Logic, Data Examples and Learning

Lecture 1/5: Setting the stage & the power and limitations of data examples

Balder ten Cate

Tsinghua University, May 7, 2026

Lost in Translation

Exercise: Please translate the following sentence to linear temporal logic (LTL):

"Trump is going to remain president until the next elections"

Solution (?) $\varphi = \mathrm{Until}(e, \mathrm{G}t)$ where

- t means "*Trump is president*", and
- e mean "*elections are held*".

Lost in Translation

Exercise: Please translate the following sentence to linear temporal logic (LTL):

"Trump is going to remain president until the next elections"

Solution (?) $\varphi = \text{Until}(e, \mathrm{G}t)$ where

- t means "*Trump is president*", and
- e mean "*elections are held*".

Meaning of the LTL operators

- $\mathrm{F}\phi$: at some point in the Future, ϕ will be true.
- $\mathrm{G}\phi$: from now on, ϕ is Going to be true.
- $\text{Until}(\alpha, \beta)$: at some future moment, α becomes true, and until then, β is true.
- $\dots$

Lost in Translation

Exercise: Please translate the following sentence to linear temporal logic (LTL):

"Trump is going to remain president until the next elections"

Solution (?) $\varphi = \text{Until}(e, \mathrm{G}t)$ where

- t means "*Trump is president*", and
- e mean "*elections are held*".

Meaning of the LTL operators

- $\mathrm{F}\phi$: at some point in the Future, ϕ will be true.
- $\mathrm{G}\phi$: from now on, ϕ is Going to be true.
- $\text{Until}(\alpha, \beta)$: at some future moment, α becomes true, and until then, β is true.
- $\ldots$

Example trace:

	0	1	2	3	4
t	1	1	1	0	0
e	0	0	1	0	0

Examples and Logical Specifications

Examples play two complementary roles in logical specification:

- they help explain, debug, and refine formulas that are already written;
- they can also be used as data from which a formula is constructed.

formulas classify examples   examples constrain formulas

The Back-and-Forth View

We will move between two directions:

From Formulas to Examples

Given a formula, find examples that explain what it means, expose borderline cases, or identify it uniquely.

(Today)

From Examples to Formulas

Given labeled examples, construct a fitting formula that is small, useful, and ideally generalizes.

(Lecture 2)

What Counts as an Example?

The example space depends on the logical language and the task.

Setting	Examples
propositional logic	truth assignments
first-order sentences	finite structures
temporal logic	traces

Choosing the right kind of example is part of the theory.

Roadmap

Lecture	Secondary theme
1	Setting the stage; the power and limitations of data examples
2	Fitting algorithms for logical concept classes
3	When fitting concepts generalize to unseen examples
4	Tentative: Active learning algorithms
5	Tentative: Graph neural networks and their connections to logic

Hidden secondary agenda:

- Exposure to basic finite model theory (e.g., homomorphisms), complexity theory (e.g., approximation algorithms), and probability/combinatorics (sample bounds).

Assumed Background

I will assume familiarity with:

- propositional logic and first-order logic: syntax and semantics;
- basic probability theory: probability distributions;
- basic complexity theory: polynomial-time algorithms (NP-hardness).

1. Setting the Stage

Outline for Today

1. Basic definitions
2. Generating "good" examples for concepts
3. Case study: $\mathsf{FO}_{\exists,\wedge,\top}$-formulas

Concept Classes

Definition (Concept class)

A *concept class* is a triple

$$\mathsf{C} = (C, \mathcal{X}, \lambda)$$

where:

- C is the set of concept representations;
- $\mathcal{X}$ is the example space;
- $\lambda : C \to 2^{\mathcal{X}}$ maps each representation to the examples satisfying it.

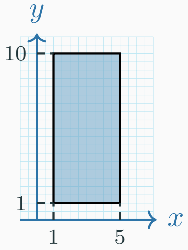
Rectangles

Definition (RECT^{2D})

$$\text{RECT}^{2D} = (C, \mathcal{X}, \lambda)$$

- $C = \{\underline{\text{rect}}(x_1, x_2, y_1, y_2) \mid x_1, x_2, y_1, y_2 \in \mathbb{R}\}$,
- $\mathcal{X} = \mathbb{R}^2$,
- $\lambda(\underline{\text{rect}}(x_1, x_2, y_1, y_2))$ is the set of all $(x, y) \in \mathbb{R}^2$ such that

$$x_1 \leq x \leq x_2 \quad \text{and} \quad y_1 \leq y \leq y_2.$$

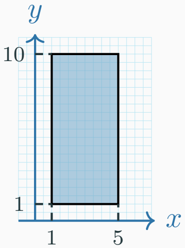


Let

$$c = \underline{\mathsf{rect}}(1, 5, 1, 10).$$

Then

$$(5,5) \in \lambda(c) \quad \text{while} \quad (15,15) \notin \lambda(c).$$



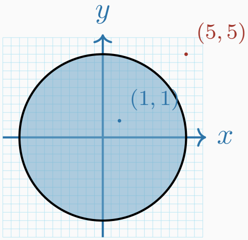
Origin-Centered Disks

Definition (DISK_o^{2D})

$$\mathsf{DISK}_o^{2D} = (C, \mathbb{R}^2, \lambda)$$

- $C = \{\underline{\mathsf{disk}}(r) \mid r \in \mathbb{R}\}$.
- $\lambda(\underline{\mathsf{disk}}(r))$ is the set of all points at distance at most r from the origin:

$$\lambda(\underline{\mathsf{disk}}(r)) = \{(x, y) \in \mathbb{R}^2 \mid x^2 + y^2 \leq r^2\}.$$



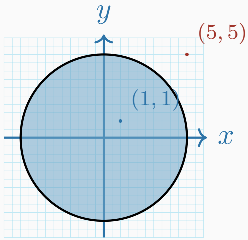
Origin-Centered Disks: Example

Let

$$c = \underline{\text{disk}}(5).$$

Then

$$(1,1) \in \lambda(c) \quad \text{while} \quad (5,5) \notin \lambda(c).$$



Halfspaces

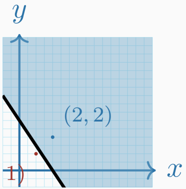
Definition (HALFSPACES^k)

$$\mathsf{HALFSPACES}^k = (C, \mathbb{R}^k, \lambda)$$

- C consists of all linear inequalities of the form

$$a_1x_1 + \cdots + a_kx_k \geq b$$

  with $a_1, \ldots, a_k, b \in \mathbb{R}$.
- $\lambda(\varphi)$ is the set of all tuples satisfying φ.



Halfspaces: Example

For the concept class HALFSPACES^2, let

$$\varphi = 3x_1 + 2x_2 \geq 6.$$

Then

$$(2,2) \in \lambda(\varphi) \quad \text{while} \quad (1,1) \notin \lambda(\varphi).$$

Labeled Examples

Let $\mathsf{C} = (C, \mathcal{X}, \lambda)$ be a concept class.

Definition (Labeled example)

- A *labeled example* is a pair (e, ℓ) where $e \in \mathcal{X}$ and $\ell \in \{+, -\}$.
- A concept c *fits* a positive example $(e, +)$ if $e \in \lambda(c)$.
- A concept c *fits* a negative example $(e, -)$ if $e \notin \lambda(c)$.

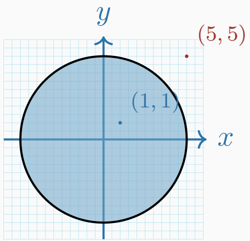
Labeled Examples: Example

Consider the concept class DISK_o^{2D} and let $c = \underline{\mathrm{disk}}(5)$.

Example

The concept c fits the labeled examples

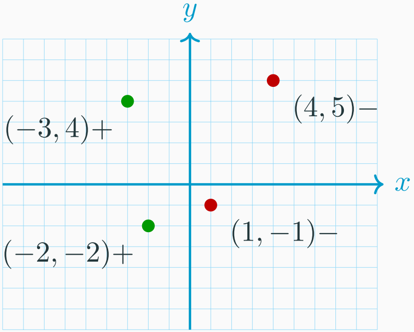
$$((1,1),+) \quad \text{and} \quad ((5,5),-).$$



Quiz: Fitting Geometric Examples

For the following labeled examples,

- is there a fitting concept in RECT^{2D}?
- is there a fitting concept in $\mathrm{HALFSPACES}^{2}$?
- is there a fitting concept in DISK_o^{2D}?



Answer

RECT^{2D}: Yes,

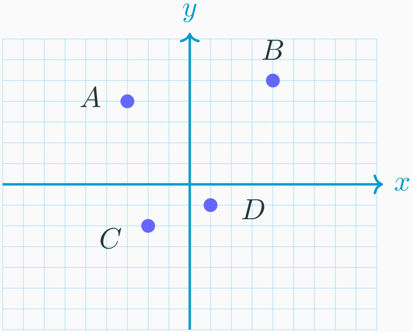
$\mathrm{HALFSPACES}^{2}$: Yes,

DISK_o^{2D}: No

Quiz: Counting Consistent Labelings

How many consistent labelings are there for these 4 points for:

- $\text{RECT}^{2\text{D}}$;
- HALFSPACES^{2};
- $\text{DISK}_o^{2\text{D}}$.



Answer

$\text{RECT}^{2\text{D}}$: 14,

HALFSPACES^{2}: 14,

$\text{DISK}_o^{2\text{D}}$: 5.

Logical Concept Classes

Propositional Logic as a Concept Class

Definition (PL[PROP])

Let PROP be a finite set of propositional variables.

$$\mathsf{PL}[\mathsf{PROP}] = (C, 2^{\mathsf{PROP}}, \lambda)$$

where:

- C is the set of propositional formulas over PROP;
- 2^{PROP} is the set of truth assignments;
- $\lambda(\varphi) = \{t \in 2^{\mathsf{PROP}} \mid t \models \varphi\}$.

Example

For $\mathsf{PROP} = \{p_1, p_2, p_3\}$, the formula $p_2 \wedge p_3$ fits $(\langle 011 \rangle, +)$ and $(\langle 001 \rangle, -)$.

Fragments of Propositional Logic

Definition ($\mathsf{PL}_O[\mathsf{PROP}]$)

Let $O \subseteq \{\wedge, \vee, \neg, \top, \bot\}$.

$$\mathsf{PL}_O[\mathsf{PROP}] = (C, 2^{\mathsf{PROP}}, \lambda)$$

where C contains exactly the formulas built from PROP using connectives from O.

Example

$\mathsf{PL}_{\wedge,\top}[\{p_1, p_2, p_3\}]$ contains $p_1 \wedge p_2$ but not $p_1 \vee p_2$.

Quiz: How Many Fitting Concepts?

For $\mathsf{PROP} = \{p_1, p_2, p_3\}$, consider the examples

$$(\langle 011 \rangle, +) \qquad \text{and} \qquad (\langle 001 \rangle, -).$$

How many non-equivalent fitting concepts are there?

Concept class	Number of fitting concepts
$\mathsf{PL}_{\wedge,\top}[\mathsf{PROP}]$	2
$\mathsf{PL}_{\wedge,\vee,\top,\bot}[\mathsf{PROP}]$	5
$\mathsf{PL}[\mathsf{PROP}]$	64

First-Order Logic as a Concept Class

Definition ($\mathsf{FO}_O[\mathbf{S}]$)

Let $\mathbf{S}$ be a finite relational schema and $O \subseteq \{\wedge, \vee, \neg, \top, \bot, \exists, \forall\}$.

$$\mathsf{FO}_O[\mathbf{S}] = (C, \mathrm{FinStruct}(\mathbf{S}), \lambda)$$

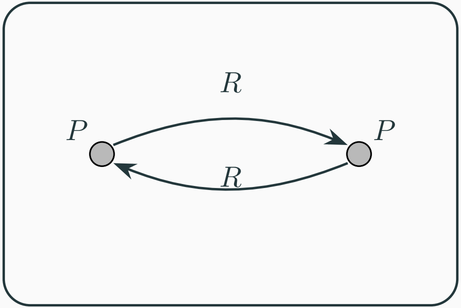
where:

- C is the set of first-order sentences over $\mathbf{S}$ using symbols in O;
- $\mathrm{FinStruct}(\mathbf{S})$ is the set of finite $\mathbf{S}$-structures;
- $\lambda(\varphi) = \{A \mid A \models \varphi\}$.

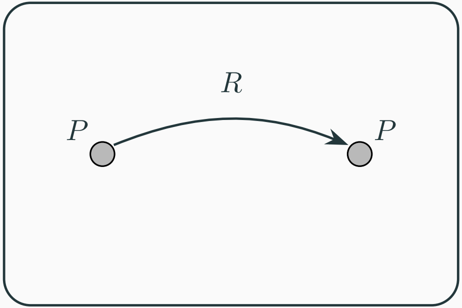
First-Order Example

Let $\mathbf{S} = \{R, P\}$, with R binary and P unary, and let

$$\varphi = \exists x\, P(x) \land \forall x \big(P(x) \to \exists y (R(x, y) \land P(y))\big).$$



positive example for ϕ



negative example for ϕ

Examples So Far

Concept class	Concepts	Examples
RECT^{2D}	axis-aligned rectangles	$\mathbb{R}^2$
DISK_o^{2D}	origin-centered disks	$\mathbb{R}^2$
$\mathsf{HALFSPACES}^k$	linear inequalities	$\mathbb{R}^k$
$\mathsf{PL}_O[\mathsf{PROP}]$	propositional formulas	truth assignments
$\mathsf{FO}_O[\mathbf{S}]$	first-order sentences	finite structures

Monotone Concept Classes

Pre-Orders

Definition (Pre-order)

A *pre-order* on a set X is a binary relation $\leq$ that is:

- reflexive: $x \leq x$ for all $x \in X$;
- transitive: if $x \leq y$ and $y \leq z$, then $x \leq z$.

Upward and Downward Closure

Let $\leq$ be a pre-order on X and let $Y \subseteq X$.

The *upward closure* of Y is

$$Y^{\uparrow} = \{x \in X \mid \exists y \in Y : y \leq x\}$$

and the *downward closure* of Y is

$$Y^{\downarrow} = \{x \in X \mid \exists y \in Y : x \leq y\}$$

We say that Y is *upward closed* if $Y^{\uparrow} = Y$.

Monotone Concept Classes

Definition (Monotone concept class)

Let $\mathsf{C} = (C, \mathcal{X}, \lambda)$ be a concept class and let $\leq$ be a pre-order on $\mathcal{X}$.

The concept class C is *monotone with respect to* $\leq$ if $\lambda(c)$ is upward closed for every $c \in C$.

$\mathsf{PL}_{\wedge,\top}[\mathsf{PROP}]$ is Monotone

For truth assignments $t, t' \in 2^{\mathsf{PROP}}$, define

$$t \leq_b t' \quad \text{iff} \quad \text{every variable true in } t \text{ is true in } t'.$$

Fact

$\mathsf{PL}_{\wedge,\top}[\mathsf{PROP}]$ is monotone with respect to $\leq_b$.

$$\begin{tikzcd}
& 011 \arrow[rr] & & 111 \\
001 \arrow[ur] \arrow[rr] & & 101 \arrow[ur] & \\
& 010 \arrow[uu] \arrow[rr] & & 110 \arrow[uu] \\
000 \arrow[uu] \arrow[ur] \arrow[rr] & & 100 \arrow[uu] \arrow[ur] &
\end{tikzcd}$$

In fact the same holds even for $\mathsf{PL}_{\wedge,\vee,\top,\bot}$.

Full Propositional Logic is Not Monotone

Full $\mathrm{PL}[\mathrm{PROP}]$ is not monotone with respect to $\leq_b$.

Example

Let $\varphi = p_1 \land \neg p_3$.

The assignment $\langle 100 \rangle$ satisfies φ, but $\langle 101 \rangle$ is larger under $\leq_b$ and does not satisfy φ.

DISK_o^{2D} is Monotone

DISK_o^{2D} is monotone with respect to the reverse radial order (distance from the origin):

$$(x, y) \leq_{\mathsf{rad}} (x', y') \quad \text{iff} \quad (x')^2 + (y')^2 \leq x^2 + y^2.$$

Monotonicity Summary

Concept class	Monotone?
$\mathrm{RECT}^{\mathrm{2D}}$	no natural non-trivial order for all rectangles
$\mathrm{DISK}_o^{\mathrm{2D}}$	yes, using distance from the origin
$\mathrm{HALFSPACES}^k$	not in general
$\mathrm{PL}_O[\mathrm{PROP}]$	depends on O
$\mathrm{FO}_O[\mathbf{S}]$	depends on O

From Concepts to Examples

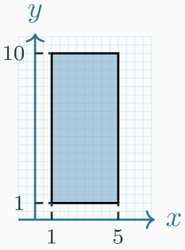
Choosing Examples for Rectangles

For $c = \underline{\text{rect}}(1, 5, 1, 10)$ in $\text{RECT}^{2\text{D}}$, if you could only choose 6 labeled examples, which would you pick?

Answer

There is no single optimal choice of labeled examples. A natural choice is:

- positive examples: two opposing corners;
- negative examples: close to the boundary on all 4 sides.



Choosing Examples for Monotone Monomials

Let $\mathsf{PROP} = \{p_1, p_2, p_3, p_4, p_5\}$ and

$$\varphi = p_2 \wedge p_3 \wedge p_4.$$

A compact set of examples is:

$$(\langle 01110 \rangle, +)$$

and

$$(\langle 00110 \rangle, -), \quad (\langle 01010 \rangle, -), \quad (\langle 01100 \rangle, -).$$

What Makes a Set of Labeled Examples Good?

Starting from a target concept c, there are several natural answers:

1. c is the *only* concept, up to equivalence, that fits it (**unique characterization**)
2. it makes some (specific) learner $\mathcal{L}$ output c (**characteristic sample** relative to L)
3. every other concept c' that fits is ε-close to c (ε-**representative sample**)

Today

In this lecture we focus on *unique characterizations*. The other notions will return later.

Definition (Unique characterization)

Let C $= (C, \mathcal{X}, \lambda)$ be a concept class and let $c \in C$. A set of labeled examples E *uniquely characterizes* c with respect to C if:

1. c fits E;
2. for every $c' \in C$, if c' fits E, then $\lambda(c') = \lambda(c)$.

Unique Characterization Depends on the Class

Let $\mathsf{PROP} = \{p_1, \ldots, p_5\}$.

The examples

$$(\langle 01110 \rangle, +), \quad (\langle 00110 \rangle, -), \quad (\langle 01010 \rangle, -), \quad (\langle 01100 \rangle, -)$$

uniquely characterize $p_2 \wedge p_3 \wedge p_4$ in $\mathsf{PL}_{\wedge,\top}[\mathsf{PROP}]$.

But

They do not uniquely characterize the same formula in $\mathsf{PL}_{\wedge,\vee,\top,\bot}[\mathsf{PROP}]$.

A Monotone Formula Example

Let

$$\varphi = (p_1 \land p_2) \lor (p_3 \land p_4).$$

In $\mathsf{PL}_{\land,\lor,\top,\bot}[\mathsf{PROP}]$, φ is uniquely characterized by:

$$(\langle 11000\rangle, +), \qquad (\langle 00110\rangle, +)$$

and

$$(\langle 01011\rangle, -), \quad (\langle 01101\rangle, -), \quad (\langle 10011\rangle, -), \quad (\langle 10101\rangle, -).$$

The positive examples are minimal satisfying assignments.

The negative examples are maximal falsifying assignments.

How Many Examples Are Needed?

Concept class	Examples needed	Growth
$\mathsf{PL}_{\wedge,\top}[\mathsf{PROP}]$	$\lvert\varphi\rvert + 1$	polynomial
$\mathsf{PL}_{\wedge,\vee,\top,\bot}[\mathsf{PROP}]$	$\lvert\mathrm{cnf}(\varphi)\rvert + \lvert\mathrm{dnf}(\varphi)\rvert$	sometimes exponential
full $\mathsf{PL}[\mathsf{PROP}]$	2^n, where $n = \lvert\mathsf{PROP}\rvert$	always exponential

Getting a Deeper Understanding

Some structural notions explain these bounds:

1. canonical examples;
2. closure under intersection and union;
3. frontiers and finite dualities.

These ideas also apply beyond propositional logic, for example to fragments of FO logic.

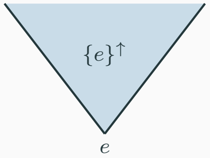
Frontiers and Dualities

Definition (Canonical example)

Let $\mathsf{C} = (C, \mathcal{X}, \lambda, \leq)$ be a monotone concept class. An example $e \in \mathcal{X}$ is a *canonical example* for $c \in C$ if

$$\lambda(c) = \{e\}^{\uparrow}.$$

We also say that c is a canonical concept for e.



Canonical Examples and Concepts

- C *admits canonical examples* if every concept has a canonical example,
- C *admits canonical concepts* if every example has a canonical concept.

Concept class	canonical examples?	canonical concepts?
$\mathsf{PL}_{\wedge,\top}[\mathsf{PROP}]$	yes	yes
$\mathsf{PL}_{\wedge,\vee,\top,\bot}[\mathsf{PROP}]$	not always	yes

A Small Caveat

The presence of constants such as $\top$ matters.

Example

In $\mathrm{PL}_\wedge[\mathrm{PROP}]$, the all-zero assignment need not have a canonical concept, because the required concept would be $\top$.

Closure Under Intersection and Union

Definition (Closure properties)

A concept class $\mathsf{C} = (C, \mathcal{X}, \lambda)$ is:

- closed under intersection if every $\lambda(c_1) \cap \lambda(c_2)$ is represented by some $c \in C$;
- closed under union if every $\lambda(c_1) \cup \lambda(c_2)$ is represented by some $c \in C$.

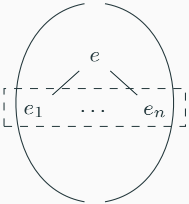
Concept class	intersection	union
$\mathsf{PL}_{\wedge,\top}[\mathsf{PROP}]$	yes	no
$\mathsf{PL}_{\wedge,\vee,\top,\bot}[\mathsf{PROP}]$	yes	yes

Frontiers

Definition (Frontier)

Let $\leq$ be a pre-order on X. A frontier for $e \in X$ is a finite set $F \subseteq X$ such that:

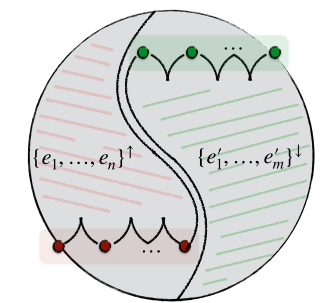
1. $e' < e$ for all $e' \in F$;
2. for every $e'' < e$, there is some $e' \in F$ with $e'' \leq e'$.



Definition (Finite duality)

Let $\leq$ be a pre-order on X. A *finite duality* is a pair of finite sets (F, D) such that

$$F^{\uparrow} \cap D^{\downarrow} = \emptyset \qquad \text{and} \qquad F^{\uparrow} \cup D^{\downarrow} = X.$$



Revisiting $\mathsf{PL}_{\wedge,\top}$

For

$$\varphi = p_2 \wedge p_3 \wedge p_4,$$

the unique characterization in $\mathsf{PL}_{\wedge,\top}[\mathsf{PROP}]$ has:

- positive example $\langle 01110 \rangle$, the canonical example;
- negative examples $\langle 00110 \rangle$, $\langle 01010 \rangle$, $\langle 01100 \rangle$, a frontier for the canonical example.

Revisiting $\mathsf{PL}_{\wedge,\vee,\top,\bot}$

For

$$\varphi = (p_1 \wedge p_2) \vee (p_3 \wedge p_4),$$

the positive examples

$$\langle 11000 \rangle, \quad \langle 00110 \rangle$$

and negative examples

$$\langle 01011 \rangle, \quad \langle 01101 \rangle, \quad \langle 10011 \rangle, \quad \langle 10101 \rangle$$

form a finite duality.

Frontiers Give Unique Characterizations

Theorem

Let $\mathsf{C} = (C, \mathcal{X}, \lambda, \leq)$ *be a monotone concept class that admits canonical examples, and let* $c \in C$. *If the canonical example* e_c *of* c *has frontier* $\{e_1, \ldots, e_n\}$, *then*

$$\{(e_c, +), (e_1, -), \ldots, (e_n, -)\}$$

uniquely characterizes c.

Theorem

Let $\mathsf{C} = (C, \mathcal{X}, \lambda, \leq)$ *be a monotone concept class that admits canonical concepts and is closed under union. Then for every* $c \in C$*, the following are equivalent:*

1. c *is uniquely characterized by a finite set of labeled examples;*
2. *there is a finite duality* (F, D) *such that*

$$F^{\uparrow} = \lambda(c) \qquad \textit{and} \qquad D^{\downarrow} = \mathcal{X} \setminus \lambda(c).$$

Summary

- If a concept class admits canonical examples and canonical concepts, unique characterizations correspond to frontiers. *
- If a concept class admits canonical concepts and is closed under union, unique characterizations correspond to finite dualities.

* assumes that the pre-order has greatest lower bounds.

First-Order Logic and Homomorphisms

Recall: $\mathsf{FO}[\mathbf{S}]$ is the concept class of all first-order sentences over a relational schema $\mathbf{S}$.

Theorem

No sentence in $\mathsf{FO}[\mathbf{S}]$ is uniquely characterizable by a finite set of labeled examples.

If we want finite unique characterizations, we need to move to better behaved fragments of first-order logic.

Why This Already Fails Over the Empty Signature

The negative result already appears when $\mathbf{S} = \emptyset$.

- Finite examples are then just finite sets, so they are determined by their cardinalities.
- For every n, first-order logic can express "the structure has exactly n elements".
- Given any finite labeled sample E for φ, choose n that is not the size of any structure in E. Then $\varphi \vee \exists^{=n}\top$ and $\varphi \wedge \neg\exists^{=n}\top$ both fit E, but are not equivalent.

The Fragment $\mathsf{FO}_{\exists,\wedge,\top}[\mathbf{S}]$

For a relational schema $\mathbf{S}$,

$$\mathsf{FO}_{\exists,\wedge,\top}[\mathbf{S}] = (C, \mathrm{FinStruct}(\mathbf{S}), \lambda)$$

where:

- C contains first-order sentences over $\mathbf{S}$ using only $\exists, \wedge, \top$;
- examples are finite $\mathbf{S}$-structures;
- $\lambda(\varphi) = \{A \mid A \models \varphi\}$.

Question

Is there a natural pre-order with respect to which this concept class is monotone?

Definition (Homomorphism)

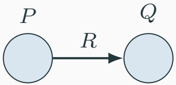
A homomorphism $h : A \to B$ between $\mathbf{S}$-structures maps $\mathrm{Dom}(A)$ to $\mathrm{Dom}(B)$ and preserves:

- constants: $h(c^A) = c^B$;
- relation facts: if $R^A(a_1, \ldots, a_k)$, then $R^B(h(a_1), \ldots, h(a_k))$.

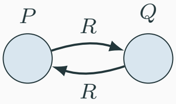
(We don't consider function symbols in this course)

Examples of Homomorphisms

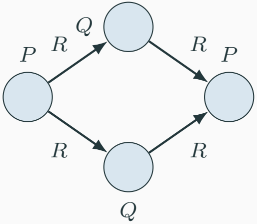
D_1:



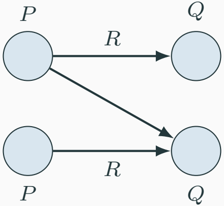
D_2:



D_3:



D_4:



Homomorphisms can *embed a structure into a larger structure* and can *identify elements.*

Homomorphisms are Everywhere

Homomorphisms arise in many areas of math, computer science, and AI:

- database theory;
- the study of constraint satisfaction problems;
- complexity theory;
- machine learning (graph neural networks);
- combinatorial graph theory;
- universal algebra;
- ...

An entire course could be taught just on homomorphisms.

Two Views of Homomorphisms

Homomorphisms can be read in several useful ways, including:

1. as information growth in databases;
2. as constraint satisfaction;

Homomorphism as Constraint Satisfaction

Many natural constraint satisfaction problems (e.g., graph colorability, SAT, Sudoku, …) can be seen as homomorphism testing problems.

Example

A graph G is 3-**colorable** iff there is a homomorphism $G \to \mathrm{K}_3$, where K_3 is the 3-element clique.

- Vertices of K_3 are the colors.
- Edges of G must map to edges of K_3.
- Therefore adjacent vertices receive different colors.

Likewise, a graph G is n-colorable iff there is a homomorphism $G \to \mathrm{K}_n$

The Homomorphism Pre-Order

We write $A \leq_{\text{hom}} B$ if there is a homomorphism $A \to B$.

The relation $\leq_{\text{hom}}$ is a pre-order:

- reflexive: the identity map is a homomorphism $A \to A$;
- transitive: if $A \to B$ and $B \to C$, then $A \to C$.

(It is not generally a partial order, since two non-isomorphic structures can map homomorphically to each other.)

Fact

The concept class $\mathrm{FO}_{\exists,\wedge,\top}[\mathbf{S}]$ is monotone with respect to $\leq_{\text{hom}}$.

(In fact, the same holds for $\mathrm{FO}_{\exists,\wedge,\vee,\top,\bot}[\mathbf{S}]$)

Properties of the Concept Class $\mathsf{FO}_{\exists,\wedge,\top}[\mathbf{S}]$

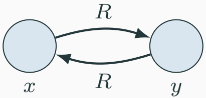
class	intersections	unions	pre-order	can. ex.	can. con.
$\mathsf{PL}_{\wedge,\top}[\mathsf{PROP}]$	yes	no	$\leq_b$	yes	yes
$\mathsf{PL}_{\wedge,\vee,\top,\bot}[\mathsf{PROP}]$	yes	yes	$\leq_b$	no	yes
$\mathsf{FO}_{\exists,\wedge,\top}[\mathbf{S}]$	yes	no	$\leq_{\mathrm{hom}}$	yes	yes

Properties of the Concept Class $\mathsf{FO}_{\exists,\wedge,\top}[\mathbf{S}]$

class	intersections	unions	pre-order	can. ex.	can. con.
$\mathsf{PL}_{\wedge,\top}[\mathsf{PROP}]$	yes	no	$\leq_b$	yes	yes
$\mathsf{PL}_{\wedge,\vee,\top,\bot}[\mathsf{PROP}]$	yes	yes	$\leq_b$	no	yes
$\mathsf{FO}_{\exists,\wedge,\top}[\mathbf{S}]$	yes	no	$\leq_{\mathrm{hom}}$	yes	yes

Canonical example of

$$\varphi = \exists xy\,(R(x,y) \wedge R(y,x)) :$$



Properties of the Concept Class $\mathsf{FO}_{\exists,\wedge,\top}[\mathbf{S}]$

class	intersections	unions	pre-order	can. ex.	can. con.
$\mathsf{PL}_{\wedge,\top}[\mathsf{PROP}]$	yes	no	$\leq_b$	yes	yes
$\mathsf{PL}_{\wedge,\vee,\top,\bot}[\mathsf{PROP}]$	yes	yes	$\leq_b$	no	yes
$\mathsf{FO}_{\exists,\wedge,\top}[\mathbf{S}]$	yes	no	$\leq_{\mathrm{hom}}$	yes	yes

In general:

- the canonical example of φ has the variables of φ as domain elements and the conjuncts of φ as facts;
- if equalities occur, we may need to quotient by them;
- the canonical concept of a structure A has one existential variable per element and one conjunct per fact.

Properties of the Concept Class $\mathsf{FO}_{\exists,\wedge,\top}[\mathbb{S}]$

class	intersections	unions	pre-order	can. ex.	can. con.
$\mathsf{PL}_{\wedge,\top}[\mathsf{PROP}]$	yes	no	$\leq_b$	yes	yes
$\mathsf{PL}_{\wedge,\vee,\top,\bot}[\mathsf{PROP}]$	yes	yes	$\leq_b$	no	yes
$\mathsf{FO}_{\exists,\wedge,\top}[\mathbb{S}]$	yes	no	$\leq_{\mathrm{hom}}$	yes	yes

Question

Can an $\mathsf{FO}_{\exists,\wedge,\top}$-sentence be uniquely characterized by a finite set of labeled examples?

Approach

By studying frontiers in the homomorphism pre-order.

A Structure With No Frontier

Let $\mathbf{S} = \{E\}$ and let

$$A = (\{a\}, \{E(a, a)\})$$

be the one-node graph with a loop.

Theorem (Idea)

A has no frontier with respect to $\leq_{\mathrm{hom}}$.

Why No Frontier?

The proof uses Erdos' theorem:

Theorem (Erdos)

For every $k > 0$ and $m > 0$, there is a graph with chromatic number at least k and girth at least m.

Suppose, for a contradiction, that A has a frontier $F = \{B_1, \ldots, B_n\}$.

- Observation 1: The structures B_i don't contain loops. Therefore, each B_i is k-colorable for $k = |B_i|$ (just give each vertex a different color). In other words, $B_i \to \mathsf{K}_k$.
- Pick any graph G with girth and chromatic number larger than every $|B_i|$.
- Then $B_i \leq_{\mathrm{hom}} A$ and $A \not\leq_{\mathrm{hom}} G$. Therefore, $G \leq_{\mathrm{hom}} B_i$ for some i. But then, by transitivity, $G \to \mathsf{K}_k$ for $k = |B_i|$, contradicting the fact that G has chromatic number larger than n.

Which Structures Have Finite Frontiers?

So the question becomes: which finite structures have a frontier with respect to $\leq_{\mathrm{hom}}$?

The answer is expressed by *c-acyclicity*.

Definition (Incidence graph and c-acyclicity)

The *incidence graph* of a finite structure A is the following bipartite multigraph:

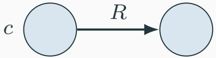
- Vertices: the elements of A and the facts of A.
- Edges: for every occurrence of an element a in a fact f, we add an edge between a and f.

A structure is *c-acyclic* if every cycle in its incidence graph passes through an element named by a constant symbol.

Example

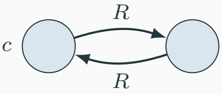
If the schema has no constants, c-acyclicity is the same as ordinary acyclicity.

Examples of c-Acyclicity



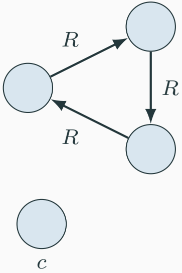
c-acyclic

(in fact acyclic)



c-acyclic

the only cycle goes through c



not c-acyclic

there is a cycle away from the constant

Frontiers and c-Acyclicity

Theorem (Foniok et al. 2006; tC and Dalmau 2022)

For every finite structure A:

1. *if A is c-acyclic, then A has a frontier with respect to $\leq_{\mathrm{hom}}$, computable in polynomial time;*
2. *conversely, if A has a finite frontier, then A is c-acyclic, modulo homomorphic equivalence.*

Putting Everything Together

Let $\phi \in \mathsf{FO}_{\exists,\wedge,\top}[\mathbf{S}]$.

- If the canonical example of ϕ is c-acyclic, then ϕ is uniquely characterizable by a finite set of labeled examples, which can be computed in polynomial time.
- Conversely, ϕ is uniquely characterizable by a finite set of labeled examples if and only ϕ has a c-acyclic canonical example.
- Equivalently, ϕ is uniquely characterizable if and only if it is equivalent to a c-acyclic sentence in the same fragment.

Complexity remark

Testing whether an $\mathsf{FO}_{\exists,\wedge,\top}[\mathbf{S}]$-sentence has a c-acyclic canonical example is decidable and NP-complete.

What’s Next?

Today

Exploit structural properties of concept classes and pre-orders to construct examples in a clever way.

Lecture 2

Exploit the same structural properties to construct fitting concepts from from examples.