Structures de Contrôle

Traduction C/langage d'assemblage

Charles André – Université de Nice – Sophia Antipolis

Alternative

```
if (a <b) {
        ...
        <then part>
        ...
} else {
        ...
        <else part>
        ...
}
```

```
if:     cmp   a,b
        bgez else
        ...
        <then part>
        ...
        bt    endif
else:
        ...
        <else part>
        ...
endif:
```

Alternative

```
if (a <b) {
        ...
        <then part>
        ...
} else {
        ...
        <else part>
        ...
}
```

```
if:     load    a
        sub     b
        bgez    else
        ...
        <then part>
        ...
        bt      endif
else:
        ...
        <else part>
        ...
endif:
```

Si on n'a pas **cmp**

Condition complexe (and)

```
if (
   (a < b)
   &&
   (c >= d)
)
{
   …
   <code>
   …
}
```

```
if:    cmp  a,b
       bgez endif
       cmp  c,d
       bltz  endif
       …
       <code>
       …
endif:
```

Condition complexe (or)

```
if (
   (a < b)
   ||
   (c >= d)
)
{
   …
   <code>
   …
}
```

```
if:    cmp  a,b
       bltz  todo
       cmp  c,d
       bltz  endif
todo:
       …
       <code>
       …
endif:
```

Tant que

```
while
        ( a < b ) {
   …
   <code>
   …
}
```

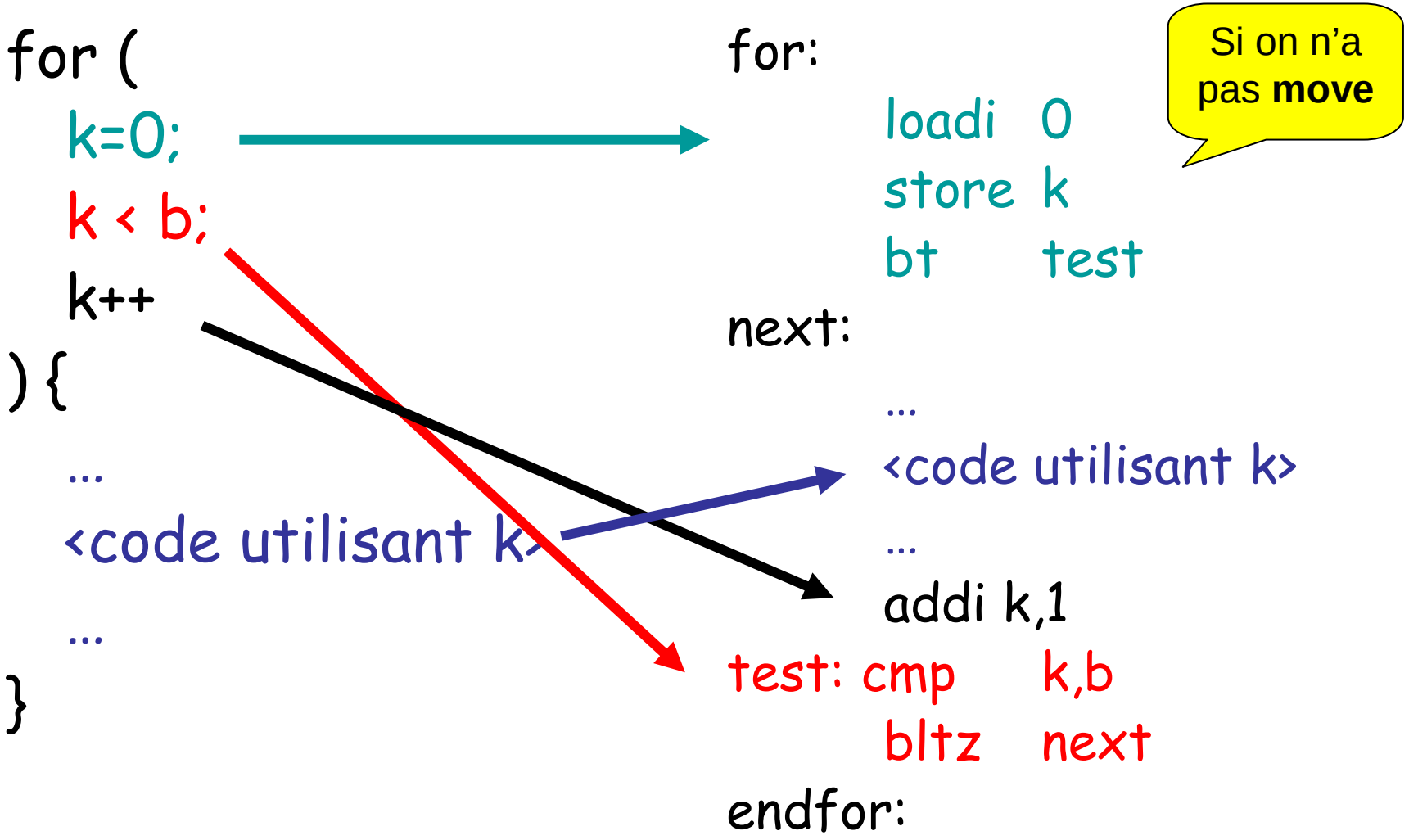
```
while:
        cmp   a,b
        bgez endwhile
        …
        <code>
        …
        bt    while
endwhile:
```

For

```
for (
   k=0;
   k < b;
   k++
) {
   …
   <code utilisant k>
   …
}
```

```
for:
        move k,0
        bt    test
next:
        …
        <code utilisant k>
        …
        addi k,1
test:   cmp  k,b
        bltz  next
endfor:
```

For


for (
k=0;
k < b;
k++
) {
...
<code utilisant k>
...
}
for:
loadi 0
store k
bt test
next:
...
<code utilisant k>
...
addi k,1
test: cmp k,b
bltz next
endfor:
Si on n'a pas **move**

For

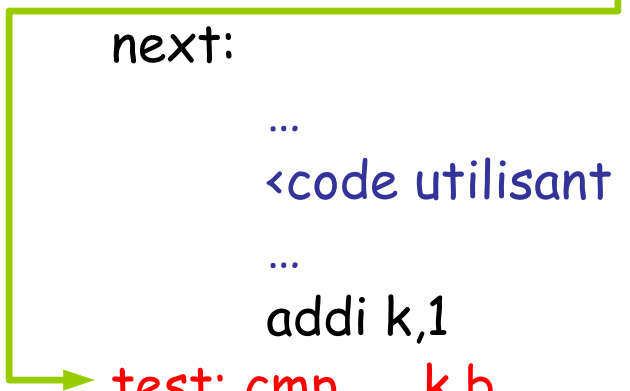
```
for (
   k=0;
   k < b;
   k++
) {
   …
   <code utilisant k>
   …
}
```

```
for:     loadi  0
         store  k
         bt     test
next:
         …
         <code utilisant k>
         …
         addi k,1
test:    cmp    k,b
         bltz   next
endfor:
```

For

```
for (
   k=0;
   k < b;
   k++
) {
   …
   <code utilisant k>
   …
}
```

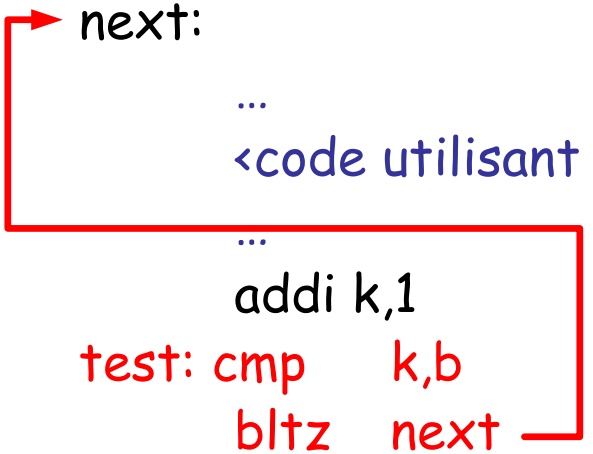
```
for:
        loadi  0
        store  k
        bt     test
next:
        …
        <code utilisant k>
        …
        addi k,1
test:   cmp    k,b
        bltz   next
endfor:
```



For

```
for (
   k=0;
   k < b;
   k++
) {
   …
   <code utilisant k>
   …
}
```

```
for:
        loadi   0
        store   k
        bt      test
next:
        …
        <code utilisant k>
        …
        addi k,1
test:   cmp     k,b
        bltz    next
endfor:
```



For

```
for (
   k=0;
   k < b;
   k++
) {
   …
   <code utilisant k>
   …
}
```

```
for:
        loadi   0
        store   k
        bt      test
next:
        …
        <code utilisant k>
        …
        addi k,1
test:   cmp     k,b
        bltz    next
endfor:
```

For

- Remarque:
  - Si k n'intervient pas explicitement dans la boucle
  - Alors, il vaut mieux faire un décompteur et détecter son passage au 0
  - Flag Z, instructions beqz, bnez

For

```
for (
    k = b;
    k > 0;
    k--
) {
    ...
    <code n'utilisant pas k>
    ...
}
```

```
for:
        move  k,b
        bt    test
next:
        ...
        <code n'utilisant pas k>
        ...
        load  k
        subi  1
        store k
test:   load  k
        bnez  next
endfor:
```

cas

```
switch (k) {
   case 1:     …
               …
               break;
   case 2:     …
               …
               break;
   …
   default: …
}
```

```
switch:
case1: cmp   k,1
             bnez  case2
             …
             …
             bt    endsw
case2:cmp    k,2
             bnez  case3
             …
             bt    endsw
default:     …
endsw:
```