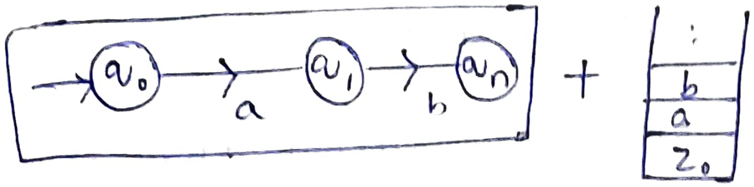
Unit-5

PUSH DOWN Automata

PDA: F.A + Stack



Definition :-

The mathematical representation of CFL is known as push Down Automata (PDA).

→ PDA is a 7 tuple variable

$M = \{Q, \Sigma, T, z_0, \delta, q_0, F\}$

where

Q = Finite set of all states

Σ : Input Alphabet (Symbols)

T = Set of all tape symbols,

z_0 = Top most symbol of the stack

q_0 = Initial state

F = The set of all final states

δ : Transition function

$\boxed{\delta : Q \times \Sigma \times T \longrightarrow Q \times T^*}$

Note :-

① The purpose of the symbol z_0 is to check wheth the stack is empty or not.

② PDA uses the stack as external storage

③ PDA can recognize the data through the stack symbols.

④ The Language accepted by PDA is Context Free Language (CFL)

⑤ Every Regular Language is also accepted by PDA

⑥ Some of NRL's are accepted by PDA

⑦ Because of external memory PDA can accept more language than FA but still there are some languages which are not accepted by PDA also

⑧ PDA is not so rich (accepting power or Computing powe in computing mathematical functions. when compare with FA.

⑨ PDA also can be constructed in 2 ways

a) DPDA (Deterministic PDA)

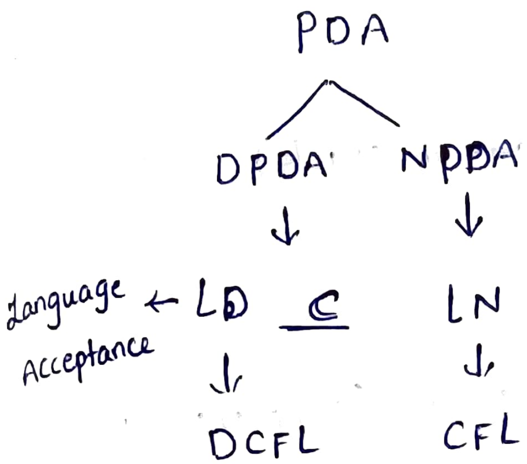
b) NPDA (Non-Deterministic PDA)

* → By defalult the PDA is NPDA

→ The accepting power of NPDA and DPDA are not same

→ NPDA is more powerful than DPDA

→ DPDA is more efficient than NPDA



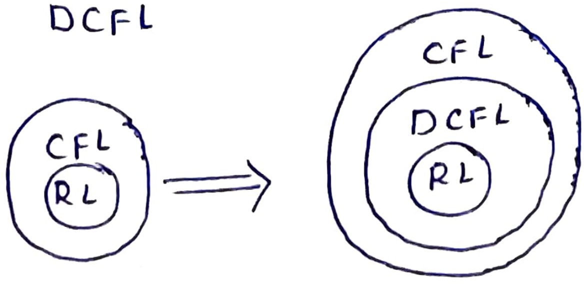
DCFL $\subseteq$ CFL
LD $\subseteq$ LN

CFL → The language which is accepted by NDPA is CFL

DCFL → the language accepted by DPDA is called DCFL

→ Every Language accepted by DPDA is also accepted by NPDA, but the language accepted by NPDA need not be accepted by DPDA.

→ Every DCFL is CFL but every CFl need not be a DCFL

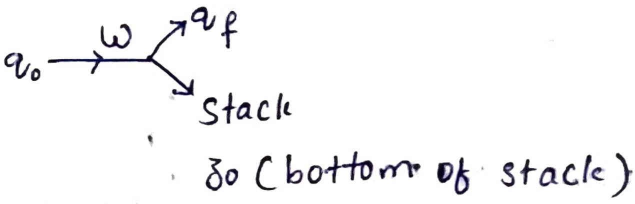


Acceptance by PDA

PDA can accept the strings in 2 ways

① Acceptance by empty stack

② Acceptance by Final state



Acceptance by Empty stack

After reading the entire (complete) input string if the ~~PDA~~ ~~reaches~~ ~~the~~ ~~final~~ ~~state~~ ~~then~~ stack is empty then that input string is accepted by PDA

Acceptance by Final state

After reading the complete input string if PDA reaches the final state then the input string is accepted by PDA.

Stack operations

① push

② pop

③ skip

→FA:- Transition function $\delta(Q, a) = P$

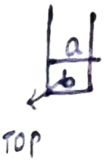
Next state (P) is always depends upon i/p symbol (a) and current state (Q)

→PDA:- The movement of PDA depends on 3 entities

① Current state

② processing symbol

③ Top most symbol of stack



$\delta(q_i, a, b) = (q_j, ab)$ $\boxed{(q_j, a)}$ → can write like this

q_i ↓ Current state

a ↓ processing state (Current i/p symbol)

b ↓ Top of the stack

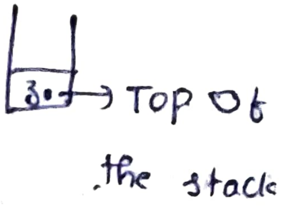
q_j ↘ New State/ Next state

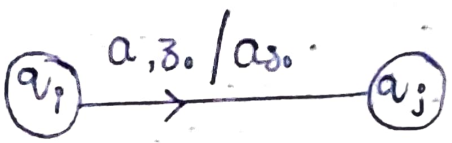
ab → New Top most symbol of stack

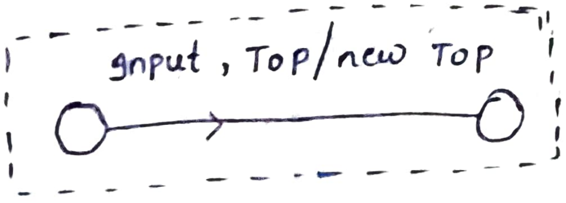
Stack operations

① push:-

$\delta(q_i, a, z_0) = (q_j, az_0)$

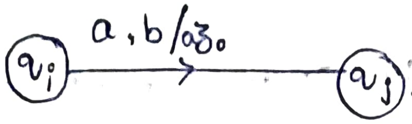




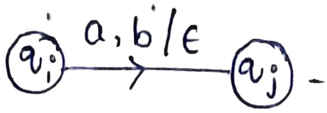


② pop:

a) $\delta(q_i, a, b) = \delta(q_j, az_0)$



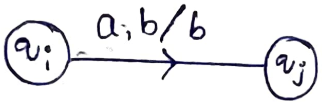
b) $\delta(q_i, a, b) = \delta(q_j, \epsilon)$



→ Both are valid

③ Skip

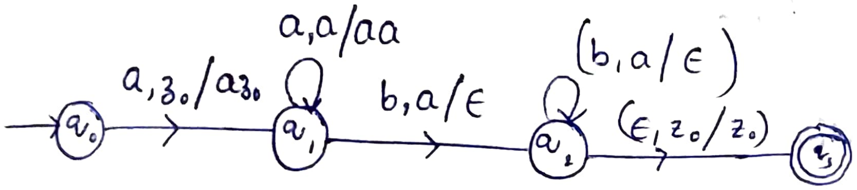
$\delta(q_i, a, b) = (q_j, b)$

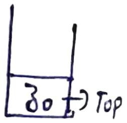


Ex:

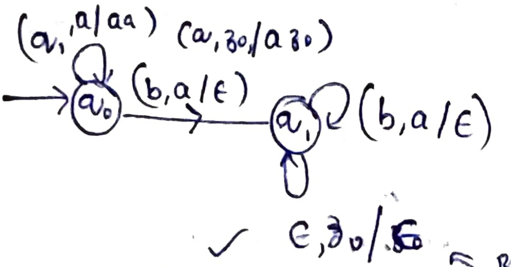
① Construct PDA for the lang $L = \{a^n b^n / n \geq 1\}$

$L = \{ab, aabb, aaabb, \ldots\}$



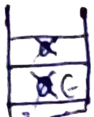


Acceptance by empty stack

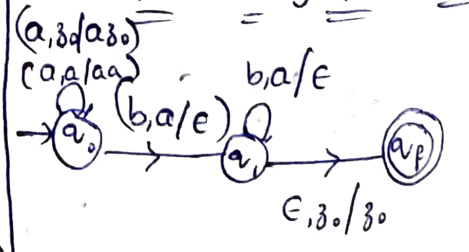


W = aabbε

W = aabε ← By this transition Halt condition

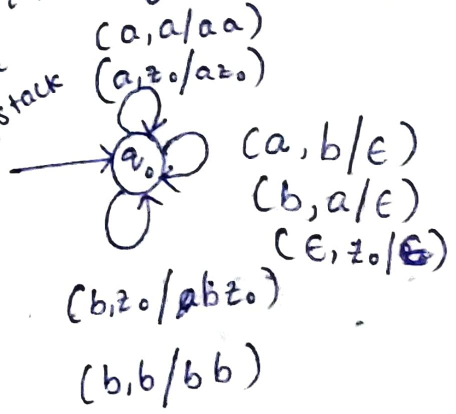


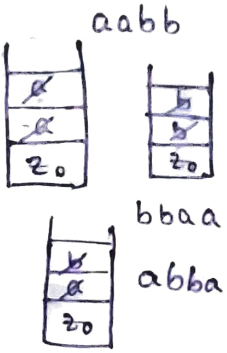
Acceptance by final state



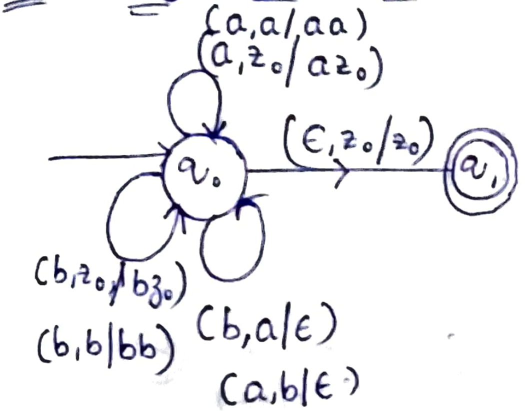
② L = { No. of a's and No. of b's are equal }

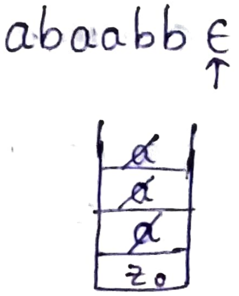
Acceptance by Empty stack





Acceptance by Final state

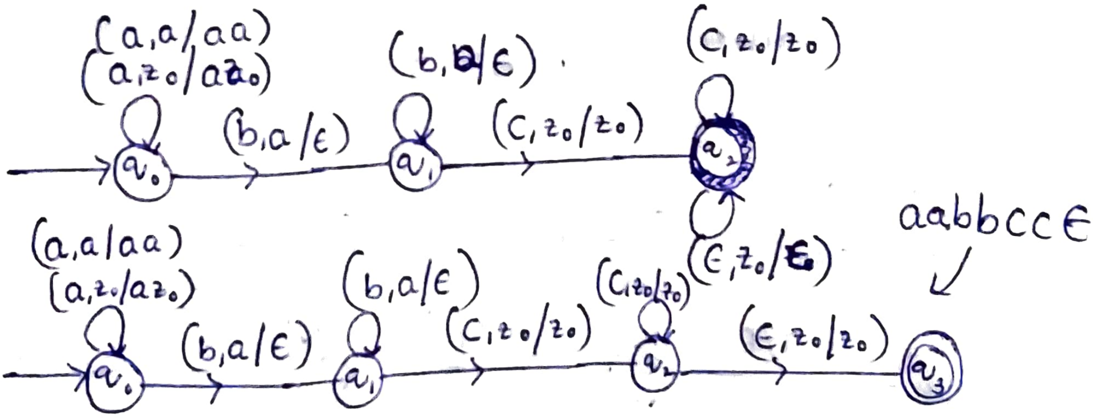




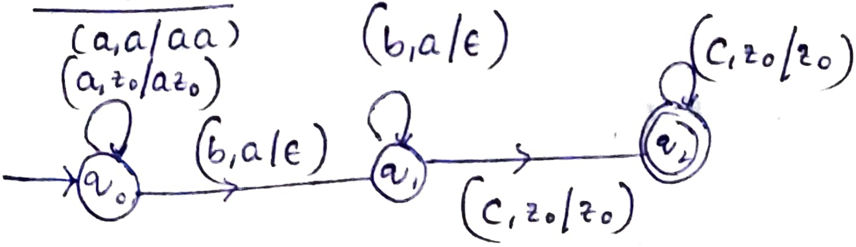
③ $L = \{ a^n b^n c^m / m,n, \geq 1 \}$

Stack

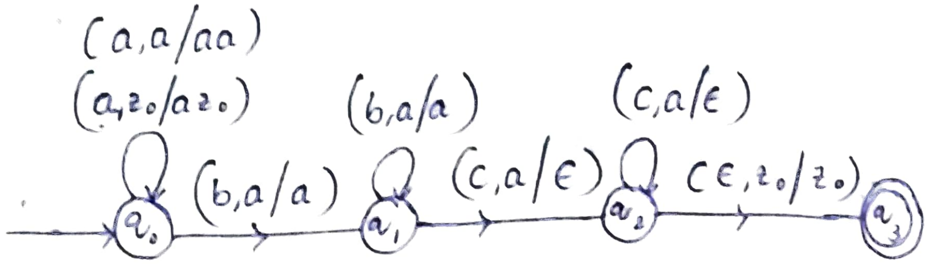
Final state



aabbccc

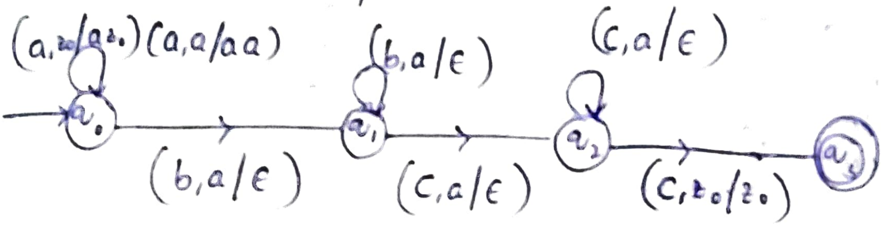


③ $L = \{a^n b^m c^n / m,n \geq 1\}$

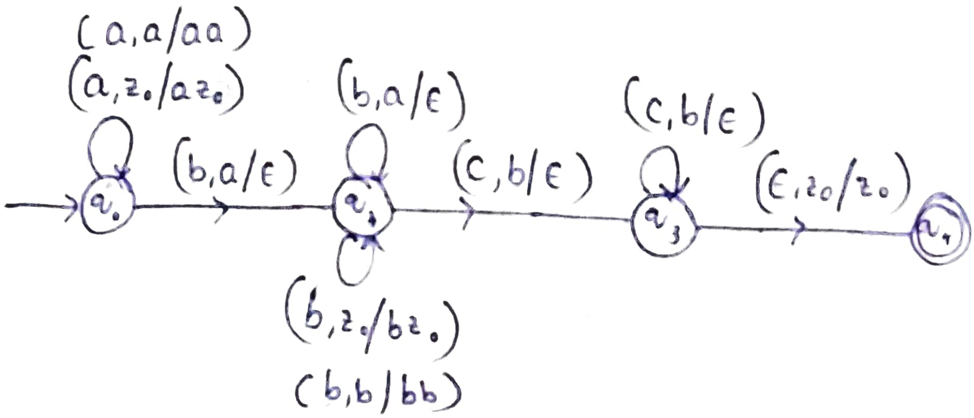


④ $L = \{a^{m+n} b^m c^n / m,n \geq 1\}$

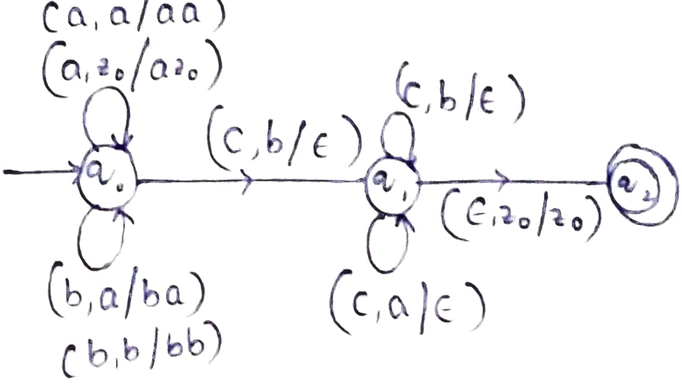
$L = \{a^m . a^n b^m c^n / m,n \geq 1\}$



⑤ $L = \{a^n b^{m+n} c^m / m,n \geq 1\}$



⑥ $L = \{a^n b^m c^{m+n} / m,n \geq 1\}$

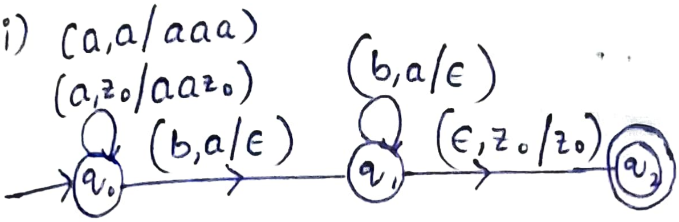


aa

$L = \{ a^n b^{2n} / n \geq 1 \}$

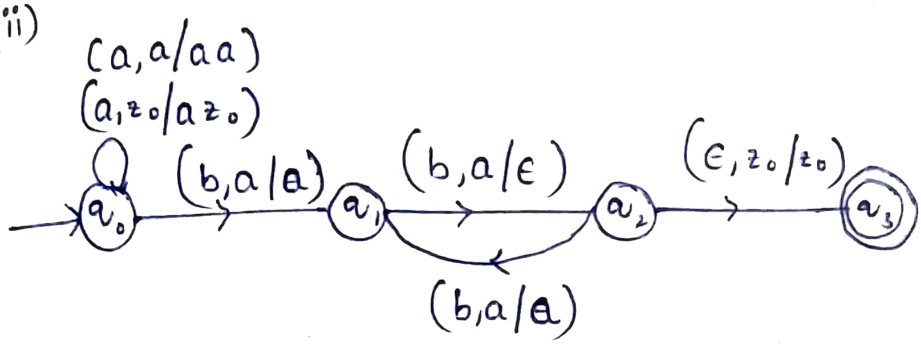
→ We can push 2'as for a single a

case i)



a bb
↓ ↓ ↓
aa bb

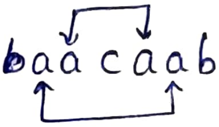
case ii)

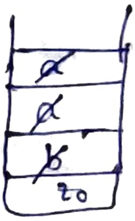


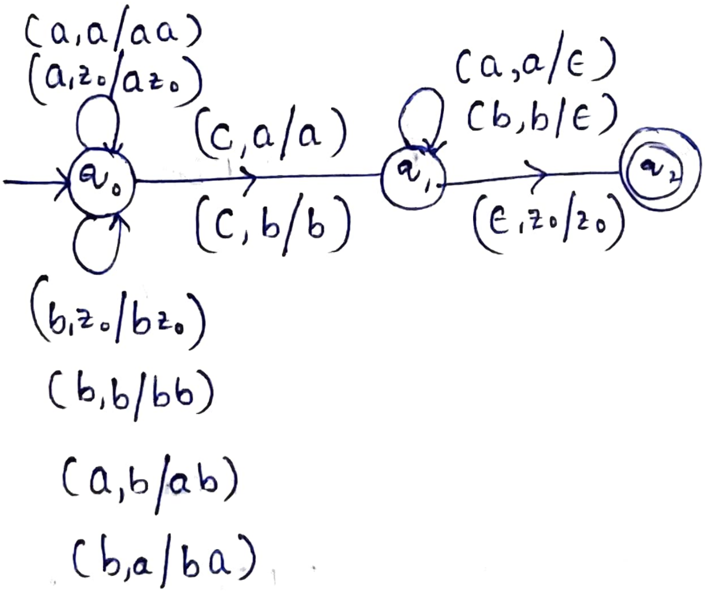
⑧ $L = \{ WCW^R / W \in \{a,b\}^* \}$

abbcbba

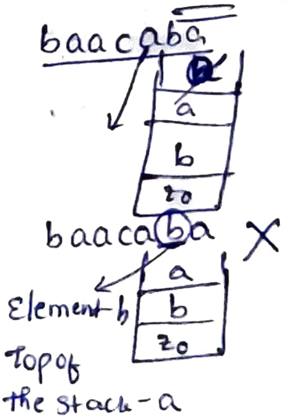
babcbab







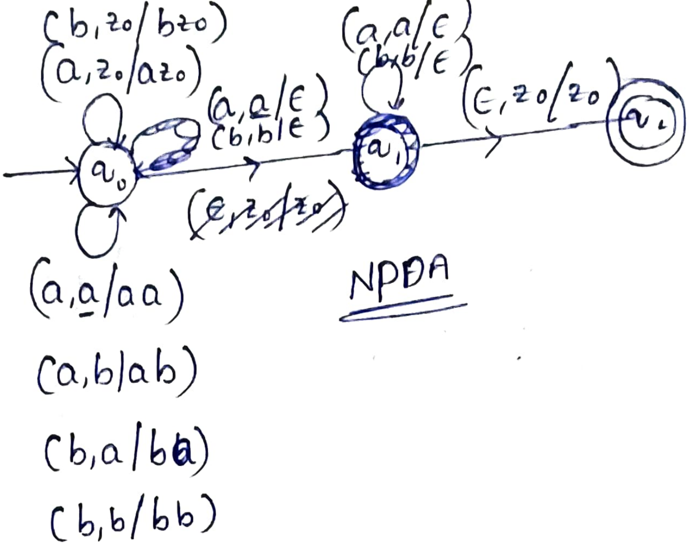
Controversy



⑨ $L = \{ WW^R / W \in (a+b)^* \}$

aabbaa

abaaba



Deterministic PDA:-

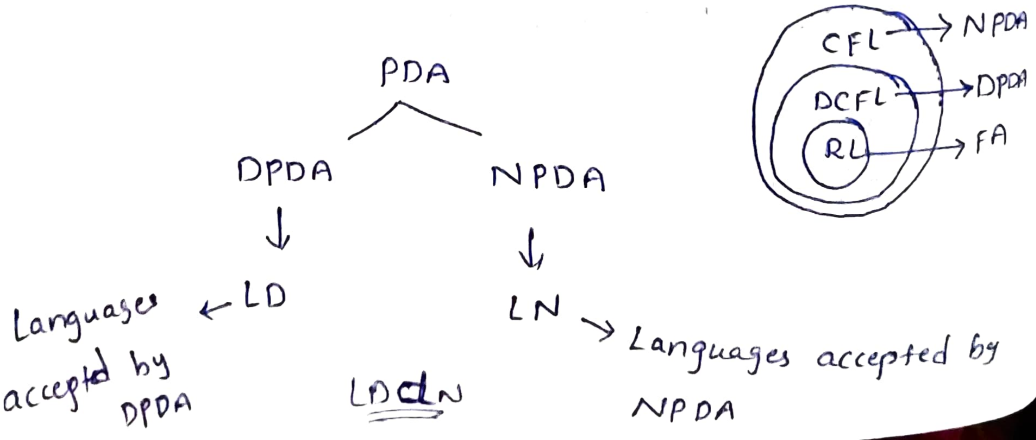
The PDA is said to be deterministic if $\delta(q,x,y)$ has atmost one outcome for every pair (x,y)

→ The language which is accepted by a deterministic PDA is DCFL

Note:- ① Every CFL is accepted by NPDA.

② Every CFL is need not be accepted by DPDA

③ The Language which is accepted by DPDA is also accepted by NPDA but reverse need not be true.



Ex: $L = \{a^m b^m / m=n\} \Rightarrow$ DCFL

$L = \{a^m b^n / m>n\} \Rightarrow$ DCFL

$L = \{a^n c b^n / n\geq 1\} \Rightarrow$ DCFL

$L = \{W C W^R / W \in \{a+b\}^*\} \Rightarrow$ DCFL

$L = \{ |W|a = |W|b \} \; W \in (a+b)^* \} \Rightarrow$ DCFL

$L = \{W W^R / W \in (a+b)^*\} \Rightarrow$ CFL

⇒ Closure properties of CFL

closed	Notclosed
① union L_1-CFL L_2-CFL $L_1 \cup L_2 \rightarrow$ CFL	① Intersection $L_1 \rightarrow$ CFL, $L_2 \rightarrow$ CFL $L_1 \cap L_2 \neq$ CFL
② Concatination L_1- CFL, L_2-CFL $L_1 \cdot L_2 \rightarrow$ CFL	② Compliment
③ kleen closure L-CFL $\Rightarrow$ $L^* \rightarrow$ CFL	③ Difference operator
④ Substitution	④ Symmetric Difference
⑤ Homo morphism	⑤ Quotient
⑥ Inverse Homomorphism	⑥ Inverse substitution
⑦ Intersection with Regular set $L_1 \rightarrow$ CFL, $L_2 \rightarrow$ RL $L_1 \cap L_2 \rightarrow$ CFL	⑦ MIN(L)
	⑧ MAX(L)

⑧ Quotient with Regular sets

⑨ Reverse of CFL is also CFL

⑩ cycle (L)

⑪ INIT (L)

closure properties of DCFL

closed	Not closed
① compliment $L \rightarrow L^c$	① union
② Intersection with RL	② Cancatenation
③ Quotient with RL	③ kleen closure
④ MIN (L)	④ Homomorphism
⑤ MAX (L)	⑤ substitution
⑥ Inverse Homomorphism	⑥ Reversed
	⑦ Intersection