# Well-typed programs don't get stuck

Typing judgment $\Gamma \vdash E : \tau$

$\Gamma$: context maps vars to types
e.g. $\emptyset$ $x : unit, y : unit \to unit$
we write $\Gamma, x:\tau$ for $\Gamma$ extended w/ $x:\tau$

$$\text{UNIT}\ \frac{}{\Gamma \vdash () : unit} \qquad \text{VAR}\ \frac{}{\Gamma, x:\tau \vdash x : \tau}$$

$\Gamma$: any context; $\Gamma, x:\tau$: any ctx w/ $x:\tau$

$$\text{ABS}\ \frac{\Gamma, x:\tau \vdash E : \tau'}{\Gamma \vdash \lambda x:\tau.\, E : \tau \to \tau'} \qquad \text{APP}\ \frac{\Gamma \vdash E_1 : \tau' \to \tau \quad \Gamma \vdash E_2 : \tau'}{\Gamma \vdash E_1\ E_2 : \tau}$$

$$\text{APP}\ \frac{\text{ABS}\ \dfrac{\text{VAR}\ \dfrac{}{x:unit \vdash x : unit}}{\emptyset \vdash \lambda x:unit.\, x : unit \to unit} \qquad \text{UNIT}\ \dfrac{}{\emptyset \vdash () : unit}}{\emptyset \vdash (\lambda x:unit.\, x)\ () : unit}$$

Type Safety:
If $\emptyset \vdash E : \tau$ and $E \to^* E'$ (0 or more steps) then either $E'$ is a value or there exists $E''$ such that $E' \to E''$

2 parts:
"preservation": If $\emptyset \vdash E : \tau$ and $E \to^* E'$ then $\emptyset \vdash E' : \tau$
"progress": If $\emptyset \vdash E : \tau$ then either $E$ is a value or there exists $E'$ such that $E \to E'$.

$$\emptyset \vdash E : \tau \overset{\text{prog}}{\Rightarrow} E \to E' \overset{\text{pres}}{\Rightarrow} \emptyset \vdash E' : \tau \overset{\text{prog}}{\Rightarrow} E' \to E'' \overset{\text{pres}}{\Rightarrow} \emptyset \vdash E'' : \tau \Rightarrow \dots$$

Also: If $\emptyset \vdash E : \tau$ and $E \to^* E'$, then $\emptyset \vdash E' : \tau$

Does type safety guarantee that all programs that don't get stuck are well-typed? No!

e.g. $(\lambda x{:}unit.\, x)\ (\lambda x{:}unit.\, x) \to \lambda x{:}unit.\, x$

How about our self-application from last time?
$\lambda f.\ f\ f$
Not well-typed (for the same reason it's not well-typed in OCaml)

Does that mean we can't define an infinite loop / Y combinator?
Unfortunately, yes.
Theorem: If $\emptyset \vdash E : \tau$ then there exists a value $E'$ such that $E \to^* E'$
(not true in all statically typed languages, of course!
we would need to add features to allow recursion)

## Pairs

OCaml: *.

$\tau ::= \ldots \mid \tau \times \tau$
$E ::= \ldots \mid (E, E) \mid \text{fst}\ E \mid \text{snd}\ E$

$$\frac{E_1 \to E_1'}{(E_1, E_2) \to (E_1', E_2)} \qquad \frac{E_1 \text{ value} \quad E_2 \to E_2'}{(E_1, E_2) \to (E_1, E_2')}$$

$$\frac{E \to E'}{\text{fst}\ E \to \text{fst}\ E'} \qquad \frac{E_1, E_2 \text{ values}}{\text{fst}\ (E_1, E_2) \to E_1}$$
(snd) (snd) (snd) (2)

$$\frac{\Gamma \vdash E_1 : \tau_1 \quad \Gamma \vdash E_2 : \tau_2}{\Gamma \vdash (E_1, E_2) : \tau_1 \times \tau_2} \qquad \frac{\Gamma \vdash E : \tau_1 \times \tau_2}{\Gamma \vdash \text{fst}\ E : \tau_1} \qquad \frac{\cdot\ \Gamma \vdash E : \tau_1 \times \tau_2}{\Gamma \vdash \text{snd}\ E : \tau_2}$$