Introduction to Logical Foundations

Prof. Brigitte Pientka @ OPLSS25
Notes by: Yanning Chen

Contents

0.1 Judgment

Analytic judgments are those that become evident by conceptual analysis.

$$\frac{J_1 \quad \cdots \quad J_n}{J_c}$$

1. **Syntax**: well-formed

   E.g.

   $$\frac{}{T \ \texttt{wf}} \qquad \frac{A \ \texttt{wf} \quad B \ \texttt{wf}}{A \circ B \ \texttt{wf}}$$

2. **Semantics**: true
   1. **I**: *introduce* a connective
   2. **E**: *extract* info out of a pure connective

   E.g.

   $$\frac{A \quad B}{A \wedge B} \wedge I \qquad \frac{A \wedge B}{A} \wedge E_1$$

0.2 Hypothetical Reasoning

Given P, Q holds.

1. $\Rightarrow I$: Internalizing a reasoning process.

E.g.

$$\cfrac{\cfrac{\cfrac{P_1 \qquad P_2}{\cdots}}{Q}}{P_1, P_2 \Rightarrow Q}$$

Note that Ps and Qs cease to exist in the conclusion, thus “internalized”.

1. $\Rightarrow E$: modus ponens

0.3 Properties of assumtions (structural)

Weakening: assumptions can be unused

$$\cfrac{\cfrac{\cfrac{}{A}u \qquad (\text{unused}) \cfrac{}{B}w}{B \Rightarrow A} \Rightarrow I^w}{A \Rightarrow B \Rightarrow A} \Rightarrow I^u$$

Contraction: assumptions can be used multiple times

Substitution: TODO

0.4 Context

Notice the two dimensional rule of $\Rightarrow I$:

$$\cfrac{\cfrac{\cfrac{P_1 \qquad P_2}{\cdots}}{Q}}{P_1, P_2 \Rightarrow Q}$$

It’s kind of awkward to keep this hypothetical strcture around. Instead, we write

$$\frac{\Gamma, h_1 : P_1, h_2 : P_2 \vdash Q}{\Gamma \vdash P_1, P_2 \Rightarrow Q}$$

Note that we declare

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

By following the same principle, we can rewrite **Weakening** and **Contraction**.

Weakening:

$$\frac{\Gamma \vdash B}{\Gamma, x : A \vdash B}$$

Contraction:

$$\frac{\Gamma, x : A \vdash B}{\Gamma, x : A, y : A \vdash B}$$

Substitution:

$$\frac{\Gamma, x : A \vdash C \qquad \Gamma \vdash A}{\Gamma \vdash C}$$

0.5 Local Soundness/Completeness

Local in the sense that these properties only discuss **a single connective**, not the whole system. It's a weak witness that this system makes sense.

1. **Local Soundness**: the combination of intro and elim is not too strong, i.e. they don't allow us to infer more than what's already known.

   E.g. the proof

   $$\dfrac{\dfrac{\Gamma \vdash A \qquad \Gamma \vdash B}{\Gamma \vdash A \times B} \times I}{\Gamma \vdash A} \times E_1$$

   collapses to

   $$\dfrac{\Gamma \vdash A}{\Gamma \vdash A}$$

   so there's no additional information provided by the intro and elim rules.

2. **Local Completeness**: the combination of intro and elim is sufficiently strong, in terms of rebuilding the info we have.

   E.g. there's no information loss in the rebuild process of the connective $A \times B$

   $$\dfrac{\dfrac{\Gamma \vdash A \times B}{\Gamma \vdash A} \times E_1 \qquad \dfrac{\Gamma \vdash A \times B}{\Gamma \vdash B} \times E_2}{\Gamma \vdash A \times B} \times I$$

0.6 C.H.

- Propositions - Types
- Proof - Programs
- Nat. Ded. - λ-calculus

Importance:

1. Guiding program language design
2. Basis of Type Theory
3. Proving logic consistency by looking at programs

0.6.1 Typing Judgment

$$\boxed{\Gamma \vdash M : A}$$

Given some assumptions in the context Γ,

- M is a proof term corresponding to the proposition A.
- M is a program that has a type A.

C.H.: $\Gamma \vdash A$ iff $\Gamma \vdash M : A$

Example: conjunction

$$\dfrac{\Gamma \vdash M : A \qquad \Gamma \vdash N : B}{\Gamma \vdash \langle M, N \rangle : A \wedge B} \wedge I \qquad \dfrac{\Gamma \vdash M : A \wedge B}{\Gamma \vdash M.1 : A} \qquad \dfrac{\Gamma \vdash M : A \wedge B}{\Gamma \vdash M.2 : B}$$

Example: implication

$$\dfrac{\Gamma, x : A \vdash \lambda x : A.M : B}{\Gamma \vdash M : A \to B} \Rightarrow I^x \qquad \dfrac{\Gamma \vdash M : A \to B \quad \Gamma \vdash N : A}{\Gamma \vdash MN : B} \Rightarrow E$$

Thm (Local Soundness)

$$\dfrac{\dfrac{\Gamma \vdash M : A \quad \Gamma \vdash N : B}{\Gamma \vdash \langle M, N\rangle : A \wedge B}}{\Gamma \vdash \langle M, N\rangle.1 : A} \quad \text{collapses to} \quad \dfrac{}{\Gamma \vdash M : A}$$

$$\dfrac{\dfrac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x : A.M : A \to B} \Rightarrow I^x \qquad \Gamma \vdash N : A}{\Gamma \vdash (\lambda x : A.M)N : B} \Rightarrow E \quad \text{collapses to} \quad \dfrac{}{\Gamma \vdash M[N/x] : B}$$

1. **Logic**: not gaining any information through this intro and elim detour.
2. **Program**: type is **preserved** through this detour.

The “collapse” can then be written as:
Reduction $\langle M, N\rangle.1 \Longrightarrow M$ and $(\lambda x : A.M)N \Longrightarrow M[N/x]$.

Thm (Subject Reduction)

If $\Gamma \vdash M : A$ and $M \Longrightarrow M'$, then $\Gamma \vdash M' : A$

Thm (Local Completeness)

$$\dfrac{}{\Gamma \vdash M : A \wedge B} \ \text{expands to} \quad \dfrac{\dfrac{\Gamma \vdash M : A \wedge B}{\Gamma \vdash M.1 : A} \wedge E_1 \qquad \dfrac{\Gamma \vdash M : A \wedge B}{\Gamma \vdash M.2 : B} \wedge E_2}{\Gamma \vdash \langle M.1, M.2\rangle : A \wedge B} \wedge I$$

$$\dfrac{}{\Gamma \vdash M : A \to B} \ (\eta\text{-})\text{expands to} \quad \dfrac{\dfrac{\dfrac{\Gamma \vdash M : A \to B}{\Gamma, x : A \vdash M : A \to B} \qquad \Gamma, x : A \vdash x : A}{\Gamma, x : A \vdash Mx : B} \Rightarrow E}{\Gamma \vdash \lambda x : A.Mx : A \to B} \Rightarrow I^x$$

This **exhibits the actual structure** (i.e. **expands its internal term/proof structure**) of M while preserving the type.

Insight:
- Proof Reduction - Program Reduction
- Normalizing Proofs - Normalizing Programs

Compare both proofs

TODO fill in the blanks

$$\dfrac{}{\vdash \lambda x : A \wedge A.(\lambda y : A.y)x.2 : A \wedge A \to A}$$

0.7 Disjunction

Let’s now expand the previous example to disjunction.

0.7.1 Rules

$$\frac{\Gamma \vdash M : A}{\Gamma \vdash \iota_1 M : A \lor B}\lor I_1$$

$$\frac{\Gamma \vdash M : B}{\Gamma \vdash \iota_2 M : A \lor B}\lor I_2 \qquad \frac{\Gamma \vdash M : A \lor B \quad \Gamma, x : A \vdash N_1 : C \quad \Gamma, x : B \vdash N_2 : C}{\Gamma \vdash \text{cases } M \text{ of } \iota_1 x \Rightarrow N_1 \mid \iota_2 x \Rightarrow N_2 : C}\lor E$$

0.8 Local Soundness

TODO Fill in the blanks

$$\frac{}{\Gamma \vdash \text{case}}$$

1 Modal Logic S4

> Truth is living in the moment - here and now.
>
> Validity is living forever and everywhere.
>
> — Brigitte Pientka

“The world is full of possibilities, but not today.”

1.1 Necessity Modality $\Box A$

> Note what I originally wrote as $\Gamma \vdash A$ is now $\Gamma \vdash A$ true to make a clear distinction between **validity** and **true**.

1.1.1 Validity

- If $\varepsilon \vdash A$ true then A valid
- If A valid then $\Gamma \vdash A$ true

Notice how the second rule allows weakening while the first rule does not. This means that **validity does not depend on any local assumptions**.

1.1.2 Judgment

$\Delta; \Gamma \vdash A$ true

- Δ - the **global** context. It contains valid assumptions that holds **forever**.
- Γ - the **local** context. It contains assumptions that hold **here and now**.

So, the definition of validity can be written as:

$$\frac{y : A \text{ true} \in \Gamma}{\Delta; \Gamma \vdash A \text{ true}} \qquad \frac{x : A \text{ valid} \in \Delta}{\Delta; \Gamma \vdash A \text{ true}}$$

Now we start to introduce the modality $\Box$:

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

Just as → is internalizing reasoning on implication, $\Box$ is internalizing reasoning on validity.

T law (*reflexivity*).

$$\frac{\cdot; x : \Box A \vdash A \text{ true}}{\cdot;\cdot \vdash \Box A \to A \text{ true}} \to I^x$$

i.e. If it's true everywhere and forever, then it's also true here and now.

A detour

One may be tempted to define $\Box_E$ as such

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

Let's see if it's locally complete as a sanity check.

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

should be able to expand to

$$\cfrac{\Delta;\Gamma \vdash \Box A \text{ true}}{\cfrac{\cfrac{\cdots}{\Delta;\cdot \vdash A \text{ true}}\ ???}{\Delta;\Gamma \vdash \Box A \text{ true}}\ \Box I}$$

Boom!

The correct solution

A `let`-style definition!

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

Let's see if it's locally sound.

$$\frac{\dfrac{\Delta;\cdot \vdash A \text{ true}}{\Delta;\Gamma \vdash \Box A \text{ true}}\ \Box I \qquad \Delta, u : A \text{ valid};\Gamma \vdash C \text{ true}}{\Delta;\Gamma \vdash C \text{ true}}\ \Box E^u$$

($\Delta;\cdot \vdash A$ true goes to $\Delta, u : A$ valid; $\Gamma \vdash C$ true via *modal substitution*)

collapses to

$\Delta;\Gamma \vdash C$ true

Lemma.

1. (*Substitution*) If $\Delta;\Gamma, x : A$ true $\vdash C$ true and $\Delta;\Gamma \vdash A$ true, then $\Delta;\Gamma \vdash C$ true.
2. (*Modal Substitution*) If $\Delta, y : A$ valid; $\Gamma \vdash C$ true and $\Delta;\cdot \vdash A$ true, then $\Delta;\Gamma \vdash C$ true.

Thm (*T*). $\Box$ satisfies *reflexivity*.

–

Thm (*Distributivity*). $\Box$ satisfies *distributivity*.

$$\Box(A \to B) \to \Box A \to \Box B$$

Thm (*4*). $\Box$ satisfies *transitivity*.

$$\Box A \to \Box\Box A$$

To introduce a $\Box$, one may attempt to use $\Box_I$, but this would leave us with an empty Γ which is not good. So, we need to first preserve x forever, which leads us to the use of $\Box_E$ - saving it to Δ.

NOTE We usually put something to eternality via $\Box_E$ rule. Note that we are gaining information via $\Box_E$ while we give up information via $\Box_I$.

$$\dfrac{\cdot; x : \Box A \text{ true} \vdash \Box A \text{ true} \qquad \dfrac{\dfrac{\dfrac{}{u : A \text{ valid}; \cdot \vdash A \text{ true}}\,u}{u : A \text{ valid}; \cdot \vdash \Box A \text{ true}}\,\Box I}{u : A \text{ valid}; x : \Box A \text{ true} \vdash \Box\Box A \text{ true}}\,\Box I}{\dfrac{\cdot; x : \Box A \text{ true} \vdash \Box\Box A \text{ true}}{\cdot; \cdot \vdash \Box A \to \Box\Box A \text{ true}}\,\to I^x}\,\Box E^u$$

1.2 Syntax

$$\text{Terms } M \boxed{?} x \mid \lambda x : A.M \mid M\ N \mid \langle M, N \rangle \mid M.1 \mid M.2 \mid u \mid \text{box } M \mid$$

NOTE u for valid assumptions.

Now let's add proof terms to logic rules.

$$\dfrac{\Delta; \cdot \vdash M : A \text{ true}}{\Delta; \Gamma \vdash \text{box } M : \Box A \text{ true}}\,\Box I \text{ (} M \text{ is a closed term w.r.t. } \textit{local} \text{ (} \textit{runtime} \text{) assumptions)}$$

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

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

And now we have

1. (*Substitution*) If $\Delta; \Gamma, x : A \text{ true} \vdash C$ true and $\Delta; \Gamma \vdash A$ true, then $\Delta; \Gamma \vdash C$ true. e.g.

$$(\text{box } N)[M/x] = \text{box } N$$

2. (*Modal Substitution*) If $\Delta, u : A \text{ valid}; \Gamma \vdash C$ true and $\Delta; \cdot \vdash A$ true, then $\Delta; \Gamma \vdash C$ true. e.g.

$$(\text{box } N)[\![M/u]\!] = \text{box } (N[\![M/u]\!])$$

Now let's try to rephrase *locally soundness*

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

collapses to

$\Delta; \Gamma \vdash N[\![M/u]\!] : C$

1.3 Real Programming Example

```haskell
nth: int -> (bool_vec -> bool)
```

This function does not do anything if you only pass it an integer. It just sits there, not producing anything meaningful.

How to avoid this situation and force it generate a real function?

```haskell
nth: int -> □(bool_vec -> bool)
nth 0 = box (fun v -> hd v)
nth (S n) =
  let box r = nth n in box (fun v -> r (tl v))
```

In this case, box makes sure the function generated does not depend on `int`.

Compare

```haskell
nth 1
= fun v -> tl (nth 0 v)
= fun v -> tl (hd v)
```

with

```haskell
nth 1
= let box r = nth 0 in box (fun v -> r (tl v))
= let box r = box (fun v -> hd v) in (fun v -> r (tl v))
= box (fun v -> (fun v0 -> hd v0) (tl v))
```

Notice how the returned function is not a closure over n.

However, if you compare these two functions you still find the latter one not satisfying cuz it's returning a redex and it's in a box so it get stuck.

Contextual types to the rescue!

1.4 Contextual types

Previously, we wrote $\square A$ to mean A starts with an empty context, which is not sufficient in many cases. So instead, let's allow specifying a context Γ for A.

1.4.1 Examples

Cooking metaphor

1. Add eggs, flour, sugar
2. Add $\boxed{}$ (a liquid)

To type $\boxed{}$, it's eggs, flour, sugar ⊩ liquid

Theorem prover

Holes in programs:

fun $x \to \boxed{} +_{\text{int}} x$, you can see the hole here accepts an $x : \text{int} \Vdash \text{int}$

Or $\lambda x.\lambda y.\boxed{}\, y.2 : (A \to B \to C) \to (A \times B) \to C$, where the hole accepts an $x : A \to B \to C, y : A \times B \Vdash B \to C$

1.4.2 Syntax

$$\begin{array}{rl} \text{Types} & A \boxed{?} \ldots \mid \square(\psi \Vdash A) \\ \text{Terms} & M \boxed{?} \ldots \mid \text{box}\ (\psi.M) \\ \text{Contexts} & \Gamma, \psi \boxed{?} \ldots \end{array}$$

E.g. box $(x : \text{int}.x + x) : \square(x : \text{int} \Vdash \text{int})$

NOTE But how to keep this thing stable under renaming?

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

$$\frac{\Delta; \Gamma \vdash M : \Box(\psi \Vdash A) \text{ true} \qquad \Delta, u : A \text{ valid}; \Gamma \vdash N : C \text{ true}}{\Delta; \Gamma \vdash \text{let box } u \coloneqq M \text{ in } N : C \text{ true}}\ \Box E$$

$$\frac{x : A \text{ true} \in \Gamma}{\Delta; \Gamma \vdash x : A \text{ true}} \qquad \frac{u : \psi \Vdash A \text{ valid} \in \Delta \qquad \Delta, \Gamma \vdash \sigma : \psi}{\Delta; \Gamma \vdash \text{clo}(u, \sigma) : A \text{ true}}$$

Notes

- σ - substitution from ψ to Δ, Γ i.e.

$$\frac{\Delta; \Gamma \vdash \sigma : \psi \qquad \Delta; \Gamma \vdash M : A}{\Delta; \Gamma \vdash (\sigma, M/x) : \psi, x : A}$$

- $\text{clo}(u, \sigma)$ - delayed substitution σ that can be applied once u is available.

NOTE **Computation rules for `clo`**

Recall how we have

$$(\text{box } N)[M/x] = \text{box } N$$

$$(\text{box } N)\llbracket M/u \rrbracket = \text{box } (N\llbracket M/u \rrbracket)$$

Now also,

$$\text{clo}(u, \sigma)\llbracket \psi.M/u \rrbracket = M[\sigma]$$

Beware that $M[\sigma]$ is a **local** substitution.

E.g.

$$\lambda x.\text{ let box } u \coloneqq x \text{ in box } (\lambda y.\lambda z.u\ y) : \Box(C \to A) \to \Box(C \to D \to A)$$

$$\lambda x.\text{ let box } u \coloneqq x \text{ in box}(y : C, z : D.\ \text{clo}(u, y/x')) : \Box(x' : C \Vdash A) \to \Box(y : C, z : D \Vdash A)$$

With this, we can revise our `nth` example

```haskell
nth: int -> □(bool_vec -> bool)
nth 0 = box (fun v -> hd v)
nth (S n) =
  let box r =
    nth n in box (fun v -> r (tl v))
```

into this

```haskell
nth: int -> □(v: bool_vec ⊨ int)
nth 0 = box (v: bool_vec. hd v)
nth (S n) =
  let box u = nth n in
    box (v: bool_vec. clo(u, (tl v)/v)
```

then we make

```haskell
nth 1
```

```
= let box r = nth 0 in box (fun v -> r (tl v))
= let box r = box (fun v -> hd v) in (fun v -> r (tl v))
= box (fun v -> (fun v0 -> hd v0) (tl v))
```

into this

```haskell
nth 1
= let box u = nth 0 in
    box (v: bool_vec. clo(u, (tl v)/v))
= let box u = box (v: bool_vec. hd v) in
    box (v: bool_vec. clo(u, (tl v)/v))
= box (v: bool_vec. clo(hd v0, (tl v)/v0))
= box (v: bool_vec. hd (tl v))
```

Notice how the nested evaluation is eager.

TODO What’s the difference between functions and `clo`?