WEAKEST LIBERAL PRECONDITIONS

- The weakest liberal pre-condition of a statement S wrt a post-condition Q is an assertion that holds for all states such that if execution of S from this state terminates, the final state satisfies Q.

- If such a formula exists, let's denote it $wlp(S,Q)$, then:

$$\models_{par} \{P\}\, S\, \{Q\} \quad \text{whenever} \quad P \Rightarrow wlp(S,Q) \text{ is valid}$$

So, if one can compute $wlp(S,Q)$, verification of $\{P\}S\{Q\}$ can be reduced to checking the validity of the proof obligation $P \Rightarrow wlp(S,Q)$.

- An assertion languge Assn is called <u>expressive</u> if $wlp(S,Q)$ can be expressed as an assertion, for every $S \in$ Stm and $Q \in$ Assn.

- One can prove that:

$$\begin{array}{lcl}
wlp(\text{skip}, Q) & \Leftrightarrow & Q \\
wlp(x := a, Q) & \Leftrightarrow & Q[a/x] \\
wlp(S_1 ; S_2, Q) & \Leftrightarrow & wlp(S_1, wlp(S_2, Q)) \\
wlp(\underline{\text{if}}\ b\ \underline{\text{then}}\ S_1\ \underline{\text{else}}\ S_2, Q) & \Leftrightarrow & (b \Rightarrow wlp(S_1, Q)) \\
 & & \wedge (\neg b \Rightarrow wlp(S_2, Q))
\end{array}$$

Notice that these equivalences can be seen as a definition by structural induction! Call the function $wlp'(S,Q)$.

<u>EXAMPLE</u> Verify $\{x \le 0\}$ <u>if</u> $0 \le x$ <u>then</u> $x := x+1$ <u>else</u> $x := 1$ $\{x = 1\}$

First, we compute:

$$\begin{aligned} & wlp'(\underline{\text{if}}\ 0 \le x\ \underline{\text{then}}\ x := x+1\ \underline{\text{else}}\ x := 1,\ x = 1) \\ = & (0 \le x \Rightarrow wlp'(x := x+1, x = 1)) \wedge ((\neg 0 \le x \Rightarrow wlp'(x := 1, x = 1)) \\ = & (0 \le x \Rightarrow x+1 = 1) \wedge (\neg 0 \le x \Rightarrow 1 = 1) \end{aligned}$$

Next, we obtain the proof obligation:

$$x \le 0 \Rightarrow (0 \le x \Rightarrow x+1 = 1) \wedge (\neg 0 \le x \Rightarrow 1 = 1)$$

which is easily shown to be valid.

- In general, extending this approach to the <u>while</u> b <u>do</u> S construct is cumbersome. Two possible approaches are:
  - unfolding the loop some fixed number k of times (unsound)
  - using user-provided loop invariants

<u>EXERCISE</u> Extend the definition of wlp' to loops of the shape $\{I\}$ <u>while</u> b <u>do</u> S, i.e. loops annotated with candidate invariants.

Hint: it may be easier to use an accumulator argument if the definition should be by structural induction on S.