COMPUTER SCIENCE

Combine CS
The Extra Step

COMPLETE COMPILER DESIGN IN 1 VIDEO

LIVE

21ST MAY, 2022

By, Rashmi Ma'am

Join telegram For FREE Test Practice: https://t.me/RashmiCCS

Combinecs

Combinecs

"ALL PROGRESS TAKES PLACE OUTSIDE OF YOUR COMFORT ZONE."

Michael John Bobak

+91-7666980624
UGC NET/SET COMPUTER SCIENCE
| CS/IT SUBJECTS
CAREER GUIDANCE | SKILL DEVELOPMENT

UGCNET 2022 Crash Course Launched....

/COMBINECS.COM

Enroll for Free Class | Visit combinecs.com for all new courses | Contact Us +91-7666980624

What is a compiler?

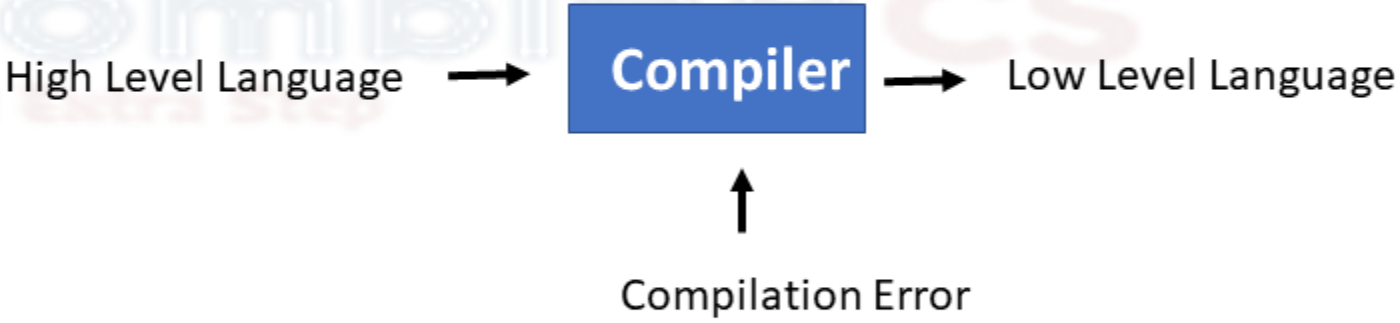
A compiler is a program that reads a program written in one language and translates it into an equivalent program in another language.

A compiler also reports errors present in the source program as a part of its translation process.

Types of the compiler.

1.Single-pass compiler

2.Multi-pass compiler

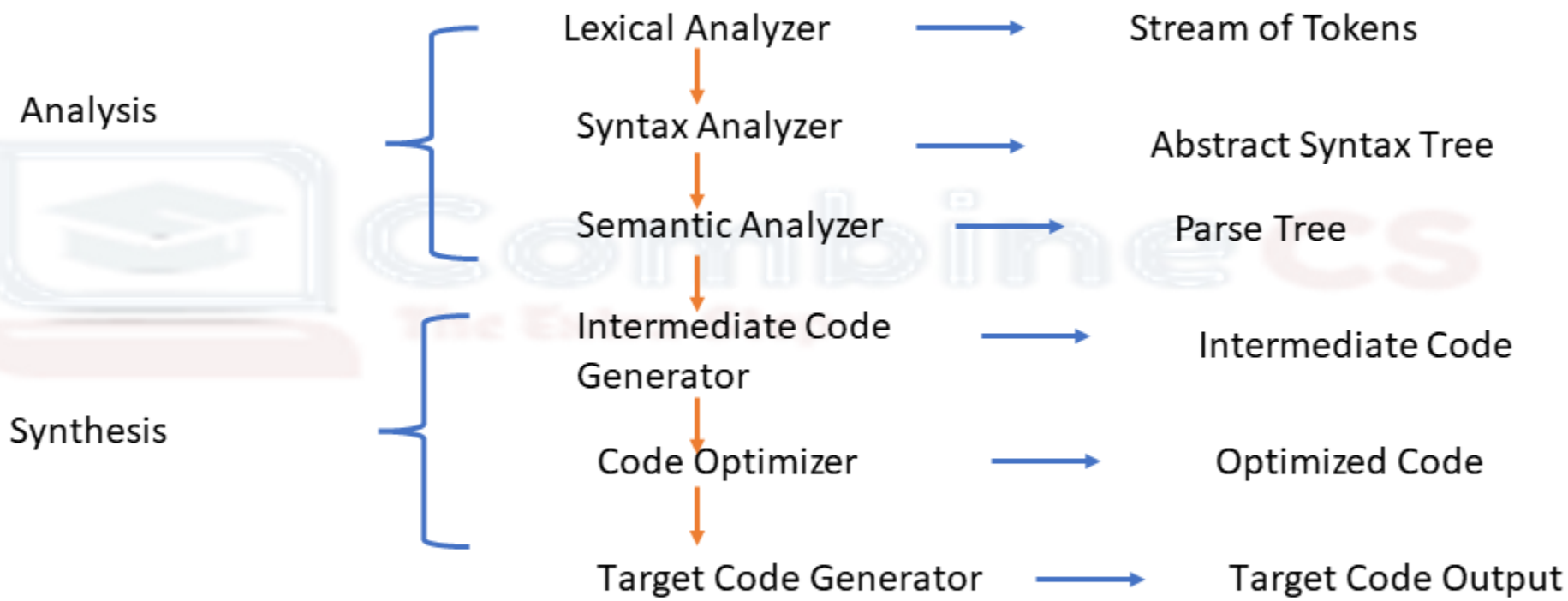


Difference between single-pass and multi-pass compiler.

	Single-pass compiler	Multi-pass compiler
Speed	fast	slow
Time	less	more
portability	No	YES
Efficient	More	Less
Memory	More	Less

Join telegram For FREE Test Practice: https://t.me/RashmiCCS

Phases of Compiler Design



Enroll for Free Class | Visit combinecs.com for all new courses | Contact Us +91-7666980624

Phases of a Compiler

Lexical Analysis: It reads the input program character by character and produces a string of tokens.
It removes comments, white spaces, blanks, newline characters, tab.

Parser/Syntax Analysis: It takes a string of token as an input and generate a parse tree and verifies it that string of token is a valid sequence.

Semantic Analysis: It verifies the parse tree, whether it's meaning full or not. If not then generate a new parse tree.

Intermediate code generator: it generates intermediate code, which can be readily executed by a machine.

Code optimizer: It optimized the code so that it consumes fewer resources and increases the speed.

Target code generator: It generates a code that the machine can understand. Output is dependent on the type of assembler.

Parts of the compilation process

Analysis: It takes the input source program and creates an intermediate representation of the source program.

Synthesis: It takes the intermediate representation as input and creates the desired target program.

Run-Time Error : A run-time error is an error which takes place during the execution of a program, and usually happens because of adverse system parameters or invalid input data.

Compile Time Error handling in a compiler.

1.lexical: includes misspellings of identifiers, keywords, and operators.
2.syntactical: a missing semicolon or unbalanced parenthesis.
3.semantical: incompatible value assignment or type mismatches between operator and operand.
4.logical: code not reachable or infinite loop.

What are patterns, tokens, and lexemes?

A **token** is a sequence of characters like identifiers, keywords, operators, etc.
A **pattern** describes a rule that must be matched by the sequence of characters to form a token.
lexeme is a sequence of characters in the source program that is matched by the pattern for a token.

What is a symbol table?

The symbol table is a data structure that stores various identifiers and their attributes.

Join telegram For FREE Test Practice: https://t.me/RashmiCCS

CombineCS

UGC-NET CS 2017 Nov - II

	List - I		List - II
(a)	A part of a compiler that is responsible for recognizing syntax.	(i)	Optimizer
(b)	A part of a compiler that takes as input a stream of characters and produces as output a stream of words along with their associated syntactic categories.	(ii)	Semantic Analysis
(c)	A part of a compiler that understand the meanings of variable names and other symbols and checks that they are used in ways consistent with their definitions.	(iii)	Parser
(d)	An IR-to-IR transformer that tries to improve the IR program in some way (Intermediate Representation).	(iv)	Scanner

Code :

	(a)	(b)	(c)	(d)
(1)	(iii)	(iv)	(ii)	(i)
(2)	(iv)	(iii)	(ii)	(i)
(3)	(ii)	(iv)	(i)	(iii)

Enroll for Free Class | Visit combinecs.com for all new courses | Contact Us +91-7666980624

Join telegram For FREE Test Practice: https://t.me/RashmiCCS

CombineCS

UGC-NET CS 2017 Nov - II

	List - I		List - II
(a)	A part of a compiler that is responsible for recognizing syntax.	(i)	Optimizer
(b)	A part of a compiler that takes as input a stream of characters and produces as output a stream of words along with their associated syntactic categories.	(ii)	Semantic Analysis
(c)	A part of a compiler that understand the meanings of variable names and other symbols and checks that they are used in ways consistent with their definitions.	(iii)	Parser
(d)	An IR-to-IR transformer that tries to improve the IR program in some way (Intermediate Representation).	(iv)	Scanner

Code :

	(a)	(b)	(c)	(d)
(1)	(iii)	(iv)	(ii)	(i)
(2)	(iv)	(iii)	(ii)	(i)
(3)	(ii)	(iv)	(i)	(iii)

1

56980624

Q) Consider the following statements: (GATE 2018)

(I) The output of a lexical analyzer is groups of characters.
(II) Total number of tokens in printf("i=%d, &i=%x", i, &i); are 11.
(III) Symbol table can be implementation by using array and hash table but not tree.

Which of the following statement(s) is/are correct?

a) Only 1
b) Only 2 & 3
c) All
d) None

Q) Consider the following statements: (GATE 2018)

(I) The output of a lexical analyzer is groups of characters.
(II) Total number of tokens in printf("i=%d, &i=%x", i, &i); are 11.
(III) Symbol table can be implementation by using array and hash table but not tree.

Which of the following statement(s) is/are correct?

a) Only 1
b) Only 2 & 3
c) All
d) None

What is left recursion?

If a non-terminal(A) finds non-terminal(A) itself then it is called left recursive.

A- > Aa

top-down parsing technique can not handle left recursive grammar so we convert left recursive to right recursive.

A -> Aa | a

=> A -> aA'

A'-> aA' | a

What is left factoring?

left factoring is used when a non-terminal finds the same terminal symbol as a prefix in more than one production rule.

A -> ab | ac

A -> aA'

A' -> A -> b|c

Join telegram For FREE Test Practice: https://t.me/RashmiCCS

Grammar (Generator)

↓

Language

↓

Automata (M/c) (Acceptor)

start symbol $S \rightarrow id$ } $\rightarrow E$ – expression

identifier

$E \rightarrow E + T \mid T$ → Term

$T \rightarrow T * F \mid F$ – factor

$F \rightarrow id$

Enroll for Free Class | Visit combinecs.com for all new courses | Contact Us +91-7666980624

Join telegram For FREE Test Practice: https://t.me/RashmiCCS

Combinecs

Ambiguous Grammar

A grammar is said to be ambiguous if there exists more than

1. **one leftmost derivation or**
2. **more than one rightmost derivation or**
3. **more than one parse tree (derivation tree. Or syntax tree) for the given input string.**

If the grammar is not ambiguous, then it is called unambiguous.

Exam: Try to make tree & check for reconfirmation.

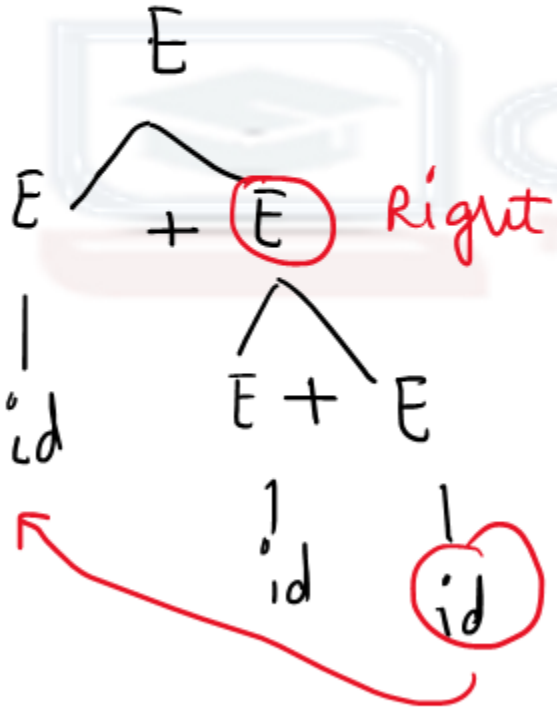
Enroll for Free Class | Visit combinecs.com for all new courses | Contact Us +91-7666980624

Join telegram For FREE Test Practice: https://t.me/RashmiCCS

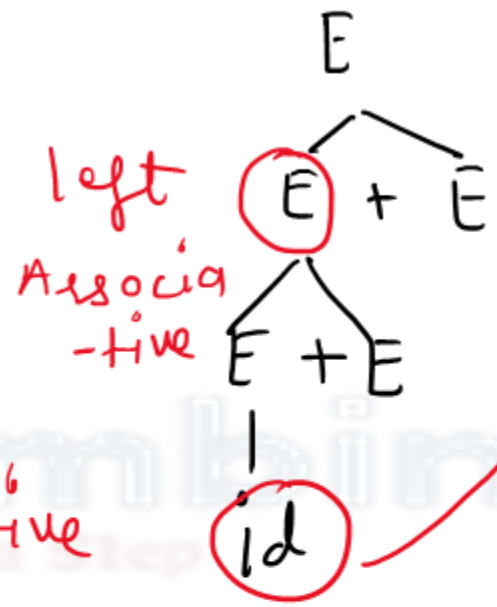
CombineCS

① Associativity Rule

$E \rightarrow E+E \mid E+E \mid id$



Right Associative



left Associa-tive

Note: Symbol is same in production & in '+' symbol associativity does not matter.

Hence, left Associative is Preferred.

Q) which is correct derivation tree?

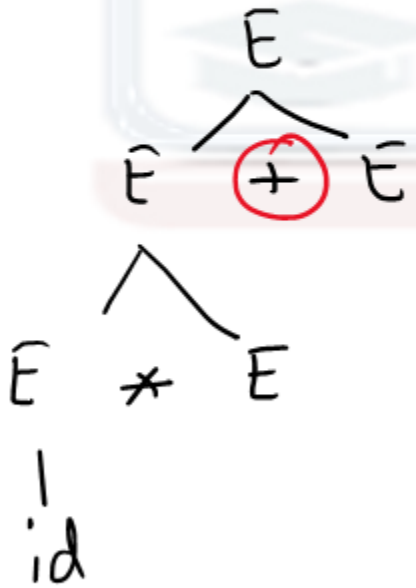
$\boxed{\text{Left Associative}}$

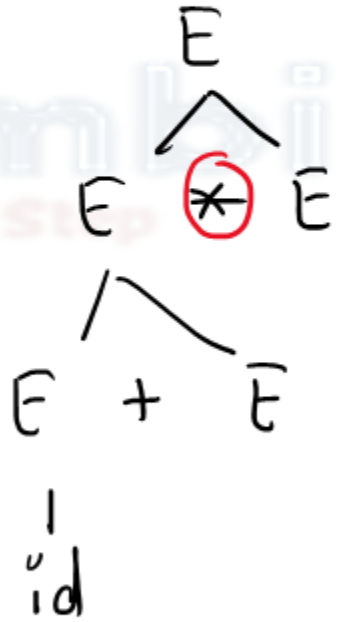
Enroll for Free Class | Visit combinecs.com for all new courses | Contact Us +91-7666980624

Join telegram For FREE Test Practice: https://t.me/RashmiCCS

$E \rightarrow E + E \mid E * E \mid id$

+, * high precedence is '*'





(1) Both are left or Right associative?

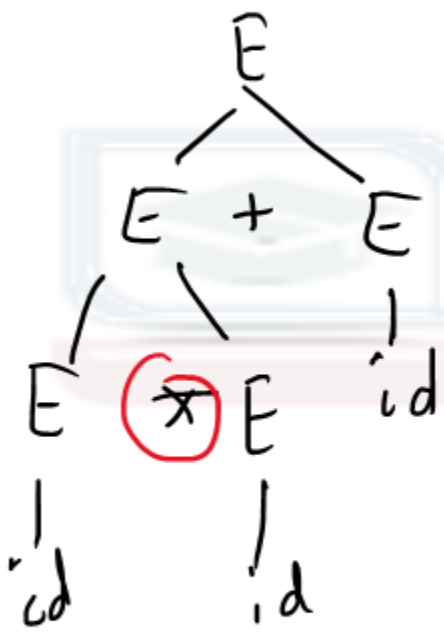
(2) which is correct derivation tree?

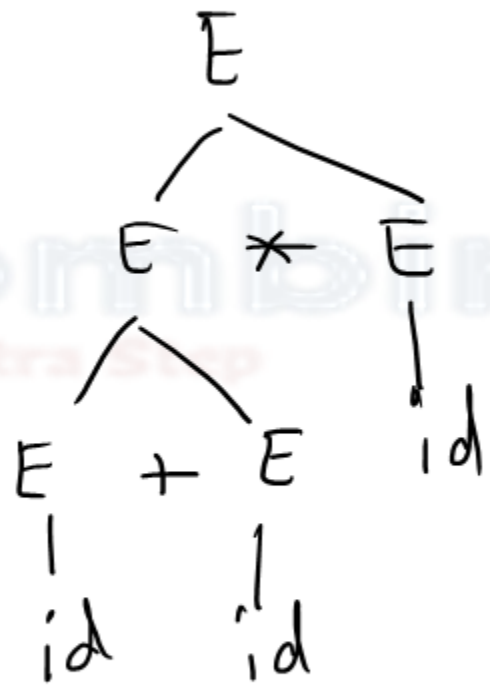
(3) Hint: operator which is far possess high precedence

Enroll for Free Class | Visit combinecs.com for all new courses | Contact Us +91-7666980624

Join telegram For FREE Test Practice: https://t.me/RashmiCCS

$E \rightarrow E+E \mid E*E \mid id$





Why?

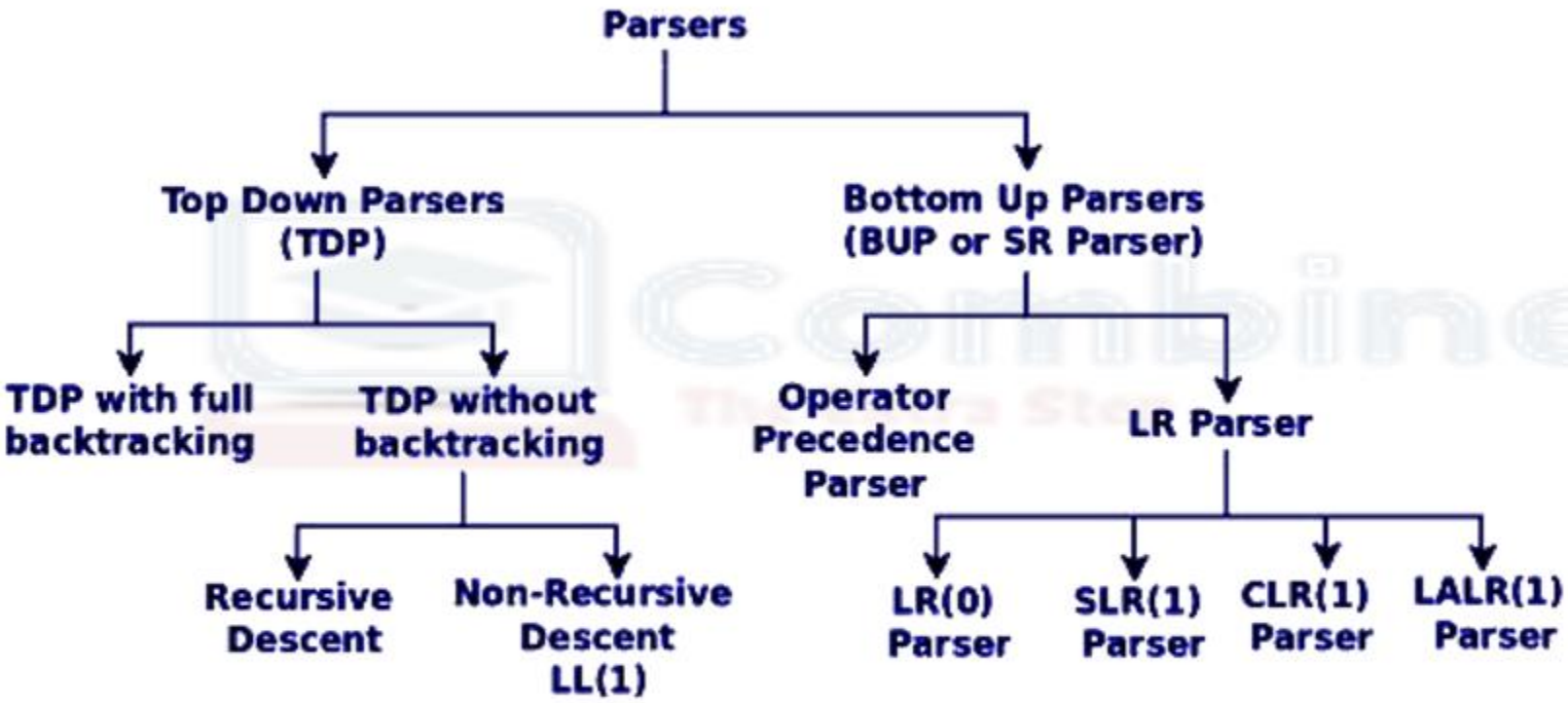
'*' symbol has high precedence than '+' and in the parse tree it is far from Start Symbol.

✓ correct derivation tree

Enroll for Free Class | Visit combinecs.com for all new courses | Contact Us +91-7666980624

Join telegram For FREE Test Practice: https://t.me/RashmiCCS

Combinecs



Enroll for Free Class | Visit combinecs.com for all new courses | Contact Us +91-7666980624

Join telegram For FREE Test Practice: https://t.me/RashmiCCS

Combinecs

Types of Parsing

1.Universal parser: It can parse any kind of grammar but not very efficient.
2.Top-down: Build the parse tree from the root to the leaves.
3.Bottom-up: Builds the parse tree by starting at the leaves and ending at root.

FIRST(A) is a set of the terminal symbols which occur as first symbols in string derived from A
FOLLOW(A) is the set of terminals which occur immediately after the nonterminal A in the strings derived from the starting symbol.

Enroll for Free Class | Visit combinecs.com for all new courses | Contact Us +91-7666980624

LL(1) Parser : LL(1) grammar is unambiguous, left factored and non left recursive.

To check whether a grammar is LL(1) or not :

1. If A -> B1 | C2 ⇒ { FIRST(B1) ∩ FIRST(C2) = ϕ }
2. If A -> B | ∈ ⇒ { FIRST(B) ∩ FOLLOW(A) = ϕ }

Enroll for Free Class | Visit combinecs.com for all new courses | Contact Us +91-7666980624

LL(1) Parser: LL(1) grammar is unambiguous, left factored, and non-left recursive. It is a top-down parser.

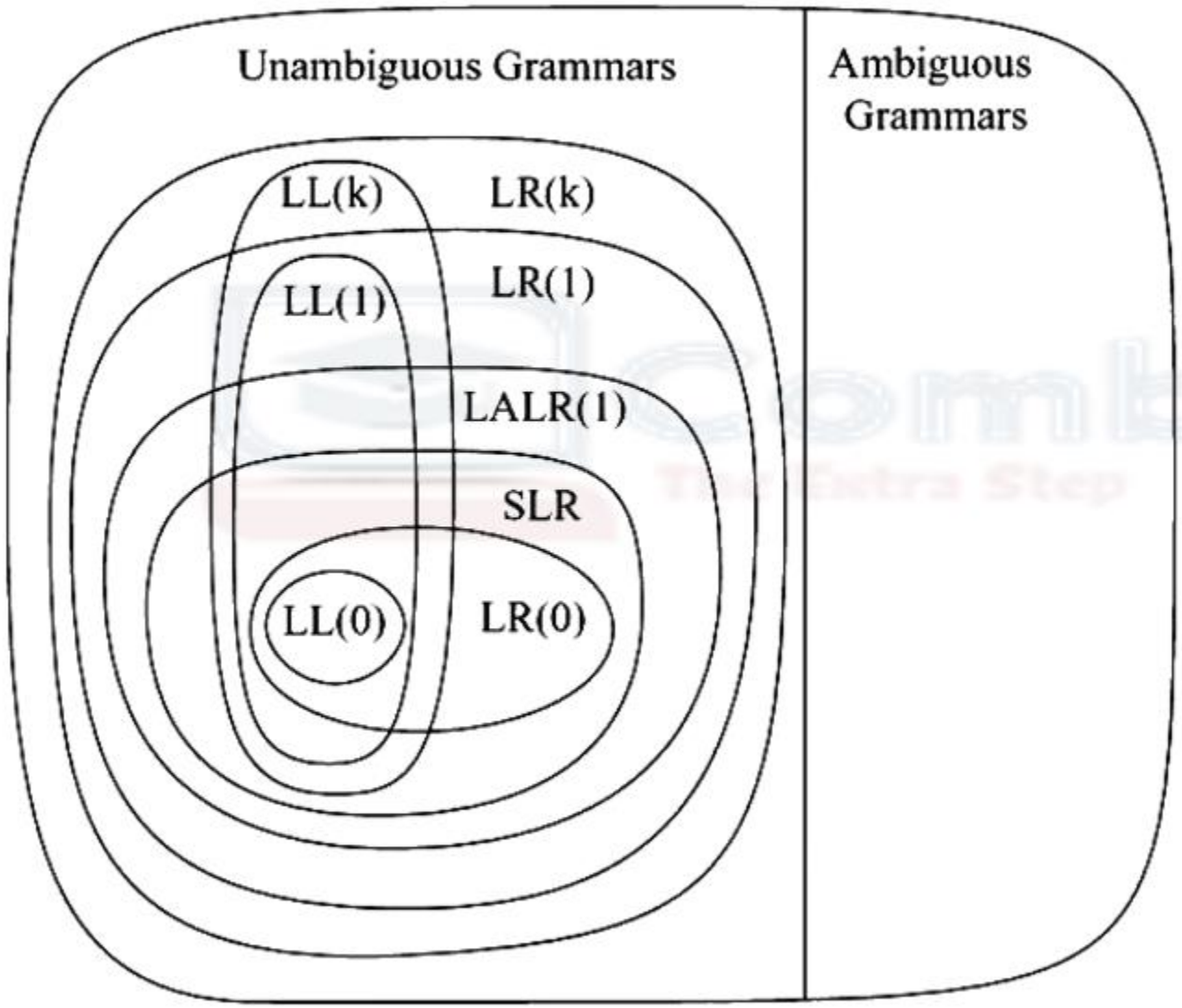
LR(0) Parser: It creates canonical form of LR(0) items.

SLR: It is more powerful than LR(0). Every LR(0) is SLR but every SLR is not LR(0).

CLR: CLR stands for canonical lookahead. CLR parsing uses the canonical collection of LR(1) items.

LALR: LALR stands for canonical lookahead LR. LALR parsing uses the canonical collection of LR(1) items.

Join telegram For FREE Test Practice: https://t.me/RashmiCCS



Enroll for Free Class | Visit combinecs.com for all new courses | Contact Us +91-7666980624

Q) Consider the following two statements:

P: Every regular grammar is LL(1)
Q: Every regular set has LR(1) grammar

Which of the following is TRUE?

a) Both P and Q are true
b) P is true and Q is false
c) P is false and Q is true
d) Both P and Q are false

Enroll for Free Class | Visit combinecs.com for all new courses | Contact Us +91-7666980624

Q) Consider the following two statements:

P: Every regular grammar is LL(1)
Q: Every regular set has LR(1) grammar

Which of the following is TRUE?

a) Both P and Q are true
b) P is true and Q is false
c) P is false and Q is true
d) Both P and Q are false

LL(1) parsers can recognize the regular grammars also LL(1) is subset of LR(1) or CLR grammar so it also recognizes regular sets. So both accept regular grammar.

Q) Which of the following cannot be used as an intermediate code form? (NET 2020)

(A)Quadruples
(B)Syntax trees
(C)Three address codes
(D)Post fix notation

Q) Which of the following cannot be used as an intermediate code form? (NET 2020)

(A)Quadruples
(B)Syntax trees
(C)Three address codes
(D)Post fix notation

Q) S -> (S)| a

Let the number of states in SLR(1), LR(1) and LALR(1) parsers for the grammar n1 n2 and n3 respectively.

a) n1 < n2 < n3

b) n1 = n3 < n2

c) n1 = n2 = n3

d) n1 $ n3 $ n2

Enroll for Free Class | Visit combinecs.com for all new courses | Contact Us +91-7666980624

Q) S -> (S)| a

Let the number of states in SLR(1), LR(1) and LALR(1) parsers for the grammar n1 n2 and n3 respectively.

a) n1 < n2 < n3

b) n1 = n3 < n2

c) n1 = n2 = n3

d) n1 $ n3 $ n2

Q) Which of the following statement is true?

a) SLR powerful than LALR
b) LALR powerful than Canonical LR parser
c) Canonical LR powerful than LALR parser
d) The parsers SLR= Canonical LR=LALR

Q) Which of the following statement is true?

a) SLR powerful than LALR
b) LALR powerful than Canonical LR parser
c) Canonical LR powerful than LALR parser
d) The parsers SLR= Canonical LR=LALR

Q) Replacement of an expensive operation by a cheaper one is called .....

(A)Reduction in strength
(B)Loop-invariant computation
(C)Code motion
(D)None of these

Enroll for Free Class | Visit combinecs.com for all new courses | Contact Us +91-7666980624

Q) Replacement of an expensive operation by a cheaper one is called .....

(A)Reduction in strength
(B)Loop-invariant computation
(C)Code motion
(D)None of these

Enroll for Free Class | Visit combinecs.com for all new courses | Contact Us +91-7666980624

Q) The parsing technique that avoids backtracking is .....

(A)Top-down parsing
(B)Recursive-descent parsing
(C)Predictive parsing
(D)Both (b) and (c)

Q) The parsing technique that avoids backtracking is .....

(A)Top-down parsing
(B)Recursive-descent parsing
(C)Predictive parsing
(D)Both (b) and (c)

Enroll for Free Class | Visit combinecs.com for all new courses | Contact Us +91-7666980624

Join telegram For FREE Test Practice: https://t.me/RashmiCCS

CombineCS

E	F	G	H
What we Deliver?	Why CombineCS?	Paid Class	Free Class
Content/Study Material			
	Structured & Quality Content	✓	✗
	Notes uploaded immediate after the class.	✓	✗
	Daily assignments	✓	✗
	100% 1-to 1 Doubt Resolution	✓	✗
	Last Minute Revision Notes	✓	✗
	Life Time Accessibility		
Interactive Live classes			
	Daily 4 hrs	✓	1 hour
	Recordings available, If missed & given prior information	✓	✓
	10+ PYQ Discussion NET/GATE	✓	✓
	Live Tests & Quizzes	✓	✓
	Personalized Test Analysis	✓	✗
	Monitoring your progress on a regular basis.	✓	✗
	Study Booster Sessions	✓	✓
100% Support Till Your Exam			
	Disciplined & Continuity	✓	✓
	24*7 you can drop your query.	✓	✗
	Daily Study Planner	✓	✗
	Short Tricks & Tips for Time Management	✓	✓
	85-90% Result/Success Story	✓	✗
	Career Guidance	✓	✓
	Motivational Support	✓	✓
If Exam not Cleared			
	Full support for Next time	✓	✗

Enroll for Free Class | Visit combinecs.com for all new courses | Contact Us +91-7666980624

Join telegram For FREE Test Practice: https://t.me/RashmiCCS

CombineCS
The Extra Step

Academic Enrichment And Online Tutoring

CombineCS
Online Classroom Program
UGC NET | GATE | KVS | NVS | UNIVERSITY EXAMS
Website: www.combinecs.com

Enroll now for Free Trial Session
20% OFF
Telegram

CombineCS The Extra Step
CUSTOMIZE CHANNEL
MANAGE VIDEOS
HOME VIDEOS PLAYLISTS COMMUNITY CHANNELS ABOUT

Suggested: Expected MCQ | UGC NET | SET | GATE CSE | ...

UGCNET Batch Course | Complete Details UGC NET 2022
23 views • 5 hours ago
🙏 Welcome to CombineCS - your one-stop solution for all CS/IT Competitive/University Exams. We will cover the entire syllabus, strategy, updates, and notifications which will help you to crack the UGC NET/SET/GATE/KVS/NVS/DSSSB/PGT/TGT/University Exams. 🎯
Buy our Paid Course...
READ MORE

DAILY JOB UPDATES
SUBSCRIBE NOW FOR DAILY UPDATES
Website: www.combinecs.com
website

CombineCS Jobs
SUBSCRIBED
HOME VIDEOS PLAYLISTS CHANNELS ABOUT

BYJU's is hiring
Work From Home
Multiple Job Updates
Any Graduate/PG/Experienced/Freshers
3:04

BYJU's is hiring for Work From Home | Freshers can apply | Salary upto Rs 30,000 per month
CombineCS Jobs • 44 views • 6 days ago
BYJU's has released the notification for recruitment to the Various posts. Those Candidates who are interested in the vacancy details & completed all eligibility criteria can watch full video...

s | Contact Us +91-7666980624

👉 Follow us on Social media:

▶ YouTube : https://www.youtube.com/c/CombineCSTheExtraStep
👥 Facebook : https://www.facebook.com/groups/combinecs
📸 Instagram : https://www.instagram.com/combinecs/
Telegram Group : https://t.me/RashmiCCS
Telegram Channel : https://t.me/combinecs

🌟 For all our latest courses launched
visit: 🌍 combinecs.com
👨‍💼 Student Support:- team.combinecs@gmail.com ☎ 7666980624
==========

"Effort Never Dies"
👍 Like || Share || Comment || Subscribe


Thank you
Post your doubts in comment section.
Stay subscribed for all updates.