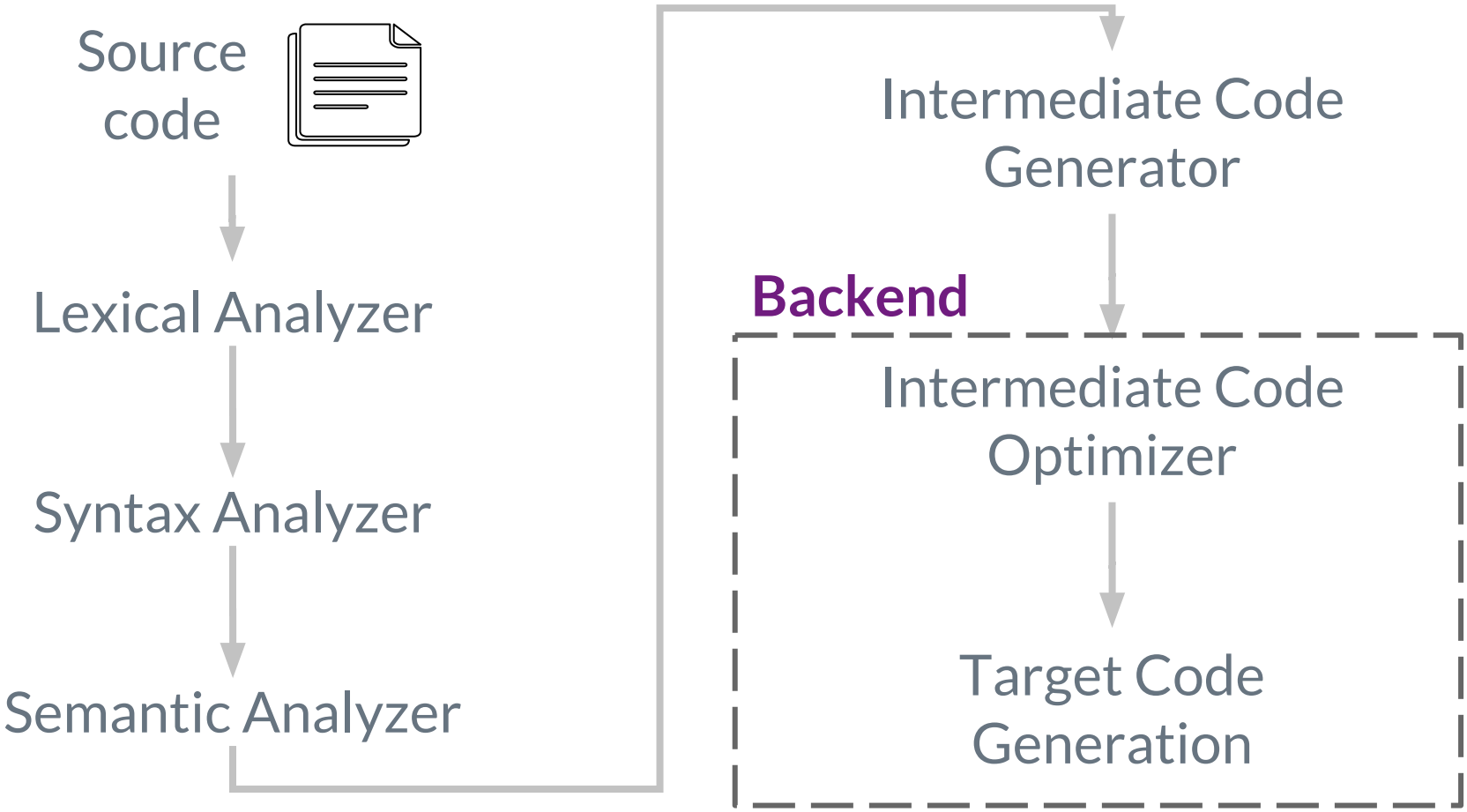
Compilers
Lab VI

Mo.Agamia

Plan

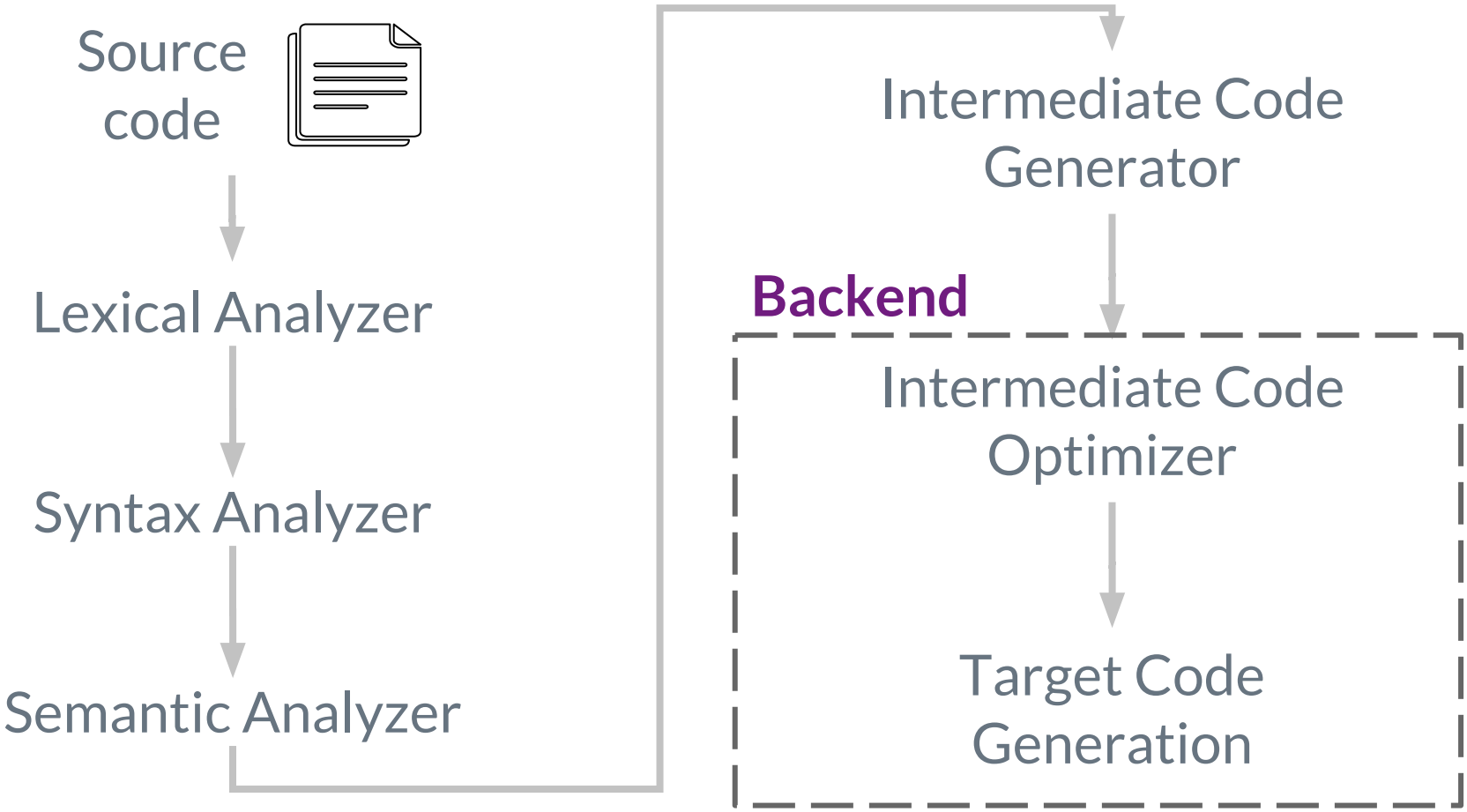
- Overview
- Parsers overview
- First & Follow
- LL(1) table

Compiler phases



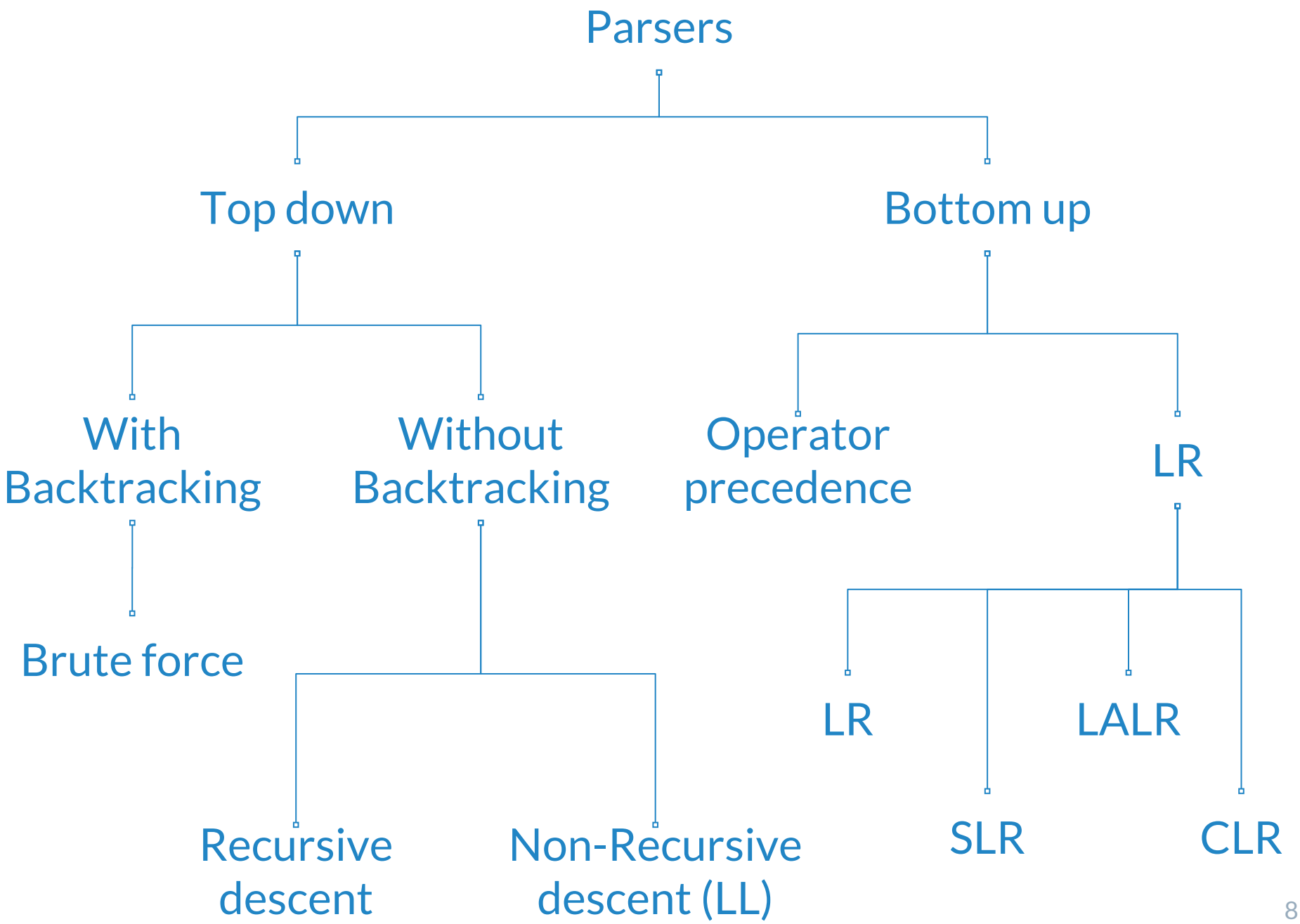
1.
Overview

Compiler phases



2.
Parsers

Parsing is the process of analysing a string of symbols, conforming to the rules of a formal grammar.


Parsers
Top down
Bottom up
With Backtracking
Without Backtracking
Operator precedence
LR
Brute force
LR
LALR
Recursive descent
Non-Recursive descent (LL)
SLR
CLR

LL parser is a top-down parser for a subset of context-free languages.

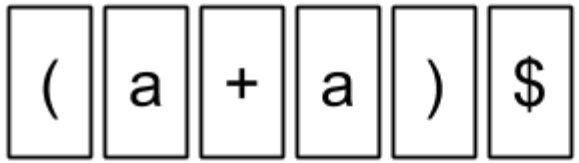
It parses the input from left to right using left-most derivation & k represents the number of look aheads

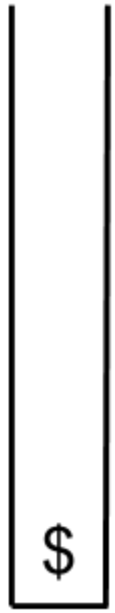
LL(1)

Example :

1. $S \rightarrow F$
2. $S \rightarrow (S + F)$
3. $F \rightarrow a$

	(	)	a	+	$
S	2	-	1	-	-
F	-	-	3	-	-





N.B.

- If a grammar is not left factored, then it can not be LL(1)

  Eg - S -> aS | a ---- both productions go in a

- If a grammar is left recursive, it can not be LL(1)

  Eg - S -> Sa | b

  S -> Sa goes to FIRST(S) = b

  S -> b goes to b, thus b has 2 entries hence not LL(1)

- If a grammar is ambiguous then it can not be LL(1)
- Every regular grammar need not be LL(1) because

  regular grammar may contain left factoring, left recursion or ambiguity.

3.
First & Follow

First & follow are needed by the parser, so that it can properly apply the needed rule at the correct position.

4.
LL(1) table

LL(1)

Example :

1. S → (1) if expr then S else S
   | (2) while E do S
   | (3) begin T end
2. T → (4) S ; T | (5) ε
3. E → (6) id

	First	Follow
S		
T		
E		

	if	then	else	while	do	begin	end	id	;	$
S										
T										
E										

LL(1)

Example :

1. S → (1) if expr then S else S
   | (2) while E do S
   | (3) begin T end
2. T → (4) S ; T | (5) ε
3. E → (6) id

	First	Follow
S	If, while, begin	$, else, ;
T	If, while, begin, ε	end
E	id	do

	if	then	else	while	do	begin	end	id	;	$
S										
T										
E										

LL(1)

Example :

1. S → (1) if expr then S else S
   | (2) while E do S
   | (3) begin T end
2. T → (4) S ; T | (5) ε
3. E → (6) id

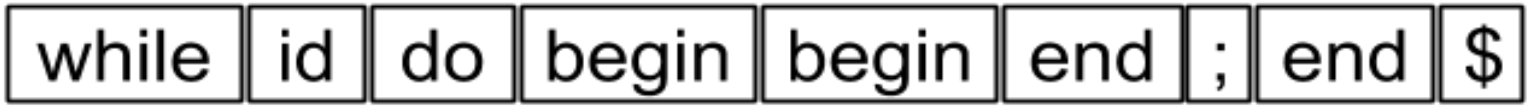
	First	Follow
S	If, while, begin	$, else, ;
T	If, while, begin, ε	end
E	id	do

	if	then	else	while	do	begin	end	id	;	$
S	1			2		3				
T	4			4		4	5			
E								6		

Example :

1. S → (1) if expr then S else S
   | (2) while E do S
   | (3) begin T end
2. T → (4) S ; T | (5) ε
3. E → (6) id

	if	then	else	while	do	begin	end	id	;	$
S	1			2		3				
T	4			4		4	5			
E								6		



Thanks!

Any questions?

You can find me at:

@piazza

mohammed.agamia@guc.edu.eg

References

https://courses.cs.washington.edu/courses/cse401/04sp/slides/03b-LL1-example.pdf
https://www.geeksforgeeks.org/parsing-set-1-introduction-ambiguity-and-parsers/