Provability of Triples

- Remember that we want to use valid correctness triples to show a program works as expected. In other words, given a program S, given a precondition p that can be provided to this program before it executes, and given a postcondition q that we expect to get after this program executes, we need to show this correctness triple $\{p\}\ S\ \{q\}$ is valid; denoted as $\vDash \{p\}\ S\ \{q\}$ or $\vDash_{tot} \{p\}\ S\ \{q\}$ (depend on what correctness level we need).

- In fact, not all triples can be decided to be valid or invalid. This is like not everything true can be proved to be true, or not all yes-or-no problems have an algorithm that can guarantee a solution (This is taught in $CS530$).

- If a triple $\{p\}\ S\ \{q\}$ can be proved to be valid, then we say this triple is **provable**, denoted as $\vdash \{p\}\ S\ \{q\}$ or $\vdash_{tot} \{p\}\ S\ \{q\}$ (depend on what correctness level we need).
  - In this course we care about provable triples. We focus on creating proofs for provable triples; we don't focus on deciding whether a valid triple is provable or not.

- To prove a triple (with large body) being valid, we will create a **proof system *of triples***, which is a set of logical formulas determined by a set of ***axioms*** and ***rules of inference*** using a set of syntactic algorithms.
  - Each true statement in a proof system is called a **judgement**, in a proof system of triples, each judgement is a provable triple or a valid predicate.
  - In math and logic, an **axiom** is something accepted to be true that is unprovable. In a proof system, some axioms can tell us some judgements are true. For example: $\{p\}\ \textbf{skip}\ \{p\}$ is an axiom, $x + 0 = x$ is also an axiom.
  - **Rules of Inference** are a set of rules that can combine several truths into a more complicated truth. In a proof system, rules of Inference can combine several judgements into one larger judgement. For example, modus ponens is an inference rule: $p \wedge (p \to q) \Rightarrow q$.
    Another example, Conditional Rule 1 is an inference rule:
    $$\{p \wedge B\}\ S_1\{q_1\} \wedge \{p \wedge \neg B\}\ S_2\ \{q_2\} \Rightarrow \{p\}\ \textbf{if}\ B\ \textbf{then}\ S_1\ \textbf{else}\ S_2\ \textbf{fi}\ \{q_1 \vee q_2\}.$$

Proof Formats

- Here is an example of a proof system with Conditional Rule 1:

$$\frac{\{p \wedge B\}\ S_1\{q_1\} \qquad\qquad \{p \wedge \neg B\}\ S_2\ \{q_2\}}{\{p\}\ \textbf{if}\ B\ \textbf{then}\ S_1\ \textbf{else}\ S_2\ \textbf{fi}\ \{q_1 \vee q_2\}}\ \text{if} - \text{else}$$

  - The above format is called a **proof tree**. The two child judgements (**antecedents**) are above a straight line, and they together logically imply the parent judgement (**consequent**). The **rule name** is attached to the straight line.
  - The advantage of proof trees is that you can read them easily. The disadvantage is that it is hard to draw since it can be wide.

- In this class, we use **Hilbert-style proofs**. Here is Conditional Rule 1 in Hilbert-style:
  1. $\{p \wedge B\}\ S_1\{q_1\}$
  2. $\{p \wedge \neg B\}\ S_2\ \{q_2\}$
  3. $\{p\}\ \textbf{if}\ B\ \textbf{then}\ S_1\ \textbf{else}\ S_2\ \textbf{fi}\ \{q_1 \vee q_2\}$ $\qquad\qquad$ if − else 1,2

- A Hilbert-style proof has two columns. On the left, we write judgements: antecedents must appear above the consequent (not necessarily immediately above). On the right, we write the axiom we use or the rule names together with line numbers of its antecedents involved.

Some Axioms and Rules of Inference

We have seen most of these axioms and rules already.

You may assume that so far we are creating proofs under **partial correctness**, but most of the axioms and inference rules we introduce here also work for total correctness.

- **Skip Axiom**:
  1. $\{p\}$ **skip** $\{p\}$ skip

- **Backward Assignment Axiom**:
  1. $\{p[e \,/\, v]\}\ v \coloneqq e\ \{p\}$ backward assignment

- **Forward Assignment Axiom**:
  1. $\{p\}\ v \coloneqq e\ \{p[v_0 \,/\, v]\ \wedge v = e[v_0 \,/\, v]\}$ forward assignment

- **Strengthen Precondition Rule:**
  1. $p \Rightarrow p_1$ (or $p \rightarrow p_1$)
  2. $\{p_1\}\ S\ \{q\}$
  3. $\{p\}\ S\ \{q\}$ strengthen precondition 1,2

  - Each judgment in a proof is true/valid, so we can use $p \rightarrow p_1$ instead of $p \Rightarrow p_1$.

- **Weaken Postcondition Rule:**
  1. $\{p\}\ S\ \{q_1\}$
  2. $q_1 \Rightarrow q$ (or $q_1 \rightarrow q$)
  3. $\{p\}\ S\ \{q\}$ weaken postcondition 1,2

- **Sequence Rule**:
  1. $\{p\}\ S_1\ \{r\}$
  2. $\{r\}\ S_2\ \{q\}$
  3. $\{p\}\ S_1; S_2\ \{q\}$ sequence 1,2

  - Note that we previously defined a more general version of the sequence rule; we only need the postcondition in line 1 to be stronger than the precondition in line 2. Here, we define the extended sequence rule.

- Extended Sequence Rule:
  1. $\{p_1\}\ S_1\ \{q_1\}$
  2. $q_1 \Rightarrow p_2$ (or $q_1 \rightarrow p_2$)
  3. $\{p_2\}\ S_2\ \{q_2\}$
  4. $\{p_1\}\ S_1; S_2 \{q_2\}$ extended sequence 1,2,3

- **Conjunction Rule:**

1. $\{p_1\}\ S\ \{q_1\}$
2. $\{p_2\}\ S\ \{q_2\}$
3. $\{p_1 \wedge p_2\}\ S\ \{q_1 \wedge q_2\}$ conjunction 1,2

- Disjunction Rule:
  1. $\{p_1\}\ S\ \{q_1\}$
  2. $\{p_2\}\ S\ \{q_2\}$
  3. $\{p_1 \vee p_2\}\ S\ \{q_1 \vee q_2\}$ disjunction 1,2

1. We know that $\{T\}\ x \coloneqq 1; k \coloneqq e\ \{x = 2^k\}$ is provable. Show a proof of the triple and what can be used for e?
   1. $\{x = 2^e\}\ k \coloneqq e\ \{x = 2^k\}$ backward assignment
   2. $\{1 = 2^e\}\ x \coloneqq 1\ \{x = 2^e\}$ backward assignment
   3. $\{1 = 2^e\}\ x \coloneqq 1; k \coloneqq e\ \{x = 2^k\}$ sequence 2,1

      # We need $T \to 1 = 2^e$ so $e = 0$

   4. $T \to 1 = 2^0$ predicate logic
   5. $\{T\}\ x \coloneqq 1; k \coloneqq e\ \{x = 2^k\}$ strengthen precondition 4,3

2. Recreate the above proof (with $e = 0$) but use forward assignments instead of backward assignments.
   1. $\{T\}\ x \coloneqq 1\ \{x = 1\}$ forward assignment
   2. $\{x = 1\}\ k \coloneqq 0\ \{x = 1 \wedge k = 0\}$ forward assignment
   3. $\{T\}\ x \coloneqq 1; k \coloneqq 0\ \{x = 1 \wedge k = 0\}$ sequence 1,2
   4. $x = 1 \wedge k = 0 \to x = 2^k$ predicate logic
   5. $\{T\}\ x \coloneqq 1; k \coloneqq 0\ \{x = 2^k\}$ weaken postcondition 3,4

- Conditional Rule 1:
  1. $\{p \wedge B\}\ S_1\ \{q_1\}$
  2. $\{p \wedge \neg B\}\ S_2\ \{q_2\}$
  3. $\{p\}$ **if** B **then** S_1 **else** S_2 **fi** $\{q_1 \vee q_2\}$ if $-$ else 1,2

- Conditional Rule 2:
  1. $\{p_1\}\ S_1\ \{q_1\}$
  2. $\{p_2\}\ S_2\ \{q_2\}$
  3. $\{(B \to p_1) \wedge (\neg B \to p_2)\}$ **if** B **then** S_1 **else** S_2 **fi** $\{q_1 \vee q_2\}$ if $-$ else 1,2

3. Prove conditional rule 2 using conditional rule 1.
   1. $\{p_1\}\ S_1\ \{q_1\}$ premise (assumption)
   2. $(B \to p_1) \wedge B \Rightarrow p_1$ modus ponens
   3. $p_0 \wedge B \Rightarrow (B \to p_1) \wedge B$ predicate logic (or $-$ introduction)

      # Where $p_0 \equiv (B \to p_1) \wedge (\neg B \to p_2)$

   4. $p_0 \wedge B \Rightarrow p_1$ predicate logic
   5. $\{p_0 \wedge B\}\ S_1\ \{q_1\}$ strengthen precondtion 4, 1
   6. $\{p_2\}\ S_2\ \{q_2\}$ premise
   7. $p_0 \wedge \neg B \Rightarrow p_2$ predicate logic
   8. $\{p_0 \wedge \neg B\}\ S_2\ \{q_2\}$ strengthen precondtion 7, 6
   9. $\{p_0\}$ **if** B **then** S_1 **else** S_2 **fi** $\{q_1 \vee q_2\}$ if $-$ else 5,8

4. Use conditional rule 1 to create a proof for an **if − then** statement.
   1. $\{p \wedge B\}\, S_1\, \{q_1\}$
   2. $\{p \wedge \neg B\}$ **skip** $\{p \wedge \neg B\}$
   3. $\{p\}$ **if** B **then** S_1 **fi** $\{q_1 \vee p \wedge \neg B\}$ — if − else 1,2

- Nondeterministic Conditional Rule:
   1. $\{p \wedge B_1\}\, S_1\, \{q_1\}$
   2. $\{p \wedge B_2\}\, S_2\, \{q_2\}$
   3. $\{p\}$ **if** $B_1 \rightarrow S_1 \,\square\, B_2 \rightarrow S_2$ **fi** $\{q_1 \vee q_2\}$ — if − fi 1,2

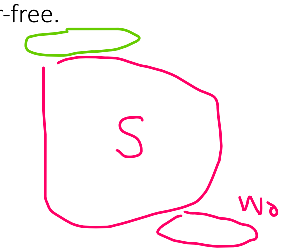
5. Find a p such that $\{p\}\, S\, \{l < r\}$ is provable under partial correctness where $S \equiv$ **if** $b[m] < x \rightarrow l := m \,\square\, b[m] > x \rightarrow r := m$ **fi**.
   - We can calculate p such that $p \Leftrightarrow wlp(S, l < r)$ then try to prove the triple.
     $$\begin{aligned} wlp(S, l < r) &\equiv \big(b[m] < x \rightarrow wlp(l := m,\ l < r)\big) \wedge \big(b[m] > x \rightarrow wlp(r := m,\ l < r)\big) \\ &\equiv (b[m] < x \rightarrow m < r) \wedge (b[m] > x \rightarrow l < m) \end{aligned}$$

   - Now let us use the above expression as p then try to prove the triple $\{p\}\, S\, \{l < r\}$. We want to utilize the nondeterministic conditional rule.

     1. $\{m < r\}\, l := m\, \{l < r\}$ — backfward assignment
     2. $(b[m] < x \rightarrow m < r) \wedge b[m] < x \Rightarrow m < r$ — modus ponens
     3. $p \wedge b[m] < x \Rightarrow (b[m] < x \rightarrow m < r) \wedge b[m] < x$ — predicate logic
        # Where $p \equiv (b[m] < x \rightarrow m < r) \wedge (b[m] > x \rightarrow l < m)$
     4. $p \wedge b[m] < x \Rightarrow m < r$ — predicate logic
     5. $\{p \wedge b[m] < x\}\, l := m\, \{l < r\}$ — strengthen precondition 4,1
     6. $\{l < m\}\, r := m\, \{l < r\}$ — backfward assignment
     7. $p \wedge b[m] > x \Rightarrow l < m$ — predicate logic
     8. $\{p \wedge b[m] > x\}\, r := m\, \{l < r\}$ — strengthen precondtion 7,6
     9. $\{p\}\, S\, \{l < r\}$ — if − fi 5,8

<u>Loop Invariant and While Loop Rule</u>

- What should be the $wp(W, q)$ for $W \equiv$ **while** B **do** S **od**? Here let's assume that W is error-free.
   - Denote w_i as the weakest precondition where loop W runs exactly i iterations.
     - If we never enter the loop body, then $wp(W, q) = w_0 = \neg B \wedge q$
     - If W runs exactly one iteration, then $wp(W, q) = w_1 = B \wedge wp(S, w_0)$
     - If W runs exactly two iterations, then $wp(W, q) = w_2 = B \wedge wp(S, w_1)$
     - …
     - If W runs exactly $k > 0$ iterations, then $wp(W, q) = w_k = B \wedge wp(S, w_{k-1})$



   - In general, we cannot predict how many iterations are needed for a loop, thus $wp(W, q) \equiv w_0 \vee w_1 \vee w_2 \ldots \vee w_k$ where k is the is the number of iterations W being executed, and k can be arbitrarily large. In other words, .
   - The same problem also happens when we calculate $sp(p, W)$.