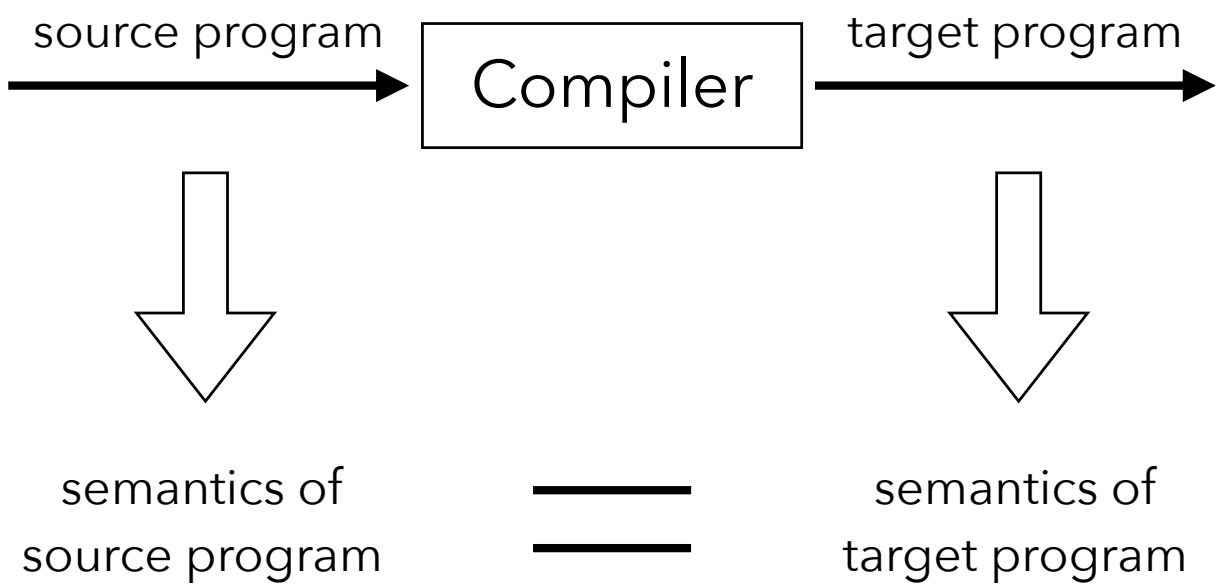
COSE312: Compilers

Lecture 18 — Course Review

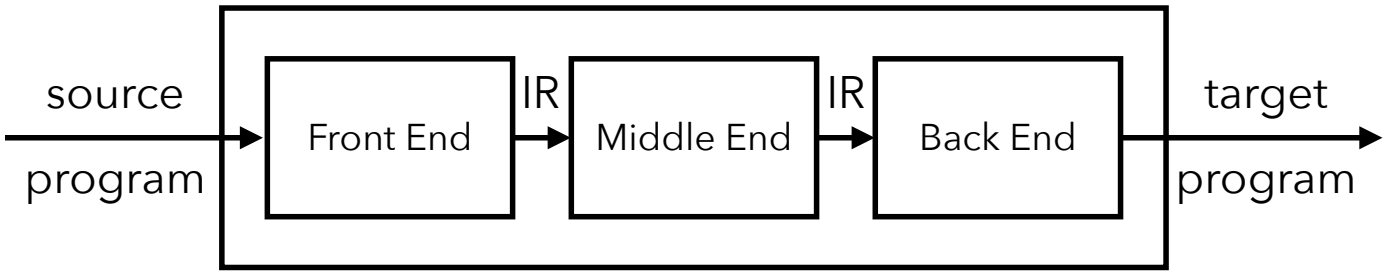
Hakjoo Oh
2026 Spring

Compilers

Software systems that translate a program written in one language ("source language") into a program written in another language ("target language").

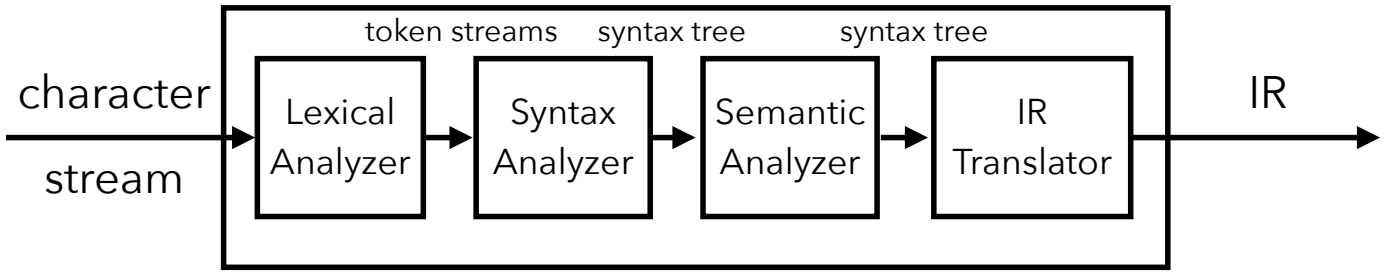


Structure of Modern Compilers



- The front-end understands the source program and translates it to an intermediate representation (IR).
- The middle-end takes a program in IR and optimizes it in terms of efficiency, energy consumption, and so on.
- The back-end transforms the IR program into machine-code.

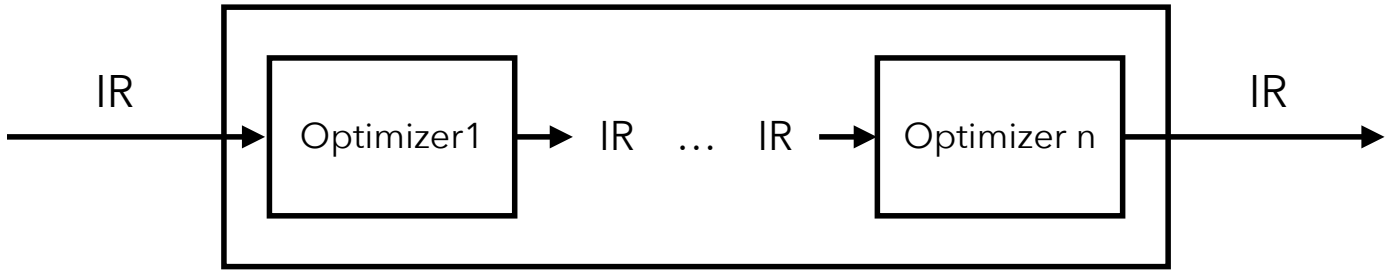
Front End



- The lexical analyzer transforms the character stream into a stream of tokens.
- The syntax analyzer transforms the stream of tokens into a syntax tree.
- The semantic analyzer checks if the program is semantically well-formed.
- The IR translator translates the syntax tree into IR.

Middle End

Transform IR to have better performance:



ex)

```
t1 = 10
t2 = rate * t1
t3 = init + t2
pos = t3
```

original IR

```
t1 = 10
t2 = rate * 10
t3 = init + t2
pos = t3
```

```
t2 = rate * 10
t3 = init + t2
pos = t3
```

```
t2 = rate * 10
pos = init + t2
```

final IR

Back End

Generate the target machine code:

ex) from the IR

```
t2 = rate * 10
pos = init + t2
```

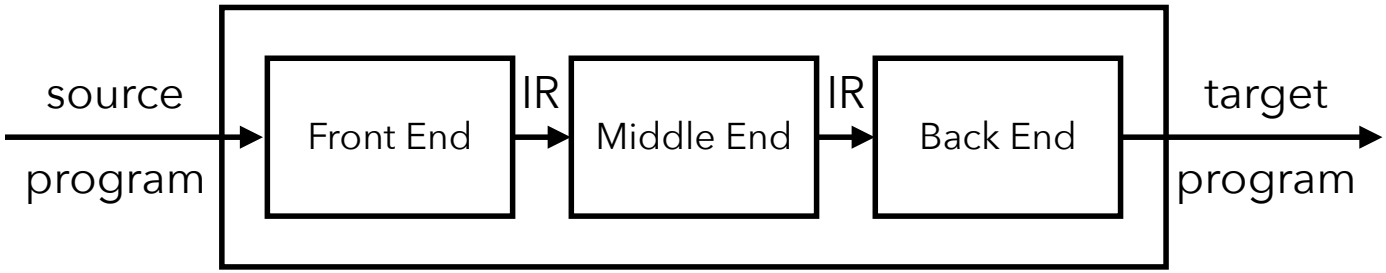
generate the machine code

```
LOAD  R2, rate
MUL   R2, R2, #10
LOAD  R1, init
ADD   R1, R1, R2
STORE pos, R1
```

- A key component of compiler back-end is register allocation.
- The remaining translation from IR to machine code is not difficult.

Summary

A modern compiler consists of three phases:



- Front end understands the syntax and semantics of source program.
- Middle end improves the efficiency of the program.
- Back end generates the target program.

한학기 수고 많았습니다!