CS6843: Program Analysis

Instructor: Rupesh Nasre (rupesh@cse)
TAs: Bikash, Meghana, Seshagiri Rao

Web: ~rupesh/teaching/pa/jan19/

Jan 2019

What is Program Analysis?

For an end-goal, identify “interesting aspects” of a program's representation.

What is Program Analysis?

For an end-goal,

identify "interesting aspects"

of a program's representation.

Checking security

Array index range

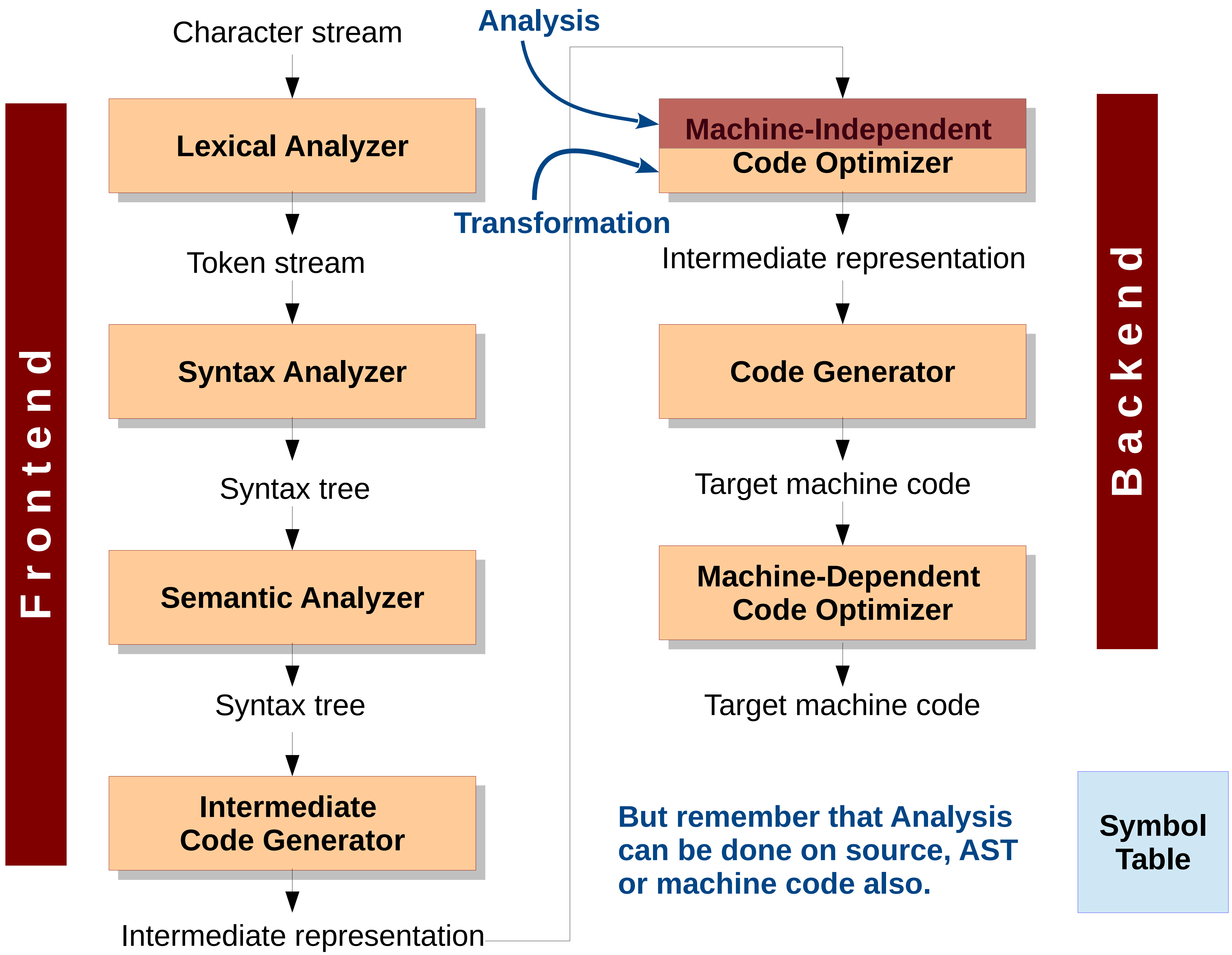
Source, AST, binary,
executed instruction

Classwork: Write down two types of information you can extract from programs.

Examples

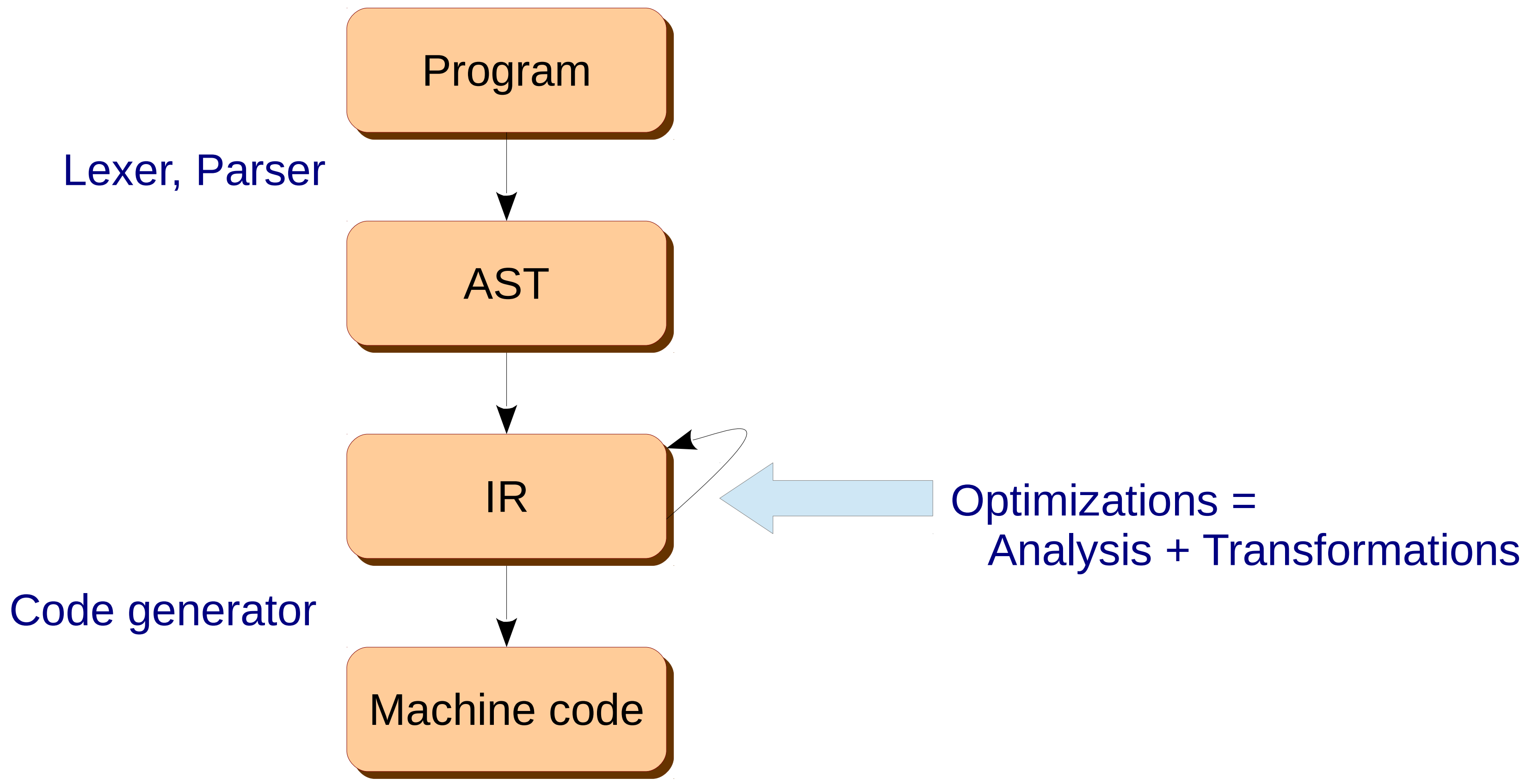
End goal	Interesting aspect
Dead code elimination	Reachability
Constant propagation	use-def
Security	Array index range, dangling pointers
Parallelization	Data dependence, SIMD opportunities
Debugging	Slice
Cache performance	Memory access pattern
Memory reduction	Live ranges
...	...

Program Analysis is often a pre-cursor to Optimization.



But remember that Analysis can be done on source, AST or machine code also.

Compiler Organization



Learning Outcomes

- To apply data-flow analysis and its variants on input programs and collect relevant information
  - reaching definitions, points-to information, etc.
- To design and implement analyses for new problems

Example Three

```
void main() {
  int a, b, c, d, *p;

  p = &a;
  c = a + b;
  d = *p + b;
}
```

Can this computation be avoided?
(*common subexpression elimination*)

This requires a program analysis called *pointer analysis*.

```
void main() {
  int a, b, c, d, *p;

  p = &a;
  int t = a + b;
  c = t;
  d = t;
}
```

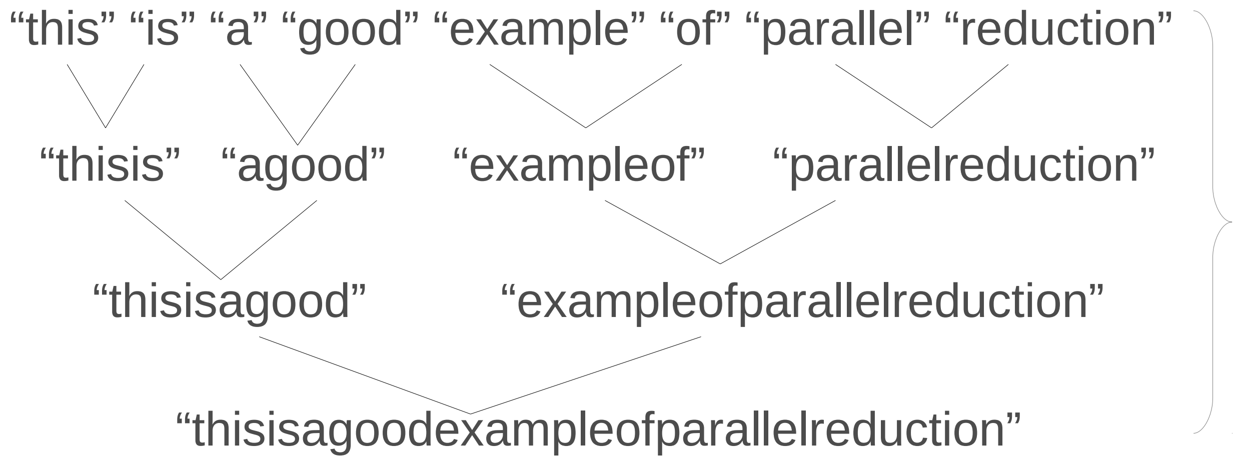
This requires another analysis called *type analysis*.

Example Two

```
*sresult = 0;

for (ii = 0; ii < nn; ++ii) {
  strcat(sresult, str[ii]);
}
```

Can you parallelize this code?



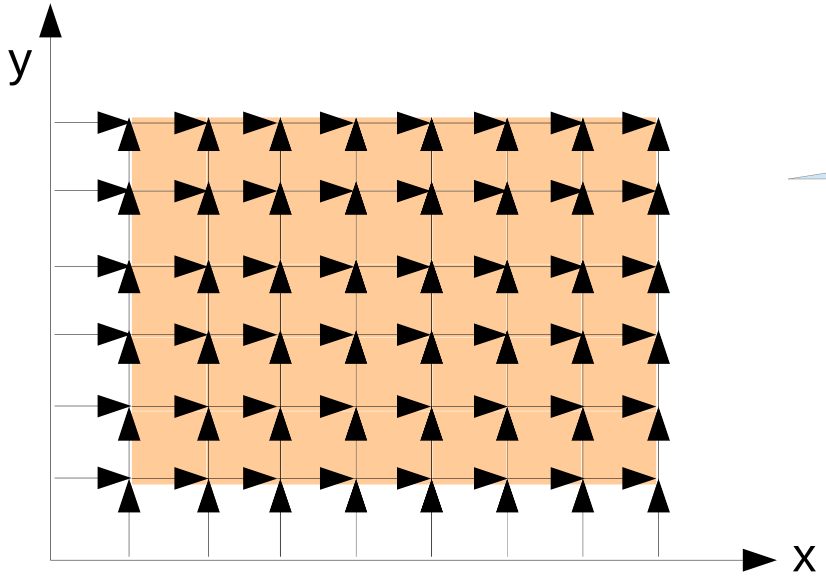
“this” “is” “a” “good” “example” “of” “parallel” “reduction”
“thisis” “agood” “exampleof” “parallelreduction”
“thisisagood” “exampleofparallelreduction”
“thisisagoodexampleofparallelreduction”

Requires *semantic analysis* to figure out that *strcat* performs an **associative** operation.

Example One

```
for (x = 1; x < M; ++x)
  for (y = 1; y < N; ++y)
    a[x, y] = a[x – 1, y] + a[x, y – 1];
```

Can you parallelize iterations?



Requires
loop dependence analysis

In This Course

7. Dynamic Analysis (DYN)
6. Shape Analysis (SHA)
5. Program Slicing (SLI)
4. Parallelization (PAR)
3. Security Analysis (SEC)
2. Pointer Analysis (PTR)
1. Data Flow Analysis (DFA)

Support material

- These slides
- Scribes
- Online tools

Logistics

- Moodle for submissions, announcements, discussions.
  - Your responsibility to subscribe to it.
- Evaluation:
  - assignments (50%)
  - midsem (25%)
  - endsem (25%)
- G slot (Mon 12, Wed 16:50, Thu 10, Fri 9).
- Room CS 24.

Assignments

- Four programming assignments (50%).
  - 5 + 10 + 15 + 20
- Assignments would be in LLVM.
- You should work individually.
- A1 is due January 22.
- You have this week to suggest me any date changes for A2, A3, A4.

Course Schedule

Month	Lectures	Evaluations
JAN	DFA	A1
FEB	PTA, PAR	A2
MAR	SEC, DYN	A3, MIDSEM
APR	SHA, SLI	A4
MAY		ENDSEM

MidSem and EndSem will have
mutually exclusive topics.

CS6843: Program Analysis

Instructor: Rupesh Nasre (rupesh@cse)
TAs: Bikash, Meghana, Seshagiri Rao

Web: ~rupesh/teaching/pa/jan19/

Jan 2019