Az F_2 típusrendszer

F_2: bevezetés

Polimorfikus λ-kalkulus (PLC, $\lambda 2$), F-típusrendszer (System F)

F_1:

$$
\begin{array}{lcl}
\texttt{id\_Nat} & \equiv & \lambda x : Nat.x \\
\texttt{id\_Int} & \equiv & \lambda x : Int.x \\
\texttt{id\_Bool} & \equiv & \lambda x : Bool.x \\
\texttt{id\_Type}_1 & \equiv & \lambda x : Nat \to Nat.x \\
\texttt{id\_Type}_2 & \equiv & \lambda x : (Nat \to Nat) \to (Nat \to Nat).x
\end{array}
$$

Mivel ezek a függvényekben csak a típusok különböznek, jelöljük a típust egy α típusváltozóval:

$\texttt{id\_}\alpha$	$\equiv$	$\lambda x : \alpha.x$	típusváltozó
$\texttt{id}$	$\equiv$	$\Lambda\alpha.\lambda x : \alpha.x$	típusabsztrakció (polimorf függvény)
$\texttt{id}$	$:$	$\forall\alpha.\alpha \to \alpha$	polimorf típus
$\texttt{id}\ [Nat]$	$\equiv$	$\texttt{id\_Nat}$	típusalkalmazás (példányosítás)

$$\texttt{id}\ [Nat] \equiv (\Lambda\alpha.\lambda x : \alpha.x)\ [Nat] \twoheadrightarrow \lambda x : Nat.x \equiv \texttt{id\_Nat}$$

F_2: szintaktika

Definíció. ***Az F_2 típusrendszer szintaktikája***

<típus>	::=	*<típusváltozó>*
	|	*(<típus> → <típus>)*
	|	*(∀ <típusváltozó> . <típus>)*
<λ-kifejezés>	::=	*<változó>*
	|	*(λ <változó> : <típus> . <λ-kifejezés>)*
	|	*(<λ-kifejezés> <λ-kifejezés>)*
	|	*(Λ <típusváltozó> . <λ-kifejezés>)*
	|	*(<λ-kifejezés> [<típus>])*

$\forall\alpha.(\alpha \to \alpha)$	$\equiv$	$\forall\alpha.\alpha \to \alpha$	$\alpha, \beta, \ldots$: típusváltozók
$\Lambda\alpha.(\Lambda\beta.F)$	$\equiv$	$\Lambda\alpha.\Lambda\beta.F$	τ: típus
$(E[\tau'])[\tau'']$	$\equiv$	$E[\tau'][\tau'']$	$\forall \ldots$: polimorf típus
$\Lambda\alpha.(E[\tau])$	$\equiv$	$\Lambda\alpha.E[\tau]$	$\forall\alpha.\tau$: τ az α hatásköre

Az F_1 és a Hindley-Milner-típusrendszerek típusainak általánosítása

F_2: szabad változók

Definíció. ***A szabad változó***

$$
\begin{aligned}
FV(x) &= \{x\} \\
FV(E\ F) &= FV(E) \cup FV(F) \\
FV(\lambda x : \tau . E) &= FV(E) \setminus \{x\} \\
FV(\Lambda \alpha . E) &= FV(E) \\
FV(E\ [\tau]) &= FV(E)
\end{aligned}
$$

Definíció. ***A szabad típusváltozó***

$$
\begin{aligned}
FTV(\alpha) &= \{\alpha\} \\
FTV(\tau' \to \tau'') &= FTV(\tau') \cup FTV(\tau'') \\
FTV(\forall \alpha . \tau) &= FTV(\tau) \setminus \{\alpha\}
\end{aligned}
$$

$$FTV(\alpha \to \beta) = \{\alpha, \beta\}$$
$$FTV(\forall \alpha . \alpha \to \beta) = \{\beta\}$$
$$FTV((\forall \alpha . \alpha \to \beta) \to (\forall \beta . \alpha \to \beta)) = \{\beta, \alpha\}$$

F_2: típuskörnyezet

Definíció. ***A típuskörnyezet szintaktikája***

$$\begin{array}{lcl} \textit{<típuskörnyezet>} & ::= & \emptyset \\ & | & \textit{<típuskörnyezet>, <változó> : <típus>} \\ & | & \textit{<típuskörnyezet>, <típusváltozó>} \end{array}$$

valamint a következő tulajdonságokat követeljük meg:

- *ha* $x_1 : \tau_1, x_2 : \tau_2 \in \Gamma$, *akkor* $x_1 \neq x_2$
- *ha* $\alpha_1, \alpha_2 \in \Gamma$, *akkor* $\alpha_1 \neq \alpha_2$

$dom(\Gamma) = \Gamma_V \cup \Gamma_{TV}$, ahol $\Gamma_V = \{x \mid x : \tau \in \Gamma\}$

Definíció. ***A típuskörnyezet szabad típusváltozói***

$FTV(\Gamma) = \{\alpha \mid \{x : \tau\} \in \Gamma, \alpha \in FTV(\tau)\}$

Definíció. ***Jól formált típuskörnyezet***

A Γ *típuskörnyezet jól formált, ha szintaktikusan helyes és* $FTV(\Gamma) \subseteq \Gamma_{TV}$.

F_2: típuskörnyezet, következtetések

Példák a típuskörnyezet jól formáltságának vizsgálatára:

$\Gamma = \{x : \alpha \to \beta, y : \forall\gamma.\gamma \to \delta\}$
$FTV(\Gamma) = \{\alpha, \beta, \delta\}$, $\Gamma_{TV} = \varnothing$

$\Gamma = \{\alpha, \beta, \delta, \varphi, \psi, x : \alpha \to \beta, y : \forall\gamma.\gamma \to \delta\}$
$FTV(\Gamma) = \{\alpha, \beta, \delta\}$, $\Gamma_{TV} = \{\alpha, \beta, \delta, \varphi, \psi\}$

Definíció. ***Az F_2 típusrendszer következtetései***

$\Gamma \vdash wf$	*Γ jól formált típuskörnyezet*
$\Gamma \vdash \tau$	*a Γ környezetben τ egy jól formált típus*
$\Gamma \vdash E : \tau$	*a Γ környezetben az E kifejezés típusa τ*

F_2: a típusrendszer szabályai (1/2)

$$\frac{}{\varnothing \vdash \mathsf{wf}}\ (\text{ENV } \varnothing) \qquad \frac{\Gamma \vdash \tau,\, x \notin dom(\Gamma)}{\Gamma, x : \tau \vdash \mathsf{wf}}\ (\text{ENV } x)$$

$$\frac{\Gamma \vdash \mathsf{wf},\, \alpha \notin dom(\Gamma)}{\Gamma, \alpha \vdash \mathsf{wf}}\ (\text{ENV } \alpha) \qquad \frac{\Gamma', \alpha, \Gamma'' \vdash \mathsf{wf}}{\Gamma', \alpha, \Gamma'' \vdash \alpha}\ (\text{TYPE } \alpha)$$

$$\frac{\Gamma \vdash \tau' \qquad \Gamma \vdash \tau''}{\Gamma \vdash \tau' \to \tau''}\ (\text{TYPE ARROW}) \qquad \frac{\Gamma, \alpha \vdash \tau}{\Gamma \vdash \forall \alpha . \tau}\ (\text{TYPE FORALL})$$

F_2: a típusrendszer szabályai (2/2)

$$\frac{\Gamma', x : \tau, \Gamma'' \vdash \mathrm{wf}}{\Gamma', x : \tau, \Gamma'' \vdash x : \tau} \text{ (VAL } x\text{)} \qquad \frac{\Gamma, x : \tau' \vdash E : \tau''}{\Gamma \vdash \lambda x : \tau'.E : \tau' \to \tau''} \text{ (VAL ABS)}$$

$$\frac{\Gamma \vdash E : \tau' \to \tau'' \qquad \Gamma \vdash F : \tau'}{\Gamma \vdash E\ F : \tau''} \text{ (VAL APPL)}$$

$$\frac{\Gamma, \alpha \vdash E : \tau \quad (\alpha \notin FTV(\Gamma))}{\Gamma \vdash \Lambda\alpha.E : \forall\alpha.\tau} \text{ (TYPE ABS)}$$

$$\frac{\Gamma \vdash E : \forall\alpha.\tau' \qquad \Gamma \vdash \tau''}{\Gamma \vdash E[\tau''] : \tau'[\alpha := \tau'']} \text{ (TYPE APPL)}$$

Hindley-Milner:

gen(Γ, τ) = TYPE FORALL + ENV α + TYPE α inst(τ) = TYPE APPL

F_2: példák típusokra

Az F_2 típusrendszerben a λ-absztrakciók már rendelkezhetnek polimorf változókkal, nincs szükség a `let` utasításra.

`id` $(\lambda x.x)$
$\varnothing \vdash (\lambda x.x \equiv \Lambda\alpha.\lambda x : \alpha.x) : \forall\alpha.\alpha \to \alpha$

`twice` $(\lambda f\ \lambda x.f\ (f\ x))$
$\varnothing \vdash (\Lambda\alpha.\lambda f : \alpha \to \alpha.\lambda x : \alpha.f\ (f\ x)) : \forall\alpha.(\alpha \to \alpha) \to \alpha \to \alpha$

önalkalmazás $(\lambda x.x\ x)$
$\varnothing \vdash (\lambda x : \forall\alpha.\alpha \to \alpha.x[\forall\alpha.\alpha \to \alpha]\ x) : (\forall\alpha.\alpha \to \alpha) \to (\forall\alpha.\alpha \to \alpha)$

Az F_2 típusrendszerben már megjelenik az önapplikáció típusozása!

F_2: példák típuslevezetésekre

$\texttt{id}\ (\lambda x.x)$

$\varnothing \vdash (\lambda x.x \equiv \Lambda\alpha.\lambda x:\alpha.x) : \forall\alpha.\alpha \to \alpha$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\varnothing \vdash \mathsf{wf}, \alpha \notin dom(\Gamma)}}{\alpha \vdash \mathsf{wf}}}{\alpha \vdash \alpha, x \notin dom(\Gamma)}}{\alpha, x:\alpha \vdash \mathsf{wf}}}{\alpha, x:\alpha \vdash x:\alpha}}{\alpha \vdash (\lambda x:\alpha.x):\alpha \to \alpha}$$

$$\dfrac{\alpha \vdash (\lambda x:\alpha.x):\alpha \to \alpha}{\varnothing \vdash (\Lambda\alpha.\lambda x:\alpha.x):\forall\alpha.\alpha \to \alpha}$$

F_2: példák típuslevezetésekre

`twice` $(\lambda f\ \lambda x.f\ (f\ x))$
$\varnothing \vdash (\Lambda\alpha.\lambda f : \alpha \to \alpha.\lambda x : \alpha.f\ (f\ x)) : \forall\alpha.(\alpha \to \alpha) \to \alpha \to \alpha$

A függvény F_1-beli alakja:
$(\lambda f : A \to A.\lambda x : A.f\ (f\ x)) : (A \to A) \to A \to A$

Helyettesítsük az A típust az α típusváltozóra:
$\lambda f : \alpha \to \alpha.\lambda x : \alpha.f\ (f\ x)$

Típusabsztrakció bevezetése:
$\Lambda\alpha.\lambda f : \alpha \to \alpha.\lambda x : \alpha.f\ (f\ x)$

A polimorf függvény típusa:
$\forall\alpha.(\alpha \to \alpha) \to \alpha \to \alpha$

F_2: operációs szemantika, típusváltozók helyettesítése

Definíció. ***Típusváltozó helyettesítése típusban***

1. $$\alpha[\beta := \tau] \equiv \begin{cases} \tau & \text{ha } \alpha \equiv \beta \\ \alpha & \text{egyébként} \end{cases}$$
2. $$(\tau' \to \tau'')[\beta := \tau'] \equiv \tau'[\beta := \tau] \to \tau''[\beta := \tau]$$
3. $$(\forall\alpha.\tau')[\beta := \tau''] \equiv \begin{cases} \forall\alpha.\tau'[\beta := \tau''] & \text{ha } \alpha \not\equiv \beta, \alpha \notin FTV(\tau'') \\ \forall\alpha.\tau' & \text{egyébként} \end{cases}$$

Definíció. ***Típusváltozó helyettesítése típusabsztrakcióban***

$$(\Lambda\alpha.E)[\beta := \tau] \equiv \begin{cases} \Lambda\alpha.E & \text{ha } \alpha \equiv \beta \\ \Lambda\alpha.E[\beta := \tau] & \text{ha } \alpha \not\equiv \beta, \alpha \notin FTV(\tau) \\ \Lambda\alpha.E & \text{egyébként} \end{cases}$$

F_2: operációs szemantika, α-konverzió

Definíció. ***Típusok α-konverziója***

Ha $\beta \notin FTV(\tau)$, akkor $\forall\alpha.\tau \leftrightarrow_\alpha \forall\beta.\tau[\alpha := \beta]$.

Definíció. ***Az α-konverzió***

Ha $\beta \notin FTV(E)$, akkor $\Lambda\alpha.E \leftrightarrow \Lambda\beta.E[\alpha := \beta]$.

Lemma. ***Az α-konverzió típusszabálya***

$$\frac{\Gamma \vdash \Lambda\alpha.E : \forall\alpha.\tau, \beta \notin FTV(E), \beta \notin dom(\Gamma)}{\Gamma \vdash \Lambda\beta.E[\alpha := \beta] : \forall\beta.\tau[\alpha := \beta]} \text{ (CONV } \alpha\text{)}$$

F_2: operációs szemantika, kötés, típusalkalmazás

Tétel. ***Kötött típusváltozó helyettesítése a típusban***

Ha $\forall\alpha.\tau \leftrightarrow_\alpha \forall\beta.\tau[\alpha := \beta]$, *akkor* $\Gamma \vdash E : \forall\alpha.\tau$ *esetén* $\Gamma \vdash E : \forall\beta.\tau[\alpha := \beta]$ *is fennáll.*

Tétel. ***Kötött típusváltozó helyettesítése a környezetben***

Ha $\forall\alpha.\tau \leftrightarrow_\alpha \forall\beta.\tau[\alpha := \beta]$, *akkor* $\Gamma, \{x : \forall\alpha.\tau\} \vdash E : \tau'$ *esetén* $\Gamma, \{x : \forall\beta.\tau[\alpha := \beta]\} \vdash E : \tau'$ *is teljesül.*

Tétel. ***A típusalkalmazás β-redukciója***

Ha $FTV(\tau) \subseteq FTV(E[\alpha := \tau])$, *akkor* $(\Lambda\alpha.E)\ [\tau] \rightarrow_\beta E[\alpha := \tau]$.

Tétel. ***A típusalkalmazás β-redukciójának típusszabálya***

$$\frac{\Gamma \vdash (\Lambda\alpha.E)[\tau] : \tau', FTV(\tau) \subseteq FTV(E[\alpha := \tau])}{\Gamma \vdash E[\alpha := \tau] : \tau'} \ (\text{CONV } \beta)$$

F_2: példák típusalkalmazásokra

$\texttt{id} \equiv \Lambda\alpha.\lambda x : \alpha.x : \forall\alpha.\alpha \to \alpha$

$\texttt{id}\ [\mathit{Nat}] \equiv (\Lambda\alpha.\lambda x : \alpha.x)\ [\mathit{Nat}] \twoheadrightarrow_\beta \lambda x : \mathit{Nat}.x \equiv \texttt{id\_Nat}$
$\texttt{id}\ [\mathit{Int}] \twoheadrightarrow_\beta \texttt{id\_Int}$
$\texttt{id}\ [\mathit{Bool}] \twoheadrightarrow_\beta \texttt{id\_Bool}$
$\texttt{id}\ [\mathit{Nat}]\ 3 \equiv (\Lambda\alpha.\lambda x : \alpha.x)[\mathit{Nat}]\ 3 \twoheadrightarrow_\beta (\lambda x : \mathit{Nat}.x)\ 3 \equiv$
$\quad \texttt{id\_Nat}\ 3 \twoheadrightarrow_\beta 3$

$\texttt{twice} \equiv \Lambda\alpha.\lambda f : \alpha \to \alpha.\lambda x : \alpha.f\ (f\ x) : \forall\alpha.(\alpha \to \alpha) \to \alpha \to \alpha$

$\texttt{twice}\ [\mathit{Nat}] : (\mathit{Nat} \to \mathit{Nat}) \to \mathit{Nat} \to \mathit{Nat}$
$\texttt{twice}\ [\mathit{Nat} \to \mathit{Nat}] :$
$((\mathit{Nat} \to \mathit{Nat}) \to (\mathit{Nat} \to \mathit{Nat})) \to (\mathit{Nat} \to \mathit{Nat}) \to \mathit{Nat} \to \mathit{Nat}$

F_2: tulajdonságok

Tétel. ***Az erős normalizálás tétele***

Az F_2 típusrendszerben az $E : \tau$ jól típusozott kifejezésnek nincs végtelen β-redukciós sorozata.

Tétel. ***A normalizálás tétele***

Az F_2 típusrendszerben minden jól típusozott λ-kifejezésnek van normálformája.

Tétel. ***A típusmegmaradás tétele***

Ha $\Gamma \vdash E : \tau$ és $E \twoheadrightarrow F$, akkor $\Gamma \vdash F : \tau$.

Tétel. ***A haladás tétele***

Ha E egy jól típusozott, zárt kifejezés, akkor

- *E egy érték (változó, absztrakció), vagy*
- *van olyan F kifejezés, hogy $E \twoheadrightarrow F$.*

Típusok az F_2 típusrendszerben

F_2: az *Absurd* típus

$Absurd \equiv \forall\alpha.\alpha$

$Absurd := \emptyset$

Nincs olyan E kifejezés, hogy $\Lambda\alpha.E : \forall\alpha.\alpha$. Ekkor E típusa α, ezért:

- $E \not\equiv \lambda x. \ldots$, mert ekkor E típusa $\beta \to \tau$ alakú,
- $E \not\equiv \Lambda\beta. \ldots$, mert ekkor E típusa $\forall\beta \ldots$ alakú,
- $E \not\equiv x$, mert ekkor x szabad változó

lenne.

F_2: a *Unit* típus

$$Unit \equiv \forall\alpha.\alpha \to \alpha$$

$$Unit := \{\mathtt{unit}\}$$

$$\mathtt{unit} \equiv \underbrace{\Lambda\alpha}_{\forall\alpha} . \underbrace{\lambda x : \alpha}_{\alpha} . \underbrace{x}_{\alpha}$$

F_2: a *Bool* típus

$$Bool \equiv \forall\alpha.\alpha \to \alpha \to \alpha$$

$$Bool := \{\texttt{true}, \texttt{false}\}$$

$$\begin{array}{lcl} \texttt{true} & \equiv & \Lambda\alpha.\lambda x : \alpha.\lambda y : \alpha.x \\ \texttt{false} & \equiv & \Lambda\alpha.\lambda x : \alpha.\lambda y : \alpha.y \end{array}$$

$\texttt{if} \equiv \lambda x : Bool.\Lambda\alpha.\lambda y : \alpha.\lambda z : \alpha.x\ [\alpha]\ y\ z$
$\texttt{if} : Bool \to (\forall\alpha.\alpha \to \alpha \to \alpha)$

$\texttt{if}\ \texttt{true}\ [Nat]\ 1\ 2 \equiv$
$(\lambda x : Bool.\Lambda\alpha.\lambda y : \alpha.\lambda z : \alpha.x\ [\alpha]\ y\ z)\ \texttt{true}\ [Nat]\ 1\ 2 \twoheadrightarrow_\beta$
$(\Lambda\alpha.\lambda y : \alpha.\lambda z : \alpha.\texttt{true}\ [\alpha]\ y\ z)\ [Nat]\ 1\ 2 \twoheadrightarrow_\beta$
$(\lambda y : Nat.\lambda z : Nat.\texttt{true}\ [Nat]\ y\ z)\ 1\ 2 \twoheadrightarrow_\beta^+$
$\texttt{true}\ [Nat]\ 1\ 2 \equiv (\Lambda\alpha.\lambda x : \alpha.\lambda y : \alpha.x)\ [Nat]\ 1\ 2 \twoheadrightarrow_\beta$
$(\lambda x : Nat.\lambda y : Nat.x)\ 1\ 2 \twoheadrightarrow_\beta^+ 1$

F_2: a *Bool* típus

$\texttt{and} \equiv \lambda x : Bool.\lambda y : Bool.\Lambda\alpha.\lambda u : \alpha.\lambda v : \alpha.x\ [\alpha]\ (y\ [\alpha]\ u\ v)\ v$
$\texttt{and} : Bool \to Bool \to Bool$

$\texttt{or} \equiv \lambda x : Bool.\lambda y : Bool.\Lambda\alpha.\lambda u : \alpha.\lambda v : \alpha.x\ [\alpha]\ u\ (y\ [\alpha]\ u\ v)$
$\texttt{or} : Bool \to Bool \to Bool$

$\texttt{not} \equiv \lambda x : Bool.\Lambda\alpha.\lambda y : \alpha.\lambda z : \alpha.x\ [\alpha]\ z\ y$
$\texttt{not} : Bool \to Bool$

$\texttt{not}\ \texttt{true} \equiv$
$(\lambda x : Bool.\Lambda\beta.\lambda y : \beta.\lambda z : \beta.x\ [\beta]\ z\ y)\ (\Lambda\alpha.\lambda x : \alpha.\lambda y : \alpha.x) \twoheadrightarrow_\beta$
$\Lambda\beta.\lambda y : \beta.\lambda z : \beta.(\Lambda\alpha.\lambda x : \alpha.\lambda y : \alpha.x)\ [\beta]\ z\ y \twoheadrightarrow_\beta$
$\Lambda\beta.\lambda y : \beta.\lambda z : \beta.(\lambda x : \beta.\lambda y : \beta.x)\ z\ y \twoheadrightarrow_\beta^+$
$\Lambda\beta.\lambda y : \beta.\lambda z : \beta.z \leftrightarrow_\alpha \texttt{false}$

F_2: a *Nat* típus

$$Nat \equiv \forall\alpha.(\alpha \to \alpha) \to \alpha \to \alpha$$

$$Nat := \{c_0, c_1, c_2, c_3, \ldots\}$$

$$\begin{array}{lcl}
c_0 & \equiv & \Lambda\alpha.\lambda f:\alpha\to\alpha.\lambda x:\alpha.x \\
c_1 & \equiv & \Lambda\alpha.\lambda f:\alpha\to\alpha.\lambda x:\alpha.f\ x \\
c_2 & \equiv & \Lambda\alpha.\lambda f:\alpha\to\alpha.\lambda x:\alpha.f\ (f\ x) \\
c_3 & \equiv & \Lambda\alpha.\lambda f:\alpha\to\alpha.\lambda x:\alpha.f\ (f\ (f\ x)) \\
 & \ldots & \\
c_n & \equiv & \Lambda\alpha.\lambda f:\alpha\to\alpha.\lambda x:\alpha.f^n(x)
\end{array}$$

$\texttt{succ} \equiv \lambda n:Nat.\Lambda\alpha.\lambda f:\alpha\to\alpha.\lambda x:\alpha.f\ (n\ [\alpha]\ f\ x)$
$\texttt{succ} : Nat \to Nat$

$\texttt{zero} \equiv \lambda x:Nat.x[Bool]\ (\lambda y:Bool.\texttt{false})\ \texttt{true}$
$\texttt{zero} : Nat \to Bool$

F_2: a *Nat* típus

$$
\begin{aligned}
\texttt{add} &\equiv \lambda x : Nat.\lambda y : Nat.\Lambda\alpha.\lambda p : \alpha \to \alpha.\lambda q : \alpha.x\ [\alpha]\ p\ (y\ [\alpha]\ p\ q) \\
\texttt{mul} &\equiv \lambda x : Nat.\lambda y : Nat.\Lambda\alpha.\lambda p : \alpha \to \alpha.y\ [\alpha]\ (x\ [\alpha]\ p) \\
\texttt{exp} &\equiv \lambda x : Nat.\lambda y : Nat.\Lambda\alpha.y\ [\alpha \to \alpha]\ (x\ [\alpha])
\end{aligned}
$$

$\texttt{exp}\ 1\ 2 \equiv$
$(\lambda x : Nat.\lambda y : Nat.\Lambda\alpha.y\ [\alpha \to \alpha]\ (x\ [\alpha]))\ 1\ 2 \twoheadrightarrow_\beta^+$
$\Lambda\alpha.2\ [\alpha \to \alpha]\ (1\ [\alpha]) \equiv$
$\Lambda\alpha.(\Lambda\alpha.\lambda f : \alpha \to \alpha.\lambda x : \alpha.f\ (f\ x))\ [\alpha \to \alpha]\ (1\ [\alpha]) \twoheadrightarrow_\beta$
$\Lambda\alpha.(\lambda f : (\alpha \to \alpha) \to (\alpha \to \alpha).\lambda x : \alpha \to \alpha.f\ (f\ x))\ (1\ [\alpha]) \twoheadrightarrow_\beta$
$\ldots$
$\Lambda\alpha.\lambda f : \alpha \to \alpha.\lambda x : \alpha.f\ x) \equiv 1$

F_2: a *Pair* típus

$$Pair \equiv \Lambda\alpha.\Lambda\beta.\forall\gamma.(\alpha \to \beta \to \gamma) \to \gamma$$

$$Pair := \{\texttt{pair}\ x\ y\}$$

$$\texttt{pair} \equiv \Lambda\alpha.\Lambda\beta.\lambda x : \alpha.\lambda y : \beta.\Lambda y.\lambda z : \alpha \to \beta \to \gamma.z\ x\ y$$
$$\texttt{pair} : \forall\alpha.\forall\beta.\alpha \to \beta \to Pair_{\alpha\times\beta}$$

$$\texttt{first} \equiv \Lambda\alpha.\Lambda\beta.\lambda z : Pair_{\alpha\times\beta}.z\ [\alpha]\ (\lambda x : \alpha.\lambda y : \beta.x)$$
$$\texttt{first} : \forall\alpha.\forall\beta.Pair_{\alpha\times\beta} \to \alpha$$

$$\texttt{second} \equiv \Lambda\alpha.\Lambda\beta.\lambda z : Pair_{\alpha\times\beta}.z\ [\beta]\ (\lambda x : \alpha.\lambda y : \beta.y)$$
$$\texttt{second} : \forall\alpha.\forall\beta.Pair_{\alpha\times\beta} \to \beta$$

$$\begin{aligned}
&\texttt{first}_{\alpha\times\beta}\ (\texttt{pair}_{\alpha\times\beta}\ v\ w) \equiv\\
&(\lambda x : Pair_{\alpha\times\beta}.x\ [\alpha]\ (\lambda y : \alpha.\lambda z : \beta.y))\ (\texttt{pair}_{\alpha\times\beta}\ v\ w) \twoheadrightarrow\\
&(\texttt{pair}_{\alpha\times\beta}\ v\ w)\ [\alpha]\ (\lambda y : \alpha.\lambda z : \beta.y) \equiv\\
&(\Lambda\gamma.\lambda z : \alpha \to \beta \to \gamma.z\ v\ w)\ [\alpha]\ (\lambda y : \alpha.\lambda z : \beta.y) \twoheadrightarrow^{+}\\
&(\lambda y : \alpha.\lambda z : \beta.y)\ v\ w \twoheadrightarrow^{+} v
\end{aligned}$$

F_2: a *List* típus

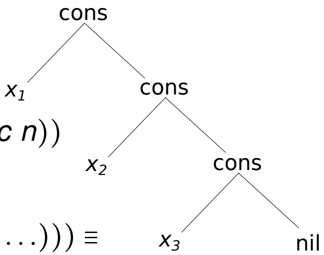
$$List \equiv \Lambda\alpha.\forall\beta.(\alpha \to \beta \to \beta) \to \beta \to \beta$$

$$List := \{\texttt{nil}, \texttt{cons}\ x\ xs\}$$

$$\texttt{nil} \equiv \Lambda\alpha.(\Lambda\beta.\lambda c : \alpha \to \beta \to \beta.\lambda n : \beta.n)$$
$$\texttt{nil} : \forall\alpha.List_\alpha$$

$$\texttt{cons} \equiv \Lambda\alpha.\lambda h : \alpha.\lambda t : (List\ [\alpha]).$$
$$(\Lambda\beta.\lambda c : \alpha \to \beta \to \beta.\lambda n : \beta.c\ h\ (t\ [\beta]\ c\ n))$$
$$\texttt{cons} : \forall\alpha.\alpha \to List_\alpha \to List_\alpha$$

$$[x_1, x_2, \ldots, x_n] \equiv (x_1 : (x_2 : (\ldots : (x_n : [\,])\ldots))) \equiv$$
$$c\ x_1\ (c\ x_2\ (\ldots\ (c\ x_m\ n)\ \ldots))$$



F_2: a *List* típus

$$\mathtt{cons}\ [Nat]\ 3\ \mathtt{nil}_{Nat} \equiv$$
$$(\Lambda\alpha.\lambda h:\alpha.\lambda t:List_\alpha.(\Lambda\beta.\lambda c:\alpha\to\beta\to\beta.\lambda n:\beta.c\ h\ (t\ [\beta]\ c\ n)))$$
$$[Nat]\ 3\ \mathtt{nil}_{Nat} \twoheadrightarrow^+$$
$$\Lambda\beta.\lambda c:Nat\to\beta\to\beta.\lambda n:\beta.c\ 3\ (\mathtt{nil}_{Nat}\ [\beta]\ c\ n)$$

$$\mathtt{nil}_{Nat}\ [\beta]\ c\ n \equiv (\Lambda\beta.\lambda c:Nat\to\beta\to\beta.\lambda n:\beta.n)\ [\beta]\ c\ n \twoheadrightarrow^+ n$$

$$\mathtt{cons}\ 3\ \mathtt{nil}_{Nat} \equiv$$
$$\Lambda\beta.\lambda c:Nat\to\beta\to\beta.\lambda n:\beta.c\ 3\ n$$
$$\mathtt{cons}\ 4\ (\mathtt{cons}\ 3\ \mathtt{nil}_{Nat}) \equiv$$
$$\Lambda\beta.\lambda c:Nat\to\beta\to\beta.\lambda n:\beta.c\ 4\ (c\ 3\ n)$$
$$\mathtt{cons}\ 5\ (\mathtt{cons}\ 4\ (\mathtt{cons}\ 3\ \mathtt{nil}_{Nat})) \equiv$$
$$\Lambda\beta.\lambda c:Nat\to\beta\to\beta.\lambda n:\beta.c\ 5\ (c\ 4\ (c\ 3\ n))$$

F_2: a *List* típus

$$
\begin{aligned}
\texttt{null} &\equiv \Lambda\alpha.\lambda x : List_\alpha.x\ [Bool]\ (\lambda y : \alpha.\lambda z : Bool.\texttt{false})\ \texttt{true} \\
\texttt{head} &\equiv \Lambda\alpha.\lambda x : List_\alpha.(x\ [\alpha]\ (\lambda y : \alpha.\lambda z : \alpha.y)\ \texttt{error}_\alpha) \\
\texttt{length} &\equiv \Lambda\alpha.\lambda x : List_\alpha.x\ [Nat]\ (\lambda y : \alpha.\lambda z : Nat. +\ z\ 1)\ 0 \\
\texttt{sum} &\equiv \Lambda\alpha.\lambda x : List_{Nat}.x\ [Nat]\ (\lambda y : Nat.\lambda z : Nat. +\ y\ z)\ 0
\end{aligned}
$$

$$
\begin{aligned}
&\texttt{null}\ [\alpha]\ \texttt{nil}_\alpha \equiv \\
&(\Lambda\alpha.\lambda x : List_\alpha.x\ [Bool]\ (\lambda y : \alpha.\lambda z : Bool.\texttt{false})\ \texttt{true})\ [\alpha] \\
&\quad (\Lambda\beta.\lambda c : \alpha \to \beta \to \beta.\lambda n : \beta.n) \twoheadrightarrow \\
&(\lambda x : List_\alpha.x\ [Bool]\ (\lambda y : \alpha.\lambda z : Bool.\texttt{false})\ \texttt{true}) \\
&\quad (\Lambda\beta.\lambda c : \alpha \to \beta \to \beta.\lambda n : \beta.n) \twoheadrightarrow \\
&(\Lambda\beta.\lambda c : \alpha \to \beta \to \beta.\lambda n : \beta.n)\ [Bool] \\
&\quad (\lambda y : \alpha.\lambda z : Bool.\texttt{false})\ \texttt{true} \twoheadrightarrow \\
&(\lambda c : \alpha \to Bool \to Bool.\lambda n : Bool.n) \\
&\quad (\lambda y : \alpha.\lambda z : Bool.\texttt{false})\ \texttt{true} \twoheadrightarrow^{+} \texttt{true}
\end{aligned}
$$