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

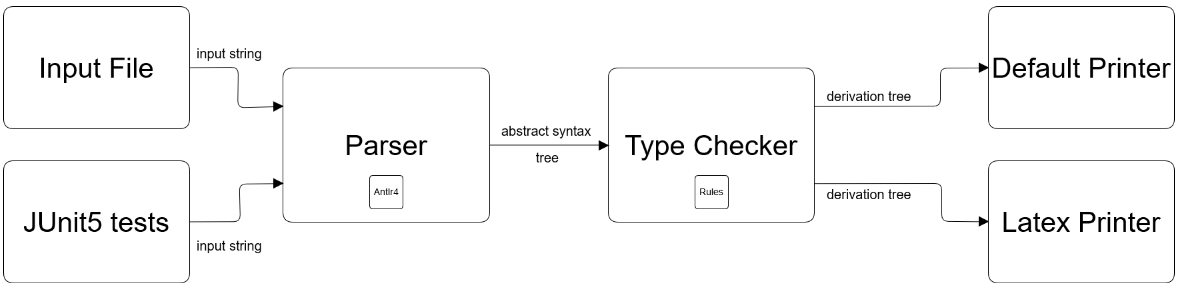

Figure 1: Project architecture.

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:

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

3 Simple types

4 Rules

$$1. \quad \frac{\Gamma(x) = T}{\Gamma \vdash x : T}\ \text{var}$$

2. $$\frac{\Gamma \vdash t_1 : T_1 \to T_2 \qquad \Gamma \vdash t_2 : T_1}{\Gamma \vdash (t_1 t_2)[T_1] : T_2}\ \text{app}$$

3. $$\frac{\Gamma, x : T_1 \vdash t : T_2}{\Gamma \vdash \lambda x.t : T_1 \to T_2}\ \lambda$$

4. $$\frac{\Gamma \vdash t : T_1 \qquad T_1 <: T_2}{\Gamma \vdash t : T_2}\ \text{subsumption}$$

5. $$\frac{}{T <: T}\ \text{reflexive}$$

6. $$\frac{SubBase(b_1, b_2)}{b_1 <: b_2}\ \text{subBase}$$

7. $$\frac{T_1' <: T_1 \qquad T_2 <: T_2'}{T_1 \to T_2 <: T2_1 \to T_2'}\ \text{arrow}$$

8. $$\frac{T_1 <: T_2 \qquad T_2 <: T_3}{T_1 <: T_3}\ \text{transitive}$$

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

$$\frac{\dfrac{\dfrac{}{x : (T1 \to T2), y : T1 \vdash x : (T1 \to T2)}\ \text{var} \qquad \dfrac{}{x : (T1 \to T2), y : T1 \vdash y : T1}\ \text{var}}{\dfrac{x : (T1 \to T2), y : T1 \vdash (x\ y)[T1] : T2}{x : (T1 \to T2) \vdash \lambda y.(x\ y)[T1] : (T1 \to T2)}\ \lambda}\ \text{app}}{\cdot \vdash \lambda x.\lambda y.(x\ y)[T1] : ((T1 \to T2) \to (T1 \to T2))}\ \lambda$$

7 Direct Subtyping

7.1 Valid

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

7.2 Invalid

$$\frac{\dfrac{}{x : int \vdash x : int}\ \text{var} \qquad \dfrac{\bot}{int <: bool}\ \text{invalid}}{x : int \vdash x : bool}\ \text{subsumption}$$

8 Transitive Subtyping

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

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

9 Arrow Types

9.1 Valid

$$\dfrac{\dfrac{}{x : (\mathit{int} \to \mathit{bool}) \vdash x : (\mathit{int} \to \mathit{bool})}\,\text{var} \qquad \dfrac{\dfrac{\text{SubBase}(\mathit{bool}, \mathit{int})}{\mathit{bool} <: \mathit{int}}\,\text{subBase} \qquad \dfrac{\text{SubBase}(\mathit{bool}, \mathit{int})}{\mathit{bool} <: \mathit{int}}\,\text{subBase}}{(\mathit{int} \to \mathit{bool}) <: (\mathit{bool} \to \mathit{int})}\,\text{arrow}}{x : (\mathit{int} \to \mathit{bool}) \vdash x : (\mathit{bool} \to \mathit{int})}\,\text{subsumption}$$

9.1.1 Reflexive

$$\dfrac{\dfrac{}{x : (\mathit{int} \to \mathit{bool}) \vdash x : (\mathit{int} \to \mathit{bool})}\,\text{var} \qquad \dfrac{\dfrac{}{\mathit{int} <: \mathit{int}}\,\text{reflexive} \qquad \dfrac{\text{SubBase}(\mathit{bool}, \mathit{int})}{\mathit{bool} <: \mathit{int}}\,\text{subBase}}{(\mathit{int} \to \mathit{bool}) <: (\mathit{int} \to \mathit{int})}\,\text{arrow}}{x : (\mathit{int} \to \mathit{bool}) \vdash x : (\mathit{int} \to \mathit{int})}\,\text{subsumption}$$

9.1.2 Nested

$$\dfrac{\dfrac{\dfrac{\dfrac{}{x : (\mathit{int} \to T), y : \mathit{bool} \vdash x : (\mathit{int} \to T)}\,\text{var} \qquad \dfrac{\dfrac{\text{SubBase}(\mathit{bool}, \mathit{int})}{\mathit{bool} <: \mathit{int}}\,\text{subBase} \qquad \dfrac{}{T <: T}\,\text{reflexive}}{(\mathit{int} \to T) <: (\mathit{bool} \to T)}\,\text{arrow}}{x : (\mathit{int} \to T), y : \mathit{bool} \vdash x : (\mathit{bool} \to T)}\,\text{subsumption} \qquad \dfrac{}{x : (\mathit{int} \to T), y : \mathit{bool} \vdash y : \mathit{bool}}\,\text{var}}{\dfrac{x : (\mathit{int} \to T), y : \mathit{bool} \vdash (x\ y)[\mathit{bool}] : T}{x : (\mathit{int} \to T) \vdash \lambda y.(x\ y)[\mathit{bool}] : (\mathit{bool} \to T)}\,\lambda}\,\text{app}}{\cdot \vdash \lambda x.\lambda y.(x\ y)[\mathit{bool}] : ((\mathit{int} \to T) \to (\mathit{bool} \to T))}\,\lambda$$

9.2 Invalid

$$\dfrac{\dfrac{}{x : (\mathit{bool} \to \mathit{bool}) \vdash x : (\mathit{bool} \to \mathit{bool})}\,\text{var} \qquad \dfrac{\dfrac{\bot}{\mathit{int} <: \mathit{bool}}\,\text{invalid} \qquad \dfrac{\text{SubBase}(\mathit{bool}, \mathit{int})}{\mathit{bool} <: \mathit{int}}\,\text{subBase}}{(\mathit{bool} \to \mathit{bool}) <: (\mathit{int} \to \mathit{int})}\,\text{arrow}}{x : (\mathit{bool} \to \mathit{bool}) \vdash x : (\mathit{int} \to \mathit{int})}\,\text{subsumption}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{}{x : (\mathit{bool} \to T), y : \mathit{bool} \vdash x : (\mathit{bool} \to T)}\,\text{var} \qquad \dfrac{\dfrac{\bot}{\mathit{int} <: \mathit{bool}}\,\text{invalid} \qquad \dfrac{}{T <: T}\,\text{reflexive}}{(\mathit{bool} \to T) <: (\mathit{int} \to T)}\,\text{arrow}}{x : (\mathit{bool} \to T), y : \mathit{bool} \vdash x : (\mathit{int} \to T)}\,\text{subsumption} \qquad \dfrac{\dfrac{}{x : (\mathit{bool} \to T), y : \mathit{bool} \vdash y : \mathit{bool}}\,\text{var} \qquad \dfrac{\text{SubBase}(\mathit{bool}, \mathit{int})}{\mathit{bool} <: \mathit{int}}\,\text{subBase}}{x : (\mathit{bool} \to T), y : \mathit{bool} \vdash y : \mathit{int}}\,\text{subsumption}}{\dfrac{x : (\mathit{bool} \to T), y : \mathit{bool} \vdash (x\ y)[\mathit{int}] : T}{x : (\mathit{bool} \to T) \vdash \lambda y.(x\ y)[\mathit{int}] : (\mathit{bool} \to T)}\,\lambda}\,\text{app}}{\cdot \vdash \lambda x.\lambda y.(x\ y)[\mathit{int}] : ((\mathit{bool} \to T) \to (\mathit{bool} \to T))}\,\lambda$$

9.3 Delaying applying the subsumption rule

The sumbumption rule can be delayed until the term in the judgment is a 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.

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

Proof:
By induction hypothesis on the structure of the term.

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

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

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

$$\frac{\dfrac{\Gamma \vdash t_1 : T_1 \to T_2 \qquad \Gamma \vdash t_2 : T_1}{\Gamma \vdash t_1 t_2[T_1] : T_2}\ \text{app} \qquad T_2 <: T_2'}{\Gamma \vdash t_1 t_2[T_1] : T_2'}\ \text{subsumption}$$

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

$$\frac{\dfrac{\Gamma \vdash t_1 t_2[T_1] : T_2 \qquad \dfrac{\dfrac{}{T_1 <: T_1}\ \text{reflexive} \qquad T_2 <: T_2'}{T_1 \to T_2 <: T_1 \to T_2'}\ \text{arrow}}{\Gamma \vdash t_1 : T_1 \to T_2'}\ \text{subsumption} \qquad \Gamma \vdash t_2 : T_1}{\Gamma \vdash t_1 t_2[T_1] : T_2'}\ \text{app}$$

By the induction hypothesis, the subsumption rule can be delayed until variable terms are generated in the derivation of $\Gamma \vdash t_1 : T_1 \to T_2'$ and $\Gamma \vdash t_2 : T_1$.

3. Case $t = \lambda x.t'$: assume $T_1 \to T_2 <: T_1' \to T_2'$ and the following derivation:

$$\frac{\dfrac{\Gamma, x : T_1 \vdash t' : T_2}{\Gamma \vdash \lambda x.t' : T_1 \to T_2}\ \lambda \qquad \dfrac{T_1' <: T_1 \qquad T_2 <: T_2'}{T_1 \to T_2 <: T_1' \to T_2'}\ \text{arrow}}{\Gamma \vdash \lambda x.t' : T_1' \to T_2'}\ \text{subsumption}$$

Alternatively we can derive:

$$\frac{\dfrac{\Gamma, x : T_1' \vdash t' : T_2 \qquad T_2 <: T_2'}{\Gamma, x : T_1' \vdash t' : T_2'}\ \text{subsumption}}{\Gamma \vdash \lambda x.t' : T_1' \to T_2'}\ \lambda$$

Since $T_1' <: T_1$ then by the subsumption rule:

$$\frac{\dfrac{}{\Gamma, x : T_1' \vdash x : T_1'}\ \text{var} \qquad T_1' <: T_1}{\Gamma, x : T_1' \vdash x : T_1}\ \text{subsumption}$$

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

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

However

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

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.

$$\dfrac{\dfrac{\color{red}{x : double \vdash x : int}}{\Gamma \vdash \lambda x.x : double \to int}\,\lambda \qquad \dfrac{\dfrac{SubBase(bool, double)}{bool <: double}\,\text{subBase} \qquad \dfrac{}{int <: int}\,\text{reflexive}}{double \to int <: bool \to int}\,\text{arrow}}{\Gamma \vdash \lambda x.x : bool \to int}\,\text{subsumption}$$

$$\dfrac{\dfrac{\dfrac{}{\Gamma, x : bool \vdash x : bool}\,\text{var} \qquad \dfrac{SubBase(bool, int)}{bool <: int}\,\text{suBase}}{\Gamma, x : bool \vdash x : int}\,\text{subsumption}}{\Gamma \vdash \lambda x.x : bool \to int}\,\lambda$$

10 System F

10.1 Variables

10.1.1 Valid

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

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

$$\dfrac{}{x : \forall X.(X \to X) \vdash x : \forall Y.(Y \to Y)}\,\text{var}$$

Y is free in the context and the type:

$$\dfrac{}{x : \forall X.(X \to Y) \vdash x : \forall Z.(Z \to Y)}\,\text{var}$$

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

$$\dfrac{\dfrac{}{x : \forall X.X \vdash x : \forall X1.X1}\,\text{var}}{x : \forall X.X \vdash x[(Y \to Y)] : (Y \to Y)}\,\text{elimination}$$

$$\dfrac{\dfrac{\dfrac{}{x : \forall X.X \vdash x : \forall X1.X1}\,\text{var}}{x : \forall X.X \vdash x[[(\forall X.X \to \forall X.X)]][(\forall X.X \to \forall X.X)] : (\forall X.X \to \forall X.X)}\,\text{elimination} \qquad \dfrac{}{x : \forall X.X \vdash x : \forall X.X}\,\text{var}}{x : \forall X.X \vdash (x[[(\forall X.X \to \forall X.X)]]\ x)[\forall X.X] : \forall X.X}\,\text{app}$$

10.1.2 Invalid

Y is free in the context:

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

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

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

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

10.2 Numbers

Zero

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

Zero with free variable X

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

Successor missing annotation

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{n : \forall X.((X \to X) \to (X \to X)), z : X, s : (X \to X) \vdash s : (X \to X)}\,\text{var} \qquad \dfrac{\dfrac{\dfrac{}{\color{blue}{n : \forall X.((X \to X) \to (X \to X)), z : X, s : (X \to X) \vdash n : ((X \to X) \to (X \to X))}}\,\color{blue}{\text{unknown}} \quad \dfrac{}{n : \forall X.((X \to X) \to (X \to X)), z : X, s : (X \to X) \vdash s : (X \to X)}\,\text{var}}{n : \forall X.((X \to X) \to (X \to X)), z : X, s : (X \to X) \vdash (n\ s)[(X \to X)] : (X \to X)}\,\text{app} \quad \dfrac{}{n : \forall X.((X \to X) \to (X \to X)), z : X, s : (X \to X) \vdash z : X}\,\text{var}}{n : \forall X.((X \to X) \to (X \to X)), z : X, s : (X \to X) \vdash ((n\ s)[(X \to X)]\ z)[X] : X}\,\text{app}}{n : \forall X.((X \to X) \to (X \to X)), z : X, s : (X \to X) \vdash (s\ ((n\ s)[(X \to X)]\ z)[X])[X] : X}\,\text{app}}{n : \forall X.((X \to X) \to (X \to X)), s : (X \to X) \vdash (\lambda z.(s\ ((n\ s)[(X \to X)]\ z)[X])[X]) : (X \to X)}\,\lambda}{n : \forall X.((X \to X) \to (X \to X)) \vdash (\lambda s.(\lambda z.(s\ ((n\ s)[(X \to X)]\ z)[X])[X])) : ((X \to X) \to (X \to X))}\,\lambda}{n : \forall X.((X \to X) \to (X \to X)) \vdash (\lambda s.(\lambda z.(s\ ((n\ s)[(X \to X)]\ z)[X])[X])) : \forall X.((X \to X) \to (X \to X))}\,\text{introduction}}{\cdot \vdash (\lambda n.(\lambda s.(\lambda z.(s\ ((n\ s)[(X \to X)]\ z)[X])[X]))) : (\forall X.((X \to X) \to (X \to X)) \to \forall X.((X \to X) \to (X \to X)))}\,\lambda
$$

Successor well annotated

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{n : \forall X.((X \to X) \to (X \to X)), z : X, s : (X \to X) \vdash s : (X \to X)}\,\text{var} \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{n : \forall X.((X \to X) \to (X \to X)), z : X, s : (X \to X) \vdash n : \forall X1.((X1 \to X1) \to (X1 \to X1))}\,\text{var}}{n : \forall X.((X \to X) \to (X \to X)), z : X, s : (X \to X) \vdash n[[X]][X] : ((X \to X) \to (X \to X))}\,\text{elimination} \quad \dfrac{}{n : \forall X.((X \to X) \to (X \to X)), z : X, s : (X \to X) \vdash s : (X \to X)}\,\text{var}}{n : \forall X.((X \to X) \to (X \to X)), z : X, s : (X \to X) \vdash (n[[X]]\ s)[(X \to X)] : (X \to X)}\,\text{app} \quad \dfrac{}{n : \forall X.((X \to X) \to (X \to X)), z : X, s : (X \to X) \vdash z : X}\,\text{var}}{n : \forall X.((X \to X) \to (X \to X)), z : X, s : (X \to X) \vdash ((n[[X]]\ s)[(X \to X)]\ z)[X] : X}\,\text{app}}{n : \forall X.((X \to X) \to (X \to X)), z : X, s : (X \to X) \vdash (s\ ((n[[X]]\ s)[(X \to X)]\ z)[X])[X] : X}\,\text{app}}{n : \forall X.((X \to X) \to (X \to X)), s : (X \to X) \vdash (\lambda z.(s\ ((n[[X]]\ s)[(X \to X)]\ z)[X])[X]) : (X \to X)}\,\lambda}{n : \forall X.((X \to X) \to (X \to X)) \vdash (\lambda s.(\lambda z.(s\ ((n[[X]]\ s)[(X \to X)]\ z)[X])[X])) : ((X \to X) \to (X \to X))}\,\lambda}{n : \forall X.((X \to X) \to (X \to X)) \vdash (\lambda s.(\lambda z.(s\ ((n[[X]]\ s)[(X \to X)]\ z)[X])[X])) : \forall X.((X \to X) \to (X \to X))}\,\text{introduction}}{\cdot \vdash (\lambda n.(\lambda s.(\lambda z.(s\ ((n[[X]]\ s)[(X \to X)]\ z)[X])[X]))) : (\forall X.((X \to X) \to (X \to X)) \to \forall X.((X \to X) \to (X \to X)))}\,\lambda
$$