DEPENDENT TYPE THEORY

HOJUNE LEE, 20210541

Contents

1. Contexts and Substitutions 2
1.1. Intuitions 2
1.2. Contexts Judgements 3
1.3. Substitution Judgements 3

1. Contexts and Substitutions

1.1. **Intuitions.** In simple type theory, every thing was quite clear. This is because that, if we once fix the rules for types in simple type theory, then all possible types are determined immediately by inductive way. We can informally think this as 'propositional logic'-like type construction. However, many things are different in dependent type system. Let's think dependent type system as 'first order logic'-like type construction. Imagine the formulas in first order logic as types in dependent type theory. I hope that this is proper intuitive thinking for the main differences between two type systems.

Once we imagine FOL formula-style type, it means that each type can contain variables and constants as it's representation. It means that for each variables which are appeared in type representation, we must know what are the types of each variables. That is Context's role.

Now, we need to construct the intuition for contexts and substitutions. Contexts are 'wolrds' of terms and types. And substitutions are 'traffic way' between each worlds(contexts). Then naturally, following questions are arising.

Questions

(1) Is this world (Context) well-formed ?
(2) In this world (Context) Γ, what terms of type A are well-formed ?
(3) In this world (Context) Γ, what types are well-formed ?
(4) Between 2 worlds (Context) Δ and Γ, is traffic way (substitution) γ well-formed ?
(5) In this world (Context), which terms/types/substitutions are equivalent?

Actually, this is all about 'judgement rules' referred in Notation 2.3.1. of textbook. Formally, we can write format for above judgements.

Formal Representations for Judgements

(1) $\vdash \Gamma$ cx
(2) $\Gamma \vdash a : A$
(3) $\Gamma \vdash A$ type
(4) $\Delta \vdash \gamma : \Gamma$
(5) $\Gamma \vdash a = a' : A, \quad \Gamma \vdash A = A'$ type, $\quad \Delta \vdash \gamma = \gamma' : \Gamma$

In this chapter, we'll discuss the rules for above judgements. Now, it's time to define explicit judgement rules that we referred.

1.2. **Contexts Judgements.**

Construction 1.2.1.

$$\frac{}{\vdash \mathbf{1}\ \mathrm{cx}} \qquad\qquad \frac{\vdash \Gamma\ \mathrm{cx} \quad \Gamma \vdash A\ \mathrm{type}}{\vdash \Gamma.A\ \mathrm{cx}}$$

Here, **1** is empty context and each context is just list of types. (No variable names) It means that we use De Bruijn index, i.e. in each context, index automatically determines the variable in context.

1.3. **Substitution Judgements.**

However, before that the most important one is understanding direction of traffic way (from now on, substitution).

Definition 1.3.1. If $\Delta \vdash \gamma : \Gamma$, then γ is substitution from Δ to Γ. i.e.

$$\gamma : \Delta \to \Gamma$$

However, it's role is send types and terms of Γ into Δ. (Note : Direction is important)

Intuitions for direction of substitutions

To understand why here use this notation, see following example. Imagine that contexts are 'sets' and substitutions are function between them. There are 2 sets Δ, Γ and mapping $\gamma : \Delta \to \Gamma$. Suppose that there is a function $g : \Gamma \to \mathbb{R}$, which is defined on set Γ. How can we 'use' this function in Δ set? One way is that,

$$g : \Gamma \to \mathbb{R} \implies g \circ \gamma : \Delta \to \mathbb{R}$$

Then we can 'use' the function g in domain Δ. This exactly corresponds in our notation. For $\Delta \vdash \gamma : \Gamma$, when we define $\gamma : \Delta \to \Gamma$,

$$\Gamma \vdash A\ \mathrm{type} \implies \Delta \vdash A[\gamma]\ \mathrm{type}$$

Since A is type in Γ and the direction of substitution is $\gamma : \Delta \to \Gamma$, we say that $A[\gamma]$ is pull-backed type of A through $\gamma : \Delta \to \Gamma$.

First, as above we can easily define the application rule of substitution.

Construction 1.3.2.

$$\frac{\Delta \vdash \gamma : \Gamma \quad \Gamma \vdash A\ \mathrm{type}}{\Delta \vdash A[\gamma]\ \mathrm{type}} \qquad\qquad \frac{\Delta \vdash \gamma : \Gamma \quad \Gamma \vdash a : A}{\Delta \vdash a[\gamma] : A[\gamma]}$$

These rules give us that we can immigrate(pull-back) terms and types of Γ into Δ. And now, in our setting each contexts are different 'worlds'. So we need to introduce explicit **weakening rule**, which intuitively means that we can bring well-formed types and terms into Γ into $\Gamma.A$, expanded context.

Construction 1.3.3.

$$\frac{\Gamma \vdash A \text{ type}}{\Gamma.A \vdash \mathbf{p} : \Gamma}$$

In diagram,

$$\Gamma.A \xrightarrow{p} \Gamma$$

Means that, we can pull-back types B and terms t in world Γ into $\Gamma.A$ by write $B[\mathbf{p}], t[\mathbf{p}]$ Moreover, we can introduce following rules. (Actually, following rules need because our context-substitution system become a category.)

Construction 1.3.4.

$$\frac{\vdash \Gamma \text{ cx}}{\Gamma \vdash \mathbf{id} : \Gamma} \qquad \frac{\Gamma_2 \vdash \gamma_1 : \Gamma_1 \quad \Gamma_1 \vdash \gamma_0 : \Gamma_0}{\Gamma_2 \vdash \gamma_0 \circ \gamma_1 : \Gamma_0}$$

Second one can be conflict when we see first. However, let's draw the diagram.

$$\begin{tikzcd}
\Gamma_2 \arrow[r, "\gamma_1"] \arrow[d, "\gamma_0 \circ \gamma_1"'] & \Gamma_1 \arrow[dl, "\gamma_0"] \\
\Gamma_0 &
\end{tikzcd}$$

Then, above representation is very clear. It become same notation in our function calculus. Similarly,

Construction 1.3.5.

$$\frac{\Delta \vdash \gamma : \Gamma}{\Delta \vdash \gamma \circ \mathbf{id} = \mathbf{id} \circ \gamma = \gamma : \Gamma} \qquad \frac{\Gamma_3 \vdash \gamma_2 : \Gamma_2 \quad \Gamma_2 \vdash \gamma_1 : \Gamma_1 \quad \Gamma_1 \vdash \gamma_0 : \Gamma_0}{\Gamma_3 \vdash (\gamma_0 \circ \gamma_1) \circ \gamma_2 = \gamma_0 \circ (\gamma_1 \circ \gamma_2) : \Gamma_0}$$

This two rules are also very clear when we see the diagram.

$$\begin{tikzcd}
\Delta \arrow[loop left, "id"] \arrow[r, "\gamma"] & \Gamma \arrow[loop right, "id"]
\end{tikzcd}$$

$$\begin{tikzcd}
\Gamma_3 \arrow[r, "\gamma_2"] \arrow[d, "(\gamma_0\circ\gamma_1)\circ\gamma_2"'] & \Gamma_2 \arrow[d, "\gamma_1"] \arrow[dl, "\gamma_0\circ\gamma_1"] \\
\Gamma_0 & \Gamma_1 \arrow[l, "\gamma_0"]
\end{tikzcd}
\iff
\begin{tikzcd}
\Gamma_3 \arrow[r, "\gamma_2"] \arrow[d, "\gamma_0\circ(\gamma_1\circ\gamma_2)"'] \arrow[dr, "\gamma_1\circ\gamma_2"] & \Gamma_2 \arrow[d, "\gamma_1"] \\
\Gamma_0 & \Gamma_1 \arrow[l, "\gamma_0"]
\end{tikzcd}$$

Then we can imagine following equivalent rules clearly.

.