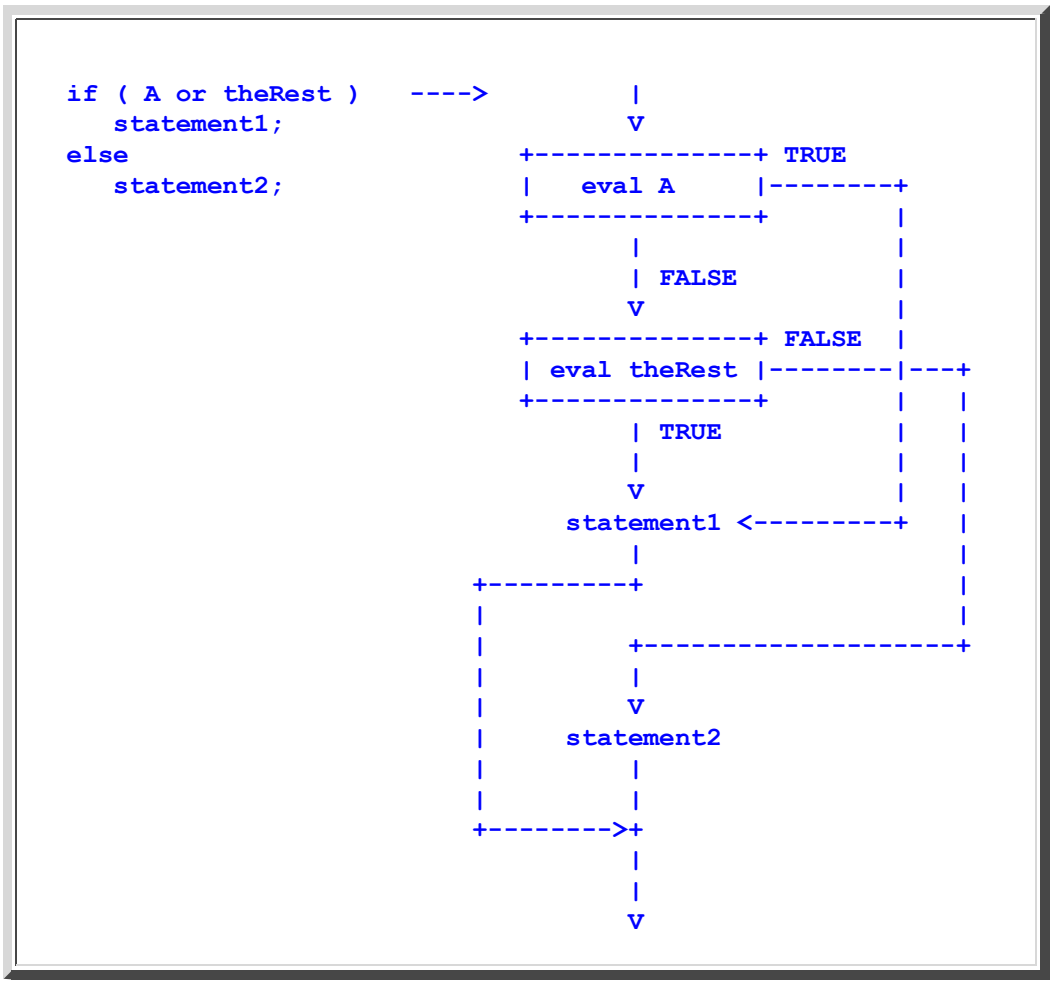
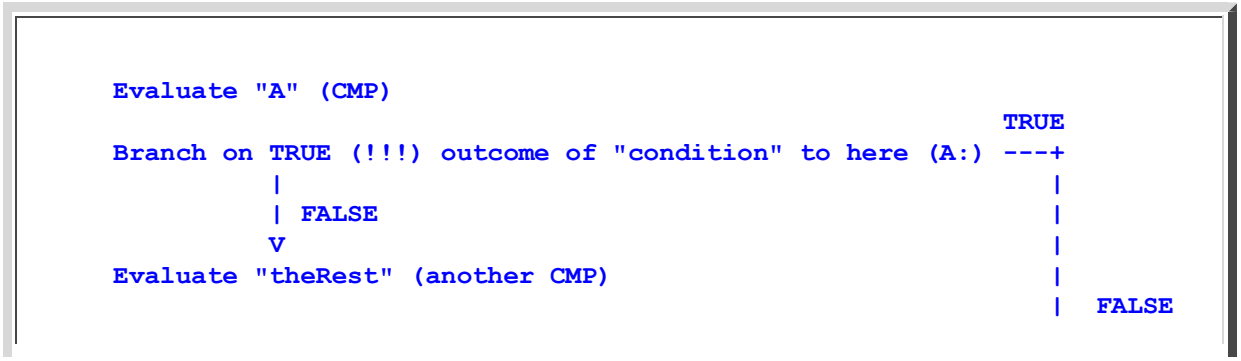
Compound condition with OR

- **Program control** of the `if`-statement with an compound condition "A or theRest":



  - If **A** is true, then **A** `or` "something" is also true, regardless what the "something" evaluates to.
  - Therefore, we can branch (jump) to the then-part as soon as we find that **A** is true.

- Assembler construct that realizes the control flow of the `if`-statement with an compound condition "A or theRest" is as follows:



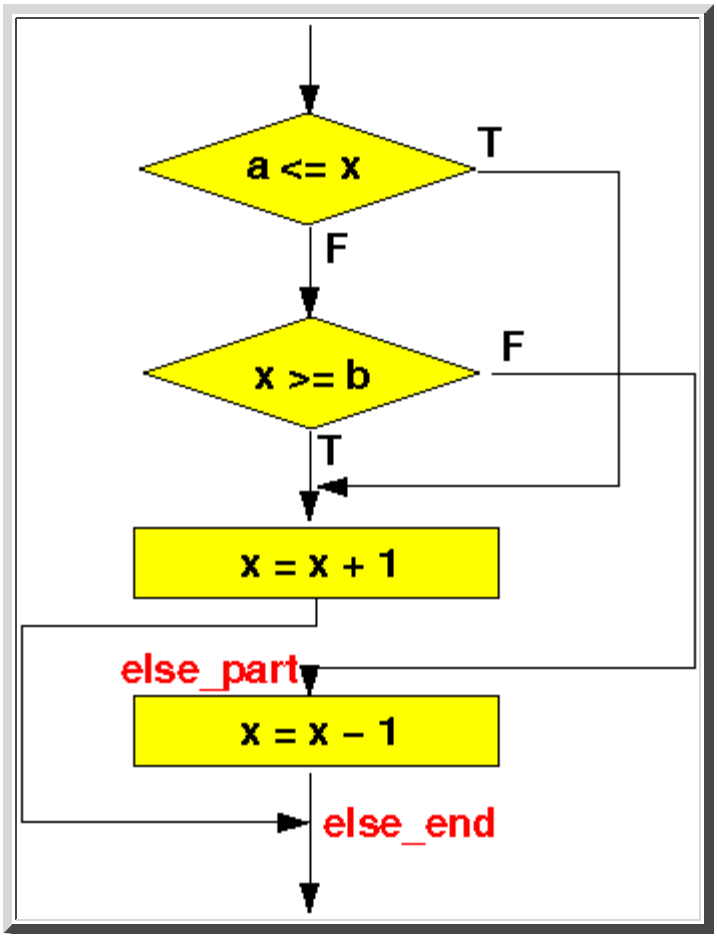
```
     Branch on FALSE (!!!) outcome of "theRest" to here (B:) ----|-----+
               |                                                 |     |
               | TRUE                                            |     |
               V                                                 |     |
A:        "statement1" assembler code <--------------------------+     |
               |                                                      |
       Branch always to there (C:) -------------+                     |
                                                |                     |
                                                |                     |
B:        "statement2" assembler code <---------|---------------------+
               |                                |
               V                                |
C:             +<-------------------------------+
               |
               V
```

- Example:

```
int x, y, a;

if (x <= a || x >= b)
   x = x + 1;
else
   x = x - 1;
```

The **flow chart** of the **above program** is:



Assembler program for this compound if-statement: click here