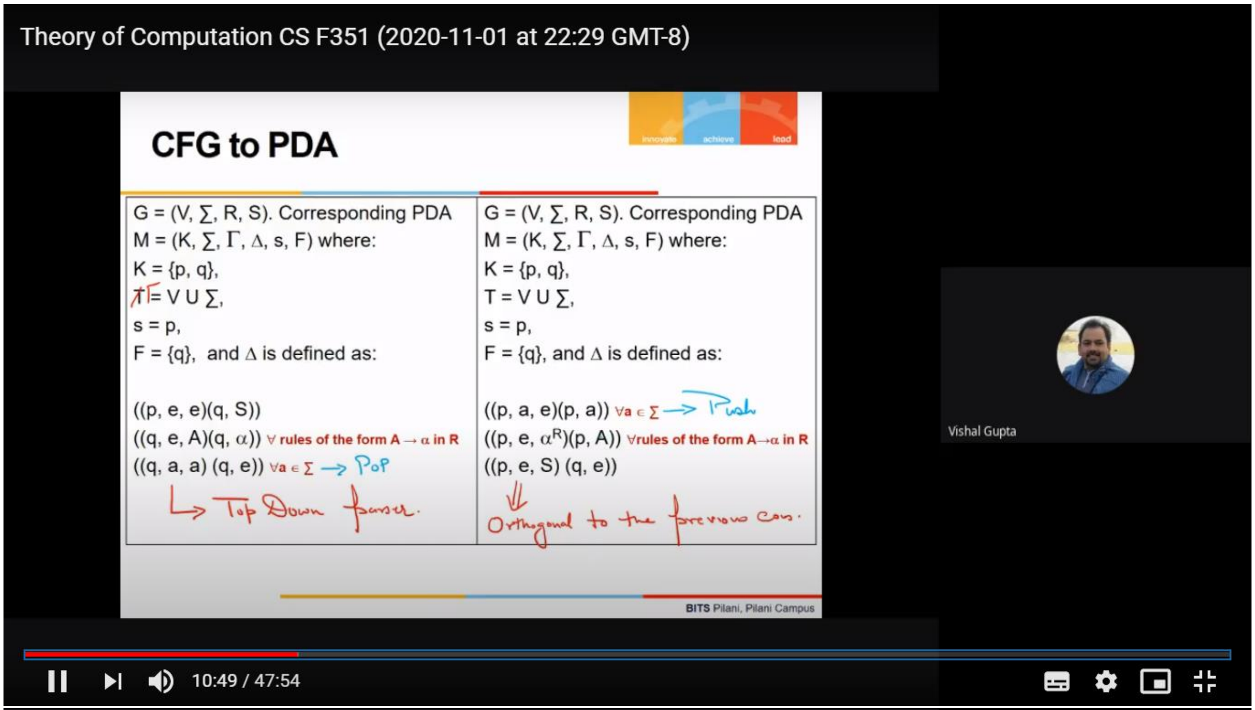
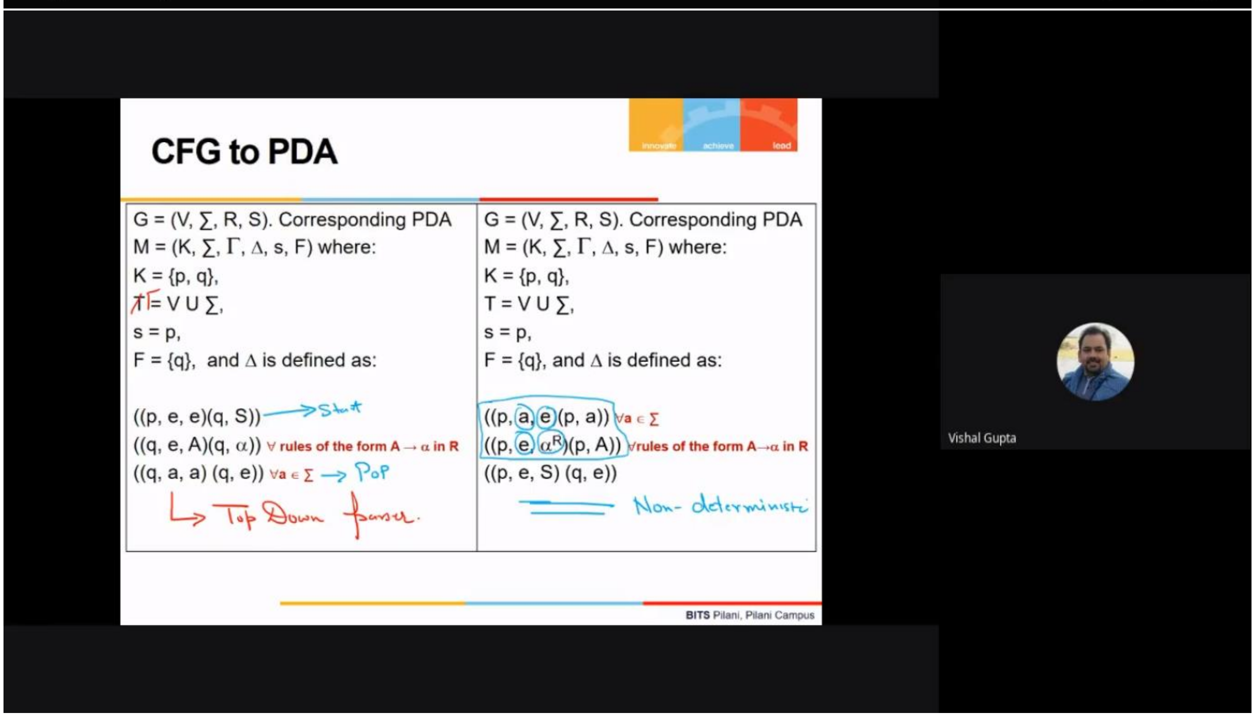
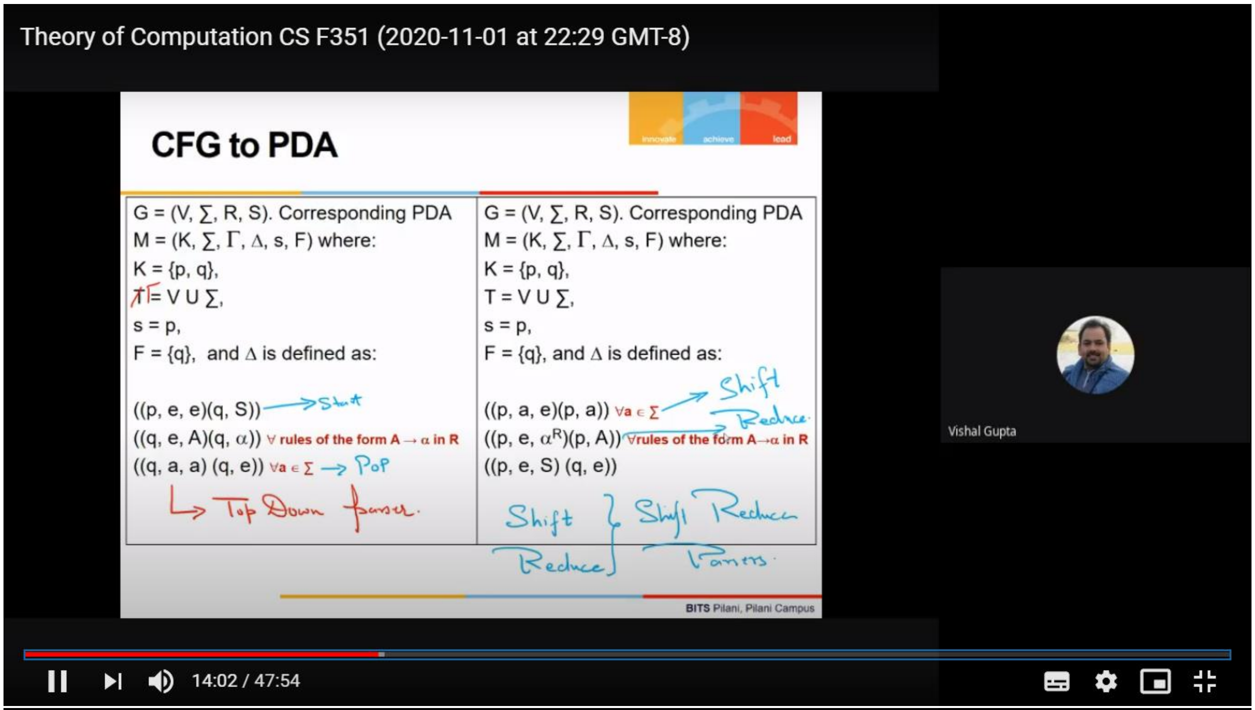
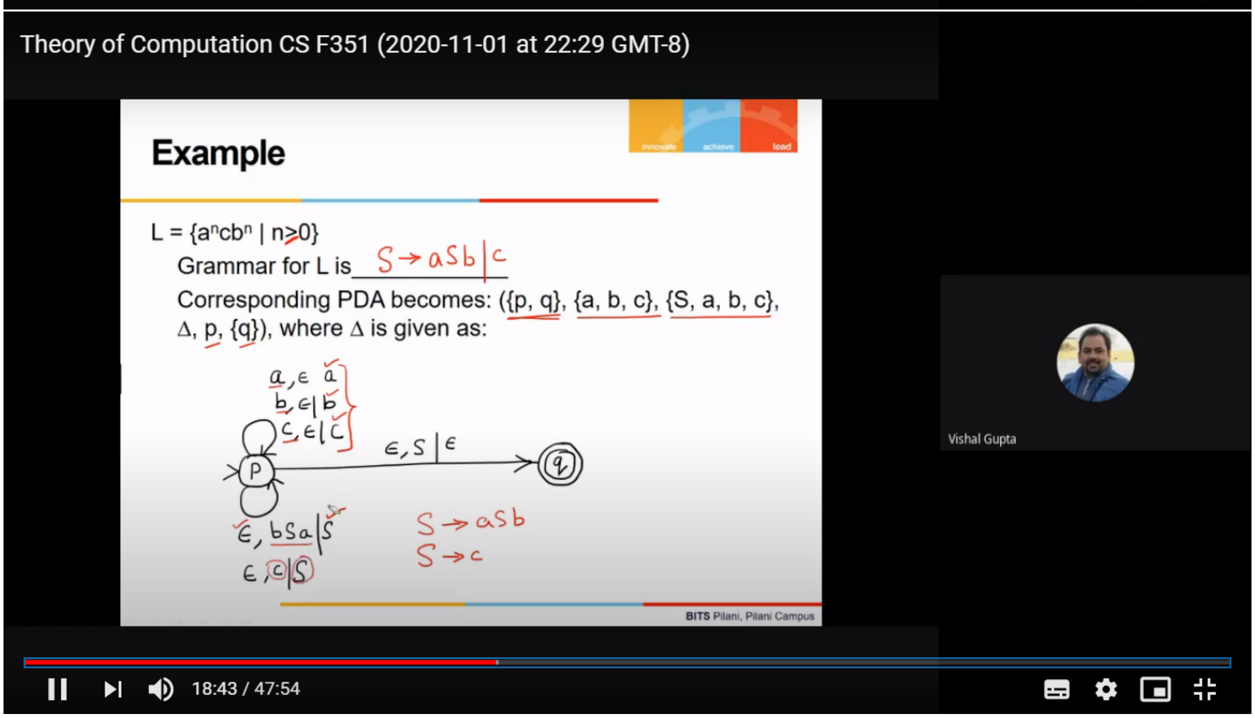
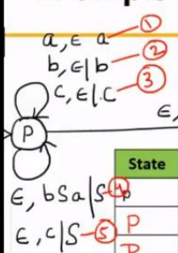

Theory of Computation CS F351 (2020-11-01 at 22:29 GMT-8)
CFG to PDA
innovate achieve lead
G = (V, ∑, R, S). Corresponding PDA
M = (K, ∑, Γ, Δ, s, F) where:
K = {p, q},
Γ = V U ∑,
s = p,
F = {q}, and Δ is defined as:
((p, e, e)(q, S))
((q, e, A)(q, α)) ∀ rules of the form A → α in R
((q, a, a) (q, e)) ∀a ∈ ∑ → PoP
Top Down parser.
G = (V, ∑, R, S). Corresponding PDA
M = (K, ∑, Γ, Δ, s, F) where:
K = {p, q},
T = V U ∑,
s = p,
F = {q}, and Δ is defined as:
((p, a, e)(p, a)) ∀a ∈ ∑ → Push
((p, e, α^R)(p, A)) ∀rules of the form A→α in R
((p, e, S) (q, e))
Orthogonal to the previous case.
BITS Pilani, Pilani Campus
Vishal Gupta
10:49 / 47:54

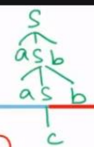

CFG to PDA
innovate achieve lead
G = (V, ∑, R, S). Corresponding PDA
M = (K, ∑, Γ, Δ, s, F) where:
K = {p, q},
Γ = V U ∑,
s = p,
F = {q}, and Δ is defined as:
((p, e, e)(q, S)) → Start
((q, e, A)(q, α)) ∀ rules of the form A → α in R
((q, a, a) (q, e)) ∀a ∈ ∑ → PoP
Top Down parser.
G = (V, ∑, R, S). Corresponding PDA
M = (K, ∑, Γ, Δ, s, F) where:
K = {p, q},
T = V U ∑,
s = p,
F = {q}, and Δ is defined as:
((p, a, e)(p, a)) ∀a ∈ ∑
((p, e, α^R)(p, A)) ∀rules of the form A→α in R
((p, e, S) (q, e))
Non-deterministic
BITS Pilani, Pilani Campus
Vishal Gupta


Theory of Computation CS F351 (2020-11-01 at 22:29 GMT-8)
CFG to PDA
innovate achieve lead
G = (V, ∑, R, S). Corresponding PDA
M = (K, ∑, Γ, Δ, s, F) where:
K = {p, q},
Γ = V U ∑,
s = p,
F = {q}, and Δ is defined as:
((p, e, e)(q, S)) → Start
((q, e, A)(q, α)) ∀ rules of the form A → α in R
((q, a, a) (q, e)) ∀a ∈ ∑ → Pop
Top Down Parser.
G = (V, ∑, R, S). Corresponding PDA
M = (K, ∑, Γ, Δ, s, F) where:
K = {p, q},
T = V U ∑,
s = p,
F = {q}, and Δ is defined as:
((p, a, e)(p, a)) ∀a ∈ ∑ → Shift
((p, e, α^R)(p, A)) ∀rules of the form A→α in R → Reduce
((p, e, S) (q, e))
Shift Reduce } Shift Reduce Parsers.
BITS Pilani, Pilani Campus
Vishal Gupta
14:02 / 47:54


Theory of Computation CS F351 (2020-11-01 at 22:29 GMT-8)
Example
innovate achieve lead
L = {a^n cb^n | n≥0}
Grammar for L is S → aSb | c
Corresponding PDA becomes: ({p, q}, {a, b, c}, {S, a, b, c}, Δ, p, {q}), where Δ is given as:
a, ε | a
b, ε | b
c, ε | c
ε, S | ε
p
q
ε, bSa | S
ε, c | S
S → aSb
S → c
BITS Pilani, Pilani Campus
Vishal Gupta
18:43 / 47:54

Example





Parse the string aacbb using the given parser.

Shift/Reduce

State	Input String	Stack	Action	Transition used	Rule of G
p	aacbb	--	--		--
P	acbb	a	Shift	1	—
P	cbb	aa	Shift	1	—
P	bb	caa	Shift	3	—
P	bb	Saa	Reduce	5	S → c
P	b	$\boxed{bSa}$a	Shift	2	—
P	b	Sa	Reduce	4	S → aSb
P	—	bSa	Shift	2	—
P	—	S	Reduce	4	S → aSb
q	—	—	—	6	

BITS Pilani, Pilani Campus

Theory of Computation CS F351 (2020-11-01 at 22:29 GMT-8)

Example

Let $L = \{1^k 0^i 1^i 0^j 1^j 0^k \mid i, j, k > 0\}$.

Following is the CFG for L:

S -> 1S0 | 1AA0

A -> 0A1 | 01

Convert the above CFG to PDA which is suitable for bottom up parser. Also, make a parsing table for the string w = 1100110100.

Sol: *Corresponding PDA is: ({p, q}, {0, 1}, {S, A, 0, 1}, Δ, p, {q}) such that Δ is given by the following:*

Transition	Transition Number
(p, 0, e)(p, 0)	1
(p, 1, e)(p, 1)	2
(p, e, 0S1)(p, S)	3
(p, e, 0AA1)(p, S)	4
(p, e, 1A0)(p, A)	5
(p, e, 10)(p, A)	6
(p, e, S)(q, e)	7

BITS Pilani, Pilani Campus

34:38 / 47:54

Theory of Computation CS F351 (2020-11-01 at 22:29 GMT-8)

Example

Let L = $\{1^k 0^i 1^i 0^j 1^j 0^k \mid i, j, k > 0\}$. Bottom up parsing table for **1100110100** is as follows:

Transition	Transition Number
(p, 0, e) (p, 0)	1
(p, 1, e) (p, 1)	2
(p, e, 0S1) (p, S)	3
(p, e, 0AA1) (p, S)	4
(p, e, 1A0) (p, A)	5
(p, e, 10) (p, A)	6
(p, e, S) (q, e)	7

State	Unread input	Stack	Shift / Reduce	Transition No	Rule of G
p	1100110100	--	--	--	--
P	100110100	1	Shift	2	—
P	00110100	11	Shift	2	
P	0110100	011	Shift	1	
P	110100	0011	Shift	1	
P	10100	10011	Shift	2	
P	10100	A011	Reduce	6	A → 01
P	0100	1A011	Shift	2	
P	0100	A11	Reduce	5	A → 0A1
P	100	0A11	Shift	—	
P	00	10A11	Shift		
P	00	AA11	Reduce	–	A → 01
P	0	0AA11	Shift	–	—
P	0	S1	Reduce	4	S → 1AA0
P	–	0S1	Shift	1	—
P	–	S	Reduce		S → 1S0
q	—	—	—	7	—

Vishal Gupta

46:28 / 47:54