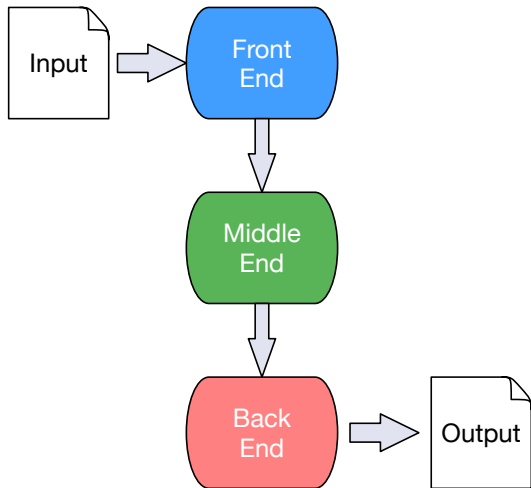


Compiler Construction

⤿ Middle End ⤿

General layout – Compiler structure



Goal of middle end

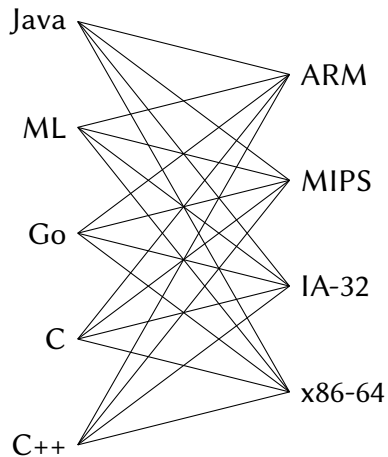
- Generic optimisations
- Prepare and simplify backend !
 - ▶ No nested sequences in ASM
 - ▶ No expressions in ASM
 - ▶ No two-way conditional jumps in ASM
 - ▶ ASM has calling convention

Goal of middle end

- Generic optimisations
- Prepare and simplify backend !
 - ▶ No nested sequences in ASM
 - ▶ No expressions in ASM
 - ▶ No two-way conditional jumps in ASM
 - ▶ ASM has calling convention

Do we require an intermediate representation?

Retargetable Compilers (1/2)



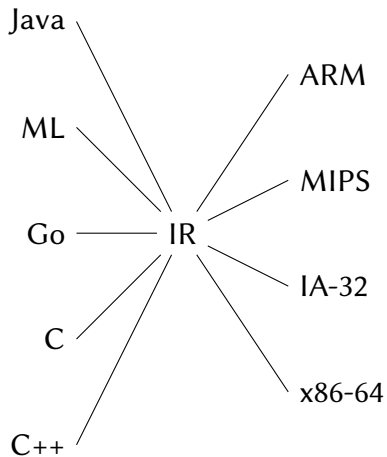
Goal of middle end

An intermediate representation is not required

BUT

.. preferable for optimisations and retargetable compilers

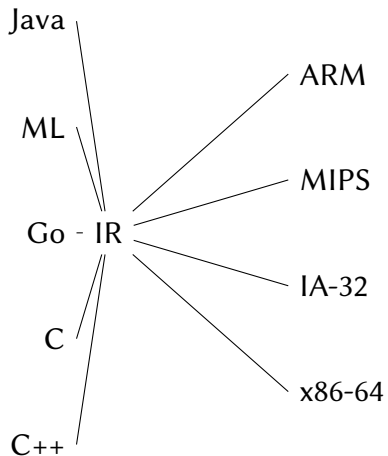
Retargetable Compilers (2/2)



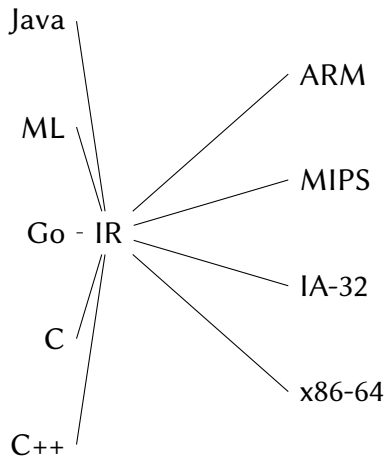
What abstraction?

What is the best abstraction level for the intermediate representation?

Strategy 1: close to front end

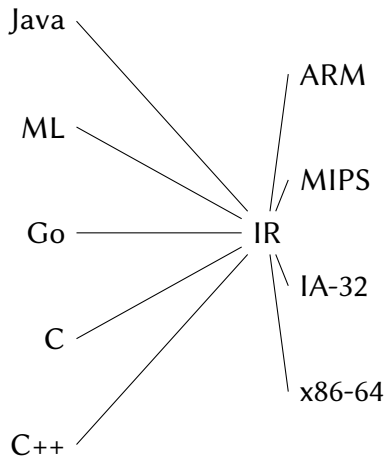


Strategy 1: close to front end

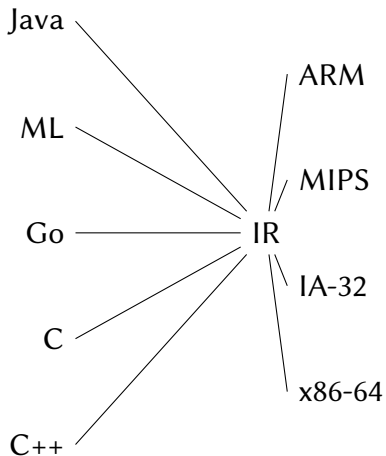


- Easy translation from source language
- Very hard translation from IR to ASM

Strategy 2: close to back end

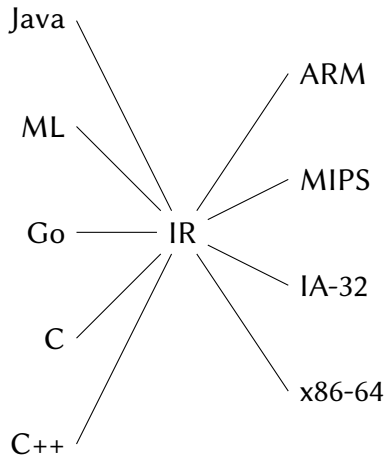


Strategy 2: close to back end

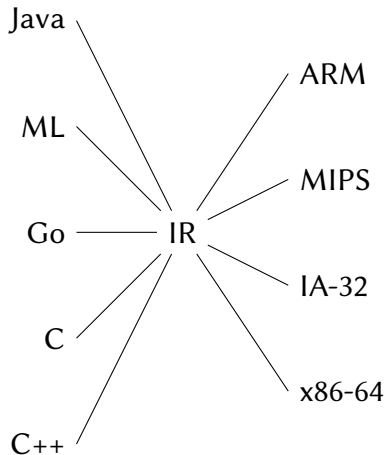


- Very hard translation from source language
- Easy translation from IR to ASM

Strategy 3: barycenter of sources and targets

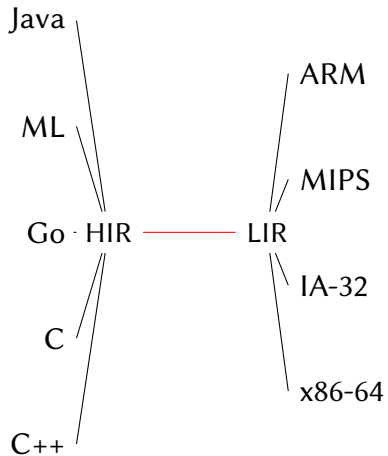


Strategy 3: barycenter of sources and targets

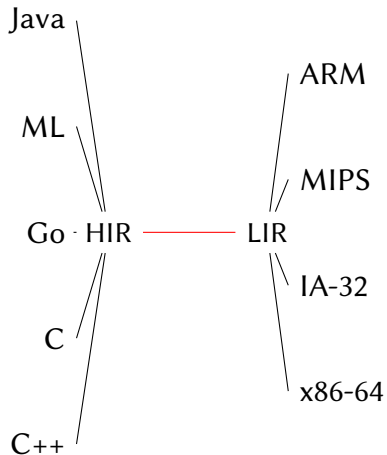


- Hard to find the barycenter of languages
- Not so easy translation from source to IR and from IR to ASM

Strategy 4: Two IRs

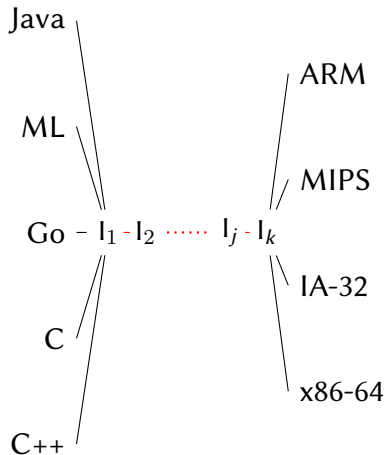


Strategy 4: Two IRs

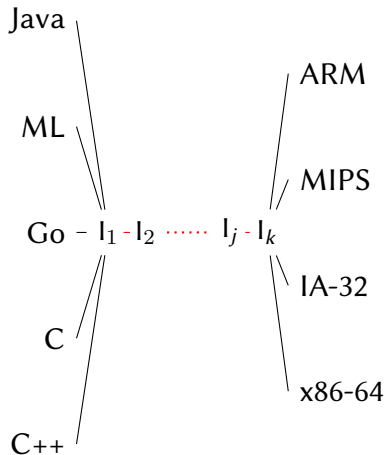


- Easy translation from source to HIR and from LIR to ASM
- Hard translation from HIR to LIR

Strategy 5: Many IRs



Strategy 5: Many IRs



- Easy translation from source to I_1 and from I_k to ASM
- Hard to define multiple language

Solution

$l_1, l_2, \dots, l_j, l_k$ are the same language!

l_j is sugar over l_k

...

l_2 is sugar over l_3

l_1 is sugar over l_2

Solution

$l_1, l_2, \dots, l_j, l_k$ are the same language!

l_j is sugar over l_k

...

l_2 is sugar over l_3

l_1 is sugar over l_2

⇒ Translation from HIR to LIR will only
a **form** of unsugarring

Summary

