Type Rules

AST Node Type	AST rule	Constraints and Actions
Prog	Program ::= IDENT Dec* Stmt*	ProgramName ← IDENT
Dec	Dec ::= Type IDENT	TypeOf(IDENT) = ⊥ IDENT ≠ ProgramName TypeOf(IDENT) ← Type
AlternativeStmt	Stmt ::= Expr (Stmt*)$_{\text{ifStmtList}}$ (Stmt*)$_{\text{elseStmtList}}$	TypeOf(Expr) = boolean
AssignExprStmt	Stmt ::= IDENT Expr	TypeOf(Expr) = TypeOf(IDENT)
AssignPixelStmt	Stmt ::= IDENT Pixel	TypeOf(IDENT) = pixel
FileAssignStmt	Stmt ::= IDENT FileName	TypeOf(IDENT) = image

ScreenLocationAssignmentStmt	Stmt ::= IDENT Expr$_{\text{xScreenExpr}}$ Expr$_{\text{yScreenExpr}}$	TypeOf(IDENT) = image;
		TypeOf(Expr$_{\text{xScreenExpr}}$)= int;
		TypeOf(Expr$_{\text{yScreenExpr}}$)= int

SetVisibleAssignmentStmt	Stmt ::= IDENT Expr	TypeOf(IDENT) = image; TypeOf(Expr) = boolean

ShapeAssignment Stmt	Stmt ::= IDENT Expr$_{\text{width}}$ Expr$_{\text{height}}$	TypeOf(IDENT) = image;
		TypeOf(Expr$_{\text{width}}$)=int;
		TypeOf(Expr$_{\text{height}}$)=int

SinglePixelAssignmentStatement	Stmt ::= IDENT Expr$_{\text{xExpr}}$ Expr$_{\text{yExpr}}$ Pixel	TypeOf(IDENT) = image;
		TypeOf(Expr$_{\text{xExpr}}$) = int;
		TypeOf(Expr$_{\text{yExpr}}$) = int

SingleSampleAssignmentStmt	Stmt ::= IDENT Expr$_{\text{xExpr}}$ Expr$_{\text{yExpr}}$ COLOR Expr$_{\text{rhsExpr}}$	TypeOf(IDENT) = image TypeOf(Expr$_{\text{xExpr}}$) = int TypeOf(Expr$_{\text{yExpr}}$) = int TypeOf(Expr$_{\text{rhsExpr}}$) = int
IterationStatement	Stmt ::= Expr Stmt*	TypeOf(Expr) = boolean
PauseStatement	Stmt ::= Expr	TypeOf(Expr) = int
BinaryExpr	Expr ::= Expr$_{\text{e0}}$ Op Expr$_{\text{e1}}$	See table below
BooleanLitExpr	Expr ::= BooleanLit	TypeOf(Expr) ←boolean

ConditionalExpr	Expr ::= Expr$_{\text{condition}}$ Expr$_{\text{trueValue}}$ Expr$_{\text{falseValue}}$	TypeOf(Expr$_{\text{condition}}$) = Boolean
		TypeOf(Expr$_{\text{trueValue}}$) = TypeOf(Expr$_{\text{falseValue}}$)
		TypeOf(Expr) ← TypeOf(Expr$_{\text{trueValue}}$)

IdentExpr	Expr ::= IDENT	TypeOf(Expr) ← TypeOf(Ident)
ImageAttributeExpr	Expr ::= IDENT SELECTOR	TypeOf(Ident) = image TypeOf(Expr) ← int
IntLitExpr	Expr ::= INT_LIT	TypeOf(Expr) ← int
PreDefExpr	Expr ::= CONSTANT_LIT	TypeOf(Expr) ← int

SampleExpr	Expr ::= IDENT Expr$_{xLoc}$ Expr$_{yLoc}$ COLOR	TypeOf(IDENT) = image TypeOf(Expr$_{xLoc}$) = int TypeOf(Expr$_{yLoc}$) = int TypeOf(Expr) ← int
Pixel	Pixel ::= Expr$_{redExpr}$ Expr$_{greenExpr}$ Expr$_{blueExpr}$	TypeOf(Expr$_{redExpr}$) = int TypeOf(Expr$_{greenExpr}$) = int TypeOf(Expr$_{blueExpr}$) = int

Binary Expressions

Op	Constraints	Type of result (TypeOf←)
&, |	TypeOf(Expr$_{e0}$) = boolean TypeOf(Expr$_{e1}$) = boolean	boolean
+, - , *, /, %	TypeOf(Expr$_{e0}$) = int TypeOf(Expr$_{e1}$) = int	int
==, !=	TypeOf(Expr$_{e0}$) = TypeOf(Expr$_{e1}$)	boolean
≪, ≫	TypeOf(Expr$_{e0}$) = int TypeOf(Expr$_{e1}$) = int	int
<, >, ≤ , ≥	TypeOf(Expr$_{e0}$) = int TypeOf(Expr$_{e1}$) = int	boolean