Propositional Logic – Logic of statements

Statements about programs

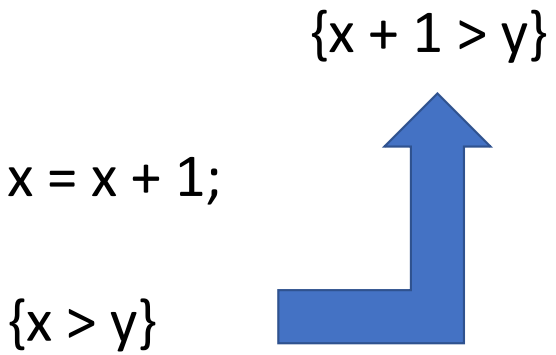
{P: true before execution}

Program_statement

{Q: true after execution}

Pre conditions and Post conditions

Proofs proceed from Post Conditions to Preconditions



Back substitution

$$\frac{\{P\} \rightarrow \{Q(x)\}}{\{P: e/x[Q]\}\ x := e\ \{Q\}}$$

{P: (a / 10) > y}

x := a / 10;

{Q: x > y}

Chaining

F0->F1,F2->F'2,F3->F4,{F1}S1{F2},{F'2}S2{F3}

{F0}S1;S2{F4}

{F0}
{F1} S1 {F2}
{F'2} S2 {F3}
{F4}

{F0: **(y + z) > (k – x)** }
a := y + z;
b := k - x;
{F4: a > b}

{F0: **(y + z) > (k – x)**}
{F1: **(y + z) > (k – x)**}
a := y + z;
{F2: **a > (k – x)**}
{F2': **a > (k – x)**}
b := k - x;
{F3: **a > b**}
{F4: **a > b**}

Conditional

{Pre **and** cond}S1{Post}, {Pre **and not** cond}S2{Post}

{Pre}**if** (cond) S1 **else** S2{Post}

```
{Pre: (P1 and cond) or (P2 and not cond)}
if (cond)
        {P1} S1 {Post}
Else
        {P2} S2 {Post}
{Post}
```

```
{Pre: (z > 5 and x > 0) or (-z > 5 and x <= 0)}
   if (x > 0)
           {P1: z > 5} y := z {Post: y > 5}
   else
           {P2: -z > 5} y := -z {Post: y > 5}
{Post: y > 5}
```

Loop Invariant

$$\frac{\{I \textbf{ and } cond\}S\{I\}}{\{I\}\textbf{while } (cond) \text{ do } S\{I \textbf{ and not } cond\}}$$

{I}
while (cond)
do
    {I and cond}S{I}
end
{I and not cond}

{i = 0 and mysum = 0}
I = **{ mysum = SUM(k=0..i-1)(a[k]) }**
while (i < n) do
{ mysum = SUM(k=0..i-1)(a[k]), i < n } ->
{ mysum + a[i] = SUM(k=0..i)(a[k]), 0 <= i + 1 <= n}
  mysum = mysum + a[i];
{ mysum = SUM(k=0..i)(a[k]), 0 <= i + 1 <= n }
  i = i + 1;
{ mysum = SUM(k=0..i-1)(a[k]), 0 < i <= n }
end
{ mysum = SUM(k=0..i-1)(a[k]), i = n }

```
// counting loop
        Pre:{0<=n} ->
        {0<=n}
i = 0;
        I:{i<=n}
while (i < n)
begin
        I:{i<=n} and C:{i<n} ->
        {i+1<=n}
    i = i + 1;
        I:{i<=n}
end
        I: {i<=n} and nC:{i>=n} ->
        Post:{i=n}
```

```
// find largest between m and n
        Pre:{m<n} ->
        {A[m] >= A[m:m-1] and m-1<=n}
j = m;
        {A[j] >= A[m:m-1] and m-1<=n}
i = m;
        I:{A[j] >= A[m:i-1] and i-1<=n}
while( i <= n )
begin
           I:{A[j] >= A[m:i-1] and i-1<=n} and C:{i<=n} ->
           {A[i] >= A[m:i] and i<=n and A[i] > A[j]},
           {A[j] >= A[m:i] and i<=n and A[i] <= A[j]}
   if (A[i] > A[j])
      j = i;
           {A[j] >= A[m:i] and i<=n}
   i++;
           I:{A[j] >= A[m:i-1] and i-1<=n}
end
        I:{A[j] >= A[m:i-1] and i-1<=n} and nC:{i>n} ->
        {A[j] >= A[m:i-1] and i=n+1} ->
        Post:{A[j]>=A[m:n]}
```