id	::=			
v	::=			Values
	|	**b**		Boolean
	|	**n**		Number
	|	**str**		String
	|	**undefined**		Undefined
	|	**null**		null
	|	$get(s, id)$	M	GetStore
s	::=			Store
	|	$put(s, id, v)$	M	PutStore
vd	::=			VariableDeclaration
	|	id		Declaration
	|	$id = e$		Definition
	|	vd, vd'		Multiple
e	::=			Expression
	|	v		Value
	|	id		Deref
	|	$id = e$		Ref
m	::=			Statement
	|	e		Expression
	|	ϵ		Skip
	|	$m; m'$		Seq
	|	**var** vd		VarDeclaration
Tv	::=			Value Type
	|	**number**		Number
	|	**boolean**		Boolean
	|	**string**		String
	|	**undefined**		undefined
	|	**null**		null
T	::=			Expression Type
	|	Tv		ValueType
	|	$ref\langle Tv \rangle$		Location Type
X	::=			
	|	S		
C	::=			Constraints
	|	S		
S	::=			Set

	$\varnothing$	EmptySet
	$\{T\}$	Singleton
	$X \cup X'$	Union
	$X \cap X'$	Intersect

terminals ::=

	$\rightarrow$
	Γ
	$\vdash$
	;
	var
	$\varnothing$

formula ::=

	judgement

Jop ::=

	$< m, s > \rightarrow < m', s' >$	
	$\Gamma\,(id) = T$	M
	$\Gamma \vdash e : T \mid_X C$	
	$\Gamma \vdash m \mid_X C$	

judgement ::=

	Jop

user_syntax ::=

	id
	v
	s
	vd
	e
	m
	Tv
	T
	X
	C
	S
	terminals

$\boxed{< m, s > \rightarrow < m', s' >}$

$$\frac{}{< v; e, s > \rightarrow < e, s >}\ \text{SEQ1}$$

$$\frac{< e_1, s > \rightarrow < e_1', s' >}{< e_1; e_2, s > \rightarrow < e_1'; e_2, s' >}\ \text{SEQ2}$$

$$\frac{< e, s > \rightarrow < e', s' >}{< id = e, s > \rightarrow < id = e', s' >}\ \text{ASSIGN1}$$

$$\frac{}{< id = v, s > \rightarrow < v, put(s, id, v) >}\ \text{ASSIGN2}$$

$$\frac{}{< id, s > \rightarrow < get(s, id), s >}\ \text{DEREF}$$

$$\frac{}{< \textbf{var}\ id = e, s > \rightarrow < id = e, s >}\ \text{VAR1}$$

$$\frac{}{< \textbf{var}\ id, s > \rightarrow < \textbf{undefined}, s >}\ \text{VAR2}$$

$$\frac{}{< \textbf{var}\ vd, vd', s > \rightarrow < \textbf{var}\ vd; \textbf{var}\ vd', s >}\ \text{VAR3}$$

$\boxed{\Gamma\,(id) = T}$

$\boxed{\Gamma \vdash e : T \ |_X \ C}$

$$\frac{}{\Gamma \vdash \textbf{n} : \textbf{number} \ |_\varnothing \varnothing}\ \text{V\_NUM}$$

$$\frac{}{\Gamma \vdash \textbf{b} : \textbf{boolean} \ |_\varnothing \varnothing}\ \text{V\_BOOL}$$

$$\frac{}{\Gamma \vdash \textbf{str} : \textbf{string} \ |_\varnothing \varnothing}\ \text{V\_STRING}$$

$$\frac{}{\Gamma \vdash \textbf{undefined} : \textbf{undefined} \ |_\varnothing \varnothing}\ \text{V\_UNDEFINED}$$

$$\frac{}{\Gamma \vdash \textbf{null} : \textbf{null} \ |_\varnothing \varnothing}\ \text{V\_NULL}$$

$$\frac{\Gamma\,(id) = T}{\Gamma \vdash id : T \ |_{\{T\}} \varnothing}\ \textsc{IdType}$$

$\Gamma \vdash id : T_1 \ |_{X_1} \ C_1$
$\Gamma \vdash e : T_2 \ |_{X_2} \ C_2$
<<no parses (char 2): !*** X1 n X2 = X1 n X(T2) = X2 n X(T1) = {} >>
<<no parses (char 2): !*** X' = X1 u X2 >>
<<no parses (char 2): !*** C' = C1 u C2 u {T1 = ref<T2>} >>

$$\frac{}{\Gamma \vdash id = e : T_2 \ |_{X'} \ C'}\ \textsc{AssignType}$$

$\boxed{\Gamma \vdash m \ |_X \ C}$

$$\frac{}{\Gamma \vdash \epsilon \ |_\varnothing \varnothing}\ \textsc{SkipTypable}$$

$$\frac{}{\Gamma \vdash \textbf{var}\ id \ |_\varnothing \varnothing}\ \textsc{DecTypable}$$

$$\frac{\Gamma \vdash e : T \ |_X \ C}{\Gamma \vdash e \ |_X \ C}\ \textsc{ExpTypable}$$

$\Gamma \vdash m_1 \ |_{X_1} \ C_1$
$\Gamma \vdash m_2 \ |_{X_2} \ C_2$
<<no parses (char 2): !***X1 n X2 = {} >>
<<no parses (char 2): !***C' = C1 u C2 >>
<<no parses (char 2): !***X' = X1 u X2 >>

$$\frac{}{\Gamma \vdash m_1; m_2 \ |_{X'} \ C'}\ \textsc{SeqTypable}$$

$\Gamma \vdash \textbf{var}\ id \ |_{X_1} \ C$
$\Gamma \vdash id = e : T \ |_{X_2} \ C$
<<no parses (char 2): !***X1 n X2 = {} >>
<<no parses (char 2): !***C' = C1 u C2 >>
<<no parses (char 2): !***X' = X1 u X2 >>

$$\frac{}{\Gamma \vdash \textbf{var}\ id = e \ |_{X'} \ C'}\ \textsc{DefTypable}$$

$$\frac{\Gamma \vdash \mathbf{var}\ vd \ |_{X_1}\ C_1 \quad \Gamma \vdash \mathbf{var}\ vd' \ |_{X_1}\ C_2 \quad \texttt{<<no parses (char 2): !***X1 n X2 = \{\} >>} \quad \texttt{<<no parses (char 2): !***C' = C1 u C2 >>} \quad \texttt{<<no parses (char 2): !***X' = X1 u X2 >>}}{\Gamma \vdash \mathbf{var}\ vd, vd' \ |_{X'}\ C'}\ \textsc{MultiDecTypable}$$

```
Definition rules:          17 good    4 bad
Definition rule clauses: 33 good    12 bad
```