Ocaml

KOSMOS

Automatic Type Inference

$$\underbrace{\texttt{fun }\underbrace{\texttt{x}}_{t_1}\texttt{ -> }\underbrace{\texttt{x + 1}}_{t_2}}_{t_0}$$

Equations	Substitutions
$t_0 = t_1 \rightarrow t_2$	

Automatic Type Inference

$$\texttt{fun x -> } \underbrace{\underbrace{\texttt{x}}_{t_3} \texttt{ + } \underbrace{\texttt{1}}_{t_4}}_{t_2}$$

Equations	Substitutions
$t_0 = t_1 \rightarrow t_2$ $t_2 = int$ $t_3 = int$ $t_4 = int$	

Automatic Type Inference

```
fun x -> x + 1
```

t_1

t_3

Equations	Substitutions
$t_0 = t_1 \to t_2$ $t_2 = int$ $t_3 = int$ $t_4 = int$ $t_3 = t_1$	

Automatic Type Inference

```
fun x -> x + 1
```

Equations	Substitutions
$\boxed{t_0 = t_1 \rightarrow t_2}$ $t_2 = int$ $t_3 = int$ $t_4 = int$ $t_3 = t_1$	$t_0 = t_1 \rightarrow t_2$

Automatic Type Inference

```
fun x -> x + 1
```

Equations	Substitutions
$t_0 = t_1 \to t_2$	$t_0 = t_1 \to \boxed{int}$
$\boxed{t_2 = int}$	$t_2 = int$
$t_3 = int$	
$t_4 = int$	
$t_3 = t_1$	

Automatic Type Inference

```
fun x -> x + 1
```

Equations	Substitutions
$t_0 = t_1 \rightarrow t_2$ $t_2 = int$ $\boxed{t_3 = int}$ $t_4 = int$ $t_3 = t_1$	$t_0 = t_1 \rightarrow int$ $t_2 = int$ $t_3 = int$

Automatic Type Inference

```
fun x -> x + 1
```

Equations	Substitutions
$t_0 = t_1 \to t_2$	$t_0 = t_1 \to int$
$t_2 = int$	$t_2 = int$
$t_3 = int$	$t_3 = int$
$\boxed{t_4 = int}$	$t_4 = int$
$t_3 = t_1$	

Automatic Type Inference

`fun x -> x + 1`

Equations	Substitutions
$t_0 = t_1 \to t_2$	$t_0 = t_1 \to int$
$t_2 = int$	$t_2 = int$
$t_3 = int$	$t_3 = int$
$t_4 = int$	$t_4 = int$
$\boxed{t_3 = t_1} \to int = t_1$	

Automatic Type Inference

```
fun x -> x + 1
```

Equations	Substitutions
$t_0 = t_1 \rightarrow t_2$	$t_0 = \boxed{int} \rightarrow int$
$t_2 = int$	$t_2 = int$
$t_3 = int$	$t_3 = int$
$t_4 = int$	$t_4 = int$
$t_3 = t_1 \longrightarrow \boxed{int = t_1} \longrightarrow$	$t_1 = int$

Small Ocaml Syntax

$$
\begin{aligned}
F &\rightarrow E \\
E &\rightarrow n \\
&\mid x \\
&\mid true \\
&\mid false \\
&\mid E + E \\
&\mid E - E \\
&\mid E * E \\
&\mid E / E \\
&\mid iszero\ E \\
&\mid if\ E\ then\ E\ else\ E \\
&\mid let\ [rec]\ x = E\ in\ E \\
&\mid fun\ x \rightarrow E \\
&\mid E\ E \\
&\mid read
\end{aligned}
$$

Small Ocaml Typing Rule

$$\frac{}{\Gamma \vdash n : int} \quad \frac{}{\Gamma \vdash x : \Gamma(x)} \quad \frac{}{\Gamma \vdash true : bool} \quad \frac{}{\Gamma \vdash false : bool} \quad \frac{}{\Gamma \vdash read : int}$$

$$\frac{\Gamma \vdash E_1 : int \quad \Gamma \vdash E_2 : int}{\Gamma \vdash E_1 + E_2 : int} \quad \frac{\Gamma \vdash E_1 : int \quad \Gamma \vdash E_2 : int}{\Gamma \vdash E_1 - E_2 : int} \quad \frac{\Gamma \vdash E_1 : int \quad \Gamma \vdash E_2 : int}{\Gamma \vdash E_1 * E_2 : int} \quad \frac{\Gamma \vdash E_1 : int \quad \Gamma \vdash E_2 : int}{\Gamma \vdash E_1 / E_2 : int}$$

$$\frac{\Gamma \vdash E : int}{\Gamma \vdash iszero\ E : bool} \quad \frac{\Gamma \vdash E_1 : bool \quad \Gamma \vdash E_2 : t \quad \Gamma \vdash E_3 : t}{\Gamma \vdash if\, E_1\, then\ E_2\, else E_3 : t}$$

$$\frac{\Gamma \vdash E_1 : t_1 \quad [x \longmapsto t_1]\Gamma \vdash E_2 : t_2}{\Gamma \vdash let\ x = E_1\, in\ E_2 : t_2} \quad \frac{[x \longmapsto t_1]\Gamma \vdash E_1 : t_1 \quad [x \longmapsto t_1]\Gamma \vdash E_2 : t_2}{\Gamma \vdash let\ rec\ x = E_1\, in\ E_2 : t_2}$$

$$\frac{[x \longmapsto t_1]\Gamma \vdash E : t_2}{\Gamma \vdash fun\ x \rightarrow E : t_1 \rightarrow t_2} \quad \frac{\Gamma \vdash E_1 : t_1 \rightarrow t_2 \quad \Gamma \vdash E_2 : t_1}{\Gamma \vdash E_1 E_2 : t_2}$$

Generate Equations

$$
\begin{aligned}
T \rightarrow\ & int \\
\mid\ & bool \\
\mid\ & T \rightarrow T \\
\mid\ & \alpha\ (\in TVar)
\end{aligned}
$$

$$TyEqn \;\rightarrow\; \emptyset \mid T \doteq T \wedge TyEqn$$

$$\Upsilon: (Var \rightarrow T) \times E \times T \rightarrow TyEqn$$

Generate Equations

$$\frac{}{\Gamma \vdash n : int}$$

$$\Upsilon(\Gamma, n, t) = t \doteq int$$

$$\frac{}{\Gamma \vdash read : int}$$

$$\Upsilon(\Gamma, read, t) = t \doteq int$$

$$\frac{}{\Gamma \vdash x : \Gamma(x)}$$

$$\Upsilon(\Gamma, x, t) = t \doteq \Gamma(x)$$

Generate Equations

$$\frac{}{\Gamma \vdash true : bool}$$

$$\Upsilon(\Gamma, true, t) = t \doteq bool$$

$$\frac{}{\Gamma \vdash false : bool}$$

$$\Upsilon(\Gamma, false, t) = t \doteq bool$$

Generate Equations

$$\frac{\Gamma \vdash E_1 : int \quad \Gamma \vdash E_2 : int}{\Gamma \vdash E_1 + E_2 : int}$$

$$\Upsilon(\Gamma, E_1 + E_2, t) = \Upsilon(\Gamma, E_1, int) \wedge \Upsilon(\Gamma, E_2, int) \wedge t \doteq int$$

$$\frac{\Gamma \vdash E_1 : int \quad \Gamma \vdash E_2 : int}{\Gamma \vdash E_1 - E_2 : int}$$

$$\Upsilon(\Gamma, E_1 - E_2, t) = \Upsilon(\Gamma, E_1, int) \wedge \Upsilon(\Gamma, E_2, int) \wedge t \doteq int$$

Generate Equations

$$\frac{\Gamma \vdash E_1 : int \quad \Gamma \vdash E_2 : int}{\Gamma \vdash E_1 * E_2 : int}$$

$$\Upsilon(\Gamma, E_1 * E_2, t) = \Upsilon(\Gamma, E_1, int) \wedge \Upsilon(\Gamma, E_2, int) \wedge t \doteq int$$

$$\frac{\Gamma \vdash E_1 : int \quad \Gamma \vdash E_2 : int}{\Gamma \vdash E_1 / E_2 : int}$$

$$\Upsilon(\Gamma, E_1 / E_2, t) = \Upsilon(\Gamma, E_1, int) \wedge \Upsilon(\Gamma, E_2, int) \wedge t \doteq int$$

Generate Equations

$$\frac{\Gamma \vdash E : int}{\Gamma \vdash iszero\ E : bool}$$

$$\Upsilon(\Gamma, iszero\ E, t) = \Upsilon(\Gamma\ , E, int) \wedge t \doteq bool$$

$$\frac{\Gamma \vdash E_1 : bool \quad \Gamma \vdash E_2 : t \quad \Gamma \vdash E_3 : t}{\Gamma \vdash if\, E_1 then\ E_2 else E_3 : t}$$

$$\Upsilon(\Gamma, if\ \ E_1\ then\ E_2\ else\ E_3, t) = \Upsilon(\Gamma, E_1, bool) \wedge \Upsilon(\Gamma, E_2, t) \wedge \Upsilon(\Gamma, E_3, t)$$

Generate Equations

$$\frac{\Gamma \vdash E_1 : \alpha \quad [x \longmapsto \alpha]\Gamma \vdash E_2 : t}{\Gamma \vdash let\ x = E_1 in\ E_2 : t}$$

$$\Upsilon(\Gamma, let\ x = E_1\ in\ E_2, t) = \Upsilon(\Gamma, E_1, \alpha) \wedge \Upsilon([x \mapsto \alpha]\Gamma, E_2, t)\ (\text{new}\ \alpha)$$

$$\frac{[x \longmapsto \alpha]\Gamma \vdash E_1 : \alpha \quad [x \longmapsto \alpha]\Gamma \vdash E_2 : t}{\Gamma \vdash let\ rec\ x = E_1 in\ E_2 : t}$$

$$\Upsilon(\Gamma, let\ rec\ x = E_1\ in\ E_2, t) = \Upsilon([x \mapsto \alpha]\Gamma, E_1, \alpha) \wedge \Upsilon([x \mapsto \alpha]\Gamma, E_2, t)\ (\text{new}\ \alpha)$$

Generate Equations

$$\frac{[x \longmapsto \alpha_1]\Gamma \vdash E\colon \alpha_2}{\Gamma \vdash fun\ x \to E\colon \alpha_1 \to \alpha_2}$$

$$\Upsilon(\Gamma, fun\ x \to E, t) = t \doteq \alpha_1 \to \alpha_2 \wedge \Upsilon([x \longmapsto \alpha_1]\Gamma, E, \alpha_2)\ (\text{new } \alpha_1, \alpha_2)$$

$$\frac{\Gamma \vdash E_1\colon \alpha \to t \quad \Gamma \vdash E_2\colon \alpha}{\Gamma \vdash E_1 E_2\colon t}$$

$$\Upsilon(\Gamma, E_1 E_2, t) = \Upsilon(\Gamma, E_1, \alpha \to t) \wedge \Upsilon(\Gamma, E_2, \alpha)\ (\text{new } \alpha)$$

Substitution

$$S \in Subst = TVar \to T$$

$$\begin{aligned} S(int) &= int \\ S(bool) &= bool \\ S(\alpha) &= \begin{cases} t & if\ \alpha \mapsto t \in S \\ \alpha & otherwise \end{cases} \\ S(T_1 \to T_2) &= S(T_1) \to S(T_2) \end{aligned}$$

Unify

$$unify: T \times T \times Subst \to Subst$$

$$
\begin{aligned}
unify(int, int, S) &= S \\
unify(bool, bool, S) &= S \\
unify(\alpha, \alpha, S) &= S \\
unify(\alpha, t, S) &= \begin{cases} TypeError & \alpha \; occurs \; in \; t \\ extend \; S \; with \; \alpha \doteq t & otherwise \end{cases} \\
unify(t, \alpha, S) &= unify(\alpha, t, S) \\
unify(t_1 \to t_2, t_1' \to t_2', S) &= let \; S' = unify(t_1, t_1', S) in \\
&\quad\; unify(S'(t_2), S'(t_2'), S') \\
unify(\_, \_, \_) &= TypeError
\end{aligned}
$$

Unify

$$unifyall: TyEqn \rightarrow Subst \rightarrow Subst$$

$$\begin{aligned}
unifyall(\emptyset, S) &= S \\
unifyall(t_1 \doteq t_2 \wedge u, S) &= let\ S' = unify(S(t_1), S(t_2), S)\ in \\
&\quad\ unifyall(u, S')
\end{aligned}$$

Typeof

$$typeof : E \rightarrow T$$

$$typeof(E) = let\ S = unifyall(\Upsilon(\emptyset, E, \alpha), \emptyset)\ in\ \ (\text{new}\ \alpha)$$
$$S(\alpha)$$