General properties of $\lambda\to$

Definition: 1) The *domain* $dom(\Gamma)$ of a context $\Gamma \equiv x_1:\sigma_1,\dots,x_n:\sigma_n$ is the list of variables $(x_1,\dots,x_n)$
2) Γ' is a *subcontext* of Γ ($\Gamma' \subseteq \Gamma$) if all declarations in Γ' also occur in Γ in the same order
3) Γ' is a *permutation* of Γ if all declarations in Γ' also occur in Γ and vice versa.
4) For a context Γ and a set of variables Φ, the *projection* of Γ on Φ ($\Gamma \upharpoonright \Phi$) is the subcontext Γ' of Γ satisfying $dom(\Gamma') = dom(\Gamma) \cap \Phi$

Examples: 1) $dom(\emptyset) = \emptyset$, $dom(y:\sigma, x_1:\tau_1, x_2:\tau_2) = (y, x_1, x_2)$
2) $\emptyset \subseteq x_1:\tau_1 \subseteq y:\sigma, x_1:\tau_1, x_2:\tau_2$
3) $x_1:\tau_1, y:\sigma, x_2:\tau_2$ is a permutation of $y:\sigma, x_1:\tau_1, x_2:\tau_2$
4) $y:\sigma, x_1:\tau_1, x_2:\tau_2 \upharpoonright \{y, x_2\} = y:\sigma, x_2:\tau_2$

Free variable lemma If $\Gamma \vdash L:\sigma$, then $FV(L) \subseteq dom(\Gamma)$

Proof: see lemma 2.10.3 (induction)

Thinning, Condensing, Permutation lemma (1) $\Gamma' \subseteq \Gamma''$, $\Gamma' \vdash M:\sigma \Rightarrow \Gamma'' \vdash M:\sigma$
(2) $\Gamma \vdash M:\sigma \Rightarrow \Gamma \upharpoonright FV(M) \vdash M:\sigma$
(3) $\Gamma' \vdash M:\sigma$, Γ'' permutation of Γ' $\Rightarrow \Gamma'' \vdash M:\sigma$

Remark: The above lemma implies that in $\lambda\to$ we could define a context to be a set instead of a list. In later systems lists are necessary because the order will be important.

Generation lemma: (1) $\Gamma \vdash x:\sigma \Rightarrow x:\sigma \in \Gamma$
(2) $\Gamma \vdash MN:\tau \Rightarrow \exists \sigma \in \mathbb{T}: \Gamma \vdash M:\sigma\to\tau$ and $\Gamma \vdash N:\sigma$
(3) $\Gamma \vdash \lambda x:\sigma.M:\tau \Rightarrow \exists \rho \in \mathbb{T}: \Gamma, x:\sigma \vdash M:\rho$ and $\tau \equiv \sigma\to\rho$

Subterm lemma: If M is legal then every subterm of M is legal, i.e.
$\exists \Gamma_1, \sigma_1: \Gamma_1 \vdash M:\sigma_1$, L subterm of M $\Rightarrow \exists \Gamma_2, \sigma_2: \Gamma_2 \vdash L:\sigma_2$

Example: Let $M \equiv (\lambda z:\beta.\lambda u:\gamma.z)(yx)$, $L \equiv \lambda u:\gamma.z$
$x:\alpha\to\alpha, y:(\alpha\to\alpha)\to\beta \vdash M:\gamma\to\beta$
$x:\alpha\to\alpha, y:(\alpha\to\alpha)\to\beta, z:\beta \vdash L:\gamma\to\beta$ or $z:\delta \vdash L:\gamma\to\delta$

Uniqueness of types lemma $\Gamma \vdash M:\sigma$, $\Gamma \vdash M:\tau \Rightarrow \sigma \equiv \tau$

Decidability lemma In $\lambda\to$ the following problems are decidable:
(1) Well-typedness : $? \vdash$ term : ?
(1a) Type assignment : context $\vdash$ term : ?
(2) Type checking : context $\overset{?}{\vdash}$ term : type
(3) Term finding : context $\vdash$? : type

Reduction in $\lambda\!\to$

Definition (substitution)
(1a) $x[x:=N] \equiv N$
(1b) $y[x:=N] \equiv y$ if $x \not\equiv y$
(2) $(PQ)[x:=N] \equiv (P[x:=N])(Q[x:=N])$
(3) $(\lambda y:\sigma.P)[x:=N] \equiv \lambda z:\sigma.(P^{y\to z}[x:=N])$
if $\lambda z:\sigma.P^{y\to z}$ is an α-variant of $\lambda y:\sigma.P$ such that $z \notin FV(N)$

Substitution lemma $\Gamma', x:\sigma, \Gamma'' \vdash M:\tau$ and $\Gamma' \vdash N:\sigma \Rightarrow \Gamma', \Gamma'' \vdash M[x:=N]:\tau$

Note: $\Gamma' \vdash N:\sigma$ implies that $x \notin FV(N)$ ($\Leftarrow$ Free variable lemma)

Definition (one-step β-reduction, $\to_\beta$ for Λ_T) (1) (Basis) $(\lambda x:\sigma.M)N \to_\beta M[x:=N]$
(2) (Compatibility) $M \to_\beta N \Rightarrow$ $ML \to_\beta NL$
$LM \to_\beta LN$
$\lambda x:\sigma.M \to_\beta \lambda x:\sigma.N$

Definition (zero-or-more-step β-reduction, $\twoheadrightarrow_\beta$, $=_\beta$ in Λ_T): Analogous to $\twoheadrightarrow_\beta, =_\beta$ in Λ

Church-Rosser Theorem holds for $\lambda\!\to$

Corollary $M =_\beta N \Rightarrow \exists L : M \twoheadrightarrow_\beta L,\ N \twoheadrightarrow_\beta L$

Subject reduction lemma $\Gamma \vdash L:\sigma,\ L \twoheadrightarrow_\beta L' \Rightarrow \Gamma \vdash L':\sigma$

Consequence: $\boxed{\beta\text{-reduction does not affect typability!}}$

Example: $x:\alpha\to\alpha,\ y:(\alpha\to\alpha)\to\beta \vdash \lambda u:\gamma.yx : \gamma\to\beta$

because $x:\alpha\to\alpha,\ y:(\alpha\to\alpha)\to\beta \vdash (\lambda z:\beta.\lambda u:\gamma.z)(yx) : \gamma\to\beta$

Strong normalisation theorem: Every legal term is strongly normalising

Consequences:
1) There is no self-application in $\lambda\!\to$ (Generation lemma)
2) Existence of β-nfs is guaranteed (Strong normalisation theorem)
3) Not every legal λ-term has a fixed point
Proof: Let F legal s.t. $\Gamma \vdash F:\sigma\to\tau$ with $\sigma \not\equiv \tau$ and assume $\exists M$ legal: $FM =_\beta M$
Then, $M:\sigma$, $FM:\tau$. Consequently, $\exists N : FM \twoheadrightarrow_\beta N,\ M \twoheadrightarrow_\beta N \Rightarrow$ (subject reduction)
$\Gamma \vdash N:\tau,\ \Gamma \vdash N:\sigma$ ↯

$\lambda\!\to$ is not Turing complete, but natural numbers and addition and multiplication can be defined:

$n \equiv \lambda f:\alpha\to\alpha.\lambda x:\alpha.\underbrace{f(\ldots(f\,x)\ldots)}_{n\text{-times}}$, $n : (\alpha\to\alpha)\to\alpha\to\alpha$

Class of functions definable in $\lambda\!\to$ (on natural numbers) = generalised polynomials,
i.e. piecewise polynomial functions where cases are defined by whether variables do or do not vanish,
e.g.:
$$f(m,n) = \begin{cases} k & \text{if } m=0,\ n=0 \\ P_1(n) & \text{if } m=0,\ n\neq 0 \\ P_2(m) & \text{if } m\neq 0,\ n=0 \\ P_3(m,n) & \text{if } m\neq 0,\ n\neq 0 \end{cases}$$

Make $\lambda\!\to$ Turing complete by introducing Y-combinators: $Y_\sigma : (\sigma\to\sigma)\to\sigma$, $Y_\sigma f \to f(Y_\sigma f)$