lec9: Intuitionism, negation, sequent calculus

Joshua Dunfield

February 7, 2018

See lec6–8 for our development of natural deduction.

1 Negation ($\neg$)

In Boolean logic, we may treat propositions as variables that are either true or false. Up to about 1900, this conception was predominant in mathematical logic; its first main challenge came from Brouwer (who, like Gentzen, wasn't a great human being) in 1907. By the 1930s, Brouwer's *intuitionism* had gained some popularity; two of the four calculi of Gentzen's thesis are intuitionistic.

I won't cover the philosophical side of intuitionism; for that, look at the Stanford Encyclopedia's "Intuitionistic Logic" and follow the links in its second paragraph. I will discuss a narrow technical side of intuitionism—how it affects the specific rules of natural deduction—and the broader relationship of intuitionism and computing.

A bit is a binary digit, either 0 or 1. Thinking of the truth of a logical proposition as a bit-value is well suited to digital circuits, and to applications such as SAT solving. Being clearly either 0 or 1 is the essential feature of a bit, which may suggest that the truth of a logical proposition should be similarly clear:

- Consider an integer m. If m is odd, m is not even; if m is even, m is not odd. Since every integer is either odd or even, every integer is either odd or not-odd. So it seems reasonable for "oddness" to be a Boolean property, either true (odd) or false (not-odd = even).
- Consider a Turing machine, whose configuration is given by a tuple including the number of its current state. Some states are designated as *final states* (meaning that the machine has halted). At any step of time, the machine is either *in* a final state (its current state is in the set of final states) or *not* in a final state (its current state is not in the set of final states), so the truth value of the proposition "the current state" is either true or false.

But undecidability spoils this clarity. We can ask whether a known integer (the current state) is part of a small finite set of integers (the set of final states), and the answer will be immediate. However, not all questions have answers.

- Consider a Turing machine H, whose configuration is given by a tuple including the number of its current state and T, its input tape. If the machine is allowed to run indefinitely, will it reach a halting state?

While the Boolean formula *halts*(H) *looks* like the Boolean formula *in-a-final-state*(H), the truth value of *halts*(H) is much less clear, because this is an undecidable problem. For *specific* Turing machines we can give an answer. For example:

- If the initial state of H is a final state, then *halts*(H) is true.

- If the Turing machine has only one state, which is not a final state, then *halts*(H) is false.

Being able to answer specific instances of the general question does not mean the problem is decidable. In fact, we can answer the question for a large number of Turing machines, by simulating the machine for a few million steps and checking if it halts. If it halts, then *halts*(H) is true. If it doesn't halt, we haven't answered the question. For some machines that never halt, we can say definitively that they don't halt by providing a proof: "The machine has only one state; therefore, it will always be in that state; since that state is not a final state, the machine never halts."

In intuitionistic logics, statements of existence must be backed with a *witness* to the existence. If I claim that the machine halts, I must show you when it halts. If I claim that the machine doesn't halt, I must prove that to you. A proposition is *true* if *and only if* you have a proof.

Thus, in intuitionistic logics, a statement like

$$\text{either } \mathit{halts}(\mathsf{H}) \text{ or } \neg\mathit{halts}(\mathsf{H})$$

is interpreted as

$$\text{either there exists a proof of } \mathit{halts}(\mathsf{H}) \text{ or there exists a proof of } \neg\mathit{halts}(\mathsf{H})$$

(I am retroactively declaring that H includes both a description of the machine (states and transitions) and the input tape T. Whether T is included does not affect decidability.)

Since the halting problem is undecidable, the statement

$$\text{for all Turing machines H, either there exists a proof of } \mathit{halts}(\mathsf{H}) \text{ or there exists a proof of } \neg\mathit{halts}(\mathsf{H})$$

is not true. (Whether we also say the statement is false is, I think, a matter of taste.) So the statement

for all propositions P,
    either there exists a proof of P or there exists a proof of ¬P

is false.

In natural deduction, the truth of a formula A is determined by whether there exists a derivation of A true. Since natural deduction (through the atomic formulas that I deliberately haven't specified) can talk about pretty much anything, we should not expect (after we add negation)

for all formulas A,
    either there exists a natural-deduction derivation of A true
    or there exists a natural-deduction derivation of $(\neg A)$ true

This means we should not want a rule "LEM" (Law of the Excluded Middle):

$$\frac{}{(A \vee \neg A)\ \mathsf{true}}\ \text{LEM?}$$

If our mental model includes undecidable questions (and it should), including this rule would make natural deduction unsound with respect to that mental model.

Rejecting LEM does *not* reject negation entirely! We can still have the Law of Contradiction:

$$\frac{A\ \mathsf{true} \qquad (\neg A)\ \mathsf{true}}{\mathsf{False}\ \mathsf{true}}\ \text{LoC}$$

In fact, LoC was Gentzen's elimination rule for ¬. What was Gentzen's *introduction* rule for ¬?

$$\dfrac{\begin{array}{c} x[A\ \mathsf{true}] \\ \vdots \\ \mathsf{False\ true} \end{array}}{(\neg A)\ \mathsf{true}}\ \neg\text{Intro}^x$$

Both LoC and ¬Intro are laws permitted by Brouwer's intuitionism.

During lecture, based on a student suggestion, we developed an alternate ¬Intro:

$$\dfrac{\begin{array}{c} x[A\ \mathsf{true}] \\ \vdots \\ B\ \mathsf{true} \end{array} \qquad (\neg B)\ \mathsf{true}}{(\neg A)\ \mathsf{true}}\ \neg\text{Intro-alternate}^x$$

I believe this rule is equivalent to Gentzen's ¬Intro. That is, if our system has ¬Intro, then ¬Intro-alternate is *admissible* (redundant). On the other hand, if we have ¬Intro-alternate *instead of* ¬Intro, then ¬Intro is admissible in *that* system.

- Part 1: ¬Intro-alternate is admissible.

  To show admissibility of a rule R, we assume that we have derivations of all premises of R, and construct a derivation of the conclusion of R without using R. (If we were allowed to use R to derive the conclusion, any imaginable rule would be "admissible" and the idea of admissibility would be useless.)

  1. *Assume the first premise of ¬Intro-alternate:* Assume a derivation of $B\ \mathsf{true}$ with the floating assumption $x[A\ \mathsf{true}]$.
  2. *Assume the second premise of ¬Intro-alternate:* Assume a derivation of $(\neg B)\ \mathsf{true}$.
  3. Our goal is to derive $(\neg A)\ \mathsf{true}$ using our existing rules, including ¬Intro.
  4. Since we are trying to derive $(\neg \ldots)\ \mathsf{true}$, using the existing ¬-introduction rule, ¬Intro, seems likely to work. That rule also has one premise under the floating assumption $x[A\ \mathsf{true}]$. So, assume $x[A\ \mathsf{true}]$.
  5. By our first assumption, if we assume $A\ \mathsf{true}$ we can get $B\ \mathsf{true}$:

  $$\begin{array}{c} x[A\ \mathsf{true}] \\ \vdots \\ B\ \mathsf{true} \end{array}$$

  6. By our second assumption, we have a derivation of $(\neg B)\ \mathsf{true}$.
  7. If we can derive $(\neg B)\ \mathsf{true}$ under no assumptions, which is what the second premise of ¬Intro-alternate says, then we can derive it under additional assumptions: we can simply ignore the assumption. (This is called a *weakening* property, which we should prove but won't. I think it's easier to state and prove in sequent calculus.) Therefore, we also have a derivation

  $$\begin{array}{c} x[A\ \mathsf{true}] \\ \vdots \\ (\neg B)\ \mathsf{true} \end{array}$$

8. Since we have B true and $(\neg \mathsf{B})$ true, we can apply rule LoC (Law of Contradiction):

$$\cfrac{\begin{matrix} x[\mathsf{A}\ \mathsf{true}] \\ \vdots \\ \mathsf{B}\ \mathsf{true} \qquad (\neg \mathsf{B})\ \mathsf{true} \end{matrix}}{\mathsf{False}\ \mathsf{true}}\ \text{LoC}$$

9. The above derivation matches the premise of Gentzen's ¬Intro, so:

$$\cfrac{\cfrac{\begin{matrix} x[\mathsf{A}\ \mathsf{true}] \\ \vdots \\ \mathsf{B}\ \mathsf{true} \qquad (\neg \mathsf{B})\ \mathsf{true} \end{matrix}}{\mathsf{False}\ \mathsf{true}}\ \text{LoC}}{(\neg \mathsf{A})\ \mathsf{true}}\ \neg\text{Intro}^x$$

10. Observe that $(\neg \mathsf{A})$ true is the conclusion of ¬Intro-alternate.

Therefore, ¬Intro-alternate is admissible.

- Part 2: On the other hand, suppose we have all our rules *except* ¬Intro, *and* add ¬Intro-alternate. Can we show that Gentzen's ¬Intro is admissible?

1. *Assume the first (and only) premise of ¬Intro:* Assume a derivation of False true with the floating assumption $x[\mathsf{A}\ \mathsf{true}]$.
2. Our goal is to derive $(\neg \mathsf{A})$ true using ¬Intro-alternate (and possibly LoC and any other rule, *except* ¬Intro).
3. Since we are trying to derive $(\neg \ldots)$ true, using ¬Intro-alternate seems likely to work. That rule has two premises. One is under the floating assumption $x[\mathsf{A}\ \mathsf{true}]$. So, assume $x[\mathsf{A}\ \mathsf{true}]$.
4. By our first assumption, if we assume A true we can get B true:

$$\begin{matrix} x[\mathsf{A}\ \mathsf{true}] \\ \vdots \\ \mathsf{False}\ \mathsf{true} \end{matrix}$$

5. At this point, I got stuck. Is there a solution?

1.1 Classical natural deduction

Classical (not intuitionistic) logics do include LEM. In addition to NJ (which is essentially the same as our rules), Gentzen presented NK (for *klassische*), which is identical to NJ but adds the rule LEM.

2 Natural deduction with negation

$$
\begin{array}{llcll}
\text{atomic formulas} & P, Q & & & \\
\text{formulas} & A, B, C & ::= & P & \text{atomic formula} \\
 & & & \mid A \supset B & \text{implication} \\
 & & & \mid A \,\&\, B & \text{conjunction (and)} \\
 & & & \mid A \vee B & \text{disjunction (or)} \\
 & & & \mid \forall a : \mathsf{Nat}.\, A & \text{universal quantification} \\
 & & & \mid \exists a : \mathsf{Nat}.\, A & \text{existential quantification} \\
 & & & \mid \mathsf{True} & \text{truth} \\
 & & & \mid \mathsf{False} & \text{falsehood} \\
 & & & \mid \neg A & \text{negation}
\end{array}
$$

$\boxed{A\ \mathsf{true}}$ A is true

$$
\dfrac{\begin{matrix} x\,[A\ \mathsf{true}] \\ \vdots \\ B\ \mathsf{true} \end{matrix}}{(A \supset B)\ \mathsf{true}}\ \supset\!\mathrm{Intro}^{x}
\qquad
\dfrac{A \supset B\ \mathsf{true} \qquad A\ \mathsf{true}}{B\ \mathsf{true}}\ \supset\!\mathrm{Elim}
$$

$$
\dfrac{}{\mathsf{True}\ \mathsf{true}}\ \mathrm{TrueIntro}
\qquad \text{no elim. for True} \qquad \text{intro. for False: see } \neg\text{Elim below} \qquad
\dfrac{\mathsf{False}\ \mathsf{true}}{C\ \mathsf{true}}\ \mathrm{FalseElim}
$$

$$
\dfrac{A\ \mathsf{true} \qquad B\ \mathsf{true}}{A \,\&\, B\ \mathsf{true}}\ \&\mathrm{Intro}
\qquad
\dfrac{A \,\&\, B\ \mathsf{true}}{A\ \mathsf{true}}\ \&\mathrm{Elim1}
\qquad
\dfrac{A \,\&\, B\ \mathsf{true}}{B\ \mathsf{true}}\ \&\mathrm{Elim2}
$$

$$
\dfrac{\begin{matrix} x\,[a : \mathsf{Nat}] \\ \vdots \\ B\ \mathsf{true} \end{matrix}}{(\forall a : \mathsf{Nat}.\, B)\ \mathsf{true}}\ \forall\mathrm{Intro}^{x}
\qquad
\dfrac{(\forall a : \mathsf{Nat}.\, B)\ \mathsf{true} \qquad n : \mathsf{Nat}}{[n/a]B\ \mathsf{true}}\ \forall\mathrm{Elim}
$$

$$
\dfrac{A\ \mathsf{true}}{A \vee B\ \mathsf{true}}\ \vee\mathrm{Intro1}
\qquad
\dfrac{B\ \mathsf{true}}{A \vee B\ \mathsf{true}}\ \vee\mathrm{Intro2}
\qquad
\dfrac{A \vee B\ \mathsf{true} \qquad \begin{matrix} x\,[A\ \mathsf{true}] \\ \vdots \\ C\ \mathsf{true} \end{matrix} \qquad \begin{matrix} y\,[B\ \mathsf{true}] \\ \vdots \\ C\ \mathsf{true} \end{matrix}}{C\ \mathsf{true}}\ \vee\mathrm{Elim}^{x,y}
$$

$$
\dfrac{n : \mathsf{Nat} \qquad ([n/a]B)\ \mathsf{true}}{(\exists a : \mathsf{Nat}.\, B)\ \mathsf{true}}\ \exists\mathrm{Intro}
\qquad
\dfrac{(\exists a : \mathsf{Nat}.\, B)\ \mathsf{true} \qquad \begin{matrix} x\,[a : \mathsf{Nat}] \\ y\,[B\ \mathsf{true}] \\ \vdots \\ C\ \mathsf{true} \end{matrix}}{C\ \mathsf{true}}\ \exists\mathrm{Elim}^{x,y}
$$

$$
\dfrac{\begin{matrix} x\,[A\ \mathsf{true}] \\ \vdots \\ \mathsf{False}\ \mathsf{true} \end{matrix}}{(\neg A)\ \mathsf{true}}\ \neg\mathrm{Intro}^{x}
\qquad
\dfrac{A\ \mathsf{true} \qquad (\neg A)\ \mathsf{true}}{\mathsf{False}\ \mathsf{true}}\ \neg\text{Elim—or should it be FalseIntro? (LoC)}
$$