Propositions and Types, Proofs and Programs

Part II: Natural Deduction

Ranald Clouston

School of Computing

Australian National University

ANU Logic Summer School 2023

Sequent Calculus vs Natural Deduction

Many similarities between the two systems, including translations back and forth.

Sequent calculus is often the more convenient setting for automated **proof search**, and for proving results about a logic.

Natural deduction more closely resembles ‘natural’ human reasoning

- And, via the Curry-Howard isomorphism, programming!

The Structure of Natural Deduction

Sequent calculus derivations form a **tree** of sequents with the conclusion at the root, axioms at the leaves, and each connection inside the tree a (substitution instance of a) proof rule.

Natural deduction derivations form a **tree of formulae**

- We will see that they can also be presented ‘sequent-style’

The formulae on the leaves are ‘alive’ or ‘dead’

- If alive, they are **assumptions**
- If dead we write them inside square brackets. They are **discharged** assumptions from a subproof, ‘killed’ by one of our proof rules.

The Structure of Natural Deduction

The simplest natural deduction proof is

$$A$$

This means `from the assumption A, conclude A'

- No proof rules were involved

A larger derivation might look like something like:

$$\dfrac{\dfrac{C \qquad \dfrac{[A] \qquad B}{D}}{E}}{F}$$

"From assumptions B and C (not A!), conclude F"

The Rules of Natural Deduction

Rules may have any number of premises, but only 1 conclusion

Rules **introduce** or **eliminate** a particular connective

- The introduction rules should follow the BHK interpretation, and resemble sequent calculus **right** rules
- The elimination rules should be the introduction rules, ‘upside down’ (we will not make this condition formal)

Some rules can ‘kill’ formulae in the leaves

- If a formula A is written in square brackets above a premise in a rule, it means ‘you may kill any number (0, 1, or more) of occurrences of A in the leaves’.
- You do not have to kill all occurrences of such an A

Conjunction

"A proof of $A \wedge B$ *is given by presenting a proof of* A *and a proof of* B*"*

$$\frac{A \qquad B}{A \wedge B}\ \wedge I$$

If we have a proof of $A \wedge B$ then we can eliminate it to get a proof of either A, or if we prefer, B :

$$\frac{A \wedge B}{A}\ \wedge E_1 \qquad \frac{A \wedge B}{B}\ \wedge E_2$$

Conjunction Example

From $A \land B$, conclude $B \land A$:

$$\dfrac{\land E_2\ \dfrac{A \land B}{B} \qquad \dfrac{A \land B}{A}\ \land E_1}{B \land A}\ \land I$$

False

"⊥ has no proof"

- So ⊥ has no introduction rule

But it *does* have an elimination rule: if we have somehow concluded ⊥, then anything goes!

$$\frac{\perp}{A}\ \perp E$$

Simple exercise: from $A \wedge \perp$, prove B

Implication

"A proof of A→B *is a construction which permits us to transform any proof of* A *into a proof of* B*"*

$$\frac{\begin{array}{c}[A]\\ B\end{array}}{A \to B}\ \to\mathrm{I}$$

If we have a proof of A→B and a proof of A, then we combine them:

$$\frac{A \to B \qquad A}{B}\ \to\mathrm{E}$$

Implication Example

$A \rightarrow (B \rightarrow A)$ without assumptions:

$$\dfrac{\begin{array}{c}[A] \\ \vdots \\ B \rightarrow A\end{array}}{A \rightarrow (B \rightarrow A)} \; \rightarrow I$$

Implication Example

$A \to (B \to A)$ without assumptions:

$$
\dfrac{\dfrac{\begin{matrix}[A, B] \\ \vdots \\ A\end{matrix}}{B \to A}\;{\to}\mathrm{I}}{A \to (B \to A)}\;{\to}\mathrm{I}
$$

Implication Example

$A \rightarrow (B \rightarrow A)$ without assumptions:

$$\dfrac{\dfrac{[A]}{B \rightarrow A}\ {\rightarrow}\mathrm{I}}{A \rightarrow (B \rightarrow A)}\ {\rightarrow}\mathrm{I}$$

B is never used, so does not need to be killed.

Implication Example II

$(A\to B)\to((A\to(B\to C))\to(A\to C))$ without assumptions:

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[A\to(B\to C)] \qquad [A]}{B\to C}\ {\to}E \qquad \dfrac{[A\to B] \qquad [A]}{B}\ {\to}E}{C}\ {\to}E}{A\to C}\ {\to}I}{(A\to(B\to C))\to(A\to C)}\ {\to}I}{(A\to B)\to((A\to(B\to C))\to(A\to C))}\ {\to}I
$$

Disjunction

"*A proof of* $A \vee B$ *is given by presenting either a proof of* A *or a proof of* B"

$$\frac{A}{A \vee B}\ \vee I_1 \qquad\qquad \frac{B}{A \vee B}\ \vee I_2$$

For elimination, we cannot always know if $A \vee B$ came from A or B, so we have to argue by cases:

$$\frac{A \vee B \qquad \begin{matrix}[A] \\ C\end{matrix} \qquad \begin{matrix}[B] \\ C\end{matrix}}{C}\ \vee E$$

Disjunction Example

From $A \vee B$, conclude $B \vee A$:

$$\dfrac{A \vee B \qquad \dfrac{[A]}{B \vee A}\,{\scriptstyle \vee I_2} \qquad \dfrac{[B]}{B \vee A}\,{\scriptstyle \vee I_1}}{B \vee A}\,{\scriptstyle \vee E}$$

All Natural Deduction Rules

$$\frac{A \qquad B}{A \land B} \qquad \frac{A \land B}{A} \qquad \frac{A \land B}{B}$$

$$\frac{A}{A \lor B} \qquad \frac{B}{A \lor B} \qquad \frac{A \lor B \quad \begin{matrix}[A] \\ C\end{matrix} \quad \begin{matrix}[B] \\ C\end{matrix}}{C}$$

$$\frac{\begin{matrix}[A] \\ B\end{matrix}}{A \to B} \qquad \frac{A \to B \qquad A}{B} \qquad \frac{\bot}{A}$$

Natural Deduction, Sequent Style

It is sometimes to convenient to record all our current assumptions in every node of the tree

- Does clutter our proofs, but can help us keep track

We do this via a (single conclusion) sequent $\Gamma \vdash A$

- This is *not* sequent calculus; just a different notation for natural deduction

Our assumptions can then be read off the left hand side of the root, rather than from the leaves.

Killing an assumption means removing it from our left hand side.

Sequent Style: Assumption Rule

Because we are building a tree of sequents, not formulae, we need to change our leaves from lone formulae to explicit uses of assumptions:

$$\dfrac{}{\Gamma,\ A \vdash A}\ \text{AX}$$

This is a 0-premise rule, i.e. an axiom.

As usual we do not care about the ordering of the left hand side: A could appear anywhere in Γ.

Sequent Style: Killing Assumptions

Our two rules that kill assumptions:

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

As usual Γ is a multiset (can contain more than one occurrence of a formula) so killing A does not mean A cannot still be in Γ.

- This is necessary to prove e.g. $A \to (A \to A)$

All Sequent-Style Natural Deduction Rules

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

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

$$\frac{\Gamma,\ A \vdash B}{\Gamma \vdash A \rightarrow B} \qquad \frac{\Gamma \vdash A \rightarrow B \qquad \Gamma \vdash A}{\Gamma \vdash B} \qquad \frac{\Gamma \vdash \bot}{\Gamma \vdash A}$$

Sequent-Style Natural Deduction Example

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{A \to B,\ A \to (B \to C),\ A \vdash A \to (B \to C) \quad A \to B,\ A \to (B \to C),\ A \vdash A}{A\to B,\ A\to(B\to C),\ A \vdash B\to C} \quad \dfrac{A \to B,\ A \to (B \to C),\ A \vdash A \to B \quad A \to B,\ A \to (B \to C),\ A \vdash A}{A\to B,\ A\to(B\to C),\ A \vdash B}}{A\to B,\ A\to(B\to C),\ A \vdash C}}{A\to B,\ A\to(B\to C) \vdash A\to C}}{A\to B \vdash (A\to(B\to C))\to(A\to C)}}{\vdash (A\to B)\to((A\to(B\to C))\to(A\to C))}$$

The extra book-keeping gets a bit heavy at the top!

- But it does help us to understand the ‘state of play’ at any point of the tree

Natural Deduction for Classical Logic?

Sequent calculus seems equally good for classical and intuitionistic logic

- Multiple conclusion vs single conclusion

But, with natural deduction, classical logic does not feel so good.

We can chuck in LEM or double negation elimination

- But that ‘breaks’ the principle of using only introduction and elimination rules

We can also work with multiple conclusions

- Not obvious how to do this correctly! Took until the 1990s to work out
- And what might this mean for programming purposes?

Organising our Proofs

One theorem might have many proofs.

- e.g. to prove $(A \land A) \to A$, do we choose to go via $\land E1$ or $\land E2$?

But that does not mean every proof is equally good!

What if our proof contained this 'roundabout' argument?

$$\dfrac{\dfrac{A \qquad\qquad B}{A \land B}\,\land I}{A}\,\land E_1$$

We've done work to conclude A from A, and should cut this part out.

- This process can be done algorithmically, and is called **normalisation**

Towards Normalisation

The concept of normalisation was invented by Gentzen and perfected by **Dag Prawitz** (1936-) without concern for computer science.

But it is exactly this concept that animates the Curry-Howard isomorphism that our lectures are driving towards.

We will therefore briefly put logic to one side to look at the programming side of our picture.

Our topic will be the **lambda-calculus**.

Photo c/o Stockholm University