

# CS251: Introduction to Language Processing

## Machine Independent Optimizations

**Vishwesh Jatala**

Assistant Professor

Department of CSE

Indian Institute of Technology Bhilai

[vishwesh@iitbhilai.ac.in](mailto:vishwesh@iitbhilai.ac.in)

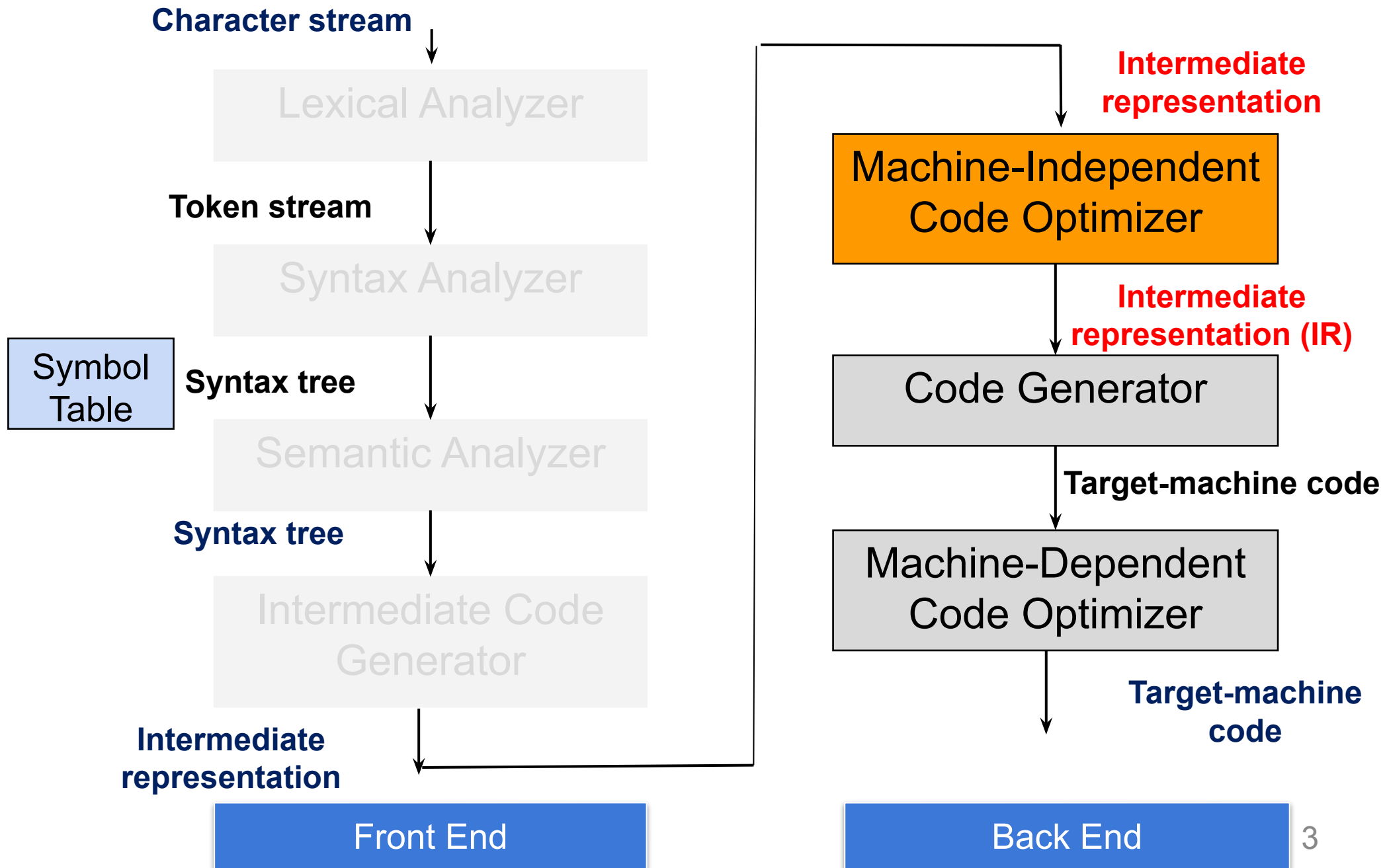


2023-24 M

# Acknowledgement

- References for today's slides
  - *Stanford University*  
*<https://web.stanford.edu/class/archive/cs/cs143/cs143.1128/>*
  - *Prof. Y. N Srikant, IISc Bangalore*  
*<https://iith.ac.in/~ramakrishna/Compilers-Aug14/slides/>*
  - *<http://sei.pku.edu.cn/~yaoguo/ACT11/slides/lect2-opt.ppt>*
  - *Course textbook*

# Compiler Design



# Optimizations

- Global common subexpression elimination
- Copy propagation
- Constant propagation and constant folding
- **Loop invariant code motion**
- Induction variable elimination and strength reduction

# Loop Invariant Code Motion Example

```
t1 = 202
i = 1
L1: t2 = i > 100
    if t2 goto L2
    t1 = t1 - 2
    t3 = addr(a)
    t4 = t3 - 4
    t5 = 4 * i
    t6 = t4 + t5
    *t6 = t1
    i = i + 1
    goto L1
L2:
```

Before LIV  
code motion

```
t1 = 202
i = 1
    t3 = addr(a)
    t4 = t3 - 4
L1: t2 = i > 100
    if t2 goto L2
    t1 = t1 - 2
    t5 = 4 * i
    t6 = t4 + t5
    *t6 = t1
    i = i + 1
    goto L1
L2:
```

After LIV  
code motion

# Optimizations

- Global common subexpression elimination
- Copy propagation
- Constant propagation and constant folding
- Loop invariant code motion
- Induction variable elimination and strength reduction

# Strength Reduction

```
t1 = 202
i = 1
t3 = addr(a)
t4 = t3 - 4
L1: t2 = i > 100
    if t2 goto L2
    t1 = t1 - 2
    t5 = 4 * i
    t6 = t4 + t5
    *t6 = t1
    i = i + 1
    goto L1
L2:
```

Before strength  
reduction for t5

```
t1 = 202
i = 1
t3 = addr(a)
t4 = t3 - 4
t7 = 4
L1: t2 = i > 100
    if t2 goto L2
    t1 = t1 - 2
    t6 = t4 + t7
    *t6 = t1
    i = i + 1
    t7 = t7 + 4
    goto L1
L2:
```

After strength reduction  
for t5 and copy propagation

# Induction Variable Elimination

```
t1 = 202
i = 1
t3 = addr(a)
t4 = t3 - 4
t7 = 4
L1: t2 = i > 100
    if t2 goto L2
    t1 = t1 - 2
    t6 = t4 + t7
    *t6 = t1
    i = i + 1
    t7 = t7 + 4
    goto L1
L2:
```

Before induction variable  
elimination (i)

```
t1 = 202
t3 = addr(a)
t4 = t3 - 4
t7 = 4
L1: t2 = t7 > 400
    if t2 goto L2
    t1 = t1 - 2
    t6 = t4 + t7
    *t6 = t1
    t7 = t7 + 4
    goto L1
L2:
```

After eliminating i and  
replacing it with t7



# Summary

- Machine Independent Optimizations
  - Improve the quality of code: performance, memory, and energy efficiency
  - Still hot area of research
- Formalize:
  - Basic blocks
  - Control flow graph
- Optimizations:
  - Examples