東邦大学

生命（いのち）の科学で未来をつなぐ

条件分岐とIF文

条件分岐　⇒　if 文だ

東邦大学

条件分岐　⇒　if 文だ

if (x≦0)　x=－x

東邦大学

条件分岐　⇒　if 文だ

if (x≦0)　x=－x

もし (x≦0) なら xの符号を反転

東邦大学

フローチャート（流れ図）

if (x≦0) x=－x

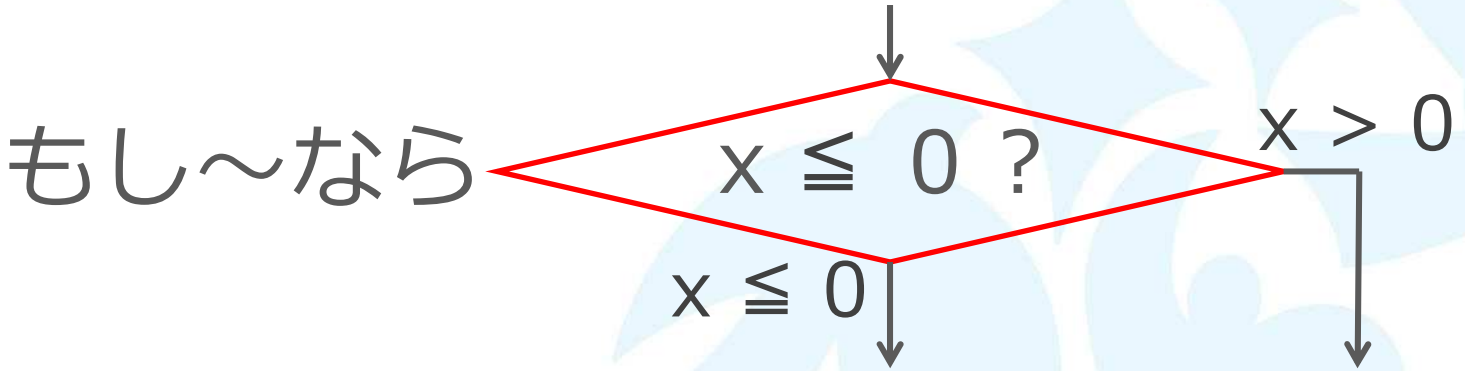
フローチャート（流れ図）

ここは？

if (x≦0)　x=－x

フローチャート（流れ図）

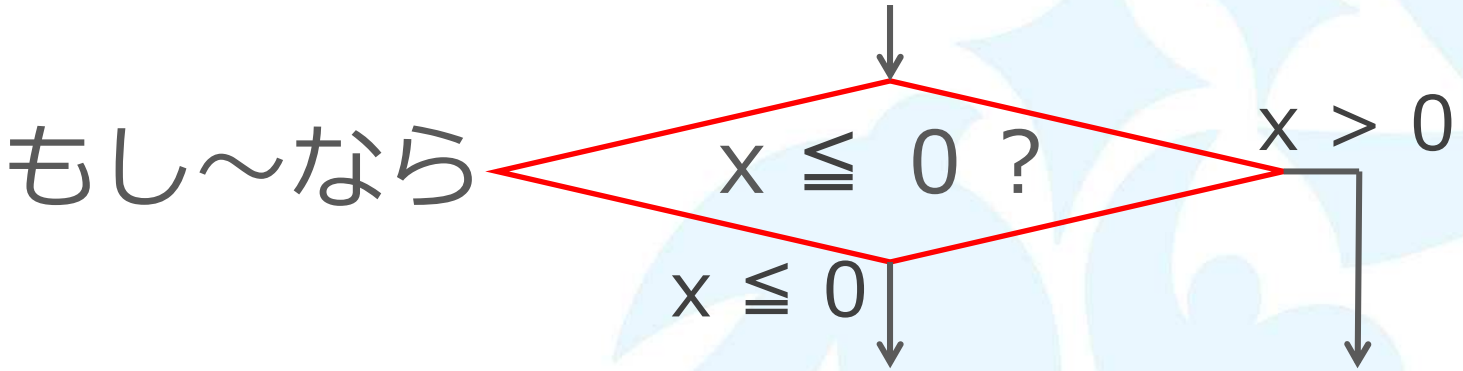
if (x≦0)　x=－x



東邦大学

フローチャート（流れ図）

if (x≦0)　x=－x

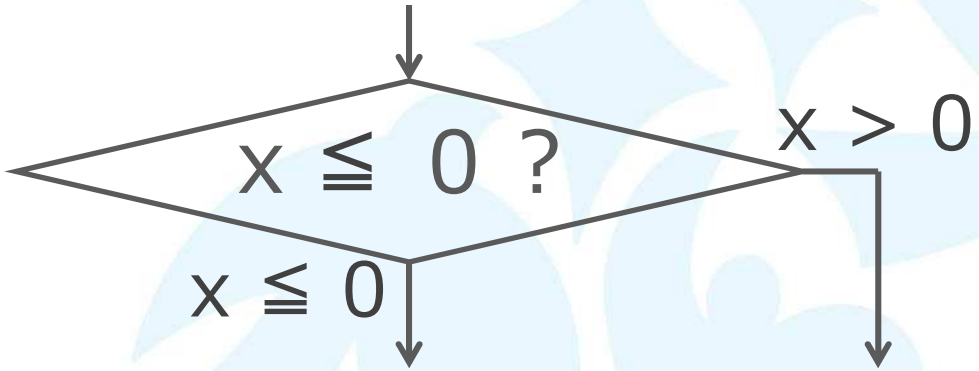


条件分岐と呼ぶ

フローチャート（流れ図）

if (x≦0)　x=－x

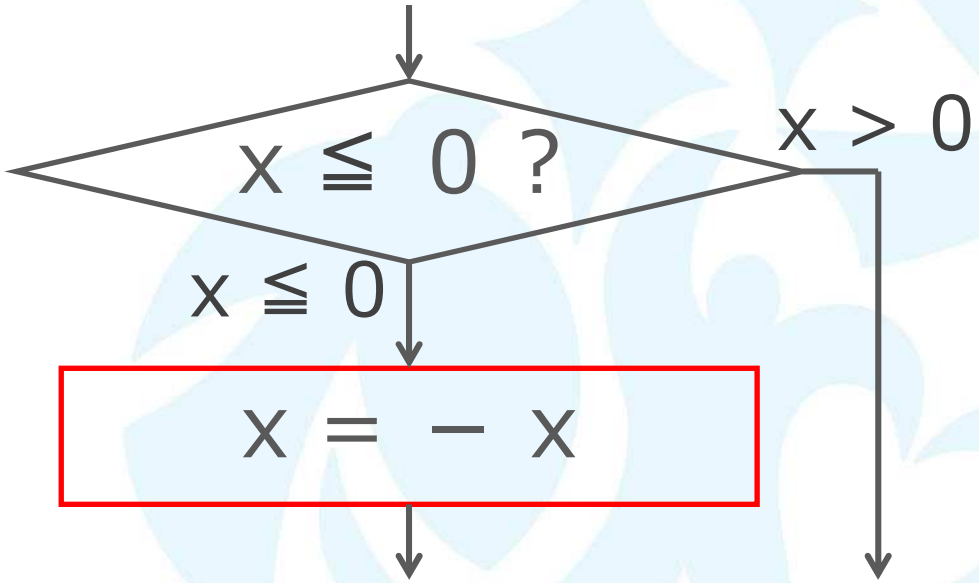
ここは？



東邦大学

フローチャート（流れ図）

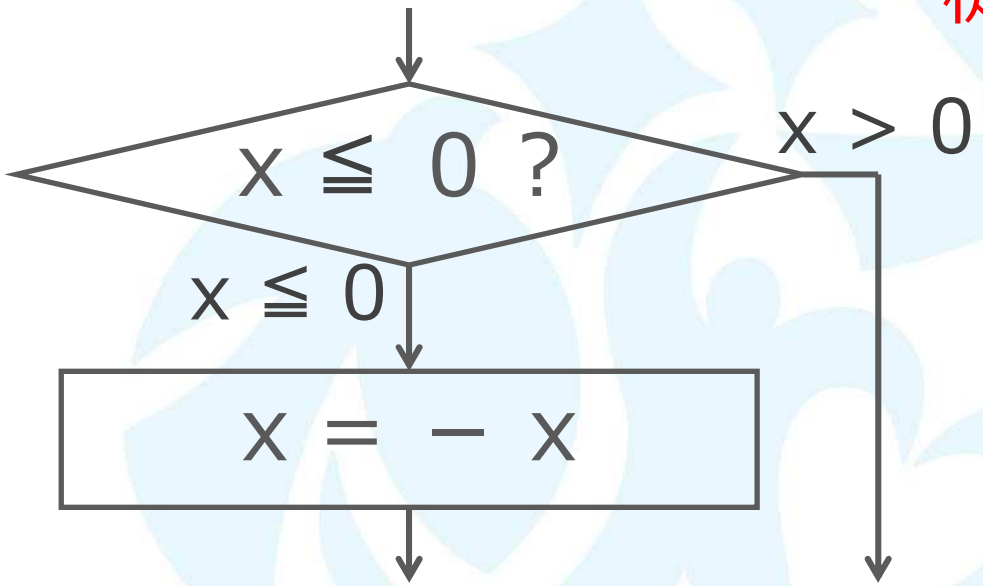
if (x≦0) x=－x



東邦大学

機械命令では

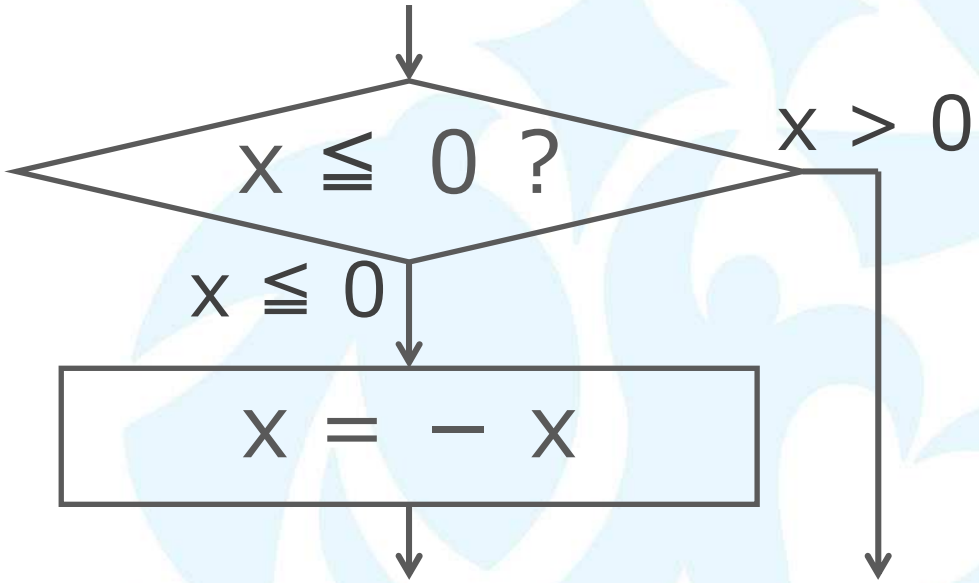
流れ図
↓
機械命令



東邦大学

機械命令では

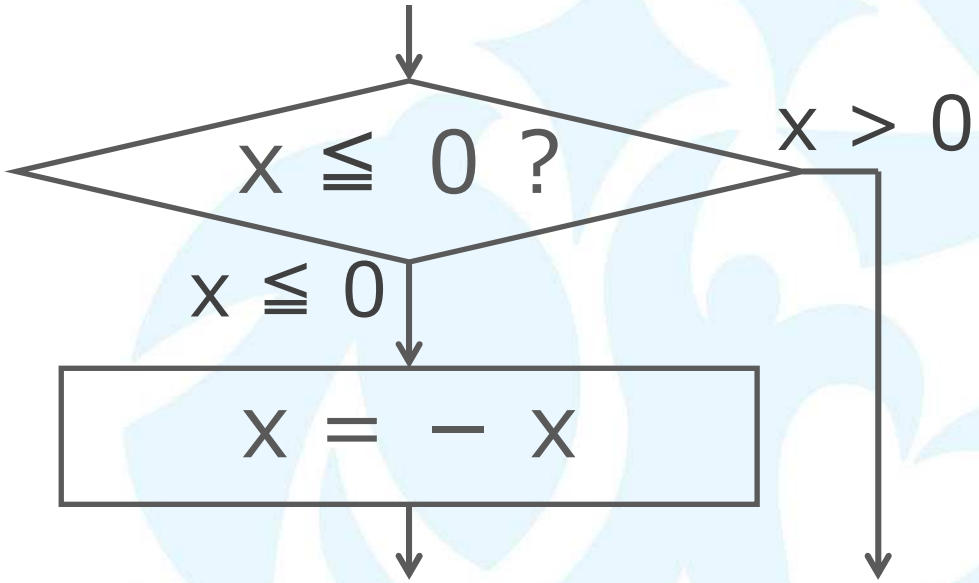
１命令ごとにして、並べなければならない



東邦大学

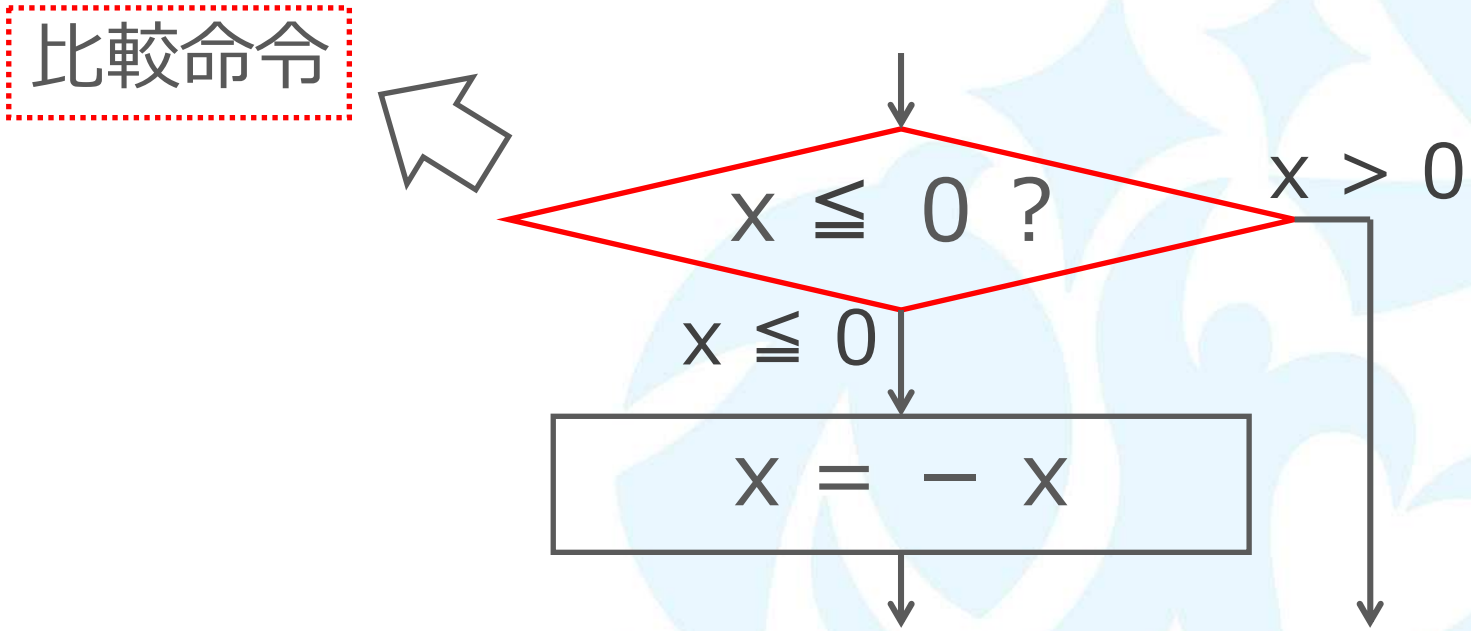
機械命令では

注）ここではCOMET IIの機械命令を考える
（CPUの種類によって異なる）



東邦大学

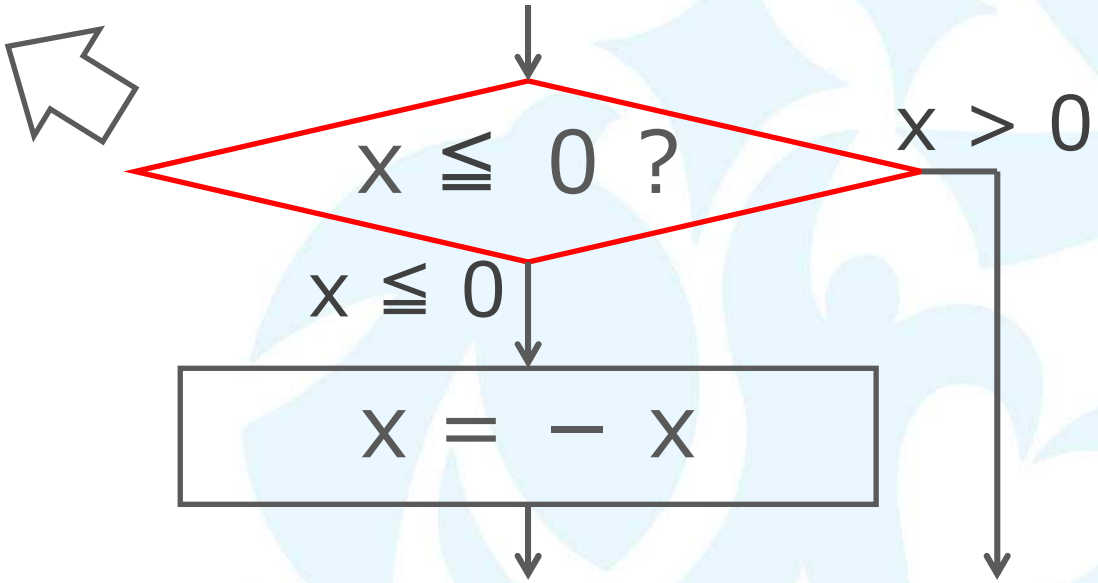
機械命令では



東邦大学

機械命令では

比較命令
(CPA命令)

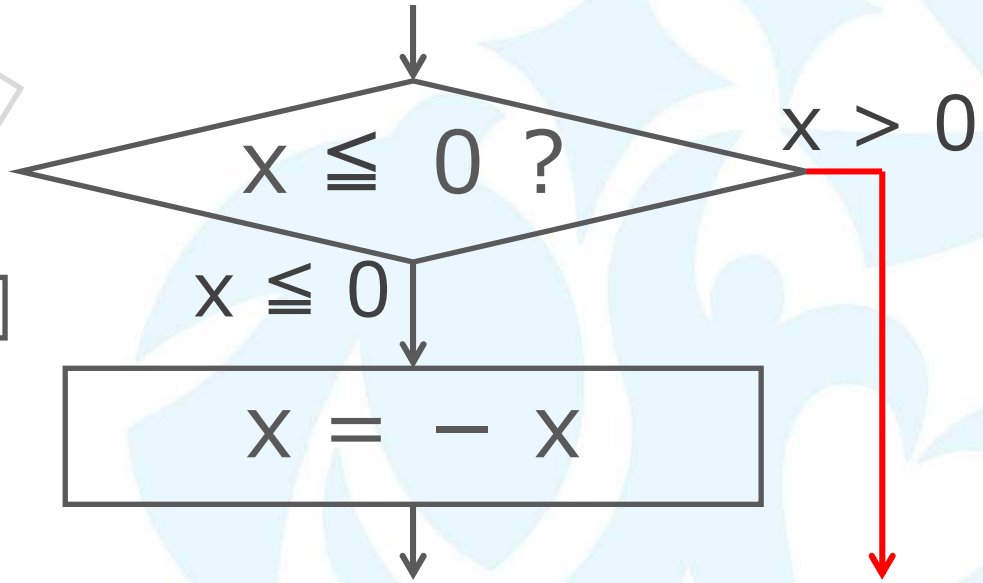


東邦大学

機械命令では

比較命令
(CPA命令)

条件分岐
命令

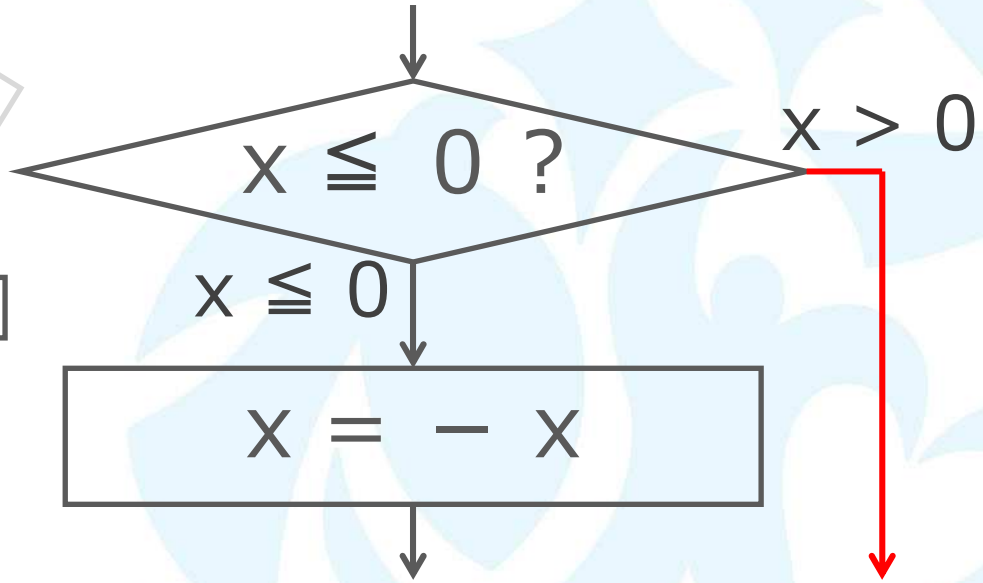


東邦大学

機械命令では

比較命令
(CPA命令)

条件分岐
命令
(JPL命令)

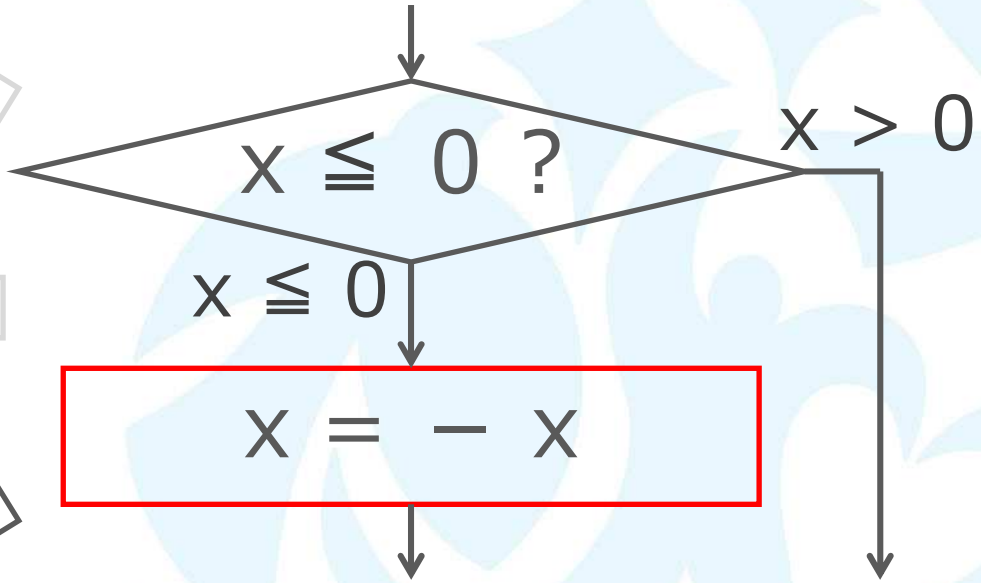


東邦大学

機械命令では

比較命令
(CPA命令)

条件分岐
命令
(JPL命令)



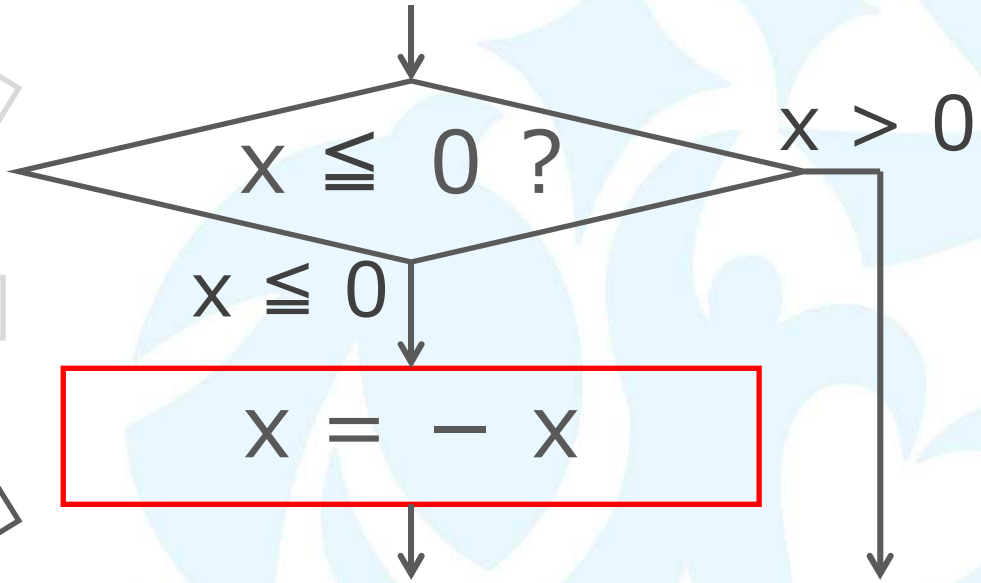
ここは前
やった代入文

東邦大学

機械命令では

比較命令
(CPA命令)

条件分岐
命令
(JPL命令)



ここは前に
やった代入文

(LD命令)
(SUBA命令)
(ST命令)

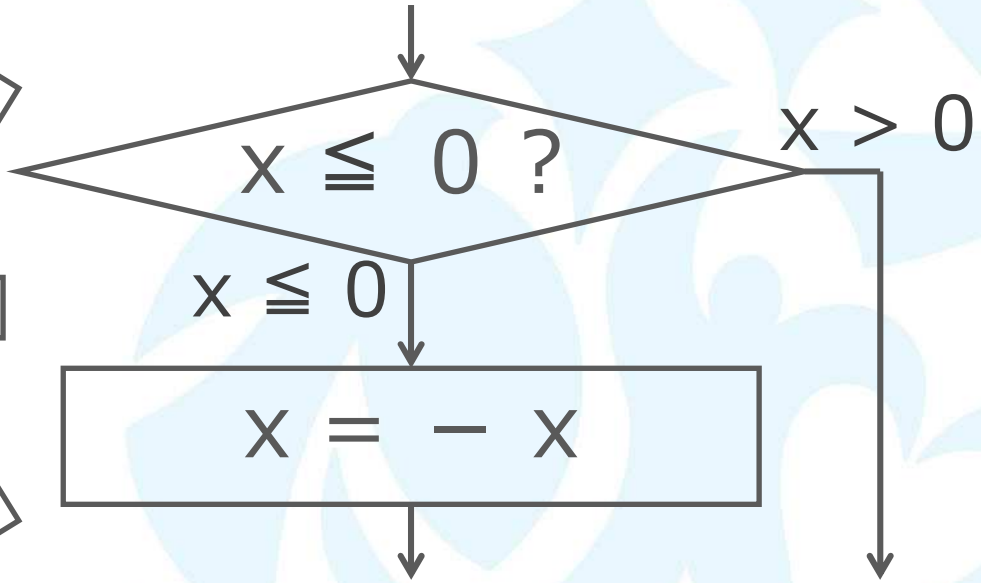
東邦大学

機械命令では

比較命令
(CPA命令)

条件分岐
命令
(JPL命令)

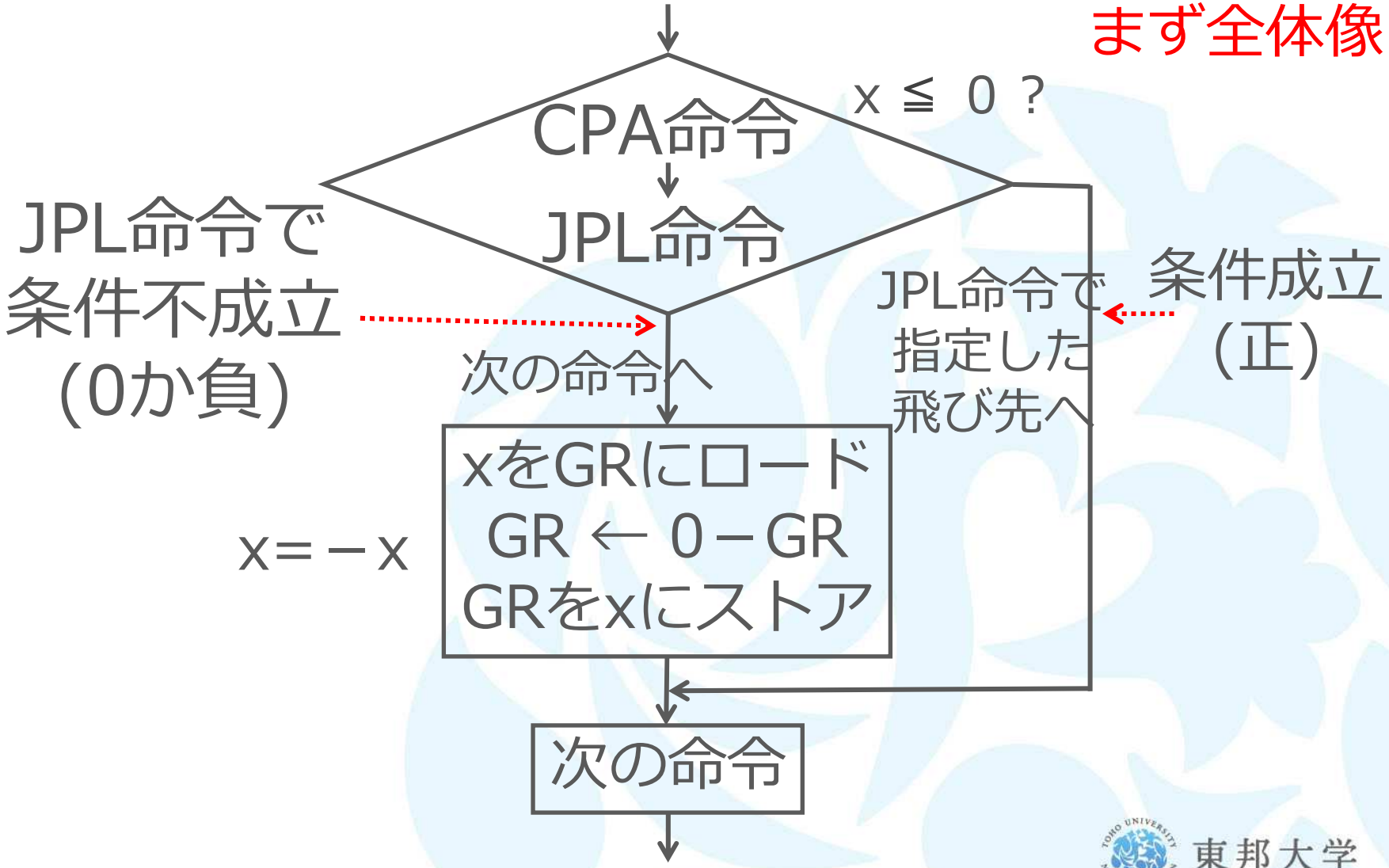
ここは前に
やった代入文
(LD命令)
(SUBA命令)
(ST命令)



東邦大学

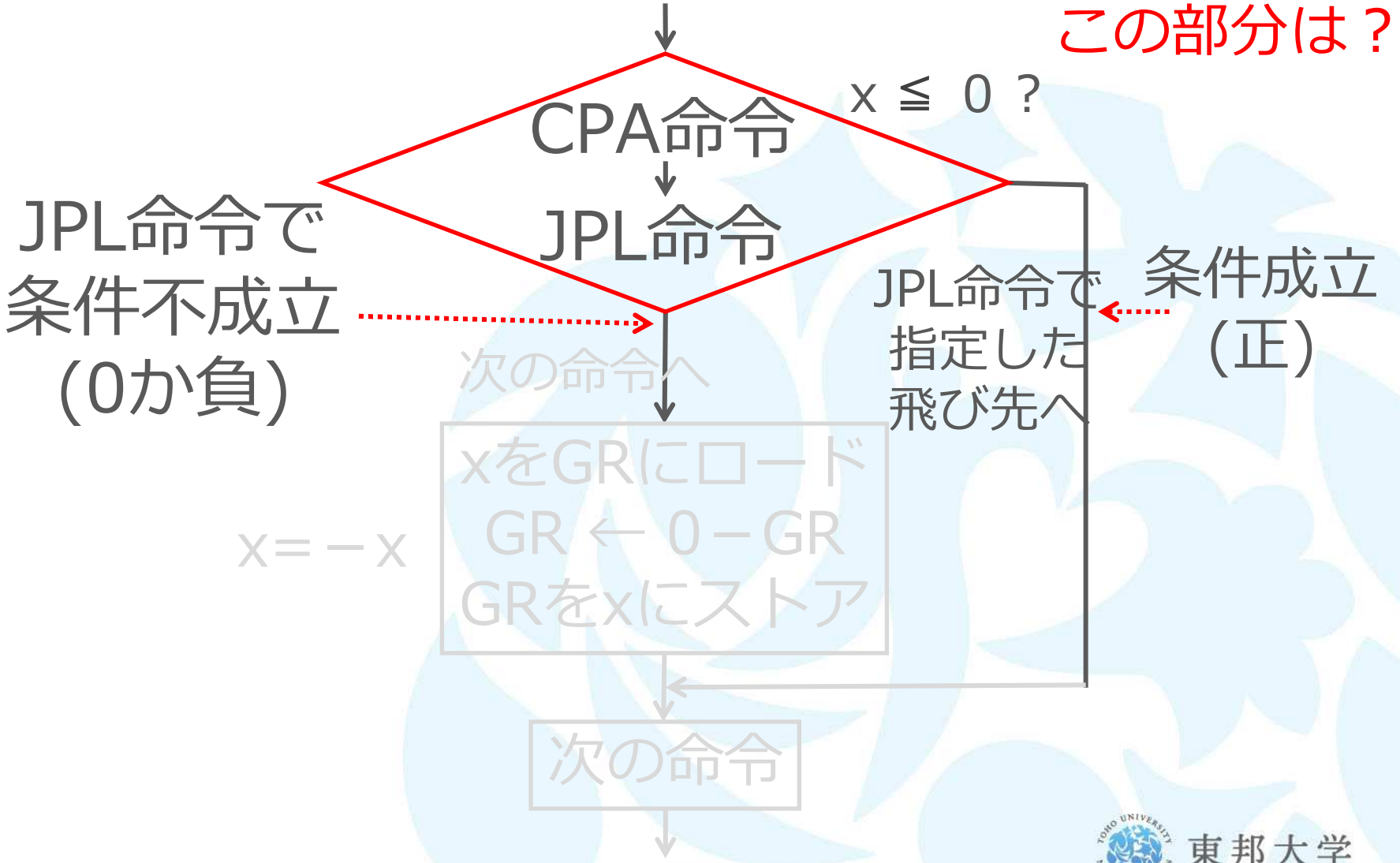
命令に分解したイメージ

まず全体像



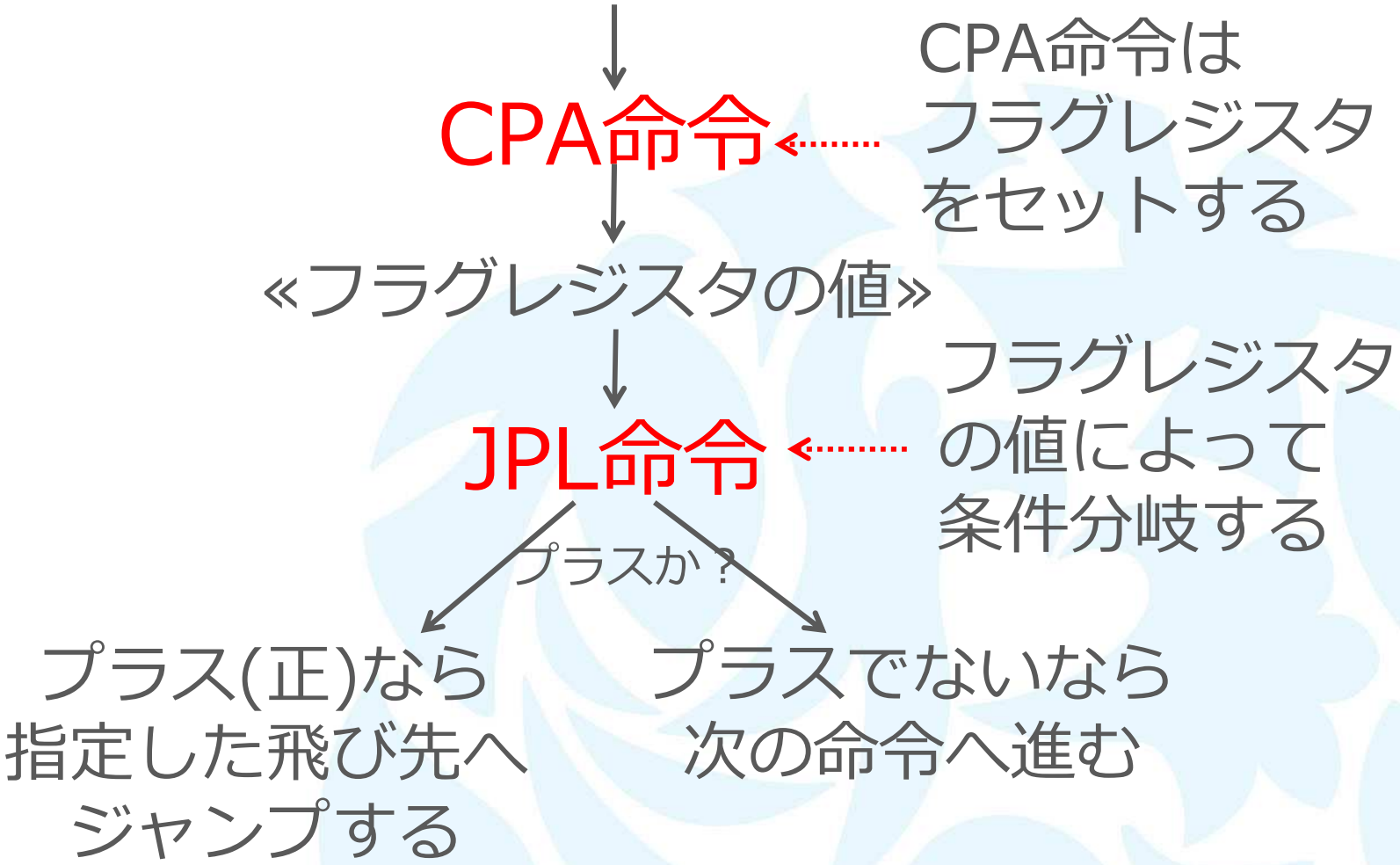
東邦大学

命令に分解したイメージ



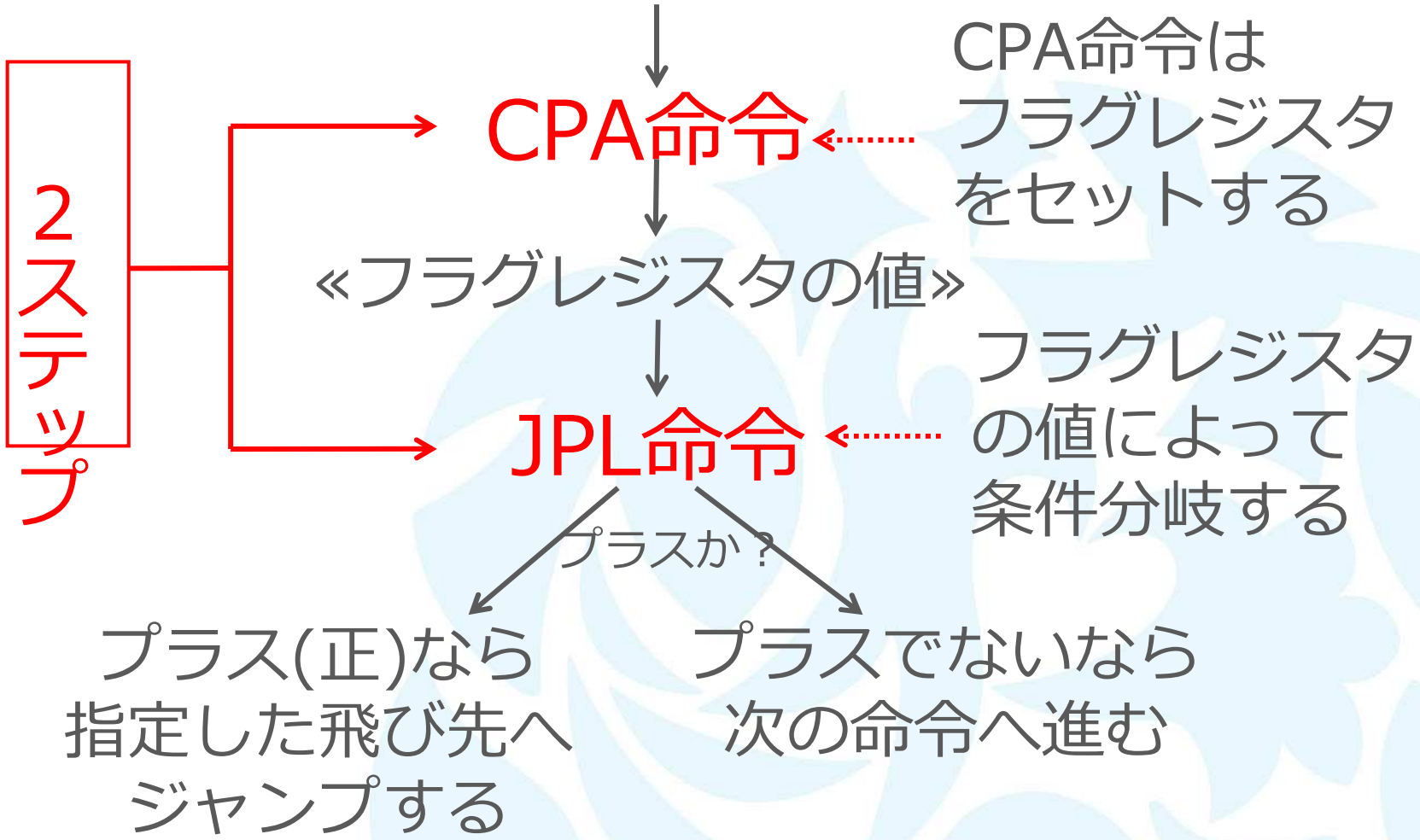
東邦大学

CPA命令＋JPL命令の動き



東邦大学

CPA命令＋JPL命令の動き



CPA命令＋JPL命令の動き

フラグレジスタは３ビットある

CPA命令＋JPL命令の動き

フラグレジスタは３ビットある

ＯＦ	オーバーフロー・フラグ 演算結果が(符号付)16ビットに収まらなくなったとき１になり、それ以外は０になる
ＳＦ	サイン・フラグ 演算結果の符号が負(ビット15が1)のとき１、それ以外は０になる
ＺＦ	ゼロ・フラグ 演算結果がゼロ(全部のビットが0)のとき１、それ以外は0になる

CPA命令＋JPL命令の動き

ＳＦ	サイン・フラグ 演算結果の符号が負(ビット15が1)のとき１、 それ以外は０になる
ＺＦ	ゼロ・フラグ 演算結果がゼロ(全部のビットが0)のとき１、 それ以外は0になる

SF=0とZF=0を組合わせて使える

CPA命令＋JPL命令の動き

ＳＦ	サイン・フラグ 演算結果の符号が負(ビット15が1)のとき１、 それ以外は０になる
ＺＦ	ゼロ・フラグ 演算結果がゼロ(全部のビットが0)のとき１、 それ以外は0になる

SF=0とZF=0を組合わせて使える

(SF=0) ＝ 符号ビットが0 ⇒ X≧0

CPA命令＋JPL命令の動き

ＳＦ	サイン・フラグ 演算結果の符号が負(ビット15が1)のとき１、 それ以外は０になる
ＺＦ	ゼロ・フラグ 演算結果がゼロ(全部のビットが0)のとき１、 それ以外は0になる

SF=0とZF=0を組合わせて使える

(SF=0) ＝ 符号ビットが0　⇒　X≧0

SF=0 かつ ZF=0　とすると　X>0

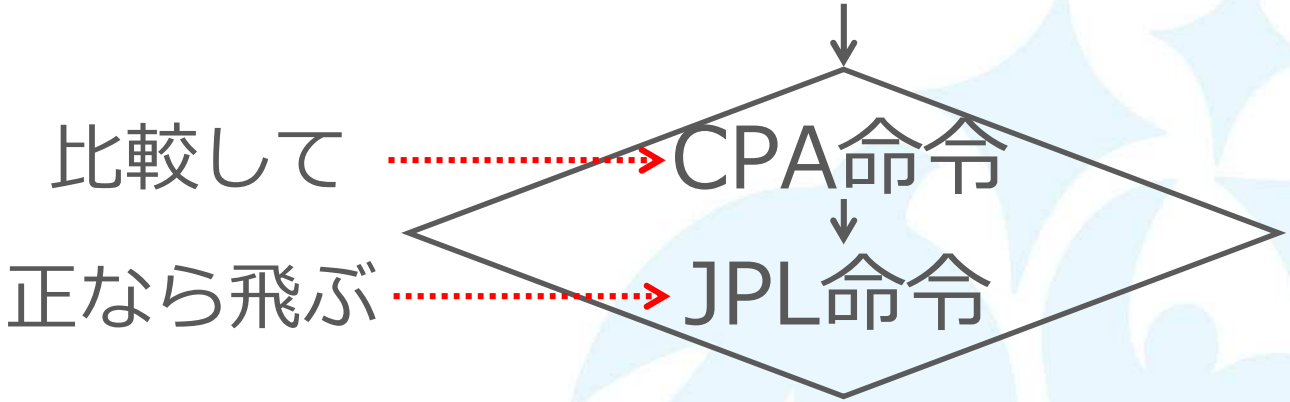
CPA命令＋JPL命令の動き

分岐命令（JPLの仲間）

JPL	正分岐。SF=0(負でない)かつZF=0(零でない)時に 指定先へジャンプ、それ以外は次の命令へ進む
JMI	負分岐。SF=1(負)の時に指定先へジャンプ、 それ以外は次の命令へ進む
JZE	ゼロ分岐。ZF=1(零)の時に指定先へジャンプ、 それ以外は次の命令へ進む
JNZ	非ゼロ分岐。ZF=0(非零)の時に指定先へジャンプ、 それ以外は次の命令へ進む
JUMP	無条件分岐。常に指定先へジャンプ

CPA命令＋JPL命令の動き

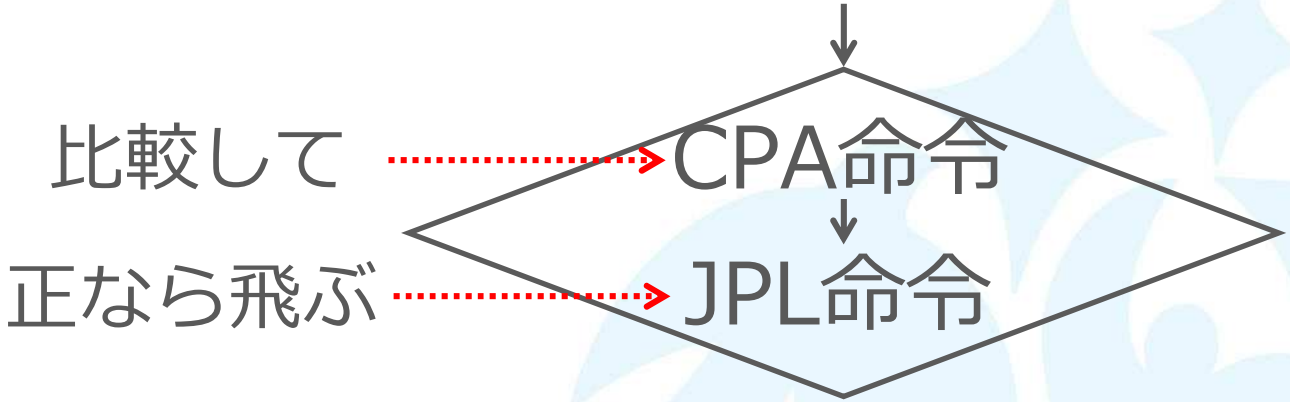
なので CPA ＋ JPL だと



東邦大学

CPA命令＋JPL命令の動き

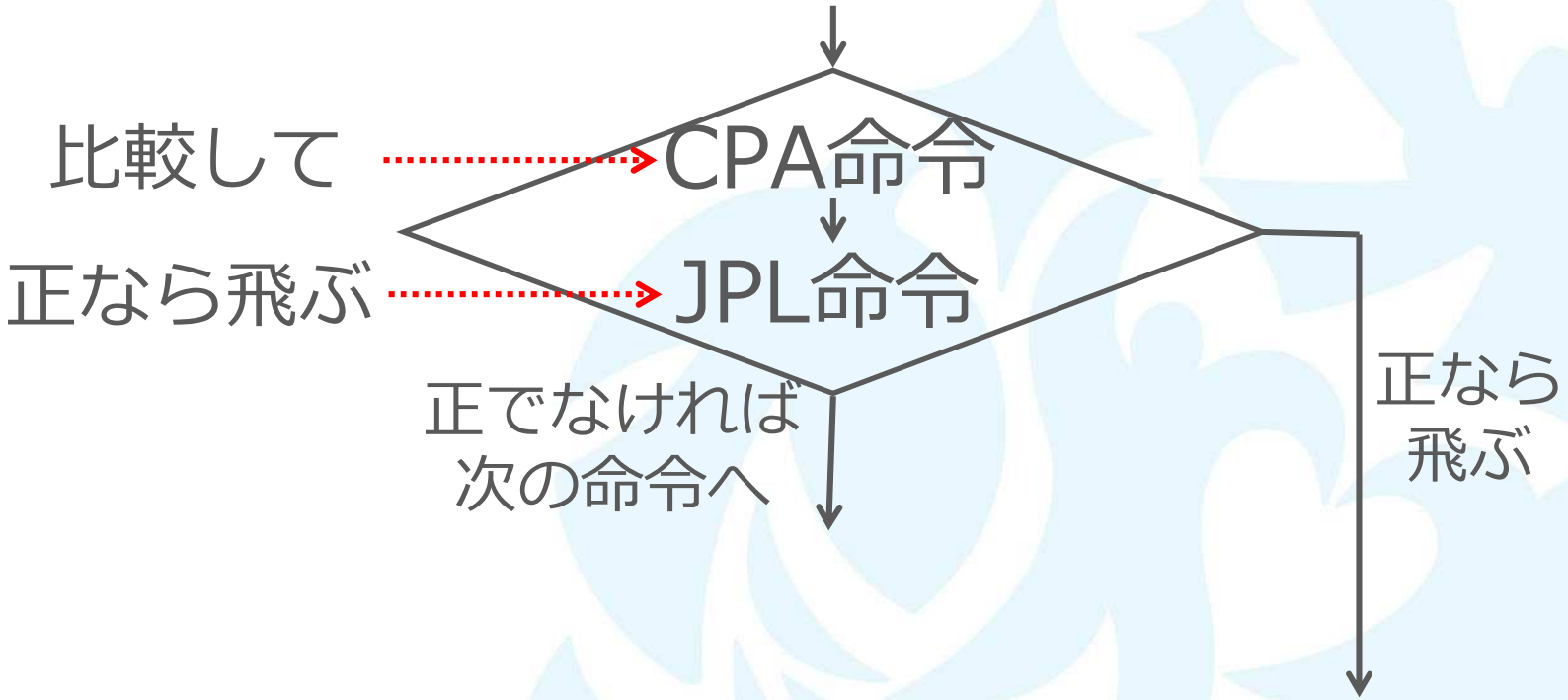
CPA ＋ JPL だと



東邦大学

CPA命令＋JPL命令の動き

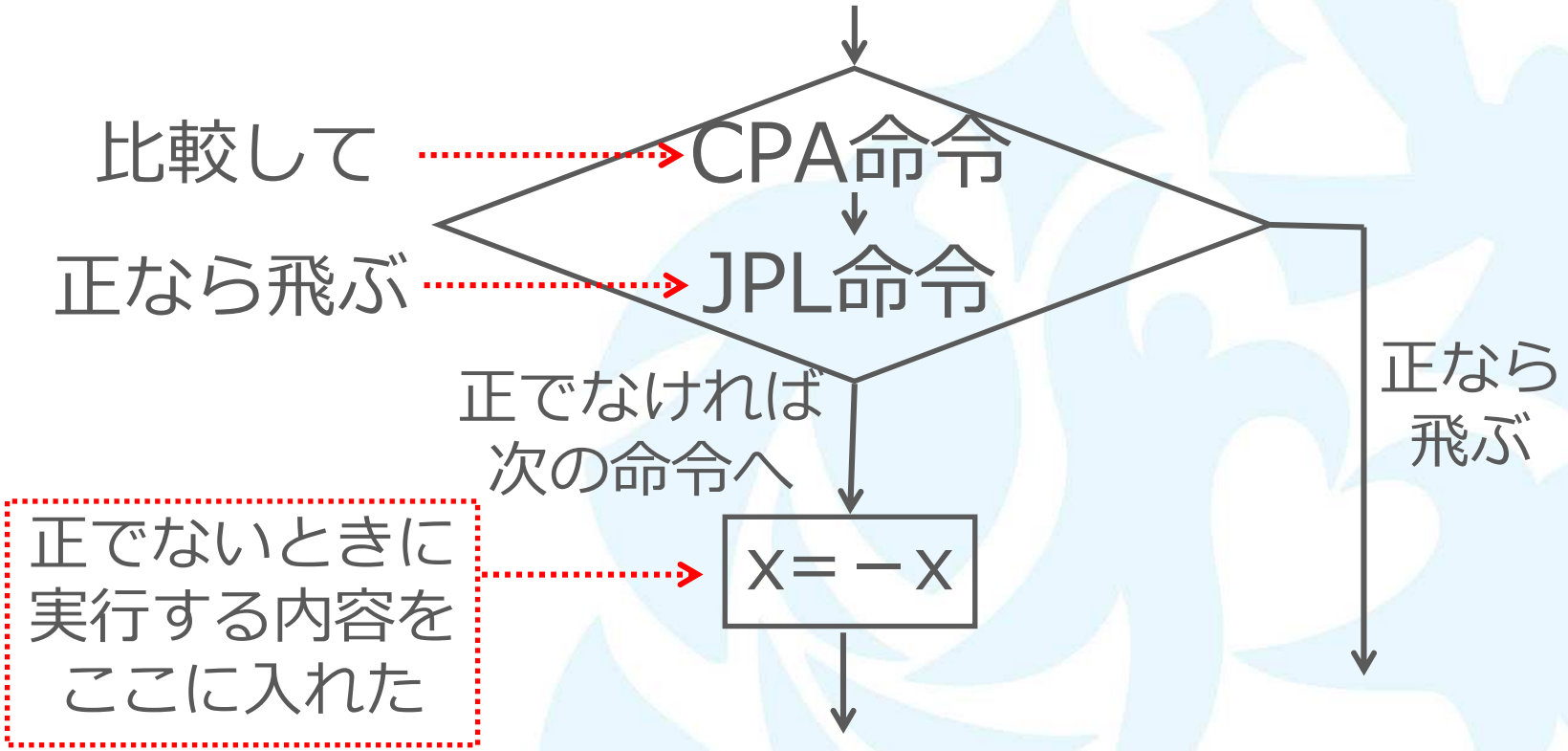
CPA ＋ JPL だと



東邦大学

CPA命令＋JPL命令の動き

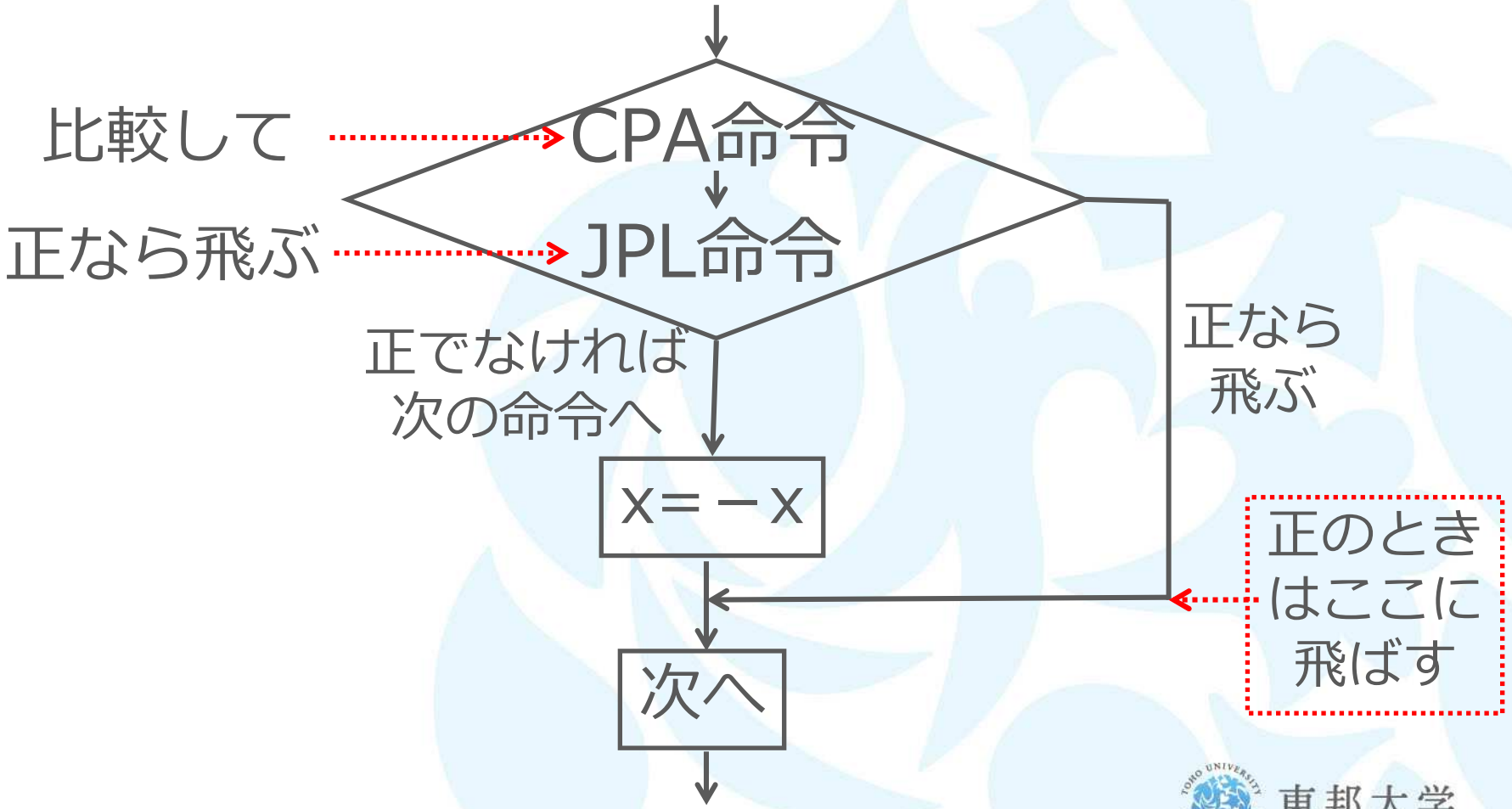
CPA ＋ JPL だと



東邦大学

CPA命令＋JPL命令の動き

CPA ＋ JPL だと

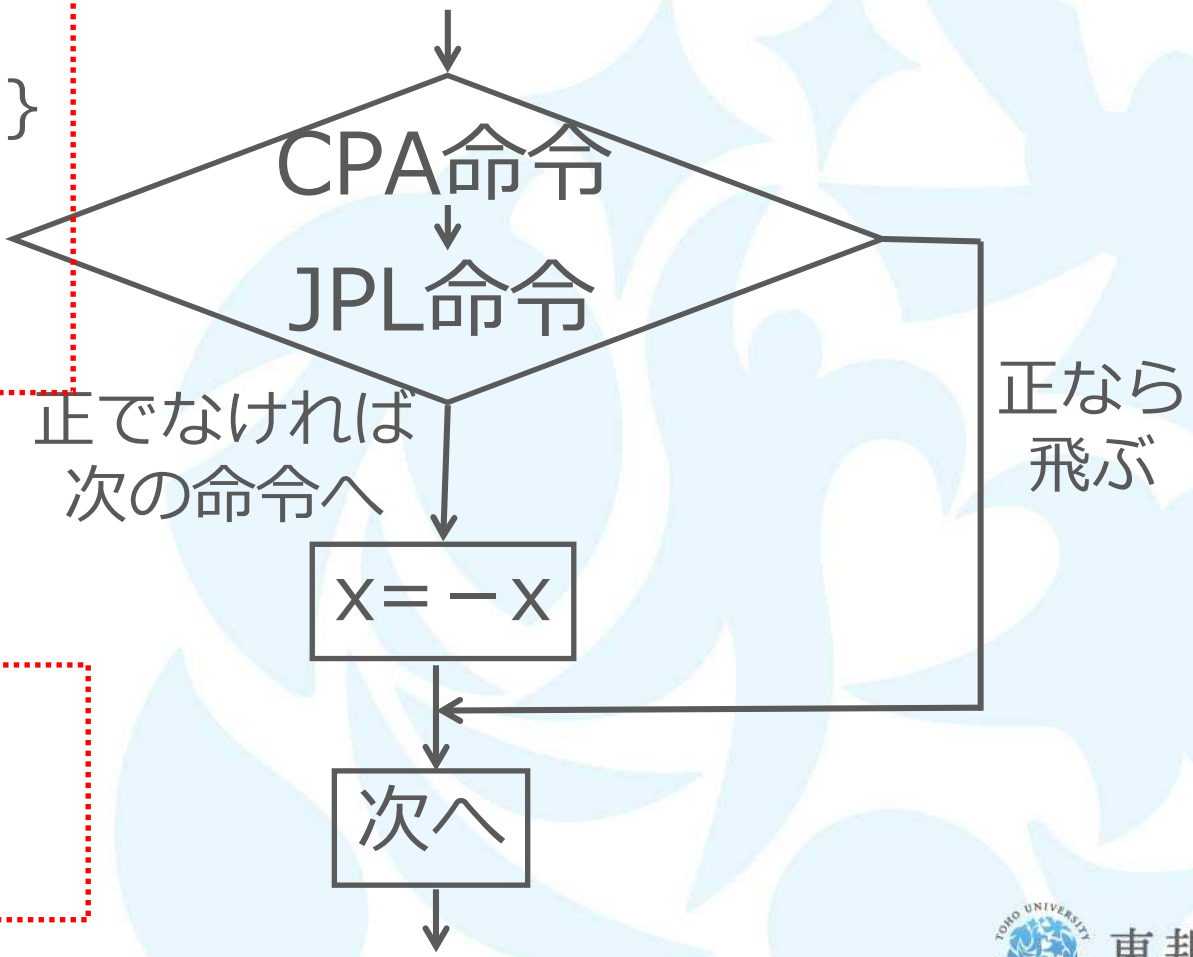


東邦大学

CPA命令＋JPL命令の動き

CPA ＋ JPL だと

```
if (x > 0) {
   何もしない }
else {
   x = − x
}
```



言い換えると

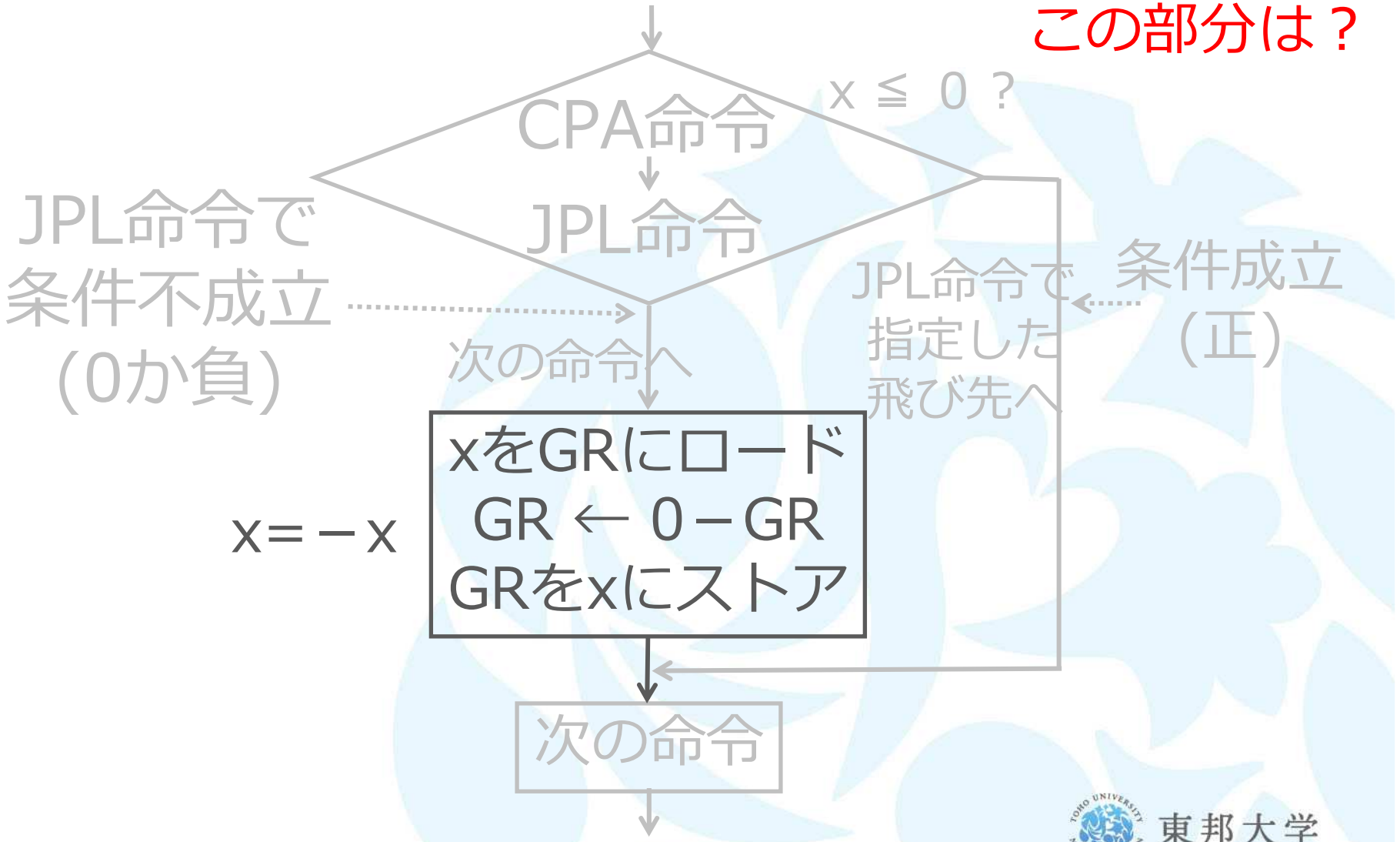
```
if (x ≦ 0) {
   x = − x
}
```

CPA+JPL と if 文の関係がわかりましたか？

東邦大学

命令に分解したイメージ

この部分は？



東邦大学

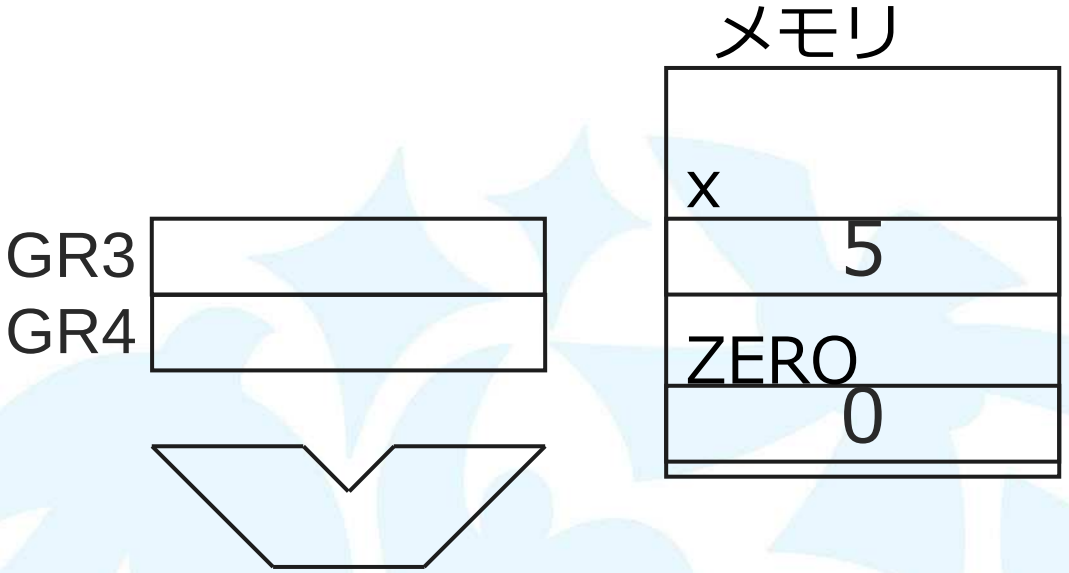
命令に分解したイメージ

x＝－x

xをGRにロード

GR ← 0－GR

GRをxにストア



「計算式と代入」と同じ原理

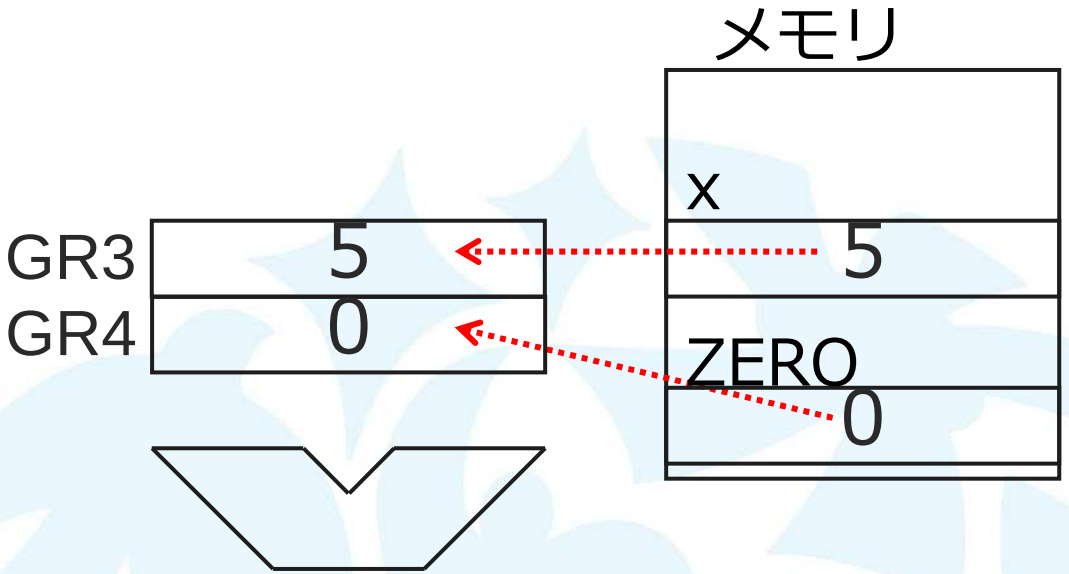
命令に分解したイメージ

x＝－x

xをGRにロード

GR ← 0－GR

GRをxにストア



東邦大学

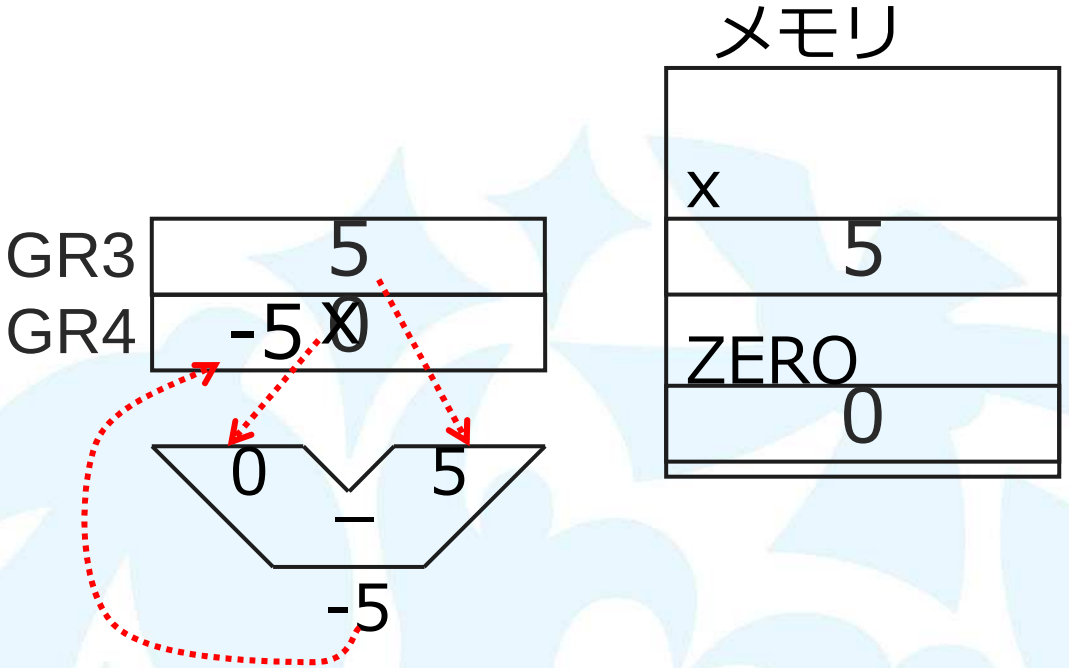
命令に分解したイメージ

x＝－x

xをGRにロード

GR ← 0－GR

GRをxにストア



東邦大学

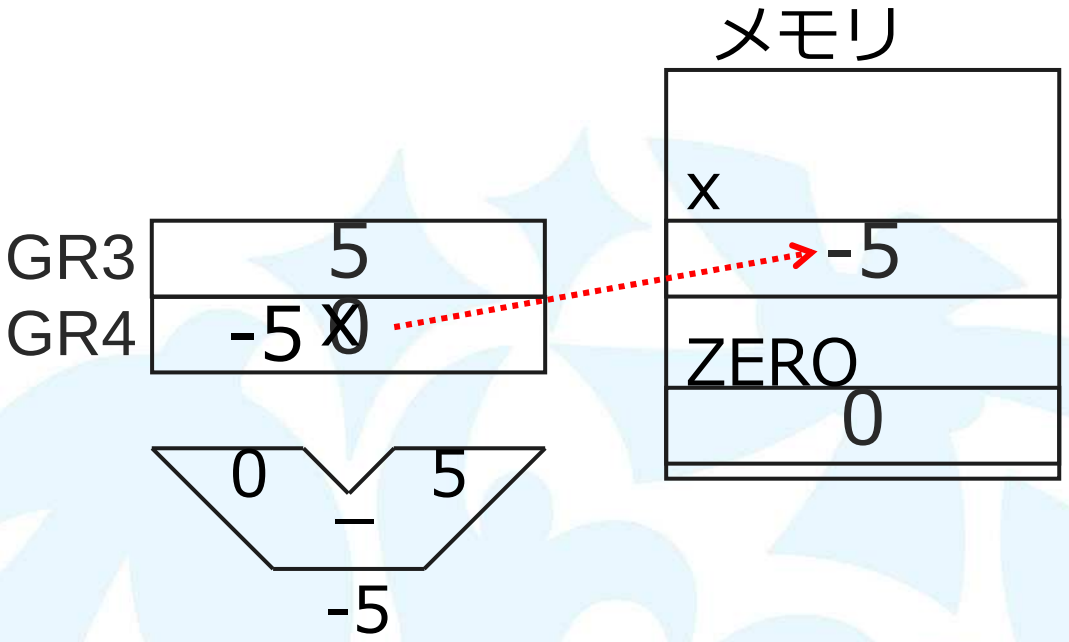
命令に分解したイメージ

x＝－x

xをGRにロード

GR ← 0－GR

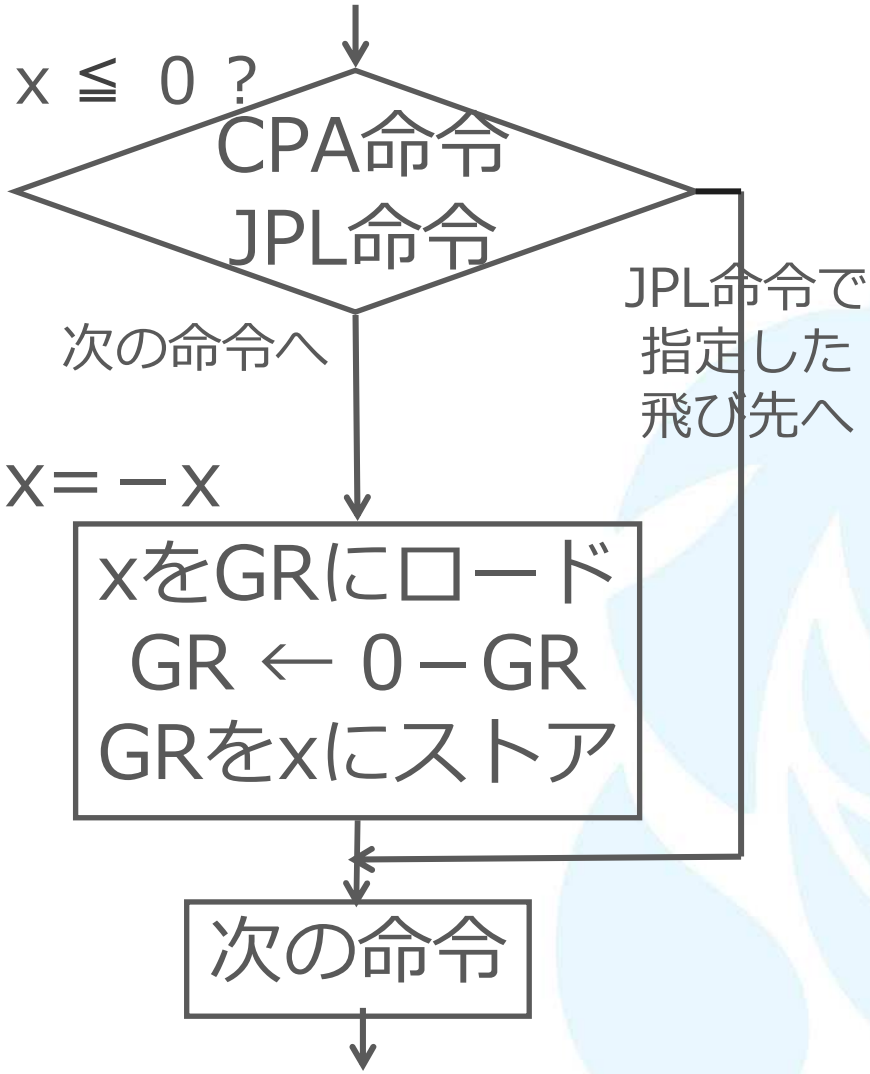
GRをxにストア



東邦大学

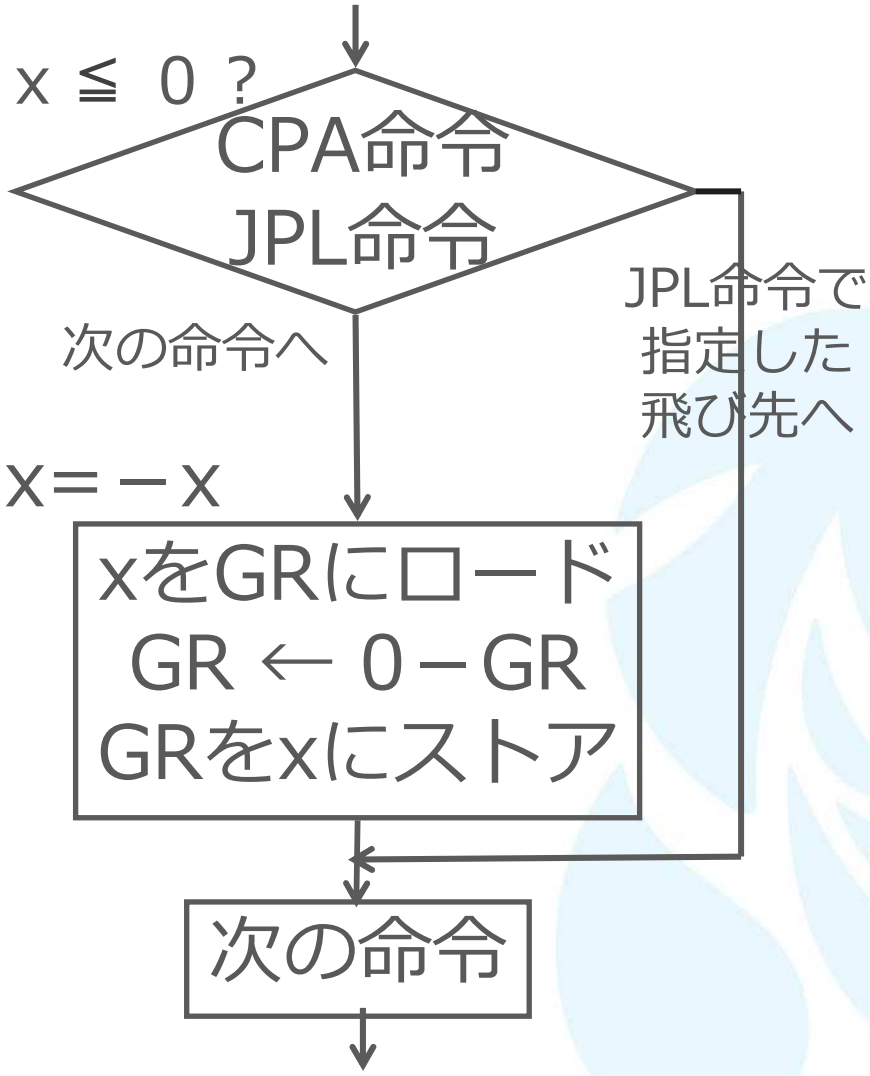
プログラムとしては

プログラムとしては



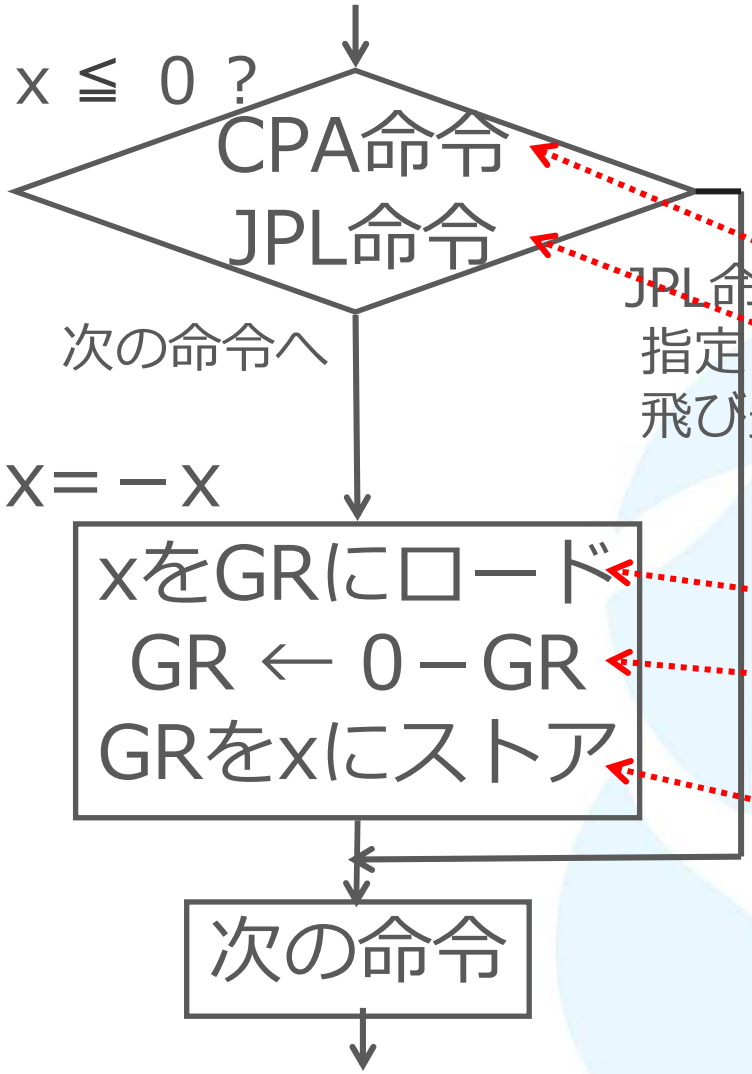
東邦大学

プログラムとしては



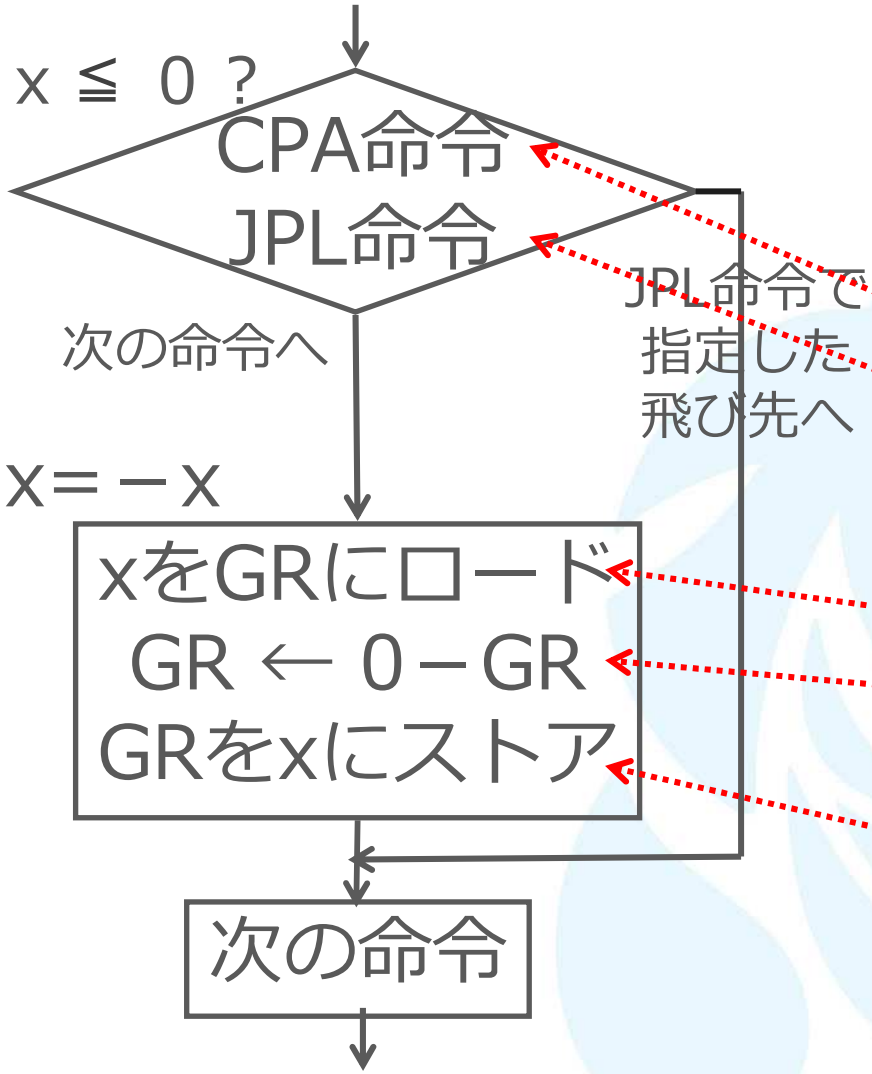

```
LD   GR4, ZERO     GR4←0
LD   GR3, X        GR3←x
CPA  GR3, GR4      比較
JPL  L1            x>0なら
                   L1へ飛ぶ
LD   GR4, ZERO     GR4←0
LD   GR3, X        GR3←x
SUBA GR4, GR3      GR4-GR3
                    →GR4
ST   GR4, X        X←GR4

L1: 次の命令
```

プログラムとしては




```
LD   GR4, ZERO     GR4←0
LD   GR3, X        GR3←x
CPA  GR3, GR4      比較
JPL  L1            x>0なら
                   L1へ飛ぶ
LD   GR4, ZERO     GR4←0
LD   GR3, X        GR3←x
SUBA GR4, GR3      GR4-GR3
                    →GR4
ST   GR4, X        X←GR4

L1: 次の命令
```

プログラムとしては



比較のためのレジスタ設定

```
LD   GR4, ZERO     GR4←0
LD   GR3, X        GR3←x
CPA  GR3, GR4      比較
JPL  L1            x>0なら
                   L1へ飛ぶ
```

引き算のためのレジスタ設定

```
LD   GR4, ZERO     GR4←0
LD   GR3, X        GR3←x
SUBA GR4, GR3      GR4-GR3
                   →GR4
ST   GR4, X        X←GR4
```

L1: 次の命令

命令に変換するとどうなるか
わかりましたか？

東邦大学

もう少しいろいろな例を見てみましょう

もう少しいろいろな例を見てみましょう

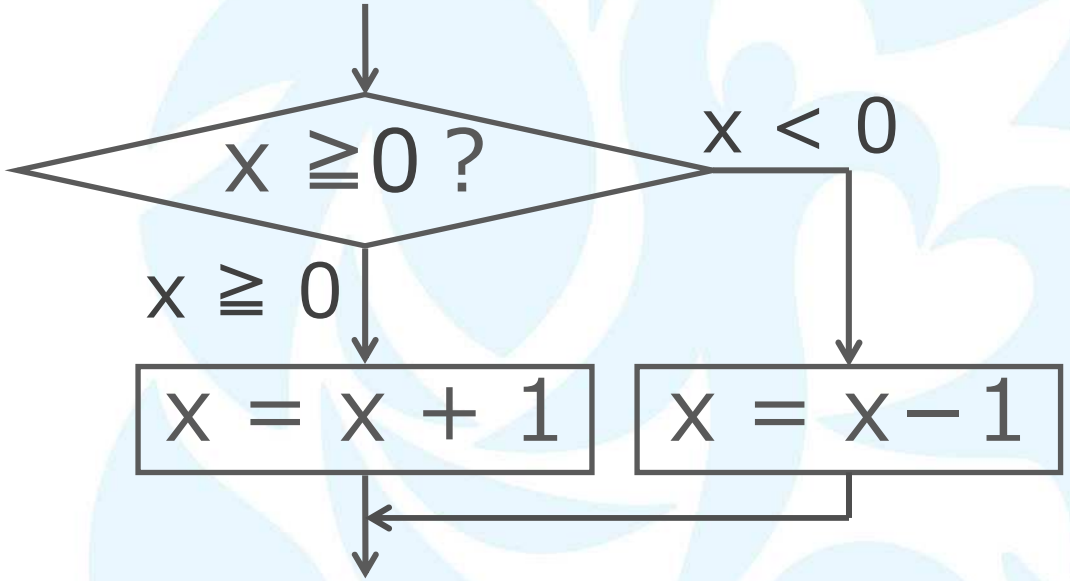
if (x>0)　x=x+1

else x=x－1

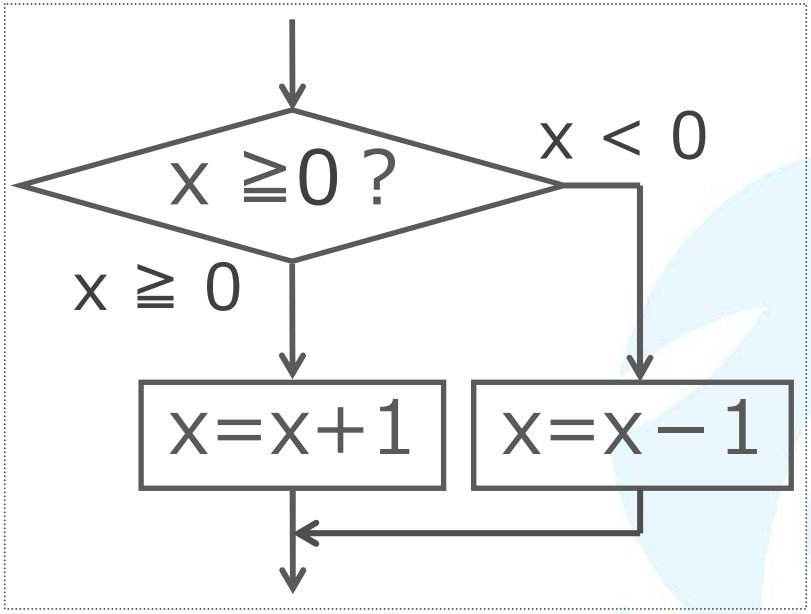
もう少しいろいろな例を見てみましょう

if (x≧0)　x=x+1
else x=x－1

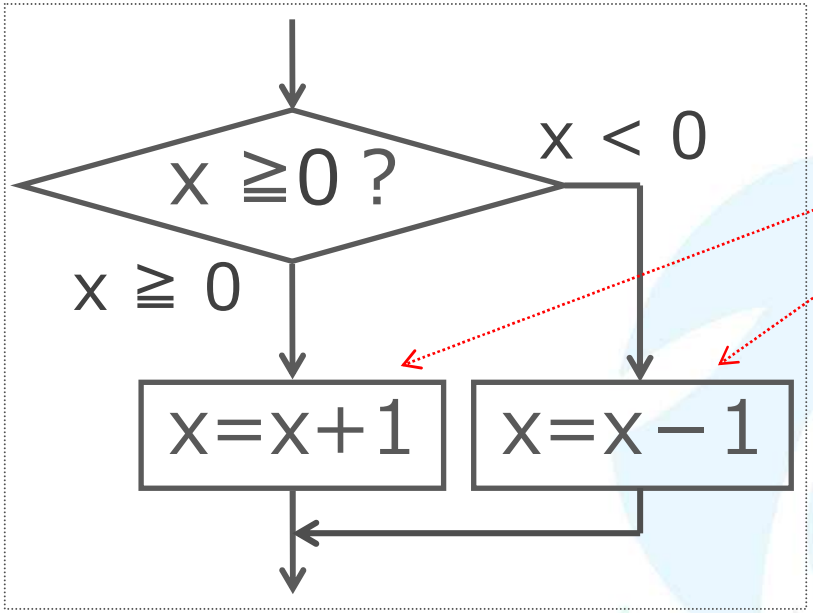
流れ図を描くと



これを命令の流れに書き直すには



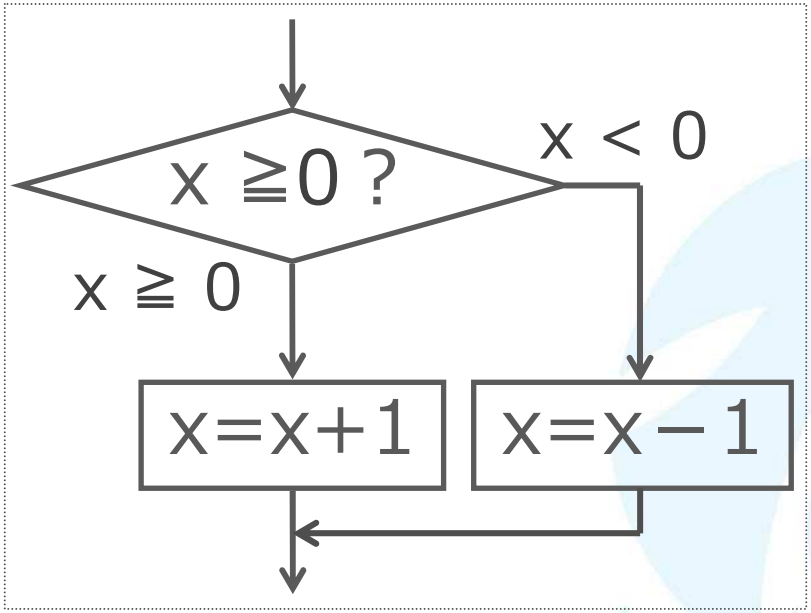
これを命令の流れに書き直すには



横に２列に
なる所がある

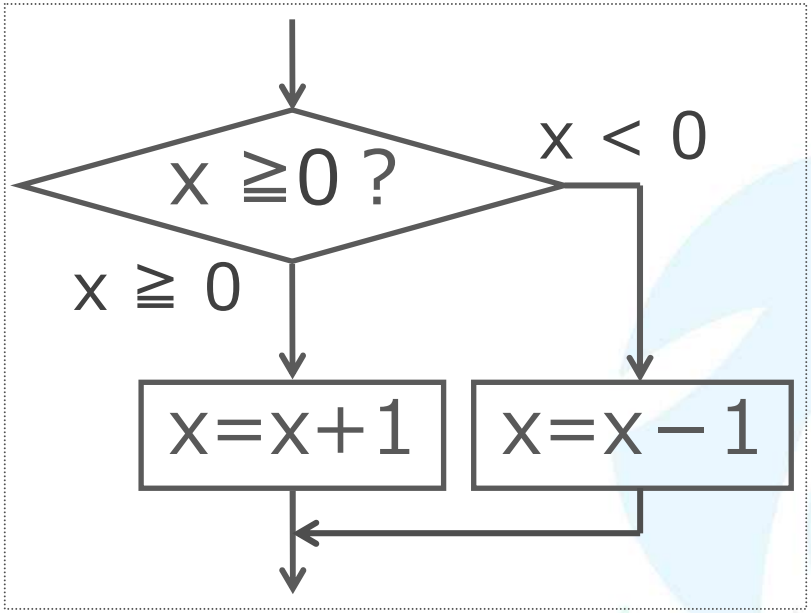
東邦大学

これを命令の流れに書き直すには

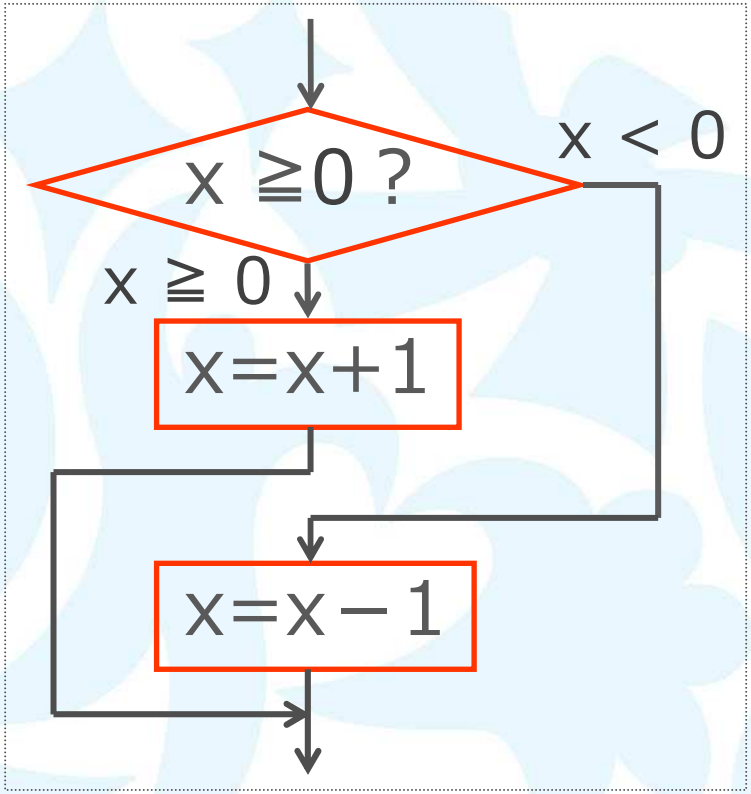


横に２列に
なる所がある
↓
１列の流れに
書き換える

これを命令の流れに書き直すには

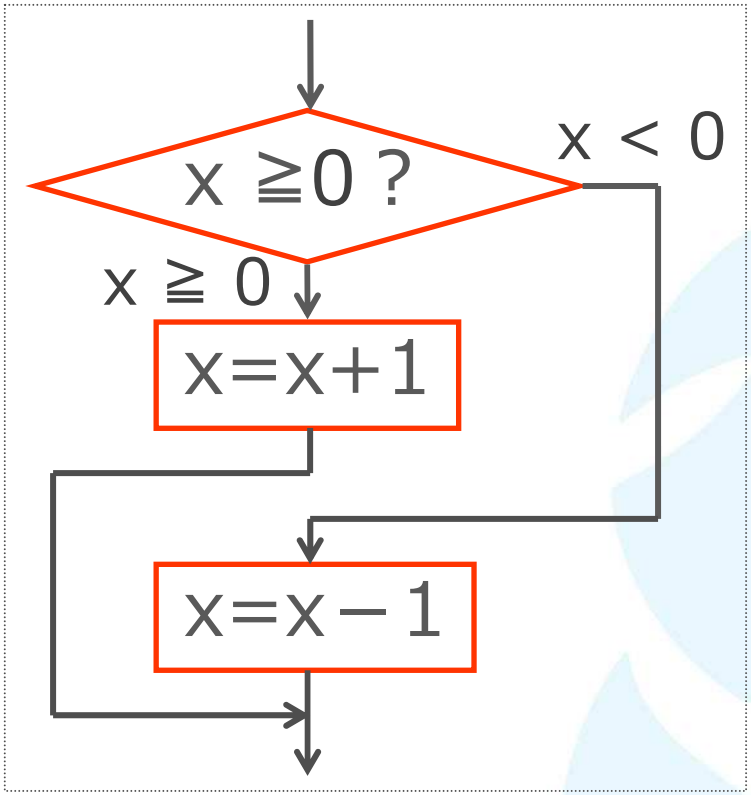


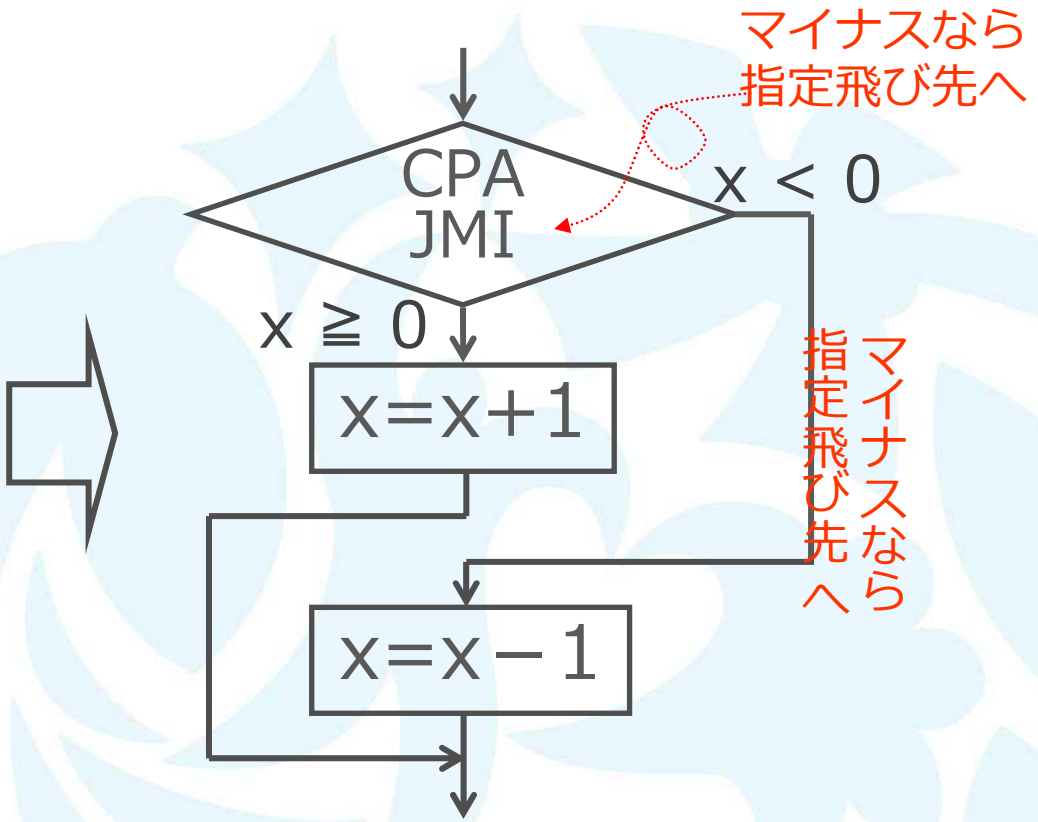
１列の流れ～横に並んでいない



東邦大学

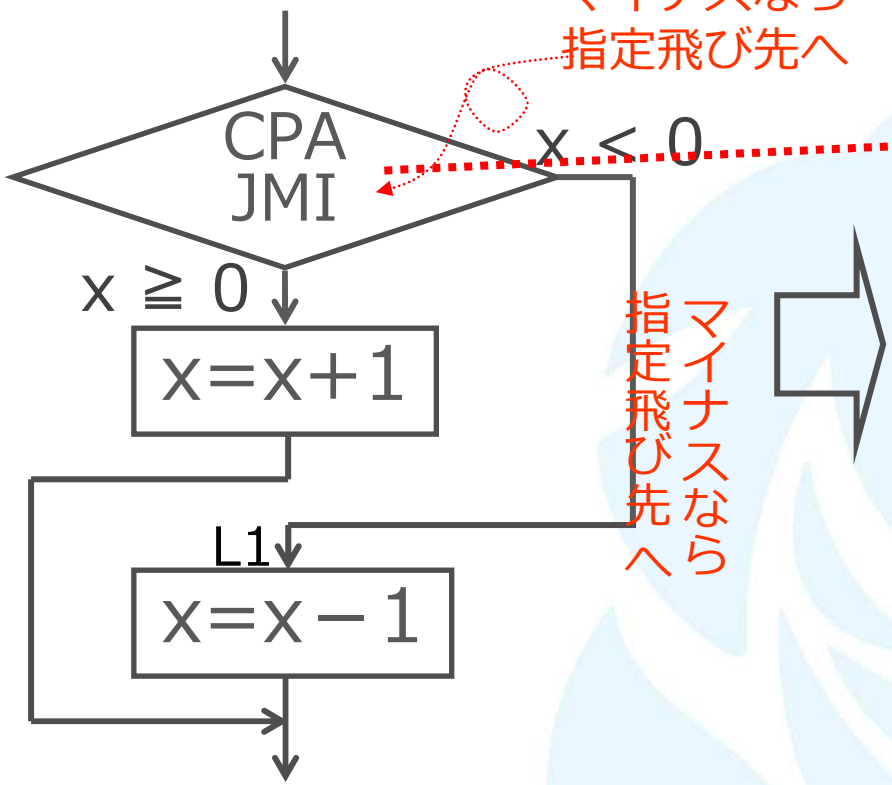
これを命令の流れに書き直すには





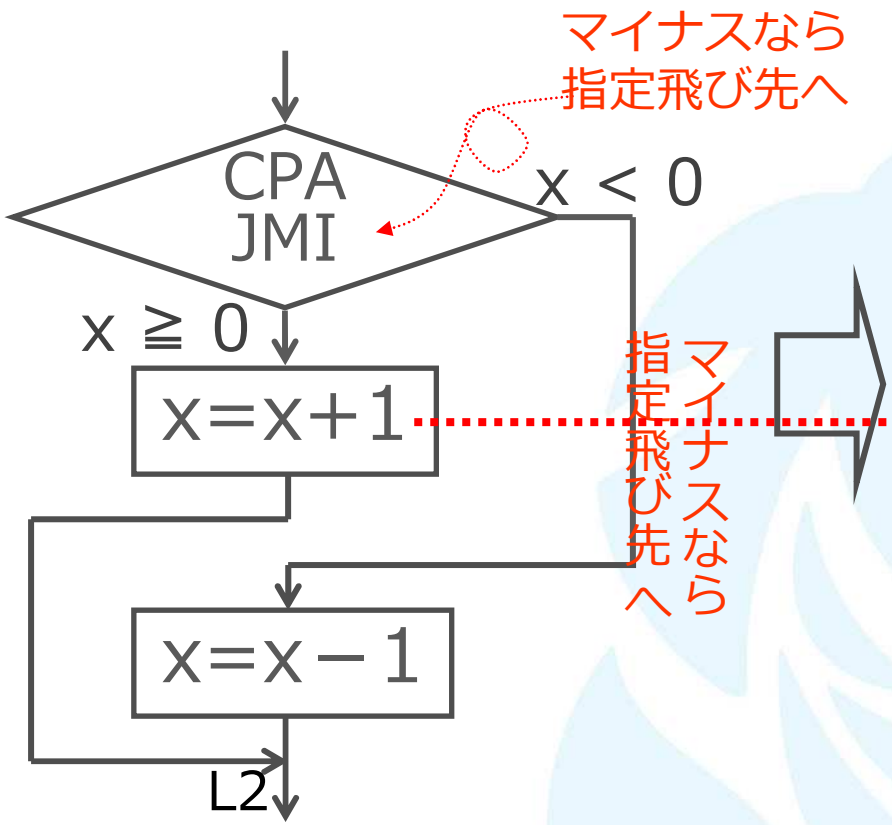
東邦大学

これを命令の流れに書き直すには



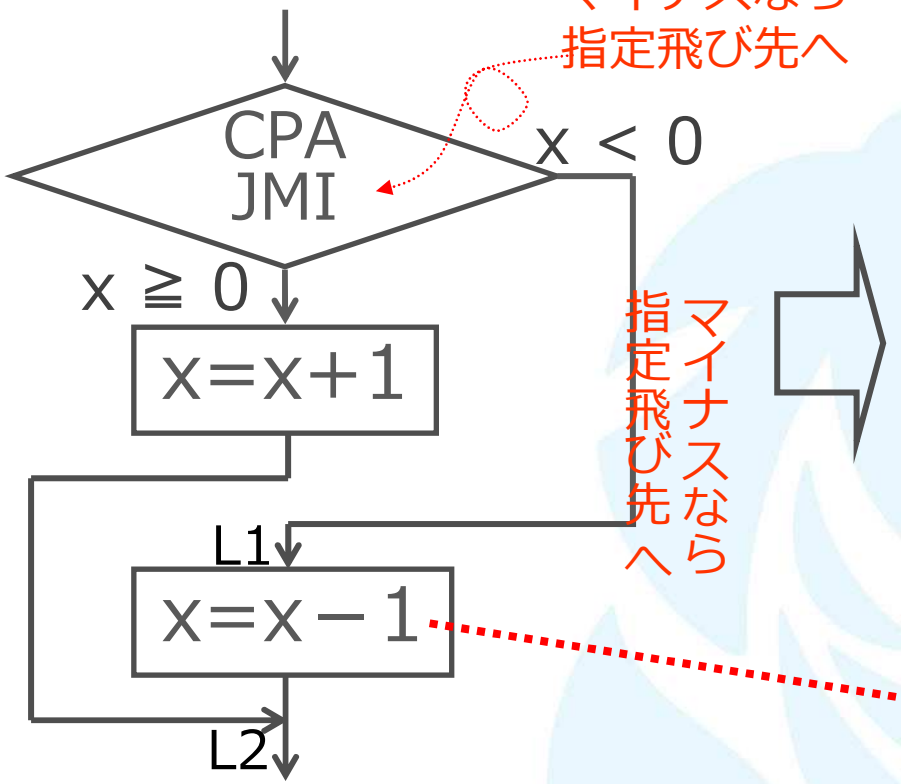

```
LD    GR3, ZERO
LD    GR4, X
CPA   GR4, GR3    (x-0)
JMI   L1          負→L1

      LD    GR3, ONE
      ADDA  GR4, GR3   (x+1)
      ST    GR4, X
      JUMP  L2

L1    LD    GR3, ONE
      SUBA  GR4, GR3   (x-1)
      ST    GR4, X

L2    次の命令
```

これを命令の流れに書き直すには



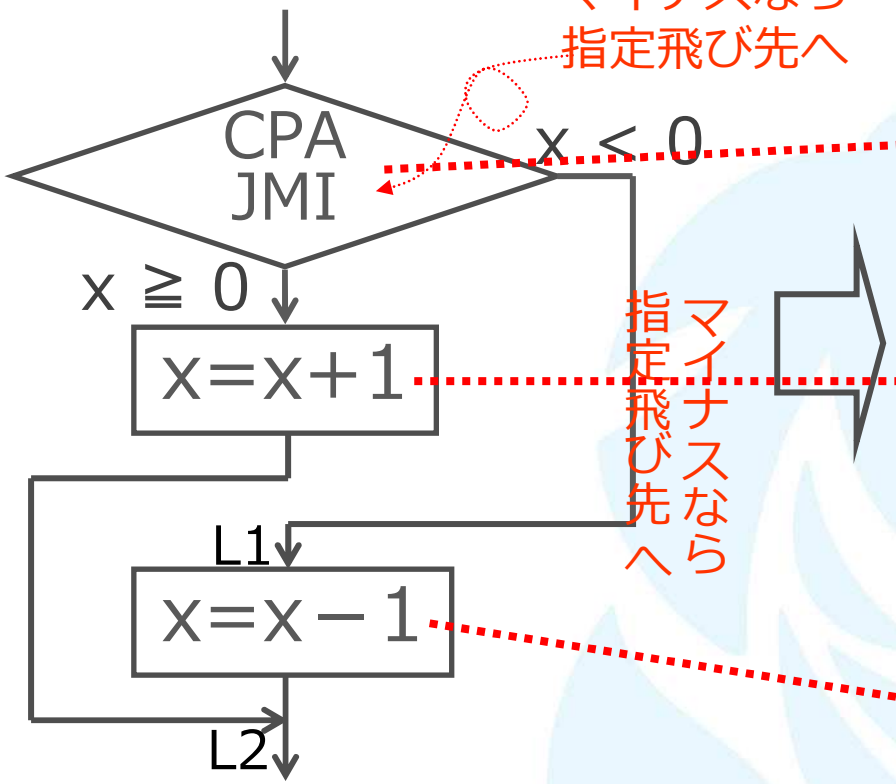

```
LD    GR3, ZERO
LD    GR4, X
CPA   GR4, GR3    (x-0)
JMI   L1          負→L1

LD    GR3, ONE
ADDA  GR4, GR3    (x+1)
ST    GR4, X
JUMP  L2

L1 LD   GR3, ONE
   SUBA GR4, GR3    (x-1)
   ST   GR4, X

L2 次の命令
```

これを命令の流れに書き直すには




```
LD    GR3, ZERO
LD    GR4, X
CPA   GR4, GR3    (x-0)
JMI   L1          負→L1

LD    GR3, ONE
ADDA  GR4, GR3    (x+1)
ST    GR4, X
JUMP  L2

L1 LD   GR3, ONE
   SUBA GR4, GR3    (x-1)
   ST   GR4, X

L2 次の命令
```

これを命令の流れに書き直すには




```
LD    GR3, ZERO
LD    GR4, X
CPA   GR4, GR3    (x-0)
JMI   L1          負→L1

LD    GR3, ONE
ADDA  GR4, GR3    (x+1)
ST    GR4, X
JUMP  L2

L1 LD   GR3, ONE
SUBA  GR4, GR3    (x-1)
ST    GR4, X

L2 次の命令
```

<完成>

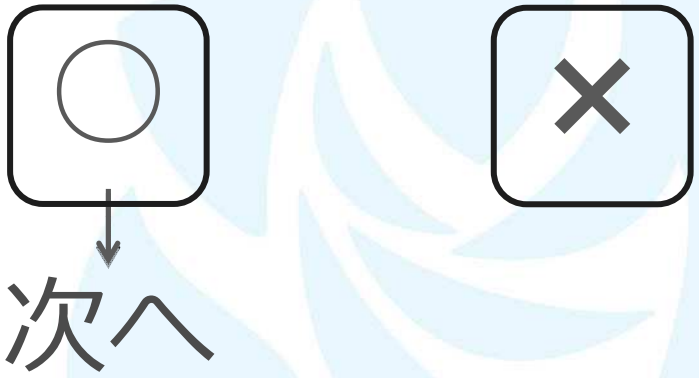
条件分岐をまとめると

条件分岐は　CPA+JPL (や CPA+JMI) で作ることが出来る

条件 (> とか ≧ とか < とか ≦ など) は JPL, JMI, JZE, JNE を選んで作る

左右２つの経路があるときは、縦に並べて JUMP命令でつなぐ必要がある

if 文（条件分岐）から機械命令へ変換の仕方がわかりましたか？



では、演習問題で試してみましょう

東邦大学

演習問題

```
if ( x>0 )   x = x + x
  else  x = 0
```

演習問題

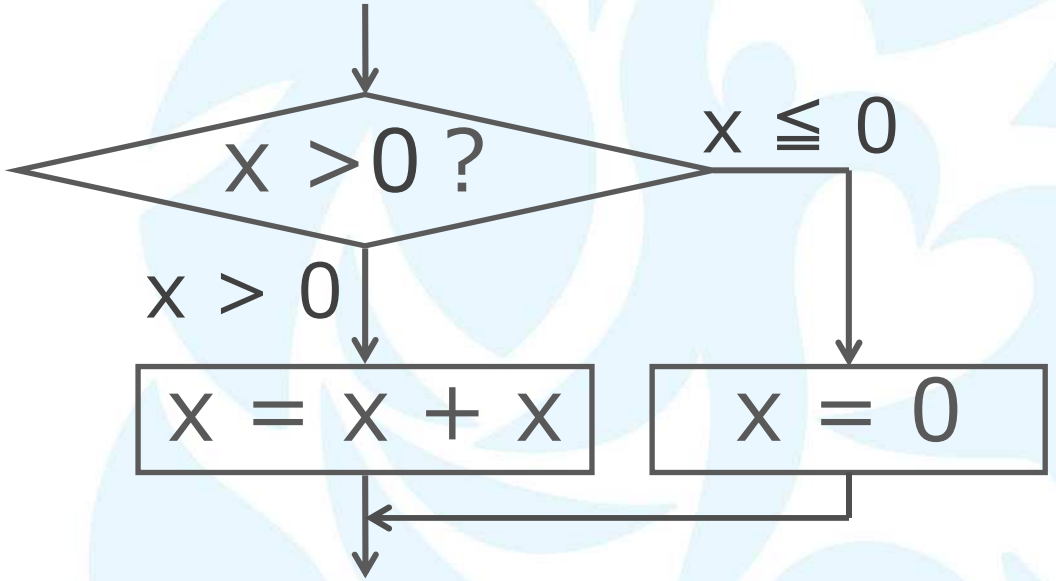
```
if ( x>0 )   x = x + x
  else  x = 0
```

流れ図を描くと

演習問題

```
if ( x>0 )   x = x + x
  else  x = 0
```

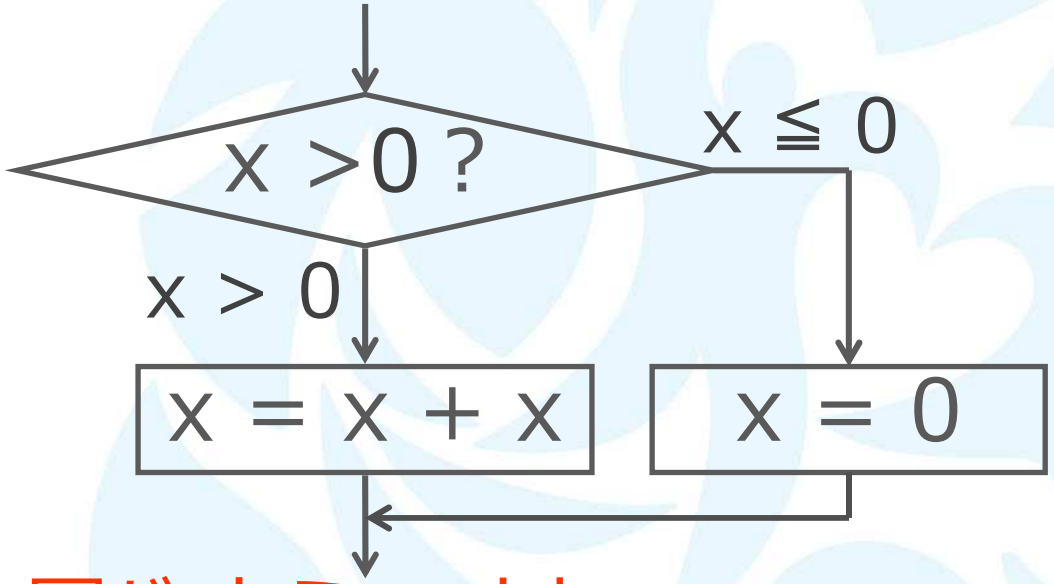
流れ図を描くと



演習問題

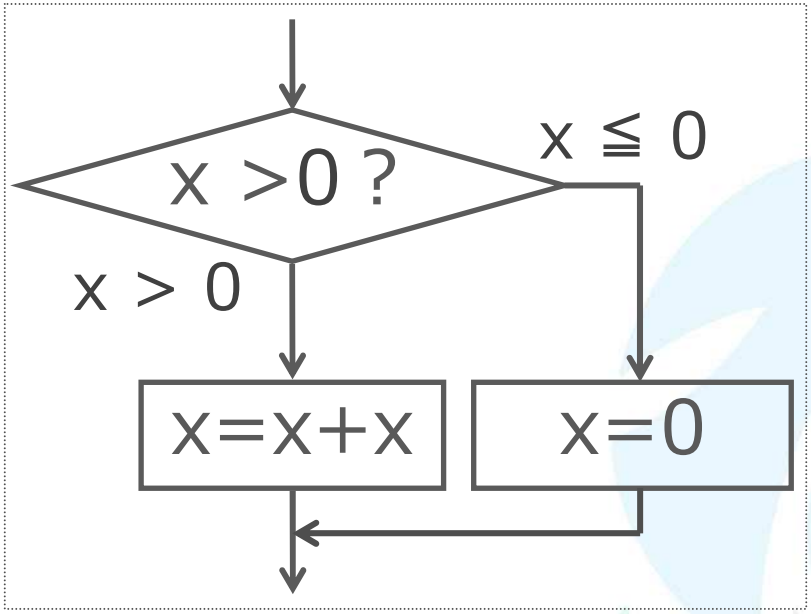
```
if ( x>0 )   x = x + x
  else  x = 0
```

流れ図を描くと

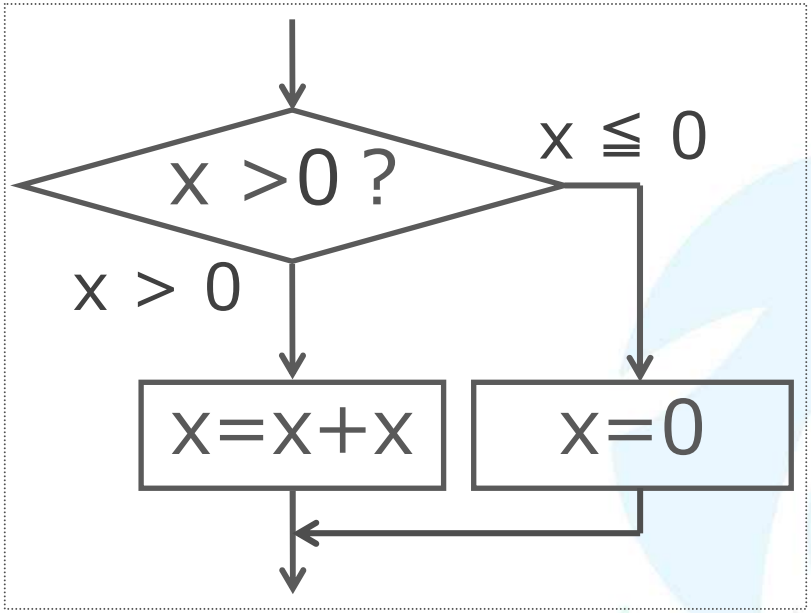


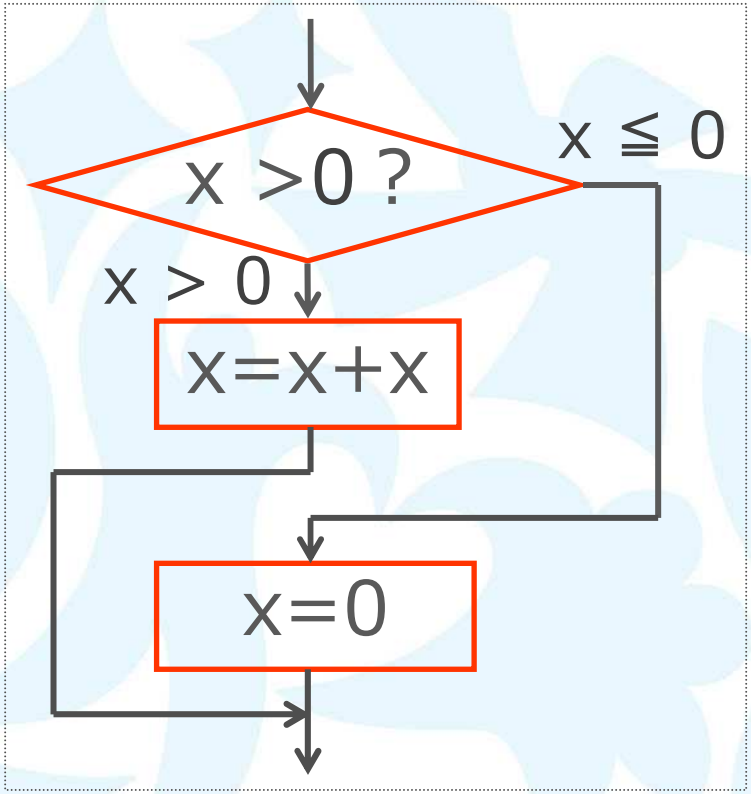
前の例題と同じようですね

これを縦一列に書き直すと



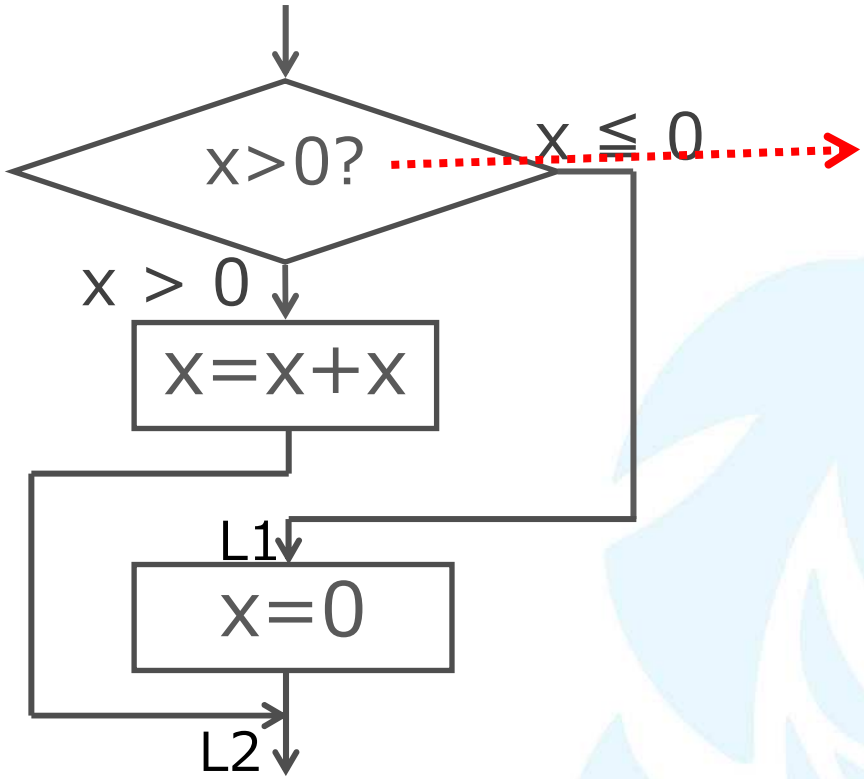
これを縦一列に書き直すと





ここまで前の例題と同じやり方です

これを命令の流れに書き直すには

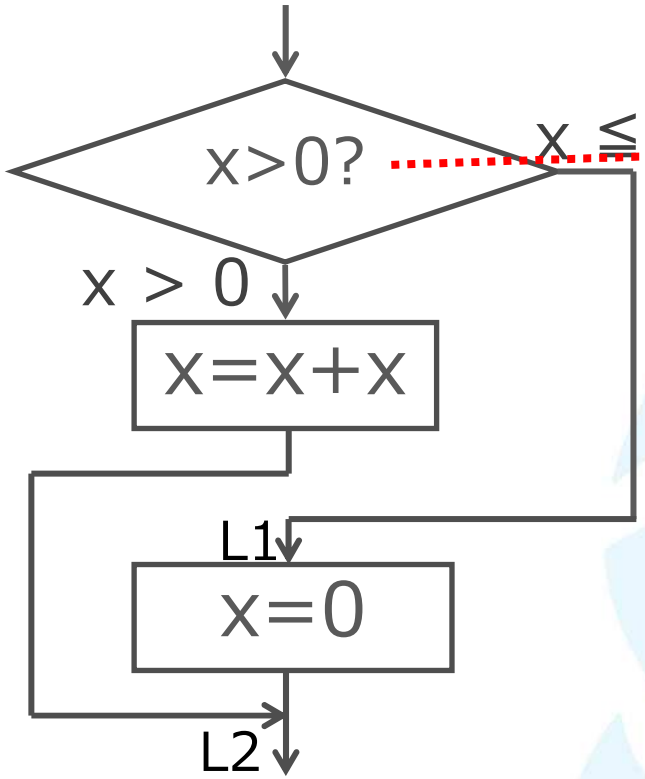



```
CPA   GR4, GR3    (x-0)
JMI   L1                負→L1
```

でいいだろうか？

x=0の時を考えてみよう

これを命令の流れに書き直すには




```
CPA   GR4, GR3    (x-0)
JMI   L1                負→L1
```

でいいだろうか？

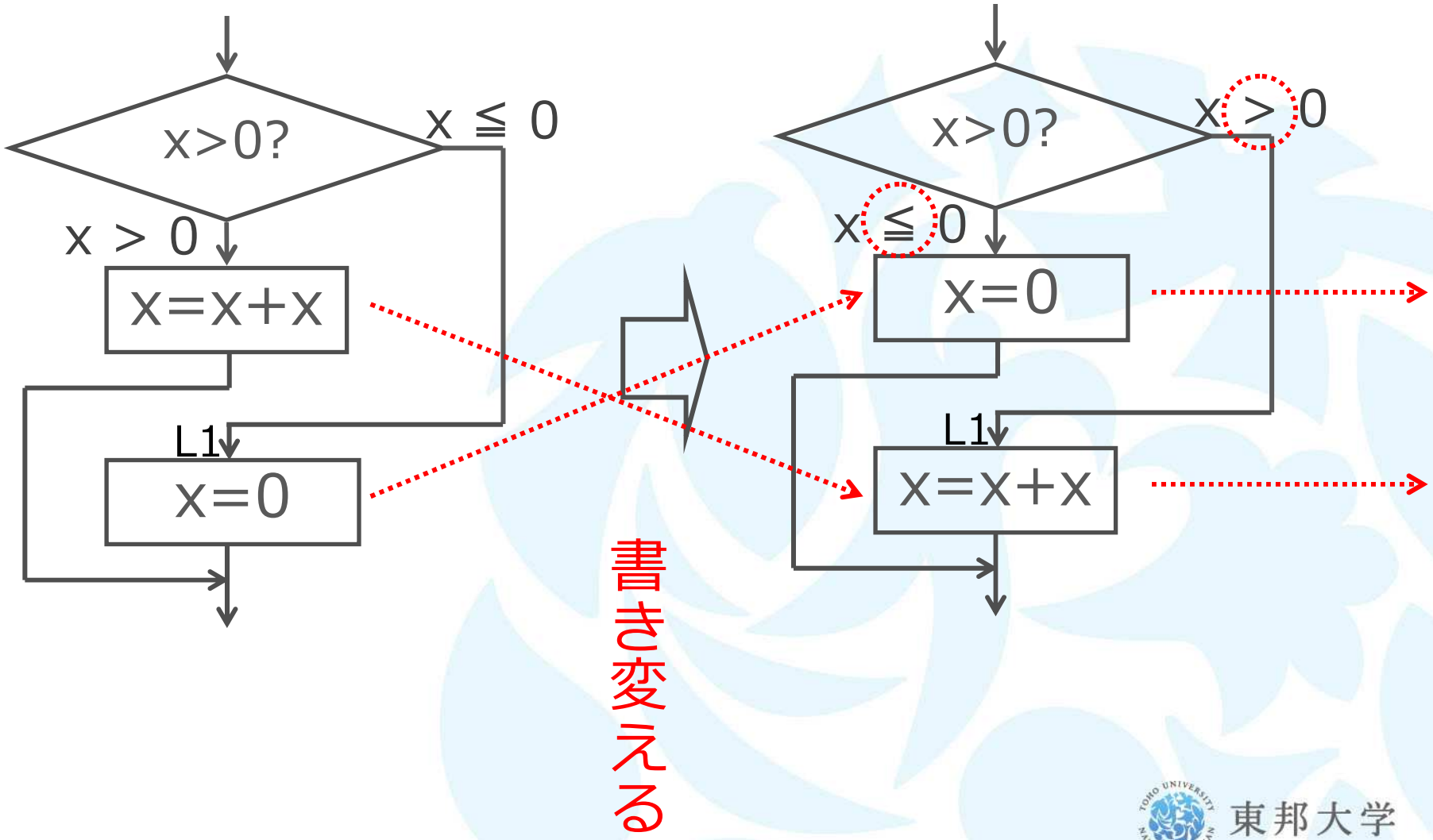
x=0 の時を考えてみよう

流れ図ではL1へ飛ぶ

JMIだと0の時は
ジャンプしない

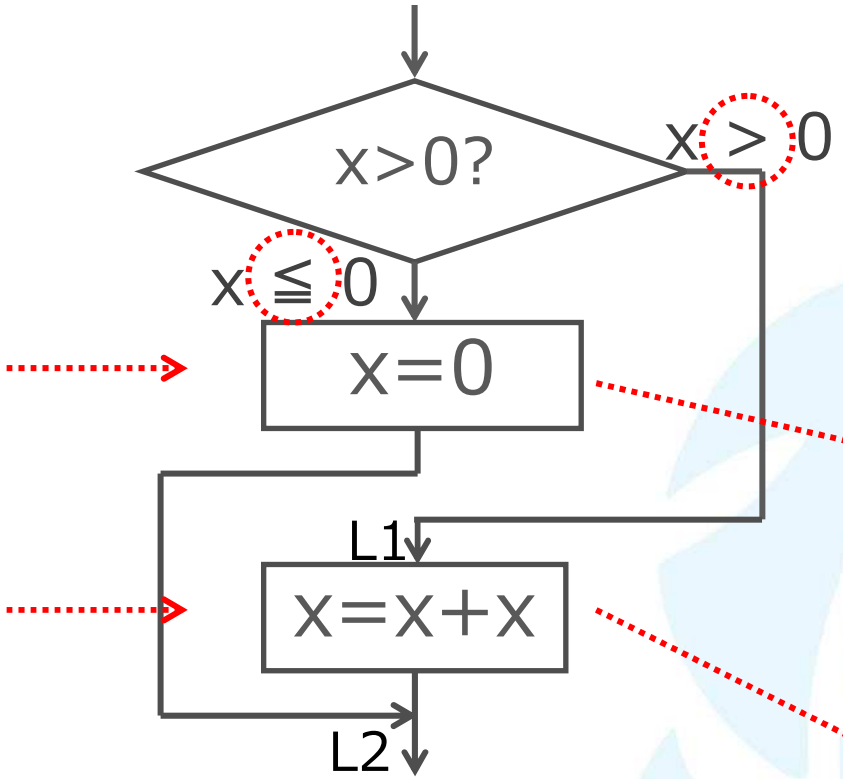
流れ図と違う

方法１）判定条件をひっくり返す



東邦大学

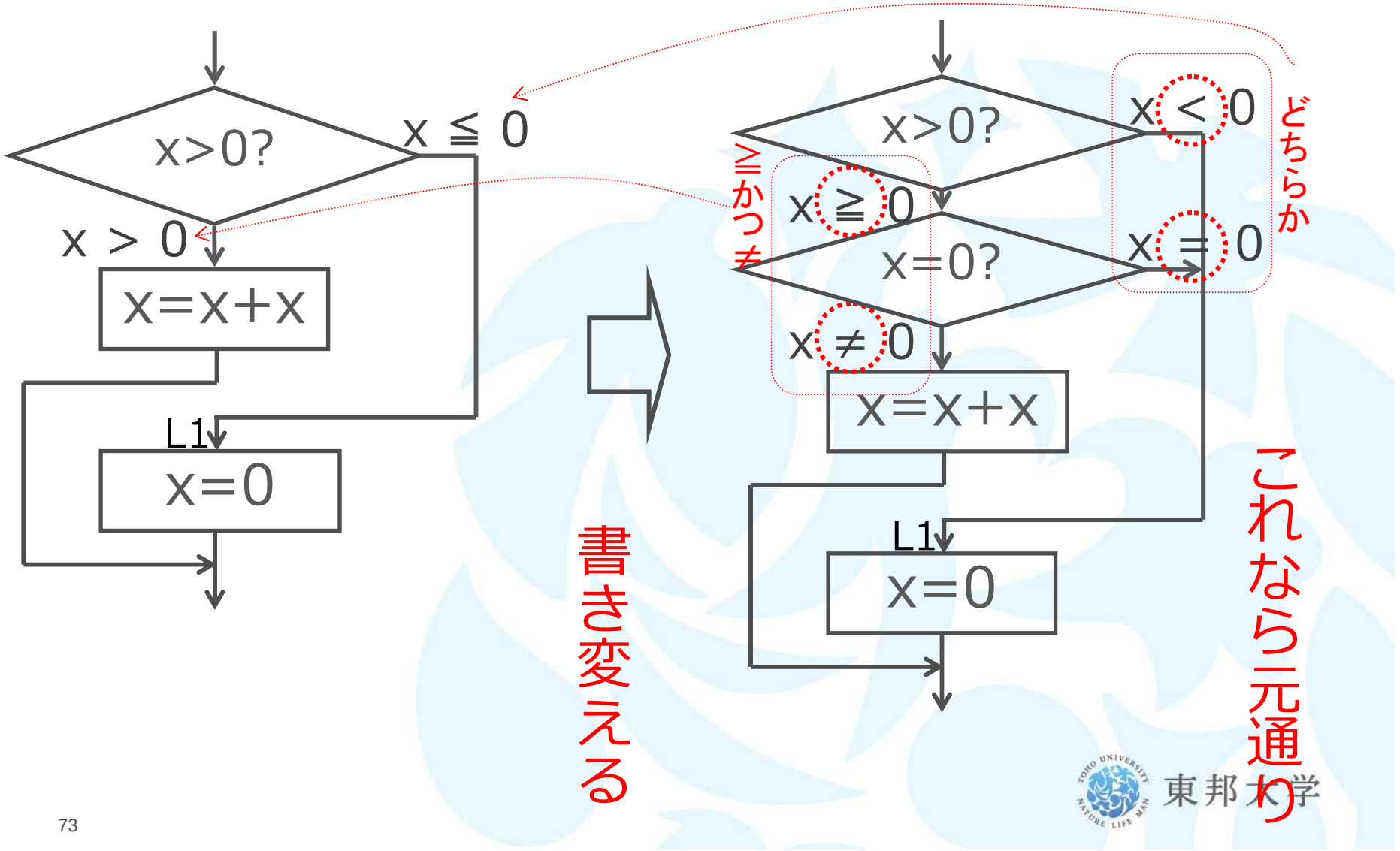
方法１）判定条件をひっくり返す



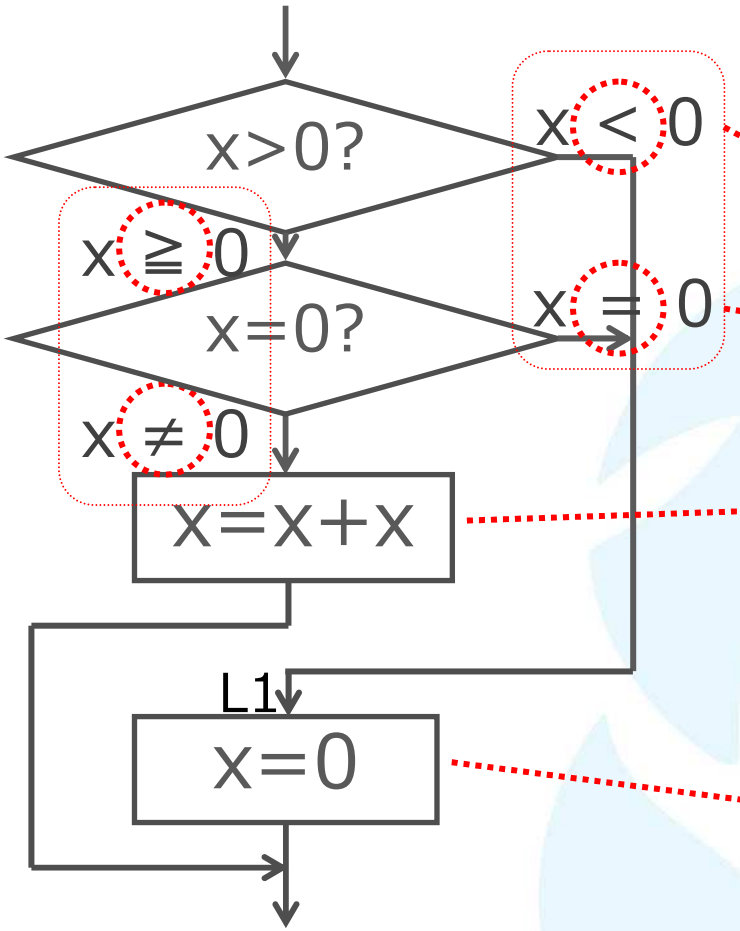

```
LD    GR3, ZERO
LD    GR4, X        逆にした
CPA   GR3, GR4    (0-x)
JPL   L1              正→L1

x = 0 の処理

JUMP L2

L1
x = x+x の処理

L2 次の命令
```


東邦大学

方法２）＜と＝の２回判定



方法２）＜と＝の２回判定




```
LD    GR3, ZERO
LD    GR4, X
CPA   GR4, GR3    (x-0)
JPL   L1          正→L1
JEZ   L1          0 →L1
```

x = 0 の処理

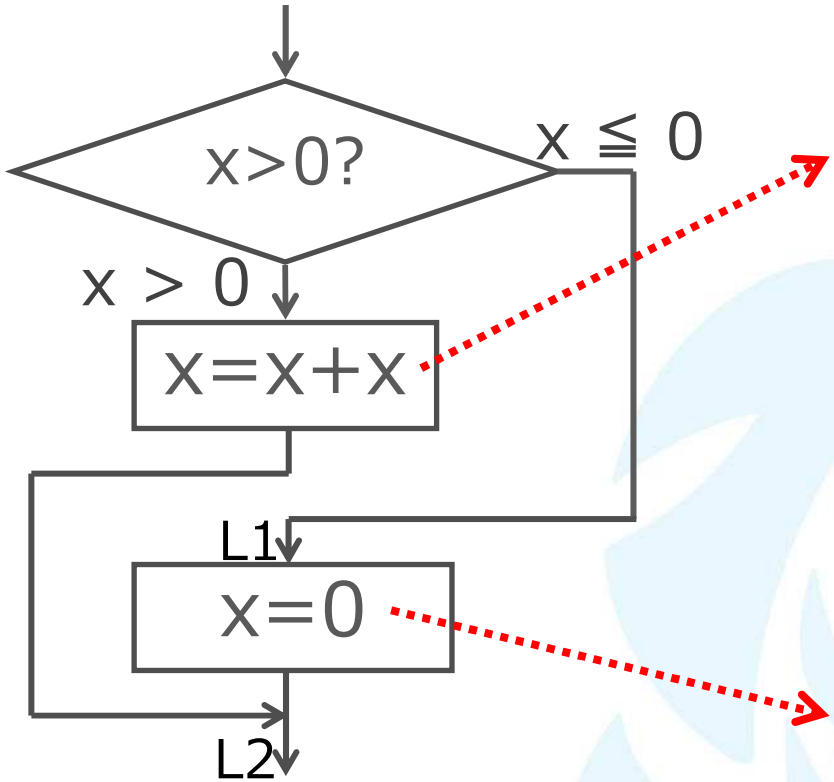
JUMP L2

L1

x = x+x の処理

L2 次の命令

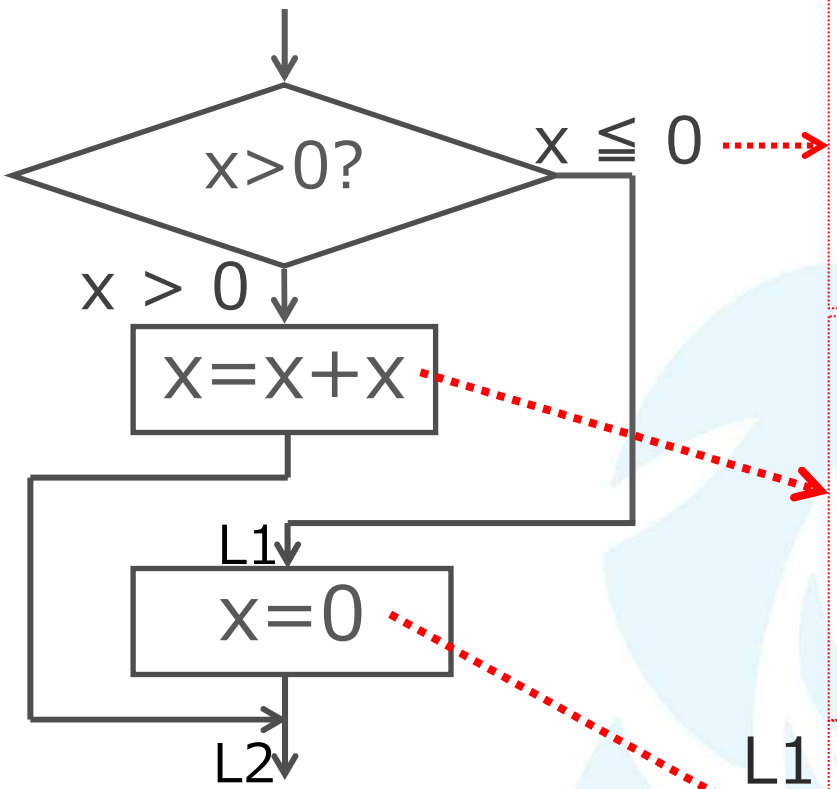
下の２つは既にできるはず




```
LD      GR3, X      GR3←X
ADDA    GR3, GR3
                    GR3←GR3+GR3
ST      GR3, X      X←GR3
```

```
LD      GR3, ZERO
                    GR3←ZERO
ST      GR3, X      X←GR3
```

全体を合わせると




```
LD    GR3, ZERO      ２番目の方法
LD    GR4, X
CPA   GR4, GR3    (x-0)
JPL   L1             正→L1
JEZ   L1             0 →L1

LD      GR3, X       GR3←X
ADDA    GR3, GR3
                  GR3←GR3+GR3
ST      GR3, X       X←GR3
JUMP    L2

L1  LD      GR3, ZERO
                     GR3←ZERO
    ST      GR3, X   X←GR3

L2  次の命令
```

課題で追加した話は

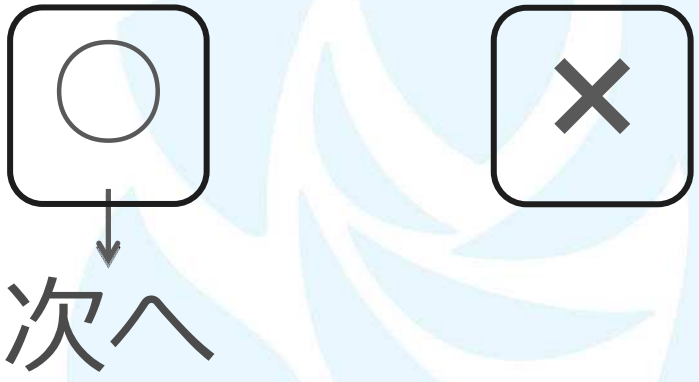
条件分岐での JPL や JMI の選び方は
　必ずしも一通りではない

条件を逆にし、飛び先も逆にすれば、
　同じような事ができる（=の時に注意)

２つ連続で JPL と JZE などを使っていい
　１つ目でジャンプしなかったときに
　　２つ目の条件分岐 Jxx が実行される
　　(注：CPAは１回だけでよい)

東邦大学

if 文（条件分岐）から機械命令へ変換の仕方がわかりましたか？



東邦大学