Mathematical and Logical Basis of Computing: a Workbook

Howard A. Blair

1 A quick reference: some definitions and notation

This section is intended to serve only as a quick reference when you need to look up a definition or what some symbol or notation means. We will add to this section from time to time.

The next section contains a few questions that illustrate the kinds of questions that will be on the first exam. This exam is a tests the mathematical reasoning skills you should have when you **finish** the course. It does not count in any way towards your grade in this course.

The course properly begins with section 3 where the assessment exam will be discussed.

Definition 1.1: For any sets A and B, the set $A \times B$, called the *Cartesian Product* of A and B, is the set of all ordered pairs (a, b) such that $a \in A$ and $b \in B$. In other words,

$$A \times B = \{(a, b) \mid a \in A \text{ and } b \in B\}.$$

△

Example 1.1: Let $A = \{0, 1, 2\}$ and let $B = \{0, 1\}$. Then

$$A \times B = \{(0, 0), (0, 1), (1, 0), (1, 1), (2, 0), (2, 1)\}.$$

△

Example 1.2: Let $\mathbb{N}$ be the set of non-negative integers. That is, $\mathbb{N} = \{0, 1, 2, 3, \dots\}$. Then

$$\mathbb{N} \times \mathbb{N} = \{(m, n) \mid m \in \mathbb{N},\ n \in \mathbb{N}\}.$$

△

Definition 1.2: For any sets A and S, A is a *subset* of S if, and only if, every member of A is a member of S. We denote that A is a subset of S by $A \subseteq S$. △

Definition 1.3: For any set S, the *power set* of S, denoted by $\mathbf{P}(S)$, is the set of all subsets of S. That is,

$$\mathbf{P}(S) = \{B \mid B \subseteq S\}.$$

△

Note 1.1: The definition of power set implies that for anything x, $x \in \mathbf{P}(S)$ if, and only if, $x \subseteq S$. △

Definition 1.4: A *dyadic relation* (sometimes called a *binary* relation) R from set A to set B is a triple (A, B, R'), where R' is subset of $A \times B$. We often refer just to the subset of $A \times B$ when A and B are clear from the context; we say or write R when we mean (A, B, R'). △

Definition 1.5: A dyadic relation f from A to B is called a *function* from A to B if, and only if, for each $a \in A$ there is exactly one $b \in B$ such that $(a, b) \in f$. Suppose f is a function from A to B. Given $a \in A$, the $b \in B$ for which $(a, b) \in f$ is denoted by $f(a)$. The expression $f : A \longrightarrow B$ means that f is a function from A to B. △

Definition 1.6: The *composition* of dyadic relation R from set A to set B with dyadic relation S from B to set C is denoted by $S \circ R$ and is a dyadic relation from A to C defined by

$$(x, z) \in S \circ R \quad \text{iff} \quad \text{there is an element } y \text{ of } B \text{ such that } (x, y) \in R \text{ and } (y, z) \in S.$$

With $S \circ R$ we apply R first, then S. Another convenient notation is to write $R; S$. By definition, $R; S = S \circ R$. △

Remark 1.1 The definition of the composition of dyadic relations also applies, in particular, to the composition of functions. △

Definition 1.7: A function $f : A \longrightarrow B$ is called an *injection*, if and only if the following condition for f is true: for all elements x and y of A such that $x \neq y$: $f(x) \neq f(y)$. When a function is an injection, we also say that the function is *injective*, and we also say that the function is *one-to-one*. These phrases are equivalent ways of expressing the same thing. △

Note 1.2: The definition says that $f : A \longrightarrow B$ is injective if, and only if, for every two different inputs to f we must get two different outputs. Equivalently, we cannot get the same output from f from two different inputs. The condition for f to be injective can be

restated in the following equivalent form: for all elements x and y of A, if $f(x) = f(y)$, then $x = y$. △

Definition 1.8: A function $f : A \longrightarrow B$ is called a *surjection*, if and only if, the following condition for f is true: for each element b of B, there exists at least one $a \in A$ such that $f(a) = b$. A surjection is also said to be *onto*. △

Example 1.3: Let $\mathbf{R}$ be the set of real numbers. The function $g : \mathbf{R} \longrightarrow \mathbf{R}$ that is specified by $g(x) = x^2$ is not surjective and is not injective. The function $h : \mathbf{R} \longrightarrow \mathbf{R}$ that is specified by $h(x) = x^3 - x$ is surjective but not injective. The function $\exp : \mathbf{R} \longrightarrow \mathbf{R}$ specified by $\exp(x) = e^x$ is injective, but not surjective, and the function $f : \mathbf{R} \longrightarrow \mathbf{R}$ specified by $f(x) = x^3$ is both injective and surjective. △

Definition 1.9: Let $f : X \longrightarrow Y$. There are two functions associated with f that we will now define. The function $\hat{f} : \mathbf{P}(X) \longrightarrow \mathbf{P}(Y)$ is specified by

$$\hat{f}(A) = \{y \in Y \mid \text{for some } a \in A, y = f(a)\}.$$

$\hat{f}(A)$ is called the *f-image* of A or just the *image* of A when f is clear from the context.

The function $\hat{f}^{-1} : \mathbf{P}(Y) \longrightarrow \mathbf{P}(X)$ is specified by

$$\hat{f}^{-1}(B) = \{x \in X \mid f(x) \in B\}.$$

$\hat{f}^{-1}(A)$ is called the *f-inverse image* of A or just the *inverse image* of A when f is clear from the context. △

Note 1.3: Do not jump to conclusions. The function $\hat{f}^{-1}$ is not the *inverse* of f or even the inverse of $\hat{f}$.

With the following definition we begin to pick out certain special restricted kinds of mathematical entities that were defined above. It is important that you realize that, although the idea introduced next is new relative to the above definitions, the new idea *restricts* the preceding definitions. Restriction buys special properties useful in important contexts, but not necessarily in all contexts. You should organize your knowledge and understanding in such a way that you automatically see these kinds of logical relationships among the ideas you know about. △

Definition 1.10: A partial ordering on set A is a dyadic relation from A to A which is reflexive, anti-symmetric and transitive. (We typically use some such as symbol as $\sqsubseteq$ to denote an ordering relation, and write it in infix position. e.g. $x \sqsubseteq y$.) Specifically,

- (reflexive property) For every $a \in A$, $a \sqsubseteq a$.
- (anti-symmetric property) For every $a \in A$ and $b \in A$ if $a \sqsubseteq b$ and $b \sqsubseteq a$ then $a = b$.

- (transitive property) For every $a \in A$, $b \in A$ and $c \in A$, if $a \sqsubseteq b$ and $b \sqsubseteq c$ then $a \sqsubseteq c$.

A partial order is a pair $(A, \sqsubseteq)$ where A is a set and $\sqsubseteq$ is a partial ordering on A. A partial order is also called a *poset*. When the partial ordering relation is clear from the context we refer to a poset $(A, \sqsubseteq)$ by just mentioning A as in for example, "Let A be a poset such that ...". Often, we read expressions such as $x \sqsubseteq y$ as, "x is below y". △

Definition 1.11: *(Symmetric relation)* A dyadic relation R on a set A is *symmetric* iff for every $x \in A$ and every $y \in A$, if $(x, y) \in R$, then $(y, x) \in R$. △

Definition 1.12: *(Equivalence relation)* An *equivalence relation* on a set A is a dyadic relation on A is that is reflexive, symmetric and transitive. △

2 Some previous preliminary assessment exam questions with answers

Question 2.1: Let $f : X \longrightarrow Y$.

Part 1: Let $B \subseteq Y$ and suppose B is a subset of the f-image of A, where $A \subseteq X$. Must it be the case that there is a set $E \subseteq A$ such that B is the f-image of E? If it must be the case, prove it; otherwise give a counterexample.

There is a set $E \subseteq A$ such that B is the f-image of E. Consider letting $E = \hat{f}^{-1}(B) \cap A$. It remains to prove

$$\hat{f}(\hat{f}^{-1}(B) \cap A) = B$$

We will first prove

$$\hat{f}(\hat{f}^{-1}(B) \cap A) \subseteq B$$

by proving that every y in $\hat{f}(\hat{f}^{-1}(B) \cap A)$ is in B. Suppose y is any element in $\hat{f}(\hat{f}^{-1}(B) \cap A)$. Then there exists $x \in \hat{f}^{-1}(B) \cap A$ such that $y = f(x)$. Since $x \in \hat{f}^{-1}(B) \cap A$, $x \in \hat{f}^{-1}(B)$. Therefore, $y = f(x) \in B$. Thus, every y in $\hat{f}(\hat{f}^{-1}(B) \cap A)$ is in B.

We will now prove

$$B \subseteq \hat{f}(\hat{f}^{-1}(B) \cap A)$$

Suppose y is any element in B. Since $B \subseteq \hat{f}(A)$, $A \subseteq X$, $y \in \hat{f}(A)$. Therefore, there exists $x \in A$ such that $y = f(x)$. Since $y \in B$, $x \in \hat{f}^{-1}(B)$. Therefore $x \in (\hat{f}^{-1}(B) \cap A)$ and then $y = f(x) \in \hat{f}(\hat{f}^{-1}(B) \cap A)$.

Part 2: Let $B \subseteq Y$. Suppose $A \subseteq \hat{f}^{-1}(B)$. Must it be the case that there is a subset D of Y such that A is the f-preimage of D? If it must be the case, prove it; otherwise give a counterexample.

It does not have to be that there is a subset D of Y such that A is the f-preimage of D, if f is not injective. Here is a counterexample: Let $X = \{0, 1, 2\}$ and $Y = \{0, 1\}$. Let $f : X \longrightarrow Y$ be the following function: $f(0) = 0$, and $f(1) = f(2) = 1$. Note that f is not injective, but f is surjective. Let $B = \{1\}$. Then $\hat{f}^{-1}(B) = \{1, 2\}$. Let $A = \{1\} \subseteq \hat{f}^{-1}(B)$. There are four subsets of Y: $\emptyset$, $\{0\}$, $\{1\}$ and $\{1, 2\}$. A is not the f-preimage of any of them, since $\hat{f}^{-1}(\emptyset) = \emptyset$, $\hat{f}^{-1}(\{0\}) = \{0\}$, $\hat{f}^{-1}(\{1\}) = \{1, 2\}$ and $\hat{f}^{-1}(\{0, 1\}) = \{0, 1, 2\}$. $\triangle$

Question 2.2: Let $f : X \longrightarrow Y$.

Part 1: If the following statement is true, prove it; if false, give a counterexample: If A_1 and A_2 are any subsets of X, then

$$\hat{f}(A_1 \cap A_2) = \hat{f}(A_1) \cap \hat{f}(A_2)$$

It need not be true that $\hat{f}(A_1 \cap A_2) = \hat{f}(A_1) \cap \hat{f}(A_2)$. Let $X = \{0, 1\}$ and $Y = \{0\}$. Let $A_1 = \{0\}$ and $A_2 = \{1\}$. Let $f(0) = f(1) = 0$. Then, $\hat{f}(A_1) = \hat{f}(A_2) = \{0\}$. Thus, $\hat{f}(A_1) \cap \hat{f}(A_2) = \{0\}$, but $\hat{f}(A_1 \cap A_2) = \hat{f}(\emptyset) = \emptyset$. (Homework: Under what conditions on f can you guarantee that $\hat{f}(A_1 \cap A_2) = \hat{f}(A_1) \cap \hat{f}(A_2)$? Prove it.)

Part 2: If the following statement is true, prove it; if false, give a counterexample: If A_1 and A_2 are any subsets of X, then

$$\hat{f}(A_1 \cup A_2) = \hat{f}(A_1) \cup \hat{f}(A_2)$$

The equation must hold:

$$\begin{aligned} \hat{f}(A_1 \cup A_2) &= \{f(x) \mid x \in A_1 \cup A_2\} \\ &= \{f(x) \mid x \in A_1 \text{ or } x \in A_2\} \\ &= \{f(x) \mid x \in A_1\} \cup \{f(x) \mid x \in A_2\} \\ &= \hat{f}(A_1) \cup \hat{f}(A_2) \end{aligned}$$

$\triangle$

Question 2.3: Let $f : X \longrightarrow Y$.

Part 1: If the following statement is true, prove it; if false, give a counterexample: If B_1 and B_2 are any subsets of Y, then

$$\hat{f}^{-1}(B_1 \cap B_2) = \hat{f}^{-1}(B_1) \cap \hat{f}^{-1}(B_2)$$

It must be that $\hat{f}^{-1}(B_1 \cap B_2) = \hat{f}^{-1}(B_1) \cap \hat{f}^{-1}(B_2)$.

$$\begin{aligned} \hat{f}^{-1}(B_1 \cap B_2) &= \{x \in X \mid f(x) \in B_1 \cap B_2\} \\ &= \{x \in X \mid f(x) \in B_1 \text{ and } f(x) \in B_2\} \\ &= \{x \in X \mid f(x) \in B_1\} \cap \{x \in X \mid f(x) \in B_2\} \\ &= \hat{f}^{-1}(B_1) \cap \hat{f}^{-1}(B_2) \end{aligned}$$

Part 2: If the following statement is true, prove it; if false, give a counterexample: If A_1 and A_2 are any subsets of X, then

$$\hat{f}^{-1}(B_1 \cup B_2) = \hat{f}^{-1}(B_1) \cup \hat{f}^{-1}(B_2)$$

It must be that $\hat{f}^{-1}(B_1 \cup B_2) = \hat{f}^{-1}(B_1) \cup \hat{f}^{-1}(B_2)$.

$$\begin{aligned} \hat{f}^{-1}(B_1 \cup B_2) &= \{x \in X \mid f(x) \in B_1 \cup B_2\} \\ &= \{x \in X \mid f(x) \in B_1 \text{ or } f(x) \in B_2\} \\ &= \{x \in X \mid f(x) \in B_1\} \cup \{x \in X \mid f(x) \in B_2\} \\ &= \hat{f}^{-1}(B_1) \cup \hat{f}^{-1}(B_2) \end{aligned}$$

△

Question 2.4: Let X be a set with exactly 5 elements and let Y be a set with exactly 2 elements.

Part 1: How many functions are there from X to Y?

If S is a finite set, let $\|S\|$ denote the number of elements in S. A notation for the set of *all* functions from a set X to a set Y is Y^X. Then

$$\|Y^X\| = \|Y\|^{\|X\|}$$

In this question, $\|Y\|^{\|X\|} = 2^5 = 32$.

Part 2: How many functions are there from X to Y that are surjective?

If a function f from X to a set Y, where Y has just two elements, is not surjective, it must be constant; i.e. f must return the same output for every input. There are just two constant functions from X to a set Y with two elements. Therefore there are $2^{\|X\|} - 2$ surjective functions from X to Y; i.e. in this question, 30 such functions.

Part 3: How many functions are there from X to Z that are surjective, where Z is a set with exactly 3 elements?

If a function from X to Z is not surjective, it is either constant (if Z has exactly 3 elements, then there are 3 constant functions from X to Z) or its range has exactly 2 elements. There are three 2-element subsets of Z, and for each 2-element subset W of Z, there are, as we saw in Part (2), 30 functions whose range is W. Therefore there are $3 \cdot 30 + 3 = 93$ functions from X to Z that are not surjective. There are a total of $3^5 = 243$ functions from X to Z. So, there are $243 - 93 = 150$ surjective functions from X to Z. △

Remark 2.1: A current CIS 607 student sent an email to me that content concerning how many *surjective*, i.e. onto, functions there are from a set A with m elements to a set B with n elements.

> The question can be considered as partitioning A into n non-empty subsets and map all elements in each subset to one element in B, so we can get the answer $S(m, n) * n!$, where $S(m, n)$ is the *Stirling number of the second kind* [emphasis added].
>
> However, if we consider this question in another way, the answer will be different. We first randomly select n elements from A and map them "onto" B, and then map the remaining $m - n$ elements randomly to elements in B. In this way, the answer would be
>
> $$\frac{m!}{(m-n)!} * n^{(m-n)},$$
>
> where $\frac{m!}{(m-n)!}$ is the number of permutations. Could you please explain where the overcounting occurs?

There are good lessons to work through in this student's conundrum. First, if you read about Stirling numbers of the second kind (cf. Wikipedia for a basic discussion) you will see that these Stirling numbers do give the number of ways that finite sets can be partitioned into nonempty subsets. Thus, the statement in the student's first paragraph is correct. (Still,

a good exercise for self-study is to verify that the formula given in the Wikipedia article is correct.)

The second paragraph gives an incorrect formula that the student recognizes is incorrect, but does not see the failure point or points in the reasoning they [gender-neutral convention with the use of "they" here] used in deriving the formula they gave.

Here are two counterexamples that refute the given formula:

Example: Suppose $A = B$, an hence $m = n$. Then 0 occurs in a denominator in the formula. This counterexample to the formula unfortunately reveals almost nothing about any failure points in the reasoning that could have been used to derive it.

Example: Suppose $A = \{0, 1, 2\}$ and $B = \{0, 1\}$. The procedure that is presupposed for obtaining that surjective functions that the formula is alleged to count is

(a). Randomly select n elements from A. (By *randomly select* we mean select the n elements using a *uniform* probability distribution, i.e. the probability the event of selecting any particular set of n elements is the same is as the probability of selecting any other set of n elements.)

(b). Map these selected elements 1-to-1 in a random order onto B. (Again: by *random order* we mean the event consisting of a particular ordering is the same as the event consisting of any other ordering.)

(c). Map the remaining elements randomly into B.

Consider constructing a surjective function from A onto B using this procedure.

(a). Select $\{0, 1\}$.

(b). Let $f(0) = 0$ and $f(1) = 1$.

(c). Let $f(2) = 0$

Run the same random procedure over again:

(a). Select $\{1, 2\}$.

(b). Let $f(1) = 1$ and $f(2) = 0$.

(c). Let $f(0) = 0$.

Two rounds of the procedure made two different selections of the n-element subset of A in the first step, yet constructed the same function. The formula, to the extent that it reflects the procedure, would count that function twice.

To see the failure points in a purported argument, it helps to logically display every deductive step explicitly, **where each step is made trivially obvious**.

Question 2.5: Let $\mathbb{N}$ be the set of non-negative integers. For each non-negative integer n, let $f_n : \mathbb{N} \longrightarrow \{0, 1\}$. Let $g : \mathbb{N} \longrightarrow \{0, 1\}$ be defined by

$$g(k) = \begin{cases} 0 & \text{if } f_k(k) = 1 \\ 1 & \text{if } f_k(k) = 0 \end{cases}$$

Prove that for every $n \in \mathbb{N}$, $g \neq f_n$.

[**Note:** By definition, two functions f and g with the same domain and codomain are equal if, and only if, for every input x, $f(x) = g(x)$.]

Suppose the conclusion is false; i.e. for some k, $g = f_k$. Then, for every non-negative integer m,

$$g(m) = f_k(m)$$

But, by definition of g, if $f_k(k) = 0$, then $g(k) = 1$. And if $f_k(k) = 1$, then $g(k) = 0$. So,

$$g(k) \neq f_k(k)$$

Contradiction. Therefore, there is no k such that $g = f_k$. $\triangle$

Problem 2.1: First, a **definition:** A binary relation R on a set A is *euclidean* if for all x, y and z in A

$$R(x, y) \wedge R(x, z) \Rightarrow R(y, z)$$

Part 1) Is every binary relation on a set W that is reflexive and Euclidean also transitive and Euclidean? Prove or disprove. A disproof requires an explicit counter-example.

Part 2) Is a transitive, Euclidean binary relation also reflexive? Prove or disprove. A disproof requires an explicit counter-example.

Problem 2.2: (In-class Exercise)

Expanded version:

Let R be a binary relation on a set S. The *field* of R is the set

$$\{x \in S \mid \exists y \in S[(x, y) \in R\} \cup \{x \in S \mid \exists y \in S[(y, x) \in R\}$$

R is said to be *reflexive relative to a* a subset A of S if, and only if, for every $w \in A$, $(w, w) \in R$.

R is Euclidean if, and only if, for every x, y, $z \in S$, if $(x, y) \in R$ and $(x, z) \in R$, then $(y, z) \in R$.

(a). Let R be a binary relation on the set of three distinct elements listed: $\{x, y, u\}$. Let R be given by the set of pairs: $\{(x, y), (y, x), (x, x), (y, y), (u, x), (u, y)\}$. Is R transitive? Euclidean?

(b). Prove or disprove that every symmetric Euclidean relation is reflexive on its field.

(c). Prove or disprove that every symmetric transitive relation is reflexive on its field.

(d). Prove or disprove that every symmetric Euclidean relation is transitive.

(e). Prove or disprove that every symmetric transitive relation is Euclidean.

(f). Prove or disprove that every Euclidean relation is transitive.

(g). Prove or disprove that every transitive relation is Euclidean.

Problem 2.3: Some more definitions:

Definition: (*Intervals in relations*) Let R be a binary relation on a set A. Let a and b be elements of A. Then the *interval* $[a, b]$ *in* R is defined to be the set $\{x \in A \mid a\, R\, x \text{ and } x\, R\, b\}$.

Definition: (*Locally finite and past-finite relations*) A binary relation is *locally finite* if, and only if, every interval in the relation is finite. A binary relation R is *past-finite* if, and only if, for each $x \in A$ the set $\{a \in A \mid a\, R\, x\}$ is finite.

Do: Give an example of a locally finite set that is not past-finite.

Answer: Let m and n be integers and let $\leq$ be the ordinary less-then-or-equal-to relation on the set of all integers $\mathbb{Z}$. Then the interval $[m, n] = \{k \in \mathbb{Z} \mid m \leq k \leq n]$ is finite. Every interval in $\leq$ on $\mathbb{Z}$ is of the is form, so $\leq$ is locally finite. The set $\{k \in \mathbb{Z} \mid k \leq 0$ is infinite, so $\leq$ is not past finite.

Problem 2.4: (In-class Exercise)

Expanded version:

Let R be a binary relation on a set S. The *field* of R is the set

$$\{x \in S \mid \exists y \in S[(x, y) \in R\} \cup \{x \in S \mid \exists y \in S[(y, x) \in R\}$$

R is said to be *reflexive relative to a* a subset A of S if, and only if, for every $w \in A$, $(w, w) \in R$.

R is Euclidean if, and only if, for every x, y, $z \in S$, if $(x, y) \in R$ and $(x, z) \in R$, then $(y, z) \in R$.

(a). Let R be a binary relation on the set of three distinct elements listed: $\{x, y, u\}$. Let R be given by the set of pairs: $\{(x, y), (y, x), (x, x), (y, y), (u, x), (u, y)\}$. Is R transitive? Euclidean?

(b). Prove or disprove that every symmetric Euclidean relation is reflexive on its field.

(c). Prove or disprove that every symmetric transitive relation is reflexive on its field.

(d). Prove or disprove that every symmetric Euclidean relation is transitive.

(e). Prove or disprove that every symmetric transitive relation is Euclidean.

(f). Prove or disprove that every Euclidean relation is transitive.

(g). Prove or disprove that every transitive relation is Euclidean.

Problem 2.5: For this question, use the definition of *interval* given above.

Let R be a transitive relation on a set A. Let $[a_1, b_1]$ and $[a_2, b_2]$ be intervals in R. (They might or might not be the same interval, or it might or might not be that $a_1 = a_2$ etc. None of these possibilities are ruled out.)

In the following three parts, suppose the intersection $[a_1, b_1] \cap [a_2, b_2]$ of the intervals $[a_1, b_1]$ and $[a_2, b_2]$ is nonempty.

Part 1) Is $[a_1, b_1] \cap [a_2, b_2]$ an interval? Prove or disprove.

Part 2) Does $[a_1, b_1] \cap [a_2, b_2]$ have a nonempty subset that is an interval? Prove or disprove.

Part 3) If R is also reflexive, does $[a_1, b_1] \cap [a_2, b_2]$ have a nonempty subset that is an interval? Prove or disprove.

3 Propositional logic

- At *Amazon* if you search with

  Epstein Carnielli

  the first link returned will be to the book, *Computability* by Richard Epstein and Walter Carnielli. (cf. The course website in the Bibliography section.) The material

you will need for this section will be found in Chapter 19, but you do not need all of it. Alternatively, you can go into the online *Wikipedia* at

http://en.wikipedia.org/wiki/Propositional_logic (Propositional Logic)

There you will find a good comprehensive introduction to propositional logic. We do not need to study propositional logic in depth, although you should be aware that there is much more in it than you might at first suppose.

- You should also be aware that your experience in computer science has probably familiarized you already with much of what you need to know for present purposes about propositional logic and the main, classical, semantics associated with it: Boolean expressions built up purely from Boolean variables and Boolean connectives.
- The single most important idea to "get" is that formulas in logic (i.e. propositional formulas, or Boolean expressions, in the case of propositional formulas) do not actually mean anything - until a *semantics* is specified.
- When propositional logic is presented two main things need to be specified: the *syntax* and the *semantics*. The syntax is concerned with grammar: what is a formula of the logic, and what isn't. The semantics is concerned with what the formulas mean. To know something about logic you need to know about the grammar (syntax) of the logic, about the semantics of the logic, and how the logic relates to logic's big question: What follows from what?
- The concept of formal proof is on the syntax side of logic. If there is a proof of something, we would like to know that what's proved is really true, or at least is true if the premises (the "givens") are true. In other words, does what we manage to prove *really* follow from what is given? This issue is that of whether the logic is *sound*. One other main issue, usually more complicated, but less important, is whether the logic is *complete*, and if not, by how much does it miss the mark? The issue of completeness of a logic is this: If the logic can express some collection of premises, and you want to know whether something φ that the logic can express does follow from the premises, then - if φ follows - there is a proof in the logic that will show you that it follows. Soundness and completeness are about the relationship between the syntax side and semantic side of the logic.
- The first thing to do is to set up the syntax. You may want to examine the Wikipedia article on propositional logic mentioned above. Here is our set up: We start with an infinite set **Atoms**, which we call the set of *atoms*. Any member of this set - notice that we didn't actually specify what the set is - is called an *atom*. We also call a member of the set of atoms an *atomic formula*. Sometimes we call the members of the set of atoms, *Boolean variables*. We will use the terms *atom*, *atomic formula* and *Boolean variable* interchangeably. Now, we have the following *Boolean connectives*:

  (a). true

  (b). false

  (c). $\neg$

(d). $\rightarrow$

(e). $\leftrightarrow$

(f). $\wedge$

(g). $\vee$

(h). $|$

The first two in the above list are 0-ary connectives. The next one is 1-ary (also called unary). The next five are 2-ary (also called binary). We also call these connectives *propositional connectives*.

- The grammar rules for making formulas are:

  (a). `t` is a formula.

  (b). `f` is a formula.

  (c). If A is an atom, then A is a formula.

  (d). If A is a formula, then $\neg A$ is a formula.

  (e). If A and B are formulas, then $(A \rightarrow B)$ is a formula.

  (f). If A and B are formulas, then $(A \leftrightarrow B)$ is a formula.

  (g). If A and B are formulas, then $(A \wedge B)$ is a formula.

  (h). If A and B are formulas, then $(A \vee B)$ is a formula.

  (i). If A and B are formulas, then $(A \mid B)$ is a formula.

- If you are familiar with BNF-definitions, here is a similar, but more condensed, approach in BNF.

- Let Atom range over the set of atoms. Then

  Formula ::= `true`
  | `false`
  | Atom
  | $\neg$ Formula
  | (Formula $\circ$ Formula)
  | (Formula)

  $\circ$ ::= $\rightarrow$ | $\leftrightarrow$ | $\wedge$ | $\vee$ | $|$

- When we write logical formulas we will often omit parentheses when it is clear how to parse the formula.

- That's it for now with syntax.

- **Semantics:** Consider one of the easier, familiar propositional connectives, $\wedge$. What does $\wedge$ usually mean? Well, it usually means *and*. What does that mean? First of all, the symbol $\wedge$ joins together two propositional formulas to make a more complicated

one. So does $\vee$, so what is the difference? $A \wedge B$ is just a different formula from $A \vee B$, so that's the difference. But don't $\wedge$ and $\vee$ have something to do with Boolean values, and don't they get "defined by" truth tables? Yes, but what does all this cloud of vagueness mean? You might start to explain this business by saying that if you assign T to A and T to B then you must assign T to $A \wedge B$ and if you assign F either to A or to B then you must assign F to $A \wedge B$, and so on. OK, what do you mean by *assign*? And on and on and on. How are we going to make this stuff sharp and precise? Even more importantly, is the semantics we get from truth tables the right semantics? One might say that the semantics given by truth tables is by definition, so that's all end of the matter. But it's not.

- To start getting your mind out of this cloud of vagueness, go back and reconsider $\wedge$. That symbol *operates* on a pair of formulas. But the meaning of $\wedge$ operates on a pair of Boolean values, as you know if you've ever seen truth-tables. The symbol and the symbol's *meaning* are different things. The symbol is a syntactic constructor that makes a slightly more complicated formula out of two given formulas. The symbol's meaning returns a Boolean value, given a pair of Boolean-values as input.

- A fancy symbol for "the meaning of" is $[\![\ \ ]\!]$, as in $[\![\wedge]\!]$, which can be read as "the meaning of $\wedge$". Is this any better? After all, what's a meaning? At least we have some notation for distinguishing between $\wedge$ and what $\wedge$ means - and that's some progress. But we can do a lot better than that with this idea.

- **We will write an *interpreter* - we'll call it eval - to evaluate the truth-values of Boolean formulas relative to a *structure*.**

- One possible move that we could make at this point is to make $[\![\wedge]\!]$ an operator on pairs of Boolean values that returns a Boolean value. We could declare

$$[\![\wedge]\!] : \{F, T\} \times \{F, T\} \longrightarrow \{F, T\}$$

  which says that the meaning of $\wedge$ is a function which takes as input a pair of Boolean values (that's the $\{F, T\} \times \{F, T\}$) and returns a Boolean value (that's the $\longrightarrow \{F, T\}$ part).

- Now that we have declared the function $[\![\wedge]\!]$ we need to define it. This is mathematics, not a fixed programming language, so there are lot's of ways we can do that. The simplest one is perhaps the one you are familiar with: truth-tables. A truth-table is just an input/output lookup table for a function:

input		output
F	F	F
F	T	F
T	F	F
T	T	T

  This I/O table defines the function $[\![\wedge]\!]$ and tells you how the meaning of $\wedge$ behaves in the familiar way.

- The other propostional connectives are defined in exactly the same way. For example,

$$[\![\rightarrow]\!] : \{F, T\} \times \{F, T\} \longrightarrow \{F, T\}$$

where

input		output
F	F	T
F	T	T
T	F	F
T	T	T

- $[\![\rightarrow]\!]$ often seems confusing, or at least a bit forced. The third and fourth rows of the truth table seem fairly clear, but why should $p \rightarrow q$ be true, just because p is false. Is it really true that: if the moon is made of green cheese, then Donald Trump is a Liberal Democrat? Yes, according to the truth table, since the moon is not made of green cheese. (Wensleydale isn't green.) One way to convince yourself that the above truth table is correct is to ask how else could you define the meaning of $\rightarrow$. There are three other possibilities:

input		output
F	F	F
F	T	F
T	F	F
T	T	T

This one just above says that $[\![\rightarrow]\!]$ means just what $\wedge$ means.

input		output
F	F	F
F	T	T
T	F	F
T	T	T

This one just above says that $p \rightarrow q$ is true exactly when q is, and p doesn't matter.

input		output
F	F	T
F	T	F
T	F	F
T	T	T

And finally this one says that $p \rightarrow q$ is true exactly when p and q have the same Boolean value, in which case there is no difference between $p \rightarrow q$ and $q \rightarrow p$.

> "Once you eliminate the impossible, whatever remains, no matter how improbable, must be the truth."
>
> — Sherlock Holmes

- So it seems that all three of the other possibilities aren't correct. This argument might not seem convincing. A hidden premise behind the argument is that the Boolean value

of a compound proposition should be determined by the Boolean values of its parts, *and nothing else.* Consider the following argument:

> I did not strike the match at 9:00am. Therefore, if I had struck the match at 9:00am it would have lit.

- This is not a valid argument. What if the match was wet? Yet, by truth tables this seems to be a valid argument. There must be something wrong with the truth table representation of it. **Challenge problem (not part of the course):** What's wrong?

- What does the formula $p \wedge q$ mean? Remember, one of the big ideas is that formulas don't mean anything until a semantics is specified. Using our notation, the question we face is how to handle something like

$$[\![p \wedge q]\!]$$

In particular, $[\![p \wedge q]\!]$ should somehow help us with the question, is $p \wedge q$ true or false? Of course that depends on the Boolean values assigned to p and q. (Remember the question, what's an assignment?) Here's a way to deal with this.

Definition (Mathematical Logic Terminology): A *structure* for propositional logic is a function that takes an atom as input and returns a Boolean value.

Therefore a declaration for a structure ν has the form

$$\nu : \textbf{Atoms} \longrightarrow \{F, T\}$$

The following definition repeats the definition of *structure*.

Definition (Computer Science Terminology): An *environment* for propositional logic is a function that takes an atom as input and returns a Boolean value.

- Now, in order to press on we have to ask how we can describe the set of all structures. We need to have that description so that we can declare functions that take structures as inputs. Just for a moment consider two nonempty sets S_1 and S_2. The set of *all* functions from S_1 to S_2 is denoted by

$$S_1 \longrightarrow S_2$$

But what is this set? To get a grip on it in terms of S_1 and S_2 consider an example involving two small finite sets. Suppose $S_1 = \{0, 1\}$ and $S_2 = \{0, 1, 2\}$. An I/O table such as

input	output
0	2
1	1

defines a function from S_1 to S_2. So we can systematically list all of the functions from S_1 to S_2 using I/O tables. That is not the only way to describe the set $S_1 \to S_2$, but it's one way.

We get nine tables in all:

input	output
0	0
1	0

input	output
0	0
1	1

input	output
0	0
1	2

input	output
0	1
1	0

input	output
0	1
1	1

input	output
0	1
1	2

input	output
0	2
1	0

input	output
0	2
1	1

input	output
0	2
1	2

- We can condense all nine tables into one by listing the input column just once, and all output columns side by side.

input	output								
0	0	0	0	1	1	1	2	2	2
1	0	1	2	0	1	2	0	1	2

  This final table exhibits the whole set $S_1 \longrightarrow S_2$ in a single table.

  Notice also that if we delete the input column (since that doesn't change) in each of the nine individual tables, what we end up with is the set of all possible fully filled-in instances of an array indexed by S_1 whose cells are required to hold elements from S_2. The set of all such array instances is another way to describe $S_1 \longrightarrow S_2$.

- We have arrived at

$$\textbf{Atoms} \longrightarrow \{F, T\}$$

as the set of all structures for Propositional Logic - at least propositional logic with its usual, classical, semantics, with a fixed set **Atoms** as the set of atoms in its syntax. To make things a little more clear later on, we give this set a name: **Structures**. So

$$\textbf{Structures} \quad = \quad \textbf{Atoms} \longrightarrow \{F, T\}$$

- If we were to examine more deeply what we are doing with this last definition we would see that we were defining *states*. Exactly why that is would take us too far from what we are building right now, but keep in mind that propositional logic can't properly deal with state change. But there are logics, modal logics, that can deal with that in a strong sense.
- Now we can define the meanings of formulas.
- Intuitively, the Boolean value of $A \wedge B$ is fully determined by the Boolean values of A and B and the truth table for $[\![\wedge]\!]$. In turn, the Boolean values of A and B are determined by the parts of A and B and so on down to the atoms. The Boolean values of the atoms are given by some structure. So this gives us the way to define the meanings of all of the formulas recursively in a way that parallels the grammar that allows us to build up all of the formulas from atoms. The meaning $[\![\varphi]\!]$ of a formula φ is a function that takes a structure ν and returns a Boolean value.

$$[\![\varphi]\!] : \textbf{Structures} \longrightarrow \{F, T\}$$

We define $[\![\varphi]\!]$ recursively:

$[\![\texttt{false}]\!](\nu) = F$
$[\![\texttt{true}]\!](\nu) = T$
$[\![A]\!](\nu) = \nu(A)$, for each atom A in **Atoms**.
$[\![\neg A]\!](\nu) = [\![\neg]\!]([\![A]\!](\nu))$

and, for each binary propositional connective $\circ$,

$$[\![A \circ B]\!](\nu) = [\![\circ]\!]([\![A]\!](\nu), [\![B]\!](\nu))$$

- We can now define the evaluation function for propositional formulas relative to structures.

$$\mathsf{eval} : \ \textbf{Formulas} \times \textbf{Structures} \longrightarrow \{F, T\}$$

is defined by

$$\mathsf{eval}(A, \nu) = [\![A]\!](\nu)$$

- We could alternatively give a recursive definition of eval as follows.
- $\mathsf{eval}(\texttt{f}, \nu) = F$
  $\mathsf{eval}(\texttt{t}, \nu) = T$

$\mathsf{eval}(A, \nu) = \nu(A)$, for each atom A in **Atoms**
$\mathsf{eval}(\neg A,\ \nu) = [\![\neg]\!](\mathsf{eval}(A, \nu))$

and, for each binary propositional connective $\circ$,

$\mathsf{eval}(A \circ B,\ \nu) = [\![\circ]\!](\mathsf{eval}(A, \nu), \mathsf{eval}(B, \nu))$

- **Task:** Declare and define $[\![\neg]\!]$.
- **Task:** Using the definition just given, calculate

$$\mathsf{eval}(p \to (\neg q \mid (r \to (p \wedge r))),\ \nu)$$

where

$$\nu(p) = \nu(r) = T, \text{ and } \nu(q) = F$$

Answer:

$$\begin{array}{rl}
 & \mathsf{eval}(p \to (\neg q \mid (r \to (p \wedge r))),\ \nu) \\
= & [\![\to]\!](\mathsf{eval}(p, \nu), \mathsf{eval}(\neg q \mid (r \to (p \wedge r)), \nu)) \\
= & [\![\to]\!](T, [\![\mid]\!](\mathsf{eval}(\neg q, \nu), \mathsf{eval}(r \to (p \wedge r), \nu))) \\
= & [\![\to]\!](T, [\![\mid]\!]([\![\neg]\!](\mathsf{eval}(q, \nu)), [\![\to]\!](\mathsf{eval}(r, \nu), \mathsf{eval}(p \wedge r, \nu)))) \\
= & [\![\to]\!](T, [\![\mid]\!]([\![\neg]\!](F), [\![\to]\!](T, [\![\wedge]\!](\mathsf{eval}(p, \nu), \mathsf{eval}(r, \nu))))) \\
= & [\![\to]\!](T, [\![\mid]\!](T, [\![\to]\!](T, [\![\wedge]\!](T, T)))) \\
= & [\![\to]\!](T, [\![\mid]\!](T, [\![\to]\!](T, T))) \\
= & [\![\to]\!](T, [\![\mid]\!](T, T)) \\
= & [\![\to]\!](T, F) \\
= & F
\end{array}$$

By the way, we never defined $[\![\mid]\!]$. Here is its I/O table (i.e. truth table).

input		output
F	F	T
F	T	T
T	F	T
T	T	F

- **Definition:** A **finite sequence** of elements from a set S (or S-sequence) is a function of type

$$\{0, 1, 2, \ldots, n\} \longrightarrow S$$

Sometimes a finite sequence will denoted by expressions such as $x_0, x_1, \ldots, x_n$. The numerical subscripts are called *indices* [singular: *index*]. Also, sometimes the first index will be 1 instead of 0. Other variant notation will be clear from the context.

- **Example:** Does the conclusion follow from the premises? (Explain.)

p1. $B \to (L \to M)$
p2. $(M \wedge D) \to \neg G$
a3. $\neg J \to (D \wedge G)$
concl. $B \to (L \to J)$

Question 3.1:

Part 1: Let S be the set $\{a, b, c\}$. Assume a, b and c are all distinct, so that S has three elements. Here are five subsets of S: $\emptyset$, $\{a\}$, $\{c\}$, $\{a, c\}$ and $\{a, b, c\}$. There are three more. List these remaining three.

Answer:

$$\{b\},\ \{a,\ b\},\ \{b,\ c\}$$

Part 2: [The purpose of this question to see whether you understand definitions that use discrete math ideas, notations, and conventions.] Here is a definition.

Definition: A subset A of S is *basic* if, and only if, A is one of the following four sets: $\emptyset$, $\{a\}$, $\{c\}$ and $\{a, b, c\}$. The *interior* of any subset B of S is the union of all the basic subsets of B. The interior of B is denoted by $\mathsf{int}(B)$.

Find $\mathsf{int}(\{a, b\})$. Find $\mathsf{int}(\{c\})$.

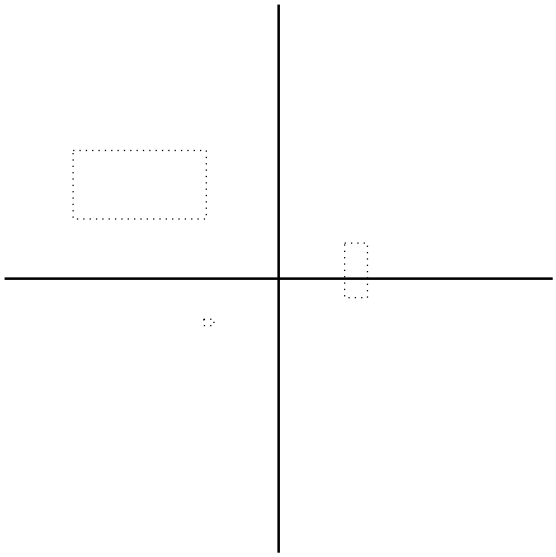
Answer: There are two subsets of $\{c\}$: $\emptyset$ and $\{c\}$. Both of them are basic are according to the definition of *basic*. The union of these two sets is $\{c\}$ since $\emptyset \cup \{c\} = \{c\}$. Therefore, $\mathsf{int}(\{c\}) = \{c\}$.

Question 3.2

Again, let S be the set $\{a, b, c\}$. Assume a, b and c are all distinct, so that S has three elements. Let B be any subset of S. The *complement* of B is the set $\{x \in S \mid x \notin B\}$. The complement of B is denoted by $\overline{B}$. The *closure* of B is the complement of the interior of the complement of B; i.e. $\overline{\mathsf{int}(\overline{B})}$. The closure of B is denoted by B^*. The *boundary* of B is the set $B^* \cap (\overline{B})^*$. The *boundary* of B is denoted by ∂B. Find $\partial\{a, b\}$. Indicate how you derived your answer.

Answer: $\partial\{a, b\} = \{a, b\}^* \cap (\overline{\{a, b\}})^*$. $\{a, b\}^* = \overline{\mathsf{int}(\overline{\{a, b\}})}$. $\overline{\{a, b\}} = \{c\}$. Therefore, $\mathsf{int}(\overline{\{a, b\}}) = \mathsf{int}\{c\} = \{c\}$, by part 2 of Problem 2. Therefore, $\{a, b\}^* = \{a, b\}$. $(\overline{\{a, b\}})^* = \{c\}^* = \mathsf{int}(\overline{\{c\}}) = \overline{\mathsf{int}\{a, b\}} = \overline{\emptyset \cup \{a\}} = \overline{\{a\}} = \{b, c\}$. Therefore, $\partial\{a, b\} = \{a, b\}^* \cap (\overline{\{a, b\}})^* = \{a, b\} \cap \{b, c\} = \{b\}$.

Note 3.1: Consider the coordinate Euclidean plane, which we might depict in the figure below.



The dotted rectangles in the above figure depict *open* filled rectangles aligned with the axes: the interior of the rectangle is part of the open filled rectangle, but points on the boundary of the rectangle are not part of the open filled rectangle. That's the *open* part. Formally, an *open interval* is a subset S of the set of real numbers such that for some real numbers a and b,

$$S = \{x \mid a < x < b\}\,.$$

If $a \geq b$, then S is empty. We consider the empty set to be an open interval.

(a). An open interval such as S is denoted by (a, b). (Yes, we are overloading the ordered pair notation, but we are computer scientists - overloading is a piece of cake.)

(b). An *open rectangle* is a subset of $\mathbf{R} \times \mathbf{R}$ of the form $(x_1, y_1) \times (x_2, y_2)$, where (x_1, y_1) and (x_2, y_2) are open intervals.

(c). Now, imagine all of the shapes in the coordinate Euclidean plane that you can build with an unlimited, infinite supply of open rectangles of every possible size but aligned with the axes: you can reuse and overlap them as you please.

(d). Think on this: you can build a disk without its boundary, but you cannot build a disk with its boundary, because any open filled rectangle that contains a point on the boundary must contain a point outside the disk.

(e). Every subset of the coordinate Euclidean plane (*plane* for short) that you can build with these rectangles is called an *open set* in the plane. By "build with" we mean that every open set is the union of all of the open rectangles in some collection of open filled rectangles.

(f). One more concept: Let S be a subset of the plane (maybe open, maybe not). The *interior* of S is the union of all of the open rectangles that are subsets of S. Notice that the interior of a set must be open. We denote the interior of S by $\operatorname{int} S$.

(g). To set up the semantics of intuitionistic logic, we take as our set of truth-values the

set of all open sets in the plane. We call this collection of all open sets $\mathcal{E}$. $\mathcal{E}$ is ordered by set inclusion (i.e. by $\subseteq$) and we take the join of two sets to be the union of the two sets, and the meet of two sets to be the intersection of the two sets, then $\mathcal{E}$, ordered by the set inclusion relation $\subseteq$, is a lattice. By the way, the intersection of two open sets and the union of any number, even infinitely many, of open sets is open.

(h). In the semantics for classical propositional logic we defined a structure to be a function in the set of functions

$$\textbf{Atoms} \longrightarrow \textbf{B}$$

where $\mathbf{B}$ is the lattice of Boolean values $\{F, T\}$. We didn't mention it at the time, but we typically consider the truth values to be ordered, with $F \leq T$.

(i). For the propositional connectives:

(i) $[\![\wedge]\!]$ is the meet operation in $\mathcal{E}$.

(ii) $[\![\vee]\!]$ is the join operation in $\mathcal{E}$.

(iii) $[\![\neg]\!]$ is the operation in $\mathcal{E}$ that maps an open set to the interior of its complement.

(iv) $[\![\rightarrow]\!]$ is defined by

$$[\![\rightarrow]\!](x, y) = \bigvee\{w \mid x \wedge w \leq y\}$$

i.e. $[\![\rightarrow]\!]$ is the least upper bound of the set $\{w \mid x \wedge w \leq y\}$.

(j). Another concept: In intuitionistic propositional logic, we say that a formula φ is *valid* in $\mathbf{R}^2$ iff

$$[\![\varphi]\!]\sigma = \mathbf{R}^2, \text{ for every } \mathbf{E}^2 \text{ Heyting algebra structure } \sigma\,.$$

One of the interesting things about $\mathbf{R}^2$ for this purpose is that there is a more general concept of validity for intuitionistic propositional logic that turns out to be equivalent after all to $\mathbf{R}^2$ validity. We will not go into Heyting algebras in any detail.

(k). The definition

$$[\![\rightarrow]\!](x, y) = \bigvee\{w \mid x \wedge w \leq y\}$$

is difficult to work with directly in the context of intuitionistic logic. It was no picnic with M_3, but at least M_3 was finite. To make this definition easier to work with, we will prove a theorem. The theorem gives an equivalent form of the definition that is easier to work with, but this equivalent form works only in lattices like $\mathcal{E}$. It wouldn't work with M_3.

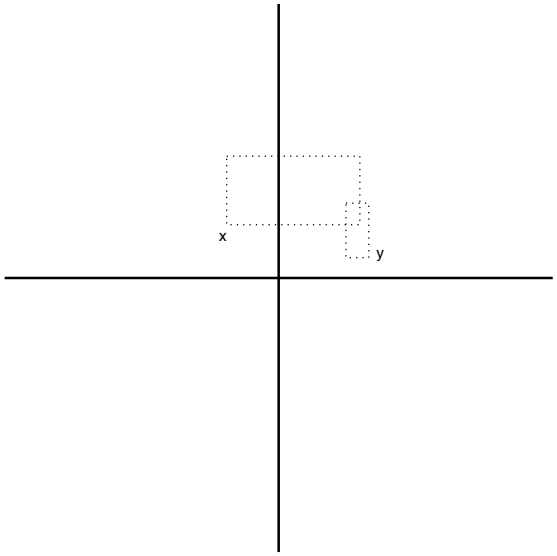
Theorem: In $\mathcal{E}$, (and in any topological Heyting algebra, of which $\mathcal{E}$ is an example)

$$\bigvee\{w \mid x \wedge w \leq y\} = \text{int}((\mathbf{R}^2 - x) \cup y)\,.$$

In more words, the theorem says that the left-hand side (the hard-to-work-with definition of $[\![\rightarrow]\!]$) is equal to the interior of the union of the complement of x with y.

The theorem makes it easy to find $[\![\rightarrow]\!](x, y)$ when x and y are both axis-aligned open filled rectangles.

(l). **Team Problem:** Consider the pair of rectangles in $\mathcal{E}$ in the figure below. On a copy of the figure, draw the set denoted by $[\![\rightarrow]\!](\mathrm{int}(\overline{x}), \mathrm{int}(\overline{y}))$. The complement of an open set u in the plane $\mathbf{R}^2$ is denoted by $\overline{u}$. (Use the theorem. We will prove the theorem in subsequent items.)



(m). **Team Problem:** Show that $A \vee \neg A$ is not valid in $\mathbf{R}^2$, by giving an explicit counterexample. (This shows that the *law of excluded middle* is not intuitionistically valid.) Show that $(A \rightarrow B) \rightarrow (\neg B \rightarrow \neg A)$ is valid in $\mathbf{R}^2$, but $(\neg A \rightarrow \neg B) \rightarrow (B \rightarrow A)$ is not valid in $\mathbf{R}^2$. You may find it easier to do this problem after doing each of the problems 22, 23 and 24, below.

(n). **Team Problem:** What if, in intuitionistic propositional logic, we defined $[\![\rightarrow]\!]$ by

$$[\![\rightarrow]\!](x, y) = ([\![\neg]\!](x)) \cup y$$

Is the definition equivalent to the original one. i.e. Is it always true that

$$\bigvee\{w \mid x \wedge w \leq y\} = ([\![\neg]\!](x)) \cup y\,?$$

(o). We will prove the theorem. You will do some of the steps in the **Team Problems**. Each side of the equation in the theorem is a set. We have to show that the sets are really the same set. One way to show that two sets S_1 and S_2 are the same set is to show that each set is a subset of the other. That is, $S_1 \subseteq S_2$ and $S_2 \subseteq S_1$. This strategy is the approach that we will take in proving this theorem. An alternative approach might be to develop a little calculus about interiors of sets. If we did that, then we could use the interiors-of-sets-calculus to give a slick calculational-looking proof, but it's not worth doing that since we won't put that calculus to further use. For the proof at hand we have to prove

$$\bigvee\{w \mid x \wedge w \leq y\} \subseteq \mathrm{int}(\overline{x} \cup y)\,. \tag{1}$$

and

$$\text{int}(\overline{x} \cup y) \subseteq \bigvee\{w \mid x \wedge w \leq y\}\,. \tag{2}$$

(p). We will prove (1) first. $\bigvee\{w \mid x \wedge w \leq y\}$ is an open set. Therefore, to prove (1), it suffices to prove (**Team Problem:** *Briefly* explain)

$$\bigvee\{w \mid x \wedge w \leq y\} \subseteq \overline{x} \cup y\,.$$

(q). Let $p \in \bigvee\{w \mid x \wedge w \leq y\}$. **Team Problem:** Why must it be the case that for some open w such that $x \cap w \subseteq y$, that $p \in w$? There are two cases: (1) $p \notin x$, and (2) $p \in x$. If $p \notin x$, then **Team Problem:** Prove: $p \in \overline{x} \cup y$.

(r). If $p \in x$, then, using the fact that $p \in w$, *Prove:* $p \in \overline{x} \cup y$.

(s). In the lattice of open subsets of the plane, for any two open subsets u, v of the plane, let

$$u \to v = \bigvee\{w \mid u \wedge w \leq v\}$$

We just overloaded the $\to$ symbol. (Again, for computer scientists and engineers, overloading is a piece of cake.) By our theorem,

$$u \to v = \text{int}(\overline{u} \cup v)$$

Now, let $A \to B$ be a formula of intuitionistic propositional logic, and let σ be an $\mathbf{E}^2$ Heyting algebra structure. (From now on we will call these structures, *intuitionistic structures*.) Let $u = [\![A]\!]\sigma$ and $v = [\![B]\!]\sigma$. Thus,

$$[\![A \to B]\!]\sigma = u \to v = \text{int}(\overline{u} \cup v)$$

(t). **Team Problem:** Analogously to the display just above,

$$[\![\neg B \to \neg A]\!]\sigma = ?$$

(u). **Theorem:** $A \to B$ is $\mathbf{R}^2$-valid iff for every intuitionistic structure σ,

$$[\![A]\!]\sigma \subseteq [\![B]\!]\sigma\,.$$

Proof: We need to show: for every intuitionistic structure σ, $[\![A \to B]\!]\sigma = \mathbf{R}^2$ iff for every intuitionistic structure σ, $[\![A]\!]\sigma \subseteq [\![B]\!]\sigma$. To show that it is sufficient to show that for every intuitionistic structure σ, $[[\![A \to B]\!]\sigma = \mathbf{R}^2$ iff $[\![A]\!]\sigma \subseteq [\![B]\!]\sigma]$.

Let σ be an arbitrary intuitionistic structure. Let $u = [\![A]\!]\sigma$ and $v = [\![B]\!]\sigma$. Suppose $A \to B$ is $\mathbf{R}^2$-valid. Then,

$$\mathbf{R}^2 = \text{int}(\overline{u} \cup v)$$

Therefore,

$$\mathbf{R}^2 = \overline{u} \cup v$$

Team Problem: Finish this half of the proof. All you have to do is prove that $u \subseteq v$.

(v). **Team Problem: The other half the proof of the previous theorem:** Suppose $u \subseteq v$. Then show that $\mathbf{R}^2 = \text{int}(\overline{u} \cup v)$.

CIS/CSE 607 PRACTICE Exam 1
October 19, 2017

Problem 1.

Part 1: Let $f : A \longrightarrow B$ and $g : B \longrightarrow C$. The function $h : A \longrightarrow C$ defined by $h(a) = g(f(a))$, for all $a \in A$, is denoted by $g \circ f$ and is called the *composition* of f and g. In this part, and part (2), a function f is a bijection if, and only if, f is an injection and a surjection, (In other words, f is one-to-one and onto.) Prove that if $g \circ f$ is a bijection then f is an injection (i.e. f is one-to-one) and g is a surjection (i.e. g is onto).

Answer: Suppose $g \circ f$ is a bijection. To show that f is a bijection, suppose $f(a_1) = f(a_2)$. It remains to show that $a_1 = a_2$. Since $f(a_1) = f(a_2)$, $(g \circ f)(a_1) = g(f(a_1)) = g(f(a_2)) = (g \circ f)(a_2)$. Since $g \circ f$ is a bijection, $g \circ f$ is 1-to-1. Therefore, since $(g \circ f)(a_1) = (g \circ f)(a_2)$, $a_1 = a_2$.

Part 2: Give an example that shows that if $g \circ f$ is a bijection, then it does not necessarily follow that f and g are bijections.

Answer: Let $A = C = \{0\}$ and $B = \{0, 1\}$. Let $f : A \longrightarrow B$ and $g : B \longrightarrow C$, where f is defined by $f(0) = 0$, and g is defined by $g(0) = g(1) = 0$. Then $g \circ f$ is a bijection, but f is not onto and g is not 1-to-1.

Problem 2.

Definition: (directed set). Let $A \subseteq D$ where D is a poset with partial ordering $\sqsubseteq$. A is *directed* in D iff A is nonempty and: for all $x \in A$ and for all $y \in A$ there exists $z \in A$ such that $x \sqsubseteq z$ and $y \sqsubseteq z$.

Notation: We denote the least upper bound of a set A (directed or not) by $\sqcup A$.

Let set $D = \{0, 1, 2, 3, \dots\}$. Give an example of a partial ordering on D such that D contains a subset that is directed but has no least upper bound. Explain.

Answer: D is the set of natural numbers $\mathbb{N}$. The natural ordering $\leq$ on $\mathbb{N}$ is a partial ordering on $\mathbb{N}$. $\mathbb{N}$ is a subset of $\mathbb{N}$ that is directed: for any two elements m and n of $\mathbb{N}$, there is an element k of $\mathbb{N}$ such that $m \leq k$ and $n \leq k$: we can choose for k any natural number greater than or equal to $\mathsf{max}(m, n)$. However, $\mathbb{N}$ has no least upper bound. In fact, $\mathbb{N}$ does not have *any* upper bounds.

Problem 3.

For each set A, the successor of A is the set $A \cup \{A\}$. The successor of A is denoted by A'. Let $S = \{a, \emptyset\}$. (Here, $\emptyset$ denotes the empty set, the set without any elements.) Determine S' and $(S')'$.

Answer:

$$\{a, \emptyset\}' = \{a, \emptyset\} \cup \{\{a, \emptyset\}\} = \{a, \emptyset, \{a, \emptyset\}\}\,.$$

$$\{a, \emptyset, \{a, \emptyset\}\}' = \{a, \emptyset, \{a, \emptyset\}\} \cup \{\{a, \emptyset, \{a, \emptyset\}\}\} = \{a, \emptyset, \{a, \emptyset\}, \{a, \emptyset, \{a, \emptyset\}\}\}\,.$$

Problem 4.

Consider:

> **Premise 1:** If Alice teleports her qubit, then if Bob doesn't power up his qubit receiver, then Bob doesn't receive Alice's qubit.
> **Premise 2:** Bob should not buy a new qubit receiver, if Bob doesn't receive Alice's qubit and Bob is angry.
> **Premise 3:** If Bob gets spied on, then Bob is angry and Bob should buy a new qubit receiver.
> **Conclusion:** If Alice teleports her qubit, then Bob doesn't receive Alice's qubit and Bob doesn't get spied on.

Let A stand for "Alice teleports her qubit." Let B stand for "Bob should buy a new qubit receiver." Let R stand for "Bob receives Alice's qubit." Let P stand for "Bob powers up his qubit receiver." Let S stand for "Bob gets spied on." Let G stand for "Bob is angry".

(1) Express the premisses and conclusion in our notation for propositional logic.

Answer:

$$\begin{aligned} &\text{premise1} : A \to (\neg P \to \neg R) \\ &\text{premise2} : (\neg R \wedge G) \to \neg B \\ &\text{premise3} : S \to (G \wedge B) \\ &\quad\text{conclusion} : A \to (\neg R \wedge \neg S) \end{aligned}$$

(2) Does the conclusion follow from the premises? (Explain.)

Answer:
No. By Gentzen's method (attempted falsification) we can make the premises true while keeping the conclusion false if we assign T to A, R and P, and F to S.

Problem 5.

Is the following formula intuitionistically $\mathbb{R}^2$-valid?

$$p \to \neg\neg p$$

Answer: Yes. We shall prove that $[\![p \to \neg\neg p]\!] = \mathbb{R}^2$, where $\mathbb{R}$ is the set of real numbers. To prove this we will first prove that

$$\mathbb{R}^2 \subseteq [\![p \to \neg\neg p]\!] = \mathsf{int}(\overline{[\![p]\!]} \cup [\![\neg\neg p]\!]) = \mathsf{int}(\overline{[\![p]\!]} \cup \mathsf{int}(\overline{\mathsf{int}\overline{[\![p]\!]}})) \subseteq \overline{[\![p]\!]} \cup \mathsf{int}(\overline{\mathsf{int}\overline{[\![p]\!]}})$$

i.e., we will prove

$$\mathbb{R}^2 \subseteq \overline{[\![p]\!]} \cup \mathsf{int}(\overline{\mathsf{int}\overline{[\![p]\!]}})$$

if we can prove this, we will have proven

$$\mathbb{R}^2 = \overline{[\![p]\!]} \cup \mathsf{int}(\overline{\mathsf{int}\overline{[\![p]\!]}})$$

But then, since $\mathbb{R}^2$ is open, and therefore is its own interior, we will have proven

$$\mathbb{R}^2 \subseteq \mathsf{int}(\overline{[\![p]\!]} \cup \mathsf{int}(\overline{\mathsf{int}\overline{[\![p]\!]}}))$$

Let x be an arbitrary element of $\mathbb{R}^2$. $x \in [\![p]\!]$ or $x \notin [\![p]\!]$. If $x \notin [\![p]\!]$, then $x \in \overline{[\![p]\!]}$. Hence $x \in \overline{[\![p]\!]} \cup \mathsf{int}(\overline{\mathsf{int}\overline{[\![p]\!]}})$. But, if $x \in [\![p]\!]$, then $x \notin \overline{[\![p]\!]}$. Hence $x \notin \mathsf{int}\overline{[\![p]\!]}$. Therefore, $x \in \overline{\mathsf{int}\overline{[\![p]\!]}}$. Since x is an arbitrary element of $[\![p]\!]$, we have $[\![p]\!] \subseteq \overline{\mathsf{int}\overline{[\![p]\!]}}$. Since, by definition of intuitionistic semantics, $[\![p]\!]$ is open, $x \in [\![p]\!] \subseteq \mathsf{int}(\overline{\mathsf{int}\overline{[\![p]\!]}}) \subseteq \overline{[\![p]\!]} \cup \mathsf{int}(\overline{\mathsf{int}\overline{[\![p]\!]}})$.

Problem 6.

Let G be the set of all permutations of a nonempty set A. (Every member of G is a one-to-one and onto function from A to A.) Let id be the identity function from A to A and let k be a fixed permutation of A such that $k \circ k = \mathsf{id}$. Every permutation f of A has an inverse f^{-1}. Thus, $f^{-1} \circ f = f \circ f^{-1} = \mathsf{id}$. Let $K = \{\mathsf{id}, k\}$. Let the relation $\sim$ on G be defined by

$$f \sim g \quad \text{iff} \quad f \circ g^{-1} \in K$$

Prove that $\sim$ is an equivalence relation.

Answer: Let f be an arbitrary permutation of nonempty set A. Then $f \circ f^{-1} = \mathsf{id} \in K$. Therefore, $f \sim f$. Since f is an arbitrary permutation of A, $\sim$ is reflexive.

Let f and g be arbitrary permutations of A such that $f \sim g$. Then $f \circ g^{-1} \in K$. Therefore, either $f \circ g^{-1} = \mathsf{id}$ or $f \circ g^{-1} = k$. If $f \circ g^{-1} = \mathsf{id}$, then $g \circ f^{-1} = \mathsf{id}^{-1} = \mathsf{id} \in K$. If $f \circ g^{-1} = k$, then $g \circ f^{-1} = k^{-1} = k \in K$. Therefore, $g \sim f$. Since f and g are arbitrary permutations of A such that $f \sim g$, $\sim$ is symmetric.

Suppose f, g and h are arbitrary permutations of A such that $f \sim g$ and $g \sim h$. Then $f \circ g^{-1} \in K$ and $g \circ h^{-1} \in K$. The composition operation on K is given by

$$\begin{array}{rcl} \mathsf{id} \circ \mathsf{id} & = & \mathsf{id} \\ \mathsf{id} \circ k & = & k \\ k \circ \mathsf{id} & = & k \\ k \circ k & = & \mathsf{id} \end{array}$$

Now, it follows that

$$f \circ h^{-1} = (f \circ g^{-1}) \circ (g \circ h^{-1}) \in K$$

Therefore, $f \sim h$. Since f, g and h are arbitrary permutations of A such that $f \sim g$ and $g \sim h$, $\sim$ is transitive.

Problem 7.

There is a directed graph called N_5. N_5 has five vertices. The set of vertices V of N_5 is $\{\bot, a, u, v, \top\}$. The edge relation of N_5 is

$$\{(\bot,\bot),(\bot,a),(\bot,u),(\bot v),(\bot,\top),(a,a),(a,\top),(u,u),(u,v),(u,\top),(v,v),(v,\top),(\top,\top)\}$$

N_5 is a partial order. (You don't have to prove it.) Denote the edge relation by $\leq$. $(V,\leq)$ has the following property:

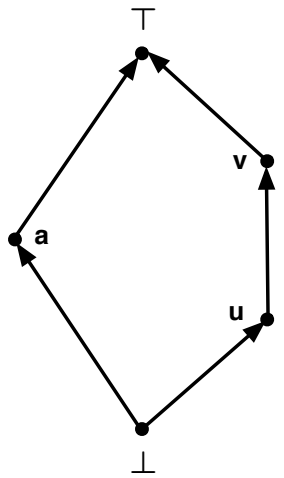
> (L) For every x and y in V, the set $\{w \mid w = x \text{ or } w = y\}$ has a least upper bound and a greatest lower bound.

(You don't have to prove that, either.) A partial order with property (L) is called a *lattice*. So, N_5 is a lattice. For every x, y in a lattice, the least upper bound of $\{w \mid w = x \text{ or } w = y\}$ is denoted by $x \vee y$, and the greatest lower bound of $\{w \mid w = x \text{ or } w = y\}$ is denoted by $x \wedge y$. A lattice is *distributive* if and only if, for every x, y and z that are elements of L,

$$\begin{aligned} x \vee (y \wedge z) &= (x \vee y) \wedge (x \vee z) \\ x \wedge (y \vee z) &= (x \wedge y) \vee (x \wedge z) \end{aligned}$$

Show that N_5 is **not** distributive.

Answer: Just an aid, here is a Hasse diagram for N_5:



If N_5 were distributive, then

$$v = v \wedge \top = v \wedge (a \vee u) = (v \wedge a) \vee (v \wedge u) = \bot \vee u = u$$

but $v \neq u$.

Problem 8.

Part 1: Let $f : X \longrightarrow Y$ and let $A \subseteq X$. Prove that

$$A \subseteq \hat{f}^{-1}(\hat{f}(A)) .$$

For this question, refer to the Workbook for an explanation of the notation. Pay attention to that little symbol $\hat{}$ that appears over some of the occurrences the f symbol here.

Answer:
Let x be an arbitrary element of A. Then $f(a) \in \hat{f}(A)$. Therefore, $a \in \hat{f}^{-1}(\hat{f}(A))$. Since x is an arbitrary element of A, $A \subseteq \hat{f}^{-1}(\hat{f}(A))$.

Part 2: Give an example of a function $f : X \longrightarrow Y$ and a subset A of X such that

$$A \neq \hat{f}^{-1}(\hat{f}(A)) .$$

Again, refer to the Workbook for an explanation of the notation.

Answer:
Let $X = Y = \{0, 1\}$ and let $f : X \longrightarrow Y$ by $f(0) = f(1) = 0$. Let $A = \{0\}$. Then

$$\hat{f}^{-1}(\hat{f}(A)) = \hat{f}^{-1}(\hat{f}(\{0\})) = \hat{f}^{-1}(\{0\}) = \{0, 1\} \neq A$$

Problem 9.

Is the following statement true for all antisymmetric relations?

The composition of two antisymmetric relations is antisymmetric.

Support your answer by giving either a rigorous proof of its validity (if you answer "yes") or a convincing/explained counterexample (if you answer "no").

Answer:
The composition of two antisymmetric relations does not have to be antisymmetric. Let $A = \{0, 1, 2, 3\}$. Let $R = \{(0, 1),\ (2, 3)\}$ and let $S = \{(1, 2),\ (3, 0)\}$. Then $S \circ R = \{(0, 2),\ (2, 0)\}$. R and S are anti-symmetric but $S \circ R$ is not.

- **First Order Logic.** By the term *classical logic* we intuitively mean a logic with the usual two familiar Boolean values, $\{F, T\}$, in which the propositional connectives have their usual meaning that can be defined via truth-tables. Informally, *first order logic* is a classical logic extended with the quantifiers $\forall$ and $\exists$ and where atoms have internal structure that allow us to refer to relationships among so-called *individuals*. So our problem now is to get technical and make all of this stuff precise.

- **The variables.** We begin by choosing an infinite (countably infinite, but the cardinality really doesn't matter) set of basic symbols, called *variables.* (These are not *Boolean variables*. There just called variables. If we want to emphasize the distinction between these variables and Boolean variables, we can call these variables *individual variables* because these variables are bound to values. Usually we don't bother with saying or writing *individual*. We will typically use symbols such as x, y, z, w, u and v, with and without subscripts, to designate variables.

- **The function symbols of a first-order language.** To set up a first order language $\mathcal{L}$ we first declare the *constant symbols* and *function symbols*: We choose a set FS_0. The members of this set are called *constant symbols*. Then for each positive integer $n > 0$, we choose a set FS_n. The members of each set FS_n are called *function symbols*. The sets FS_0, FS_1, FS_2, ... may each be empty, finite and nonempty, or infinite. They are pairwise disjoint. No constant or function symbols is a member of more than one of these sets. For each n, the *arity* of the symbols in FS_n is n. Constant symbols are function symbols with arity 0.

- **Example: The function symbols of the Language of Arithmetic.**

  Function symbols

arity 0:	`0`
arity 1:	`s`
arity 2:	`+`, `*`
arity n $>$ 2:	$\mathrm{FS}_n = \emptyset$

- **Terms.**

  Term ::= **variable**
  | **constantSymbol**
  | $\mathrm{FS}_n \mathbf{t}_1 \cdots \mathbf{t}_n$

- **Example: A term of the language of arithmetic.**

  ```
  *sss0+ss0y
  ```

  Syntax "sugar":

  ```
  3 * (2 + y)
  ```

- **Predicate symbols.** The 0-ary predicate symbols are true and false. We met these symbols before as 0-ary propositional connectives. Their true identity could only properly be revealed with predicate symbols. The third predicate symbol about which we have no choice is the symbol, =. It is an arity 2 predicate symbol. All of the remaining predicate symbols are up to us to declare.

- **The Language of Arithmetic.**

  Function symbols

arity 0:	: 0
arity 1:	s
arity 2:	+, *

  Predicate symbols:

arity 0:	true, false
arity 2:	=

  You might think we ought have constants for each number such as 1 for 1, and predicate symbols such as $<$, but we don't need them. 1 and 2 can for example be regarded as macros that expand to s(0) and s(s(0)), respectively. And so on. $<$ can be regarded as a macro also, but the way to expand it awaits the next two ideas.

- **Atomic formulas**; also called **atoms.** The definition of what a term is doesn't need to be revised. The key thing to remember about terms is that **predicate symbols never occur in terms**. Now suppose, p is a k-ary predicate symbol, where $k \geq 0$, and $t_1, \ldots, t_n$ are terms. Then $pt_1 \ldots t_n$ is an *atomic formula*. As usual, we allow ourselves to apply macros, cut up, reassemble and decorate with graffiti our syntactic constructions to help make them more readable. (These modifications are also known as *syntactic sugar* because reading the expressions is "sweetened". So we would generally write $pt_1 \ldots t_n$ as $p(t_1, \ldots, t_n)$. More concretely, a formula such as $= +xyss0$ could be written as $= (+xy, ss0)$, or as $+xy = s(s(0))$, or as $x + y = 2$.

- **The formulas of a first-order language.** First, recall the grammar for the formulas of propositional logic:

  Formula ::= `true`
  | `false`
  | Atom
  | ¬ Formula
  | (Formula ∘ Formula)

  ∘ ::= → | ↔ | ∧ | ∨ | |

  We modify the rules of this grammar to obtain the general form of the grammar for the formulas of a language $\mathcal{L}$ for first-order logic. The grammar that we present below already assumes that the set of atoms of $\mathcal{L}$, $\mathbf{Atoms}_{\mathcal{L}}$ (and therefore the set of terms $\mathbf{Terms}_{\mathcal{L}}$) have been defined. We also assume that **Var** is the set of variables. Then

  Formula ::= $\mathbf{Atoms}_{\mathcal{L}}$
  | ¬ Formula

| (Formula ∘ Formula)
| ∀ **Var** Formula
∘ ::= → | ↔ | ∧ | ∨ | |

- **Examples of formulas of the language of arithmetic.**

$$(x > 1) \vee \forall y\,[\exists z\,[y * z = x] \to (y = 1 \vee y = x)]$$

This is a formula from the language of arithmetic. To use the grammar to see that is such a formula, we proceed recursively: we first observe that there is syntactic sugar present because parentheses have been omitted. If we put them back in, we obtain

$$((x > 1) \vee \forall y\,[\exists z\,[y * z = x] \to (y = 1 \vee y = x)])$$

This is an instance of the template

$$(\text{Formula} \circ \text{Formula})$$

Next we need to verify that $(x > 1)$ is an instance of Formula. Again there is syntactic sugar present: the parentheses surrounding $x > 1$. Erasing these parentheses we are left with verifying that $x > 1$ is a formula of the language of arithmetic. $>$ is a macro in the sense that a formula $t_1 > t_2$ expands to $t_2 < t_1$. In turn, $<$ is a macro and $t_2 < t_1$ expands, by definition, to $\exists y\,[t_2 + (y + 1) = t_1]$, where y is a newly chosen variable. The symbol $\exists$ is also a macro and a syntactic expression such as $\exists y\,\varphi$ expands to $\neg\forall y\,\neg\varphi$. 1 of course expands to $\mathsf{s0}$. Thus, after macro expansion, $x > 1$ becomes

$$\neg\forall y\,\neg[\mathsf{s}(\mathsf{0}) + (y + \mathsf{s}(\mathsf{0})) = x]$$

where $\circ$ is instantiated to $\vee$. And so on.

- **Task:** Re-express

$$(1 < x) \wedge \forall y\,[\exists z\,[y * z = x] \to (y = 1 \vee y = x)]$$

using Polish (i.e prefix, parentheses-free notation) notation.

- Hereafter, we will freely use syntactic sugar.
- **Semantics.** What do these formulas mean? To understand the semantics of the formulas of first-order logic we have to understand what a *relational structure* (when the context clear we just call such a thing a *structure*) is. We will now describe how to set up a structure for a language for first-order logic. We are going to build such a structure. We'll give it a name: $\mathfrak{A}$. This structure will be a structure for a language $\mathcal{L}$.
- Part of the structure that we are building is an *algebra*. (You know what algebra is as a subject that you study in school. But if you ever wanted to know what an *algebra* is, this is it: you about to find out.) We choose a nonempty set. This is set is called the

universe of the structure we're building. We denote the universe of $\mathfrak{A}$ by $|\mathfrak{A}|$. When mathematicians, logicians and scientists are just dealing with the algebra part of $\mathfrak{A}$, they speak of the *carrier* of $\mathfrak{A}$. So *carrier* means the same thing as *universe* in the context of algebras. To simplify our usage of these terms we shall hereafter exclusively use the term *universe*.

- To fully set up an algebra $\mathfrak{A}_0$ for the constant and function symbols of a language $\mathcal{L}$ for FOL, we have to interpret each of these symbols on a nonempty set that will serve as the universe of the algebra. For a constant or function symbol σ we will denote the interpretation of σ by $[\![\sigma]\!]$. The *type* of $[\![\sigma]\!]$ - in a programming context, the type of $[\![\sigma]\!]$ is the meaning of a declaration. The type of $[\![\sigma]\!]$ is determined by the arity of σ - in setting up the algebra we have no choice about the type of σ. The type of $[\![\sigma]\!]$ is given by (in "mathematics notation")

$$[\![\sigma]\!] : \underbrace{|\mathfrak{A}| \times \ldots \times |\mathfrak{A}|}_{n} \longrightarrow |\mathfrak{A}| \tag{3}$$

  The corresponding declaration in a programming context looks like

$$\mathfrak{A}\, \sigma(\mathfrak{A}x_1, \ldots, \mathfrak{A}x_n);$$

  The meaning of $\mathfrak{A}\, \sigma(\mathfrak{A}x_1, \ldots, \mathfrak{A}x_n)$; which is denoted by

$$[\![\mathfrak{A}\, \sigma(\mathfrak{A}\, x_1, \ldots, \mathfrak{A}\, x_n);]\!]$$

  The meaning of $\mathfrak{A}\, \sigma(\mathfrak{A}x_1, \ldots, \mathfrak{A}x_n)$; could be further denoted by

$$|\mathfrak{A}|\ [\![\sigma]\!]\ (\underbrace{|\mathfrak{A}|\ \ldots, |\mathfrak{A}|}_{n}); \tag{4}$$

  Note the exact way to translate between the two ways [displays (3) and (4) just above] of expressing the type of $[\![\sigma]\!]$: "mathematical notation" and "C-style programming notation".

- As we said above, we have no choice in determining the type of each constant and function symbol. By the way, the type of a constant symbol `e` is best rendered in "C-style programming notation": $|\mathfrak{A}|\,[\![\texttt{e}]\!]();$.

- The set of all functions from a set A to a set B is denoted by

$$A \longrightarrow B$$

  When we write

$$f : A \longrightarrow B$$

  we are stating the type of function f. But, the notation for the type of f says that function f is a member of the set of all functions $A \longrightarrow B$. We could just as well have written

$$f \in (A \longrightarrow B)$$

When we fully specify a function f from set A to set B, we have to not only give the type of f, we have to choose f from set $A \longrightarrow B$. That doesn't mean we have to give a rule for somehow calculating f. So, when we are setting up an algebra, when we interpret an n-ary function symbol σ by determining the meaning of $[\![\sigma]\!]$, we just to have to choose a function from the set of functions

$$\underbrace{|\mathfrak{A}| \times \ldots \times |\mathfrak{A}|}_{n} \longrightarrow |\mathfrak{A}|$$

That doesn't mean we have to write down a rule for the one we choose. We only go that far if we want to *explicitly describe* an algebra. By the way, if the universe of an algebra is infinite, and the first order language that the algebra interprets has any nonconstant function symbols, then it turns out that almost none of the algebras with that universe can be explicitly described.

- To complete the set up of an algebra there is one more issue to take note of: the equality symbol, $=$, needs an interpretation. Once again, there is no choice. The equality symbol is interpreted as the identity binary relation on $|\mathfrak{A}|$; i.e.

$$[\![=]\!] = \{(x, x) \mid x \in |\mathfrak{A}|\}$$

If you are confused about what has bee said here about the equality symbol's meaning, then just take it as a given that the symbol means what you probably think it means.

- We will now give two different algebras to interpret the algebraic part of the language of arithmetic. The first one is known as the *standard model of arithmetic*, and the second one is Raphael Robinson's *nonstandard* model of arithmetic.

Example 1: (*The standard model of arithmetic.*) Recall the algebraic part of the language of arithmetic:

Function symbols

	arity 0:	: `0`
	arity 1:	`s`
	arity 2:	`+`, `*`
Predicate symbols:		
	arity 2:	$=$

The standard model is often denoted by $\mathfrak{N}$ ("Gothic 'N' "). The universe of $\mathfrak{N}$ is the set of natural numbers $\mathbb{N}$ (which includes 0.) Thus

$$|\mathfrak{N}| = \mathbb{N}$$

The type of, for example, $[\![\texttt{+}]\!]$ is given by

$$[\![\texttt{+}]\!] = |\mathfrak{N}| \times |\mathfrak{N}| \longrightarrow |\mathfrak{N}|$$

As we said above, we have no choice about the types of the symbols when we give the types. We will *identify* the constant fuctions with the elements in their ranges. What

is it stake is that according to rules for interpreting the constant symbols, if we wanted the symbol `0` in the language of arithmetic to mean the natural number 0, we would have say that $[\![\texttt{0}]\!]$ is the function that takes no input and returns the natural number 0. We could perhaps use a pseudo C-style declaration and definition to describe this function:

```
unsigned int 0(){
  return 0;
}
```

In mathematical notation we could use an expressin form the λ-calculus (we talked about this in class, and it is a notation that you are familiar with if you know LISP or any LISP-like language, such as Scheme):

$$[\![\texttt{0}]\!] = \lambda n[0]$$

But, this is unnecessarily complicated for our purposes. Instead, it would be easier to just have the meaning of `0` be 0:

$$[\![\texttt{0}]\!] = 0$$

We can arrange for this by identifying $\lambda n[0]$ with 0. Again, you are familiar with these sorts of identifications, but you might not realize it. In matrix algebra, we often identify a 1×1 matrix $[a]$ with the number a in its entry.

The meaning of the successor symbol `s` is given by

$$[\![\texttt{s}]\!] = \lambda n[n+1]$$

We could also give the meaning of `s` by

$$[\![\texttt{s}]\!](n) = n + 1$$

The plus symbol on the left side of the equations in the above two displays is the symbol in the language of arithmetic - the object language. The plus symbol on the right side is our usual symbol for addition on the set of natural numbers. This symbol is a symbol in the language we are using to talk about the object language and to talk about mathematical considerations - the metalanguage. What counts as object language and as metalanguage depends on the context. A metalanguage is itself a perfectly good language, and if we wanted to talk about it in another context, it would be an object language in the other context.

For the remaining two function symbols, the meaning is given by

$$\begin{aligned} [\![\texttt{+}]\!] &= + \\ [\![\texttt{*}]\!] &= * \end{aligned}$$

Again, the symbols on the right side in the above two displays are the usual symbols for addition, and multiplication on the set of natural numbers.

Example 2: *Raphael Robinson's nonstandard model of arithmetic.* We'll use $\mathfrak{Q}$ to denote this structure.

$$|\mathfrak{Q}| = \mathbb{N} \cup \{\alpha, \beta\}$$

where α and β are not natural numbers and are two distinct objects. (What they are is irrelevant - it is only the relationship between them and the natural numbers given by how we will interpret `0`, `s`, `+` and `*` that matters).

$$[\![\mathtt{0}]\!] = 0$$

We will specify the interpretations of `s`, `+` and `*` with input/output tables. Let m and n be natural numbers. Then

$[\![\mathtt{s}]\!]$:

input	output
n	$n+1$
α	α
β	β

$[\![\mathtt{+}]\!]$:

input		output
m	n	$m+n$
m	α	β
m	β	α
α	n	α
α	α	β
α	β	α
β	n	β
β	α	β
β	β	α

In the following table, let m and n be nonzero, so that we can make 0 a special case.

$[\![\mathtt{*}]\!]$:

input		output
0	0	0
0	n	0
0	α	α
0	β	β
m	0	0
m	n	$m * n$
m	α	α
m	β	β
α	0	0
α	n	β
α	α	β
α	β	β
β	0	0
β	n	α
β	α	α
β	β	α

- The language of arithmetic, as we have presented it, has the disadvantage that there are no predicate symbols in it other than true, false and $=$. Because there are no other predicate symbols, the examples we give of interpretations of this language can't exhibit the difference between an algebra and a structure. To remedy this deficiency, we can add a predicate symbol to the language of arithmetic: $<$. It turns out that everything we can express about arithmetic with the language of arithmetic can be said without this predicate symbol, by treating atomic formulas that use it as macros. We will show how to do this below, but right now, we want to give a couple of examples that show how to extend the algebras for the algebraic part of the language of arithmetic to relational structures that interpret the newly added predicate symbol.

- **Example:** (*The standard model of arithmetic.*)

$$[\![<]\!] \quad = \quad <$$

The less-than symbol on the left is the predicate symbol in the language of arithmetic. The less-than symbol on the right denotes in the usual way the less-than relation on the set of natural numbers.

It is important for the next example to have firmly in mind that the less-than relation on the natural numbers involves a set of pairs of natural numbers. Specifically, the less-than relation $<$ on the set of natural numbers is a triple (not a set of triples)

$$(\mathbb{N}, \mathbb{N}, \{(i, j) \mid i + k = j \text{ for some nonzero natural number } k\})$$

The first two components of the triple are the domain and codomain of the relation, and the third argument is the *extension* of the relation. We usually informally identify the relation with just its extension when the domain and codomain are clear from the

context. Thus we would write (by overloading the $[\![\cdot]\!]$ notation):

$$[\![<]\!] = \{(i, j) \mid i + k = j \text{ for some nonzero natural number } k\})$$

- **Example:** (*Raphael Robinson's nonstandard model of arithmetic.*) Recall that the universe of this model has all of the natural numbers together with two extra individuals that we denoted by α and β. The task here is to extend the the standard less-than relation on the set of natural numbers to take account of α and β. One way to make such an extension would be have α and β not be comparable with each other nor with any natural number. But it turns out that a tremendously important property of Robinson's model would be lost if we did that. (We will look into that important property later.) What Robinson did was arrange for the extension of $[\![<]\!]$ to obey the following equation:

  $$[\![<]\!] = \{(x, y) \mid x + z = y \text{ for some nonzero individual } j \text{ in } \mathbb{N} \cup \{\alpha, \beta\}\}$$

  In other words,

  $$[\![<]\!] \;\; = \;\; < \;\cup\; \{(n, \alpha) \mid n \in \mathbb{N}\} \cup \{(n, \beta) \mid n \in \mathbb{N}\} \cup \{(\alpha, \beta)\} \cup \{(\beta, \alpha)\}$$

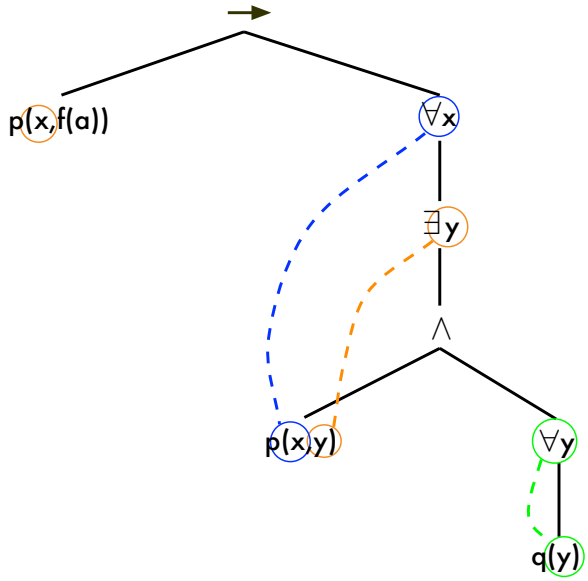
  Informally, the above equation implies that every natural number is less than α and less than β, and α and β are less than each other. So, the interpretation of $<$ in Robinson's nonstandard model violates antisymmetry.

- Now that we have some sense of how structures interpret the symbols of a language for first-order logic, we need to know how formulas are given Boolean values by a structure. To deal with that issue, we must first take up the notion of free and bound occurrences of variables. The easiest way to define the free and bound occurrences of variables in a formula is to consider the formula's phrase-structure tree in which the leaves are atomic formulas. Every variable occurrance terminates within an atomic formula; i.e. variables appear either paired with quantifiers or within atomic formulas.

- **Example:** Consider the formula

  $$\mathsf{p(x,f(a))} \rightarrow \forall \mathsf{x} \exists \mathsf{y}(\mathsf{p(x,y)} \wedge \forall \mathsf{y q(y)})$$

  Going from left to right, the first occurrence of x is free in the formula. The second occurrence of x is free in the subformula $\mathsf{p(x,y)}$, free in the subformula $(\mathsf{p(x,y)} \wedge \forall \mathsf{y q(y)})$, and free in the subformula $\exists \mathsf{y}(\mathsf{p(x,y)} \wedge \forall \mathsf{y q(y)})$. It is bound in the subformula $\forall \mathsf{x} \exists \mathsf{y}(\mathsf{p(x,y)} \wedge \forall \mathsf{y q(y)})$ by the only occurrence of $\forall \mathsf{x}$ and bound in the whole formula by the only occurrence of $\forall \mathsf{x}$. The occurrence of y in $\mathsf{p(x,y)}$ is free in $\mathsf{p(x,y)}$, free in $(\mathsf{p(x,y)} \wedge \forall \mathsf{y q(y)})$ and bound in $\forall \mathsf{x} \exists \mathsf{y}(\mathsf{p(x,y)} \wedge \forall \mathsf{y q(y)})$ by the only occurrence of the quantifier $\exists \mathsf{y}$. The occurrence of y in $\mathsf{q(y)}$ is free in $\mathsf{q(y)}$, bound in $\forall \mathsf{y q(y)}$ by the only occurrence of $\forall \mathsf{y}$, and similarly bound in all formulas containing $\forall \mathsf{y q(y)}$ as a subformula.



- **Problems of the Day**

  (a). Do the task in step 9, above.

  (b). What does the formula in step 9 say when interpreted in the standard model of arithmetic?

  (c). In Robinson's nonstandard model, do α and β have the property defined by the formula in step 9? Why?

- **Expanding a formula over a structure.** We present the steps for expanding a formula over a structure. Again, let $\mathcal{L}$ be an arbitrary first order language, and let $\mathfrak{A}$ be an arbitrary structure for $\mathcal{L}$. Suppose we are given formula A to expand.

  Step 1: (Preprocessing) Let $x_1, \ldots, x_k$ be the variables that have free occurrence in A. Replace A by its *universal closure* $\forall x_1 \ldots \forall x_n\, A$. Call the resulting formula A'. If A had no free occurrences of variables, then A' is just A. The order of the quantifiers introduced in this preprocessing step does not matter.

  Step 2: If A' is $\forall x\, B$, then replace A' by

$$\bigwedge_{i \in |\mathcal{A}|} \overline{B[x \mapsto i]}\,.$$

  where $B[x \mapsto i]$ is the formula that results from replacing all free occurrences of x in B by the individual i, and $\overline{B[x \mapsto i]}$ is the expansion of $B[x \mapsto i]$.

  Step 3: If A' is $\exists x\, B$, then replace A' by

$$\bigvee_{i \in |\mathcal{A}|} \overline{B[x \mapsto i]}\,.$$

  Step 4: If A' is a propositional combination of formulas, $B_1 \circ B_2$, then the expansion of A' is $\overline{B_1} \circ \overline{B_2}$, where $\overline{B_1}$ is the expansion of B_1 and $\overline{B_2}$ is the expansion of B_2.

Step 5: If A' is $\neg B$, then the expansion of A' is $\neg\overline{B}$, where $\overline{B}$ is the expansion of B.

- In effect we expanded the language $\mathcal{L}$ to a language $\mathcal{L}(\mathfrak{A})$ by putting in each individual i in $|\mathfrak{A}|$ as new constant symbol, and we will extend the structure $\mathfrak{A}$ to a structure for this newly expanded language by interpreting each individual as itself. Each variable-free term $\mathtt{t}$ of the expanded version of $\mathcal{L}$ then denotes an individual $[\![\mathtt{t}]\!]$ in $|\mathfrak{A}|$.

- **Definition:** A variable-free atomic formula $\mathtt{p(t}_1, \ldots, \mathtt{t}_n\mathtt{)}$ has Boolean value *True* in $\mathfrak{A}$ iff $([\![\mathtt{t}_1]\!], \ldots, \mathtt{t}_n) \in R$, where R is the n-ary relation on $|\mathfrak{A}|$ that is the interpretation of $\mathtt{p}$ in $\mathfrak{A}$. All other fully expanded formulas are (infinitarily) proposition and their truth values are obtained as in classical propositional logic. A formula A is *valid* in structure $\mathfrak{A}$ iff the expansion of A over $|\mathfrak{A}|$ has Boolean value *True* in $\mathfrak{A}$. We denote that A is valid in $\mathfrak{A}$ by
$$\mathfrak{A} \models A$$
A formula A is said to be *logically valid* iff A is valid in $\mathfrak{A}$ for every structure $\mathfrak{A}$ for every language $\mathcal{L}$ of which A is a formula. (In other words A is logically valid iff for every set of individuals, the expansion of A is a propositional tautology.) We denote that A is logically valid by
$$\models A$$
We also abbreviate the phrase *logically valid* to just *valid*.

- **Example Problem:** Show that the formula is
$$(\forall x\,[p(x,x) \to p(x,a)] \wedge \exists y\, p(y,y)) \to p(a,a)$$
not valid by giving a relational structure $\mathcal{A}$ in which the formula is false.

Answer: We guess that an interpretation with 2 individuals in its universe of discourse will provide an interpretation in which the formula evaluates to False. Call the two individuals 0 and 1, call the interpretation $\mathcal{A}$ and expand the formula over the set $\{0,1\}$.

$$\begin{aligned}
&(\forall x\,[p(x,x) \to p(x,a)] \wedge \exists y\, p(y,y)) \to p(a,a)\\
\equiv_{\mathcal{A}}\ &(\wedge_{x\in\{0,1\}}[p(x,x) \to p(x,a)] \wedge \vee_{y\in\{0,1\}} p(y,y)) \to p(a,a)\\
\equiv_{\mathcal{A}}\ &([[p(0,0) \to p(0,a)] \wedge [p(1,1) \to p(1,a)]] \wedge [p(0,0) \vee p(1,1)]) \to p(a,a)
\end{aligned}$$

We now try to falsify the above formula by assigning Boolean values to the atoms that occur in it. We see by Gentzen's method that if we interpret the constant symbol a as 0, and assign true to $p(1,1)$, $p(1,0)$ and $p(0,1)$, and assign false to $p(0,0)$, then the formula evaluates to false. This truth-value assignment tells us how to complete the specification of interpretation $\mathcal{A}$. The pairs in $\{0,1\} \times \{0,1\}$ that should be in the interpretation $p_{\mathcal{A}}$ of p are all and only the pairs (i,j) such that $p(i,j)$ evaluates to true. Thus,
$$p_{\mathcal{A}} = \{(0,1),(1,0),(1,1)\}\,.$$

- Sometimes, in order to falsify a formula, we are forced to use an infinite structure. Consider falsifying the formula

$$\forall x\, \forall y\, [f(x) = f(y) \rightarrow x = y] \rightarrow \forall y\, \exists x\, [f(x) = y]$$

A structure $\mathcal{A}$ that falsifies this formula must provide an interpretation for the function symbol f. We must have

$$\mathcal{A} \models \forall x\, \forall y\, [f(x) = f(y) \rightarrow x = y]$$

while falsifying

$$\forall y\, \exists x\, [f(x) = y]$$

The formula $\forall x\, \forall y\, [f(x) = f(y) \rightarrow x = y]$ asserts that f is a one-to-one function. The formula $\forall y\, \exists x\, [f(x) = y]$ asserts that f is onto. Since a 1-1 function from a *finite* set S to S that is one-to-one must also be onto, for it to be true that f is one-to-one but false that f is onto, it must be that $|\mathcal{A}|$ is not finite.

- **Example:** "There are exactly two individuals." Consider the formula **Two**:

$$\exists x\, \exists y\, [x \neq y \wedge \forall z\, [z = x \vee z = y]]$$

Then

$$\mathfrak{A} \models \textbf{Two} \text{ iff } \text{size}|\mathfrak{A}| = 2$$

- **Problem:** Give a formula **Three** analogous to **Two**.

- **Problem:** Give a relational structure in which the following formula does not evaluate to true:

$$\exists x \forall y\, [(p(a, b) \wedge \neg p(x, y)) \rightarrow p(y, x)]\,.$$

- Here is a simple language: it has three constant symbols $\mathsf{0}$,$\mathsf{1}$ and $\mathsf{2}$, one unary function symbol h and four predicate symbols true, false, q and $=$. We set up a structure $\mathcal{S}$ for this language as follows: Let the universe of the structure be the set $\{a, b, c\}$, where a, b and c are three distinct individuals.

- **Problem:** To finish setting up $\mathcal{S}$, assign interpretations to the symbols of the language so that all of the following hold: $\mathsf{0}$,$\mathsf{1}$ and $\mathsf{2}$ each have distinct interpretations, h is interpreted as a one-to-one monotonic function other than the identity function and q is a partial ordering with respect to which $[\![\mathsf{0}]\!]$ is the least element, but the partial ordering has no greatest element. [Note: A function $f : D \longrightarrow D$, where $(D, \sqsubseteq)$ is a partial ordering, is monotonic on $(D, \sqsubseteq)$ iff for all $x, y \in D$, if $x \sqsubseteq y$, then $f(x) \sqsubseteq f(y)$].

- **Problem:** Expand the following formula over the set $\{a, b, c\}$ and evaluate it on the structure you set up in the previous practice problem.

$$\exists x\, \forall y\, \neg \mathsf{q}(x, \mathsf{h}(x))$$

- **Definition of Prenex Form.** A formula is in *prenex form* iff it is closed and has the form

$$Q_1 x_1 \ \ldots \ Q_n x_n \, \varphi$$

where each Q_i is either $\forall$ or $\exists$, each x_i is a variable, and φ does not contain any quantifiers. The variables $x_1, \ldots, x_n$ must be pairwise distinct, i.e. no variable can appear more than once in the sequence. Every closed formula is equivalent to some formula that is in prenex form. $Q_1 x_1 \ \ldots \ Q_n x_n$ is called the quantifier *prefix* and φ is called the *matrix*.

- To find a formula in prenex form that is equivalent to a given closed formula Ψ we have to "factor out" the quantifiers in Ψ. To do that we use *equivalences from propositional logic* together with three rules for manipulating quantifiers:

(a). (α-*conversion*)

$$Qx \, \varphi \leftrightarrow Qy \, \varphi'$$

where φ' is obtained from φ by replacing each free occurrence of x in φ by y and φ can be recovered from φ' by replacing each free occurrence of y in φ' by x. (This version of the requirements for substituting variables must be satisfied is due to David Jakel.)

(b).

$$(\psi \rightarrow Qx \, \varphi) \leftrightarrow Qx \, (\psi \rightarrow \varphi)$$

provided x does not have a free occurrence in ψ.

(c).

$$Qx \, \neg\varphi \leftrightarrow \neg\overline{Q}x \, \varphi$$

where $\overline{\exists}$ is $\forall$ and $\overline{\forall}$ is $\exists$.

- **Problem:** Explain why an α-conversion cannot be performed on

$$\forall x \, [p(x) \vee \exists y \, q(x, y)]$$

where the variable y is substituted for the variable x.

- **Problem:** Find a formula in prenex form that is equivalent to

$$\forall y \forall x [p(x, y) \vee (\neg \exists y \, q(y) \rightarrow \forall x \, p(y, x))]$$

- **"Skolemizing".** Suppose that we have a formula in prenex form. When we Skolemize we seek to eliminate all of the existential quantifiers from the quantifier prefix. Each existentially quantified variable is replaced by a new function of all the preceding universally quantified variables in the prefix, and then existential quantifiers are removed.

- **Example of how to Skolemize.** Consider

$$\exists x \forall y \exists w \exists z \forall u \forall v \exists x' \Psi(x, y, w, z, u, v, x')$$

We replace x by a new function of all of the preceding universally quantified variables. In this case, there are no such variables, the new function is $0-ary$, i.e. a new constant added to the background language. In practice, we pick a new constant symbol not previously used in the example, theory, or proof we are working on.

We will choose a to obtain

$$\forall y \exists w \exists z \forall u \forall v \exists x' \Psi(a, y, w, z, u, v, x')$$

Next, we replace by w by a new function of y. Suppose that, for example, the function symbol f occurs in $\Psi(a, y, w, z, u, v, x')$ but that g does not occur. We can therefore use g as a unary (i.e. 1-ary) function symbol and replace w by $g(y)$ to obtain

$$\forall y \exists z \forall u \forall v \exists x' \Psi(a, y, g(y), z, u, v, x')$$

We next introduce another new unary function symbol, e.g. g', and replace z by $g'(y)$ to obtain

$$\forall y \forall u \forall v \exists x' \Psi(a, y, g(y), g'(y), u, v, x')$$

Lastly, we introduce a new 3-ary function symbol, e.g. h, and replace x' by $h(y, u, v)$ to obtain

$$\forall y \forall u \forall v \Psi(a, y, g(y), g'(y), u, v, h(y, u, v))$$

Here is the example redone more concretely, where for example, $\Psi(x, y, w, z, u, v, x')$ is the formula

$$p(x, f(w)) \rightarrow (q(y, w, u, v) \vee r(x', a, y, x))$$

Notice that z doesn't happen to actually occur in $\Psi(x, y, w, z, u, v, x')$ despite what the notation appears to indicate.

We are assuming that we are now starting with

$$\exists x \forall y \exists w \exists z \forall u \forall v \exists x' [p(x, f(w)) \rightarrow (q(y, w, u, v) \vee r(x', a, y, x))]$$

This time, since the constant symbol a already occurs in the formula we are working on, we start by replacing x with a new constant, e.g. b.

After completely Skolemizing, we obtain

$$\forall y \forall u \forall v [p(b, f(g(y))) \rightarrow (q(y, g(y), u, v) \vee r(h(y, u, v), a, y, b))]$$

- **Problem:** Verify the result of Skolemizing with which we concluded the immediately preceding example.

- **Problem:** Skolemize this:

$$\exists x \forall y \forall z \exists w [p(a, f(y, w)) \wedge q(b, x)]$$

Notice that z does not occur in the matrix to begin with, but be careful about what the Skolemizing procedure gives you.

- **Definition** (*theory*): A *theory* T is a pair consisting of a language L for FOL and a set of formulas Γ of L. L is called the *language of* T and is denoted by $L(T)$. Γ is called the set of *nonlogical axioms* of T and is denoted by $\mathrm{NlAx}(T)$.

- The language of theory called *Robinson arithmetic*, (denoted by Q) is the language of arithmetic. The nonlogical axioms of Q are:

Q1. $s(x_1) = s(x_2) \rightarrow x_1 = x_2$
Q2. $\neg(0 = s(x_1))$
Q3. $\neg(x_1 = 0) \rightarrow \exists x_2 \, (x_1 = s(x_2))$
Q4. $x_1 + 0 = x_1$
Q5. $x_1 + s(x_2) = s(x_1 + x_2)$
Q6. $x_1 * 0 = 0$
Q7. $x_1 * s(x_2) = (x_1 * x_2) + x_1$
Q8. $x_1 \leq x_2 \leftrightarrow \exists x_3 \, (x_1 + x_3 = x_2)$

- **Problem:** Show that Q6 and Q7 are valid in Robinson's nonstandard model.

- **Problem:** Show that $x + y = y + x$ is not valid in Robinson's nonstandard model.

- The reason we care about Robinson Arithmetic is that it turns out that in a sense that can be made rigorous and precise, the problem of computing a computable function, defined by code written in e.g. any programming language, can be made represented as a theorem proving task in Q. So, for example, any C-program with its input can be compiled to a single goal in the language of arithmetic that is to be computed by showing the resulting tableau to be valid in Q. Theory Q is a minimal theory with this property. In other words, all computation can be represented in Q, but this is not so for any theory strictly weaker than Q. This further shows that familiar laws of arithmetic such as the commutativity of addition are *not needed for computation*!

- Nevertheless, we would still like to have a theory of arithmetic in which we are able to prove familiar laws of arithmetic such as the fact that $\forall x \, [0 + x = x]$ - which cannot be proved in Q. This deficiency of Q can be remedied by Peano Arithmetic.

- *Unification*: We will proceed to illustrate unification by studying some examples.

- Find all solutions of the equation

$$k(f(x, u, a), f(g(y), y, v)) = k(f(g(y), y, v), f(w, h(z), z))$$

that hold regardless of the meaning of the constant and function symbols a, f, g, h and k. (In the preceding equation, a is a constant, and x, y, z, u, v and w are variables.)

Answer: $k(f(x,u,a),f(g(y),y,v)) = k(f(g(y),y,v),f(w,h(z),z))$

$f(x,u,a) = f(g(y),y,v)$
$f(g(y),y,v) = f(w,h(z),z)$

$x = g(y)$
$u = y$
$a = v$
$g(y) = w$
$y = h(z)$
$v = z$

$x = g(h(z))$
$u = h(z)$
$v = a$
$w = g(h(z))$
$y = h(z)$
$z = a$

$x = g(h(a))$
$u = h(a)$
$v = a$
$w = g(h(a))$
$y = h(a)$
$z = a$

$k(f(g(h(a)),h(a),a),f(g(h(a)),h(a),a)) = k(f(g(h(a)),h(a),a),f(g(h(a)),h(a),a))$

- Find a most general unifier of the two terms

$$g(x,y,f(x),z,u) \quad \text{and} \quad g(a,x,z,f(x),h(v,z))\,.$$

Answer:
$g(x,y,f(x),z,u) = g(a,x,z,f(x),h(v,z))$

$x = a$
$y = x$
$f(x) = z$
$z = f(x)$
$u = h(v,z)$

$x = a$
$y = a$

$f(a) = z$
$z = f(a)$
$u = h(v, z)$

$x = a$
$y = a$
$z = f(a)$
$z = f(a)$
$u = h(v, z)$

$x = a$
$y = a$
$z = f(a)$
$f(a) = f(a)$
$u = h(v, f(a))$

$x = a$
$y = a$
$z = f(a)$
$a = a$
$u = h(v, f(a))$

$x = a$
$y = a$
$z = f(a)$
$u = h(v, f(a))$

---

check: $g(a, a, f(a), f(a), h(v, f(a))) = g(a, a, f(a), f(a), h(v, f(a)))$

•

Exam Practice Questions WITH ANSWERS

Here are problems to practice for the final exam.

Problem 1) Find a structure that shows that $p \to q$ is not a theorem of the propositional theory whose only nonlogical axiom is $p \to (q \vee r)$.

Answer: The convention we adopted for our notation is that lowercase letters p, q, and r denote atoms. Thus, we can choose a structure ν such that $\nu(p) = \nu(r) = T$ and $\nu(q) = F$. Then, $[\![p \to (q \vee r)]\!]\nu = T$, but $[\![p \to q]\!]\nu = F$.

Problem 2) Give a formula in prenex form equivalent to

$$\exists y \forall x\, p(x, y) \to \forall x \exists y\, p(x, y)\,.$$

Answer: $\forall y \exists x \forall w \exists z\, [p(x,y) \rightarrow p(w,z)]$.

Problem 3a) "Prenex" and Skolemize as assertions the two formulas in the set Γ given below.

$$\{\exists y \forall x\, p(x,y),\ \neg \forall w \exists z\, p(w,z)\}$$

Answer: Let a and b be new constant symbols. Then the set of formulas that results from prenexing and Skolemizing as assertions the formulas in Γ is:

$$\{\forall x\, p(x,a),\ \forall z\, \neg p(b,z)\}$$

Problem 3b) Skolemize as goals the two formulas in the set Γ below.

$$\{\exists y \forall x\, p(x,y),\ \neg \forall w \exists z\, p(w,z)\}$$

Answer: Let g and h be new unary function symbols. Then the set of formulas that results from prenexing and Skolemizing as goals the formulas in Γ is:

$$\{\exists y\, p(g(y),y),\ \exists w \neg p(w,h(w))\}$$

Problem 3c) Consider a theory whose nonlogical axioms are the formulas in the set you gave as the answer in part a. Use tableau methods to prove that the theory is inconsistent; i.e. import the nonlogical axioms as assertions into the tableau

$g1$. false

and prove that the resulting tableau is valid.

Answer: A formula φ is provable in a theory T (written $T \vdash \varphi$) iff the tableau

$g1$. φ

is valid relative to theory T.

$g1$. false	given
a1. $\forall x\, p(x,a)$	Nonlogical axiom
a2. $\forall z\, \neg p(b,z)$	Nonlogical axiom
a3. $p(x,a)$	1, $\forall$ elimination
a4. $p(b,a)$	3, Substitution, $\{x := b\}$
a5. $\neg p(b,z)$	2, $\forall$ elimination
a6. $\neg p(b,a)$	5, Substitution, $\{z := a\}$
a7. false	4,6, Propositional logic

The derived tableau is valid because it has the assertion false.

Problem 4) The Boolean connective NAND is symbolized by |. In other words, NAND(p,q) is written p | q.

Show how to define exclusive-or in terms of $|$.

Some helpful information and hints: Exclusive-or(P, Q) is written $P + Q$.

Note that $P \mid Q$ is false if P and Q are both true. Otherwise P | Q is true. $P + Q$ is true if P is true, or Q is true, but P and Q are not both true.

Define $\vee$ in terms of $|$. Define $\wedge$ in terms of $|$. Define $\neg$ in terms of $|$.

Answer:

$$\neg P := P|P$$
$$P \vee Q := (P|P)|(Q|Q)$$
$$P \wedge Q := (P|Q)|(P|Q)$$

Now, since

$$P + Q \leftrightarrow ((P \vee Q) \wedge \neg(P \wedge Q))$$

we can define $P + Q$ in terms of $|$ by

$$P + Q := (((P|P)|(Q|Q))|(P|Q))|(((P|P)|(Q|Q))|(P|Q))$$

Problem 5) Find a prenex form equivalent to

$$\exists x[p(x) \wedge \forall y[\forall x[q(y, x)] \rightarrow y = x]]$$

Answer:

$$\exists x \forall y \exists x_1[p(x) \wedge (q(y, x_1) \rightarrow y = x)]$$

Problem 6) Skolemize as an assertion:

$$(**) \quad \forall x \exists y \forall z \exists w[p(x, f(a, w), f(x, f(z, y)))] \,.$$

Answer:

$$\forall x \forall z[p(x, f(a, h(x, z)), f(x, f(z, g(x))))]$$

Note: We are assuming that the function symbols g and h are new; i.e. g and h are not function symbols of the language of which (**) is a formula.

Problem 7) Skolemize as an assertion:

$$\exists x \forall y \exists z \forall w[p(x, f(a, w), f(x, f(z, y)))] \,.$$

Answer:

$$\forall y \forall w[p(b, f(a, w), f(b, f(k(y), y)))]$$

where b is a new constant symbol, and k is a new function symbol.

Problem 8) Consider

$$g1.\ \exists x\,[p(x) \wedge \forall y[p(y) \rightarrow q(x,y)]] \rightarrow \exists x\,[p(x) \vee q(x,x)]$$

Prove that the tableau is valid by tableau methods.

Answer:

$g1.\ \exists x\,[p(x) \wedge \forall y[p(y) \rightarrow q(x,y)]] \rightarrow \exists x\,[p(x) \vee q(x,x)]$	given
a1. $\exists x\,[p(x) \wedge \forall y[p(y) \rightarrow q(x,y)]]$	g1, if-split
$g2.\ \exists x\,[p(x) \vee q(x,x)]$	g1, if-split
a2. $\exists x \forall y[p(x) \wedge (p(y) \rightarrow q(x,y)]$	a1, prenex
a3. $\forall y[p(a) \wedge (p(y) \rightarrow q(a,y)]$	a2, Skolemization
a4. $p(a) \wedge (p(y) \rightarrow q(a,y))$	a3, $\forall$ elimination
$g3.\ p(x) \vee q(x,x)$	g2, $\exists$ elimination
a5. $\neg(p(x) \vee q(x,x))$	g3, duality
a6. $p(a) \wedge (p(a) \rightarrow q(a,a))$	a4, Substitution, $\{y := a\}$
a7. $\neg(p(a) \vee q(a,a))$	a5, Substitution, $\{x := a\}$
a8. false	a6,a7, Propostional logic

Problem 9) Intentionally omitted.

Problem 10) Consider

$$\mathsf{PA} \vdash \neg(x = s(x))$$

Part a) What is the induction axiom for

$$\neg(x = s(x))$$

Answer:

$$(\neg(0 = s(0)) \wedge \forall y[\neg(y = s(y)) \rightarrow \neg(s(y) = s(s(y)))]) \rightarrow \neg(x = s(x))$$

Part b) Give a formal proof that shows

$$\mathsf{PA} \vdash \neg(x = s(x))$$

Answer:

Lemma A: (Base step:) $\mathrm{PA} \vdash \neg(0 = s(0))$.

1. $\neg(0 = s(x_1))$	Q2
2. $\neg(0 = s(0))$	1, Substitution $\{x_1 := 0\}$

This completes the proof of Lemma A.

Lemma B: (Induction Step)

$$\mathrm{PA} \vdash \forall y[\neg(y = s(y)) \rightarrow \neg(s(y) = s(s(y)))]$$

	g1. $\forall y[\neg(y = s(y)) \rightarrow \neg(s(y) = s(s(y)))]$	given
	g2. $\neg(a = s(a)) \rightarrow \neg(s(a) = s(s(a)))$	g1, Skolemization
a1.	$\neg(a = s(a))$	g2, if-split // This is the induction assumption
	g3. $\neg(s(a) = s(s(a)))$	g2, if-split
a2.	$s(x_1) = s(x_2) \rightarrow x_1 = x_2$	Q1
a3.	$s(a) = s(s(a)) \rightarrow a = s(a)$	Substitution $\{x_1 := 0; x_2 := s(a)\}$
	g4. $s(a) = s(s(a)) \rightarrow a = s(a)$	g2, Propositional Logic
	g5. True	a3, g4, assertion=goal

This completes the proof of Lemma B.

	g1. $\neg(x = s(x))$	given
a1.	$(\neg(0 = s(0)) \wedge \forall y[\neg(y = s(y)) \rightarrow \neg(s(y) = s(s(y)))]) \rightarrow \neg(x = s(x))$	induction axiom
a2.	$\neg(0 = s(0))$	Lemma A
a3.	$\forall y[\neg(y = s(y)) \rightarrow \neg(s(y) = s(s(y)))]$	Lemma B
a4.	$\neg(x = s(x))$	a1, a3, Modus Ponens)
	g2. True	a4, g1, assertion=goal

Problem 11) Find a structure in which the following equivalence is not valid:

$$((\forall x\, p(x)) \rightarrow q(a)) \leftrightarrow (\forall x\, (p(x) \rightarrow q(a)))$$

Answer: Let L be a language whose nonlogical symbols are the predicate symbols p, q and whose only function symbol is the constant symbol a. Let $\mathcal{M}$ be a structure for L in which there are two individuals i and j. Let $a_{\mathcal{M}} = i$, $p_{\mathcal{M}} = \{i\}$ and $q_{\mathcal{M}} = \{j\}$. Then

$$\mathcal{M} \models \neg\forall x\, p(x)$$

Therefore,

$$\mathcal{M} \models \forall x\, p(x) \rightarrow q(a)$$

But,

$$\mathcal{M} \models p(a)$$

and

$$\mathcal{M} \models \neg q(a)$$

Therefore,

$$\mathcal{M} \not\models p(a) \rightarrow q(a)$$

Therefore,

$$\mathcal{M} \not\models \forall x[p(x) \rightarrow q(x)]$$

Therefore,

$$\mathcal{M} \not\models ((\forall x\, p(x)) \rightarrow q(a)) \leftrightarrow (\forall x\, (p(x) \rightarrow q(a)))$$

4 Practice Final Exam

Question 1: Use first order logic to express: Carlos has at least two uncles who live in Brazil. [Let: c = Carlos; uncle(x,y) iff x is the uncle of y; lives(x,y) iff x lives in y; b = Brazil.]

Question 2: Consider the following two sentences:

Sentence A:

$$\forall x[\text{greaterThan}(x, \text{Yertle}) \rightarrow \neg\, \text{turtle}(x)]$$

Sentence B:

$$\forall x[\text{turtle}(x) \rightarrow \text{greaterThan}(\text{Yertle}, x)].$$

Part 1: Express sentences A and B in English so that you show that you understand what is expressed.

Part 2: Does sentence B logically follow from sentence A? Explain.

Question 3: Prove by induction that for every natural number $n > 1$, $5^{2n} - 24n - 1$ is divisible by 576. [Hint: Let $f(n) = 5^{2n} - 24n - 1$, for every n and consider $f(n+1) - f(n)$.]

Question 4: We define the **is weaker than** relation $\sqsubseteq$ among propositional formulas by $A \sqsubseteq B$ iff $B \rightarrow A$ is a tautology. A is said to be stronger then B if B is weaker than A. (Recall that a propositional formula is a tautology iff it must be true for every assignment of truth values to its parts (i.e. every row of its truth-table evaluation is true.)

$P + Q$ is stronger than which of the following: $P \vee Q$, $P \wedge Q$, $P \rightarrow Q$, $P + Q$, $P|Q$? [+ is exclusive-or, | is not-both (i.e. NAND).]

Question 5: Find a most general unifying substitution that solves the equation

$$k(f(x,u,a),f(g(y),y,v)) = k(f(g(y),y,v),f(w,h(z),z))$$

if one exists, or explain why there is no unifier.

Question 6:

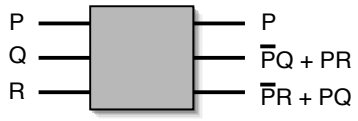
Part 1: Give a formula in prenex normal form equivalent to the following formula:

$$\exists x \, \forall y \, [\exists x \, [(\forall x \, p(x)) \rightarrow [\exists y \, q(f(x), y) \vee r(x, y)]] \rightarrow \neg q(g(x, y))]$$

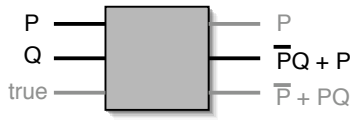
Part 2: Skolemize

$$\exists x \forall y \forall z \exists w \, [p(w, a, z) \rightarrow q(x, f(x), y)]$$

Question 7: The diagram below denotes a *Fredkin gate*. P, Q and R are Boolean-valued. The expression $\overline{\mathsf{P}}\mathsf{Q} + \mathsf{PR}$ in the diagram denotes the Boolean-value of the propositional formula $(\neg\, \mathsf{P} \wedge \mathsf{Q}) \oplus (\mathsf{P} \wedge \mathsf{R})$. ($\oplus$ denotes exclusive-or.)



Notice that if the value of R is fixed at true, then the Fredkin gate acts as an OR gate:



Show how to connect two Fredkin gates to act as a NAND gate. You can fix any of the inputs on either gate to a constant value as illustrated.

5 GNU Free Documentation License

Version 1.3, 3 November 2008

<http://fsf.org/>

Preamble

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "**Document**", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "**you**". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "**Modified Version**" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "**Secondary Section**" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "**Invariant Sections**" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "**Cover Texts**" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "**Transparent**" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "**Opaque**".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "**Title Page**" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "**publisher**" means any person or entity that distributes copies of the Document to the public.

A section "**Entitled XYZ**" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "**Acknowledgements**", "**Dedications**", "**Endorsements**", or "**History**".) To "**Preserve the Title**" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or non-commercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option,

you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and

list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of

some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See `http://www.gnu.org/copyleft/`.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

> Copyright © YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with ... Texts." line with this:

> with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

6 We shall begin with nothing other than our ability to reason logically

Directive 6.1: Read the Wikipedia article on *Set Theory* at

http://en.wikipedia.org/wiki/Set_theory

Then work through the material in the Wikibook *Discrete Mathematics/Set Theory* found at

http://en.wikibooks.org/wiki/Discrete_Mathematics/Set_theory#Set_Theory_Exercise_1

Work through the first three exercise sections whose links are found on the Wikibook page. △

Directive 6.2: Consider the empty set, which is something but has no elements in it. We will denote the empty set with the symbol:

$$\emptyset$$

Now contemplate what it is in it. We will take as a starting point an assumption, i.e. an *axiom*:

$$\emptyset \text{ is a set.}$$

This kind of assumption is called a *comprehension axiom*. Intuitively, the idea is that we can comprehend the idea of a set with no elements in it in a way that is logically consistent with lots of other ideas we have about sets. It is an assumption. We cannot prove there is such a set. We will make that assumption one of our starting points. The other starting points are the basic operations that can be applied to sets that are discussed in the wiki article and wiki book in the previous directive. In this section, we will build the low levels of the *Von Neumann Universe* that are of interest to us in computer science and engineering. △

Note 6.1: When you read technical material in this course, such as textbooks, wikibooks, technical articles, etc., verify what you read and read critically. △

Note 6.2: According to the wikibook on sets a set can be defined as *a collection of things that are brought together because they obey a certain rule*. What's a *collection*? What's a *rule*? What does it mean, *mathematically*, to bring things together? You will never succeed in making a mathematical model of these things without presupposing mathematics that amounts to presupposing that we already have sets available. And around and around we

go – which makes the wikibook definition of a set a piece of crap. But - this is extremely important - a piece of crap like this so-called definition can still embody powerful and highly useful intuitions.

As we proceed we will state various comprehension axioms that conveniently allow us to express specific rules for forming sets without the need to have a general theory of rules, collections, and "bringings together".

In this note I have criticized using a term like *rule* in a mathematical definition, and then I used the very word I criticized in the previous paragraph. This seems inconsistent. What needs to be reflected upon here is that the use of the word *rule* in the previous paragraph is in *commentary* on mathematical description and not in mathematical description itself.

In section (1) we used sets to define relations. We now introduce a few specific relation-like ideas that are not relations according to what was given in section (1). We make the assumption that for anything x, it is true that x is a set, or it is true that x is not a set, but not both. Secondly, we make the assumption that for anything x and any set S, that it is either true that x is a *member* of S or it is not true that x is a *member* of S but not both. (If everything is a set, then we can dispense with the first of these two assumptions - and that is what is done in formal set theory. We need not worry about that for now.) $\triangle$

7 Some set-theoretic constructions

Definition 7.1: For any sets A and B, the set $A \times B$, called the *Cartesian Product* of A and B, is the set of all ordered pairs (a, b) such that $a \in A$ and $b \in B$. In other words,

$$A \times B = \{(a, b) \mid a \in A \text{ and } b \in B\}.$$

$\triangle$

Remark 7.1: The preceding definition begs a question: What is an ordered pair? The issue is that whatever an ordered pair is, it should be something independent of how it is denoted. One way to rigorously define ordered pairs, and while we're at it, ordered n-tuples in terms of previously understood notions is the following: Let $\mathbb{N}$ be the set of non-negative integers. Let $n \in \mathbb{N}$. An sequence of elements of a set A of length n is a function from $\{0, \ldots, n-1\}$ to A. Let $\mathfrak{a}$ be a sequence of length n, where $n > 0$. Then we also denote $\mathfrak{a}$ by $(\mathfrak{a}_0, \ldots, \mathfrak{a}_{n-1})$. Thus, the ordered pair (x, y) of elements of a set S, is a function from $\{0, 1\}$ to S where the application of the function to 0 is x and the application of the function to 1 is y. An n-tuple is a sequence of length n. $\triangle$

Remark 7.2: There's a problem with the previous remark: An ordered pair is a function and a function is a certain kind of set of ordered pairs - which seems circular. What's the way

out of the circularity? One way out is to construct *basic* ordered pairs out of sets. Given two things x, y, which may or may not be the same, we obtain *basic* ordered pairs $\{\{x\},\{x,y\}\}$ and $\{\{y\},\{y,x\}\}$. We denote these two *basic* ordered pairs by (x,y) and (y,x), respectively. We can then take functions to be a certain kinds of sets of *basic* ordered pairs. △

Directive 7.1: Prove that $(x,y) = (x',y')$ iff $x = x'$ and $y = y'$, where (x,y) and (x',y') are *basic* ordered pairs. △

Notation 7.1: Let $A_0, \ldots, A_{n-1}$ be a sequence of length n of sets. Then

$$A_0 \times \ldots \times A_{n-1}$$

is the set of sequences $\mathfrak{a}$ of length n to the set $A_0 \cup \ldots \cup A_{n-1}$ such that $\mathfrak{a}_k \in A_k$, for each $k \in \{0, \ldots, n-1\}$. △

Example 7.1: Let $A = \{0,1,2\}$ and let $B = \{0,1\}$. Then

$$A \times B = \{(0,0),(0,1),(1,0),(1,1),(2,0),(2,1)\}.$$

△

Example 7.2: Let $\mathbb{N}$ be the set of non-negative integers. That is, $\mathbb{N} = \{0,1,2,3,\ldots\}$. Then

$$\mathbb{N} \times \mathbb{N} = \{(m,n) \mid m \in \mathbb{N} \text{ and } n \in \mathbb{N}\}.$$

△

Definition 7.2: A *relation* R from set A to set B is a triple (P,A,B) where P is a subset of $A \times B$. (We usually just write R instead of writing the whole triple (R,A,B) when we refer to a relation R from a set A to a set B; i.e. we abuse notation and let the subset of $A \times B$ stand in for R.) △

Remark 7.3: We informally identify a *relation* R from $A_0 \times \ldots \times A_{n-1}$ to B with a subset of $A_0 \times \ldots \times A_{n-1} \times B$. We call such a relation an n-ary relation. This begs a question: what is $A_0 \times \ldots \times A_{n-1}$, if $n = 0$? This cartesian product is a set of sequences from the empty set, to some set. There is only one such such function (remember: a sequence is a function), the empty function. The empty function takes no inputs and returns nothing. So, the *empty cartesian product* is a set containing the empty function and no other elements. △

Definition 7.3: *(Cartesian powers of a set).* Let A be a set. For each $n \in \mathbb{N}$, A^n is the set $A_0 \times \ldots \times A_{n-1}$, where each A_k is A. A^n is called the n^{th} *cartesian power* of A. A^0 is the empty cartesian product. △

Definition 7.4: *(n-ary relation on a set).* Let $n \in \mathbb{N}$. An n-ary relation on a set A, where $n > 0$ is a relation from A^{n-1} to A. A 0-ary relation on a set A is a subset of A^0. A *unary* relation is a 1-ary relation. A *binary* relation is a 2-ary relation. △

Problem 7.1: How many 0-ary relations are there on a nonempty set A? Describe them. How many 0-ary relations are there on the empty set? Describe them. Explain why no 0-ary relation on the set $\{0\}$ can be same relation as any 0-ary relation on the set $\{1\}$. △

Problem 7.2: Show that A^1 is in one-to-one correspondence with A. △

Definition 7.5: *(Directed graph).* A *directed graph* is a pair (V, E), consisting of a set V and a binary relation E on V. A member of V is called a *vertex* (*vertices* [pl.]). A member of E is called an *edge*. △

Notation 7.2: Let $D = (V_D, E_D)$ be a directed graph. Let u be a vertex in V_D. Let $E_D\restriction u = \{(u, v) \mid (u, v) \in E_D\}$. (These are the edges in D from u to some vertex.) △

Problem 7.3: How many directed graphs are there with vertex set $\{0, 1, 2\}$? Explain your reasoning. △

8 Minimal, maximal, minimum, maximum, least, greatest, etc.

Note 8.1: What does *least* mean? To answer this question we will sort out the concepts of *mimimal*, *minimum*, *maximal* and *maximum*. *least* and *smallest* are synonyms for *minimum*. *greatest* and *largest* are synonyms for *maximum*. We will also consider *lower bound* and *upper bound*. △

Definition 8.1: *(minimum, maximum)* Let $(D, \sqsubseteq)$ be a poset and let $A \subseteq D$. a is a *minimum element* of A iff $a \in A$ and for every $b \in A$, $a \sqsubseteq b$. a is a *maximum element* of A iff $a \in A$ and for every $b \in A$, $b \sqsubseteq a$. △

Definition 8.2: *(minimal)* Let $(D, \sqsubseteq)$ be a poset and let $A \subseteq D$. a is a *mimimal element* of A iff $a \in A$ and for every $b \in A$, if $b \sqsubseteq a$, then $b = a$. **Comment:** Less formally, a minimal element of A is an element a of A such that there is no element of A below a. △

Definition 8.3: *(maximal)* Let $(D, \sqsubseteq)$ be a poset and let $A \subseteq D$. a is a *maximal element* of A iff $a \in A$ and for every $b \in A$, if $a \sqsubseteq b$, then $a = b$. △

Definition 8.4: *(lower bound, upper bound)* Let $(D, \sqsubseteq)$ be a poset and let $A \subseteq D$. An element x in D is a *lower bound* of A iff for every a in A, $x \sqsubseteq a$. An element x in D is an *upper bound* of A iff for every a in A, $a \sqsubseteq x$. We use the notation $x \sqsubseteq A$ to say that x is a lower bound of A. Similarly, we write $A \sqsubseteq x$ to say that x is an upper bound of A. **Comment:** An upper bound of A may or may not be an element of A. △

Proposition 8.1: Let $(D, \sqsubseteq)$ be a poset and let $A \subseteq D$. Then,
(i) a is a *least* element of A iff a is a greatest lower bound of A, and $a \in A$.
(ii) a is a *greatest* element of A iff a is a least upper bound of A, and $a \in A$. △

Problem 8.1: Find an example of an infinite poset D and subset A of D that has a *unique* minimal element that is not the least element of A. You may find it helpful to sketch posets by using Hasse diagrams - check out the wiki article at

http://en.wikipedia.org/wiki/Hasse_diagram

△

Problem 8.2: Let $(D, \sqsubseteq)$ be a poset and let $A \subseteq D$. Show that there is at most one greatest lower bound of A. (Therefore, if A has a greatest lower bound, then it is unique.) Show that there is at most one least upper bound of A. △

9 Additional practice problems for the midterm exam

Problem 9.1: Use the principle of mathematical induction to prove that every nonempty set of nonnegative integers has a least member. △

Problem 9.2: Let $|$ be the binary relation on the set $\mathbb{N}$ of nonnegative integers defined by $m|n$ iff n is an integer multiple of m. Is $|$ a partial ordering? Prove your answer. △

Problem 9.3: Let $|$ and $\mathbb{N}$ be as in the previous problem. Is $\mathbb{N}$ directed with respect to $|$? Prove your answer. △

Problem 9.4: A (proper) *filter* F on a nonempty set S is a nonempty collection of nonempty subsets of S with two properties:
For all subsets A and B of S:

(a). If $A \in F$ and $A \subseteq B$ then $B \in F$.

(b). If $A \in F$ and $B \in F$ then $(A \cap B) \in F$.

Consider the set $\mathbb{N} \longrightarrow \mathbb{N}$ of all functions that map nonnegative integers to nonnegative integers. Let F be a filter on $\mathbb{N}$. The binary relation $\sim_F$ on $\mathbb{N} \longrightarrow \mathbb{N}$ defined by

$$f \sim_F g \text{ iff } \{n \in N \mid f(n) = g(n)\} \in F$$

$\sim_F$ is a reflexive and symmetric relation which we take at this point without proof. Prove that $\sim_F$ is transitive, and is therefore an equivalence relation. △

Problem 9.5: For any two (not necessarily different) functions $f : \mathbb{N} \longrightarrow \mathbb{N}$ and $g : \mathbb{N} \longrightarrow \mathbb{N}$, the *pointwise sum* $(f + g) : \mathbb{N} \longrightarrow \mathbb{N}$ of f and g is defined by

$$(f + g)(n) = f(n) + g(n)$$

Similarly the *pointwise product* $(f * g) : \mathbb{N} \longrightarrow \mathbb{N}$ of f and g is defined by

$$(f * g)(n) = f(n) * g(n)$$

Let f, g, h, k be functions in $\mathbb{N} \longrightarrow \mathbb{N}$ and, as in previous problems, let F be a proper filter on $\mathbb{N}$. Prove that if $f \sim_F g$, where $\sim_F$ is defined as in the previous problem, and $h \sim_F k$, then $(f + g) \sim_F (h + k)$. △

10 Complex numbers

Note 10.1: We begin the study of generating functions and some special seqeunces by following chapter 7 (and a little of chapter 6) of Graham et.al. △

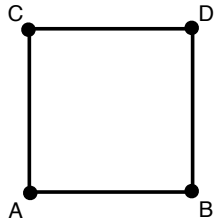
Definition 10.1: Let $G : \mathbb{C} \rightharpoonup \mathbb{C}$, be a partial function defined by

$$G(z) = \sum_{n=0}^{\infty} g_n z^n$$

where the infinite series on the right-hand side of the above equation is absolutely convergent within a radius of convergence, and where $\mathbb{C}$ is the set of complex numbers and for each n, $g_n \in \mathbb{C}$. Given such a fuction G, we use the notation $[z^n]G(z)$ to denote g_n. △

Question 10.1: What are the complex numbers? △

Note 10.2: Consider a graph $\mathcal{S}$ (a graph is a set of vertices together with a symmetric relation on the set of vertices called the edge relation).



The relation depicted by this graph is a relation from $\{A, B, C, D\}$ to $\{A, B, C, D\}$ and is determined by one of the subsets of

$$\{A, B, C, D\} \times \{A, B, C, D\}$$

In other words, the relation depicted by this graph is a triple

$$(\{A, B, C, D\},\ \{A, B, C, D\},\ R)$$

where

$$R \subseteq \{A, B, C, D\} \times \{A, B, C, D\}$$

Thus, R is a set of pairs. (We regard a graph as a directed graph that has a symmetric edge relation. For example, the edge between A and C is really two edges: (A, C) and (C, A). △

Problem 10.1: List the pairs in R. [**Hint:** There are eight pairs (8 because the edges point both ways).] △

Definition 10.2: Let v be vertex in a directed graph (V, E). The *neighborhood* of v, denoted by $\text{Nbhd}(v)$ is the set of vertices $\{u \mid (v, u) \in E\}$. △

Example 10.1: In $\mathcal{S}$,

$$\text{Nbhd}(A) = \{B, C\}$$

△

Problem 10.2: List the neighborhoods of each of the vertices in $\mathcal{S}$. (Note that the edge relation in $\mathcal{S}$ is irreflexive.) △

Definition 10.3: *(homomorphism)* Suppose we have two directed graphs (which might or might not actually be different) $D_1 = (V_1, E_1)$ and $D_2 = (V_2, E_2)$. Suppose

$$\varphi : V_1 \longrightarrow V_2$$

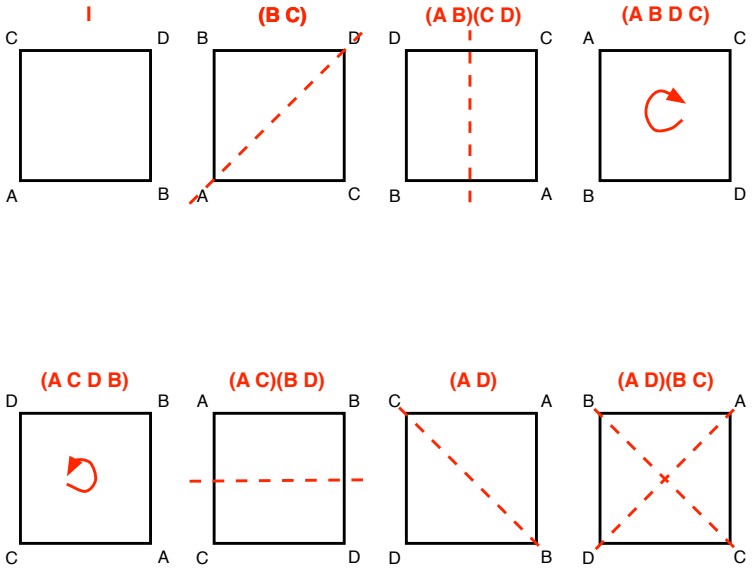
φ is a *homomorphism* of D_1 and D_2 iff for all vertices x and y of D_1, if $(x, y) \in E_1$, then $(\varphi(x), \varphi(y)) \in E_2$. An *isomorphism* from D_1 to D_2 is a 1-to-1 and onto homomorphism from D_1 to D_2 whose inverse is also a homomorphism. An isomorphism from D to D is called an *automorphism* of D. The automorphisms of D are also called the *symmetries* of D. △

Problem 10.3: Suppose that $D = (V, E)$ is a finite graph, and $\varphi : V \longrightarrow V$ is a 1-to-1 and onto homomorphism. Let $\hat{\varphi} : E \longrightarrow E$ be defined by $\hat{\varphi}(u, v) = (\varphi(u), \varphi(v))$. Show that $\hat{\varphi}$ is well-defined; i.e. this definition of $\hat{\varphi}$ is consistent; i.e. $(\varphi(u), \varphi(v)) \in E$ if $(u, v) \in E$. △

Problem 10.4: Show that $\hat{\varphi}$ is 1-to-1 and onto. Then show that φ^{-1} is a homomorphism. Show that this result can fail if D is an infinite graph. △

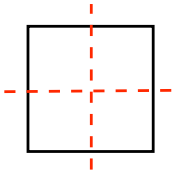
Problem 10.5: Find all symmetries of $\mathcal{S}$. Verify the answer given below. (There are eight such homomorphisms.)

Answer:



△

Problem 10.6: What automorphism is suggested by:



△

Example 10.2: Here are four automorphisms (i.e. symmetries) of the square graph of Note 3.1:

(1) $\pi_{00}(X) = X$ for every vertex X of the graph.
$\pi_{01}(A) = C$, $\pi_{01}(C) = A$, $\pi_{01}(B) = D$, $\pi_{01}(D) = B$. π_{01} flips the figure over a horizontal axis.
$\pi_{10}(A) = B$, $\pi_{10}(B) = A$, $\pi_{10}(C) = D$, $\pi_{10}(D) = C$. π_{10} flips the figure over a vertical axis.
$\pi_{11}(A) = D$, $\pi_{11}(D) = A$, $\pi_{11}(B) = C$, $\pi_{11}(C) = B$. π_{11} rotates the figure 180 degrees, or π radians.

Note for example that $\pi_{10} \circ \pi_{01} = \pi_{11}$, where $\circ$ is the function composition operation. In particular, for example,

$$(\pi_{10} \circ \pi_{01})(A) = \pi_{10}(\pi_{01}(A)) = \pi_{10}(C) = D$$

△

Problem 10.7: Fill in the rest of the table below:

$\circ$	π_{00}	π_{01}	π_{10}	π_{11}
π_{00}				
π_{01}			π_{11}	
π_{10}				
π_{11}				

△

Problem 10.8: The four symmetries of the previous problem together with the composition operation $\circ$ form a system called an *Abelian group*. (Look up the definition of a group later. A group is Abelian iff (by definition) the order in which two elements of the group are combined using the group operation does not matter to the result. Verify that for any two vertices X and Y in $\mathcal{S}$, there is exactly one symmetry that maps X to Y. △

Definition 10.4: A group of symmetries of a directed graph that has the property that there is exactly one symmetry that maps from a given vertex to a given vertex is called a *regular action* on the digraph. The symmetries in a regular action are called *translations*.
△

Problem 10.9: What familar system is this?

$\circ$	0	1
0	0	1
1	1	0

△

Problem 10.10:

(a). How do we add matrices? For example,

$$\begin{bmatrix} -2 & 5 \\ 7 & -1 \end{bmatrix} + \begin{bmatrix} 3 & 4 \\ -1 & 1 \end{bmatrix} = ?$$

(b). Vectors are matrices with just 1 column. How do we add vectors? For example,

$$\begin{bmatrix} -2 \\ 7 \end{bmatrix} + \begin{bmatrix} 3 \\ -1 \end{bmatrix} = ?$$

(c). How do we add $0, 1$-valued vectors mod 2? For example, using mod-2 arithmetic:

$$\begin{bmatrix} 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} = ?$$

(d). Fill in the following table for addition mod-2 of 2-dimensional $0, 1$-valued vectors.

$+_2$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \end{bmatrix}$
$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$				
$\begin{bmatrix} 0 \\ 1 \end{bmatrix}$			$\begin{bmatrix} 1 \\ 1 \end{bmatrix}$	
$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$				
$\begin{bmatrix} 1 \\ 1 \end{bmatrix}$				

(e). Compare this mod-2 addition table to the composition table for the translations, π_{00}, π_{01}, π_{10}, π_{11}. What correspondences do you see?

△

Example 10.3: A scalar is an object that you can multiply a vector by with some of the basic algebraic laws holding. For example, for any pair of $0, 1$-valued vectors that are added mod-2, a scalar a should satisfy the following distributivity law, among others:

$$a\left(\begin{bmatrix} x \\ y \end{bmatrix} +_2 \begin{bmatrix} u \\ v \end{bmatrix}\right) = a\begin{bmatrix} x \\ y \end{bmatrix} +_2 a\begin{bmatrix} u \\ v \end{bmatrix}$$

Notice that multiplying by the scalar a is a linear function. So the candidates for what could be the scalars are linear functions that map 2-dimensional vectors to 2-dimensional vectors and satisfy mod-2 arithmetic. △

Problem 10.11: All of the linear functions that map 2-dimensional $0, 1$-valued vectors to 2-dimensional $0, 1$-valued vectors using mod-2 arithmetic can be described by 2×2 matrices. There are 16 of them:

$$\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix} \quad \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

(a). There are other rules. For example, nonzero scalars must be invertible. Which of these matrices are invertible?

(b). Which of these matrices are self-invertible?

(c). Find two invertibles that add to give a sum that is neither the zero matrix nor invertible.

(d). We are looking for a system of scalars. The sum of two scalars and the product of two scalars has to be a scalar. Among the matrices that will serve as scalars, the sum of two invertibles has to be invertible and the identity matrix has to be a scalar. Therefore, what is the largest set of matrices among the 16 that can form a system of scalars?

△

Definition 10.5: *(Subgroup)* A *subgroup* H of a group G ($H < G$ denotes that H is a subgroup of G) is a group such that the set of elements of H is a subset of the set of elements of G. △

Theorem 10.1: *(Lagrange's Theorem).* The size of a subgroup H of a group G divides the size of G, if G is finite. ∎

Question 10.2: So, where are the complex numbers? △

Problem 10.12: Consider the set of all 2-by-2 matrices

$$M_{\mathbb{C}} = \left\{ \left[\begin{array}{cc} r\cos\Theta & -r\sin\Theta \\ r\sin\Theta & r\cos\Theta \end{array} \right] \;\middle|\; r, \Theta \in \mathbb{R} \right\}$$

(a). Show that for any matrix $A_{(r,\Theta)}$ in $M_{\mathbb{C}}$, if $r \neq 0$, then $A_{(r,\Theta)}$ is invertible and find its inverse.

(b). Show that if $r = 1$, then $A_{(r,\Theta)}$ acts on the vector space $\mathbb{R}^2$ by rotating it counterclockwise through angle Θ.

(c). Show that the product of any two matrices in $M_{\mathbb{C}}$ is in $M_{\mathbb{C}}$.

(d). Is the sum any two matrices in $M_{\mathbb{C}}$ in $M_{\mathbb{C}}$? Prove your answer.

△

11 Mathematical Induction

www.math.uga.edu/∼pete/3200induction.pdf

12 Generating functions

Remark 12.1: This section is about functions that generate certain other things, e.g. Fibonacci numbers. But, to think that this section is about how to generate functions would, at least at first, lead you to a misguided understanding. ∎

To repeat the basic definition pertinent to generating functions:

Definition 12.1: Let $G : \mathbb{C} \rightharpoonup \mathbb{C}$, be a partial function defined by

$$G(z) = \sum_n g_n z^n$$

defined on an open disk of (absolute) convergence, where $\mathbb{C}$ is the set of complex numbers and for each n, $g_n \in \mathbb{C}$. Given such a partial fuction G, we use the notation $[z^n]G(z)$ to denote g_n. △

Proposition 12.1: Let $\alpha, \beta, c \in \mathbb{C}$ and let F and G be generating functions. Then

(a). $\alpha F(z) + \beta G(z) = \sum_n^{\infty} (\alpha f_n + \beta g_n) z^n$.

(b). $z^m G(z) = \sum_n g_n z^{n+m} = \sum_n g_{n-m} z^n$.

(c). $\sum_n g_{n+m} z^n = \frac{1}{z^m} G(z)$.

(d). $G(cz) = \sum_n (c^n g_n) z^n$.

(e). $\frac{dG}{dz}(z) = G'(z) = \sum_n n g_n z^{n-1} = \sum_n n g_n z^{n-1} = \sum_n (n+1) g_{n+1} z^n$.

(f). $z\frac{dG}{dz}(z) = G'(z) = z\sum_n n g_n z^n$.

(g). If $G(z) = \sum_{n\geq 0} g_n z^n$, then $\int_0^z G(t)\,dt = \sum_{n\geq 0} \frac{1}{n+1} g_n z^{n+1} = \sum_{n\geq 1} \frac{1}{n} g_{n-1} z^n$

(h). $F(z)G(z) = \sum_n \left(\sum_k f_k g_{n-k} \right) z^n$.

■