Typing rules for MiniC

1 Declarations

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

$$\textsc{FunDecl(f)}\ \frac{}{add\ \langle f : \overline{U} \to \mathbf{T}\rangle\ to\ \Gamma}$$

2 Expressions

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

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

$$\textsc{FunCallExpr(f)}\ \frac{\vdash \langle f : \overline{U} \to \mathbf{T}\rangle \in \Gamma \qquad \Gamma \vdash \overline{Var} : \overline{U}}{\Gamma \vdash f(\overline{Var}) : \mathbf{T}}$$

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

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

$$\textsc{ArrayAccessExpr}\ \frac{\Gamma \vdash e_1 : \mathbf{T} \in \{\mathbf{ArrrayType}_{\mathbf{elemType}}, \mathbf{PointerType}_{\mathbf{elemType}}\} \qquad \vdash e_2 : \mathbf{int}}{\Gamma \vdash e_1[e_2] : \mathbf{elemType}}$$

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

$$\textsc{ValueAtExpr}\ \frac{\Gamma \vdash e : \mathbf{PointerType}_{\mathbf{elemType}}}{\Gamma \vdash *e : \mathbf{elemType}}$$

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

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

$$\textsc{TypeCastExpr(array to pointer)}\ \frac{\Gamma \vdash e : \mathbf{ArrayType_{elemType}}}{\Gamma \vdash\ (\mathbf{*elemType})e : \mathbf{PointerType_{elemType}}}$$

$$\textsc{TypeCastExpr(pointer to pointer)}\ \frac{\Gamma \vdash e : \mathbf{PointerType_{elemType1}}}{\Gamma \vdash\ (\mathbf{*elemType2})e : \mathbf{PointerType_{elemType2}}}$$

3 Statements

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

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

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

$$\textsc{Return(from f)}\ \frac{\Gamma \vdash f : \overline{U} \to \mathbf{T} \qquad \Gamma \vdash e : \mathbf{T}}{\Gamma \vdash return\ e}$$

$$\textsc{Return(nothing from f)}\ \frac{\Gamma \vdash f : \overline{U} \to \mathbf{void}}{\Gamma \vdash return\ \varnothing}$$