Subtyping!

Why now?

I wanted to talk about bidirectional typing, and subtyping was necessary or at least makes things simpler.

Purpose: Records

```plaintext
& name : string ; age : int3 ; > & name : string ; age : int ; sin : int3
( "hi" , 4 , 123 )
```
Purpose: Polymorphism without annotations

\((\lambda f : \text{int} \to \text{int}, f \circ 0) \ (\lambda X \ (x : X), x)\)

doesn't work

\((\lambda f : \text{int} \to \text{int}, f \circ 0) \ (\lambda X \ (x : X), x) [\text{int}]\)

does

But this is silly, as \(\forall X. X \to X\) should always work as \(\text{int} \to \text{int}\)

We will introduce \(F_{\text{c}}\), a type system for System \(F\) with subtyping
Recall that we can think of types as sets of values contained in them.

We can think of subtyping as subsetting.

We add two new types:

\[ \mathcal{X} = \ldots \top \]

\[ \mathcal{I}, \mathcal{C} : \mathcal{X} \]

Means

\[ \text{Set}(\mathcal{X}_1) \subseteq \text{Set}(\mathcal{X}_2) \]
Note, subtyping is contravariant in the first argument.

Examples:

\[
\Gamma + \text{true} \subseteq \Gamma
\]

\[
\Gamma + Y \Rightarrow Y_c : Y \Rightarrow Y
\]

\[
\Pi : X \Rightarrow X_c : Y \Rightarrow Y
\]
Theorem: If $\Gamma \vdash \tau_1 : \tau_2$ and $\Gamma + \tau_2 : \tau_3$, then $\Gamma + \tau_1 : \tau_3$

Proof: Left as an exercise to the reader

Now what are the rules for Fe?
\[ \Gamma \vdash \alpha \quad \Gamma \vdash \alpha' \]

**Thm: Progress:** If \( \Gamma \vdash \alpha \) then \( \exists \alpha' \) or \( \exists \) \( e \) value

**Thm: Preservation:** If \( \Gamma \vdash \alpha \) and \( \exists \alpha' \) then \( \Gamma \vdash \alpha \)

**Issues:**
\[ \Gamma \vdash \alpha \quad \Gamma \vdash \alpha' \]

\[ \Gamma \vdash \alpha' \]

Look at this non-determinism! It comes out of nowhere.

Can we do it automatically?
Actually still no!
Even without full erasure of types, this problem is also undecidable.

What about subtyping \textbf{NOT} in System $F$?

Yes!

Fundamentally, it is the universals causing issues.

Record STLC$_c$:

\[ e = \lambda x. e \]
\[ e_2 e_1 \]
\[ e_1 : e_1 \cdot \cdots \cdot e_n = e_n \]

\[ \gamma : \text{field}_1 \leq \cdots \leq \text{field}_n = \mathbb{R}^3 \]

\[ \gamma : \mathbb{R} \leq \cdots \leq \mathbb{R}^3 \]

\[ \gamma : \mathbb{R}_1 \leq \mathbb{R}_2 \]

\[ \Gamma \left( \mathbb{R}_1 + \mathbb{R}_2 \right) \]

\[ \Gamma \left( \mathbb{R}_1 \oplus \mathbb{R}_2 \right) : \mathbb{R}_1 \rightarrow \mathbb{R}_2 \]
\[ \exists ! \tau : \exists ! \tau \quad \tau_1 : \tau_2 \]
\[ \Gamma \vdash e : \tau \] algorithmic typing

\[ \Gamma, x : \tau \vdash x : \tau \]

\[ \Gamma \vdash e_1 : \tau_1 \]

\[ \vdots \]

\[ \Gamma \vdash e_n : \tau_n \]

\[ \Gamma \vdash f_1 = e_1, \ldots, f_n = e_n : \{ f_1 : \tau_1, \ldots, f_n : \tau_n \} \]

\[ \Gamma \vdash e_2 : \tau_1 \rightarrow \tau_2 \]

\[ \Gamma \vdash e_1 : \tau_1' \]