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.)

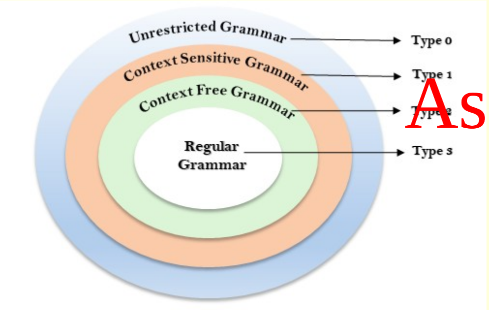
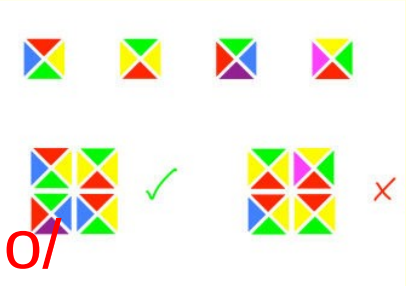
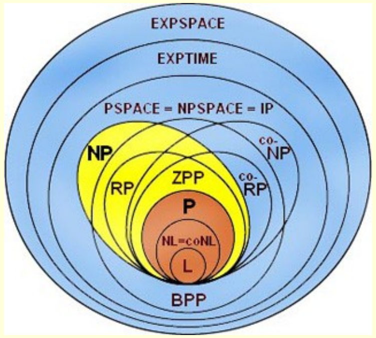
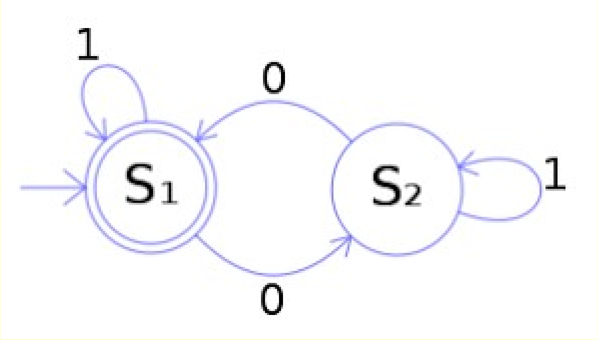
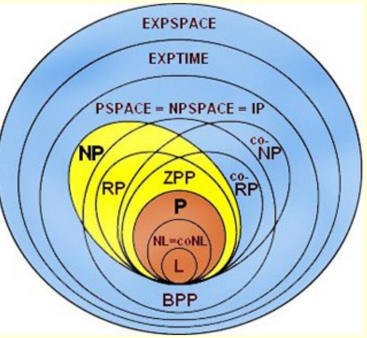


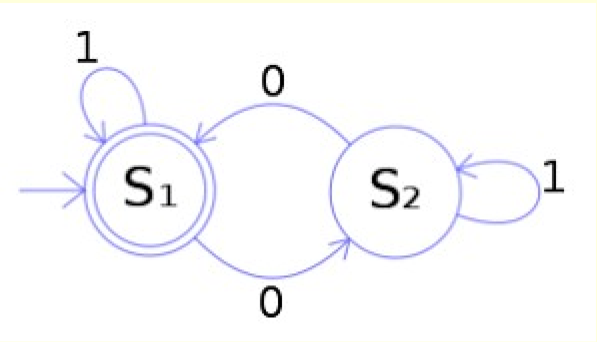
Fig: Chomsky Hierarchy







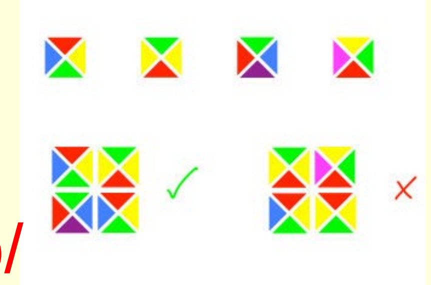




COSC1107

C ry

Week



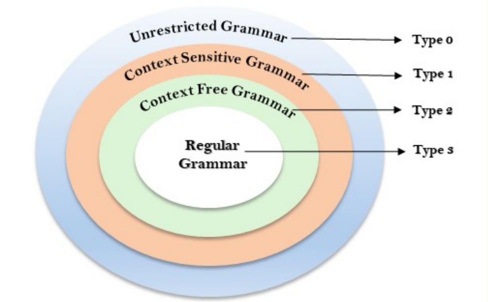


Fig: Chomsky Hierarchy

James Harland

james.harland@rmit.edu.au

*** With thanks to Sebastian Sardina**

Intro music 'Far Over' playing now …

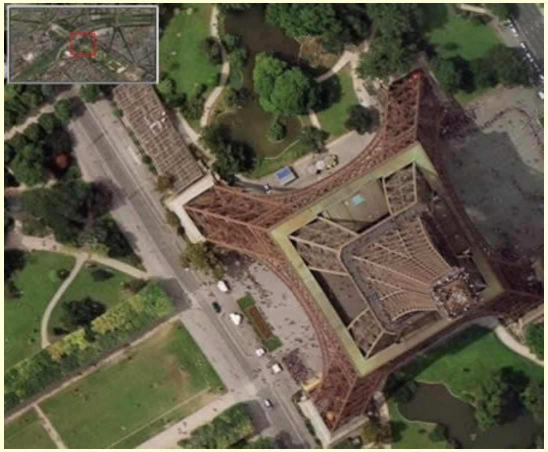
Acknowledgement

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 o University respectfully ac stors 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?

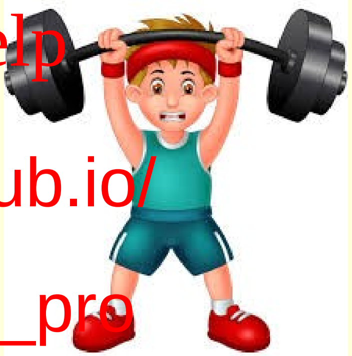
- Context-sensit

- Questions?

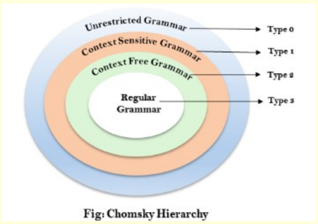
- Linear Bounded Automata
- Questions?
- Platypus Game ← Of course!
- Questions?

Overview

Week 5	Undecidability
Week 6	NFA -> DFA conversion, Pumping Lemma for regular languages
Week 7	Grammar normal forms, Pumping Lemma for context-fre
Week 8	Unrestricte e Grammars, Linear Bounded Aut



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 conversio[illegible]	[illegible]sm 'compiled out'
Pumping Lemma for Regular Languages	Bou[illegible]s in DFA Gen[illegible]n 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?

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Questions?

Computing Theory Topics

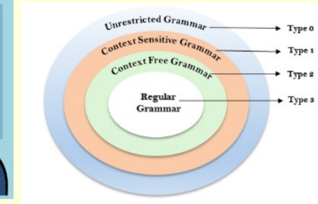
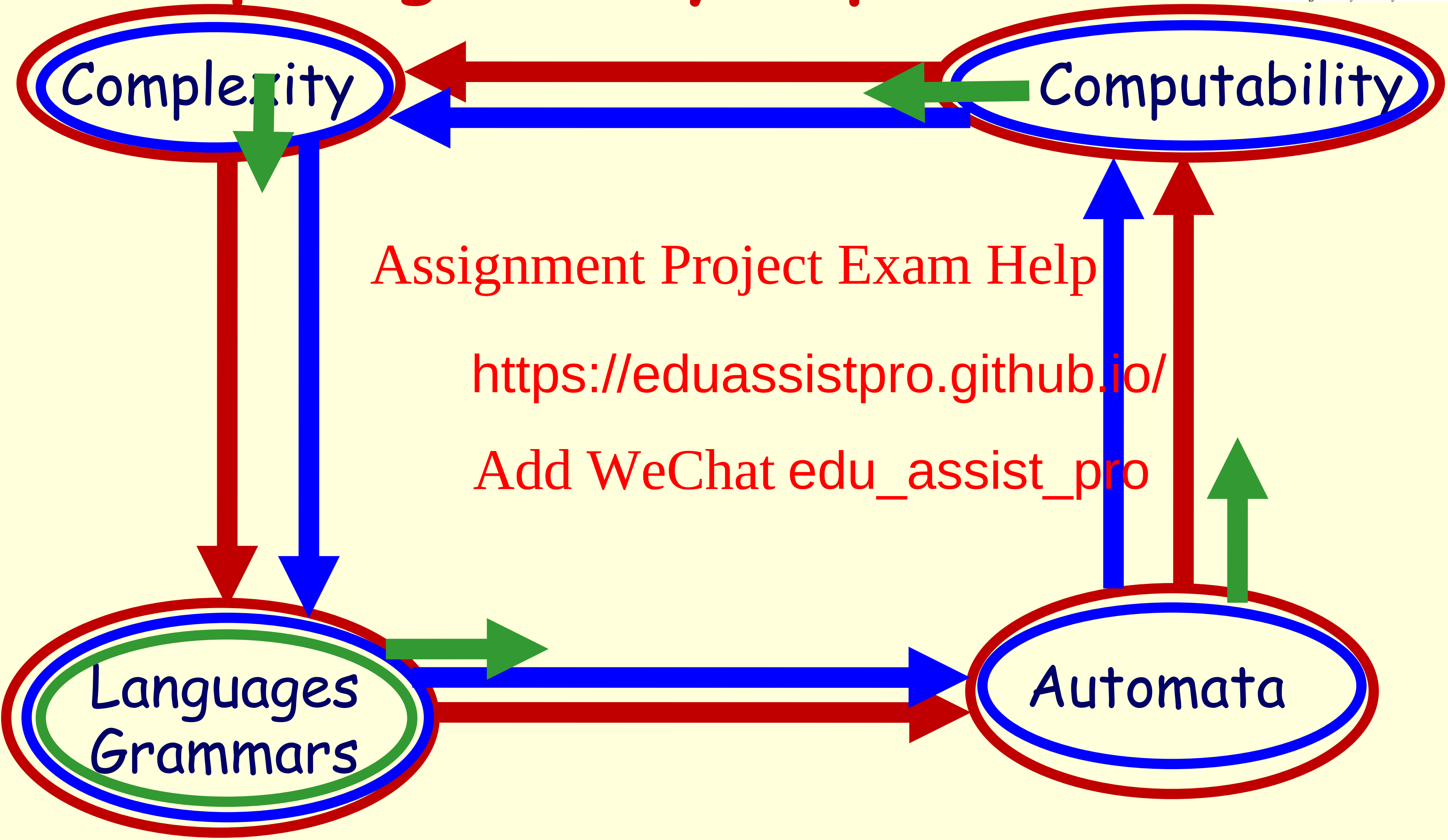


Fig: Chomsky Hierarchy



Chomsky Hierarchy

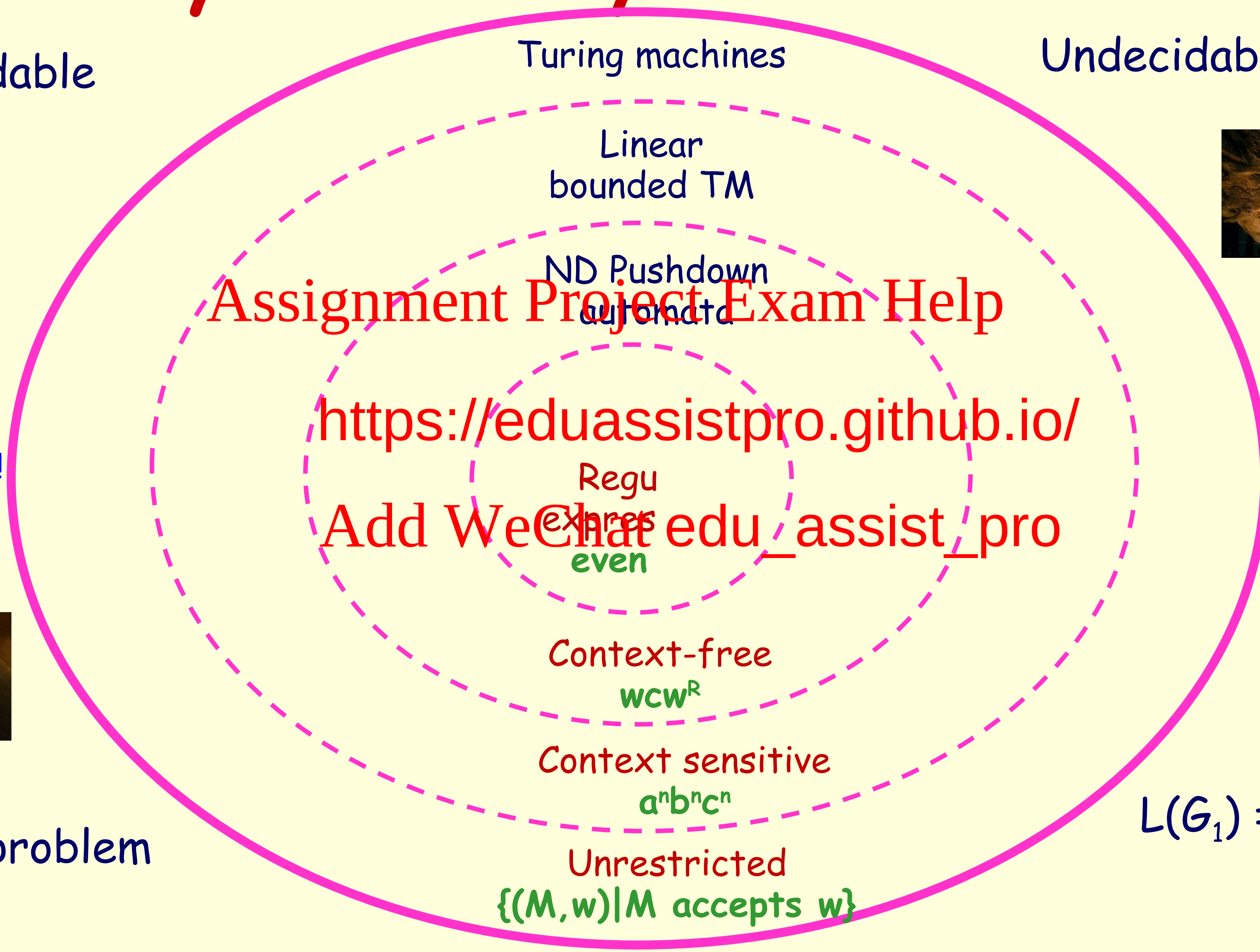
$L(G) = \Sigma^*$

Undecidable

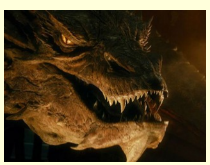
Undecidable

$L(G) =$

"I love a Turing-machine-free environment!"



"No pesky Turing machines out here!"



Halting problem

$L(G_1) = L(G_2)$

Chomsky Hierarchy

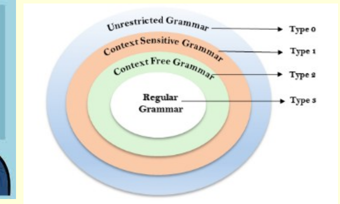
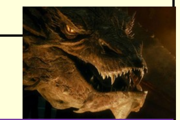


Fig: Chomsky Hierarchy



Automata	Languages Week 8	Grammars
	Undecidable languages	---
Turing Machines ⟷	Recursively enumerable languages ⟷	Unrestricted grammars
Linear Bounded Automata ⟷	⟷	Context-sensitive grammars
(Nondeterministic) Pushdown Automata ⟷	Context-free l Week ⟷	ntext-free grammars
Deterministic Pushdown Automata	?? (Deterministic CF?)	???
Nondeterministic Finite Automata & Deterministic Finite Automata ⟷	Regular languages ⟷ Week 6	Regular grammars & regular expressions

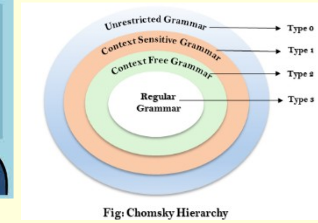


Fig: Chomsky Hierarchy

Grammars

A rule is of the form $\underbrace{(V \quad T)^{+}}_{\text{non-empty string over } V \quad T}$ $\underbrace{(V \quad T)^{*}}_{\text{string over } V \quad T}$

L R_1 | R_2 | … | R

ammars result from

on L and R_i

Name	Type	Constraint	Example
Unrestricted	0	--	AbC AC
Context-sensitive	1	|L| ≤ |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

Problem Reduction

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

Closure properties

Pumping Lemma

Questions?

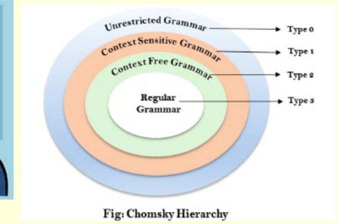
Questions?

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Questions?



Unrestricted Grammars

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

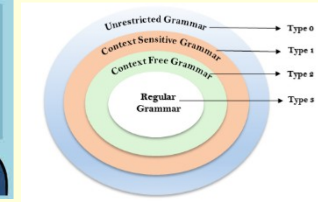


Fig: Chomsky Hierarchy

Example

$S \quad ABCS \mid T_c$ Generate same number of As,Bs,Cs

$CA \quad AC$ Sort them alphabetically …

$BA \quad AB$

$CB \quad BC$

$CT_c \quad T_c c \mid T_b c$ Move T_c t n C into c

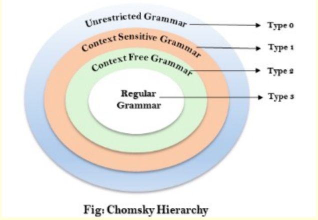
$BT_b \quad T_b b \mid T_a b$ Move T_b t n B into b

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

What is the language of this grammar?

T_a

Done!



Example

S $\underline{ABCS}$ $ABC\underline{ABCS}$ $ABCABC\underline{T_c}$

$AB\underline{AC}BCT_c$ $A\underline{ABC}BCT_c$ $AAB\underline{BC}CT_c$

$AABBC\underline{T_c c}$

$AABB\underline{T_b c}c$

$AAB\underline{T_b b}cc$

$AA\underline{T_a b}bcc$

$A\underline{T_a a}bbcc$

$\underline{T_a a}abbcc$

$\underline{a}abbcc$

S $ABCS \mid T_c$

CA AC
BA AB
CB BC

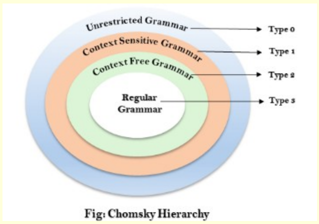
CT_c $T_c c \mid T_b c$
BT_b $T_b b \mid T_a b$
AT_a $T_a a$

T_a

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

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

Example



$S \;\; ABCS \mid T_c$

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

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

T_a

S ABCS ABCABCS ABACBCS

AABCBCS AABCBCT_c

AABCBT_cc AABBCT_cc

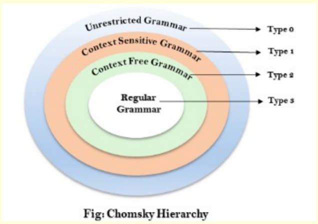
AABB$T_b$cc

AAB$T_a$bcc

AABbcc

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

Example



$S \quad ABCS \mid T_c$

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

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

T_a

$S \quad \underline{ABCS} \quad ABC\underline{ABCS} \quad AB\underline{AC}BCS$

$A\underline{AB}CBCS \quad AABCBC\underline{T_c}$

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

$AABB\underline{T_b c}c$

$AAB\underline{T_b b}cc$

$AA\underline{T_a b}bcc$

$A\underline{T_a a}bbcc$

$\underline{T_a a}abbcc$
$aabbcc$

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

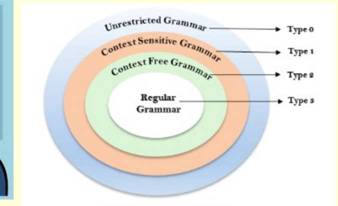


Fig: Chomsky Hierarchy

Example

$S \ \underline{ABCS} \ ABC\underline{ABCS} \ ABCABC\underline{T_c}$

$ABCAB\underline{T_c c} \ ABA\underline{CB}T_c c$

$ABAB\underline{CT_c}c \ ABAB\underline{T_c c}c$

$A\underline{AB}BT_c cc$

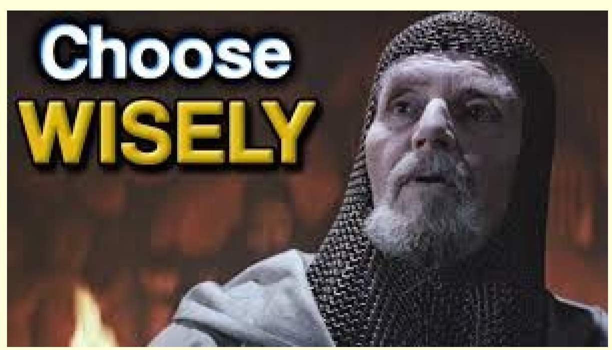
$S \ ABCS \mid T_c$

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

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

T_a

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



Questions?

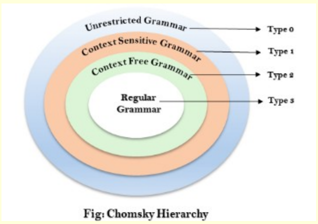
Questions?

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro

Questions?



Unrestricted Grammar

???

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, ...

Strategy:

1. Generate three copies of ' rs' 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

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

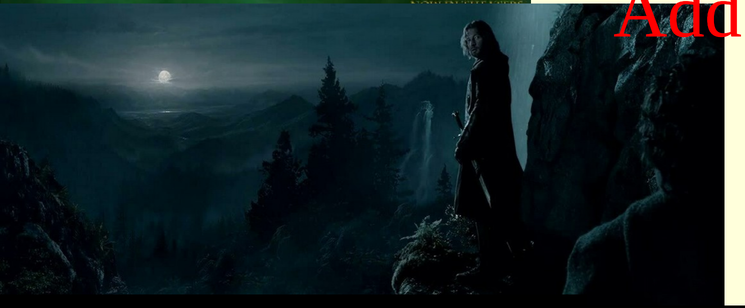
- Consult with other students if you wish
- Time limit will be 10 minutes

Are you ready? Are you sure?

Go!

The pictures will take 10 minutes to disappear!

Thomas music means 1 minute left!



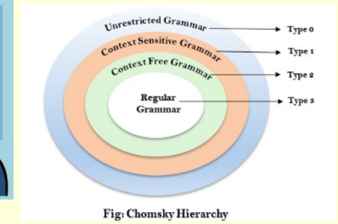


Fig: Chomsky Hierarchy

Unrestricted Grammar

???

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 ...)

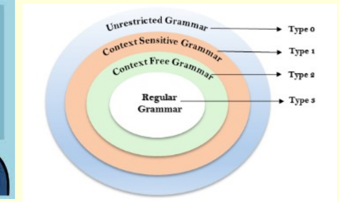


Fig: Chomsky Hierarchy

Unrestricted Grammar

$S \quad A_1A_2A_3S \mid B_1B_2B_3S \mid T_3$

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

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

$A_1T_1 \quad T_1a$
$A_2T_2 \quad T_2a \mid T_1a$
$A_3T_3 \quad T_3a \mid T_2a$
$B_1T_1 \quad T_1b$
$B_2T_2 \quad T_2b \mid T_1b$
$B_3T_3 \quad T_3b \mid T_2b$
T_1

$S \quad A_1A_2A_3S$
$A_1A_2A_3A_1A_2A_3S$
$A_1A_2A_3A_1A_2A_3B_1B_2B_3S$
A
$A_1 \quad {}_2 \quad {}_1 \quad {}_2 \quad {}_3 \quad {}_3$
$A_1A_1A_2A_2A_3A_3B$
...
$A_1A_1B_1A_2A_2B_2A_3A_3B_3S$
$A_1A_1B_1A_2A_2B_2A_3A_3B_3T_3$
$A_1A_1B_1A_2A_2B_2A_3A_3T_3b$
...
Taabaabaab
aabaabaab

ababab,
baabaabaa,
abbaabbaabba,
...

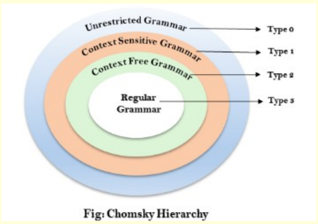


Fig: Chomsky Hierarchy

Unrestricted Grammar

$S \quad A_1A_2A_3S \mid B_1B_2B_3S \mid T$

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

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

$A_1 \quad a$
$A_2 \quad a$
$A_3 \quad a$
$B_1 \quad b$
$B_2 \quad b$
$B_3 \quad b$
T

$S \quad A_1A_2A_3S$
$A_1A_2A_3A_1A_2A_3S$
$A_1A_2A_3A_1A_2A_3B_1B_2B_3S$
A
$A_1 \quad 2 \quad 1 \quad 2 \quad 3 \quad 3$
$aA_1A_2A_2A_3A_3B_1$
...
aaaaaabbb (!!)

ababab,
baabaabaa,
abbaabbaabba,
...

Too simple!

"Don't generate too much!"

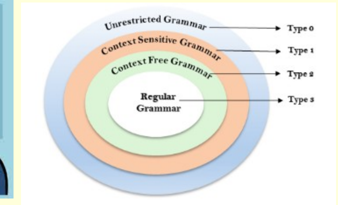


Fig: Chomsky Hierarchy

Chomsky Hierarchy

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

UG to Turing Machine

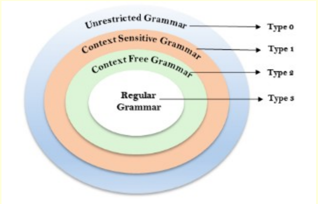


Fig: Chomsky Hierarchy

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

Turing machine wo

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
$ABCABCT_c$
$ABACBCT_c$
$AABCBCT_c$
$AABBCCT_c$
$AABBCT_cc$
$AABBT_bcc$
$AABT_bbcc$
AAT_abbcc
AT_aabbcc
$T_aaabbcc$
aabbcc

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

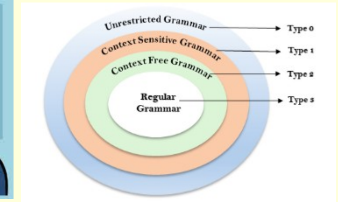


Fig: Chomsky Hierarchy

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,
- Otherwise [..] r

Grammar rules

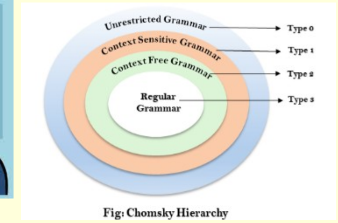
- q_ixy zq_jy for each transition q_i,x) = (q_j,z,R)
- $q_ix]$ $zq_jB]$ for each transition q_i,x) = (q_j,z,R)
- yq_ix q_jyz for each transition q_i,x) = (q_j,z,L)
- $[q_ix$ $[q_jBz$ for each transition q_i,x) = (q_j,z,L)
- (other rules deal with the final case above)

Questions?

Questions?

Questions?

Context-sensitive Grammar



$S \;\; ABCS \mid T_c$

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

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

T_a

|Left| |Right|

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

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

AC
AB

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

Context-sensitive Grammar

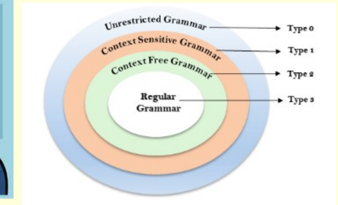


Fig: Chomsky Hierarchy

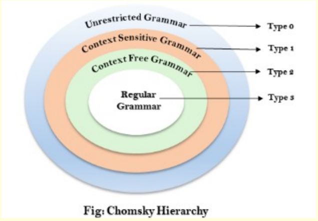
S ZBCW | ZBC
W ABCW | ABC

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

CA AC
A AB

B BC

a
aA aa
aB ab
bB bb
bC bc
cC cc



Context-sensitive Grammar

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

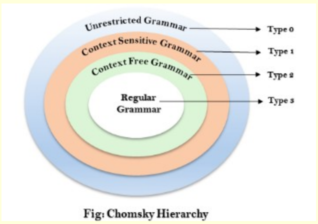
$L = \{a^nb^nc^n \mid n \geq 1\}$

S abc

S aAbc
aabCbc
aabbCc

S ...
$a^n(bC)^{n-1}$
$a^nb(Cb)^{n-1}c$
...
$a^nb^nC^{n-1}c$
...
$a^nb^nc^n$

S aAbc
aaAbCbc
aaabCbCbc
aaabbCCbc
aaabbCbCc
aaabbbCCc
aaabbbCcc
aaabbbccc



Context-sensitive Grammar

S SBA | a
BA AB
aA aaB
B b

S a

S SBA
aBA
aAB
aaBB
aabB
aabb

S SBA
SBABA
aBABA
aABBA
a __
a ___
a __
aaABBB
aaaBBBB
aaabBBB
aaabbBB
aaabbbB
aaabbbb

S SBA
SBABA
SBABABA
aBABABA
___
a ___
a ___
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

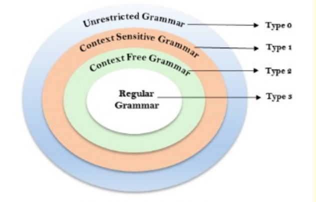


Fig: Chomsky Hierarchy

Languages known to be context-sensitive include

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

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

CSG and LBA

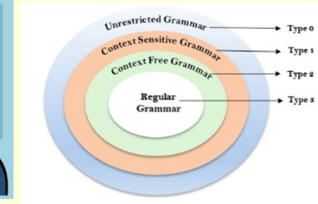
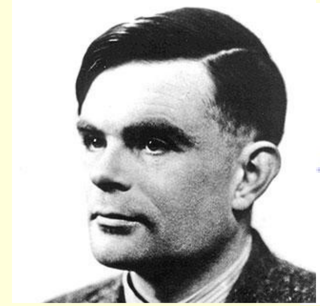


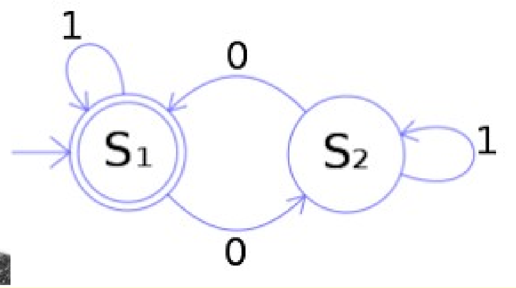
Fig: Chomsky Hierarchy

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 r
- CSG derivations
- LBA strings never exceed |w|
- Simulate CSG via LBA by appl backwards" from w
- Simulate LBA by CSG the same way

Formal Definition

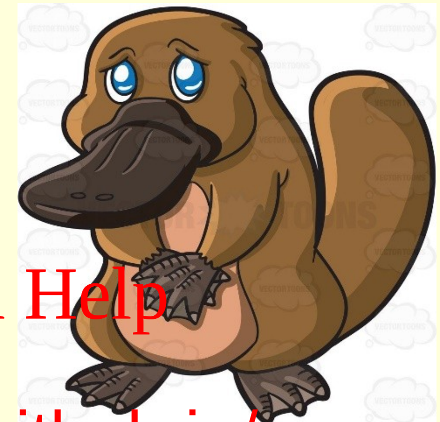




A Linear Bounded Automaton is M is a 8-tuple (Q, , <, >, q_0, F)

- Q is a finite set of **states**
- is a finite **alphabet**
- is the **blank sy**
- Q x (subsets o
  is the **partial transition fun**
- q_0 is the **start state** of the m
- 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 = 26
this is 3,24

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

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

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

n+1)/2 + n(n+1)

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

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

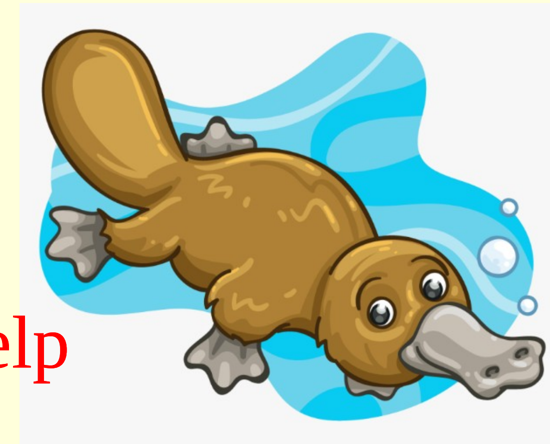
When n = 90, this is **2,919,735**

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 intractable problems

The Platypus Game

Assignment 2



Variations

- Standard (as previously)
- Variable length (50, 200 vs 100 steps)
- Green – score 2 po ow
- Tree – score 5 poi
- Tiebreak – plays an e for up to 200 steps

Report on your results with 2,500 ma

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

That's it!

I am out of here!

The lecture ed.

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