# Proof Systems

Lecture 3

Logic and Programming Languages

# Proof system

- A proof system is a formalized system for proving things.
- Most systems have several components
  1. A set of Axioms. Things that are known to be true without any work
  2. A set if inference rules for deriving larger true statements from smaller true statements
  3. A set of assumptions from which to work

1. In a mechanized logic, a proof is a data structure that can be checked by a machine

# Consistency, completeness, normal forms

- Consistency
  - A system is consistent if falsehood is not provable (from the empty set of assumptions)
  - A system is complete if every theorem is provable from the inference rules of the logic
  - A Normal  Form exists of there exists a unique smallest proof  for every theorem, and other proofs of the same theorem “reduce” to this proof.

# Natural Deduction

- A style of proof with several elements that have become widely used
    1. Introduction rules
    2. Elimination rules
    3. Hypothetical judgements
        1. Reasoning from assumptions

1. Proofs are represented by a tree of “true statements” rooted at the bottom.

# Proof trees

- A proof tree has several parts
  1. A statement of what is proven (the root). Drawn below the line
  2. A set of sub trees that represent proofs of the required components. Drawn above the line
  3. A name for the inference rule used. Draw to the left of the line.
  4. A set of premises. Drawn in brackets

```
[p0 /\ p1]
----------- /\e2
    p1               [p2]
------------------------ /\i
        p1 /\ p2
---------------------------- ~~i
        ~~(p1 /\ p2)
```

# Introduction rules

- For each connective of the logic, there is an introduction rule, where the root (below the line) has that connective has its outermost form.

$$\frac{A \quad B}{A \wedge B}(\wedge I) \qquad \frac{A}{\neg\neg A}(\neg\neg I) \qquad \frac{A}{A \vee B}(\vee I_1)$$

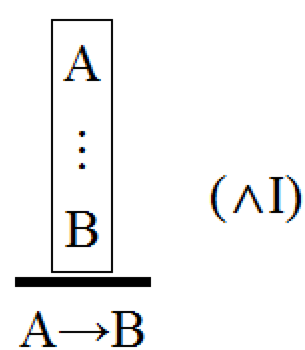
# Elimination rules

- For each connective there is a rule that tells how to "consume" a formula with that connective to prove something else. Here the formula with that connective is above the line.

$$\frac{A \wedge B}{A}\,(\wedge E_1) \qquad \frac{\neg\neg A}{A}\,(\neg\neg E) \qquad \frac{A \quad A \rightarrow B}{B}\,(\rightarrow E)$$

# Hypothetical Judgements

- Somethings can be proven using a sort of conditional reasoning.
- We need a way to “temporarily” assume a new condition, and then cut of this assumption when we are done.
  - Assume some formula are true
  - Infer other things follow from these assumptions
    - These are consequences of the assumptions



# Natural deduction by the rules

- We will look at each connective, and then study both the introduction and elimination rules for it.

# And

$$\frac{A \quad B}{A \wedge B}\,(\wedge I) \qquad \frac{A \wedge B}{A}\,(\wedge E_1) \qquad \frac{A \wedge B}{B}\,(\wedge E_2)$$

# Or

$$\frac{A}{A \vee B}(\vee I_1) \qquad \frac{B}{A \vee B}(\vee I_2) \qquad \frac{A \vee B \quad A \rightarrow C \quad B \rightarrow C}{C}(\vee E)$$

# Not

$$\frac{A \rightarrow \bot}{\neg A}\ (\neg I) \qquad \frac{A \quad \neg A}{\bot}\ (\neg E) \text{ or } (\bot I) \qquad \frac{\bot}{A}\ (\bot E)$$

$$\frac{A}{\neg\neg A}\ (\neg\neg I) \qquad \frac{\neg\neg A}{A}\ (\neg\neg E)$$

# Implies

$$\frac{A \quad A \rightarrow B}{B}\ (\rightarrow E)$$

$$\frac{\boxed{\begin{array}{c} A \\ \vdots \\ B \end{array}}}{A \rightarrow B}\ (\wedge I)$$

# Semantics

- The statement below the line is a consequence of the premises, and if it is in a box, the assumption of the box.
- Natural deduction works by maintaining this invariant
- Every step keeps the invariant true

# Natural Deduction as a mechanized proof system.

```
data NatDed n
  = Premise (Prop n)
  | AndI (NatDed n) (NatDed n)
  | AndE1 (NatDed n)
  | AndE2 (NatDed n)
  | Neg2I (NatDed n)
  | Neg2E (NatDed n)
  | ImplyI (Prop n) (NatDed n)
  | ImplyE (NatDed n) (NatDed n)
  | OrI1 (NatDed n) (Prop n)
  | OrI2 (Prop n) (NatDed n)
  | OrE (NatDed n) (NatDed n) (NatDed n)
  | AbsurdE (NatDed n) (Prop n)
  | AbsurdI (NatDed n) (NatDed n)
  | NegI (Prop n) (NatDed n)
```

# Using NatDed

- Building a term of type NatDed is a tree-like structure
- This tree might be a proof tree. If it maintains the invariant.
- A computer program can “check” if that is the case.

# Constructing proof trees

- Constructing proof trees is a lot like programming.
- You are given some premises. These are input to the checker.
- You must build a NatDed data structure that relies only on the given premises.
- Building this tree is a lot like programming. You must build it out if the constructors of NatDed in such a way that the checker will succeed.

# Representing the Premises as Data

- data Sequent n = Seq [Prop n] (NatDed n)

# Difficulties

- One must think to build a proof tree that will check.
- What pieces do you have?
  - What do they prove?
- What other pieces can you make?
- How can you put them together.
- Sometimes working bottom up helps.
- Mechanized help is useful.

# Strategy

- Construct a term.
- Name it.
- Let the system check and print it.
- Does it prove what you expect?
- Did  the check complain?
- Make some more terms
- Put them together.

# Gentzen style Proofs

- In a Gentzen style proof, we build a tree of hypothetical judgments, instead of a tree of true statements.
- Here the set of assumptions (hypotheses, premises) is an explicit part of the proof.

- $a \mid\!- a \wedge T$

# Gentzen approach

- Here we manipulate both the term to the right of the turnstile ( |- ) and the premises to the left of the turnstile.
- This approach is called the sequent calculus

# The sequent calculus

- The rules are broken into 4 cases.
- Some of the cases (the last 2) are broken into left and right variants
- The cases
  - Axiom
  - Cut
  - Logical rules
  - Structural rules

# Axiom and Cut

| Axiom: | Cut: |
| --- | --- |
| $$\dfrac{}{A \vdash A} \quad (I)$$ | $$\dfrac{\Gamma \vdash \Delta, A \qquad A, \Sigma \vdash \Pi}{\Gamma, \Sigma \vdash \Delta, \Pi} \quad (Cut)$$ |

# Logical Rules

| Left logical rules: | Right logical rules: |
| --- | --- |
| $\dfrac{\Gamma, A \vdash \Delta}{\Gamma, A \wedge B \vdash \Delta} \quad (\wedge L_1)$ | $\dfrac{\Gamma \vdash A, \Delta}{\Gamma \vdash A \vee B, \Delta} \quad (\vee R_1)$ |
| $\dfrac{\Gamma, B \vdash \Delta}{\Gamma, A \wedge B \vdash \Delta} \quad (\wedge L_2)$ | $\dfrac{\Gamma \vdash B, \Delta}{\Gamma \vdash A \vee B, \Delta} \quad (\vee R_2)$ |
| $\dfrac{\Gamma, A \vdash \Delta \qquad \Sigma, B \vdash \Pi}{\Gamma, \Sigma, A \vee B \vdash \Delta, \Pi} \quad (\vee L)$ | $\dfrac{\Gamma \vdash A, \Delta \qquad \Sigma \vdash B, \Pi}{\Gamma, \Sigma \vdash A \wedge B, \Delta, \Pi} \quad (\wedge R)$ |
| $\dfrac{\Gamma \vdash A, \Delta \qquad \Sigma, B \vdash \Pi}{\Gamma, \Sigma, A \to B \vdash \Delta, \Pi} \quad (\to L)$ | $\dfrac{\Gamma, A \vdash B, \Delta}{\Gamma \vdash A \to B, \Delta} \quad (\to R)$ |
| $\dfrac{\Gamma \vdash A, \Delta}{\Gamma, \neg A \vdash \Delta} \quad (\neg L)$ | $\dfrac{\Gamma, A \vdash \Delta}{\Gamma \vdash \neg A, \Delta} \quad (\neg R)$ |

# Structural Rules

| Left structural rules: | Right structural rules: |
|---|---|
| $\dfrac{\Gamma \vdash \Delta}{\Gamma, A \vdash \Delta}\ (WL)$ | $\dfrac{\Gamma \vdash \Delta}{\Gamma \vdash A, \Delta}\ (WR)$ |
| $\dfrac{\Gamma, A, A \vdash \Delta}{\Gamma, A \vdash \Delta}\ (CL)$ | $\dfrac{\Gamma \vdash A, A, \Delta}{\Gamma \vdash A, \Delta}\ (CR)$ |
| $\dfrac{\Gamma_1, A, B, \Gamma_2 \vdash \Delta}{\Gamma_1, B, A, \Gamma_2 \vdash \Delta}\ (PL)$ | $\dfrac{\Gamma \vdash \Delta_1, A, B, \Delta_2}{\Gamma \vdash \Delta_1, B, A, \Delta_2}\ (PR)$ |

# Intuition

- Logical rules introduce new formula either on the left or the right. They maintain a logical invariant just like the Natural Deduction rules.
  - What is the invariant?
- Structural rules manipulate the formula regardless of the shape or connective that the formula have.

# Intuiton 2

- Think of the rules as instructions for constructing a proof.
- Some of the instructions are ambiguous. There may be many ways to follow them
- Next time we will study automated methods for finding a proof.