Math Logic, Week#IV

Moon's Day

¿Any remaining questions on chapter §§2.0-2.2 (except for Definablity & homework problems)**?**

Definability: In human language, we describe tend to create new words as needed (as well as for other reasons).

(But you might want to look up the Sapir-Wharf hypothesis in linguistics.)

- But in mathematics and computer science, we often investigate what we can do in a fixed language. (For example, in many programming languages, there is no way to discuss the address of a particular variable.)

  If we change the language, we change the topic.
- So we ask "*what can we say in this particular language.*"

Example from Database Theory: In relational algebra — and thus in SQL — given a relation such as *parentOf*, we cannot, in general, define its transitive closure, *ancestorOf*.

(But SQL is often extended to include a transitive closure operator.)

Definition & Examples:

1. **Review a database example:** With relations **student**(stdnt#, name1, name2, name3) and
   took(stdnt#, class, semester, year)
   ask for the names of all students with first name "Adam" who *ever* took "Biol-1001":

$$\exists \text{ st\#} \left(\begin{array}{l} \textbf{student}(\text{st\#, n1, n2, n3}) \ \wedge \ (\text{n1 = Adam}) \ \wedge \\ \quad \exists \text{ smstr } \exists \text{ yr}(\textbf{took}(\text{st\#, Biol1001, smstr, yr})) \end{array} \right).$$

   Since I haven't quantified n1, n2, or n3, all those are *free* — they're uninterpreted here. *In this context* we asking for all triples (n1, n2, n3) that satisfy the formula.

2. Enderton defines a k-ary relation r on a structure $\mathfrak{A}$ to be *first order definable* in $\mathfrak{A}$ if there if a first order formula φ (of the language of the structure) with free variables among $v_1, \ldots, v_k$ where, for $a_1, \ldots, a_k \in |\mathfrak{A}|$,

$$(a_1, \ldots, a_k) \in r \text{ if and only if } \mathfrak{A} \models \varphi[s]$$

   where $s(v_i) = a_i, 1 \leq i \leq k$.
   We typically write that last as $\mathfrak{A} \models \varphi[a_1, \ldots, a_k]$.

3. Suppose we have a structure $\mathfrak{A}$ but an empty language and no equality symbol. Then we can't form any atomic formulas, so we can't form any formulas, and we can't say anything.

   Suppose we have an empty language but do have an equality symbol. We can prove — by induction on formulas — that every formula with just 2 free variables is logically equivalent to one of $v_1 = v_2$, $v_1 \neq v_2$, $\bot$ or $\top$. So four binary relations are definable:

   - $\{(a, b) : a, b \in |\mathfrak{A}|, a \neq b\}$;
   - $\{(a, b) : a, b \in |\mathfrak{A}|, a = b\}$;
   - $|\mathfrak{A}|^2$; and
   - $\emptyset$.

4. In many modern computer languages, we have no way to ask about the memory location of a variable. The computer language *abstracts* from reality, getting us to express some "nature" of the problem without worrying about computer-centric issues. And, *if the compiler is correct*, this is often desirable.

   By contrast, in (¡ancient!) machine-language programming, the programmer had to make all those decisions. One of the first programmers at UC, Prof. Paul Hergot, is quoted as having said that "any programmer worth his salt knows the locations of all his variables."

   But now, if we load the same program twice, it might go into two different sets of memory locations! Since we (often) can't access those memory locations directly, this may make no difference!

   Moral: We need to find ways to show that some information is not visible.

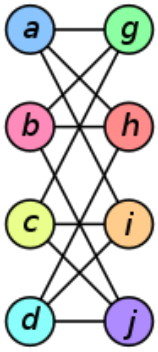
5. **¿How can we show a relation is definable in a structure $\mathfrak{A}$?**

   Easiest: write down a formula that "obviously" defines the relation.

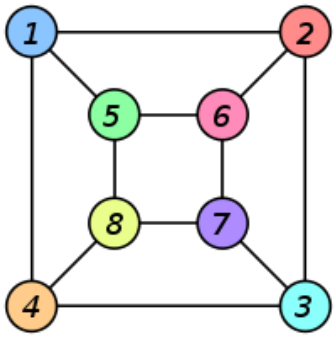
6. **¿But how can we show a relation is not definable?**
   Except with empty languages, that's harder.

7. A graph consists of (1) as the universe, a set of vertices, and (2) an interpretation of a binary relation E.

   Consider the two graphs (copied from Wikipedia,
   `https://en.wikipedia.org/wiki/Graph_isomorphism`)



and



$\mathfrak{A}_1$ $\mathfrak{A}_2$

- The picture and the names of the vertices are irrelevant — they're not part of the structure.
- The function f mapping each vertex of $\mathfrak{A}_1$ to the vertex of the same color in $\mathfrak{A}_2$ is 1-1 and onto.

  and "preserves the structure": for any vertices x, y of $\mathfrak{A}_1$,

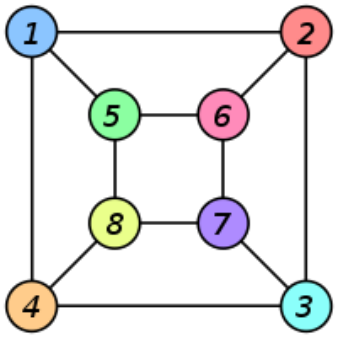
$$(x, y) \in E^{\mathfrak{A}_1} \quad \text{if and only if} \quad (f(x), f(y)) \in E^{\mathfrak{A}_2}.$$

  We say that graphs $\mathfrak{A}_1$ and $\mathfrak{A}_2$ are isomorphic.

  Consequence of "Homomorphism Theorem" For any formula $\varphi(v_1, v_2)$ of the language of graph theory,

$$\mathfrak{A}_1 \models \varphi[a, g] \quad \text{if and only if} \quad \mathfrak{A}_2 \models \varphi[f(a), f(g)].$$

8. Back to:



$\mathfrak{A}_2$

- The function that maps each point 90 degrees clockwise is also an isomorphism.
- An isomorophism from a structure onto itself is called an *automorphism.*
- So any formula $\varphi(v_1)$ satisfied by vertex 2 is also satisfied by vertex 3.
- So any definable subset containing 2 must also contain 3.
- By extending this argument, we find that there are only 2 definable sets of (unary relations on the) nodes:

$$\emptyset \quad \text{and} \quad \{1, 2, 3, 4, 5, 6, 7, 8\}.$$

- But note: there are more complex definable k-ary relations for $k > 1$. Can you identify any?

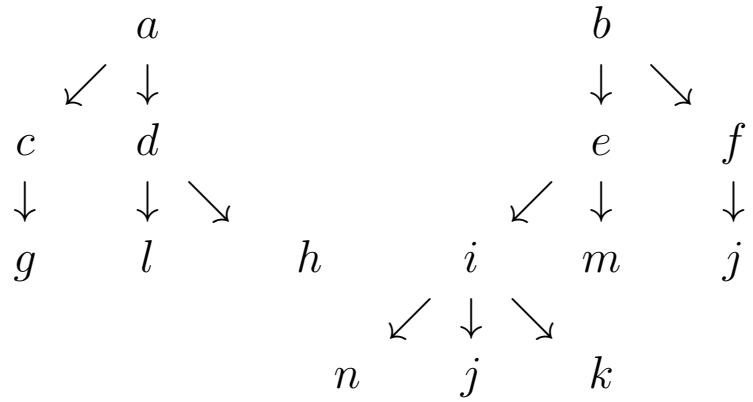
- **Read & work through** the proof of the Homomorphism Theorem — and ask questions as needed.
- **Vocabulary Caution:** Enderton uses the term "*isomorphism*" in an unusual way: he does *not* require that the function be onto (*i.e.*, *surjective*). But he does require *onto*-ness (*i.e.*, *surjectivity*) in defining the terms "*isomorphic to*" and "*automorphim.*"

Wodin's Day

Quick coverage of the notes I sent you on trees & forests: In particular,

- defining ("axiomatizing") the class of trees, and
- in a tree or forest, defining 2 relaions

On this sheet I'll turn my trees and forests upside-down, with the roots at the top, *e.g.*,



And in this case I'll use $x \leq y$ to mean that there is a path *going up* from x to y — so a and b become the *maximal* elements of the 2 trees — but we'll still call them the *roots* of the 2 trees in the forest.

Choice of language: Given my direction on the tres, there are 4 natural languages to put on the structure to describe the relations:

- $\{\leq\}$ — $x \leq y$ if there is a path from x *upward* to y, of length ≥ 0.
- $\{<\}$ — $x < y$ if there is a path from x *upward* to y of length ≥ 1.
- $\{\texttt{isChildOf}\}$ — $\texttt{isChildOf}(x, y)$ if x is the child of y, *i.e.*, x is immediately below y.
- $\{\texttt{isParentOf}\}$ — but that's just an order-of-arguments change from `isChildOf`, so I'll ignore it from here on.

I shall also include $=$ as a logical symbol.

What language we choose affects how we define things.

- Relations $\leq$ and $<$ are easily defined in terms of each other, so that anything we can say in one language we can say in the other.
- Not so with `isChildOf`: we cannot define $\leq$ or $<$ in terms of `isChildOf`.
  large (We can define "*there is a path of length at most k from x to y*" for any fixed integer k.)

Partial ordering axioms (Trees and forests, with the $\leq$ relation, are special kinds of partial orderings)**:**

$$\forall x \forall y \forall z((x \leq y \wedge y \leq z) \rightarrow x \leq z) \quad \wedge \quad \forall x(x \leq x)$$
$$\wedge \quad \forall x \forall y((x \leq y \wedge y \leq x) \rightarrow x = y)$$

Tree axiom: To say each of the trees in the forest is a tree, say the set of all ancestors of any node is linearly ordered by $\leq$. To say linear ordering, we need to add that any 2 elements are *comparable* — one is $\leq$ the other.

$$\forall x \forall y \forall z(\ (x \leq y \wedge x \leq z) \quad \rightarrow \quad (y \leq z \vee z \leq y)\ ).$$

Does thix axiom really describe the property of being a tree?
Either convince yourself that it does or come up with an argument that it does not.

(You might break into 2 cases: the universe is finite, or the universe is infinite. We can't say, in first order logic, that the universe is finite unless we say, further, that the universe has size $\leq k$ for some natural number k. Indeed, in all cases I have heard people discuss infinite trees, they have added extra axioms.)

Roots are maximal elements: Here I'll switch to defining the unary relation `isARoot` in the tree:

$$\forall v_2(v_2 \geq v_1 \rightarrow v_2 = v_1)$$

So to say that the forest consists of a single tree, you can add an axiom saying that:

- There is only 1 root, and
- There is at least 1 root.

  (If the forest has only finitely many nodes, this follows from the other axioms since, in first order logic, we assume the universe is non-empty. You can prove there must be a root by induction on the size of the structure. But over infinite structures modelling the axims it's possible to have no roots! ¿Can you see how to construct an example?)

¿In language $\{\leq\}$, how can we define relation `isChildOf`? (Here we're defining a binary relation on a structure, so our formula will have 2 free varibles, v_1 and v_2:)

$$(v_1 \leq v_2) \ \wedge \ (\neg v_1 = v_2) \ \wedge$$
$$\forall v_3((v_1 \leq v_3 \wedge v_3 \leq v_2) \rightarrow (v_3 = v_1 \vee v_3 = v_2))$$

(Another comment on those beastly infinite structures: An infinte structure could satisfy all the axioms, have all the elements in a single tree, and yet for no node to be a child of another — by the definition above. Can you see why?)

Homework Exercise D: Find an infinite model for the axioms above which is linearly ordered by $\leq$ but has no root and where no node has any children. (You are very familiar with such an example — although you may not have thought of it this way.)

Note on coverage: We're skipping §2.3.

On Frea's Day (Friday), turn in sulutions to:

- Exercise C
- §1.7 ## 2, 4
- §2.1 # 5
- §2.2 # 2

Notes on turned in homework:

- The deadline is soon, so I'm expecting you mostly to write up what you've already done — carefully — and turn that in.
- I accept homework either in class or by email
  (to `John`"dot"`Schlipf`"at"`UC`"dot"`Edu`)
  Emailed solutions must be in .pdf.
  If you hand-write your solutions, please try to be neat and legible! And leave fairly large margins for me to write in.
- I expect homework to be on time — keeping up in this class is vital.
  Homework is due, in paper, *at the start of class*
  or, by email, *mailed at least 5 minutes before* the start of class.
- I shall accept late homeworks on Mon's Day (Monday) with a 34% grade penalty.
  (But do not turn in part of the assignment on Friday and part on Monday. And do not, on Monday, ask to replace what you turned in on Friday.)
- I expect each student to have done her(his) work alone. See my web site notes on plagiarism.
  But, if you did do the homework with another student, give full credit to the other student, and I'll give you each partial credit.
- I welcome solutions by computer word-processing or type-setting *if* you can produce sybols accurately. (LaTeXis wonderful for that. The newest versions of Word are decent.)
- Please put your name on every page you turn in. And if you turn in > 1 page in paper, please *staple* the pages together.
- I'll give you solutions in due time — probably next week sometime.

Freya's Day

For Moon's Day (Monday): Read at least once through §2.4 up to Deductions and Metatheorems — that is, up at least throiugh the first 5 lines of page 116. (I'll also talk about it next week, but I shall expect you to have at least a general idea of what's going on.)

Still open problems:

whichever of the following we don't go through today:

- Exercise B
- §1.7 ## 3, 8
- §2.1 ## 1
- §2.2 ## 1, 2, 6, 9, 11, 12, 14, 15

Homework Presentations:

- Finish §1.5 #2 **¿Does anyone still claim it?**

  (If nobody has it today, I'll declare the problem "dead" — no longer open for claims — and I'll write up a solution.)
- Exercise B
- §1.7 #3 **&** #8
- §2.1 #1
- §2.2 #1 **&** #2 **&** #6 **&** #9