Software for Constructing Proofs in Natural Deduction Systems

Koissi Adjorlolo
Advisor: Prof. Branden Fitelson

April 2020

Abstract

For my Junior/Senior honors thesis, I looked into how the educational proof assistant **MacLogic** [3] works and implemented its proof construction mode using the Scala programming language. This is a continuation of the "Coastline" project, where I've been developing non-traditional interfaces which allow students to solve a variety of math problems on their computers. This paper defines the set of rules within MacLogic which allow users to construct proofs in intuitionistic sentential logic.

1 Introduction

I started looking into software for helping students learn mathematics in 2017 when I got a job as an online math and science tutor. The job required that I walk a student through algebraic manipulations or proofs by using a chat window or pulling up a collaborative online whiteboard. Neither medium was sufficient, as I was forced to tediously rewrite each step of the manipulation or proof by hand. I would then ask the student to walk me through the same steps to ensure they understood what I had done. They found this experience more difficult than I did, since chat windows and online whiteboards are not sufficient platforms for doing math homework. I was inspired from this experience to start working on **Coastline** [1] [2], a project which began as an attempt to develop an interface that allows students to manually solve high school algebra problems on their computers. The scope of the project has since expanded to developing an interface which will allows students to construct various kinds of mathematical proofs.

With this goal in mind, I spent this semester learning about software which enables the construction of proofs in various logical systems. I focused on **MacLogic** [3], a Classic Macintosh application which teaches students about natural deduction proofs. Along with this thesis, I implemented a prototype application which allows users to construct proofs in a similar manner to MacLogic. This paper explains some of the concepts required to use the software that I developed, focusing on the tactic system in MacLogic's construction mode.

I begin by defining the **Language of Sentential Logic** in terms of **judgements**, **rules**, and **derivations**. These concepts are later used to define rules of inference in Natural Deduction and tactics in MacLogic.

2 Language of Sentential Logic

The **Language of Sentential Logic** (LSL) is a formal language made up of strings representing logical structure within sentential logic [4]. Sentential logic concerns itself with the most basic of logical forms and doesn't include strings containing predicates or quantifiers. These omissions limit the expressive power of the system while allowing me to focus on the complexities of proof construction.

Strings in LSL will be referred to as **propositions**. A proposition is any statement that is associated with a truth value. Propositions are built up from **sentence-letters** and **propositional connectives**.

2.1 Sentence-Letters

A **sentence-letter** is an italicized uppercase letter of the english alphabet, e.g. A, B, ..., Z. There should be an infinite number of sentence-letters to choose from, but none of the examples presented here are complex enough to exhaust our options.

Sentence-letters are used so symbolize sentences within a natural language that are associated with truth values. For example, we could use the sentence-letter W to symbolize the English sentence "You should wash your hands for at least 20 seconds". Then, W is associated with the truth value *true*, which will be symbolized in LSL as $\top$. The English sentence "You should leave your apartment" could then be symbolized with the sentence-letter L. L would then be associated with the truth value *false*, which will be symbolized in LSL as $\bot$.

These sentence-letters can then be joined using connectives in order to symbolize more complicated sentences.

2.2 Propositional Connectives

The propositional connectives discussed in this paper are written as $\wedge$ (and/conjunction), $\rightarrow$ (conditional/implication), $\neg$ (not/negation), and $\vee$ (or/disjunction). We can then symbolize the following sentence using the sentence-letters W and L along with the connectives $\wedge$ and $\neg$.

"You should wash your hands for at least 20 seconds and you should not leave your apartment"

$$W \wedge (\neg L)$$

Formal languages such as LSL are often defined using grammars. I will instead define LSL using **judgements** and **formation rules** in order to illustrate ideas that will be used throughout this paper.

2.3 Judgement: ϕ *prop*

In logic, a **judgement** represents something that can be known [5]. The simplest of judgements presented here says that an object ϕ is a proposition within LSL. Such judgement will be presented as follows.

$$\phi \textit{ prop}$$

The object ϕ does not have to be a proposition in order for the judgement ϕ *prop* to be spoken of. For example, the number 32 is not associated with a truth value, but the judgement 32 *prop* can still be discussed. However, 32 *prop* cannot be known.

The only way to know a judgement is to have evidence for it. The construction of such evidence is defined in terms of a set of **rules**. The rules presented in the next section define the ways in which the judgement ϕ *prop* can be known.

2.4 Formation Rules

A **rule** is anything of the following form, where $J_1, J_2, ..., J_n$ are judgements referred to as the rule's **premises** and J is a judgement referred to as the rule's **conclusion**.

$$\frac{J_1 \qquad J_2 \qquad ... \qquad J_n}{J}\ \text{Rule Name}$$

A **formation rule** is a rule where $J_1, J_2, ..., J_n$ and J are all judgements of the form ϕ *prop* [5]. This section will define the set LSL_F of formation rules used to determine what objects are contained within LSL. LSL will then be defined as a set of objects that can be formed using LSL_F.

2.4.1 Sentence Letter Formation (SL-F)

The most basic rules in any system are without premises, and are used to define judgements that are known without requiring any evidence for them. In LSL_F, those rules are the formation rules for sentence-letters. These rules will each be labelled with the name SL-F, but each rule is distinct.

$$\frac{}{A\ prop}\ \text{SL-F} \qquad \frac{}{B\ prop}\ \text{SL-F} \qquad \cdots \qquad \frac{}{Z\ prop}\ \text{SL-F}$$

Each one of the above formation rules says that we can know the sentence-letters A, B, etc. to be propositions without requiring evidence.

2.4.2 Formation rules for $\wedge$, $\rightarrow$, $\vee$, and $\neg$ ($\wedge$F)

More complex propositions can be formed by applying formation rules for the propositional connectives. One of these rules says that knowledge of α *prop* along with knowledge of β *prop* allows one to know $\alpha \wedge \beta$ *prop*.

$$\frac{\alpha\ prop \qquad \beta\ prop}{\alpha \wedge \beta\ prop}\ \wedge\text{F}$$

Formation rules for the other infix connectives take on a similar form, requiring knowledge of two other propositions to form a third proposition. This third proposition contains the connective associated with the formation rule used.

$$\frac{\alpha\ prop \qquad \beta\ prop}{\alpha \rightarrow \beta\ prop}\ \rightarrow\text{F} \qquad \frac{\alpha\ prop \qquad \beta\ prop}{\alpha \vee \beta\ prop}\ \vee\text{F}$$

Forming a proposition of the form $\neg\alpha$ from a proposition α only requires the judgement α *prop* to be known.

$$\frac{\alpha\ prop}{\neg\alpha\ prop}\ \neg\text{F}$$

The set LSL_F is defined as follows.

$$LSL_F = \{\, \text{SL-F}, \wedge\text{F}, \rightarrow\text{F}, \vee\text{F}, \neg\text{F} \,\}$$

As stated above, these formation rules are combined in order to know the judgement ϕ *prop*, but the result of that combination has not yet been defined.

2.5 Derivations

A **derivation** acts as sufficient evidence for knowing a judgement [6]. Derivations look like trees, whose leaves are the formation rules that don't rely on premises, and whose root is the judgement we are trying to know. For example, constructing the following derivation allows us to know $W \wedge (\neg L)$ *prop*.

$$\dfrac{\dfrac{}{W\ prop}\text{SL-F} \qquad \dfrac{\dfrac{}{L\ prop}\text{SL-F}}{\neg L\ prop}\neg\text{F}}{W \wedge (\neg L)\ prop}\wedge\text{F}$$

The **main connective** of a proposition ϕ is the connective corresponding to the last formation rule used in the derivation of ϕ *prop*. The last rule used to derive $W \wedge (\neg L)$ *prop* is $\wedge$F. Therefore, $\wedge$ is the main connective in $W \wedge (\neg L)$. Equivalently, connectives that aren't surrounded in parenthesis will also be considered as main connectives, which excludes $\neg$ from being the main connective of $W \wedge (\neg L)$.

With all of the above definitions in place, we can now define LSL formally. The set of objects in LSL is defined as the set LSL of objects ϕ for which the formation rules of LSL_F can be used to derive ϕ *prop*.

$$LSL = \{\, \phi \mid \phi\, prop \text{ can be derived from } LSL_F \,\}$$

If ϕ is an element of LSL, then ϕ can be associated with a truth value, but this doesn't tell us what ϕ's truth value is. The next section defines a proof system in which we can construct evidence that allows us to know that a proposition is true in intuitionistic logic.

3 Natural Deduction and NJ

NJ is a proof system developed by Gerhard Gentzen. It is a kind of **natural deduction** system. Natural deduction systems attempt to formalize how mathematicians "naturally" reason. It is used to construct valid proofs within intuitionistic logic. [4]

NJ will be defined in the same way LSL was defined in the previous section. The only difference between this section and the last is in the judgements being derived and the rules used to derive them. The words "proof" and "derivation" will be used interchangeably.

3.1 Judgements: ϕ *prop*, ϕ *true*, and, $J_1, J_2, ..., J_n \vdash J$

Derivations in NJ form evidence to know certain kinds of judgements. The traditional kind of judgement used in NJ takes the following form for some object ϕ.

$$\phi\ true$$

An object must be associated with a truth value before its truth value can be determined. Therefore, evidence of ϕ *true* must in part be constructed from evidence of ϕ *prop*. This requires that the set of rules which derive ϕ *true* include the formation rules in LSL_F.

Reasoning from assumptions is a concept common to all natural deduction systems, and NJ is no different. This paper will use a special kind of judgement called a **sequent** in order to capture this kind of reasoning.

3.1.1 Sequents

A **sequent** is a judgement of the following form, where $J_1, J_2, ..., J_n$, and J are all judgements of the form ϕ *true* for some object ϕ. [4]

$$J_1, J_2, ..., J_n \vdash J$$

This sequent is read as "J is provable from $J_1, J_2, ..., J_n$". A known sequent is said to be **valid**.

The symbol $\vdash$ will be referred to as the **turnstile**. Everything to the left of the turnstile is an unproven **assumption**, and the judgement to the right of the turnstile is the **conclusion**. For example, the following sequent is assuming A *true* and B *true* to conclude $A \wedge B$ *true*.

$$A\ true, B\ true \vdash A \wedge B\ true$$

Judgements within sequents will be presented without the word *true* in each judgement, so the above sequent and the following sequent will be considered equivalent.

$$A, B \vdash A \wedge B$$

Sequents don't need to include any assumptions. Such sequents are used to express tautologies.

$$\vdash A \to A$$

The variables $\Gamma, \Gamma', \Gamma''$, etc. will be used to represent a list of judgements. Therefore, the sequent $J_1, J_2, ..., J_n \vdash J$ can be represented as follows, where $\Gamma = J_1, J_2, ..., J_n$.

$$\Gamma \vdash J$$

We will at times surround a single assumption α with these variables to show that α can occur anywhere in the list of assumptions.

$$\Gamma, \alpha, \Gamma' \vdash J$$

The non-formation rules in NJ are split up into **structural rules**, **introduction rules**, and **elimination rules**.

3.2 Structural Rules

Structural rules define the ways in which we can rearrange judgements within sequents. Their inclusion allows us to think of a sequent's list of assumptions as a set of assumptions. This will in turn simplify the definitions of the non-structural rules for NJ. [7]

Exchange allows us to rearrange two assumptions to derive another sequent.

$$\frac{\Gamma, \alpha, \beta, \Gamma' \vdash \phi}{\Gamma, \beta, \alpha, \Gamma' \vdash \phi}\ \mathrm{E}$$

This rule removes the need to surround assumptions with Γ and Γ'.

Weakening allows us to add arbitrary assumptions to a valid sequent so that it remains valid.

$$\frac{\Gamma, \alpha \vdash \phi}{\Gamma, \beta, \alpha \vdash \phi}\ \mathrm{W}$$

It says that a proof of the proposition ϕ from a list of assumptions Γ along with α means that we can also prove ϕ by assuming α and β.

Contraction allows us to remove duplicate assumptions from a proof.

$$\frac{\Gamma, \alpha, \alpha \vdash \phi}{\Gamma, \alpha, \vdash \phi}\ \mathrm{C}$$

These rules allow the sequent $\Gamma \vdash \phi$ to stand in for the following sequents.

$$\vdash A \to A$$
$$A \wedge B, A \wedge B \vdash B$$

Furthermore, the sequent $\Gamma, \alpha, \Gamma,' \beta, \Gamma'' \vdash \phi$ can now stand in for the following sequents.

$$\alpha, \beta \vdash \phi$$
$$\beta, A, B, \alpha \vdash \phi$$

These structural rules must be kept in mind when constructing derivations as some logical systems exclude them or add others. However, derivations presented within this paper will apply these rules implicitly.

3.3 Making Assumptions

Non-structural rules within NJ are called **rules of inference**, since they are used to infer a sequent's validity [4]. The first such rule of inference corresponds to making an assumption that will be accounted for later in a derivation. It says that for any proposition ϕ, the sequent $\phi \vdash \phi$ can be inferred.

$$\frac{\phi\ \mathit{prop}}{\phi \vdash \phi}\ \text{Assume}$$

This sequent is valid because the truth of a proposition ϕ can be proved trivially from the truth of ϕ.

The name "Assume" as well as the premise ϕ *prop* will be removed when presenting derivations, as their inclusion makes reading derivations cumbersome. All other rules of inference presented here will include their names when used within derivations.

$$\frac{}{\phi \vdash \phi}$$

These exclusions make Assume a premise-less rule so that it can occur as a leaf in a derivation.

3.4 Introduction and Elimination Rules

The remaining rules in NJ can be split up into **introduction** and **elimination** rules. Introduction rules for a propositional connective $*$ tell us when we can derive a sequent of the form $\Gamma \vdash \phi$, where ϕ is a proposition whose main connective is $*$. Conversely, elimination rules for $*$ tell us what we can use sequents of the form $\Gamma \vdash \phi$ to derive. Therefore, each propositional connective has a corresponding set of introduction and elimination rules.

3.4.1 And

And introduction, presented as $\wedge$I, has the following form.

$$\frac{\alpha\ \mathit{prop} \qquad \beta\ \mathit{prop} \qquad \Gamma \vdash \alpha \qquad \Gamma' \vdash \beta}{\Gamma, \Gamma' \vdash \alpha \wedge \beta}\ \wedge\mathrm{I}$$

It says that for the propositions α and β, the valid sequent $\Gamma \vdash \alpha$ can be combined with the valid sequent $\Gamma \vdash \beta$ to infer a sequent which retains Γ and Γ' and concludes $\alpha \wedge \beta$. Like for Assume, the judgements α *prop* and β *prop* will be taken as implicit and excluded from this rule of inference and the ones yet to be defined.

$$\frac{\Gamma \vdash \alpha \qquad \Gamma' \vdash \beta}{\Gamma, \Gamma' \vdash \alpha \wedge \beta}\ \wedge\mathrm{I}$$

There are two rules for **and elimination**, presented as $\wedge\mathrm{E}_{\mathrm{left}}$ and $\wedge\mathrm{E}_{\mathrm{right}}$. The former rule says that if $\alpha \wedge \beta$ is true for a given set of assumptions Γ, then we can infer that α must also be true. The latter rule is of a similar form, but instead infers the truth of β from Γ.

$$\frac{\Gamma \vdash \alpha \wedge \beta}{\Gamma \vdash \alpha}\ \wedge\mathrm{E}_{\mathrm{left}} \qquad \frac{\Gamma \vdash \alpha \wedge \beta}{\Gamma \vdash \beta}\ \wedge\mathrm{E}_{\mathrm{right}}$$

We can use the previously defined rules to derive the sequent $A \wedge B \vdash B \wedge A$. As previously mentioned, we will assume that ϕ *prop* has been derived for all propositions presented in NJ.

Natural deduction proofs can be constructed from the **top-down**, from the **bottom-up**, or in both directions. Top-down proof construction starts from the assumptions and applies rules until the desired conclusion is reached. The sequent $A \wedge B \vdash B \wedge A$ will be derived using the top-down approach. First, we make our assumption.

$$\begin{array}{c} \dfrac{}{A \wedge B \vdash A \wedge B} \\ \vdots \end{array}$$

Next, we'll use our assumption to conclude the truth of both B and A.

$$\begin{array}{c} \dfrac{\dfrac{}{A \wedge B \vdash A \wedge B}}{A \wedge B \vdash B}\ \wedge\mathrm{E}_{\mathrm{right}} \\ \vdots \end{array} \qquad \begin{array}{c} \dfrac{\dfrac{}{A \wedge B \vdash A \wedge B}}{A \wedge B \vdash A}\ \wedge\mathrm{E}_{\mathrm{left}} \\ \vdots \end{array}$$

Finally, our two derivations are combined to conclude the validity of $A \wedge B \vdash B \wedge A$.

$$\dfrac{\dfrac{\dfrac{}{A \wedge B \vdash A \wedge B}}{A \wedge B \vdash B}\ \wedge\mathrm{E}_{\mathrm{right}} \qquad \dfrac{\dfrac{}{A \wedge B \vdash A \wedge B}}{A \wedge B \vdash A}\ \wedge\mathrm{E}_{\mathrm{left}}}{A \wedge B \vdash B \wedge A}\ \wedge\mathrm{I}$$

3.4.2 Implication

The introduction rule for $\to$ introduces us to the notion of **discharging** assumptions. A rule of inference discharges an assumption when it removes an assumption from a sequent's left hand side. **Implication introduction**, presented as $\to$I, discharges the assumption α while deriving $\alpha \to \beta$.

$$\frac{\Gamma, \alpha \vdash \beta}{\Gamma \vdash \alpha \to \beta}\ \to\mathrm{I}$$

This gives us the power to derive the valid sequent $\vdash A \to A$ in two steps. We start by assuming the truth of A.

$$\begin{array}{c} \dfrac{}{A \vdash A} \\ \vdots \end{array}$$

Then we apply $\to$I, discharging our assumption in the process.

$$\dfrac{\dfrac{}{A \vdash A}}{\vdash A \to A}\ \to\mathrm{I}$$

This completes our derivation.

Some presentations of discharging rules of inference label the assumptions being discharged upon the rule's application. The derivations in this paper will not be labelling discharged assumptions. For example, take the following derivation.

$$\dfrac{\begin{matrix}\vdots\\ A, B, C \vdash A \wedge B\end{matrix}}{A, C \vdash B \to (B \wedge C)}\ {\to}\mathrm{I}$$

It is clear that B is being discharged by the fact that B no longer appears in the derived sequent's list of assumptions.

Implication elimination, presented as $\to$E, says that given the truth of $\alpha \to \beta$ from a set of assumptions Γ, and the truth of α from a set of assumptions Γ', we can infer the truth of β from the assumptions in Γ and the assumptions in Γ'.

$$\dfrac{\Gamma \vdash \alpha \to \beta \qquad \Gamma' \vdash \alpha}{\Gamma, \Gamma' \vdash \beta}\ {\to}\mathrm{E}$$

The following proof uses both $\to$I and $\to$E to derive the validity of $R \to (S \to T), S \vdash R \to T$. Again, we start by making our assumptions.

$$\begin{matrix}\dfrac{}{R \to (S \to T) \vdash R \to (S \to T)} & \qquad & \dfrac{}{S \vdash S}\\ \vdots & & \vdots\end{matrix}$$

We can then apply $\to$E to the sequent $R \to (S \to T) \vdash R \to (S \to T)$ along with the new assumption $R \vdash R$ to derive a sequent whose conclusion is $S \to T$.

$$\begin{matrix}\dfrac{\dfrac{}{R \to (S \to T) \vdash R \to (S \to T)} \qquad \dfrac{}{R \vdash R}}{R \to (S \to T), R \vdash S \to T}\ {\to}\mathrm{E} & \qquad & \dfrac{}{S \vdash S}\\ \vdots & & \vdots\end{matrix}$$

By deriving $R \vdash R$ using Assume, R is added to the list of assumptions in our derived sequent. R is not present in the list of assumptions of our goal sequent, so we must discharge it with a rule of inference before our derivation is complete.

Another application of $\to$E gives us a sequent whose conclusion is just T.

$$\begin{matrix}\dfrac{\dfrac{\dfrac{}{R \to (S \to T) \vdash R \to (S \to T)} \qquad \dfrac{}{R \vdash R}}{R \to (S \to T), R \vdash S \to T}\ {\to}\mathrm{E} \qquad \dfrac{}{S \vdash S}}{R \to (S \to T), R, S \vdash T}\ {\to}\mathrm{E}\\ \vdots\end{matrix}$$

Finally, we'll apply $\to$I to discharge R from our list of assumptions and derive our goal.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{R \to (S \to T) \vdash R \to (S \to T)} \qquad \dfrac{}{R \vdash R}}{R \to (S \to T), R \vdash S \to T}\ {\to}\mathrm{E} \qquad \dfrac{}{S \vdash S}}{R \to (S \to T), R, S \vdash T}\ {\to}\mathrm{E}}{R \to (S \to T), S \vdash R \to T}\ {\to}\mathrm{I}$$

Implication introduction is the first of three discharging rules of inference in NJ.

3.4.3 Not

Not introduction, presented as ¬I, allows us to derive the negation of a proposition α from a set of assumptions Γ if we are able to derive falsehood by assuming Γ and α.

$$\frac{\Gamma, \alpha \vdash \bot}{\Gamma \vdash \neg\alpha}\neg\mathrm{I}$$

We can't use ¬I until we have a rule of inference that allows us to derive a sequent concluding $\bot$. **Not elimination**, presented as ¬E, allows us to do this if we first derive a proposition α and its negation. Then, $\neg\alpha$ and α would form a contradiction, and anything can be derived from a contradiction, including the false proposition $\bot$.

$$\frac{\Gamma \vdash \neg\alpha \qquad \Gamma' \vdash \alpha}{\Gamma, \Gamma' \vdash \bot}\neg\mathrm{E}$$

These rules will be illustrated in the following derivation of the sequent $A \to B, \neg B \vdash \neg A$. This time, we will construct our derivation from the **bottom-up**, starting with what we are trying to derive and working backwards to construct our derivation.

$$\begin{array}{c} \vdots \\ A \to B, \neg B \vdash \neg A \end{array}$$

First, we'll apply ¬I, requiring us to then derive $\bot$ from our assumptions and A.

$$\frac{\begin{array}{c} \vdots \\ A \to B, \neg B, A \vdash \bot \end{array}}{A \to B, \neg B \vdash \neg A}\neg\mathrm{I}$$

Deriving $\bot$ requires us to first derive a contradiction. We already have $\neg B$ in our list of assumptions, and the presence of $A \to B$ and A suggests that we can eventually use →E to derive B. We need to keep this strategy in mind as we apply ¬E.

$$\frac{\dfrac{\dfrac{}{\neg B \vdash \neg B} \qquad \begin{array}{c} \vdots \\ A \to B, A \vdash B \end{array}}{A \to B, \neg B, A \vdash \bot}\neg\mathrm{E}}{A \to B, \neg B \vdash \neg A}\neg\mathrm{I}$$

The proposition $\neg B$ appears in the left hand side of our goal sequent, so we should derive $\neg B \vdash \neg B$ and remember not to discharge it. It is therefore safe to assume that the sequent concluding B will not need $\neg B$ as an assumption, so our new subgoal, $A \to B, A \vdash B$, does not include $\neg B$ as an assumption.

Our derivation is complete once we apply →E to derive the remaining subgoal.

$$\frac{\dfrac{\dfrac{}{\neg B \vdash \neg B} \qquad \dfrac{\dfrac{}{A \to B \vdash A \to B} \qquad \dfrac{}{A \vdash A}}{A \to B, A \vdash B}\to\mathrm{E}}{A \to B, \neg B, A \vdash \bot}\neg\mathrm{E}}{A \to B, \neg B \vdash \neg A}\neg\mathrm{I}$$

3.4.4 Or

There are two rules for **or introduction**. The first says that the truth of $\alpha \vee \beta$ can be concluded from a set of assumptions Γ if α can be proven from Γ. This rule is presented as $\vee\mathrm{I}_{\mathrm{left}}$. The second says the same for β and is presented as $\vee\mathrm{I}_{\mathrm{right}}$.

$$\dfrac{\Gamma \vdash \alpha}{\Gamma \vdash \alpha \vee \beta}\,\vee\mathrm{I}_{\mathrm{left}} \qquad\qquad \dfrac{\Gamma \vdash \beta}{\Gamma \vdash \alpha \vee \beta}\,\vee\mathrm{I}_{\mathrm{right}}$$

The rule for **or elimination**, presented as $\vee$E, is the most complicated of the rules in this presentation of NJ. It says that the truth of a proposition ϕ can be proved from the proposition $\alpha \vee \beta$ if we can prove ϕ from α, and separately derive ϕ from β. The assumptions for which the truth of these propositions depend on without α and β are then combined to form the assumptions which prove ϕ.

$$\dfrac{\Gamma \vdash \alpha \vee \beta \qquad \Gamma', \alpha \vdash \phi \qquad \Gamma'', \beta \vdash \phi}{\Gamma, \Gamma', \Gamma'' \vdash \phi}\,\vee\mathrm{E}$$

The following derivation of $A \vee B \vdash B \vee A$ illustrates how these rules might be used. It will be constructed both from the top-down and from the bottom-up. We'll start by making our first assumption and writing down our desired goal.

$$\begin{array}{c} \dfrac{}{A \vee B \vdash A \vee B} \\ \vdots \end{array} \qquad\qquad \begin{array}{c} \vdots \\ A \vee B \vdash B \vee A \end{array}$$

We'll then derive our goal by applying $\vee$E.

$$\dfrac{\dfrac{}{A \vee B \vdash A \vee B} \qquad \begin{array}{c}\vdots \\ A \vdash B \vee A\end{array} \qquad \begin{array}{c}\vdots \\ B \vdash B \vee A\end{array}}{A \vee B \vdash B \vee A}\,\vee\mathrm{E}$$

The inclusion of A and B as assumptions to be discharged tells us that we need to assume A and B within the derivations of $A \vdash B \vee A$ and $B \vdash B \vee A$, so we'll make those assumptions now.

$$\dfrac{\dfrac{}{A \vee B \vdash A \vee B} \qquad \begin{array}{c}\dfrac{}{A \vdash A} \\ \vdots \\ A \vdash B \vee A\end{array} \qquad \begin{array}{c}\dfrac{}{B \vdash B} \\ \vdots \\ B \vdash B \vee A\end{array}}{A \vee B \vdash B \vee A}\,\vee\mathrm{E}$$

Our subgoals can be derived by applying $\vee\mathrm{I}_{\mathrm{left}}$ and $\vee\mathrm{I}_{\mathrm{right}}$ to the sequents $A \vdash A$ and $B \vdash B$ respectively. This completes the derivation.

$$\dfrac{\dfrac{}{A \vee B \vdash A \vee B} \qquad \dfrac{\dfrac{}{A \vdash A}}{A \vdash B \vee A}\,\vee\mathrm{I}_{\mathrm{right}} \qquad \dfrac{\dfrac{}{B \vdash B}}{B \vdash B \vee A}\,\vee\mathrm{I}_{\mathrm{left}}}{A \vee B \vdash B \vee A}\,\vee\mathrm{E}$$

This paper will use the set NJ_{ROI} to represent the collection of the rules of inference defined above.

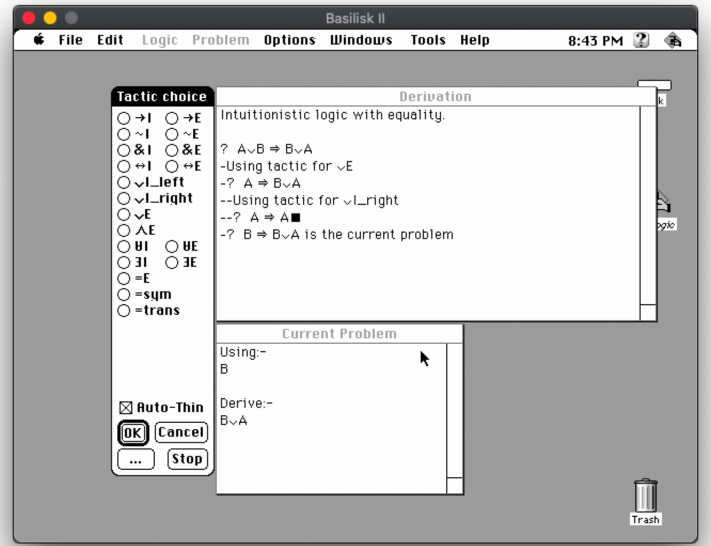
$$NJ_{ROI} = \{\, \wedge\mathrm{I}, \wedge\mathrm{E}_{\mathrm{left}}, \wedge\mathrm{E}_{\mathrm{right}}, \rightarrow\mathrm{I}, \rightarrow\mathrm{E}, \neg\mathrm{I}, \neg\mathrm{E}, \vee\mathrm{I}_{\mathrm{left}}, \vee\mathrm{I}_{\mathrm{right}}, \vee\mathrm{E} \,\}$$

NJ can then be defined as the set NJ, defined as the union of LSL_F, SR, and NJ_{ROI}.

$$NJ = LSL_F \cup SR \cup NJ_{ROI}$$

4 MacLogic and ML

MacLogic is a computer program developed by Roy Dyckhoff et al. for teaching and learning how to do proofs in natural deduction [3]. Professor Branden Fitelson says that it is the "...best computer program in existence for teaching and learning natural deduction proofs". I spent part of this last semester learning about how it works, and am beginning to understand why Professor Fitelson likes it so much. The biggest downside to the program is that it was originally developed for Classic Macintosh and requires an emulator to run.



I focused primarily on MacLogic's construction mode, which allows users to build derivations in a handful of proof systems, including one similar to NJ. Derivations are built using **tactics**, which are selected from the window titled "Tactic Choices". Each tactic corresponds to a rule of inference in NJ. After a tactic is selected and the "OK" button is pressed, MacLogic applies the tactic to the **current problem**, which is displayed in the window titled "Current Problem". The entire derivation is displayed in the window titled "Derivation" in an outline format.

Speaking from personal experience, MacLogic made learning how to construct proofs in natural deduction fun. I attribute the joy I experienced from using the software to its central interaction loop, which has the user select their desired tactic, apply the tactic to the current problem, and repeating the process. This loop requires a very low barrier of entry in terms of learning how to use the interface, which frees up cognitive power to learn how the proofs are built. For these reasons and others I determined that my ongoing development of Coastline would benefit from a reimplementation of MacLogic's construction mode.

4.1 NJ_T vs. NJ

The tactics in MacLogic's construction mode simplify building derivations from both the top-down and the bottom-up. In this section I will be defining NJ_T, the set of tactics MacLogic uses in its construction mode which correspond to the rules of inference in NJ.

4.1.1 Tactic for Making Assumptions

Users of MacLogic's construction mode aren't required to explicitly make assumptions. Instead, MacLogic automatically moves on from a subgoal once a sequent of the following form is derived.

$$\Gamma, \alpha \vdash \alpha$$

This sequent can be derived from the rule of inference Assume and then weakening the resulting sequent.

Derivations occurring in this section will use the following rule in place of Assume, presented as Assume^T.

$$\frac{}{\Gamma, \alpha \vdash \alpha}$$

4.1.2 Tactics for $\wedge$

The tactic corresponding to $\wedge$I, presented as $\wedge \mathrm{I}^T$, is almost identical to $\wedge$I. The only difference is in how the assumptions in each sequent are related.

$$\frac{\Gamma \vdash \alpha \qquad \Gamma' \vdash \beta}{\Gamma, \Gamma' \vdash \alpha \wedge \beta} \wedge\mathrm{I} \qquad \Bigg| \qquad \frac{\Gamma \vdash \alpha \qquad \Gamma \vdash \beta}{\Gamma \vdash \alpha \wedge \beta} \wedge\mathrm{I}^T$$

As shown above, $\wedge$I deals with two different set of assumptions: Γ and Γ'. These sets are then combined if the derivation is being built from the top down. If the derivation is built from the bottom-up, a set of assumptions Γ is split into Γ and Γ'. On the other hand, $\wedge \mathrm{I}^T$ requires that the set of assumptions among its sequents are all the same. This simplifies things for MacLogic users when reasoning from the bottom-up, since the set of assumptions they have when trying to derive the sequent concluding $\alpha \wedge \beta$ doesn't change.

The tactic corresponding to to NJ's $\wedge$E rules, presented as $\wedge \mathrm{E}^T$, combines $\wedge \mathrm{E}_{\mathrm{left}}$ and $\wedge \mathrm{E}_{\mathrm{right}}$ into one.

$$\frac{\Gamma \vdash \alpha \wedge \beta}{\Gamma \vdash \alpha} \wedge\mathrm{E}_1 \qquad \frac{\Gamma \vdash \alpha \wedge \beta}{\Gamma \vdash \beta} \wedge\mathrm{E}_2 \qquad \Bigg| \qquad \frac{\Gamma, \alpha, \beta \vdash \phi}{\Gamma, \alpha \wedge \beta \vdash \phi} \wedge\mathrm{E}^T$$

It says that a proof of the proposition ϕ from the set of assumptions Γ along with $\alpha \wedge \beta$ can be found if a proof of ϕ exists from Γ, α, and β. Like with $\wedge \mathrm{I}^T$, the set of assumptions Γ is preserved across all the sequents involved in $\wedge \mathrm{E}^T$.

The derivation of $A \wedge B \vdash B \wedge A$ in section 3.4.1 was built from the top-down, so we started off by listing the assumptions available to us. Derivations in NJ_T can only be built from the bottom-up, so we will start the following derivation in NJ_T by listing our goal. The derivation using rules from NJ is on the left, and the derivation using rules from NJ_T is on the right.

$$\begin{array}{c} \overline{A \wedge B \vdash A \wedge B} \\ \vdots \\ A \wedge B \vdash B \wedge A \end{array} \qquad \Bigg| \qquad \begin{array}{c} \vdots \\ A \wedge B \vdash B \wedge A \end{array}$$

In NJ, we applied $\wedge \mathrm{E}_{\mathrm{left}}$ and $\wedge \mathrm{E}_{\mathrm{right}}$ to our assumption to derive a sequent concluding B and another sequent concluding A.

$$\begin{array}{c} \dfrac{\overline{A \wedge B \vdash A \wedge B}}{A \wedge B \vdash B} \wedge\mathrm{E}_{\mathrm{right}} \qquad \dfrac{\overline{A \wedge B \vdash A \wedge B}}{A \wedge B \vdash A} \wedge\mathrm{E}_{\mathrm{left}} \\ \vdots \qquad\qquad\qquad\qquad \vdots \\ \vdots \\ A \wedge B \vdash B \wedge A \end{array}$$

The equivalent step in NJ_T only requires one application of $\wedge\mathrm{E}^T$.

$$\dfrac{\begin{array}{c}\vdots\\ A, B \vdash B \wedge A\end{array}}{A \wedge B \vdash B \wedge A}\wedge\mathrm{E}^T$$

We finished the derivation in NJ with an application of $\wedge$I. Similarly, the derivation in NJ_T is completed by applying $\wedge\mathrm{I}^T$.

$$\dfrac{\dfrac{\overline{A \wedge B \vdash A \wedge B}}{A \wedge B \vdash B}\wedge\mathrm{E}_{\text{right}} \qquad \dfrac{\overline{A \wedge B \vdash A \wedge B}}{A \wedge B \vdash A}\wedge\mathrm{E}_{\text{left}}}{A \wedge B \vdash B \wedge A}\wedge\mathrm{I} \qquad\qquad \dfrac{\dfrac{\overline{A, B \vdash B} \qquad \overline{A, B \vdash A}}{A, B \vdash B \wedge A}\wedge\mathrm{I}^T}{A \wedge B \vdash B \wedge A}\wedge\mathrm{E}^T$$

4.1.3 Tactics for $\rightarrow$

The tactic corresponding to implication introduction, presented as $\rightarrow\mathrm{I}^T$, is identical to $\rightarrow$I.

$$\dfrac{\Gamma, \alpha \vdash \beta}{\Gamma \vdash \alpha \rightarrow \beta}\rightarrow\mathrm{I} \qquad\Bigg|\qquad \dfrac{\Gamma, \alpha \vdash \beta}{\Gamma \vdash \alpha \rightarrow \beta}\rightarrow\mathrm{I}^T$$

The tactic corresponding to implication elimination, presented as $\rightarrow\mathrm{E}^T$, requires more of an explanation than previous tactics. It says that if we can prove that α is true from a set of assumptions Γ, and that ϕ can be proved true from Γ along with β, then ϕ can be proven true from Γ and $\alpha \rightarrow \beta$.

$$\dfrac{\Gamma \vdash \alpha \rightarrow \beta \qquad \Gamma' \vdash \alpha}{\Gamma, \Gamma' \vdash \beta}\rightarrow\mathrm{E} \qquad\Bigg|\qquad \dfrac{\Gamma \vdash \alpha \qquad \Gamma, \beta \vdash \phi}{\Gamma, \alpha \rightarrow \beta \vdash \phi}\rightarrow\mathrm{E}^T$$

Comparing derivations of the sequent $R \rightarrow (S \rightarrow T), S \vdash R \rightarrow T$ within NJ and NJ_T should clear up how $\rightarrow$E relates to $\rightarrow\mathrm{E}^T$. The derivation in NJ will be constructed both from the top-down and the bottom-up, while the derivation in NJ_T will be built from the bottom-up. The derivation using rules in NJ is shown on the left, while the derivation using rules from NJ_T is shown on the right.

$$\begin{array}{c}\begin{array}{cc}\overline{R \rightarrow (S \rightarrow T) \vdash R \rightarrow (S \rightarrow T)} & \overline{S \vdash S}\\ \vdots & \vdots\end{array}\\ \vdots\\ R \rightarrow (S \rightarrow T), S \vdash R \rightarrow T\end{array} \qquad\Bigg|\qquad \begin{array}{c}\vdots\\ R \rightarrow (S \rightarrow T), S \vdash R \rightarrow T\end{array}$$

In section 3.4.2 when we first derived our current goal, $\rightarrow$E was the first rule of inference that we applied.

$$\begin{array}{c}\dfrac{\overline{R \rightarrow (S \rightarrow T) \vdash R \rightarrow (S \rightarrow T)} \qquad \overline{R \vdash R}}{R \rightarrow (S \rightarrow T), R \vdash S \rightarrow T}\rightarrow\mathrm{E}\\ \vdots\end{array} \qquad \begin{array}{c}\overline{S \vdash S}\\ \vdots\end{array}$$

This step requires that we know that R will eventually be discharged, but this isn't obvious to a student learning natural deduction for the first time. We could try to mirror this step in NJ by applying $\rightarrow\mathrm{E}^T$ in NJ_T, but will immediately run into a problem.

$$\dfrac{\begin{array}{c}\vdots\\ \vdash R\end{array} \qquad \begin{array}{c}\vdots\\ R \rightarrow (S \rightarrow T), S \rightarrow T \vdash R \rightarrow T\end{array}}{R \rightarrow (S \rightarrow T), S \vdash R \rightarrow T}\rightarrow\mathrm{E}^T$$

It is impossible to derive the sequent $\vdash R$, since the proposition R is not a tautology in intuitionistic logic. In this way, NJ_T is forcing us to use another tactic here. We'll use $\to$I and $\to$I^T at this step instead.

$$\dfrac{\begin{array}{c}\dfrac{}{R \to (S \to T) \vdash R \to (S \to T)} \qquad \dfrac{}{S \vdash S} \\ \vdots \qquad\qquad\qquad\qquad \vdots \\ \vdots \\ R \to (S \to T), S, R \vdash T\end{array}}{R \to (S \to T), S \vdash R \to T} \to\mathrm{I} \qquad \Bigg| \qquad \dfrac{\begin{array}{c}\vdots \\ R \to (S \to T), S, R \vdash T\end{array}}{R \to (S \to T), S \vdash R \to T} \to\mathrm{I}^T$$

Applying implication introduction as our first step ensures that R is discharged before the end of the proof. I suspect the creators of MacLogic designed the rules in NJ_T so that students couldn't make assumptions before they knew where they might be discharged.

We now know that R needs to be assumed so that it can be discharged with an application to $\to$I, so we can safely apply $\to$E to our NJ derivation.

$$\dfrac{\begin{array}{c}\dfrac{\dfrac{}{R \to (S \to T) \vdash R \to (S \to T)} \qquad \dfrac{}{R \vdash R}}{R \to (S \to T), R \vdash S \to T} \to\mathrm{E} \qquad \dfrac{}{S \vdash S} \\ \vdots \qquad\qquad\qquad\qquad\qquad \vdots \\ \vdots \\ R \to (S \to T), S, R \vdash T\end{array}}{R \to (S \to T), S \vdash R \to T} \to\mathrm{I}$$

Applying $\to$E^T to our NJ_T derivation produces two new subproblems. R appears on both sides of the turnstile in the sequent $S, R \vdash R$, so we can apply Assume^T and be done with it.

$$\dfrac{\dfrac{\dfrac{}{S, R \vdash R} \qquad \begin{array}{c}\vdots \\ S, R, S \to T \vdash T\end{array}}{R \to (S \to T), S, R \vdash T} \to\mathrm{E}^T}{R \to (S \to T), S \vdash R \to T} \to\mathrm{I}^T$$

The last subproblem to derive in the NJ_T derivation is $S, R, S \to T \vdash T$, which is the sequent we'd get if we were to weaken $S, S \to T \vdash T$. The sequent $S, S \to T \vdash T$ says that T is provable from S and $S \to T$, which follows from $\to$E in NJ. Therefore, we can finish the NJ derivation by applying $\to$E.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{R \to (S \to T) \vdash R \to (S \to T)} \qquad \dfrac{}{R \vdash R}}{R \to (S \to T), R \vdash S \to T} \to\mathrm{E} \qquad \dfrac{}{S \vdash S}}{R \to (S \to T), R, S \vdash T} \to\mathrm{E}}{R \to (S \to T), S \vdash R \to T} \to\mathrm{I}$$

The NJ_T derivation is also finished with another application of $\to$E^T and two applications of Assume^T to close out the remaining subproblems.

$$\dfrac{\dfrac{\dfrac{}{S, R \vdash R} \qquad \dfrac{\dfrac{}{S, R \vdash S} \qquad \dfrac{}{S, R, T, \vdash T}}{S, R, S \to T \vdash T} \to\mathrm{E}^T}{R \to (S \to T), S, R \vdash T} \to\mathrm{E}^T}{R \to (S \to T), S \vdash R \to T} \to\mathrm{I}^T$$

4.1.4 Tactics for $\neg$

Like $\wedge$I and $\rightarrow$I, the tactic corresponding to negation introduction, presented as $\neg \mathrm{I}^T$, is equivalent to $\neg$I.

$$\frac{\Gamma, \alpha \vdash \bot}{\Gamma \vdash \neg\alpha}\neg\mathrm{I} \quad\Bigg|\quad \frac{\Gamma, \alpha \vdash \bot}{\Gamma \vdash \neg\alpha}\neg\mathrm{I}^T$$

The tactic corresponding to negation elimination is similar enough to the tactic corresponding to implication elimination that there isn't a need to illustrate it by comparing derivations in NJ and NJ_T.

$$\frac{\Gamma \vdash \neg\alpha \qquad \Gamma' \vdash \alpha}{\Gamma, \Gamma' \vdash \bot}\neg\mathrm{E} \quad\Bigg|\quad \frac{\Gamma \vdash \alpha \qquad \Gamma, \bot \vdash \phi}{\Gamma, \neg\alpha \vdash \phi}\neg\mathrm{E}^T$$

4.1.5 Tactics for $\vee$

The tactics corresponding to the rules for or introduction, presented as $\vee \mathrm{I_{left}}^T$ and $\vee \mathrm{I_{right}}^T$, are equivalent to $\vee \mathrm{I_{left}}$ and $\vee \mathrm{I_{right}}$.

$$\frac{\Gamma \vdash \alpha}{\Gamma \vdash \alpha \vee \beta}\vee\mathrm{I_{left}} \qquad \frac{\Gamma \vdash \beta}{\Gamma \vdash \alpha \vee \beta}\vee\mathrm{I_{right}} \quad\Bigg|\quad \frac{\Gamma \vdash \alpha}{\Gamma \vdash \alpha \vee \beta}\vee\mathrm{I_{left}}^T \qquad \frac{\Gamma \vdash \beta}{\Gamma \vdash \alpha \vee \beta}\vee\mathrm{I_{right}}^T$$

The final tactic presented here correspond to $\vee$E, and is presented as $\vee \mathrm{E}^T$. It says that if ϕ is provable from a set of assumptions Γ along with α, and ϕ is provable from Γ and β, then ϕ is provable from Γ along with $\alpha \vee \beta$.

$$\frac{\Gamma \vdash \alpha \vee \beta \qquad \Gamma', \alpha \vdash \phi \qquad \Gamma'', \beta, \vdash \phi}{\Gamma, \Gamma', \Gamma'' \vdash \phi}\vee\mathrm{E} \quad\Bigg|\quad \frac{\Gamma, \alpha \vdash \phi \qquad \Gamma, \beta \vdash \phi}{\Gamma, \alpha \vee \beta \vdash \phi}\vee\mathrm{E}^T$$

We will be comparing derivations of the sequent $A \vee B \vdash B \vee A$ in NJ and NJ_T. The derivation in NJ will be constructed both from the top-down and from the bottom-up, while the derivation in NJ_T will be built from the bottom-up. The derivation in NJ is on the left, while the derivation in NJ_T is on the right.

$$\begin{array}{c} \overline{A \vee B \vdash A \vee B} \\ \vdots \\ A \vee B \vdash B \vee A \end{array} \quad\Bigg|\quad \begin{array}{c} \vdots \\ A \vee B \vdash B \vee A \end{array}$$

Applying $\vee$E to the NJ derivation gives us the following.

$$\frac{\overline{A \vee B \vdash A \vee B} \qquad \begin{array}{c} \overline{A \vdash A} \\ \vdots \\ A \vdash B \vee A \end{array} \qquad \begin{array}{c} \overline{B \vdash B} \\ \vdots \\ B \vdash B \vee A \end{array}}{A \vee B \vdash B \vee A}\vee\mathrm{E}$$

Applying $\vee \mathrm{E}^T$ to the NJ_T derivation gives us the following.

$$\frac{\begin{array}{c} \vdots \\ A \vdash B \vee A \end{array} \qquad \begin{array}{c} \vdots \\ B \vdash B \vee A \end{array}}{A \vee B \vdash B \vee A}\vee\mathrm{E}^T$$

We'll finish the NJ derivation by applying $\vee \mathrm{I_{right}}$ and $\vee \mathrm{I_{left}}$ to our remaining subgoals.

$$\dfrac{\dfrac{}{A \vee B \vdash A \vee B} \qquad \dfrac{\dfrac{}{A \vdash A}}{A \vdash B \vee A}\,\vee\mathrm{I}_{\mathrm{right}} \qquad \dfrac{\dfrac{}{B \vdash B}}{B \vdash B \vee A}\,\vee\mathrm{I}_{\mathrm{right}}}{A \vee B \vdash B \vee A}\,\vee\mathrm{E}$$

The NJ_T derivation will be finished in the same way.

$$\dfrac{\dfrac{\dfrac{}{A \vdash A}}{A \vdash B \vee A}\,\vee\mathrm{I}_{\mathrm{right}} \qquad \dfrac{\dfrac{}{B \vdash B}}{B \vdash B \vee A}\,\vee\mathrm{I}_{\mathrm{left}}}{A \vee B \vdash B \vee A}\,\vee\mathrm{E}^{T}$$

5 Implementation and Next Steps

I spent most of this semester implementing various logical systems using the Scala programming language. The most substantial of these systems comes in the form of a clone of MacLogic's construction mode. Its interface is much simpler than MacLogic's, and it currently doesn't allow users to write proofs relying on predicates or quantifiers, but I plan to write extensions enabling this functionality soon.

The software allows users to construct proofs using rules from NJ_T as described in section 4.1. However, the tactics are not specified as elegantly in code as they are here. This paper's presentation of NJ_T came out of efforts I made to generalize my implementation so that rules could be swapped out without much effort. This generalization process is still ongoing.

I intend to use the data representations I developed for expressing derivations, expressions, and rules in future iterations of the Coastline project. My project's source code can be found here: `https://github.com/imapersonman/MacLogic2`.

The Coastline project is still focused on mathematics education, despite this paper's focus on logic. I've always imagined Coastline as allowing students the freedom to split a problem into multiple subproblems and combine their solutions to solve more complex problems. This kind of reasoning is integral to mathematics education. I wanted to include this in my original implementation of Shoreline but felt I didn't understand enough about abstract reasoning to properly implement this kind of interaction. Natural deduction formally describes this kind of reasoning. Therefore, I should be able to use the software I implemented for my thesis as a foundation for future educational interfaces that I intend to develop.

6 Acknowledgements

This project would not have been possible without the support of my thesis committee. Professor Branden Fitelson has been an invaluable source of resources, knowledge, and good conversation over the past two semesters.

Professor Anthony Iarrobino first suggested that I enroll in Math 4020 to work on my project, and helped me secure funding for a trip to Montréal. He and the undergraduate research fund made that trip possible, and encouraged me to continue working on my project. I am grateful for their continued support.

At the beginning of my time in Math 4020, Professor Stanley Eigen, along with Professors Fitelson, Iarrobino, and Lee Peng Lee, met with me to discuss my goals for the semester. Since then, Professor Eigen has shown an interest in my project that I appreciated.

References

[1] Adjorlolo, Koissi. “Spring 2019 write-up on Shoreline” https://koissiadjorlolodotnet.wordpress.com/2019/08/31/spring-2019-write-up-on-shoreline/

[2] Adjorlolo, Koissi. “Coastline Final Report” https://koissiadjorlolodotnet.files.wordpress.com/2019/12/final.pdf

[3] Fitelson, B. “Running MacLogic Under Emulation” http://fitelson.org/maclogic.htm

[4] Forbes, Graeme. “Modern logic: A text in elementary symbolic logic.” (1994).

[5] Martin-Löf, P. (1996). On the meanings of the logical constants and the justifications of the logical laws. Nordic journal of philosophical logic, 1(1), 11-60.

[6] Pfenning, F. (2009). Lecture notes on natural deduction. Course CMU, 15317.

[7] https://builds.openlogicproject.org/open-logic-complete.pdf