1 Basic Hoare Logic

We began by reviewing some of the basic rules of Hoare logic:

\[\begin{align*}
\{α\} P{β} & \quad β \Rightarrow γ \quad \text{weaken postcondition} \\
α \Rightarrow β & \quad \{β\} P{γ} \quad \text{strengthen precondition} \\
\{α[e/x]\} x := e{α} & \quad \text{assignment} \\
\{α\} P{β} & \quad \{β\} Q{γ} \quad \text{sequencing} \\
\{α \land c\} P{β} & \quad \{α \land \neg c\} Q{β} \quad \text{if-then-else} \\
\{α\} \text{if } c \text{ then } P \text{ else } Q{β} & \quad \{α\} \text{ if } c \text{ then } P{β} \quad \text{if-then} \\
\{α \land c\} P{α} & \quad \{α \land \neg c\} \quad \text{while}
\end{align*}\]

We then discussed some simple programs, and saw that these could be derived from their specifications.

2 Summing an array

Notation: \([0, n) = \{i \in \mathbb{N} \mid 0 \leq i < n\}\).

We are given an array \(A[0, n)\) with indices from \([0, n)\). Here the specification is:

\[\{\text{true}\} \quad P\{s = \sum_{j \in [0, n)} A[j]\}\]
We guess the loop invariant $s = \sum_{j \in [0,i)} A[j]$, and obtain the following program, annotated with assertions {...} at various points:

{true}
\begin{align*}
s &:= 0; \\
i &:= 0; \\
\{s = \sum_{j \in [0,i)} A[j]\} & \text{(loop invariant)}
\end{align*}

while $i \neq n$
\begin{align*}
s &:= s + A[i]; \\
\{s = \sum_{j \in [0,i+1)} A[j]\} \\
i &:= i+1; \\
\{s = \sum_{j \in [0,i)} A[j]\}
\end{align*}

od
\{s = \sum_{j \in [0,n)} A[j]\}

To verify correctness of these annotations, we need to check the following basic statements:

- {true} $s := 0; i := 0$

  By two applications of the assignment rule and sequencing, it suffices for this that true $\Rightarrow 0 = \sum_{j \in [0,0)} A[j]$. This holds by the convention that an empty sum equals 0.

- {$s = \sum_{j \in [0,i)} A[j] \land i \neq n$} $s := s + A[i]$ \{$s = \sum_{j \in [0,i+1)} A[j]\}$

  We prove this by the assignment rule and the strengthen precondition rule. The assignment rule gives
  \begin{align*}
  \{s + A[i] = \sum_{j \in [0,i+1)} A[j]\} \\
  s &:= s + A[i]; \{s = \sum_{j \in [0,i+1)} A[j]\}
  \end{align*}
  Simple mathematics gives
  \begin{align*}
  (s = \sum_{j \in [0,i)} A[j] \land i \neq n) & \Rightarrow s + A[i] = \sum_{j \in [0,i+1)} A[j] \\
  \text{The conclusion then follows using the strengthen precondition rule.}
  \end{align*}

- {$s = \sum_{j \in [0,i+1)} A[j]$} $i := i+1$ \{$s = \sum_{j \in [0,i)} A[j]\}$

  This is immediate using the assignment rule.

- Note that the previous two steps give, using sequencing, that
  \{s = \sum_{j \in [0,i)} A[j] \land i \neq n\} $s := s + A[i]; i := i+1$ \{$s = \sum_{j \in [0,i)} A[j]\}$

  Using the while rule, we now conclude
  \{s = \sum_{j \in [0,i)} A[j]\} while $i \neq n$ do $s := s + A[i]; i := i+1$ od \{$s = \sum_{j \in [0,i)} A[j] \land i = n$\}

- By simple logic, $s = \sum_{j \in [0,i)} A[j] \land i = n \Rightarrow s = \sum_{j \in [0,n)} A[j]$, so by the weakening the postcondition rule, we have
  \{s = \sum_{j \in [0,i)} A[j]\} while $i \neq n$ do $s := s + A[i]; i := i+1$ od \{$s = \sum_{j \in [0,n)} A[j]\}$
Combining this with the conclusion about the initialization section, using the sequencing rule, we obtain that the program as a whole satisfies its spec.

We discussed in class how this reasoning could be run backwards to derive the program pieces from their local specifications.

We also mentioned a subtlety (are the array accesses always safe?) that is actually a very important question from the point of view of security, but did not carefully develop a proof that shows this. This will be taken up in your mentor meeting.

### 3 Finding the rightmost element

The second example we considered was that of finding the rightmost occurrence of the value 1 in an array. We discussed a program that does this by starting at the right of the array and moves to the left until a 1 is found. We started to develop a specification and proof, but did not complete this. You will finish this example in Assignment 1b!