Programming Languages:
Guarded Command Language Summary

Weakest Precondition

The weakest precondition transformer wp satisfies the following rules:

- $wp\ S\ False = False$.
- $wp\ S\ Q \wedge wp\ S\ R = wp\ S\ (Q \wedge R)$.
- $wp\ S\ Q \vee wp\ S\ R \Rightarrow wp\ S\ (Q \vee R)$.

Denote **if** $B_0 \rightarrow S_0 \mid B_1 \rightarrow S_1$ **fi** by IF, **do** $B_0 \rightarrow S_0 \mid B_1 \rightarrow S_1$ **od** by DO, and $B_0 \vee B_1$ by BB.

$$
\begin{array}{ll}
wp\ abort & P = False \\
wp\ skip & P = P \\
wp\ (x := E) & P = P[x \backslash E] \\
wp\ (S; T) & P = wp\ S\ (wp\ T\ P) \\
\multicolumn{2}{l}{wp\ IF\ P} \\
\multicolumn{2}{l}{\quad = (B_0 \Rightarrow wp\ S_0\ P) \wedge (B_1 \Rightarrow wp\ S_1\ P) \wedge BB} \\
\multicolumn{2}{l}{\quad = ((B_0 \wedge wp\ S_0\ P) \vee (B_1 \wedge wp\ S_1\ P)) \wedge BB} \\
\multicolumn{2}{l}{wp\ DO\ P =} \\
\multicolumn{2}{l}{\quad \mu\ (\lambda X \rightarrow wp\ IF\ X \vee (\neg\ BB \wedge P))\ \ ,}
\end{array}
$$

where $\mu\ F$ denotes the strongest X such that $X = F\ X$. The $x := E$ case shall have a side condition that E is defined.

General case: denote by $B_i \rightarrow S_i$ the guarded commands $B_0 \rightarrow S_0 \mid ...B_{n-1} \rightarrow S_{n-1}$.

$$
\begin{array}{l}
wp\ (\mathbf{if}\ B_i \rightarrow S_i\ \mathbf{fi})\ P = \\
\quad \langle \forall i : 0 \leqslant i < n : B_i \Rightarrow wp\ S_i\ P \rangle \wedge \\
\quad \langle \exists i : 0 \leqslant i < n : B_i \rangle\ \ .
\end{array}
$$

When $n = 0$, we have $\mathbf{if\ fi} = abort$. Similarly,

$$
\begin{array}{l}
wp\ (\mathbf{do}\ B_i \rightarrow S_i\ \mathbf{od})\ P = \mu\ (\lambda X \rightarrow \\
\quad wp\ (\mathbf{if}\ B_i \rightarrow S_i\ \mathbf{fi})\ X \vee \\
\qquad (\neg\ \langle \exists i : 0 \leqslant i < n : B_i \rangle \wedge P))\ \ .
\end{array}
$$

When $n = 0$, we have $\mathbf{do\ od} = skip$.

Hoare Logic

Definition: $\{P\}\ S\ \{Q\} \equiv P \Rightarrow wp\ S\ Q$.

The definition entails that

$$
\begin{array}{ll}
\{P\}\ skip\ \{Q\} & \equiv P \Rightarrow Q \\
\{P\}\ x := E\ \{Q\} & \equiv P \Rightarrow Q[x \backslash E] \\
\{P\}\ S; T\ \{Q\} & \equiv \\
\multicolumn{2}{l}{\quad \langle \exists R :: \{P\}\ S\ \{R\} \wedge \{R\}\ T\ \{Q\} \rangle} \\
\{P\}\ IF\ \{Q\} & \equiv \\
\multicolumn{2}{l}{\quad (P \Rightarrow B_0 \vee B_1) \wedge} \\
\multicolumn{2}{l}{\quad \{P \wedge B_0\}\ S_0\ \{Q\} \wedge \{P \wedge B_1\}\ S_1\ \{Q\}}
\end{array}
$$

There is also a side condition that E is defined.

Regarding loops, by the *Fundamental Invariance Theorem*, the weakest precondition of DO entails that $\{P\}\ DO\ \{Q\}$ follows from

1. $P \wedge \neg B_0 \wedge \neg B_1\ \Rightarrow\ Q$,
2. $\{P \wedge B_0\}\ S_0\ \{P\}$ and $\{P \wedge B_1\}\ S_1\ \{P\}$, and
3. there exists an integer function t on the state space such that
   (a) $[P \wedge (B_0 \vee B_1)\ \Rightarrow\ bnd \geqslant 0]$,
   (b) $\{P \wedge B_0 \wedge t = C\}\ S_0\ \{t < C\}$, and
   (c) $\{P \wedge B_1 \wedge t = C\}\ S_1\ \{t < C\}$.

Properties Hoare triples satisfy the following rules:

- $\{P\}\ S\ \{false\}\ \equiv\ \neg P$,
- $\{P\}\ S\ \{Q\}\ \wedge\ (P_0 \Rightarrow P)\ \Rightarrow\ \{P_0\}\ S\ \{Q\}$,
- $\{P\}\ S\ \{Q\}\ \wedge\ (Q \Rightarrow Q_0)\ \Rightarrow\ \{P\}\ S\ \{Q_0\}$,
- $\{P\}\ S\ \{Q\}\ \wedge\ \{P\}\ S\ \{R\}\ \Rightarrow\ \{P\}\ S\ \{Q \wedge R\}$,
- $\{P\}\ S\ \{Q\}\ \wedge\ \{R\}\ S\ \{Q\}\ \Rightarrow\ \{P \vee R\}\ S\ \{Q\}$.

References

[Dij76] E. W. Dijkstra. *A Discipline of Programming*. Prentice Hall, 1976.

[Kal90] A. Kaldewaij. *Programming: the Derivation of Algorithms*. Prentice Hall, 1990.