

# Compiler Construction

~ Complex Translations ~

# Translation to HIR

How to translate complex expressions:

- array creation
- record creation
- string comparison
- while loops
- for loops

# Array Creation

An array is composed of:

- A type
- A dimension
- An initial value

How to perform a malloc?

# Array Creation: solution

Introduce a Runtime!

Offer a function that will provide anything required for allocating an array: malloc, memcpy, etc.

⇒ Same solution for string comparison, and record creation

# While Loops

How to translate:

```
while condition  
do body
```

Possible translation:

# While Loops

How to translate:

```
while condition  
do body
```

Possible translation:

```
test:  
    if not (condition)  
        goto done  
    body  
    goto test  
done:
```

# For-loops

How to translate:

```
for i := min to max  
do body
```

Erroneous translation:

# For-loops

How to translate:

```
for i := min to max  
do body
```

Erroneous translation:

```
let i := min  
    limit := max  
in  
    while i <= limit  
    do  
        (body; ++i)  
end
```



# For-loops

```
let i := min
    limit := max
in
  test:
    if not (i <= limit)
      goto done
    (body; ++i)
    goto test
  done:
end
```

# For-loops

```
let i := min
    limit := max
in
  test:
    if not (i <= limit)
      goto done
    (body; ++i)
    goto test
  done:
end
```

What if  $i$  is unsigned, i.e. with arithmetic performed modulo  $2^n$

# For-loops

Correct translation:

```
let i := min
    limit := max
in
    if (i > limit)
        goto end
loop:
    body
    if (i >= limit)
        goto end
    ++i
    goto loop
end:
```

# Additional Features

- Bounds checking
- Nil checking
- ...

# Summary

