CMP321: OPERATING SYSTEMS. Date . . No.

EXAM - 60%.

TESTS - 30%.

ATTENDANCE - 10%. NO ASSIGNMENT.

CONCURRENCY. {JOINING}

Occurs when there are several process threads running in parallel.

MESSAGE PASSING - communication method running process threads use to communicate.

CONCURRENT PROCESSES WITH SIMILAR PROBLEMS. { INTERLEAVED / OVERLAPPING

FACTORS: handling interrupts.
other processes' activities.
scheduling of processes.

Non-Atomic: operation depending on other processes.
Atomic: run independently of other processes.

13/10/2022 Date . . No.

COMPONENTS - Processes
Memory management

OPERATIONS (PROCESS) - Creation.
Scheduling
Execution
Deletion

CMP 313: AUTOMATA THEORY, COMPUTABILITY & FORMAL LANGUAGES.

LANGUAGE: any notation for communication of ideas or information. It doesn't need to be verbal i.e spoken. It can be a signal e.g ~~spatial~~ the wave of a hand "Good bye".

However, a programming language is a special notation for communicating with the computer system.

LANGUAGE AND SYNTAX

Every language displays instruction called it's Grammar or syntax eg a correct sentence always consist a subject followed by a ~~predicate~~ predicate

Sentence - Subject + PREDICATE.

Date . . No.

Subject = "Trinity" | "Esther"
Predicate = "eats" | "Talks"

LEXICON: is the list of valid words in a particular language.
SYNTAX: the rules that govern the combination of valid words in a language.
SEMANTICS: is the meaning associated with a particular syntactic entity
SYNTAX TREE: tree like diagram that depicts the structure of a sentence by breaking it down into its constituent parts.

"The big elephant ate the Peanut"

METHOD OF DRAWING THE TREE.

1. <Sentence> (root) → Top down vs Bottom up
<Subject> <Predicate>
<Article> <Noun phrase> <Verb> <Direct Object>
<Adjective> <Noun> <Article> <Noun>
the Big Elephant ate the Peanut

Internal node (< >) - roots, leaves.
External node (Leaves)

Date . . No.

PROPERTIES OF SYNTAX TREE.

- Roots are labelled by a start symbol.
- Each leave is labelled by a terminal symbol or null symbol usually "Σ".
- Each interior node is labelled by a non terminal symbol.
- It consist of one or more nodes.
- There is a production for none terminal node labelling some interior nodes e.g.:
If "A" is a non terminal node (interior node) with $X_1, X_2, X_3 \ldots X_n$
Then $A \longrightarrow X_1, X_2, \ldots X_n$ {Production}.
Hence, we see that <Sentence> is composed by <subject> and <Predicate>
while <subject> is composed by <article> and <noun phrase>

is the Syntax Tree

elements of a syntax tree is a syntax entity.

META LANGUAGE: language used to learn the structure of other languages.
E.g English used to teach French, hence English is a meta language.

NB: Computer trees are always inverted ????

Date . . No.

However, in computing "Backus Naur Form (BNF)
it is a notation for writing grammar that is commonly
used to specify the syntax of programming languages.
it is used to describe a programming language.

CMP 323

- Revisit Data Structures & Algorithms

JAVA is platform independent cos it relies on
virtual machines.

{32 bits per process cycle}.

SEMANTICS: is the meaning of associated with
specified syntactic entities.

SYNTAX: Structure.

SEMANTICS: Meaning.

COMPILED LANGS:

INTERPRETED LANGS:

16 &
32
64
2, 1

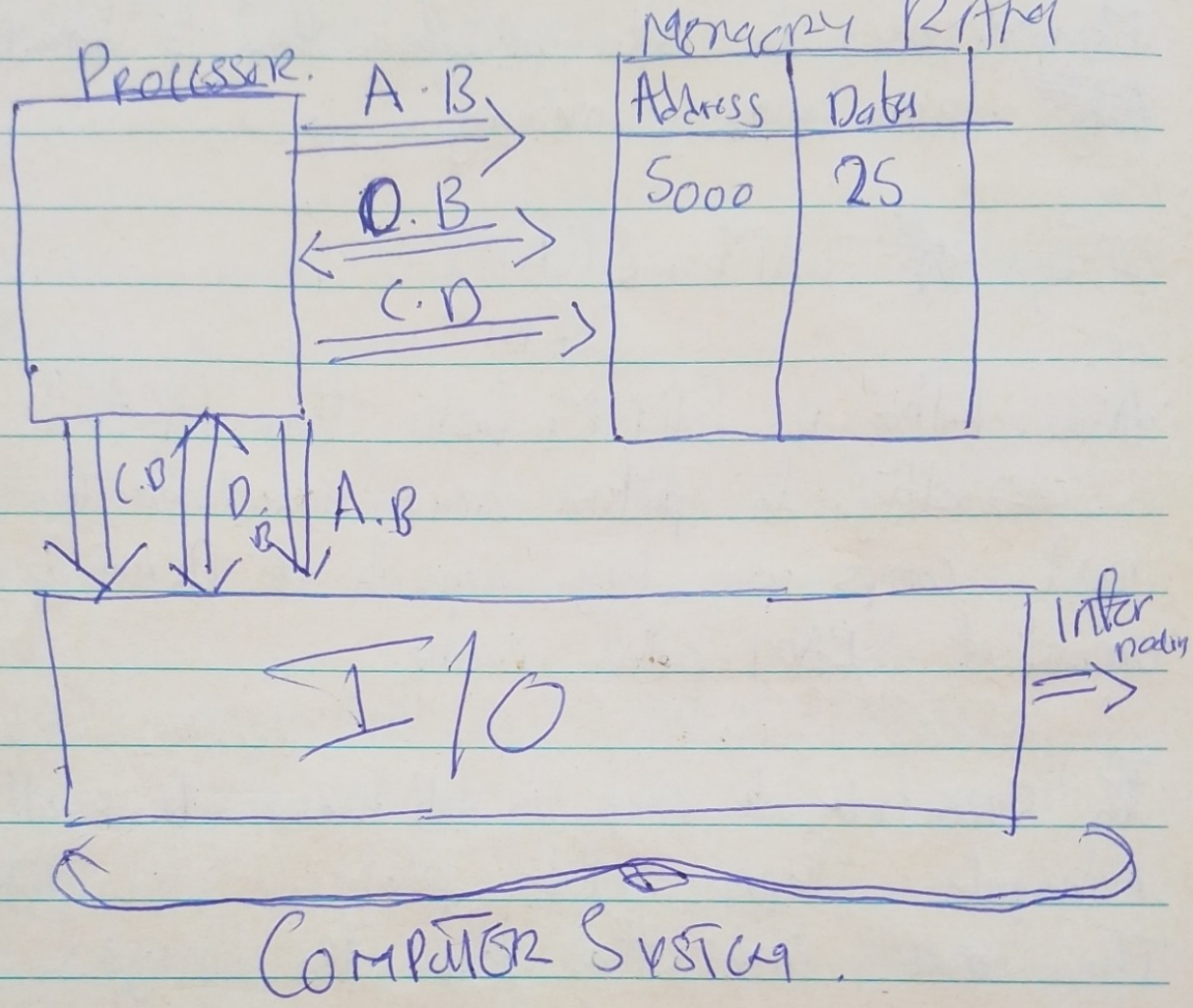
Javascript
node
Html

2 Theory & 3 Theory

Static variables / Dynamic Variables.

CMP 301: COMPUTER ARCHITECTURE

Date . . No.



Comp. Arch is the sign and arts of ~~comput~~ selecting & interalleling hardware components to create computers that meets functions, performance and cost goals.

It is the process of loosing a system component to meet user-desired needs.

Computer can be broadly classified into 3 basic components: I/O units who send data into and interacts from the system.

Date . . No.

Memory. used to store programs files & data files
. Internal / primary memory

2 Types RAM : SRAM
DRAM

Ass: Klhg is SRAM faster than DRAM
(2) Klhg do pictures occupy more space than songs.
most comps use large amounts of DRAM & small amount of SRAM called cache.

The screen of each comp is divided into small rectangular shapes called pixels.
The more pixel, the clearer the image.

For colored pictures, different numbers of pixels representing colors are used. To store bits of pixels, power of twos are used. Eg to store a 2 bit pixel
for 2bt pixel.
2 bits pixel 2^2 - 4 bits are used.
Processor: Fetches instruction, decodes it & executes it.
Bus: set of lines used to transfer data / information i/p o/p of computer memory.

Date . . No.

TYPES OF BASES: Address bus
Control bus
Data bus

Address: gives ~~the~~ determines address of location where data is to be stored

Data: puts data into the selected address location.

Control: ~~carrys data in & out of the memory~~.
sends instruction of wat to be done with the data in & out of & mem.

2 basic functions: rgt & read from memory.

PROCESSOR HISTORY: 1. 8085
2. 8086, 80186, 80286
80386, 80486
3. 80586: Pentium I
PII, PIII, PIV, C2O, C13
C15, C17, C19

CMP313. Date . . No.

ALPHABET: finite set of symbols

NB Comp. has only 2 alphabets {0, 1}.

Alphabet can be a character or digit.

NB ~~Alphabet can also be called~~ Σ

A FINITE STRING is a finite sequence of letters in an alphabet.

<Kleene star) denotes a set of all finite strings over the alphabet.

(Kleene Star(Σ*))

Also, Σ denotes the set of all infinite sequences.

E.g. Alphabet (0,1) generates strings (ε, 0, 1, 00, 11, 01, 10, 000, 001, 010 ... etc.) will be a Kleene star

ε - null

FORMAL THEORY.????

Let Σ alphabet (Σ) and a non-empty finite set/string. The element of Σ are all symbols or characters. A string or word over an alphabet (Σ) is any finite sequence of characters formed from Σ

If Σ (0,1) = (0101) then 0101 is a string over Σ. The length of an alphabet is the number of characters in that string. An empty string ε has

length zero.

SET OF STRINGS

If V is a set of strings, then V^* denotes the set of all finite strings of V, including the empty string. However if V is a set, then V^+ denotes set of all finite elements except ε.

Also two strings/words can be concatenated

i.e $\Sigma(0,1)$ (a, b, c)

-> (0a, 0b, 0c, 1a, 1b, 1c)

BACKUS NAUR FORM {BNF}.

In BNF, symbols ":: = =>" are used to denote "composed by"

Also Syntactic Variable are specified using < > and "|" means "Or".

EXAMPLE.

Use BNF as a meta language for the Syntax tree above. <sentence> ::= => <subject> <predicate>

<subject> ::= => <article> <noun phrase> - rule 2

<noun phrase> ::= => <adjective> <Noun> - rule 3

<predicate> ::= => <verb> <direct> <object> - rule 4

<direct object> ::= => <article> <Noun> - rule 5

Date . . No.

<article> ⇒ The rule6
<adjective> ⇒ big rule7
<verb> ⇒ ate rule8
<Noun> ⇒ Elephant / Peanut. rule9.

Using production ~~rules~~ rule we can check the length of a given ~~alphabet~~ word.

1) A GRAMMAR can contain more than one rule describing a syntactic entity
2) The length |x| of a given string
2) The purpose of grammar is to describe all the rules of the language.

~~Given~~

<Sentence> ⇒ <subject><predicate>
<subject> ⇒ ~~Ade~~ / He / I
<predicate> ⇒ run / sat / ate

WebApp???
built using payStack and word press

Date . . No.

<Sentence> ⇒ <Subject><predicate>

<Subject> ⇒ <article><Noun phrase>

~~<predicate> ⇒ ran/sat/ate~~

<article> ⇒

<Noun phrase> ⇒ <adjective><Noun>

CMP 321:

Deadlock vs Blocking

Bounce width - speed people leave the website.

Size & dimension shall be considered first when building a website. Cos it increases bandwidth cost

Name, mambas, business idea. build website using wordpress; one laptop at optimal performance to use.

Wamp.

create an account on paystack.

Search engine optimisation 4-5 max per group.

+1 for hosting the website..

CMP 325 Date . . No.

Multi-Threading

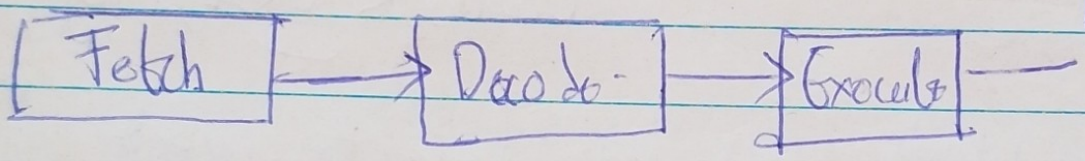
Data Structures } Algorithms { stacks, queues } sorting }

Design Patterns (MVC)

* Java Spring Framework * projects based

CMP 301 : Computer Architecture : Buses. } related

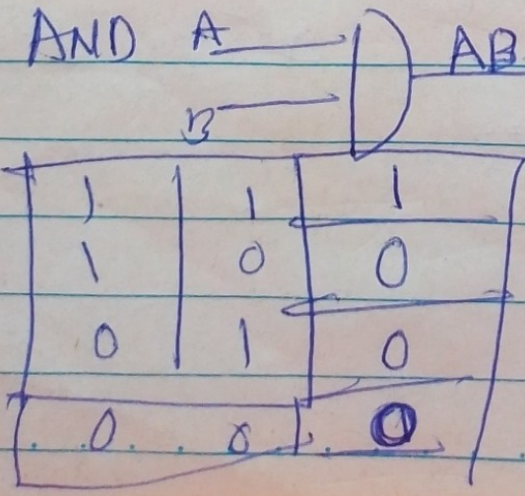
Instruction Cycle



~~Boolean~~ Gates. Logic Gates

Logic gates are basic building blocks for any digital system. It is an electronic circuit having 1 or more input & only 1 output.

Basic Logic Gate: AND, OR, NOT.

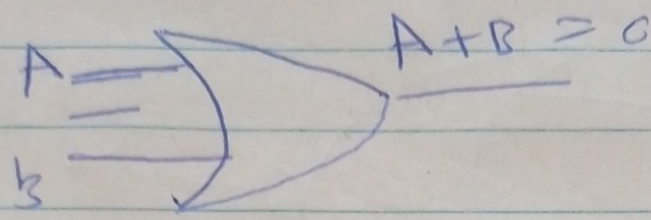


Truth Table

T	T	T
T	F	F
F	T	F
F	F	T

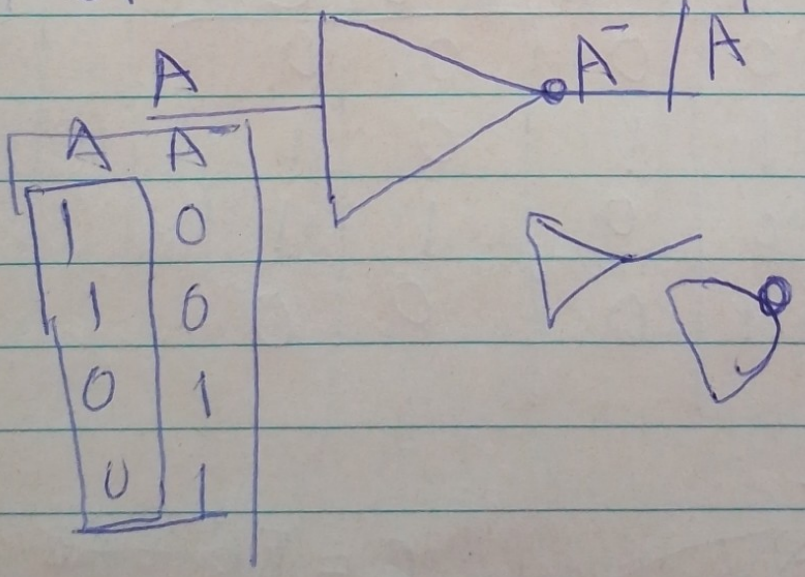
Date . . No.

OR

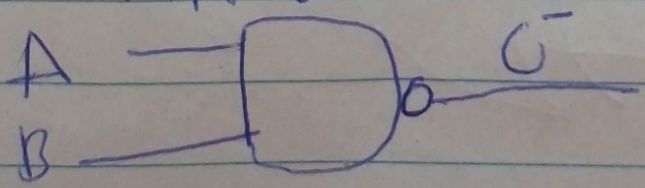


A	B	A+B = C
1	1	1
1	0	1
0	1	1
0	0	0

NOT

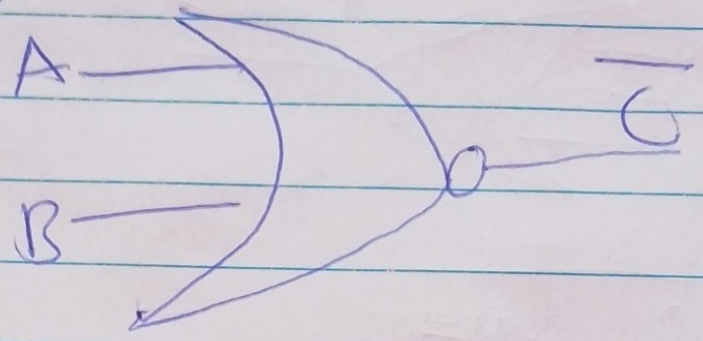


OTHERS : NAND

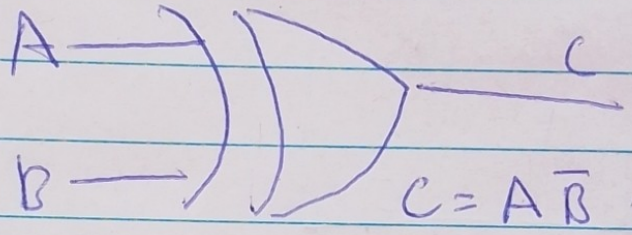


Date . . No.

NOR GATE



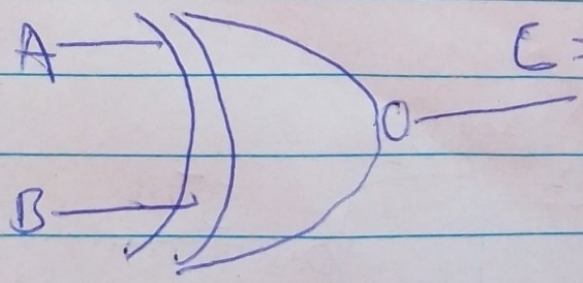
Ex-OR



$C = A\bar{B} + \bar{A}B$

A	B	$\bar{A}$	$\bar{B}$	$A \cdot \bar{B}$	$\bar{A} \cdot B$	C
1	1	0	0	0	0	0
1	0	0	1	1	0	1
0	1	1	0	0	1	1
0	0	1	1	0	0	0

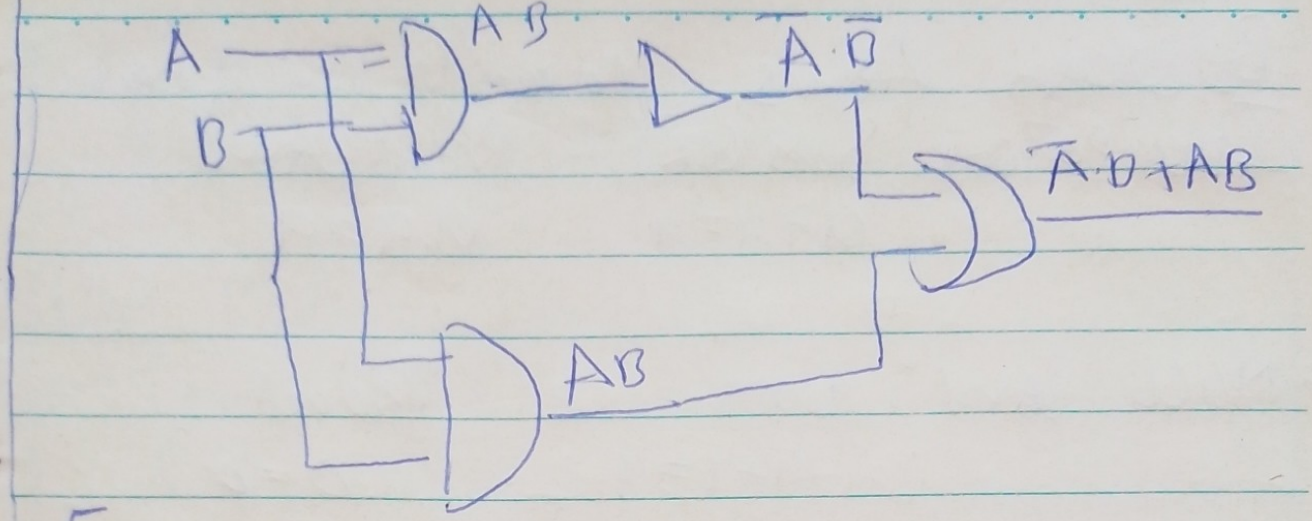
Ex - NOR



$C = \bar{A}\bar{B} + AB$

1
0
0
1

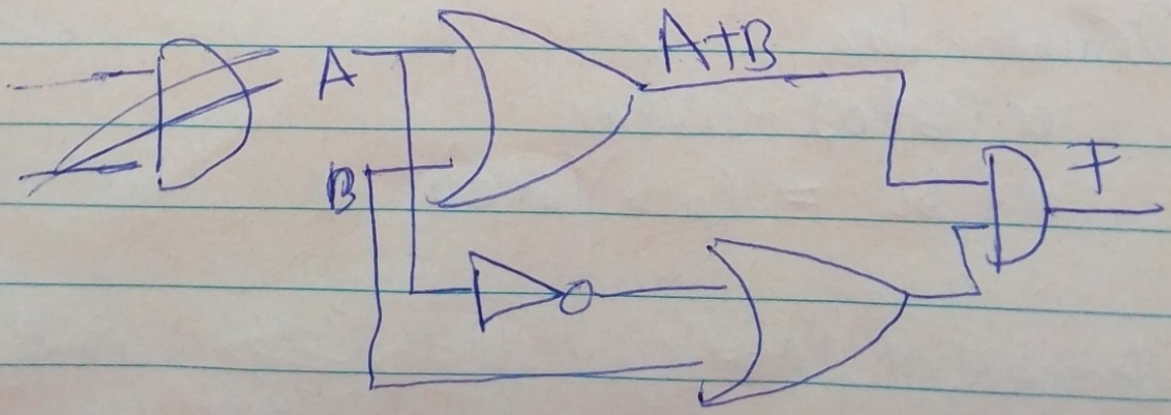
Date . . No.



Example.

$F = (A+B) \cdot (A'+B)$

A	B	A'	A+B	A'+B	(A+B)(A'+B)
1	1	0	1	1	1
1	0	0	0	1	0
0	1	1	1	1	1
0	0	1	1	0	0



Date . . No.

BOOLEAN LAWS: No need to prove at exam.

IDEMPOTENT	SUM (OR)	Product (AND)
	$a+a=a$	$a.a=a$

2. NULL LAW: SUM $a+1=1$ PRODUCT $a.0=0$

3. IDENTITY LAW

$a+0=a$ $a.1=a$

4. COMMUTATIVE LAW

$a+b=b+a$ $ab=ba$

5. Associative laws

$(a+b)+c=a+(b+c)$ $(a.b)c=a(bc)$

6. Distributive law.

$a(b+c)=ab+ac$

$a+bc=(a+b)(a+c)$

7. ABSORPTION LAW

$a+ab=a$ $a(a+b)=a$

Date . No.

8 Complement law

$a + \bar{a} = 1 \qquad a \cdot \bar{a} = 0$

9 DE- MORGAN'S LAW

$\overline{(a+b)} = \bar{a} \cdot \bar{b}$

$\overline{(a \cdot b)} = \bar{a} + \bar{b}$

10 INVOLUTION

$(a')' = a$

SIMPLIFICATION OF BOOLEAN EXPRESSION

EMP 313: Given the production rules below, use canonical parse tree, check if this sentence is a valid word.

Rules: <Decimal.No> ⟹ <point><number>

<No><point><no>

<Number> ⟹ <digit> | <Number><digit>

<digit> ⟹ 0|1|2|3|4|5|6|7|8|9

<point> ⟹ .

(33) 142 Parse

Date . . No.

3.142

<No> ⇒ 3

<point> ⇒ .

<Number> ⇒ 142

<~~dig~~ <N> <point> ⇒ 142

<No> <point> <digit> 42

<No> <point> <digit> <digit> 2

<No> <point> <digit>

GRAMMAR is a set of rules by which valid sentences are constructed.

Formal Grammar (G) is designed to be a set of quadruple (V, T, S, P)

where V is a set of non-terminal symbols

T is a set of terminal symbols

S is a set of start symbols

P is a production rules

~~Example~~: Odd <No> ⇒ <No> <Digit>

<Digit>

<No> ⇒ <No> <Digit> | <digit>

~~<digit>~~ <Digit> ⇒ 0 | 2 | 4 | 6 | 8 | <digit>

<digit> ⇒ 1 | 3 | 5 | 7 | 9

Date . . No.

From ⇐ above, symbols on the left side is these inside the bracket or non-terminal symbols V, symbols on the right, not in bracket are terminal represented by T,

NB any production rule that start the production is a start Symbol; S

and the four laws are the rules the guide the generation of the sentence (P).

NB: Sometimes, apart from BNF notation, roman letters are used as non-terminal symbols while small alphabet are used as terminal symbols.

$S \Rightarrow AB$

$A \Rightarrow A/a$

$B \Rightarrow Bb/b$

S, A & B are non-terminal symbols while a, b, are terminal symbols.

DERIVATION OF GRAMMAR

Let G be a grammar which says V is a correct derivation of x

$\therefore V \Rightarrow x$

Date . . No.

$V \Rightarrow xUy$

$x \Rightarrow uwy$

x is a direct derivation of V.

$\alpha \Rightarrow \beta$ (direct derivation with 1 production)

$\alpha \Rightarrow {}^{+}\beta$ (one or more production rules)
+ (many other production)

$\alpha \Rightarrow {}^{*}\beta$ (0 or more productions

$\alpha \Rightarrow \beta$ $\quad V \Rightarrow a, b, \epsilon$

$\beta \Rightarrow V$

Types Of Grammar

According to chomsky, there are 4 basic types of grammar: Type 0
Type 1
Type 2
Type 3

Assignment
Note
+ Example

GRACE-OKOYE KAMASI
BHU/20/04/05/0094

CMP 318: AUTOMATA THEORY.

Type 0: include all formal grammars. Its grammar is Unrestricted i.e there productions have no restrictions. Type-0 generates recursively enumerable languages.

Production can be in form of $\alpha \longrightarrow \beta$

where α - string of terminals & non-terminals with at least 1 non-terminal and α cannot be null. - Understood by a Turing machine.

β - string of terminals and non-terminals.

EXAMPLE

$S \longrightarrow ACaB$

$Bc \longrightarrow aCB$

$CB \longrightarrow DB$

$aD \longrightarrow Db$

Type 1: generate context-sensitive languages. Production must be in the form.

$$\alpha A \beta \longrightarrow \alpha V \beta$$

where $A \in N$ (i.e A are non-terminal)

& $\alpha, \beta, V \in (T \cup N)$ (i.e α, β, V are strings of terminals & non-terminals)

Strings α & β may be empty, but V must be non-empty.

- Understood by a linear bounded automaton

EXAMPLE

$AB \longrightarrow AbBc$

$A \longrightarrow bcA$

$B \longrightarrow b$

Type 2: generate context-free languages. The Production must be in the form.

$$A \longrightarrow V$$

where A is non-terminal

and V is a string of terminals & non-terminals.

E.g $S \longrightarrow Xa$ - Understood by non-deterministic pushdown

$X \longrightarrow a$ - automata.

$X \longrightarrow aX$

$X \longrightarrow abc$

$X \longrightarrow \varepsilon$

Type 3 : generate regular languages. The production must be in form

$$X \longrightarrow a \quad \text{OR}$$

$$X \longrightarrow a\,Y$$

∴ Type 3 grammars must have a single non-terminal on the left-hand side not having a single terminal or a single terminal followed by a non-terminal