Lecture 14

Linear Logic

Compiler Math

Rust

Official Rust logo

Paradigms	Multi-paradigm: concurrent, functional, generic, imperative, structured
Designed by	Graydon Hoare
Developer	Mozilla
First appeared	July 7, 2010; 9 years ago
Stable release	1.41.1[1] / February 27, 2020; 6 days ago
Typing discipline	Inferred, linear, nominal, static, strong
Implementation language	Rust
Platform	ARM, IA-32, x86-64, MIPS, PowerPC, SPARC, RISC-V[2][3]
OS	Linux, macOS, Windows, FreeBSD, OpenBSD,[4] Redox, Android, iOS[5]
License	MIT or Apache 2.0[6]
Filename extensions	.rs, .rlib
Website	www.rust-lang.org
Influenced by	
Alef,[7] C#,[7] C++,[7] Cyclone,[7][8] Erlang,[7] Haskell,[7] Limbo,[7] Newsqueak,[7] OCaml,[7] Ruby,[7] Scheme,[7] Standard ML,[7] Swift[7][9]	
Influenced	
Crystal, Elm,[10] Idris,[11] Spark,[12] Swift,[13] Project Verona[14]	

Typing discipline	Inferred, linear, nominal, static, strong

Traditional Logic

$$\frac{A \rightarrow B \quad A \rightarrow C}{A \rightarrow B \wedge C}$$

"**If A then B**" and "**if A then C**", we conclude:

If A then B and C

In usual mathematical reasoning, this is true

$$\frac{f(x) < a \rightarrow b < x \quad f(x) < a \rightarrow x < c}{f(x) < a \rightarrow b < x \; and \; x < c}$$ ✓

Linear Logic

"If you have \$1 then you get a chocolate" and "If you have \$1 then you get a candy", then one may have the following inference:

$$\frac{you\ have\ \$1 \rightarrow get\ a\ chocolate \quad you\ have\ \$1 \rightarrow get\ a\ candy}{you\ have\ \$1 \rightarrow get\ a\ chocolate\ and\ a\ candy}$$

Be careful to apply these logical inference rules when dealing with **resources**.

Linear Logic

- Instead of the traditional logical connective ∧:
- A ⊗ B can have both A and B at the same time
- A & B can A or B (not both) as you like
- ⟶ is replaced with ⊸

$$\frac{C \multimap A \quad D \multimap B}{C, D \multimap A \otimes B}$$

If you've two \$1 (\$2) then you get both chocolate and candy at the same time

$$\frac{C \multimap A \quad C \multimap B}{C \multimap A \& B}$$

If you've \$1 then you get either chocolate or candy as you like

Example 1 (Vending Machines)

- Let R be a vending machine that accepts rubles and dispenses packs of rolos, and
- let S be a vending machine that accepts shillings and dispenses packs of softmints.
- Let P_1, P_2, and P_3 be purchase orders.

Example 1 (Vending Machines)

- $P_1 \multimap (R \otimes S)$
- If we submit P_1, the college will install both machines side-by-side.
- $P_2 \multimap (R \,\&\, S)$
- If we submit P_2, the college will install one machine of our choice.
- $P_3 \multimap (R \oplus S)$
- If we submit P_3, the college will install one machine of their choice.
- Essentially, like cloning, use once.

Exponentials

- !A: Produces an unlimited number of A’s
- ?A: Consumes an unlimited number of A’s
- Connected via linear negation:

$$?A = !A$$

- Essentially, like copying, use infinite amount of times

Example 2 (Burger Shake)

\$5 per person Hamburger Fries or Wedges Unlimited Pepsi Ice-cream or sorbet (subject to availability)

D $:=$ **"one dollar"**

H $:=$ "hamburger"

F $:=$ "fries"

W $:=$ "Wedges"

P $:=$ "Pepsi"

I $:=$ " ice cream"

S $:=$ " sorbet"

Example 2 (Burger Shake)

$5 per person
Hamburger
Fries or Wedges
Unlimited Pepsi
Ice-cream or sorbet (subject to availability)

$$D \otimes D \otimes D \otimes D \otimes D \multimap H \otimes (F \& W) ! P \otimes (I \oplus S)$$

Example 3 (Lafont's Restaurant)

Today's Menu

((Vegetable Soup ⊕ Consomme Soup) &Salad) ⊗ (Fish & Meat) ⊗ !(Coffee) ⊗ ($2 ⊸ (Cake & Ice Cream))

Example 3 (Lafont's Restaurant)

Today's Menu

((Vegetable Soup $\oplus$ Consomme Soup) &Salad) $\otimes$ (Fish & Meat) $\otimes$!(Coffee) $\otimes$ (\$2 $\multimap$ (Cake & Ice Cream))

- **Entrée**: Either Soup or Salad as the customer chooses
- Either Vegetable Soup or Consomme Soup will be served depending on the day (not by the customer).
- **Main dish:** The customer can choose either a fish dish or a meat dish.
- **Coffee**: can be refilled as many times.
- **Dessert**: if customer will pay extra 2 dollars, they can choose either Cake or Ice Cream.

Language Specification

Remember this Slide!

$$\frac{A \to B \quad A \to C}{A \to B \wedge C}$$

"**If A then B**" and "**if A then C**", we conclude:

If A then B and C

In usual mathematical reasoning, this is true

$$\frac{f(x) < a \to b < x \quad f(x) < a \to x < c}{f(x) < a \to b < x \; and \; x < c}$$

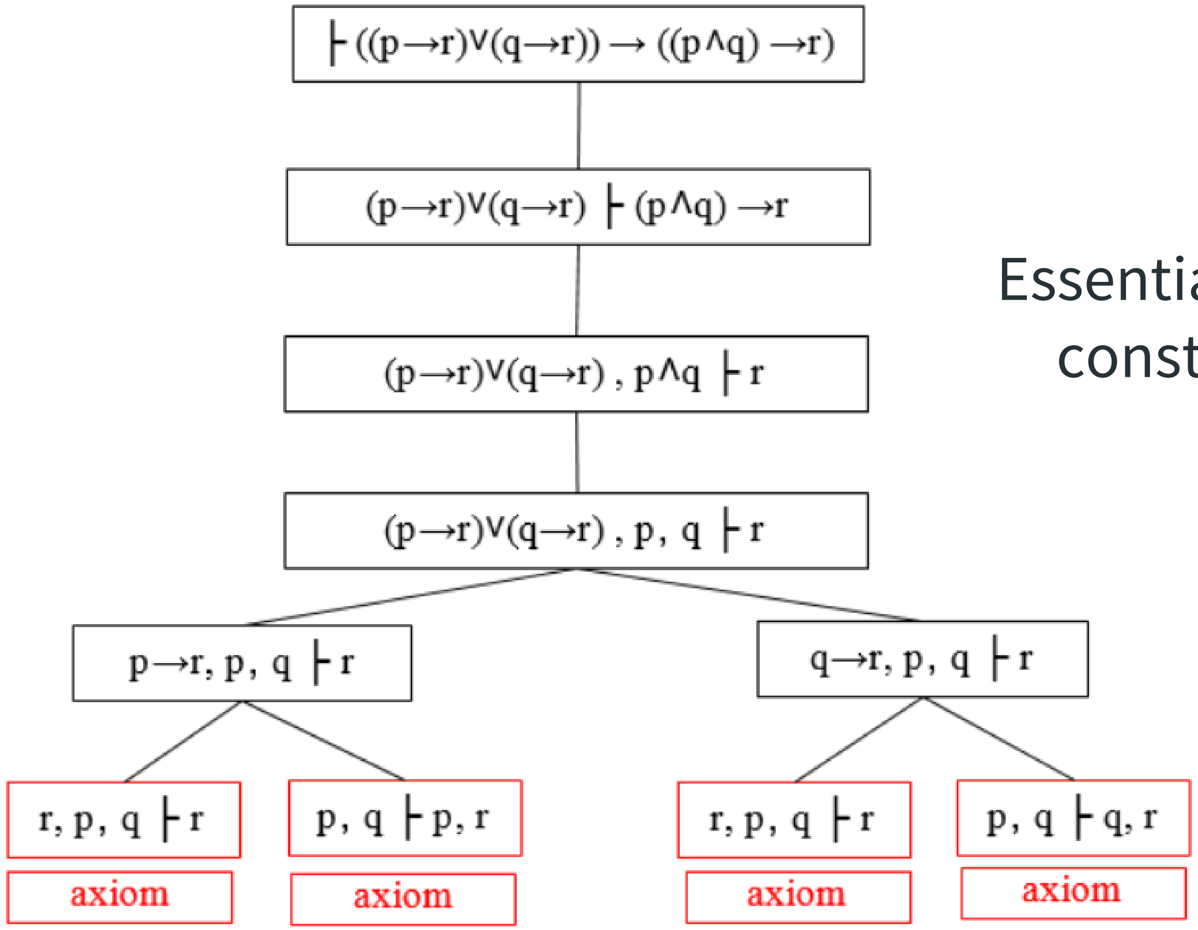
The turnstile symbol (⊢)

- How everyone writes these equations.

$$\frac{A \vdash A}{A \vdash A}$$

- Can be thought of as a sort of implication.

This is best left to compilers



Essentially, proof construction

Hypotheses on the left

- Read the following sequent as "Γ and A imply A."
- Γ (a capital Greek gamma) conventionally indicates hypotheses which are not relevant.

$$\Gamma, A \vdash A$$

Conclusions on the right

- Read the following sequent as "A implies A or Δ."
- Δ (a capital Greek delta) conventionally indicates conclusions which are not relevant.

$$A \vdash A, \Delta$$

Backwards deduction

- To show A ∨ B ⊢ A, B is true, you need to show A ⊢ A, B is true and B ⊢ A, B is true.

$$A \vee B \vdash A, B$$

$$\frac{A \vdash A, B \quad B \vdash A, B}{A \vee B \vdash A, B} \quad (\vee\ \mathrm{l(eft)})$$

Inference rules

Left side (hypothesis) ⊢ the right (conclusion)

$$\Gamma, A \vee B \ \vdash \ \Delta$$

$$\frac{\Gamma, A \ \vdash \Delta \qquad \Gamma, B \ \vdash \Delta}{\Gamma, A \vee B \ \vdash \ \Delta} \qquad (\vee\ \mathrm{l(eft)})$$

- The Γ and Δ are placeholders for other hypotheses and conclusions.

More Notation

- All the hypotheses on the left side can be thought of as **<u>ANDed</u>** together,
- and the conclusions on the right side **<u>ORed</u>** together.
- People now use short-hand notation.

$$\Gamma, A \wedge B \vdash A, \Delta \qquad\qquad \Gamma, A \vdash A \vee B, \Delta$$

$$\frac{\Gamma, A, B \vdash A, \Delta}{\Gamma, A \wedge B \vdash A, \Delta}\ (\wedge\ \mathrm{l(eft)}) \qquad\qquad \frac{\Gamma, A \vdash A, B, \Delta}{\Gamma, A \vdash A \vee B, \Delta}\ (\vee\ \mathrm{r(ight)})$$

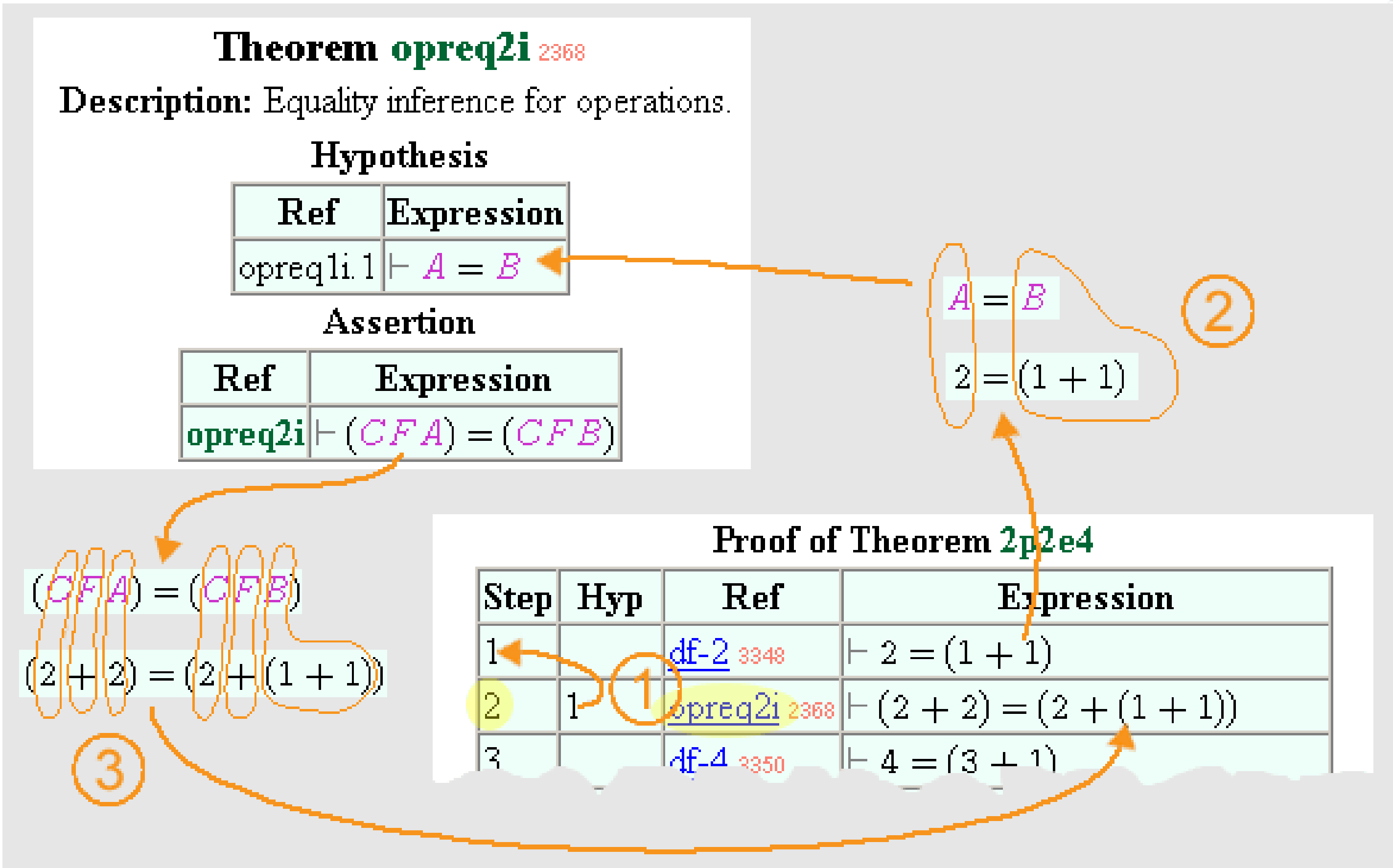
Meta-implication rule

- The turnstile itself can be thought of as implication.
- **Left side (hypothesis) ⊢ the right (conclusion)**
- To prove A → B, we can assume A as a hypothesis and prove B instead ("moving" the clause to the left side of the turnstile.)

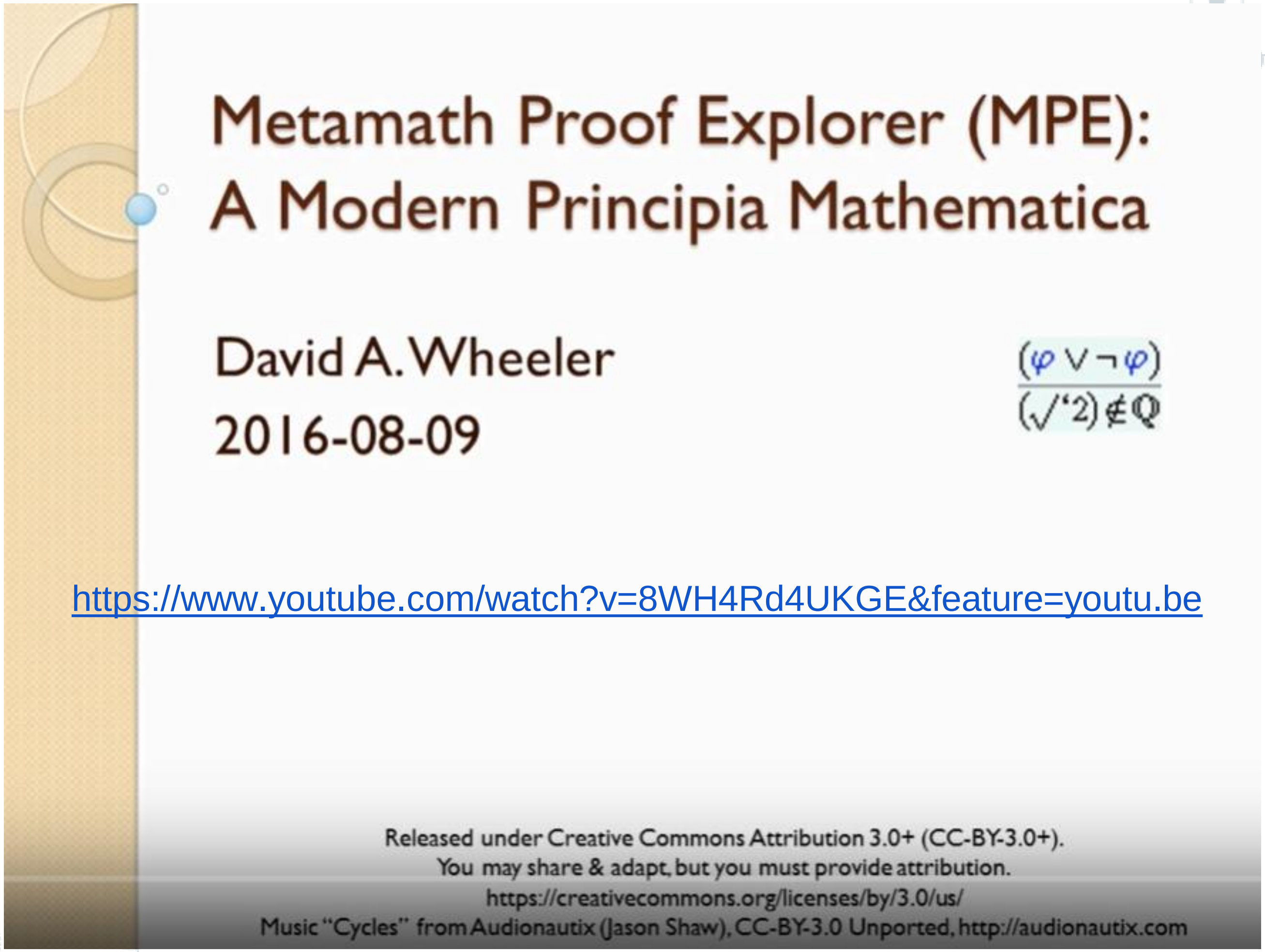
$$\Gamma \vdash A \rightarrow \ A, \Delta$$

$$\frac{A, \Gamma \vdash A, \Delta}{\Gamma \vdash A \rightarrow \ A, \Delta} \quad (\rightarrow \text{r(ight)})$$

Proof Construction Example



On YouTube, Must be Popular



https://www.youtube.com/watch?v=8WH4Rd4UKGE&feature=youtu.be