termvar, x, f	term variable
typevar, α, β	type variable
capvar, δ	capability variable
integer, i	integer
s	string
b	boolean
j, k	index

const, *K*	::=		constants
	|	**noop**	no-op
	|	**wait**	wait
	|	**selfdestruct**	self destruct
	|	**move**	move
	|	**turn**	turn
	|	**grab**	grab
	|	**place**	place
	|	**give**	give
	|	**install**	install
	|	**make**	make
	|	**has**	inventory test
	|	**count**	inventory count
	|	**drill**	drill
	|	**build**	build
	|	**salvage**	salvage
	|	**reprogram**	reprogram
	|	**say**	say
	|	**log**	log
	|	**view**	view
	|	**appear**	appear
	|	**create**	create
	|	**whereami**	location query
	|	**blocked**	front query
	|	**scan**	scan
	|	**upload**	upload
	|	**ishere**	cell query
	|	**whoami**	self name query
	|	**random**	random
	|	**run**	run
	|	**if**	if
	|	**inl**	left injection
	|	**inr**	right injection
	|	**case**	case analysis
	|	**pair**	pair formation
	|	**fst**	first projection
	|	**snd**	second projection
	|	**force**	force
	|	**return**	return
	|	**try**	try
	|	**raise**	raise exception
	|	**not**	not
	|	**neg**	negate
	|	**eq**	equal to
	|	**neq**	not equal to
	|	**lt**	less than
	|	**gt**	greater than
	|	**leq**	less-or-equal
	|	**geq**	greater-or-equal
	|	**add**	addition

	| **sub**		subtraction
	| **mul**		multiplication
	| **div**		division
	| **exp**		exponentiation

d	::=		direction
	| **left**		
	| **right**		
	| **back**		
	| **forward**		
	| **north**		
	| **south**		
	| **east**		
	| **west**		
	| **down**		

term, t	::=		term
	| $()$		unit value
	| K		constant
	| d		direction literal
	| i		integer literal
	| s		string literal
	| b		boolean literal
	| x		variable
	| $\lambda x.t$	bind x in t	function abstraction
	| $t_1\ t_2$		function application
	| $\mathbf{let}\ x = t_1\ \mathbf{in}\ t_2$	bind x in t_2	let
	| $\mathbf{def}\ x = t\ \mathbf{end}$		definition
	| $x \leftarrow t_1 \fatsemi t_2$	bind x in t_2	bind
	| $\{t\}$		delay
	| (t)	S	

basety, B	::=		base type
	| $()$		unit
	| **int**		integers
	| **string**		strings
	| **dir**		directions
	| **bool**		booleans

type, τ	::=		monotype
	| α		variable
	| B		base type
	| $\mathbf{cmd}^{\Delta}\ \tau$		command type
	| $\{\tau\}^{\Delta}$		delay type
	| $\tau_1 + \tau_2$		sum type
	| $\tau_1 \times \tau_2$		product type
	| $\tau_1 \xrightarrow{\Delta} \tau_2$		function type
	| $\tau_1 \rightarrow \tau_2$		function type with empty capability set
	| (τ)	S	

polytype, σ	::=		polytype
	|	τ	monotype
	|	$\forall\, \alpha_1 \,..\, \alpha_j\, \delta_1 \,..\, \delta_k . \tau$	quantified type

capability, κ	::=		capability
	|	GOD	
	|	SELFDESTRUCT	
	|	MOVE	
	|	TURN	
	|	GRAB	
	|	PLACE	
	|	GIVE	
	|	INSTALL	
	|	MAKE	
	|	COUNT	
	|	BUILD	
	|	SALVAGE	
	|	DRILL	
	|	SENSELOC	
	|	SENSEFRONT	
	|	SENSEHERE	
	|	SCAN	
	|	RANDOM	
	|	APPEAR	
	|	CREATE	
	|	LOG	
	|	FLOAT	
	|	COND	
	|	COMPARE	
	|	ARITH	
	|	ENV	
	|	LAMBDA	
	|	RECURSION	
	|	REPROGRAM	
	|	WHOAMI	

Δ	::=			capability set
	|	$\bullet$		empty
	|	δ		variable
	|	κ		singleton
	|	Δ, κ		snoc
	|	(Δ)	S	

Γ	::=		type context
	|	$\bullet$	empty context
	|	$\Gamma, x : \tau$	snoc

$\boxed{\vdash_c K : \sigma}$ Constant K has type σ

$$\frac{}{\vdash_c \mathbf{noop} : \mathbf{cmd}^{\bullet}\ ()}\ \text{CTYPE\_NOOP}$$

$$\frac{}{\vdash_c \mathbf{wait} : \mathbf{int} \to \mathbf{cmd}^{\bullet}\ ()}\ \text{CTYPE\_WAIT}$$

$$\frac{}{\vdash_c \mathbf{selfdestruct} : \mathbf{cmd}^{\mathsf{SELFDESTRUCT}}\ ()}\ \text{CTYPE\_SELFDESTRUCT}$$

$$\frac{}{\vdash_c \mathbf{move} : \mathbf{cmd}^{\mathsf{MOVE}}\ ()}\ \text{CTYPE\_MOVE}$$

$$\frac{}{\vdash_c \mathbf{turn} : \mathbf{dir} \to \mathbf{cmd}^{\mathsf{TURN}}\ ()}\ \text{CTYPE\_TURN}$$

$$\frac{}{\vdash_c \mathbf{grab} : \mathbf{cmd}^{\mathsf{GRAB}}\ \mathbf{string}}\ \text{CTYPE\_GRAB}$$

$$\frac{}{\vdash_c \mathbf{place} : \mathbf{string} \to \mathbf{cmd}^{\mathsf{PLACE}}\ ()}\ \text{CTYPE\_PLACE}$$

$$\frac{}{\vdash_c \mathbf{give} : \mathbf{string} \to \mathbf{string} \to \mathbf{cmd}^{\mathsf{GIVE}}\ ()}\ \text{CTYPE\_GIVE}$$

$$\frac{}{\vdash_c \mathbf{install} : \mathbf{string} \to \mathbf{string} \to \mathbf{cmd}^{\mathsf{INSTALL}}\ ()}\ \text{CTYPE\_INSTALL}$$

$$\frac{}{\vdash_c \mathbf{make} : \mathbf{string} \to \mathbf{cmd}^{\mathsf{MAKE}}\ ()}\ \text{CTYPE\_MAKE}$$

$$\frac{}{\vdash_c \mathbf{has} : \mathbf{string} \to \mathbf{cmd}^{\bullet}\ \mathbf{bool}}\ \text{CTYPE\_HAS}$$

$$\frac{}{\vdash_c \mathbf{count} : \mathbf{string} \to \mathbf{cmd}^{\mathsf{COUNT}}\ \mathbf{int}}\ \text{CTYPE\_COUNT}$$

$$\frac{}{\vdash_c \mathbf{drill} : \mathbf{dir} \to \mathbf{cmd}^{\mathsf{DRILL}}\ ()}\ \text{CTYPE\_DRILL}$$

$$\frac{}{\vdash_c \mathbf{build} : \forall\, \alpha\, \delta_1\, \delta_2.\mathbf{string} \to \{\mathbf{cmd}^{\delta_1}\ \alpha\}^{\delta_2} \to \mathbf{cmd}^{\mathsf{BUILD}}\ \mathbf{string}}\ \text{CTYPE\_BUILD}$$

$$\frac{}{\vdash_c \mathbf{salvage} : \mathbf{cmd}^{\mathsf{SALVAGE}}\ ()}\ \text{CTYPE\_SALVAGE}$$

$$\frac{}{\vdash_c \mathbf{reprogram} : \forall\, \alpha\, \delta_1\, \delta_2.\mathbf{string} \to \{\mathbf{cmd}^{\delta_1}\ \alpha\}^{\delta_2} \to \mathbf{cmd}^{\mathsf{BUILD}}\ ()}\ \text{CTYPE\_REPROGRAM}$$

$$\frac{}{\vdash_c \mathbf{say} : \mathbf{string} \to \mathbf{cmd}^{\bullet}\ ()}\ \text{CTYPE\_SAY}$$

$$\frac{}{\vdash_c \mathbf{log} : \mathbf{string} \to \mathbf{cmd}^{\mathsf{LOG}}\ ()}\ \text{CTYPE\_LOG}$$

$$\frac{}{\vdash_c \mathbf{view} : \mathbf{string} \to \mathbf{cmd}^{\bullet}\ ()}\ \text{CTYPE\_VIEW}$$

$$\frac{}{\vdash_c \mathbf{appear} : \mathbf{string} \to \mathbf{cmd}^{\mathsf{APPEAR}}\ ()}\ \text{CTYPE\_APPEAR}$$

$$\frac{}{\vdash_c \mathbf{create} : \mathbf{string} \to \mathbf{cmd}^{\mathsf{GOD}}\ ()}\ \text{CTYPE\_CREATE}$$

$$\frac{}{\vdash_c \mathbf{blocked} : \mathbf{cmd}^{\mathsf{SENSEFRONT}}\ \mathbf{bool}}\ \text{CTYPE\_BLOCKED}$$

$$\frac{}{\vdash_c \mathbf{scan} : \mathbf{dir} \to \mathbf{cmd}^{\mathsf{SCAN}}\ (() + \mathbf{string})}\ \text{CTYPE\_SCAN}$$

$$\frac{}{\vdash_c \mathbf{upload} : \mathbf{string} \to \mathbf{cmd}^{\mathsf{SCAN}}\ ()}\ \text{CTYPE\_UPLOAD}$$

$$\frac{}{\vdash_c \mathbf{ishere} : \mathbf{string} \to \mathbf{cmd}^{\mathsf{SENSEHERE}}\ \mathbf{bool}}\ \text{CTYPE\_ISHERE}$$

$$\frac{}{\vdash_c \mathbf{whereami} : \mathbf{cmd}^{\mathsf{SENSELOC}}\ (\mathbf{int} \times \mathbf{int})}\ \text{CTYPE\_WHEREAMI}$$

$$\frac{}{\vdash_c \mathbf{whoami} : \mathbf{cmd}^{\mathsf{WHOAMI}}\ \mathbf{string}}\ \text{CTYPE\_WHOAMI}$$

$$\frac{}{\vdash_c \mathbf{random} : \mathbf{int} \to \mathbf{cmd}^{\mathsf{RANDOM}}\ \mathbf{int}}\ \text{CTYPE\_RANDOM}$$

$$\frac{}{\vdash_c \mathbf{run} : \mathbf{string} \to \mathbf{cmd}^{\bullet}\ ()}\ \text{CTYPE\_RUN}$$

$$\frac{}{\vdash_c \mathbf{if} : \forall\, \alpha\, \delta.\mathbf{bool} \to \{\alpha\}^{\delta} \to \{\alpha\}^{\delta} \xrightarrow{\delta} \alpha}\ \text{CTYPE\_IF}$$

$$\frac{}{\vdash_c \mathbf{inl} : \forall\, \alpha\, \beta\, .\alpha \to \alpha + \beta}\ \text{CTYPE\_INL}$$

$$\frac{}{\vdash_c \mathbf{inr} : \forall\, \alpha\, \beta\, .\beta \to \alpha + \beta}\ \text{CTYPE\_INR}$$

$$\frac{}{\vdash_c \mathbf{case} : \forall\, \alpha_1\, \alpha_2\, \beta\, \delta.(\alpha_1 \xrightarrow{\delta} \beta) \to (\alpha_2 \xrightarrow{\delta} \beta) \to ((\alpha_1 + \alpha_2) \xrightarrow{\delta} \beta)}\ \text{CTYPE\_CASE}$$

$$\frac{}{\vdash_c \mathbf{pair} : \forall\, \alpha_1\, \alpha_2\, .\alpha_1 \to \alpha_2 \to \alpha_1 \times \alpha_2}\ \text{CTYPE\_PAIR}$$

$$\frac{}{\vdash_c \mathbf{fst} : \forall\, \alpha_1\, \alpha_2\, .\alpha_1 \times \alpha_2 \to \alpha_1}\ \text{CTYPE\_FST}$$

$$\frac{}{\vdash_c \mathbf{snd} : \forall\, \alpha_1\, \alpha_2\, .\alpha_1 \times \alpha_2 \to \alpha_2}\ \text{CTYPE\_SND}$$

$$\frac{}{\vdash_c \mathbf{return} : \forall\, \alpha\, .\alpha \to \mathbf{cmd}^{\bullet}\ \alpha}\ \text{CTYPE\_RETURN}$$

$$\frac{}{\vdash_c \mathbf{try} : \forall\, \alpha\, \delta_1\, \delta_2\, \delta_3\, \delta_4.\{\mathbf{cmd}^{\delta_1}\ \alpha\}^{\delta_2} \to \{\mathbf{cmd}^{\delta_3}\ \alpha\}^{\delta_4} \to \mathbf{cmd}^{(\delta_1 \cup \delta_2 \cup \delta_3 \cup \delta_4)}\ \alpha}\ \text{CTYPE\_TRY}$$

$$\frac{}{\vdash_c \mathbf{raise} : \forall\, \alpha\, .\mathbf{string} \to \mathbf{cmd}^{\bullet}\ \alpha}\ \text{CTYPE\_RAISE}$$

$$\frac{}{\vdash_c \mathbf{not} : \mathbf{bool} \to \mathbf{bool}}\ \text{CTYPE\_NOT}$$

$$\frac{}{\vdash_c \mathbf{eq} : \forall\, \alpha\, .\alpha \to \alpha \xrightarrow{\mathsf{COMPARE}} \mathbf{bool}}\ \text{CTYPE\_EQ}$$

$$\frac{}{\vdash_c \mathbf{neq} : \forall\, \alpha\, .\alpha \to \alpha \xrightarrow{\mathsf{COMPARE}} \mathbf{bool}}\ \text{CTYPE\_NEQ}$$

$$\frac{}{\vdash_c \mathbf{lt} : \forall\, \alpha\, .\alpha \to \alpha \xrightarrow{\mathsf{COMPARE}} \mathbf{bool}}\ \text{CTYPE\_LT}$$

$$\frac{}{\vdash_c \mathbf{gt} : \forall\, \alpha\, .\alpha \to \alpha \xrightarrow{\mathsf{COMPARE}} \mathbf{bool}}\ \text{CTYPE\_GT}$$

$$\frac{}{\vdash_c \mathbf{leq} : \forall\, \alpha\, .\alpha \to \alpha \xrightarrow{\mathsf{COMPARE}} \mathbf{bool}}\ \text{CTYPE\_LEQ}$$

$$\frac{}{\vdash_c \mathbf{geq} : \forall\, \alpha\, .\alpha \to \alpha \xrightarrow{\mathsf{COMPARE}} \mathbf{bool}}\ \text{CTYPE\_GEQ}$$

$$\frac{}{\vdash_c \mathbf{neg} : \mathbf{int} \xrightarrow{\mathsf{ARITH}} \mathbf{int}}\ \text{CTYPE\_NEG}$$

$$\frac{}{\vdash_c \mathbf{add} : \mathbf{int} \to \mathbf{int} \xrightarrow{\mathsf{ARITH}} \mathbf{int}}\ \text{CTYPE\_ADD}$$

$$\frac{}{\vdash_c \mathbf{sub} : \mathbf{int} \to \mathbf{int} \xrightarrow{\mathsf{ARITH}} \mathbf{int}}\ \text{CTYPE\_SUB}$$

$$\frac{}{\vdash_c \mathbf{mul} : \mathbf{int} \to \mathbf{int} \xrightarrow{\mathsf{ARITH}} \mathbf{int}}\ \text{CTYPE\_MUL}$$

$$\frac{}{\vdash_c \mathbf{div} : \mathbf{int} \to \mathbf{int} \xrightarrow{\mathsf{ARITH}} \mathbf{int}}\ \text{CTYPE\_DIV}$$

$$\frac{}{\vdash_c \mathbf{exp} : \mathbf{int} \to \mathbf{int} \xrightarrow{\mathsf{ARITH}} \mathbf{int}}\ \text{CTYPE\_EXP}$$

$\boxed{\Gamma \vdash t : \sigma; \Delta}$ In context Γ, t has type σ, and its evaluation requires capabilites Δ

$$\frac{}{\Gamma \vdash () : (); \bullet}\ \text{TYPE\_UNIT}$$

$$\frac{}{\Gamma \vdash d : \mathbf{dir}; \bullet}\ \text{TYPE\_DIR}$$

$$\frac{}{\Gamma \vdash i : \mathbf{int}; \bullet}\ \text{TYPE\_INT}$$

$$\frac{}{\Gamma \vdash s : \mathbf{string}; \bullet}\ \text{TYPE\_STRING}$$

$$\frac{}{\Gamma \vdash b : \mathbf{bool}; \bullet}\ \text{TYPE\_BOOL}$$

$$\frac{\vdash_c K : \tau}{\Gamma \vdash K : \tau; \bullet}\ \text{TYPE\_CONST}$$

$$\frac{x : \tau \in \Gamma}{\Gamma \vdash x : \tau; \bullet}\ \text{TYPE\_VAR}$$

$$\frac{\Gamma, x : \tau_1 \vdash t : \tau_2; \Delta}{\Gamma \vdash \lambda x.t : \tau_1 \xrightarrow{\Delta} \tau_2; \bullet}\ \text{TYPE\_LAM}$$

$$\frac{\Gamma \vdash t_1 : \tau_1 \xrightarrow{\Delta_1} \tau_2; \Delta_2 \quad \Gamma \vdash t_2 : \tau_1; \Delta_3}{\Gamma \vdash t_1\ t_2 : \tau_2; \Delta_1 \cup \Delta_2 \cup \Delta_3}\ \text{TYPE\_APP}$$

$$\frac{\Gamma \vdash t_1 : \tau_1; \Delta_1 \quad \Gamma, x : \tau_1 \vdash t_2 : \tau_2; \Delta_2}{\Gamma \vdash \mathbf{let}\, x = t_1\, \mathbf{in}\, t_2 : \tau_2; \Delta_1 \cup \Delta_2}\ \text{TYPE\_LET}$$

$$\frac{\Gamma \vdash t : \tau; \Delta}{\Gamma \vdash \{t\} : \{\tau\}^{\Delta}; \bullet}\ \text{TYPE\_DELAY}$$

$$\frac{\Gamma \vdash t : \{\tau\}^{\Delta_1}; \Delta_2}{\Gamma \vdash \mathbf{force}\, t : \tau; \Delta_1 \cup \Delta_2}\ \text{TYPE\_FORCE}$$

$$\frac{\Gamma \vdash t_1 : \mathbf{cmd}^{\Delta_{11}}\ \tau_1; \Delta_{12} \quad \Gamma, x : \tau_1 \vdash t_2 : \mathbf{cmd}^{\Delta_{21}}\ \tau_2; \Delta_{22}}{\Gamma \vdash x \leftarrow t_1 \fatsemi t_2 : \mathbf{cmd}^{(\Delta_{11} \cup \Delta_{12} \cup \Delta_{21} \cup \Delta_{22})}\ \tau_2; \bullet}\ \text{TYPE\_BIND}$$

```
Definition rules:         65 good    0 bad
Definition rule clauses: 73 good    0 bad
```