Lecture 1: From Assignments to Teams

Aleksi Anttila Marco Degano

Team Semantics: Linguistic and Philosophical Applications
ESSLLI 2026, Prague

10 August 2026

Course Team

Marco Degano

Aleksi Anttila

Course Aim

- Introduce team semantics with linguistic and philosophical applications and audience in mind.
- Study the basic syntax, semantics, and structural properties of team-based logics.
- Applied to:
  - free choice and disjunction
  - questions and inquisitive content
  - indefinites, scope, and specificity
  - modality, negation, and compositionality

Materials

- The slides are the primary study material.
- https://team-logics.github.io/
- *Suggested* readings will be provided for each class.
- *Optional* exercises for each class will be provided at the end of each slide deck.

Tentative schedule

- **Monday:** Introduction and formal basics
- **Tuesday:** Propositional team logics and neglect-zero
- **Wednesday:** Inquisitive semantics and inquisitive logic
- **Thursday:** First-order team logics and indefinites
- **Friday:** Philosophical applications: compositionality, negation, modality.

Readings

- The first team semantics paper: W Hodges, Compositional semantics for a language of imperfect information, Logic Journal of the IGPL, Volume 5, Issue 4, July 1997, Pages 539–563, https://doi.org/10.1093/jigpal/5.4.539
- High-level overview: https://plato.stanford.edu/entries/logic-dependence/

Outline

1. Motivation

2. Precursors

3. Team Semantics

The Classical Point of Evaluation

- In ordinary propositional semantics, a valuation assigns truth values to propositional variables.

	p	q
$v_{p\bar{q}}$	1	0

- Then we ask whether a formula is true at this single valuation: $v_{p\bar{q}} \vDash p \wedge \neg q$

Information Is Often Not Point-Like

- Suppose our information does not settle the values of p and q. How to express this?
- Using a team as an information state: the actual valuation is somewhere in T, but the information state does not determine which row is actual.

T	p	q
v_{pq}	1	1
$v_{p\bar{q}}$	1	0
$v_{\bar{p}\bar{q}}$	0	0

- This team rules out that $p = 0$ and $q = 1$, but it leaves open three possibilities.

The Space of Possibilities

- For $N = \{p, q\}$, 2^N contains four valuations.

	p	q
v_{pq}	1	1
$v_{p\bar{q}}$	1	0
$v_{\bar{p}q}$	0	1
$v_{\bar{p}\bar{q}}$	0	0

The four possible valuations. $T_1 = \{v_{pq}\}$ $T_2 = \{v_{pq}, v_{p\bar{q}}\}$
$T_3 = \{v_{pq}, v_{\bar{p}q}\}$ $T_4 = \{v_{pq}, v_{p\bar{q}}, v_{\bar{p}\bar{q}}\}$ $T_5 = 2^N$

- A valuation selects one row.
- A team selects a set of rows.

Support and Compatibility

- $T \vDash p$ means that every $v \in T$ has $v(p) = 1$.
- p is compatible with T means that some $v \in T$ has $v(p) = 1$.
- $T \vDash \neg p$ means that every $v \in T$ has $v(p) = 0$.
- T leaves p open if some $v \in T$ has $v(p) = 1$ and some $w \in T$ has $w(p) = 0$.

A First Look at Dependence

- Now consider a different kind of question.
- Not is q true throughout the team? But is the value of q determined by the value of p?

T	p	q
v_1	0	1
v_2	1	0
v_3	1	0

  - If $p = 0$, then $q = 1$.
  - If $p = 1$, then $q = 0$.
- So, in this sense of dependence, in this team q depends on p.
- But if we say $v_3(q) = 1$, then dependence is lost.
- Team semantics can add atoms that express team-level properties, like dependence.

Outline

What Counts as Team-Like?

- A framework is team-like in the strict sense if formulas are evaluated relative to a set of points.
- The points may be valuations, assignments, or world-assignment pairs.
- The semantic relation then has the form $M, T \vDash \phi$ or $s \vDash \phi$, where T, or s is a set.
- The clauses for connectives or quantifiers must operate on that set.

Three Degrees of Team-Likeness

- Strictly team-like: formulas are evaluated at sets of assignments.
- Example: Hodges-style team semantics, dependence logic, inquisitive semantics.
- Operationally team-like: meanings are updates on information states.
- Example: dynamic semantics.
- Only indirectly team-like: the framework can be represented using sets of assignments, but this is not the primitive satisfaction format.
- Example: discourse representation theory.
- Not team-like in the relevant sense: sets appear as denotations or alternatives.
- Example: ordinary possible-worlds semantics

Possible-Worlds Semantics

- Ordinary possible-worlds semantics is not team-like in the relevant sense.
- A proposition is a set of worlds.
- But satisfaction is evaluated at a single world: $M, w \vDash \phi$.
- The set $[\![\phi]\!] = \{w \in W \mid M, w \vDash \phi\}$ is a denotation.
- In a team-setting, the denotation for ϕ would be a set of sets of points.

Discourse Representation Theory

- DRT is only indirectly team-like.
- A discourse representation structure introduces discourse referents and conditions.
- A DRS is satisfied if its discourse referents can be assigned objects in the model so that all its conditions hold.
- The set of satisfying embeddings can be viewed as a set of assignments.
- In that derived sense, a DRS can determine a team-like object.
- But the primitive semantic object of DRT is a discourse representation structure, not a team.

$$K = \left[\begin{array}{c} x \quad y \\ \hline \text{MAN}(x) \\ \text{DONKEY}(y) \\ \text{OWNS}(x, y) \end{array}\right] \qquad \rightsquigarrow \qquad X_K = \{g \mid M, g \vDash K\}$$

Kamp (1981), Kamp & Reyle (1993)

From Truth Conditions to Updates

- Static semantics asks whether a sentence is true at a point:

$$M, w \vDash \phi$$

- A sentence denotes a proposition:

$$[\![\phi]\!]_M = \{w \in W_M \mid M, w \vDash \phi\}$$

- Dynamic semantics asks what accepting a sentence does to information.
- A meaning is therefore a context-change potential.
- Instead of only asking whether ϕ is true at w, we ask how ϕ updates an information state.

Heim (1982), Veltman (1996), Groenendijk, Stokhof & Veltmam (1996)

Meaning as Context Change

- Contexts are modeled as information states.
- An information state $s \subseteq W$ contains the worlds still compatible with the available information.
- Updating s with ϕ removes the worlds incompatible with ϕ:

$$s[\phi] = s \cap [\![\phi]\!]_M$$

- Thus a sentence can be seen as an operation:

$$[\![\phi]\!]^{Dyn}_M : s \mapsto s[\phi]$$

- More information means fewer live possibilities:

$$s[\phi] \subseteq s$$

Dynamic Plural Logic

- Van den Berg's Dynamic Plural Predicate Logic extends dynamic predicate logic to account for plural anaphora.
- A discourse state is a set of finite partial assignments $G = \{g_1, \ldots, g_n\}$, where $g_i : V \rightharpoonup |M|$.

G	x	y
g_1	Bill	Mary
g_2	John	Ann
g_3	Harry	Joan
g_4	Charles	Joan

- The rows preserve dependencies between discourse referents.
- A formula specifies a transition between assignment states, written $G[\![\phi]\!]H$.

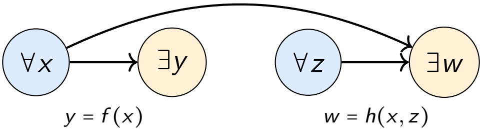
van den Berg (1996)

Why This Is Team-Like

- Dynamic semantics shifts attention from single points to sets of points.
- In propositional dynamic semantics, the relevant state is a set of worlds $s \subseteq W$
- In first-order dynamic semantics, the relevant state may be a set of assignments, or a set of world-assignment pairs.
- This is team-like because semantic operations act on sets of evaluation points.

Linear Quantification

- In ordinary first-order logic, quantifiers are linearly ordered.
- Example: $\forall x \exists y \forall z \exists w \, \phi(x, y, z, w)$.
- The witness for y may depend on x.
- The witness for w may depend on x, z
- Dependence is implicit in the order of quantifiers.



The Limitation of Linear Prefixes

- Linear prefixes impose a total order on quantifier choices.
- But some dependence patterns are naturally partial.
- Example: one may want y to depend only on x and w to depend only on z.
- Ordinary linear order cannot directly express this independence pattern.

Branching Quantifiers

- Branching quantifiers make dependence patterns partially ordered.

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

- The choice of y does not depend on z.
- The choice of w does not depend on x.
- This is not yet team semantics, but it isolates dependence and independence as semantic phenomena.

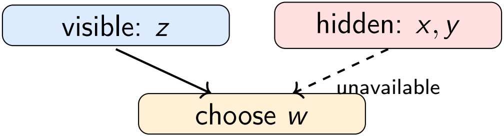
$$\forall x \longrightarrow \exists y \qquad y = f(x)$$

$$\forall z \longrightarrow \exists w \qquad w = g(z)$$

Henkin (1961), Barwise (1979)

From Henkin to IF Logic

- Independence-friendly logic makes independence restrictions syntactic.
- The notation $(\exists w/W)$ means that the choice of w must be independent of the variables in W.
- The Henkin pattern can be written linearly as $\forall x \exists y \forall z (\exists w/\{x,y\})\, \phi(x,y,z,w)$.
- The slash set $\{x,y\}$ prevents the choice of w from using information about x or about the earlier choice of y.
- The choice of w may still depend on z.



Hintikka & Sandu (1989, 1997), Hintikka (1996)

Informal Game-Theoretical Semantics

Two players: a verifier trying to show the formula ϕ is true, and a falsifier trying to show it is false, with respect to a model $\mathcal{M}$ and assignment s. ϕ is true/false w.r.t. $\mathcal{M}$ and s if the verifier/falsifier has a winning strategy.

- If ϕ is atomic, the verifier wins if $\mathcal{M} \vDash_s \phi$. The falsifier wins if $\mathcal{M} \nvDash_s \phi$
- If ϕ is $\psi \vee \chi$, the verifier picks $\theta := \psi$ or $\theta := \chi$. Play the game with θ.
- If ϕ is $\psi \wedge \chi$, the falsifier picks $\theta := \psi$ or $\theta := \chi$. Play the game with θ.
- ϕ is $\exists x \psi$: The verifier picks $a \in \mathcal{M}$. Play the game with $s(a/x)$ & ψ.
- ϕ is $\forall x \psi$: The falsifier picks $a \in \mathcal{M}$. Play the game with $s(a/x)$ & ψ.
- ϕ is $\neg\phi$: The players swap their falsifier/verifier roles.

Example: Model: $\mathcal{M}$ that has an element a with $a \in R^{\mathcal{M}}$. Assignment: empty assignment s. Sentence: $\exists x(Rx \wedge x = x)$. The verifier has a winning strategy:

- For $\exists x(Rx \wedge x = x)$, choose a.
- If the falsifier chooses Rx, the verifier wins because $a \in R^{\mathcal{M}}$.
- If the falsifier chooses $x = x$, the verifier wins because $a = a$.

Informal Game-Theoretical Semantics for IF

The standard first-order logic game is one of *perfect information*: each player has knowledge of each prior choice made in the game.

For slashed quantifiers, we have rules such as:

- ϕ is $\exists x/\{y\}\psi$: The verifier picks $a \in \mathcal{M}$ without knowing $s(y)$. Play the game with $s(a/x)$ & ψ.

With these quantifiers, the game is one of *imperfect information*.

Matching pennies game in independence-friendly logic

Each player has a coin that they secretly turn to heads or tails. The coins are revealed simultaneously. *I* wins if the coins are both heads or both tails. Otherwise *II* wins.

Let $\mathcal{M}$ be a model with domain $\{h, t\}$ and ϕ be the sentence

$$\forall x(\exists y/\{x\})x = y$$

Neither player has a winning strategy, so $\mathcal{M} \not\models \phi$ and $\mathcal{M} \not\models \neg\phi$.

II (falsifier) does not have a winning strategy: all they can do is choose the value of x. They cannot guarantee that *I* will not choose the same value for y.

I (verifier) does not have a winning strategy: *II* first chooses the value of x. *I* does not know which value was picked, so they cannot ensure that they pick the same value for y.

Hintikka had a wider philosophical program with IF logic at its centre:

- Unlike standard Tarskian semantics, the game-theoretical semantics for logic explain *why* the resulting truth relation is a truth relation: game-theoretical truth is justified in terms the actual activities by which we verify or falsify sentences (a language game à la Wittgenstein for truth).
- From the viewpoint of game-theoretical semantics, the restriction to only perfect-information games in arbitrary.
- Given this (and more), IF logic should replace classical first-order logic as the 'fundamental' logic used in mathematics.

Hintikka 1996

Hintikka also argued against the importance commonly placed on compositionality in the semantics of both natural and formal languages. He assumed that it would be impossible to provide a (reasonable) compositional semantics for IF logic, and used this as a point against compositionality.

Barwise, commenting on Hintikka:

> *Linguistically, the discovery of branching quantification [in natural language] would force us to re-examine, and perhaps re-interpret, Frege's principle of compositionality according to which the meaning of a given expression is determined by the meanings of its constituent phrases. For example, the meaning of a branching quantifier expression of logic*
>
> $$\begin{pmatrix} \forall x & \exists y \\ \forall z & \exists w \end{pmatrix} A(x, y, z, w)$$
>
> *cannot be defined inductively in terms of simpler formulas, by explaining away one quantifier at a time.*

Hodges' development of team semantics showed that there was a reasonable compositional semantics for IF.

Hintikka 1996, Barwise 1979

Hodges' Move

- Tarski semantics evaluates first-order formulas at one assignment: $M \vDash_g \alpha$.
- Hodges gives a compositional semantics for IF logic using sets of assignments.
- These sets were called trumps (!). In later terminology, they are teams.
- The satisfaction relation becomes $M \vDash_X \alpha$, where X is a set of assignments.
- This is the first-order analogue of evaluating propositions on information states.

Hodges (1997, 1998), Väänänen (2007)

Outline

Propositional Team Semantics

- Let Prop be a fixed set of propositional variables.
- For finite $N \subseteq \mathrm{Prop}$, let $2^N = \{v \mid v : N \to \{0,1\}\}$.
- A *team over* N is a set $T \subseteq 2^N$.
- Thus $v \in T$ is a valuation and T is a set of valuations.
- If $V \subseteq N$, write $v \restriction V$ for restriction and $T \restriction V = \{v \restriction V \mid v \in T\}$.

T	p	q
v_1	1	1
v_2	1	0
v_3	0	1

$T \restriction \{p\}$	p
v_1	1
v_2	1
v_3	0

Yang & Väänänen (2016, 2017)

Formulas

- Propositional dependence logic PD has formulas

$$\phi ::= p \mid \neg p \mid \mathrm{dep}(p_1, \dots, p_n, q) \mid \phi \wedge \psi \mid \phi \vee \psi$$

- Negation occurs only in literals p and $\neg p$.
- $\mathrm{Var}(\phi)$ is the finite set of variables occurring in ϕ.
- Satisfaction $T \vDash \phi$ is defined whenever $\mathrm{Var}(\phi) \subseteq \mathrm{Dom}(T)$.

Literal Clauses

- $T \vDash p$ iff $v(p) = 1$ for every $v \in T$.
- $T \vDash \neg p$ iff $v(p) = 0$ for every $v \in T$.
- Literals are evaluated universally over the team.

T	p	q
v_1	1	1
v_2	1	0

- $T \vDash p$
- $T \nvDash q$
- $T \nvDash \neg q$

Dependence Atom

$$T \models \mathrm{dep}(p_1, \ldots, p_n, q)$$

iff for all $v, w \in T$,

$$v(p_i) = w(p_i) \text{ for every } i \leq n \quad \Longrightarrow \quad v(q) = w(q)$$

Equivalently, there is a function

$$f : \{0,1\}^n \to \{0,1\}$$

such that, for every $v \in T$,

$$v(q) = f(v(p_1), \ldots, v(p_n))$$

T	p	q
v_1	0	1
v_2	1	0
v_3	1	0

Failure of Dependence

- Dependence fails when two rows agree on the determining variables but disagree on the determined variable.

T	p	q
v_1	0	0
v_2	0	1
v_3	1	1

- v_1 and v_2 agree on p.
- v_1 and v_2 disagree on q.
- Hence $T \not\models \mathrm{dep}(p, q)$.

Conjunction

- $T \vDash \phi \wedge \psi$ iff $T \vDash \phi$ and $T \vDash \psi$.
- Conjunction keeps the same team.

T	p	q
v_1	1	1
v_2	1	1

- $T \vDash p$.
- $T \vDash q$.
- Hence $T \vDash p \wedge q$.

Split Disjunction

- $T \vDash \phi \vee \psi$ iff there are $S, U \subseteq T$ such that $S \cup U = T$, $S \vDash \phi$, and $U \vDash \psi$.
- The team is split into a ϕ-part and a ψ-part.
- Empty subteams are allowed.
- The subteams may overlap.
- This is the lax split disjunction clause.

Example: Split Disjunction

- Let $T = \{v_1, v_2, v_3\}$.

T	p	q
v_1	1	1
v_2	1	0
v_3	0	1

- $T \vDash p \vee \neg p$.
- Take $S = \{v_1, v_2\}$ and $U = \{v_3\}$.
- $S \vDash p$.
- $U \vDash \neg p$.
- $S \cup U = T$.

Locality

- Satisfaction depends only on the variables occurring in the formula.
- If $\mathrm{Var}(\phi) \subseteq \mathrm{Dom}(T)$, then $T \vDash \phi$ iff $T \upharpoonright \mathrm{Var}(\phi) \vDash \phi$.
- If $T \upharpoonright \mathrm{Var}(\phi) = U \upharpoonright \mathrm{Var}(\phi)$, then $T \vDash \phi$ iff $U \vDash \phi$.
- Extra columns are semantically irrelevant.

E.g. for $Var(\phi) = \{p\}$:

T	p	q
v_1	1	1
v_2	1	0
v_3	0	1

Semantic Consequence

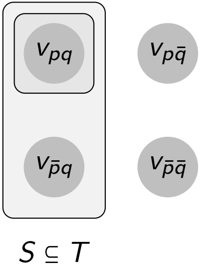
- For a set of formulas Γ, define $\Gamma \vDash \phi$ iff for every team T, if $T \vDash \gamma$ for all $\gamma \in \Gamma$, then $T \vDash \phi$.
- A formula ϕ is valid iff $T \vDash \phi$ for every team T over a domain containing $\mathsf{Var}(\phi)$.
- By locality, validity may be checked over teams $T \subseteq 2^{\mathsf{Var}(\phi)}$.

Closure Properties

- Team logics are classified by how formulas behave under changes of team.
- We ask what happens when we remove rows.
- We ask what happens when we add rows.
- We ask what happens when we take unions of teams.
- These properties are useful for understanding definability and the characteristics of meaning in different team logics, as well as for classifying logics: many later logics differ exactly here.

Downward Closure

- A formula ϕ is downward closed if $T \vDash \phi$ and $S \subseteq T$ imply $S \vDash \phi$.
- Removing rows preserves truth.



- Example: if $T \vDash q$, then every subteam $S \subseteq T$ also satisfies q.

Lax and Strict Split

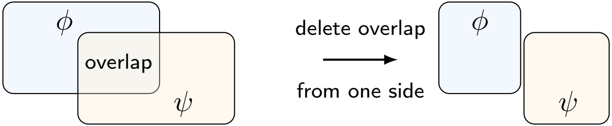
- In the lax clause for disjunction, S and U may overlap.
- In the strict clause for disjunction, one additionally requires $S \cap U = \varnothing$.
- For downward closed formulas, lax and strict split are equivalent.

Exercise: Lax vs. Strict Split

Assume ϕ and ψ are downward closed. Show that:

$$T \vDash_{\text{lax}} \phi \vee \psi \iff T \vDash_{\text{strict}} \phi \vee \psi$$

lax split **strict split**



$S \cup U = T$, possibly $S \cap U \neq \varnothing$

$S' \cup U' = T$, $S' \cap U' = \varnothing$

By downward closure: $S \vDash \phi \Rightarrow S' \vDash \phi$ and $U \vDash \psi \Rightarrow U' \vDash \psi$.

Upward Closure

- A formula ϕ is upward closed if $T \vDash \phi$ and $T \subseteq S$ imply $S \vDash \phi$.
- Adding rows preserves truth.
- Literals are not upward closed.
- Dependence atoms are not upward closed.

T	p
v_1	1

S	p
v_1	1
v_2	0

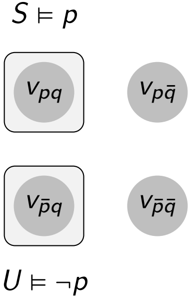
- $T \vDash p$
- $T \subseteq S$
- $S \nvDash p$

Flatness of the Classical Fragment

- A formula ϕ is flat if $T \vDash \phi$ iff $\{v\} \vDash \phi$ for every $v \in T$.
- Let ϕ contain no dependence atoms.
- Then $T \vDash \phi$ iff $\{v\} \vDash \phi$ for every $v \in T$ iff $v \vDash_{\mathrm{cl}} \phi$ for every $v \in T$.
- That is, classical formulas are not only flat, but their team semantics coincides with their standard semantics on singletons.
- Hence the team semantics of ordinary propositional formulas is pointwise classical.

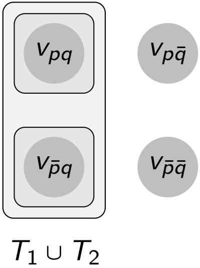
Split Disjunction Is Not Boolean Choice

- $T \vDash \phi \vee \psi$ does not generally mean that $T \vDash \phi$ or $T \vDash \psi$.
- It means that T can be covered by a ϕ-subteam and a ψ-subteam.
- For flat formulas, this agrees with ordinary pointwise disjunction.
- For non-flat formulas, the split itself can matter.



Union Closure

- A formula ϕ is union closed if $T_i \vDash \phi$ for all $i \in I \neq \varnothing$ implies $\bigcup_{i \in I} T_i \vDash \phi$.
- Pooling satisfying teams preserves truth.



- Example: if $T_1 \vDash q$ and $T_2 \vDash q$, then $T_1 \cup T_2 \vDash q$.

Union Closure Can Fail

- Dependence atoms are not union closed. Why?
- Singleton teams always satisfy dependence atoms.

T_1	p	q
v_1	0	0

T_2	p	q
v_2	0	1

$T_1 \cup T_2$	p	q
v_1	0	0
v_2	0	1

- $T_1 \vDash \mathrm{dep}(p, q)$ and $T_2 \vDash \mathrm{dep}(p, q)$.
- But $T_1 \cup T_2 \nvDash \mathrm{dep}(p, q)$.
- The union creates a disagreement on q among rows that agree on p.

Empty Team Property

- A formula ϕ has the empty team property if $\emptyset \vDash \phi$.
- Every PD-formula has the empty team property.
- For literals, this is vacuous universal quantification.
- For dependence atoms, this is vacuous universal quantification.
- For conjunction, the property is preserved immediately.
- For split disjunction? Take both subteams to be $\emptyset$.

Relations between Closure Properties

- Downward closure implies the empty team property, provided some team satisfies ϕ: if $T \vDash \phi$, then $\varnothing \subseteq T$, hence $\varnothing \vDash \phi$.
- Upward closure implies union closure.
- Flatness is equivalent to downward closure + union closure + empty team property:
  - Suppose ϕ is flat.
  - Downward closed: Let $T \vDash \phi$ and $S \subseteq T$. By flatness, $\{v\} \vDash \phi$ for all $v \in T$, so $\{v\} \vDash \phi$ for all $v \in S$, so by flatness again, $S \vDash \phi$.
  - Empty team property: Vacuously $\{v\} \vDash \phi$ for all $v \in \varnothing$, so $\varnothing \vDash \phi$.
  - Union closed: If $S_i \vDash \phi$ for all $i \in I \neq \varnothing$, then by flatness, $\{v\} \vDash \phi$ for all $v \in \bigcup_{i \in I} S_i$, so that by flatness again, $\bigcup_{i \in I} S_i \vDash \phi$.
  - Now suppose ϕ is downward- and union closed, and has the empty team property.
  - By downward closure, $T \vDash \phi$ implies $\{v\} \vDash \phi$ for all $v \in T$.
  - By union closure (if $T \neq \varnothing$) and the empty team property (if $T = \varnothing$), $\{v\} \vDash \phi$ for all $v \in T$ implies $T \vDash \phi$.

Closure Properties as Information Principles

- Closure properties encode assumptions about the kinds of meanings formulas express.
- If teams are information states, downward closure says: *once established, a fact remains established when we learn more.*
- Union closure says that satisfaction is preserved when compatible information states are pooled.
- Failure of a closure property is also informative: it shows that the logic can express genuinely team-level properties.

First-Order Teams

- Let M be a first-order model with domain $|M|$.
- Let V be a finite set of variables.
- An assignment over V is a function $g : V \to |M|$.
- A team over M with domain V is a set $X \subseteq |M|^V$.

From Propositional to First-Order Teams

- Propositional teams were sets of valuations: $T \subseteq 2^N$
- First-order teams are sets of assignments: $X \subseteq |M|^V$
- The guiding idea is the same.

X	x	y	z
g_1	a	b	c
g_2	a	b	d
g_3	e	f	c

First-Order Literals and Boolean Clauses

- First-order literals are evaluated pointwise:

$$M, X \vDash \alpha \quad \Longleftrightarrow \quad M, g \vDash \alpha \text{ for every } g \in X$$

- Similarly,

$$M, X \vDash \neg\alpha \quad \Longleftrightarrow \quad M, g \nvDash \alpha \text{ for every } g \in X$$

- Conjunction and disjunction work exactly as in the propositional case.

Quantifiers as Team Operations

- If $a \in |M|$, write $g[a/x]$ for the assignment just like g, except that x is mapped to a.
- Existential quantification supplements the team: $M, X \vDash \exists x\, \phi$ iff there is a function

  $$F : X \to \mathcal{P}(|M|) \setminus \{\varnothing\} \text{ such that } M, X[F/x] \vDash \phi$$

  where

  $$X[F/x] = \{g[a/x] \mid g \in X,\ a \in F(g)\}$$

Example: Supplementing

- Let $|M| = \{a, b, c, d\}$.
- Let $X = \{g_1, g_2\}$.
- Suppose $F(g_1) = \{a, b\}$ and $F(g_2) = \{b\}$.

X	y
g_1	c
g_2	d

$X[F/x]$	y	x
$g_1[a/x]$	c	a
$g_1[b/x]$	c	b
$g_2[b/x]$	d	b

Universal Quantification

- Universal quantification duplicates the team:

$$M, X \vDash \forall x\, \phi \quad \Longleftrightarrow \quad M, X[|M|/x] \vDash \phi$$

- Here

$$X[|M|/x] = \{g[a/x] \mid g \in X,\ a \in |M|\}$$

- So every assignment receives every possible value for x.

Example: Duplicating

X	y
g_1	a
g_2	b

$|M| = \{a, b\}$

$X[\lvert M\rvert/x]$	y	x
$g_1[a/x]$	a	a
$g_1[b/x]$	a	b
$g_2[a/x]$	b	a
$g_2[b/x]$	b	b

Strict and Lax Existential Quantification

- In the lax clause:

$$F : X \to \mathcal{P}(|M|) \setminus \{\emptyset\}$$

  Each assignment may receive several possible values.
- In the strict clause:

$$F : X \to |M|$$

  Each assignment receives exactly one value.
- The distinction matters for logics that are not downward closed.

First-Order Dependence

- First-order dependence atoms have the form

$$\mathrm{dep}(x_1, \ldots, x_n, y)$$

- Their truth is defined exactly like propositional dependence:

$$M, X \vDash \mathrm{dep}(x_1, \ldots, x_n, y)$$

iff for all $g, h \in X$,

$$g(x_i) = h(x_i) \text{ for every } i \leq n \quad \Rightarrow \quad g(y) = h(y)$$

- Thus, inside X, the value of y is determined by the values of $x_1, \ldots, x_n$.

Example: First-Order Dependence

X	x	y	z
g_1	a	b	c
g_2	a	b	d
g_3	e	f	c

- $M, X \vDash \text{dep}(x, y)$.
- g_1 and g_2 agree on x and also agree on y.
- $M, X \nvDash \text{dep}(x, z)$
- g_1 and g_2 agree on x but disagree on z.

Väänänen (2007)

Dependence Logic and Second-Order Logic

- First-order dependence logic is first-order logic plus dependence atoms
- Although the syntax looks first-order, dependence atoms give the logic second-order strength.
- At the level of sentences: $\mathrm{D} \equiv \mathrm{ESO}$
- Here ESO is existential second-order logic: $\exists R_1 \cdots \exists R_n \, \psi$, where ψ is first-order.
- The intuition is that dependence atoms encode the existence of functions governing the team.
- $\forall x \forall z \exists y (\mathrm{dep}(x, y) \wedge P(x, y))$
- $\exists f \, \forall x \forall z (P(x, f(x)))$

Exercise 1: Semantics

Consider the following team T in a model M with domain the set of numbers $\{0, 1, 2, 3\}$:

T	x	y	z
g_1	0	2	3
g_2	0	1	1
g_3	1	2	3
g_4	1	1	0

Determine whether:

(i) $M, T \models \mathrm{dep}(x, y)$
(ii) $M, T \models \mathrm{dep}(z, y)$
(iii) $M, T \models \mathrm{dep}(x, z) \vee \mathrm{dep}(y, z)$
(iv) $M, T \models \exists q(q \neq x \wedge q \neq y \wedge q \neq z)$
(v) $M, T \models \forall q(\mathrm{dep}(q, x)) \vee \forall q(\mathrm{dep}(q, y))$

Consider a dependence atom $\mathrm{dep}(\varnothing, y)$ with an empty list of variables $x_1, \ldots, x_n$. When is $\mathrm{dep}(\varnothing, y)$ true?

Exercise 2: Closure Properties

Classify each property as downward closed, union closed, upward closed, and/or empty-team valid.

(i) $T \vDash p$

(ii) $T \vDash \mathrm{dep}(p, q)$

(iii) T is compatible with p

(iv) $T = \varnothing$

(v) $T \subseteq \{v \mid v(p) = 1\}$

For every failed closure property, give a counterexample.

Exercise 3: A new extension

Let $D = \{a, b\}$ and let $X = \{i_1, i_2\}$ with $i_1(y) = a$ and $i_2(y) = b$.

We extend X with a new variable x. Write (c, d) for the assignment with $y = c$ and $x = d$.

	extension Y	allowed?
Y_1	$\{(a,a),(b,a)\}$	yes
Y_2	$\{(a,a),(b,b)\}$	no
Y_3	$\{(a,a),(b,a),(b,b)\}$	no
Y_4	$\{(a,b),(b,a)\}$	no
Y_5	$\{(a,b),(b,b)\}$	yes

	extension Y	allowed?
Y_6	$\{(a,b),(b,a),(b,b)\}$	no
Y_7	$\{(a,a),(a,b),(b,a)\}$	no
Y_8	$\{(a,a),(a,b),(b,b)\}$	no
Y_9	$\{(a,a),(a,b),(b,a),(b,b)\}$	yes

1. What property distinguishes the admissible extensions from the non-admissible ones?
2. Find a general description of the admissible x-extensions of an arbitrary team X. Your description should have the form $Y = \{i[c/x] \mid \dots\}$. Use this extension to define a new quantifier $\exists^{\mathrm{new}} x$.
3. Can this quantifier be defined using the ordinary lax existential quantifier together with a new atom of the form $\#(x, \vec{y})$?

Exercise 4: Second-order logic and expressible properties

Reformulate the following sentences of dependence logic in ESO, using function quantifiers:

1. $\forall x \forall u \exists y \exists z(\text{dep}(x, y) \wedge \text{dep}(u, z) \wedge (u = x \rightarrow y = z))$
2. $\exists q(\text{dep}(\varnothing, q))$ (Hint: we also allow 0-ary function quantifiers, i.e., constant quantifiers)

What do the following sentences express?

1. $\forall x \forall u \exists y \exists z(\text{dep}(x, y) \wedge \text{dep}(u, z) \wedge (u = x \leftrightarrow y = z))$
2. $\forall x \exists y \exists q(\text{dep}(x, y) \wedge \text{dep}(\varnothing, q) \wedge y \neq q)$

What can we say about a model M that makes the following sentence true?

$$\forall x \forall u \exists y \exists z \exists q(\text{dep}(x, y) \wedge \text{dep}(u, z) \wedge (u = x \leftrightarrow y = z) \wedge \text{dep}(\varnothing, q) \wedge y \neq q)$$