Discussion 8

October 18

Logistics

- Midterm exams are graded, released soon
  - Regrade requests open for roughly 2 weeks
- A6 will be released later today
  - Constructive logic
  - Gradual typing

Preview for today

- Very brief review of first order intuitionistic propositional logic
  - Classical vs. intuitionistic/constructive logic
- Curry-Howard correspondence
- Gradual typing

First Order Intuitionistic Propositional Logic

Intuitionistic propositional logic (IPL)

- Also called “constructive” logic
  - For our purposes, held in contrast to classical logic
- A fundamental building block for PL

Propositions and establishing them

- Prop A, B, … ::= $A \wedge B \mid A \vee B \mid A \supset B \mid \top \mid \bot$

- Establishing a proposition

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

  - "Γ entails A is true" "A is true assuming Γ"
  - "true" in "A true" isn't a boolean value
- Γ is a list of propositions which are **assumptions**

Assumptions

- Γ is a list of assumptions—if A is in Γ, it's an assumption
- So we can establish it!

$$\frac{}{\Gamma, A \vdash A}\ (\text{Assumption})$$

Conjunction

- “and”

$$\frac{\Gamma \vdash A \qquad \Gamma \vdash B}{\Gamma \vdash A \land B}\ (\land\text{-}I)$$

“conjunction/and introduction”

$$\frac{\Gamma \vdash A \land B}{\Gamma \vdash A}\ (\land\text{-}E\text{-}L)$$

$$\frac{\Gamma \vdash A \land B}{\Gamma \vdash B}\ (\land\text{-}E\text{-}R)$$

“conjunction/and elimination”

Disjunction

- “or”

“disjunction/or introduction”

$$\frac{\Gamma \vdash A}{\Gamma \vdash A \vee B}\ (\vee\text{-I-L})$$

$$\frac{\Gamma \vdash B}{\Gamma \vdash A \vee B}\ (\vee\text{-I-R})$$

$$\frac{\Gamma \vdash A \vee B \qquad \Gamma, A \vdash C \qquad \Gamma, B \vdash C}{\Gamma \vdash C}\ (\vee\text{-E})$$

“disjunction/or elimination”

Implication

- “if-then” “implies”

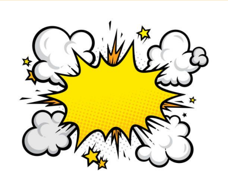
$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \supset B}\ (\supset\text{-}I) \qquad\qquad \frac{\Gamma \vdash A \supset B \qquad \Gamma \vdash A}{\Gamma \vdash B}\ (\supset\text{-}E)$$

“implication introduction” “implication elimination”

Truth and absurdity

- “tautology” “top”
- No elimination rule

- “falsity” “bot”
- No introduction rule



$$\frac{}{\Gamma \vdash \top}\ (\top\text{-}I)$$

“truth/top introduction”

$$\frac{\Gamma \vdash \bot}{\Gamma \vdash A}\ (\bot\text{-}E)$$

“bot elimination”

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

$$\frac{}{\Gamma \vdash \top}\ (\top\text{-}I)$$

$$\frac{\Gamma \vdash \bot}{\Gamma \vdash A}\ (\bot\text{-}E)$$

$$\frac{\Gamma \vdash A \qquad \Gamma \vdash B}{\Gamma \vdash A \wedge B}\ (\wedge\text{-}I)$$

$$\frac{\Gamma \vdash A \wedge B}{\Gamma \vdash A}\ (\wedge\text{-}E\text{-}L)$$

$$\frac{\Gamma \vdash A \wedge B}{\Gamma \vdash B}\ (\wedge\text{-}E\text{-}R)$$

$$\frac{}{\Gamma, A \vdash A}\ (\text{Assumption})$$

$$\frac{\Gamma \vdash A}{\Gamma \vdash A \vee B}\ (\vee\text{-}I\text{-}L)$$

$$\frac{\Gamma \vdash A \vee B \qquad \Gamma, A \vdash C \qquad \Gamma, B \vdash C}{\Gamma \vdash C}\ (\vee\text{-}E)$$

$$\frac{\Gamma \vdash B}{\Gamma \vdash A \vee B}\ (\vee\text{-}I\text{-}R)$$

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \supset B}\ (\supset\text{-}I)$$

$$\frac{\Gamma \vdash A \supset B \qquad \Gamma \vdash A}{\Gamma \vdash B}\ (\supset\text{-}E)$$

Exercise: proof I

Task: prove the following:

$$(A \supset B) \vee B \vdash A \supset B$$

Relevant rules:

$$\frac{\Gamma \vdash A}{\Gamma \vdash A \vee B}\,(\vee\text{-}I\text{-}L) \qquad \frac{\Gamma \vdash A \vee B \qquad \Gamma, A \vdash C \qquad \Gamma, B \vdash C}{\Gamma \vdash C}\,(\vee\text{-}E) \qquad \frac{}{\Gamma, A \vdash A}\,(\text{Assumption})$$

$$\frac{\Gamma \vdash B}{\Gamma \vdash A \vee B}\,(\vee\text{-}I\text{-}R) \qquad \frac{\Gamma, A \vdash B}{\Gamma \vdash A \supset B}\,(\supset\text{-}I) \qquad \frac{\Gamma \vdash A \supset B \qquad \Gamma \vdash A}{\Gamma \vdash B}\,(\supset\text{-}E)$$

Exercise: proof I [solution]

$\Gamma_1 \stackrel{\text{def}}{=} (A \supset B) \vee B$

$$\dfrac{\dfrac{}{\Gamma_1 \vdash (A \supset B) \vee B}\,(\text{Assumption}) \qquad \dfrac{}{\Gamma_1, A \supset B \vdash A \supset B}\,(\text{Assumption}) \qquad D_1}{(A \supset B) \vee B \vdash A \supset B}\,(\vee\text{-}E)$$

where D_1 :

$$\dfrac{\dfrac{}{\Gamma_1, B, A \vdash B}\,(\text{Assumption})}{\Gamma_1, B \vdash A \supset B}\,(\supset\text{-}I)$$

Exercise: distributive law

Task: prove one direction of the distributive law for conjunction over disjunction:

$$\Gamma \vdash A \wedge (B \vee C) \supset (A \wedge B) \vee (A \wedge C)$$

$$\frac{\Gamma \vdash A \vee B \qquad \Gamma, A \vdash C \qquad \Gamma, B \vdash C}{\Gamma \vdash C}\ (\vee\text{-}E) \qquad \frac{}{\Gamma, A \vdash A}\ (\text{Assumption})$$

$$\frac{\Gamma \vdash A \qquad \Gamma \vdash B}{\Gamma \vdash A \wedge B}\ (\wedge\text{-}I) \qquad \frac{\Gamma \vdash A \wedge B}{\Gamma \vdash A}\ (\wedge\text{-}E\text{-}L) \qquad \frac{\Gamma \vdash A}{\Gamma \vdash A \vee B}\ (\vee\text{-}I\text{-}L)$$

$$\frac{\Gamma \vdash A \wedge B}{\Gamma \vdash B}\ (\wedge\text{-}E\text{-}R) \qquad \frac{\Gamma \vdash B}{\Gamma \vdash A \vee B}\ (\vee\text{-}I\text{-}R)$$

But what about '¬'?

- Instead of defining a primitive negation, we can define it in terms of existing connectives:

$$\neg A \stackrel{def}{=} A \supset \bot$$

- ¬A is true if the acceptance of A leads to absurdity
- Examples:
  - $\neg(A \vee B) = (A \vee B) \supset \bot$
  - $\neg A \wedge \neg B = (A \supset \bot) \wedge (B \supset \bot)$

Remember classical logic?

- Propositions are assigned truth values “true” or “false”
  - Every proposition is either true or false
  - Connectives are Boolean functions
- Primitive notion of negation

- Classical notions:
  - Law of the excluded middle (LEM): $\Gamma \vdash A \lor \neg A$
  - Law of double negation (DNE): $\Gamma \vdash \neg\neg A \supset A$
  - We get stuff like proofs by contradiction from these

Mathematical constructivism

- In classical logic, all propositions are either true or false
  - LEM, DNE, etc.
- But constructivism demands *concrete evidence*
  - We can’t just say A is true or A is false without concrete evidence for either
  - Classical logic -> truth; IPL -> provability
- LEM doesn’t hold in IPL
  - An exact proof is more complicated (and fairly out of scope for this class)
  - We can also get classical propositional logic from IPL by adding LEM, DNE, or an equivalent axiom

Exercise: intuitionistic double negation

We can’t prove $\Gamma \vdash \neg\neg A \supset A$, but can we prove $\Gamma \vdash A \supset \neg\neg A$?

$$\frac{}{\Gamma, A \vdash A}\ (\text{Assumption}) \qquad \frac{\Gamma \vdash A \supset B \qquad \Gamma \vdash A}{\Gamma \vdash B}\ (\supset\text{-E})$$

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \supset B}\ (\supset\text{-I}) \qquad \frac{}{\Gamma \vdash \top}\ (\top\text{-I}) \qquad \frac{\Gamma \vdash \bot}{\Gamma \vdash A}\ (\bot\text{-E})$$

Curry-Howard Correspondence

Propositions as types

- Did anything about the IPL rules and our proofs feel familiar?
- Key insight: programs are just like proofs

- **Curry-Howard Correspondence:**
  proofs are programs, and propositions are types

Proofs as programs

If we change our judgment to establish propositions…

$\boxed{\Gamma \vdash A}$

$\boxed{\Gamma \vdash e : A}$ "e is a proof of A assuming Γ"
where Γ is now $x_1 : A_1, x_2 : A_2, \ldots$

In 490, we can view propositions as ALFA types,
and construct proofs by writing ALFA programs!

Proofs as programs

- Variables in the context represent different assumptions
- Expressions of the same type are different proofs of the same statement

- To prove A: can we find a program of type A?

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

"e is a proof of A assuming Γ"
where Γ is now $x_1 : A_1, x_2 : A_2, \ldots$

Proofs as programs

$$\frac{}{\Gamma, x : A \vdash x : A}\ (\text{Assumption})$$

$$\frac{\Gamma \vdash e_1 : A \qquad \Gamma \vdash e_2 : B}{\Gamma \vdash (e_1, e_2) : A \wedge B}\ (\wedge\text{-}I)$$

$$\frac{}{\Gamma \vdash () : \top}\ (\top\text{-}I)$$

$$\frac{\Gamma \vdash e : \bot}{\Gamma \vdash \mathrm{raise}(e) : A}\ (\bot\text{-}E)$$

$$\frac{\Gamma \vdash e : A}{\Gamma \vdash L\, e : A \vee B}\ (\vee\text{-}I\text{-}L)$$

$$\frac{\Gamma \vdash e : B}{\Gamma \vdash R\, e : A \vee B}\ (\vee\text{-}I\text{-}R)$$

$$\frac{\Gamma \vdash e : A \wedge B}{\Gamma \vdash e.0 : A}\ (\wedge\text{-}E\text{-}L)$$

$$\frac{\Gamma \vdash e : A \wedge B}{\Gamma \vdash e.1 : B}\ (\wedge\text{-}E\text{-}R)$$

$$\frac{\Gamma \vdash e_1 : A \supset B \qquad \Gamma \vdash e_2 : A}{\Gamma \vdash e_1(e_2) : B}\ (\supset\text{-}E)$$

$$\frac{\Gamma, x : A \vdash e : B}{\Gamma \vdash (\mathrm{fun}\ (x : A) \to e) : A \supset B}\ (\supset\text{-}I)$$

$$\frac{\Gamma \vdash e : A \vee B \qquad \Gamma, x : A \vdash e_1 : C \qquad \Gamma, y : B \vdash e_2 : C}{\Gamma \vdash \begin{array}{l}\mathrm{case}\ e\ \mathrm{of}\ L(x) \to e_1 \\ \quad\mathrm{else}\ R(y) \to e_2\end{array} : C}\ (\vee\text{-}E)$$

Exercise: proof II (proof I again)

Task: construct a *program* that proves the following:

$$(A \supset B) \vee B \vdash A \supset B$$

$$\frac{\Gamma \vdash e : A}{\Gamma \vdash L\, e : A \vee B}\ (\vee\text{-I-L})$$

$$\frac{\Gamma \vdash e : B}{\Gamma \vdash R\, e : A \vee B}\ (\vee\text{-I-R})$$

$$\frac{}{\Gamma, x : A \vdash x : A}\ (\text{Assumption})$$

$$\frac{\Gamma \vdash e_1 : A \supset B \quad \Gamma \vdash e_2 : A}{\Gamma \vdash e_1(e_2) : B}\ (\supset\text{-E})$$

$$\frac{\Gamma, x : A \vdash e : B}{\Gamma \vdash (\text{fun } (x : A) \to e) : A \supset B}\ (\supset\text{-I})$$

$$\frac{\Gamma \vdash e : A \vee B \quad \Gamma, x : A \vdash e_1 : C \quad \Gamma, y : B \vdash e_2 : C}{\Gamma \vdash \begin{array}{l}\text{case } e \text{ of } L(x) \to e_1 \\ \text{else } R(y) \to e_2\end{array} : C}\ (\vee\text{-E})$$

Gradual Typing

Entry level nightmare

Imagine: you’re working on a huge Javascript codebase that has seen hundreds of (incompetent) contributors over the last 15 years

- Frequent runtime type error bug reports
- Impossible to grok due to lack of types

“It’d be nice if we used a statically typed language…”

Entry level nightmare

What can you do?

- Do nothing, suffer
  - :(
- Start to rewrite the project in your favorite statically typed language
  - Simply infeasible
- Slowly migrate the project to use static types?

Entry level nightmare

What would this look like?

- Parts (say A) of the program that are completely dynamically typed
- Parts (say B) of the program that are now fully statically typed
- Parts of the program where A and B interact…
  - How to make this interaction easy to use?
  - How to make sure guarantees provided by types to B are not violated by A?

Gradual typing

Idea: a statically typed language, except you can leave types as *unknown* (to represent dynamic typing)

- Type system somehow facilitates clean and correct/safe interaction between unknown type and other types
  - Guaranteed by runtime tag checks, but these have performance overhead
- Can gradually fill in unknown types with concrete ones
  - Gradually reduce chance of runtime type errors
  - Gradually reduce performance overhead

Who cares, anyways?

- Flexibility to omit types when prototyping
  - We might not even know what types we need yet!
- Lots of existing codebases written in dynamically typed languages
  - Python, JavaScript, JavaScript, JavaScript…
  - *Gradually* migrate these to be statically typed (to avoid whole classes of bugs!)
- Lots of people, apparently:
  - PEP 484, Typescript, C#, etc.

Gradual ALFA

- Take ALFA (not Recursive ALFA) and add gradual typing
- A new unknown type, written ?
- Type system is given bidirectionally

Recall subsumption

- Subsumption in ALFA

$$\frac{\Gamma \vdash e \Rightarrow \tau}{\Gamma \vdash e \Leftarrow \tau} \ (\text{A-Subsumption})$$

- Subsumption in Gradual ALFA

$$\frac{\Gamma \vdash e \Rightarrow \tau' \qquad \boxed{\tau' \sim \tau}}{\Gamma \vdash e \Leftarrow \tau} \ (\text{A-Subsumption})$$

From type equality to consistency

Consistency: Two types are consistent if they differ only up to the appearance of an unknown type. Type consistency is symmetric and reflexive.

$$\frac{}{\tau \sim \tau}\ (\text{C-Refl}) \qquad \frac{}{? \sim \tau}\ (\text{C-UnkL}) \qquad \frac{}{\tau \sim ?}\ (\text{C-UnkR}) \qquad \frac{\tau_{\mathsf{in}} \sim \tau'_{\mathsf{in}} \qquad \tau_{\mathsf{out}} \sim \tau'_{\mathsf{out}}}{\tau_{\mathsf{in}} \to \tau_{\mathsf{out}} \sim \tau'_{\mathsf{in}} \to \tau'_{\mathsf{out}}}\ (\text{C-Arrow})$$

$$\frac{\tau_\ell \sim \tau'_\ell \qquad \tau_r \sim \tau'_r}{\tau_\ell \times \tau_r \sim \tau'_\ell \times \tau'_r}\ (\text{C-Prod}) \qquad \frac{\tau_\ell \sim \tau'_\ell \qquad \tau_r \sim \tau'_r}{\tau_\ell + \tau_r \sim \tau'_\ell + \tau'_r}\ (\text{C-Sum})$$

Question: Why can’t type consistency be transitive?

From type equality to consistency

$$\frac{}{\tau \sim \tau}\ \text{(C-Refl)} \qquad \frac{}{? \sim \tau}\ \text{(C-UnkL)} \qquad \frac{}{\tau \sim ?}\ \text{(C-UnkR)} \qquad \frac{\tau_{\mathsf{in}} \sim \tau'_{\mathsf{in}} \qquad \tau_{\mathsf{out}} \sim \tau'_{\mathsf{out}}}{\tau_{\mathsf{in}} \to \tau_{\mathsf{out}} \sim \tau'_{\mathsf{in}} \to \tau'_{\mathsf{out}}}\ \text{(C-Arrow)}$$

$$\frac{\tau_\ell \sim \tau'_\ell \qquad \tau_r \sim \tau'_r}{\tau_\ell \times \tau_r \sim \tau'_\ell \times \tau'_r}\ \text{(C-Prod)} \qquad \frac{\tau_\ell \sim \tau'_\ell \qquad \tau_r \sim \tau'_r}{\tau_\ell + \tau_r \sim \tau'_\ell + \tau'_r}\ \text{(C-Sum)}$$

Suppose type consistency is transitive, we can show that Bool ~ Num by showing that **Bool ~ ?** (C-UnkR) and **? ~ Num** (C-UnkL). By transitivity, we conclude that **Bool ~ Num**.

Analysing functions with consistency

- A-Fun rule in ALFA

$$\frac{\Gamma, x : \tau_{\mathsf{in}} \vdash e_{\mathsf{body}} \Leftarrow \tau_{\mathsf{out}}}{\Gamma \vdash \mathsf{fun}\ x \rightarrow e_{\mathsf{body}} \Leftarrow \tau_{\mathsf{in}} \rightarrow \tau_{\mathsf{out}}} \ \text{(A-Fun rule from ALFA)}$$

- A-Fun rule in Gradual ALFA

$$\frac{\boxed{\tau \blacktriangleright_{\rightarrow} \tau_{\mathsf{in}} \rightarrow \tau_{\mathsf{out}}} \qquad \Gamma, x : \tau_{\mathsf{in}} \vdash e_{\mathsf{body}} \Leftarrow \tau_{\mathsf{out}}}{\Gamma \vdash \mathsf{fun}\ x \rightarrow e_{\mathsf{body}} \Leftarrow \tau} \ \text{(A-Fun)}$$

Matched arrow type

$\tau \blacktriangleright_{\rightarrow} \tau_{\text{in}} \rightarrow \tau_{\text{out}}$ reads τ has matched arrow type $\tau_{\text{in}} \rightarrow \tau_{\text{out}}$

$$\frac{}{\tau_\ell \rightarrow \tau_r \blacktriangleright_{\rightarrow} \tau_\ell \rightarrow \tau_r} \quad \text{(MA-Arrow)}$$

$$\frac{}{? \blacktriangleright_{\rightarrow} ? \rightarrow ?} \quad \text{(MA-Hole)}$$

Exercise: proof III

```
⊢ fun b → if b then 0 else True ⩽ Bool → ?
```

Exercise: proof III

$\vdash$ `fun b → if b then` $\underline{\texttt{0}}$ `else True ⩽ Bool → ?`

$$\frac{}{\tau_\ell \to \tau_r \blacktriangleright_\to \tau_\ell \to \tau_r}\ (\text{MA-Arrow})$$

$$\frac{\Gamma \vdash e \Rightarrow \tau' \qquad \tau' \sim \tau}{\Gamma \vdash e \Leftarrow \tau}\ (\text{A-Subsumption})$$

$$\frac{\Gamma \vdash e_{\text{cond}} \Leftarrow \mathsf{Bool} \qquad \Gamma \vdash e_{\text{then}} \Leftarrow \tau \qquad \Gamma \vdash e_{\text{else}} \Leftarrow \tau}{\Gamma \vdash (\mathsf{if}\ e_{\text{cond}}\ \mathsf{then}\ e_{\text{then}}\ \mathsf{else}\ e_{\text{else}}) \Leftarrow \tau}\ (\text{A-If})$$

$$\frac{}{\tau \sim \tau}\ (\text{C-Refl})$$

$$\frac{}{\tau \sim ?}\ (\text{C-UnkR})$$

$$\frac{\tau \blacktriangleright_\to \tau_{\mathsf{in}} \to \tau_{\mathsf{out}} \qquad \Gamma, x : \tau_{\mathsf{in}} \vdash e_{\mathsf{body}} \Leftarrow \tau_{\mathsf{out}}}{\Gamma \vdash \mathsf{fun}\ x \to e_{\mathsf{body}} \Leftarrow \tau}\ (\text{A-Fun})$$

$$\frac{}{\Gamma \vdash \underline{n} \Rightarrow \mathsf{Num}}\ (\text{S-NumLiteral})$$

$$\frac{}{\Gamma \vdash \mathsf{True} \Rightarrow \mathsf{Bool}}\ (\text{S-True})$$

$$\frac{x : \tau \in \Gamma}{\Gamma \vdash x \Rightarrow \tau}\ (\text{S-Var})$$

Exercise: proof III [solution]

$$\mathcal{D}_3 = \dfrac{\dfrac{}{b:\mathrm{Bool} \vdash \mathrm{True} \Rightarrow \mathrm{Bool}}\,\text{(S-True)} \qquad \dfrac{}{\mathrm{Bool} \sim ?}\,\text{(C-UnkR)}}{b:\mathrm{Bool} \vdash \mathrm{True} \Leftarrow ?}\,\text{(A-Subsumption)}$$

$$\mathcal{D}_2 = \dfrac{\dfrac{}{b:\mathrm{Bool} \vdash \underline{0} \Rightarrow \mathrm{Num}}\,\text{(S-NumLiteral)} \qquad \dfrac{}{\mathrm{Num} \sim ?}\,\text{(C-UnkR)}}{b:\mathrm{Bool} \vdash \underline{0} \Leftarrow ?}\,\text{(A-Subsumption)}$$

$$\mathcal{D}_1 = \dfrac{\dfrac{b:\mathrm{Bool} \in b:\mathrm{Bool}}{b:\mathrm{Bool} \vdash b \Rightarrow \mathrm{Bool}}\,\text{(S-Var)} \qquad \dfrac{}{\mathrm{Bool} \sim \mathrm{Bool}}\,\text{(C-Refl)}}{b:\mathrm{Bool} \vdash b \Leftarrow \mathrm{Bool}}\,\text{(A-Subsumption)}$$

$$\dfrac{\dfrac{}{\mathrm{Bool} \to ? \blacktriangleright_{\to} \mathrm{Bool} \to ?}\,\text{(MA-Arrow)} \qquad \dfrac{\mathcal{D}_1 \quad \mathcal{D}_2 \quad \mathcal{D}_3}{b:\mathrm{Bool} \vdash \text{if } b \text{ then } \underline{0} \text{ else True} \Leftarrow ?}\,\text{(A-If)}}{\bullet \vdash \text{fun } b \to \text{if } b \text{ then } \underline{0} \text{ else True} \Leftarrow \mathrm{Bool} \to ?}\,\text{(A-Fun)}$$

Exercise: proof IV

$\vdash$ `fun n → (n + 1, n) ≤ ? → (Num x Bool)`

Exercise: proof IV

$\vdash$ `fun n → (n + 1, n) ⩽ ? → (Num x Bool)`

$$\frac{}{\tau_\ell \to \tau_r \blacktriangleright_\to \tau_\ell \to \tau_r} \text{ (MA-Arrow)}$$

$$\frac{\Gamma \vdash e \Rightarrow \tau' \qquad \tau' \sim \tau}{\Gamma \vdash e \Leftarrow \tau} \text{ (A-Subsumption)}$$

$$\frac{\Gamma \vdash e_\ell \Leftarrow \tau_\ell \qquad \Gamma \vdash e_r \Leftarrow \tau_r}{\Gamma \vdash (e_\ell, e_r) \Leftarrow \tau_\ell \times \tau_r} \text{ (A-Pair)}$$

$$\frac{\Gamma \vdash e_\ell \Leftarrow \mathsf{Num} \qquad \Gamma \vdash e_r \Leftarrow \mathsf{Num}}{\Gamma \vdash (e_\ell + e_r) \Rightarrow \mathsf{Num}} \text{ (S-Plus)}$$

$$\frac{\tau \blacktriangleright_\to \tau_{\mathsf{in}} \to \tau_{\mathsf{out}} \qquad \Gamma, x : \tau_{\mathsf{in}} \vdash e_{\mathsf{body}} \Leftarrow \tau_{\mathsf{out}}}{\Gamma \vdash \mathsf{fun}\ x \to e_{\mathsf{body}} \Leftarrow \tau} \text{ (A-Fun)}$$

$$\frac{}{\tau \sim \tau} \text{ (C-Refl)}$$

$$\frac{}{? \sim \tau} \text{ (C-UnkL)}$$

$$\frac{}{\Gamma \vdash \underline{n} \Rightarrow \mathsf{Num}} \text{ (S-NumLiteral)}$$

$$\frac{x : \tau \in \Gamma}{\Gamma \vdash x \Rightarrow \tau} \text{ (S-Var)}$$

Exercise: proof IV [solution]

$$
\mathcal{D}_4 = \dfrac{\dfrac{\dfrac{}{n{:}? \vdash n \Rightarrow ?}\,(\text{S-Var}) \qquad \dfrac{}{? \sim \mathit{Num}}\,(\text{C-UnkL})}{n{:}? \vdash n \Leftarrow \mathit{Num}}\,(\text{A-Subsumption}) \qquad \dfrac{\dfrac{}{n{:}? \vdash \underline{1} \Rightarrow \mathit{Num}}\,(\text{S-NumLiteral}) \qquad \dfrac{}{\mathit{Num} \sim \mathit{Num}}\,(\text{C-Refl})}{n{:}? \vdash \underline{1} \Leftarrow \mathit{Num}}}{n{:}? \vdash n + \underline{1} \Rightarrow \mathit{Num}}\,(\text{S-Plus})
$$

$$
\mathcal{D}_2 = \dfrac{\mathcal{D}_4 \qquad \dfrac{}{\mathit{Num} \sim \mathit{Num}}\,(\text{C-Refl})}{n{:}? \vdash n + \underline{1} \Leftarrow \mathit{Num}}\,(\text{A-Subsumption}) \qquad \dfrac{\dfrac{}{n{:}? \vdash n \Rightarrow ?}\,(\text{S-Var}) \qquad \dfrac{}{? \sim \mathit{Bool}}\,(\text{C-UnkL})}{n{:}? \vdash n \Leftarrow \mathit{Bool}} = \mathcal{D}_3
$$

$$
\mathcal{D}_1 = \dfrac{\mathcal{D}_2 \qquad \mathcal{D}_3}{n{:}? \vdash (n + \underline{1},\ n) \Leftarrow \mathit{Num} \times \mathit{Bool}}\,(\text{A-Pair})
$$

$$
\dfrac{\dfrac{}{? \to (\mathit{Num} \times \mathit{Bool}) \blacktriangleright_{\to} ? \to (\mathit{Num} \times \mathit{Bool})}\,(\text{MA-Arrow}) \qquad \mathcal{D}_1}{\vdash \mathtt{fun}\ n \to (n + \underline{1},\ n) \Leftarrow ? \to (\mathit{Num} \times \mathit{Bool})}\,(\text{A-Fun})
$$