## Grammar

Base type

$$
\begin{array}{lll}
\tau & ::= & \\
& | & \textsc{Bool} \\
& | & \textsc{UInt}\langle n \rangle \\
& | & \textsc{Int}\langle n \rangle \\
& | & \textsc{Arr}\langle \tau, n \rangle
\end{array}
$$

Label

$$
\begin{array}{lll}
\ell & ::= & \\
& | & \textsc{Public} \\
& | & \textsc{Secret}
\end{array}
$$

Mutability

$$
\begin{array}{lll}
\sigma & ::= & \\
& | & \textsc{Const} \\
& | & \textsc{Mut}
\end{array}
$$

Function argument

$$
\begin{array}{lll}
v & ::= & \\
& | & e \quad \text{by value} \\
& | & \textsc{Mut}\ x \quad \text{by reference}
\end{array}
$$

Expression

$$
\begin{array}{lll}
e & ::= & \\
& | & \textsc{true} \\
& | & \textsc{false} \\
& | & k \quad \text{integer literal} \\
& | & x \quad \text{variable} \\
& | & a[e] \quad \text{array get} \\
& | & \ominus e \quad \text{unary op} \\
& | & e_1 \oplus e_2 \quad \text{binary op} \\
& | & f(v_1, \ldots, v_n) \quad \text{function call}
\end{array}
$$

ArrInit

$$
\begin{array}{lll}
init & ::= & \\
& | & \langle \tau, n \rangle \textsc{zeros} \quad \text{memset zero} \\
& | & \langle \tau, n \rangle \textsc{fill}(e) \quad \text{memset} \\
& | & \langle \tau, n \rangle x \Rightarrow e \quad \text{array comprehension} \\
& | & \langle \tau, n \rangle \{e_1, \ldots, e_n\} \quad \text{literal} \\
& | & \textsc{copy}(a) \quad \text{memcpy}
\end{array}
$$

Statement

$$
\begin{array}{lll}
s & ::= & \\
& | & s_1; s_2 \quad \text{sequence} \\
& | & \langle \tau, \ell, \sigma \rangle x := e \quad \text{variable declaration} \\
& | & \langle \textsc{Arr}\langle \tau, n \rangle, \ell, \sigma \rangle a := init \quad \text{array declaration} \\
& | & \langle \textsc{Arr}\langle \tau, n \rangle, \ell, \sigma \rangle a := \textsc{view}(a', e) \quad \text{array view declaration} \\
& | & x := e \quad \text{variable assignment} \\
& | & a[e_1] = e_2 \quad \text{array assignment} \\
& | & \textsc{if}\ e\ \{s_1\}\ \textsc{else}\ \{s_2\} \quad \text{conditional} \\
& | & \textsc{for}\ \langle \tau \rangle x\ \textsc{from}\ e_1\ \textsc{to}\ e_2\ \{s\} \quad \text{loop} \\
& | & \textsc{return}\ e \quad \text{return}
\end{array}
$$

## Type Lattice

$$\frac{n_1 < n_2}{\textsc{UInt}\langle n_1 \rangle <_\tau \textsc{UInt}\langle n_2 \rangle} \qquad \frac{n_1 < n_2}{\textsc{Int}\langle n_1 \rangle <_\tau \textsc{Int}\langle n_2 \rangle} \qquad \frac{}{\textsc{UInt}\langle n \rangle <_\tau \textsc{Int}\langle 2n \rangle}$$

$$\frac{}{\textsc{Public} <_\ell \textsc{Secret}} \qquad \frac{}{\textsc{Const} <_\sigma \textsc{Mut}} \qquad \frac{\tau_1 <_\tau \tau_2 \qquad \Gamma \vdash e : \langle \tau_1, \ell \rangle}{\Gamma \vdash e : \langle \tau_2, \ell \rangle} \qquad \frac{}{\ell \cup \ell = \ell}$$

$$\frac{}{\ell \cup \textsc{Secret} = \textsc{Secret}}$$

**Parameter Passing**

$$\frac{\Gamma \vdash e : \langle \tau, \ell_1 \rangle \qquad \ell_1 \leq_\ell \ell_2}{\langle \tau, \ell_2, \textsc{Const} \rangle \leftarrow e} \qquad \frac{\mu(x) = \langle \tau, \ell, \textsc{Mut} \rangle}{\langle \tau, \ell, \textsc{Mut} \rangle \leftarrow \textsc{Mut}\ x}$$

**Expressions**

$$\textsc{Var}\ \frac{\mu(x) = \langle \tau, \ell, \sigma \rangle}{\Gamma \vdash x : \langle \tau, \ell \rangle} \qquad \textsc{Unop}\ \frac{\Gamma \vdash e : \langle \tau_1, \ell_1 \rangle \qquad \ominus : \langle \tau_1, \ell_1 \rangle \to \langle \tau_2, \ell_2 \rangle}{\Gamma \vdash \ominus e : \langle \tau_2, \ell_2 \rangle}$$

$$\textsc{Binop}\ \frac{\Gamma \vdash e_1 : \langle \tau_1, \ell_1 \rangle \qquad \Gamma \vdash e_2 : \langle \tau_2, \ell_2 \rangle \qquad \oplus : \langle \tau_1, \ell_1 \rangle \to \langle \tau_2, \ell_2 \rangle \to \langle \tau_3, \ell_3 \rangle}{\Gamma \vdash e_1 \oplus e_2 : \langle \tau_3, \ell_3 \rangle}$$

$$\textsc{ArrGet}\ \frac{\mu(a) = \langle \textsc{Arr}\langle \tau, n \rangle, \ell, \sigma \rangle \qquad \Gamma \vdash e : \langle \textsc{UInt}\langle max \rangle, \textsc{Public} \rangle}{\Gamma \vdash a[e] : \langle \tau, \ell \rangle}$$

$$\textsc{FnCall}\ \frac{\mathbb{F}(f) = fdec(p_1, \ldots, p_n) : \langle \tau, \ell \rangle \qquad p_1 \leftarrow v_1 \qquad \cdots \qquad p_n \leftarrow v_n}{\Gamma \vdash f(v_1, \ldots, v_n) : \langle \tau, \ell \rangle} \qquad \textsc{True}\ \frac{}{\Gamma \vdash \textsc{true} : \langle bool, \textsc{Public} \rangle}$$

$$\textsc{False}\ \frac{}{\Gamma \vdash \textsc{false} : \langle bool, \textsc{Public} \rangle} \qquad \textsc{PosNumber}\ \frac{k >= 0 \qquad n = \lceil \log_2 k \rceil}{\Gamma \vdash k : \langle \textsc{UInt}\langle n \rangle, \textsc{Public} \rangle} \qquad \textsc{NegNumber}\ \frac{k < 0 \qquad n = \lceil \log_2 |k| \rceil + 1}{\Gamma \vdash k : \langle \textsc{Int}\langle n \rangle, \textsc{Public} \rangle}$$

**Array Initializers**

$$\textsc{ZeroInit}\ \frac{\tau = \textsc{UInt}\langle s \rangle \vee \tau = \textsc{Int}\langle s \rangle}{\Gamma \vdash \langle \tau, n \rangle \textsc{zeros} : \langle \textsc{Arr}\langle \tau, n \rangle, \textsc{Public} \rangle} \qquad \textsc{FillInit}\ \frac{\Gamma \vdash e : \langle \tau, \ell \rangle}{\Gamma \vdash \langle \tau, n \rangle \textsc{fill}(e) : \langle \textsc{Arr}\langle \tau, n \rangle, \ell \rangle}$$

$$\textsc{CompInit}\ \frac{\Gamma \vdash e : \langle \tau, \ell \rangle}{\Gamma \vdash \langle \tau, n \rangle x \Rightarrow e : \langle \textsc{Arr}\langle \tau, n \rangle, \ell \rangle} \qquad \textsc{LitInit}\ \frac{\Gamma \vdash e_i : \langle \tau, \ell_i \rangle \qquad \bigcup_i \ell_i = \ell}{\Gamma \vdash \langle \tau, n \rangle \{e_1, \ldots, e_n\} : \langle \textsc{Arr}\langle \tau, n \rangle, \ell \rangle}$$

$$\textsc{CopyInit}\ \frac{\mu(a) = \langle \textsc{Arr}\langle \tau, n \rangle, \ell, \sigma \rangle}{\Gamma \vdash \textsc{copy}(a) : \langle \textsc{Arr}\langle \tau, n \rangle, \ell \rangle}$$

**Statements**

$$\textsc{Seq}\ \frac{\Sigma\langle \ell_s \rangle \vdash s_1 : \ell'_s \qquad \Sigma\langle \ell'_s \rangle \vdash s_2 : \ell''_s}{\Sigma\langle \ell_s \rangle \vdash s_1; s_2 : \ell'_s \cup \ell''_s}$$

$$\textsc{VarDec}\ \frac{\Gamma \vdash e : \langle \tau, \ell_1 \rangle \qquad \tau \neq \textsc{Arr}\langle \tau', n \rangle \qquad \ell_1 \leq_\ell \ell_2}{\Sigma\langle \ell_s \rangle \vdash \langle \tau, \ell_2, \sigma \rangle x := e : \textsc{Public}} \quad \mu(x) = \langle \tau, \ell_2, \sigma \rangle \text{ (scoping?)}$$

$$\textsc{ArrDec}\ \frac{\Gamma \vdash \mathit{init} : \langle \textsc{Arr}\langle \tau, n \rangle, \ell_1 \rangle \qquad \ell_1 \leq_\ell \ell_2}{\Sigma\langle \ell_s \rangle \vdash \langle \textsc{Arr}\langle \tau, n \rangle, \ell_2, \sigma \rangle a := \mathit{init} : \textsc{Public}} \quad \mu(a) = \langle \textsc{Arr}\langle \tau, n \rangle, \ell_2, \sigma \rangle \text{ (scoping?)}$$

$$\textsc{ArrView}\ \frac{\mu(a) = \langle \textsc{Arr}\langle \tau, n \rangle, \ell, \sigma \rangle \qquad \Gamma \vdash e : \langle \textsc{UInt}\langle \mathit{max} \rangle, \textsc{Public} \rangle \qquad n' \leq n \qquad \ell \leq_\ell \ell' \qquad \sigma' \leq_\sigma \sigma}{\Sigma\langle \ell_s \rangle \vdash \langle \textsc{Arr}\langle \tau, n' \rangle, \ell', \sigma' \rangle a' := \textsc{view}(a, e) : \textsc{Public}} \quad \mu(a') = \langle \textsc{Arr}\langle \tau, n' \rangle, \ell', \sigma' \rangle \text{ (scoping?)}$$

$$\textsc{VarAssign}\ \frac{\mu(x) = \langle \tau, \ell_1, \textsc{Mut} \rangle \qquad \tau \neq \textsc{Arr}\langle \tau', n \rangle \qquad \Gamma \vdash e : \langle \tau, \ell_2 \rangle \qquad \ell_2 \leq_\ell \ell_1}{\Sigma\langle \ell_s \rangle \vdash x := e : \textsc{Public}}$$

$$\textsc{ArrAssign}\ \frac{\mu(a) = \langle \textsc{Arr}\langle \tau, n \rangle, \ell_1, \textsc{Mut} \rangle \qquad \Gamma \vdash e_1 : \langle \textsc{UInt}\langle \mathit{max} \rangle, \textsc{Public} \rangle \qquad \Gamma \vdash e_2 : \langle \tau, \ell_2 \rangle \qquad \ell_2 \leq_\ell \ell_1}{\Sigma\langle \ell_s \rangle \vdash a[e_1] := e_2 : \textsc{Public}}$$

$$\textsc{If}\ \frac{\Gamma \vdash e : \langle \textsc{Bool}, \ell \rangle \qquad \Sigma\langle \ell \cup \ell_s \rangle \vdash s_1 : \ell'_s \qquad \Sigma\langle \ell \cup \ell_s \rangle \vdash s_2 : \ell''_s}{\Sigma\langle \ell_s \rangle \vdash \textsc{if}\ e\ \{s_1\}\ \textsc{else}\ \{s_2\} : \ell'_s \cup \ell''_s}$$

$$\textsc{For}\ \frac{\Gamma \vdash e_1 : \langle \tau, \textsc{Public} \rangle \qquad \Gamma \vdash e_2 : \langle \tau, \textsc{Public} \rangle \qquad \tau = \textsc{UInt}\langle s \rangle \vee \tau = \textsc{Int}\langle s \rangle \qquad \Sigma\langle \ell_s \rangle \vdash s : \ell'_s}{\Sigma\langle \ell_s \rangle \vdash \textsc{for}\ \langle \tau \rangle x\ \textsc{from}\ e_1\ \textsc{to}\ e_2\ \{s\} : \ell'_s} \quad \mu(x) = \langle \tau, \textsc{Public}, \textsc{Const} \rangle \text{ (scoping?)}$$

$$\textsc{Ret}\ \frac{\Gamma \vdash e : \langle \tau, \ell_1 \rangle \qquad \mathbb{F}(f) = \mathit{fdec} : \langle \tau, \ell_2 \rangle \qquad \ell_1 \leq_\ell \ell_2}{\Sigma\langle \ell_s \rangle \vdash \textsc{return}\ e : \ell_s}$$