iff $\neg(\phi)$: X $\rightarrow$ Y

Dynamic Dependence Logic

M = ({0,1})

Dynamic Dependence Logic

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

$(\exists x \vee \exists y) \,.\, (x=y)$

Dynamic Dependence Logic

M = ({0,1})

$(\exists x \vee \exists y) \,.\, (x{=}y)$

X =

	x	y	...
s_0	0	0	...
s_1	0	1	...

Dynamic Dependence Logic

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

$((\exists x \vee \exists y) \,.\, (x=y), s, \mathrm{II})$ $(s \in X)$

X =

	x	**y**	**...**
s_0	0	0	...
s_1	0	1	...

Dynamic Dependence Logic

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

$X =$

	x	y	...
s_0	0	0	...
s_1	0	1	...

$((\exists x \vee \exists y) \,.\, (x{=}y), s, \mathrm{II})$ *($s \in X$)*

↓

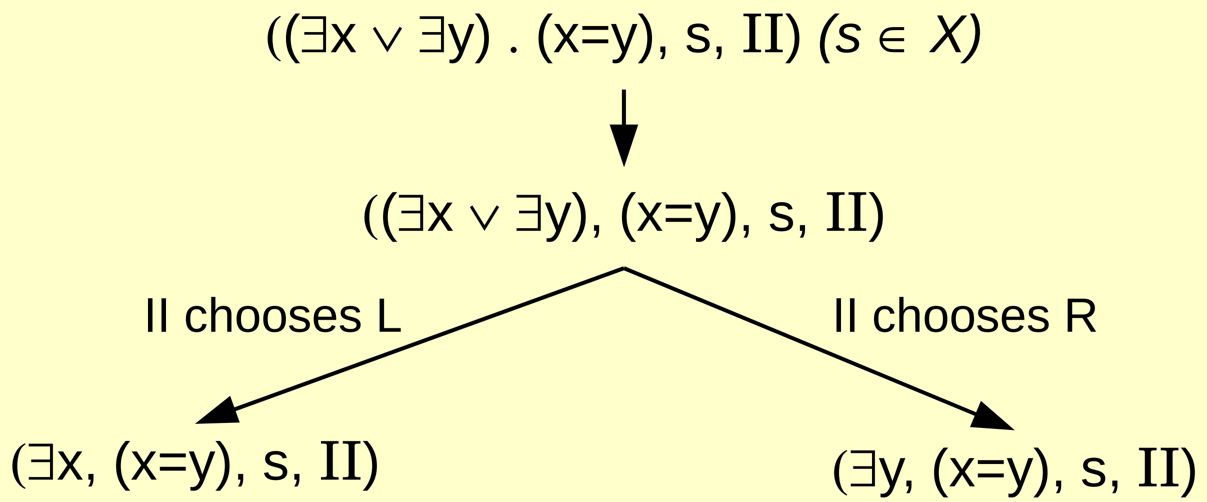
$((\exists x \vee \exists y), (x{=}y), s, \mathrm{II})$

Dynamic Dependence Logic

M = ({0,1})

X =

	x	y	...
s_0	0	0	...
s_1	0	1	...

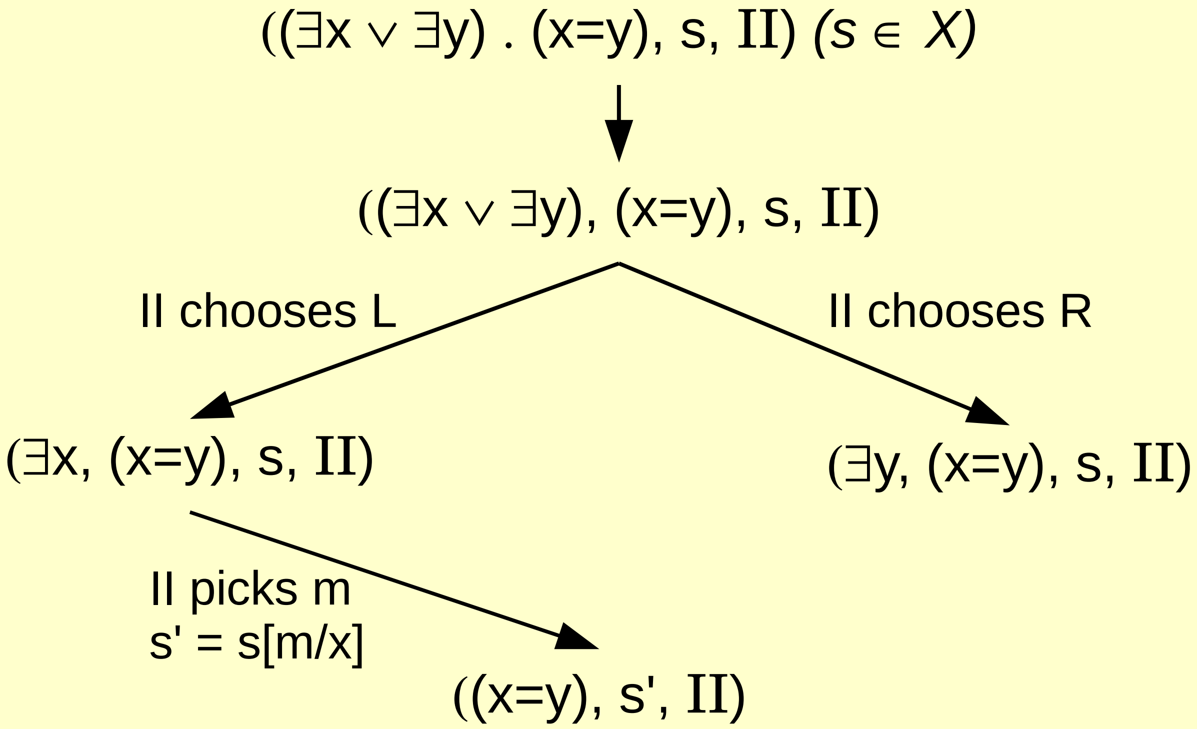


Dynamic Dependence Logic

M = ({0,1})

X =

	x	y	...
s_0	0	0	...
s_1	0	1	...

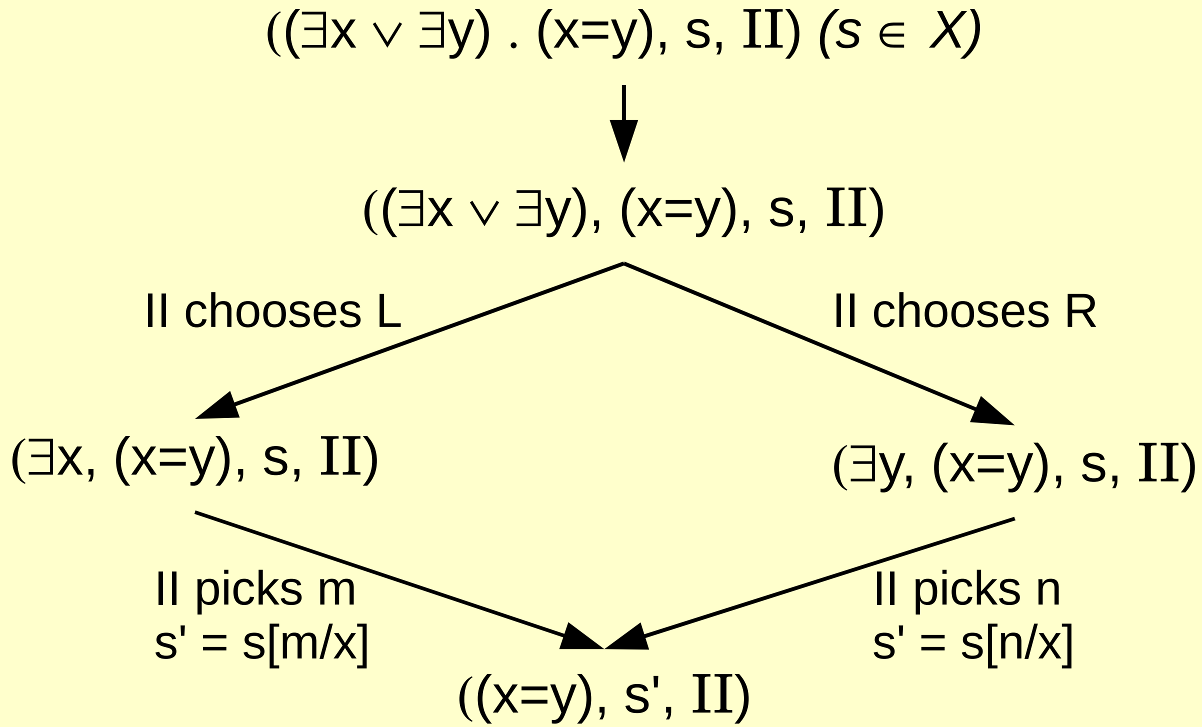


Dynamic Dependence Logic

M = ({0,1})

X =

	x	y	...
s_0	0	0	...
s_1	0	1	...

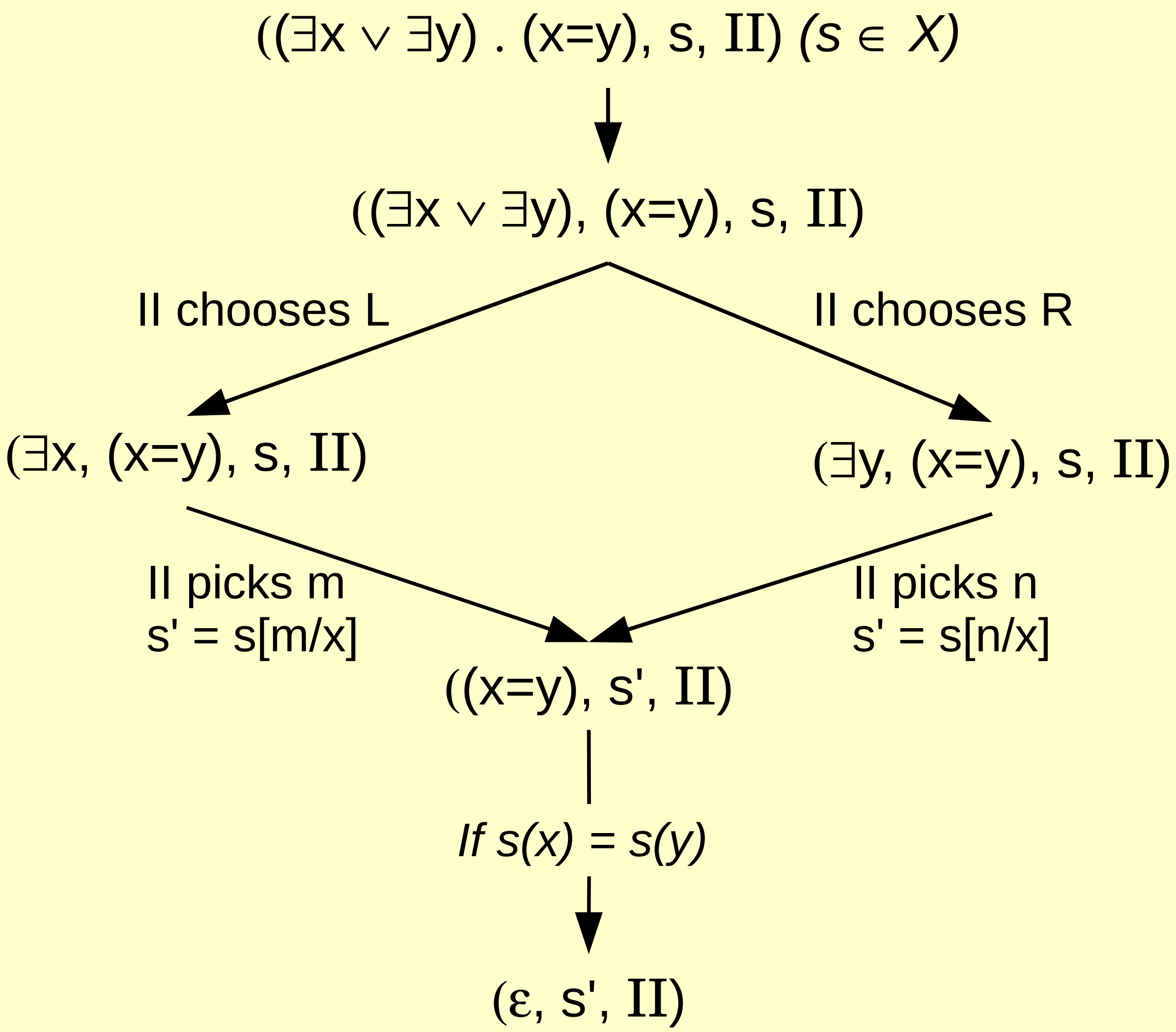


Dynamic Dependence Logic

M = ({0,1})

X =

	x	y	...
s_0	0	0	...
s_1	0	1	...

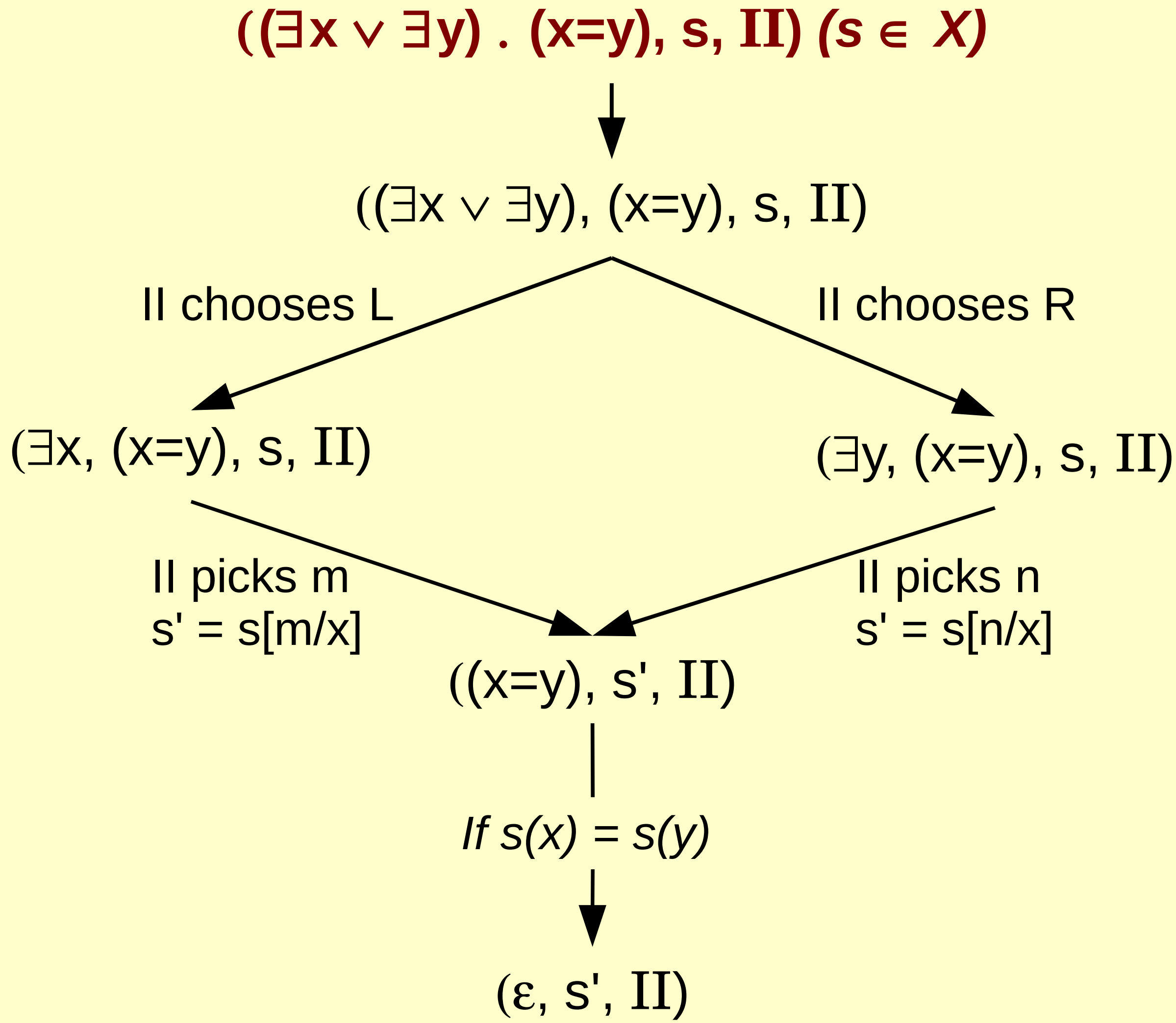


Dynamic Dependence Logic

M = ({0,1})

X =

	x	y	...
s_0	0	0	...
s_1	0	1	...

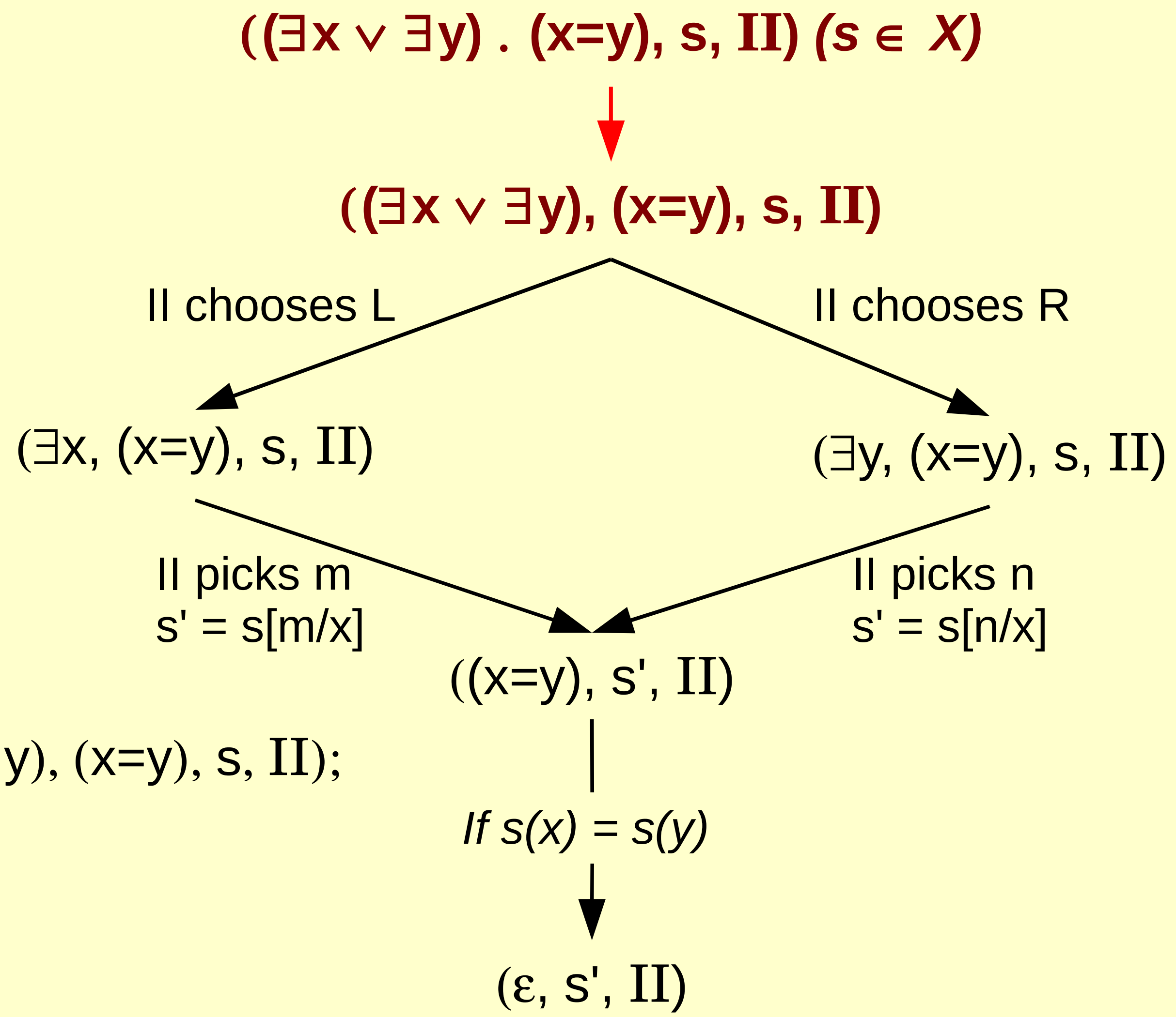


Dynamic Dependence Logic

M = ({0,1})

X =

	x	y	...
s_0	0	0	...
s_1	0	1	...



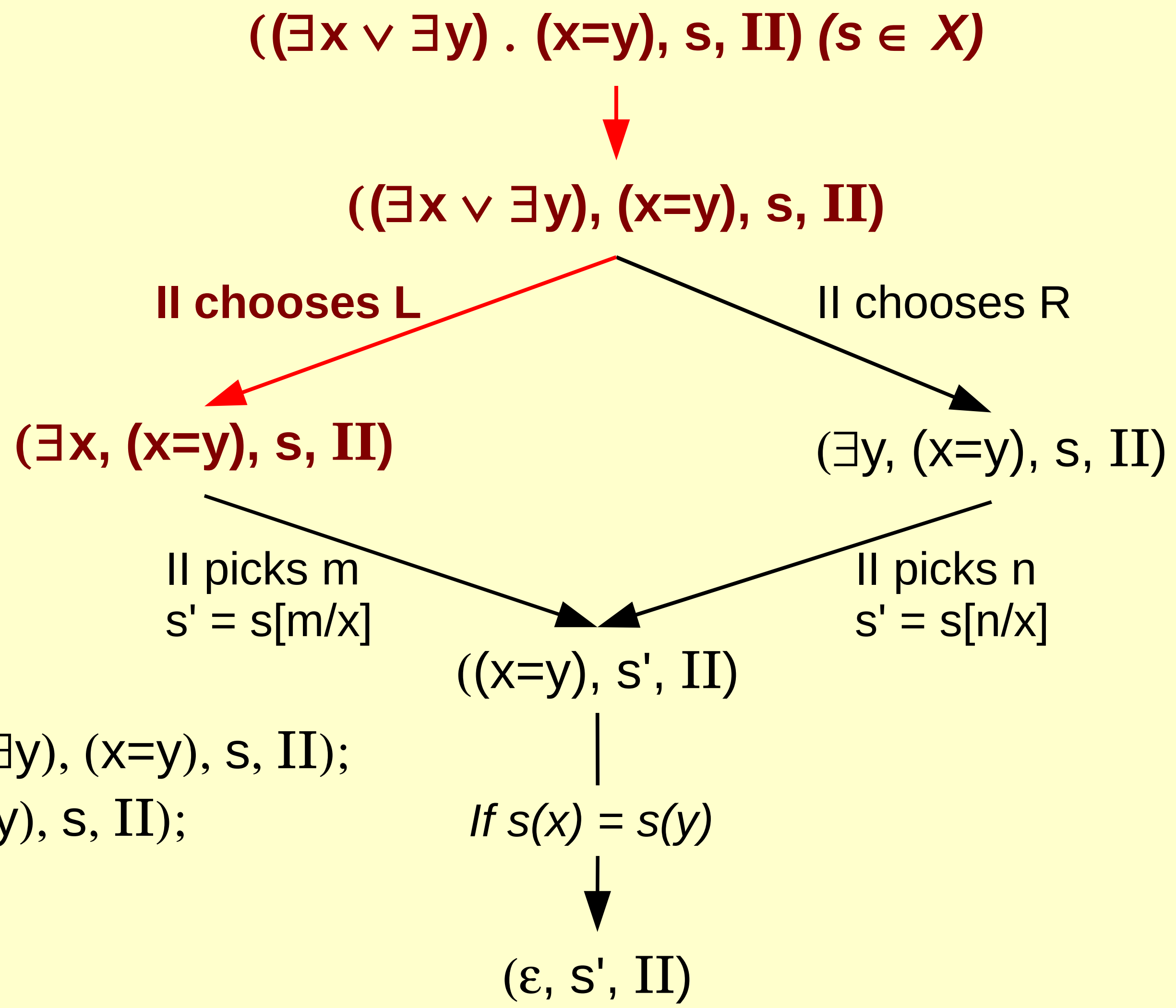
$\tau((\exists x \vee \exists y) \times (x=y), s, II) = ((\exists x \vee \exists y), (x=y), s, II);$

Dynamic Dependence Logic

M = ({0,1})

X =

	x	y	...
s_0	0	0	...
s_1	0	1	...



τ((∃x ∨ ∃y) ×(x=y), s, II) = ((∃x ∨ ∃y), (x=y), s, II);

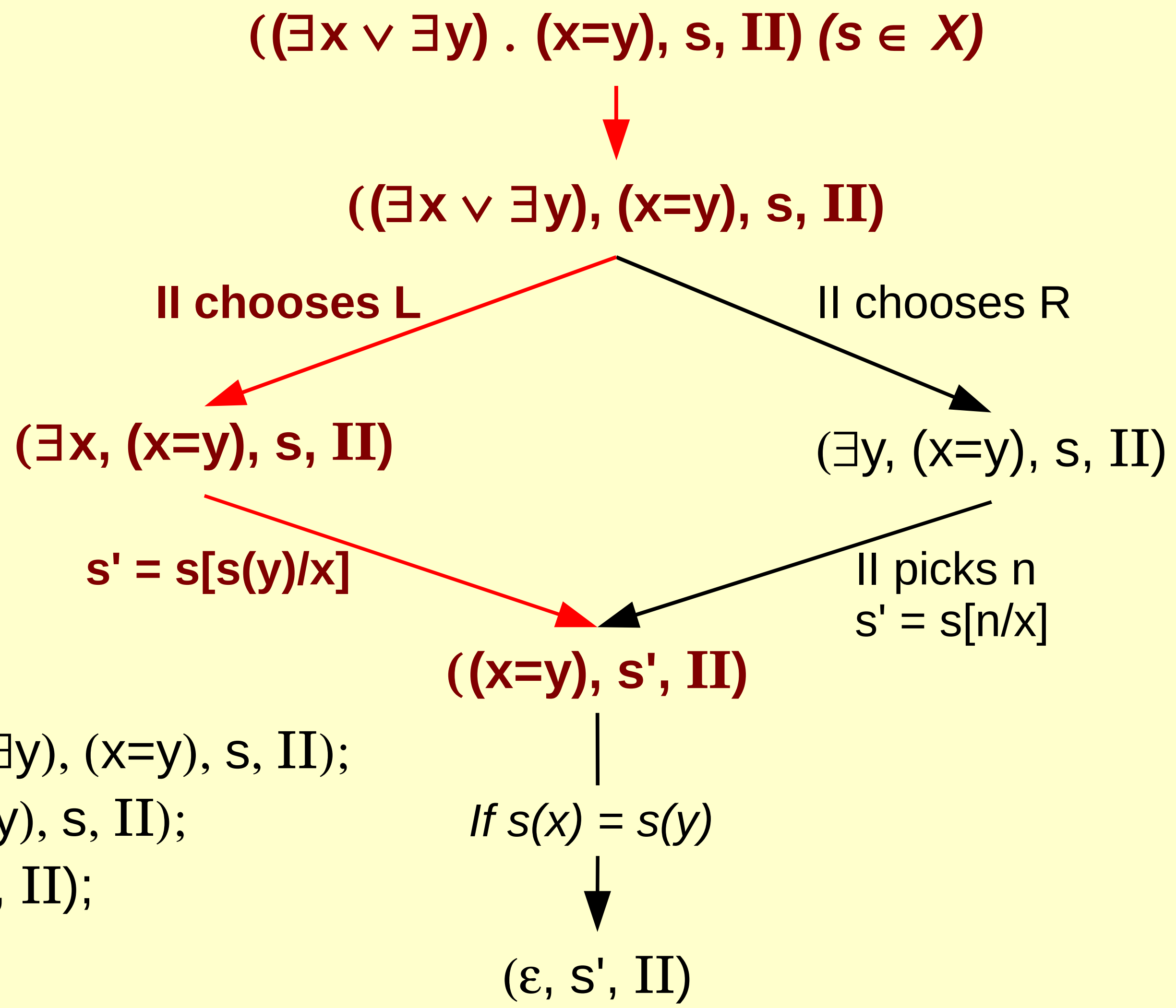
τ((∃x ∨ ∃y), (x=y), s, II) = (∃x, (x=y), s, II);

Dynamic Dependence Logic

M = ({0,1})

X =

	x	y	...
s_0	0	0	...
s_1	0	1	...



τ((∃x ∨ ∃y) ×(x=y), s, II) = ((∃x ∨ ∃y), (x=y), s, II);

τ((∃x ∨ ∃y), (x=y), s, II) = (∃x, (x=y), s, II);

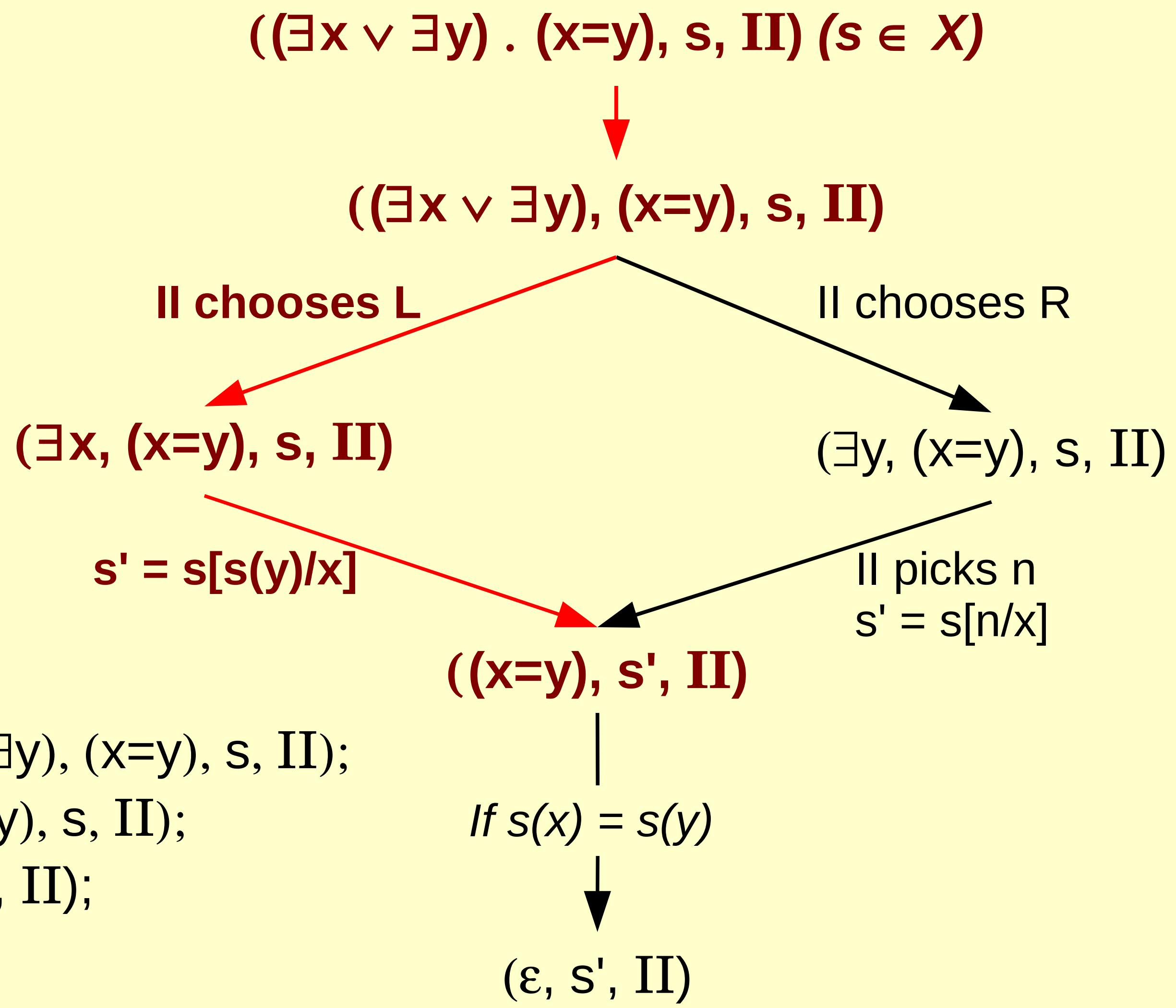
τ(∃x, (x=y), s, II) = (x=y, s[s(y)/x], II);

Dynamic Dependence Logic

M = ({0,1})

X' =

	x	y	...
s_0	0	0	...
s'_1	*1*	*1*	...



$\tau((\exists x \vee \exists y) \times (x{=}y), s, II) = ((\exists x \vee \exists y), (x{=}y), s, II);$

$\tau((\exists x \vee \exists y), (x{=}y), s, II) = (\exists x, (x{=}y), s, II);$

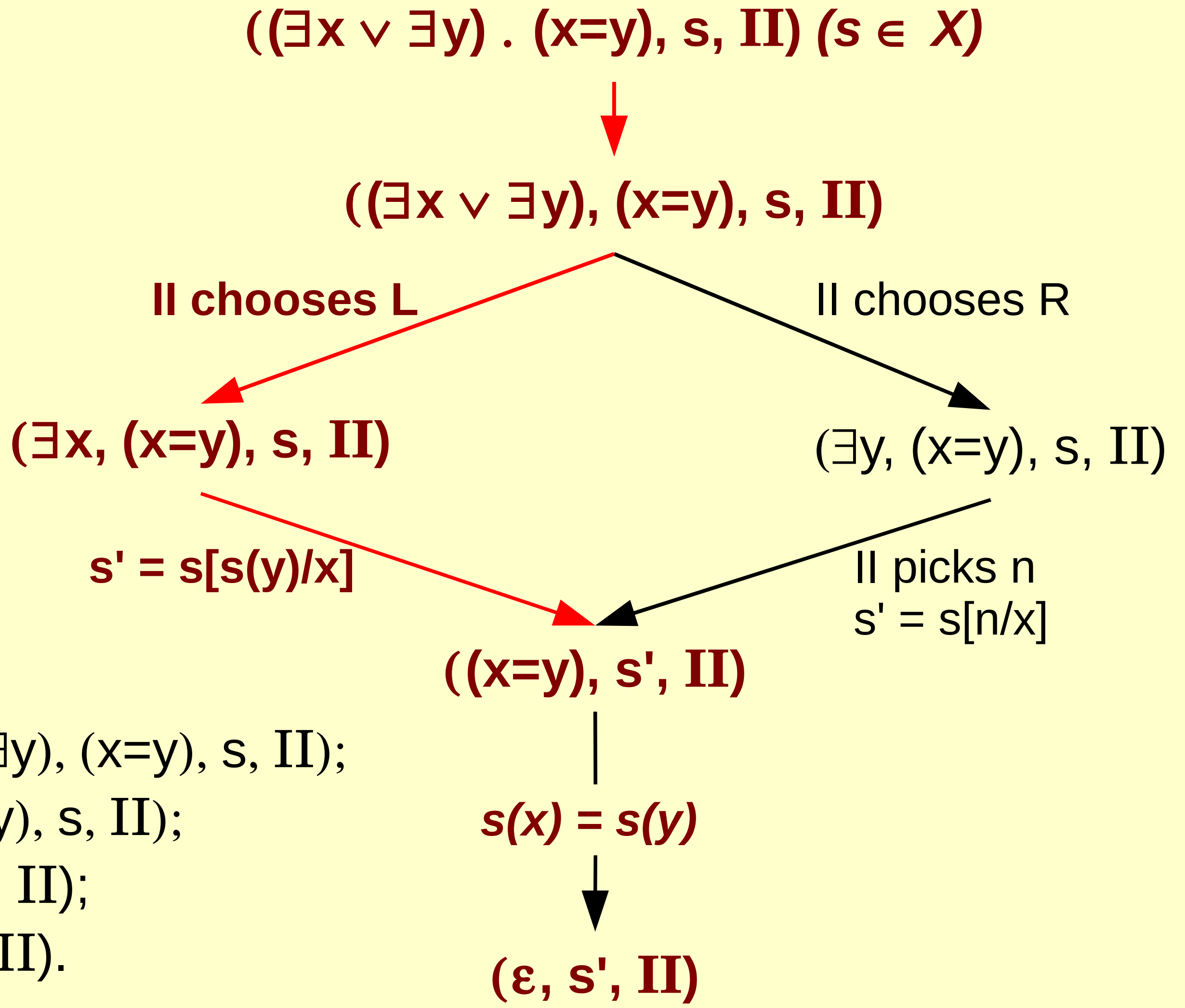
$\tau(\exists x, (x{=}y), s, II) = (x{=}y, s[s(y)/x], II);$

Dynamic Dependence Logic

M = ({0,1})

X' =

	x	y	...
s_0	0	0	...
s'_1	*1*	*1*	...



τ((∃x ∨ ∃y) ×(x=y), s, II) = ((∃x ∨ ∃y), (x=y), s, II);

τ((∃x ∨ ∃y), (x=y), s, II) = (∃x, (x=y), s, II);

τ(∃x, (x=y), s, II) = (x=y, s[s(y)/x], II);

τ(x=y, s[s(y)/x], II) = (ε, s[s(y)/x], II).

Dynamic Dependence Logic

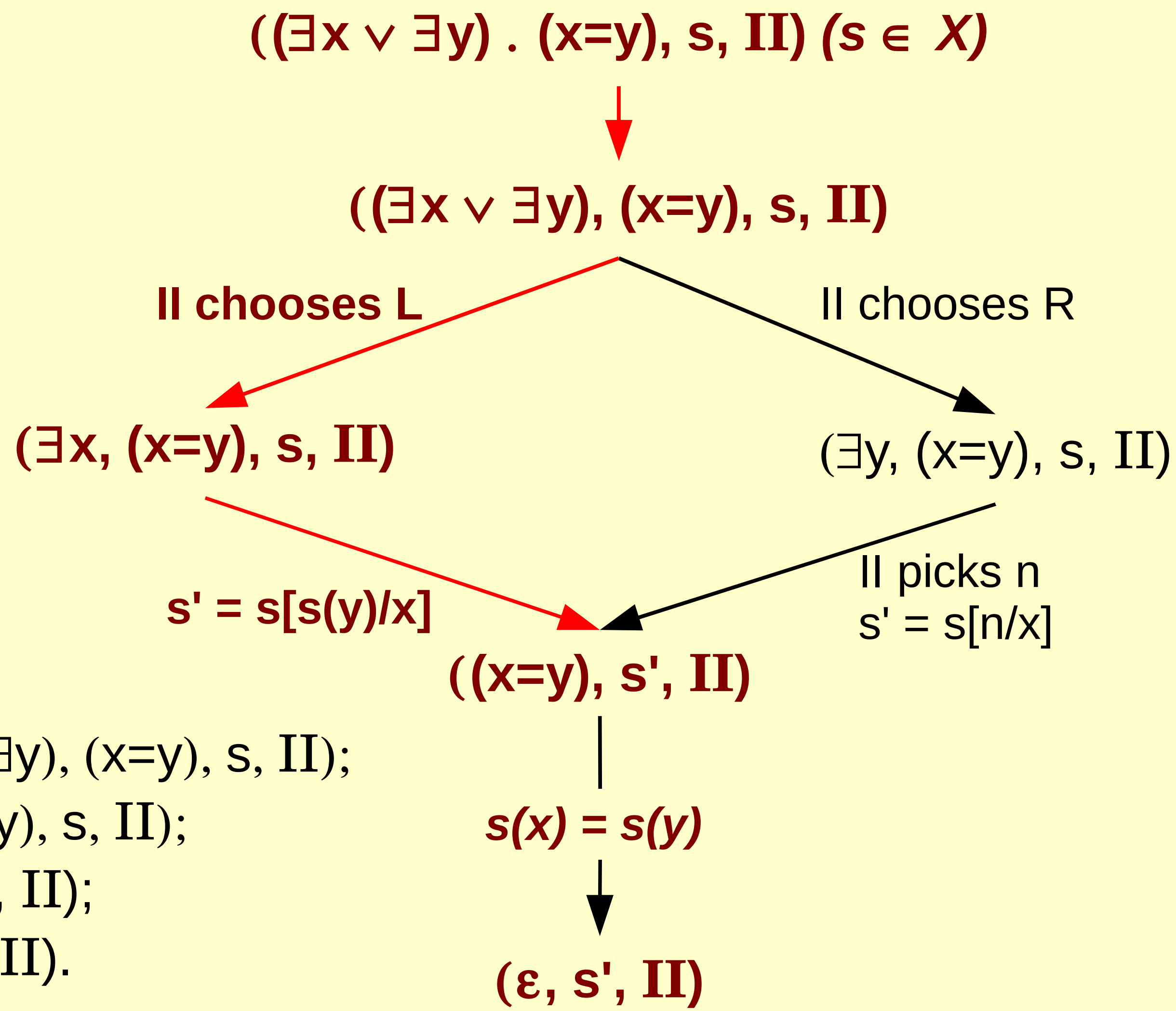
M = ({0,1})

X =

	x	y	...
s_0	0	0	...
s_1	0	1	...

X' =

	x	y	...
s'_0	0	0	...
s'_1	***1***	*1*	*...*



τ((∃x ∨ ∃y) ×(x=y), s, II) = ((∃x ∨ ∃y), (x=y), s, II);

τ((∃x ∨ ∃y), (x=y), s, II) = (∃x, (x=y), s, II);

τ(∃x, (x=y), s, II) = (x=y, s[s(y)/x], II);

τ(x=y, s[s(y)/x], II) = (ε, s[s(y)/x], II).

(∃x ∨ ∃y) . (x=y): X → X'

Dynamic Dependence Logic

M = ({0,1})

X =

	x	y	...
s_0	0	0	...
s_1	0	1	...

Dynamic Dependence Logic

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

$(\exists x \vee \exists y) \,.\, (x{=}y) \,.\, {=}(x)$

X =

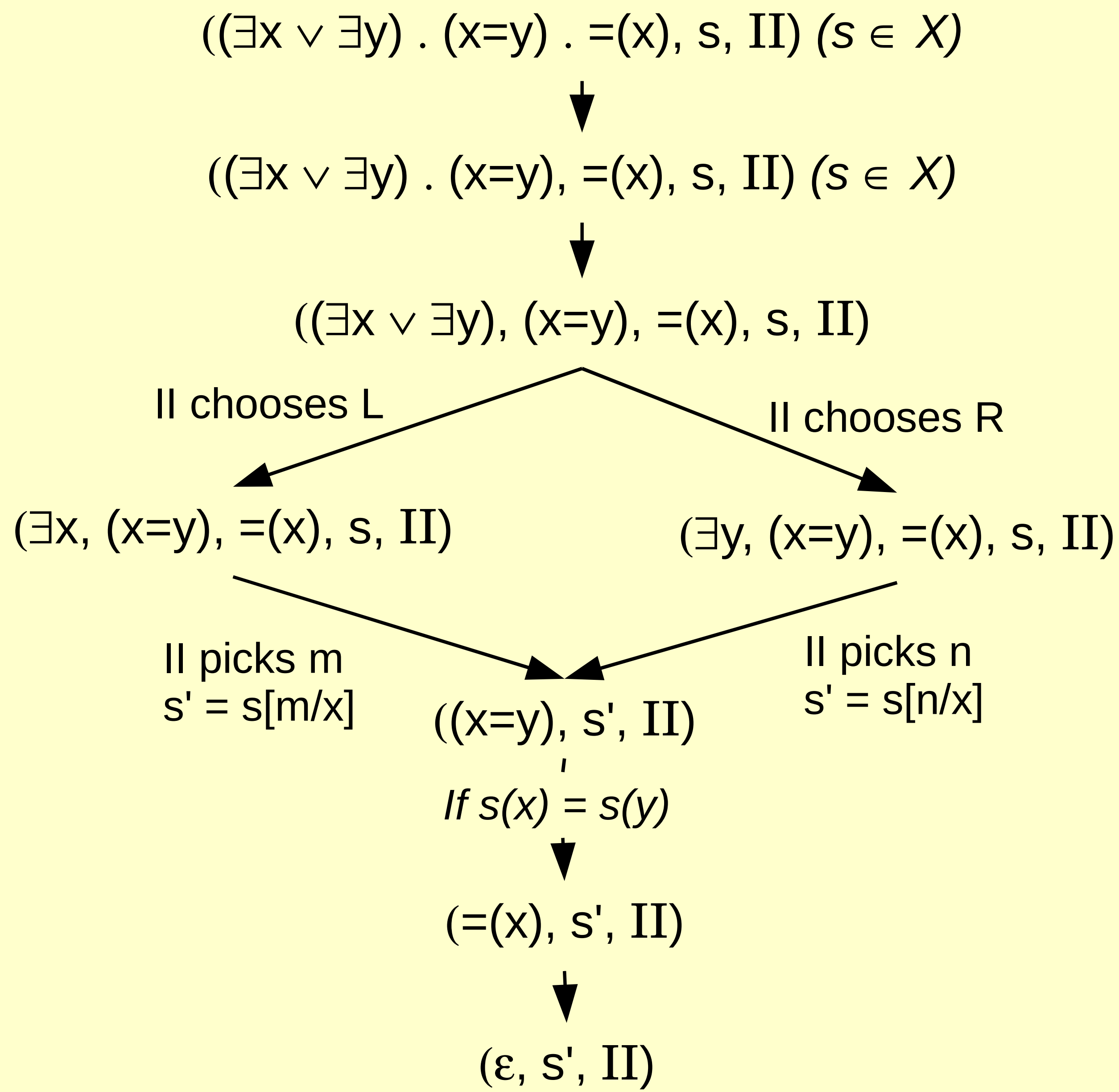
	x	y	...
s_0	0	0	...
s_1	0	1	...

Dynamic Dependence Logic

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

$X =$

	x	y	...
s_0	0	0	...
s_1	0	1	...

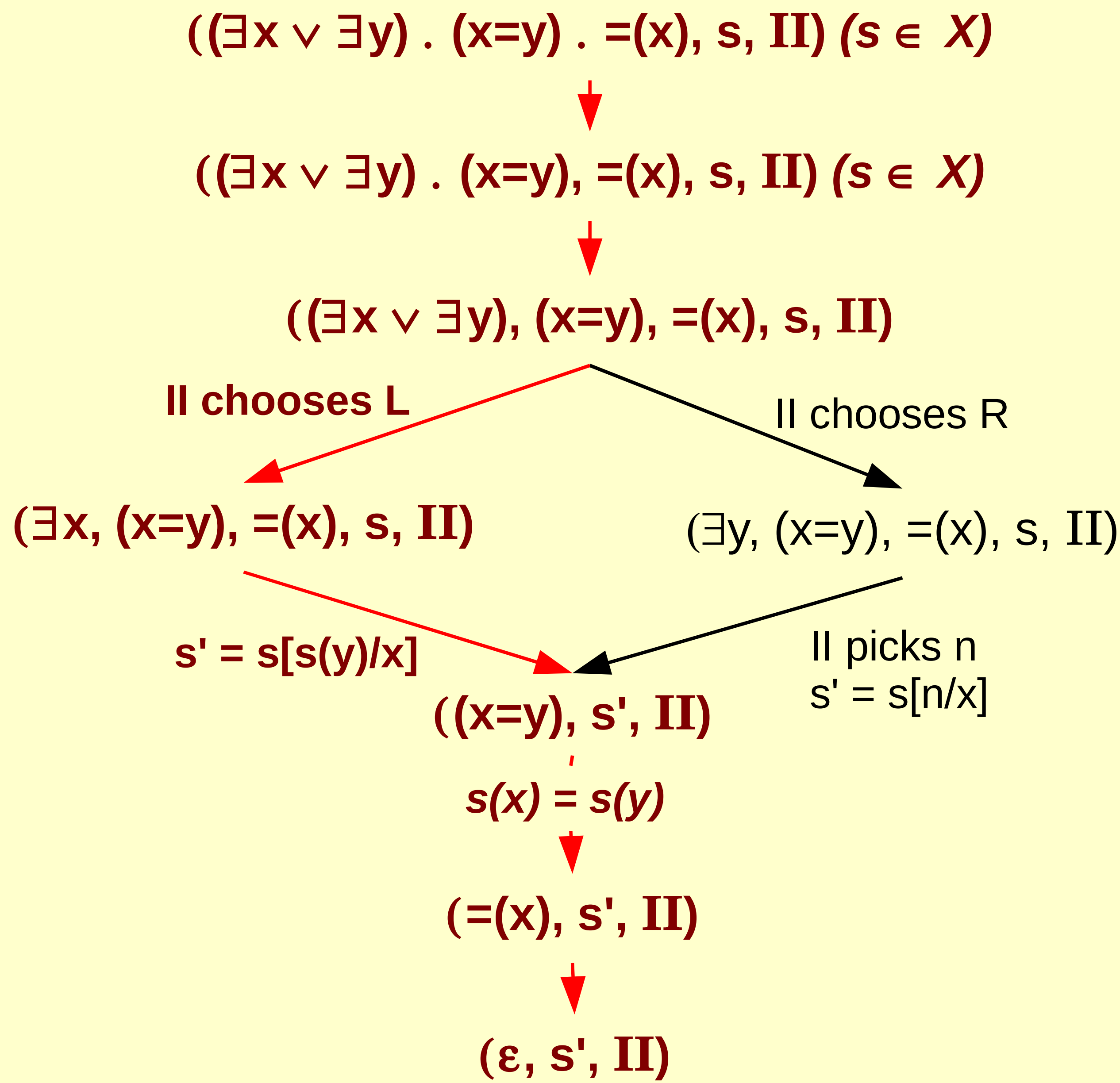


Dynamic Dependence Logic

M = ({0,1})

X =

	x	y	...
s_0	0	0	...
s_1	0	1	...



Dynamic Dependence Logic

M = ({0,1})

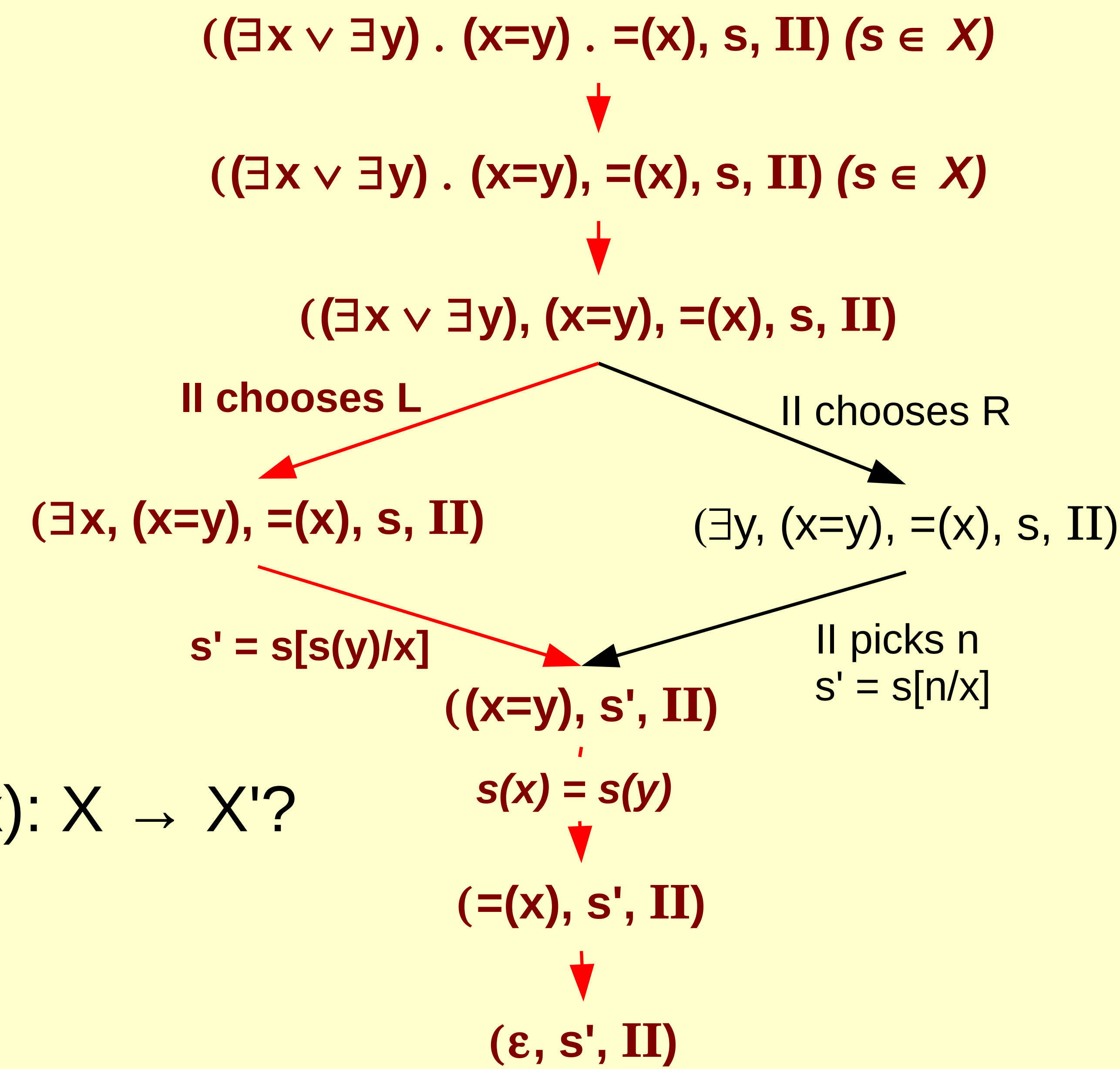
X =

	x	y	...
s_0	0	0	...
s_1	0	1	...

X' =

	x	y	...
s'_0	0	0	...
s'_1	1	1	...

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



Dynamic Dependence Logic

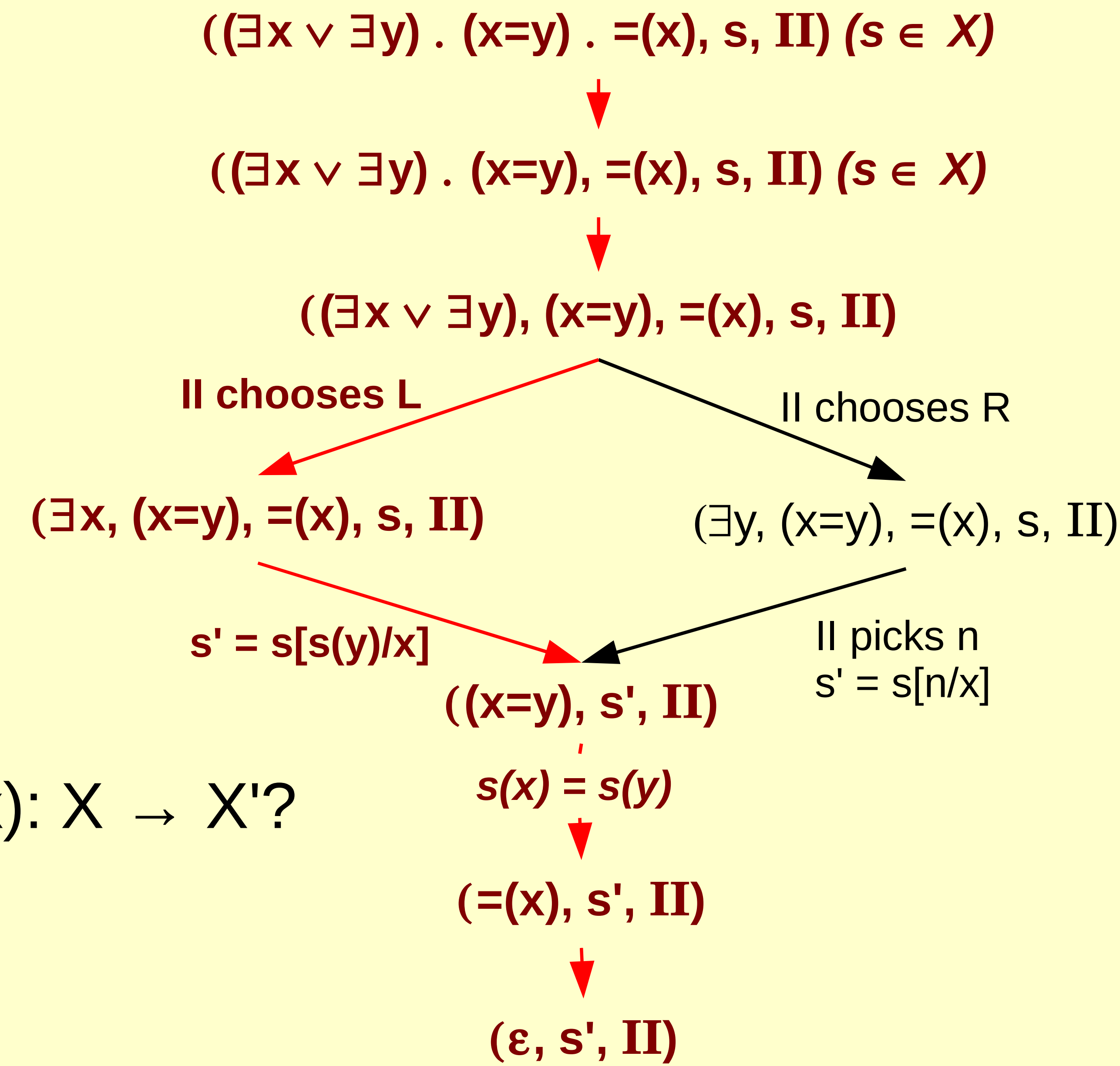
M = ({0,1})

X =

	x	y	...
s_0	0	0	...
s_1	0	1	...

X' =

	x	y	...
s'_0	0	0	...
s'_1	1	1	...



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

No: not uniform!

Dynamic Dependence Logic

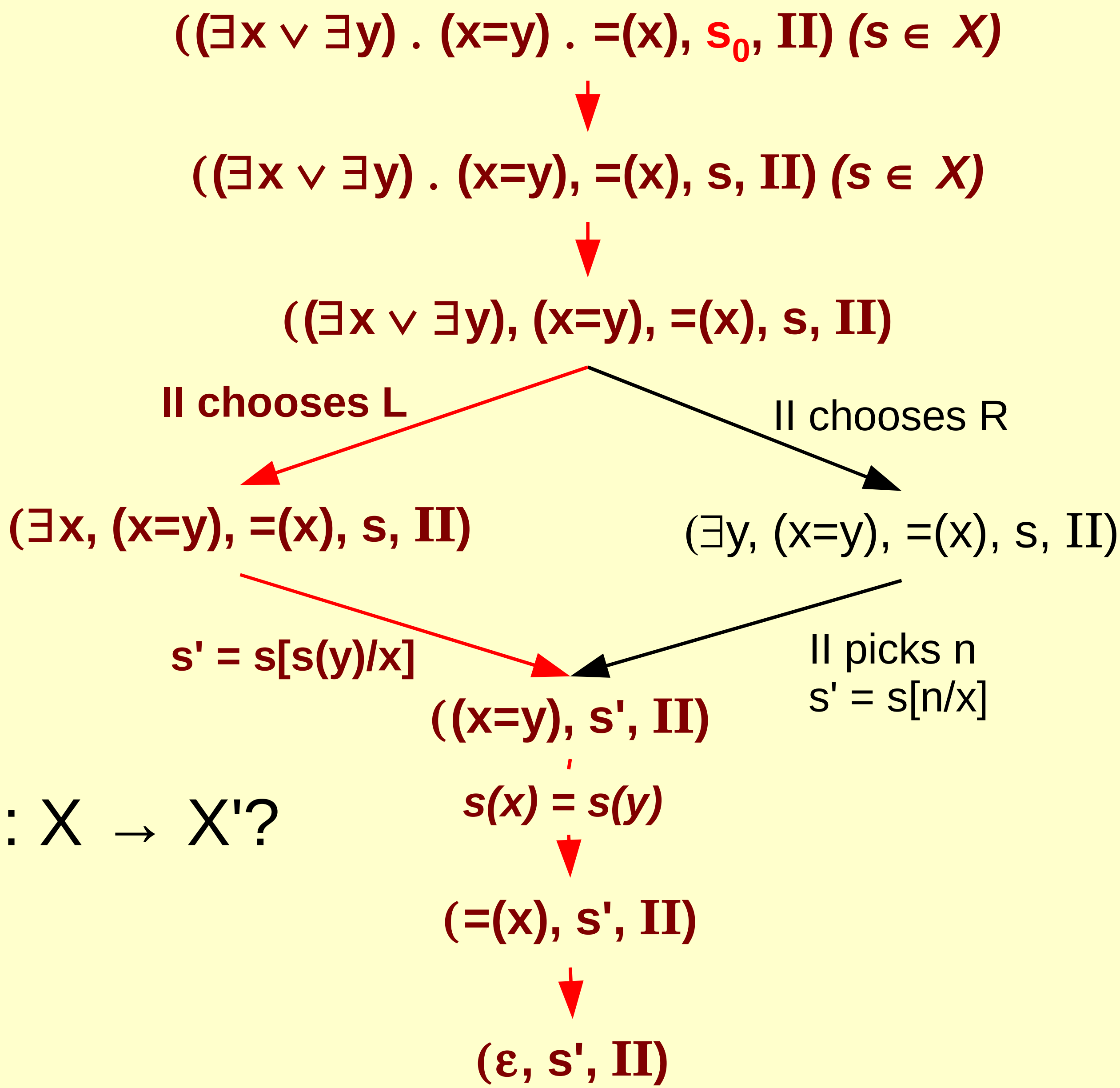
$M = (\{0,1\})$

$X =$

	x	y	...
s_0	*0*	*0*	*...*
s_1	0	1	...

$X' =$

	x	y	...
s'_0	0	0	...
s'_1	1	1	...



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

No: not uniform!

Dynamic Dependence Logic

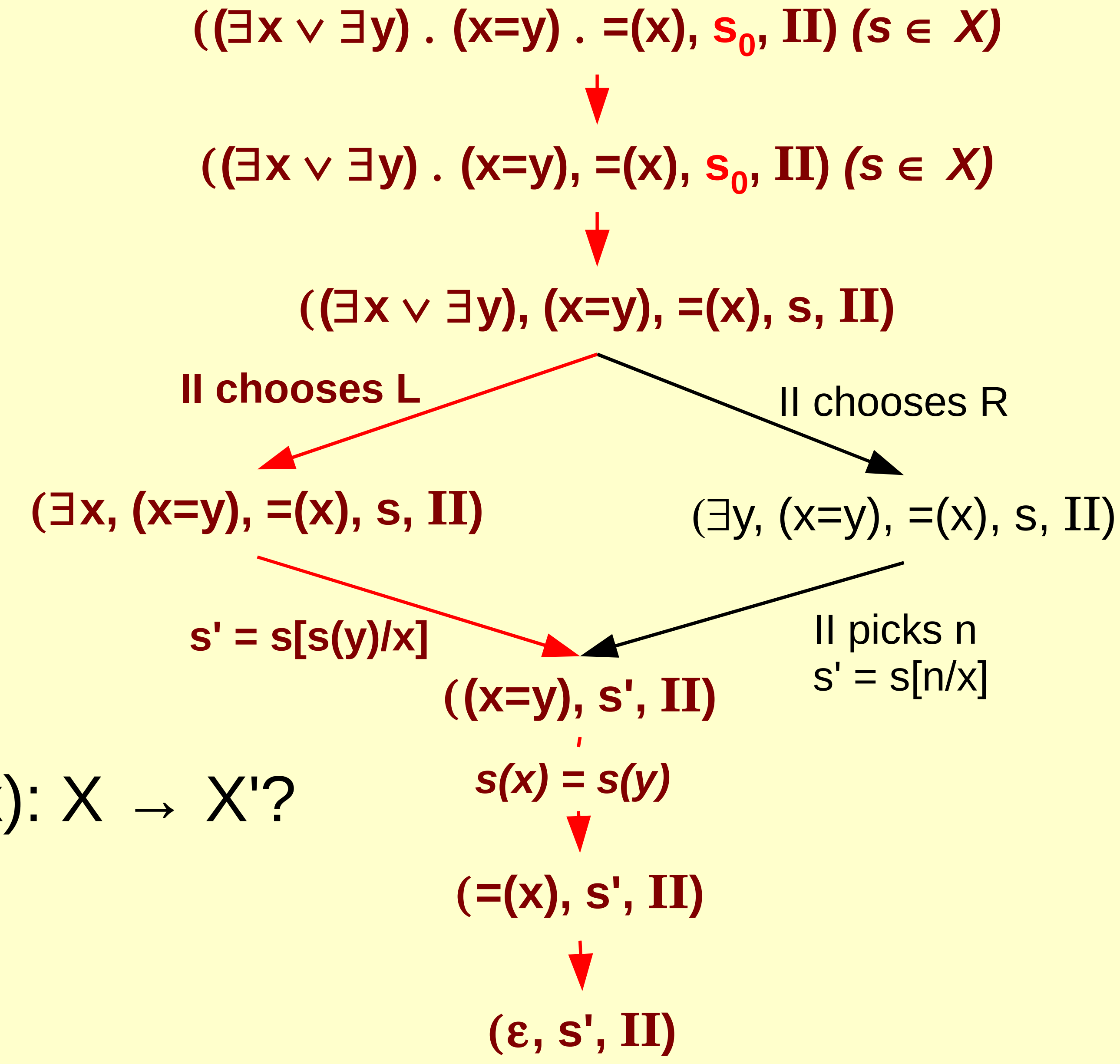
$M = (\{0,1\})$

X =

	x	y	...
s_0	0	0	...
s_1	0	1	...

X' =

	x	y	...
s'_0	0	0	...
s'_1	1	1	...



$(\exists x \vee \exists y)\ .\ (x{=}y)\ .\ {=}(x) : X \rightarrow X'$?

No: not uniform!

Dynamic Dependence Logic

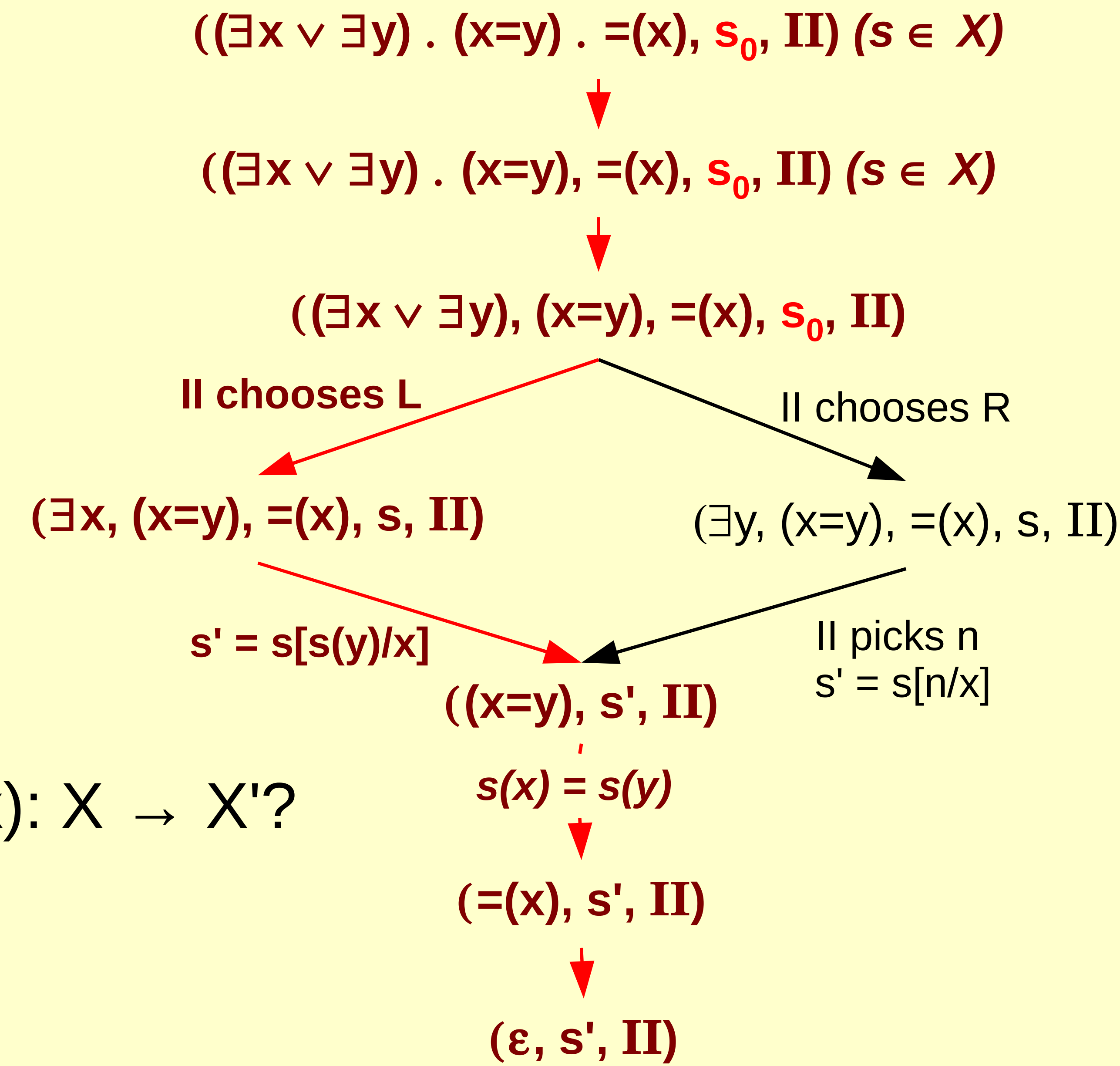
M = ({0,1})

X =

	x	y	...
s_0	*0*	*0*	*...*
s_1	0	1	...

X' =

	x	y	...
s'_0	0	0	...
s'_1	1	1	...



$(\exists x \vee \exists y) \,.\, (x=y) \,.\, {=}(x): X \rightarrow X'$?

No: not uniform!

Dynamic Dependence Logic

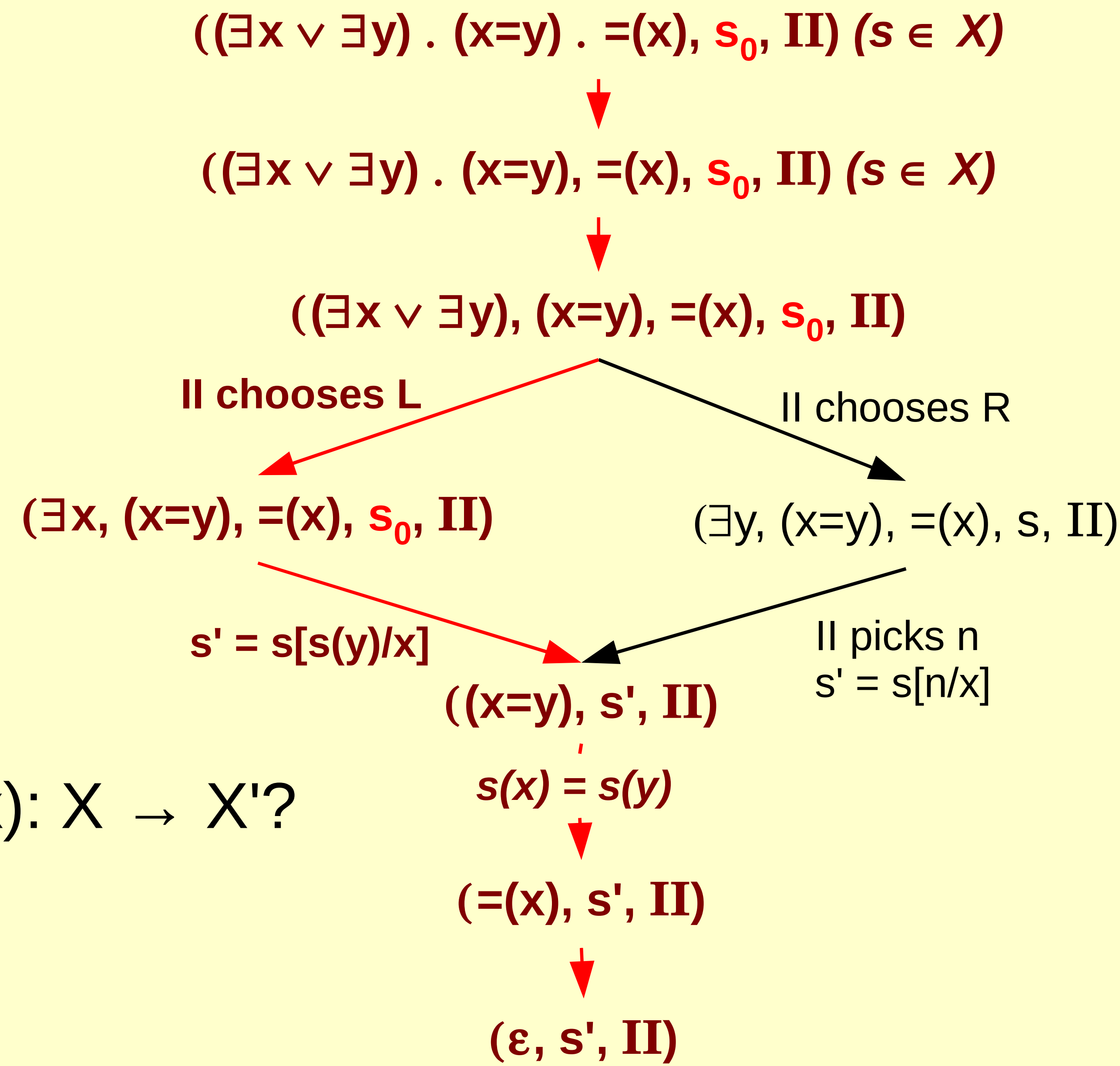
$M = (\{0,1\})$

$X =$

	x	y	...
s_0	0	0	...
s_1	0	1	...

$X' =$

	x	y	...
s'_0	0	0	...
s'_1	1	1	...



$(\exists x \vee \exists y) \,.\, (x=y) \,.\, {=}(x): X \rightarrow X'$?

No: not uniform!

Dynamic Dependence Logic

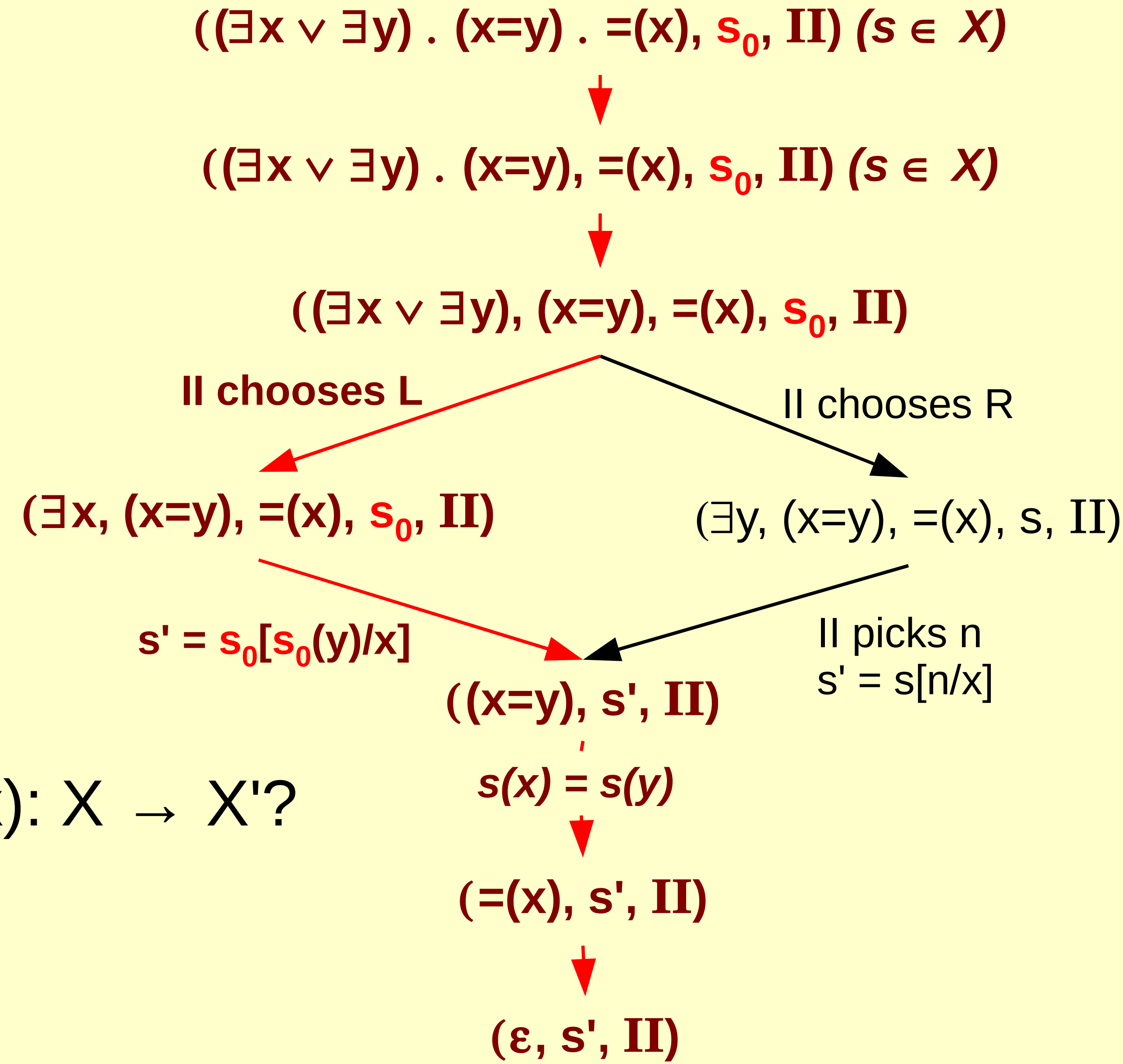
M = ({0,1})

X =

	x	y	...
s_0	0	0	...
s_1	0	1	...

X' =

	x	y	...
s'_0	0	0	...
s'_1	1	1	...



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

No: not uniform!

Dynamic Dependence Logic

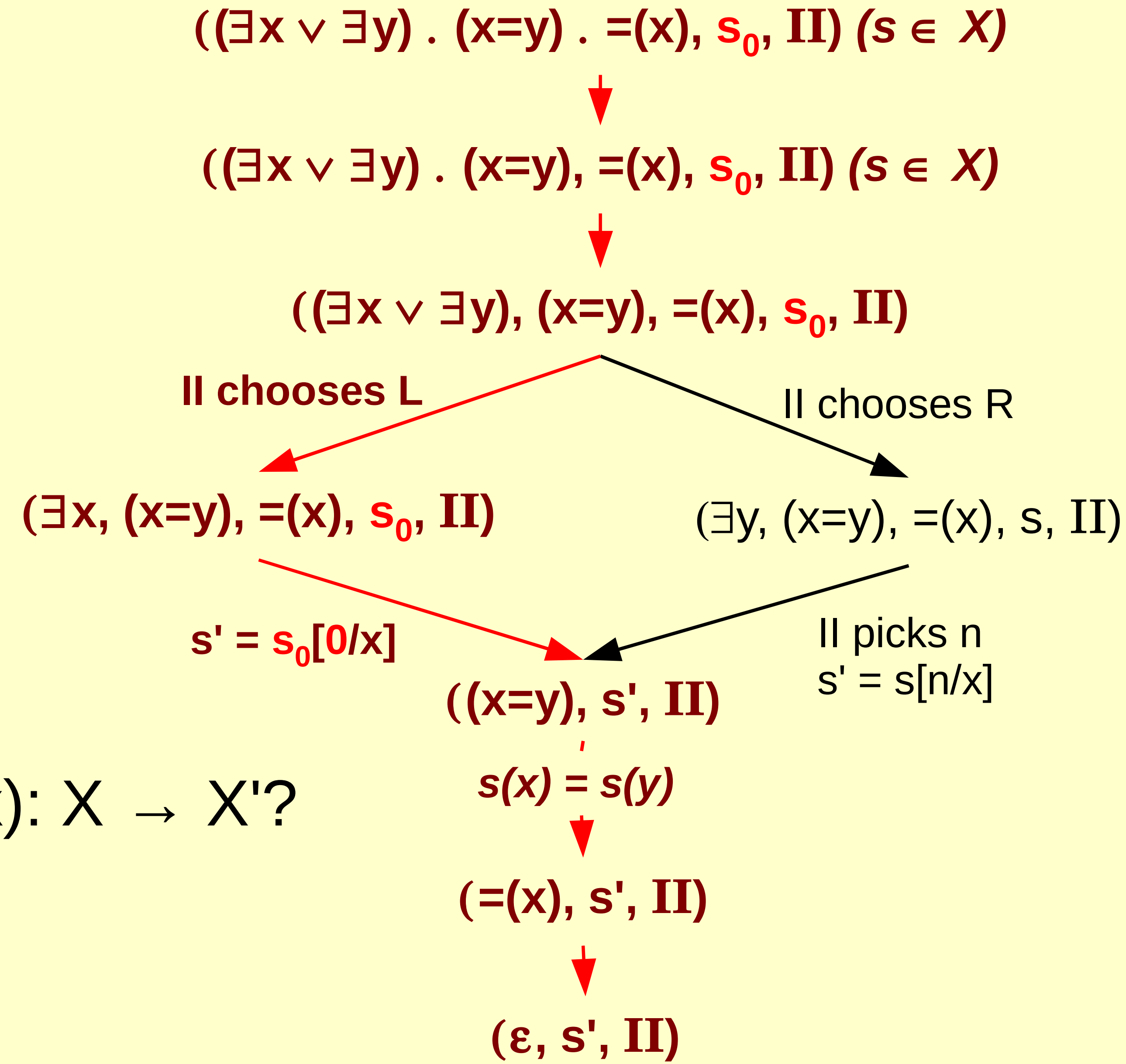
$M = (\{0,1\})$

$X =$

	x	y	...
s_0	*0*	*0*	*...*
s_1	0	1	...

$X' =$

	x	y	...
s'_0	0	0	...
s'_1	1	1	...



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

No: not uniform!

Dynamic Dependence Logic

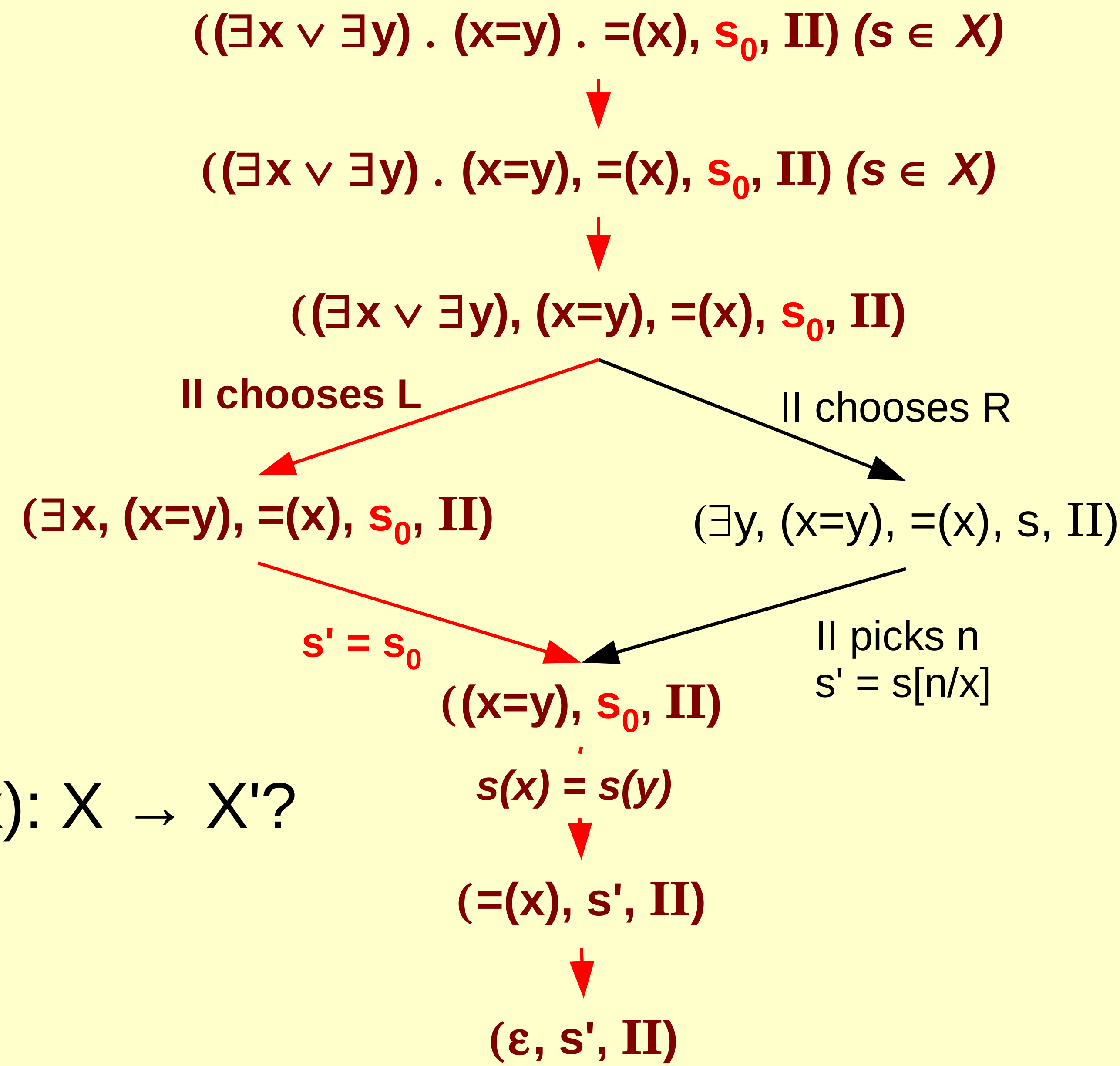
M = ({0,1})

X =

	x	y	...
s_0	*0*	*0*	*...*
s_1	0	1	...

X' =

	x	y	...
s'_0	0	0	...
s'_1	1	1	...



(∃x ∨ ∃y) . (x=y) . =(x): X → X'?

No: not uniform!

Dynamic Dependence Logic

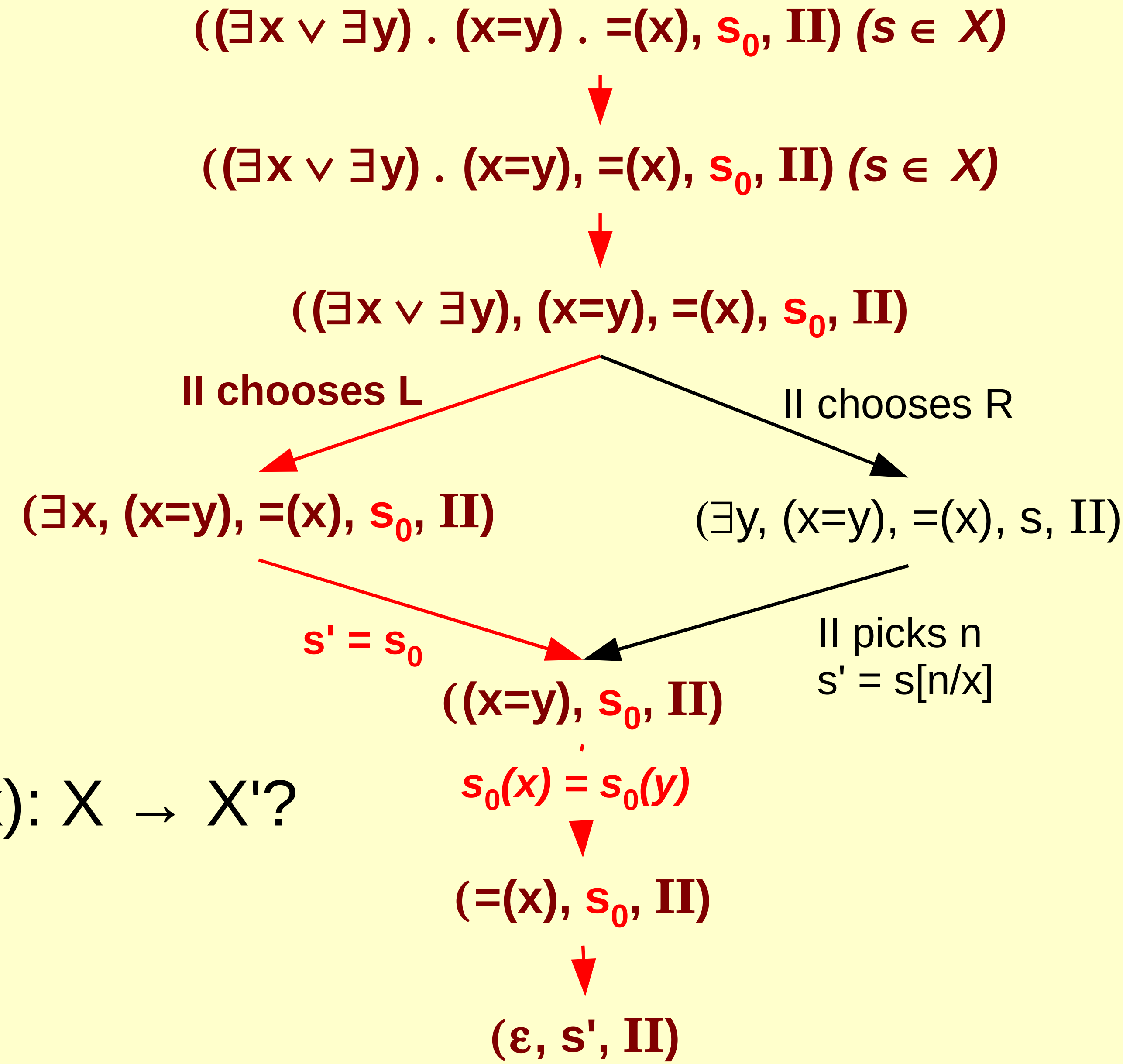
$M = (\{0,1\})$

$X =$

	x	**y**	**...**
$\boldsymbol{s_0}$	*0*	*0*	*...*
s_1	0	1	...

$X' =$

	x	**y**	...
s'_0	0	0	...
s'_1	1	1	...



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

No: not uniform!

Dynamic Dependence Logic

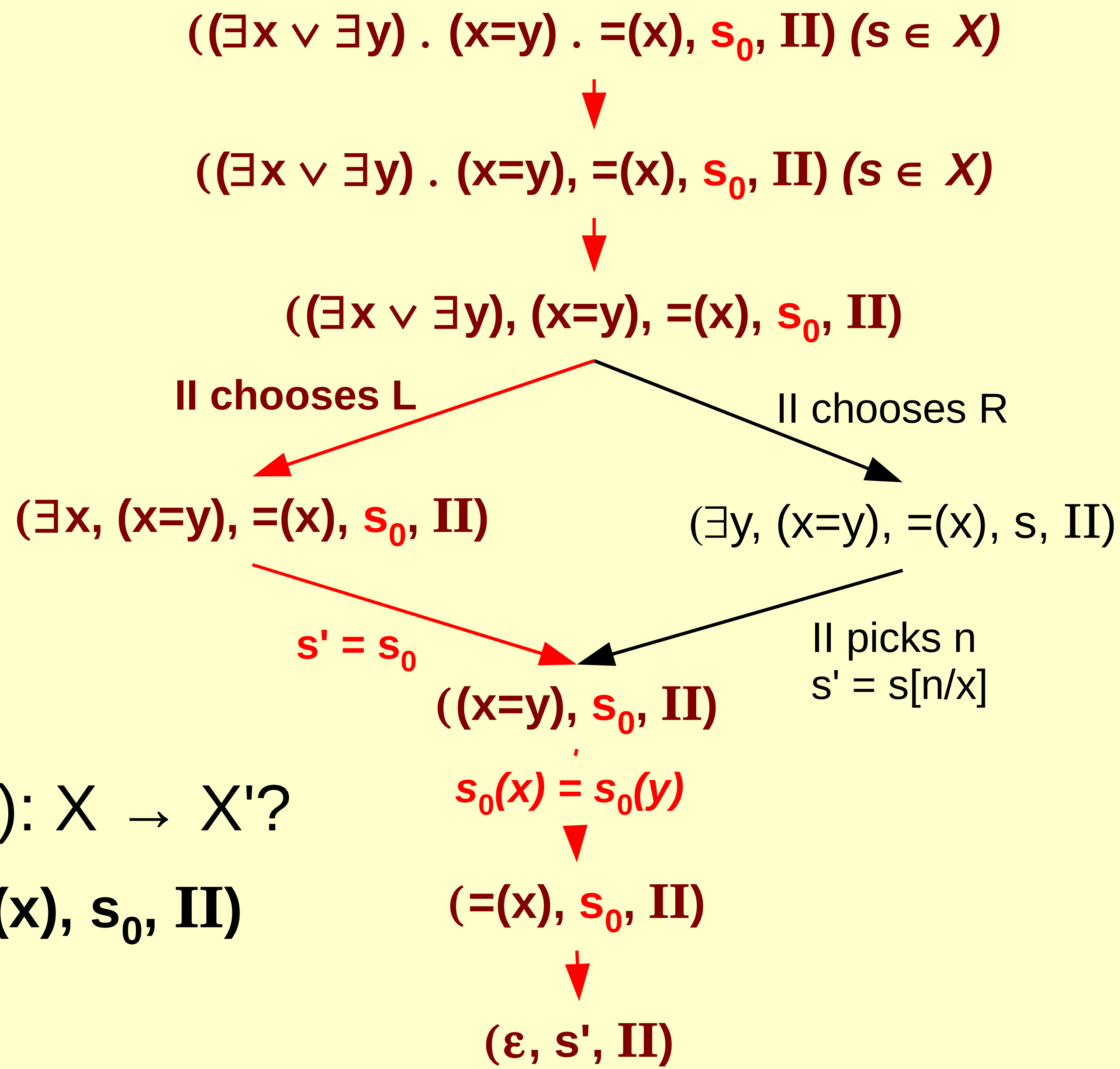
$M = (\{0,1\})$

$X =$

	x	**y**	**...**
s_0	*0*	*0*	*...*
s_1	0	1	...

$X' =$

	x	**y**	...
s'_0	0	0	...
s'_1	1	1	...



$(\exists x \vee \exists y) \,.\, (x{=}y) \,.\, {=}(x): X \rightarrow X'$?

No: not uniform! $({=}(x), s_0, \mathrm{II})$

Dynamic Dependence Logic

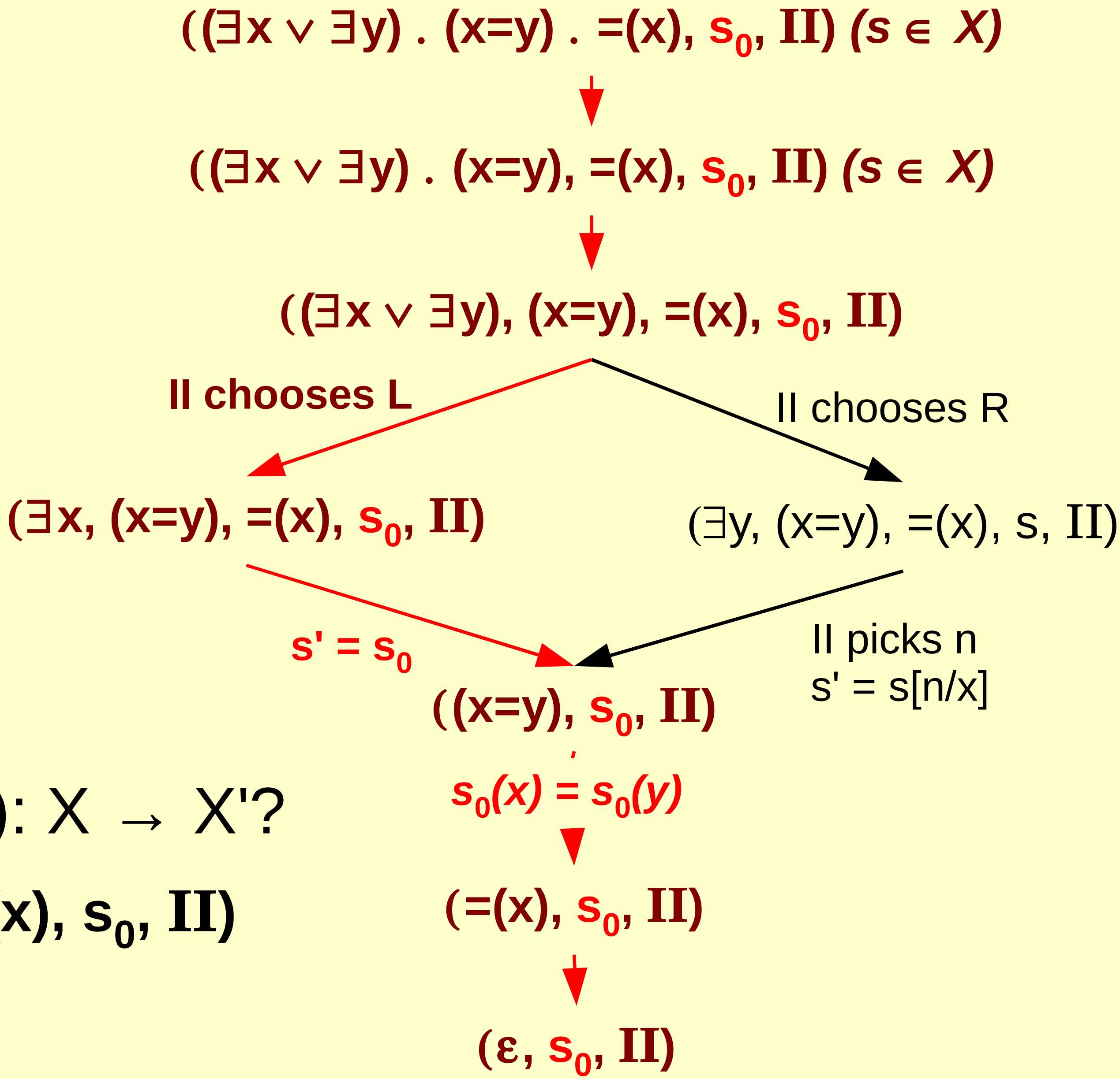
M = ({0,1})

X =

	x	y	...
s_0	0	0	...
s_1	0	1	...

X' =

	x	y	...
s'_0	0	0	...
s'_1	1	1	...



(∃x ∨ ∃y) . (x=y) . =(x): X → X'?

No: not uniform! **(=(x), s_0, II)**

Dynamic Dependence Logic

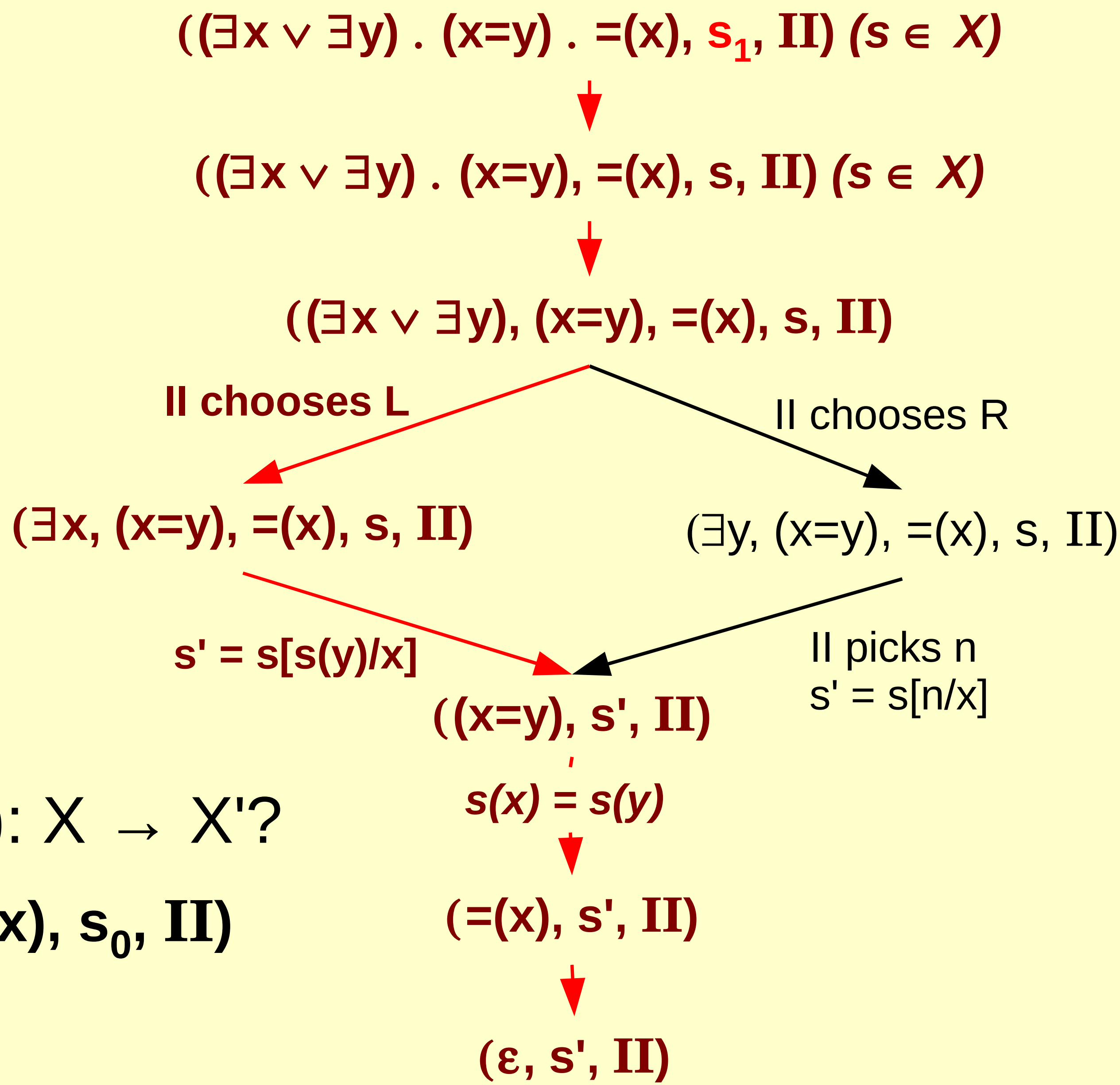
M = ({0,1})

X =

	x	y	...
s_0	0	0	...
s_1	*0*	*1*	...

X' =

	x	**y**	...
s'_0	0	0	...
s'_1	1	1	...



$(\exists x \vee \exists y) \,.\, (x=y) \,.\, =(x)\colon X \rightarrow X'$?

No: not uniform! **$(=(x), s_0, \mathrm{II})$**

Dynamic Dependence Logic

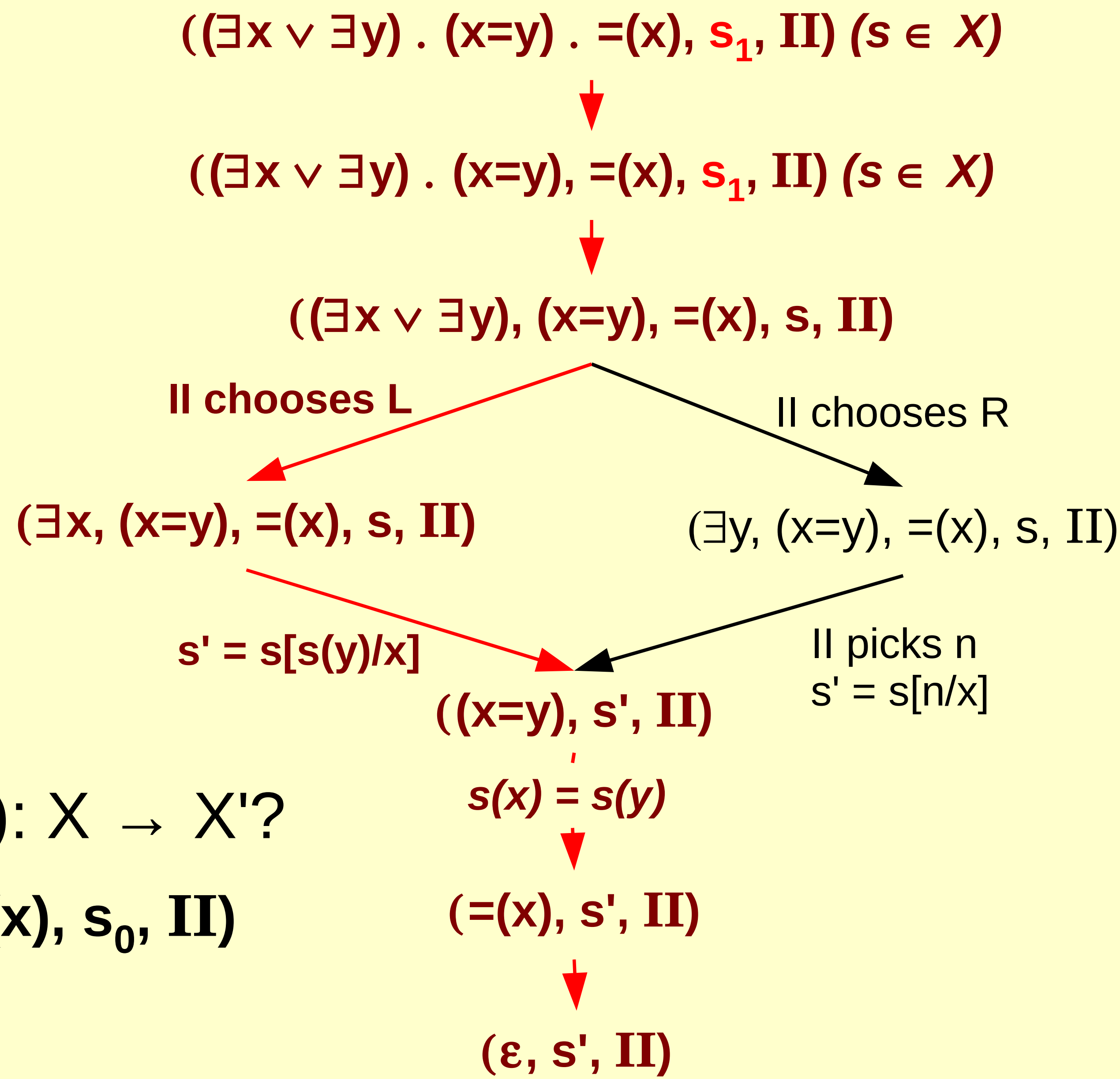
M = ({0,1})

X =

	x	y	...
s_0	0	0	...
s_1	*0*	*1*	...

X' =

	x	**y**	...
s'_0	0	0	...
s'_1	1	1	...



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

No: not uniform! **$(=(x), s_0, II)$**

Dynamic Dependence Logic

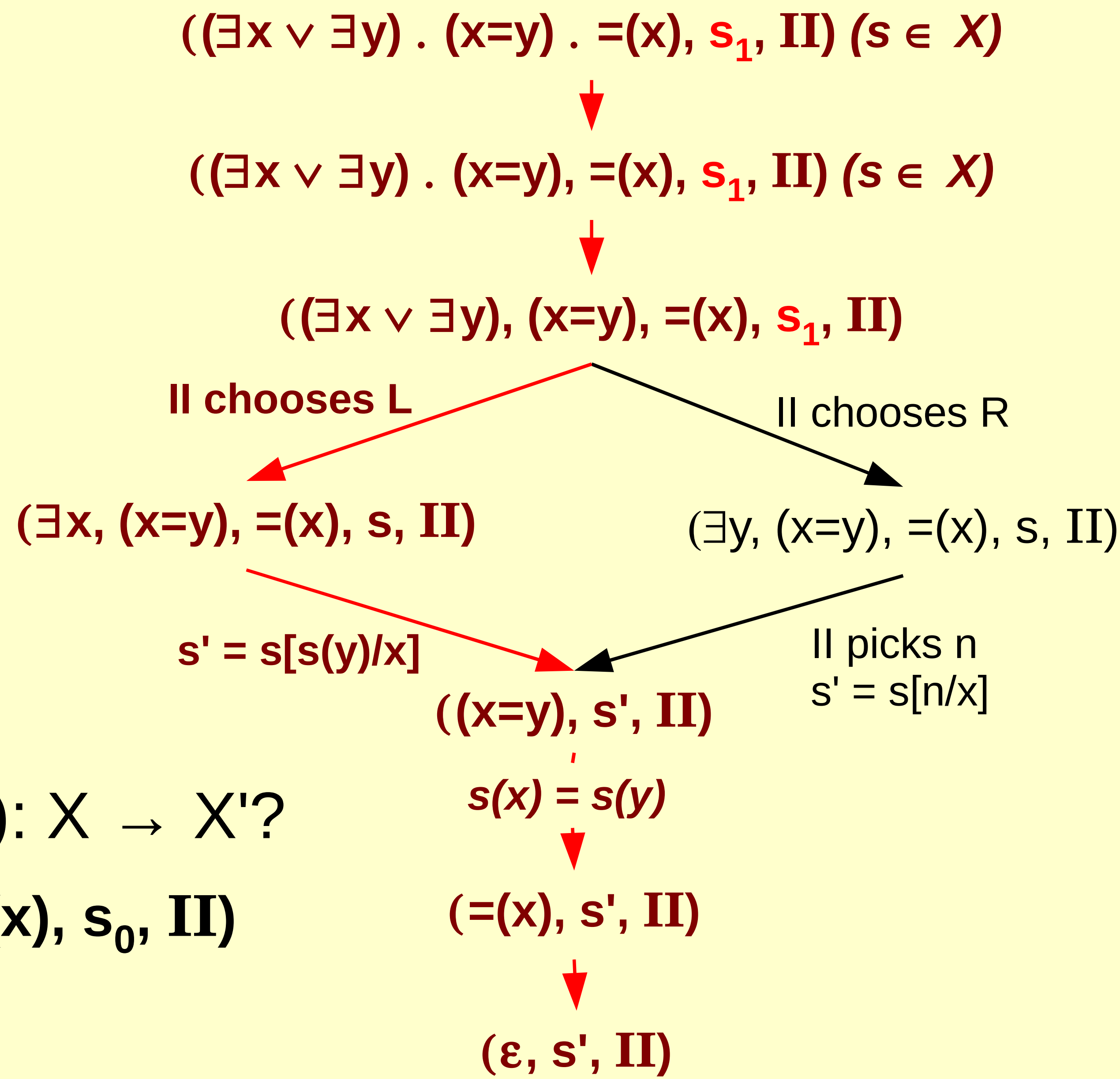
$M = (\{0,1\})$

$X =$

	x	y	...
s_0	0	0	...
$\boldsymbol{s_1}$	*0*	*1*	...

$X' =$

	x	**y**	...
s'_0	0	0	...
s'_1	1	1	...



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

No: not uniform! $\mathbf{(=(x), s_0, II)}$

Dynamic Dependence Logic

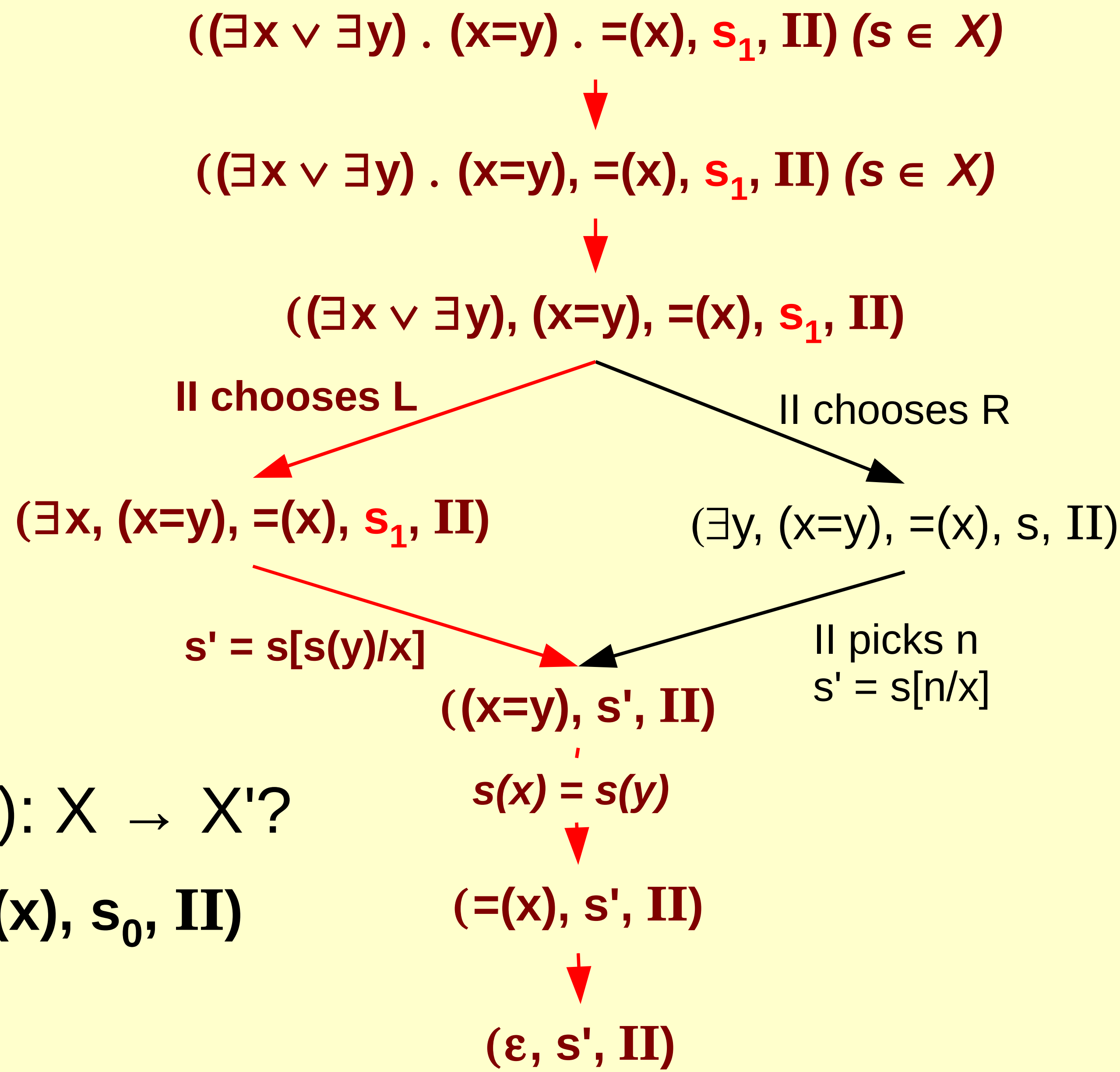
$M = (\{0,1\})$

$X =$

	x	y	...
s_0	0	0	...
$\boldsymbol{s_1}$	*0*	*1*	...

$X' =$

	x	**y**	...
s'_0	0	0	...
s'_1	1	1	...



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

No: not uniform! **$(=(x), s_0, \mathrm{II})$**

Dynamic Dependence Logic

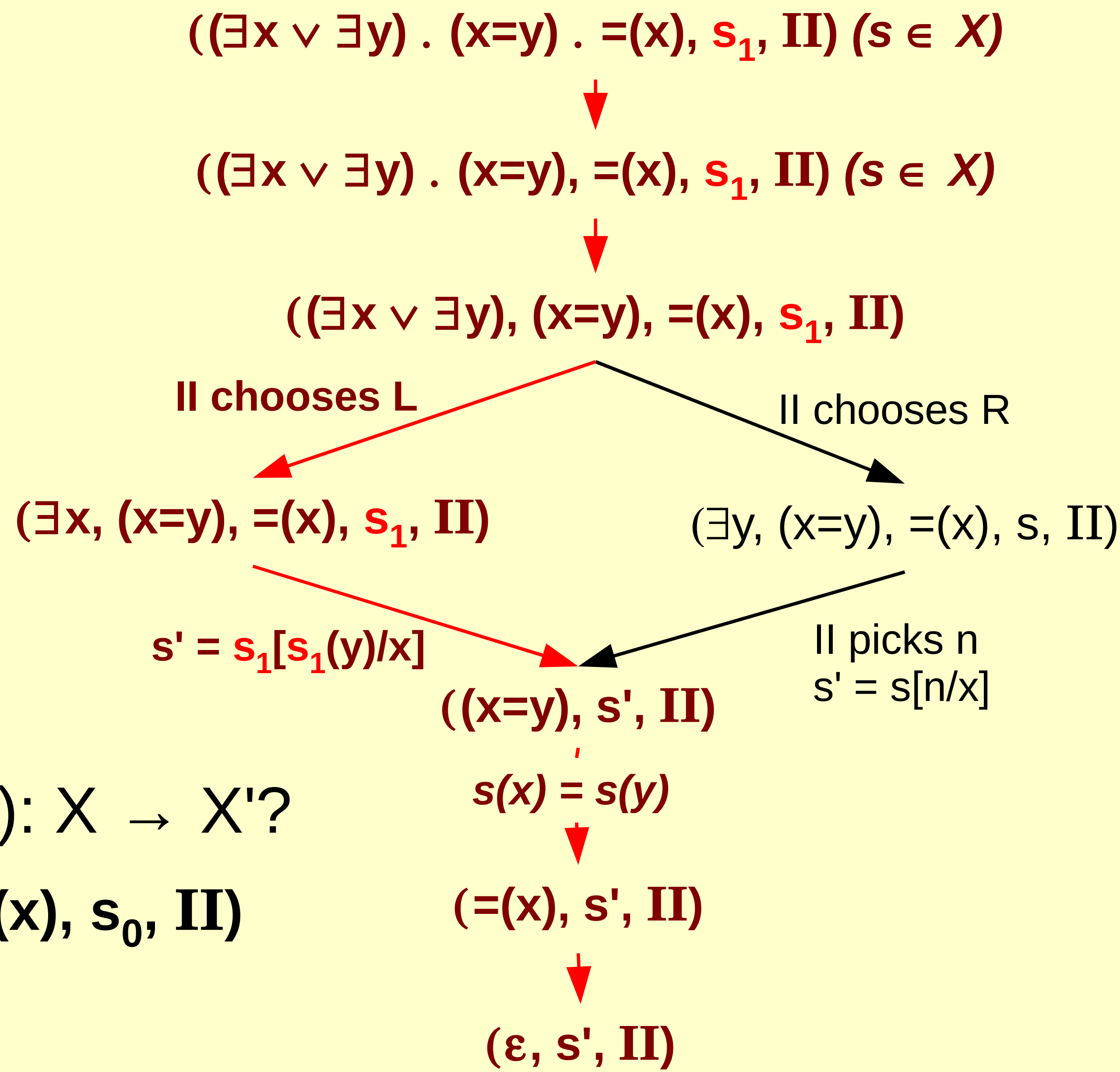
M = ({0,1})

X =

	x	y	...
s_0	0	0	...
s_1	*0*	*1*	...

X' =

	x	**y**	...
s'_0	0	0	...
s'_1	1	1	...



$(\exists x \vee \exists y) . (x{=}y) . {=}(x)\text{: } X \rightarrow X'$?

No: not uniform!

$(=(x), s_0, \mathrm{II})$

Dynamic Dependence Logic

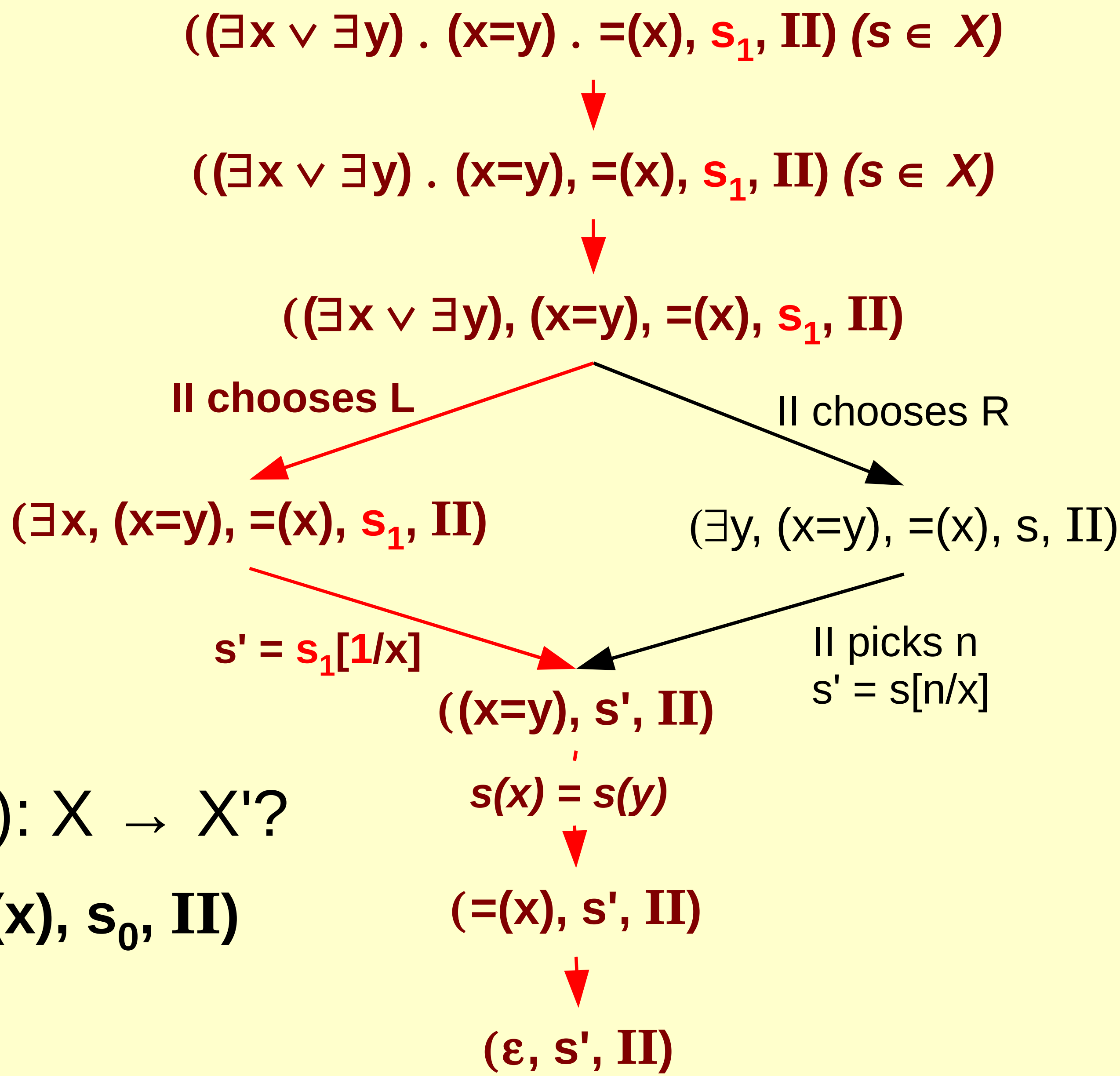
M = ({0,1})

X =

	x	y	...
s_0	0	0	...
s_1	*0*	*1*	...

X' =

	x	**y**	...
s'_0	0	0	...
s'_1	1	1	...



$(\exists x \vee \exists y) \,.\, (x{=}y) \,.\, {=}(x): X \rightarrow X'$?

No: not uniform!

$({=}(x), s_0, \text{II})$

Dynamic Dependence Logic

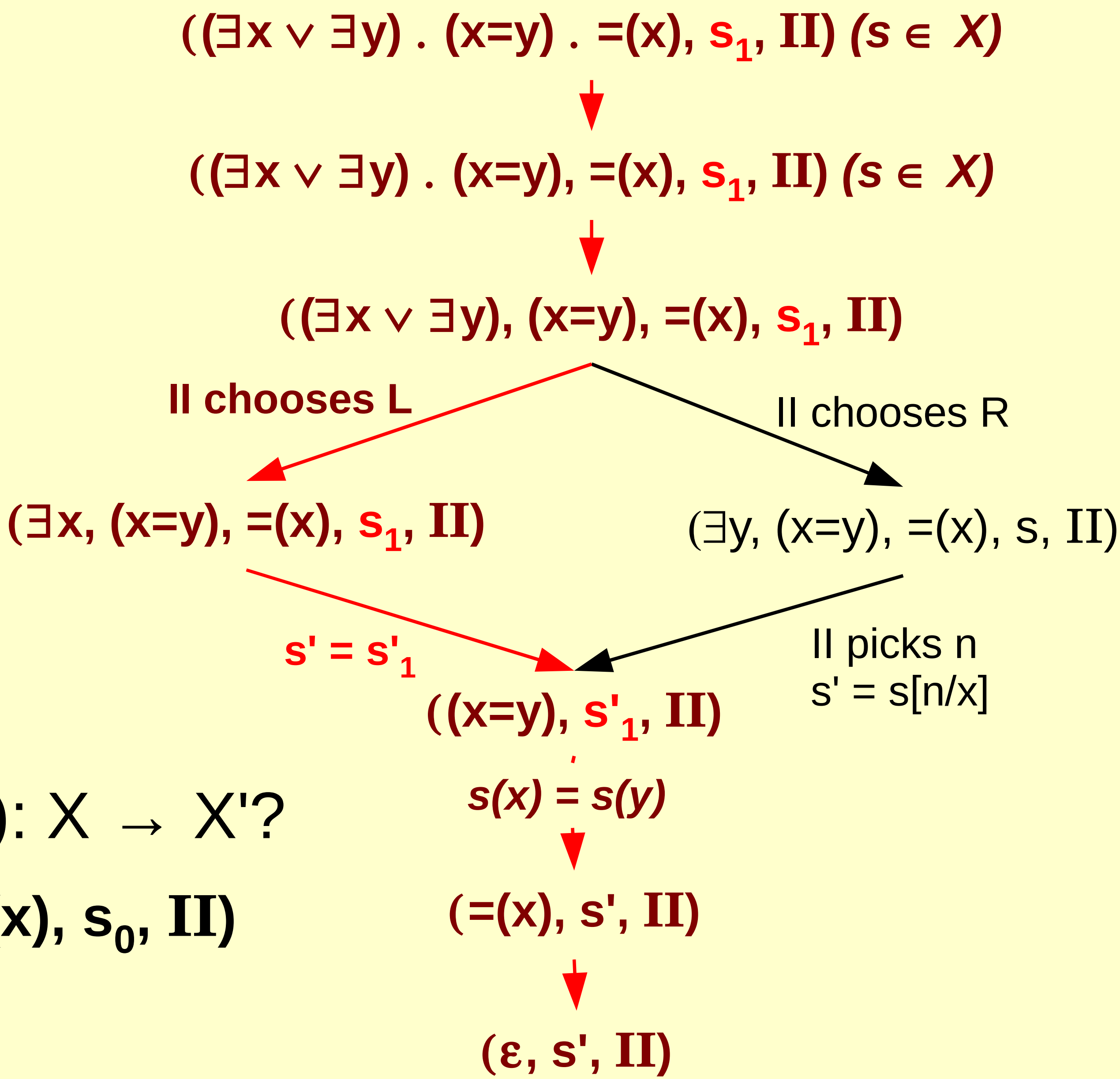
$M = (\{0,1\})$

$X =$

	x	y	...
s_0	0	0	...
s_1	0	1	...

$X' =$

	x	**y**	...
s'_0	0	0	...
s'_1	*1*	*1*	...



$(\exists x \vee \exists y) \,.\, (x{=}y) \,.\, {=}(x) : X \to X'$?

No: not uniform! **$({=}(x), s_0, \mathrm{II})$**

Dynamic Dependence Logic

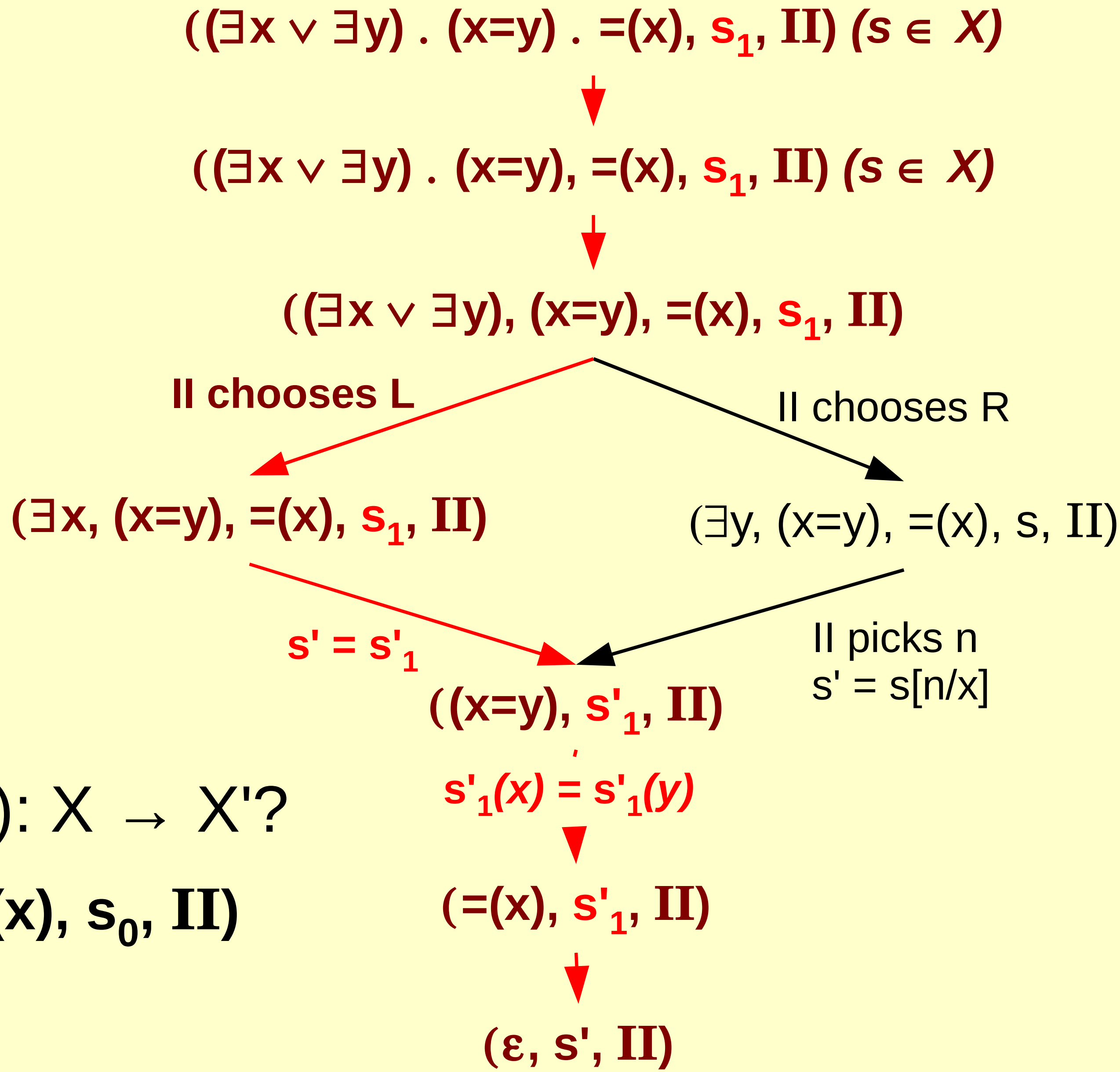
M = ({0,1})

X =

	x	y	...
s_0	0	0	...
s_1	0	1	...

X' =

	x	**y**	...
s'_0	0	0	...
s'_1	*1*	*1*	...



$(\exists x \vee \exists y)\ .\ (x{=}y)\ .\ {=}(x): X \to X'$?

No: not uniform! **$({=}(x), s_0, \mathrm{II})$**

Dynamic Dependence Logic

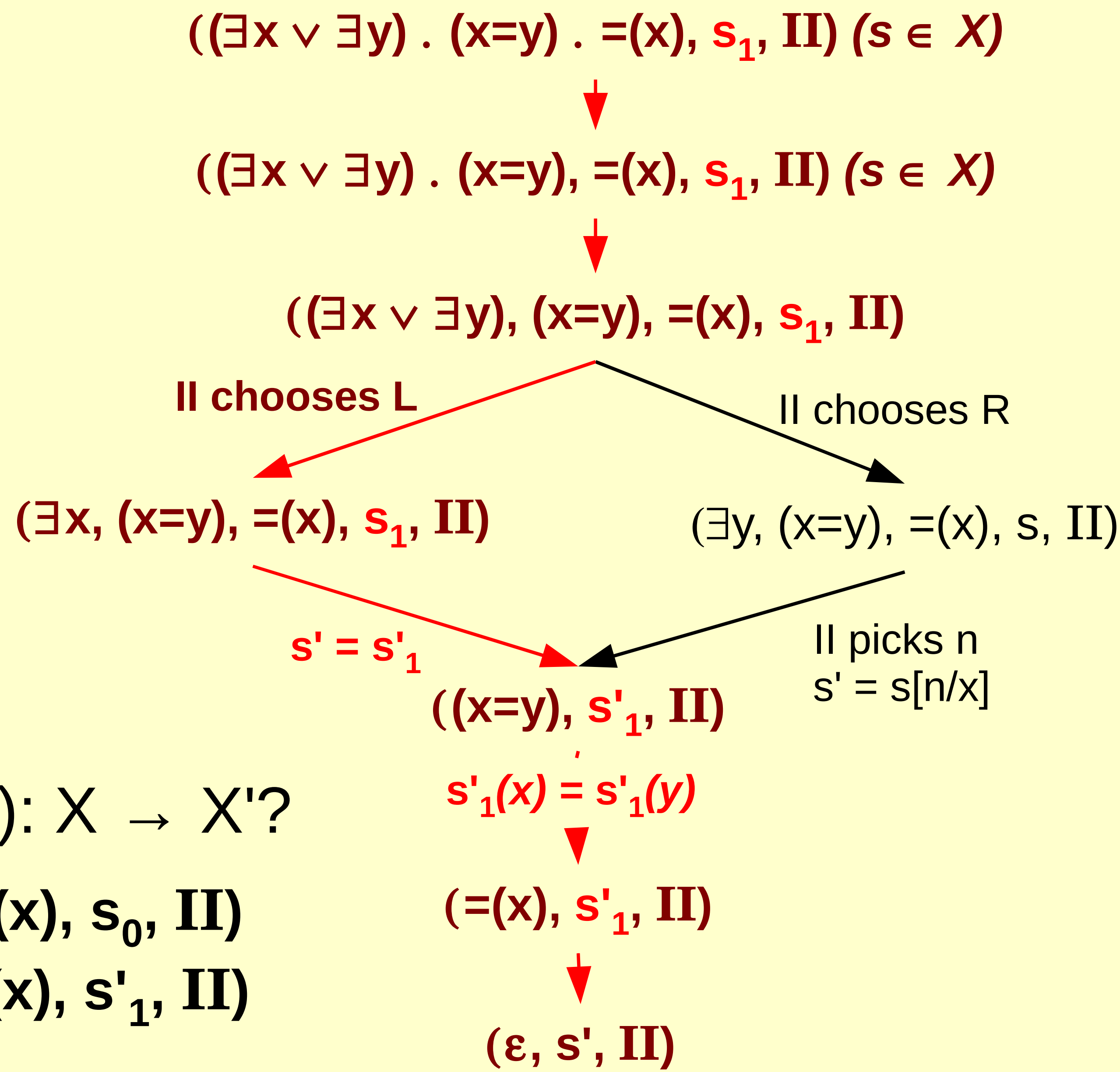
M = ({0,1})

X =

	x	y	...
s_0	0	0	...
s_1	0	1	...

X' =

	x	**y**	...
s'_0	0	0	...
s'_1	*1*	*1*	...



(∃x ∨ ∃y) . (x=y) . =(x): X → X'?

No: not uniform!

(=(x), s_0, II)

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

Dynamic Dependence Logic

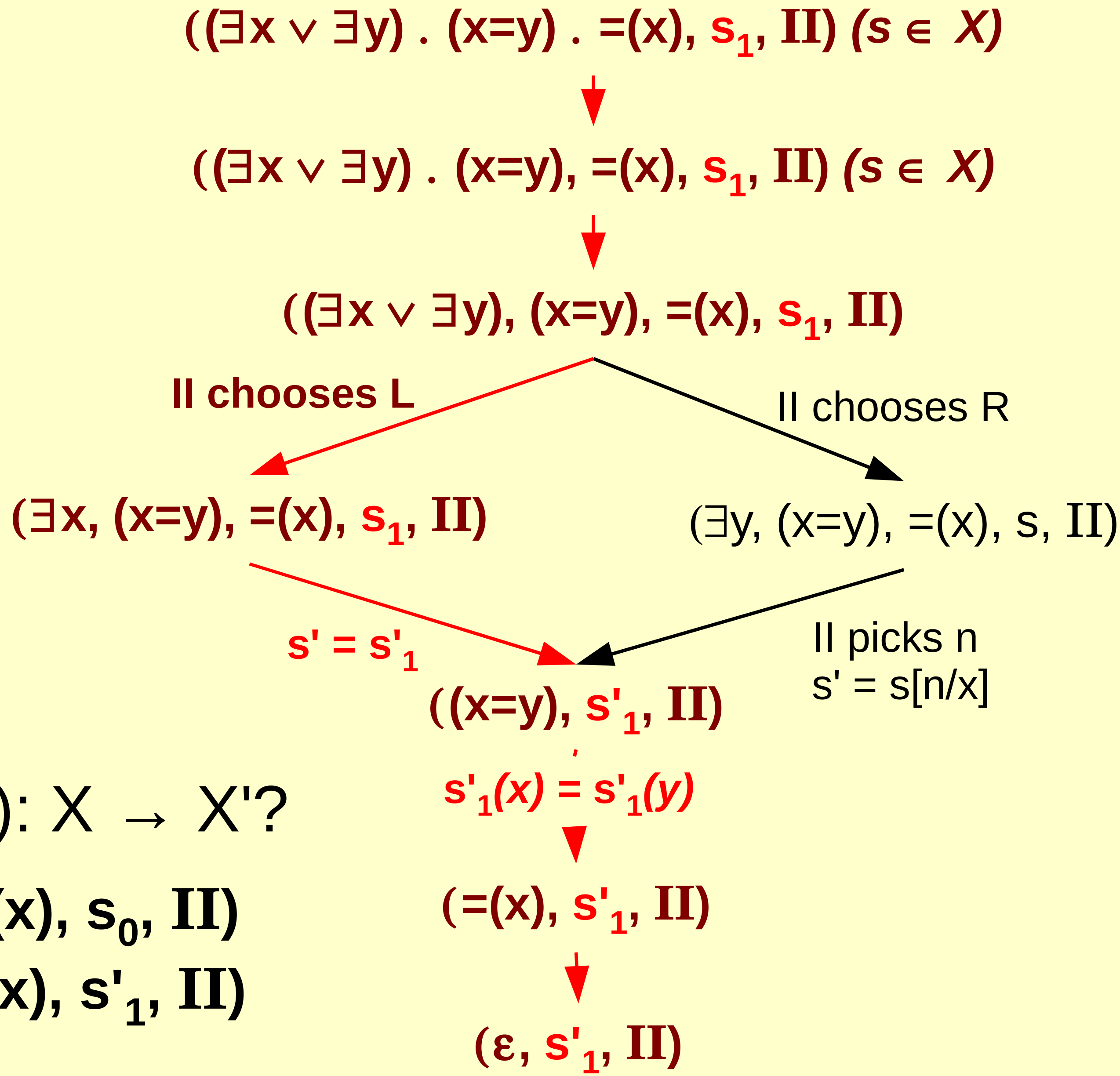
M = ({0,1})

X =

	x	y	...
s_0	0	0	...
s_1	0	1	...

X' =

	x	**y**	...
s'_0	0	0	...
s'_1	*1*	*1*	...



$(\exists x \vee \exists y) . (x{=}y) . {=}(x): X \rightarrow X'$?

No: not uniform!

(=(x), s_0, II)

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

Dynamic Dependence Logic

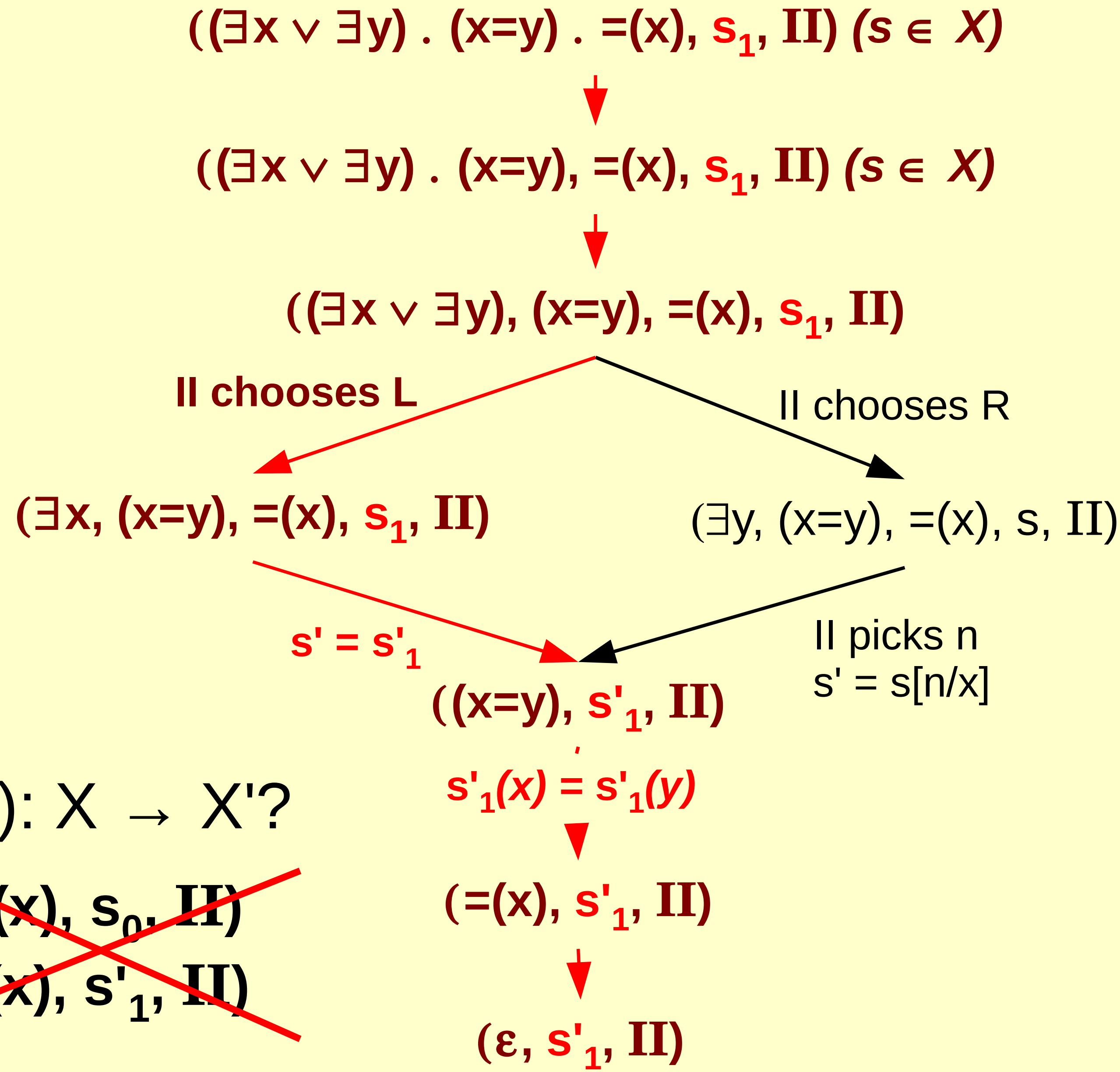
M = ({0,1})

X =

	x	y	...
s_0	0	0	...
s_1	0	1	...

X' =

	x	**y**	...
s'_0	0	0	...
s'_1	*1*	*1*	...



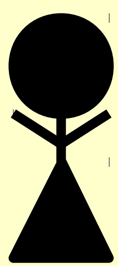
$(\exists x \vee \exists y)\ .\ (x{=}y)\ .\ {=}(x): X \rightarrow X'?$

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

~~$(=(x), s_0, II)$~~

~~$(=(x), s'_1, II)$~~

Dynamic Dependence Logic



Thank you!