The Assignment Axiom (Hoare)

- **Syntax:** V `:=` E
- **Semantics: value of V in final state is value of E in initial state**
- **Example:** `X:=X+1` **(adds one to the value of the variable** `X`**)**

The Assignment Axiom

$$\vdash \ \{Q[E/V]\}\ V := E\ \{Q\}$$

Where V is any variable, E is any expression, Q is any statement.

- **Instances of the assignment axiom are**
  - $\vdash\ \{\mathtt{E} = \mathtt{x}\}\ \mathtt{V} := \mathtt{E}\ \{\mathtt{V} = \mathtt{x}\}$
  - $\vdash\ \{\mathtt{Y} = 2\}\ \mathtt{X} := 2\ \{\mathtt{Y} = \mathtt{X}\}$
  - $\vdash\ \{\mathtt{X} + 1 = \mathtt{n} + 1\}\ \mathtt{X} := \mathtt{X} + 1\ \{\mathtt{X} = \mathtt{n} + 1\}$
  - $\vdash\ \{E = E\}\ \mathtt{X} := E\ \{\mathtt{X} = E\}$ **(if** `X` **does not occur in** E**)**

Precondition Strengthening

- **Recall that**

$$\frac{\vdash\ S_1,\ \ldots\ ,\ \vdash\ S_n}{\vdash\ S}$$

means $\vdash\ S$ **can be deduced from** $\vdash\ S_1,\ \ldots\ ,\ \vdash\ S_n$

- **Using this notation, the rule of precondition strengthening is**

Precondition strengthening

$$\frac{\vdash\ P \Rightarrow P', \qquad \vdash\ \{P'\}\ C\ \{Q\}}{\vdash\ \{P\}\ C\ \{Q\}}$$

- **Note the two hypotheses are different kinds of judgements**

Postcondition weakening ✓

- **Just as the previous rule allows the precondition of a partial correctness specification to be strengthened, the following one allows us to weaken the postcondition**

Postcondition weakening

$$\frac{\vdash\ \{P\}\ C\ \{Q'\}, \qquad \vdash\ Q' \Rightarrow Q}{\vdash\ \{P\}\ C\ \{Q\}}$$

An Example Formal Proof ✓

- **Here is a little formal proof**

1.	$\vdash$	$\{\mathtt{R{=}X} \wedge \mathtt{0{=}0}\}\ \mathtt{Q{:=}0}\ \{\mathtt{R{=}X} \wedge \mathtt{Q{=}0}\}$	By the assignment axiom
2.	$\vdash$	$\mathtt{R{=}X} \Rightarrow \mathtt{R{=}X} \wedge \mathtt{0{=}0}$	By pure logic
3.	$\vdash$	$\{\mathtt{R{=}X}\}\ \mathtt{Q{:=}0}\ \{\mathtt{R{=}X} \wedge \mathtt{Q{=}0}\}$	By precondition strengthening
4.	$\vdash$	$\mathtt{R{=}X} \wedge \mathtt{Q{=}0} \Rightarrow \mathtt{R{=}X{+}(Y} \times \mathtt{Q)}$	By laws of arithmetic
5.	$\vdash$	$\{\mathtt{R{=}X}\}\ \mathtt{Q{:=}0}\ \{\mathtt{R{=}X{+}(Y} \times \mathtt{Q)}\}$	By postcondition weakening

- **The rules precondition strengthening and postcondition weakening are sometimes called the** *rules of consequence*

The sequencing rule ✓

- **Syntax:** $C_1;\ \cdots\ ;C_n$

- **Semantics: the commands $C_1, \cdots, C_n$ are executed in that order**

- **Example:** `R:=X; X:=Y; Y:=R`

  - **the values of `X` and `Y` are swapped using `R` as a temporary variable**

  - **note *side effect*: value of `R` changed to the old value of `X`**

The sequencing rule

$$\frac{\vdash\ \{P\}\ C_1\ \{Q\}, \qquad \vdash\ \{Q\}\ C_2\ \{R\}}{\vdash\ \{P\}\ C_1;C_2\ \{R\}}$$

Example Proof

Example: By the assignment axiom:

(i) $\vdash$ $\{\texttt{X=x}\wedge\texttt{Y=y}\}$ `R:=X` $\{\texttt{R=x}\wedge\texttt{Y=y}\}$

(ii) $\vdash$ $\{\texttt{R=x}\wedge\texttt{Y=y}\}$ `X:=Y` $\{\texttt{R=x}\wedge\texttt{X=y}\}$

(iii) $\vdash$ $\{\texttt{R=x}\wedge\texttt{X=y}\}$ `Y:=R` $\{\texttt{Y=x}\wedge\texttt{X=y}\}$

Hence by (i), (ii) and the sequencing rule

(iv) $\vdash$ $\{\texttt{X=x}\wedge\texttt{Y=y}\}$ `R:=X; X:=Y` $\{\texttt{R=x}\wedge\texttt{X=y}\}$

Hence by (iv) and (iii) and the sequencing rule

(v) $\vdash$ $\{\texttt{X=x}\wedge\texttt{Y=y}\}$ `R:=X; X:=Y; Y:=R` $\{\texttt{Y=x}\wedge\texttt{X=y}\}$

Conditionals

✓

- **Syntax:** IF S THEN C_1 ELSE C_2

- **Semantics:**
  - **if the statement S is true in the current state, then C_1 is executed**
  - **if S is false, then C_2 is executed**

- **Example:** `IF X<Y THEN MAX:=Y ELSE MAX:=X`
  - **the value of the variable `MAX` it set to the maximum of the values of `X` and `Y`**

The Conditional Rule ✓

The conditional rule

$$\frac{\vdash\ \{P \wedge S\}\ C_1\ \{Q\}, \qquad \vdash\ \{P \wedge \neg S\}\ C_2\ \{Q\}}{\vdash\ \{P\}\ \texttt{IF}\ S\ \texttt{THEN}\ C_1\ \texttt{ELSE}\ C_2\ \{Q\}}$$

- **From Assignment Axiom + Precondition Strengthening and**

  $\vdash\ (\texttt{X}\geq\texttt{Y} \Rightarrow \texttt{X = max(X,Y)}) \wedge (\neg(\texttt{X}\geq\texttt{Y}) \Rightarrow \texttt{Y = max(X,Y)})$

  it follows that

  $\vdash\ \{\texttt{T} \wedge \texttt{X}\geq\texttt{Y}\}\ \texttt{MAX:=X}\ \{\texttt{MAX=max(X,Y)}\}$

  and

  $\vdash\ \{\texttt{T} \wedge \neg(\texttt{X}\geq\texttt{Y})\}\ \texttt{MAX:=Y}\ \{\texttt{MAX=max(X,Y)}\}$

- **Then by the conditional rule it follows that**

  $\vdash\ \{\texttt{T}\}\ \texttt{IF X}\geq\texttt{Y THEN MAX:=X ELSE MAX:=Y}\ \{\texttt{MAX=max(X,Y)}\}$

`WHILE`-commands ✓

- **Syntax:** `WHILE` S `DO` C

- **Semantics:**

  - **if the statement S is true in the current state, then C is executed and the `WHILE`-command is repeated**

  - **if S is false, then nothing is done**

  - **thus C is repeatedly executed until the value of S becomes false**

  - **if S never becomes false, then the execution of the command never terminates**

- **Example:** `WHILE ¬(X=0) DO X:= X-2`

  - **if the value of `X` is non-zero, then its value is decreased by `2` and then the process is repeated**

- **This `WHILE`-command will terminate (with `X` having value `0`) if the value of `X` is an even non-negative number**

  - **in all other states it will not terminate**

Invariants ✓

- **Suppose** $\vdash \{P \wedge S\}\ C\ \{P\}$

- P **is said to be an** *invariant* **of** C **whenever** S **holds**

- **The** `WHILE`**-rule says that**
  - **if** P **is an invariant of the body of a** `WHILE`**-command whenever the test condition holds**
  - **then** P **is an invariant of the whole** `WHILE`**-command**

- **In other words**
  - **if executing** C *once* **preserves the truth of** P
  - **then executing** C *any number of times* **also preserves the truth of** P

- **The** `WHILE`**-rule also expresses the fact that after a** `WHILE`**-command has terminated, the test must be false**
  - **otherwise, it wouldn't have terminated**

The WHILE-Rule ✓

The WHILE-rule

$$\frac{\vdash\ \{P \wedge S\}\ C\ \{P\}}{\vdash\ \{P\}\ \mathtt{WHILE}\ S\ \mathtt{DO}\ C\ \{P \wedge \neg S\}}$$

- **It is easy to show**

$\vdash\ \{\mathtt{X=R+(Y}\times\mathtt{Q)}\wedge\mathtt{Y}\leq\mathtt{R}\}\ \mathtt{R:=R-Y;\ Q:=Q+1}\ \{\mathtt{X=R+(Y}\times\mathtt{Q)}\}$

- **Hence by the WHILE-rule with** P = 'X=R+(Y×Q)' **and** S = 'Y≤R'

$\vdash\ \{\mathtt{X=R+(Y}\times\mathtt{Q)}\}$
$\quad\mathtt{WHILE\ Y}\leq\mathtt{R\ DO}$
$\qquad\mathtt{(R:=R-Y;\ Q:=Q+1)}$
$\quad\{\mathtt{X=R+(Y}\times\mathtt{Q)}\ \wedge\ \neg\mathtt{(Y}\leq\mathtt{R)}\}$

Example

- **From the previous slide**

$$\vdash \ \{\texttt{X=R+(Y}\times\texttt{Q)}\}$$

```
WHILE Y≤R DO
  (R:=R-Y; Q:=Q+1)
```

$$\{\texttt{X=R+(Y}\times\texttt{Q)} \wedge \neg(\texttt{Y}\leq\texttt{R})\}$$

- **It is easy to deduce that**

$$\vdash \ \{\texttt{T}\}\ \texttt{R:=X; Q:=0}\ \{\texttt{X=R+(Y}\times\texttt{Q)}\}$$

- **Hence by the sequencing rule and postcondition weakening**

$$\vdash \ \{\texttt{T}\}$$

```
R:=X;
Q:=0;
WHILE Y≤R DO
  (R:=R-Y; Q:=Q+1)
```

$$\{\texttt{R<Y} \wedge \texttt{X=R+(Y}\times\texttt{Q)}\}$$

Summary ✓

- **We have given:**
  - a notation for specifying what a program does
  - a way of proving that it meets its specification
- **Now we look at ways of finding proofs and organising them:**
  - finding invariants
  - derived rules
  - backwards proofs
  - annotating programs prior to proof
- **Then we see how to automate program verification**
  - the automation mechanises some of these ideas

How does one find an invariant? ✓

The WHILE-rule

$$\frac{\vdash \ \{P \wedge S\}\ C\ \{P\}}{\vdash \ \{P\}\ \texttt{WHILE}\ S\ \texttt{DO}\ C\ \{P \wedge \neg S\}}$$

- **Look at the facts:**
  - **invariant P must hold initially**
  - **with the negated test $\neg S$ the invariant P must establish the result**
  - **when the test S holds, the body must leave the invariant P unchanged**

- **Think about how the loop works – the invariant should say that:**
  - **what has been done so far together with what remains to be done**
  - **holds at each iteration of the loop**
  - **and gives the desired result when the loop terminates**

Example

✓

- **Consider a factorial program**

```
{X=n ∧ Y=1}
 WHILE X≠0 DO
   (Y:=Y×X; X:=X-1)
{X=0 ∧ Y=n!}
```

- **Look at the facts**
  - initially `X=n` and `Y=1`
  - finally `X=0` and `Y=n!`
  - on each loop `Y` is increased and, `X` is decreased

- **Think how the loop works**
  - `Y` holds the result so far
  - `X!` is what remains to be computed
  - `n!` is the desired result

- **The invariant is `X!×Y = n!`**
  - 'stuff to be done' × 'result so far' = 'desired result'
  - decrease in `X` combines with increase in `Y` to make invariant

Related example ✓

$\{$X=0 $\wedge$ Y=1$\}$
 WHILE X<N DO (X:=X+1; Y:=Y$\times$X)
$\{$Y=N!$\}$

- **Look at the Facts**
  - **initially** X=0 **and** Y=1
  - **finally** X=N **and** Y=N!
  - **on each iteration both** X **an** Y **increase:** X **by** 1 **and** Y **by** X

- **An invariant is** Y = X!
- **At end need** Y = N!, **but** WHILE**-rule only gives** $\neg$(X<N)
- **Ah Ha! Invariant needed:** $\boxed{\texttt{Y = X!} \ \wedge \ \texttt{X} \leq \texttt{N}}$
- **At end** X $\leq$ N $\wedge\neg$(X<N) $\Rightarrow$ X=N
- **Often need to strenthen invariants to get them to work**
  - **typical to add stuff to 'carry along' like** X$\leq$N

Conjunction and Disjunction

Specification conjunction

$$\frac{\vdash\ \{P_1\}\ C\ \{Q_1\}, \qquad \vdash\ \{P_2\}\ C\ \{Q_2\}}{\vdash\ \{P_1 \wedge P_2\}\ C\ \{Q_1 \wedge Q_2\}}$$

Specification disjunction

$$\frac{\vdash\ \{P_1\}\ C\ \{Q_1\}, \qquad \vdash\ \{P_2\}\ C\ \{Q_2\}}{\vdash\ \{P_1 \vee P_2\}\ C\ \{Q_1 \vee Q_2\}}$$

- **These rules are useful for splitting a proof into independent bits**
  - **they enable** $\vdash\ \{\mathtt{P}\}\ \mathtt{C}\ \{\mathtt{Q}_1 \wedge \mathtt{Q}_2\}$ **to be proved by proving separately that both** $\vdash\ \{\mathtt{P}\}\ \mathtt{C}\ \{\mathtt{Q}_1\}$ **and also that** $\vdash\ \{\mathtt{P}\}\ \mathtt{C}\ \{\mathtt{Q}_2\}$

- **Any proof with these rules could be done without using them**
  - **i.e. they are theoretically redundant (proof omitted)**
  - **however, useful in practice**

Derived rules for finding proofs ✓

- **Suppose the goal is to prove** $\{\mathit{Precondition}\}\ \mathit{Command}\ \{\mathit{Postcondition}\}$

- **If there were a rule of the form**

$$\frac{\vdash\ H_1,\ \cdots,\vdash\ H_n}{\vdash\ \{P\}\ C\ \{Q\}}$$

  then we could instantiate

  $P \mapsto \mathit{Precondition}$, $C \mapsto \mathit{Command}$, $Q \mapsto \mathit{Postcondition}$

  to get instances of $H_1,\ \cdots, H_n$ **as subgoals**

- **Some of the rules are already in this form e.g. the sequencing rule**

- **We will derive rules of this form for all commands**

- **Then we use these derived rules for mechanising Hoare Logic proofs**

Derived Rules ✓

- **We will establish derived rules for all commands**

$$\frac{\cdots}{\vdash\ \{P\}\ V\,\texttt{:=}E\ \{Q\}}$$

$$\frac{\cdots}{\vdash\ \{P\}\ C_1\,\texttt{;}C_2\ \{Q\}}$$

$$\frac{\cdots}{\vdash\ \{P\}\ \texttt{IF}\ S\ \texttt{THEN}\ C_1\ \texttt{ELSE}\ C_2\ \{Q\}}$$

$$\frac{\cdots}{\vdash\ \{P\}\ \texttt{WHILE}\ S\ \texttt{DO}\ C\ \{Q\}}$$

- **These support 'backwards proof' starting from a goal** $\{P\}\ C\ \{Q\}$

The Derived Assignment Rule ✓

- **An example proof**

1. $\vdash\ \{\mathtt{R=X} \wedge \mathtt{0=0}\}\ \mathtt{Q:=0}\ \{\mathtt{R=X} \wedge \mathtt{Q=0}\}$ By the assignment axiom.
2. $\vdash\ \mathtt{R=X}\ \Rightarrow\ \mathtt{R=X} \wedge \mathtt{0=0}$ By pure logic.
3. $\vdash\ \{\mathtt{R=X}\}\ \mathtt{Q:=0}\ \{\mathtt{R=X} \wedge \mathtt{Q=0}\}$ By precondition strengthening.

- **Can generalise this proof to a proof schema:**

1. $\vdash\ \{Q[E/V]\}\ V:=E\ \{Q\}$ By the assignment axiom.
2. $\vdash\ P\ \Rightarrow\ Q[E/V]$ By assumption.
3. $\vdash\ \{P\}\ V:=E\ \{Q\}$ By precondition strengthening.

- **This proof schema justifies:**

Derived Assignment Rule

$$\frac{\vdash\ P \Rightarrow Q[E/V]}{\vdash\ \{P\}\ V:=E\ \{Q\}}$$

- **Note: $Q[E/V]$ is the weakest liberal precondition $wlp(V:=E, Q)$**

- **Example proof above can now be done in one less step**

1. $\vdash\ \mathtt{R=X}\ \Rightarrow\ \mathtt{R=X} \wedge \mathtt{0=0}$ By pure logic.
2. $\vdash\ \{\mathtt{R=X}\}\ \mathtt{Q:=0}\ \{\mathtt{R=X} \wedge \mathtt{Q=0}\}$ By derived assignment.

Derived Sequenced Assignment Rule

- **The following rule will be useful later**

Derived Sequenced Assignment Rule

$$\frac{\vdash\ \{P\}\ C\ \{Q[E/V]\}}{\vdash\ \{P\}\ C;V:=E\ \{Q\}}$$

- **Intuitively work backwards:**
  - **push** Q **'through'** $V:=E$**, changing it to** $Q[E/V]$

- **Example: By the assignment axiom:**

$\vdash$ {X=x∧Y=y} R:=X {R=x∧Y=y}

- **Hence by the sequenced assignment rule**

$\vdash$ {X=x∧Y=y} R:=X; X:=Y {R=x∧X=y}

The Derived Sequencing Rule

- **The rule below follows from the sequencing and consequence rules**

The Derived Sequencing Rule

$$\frac{\begin{array}{ll} & \vdash\ P \Rightarrow P_1 \\ \vdash\ \{P_1\}\ C_1\ \{Q_1\} & \vdash\ Q_1 \Rightarrow P_2 \\ \vdash\ \{P_2\}\ C_2\ \{Q_2\} & \vdash\ Q_2 \Rightarrow P_3 \\ . & . \\ . & . \\ . & . \\ \vdash\ \{P_n\}\ C_n\ \{Q_n\} & \vdash\ Q_n \Rightarrow Q \end{array}}{\vdash\ \{P\}\ C_1;\ \ldots\ ;\ C_n\ \{Q\}}$$

- **Exercise: why no derived conditional rule?**

The Derived While Rule ✓

Derived While Rule

$$\frac{\vdash\ P \Rightarrow R \qquad \vdash\ \{R\ \wedge\ S\}\ C\ \{R\} \qquad \vdash\ R\wedge\ \neg S\ \Rightarrow Q}{\vdash\ \{P\}\ \texttt{WHILE}\ S\ \texttt{DO}\ C\ \{Q\}}$$

- **This follows from the While Rule and the rules of consequence**

- **Example: it is easy to show**

$\vdash\ \texttt{R=X}\ \wedge\ \texttt{Q=0}\ \Rightarrow\ \texttt{X=R+(Y}\times\texttt{Q)}$

$\vdash\ \{\texttt{X=R+(Y}\times\texttt{Q)}\wedge\texttt{Y}\leq\texttt{R}\}\ \texttt{R:=R-Y; Q:=Q+1}\ \{\texttt{X=R+(Y}\times\texttt{Q)}\}$

$\vdash\ \texttt{X=R+(Y}\times\texttt{Q)}\wedge\neg\texttt{(Y}\leq\texttt{R)}\ \Rightarrow\ \texttt{X=R+(Y}\times\texttt{Q)}\wedge\neg\texttt{(Y}\leq\texttt{R)}$

- **Then, by the derived While rule**

$\vdash\ \{\texttt{R=X}\ \wedge\ \texttt{Q=0}\}$
$\quad\texttt{WHILE Y}\leq\texttt{R DO}$
$\quad\quad\texttt{(R:=R-Y; Q:=Q+1)}$
$\{\texttt{X=R+(Y}\times\texttt{Q)}\ \wedge\ \neg\texttt{(Y}\leq\texttt{R)}\}$

Forwards and backwards proof

- **Previously it was shown how to prove $\{P\}C\{Q\}$ by**
  - **proving properties of the components of C**
  - **and then putting these together, with the appropriate proof rule, to get the desired property of C**
- **For example, to prove $\vdash \{P\}C_1;C_2\{Q\}$**
- **First prove $\vdash \{P\}C_1\{R\}$ and $\vdash \{R\}C_2\{Q\}$**
- **then deduce $\vdash \{P\}C_1;C_2\{Q\}$ by sequencing rule**
- **This method is called** *forward proof*
  - **move forward from axioms via rules to conclusion**
- **The problem with forwards proof is that it is not always easy to see what you need to prove to get where you want to be**
- **It is more natural to work backwards**
  - **starting from the goal of showing $\{P\}C\{Q\}$**
  - **generate subgoals until problem solved**