* Some Clarifications.

1. $$\frac{\Gamma \vdash M : \bot}{\Gamma \vdash \text{abort}_A\, M : A}$$

There is no evaluation rule for $\text{abort}_A M$. and it is not a value.

≡ ~~So there is some terms~~

2. Termination/Strong Normalization

If $\vdash M : A$ then $\exists V$ s.t. $M \rightsquigarrow^* V$.

* Supporting recursion (Leaving Curry-Howard)

$M ::= \cdots \mid 0 \mid S(M) \mid Y(M) \mid \text{pred}(M) \mid \text{iszero}(M)$

$V ::= \cdots \mid 0 \mid S(V) \mid \text{True} \mid \text{False}$

$A ::= \cdots \mid \text{Nat} \mid \text{Bool}$.

Typing rules: $\dfrac{\Gamma \vdash M : \text{Nat}}{\Gamma \vdash S(M) : \text{Nat}}$ $\qquad \dfrac{}{\Gamma \vdash 0 : \text{Nat}}$ $\qquad \dfrac{\Gamma \vdash M : \text{Nat}}{\Gamma \vdash \text{pred}(M) : \text{Nat}}$

$$\frac{\Gamma \vdash M : A \to A}{\Gamma \vdash Y(M) : A} \qquad \frac{\Gamma \vdash M : \text{Nat}}{\Gamma \vdash \text{iszero}(M) : \text{Bool}}$$

Evaluation:

$$\frac{}{\text{pred } 0 \rightsquigarrow 0} \qquad \frac{M \rightsquigarrow M'}{Y(M) \rightsquigarrow Y(M')}$$

$$\frac{M \rightsquigarrow M'}{\text{pred}(M) \rightsquigarrow \text{pred}(M')} \qquad \frac{}{Y(\lambda x.M) \rightsquigarrow [Y(\lambda x.M)/x]M}$$

$$\frac{}{\text{pred}(S(V)) \rightsquigarrow V} \qquad \frac{}{\text{iszero}(0) \rightsquigarrow \text{True}}$$

$$\frac{M \rightsquigarrow M'}{S(M) \rightsquigarrow S(M')} \qquad \frac{}{\text{iszero}(S(V)) \rightsquigarrow \text{False}}$$

$$\frac{M \rightsquigarrow M'}{\text{iszero}(M) \rightsquigarrow \text{iszero}(M')}$$

Remarks:

* STLC + Fixpoint + Natural numbers is called 'PCF' (Programming Computable Function)

* We can program alot of algorithms within PCF.

E.g.

```
add : Nat → Nat → Nat
add x y = if iszero x then y
          else S(add (pred x) y)
```

In Haskell syntax

```
add :: Nat → Nat → Nat
add 0 n = n
add (S m) n = S(add m n)
```

In PCF.

$$add = Y(\lambda f.\lambda x.\lambda y.\ \text{if iszero } x \text{ then } y \text{ else } S(f(\text{pred } x)\ y))$$

has type $(Nat \to Nat \to Nat) \to (Nat \to Nat \to Nat)$

e.g. add 1 1

$= add\ (S(0))\ (S(0))$

$= Y(\lambda f.\lambda x.\lambda y.\ \ldots.)\ (S(0))\ (S(0))$

$\rightsquigarrow (\lambda x.\lambda y.\ \text{if iszero } x \text{ then } y \text{ else } S(add\ (pred\ x)\ y))\ (S(0))\ (S(0))$

$\rightsquigarrow$ if iszero$(S(0))$ then $S(0)$ else $S(add\ (pred\ (S(0)))\ (S(0)))$

$\rightsquigarrow$ if False then .. else ..

$\rightsquigarrow S(add\ (pred\ (S(0)))\ (S(0)))$

$\rightsquigarrow S(add\ 0\ (S(0)))$

$\overset{*}{\rightsquigarrow} S(S(0)) = 2.$

Remarks.

1. PCF does not have termination property.

$\vdash Y(\lambda x.x) : A$

$Y(\lambda x.x) \rightsquigarrow Y(\lambda x.x) \rightsquigarrow \cdots$

2. We can continue to use Y to define alot of functions.

e.g. iseven : Nat → Bool.

iseven $= Y(\lambda f.\lambda x.$ if iszero(x) then True
else if iszero(pred(x)) then False
else f (pred(pred(x))))

3. PCF is also type preserving, i.e.

If $\Gamma \vdash M : A$ and $M \rightsquigarrow M'$, then $\Gamma \vdash M' : Nat$.

So we know the values of Nat can only be $\{0, 1, \cdots\}$ and nothing else.