Lecture 1: Course Overview and Propositional Logic

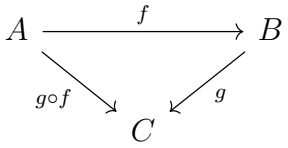
Lecturer: Max S. New
Scribe: Max S. New

January 4, 2023

See the course homepage for the syllabus. Sign up for scribing and problem set solution presentations this week.

1 What is this course About?

This is a course on *category theory* and its application to computer science, specifically applications to formal logics and programming languages. We will get into category theory proper after a few lectures but at a very high level, category theory is a branch of mathematics that studies the algebra of structures and transformations. At a very high level, category theory studies the contexts in which it makes sense to draw the following diagram[1]:



Here A, B, C are what we call *objects* and f, g and $g \circ f$ are what we call *morphisms* between objects. These simple kind of diagrams are ubiquitous in mathematics:

1. In one context the objects A, B, C might represent sets and f, g are functions with $g \circ f$ their composition.

2. In a programming language context, A, B, C are typically datatypes of a programming language and f, g are expressions in the language and $g \circ f$ is a syntactic substitution.

3. In a logical context, A, B, C can stand for propositions and f, g are proofs of an implication, with $g \circ f$ a proof by transitivity.

4. In an algebraic context, A, B, C might represent groups and f, g homomorphisms.

[1] see the LaTeX source for advice on how to make a diagram like this

5. In topology, A, B, C might be topological spaces or manifolds and f, g continuous functions.

Among the many examples of categories, the ones we study in logic and programming languages turn out to have a special property: they are in a sense *minimalistic* structures. In category-theoretic terminology, logics and programming languages are *initial* or *free* structures. This means that we can use them as a *syntax* for representing mathematical structures and transformations. The benefits of this situation go both ways: we can use syntactic methods to help us do our mathematics, but also it means that we can use mathematical *models* to help us define programming languages and logics that precisely capture an intended semantics. In particular, we will see throughout the course that the category-theoretic notion of a *universal property* corresponds to certain well-behaved *type constructors* in programming languages and *connectives* in logic.

Category theory is a kind of algebra and so is inherently quite abstract. But abstraction is nothing to fear to a computer scientist: all of our programming languages are abstract systems of symbols that are interpreted in various ways. To emphasize this connection, the topics of this course are structured in pairs: we will study mathematical notions from category theory but paired with the logic or programming language that is their syntactic form. To start we will study the more familiar syntax first and then the semantic models, but eventually we will work as practioners do and allow each method to inform the other. Historically some of these systems were developed syntactically and then a semantics was designed, some the semantics inspired a new syntax and for some the syntax and semantics were developed independently and the relationship established later.

Here is a tentative overview of the systems we will study this semester.

Syntax	Semantics
Intuitionistic Propositional Logic	Heyting Algebras
Simply Typed Lambda Calculus	Bi-cartesian Closed Categories
Monadic Metalangauge	Strong Monads
Call-by-push-value	Strong Adjunctions
Linear Lambda Calculus	Monoidal Categories
Dependent Type Theory	Toposes

2 Intuitionistic Propositional Logic

Logic is the study of *inference*, i.e., how to derive new facts from those previously established. We will study a fairly simple system of logic called *intuitionistic propositional logic* (IPL) using a presentation called *natural deduction*[2].

The basic notions of propositional logic are *propositions*, *judgments* and *proofs*. First, propositions are the subjects of inference: we can think of them as questions

[2]alternative presentations of logic include Hilbert systems and Sequent Calculus

whose truth we are reasoning about. We will write propositions as A, B, C. A judgment is a statement that our logic is concerned with reasoning about. In IPL there is a single judgment called the *hypothetical* judgment which is of the form

$$A_1 \,\text{true}, \ldots \vdash A \,\text{true}$$

which we read as "if all of A_1,... are true then A is true". The symbol $\vdash$ is called a "turnstile". In other logics we might have different judgments such as A false or A true tomorrow but for IPL we only have one, and so we will usually abbreviate this as $A_1, \ldots \vdash A$. Further to avoid too many "..." we will abbreviate a finite sequence of 0 or more propositions as Γ or Δ and call it a "context". So our judgment above will be abbreviated as $\Gamma \vdash A$.

We describe what proofs are valid in the system by giving *rules* of the form[3].

$$\frac{\Gamma_1 \vdash A_1 \ldots}{\Delta \vdash B}$$

That is if we can establish proofs of all of the judgments above the line then we can construct a proof of the judgment below. We can view this rule as a node of a proof tree with the bottom judgment being the root.

The most basic rules of IPL are true for any proposition: the assumption rule and the substitution principle. The assumption rule simply states that if we know A to be true then A is true. For a first attempt we might write this as

$$\frac{}{A \vdash A}$$

That is we have a complete proof that if A is true then A is true. However, this rule as written is actually restrictive: it says we can conclude A is true only if the *only* thing we know is that A is true. Instead we will use the following rule:

$$\frac{}{\Gamma, A, \Delta \vdash A}\ \textsc{Assumption}$$

This version says we can use the fact that A is true regardless of however many other facts there are to the right or left. We will consider contexts as ordered lists. Note here that Γ, A, Δ are what are called *schematic variables*: the rule can be applied for any choice of concrete contexts Γ, Δ and proposition A.

The substitution principle gives us a form of transitive reasoning: if we can prove A and from A we can prove B then we can prove B. As a rule we write this as

$$\frac{\Gamma \vdash A \qquad \Gamma, A \vdash B}{\Gamma \vdash B}\ \textsc{subst}$$

The propositions in IPL are built up from *propositional variables* and *logical connectives*. The propositional variables are chosen based on what the logic used for. Here are some examples:

[3]see the LaTeX source for how these rules are typeset.

1. "Socrates is a man"
2. "category theory is fun"
3. "variable x has value 5"
4. "x is greater than 5"
5. "x is greater than 6"

From the perspective of logic, these are just uninterpreted symbols and so we write them as X, Y, Z. They do not have a predetermined meaning and by default don't have any rules specific to them. We can add rules codifying domain-specific knowledge as *axioms* to the system. We write axioms as $A_1, \ldots \Rightarrow A$. Some examples:

1. $\cdot \Rightarrow$ "cateogry theory is fun"
2. "x is greater than 6" $\Rightarrow$ "x is greater than 5"
3. "Socrates is a man" $\Rightarrow$ "Socrates is mortal"

By picking propositional variables and axioms from our chosen domain we can use propositional logic to make correct inferences about our domain of interest. For each axiom of the system we add a corresponding rule:

$$\frac{\Gamma \vdash A_1 \ldots}{\Gamma \vdash A} \,\mathrm{Ax}(A_1 \ldots \Rightarrow A)$$

We will usually work with an implicit, fixed set of propositional variables and axioms. If we want to be explicit, we group them together as a *signature* $\Sigma = (\Sigma_0, \Sigma_1)$ where Σ_0 is a list of names of propositional symbols and Σ_1 is a list of axioms using propositions generated by those symbols.

Besides the propositional variables, all other propositions are built up using the *logical connectives* $\top, \wedge, \bot, \vee, \supset$. We'll introduce each of these with their corresponding rules. Each connective comes with 0 or more *introduction rules* which tell us how to proof the proposition and 0 or more *elimination* rules that tell us how we can *use* the proposition to prove other things.

First, we have $\top$, pronounced "truth" which is the trivially true proposition. Its introduction form says that we can prove it under any assumptions, trivially.

$$\frac{}{\Gamma \vdash \top} \,\top\mathrm{I}$$

Since we take nothing to prove $\top$, there is no elimination form as no information can be gained from its proof.

Next, we have $A \wedge B$, pronounced "A and B" also known as (binary) conjunction. Its introduction form says that to prove $A \wedge B$ we must prove A and B are true separately:

$$\frac{\Gamma \vdash A \qquad \Gamma \vdash B}{\Gamma \vdash A \wedge B} \,\wedge\mathrm{I}$$

What should the elimination form be? Well to prove it we proved both A and B so if we know it is true we should be able to extract that A and B are true. That is, we will have two elimination forms, which give us each side:

$$\frac{\Gamma \vdash A \wedge B}{\Gamma \vdash A} \wedge\text{E1} \qquad\qquad \frac{\Gamma \vdash A \wedge B}{\Gamma \vdash B} \wedge\text{E2}$$

Note that $\top$ is a kind of nullary (0-ary) form of the conjunction.

Next, the dual to conjunction are the nullary and binary *disjunction*. The nullary disjunction $\bot$, pronounced "false" is a trivially false proposition. Since the judgment of IPL is in terms of truth rather than falsity we can't state this as directly as truth. Rather we define it by the fact that if falsity holds then we have derived a contradiction and therefore anything is provable. This is the *principle of explosion* or in Latin, *ex falso quodlibet*:

$$\frac{\Gamma \vdash \bot}{\Gamma \vdash A} \bot\text{E}$$

Dually to trivial truth, there is no introduction form for false.

Then the dual to binary conjunction is binary disjunction $A \vee B$, pronounced "A or B". This has two introduction forms, which allow us to prove $A \vee B$ from A or from B:

$$\frac{\Gamma \vdash A}{\Gamma \vdash A \vee B} \vee\text{I1} \qquad\qquad \frac{\Gamma \vdash B}{\Gamma \vdash A \vee B} \vee\text{I2}$$

The elimination rule says that if $A \vee B$ is true and we can prove some third proposition C both from assuming A to be true and from assuming B to be true, then we can conclude that C is true:

$$\frac{\Gamma \vdash A \vee B \qquad \Gamma, A \vdash C \qquad \Gamma, B \vdash C}{\Gamma \vdash C} \vee\text{E}$$

The final connective is *implication*, which gives us a way of talking about hypothetical/conditional reasoning within the language of propositions. Written $A \supset B$ and pronounced "A implies B", the introduction form says that to prove A implies B it is sufficient to prove B under the additional assumption that A is true:

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \supset B} \supset\text{I}$$

Then the elimination form says that if we have established $A \supset B$ and A is true, then B is true, which is sometimes called in Latin *modus ponens*:

Notice the similarity between the implication elimination rule and the substitution principle: one is a statement about the "external" notion of hypothetical reasoning $\vdash$ and the other is about the "internal" notion of implication.

Now that we have defined the rules of IPL we can write proofs within it. Here's a proof of $\cdot \vdash P \supset (Q \supset P)$:

$$\dfrac{\dfrac{\dfrac{}{P, Q \vdash P}\,\textsc{Assumption}}{P \vdash Q \supset P}\,\supset\text{I}}{\cdot \vdash P \supset (Q \supset P)}\,\supset\text{I}$$

Here's (most of) a proof that if $A \supset (B \wedge C)$ is true then $(A \supset B) \wedge (A \supset C)$ is true:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{A \supset (B \wedge C), A \vdash A \supset (B \wedge C)} \quad \dfrac{}{A \supset (B \wedge C), A \vdash A}}{A \supset (B \wedge C), A \vdash B \wedge C}\,\supset\text{E}}{A \supset (B \wedge C), A \vdash B}\,\wedge\text{E1}}{A \supset (B \wedge C) \vdash A \supset B}\,\supset\text{I} \quad \dfrac{\vdots}{A \supset (B \wedge C) \vdash A \supset C}}{A \supset (B \wedge C) \vdash (A \supset B) \wedge (A \supset C)}\,\wedge\text{I}$$

where the ($\vdots$) stands in for a similar proof derivation to the left side. This is quite an explicit notation but it emphasizes in a visual way the *tree* structure of our formal proofs.

2.1 Admissible and Derivable Rules

The rules we have considered so far are called the *primitive* rules of IPL. But when we are doing proofs, it can be useful to consider additional rules that don't change which judgments are provable. Such a rule is called an *admissible rule* of the system. Among the admissible rules are the *derivable rules*, those that can be shown to be admissible by a uniform finite sequence of primitive rules.

For instance, we can show that the proposition $A_1 \wedge (A_2 \wedge A_3)$ satisfies the rules of a "trinary" conjunction, i.e., for each of $i = 1, 2, 3$ we can show the rule

$$\dfrac{\Gamma \vdash A_1 \wedge (A_2 \wedge A_3)}{\Gamma \vdash A_i}$$

is admissible by simply composing two elimination forms. These derivable rules are stable under extension: no matter what additional rules are added, a derivable rule will remain admissible.

On the other hand, some admissible rules are *not* derivable, and so might not remain admissible in the presence of other rules. These rules are only admissible *because* many of the rules work in concert to ensure admissibility. The most well known of these are the *structural rules*:

1. The *exchange* rule states that the *order* of propositions doesn't matter:

$$\frac{\Gamma, A, B, \Delta \vdash C}{\Gamma, B, A, \Delta \vdash C}\ \textsc{Exchange}$$

2. The *weakening* rules states that adding assumptions only makes proving a proposition *easier*:

$$\frac{\Gamma \vdash C}{\Gamma, A \vdash C}\ \textsc{Weakening}$$

   That is if we can prove something without an assumption we can also prove it with an additional assumption.

3. The *contraction* rule states that adding an assumption you already have doesn't affect provability:

$$\frac{\Gamma, A, A \vdash B}{\Gamma, A \vdash B}\ \textsc{Contraction}$$

These are not derivable rules. Instead, we can prove these are admissible by showing that by inspecting a proof of the premise (above the line) we can constrcut a proof of the conclusion (below).

Lemma 1. *In IPL the exchange rule is admissible.*

Proof. Given a proof of $\Gamma, A, B, \Delta \vdash C$ we seek to construct a proof of $\Gamma, B, A, \Delta \vdash C$. We proceed by induction on the structure of the proof. We show two cases, the remainder are left as an exercise.

- If $\Gamma, A, B, \Delta \vdash C$ is proven by an assumption, then C is either in Γ, Δ or equal to A or B. In any case, $\Gamma, B, A, \Delta \vdash C$ is also provable by assumption.
- If $\Gamma, A, B, \Delta \vdash C$ is proven by the substitution principle, then we have sub-proofs $\Gamma, A, B, \Delta \vdash C'$ and $\Gamma, A, B, \Delta, C' \vdash C$. Then we can inductively apply the exchange proof to the two sub-proofs and apply the substitution principle to the resulting proofs.

□

Additionally, in IPL, the substitution principle can also be an admissible taken as an admissible rule:

Theorem 1. *In IPL without the substitution principle as a primitive rule, the substitution principle is admissible.*

Proof. Exercise. Hint: you may need a stronger inductive hypothesis. □

2.2 Invertible Rules

What is the logic behind the definitions of the connectives? Why did we give them the particular rules that we did? We explained them in terms of introduction and elimination rules, but another way to think about them is that each connective is defined by an *invertible rule*. That is, for each connective, there is an associated admissible rule where the provability of the conclusion implies the provability of all of the premises.

For instance, the introduction rule for conjunction

$$\frac{\Gamma \vdash A \qquad \Gamma \vdash B}{\Gamma \vdash A \wedge B}\,\wedge\mathrm{I}$$

is invertible: if we know $\Gamma \vdash A \wedge B$ is provable then we can show both of the premises $\Gamma \vdash A$ and $\Gamma \vdash B$ are provable. We write an invertible rule with a double line only to indicate that it is reversible. In fact this is exactly what the two elimination rules do. We could say that the introduction rule is *precisely* an "inverse" to the pair of elimination rules. Similarly, the introduction rule for truth is (trivially) invertible: You can always prove $\Gamma \vdash \top$ just as it is trivial to provide a proof for all of its 0

The dual disjunction connectives instead have invertible rules for *eliminating* the connective:

$$\frac{\Gamma, A \vdash C \qquad \Gamma, B \vdash C}{\Gamma, A \vee B \vdash C}$$

That is, if we have an assumption of a disjunction, we can prove the judgment if and only if we can prove it by cases. We can show the inverse is admisible by using the substitution principle and the disjunction introduction rules. The false proposition is a dual to the true proposition, with falsehood on the left:

$$\frac{}{\Gamma, \bot \vdash C}$$

For the implication rule, the introduction rule is invertible:

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \supset B}$$

This principle has applications in automated proof search algorithms: if a rule is derivable, then it is always safe to use it. So when proving a propositional logic judgment, it is a sound strategy to apply as many $\wedge, \top, \supset$ introduction rules and $\bot, \vee$ elimination rules to assumptions as you can: since these rules are invertible, if the original judgment is provable you will never get stuck with an unprovable subproof. On the other hand if you are proving a disjunction $A \vee B$, then choosing to use a particular *introduction* rule is not invertible, for instance when proving $P \vee (P \supset \bot)$ at most one of the two will be provable.

2.3 Intuitionistic vs Classical Logic

If you have prior experience with formal logic, the connectives and rules we have used are probably different from what you've seen before. That is because we are using *intutionistic* propositional logic, which is less common than *classical* logic. One notable difference is that in classical logic, implication is not typically a primitive connective, but rather *negation* is $\neg A$ and implication is defined as $A \supset_{\text{classical}} B = (\neg A) \vee B$. In intuitionistic logic, we take implication as primitive and instead define intuitionistic negation as $\neg A = A \supset \bot$, i.e., A implies false. Classically, this is equivalent to negation, but in the intuitionistic system, negation does not behave the same as the classical negation. In particular, the following principle, the *law of excluded middle* is *not* admissible in IPL:

$$\Gamma \vdash A \vee (\neg A)$$

an equivalent principle is the *double negation elimination* principle that ensures that $\neg\neg A$ is equi-derivable from A[4]:

$$\Gamma \vdash (\neg(\neg A)) \supset A$$

The *philosophical* reason for this is that intuitionistic logic is *constructive*: a proof of a proposition should construct explicit evidence of its truth. In particular, when proving a disjunction, we must actually identify which one is true. From this perspective the law of excluded middle could be called the "principle of omniscience": it tells us no matter what proposition A we have, we can show that it is true or false.

If the law of excluded middle is added as a principle to IPL, then the system is equivalent to the usual presentations of classical propositional logic.

2.4 Takeaway Questions

When designing a new logical system, say for formal verification of computer programs, it is of critical importance that we establish that the system is *consistent*, i.e., that we cannot prove falsehood under no assumptions. That is, that the judgment

$$\cdot \vdash \bot$$

is not provable. If this were true, then we would be able to prove any judgment

$$\dfrac{\dfrac{\cdot \vdash \bot}{\Gamma \vdash \bot}\ \textsc{Wkn}}{\Gamma \vdash A}\ \bot\text{E}$$

and so knowing that a judgment is provable in such a logic would provide no information.

For next time, think about how you might attempt to prove

[4]Exercise: the opposite $\Gamma \vdash A \supset (\neg(\neg A))$ is provable in IPL

1. That IPL is consistent, i.e., that $\cdot \vdash \bot$ is not provable.
2. That the law of excluded middle is not generally valid. I.e., for a propositional variable X that the judgment $\cdot \vdash X \vee \neg X$ is not provable.