Continuing from last lecture, let us write a state-based program for computing factorial.

\[ \text{FACT}(n) \]
1. \( x := 1 \)
2. \( y := 1 \)
3. while \( x < n \)
4. \( x := x + 1 \)
5. \( y := y \times x \)
6. \( z := y \)

\{ z = n! \} - assertion that should be true after executing the program.
Here \( n! = n \cdot (n - 1) \cdot (n - 2) \ldots 3 \cdot 2 \cdot 1 \)

State: function from variables to values, e.g. \( \{(n, 5), (x, 1), (y, 17), (z, 1)\} \)
Procedure (imperative or while program): function from states to states
Fact:
\[ \{(n, 5), (x, 1), (y, 17), (z, 1)\}, \{(n, 5), (x, 5), (y, 120), (z, 120)\}, \{(n, 4), (x, 3), (y, 3), (z, 3)\}, \{(n, 4), (x, 4), (y, 24), (z, 24)\}, \ldots \]  

From now on, let
\[ s_0 = \{(n, 5), (x, 1), (y, 17), (z, 1)\} \]
\[ s_1 = \{(n, 5), (x, 5), (y, 120), (z, 120)\} \]

Rules for value of an expression in a state:
For numbers \( n \in \mathbb{N} \), \[ [n](s) = n \]
For variables \( x \), \[ [x](s) = s(x) \]
\[ [c](s) = v \quad [d](s) = u \]
\[ [c + d](s) = v + u \]

Rules for operational semantics of while programs
\[ < A, s > \rightarrow s' \]
\[ < \text{program}, \text{state} > \rightarrow \text{state} \]
“If program A is executed in state s, the result is state \( s' \)”, e.g. \[ < \text{Fact}, s_0 > \rightarrow s_1 \]

\[ [E](s) = v \]
\[ < x := E, s > \rightarrow s[(x, v)] \]

\( s[(x, v)] \) - the same state as s, except that \( x \) is mapped to \( v \).

Meaning of if-then clause:
\[ [c](s) \text{ true} \quad < A, s > \rightarrow s' \]
\[ < \text{if } c \text{ then } A, s > \rightarrow s' \]
\[ [c](s) \text{ false} \]
\[ < \text{if } c \text{ then } A, s > \rightarrow s \]
Meaning of sequential computation of $A; B$:

$$< A, s > \rightarrow s' \quad < B, s' > \rightarrow s'' \quad < A, B, s > \rightarrow s''$$

Meaning of while:

$$\begin{align*}
& [c] (s) \text{ false} \quad < \text{while } c \text{ do } A, s > \rightarrow s \\
& [c] (s) \text{ true} \quad < A, s > \rightarrow s' \quad < \text{while } c \text{ do } A, s' > \rightarrow s''
\end{align*}$$

Example:

\[
\begin{align*}
[x < n](\ldots) &= \text{false} \\
\ldots < \text{while } \{(n, 5), (x, 5), (y, 120), (z, 1)\} > \rightarrow s_1 \\
\ldots < \text{while } \{(n, 5), (x, 4), (y, 24), (z, 1)\} > \rightarrow s_1 \\
\ldots < \text{while } \{(n, 5), (x, 3), (y, 6), (z, 1)\} > \rightarrow s_1 \\
< \text{while } x < n \text{ do } x := x + 1, y = y \times x, \{(n, 5), (x, 1), (y, 1), (z, 1)\} > \rightarrow \{(n, 5), (x, 5), (y, 120), (z, 1)\} \\
& \cdots \\
& < \text{Fact, } s_0 > \rightarrow s_1
\end{align*}
\]

**Control flow in the graph representation of FACT**

Now, let us go back to the pseudocode from the beginning of the lecture:

```
FACT(n)
1  x := 1
2  y := 1
3  while x < n
4    x := x + 1
5  y := y \times x
6  z := y
```

We have two tasks at hand:

1. Illustrate the running of the program using a control-flow diagram.
2. Show why every step of the while program holds, i.e. find a loop invariant.

In the following graph
edges are labeled with instructions
nodes are labeled with assertions
Loop invariant: $y = x! \land (x \leq n \lor (n = 0 \land y = 1))$

**Rules for axiomatic semantics of while programs (Hoare logic)**

```
 assertion (precondition) ↑  program ↑  assertion (postcondition)

"If assertion $p$ is true and you execute $A$, and $A$ terminates, then assertion $q$ is true afterwards"
```

**Rules:**

**R1:**

\[
\{q[E/X]\} \quad X := E \{q\}
\]

Example:

\[
\{y = (x)\} \quad x := x + 1 \quad \{y = (x - 1)\}
\]

**R2:**

\[
p \land \neg c = q \quad \{p \land c\} \quad A \{q\}
\]

\[
\{p\} \quad \text{if } c \text{ then } A \{q\}
\]

**R3:**

\[
\{p\} \quad A \{r\} \quad \{r\} \quad B \{q\}
\]

\[
\{p\} \quad A; B \{q\}
\]

**R4:**

\[
\{p \land c\} \quad A \{p\}
\]

\[
\{p\} \quad \text{while } c \text{ do } A \{-c \land p\}
\]

**R5:**

\[
p \Rightarrow p' \quad \{p'\} \quad A \{q'\} \quad q \Rightarrow q'
\]

Example:
\{y = x! \land x < n\} \text{while } x < n \text{ do } x = x + 1 ; y = y \ast x \{y = x!\}

\{\text{true}\}\text{Fact}\{z = n!\}

**Nondeterminism:**

\[
\frac{\{p\} A \{q\} \quad \{p\} B \{q\}}{\{p\} A|B \{q\}}
\]

**Homework**

1) Use operational rules to devise the sentence

\[
< \text{gcd}, \{(m,4), (n,2), (\ldots)\} >
\]

2) Find a loop invariant and prove

\[
\{n \geq 0, m \geq 0\} P \{r = \text{gcd}(m,n)\}