

## Types (lax logic)

$\langle \text{assign} \mid tp \mid \langle \text{macro} \mid \text{type} \rangle \rangle$   
 $\langle \text{assign} \mid \text{bottom} \mid \langle \text{macro} \mid \text{void} \rangle \rangle$   
 $\langle \text{assign} \mid \text{bottom}\#0 \mid \langle \text{macro} \mid \perp \rangle \rangle$   
 $\langle \text{assign} \mid \text{unit} \mid \langle \text{macro} \mid \text{unit} \rangle \rangle$   
 $\langle \text{assign} \mid \text{unit}\#0 \mid \langle \text{macro} \mid \top \rangle \rangle$   
 $\langle \text{assign} \mid \text{arrow} \mid \langle \text{macro} \mid tp1 \mid tp2 \mid tp1 \Rightarrow tp2 \rangle \rangle$   
 $\langle \text{assign} \mid \text{product} \mid \langle \text{macro} \mid tp1 \mid tp2 \mid tp1 \wedge tp2 \rangle \rangle$   
 $\langle \text{assign} \mid \text{disjunction} \mid \langle \text{macro} \mid tp1 \mid tp2 \mid tp1 \vee tp2 \rangle \rangle$   
 $\langle \text{assign} \mid \text{for\_all} \mid \langle \text{macro} \mid x \mid u \mid \forall x. u \rangle \rangle$   
 $\langle \text{assign} \mid \text{monad} \mid \langle \text{macro} \mid f \mid \diamond f \rangle \rangle$

## Terms (system F)

$\langle \text{assign} \mid tm \mid \langle \text{macro} \mid \text{term} \rangle \rangle$   
 $\langle \text{assign} \mid \text{single} \mid \langle \text{macro} \mid \langle \rangle \rangle \rangle$   
 $\langle \text{assign} \mid \text{inl} \mid \langle \text{macro} \mid f \mid \mathbf{inl} f \rangle \rangle$   
 $\langle \text{assign} \mid \text{inr} \mid \langle \text{macro} \mid f \mid \mathbf{inr} f \rangle \rangle$   
 $\langle \text{assign} \mid \text{cases} \mid \langle \text{macro} \mid t \mid x \mid u1 \mid y \mid u2 \mid \text{case } t \text{ of } (\mathbf{inl} x) \mapsto u1 \mid (\mathbf{inr} y) \mapsto u2 \rangle \rangle$   
 $\langle \text{assign} \mid \text{lam} \mid \langle \text{macro} \mid x \mid \tau \mid u \mid \lambda x: \tau. u \rangle \rangle$   
 $\langle \text{assign} \mid \text{lam2} \mid \langle \text{macro} \mid \tau \mid u \mid \Lambda \tau. u \rangle \rangle$   
 $\langle \text{assign} \mid \text{pair} \mid \langle \text{macro} \mid x \mid y \mid \langle x, y \rangle \rangle \rangle$   
 $\langle \text{assign} \mid \text{let} \mid \langle \text{macro} \mid x \mid y \mid t \mid u \mid \text{let } \langle x, y \rangle = t \text{ in } u \rangle \rangle$   
 $\langle \text{assign} \mid \text{letv} \mid \langle \text{macro} \mid x \mid \tau \mid t \mid u \mid \text{let } x: \tau = t \text{ in } u \rangle \rangle$   
 $\langle \text{assign} \mid \text{app} \mid \langle \text{macro} \mid t \mid u \mid t u \rangle \rangle$   
 $\langle \text{assign} \mid \text{inst} \mid \langle \text{macro} \mid t \mid u \mid t \{u\} \rangle \rangle$

## Polymorphic monadic constants

$\langle \text{assign} \mid \text{return} \mid \langle \text{macro} \mid \text{unit} \rangle \rangle$   
 $\langle \text{assign} \mid \text{bind} \mid \langle \text{macro} \mid \text{bind} \rangle \rangle$

## Monadic constants (schemas)

$\langle \text{assign} \mid \text{mreturn} \mid \langle \text{macro} \mid a \mid \text{unit}_a \rangle \rangle$   
 $\langle \text{assign} \mid \text{mbind} \mid \langle \text{macro} \mid a \mid b \mid \text{bind}_{a,b} \rangle \rangle$

## Modal continuation monad

`<assign|kmonad|<macro|f| $\nabla$ f>>`  
`<assign|kreturn|<macro|unit $\nabla$ >>`  
`<assign|kbind|<macro|bind $\nabla$ >>`

#### Monadic reflection

`<assign|reflect|<macro|f| $\mu$ (f)>>`  
`<assign|reify|<macro|f|[f]>>`

#### Delimited control operators

`<assign|mshift|<macro|shift>>`  
`<assign|mreset|<macro|reset>>`

#### Polymorphic continuation monad

`<assign|cpsshift|<macro|shift>>`  
`<assign|cpsreset|<macro|reset>>`

#### Typing

`<assign|typing|<macro|term|tau| $\vdash$  term : tau>>`  
`<assign|of_letv|<macro|stp|oft1|oft2|stp  $\wedge$  oft1  $\implies$  oft2>>`  
`<assign|of_reify|<macro|oft|oft2|oft  $\implies$  oft2>>`  
`<assign|of_reflect|<macro|oft|oft2|oft  $\implies$  oft2>>`  
`<assign|of_cpsshift|<macro|oft|oft>>`  
`<assign|of_cpsreset|<macro|oft|oft>>`

#### The continuation monad

`<assign|answer|<macro|o>>`  
`<assign|escape|<macro|escape>>`

#### The state monad

`<assign|state|<macro| $\sigma$ >>`  
`<assign|mset|<macro|store>>`  
`<assign|mget|<macro|fetch>>`

#### The exception monad

`<assign|exn|<macro| $\varepsilon$ >>`  
`<assign|raise|<macro|raise>>`  
`<assign|try|<macro|handle>>`  
`<assign|handle|<macro|f|e|g|f handle e  $\mapsto$  g>>`

#### Display tiny derivation trees

`<assign|micro|<macro|f| $\iota$ >>`