Chapter 6: Nameless Representation of Terms

According to *TAPL* p. 75:

> In the previous chapter, we worked with terms "up to renaming of bound variables," introducing a general convention that bound variables can be renamed, at any moment, to enable substitution or because a new name is more convenient for some other reason. In effect, the "spelling" of a bound variable name is whatever we want it to be. This convention works well for discussing basic concepts and for presenting proofs cleanly, but for building an implementation we need to choose a single representation for each term; in particular, we must decide how occurrences of variables are to be represented.

Representations of Terms

We can:

- "represent variables symbolically, as we have done so far, but replace the convention about implicit renaming of bound variables with an operation that explicitly replaces bound variables with 'fresh' names during substitution as necessary to avoid capture."
- "devise some 'canonical' representation of variables and terms that does not require renaming".

We choose the second of these options, using an idea due to Nicolas de Bruijn. We replace named variables by natural numbers, where the number k stands for "the variable bound by the k'th enclosing λ."

Nameless Terms and Naming Contexts

6.1.2: Definition [Terms]: Let $\mathcal{T}$ be the smallest family of sets $\{\mathcal{T}_0, \mathcal{T}_1, \mathcal{T}_2, \ldots\}$ such that

- $k \in \mathcal{T}_n$ whenever $0 \leq k < n$;
- if $t_1 \in \mathcal{T}_n$ and $t_2 \in \mathcal{T}_n$, then $t_1\, t_2 \in \mathcal{T}_n$;
- if $t_1 \in \mathcal{T}_n$ and $n > 0$, then $\lambda.\, t_1 \in \mathcal{T}_{n-1}$.

Thus $\mathcal{T}_n$ consists of those terms whose free variables are in $\{0, \ldots, n-1\}$.

6.1.3: Definition. A *naming context* Γ has the form $x_n, x_{n-1}, \ldots, x_1, x_0$, where the x_i's are variables.

Shifting

6.2.1: Definition [Shifting]: The *d-place shift of a term t above cutoff c*, written $\uparrow_c^d (t)$, is defined as follows:

$$\uparrow_c^d (k) = \begin{cases} k & \text{if } k < c \\ k + d & \text{if } k \geq c, \end{cases}$$

$$\uparrow_c^d(t_1\, t_2) = \uparrow_c^d(t_1) \uparrow_c^d(t_2),$$

$$\uparrow_c^d(\lambda.\, t_1) = \lambda.\, \uparrow_{c+1}^d(t_1).$$

We write $\uparrow^d(t)$ for $\uparrow_0^d(t)$.

Substitution and Evaluation

6.2.4: Definition [Substitution]: The substitution of a term s for variable number j in a term t, written $[j \mapsto s]t$, is defined as follows:

$$[j \mapsto s]k = \begin{cases} s & \text{if } k = j \\ k & \text{otherwise,} \end{cases}$$

$$[j \mapsto s](t_1\, t_2) = ([j \mapsto s]t_1\, [j \mapsto s]t_2),$$

$$[j \mapsto s](\lambda.\, t_1) = \lambda.\, [j + 1 \mapsto \uparrow^1(s)]t_1.$$

Evaluation is defined as before, except that

$$(\lambda.t_{12})v_2 \to \uparrow^{-1}([0 \mapsto \uparrow^1(v_2)]t_{12}).$$

But if $(\lambda.t_{12})$ and v_2 are in $\mathcal{T}_0$, then

$$(\lambda.t_{12})v_2 \to [0 \mapsto v_2]t_{12}.$$