Verification

$\{\alpha\}P\{\beta\}$ means that if formula α is true before execution of P, P is executed and terminates, then formula β will be true.

α, β, are the **precondition** and **postcondition** of the program P, respectively. They are given as formulas of $\mathcal{L}_A$.

Using a finite set of rules for program verification, we want to show that $\{\alpha\}P\{\beta\}$ holds, and conclude that the program is correct *with respect to the specification* α, β.

Rules for program verification

Consequence right and left:

$$\frac{\{\alpha\}P\{\beta\} \qquad (\beta \supset \gamma)}{\{\alpha\}P\{\gamma\}} \qquad \frac{(\gamma \supset \alpha) \qquad \{\alpha\}P\{\beta\}}{\{\gamma\}P\{\beta\}}$$

Composition and assignment:

$$\frac{\{\alpha\}P_1\{\beta\} \qquad \{\beta\}P_2\{\gamma\}}{\{\alpha\}P_1P_2\{\gamma\}} \qquad \frac{x = t}{\{\alpha(t)\}x = t\{\alpha(x)\}}$$

If:

$$\frac{\{\alpha \wedge \beta\}P_1\{\gamma\} \qquad \{\alpha \wedge \neg\beta\}P_2\{\gamma\}}{\{\alpha\} \textbf{ if } \beta \textbf{ then } P_1 \textbf{ else } P_2 \ \{\gamma\}}$$

While:

$$\frac{\{\alpha \wedge \beta\}P\{\alpha\}}{\{\alpha\} \textbf{ while } \beta \textbf{ do } P \ \{\alpha \wedge \neg\beta\}}$$

Verification of `mult(A,B)`

We verify `mult(A,B)` which computes $y = A \cdot B$:

```
mult(int A, int B) {
  int a = A;
  int b = B;
  int y = 0;
  while (b > 0) {
    y = y + a;
    b = b - 1; }}
```

Each pass through the **while** loop adds a to y, but $a \cdot b$ decreases by a because b is decremented by 1.

Loop invariant:

$$(y + (a \cdot b) = A \cdot B) \wedge b \geq 0$$

We want to show:

$$\{B \geq 0\}\texttt{mult(A,B)}\{y = AB\} \tag{1}$$

using the loop invariant (the "creative" part of the proofs of correctness). To save space, write tu instead of $t \cdot u$. Let $t \geq u$ abbreviate the $\mathcal{L}_A$-formula $\exists x(t = u + x)$, and let $t \leq u$ abbreviate $u \geq t$.

1	$\{y + a(b-1) = AB \wedge (b-1) \geq 0\}$`b=b-1;`$\{y + ab = AB \wedge b \geq 0\}$	assignment
2	$\{(y+a) + a(b-1) = AB \wedge (b-1) \geq 0\}$`y=y+a;`$\{y + a(b-1) = AB \wedge (b-1) \geq 0\}$	assignment
3	$(y + ab = AB \wedge b - 1 \geq 0) \supset ((y+a) + a(b-1) = AB \wedge b - 1 \geq 0)$	theorem
4	$\{y + ab = AB \wedge b - 1 \geq 0\}$`y=y+a;`$\{y + a(b-1) = AB \wedge b - 1 \geq 0\}$	consequence left 2 and 3
5	$\{y + ab = AB \wedge b - 1 \geq 0\}$`y=y+a;b=b-1;`$\{y + ab = AB \wedge b \geq 0\}$	composition on 4 and 1
6	$(y + ab = AB) \wedge b \geq 0 \wedge b > 0 \supset (y + ab = AB) \wedge b - 1 \geq 0$	theorem
7	$\{(y + ab = AB) \wedge b \geq 0 \wedge b > 0\}$`y=y+a; b=b-1;`$\{y + ab = AB \wedge b \geq 0\}$	consequence left 5 and 6
8	$\{(y + ab = AB) \wedge b \geq 0\}$ `while (b>0) {` `y=y+a;` `b=b-1; }` $\{y + ab = AB \wedge b \geq 0 \wedge \neg(b > 0)\}$	while on 7
9	$\{(0 + ab = AB) \wedge b \geq 0\}$ `int y=0;` $\{(y + ab = AB) \wedge b \geq 0\}$	assignment

10	$\{(0 + ab = AB) \wedge b \geq 0\}$ `int y=0;`
	`while (b>0) {`
	`y=y+a;`
	`b=b-1; }` $\{y + ab = AB \wedge b \geq 0 \wedge \neg(b > 0)\}$
		composition on 9 and 8

11	$\{(0 + aB = AB) \wedge B \geq 0\}$ `int b=B;` $\{(0 + ab = AB) \wedge b \geq 0\}$	assignment

12	$\{(0 + aB = AB) \wedge B \geq 0\}$ `int b=B;`
	`int y=0;`
	`while (b>0) {`
	`y=y+a;`
	`b=b-1; }` $\{y + ab = AB \wedge b \geq 0 \wedge \neg(b > 0)\}$
		composition on 11 and 10

13	$\{(0 + AB = AB) \wedge B \geq 0\}$ `int a=A;` $\{(0 + aB = AB) \wedge B \geq 0\}$	assignment
14	$\{(0 + AB = AB) \wedge B \geq 0\}$ `mult(int A, int B)` $\{y + ab = AB \wedge b \geq 0 \wedge \neg(b > 0)\}$	composition on 13 and 12
15	$B \geq 0 \supset ((0 + AB = AB) \wedge B \geq 0)$	theorem
16	$(y + ab = AB \wedge b \geq 0 \wedge \neg(b > 0)) \supset y = AB$	theorem
17	$\{B \geq 0\}$ `mult(int A, int B)` $\{y + ab = AB \wedge b \geq 0 \wedge \neg(b > 0)\}$	consequence left on 15 and 14
18	$\{B \geq 0\}$ `mult(int A, int B)` $\{y = AB\}$	consequence right on 16 and 17