

**MyPL Type Inference Rules.** The goal of the following rules are to help clarify type inference and type checking in **MyPL**. We make the following assumptions.

- The notation  $e : t$  says that expression  $e$  has type  $t$ .
- The **null** value has type **void** (i.e.,  $\text{null} : \text{void}$ ) and literals (constants)  $c$  have their corresponding type  $t$  (i.e.,  $c : t$ ).
- In general,  $e$  denotes an expression,  $t$  a data type,  $x$  a variable/field name,  $s$  a struct name, and  $f$  a function name.
- Function types are denoted as mappings from parameter type lists (in the order of function parameters) to return types. For example,  $f : \text{int}, \text{string} \rightarrow \text{bool}$  says  $f$  takes an int and string, and returns a bool.
- Struct types are represented as mappings from field names to types. For example,  $s : \{x_1 \rightarrow t_1, \dots, x_n \rightarrow t_n\}$  says struct  $s$  has fields  $x_1$  to  $x_n$  with types  $t_1$  to  $t_n$  respectively.
- Array types are represented using square brackets, e.g., an int array is denoted as  $[\text{int}]$ .
- $\Gamma$  is the typing context (environment). The notation  $\Gamma \vdash e : t$  says that the current context implies that expression  $e$  has type  $t$ . Similarly, the notation  $\Gamma, \text{stmt} \vdash e : t$  says that the current context extended with the statement implies  $e$  has type  $t$ . We take some liberties below by assuming we are “in” the statement (stmt) when it extends the scope.
- Unlike syntax rules, the typing rules are meant to provide a guide to some of the details as opposed to an implementation strategy.

### Typing Rules for MyPL Expressions:

$$\frac{\Gamma \vdash e_1 : \text{string} \quad \Gamma \vdash e_2 : \text{string}}{\Gamma \vdash e_1 + e_2 : \text{string}} \quad (1)$$

$$\frac{\Gamma \vdash e_1 : t \quad \Gamma \vdash e_2 : t \quad t \in \{\text{int}, \text{double}\} \quad \text{op} \in \{+, -, *, /\}}{\Gamma \vdash e_1 \text{ op } e_2 : t} \quad (2)$$

$$\frac{\Gamma \vdash e_1 : t_1 \quad \Gamma \vdash e_2 : t_2 \quad (t_1 = t_2 \vee t_1 = \text{void} \vee t_2 = \text{void}) \quad \text{op} \in \{==, !=\}}{\Gamma \vdash e_1 \text{ op } e_2 : \text{bool}} \quad (3)$$

$$\frac{\Gamma \vdash e_1 : t \quad \Gamma \vdash e_2 : t \quad t \in \{\text{int}, \text{double}, \text{string}\} \quad \text{op} \in \{<, >, <=, >=\}}{\Gamma \vdash e_1 \text{ op } e_2 : \text{bool}} \quad (4)$$

$$\frac{\Gamma \vdash e_1 : \text{bool} \quad \Gamma \vdash e_2 : \text{bool}}{\Gamma \vdash e_1 \text{ and } e_2 : \text{bool}} \quad (5)$$

$$\frac{\Gamma \vdash e_1 : \text{bool} \quad \Gamma \vdash e_2 : \text{bool}}{\Gamma \vdash e_1 \text{ or } e_2 : \text{bool}} \quad (6)$$

$$\frac{\Gamma \vdash e : \text{bool}}{\Gamma \vdash \text{not } e : \text{bool}} \quad (7)$$

### Typing Rules for MyPL Statements:

$$\frac{\Gamma \vdash e : t \vee e : \mathbf{void}}{\Gamma, \mathbf{var } x : t = e \vdash x : t} \quad (8)$$

$$\frac{\Gamma \vdash e : t \wedge t \neq \mathbf{void}}{\Gamma, \mathbf{var } x = e \vdash x : t} \quad (9)$$

$$\overline{\Gamma, \mathbf{var } x : t \vdash x : t} \quad (10)$$

$$\frac{\Gamma \vdash x : t}{\Gamma, x = e \vdash e : t \vee e : \mathbf{void}} \quad (11)$$

$$\overline{\Gamma, \mathbf{while } e \{ \dots \} \vdash e : \mathbf{bool}} \quad (12)$$

$$\frac{\Gamma \vdash e_1 : \mathbf{int} \wedge e_2 : \mathbf{int}}{\Gamma, \mathbf{for } x \mathbf{ from } e_1 \mathbf{ to } e_2 \{ \dots \} \vdash x : \mathbf{int}} \quad (13)$$

$$\overline{\Gamma, \mathbf{if } e \{ \dots \} \vdash e : \mathbf{bool}} \quad (14)$$

$$\overline{\Gamma, \mathbf{if } \{ \dots \} \mathbf{ else if } e \{ \dots \} \vdash e : \mathbf{bool}} \quad (15)$$

### Typing Rules for MyPL Structs:

$$\overline{\Gamma, \mathbf{struct } s \{ x_1 : t_1, \dots, x_n : t_n \} \vdash s : \{ x_1 \rightarrow t_1, \dots, x_n \rightarrow t_n \}} \quad (16)$$

$$\frac{\Gamma \vdash e : s \quad \Gamma \vdash s : \{ \dots, x_i \rightarrow t_i, \dots \}}{\Gamma \vdash e.x_i : t_i} \quad (17)$$

$$\frac{\Gamma \vdash s : \{ x_1 \rightarrow t_1, \dots, x_n \rightarrow t_n \} \quad \Gamma \vdash e_1 : t_1 \vee e_1 : \mathbf{void} \quad \dots \quad \Gamma \vdash e_n : t_n \vee e_n : \mathbf{void}}{\Gamma \vdash \mathbf{new } s(e_1, \dots, e_n) : s} \quad (18)$$

### Typing Rules for MyPL Functions:

$$\frac{}{\Gamma, t \ f(x_1: t_1, \dots, x_n: t_n)\{ \dots \} \vdash f: t_1, \dots, t_n \rightarrow t} \quad (19)$$

$$\frac{\Gamma \vdash f: t_1, \dots, t_n \rightarrow t \quad \Gamma \vdash e_1: t_1 \vee e_1: \mathbf{void} \quad \dots \quad \Gamma \vdash e_n: t_n \vee e_n: \mathbf{void}}{\Gamma \vdash f(e_1, \dots, e_n): t} \quad (20)$$

$$\frac{\Gamma \vdash \mathit{return}: t}{\Gamma, \mathbf{return} \ e \vdash e: t \vee e: \mathbf{void}} \quad \dagger \quad (21)$$

### Additional Typing Rules for MyPL Arrays:

$$\frac{\Gamma \vdash e: \mathbf{int} \quad t \notin \{\mathbf{void}, [t']\}}{\Gamma \vdash \mathbf{new} \ t[e]: [t]} \quad (22)$$

$$\frac{\Gamma \vdash e_1: [t] \quad \Gamma \vdash e_2: \mathbf{int}}{\Gamma \vdash e_1[e_2]: t} \quad (23)$$

---

<sup>†</sup>where “*return*” is a special variable assumed in each function context with the corresponding return type