Minimal Proof Outlines and Partial Proof Outlines

- In the previous class, we learned how to get a full proof outline from a formal proof: in a program, we wrap each statement with the triple used in the formal proof for this program.
  - For the sake of completeness, we do not have a good way to show conjunction/disjunction rule in a proof outline, since we need to wrap the same statement in different triples.

- In a **minimal proof outline**, we have the minimum amount of program annotation that allows us to infer the rest of the formal proof outline: in general, we can't infer the *initial precondition* and *initial postcondition*, nor can we infer the *invariants of loops*, so a minimal outline will include those conditions and no others.
- A **partial proof outline** is somewhere in the middle: More filled-in than a minimal outline but not completely full.

1. In Lecture 15, we have seen the following full proof outline:

   $\{n \geq 0\}$
   $k \coloneqq 0; \{n \geq 0 \wedge k = 0\}\ s \coloneqq 0;\ \{n \geq 0 \wedge k = 0 \wedge s = 0\}$
   $\{\mathbf{inv}\ p \equiv 0 \leq k \leq n \wedge s = sum(0\ , k)\}$
   while $k < n$ **do**
   $\{p \wedge\ k < n\}$
   $\{p[k+1\ /\ k][s+k+1\ /\ s]\}\ s \coloneqq s + k + 1; \{p[k+1\ /\ k]\}\ k \coloneqq k + 1\ \{p\}$
   od
   $\{p \wedge k \geq n\}$
   $\{s = sum\ (0, n)\}$

   What is the minimal proof outline of the above statement? (Use red color)

2. Consider the following full proof outlines and their minimal proof outlines:
   a. $\{T\}\ k \coloneqq 0;\ \{k = 0\}\ x \coloneqq 1\ \{x = 1 \wedge k = 0\}\{k \geq 0 \wedge x = 2^k\}$
   b. $\{T\}\{0 \geq 0 \wedge 1 = 2^0\}\ k \coloneqq 0; \{k \geq 0 \wedge 1 = 2^k\}\ x \coloneqq 1\ \{k \geq 0 \wedge x = 2^k\}$

   - These full proof outlines have the same minimal proof outlines: $\{T\}\ k \coloneqq 0\ x \coloneqq 1\{k \geq 0 \wedge x = 2^k\}$. The reason multiple full proof outlines can have the same minimal outline is because different organizations of backward assignment and forward assignment can have the same minimal outline. There can also be differences in whether and where preconditions are strengthened or postconditions are weakened.

Expanding Minimal/Partial Proof Outlines to Full Proof Outlines

- Remember that, in a full proof outline, every statement needs to be wrapped in a probable triple. To expand a minimal/partial proof outline into a full proof outline, basically we need to infer all the missing predicates. Postconditions are inferred from preconditions using $sp(\dots)$, and preconditions are inferred from postconditions using $wlp\ (\dots)$. Loop invariants tell us how to annotate the loop body and postcondition.

- Expanding a minimal/partial outline can lead to several different full outlines (because multiple full outlines can have the same minimal outline), but all the full outlines will be correct, and the differences between them are generally stylistic.

3. Expand the following minimal proof outline to a full proof outline. Here $abs(x)$ gives the absolute value of x.

$$\{y = x\}\ \textbf{if}\ x < 0\ \textbf{then}\ y := -x\ \textbf{fi}\ \{y = abs\ (x)\}$$

   a. First, let's use wlp on each branch then use Conditional Rule 1 for the whole conditional statement. Then we have:

$\{y = x\}$
if $x < 0$ **then**
    $\{y = x \wedge x < 0\}\ \{-x = abs\ (x)\}\ y := -x\ \{y = abs\ (x)\}$
else
    $\{y = x \wedge x \geq 0\}\ \{y = abs\ (x)\}$ **skip** $\{y = abs\ (x)\}$
fi $\{y = abs\ (x) \vee y = abs(x)\}$
$\{y = abs\ (x)\}$

   b. Let's use sp on each branch then use Conditional Rule 1 for the whole conditional statement. Then we have:

$\{y = x\}$
if $x < 0$ **then**
    $\{y = x \wedge x < 0\}\ y := -x\ \{y_0 = x \wedge x < 0 \wedge y = -x\}\ \{y = abs\ (x)\}$
else
    $\{y = x \wedge x \geq 0\}$ **skip** $\{y = x \wedge x \geq 0\}\ \{y = abs\ (x)\}$
fi $\{y = abs\ (x)\}$

   c. Let's use wlp on each branch then use Conditional Rule 2 for the whole conditional statement. Then we have:

$\{y = x\}$
$\{(x < 0 \rightarrow -x = abs\ (x)) \wedge (x \geq 0 \rightarrow y = abs\ (x))\}$
if $x < 0$ **then**
    $\{-x = abs\ (x)\}\ y := -x\ \{y = abs\ (x)\}$
else
    $\{y = abs\ (x)\}$ **skip** $\{y = abs\ (x)\}$
fi $\{y = abs\ (x)\}$

Proof under Total Correctness 1: Avoid Runtime Errors in Loop-free Programs

So far, all the proof of triples (formal proofs, full proof outlines, minimal and partial proof outlines) and all the axioms and inference rules we learned are for partial correctness (although most of them work for both correctness). Each proof system we have seen is a proof system for partial correctness, which is the set of logical formulas determined by the sets of axioms and rules of inference for partial correctness.

Now let's see how to come up with a proof system for total correctness which can prove a triple $\{p\}\ S\ \{q\}$ provable under total correctness: $\vdash_{tot} \{p\}\ S\ \{q\}$: we need to avoid possible runtime errors and divergence. Here, let's look at how to avoid runtime errors in loop-free programs first.

- Previously, we have learned how to get wp from wlp: we include domain predicates for expressions $D(e)$, predicates $D(p)$, and statements $D(S)$ that guarantee the evaluation of e, p and S won't cause a runtime error.
- We use a similar idea to get $\vdash_{tot} \{p'\}\ S\ \{q'\}$ from $\vdash \{p\}\ S\ \{q\}$ by including domain predicates:
  - In general, for a loop-free S:
    - If we get $\vdash \{wlp(S, q)\}\ S\ \{q\}$, then for total correctness we have $\vdash_{tot} \{wp(S, q \wedge D(q))\}\ S\ \{q \wedge D(q)\}$.

- If we get $\vdash \{p\}\ S\ \{sp(p, S)\}$, then for total correctness we have $\vdash_{tot} \{p \wedge D(p) \wedge D(S)\}\ S\ \{sp(p \wedge D(p) \wedge D(S), S)\}$.

- For a predicate p, we define the **safe version** of p as $\downarrow p \equiv p \wedge D(p)$.
  - With this definition we can rewrite $wp(S, q) \Leftrightarrow D(S) \wedge \downarrow wlp(S, q)$
- If a predicate $p \Leftrightarrow \downarrow p$, then we say p **is safe**.
- In a triple $\{p\}\ S\ \{q\}$, if p and q are both safe, then $\{p\}\ S\ \{q\}$ is **safe**. In a formal proof under total correctness, all the triple judgments should be **provable under total correctness and safe**, so we cannot prove any unsafe triple. In this case, we need to use the safe version of the precondition and/or the postcondition while creating the proof.

(Inference Rules for Loop-free Statement under Total Correctness)

- We have the following non-loop axioms / inference rules for total correctness. Note that the antecedent triples and consequent triples in each rule are valid under total correctness.
  - **Skip Axiom** (total correctness version):

    $\{\downarrow p\}\ \mathbf{skip}\ \{\downarrow p\}$ skip

  - **Backward Assignment Axiom** (total correctness version):

    $\{D(e) \wedge \downarrow ((\downarrow q)\ [e\ /\ v])\}\ v \coloneqq e\ \{\downarrow q\}$ backward assignment

  - **Forward Assignment Axiom** (total correctness version):

    $\{D(e) \wedge \downarrow p\}\ v \coloneqq e\ \{(D(e) \wedge \downarrow p)[v_0\ /\ v] \wedge v = e[v_0\ /\ v]\}$ forward assignment

  - Sequence rule, strengthen precondition rule, weaken postcondition rule, conjunction rule, disjunction rule, conditional rule 1, nondeterministic conditional rule is still the same. The only difference is that now the antecedent triples and the consequent triples are safe and provable under total correctness.
    For example, the 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

    This rule still holds in the proof for total correctness, but now $\vdash_{tot} \{p_1\}\ S\ \{q_1\}$, $\vdash_{tot} \{p_2\}\ S\ \{q_2\}$, and p_1, q_1, p_2, q_2 are all safe, thus the consequent $\vdash_{tot} \{p_1 \wedge p_2\}\ S\ \{q_1 \wedge q_2\}$, and $p_1 \wedge p_2, q_1 \wedge q_2$ are safe.

  - **Conditional Rule 2** (total correctness version):
    1. $\{p_1\}\ S_1\ \{q_1\}$
    2. $\{p_2\}\ S_2\ \{q_2\}$
    3. $\{D(B) \wedge (B \rightarrow p_1) \wedge (\neg B \rightarrow p_2)\}\ \mathbf{if}\ B\ \mathbf{then}\ S_1\ \mathbf{else}\ S_2\ \mathbf{fi}\ \{q_1 \vee q_2\}$ if − else 1,2

    What is special about conditional rule 2 is that it contains a fresh expression B in the consequent.

4. Create a full proof outline for $\{b[x] < y\}\ y \coloneqq y + 1; z \coloneqq y\ \{q\}$ under total correctness. Find a possible q.
   - First, we add domain predicate in the precondition, then we can use forward assignment to find a provable q.
   - We can create the following proof outline:
     $\{b[x] < y \wedge 0 \leq x < size(b)\}$
     $y \coloneqq y + 1; \{b[x] < y_0 \wedge 0 \leq x < size(b) \wedge y = y_0 + 1\}$
     $z \coloneqq y\ \{b[x] < y_0 \wedge 0 \leq x < size(b) \wedge y = y_0 + 1 \wedge z = y\}$

5. Under total correctness, create a full proof outline for provable triple $\{p\}$ **if** $x \geq 0$ **then** $x \coloneqq sqrt(x)$ **else** $x \coloneqq y/b[k]$ **fi** $\{0 \leq x < y\}$. Find a possible p.
   - The postcondition is already safe.
   - We can calculate the $wp(\dots)$ as p first. But the calculation can be time consuming, and we might not be able to prove the triple with $wp(\dots)$ as p. Here, we use the conditional rule to create proof and come a provable precondition naturally.

   - We can create the following proof outline:
     $\{p\}$
     if $x \geq 0$ **then**
     $\{0 \leq sqrt(x) < y \wedge x \geq 0\}\ x \coloneqq sqrt(x)\ \{0 \leq x < y\}$
     else
     $\{0 \leq y/b[k] < y \wedge 0 \leq k < size(b) \wedge b[k] \neq 0\}\ x \coloneqq y/b[k]\ \{0 \leq x < y\}$
     fi $\{0 \leq x < y\}$

     Where $p \equiv T \wedge (x \geq 0 \rightarrow 0 \leq sqrt(x) < y \wedge x \geq 0) \wedge (x < 0 \rightarrow 0 \leq y/b[k] < y \wedge 0 \leq k < size(b) \wedge b[k] \neq 0)$

6. Under total correctness, create a full proof outline for provable triple $\{T\}$ **if** $x \geq 0$ **then** $x \coloneqq sqrt(x)$ **else** $x \coloneqq y/b[k]$ **fi** $\{q\}$. Find a possible q.
   - We find q naturally following the proof outline.
   - T is a safe precondition, but the program has the probability to create a runtime error so we cannot take all states into consideration. To create proof under total correctness, we need to strengthen the precondition to $\{T \wedge D(IF)\} \equiv T \wedge D(x \geq 0) \wedge \big(x \geq 0 \rightarrow D\big(x \coloneqq sqrt(x)\big)\big) \wedge \big(x < 0 \rightarrow D(x \coloneqq\ y/b[k])\big)$
     $\Leftrightarrow (x \geq 0 \rightarrow x \geq 0) \wedge (x < 0 \rightarrow 0 \leq k < size(b) \wedge b[k] \neq 0)$
     $\Leftrightarrow x < 0 \rightarrow 0 \leq k < size(b) \wedge b[k] \neq 0$
     $\Leftrightarrow x \geq 0 \vee 0 \leq k < size(b) \wedge b[k] \neq 0$

   - We can use Conditional Rule 1 to create the following proof outline:
     $\{x \geq 0 \vee 0 \leq k < size(b) \wedge b[k] \neq 0\}$
     if $x \geq 0$ **then**
     $\{(x \geq 0 \vee 0 \leq k < size(b) \wedge b[k] \neq 0) \wedge x \geq 0\}\ \{x \geq 0\}\ x \coloneqq sqrt(x)\ \{x_0 \geq 0 \wedge x = sqrt(x_0)\}$
     else
     $\{(x < 0 \rightarrow 0 \leq k < size(b) \wedge b[k] \neq 0) \wedge x < 0\}$
     $\{0 \leq k < size(b) \wedge b[k] \neq 0\}\ x \coloneqq y/b[k]\ \{0 \leq k < size(b) \wedge b[k] \neq 0 \wedge x = y/b[k]\}$
     fi
     $\{x_0 \geq 0 \wedge x = sqrt(x_0) \vee\ 0 \leq k < size(b) \wedge b[k] \neq 0 \wedge x = y/b[k]\}$

Examples 4, 5 and 6 have other solutions, here I only provide one of the possible proofs.