Bryan Melanson

How to Not Fail

Algorithm Correctness and Complexity

While never going to class

Computer Engineering 2020 $\mathcal{BM}$

Contents

1 Proof Outline Logic

1.1 Assertions

An assertion is a **condition** that is expected to be true every time execution passes a particular point in a program. Available at run time in C, C++ and Java, an assertion which proves to not be true will prevent the program from running.

Assertions are valuable from a documentation standpoint, and for testing.

1.2 Substitutions

Replacing a free (ie global, not restricted to a certain scope) variable follows the syntax $P[x : E]$ where all occurences of x are replaced by the expression E. This can be extended to multiple variables as $[x, y : z, x]$.

1.3 Contracts

A condition might be required before other conditions become valid:

$953I \leq V \leq 1050I$, provided $0 \leq V \leq 10$
OR [$953I \leq V \leq 1050I$, $0 \leq V \leq 10$]
A contract [y = 5, x = 5] can be represented as:

$\{y = 4\}$
$x := y + 1$
$\{x = 5\}$

This is a **Hoare Triple**, made up of a precondition, a command and a postcondition. This statement can be considered **Partially Correct** if for any values, when the precondition is satisfied, and the command is executed, it can only end in a state satisfying the postcondition.

1.4 Proof Outlines

A **Proof Outline** is a command annotated with assertions. It represents the summary of a proof, if correct. This can be proven to be correct using partial correctness. In sequence, if $\{P\}\ S\ \{Q\}\ T\ \{R\}$, and both $\{P\}\ S\ \{Q\}$ and $\{Q\}\ T\ \{R\}$ can be proven to be partially correct, then the full statement is a partially correct outline.

In a proof outline, each command will be preceded by an assertion. This is the **precondition**.

1.4.1 Two Tailed If Rule

$\{P\}$ if (E) $\{Q\}$ else $\{Q\}\ T\ \{R\}$ relies on the if/else conditions being provable along with the equivalent E.

1.4.2 One Tailed If Rule

$\{P\}$ if (E) $\{Q\}\ T\ \{R\}$ relies on the if conditions being provable with E, and $\neg E$ and R being provable.

The initial precondition of a loop P is known as the **Invariant**. It must be proven true at the start of each iteration of a loop, as well as when the loop terminates.