1 Lecture 01.11

We began with the question: Is there a numerically diverse group of Philadelphians? (We call a group of people numerically diverse if no two people in the group have the same number of friends in the group - we assume groups are of size at least two and that friendship is always mutual.) We demonstrated that the answer is no by an application of

Principle 1 *The Pigeonhole Principle: If you distribute m pigeons into n pigeonholes and $m \geq n+1$, then some hole contains at least two pigeons.*

We argued as follows. Suppose we have a group $G = \{1, \ldots, n\}$ of n people (we use numerals to name the people for privacy concerns). For brevity, let's write p_{ij} to signify that i is a friend of j. We assume friendship is symmetric, that is, if p_{ij}, then p_{ji}, for all $i, j \in G$, and irreflexive, that is, it is not the case that p_{ii}, for all $i \in G$. Let's write $f(i)$ for the number of friends of i, that is, the number of j such that p_{ji}. Since friendship is irreflexive, the possible values of f are the n numbers $0, 1, \ldots, n-1$. We are thinking of these values as the pigeonholes for application of the principle 1 and the members of G as being placed in these holes by f. We want to argue that the value of f must agree on at least two members of G. But so far, since we have n members of G and n pigeonholes into which they are sorted by f, we may not yet draw that conclusion via principle 1. But now we consider the question, "can f really take all the values from 0 to $n-1$?" In particular, can it take on both the value 0 and the value $n-1$. We argue that the answer is no. Suppose that there is some i with $f(i) = 0$, that is, for every j, it is not the case that p_{ji}. Then, by symmetry, for every j, it is not the case that p_{ij}. So, if i has no friends, then the maximum number of friends of any j is $n-2$, that is, f cannot take on the value $n-1$. Thus, the possible values of f are the $n-1$ numbers $0, \ldots, n-2$. But now, by principle 1, we can conclude that f takes on the same value for at least two members of G. This concludes our argument that there cannot be a numerically diverse group of Philadelphians.

We mentioned that the course will explore relationships and that love differs from friendship in that there are narcissists (so we can't assume the relation is irreflexive) and is not always requited (so we can't assume the relationship is symmetric). We observed that this difference between friendship and love allows the existence of numerically diverse groups of lovers, that is, groups where each person in the group loves a different number of people in the group. Consider, for example, a group of four people, call them $1, 2, 3, 4$, and suppose that 1 doesn't love anyone, 2 loves 1, 3 loves both 1 and 2, and 4 loves all of $1, 2$, and 3, and that this exhausts all the love among our group of four. We achieve numerical diversity at the sacrifice of requital.

How many different patterns of love might obtain among a group of four people, again call them $1, 2, 3, 4$. Now, we decided to recycle the sentence letters and use p_{ij} to signify the statement that i loves j; we noted that 16 sentence letters would be required to record all the relevant statements. Since each pattern of love among $1, 2, 3, 4$ is determined by assigning one of the truth values

$\top$ or $\bot$ to each of these 16 sentence letters, we concluded that the number of such patterns is 2^{16}. Why? Because there are two assignments to p_{11} and for each of these, there are two assignments to p_{12}, and thus $2 \cdot 2 = 2^2$ assignments to them jointly (this observation is given the exalted title, "The Product Rule"). Thus, by iterating application of the product rule another fourteen times, we arrive at the conclusion that there are 2^{16} possible truth assignments to the 16 sentence letters. We marveled at the fact that there are as many as 65,536 different potential love-scenarios at a table for four.

On the other hand, we considered how tame friendship is as compared with love, in terms of the number of possible friendship-scenarios. In virtue of the fact that friendship is symmetric and irreflexive, a friendship-scenario is determined by assigning one of the truth values $\top$ or $\bot$ to each of the 6 sentence letters p_{ij}, for $1 \leq i < j \leq 4$. Hence, there are only $2^6 = 64$ possible patterns of friendship among the group of four, less than 1/1000 of the number of potential love-scenarios.

2 Lecture 01.18[1]

2.1 Introduction

Today, we began our systematic treatment of truth-functional logic. Throughout the course we will see a few different systems for formalizing statements. Each consists of a formal language to represent statements, and a way to interpret the meaning of statements in that language. Truth-functional logic is the simplest of these systems we will learn.

2.2 Components of Truth Functional Logic

1. Language
   (a) sentence letters
   (b) connectives
2. Interpretation
   (a) A function that assigns $\top$ or $\bot$ (true or false) to each sentence letter, called a **truth-assignment**
   (b) Fixed **truth-functional semantics** for each connective

Sentence letters such as $p, q, r, \ldots$ schematize statements (in natural language) which are true or false, and **connectives** such as $\wedge, \vee, \neg, \supset, \ldots$ are used to combine sentence letters into compound schemata.

2.3 Definitions of some truth-functional connectives

Consider using the sentence letter p_{ij} to schematize the statement "i loves j," where $1 \leq i, j, \leq 4$. For example, p_{11} schematizes the statement "1 loves 1", or briefly, "1 is a narcissist."

Suppose we wish to schematize the following statements using those sentence letters:

1. all of 1, 2, 3, and 4 are narcissists;
2. none of 1, 2, 3, and 4 are narcissists;
3. at least one of 1, 2, 3, and 4 is a narcissist;
4. an odd number of 1, 2, 3, and 4 are narcissists.

In order to do so, we introduce the following truth-functional connectives. For each connective, we display its truth-functional interpretation via a table indicating the truth value of the compound schema as a function of the truth values of its components.

[1]This lecture was given by Joel McCarthy and Owain West. Grace Zhang wrote these notes.

- Conjunction (and):

p	q	$(p \wedge q)$
$\top$	$\top$	$\top$
$\top$	$\bot$	$\bot$
$\bot$	$\top$	$\bot$
$\bot$	$\bot$	$\bot$

- Negation (not):

p	$\neg p$
$\top$	$\bot$
$\bot$	$\top$

- Inclusive Disjunction (or)

p	q	$(p \vee q)$
$\top$	$\top$	$\top$
$\top$	$\bot$	$\top$
$\bot$	$\top$	$\top$
$\bot$	$\bot$	$\bot$

- Exclusive Disjunction (exclusive or, xor)

p	q	$(p \oplus q)$
$\top$	$\top$	$\bot$
$\top$	$\bot$	$\top$
$\bot$	$\top$	$\top$
$\bot$	$\bot$	$\bot$

  Note that the truth/falsity of a compound schema is completely determined by, or purely a function of, the truth/falsity of its components. Hence, the term "truth-functional logic."

We can now schematize conditions 1 – 4 in the above example as follows.

S1: $((p_{11} \wedge p_{22}) \wedge p_{33}) \wedge p_{44}$

S2: $((\neg p_{11} \wedge \neg p_{22}) \wedge \neg p_{33}) \wedge \neg p_{44}$

S3: $((p_{11} \vee p_{22}) \vee p_{33}) \vee p_{44}$

S4: $((p_{11} \oplus p_{22}) \oplus p_{33}) \oplus p_{44}$

The first three are quite straightforward to verify; the fourth we will prove later in Proposition 1.

2.4 Truth assignments

Given a truth-functional schema like $((p \wedge q) \vee r)$, we cannot determine whether the schema is true or false unless we know whether p, q, and r are true or false. That is, any schema requires a truth-assignment to its sentence letters before it can be evaluated.

Definition 1 (Truth-assignment) *Let X be a set of sentence letters. A truth-assignment A for X is a mapping which associates with each sentence letter $q \in X$ one of the two truth values $\top$ or $\bot$; we write $A(q)$ for the value that A associates to q.*

Suppose S is a truth-functional schema such that every sentence letter with an occurrence in S is a member of X. We say a truth assignment A for X satisfies such a schema S ($A \models S$) if and only if S receives the value $\top$ relative to the truth assignment A.

Example 1 *Take the schema $S = ((p \wedge q) \vee r)$, with truth assignment A such that $A(p) = \top$, $A(q) = \bot$, and $A(r) = \bot$, we have that S receives the value $\bot$. In other words A does not satisfy S. ($A \not\models S$).*

2.5 An Inductive Proof

Proposition 1 *For every $n \geq 2$ and every set $X = \{q_1, \ldots, q_n\}$ of n distinct sentence letters, a truth assignment A for X satisfies the schema*

$$S_n : (\ldots(q_1 \oplus q_2) \ldots \oplus q_n)$$

if and only if A assigns an odd number of the sentence letters in X the value $\top$.

Proof: We proved the proposition by induction on n.

- Basis: Examination of the truth table for $\oplus$ suffices to establish the proposition for the case $n = 2$.

- Induction Step: Suppose the proposition holds for a number $k \geq 2$, that is, for every truth assignment A for $\{q_1, \ldots, q_k\}$, $A \models S_k$ if and only if A assigns an odd number of the sentence letters in $\{q_1, \ldots, q_k\}$ the value $\top$; this is our induction hypothesis. We proceed to show that the proposition also holds for $k + 1$. Let A' be an assignment to the sentence letters $\{q_1, \ldots, q_{k+1}\}$ and let A be its restriction to $\{q_1, \ldots, q_k\}$. We consider two cases. First, suppose that $A'(q_{k+1}) = \top$. In this case, $A' \models S_{k+1}$ if and only if $A \not\models S_k$ if and only if (by our induction hypothesis) A assigns an even number of the sentence letters $\{q_1, \ldots, q_k\}$ the value $\top$. Hence, if $A'(q_{k+1}) = \top$, then $A' \models S_{k+1}$ if and only if A' assigns an odd number of the sentence letters in $\{q_1, \ldots, q_{k+1}\}$ the value $\top$. On the other hand, suppose that $A'(q_{k+1}) = \bot$. In this case, $A' \models S_{k+1}$ if and only if $A \models S_k$ if and only if (by our induction hypothesis) A assigns an odd number of the sentence letters $\{q_1, \ldots, q_k\}$ the value $\top$. Hence, if $A'(q_{k+1}) = \bot$, then $A' \models S_{k+1}$ if and only if A' assigns an odd number of the sentence letters in $\{q_1, \ldots, q_{k+1}\}$ the value $\top$. This concludes the proof, since either $A'(q_{k+1}) = \top$ or $A'(q_{k+1}) = \bot$.

3 Lecture 01.23

3.1 The material conditional

We returned to our potential lovers and restricted attention to just two of them, 1 and 2. We asked how we could express the statement that all love is requited among these two. The natural mode of expression is: if 1 loves 2, then 2 loves 1, and if 2 loves 1, then 1 loves 2. In order to render this directly, we introduced the

- Material Conditional

p	q	$p \supset q$
$\top$	$\top$	$\top$
$\top$	$\bot$	$\bot$
$\bot$	$\top$	$\top$
$\bot$	$\bot$	$\top$

Now, using the sentence letter $p_{11}, p_{12}, p_{21}, p_{22}$ as earlier interpreted, we can express the happy state that all love among 1 and 2 is requited by the schema

$$R : (p_{12} \supset p_{21}) \wedge (p_{21} \supset p_{12}).$$

We asked in how many of the possible love scenarios among 1 and 2 is all love requited, and we computed that the answer is eight out of a total of sixteen such scenarios, by determining how many truth assignments to the sentence letters $p_{11}, p_{12}, p_{21}, p_{22}$ satisfy the schema R.

We discussed generalized conditionals as a route to motivating the truth-functional interpretation of the conditional offered above. We agreed that the statement “if an integer is divisible by six, then it is divisible by three,” is true, and thence that each of the following statements, which are instances of this general statement, are true.

- “If twelve is divisible by six, then twelve is divisible by three.”
- “If three is divisible by six, then three is divisible by three.”
- “If two is divisible by six, then two is divisible by three.”

Therefore, if the conditional involved is to be understood truth-functionally, then its interpretation must satisfy the conditions imposed by the first, third, and fourth rows of the truth-table above. On the other hand, the falsity of the conditional “if twelve is divisible by six, then twelve is divisible by seven,” mandates the condition imposed by the second row of the truth-table above.

3.2 The centrality of satisfaction

We emphasized that the satisfaction relation is the fundamental semantic relation, it is where language and the world meet; in the case to hand, language consists of truth-functional schemata and the possible worlds they describe are truth assignments to sentence letters. As the course progresses, we will encounter more textured representations of the world (relational structures) and richer languages to describe them (monadic and polyadic quantification theory). We now define some of the central notions of truth-functional logic in terms of satisfaction. These definitions will generalize directly to the more textured structures and richer languages we encounter later.

Definition 2 *For the following definitions, we suppose that S and T are truth-functional schemata and that A ranges over truth assignments to sets of sentence letters which include all those that occur in either S or T.*

- *S implies T if and only if for every truth assignment A, if $A \models S$, then $A \models T$.*
- *S is equivalent to T if and only if S implies T and T implies S.*
- *S is satisfiable if and only if for some A, $A \models S$.*
- *S is valid if and only if every truth assignment satisfies S.*

3.3 Examples of equivalence and the material biconditional[2]

We noted various equivalences, for example,

- $p \oplus q$ is equivalent to $q \oplus p$ (commutativity of exclusive disjunction)
- $(p \oplus q) \oplus r$ is equivalent to $p \oplus (q \oplus r)$ (associativity of exclusive disjunction).

We noted that both conjunction and inclusive disjunction are also commutative and associative, whereas the material conditional is neither. We encouraged the audience to think of examples of (binary) truth-functional connectives which are commutative but not associative, and associative but not commutative.

We introduced one further connective $\equiv$, the material biconditional. We specified its truth-functional interpretation by indicating that $p \equiv q$ is truth-functionally equivalent to both $(p \supset q) \wedge (q \supset p)$ and $\neg(p \oplus q)$.

3.4 Propositions as a heuristic

It is sometimes useful to think of a schema S as expressing a proposition, to whit, the set of truth assignments A that satisfy S; of course, this needs to be relativized to a collection of sentence letters X which includes all those occurring in S. We suggested the notation:

$$\mathbb{P}_X(S) = \{A \mid A \text{ is a truth assignment for } X \text{ and } A \models S\}.$$

When we use this notation without the subscript X, we assume A is a truth assignment for exactly the set of sentence letters with occurrences in S.

[2]This section was omitted from Monday's lecture, but is worth reading nonetheless.

3.5 Expressive completeness

We explored the expressive power of truth-functional logic. In the last section, we suggested using the notion of the proposition expressed by a schema as an intuitive vehicle for pursuing this investigation. Since the semantical correlate of a truth-functional schema is a set of truth assignments to some finite set of sentence letters, we can frame the question of the *expressive completeness of truth-functional logic* in terms of propositions. Let X be a non-empty finite set of sentence letters. We deploy the notation: $\mathbb{A}(X)$ for the set of truth assignments to the sentence letters X, and $\mathbb{S}(X)$ for the set of truth-functional schemata compounded from sentence letters all of which are members of X. We provided the following inductive definition of $\mathbb{S}(X)$.

Definition 3 *Let X be a nonempty finite set of sentence letters. $\mathbb{S}(X)$ is the smallest set $\mathbb{U}$ (in the sense of the subset relation) satisfying the following conditions.*

- $X \subseteq \mathbb{U}$.
- *If σ and τ are strings over the finite alphabet $X \cup \{), (, \neg, \supset, \equiv, \vee, \wedge, \oplus\}$, and $\sigma, \tau \in \mathbb{U}$, then each of the strings $\neg\sigma, (\sigma \supset \tau), (\sigma \equiv \tau), (\sigma \vee \tau), (\sigma \wedge \tau), (\sigma \oplus \tau)$ belong to $\mathbb{U}$.*[3]

If $\mathfrak{P} \subseteq \mathbb{A}(X)$, we call $\mathfrak{P}$ a *proposition over* X. Let X be a non-empty finite set of sentence letters and let $\mathfrak{P}$ be a proposition over X. Is there a schema $S \in \mathbb{S}(X)$ such that $\mathbb{P}_X(S) = \mathfrak{P}$? In other words, is truth-functional logic *expressively complete*? We will answer this question on Wednesday.

We briefly discussed how many propositions there are over a fixed finite set of sentence letters. Since this, and related questions, will bulk large in Wednesday's lecture, I've included this discussion in the preview of our next class meeting.

[3]Here "$(\sigma \supset \tau)$" denotes the string with the initial symbol "(" concatenated with the string denoted by σ concatenated with the symbol "$\supset$" concatenated with the string denoted by τ and with terminal symbol ")", and likewise in all the other cases.

4 Lecture 01.25

4.1 The expressive completeness theorem

Theorem 1 (Expressive Completeness of Truth-functional Logic) *Let X be a non-empty finite set of sentence letters and let $\mathfrak{P}$ be a proposition over X. There is a schema $S \in \mathbb{S}(X)$ such that $\mathbb{P}_X(S) = \mathfrak{P}$.*

For the proof of Theorem 1, the following terminology and lemma will be useful.

Definition 4 *Let X be a non-empty finite set of sentence letters and let $S \in \mathbb{S}_X$.*

- *S is a* literal *over X just in case $S = p$ or $S = \neg p$, for some $p \in X$.*
- *S is a* term *over X just in case S is a conjunction of literals over X (we allow conjunctions of length 1).*
- *S is in disjunctive normal form over X if and only if S is a disjunction of terms over X (we allow disjunctions of length 1).*

If Λ is a set of literals over X we write $\bigwedge \Lambda$ to abbreviate a term which is formed as a conjunction of the literals in Λ. Similarly, if Γ is a set of terms over X we write $\bigvee \Gamma$ to abbreviate a schema in disjunctive normal form which is formed as a disjunction of the terms in Γ.

Lemma 1 *Let X be a non-empty finite set of sentence letters. For every $A \in \mathbb{A}(X)$ there is a schema T_A which is a term over X such that for every $A' \in \mathbb{A}(X)$*

$$A' \models T_A \quad \textit{if and only if} \quad A' = A.$$

Proof: Let X be a finite set of sentence letters and suppose $A \in \mathbb{A}(X)$. For each $p \in X$, let $l_p = p$, if $A \models p$, and let $l_p = \neg p$, if $A \not\models p$. Let $\Lambda = \{l_p \mid p \in X\}$ and let $T_A = \bigwedge \Lambda$. It is easy to verify that for every $A' \in \mathbb{A}(X)$, $A' \models T_A$ if and only if $A' = A$. ■

Proof of Theorem 1: Fix a finite non-empty set of sentence letters X and suppose $\mathfrak{P}$ is a proposition over X. If $\mathfrak{P} = \emptyset$, then pick $p \in X$ and note that $\mathbb{P}_X(p \wedge \neg p) = \mathfrak{P}$. Otherwise, for each $A \in \mathfrak{P}$, choose a term T_A, as guaranteed to exist by Lemma 1, such that for every $A' \in \mathbb{A}(X)$, $A' \models T_A$ if and only if $A' = A$. Let $\Gamma = \{T_A \mid A \in \mathfrak{P}\}$ and let $S = \bigvee \Gamma$. It is easy to verify that $\mathbb{P}_X(S) = \mathfrak{P}$. ■

Corollary 1 *Every truth-functional schema is equivalent to a schema in disjunctive normal form.*

4.2 The power of a truth-functional schema: definition and examples

We will introduce the following useful terminology.

Definition 5 *All schemata are drawn from $\mathbb{S}(X)$ for a fixed non-empty finite set of sentence letters X.*

- *A list of truth-functional schemata is succinct if and only if no two schemata on the list are equivalent.*
- *A truth-functional schema implies a list of schemata if and only if it implies every schema on the list.*
- *The power of a truth-functional schema is the length of a longest succinct list of schemata it implies.*

Examples For concreteness, we considered $X = \{p, q, r\}$. What is the length of a longest succinct list of truth-functional schemata over X? We arrived at the answer by proving an *upper bound* and a *lower bound* on this length.

- Upper bound: It is easy to verify that schemata S and S' are equivalent if and only if $\mathbb{P}(S) = \mathbb{P}(S')$. Hence, the length of a succinct list of schemata cannot exceed the number of propositions over X, that is, the number of subsets of the set $\mathbb{A}(X)$. The size of X is 3, so the size of $\mathbb{A}(X)$ is 2^3, since determining a truth assignment to X involves three binary choices. By the same reasoning, the number of propositions over X is 2^{2^3}, since determining a proposition involves deciding, for each of the 2^3 truth assignments, whether to include or omit it. Hence, the length of the longest succinct list is no more than 256.
- Lower bound: By Theorem 1, for every proposition over X, there is a schema expressing it. Since schemata expressing distinct propositions are not equivalent, it follows at once that there is a succinct list of schemata of length 256.

We proceeded to compute the power, as defined above, of an exemplary schema; let's do $p \wedge (q \vee r)$ here. Note that a schema S implies a schema S' if and only if $\mathbb{P}(S) \subseteq \mathbb{P}(S')$. Thus, the power of S is the number of sets Z satisfying the condition:

$$\mathbb{P}(S) \subseteq Z \subseteq \mathbb{A}(X). \tag{1}$$

The size of $\mathfrak{P} = \mathbb{P}(p \wedge (q \vee r))$ is 3, so the size of $\mathbb{A}(X) - \mathfrak{P} = 5$. It follows at once that $2^5 = 32$ sets Z satisfy condition (1); hence, the power of $p \wedge (q \vee r)$ is 32.

We went on to list the numbers which are powers of truth-functional schemata over $X = \{p, q, r\}$.

- First note that for every $S, S' \in \mathbb{S}(X)$ the power of S = the power of S' if and only if $|\mathbb{P}_X(S)| = |\mathbb{P}_X(S')|$, where we use $|U|$ to denote the number of members of the finite set U.
- In particular, if $\mathfrak{P} = \mathbb{P}_X(S)$, then the power of $S = 2^{(8-|\mathfrak{P}|)}$.
- It follows at once that for each $S \in \mathbb{S}(X)$, the power of $S = 2^i$, for some $0 \leq i \leq 8$.

More generally, suppose Y is a finite set of sentence letters with $|Y| = n$. In this case

- $|\mathbb{A}(Y)| = 2^n$, and
- for each $S \in \mathbb{S}(Y)$, if $\mathfrak{P} = \mathbb{P}_Y(S)$, then the power of $S = 2^{(2^n-|\mathfrak{P}|)}$.

4.3 A question to ponder

We ended by posing a question: What is the length of a longest succinct list of truth-functional schemata over X each of which has power 32?

5 Lecture 01.30

5.1 Pondering our question

We will take up the question on which we ended Wednesday's class meeting. Let $X = \{p, q, r\}$. What is the length of a longest succinct list of truth-functional schemata over X each of which has power 32? It follows from the considerations we advanced last time that a schema has power 32 if and only if exactly three truth assignments satisfy it. Hence the length of a longest such succinct list is exactly the number of subsets of size three contained in a set of size eight.

5.2 Counting Selections

This led to an interlude on permutations and combinations: how many ways can we select k members of a set of size n? There is an ambiguity here: are we counting modes of selection, which involve the order of choices, or collections of members selected, where the order of selection is irrelevant. Once we recognize the ambiguity, we can proceed to count both. We introduced notation for each: $(n)_k$ for the number of ordered sequences of k distinct elements that can be drawn from a set of size n and $\binom{n}{k}$ for the number of subsets of size k that are included in set of size n. To evaluate $(n)_k$ we argued as follows. Suppose we think of counting the ways n students could fill a row of length k in a lecture hall. Let's suppose the seats are labelled $1, 2, \ldots, k$. There are n choices for the student to fill seat 1; once that seat is filled, there are $n - 1$ choices for the student to fill seat 2; and so on until there are $(n-k)+1$ choices for the student to fill seat k. Hence, by the product rule, there are $n \cdot (n - 1) \cdots ((n - k) + 1)$ ways of filling all k seats, that is, $(n)_k = n \cdot (n - 1) \cdots ((n - k) + 1)$. Now that we have counted the number of ordered sequences, we can see how to count the number of subsets. By the same reasoning, each subset of size k appears as the content of $k \cdot (k-1) \cdots 2 \cdot 1$ ordered sequences of length k; this number is called *k factorial* and is often abbreviated as $k!$. Hence,

$$\binom{n}{k} = \frac{(n)_k}{k!}.$$

Observe that

$$(n)_k = \frac{n!}{(n-k)!}$$

from which it follows that

$$\binom{n}{k} = \frac{n!}{k!\,(n-k)!}.$$

This last formulation makes transparent a symmetry in the values of $\binom{n}{k}$, namely, for every k between 0 and n, $\binom{n}{k} = \binom{n}{n-k}$. This accords nicely with the observation that complementation induces a one-one correspondence between the subsets of size k and the subsets of size $(n - k)$ that can be selected from a set of size n. Note also that it determines in a non-arbitrary way that the value of $0!$ is 1.

Let's not forget how this all began. The length of the longest succinct list of schemata with power 32 is $\binom{8}{3} = 56$.

5.3 The length of an "implicational anti-chain"

We actually used our new found ability to count selections to answer a different question: Is there a sequence of seventy schemata $S_1, \ldots, S_{70} \in \mathbb{S}(X)$ such that for every $1 \leq i \neq j \leq 70$, S_i does *not* imply S_j? Such a sequence of schemata is called an *implicational anti-chain* (of length 70). As observed earlier, a schema $S \in \mathbb{S}(X)$ implies a schema $T \in \mathbb{S}(X)$ if and only if $\mathbb{P}_X(S) \subseteq \mathbb{P}_X(T)$. It follows that the answer to our question about an implicational anti-chain of length seventy will be the same as the answer to the following question about an anti-chain of length seventy with respect to the subset relation: Is there a list of seventy subsets of $\mathbb{A}(X)$, $P_1, \ldots, P_n$, such that for every $1 \leq i \neq j \leq 70$, P_i is *not* a subset of P_j? We noted that if two finite sets, P and Q, have the same number of members, and P is not equal to Q, then P is not a subset of Q and Q is not a subset of P. Therefore, is there are seventy distinct subsets of $\mathbb{A}(X)$ all of the same size, then the answer to our question is yes. Since $\mathbb{A}(X)$ has eight members, a positive answer to our question followed immediately by evaluating

$$\binom{8}{4} = \frac{8 \cdot 7 \cdot 6 \cdot 5}{4 \cdot 3 \cdot 2 \cdot 1} = 70.$$

Note that our argument merely shows that there is an implicational anti-chain of length 70; it does not establish that there is no longer implicational anti-chain consisting of schemata in $\mathbb{S}(X)$. This is, indeed, true, but a more sophisticated argument is required to establish this result. For those interested in the matter, I will post a suitable reference containing a proof of Sperner's Theorem on the Canvas site (Van Lint and Wilson, *A course in combinatorics*, Chapter 6: Dilworth's theorem and extremal set theory).

5.4 Truth-functional Satisfiablity: Is there an efficient decision procedure?

We observed that the finitary character of the semantics for truth-functional logic immediately yields an algorithm to decide the satisfiability of schemata of truth-functional logic. In particular, suppose $S \in \mathbb{S}(X)$ for some finite set of sentence letters X. Note first that for each truth-assignment $A \in \mathbb{A}(X)$ there is a simple and efficient algorithm, call it M, to determine whether $A \models S$. Thus, in order to test the satisfiability of S, we need only list $\mathbb{A}(X)$ in some canonical order $A_1, \ldots, A_{2^{|X|}}$ and use M to test whether the successive A_i satisfy S. Of course, this algorithm is not efficient, in the sense that it's running time is potentially exponential in the length of its input. The question whether there is an efficient algorithm to decide the satisfiability of truth-functional schemata is generally regarded as one of the most significant open mathematical problems of our time – for further information visit:

http://www.claymath.org/millennium-problems/p-vs-np-problem.

6 Lecture 02.01

6.1 Monadic Quantification Theory

6.1.1 Sub-sentential logical structure: monadic predicates

We then initiated our study of monadic quantification theory. Statements have significant logical form beyond the structure that can be exhibited in terms of truth-functional compounding. For example, the conjunction of the first two statements below implies, but does not truth-functionally imply, the third.

- All collies are mortal.
- Lassie is a collie.
- Lassie is mortal.

In order to analyze this example, we considered the following statements.

- Lassie is a collie.
- Scout is a collie.
- Rin-Tin-Tin is a collie.

These statements share the *monadic predicate* "◯ is a collie." Monadic predicates, unlike statements, are not true or false; rather, they are *true of* some objects and *false of* other objects. For example, "◯ is a prime number" is true of 2,3,5 and 7, and false of all even numbers greater than 2.

6.1.2 The extension of a monadic predicate

The *extension* of a monadic predicate is the collection of objects of which it is true. The extension of the monadic predicate "◯ is an even number" is the set $\{2, 4, 6, \ldots\}$. The extension of the monadic predicate "◯ is an even prime number" is the set $\{2\}$. The extension of the monadic predicate "◯ is an even prime number greater than 2" is the empty set. Distinct monadic predicates may have the same extension. For example, the extension of the predicate "◯ is a warm–blooded reptile" is also the empty set as is the extension of the predicate "◯ is a collie weighing more than 300 kilograms." We say that monadic predicates with the same extension are *coextensive*. We will focus on statements whose truth depends only on the extensions of the monadic predicates which occur in them. We call such sentential contexts in which interchange of coextensive predicates preserves truth–value *extensional*. Our focus on extensional contexts is the natural continuation of our earlier focus on truth–functional contexts.

6.1.3 Open sentences and the use of variables

Consider again the argument above. Intuitively, the validity of this argument does not depend on the particular name "Lassie" being used; it would be equally valid with any name in place of "Lassie." This generality may be brought out by the use of variables in place of particular names. We will form new expressions called *open sentences* by putting variables "$x, y, z, \ldots$" for the placeholders in monadic predicates. Open sentences are not statements. They are true or false with respect to assignments of values to the variables they contain. For example, the open sentence "x is an even number" is true with respect to the assignment of 16 to "x" and false with respect to the assignment of 17 to "x" and false with respect to the assignment of Lassie to "x."

6.1.4 Truth-functional compounding of open sentences

We may form compounds of open sentences using truth–functional connectives. For example, the following open sentences are truth–functionally complex.

- If x is divisible by six, then x is divisible by three.
- x is a collie and it is not the case that x weighs more than 300 kg.

We may use our prior understanding of the truth–functional connectives to determine the truth–values of such open sentences with respect to particular assignments of values to their variables.

6.1.5 Existential Quantification

We proceeded to introduce the existential quantifier. Consider the statement, "there is an even prime number." We render this statement as the application of the existential quantifier to the open sentence,

- x is an even number $\wedge$ x is a prime number, thus
- $(\exists x)(x$ is an even number $\wedge$ x is a prime number).

This last sentence is true just in case there is an assignment of some object to the variable x with respect to which the preceding open sentence is true.

6.1.6 Free and bound occurrences of variables

Consider again the example the example above.

- x is an even number $\wedge$ x is a prime number
- $(\exists x)(x$ is an even number $\wedge$ x is a prime number)

As noted, the first of these sentences is not simply true or false, it is true or false with respect to an assignment to the variable "x"; we say in this instance that the occurrences of the variable "x" are *free* in this sentence. On the other hand, the occurrences of the variable x are *bound* by the existential quantifier in the second sentence; this sentence is true or false independent of any assignment to the variable x. Note that a variable may have both free and bound occurrences within a single sentence:

- $(\exists x)$(x is an even number) $\wedge$ (x is a prime number);

and may have occurrences bound by distinct quantifiers:

- $(\exists x)$(x is an even number) $\wedge$ $(\exists x)$(x is a prime number).

6.1.7 Universal Quantification

Next we consider the use of the universal quantifier. We can render the statement

- all numbers are even or odd

as

- $(\forall x)$ [(x is an even number) or (x is an odd number)].

The last statement is true, just in case whatever integer is assigned to the variable x satisfies the open statement within the square brackets. Here we see the contextual determination of a *universe of discourse* – when we say "all numbers" in this context, we intend that the variable of quantification range over all integers and not, for example, all complex numbers.

6.1.8 Monadic Schemata

As we did in the case of truth-functional logic, we will introduce a schematic language for monadic quantificational logic. We specify the following categories of monadic schemata.

- A *one variable open schema* is a truth functional compound of expressions such as
  $Fx, Gx, Hx, \ldots$.
- A *simple monadic schema* is the existential or universal quantification of a one variable open schema with variable of quantification x.
- A *pure monadic schema* is a truth functional compound of simple monadic schemata.

6.1.9 Structures as interpretations of monadic schemata

We introduce *structures* as interpretations of monadic schemata. These play the role that truth-assignments played in the context of truth-functional logic. In order to specify a structure A for a schema S we need to

- specify a nonempty set U^A, the universe of A;

- specify sets $F^A, G^A, \ldots$ each of which is a subset of U^A as the extensions of the monadic predicate letters which occur in S;
- specify an element $a \in U^A$ to assign to the variable x, if x occurs free in S.

When the variable x has no free occurrences in the schema S, we write $A \models S$ as shorthand for "the schema S is true in the structure A," alternatively "the structure A satisfies the schema S." Otherwise, we write $A \models S[a]$ as shorthand for "the structure A satisfies the schema S relative to the assignment of a to the variable x."

6.1.10 Validity, satisfiability, implication, and equivalence

We extend the notions of validity, satisfiability, implication, and equivalence to monadic quantificational schemata.

- A monadic schema S is *valid* if and only if for every structure $A, A \models S$.
- A monadic schema S is *satisiable* if and only if for some structure $A, A \models S$.
- A monadic schema S *implies* a monadic schema T if and only if for every structure A, if $A \models S$, then $A \models T$.
- Monadic schemata S and T are equivalent if and only if S implies T, and T implies S.

6.1.11 Counting the number of structures with fixed universe of discourse that satisfy a schema

We discussed how to count the number of structures with a fixed universe of discourse that satisfy a given schema. We asked, how many structures with universe of discourse $U = \{1, 2, 3, 4, 5, 6\}$ interpreting the monadic predicate letters F and G satisfy the schema

$$S : \quad (\forall x)(Fx \supset Gx).$$

We observed that a structure A satisfies S if and only if $F^A \subseteq G^A$. So we need to determine the number, call it n, of pairs of subsets Y, Z of U with $Y \subseteq Z$. By using what we learned earlier about binomial coefficients, we see that

$$n = \sum_{i=0}^{i=6} \binom{6}{i} 2^i = \sum_{i=0}^{i=6} \binom{6}{i} 2^i \cdot 1^{6-i} = (2+1)^6 = 3^6.$$

The next to last equality is justified by the celebrated *Binomial Theorem.* For those of us with no taste for binomial coefficients, we will discuss a much simpler and direct combinatorial argument for the conclusion that $n = 3^6$.

6.1.12 Element Types

Consider the following four one variable open schemata; we will call them (element) types.

- $T_1(x) : Fx \wedge Gx$
- $T_2(x) : Fx \wedge \neg Gx$
- $T_3(x) : \neg Fx \wedge Gx$
- $T_4(x) : \neg Fx \wedge \neg Gx$

Note that a structure A satisfies the schema S if and only if it contains no element satisfying the type T_2. Since a structure is determined by the type of each of its elements, there are as many structures with universe U satisfying S as there are ways of sorting the members of U into the three remaining types. For each of the six members of U, there are three types into which it could be sorted, so by the product rule, the number of structures satisfying S is 3^6.

6.1.13 Counting counterexamples to an alleged implication

If R and R^* are monadic schemata we say that a structure A is a *counterexample* to the claim that R implies R^* if and only if $A \models R$ and $A \not\models R^*$. We continued with the preceding example and counted the number of counterexamples to the claim that the schema S implies the schema

$$T : \qquad (\forall x)(Gx \supset Fx).$$

Again, we suppose that our structures have universe of discourse U and interpret exactly the monadic predicate letters F and G. If a structure A satisfies both S and T, then $F^A = G^A$. Hence, of the 3^6 structures satisfying S, the number that also satisfy T is 2^6, that is, the number of subsets of U, assigned within a single structure to both F and G. So the number of counterexamples to the claim that S implies T is $3^6 - 2^6$.

7 Lecture 02.06

The central occupation of this week's classes will be an approach to establishing the decidability of satisfiability of pure monadic schemata complementary to that developed in sections 25 and 26 of *Deductive Logic*. Our approach introduces notions that we will elaborate further, when we turn to study polyadic quantificational logic.

7.1 Three views of structures

As a warm-up to the main event, we noted that we now have three (equivalent) ways of viewing structures, each of which may contribute a useful perspective, depending on the problem to hand. These are

- the Canonical View, which consists of specifying the universe of discourse and extensions for each of the (finitely many) predicate letters in play,
- the Types View, which consists of specifying a universe of discourse and sorting it into types, that is, maximally specific descriptions that can be framed in terms of the predicate letters in play, and
- the Venn View, which pictures the extensions of all the predicate letters in play as intersecting regions contained in a rectangle that represents the universe of discourse.

7.2 The small model theorem

We will prove the following *Small Model Theorem* for monadic logic; the decidability of satisfiability of pure monadic schemata is a corollary to this result.

Theorem 2 *Let S be a pure monadic schema containing occurrences of at most n distinct monadic predicate letters. If S is satisfiable then there is a structure A of size at most 2^n such that $A \models S$.*

7.3 Monadic similarity

The proof of Theorem 2 rests on the following lemma. In order to state the lemma, we need to introduce some new concepts. Suppose without loss of generality that we restrict our attention to monadic schemata in which only the predicate letters F and G occur. We say that two structures A and B are *monadically similar* if and only if they satisfy exactly the same pure monadic schemata. We explore a sufficient condition for the monadic similarity of structures.

7.4 Homomorphisms

A function h is a mapping from one set, called the *domain* of h to another set (it may be the same set), called the *range* of h. For every element a of the domain of h we write "$h(a)$" to denote the element of the range of h to which it is mapped. We sometimes call $h(a)$ the h *image* of a or the *image* of a under h. We sometiems use the notation

$$h : X \longrightarrow Y$$

to indicate that h is a function with domain X and range Y. If $h : X \longrightarrow Y$ we say that h is *onto* if and only if for every $b \in Y$ there is an $a \in X$ such that $h(a) = b$. In this case, we will also say that h is *surjective*.

Let A and B be structures. We call h a *homomorphism from A onto B* just in case h is an onto function with domain U^A and range U^B satisfying the following condition: for every monadic predicate letter P and every $m \in U^A$,

$$m \in P^A \quad \text{if and only if} \quad h(m) \in P^B.$$

If there is a homomorphism from A onto B, we say that B is a *surjective homomorphic image* of A.

7.5 Examples

We illustrated the above notions with some examples. Consider the following structures.

$$\begin{array}{ll} A: & U^A = \{n \mid n \text{ is a positive integer.}\} \\ & F^A = \{n \mid n \text{ is an even positive integer.}\} \\ & G^A = \{n \mid n \text{ is a prime positive integer.}\} \end{array}$$

$$\begin{array}{ll} B: & U^B = \{n \mid n \text{ is a positive integer.}\} \\ & F^B = \{n \mid n \text{ is an odd positive integer.}\} \\ & G^B = \{n \mid n \text{ is a prime positive integer.}\} \end{array}$$

We observed that though A and B have the same regions occupied in their respective Venn diagrams, and thus realize the same types, there is no homomorphism from A onto B, nor is there a homomorphism from B onto A. We will shortly see that A and B have a common surjective homomorphic image, that is, there is a structure C such that there is a homomorphism from A onto C and a homomorphism from B onto C.

8 Lecture 02.08

8.1 Homomorphisms and monadic similarity: the central lemma

The next lemma provides a useful sufficient condition for monadic similarity.

Lemma 2 *Let A and B be structures. If there is a homomorphism from A onto B, then A is monadically similar to B.*

Proof: Let A and B be structures and suppose that h is a homomorphism of A onto B. It suffices to show that for every simple monadic schema S,

$$A \models S \text{ if and only if } B \models S,$$

since every pure monadic schema is a truth-functional compound of simple monadic schemata.

We begin by observing that for every $c \in U^A$ and every one variable open schema S, A makes S true with respect to the assignment of c to "x," if and only if B makes S true with respect to the assignment of $h(c)$ to "x." This follows immediately from the fact that h is a homomorphism.

Consider the simple schema S and suppose that S is the existential quantification of the the one variable open schema T (the case of universal quantification is treated similarly). Suppose $A \models S$. Then, for some $c \in U^A$, A makes T true with respect to the assignment of c to "x." It follows that B makes T true with respect to the assignment of $h(c)$ to "x." Hence, $B \models S$.

Conversely, suppose $B \models S$. Then, for some $c \in U^B$, B makes T true with respect to the assignment of c to "x." Since h is surjective, there is a $d \in U^A$ with $h(d) = c$. It follows at once that A makes T true with respect to the assignment of d to "x." Hence, $A \models S$. ■

8.2 Types and monadic similarity

We recall our discussion of element types:

- $T_1(x) : Fx \land Gx$
- $T_2(x) : Fx \land \neg Gx$
- $T_3(x) : \neg Fx \land Gx$
- $T_4(x) : \neg Fx \land \neg Gx$

We say that a structure *realizes* a given type T_i just in case it makes the existential simple schema $(\exists x)T_i$ true.

Example 2 *The following structure realizes all four of the types listed above.*

$$A : U^A = \{1, 2, 3, 4\}, F^A = \{1, 3\}, G^A = \{1, 2\}$$

Moreover, the 14 proper substructures of A realize exactly the fourteen proper nonempty subsets of the types listed above.

Lemma 2 yields a useful necessary and sufficient condition for monadic similarity.

Lemma 3 *A and B realize the same types if and only if they are monadically similar.*

Proof: If A and B realize the same types, then there is a single structure C which is a surjective homomorphic image of both A and B. Therefore, by our earlier result, A is monadically similar to C and B is monadically similar to C. It follows at once that A is monadically similar to B. The reverse implication follows immediately from the fact realization of a type is expressed by a pure monadic schema. ■

8.3 The small model theorem and the decidability of satisfiability

Theorem 2 is an immediate corollary to Lemma 3.

Proof (of Theorem 2): It follows at once from Lemma 3 and Example 2, that there is a collection X of 15 structures each of size ≤ 4 such that for any pure monadic schema S involving only the predicate letters "F" and "G," if S is satisfiable, then there is a structure $A \in X$ such that $A \models S$. More generally, there is a collection X of $2^{(2^n)} - 1$ structures each of size $\leq 2^n$ such that for any pure monadic schema S involving only the predicate letters "F_1,"… "F_n," if S is satisfiable, then there is a structure $A \in X$ such that $A \models S$. ■

Corollary 2 *There is a decision procedure to determine whether a pure monadic schema is satisfiable.*

8.4 The small model theorem and the decidability of satisfiability: an elaboration

We elaborated the proof of the Small Model Theorem. Again, we focused on the case of schemata involving only the monadic predicate letters F and G. We drew pictures, in "Types View", of 15 structures $A_1, \ldots, A_{15}$ each with universe of discourse included in $\{1, 2, 3, 4\}$ and no two with the same universe of discourse. Recall the element types:

- $T_1(x) : Fx \wedge Gx$
- $T_2(x) : Fx \wedge \neg Gx$
- $T_3(x) : \neg Fx \wedge Gx$
- $T_4(x) : \neg Fx \wedge \neg Gx$

We constructed the structures A_i by letting j realize the type $T_j(x)$ for each $j \in U^{A_i}$. Let A be an arbitrary structure. It is clear from our construction that there is an i such that A_i realizes exactly the same types as A. Moreover, since

A_i has exactly one element realizing any type that it realizes, A_i is a surjective homomorphic image of A. It follows at once from the result of our last class that A is monadically similar to A_i, that is, they satisfy exactly the same set of pure monadic schemata. Having thus concluded the proof, we turned to deriving a list of useful corollaries.

Corollary 3 *1. For every schema S, if S is satisfiable, then there is an $1 \leq i \leq 15$ such that $A_i \models S$.*

2. *There is an algorithmic decision procedure to determine whether a schema S is satisfiable.*

3. *Schema S implies schema T if and only if*

$$\{i \mid A_i \models S \text{ and } 1 \leq i \leq 15\} \subseteq \{i \mid A_i \models T \text{ and } 1 \leq i \leq 15\}.$$

4. *Schemata S and T are equivalent if and only if*

$$\{i \mid A_i \models S \text{ and } 1 \leq i \leq 15\} = \{i \mid A_i \models T \text{ and } 1 \leq i \leq 15\}.$$

9 Lecture 02.15

9.1 The expressive power of monadic quantification theory

With these results in hand, we proceeded to analyze the expressive power of monadic schemata. Recall the notions deployed in Problem Set 2, but now upgraded to apply to monadic schemata.

- A list of pure monadic schemata is *succinct* if and only if no two schemata on the list are equivalent.
- A pure monadic schema *implies a list of schemata* if and only if it implies every schema on the list.
- The *power* of a pure monadic schema is the length of a longest succinct list of pure monadic schemata it implies.

We continued to focus on the vocabulary consisting of the monadic predicate letters F and G and answered the following questions.

Question 1 *What is the length of a longest succinct list of pure monadic schemata (in the vocabulary consisting of just the monadic predicate letters F and G)?*

Answer: It follows immediately from Corollary 3, part (4) that the length of a longest such list is 2^{15}, since a schema is determined, up to equivalence, by which of the structures $A_1, \dots, A_{15}$ satisfy it.

Question 2 *For which numbers n is there a schema S whose power is n?*

Answer: It follows from Corollary 3, parts (3) and (4), that the power of a schema S is determined by the size j of $\{i \mid A_i \models S \text{ and } 1 \leq i \leq 15\}$, in particular, the power of S is 2^{15-j}; for pure schemata S, j may be any number between 0 and 15. This answers Question 2.

Definition 6
- *If X is a finite set, we write $|X|$ for the number of members of X.*
- *If S is a schema, we write $\mathsf{mod}(S, n)$ for the set of structures A such that $A \models S$ and $U^A = \{1, \dots, n\}$.*

Question 3 *What is the length of a longest succinct list of pure schemata S such that $\mathsf{mod}(S, 4) = 4$?*

Answer: Let $\mathbb{V} = \{A \mid U^A = \{1, 2, 3, 4\}\}$. Recall that $A \approx_M B$ if and only if for all pure monadic schemata S, $A \models S$ if and only $B \models S$. For $A \in \mathbb{V}$, let $\hat{A} = \{B \in \mathbb{V} \mid B \approx_M A\}$. In order to answer the question, it suffices to determine the size of $\hat{A}$ for each $A \in \mathbb{V}$. First, note that the size of $\hat{A}$ is determined by the number of types realized by A. We computed these sizes:

- If A realizes exactly 1 type, then the size of $\hat{A}$ is 1. There are $\binom{4}{1}$ structures in $\mathbb{V}$ satisfying exactly 1 type.
- If A realizes exactly 2 types, then the size of $\hat{A}$ is $2^4 - 2$. There are $\binom{4}{2}$ structures in $\mathbb{V}$ satisfying exactly 2 types.
- If A realizes exactly 3 types, then the size of $\hat{A}$ is $\binom{4}{2} \cdot 3!$. There are $\binom{4}{3}$ structures in $\mathbb{V}$ satisfying exactly 3 types.
- If A realizes exactly 4 types, then the size of $\hat{A}$ is $4!$. There are $\binom{4}{4}$ structures in $\mathbb{V}$ satisfying exactly 4 types.

It is now easy to see that the answer to Question 3 is 1; in particular, one such list consists of the single schema

$$(\forall x)(Fx \wedge Gx) \vee (\forall x)(Fx \wedge \neg Gx) \vee (\forall x)(\neg Fx \wedge Gx) \vee (\forall x)(\neg Fx \wedge \neg Gx).$$

10 Lecture 02.20

10.1 Polyadic predicates and their extensions

We will commence our study of polyadic quantification theory. This topic will remain our focus through the end of the Term. As opposed to truth-functional and monadic logic which, as we've seen, are of limited expressive power, polyadic quantification theory allows for faithful schematization of vast tracts of scientific discourse. But we begin, not with science, but with literature.

Consider the sentences

- Romeo loves Juliet.
- Someone loves Juliet.
- Romeo loves someone.

The first sentence implies the second and the third sentence. We can schematize the second, by making use of the monadic predicate "◯ loves Juliet" thus

$$(\exists x)(x \text{ loves Juliet}).$$

And we can schematize the third, by making use of the monadic predicate "Romeo loves ◯" thus

$$(\exists x)(\text{Romeo loves } x).$$

But if we wish to schematize the sentence "someone loves someone," which is also implied by the first sentence above, we need to expand our resources to include *dyadic predicates*.

- $\boxed{1}$ loves $\boxed{2}$
- $\langle \text{Romeo}, \text{Juliet} \rangle$ is in the extension of "$\boxed{1}$ loves $\boxed{2}$."
- $(\exists x)(\exists y)(x \text{ loves } y)$

The extension of a dyadic predicate is a set of *ordered* pairs.

- $\langle 45, 47 \rangle$ is in the extension of "$\boxed{1} \leq \boxed{2}$."
- $\langle 45, 47 \rangle$ is not in the extension of "$\boxed{2} \leq \boxed{1}$."
- $\langle 47, 45 \rangle$ is in the extension of "$\boxed{2} \leq \boxed{1}$."

Similarly, the extension of a triadic predicate, such as

"$\boxed{1}$ is further from $\boxed{2}$ than it is from $\boxed{3}$,"

is a set of ordered triples.

10.2 Quantifier alternation

Consider the following statements involving alternation of quantifiers.

- Everyone loves someone (or other).

$$S_1 : \quad (\forall x)(\exists y)(x \text{ loves } y).$$

- There is someone whom everyone loves.

$$S_2 : \quad (\exists y)(\forall x)(x \text{ loves } y).$$

- Everyone is loved by someone.

$$S_3 : \quad (\forall y)(\exists x)(x \text{ loves } y).$$

- Someone loves everyone.

$$S_4 : \quad (\exists x)(\forall y)(x \text{ loves } y).$$

The second statement implies the first, and the fourth implies the third. We gave counterexamples to show that no other implications obtain. Consider the following three structures A, B, C.

Structure	Universe	Extension of L
A	$\{a, b\}$	$\{\langle a, a\rangle, \langle b, b\rangle\}$
B	$\{a, b\}$	$\{\langle b, b\rangle, \langle a, b\rangle\}$
C	$\{a, b\}$	$\{\langle b, b\rangle, \langle b, a\rangle\}$

Note that $A \models S_1$ and $A \models S_3$, while $A \not\models S_2$ and $A \not\models S_4$, from which it follows, by definition, that S_1 does not imply S_2, nor does S_3 imply S_4. Moreover $B \models S_2$, but $B \not\models S_3$, and $C \models S_4$, but $C \not\models S_1$; thus S_2 does not imply S_3, and S_4 does not imply S_1. Failure of the remaining (non-trivial) implications now follows. For example, S_1 does not imply S_4, for otherwise, since S_2 implies S_1, and S_4 implies S_3, it would follow that S_2 implies S_3, to which B is a counterexample. We summarize the results of this discussion in the following matrix $\langle a_{ij} \mid 1 \leq i, j \leq 4\rangle$, where $a_{ij} = 1$ if and only if the schema in the i-th row implies the schema in the j-th column.

S_i implies S_j	S_1	S_2	S_3	S_4
S_1	1	0	0	0
S_2	1	1	0	0
S_3	0	0	1	0
S_4	0	0	1	1

10.3 Scope ambiguity

We proceeded to explore "scope ambiguities." Consider the statement, "everybody loves a lover." We observed that "x is a lover" can be schematized as $(\exists y)Lxy$, and corresponding to the two readings, "everybody loves someone who is a lover", and "if someone is a lover, then everybody loves her" we have the respective schematizations:

- $$(\forall z)(\exists x)((\exists y)Lxy \wedge Lzx), \text{ versus}$$
- $$(\forall x)((\exists y)Lxy \supset (\forall z)Lzx).$$

We observed that a structure A satisfies the second schema if and only if either L^A is empty or $L^A = U^A \times U^A$, the cartesian product of the universe of A with itself. On the other hand, if a structure B satisfies the first schema, then L^B is non-empty; moreover, if B consists of a pair of requiting lovers at least one of whom is not a narcissist, B satisfies the first, but not the second, schema. Thus, neither disambiguation of the original sentence implies the other.

11 Lecture 02.22

11.1 Some properties of binary relations

We went on to discuss several important properties of relations.

- L^A is *reflexive* if and only if
$$A \models (\forall x)Lxx.$$
- L^A is *irreflexive* if and only if
$$A \models (\forall x)\neg Lxx.$$
- L^A is *symmetric* if and only if
$$A \models (\forall x)(\forall y)(Lxy \supset Lyx).$$
- L^A is *asymmetric* if and only if
$$A \models (\forall x)(\forall y)(Lxy \supset \neg Lyx).$$
- L^A is *transitive* if and only if
$$A \models (\forall x)(\forall y)(\forall z)(Lxy \supset (Lyz \supset Lxz)).$$
- A is a *simple graph* if and only if L^A is irreflexive and symmetric.

11.2 Identity

We continued our discussion of the expressive power of polyadic quantification theory. We started by introducing a new logical dyadic predicate, identity, which allows us to "put the quant into quantification." The identity relation "=" has a uniform interpretation over all structures A namely $=^A$ is equal to $\{\langle a, a\rangle \mid a \in U^A\}$. Since the interpretation of the identity relation is uniform, we omit mention of it when we specify structures.

11.3 Numerical quantifiers

By making use of the identity relation, we can introduce, for each integer $k \geq 1$, the quantifiers "there are at least k x's such that $S(x)$", "there are at most k x's such that $S(x)$", and "there are exactly k x's such that $S(x)$" as follows.

$$\begin{array}{ll}
(\exists^{k\leq} x)S(x): & (\exists x_1)\ldots(\exists x_k)(\bigwedge_{1\leq i<j\leq k} x_i \neq x_j \wedge \bigwedge_{1\leq i\leq k} S(x_i)) \\
(\exists^{\leq k} x)S(x): & \neg(\exists^{k+1\leq} x)S(x) \\
(\exists^{=k} x)S(x): & (\exists^{\leq k} x)S(x) \wedge (\exists^{k\leq} x)S(x)
\end{array}$$

Let's use $|X|$ to denote the number of members of a set X. In order to clarify the import of these quantifiers we introduced the notion of the *set defined by a one variable open schema $S(x)$ in a structure A* (written $S[A]$):

$$S[A] = \{a \in U^A \mid A \models S[x|a]\}.$$

That is, $S[A]$ is the set of members of U^A that satisfy $S(x)$ in A. Observe that $A \models (\exists^{k\leq} x)S(x)$ if and only if $k \leq |S[A]|$, and similarly for the other two newly introduced quantifiers. We proceeded to explore the use of these quantifiers to define regular simple graphs.

11.4 Regular graphs

Recall that a *graph* is structure that interprets a single dyadic predicate letter "L" (these are sometimes also called directed graphs to emphasize that the edges have directionality), and we declared that, unless otherwise clearly stated, we will restrict our attention for (at least) this lecture and the next to structures that are graphs. A graph A is *simple* if and only if L^A is both irreflexive and symmetric. We introduced the abbreviation SG for the conjunction of the schemata expressing irreflexivity and symmetry, which we abbreviated as Irr and Sym, respectively.

Suppose A is a simple graph and $a \in U^A$. The *neighborhood of a in A* is $\{b \in U^A \mid \langle a, b\rangle \in L^A\}$ and the *degree of a* is $|\{b \in U^A \mid \langle a, b\rangle \in L^A\}|$. That is, the degree of a node a in a simple graph A is the number of neighbors of a in A, equivalently, the number of edges incident with a in A. A simple graph is *k-regular* if and only if all nodes of the graph have degree k. We can schematize this condition, using the dyadic predicate L for the edge relation, as

$$(\forall y)(\exists^{=k} x)Lyx.$$

We discussed the collections of 1-regular and 2-regular simple graphs. We noted that every 1-regular graph consists of a set of independent edges, and that a *finite* 2-regular graph consists of a collection of independent simple cycles, that is, graphs that may be drawn in the plane as a disjoint finite collection of disjoint polygons. We observed that the bi-infinite simple chain is also 2-regular and that polygons and bi-infinite chains exhaust the possible connected components of 2-regular graphs.

11.5 Counting graphs

We proceeded to count graphs with a fixed universe of discourse. We defined

$$\mathsf{mod}(S, n) = \{A \mid A \models S \text{ and } U^A = \{1, \ldots, n\}\}.$$

Note that for every structure A, $A \models (\forall x)x = x$, thus $\mathsf{mod}((\forall x)x = x, n)$ is the set of all graphs with universe of discourse $\{1, \ldots, n\}$. We counted the number of graphs A with $U^A = \{1, 2, 3, 4\}$ ($= |\mathsf{mod}((\forall x)x = x, 4)|$) as follows. We noted that any such graph is determined by choosing which of the sixteen possible edges from i to j to draw, where $1 \leq i \leq 4$ and $1 \leq j \leq 4$; that is, a graph with this universe of discourse is determined by 16 binary choices, so, by the product rule, there are 2^{16} such graphs. We noted that analogous reasoning leads to the conclusion that there are 2^{n^2} graphs with universe of discourse $\{1, \ldots, n\}$. And similarly, since a simple graph with universe of discourse $\{1, \ldots, n\}$ is determined by making a choice from a collection of $\binom{n}{2}$ possible *undirected* edges, there are $2^{\binom{n}{2}}$ simple graphs A with $U^A = \{1, \ldots, n\}$.

We left it as a stimulating recreational activity to calculate the number of 1-regular simple graphs with universe of discourse $\{1, \ldots, n\}$.

12 Lecture 02.27

12.1 Functional relations

You may already have encountered functions, such as the mapping f that sends a real number x to its square x^2. If so, you probably saw this function represented in cartesian coordinates via a graph, that is, the set of all ordered pairs of real numbers $\langle x, x^2\rangle$ for $x \in \mathbb{R}$. For our purposes, we consider this as a structure, in particular, a directed graph A with $U^A = \mathbb{R}$ and $L^A = \{\langle x, x^2\rangle \mid x \in \mathbb{R}\}$. This structure satisfies the following schemata.

- Tot: $(\forall x)(\exists y)Lxy$
- SV: $(\forall x)(\forall y)(\forall z)((Lxy \wedge Lxz) \supset y = z)$

The first of these says that the L is *total*, that is, everything is related (here think "mapped to") at least one thing, and the second says that L is *single-valued*, that is, everything is mapped to at most one thing. Their conjunction, which we abbreviate to Fun, says that L is a total function, that is, if $A \models \mathsf{Fun}$, then L^A is the graph of a total function with domain U^A and range (contained in) U^A. We went on to consider some special types of function, namely injections, surjections, and bijections. An *injection* is a 1-1 function; you may be familiar with the idea in terms of the "horizontal line rule"; we applied this rule to verify that the squaring function mentioned above is not an injection. We schematize the property that "L" is the graph of an injection as follows.

- Inj: $(\forall x)(\forall y)(\forall z)((Lxz \wedge Lyz) \supset x = y)$

A *surjection* is an onto function, that is, every member of the universe is the image of some input to the function, schematically:

- Sur: $(\forall x)(\exists y)Lyx$

We noted that the squaring function is not a surjection on $\mathbb{R}$: no negative number is the square of a real number. We observed that the function which maps a real number to its cube is both an injection and a surjection on $\mathbb{R}$; we call such functions *bijections* and we introduced Bij to abbreviate the conjunction of Inj and Sur.

Since the only examples of functions we considered so far were either bijections (the cubing function) or neither injections nor surjections (the squaring function) we sought for examples of functions which are one but not the other. This led us to Dedekind's definition of "infinite", via the following route. We first observed that for any structure A with a finite universe of discourse, $A \models \mathsf{Fun} \wedge \mathsf{Inj}$ if and only if $A \models \mathsf{Fun} \wedge \mathsf{Sur}$ (and hence, if and only if $A \models \mathsf{Fun} \wedge \mathsf{Bij}$). We then noted that there are functions which are injections but not surjections. For example, consider the structure B where $U^B = \mathbb{N}$ and $L^B = \{\langle n, n+1\rangle \mid n \in \mathbb{N}\}$ and observe that $B \models \mathsf{Fun} \wedge \mathsf{Inj} \wedge \neg\mathsf{Sur}$. It is similarly easy to construct functions which are surjections but not injections, for example, the function on $\mathbb{N}$ that maps a number n to $\lceil n/2\rceil$. A set X is said to be *Dedekind infinite* if and only if there is a function with domain X and range contained in X which is injective but not surjective.

Next, we touched briefly on the topic of multivariate functions; we restricted our attention to binary functions whose graphs we represent as the interpretation of a triadic predicate symbol R. The following schema Bfun expresses both totality and single-valuedness, that is, a structure A satisfies Bfun if and only if R^A is the graph of a total binary function on U^A..

- Bfun: $(\forall x)(\forall y)(\exists z)(\forall w)(Rxyw \equiv w = z)$

The next schema Binj schematizes the notion of injection for binary functions, that is, a structure A satisfies the conjunction of Bfun and Binj if and only if R^A is the graph of an injective binary function.

- Binj: $(\forall v)(\forall w)(\forall x)(\forall y)(\forall z)((Rvwz \wedge Rxyz) \supset (v = x \wedge w = y))$

We observed that if A is a finite structure and $A \models \mathsf{Bfun} \wedge \mathsf{Binj}$, then $|U^A| = 1$. On the other hand, we noted that the binary function which maps a pair of positive integers m and n to $2^m \cdot 3^n$ is an injection. This shows that there are at least as many positive integers as there are positive rational numbers, since every positive rational number can be represented by a pair of integers. This may seem odd, since, in their usual order, between any two positive integers there are infinitely many rational numbers.

12.2 Tournaments and orderings

We went on to consider (all-play-all, no-ties) tournaments. We say a directed graph is a *tournament* if and only if it satisfies the conjunction of the following two conditions, called asymmetry and comparability.

- Asy: $(\forall x)(\forall y)(Lxy \supset \neg Lyx)$
- Comp: $(\forall x)(\forall y)((x \neq y \supset (Lxy \vee Lyx)$

We abbreviate the conjunction of Asy and Comp to Tour. Finally, we picked out a particularly important class of tournaments, those without cycles. We characterized these as the *transitive* tournaments, that is, those satisfying the following schema.

- Trans: $(\forall x)(\forall y)(\forall z)(Lxy \supset (Lyz \supset Lxz))$

Transitive tournaments are called *strict linear orders*; we abbreviate the conjunction of Tour and Trans to SLO.

12.3 Counting functions and tournaments

We counted the number of finite structures with universe of discourse $\{1, \ldots, n\}$ that satisfy various conditions. We'd already noted that there are 2^{n^2} graphs and $2^{\binom{n}{2}}$ simple graphs with universe of discourse $\{1, \ldots, n\}$. We began by showing that

- $|\mathsf{mod}(\mathsf{Fun}, n)| = n^n$;
- $|\mathsf{mod}((\mathsf{Fun} \wedge \mathsf{Inj}), n)| = n!$;
- $|\mathsf{mod}(\mathsf{Asy}, n)| = 3^{\binom{n}{2}}$;
- $|\mathsf{mod}(\mathsf{Tour}, n)| = 2^{\binom{n}{2}}$;
- $|\mathsf{mod}(\mathsf{SLO}, n)| = n!$;
- $|\mathsf{mod}(\mathsf{Bfun}, n)| = n^{(n^2)}$.

In each case, clear thinking and the product rule sufficed for the calculation.

13 Lecture 03.01

We counted the number of labelled (colored) simple graphs that satisfy various conditions that can be expressed by quantificational schemata.

Recall that a simple graph is 2-regualr if and only if it satisfies the schema:

- $\mathsf{2reg}$: $(\forall x)(\exists^{=2} y)Lxy$, which is equivalent to
- $(\forall x)(\exists y)(\exists z)(y \neq z \wedge (\forall w)(Lxw \equiv (w = y \vee w = z)))$.

Let S be the conjunction of $\mathsf{2reg}$ and SG. We calculated $|\mathsf{mod}(S, 6)|$. We began by reminding ourselves that if A is finite and $A \models S$ then A is a disjoint union of cycles. This led immediately to the observation that if $A \in \mathsf{mod}(S, 6)$ then A must consist of two disjoint triangles, or a single hexagon. So in order to complete our calculation, we just need to determine how many distinct ways we can label a structure of one or the other of these shapes. Suppose the unlabeled structure $\mathbb{T}$ consists of two triangles, call them the top triangle and the bottom triangle. We can label the top triangle with any set $X \subseteq [6]$ of size three, leaving $[6] - X$ to label the bottom triangle. At first blush, this suggests that there are $\binom{6}{3}$ distinct labelings of $\mathbb{T}$. But notice that we get the same labeled structure, if we use $[6] - X$ to label the top triangle, and X to label the bottom triangle, so there are $\binom{6}{3}/2 = 10$ distinct labelings of $\mathbb{T}$. Next, suppose the unlabeled structure $\mathbb{H}$ consists of a single hexagon. We used our prior calculation that there are 6! strict linear orders of [n] to calculate the number of distinct labelings of $\mathbb{H}$. For each such linear order, we can "wrap it around" the hexagon starting from a fixed position to arrive at a labeling. It is clear that the reverse of any order gives the same labeling as the order itself, as do each of the orders that arise by starting at the i-th position of the given order, for $i > 1$, and continuing on beyond the sixth position with the first $i - 1$ elements of the given order. Thus, the total number of labelings of $\mathbb{H}$ is $6!/(6 \cdot 2) = 60$. It follows that $|\mathsf{mod}(S, 6)| = 10 + 60 = 70$.

Next, we introduced a monadic predicate letter "F" to "color" the nodes of our graphs. We introduced a further condition, *distinguished end*:

- DE : $(\forall x)(\forall y)(Lxy \supset (Fx \oplus Fy))$.

We considered the schema T: the conjunction of SG, $\mathsf{2reg}$, and DE. We noted that the connected graphs that satisfy T are exactly the even length cycles. It follows at once that $|\mathsf{mod}(T, n)| > 0$ if and only if n is an even number greater than 2. We introduced the notion of the *spectrum* of a schema to describe this.

Definition 7 *Let S be a schema.* $\mathsf{Spec}(S) = \{n \mid |\mathsf{mod}(S, n)| > 0\}$.

Thus, $\mathsf{Spec}(T) = \{2i \mid i > 1\}$.

We calculated $|\mathsf{mod}(T, 6)|$. The only shape allowed in this case is the hexagon, and each hexagon admits two possible colorings that satisfy DE. Hence, it follows from our earlier calculation that $|\mathsf{mod}(T, 6)| = 2 \cdot 6!/(6 \cdot 2) = 120$.

14 Lecture 03.13

14.1 The Spectrum of a Schema

We began to discuss another interesting aspect of the expressive power of polyadic quantification theory. We write $\mathbb{Z}^+$ for the set of positive integers $\{1, 2, 3, \ldots\}$. The *spectrum* of a schema S (written $\mathsf{Spec}(S)$) is defined as follows.

$$\mathsf{Spec}(S) = \{n \in \mathbb{Z}^+ \mid \mathsf{mod}(S, n) \neq \emptyset\}.$$

We can restate the definition in slightly different terms. Say that a schema S *admits* a positive integer n if and only if there is a structure A such that $A \models S$ and $|U^A| = n$. Then $\mathsf{Spec}(S)$ is exactly the set of positive integers n such that S admits n.

14.2 Finite Sets and Co-finite Sets are Spectra

Let F be a finite set of positive integers. We asked, "Is there a schema S such that $\mathsf{Spec}(S) = F$?" We began with singletons and showed that for every positive integer n, there is a schema, call it S_n such that $\mathsf{Spec}(\mathsf{S}_n) = \{n\}$. We may take S_n to be the following schema.

$$(\exists x_1) \ldots (\exists x_n) \bigwedge_{1 \leq i < j \leq n} x_i \neq x_j \wedge \neg(\exists x_1) \ldots (\exists x_{n+1}) \bigwedge_{1 \leq i < j \leq n+1} x_i \neq x_j$$

It follows at once that for any finite set of positive integers $F = \{n_1, \ldots, n_k\}$,

$$\mathsf{Spec}(\mathsf{S}_{n_1} \vee \ldots \vee \mathsf{S}_{n_k}) = F.$$

Moreover, we noted that

$$\mathsf{Spec}(\neg(\mathsf{S}_{n_1} \vee \ldots \vee \mathsf{S}_{n_k})) = \mathbb{Z}^+ - F.$$

Thus, every finite set of positive integers and the complement of every finite set of positive integers is a spectrum (the latter sets are called *co-finite*).

14.3 Complementation and the Spectrum Problem

It is actually quite unusual that the spectrum of the negation of a schema S is equal to the complement of the spectrum of S. We considered the following example. Recall the schema $\mathsf{SG} \wedge \mathsf{1reg}$ which defines the collection of 1-regular simple graphs. We reminded ourselves that we'd already noticed that $\mathsf{Spec}(\mathsf{SG} \wedge \mathsf{1reg})$ is the set of even numbers, that is, $\mathsf{Spec}(\mathsf{SG} \wedge \mathsf{1reg}) = \{2i \mid i \in \mathbb{Z}^+\}$. On the other hand, $\mathsf{Spec}(\neg(\mathsf{SG} \wedge \mathsf{1reg})) = \mathbb{Z}^+$. This behavior is actually typical. Later in the course we may be in a position to prove the following important fact: if the spectrum of a schema S is neither finite nor cofinite, then the spectrum of the negation of S is not equal to the complement of the spectrum of S. This led to a brief discussion of the question, "Is there a schema S such that the complement of the spectrum of S is not the spectrum of any schema whatsoever?" Nobody knows the answer to this question. It is known that a set of positive integers is a spectrum if and only if it is in the complexity class NE, the set of problems solvable in non-deterministic (linear) exponential time on a Turing machine. For those of you who might like to learn more about this open problem, I've uploaded a paper "Fifty Years of the Spectrum Problem" to the course Canvas site.

14.4 Further Examples of Infinite, Co-infinite Spectra

We went on to modify the schema $\mathsf{SG} \wedge \mathsf{1reg}$ to give an example of a schema whose spectrum is the set of odd numbers. The modified schema states the condition that there is an isolated node w, and every node other than w has degree one, in addition to ensuring that any satisfying structure is a simple graph.

We presented a more substantial example, a schema S with $\mathsf{Spec}(S) = \{k^2 \mid k \in \mathbb{Z}^\}$. The schema involved a triadic predicate letter H and a monadic predicate F. S is the conjunction of the following schemata.

- $(\forall x)(\forall y)((Fx \wedge Fy) \supset (\exists z)(\forall w)(Hxyw \equiv w = z))$
- $(\forall x)(\forall y)(\forall z)(Hxyz \supset (Fx \wedge Fy))$
- $(\forall x)(\exists y)(\exists z)Hyzx$
- $(\forall x)(\forall y)(\forall z)(\forall w)(\forall v)((Hxyv \wedge Hzwv) \supset (x = z \wedge y = w))$

Suppose $A \models S$. The conjunction of the first two schemata guarantee that H^A is the graph of a binary function mapping $F^A \times F^A$ to U^A. Further conjoining the third and fourth schemata guarantee that this function is a bijection, thereby insuring that $|U^A|$ is a perfect square.

15 Lecture 03.15

Today, we looked at another important class of graphs, namely, equivalence relations, and saw how they can be put to use in generating schemata with a wide range of spectra. A graph A is an *equivalence relation* if and only if L^A is reflexive, symmetric, and transitive, that is, if and only if $A \models \mathsf{Eq}$, where Eq is the conjunction of the following schemata.

- Refl: $(\forall x)Lxx$
- Sym: $(\forall x)(\forall y)(Lxy \supset Lyx)$
- Trans: $(\forall x)(\forall y)(\forall z)(Lxy \supset (Lyz \supset Lxz))$

Now suppose we'd like to construct a schema S such that

- S implies Eq, and
- $\mathsf{Spec}(S) = \{3i + 2 \mid i \in \mathbb{Z}^+ \cup \{0\}\}$.

The easiest way to meet the first condition is to formulate S as a conjunction, one conjunct of which is Eq itself. But what more should we say? Well, the universe U^A of an equivalence relation A is partitioned into mutually disjoint *equivalence classes* by the relation L^A; for each $a \in U^A$, the equivalence class $\hat{a}$ of a, is $\{b \in U^A \mid \langle a, b\rangle \in L^A\}$. Now if we can construct a schema T that says every equivalence class but one is of size three, and that the exceptional equivalence class is of size two, then we may take S to be the conjunction of Eq and T. The following schema T does the job.

$$
\begin{aligned}
&(\exists x_1)(\exists x_2)(x_1 \neq x_2 \wedge (\forall w)(Lwx_1 \equiv (w = x_1 \vee w = x_2))\wedge \\
&\quad (\forall y_1)((y_1 \neq x_1 \wedge y_1 \neq x_2) \supset (\exists y_2)(\exists y_3)(y_1 \neq y_2 \wedge y_1 \neq y_3 \wedge y_2 \neq y_3 \wedge \\
&\quad (\forall v)(Lvy_1 \equiv (v = y_1 \vee v = y_2 \vee v = y_3)))))
\end{aligned}
$$

We generalized this to show that for every j and $0 \leq k < j$, there is a schema S such that S implies Eq, and $\mathsf{Spec}(S) = \{nj + k \mid n \in \mathbb{Z}^+ \cup \{0\}\}$.

16 Lecture 03.20

16.1 Isomorphisms and automorphisms

We began with the following example. Consider the structures

- A: $U^A = [3]$, $L^A = \{\langle 1, 2\rangle, \langle 1, 3\rangle\}$, and
- B: $U^B = [3]$, $L^B = \{\langle 2, 1\rangle, \langle 2, 3\rangle\}$.

A and B look very similar. We can bring out their similarity by considering the the function $f : [3] \mapsto [3]$ with $f(1) = 2$, $f(2) = 1$, and $f(3) = 3$. The function f is a bijection and is edge-preserving, that is, for every $i, j \in [3]$, $\langle i, j\rangle \in L^A$ if and only if $\langle f(i), f(j)\rangle \in L^B$. We say f is an *isomorphism* of A onto B, and that A and B are *isomorphic* (written $A \cong B$). These notions are so important that we pause to enshrine them in a definition.

Definition 8 *A function h is an* isomorphism *from A onto B if and only if h is a bijection from U^A onto U^B such that for all $a, b \in U^A$, $\langle a, b\rangle \in L^A$ if and only if $\langle h(a), h(b)\rangle \in L^B$.*

A is isomorphic to B *($A \cong B$) if and only if there is an isomorphism h from A onto B.*

Consider again the structure A described above, but now consider the function g with $g(1) = 1$, $g(2) = 3$, and $g(3) = 2$. The function g is an *automorphism* of A, that is, an isomorphism of A onto itself. Again, a definition is in order.

Definition 9 *A function h is an* automorphism *of A if and only if h is an isomorphism of A onto A. $\mathsf{Aut}(A) = \{h \mid h \text{ is an automorphism of } A\}$.*

Note that if $A \cong B$, then for every schema S, $A \models S$ if and only if $B \models S$.

16.2 The image of a structure

We continued to consider the structure A. We listed all the bijections of [3] onto [3].

	1	2	3
f_1	1	2	3
f_2	2	1	3
f_3	3	2	1
f_4	1	3	2
f_5	2	3	1
f_6	3	1	2

We called this set of bijections $\mathbb{S}_3$ (more on this notation below) and introduced the notion of the *image of* a structure C with $U^C = [3]$ ($h[C]$) under $h \in \mathbb{S}_3$: $U^{h[C]} = U^C$ and $L^{h[C]} = \{\langle h(i), h(j)\rangle \mid \langle i, j\rangle \in L^C\}$. It follows that for every such C and h, h is an isomorphism of C onto $h[C]$. We next observed, with respect to the examples A and B above, that $B = f_2[A]$ and that $\mathsf{Aut}(A) =$

$\{f_1, f_4\}$. We also noted that $f_5[A] = B$ and that $f_3[A] = f_6[A]$ is a third isomorphic copy of A distinct from both A and B. That is, there are three labeled structures with universe [3] that are isomorphic to A. We marveled at the identity

$$|\mathbb{S}_3| = |\mathsf{Aut}(A)| \cdot (\text{the number of labeled copies of } A). \tag{2}$$

The next section contains a more substantial explanation of this identity than we had time for in class.

16.3 A new way of counting labeled structures: the Orbit-Stabilizer Theorem

For every positive integer k we write $[k]$ for $\{1, \ldots, k\}$ and $\mathbb{S}_k$ for the set of bijections from $[k]$ onto $[k]$ (also called the *permutation group on* or the *symmetric group on* $[k]$). These latter terms emphasize the following algebraic aspect: we may think of $\mathbb{S}_k$ as an algebra with a binary operation $\circ$, a unary operation $^{-1}$, and a distinguished element e, where, for permutations $f, g \in \mathbb{S}_k$, $f \circ g$ is the permutation resulting from the composition of f and g, that is, $f \circ g = h$ if and only if for every $i \in [k]$, $h(i) = f(g(i))$; f^{-1} is the permutation which is the inverse of f; and e stands for the identity function on $[k]$. With these understandings, you can verify that $\mathbb{S}_k$ is a group:

- $\circ$ is an associative operation, that is, $(f \circ g) \circ h = f \circ (g \circ h)$, for all $f, g \in \mathbb{S}_k$;
- e is an identity with respect to $\circ$, that is, $e \circ f = f \circ e = f$, for all $f \in \mathbb{S}_k$; and
- $f \circ f^{-1} = f^{-1} \circ f = e$, for all $f \in \mathbb{S}_k$.

We write $\mathbb{G}_k$ for the set of simple graphs A with $U^A = [k]$. For each $f \in \mathbb{S}_k$ and $A \in \mathbb{G}_k$, we define $f[A]$ to be the graph with universe $[k]$ such that for all $i, j \in [n]$, $\langle f(i), f(j) \rangle \in L^{f([A])}$ if and only if $\langle i, j \rangle \in L^A$. Note that f is an isomorphism of A onto $f[A]$. This is an example of a *group action* – the group $\mathbb{S}_k$ *acts on* the set $\mathbb{G}_k$ via the assignment of $f[A]$ to A. Verify that for all $A \in \mathbb{G}_k$ and $f, g \in \mathbb{S}_k$,

- $(f \circ g)[A] = f[g[A]]$, and
- $e[A] = A$.

Recall that $\mathsf{Aut}(A)$ is the set of automorphisms of A. In the current context, for $A \in \mathbb{G}_k$, $\mathsf{Aut}(A)$ is often called the *stabilizer* of A, since $f \in \mathsf{Aut}(A)$ if and only if $f[A] = A$. The *orbit of* A under the action of $\mathbb{S}_k$ (written $\mathsf{orb}(A, \mathbb{S}_k)$) is $\{h[A] \mid h \in \mathbb{S}_k\}$. The following result is a special case of the *Orbit-Stabilizer Theorem.*

Theorem 3 *For all $A \in \mathbb{G}_n$,*

$$|\mathbb{S}_n| = |\mathsf{orb}(A, \mathbb{S}_n)| \cdot |\mathsf{Aut}(A)|.$$

Proof: Let $A \in \mathbb{G}_k$. We define an equivalence relation $\sim$ on $\mathbb{S}_k$: for all $f, g \in \mathbb{S}_k$, $f \sim g$ if and only if $(f^{-1} \circ g) \in \mathsf{Aut}(A)$. (You should verify that $\sim$ is an equivalence relation, for example, it is reflexive, that is, $f \sim f$, because $f^{-1} \circ f = e$ and $e \in \mathsf{Aut}(A)$; continue and show $\sim$ is symmetric and transitive.) We establish the following two claims about $\sim$ from which the Theorem follows immediately.

1. each equivalence class of $\sim$ has size $|\mathsf{Aut}(A)|$, and
2. the number of equivalence classes of $\sim$ is $|\mathsf{orb}(A, \mathbb{S}_k)|$.

Ad claim 1: Fix $f \in \mathbb{S}_k$. For each $h \in \mathsf{Aut}(A)$ there is a unique $g \in \mathbb{S}_k$ such that $f^{-1} \circ g = h$. (Verify!) It follows at once that there is a bijection between $\{g \mid f \sim g\}$ and $\mathsf{Aut}(A)$.
Ad claim 2: We show that for every $f, g \in \mathbb{S}_k$ $f[A] = g[A]$ if and only if $f \sim g$. We prove each direction of the bi-conditional. So suppose $f \sim g$. Then $f^{-1} \circ g \in \mathsf{Aut}(A)$. Hence, $(f^{-1} \circ g)[A] = A$. Hence, $f[(f^{-1} \circ g)[A]] = f[A]$. Hence, $(f \circ (f^{-1} \circ g))[A] = f[A]$. Hence, $((f \circ f^{-1}) \circ g)[A] = f[A]$. Hence, $(e \circ g)[A] = f[A]$. Hence, $g[A] = f[A]$. In the other direction, suppose $f[A] = g[A]$. Then, $f^{-1}[f[A]] = f^{-1}[g[A]]$. Hence, $(f^{-1} \circ f)[A] = (f^{-1} \circ g)[A]$. Hence, $(f^{-1} \circ g)[A] = e[A] = A$. Hence, $f^{-1} \circ g \in \mathsf{Aut}(A)$, that is, $f \sim g$. Thus, there is a bijection between the equivalence classes of $\sim$ and $\mathsf{orb}(A, \mathbb{S}_k)$. ■

We now have the explanation of identity (2), since

$$|\mathsf{orb}(A, \mathbb{S}_k)| = \text{the number of labeled copies of } A.$$

17 Lecture 03.22

17.1 Counting labeled 1-regular graphs

Let S be the conjunction of SG and $\mathsf{1reg}$, that is, a graph A satisfies S if and only if A is a 1-regular, simple graph. As we discussed earlier, every such finite graph A has an even number, say $2n$, of nodes; moreover, if $A, B \models S$ and $|U^A| = |U^B|$, then A is isomorphic to B. We calculate the value of $\mathsf{mod}(S, 2n)$ in two ways, both for the intrinsic interest of each, and for the opportunity to "check our work."

17.1.1 Via the Orbit-Stabilizer Theorem

Let $A \in \mathsf{mod}(S, 2n)$. As we've just noted above, if $B \in \mathsf{mod}(S, 2n)$, then $A \cong B$. It follows at once that

$$\mathsf{mod}(S, 2n) = \mathsf{orb}(A, \mathbb{S}_{2n}). \tag{3}$$

Let's calculate $|\mathsf{Aut}(A)|$, since Theorem 3 will then allow us to calculate $|\mathsf{mod}(S, 2n)|$. Observe that A consists of n independent edges. Imagine them standing upright and lined up horizontally in some order. Now any permutation of the edges generates an automorphism of A. Moreover, in the process of permuting the edges, we may choose to "flip" any subset of them having those land on the edge to which they are permuted "head to foot" and "foot to head". Since there are $n!$ permutations of the n edges, and 2^n choices of which set of edges to flip, there are a total of $n! \cdot 2^n$ automorphisms of A. Hence, by Theorem 3 and equation (3),

$$|\mathsf{mod}(S, 2n)| = (2n)!/n! \cdot 2^n.$$

17.1.2 Directly

Here is a second direct method of calculating $|\mathsf{mod}(S, 2n)|$ which, thankfully, yields the same result. We construct a member A of $\mathsf{mod}(S, 2n)$ as follows. We successively choose the n independent edges that constitute A. So for the first edge, we have $\binom{2n}{2}$ choices of a pair of nodes between which to place an edge, and for the second edge, we have $\binom{2n-2}{2}$ choices, ... So the number of ways we can choose a sequence of n independent edges is

$$\binom{2n}{2} \cdot \binom{2n-2}{2} \cdots \binom{4}{2} \cdot \binom{2}{2} = \frac{(2n)!}{2^n}.$$

Now any *set* of n edges chosen via this process will appear as the result of $n!$ such sequences of choices; thus, the total number of members of $\mathsf{mod}(S, 2n)$ we can construct is

$$\frac{(2n)!}{n! \cdot 2^n}.$$

17.2 Definability

Up to this point we have neglected schemata containing free variables. Today we will correct this oversight. Recall the structure A we discussed last time: A: $U^A = [3]$, $L^A = \{\langle 1,2\rangle, \langle 1,3\rangle\}$. We considered the schema

$$S(x): \quad \neg(\exists y)Lyx.$$

We observed that $S(x)$ picks out 1 uniquely from the structure A. That is

$$\{a \in U^A \mid A \models S[x|a]\} = \{1\}.$$

$S(x)$ expresses the property of having in-degree zero. Since we only consider properties extensionally, we can also say that, in a given structure, $S(x)$ defines the set of nodes of in-degree zero. The concept of definability is central in logic (and many other disciplines). We enshrine it in a definition.

Definition 10 *Let $S(x)$ be a schema with one free variable x and let A be a structure. We define $S[A] = \{a \in U^A \mid A \models S[x|a]\}$, that is, $S[A]$ is the set of nodes of A that satisfy the schema $S(x)$ in A when assigned to the variable x. We call $S[A]$ the* set defined by *$S(x)$ in A. We say a set $V \subseteq U^A$ is a* definable subset of *A if and only if there is a schema $S(x)$ such that $S[A] = V$.*

We pursued the example of the particular structure A described above a bit further and noted that the set $\{2,3\}$ is defined by the schema

$$S'(x): \quad \neg(\exists y)Lxy.$$

We asked whether either of the sets $\{2\}$ or $\{3\}$ is a definable subset of A. We despaired of finding a schema which defined either of these sets. We noticed that the nodes labelled 2 and 3 appear to be "indistinguishable from a structural point of view" which is borne out by the fact that the function h mapping 1 to 1, 2 to 3, and 3 to 2, is an automorphism of A. The relevance of this to the question of definability is the content of the following fundamental theorem.

17.2.1 The Automorphism Theorem, Orbits, and Definability over finite structures

Theorem 4 *Let A be a graph and $h \in \mathsf{Aut}(A)$. For every $a \in U^A$ and every schema $S(x)$,*

$$A \models S[x|a] \textit{ if and only if } A \models S[x|h(a)].$$

Theorem 4 enables us to give a characterization of the definable subsets of finite structures. If f is a function with domain U and $V \subseteq U$, we define $f[V] = \{f(a) \mid a \in V\}$ (the f *image* of V). With this notation in hand, we can now state a corollary to Theorem 4 which bears on definability.

Corollary 4 *Let A be a graph and $h \in \mathsf{Aut}(A)$. If V is a definable subset of A, then $h[V] = V$.*

Thus, in order to show that V is *not* a definable subset of A it suffices to exhibit an $h \in \mathsf{Aut}(A)$ and $a \in V$ such that $h(a) \notin V$. Moreover, in the case of finite structures, the converse of Corollary 4 is true.

Theorem 5 *Let A be a finite graph and $V \subseteq U^A$. V is a definable subset of A, if for every $h \in \mathsf{Aut}(A)$, $h[V] = V$.*

18 Lecture 03.29

18.1 Orbits and Definability over finite structures

In order to prove Theorem 5, and to apply it to questions of counting definable sets, it will be useful to introduce the notion of the *orbit of a node* $a \in U^A$ *under the action of* $\mathsf{Aut}(A)$:

$$\mathsf{orb}(a, \mathsf{Aut}(A)) = \{h(a) \mid h \in \mathsf{Aut}(A)\}.$$

We define $\mathsf{Orbs}(A, \mathsf{Aut}(A)) = \{\mathsf{orb}(a, \mathsf{Aut}(A)) \mid a \in U^A\}$. As a corollary to Corollary 4 and Theorem 5 we have:

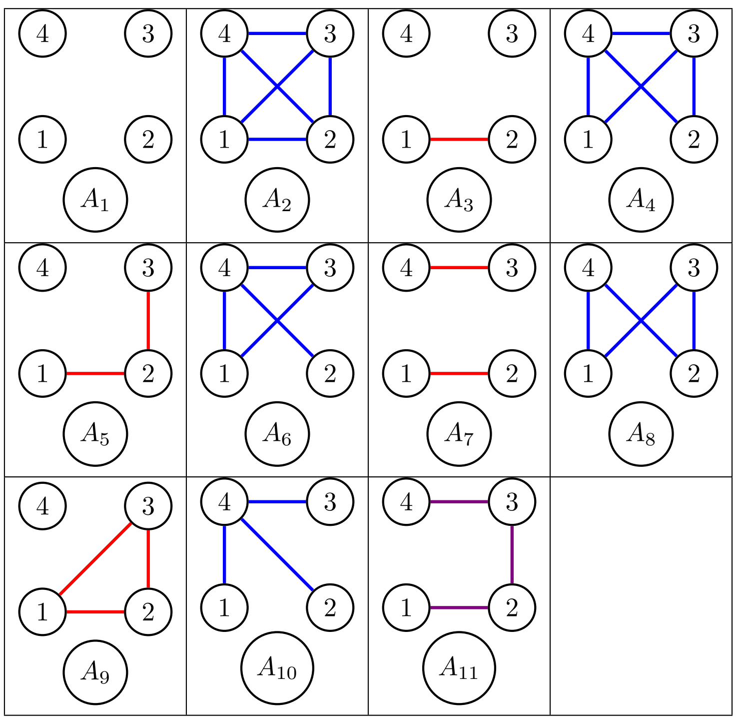
Corollary 5 *Let A be a finite graph and $V \subseteq U^A$. V is a definable subset of A if and only if either $V = \emptyset$ or there is a sequence of sets $O_1, \ldots, O_k$, where each $O_i \in \mathsf{Orbs}(A)$, and $V = O_1 \cup \ldots \cup O_k$.*

It follows at once from Corollary 5, that if A is a finite graph, then the number of definable subsets of A is $2^{|\mathsf{Orbs}(A, \mathsf{Aut}(A))|}$.

19 Lecture 04.03

19.1 An example: definable subsets of simple graphs with four nodes

We proceeded to give a complete analysis of the definable subsets of simple graphs with four nodes. First, we classified all the members of $\mathsf{mod}(\mathsf{SG}, 4)$ up to isomorphism. We discovered that any maximal collection of pairwise non-isomorphic graphs in $\mathsf{mod}(\mathsf{SG}, 4)$ has exactly 11 members. We listed such a collection $A_1, \ldots, A_{11}$ and calculated $|\mathsf{orb}(A_i, \mathbb{S}_4)|$ and $|\mathsf{Aut}(A_i)|$ for each $1 \leq i \leq 11$. See the tables below. The *complement* A^c of a simple graph A is defined as follows: $U^{A^c} = U^A$; for $a \neq b$, $\langle a, b\rangle \in L^{A^c}$ if and only if $\langle a, b\rangle \notin L^A$. In the table of graphs below, each A_i with i odd, is drawn in red, and $A_{i+1} = A_i^c$ is drawn in blue. The exceptional graph A_{11} is drawn in purple since it is isomorphic to its own complement.



Note that $\mathsf{Aut}(A) = \mathsf{Aut}(A^c)$, for every simple graph A. This made it quick work to complete the following table.

A_i	$\lvert\mathsf{orb}(A_i, \mathbb{S}_4)\rvert$	$\lvert\mathsf{Aut}(A_i)\rvert$
A_1	1	24
A_2	1	24
A_3	6	4
A_4	6	4
A_5	12	2
A_6	12	2
A_7	3	8
A_8	3	8
A_9	4	6
A_{10}	4	6
A_{11}	12	2

Note the "verification" of the result predicted by the Orbit-Stabilizer Theorem: $|\mathsf{orb}(A_i, \mathbb{S}_4)| \cdot |\mathsf{Aut}(A_i)| = |\mathbb{S}_4| (= 24)$.

We gave a systematic account of which sets are definable in the structures $A_1, \ldots, A_{11}$. The following table, together with Corollary 5, suffices. We write $\mathsf{Orbs}(A, \mathsf{Aut}(A))$ to denote the collection of orbits of $\mathsf{Aut}(A)$ acting on U^A. We list only the odd numbered structures, since, as already observed, $\mathsf{Aut}(A) = \mathsf{Aut}(A^c)$.

A_i	$\mathsf{Orbs}(A_i, \mathsf{Aut}(A_i))$
A_1	$\{[4]\}$
A_3	$\{\{1,2\},\{3,4\}\}$
A_5	$\{\{2\},\{4\},\{1,3\}\}$
A_7	$\{[4]\}$
A_9	$\{\{1,2,3\},\{4\}\}$
A_{11}	$\{\{1,4\},\{2,3\}\}$

19.2 Additional topics not covered fully in class

19.2.1 Automorphisms preserve degree

Let A be a graph and $a \in U^A$. the *neighborhood of* a in A (written $\mathsf{nbh}(a, A)$) is $\{b \in U^A \mid \langle a, b\rangle \in L^A\}$. The *degree of* a in A (written $\mathsf{deg}(a, A)$) is $|\{b \in U^A \mid \langle a, b\rangle \in L^A\}|$. The next proposition follows directly from the definition of an automorphism.

Proposition 2 *For every graph A, $a \in U^A$, and $h \in \mathsf{Aut}(A)$,*

$$h[\mathsf{nbh}(a, A)] = \mathsf{nbh}(h(a), A).$$

Hence,

$$\mathsf{deg}(a, A) = \mathsf{deg}(h(a), A).$$

19.2.2 Rigidity

We introduced the notion of rigidity: a graph A is *rigid* if and only if $\mathsf{Aut}(A) = \{e\}$, that is, A has no non-trivial automorphisms. It follows at once from Theorem 5 that if A is a finite rigid structure and $V \subseteq U^A$, then $V \in \mathsf{Def}(A)$. We noted that no member of $\mathsf{mod}(\mathsf{SG}, 4)$, is rigid, and mused about the question: "what is the least n such that $\mathsf{mod}(\mathsf{SG}, n)$ contains a rigid graph?"

19.2.3 Proof Sketch of Theorem 5

We give the argument for graphs; the generalization to structures interpreting multiple polyadic predicates is straightforward.

Suppose A is a finite graph, $a \in U^A$, and $V = \mathsf{orb}(A, \mathsf{Aut}(A))$. We construct a schema $S(x)$ such that $S[A] = V$. We may suppose without loss of generality that $U^A = [k]$ for some $k \in \mathbb{Z}^+$ and that $a = 1$. For each $1 \leq i, j \leq k$, let the schema $S_{i,j}$ be Lx_ix_j if $\langle i, j\rangle \in L^A$, and $\neg Lx_ix_j$ otherwise. Let $S(x)$ be the schema

$$(\exists x_2)\ldots(\exists x_k)(\bigwedge_{1\leq i,j\leq k} S_{i,j} \wedge \bigwedge_{1\leq i<j\leq k} x_i \neq x_j \wedge (\forall y)\bigvee_{1\leq i\leq k} y = x_i).$$

Let $a_1, \ldots, a_k$ be a sequence of nodes from U^A and observe that

$$A \models (\bigwedge_{1\leq i,j\leq k} S_{i,j} \wedge \bigwedge_{1\leq i<j\leq k} x_i \neq x_j \wedge (\forall y)\bigvee_{1\leq i\leq k} y = x_i)[(x_1|a_1), \ldots, (x_k|a_k)]$$

if and only if the function mapping i to a_i is an automorphism of A. ■

19.2.4 Proof Sketch of Theorem 4

Theorem 4 is a corollary of the following more general result concerning isomorphisms of structures.

Theorem 6 *Suppose A and B are structures and f is an isomorphism of A onto B. Then for every schema $S(x_1, \ldots, x_k)$ and sequence of elements $a_1, \ldots, a_k \in U^A$,*

$$A \models S[(x_1|a_1), \ldots, (x_k|a_k)] \textit{ iff } B \models S[(x_1|f(a_1)), \ldots, (x_k|f(a_k))]. \tag{4}$$

Proof sketch of Theorem 6: We give the argument for graphs; the generalization to structures interpreting multiple polyadic predicates is straightforward. The argument proceeds by induction on the syntactic structure of schemata. The base case verifies (4) for atomic schemata, that is, schemata of the form Lx_ix_j or $x_i = x_j$, for some i, j. In this case, the verification follows directly from the hypothesis that f is an isomorphism from A onto B, in particular, that it is edge-preserving and injective.

Suppose S is a truth-functional combination, for example the conjunction, of schemata S' and S'', where, as hypothesis of induction, (4) holds for both S'

and S''. Then,

$$\begin{array}{ll}
A \models S[(x_1|a_1),\ldots,(x_k|a_k)] & \text{iff}\\
A \models S'[(x_1|a_1),\ldots,(x_k|a_k)] \text{ and } A \models S''[(x_1|a_1),\ldots,(x_k|a_k)] & \text{iff}\\
B \models S'[(x_1|f(a_1)),\ldots,(x_k|f(a_k))] \text{ and } B \models S''[(x_1|f(a_1)),\ldots,(x_k|f(a_k))] & \text{iff}\\
B \models S[(x_1|f(a_1)),\ldots,(x_k|f(a_k))]. &
\end{array}$$

The first and third biconditionals follow from the truth-functional semantics of conjunction, while the second follows from the induction hypothesis.

Finally, suppose that S is $(\exists y)S'(x_1,\ldots,x_k,y)$ and (4) holds for S' (the universal quantifier is handled similarly). Then,

$$\begin{array}{ll}
A \models S[(x_1|a_1),\ldots,(x_k|a_k)] & \text{iff}\\
\text{for some } a \in U^A \; A \models S'[(x_1|a_1),\ldots,(x_k|a_k),(y|a)] & \text{iff}\\
\text{for some } a \in U^A \; B \models S'[(x_1|f(a_1)),\ldots,(x_k|f(a_k)),(y|f(a))] & \text{iff}\\
\text{for some } b \in U^B \; B \models S'[(x_1|f(a_1)),\ldots,(x_k|f(a_k)),(y|b)] & \text{iff}\\
B \models S[(x_1|f(a_1)),\ldots,(x_k|f(a_k))]. &
\end{array}$$

The first and fourth biconditionals follow from the semantics for the existential quantifier, the second from the induction hypothesis, and the third from the hypothesis that f is an isomorphism from A onto B, in particular, that it is surjective. ∎

20 Lecture 04.05

20.1 Definability in Infinite Structures: Two Examples

We began to look at definability in infinite structures.

20.1.1 The integers with the absolute value relation - a structure with many automorphisms

We first analyzed definability in the infinite graph A described as follows:

- $U^A = \mathbb{Z}$, the set of all integers, $\{\ldots -2, -1, 0, 1, 2, \ldots\}$;
- $L^A = \{\langle i, j\rangle \mid j \text{ is the absolute value of } i\}$. (Recall that the absolute value of an integer i is i, if $i \geq 0$, and is $-i$, if $i < 0$.)

We observed that every permutation g of $\mathbb{Z}^+$ can be extended to an automorphism h of A by setting $h(i) = g(i)$, for $i \in \mathbb{Z}^+$; $h(0) = 0$; and $h(i) = -g(-i)$, for $i < 0$. Let's write $\mathbb{Z}^-$ for the set of negative integers. Thus, $\mathsf{Orbs}(A, \mathsf{Aut}(A)) = \{\mathbb{Z}^+, \{0\}, \mathbb{Z}^-\}$. Each orbit of $\mathsf{Aut}(A)$ acting on U^A is definable:

- $S_1[A] = \mathbb{Z}^+$, where $S_1(x)$ is $(\exists y)(y \neq x \wedge Lyx)$;
- $S_2[A] = \mathbb{Z}^-$, where $S_2(x)$ is $(\forall y)\neg Lyx$;
- $S_3[A] = \{0\}$, where $S_3(x)$ is $\neg S_1(x) \wedge \neg S_2(x)$.

Hence, there are exactly eight sets definable in A:

1. $\emptyset$,
2. $\{0\}$,
3. $\mathbb{Z}^+$,
4. $\mathbb{Z}^-$,
5. $\mathbb{Z}^+ \cup \mathbb{Z}^-$,
6. $\mathbb{Z}^+ \cup \{0\}$,
7. $\mathbb{Z}^- \cup \{0\}$,
8. $\mathbb{Z}$.

20.1.2 The non-negative integers with the successor relation - a rigid structure

We next looked at another infinite structure B where definability behaves very differently. B is described as follows:

- $U^B = \mathbb{Z}^+ \cup \{0\}$;

- $L^B = \{\langle i, j\rangle \mid j = i + 1\}$.

We first observed that $\mathsf{Aut}(B) = \{e\}$, that is, B is a rigid structure. This can be established by mathematical induction. Suppose h is an automorphism of B. Since 0 is the only node of B with in-degree 0, we must have $h(0) = 0$. Now suppose, as induction hypothesis, that $h(n) = n$. Since $n + 1$ is the only member of U^B to which n is related, it follows from the hypothesis that h is an automorphism that $h(n + 1) = n + 1$. This completes the induction; thus, for all $k \in U^B$, $h(k) = k$. Hence, $\mathsf{Aut}(B) = \{e\}$.

This argument suggests that for every $k \in U^B$, $\{k\}$ is definable over B. Let's show this, again by induction. First, the schema $S^0(x) : (\forall y)\neg Lyx$ defines $\{0\}$ over B. Next, as induction hypothesis, suppose that $S^n(x)$ defines $\{n\}$ over B. Let z be a variable which does not occur anywhere in $S^n(x)$ and let $S^n(z)$ be the result of replacing x with z at all its occurrences in $S^n(x)$. then the schema $(\exists z)(S^n(z) \wedge Lzx)$ defines $\{n + 1\}$ over B. this completes the induction and establishes that for every $k \in U^B$, $\{k\}$ is definable over B. It follows at once that every finite subset of U^B and every co-finite subset of U^B is definable over B.

What other subsets of U^B are definable over B? Note that since B is rigid, there is no possibility of exhibiting an automorphism h of B with $h[X] \neq X$, that is, the "automorphism method" is powerless to establish the undefinability of any subset of U^B in B. Could it be that every subset of U^B is definable over B? We will answer this question next time.

21 Lecture 04.10

21.1 Undefinability in Infinite Structures: Two Techniques

21.1.1 Cantor's Theorem and Cardinality Arguments

We show that for every infinite structure C there is a subset $X \subseteq U^C$ which is *not* definable over C. This result is a corollary to the celebrated Cantor Diagonal Theorem.

Theorem 7 (Cantor) *Let U be an infinite set and let $V_1, V_2, \ldots$ be a sequence of subsets of U. There is subset W of U such that for all $i \geq 1$, $W \neq V_i$.*

Proof: Suppose U is an infinite set. Let $U^* = \{a_1, a_2, \ldots\}$ be a countably infinite subset of U and let $V_1, V_2, \ldots$ be a sequence of subsets of U. Let $W = \{i \mid a_i \notin V_i\}$. Note that for every i $a_i \in W$ if and only if $a_i \notin V_i$. It follows that for all i, $W \neq V_i$. ■

In order to apply Theorem 7 to questions about definable sets we require the following result.

Theorem 8 *For every structure C, there is a sequence $V_1, V_2, \ldots$ of subsets of U^C such that for every set X definable over C, there is an i such that $X = V_i$.*

Proof: Every schema is a finite sequence of symbols drawn from a finite alphabet. Thus, we may arrange all schemata $S(x)$ in a list $S_1(x), S_2(x), \ldots$, first ordered by length, and then within length, alphabetically. We obtain a list $V_1, V_2, \ldots$ of all the sets definable over C by setting $V_i = S_i[C]$ for all i. ■

The following result is an immediate consequence of Theorems 7 and 8.

Corollary 6 *For every infinite structure C there is a subset $X \subseteq U^C$ which is not definable over C.*

21.1.2 The Compactness Theorem and Automorphisms of "Non-standard Models"

Of course, this gives us no idea which particular sets are not definable over a given infinite structure. In the case of the graph B introduced above, we will show that if a set is neither finite nor co-finite, it is *not* definable over B. In order to establish this, we will deploy one of the fundamental properties of polyadic quantification theory: *compactness*. First, some definitions requisite to the Compactness Theorem for Polyadic Quantification Theory.

- A schema S is *satisfiable* if and only if for some structure A, $A \models S$.
- A set of schemata Γ is *satisfiable* if and only if there is structure A such that for every schema $S \in \Gamma$, $A \models S$.
- A set of schemata Γ is *finitely satisfiable* if and only if for every finite set $\Delta \subseteq \Gamma$, Δ is satisfiable.

Theorem 9 (Compactness Theorem) *For every set Γ of schemata of polyadic quantification theory, if Γ is finitely satisfiable, then Γ is satisfiable.*

Though the Compactness Theorem makes no mention of the notion of a derivation, one of its well-known proofs proceeds via the elaboration of a sound and complete formal system for logical deduction. This development will occupy our attention for much of the remainder of the Term. But for the moment, let's see how we can apply the Compactness Theorem to complete the analysis of the definable subsets of the structure B specified above.

Theorem 10 *If $V \subseteq U^B$ is definable over B, then V is finite or V is co-finite.*

Proof: Suppose to the contrary, that there is a set V, definable over B, which is neither finite nor co-finite, and suppose that the schema $S(x)$ defines V over B. We derive a contradiction from this hypothesis. Let $\Lambda = \{S \mid B \models S\}$; Λ is the set of all schemata true in the structure B and is often called the *complete theory* of B. Let y and z be fresh variables which occur nowhere in Λ, $S(x)$, or any of the schemata $S^n(x)$ for $n \geq 0$ defined above. Define the set of schemata Γ as follows.

$$\Gamma = \Lambda \cup \{y \neq z, S(y), \neg S(z)\} \cup \{\neg S^n(y), \neg S^n(z) \mid n \geq 0\}.$$

Let Δ be a finite subset of Γ. It follows from the fact that both $S[B]$ and $\neg S[B]$ are infinite, that Δ is satisfied by B with suitable assignments from U^B to the variables y and z. Hence, by the Compactness Theorem, Γ itself is satisfiable. Of course, if the structure C satisfies Γ, then C is not isomorphic to B since the the elements of U^C assigned to y and z in C (call them a and b respectively) are not reachable in C from the unique element of C with no predecessor. We will show that there is an automorphism h of C with $h(a) = b$. This will yield the desired contradiction, since $C \models S(y|a)$ and $C \models \neg S(z|b)$. Note that B, and hence C, satisfy the following schemata.

- $(\exists x)(\forall y)((\forall z)\neg Lzy \equiv x = y)$
- $(\forall x)(\exists y)(\forall z)(Lxz \equiv z = y)$
- $(\forall x)(\forall y)(\forall z)((Lxz \wedge Lyz) \supset x = y)$
- $(\forall x)\neg Lxx$

  $\vdots$

  $(\forall x)(\forall y_1)\dots(\forall y_n)\neg Lxy_1 \wedge Ly_1y_2 \dots \wedge Ly_nx$

  $\vdots$

The first three schemata guarantee that L^C is an injective functional relation which is "almost" surjective – there is a unique element of U^C which lacks a pre-image under the function whose graph is L^C. Note that this guarantees that U^C is infinite. The final infinite list of schemata guarantee that the the function whose graph is L^C contains no finite cycles. Since C is not isomorphic to B all this implies that C consists of an L^C chain that is isomorphic to B and a non-empty set of L^C chains each of which is isomorphic to $\mathbb{Z}$ (the set of all integers) equipped with its usual successor relation. But, since a and b must lie on one or two of these "$\mathbb{Z}$-chains," there is an automorphism h of C with $h(a) = b$. ∎

22 Lecture 04.12

Up to this point, we have focussed primarily on questions surrounding the expressive power of polyadic quantification: which classes of structures can be characterized by (sets of) schemata of polyadic quantification theory; which sets of numbers are the spectra of schemata; what subsets of the universe of discourse of a structure can be defined by schemata. Today we begin our study of implication in the context of polyadic quantification theory. As emphasized earlier, the mechanical decidability of validity of schemata (over a fixed, effectively presented vocabulary of sentence letters) in the case of truth-functional logic, follows immediately from the definition of validity, since there are only finitely many truth assignments to a finite collection of sentence letters, and since the truth-value of a schema under any such assignment can be mechanically (even efficiently) evaluated. (As we discussed, it remains an open problem whether validity itself can be decided efficiently.) In the case of monadic quantification theory, though there are infinitely many structures interpreting the vocabulary of a fixed schema S, we were able to establish that we could effectively determine from S a finite collection of finite structures such that S is valid if and only if satisfied by every structure in this collection. Again, the satisfaction relation itself is mechanically decidable for finite structures, and thus validity of monadic schemata is mechanically decidable.

When we come to polyadic quantification theory, the situation is dramatically different. We will later see that the set of valid schemata of polyadic quantification theory, even restricted to the language of directed graphs, is *not* decidable (the Church-Turing Theorem), though it is semi-decidable (the Gödel Completeness Theorem). Today, we begin a detailed study of systematic techniques to establish that a schema of polyadic quantification theory is valid. These techniques are embodied in the deductive apparatus for polyadic quantification theory expounded in Warren Golfarb's text *Deductive Logic.* We started with an example of a deduction using the rules described on pages 183 – 185 of the text which shows that if a relation is asymmetric, then it is irreflexive.

$$\{(\forall x)(\forall y)(Lxy \supset \neg Lyx)\} \text{ implies } (\forall x)\neg Lxx.$$

$\{1\}$	(1)	$(\forall x)(\forall y)(Lxy \supset \neg Lyx)$	P
$\{1\}$	(2)	$(\forall y)(Lxy \supset \neg Lyx)$	(1) UI
$\{1\}$	(3)	$Lxx \supset \neg Lxx$	(2) UI
$\{1\}$	(4)	$\neg Lxx$	(3) TF
$\{1\}$	(5)	$(\forall x)\neg Lxx$	(4) UG

23 Lecture 04.17

We continued our study of deduction, and began by showing that if a relation is transitive and irreflexive, then it's asymmetric.

$$\{(\forall x)(\forall y)(\forall z)(Lxy \supset (Lyz \supset Lxz)), (\forall x)\neg Lxx\} \text{ implies } (\forall x)(\forall y)(Lxy \supset \neg Lyx).$$

$\{1\}$	(1) $(\forall x)(\forall y)(\forall z)(Lxy \supset (Lyz \supset Lxz))$	P
$\{1\}$	(2) $(\forall y)(\forall z)(Lxy \supset (Lyz \supset Lxz))$	(1) UI
$\{1\}$	(3) $(\forall z)(Lxy \supset (Lyz \supset Lxz))$	(2) UI
$\{1\}$	(4) $Lxy \supset (Lyx \supset Lxx)$	(3) UI
$\{5\}$	(5) $(\forall x)\neg Lxx$	P
$\{5\}$	(6) $\neg Lxx$	(5) UI
$\{1,5\}$	(7) $(Lxy \supset \neg Lyx)$	$(4,6)$ TF
$\{1,5\}$	(8) $(\forall y)(Lxy \supset \neg Lyx)$	(7) UG
$\{1,5\}$	(9) $(\forall x)(\forall y)(Lxy \supset \neg Lyx)$	(8) UG

We next gave a classic "argument by cases."

$$\{(\forall x)Fx \vee (\forall x)Gx\} \text{ implies } (\forall x)(Fx \vee Gx).$$

$\{1\}$	(1) $(\forall x)Fx \vee (\forall x)Gx$	P
$\{2\}$	(2) $(\forall x)Fx$	P
$\{2\}$	(3) Fx	(2) UI
$\{2\}$	(4) $Fx \vee Gx$	(3) TF
$\{2\}$	(5) $(\forall x)(Fx \vee Gx)$	(4) UG
$\{\}$	(6) $(\forall x)Fx \supset (\forall x)(Fx \vee Gx)$	$\{2\}$(5) D
$\{7\}$	(7) $(\forall x)Gx$	P
$\{7\}$	(8) Gx	(7) UI
$\{7\}$	(9) $Fx \vee Gx$	(8) TF
$\{7\}$	(10) $(\forall x)(Fx \vee Gx)$	(9) UG
$\{\}$	(11) $(\forall x)Gx \supset (\forall x)(Fx \vee Gx)$	$\{7\}$(10) D
$\{1\}$	(12) $(\forall x)(Fx \vee Gx)$	$(1,6,11)$ TF

We followed with an example (to be concluded next time) of argument by *reductio ad absurdum*, that, in addition, illustrated use of the "conversion of quantifiers" rule, also known as "driving a negation across a quantifier."

$$(\exists y)(Py \supset (\forall x)Px) \text{ is valid}$$

$\{1\}$	(1) $\neg(\exists y)(Py \supset (\forall x)Px)$	P
$\{1\}$	(2) $(\forall y)\neg(Py \supset (\forall x)Px)$	(1) CQ
$\{1\}$	(3) $\neg(Py \supset (\forall x)Px)$	(2) UI
$\{1\}$	(4) Py	(3) TF
$\{1\}$	(5) $(\forall x)Px$	(4) UG
$\{1\}$	(6) $\neg(\forall x)Px \wedge (\forall x)Px$	(3)(5) TF
$\{\}$	(7) $\neg(\exists y)(Py \supset (\forall x)Px) \supset (\neg(\forall x)Px \wedge (\forall x)Px)$	$\{1\}$(6) D
$\{\}$	(8) $(\exists y)(Py \supset (\forall x)Px)$	(7) TF

Here are a pair of deductions that legitimate extending the "conversion of quantifiers rule" to allow passing directly from $\neg(\forall x)S$ to $(\exists x)\neg S$ and *vice versa.* They provide further illustration of argument by *reductio ad absurdum.* I include them here, though we will not cover them in class.

$\{\neg(\forall x)Fx\}$ implies $(\exists x)\neg Fx$.

$\{1\}$	(1) $\neg(\forall x)Fx$	P
$\{2\}$	(2) $\neg(\exists x)\neg Fx$	(1) P
$\{2\}$	(3) $(\forall x)\neg\neg Fx$	(2) CQ
$\{2\}$	(4) $\neg\neg Fx$	(3) UI
$\{2\}$	(5) Fx	(4) TF
$\{2\}$	(6) $(\forall x)Fx$	(5) UG
$\{1, 2\}$	(7) $(\forall x)Fx \land \neg(\forall x)Fx$	(6) TF
$\{1\}$	(8) $\neg(\exists x)\neg Fx \supset ((\forall x)Fx \land \neg(\forall x)Fx)$	$\{2\}$(7) D
$\{1\}$	(9) $(\exists x)\neg Fx$	(8) TF

$\{(\exists x)\neg Fx\}$ implies $\neg(\forall x)Fx$.

$\{1\}$	(1) $(\forall x)Fx$	P
$\{2\}$	(2) $(\exists x)\neg Fx$	(1) P
$\{1\}$	(3) Fx	(1) UI
$\{1\}$	(4) $\neg\neg Fx$	(3) TF
$\{1\}$	(5) $(\forall x)\neg\neg Fx$	(4) UG
$\{1\}$	(6) $\neg(\exists x)\neg Fx$	(5) CQ
$\{1, 2\}$	(7) $\neg(\exists x)\neg Fx \land (\exists x)\neg Fx$	(6) TF
$\{2\}$	(8) $(\forall x)Fx \supset (\neg(\exists x)\neg Fx \land (\exists x)\neg Fx)$	$\{1\}$(7) D
$\{1\}$	(9) $\neg(\forall x)Fx$	(8) TF

24 Lecture 04.19

We then extended the rules to include existential generalization and existential instantiation which allow us to mirror common informal forms of argument involving the existential quantifier. The following gives an example of their use.

$\{(\forall x)((\exists y)Lxy \supset (\forall z)Lzx), (\exists x)(\exists y)Lxy\}$ implies $(\forall v)(\forall z)Lzv$.

$\{1\}$	(1) $(\exists x)(\exists y)Lxy$	P
$\{1, 2\}$	(2) $(\exists y)Lwy$	(1)w EII
$\{3\}$	(3) $(\forall x)((\exists y)Lxy \supset (\forall z)Lzx)$	P
$\{3\}$	(4) $(\exists y)Lwy \supset (\forall z)Lzw$	(3) UI
$\{1, 2, 3\}$	(5) $(\forall z)Lzw$	(2)(4) TF
$\{1, 2, 3\}$	(6) Lvw	(5) UI
$\{1, \cancel{2}, 3\}$	(7) $(\exists y)Lvy$	(5) EG; $\{2\}$ EIE
$\{3\}$	(8) $(\exists y)Lvy \supset (\forall z)Lzv$	(3) UI
$\{1, 3\}$	(9) $(\forall z)Lzv$	(7)(8) TF
$\{1, 3\}$	(10) $(\forall v)(\forall z)Lzv$	(9) UG

25 Lecture 04.24

Finally, we added rules for deriving schemata involving the identity predicate and illustrated their use with the following deduction.

$$\{(\forall x)Rxx, \neg(\forall x)(\forall y)Rxy\} \text{ implies } \neg(\exists x)(\forall y)x = y.$$

$\{1\}$	(1) $(\forall x)Rxx$	P
$\{2\}$	(2) $\neg(\forall x)(\forall y)Rxy$	P
$\{3\}$	(3) $(\exists x)(\forall y)x = y$	P
$\{3,4\}$	(4) $(\forall y)u = y$	(3)u EII
$\{1\}$	(5) Ruu	(1) UI
$\{3,4\}$	(6) $u = y$	(4) UI
$\{\}$	(7) $u = y \supset (Ruu \equiv Ruy)$	III
$\{3,4\}$	(8) $u = x$	(4) UI
$\{\}$	(9) $u = x \supset (Ruy \equiv Rxy)$	III
$\{1,3,$ $\cancel{4}\}$	(10) Rxy	(5)(6) TF; (7)(8) $\{4\}$ EIE (9)
$\{1,3\}$	(11) $(\forall y)Rxy$	(10) UG
$\{1,3\}$	(12) $(\forall x)(\forall y)Rxy$	(11) UG
$\{1,2,3\}$	(13) $p \wedge \neg p$	(2)(12) TF
$\{1,2\}$	(14) $(\exists x)(\forall y)x = y \supset$ $(p \wedge \neg p)$	$\{3\}$(13) D
$\{1,2\}$	(15) $\neg(\exists x)(\forall y)x = y$	(14) TF

26 Lecture 04.26

We considered the problem of establishing that a schema S is not implied by a set of schemata X, or equivalently, that the set of schemata $X \cup \{\neg S\}$ is not satisfiable. As we noted last time, there is no uniform approach to this problem, that is, the collection of satisfiable schemata is *not* semi-decidable.

Let X be the conjunction of the following schemata.

- $(\forall x)(\forall y)(\forall z)((Lxy \wedge Lyz) \supset Lxz)$
- $(\forall x)(\forall y)(x \neq y \supset (Lxy \vee Lyx))$
- $(\forall x)\neg Lxx$
- $(\forall x)(\exists y)(Lxy \wedge (\forall z)\neg(Lxz \wedge Lzy))$
- $(\forall x)(\exists y)(Lyx \wedge (\forall z)\neg(Lyz \wedge Lzx))$
- $(\forall x)(\exists y)(Lyx \wedge Fy)$
- $(\forall x)(\exists y)(Lxy \wedge Fy)$
- $(\forall x)(\forall y)((Fx \wedge Fy \wedge Lxy) \supset (\exists z)(Fz \wedge Lxz \wedge Lzy))$

We showed that $X \not\models (\forall x)Lxx$, that is, we showed X is satisfiable by constructing a structure A with $A \models X$. The structure A is defined as follows. Recall that $\mathbb{Z}$ is the set of integers and $\mathbb{Q}^+$ is the set of positive rational numbers.

- $U^A = \mathbb{Q}^+ \times \mathbb{Z} = \{\langle r, i\rangle \mid r \in \mathbb{Q}^+ \text{ and } i \in \mathbb{Z}\}$ (the cartesian product of $\mathbb{Q}^+$ and $\mathbb{Z}$).
- $L^A = \{\langle\langle r, i\rangle, \langle s, j\rangle\rangle \mid r < s\} \cup \{\langle\langle r, i\rangle, \langle s, j\rangle\rangle \mid r = s \text{ and } i < j\}$.

We gave another example of demonstrating satisfiability, this time for an infinite collection of schemata. Let S be the conjunction of the following schemata.

- $(\forall x)(\forall y)(\forall z)((Lxy \wedge Lyz) \supset Lxz)$
- $(\forall x)(\forall y)(x \neq y \supset (Lxy \vee Lyx))$
- $(\forall x)\neg Lxx$
- $(\forall x)((\exists y)Lxy \supset (\exists y)(Lxy \wedge (\forall z)\neg(Lxz \wedge Lzy)))$
- $(\forall x)((\exists y)Lyx \supset (\exists y)(Lyx \wedge (\forall z)\neg(Lyz \wedge Lzx)))$
- $\neg(\forall x)(\exists y)Lyx$
- $\neg(\forall x)(\exists y)Lxy$

For each $n \geq 2$, let R^n be the schema,

$$(\exists x_1)\ldots(\exists x_n) \bigwedge_{1 \leq i < j \leq n} Lx_ix_j.$$

Finally, let $X = \{S\} \cup \{R^n \mid n \geq 2\}$. We gave two proofs that X is satisfiable. The first appealed to the

Theorem 11 (Compactness Theorem) *Let Σ be a set of schemata of polyadic quantification theory. If every finite $\Delta \subseteq \Sigma$ is satisfiable, then Σ is satisfiable.*

First Proof: Observe that for every $n \geq 2$, $\{S\} \cup \{R^m \mid m \leq n\}$ is satisfied by a linear order of length n. Hence, by the Compactness Theorem, X is satisfiable. ■

Second Proof: Define the structure B as follows.

- $U^B = \mathbb{Z}$.
- $L^B = \{\langle i, j\rangle \mid (0 \leq i \text{ and } j < 0) \text{ or } (i < j \text{ and } (0 \leq i, j \text{ or } i, j < 0))\}$.

Observe that $B \models X$. ■