Typing rules for MiniC

1 Declarations

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

$$\textsc{FunProto(f)}\ \frac{}{add\ \langle f : (U_1, \ldots, U_n) \to \mathbf{T}\rangle\ to\ \Gamma} \qquad \textsc{FunDecl(f)}\ \frac{}{add\ \langle f : (U_1, \ldots, U_n) \to \mathbf{T}\rangle\ to\ \Gamma}$$

2 Expressions

$$\textsc{IntLiteral(i)}\ \frac{}{\Gamma \vdash i : \mathbf{int}} \qquad \textsc{StrLiteral(s)}\ \frac{}{\Gamma \vdash s : \mathbf{char[s.length+1]}} \qquad \textsc{ChrLiteral(c)}\ \frac{}{\Gamma \vdash c : \mathbf{char}}$$

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

$$\textsc{FunCall(f)}\ \frac{\Gamma \vdash f : (U_1, \ldots, U_n) \to \mathbf{T} \qquad \Gamma \vdash x_1 : U_1 \qquad \ldots \qquad \Gamma \vdash x_n : U_n}{\Gamma \vdash f(x_1, \ldots, x_n) : \mathbf{T}}$$

$$\textsc{BinOp(e1,e2,Op=\{ADD,SUB,MUL,DIV,MOD,OR,AND,GT,LT,GE,LE\})}\ \frac{\Gamma \vdash e_1 : \mathbf{int} \qquad \vdash e_2 : \mathbf{int}}{\Gamma \vdash e_1 Op\ e_2 : \mathbf{int}}$$

$$\textsc{BinOp(e1,e2,Op=\{NE,EQ\})}\ \frac{\Gamma \vdash e_1 : \mathbf{T} \notin \{\mathbf{StructType,\ ArrayType,\ void}\} \qquad \vdash e_2 : \mathbf{T}}{\Gamma \vdash e_1\ Op\ e_2 : \mathbf{int}}$$

$$\textsc{ArrayAccessExpr(e1,e2)}\ \frac{\Gamma \vdash e_1 : \mathbf{T} \in \{\mathbf{ArrrayType_{elemType}}, \mathbf{PointerType_{elemType}}\} \qquad \vdash e_2 : \mathbf{int}}{\Gamma \vdash e_1[e_2] : \mathbf{elemType}}$$

$$\textsc{FieldAccessExpr(e,fieldName)}\ \frac{\Gamma \vdash e : \mathbf{StructType}_{fieldName:\mathbf{T}}}{\Gamma \vdash e.fieldName : \mathbf{T}}$$

$$\textsc{ValueAtExpr(e)}\ \frac{\Gamma \vdash e : \mathbf{PointerType_T}}{\Gamma \vdash *e : \mathbf{T}}$$

$$\textsc{AddressOfExpr(e)}\ \frac{\Gamma \vdash e : \mathbf{T}}{\Gamma \vdash \&e : \mathbf{PointerType_T}}$$

$$\textsc{SizeOf(t)}\ \frac{}{\Gamma \vdash\ sizeof(t) : \mathbf{int}}$$

$$\textsc{TypeCastExpr(char to int)}\ \frac{\Gamma \vdash e : \textbf{char}}{\Gamma \vdash\ (\textbf{int})e : \textbf{int}}$$

$$\textsc{TypeCastExpr(array to pointer)}\ \frac{\Gamma \vdash e : \textbf{ArrayType}_{\textbf{elemType}}}{\Gamma \vdash\ (\textbf{*elemType})e : \textbf{PointerType}_{\textbf{elemType}}}$$

$$\textsc{TypeCastExpr(pointer to pointer)}\ \frac{\Gamma \vdash e : \textbf{PointerType}_{\textbf{elemType1}}}{\Gamma \vdash\ (\textbf{*elemType2})e : \textbf{PointerType}_{\textbf{elemType2}}}$$

$$\textsc{Assign}\ \frac{\Gamma \vdash e_1 : \mathbf{T} \notin \{\textbf{void, ArrayType}\} \qquad \Gamma \vdash e_2 : \mathbf{T}}{\Gamma \vdash e_1 = e_2 : \mathbf{T}}$$

3 Statements

$$\textsc{While}\ \frac{\Gamma \vdash e : \textbf{int}}{\Gamma \vdash \mathit{while}(e)\ s}$$

$$\textsc{If(no else)}\ \frac{\Gamma \vdash e : \textbf{int}}{\Gamma \vdash \mathit{if}(e)\ s} \qquad\qquad \textsc{If(with else)}\ \frac{\Gamma \vdash e : \textbf{int}}{\Gamma \vdash \mathit{if}(e)\ s_1\ \mathit{else}\ s_2}$$

$$\textsc{Return(from f)}\ \frac{\Gamma \vdash f : (U_1, \ldots, U_n) \to \mathbf{T} \qquad \Gamma \vdash e : \mathbf{T}}{\Gamma \vdash \mathit{return}\ e}$$

$$\textsc{Return(nothing from f)}\ \frac{\Gamma \vdash f : (U_1, \ldots, U_n) \to \textbf{void}}{\Gamma \vdash \mathit{return}\ \varnothing}$$