SmallC Formal Type Checking and Type Inference Rules

April 16, 2025

1 Introduction

This document presents the formal type checking and type inference rules using the syntax used in lecture.

2 Preliminaries

2.1. Environments. Type rules define judgments that make use of a context Γ (Stylized as G). The rules show two operations on environments:

- $G(x)$ means to look up the type of x mapped to in the context G. This operation is undefined if there is no mapping for x in G.
- The second operation is written $G[x : t]$. It defines a new environment that is the same as G but maps x to t. It thus overrides any prior mapping for x in G.

2.2. Constraints. In addition to the context, we also will need to keep track of the constraints we should use when we try and infer types of variables.

- A constrain set is a set of bindings of one type to another. We will use the syntax $t_1 : t_2$ to indicate that type t_1 should be the same as type t_2.
- To keep types consistent, we may use a placeholder type to refer to an expression. You can see more in the below constraints rules. These placeholder types will look like t or t_n. If two types use the same n value, then they are considered the same type.

2.3. Syntax. In this document, we have simplified the presentation of the syntax, so it may not correspond exactly to the files that your checker/inferencer will read in. For example, we write `while` $e\,s$ to represent the syntax of a while-loop, where e is the guard and s is the body. This corresponds to `While of expr * stmt` in the `ast.ml` file. Hopefully the connection between what we show here at that file is clear enough from context.

2.4. Error conditions. The semantics here defines only *correct* evaluations. It says nothing about what happens when, say, you have a type error. For example, for the rules below there is no type t for which you can prove the judgment $\Gamma \vdash 1 + \mathbf{true} - - > t$. In your actual implementation, erroneous programs will cause an exception to be raised, as indicated in the project README.

3 Type Checking Rules

$$\text{int}\ \frac{}{G \vdash n : int} \qquad \text{bool}\ \frac{}{G \vdash b : bool} \qquad \text{value}\ \frac{}{G \vdash read() : UT} \qquad \text{var-lookup}\ \frac{G(x) = t}{G \vdash x : t}$$

$$\text{Subtype Bool}\ \frac{}{UT <: bool} \qquad \text{Subtype Int}\ \frac{}{UT <: int} \qquad UT = Unknown\_Type$$

$$\text{add}\ \frac{G \vdash e_1 : int \quad G \vdash e_2 : int}{G \vdash e_1 + e_2 : int} \qquad \text{sub}\ \frac{G \vdash e_1 : int \quad G \vdash e_2 : int}{G \vdash e_1 - e_2 : int} \qquad \text{mult}\ \frac{G \vdash e_1 : int \quad G \vdash e_2 : int}{G \vdash e_1 * e_2 : int}$$

$$\text{div}\ \frac{G \vdash e_1 : int \quad G \vdash e_2 : int}{G \vdash e_1 / e_2 : int} \qquad \text{pow}\ \frac{G \vdash e_1 : int \quad G \vdash e_2 : int}{G \vdash e_1 \hat{}\, e_2 : int}$$

$$\text{and}\ \frac{G \vdash e_1 : bool \quad G \vdash e_2 : bool}{G \vdash e_1 \ \&\&\ e_2 : bool} \qquad \text{or}\ \frac{G \vdash e_1 : bool \quad G \vdash e_2 : bool}{G \vdash e_1 \ ||\ e_2 : bool} \qquad \text{not}\ \frac{G \vdash e : bool}{G \vdash \text{not } e : bool}$$

$$\text{equal}\ \frac{G \vdash e_1 : t \quad G \vdash e_2 : t}{G \vdash e_1 == e_2 : bool} \qquad \text{not-equal}\ \frac{G \vdash e_1 : t \quad G \vdash e_2 : t}{G \vdash e_1 != e_2 : bool} \qquad \text{greater-equal}\ \frac{G \vdash e_1 : t \quad G \vdash e_2 : t}{G \vdash e_1 >= e_2 : bool}$$

$$\text{less-equal}\ \frac{G \vdash e_1 : t \quad G \vdash e_2 : t}{G \vdash e_1 <= e_2 : bool} \qquad \text{greater}\ \frac{G \vdash e_1 : t \quad G \vdash e_2 : t}{G \vdash e_1 > e_2 : bool} \qquad \text{less}\ \frac{G \vdash e_1 : t \quad G \vdash e_2 : t}{G \vdash e_1 < e_2 : bool}$$

Statements can modify the context. So $\rightarrow G'$ means the context updated the environment to G'.

$$\text{seq}\ \frac{G \vdash s_1 : () \rightarrow G' \quad G' \vdash s_2 : () \rightarrow G''}{G \vdash s_1; s_2 : () \rightarrow G''} \qquad \text{print}\ \frac{G \vdash e : t}{G \vdash printf(e) : () \rightarrow G}$$

$$\text{if}\ \frac{G \vdash e_1 : bool \quad G \vdash s_1 : () \rightarrow G' \quad G \vdash s_2 : () \rightarrow G''}{G \vdash if\ e\ s_1\ s_2 : () \rightarrow G' \cup G''} \qquad \text{while}\ \frac{G \vdash e : bool \quad G \vdash s : () \rightarrow G'}{G \vdash while\ e\ s : () \rightarrow G'}$$

$$\text{for (x previously assigned)}\ \frac{G(x) : int \quad G \vdash e_1 : int \quad G \vdash e_2 : int \quad G \vdash s : () \rightarrow G'}{G \vdash for\ x\ e_1\ e_2\ s : () \rightarrow G'}$$

$$\text{for (x undefined)}\ \frac{G \vdash e_1 : int \quad G \vdash e_2 : int \quad G[x : int] \vdash s : () \rightarrow G'}{G \vdash for\ x\ e_1\ e_2\ s : () \rightarrow G'}$$

$$\text{assign}\ \frac{G \vdash e : t_1 \qquad Ast(x, t_0, e) \qquad t_1 <: t_0}{G \vdash x = e : () \rightarrow G[x : t_0]} \qquad \text{assign-UT}\ \frac{G \vdash e : t \qquad Ast(x, UT, e)}{G \vdash x = e : () \rightarrow G[x : UT]}$$

4 Type Inference Rules

$$\text{int}\ \frac{}{G \vdash n : int \dashv \{\}} \qquad \text{bool}\ \frac{}{G \vdash\ b\ :\ bool\ \dashv\ \{\}} \qquad \text{value}\ \frac{}{G \vdash read() : t \dashv \{\}} \qquad \text{var-lookup}\ \frac{G(x) = t}{G \vdash\ x : t \dashv \{\}}$$

$$\text{add}\ \frac{G \vdash e_1 : t_1 \dashv C_1 \quad G \vdash e_2 : t_2 \dashv C_2}{G \vdash e_1 + e_2 : int \dashv \{t_1 : int, t_2 : int\} \cup C_1 \cup C_2} \qquad \text{sub}\ \frac{G \vdash e_1 : t_1 \dashv C_1 \quad G \vdash e_2 : t_2 \dashv C_2}{G \vdash e_1 - e_2 : int \dashv \{t_1 : int, t_2 : int\} \cup C_1 \cup C_2}$$

$$\text{mult}\ \frac{G \vdash e_1 : t_1 \dashv C_1 \quad G \vdash e_2 : t_2 \dashv C_2}{G \vdash e_1 * e_2 : int \dashv \{t_1 : int, t_2 : int\} \cup C_1 \cup C_2} \qquad \text{div}\ \frac{G \vdash e_1 : t_1 \dashv C_1 \quad G \vdash e_2 : t_2 \dashv C_2}{G \vdash e_1 / e_2 : int \dashv \{t_1 : int, t_2 : int\} \cup C_1 \cup C_2}$$

$$\text{pow}\ \frac{G \vdash e_1 : t_1 \dashv C_1 \quad G \vdash e_2 : t_2 \dashv C_2}{G \vdash e_1 \hat{}\, e_2 : int \dashv \{t_1 : int, t_2 : int\} \cup C_1 \cup C_2} \qquad \text{and}\ \frac{G \vdash e_1 : t_1 \dashv C_1 \quad G \vdash e_2 : t_2 \dashv C_2}{G \vdash e_1 - e_2 : bool \dashv \{t_1 : bool, t_2 : bool\} \cup C_1 \cup C_2}$$

$$\text{or}\ \frac{G \vdash e_1 : t_1 \dashv C_1 \quad G \vdash e_2 : t_2 \dashv C_2}{G \vdash e_1 + e_2 : bool \dashv \{t_1 : bool, t_2 : bool\} \cup C_1 \cup C_2} \qquad \text{equal}\ \frac{G \vdash e_1 : t_1 \dashv C_1 \quad G \vdash e_2 : t_2 \dashv C_2}{G \vdash e_1 == e_2 : bool \dashv \{t_1 : t_2\} \cup C_1 \cup C_2}$$

$$\text{not-equal}\ \frac{G \vdash e_1 : t_1 \dashv C_1 \quad G \vdash e_2 : t_2 \dashv C_2}{G \vdash e_1 != e_2 : bool \dashv \{t_1 : t_2\} \cup C_1 \cup C_2} \qquad \text{greater-equal}\ \frac{G \vdash e_1 : t_1 \dashv C_1 \quad G \vdash e_2 : t_2 \dashv C_2}{G \vdash e_1 >= e_2 : bool \dashv \{t_1 : t_2\} \cup C_1 \cup C_2}$$

$$\text{greater}\ \frac{G \vdash e_1 : t_1 \dashv C_1 \quad G \vdash e_2 : t_2 \dashv C_2}{G \vdash e_1 > e_2 : bool \dashv \{t_1 : t_2\} \cup C_1 \cup C_2} \qquad \text{less-equal}\ \frac{G \vdash e_1 : t_1 \dashv C_1 \quad G \vdash e_2 : t_2 \dashv C_2}{G \vdash e_1 <= e_2 : bool \dashv \{t_1 : t_2\} \cup C_1 \cup C_2}$$

$$\text{less}\ \frac{G \vdash e_1 : t_1 \dashv C_1 \quad G \vdash e_2 : t_2 \dashv C_2}{G \vdash e_1 < e_2 : bool \dashv \{t_1 : t_2\} \cup C_1 \cup C_2} \qquad \text{not}\ \frac{G \vdash e : t \dashv C_1}{G \vdash \text{not } e : bool \dashv \{t : bool\} \cup C_1 \cup C_2}$$

Statements can modify the context. So $\to G'$ means the context updated the environment to G'.

$$\text{seq}\ \frac{G \vdash s_1 : () \dashv C_1 \to G' \quad G' \vdash s_2 : () \dashv C_2 \to G''}{G \vdash s_1; s_2 : () \dashv C_1 \cup C_2 \to G''} \qquad \text{assign}\ \frac{G \vdash e : t_1 \dashv C \quad Ast(, t_0, e)}{G \vdash x = e : () \dashv \{t_0 : t_1, G(x) : t_0\} \cup C \to G[x : t_1]}$$

$$\text{if}\ \frac{G \vdash e_1 : t \dashv C_1 \quad G \vdash s_1 : () \dashv C_2 \to G' \quad G \vdash s_2 : () \dashv C_3 \to G''}{G \vdash if\ e\ s_1\ s_2 : () \dashv \{t : bool\} \cup C_1 \cup C_2 \cup C_3 \to G' \cup G''}$$

$$\text{for}\ \frac{G \vdash e_1 : t_1 \dashv C_1 \quad G \vdash e_2 : t_2 \dashv C_2 \quad G \vdash s : () \dashv C_3 \to G'}{G \vdash for\ x\ e_1\ e_2\ s : () \dashv \{t_1 : int, t_2 : int, G'(x) : int\} \cup C_1 \cup C_2\ C_3 \to G'}$$

$$\text{while}\ \frac{G \vdash e : t \dashv C_1 \quad G \vdash s : () \dashv C_2 \to G'}{G \vdash while\ e\ s : () \dashv \{t : bool\} \cup C_1 \cup C_2 \to G'} \qquad \text{print}\ \frac{G \vdash e : t \dashv C}{G \vdash printf(e) : () \dashv C \to G}$$