**Ex.** $\forall x(\alpha \vee \beta) \models (\forall x\alpha \vee \forall x\beta)$ ?

No, not necessarily. How to prove this ?

Use the def of logical consequence.

To prove that the RHS is **not** a logical consequence of the LHS, we must:

Exhibit:

1. a model $\mathcal{M}$
2. an object assignment $\sigma$
3. formulas $\alpha, \beta$

such that:

$\mathcal{M} \models \forall x(\alpha \vee \beta)[\sigma]$, but $\mathcal{M} \not\models (\forall x\alpha \vee \forall x\beta)[\sigma]$.

Let $\alpha$ and $\beta$ be $Px$ and $Qx$, respectively ($P, Q$ unary predicates).

Now define $\mathcal{M}$ and $\sigma$.

Since the formulas are sentences, no need to define $\sigma$.

$\mathcal{M}$: let the universe of discourse be $M = \mathbb{N}$.

We still need to give meaning in $\mathcal{M}$ to $P, Q$. Let
$P^{\mathcal{M}} = \{0, 2, 4, \ldots\}$, and $Q^{\mathcal{M}} = \{1, 3, 5, \ldots\}$.

Then: $\mathcal{M} \vDash \forall x(Px \vee Qx)$
(bec. every number is even or odd).

But, $\mathcal{M} \nvDash (\forall x Px \vee \forall x Qx)$
(bec. it is not true that either all numbers are even or all numbers are odd).

## Program Verification

- $\{\alpha\}P\{\beta\}$ means that if formula $\alpha$ is true before execution of $P$, $P$ is executed and terminates, then formula $\beta$ will be true.
- $\alpha, \beta$, 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* $\alpha, \beta$.
- The rules are of the form $\frac{\text{antecedent}}{\text{consequent}}$, and their meaning is that if the antecendent is true, then so is the consequent. (Truth is here given by $\underline{\mathbb{N}}$.) In other words, the rules are **sound**.

## Rules for program verification

Consequence right and left:

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

Composition and assignment:

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

If:

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

While:

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

**Ex.** Verify `mult(A,B)` which computes $y = A * B$:

```
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 $a * b$ decreases by $a$ because $b$ is decremented by 1.

**Loop invariant:**

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

We want to show:

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

using the loop invariant (the "creative" part of the proofs of correctness).

$B \geq 0$ is the precondition.

$y = AB$ is the postcondition.

To save space, write $tu$ instead of $t * u$. Let $t \geq u$ abbreviate the $\mathcal{L}_A$-formula $\exists x(t = u + x)$, and let $t \leq u$ abbreviate $u \geq t$.

Correctness follows from (1).

**(1)** Using **assignment** on line 7 we obtain:

$$\{y + a(b-1) = AB \wedge (b-1) \geq 0\}$$

```
b=b-1;
```

$$\{y + ab = AB \wedge b \geq 0\}$$

**(2)** Using **assignment** on line 6 we obtain:

$$\{(y + a) + a(b-1) = AB \wedge (b-1) \geq 0\}$$

```
y=y+a;
```

$$\{y + a(b-1) = AB \wedge (b-1) \geq 0\}$$

**Theorem 1:**

$$(y + ab = AB \land b - 1 \geq 0) \supset ((y + a) + a(b - 1) = AB \land b - 1 \geq 0)$$

**(3)** Therefore, using **consequence left**, we can conclude from **(2)** and theorem 1:

$$\{y + ab = AB \land b - 1 \geq 0\}$$

```
y=y+a;
```

$$\{y + a(b - 1) = AB \land b - 1 \geq 0\}$$

**(4)** Thus, with **composition** on **(3)** and **(1)**:

$$\{y + ab = AB \wedge b - 1 \geq 0\}$$

```
y=y+a;
b=b-1;
```

$$\{y + ab = AB \wedge b \geq 0\}$$

**Theorem 2:**

$$(y + ab = AB) \wedge b \geq 0 \wedge b > 0 \supset (y + ab = AB) \wedge b - 1 \geq 0$$

(Let $t > u$ denote $t \geq u \wedge t \neq u$.)

**(5)** Using **consequence left** on **(4)** with theorem 2:

$$\{(y + ab = AB) \wedge b \geq 0 \wedge b > 0\}$$

```
y=y+a;
b=b-1;
```

$$\{y + ab = AB \wedge b \geq 0\}$$

**(6)** Using **while** on **(5)**:

$$\{(y + ab = AB) \wedge b \geq 0\}$$

```
while (b>0){
 y=y+a;
 b=b-1;}
```

$$\{y + ab = AB \wedge b \geq 0 \wedge \neg(b > 0)\}$$

**(7)** Using **assignment** on lines 2,3,4, a theorem, and compositions:

$\{B \geq 0\}$

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

$\{y = AB\}$

which is just $\{B \geq 0\}$`mult(A,B)`$\{y = AB\}$ as we wanted.

**Ex.** Fill in the details of step **(7)**.