Discussion 4

Discussion 4 - Thursday, July 10th

Reminders

1. Quiz 4 on **July 15th**. All usual rules apply, logistics post soon
2. Project 4 released, due **Wednesday, July 16th @ 11:59pm**.

Operational Semantics

Problem 1:

Using the rules below, show: $1 + (2 + 3) \Longrightarrow \; 6$

$$\frac{}{n \Longrightarrow n} \qquad \frac{e_1 \Longrightarrow n_1 \quad e_2 \Longrightarrow n_2 \quad n_3 \text{ is } n_1 + n_2}{e_1 + e_2 \Longrightarrow n_3}$$

Problem 2:

Using the rules given below, show: $A; \textbf{let } y = 1 \textbf{ in let } x = 2 \textbf{ in } x \Longrightarrow \; 2$

$$\frac{}{A;\, n \Longrightarrow n} \qquad \frac{A(x) = v}{A;\, x \Longrightarrow v} \qquad \frac{A;\, e_1 \Longrightarrow v_1 \quad A, x : v_1;\, e_2 \Longrightarrow v_2}{A;\, \textbf{let } x = e_1 \textbf{ in } e_2 \Longrightarrow v_2} \qquad \frac{A;\, e_1 \Longrightarrow n_1 \quad A;\, e_2 \Longrightarrow n_2 \quad n_3 \text{ is } n_1 + n_2}{A;\, e_1 + e_2 \Longrightarrow n_3}$$

Problem 3:

Using the rules given below, show: $A; \textbf{let } x = 3 \textbf{ in let } x = x + 6 \textbf{ in } x \Longrightarrow \; 9$

$$\frac{}{A;\, n \Longrightarrow n} \qquad \frac{A(x) = v}{A;\, x \Longrightarrow v} \qquad \frac{A;\, e_1 \Longrightarrow v_1 \quad A, x : v_1;\, e_2 \Longrightarrow v_2}{A;\, \textbf{let } x = e_1 \textbf{ in } e_2 \Longrightarrow v_2} \qquad \frac{A;\, e_1 \Longrightarrow n_1 \quad A;\, e_2 \Longrightarrow n_2 \quad n_3 \text{ is } n_1 + n_2}{A;\, e_1 + e_2 \Longrightarrow n_3}$$

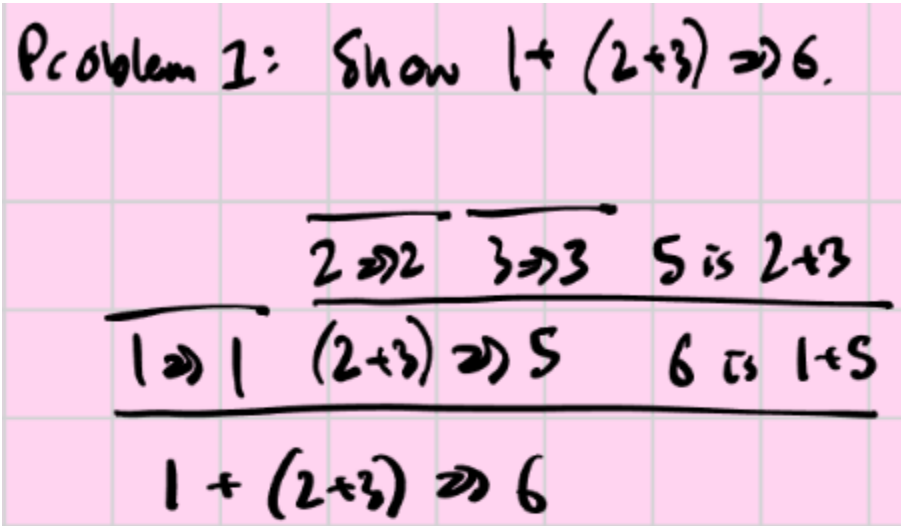
Problem 4:

Note: This problem takes a long time, I recommend completing in your own time!

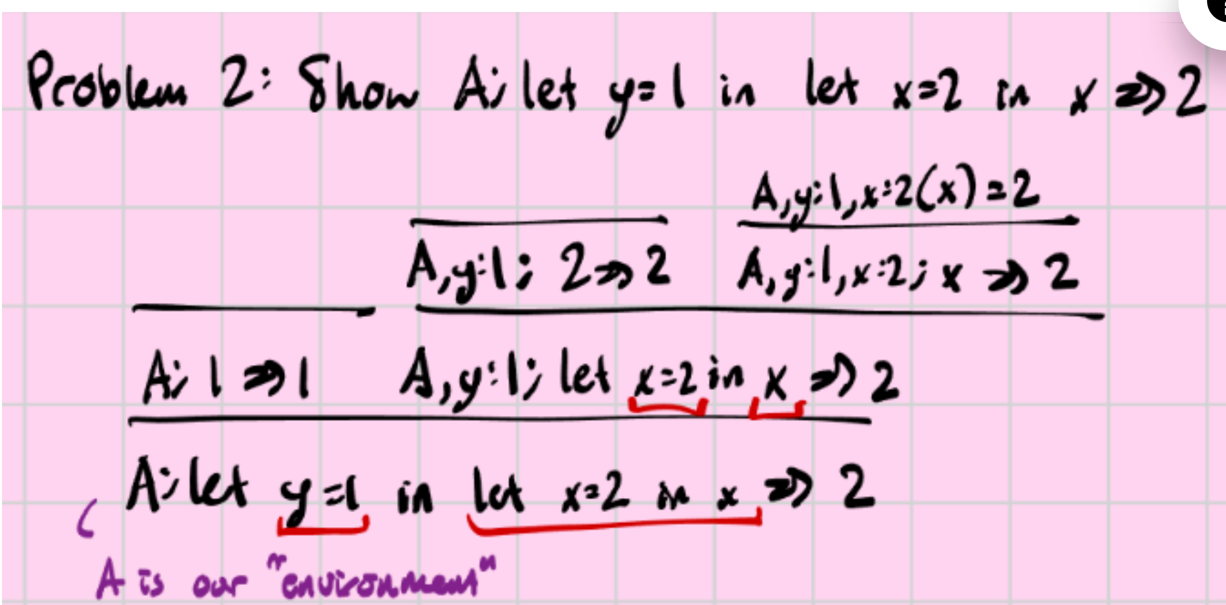
Using the rules given below, show:
$A; \textbf{let } x = 2 \textbf{ in let } y = 3 \textbf{ in let } x = x + 2 \textbf{ in } x + y \Longrightarrow \; 7$

$$\frac{}{A;\, n \Longrightarrow n} \qquad \frac{A(x) = v}{A;\, x \Longrightarrow v} \qquad \frac{A;\, e_1 \Longrightarrow v_1 \quad A,\, x : v_1;\, e_2 \Longrightarrow v_2}{A;\, \textbf{let}\; x = e_1 \;\textbf{in}\; e_2 \Longrightarrow v_2} \qquad \frac{A;\, e_1 \Longrightarrow n_1 \quad A;\, e_2 \Longrightarrow n_2 \quad n_3 \text{ is } n_1 + n_2}{A;\, e_1 + e_2 \Longrightarrow n_3}$$

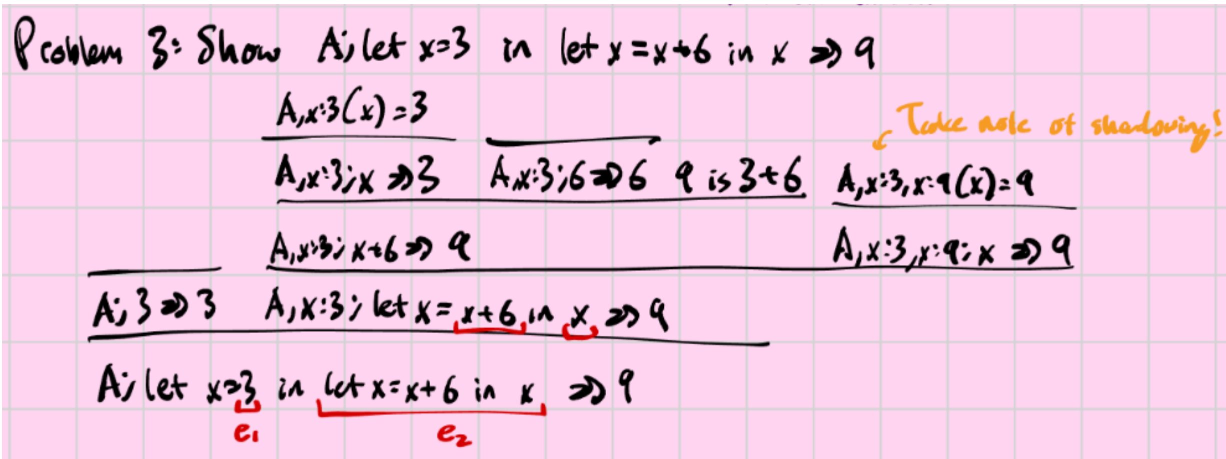
▼ Solutions!



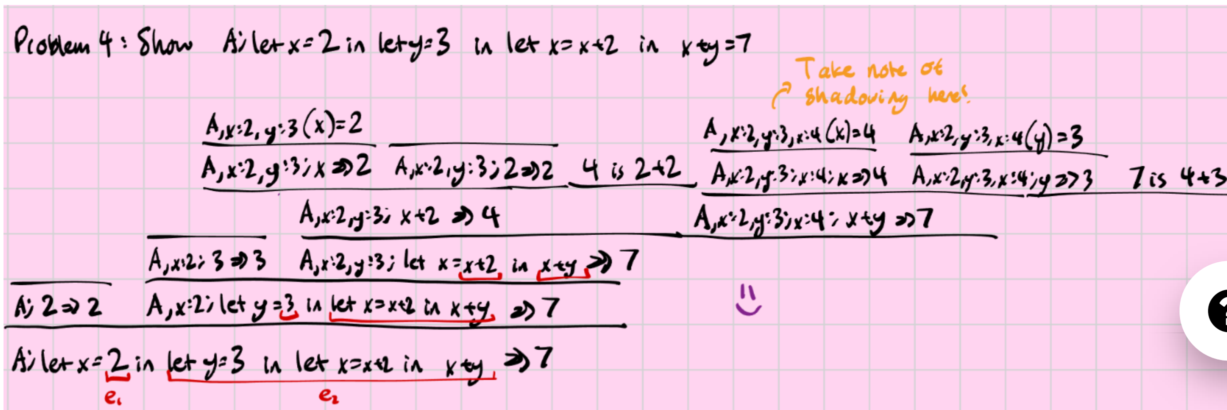
image



image



image



image

Type Checking

$$\frac{}{G \vdash x : G(x)} \qquad \frac{}{G \vdash \text{true} : \text{bool}} \qquad \frac{}{G \vdash \text{false} : \text{bool}} \qquad \frac{}{G \vdash n : \text{int}}$$

$$\frac{G \vdash e1 : \text{t1} \quad G, x : \text{t1} \vdash e2 : \text{t2}}{G \vdash \text{let } x = e1 \text{ in } e2 : t2} \qquad \frac{G \vdash e1 : \text{bool} \quad G \vdash e2 : \text{bool}}{G \vdash e1 \text{ and } e2 : \text{bool}}$$

$$\frac{G \vdash e : \text{int}}{G \vdash \text{eq0 } e : \text{bool}} \qquad \frac{G \vdash e1 : \text{bool} \quad G \vdash e2 : \text{t} \quad G \vdash e3 : \text{t}}{G \vdash \text{if } e1 \text{ then } e2 \text{ else } e3 : \text{t}}$$

Using the rules given above, show that the following statements are **well typed**: 1. `eq0 if true then 0 else 1` 2. `let x = 5 in eq0 x and false`

▼ Solutions!

1.

$$\dfrac{\dfrac{\overline{G \vdash \text{true} : \text{bool}} \quad \overline{G \vdash 0 : \text{int}} \quad \overline{G \vdash 1 : \text{int}}}{G \vdash \text{if true then 0 else 1} : \text{int}}}{G \vdash \text{eq0 if true then 0 else 1} : \text{bool}}$$

image

2.

$$\dfrac{\overline{G \vdash 5 : \text{int}} \quad \dfrac{\dfrac{\dfrac{\overline{G, x : \text{int}\ (x) = \text{int}}}{G, x : \text{int} \vdash x : \text{int}}}{G, x : \text{int} \vdash \text{eq0}\ x : \text{bool}} \quad \overline{G, x : \text{int} \vdash \text{false} : \text{bool}}}{G, x : \text{int} \vdash \text{eq0}\ x \text{ and false} : \text{bool}}}{G \vdash \text{let } x = 5 \text{ in eq0 } x \text{ and false} : \text{bool}}$$

image

Type Inference

There is 2 parts of type inference that we care about: - constraint construction - constraint solving

Constraint construction problems

$$\frac{}{G \vdash \text{true} : bool \dashv} \qquad \frac{}{G \vdash \text{false} : bool \dashv} \qquad \frac{}{G \vdash n : int \dashv}$$

$$\frac{G \vdash e_1 : t_1 \dashv C_1 \quad G \vdash e_2 : t_2 \dashv C_2}{G \vdash e_1 + e_2 : int \dashv t_1{:}t_2,\ t_1{:}int \cup C_1 \cup C_2}$$

$$\frac{G \vdash e_1 : t_1 \dashv C_1 \quad G \vdash e_2 : t_2 \dashv C_2}{G \vdash e_1\ \&\&\ e_2 : bool \dashv t_1{:}t_2,\ t_1{:}bool \cup C_1 \cup C_2}$$

$$\frac{G \vdash e : t_1 \dashv C}{G \vdash \text{eq0}\ e : bool \dashv t_1{:}int \cup C}$$

$$\frac{G \vdash e_1 : t_1 \dashv C_1 \quad G \vdash e_2 : t_2 \dashv C_2 \quad G \vdash e_3 : t_3 \dashv C_3}{G \vdash \text{if } e_1 \text{ then } e_2 \text{ else } e_3 : t_2 \dashv t_1{:}bool,\ t_2{:}t_3 \cup C1 \cup C_2 \cup}$$

Use the above rules to show the type inference proof for the following: 1. `true && eq0 5` 2. `true + 4` 3. `if true then 1 + 2 else 5` 4. `if true && false then 1 + 2 else eq0 0`

▼ Solutions!

1. $$\dfrac{\dfrac{}{G \vdash true : bool \dashv} \qquad \dfrac{\dfrac{}{G \vdash 5 : int \dashv}}{G \vdash \text{eq0}\ 5 : bool \dashv int{:}int}}{G \dashv true\ \&\&\ \text{eq0}\ 5 : bool \dashv bool{:}bool,\ bool{:}bool,\ int{:}int}$$

   - proof:

$$\dfrac{\dfrac{}{G \vdash true : t_1 \dashv} \qquad \dfrac{\dfrac{}{G \vdash 5 : t_3 \dashv}}{G \vdash \text{eq0}\ 5 : t_2 \dashv t_3{:}int}}{G \dashv true\ \&\&\ \text{eq0}\ 5 : bool \dashv t_1{:}bool,\ t_2{:}bool,\ t_3{:}int}$$

   where
   - t_1 = bool
   - t_2 = bool
   - t_3 = int

2. INVALID - $$\dfrac{\dfrac{}{G \vdash true : bool \dashv} \quad \dfrac{}{G \vdash 4 : int \dashv}}{G \vdash true + 4 : bool \dashv bool{:}int}$$

 - proof:

$$\dfrac{\dfrac{}{G \vdash true : t_1 \dashv} \quad \dfrac{\dfrac{}{G \vdash 5 : t_3 \dashv}}{G \vdash \text{eq0}\ 5 : t_2 \dashv t_3{:}int}}{G \dashv true\ \&\&\ \text{eq0}\ 5 : bool \dashv t_1{:}bool,\ t_2{:}bool,\ t_3{:}int}$$

   where
 - t_1 = bool
 - t_2 = int
 - $bool : int, bool : int$ is a contradiction, hence invalid

3. $$\dfrac{\dfrac{}{G \vdash true : bool \dashv} \quad \dfrac{\dfrac{}{G \vdash 1 : int \dashv} \quad \dfrac{}{G \vdash 2 : int \dashv}}{G \vdash 1 + 2 : int \dashv int{:}int} \quad \dfrac{}{G \vdash 5 : int \dashv}}{G \vdash \text{if } true \text{ then } 1 + 2 \text{ else } 5 : int \dashv int{:}int,\ bool{:}bool}$$

 - proof:

$$\dfrac{\dfrac{}{G \vdash true : t_1 \dashv} \quad \dfrac{\dfrac{}{G \vdash 1 : t_4 \dashv} \quad \dfrac{}{G \vdash 2 : t_5 \dashv}}{G \vdash 1 + 2 : t_2 \dashv t_4{:}t_5, t_4{:}int} \quad \dfrac{}{G \vdash 5 : t_3 \dashv}}{G \vdash \text{if } true \text{ then } 1 + 2 \text{ else } 5 : int \dashv t1{:}bool, t_2{:}t_3, t_4{:}t_5, t_4{:}int}$$

   where
 - t_1 = bool
 - t_2 = int
 - t_3 = int
 - t_4 = int
 - t_5 = int

4. INVALID -

$$\dfrac{\dfrac{\dfrac{}{G \vdash true : bool \dashv} \quad \dfrac{}{G \vdash false : bool \dashv}}{G \vdash true\ \&\&\ false : bool \dashv bool{:}bool} \quad \dfrac{\dfrac{}{G \vdash 1 : int \dashv} \quad \dfrac{}{G \vdash 2 : int \dashv}}{G \vdash 1 + 2 : int \dashv int{:}int} \quad \dfrac{\dfrac{}{G \vdash 0 : int \dashv}}{G \vdash eq0\ 0 : bool \dashv int{:}int}}{G \vdash \text{if } true\ \&\&\ false \text{ then } 1 + 2 \text{ else } eq0\ 0 : int \dashv bool{:}bool,\ int{:}int,\ int{:}bool}$$

- proof:

$$\dfrac{\dfrac{}{G \vdash true : t_4 \dashv} \quad \dfrac{}{G \vdash false : t_5 \dashv}}{G \vdash true \mathbin{\&\&} false : t_1 \dashv t_4{:}t_5, t_4{:}bool} \quad \dfrac{\dfrac{}{G \vdash 1 : t_6 \dashv} \quad \dfrac{}{G \vdash 2 : t_7 \dashv}}{G \vdash 1 + 2 : t_2 \dashv t_6{:}t_7, t_6{:}int} \quad \dfrac{\dfrac{}{G \vdash 0 : t_8 \dashv}}{G \vdash eq0\ 0 : t_3 \dashv t_8{:}int}$$
$$\overline{G \vdash \text{if } true \mathbin{\&\&} false \text{ then } 1 + 2 \text{ else } eq0\ 0 : int \dashv t_1{:}bool,\ t_2{:}t_3,\ t_4{:}t_5, t_4{:}bool, t_6{:}t_7, t_6{:}int, t_8{:}int}$$

where

- t_1 = bool
- t_2 = int
- t_3 = bool
- t_4 = bool
- t_5 = bool
- t_6 = int
- t_7 = int
- t_8 = int
- t_2= $t_2 = int \wedge t_3 = bool \wedge t_2 : t_3 \Rightarrow bool : int$ which is a contradiction

Now let's add variables and an unknown type: $\dfrac{}{G \vdash read\ () : t \dashv}$ (read returns some unknown type)

$$\dfrac{}{G \vdash x : G(x)} \qquad \dfrac{G \vdash e_1 : t_1 \dashv C_1 \quad G, x{:}t_1 \vdash e_2 : t_2 \dashv C_2}{G \vdash \text{let } x = e_1 \text{ in } e_2 : t_2 \dashv C_1 \cup C_2}$$

Note: this does not support shadowing of variables (behavior is more like binding a mutable variable)

1. `let x = read () in x + 1`
2. `let x = read () in let y = read () in x && y`
3. `let x = read () in if x then x else eq0 x`

▼ Solutions!

- $$\dfrac{\dfrac{}{G \vdash read\,() : t \dashv} \quad \dfrac{\dfrac{}{G, x{:}t \vdash x : G, x{:}t(x) \dashv} \quad \dfrac{}{G, x{:}t \vdash 1 : int \dashv}}{G, x{:}t_0 \vdash x + 1 : int \dashv t{:}int,\ int{:}int}}{G \vdash \text{let } x = read() \text{ in } x + 1 : int \dashv t{:}int,\ int{:}int}$$

  - proof:

$$\dfrac{\dfrac{}{G \vdash read():t_1 \dashv} \qquad \dfrac{\dfrac{}{G, x{:}t_1 \vdash x : G, x{:}t_1(x) \dashv} \qquad \dfrac{}{G, x{:}t_1 \vdash 1 : t_3 \dashv}}{G, x{:}t_1 \vdash x + 1 : t_2 \dashv t_1{:}int,\ t_3{:}int}}{G \vdash \text{let } x = read() \text{ in } x + 1 : t_2 \dashv t_1{:}int,\ t_3{:}int}$$

where

- t_1 = Unknown Type ('a)
- t_2 = int
- t_3 = int
- notice our constraints let us show that 'a = int!

- $$\dfrac{\dfrac{}{G \vdash read():t_1 \dashv} \qquad \dfrac{\dfrac{}{G, x{:}t_1 \vdash read():t_2 \dashv} \qquad \dfrac{\dfrac{}{G, y{:}t_2 \vdash x : G, y{:}t_2(x) \dashv} \qquad \dfrac{}{G, y{:}t_2 \vdash y : G, y{:}t_2(y) \dashv}}{G, y{:}t_2 \vdash x \,\&\&\, y : bool \dashv t_1{:}bool,\ t_2{:}bool}}{G, x{:}t_1 \vdash \text{let } y = read() \text{ in } x \,\&\&\, y : bool \dashv t_1{:}bool,\ t_2{:}bool}}{G \vdash \text{let } x = read() \text{ in let } y = read() \text{ in } x \,\&\&\, y : bool \dashv t_1{:}bool,\ t_2{:}bool}$$
  - proof

$$\dfrac{\dfrac{}{G \vdash read():t_1 \dashv} \qquad \dfrac{\dfrac{}{G, x{:}t_1 \vdash read():t_2 \dashv} \qquad \dfrac{\dfrac{}{G, y{:}t_2 \vdash x : G, y{:}t_2(x) \dashv} \qquad \dfrac{}{G, y{:}t_2 \vdash y : G, y{:}t_2(y) \dashv}}{G, y{:}t_2 \vdash x \,\&\&\, y : t_3 \dashv t_1{:}bool,\ t_2{:}bool}}{G, x{:}t_1 \vdash \text{let } y = read() \text{ in } x \,\&\&\, y : t_3 \dashv t_1{:}bool,\ t_2{:}bool}}{G \vdash \text{let } x = read() \text{ in let } y = read() \text{ in } x \,\&\&\, y : t_3 \dashv t_1{:}bool,\ t_2{:}bool}$$

where

- t_1 = Unknown Type ('a)
- t_2 = Unknown Type ('b)
- t_3 = bool
- notice our constraints let us show that ‘a = bool, and ’b = bool’!

- INVALID -

$$\dfrac{\dfrac{}{G \vdash read():t \dashv} \qquad \dfrac{\dfrac{}{G, x{:}t \vdash x : G, x{:}t(x) \dashv} \qquad \dfrac{}{G, x{:}t \vdash x : G, x{:}t(x) \dashv} \qquad \dfrac{\dfrac{}{G, x{:}t \vdash x : G, x{:}t(x) \dashv}}{G, x{:}t \vdash eq0\ x : bool \dashv t{:}int}}{G, x{:}t \vdash \text{if } x \text{ then } x \text{ else } eq0\ x : t \dashv t{:}int,\ t{:}bool}}{G \vdash \text{let } x = read() \text{ in if } x \text{ then } x \text{ else } eq0\ x : t \dashv t{:}int,\ t{:}bool}$$

- proof:

$$\dfrac{\dfrac{}{G \vdash read():t_1 \dashv} \qquad \dfrac{\dfrac{}{G, x{:}t_1 \vdash x : G, x{:}t_1(x) \dashv} \qquad \dfrac{}{G, x{:}t_1 \vdash x : G, x{:}t_1(x) \dashv} \qquad \dfrac{\dfrac{}{G, x{:}t_1 \vdash x : G, x{:}t_1(x) \dashv}}{G, x{:}t_1 \vdash eq0\ x : t_2 \dashv t_1{:}int}}{G, x{:}t \vdash \text{if } x \text{ then } x \text{ else } eq0\ x : t_1 \dashv t_1{:}bool, t_1{:}t_2, t_1{:}int}}{G \vdash \text{let } x = read() \text{ in if } x \text{ then } x \text{ else } eq0\ x : t_1 \dashv t_1{:}bool, t_1{:}t_2, t_1{:}int}$$

where

- t_1 = Unknown Type ('a))
- t_2 = bool (Notice our constraints show a contradiction: $t_1 : bool \land t_1 : int \Rightarrow bool : int$

Constraint Solving

Indicate if each of the following constraint sets contain any contradictions. (Maybe show unification steps if needed. idk.)

1. {Int:Int}
2. {Bool:Int}
3. {Int:Int, Bool:Bool, Int:Int}
4. {Bool:Bool, Int:Bool}
5. {‘a:Int}
6. {’a:Bool, ’a:’a}
7. {’a:’b, ’a:bool, ’b:bool}
8. {’a:’b, ’a:bool, ’b:’c}
9. {’a:’b, ’b:’c, ’a:’d}
10. {’a:’b, ’a: int, ’b:int’, ’c:’b, ’c:bool}
11. {’a:’b, ’b:int, ’a:’c, ’c:’a, ’c:’d, ’d:bool, ’c:int}

Unification steps

Go through the contraints, and take a contraint pair c and do the following - if c is x:x (like int:int, or ’a:’a), remove from constraints - if c is x:type or type:x (like ’a:int or int:’a), remove from constraints and edit all the other constraints in the set by replacing x with type - if c is type_1:type_2 (like int:bool), then stop and say contradiction, inference fails

Do the above until the set is empty or error occurs

▾ Solutions!

1. No contradictions
2. Yes, Bool $\neq$ Int
3. No contradictions
4. Yes, Bool $\neq$ Int
5. No contradictions
6. No contradictions
7. No contradictions
8. No contradictions
9. No contradictions
10. Yes, $'a = int =' b =' c = bool \Rightarrow Bool = Int$
11. Yes

Subtyping

Definition:

If `S` is a subtype of `T`, written S <: T,then an `S` can be used anywhere a `T` is expected.

`int` <: `int` `int` is not a subtype of `bool`

In the context of subtyping records, we can take a more liberal view of record types, seeing a record as “the set of all records with *at least* a field [name] of type [type]”. This means that the “smaller” subtype is actually the one with more fields in the record.

`{x:int, y:int, z:bool}` <: `{x:int, y:int}` `{y:int}` <: `{}` `{x:int}` is not a subtype of `{y:int}`

Also, the types within the records can contain subtypes. A record `A` is a subtype of record `B` if all of the variables within `A` are subtypes of equivalently named variables in `B`.

`{x:{a:int, b:int}, y:{m:int}}` <: `{x:{a:int}, y}`

Properties of subtyping:

- Reflexivity: for any type `A`, `A` <: `A`
- Transitivity: if `A` <: `B` <: `C`, then `A` <: `C`
- Permutation: components of a record are not ordered
  - `{x:a, y:b}` <: `{x:b, y:a}`
  - `{x:b, y:a}` <: `{x:a, y:b}`

For more information about subtyping, take a look at the subtyping information from this **past 330 type checking project (https://github.com/cmsc330spring24/cmsc330spring24/blob/main/projects/project5.md)**.

A different type of opsem problem

Another one like this is available on **Exam 2 Spring 2024 (https://bakalian.cs.umd.edu/assets/past_assignments/spring24e2sols.pdf)**, the one below is taken from **Exam 2 Fall 2024 (https://bakalian.cs.umd.edu/assets/past_assignments/fall24e2solns.pdf)**.

You are given two sets of rules for two different langauges.

These rules are valid for both languages:

$$\frac{}{\text{true} \rightarrow \text{true}} \qquad \frac{}{\text{false} \rightarrow \text{false}} \qquad \frac{A(x) = v}{A; x \rightarrow v}$$

These two rules are valid *only for language **1***:

$$\frac{A;e_1 \to v_1 \qquad A;e_2 \to v_2 \qquad v_3 = \text{if } v_1 \text{ then not } v_2 \text{ else } v_2}{A;e_1\ e_2\ \text{op1} \to v_3}$$

$$\frac{A;e_1 \to v_1 \qquad A,x{:}v_1;e_2 \to v_2}{A;\text{let } x = e_1 \text{ in } e_2 \to v_2}$$

These two rules are valid *only for language **2***:

$$\frac{A;e_1 \to v_1 \qquad A;e_2 \to v_2 \qquad v_3 = \text{if } v_1 \text{ then not } v_2 \text{ else } v_2}{A;\text{op2}\ e_2\ e_1 \to v_3}$$

$$\frac{A;e_2 \to v_1 \qquad A,x{:}v_1;e_1 \to v_2}{A;(\text{fun } x \to e_1)\ e_2 \to v_2}$$

Convert the following Language 1 sentence to its language 2 counterpart

```
A; let x = true in (true x op1)
```

How do we approach this sort of problem?

First, figure out what the sentence *means* - what is it actually doing?

In this case, Language 1 says that $A;\ let\ x\ =\ true\ in\ true\ x\ \text{op1}$ tells us to use this rule first, with $x\ =\ x$, $e1\ = true$, and $e2\ =\ true\ x\ \text{op1}$:

$$\frac{A;e_1 \to v_1 \qquad A,x{:}v_1;e_2 \to v_2}{A;\text{let } x = e_1 \text{ in } e_2 \to v_2}$$

Recall that the bottom half of this is just a representation of what the sentence looks like in the language. The top half is what the language *means*.

Which rule in lanugage 2 has the same *meaning* as that one? This one looks very similar…

$$\frac{A; e_2 \rightarrow v_1 \qquad A, x{:}v_1; e_1 \rightarrow v_2}{A; (\text{fun } x \rightarrow e_1)\ e_2 \rightarrow v_2}$$

▸ So, can we just plug in e1 for e1 and e2 for e2?

Given that, we can convert like this:

$A;\ let\ x\ =\ true\ in\ true\ x\ \mathbf{op1}$

to

$A;\ (\mathbf{fun}\ x \rightarrow true\ x\ \mathbf{op1})\ true \rightarrow v_2$

Okay, now we're closer, but $true\ x\ \mathbf{op1}$ is still a Language 1 construct, so we need convert that part as well.

What does $true\ x\ \mathbf{op1}$ mean in Language 1?

$$\frac{A; e_1 \rightarrow v_1 \qquad A; e_2 \rightarrow v_2 \qquad v_3 = \text{if } v_1 \text{ then not } v_2 \text{ else } v_2}{A; e_1\ e_2\ \text{op1} \rightarrow v_3}$$

With $e1\ = true$, $e2 = x$. What's the matching Language 2 rule?

$$\frac{A; e_1 \rightarrow v_1 \qquad A; e_2 \rightarrow v_2 \qquad v_3 = \text{if } v_1 \text{ then not } v_2 \text{ else } v_2}{A; \text{op2}\ e_2\ e_1 \rightarrow v_3}$$

▸ Here, can we just plug in e1 for e1 and e2 for e2?

So Language 1's $true\ x\ \mathbf{op1}$ becomes Language 2's $\mathbf{op2}\ x\ true$

Let's put it all together!

▸ Solution!

Additional Readings & Resources

- Professor Mamat's Type Checking Slides (https://bakalian.cs.umd.edu/assets/slides/19-Typechecking.pdf)
- Type Checker Problem Generator (https://bakalian.cs.umd.edu/330/practice/typechecker)
- Subtyping Reference from TAPL (https://www.cs.umd.edu/class/spring2024/cmsc330-030X-040X/assets/slides/TAPL_Ch._15.pdf)
- Professor Mamat's Operational Semantics Slides (https://bakalian.cs.umd.edu/assets/slides/17-semantics.pdf)
- Fall 2022 - Discussion 10 (Operational Semantics) (https://github.com/umd-cmsc330/fall2022/tree/main/discussions/discussion10#operational-semantics)
- OpSem Problem Generator (https://bakalian.cs.umd.edu/330/practice/opsem)