Dependence Logic

Jouko Väänänen

University of Helsinki

University of Amsterdam

LOGICCC - LINT

The dependence concept

Dependence of health on genes.

Dependence of future events on past decisions.

Dependence of moves of a player on previous moves.

Arrow's Theorem

If the social welfare function respects unanimity and independence of irrelevant alternatives, it is a dictatorship.

Classical logic

there is	677
for all	399
for some	399
for every	146

Modal logic

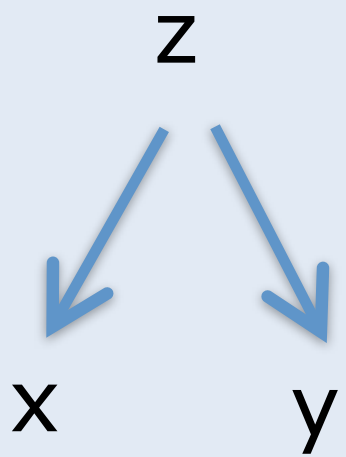
possibe	609
probably	313
likely	234
perhaps	201
it is possible that	146
possibly	118
necessarily	85
knows that	38
believes that	30
it is necessary that	23
it is obligatory that	0.1
it is permissible that	0.1

Dependence logic

is part of	720
includes	464
subject to	335
liable to	150
open to	140
dependent	88
determined by	78
given by	70
independence	64
function of	60
dependent on	43
dependence	32
independent of	30
belonging to	26
modified by	20
dependency	19
dependence on	13
vulnerable to	11
independence of	7
computed from	4
totally dependent on	3
uniquely determined by	3
contingent on	3
qualified by	2
totally independent of	1.6
conditioned by	1.5
left open by	1.3
mutually dependent	0.2
totally determined by	0.1
mutual dependency	0.06
mutually dependent on e/o	0.02


x
y


x
y


z
x
y

Question

Can one add the *dependence* concept to
first order logic (or other logics) in a coherent way?

What is the *logic* of dependence?

Solution

- We consider the strongest form of dependence, namely functional determination $z = f(x_1,...,x_n)$, where $x_1,...,x_n,z$ are individual variables.
- We denote it $=(x_1,...,x_n,z)$ and call it a dependence atom. Weaker forms of dependence are derived from this.
- In computer science: $x_1...x_n \Rightarrow z$, where $x_1,...,x_n,z$ are

Solution

- We consider the strongest form of dependence, namely functional determination $z = f(x_1,\ldots,x_n)$, where $x_1,\ldots,x_n,z$ are individual variables.
- We denote it $=(x_1,\ldots,x_n,z)$ and call it a dependence atom. Weaker forms of dependence are derived from this.
- In computer science: $x_1\ldots x_n \Rightarrow z$, where $x_1,\ldots,x_n,z$ are

Solution

- We consider the strongest form of dependence, namely functional determination $z = f(x_1,\ldots,x_n)$, where $x_1,\ldots,x_n,z$ are individual variables.
- We denote it $=(x_1,\ldots,x_n,z)$ and call it a dependence atom. Weaker forms of dependence are derived from this.
- In computer science: $x_1\ldots x_n \Rightarrow z$, where $x_1,\ldots,x_n,z$ are

	Name	Job	Gender	Salary group
s_0	Jeff	analyst	M	C
s_1	Paula	assistant	F	A
s_2	Laurie	assistant	M	C

Multitude

- Dependence does not manifest itself in a **single** play, event or observation.
- The underlying concept of dependence logic is a multitude – a **collection** - of such plays, events or observations.
- These collections are called in this talk **teams**.
- They are the basic objects of our approach.

Multitude

- Dependence does not manifest itself in a **single** play, event or observation.
- The underlying concept of dependence logic is a multitude – a **collection -** of such plays, events or observations.
- These collections are called in this talk **teams**.
- They are the basic objects of our approach.

Multitude

- Dependence does not manifest itself in a **single** play, event or observation.
- The underlying concept of dependence logic is a multitude – a **collection -** of such plays, events or observations.
- These collections are called in this talk **teams**.
- They are the basic objects of our approach.

Multitude

- Dependence does not manifest itself in a **single** play, event or observation.
- The underlying concept of dependence logic is a multitude – a **collection -** of such plays, events or observations.
- These collections are called in this talk **teams**.
- They are the basic objects of our approach.

Teams

- A set of records of stock exchange transactions of a particular dealer.
- A set of possible histories of mankind written as decisions and consequences.
- A set of chess games between Susan and Max, as lists of moves.

Teams

- 1st intuition: A team is a set of plays of a game.

Teams

- 1st intuition: A team is a set of plays of a game.
- 2nd intuition: A team is a database.

	x_0	x_1	x_2
s_0	0	1	0
s_1	0	1	1
s_2	2	5	5

Towards a logic based on teams

- A set of plays satisfies $x_2 > x_0$ if move x_2 is in each play greater than move x_0.
- A set of plays satisfies $=(x_1,\ldots,x_n,y)$ if move y is in each play determined by the moves $x_1,\ldots,x_n$.
- A database satisfies $x_2 > x_0$ if field x_2 is always greater than field x_0.
- A database satisfies $=(x_1,\ldots,x_n,y)$ if field y is functionally determined by the fields $x_1,\ldots,x_n$.

Towards a logic based on teams

- A set of plays satisfies $x_2 > x_0$ if move x_2 is in each play greater than move x_0.
- A set of plays satisfies $=(x_1,\ldots,x_n,y)$ if move y is in each play determined by the moves $x_1,\ldots,x_n$.
- A database satisfies $x_2 > x_0$ if field x_2 is always greater than field x_0.
- A database satisfies $=(x_1,\ldots,x_n,y)$ if field y is functionally determined by the fields $x_1,\ldots,x_n$.

Dependence atoms $=(x_1,\ldots,x_n,z)$

+

First order logic

=

Dependence logic

Syntax of dependence logic

$=, \neg, \vee, \wedge, \exists, \forall,), (, x_i$

x_i, c, $ft_1 \ldots t_n$

$t=t'$

$=(x_1, \ldots, x_n, z)$

$Rt_1 \ldots t_n$

$t=t'$

$Rt_1 \ldots t_n$

$\neg \varphi$

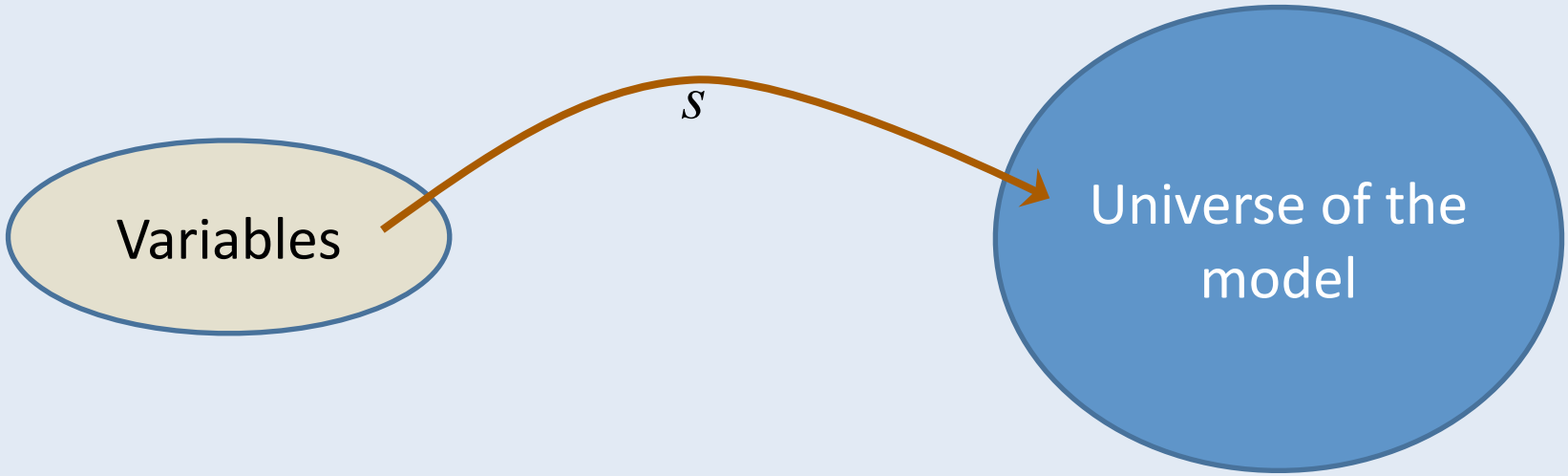
$\varphi \vee \psi$

$\varphi \wedge \psi$

$\exists x_i \varphi$

$\forall x_i \varphi$

Assignment



Teams – exact definition

- A team is just a **set** of assignments for a model.
  (Propositional logic – a set of valuations. Modal logic – a set of possible worlds)
- Empty team $\emptyset$.
  - Database with no rows.
  - No play was played.
- The team $\{\emptyset\}$ with the empty assignment.
  - Database with no columns, and hence with at most one row.
  - Zero moves of the game were played.

Teams – exact definition

- A team is just a **set** of assignments for a model.
  (Propositional logic – a set of valuations. Modal logic – a set of possible worlds)
- Empty team ∅.
  - Database with no rows.
  - No play was played.
- The team {∅} with the empty assignment.
  - Database with no columns, and hence with at most one row.
  - Zero moves of the game were played.

Teams – exact definition

- A team is just a **set** of assignments for a model.
  (Propositional logic – a set of valuations. Modal logic – a set of possible worlds)
- Empty team $\emptyset$.
  - Database with no rows.
  - No play was played.
- The team $\{\emptyset\}$ with the empty assignment.
  - Database with no columns, and hence with at most one row.
  - Zero moves of the game were played.

For the truth definition: Negation Normal Form

We push negations all the way

to atomic formulas using de Morgan laws.

Thus $\neg\neg\varphi$ will have the same meaning as φ.

Truth definition

A team **satisfies a formula** if

every assignment in the team does,

and …

A team satisfies $Rt_1 \ldots t_n$ if every team member does.

	x_0	x_1	x_2
s_0	0	1	0
s_1	0	1	1
s_2	2	5	5

$x_0 < x_1$

A team satisfies $\neg Rt_1 \ldots t_n$ if every team member does.

	x_0	x_1	x_2
s_0	0	1	0
s_1	0	1	1
s_2	2	5	5

$$\neg x_1 < x_0$$

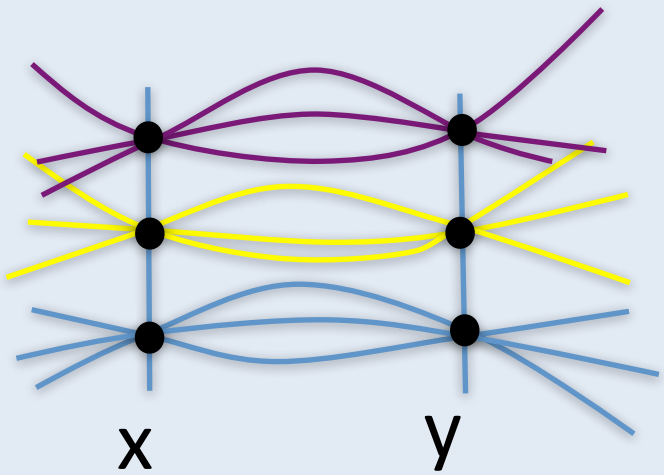
A team satisfies $\neg Rt_1 \ldots t_n$ if every team member does.

	x_0	x_1	x_2
s_0	0	1	0
s_1	0	1	1
s_2	2	5	5

$\neg x_1 < x_0$

Note: some X satisfy neither $Rt_1 \ldots t_n$ nor $\neg Rt_1 \ldots t_n$.

A team satisfies t=t’ if every team member does.



x=y

	x_0	x_1	x_2
s_0	1	0	0
s_1	0	1	1
s_2	2	5	5

$x_1=x_2$

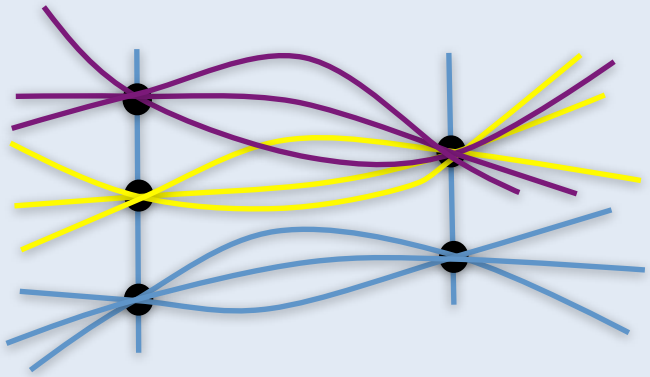
A team satisfies $\neg t=t'$ if every team member does.

	x_0	x_1	x_2
s_0	1	0	0
s_1	0	1	1
s_2	2	5	5

$$\neg x_0 = x_1$$

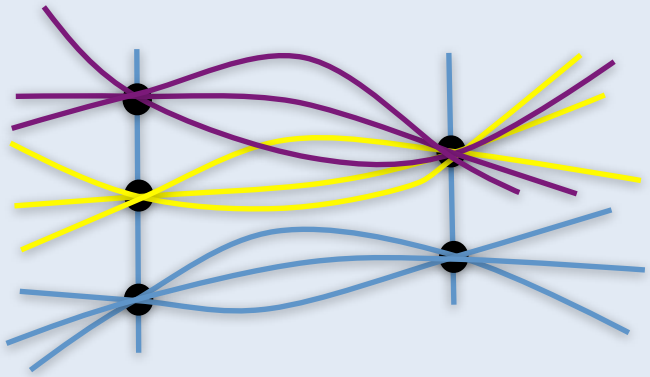
- A team X satisfies $=(x_1,\ldots,x_n,z)$ if in any two assignments in X, in which $x_1,\ldots,x_n$ have the same values, also z has the same value.

- A team X satisfies $=(x_1,\ldots,x_n,z)$ if in any two assignments in X, in which $x_1,\ldots,x_n$ have the same values, also z has the same value.



$=(x,y)$

- A team X satisfies $=(x_1,\ldots,x_n,z)$ if in any two assignments in X, in which $x_1,\ldots,x_n$ have the same values, also z has the same value.



$=(x,y)$

	x	y	u	z
s_0	0	0	1	0
s_1	0	1	0	2
s_2	2	5	0	5
s_3	0	1	1	2

$=(x,y,z)$

An extreme case

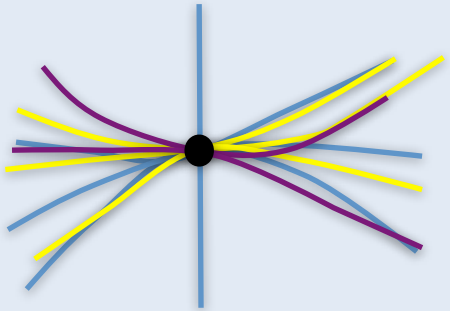
=(x)

”x is constant in the team”

An extreme case

=(x)

”x is constant in the team”



x

=(x)

record	A1	A2	A3	A4	A5	A6
100000	8	6	7	3	0	6
100002	7	5	6	3	0	6
100003	4	8	7	3	0	6
100004	6	5	4	3	0	6
100005	6	12	65	3	0	6
100006	5	56	9	3	0	6
100007	6	23	0	4	0	8
...	...	...	...	...	...	...
408261	77	2	11	1	0	2

Negation of dependence atom

A team satisfies $\neg =(x_1,\ldots,x_n,z)$ only if it is empty.

Why?

Negation of dependence atom

A team satisfies $\neg=(x_1,\ldots,x_n,z)$ only if it is empty.

Why?

Because every *singleton* team satisfies $=(x_1,\ldots,x_n,z)$,

and we want downward closure (see later).

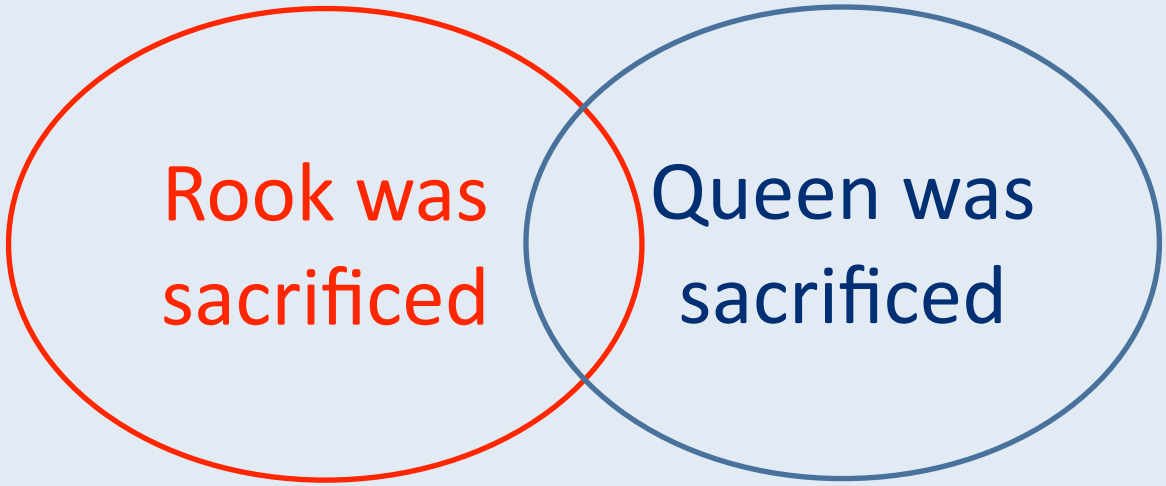
- A team X satisfies $\varphi \vee \psi$ if

  $X = Y \cup Z$, where Y satisfies φ and Z satisfies ψ.

- A team X satisfies $\varphi \vee \psi$ if

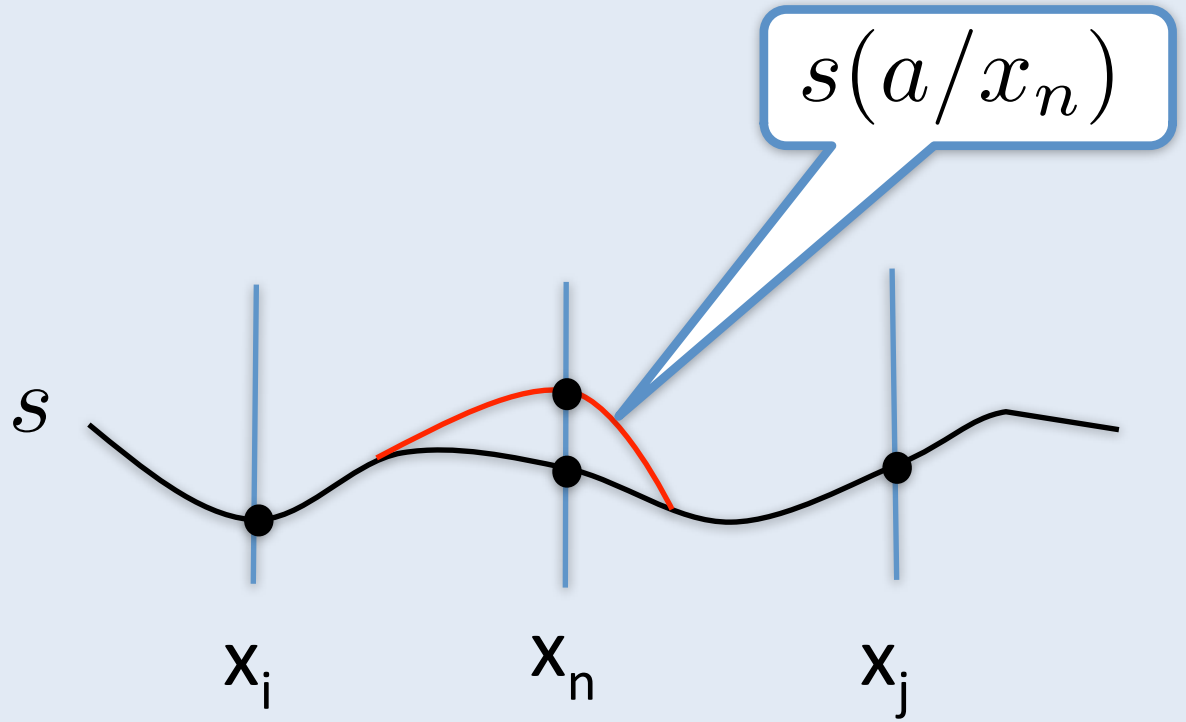
  $X = Y \cup Z$, where Y satisfies φ and Z satisfies ψ.

Plays where rook **or** queen was sacrificed:



- A team X satisfies $\varphi \wedge \psi$ if it satisfies φ and ψ.

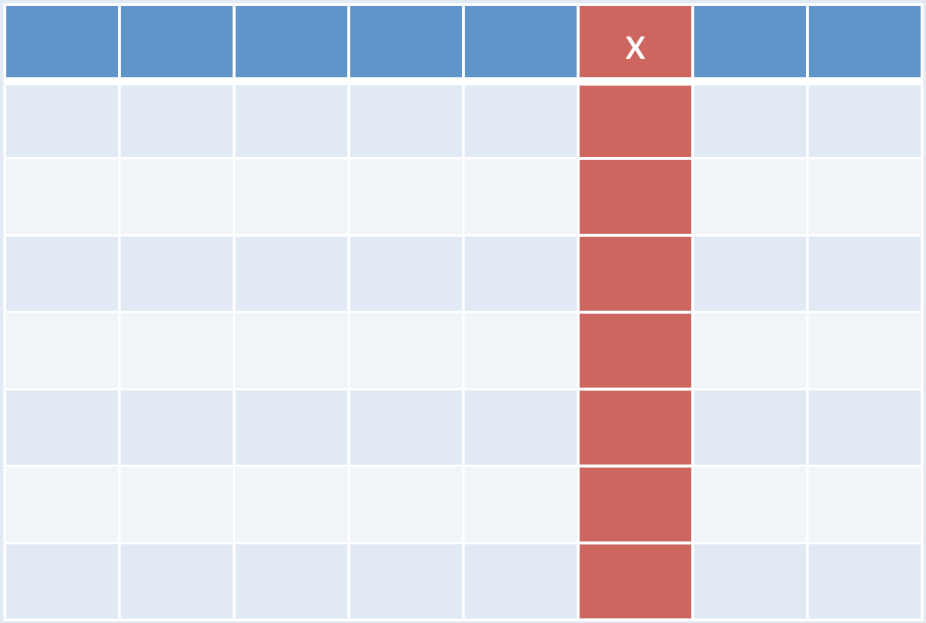
Quantifiers - modified assignment



- A team X satisfies $\exists x\varphi$ if

  team X can be supplemented with values for x so that φ is satisfied.

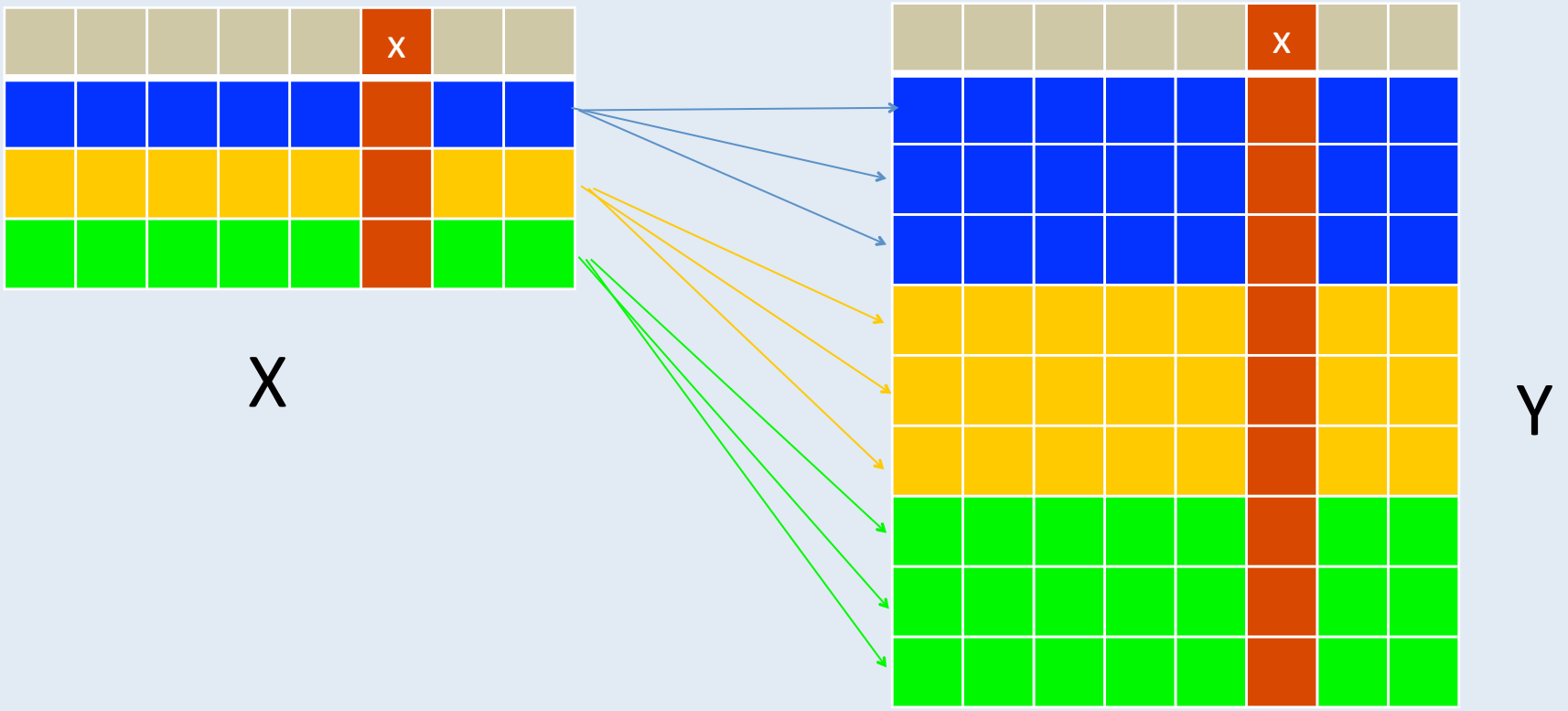
Team X is supplemented with values for x.

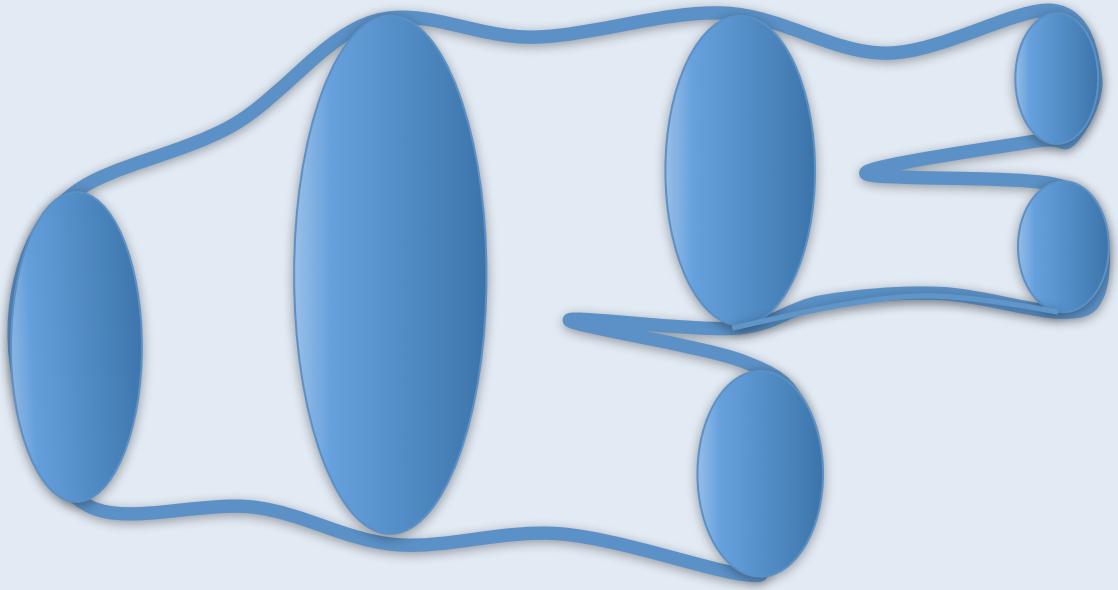


- A team *X* satisfies $\forall x\varphi$ if

  team X, after it is duplicated along x, by letting x get all possible values, satisfies φ.

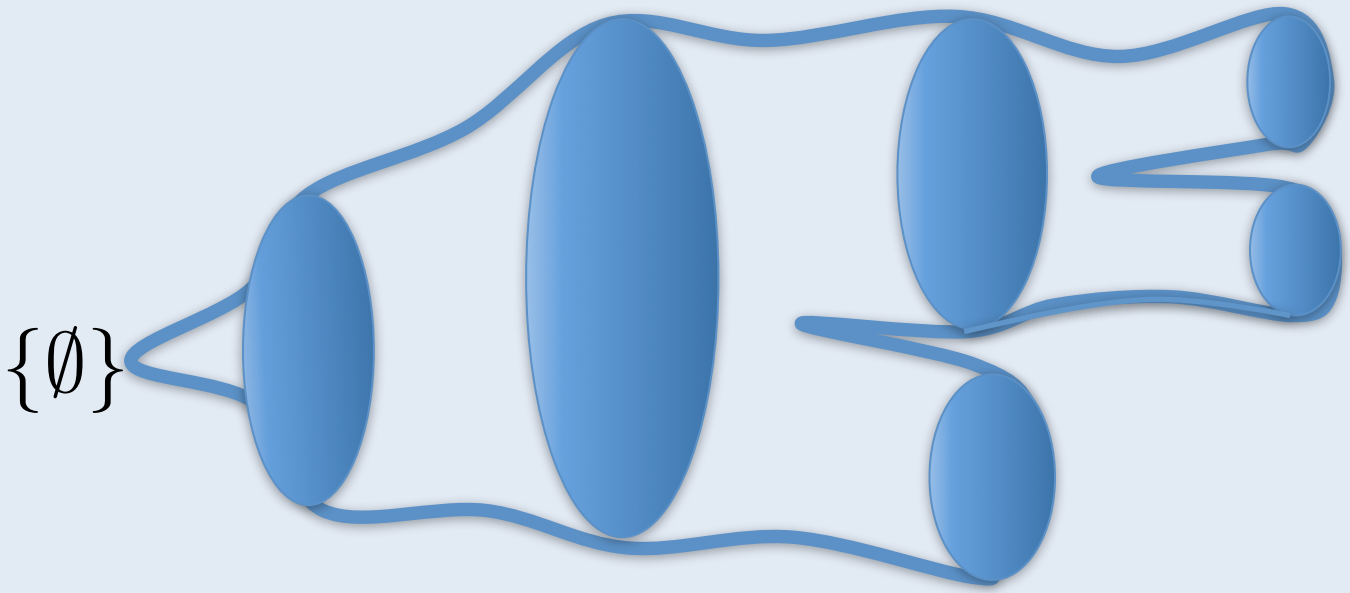
Team X is duplicated along x, by letting x get all possible values.





Truth

- A sentence is **true** if $\{\emptyset\}$ satisfies it.



Example: even cardinality

$$\forall x_0 \exists x_1 \forall x_2 \exists x_3 (=(x_2, x_3) \ \wedge \ \neg(x_0 = x_1) \ \wedge \ (x_0 = x_2 \rightarrow x_1 = x_3) \ \wedge \ (x_1 = x_2 \rightarrow x_3 = x_0))$$

Equicardinality

$$\forall x_0 \exists y_0 \forall x_1 \exists y_1 (\ =(x_1, y_1) \wedge (x_0 = x_1 \leftrightarrow y_0 = y_1))$$

Partially ordered quantifiers

$$\begin{pmatrix} \forall x & \exists y \\ \forall u & \exists v \end{pmatrix} \phi \iff \forall x \exists y \forall u \exists v (=(u,v) \wedge \phi)$$

Conservative over FO

A team $\{s\}$ satisfies a **first order formula** φ

iff

s satisfies φ in the usual sense.

Two important properties

Downward closure: If a team satisfies a formula, every subset does. (Hodges: optimal on finite structures!)

Empty team property: The empty team satisfies every formula.

No Law of Excluded Middle

Suppose the universe has at least two elements.

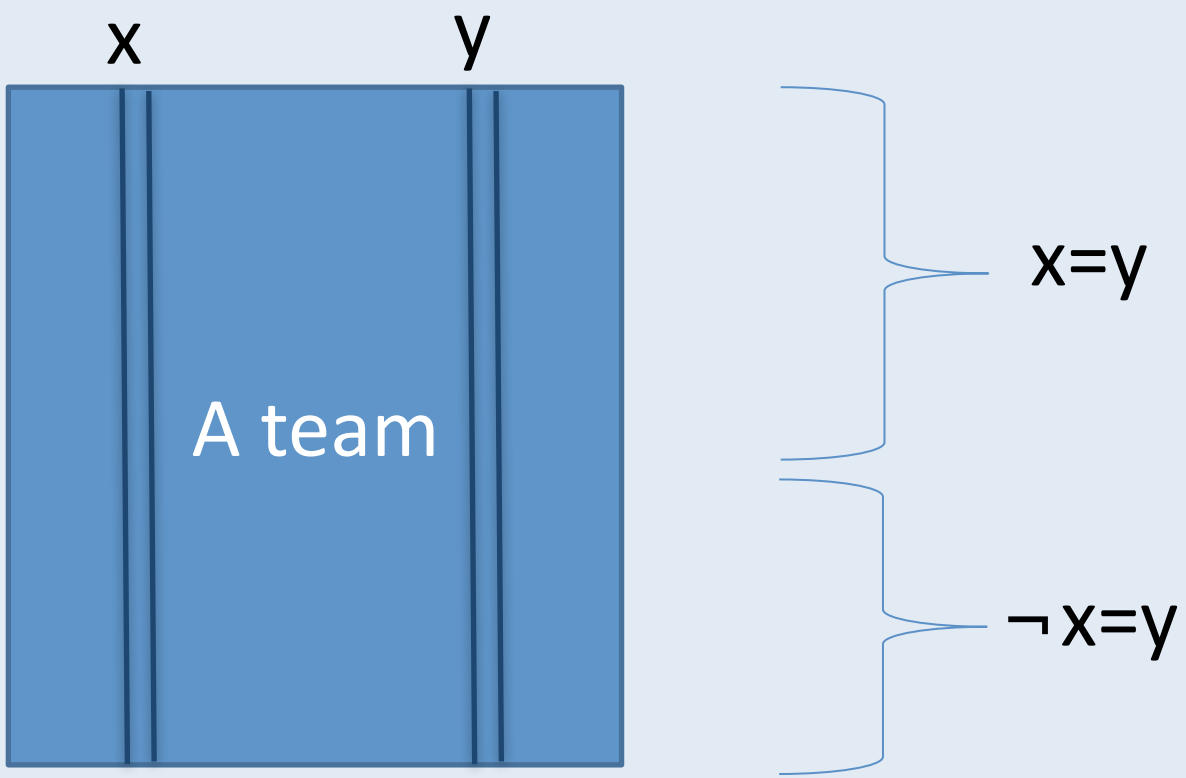
$\forall x =(x)$ not true

$\neg\forall x =(x)$ not true either

because it means $\exists x \neg=(x)$.

LEM holds (exactly) for the FO part

- Every team satisfies $x=y \vee \neg x=y$:



A special axiom schema

- **Comprehension Axioms:**

$$\forall x(\varphi \vee \neg\varphi),$$

if φ is FO.

A special axiom schema

- **Comprehension Axioms:**

$$\forall x(\varphi \vee \neg \varphi),$$

if φ is FO.

``LEM = Comprehension Axiom”

Armstrong's Axioms

Always =(x,x)

If =(x,y,z), then =(y,x,z).

If =(x,x,y), then =(x,y).

If =(x,z), then =(x,y,z).

If =(x,y) and =(y,z), then =(x,z).

Incorrect rules

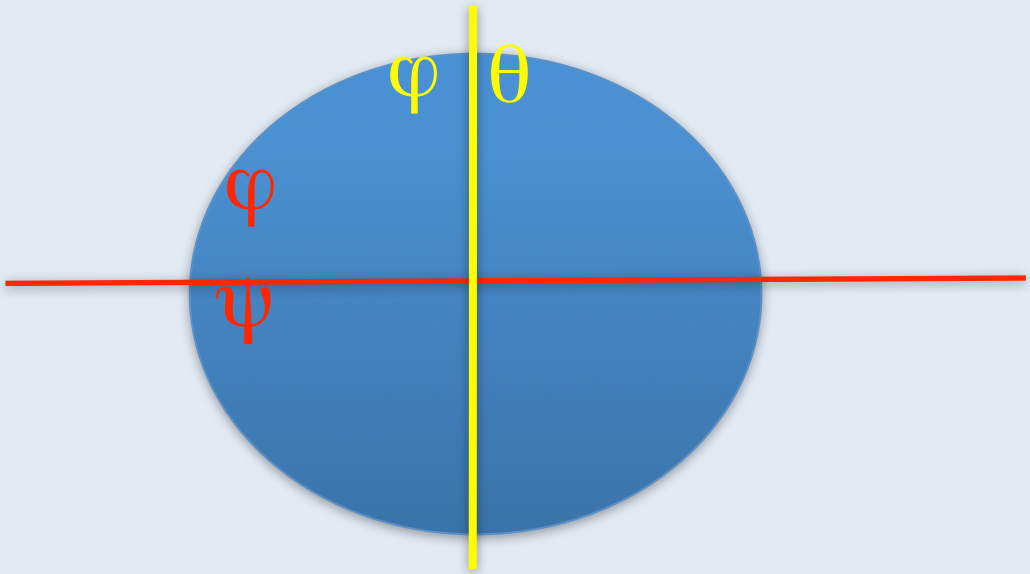
No absorption

- From $\varphi \vee \varphi$ follows φ. Wrong!
- From $(\varphi \wedge \psi) \vee (\varphi \wedge \theta)$ follows $\varphi \wedge (\psi \vee \theta)$. Wrong!
- From $(\varphi \vee \psi) \wedge (\varphi \vee \theta)$ follows $\varphi \vee (\psi \wedge \theta)$. Wrong!

Non-distributive

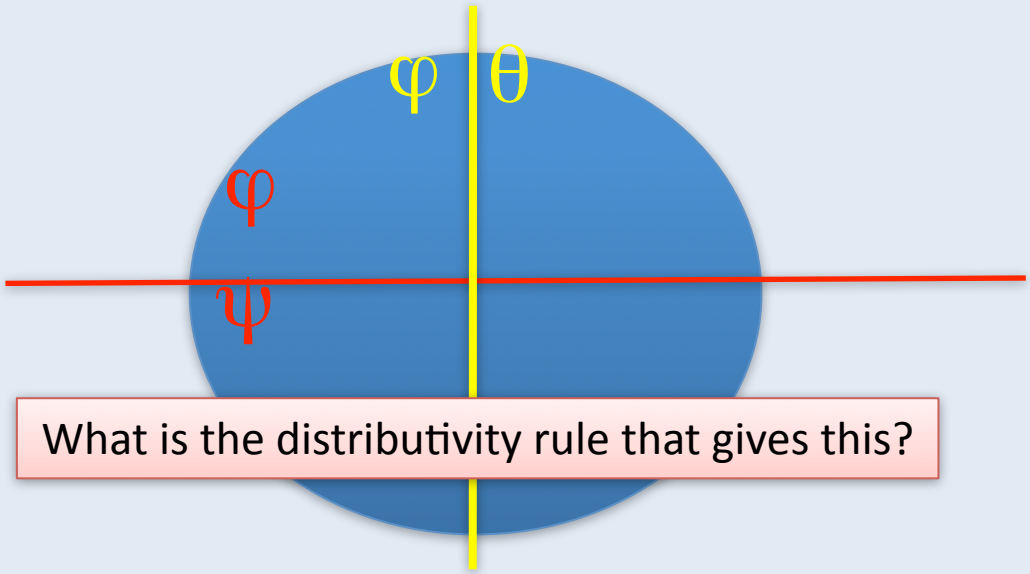
Correct intermediate rule

- From $(\varphi \vee \psi) \wedge (\varphi \vee \theta)$ follows $\varphi \vee (\psi \wedge \varphi) \vee (\psi \wedge \theta)$.



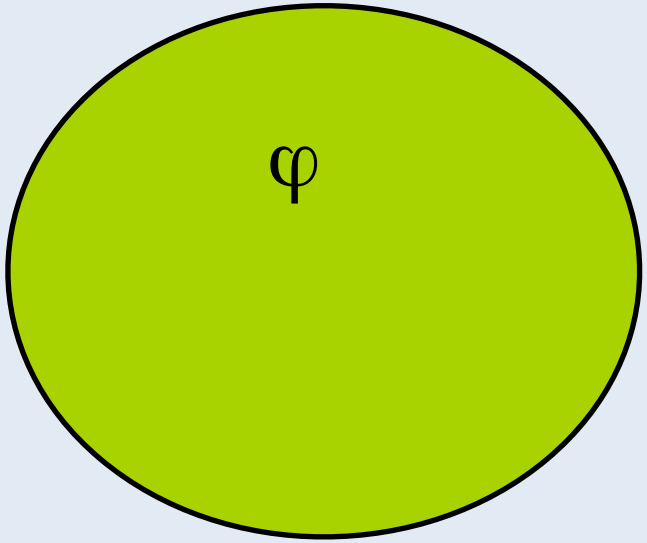
Correct intermediate rule

- From $(\varphi \vee \psi) \wedge (\varphi \vee \theta)$ follows $\varphi \vee (\psi \wedge \varphi) \vee (\psi \wedge \theta)$.



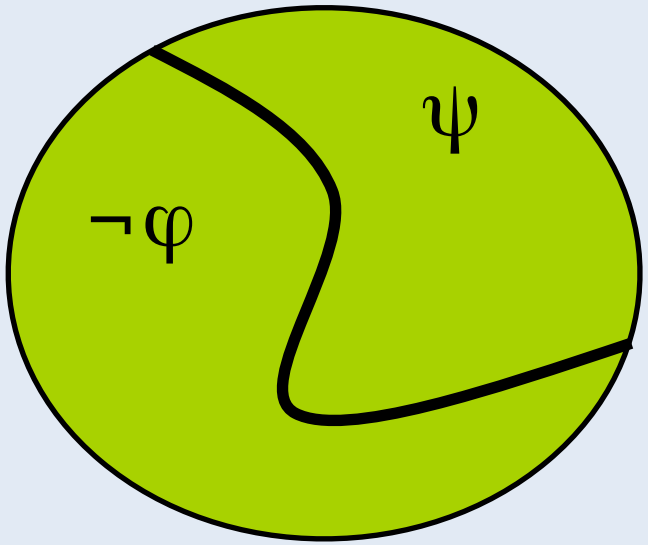
Example

- If $\neg\varphi \vee \psi$ is valid then φ *logically implies* ψ.



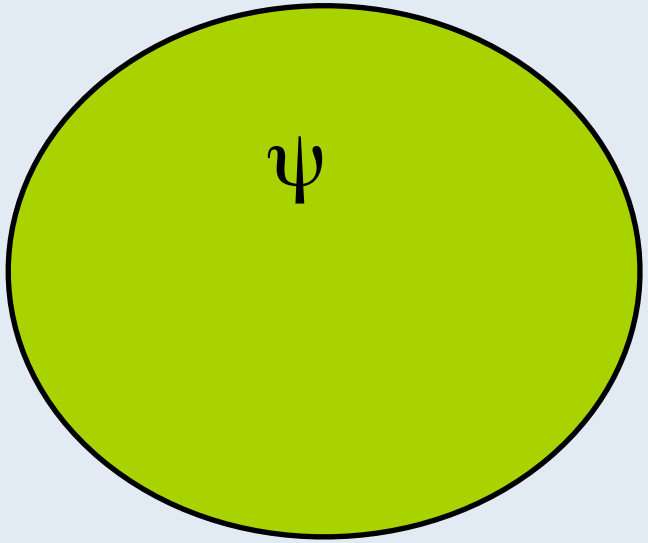
Example

- If $\neg\varphi \vee \psi$ is valid then φ *logically implies* ψ.



Example

- If $\neg\varphi \vee \psi$ is valid then φ *logically implies* ψ.



Game theoretic semantics

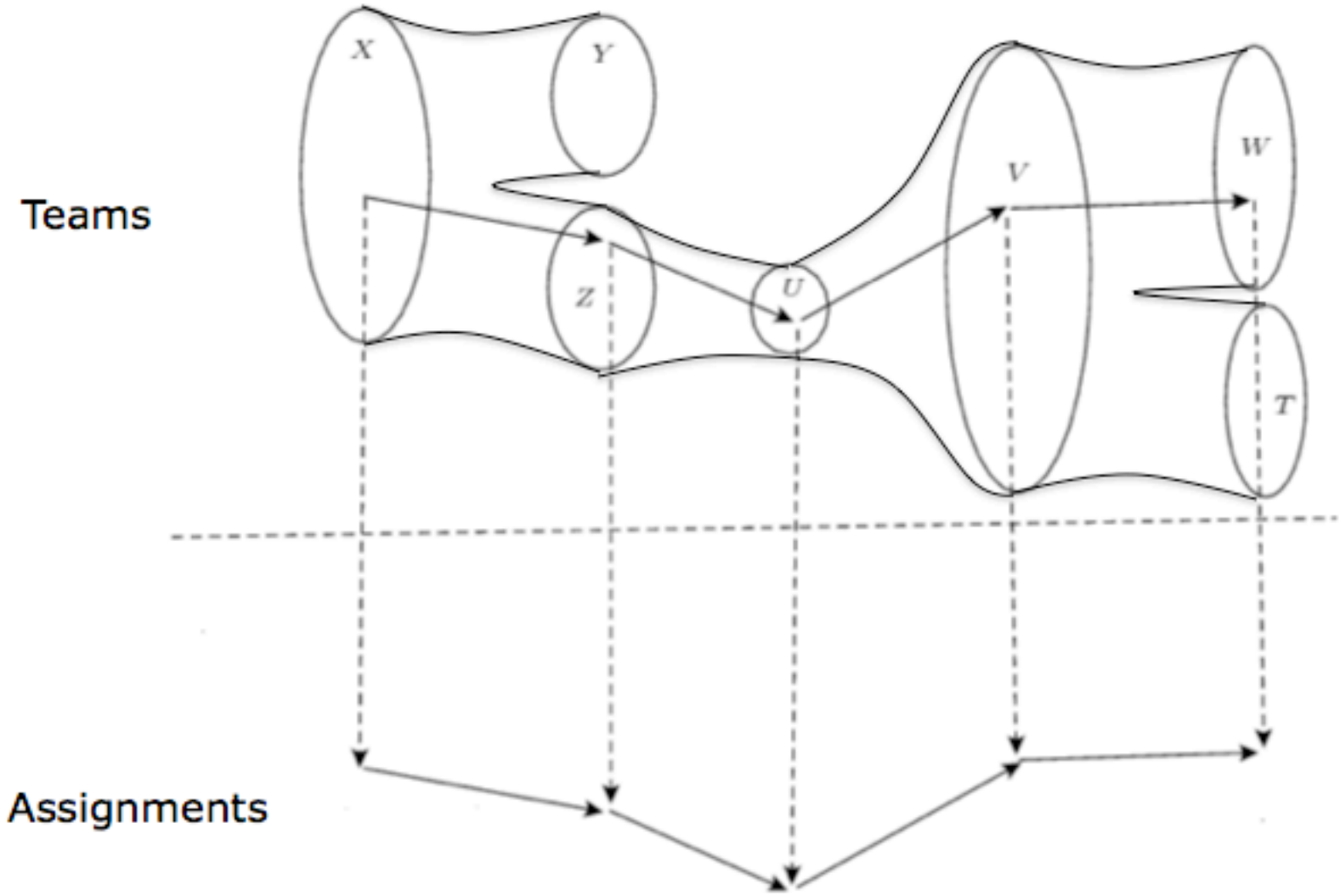
- Dependence logic has two versions of the following games
  - Semantic (evaluation) game
  - Ehrenfeucht-Fraisse game

Game theoretic semantics

- Dependence logic has two versions of the following games
  - Semantic (evaluation) game
  - Ehrenfeucht-Fraisse game
- Version 1: Players move assignments.
  - Non-deterministic, imperfect information.

Game theoretic semantics

- Dependence logic has two versions of the following games
  - Semantic (evaluation) game
  - Ehrenfeucht-Fraisse game
- Version 1: Players move assignments.
  - Non-deterministic, imperfect information.
- Version 2: Players move teams.
  - Deterministic, perfect information.


Teams
X
Y
Z
U
V
W
T
Assignments

The game-intuition

- Teams are records of playing the game, formulas describe rules of the game
- Atomic type: a simple rule
- Negative atomic type: what is forbidden
- Dependence atom: what player is allowed to know
- Disjunction: playing in parallel
- Conjunction: playing in sequence
- Existential quantifier: to have a move
- Universal quantifier: trying all moves

Semantic game of FO

I

II

Semantic game of FO

I

II

Players hold a formula, one player at a time. Each thinks that if he or she holds the formula, it is true.

Semantic game of FO

I

II

Players hold a formula, one player at a time. Each thinks that if he or she holds the formula, it is true.

To account for free variables, they actually hold a pair (φ,s), where s is an assignment.

Beginning of the game

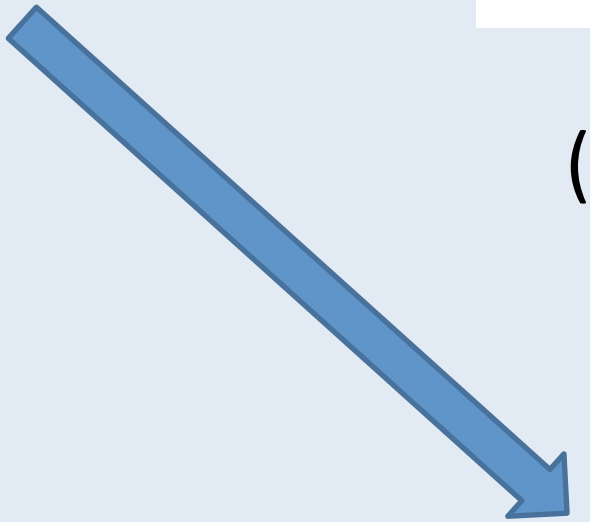
(φ, s)

Conjunction move: “other”

Conjunction move: “other”

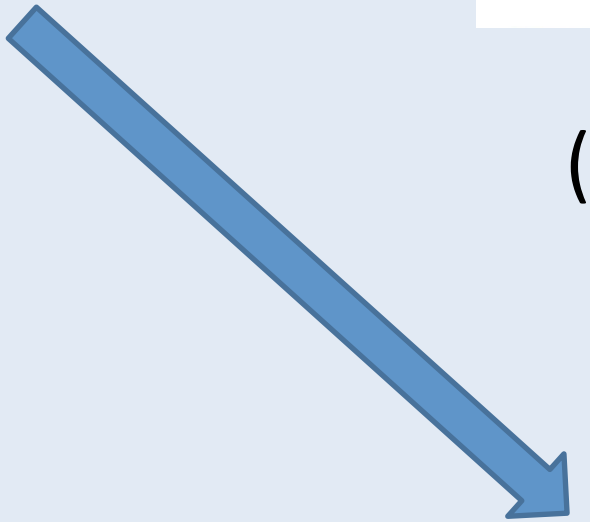
$(\varphi \wedge \psi, s)$

Conjunction move: “other”



$(\varphi \wedge \psi, s)$

Conjunction move: “other”



$(\varphi \wedge \psi, s)$

(ψ, s)

Conjunction move: “other”

Conjunction move: “other”

$(\varphi \wedge \psi, s)$

Conjunction move: “other”

$(\varphi \wedge \psi, s)$

Conjunction move: “other”

$(\varphi \wedge \psi, s)$

(φ, s)

Disjunction move: “self”

Disjunction move: “self”

$(\varphi \vee \psi, s)$

Disjunction move: “self”

$(\varphi \vee \psi, s)$

Disjunction move: “self”

$(\varphi \vee \psi, s)$

(φ, s)

Disjunction move: “self”

Disjunction move: “self”

$(\varphi \vee \psi, s)$

Disjunction move: “self”

$(\varphi \vee \psi, s)$

Disjunction move: “self”

$(\varphi \vee \psi, s)$

(ψ, s)

Negation move: ``swap”

Negation move: ``swap”

$(\neg\varphi, s)$

Negation move: ``swap”

$(\neg\varphi, s)$

Negation move: ``swap”

(φ,s) $\longleftarrow$ $(\neg\varphi,s)$

Negation move: ``swap”

$(\neg\varphi, s)$

Negation move: ``swap”

$(\neg\varphi, s)$ 

Negation move: ``swap”

$(\neg\varphi, s)$ → (φ, s)

Existential quantifier move: “self”

Existential quantifier move: “self”

$(\exists x\varphi,s)$

Existential quantifier move: “self”

$(\exists x\varphi,s)$

Existential quantifier move: “self”

$(\exists x\varphi,s)$

$(\varphi,s(a/x))$

Existential quantifier move: “self”

Existential quantifier move: “self”

$(\exists x\varphi,s)$

Existential quantifier move: “self”

$(\exists x\varphi,s)$

Existential quantifier move: “self”

$(\exists x\varphi,s)$

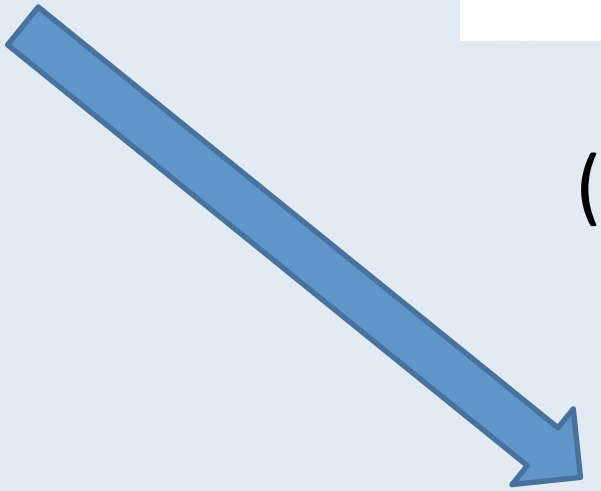
$(\varphi,s(a/x))$

Universal quantifier move: “other”

Universal quantifier move: “other”

$(\forall x\varphi,s)$

Universal quantifier move: “other”



$(\forall x\varphi,s)$

Universal quantifier move: “other”

$(\forall x\varphi,s)$

$(\varphi,s(a/x))$

Universal quantifier move: “other”

Universal quantifier move: “other”

$(\forall x\varphi,s)$

Universal quantifier move: “other”

$(\forall x\varphi,s)$

Universal quantifier move: “other”

$(\forall x\varphi,s)$

$(\varphi,s(a/x))$

Atomic formula: game ends

Atomic formula: game ends

(φ,s) true

Atomic formula: game ends

(φ,s)

true

Atomic formula: game ends

(φ,s)

Atomic formula: game ends

(φ,s)

false

Atomic formula: game ends

(φ,s)

false

Atomic formula: game ends

Atomic formula: game ends

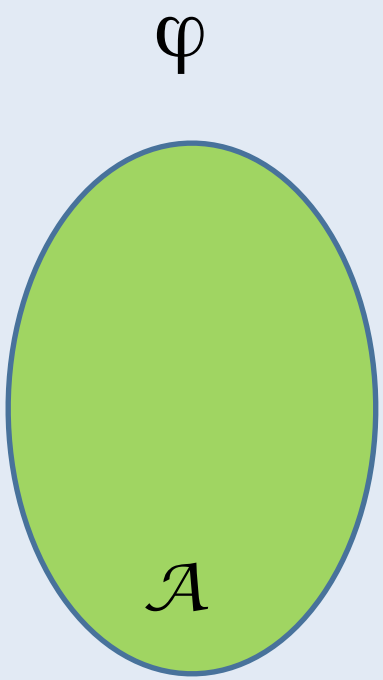
(φ,s)

Atomic formula: game ends

(φ,s)

false

Game theoretical semantics



φ is true in $\mathcal{A}$ if and only if II has a winning strategy

φ is false in $\mathcal{A}$ if and only if I has a winning strategy

Truth → winning strategy

- Winning strategy of a player: make sure that if you hold a formula, it is true, and if the other guy holds a formula it is false.

Truth ⇐ winning strategy

- By induction on the formula: If II is playing her winning strategy and
  - she holds a formula then it is true, and if
  - he is holding a formula, it is false.

First semantic game of *D*

Moves for all logical operations and atomic formulas are exactly the same as for first order logic, except for the new dependence atom.

Dependence atom

Dependence atom

$(=(t_1,\ldots,t_n),s)$

Dependence atom

$(=(t_1,\ldots,t_n),s)$

Dependence atom

$(=(t_1,\ldots,t_n),s)$

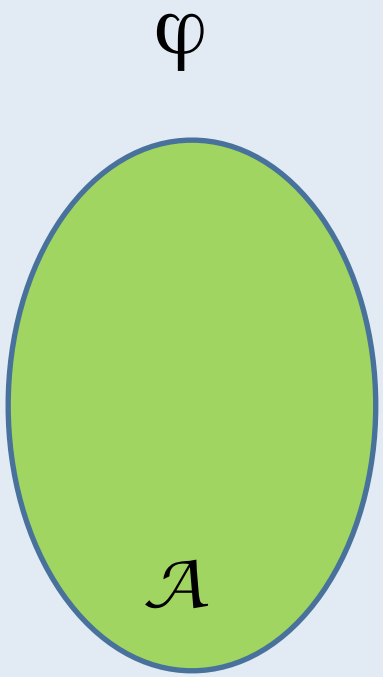
Dependence atom

$(=(t_1,\ldots,t_n),s)$

Uniform strategy

- A strategy of II is uniform if whenever the game ends in II holding $(=(t_1,\ldots,t_n),s)$ with the same $=(t_1,\ldots,t_n)$ and the same values of $t_1,\ldots,t_{n-1}$, then also the value of t_n is the same.
- Imperfect information: II cannot use anything but the values of $t_1,\ldots,t_{n-1}$ when she chooses t_n.

Game theoretical semantics of *D*



φ is true in $\mathcal{A}$ if and only if II has a uniform winning strategy.

Non-determined

Truth ➔ winning strategy

- **Winning strategy of II:** keep holding an auxiliary team X and make sure that if you hold a pair (φ,s), then $s \in X$ and X satisfies φ, and if the opponent holds (φ,s), then $s \in X$ and X satisfies $\neg\varphi$.

Truth ⇐ winning strategy

- Suppose II has a uniform winning strategy τ starting from $(\varphi,\{\varnothing\})$.
- **Idea**: Let X_ψ be the set of assignments s such that (ψ,s) is a position in the game, II playing τ.
- **By induction on** ψ: If II holds (ψ,s), then X_ψ satisfies ψ. If I holds (ψ,s), then X_ψ satisfies $\neg\psi$.

Second semantic game of *D*

This is like playing many semantic games in parallel.

Beginning of the game

(φ, X)

Disjunction move “self”

Disjunction move “self”

$(\varphi \vee \psi, X)$

Disjunction move “self”

$(\varphi \vee \psi, X)$

(φ, Y)

(ψ, Z)

Disjunction move “self”

$(\varphi \vee \psi, X)$

(ψ, Z)

Disjunction move “self”

$(\varphi \vee \psi, X)$

Disjunction move “self”

$(\varphi \vee \psi, X)$

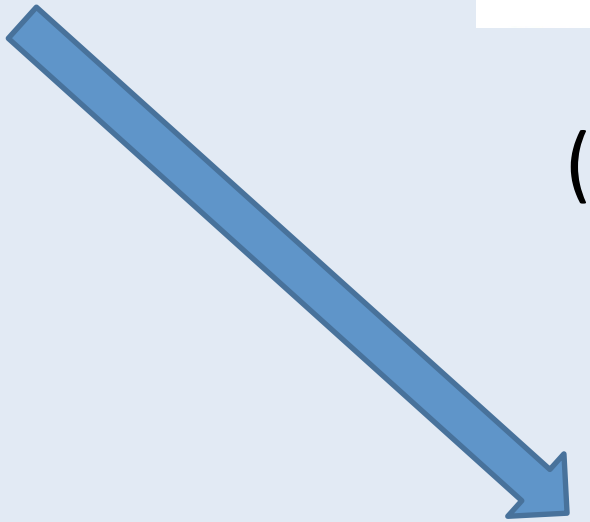
(ψ, X)

Conjunction move “other”

Conjunction move “other”

$(\varphi \wedge \psi, X)$

Conjunction move “other”



$(\varphi \wedge \psi, X)$

(ψ, X)

Conjunction move “other”

$(\varphi \wedge \psi, X)$

Conjunction move “other”

$(\varphi \wedge \psi, X)$

(φ, Y)

(ψ, Z)

Conjunction move “other”

$(\varphi \wedge \psi, X)$

(φ, Y)

Negation move ``swap”

Negation move ``swap”

$(\neg\varphi, X)$

Negation move ``swap”

$(\neg\varphi, X)$ 

Negation move ``swap”

$(\neg\varphi, X)$ → (φ, X)

Negation move ``swap”

Negation move ``swap”

$(\neg\varphi, X)$

Negation move ``swap”

$(\neg\varphi, X)$

Negation move ``swap”

(φ, X) ⟵ $(\neg\varphi, X)$

Existential quantifier move (“self”)

Existential quantifier move (“self”)

$(\exists x\varphi, X)$

Existential quantifier move (“self”)

$(\exists x\varphi, X)$

Existential quantifier move (“self”)

$(\exists x\varphi, X)$

$(\varphi, X(F/x))$

Existential quantifier move (“self”)

Existential quantifier move (“self”)

$(\exists x\varphi, X)$

Existential quantifier move (“self”)

$(\exists x\varphi, X)$

Existential quantifier move (“self”)

$(\exists x\varphi, X)$

$(\varphi, X(M/x))$

Universal quantifier move (“other”)

Universal quantifier move (“other”)

$(\forall x\varphi, X)$

Universal quantifier move (“other”)

$(\forall x\varphi, X)$

Universal quantifier move (“other”)

$(\forall x\varphi, X)$

$(\varphi, X(M/x))$

Universal quantifier move (“other”)

Universal quantifier move (“other”)

$(\forall x\varphi, X)$

Universal quantifier move (“other”)

$(\forall x\varphi, X)$

Universal quantifier move (“other”)

$(\forall x\varphi, X)$

$(\varphi, X(F/x))$

Atomic formula

Atomic formula

(φ, X)

Atomic formula

(φ, X)

$X \cap \varphi = \varnothing$

Atomic formula

(φ, X)

$X \cap \varphi = \varnothing$

Atomic formula

(φ, X)

Atomic formula

(φ, X) $X \subseteq \varphi$

Atomic formula

(φ, X) $X \subseteq \varphi$

Dependence atom

Dependence atom

$(=(t_1,\ldots,t_n),x)$

Dependence atom

$(=(t_1,\ldots,t_n),x)$ $x=\emptyset$

Dependence atom

$(=(t_1,\ldots,t_n),x)$ $x=\emptyset$

Dependence atom

$(=(t_1,\ldots,t_n),X)$

Dependence atom

$(=(t_1,\ldots,t_n),X)$

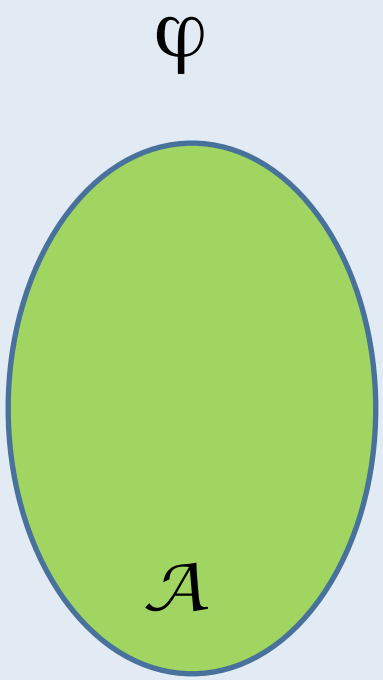
X satisfies the dependence

Dependence atom

$(=(t_1,\ldots,t_n),X)$

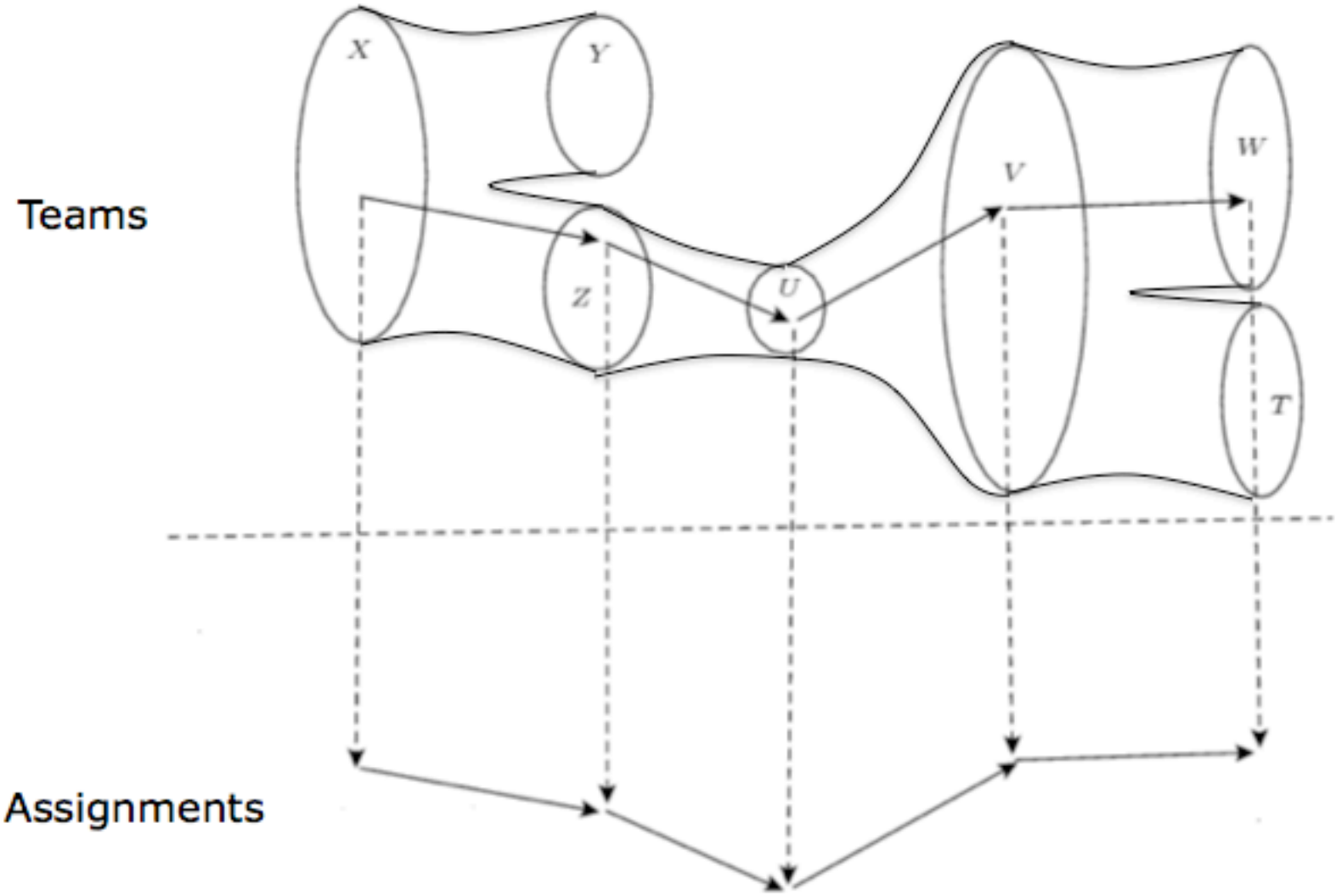
X satisfies the dependence

Game theoretical semantics



X satisfies φ in $\mathcal{A}$ if and only if
II has a winning strategy.

Determined, perfect information


Teams
X
Y
Z
U
V
W
T
Assignments

Winning str ⇐ truth

- Winning strategy of II: make sure that if you hold (φ,X), then X satisfies φ, and if he holds (φ,X), then X satisfies $\neg\varphi$.

Winning str ➔ truth

- By induction on the formula φ: If II is playing her winning strategy and
  - she holds (φ,X), then X satisfies φ, and if
  - he is holding (φ,X), then X satisfies $\neg\varphi$.

Wrap up of games

- Version 1: Players move assignments.
  - Non-deterministic, imperfect information.
- Version 2: Players move teams.
  - Deterministic, perfect information.
- Same with EF-game.

Model theory of dependence logic

Hodges 1997: For every formula $\varphi(x_1,\ldots,x_n)$ there is an existential second order sentence $\Phi(P)$ with P negative such that a team X satisfies φ iff $\Phi(X)$ is true.

Model theory of dependence logic

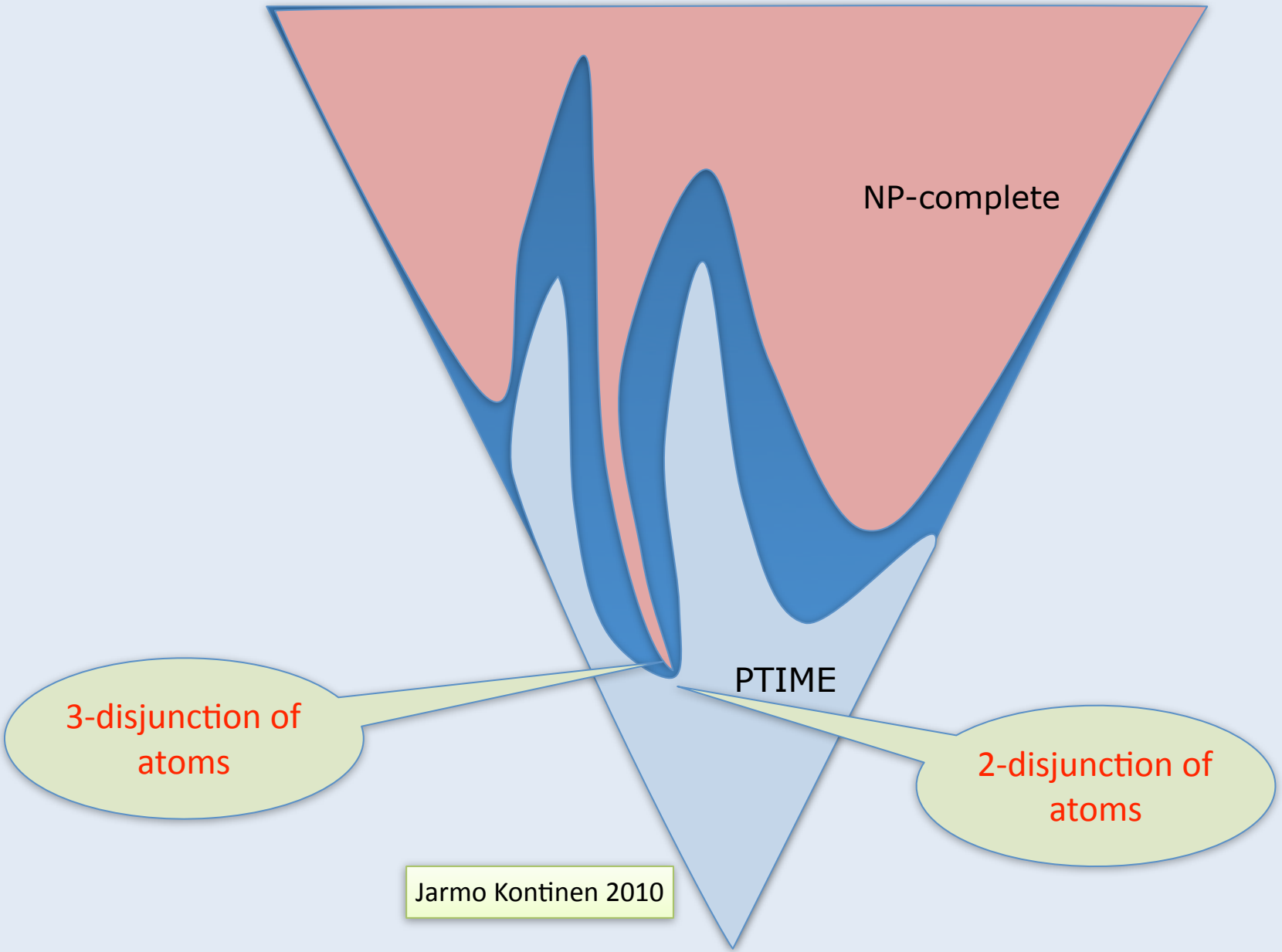
Hodges 1997: For every formula $\varphi(x_1,\ldots,x_n)$ there is an existential second order sentence $\Phi(P)$ with P negative such that a team X satisfies φ iff $\Phi(X)$ is true.

Theorem (Kontinen-V. 2008): The converse is also true (for non-empty teams).

Answers a question of Hodges.

Consequences

- A language for NP on finite models.
- Compactness.
- Löwenheim-Skolem.
- Separation (Interpolation).


NP-complete
PTIME
3-disjunction of atoms
2-disjunction of atoms
Jarmo Kontinen 2010

Coherence (Jarmo Kontinen, 2010)

- Formula is n-coherent if a team satisfies it whenever all subteams of size ≤n do.
- First order formulas are 1-coherent.
- Dependence atoms are 2-coherent.
- Disjunctions of two dependence atoms need not be n-coherent for any n.

Deskolemization (with V. Goranko)

- Skolemize first order formula.
- Do something.
- Can you get back to a first order formula.
- Non-arithmetical on all models.
- Nonrecursive on finite models.
- Can always deskolemize into dependence logic.

Classical negation

- The closure of dependence logic under classical negation has the exact strength of second order logic (Ville Nurmi, 2008).
- But we need negation to express Arrow's Theorem?

How about intuitionistic negation?

Joint work with S. Abramsky.

- *Definition:* X satisfies $\varphi \to \psi$ iff every subteam of X which satisfies φ also satisfies ψ.
- *Definition:* X satisfies $\bot$ iff X is the empty team.
- $\neg\, \varphi$ is now equivalent to $\varphi \to \bot$ for atomic φ.
- Intuitionistic negation ($\varphi \to \bot$) is an alternative way to extend negation from atomic to non-atomic formulas.

How about intuitionistic negation?

Joint work with S. Abramsky.

- *Definition:* X satisfies $\varphi \to \psi$ iff every subteam of X which satisfies φ also satisfies ψ.
- *Definition:* X satisfies $\bot$ iff X is the empty team.
- $\neg\, \varphi$ is now equivalent to $\varphi \to \bot$ for atomic φ.
- Intuitionistic negation ($\varphi \to \bot$) is an alternative way to extend negation from atomic to non-atomic formulas.

How about intuitionistic negation?

Joint work with S. Abramsky.

- *Definition:* X satisfies $\varphi \to \psi$ iff every subteam of X which satisfies φ also satisfies ψ.
- *Definition:* X satisfies $\bot$ iff X is the empty team.
- $\neg \varphi$ is now equivalent to $\varphi \to \bot$ for atomic φ.
- Intuitionistic negation ($\varphi \to \bot$) is an alternative way to extend negation from atomic to non-atomic formulas.

How about intuitionistic negation?

- Dependence atoms can now be defined in terms of constancy:

$$=(x_1,\ldots,x_n,z) \quad \equiv \quad (=(x_1) \wedge \ldots \wedge =(x_n)) \rightarrow =(z).$$

- Downward closure and the empty set property are preserved.
- Compactness fails.
- Goes beyond NP, unless NP=co-NP.

How about intuitionistic negation?

- Dependence atoms can now be defined in terms of constancy:

$$=(x_1,\ldots,x_n,z) \quad \equiv \quad (=(x_1) \wedge \ldots \wedge =(x_n)) \rightarrow =(z)$$

- Downward closure and the empty set property are preserved.
- Compactness fails.
- Goes beyond NP, unless NP=co-NP

How about intuitionistic negation?

- Dependence atoms can now be defined in terms of constancy:

$$=(x_1,\ldots,x_n,z) \quad \equiv \quad (=(x_1) \wedge \ldots \wedge =(x_n)) \rightarrow =(z)$$

- Downward closure and the empty set property are preserved.
- Compactness fails.
- Goes beyond NP, unless NP=co-NP.

How about intuitionistic negation?

- Dependence atoms can now be defined in terms of constancy:

$$=(x_1,\ldots,x_n,z) \quad \equiv \quad (=(x_1) \wedge \ldots. \wedge =(x_n)) \rightarrow =(z)$$

- Downward closure and the empty set property are preserved.
- Compactness fails.
- Goes beyond NP, unless NP=co-NP.

We can prove Armstrong's Axioms

Dependence logic	Heyting's intuitionistic logic
$=(x,x)$	$=(x) \rightarrow =(x)$
If $=(x,y,z)$, then $=(y,x,z)$.	If $(=(x) \wedge =(y)) \rightarrow =(z)$, then $(=(y) \wedge =(x)) \rightarrow =(z)$
If $=(x,x,y)$, then $=(x,y)$.	If $(=(x) \wedge =(x)) \rightarrow =(y)$, then $=(x) \rightarrow =(y)$
If $=(x,z)$, then $=(x,y,z)$.	If $=(x) \rightarrow =(z)$, then $(=(x) \wedge =(y)) \rightarrow =(z)$
If $=(x,y)$ and $=(y,z)$, then $=(x.z)$.	If $=(x) \rightarrow =(y)$, and $=(y) \rightarrow =(z)$ then $=(x) \rightarrow =(z)$

Linear implication

- X satisfies φ —o ψ iff for every team Y which satisfies φ the team X ∪ Y satisfies ψ.
- Downward closure is preserved.
- Compactness fails.
- Goes beyond NP unless NP=co-NP.

Galois connections

- Intuitionistic implication is the adjoint of conjunction:

$$(\phi \wedge \psi) \models \theta \iff \phi \models \psi \to \theta$$

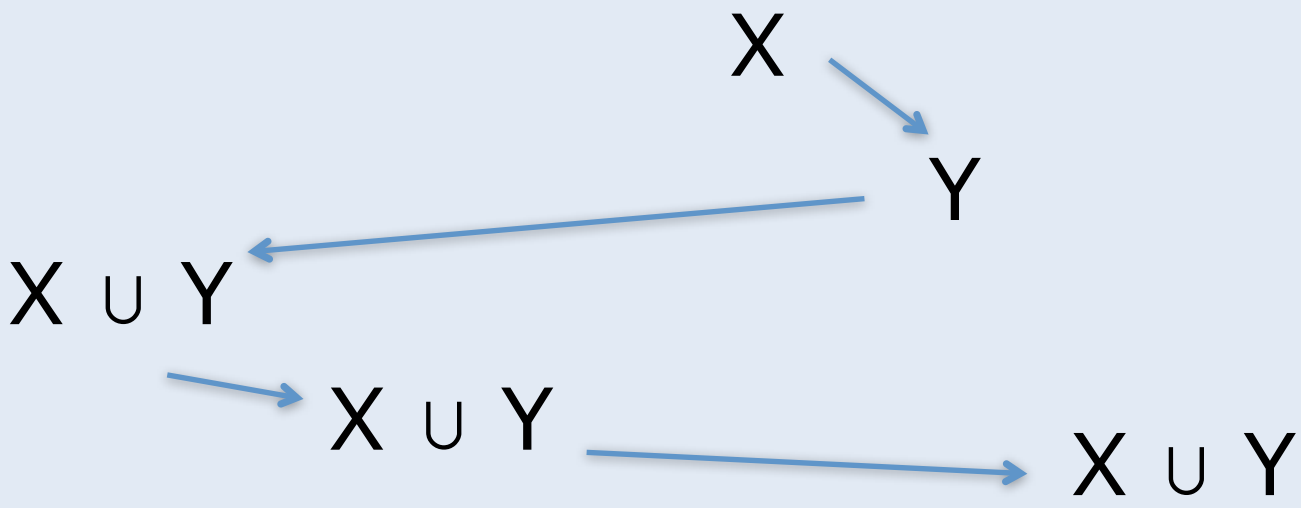
- Linear implication is the adjoint of disjunction.

$$(\phi \vee \psi) \models \theta \iff \phi \models \psi \multimap \theta$$

Proof

- Linear implication is the adjoint of disjunction.

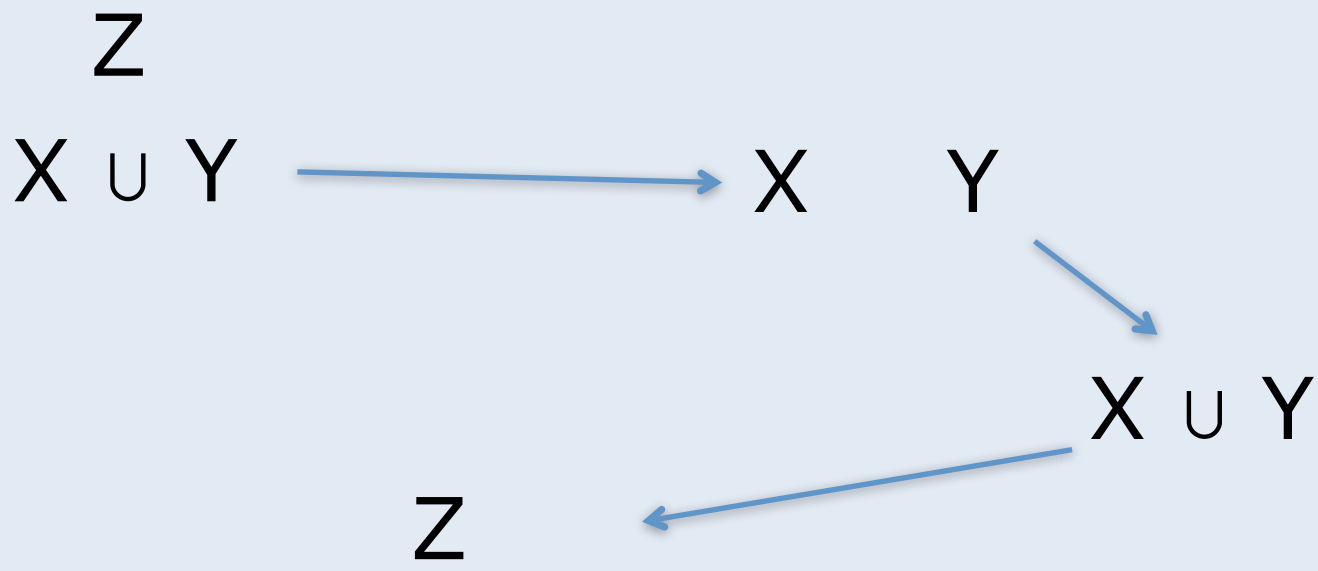
$$(\phi \vee \psi) \models \theta \iff \phi \models \psi \multimap \theta$$



Proof

- Linear implication is the adjoint of disjunction.

$$(\phi \vee \psi) \models \theta \iff \phi \models \psi \multimap \theta$$



The moral of the story

- One can add both intuitionistic and linear implication to dependence logic without losing the downward closure.
- Intuitionistic negation agrees with the original negation on the atomic level, and basic axioms of dependence become provable.
- Good (?) for proof theory, but bad (?) for model theory. Is there a reason for this?

What is dependence logic good for?

- Language for NP.
- Tool for the study of more complex dependencies than just the Armstrong ones.
- A vehicle for uncovering the mathematics of dependence in a variety of contexts
  - Data mining
  - Social choice theory
  - Logic for Rationality and Interaction

- J. Väänänen, *Dependence Logic*, Cambridge University Press, 2007.
- Logic for Interaction (LINT), ESF LogICCC

Thank you!