From System T to Continuation-Passing Style

Yishuai Li

January 7, 2017

1 Introduction

1.1 System T

System T is an extension of simply typed lambda calculus with natural numbers. It provides *recursors* to define primitive recursive functions. [1, 2] This project focuses on a subset of system T that has if0 functions. It can be extended by fix operators to define recursive functions. The syntax and typing rules are defined in Chapter 2.

1.2 Continuation-passing style

Continuation is a data structure that represents the computational process at a given point in the process' execution; the created data structure can be accessed by the programming language, instead of being hidden in the runtime environment. [3]

In direct style, the **and** operation of boolean is defined as

```
and = fun b1 b2 : bool => if b1 then b2 else false
      : bool -> bool -> bool
```

The expression (**and** true (**and** false true)) can step either call by value to:

```
if true then false else false
```

or call by name to:

```
if false then true else false
```

depending on the evaluation strategy of the language.

In continuation-passing style, the **and** operation can be defined as

```
andb1 =
fun (b1 b2 : bool) (k : bool -> Set) =>
(fun x1 x2 : bool => k (if x1 then x2 else false)) b1 b2
      : bool -> bool -> (bool -> Set) -> Set
```

or

```
andb2 =
fun (b1 b2 : bool) (k : bool -> Set) =>
(fun x1 : bool => if x1 then (fun x2 : bool => k x2) b2 else k false) b1
      : bool -> bool -> (bool -> Set) -> Set
```

The expression (andb1 false true k) steps to

```
(fun x2 : bool =>
   k (if false then x2 else false)) true
```

while (andb2 false true k) steps to

```
if false then (fun x2 : bool => k x2) true else k false
```

In this way, we can define the evaluation strategy within the function definition.

This project is to formalize a translation from system T to continuation-passing style and provide an informal proof of its type correctness. The formalization of continuation-passing style is defined in Chapter 2. The translation is based on this [4] paper, and is defined in Chapter 3.

2 Formal definitions

2.1 Syntax definition

In the original paper [4] that translates system F to continuation-passing style, two different languages are used to represent the terms and expressions. For this project, the representation of system T and continuation-passing style (system K) are unified, and the languages T and K are specified by their typing rules. This unification provides convenience in case the formalization is represented in proof assistants *e.g.* Coq where substitutions are defined for each type and term definition.

$x,\ y$	variables			
i	integer literals			
$\tau,\ \sigma$	$::=$			types
	$\vert$	$\mathbb{Z}$		
	$\vert$	$\texttt{void}$		
	$\vert$	$\tau_1 \to \tau_2$		
	$\vert$	(τ)	S	
e	$::=$			annotated terms
	$\vert$	u^τ		
$u,\ v$	$::=$			raw terms
	$\vert$	x		
	$\vert$	i		
	$\vert$	$\lambda x : \tau . e$	bind x in e	
	$\vert$	$e_1\ e_2$		
	$\vert$	$e_1\ p\ e_2$		
	$\vert$	$\mathsf{if0}(e_1, e_2, e_3)$		
	$\vert$	$\mathsf{let}\ x = v\ \mathsf{in}\ u$	bind x in u	
	$\vert$	$\mathsf{halt}\,[\tau] e$		
	$\vert$	(u)	S	
p	$::=$			primitives
	$\vert$	$+$		
	$\vert$	$-$		
Γ	$::=$			contexts
	$\vert$			
	$\vert$	$\Gamma, x : \tau$		

2.2 Typing rules

In both system T and K, there are typing rules for annotated terms and raw terms, in which the typing of annotated terms simply checks if the raw term has the annotated type. This rule seems trivial, but is essential for the type checking of the translated terms.

$\boxed{\Gamma \vdash_{\mathrm{T}} e : \tau}$ annotated typing

$$\frac{\Gamma \vdash_{\mathrm{T}} u : \tau}{\Gamma \vdash_{\mathrm{T}} u^\tau : \tau}\ \textsc{T\_ant\_ann}$$

$\boxed{\Gamma \vdash_{\mathrm{T}} u : \tau}$ typing

$$\frac{\Gamma(x) = \tau}{\Gamma \vdash_{\mathrm{T}} x : \tau}\ \textsc{T\_term\_var}$$

$$\frac{}{\Gamma \vdash_{\mathrm{T}} i : \mathbb{Z}} \quad \text{T\_TERM\_INT}$$

$$\frac{\Gamma, x_1 : \tau_1 \vdash_{\mathrm{T}} e : \tau_2}{\Gamma \vdash_{\mathrm{T}} \lambda x_1 : \tau_1 . e : \tau_1 \to \tau_2} \quad \text{T\_TERM\_LAM}$$

$$\frac{\begin{array}{c}\Gamma \vdash_{\mathrm{T}} e_1 : \tau_1 \to \tau_2 \\ \Gamma \vdash_{\mathrm{T}} e_2 : \tau_1\end{array}}{\Gamma \vdash_{\mathrm{T}} e_1\, e_2 : \tau_2} \quad \text{T\_TERM\_APP}$$

$$\frac{\begin{array}{c}\Gamma \vdash_{\mathrm{T}} e_1 : \mathbb{Z} \\ \Gamma \vdash_{\mathrm{T}} e_2 : \mathbb{Z}\end{array}}{\Gamma \vdash_{\mathrm{T}} e_1\, p\, e_2 : \mathbb{Z}} \quad \text{T\_TERM\_PRIM}$$

$$\frac{\begin{array}{c}\Gamma \vdash_{\mathrm{T}} e_1 : \mathbb{Z} \\ \Gamma \vdash_{\mathrm{T}} e_2 : \tau \\ \Gamma \vdash_{\mathrm{T}} e_3 : \tau\end{array}}{\Gamma \vdash_{\mathrm{T}} \mathsf{if0}(e_1, e_2, e_3) : \tau} \quad \text{T\_TERM\_IF0}$$

In system K, programs are expressions of type `void`. It can be observed from the typing rules that halt is essential to construct any terms of type `void`. Therefore, all well-formed programs in this language must terminate.

$\boxed{\Gamma \vdash_{\mathrm{K}} e : \tau}$ annotated typing

$$\frac{\Gamma \vdash_{\mathrm{K}} u : \tau}{\Gamma \vdash_{\mathrm{K}} u^{\tau} : \tau} \quad \text{K\_ANT\_ANN}$$

$\boxed{\Gamma \vdash_{\mathrm{K}} u : \tau}$ typing

$$\frac{\Gamma(x) = \tau}{\Gamma \vdash_{\mathrm{K}} x : \tau} \quad \text{K\_TERM\_VAR}$$

$$\frac{}{\Gamma \vdash_{\mathrm{K}} i : \mathbb{Z}} \quad \text{K\_TERM\_INT}$$

$$\frac{\Gamma, x : \tau_1 \vdash_{\mathrm{K}} e : \tau_2}{\Gamma \vdash_{\mathrm{K}} \lambda x : \tau_1 . e : \tau_1 \to \tau_2} \quad \text{K\_TERM\_LAM}$$

$$\frac{\begin{array}{c}\Gamma \vdash_{\mathrm{K}} v : \tau \\ \Gamma, x : \tau \vdash_{\mathrm{K}} u : \mathtt{void}\end{array}}{\Gamma \vdash_{\mathrm{K}} \mathsf{let}\, x = v\, \mathsf{in}\, u : \mathtt{void}} \quad \text{K\_TERM\_LET}$$

$$\frac{\begin{array}{c}\Gamma \vdash_{\mathrm{K}} e_1 : \mathbb{Z} \\ \Gamma \vdash_{\mathrm{K}} e_2 : \mathbb{Z}\end{array}}{\Gamma \vdash_{\mathrm{K}} e_1\, p\, e_2 : \mathbb{Z}} \quad \text{K\_TERM\_PRIM}$$

$$\frac{\begin{array}{c}\Gamma \vdash_{\mathrm{K}} e' : \tau_1 \to \tau_2 \\ \Gamma \vdash_{\mathrm{K}} e : \tau\end{array}}{\Gamma \vdash_{\mathrm{K}} e'\, e : \tau_2} \quad \text{K\_TERM\_APP}$$

$$\frac{\begin{array}{c}\Gamma \vdash_{\mathrm{K}} e : \mathbb{Z} \\ \Gamma \vdash_{\mathrm{K}} e_1 : \tau \\ \Gamma \vdash_{\mathrm{K}} e_2 : \tau\end{array}}{\Gamma \vdash_{\mathrm{K}} \mathsf{if0}(e, e_1, e_2) : \tau} \quad \text{K\_TERM\_IF0}$$

$$\frac{\Gamma \vdash_{\mathrm{K}} e : \tau}{\Gamma \vdash_{\mathrm{K}} \mathsf{halt}\,[\tau] e : \mathtt{void}} \quad \text{K\_TERM\_HALT}$$

3 Translation

3.1 Type translation

In system K, continuations are expressions that take an expression and yields a program. Functions in system K take an expression of the argument type and passes the continuation to calculate the result value.

$$
\begin{aligned}
\mathcal{K}[\![\mathbb{Z}]\!] &\triangleq \mathbb{Z} \\
\mathcal{K}[\![\tau_1 \to \tau_2]\!] &\triangleq \mathcal{K}[\![\tau_1]\!] \to \mathcal{K}_{\mathtt{cont}}[\![\tau_2]\!] \to \mathtt{void} \\
\mathcal{K}_{\mathtt{cont}}[\![\tau]\!] &\triangleq \mathcal{K}[\![\tau]\!] \to \mathtt{void}
\end{aligned}
$$

3.2 Program translation

To translate a program in system T into system K, we first translate the expression into the corresponding type, and apply it to the continuation that takes the translated expression and halts.

$$
\begin{aligned}
\mathcal{K}_{\mathtt{prog}}[\![u^\tau]\!] &\triangleq \mathcal{K}_{\mathtt{exp}}[\![u^\tau]\!](\lambda x : \mathcal{K}[\![\tau]\!].\mathsf{halt}[\mathcal{K}[\![\tau]\!]]x^{\mathcal{K}[\![\tau]\!]})^{\mathcal{K}_{\mathtt{cont}}[\![\tau]\!]} \\
\mathcal{K}_{\mathtt{exp}}[\![y^\tau]\!]k &\triangleq k(y^{\mathcal{K}[\![\tau]\!]}) \\
\mathcal{K}_{\mathtt{exp}}[\![i^\tau]\!]k &\triangleq k(i^{\mathcal{K}[\![\tau]\!]}) \\
\mathcal{K}_{\mathtt{exp}}[\![(\lambda x_1 : \tau_1.u_2^{\tau_2})^\tau]\!]k &\triangleq k((\lambda x : \mathcal{K}[\![\tau_1]\!].\lambda c : \mathcal{K}_{\mathtt{cont}}[\![\tau_2]\!]. \\
&\qquad \mathcal{K}_{\mathtt{exp}}[\![u_2^{t_2}]\!]c^{\mathcal{K}_{\mathtt{cont}}[\![\tau_2]\!]})^{\mathcal{K}[\![\tau]\!]}) \\
\mathcal{K}_{\mathtt{exp}}[\![(u_1^{\tau_1} u_2^{\tau_2})^\tau]\!]k &\triangleq \mathcal{K}_{\mathtt{exp}}[\![u_1^{\tau_1}]\!](\lambda x_1 : \mathcal{K}[\![\tau_1]\!]. \\
&\qquad \mathcal{K}_{\mathtt{exp}}[\![u_2^{\tau_2}]\!](\lambda x_2 : \mathcal{K}[\![\tau_2]\!]. \\
&\qquad\quad x_1^{\mathcal{K}[\![\tau_1]\!]} x_2^{\mathcal{K}[\![\tau_2]\!]} k)^{\mathcal{K}_{\mathtt{cont}}[\![\tau_2]\!]})^{\mathcal{K}_{\mathtt{cont}}[\![\tau_1]\!]} \\
\mathcal{K}_{\mathtt{exp}}[\![(e_1\ p\ e_2)^\tau]\!]k &\triangleq \mathcal{K}_{\mathtt{exp}}[\![e_1]\!](\lambda x_1 : \mathbb{Z}. \\
&\qquad \mathcal{K}_{\mathtt{exp}}[\![e_2]\!](\lambda x_2 : \mathbb{Z}. \\
&\qquad\quad \mathsf{let}\ y = x_1\ p\ x_2\ \mathsf{in}\ k(y^{\mathbb{Z}}))^{\mathcal{K}_{\mathtt{cont}}[\![\mathbb{Z}]\!]})^{\mathcal{K}_{\mathtt{cont}}[\![\mathbb{Z}]\!]} \\
\mathcal{K}_{\mathtt{exp}}[\![\mathsf{if0}(e_1, e_2, e_3)^\tau]\!]k &\triangleq \mathcal{K}_{\mathtt{exp}}[\![e_1]\!](\lambda x : \mathbb{Z}. \\
&\qquad \mathsf{if0}(x^{\mathbb{Z}}, \mathcal{K}_{\mathtt{exp}}[\![e_2]\!]k, \mathcal{K}_{\mathtt{exp}}[\![e_3]\!]k))^{\mathcal{K}_{\mathtt{cont}}[\![\mathbb{Z}]\!]}
\end{aligned}
$$

4 Type correctness

An important property of the translation is that it translates well-formed λ^{T} expressions to well-formed λ^{K} expressions. The full correctness of a translation is the semantic coherency between the original expression and the translated version. Since type correctness is also a strong restriction for the translation required by the semantic correctness, it is worth checking the type correctness of a translation design beforehand. This is helpful when the semantic proof gets stuck and the designer questions whether the implementation or the proof goes wrong. That there are few habitants of a compact type provides more confidence of the correctness of the translation implementation.

4.1 Correctness of terms

Lemma 1. $\Gamma \vdash_{\mathrm{T}} e : \tau \implies \mathcal{K} \circ \Gamma \vdash_{\mathrm{K}} k : \mathcal{K}_{\mathtt{cont}}[\![\tau]\!] \implies \mathcal{K} \circ \Gamma \vdash_{\mathrm{K}} \mathcal{K}_{\mathtt{exp}}[\![e]\!]k : \mathtt{void}$

Proof. T_ANT_ANN is the only typing derivation of the hypothesis. Therefore,

$$e = u^\tau \tag{1}$$

$$\Gamma \vdash_{\mathrm{T}} u^\tau : \tau \tag{2}$$

We must show

$$\mathcal{K} \circ \Gamma \vdash_{\mathrm{K}} k : \mathcal{K}_{\mathtt{cont}}[\![\tau]\!] \implies \mathcal{K} \circ \Gamma \vdash_{\mathrm{K}} \mathcal{K}_{\mathtt{exp}}[\![u^\tau]\!]k : \mathtt{void} \tag{3}$$

By induction on a derivation of typing judgement (2):

1. If the last rule in the derivation is T_TERM_VAR, then

$$u = x \tag{4}$$

$$\Gamma(x) = \tau \tag{5}$$

$$\mathcal{K} \circ \Gamma \vdash_{\mathrm{K}} k : \mathcal{K}[\![\tau]\!] \to \texttt{void} \tag{6}$$

We must show

$$\mathcal{K} \circ \Gamma \vdash_{\mathrm{K}} \mathcal{K}_{\texttt{exp}}[\![x^{\tau}]\!]k : \texttt{void} \tag{7}$$

i.e.

$$\mathcal{K} \circ \Gamma \vdash_{\mathrm{K}} k(x^{\mathcal{K}[\![\tau]\!]}) : \texttt{void} \tag{8}$$

By K_TERM_APP and typing judgement (6), this follows from

$$\mathcal{K} \circ \Gamma \vdash_{\mathrm{K}} x^{\mathcal{K}[\![\tau]\!]} : \mathcal{K}[\![\tau]\!] \tag{9}$$

By K_ANT_ANN, this follows from

$$\mathcal{K} \circ \Gamma \vdash_{\mathrm{K}} x : \mathcal{K}[\![\tau]\!] \tag{10}$$

By K_TERM_VAR, this follows from

$$\mathcal{K} \circ \Gamma(x) = \mathcal{K}[\![\tau]\!] \tag{11}$$

which is immediate from typing judgement (5).

2. If the last rule in the derivation is T_TERM_INT

We have

$$u = i \tag{12}$$

$$\tau = \mathbb{Z} \tag{13}$$

$$\mathcal{K} \circ \Gamma \vdash_{\mathrm{K}} k : \mathbb{Z} \to \texttt{void} \tag{14}$$

We must show

$$\mathcal{K} \circ \Gamma \vdash_{\mathrm{K}} \mathcal{K}_{\texttt{exp}}[\![i^{\mathbb{Z}}]\!]k : \texttt{void} \tag{15}$$

i.e.

$$\mathcal{K} \circ \Gamma \vdash_{\mathrm{K}} k(i^{\mathcal{K}[\![\mathbb{Z}]\!]}) : \texttt{void} \tag{16}$$

By K_TERM_APP and typing judgement (14), this follows from

$$\mathcal{K} \circ \Gamma \vdash_{\mathrm{K}} i^{\mathcal{K}[\![\mathbb{Z}]\!]} : \mathbb{Z} \tag{17}$$

By K_ANT_ANN, this follows from

$$\mathcal{K} \circ \Gamma \vdash_{\mathrm{K}} i : \mathbb{Z} \tag{18}$$

which is immediate from K_TERM_INT.

3. If the last rule in the derivation is T_TERM_LAM, then

$$u = \lambda x : \tau_1 . e_2 \tag{19}$$

$$\tau = \tau_1 \to \tau_2 \tag{20}$$

$$\Gamma, x : \tau_1 \vdash_{\mathrm{T}} e_2 : \tau_2 \tag{21}$$

$$\mathcal{K} \circ \Gamma \vdash_{\mathrm{K}} k : \mathcal{K}[\![\tau_1 \to \tau_2]\!] \to \texttt{void} \tag{22}$$

T_ANT_ANN is the only derivation of typing judgement (21). Therefore,

$$e_2 = u_2^{\tau_2} \tag{23}$$

$$\Gamma, x : \tau_1 \vdash_{\mathrm{T}} u_2 : \tau_2 \tag{24}$$

We must show

$$\mathcal{K} \circ \Gamma \vdash_{\mathrm{K}} \mathcal{K}_{\texttt{exp}}[\![\lambda x : \tau_1 . u_2^{\tau_2}]\!]k : \texttt{void} \tag{25}$$

i.e.

$$\mathcal{K} \circ \Gamma \vdash_{\mathrm{K}} k((\lambda x : \mathcal{K}[\![\tau_1]\!] . \lambda c : \mathcal{K}_{\texttt{cont}}[\![\tau_2]\!] . \mathcal{K}_{\texttt{exp}}[\![u_2^{\tau_2}]\!]c^{\mathcal{K}_{\texttt{cont}}[\![\tau_2]\!]})^{\mathcal{K}[\![\tau_1 \to \tau_2]\!]}) : \texttt{void} \tag{26}$$

By K_TERM_APP and typing judgement (22), this follows from

$$\mathcal{K} \circ \Gamma \vdash_{\mathrm{K}} (\lambda x : \mathcal{K}[\![\tau_1]\!].\lambda c : \mathcal{K}_{\mathtt{cont}}[\![\tau_2]\!].\mathcal{K}_{\mathtt{exp}}[\![u_2^{\tau_2}]\!]c^{\mathcal{K}_{\mathtt{cont}}[\![\tau_2]\!]})^{\mathcal{K}[\![\tau_1 \to \tau_2]\!]} : \mathcal{K}[\![\tau_1 \to \tau_2]\!] \tag{27}$$

By K_ANT_ANN, this follows from

$$\mathcal{K} \circ \Gamma \vdash_{\mathrm{K}} \lambda x : \mathcal{K}[\![\tau_1]\!].\lambda c : \mathcal{K}_{\mathtt{cont}}[\![\tau_2]\!].\mathcal{K}_{\mathtt{exp}}[\![u_2^{\tau_2}]\!]c^{\mathcal{K}_{\mathtt{cont}}[\![\tau_2]\!]} : \mathcal{K}[\![\tau_1]\!] \to \mathcal{K}_{\mathtt{cont}}[\![\tau_2]\!] \to \mathtt{void} \tag{28}$$

By K_TERM_LAM, this follows from

$$\mathcal{K} \circ \Gamma, x : \mathcal{K}[\![\tau_1]\!] \vdash_{\mathrm{K}} \lambda c : \mathcal{K}_{\mathtt{cont}}[\![\tau_2]\!].\mathcal{K}_{\mathtt{exp}}[\![u_2^{\tau_2}]\!]c^{\mathcal{K}_{\mathtt{cont}}[\![\tau_2]\!]} : \mathcal{K}_{\mathtt{cont}}[\![\tau_2]\!] \to \mathtt{void} \tag{29}$$

which follows

$$\mathcal{K} \circ \Gamma, x : \mathcal{K}[\![\tau_1]\!], c : \mathcal{K}_{\mathtt{cont}}[\![\tau_2]\!] \vdash_{\mathrm{K}} \mathcal{K}_{\mathtt{exp}}[\![u_2^{\tau_2}]\!]c^{\mathcal{K}_{\mathtt{cont}}[\![\tau_2]\!]} : \mathtt{void} \tag{30}$$

which is immediate from K_ANT_ANN, K_TERM_VAR and the induction hypothesis.

4. If the last rule in the derivation is T_TERM_APP, then

$$u = e_1 e_2 \tag{31}$$

$$\Gamma \vdash_{\mathrm{T}} e_1 : \tau_1 \to \tau \tag{32}$$

$$\Gamma \vdash_{\mathrm{T}} e_2 : \tau_1 \tag{33}$$

$$\mathcal{K} \circ \Gamma \vdash_{\mathrm{K}} k : \mathcal{K}_{\mathtt{cont}}[\![\tau]\!] \tag{34}$$

T_ANT_ANN is the only derivation of typing judgements (32)(33). Therefore,

$$e_1 = u_1^{\tau_1 \to \tau} \tag{35}$$

$$e_2 = u_2^{\tau_1} \tag{36}$$

$$\Gamma \vdash_{\mathrm{T}} u_1 : \tau_1 \to \tau \tag{37}$$

$$\Gamma \vdash_{\mathrm{T}} u_2 : \tau_1 \tag{38}$$

We must show

$$\mathcal{K} \circ \Gamma \vdash_{\mathrm{K}} \mathcal{K}_{\mathtt{exp}}[\![(u_1^{\tau_1 \to \tau} u_2^{\tau_1})^{\tau}]\!]k : \mathtt{void} \tag{39}$$

i.e.

$$\mathcal{K} \circ \Gamma \vdash_{\mathrm{K}} \mathcal{K}_{\mathtt{exp}}[\![{u_1}^{\tau_1 \to \tau}]\!](\lambda x_1 : \mathcal{K}[\![\tau_1 \to \tau]\!].\mathcal{K}_{\mathtt{exp}}[\![u_2^{\tau_1}]\!](\lambda x_2 : \mathcal{K}[\![\tau_1]\!]. x_1^{\mathcal{K}[\![\tau_1 \to \tau]\!]} x_2^{\mathcal{K}[\![\tau_1]\!]} k)^{\mathcal{K}_{\mathrm{cont}}[\![\tau_1]\!]})^{\mathcal{K}_{\mathrm{cont}}[\![\tau_1 \to \tau]\!]} : \mathtt{void} \tag{40}$$

By the induction hypothesis, this follows from

$$\mathcal{K} \circ \Gamma \vdash_{\mathrm{K}} (\lambda x_1 : \mathcal{K}[\![\tau_1 \to \tau]\!].\mathcal{K}_{\mathtt{exp}}[\![u_2^{\tau_1}]\!](\lambda x_2 : \mathcal{K}[\![\tau_1]\!]. x_1^{\mathcal{K}[\![\tau_1 \to \tau]\!]} x_2^{\mathcal{K}[\![\tau_1]\!]} k)^{\mathcal{K}_{\mathrm{cont}}[\![\tau_1]\!]})^{\mathcal{K}_{\mathrm{cont}}[\![\tau_1 \to \tau]\!]} : \mathcal{K}_{\mathtt{cont}}[\![\tau_1 \to \tau]\!] \tag{41}$$

By K_ANT_ANN, this follows from

$$\mathcal{K} \circ \Gamma \vdash_{\mathrm{K}} \lambda u_1 : \mathcal{K}[\![\tau_1 \to \tau]\!].\mathcal{K}_{\mathtt{exp}}[\![u_2^{\tau_1}]\!](\lambda x_2 : \mathcal{K}[\![\tau_1]\!].u_1^{\mathcal{K}[\![\tau_1 \to \tau]\!]} x_2^{\mathcal{K}[\![\tau_1]\!]} k)^{\mathcal{K}_{\mathrm{cont}}[\![\tau_1]\!]} : \mathcal{K}[\![\tau_1 \to \tau]\!] \to \mathtt{void} \tag{42}$$

By K_TERM_LAM, this follows from

$$\mathcal{K} \circ \Gamma, x_1 : \mathcal{K}[\![\tau_1 \to \tau]\!] \vdash_{\mathrm{K}} \mathcal{K}_{\mathtt{exp}}[\![u_2^{\tau_1}]\!](\lambda x_2 : \mathcal{K}[\![\tau_1]\!].x_1^{\mathcal{K}[\![\tau_1 \to \tau]\!]} x_2^{\mathcal{K}[\![\tau_1]\!]} k)^{\mathcal{K}_{\mathrm{cont}}[\![\tau_1]\!]} : \mathtt{void} \tag{43}$$

By the induction hypothesis, We must show

$$\mathcal{K} \circ \Gamma, x_1 : \mathcal{K}[\![\tau_1 \to \tau]\!] \vdash_{\mathrm{K}} (\lambda x_2 : \mathcal{K}[\![\tau_1]\!].x_1^{\mathcal{K}[\![\tau_1 \to \tau]\!]} x_2^{\mathcal{K}[\![\tau_1]\!]} k)^{\mathcal{K}_{\mathrm{cont}}[\![\tau_1]\!]} : \mathcal{K}_{\mathtt{cont}}[\![\tau_1]\!] \tag{44}$$

By K_ANT_ANN, this follows from

$$\mathcal{K} \circ \Gamma, x_1 : \mathcal{K}[\![\tau_1 \to \tau]\!] \vdash_{\mathrm{K}} \lambda x_2 : \mathcal{K}[\![\tau_1]\!].x_1^{\mathcal{K}[\![\tau_1 \to \tau]\!]} x_2^{\mathcal{K}[\![\tau_1]\!]} k : \mathcal{K}[\![\tau_1]\!] \to \mathtt{void} \tag{45}$$

By K_TERM_LAM, this follows from

$$\mathcal{K} \circ \Gamma, x_1 : \mathcal{K}[\![\tau_1 \to \tau]\!], x_2 : \mathcal{K}[\![\tau_1]\!] \vdash_{\mathrm{K}} x_1^{\mathcal{K}[\![\tau_1 \to \tau]\!]} x_2^{\mathcal{K}[\![\tau_1]\!]} k : \mathtt{void} \tag{46}$$

which is immediate from K_TERM_APP, K_ANT_ANN, K_TERM_VAR and typing judgement (34).

5. If the last rule in the derivation is T_TERM_PRIM, then

$$u = e_1 \; p \; e_2 \tag{47}$$

$$\tau = \mathbb{Z} \tag{48}$$

$$\Gamma \vdash_{\mathrm{T}} e_1 : \mathbb{Z} \tag{49}$$

$$\Gamma \vdash_{\mathrm{T}} e_2 : \mathbb{Z} \tag{50}$$

$$\mathcal{K} \circ \Gamma \vdash_{\mathrm{K}} k : \mathbb{Z} \to \mathtt{void} \tag{51}$$

T_ANT_ANN is the only derivation of typing judgement (49)(50). Therefore,

$$e_1 = u_1^{\mathbb{Z}} \tag{52}$$

$$e_2 = u_2^{\mathbb{Z}} \tag{53}$$

$$\Gamma \vdash_{\mathrm{T}} u_1 : \mathbb{Z} \tag{54}$$

$$\Gamma \vdash_{\mathrm{T}} u_2 : \mathbb{Z} \tag{55}$$

We must show

$$\mathcal{K} \circ \Gamma \vdash_{\mathrm{K}} \mathcal{K}_{\mathsf{exp}}[\![(u_1^{\mathbb{Z}} \; p \; u_2^{\mathbb{Z}})^{\mathbb{Z}}]\!]k : \mathtt{void} \tag{56}$$

i.e.

$$\mathcal{K} \circ \Gamma \vdash_{\mathrm{K}} \mathcal{K}_{\mathsf{exp}}[\![u_1^{\mathbb{Z}}]\!](\lambda x_1 : \mathbb{Z}.\mathcal{K}_{\mathsf{exp}}[\![u_2^{\mathbb{Z}}]\!](\lambda x_2 : \mathbb{Z}.\mathsf{let}\; y = x_1 \; p \; x_2 \;\mathsf{in}\; k(y^{\mathbb{Z}}))^{\mathcal{K}_{\mathsf{cont}}[\![\mathbb{Z}]\!]})^{\mathcal{K}_{\mathsf{cont}}[\![\mathbb{Z}]\!]} : \mathtt{void} \tag{57}$$

By the induction hypothesis, this follows from

$$\mathcal{K} \circ \Gamma \vdash_{\mathrm{K}} (\lambda x_1 : \mathbb{Z}.\mathcal{K}_{\mathsf{exp}}[\![u_2]\!](\lambda x_2 : \mathbb{Z}.\mathsf{let}\; y = x_1 \; p \; x_2 \;\mathsf{in}\; k(y^{\mathbb{Z}}))^{\mathcal{K}_{\mathsf{cont}}[\![\mathbb{Z}]\!]})^{\mathcal{K}_{\mathsf{cont}}[\![\mathbb{Z}]\!]} : \mathcal{K}_{\mathsf{cont}}[\![\mathbb{Z}]\!] \tag{58}$$

By K_ANT_ANN, this follows from

$$\mathcal{K} \circ \Gamma \vdash_{\mathrm{K}} \lambda x_1 : \mathbb{Z}.\mathcal{K}_{\mathsf{exp}}[\![u_2]\!](\lambda x_2 : \mathbb{Z}.\mathsf{let}\; y = x_1 \; p \; x_2 \;\mathsf{in}\; k(y^{\mathbb{Z}}))^{\mathcal{K}_{\mathsf{cont}}[\![\mathbb{Z}]\!]} : \mathbb{Z} \to \mathtt{void} \tag{59}$$

By K_TERM_LAM, this follows from

$$\mathcal{K} \circ \Gamma, x_1 : \mathbb{Z} \vdash_{\mathrm{K}} \mathcal{K}_{\mathsf{exp}}[\![u_2]\!](\lambda x_2 : \mathbb{Z}.\mathsf{let}\; y = x_1 \; p \; x_2 \;\mathsf{in}\; k(y^{\mathbb{Z}}))^{\mathcal{K}_{\mathsf{cont}}[\![\mathbb{Z}]\!]} : \mathtt{void} \tag{60}$$

By the induction hypothesis, this follows from

$$\mathcal{K} \circ \Gamma, x_1 : \mathbb{Z} \vdash_{\mathrm{K}} (\lambda u_2 : \mathbb{Z}.\mathsf{let}\; y = x_1 \; p \; u_2 \;\mathsf{in}\; k(y^{\mathbb{Z}}))^{\mathcal{K}_{\mathsf{cont}}[\![\mathbb{Z}]\!]} : \mathcal{K}_{\mathsf{cont}}[\![\mathbb{Z}]\!] \tag{61}$$

By K_ANT_ANN, this follows from

$$\mathcal{K} \circ \Gamma, x_1 : \mathbb{Z} \vdash_{\mathrm{K}} \lambda x_2 : \mathbb{Z}.\mathsf{let}\; y = x_1 \; p \; x_2 \;\mathsf{in}\; k(y^{\mathbb{Z}}) : \mathbb{Z} \to \mathtt{void} \tag{62}$$

By K_TERM_LAM, this follows from

$$\mathcal{K} \circ \Gamma, x_1 : \mathbb{Z}, x_2 : \mathbb{Z} \vdash_{\mathrm{K}} \mathsf{let}\; y = x_1 \; p \; x_2 \;\mathsf{in}\; k(y^{\mathbb{Z}}) : \mathtt{void} \tag{63}$$

By K_ANT_LET, K_ANT_PRIM and K_ANT_VAR, We must show

$$\mathcal{K} \circ \Gamma, x_1 : \mathbb{Z}, x_2 : \mathbb{Z}, y : \mathbb{Z} \vdash_{\mathrm{K}} k(y^{\mathbb{Z}}) : \mathtt{void} \tag{64}$$

By K_TERM_APP and typing judgement (51), this follows from

$$\mathcal{K} \circ \Gamma, x_1 : \mathbb{Z}, x_2 : \mathbb{Z}, y : \mathbb{Z} \vdash_{\mathrm{K}} y^{\mathbb{Z}} : \mathbb{Z} \tag{65}$$

which is immediate from K_ANT_ANN and K_ANT_VAR.

6. If the last rule in the derivation is T_TERM_IF0, then

$$u = \mathsf{if0}(e_1, e_2, e_3) \tag{66}$$

$$\Gamma \vdash_{\mathrm{T}} e_1 : \mathbb{Z} \tag{67}$$

$$\Gamma \vdash_{\mathrm{T}} e_2 : \tau \tag{68}$$

$$\Gamma \vdash_{\mathrm{T}} e_3 : \tau \tag{69}$$

$$\mathcal{K} \circ \Gamma \vdash_{\mathrm{K}} k : \mathcal{K}_{\mathsf{cont}}[\![\tau]\!] \tag{70}$$

T_ANT_ANN is only constructor of typing judgements (67)(68)(69). Therefore,

$$e_1 = u_1^{\mathbb{Z}} \tag{71}$$

$$e_2 = u_2^{\tau} \tag{72}$$

$$e_3 = u_3^{\tau} \tag{73}$$

$$\Gamma \vdash_{\mathrm{T}} u_1 : \mathbb{Z} \tag{74}$$

$$\Gamma \vdash_{\mathrm{T}} u_2 : \tau \tag{75}$$

$$\Gamma \vdash_{\mathrm{T}} u_3 : \tau \tag{76}$$

We must show

$$\mathcal{K} \circ \Gamma \vdash_{\mathrm{K}} \mathcal{K}_{\mathsf{exp}}[\![\mathsf{if0}(u_1^{\mathbb{Z}}, u_2^{\tau}, u_3^{\tau})]\!]k : \mathtt{void} \tag{77}$$

i.e.

$$\mathcal{K} \circ \Gamma \vdash_{\mathrm{K}} \mathcal{K}_{\mathsf{exp}}[\![u_1^{\mathbb{Z}}]\!](\lambda x : \mathbb{Z}.\mathsf{if0}(x^{\mathbb{Z}}, \mathcal{K}_{\mathsf{exp}}[\![u_2^{\tau}]\!]k, \mathcal{K}_{\mathsf{exp}}[\![u_3^{\tau}]\!]k))^{\mathcal{K}_{\mathsf{cont}}[\![\mathbb{Z}]\!]} : \mathtt{void} \tag{78}$$

By the induction hypothesis, We must show

$$\mathcal{K} \circ \Gamma \vdash_{\mathrm{K}} (\lambda x : \mathbb{Z}.\mathsf{if0}(x^{\mathbb{Z}}, \mathcal{K}_{\mathsf{exp}}[\![u_2^{\tau}]\!]k, \mathcal{K}_{\mathsf{exp}}[\![u_3^{\tau}]\!]k))^{\mathcal{K}_{\mathsf{cont}}[\![\mathbb{Z}]\!]} : \mathcal{K}_{\mathsf{cont}}[\![\mathbb{Z}]\!] \tag{79}$$

By K_ANT_ANN, this follows from

$$\mathcal{K} \circ \Gamma \vdash_{\mathrm{K}} \lambda x : \mathbb{Z}.\mathsf{if0}(x^{\mathbb{Z}}, \mathcal{K}_{\mathsf{exp}}[\![u_2^{\tau}]\!]k, \mathcal{K}_{\mathsf{exp}}[\![u_3^{\tau}]\!]k) : \mathbb{Z} \to \mathtt{void} \tag{80}$$

By K_TERM_LAM, this follows from

$$\mathcal{K} \circ \Gamma, x : \mathbb{Z} \vdash_{\mathrm{K}} \mathsf{if0}(x^{\mathbb{Z}}, \mathcal{K}_{\mathsf{exp}}[\![u_2^{\tau}]\!]k, \mathcal{K}_{\mathsf{exp}}[\![u_3^{\tau}]\!]k) : \mathtt{void} \tag{81}$$

By K_TERM_IF0, this follows from

$$\mathcal{K} \circ \Gamma, x : \mathbb{Z} \vdash_{\mathrm{K}} x^{\mathbb{Z}} : \mathbb{Z} \tag{82}$$

$$\mathcal{K} \circ \Gamma, x : \mathbb{Z} \vdash_{\mathrm{K}} \mathcal{K}_{\mathsf{exp}}[\![u_2^{\tau}]\!]k : \mathtt{void} \tag{83}$$

$$\mathcal{K} \circ \Gamma, x : \mathbb{Z} \vdash_{\mathrm{K}} \mathcal{K}_{\mathsf{exp}}[\![u_3^{\tau}]\!]k : \mathtt{void} \tag{84}$$

in which (82) is immediate from K_ANT_ANN and K_TERM_VAR. (83)(84) come from the induction hypotheses, K_ANT_ANN and typing judgements (70)(75)(76).

□

4.2 Correctness of programs

Theorem 1. $\vdash_{\mathrm{T}} e : \tau \implies \vdash_{\mathrm{K}} \mathcal{K}_{\texttt{prog}}[\![e]\!] : \texttt{void}$

Proof. T_ANT_ANN is the only typing derivation for the hypothesis. Therefore,

$$e = u^\tau \tag{85}$$

$$\vdash_{\mathrm{T}} u^\tau : \tau \tag{86}$$

We must show

$$\vdash_{\mathrm{K}} \mathcal{K}_{\texttt{prog}}[\![u^\tau]\!] : \texttt{void} \tag{87}$$

i.e.

$$\vdash_{\mathrm{K}} \mathcal{K}_{\texttt{exp}}[\![u^\tau]\!](\lambda x : \mathcal{K}[\![\tau]\!].\mathsf{halt}[\mathcal{K}[\![\tau]\!]]x^{\mathcal{K}[\![\tau]\!]})^{\mathcal{K}_{\texttt{cont}}[\![\tau]\!]} : \texttt{void} \tag{88}$$

By Lemma 1, this follows from

$$\vdash_{\mathrm{K}} (\lambda x : \mathcal{K}[\![\tau]\!].\mathsf{halt}[\mathcal{K}[\![\tau]\!]]x^{\mathcal{K}[\![\tau]\!]})^{\mathcal{K}_{\texttt{cont}}[\![\tau]\!]} : \mathcal{K}_{\texttt{cont}}[\![\tau]\!] \tag{89}$$

By K_ANT_ANN, this follows from

$$\vdash_{\mathrm{K}} \lambda x : \mathcal{K}[\![\tau]\!].\mathsf{halt}[\mathcal{K}[\![\tau]\!]]x^{\mathcal{K}[\![\tau]\!]} : \mathcal{K}[\![\tau]\!] \to \texttt{void} \tag{90}$$

By K_TERM_LAM, this follows from

$$x : \mathcal{K}[\![\tau]\!] \vdash_{\mathrm{K}} \mathsf{halt}[\mathcal{K}[\![\tau]\!]]x^{\mathcal{K}[\![\tau]\!]} : \texttt{void} \tag{91}$$

By K_TERM_HALT, this follows from

$$x : \mathcal{K}[\![\tau]\!] \vdash_{\mathrm{K}} x^{\mathcal{K}[\![\tau]\!]} : \mathcal{K}[\![\tau]\!] \tag{92}$$

which is immediate from K_ANT_ANN and K_TERM_VAR. □

5 Future work

It was planned that the proof of the type correctness theorem to be formalized in Coq [5], using definitions generated by Ott [6] and locally nameless representation by LNgen [7]. However, the proof was stuck due to my inadequacy in proof techniques. In particular, for Lemma 1, Γ was defined as a mapping from free variables to types. The composition of $\mathcal{K}$ and Γ represents the binding of free variables to the translated type in system K. It is worth studying the proper way to formalize the context to meet the need of composing translations with the type mapping.

References

[1] Kurt Gödel. On a hitherto unexploited extension of the finitary standpoint. *Journal of Philosophical Logic*, 9(2):133–142, 1980.

[2] Robert Harper. *Practical foundations for programming languages*. Cambridge University Press, 2016.

[3] John C Mitchell. *Concepts in programming languages*. Cambridge University Press, 2003.

[4] Greg Morrisett, David Walker, Karl Crary, and Neal Glew. From system f to typed assembly language. *ACM Trans. Program. Lang. Syst.*, 21(3):527–568, May 1999.

[5] Yishuai Li and Stephanie Weirich. cps. https://github.com/liyishuai/cps, 2016.

[6] Peter Sewell, Francesco Zappa Nardelli, Scott Owens, Gilles Peskine, Thomas Ridge, Susmit Sarkar, and Rok Strniša. Ott: Effective tool support for the working semanticist. In *Proceedings of the 12th ACM SIGPLAN International Conference on Functional Programming*, ICFP '07, pages 1–12, New York, NY, USA, 2007. ACM.

[7] Brian Aydemir and Stephanie Weirich. Lngen: Tool support for locally nameless representations. 2010.