

# Typing rules for MiniC

## 1 Declarations

$$\text{VARDECL}(v:\mathbf{T}) \frac{\mathbf{T} \notin \{\mathbf{void}\}}{\text{add } \langle v : \mathbf{T} \rangle \text{ to } \Gamma}$$

$$\text{FUNPROTO}(f) \frac{}{\text{add } \langle f : (U_1, \dots, U_n) \rightarrow \mathbf{T} \rangle \text{ to } \Gamma}$$

$$\text{FUNDECL}(f) \frac{}{\text{add } \langle f : (U_1, \dots, U_n) \rightarrow \mathbf{T} \rangle \text{ to } \Gamma}$$

## 2 Expressions

$$\text{INTLITERAL}(i) \frac{}{\Gamma \vdash i : \mathbf{int}}$$

$$\text{STRLITERAL}(s) \frac{}{\Gamma \vdash s : \mathbf{char}[s.\text{length}+1]}$$

$$\text{CHRLITERAL}(c) \frac{}{\Gamma \vdash c : \mathbf{char}}$$

$$\text{VAREXPR}(v) \frac{\vdash \langle v : \mathbf{T} \rangle \in \Gamma}{\Gamma \vdash v : \mathbf{T}}$$

$$\text{FUNCALL}(f) \frac{\Gamma \vdash f : (U_1, \dots, U_n) \rightarrow \mathbf{T} \quad \Gamma \vdash x_1 : U_1 \quad \dots \quad \Gamma \vdash x_n : U_n}{\Gamma \vdash f(x_1, \dots, x_n) : \mathbf{T}}$$

$$\text{BINOP}(e_1, e_2, \text{Op} = \{\text{ADD}, \text{SUB}, \text{MUL}, \text{DIV}, \text{MOD}, \text{OR}, \text{AND}, \text{GT}, \text{LT}, \text{GE}, \text{LE}\}) \frac{\Gamma \vdash e_1 : \mathbf{int} \quad \vdash e_2 : \mathbf{int}}{\Gamma \vdash e_1 \text{Op } e_2 : \mathbf{int}}$$

$$\text{BINOP}(e_1, e_2, \text{Op} = \{\text{NE}, \text{EQ}\}) \frac{\Gamma \vdash e_1 : \mathbf{T} \notin \{\mathbf{StructType}, \mathbf{ArrayType}, \mathbf{void}\} \quad \vdash e_2 : \mathbf{T}}{\Gamma \vdash e_1 \text{Op } e_2 : \mathbf{int}}$$

$$\text{ARRAYACCESSEXPR}(e_1, e_2) \frac{\Gamma \vdash e_1 : \mathbf{T} \in \{\mathbf{ArrayType}_{\text{elemType}}, \mathbf{PointerType}_{\text{elemType}}\} \quad \vdash e_2 : \mathbf{int}}{\Gamma \vdash e_1[e_2] : \text{elemType}}$$

$$\text{FIELDACCESSEXPR}(e, \text{fieldName}) \frac{\Gamma \vdash e : \mathbf{ClassType}, \mathbf{StructType}_{\text{fieldName}:\mathbf{T}}}{\Gamma \vdash e.\text{fieldName} : \mathbf{T}}$$

$$\text{INSTANCEFUNCALEXPR}(e, f) \frac{\Gamma \vdash e : \mathbf{ClassType} \quad \Gamma \vdash f : (U_1, \dots, U_n) \rightarrow \mathbf{T} \quad \Gamma \vdash x_1 : U_1 \quad \dots \quad \Gamma \vdash x_n : U_n}{\Gamma \vdash e.f(x_1, \dots, x_n) : \mathbf{T}}$$

$$\text{NEWINSTANCE}(C) \frac{\vdash \langle C : \mathbf{CT} \in \mathbf{ClassType} \rangle \in \Gamma}{\Gamma \vdash \text{new } C() : \mathbf{CT}}$$

$$\text{VALUEATEXPR}(e) \frac{\Gamma \vdash e : \mathbf{PointerType}_{\mathbf{T}}}{\Gamma \vdash *e : \mathbf{T}}$$

$$\text{ADDRESSOFEXPR}(e) \frac{\Gamma \vdash e : \mathbf{T}}{\Gamma \vdash \&e : \mathbf{PointerType_T}}$$

$$\text{SIZEOF}(t) \frac{}{\Gamma \vdash \text{sizeof}(t) : \mathbf{int}}$$

$$\text{TYPECASTEXPR}(\text{CHAR TO INT}) \frac{\Gamma \vdash e : \mathbf{char}}{\Gamma \vdash (\mathbf{int})e : \mathbf{int}}$$

$$\text{TYPECASTEXPR}(\text{SUBCLASS TO ANCESTOR CLASS}) \frac{\Gamma \vdash e : \mathbf{T_1} \quad \mathbf{T_1} <: \mathbf{T_2}}{\Gamma \vdash (\mathbf{T_2})e : \mathbf{T_2}}$$

$$\text{TYPECASTEXPR}(\text{ARRAY TO POINTER}) \frac{\Gamma \vdash e : \mathbf{ArrayType_{elemType}}}{\Gamma \vdash (*\mathbf{elemType})e : \mathbf{PointerType_{elemType}}}$$

$$\text{TYPECASTEXPR}(\text{POINTER TO POINTER}) \frac{\Gamma \vdash e : \mathbf{PointerType_{elemType1}}}{\Gamma \vdash (*\mathbf{elemType2})e : \mathbf{PointerType_{elemType2}}}$$

$$\text{ASSIGN} \frac{\Gamma \vdash e_1 : \mathbf{T} \notin \{\mathbf{void}, \mathbf{ArrayType}\} \quad \Gamma \vdash e_2 : \mathbf{T}}{\Gamma \vdash e_1 = e_2 : \mathbf{T}}$$

### 3 Statements

$$\text{WHILE} \frac{\Gamma \vdash e : \mathbf{int}}{\Gamma \vdash \text{while}(e) \ s}$$

$$\text{IF}(\text{NO ELSE}) \frac{\Gamma \vdash e : \mathbf{int}}{\Gamma \vdash \text{if}(e) \ s}$$

$$\text{IF}(\text{WITH ELSE}) \frac{\Gamma \vdash e : \mathbf{int}}{\Gamma \vdash \text{if}(e) \ s_1 \ \text{else} \ s_2}$$

$$\text{RETURN}(\text{FROM } f) \frac{\Gamma \vdash f : (U_1, \dots, U_n) \rightarrow \mathbf{T} \quad \Gamma \vdash e : \mathbf{T}}{\Gamma \vdash \text{return } e}$$

$$\text{RETURN}(\text{NOTHING FROM } f) \frac{\Gamma \vdash f : (U_1, \dots, U_n) \rightarrow \mathbf{void}}{\Gamma \vdash \text{return } \emptyset}$$