Chapter 1: Specifications.

In which we learn about specifying problems.

My readers might accuse me of stating the obvious, but there are times when it should be stated. The very beginning of the programming task is to describe precisely what the program is supposed to achieve. Even here, we must be careful not to fall into the trap of describing what is to be achieved in terms of how it is to be achieved. These are two things here which need to be separated, and kept separate. **What** is to be achieved and **how** it is to be achieved are different and mixing them can lead to confusion.

To facilitate our precise description of **what** is to be achieved we make use of a notation called a **Hoare triple**.[1]

Here is an example of a Hoare triple.

$$\{P\}\ S\ \{Q\}$$

P and Q describe **States**. This means they tell you some[2] things that will be true at that point. S describes a mechanism, you can think of this as some instructions. P is generally called the **Precondition** and Q is called the **Postcondition**. Here are a few examples:

$$\{0 \leq x\}\ S\ \{y = x^2\}$$

Here the Precondition tells us that a variable x will have a value that is at least 0 at this point. The Postcondition tells us that at that point the variable y will have a value which is x squared.

$$\{x, y : nat\}\ S\ \{z = max(x,y)\}$$

Here the Precondition tells us that both x and y contain natural numbers and the Postcondition tells us that at that point z will contain the largest of the values x and y.

A Hoare triple is true whenever, starting in a state described by P, the mechanism S, if it terminates, will terminate in a state described by Q. Let us look at a few examples. In what follows we make use of an assignment of the form x := E, you can think of

---

[1] Named after C.A.R. Hoare who was once the professor of Computer Science in Queens University in Belfast before moving on to Oxford. He has since retired and joined Microsoft.

[2] We only focus on the true things that are of interest to us. Of course many other things may be true in a particular state, for example 1+1=2 :-) but it would be silly to write that down.

this operationally as a command which evaluates the expression E and places the result in the variable x. Later we will show you non-operational interpretations of assignment.

$\{x = 7\}\ x := x + 4\ \{x = 11\}$

$\{x = 8\}\ x := x + 3\ \{x > 10\}$

$\{x > 5\}\ x := x + 1\ \{x > 6\}$

$\{x = 12\}\ x := x - 2\ \{ x > 11\}$

$\{x > 100\}\ x := x + 2\ \{x < 200\}$

The first 3 of these are valid, the final 2 are invalid.

Some Laws of Hoare Triples.

$\{P\}\ S\ \{Q\} \wedge \{P\}\ S\ \{R\} \Rightarrow \{P\}\ S\ \{Q \wedge R\}$

$\{P\}\ S\ \{Q\} \wedge \{R\}\ S\ \{Q\} \Rightarrow \{P \vee R\}\ S\ \{Q\}$

$\{P\}\ S\ \{Q\} \wedge Q \Rightarrow R \Rightarrow \{P\}\ S\ \{R\}$ "weakening/strengthening post"

$\{P\}\ S\ \{Q\} \wedge R \Rightarrow P \Rightarrow \{R\}\ S\ \{Q\}$ "weakening/strengthening pre"

$\{P\}\ S0\ \{Q\} \wedge \{Q\}\ S1\ \{R\} \Rightarrow \{P\}\ S0;S1\ \{R\}$ "concatenation"

It is worth noting that these laws apply to all mechanisms and not just programs.

Hoare introduced these laws in the early 1970's and used them to prove the correctness of programs. It was generally believed that the effort involved in proving even the simplest programs to be correct was very big. Indeed, we now know that proving a program to be correct after it has been written is indeed difficult if not impossible. [3]

One of the world's greatest computing scientists, Edsger W Dijkstra, developed Hoare's work further and eventually developed the method of programming which we now use. His first major contribution to the field was the notion of the Weakest Precondition.

---

[3] After the fact proof is generally known as program verification. It is still popular and people devote lots of effort into developing tools to assist in the proof.

An introduction to the Weakest Precondition.

Consider the following Hoare triples

(0) $\{ 11 \leq x \}\ x := x + 1\ \{ 5 \leq x \}$
(1) $\{ 10 \leq x \}\ x := x + 1\ \{ 5 \leq x \}$
(2) $\{ 9 \leq x \}\ x := x + 1\ \{ 5 \leq x \}$
(3) $\{ 8 \leq x \}\ x := x + 1\ \{ 5 \leq x \}$
(4) $\{ 7 \leq x \}\ x := x + 1\ \{ 5 \leq x \}$
(5) $\{ 6 \leq x \}\ x := x + 1\ \{ 5 \leq x \}$
(6) $\{ 5 \leq x \}\ x := x + 1\ \{ 5 \leq x \}$
(7) $\{ 4 \leq x \}\ x := x + 1\ \{ 5 \leq x \}$
(8) $\{ 3 \leq x \}\ x := x + 1\ \{ 5 \leq x \}$

We note that (0) through (7) are all valid, (8) is not. In each of the examples the assignment and the postconditions remain the same, what changes is the precondition. They form a weakening chain

$$\{11 \leq x\} \Rightarrow \{10 \leq x\} \Rightarrow \ldots\ldots \Rightarrow \{4 \leq x\}$$

In this chain we say that $\{11 \leq x\}$ is the strongest and $\{4 \leq x\}$ is the weakest.

$\{4 \leq x\}$ is **the minimum which must be true** in order that performing the assignment $x := x + 1$ will bring us to the state $\{5 \leq x\}$. We say that $\{4 \leq x\}$ is the **Weakest Precondition.**

We write the Weakest Precondition as follows

$$WP.(x := x + 1).(5 \leq x)$$

The relationship between Hoare triples and WP.

We link these two as follows

$$\{P\}\ S\ \{Q\} \quad \equiv \quad P \Rightarrow WP.S.Q$$

Weakest preconditions were first proposed by Dijkstra in the early 1970's. He used them to describe the semantics of a minimal language called the Guarded Command Language which we use.

The guarded command language has become the standard way in which Computing Scientists express algorithms. It has a very small number of commands and is machine independent which makes it an excellent vehicle for a clean and precise

description of algorithms. Weakest Preconditions have proved to be even more useful and form the basis of the methods of program calculation which we shall explore.

Exercises.

Determine whether the following Hoare triples are valid.

$\{y > 16\}\ x := x + 2\ \{y > 15\}$

$\{x = 4\}\ x := x + 1\ \{x > 4\}$

$\{y = z\}\ z := z + 1\ \{y - 1 > z\}$

$\{17 = 18\}\ x := x+1\ \{\ x > z\}$

$\{x = 4\}\ x := 4\ \{x = 4\}$

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

Determine the Weakest Preconditions of the following.

$x := x + 1\ \{x = 12\}$

$x := y + 2\ \{x > 17\}$

$y := 4\ \{x = y + 2\}$

$x := y\ \{x \leq 0\}$

$x := 74\ \{x = 73\}$

What assignments would make the following into valid Hoare triples?

$\{x = y + z\}$	$\{x = z\}$
$\{y * 2 = 12\}$	$\{y = 24\}$
$\{true\}$	$\{y = 12\}$
$\{x > y\}$	$\{x < y\}$
$\{x = X \wedge y = Y\}$	$\{x = Y \wedge y = X\}$

Note, in the last problem x and y are variables and X and Y are the values they contain at the start.