C to assembly

Purpose

- To translate basic C statements into assembly instructions
- **These are just examples: different solutions exist**
- MIPS64 as reference.

If-then-else

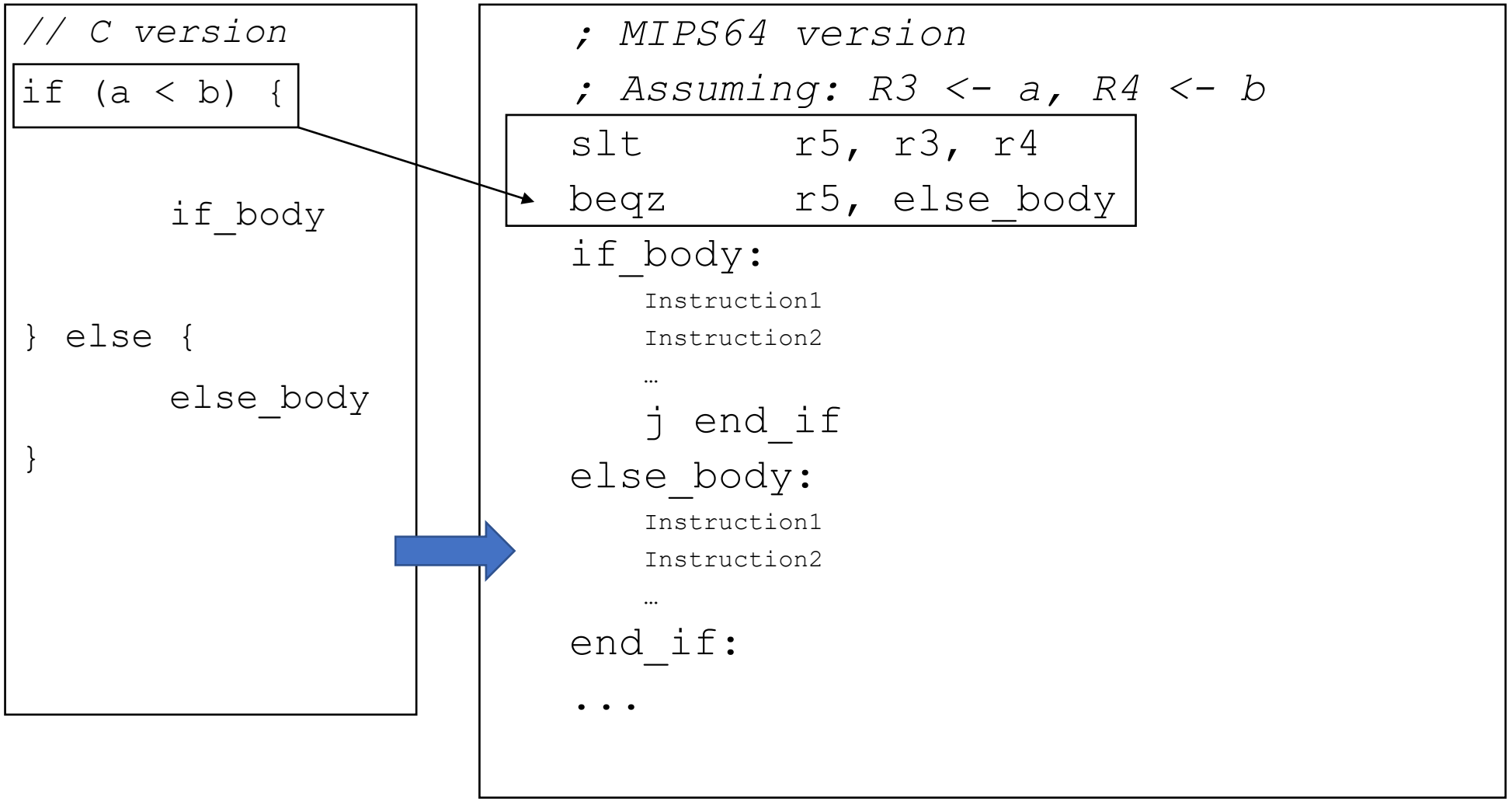
```
// C version
if (a < b) {

        if_body

} else {
        else_body
}
```

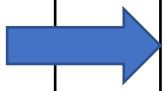
```
; MIPS64 version
; Assuming: R3 <- a, R4 <- b
slt       r5, r3, r4
beqz      r5, else_body
if_body:
    Instruction1
    Instruction2
    …
    j end_if
else_body:
    Instruction1
    Instruction2
    …
end_if:
...
```

If-then-else


// C version
if (a < b) {
if_body
} else {
else_body
}
; MIPS64 version
; Assuming: R3 <- a, R4 <- b
slt r5, r3, r4
beqz r5, else_body
if_body:
Instruction1
Instruction2
…
j end_if
else_body:
Instruction1
Instruction2
…
end_if:
...

For loop – Version 1

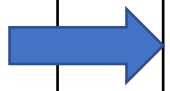
```
// C version
for (i=0; i<5;i++){
        for_body
}
```



```
; MIPS64 version
; Assuming: R2 <- i
init:
;i = 0
daddi r2, r0,0
for_body:
    ; i < 5
    slti r8, r2, 5
    beq r8, r0, end_loop
    Instruction1
    Instruction2
    …
    ;i++
    daddi r2,r2, 1
    j for_body
end_loop:
```

For loop – Version 2

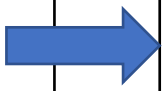
```
// C version
for (i=0; i<5;i++){
        for_body
}
```



```
; MIPS64 version
; Assuming: R2 <- i
init: daddi r2, r0,5
for_body:
    Instruction1
    Instruction2
    …
    daddi r2, r2, -1
    bnez r2, for_body
```

While loop

```
// C version
i = 0;
do {
        while_body
        i++;
} while(i!=10);
```



```
; MIPS64 version
; Assuming: R2 <- i
init:
daddi r2, r0,0
daddi r8, r0, 10
while_body:
    Instruction1
    Instruction2
    …
    daddi r2,r2, 1
    bne r2,r8, while_body
end_loop:
```