COSC1107 Computing Theory

(We will commence soon. We are just allowing a few minutes for people to join and set up. *Please mute your microphone unless you are speaking.* You can raise your hand or use the chat at any time.)

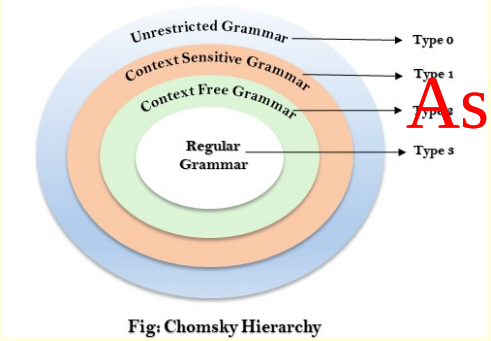
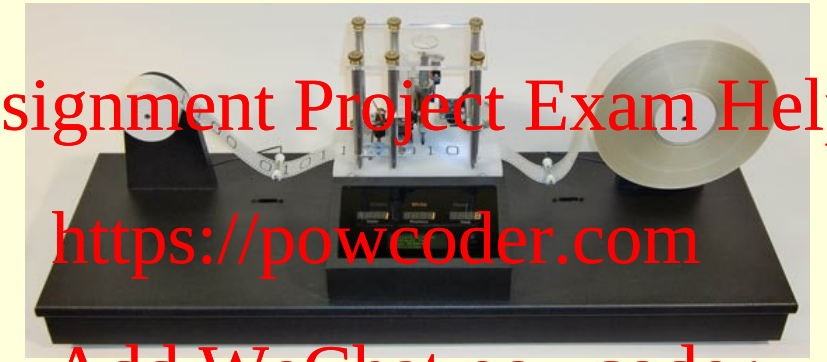
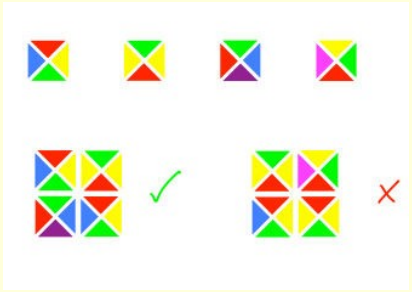
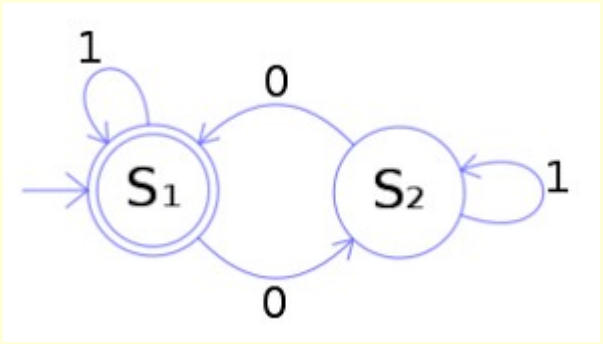
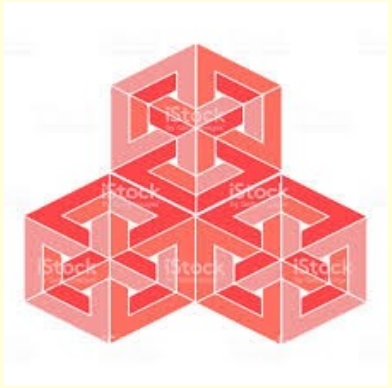


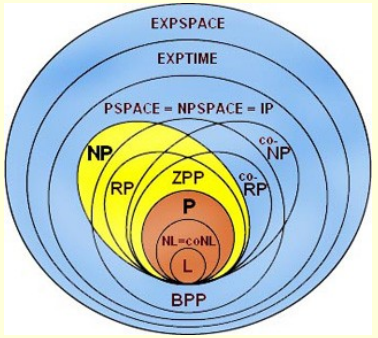
Fig: Chomsky Hierarchy

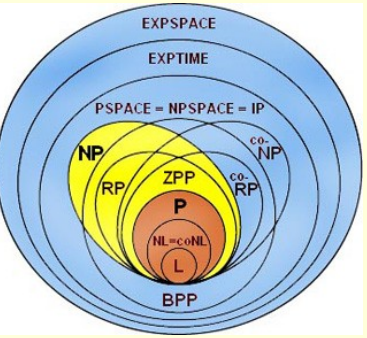


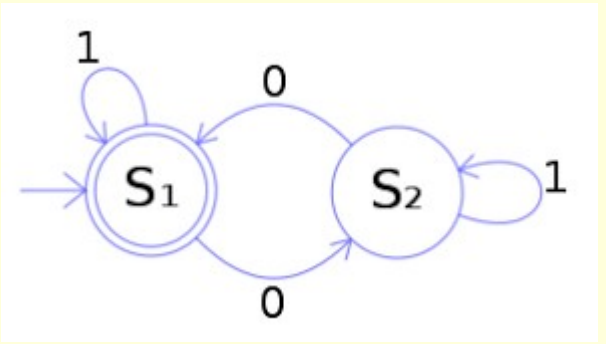










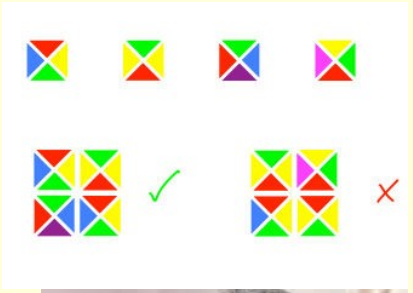


COSC1107

Computing Theory

Chomsky Hierarchy

Week 8



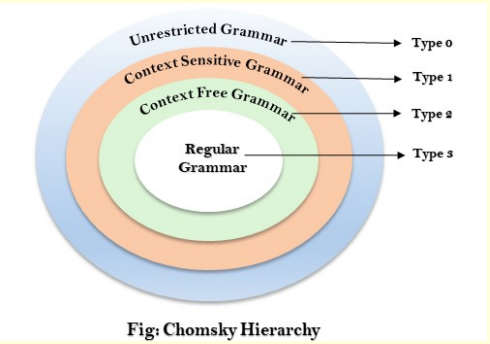


Fig: Chomsky Hierarchy

James Harland

james.harland@rmit.edu.au

* With thanks to Sebastian Sardina

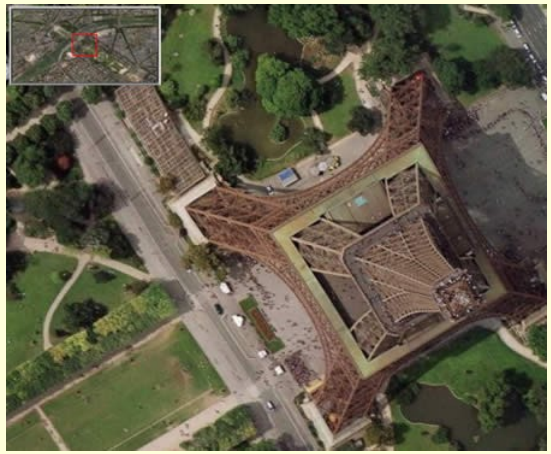
Intro music 'Far Over' playing now …

Acknowledgement

RMIT University acknowledges the people of the Woi wurrung and Boon wurrung language groups of the eastern Kulin Nations on whose unceded lands we conduct the business of the University. RMIT University respectfully acknowledges their Ancestors and Elders, past and present.

RMIT also acknowledges the Traditional Custodians and their Ancestors of the lands and waters across Australia where we conduct our business.

(add your name here to volunteer for this or email me)

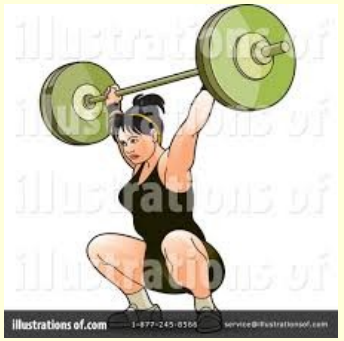


Overview

- Questions?
- Unrestricted Grammars
- Questions?
- Context-sensitive Grammars
- Questions?
- Linear Bounded Automata
- Questions?
- Platypus Game ⟵ Of course!
- Questions?

Overview

Week 5	Undecidability
Week 6	NFA -> DFA conversion, Pumping Lemma for regular languages
Week 7	Grammar normal forms, Pumping Lemma for context-free languages, DPDAs vs NPDAs
Week 8	Unrestricted Grammars, Context-Sensitive Grammars, Linear Bounded Automata



Chomsky Hierarchy

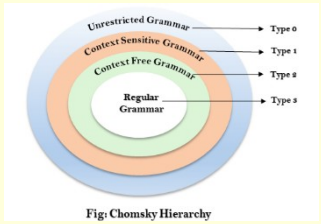


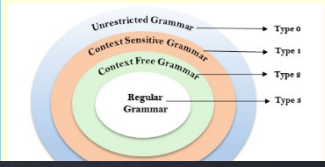
Fig: Chomsky Hierarchy

Key concept	Basis
Universal Turing machines	Encoding of TMs as input to TMs
Halting problem undecidable	Proof by contradiction
Reductions to other problems	More proofs by contradiction
NFA to DFA conversion	Nondeterminism 'compiled out'
Pumping Lemma for Regular Languages	Bounded #states in DFA Generally used in proofs by contradiction
Grammar Normal forms	Grammar restructuring
Pumping Lemma for Context-free languages	Bounded #variables in CFG Generally used in proofs by contradiction
Technical observations	Proof techniques

Questions?

Questions?

Questions?



Computing Theory Topics

Chomsky Hierarchy

$L(G) = \Sigma^*$

Undecidable

Turing machines

Undecidable

Linear bounded TM

$L(G) =$

ND Pushdown automata

"I love a Turing-machine-free environment!

Finite state machines

Regular expressions

even 1's

Context-free

wcw^R

"No pesky Turing machines out here!"

Context sensitive

$a^nb^nc^n$

$L(G_1) = L(G_2)$

Halting problem

Unrestricted

{(M,w)|M accepts w}

Chomsky Hierarchy

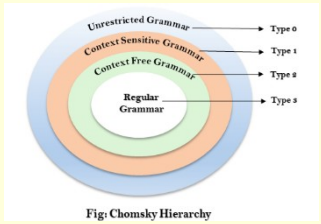


Fig: Chomsky Hierarchy

Automata	Languages	Grammars
	Undecidable languages	---
Turing Machines	Recursively enumerable languages	Unrestricted grammars
Linear Bounded Automata	Context-sensitive languages	Context-sensitive grammars
(Nondeterministic) Pushdown Automata	Context-free languages	Context-free grammars
Deterministic Pushdown Automata	?? (Deterministic CF?)	???
Nondeterministic Finite Automata & Deterministic Finite Automata	Regular languages	Regular grammars & regular expressions

Week 8

Week 7

Week 6

Grammars

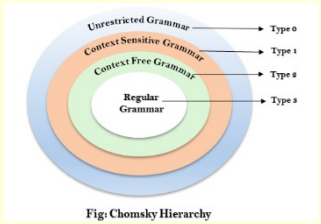


Fig: Chomsky Hierarchy

A rule is of the form $(V \quad T)^+ \quad (V \quad T)^*$

non-empty string over V T

string over V T

$L \quad R_1 \mid R_2 \mid \ldots \mid R_n$

Different types of grammars result from different restrictions on L and R_i

Name	Type	Constraint	Example
Unrestricted	0	--	AbC AC
Context-sensitive	1	$\|L\| \leq \|Ri\|$ or S	AbC abc
Context-free	2	$\|L\| = 1$	A AC
Regular	3	$\|L\| = 1$ (ie L V) and R_i T {} TV	A a A bB A

Chomsky Hierarchy

Automata	Languages	Grammars
---	*Undecidable languages*	---
Turing Machines	Recursively enumerable languages	Unrestricted grammars
Linear Bounded Automata	Context-sensitive languages	Context-sensitive grammars
(Nondeterministic) Pushdown Automata	Context-free languages	Context-free grammars
Deterministic Pushdown Automata	?? (Deterministic CF?)	???
Nondeterministic Finite Automata & Deterministic Finite Automata	Regular languages	Regular grammars & regular expressions

Problem Reduction

Closure properties

Pumping Lemma

Questions?

Questions?

Questions?

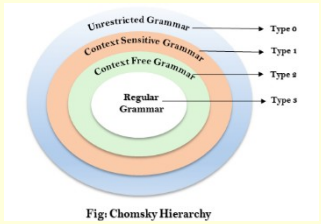


Fig: Chomsky Hierarchy

Unrestricted Grammars

- Like context-free grammars but more intricate
- Context-sensitivity means that rearrangements can be made (not just replacement)
- Nondeterminism means the correct choice of order becomes more important
- Need to think more 'algorithmically' when constructing grammars

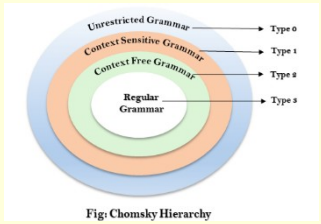


Fig: Chomsky Hierarchy

Example

S $\rightarrow$ ABCS | T_c Generate same number of As,Bs,Cs

CA $\rightarrow$ AC
BA $\rightarrow$ AB
CB $\rightarrow$ BC

Sort them alphabetically …

$CT_c \rightarrow T_c c \mid T_b c$ Move T_c to left, turn C into c

$BT_b \rightarrow T_b b \mid T_a b$ Move T_b to left, turn B into b

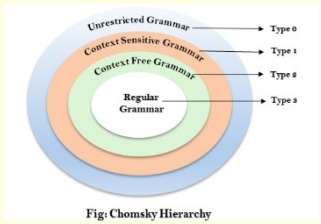
$AT_a \rightarrow T_a a$ Move T_a to left, turn A into a

What is the language of this grammar?

T_a

Done!

Example



S ABCS ABCABCS ABCABCT_c

ABACBCT_c AABCBCT_c AABBCCT_c

AABBCT_cc

AABB$T_b$cc

AAB$T_b$bcc

AA$T_a$bbcc

A$T_a$abbcc

$T_a$aabbcc

aabbcc

S ABCS | T_c

CA AC
BA AB
CB BC

CT_c T_cc | T_bc
BT_b T_bb | T_ab
AT_a T_aa

T_a

What is the language of this grammar?

$L = \{a^n b^n c^n \mid n \geq 1\}$

How do I know it *only* generates strings of this form?

Example

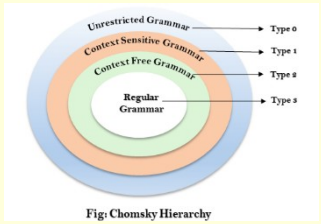


Fig: Chomsky Hierarchy

S ABCS | T_c

CA AC
BA AB
CB BC

CT_c $T_c c$ | $T_b c$
BT_b $T_b b$ | $T_a b$
AT_a $T_a a$

T_a

S $\underline{ABCS}$ AB$\underline{CABCS}$ A$\underline{BAC}$BCS

A$\underline{AB}$CBCS AABCB$\underline{CT_c}$

AABCB$\underline{T_c c}$ AAB$\underline{BC}$ T_cc

AAB$\underline{BT_b}$cc

AAB$\underline{T_a b}$cc

AABbcc

$L = \{a^n b^n c^n \mid n \geq 1\}$

Example

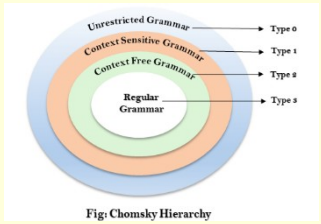


Fig: Chomsky Hierarchy

S ABCS | T_c

CA AC
BA AB
CB BC

CT_c $T_c c$ | $T_b c$
BT_b $T_b b$ | $T_a b$
AT_a $T_a a$

T_a

S ABCS ABCABCS ABACBCS

AABCBCS AABCBCT_c

AABCB$T_c c$ AABB$T_c c$

AABB$T_b cc$

AAB$T_b bcc$

A$AT_a bbcc$

$AT_a abbcc$

$T_a aabbcc$
aabbcc

$L = \{a^n b^n c^n \mid n \geq 1\}$

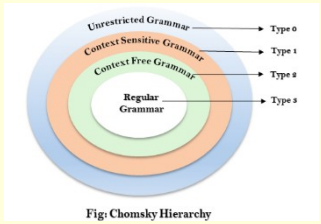


Fig: Chomsky Hierarchy

Example

S ABCS | T_c

S <u>ABC</u>S ABC<u>ABC</u>S ABCAB<u>C</u>T_c

ABCABT_cc AB<u>AC</u>BT_cc

ABAB<u>C</u>T_cc AB<u>AB</u>$T_c$cc

AABB$T_c$cc

CA AC
BA AB
CB BC

CT_c T_cc | T_bc
BT_b T_bb | T_ab
AT_a T_aa

T_a

$L = \{a^n b^n c^n \mid n \geq 1\}$

Questions?

Questions?

Questions?

Unrestricted Grammar

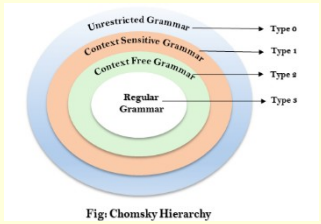


Fig: Chomsky Hierarchy

Give an unrestricted grammar for the language

$L = \{www \mid w \in \{a,b\}^*\}$

???

S → AS | BS | ε
A → a
B → b

$\{w \mid w \in \{a,b\}^*\}$

ababab, baabaabaa, abbaabbaabba, ...

Strategy:

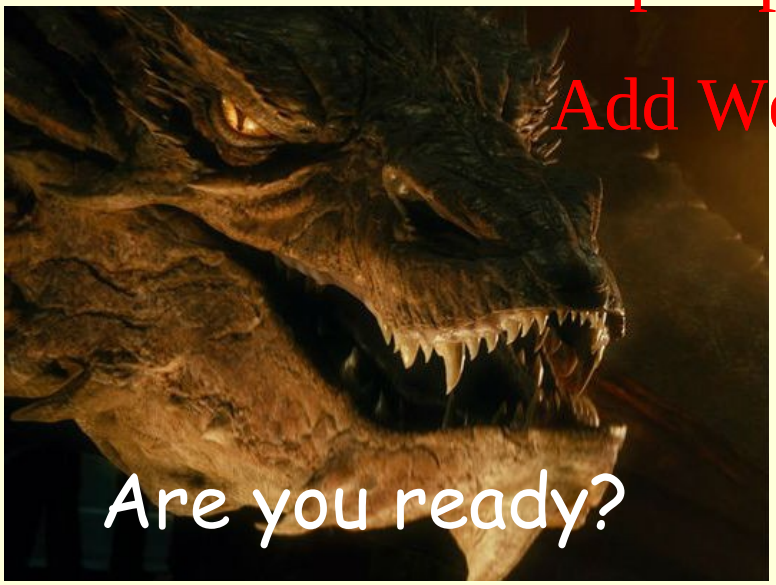
1. Generate three copies of "placeholders" for a &b
2. Sort the copies (first copies to left, third copies to right, second copies to middle)
3. Replace placeholders with a & b as appropriate
4. Put the kettle on and make a cup of tea!

Have a go!

- Give an unrestricted grammar for the language

  $L = \{www \mid w \in \{a,b\}^*\}$

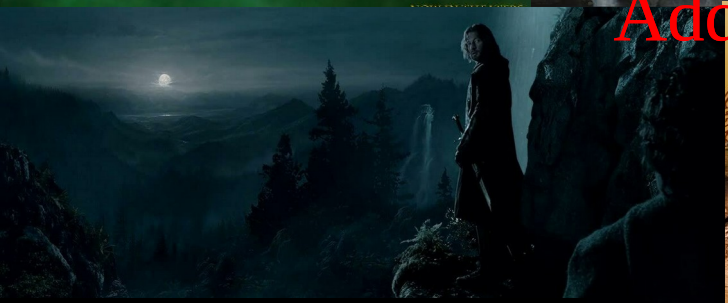
- Consult with other students if you wish
- Time limit will be 10 minutes

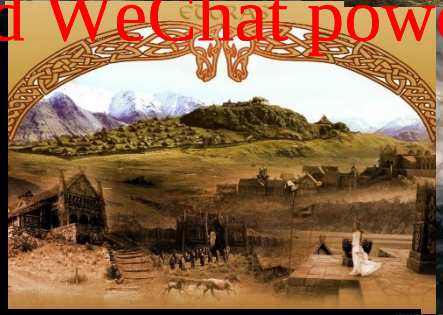


Go!

The pictures will take 10 minutes to disappear!

Thomas music means 1 minute left!





Unrestricted Grammar

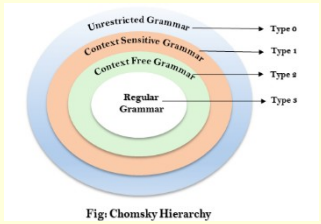


Fig: Chomsky Hierarchy

Give an unrestricted grammar for the language
L = {www | w {a,b}* }

???

S AS | BS |
A a
B b
{w | w {a,b}* }

ababab, baabaabaa, abbaabbaabba, ...

How did you go?
Complete solution?

(remember, JFLAP is your friend ...)

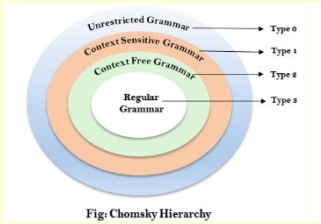


Fig: Chomsky Hierarchy

Unrestricted Grammar

S $A_1A_2A_3S$ | $B_1B_2B_3S$ | T_3

$L = \{www \mid w \ \{a,b\}^*\}$

A_2A_1 A_1A_2
A_3A_1 A_1A_3
A_3A_2 A_2A_3
B_2B_1 B_1B_2
B_3B_1 B_1B_3
B_3B_2 B_2B_3
A_2B_1 B_1A_2
A_3B_1 B_1A_3
A_3B_2 B_2A_3
B_2A_1 A_1B_2
B_3A_1 A_1B_3
B_3A_2 A_2B_3

A_1T_1 T_1a
A_2T_2 T_2a | T_1a
A_3T_3 T_3a | T_2a
B_1T_1 T_1b
B_2T_2 T_2b | T_1b
B_3T_3 T_3b | T_2b
T_1

S $A_1A_2A_3S$
$A_1A_2A_3A_1A_2A_3S$
$A_1A_2A_3A_1A_2A_3B_1B_2B_3S$
$A_1A_2A_3A_1A_2A_3B_1B_2B_3S$
$A_1A_2A_1A_2A_3A_3B_1B_2B_3S$
$A_1A_1A_2A_2A_3A_3B_1B_2B_3S$
...
$A_1A_1B_1A_2A_2B_2A_3A_3B_3S$
$A_1A_1B_1A_2A_2B_2A_3A_3B_3T_3$
$A_1A_1B_1A_2A_2B_2A_3A_3T_3b$
...
Taabaabaab
aabaabaab

ababab,
baabaabaa,
abbaabbaabba,
...

Unrestricted Grammar

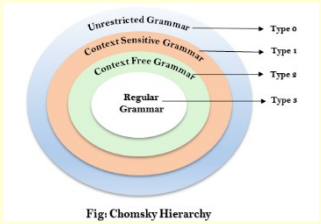


Fig: Chomsky Hierarchy

S $A_1A_2A_3S$ | $B_1B_2B_3S$ | T

$L = \{www \mid w \ \{a,b\}^* \}$

A_2A_1 A_1A_2
A_3A_1 A_1A_3
A_3A_2 A_2A_3
A_2B_1 B_1A_2
A_3B_1 B_1A_3
A_3B_2 B_2A_3
A_2B_1 B_1A_2
A_3B_1 B_1A_3
A_3B_2 B_2A_3
B_2B_1 B_1B_2
B_3B_1 B_1B_3
B_3B_2 B_2B_3

A_1 a
A_2 a
A_3 a
B_1 b
B_2 b
B_3 b
T

S $A_1A_2A_3S$

$A_1A_2A_3A_1A_2A_3S$

$A_1A_2A_3A_1A_2A_3B_1B_2B_3S$

$A_1A_2A_3A_1A_2A_3B_1B_2B_3T$

$A_1A_2A_1A_2A_3A_3B_1B_2B_3$

$aA_1A_2A_2A_3A_3B_1B_2B_3$

...

aaaaaabbb (!!)

abababa,
baabaabaa,
abbaabbaabba,
...

Too simple!

"Don't generate too much!"

Chomsky Hierarchy

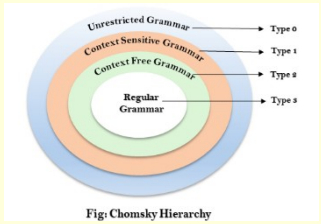


Fig: Chomsky Hierarchy

Automata	Languages	Grammars
---	*Undecidable languages*	---
Turing Machines	Recursively enumerable languages	Unrestricted grammars
Linear Bounded Automata	Context-sensitive languages	Context-sensitive grammars
(Nondeterministic) Pushdown Automata	Context-free languages	Context-free grammars
Deterministic Pushdown Automata	?? (Deterministic CF?)	???
Nondeterministic Finite Automata & Deterministic Finite Automata	Regular languages	Regular grammars & regular expressions

UG to Turing Machine

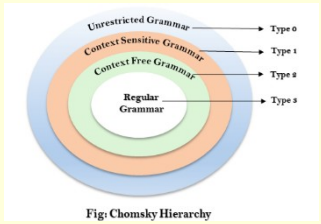


Fig: Chomsky Hierarchy

- Use a two-tape non-deterministic Turing machine
- Tape 1 contains input w (which is never changed)
- Tape 2 simulates the derivation

Turing machine works like this

1. Write S on Tape 2
2. If w is on Tape 2 then halt
3. Update Tape 2 according to the grammar rules
4. Go to 2.

UG to Turing Machine

S ABCS
ABCABCS
ABCABCT_c
ABACBCT_c
AABCBCT_c
A$AB$$BCCT_c$
AABBCT_cc
AAB$B$$T_b$cc
AA$B$$T_b$bcc
A$A$$T_a$bbcc
A$T_a$abbcc
$T_a$aabbcc
aabbcc

aabbcc
S
aabbcc
ABCS
aabbcc
ABCABCS
aabbcc
ABCABCT_c
aabbcc
ABACBCT_c
⋮
aabbcc
$T_a$aabbcc
aabbcc
aabbcc

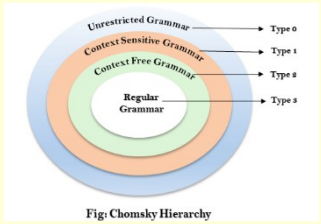


Fig: Chomsky Hierarchy

Turing Machine to UG

Given M and input u:

- Grammar first generates $u[q_0Bu]$
- Grammar simulates computation on $[q_0Bu]$
- First u is never changed!
- If M accepts u, [..] part is deleted
- Otherwise [..] remains

Grammar rules

- $q_ixy \rightarrow zq_jy$ for each transition $q_i,x) = (q_j,z,R)$
- $q_ix] \rightarrow zq_jB]$ for each transition $q_i,x) = (q_j,z,R)$
- $yq_ix \rightarrow q_jyz$ for each transition $q_i,x) = (q_j,z,L)$
- $[q_ix \rightarrow [q_jBz$ for each transition $q_i,x) = (q_j,z,L)$
- (other rules deal with the final case above)

Questions?

Questions?

Questions?

Context-sensitive Grammar

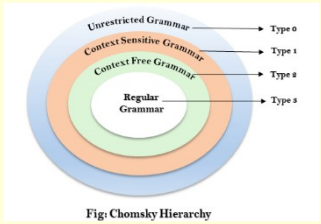


Fig: Chomsky Hierarchy

$S \rightarrow ABCS \mid T_c$

$CA \rightarrow AC$
$BA \rightarrow AB$
$CB \rightarrow BC$

$CT_c \rightarrow T_c c \mid T_b c$
$BT_b \rightarrow T_b b \mid T_a b$
$AT_a \rightarrow T_a a$

$T_a \rightarrow$

|Left| |Right|

$L = \{a^n b^n c^n \mid n \geq 1\}$

$S \rightarrow ZBCW \mid ZBC$
$W \rightarrow ABCW \mid ABC$

$CA \rightarrow AC$
$BA \rightarrow AB$
$CB \rightarrow BC$

$Z \rightarrow a$
$aA \rightarrow aa$
$aB \rightarrow ab$
$bB \rightarrow bb$
$bC \rightarrow bc$
$cC \rightarrow cc$

Context-sensitive Grammar

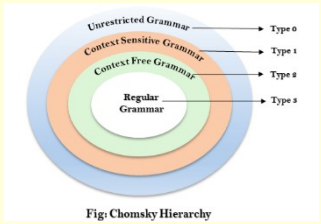


Fig: Chomsky Hierarchy

S ZBCW
ZBCABC
ZBACBC
ZABCBC
ZABBCC
aABBCC
aaBBcc
aabBCC
aabbCC
aabbcC
aabbcc

S ZBCW | ZBC
W ABCW | ABC

CA AC
BA AB
CB BC

Z a
aA aa
aB ab
bB bb
bC bc
cC cc

Context-sensitive Grammar

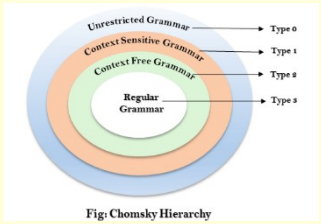


Fig: Chomsky Hierarchy

S aAbc | abc
A aAbC | abC
Cb bC
Cc cc

S abc

S aAbc
aabCbc
aabbCc
aabbcc

S aAbc
aaAbCbc
aaabCbCbc
aaabbCCbc
aaabbCbCc
aaabbbCCc
aaabbbCcc
aaabbbccc

$L = \{a^n b^n c^n \mid n \geq 1\}$

S
$a^n(bC)^{n-1}bc$ or
$a^n b(Cb)^{n-1}c$
…
$a^n b^n C^{n-1} c$
…
$a^n b^n c^n$

Context-sensitive Grammar

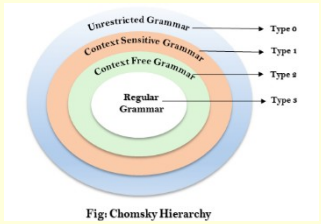


Fig: Chomsky Hierarchy

S → SBA | a
BA → AB
aA → aaB
B → b

S ⇒ a

S ⇒ SBA
⇒ aBA
⇒ aAB
⇒ aaBB
⇒ aabB
⇒ aabb

S ⇒ SBA
⇒ SBABA
⇒ aBABA
⇒ aABBA
⇒ aABAB
⇒ aAABB
⇒ aaBABB
⇒ aaABBB
⇒ aaaBBBB
⇒ aaabBBB
⇒ aaabbBB
⇒ aaabbbB
⇒ aaabbbb

S ⇒ SBA
⇒ SBABA
⇒ SBABABA
⇒ aBABABA
⇒ ...
⇒ aAAABBB
⇒ aaBAABBB
⇒ aaABABBB
⇒ aaAABBBB
⇒ aaaBABBBB
⇒ aaaABBBBB
⇒ aaaaBBBBBB
⇒ ...
⇒ aaaabbbbbb

S ⇒ ...
⇒ $a(BA)^n$
⇒ ...
⇒ $a(AB)^n$
⇒ ...
⇒ aA^nB^n
⇒ ...
⇒ $a^{n+1}B^nB^n$
⇒ b^{2n}

$L = \{a^{n+1}b^{2n} \mid n \geq 0\}$

CSGs, CSLs and Automata

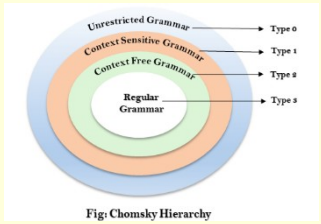


Fig: Chomsky Hierarchy

Languages known to be context-sensitive include

$L = \{a^i b^i c^i \mid i \geq 0\}$
$L = \{a^i b^j a^i b^j \mid i, j \geq 0\}$
$L = \{a^i b^j c^i d^j \mid i, j \geq 0\}$
$L = \{ xx \mid x \in \{a,b\}^* \}$
$L = \{ a^m \mid m \text{ is prime}\}$
$L = \{ a^m \mid m = n^2 \text{ for some } n \geq 0\}$
$L = \{ a^m \mid m = 2^n \text{ for some } n \geq 0\}$

Languages generated by CSGs are recognised by **Linear Bounded Automata**

CSG and LBA

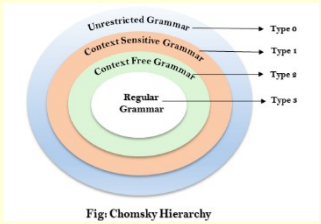
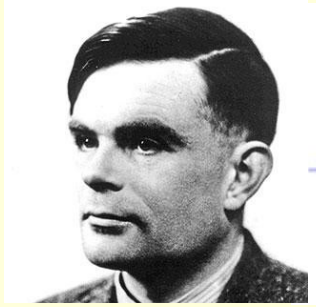


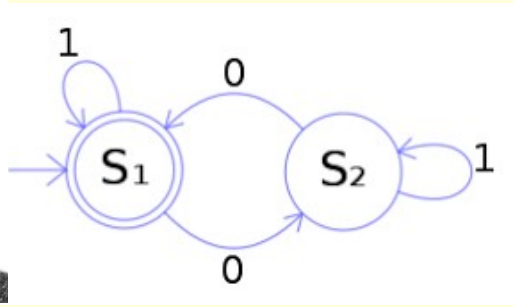
Fig: Chomsky Hierarchy

Linear Bounded Automaton (LBA): Turing Machine which can use only a **bounded** amount of tape

- Given input w, an LBA can only use $|w| + 2$ tape cells
  - $|w|$ to hold w
  - 2 for left and right markers
- CSG derivations never get shorter
- LBA strings never exceed $|w|$ in length
- Simulate CSG via LBA by applying rules "backwards" from w
- Simulate LBA by CSG the same way

Formal Definition

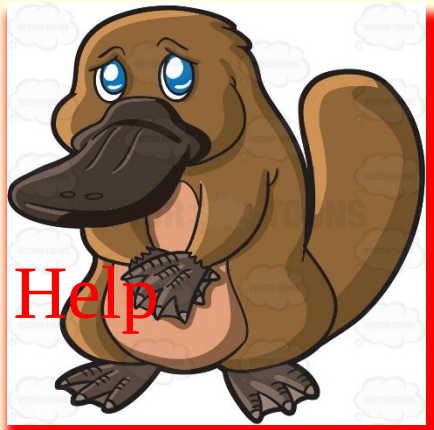




A Linear Bounded Automaton is M is a 8-tuple (Q, , <, >, q_0, F)

- Q is a finite set of **states**
- is a finite **alphabet**
- is the **blank symbol**
- Q x (subsets of) Q x x {L,R}
  is the **partial transition function**
- q_0 is the **start state** of the machine
- F Q is the set of **accepting or final states**
- < and > are **left** and **right** markers
  - Can be read, but not erased
  - Transitions **must** move right on < and left on >
  - All execution takes place in at most |w|+2 cells

The Platypus Game



https://www.youtube.com/watch?reload=9&v=0gM5TjSOQ48
https://zoneringtones.com

The Platypus Game

3 player tournament

1 vs 1 vs 1
1 vs 1 vs 2
1 vs 1 vs 3
1 vs 2 vs 2
1 vs 2 vs 3
…
1 vs n vs n
2 vs 2 vs 2
2 vs 2 vs 3
…
3 vs 3 vs 3
…
(n-1) vs (n-1) vs (n-1)
(n-1) vs (n-1) vs n
n vs n vs n

$$\sum_{i=1}^{n} i(i+1)/2 = \left(\sum_{i=1}^{n} i^2 + \sum_{i=1}^{n} i\right)/2$$
$$= n(n+1)(2n+1)/12 + n(n+1)/4$$
$$= n(n+1)(n+2)/6$$

When n = 268,
this is **3,244,140**

Around 100 times more than a 2-player tournament!

The Platypus Game

4 player tournament

1 vs 1 vs 1 vs 1
1 vs 1 vs 1 vs 2
…
1 vs 1 vs 1 vs n
1 vs 1 vs 2 vs 2
…
1 vs 2 vs 2 vs 2
…
1 vs n vs n vs n
2 vs 2 vs 2 vs 2
2 vs 2 vs 2 vs 3
…
2 vs n vs n vs n
…
3 vs 3 vs 3 vs n
…
(n-1) vs (n-1) vs (n-1) vs n
n vs n vs n vs n

$$\sum_{i=1}^{n} i(i+1)(i+2)/6$$

$$= (\sum_{i=1}^{n} i^3 + 3\sum_{i=1}^{n} i^2 + 2\sum_{i=1}^{n} i)/6$$

$$= n^2(n+1)^2/4 + n(n+1)(2n+1)/2 + n(n+1)$$

$$= n(n+1)(n+2)(n+3)/24$$

When n = 268,
this is 219,790,485

Around 10,000 times more than a 2-player tournament!

When n = 90, this is 2,919,735

Assignment 2

- Detailed specification is out **soon!**
- Platypus tournament for 2,500 machines
- 'Second version' of Universality task from Assignment 1
- Research on at least two interaction problems

The Platypus Game

Assignment 2

Variations

- Standard (as previously)
- Variable length (50, 200 vs 100 steps)
- Green – score 2 points for changing green to yellow
- Tree – score 5 points for reaching either tree
- Tiebreak – plays an extra match on a random tape for up to 200 steps

Report on your results with 2,500 machines (!!)

Either use new distribution or 2,500 selected from your old one (!!)

New OneDrive folder has been shared with you (find the file matching your student number)

Top 10 from each of you will go into the 'knockout' phase


Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

Break time! (We resume when all the pictures are gone! This will take 3 minutes!)