

Type checker for System F

Mudathir Mahgoub

May 2, 2018

1 Project Problem

This project is a type checker for annotated simple typed lambda calculus and system F. It supports subtyping for simple types. In general, a type checker would decide whether $\Gamma \vdash t : T$ is derivable: can the term t be assigned the type T under the typing context Γ ?

Section 2 describes the software and its architecture. Section 3 describes the rules used for simple types and provides some examples. It also explains how subsumption rule can be delayed until variable terms are generated, and provides a proof for correctness. Section 3 describes the rules used for system F and provides some examples.

2 Software description

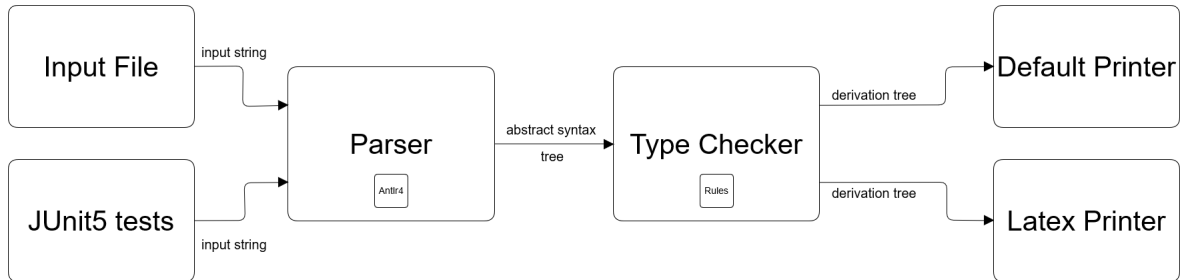


Figure 1: Project architecture.

The project is implemented using Java and the executable is a jar file (TypeChecker.jar). The program receives as an input a text file containing subtypes definitions and a judgment to be checked. For testing JUnit5 was used to test the program directly without files. Below is an example of an input:

Listing 1: test.txt

```
SubBase(bool , int );  
. |- \lambda x. \lambda y. (x y)[bool]: (int ->T) -> (bool -> T);
```

Listing 2: test.txt

```
Yes  
\begin{prooftree}  
\AxiomC{} \RightLabel{\scriptsize var}  
\UnaryInfC{$x: (\mathbf{int} \rightarrow T), y: \text{bool} \vdash x : (\mathbf{int} \rightarrow T)$}  
\AxiomC{\scriptsize $SubBase(\text{bool}, \mathbf{int})$}  
\RightLabel{\scriptsize subBase}  
\UnaryInfC{$\text{bool} <: \mathbf{int}$}  
\AxiomC{} \RightLabel{\scriptsize reflexive}  
\UnaryInfC{$T <: T$}  
\RightLabel{\scriptsize arrow}  
\BinaryInfC{$(\mathbf{int} \rightarrow T) <: (\text{bool} \rightarrow T)$}  
\RightLabel{\scriptsize subsumption}
```

```

\BinaryInfC{$x: (\mathbf{int} \rightarrow T), y: \text{bool} \vdash x : (\text{bool} \rightarrow T)$}
\AxiomC{} \RightLabel{\scriptsize var}
\UnaryInfC{$x: (\mathbf{int} \rightarrow T), y: \text{bool} \vdash y : \text{bool}$}
\RightLabel{\scriptsize app}
\BinaryInfC{$x: (\mathbf{int} \rightarrow T), y: \text{bool} \vdash (x\backslash y) [\text{bool}] : T$}
\RightLabel{$\lambda$}
\UnaryInfC{$x: (\mathbf{int} \rightarrow T) \vdash (\lambda y. (x\backslash y) [\text{bool}]) : (\text{bool} \rightarrow T)$}
\RightLabel{$\lambda$}
\UnaryInfC{$\vdash (\lambda x. (\lambda y. (x\backslash y) [\text{bool}])) : ((\mathbf{int} \rightarrow T) \rightarrow (\text{bool} \rightarrow T))$}
\end{prooftree}

```

3 Simple types

4 Rules

1. $\frac{\Gamma(x) = T}{\Gamma \vdash x : T} \text{ var}$
2. $\frac{\Gamma \vdash t_1 : T_1 \rightarrow T_2 \quad \Gamma \vdash t_2 : T_1}{\Gamma \vdash (t_1 t_2)[T_1] : T_2} \text{ app}$
3. $\frac{\Gamma, x : T_1 \vdash t : T_2}{\Gamma \vdash \lambda x. t : T_1 \rightarrow T_2} \lambda$
4. $\frac{\Gamma \vdash t : T_1 \quad T_1 <: T_2}{\Gamma \vdash t : T_2} \text{ subsumption}$
5. $\frac{}{T <: T} \text{ reflexive}$
6. $\frac{\text{SubBase}(b_1, b_2)}{b_1 <: b_2} \text{ subBase}$
7. $\frac{T'_1 <: T_1 \quad T_2 <: T'_2}{T_1 \rightarrow T_2 <: T'_1 \rightarrow T'_2} \text{ arrow}$
8. $\frac{T_1 <: T_2 \quad T_2 <: T_3}{T_1 <: T_3} \text{ transitive}$

5 Variable terms

5.1 Valid

$$\frac{}{x : T \vdash x : T} \text{ var}$$

5.2 Invalid

$$\frac{}{\cdot \vdash x : T} \text{ invalid var}$$

6 Lambda & application terms

$$\frac{\frac{}{x : (T_1 \rightarrow T_2), y : T_1 \vdash x : (T_1 \rightarrow T_2)} \text{ var} \quad \frac{}{x : (T_1 \rightarrow T_2), y : T_1 \vdash y : T_1} \text{ var}}{\frac{\frac{}{x : (T_1 \rightarrow T_2), y : T_1 \vdash (x y)[T_1] : T_2} \text{ app}}{x : (T_1 \rightarrow T_2) \vdash \lambda y. (x y)[T_1] : (T_1 \rightarrow T_2)} \lambda} \lambda$$

7 Direct Subtyping

7.1 Valid

$$\frac{\frac{}{x : \text{bool} \vdash x : \text{bool}} \text{var} \quad \frac{\text{SubBase}(\text{bool}, \text{int})}{\text{bool} <: \text{int}} \text{subBase}}{x : \text{bool} \vdash x : \text{int}} \text{subsumption}$$

7.2 Invalid

$$\frac{\frac{}{x : \text{int} \vdash x : \text{int}} \text{var} \quad \frac{\perp}{\text{int} <: \text{bool}} \text{invalid}}{x : \text{int} \vdash x : \text{bool}} \text{subsumption}$$

8 Transitive Subtyping

$$\frac{\frac{}{x : \text{bool} \vdash x : \text{bool}} \text{var} \quad \frac{\frac{\text{SubBase}(\text{bool}, \text{int})}{\text{bool} <: \text{int}} \text{subBase} \quad \frac{\text{SubBase}(\text{int}, \text{double})}{\text{int} <: \text{double}} \text{subBase}}{\text{bool} <: \text{double}} \text{transitive}}{x : \text{bool} \vdash x : \text{double}} \text{subsumption}$$

$$\frac{\frac{}{x : \text{bool} \vdash x : \text{bool}} \text{var} \quad \frac{\frac{\frac{\text{SubBase}(\text{bool}, \text{int})}{\text{bool} <: \text{int}} \text{subBase} \quad \frac{\text{SubBase}(\text{int}, \text{quotient})}{\text{int} <: \text{quotient}} \text{subBase}}{\text{bool} <: \text{quotient}} \text{transitive} \quad \frac{\text{SubBase}(\text{quotient}, \text{double})}{\text{quotient} <: \text{double}} \text{subBase}}{\text{bool} <: \text{double}} \text{transitive}}{x : \text{bool} \vdash x : \text{double}} \text{subsumption}$$

9 Arrow Types

9.1 Valid

$$\frac{\frac{}{x : (\text{int} \rightarrow \text{bool}) \vdash x : (\text{int} \rightarrow \text{bool})} \text{var} \quad \frac{\frac{\text{SubBase}(\text{bool}, \text{int})}{\text{bool} <: \text{int}} \text{subBase} \quad \frac{\text{SubBase}(\text{bool}, \text{int})}{\text{bool} <: \text{int}} \text{subBase}}{(\text{int} \rightarrow \text{bool}) <: (\text{bool} \rightarrow \text{int})} \text{arrow}}{x : (\text{int} \rightarrow \text{bool}) \vdash x : (\text{bool} \rightarrow \text{int})} \text{subsumption}$$

9.1.1 Reflexive

$$\frac{\frac{}{x : (\text{int} \rightarrow \text{bool}) \vdash x : (\text{int} \rightarrow \text{bool})} \text{var} \quad \frac{\frac{}{\text{int} <: \text{int}} \text{reflexive} \quad \frac{\text{SubBase}(\text{bool}, \text{int})}{\text{bool} <: \text{int}} \text{subBase}}{(\text{int} \rightarrow \text{bool}) <: (\text{int} \rightarrow \text{int})} \text{arrow}}{x : (\text{int} \rightarrow \text{bool}) \vdash x : (\text{int} \rightarrow \text{int})} \text{subsumption}$$

9.1.2 Nested

$$\frac{\frac{\frac{}{x : (\text{int} \rightarrow T), y : \text{bool} \vdash x : (\text{int} \rightarrow T)} \text{var} \quad \frac{\frac{\text{SubBase}(\text{bool}, \text{int})}{\text{bool} <: \text{int}} \text{subBase} \quad \frac{}{T <: T} \text{reflexive}}{(\text{int} \rightarrow T) <: (\text{bool} \rightarrow T)} \text{arrow}}{x : (\text{int} \rightarrow T), y : \text{bool} \vdash x : (\text{bool} \rightarrow T)} \text{subsumption} \quad \frac{}{x : (\text{int} \rightarrow T), y : \text{bool} \vdash y : \text{bool}} \text{var}}{x : (\text{int} \rightarrow T), y : \text{bool} \vdash (x y)[\text{bool}] : T} \text{app}$$

$$\frac{\frac{x : (\text{int} \rightarrow T), y : \text{bool} \vdash (x y)[\text{bool}] : T}{x : (\text{int} \rightarrow T) \vdash \lambda y. (x y)[\text{bool}] : (\text{bool} \rightarrow T)} \lambda}{\vdash \lambda x. \lambda y. (x y)[\text{bool}] : ((\text{int} \rightarrow T) \rightarrow (\text{bool} \rightarrow T))} \lambda$$

9.2 Invalid

$$\begin{array}{c}
\frac{}{x : (bool \rightarrow bool) \vdash x : (bool \rightarrow bool)} \text{var} \quad \frac{\frac{\perp}{int <: bool} \text{invalid} \quad \frac{SubBase(bool, int)}{bool <: int} \text{subBase}}{(bool \rightarrow bool) <: (int \rightarrow int)} \text{arrow} \\
\hline
x : (bool \rightarrow bool) \vdash x : (int \rightarrow int) \quad \text{subsumption}
\end{array}$$

$$\begin{array}{c}
\frac{}{x : (bool \rightarrow T), y : bool \vdash x : (bool \rightarrow T)} \text{var} \quad \frac{\frac{\perp}{int <: bool} \text{invalid} \quad \frac{T <: T}{T <: T} \text{reflexive}}{(bool \rightarrow T) <: (int \rightarrow T)} \text{arrow} \\
\hline
\frac{}{x : (bool \rightarrow T), y : bool \vdash x : (int \rightarrow T)} \text{subsumption} \quad \frac{}{x : (bool \rightarrow T), y : bool \vdash y : bool} \text{var} \quad \frac{SubBase(bool, int)}{bool <: int} \text{subBase} \\
\hline
\frac{}{x : (bool \rightarrow T), y : bool \vdash y : int} \text{app} \quad \frac{}{x : (bool \rightarrow T), y : bool \vdash (x y)[int] : T} \lambda \\
\hline
\frac{}{\vdash \lambda x. \lambda y. (x y)[int] : ((bool \rightarrow T) \rightarrow (bool \rightarrow T))} \lambda
\end{array}$$

9.3 Delaying applying the subsumption rule

The subsumption rule can be delayed until the term in the judgment is just a variable. This simplifies the code since there is only one rule to be applied for application and λ abstraction terms. Here is the correctness proof.

$$\frac{\Gamma \vdash t : T \quad T <: T'}{\Gamma \vdash t : T'} \text{subsumption}$$

Proof:

By induction hypothesis on the structure of the term.

1. Case $t = x$: trivial since t is a variable.

$$\frac{\Gamma \vdash x : T \quad T <: T'}{\Gamma \vdash x : T'} \text{subsumption}$$

2. Case $t = t_1 t_2$: assume $T_2 <: T'_2$ and the following derivation:

$$\frac{\frac{\Gamma \vdash t_1 : T_1 \rightarrow T_2 \quad \Gamma \vdash t_2 : T_1}{\Gamma \vdash t_1 t_2[T_1] : T_2} \text{app} \quad T_2 <: T'_2}{\Gamma \vdash t_1 t_2[T_1] : T'_2} \text{subsumption}$$

We can get a different derivation tree where the subsumption rule is delayed:

$$\frac{\frac{\Gamma \vdash t_1 t_2[T_1] : T_2 \quad \frac{T_1 <: T_1 \text{ reflexive} \quad T_2 <: T'_2}{T_1 \rightarrow T_2 <: T_1 \rightarrow T'_2} \text{arrow}}{\Gamma \vdash t_1 : T_1 \rightarrow T'_2} \text{subsumption} \quad \Gamma \vdash t_2 : T_1}{\Gamma \vdash t_1 t_2[T_1] : T'_2} \text{app}$$

By the induction hypothesis, the subsumption rule can be delayed until variable terms are generated in the derivation of $\Gamma \vdash t_1 : T_1 \rightarrow T'_2$ and $\Gamma \vdash t_2 : T_1$.

3. Case $t = \lambda x. t'$: assume $T_1 \rightarrow T_2 <: T'_1 \rightarrow T'_2$ and the following derivation:

$$\frac{\frac{\Gamma, x : T_1 \vdash t' : T_2}{\Gamma \vdash \lambda x. t' : T_1 \rightarrow T_2} \lambda \quad \frac{T'_1 <: T_1 \quad T_2 <: T'_2}{T_1 \rightarrow T_2 <: T'_1 \rightarrow T'_2} \text{arrow}}{\Gamma \vdash \lambda x. t' : T'_1 \rightarrow T'_2} \text{subsumption}$$

Alternatively we can derive:

$$\frac{\frac{\Gamma, x : T'_1 \vdash t' : T_2 \quad T_2 <: T'_2}{\Gamma, x : T'_1 \vdash t' : T'_2} \text{subsumption}}{\Gamma \vdash \lambda x. t' : T'_1 \rightarrow T'_2} \lambda$$

Since $T'_1 <: T_1$ then by the subsumption rule:

$$\frac{\frac{\Gamma, x : T'_1 \vdash x : T'_1}{\Gamma, x : T'_1 \vdash x : T_1} \text{var} \quad T'_1 <: T_1}{\Gamma, x : T'_1 \vdash x : T_1} \text{subsumption}$$

Therefore if $\Gamma, x : T_1 \vdash t' : T_2$ then $\Gamma, x : T'_1 \vdash t' : T_2$ which concludes the proof.

Note:

If $\Gamma, x : T'_1 \vdash t' : T_2$, it is not always true that $\Gamma, x : T_1 \vdash t' : T_2$ where $T'_1 <: T_1$. A counter example would be

$$x : \text{bool} \vdash x : \text{int} \not\Rightarrow x : \text{double} \vdash x : \text{int}, \quad \text{bool} <: \text{int} <: \text{double}$$

However

$$x : \text{int} \vdash x : \text{double} \Rightarrow x : \text{bool} \vdash x : \text{double}, \quad \text{bool} <: \text{int} <: \text{double}$$

Therefore in the following example, using the subsumption rule first would fail and slow the type checker because it needs to backtrack and check the λ rule. However using the subsumption rule would prove the type checking and is faster.

$$\frac{\frac{x : \text{double} \vdash x : \text{int}}{\Gamma \vdash \lambda x. x : \text{double} \rightarrow \text{int}} \lambda \quad \frac{\frac{\text{SubBase}(\text{bool}, \text{double})}{\text{bool} <: \text{double}} \text{subBase} \quad \frac{\text{int} <: \text{int}}{\text{int} <: \text{int}} \text{reflexive arrow}}{\text{double} \rightarrow \text{int} <: \text{bool} \rightarrow \text{int}} \text{subsumption}}{\Gamma \vdash \lambda x. x : \text{bool} \rightarrow \text{int}} \text{subsumption}$$

$$\frac{\frac{\Gamma, x : \text{bool} \vdash x : \text{bool}}{\Gamma, x : \text{bool} \vdash x : \text{int}} \text{var} \quad \frac{\text{SubBase}(\text{bool}, \text{int})}{\text{bool} <: \text{int}} \text{subBase}}{\Gamma, x : \text{bool} \vdash x : \text{int}} \text{subsumption} \quad \frac{}{\Gamma \vdash \lambda x. x : \text{bool} \rightarrow \text{int}} \lambda$$

10 System F

10.1 Variables

10.1.1 Valid

$$\frac{}{x : \forall X. X \vdash x : \forall X. X} \text{var}$$

$$\frac{}{x : \forall X. X \vdash x : \forall Y. Y} \text{var}$$

$$\frac{}{x : \forall X. (X \rightarrow X) \vdash x : \forall Y. (Y \rightarrow Y)} \text{var}$$

Y is free in the context and the type:

$$\begin{array}{c}
\frac{}{x : \forall X.(X \rightarrow Y) \vdash x : \forall Z.(Z \rightarrow Y)} \text{var} \\
\\
\frac{\frac{}{x : \forall X.X \vdash x : \forall X1.X1} \text{var}}{x : \forall X.X \vdash x[Y] : Y} \text{elimination} \\
\\
\frac{\frac{}{x : \forall X.X \vdash x : \forall X1.X1} \text{var}}{x : \forall X.X \vdash x[(Y \rightarrow Y)] : (Y \rightarrow Y)} \text{elimination} \\
\\
\frac{\frac{\frac{}{x : \forall X.X \vdash x : \forall X1.X1} \text{var}}{x : \forall X.X \vdash x[[\forall X.X \rightarrow \forall X.X]] : (\forall X.X \rightarrow \forall X.X)} \text{elimination}}{x : \forall X.X \vdash (x[[\forall X.X \rightarrow \forall X.X]])[\forall X.X] : \forall X.X} \text{var app}
\end{array}$$

10.1.2 Invalid

Y is free in the context:

$$\frac{}{x : \forall X.(X \rightarrow Y) \vdash x : \forall Y.(Y \rightarrow Y)} \text{invalid var}$$

Y is free in the context, and Z is free in the type:

$$\frac{}{x : \forall X.(X \rightarrow Y) \vdash x : \forall Y.(Y \rightarrow Z)} \text{invalid var}$$

$$\frac{\frac{}{x : \forall X.X \vdash x : \forall X1.X1} \text{var}}{x : \forall X.X \vdash x[Y] : Y} \text{elimination}$$

10.2 Numbers

Zero

$$\frac{\frac{\frac{}{z : X, s : (X \rightarrow X) \vdash z : X} \text{var}}{s : (X \rightarrow X) \vdash (\lambda z.z) : (X \rightarrow X)} \lambda}{\cdot \vdash (\lambda s.(\lambda z.z)) : ((X \rightarrow X) \rightarrow (X \rightarrow X))} \lambda \\
\frac{}{\cdot \vdash (\lambda s.(\lambda z.z)) : \forall X.((X \rightarrow X) \rightarrow (X \rightarrow X))} \text{introduction}$$

Zero with free variable X

$$\frac{\frac{\frac{\frac{}{y : X, z : X_2, s : (X_2 \rightarrow X_2) \vdash z : X_2} \text{var}}{y : X, s : (X_2 \rightarrow X_2) \vdash (\lambda z.z) : (X_2 \rightarrow X_2)} \lambda}{y : X \vdash (\lambda s.(\lambda z.z)) : ((X_2 \rightarrow X_2) \rightarrow (X_2 \rightarrow X_2))} \lambda}{y : X \vdash (\lambda s.(\lambda z.z)) : \forall X_2.((X_2 \rightarrow X_2) \rightarrow (X_2 \rightarrow X_2))} \text{introduction} \\
\frac{}{y : X \vdash (\lambda s.(\lambda z.z)) : \forall X.((X \rightarrow X) \rightarrow (X \rightarrow X))} \text{renaming}$$

