COMP 527: Logic and Computation

Julian Lore

Last updated: April 30, 2019

Contents

1 01/08/19

Goal:

- Introduction to the proof theoretic foundations
- Study the relationship to programming languages and their design

Schedule:

- Week 1: Natural Deduction
- Week 2: Tutorial (Jacob) on using Tutch (proof assistant, program on writing Natural Deduction proofs)
- Week 3: Proofs in the natural deduction system and programs in the λ-calculus
- Week 4: First Order Logic
- Week 5: Induction (how we can add induction to logic, corresponds to recursion)
- Week 6/7: Programming with dependent types
- Week 8: Sequent calculus
- Week 9: Consistency
- Week 10: Proof search
- Week 11/12: Linear logic, modal logic or temporal logic

1.1 Natural Deduction

- G. Gentzen started describing it in the mid 30s
- Martin Löf continued in the mid 80s

Motivation To design a modular system for reasoning. Want to capture the reasoning of mathematicians (that's why it's called natural). It is modular because we define the meaning of each connective by themselves (will not refer to any other connective in the definition).

Judgment "Something we know" or "something that is evident"
Ex. A true, "The proposition A is true" (Semantics)
A wf, "The proposition A is syntactically well-formed" (Syntax)
A true @ t, "The proposition A is true at time t"
Can describe both aspects (semantics and syntax) as judgment

Example Grammar Proposition $A, B := T \mid \bot \mid A \wedge B \mid \ldots$

$$\frac{}{\text{T wf}}$$

$$\frac{}{\bot \text{ wf}}$$

$$\frac{\text{A wf} \qquad \text{B wf}}{\text{A} \wedge \text{B wf}}$$

$$\frac{J_1 \qquad \ldots \qquad J_n}{J}$$

J_i are the premises, J is the conclusion
A true

- Introduction rules: How can we conclude A
- Elimination rules: What information can we extract from A? (i.e. T, $\bot$, $A \wedge B$, …)

$$\frac{}{\text{T true}}\ \text{T I (Introduction)}$$

No elim rule for T

Conjunction

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

$$\frac{A \wedge B \text{ true}}{A \text{ true}} \wedge\text{E}_l \text{ (Elimination)}$$

$$\frac{A \wedge B \text{ true}}{B \text{ true}} \wedge\text{E}_r$$

What can we now prove?

$$\frac{\dfrac{}{T \text{ true}}\, T\ \text{I} \qquad \dfrac{}{T \text{ true}}\, T\ \text{I}}{T \wedge T \text{ true}} \wedge\ \text{I}$$

Is the following a proof?

$$\frac{\dfrac{A \wedge (B \wedge C) \text{ true}}{B \wedge C \text{ true}} \wedge\text{E}_r}{B \text{ true}} \wedge\ \text{E}_l$$

No, how do we know $A \wedge (B \wedge C)$ is true?

Given the assumption (hypothesis) $A \wedge (B \wedge C)$ true, we can construct a proof for B true (reasoning by assumption/hypothetical reasoning/derivation).

$$\begin{array}{c} \dfrac{}{J_1}\, u_1 \qquad \ldots \qquad \dfrac{}{J_n}\, u_n \\ \vdots \\ J \end{array}$$

Implication

$$\begin{array}{c} \dfrac{}{A \text{ true}}\, u \\ \vdots \\ \dfrac{B \text{ true}}{A \supset B \text{ true}} \supset \text{I}^u \end{array}$$

$$\frac{A \supset B \text{ true} \qquad A \text{ true}}{B \text{ true}} \supset \text{E}$$

We do not include $A \supset B$ true, B false, implies A false, because we did not add a judgment for false, only talking about things that are true. If we cannot say something is true, it is implied that it is false.

Let's prove:

$$\frac{\dfrac{\dfrac{\dfrac{}{A \text{ true}}\, u \qquad \dfrac{}{B \text{ true}}\, v}{A \wedge B \text{ true}} \wedge\ \text{I}}{B \supset A \wedge B \text{ true}} \supset I^v}{A \supset (B \supset A \wedge B) \text{ true}} \supset I^u$$

Observations

1. Order of assumptions does not matter.

2. Do I have to use an assumption? No. This is called weakening.
   Floating assumption that isn't used:

$$\frac{}{B \text{ true}}\, v$$

$$\frac{\dfrac{\dfrac{}{A \text{ true}}\, u}{B \supset A \text{ true}} \supset I^v}{A \supset B \supset A \text{ true}} \supset I^u$$

3. Can we use assumptions more than once? Yes. This is called strengthening (you contract multiple of these assumptions into one).

$$\frac{\dfrac{\dfrac{}{A \text{ true}}\, u \quad \dfrac{}{A \text{ true}}\, u}{A \wedge A \text{ true}} \wedge\text{I}}{A \supset (A \wedge A) \text{ true}} \supset I^u$$

   This is a program that takes in an input and returns it twice.

A simple proof:

$$\frac{\dfrac{\dfrac{\dfrac{}{A \wedge B \text{ true}}\, u}{A \text{ true}} \wedge \text{E}_l \quad \dfrac{\dfrac{}{A \wedge B \text{ true}}\, u}{B \text{ true}} \wedge \text{E}_r}{B \wedge A \text{ true}} \wedge\text{I}}{(A \wedge B) \supset (B \wedge A) \text{ true}} \supset I^u$$

2 01/10/18

2.1 Natural Deduction

A true

Conjunction

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

$$\frac{A \wedge B \text{ true}}{A \text{ true}} \wedge\text{E}_l \text{ (Elimination)}$$

$$\frac{A \wedge B \text{ true}}{B \text{ true}} \wedge\text{E}_r$$

Implications This discharges the assumption u and is the only rule we have right now that discharges assumptions.

$$\dfrac{\dfrac{}{A \text{ true}}\, u \\ \vdots \\ B \text{ true}}{A \supset B \text{ true}} \supset \mathrm{I}^u$$

$$\dfrac{A \supset B \text{ true} \qquad A \text{ true}}{B \text{ true}} \supset \mathrm{E}$$

Truth

$$\dfrac{}{\mathrm{T} \text{ true}} \mathrm{T\ I}$$

To get the smallest complete example, we need to introduce $\bot$.

Falsehood No Intro-Rule for $\bot$.

$$\dfrac{\bot \text{ true}}{A \text{ true}} \bot \mathrm{E}$$

Define Negation as an abbreviation (notational definition)

$$\neg A \equiv A \supset \bot$$

$$\dfrac{\dfrac{\dfrac{\dfrac{}{A \wedge \neg A}\,\mathrm{u}}{A} \wedge \mathrm{E}_l \qquad \dfrac{\dfrac{}{A \wedge \neg A}\,\mathrm{u}}{\neg A \equiv (A \supset \bot)} \wedge \mathrm{E}_r}{\dfrac{\bot}{\neg(A \wedge \neg A) \text{ true}} \supset \mathrm{I}^u} \supset \mathrm{E}$$

If we could prove $\bot$ from no assumptions, then our system would be inconsistent.

Examples

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{A \supset (B \wedge C) \text{ true}}\,\mathrm{u} \qquad \dfrac{}{A \text{ true}}\,\mathrm{v}}{B \wedge C \text{ true}} \supset \mathrm{E}}{B \text{ true}} \wedge \mathrm{E}_l}{A \supset B \text{ true}} \supset \mathrm{I}^v \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{A \supset (B \wedge C) \text{ true}}\,\mathrm{u} \qquad \dfrac{}{A \text{ true}}\,\mathrm{v}}{B \wedge C} \supset \mathrm{E}}{C \text{ true}} \wedge \mathrm{E}}{A \supset C \text{ true}} \supset \mathrm{I}^v}{(A \supset B) \wedge (A \supset C) \text{ true}} \wedge \mathrm{I}}{(A \supset (B \wedge C)) \supset ((A \supset B) \wedge (A \supset C)) = (A \wedge \neg A) \supset \bot \text{ true}} \supset \mathrm{I}^u$$

Note that the assumptions u and v are used twice, but in different areas.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{A}\,\mathrm{u} \quad \dfrac{}{\neg A}\,\mathrm{v}}{\bot}\supset \mathrm{E}}{\neg\neg A = \neg A \supset \bot \text{ true}}\supset \mathrm{I}^{v}}{A \supset \neg\neg A \text{ true}}\supset \mathrm{I}^{u}$$

Is $A = \neg\neg A$? We cannot prove this in this logic unless we assume an axiom (although it is true for classical logic).

For a proof, we must discharge all assumptions we make.

Local Soundness How do we know the rules we defined for natural deduction make sense?

Elimination Rules are not too strong (they don't allow us to conclude more than we should be able to). If the rules are too strong, we'll be able to prove things that we shouldn't be able to, more than we had to start with.

Given a proof $\mathcal{D}$ for A true and a proof $\mathcal{E}$ for B true, we can prove A true.

$$\dfrac{\dfrac{\begin{array}{cc}\mathcal{D} & \mathcal{E} \\ A \text{ true} & B \text{ true}\end{array}}{A \wedge B \text{ true}}\wedge \mathrm{I}}{A \text{ true}}\wedge \mathrm{E} \implies \begin{array}{c}\mathcal{D} \\ A \text{ true}\end{array}$$

$$\dfrac{\dfrac{\begin{array}{cc}\mathcal{D} & \mathcal{E} \\ A \text{ true} & B \text{ true}\end{array}}{A \wedge B \text{ true}}\wedge \mathrm{I}}{B \text{ true}}\wedge \mathrm{E} \implies \begin{array}{c}\mathcal{E} \\ B \text{ true}\end{array}$$

Local Completeness Elimination Rules are not too weak, i.e. we are expanding proofs.

$$\begin{array}{c}\mathcal{D} \\ A \wedge B \text{ true}\end{array} \implies \dfrac{\dfrac{\begin{array}{c}\mathcal{D} \\ A \wedge B \text{ true}\end{array}}{A \text{ true}}\wedge \mathrm{E}_l \quad \dfrac{\begin{array}{c}\mathcal{D} \\ A \wedge B \text{ true}\end{array}}{B \text{ true}}\wedge \mathrm{E}_r}{A \wedge B \text{ true}}\wedge \mathrm{I}$$

Local Soundness (Implication)

$$\dfrac{\dfrac{\begin{array}{c}\dfrac{}{A \text{ true}}\,\mathrm{u} \\ \vdots \\ \mathcal{D}^{u} \\ B \text{ true}\end{array}}{A \supset B \text{ true}}\supset \mathrm{I}^{u} \quad \begin{array}{c}\mathcal{E} \\ A \text{ true}\end{array}}{B \text{ true}}\supset \mathrm{E} \implies \begin{array}{c}\mathcal{E} \\ A \text{ true} \\ [\mathcal{E}/u](\mathcal{D}) \\ B \text{ true}\end{array}$$

Substitution Principle

If $\begin{array}{c}\dfrac{}{A \text{ true}}\,\mathrm{u} \\ \vdots \\ \mathcal{D} \\ B \text{ true}\end{array}$ and $\begin{array}{c}\mathcal{E} \\ A \text{ true}\end{array}$, then $\begin{array}{c}\mathcal{E} \\ A \text{ true} \\ \mathcal{D} \\ B \text{ true}\end{array}$

Basically, if we have a proof for A true and from A true we can show B true, then we have a proof for B true.

Local Completeness

$$\begin{array}{c}\alpha\\ A \supset B \text{ true}\end{array} \implies \dfrac{\dfrac{\begin{array}{c}\alpha\\ A \supset B \text{ true}\end{array} \qquad \dfrac{}{A \text{ true}}\,\mathrm{u}}{B \text{ true}} \supset \mathrm{E}}{A \supset B \text{ true}} \supset \mathrm{I}^u$$

So we can prove $A \supset B$ from $A \supset B$.

Another form of $\wedge$ elimination:

$$\dfrac{\begin{array}{c}\dfrac{}{A \text{ true}}\,\mathrm{u} \qquad \dfrac{}{B \text{ true}}\,\mathrm{v}\\ \vdots\\ A \wedge B \text{ true} \qquad\qquad C \text{ true}\end{array}}{C \text{ true}} \wedge E^{u,v}$$

We could use this new rule and check for local soundness and completeness like above.

Could have unused, floating assumption:

$$\dfrac{}{A \text{ true}}\,\mathrm{a}$$

$$\dfrac{\dfrac{\dfrac{}{A \wedge B}\,\mathrm{u}}{B \text{ true}}\,\mathrm{b} \qquad \begin{array}{c}\vdots\\ A \text{ true}\end{array}}{\dfrac{B \wedge A \text{ true}}{A \wedge B \supset B \wedge A \text{ true}} \supset \mathrm{I}^u}$$

2.2 Disjunction

Intro-Rules:

$$\dfrac{A \text{ true}}{A \vee B \text{ true}} \vee \mathrm{I}_l$$

$$\dfrac{B \text{ true}}{A \vee B \text{ true}} \vee \mathrm{I}_r$$

Might be tempted to make the following rule:

$$\dfrac{\dfrac{\dfrac{B \text{ true}}{A \vee B \text{ true}} \vee \mathrm{I}_r}{A \text{ true}}\,?$$

This is locally unsound! Got information we didn't start with.

We also cannot introduce:

$$\dfrac{A \vee B}{\neg A \supset B \text{ true}}$$

This violates our principle of modularity, as we don't want to refer to a different kind of connective.

Our elimination rule:

$$\dfrac{A \vee B \text{ true} \qquad \begin{array}{c}\overline{A \text{ true}}\,\text{u} \\ \vdots \\ C \text{ true}\end{array} \qquad \begin{array}{c}\overline{B \text{ true}}\,\text{v} \\ \vdots \\ C \text{ true}\end{array}}{C \text{ true}} \vee \text{E}^{u,v}$$

Example

$$\dfrac{\dfrac{\dfrac{\dfrac{}{A \vee (B \wedge C) \text{ true}}\,\text{u} \quad \dfrac{\dfrac{}{A \text{ true}}\,\text{a}}{A \vee C \text{ true}} \vee \text{I}_l \quad \dfrac{\dfrac{\dfrac{}{B \wedge C \text{ true}}\,\text{b}}{B \text{ true}} \wedge \text{E}_l}{A \vee B \text{ true}} \vee \text{I}_r}{A \vee B \text{ true}} \vee \text{E} \qquad \begin{array}{c}\vdots \\ A \vee C \text{ true}\end{array}}{(A \vee B) \wedge (A \vee C)} \vee \text{I}}{(A \vee (B \wedge C)) \supset (A \vee B) \wedge (A \vee C)} \supset \text{I}$$

3 01/15/19

Recall: We defined $\neg A$ as $A \supset \bot$. But, there are other ways you can introduce (define) negation.

3.1 Negation

$$\dfrac{\begin{array}{c}\overline{A \text{ true}}\,\text{u} \\ \vdots \\ p \text{ true}\end{array}}{\neg A \text{ true}} \neg \text{I}^{u,p}$$

If we can prove any parameter p from A true, then $\neg A$ is true. This discharges the assumption u and p.

$$\dfrac{\neg A \text{ true} \qquad A \text{ true}}{C \text{ true}} \neg \text{E}$$

For these new rules, as usual, we want to prove that they are locally sound and complete. In general, for local soundness, you want to introduce the rule and then eliminate it and show that you've learned nothing new.

Local Soundness

$$\dfrac{\dfrac{\begin{array}{c}\overline{A \text{ true}}\,\text{u} \\ D \\ p \text{ true}\end{array}}{\neg A \text{ true}} \neg \text{I}^{p,u} \qquad \begin{array}{c}\mathcal{E} \\ A \text{ true}\end{array}}{C \text{ true}} \neg \text{E} \quad \Longrightarrow \quad \begin{array}{c}[\mathcal{E}/u, C/p]D \\ C \text{ true}\end{array}$$

Local Completeness

$$\dfrac{\mathcal{D}}{\neg A \text{ true}} \implies \dfrac{\dfrac{\mathcal{D}}{\neg A \text{ true}} \quad \dfrac{}{A \text{ true}}\,\text{u}}{\dfrac{p \text{ true}}{\neg A \text{ true}}\,\neg \text{ I}^{p,u}}\,\neg \text{ E}$$

3.2 Classical Reasoning

Basic idea behind proof by contradiction, you assume $\neg A$, arrive at a contradiction and then you can prove A true.

$$\dfrac{\dfrac{}{\neg A \text{ true}}\,\text{u}}{\dfrac{\begin{array}{c}\vdots\\ p \text{ true}\end{array}}{A \text{ true}}\,C^{u,p}}$$

where $C^{u,p}$ means a proof by contradiction under u and p.

Law of Excluded Middle Want to show $\neg A \vee A$ is true for all A.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\neg(\neg A \vee A) \text{ true}}\,\text{u} \quad \dfrac{\dfrac{}{\neg A}\,\text{v}}{\neg A \vee A}\vee \text{I}_l}{p \text{ true}}\,\neg \text{ E}}{\dfrac{A}{\neg A \vee A}\vee \text{I}_r}\,C^{v,p} \qquad \dfrac{}{\neg(\neg A \vee A) \text{ true}}\,\text{u}}{\dfrac{q \text{ true}}{\neg A \vee A}\,C^{u,q}}\,\neg \text{ E}$$

```
% classical keyword to say we are under classical logic
classical proof lem : ~A | A =
begin
[ ~(~A | A);
  [ ~A;
        ~A | A;
        F ];
  A;
  ~A | A;
  F ];
~ A | A
end;
```

```
proof disj_assoc : A | (B | C) => (A | B) | C =
begin
[ A | (B | C);
  [ A;
        A | B;
        (A | B) | C ];
  [ B | C;
        [ B;
          A | B;
          (A | B) | C ];
        [ C ;
          (A | B) | C ];
        (A | B) | C
  ];
  (A | B) | C
];
A | (B | C) => (A | B) | C
end;

% Disjunction distributes over conjunctions
proof disj_conj : A | (B & C) => (A | B) & (A | C) =
begin
[ A | (B & C);
  [ A;
        A | B;
        A | C;
        (A | B) & (A | C) ];
  [ B & C;
        B;
        C;
        A | B;
        A | C;
        (A | B) & (A | C) ];
  (A | B) & (A | C) ];
A | (B & C) => (A | B) & (A | C)
```

```
end;

proof conj_assoc : (A & (B & C)) => (A & B) & C =
begin
[ A & (B & C);
  A;
  B & C;
  B;
  C;
  A & B;
  (A & B) & C ];
(A & (B & C)) => (A & B) & C
end;

proof conj_disj : A & (B | C) => A & B | A & C =
begin
[ A & (B | C);
  A;
  B | C;
  [ B;
        A & B;
        A & B | A & C ];
  [ C;
        A & C;
        A & B | A & C ];
  A & B | A & C ];
A & (B | C) => A & B | A & C
end;


TUTCH 0.52 beta, $Date: 2002/10/24 19:25:49 $
[Opening file lem.tut]

Proving lem: ~A | A ... (classically)
  1  [ ~(~A | A);
  2    [ ~A;
  3      ~A | A;          by OrIL 2
```

```
  4       F ];              by ImpE 1 3
  5     A;                  by Class 4
  6     ~A | A;             by OrIR 5
  7     F ];                by ImpE 1 6
  8   ~A | A                by Class 7
QED

Proving disj_assoc: A | B | C => (A | B) | C ...
  1   [ A | B | C;
  2     [ A;
  3       A | B;                          by OrIL 2
  4       (A | B) | C ];                  by OrIL 3
  5     [ B | C;
  6       [ B;
  7         A | B;                        by OrIR 6
  8         (A | B) | C ];                by OrIL 7
  9       [ C;
 10         (A | B) | C ];                by OrIR 9
 11       (A | B) | C ];                  by OrE 5 8 10
 12     (A | B) | C ];                    by OrE 1 4 11
 13   A | B | C => (A | B) | C            by ImpI 12
QED

Proving disj_conj: A | B & C => (A | B) & (A | C) ...
  1   [ A | B & C;
  2     [ A;
  3       A | B;                                  by OrIL 2
  4       A | C;                                  by OrIL 2
  5       (A | B) & (A | C) ];                    by AndI 3 4
  6     [ B & C;
  7       B;                                      by AndEL 6
  8       C;                                      by AndER 6
  9       A | B;                                  by OrIR 7
 10       A | C;                                  by OrIR 8
 11       (A | B) & (A | C) ];                    by AndI 9 10
 12     (A | B) & (A | C) ];                      by OrE 1 5 11
```

```
 13  A | B & C => (A | B) & (A | C)          by ImpI 12
QED
[Closing file lem.tut]
```

3.3 NAND

$A \overline{\wedge} B \equiv \neg(A \wedge B)$

Showing this introduction rule is often a midterm question. Often, people make the mistake of using $A \wedge B$, but we cannot introduce other connectives, make sure A and B are separate.

$$\dfrac{\dfrac{}{A \text{ true}}\,\text{u} \quad \dfrac{}{B \text{ true}}\,\text{v}}{\dfrac{\begin{array}{c}\vdots \\ p \text{ true}\end{array}}{A \overline{\wedge} B \text{ true}}\,\overline{\wedge}\,\text{I}}$$

$$\dfrac{A \text{ true} \qquad B \text{ true} \qquad A \overline{\wedge} B \text{ true}}{C \text{ true}}\,\overline{\wedge}\,\text{E}$$

Local Soundness Get rid of assumptions by substituting $\mathcal{E}_1, \mathcal{E}_2$ and C into $\mathcal{D}$.

$$\dfrac{\dfrac{\dfrac{}{A \text{ true}}\,\text{u} \quad \dfrac{}{B \text{ true}}\,\text{v}}{\dfrac{\begin{array}{c}\mathcal{D} \\ p \text{ true}\end{array}}{A \overline{\wedge} B \text{ true}}\,\overline{\wedge}\,\text{I}^{u,v,p} \qquad \begin{array}{c}\mathcal{E}_1 \\ A \text{ true}\end{array} \qquad \begin{array}{c}\mathcal{E}_2 \\ B \text{ true}\end{array}}{C \text{ true}} \quad \Longrightarrow \quad \begin{array}{c}[\mathcal{E}_1/u, \mathcal{E}_2/v, C/p]\mathcal{D} \\ C \text{ true}\end{array}$$

Local Completeness

$$\begin{array}{c}\mathcal{D} \\ A \overline{\wedge} B\end{array} \quad \Longrightarrow \quad \dfrac{\dfrac{}{A \text{ true}}\,\text{u} \quad \dfrac{}{B \text{ true}}\,\text{v} \qquad \begin{array}{c}\mathcal{D} \\ A \overline{\wedge} B \text{ true}\end{array}}{\dfrac{p \text{ true}}{A \overline{\wedge} B}\,\overline{\wedge}\,\text{I}^{u,v,p}}\,\overline{\wedge}\,\text{E}$$

Look for other connectives and do these proofs, as they will probably be on the midterm.

4 01/17/19

4.1 Context

So we've seen several rules so far:

$$\dfrac{}{T \text{ true}}$$

$$\frac{\bot \text{ true}}{C \text{ true}}$$

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

$$\frac{A \wedge B \text{ true}}{A \text{ true}} \wedge \text{E}_l$$

$$\frac{A \vee B \text{ true} \qquad \begin{array}{c}\overline{A \text{ true}}^{\,\text{u}} \\ \vdots \\ C \text{ true}\end{array} \qquad \begin{array}{c}\overline{B}^{\,\text{v}} \\ \vdots \\ C \text{ true}\end{array}}{C \text{ true}} \vee \text{E}^{u,v}$$

We are now going to extend these rules with contexts, which allows us to know what we've proved and what assumptions we have at each step.

Instead of A true, we will write $\Gamma \vdash A$ true. This means:

"A is true in context Γ."

$$\Gamma ::= \cdot \mid \Gamma, u : A \text{ true}$$

So:

$$\frac{\begin{array}{c}\overline{A \text{ true}}^{\,\text{u}} \\ \vdots \\ B \text{ true}\end{array}}{A \supset B \text{ true}} \supset \text{I}^u \quad \xrightarrow{+\Gamma} \quad \frac{\Gamma, u : A \text{ true} \vdash B \text{ true}}{\Gamma \vdash A \supset B \text{ true}}$$

$$\frac{}{\Gamma \vdash T \text{ true}}$$

$$\frac{\Gamma \vdash \bot \text{ true}}{\Gamma \vdash C \text{ true}}$$

$$\frac{\Gamma \vdash A \text{ true} \qquad \Gamma \vdash B \text{ true}}{\Gamma \vdash A \wedge B} \wedge \text{I}$$

$$\frac{\Gamma \vdash A \wedge B \text{ true}}{\Gamma \vdash A \text{ true}} \wedge \text{E}_l$$

$$\frac{\Gamma \vdash A \vee B \text{ true} \qquad \Gamma, u : A \text{ true} \vdash C \qquad \Gamma, v : B \text{ true} \vdash C}{\Gamma \vdash C \text{ true}} \vee \text{E}^{u,v}$$

For something to be derived from a context:

$$\frac{u : A \text{ true} \in \Gamma}{\Gamma \vdash A \text{ true}} \text{H}$$

Example

$$\dfrac{\dfrac{\dfrac{\dfrac{A \in \Gamma, A, B}{\Gamma, A, B \vdash \text{A}}\,\text{H} \quad \dfrac{B \in \Gamma, A, B}{\Gamma, A, B \vdash \text{B}}\,H}{\Gamma, u : A \text{ true}, B \text{ true} \vdash A \wedge B}\,\wedge \text{I}}{\Gamma, u : A \text{ true} \vdash B \supset A \wedge B}\,\supset \text{I}}{\Gamma \vdash A \supset B \supset A \wedge B}\,\supset \text{I}$$

4.2 The Curry-Howard Correspondence

We can see that natural deduction is the same as functional programming:

Logic	Types
$\bot$	unit
$A \wedge B$	$A \times B$
$A \vee B$	$A + B$
$A \supset B$	$A \implies B$
$\bot$	$\emptyset$
proofs	programs
checking a proof	type checker

$\emptyset$ is the empty type. There is no proof for $\bot$, which is why there is no program for $\emptyset$, the two sides are equivalent.

Proof terms capture the structure of a derivation.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma, x : A, y : B \vdash x : A}\,\text{H} \quad \dfrac{}{\Gamma, x : A, y : B \vdash y : B}\,\text{H}}{\Gamma, x : A, y : B \vdash (x, y) : A \times B}\,\times \text{I}}{\Gamma, x : A \vdash fn\ y \implies (x, y) : B \implies A \times B}\,\implies \text{I}^{y}}{\Gamma \vdash fn\ x \implies fn\ y \implies (x, y) : A \implies B \implies A \times B}\,\implies \text{I}^{x}$$

$\Gamma \vdash M : A$ means:
"M is a proof term (program) for proposition (of type) A."

4.3 Proof Terms

Let's upgrade the rules we mentioned earlier to types:

$$\dfrac{}{\Gamma \vdash () : unit}$$

$$\dfrac{\Gamma \vdash M : \emptyset}{\Gamma \vdash abort\ M : C}$$

$$\frac{\Gamma \vdash M_1 : A \text{ true} \qquad \Gamma \vdash M_2 : B \text{ true}}{\Gamma \vdash (M_1, M_2)\ A\ B \text{ true}} \times \mathrm{I}$$

$$\frac{\Gamma \vdash M : A \times B}{\Gamma \vdash fst\ M : A} \times \mathrm{E}_l$$

$$\frac{\Gamma \vdash M : A \times B}{\Gamma \vdash snd\ M : B} \times \mathrm{E}_r$$

$$\frac{\Gamma \vdash M : A + B \text{ true} \qquad \Gamma, u : A \vdash N_1 : C \qquad \Gamma, v : B \vdash N_2 : C}{\begin{array}{l}\Gamma \vdash \text{case } M \text{ of} \\ \quad \text{inl } u \implies N_1 \\ \quad \text{inr } v \implies N_2\end{array}} + \mathrm{E}$$

(Where inl is inject left and inr is inject right) Note that this is pattern matching!

$$\frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash (fn\ x \implies M) : A \implies B}$$

$$\frac{\Gamma \vdash M : A}{\Gamma \vdash inl\ M : A + B} + \mathrm{I}_l$$

$$\frac{\Gamma \vdash M : B}{\Gamma \vdash inr\ M : A + B} + \mathrm{I}_r$$

$$\frac{\Gamma \vdash M : A \implies B \qquad \Gamma \vdash N : A}{\Gamma \vdash M\_N : B}$$

where _ is a space.

Example We did this on Tuesday, now we'll see it with proof terms.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma, x : A \times B \vdash x : A \times B}\mathrm{H}}{\Gamma, x : A \times B \vdash snd\ x : B} \times \mathrm{E} \qquad \dfrac{\dfrac{}{\Gamma, x : A \times B \vdash x : A \times B}\mathrm{H}}{\Gamma, x : A \times B \vdash fst\ x : A} \times \mathrm{E}}{\Gamma, x : A \times B \vdash (snd\ x, fst\ x) : B \times A} \times \mathrm{I}}{\Gamma \vdash fn\ x \implies (snd\ x, fst\ x) : A \times B \implies B \times A} \implies \mathrm{I}$$

```
annotated proof pair : A => B => A & B =
begin
[ x : A;
  [ y : B;
   (x, y) : A & B ];
 fn y => (x, y) : B => A & B ];
fn x => fn y => (x, y) : A => B => A & B
end;

annotated proof conj_comm : A & B => B & A =
begin
% You cannot do pattern matching when introducing variables, i.e.
    (x, y) : A & B
[ x : A & B;
  fst x : A;
  snd x : B;
  (snd x, fst x) : B & A ];
fn x => (snd x, fst x) : A & B => B & A
end;

annotated proof disj_comm : A | B => B | A =
begin
[ x : A | B;
  [ a : A;
        inr a : B | A  ];
  [ b : B;
        inl b : B | A  ];
  case x of
  inl a => inr a |
  inr b => inl b
  end : B | A ];
fn x => case x of
inl a => inr a |
inr b => inl b
end : A | B => B | A
end;
```

```
% Terms only, tutch can infer the proof tree itself
term disj_comm_tm : A | B => B | A =
fn x =>
case x of
inl a => inr a |
inr b => inl b
end;

annotated proof conj_disj : A & (B | C) => A & B | A & C =
begin
[ x : A & (B | C);
  fst x : A;
  snd x : B | C;
  [ b : B;
        [ c : C;
          (fst x, c) : A & C ];
        (fst x, b) : A & B;
        inl (fst x, b) : A & B | A & C ];
  [ c : C;
        [ b : B;
          (fst x, b) : A & B];
        (fst x, c) : A & C;
        inr (fst x, c) : A & B | A & C ];
  case snd x of
  inl b => inl (fst x, b) |
  inr c => inr (fst x, c)
  end : A & B | A & C ];
fn x => case snd x of
inl b => inl (fst x, b) |
inr c => inr (fst x, c)
end : A & (B | C) => A & B | A & C
end;

term conj_disj_tm : A & (B | C) => A & B | A & C =
fn x => case snd x of
```

```
inl b => inl (fst x, b) |
inr c => inr (fst x, c)
end;
```

5 01/22/19

5.1 Metatheory

Proving things (properties) about a given theory, this will eventually lead us to completeness.

So far, we've seen:

Propositions/Types $A, B := T \mid \bot \mid A \supset B \mid A \wedge B \mid A \vee B$

Proof Terms $M, N := \langle\rangle \mid abort^A\ M \mid \underbrace{\lambda x : A.M}_{fn\ x \Longrightarrow M} \mid M\ N$

$$\frac{M : \bot}{abort^A\ M : A}$$

Note that: $M : A$ means: Term M has type A or M is a proof witness for the proposition A

Types that have one type are sometimes just called singleton types.

Earlier we saw:

$$\frac{\dfrac{}{A_1 \text{ true}}\,u_1 \quad \cdots \quad \dfrac{}{A_n \text{ true}}\,u_n \qquad \begin{matrix}\ddots & & \ddots \\ & \cdots & \end{matrix}}{C \text{ true}}$$

More compactly: Context $\Gamma := \cdot \mid T, A \text{ true}$

$$\Gamma \underbrace{\vdash}_{\text{turnstyle}} C \text{ true}$$

Reminder about Contexts Context $\Gamma := \cdot \mid \Gamma, x : A$

$$\frac{\Gamma \vdash M : \bot}{\Gamma \vdash abort^A\ M : A}$$

$$\frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x : A.M : A \supset B}$$

$$\frac{\Gamma \vdash M : A \supset B \qquad \Gamma \vdash N : A}{\Gamma \vdash M\ N : B}$$

$$\frac{x : A \in \Gamma}{\Gamma \vdash x : A}$$

$$\frac{\Gamma \vdash M : A \qquad \Gamma \vdash N : B}{\Gamma \vdash \langle M, N \rangle : A \wedge B}$$

$$\frac{\Gamma \vdash M : A \wedge B}{\Gamma \vdash fst\ M : A}$$

Note that, some may be inclined to make $A \wedge B$ have a type of some pair for $\wedge$ E, i.e. $\langle M, N \rangle$, but we don't want to limit ourselves to a pair (we also don't yet know it's a pair), so we cannot make it a pair right off the bat.

$$\frac{\Gamma \vdash M : A \wedge B}{\Gamma \vdash snd\ M : B}$$

What would we like to compute with these rules?

General form: $M \implies M'$, Term M reduces to M'.

$fst\ \langle M, N \rangle \implies M$

$snd\ \langle M, N \rangle \implies N$

So we can now show things like local soundness with terms:

$$\dfrac{\dfrac{\begin{matrix}\mathcal{D}_1 \\ \Gamma \vdash M : A\end{matrix} \qquad \begin{matrix}\mathcal{D}_2 \\ \Gamma \vdash N : B\end{matrix}}{\Gamma \vdash \langle M, N \rangle : A \wedge B}\wedge \text{I}}{\Gamma \vdash fst\ \langle M, N \rangle : A}\wedge \text{E}_l \implies \begin{matrix}\mathcal{D}_1 \\ M : A\end{matrix}$$

Local Soundness here corresponds to Subject Reduction/Type Preservation.

We've essentially proved that if we start with something well typed (of type A), we end up with something that is still well typed, we can show that our program is type **safe**.

Now if we have a function:

$(\lambda x : A.M)N \implies [N/x]M$

Simple Program

$$\begin{aligned} & fst\ ((\lambda x.\lambda y.\langle x, y \rangle)\langle\rangle\langle\rangle) \\ \implies & fst\ \langle\langle\rangle, \langle\rangle\rangle \\ \implies & \langle\rangle : T \end{aligned}$$

The type of this program is unit, because we have stated earlier that the type we start with is the type we end with. How do we know the first line is of type unit?

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{x:T,y:T\vdash x:T}\,\text{var}\quad\dfrac{}{x:T,y:T\vdash y:T}\,\text{var}}{x:T,y:T\vdash\langle x,y\rangle:T\wedge T}\wedge\text{I}}{x:T\vdash\lambda y:T\langle x,y\rangle:T\supset(T\wedge T)}\supset\text{I}}{\lambda x:T\lambda.y:T\langle x,y\rangle:T\supset T\supset(T\wedge T)}\supset\text{I}\qquad\dfrac{}{\langle\rangle:\text{T}}\,T\text{ I}}{(\lambda x:T\lambda.y:T\langle x,y\rangle):\langle\rangle}\supset\text{E}\qquad\dfrac{}{\langle\rangle:\text{T}}\,T\text{ I}}{((\lambda x:T.\lambda y:T\langle x,y\rangle)\langle\rangle)\langle\rangle:}\wedge\text{E}_l}{fst\ (\lambda x:T.\lambda y:T\langle x,y\rangle)\langle\rangle\langle\rangle:}$$

Note that we have:

If $\Gamma\vdash M:A$ and $M\implies M'$ then $\Gamma\vdash M':A$ (subject reduction)

If $\Gamma\vdash M':A$ and $M\implies M'$ then $\Gamma\vdash M:A$ (won't prove this second one, as it's much less obvious)

5.2 Congruence Rules

We can go into functions. This strategy is non-deterministic (will this terminate? Yes, but it is not obvious how to prove it. This is a metatheoretic property)

$$\frac{M\implies M'}{fst\ M\implies fst\ M'}$$

$$\frac{M\implies M'}{snd\ M\implies snd\ M'}$$

$$\frac{M\implies M'}{M\ N\implies M'\ N}$$

$$\frac{N\implies N'}{M\ N\implies M\ N'}$$

$$\frac{M\implies M'}{\lambda x:A.M\implies\lambda x:A.M'}$$

Syntactic Equality is important because we want to introduce substitution.

$\lambda x.x=\lambda y.y$ (the name of a bound var does not matter)

$\lambda x.\lambda y.x\neq\lambda x.\lambda y.y$

$\lambda x.\lambda y.x=\lambda x'.\lambda y.x'$ These are the same because x has the same binding site as x'

Terms that only differ in the names of the bound variables are considered equal. This is defined as α-renaming. Capture-avoiding substitution.

Note in the following expression, x is new ("fresh"). We cannot have two variables with the same name, if this was the case, we'd have to rename x.

$$\frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x : A.M : A \supset B}$$

Otherwise a rule such as the following would not give you a unique type.

$$\frac{x : A \in \Gamma}{\Gamma \vdash x : A}$$

Example of substitution What does this mean?

$[u/w](\lambda u.wu) \neq \lambda u.uu$

You cannot introduce a new variable with the same name as an existing one, this will have completely different meaning than you intended.

Solution: $[u/w](\lambda u.wu) \stackrel{\alpha}{=} [u/w](\lambda u'.wu') = \lambda u'.uu'$

Let's assume that $u : T \supset A$.

$[u/w](\underbrace{\lambda u : T.\ w\ u}_{\text{well-typed, } T \supset A}) = \lambda u : T.\ \underline{u}\ u$? This is ill-typed!

5.3 Restatement of Definitions Using Context

Weakening If $\Gamma \vdash M : A$ then $\Gamma, \Gamma' \vdash M : A$

Substitution If $\Gamma, x : A, \Gamma' \vdash M : B$ and $\Gamma \vdash N : A$ then $\Gamma, \Gamma' \vdash [N/x]M : B$

How do we prove this? Proof by structural on $\Gamma, x : A, \Gamma' \vdash M : B$

First we must consider each rule that could have been used to derive $M : B$.

Base case would be T, trivially true because there is nothing to substitute.

Definition of substitution

$[N/x]M = M'$, $[N/x]x = N$. Replace the "free" occurence of x in M with N

Case $\dfrac{\overset{\mathcal{D}'}{\Gamma, x : A, \Gamma' \vdash M : C \wedge D}}{\Gamma, x : A, \Gamma' \vdash fst\ M : C} \wedge \mathrm{E}_l$

$\Gamma, \Gamma' \vdash [N/x]M : C \wedge D$ by IH

$\Gamma, \Gamma' \vdash fst\ ([N/x]M) : C$ by $\wedge \mathrm{E}_l$

$\Gamma, \Gamma' \vdash [N/x](fst\ M) : C$ by definition of substitution

Case $\dfrac{\overset{\mathcal{D}_1}{\Gamma, x : A, \Gamma' \vdash M_1 : D \supset C} \qquad \overset{\mathcal{D}_2}{\Gamma, x : A, \Gamma' \vdash M_2 : D}}{\Gamma, x : A, \Gamma' \vdash M_1\ M_2 : C} \supset \mathrm{E}$

$\Gamma, \Gamma' \vdash [N/x]M_1 : D \supset C$ by IH ($\mathcal{D}_1$)

$\Gamma, \Gamma' \vdash [N/x]M_2 : D$ by IH ($\mathcal{D}_2$)

$\Gamma, \Gamma' \vdash [N/x]M_1\ [N/x]M_2 : C$ by $\supset$ E

$\Gamma, \Gamma' \vdash [N/x](M_1\ M_2) : C$

The more interesting case is with variables:

Case $\dfrac{x : A \in (\Gamma, x : A, \Gamma')}{\Gamma, x : A, \Gamma' \vdash x : A}$

$\Gamma \vdash N : A$ by assumption

$\Gamma, \Gamma' \vdash N : A$ by weakening

$\Gamma, \Gamma' \vdash [N/x]x : A$ by substitution definition

Case $\dfrac{y : B \in (\Gamma, x : A, \Gamma') \qquad x \neq y}{\Gamma, x : A, \Gamma' \vdash y : B}$

$y : B \in (\Gamma, \Gamma')$

$\Gamma, \Gamma' \vdash y : B$ by variable

Case $\dfrac{\Gamma, x : A, \Gamma', y : B \vdash M : C}{\Gamma, x : A, \Gamma' \vdash \lambda y : B.M : B \supset C}$

$\Gamma, \Gamma', y : B \vdash [N/x]M : C$ by IH.

$\Gamma, \Gamma', \lambda y : B.[N/x]M : B \supset C$ by $\supset$ I

$\Gamma \vdash N : A$ by assumption (i.e any var in N is in Γ). y does not occur in N

$\Gamma, \Gamma' \vdash [N/x](\lambda y : B.M) : B \supset C$ by substitution definition

6 01/24/19

6.1 More on Proof Terms

Last class we saw:

Prop/Types $A, B := T \mid \bot \mid A \wedge B \mid A \supset B \mid A \vee B$

Terms $M, N := \langle\rangle \mid abort^c\ M \mid \langle M, N\rangle \mid fst\ M \mid snd\ M \mid \lambda x.A.M \mid M\ N \mid inl^B\ M \mid inr^A\ B \mid (case\ M\ of\ inl\ x \implies N_1 \mid inr\ y \implies N_2)$

We can think of $M : A$ iff A true.

Warm-up

Program	**Proposition**
$\lambda x : A.\lambda y : B.\langle x, y\rangle$	$A \supset B \supset (A \wedge B)$
$\lambda x : A \supset B \wedge B \supset C.$ $\lambda y : A.(snd\ x)\ \underbrace{\underbrace{(\underbrace{(fst\ x)}_{A \supset B}\ y)}_{B}}$	$(A \supset B \wedge B \supset C) \supset A \supset C$
$\lambda x : A \vee B.\lambda y : C.$ $\langle case\ of\ inl\ a \implies inr\ a \mid inr\ b \implies inl\ b, y\rangle$	$(A \vee B) \supset C \supset (B \vee A) \wedge C$

	Prop	Type	
	T	unit	
Reminder	$A \wedge B$	$A * B$	The $+$ is usually hidden in programming languages, i.e.
	$A \supset B$	$A \to B$	
	$A \vee B$	$A + B$	

`type int_or_string = Int of int | Str of string` is actually `Int + String`.

Proposition	**Program**
$A \supset A$ (has many programs)	$\lambda x : A.x$ normal/"canonical form"
Programs on the right are syntactically different	$\lambda x : A.(\lambda y : A.y)x$
	$\lambda x : A.fst((\lambda y : A \wedge A.y)\langle x, x\rangle)$

We hope that all of the functions on the right can reduce to the normal form.

If $\Gamma \vdash M : A$ and $M \implies M'$ then $\Gamma \vdash M' : A$ } Subject Reduction/Type Preservation

By induction (case analysis) on $M \implies M'$

Case $fst\langle M, N\rangle \implies M$

$\Gamma \vdash fst\ \langle M, N\rangle : A$ by assumption

$\Gamma \vdash \langle M, N\rangle : A \wedge B$ by <u>inversion</u> on $\wedge\ \mathrm{E}_l$

Note the inversion rule tells us:

$$\frac{M : A \wedge B}{fst\ M : A} \wedge \mathrm{E}_l$$

Inversion Lemma: If $fst\ M : A$ then $M : A \wedge B$. This is obvious in this system because there is only one rule. If there was another rule that we could use, then we'd have to prove that it came from this inversion rule.

$\Gamma \vdash M : A$ by <u>inversion</u> of $\wedge I$

The concept of <u>inversion</u> is very important, we want to know where the term comes from.

Case $(\lambda x : A.M)N \implies [N/x]M$

$\Gamma \vdash (\lambda x : A.M)N : B$	by assumption
$\Gamma \vdash \lambda x : A.M : A \supset B,\ \Gamma \vdash N : A$	by inversion on $\supset$ E
$\Gamma, x : A \vdash M : B$	by inversion on $\supset$ I
$\Gamma \vdash [N/x]M : B$	by substitution lemma (on above line and $\Gamma \vdash N : A$)

Case $case\ (inl^B\ M)\ of\ inl\ x \implies N_1 \mid inr\ y \implies N_2 \implies [M/x]N_1$

$\Gamma \vdash case\ (inl^B\ M)\ of\ inl\ x \implies N_1 \mid inr\ y \implies N_2 : C$	by assumption
$\Gamma \vdash inl^B\ M : A \vee B$	by inversion $\vee$ E
$\Gamma, x : A \vdash N_1 : C$	
$\Gamma, y : B \vdash N_2 : C$	
$\Gamma \vdash M : A$	by inversion $\vee$ I_l
$\Gamma \vdash [M/x]N_1 : C$	by substitution

Case $\dfrac{M_1 \implies M'}{M_1\ M_2 \implies M_1'M_2}$

$\Gamma \vdash M_1\ M_2 : B$	by assumption
$\Gamma \vdash M_1 : A \supset B$ and $\Gamma \vdash M_2 : A$	by inversion $\supset$E
$\Gamma \vdash M_1' : A \supset B$	by IH
$\Gamma \vdash M_1'\ M_1 : B$	by $\supset$ E

The last step is forward reasoning instead of backward reasoning with inversions. Basically show what your case gives you and then apply IH to get what you want.

Case $\dfrac{M \implies M'}{\lambda x : A.M \implies \lambda x : A.M'}$

$\Gamma \vdash \lambda x : A.M : A \supset B$	by assumption
$\Gamma, x : A \vdash M : B$	by inversion on $\supset$ I
$\Gamma, x : A \vdash M' : B$	by IH
$\Gamma \vdash \lambda x : A.M' : A \supset B$	by $\supset$ I

Note that in trying to reduce $fst\ ((\lambda y : A \wedge A.y)\langle x, x\rangle)$, we will need to step twice:

$\implies \lambda x : A.fst\langle x, x\rangle$

$\implies \lambda x : A.x$

But so far we are only stepping once.

Can we prove the opposite? If $\Gamma \vdash M' : A$ and $M \implies M'$ then $\Gamma \vdash M : A$

Case $fst\langle M, N\rangle \implies M$

$\Gamma \vdash M : A$	by assumption
$\Gamma \vdash N : B$	How to show this?
$\Gamma \vdash \langle M, N\rangle : A \wedge B$	
$\Gamma \vdash fst\ \langle M, N\rangle : A$	To show

Cannot prove this syntactically right now.

Multi Step Want to step multiple times. $M \implies {}^*M'$, meaning M steps in more than one step.

$$\frac{}{M \implies {}^*M}\ \text{refl(ectivity)}$$

$$\frac{M \implies {}^*K \qquad K \implies {}^*N}{M \implies {}^*N}\ \text{Trans(itivity)}$$

$$\frac{M \implies N}{M \implies {}^*N}\ \text{one-step}$$

6.2 Normal Form

What is a normal form when no reduction is possible anymore?
$\lambda : A \supset B.\lambda y : A.x\ y$ is a normal form
So is $\lambda x : (A \supset A) \supset C.x \underbrace{(\lambda y : A.y)}_{A \supset A}$
Basically, you can no longer reduce the expression.

Normal Form $m, n := \lambda x : A.m \mid r$

Neutral Form $r := x \mid r\ m \mid fst\ r \mid snd\ r$

Questions

1. Can we always reach a normal form? (Does reduction terminate?)
2. Can we change the proof that correspond to the normal forms? "Normal Proof" (without any detours)
3. Is there a unique normal form that can be computed?
   If $M \implies {}^*N_1$ and $M \implies {}^*N_2$ then $N_1 = N_2$

Will look at 1 and 2 before the midterm.

7 01/29/19

7.1 First-order Logic

Reason about domains (nat, list, trees, programs, …)

How to reason in general

$$\forall x : \underbrace{\tau}_{\text{domain}} : A(x)\ true$$

For all elements in domain τ, $A(x)$ is true.

$$\forall x : nat\ even(x) \supset \neg odd(x)$$
$$\forall x : \tau : \forall y : \tau P(x) \supset P(y)$$
$$(\forall x : \tau.P(x)) \supset \exists x : \tau.P(x)$$

We define truth of a proposition for all possible domains.

Well-Formedness: A wf

$$\frac{\tau \text{ type } (\tau \text{ itself is wf}) \qquad \begin{array}{c}\overline{x : \tau} \\ \vdots \\ A(x) \text{ wf}\end{array}}{\forall x : \tau.A(x) \text{ wf}}$$

$$\frac{t_1 : \tau \qquad \ldots \qquad t_n : \tau}{P(t_1, \ldots, t_n) \text{ wf}}$$

$\forall x : nat\ even\ y \supset even(x)$, this is not well formed!

What is a type? What are its elements?

nat type $\quad \dfrac{}{z\text{:nat}} \qquad \dfrac{n\text{:nat}}{succ\ n\text{:nat}}$

nat_list type $\quad \dfrac{}{nil\text{:nat\_list}} \qquad \dfrac{n\text{:nat} \qquad t\text{:nat\_list}}{cons(n,t)\text{:nat\_list}}$

Prop. $A, B := T \mid \bot \mid A \wedge B \mid \forall x : \tau.A(x)$

Terms $t := x \mid f(t_1, \ldots, t_n)$

Context $\Gamma := \cdot \mid \Gamma, u : A \mid \Gamma, a : \tau$

Introduction We know what it means for something to be well formed, how do we show it? How do we introduce it? We could prove the predicate $A(x)$ for all x, but what if our domain is infinite? Often we may not even know much about our domain, so we can't prove it for each element either.

$$\dfrac{\dfrac{}{a : \tau}\\ \vdots \\ A(a)\ \text{true}}{\forall x : \tau . A(x)\ \text{true}}\ \forall \text{I}^a\ (a \text{ is a parameter})$$

This satisfies everything in τ because a is generic/new, could not exist before in the proof.

This rule is true whether you have something in τ or not, if it's impossible to get something from τ, everything in τ still satisfies $A(x)$ true.

Elimination Now that we have the information, what can we do with it?

$$\dfrac{\forall x : \tau . A(x)\ \text{true} \qquad t : \tau}{A(t)\ \text{true}}\ \forall\ \text{E}$$

We have to show that there is a t of type τ. This is similar to conjunction.

Example

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\forall x : \tau . A(x) \wedge B(x)\ \text{true}}\ \text{u} \quad \dfrac{}{a : \tau}}{A(a) \wedge B(a)\ \text{true}}\ \forall\ \text{E}}{A(a)\ \text{true}}\ \wedge\ \text{E}_l}{\forall x : \tau . A(x)}\ \forall\ \text{I}^a \quad \dfrac{\dfrac{\dfrac{\dfrac{}{\forall x : \tau . A(x) \wedge B(x)\ \text{true}}\ \text{u} \quad \dfrac{}{b : \tau}}{A(b) \wedge B(b)\ \text{true}}\ \forall\ \text{E}}{A(a)\ \text{true}}\ \wedge\ \text{E}_r}{\forall x : \tau . B(x)}\ \forall\ \text{I}^b}{((\forall x : \tau . A(x)) \wedge (\forall x : \tau . B(x))\ \text{true})}\ \wedge\ \text{I}}{(\forall x : \tau . A(x) \wedge B(x)) \supset ((\forall x : \tau . A(x)) \wedge (\forall x : \tau . B(x)))\ \text{true}}\ \supset \text{I}^u$$

The following proof is <u>invalid</u>! a is not fresh and we cannot prove $\forall x : \tau . \forall y : \tau . P(x) \supset P(y)$ true (it doesn't make sense after all).

$$\dfrac{\dfrac{P(a) \supset P(a)}{\forall y : \tau . P(a) \supset P(y)}\ \forall \text{I}^a \times}{\forall x : \tau . \forall y : \tau P(x) \supset P(y)\ \text{true}}\ \forall\ \text{I}^a$$

Overloaded functions

$$\dfrac{\dfrac{}{a : \tau}\\ \vdots \\ M : A(a)\ \text{true}}{\lambda a : \tau . M : \forall x : \tau . A(x)\ \text{true}}\ \forall \text{I}^a\ (a \text{ is a parameter})$$

λa is an overloaded function here.

$$\dfrac{M : \forall x : \tau . A(x)\ \text{true} \qquad t : \tau}{M\ t : A(t)\ \text{true}}\ \forall\ \text{E}$$

Types and Propositions with Quantifiers

Prop.	Types
T	unit
$A \wedge B$	$A * B$
$A \vee B$	$A + B$
$A \supset B$	$A \to B$
$\forall x : \tau.A(x)$	$\underbrace{\Pi x : \tau.A(x)}_{\text{dependent type "indexed"}}$

$\Pi n : nat$ rev:$'\alpha$ list (n) $\to' a$ list(n)

Local Soundness

$$\dfrac{\dfrac{\overline{a : \tau}\ \text{a}}{\dfrac{\mathcal{D}}{A(a)\ \text{true}}}\forall\ \text{I}^a \qquad \dfrac{\mathcal{E}}{t : \tau}}{A(t)\ \text{true}}\forall\ \text{E} \quad \Longrightarrow \quad \begin{array}{c}\mathcal{E}\\ [t/a]\mathcal{D}\\ A(t)\ \text{true}\end{array}$$

Sometimes we omit $\mathcal{E}$ from our proof tree and implication (as it is obvious there's a derivation for $t : \tau$).

Substitution Lemma (2nd part): If $\Gamma, a : \tau, \Gamma' \vdash A(a)$ true and $\Gamma \vdash t : \tau$ then $\Gamma, [t/a]\Gamma' \vdash A(t)$ true

$(\lambda x : \tau.M)t \overset{\beta}{\Longrightarrow} [t/x]M$

Local Completeness

$$\begin{array}{c}\mathcal{D}\\ \forall x : \tau.A(x)\ \text{true}\end{array} \quad \Longrightarrow \quad \dfrac{\begin{array}{c}\mathcal{D}\\ \forall x : \tau.A(x)\ \text{true}\end{array} \qquad \overline{a : \tau}}{\dfrac{A(a)\ \text{true}}{\forall x : \tau.A(x)\ \text{true}}\forall\ \text{I}^a}\forall\ \text{E}$$

Can also expand local completeness:

$$\begin{array}{c}\mathcal{D}\\ \forall x : \tau.A(x)\ \text{true}\end{array} \quad \Longrightarrow \quad \dfrac{\dfrac{\begin{array}{c}\mathcal{D}\\ M : \forall x : \tau.A(x)\ \text{true}\end{array} \qquad \overline{a : \tau}}{M\ a : A(a)\ \text{true}}\forall\ \text{E}}{\lambda a : \tau.M\ a : \forall x : \tau.A(x)\ \text{true}}\forall\ \text{I}^a$$

$$M : \forall x : \tau.A(x) \overset{\eta}{\Longrightarrow} \lambda x : \tau.M\ x$$

Expansion

$$M : A \supset B \Longrightarrow \lambda x : A.M\ x$$

Existential quantifier:

Well-formedness: A wf

$$\dfrac{\begin{array}{c}\overline{x : \tau} \\ \vdots \\ A(x) \text{ wf}\end{array}}{\exists x : \tau . A(x) \text{ wf}}$$

Truth: A true

$$\dfrac{A(t) \text{ true} \qquad t : \tau}{\exists x : \tau . A(x) \text{ true}} \exists \text{ I}$$

$$\dfrac{\exists x : \tau . A(x) \text{ true} \qquad \begin{array}{c}\overline{\overline{A(a) \text{ true}}^{\,\text{u}} \quad \overline{a : \tau}} \\ \vdots \\ C \text{ true}\end{array}}{C \text{ true}} \exists \text{ E}^{a,u}$$

Note that the a in $A(a)$ is the same as in $a : \tau$. This is similar to disjunction.

$$\dfrac{A \vee B \qquad \begin{array}{c}\overline{A \text{ true}} \\ \vdots \\ C \text{ true}\end{array} \qquad \begin{array}{c}\overline{B \text{ true}} \\ \vdots \\ C \text{ true}\end{array}}{C \text{ true}}$$

$$\dfrac{\forall x \in \{r, b, g\} . A(x) \qquad \begin{array}{c}\overline{A(r) \text{ true}} \\ \vdots \\ C \text{ true}\end{array} \qquad \begin{array}{c}\overline{A(b) \text{ true}} \\ \vdots \\ C \text{ true}\end{array} \qquad \begin{array}{c}\overline{A(c) \text{ true}} \\ \vdots \\ C \text{ true}\end{array}}{C \text{ true}}$$

Another FOL Example

$$\dfrac{\dfrac{\dfrac{\overline{A(a) \text{ true}}^{\,\text{u}}}{A(a) \supset A(a) \text{ true}} \supset \text{I}^u \quad \overline{a : \tau}}{\exists y : \tau A(a) \supset A(y) \text{ true}}}{\forall x : \tau \exists y : \tau . A(x) \supset A(y) \text{ true}} \forall \text{ I}^a$$

The following cannot be proved in constructive logic (in classical logic it is not a problem):

$$\dfrac{\overline{\forall x : \tau . A(x) \text{ true}}^{\,\text{u}}}{\exists x : \tau . P(x) \text{ true}}$$

$$\dfrac{\exists x : \tau . P(x) \text{ true}}{(\forall x : \tau . P(x) \supset (\exists x : \tau P(x))) \text{ true}} \supset \text{I}^u$$

If we add an additional assumption (prove the statement with τ not being empty), we can prove it:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\overline{\forall x : \tau . A(x) \text{ true}}^{\,\text{u}} \quad \overline{a : \tau}}{P(a) \text{ true}} \quad \overline{a : \tau}}{\exists x : \tau . P(x) \text{ true}}}{(\forall x : \tau . P(x) \supset (\exists x : \tau P(x))) \text{ true}} \supset \text{I}^{u}}{\forall y : \tau((\forall x : \tau . P(x) \supset (\exists x : \tau P(x)))) \text{ true}} \forall \text{ I}^{a}$$

8 01/31/19

8.1 More on First-order Logic

Reminder:
Prop $A, B := \ldots \mid \forall x : \tau . A(x) \mid \exists x : \tau . A(x)$
Terms $t, s := x \mid f(t_1, \ldots, t_n)$

$$\dfrac{\begin{array}{c}\overline{a : \tau} \\ \vdots \\ A(a) \text{ true}\end{array}}{\forall x : \tau . A(x) \text{ true}} \forall \text{I}^{a} \qquad \dfrac{\forall x : \tau . A(x) \text{ true} \quad t : \tau}{A(t) \text{ true}} \forall \text{ E}$$

$$\overline{\text{nat type}} \qquad \dfrac{}{Z : \text{nat}}\,\text{z\_n} \qquad \dfrac{n : \text{nat}}{s\ n : \text{nat}}\,\text{s\_n}$$

$$\dfrac{A(t) \text{ true} \quad t : \tau}{\exists x : \tau . A(x) \text{ true}} \exists \text{ I} \qquad \dfrac{\exists x : \tau . A(x) \text{ true} \quad \begin{array}{c}\overline{A(a) \text{ true}}^{\,\text{w}} \quad \overline{a : \tau} \\ \vdots \\ C \text{ true}\end{array}}{C \text{ true}} \exists \text{ E}^{w,a}$$

What can we do with the existential quantifier?

$A(t) \hat{=} [t/x]A(x)$

$even\ z \supset odd(s\ z)$

$\exists x : nat . even\ x \supset odd(s\ x)$

Examples

$$\dfrac{\dfrac{}{\exists x:\tau.\neg A(x)\text{ true}}\,\mathrm{u}\quad\dfrac{\dfrac{\dfrac{}{\neg A(a)}\,\mathrm{v}\quad\dfrac{\dfrac{\dfrac{}{\forall x:\tau.A(x)\text{ true}}\,\mathrm{w}\quad\dfrac{}{a:\tau}}{A(a)\text{ true}}\supset\mathrm{E}}{\bot}}{\exists\,\mathrm{E}^{v,a}}}{\dfrac{\bot}{\neg\forall x:\tau.A(x)\text{ true}}\supset\mathrm{I}^{w}}}{(\exists x:\tau.\neg A(x))\supset\neg(\forall x:\tau.A(x))\text{ true}}\supset\mathrm{I}^{u}$$

$$\dfrac{\dfrac{}{\exists x:nat.A(s\ x)\text{ true}}\,\mathrm{u}\quad\dfrac{\dfrac{}{A(s\ a)}\,\mathrm{w}\quad\dfrac{\overline{a:nat}}{s\ a:nat}\,s\_n}{\exists y:nat.A(y)\text{ true}}\exists\,\mathrm{I}}{\dfrac{\exists y:nat.A(y)\text{ true}}{\exists x:nat.A(s\ x)\supset\exists y:nat.A(y)\text{ true}}\supset\mathrm{I}^{u}}\exists\,\mathrm{E}^{w,a}$$

We now show local soundness and completeness for $\exists$:

Local Soundness $\Gamma, u:A(a)\text{ true}, a:\tau\vdash C\text{ true}$

$$\dfrac{\begin{array}{cc}\mathcal{E}_1 & \mathcal{E}_2\\ A(t)\text{ true} & t:\tau\end{array}}{\dfrac{\exists x:\tau.A(x)\text{ true}}{C\text{ true}}\exists\,\mathrm{I}}\exists\,\mathrm{E}^{u,a}\quad\begin{array}{cc}\dfrac{}{A(a)\text{ true}}\,\mathrm{u} & \dfrac{}{a:\tau}\\ \mathcal{D}^{u,a} & \\ C\text{ true} & \end{array}\quad\Longrightarrow\quad\begin{array}{c}[\mathcal{E}_1/u]([t/a]\mathcal{D})\\ C\text{ true}\end{array}$$

(By substitution lemmas, we use both here)

1: If $\Gamma, a:\tau, \Gamma'\vdash A\text{ true}$ and $\Gamma\vdash t:\tau$ then $\Gamma, [t/a]\Gamma'\vdash [t/a]A\text{ true}$

2: If $\Gamma, u:B\text{ true}$, $\Gamma'\vdash A\text{ true}$ and $\Gamma\vdash B\text{ true}$ then $\Gamma,\Gamma'\vdash A\text{ true}$

Local Completeness

$$\begin{array}{c}\mathcal{D}\\ \exists x:\tau.A(x)\text{ true}\end{array}\quad\Longrightarrow\quad\dfrac{\begin{array}{cc}\mathcal{D} & \dfrac{\dfrac{}{A(a)}\,\mathrm{u}\quad\dfrac{}{a:\tau}}{\exists x:\tau.A(x)}\exists\,\mathrm{I}\\ \exists x:\tau.A(x)\text{ true} & \end{array}}{\exists x:\tau.A(x)\text{ true}}\exists\,\mathrm{E}^{u,a}$$

Note that completeness isn't always an elimination and then an introduction, however, soundness must always introduce and then eliminate the rule.

Existential Rules with Terms

$$\dfrac{\begin{array}{cc}M:A(t)\text{ true} & t:\tau\end{array}}{\langle M,t\rangle:\exists x:\tau.A(x)\text{ true}}\exists\,\mathrm{I}\qquad\dfrac{\begin{array}{cc}M:\exists x:\tau.A(x)\text{ true} & \begin{array}{c}\dfrac{}{w:A(a)\text{ true}}\,\mathrm{w}\quad\dfrac{}{a:\tau}\\ \vdots\\ N:C\text{ true}\end{array}\end{array}}{let\ \langle w,a\rangle = M\ in\ N:C\text{ true}}\exists\,\mathrm{E}^{w,a}$$

Soundness:

$$\dfrac{\dfrac{\begin{array}{cc}\mathcal{E}_1 & \mathcal{E}_1 \\ M : A(t)\ \text{true} & t : \tau\end{array}}{\langle M, t\rangle : \exists x : \tau.A(x)\ \text{true}}\exists\ \text{I} \qquad \begin{array}{c}\overline{A(a)\ \text{true}}^{\,\text{u}} \\ \mathcal{D}^{u,a} \\ N : C\ \text{true}\end{array}}{let\ \langle u, a\rangle = \langle M, t\rangle\ in\ N : C\ \text{true}}\exists\ \text{E}^{u,a} \quad \overline{a : \tau}^{\,\text{e}} \quad \Longrightarrow \quad \begin{array}{c}[\mathcal{E}_1/u]([t/a]\mathcal{D}) \\ C\ \text{true}\end{array}$$

$$let\langle u, a\rangle = \langle M, t\rangle\ in\ N \implies \begin{array}{l}[M/u, t/a]N \\ [M/u][t/a]N\end{array}$$

Choice:

1. Build $=, \leq, \ldots$ on nats into the logic
2. Define formulas that describe $=, \leq, even, odd, \ldots$

   Could add the following rules:

$$n < s\ n \qquad \dfrac{a < b \qquad b < c}{a < c}$$

$$\dfrac{n < m}{n < s\ m} \qquad \dfrac{n < m}{s\ n < s\ m}\ \text{I}_s \qquad \dfrac{}{0 < s\ n}\ \text{I}_0 \qquad \dfrac{m < 0\ \text{true}}{C\ \text{true}}$$

   Some of these rules are redundant, we only need the last four rules.

   Are the first two rules and the last four rules equivalent? If we have both sets, is our language sound and complete?

   There are many choices we make here, but if we look at I_s and E_s, we can see that we are guided by the possibility of introducing and eliminating.

   What if we build on instead of defining formulas?

$\forall n : nat.\ 0 < s\ n$

Could also call this $less(0, n)$, but we wrote it in infix notation

$\forall n : nat.\ n < m \supset s\ n < s\ m$

$\forall n : nat.\ s\ n < s\ m \supset n < m$

$\forall n : nat.\ m < 0 \supset \bot$

We can prove $\forall x.s\ x < s\ 0 \supset P(x)$ in both ways that we defined $<$. If we define new formulas, it makes our proofs more compact. However, adding new rules makes us unsure about whether our system is consistent, therefore requiring us to prove consistency again.

Consistency: $\nvdash \bot$ true, i.e. there is no way of proving $\bot$ in our system.

Rules for governing equality for nats

$$\frac{}{0 = 0} \quad \frac{m = n}{s\ m = s\ n} \quad \frac{s\ m = s\ n}{m = n} \quad \frac{0 = s\ m}{C \text{ true}} \quad \frac{s\ m = 0}{C \text{ true}}$$

Third choice could be writing a program instead.

9 02/05/19

So Far: First-order Logic (independent of a given domain) + domain (for example natural numbers)

$$\text{nat type} \quad \frac{}{z : \text{nat}}\ \text{nat I}_z \quad \frac{n : nat}{suc\ n : nat}\ \text{nat I}_s$$

What are good elimination rules for nats? How to reason inductively about nats?

9.1 Relation/Predicates

$n \leq m$

$$\frac{}{z \leq \text{n}}\ \text{le}_z \quad \frac{n \leq m}{suc\ n \leq suc\ m}\ \text{le}_s$$

Can encode these rules as:
le_z: $\forall n : nat.z \leq n$
le_s: $\forall n : nat.\forall m.nat.n \leq m \supset suc\ n \leq suc\ m$
$ref : \forall x : nat.x = x$
These rules are called the signature $\mathcal{Y}$

What do I want to prove about our domain?

1. $\forall x : nat.x \leq x$
2. $\forall x : nat.\neg(x = z) \supset \exists y : nat.suc\ y = x$
3. $\forall x : nat.\forall y : nat.\forall w : nat, x \leq y \wedge y \leq w \supset x \leq w$

Can prove these via induction.

Elimination Rule Induction Rule

$$\frac{t : nat \qquad A(z)\ \text{true} \qquad \begin{array}{c}\overline{n : nat}\\ \ddots\\ A(suc\ n)\ \text{true}\end{array}\quad \begin{array}{c}\overline{A(n)\ \text{true}}\ \text{ih}\\ \vdots\end{array}}{A(t)\ \text{true}}\ nat\ \text{E}^{n,ih}$$

Generalization built-in: $A(42)$ true.

Different way of writing the previous rule:

$$\frac{\Gamma \vdash t : nat \qquad \Gamma \vdash A(z)\ \text{true} \qquad \Gamma, n : nat, ih : A(n)\ \text{true} \vdash A(suc\ n)\ \text{true}}{\Gamma \vdash A(t)\ \text{true}}\ nat\ \text{E}^{n,ih}$$

Example

$$\frac{\overline{\mathcal{Y}, a : nat \vdash a : nat} \qquad \begin{array}{c}\mathcal{D}_1\\ \mathcal{Y}, a : nat \vdash z \leq z\ \text{true}\end{array} \qquad \begin{array}{c}\mathcal{D}_2\\ \mathcal{Y}, a : nat, n : nat,\\ ih : n \leq n \vdash suc\ n \leq suc\ n\end{array}}{\begin{array}{c}\mathcal{Y}, a : nat \vdash a \leq a\ \text{true}\\ \hline \mathcal{Y} \vdash \forall x : nat.x \leq x\end{array}\ \forall\ \text{I}^a}\ nat\ \text{E}^{n,ih}$$

$$\mathcal{D}_1 = \frac{\overline{\mathcal{Y}, n : nat \vdash z \leq n}\ \text{le}_z \qquad \overline{\mathcal{Y} \vdash z : nat}\ \text{nat I}_z}{\mathcal{Y} \vdash z \leq z}$$

Let $\mathcal{Y}' = \mathcal{Y}, a : nat, n : nat, ih : n \leq n$

$\mathcal{D}_2 =$

$$\frac{\overline{\mathcal{Y} \vdash n \leq n}\ \text{ih} \qquad \begin{array}{c}\overline{\mathcal{Y}' \vdash \forall n : nat.\forall m : nat.n \leq m \supset suc\ n \leq suc\ m}\ \text{le}_s \qquad \overline{\mathcal{Y}' \vdash n : nat}\\ \hline \mathcal{Y}' \vdash n \leq n \supset suc\ n \leq suc\ n\end{array}\ \supset\ \text{E}}{\mathcal{Y}' \vdash suc\ n \leq suc\ n}\ \forall E(2\text{x})$$

Elimination Rule with Proof Terms

$$\frac{t : nat \qquad M_z : A(z)\ \text{true} \qquad \begin{array}{c}\overline{n : nat}\\ \ddots\\ M_s : A(suc\ n)\ \text{true}\end{array}\quad \begin{array}{c}\overline{f\ n : A(n)\ \text{true}}\ \text{ih}\\ \vdots\end{array}}{rec(t, M_z, n, ih, M_s) : A(t)\ \text{true}}\ nat\ \text{E}^{n,ih}$$

Maybe it would be nicer to write $rec(t, M_z, n, ih, M_s)$ as:
$rec\ t$ with $f\ z \to M_z \mid f(suc\ n) \to M_s$

Writing our earlier example as a program:
$\lambda a : nat.rec\ a$ with $f\ z \to le_z\ z \mid f\ suc\ n \to (le_s\ n\ n)(f\ n)$

Example

$$\dfrac{\dfrac{}{\mathcal{Y}, a : nat \vdash a : nat} \quad \overset{\mathcal{D}_1}{\mathcal{Y}, a : nat \vdash \neg(z = z) \supset \exists y : nat.\ suc\ y = z} \quad \overset{\mathcal{D}_2}{\mathcal{Y}, a : nat, n : nat, ih = \ldots \vdash \neg(suc\ n = z) \supset \exists y : nat.\ suc\ y = suc\ n}}{\dfrac{\mathcal{Y}, a : nat \vdash \neg(a = z) \supset \exists y : nat.\ suc\ y = a}{\mathcal{Y} \vdash \forall x : nat. \neg(x = z) \supset \exists y : nat.\ suc\ y = x}\ \forall\ \mathrm{I}^a}$$

$$\mathcal{D}_1 = \dfrac{\dfrac{\dfrac{\mathcal{Y}, u : \neg(z = z) \vdash \neg(z = z) \qquad \dfrac{\dfrac{}{\mathcal{Y} \vdash \forall x : nat. x = x}\ \text{refl}}{\mathcal{Y} \vdash z = z}\ \forall\ \mathrm{E}}{\bot}\ \supset \mathrm{E}}{\mathcal{Y}, u : \neg(z = z) \vdash \exists y : nat. suc\ y = z}\ \bot\ \mathrm{E}}{\mathcal{Y} \vdash \neg(z = z) \supset \exists y : nat. suc\ y = z}\ \supset \mathrm{I}^u$$

The above is like a proof by contradiction.
Let $\mathcal{Y}' = \mathcal{Y}, n : nat. ih : \ldots$

$$\mathcal{D}_2 = \dfrac{\dfrac{\dfrac{}{\mathcal{Y}' \vdash n : nat}}{\mathcal{Y}' \vdash suc\ n : nat} \quad \dfrac{\dfrac{}{\mathcal{Y}' \vdash \forall x : nat. x = x}\ \text{refl} \quad \dfrac{\dfrac{}{\mathcal{Y}' \vdash n : nat}}{\mathcal{Y}' \vdash suc\ n : nat}\ \forall\ \mathrm{E}}{\mathcal{Y}', u : \neg(suc\ n = z) : suc\ n = suc\ n}\ \exists\ \mathrm{I}}{\mathcal{Y}', u : \neg(suc\ n = z) \vdash \exists y : nat. suc\ y = suc\ n}$$
$$\mathcal{Y} \vdash \neg(suc\ n = z) \supset \exists y : nat. suc\ y = suc\ n$$

This is a weird proof since it never uses its IH and really is just proof on the cases. What does this program do?

$\lambda a : nat. rec\ a$ with $f\ z \to \lambda u : \neg(z = z). abort(u\ (refl\ z)) \mid f\ (suc\ n) \to \lambda u : \neg(suc\ n = z). \langle ref\ (suc\ n), n \rangle$

This is a provably correct predecessor function. If we erase "Proofs for equality" from the above program, we get:
$\lambda a : nat. rec\ a$ with $f\ z \to \_ \mid f(suc\ n) \to n$
This is what we call program extraction. Note in the z case the program is undefined (zero has no predecessor).

How do we come up with reduction rules for these kind of recursers?

Reduction Rule:

$$rec\ z \text{ with } \begin{array}{l} f\ z \to M_z \\ \mid f\ (suc\ n) \to M_s \end{array} \to M_z$$

$$rec\ (suc\ m) \text{ with } \begin{array}{l} f\ z \to M_z \\ \mid f\ (suc\ n) \to M_s \end{array} \to [m/n, f_m/f_n]M_s, \text{ where}$$

$f_m = rec\ m$ with $f\ z \to M_z \mid f\ (suc\ n) \to M_s$
f_n is a placeholder for the recursive call. Like in functional languages, we recurse here by "pasting" f_m when we look for the recursive call.

$$\dfrac{\Gamma \vdash t : nat \qquad \Gamma \vdash M_z : A(z) \qquad \Gamma, n : nat, f\ n : A(n) \vdash M_s : A(suc\ n)}{\Gamma \vdash rec\ t \text{ with } f\ z \to M_z \mid f\ (suc\ n) \to M_s : A(t)}$$

10 02/07/19

10.1 Normal Proof

Let's try and prove something basic directly:

$$\dfrac{\dfrac{\dfrac{}{A \text{ true}}\,\text{u}}{B \supset A \text{ true}}\supset \text{I}^v}{A \supset B \supset A \text{ true}}\supset \text{I}^u$$

(Here we omitted an unused assumption for B being true).

The proof term for this is: $\lambda u : A.\lambda v : B.u$, it's a program that will just ignore its second input.

The following is another way to prove the same conjecture:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{A \text{ true}}\,\text{u} \quad \dfrac{}{B \text{ true}}\,\text{v}}{A \wedge B \text{ true}}\wedge \text{I}}{A \text{ true}}\wedge \text{E}_l}{B \supset A \text{ true}}\supset \text{I}^v}{A \supset B \supset A \text{ true}}\supset \text{I}^u$$

The proof term for this is: $\lambda u : A.\lambda v : B.\underline{fst\langle u, v\rangle}$. However, with our reduction rules, we know that we can reduce $fst\langle u, v\rangle$ to u and end up with the same proof term as before.

What are normal proofs?

1. Can we characterize normal proof terms?
2. Can we describe normal derivations?

Normal (proof) terms $M := \lambda x : A.M \mid \langle M, N\rangle \mid inl^A\ M \mid inr^B\ M \mid R$

Neutral terms $R := fst\ R \mid snd\ R \mid x \mid R\ M$

The basic idea is that we want to rule out terms like $\lambda u : A.\lambda v : B.fst\langle u, v\rangle$.

A **normal term** cannot be reduced further.

$$\begin{array}{ll}
(\lambda x : A.M)\ N & \Longrightarrow\ [N/x]M \\
fst\langle M, N\rangle & \Longrightarrow\ M \\
snd\langle M, N\rangle & \Longrightarrow\ N \\
case\ (inl\ M)\ of\ inl\ x \to N_1 \mid inr\ y \to N_2 & \Longrightarrow\ [M/x]N_1
\end{array}$$

Examples

- $\lambda x : A \wedge B.fst\ x$ (we can have fst, but not on a pair directly, else it would reduce)
- $\lambda x : A \wedge (B \wedge C).snd\ (fst\ x)$

Normal Derivations

$$A \text{ true} \begin{cases} \Gamma^{\downarrow} \vdash M : A \uparrow & \text{There is a normal derivation} \\ \Gamma^{\downarrow} \vdash R : A \downarrow & \text{There is a neutral derivation} \end{cases}$$

Intuitively, $\uparrow$ means applying intro-rules bottom up (we have a proof for this), whereas $\downarrow$ means synthesizing information top down (from the assumptions).

$\Gamma^{\downarrow} = x_1 : A_1 \downarrow, \ldots, x_n : A_n \downarrow$

Note, the superscript for both Gammas is $\downarrow$ because when you have an assumption all you can do is synthesize information from it.

Rules

$$\frac{x : A \downarrow}{\Gamma^{\downarrow} \vdash x : A \downarrow}$$

Generally, introduction rules are $\uparrow$.

$$\frac{\Gamma^{\downarrow}, x : A \downarrow \vdash M : B \uparrow}{\Gamma^{\downarrow} \vdash \lambda x.M : A \supset B \uparrow} \supset \mathrm{I}^x$$

$$\frac{\Gamma^{\downarrow} \vdash M : A \uparrow \qquad \Gamma^{\downarrow} \vdash N : B \uparrow}{\Gamma^{\downarrow} \vdash \langle M, n \rangle : A \wedge B \uparrow} \wedge \uparrow$$

When do we extract information? With the elimination rules.

$$\frac{\Gamma^{\downarrow} \vdash R : A \wedge B \downarrow}{\Gamma^{\downarrow} \vdash fst\ R : A \downarrow}$$

$$\frac{\Gamma^{\downarrow} \vdash R : A \wedge B \downarrow}{\Gamma^{\downarrow} \vdash snd\ R : B \downarrow}$$

$$\frac{\Gamma^{\downarrow} \vdash R : A \supset B \downarrow \qquad \Gamma^{\downarrow} \vdash M : A \uparrow}{\Gamma^{\downarrow} \vdash R\ M : B \downarrow}$$

$$\frac{\Gamma^{\downarrow} \vdash M : A \uparrow}{\Gamma^{\downarrow} \vdash inl\ M : A \vee B \uparrow}$$

$$\frac{\Gamma^{\downarrow} \vdash R : A \vee B \downarrow \qquad \Gamma^{\downarrow}, x : A \downarrow \vdash N_1 : C \uparrow \qquad \Gamma^{\downarrow}, x : B \downarrow \vdash N_z : C \uparrow}{\Gamma^{\downarrow} \vdash \mathit{case}\ R\ \mathit{of}\ \mathit{inl}\ x \to N_1 \mid \mathit{inr}\ x \to N_2 : C \uparrow}$$

Here all the Cs can be $\downarrow$ or $\uparrow$, but for us it will be much simpler to have it as $\uparrow$.

Sometimes we may need to go in the other direction. Here's the switch from going $\downarrow$ to $\uparrow$.

$$\frac{\Gamma^{\downarrow} \vdash R : A \downarrow}{\Gamma^{\downarrow} \vdash R : A \uparrow} \uparrow\downarrow$$

Example

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{x : A \wedge (B \wedge C) \downarrow \vdash x : A \wedge (B \wedge C) \downarrow}}{x : A \wedge (B \wedge C) \downarrow \vdash \mathit{snd}\ x : B \wedge C \downarrow}}{x : A \wedge (B \wedge C) \downarrow \vdash B \downarrow}}{x : A \wedge (B \wedge C) \downarrow \vdash B \uparrow}}{\lambda x.\mathit{fst}(\mathit{snd}\ x) : (A \wedge (B \wedge C)) \supset B \uparrow}$$

For the examples we originally saw at the beginning of this lecture, if we try to annotate them:

$$\dfrac{\dfrac{\dfrac{\dfrac{\overline{A \downarrow}^{\,\mathrm{u}}}{A \uparrow}}{B \supset A \uparrow}\supset \mathrm{I}^{v}}{A \supset B \supset A \uparrow}\supset \mathrm{I}^{u}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\overline{A \downarrow}}{A \uparrow} \quad \dfrac{\overline{B \downarrow}}{B \uparrow}}{A \wedge B \uparrow}\wedge \mathrm{I}}{A \uparrow}\wedge \mathrm{E}_l \times \lightning}{B \supset A \uparrow}\supset \mathrm{I}^{v}}{A \supset B \supset A \uparrow}\supset \mathrm{I}^{u}$$

This proof isn't normal and we wouldn't be able to derive it as we have no rule to go from $A \wedge B \uparrow$ to $A \uparrow$, only $A \wedge B \downarrow$ to $A \downarrow$.

$$\dfrac{\dfrac{\dfrac{}{x : A \supset B \downarrow \vdash x : A \supset B \downarrow}}{x : A \supset B \downarrow \vdash x : A \supset B \uparrow}}{\lambda x : A \supset B.X : (A \supset B) \supset (A \supset B) \uparrow}$$

The proof terms aren't necessarily unique. Consider: $\lambda x : A \supset B.\lambda y : A.x\ y$

Terms are β-normal form (none of the reductions apply).

Soundness If $\Gamma^{\downarrow} \vdash M : A \uparrow$ then $\Gamma \vdash M : A$.
If $\Gamma^{\downarrow} \vdash R : A \downarrow$ then $\Gamma \vdash R : A$.

This is a way for us to relate one proof system to another.

Proof by mutual structural induction on the first derivation

$$\textbf{Case:} \quad \mathcal{D} = \frac{\mathcal{D}_0 \qquad \mathcal{D}_1 \qquad \mathcal{D}_2}{\Gamma^{\downarrow} \vdash R : A \vee B \downarrow \qquad \Gamma^{\downarrow}, x : A \downarrow \vdash N_1 : C \uparrow \qquad \Gamma^{\downarrow}, x : B \downarrow \vdash N_z : C \uparrow}{\Gamma^{\downarrow} \vdash \mathit{case}\ R\ \mathit{of}\ \mathit{inl}\ x \to N_1 \mid \mathit{inr}\ x \to N_2 : C \uparrow}$$

$\Gamma \vdash R : A \vee B$ by IH(2) using $\mathcal{D}_0$
$\Gamma, x : A \vdash N_1 : C$ by IH(1) using $\mathcal{D}_1$
$\Gamma, y : B \vdash N_2 : C$ by IH(1) using $\mathcal{D}_2$
$\Gamma \vdash \mathit{case}\ R\ \mathit{of}\ \mathit{inl}\ x \to N_1 \mid \mathit{inr}\ y \to N_2 : C$ by $\vee E^{x,y}$

This is a good example because we had to use both inductive hypotheses. We can think of this as writing mutually recursive functions, one that deals with normal derivations, one that deals with neutral derivations.

$$\textbf{Case:} \quad \mathcal{D} = \frac{\mathcal{D}'}{\Gamma^{\downarrow} \vdash R : A \downarrow}{\Gamma^{\downarrow} \vdash R : A \uparrow}$$

$\Gamma \vdash R : A$ by IH(2) using $\mathcal{D}'$

Completeness Our new system isn't complete (because we designed it that way, we cannot represent everything). To make this complete, we'd have to be able to go in both directions:

$$\frac{\Gamma^{\downarrow} \vdash M : A \uparrow}{\Gamma^{\downarrow} \vdash (M : A) : A \downarrow}$$

The "offending" rule which makes proofs non-normal, however it allows us to prove completeness.

Completeness: If $\Gamma \vdash M : A$ then $\Gamma^{\downarrow} \vdash M : A \uparrow$ and $\Gamma^{\downarrow} \vdash M : A \downarrow$

Proof by induction on $\Gamma \vdash M : A$

$$\textbf{Case:} \quad \frac{\Gamma \vdash A \supset B \qquad \Gamma \vdash A}{\Gamma \vdash B} \supset \mathrm{E}$$

$\Gamma \vdash A \supset B \downarrow$ by IH($\mathcal{D}_1$)
$\Gamma \vdash A \uparrow$ by IH($\mathcal{D}_2$)
$\Gamma \vdash B \downarrow$ by rule $\supset$ E$\downarrow$
$\Gamma \vdash B \uparrow$ $\uparrow\downarrow$

Why introduce a new "offending" rule to show completeness for a new system? Say we want to prove consistency of $\Gamma \vdash A$ true.

We have $\Gamma \vdash A$ true in one system and
$\Gamma \vdash A\uparrow, \Gamma \vdash A\downarrow$, $\dfrac{\Gamma \vdash A\uparrow}{\Gamma \vdash A\downarrow}$. But we cannot tell if $\bot$ can be proven from here.

We will see that using the rules $\Gamma \vdash A\uparrow, \Gamma \vdash A\downarrow$, we will develop Sequent Calculus (directed, easy to see that $\bot$ cannot be proven unless you have assumptions). We will then introduce the Cut-Rule, which can be proven admissible, not necessary, like the substitution rule. The Cut-Rule corresponds to the $\dfrac{\Gamma \vdash A\uparrow}{\Gamma \vdash A\downarrow}$ and Sequent Calculus with the Cut-Rule will be complete, however because the Cut-Rule is admissible/not needed, we can remove it.

Note: we say the proof term M for completeness is conserved, but that is not exactly true, we may add additional annotations and so we say:

Completeness: If $\Gamma \vdash (M)^- : A$ then $\Gamma^{\downarrow} \vdash M : A\uparrow$ and $\Gamma^{\downarrow} \vdash M : A\downarrow$

11 02/12/19

Going to go over how we show two proof systems are equivalent. Last class we saw:

Natural Deduction	**Normal Natural Deduction**
$\Gamma \vdash M : A$	$\Gamma^{\downarrow} \vdash m : A\uparrow$
	$\Gamma^{\downarrow} \vdash r : A\downarrow$
Term $M := x \mid \lambda x : A.M \mid$	Normal terms $m := \lambda x.m \mid \langle m_1, m_2\rangle \mid r$
$M_1\ M_2 \mid \langle M_1, M_2\rangle \mid fst\ M \mid snd\ M$	Neutral terms $r := x \mid fst\ r \mid snd\ r \mid r\ m \mid \underline{(m : A)}$

Note that showing normal natural $\to$ natural was easy, however to go the other way we had to introduce a new rule $\underline{(m : A)}$, which was:

$$\frac{\Gamma \vdash m : A\uparrow}{\Gamma \vdash (m : A) : A\downarrow}$$

Recall HW1 $m := x \mid \lambda x.m \mid m_1\ m_2 \mid \langle m_1, m_2\rangle \mid let\ \langle x, y\rangle = m_1\ in\ m_2$
$\Gamma \vdash m : A$

$$\frac{\Gamma \vdash m_1 : A \wedge B \qquad \Gamma, x : A, y : B \vdash m_2 : C}{\Gamma \vdash let\ \langle x, y\rangle = m_1\ in\ m_2 : C}$$

If $\Gamma \vdash \underline{m} : A$ then $\Gamma \vdash \underline{M} : A$. How to translate between languages? (We want to relate m and M, this is very much like compiling one language to another)

$\to$ Function that translates m to M: $(m)^-$

$$\begin{aligned}
(x)^- &= x \\
(\langle m_1, m_2\rangle)^- &= \langle (m_1)^-, (m_2)^-\rangle \text{ (recursive)} \\
(let\ \langle x, y\rangle)^- = m_1\ in\ m_2 &= [fst\ (m_1)^-/x, snd\ (m_1)^-/y](m_2)^-
\end{aligned}$$

(Note: another option for let: $let\ \langle x, y\rangle = m_1\ in\ m_2 = (\lambda x.\lambda y.(m_2)^-\ (fst\ (m_1)^-))(snd\ (m_1)^-)$, however this will be much harder to prove equivalence for)

Case: $\mathcal{D} = \dfrac{\begin{array}{cc}\mathcal{D}_1 & \mathcal{D}_2 \\ \Gamma \vdash m_1 : A \wedge B & \Gamma, x : A, y : B \vdash m_2 : C\end{array}}{\Gamma \vdash let\ \langle x, y\rangle = m_1\ in\ m_2 : C}$

$\Gamma \vdash (m_1)^- : A \wedge B$	by IH on $\mathcal{D}_1$
$\Gamma \vdash fst\ (m_1)^- : A$, $\Gamma \vdash snd\ (m_1)^- : B$	by IH on $\mathcal{D}_2$
$\Gamma, x : A, y : B \vdash (m_2)^- : C$	by IH on $\mathcal{D}_2$
$\Gamma \vdash [fst\ (m_1)^-/x, snd\ (m_1)^-/y](m_2)^- : C$	by substitution lemma
$\Gamma \vdash (let\ \langle x, y\rangle)^- = m_1\ in\ m_2 : C$	by translation function

We could also prove $let\ \langle x, y\rangle = m_1\ in\ m_2 = (\lambda x.\lambda y.(m_2)^-\ (fst\ (m_1)^-))(snd\ (m_1)^-)$ by solely using implication elimination and introduction.

We will be able to show that the normal natural deduction system cannot prove bottom because sequent calculus cannot prove bottom, we didn't need the extra rule because the Cut Rule is admissible.

If $\Gamma \vdash M : A$ then there exists a normal proof $\Gamma \vdash m : A \uparrow$

Normal ND

$$\frac{\Downarrow \Gamma^{\downarrow} \vdash r : A \downarrow \text{ Elim-Rules}}{\Uparrow \Gamma^{\downarrow} \vdash m : A \uparrow \text{ Intro-Rules}}$$

Sequent Calculus

$$\text{initial rule}\ \frac{}{\underbrace{\Uparrow}_{\text{Left rules}}\ \Gamma \implies \underbrace{A}_{\text{Right rules}}\ \Uparrow}$$

Rules Init: $\dfrac{u : A \in \Gamma}{\Gamma \implies A}$

Conjunction: $\dfrac{\Gamma \implies A \quad \Gamma \implies B}{\Gamma \implies A \wedge B}$

$\dfrac{\Gamma, u : A \wedge B, w : A \implies C}{\Gamma, u : A \wedge B \implies C}$ Here we see that our set of assumption grows, we extract A from $A \wedge B$, like a $\downarrow$.

$\dfrac{\Gamma, u : A \wedge B, w : B \implies C}{\Gamma, u : A \wedge B \implies C}$ Extract B.

Implication: $\dfrac{\Gamma, u : A \implies B}{\Gamma \implies A \supset B}\ \dfrac{\Gamma, u : A \supset B, w : B \implies C \quad \Gamma, u : A \supset B \implies A}{\Gamma, u : A \supset B \implies C}$

Top: $\dfrac{}{\Gamma \implies T}\,\mathrm{T\ R}$

Falsehood: $\dfrac{}{\Gamma, u : \bot \implies C}$

Disjunctions: $\dfrac{\Gamma \implies A}{\Gamma \implies A \vee B}\quad \dfrac{\Gamma \implies B}{\Gamma \implies A \vee B}$

$$\frac{\Gamma, u : A \vee B, v : A \implies C \qquad \Gamma, u : A \vee B, w : B \implies C}{\Gamma, u : A \vee B \implies C}$$

Quantifiers: $\mathrm{R}\forall^a\,\dfrac{\Gamma \implies A(a)}{\Gamma \implies \forall x : \tau . A(x)}\quad \dfrac{\Gamma \vdash t : \tau \qquad \Gamma, u : \underline{\forall x : \tau . A(x)}, w : \underline{A(t)} \implies C}{\Gamma, u : \forall x : \tau . A(x) \implies C}\,\mathrm{Can}$ only instantiate witness once

$$\frac{\Gamma \vdash t : \tau \qquad \Gamma \implies A(t)}{\Gamma \implies \exists x : \tau . A(x)}\quad \frac{\Gamma, u : \exists x : \tau . A(x), w : A(a) \implies C}{\Gamma, u : \exists x : \tau . A(x) \implies C}\,\exists\ \mathrm{L}$$

Γ accumulates everything we extract from rules, things are kind of inverted as to how they were with natural deduction. Note that you can keep $a : \tau$ in your assumptions for $\forall$ I and $\exists$ E.

Example

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{w : A \in \Gamma'}{\Gamma' \implies A}\,\text{init} \quad \dfrac{}{\Gamma', v : B \implies B}\,\text{init}}{\Gamma' \implies B}\,\supset \mathrm{L} \quad \dfrac{}{\Gamma', v : C \implies C}\,\text{init}}{\overbrace{\Gamma, u_1 : (A \supset B), u_2 : B \supset C}^{\Gamma'} \implies C}\,\supset \mathrm{L}}{\overbrace{u : (A \supset B) \wedge (B \supset C), w : A}^{\Gamma} \implies C}\,\wedge_1 \mathrm{L},\ \wedge_2 \mathrm{L}}{u : (A \supset B) \wedge (B \supset C) \implies A \supset C}\,\supset \mathrm{R}}{\cdot \implies ((A \supset B) \wedge (B \supset C)) \supset A \supset C}\,\supset \mathrm{R}$$

Now, in natural deduction:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{(A \supset B) \wedge (B \supset C)\downarrow}\,\mathrm{u}}{A \supset B \downarrow}\,\wedge\ \mathrm{E}_l \quad \dfrac{}{A \downarrow\uparrow}\,\mathrm{w}}{B \downarrow\uparrow}\,\supset \mathrm{E} \quad \dfrac{\dfrac{}{(A \supset B) \wedge (B \supset C)\downarrow}\,\mathrm{u}}{B \supset C \downarrow}\,\wedge\ \mathrm{E}_r}{C \downarrow\uparrow}\,\supset \mathrm{E}}{A \supset C \uparrow}\,\supset \mathrm{I}^w}{(A \supset B) \wedge (B \supset C) \supset A \supset C \uparrow}\,\supset \mathrm{I}^u$$

So we see that init does not correspond to the assumptions, it corresponds to the $\uparrow\downarrow$.

Is there a proof for $\cdot \implies \bot$? No. (Obviously, we don't have a R rule for it and we have no assumptions. This system is complete and it is clearer to see that we don't have a proof for $\bot$ unlike natural deduction.)

In natural deduction, we could have hypothetically had some proof like:

$$\dfrac{A \vee B \qquad \begin{array}{c}\overline{\text{A}} \\ \vdots \\ \bot\end{array} \qquad \begin{array}{c}\overline{\text{B}} \\ \vdots \\ \bot\end{array}}{\bot}$$

Soundness If $\Gamma \implies A$ then $\Gamma^{\downarrow} \vdash A \uparrow$ Induction on $\Gamma \implies A$.

Case: $\mathcal{D} = \dfrac{u : A \in \Gamma}{\Gamma \implies A}\,\text{init}$

$u : A \downarrow \in \Gamma^{\downarrow}$

$\Gamma^{\downarrow} \vdash A \downarrow$ by assumption

$\Gamma^{\downarrow} \vdash A \uparrow$ **by coercion**

Right rules are not interesting as they have direct correspondence to $\Gamma^{\downarrow} \vdash A \uparrow$ as that's how we defined them. Let's look at a left rule.

Case: $\mathcal{D} = \dfrac{\Gamma, u : A \wedge B, v : A \implies C}{\Gamma, u : A \wedge B \implies C} \wedge \text{L}_1$

$\Gamma^{\downarrow}, u : A \wedge B \downarrow, v : A \downarrow \vdash C \uparrow$ by IH (1)

$\Gamma, u : A \wedge B \downarrow \vdash A \wedge B \downarrow$ by assumption

$\Gamma, u : A \wedge B \downarrow \vdash A \downarrow$ by $\wedge \text{E}_l$ (2)

$\Gamma^{\downarrow}, u : A \wedge B \downarrow \vdash C \uparrow$ by substitution lemma (1) + (2)

12 02/14/19

ND	Normal ND	Sequent Calculus
$\Gamma \vdash A$ true $\rightarrow$	$\Gamma \vdash A \uparrow$	$\overset{\text{Soundness}}{\leftarrow} \Gamma \implies A$
	$\Gamma \vdash A \downarrow$	Cut Rule
	$\dfrac{\Gamma \vdash A \uparrow}{\Gamma \vdash A \downarrow}\ \overset{\text{Complete}}{\rightarrow}$	$\dfrac{\Gamma \implies A \qquad \Gamma, A \implies C}{\Gamma \implies C}$

Today we want to prove:

1. Completeness of Sequent Calculus.
   If $\Gamma^{\uparrow} \vdash A \uparrow$ then $\Gamma \implies A$

2. Cut is **admissible**. If $\Gamma \implies A$ and $\Gamma, A \implies C$, then $\Gamma \implies C$

Remember

- Right rules in $\Gamma \implies A$ correspond directly to the intro rules in ND
- Left rules in $\Gamma \implies A$ extract more information from an assumption present in Γ.

$$\frac{\Gamma, A \wedge B, A \implies C}{\Gamma, A \wedge B \implies C}$$

Attempt

(A) If $\Gamma^{\downarrow} \vdash A \uparrow$ then $\Gamma \implies A$

(B) If $\Gamma^{\downarrow} \vdash A \downarrow$ then $\Gamma \implies A$

This is incorrect and will easily breakdown, i.e.

Consider Elim rule for $A \wedge B$
Assumption: $\dfrac{\Gamma^{\downarrow} \vdash A \wedge B \downarrow}{\Gamma^{\downarrow} \vdash A \downarrow}$
(IH) $\Gamma \implies A \wedge B$
To show. $\Gamma \implies A$? We need to show what we assumed? That doesn't make sense.
So what will we show instead?

(B) If $\Gamma^{\downarrow} \vdash A \downarrow$ and $\Gamma, A \implies C$ then $\Gamma \implies C$.

Case $\mathcal{D} = \dfrac{\Gamma^{\downarrow} \vdash A \wedge B \downarrow}{\Gamma^{\downarrow} \vdash A \downarrow}$

$\Gamma, A \implies C$	by assumption
$\Gamma, A \wedge B, A \implies C$	by weakening (can throw in extra assumptions)
$\Gamma, A \wedge B \implies C$	by $\wedge L$
$\Gamma \implies C$	by IH

Case $\mathcal{D} = \dfrac{\Gamma^{\downarrow} \vdash A \supset B \downarrow \qquad \Gamma^{\downarrow} \vdash A \uparrow}{\Gamma^{\downarrow} \vdash B \downarrow}$

$\Gamma, B \implies C$	by assumption
$\Gamma \implies A$	by IH (A)
$\Gamma, A \supset B \implies A$	by weakening
$\Gamma, A \supset B, B \implies C$	by weakening
$\Gamma, A \supset B \implies C$	by $\supset$ L
$\Gamma \implies C$	by IH

Case $\dfrac{\Gamma^{\downarrow} \vdash A \downarrow}{\Gamma^{\downarrow} \vdash A \uparrow}$

$\Gamma, A \implies A$ by init

$\Gamma \implies A$ by IH

Case $\dfrac{\Gamma^{\downarrow} \vdash A \uparrow}{\Gamma^{\downarrow} \vdash A \downarrow}$ (The rule that makes our proofs non-normal)

$\Gamma \implies A$ by IH

Assuming $\Gamma, A \implies C$, we have $\Gamma \implies C$. That is the cut rule!

$$\frac{\Gamma \implies A \qquad \Gamma, A \implies C}{\Gamma \implies C}$$

So the cut rule corresponds to non-normal proofs (as expected).

Cut is admissible If $\Gamma \overset{\mathcal{D}}{\implies} A$ and $\Gamma, \underline{A} \overset{\mathcal{E}}{\implies} C$ then $\Gamma \implies C$, where A is the principle formula/cut formula.

When can we appeal to the IH?

1. The cut formula is getting smaller (will in some way get to the base case)
2. Either $\mathcal{D}$ is smaller and $\mathcal{E}$ stays the same or $\mathcal{E}$ is smaller and $\mathcal{D}$ stays the same. i.e. one of them must be getting smaller.

Initial sequences (base cases):

Case : $\mathcal{D} = \dfrac{A \in \Gamma}{\Gamma \implies A}$ init

$\Gamma = (\Gamma', A)$ (we know we have A in Γ)

$\underbrace{\Gamma', A}_{\Gamma}, A \implies C$ by assumption ($\mathcal{E}$)

$\Gamma \implies C$ by contraction ($\mathcal{E}$), don't need to use the second A because we already have A

Case : $\mathcal{E} = \dfrac{}{\Gamma, A \implies \underbrace{A}_{C}}$ init

$\Gamma \implies A$ by $\mathcal{D}$

Case : $\mathcal{E} = \dfrac{C \in \Gamma}{\Gamma, A \implies C}$ init

$\Gamma \implies C$ by $C \in \Gamma$ (init) Non base cases:

Case : A is the principal formula and it is the cut form.

$$\mathcal{D} = \dfrac{\overset{\mathcal{D}_1}{\Gamma \implies A} \qquad \overset{\mathcal{D}_2}{\Gamma \implies B}}{\Gamma \implies A \wedge B}, \mathcal{E} = \dfrac{\overset{\mathcal{E}'}{\Gamma, A \wedge B, A \implies C}}{\Gamma, A \wedge B \implies C}$$

$\Gamma, A \implies A \wedge B$ by weakening $\mathcal{D}$

$\mathcal{F} : \Gamma, A \implies C$ by IH $(A \wedge B, \mathcal{D}, \mathcal{E}')$

$\Gamma \implies C$ by IH because $A < A \wedge B$ $(A, D_1, \mathcal{F})$

Case : $\mathcal{D} = \dfrac{\overset{\mathcal{D}'}{\Gamma, A \implies B}}{\Gamma \implies A \supset B}, \mathcal{E} = \dfrac{\overset{\mathcal{E}_1}{\Gamma, A \supset B \implies A} \qquad \overset{\mathcal{E}_2}{\Gamma, A \supset B, B \implies C}}{\Gamma, A \supset B \implies C}$

$\Gamma, B \implies A \supset B$ by weakening on $\mathcal{D}$

$\Gamma, B \implies C$ by IH $(A \supset B, \mathcal{D}, \mathcal{E}_2)$ $(\mathcal{E}_2 < \mathcal{E})$

$\mathcal{F} : \Gamma \implies A$ by IH $(A \supset B, \mathcal{D}, \mathcal{E}_1)$ $(\mathcal{E}_1 < \mathcal{E})$

$\Gamma \implies B$ by IH $(A, \mathcal{F}, \mathcal{D}')$ $(A < A \supset B)$

$\Gamma \implies C$ by IH $(B, \ldots)$ $(B < A \supset B)$

Note that the order of our cuts here is important, if we cut in the wrong order it may not be justified as we may not know if the formula is getting smaller.

Case : $\Gamma \overset{\mathcal{D}}{\implies} A$, $\mathcal{E} = \dfrac{\overset{\mathcal{E}_1}{\Gamma, A \implies C} \qquad \overset{\mathcal{E}_2}{\Gamma, A \implies D}}{\Gamma, A \implies C \wedge D} \wedge \mathrm{R}$

$\Gamma \implies C$ by IH $(A, \mathcal{D}, \mathcal{E}_1)$ $\mathcal{E}_1 < \mathcal{E}$

$\Gamma \implies D$ by IH $(A, \mathcal{D}, \mathcal{E}_2)$ $\mathcal{E}_2 < \mathcal{E}$

Case : $\mathcal{D} = \dfrac{\overset{\mathcal{D}_1}{\Gamma', B_1, \wedge B_2, B_1 \implies A}}{\underbrace{\Gamma', B_1 \wedge B_2}_{\Gamma} \implies A}$, $\underbrace{\Gamma'_1, B_1 \wedge B_2}_{\Gamma}, A \implies C$ ($\mathcal{E}$)

$\Gamma', B_1 \wedge B_2, B_1, A \implies C$ by weakening $\mathcal{E}$

$\Gamma', B_1 \wedge B_2, B_1 \implies C$ by IH$(A, \mathcal{D}_1, \mathcal{E})$

$\underbrace{\Gamma', B_1 \wedge B_2}_{\Gamma} \implies C$ by $\wedge$ L_1

13 02/19/19

13.1 Sequent Calculus

$\Gamma \implies A$

$$\dfrac{\Gamma \implies A \qquad \Gamma \implies B}{\Gamma \implies A \wedge B} \quad \dfrac{\Gamma, A \wedge B, A \implies C}{\Gamma, A \wedge B \implies C} \quad \dfrac{\Gamma, A \wedge B, B \implies C}{\Gamma, A \wedge B \implies C} \quad \dfrac{\Gamma, A \implies B}{\Gamma \implies A \supset B}$$

$$\frac{\Gamma, A \supset B \implies A \qquad \Gamma, A \supset B, B \implies C}{\Gamma, A \supset B \implies C} \quad \frac{A \in \Gamma}{\Gamma \implies A} \quad \frac{}{\Gamma, \bot \implies C} \quad \frac{}{\Gamma \implies \Gamma} \quad \frac{\Gamma \implies A}{\Gamma \implies A \vee B}$$

$$\frac{\Gamma \implies B}{\Gamma \implies A \vee B} \quad \frac{\Gamma, A \vee B, A \implies C \qquad \Gamma, A \vee B, B \implies C}{\Gamma, A \vee B \implies C} \quad \frac{\Gamma, a : \tau \implies A(a)}{\Gamma, \forall x : \tau . A(x)}$$

$$\frac{\Gamma, \forall x : \tau . A(x), A(t) \implies C}{\Gamma, \forall x : \tau . A(x) \implies C} \quad \frac{\Gamma \implies A(t)}{\Gamma \implies \exists x : \tau . A(x)}$$

Cut If $\Gamma \implies A$ and $\Gamma, A \implies C$ then $\Gamma \implies C$.

We showed that:

ND		Sequent Calc (without cut)
$\Gamma \vdash A$ true	$\iff$	$\Gamma \implies A$

What follows?

1. There is no derivation in ND for $\bot$
   - $\nvdash \bot$ true (since there is no proof $\cdot \implies \bot$)
2. If $\bot$ $A \vee B$ then $\vdash A$ true or $\vdash B$ true
3. If $\cdot \vdash \exists : \tau . A(x)$ true then $\vdash A(t)$ true for some t

Independence Results Is it true that $\vdash A \vee \neg A$ true?

If $\cdot \vdash A \vee \neg A$ true, then there must be some derivation $\cdot \implies A \vee \neg A$

By inversion (looking at the rules for $\vee$ R_1, $\vee$ R_2) either there is a proof for $\implies A$ or there is a proof for $\implies \neg A$, i.e. $A \implies \bot$

Idea

1. Apply right rules until we have to make a choice on the right
2. Apply all left rules until we have only propositions left in Γ that force a choice

We have:

- do-care non-determinism ($\vee$ R_1, $\vee$ R_2, $\supset$ L, $\forall$ L, $\exists$ R)
- don't-care non-determinism ($\wedge$ R, $\wedge$ L_1, $\wedge$ L_2, $\forall$ R, $\exists$ L)

  For the $\wedge$ assumption extraction rules, we don't care because we can extract the assumptions at any stage.

Inversion Properties

1. If $\Gamma \implies A \wedge B$ then $\Gamma \implies A$ and $\Gamma \implies B$ (basically what we mean by having a don't care non-determinism)

   Proof
   $\Gamma \implies A \wedge B$ by assumption
   $\Gamma, A \wedge B, A \implies A$ by init
   $\Gamma, A \wedge B \implies A$ by $\wedge\, L_1$
   $\Gamma \implies A$ by cut
   Similarly,
   $\Gamma, A \wedge B, B \implies B$ by init
   $\Gamma, A \wedge B \implies B$ by $\wedge\, L_2$
   $\Gamma \implies B$ by cut

   Cut allows you to construct very short proofs.

   Otherwise you'll have to go the long way using induction, like:

   Case $\dfrac{\Gamma \implies A \qquad \Gamma \implies B}{\Gamma \implies A \wedge B}$ Therefore we immediately have $\Gamma \implies A$ and $\Gamma \implies B$

   Case $\dfrac{\overset{\mathcal{D}'}{\Gamma, C \wedge D, C \implies A \wedge B}}{\Gamma, C \wedge D \implies A \wedge B}$

   $\Gamma, C \wedge D, C \implies A$ by IH
   $\Gamma, C \wedge D, C \implies B$ by IH
   $\Gamma, C \wedge D \implies A$ by $\wedge\, L_1$
   $\Gamma, C \wedge D \implies B$ by $\wedge\, L_1$

2. If $\Gamma \implies A \supset B$ then $\Gamma, A \implies B$

3. If $\Gamma, A \wedge B \implies C$ then $\Gamma, A \wedge B, A \implies C$

4. If $\Gamma, A \wedge B \implies C$ then $\Gamma, A \wedge B, B \implies C$

   3 and 4 give you: If $\Gamma, A \wedge B \implies C$ then $\Gamma, A, B \implies C$ (stronger inversion)

5. If $\Gamma, A \vee B \implies C$ then $\Gamma, A \implies C$ and $\Gamma, B \implies C$

13.2 Horn Clauses

$D := \overbrace{P}^{Atomic} \mid G \supset D \mid \forall x : \tau.D$

Horn Goals $G := P \mid G_1 \wedge G_2$ (although usually just atomic because if you have a conjunction you can just ask multiple questions)

Atomic Formula $P := P(t_1, \ldots, t_n)$

Prolog is like the original form of this.

For example, we can try to prove stuff about natural numbers:

$even\ z$ (Atomic formula)

$add\ (suc\ z)$

$\forall x : nat.even\ x \supset \underbrace{add\ (suc\ x)}_{Head}$

Prolog	Inf Rules
$add\ (suc\ x) :-$	$\frac{}{even\ z}$
$even\ x$	

$\frac{P_1 \quad \ldots \quad P_n}{C}$ Clause $P_1 \wedge \ldots \wedge P_n \supset C$

Graph Reachability (good example for what can be represented)

edge $a\ b$	$\forall x, \forall y.edge\ x\ y \supset reach\ x\ y$
edge $b\ c$	$\forall x \forall y \forall w.$
edge $c\ d$	$reach\ x\ y \wedge reach\ y\ w$
edge $c\ a$	$\supset reach\ x\ w$

This is our set of clauses. Left side are our axioms/facts, right side are clauses that let us get further with the axioms. This is widely used for things like databases.

Search Problem $\underbrace{\Gamma}_{\text{set of Horn Clauses}} \implies G$

Programs you would write would look like:

```
reach x y:-
    edge x y
reach x y:-
    reach x w,
    reach w y
```

Can I reach a d? It will try the rules we wrote above.

Uniform Proofs

1. $\Gamma \implies G$ (don't care non-determinism, can always eagerly split conjunctions)
2. $\Gamma > D \implies P$ (pick a clause from Γ, do care because we're making a choice)

14 02/21/19

14.1 Horn Clauses, Logic Programming

Very limited subset of natural deduction.

Uniform Proofs

Horn Clause $D := P \mid G \supset P \mid \forall x : \tau.D$

Horn Goal $G := P \mid G_1 \wedge G_2$

Atomic Form $P := p(1, \ldots, t_n)$

e_z : $even\ z$
o_sz : $odd(suc\ z)$
e_s : $\forall x : nat.odd\ x \supset even(suc\ x)$
o_s : $\forall x : nat.even\ x \supset odd(suc\ x)$
r_e : $\forall x : node.\forall y : node.edge\ x\ y \supset reach\ x\ y$
r_tr : $\forall x : node.\forall y : node.\forall w : node.reach\ x\ w \wedge reach\ w\ y \supset reach\ x\ y$

Plan Extend the Horn Fragment to Hereditary Harrop Formulas (Higher-order logic programming) $\hat{=} \supset, \forall, P$
Clauses $C := P \mid \forall x : \tau.C \mid C \supset C$

We'll be able see if proofs exist for certain things:

$\Delta \implies P$
$\Delta \implies reach\ a\ c$
$\Delta \implies odd\ (suc\ (suc\ z))$ (hopefully not true as 2 is even)
$\Delta \implies even\ (suc\ (suc\ z))$

Our rules are as follows, which are just the sequent calculus rules that we need.

$$\frac{P \in \Delta}{\Delta \implies P} \qquad \frac{\Delta \implies G_1 \qquad \Delta \implies G_2}{\Delta \implies G_1 \wedge G_2} \wedge \text{R} \qquad \frac{\Delta, \forall x : \tau.D, [t/x]D \implies G}{\Delta, \forall x :\ \tau.D \implies G} \forall \text{ L}$$

$$\frac{\Delta, G \supset P \implies G \qquad \Delta, G \supset P, P \implies G'}{\Delta, G \supset C \implies G'} \supset \text{L}$$

Recall that last class we proved:

Inversion: If $\Delta \implies G_1 \wedge G_2$ then $\Delta \implies G_1$ and $\Delta \implies G_2$.

So if the clause on the right is not atomic, we can split it up.

Idea: (we want to come up with a search strategy)

1. Apply init and $\wedge$ R until $\Delta \implies P$ (atomic formula)

   We'll have two steps we can take: $\Delta \overset{u}{\implies} G_1$, $\Delta \overset{e}{\implies} P$, where u is uniform and we work with the left rules.

$$\frac{\Delta \overset{u}{\implies} G_1 \qquad \Delta \overset{u}{\implies} G_2}{\Delta \overset{u}{\implies} G_1 \wedge G_2} \qquad \frac{\Delta \overset{e}{\implies} P}{\Delta \overset{u}{\implies} P}$$

Where:
$$\frac{P \in \Delta}{\Delta \overset{e}{\implies} P} \qquad \frac{\Delta, \forall x : \tau.\mathcal{D}, [t/x]\mathcal{D} \implies P}{\Delta, \forall x : \tau.D \implies P}$$

$$\frac{\Delta, G \supset P \implies G \qquad \Delta, G \supset P, P \implies P'}{\Delta, G \supset P \implies P'}$$

With this new proof system, we can show that it is equivalent to the Horn Fragment.

Sound If $\Delta \overset{u}{\implies} G$ then $\Delta \implies G$
If $\Delta \overset{e}{\implies} P$ then $\Delta \implies P$
(you can prove this via induction, associating the corresponding rules from our new system to the Horn Fragment rules)

Complete If $\Delta \implies G$ then $\Delta \overset{u}{\implies} G$

Example of Proof Search

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\Delta \implies even\ z}\, e\_z \qquad \dfrac{}{\Delta, add\ (suc\ z) \implies add\ (suc\ z)}\,\text{init}}{\Delta, even\ z \supset odd\ (suc\ z) \implies odd\ (suc\ z)}\,\supset \text{L}}{\Delta \implies odd\ (suc\ z)}\,\forall\ \text{L}o\_s \qquad \dfrac{}{\Delta, even\ (suc\ (suc\ z)) \implies even\ (suc\ (suc\ z))}\,\text{init}}{\Delta, odd\ (succ\ z) \supset even\ (suc\ (suc\ z)) \implies even\ (suc\ (suc\ z))}\,\supset \text{L}$$
$$\Delta \implies even(suc\ (suc\ z))\,\forall\ \text{L}\ e\_s$$

The consecutive uses of L rules are called **focusing** ($\supset$ L and $\forall$ Lo_s are coupled together and $\supset$ L and $\forall$ L e_s are coupled together)

We can do the proof either forwards (top to bottom) or backwards (bottom to top). Logic programming goes bottom up. Surprisingly, when you have to make a choice, you can stick with it as it won't make a difference. So we can now make some adjustments to our rules to achieve a narrower search space:

1. Apply init and $\wedge$ R until $\Delta \implies P$ (atomic formula)
   $\Delta \overset{u}{\implies} G_1, \Delta > D \implies P$

$$\frac{\Delta \overset{u}{\implies} G_1 \quad \Delta \overset{u}{\implies} G_2}{\Delta \overset{u}{\implies} G_1 \wedge G_2} \qquad \frac{\Delta > D \implies P \quad D \in \Delta}{\Delta \overset{u}{\implies} P}$$

Where: $\dfrac{P \in \Delta}{\Delta > P \implies P}$ $\dfrac{\Delta > [t/x]\mathcal{D} \implies P}{\Delta > \forall x : \tau . D \implies P}$

$$\frac{\Delta, G \supset P \implies G \quad \Delta > G \supset P, P \implies P'}{\Delta > G \supset P \implies P'}$$

2. Once we pick a clause from Δ, we focus on it.

Now how do we show this is sound and complete?
If $\Delta \implies G$ then $\Delta \overset{u}{\implies} G$
If $\Delta \implies P$ then $\Delta > D \implies P$ for $D \in \Delta$

Case $\mathcal{D} = \dfrac{\begin{matrix}\mathcal{D}_1 & \mathcal{D}_2 \\ \Delta \implies G_1 & \Delta \implies G_2\end{matrix}}{\Delta \implies G_1 \wedge G_2} \wedge \text{R}$

$\Delta \overset{u}{\implies} G_1$ by IH
$\Delta \overset{u}{\implies} G_2$ by IH
$\Delta \overset{u}{\implies} G_1 \wedge G_2$ by $\wedge \text{R}_u$

Case $\mathcal{D} = \dfrac{\Delta, G_1 \supset P_1 \implies G_1 \quad \Delta, G_1 \supset P_1, P_1 \implies G}{\Delta, G_1 \supset P_1 \implies G}$

To show $\Delta, G_1 \supset P_1 \overset{u}{\implies} G$
$\Delta, G_1 \supset P_1 \overset{u}{\implies} G_1$ by IH
$\Delta, G_1 \supset P_1, P_1 \overset{u}{\implies} G$ by IH
Where do we go from here? We have no implication rule (we worked hard to get rid of the rule). This suggests that we need a lemma.

Postponement Lemma If $\Delta, G_1 \supset P_1 \overset{u}{\implies} G_1$ and $\Delta, G_1 \supset P_1, P_1 \overset{u}{\implies} G$ then $\Delta, G_1 \supset P_1 \overset{u}{\implies} G$. Basically we can postpone working on the left for higher up in the tree.

Case $\mathcal{D} = \dfrac{\Delta, G_1 \supset P_1 \implies G_1 \quad \Delta, G_1 \supset P_1, P_1 \implies P}{\Delta, G_1 \supset P_1 \implies P}$

To show : $(\Delta, G_1 \supset P_1) > D \implies P$ where $D \in (\Delta, G_1 \supset P_1)$
$\Delta, G_1 \supset P_1, P_1 > D' \implies P$ by IH
$\Delta, G_1 \supset P_1 \overset{u}{\implies} G_1$ by postponement lemma
This is another form of the postponement lemma. Both forms of the postponement lemma can be proven via induction.

15 02/26/19

Logic Programming

Recall Horn Clause $C := P(t_1 \dots t_n) \mid G \supset P(t_1 \dots t_n) \mid \forall x : \tau.C$
Goals $G := P(t_1 \dots t_n) \mid G_1 \wedge G_2$
Invertible on the right (asynchronous)
Focusing (postpone certain choices) (synchronous)
$\Delta \implies G$ uniform
$\Delta > C \implies P(t_1 \dots t_n)$ focusing

15.1 Higher-Order Logic Programming

(Hereditary Harrop formulas)
$F := P(t_1 \dots t_n) \mid F_1 \supset F_2 \mid \forall x : \tau.F$
Horn Fragment:
$P(t_1 \dots t_n) \wedge Q(s_1 \dots s_k) \supset R(w_1 \dots w_m) \to P(t_1 \dots t_n) \supset Q(s_1 \dots s_k) \supset R(w_1 \dots w_m)$

Uniform Phase $\dfrac{\Delta \implies [a/x]F}{\Delta \implies \forall x : \tau.F} \forall \mathrm{R}^a \dfrac{\Delta, u : F_1 \implies F_2}{\Delta \implies F_1 \supset F_2} \dfrac{\Delta > F \implies P(t_1 \dots t_n)}{\Delta \implies P(t_1 \dots t_n)} F \in \Delta$
Where the last rule is new.

Focusing phase $\dfrac{}{\Delta > P(t_1 \dots t_n) \implies P(t_1 \dots t_n)} \quad \dfrac{\Delta > [t/x]F \implies P(t_1 \dots t_n)}{\Delta > \forall x : \tau.F \implies P(t_1 \dots t_n)}$

$$\frac{\Delta \implies F_1 \qquad \Delta > F_2 \implies P(t_1 \dots t_n)}{\Delta > F_1 \supset F_2 \implies P(t_1 \dots t_n)}$$

(To not reprove things like F_1, you could keep a memoization table on the side)
"Reflect" natural deduction into HHF using one predicate nd A

ND Rules

$$\frac{}{T \text{ true}}\, T\,\text{I}$$

$$\frac{A \text{ true} \qquad B \text{ true}}{(A \wedge B) \text{ true}} \wedge \text{I}$$

$$\frac{A \wedge B \text{ true}}{A \text{ true}}\, \text{and}E_1$$

$$\frac{\begin{array}{c}\overline{A \text{ true}}\,\text{u} \\ \vdots \\ B \text{ true}\end{array}}{(A \supset B) \text{ true}} \supset \text{I}^u$$

$$\frac{A \vee B \text{ true} \qquad \begin{array}{c}\overline{A \text{ true}}\,\text{u} \\ \vdots \\ C \text{ true}\end{array} \qquad \begin{array}{c}\overline{B \text{ true}}\,\text{v} \\ \vdots \\ C \text{ true}\end{array}}{C \text{ true}}$$

$$\frac{A \wedge B \text{ true} \qquad \begin{array}{c}\overline{A \text{ true}}\,\text{u} \quad \overline{B \text{ true}}\,\text{v} \\ \vdots \\ C \text{ true}\end{array}}{C \text{ true}} \wedge \text{E}$$

Judgment $\vdash$ even n

Rules:

$$\frac{}{\vdash \text{even } 0}\,\text{ev\_z} \qquad \frac{\vdash \text{even } n}{\vdash \text{even } (ss\ n)}$$

ev_z: even z

ev_s: $\forall n$: nat. even $n \supset$ even $(ss\ n)$

HHF

nd_topI: nd top

nd_andI: $\forall A : o . \forall B : o$ (o is our predicate)

nd A $\supset$ (nd $B \supset$ nd (and A B))

nd_andE_1:$\forall A : o . \forall B : o$. nd (and A B) $\supset$ nd A

Hypothetical derivation, use $\supset$ from HHF

nd_impI: $\forall A : o . \forall B : o . \underline{(nd\ A \supset nd\ B)} \supset nd\ (imp\ A\ B)$

(underline corresponds to the hypothetical derivation)

nd_orE: $\forall A : o . \forall B : o . \forall C : o$ nd (or A B) $\supset$
$\underline{(\text{nd } A \supset \text{nd } C)} \supset \underline{(nd\ B \supset nd\ C)} \supset nd\ C$

Translating HHF to FOL:

o:	type
top:	o
and:	o $\to$ o $\to$ o
imp:	o $\to$ o $\to$ o
or:	o $\to$ o $\to$ o
all:	? $\to$ o
nd	is predicate on o
nd:	o $\to$ type

Phases with terms

Uniform Phase $\dfrac{\Delta \implies M : [a/x]F}{\Delta \implies \lambda a.M : \forall x:\tau.F}\ \forall\ \mathrm{R}^a \quad \dfrac{\Delta, u : F_1 \implies M : F_2}{\Delta \implies \lambda u.M : F_1 \supset F_2}$

$\dfrac{x : F \in \Delta \quad \Delta > F \implies R : P(t_1 \dots t_n)}{\Delta \implies R : P(t_1 \dots t_n)}$ Where the last rule is new.

Focusing phase $\dfrac{}{\Delta > R : P(t_1 \dots t_n) \implies R : P(t_1 \dots t_n)} \quad \dfrac{\Delta > R\ t : [t/x]F \implies R_0 : P(t_1 \dots t_n)}{\Delta > R : \forall x:\tau.F \implies R_0 : P(t_1 \dots t_n)}$

$\dfrac{\Delta \implies M : F_1 \qquad \Delta > R\ M : F_2 \implies R_0 : P(t_1 \dots t_n)}{\Delta > R : F_1 \supset F_2 \implies R_0 : P(t_1 \dots t_n)}$

Can define inference rules in Beluga and proof terms and use it as a proof checker.

16 02/28/19

Midterm (Homework 1 and 2, no normal, sequent, etc.)

1. ND: propositional & 1^{st} order
2. Inductive proofs
3. New rules (and prove how they preserve provability)

Example Source Language

Terms $M := \lambda x.M \mid M_1\ M_2 \mid nor\ (x.M_1)(y.M_2) \mid abort_l M_0\ M_1 \mid abort_r M_0\ M_2$

Types $A := A_1 \supset A_2 \mid A_1 \triangledown A_2$

Contexts $\Gamma := \cdot \mid \Gamma, x : A$

Target Language

Terms $N := \lambda x.N \mid N_1\ N_2 \mid (N_1, N_2) \mid fst\ N \mid snd\ N \mid abort\ N$

Types $B := B_1 \supset B_2 \mid B_1 \wedge B_2 \mid \bot$

Contexts $\Delta := \cdot \mid \Delta, x : B$

Want to show that source language preservers provability of target language.

Rules:

$$\frac{\Gamma \vdash M_0 : A_1 \triangledown A_2 \qquad \Gamma \vdash M_1 : A_1}{\Gamma \vdash abort_L\ M_0\ M_1 : C}$$

$$\frac{\Gamma \vdash M_0 : A_1 \triangledown A_2 \qquad \Gamma \vdash M_2 : A_2}{\Gamma \vdash abort_R\ M_0\ M_2 : C}$$

$$\frac{\Gamma, x : A_1 \vdash M_1 : p \qquad \Gamma, y : A_2 \vdash M_2 : q}{\Gamma \vdash nor(x.M_1)(y.M_2) : A_1 \overline{\vee} A_2}$$

$$\frac{\Gamma, x : A_1 \vdash M : A_2}{\Gamma \vdash \lambda x.M : A_1 \supset A_2}$$

$$\frac{\Gamma \vdash M_0 : A_1 \supset A_2 \qquad \Gamma \vdash M_1 : A_1}{\Gamma \vdash M_0\ M_1 : A_2}$$

Our rules are not annotated in a certain way to ensure type uniqueness because we don't care about that right now, but if we wanted to guarantee type uniqueness, we'd have to annotate our terms more.

Local Soundness

$$\frac{\dfrac{\dfrac{\mathcal{D}_1}{\Gamma, x : A_1 \vdash M_1 : p} \qquad \dfrac{\mathcal{D}_2}{\Gamma, y : A_2 \vdash M_2 : q}}{\Gamma \vdash nor(x.M_1)(y.M_2) : A_1 \overline{\vee} A_2} \qquad \dfrac{\mathcal{D}'}{\Gamma \vdash M' : A_1}}{\Gamma \vdash abort_L(nor(x.M_1)(y.M_2))M' : C} \implies \begin{array}{l}[\mathcal{D}'/x, C/p]\mathcal{D}_1 \\ \Gamma \vdash [M'/x]M_1 : C\end{array}$$

We end up with the following reduction rule:
$abort_L(nor(x.M_1)(y.M_2))M' \implies [M'/x]M_1$

In general, with hypothetical derivations for soundness, you will end up with a substitution.

We then repeat this for abort right and combine them, just like for conjunction local soundness, where we had two elimination rules and we made two proof trees.

Local Completeness

$$\dfrac{\mathcal{D}}{\Gamma \vdash M_0 : A_1 \overline{\vee} A_2} \implies \dfrac{\dfrac{\dfrac{\dfrac{\mathcal{D}}{\Gamma \vdash M_0 : A_1 \overline{\vee} A_2}}{\Gamma, x : A_1 \vdash M_0 : A_1 \overline{\vee} A_2}\,\text{weakening} \qquad \dfrac{}{\Gamma, x : A_1 \vdash x : A_1}}{\Gamma, x : A_1 \vdash abort_L\ M_0\ x : p} \qquad \dfrac{\dfrac{\dfrac{\mathcal{D}}{\Gamma \vdash M_0 : A_1 \overline{\vee} A_2}}{\Gamma, y : A_2 \vdash M_0 : A_1 \overline{\vee} A_2}\,\text{weakening} \qquad \dfrac{}{\Gamma, y : A_2 \vdash y : A_2}}{\Gamma, y : A_2 \vdash abort_R\ M_0\ y : q}}{\Gamma \vdash nor(x.abort_L\ M_0\ x)(y.abort_R M_0\ y) : A_1 \overline{\vee} A_2}\,\overline{\vee}\ \mathrm{I}^{x,y,p,q}$$

Translation function

$$\begin{aligned}[\![M]\!] &= N \\ [\![\lambda x.M]\!] &= \lambda x.[\![M]\!] \\ [\![x]\!] &= x \\ [\![nor(x.M_1)(y.M_2)]\!] &= (\lambda x.[\![M_1]\!], \lambda y.[\![M_2]\!]) \\ [\![abort_L\ M_0\ M_1]\!] &= abort_L[(fst\ [\![M_0]\!])[\![M_1]\!]]\end{aligned}$$

$$
\begin{aligned}
[\![A]\!] &= B \\
[\![A_1 \supset A_2]\!] &= [\![A_1]\!] \supset [\![A_2]\!] \\
[\![A_1 \bar{\vee} A_2]\!] &= \neg[\![A_1]\!] \wedge \neg[\![A_2]\!] = ([\![A_1]\!] \supset \bot) \wedge ([\![A_2]\!] \supset \bot) \\
[\![\Gamma]\!] &= \Delta \\
[\![\cdot]\!] &= \cdot \\
[\![\Gamma, x : A]\!] &= [\![\Gamma]\!], x : [\![A]\!]
\end{aligned}
$$

Thm If $\Gamma \vdash M : A$ then $[\![\Gamma]\!] \vdash [\![M]\!] : [\![A]\!]$.

Proof by induction on $\mathcal{D} :: \Gamma \vdash M : A$

Case $\mathcal{D} = \dfrac{\begin{array}{cc}\mathcal{D}_1 & \mathcal{D}_2 \\ \Gamma, x : A_1 \vdash M_1 : p & \Gamma, y : A_2 \vdash M_2 : q\end{array}}{\Gamma \vdash nor(x.M_1)(y.M_2) : A_1 \bar{\vee} A_2} \bar{\vee}\ \mathrm{I}^{x,y,p,q}$

$[\![\Gamma]\!] \vdash [\![nor(x.M_1)(y.M_2)]\!] : [\![A_1 \bar{\vee} A_2]\!]$

$[\![\Gamma]\!] \vdash (\lambda x.[\![M_1]\!], \lambda y.[\![M_2]\!]) : \neg[\![A_1]\!] \wedge \neg[\![A_2]\!]$ (Want to show)

$\mathcal{E}_1 :: [\![\Gamma, x : A_1]\!] \vdash [\![M_1]\!] : p$ by IH on $\mathcal{D}_1$

$[\bot\ /p]\mathcal{E}_1 :: [\![\Gamma]\!], x : [\![A_1]\!] \vdash [\![M_1]\!] : \bot$

$\mathcal{E}_2 :: [\![\Gamma]\!], y : [\![A_2]\!] \vdash [\![M_2]\!] : \bot$ IH on $\mathcal{D}_2$

$$
\dfrac{\dfrac{\begin{array}{c}[\bot\ /p]\mathcal{E}_1 \\ [\![\Gamma]\!], x : [\![A_1]\!] \vdash [\![M_1]\!] : \bot\end{array}}{[\![\Gamma]\!] \vdash \lambda x.[\![M_1]\!] : \neg[\![A_1]\!]} \supset \mathrm{I} \quad \dfrac{\begin{array}{c}[\bot\ /p]\mathcal{E}_2 \\ [\![\Gamma]\!], y : [\![A_2]\!] \vdash [\![M_2]\!] : \bot\end{array}}{[\![\Gamma]\!] \vdash \lambda x.[\![M_2]\!] : \neg[\![A_2]\!]} \supset \mathrm{I}}{[\![\Gamma]\!] \vdash (\lambda x.[\![M]\!], \lambda y.[\![M_1]\!]) : \neg[\![A_1]\!] \wedge \neg[\![A_2]\!]} \wedge \mathrm{I}
$$

Case $\mathcal{D} = \dfrac{\begin{array}{cc}\mathcal{D}_1 & \mathcal{D}_2 \\ \Gamma \vdash M_0 : A_1 \bar{\vee} A_2 & \Gamma \vdash M_1 : A_1\end{array}}{\Gamma \vdash abort_L\ M_0\ M_1 : C}$

$[\![\Gamma]\!] \vdash [\![abort_L\ M_0\ M_1]\!] : [\![C]\!]$

$[\![\Gamma]\!] \vdash abort[(fst\ [\![M_0]\!])[\![M_1]\!]] : [\![C]\!]$ (want to show)

$\mathcal{E}_1 :: [\![\Gamma]\!] \vdash [\![M_0]\!] : \neg[\![A_1]\!] \wedge \neg[\![A_2]\!]$ IH on $\mathcal{D}_1$

$\mathcal{E}_2 :: [\![\Gamma]\!] \vdash [\![M]\!] : [\![A_1]\!]$ IH on $\mathcal{D}_2$

$$
\dfrac{\dfrac{\dfrac{\mathcal{E}_1}{[\![\Gamma]\!] \vdash [\![M_0]\!] : \neg[\![A_1]\!] \wedge \neg[\![A_2]\!]} \wedge \mathrm{E}_L}{[\![R]\!] \vdash fst[\![M_0]\!] : \neg[\![A_1]\!]} \quad \dfrac{\mathcal{E}_2}{[\![\Gamma]\!] \vdash [\![M_1]\!] : [\![A]\!]}}{[\![\Gamma]\!] \vdash (fst[\![M_0]\!])[\![M_1]\!] : \bot} \supset \mathrm{E}
$$

$$
\dfrac{}{[\![\Gamma]\!] \vdash abort[(fst[\![M_0]\!])[\![M_1]\!]] : [\![C]\!]} \bot\ \mathrm{E}
$$

Type Uniqueness Rules annotated for type uniqueness:

$$\frac{\Gamma \vdash M_0 : A_1 \overline{\vee} A_2 \qquad \Gamma \vdash M_1 : A_1}{\Gamma \vdash abort_L^C\ M_0\ M_1 : C}$$

$$\frac{\Gamma \vdash M_0 : A_1 \overline{\vee} A_2 \qquad \Gamma \vdash M_2 : A_2}{\Gamma \vdash abort_R^C\ M_0\ M_2 : C}$$

$$\frac{\Gamma, x : A_1 \vdash M_1 : p \qquad \Gamma, y : A_2 \vdash M_2 : q}{\Gamma \vdash nor(x : A_1.M_1)(y : A_2.M_2) : A_1 \overline{\vee} A_2}$$

If $\Gamma \vdash M : A_1'$ and $\Gamma \overset{\mathcal{E}}{\vdash} M : A_2'$ then $A_1' = A_2'$

Proof by induction on $\mathcal{D} :: \Gamma \vdash M : A_1'$

Case $\mathcal{D} = \dfrac{\begin{matrix}\mathcal{D}_1 & \mathcal{D}_2 \\ \Gamma, x : A_1 \vdash M_1 : p & \Gamma, y : A_2 \vdash M_2 : q\end{matrix}}{\Gamma \vdash nor(x : A_1.M_1)(y : A_2.M_2)}\ \overline{\vee}\ \mathrm{I}^{x,y,p,q}$

$M = nor(x : A_1.M_1)(y : A_2.M_2)$	by assumption
$A_1' = A_1 \overline{\vee} A_2$	by assumption
$\mathcal{E} :: \Gamma \vdash nor(x : A_1.M_1)(y : A_2.M_2) : A_2'$	by assumption
$\mathcal{E}_1 :: \Gamma, x : A_1 \vdash M_1 : p'$	by inversion
$\mathcal{E}_2 :: \Gamma, y : A_2 \vdash M_2 : q'$	by inversion
$A_2' = A_1 \overline{\vee} A_2$	by inversion

17 03/14/19

17.1 Modal Logic (S4)

Logic in a type of world. In particular we will look at S4 Modal Logic. In S4, we have two laws:

- Reflexivity + Transitivity to reason about the truth in different worlds
- We will distinguish between "global" notion of truth (valid/validity) and "local" notion of truth (two worlds)

  Computationally we can think of it as a network topology (every computer lives in its own world). Want to talk about truth of a program at a given node in the network and truth at every possible node in a network.

  We want to see what the difference between validity and truth is.

Today:

- Necessity: A proposition A is necessarily true i.e. valid, globally true, true in every possible world

  Can extend this with:

  - Possibility (it is possible that in some world it is true)
  - "Time": it is true in the next timestamp (lax modality), not thinking of absolute time (e.g. at exactly 5:05 pm it will be true), but rather clock ticks, think of time in an abstract fashion

Distinguish between $\underbrace{\text{validity}}_{A \text{ valid}}$ and $\underbrace{\text{truth}}_{A \text{ true}}$.

Def. of Validity If $\cdot \vdash A$ true then $\vdash A$ valid
If $\vdash A$ valid then $\vdash A$ true
We can express that things are true unconditionally and everyone will be able to access that truth
$\underbrace{A_1 \text{ valid}, \ldots, A_n \text{ valid}}_{\Delta\text{:global assumptions}}; \underbrace{B_1 \text{ true}, \ldots, B_k \text{ true}}_{\Gamma\text{:local assumptions}} \vdash C$ true
So we are distinguishing between what everyone knows and what I know.
So how can we then prove something is true?

$$\frac{A \text{ true} \in \Gamma}{\Delta; \Gamma \vdash A \text{ true}}$$

$$\frac{A \text{ valid} \in \Delta}{\Delta; \Gamma \vdash A \text{ true}}$$

Substitution Principles:

1. If $\Delta; \Gamma \vdash A$ true and $\Delta; \Gamma, A \text{ true}, \Gamma' \vdash C$ true then $\Delta; \Gamma, \Gamma' \vdash C$ true
2. If $\Delta; \cdot \vdash B$ true and $\Delta, B \text{ valid}, \Delta'; \Gamma \vdash C$ true then $(\Delta, \Delta'); \Gamma \vdash C$ true

Necessity: $\Box A$ internalizes validity
A is necessarily true
Prop. $A, B := T \mid A \supset B \mid A \wedge B \mid \Box A$
Term $M, N := \lambda x.M \mid M\ N \mid box\ M \mid let\ box\ u = M\ in\ N$

Introduction:

$$\frac{\Delta; \cdot \vdash A \text{ true}}{\Delta; \Gamma \vdash \Box A \text{ true}} \Box \text{ I}$$

Elimination:

$$\frac{\Delta; \Gamma \vdash \Box A \text{ true}}{\Delta; \Gamma \vdash A \text{ true}}$$

Alternative?

$$\frac{\Delta; \Gamma \vdash \Box A \text{ true}}{\Delta; \cdot \vdash A \text{ true}} \implies \text{UNSOUND}$$

Local Soundness:

$$\frac{\dfrac{\dfrac{\mathcal{D}}{\Delta; \cdot \vdash A \text{ true}}}{\Delta; \Gamma \vdash \Box A \text{ true}} \Box \text{ I}}{\Delta; \Gamma \vdash A \text{ true}} \implies \begin{array}{c} \mathcal{D}^{\text{weak}} \\ \Delta; \Gamma \vdash A \text{ true} \end{array}$$

Local Completeness:

$$\begin{array}{c} \mathcal{D} \\ \Delta; \Gamma \vdash \Box A \text{ true} \end{array} \implies \begin{array}{c} \mathcal{D} \\ \dfrac{\Delta; \Gamma \vdash \Box A \text{ true}}{\Delta; \Gamma \vdash A \text{ true}} \\ \vdots \\ \Delta; \Gamma \vdash \Box A \text{ true} \end{array} (\textbf{Too weak})$$

So what elimination rule should we have? Take inspiration from disjunction elimination.

$$\frac{\Delta; \Gamma \vdash \Box A \text{ true} \qquad \Delta, A \text{ valid}; \Gamma \vdash C \text{ true}}{\Delta; \Gamma \vdash C \text{ true}}$$

Revisiting Local Soundness

$$\frac{\dfrac{\dfrac{\mathcal{D}_1}{\Delta; \cdot \vdash A \text{ true}}}{\Delta; \Gamma \vdash \Box A \text{ true}} \Box \text{ I} \qquad \dfrac{\mathcal{D}_2}{\Delta, u : A \text{ valid}; \Gamma \vdash C \text{ true}}}{\Delta; \Gamma \vdash C \text{ true}} \Box \text{ E} \implies \Delta; \Gamma \vdash C \text{ true (Substitution Lemma) } [\mathcal{D}_1/u]\mathcal{D}_\in$$

Local Completeness

$$\begin{array}{c} \mathcal{D} \\ \Delta; \Gamma \vdash \Box A \text{ true} \end{array} \implies \frac{\begin{array}{c} \mathcal{D} \\ \Delta; \Gamma \vdash \Box A \text{ true} \end{array} \qquad \dfrac{\dfrac{A \text{ valid} \in \Delta}{\Delta, A \text{ valid}; \cdot \vdash A \text{ true}}}{\Delta, A \text{ valid}; \Gamma \vdash \Box A \text{ true}}}{\Delta; \Gamma \vdash \Box A \text{ true}}$$

Examples We want to prove: $A \supset \Box A, \Box A \supset A, \Box A \supset \Box\Box A$

$$\dfrac{\dfrac{}{\cdot;\Box(A\supset B)\text{ true},\Box A\text{ true}\vdash\Box A\text{ true}}\quad\dfrac{A\text{ valid};\Gamma\vdash\Box(A\supset B)\text{ true}\qquad\dfrac{\dfrac{\dfrac{\dfrac{}{A\text{ valid},A\supset B\text{ valid};\cdot\vdash A\supset B\text{ true}}\quad\dfrac{}{\ldots\vdash A\text{ true}}}{A\text{ valid},A\supset B\text{ valid};\cdot\vdash B\text{ true}}}{A\text{ valid},A\supset B\text{ valid};\Gamma\vdash\Box B\text{ true}}}{\cdot;\Box(A\supset B)\text{ true},\Box A\text{ true}\vdash\Box A\text{ true}}}{\dfrac{\overbrace{\cdot;\Box(A\supset B)\text{ true},\Box A\text{ true}}^{\Gamma}\vdash\Box B\text{ true}}{\cdot;\Box(A\supset B)\text{ true}\cdot\vdash\Box()}}}{\cdot;\cdot\vdash\Box(A\supset B)\supset\Box A\supset\Box B\text{ true}}$$

In general, we want to promote validity to truths/assumptions.

$$\dfrac{\dfrac{\cdot;\Box A\text{ true}\vdash\Box A\text{ true}\qquad\dfrac{}{A\text{ valid};\Box A\text{ true}\vdash A\text{ true}}}{\cdot;\Box A\text{ true}\vdash A\text{ true}}}{\cdot;\cdot\vdash\Box A\supset A\text{ true}}$$

$$\dfrac{\dfrac{\cdot;\Box A\text{ true}\vdash\Box A\text{ true}\qquad\dfrac{\dfrac{}{A\text{ valid};\cdot\vdash A\text{ true}}}{A\text{ valid};\cdot\vdash\Box A\text{ true}}}{A\text{ valid};\Box A\text{ true}\vdash\Box\Box A\text{ true}}}{\cdot;\Box A\text{ true}\vdash\Box\Box A\text{ true}}$$

$$\cdot;\cdot\vdash\Box A\supset\Box\Box A\text{ true}$$

Variables We have two different classes of variables (depending on where they come from)

$$\dfrac{x : A\text{ true}\in\Gamma}{\Delta;\Gamma\vdash x : A\text{ true}}$$

$$\dfrac{u : A\text{ valid}\in\Delta}{\Delta;\Gamma\vdash u : A\text{ true}}$$

Rules with Terms

$$\dfrac{\Delta;\cdot\vdash M : A\text{ true}}{\Delta;\Gamma\vdash box\ M : \Box A\text{ true}}$$

$$\dfrac{\Delta;\Gamma\vdash M : \Box A\text{ true}\qquad\Delta,u : A\text{ valid};\Gamma\vdash N : C\text{ true}}{\Delta;\Gamma\vdash let\ box\ u = M\ in\ N : C\text{ true}}$$

Staged Programming Distinguish between code that is generated **AND** code that is actually doing the work of generating code (code generator, for example, $\Box$ can be the code generator and A can be the code that is generated)

Reminder from 302: `pow 2`$\rightarrow$ squaring function $x * x * 1$

`pow 3` $\rightarrow x * x * x * 1$

```
(* pow n x = x^n*)
pow : int -> □ (int -> int)
let rec pow n =
if n = 0 then
   box(λx.1)
else
   box f = pow (n-1) in
   box(λx.x*f x)
end if
```

`pow 2` $\neq$ `box (λx. x * x * 1)`

Instead: `box (λx . x *` $(\lambda x_1.x_1*(\lambda x_0.1)\ x_1)$ `x)`

Administrative redeces

18 03/19/19

Papers posted on myCourses:

- Davies, Pfenning: A judgmental reconstruction of modal logic, MSCS 2001
- Nanevski, Pfenning, Pientka: Contextual Modal Type Theory, ACM Transactions on Computational Logic, 2008

Don't have to read the whole papers, just look at them with your notes and read the sections relevant to what we covered.

Recap $\mathcal{A} := A \supset B \mid \Box A$

$\Box A \hat{=} A$ is true in every world "A valid"

$$\underbrace{\Delta}_{\substack{\text{global assumptions} \\ A \text{ valid}}} \quad ; \quad \underbrace{\Gamma}_{\substack{\text{current local assumptions} \\ \mathcal{A} \text{ true}}} \quad \vdash A \text{ true}$$

$$\frac{u : A \text{ valid} \in \Delta}{\Delta; \Gamma \vdash A \text{ true}}$$

$$\frac{\Delta; \cdot \vdash A \text{ true}}{\Delta; \Gamma \vdash \Box A \text{ true}} \Box \text{ I}$$

$$\frac{\Delta; \Gamma \vdash \Box A \text{ true} \qquad \Delta, u : A \text{ valid}; \Gamma \vdash C \text{ true}}{\Delta; \Gamma \vdash C \text{ true}}$$

We saw the power function:
$pow : int \to \Box(int \to int)$
$pow\ 3 \twoheadrightarrow box(fun\ x \to x * x * x * 1)$
$pow\ n = if\ n = 0\ then\ \boxed{?}\ else\ \boxed{?}$, where $\boxed{?}$ after the then is a hole for n
$\forall x : nat.even\ x \supset \boxed{?}$

Because of these holes, we want to redefine things a bit (like in code, we want to be able to import things without having to change overall logic): $\mathcal{A} := A \supset B \mid [\psi]A$
$[\psi]A \hat{=} A$ is true in every world where we have assumptions ψ
$[\psi]A \cong \Box(\psi \supset A)$
$[x_1 : B_1, \ldots, x_n : B_n]A \cong \Box(B_1 \supset B_2 \supset \ldots \supset B_n \supset A)$
$\implies$ Logically equivalent (we haven't expanded the proof system)
$\implies$ Computationally different
$\cong$ Structure of proofs is different

$$\underbrace{\Delta}_{\substack{\text{global assumptions} \\ [\psi]A}} \quad ; \quad \underbrace{\Gamma}_{\substack{\text{current local assumptions} \\ \mathcal{A} \text{ true}}} \quad \vdash A \text{ true}$$

$$\frac{u : [\psi]A \in \Delta \qquad \Delta; \Gamma \vdash \psi}{\Delta; \Gamma \vdash A \text{ true}} \text{ hyp}^*$$

$$\frac{\Delta; \psi \vdash A \text{ true}}{\Delta; \Gamma \vdash [\psi]A \text{ true}} \Box \text{ I, Context switch}$$

$$\frac{\Delta; \Gamma \vdash [\psi]A \text{ true} \qquad \Delta, u : [\psi]A; \Gamma \vdash C \text{ true}}{\Delta; \Gamma \vdash C \text{ true}} \Box \text{ E}^u$$

$$\frac{\Delta, \Gamma \vdash A_1 \text{ true} \qquad \ldots \qquad \Delta; \Gamma \vdash A_n \text{ true}}{\Delta; \Gamma \vdash x_1 : A_1, \ldots, x_n : A_n = \psi} \text{ ctx}$$

$$\cfrac{\cfrac{\cfrac{}{\Delta; x_1 : B_1, \ldots, x_n : B_n \vdash A \text{ true}}\, n\text{-times}}{\Delta; \cdot \vdash B_1 \supset B_2 \supset \ldots \supset B_n \supset A} \supset \text{ I}}{\Delta; \Gamma \vdash \Box(B_1 \supset B_2 \supset \ldots \supset B_n \supset A)} \Box \text{ I}$$

Examples $\Box(C \supset A) \supset \Box(C \supset D \supset A)$

$$\dfrac{\dfrac{}{\cdot; [x:C]A \text{ true} \vdash [x:C]A \text{ true}} \quad \dfrac{\dfrac{\dfrac{}{u : [x:C]A \text{ valid}; C \text{ true}, D \text{ true} \vdash C \text{ true}}}{u : [x:C]A \text{ valid}; C \text{ true}, D \text{ true} \vdash A \text{ true}}\,\text{hyp}^*}{u : [x:C]A \text{ valid}; [x:C]A \text{ true} \vdash [x:C, y:D]A \text{ true}}\,\Box\,\text{I}}{\dfrac{\cdot; [x:C]A \text{ true} \vdash [x:C, y:D]A \text{ true}}{\cdot \vdash [x:C]A \supset [x:C, y:D]A \text{ true}}\,\supset\text{I}}\,\Box\,\text{E}$$

In this proof we don't waste time using $\supset$ E and instead use hyp*, nor do we use $\supset$ I.

$$\dfrac{\dots \quad \dfrac{\dfrac{\dfrac{}{u : [A]B, v : [A]([B]C); A \text{ true} \vdash A \text{ true}}}{u : [A]B, v : [A]([B]C); A \text{ true} \vdash [B]C \text{ true}} \quad \dfrac{\dfrac{\dfrac{}{u : [A]B, v : [A]([B]C), w : [B]C; A \text{ true} \vdash A \text{ true}}}{u : [A]B, v : [A]([B]C), w : [B]C; A \text{ true} \vdash B \text{ true}}\,\text{hyp}^*}{u : [A]B, v : [A]([B]C), w : [B]C; A \text{ true} \vdash C \text{ true}}\,\text{hyp}^*}{u : [A]B, v : [A]([B]C); A \text{ true} \vdash C \text{ true}}\,\Box\,\text{E}}{u : [A]B, v : [A]([B]C); \Gamma \vdash [A]C \text{ true}}\,\Box\,\text{I}}{\dfrac{}{\cdot; \underbrace{[A]B \text{ true}, [A]([B]C) \text{ true}}_{\Gamma} \vdash [A]C \text{ true}}}\,2 \text{ - times } \Box\,\text{E}$$

For the last line, we can't use $\Box$ I right away, as we'd drop all the assumptions we have. We therefore have to transfer these assumptions to our global context such that we can use them. It is hard to reason directly from assumptions.

This proof is similar to $(A \supset B \supset C) \supset (A \supset C)$, however the proof structure changes a lot.

$$\dfrac{\dfrac{}{\cdot; A \text{ true} \vdash A \text{ true}}}{\cdot; \vdash [A]A \text{ true}}\,\Box\,\text{I}$$

Similar to $[\cdot]A \supset A$ true.

If ψ in hyp* is empty, then things become trivially true, we don't have to prove everything in ψ and we get an axiom.

We cannot prove $[A]([B]C)$ true $\vdash [B]([A]C)$ true. We only have reflexivity and transitivity, no symmetry.

$$\dfrac{\dots \quad \dfrac{\dfrac{\dfrac{?}{u : [A]([B]C); B \text{ true} \vdash [A]C \text{ true}}}{u : [A]([B]C); \Gamma \vdash [B]([A]C) \text{ true}}}{\cdot; [A]([B]C) \text{ true} \vdash [B]([A]C) \text{ true}}\,\Box\,\text{E}$$

However, we can prove:
$[A][B]C \supset [[]B][A]C$
$[A, B]C \supset [A \wedge B]C$
$[A, B]C \supset [B, A]C$

Importing a global piece of code, with terms:

$$\frac{u : [\psi]A \in \Delta \qquad \Delta; \Gamma \vdash \overbrace{\sigma}^{\text{Substitution}} : \psi}{\Delta; \Gamma \vdash clo(u, \underbrace{\sigma}_{\text{Explicit/suspended substitution calculus}}) : A \text{ true}}\ \text{hyp}^*$$

Where clo is a closure.

$$\frac{\Delta; \psi \vdash M : A \text{ true}}{\Delta; \Gamma \vdash box(\psi.M) : [\psi]A \text{ true}}\ \Box \text{ I, Context switch}$$

$$\frac{\Delta; \Gamma \vdash M : [\psi]A \text{ true} \qquad \Delta, u : [\psi]A; \Gamma \vdash N : C \text{ true}}{\Delta; \Gamma \vdash let\ box\ u = M\ in\ N : C \text{ true}}$$

$$\frac{\Delta, \Gamma \vdash M_1 : A_1 \text{ true} \qquad \ldots \qquad \Delta; \Gamma \vdash M_n : A_n \text{ true}}{\Delta; \Gamma \vdash M_1/x_1, \ldots, M_n/x_n : x_1 : A_1, \ldots, x_n : A_n}$$

Comparing `pow` with global contexts to imported contexts (note on the RHS we still expect an input $x : int$ to generate the code):

```
pow : int → □ (int → int)
let rec pow n =
    if n = 0 then box(fun x → 1)
    else let box p = pow (n-1) in
                 (p : int→int)
      box (fun x → x * p x)
```

```
pow : int -> [x : int] int
let rec pow n =
    if n = 0 then box (x . 1)
    else let box p = pow (n-1) in
                 (p : [x':int]int)
      box (x . x * clo(p, x/x'))
```

Running version 1 gives us:

```
pow 2 → let box p = pow 1 in
          box (fun x → x * p x)
      → let box p = box (fun x1 → x1 * (fun x0 → 1) x1) in
          box (fun x2 → x2 * (fun x1 →x1 * (fun x0 →1)x1)x2)
```

Running version 2 gives us:

```
pow 2 → let box p = pow 1 in
          box (x2 . x2 * clos(p, x2/x1))
      → let box p = box (x1 . x1 * 1) in
          box (x2 . x2 * clo (p, x2/x1))
      → box (x2 . x2 * x2 * 1)
```

Therefore, in the second version, we do not need to introduce functions (functions are too complicated for what we want to achieve). Clearly the second version is much more efficient and does the same thing when we compare them side by side. It generates code depending on the runtime environment we provide it.

Another example: `let rec length l =`

```
 match l with
```

`|[]` $\to$ `?`

`|h::t` $\to$ `?`

`[l:`α `list, h:`α`, t:`α `list]int`, we substitute in `l`.

19 03/21/19

19.1 Recap of Box Notation

$$\frac{x : A \in \Gamma}{\Delta; \Gamma \vdash x : A}$$

$$\frac{u : [\psi]A \in \Delta \qquad \Delta; \Gamma \vdash \sigma : \psi}{\Delta; \Gamma \vdash \text{clo}(u, \sigma) : A}\ \text{hyp}^*$$

$$\frac{\Delta; \Gamma \vdash M_1 : A_1 \qquad \ldots \qquad \Delta; \Gamma \vdash M_n : A_n}{\Delta; \Gamma \vdash \underbrace{M_1/x_1 \ldots M_n/x_n}_{\sigma} : \underbrace{\psi}_{x_1:A_1 \ldots x_n:A_n}}$$

$$\frac{\Delta; \psi \vdash M : A}{\Delta; \Gamma \vdash \text{box}(\psi.M) : [\psi]A}\ \Box\ \text{I}$$

$$\frac{\Delta; \Gamma \vdash M : [\psi]A \qquad \Delta, u : [\psi]A; \Gamma \vdash N : C}{\Delta; \Gamma \vdash \text{let box } u = M \text{ in } N : C}\ \Box\ \text{E}$$

Today:

1. Revisiting substitution op
2. Subject reduction
3. ND $\to$ Sequent Calculus
4. Cut for contextual $\Box$

Reduction

$$\begin{array}{ll} (\lambda x.M)N & \to [N/x]M \\ \text{let box } u = \text{box}(\psi.M) \text{ in } N & \to [\![\psi.M/u]\!]N \end{array}$$

Coming up with the let reduction is tricky. `let box u = box(x.x+1) in box(y.y*clo(u,y/x))`

Idea

- Replace u by `(x.x+1)`
- Followed by applying the substitution $[y/x](x+1)$

$\to$ `box(y.y*` $\underbrace{y+1}_{[y/x](x+1)}$ `)`

$[M/x]N = N'$ Replace x (from Γ, local context) with M in N.

Capture avoiding!

$$
\begin{aligned}
[M/x]x &= M \\
[M/x]y &= y, x \neq y \\
[M/x](\lambda y.N) &= \lambda y.[M/x]N, y \notin FV(M) \\
&\quad \text{(avoid capture "local")} \\
[M/x](box(\psi.N)) &= box(\psi.N) \\
&\quad (N \text{ can only contain things in } \psi \text{ due to intro rule, no } x) \\
[M/x](let\ box\ u = N_1\ in\ N_2) &= let\ box\ u = [M/x]N_1\ in\ [M/x]N_2, u \notin FMV(M) \\
&\quad \text{(avoid capture "global", i.e. bound from let)} \\
[M/x](clo(u, \sigma)) &= clo(u, [M/x]\sigma)
\end{aligned}
$$

(Other substitution)

$$[\![\psi.M/u]\!]N = N'$$

Replace u (from global Δ with $(\psi.M)$ in N)

$$[\![\psi.M/u]\!]x = x[\![\psi.M/u]\!](\lambda y.N) = \lambda y.[\![\psi.M/u]\!]N$$

Don't need to take care of capture, since M is closed wrt ψ; in particular y cannot occur in M

$$
\begin{aligned}
[\![\psi.M/u]\!](box(\phi.N)) &= box(\phi.[\![\psi.M/u]\!]N) \\
[\![\psi.M/u]\!](let\ box\ v = N_1\ in\ N_2) &= let\ box\ v = [\![\psi.M/u]\!]N_1\ in [\![\psi.M/u]\!]N_2, v \notin FMV(M) \\
[\![\psi.M/u]\!]clo(u, \sigma) &= [[\![\psi.M/u]\!]\sigma]M \implies \\
[\![\psi.M/u]\!]clo(v, \sigma) &= clo(v, [\![\psi.M/u]\!]\sigma)
\end{aligned}
$$

Tough to come up with the closure rule here:

`let box u = box (x.x+1) in box(y.y*clo(u,y/x))`→`box(y.y*(y+1))`

Idea

1. (a) Replace u with $x + 1$

(b) Replace any free occurrence of u in (y/x)!

2. Eliminated the closure by applying $[y/x]$ to $(x+1)$, $[y/x](x+1)$

Subject Reduction If $\Delta; \Gamma \vdash M : A$ and $M \to M'$ then $\Delta; \Gamma \vdash M' : A$.
Case Analysis on $M \to M'$

Case: $(\lambda x.M)N \to [N/x]M$

$\Delta; \Gamma \vdash (\lambda x.M)N : B$	by assumption
$\Delta; \Gamma \vdash N : A$	
$\Delta; \Gamma \vdash \lambda x.M : A \supset B$	by inversion on $\supset$ E
$\Delta; \Gamma, x : A \vdash M : B$	by inversion on $\supset$ I
$\Delta; \Gamma \vdash [N/x]M : B$	by substitution Lemma for local context

Case: let box $u = \text{box}(\psi.M)$ in $N \to [\![\psi.M/u]\!]N$

$\Delta; \Gamma \vdash$ let box $u = \text{box}(\psi.M)$ in $M : B$	by assumption
$\Delta; u : [\psi]A; \Gamma \vdash N : B$	by inversion $\Box$E
$\Delta; \Gamma \vdash \text{box}(\psi.M) : [\psi]A$	
$\Delta; \psi \vdash M : A$	by inversion $\Box$ I
$\Delta; \Gamma \vdash [\![\psi.M/u]\!]N : B$	by substitution lemma for global context

So now we also want to show consistency, much like we did using normal proofs and sequent calculus. What was the idea behind sequent calculus?

Sequent Calculus (only normal derivations)
A way of proving consistency (we are not able to prove $\cdot; \cdot \vdash \bot$)

Normal ND

$$\frac{\Downarrow \Gamma^{\downarrow} \vdash r : A \downarrow \text{ Elim-Rules}}{\Uparrow \Gamma^{\downarrow} \vdash m : A \uparrow \text{ Intro-Rules}}$$

Sequent Calculus

$$\text{initial rule}\ \frac{}{\underbrace{\Uparrow}_{\text{Left rules}}\ \Gamma \implies \underbrace{A}_{\text{Right rules}}\ \Uparrow}$$

New rules:

$$\frac{\Delta; \psi \implies A}{\Delta; \Gamma \implies [\psi]A}\ \Box \text{ R}$$

$$\frac{\Delta; \Gamma, x : A \implies B}{\Delta; \Gamma \implies A \supset B} \supset \text{R}$$

$$\frac{\Delta; \Gamma \implies A \qquad \Delta; \Gamma, x : A \supset B, y : B \implies C}{\Delta; \Gamma, x : A \supset B \implies C} \supset \text{L}$$

$$\frac{}{\Delta; \Gamma, A \implies A} \text{ initial}$$

$$\frac{\Delta, u : [\psi]A; \Gamma, x : [\psi]A \implies C}{\Delta; \Gamma, x : [\psi]A \implies C} \Box\text{L}$$

$$\frac{(\Delta, u : [\psi]A); \Gamma \implies \psi \qquad \Delta, u : [\psi]A; \Gamma, A \implies C}{\Delta, u : \underbrace{[\psi]A}_{\psi \supset A}; \Gamma \implies C} \text{ reflect}$$

20 03/26/19

Recap of modal logic (with boxes):

$\Box A$, A is necessarily true or A is true in every world.

So far: $\underbrace{\Delta}_{\text{global}}; \underbrace{\Gamma}_{\text{local}} \vdash A$ true

"explicit" box and let-box

20.1 Alternative (implicit) characterization of $\Box A$

Today: Alternative characterization of $\Box A$.

- Kripke-style interpretation
- S4: Accessibility Relation over worlds
  - Reflexive
  - Transitive

$\underbrace{\Gamma_1; \Gamma_2; \ldots; \Gamma_n}_{\substack{\text{Context stack} \\ \text{initial world}}} \vdash A$ true

Idea: Consider Γ_i. Then every assumption in Γ_i is reachable from the worlds (context) $\Gamma_1; \ldots; \Gamma_{i-1}$

Based: R.Davies, F. Pfenning
A modal analysis of staged computation, ACM 2001.

Goal: Treat the elimination rule for $\Box A$ differently - implicit - (box - unbox), no let-box like we had before.

We use Ψ for a context stack. This will keep track of the order that assumptions were added (we will know in what world it was introduced).
$\Psi = \cdot; \Gamma_1; \Gamma_2; \ldots; \Gamma_n$

Variable Rule

$$\frac{x : A \in \Gamma}{\Psi; \Gamma \vdash x : A}$$

$$\frac{\Psi; \Gamma; \cdot \vdash M : A}{\Psi; \Gamma \vdash box\ M : \Box A}$$

Here we force ourselves not to refer directly to the assumptions we have from the other worlds, adding a new assumption from the current world.

$$\frac{\Psi; \Gamma \vdash M : \Box A}{\Psi; \Gamma; \Gamma_1; \ldots; \Gamma_n \vdash unbox_n\ M : A}$$

We import the fact that A is valid in our current world, have to bring in the other worlds.

Note: n can be 0. This rule degenerates to:

$$\frac{\Psi; \Gamma \vdash M : \Box A}{\Psi; \Gamma \vdash unbox_0\ M : A}$$

Properties such as reflexivity and transitivity will be reflected in the operations and structural properties of context stacks.

Local Soundness

$$\frac{\dfrac{\mathcal{D}}{\Psi; \Gamma; \cdot \vdash M : A}}{\dfrac{\Psi; \Gamma \vdash box\ M : \Box A}{\Psi; \Gamma; \Gamma_1; \ldots; \Gamma_n \vdash unbox_n\ (box\ M) : A}\ \Box\ \mathrm{I}}$$

Modal fusion + modal weakening on $\mathcal{D}$
$\implies \Psi; \Gamma; \Gamma_1; \ldots; \Gamma_n \vdash \{n/1\}M : A$
where $\{n/1\}$ means replace index 1 by n

Two weakening properties used:

1. Weakening within a context Γ.
2. Weakening with respect to context stacks. In particular: If $\Psi; \Gamma \vdash A$ true then $\Psi; \Gamma; \Gamma_1; \ldots; \Gamma_n \vdash A$ true

Example: Assume there is an $x : A$ in Γ s.t. $\Psi; \Gamma \vdash x : \Box A$. What is the witness/rule that allows us to prove: $\Psi; \Gamma; \Gamma_1; \ldots; \Gamma_n \vdash A$? Use unboxing similar to in local soundness: $\Psi; \Gamma; \Gamma_1; \ldots; \Gamma_n \vdash \mathit{unbox}_n\ x : A$. Keeping these indices is mainly only useful from a programming point of view (so we'll know where the assumption comes from), but otherwise we could get rid of it.
Just weakening isn't enough to prove soundness, it only does the transformation: $\Psi; \cdot \to \Psi; \cdot; \Gamma_1; \ldots; \Gamma_n$. We require an additional thing called fusion.

Context Fusion (Modal Fusion) If $\Psi; \Gamma; \Gamma'; \Psi' \vdash C$ true then $\Psi; (\Gamma, \Gamma'); \Psi' \vdash C$ true
Can only hold if we allow reflexivity (Γ is accessible from Γ)

Example

$$\dfrac{\dfrac{\dfrac{}{\Box(A \supset B) \vdash \Box(A \supset B)}\,\text{var}}{\Box(A \supset B); A \vdash A \supset B \text{ true}}\,\Box_1\,\text{E} \quad \dfrac{}{\Box(A \supset B); A \vdash A \text{ true}}\,\text{var}}{\Box(A \supset B); A \vdash B \text{ true}}\,\supset \text{E}$$

How does this change when we use modal fusion?

$$\dfrac{\dfrac{\dfrac{}{\Box(A \supset B), A \vdash \Box(A \supset B) \text{ true}}\,\text{var}}{\Box(A \supset B), A \vdash A \supset B \text{ true}}\,\Box_0\,\text{E} \quad \dfrac{}{\Box(A \supset B), A \vdash A \text{ true}}\,\text{var}}{\Box(A \supset B), A \vdash B \text{ true}}\,\supset \text{E}$$

So we see that the index of the unbox changes.

Local Completeness

$$\begin{array}{c}\mathcal{D}\\ \Psi; \Gamma \vdash \Box A \text{ true}\end{array} \quad \Longrightarrow \quad \dfrac{\dfrac{\begin{array}{c}\mathcal{D}\\ \Psi; \Gamma \vdash \Box A \text{ true}\end{array}}{\Psi; \Gamma; \cdot \vdash A \text{ true}}\,\Box\,\text{E}}{\Psi; \Gamma \vdash \Box A \text{ true}}\,\Box\,\text{I}$$

20.2 Translating explicit (box - let box) to implicit (box - unbox)

$\lambda x.\mathit{let\ box}\ u = x\ \mathit{in}\ u : \Box A \supset A \quad \to \lambda x.\mathit{unbox}\ (x)$
However we need to keep track of the depths of the lets for the index of unboxing.
Call the explicit box system E, the implicit box system M.

Environment $\rho := \cdot \mid \rho, \mathit{box}\ u = E$

Environment stack $R := \cdot \mid R; \rho$

$\underbrace{R}_{\text{Given the environment stack}} \quad \triangleright E \to M$

$$\frac{}{R \triangleright x \to x}\ \text{tx\_var}$$

$$\frac{R \triangleright E \to M}{R \triangleright \lambda x.E \to \lambda x.M}\ \text{tx\_lam}$$

$$\frac{R; \cdot \triangleright E \to M}{R \triangleright box\ E \to box\ M}$$

$$\frac{R; \rho, box\ u = E_1 \triangleright E_2 \to M}{R, \rho \triangleright let\ box\ u = E_1\ in\ E_2 \to M}$$

$$\frac{R_0; \rho'_n \triangleright E \to M}{R \triangleright u \to unbox_n\ M}$$

where R is $R_0; \underbrace{\rho_n}_{\rho'_n, box\ u=E, \rho''_n}; \ldots; \rho_1$. So we force things in-line with their index.

What is the relationship between context stacks $(\Psi; \Gamma)$ and the "global" context Δ?

$\Psi; \Gamma \models R : \Delta$

Intuitively:

$$\frac{\Theta_n; \ldots; \Theta_{i-1}; \Gamma_i \vdash \rho_i : \Theta_i}{\Psi_0; \Gamma_n; \ldots; \Gamma_0 \vdash \cdot; \rho_n; \ldots; \rho_0 : \Delta}$$

Where Δ is $\Theta_n, \Theta_{n-1}, \ldots, \Theta_0$, we have to carve it up into different subsets in order to split it into different sections ρ_i, since Δ is one flat object. Θ_i has access to each of the previous Θ_j (as we see above).

Soundness Theorem If $\Delta, \Gamma \vdash_e E : A$, $R \triangleright E \to M$ and $\Psi; \Gamma \models R : \Delta$ then $\Psi; \Gamma \vdash M : A$ (need to recover what Ψ is, which is why we have the third assumption to tell us something about R from $\Psi; \Gamma$).

Case: $\dfrac{}{R \triangleright x \to x}\ \text{tx\_var}$

$\Delta; \Gamma \vdash_e x : A$	by assumption
$x : A \in \Gamma$	by variable rule (e)
$\Psi; \Gamma \vdash_i x : A$	by var-rule (i)

Case: $\dfrac{R \triangleright E \to M}{R \triangleright \lambda x.E \to \lambda x.M}$

$\Delta; \Gamma \vdash \lambda x.M : A$	by assumption
$\Delta; \Gamma, x : A_1 \vdash M : A_2$ and $A = A_1 \to A_2$	by assumption
$\Psi; \Gamma \models R : A$	by assumption
$\Psi; \Gamma, x : A_1 \models R : \Delta$	by weakening
$\Psi; \Gamma, x : A_1 \vdash_i M : A_2$	by i.h.
$\Psi; \Gamma \vdash_i \lambda x.M : A_1 \to A_2$	by tx_lam (i)

21 03/28/19

21.1 Recap of Modal Logic

Explicit $\Delta; \Gamma \vdash E : A$

$E := \ldots \mid \mathit{box}\ E \mid \mathit{let\ box}\ u = E_1\ \mathit{in}\ E_2$

Δ contains assumptions, A valid

Γ contains assumptions, A true

Implicit Called implicit because we don't have let box.

$\underbrace{\Gamma_1; \ldots; \Gamma_n}_{\text{context stack}} \vdash M : A$

$M := \ldots \mid \mathit{box}\ M \mid \mathit{unbox}_n\ M$

$$\frac{\Sigma; \Gamma; \cdot \vdash M : A}{\Sigma; \Gamma \vdash \mathit{box}\ M : \Box A}$$

$$\frac{\Sigma; \Gamma \vdash M : \Box A}{\Sigma; \Gamma; \Gamma_1; \ldots; \Gamma_n \vdash \mathit{unbox}_n : A}$$

Where Σ is the context stack.

Example:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma_0 \vdash \Box(A \supset B)}}{\Gamma_0; \cdot \vdash A \supset B}\Box\,\mathrm{E}_{n=1} \quad \dfrac{}{\Gamma_0; \cdot \vdash A}\Box\,\mathrm{E}_{n=1}}{(\Box(A \supset B), \Box A); \cdot \vdash B}\supset \mathrm{E}}{\underbrace{(\Box(A \supset B), \Box A)}_{\Gamma_0} \vdash B}\Box\,\mathrm{I}$$

Note that the empty context is still a context, so unboxing the empty context still requires $n = 1$.

Proof term:

- Implicit: $\lambda x : \Box(A \supset B).\lambda y : \Box A.box((unbox_1\ x)(unbox_1\ y))$
- Explicit: $\lambda x : \Box(A \supset B).\lambda y : \Box A.let\ box\ u = x\ in\ let\ box\ v = y\ in\ box(uv)$

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\Box A \vdash \Box A}}{\Box A; \cdot; \cdot \vdash A}\Box\ \mathrm{E}_{n=2}}{\Box A; \cdot \vdash \Box A}\Box\ \mathrm{I}}{\Box A \vdash \Box\Box A}\Box\ \mathrm{I}$$

- Implicit: $\lambda x : \Box A.box(box\ (unbox_2\ x))$
- Explicit: $\lambda x : \Box A.let\ box\ u = x\ in\ box$

Proof for the explicit proof term:

$$\dfrac{\dfrac{}{\cdot; \Box A \vdash \Box A} \qquad \dfrac{\dfrac{}{A\ \text{valid}; \cdot \vdash A}}{A\ \text{valid}; \Box A \vdash \Box A}\Box\ \mathrm{I}}{\cdot; \Box A \vdash \Box\Box A}\Box\ \mathrm{E}$$

The goal was to show that the implicit and explicit systems are equivalent. In order to do so, we introduced the idea of an environment, that allows us to lookup what we need to unbox. We build and keep track of what these u variables are bound to.

Environment $\rho := \cdot \mid \rho_1\ box\ u = E$

Environment Stack $R := \cdot \mid R; \rho$
$\Sigma; \Gamma \models R : \Delta$
$\Delta; \Gamma \models \rho : \Delta'$

$$\dfrac{\Delta; \Gamma \models \rho : \Delta' \qquad \Delta; \Gamma \vdash E : \Box A}{\Delta; \Gamma \models \underbrace{\rho, box\ u = E}_{\substack{let\ box\ u_1 = E_1\ in \\ let\ box\ u_2 = E_2\ in}} : \underbrace{\Delta', u : A}_{\text{valid assumption}}}$$

$$\vdots$$

$$\dfrac{}{\Delta; \Gamma \models \cdot : \cdot}$$

Environment Stacks:

$$\dfrac{\Sigma \vdash R : \Delta_1 \qquad \Delta_1; \Gamma \vdash \rho : \Delta_2}{\Sigma; \Gamma \vdash R : \rho : (\Delta_1, \Delta_2)}$$

We can carve out our full list of global assumptions Δ into these sections ρ for a context stack. Recall that in the explicit language we kind of "forget" how many boxes we have to traverse, as long as its valid we have access to it.

Theorem: If $\Sigma; \Gamma \models R; \rho : \Delta$, $\Delta; \Gamma \vdash E : A$ and $(R; \rho) \triangleright E \to M$ (Translation) then $\Sigma; \Gamma \vdash M : A$

Case: (Variable $x : A \in \Gamma$)

$(R; \rho) \triangleright x \to x$

$\Delta; \Gamma \vdash_e x : A$ by assumption

$x : A \in \Gamma$ by assumption

$\Sigma, \Gamma \vdash_i x : A$

Case: Functions

$$\frac{(R; \rho) \triangleright E \to M}{(R; \rho) \triangleright \lambda x.E \to \lambda x.M}$$

$\Delta; \Gamma \vdash \lambda x.E : A$ by assumption

$\Delta; \Gamma, x : B_1 \vdash E : B_2$ by inversion

$A = B_1 \supset B_2$

$\Sigma; \Gamma \models (R; \rho) : \Delta$ by assumption

$\Sigma; (\Gamma, x; B_1) \models (R; \rho) : \Delta$ by weakening contexts in a context stack

$\Sigma; (\Gamma, x : B_1) \vdash M : B_2$ by IH

$\Sigma; \Gamma \vdash \lambda x.M : B_1 \supset B_2$ by $\supset$ I

Case: Box

$$\frac{R; \rho; \cdot \triangleright E \to M}{(R; \rho) \triangleright box\ E \to box\ M}$$

$\Delta; \Gamma \vdash box\ E : \Box A$ by assumption

$\Delta; \cdot \vdash E : A$ by inversion on $\Box$ I

$\Sigma; \Gamma \models R; \rho : \Delta$ by assumption

$(\Sigma; \Gamma); \cdot \models R; \rho; \cdot : \Delta$ by $\Delta; \cdot \Gamma \cdot : \cdot$ and above line (see proof tree below)

(note the RHS gives us $(\Delta, \cdot)$, but that's the same as Δ)

$(\Sigma; \Gamma); \cdot \vdash M : A$ by IH

$\Sigma; \Gamma \vdash box\ M : \Box A$ by $\Box$ I (implicit)

$$\frac{\Sigma' \vdash R' : \Delta_1 \qquad \Delta_1; \Gamma \vdash \rho : \Delta_2}{\Sigma'; \Gamma \vdash R' : \rho : (\Delta_1, \Delta_2)}$$

Case: Let-Box

$$\frac{R; (\rho, box\ u = E_1) \triangleright E_2 \to M}{(R; \rho) \triangleright let\ box\ u = E_1\ in E_2 \to M}$$

$\Delta; \Gamma \vdash E_1 : \Box B$	by inversion on $\Box E$
$(\Delta, u : B\ valid); \Gamma \vdash E_2 : A$	by inversion on $\Box$ E
$\Sigma; \Gamma \models R; \rho : \Delta$	by assumption
$\Sigma \models R : \Delta_1$	by typing def on env. stacks
$\Delta_1; \Gamma \vdash \rho : \Delta_2,\ \Delta = \Delta_1, \Delta_2$	by typing def of env. stacks
$(\underbrace{\Delta_1, \Delta_2}_{\Delta}); \Gamma \vdash E_1 : \Box B$	from above
$\Delta_1; \Gamma \vdash \rho, box\ u = E_1 : (\Delta_2, u : B\ valid)$	
$\Sigma; \Gamma \models; (\rho, box\ u = E_1) : \Delta, u : B\ valid$	by typing def of env. stacks
$\Sigma; \Gamma \vdash M : A$	by IH

Case: u (from Δ)

$$\frac{(R; \rho'_n) \triangleright E \to M}{(R; \underbrace{\rho_n}_{\rho'_n, box\ u = E, \rho''_n}; \ldots; \rho_1) \triangleright u \to unbox_n\ M}$$

$\underbrace{\Sigma}_{\Sigma_0; \Gamma_n; \ldots; \Gamma_1}; \Gamma \models R; \rho_n; \ldots; \rho_1 : \underbrace{\Delta_0, \Delta_n \ldots, \Delta_1}_{\Delta}$	by assumption
$\Delta_0 \vdash \rho_n : \Delta_n$ and $\Delta_n = \Delta'_n, u : A\ valid, \Delta'_n$	typing def on env. stacks n times
$\Delta_0, \Delta'_n; \Gamma_n \vdash E : \Box A$	by previous rules
$\Sigma; \Gamma_n \vdash M : \Box A$	by IH
$\Sigma; \Gamma_n; \ldots; \Gamma_1; \Gamma \vdash unbox_n\ M : A$	by typing rule for $\Box$ E$_n$

22 04/02/19

22.1 Modality: Possibility

A true	is a proposition A true? (is there a proof for A true?)
A valid	A is true in every possible world
	$\to$ we cannot presuppose any knowledge about A
	$\to$ Intervalize validity using $\Box$A
	A is necessarily true
A poss	A is true in some world
	$\to$ A is possibly true
	$\to$ Internalizing possibility using $\diamond A$
	$\to$ How can we use the knowledge that A is possible?

Certain things to keep in mind:

1. If A is true, then A is possible.

2. If A is possible and assuming A is true C is possible, then C is possible. We can encapsulate the logic of A being possible by further getting something else possible.

$$\underbrace{\Delta}_{\substack{\text{global}\\ A \text{ valid}}}; \underbrace{\Gamma}_{\substack{\text{local}\\ A \text{ true}}} \vdash A \text{ true}$$

Intro for Diamond:

$$\frac{\Delta; \Gamma \vdash A \text{ poss}}{\Delta; \Gamma \vdash \diamond A \text{ true}} \diamond \text{I}$$

$$\frac{\Delta; \Gamma \vdash A \text{ true}}{\Delta; \Gamma \vdash A \text{ poss}}$$

This is a silent rule, we don't need to use this because A is true means A is possible, however we can use it to be explicit.

Elim for Diamond:

$$\frac{\Delta; \Gamma \vdash \diamond A \text{ true} \qquad \Delta; A \text{ true} \vdash C \text{ poss}}{\Delta; \Gamma \vdash C \text{ poss}}$$

Examples $\Box(A \supset B) \supset \diamond A \supset \diamond B$ true

$$\dfrac{\dfrac{}{(A \supset B) \text{ valid}; \diamond A \text{ true} \vdash \diamond A \text{ true}} \qquad \dfrac{\dfrac{\dfrac{}{(A \supset B) \text{ valid}; A \text{ true} \vdash A \supset B \text{ true}} \qquad \dfrac{}{\ldots; A \text{ true} \vdash A \text{ true}}}{(A \supset B) \text{ valid}; A \text{ true} \vdash B \text{ true}}}{(A \supset B) \text{ valid}; A \text{ true} \vdash B \text{ poss}}}{(A \supset B) \text{ valid}; \diamond A \text{ true} \vdash B \text{ poss}} \diamond \text{E}$$

$$\frac{(A \supset B) \text{ valid}; \diamond A \text{ true} \vdash B \text{ poss}}{(A \supset B) \text{ valid}; \diamond A \text{ true} \vdash \diamond B} \diamond \text{I}$$

Introducing proof terms Proof Terms $M := x \mid \lambda x.M \mid M\ N \mid \langle M, N \rangle \mid \ldots \mid box(M) \mid$ *let box* $u = M$ *in* $N \mid$ *dia* E

Proof Expressions $E := M \mid$ *let dia* $x = M$ *in* E

$$\frac{\Delta; \Gamma \vdash E : A \text{ poss}}{\Delta; \Gamma \vdash \mathit{dia}\ E : \diamond A \text{ true}} \diamond \text{I}$$

$$\frac{\Delta; \Gamma \vdash M : \diamond A \text{ true} \qquad \Delta; A \text{ true} \vdash E : C \text{ poss}}{\Delta; \Gamma \vdash \mathit{let\ dia}\ x = M\ \mathit{in}\ E : C \text{ poss}}$$

$$\dfrac{\dfrac{\dfrac{}{\cdot; A \text{ true} \vdash A \text{ poss}}\ \text{hyp}^*}{\cdot; A \text{ true} \vdash \diamond A \text{ true}} \diamond \text{I}}{\cdot; \cdot \vdash A \supset \diamond A \text{ true}} \supset \text{I}$$

Proof term for this is $\lambda x : A.\mathit{dia}\ x$

Proof term for previous example is *let box* $u = x$ *in* $\lambda x : \Box(A \supset B).\lambda y : \diamond A.\mathit{dia}(\mathit{let\ dia}\ y' = y\ \mathit{in}\ u\ y')$

Local Soundness

$$\dfrac{\dfrac{\mathcal{D}}{\Delta;\Gamma \vdash A \text{ poss}} \Diamond \text{I} \qquad \mathcal{E}}{\Delta;\Gamma \vdash C \text{ poss}} \quad \Diamond;A \text{ true} \vdash C \text{ poss} \quad \Longrightarrow \quad \begin{array}{c}\mathcal{E}' \\ \Delta;\Gamma \vdash C \text{ poss} \\ \mathcal{E}' \text{ comes from 2.}\end{array}$$

(Note that 2. says: If A is possible and assuming A is true, C is possible, then C is possible)

This gives us another substitution lemma. If $\Delta;\Gamma \vdash E : A$ poss and $\Delta; x : A \text{ true} \vdash F : C$ poss then $\Delta, \Gamma \vdash \langle E/x \rangle F$ poss

With the reduction rule: $\mathit{let\ dia}\ x = \mathit{dia}\ E\ \mathit{in}\ N \implies \langle E/x \rangle N$.

What is the substitution here? Is it the usual substitution? Recall that $[N/x]M$ and $[\![N/u]\!]M$ were defined inductively on M, since that's also what allowed us to prove:

- If $\Delta;\Gamma \vdash N : A$ true and $\Delta;\gamma, x : A \text{ true} \vdash M : C$ true then $\Delta, \Gamma \vdash [N/x]M : C$ true.
- If $\Delta;\cdot \vdash A$ true and $\Delta, u : A$ valid $;\Gamma \vdash M : C$ true then $\Delta, \Gamma \vdash [\![N/u]\!]M : C$ true

We went from the proof of M to its leaves and then we plugged in N.

But the new substitution rule tries to modify things that were only introduced in the diamond operator (working on the E instead of N), which is why we use a new symbol to indicate this type of substitution.

Expr. $E := M \mid \mathit{let\ dia}\ x = M\ \mathit{in}\ E$
$\langle N/x \rangle F = [N/x]F$ (if it is in M, we fallback to the usual operation)
$\langle \mathit{let\ dia}\ y = N\ \mathit{in} E/x \rangle F = \mathit{let\ dia}\ y = N\ \mathit{in} \langle E/x \rangle F$ (we pull out the let, rearranging structure)

To prove the substitution lemma, we'd do induction on the first judgment/derivation.

Case $E = M$

$\Delta;\Gamma \vdash M$ poss	by assumption
$\Delta;\Gamma \vdash M$ true	(that's by the case we are in, i.e. inversion not on let but silent rule)
$\Delta; x : A \text{ true} \vdash F : C$ poss	
$\Delta;\Gamma \vdash [M/x]F : C$ poss	(reg. subst)

23 04/04/19

What we've seen so far for $S4$:

A true	A is true "here" (locally)
A valid, $\Box A$	A true holds in <u>every</u> accessible world
A pos, $\Diamond A$	A true holds in <u>some</u> accessible world

Computationally, we can say:

$\Box A$ mobile code of type A

$\diamond A$ remote 'A'

Proof terms $M := x \mid \lambda x.M \mid M\ N \mid box\ M \mid let\ box\ u = M\ in\ N \mid dia\ E$

Proof Expression $E, F := M \mid let\ dia\ x = M\ in\ E$

Box

$$\frac{\Delta; \Gamma \vdash M : \Box A \qquad \Delta, u :: A\ \text{valid} \vdash N : C}{\Delta, \Gamma \vdash let\ box\ u = M : C\ in\ N}$$

$$\frac{\Delta; \cdot \vdash M : A}{\Delta; \Gamma \vdash box\ M : \Box A}$$

$$\frac{u :: A\ \text{valid} \in \Delta}{\Delta; \Gamma \vdash u : A}$$

Diamond

$$\frac{\Delta; \Gamma \vdash M : \diamond A \qquad \Delta; x : A \vdash E \div C\ \text{poss}}{\Delta; \Gamma \vdash let\ dia\ x = M\ in\ E \div C\ \text{poss}}$$

$$\frac{\Delta; \Gamma \vdash E \div A\ \text{poss}}{\Delta; \Gamma \vdash dia\ E : \diamond A}$$

$$\frac{\Delta; \Gamma \vdash M : A}{\Delta; \Gamma \vdash M \div A\ \text{poss}}$$

Where $\div$ is equivalent to : but used to put emphasis on the fact that A poss (used in the paper), but also redundant as we can just use $\div$ instead of writing poss.

Is $(A \supset B) \supset \diamond A \supset \diamond B$ true?

Option 1, proving from B true:

$$\dfrac{\dfrac{\dfrac{}{\cdot; x : A \supset B, y : \diamond A \vdash A \supset B\ \text{true}} \qquad \dfrac{\text{? Stuck because we cannot use } \diamond \text{ E}}{\cdot; x : A \supset B, y : \diamond A \vdash A\ \text{true}}}{\cdot; x : A \supset B, y : \diamond A \vdash B\ \text{true}}}{\dfrac{\cdot; x : A \supset B, y : \diamond A \vdash B\ \text{poss}}{\cdot; x : A \supset B, y : \diamond A \vdash \diamond B\ \text{true}}}\ \text{coercion poss-true}$$

Note that if we had no distinction between truth and possibility, we couldn't be stuck here and we'd be able to prove something that isn't true.

The following is "ill-typed" (not even syntactically legal)

$\lambda x : A \supset B.\lambda y : \diamond A.dia(\underline{x(let\ dia\ w = y\ in\ w)})$

Would be legal with no distinction between truth and possibility.

Option 2:

$$\dfrac{\dfrac{\dfrac{}{\cdot; x : A \supset B, y : \diamond A \vdash \diamond A\ \text{true}} \qquad \dfrac{\text{? Stuck, } A \supset B \text{ only true locally, lost information}}{\cdot; w : A \vdash B\ \text{poss}}}{\cdot; x : A \supset B, y : \diamond A \vdash B\ \text{poss}}}{\cdot; x : A \supset B, y : \diamond A \vdash \diamond B\ \text{true}}$$

Another Example

$$\dfrac{\dfrac{\dfrac{\dfrac{\overline{\cdot; x : \diamond\Box A \vdash \diamond\Box A \text{ true}} \quad \overline{\cdot; y : \Box A \vdash A \text{ poss}}}{\cdot; x : \diamond\Box A \vdash A \text{ poss}}\diamond \text{E}}{\cdot; x : \diamond\Box A \vdash \diamond A \text{ true}}}{\cdot; \cdot \vdash \diamond\Box A \supset \diamond A \text{ true}}$$

Alternatively, to get A poss:

$$\dfrac{\Delta; \Gamma \vdash M : \Box A \qquad \Delta, u :: A \text{ valid}; \Gamma \vdash N \div C \text{ poss}}{\Delta; \Gamma \vdash \mathit{let\ box}\ u = M\ \mathit{in}\ N \div C \text{ poss}}$$

This rule is not required in the proof above, but it could become necessary if we have nested $\Box$ and $\diamond$.

$\Box A \supset \diamond\Box A$ is true.

$\diamond(A \wedge B) \supset \diamond A \wedge \diamond B$ is true.

$\diamond A \wedge \diamond B \supset \diamond(A \wedge B)$ is not true, the diamonds on the LHS can be for different worlds.

$\diamond\Box A \supset \Box A$ is not true, we do not have symmetry.

$$\dfrac{\dfrac{\overline{\cdot; x : \diamond(A \wedge B) \vdash \diamond(A \wedge B) \text{ true}} \quad \dfrac{\dfrac{\overline{\cdot; y : A \wedge B \vdash A \wedge B \text{ true}}}{\cdot; y : A \wedge B \vdash A \text{ true}}}{\cdot; y : A\wedge \vdash A \text{ poss}}\wedge \text{E}_1}{\cdot; x : \diamond(A \wedge B) \vdash A \text{ poss}}}{\cdot; x : \diamond(A \wedge B) \vdash \diamond A \text{ true}}$$

Where we commit to a world and show that we can get A from it.

Going back to substitution and reductions

$\mathit{let\ box}\ u = \mathit{box}\ M\ \mathit{in}\ N \implies [\![M/x]\!]$

$\mathit{let\ dia}\ x = \mathit{dia}\ E\ \mathit{in}\ F \implies \langle E/x\rangle F$

$(\lambda x.N)M \implies [M/x]N$

$\langle M/x\rangle F = [M/x]F$

$\langle \mathit{let\ dia}\ y = M\ \mathit{in}\ E/x\rangle F = \mathit{let\ dia}\ y = M\ \mathit{in}\langle E/x\rangle F$

If $\Delta; \Gamma \vdash E \div A$ poss and $\Delta; x : A \text{ true} \vdash F \div C$ poss then $\Delta; \Gamma \vdash \langle E/x\rangle F \div C$ poss.

If $\Delta; \Gamma \vdash E : A$ true and $\Delta; x : A \text{ true} \vdash F \div C$ poss then $\Delta; \Gamma \vdash [M/x]F \div C$ poss.

Case $\mathcal{D} = \dfrac{\begin{array}{c}\mathcal{D}' \\ \Delta; \Gamma \vdash M : A \text{ true}\end{array}}{\Delta; \Gamma \vdash M \div A \text{ poss}}$

$\Delta; x : A \text{ true} \vdash F \div C$ poss — by ass

$\Delta; \Gamma \vdash [M/x]F \div C$ poss — by IH (2)

$\Delta; \Gamma \vdash \langle M/x\rangle F \div C$ poss — by def of $\langle\cdot\rangle$

Case $\mathcal{D} = \dfrac{\begin{array}{cc}\mathcal{D}_1 & \mathcal{D}_2 \\ \Delta;\Gamma \vdash M : \diamond B & \Delta; x : B \text{ true} \vdash E \div A \text{ poss}\end{array}}{\Delta;\Gamma \vdash \mathit{let\ dia}\ y = M\ \mathit{in}\ E \div A \text{ poss}} \diamond \text{E}$

$\Delta; \underline{x : A} \text{ true} \vdash F \div C \text{ poss}$ by ass

$\Delta; y : B \text{ true} \vdash \langle E/x \rangle F \div C \text{ poss}$ by IH(1)

$\Delta;\Gamma \vdash \mathit{let\ dia}\ y = M\ \mathit{in}\ \langle E/x \rangle F \div C \text{ poss}$ by $\diamond$ E

$\Delta;\Gamma \vdash \langle \mathit{let\ dia}\ y = M\ \mathit{in}\ E/x \rangle F \div C \text{ poss}$ by def $\langle \cdot \rangle$

Case $\dfrac{\mathcal{D}'}{\Delta; x : A \text{ true} \vdash N : C \text{ true}}{\Delta; x : A \text{ true} \vdash N \div C \text{ poss}}$

$\Delta;\Gamma \vdash M : A \text{ true}$ by ass

$\Delta;\Gamma \vdash [M/x]N : A \text{ true}$ by substitution lemma for A true

Case $\mathcal{D} = \dfrac{\begin{array}{cc}\mathcal{D}_1 & \mathcal{D}_2 \\ \Delta; x : A \vdash N : \diamond B & \Delta; y : B \text{ true} \vdash E \div C \text{ poss}\end{array}}{\Delta; x : A \vdash \mathit{let\ dia}\ y = N\ \mathit{in}\ E \div C \text{ poss}} \diamond \text{E}$

$\Delta;\Gamma \vdash M : A \text{ true}$ by ass

$\Delta;\Gamma \vdash [M/x]N : \diamond B \text{ true}$ by substitution lemma for (A true)

$\Delta;\Gamma \vdash \mathit{let\ dia}\ y = [M/x]N\ \mathit{in}\ E \div C \text{ poss}$ by $\diamond$ E

$\Delta;\Gamma \vdash [M/x](\mathit{let\ dia}\ y = N\ \mathit{in}\ E) \div C \text{ poss}$ by def of $[\cdot]$

The modality we've seen is equivalent to lax modality: $\diamond\Box A = \circ A$

24 04/09/19

Earlier throughout the term, all of our assumptions had the same structure, they shared structural properties. Today we will see a different structure.

24.1 Linear Logic

Substructural Logic Let's assume we want to model the following:

1. If one has one dollar, the none can buy a coke. $D \supset C$
2. If one has one dollar, then one can buy a sandwich. $D \supset S$

$$\dfrac{\dfrac{\dfrac{\overline{\Gamma \vdash D \supset C \text{ true}} \quad \overline{\Gamma \vdash D \text{ true}}}{\Gamma = D \supset C, D \supset S, \mathbf{D} \vdash C \text{ true}} \supset \text{E} \quad \dfrac{\overline{\Gamma \vdash D \supset S \text{ true}} \quad \overline{\Gamma \vdash D \text{ true}}}{\Gamma \vdash S \text{ true}} \supset \text{E}}{D \supset C, \mathbf{D} \supset S, D \vdash C \wedge S \text{ true}} \wedge \text{I}}{D \supset C, D \supset S \vdash D \supset (C \wedge S) \text{ true}} \supset \text{I}$$

This proof is correct in our system, but it doesn't make sense in the real world, if you have one dollar you can't buy a coke **and** a sandwich. We could fix this by counting how many dollars we have or spending the money, however this will overcomplicate our logic. Notice that the assumption D is used in both the left and right branches of the $\wedge$ I.

Linear Logic Every assumption is used exactly once.
We don't have weakening, we have to use all our dollars, can't keep any.

Affine Logic Every assumption is used at most once.

Relevant Logic Every assumption is used at least once.

Ordered Logic Every assumption has to be used exactly once in the order it was introduced.

	exchange	weakening	contraction
Linear Logic	✓	×	×
Affine Logic	✓	✓	×
Relevant Logic	✓	×	✓
Order Logic	×	×	×

$$\underbrace{\Delta}_{\substack{\text{List of resources= assumptions that can be used exactly once}}} \vdash A \text{ true}$$

Substitution principle If $\Delta \vdash A$ true and $\Delta', A \text{ true} \vdash C$ true then $\Delta, \Delta' \vdash C$ true

Simultaneous Conjunction $A \otimes B$
"Given some resources we can achieve both A and B at the same time."

$$\frac{\Delta_1 \vdash A \text{ true} \qquad \Delta_2 \vdash B \text{ true}}{\Delta_1, \Delta_2 \vdash A \otimes B \text{ true}}$$

$$\frac{\Delta_1 \vdash A \otimes B \text{ true} \qquad \Delta_2, A \text{ true}, B \text{ true} \vdash C \text{ true}}{\Delta_1, \Delta_2 \vdash C \text{ true}}$$

Very similar to the alternative conjunction rule we've seen many times, although in this case it is literally a different conjunction rule, we can get both A and B at the same time.

Alternative Conjunction Internal choice: With the set of resources we can achieve A and B - but not at the same time! This sounds like a disjunction, but a disjunction is an external choice, i.e. the cashier chooses whether we get a coke or a sandwich, not us. In this case we get to choose.

$$\frac{\Delta \vdash A \text{ true} \qquad \Delta \vdash B \text{ true}}{\Delta \vdash A \& B \text{ true}}$$

$$\frac{\Delta \vdash A \& B \text{ true}}{\Delta \vdash A \text{ true}}$$

$$\frac{\Delta \vdash A \& B \text{ true}}{\Delta \vdash B \text{ true}}$$

Implication $A \multimap B$

$$\frac{\Delta, A \text{ true} \vdash B \text{ true}}{\Delta \vdash A \multimap B \text{ true}}$$

$$\frac{\Delta_1 \vdash A \multimap B \text{ true} \qquad \Delta_2 \vdash A \text{ true}}{\Delta_1, \Delta_2 \vdash B \text{ true}}$$

External Choice $A \oplus B$

$$\frac{\Delta \vdash A \text{ true}}{\Delta \vdash A \oplus B \text{ true}}$$

$$\frac{\Delta \vdash B \text{ true}}{\Delta \vdash A \oplus B \text{ true}}$$

$$\frac{\Delta' \vdash A \oplus B \text{ true} \qquad \Delta, A \text{ true} \vdash C \text{ true} \qquad \Delta, B \text{ true} \vdash C \text{ true}}{\Delta, \Delta' \vdash C \text{ true}}$$

So $D \supset C$ and $D \supset S$ become $D \multimap C$ and $D \multimap S$. But say we want to buy multiple cokes, will we just copy the proposition multiple times? No. So now we want to distinguish between truth and validity like we have before.

$$\overbrace{\Gamma}^{\text{unrestricted}} \; ; \; \underbrace{\Delta}_{\substack{\text{List of resources= assumptions that can be used exactly once}}} \vdash A \text{ true}$$

To accommodate unrestricted (intuitionistic) assumptions, we want to distinguish between
A valid - unrestricted
A true - resource

Internalize the fact that A is valid
$\Longrightarrow$ $!A$ ("of course")

$$\frac{\Gamma; \cdot \vdash A \text{ true}}{\Gamma; \cdot \vdash !A \text{ true}}$$

Since we don't have weakening, we can't allow additional resources on the LHS. This is a refinement of the box modality that we've seen.

$$\frac{\Gamma; \Delta \vdash !A \text{ true} \qquad \Gamma, A \text{ valid}; \Delta' \vdash C \text{ true}}{\Gamma; (\Delta, \Delta') \vdash C \text{ true}}$$

Menu Formalizing a restaurant menu: $CAD(60)$

$\multimap$ (Soup & Salad) "internal choice", you can choose one of these

$\otimes$ Salmon

$\otimes$ (Carrots $\oplus$ Asparagus) "external choice", this is a seasonal item, you have no choice as to whether carrots or asparagus are available

$\otimes((CAD(5) \multimap Mousse)\&1)$

$\otimes$ coffee $\otimes$ $\underbrace{!(\text{Coffee})}_{\text{unlimited refills}}$

Top: Consume all resources

$$\frac{}{\Delta \vdash \mathrm{T} \text{ true}}$$

Useful because sometimes we might not use up resources, i.e. leftover money after buying meal from menu, flush additional resources.

Unit: Trivial Goal which requires <u>no</u> resources

$$\frac{}{\cdot \vdash 1 \text{ true}}$$

$$\frac{\Delta \vdash 1 \text{ true} \qquad \Delta' \vdash C \text{ true}}{\Delta, \Delta' \vdash C \text{ true}}$$

Useful for when things are optional, i.e. you can buy a dessert or skip it.

Example

$$\frac{\dfrac{\dfrac{}{\varphi; \cdot \vdash D \multimap C \text{ true}} \quad \dfrac{}{\varphi; D \text{ true} \vdash D \text{ true}}}{D \multimap C \text{ valid}, D \multimap S \text{ valid}; D \textit{ true} \vdash C \text{ true}} \quad \dfrac{\dfrac{}{\varphi; \cdot \vdash D \multimap S \text{ true}} \quad \dfrac{}{\varphi; D \text{ true} \vdash D \text{ true}}}{D \multimap C \text{ valid}, D \multimap S \text{ valid}; D \textit{ true} \vdash S \text{ true}}}{D \multimap C \text{ valid}, D \multimap S \text{ valid}; \underbrace{D, D}_{D \textit{ true}, D \textit{ true}} \vdash C \otimes S \text{ true}}$$

Two different formulations, $D \multimap D \multimap C \otimes S$

$D \otimes D \multimap C \otimes S$