CIS 500
Software Foundations
Fall 2006

November 6
Some Hints
Hints

- The exam will potentially cover everything in the course so far, but will focus on material we’ve seen since the first midterm.
- There will be a question that is also a one-star exercise from the book.
- There will be a question similar to problem 6 from midterm 1 (“Which properties remain true if we change one of the type systems we’ve studied in the following way…?”)
- There will be (at least) one question based on one of the proofs in chapter 15.
- For PhD students, there will be a question involving subtyping and references.
Review
What are the types of these expressions?

- \( \lambda x: \text{Bool} \rightarrow \text{Bool}. \ x \ (x \ (x \ (\text{true}))) \)
What are the types of these expressions?

- \( \lambda x : \text{Bool} \rightarrow \text{Bool}. \ x \ (x \ (x \ (\text{true}))) \)
- \((\lambda x : \text{Bool}. \ \lambda y : \text{Bool} \rightarrow \text{Bool}. \ \text{true}) \) false \( (\lambda z : \text{Bool} \rightarrow \text{Bool}. \ \text{true}) \)
What are the types of these expressions?

- $\lambda x: \text{Bool} \rightarrow \text{Bool}. \ x \ (x \ (x \ (\text{true})))$
- $(\lambda x: \text{Bool}. \ \lambda y: \text{Bool} \rightarrow \text{Bool}. \ \text{true})$
  
  $\text{false} \ (\lambda z: \text{Bool} \rightarrow \text{Bool}. \ \text{true})$
- $(\lambda x: \text{Bool}. \ \lambda y: \text{Bool}. \ \text{error})$
  
  $\text{false} \ \text{false} \ \text{false} \ \text{false} \ \text{false}$
What are the types of these expressions?

- \( \lambda x: \text{Bool} \rightarrow \text{Bool}. \ x \ (x \ (x \ (\text{true}))) \)
- \( (\lambda x: \text{Bool}. \ \lambda y: \text{Bool} \rightarrow \text{Bool}. \ \text{true}) \)
  \quad \text{false} \ (\lambda z: \text{Bool} \rightarrow \text{Bool}. \ \text{true})
- \( (\lambda x: \text{Bool}. \ \lambda y: \text{Bool}. \ \text{error}) \)
  \quad \text{false} \ \text{false} \ \text{false} \ \text{false} \ \text{false} \ \text{false} \ \text{false} \ \text{false} \ \text{false}
- \( (\lambda x: \text{Bool}. \ \lambda y: \text{Bool}. \ \text{true}) \)
  \quad \text{false} \ \text{false} \ \text{false} \ \text{false} \ \text{false} \ \text{false} \ \text{false} \ \text{false} \ \text{false} \
What are the types of these expressions?

- \( \lambda x: \text{Bool} \rightarrow \text{Bool}. \ x \ (x \ (x \ (\text{true}))) \)
- \((\lambda x: \text{Bool}. \ \lambda y: \text{Bool} \rightarrow \text{Bool}. \ \text{true})\)
  - \(\text{false} \ (\lambda z: \text{Bool} \rightarrow \text{Bool}. \ \text{true})\)
- \((\lambda x: \text{Bool}. \ \lambda y: \text{Bool}. \ \text{error})\)
  - \(\text{false} \ \text{false} \ \text{false} \ \text{false} \ \text{false}\)
- \((\lambda x: \text{Bool}. \ \lambda y: \text{Bool}. \ \text{true})\)
  - \(\text{false} \ \text{false} \ \text{false} \ \text{false} \ \text{false}\)
- \text{try}\n  - (if \(\lambda x: \text{Bool}. \ x\) \ \text{error}\n    \text{then} \ \text{error} \ \text{false}\n    \text{else} \ \text{error})\)
  with
  \(\lambda y: \text{Bool} \rightarrow \text{Bool}. \ y\)
For reference: Typing rules for exceptions

\[ \Gamma \vdash \text{error} : T \quad (T\text{-ERROR}) \]

\[ \Gamma \vdash t_1 : T \quad \Gamma \vdash t_2 : T \]

\[ \Gamma \vdash \text{try } t_1 \text{ with } t_2 : T \quad (T\text{-TRY}) \]
Give the result of evaluation and the final store after each of these expressions is evaluated to a normal form starting in the empty store.

▶ let x = ref 0 in
    let y = ref 1 in
    y:=3
    !x

▶ let x = ref 0 in
    let y = ref 1 in
    !x

▶ let x = ref 0 in
    let y = x in
    let x = y in !x
Give the result of evaluation and the final store after each of these expressions is evaluated to a normal form starting in the empty store.

- let x = ref 0 in
  let y = ref 1 in
  y := 3
  !x
- let x = ref 0 in
  let y = ref 1 in
  let x = y in !x
Give the result of evaluation and the final store after each of these expressions is evaluated to a normal form starting in the empty store.

▶ let x = ref 0 in
    let y = ref 1 in
    y := 3
    !x

▶ let x = ref 0 in
    let y = ref 1 in
    let x = y in !x

▶ let x = ref 0 in
    let y = x in
    let x = y in !x
For reference: Evaluation rules for references

\[ \frac{l \not\in \text{dom}(\mu)}{\text{ref } v_1 | \mu \rightarrow l | (\mu, l \mapsto v_1)} \quad \text{(E-RefV)} \]

\[ \frac{\mu(l) = v}{!l | \mu \rightarrow v | \mu} \quad \text{(E-DerefLoc)} \]

\[ l := v_2 | \mu \rightarrow \text{unit} | [l \mapsto v_2]\mu \quad \text{(E-Assign)} \]

(Plus several congruence rules.)
Which of the following functions *could* evaluate to 42 when applied to a single argument and evaluated with a store of the appropriate type?

- \( \lambda x: \text{Ref Nat.} \; !x + 1 \)
Which of the following functions *could* evaluate to 42 when applied to a single argument and evaluated with a store of the appropriate type?

▶ \( \lambda x:\text{Ref Nat.} \; !x+1 \)

▶ \( \lambda x:\text{Ref Nat.} \; x \)
Which of the following functions *could* evaluate to 42 when applied to a single argument and evaluated with a store of the appropriate type?

- $\lambda x: \text{Ref Nat. } !x + 1$
- $\lambda x: \text{Ref Nat. } x$
- $\lambda x: \text{Ref Nat. } (x := 3; l_1 := 42; !l_1)$
Which of the following functions *could* evaluate to 42 when applied to a single argument and evaluated with a store of the appropriate type?

- \( \lambda x: \text{Ref Nat} . \ !x + 1 \)
- \( \lambda x: \text{Ref Nat} . \ x \)
- \( \lambda x: \text{Ref Nat} . \ (x:=3; l_1:=42; !l_1) \)
- \( \lambda f: \text{Unit} \to \text{Unit} . \ (l_1:=3; f \ \text{unit}; \ !l_1) \)
Preservation and progress for chapter 13

- The preservation and progress proofs for $\lambda \rightarrow$ with references are just sketched in TAPL.
- Working out the details for yourself is an excellent exercise.
- A question based on this proof may appear on the final exam, but will *not* appear on the coming midterm.
Subtyping

For each of the following pairs of terms, say whether the one on the left is a subtype of the one on the right, a supertype, equivalent, or incomparable.

- $(\{}\rightarrow\{}\) \rightarrow \text{Top}$ and $\text{Top} \rightarrow \text{Top}$
Subtyping

For each of the following pairs of terms, say whether the one on the left is a subtype of the one on the right, a supertype, equivalent, or incomparable.

- ({}→{})→Top and Top→Top
- (Top→Top)→{}→{} and (Top→{})→Top
Subtyping

For each of the following pairs of terms, say whether the one on the left is a subtype of the one on the right, a supertype, equivalent, or incomparable.

- $(\{\}, \rightarrow, \{\}) \rightarrow \text{Top}$ and $\text{Top} \rightarrow \text{Top}$
- $(\text{Top} \rightarrow \text{Top}) \rightarrow \{\} \rightarrow \{\}$ and $(\text{Top} \rightarrow \{\}) \rightarrow \text{Top}$
- \{a: \text{Top}, b: \{c: \text{Top}, d: \text{Top}\}\} and \{b: \{d: \text{Top}, c: \text{Top}\}, a: \text{Top}\}
Subtyping

For each of the following pairs of terms, say whether the one on the left is a subtype of the one on the right, a supertype, equivalent, or incomparable.

- \((\{\} \rightarrow \{\}) \rightarrow \text{Top}\) and \(\text{Top} \rightarrow \text{Top}\)
- \((\text{Top} \rightarrow \text{Top}) \rightarrow \{\} \rightarrow \{\}\) and \((\text{Top} \rightarrow \{\}) \rightarrow \text{Top}\)
- \{a:\text{Top}, b:\{c:\text{Top}, d:\text{Top}\}\}\) and \{b:\{d:\text{Top}, c:\text{Top}\}, a:\text{Top}\}\)
- \(<l:\text{Top}, m:\{n:\text{Top}\}>\) and \(<m:\{n:\text{Top}, o:\text{Top}\}>\)
Subtyping

For each of the following pairs of terms, say whether the one on the left is a subtype of the one on the right, a supertype, equivalent, or incomparable.

- $(\emptyset \rightarrow \emptyset) \rightarrow \text{Top}$ and $\text{Top} \rightarrow \text{Top}$
- $(\text{Top} \rightarrow \text{Top}) \rightarrow \emptyset \rightarrow \emptyset$ and $(\text{Top} \rightarrow \emptyset) \rightarrow \text{Top}$
- $\{a: \text{Top}, b: \{c: \text{Top}, d: \text{Top}\}\}$ and \{b: \{d: \text{Top}, c: \text{Top}\}, a: \text{Top}\}
- $<l: \text{Top}, m: \{n: \text{Top}\}>$ and $<m: \{n: \text{Top}, o: \text{Top}\}>$
- $<> \rightarrow \text{Top}$ and $\emptyset \rightarrow \text{Top}$
Subtyping

Draw a subtyping derivation for the following statement:

$$(\text{Top} \rightarrow \{x: \text{Nat}\}) \rightarrow \{x: \text{Nat}, y: \text{Nat}\} <: (\emptyset \rightarrow \emptyset) \rightarrow \{y: \text{Nat}\}$$
For reference: Subtyping rules

\[
S <: S \\
S <: U \quad U <: T \\
\quad S <: T
\]

(S-REFL)

(S-TRANS)

\[
\{l_i:T_i \mid i \in 1..n+k\} <: \{l_i:T_i \mid i \in 1..n\}
\]

(S-RcdWidth)

\[
\text{for each } i \quad S_i <: T_i \\
\quad \{l_i:S_i \mid i \in 1..n\} <: \{l_i:T_i \mid i \in 1..n\}
\]

(S-RcdDepth)

\[
\{k_j:S_j \mid j \in 1..n\} \text{ is a permutation of } \{l_i:T_i \mid i \in 1..n\} \\
\quad \{k_j:S_j \mid j \in 1..n\} <: \{l_i:T_i \mid i \in 1..n\}
\]

(S-RcdPerm)

\[
T_1 <: S_1 \quad S_2 <: T_2 \\
\quad S_1 \rightarrow S_2 <: T_1 \rightarrow T_2
\]

(S-Arrow)

(S-Top)
Ascription as a derived form

- Someone asked to work exercise 11.4.1 part 2 today.
- But the solution is somewhat technical and would take too much time to discuss in detail.
- This exercise is not needed for the exam.
The Hints, again

- The exam will potentially cover everything in the course so far, but will focus on material we’ve seen since the first midterm.
- There will be a question that is also a one-star exercise from the book.
- There will be a question similar to problem 6 from midterm 1 ("Which properties remain true if we change one of the type systems we’ve studied in the following way...?")
- There will be (at least) one question based on one of the proofs in chapter 15.
- For PhD students, there will be a question involving subtyping and references.