LGIC 010 Textbook

Scott Weinstein, Owain West, Grace Zhang

April 9, 2018

Contents

0 Prelude

0.1 Prerequisite Notation

Though the course has no specific mathematical prerequisites, a general familiarity with the set of integers and some of its basic properties will be assumed. We collect here some useful facts and notations that will appear from time to time throughout the course. We'll add more as the need arises.

1. Notations for important sets of numbers
   - $\mathbb{Z} = \{\ldots -2, -1, 0, 1, 2, \ldots\}$ (the integers)
   - $\mathbb{N} = \{0, 1, 2, \ldots\}$ (the non-negative integers, a.k.a. the natural numbers)
   - $\mathbb{N}^+ = \mathbb{Z}^+ = \{1, 2, 3, \ldots\}$ (the positive integers)
2. Important facts about numbers
   - The Least Number Principle: If X is a nonempty subset of $\mathbb{N}$, then X has a least element.
   - Principle of Mathematical Induction: If X is a subset of $\mathbb{N}$, and $0 \in X$, and if for every i, $i \in X$ implies $i + 1 \in X$, then $X = \mathbb{N}$.
   - The Pigeonhole Principle: If you distribute m pigeons into n pigeonholes and $m > n$, then some hole contains more than one pigeon.
3. Unique Factorization into Primes: Recall that $p \in \mathbb{N}^+$ is *prime* if and only if $p \neq 1$ and p is divisible only by 1 and p. Every $n \in \mathbb{N}^+$ with $n \neq 1$ can be written uniquely (up to reordering) as $p_1^{a_1} \cdots p_n^{a_n}$ where each p_i is prime and each $a_i \geq 1$.

0.2 A Combinatorial Warmup

Combinatorics is, roughly, the part of mathematics which deals with counting things. Its techniques are general, and its results tangible. Throughout this book, we will use combinatorial problems as concrete examples of problems which can be considered and solved by means of logical techniques. To get our feet wet, let's consider the following principle and question.

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.*

Example 1. *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 will demonstrate that the answer is no by an application of the Pigeonhole Principle.

Proof. 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. □

The above argument presupposes that there are finitely many Philedelphians. In fact, the theorem does not hold if we allow Philadelphia to have infinitely many people. As an exercise, try to describe a numerically diverse group of infinitely many Philadelphians.

The Pigeonhole Principle can take on a more general form, the *Mean Pigeonhole Principle*, which is as follows:

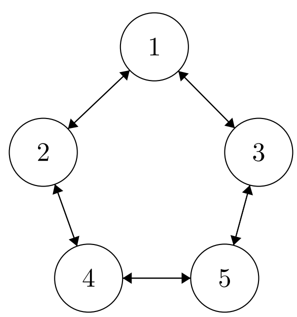
Principle 2. ***The Mean Pigeonhole Principle****: If you distribute m pigeons into n pigeonholes and $m \geq k \cdot n + 1$, then some hole contains at least $k+1$ pigeons.*

Note that Principle 1 is just the special case of Principle 2 for $k = 1$.

Example 2. *Say a group of people has three-mutuality if it contains either a group of three mutual friends or a group of three mutual strangers. How large a group of people can lack three-mutuality?*

We show that the largest such group has five members. To do this, we will give an example of a pattern of friendship among a group of five people that lacks three-mutuality, and show that every pattern of friendship among six or more people has three-mutuality. To show that every friendship pattern on six or more people lacks three-mutuality, we will use the Mean Pigeonhole Principle.

Proof. The diagram below shows a "friendship pentagon". Nodes represent people, and an edge between people represents friendship. It is easily checked that the diagram lacks 3-mutuality.



Next, we show that every group of $n \geq 6$ people must have 3-mutuality. Again, write p_{ij} to denote that i is a friend of j.

Let $G = \{1, \ldots, 6\}$ and sort the five people $2, \ldots, 6$ into two pigeonholes according to the truth value, true ($\top$) or false ($\bot$) of $p_{12}, \ldots, p_{16}$. That is, sort people $2, \ldots, 6$ into two groups, one group which are all friends of 1, and one group all of which are not friends with 1. By Principle 2, one of these holes, suppose it's the $\top$ one, contains at least three members of G.

Now, either two of these are friends, in which case they, together with 1 form a collection of three mutual friends, or none of them of friends, in which case they themselves form a collection of three mutual strangers. The argument is analogous in the case that three members of G were sorted into the $\bot$ pigeonhole. □

We might wonder whether every natural number n has a k such that every group of at least size k has n-mutuality. This happens to be true (try proving it!). The *Ramsey number* $R_{m,n}$ is the least k such that every set of k people must have either a group of m mutual friends or n mutual strangers. In the previous example, we showed that $R_{3,3} = 6$. Higher Ramsey numbers are much harder to compute. We know that

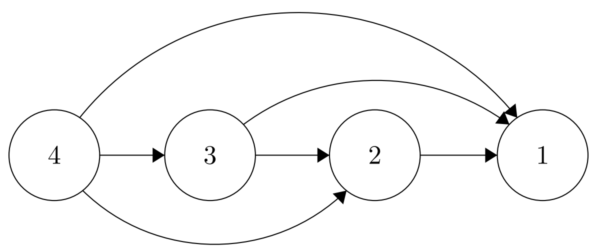
$R_{4,4} = 18$. $R_{5,5}$ is currently known to be between 43 and 48. $R_{6,6}$ is somewhere between 102 and 165.

As an exercise, prove that $R_{m,n} = R_{n,m}$ for all m, n.

"Suppose aliens invade the earth and threaten to obliterate it in a year's time unless human beings can find the Ramsey number for red five and blue five. We could marshal the world's best minds and fastest computers, and within a year we could probably calculate the value. If the aliens demanded the Ramsey number for red six and blue six, however, we would have no choice but to launch a preemptive attack." - Paul Erdos

---

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). 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, $\{1, 2, 3, 4\}$. 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 $\{1, 2, 3, 4\}$? Let's recycle the sentence letters and use p_{ij} to signify the statement that i loves j; note 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 can conclude 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.

$2^{16} = 65536$. It's kind of amazing that there are as many as 65,536 different potential love-scenarios at a table for four!

Friendship, as compared to love, is relatively tame in terms of the number of scenarios that might arise. Let's return to using p_{ij} to indicate that i and j are friends. 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.

In general, how many possible friendship scenarios are there among a group of n people? Well, every pair can either be friends or not friends, so there are 2^{num_pairs} possibilities. How many pairs are their, in terms of n?

0.3 Review

Concept Review

- **Pigeonhole Principle**: If you have $n+1$ pigeons and you try to fit them all into n holes, then there has to be at least one hole with $k > 1$ pigeons.
- **The Mean Pigeonhole Principle**: If you distribute m pigeons into n pigeonholes and $m \geq k \cdot n + 1$, then some hole contains at least $k+1$ pigeons.
- **Product Rule**: If there are n ways to do a first action and m ways to do a second action, there are $n \cdot m$ ways to do both action 1 and action 2.

Problems

1. Let X be a set, $|X| = n$ (we write $|X|$ for the size of the set X). How many subsets does X have?
2. How many subsets of even size does a set X of size $n > 0$ have?
3. Prove that the Cartesian plane cannot be colored using only two colors (Red/Blue) such that all points 1 unit away from each other are different colors.
4. Prove that for any set of $n \geq 2$ numbers, there are 2 numbers whose difference is divisible by $n - 1$.
5. Show that for any $n \in \mathbb{N}$, there is a number k whose base ten numeral contains only "5"s and "0"s such that k is divisible by n.

Solutions

1. There are 2^n many subsets of a set of size n. To see why this is the case, note that every element of the set can be either in or not in any given subset. Hence there are two choices for each of the n elements of the set, and by the product rule 2^n choices in total.

2. 2^{n-1}. We show that for every X of size at least one, the number of even-size subsets of X is equal to the number of odd-size subsets of X; it then follows from the result of the preceding problem that the answer is 2^{n-1}.

   First, suppose that the size of X is odd. Then complementation induces a one-to-one correspondence between the odd-size and even-size subsets of X. That is, we associate to each odd-size subset $Y \subseteq X$, the even-size subset $X - Y$. If, on the other hand, the size $n > 1$ of X is even, we argue as follows. Let a be an element of X and consider the set $W = X - \{a\}$. Since the size of W is odd, we already know that it has the same number of subsets of even-size as it does of odd-size; that is, there are the same number of subsets of X of odd-size that exclude a as there are subsets of X of even-size that exclude a. From this it follows at once that also X has the same number of sets of even-size that include a as it does subsets of odd-size that include a. Thus, X has the same number of subsets of odd-size as it does subsets of even-size.

3. Consider an equilateral triangle with unit-length sides. We have three points pairwise one-unit apart and only two colors. The answer follows by application of the pigeonhole principle.

4. Note that there are $n-1$ remainders when dividing by $n-1$. Hence by the pigeonhole principle two of our n numbers must have the same remainder when divided by $n-1$. Their difference is divisible by $n-1$.

5. Consider the first $n+1$ elements of the set $\{5, 55, 555...\}$. We know from above that this set has two numbers whose difference is divisible by n. Note that the difference of any two numbers in this set is written using only 5s and 0s.

1 Truth-Functional Logic

1.1 Introduction to 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.

Components of Truth Functional Logic

1. Language (the *Syntax*)
   (a) sentence letters
   (b) connectives
2. Interpretation (the *Semantics*)
   (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.

Statements are sentences whose truth or falsity is independent of context of utterance. For example, the sentence "I am bald" is not a statement, since the truth or falsity of a given utterance of this sentence depends not only on the speaker and the time of utterance, but also on whatever subtle contextual factors might partially restrict the the range of application of the vague term "bald." On the other hand, "eight times seven is fifty-four" is as a statement, since it's truth or falsity (in this case falsity) is independent of context of utterance. Neither of the sentences "is eight times seven fifty-four?" nor "please, let eight times seven be fifty-four," is a statement. Truth-functional logic deals with the truth or falsity of *statements* only, and we use sentence letters exclusively to schematize statements.

1.2 Basic Syntax of Truth-Functional Logic

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.

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

- Material Conditional

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

- Material Biconditional

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

The definitions of the truth-functional connectives suffice to determine the truth/falsity of a compound schema completely in terms of (eg as 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: $((p_{11} \land p_{22}) \land p_{33}) \land p_{44}$

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

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

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.

1.3 Basic Semantics of Truth-Functional Logic

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.*

Definition 2. *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 (and write $A \models S$) if and only if S receives the value $\top$ relative to the truth assignment A.*

Example 3. *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. ($A \not\models S$).*

Interpreting the Material Conditional

Let's return to our potential lovers and restrict attention to just two of them, 1 and 2. How could express the statement that all love is requited among these two sweethearts? The natural mode of expression is: if 1 loves 2, then 2 loves 1, and if 2 loves 1, then 1 loves 2. This is a perfect candidate for using the material conditional.

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

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

or, equivalently,

$$p_{12} \equiv p_{21}$$

In how many of the possible love scenarios among 1 and 2 is all love requited? Count the number of satisfying truth-assignments to R!

While the motivations for the truth-functional definitions for the other connectives normally seem evident to new logicians, the material conditional often gives people trouble. Let's consider generalized conditionals as a route to motivating the truth-functional interpretation of the conditional offered above. Of course, the statement "if an integer is divisible by six, then it is divisible by three," is true, and thence 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 material conditional's truth-table. 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.

An Inductive Proof

Let's do a simple inductive proof about truth-functional satisfaction, as an illustration of the use of mathematical induction, especially in application to reasoning about truth-functional schemata.

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 prove 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 $A'(q_{k+1}) = \bot$.

□

The Centrality of Satisfaction

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.

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.

Definition 3. S implies T *if and only if for every truth assignment A, if $A \models S$, then $A \models T$.*

Definition 4. S *is* equivalent *to T if and only if S implies T and T implies S*

Definition 5. S *is* satisfiable *if and only if for some A, $A \models S$.*

Definition 6. S *is* valid *if and only if every truth assignment satisfies S.*

Examples of equivalence and the material biconditional

Try to see why the following are equivalent - either by appealing to your understanding of what the connective "means" or by going back to the truth tables.

- $p \oplus q$ 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).
- $p \equiv q$ ius equivalent to $(p \supset q) \wedge (q \supset p)$

Note that both conjunction and inclusive disjunction are also commutative and associative, whereas the material conditional is neither.

Try to to think of examples of (binary) truth-functional connectives which are commutative but not associative, and associative but not commutative.

1.4 Review

Concept Review

Definitions

- A truth-assignment A for X is a function which maps every sentence letter $q \in X$ to either $\top$ or $\bot$. $A(q)$ is the notation for the value A associates with q.
- A schema S *implies* a schema T iff for all truth-assignments A, if $A \models S$ then $A \models T$.
- A schema S is *equivalent* to a schema T iff S and T are satisfied by exactly the same truth assignments (for all A, $A \models S$ iff $A \models T$).
- S is *satisfiable* iff there is a truth assignment that satisfies it (there exists an A such that $A \models S$)
- S is *valid* iff all truth assignments satisfy it (for all A, $A \models S$)

Syntax, Semantics The *syntax* of TF-logic is given by the rules for forming truth-functional schemata from sentence letters and connectives. The *semantics* of TF-logic are given by a *truth-assignment*, which associates with each letter a *truth-value*.

Satisfying Sentences The *truth-values* of the individual sentence letters in a schema are propagated to the whole schema by means of *truth-tables* which give fixed semantic interpretations to each of the *connectives*. We say that a truth-assignment A *satisfies* a sentence S (written $A \models S$) iff the sentence S evaluates to $\top$ under the truth-assignment A. Otherwise, we write $A \not\models S$ and say that A does not satisfy S.

Problems

1. Is "the University of Pennsylvania has a Logic major" a statement? Why or why not?
2. Is "should I major in Logic?" a statement? Why or why not?
3. Using the sentence letters p_{ij}, $q \leq i, j \leq 4$ to stand for "person i loves person j". Schematize the following statements:
   (a) Person 1 loves everyone else.
   (b) There is a Shakespearean love triangle (*i.e.*, no one has their love requited) between people 1, 2, 3, and person 4 is a Scrooge (he does not love anyone, even himself).
   (c) Everyone loves, exclusively, people with numbers lower than themselves.
4. How many truth-assignments to the given letters satisfy the following schema?
$$(p_1 \supset q_1) \wedge ... \wedge (p_5 \supset q_5)$$
5. How many truth-assignments to the set of sentence letters $X_n = \{p_1, q_1, \ldots, p_n, q_n\}$ satisfy the following schema S_n? Express your answer as a function of n and prove that it is correct by mathematical induction.
$$(p_1 \supset q_1) \wedge ... \wedge (p_n \supset q_n)$$
6. How many truth-assignments over the given letters satisfy the following schema?
$$p_1 \oplus p_2 \oplus p_3 \oplus p_4 \oplus p_5$$
7. Is the following sentence valid, satisfiable but not valid, or unsatisfiable?
$$(a \equiv b) \supset (a \vee \neg b)$$
8. Valid, satisfiable, or unsatisfiable?
$$(b \vee (b \supset a)) \wedge (\neg b \vee (a \supset b))$$
9. Valid, satisfiable, or unsatisfiable?
$$(a \equiv b) \wedge (b \equiv c) \wedge (a \oplus b)$$
10. How many truth-assignments for the given letters satisfy
$$(a \equiv b) \wedge (b \equiv c) \wedge (c \equiv d)$$
11. How many truth-assignments to the given letters satisfy
$$(a \oplus b) \vee (b \oplus c) \vee (c \oplus d)$$
12. I claim that if n people all shake hands with each other (once per pair), the total number of handshakes is $\frac{n(n-1)}{2}$. Prove this by induction.

Solutions

1. Yes, it is.[a]

2. No, it is not. It is not a statement because it expresses a question, which is not determinitely true or false.

   With that being said, you should - of course - major in logic[b].

3. (a) $p_{11} \wedge p_{12} \wedge p_{13} \wedge p_{14}$

   (b) $((p_{12} \wedge p_{23} \wedge p_{31}) \vee (p_{13} \wedge p_{21} \wedge p_{32})) \wedge \neg(p_{41} \vee p_{42} \vee p_{43} \vee p_{44})$

   (c) $p_{21} \wedge p_{31} \wedge p_{32} \wedge p_{41} \wedge p_{42} \wedge p_{43}$

4. 3^5. Note that each of the terms of the form $p_i \supset q_i$ is satisfied in three cases (check the truth table) and apply the product rule.

5. 3^n for $n \geq 1$.

   BASE CASE: Verify, via the truth table for the material conditional, that three of the four truth assignments to $X_1 = \{p_1, q_1\}$ satisfy the schema S_1.

   INDUCTION STEP: Suppose that 3^n of the 4^n truth-assignments to X_n satisfy S_n:

$$(p_1 \supset q_1) \wedge ... \wedge (p_n \supset q_n).$$

   Let A be one such truth-assignment. Verify, using the truth-table for the material conditional, that A may be extended to exactly three distinct truth-assignments to the sentence letters X_{n+1} each of which satisfies S_{n+1}. It follows that there are $3 \cdot 3^n = 3^{n+1}$ truth-assignments to the sentences letters X_{n+1} that satisfy S_{n+1}:

$$(p_1 \supset q_1) \wedge ... \wedge (p_n \supset q_n) \wedge (p_{n+1} \supset q_{n+1}).$$

6. $2^4 = 16$. Remember that there are 2^{n-1} ways to pick an odd-sized subset from n elements and that a sentence of the given form is satisfied iff an odd number of sentence letters are set to true.

7. This is valid. Suppose A is a truth-assignment to the sentence letters a and b. Note that if $A(a \equiv b) = \bot$, then A satisfies the given schema. So suppose $A(a \equiv b) = \top$. Then $A(a) = A(b)$, hence either $A(a) = \top$ or $A(b) = \bot$. Thus $A(a \vee \neg b) = \top$.

8. Valid. Suppose A is a truth-assignment to the sentence letters a and b. If $A(b) = \top$, then A clearly satisfies the left conjunct. If $A(b) = \bot$, then $A(b \supset a) = \top$, hence A satisfies the left conjunct as well. Similarly, if $A(b) = \top$, then A satisfies the right conjunct, and if $A(b) = \bot$, then $A(\neg b) = \top$, hence again A satisfies the right conjunct.

9. Unsatisfiable. Suppose A is a truth-assignment to the sentence letters a, b and c and A satisfies both $a \equiv b$ and $b \equiv c$. It follows that A satisfies $a \equiv c$ (in other words, $\equiv$ is *transitive*). But then A does not satisfy $a \oplus c$, since this is truth-functionally equivalent to $\neg(a \equiv c)$. So the schema is unsatisfiable.

10. 2. Picking true/false for a fixes the truth-values of the remaining letters.

11. 14. To get this answer, we note that there are 16 (2^4) truth-assignments in total; count the number which do not satisfy our sentence, and subtract that number from 16. The sentence is only not satisfied when each of a, b, c, d have the same truth-value, so there are 2 non-satisfying truth-assignments. This means there are $16 - 2 = 14$ satisfying truth assignments.

12. BASE CASE: $n = 2$. Two people shaking hands results in one handshake, and the formula gives us $\frac{2(2-1)}{2} = 1$ which is correct. Note that we pick $n = 2$ as the base case (not $n = 0$ or $n = 1$) because it doesn't really make sense to talk about those cases (since you need two people for a handshake).

    INDUCTIVE CASE: Assume that for n people, the number of handshakes (let's denote it H_n) is $H_n = \frac{n(n-1)}{2}$. We want to show (henceforth "wts") that for $n + 1$ people the number of handshakes is $H_{n+1} = \frac{(n+1)n}{2}$. The number of handshakes between $n + 1$ people is clearly the number of handshakes for n people (H_n) plus n, since our new person must shake hands with the n others. So we have $H_{n+1} = H_n + n = \frac{n(n-1)}{2} + n = \frac{n^2-n+2n}{2} = \frac{n^2+n}{2} = \frac{(n+1)n}{2}$, which is what we wanted to show.

[a] Although, one might insist that there remains an element of context dependence owing to an ambiguity in the proper name "University of Pennsylvania" - those in Indiana County, Pennsylvania might well use it to refer to a different institution. This observation invites reflection upon the intriguing question whether (virtually) all sentences of ordinary language are to some extent context dependent (at least without non-ordinary supplementation).

[b] Provided you like it and want to.

1.5 Expressive Completeness of Truth Functional Logic

Definition 7. *We use the symbol* $:=$ *to mean "is defined to be equal to".* $:=$ *expresses a* definition *of equality, whereas* $=$ *expresses a statement about equality.*

If you're a coder, $x := 10$ *is to logic/math as* `let x = 10` *is to JavaScript, whereas* $x = 10$ *is to logic as* `x === 10` *is to JavaScript.*

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 will use 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.

Expressive Completeness

In the last section, we used the notion of the proposition expressed by a schema as an intuitive vehicle for pursuing the investigation of the expressive power of truth-functional schemata. 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 provide the following inductive definition of $\mathbb{S}(X)$.

Definition 8. *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* $\mathbb{U}$.[1]

This is simply a formal way of saying that all of our sentences have to use only the letters from X and must be "well-built" in the sense that each connective has the correct number of arguments, with all the bracketing done correctly. For example, with $X := \{p, q, r\}$ then $S_1 := ((p \vee q) \wedge r)$ is well-built, whereas $S_2 := \vee p \wedge q$ is not.

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 $\mathbb{P}_X(S) = \mathfrak{P}$, *i.e.*, can every proposition be expressed by some schema? In other words, is truth-functional logic *expressively complete*?

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* $\mathbb{P}_X(S) = \mathfrak{P}$.

[1] 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.

This looks complicated, but it really isn't. In natural language, what it's saying is this: *pick any subset* $\mathfrak{P}$ (your proposition) *of truth assignments for a set of sentence letters X. Then there is a truth-functional schema S using only letters from X ($S \in \mathbb{S}(X)$) which is true of exactly those truth-assignments which are in $\mathfrak{P}$ ($\mathbb{P}_X(S) = \mathfrak{P}$).* In other words, *every proposion can be "picked out" by some schema.* This is why it's called *expressive completeness*: truth-functional logic is "expressively complete" in that it can express every such proposition.

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

Definition 9. *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.

Example 4. *Let $\Lambda = \{a, b, c\}$. Then $\bigwedge \Lambda = a \wedge b \wedge c$, and $\bigvee \Lambda = a \vee b \vee c$.*

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 $A' = A$. $\square$

Once you become a bit more familiar with the terminology, things will become much easier. Indeed, this lemma is really simple - in plain English, it says that *for every truth assignment, there is a schema which only uses logical ANDs and NOTs that is satisfied by exactly that truth assignment.* When stated like that, of course, it seems obvious - if your truth assignment assigns true to p you should have p in your schema, and if your truth-assignment assigns false to p, your schema should include $\neg p$, with all the literals joined up together by ANDs.

The proof expresses that intuition symbolically - make sure you can understand the proof now, going over the relevant terminology and symbols if necessary. If you get stuck trying to interpret all that logical symbolism, please come into Office Hours and we'll be happy to help! Logic won't be any fun if the symbolism gets in the way of your understanding, so it's best if you take the time to get comfortable with all that at the start.

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 $\mathbb{P}_X(S) = \mathfrak{P}$. $\square$

Once again, the main difficulty here is the symbolism - the proof expresses a simple intuition in symbolic form. Try rewriting this proof in your own words!

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

Corolloraries are Theorems which follow very simply or quickly from another (often proved-right-above) theorem. Whenever you see a corollary in a math textbook or notes, you should always make sure you understand why it's a consequence of the just-proven theorem!

1.6 The Power of a Truth-Functional Schema

We will introduce the following useful terminology.

Definition 10. *For the following, 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.*

Example 5. *Let's consider $X = \{p, q, r\}$. What is the length of a longest succinct list of truth-functional schemata over X? We will arrive 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 (each letter can be assigned true or false, and you make that choice for each of the three letters). 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 $2^{2^3} = 2^8 = 256$.*
- *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.*

So the longest such list is of length 256.

Example 6. *Let's compute the power, as defined above, of $p \wedge (q \vee r)$. 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.

Why is it that a schema S implies a schema S' if and only if $\mathbb{P}(S) \subseteq \mathbb{P}(S')$? Go back to the definition of both $\mathbb{P}(S)$ and schema implication if you need to.

Here's a simple way to think about it, once you know the definitions: if S implies S' then there is no truth-assignment that satisfies S but does not satisfy S' (otherwise S wouldn't imply S', by the definition of schema implication). Hence every truth-assignment satisfying S, also satisfies S', or symbolically, $\mathbb{P}(S) \subseteq \mathbb{P}(S')$. When written that way, it seems really simple!

Often throughout your study of logic you will see things which, at the surface, look confusing like the statement we just considered. Make sure to always take the time to go back to definitions and understand things in your own words - it'll make logic *much* more satisfying.

Example 7. *Let's 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* $0 \le i \le 8$.

Generalizing our last example, 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 $S = 2^{(2^n-|\mathfrak{P}|)}$.

Example 8. *What is the length of a longest succinct list of truth-functional schemata over* $X := p, q, r$ *each of which has power 32?*

Make sure you have all the relevant definitions in order - what does it mean for the power of a schema to be 32? What does it mean for a list of schemata to be succinct?

Well, from the definitions we know that a schema over $X := \{p, q, r\}$ *has power 32 if and only if exactly three truth assignments satisfy it (why?). 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 (why a set of size 8, given that we have 3 sentence letters?). In the next section, we'll take a break from logic proper to learn a bit about how we would determine how many such subsets there are.*

A Combinatorial Interlude

Our leading question from the end of the last section brings us 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 will need notation for each, so let $(n)_k$ 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.

Let's first evaluate $(n)_k$, the number of ordered sequences of size k you can pick from a set of size n. 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.

Consider picking a panel of three students from a class of 10. How many ways can you do this? Is it the same as the number of ways you could pick 7 of the 10 students to *not* be on the panel, using the non-picked students for the panel?

Let's not forget how this all began. Since the The length of the longest succinct list of schemata with power 32 is number of subsets of size three contained in a set of size eight, it follows that the length of the longest such list is $\binom{8}{3} = 56$.

Counting the Length of an "Implicational Anti-Chain"

Let's use our newfound 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? Note 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, if 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 follows immediately by evaluating

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

Prove 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.

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, which follows from the famous Sperner's Theorem.[2]

[2]See Van Lint and Wilson, *A course in combinatorics*, Chapter 6: Dilworth's theorem and extremal set theory.

1.7 Is There An Efficient Decision Procedure For Truth-Functional Logic?

It is easy to see 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.

Come up with an algorithm for checking whether $A \models S$ for $A \in \mathbb{A}(X)$ and analyze its runtime complexity as a function of the length (in terms of the number of connectives) of 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, called the $P = NP$ problem, is generally regarded as one of the most significant open mathematical problems of our time, and carries with it a $1,000,000 prize for its solution as well as eternal mathematical glory. For further information visit:

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

1.8 Review

Concept Review

Definitions

- $\mathbb{A}(X)$ is the set of all truth assignments over X.
- $\mathbb{P}_X(S) = \{A | A \in \mathbb{A}(X) \text{ and } A \models S\}$ is *the proposition expressed by* S. It's the set of truth assignments that satisfy S (where truth assignments are restricted to those for sentence letters in the set X).
- A schema S *implies* a schema T iff for all truth-assignments A, if $A \models S$ then $A \models T$. In other words, S implies T iff the proposition expressed by S is a subset of the proposition expressed by T.
- A schema S is *equivalent* to a schema T iff S and T are satisfied by exactly the same truth assignments (for all A, $A \models S$ iff $A \models T$). In other words, S and T are equivalent if they express the same proposition.
- A list of TF-schemata is called *succinct* iff no two schemata on the list are equivalent.
- The *power* of a schema S relative to a set of sentence letters X that includes all those with occurrences in S is the length of the longest succinct list of schemata drawn from $\mathbb{S}(X)$ which S implies. (When X is unspecified, we suppose it to be exactly the set of sentences letters which occur in S.)

Fun With Counting There are $n!$ ways to order a list of n items. To see why, note that there are n choices for the first element, $n-1$ for the second, $n-2$ for the third, resulting in $n(n-1)(n-2)...(1) = n!$ orderings.

There are $(n)_k := \frac{n!}{(n-k)!}$ ways to pick an ordered list of k elements from n elements, $k \leq n$. As before, there are n choices for the first thing, $n-1$ for the second, all the way down to $n-k+1$ for the k^{th}. This gives us the answer $\prod_{i=n-k+1}^{n} i = \prod_{i=1}^{n} i / \prod_{i=1}^{n-k} i = \frac{n!}{(n-k)!}$

There are $\binom{n}{k} := \frac{n!}{(n-k)!k!}$ ways to pick a subset of k elements from n elements, $k \leq n$. There are $(n)_k$ ordered lists of size k from n. Since each subset of size k corresponds to $k!$ of those ordered lists, we divide out by $k!$ to get $\frac{n!}{(n-k)!k!}$, for which we use the notation $\binom{n}{k}$, read as "n choose k".

Expressive Completeness

For any (arbitrary) proposition, there is a truth-functional schema which expresses that proposition. We noted that a schema can pick out individual truth-assignments by conjoining literals for each of the sentence letters (for example, the truth assignment A_1 which maps $p = \top, q = \top, r = \top$ is picked out by the sentence $(p \wedge q \wedge r)$). Sentences of this form are called *terms.* We further noted that a disjunction of such terms (one for each truth-assignment in our proposition) was sufficient to express any proposition.

Power

Suppose we have a sentence S over n sentence letters which is satisfied by k truth assignments. Then the power of S is 2^{2^n-k}. To see why this is the case, note that there are 2^n truth assignments for n sentence letters. If S is satisfied by k truth assignments, then those truth assignments must also satisfy T, if S implies T. So we can't "choose" whether or not to include any of those k truth-assignments in the proposition expressed by T, because $\mathbb{P}_X(T)$ must include them. So we are left with $2^n - k$ truth-assignments, and since each of these $2^n - k$ truth assignments can be either in or out of the proposition expressed by T, the power of S is then 2^{2^n-k}.

Problems

Let $X = \{p_1, p_2, p_3, p_4\}$.

1. What is the power of $p_1 \equiv p_2$ relative to X?
2. What is the length of the longest succinct list of schemata drawn from $\mathbb{S}(X)$ no two of which have the same power?
3. What is the length of the longest succinct list of schemata drawn from $\mathbb{S}(X)$ each having power 256?
4. What is the largest n such that the conjunction of any two schema of power n drawn from $\mathbb{S}(X)$ is satisfiable?
5. How many ways can you choose 3 marbles from a bag of 15 marbles, assuming the marbles are all distinct? How many ways to take out all 15 marbles from the bag, one by one?
6. How many ways are there to arrange 10 people around a circular table, if we don't count rotations of the same order as being different?
7. Is there a schema of power 22? If so, give one. If not, explain why it's not possible.
8. What is the length of the longest succinct list of schemata drawn from $\mathbb{S}(X)$ each having power greater than 256?

Solutions

1. $2^8 = 256$. For four sentence letters, $p_1 \equiv p_2$ has $2^3 = 8$ satisfying truth assignments. To see why this is the case, note that given a choice for p_1, p_2 is fixed. So we have two choices for p_1, one choice for p_2, and two choices each for p_3 and p_4.

   Plugging in to our formula, we find that the power is $2^{2^4-8} = 2^{16-8} = 2^8$.

2. 17. The power of a schema S on n sentence letters with k satisfying truth assignments is 2^{2^n-k}. k can take any value from 0 through 16 inclusive when $n = 4$ (since we have $2^4 = 16$ truth-assignments), meaning that the power can be any one of $2^{16}, 2^{15}, ..., 2^0$.

3. $\binom{16}{8}$. A schema on four sentence letters has power $256 = 2^8$ when it is satisfied by 8 truth assignments (because $2^{2^4-8} = 2^8$). Since we have $2^4 = 16$ total truth assignments, the number of pairwise inequivalent schemata each satisfied by 8 truth-assignments is the number of subsets of size 8 drawn from a set of size 16, which is $\binom{16}{8}$.

4. $n = 2^7 = 128$. With four sentence letters, we have $2^4 = 16$ truth assignments. A schema that has power 2^7 is satisfied by $16 - 7 = 9$ truth assignments. By the pigeonhole principle, two schemata of power 2^7 (hence both satisfied by 9 things) must have some satisfying truth-assignment in common (because $9 + 9 = 18 > 16$). Hence the conjunction of any two schemata of power 2^7 is satisfiable, because there must be a truth assignment that satisfies them both.

   Note that 2^7 is the highest power that works, because being satisfied by less than 9 truth-assignments (therefore having a greater power) would mean that the two sentences need not have a satisfying truth-assignment in common. For example, if both sentences were satisfied by 8 truth assignments each, those sets of satisfying truth-assignments could be disjoint, hence the conjunction of the two sentences would not be satisfiable.

5. $\binom{15}{3}$, 15!

6. 9!. There are 10! ways to order 10 people around the table, but that considers different rotations of the same order as different seating arrangements. Since there are 10 rotations of any such ordering, we divide 10! by 10, giving us the answer 9!.

7. No. The power of a schema is always some power of 2. 22 is not a power of 2.

8. $\sum_{i=0}^{7} \binom{16}{i}$. We have $2^4 = 16$ total truth-assignments. The power of a schema S on four sentence letters is greater than $256 = 2^8$ when S is satisfied by less than 8 truth-assignments (because our formula for power is 2^{2^n-k} with $n = 4$ in this case, hence power is greater than 2^8 when k is less than 8). Hence our answer equal to the number of schemata that express a proposition of size 0, plus the number that express a proposition of size 1.... plus the number that express a proposition of size 7. Remember that $\binom{n}{k}$ represents the number of size-k subsets from n things, and since propositions are simply subsets of truth-assignments, we arrive at our answer $\sum_{i=0}^{7} \binom{16}{i}$.

2 Monadic Quantification Theory

2.1 Introduction to Monadic Quantification Theory

It's now time to graduate from our humble beginnings in Truth-Functional Logic. We will now begin to consider a more expressive logic, which we'll call *Monadic Quantification Theory*[3]. This is desirable because 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, consider the following statements:

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

These statements share the *monadic predicate*[4] "◯ 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.

Definition 11 (Extension of a Monadic Predicate)**.** *The* extension *of a monadic predicate is the collection of objects of which the monadic predicate is true. For example, the extension of the monadic predicate "◯ is an even natural number" is the set* $\{0, 2, 4, 6...\}$.

You can think of the monadic predicate as "picking out" some subset of what you're talking about (your "universe of discourse"). The subset which the monadic predicate "picks out" is its extension.

What is the extension of the monadic predicate "◯ is a prime number less than 10"? What is the extension of the monadic predicate "◯ is an even prime number"?

Note that distinct monadic predicates might have the same extension - for example, the extensions of "◯ is a warm–blooded reptile" and "◯ is an even prime number greater than two" are the same, namely, they are both the emptyset. 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*. We will focus solely on extensional contexts. Our focus on extensional contexts is the natural continuation of our earlier focus on truth-functional contexts.

[3]An alternative name for this logic might be *Monadic First-Order Logic*

[4]Also known as a *unary predicate*.

2.2 Syntax

Open Sentences

Consider again the argument that "Lassie is a collie, and all collies are mortal. *Therefore* Lassie is mortal". 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."

We can achieve this kind of generality 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. For example, "x is a collie" is an open sentence.

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 $x := 16$ and false with respect to the assignment of $x := 17$. This gives a good justification of why we use the word "open" - ie, the truth of the sentence is an "open question" in absence of information about x.

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

- $\frac{x}{6} = 0 \supset \frac{x}{3} = 0$.
- x is a collie and x weighs less 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.

The Existential Quantifier

Consider the statement that "there is a prime number". How would we express this? Supposing we had a sentence $P(x)$ which says that x is prime, we would want to say something along the lines of "there is an x such that $P(x)$"[5]. In order to do this, we introduce the existential quantifier $\exists$. Our sentence, "there exists an x such that $P(x)$" can then be written as

$$(\exists x)(P(x))$$

We say that the quantifier here *binds* x. In general, a quantifier Qx binds every instance of x in the outermost parentheses following it.

Note that $(\exists x)(P(x))$ has a truth-value, without any assignment to x. This is because every variable in the sentence is *bound* by a quantifier, and so no assignments need to be made. We call a sentence in which every variable is bound a *closed* sentence. 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$)$.

The Universal Quantifier

Let's now consider the universal quantifier, which allows us to say that a property holds of "everything". For example, we can render the statement

[5] Note that $P(x)$ is an open sentence.

- 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 iff for any integer assignment to x, the open statement within the square brackets is satisfied. In other words, the statement is true for any integer substution for x. Given this interpretation, we are justified in reading the above sentence as "for all x, x is even or x is odd".

Note that context determines our *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.

Monadic Schemata

As we did in the case of truth-functional logic, we will introduce a schematic language for monadic quantificational logic. In this case, we use capital letters such as F, G and H to schematize monadic predicates (we call these *monadic predicate letters*, and lowercase letters such as x, y and z as variables. We specify the following categories of monadic schemata.

- An *atomic schema* is an expression of the form Fx where F is a monadic predicate letter and x is a variable.
- A *one variable open schema* is a truth functional compound of atomic schemata all with the same variable x.
- 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.

2.3 Semantics

We now introduce *structures* as interpretations of monadic schemata. These play the role that truth-assignments played in the context of truth-functional logic, in that they bridge the gap between the syntactic objects of our (newly strengthened) language and their truth-values.

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.

Item (1) specifies a *universe of discourse*, that is, a collection of objects over which our variables of quantification range. Item (2) specifies the extension of each monadic predicate letter occurring in any schema under consideration. Item (3) makes sure that we assign definite values to free variables, if there are any, so that we can evaluate our sentence's truth.

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."

Validity, satisfiability, implication, and equivalence

We extend the notions of validity, satisfiability, implication, and equivalence to (closed) 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.

A schema being valid means it is true in all possible interpretations, or as some would say, "all possible worlds". A schema is satisfiable if it's true in at least one possible world.

2.4 Counting Satisfying Structures

Let's consider the problem of how to count the number of structures with a fixed universe of discourse that satisfy a given schema. For example, 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

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

From now on, we will use the notation $[n] := \{1, 2, ...n\}$

Observe that a structure A satisfies S if and only if $F^A \subseteq G^A$. So we need to determine the number of pairs of sets Y, Z such that $Y \subseteq Z \subseteq [6]$. Let's call this number N. We proceed to compute N as follows.

First, recall that for every $0 \le i \le 6$, the number of sets $Z \subseteq [6]$ of size i is $\binom{6}{i}$. Second, recall that that the number of subsets of a set of size i is 2^i. It follows that the number of pairs Y, Z with $Y \subseteq Z \subseteq [6]$, for sets Z of size i is $\binom{6}{i}2^i$. Therefore,

$$\begin{aligned} N &= \sum_{i=0}^{i=6} \binom{6}{i} 2^i \\ &= \sum_{i=0}^{i=6} \binom{6}{i} 2^i \cdot 1^{6-i} \\ &= (2+1)^6 \\ &= 3^6 \end{aligned}$$

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

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.

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 $A \not\models R^*$.

Note that R implies R^* iff the number of counterexamples as defined above is zero.

Let's continue with the preceding example and count the number of counterexamples to the claim that the schema S implies the schema

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

Again, let's 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. So the number of counterexamples to the claim that S implies T (ie, structures which satisfy S but not T) is $3^6 - 2^6$.

2.5 Decidability

Our next order of business is to establish the decidability of pure monadic schemata, just as we did for truth-functional schemata.[6] Our approach introduces notions that we will elaborate further, when we turn to study polyadic quantificational logic.

Three views of structures

Note 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.

Show that these three views are equivalent by giving a correspondence between them. For example, x being of type $F \wedge \neg G$ in the Types View corresponds to the statement that $x \in$ the extension of $F \wedge x \notin$ the extension of G in the Canonical view.

The small model theorem

We will prove the following *Small Model Theorem* for monadic logic; the decidability of the 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 $A \models S$.*

Why is decidability a corollary to Theorem 2? As a hint, think about why truth-functional logic is decidable.

Monadic similarity

The proof of Theorem 2 rests on two lemmas; in order to state these lemmas, we first need to introduce some new concepts. In what follows, we will, without loss of generality, restrict our attention to monadic schemata in which only the predicate letters F and G occur.[7]

Definition 12. *We say that two structures A and B are* monadically similar *and write $A \approx_M B$ if and only if they satisfy exactly the same pure monadic schemata.*

Show that monadic similarity is an *equivalance relation*, *i. e.*, it is reflexive ($A \approx_M A$), symmetric (if $A \approx_M B$, then $B \approx_M A$), and transitive (if $A \approx_M B$ and $B \approx_M C$, then $A \approx_M C$).

We now turn towards developing the machinery required to establish our lemmas.

[6] See Warren Goldfard, *Deductive Logic*, Chs. 25-26 for an alternative treatment of the decidability of pure monadic schemata.

[7] The restriction to two monadic predicate letters is simply for pedagogical purposes. The generalization to n predicate letters is straightforward, as we observe below.

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 sometimes 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.

Intuitively, a homomorphism is a function that loosely, *"preserves the arrangement"* of elements in its domain, *i.e.*, elements which are of type P get mapped to an element in the range also of type P, *etc.*

Example

As an example, consider the following structures.

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

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

Note that A and B both have the same regions occupied in their respective Venn diagrams, ie F^A and F^B are both nonempty, as are both G^A and G^B. However, there is no homomorphism from A onto B, nor any homomorphism from B onto A.

Prove the last assertion.

Although A and B are not homomorphic, we will shortly see that A and B have a common surjective homomorphic image, *i.e.*, that there is a structure C such that there is a homomorphism from A onto C and a homomorphism from B onto C.

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,

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

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. 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, $A \models S$.

The case of universal quantification is handled similarly □

Write out the universal case formally. The argument should be very similar to the existential case, so make sure you understand that!

Types and monadic similarity

We recall our discussion of 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 say that a structure *realizes* a given type T_i just in case it makes the pure schema $(\exists x)T_i$ true (*i.e.*, there is at least one element of type T_i). Moreover, we say that $a \in U^A$ *realizes* T_i in A if and only if $A \models T_i[x|a]$.

Example 9. *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. For future reference, we list all fifteen of these structures as $A_1, \ldots, A_{15}$. Note that for every i, if A_i realizes a given type T, then there is exactly one element of A_i that realizes T. For this reason, we call these structures small models *- they are of minimal size among those structures that satisfy a given set of types.*

Lemma 3 provides 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: First, the forward direction. Suppose A, B realize the same types. Then there is a single structure C which is a surjective homomorphic image of both A and B – indeed, we may choose C to be that A_i as defined above, which realizes exactly the same types as A and B. Since A_i has exactly one element that realizes each of these types, we map every element that realizes a given type in A (or in B) onto the unique element that realizes that type in A_i. These maps are clearly surjective homomorphisms.

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 (as monadic similarity is an equivalent relation, and hence symmetric and transitive).

The reverse direction follows immediately from the fact that realization of a type is expressed by a pure monadic schema. □

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 9, that for every pure monadic schema S involving only the monadic predicate letters F and G, if S is satisfiable, then there is an $1 \leq i \leq 15$ such that $A_i \models S$. To conclude the proof, recall that for every i, $U^{A_i} \subseteq [4]$. In general, suppose that S is a schema involving only the monadic predicate letters "F_1,"... "F_n,." Let $t = 2^n$ and $k = 2^t - 1$. (In this case, t is the number of types, and k is the number of structures up to monadic similarity.) In like fashion, we can construct a list of k "small models," $A_1, \ldots, A_k$, each with universe of discourse a nonempty subset of $[t]$, such that if S is satisfiable, then for some i, $A_i \models S$. □

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

Corollary 3. *For all pure monadic schemata S and T involving only the monadic predicate letters F and G,*

S implies T if and only if

$$\{i \mid A_i \models S \textit{ and } 1 \leq i \leq 15\} \subseteq \{i \mid A_i \models T \textit{ and } 1 \leq i \leq 15\}, \textit{ and}$$

S and T are equivalent if and only if

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

Show that these are all quick corollaries to the proof of the Small Model Theorem.

2.6 Expressive Power

The expressive power of monadic quantification theory

With these results in hand, we proceed to analyze the expressive power of monadic schemata. For simplicity's sake, we'll continue to focus on the vocabulary consisting of the monadic predicate letters F and G. First, some definitions.

- 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.

Now, the main question: how expressive is MQT?

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 that the length of a longest such list is 2^{15}, since a schema is determined, up to equivalence, by which of the structures $A_1, \ldots, A_{15}$ satisfy it.

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

Answer: It follows from Corollary 3 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 13. *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 $U^A = [n]$.

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 $\hat{A} = \{B \in \mathbb{V} \mid B \approx_M A\}$.

$\hat{A}$ is the monadic similarity *equivalence class* of A, *i.e.*, all structures which are monadically similar to A. Generally, an equivalence class of an object N is the set of all objects which are equivalent to N under some equivalence relation.

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 compute these sizes:

- If A realizes exactly 1 type, then the size of $\hat{A}$ is 1, since monadically similar structures realize the same types, and there is only 1 way to place 4 elements into one given type. There are $\binom{4}{1}$ structures in $\mathbb{V}$, up to monadic equivalence, satisfying exactly 1 type (since there are 4 choices of which type is realized).
- If A realizes exactly 2 types, then the size of $\hat{A}$ is $2^4 - 2$ (since there are $2^4 - 2$ ways of distrubuting 4 elements into two given types such that each type is nonempty). There are $\binom{4}{2}$ structures in $\mathbb{V}$, up to monadic equivalence, satisfying exactly 2 types (since there are $\binom{4}{2}$ choices for which two types are realized).
- If A realizes exactly 3 types, then the size of $\hat{A}$ is $\binom{4}{2} \cdot 3!$ (since there are $\binom{4}{2} \cdot 3!$ ways of distrubuting 4 elements into three given types such that each type is nonempty). There are $\binom{4}{3}$ structures in $\mathbb{V}$, up to monadic equivalence, satisfying exactly 3 types (since there are $\binom{4}{3}$ choices for which three types are realized).

- If A realizes exactly 4 types, then the size of $\hat{A}$ is 4! (since there are 4! ways of ordering the 4 elements, with the i^{th} element in our order being places in T_i). There are $\binom{4}{4} = 1$ structures in $\mathbb{V}$, up to monadic equivalence, satisfying exactly 4 types.

If the size of $\hat{A}$ is confusing for any of the above, try to count these yourself! If you're still stuck, come into Office Hours and we'll be happy to help.

By Theorem 3, if $A \models S$ then for all $A_i \in \hat{A}$, $A_i \models S$. It follows that the answer to Question 3 is 1; in particular, one such list consists of the single schema

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

2.7 Review

Concept Review

Definitions

- A *one variable open schema* is a truth-functional compound of atomic schemata with the same variable x (for example, $(Fx \wedge Gx)$).
- A *simple monadic schema* is a schema in which some quantifier binds a one-variable open schema (for example, $(\forall x)(Fx \wedge Gx)$).
- A *pure monadic schema* is a truth-functional compound of simple monadic schemata (for example, $(\forall x)(Fx \wedge Gx) \vee (\exists x)(Gx)$).
- A schema S is satisfiable iff there is at least one structure A such that $A \models S$. Note that this is analogous to the corresponding definition for TF-logic, the only difference being that A is now a structure, not a truth-assignment.
- A schema S is valid iff for all structures A, $A \models S$
- A schema S implies a schema T iff for all A, if $A \models S$, then $A \models T$
- Schemata S and T are equivalent iff they are satisfied by exactly the same structures.
- A structure A is said to "be a counterexample to the claim that a schema S implies a schema T" iff $A \models S$ and $A \not\models T$.
- A structure A is said to "witness the inequivalence of schemata S and T" iff ($A \models S$ and $A \not\models T$), or ($A \models T$ and $A \not\models S$).
- Structures A, B are said to be *monadically similar* ($A \approx_M B$) iff they satisfy the same pure monadic schemata.
- A function h is *surjective* (or *onto*) if everything in the codomain is mapped to by h (equivalently, the image of the function - which is the set of all things that get mapped to - is equal to the codomain). In the language of first order logic, the criterion for surjectivity is

$$(\forall b \in B)(\exists a \in A)(h(a) = b)$$

- A *homomorphism* h *from* A *onto* B satisfies the following three properties:
  - h is a function from A to B.
  - h is *surjective (onto).*
  - h is "structure preserving". This just means that the extensions of predicate letters are preserved under the homomorphism. You can think of this as preservation of the types of elements in the "Type View," or preservation of sections in the Venn diagram in the "Venn View." In the language of first order logic, the criterion for this is

$$\text{for all predicate letters } P,\ (\forall a \in A)(a \in P^A \equiv h(a) \in P^B)$$

Quantifiers $\forall x$ is read "for all x", "for every x", or "for each x". A sentence of the form

$$(\forall x)(\text{some statement about } x)$$

is true just in case "some statement about x" is true no matter what x is.

$\exists x$ is read "there exists an x". A sentence of the form

$$(\exists x)(\text{some statement about } x)$$

is true just in case "some statement about x" is true of at least one thing x.

(Unary) Predicates *Predicates* are true of some things and not true of others. They correspond to the "some statement about $\bigcirc$" segment of the aforementioned sentences. For example, "$\bigcirc$ is an even number" is a predicate that is true just of the elements of the set of even integers, and false of all other things. The set of things of which a predicate is true is called its *extension*. For example, the extension of "$\bigcirc$ is an even number" is the set $\{0, 2, -2, 4, -4...\}$.

We schematize predicates by *predicate letters*. For example, we might say that Fx represents the statement that "x is an even number".

Structures In truth-functional logic, the truth value of a schema was relative to truth-assignments to its sentence letters. Structures play the same role for monadic quantification theory as truth-assignments do for truth-functional logic.

A structure A consists of:

- A set called the *universe of* A, written U^A. This set is the range for our variables of quantification.
  - For example, if I asserted that $(\forall x)(x \text{ is even} \vee x \text{ is odd})$, you'd probably assume that I was talking about all integers, not all things in general. In this case, the universe would (implicitly) be the set of integers. In ordinary speech, this implicit universe is contextually determined. When specifying a structure to interpret some monadic quantificational schemata, we explicitly identify the universe of discourse.
- The extensions of some monadic predicate letters (F^A, G^A, emphetc.), all of which are (arbitrary) subsets of the universe.
  - Let F^A be the set of even numbers, and G^A be the set of odd numbers. Then, within the universe of integers, the statement $(\forall x)(Fx \vee Gx)$ is a true statement, asserting that $(\forall x)(x \text{ is even} \vee x \text{ is odd})$.

If a sentence is true relative to some structure, we write $A \models S$ and say that "A satisfies S" (as we did for TF-logic), "A is a model of S", or "S is true in A".

Bound vs. Free Variables A variable x is said to be *bound* if it is within the scope of some quantifier. x is said to be *free* if it is not bound. The truth of sentences including free variables cannot be evaluated without an *assignment* of a value to those free variables. For example, the truth value of the sentence "x is an even number" cannot be determined without some assignment of a value (say, 2 or 3) to x. However, the sentence $(\forall x)(x \text{ is an integer} \supset x \text{ is even})$ can be evaluated (and is, of course, false).

3 Views of Structures We have three equivalent ways of looking at structures, which are

1. **The Canonical View**, which involves specifying the universe of discourse and extensions of predicates as sets.
2. **The Types View**, which involves drawing a table with sections for each "type".
3. **The Venn View**, which involves drawing a venn diagram, wherein the circles represent extensions of predicates.

Realizing Types We say that a structure realizes a type T_i iff there is some element of the structure in the quadrant T_i in the types view of our structure.

For example, with predicate letters F, G, the types would then be

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

and then we would say that a structure realizes type T_i iff it makes the sentence $(\exists x)(T_i(x))$ true.

The Small Model Theorem The Small Model Theorem states:

Theorem 3. *Let S be a pure monadic schema over n predicate letters. If S is satisfiable, then there is a structure A with $|U^A| \leq 2^n$ with $A \models S$.*

Corollaries to the Small Model Theorem include the algorithmic decidability of the satisfiability problem for schemata of MQT, and the fact that there are only finitely many schemata up to equivalence whose predicate letters are drawn from a fixed finite set.

Problems

For the later problems, it will be immensely useful to you to draw tables that look like this. Try to interpret the schema as statements about which quadrants of the table can/must have elements in them for the schema to be satisfied or falsified (this is the "types view").

	Fx	$\neg Fx$
Gx		
$\neg Gx$		

1. Are the following sentences equivalent? If not, does one imply the other?

$$S : (\forall x)(Px)$$

$$T : \neg(\exists x)(\neg Px)$$

2. Are the following sentences equivalent? If not, does one imply the other?

$$S : (\exists x)(Px)$$

$$T : \neg(\forall x)(\neg Px)$$

3. Are the following sentences equivalent? If not, does one imply the other?

$$S : (\forall x)(Px) \wedge (\forall x)(Qx)$$

$$T : (\forall x)(Px \wedge Qx)$$

4. Are the following sentences equivalent? If not, does one imply the other?

$$S : (\exists x)(Px) \wedge (\exists x)(Qx)$$

$$T : (\exists x)(Px \wedge Qx)$$

5. Are the following sentences equivalent? If not, does one imply the other?

$$S : (\forall x)(Px) \vee (\forall x)(Qx)$$

$$T : (\forall x)(Px \vee Qx)$$

6. Let

$$S : (\exists x)(Fx \wedge Gx) \wedge (\exists x)(\neg Fx \wedge Gx) \wedge (\exists x)(Fx \wedge \neg Gx) \wedge (\exists x)(\neg Fx \wedge \neg Gx)$$

$$T : (\forall x)(Fx \equiv Gx)$$

   (a) How many structures with universe $U = \{1, 2, 3\}$ are counterexamples to the claim that S implies T?

   (b) How many structures with universe $U = \{1, 2, 3, 4, 5\}$ are counterexamples to the claim that S implies T?

7. Let

$$S : (\forall x)(Fx \oplus Gx)$$

$$T : (\forall x)(Fx \equiv Gx)$$

How many structures are there with universe $U = \{1, 2, 3, 4, 5\}$ which witness the inequivalence of S and T?

8. Let

$$S : (\exists x)(Fx \wedge Gx)$$

$$T : (\forall x)(Fx \vee Gx)$$

How many structures are there with universe $U = \{1, 2, 3, 4, 5\}$ which witness the inequivalence of S and T?

9. Let

$$S : (\forall x)(Fx \oplus Gx)$$

$$T : (\forall x)(Fx) \oplus (\forall x)(Gx)$$

Given universe $U = \{1, 2, 3, 4, 5\}$, how many counterexamples are there to the claim that S implies T?

10. Let

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

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

Given universe $U = \{1, 2, 3, 4, 5\}$, how many counterexamples are there to the claim that S implies T?

Solutions

1. These are equivalent. The former asserts that everything has property P. The latter asserts that nothing lacks property P. These are equivalent statements.

2. These are equivalent. The former asserts that there is something with property P. The latter asserts that it is not the case that everything lacks P. These are equivalent statements.

3. These are equivalent. The former asserts that everything is P, as well as that everything is Q. The latter asserts that everything is both P and Q.

4. These are not equivalent, but T implies S. S asserts that there is something that is P, and there is something that is Q. T asserts that there is something that is both P and Q. Clearly if T is true, S must be as well. Hence T implies S. The inequivalence of S and T is witnessed by the structure A with $U^A = \{1,2\}$, $P^A = \{1\}$, $Q^A = \{2\}$. Then $A \models S$ but $A \not\models T$.

5. These are not equivalent, but S implies T. S asserts that everything is P or everything is Q. T asserts that everything is at least one of P or Q. Clearly if S is true then so is T, hence S implies T. The inequivalence of S and T is witnessed by the structure A with $U^A = \{1,2\}$, $P^A = \{1\}$, $Q^A = \{2\}$. Then $A \models T$ but $A \not\models S$.

6. (a) 0. To satisfy S, there must be at least one element which is each of $(Fx \wedge Gx)$, $(Fx \wedge \neg Gx)$, $(\neg Fx \wedge Gx)$, and $(\neg Fx \wedge \neg Gx)$. We only have three elements though, so at least one of those categories won't have an element in it. Hence S can't be satisfied in this universe. Hence there are no structures A such that $A \models S$ and $A \not\models T$.

   (b) $\binom{5}{2} \cdot 4! = 240$. To satisfy S, there must be at least one element which is each of $(Fx \wedge Gx)$, $(Fx \wedge \neg Gx)$, $(\neg Fx \wedge Gx)$, and $(\neg Fx \wedge \neg Gx)$. Think of these categories as "boxes" into which we are placing elements of our universe (draw a table to help yourself think through problems like this!). There are $\binom{5}{2} \cdot 4!$ ways to satisfy S. The $\binom{5}{2}$ term results from choosing two items from U which will go into the same "box". Henceforth we think of these two elements as now being a "package deal". The 4! term is the number of ways we can order our four things (the three single elements, plus our "package deal") into the four "boxes". Note that any structure that satisfies S cannot satisfy T. Hence all of the structures satisfying S are counterexamples to the claim that S implies T, and we have our answer.

7. 2^6. Note that if a structure satisfies S, it cannot satisfy T, and similarly if one satisfies T, it cannot satisfy S. Hence it suffices to count the number of ways S, T can each be satisfied and add those results together.

   To satisfy S, every element of U must be either $(Fx \wedge \neg Gx)$ or $(\neg Fx \wedge Gx)$. This corresponds to two choices for each of our 5 elements, meaning there are 2^5 ways to satisfy S. Note that none of these structures satisfy T.

   Similar reasoning suffices to show that there are 2^5 ways to satisfy T. None of these structures satisfy S.

   Our answer is then $2^5 + 2^5 = 2^6$.

8. $4^5 - 2 \cdot 3^5 + 2^6 = 602$.

   Let's start by counting the number of structures which are counterexamples to the claim that T implies S, since that direction is easier. For T to be true, every element must satisfy one of the types $(Fx \wedge Gx)$, $(Fx \wedge \neg Gx)$, or $(\neg Fx \wedge Gx)$. For S to be false, no element can satisfy the type $(Fx \wedge Gx)$. Hence, for T to be satisfied and S to be falsified, every element must satisfy either $(Fx \wedge \neg Gx)$ or $(\neg Fx \wedge Gx)$. Since we have two choices per element, there are 2^5 total ways to satisfy T and falsify S.

How many ways are there to satisfy S and falsify T? Let's begin by counting the ones that satisfy S. These are the structures which have at least one element of type $(Fx \wedge Gx)$. There are 4^5 structures in total, 3^5 of which have no element of type $(Fx \wedge Gx)$. Hence, there are $4^5 - 3^5$ structures satisfying S. From this number, we need to subtract the number that also satisfy T. 3^5 structures satisfy T (the structures in which every element is of one of the types $(Fx \wedge Gx)$, $(Fx \wedge \neg Gx)$, or $(\neg Fx \wedge Gx)$), so we subtract that from the number which satisfy S to get $4^5 - 2 \cdot 3^5$. Notice, however, that we *took away all structures whose elements were exclusively of type* $(Fx \wedge \neg Gx)$ *or* $(\neg Fx \wedge Gx)$ *twice - once when we were counting the number that satisfied* S*, and another time when counting the number that satisfed* T. Hence we "double subtracted" 2^5 possibilities, and we must add this back. This gives the result that there are $4^5 - 2 \cdot 3^5 + 2^5$ structures satisfying S which do not satisfy T.

Adding these two results together, we get $2^5 + (4^5 - 2 \cdot 3^5 + 2^5) = 4^5 - 2 \cdot 3^5 + 2^6 = 602$.

9. $2^5 - 2$. Note that S is satisfied by 2^5 structures (each element can be in either the box $(Fx \wedge \neg Gx)$ or the box $(Gx \wedge \neg Fx)$). Of these, only two satisfy T (the one in which all elements are in $(Gx \wedge \neg Fx)$, and the one in which all elements are in $(Fx \wedge \neg Gx)$). Hence, there are $2^5 - 2$ structures which satisfy S and falsify T.

10. 0. There are 3^5 structures satisfying S. Of these, none falsify T.

3 Polyadic Quantification Theory

3.1 Introduction to PQT

It's now time to turn to the final logic we will study, Polyadic Quantification Theory(PQT).[8] Unlike MQT, PQT allows predicates of arbitrary arity, not just monadic predicates. We will see that this change dramatically affects the complexity of the decision problems for satisfiability and validity, as well as the expressive power of schemata. aIndeed, polyadic quantification theory allows for the faithful schematization of vast tracts of scientific discourse.

For an example, 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*, *i.e.*, predicates which are true of *ordered pairs* of objects, not just individual objects, as in the case of monadic predicates. The *extension* of a dyadic predicate is the set of ordered pairs of which it is true.

- $\boxed{1}$ 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. In general, the extension of a predicate of arity n is a collection of n-tuples.

[8] Also called *First Order Logic.*

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 (or other).

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

- There is someone who 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.

Give an intuitive argument to show that S_2 implies S_1, and that S_4 implies S_3.

In order to show that no other implications hold, we introduce schemata to represent the statements $S_1, \ldots, S_4$, and the notion of a structure suitable for their interpretation.

Schemata and Structures

Just as we did in the case of truth-functional logic and monadic quantification theory, we introduce a formal language for schematizing statements involving polyadic predicates. The only change to the vocabulary of monadic quantification theory is the addition of polyadic predicate letters. For example, we introduce dyadic predicate letters such as $L\,\boxed{1}\,\boxed{2}$ to schematize dyadic predicates such as $\boxed{1}$ loves $\boxed{2}$ or $\boxed{1} \leq \boxed{2}$. We may then use this dyadic predicate letter to schematize the four statements above as follows.

- $S_1: \quad (\forall x)(\exists y)(Lxy)$
- $S_2: \quad (\exists y)(\forall x)(Lxy)$
- $S_3: \quad (\forall y)(\exists x)(Lxy)$
- $S_4: \quad (\exists x)(\forall y)(Lxy)$

We now extend the notion of a structure to the case of polyadic quantification theory. Again, a structure A is determined by the specification of a nonempty set U^A, the universe of discourse of A, over which the variables of quantification range, and specifications of extensions of the polyadic predicates which appear in schemata that the structure will be used to interpret. Thus, in the case to hand, the specification of A will require assigning to the dyadic predicate letter L, a set L^A of ordered pairs of elements from U^A as its extension. That is, $L^A \subseteq U^A \times U^A$, where $U^A \times U^A$ is the *Cartesian product* of U^A with itself – the set $\{\langle a, b\rangle \mid a, b \in U^A\}$.

The interpretation of the logical vocabulary, that is, truth-functional connectives and quantifiers, is the same in both monadic and polyadic quantification theory. Thus, we may introduce without further ado the notion of a schema S of polyadic quantification being true in (or satisfied by) a structure A that assigns extensions to all the polyadic predicate letters appearing in S (written $A \models S$). If S contains free variables, we must of course supplement the structure S with assignments of elements from U^A to those free variables. For example, we write

$$A \models S[(x|a)(y|b)]$$

for "the structure A satisfies the schema S with respect to the assignments of a to x and b to y." This notation is used with the understanding that no variables other than x and y occur free in S and that $a, b \in U^A$. With this definition of satisfaction, our definitions of satisfiability, validity, implication, and equivalence for closed monadic quantificational schemata generalize immediately to PQT. For ease of reference, we restate them as follows.

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

Thus, in order to show that a schema S fails to imply a schema T, it suffices to exhibit a *counterexample* to the implication, that is, a structure A such that $A \models S$ and $A \not\models T$. We proceed to illustrate this technique by show that among the four schemata $S_1, \dots, S_4$ discussed above, if $i \neq j$, then S_i does not imply S_j except in case $i = 2$ and $j = 1$, or $i = 3$ and $j = 4$.

We begin by specifying three structures A, B, C which act as a counterexamples to various of these implications. First we let $U^A = U^B = U^C = \{1, 2\}$. We specify the extension of L in each structure as follows.

- $L^A = \{\langle 1, 1\rangle, \langle 2, 2\rangle\}$.
- $L^B = \{\langle 2, 2\rangle, \langle 1, 2\rangle\}$.
- $L^C = \{\langle 2, 2\rangle, \langle 2, 1\rangle\}$.

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 implications now follows. For example, S_1 does not imply S_4. To see this, suppose, *ad reductio*, that S_1 implies S_4. Then since S_2 implies S_1 and S_4 implies S_3, it follows, by the transitivity of implication, that S_2 implies S_3. But we have already seen that S_2 does not imply S_3 (B was the counterexample), a contradiction.

Verify that each of the statements above is true. For example, if we claimed that $A \models S_i$ for some i, explain in your own words why, in fact, A actually models S_i.

Show that the remaining non-trivial implications also fail. To do this, use proof-by-contradiction as we did to show that S_1 does not imply S_4.

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

Quantificational ambiguity

We briefly explore ambiguities that can arise in natural language via the interaction of quantifiers. Consider the statement,

"everybody loves a lover." (2)

Statement (2) involves such an ambiguity. We can bring this out by offering two schematizations, each of which corresponds to a natural reading of this statement. We may schematize "x is a lover," again using the dyadic predicate letter L, as $(\exists y)Lxy$.Now, consider the following two schemata.

$$(\forall z)(\exists x)((\exists y)Lxy \wedge Lzx)$$

$$(\forall x)((\exists y)Lxy \supset (\forall z)Lzx)$$

The first schema corresponds to the reading "everybody loves someone who is a lover," while the second corresponds to the reading "if someone is a lover, then everybody loves her."

We claim 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.

Give an intuitive argument to verify this claim.

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.

3.2 Binary Relations, Functions, and Graphs

We now discuss several important properties of binary relations. Again, we deploy the dyadic predicate letter L to schematize these properties.

Definition 14. L^A *is* reflexive *if and only if*

$$A \models (\forall x)Lxx.$$

Definition 15. L^A *is* irreflexive *if and only if*

$$A \models (\forall x)\neg Lxx.$$

Definition 16. L^A *is* symmetric *if and only if*

$$A \models (\forall x)(\forall y)(Lxy \supset Lyx).$$

Definition 17. L^A *is* asymmetric *if and only if*

$$A \models (\forall x)(\forall y)(Lxy \supset \neg Lyx).$$

Definition 18. L^A *is* transitive *if and only if*

$$A \models (\forall x)(\forall y)(\forall z)(Lxy \supset (Lyz \supset Lxz)).$$

Definition 19. *A is a* simple graph *if and only if L^A is irreflexive and symmetric.*

Identity

A new logical dyadic predicate, identity, will allows us to "put the quant into quantification." The identity predicate "=" 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.[9]

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.

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

The crux of our proof of the Small Model Theorem for MQT was that realizing the same types corresponded to satisfying the same monadic sentences. Of course, realizing the same types does not mean that two structures satisfy the same sentences of PQT: the identity relation allows us to distinguish structures by *counting* how many elements realize a given type. *A priori*, this does not mean that decidability (which was a corollary of our SMT) fails for PQT - only that our particular proof of the SMT for MQT would fail for PQT. We will see more about the decidability of PQT in the future.

Let's use $|X|$ to denote the number of members of a set X (often called the *cardinality* of X). In order to clarify the import of these numerical quantifiers, we introduce 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. Our next goal is to use these quantifiers to define regular simple graphs.

[9]We call identity a *logical* predicate (as opposed to your garden-variety, run-of-the-mill predicates) because of this uniform interpretation across all structures. It means the same thing regardless of which structure you're in, so it is of logical character.

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). Unless otherwise stated, we will restrict our attention for the rest of the course to structures that are graphs; this restriction doesn't lose us much (all results we will derive are easily generalizable to structures with arbitrary relations) and it adds much tangibility. A graph A is *simple* if and only if L^A is both irreflexive and symmetric (*i.e.*, the edges are undirected, and tehre are no "self loops" at any node). We introduced the abbreviation SG for the conjunction of the schemata expressing irreflexivity and symmetry, which we abbreviated as Irr and Sym, respectively. For a structure A, $A \models \mathsf{SG}$ iff A is a simple graph.

Definition 20. *The* neighborhood of a in A *is* $\mathsf{nbh}(a, A) := \{b \in U^A \mid \langle a, b\rangle \in L^A\}$ *(the set of all neighbours of a).*

Definition 21. *The* degree of a in A *is* $\mathsf{deg}(a, A) := |\{b \in U^A \mid \langle a, b\rangle \in L^A\}|$ *(the number of neighbours of a, or equivalently the number of edges incident to a).*

Definition 22. *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*

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

What do the collections of 1-regular and 2-regular simple graphs look like? Every 1-regular graph consists of a set of independent edges, and every *finite* 2-regular graph consists of a collection of independent simple cycles, that is, graphs that may be drawn in the plane as a finite collection of disjoint polygons.[10]

Counting graphs

Just as we did for truth-functional logic and MQT, we can also count the structures that satisfy schemata of PQT. In particular, we will count graphs with a fixed universe of discourse. To that end, we have the following definition.

Definition 23. *We denote the set of simple graphs with universe of discourse $[n]$ that satisfy a schema S by $\mathsf{mod}(S, n)$, that is,*

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

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 $\{1, \ldots, n\}$.

Let's count the number of graphs A with $U^A = \{1, 2, 3, 4\}$ (by the previous comment, this is of course equal to $|\mathsf{mod}((\forall x)x = x, 4)|$). Any such graph is determined by choosing which of the sixteen possible edges from i to j to draw, where $1 \le i \le 4$ and $1 \le j \le 4$; 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. Analogous reasoning leads to the conclusion that there are 2^{n^2} graphs with universe of discourse $\{1, \ldots, n\}$ (because there are n^2 pairs of nodes, and hence 2^{n^2} possible edge-sets). 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 $U^A = \{1, \ldots, n\}$.

How many 1-regular simple graphs are there with universe of discourse $\{1, \ldots, n\}$?

[10] Infinite two regular graphs may also include copies of the doubly-infinite simple chain (think of the integers, where there is an edge $\langle a, b\rangle$ iff $a = b \pm 1$). Polygons and bi-infinite chains exhaust the possible connected components of 2-regular graphs.

Functions, Tournaments, and Orderings

You may already have encountered functions, such as the mapping f that sends a real number x to its square x^2. You were probably given the following definition of a function:

Definition 24. *A function is any mapping f from a* domain A *to a* codomain B *(written $f : A \to B$) such that no $a \in A$ maps to more than one element in b.*[11] *If $b = f(a)$, we say that* b is the image of a under f, *alternatively,* b is the f image of a. *If $X \subseteq A$, we define $f[X] =: \{fa \mid a \in X\}$; generalizing the previous terminology, we say that $f[X]$ is the f image of X.*

Herein (unless otherwise noted) we will always consider functions whose domain and codomain are the same; this will allow us to consider functions as special types of binary relations on some universe of discourse.

You have probably seen the function $f(x) = x^2$ represented in cartesian coordinates via a graph, that is, the set of all ordered pairs of real numbers $\langle x, x^2 \rangle$ for $x \in \mathbb{R}$. This suggests that a function can be thought of as a specific type of binary relation. In the example $f(x) = x^2$, this means that we consider the Cartesian graph as a structure, namely the 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.

Of course, SV serves as an alternative definition of a function. The conjunction of SV and Tot 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.

There are special types of functions which will be interesting to us: namely, *injections*, *surjections*, and *bijections*.

Definition 25. *An* injection *(also called a 1-1 function) is a function which maps distinct elements of the domain to distinct elements of the codomain. In other words, no two different elements of the domain map to the same element in the codomain. We schematize this as*

$$\mathsf{Inj} : (\forall x)(\forall y)(\forall z)((Lxz \wedge Lyz) \supset x = y)$$

You may be familiar with the idea in terms of the "horizontal line rule", which says that if a horizontal line crosses the (Cartesian) graph of a function in more than one point, that function is not injective.

Use the horizontal line rule to show that $f(x) := x^2$ is not injective. Give a general relation between distinct elements a, b which witness that f is not injective.

Definition 26. *A* surjection *(also called an onto function) is a function with the property that every element of the codomain is the image of some element of the domain. Schematically:*

$$\mathsf{Sur} : (\forall x)(\exists y)Lyx$$

Show that $f(x) := x^2$ is not surjective when our universe of discourse is $\mathbb{R}$, the set of real numbers. Fixing the domain as $\mathbb{R}$, give a codomain which would make the function surjective.

Definition 27. *A* bijection *is a function which is both injective and surjective. Schematically*

$$\mathsf{Bij} : \mathsf{Inj} \wedge \mathsf{Sur}$$

[11] Here, we only consider *unary* functions, that is, functions of one argument. The definition could be amended for general n-ary functions by letting $\overline{a} \in A^n$ denote an n-tuple of elements in the domain.

Show that $f(x) := x^3$ is a bijection.

We have now seen examples of functions which are either bijective (the cubing function) and functions which are neither injective nor surjective (the squaring function). Is it possible to have a function which is injective but not surjective, or surjective but not injective?

For any structure A with finite universe of discourse, $A \models \mathsf{Fun} \wedge \mathsf{Inj}$ iff $A \models \mathsf{Fun} \wedge \mathsf{Sur}$, that is, any injective function on a finite structure is surjective, and vice versa.

Prove this.

The same does not hold for structures with infinite universes - in fact, the prominent nineteenth-century mathematician Richard Dedekind used this distinction as his definition of infinitude. For example, consider the structure A where $U^A = \mathbb{N}$ and $L^A = \{\langle n, n+1\rangle \mid n \in \mathbb{N}\}$ and observe that $A \models \mathsf{Fun} \wedge \mathsf{Inj} \wedge \neg\mathsf{Sur}$. 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 codomain X which is injective but not surjective.

We now touch briefly on the topic of multivariate functions; we restrict 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: $(\forall v)(\forall w)(\forall x)(\forall y)(\forall z)((Rvwz \wedge Rxyz) \supset (v = x \wedge w = y))$

If A is a finite structure and $A \models \mathsf{Bfun} \wedge \mathsf{Binj}$, then $|U^A| = 1$.

Prove this.

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 follows from the fundamental theorem of arithmetic). 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.

Tournaments and Orderings

Definition 28. *We say that a directed graph is* asymmetric *if no pair of its nodes have "edges in both directions", that is,*

$$\mathsf{Asy} : (\forall x)(\forall y)(Lxy \supset \neg Lyx)$$

Definition 29. *We say that a directed graph is* comparable *if every pair of distinct nodes has at least one edge between them, that is,*

$$\mathsf{Comp} : (\forall x)(\forall y)(x \neq y \supset (Lxy \vee Lyx))$$

Definition 30. *We say a directed graph is a* tournament *iff it is both asymmetric and comparable, that is,*

$$\mathsf{Tour} : \mathsf{Asy} \wedge \mathsf{Comp}$$

The intuitive justification for the name "tournament" is that round-robin tournaments involve each team playing every other team once, and that each game between teams induces an "edge" which points from the victor to the loser (games never end in a tie).

Let's pick out a particularly important class of tournaments, those without cycles. We call these *transitive* tournaments.

Definition 31. *We say that a tournament is* transitive *iff the edge relation is transitive, that is,*

$$\mathsf{Trans} : (\forall x)(\forall y)(\forall z)(Lxy \supset (Lyz \supset Lxz))$$

Definition 32. *A* strict linear order *is a transitive tournament, that is,*

$$\mathsf{SLO} : \mathsf{Trans} \wedge \mathsf{Tour}$$

Counting Graphs

As before, we'll count the number of finite structures with universe of discourse $\{1, \ldots, n\}$ that satisfy various conditions. We already know that there are $2^{(n^2)}$ graphs and $2^{\binom{n}{2}}$ simple graphs with universe of discourse $\{1, \ldots, n\}$. It is simple to show that

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

Prove each of the above assertions by counting them yourself.

Since you're probably a pro at this sort of thing by now, let's try counting something that's a bit more difficult - the number of two-regular simple graphs of a fixed size. Recall that a simple graph is 2-regular 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. Let's calculate $|\mathsf{mod}(S, 6)|$, that is, the number of 2-regular simple graphs of size 6.

Example 10. *Let S be the conjunction of* $\mathsf{2reg}$ *and* SG. *What is $|\mathsf{mod}(S, 6)|$, that is, how many 2-regular simple graphs of size 6 are there?*

As discussed earlier, if A is finite (as it is in the question above) and $A \models S$, then A is a disjoint union of cycles. It follows that if $A \in \mathsf{mod}(S, 6)$ then A is either a disjoint union of two triangles, or a single hexagon (since a triangle is the minimal 2-regular cycle, these are the only possibilities). 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. How can we do that?

Let's consider the two triangles case first. 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. It follows that 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 can use our prior calculation that there are 6! strict linear orders of [n] to calculate the number of distinct labelings of $\mathbb{H}$. The rotational symmetry of the hexagon means that if we "wrap the ordering around" a hexagon from some fixed starting point, each of the 6 possible rotations of the labelled hexagon (each corresponding to a different linear order) are equivalent. Moreover it is clear that the reverse of any order gives the same labeling as the order itself. Thus, the total number of labelings of $\mathbb{H}$ is $6!/(6 \cdot 2) = 60$. It follows that $|\mathsf{mod}(S, 6)| = 10 + 60 = 70$.

3.3 The Spectrum of a Schema

Let's introduce a monadic predicate letter F to "color" the nodes of our graphs. A new condition, *distinguished end*, says that the "coloring" of the nodes in our graph alternates between adjacent elements.

$$\mathsf{DE} : (\forall x)(\forall y)(Lxy \supset (Fx \oplus Fy))$$

Consider the schema T which is the conjunction of SG, $\mathsf{2reg}$, and DE. 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. The notion of the *spectrum* of a schema describes this property of having (at least one) model of a given size. Writing $\mathbb{Z}^+$ for the set of positive integers, we have the definition:

Definition 33. *Let S be a schema. Then*

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

If a schema S has a model whose universe is of size n, we say that S admits *n. Then $\mathsf{Spec}(S)$ is exactly the set of positive integers n such that S admits n.*

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

Example 11. *Lest we forget how to count, let's exploit what we learned from Example 10 and calculate $|\mathsf{mod}(T, 6)|$. The only shape allowed in this case is the hexagon, and each hexagon admits two possible colorings that satisfy DE (one where the initial element in our ordering is coloured, and one where it is not). Hence, it follows from our earlier calculation that $|\mathsf{mod}(T, 6)| = 2 \cdot 6!/(6 \cdot 2) = 120$.*

Finite Sets and Co-finite Sets are Spectra

Let F be a finite set of positive integers. Here is a basic question:

"Is there a schema S such that $\mathsf{Spec}(S) = F$?"

In order to begin answering this, we'll start with singletons, and show 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, which says that there are at least n but not at least $n+1$ elements in the universe of any satisfying structure.

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

It follows at once that every finite set of positive integers is the spectrum of some schema, for if $F = \{n_1, \dots, n_k\}$, then

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

Moreover,

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

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*).

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. Let's consider the following example.

Recall the schema $\mathsf{SG} \wedge \mathsf{1reg}$ which defines the collection of 1-regular simple graphs. We've 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, $\mathsf{Spec}(\neg(\mathsf{SG} \wedge \mathsf{1reg})) = \mathbb{Z}^+$.

Why is $\mathsf{Spec}(\neg(\mathsf{SG} \wedge \mathsf{1reg})) = \mathbb{Z}^+$?

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.

You may also ask: "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, however, 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, the paper "Fifty Years of the Spectrum Problem" is a great place to start.[12]

Further Examples of Infinite, Co-infinite Spectra

One can easily 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. This suffices to make any satisfying structure a collection of disjoint pairs plus one isolated element, and hence of odd size.

Write out the above described schema formally.

Time for a more substantial example: a schema S with $\mathsf{Spec}(S) = \{k^2 \mid k \in \mathbb{Z}^+\}$. The schema involves 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 (in particular, $|U^A| = |F^A|^2$).

[12]A. Durand, N. D. Jones, J. A. Makowsky, and M. More, Fifty Years of the Spectrum Problem: Survey and New Results, *The Bulletin of Symbolic Logic*, Volume 18, Number 4, Dec. 2012, 505-553.

3.4 Equivalence Relations

Time to look at another important class of graphs (namely, equivalence relations) and how they can be put to use in generating schemata with a wide range of spectra. Recall that 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 have S imply Eq 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 $\{b \in U^A \mid \langle a, b\rangle \in L^A\}$. It follows that 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 our universe must have size $3i + 2$ for some i (i is the number of size-three equivalence classes). We could then take S to be the conjunction of Eq and T, and would be done. The following schema T does the job.

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

Roughly, T says that there are two distinguished elements x_1, x_2 which are related only to each other, and that all other elements y_1 have exactly two other elements y_2, y_3 (both of which are neither of the x's) which are all connected in a triangle.

Go through the schema T and write out exactly what each part is saying. Make sure you understand why it specifies that there are $3i + 2$ elements in the universe (for some i).

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

Using the counting quantifiers (and some new shorthand of your own devising to say that the equivalence class of x has size j), give a schema which provides the above-mentioned generalization.

3.5 Review

Concept Review

Binary Relations: Given a set S, a binary relation R between a domain A and a codomain B is a set of pairs (a, b) where $a \in A$ and $b \in B$. We will normally consider relations whose domain equals their codomain. In this case, a relation R over a set S is a subset of $S^2 = S \times S = \{(s, s')|s, s' \in S\}$ (this is the "Cartesian product" of S with itself, the set of all *ordered pairs* of members of S). Notice that we have identified the interpretation of a dyadic predicate letter in a structure with a relation in this sense; this continues our practice of treating our schematic languages *extensionally*. For example, let L be a dyadic predicate letter. The structure A with $U^A = \{0, 1, \ldots\}$ and $L^A = \{(i, j) \mid i, j \in \mathbb{N} \text{ and } i < j\}$ interprets L as the strict linear ordering relation on the set of natural numbers $\mathbb{N}$.

Arbitrary Relations: Relations need not be just binary. For example, our normal interpretation of the + symbol (along with equality) specifies a relation - we might think of the ternary relation $+abc$ as expressing that $a+b = c$. The extension of this relation is then a set of ordered triples $\{(a, b, c)|a+b = c\}$.

Graphs: A (directed) *graph* is a structure $G = (V, E)$ suitable for interpreting a dyadic predicate symbol L. V is a set of "vertices" or "nodes", and E is a binary relation (the edge relation) on V (that is, $E \subseteq V \times V$). We will often write U^G for V and L^G for E when we use G to interpret schemata involving only the dyadic predicate symbol L.

Neighborhood: The neighborhood of a node n of a graph $G = (V, E)$ is the set of all nodes which are adjacent to n in G (that is, $\{n' \in V|(n, n') \in E\}$).

Degree: The degree of a node n is the size of n's neighborhood.

Reflexive: A relation R is said to be *reflexive* iff $(\forall x)Rxx$.

If we draw an arrow diagram of such a relation, each vertex would have a "self loop." If we visualize such a relation as a bit matrix, the main diagonal is all "1s."

Irreflexive: A relation R is said to be *irreflexive* iff $(\forall x)\neg Rxx$.

If we draw an arrow diagram of such a relation, this specifies that there are no "self loops."

Symmetric: A relation R is said to be *symmetric* iff $(\forall x)(\forall y)Rxy \supset Ryx$.

This means that there are no one-way relations between elements. Specifically, either $(Rxy \wedge Ryx) \oplus (\neg Rxy \wedge \neg Ryx)$. When drawing an arrow diagram of a symmetric relation, we often omit the arrowheads on our edges to indicate that the relation "goes both ways."

Antisymmetric: A relation R is said to be *antisymmetric* iff $(\forall x)(\forall y)Rxy \supset \neg Ryx$.

This means that there are no two-way relationships between elements.

Transitive: A relation R is said to be *transitive* iff $(\forall x)(\forall y)(\forall z)((Rxy \wedge Ryz) \supset Rxz)$ or, equivalently (why?) $(\forall x)(\forall y)(\forall z)(Rxy \supset (Ryz \supset Rxy))$.

Comparable: A relation is R said to be *comparable* iff $(\forall x)(\forall y)(x \neq y \supset (Rxy \vee Ryx))$.

In words: if x and y are unequal, then they have to be related in some way (so every pair of nodes is related somehow).

Simple Graph: A relation is said to be a *Simple Graph* iff the relation is both irreflexive and symmetric. We abbreviate this SG.

Tournament: A relation is said to be a *Tournament* iff the relation is both antisymmetric and comparable. Think of the nodes in the graph as being teams, and the edge relation as being "won against" in a round robin - each team plays each other (so they all have at least one edge between) and only one team wins each game (so if a beats b, then b does not beat a). We abbreviate this Tour.

Strict Linear Order: A relation is said to define a *Strict Linear Order* iff the relation is antisymmetric, comparable, and transitive (thus, SLOs are a subclass of Tournaments). The extra criterion of transitivity requires that if a "beats" b, then a also "beats" every team which b beats. In this way, the "1st place team" beats everyone, 2nd place beats everyone except first, *etc.*, and so there is a strict ordering on the elements of our graph.

Equality: Equality is a special type of relation in that its interpretation is uniform across structures. For every structure A, equality holds just of the pairs $\{(a,a)|a \in U^A\}$.

Counting Quantifiers: In MQT, the *Small Model Property* meant we had no way to distinguish between, for example, the two structures

$$A : U^A = \{1,2\}, P^A = \{1,2\}$$

and

$$B : U^B = \{1\}, P^B = \{1\}$$

because A, B realised the same types, *even though* a different number of elements of A are in P than elements of B are. With PQT, we can of course distinguish between these two structures by *using equality to count.*

We defined sentences $\Delta_n := (\exists x_1)...(\exists x_n) \bigwedge_{1 \le i < j \le n} x_i \neq x_j$ which indicates that elements $x_1...x_n$ are all pairwise distinct. Using this, we defined

$$(\exists^{k\le} x)P(x)$$

which says there are at least k elements such that P

$$(\exists^{\le k} x)P(x)$$

which says that there are at most k elements such that P, and

$$(\exists^{=k})P(x)$$

which says that there are exactly k elements such that P.

k-Regular: We say that a graph $G = (V, E)$ is k-regular iff every node $n \in V$ has degree k. Finite 2-regular simple graphs are composed of a collection of disjoint cycles, and 1-regular simple graphs are composed of a collection of disjoint "isolated edges."

Counting Graphs: We defined

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

That is, $\mathsf{mod}(S, n)$ is the collection of all structures satisfying a sentence S which have universe $\{1, ..., n\}$.

Functions as Relations: Functions are a special subclass of relations. Specifically, they are relations which satisfy the two properties

$$\mathsf{Tot} : (\forall x)(\exists y)Rxy$$

and

$$\mathsf{SV} : (\forall xyz)((Rxy \land Rxz) \supset y = z).$$

The first says that the relation is *total* - for everything in the domain, there is something to which it maps. The second says that the relation is single valued - every element of the domain maps to at most one element. The conjunction $\mathsf{Tot} \land \mathsf{SV}$ implies that every element of the domain maps to exactly one element of the codomain. Strictly, Tot is not required for a relation to be a function (the class of relations axiomatized by only SV are called the *partial functions*). In this class, however, we restrict our attention to total functions, and will always mean "total function" whenever we say "function."

Injectivity: A function is injective (also sometimes called "one-to-one") iff the following holds

$$\mathsf{Inj} : (\forall xyz)((Rxz \wedge Ryz) \supset x = y)$$

which says that no two elements of the domain map to the same element of the codomain.

If there is an injection from S to S' then $|S| \leq |S'|$ (prove this for finite sets).

Surjectivity: A function is surjective (also sometimes called "onto") iff the following holds:

$$\mathsf{Sur} : (\forall x)(\exists y)Ryx$$

which says that every element of the codomain is the image of some element of the domain.

If there is a surjection from S to S', we know that $|S| \geq |S'|$ (prove this for finite sets).

Functions on a Single Set: We often consider functions with the same domain and codomain (say S). In this case, we say that the function is on S. We proved that, for finite sets S, a function on S is an injection iff it is a surjection. In contrast, we showed that there are injections on an infinite S which are not surjections, and similarly that there are surjections on an infinite S which are not injections. For example

$$f : \mathbb{N} \to \mathbb{N}$$

$$n \mapsto n + 1$$

is an injection but not a surjection (0 is not mapped to), and

$$h : \mathbb{N} \to \mathbb{N}$$

$$n \mapsto \lfloor n/2 \rfloor$$

is a surjection but not an injection (for all k, $2k$ and $2k + 1$ both map to k).

A set S is called *Dedekind infinite* if and only if there is an injection on S that is *not* a surjection.

Binary Functions: Just as the *graph* of a unary function f is the binary relation $\{(a, b) \mid f(a) = b\}$, the *graph* of a binary function g is the ternary relation $\{(a, b, c) \mid g(a, b) = c\}$. The following schema is satisfied by a structure A if and only if R^A is the graph of a total binary function on U^A.

$$\mathsf{Bfun} := (\forall x)(\forall y)(\exists z)(\forall w)(Rxyw \equiv w = z).$$

Binary Injection: Similarly, the following schema expresses that a binary function is injective.

$$\mathsf{Binj} := (\forall vwxyz)((Rvwz \wedge Rxyz) \supset (v = x \wedge w = y))$$

Find a schema that expresses surjectivity for binary functions.

Counting Relations: We have the following:

$$|\mathsf{mod}(\mathsf{Fun}, n)| = n^n,$$

that is, there are n^n functions on a set of size n. To see this, note that each for each of the n elements we must choose one of n possible images. In other words, to determine a function on $[n]$, we must make n n-ary choices.

$$|\mathsf{mod}(\mathsf{Fun} \wedge \mathsf{Inj}, n)| = n!$$

To see this, note that a function on a set S which is injective maps each of the n elements to a unique element. So there are n choices for the first element, $n-1$ for the second, ..., and 1 for the last.

$$|\mathsf{mod}(\mathsf{Asy}, n)| = 3^{\binom{n}{2}}$$

To see this, note that for each pair of elements $\{a, b\}$ there are three mutually exclusive possibilities - no edge, edge from a to b, or edge from b to a.

$$|\mathsf{mod}(\mathsf{Tour}, n)| = 2^{\binom{n}{2}}$$

To see this, note that for each pair of elements $\{a, b\}$ there must be exactly one of the directed edges from a to b or from b to a.

$$|\mathsf{mod}(\mathsf{SLO}, n)| = n!$$

To see this, note that there is a natural one-to-one correspondence between linear orderings of $[n]$ and injections on $[n]$ – map each $i \in [n]$ to the i-th member of the order; thus, there are the same number of strict linear orders of $[n]$ as there are injections on $[n]$.

$$|\mathsf{mod}(\mathsf{2reg}, 3)| = 1$$

To see this, note that a 2-regular graph of size 3 must be a "triangle", and there is only one way of making a triangle with three elements.

$$|\mathsf{mod}(\mathsf{2reg}, 4)| = 3$$

To see this, note that a 2-regular graph of size 4 must be a "square". Fix one element a of the square, and pick the single node from the remaining three to which it is not adjacent (three choices). This uniquely determines the square.

In general, there are $\frac{n!}{n \cdot 2} = \frac{(n-1)!}{2}$ n-cycle graphs.

$$|\mathsf{mod}(\mathsf{2reg}, 6)| = 70$$

Any 2-regular graph of size six must be composed of two "triangles" or one "hexagon." For the triangles, there are $\binom{6}{3}/2 = 10$ ways to split the six up into two groups of three (we divide by two because, say, picking elements $\{1, 2, 3\}$ for one triangle is the same as picking $\{4, 5, 6\}$ for the other). Each such division corresponds to exactly one graph, so there are 10 graphs which have two disjoint triangles. For the hexagon, there are (reasoning as above) $\frac{6!}{6 \cdot 2} = 60$ ways to label the hexagon, giving $10 + 60 = 70$ total possible graphs.

Spectrum: We defined the *spectrum* of a schema S to be the set of all integers n for which there is a model of S of size n. We determined that for each positive integer n, there is a schema S_n such that $\mathsf{Spec}(\mathsf{S}_n) = \{n\}$. The schema S_n specifies that there are exactly n elements in the universe.

By making use of the schemata S_n, we constructed, for each finite set F and for each cofinite set C, schemata S and T with $\mathsf{Spec}(S) = F$ and $\mathsf{Spec}(T) = C$. (Recall that a set of positive integers X is *cofinite* if and only if $\mathbb{Z}^+ - X = \overline{X}$ is finite)

We noted that, in general, $\mathsf{Spec}(\neg S) \neq \overline{\mathsf{Spec}(S)}$. For example, let $S := \mathsf{SG} \wedge \mathsf{1reg}$, so that $Spec(S) = \{2i | i \in \mathbb{Z}^+\}$. Then $\neg S$ is equivalent to $\neg\mathsf{SG} \vee \neg\mathsf{1reg}$. Therefore, $\mathsf{Spec}(\neg S) = \mathbb{Z}^+$.

Equivalence Relations: An equivalence relation is a relation which satisfies the schema Eq, which is the conjunction of the following schemata.

- Refl: $(\forall x)Lxx$
- Sym: $(\forall x)(\forall y)(Lxy \supset Lyx)$

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

If $A \models \mathsf{Eq}$, the set $\hat{a} = \{a' \in U^A | \langle a, a' \rangle \in L^A\}$ is the *equivalence class of* a; it is the set of all $a' \in U^A$ which are equivalent to a under the equivalence relation L^A.

The equivalence classes of an equivalence relation partition the universe, that is, every element of the universe is in some equivalence class (this follows by reflexivity) and distinct equivalence classes are disjoint (this follows by symmetry and transitivity).

For all $k \in \mathbb{Z}^+$ and $0 \leq l < k$, we can define the spectrum $\{(ki + l) \in \mathbb{Z}^+ | i \in \mathbb{N}\}$ (the set of all positive integers equivalent to l mod k) by specifying that there are exactly l elements in an equivalence class of size l, and that all other elements are in size-k equivalence classes.

Problems

1. We say that a graph $G \in \mathsf{mod}(S, n)$ is *size maximal* iff it has the most edges possible given that it satisfies S and has n nodes. Give an example of a size-maximal acyclic member of $\mathsf{mod}(\mathsf{SG}, 3)$.
2. How many size-maximal acyclic members of $\mathsf{mod}(\mathsf{SG}, 3)$ are there?
3. How many size-maximal acyclic members of $\mathsf{mod}(\mathsf{SG}, 4)$ are there?
4. Prove that if every node of a finite graph is of degree at least 2, then the graph contains a cycle.
5. Let S be the conjunction of
$$\mathsf{SLO}$$
$$(\forall x)(\forall y)((Lxy \wedge \neg(\exists z)(Lxz \wedge Lzy)) \supset Fx \oplus Fy)$$
$$(\forall x)(\neg(\exists y)Lyx \supset Fx)$$
$$(\forall x)(\neg(\exists y)Lxy \supset \neg Fx)$$
What is $\mathsf{Spec}(S)$?
6. Give a schema S such that $\mathsf{Spec}(S) = \{2i + 1 | i \in \mathbb{N}\}$, that is, the set of odd positive integers.
7. What is $\mathsf{Spec}(\mathsf{SLO})$?
8. Let S be the conjunction of
$$\mathsf{SLO}$$
$$(\forall x)(\exists y)Lxy$$
$$(\forall x)(\exists y)Lyx$$
$$(\forall x)(\forall y)(Lxy \supset (\exists z)(Lxz \wedge Lzy))$$
What is $\mathsf{Spec}(S)$?

Solutions

1. $U^A = \{1, 2, 3\}$, $R^A = \{(1, 2), (2, 1), (2, 3), (3, 2)\}$ is one such example.
2. $\binom{3}{2} = 3!/2 = 3$. Note that all size-maximal acyclic members of $mod(SG, 3)$ have two undirected edges (if they had a third, there would have to be a cycle). So to specify such a graph, we simply have to pick which two nodes are not connected. Equivalently, you might note that all such graphs look like a single chain (or, a linear order in which direction doesn't matter). There are 3! orders, and we divide by 2 because reading from right-to-left or left-to-right doesn't matter (we have a simple graph, so the edges are undirected).
3. $4!/2 + 4 = 16$. As above, there are $4!/2$ ways to order the elements into a single chain. However, unlike the case above, there are graphs which aren't just a single chain. Picking one element to be the "parent" and placing undirected edges from it to every other element (draw this!) gives a graph which is also size maximal acyclic, but not a chain. There are 4 ways to pick such a parent. So the total is $4!/2 + 4 = 16$.

   In general, acyclic size maximal members of $\mathsf{mod}(\mathsf{SG}, n)$ are *trees* of size n (a "tree" is a size-maximal acyclic graph, or equivalently a size-minimal connected graph).
4. Consider a maximal length path (an acyclic sequence of edges) in the graph. As the terminal node of that path has degree at least 2, there must be an edge coming from it which is not on our path. As our path was maximal in length, that edge must go back to some previously visited node along our path.
5. $\{2i | i \in \mathbb{Z}^+\}$.

   To answer this question, we interpret S piece by piece. The first conjunct is SLO, so we treat this like a linear order from now on (in particular, we can interpret L as meaning "less than").

   The second conjunct says that "for all x and y, if there isn't a z between them, then exactly one of x and y are in F". Interpreting Fx as meaning "x is colored red" and $\neg Fx$ as meaning "x is colored blue", we interpreted this sentence as saying that the coloring of nodes in our order switches from red to blue every element.

   The third conjunct says that if there is no element less than x (so x is the minimal element), then x is red.

   The final conjunct says that if there is no element greater than x (so x is the maximal element), then x is blue.

   Putting this all together, we have a strict linear ordering whose first node is red, node color switched every adjacent node, and the last node is blue. It follows that we have an even number of nodes, giving us our answer.
6. Let S be the conjunction of

$$\mathsf{SLO}$$

$$(\forall x)(\forall y)((Lxy \wedge \neg(\exists z)(Lxz \wedge Lzy)) \supset Fx \oplus Fy)$$

$$(\forall x)(\neg(\exists y)Lyx \supset Fx)$$

$$(\forall x)(\neg(\exists y)Lxy \supset Fx)$$

   The only difference from before is in the last conjunct; $\neg Fx$ was replaced with Fx. S now says we have a linear order, flipping colors, both ending and starting with red. So only odd-sized orders work now.
7. $\mathsf{Spec}(\mathsf{SLO}) = \mathbb{Z}^+$, as there are linear orders of every finite size.

8. $\mathsf{Spec}(S) = \emptyset$. The second conjunct says that there is no maximum element, the third says there is no minimum, and the final says that each pair of elements has a third element in between the two. Any of these conjuncts, together with SLO, requires that all models be infinite. In particular, there is no $n \in \mathbb{Z}^+$ with a model of size n, and so, by the definition of spectrum, $\mathsf{Spec}(S) = \emptyset$.

   Note, of course, that this does not mean that there are no models of S - if we interpret L as $<$, then $\mathbb{Q}$, the rational numbers together with its usual strict linear ordering. is a model of S. It only means that there are no finite models.

3.6 Isomorphisms, Automorphisms, and the Orbit-Stabilizer Theorem

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 34. *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.*

Intuitively speaking, an isomorphism is a *relabelling map*. The idea is that two structures are isomorphic if they "look the same" up to their nodes' being "relabelled". In the example above, we "relabelled" 1 as 2 and 2 as 1.

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

An automorphism is an isomorphism, that is, a relabelling map, which leaves the edge-set unchanged (and hence leaves the whole structure unchanged). h is an automorphism because the edges have exactly the same labels after the mapping. f is not an automorphism, because the edges do not have the same labels after the mapping.

Observe that if $A \cong B$, then for every schema S, $A \models S$ if and only if $B \models S$. Indeed, any language which can reasonably be called "logical" cannot distinguish between isomorphic structures. This is such a fundamental property of PQT that we pause to establish here.

Definition 36. *Structures A and B are* polyadically equivalent *(written $A \equiv_P B$) if and only if for every schema S, $A \models S$ if and only if $B \models S$.*

Theorem 4. *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 $a_1, \ldots, a_k \in U^A$,*

$$A \models S[(x_1|a_1), \ldots, (x_k|a_k)] \textit{ iff } B \models S[(x_1|f(a_1)), \ldots, (x_k|f(a_k))]. \tag{3}$$

In particular, if $A \cong B$, then $A \equiv_P B$.

Proof sketch of Theorem 4: 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 (3) 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, (3) 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 (3) 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. □

In the proof above, the only truth-functional connective we considered was conjunction. The other cases are handled similarly. Complete those cases yourself, either by writing out the whole argument, or by showing that a conditional can be defined in terms of some conditionals whose cases you already worked out (for example, each of the connectives can be defined in terms of $\neg$ and $\wedge$, so those two cases suffice).

The image of a structure

Let's continue to consider the structure A. We have the following list of 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

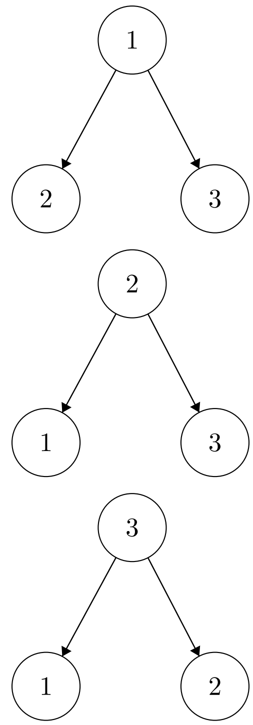
We'll call this set of bijections $\mathbb{S}_3$ (more on this notation below). We define the *image of a structure* as follows.

Definition 37. *If A is a graph with edge relation L^A and h a bijection from U^A onto U^A, then the image of A under h (written $h[A]$) is defined by*

$$U^{h[A]} := U^A, L^{h[A]} := \{\langle h(i), h(j)\rangle \mid \langle i,j\rangle \in L^A\}.$$

It follows immediately that for any D with $U^D = [3]$ and $h \in \mathbb{S}_3$, h is an isomorphism from D onto $h[D]$.

With respect to the examples A and B above, we see that $f_2[A] = f_5[A] = B$. Moreover, $C = f_3[A] = f_6[A]$ is a third isomorphic copy of A distinct from both A and B. That is, there are three labeled structures, A, B, and C, with universe [3] that are isomorphic to A (see the following diagram). Their respective automorphism classes are: $\mathsf{Aut}(A) = \{f_1, f_4\}$, $\mathsf{Aut}(B) = \{f_1, f_3\}$, and $\mathsf{Aut}(C) = \{f_1, f_2\}$.



This suggests the following marvelous identity, which we will shortly explore:

$$|\mathbb{S}_3| = |\mathsf{Aut}(A)| \cdot (\text{the number of labeled copies of } A). \tag{4}$$

The Orbit-Stabilizer Theorem

Recall that for every positive integer k, we write $[k]$ for $\{1, \ldots, k\}$.

Definition 38. *For every positive integer k, we write $\mathbb{S}_k$ for the set of bijections from $[k]$ onto $[k]$ (also called the* permutation group on *or the* symmetric group on *$[k]$).*

The names *permutation group* or *symmetric group* emphasize the agebraic nature of $\mathbb{S}_k$. Indeed, we can 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[13]:

- $\circ$ 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 $f \in \mathbb{S}_k$.
- Permutations are closed under $\circ$, that is, $f \circ g$ is a permutation for all $f, g \in \mathbb{S}_k$.

Prove that each of these conditions holds.

Definition 39. *We write $\mathbb{G}_k$ $(= \mathsf{mod}(\mathsf{SG}, k))$ for the set of simple graphs A with $U^A = [k]$.*

[13]These conditions (associativity, identity, inverse, closure) are the axioms for a *group*, which is a fundamental and widely applicable concept in algebra. Group Theory, the part of mathematics which studies groups, is a hugely influential and interesting field. MATH 370 is Penn's introductory group theory course.

Recall that for each $f \in \mathbb{S}_k$ and $A \in \mathbb{G}_k$, $f[A]$ is the image of the graph A under f. 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.

Just as with groups, group actions are axiomatized by a few simple conditions. To verify that this is indeed a group action, show that for all $A \in \mathbb{G}_k$ and $f, g \in \mathbb{S}_k$ the following properties hold:

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

Definition 40. *The* orbit of A *under the action of* $\mathbb{S}_k$ *(written* $\mathsf{orb}(A, \mathbb{S}_k)$*) is* $\{h[A] \mid h \in \mathbb{S}_k\}$.

In other words, the orbit of A under $\mathbb{S}_k$ is the set of $B \in \mathbb{G}_k$ such that $A \cong B$.

The following result is a special case of the *Orbit-Stabilizer Theorem*.

Theorem 5. *For all* $A \in \mathbb{G}_k$,

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

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 $(f^{-1} \circ g) \in \mathsf{Aut}(A)$.

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 $f^{-1} \circ g = h$. It follows at once that there is a bijection between $\{g \mid f \sim g\}$ and $\mathsf{Aut}(A)$.

Use the group axioms to first prove that inverses in a group are unique (that is, for any f in a group, there is a unique element f^{-1} such that $f \circ f^{-1} = e = f^{-1} \circ f$, where e is the identity element).

Using that fact, verify that for each fixed $f, h \in \mathbb{S}_k$, there is a unique $g \in \mathbb{S}_k$ such that $f^{-1} \circ g = h$.

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.

First, suppose suppose $f \sim g$. Then $f^{-1} \circ g \in \mathsf{Aut}(A)$. It follows that $(f^{-1} \circ g)[A] = A$ and hence that $f[(f^{-1} \circ g)[A]] = f[A]$. So $(f \circ (f^{-1} \circ g))[A] = f[A]$, and then by associativity $((f \circ f^{-1}) \circ g)[A] = f[A]$. As $f \circ f^{-1} = e$, we have $(e \circ g)[A] = f[A]$ from which it follows that $g[A] = f[A]$.

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 $\mathsf{orb}(A, \mathbb{S}_k)$. $\square$

We now have the explanation of identity (4), since

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

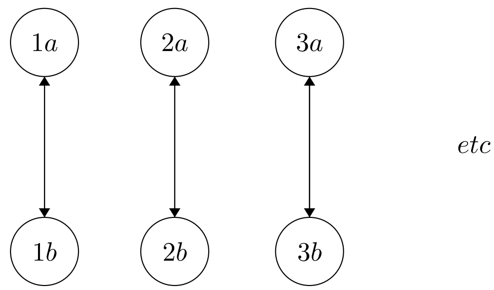
We illustrate the use of Theorem 5 via an application to counting structures that satisfy a given schema. Let S be the conjunction of SG and $\mathsf{1reg}$, that is, a graph A satisfies S if and only if A is a 1-regular, simple graph. As we discussed earlier, every such finite graph A has an even number, say $2n$, of nodes; moreover, if $A, B \models S$ and $|U^A| = |U^B|$, then A is isomorphic to B. We will calculate the value of $\mathsf{mod}(S, 2n)$ in two ways - one way using the Orbit-Stabilizer Theorem, and the other directly.

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

$$\mathsf{mod}(S, 2n) = \mathsf{orb}(A, \mathbb{S}_{2n}). \tag{5}$$

Let's calculate $|\mathsf{Aut}(A)|$, since Theorem 5 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 have for each edge a choice whether to "flip" the edge or not. 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 5 and equation (5),

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

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

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

3.7 Definability

Up to this point we have neglected schemata containing free variables. We will now correct this oversight.

Consider the structure A (which should look familiar) defined by

$$U^A = [3], L^A = \{\langle 1,2\rangle, \langle 1,3\rangle\}$$

Let $S(x)$ be the schema

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

$S(x)$ picks out 1 uniquely from the structure A, because 1 is the only element in U^A which does not have an incoming edge. Symbolically, we express this as

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

$S(x)$ 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 41. *Let $S(x)$ be a schema with one free variable x and let A be a structure. We define $S[A] = \{a \in U^A \mid A \models S[x|a]\}$. In other words, $S[A]$ is the set of nodes $a \in A$ that satisfy the schema $S(x)$ in A when we assign a to the variable x. We call $S[A]$ the* set defined by $S(x)$ *in A.*

Definition 42. *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 write $\mathsf{Def}(A)$ for the set of definable subsets of A.*

Note that the set $\{2,3\}$ is defined by the schema

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

Are either of the sets $\{2\}$ or $\{3\}$ definable as subsets of A? Try as you might, you won't find a schema which picks out either 2 or 3 individually. Intuitively, this is because the nodes labelled 2 and 3 appear to be "indistinguishable from a structural point of view". Backing up this notion of indistinguishability, we see that the function h mapping 1 to 1, 2 to 3, and 3 to 2, is an automorphism of A which happens to exchange 2 and 3. The relevance of this to the question of definability is the content of the following fundamental theorem.

The Automorphism Theorem

The following result, known as the Automorphism Theorem, is an important aid in the study of definability. It is a corollary to Theorem 4.

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

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

Show that Corollary 4 is a corollary to Theorem 4

Corollary 4 provides a useful necessary condition for a set to be definable in an arbitrary structure, and 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 Corollary 4 which bears on definability.

Corollary 5. *Let A be a graph and $h \in \mathsf{Aut}(A)$. If V is a definable subset of A, then $h[V] = V$.*

Show that Corollary 5 is a corollary to Corollary 4

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 $h(a) \notin V$.

Orbits and Definability over Finite Structures

In the case of finite structures, the converse of Corollary 5 is true.

Theorem 6. *Let A be a finite graph and $V \subseteq U^A$. V is a definable subset of A if and only if for every $h \in \mathsf{Aut}(A)$, $h[V] = V$.*

In order to prove Theorem 6, and to apply it to questions of counting definable sets, the following definitions will be useful.

Definition 43. *The* orbit of a node $a \in U^A$ under the action of $\mathsf{Aut}(A)$ *is the set of all possible images of a under actions $f \in \mathsf{Aut}(A)$. Symbolically,*

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

Definition 44. *The* orbits of A *is the set of all orbits of individual elements $a \in A$. Symbolically:*

$$\mathsf{Orbs}(A, \mathsf{Aut}(A)) = \{\mathsf{orb}(a, \mathsf{Aut}(A)) \mid a \in U^A\}$$

Proof Sketch of Theorem 6: We aim to show that for every finite graph A, $V \subset A$ is definable iff every automorphism $h \in \mathsf{Aut}(A)$ leaves V unchanged. The generalization to structures interpreting multiple polyadic predicates is straightforward.

First, 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

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

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

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

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

As a corollary to Corollary 5 and Theorem 6 we have:

Corollary 6. *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 $O_1, \dots, O_k$, where each $O_i \in \mathsf{Orbs}(A, \mathsf{Aut}(A))$, and $V = O_1 \cup \dots \cup O_k$.*

Use Corollary 5 and Theorem 6 to prove this.

It follows at once from Corollary 6, that if A is a finite graph, then the number of definable subsets of A is $2^{|\mathsf{Orbs}(A, \mathsf{Aut}(A))|}$.

Definition 45. *We say that 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 6 that if A is a finite rigid structure and $V \subseteq U^A$, then $V \in \mathsf{Def}(A)$.

Why? Think about what $\mathsf{Aut}(A) = \{e\}$ implies about the orbit of each element (and hence $\mathsf{Orbs}(A, \mathsf{Aut}(A))$).

Automorphisms and Degree

In applying our analysis of definability over finite structures in terms orbits to particular examples, it will be useful to observe the following connections between automorphisms and degree. Let A be a graph and $a \in U^A$. Recall that the *neighborhood of* a in A is $\mathsf{nbh}(a, A) := \{b \in U^A \mid \langle a, b\rangle \in L^A\}$. The *degree of* a in A is $\mathsf{deg}(a, A) := |\{b \in U^A \mid \langle a, b\rangle \in L^A\}|$. We have the following fact:

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,

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

In other words, automorphisms preserve degree.

Show that this follows from the definition of an automorphism.

An example: definable subsets of simple graphs with four nodes

To make this all a little bit more concrete, let's give a complete analysis of the definable subsets of simple graphs with four nodes.

We begin by classifying all members of $\mathsf{mod}(\mathsf{SG}, 4)$ up to isomorphism - that is, we exhibit an example of each "isomorphism-type" of size-4 simple graph. What does this mean? Recall that for every $A \in \mathsf{mod}(\mathsf{SG}, 4)$, $\mathsf{orb}(A, \mathbb{S}_4)$ is the set of B such that $B \cong A$. Thus, these orbits correspond to the isomorphism types of structures in $\mathsf{mod}(\mathsf{SG}, 4)$, and we will make a list that includes exactly one structure from each orbit. Another way of putting this is that we make a *succinct* list of structures from $\mathsf{mod}(\mathsf{SG}, 4)$, that is, a maximal length list of such structures, no two of which are isomorphic.

In general, if a group G acts on a set X, we may define an equivalence relation $\sim$ on X by $a \sim b$ if and only if there is an $f \in G$ such that $fa = b$. The equivalence class $\hat{a}$ of an element $a \in X$ with respect to this equivalence relation is the *orbit of* a under this group action, and the equivalence relation is often referred to as the *orbit equivalence relation*. Thus, in the case to hand, the orbit equivalence relation of the action of $\mathbb{S}_k$ on $\mathsf{mod}(\mathsf{SG}, k)$ we've defined is exactly the isomorphism relation.

We already know that $|\mathsf{mod}(\mathsf{SG}, 4)| = 2^{\binom{4}{2}} = 2^6 = 64$, and though this is not a very large number, nonetheless, it will useful to organize our effort systematically in order to compile a succinct list of simple graphs of size four. If graphs A and B are isomorphic, then they have the same number of edges. Let's write $\mathsf{size}(A)$ for the number of undirected edges of a simple graph A. We call $\mathsf{size}(A)$ the *edge-size* of A, in contrast to $|A|$, which is the number of vertices of A. For $A \in \mathsf{mod}(\mathsf{SG}, 4)$, $0 \leq \mathsf{size}(A) \leq 6$. For each of these seven possible edge-sizes, we will make a succinct list of the graphs of that edge-size, and then put them together to get a succinct list of all the simple graphs of size 4.

A simple observation will nearly halve our effort in compiling these lists. For a simple graph A, we define A^c, the *graph complement* of A as follows.

$$A^c : U^{A^c} = U^A; \langle i, j\rangle \in E^{A^c} \text{ iff } i \neq j \text{ and } \langle i, j\rangle \notin E^A.$$

If we like, we can visualize a simple graph as having "solid edges" (the edges that are included in the graph) and "transparent edges" (the edges that are omitted from the graph). Complementation turns the solid edges transparent, and the transparent edges solid, while retaining the loop-free character of the graph.

Lemma 4. *For all simple graphs A and B and all functions $f : U^A \mapsto U^B$, f is an isomorphism from A onto B if and only if f is an isomorphism of A^c onto B^c. Hence,*

$$A \cong B \text{ iff } A^c \cong B^c,$$

and for all functions $f : U^A \mapsto U^A$,

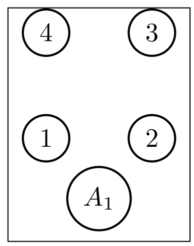
$$f \in \mathsf{Aut}(A) \text{ iff } f \in \mathsf{Aut}(A^c).$$

Moreover, $|\mathsf{orb}(A, \mathbb{S}_4)| = |\mathsf{orb}(A^c, \mathbb{S}_4)|$, *and* $|\mathsf{Aut}(A_i)| = |\mathsf{Aut}(A_j)|$.

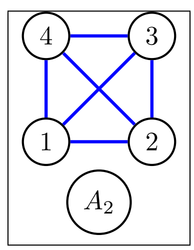
Prove Lemma 4.

It follows immediately from Lemma 4 that succinct lists of graphs in $\mathsf{mod}(\mathsf{SG}, 4)$ with edge-size $0 \leq i \leq 6$, immediately generate succinct lists with edge-size $6 - i$, via complementation. Finally, we are prepared to compile our lists.

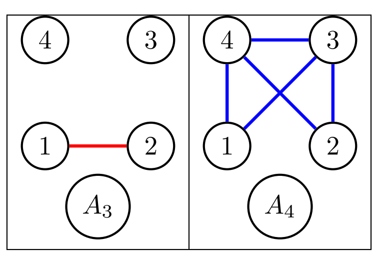
There is a single graph in $\mathsf{mod}(\mathsf{SG}, 4)$ with no edges which we will call A_1. This looks like



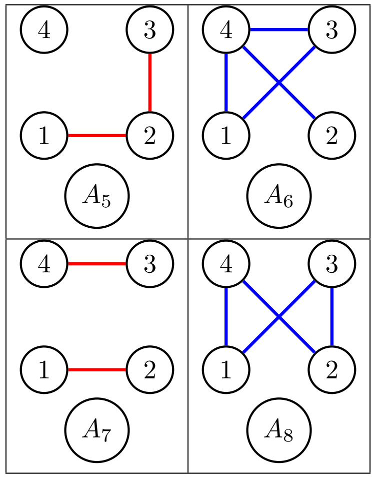
By complementation, there is also a single graph with 6 edges.



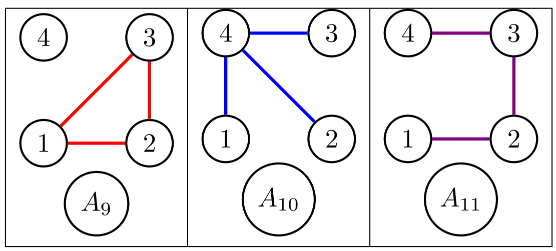
Up to isomorphism, there is one size-4 graph with a single edge (since we only care about equivalence up to isomorphism, the labels of the ends of the single edge don't matter). By complementation, there is a single size-4 graph with 5 edges.



There are two non-isomorphic size-4 graphs with two edges, and, again by complementation, two such graphs with 4 edges.



Lastly, there are three non-isomorphic size-4 simple graphs with three edges. A_9 and A_{10} are complements of each other, whereas A_{11}, is isomorphic to its own complement.



Verify that these are all of the non-isomorphic graphs of size 4 by beginning with 4 empty nodes and iteratively constructing all non-isomorphic graphs with increasing number of edges. We will later establish this via application of Theorem 5.

Now that we have a maximal collection of pairwise non-isomorphic graphs in $\mathsf{mod}(\mathsf{SG}, 4)$, we can calculate $|\mathsf{orb}(A_i, \mathbb{S}_4)|$ and $|\mathsf{Aut}(A_i)|$ for each $1 \leq i \leq 11$.

What is $|\mathsf{orb}(A_1, \mathbb{S}_4)|$, or in other words, how many distinct ways can we place 0 edges onto 4 labelled nodes? There is only one way to do this, so $|\mathsf{orb}(A_1, \mathbb{S}_4)| = 1$. What is $|\mathsf{Aut}(A_1)|$, or in other words, how many ways can we permute the edges of A_1 once the edges are fixed in place? There are no edges, so any permutation of the nodes (of which there are $4! = 24$) is valid. It follows that $|\mathsf{Aut}(A_1)| = 24$. It follows from Lemma 4 that $|\mathsf{orb}(A_2, \mathbb{S}_4)| = 1$ and $|\mathsf{Aut}(A_2)| = 24$ as well.

As another example, let's calculate $|\mathsf{orb}(A_5, \mathbb{S}_4)|$ and $|\mathsf{Aut}(A_5)|$ (and hence the values for A_6 as well). There are $4 \cdot 3 = 12$ ways of placing 2 edges onto 4 nodes such that the two edges are connected as in A_5, as there are 4 choices for the "central" node and $\binom{3}{2} = 3$ choices for which two other nodes (which we will call *leaves*) get connected to the central node. It follows that $|\mathsf{orb}(A_5, \mathbb{S}_4)| = |\mathsf{orb}(A_6, \mathbb{S}_4)| = 12$. Once the edges have been fixed, there are two possible automorphisms: the identity automorphism, and the automorphism which exchanges the two leaf nodes. It follows that $|\mathsf{Aut}(A_5)| = |\mathsf{Aut}(A_6)| = 2$.

Without too much extra work, we arrive at the complete 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

Calculate each of the values not discussed in the examples.

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

$$\sum_{1 \leq i \leq 11} |\mathsf{orb}(A_i, \mathbb{S}_4)| = 64,$$

thereby confirming that we have accounted for all structures up to isomorphism!

No member of $\mathsf{mod}(\mathsf{SG}, 4)$ is rigid, as each has non-trivial automorphisms. This suggests an interesting question: "what is the least n such that $\mathsf{mod}(\mathsf{SG}, n)$ contains a rigid graph?"

By Corollary 6, calculating $\mathsf{Orbs}(A_i, \mathsf{Aut}(A_i))$ suffices to determine which sets are definable in each A_i.

Example 12. *What is* $\mathsf{Orbs}(A_5, \mathsf{Aut}(A_5))$*?*

To determine $\mathsf{Orbs}(A_5, \mathsf{Aut}(A_5))$, it suffices to determine the orbits of individual elements. The orbit of node 4 is $\{4\}$, as it is the only isolated node. The orbit of 2 is $\{2\}$, as it is the only node of degree two. The orbit of 1 is $\{1, 3\}$ as 1 is a leaf node, and we had an automorphism that exchanged leaf nodes. As the set of orbits partition the nodes, the orbit of 3 is $\{1, 3\}$ as well. It follows that $\mathsf{Orbs}(A_5, \mathsf{Aut}(A_5)) = \{\{2\}, \{4\}, \{1, 3\}\}$.

A *partition* of a set S is a collection $\mathcal{P}$ of subsets of S such that: (1) every $s \in S$ is in some $P \in \mathcal{P}$, and (2) if $P, P' \in \mathcal{P}$ and $P \cap P' \neq 0$, then $P = P'$ (ie, no distinct elements of $\mathcal{P}$ overlap).

$\mathsf{Orbs}(A, \mathsf{Aut}(A))$ trivially satisfies condition (1), since every node in A is in its own orbit. Complete the proof $\mathsf{Orbs}(A, \mathsf{Aut}(A))$ is a partition of the nodes of A by showing that condition (2) holds.

With a little more work, we arrive at the following table.

A_i	$\mathsf{Orbs}(A_i, \mathsf{Aut}(A_i))$
A_1, A_2	$\{[4]\}$
A_3, A_4	$\{\{1, 2\}, \{3, 4\}\}$
A_5, A_6	$\{\{2\}, \{4\}, \{1, 3\}\}$
A_7, A_8	$\{[4]\}$
A_9, A_{10}	$\{\{1, 2, 3\}, \{4\}\}$
A_{11}	$\{\{1, 4\}, \{2, 3\}\}$

Derive each of the above sets of orbits yourself, to make sure all the concepts fit into place.

Definability in Infinite Structures

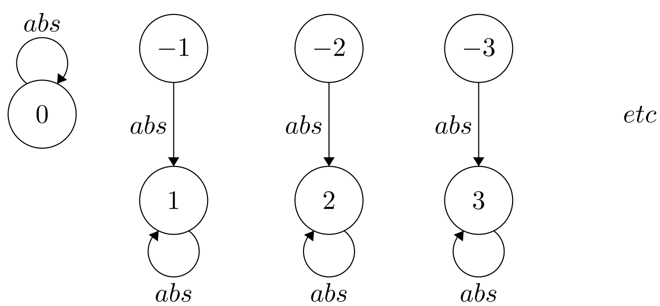
As we've just seen, Corollary 4 leads to a complete analysis of the definable subsets of finite structures - they are exactly those sets invariant under the action of the automorphism group of the structure. In the case of infinite structures, Corollary 4 still provides a useful necessary condition for definability, and in some cases leads to a complete analysis of definability. But there are many infinite structures for which this is no longer the case, and other methods are required to achieve a satisfactory understanding of definability. We explore two very different examples in this section.

A Structure With Many Automorphisms: The Integers with Absolute Value

Let A be the infinite graph defined by

$$U^A = \mathbb{Z}, L^A = \{\langle i, j\rangle \mid j \text{ is the absolute value of } i\}$$

(Recall that the absolute value of an integer i is i, if $i \geq 0$, and is $-i$, if $i < 0$.)



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 $i < 0$.

Why is this? The only relation we have in our graph is the absolute-value relation, so our graph looks like a bunch of pairs $n, -n$ (for n positive) where there is an edge from $-n$ to n and an edge from n to n (ie a self-loop at n), plus 0 all on its own with a self-loop. So long as we keep 0 fixed in place, permuting any of our $(n, -n)$-pairs gives us an automorphism, provided that we match don't "flip" any of the pairs, that is, negative numbers (which have in-degree 0) map to negative numbers, and positive numbers (which have in-degree 2) map to positive numbers. The definition given above ensures this.

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 $\neg S_1(x) \wedge \neg S_2(x)$.

$S_1[A] = \mathbb{Z}^+$ as the positive integers are the only ones which have in-neighbours distinct from themselves (because there is an edge from a negative integer to its positive absolute value). Explain in your own words why $S_2[A] = \mathbb{Z}^-$ and $S_3[A] = \{0\}$.

By Corollary 6, it follows that there are exactly eight sets definable in A:

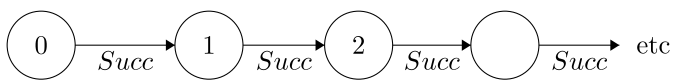
1. $\emptyset$,

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. $\mathbb{Z}$.

A Rigid Structure: The Natural Numbers with Successor

We now look at another infinite structure B where definability behaves very differently. B is described by:

$$U^B = \mathbb{N}, L^B = \{\langle i, j\rangle \mid j = i + 1\}$$



A first observation is that $\mathsf{Aut}(B) = \{e\}$, that is, B is a rigid structure. Intuitively, any automorphism must map 0 to itself, since it is the only element which doesn't have anything less than it. Similarly, any automorphism must map any positive n to itself, since n is the only number with exactly $n-1$ predecessors.

We can establish this formally 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 $h(n+1) = n+1$. It follows that 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.

Why is it important that z be a variable which occurs nowhere in $S^n(x)$?

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?

3.8 Undefinability

Cantor's Theorem and Cardinality Arguments

We will 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, $W \neq V_i$. $\square$

The idea in the above proof is to show that, regardless of which way we list the subsets V_i of U, there will always be some other subset W of U which is not in the list. We construct W by making sure it differs from each V_i by at least one element; to do this, it suffices to let $a_i \in W$ iff $a_i \notin V_i$.

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. $\square$

Theorem 8 entails that we can list all definable subsets of an infinite structure C, and Theorem 7 entails that no list can exhaust all the definable subsets of an infinite set. So we have our result:

Corollary 7. *For every infinite structure C there is a subset $X \subseteq U^C$ which is not definable over C.*

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 required to state the Compactness Theorem for Polyadic Quantification Theory.

Definition 46. *A schema S is* satisfiable *if and only if for some structure A, $A \models S$.*

Definition 47. *A set of schemata Γ is* satisfiable *if and only if there is structure A such that for every schema $S \in \Gamma$, $A \models S$.*

Definition 48. *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. We will discuss this development in Section 3.10. 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 toward a contradiction that a schema $T(x)$ defines a set V which is neither finite nor co-finite over B. 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 $T(x)$, or any of the schemata $S^n(x)$ for $n \geq 0$ (recall that $S^n(x)$ says that x is the n^{th} successor of the unique element with no predecessors).

Define the set of schemata Γ as follows.

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

Let Δ be a finite subset of Γ. As both $T[B]$ and $\neg T[B]$ are infinite by hypothesis, Δ can be 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 (whereas every element $b \in B$ is reachable in this manner).

We will show that there is an automorphism h of C with $h(a) = b$. This will yield the desired contradiction, since $C \models T(y|a)$ and $C \models \neg T(z|b)$.

Note that B, and hence C, satisfy the following schemata.

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

  $\vdots$

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

  $\vdots$

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.

Why does this ensure 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 $h(a) = b$ (if they lie on a single $\mathbb{Z}$-chain, shifting the $\mathbb{Z}$-chain works as an automorphism, whereas if they lie on two $\mathbb{Z}$-chains, interchanging the $\mathbb{Z}$-chains suffices). $\square$

3.9 The Expressive Power of PQT

In the preceding sections we've studied aspects of the expressive power of first-order logic. We've seen that various well-known properties of directed graphs can be expressed using schemata of PQT, for example, reflexivity, transitivity, and symmetry. Moreover, we've exhibited several infinite/co-infinite sets, such as the powers of two, that are spectra of PQT-schemata. Finally, in the immediately preceding section, we've considered which subsets of the universe of a given structure are definable by one variable open schemata over the structure. We gave a complete analysis of the situation for finite structures in terms of automorphisms and showed that PQT is expressive as possible in this case, since no logical language can distinguish objects that lie in the same orbit of the group of automorphisms on a structure. On the other hand, in the infinite case, we saw that there are intrinsic limitations on definability owing to general cardinality considerations. Moreover, we saw that special properties of PQT, in particular compactness, enable us to demonstrate substantial limits to definability over particular infinite structures. In this section, we will explore the expressive power of PQT in greater depth and consider the extent to which the "whole truth" about a structure, as expressed by schemata of PQT, can characterize the structure up to isomorphism. Again, we will observe a dramatic difference between the case of finite and infinite structures.

The Theory of a Structure

When we speak of the "whole truth" about a structure A, we mean the *theory of* A defined as follows.

Definition 49. *The* theory of *a structure A is the set of all PQT-schemata that are satisfied by A, that is,*

$$\mathsf{Th}(A) = \{S \mid A \models S\}.$$

Structures A and B are PQT-equivalent *(written $A \equiv_P B$) if and only if $\mathsf{Th}(A) = \mathsf{Th}(B)$. That is, $A \equiv_P B$ if and only if A and B are indistinguishable by PQT-schemata.*

We have already seen that if structures A and B are isomorphic, then $A \equiv_P B$. We turn now to consider the circumstances under which the converse may hold.

The Finite Case

In the finite case, non-isomorphic structures are distinguishable by schemata of PQT.

Theorem 11. *If A is a finite graph and $A \equiv_P B$, then $A \cong B$. Indeed, for every finite graph A, there is a schema S such that for every graph B,*

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

Proof Sketch: Intuitively, if a graph is finite, you only need to specify finitely many things about it, that is, how many nodes there are and which of these are connected by edges, in order to describe it up to isomorphism. In fact, this is exactly what done in the proof sketch of Theorem 6. We encourage the reader to refer to that proof and supply the necessary details here. □

The Infinite Case

The following result stands in sharp contrast to Theorem 11.

Theorem 12. *For every infinite graph A, there is a graph B, $B \equiv_P A$, but $B \ncong A$.*

Theorem 12 is a corollary to Theorem 7 and the following proposition, which is a version of the Löwenheim-Skolem Theorem.

Theorem 13. *For every infinite graph A and every infinite set X, there is an infinite graph B such that $U^B = X$, and $A \equiv_P B$.*

Proof Sketch of Theorem 12: It follows at once from Theorem 7 that for every infinite graph A there is an infinite set X such that there is no bijection from U^A onto X. By Theorem 13, there is a graph B such that $U^B = X$ and $A \equiv_P B$. But then there is no bijection from U^A onto U^B, hence $B \ncong A$. □

3.10 Proof

Philosophy

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. We now leave expressivity and turn towards a study of implication in the context of polyadic quantification theory.

As we saw before, 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[14]. In the case of MQT, though there are infinitely many structures interpreting the vocabulary of a fixed schema S, by means of the Small Model Theorem 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; see Definition 53 for a definition of semi-decidability).

We will soon begin a detailed study of systematic techniques to establish that a schema of polyadic quantification theory is valid. For simplicity, we will again just consider schemata in the vocabulary with identity and a single dyadic predicate letter L (the language of love, or directed graphs, as the less romantic are wont to say). Let's remind ourselves of the relevant definitions.

Definition 50. *A schema S is* valid *if and only if for every structure A, $A \models S$. We write* VAL *for the set of valid schemata (in the language of directed graphs).*

Definition 51. *A schema S is* finitely valid *if and only if for every structure A, with $U^A = [n]$, for some n, $A \models S$. We write* FVAL *for the set of finitely valid schemata (in the language of directed graphs).*

If we think about what it means for a schema to be valid (ie, to be true in all models – including arbitrarily large infinite models), there is no evident way to describe a procedure for determining if a schema is valid: there are too many structures, and for some infinite structures A we may have no mechanical procedure to determine whether A satisfies a given schema. Finite validity, or at least it's complement (finite invalidity), fares better.

FVAL is Semi-Decidable

For any directed graph A with universe $[n]$ and any schema S, we can mechanically decide whether $A \models S$. Moreover, we can design a procedure, call it M to effectively enumerate all such graphs in a sequence $A_0, A_1, \ldots$ and successively test whether $A_i \models S$ for a given schema S.

When we say "design a procedure", what we mean formally is *specify a Turing Machine*. A *Turing Machine* is a simple model of computation which (all reasonable mathematicians and compter-scientists agree) completely captures the intuitive notion of "computability"; that is, if something can be "computed" in the intuitive sense of being calculated by a sequence of mechanical actions, a suitably specified Turing Machine could compute it as ell, and vice-versa.

Turing Machines, along with other equivalent formulations of computation (eg the Lambda Calculus,

[14] Of course, it remains an open problem – the P/NP problem – whether validity itself can be decided efficiently.

Markov Algorithms, Partial Recursive Functions, etc), all act as a formal basis for our study of provability. If you are interested in Turing Machines or computation, CIS 262 is Penn's relevant introductory course.

If $S \notin \mathsf{FVAL}$, then M will eventually discover this, since in this case there is an i such that $A_i \not\models S$. On the other hand, if $S \in \mathsf{FVAL}$, M will run forever with input S and we will get no information, ever waiting to see a non-existent counterexample to S. We say M is a *semi-decision procedure* for the complement of FVAL: given any schema S, M correctly identifies S as not finitely valid, if this is the case, and provides no information (the computation via M *diverges*) otherwise.

Definition 52. *A* semi-decision procedure *for a set X is a mechanical procedure (eg a Turing Machine) which, when give input A, outputs "TRUE" within finite time if $A \in X$ and and runs indefinitely (eg, diverges) if $A \notin X$.*

Definition 53. *We say a set X is* semi-decidable *if there is a semi-decision procedure for X.*

Definition 54. *A set X is* decidable *if there is a mechanical procedure (eg, Turing Machine) which, given input A, outputs "TRUE" in finite time if $A \in X$ and outputs "FALSE" in finite time if $A \notin X$.*

By the Church-Turing Thesis (the belief that Turing Machines adequately capture the intuitive notion of computation), this coincides with the informal notion of decidability we have used throughout the course.

Note what is critical here: there is a decidable relation on finite graphs A and schemata S, namely the relation of satisfaction, and a means of effectively enumerating all finite graphs. We can think of a finite graph which falsifies a schema S as a proof that S is *not* valid. That is, we can think of the procedure M as a proof-search procedure for non-finite-validity. Note, if there were such a procedure M^* for finite validity, then finite validity would be decidable.

Why? Because with input a given schema S, we could execute the procedures M and M^* simultaneously with input S. One of the two is guaranteed to terminate and yield the correct answer. In general, if a set X and its complement are both semi-decidable, then X is decidable.

Let's return to consider VAL. Again, the definition of VAL suggests no semi-decision procedure for either VAL or its complement. Already many times during the course we have presented arguments to establish the validity of one schema or another, or for various general statements about finite graphs, *etc.* Such arguments ahve been informal, but, let us hope, rigorous. That is, they proceeded by means that established that their conclusions were valid or were implied by their premisses. Of course, the arguments were not entirely explicit, so it was always legitimate to ask for one step or another to be elaborated to clarify its legitimacy. We might wonder: was the original argument a proof of its conclusion, or only the argument with the elaboration – because if the original argument did not carry conviction, it wasn't a proof. Considerations of this sort lead in the direction of demanding ever higher standards for the explicitness of proofs and ultimately to the quest for formal proof. In the context of polyadic quantification theory, we may represent this as the quest for a system of deduction with a decidable proof relation, that is, a mechanically decidable relation $\mathsf{ded}(D, S)$ which holds between a sequence of schemata D and a schema S if and only if D is a deduction of S via the rules of the system. We require that the system allow to deduce only valid schemata, that is, it should have the *Soundness Property*: if there a deduction D such that $\mathsf{ded}(D, S)$, then $S \in \mathsf{VAL}$. Moreover, it would be desirable if our system would allow us to deduce every valid schema, that is, that it would have the *Completeness Property*: if $S \in \mathsf{VAL}$, then there is a deduction D such that $\mathsf{ded}(D, S)$.

Definition 55. *A proof-system is* sound *if every provable sentence is valid. Schematically:*

$$(\exists D)(\mathsf{ded}(D, S)) \textit{ implies } S \in \mathsf{VAL}$$

Definition 56. *A proof-system is* complete *is every valid sentence is provable. Schematically:*

$$S \in \mathsf{VAL} \textit{ implies } (\exists D)(\mathsf{ded}(D, S))$$

History and Epistemology of Proof

The elaboration of formal systems of deduction for polyadic quantification caps a long effort to achieve the highest possible degree of rigor in mathematical argumentation. This search was in part motivated by the periodic appearance of contradictions in the mathematical theory of the continuum (the real numbers). This theory, whose genesis may be dated to the Pythagoreans' proof that the square root of two is irrational, was developed with great vigor in the seventeenth century, in connection with the rise of the new physics and its effort to provide a unified theory of the motion of both terrestrial and celestial bodies. As mathematical analysis (as the theory of the continuum came to be called) developed in the nineteenth century, and became ever more enmeshed with new areas of physics, such as the theory of heat, the need for a more rigorous foundation for the subject became ever more pressing. In particular, even the greatest of mathematicians, such as Augustin Cauchy, were hampered by the lack of a perspicuous notation for iterated quantification in formulating suitable convergence conditions guaranteeing continuity for the limits of sequences of functions. Throughout the nineteenth century several mathematicians, among them Bernard Bolzano, Georg Cantor, Cauchy, and Richard Dedekind, strove to place the subject of analysis on a firm footing by reducing the the theory of the continuum to the theory of the integers (arithmetic) through the use of sets or sequences of rational numbers; the outcome of these efforts came to be known as "the arithmetization of analysis" and was celebrated by David Hilbert in his famous 1900 address to the International Congress of Mathematicians held in Paris as one of the great achievements of nineteenth-century mathematics. Late in the century, Gottlob Frege sought an even greater economy in the basic principles required for the rigorous foundation of analysis through his attempt to reduce arithmetic to logic. Though this effort was ultimately doomed by Russell's paradox, Frege's articulation of a calculus for logical deduction was a signal achievement in the development of modern logic. In reaction to the paradoxes, Hilbert, in collaboration with various of his students, and a number of other mathematicians, developed *formal systems* of logic of the sort expounded in contemporary treatments deductive logic such as Goldfarb's text.

From an epistemological point of view, one might insist that a mathematical proof should be self-certifying, that is, if the derivation D is a proof of the mathematical statement S, then this should be immediately recognizable – no further argument should be required to convince someone of this, for otherwise, it is not D itself, but only D supplemented with this additional argument, that constitutes a proof of s. The notion of formal system takes this insistence to a natural limit: in a formal system the relation "D is a proof of S" is mechanically decidable, that is, there is an algorithm which can be applied to the pair $\langle \mathsf{D}, \mathsf{S} \rangle$ to determine whether the proof relation obtains. In a formal system of deduction $\mathbb{F}$ a derivation D consists of a finite sequence of schemata, and a statement S is represented by a schema S. We write $\Pi_{\mathbb{F}}(\mathsf{D}, S)$ for "S is a proof of S in the formal system $\mathbb{F}$." The schema S is a theorem of $\mathbb{F}$ if and only if there is a derivation d such that $\Pi_{\mathbb{F}}(\mathsf{D}, S)$. We write $\vdash_{\mathbb{F}} S$ for "S is a theorem of $\mathbb{F}$." In like fashion, we write $X \vdash_{\mathbb{F}} S$ for "S is derivable from hypotheses X in $\mathbb{F}$."

Our Proof System

We will use the proof-system for PQT which is described in detail on pages 181-216 of Warren Goldfarb's *Deductive Logic*, often called *natural deduction.* The qualifier "natural" is meant to indicate that we are able to reason relatively naturally within this formal system; in particular, natural deduction allows us to make arguments of the form

1. Suppose A
2. Hence B
3. Therefore, if A, then B

This pattern involves introducing a new premise in step (1), inferring something from it in step (2), and then *discharging* the premise in step (3). The ability to introduce and then discharge premises distinguished natural deduction from other systems of deduction which often require longer proofs.

When we work out a deduction, we will do some bookkeeping in order to keep track of which premises we use at each stage, as well as which *rules of inference* are being used. As such, the form of our deductions will be as follows:

1. Each line in a deduction will be numbered, beginning at (1)
2. To the right of each line number will be the current schema under consideration
3. To the left of each line number, there will be a set indicating the line numbers of premises upon which the current schema depends
4. To the right of each schema we will place an acronym indicating which rule was used to arrive at the current schema, as well as the line number of the schema which that rule acted on.

For example, the following line would indicate that: this is the 4^{th} line of a deduction which depends on a premise from line 3, the current schema is $(\exists y)Lwy$, and was derived from line (3) by means of the rule *Universal Instantiation*

$$\{3\} \quad (4)\ (\exists y)Lwy \quad (3)\text{ UI}$$

Asymmetric implies Irreflexive

We refer the reader to Goldfarb for an explanation of each of the rules of inference. Here, we present various deductions and explanations as example. First, 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

We first introduce the premise $(\forall x)(\forall y)(Lxy \supset \neg Lyx)$ (ie, we assume asymmetry), then universally instantiate twice to strip off the universal quantifiers (and thereby achieve a truth-functional schema). We can then make the truth-functional deduction $Lxx \supset \neg Lxx$ *truth-functionally implies* $\neg Lxx$. Lastly, we universally generalize to get the intended result.

Transitive and Irreflexive implies Asymmetric

We show 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

On line (1), we introduce a premise for transitivity. Lines 2-4 strip off the universal quantifiers to get the truth-functional part of the transitivity schema. Line (5) introduces a premise for irreflexivity, line (6) strips off its quantifier. Line (7) is a truth-functional inference from lines (4) and (6), an then lines (8, 9) reintroduce the universal quantifiers to achieve the intended result.

Argument by Cases

Here is an example which illustrates the use of "argument by cases", which is used when we wish to show that a disjunction $A \vee B$ implies some schema S. To argue by cases, we show that A implies S and B implies S, then truth-functionally infer that $A \vee B$ implies S.

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

$\{1\}$	(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

Line (1) introduces our main premise. Line (2) introduces our first case, and line (3) universally instantiates it. Line (4) is a truth-functional inference from line (3), and line (5) universally generalizes line (4) in order to prepare for line (6), which discharges our premise from line 2 to show that the implication holds in the first case. Lines (7 - 11) play the same role for the second case as lines (2 - 6) did for the first case. Line (11) is a truth-functional inference from the two implications we just proved, and gives the intended result.

Quantifier Conversion (DeMorgan's Laws)

We give a pair of deductions that legitimate the "conversion of quantifiers rule" which allows passing directly from $\neg(\forall x)S$ to $(\exists x)\neg S$ and *vice versa*. These quantifier-conversion rules are often called *DeMorgan's Laws.*

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

$\{1\}$	(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) $(\forall x)Fx \wedge \neg(\forall x)Fx$	(6) TF
$\{1\}$	(8) $\neg(\exists x)\neg Fx \supset ((\forall x)Fx \wedge \neg(\forall x)Fx)$	$\{2\}$(7) D
$\{1\}$	(9) $(\exists x)\neg Fx$	(8) TF

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

$\{1\}$	(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) $\neg(\exists x)\neg Fx \wedge (\exists x)\neg Fx$	(6) TF
$\{2\}$	(8) $(\forall x)Fx \supset (\neg(\exists x)\neg Fx \wedge (\exists x)\neg Fx)$	$\{1\}$(7) D
$\{1\}$	(9) $\neg(\forall x)Fx$	(8) TF

Reductio ad Absurdum

Here is an example of argument by *reductio ad absurdum*, that, in addition, illustrates the use of the "conversion of quantifiers" rule we just deduced.

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

$\{1\}$	(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

Lines (1-7) derive a contradiction from the assumption the premise on line (1), which is the negation of what we intend to show. The intended result then follows by truth-functional implication.

Existential Generalization and Instantiation

The following gives an example of the use of existential generalization and instantiation, which allow us to mirror common informal forms of argument involving the existential quantifier.

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

$\{1\}$	(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

When existentially instantiating, we replace a quantified variable (eg, $(\exists x)$) with a new constant (eg, w). It is important to instantiate using a new variable which does not occur elsewhere in the schema.

Working With Identity

Finally, we illustrate the use of the identity rules explained in Goldfarb.

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

$\{1\}$	(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

Showing Satisfiability

Our last consideration will be the problem of establishing that a set of schemata X is satisfiable. As we have noted, there is no uniform approach to this problem, since the collection of satisfiable schemata is *not* semi-decidable. As such, showing satisfiability of a sentence X amounts to constructing a structure A such that $A \models X$.

We give an example. 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 $X = \{S\} \cup \{R^n \mid n \geq 2\}$.

Is X satisfiable? The first conjunct denotes transitivity, the second comparability, and the third irreflexivity, so we know we are working with a strict linear order. The fourth and fifth conjuncts say that successor and predecessor are both discrete. The sizth and seventh conjuncts say that there is a first and last element, respectively. The last set of schemata suffices to say that we must have infinitely many elements.

At first, you may think that X is not satisfiable - after all, a discrete linear order with endpoints certainly sounds like it must be finite. Intuition is often tricky when dealing with the infinite, though, so it's best to be careful. In fact, the union of the first seven schemata with any finite subset Δ of the set $\{R^n \mid n \geq 2\}$ is satisfiable - if m is the largest integer for which R^m appears in Δ, a strict linear order of size m suffices. So ever finite subset of X is satisfiable, and hence by the Compactness Theorem, X itself is satisfiable.

Since X is satisfiable, we must be able to construct some model for A it. One such structure A is defined as follows.

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

Explain why $A \models X$.

3.11 Review

Concept Review

Isomorphisms: An *isomorphism* is a function which preserves the structure of a model. Formally, an isomorphism from A onto B is a bijection f from U^A to U^B such that $\langle f(i), f(j) \rangle \in L^B \iff \langle i, j \rangle \in L^A$.

Automorphisms: An *automorphism* an isomorphism from a structure A onto itself; in other words, it is an isomorphism which leaves the edge-set unchanged.

The *automorphism class* of a structure is the set of all functions which are automorphisms on that structure.

Image of a Structure: The *image of a structure* A under a function f is the structure with same universe and edge relation defined by $L^{f[A]} := \{\langle f(i), f(j) \rangle \mid \langle i, j \rangle \in L^A\}$.

Orbit of a node: The orbit of a node a in a graph A is the set of all possible images of that node under automorphisms of the graph. Intuitively, this is the set of all nodes which "look the same, structurally" as a.

Orbit of a graph: The orbit of a graph A of size k under the action of $\mathbb{S}_k$ (the symmetry group of size k) is the maximal set of pairwise non-automorphic graphs which can be obtained from A by actions in $\mathbb{S}_k$. In other words, this is the maximal set of graphs isomorphic to A, all with different edge-sets.

Definability in the Finite: The only definable subsets of a finite graph are exactly the orbits of its nodes. We categorized all size-4 simple graphs and found all of their definable subsets.

Definability in the Infinite: Every definable subset of an infinite graph is the orbit of some node. It is not the case (in general) that every orbit is definable, however, and so one must give an explicit schema which defines an orbit in order to assert that it is definable. We saw the examples of the integers with absolute-value and the natural numbers with successor. In the former case, we showed that there were 8 possible definable sets, generated by the three orbits $\{0\}, \mathbb{Z}^-, \mathbb{Z}^+$. In the latter case, we used the Compactness Theorem to show that the only definable subsets were the finite and cofinite sets.

Orbit-Stabilizer Theorem: We gave a proof for the Orbit-Stabilizer Theorem, which states that $|\mathbb{S}_n| = |\mathsf{orb}(A, n)| \cdot |\mathsf{Aut}(A)|$.

Proof: We motivated the need for a formal system of proof, and showed that the set of finitely-valid sentences was *semi-decidable*. We remarked that the set of valid sentences does not have an obvious semi-decision procedure like finite validity does, but that validity is still semi-decidable (!) because of the coinciding nature of truth and proof: the *soundness* and *completeness* theorems ensure that every valid formula is provable and vice-versa, and so by enumerating proofs we obtain a semi-decision procedure for validity. We remarked that non-validity (or equivalently, satisfiability) is not semi-decidable, because if it were then validity would itself be decidable (contradicting the Church-Turing Theorem).

We gave examples of various proofs using the system of *natural deduction* explained in Goldfarb.

Problems

1. We say that a list L of structures is *succinct* iff no pair of structures on the list is isomorphic. Give a maximal succinct list of mod $(S, 3)$ where

$$S := (\forall x)(\exists y)(\forall z)(Lxz \equiv z = y)$$

2. For each structure A in your list L and each $O \in \mathsf{Orbs}(A, \mathsf{Aut}(A))$, give a schema $S(x)$ such that $S[A] = O$.

3. Let A be the structure with triadic predicate P defined by

$$U^A = \mathbb{Z}, P^A = \{\langle i, j, k\rangle \mid |i - j| = k\}$$

   Is $X = \{i \in \mathbb{Z} \mid i < 0\}$ definable in A?

4. Let B be the structure with triadic predicte Q defined by

$$U^B = \mathbb{Z}, Q^B = \{\langle i, j, k\rangle \mid i + j = k\}$$

   Is $X = \{i \in \mathbb{Z} \mid i < 0\}$ definable in B?

5. Let X be the conjunction of the following schemata.

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

   Is X satisfiable?

6. Let

$$X = (\exists y)(\forall x)(Lxy \vee Lyx)$$
$$S = (\forall x)(\exists y)(Lxy \vee Lyx)$$

   Does X imply S? If so, give a deduction. If not, give a counterexample.

7. Let

$$X = (\exists^{=5} x) \wedge (\forall x)(\neg Lxx) \wedge (\forall xy)(Lxy \supset Lyx)$$
$$S = (\exists xyz)(Lxy \wedge Lxz \wedge Lyz) \vee (\exists xyz)(\neg Lxy \wedge \neg Lxz \wedge \neg Lyz)$$

   Does X imply S? If so, give a deduction. If not, give a counterexample.

8. Let S be the schema

$$(\forall x)(Fx \supset (\exists y)(\neg Fy \wedge (\forall z)(Lxz \equiv y = z)))$$

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

$$(\forall y)(\neg Fy \supset (\exists x_1) \ldots (\exists x_n) \bigwedge_{1 \leq i < j \leq n} (x_i \neq x_j \wedge Fx_i \wedge Lx_i y));$$

and for each $n \geq 2$, let T_n be the schema

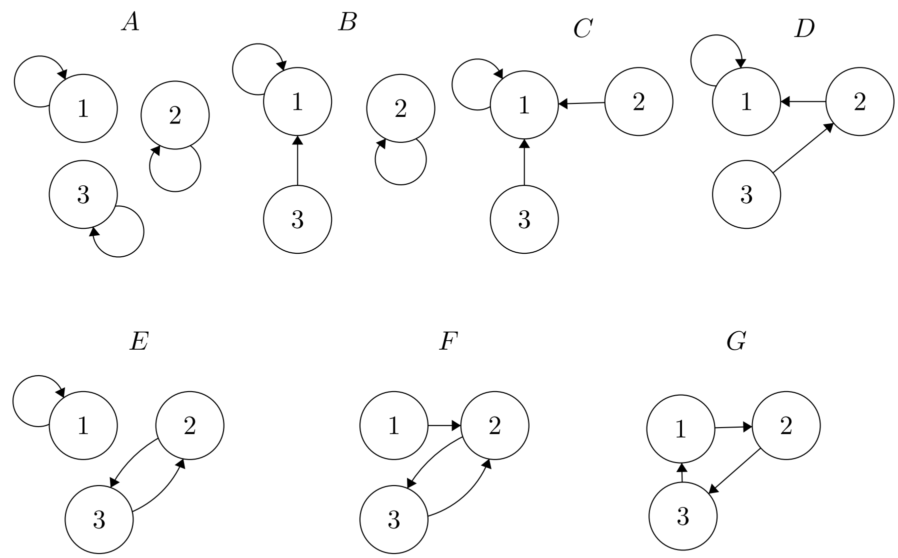
$$(\exists x_1) \ldots (\exists x_n) \bigwedge_{1 \leq i < j \leq n} (x_i \neq x_j \wedge \neg Fx_i).$$

Let $X = \{S, R_n, T_n \mid n \geq 2\}$.

Is X satisfiable?

Solutions

1. S suffices to say that L is a function. Drawing size-3 graphs leads us to the following maximal collection.



2. Let $O_1 = \{1\}, O_2 = \{2\}, O_3 = \{3\}, O_4 = \{2, 3\}, O_5 = \{1, 2, 3\}$. Then
   - $\mathsf{Orbs}(A, \mathsf{Aut}(A)) = \{O_5\}$. This can be defined by the schema $(\forall x)(x = x)$.
   - $\mathsf{Orbs}(B, \mathsf{Aut}(B)) = \{O_1, O_2, O_3\}$. O_1 can be defined by $Lxx \wedge (\exists y)(Lyx \wedge y \neq x)$, O_2 can be defined by $Lxx \wedge \neg(\exists y)(Lyx \wedge y \neq x)$, and O_3 can be defined by $\neg Lxx$.
   - $\mathsf{Orbs}(C, \mathsf{Aut}(C)) = \{O_1, O_4\}$. O_1 can be defined by Lxx, and O_4 can be defined by $\neg Lxx$.
   - $\mathsf{Orbs}(D, \mathsf{Aut}(D)) = \{O_1, O_2, O_3\}$. O_1 can be defined by Lxx, O_2 can be defined by $(\exists y)(Lyx \wedge x \neq y)$, and O_3 can be defined by $\neg(\exists y)Lyx$.
   - $\mathsf{Orbs}(E, \mathsf{Aut}(E)) = \{O_1, O_4\}$. O_1 can be defined by Lxx, and O_4 can be defined by $\neg Lxx$.
   - $\mathsf{Orbs}(F, \mathsf{Aut}(F)) = \{O_1, O_2, O_3\}$. O_1 can be defined by $\neg(\exists y)Lyx$, O_2 can be defined by $(\exists yz)(Lyx \wedge Lxz \wedge z \neq y)$, and O_3 can be defined by $(\exists y)(\forall z)(Lzx \equiv y = z)$.
   - $\mathsf{Orbs}(G, \mathsf{Aut}(G)) = \{O_5\}$. This can be defined by the schema $(\forall x)(x = x)$.

3. Yes, it is definable. P^A is the *distance relation*; $\langle i, j, k\rangle \in P^A$ iff the distance between i, j is k. No two numbers have a negative distance, so we can define the set of all negative numbers by the schema

$$(\forall yz)(\neg Pyzx)$$

4. No, it is not definable. The function h defined by $h(i) = -i$ is an automorphism of B, but $h[X] \neq X$.

5. Yes, X does imply S. Here is a derivation

$\{1\}$	(1)	$(\exists y)(\forall x)(Lxy \vee Lyx)$	P
$\{1, 2\}$	(2)	$(\forall x)(Lxy \vee Lyx)$	(1)y EII
$\{1, 2\}$	(3)	$(Lxy \vee Lyx)$	(2) UI
$\{1, 2\}$	(4)	$(\exists y)(Lxy \vee Lyx)$	(3) EG
$\{1\}$	(5)	$(\exists y)(Lxy \vee Lyx)$	(4)$\{2\}$ EIE
$\{1\}$	(6)	$(\forall x)(\exists y)(Lxy \vee Lyx)$	(5) UG

6. X does not imply S. X states that we have a simple graph of size 5, and S expresses the *three-mutuality* that we considered on day 1 of class. We shows that a "friendship pentagon" lacked 3-mutuality; that same pentagon acts as a counterexample here.

7. Yes, it is. To show this, we give a satisfying model for X. Recall that $\mathbb{Z}$ is the set of integers and $\mathbb{Q}^+$ is the set of positive rational numbers. Let A be defined by

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

   Then $A \models X$.

8. Yes, it is. We show that X is satisfiable by constructing a structure B with $B \models X$. B is defined by

   - $U^B = \mathbb{Z}^+$.
   - $F^B = \{2i \mid i \in \mathbb{Z}^+\}$.
   - $L^B = \{\langle 2^i \cdot j, j\rangle \mid i \in \mathbb{Z}^+ \text{ and } j \notin F^B\}$.

4 Appendix: A Proof of the Compactness Theorem

We prove the Compactness Theorem for First-Order Logic (PQT).

Theorem 14. *(Compactness) Let T be a set of schemata. If every finite $T_0 \subseteq T$ is satisfiable, T is satisfiable.*

Definition 57. *Let X be a nonempty set. A collection $\mathcal{F}$ of subsets of X has the* finite intersection property *if every (nonempty) finite intersection of sets in $\mathcal{F}$ is nonempty.*

Definition 58. *Let X be a nonempty set. A collection $\mathcal{F}$ of subsets of X is a* filter *iff*

1. $\mathcal{F} \neq \emptyset$
2. $\emptyset \notin \mathcal{F}$
3. $A, B \in \mathcal{F}$ *implies* $A \cap B \in \mathcal{F}$ *($\mathcal{F}$ is* closed under intersection*)*
4. $A \in \mathcal{F}$ *and* $A \subseteq B \subseteq X$ *implies* $B \in \mathcal{F}$ *($\mathcal{F}$ is* closed under supersets*).*

Lemma 5. *Suppose $\mathcal{F}$ is acollection of subsets of X with the finite intersection property. Let $\mathcal{F}'$ be the collection of nonempty finite intersections of elements of $\mathcal{F}$, and let*

$$\mathcal{F}^* := \{Y \subseteq X \mid (\exists Z \in \mathcal{F}')(Z \subseteq Y)\}$$

Then $\mathcal{F}^*$ is a filter.

Prove that each of the properties of a filter hold for $\mathcal{F}^*$. For example, (1) holds because $\mathcal{F}$ is nonempty, and (2) follows from the finite intersection property.

Definition 59. *An* ultrafilter *$\mathcal{U}$ on a set X is maximal filter (wrt containment), eg if $\mathcal{U} \subseteq \mathcal{U}'$ and $\mathcal{U}'$ is also a filter on X, then $\mathcal{U} = \mathcal{U}'$.*