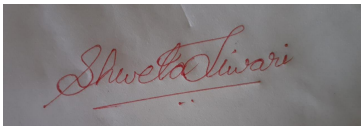
CD: COMPILER DESIGN
CD: UNIT-2 09/2022

SEPTEMBER 2022 / IT-3rd year, Vth semester
FALL SEMESTER, YEAR (Vth, 3rd)
FALL SESSION (2022-23)
(CD)
MS. SHWETA TIWARI
Published: SEPTEMBER, 2022

PREPARED FOR
Engineering Students
All Engineering College

CD: COMPILER DESIGN

INSTRUCTOR: Ms. SHWETA TIWARI
shwetatiwari08@recabn.ac.in
shwetatiwari08aug@gmail.com



September 2022

TOPIC On : QUESTION And ANSWER ON-

Left Recursion, Left Factoring, FIRST() AND FOLLOW()

1. QUESTION And ANSWER ON- Eliminate Left Recursion

1. Consider the following grammar and eliminate left recursion-

   A → ABd / Aa / a

   B → Be / b

Answer-1. The grammar after eliminating left recursion is-

A → aA'

A' → BdA' / aA' / ∈

B → bB'

B' → eB' / ∈

2. Consider the following grammar and eliminate left recursion-

   E → E + E / E x E / a

Answer-2. The grammar after eliminating left recursion is-

E → aA

A → +EA / xEA / ∈

3. Consider the following grammar and eliminate left recursion-

   E → E + T / T

   T → T x F / F

   F → id

Answer-3. The grammar after eliminating left recursion is-

E → TE'

E' → +TE' / ∈

T → FT'

T' → xFT' / ∈

F → id

4. Consider the following grammar and eliminate left recursion-

S → (L) / a

L → L , S / S

Answer-4. The grammar after eliminating left recursion is-

S → (L) / a

L → SL'

L' → ,SL' / ∈

5. Consider the following grammar and eliminate left recursion-

S → S0S1S / 01

Answer-5. The grammar after eliminating left recursion is-

S → 01A

A → 0S1SA / ∈

6. Consider the following grammar and eliminate left recursion-

S → A

A → Ad / Ae / aB / ac

B → bBc / f

Answer-6. The grammar after eliminating left recursion is-

S → A

A → aBA' / acA'

A' → dA' / eA' / ∈

B → bBc / f

7. Consider the following grammar and eliminate left recursion-

A → AAα / β

Answer-7. The grammar after eliminating left recursion is-

$A \rightarrow \beta A'$

$A' \rightarrow A\alpha A' / \in$

8. Consider the following grammar and eliminate left recursion-

A → Ba / Aa / c

B → Bb / Ab / d

Answer-8. The grammar after eliminating left recursion is-

This is a case of indirect left recursion.

Step-01:

First let us eliminate left recursion from A → Ba / Aa / c

Eliminating left recursion from here, we get-

$A \rightarrow BaA' / cA'$

$A' \rightarrow aA' / \in$

Now, given grammar becomes-

$A \rightarrow BaA' / cA'$

$A' \rightarrow aA' / \in$

B → Bb / Ab / d

Step-02:

Substituting the productions of A in B → Ab, we get the following grammar-

$A \rightarrow BaA' / cA'$

$A' \rightarrow aA' / \in$

B → Bb / BaA'b / cA'b / d

Step-03:

Now, eliminating left recursion from the productions of B, we get the following grammar-

A → BaA' / cA'

A' → aA' / ∈

B → cA'bB' / dB'

B' → bB' / aA'bB' / ∈

This is the final grammar after eliminating left recursion.

9. Consider the following grammar and eliminate left recursion-

   X → XSb / Sa / b

   S → Sb / Xa / a

Answer-9. The grammar after eliminating left recursion is-

This is a case of indirect left recursion.

Step-01:

First let us eliminate left recursion from X → XSb / Sa / b

Eliminating left recursion from here, we get-

X → SaX' / bX'

X' → SbX' / ∈

Now, given grammar becomes-

X → SaX' / bX'

X' → SbX' / ∈

S → Sb / Xa / a

Step-02:

Substituting the productions of X in S → Xa, we get the following grammar-

X → SaX' / bX'

X' → SbX' / ∈

S → Sb / SaX'a / bX'a / a

Step-03:

Now, eliminating left recursion from the productions of S, we get the following grammar-

X → SaX' / bX'

X' → SbX' / ∈

S → bX'aS' / aS'

S' → bS' / aX'aS' / ∈

This is the final grammar after eliminating left recursion.

10. Consider the following grammar and eliminate left recursion-

S → Aa / b

A → Ac / Sd / ∈

Answer-10. The grammar after eliminating left recursion is-

This is a case of indirect left recursion.

Step-01:

First let us eliminate left recursion from S → Aa / b

This is already free from left recursion.

Step-02:

Substituting the productions of S in A → Sd, we get the following grammar-

S → Aa / b

A → Ac / Aad / bd / ∈

Step-03:

Now, eliminating left recursion from the productions of A, we get the following grammar-

S → Aa / b

A → bdA' / A'

A' → cA' / adA' / ∈

This is the final grammar after eliminating left recursion.

****

****

****

2. QUESTION And ANSWER ON- Eliminate Left Factoring

1. Do left factoring in the following grammar-

   S → iEtS / iEtSeS / a

   E → b

Answer-1. The grammar after eliminating left factoring is-

S → iEtSS' / a

S' → eS / ∈

E → b

2. Do left factoring in the following grammar-

A → aAB / aBc / aAc

Answer-2. The grammar after eliminating left factoring is-

Step-01:

A → aA'

A' → AB / Bc / Ac

Again, this is a grammar with common prefixes.

Step-02:

A → aA'

A' → AD / Bc

D → B / c

This is left factored grammar.

3. Do left factoring in the following grammar-

S → bSSaaS / bSSaSb / bSb / a

Answer-3. The grammar after eliminating left factoring is-

Step-01:

S → bSS' / a

S' → SaaS / SaSb / b

Again, this is a grammar with common prefixes.

Step-02:

S → bSS' / a

S' → SaA / b

A → aS / Sb

This is left -factored grammar.

4. Do left factoring in the following grammar-

S → aSSbS / aSaSb / abb / b

Answer-4. The grammar after eliminating left factoring is-

Step-01:

S → aS' / b

S' → SSbS / SaSb / bb

Again, this is a grammar with common prefixes.

Step-02:

S → aS’ / b

S’ → SA / bb

A → SbS / aSb

This is left -factored grammar.

5. Do left factoring in the following grammar-

   S → a / ab / abc / abcd

Answer-5. The grammar after eliminating left factoring is-

Step-01:

S → aS’

S’ → b / bc / bcd / ∈

Again, this is a grammar with common prefixes.

Step-02:

S → aS’

S’ → bA / ∈

A → c / cd / ∈

Again, this is a grammar with common prefixes.

Step-03:

S → aS’

S’ → bA / ∈

A → cB / ∈

B → d / ∈

This is left factored grammar.

6. Do left factoring in the following grammar-

   S → aAd / aB

   A → a / ab

   B → ccd / ddc

Answer-6. The grammar after eliminating left factoring is-

S → aS’

S’ → Ad / B

A → aA’

A’ → b / ∈

B → ccd / ddc

This is left factored grammar.

****

****

****

3. QUESTION And ANSWER ON- Find FIRST( ) And FOLLOW( )

1. Calculate the first and follow functions for the given grammar-

S → aBDh

B → cC

C → bC / ∈

D → EF

E → g / ∈

F → f / ∈

Answer-1. The first and follow functions are as follows-

First Functions-

- First(S) = { a }
- First(B) = { c }
- First(C) = { b , ∈ }
- First(D) = { First(E) – ∈ } ∪ First(F) = { g , f , ∈ }
- First(E) = { g , ∈ }
- First(F) = { f , ∈ }

Follow Functions-

- Follow(S) = { $ }
- Follow(B) = { First(D) – ∈ } ∪ First(h) = { g , f , h }
- Follow(C) = Follow(B) = { g , f , h }
- Follow(D) = First(h) = { h }
- Follow(E) = { First(F) – ∈ } ∪ Follow(D) = { f , h }
- Follow(F) = Follow(D) = { h }

Grammar/Production	**FIRST( )**	**FOLLOW( )**
S → aBDh	{ a }	{ $ }
B → cC	{ c }	{ g , f , h }

C → bC / ∈	{ b , ∈ }	{ g , f , h }
D → EF	{ g , f , ∈ }	{ h }
E → g / ∈	{ g , ∈ }	{ f , h }
F → f / ∈	{ f , ∈ }	{ h }

2. Calculate the first and follow functions for the given grammar-

S → A

A → aB / Ad

B → b

C → g

Answer-2. The first and follow functions are as follows-

We have-

- The given grammar is left recursive.
- So, we first remove left recursion from the given grammar.

After eliminating left recursion, we get the following grammar-

S → A

A → aBA'

A' → dA' / ∈

B → b

C → g

Now, the first and follow functions are as follows-

First Functions-

- First(S) = First(A) = { a }
- First(A) = { a }
- First(A') = { d , ∈ }
- First(B) = { b }
- First(C) = { g }

Follow Functions-

- Follow(S) = { $ }
- Follow(A) = Follow(S) = { $ }
- Follow(A') = Follow(A) = { $ }
- Follow(B) = { First(A') – ∈ } ∪ Follow(A) = { d , $ }
- Follow(C) = NA

Grammar/Production	FIRST( )	FOLLOW( )
S → A	{a}	{ $ }
A → aBA'	{a}	{ $ }
A' → dA' / ∈	{d,∈}	{ $ }
B → b	{b}	{d, $ }
C → g	{g}	NA

3. Calculate the first and follow functions for the given grammar-

S → (L) / a

L → SL'

L' → ,SL' / ∈

Answer-3. The first and follow functions are as follows-

First Functions-

- First(S) = { (, a }
- First(L) = First(S) = { (, a }
- First(L') = { , , ∈ }

Follow Functions-

- Follow(S) = { $ } ∪ { First(L') – ∈ } ∪ Follow(L) ∪ Follow(L') = { $, , ,) }
- Follow(L) = {) }
- Follow(L') = Follow(L) = {) }

Grammar/Production	FIRST( )	FOLLOW( )
S → (L) / a	{(, a}	**{$, , , ∈}**
L → SL'	{(,a}	{)}
L' → ,SL' / ∈	{, , ∈}	{)}

4. Calculate the first and follow functions for the given grammar-

S → AaAb / BbBa

A → ∈

B → ∈

Answer-4. The first and follow functions are as follows-

First Functions-

- First(S) = { First(A) – ∈ } ∪ First(a) ∪ { First(B) – ∈ } ∪ First(b) = { a , b }
- First(A) = { ∈ }
- First(B) = { ∈ }

Follow Functions-

- Follow(S) = { $ }
- Follow(A) = First(a) ∪ First(b) = { a , b }
- Follow(B) = First(b) ∪ First(a) = { a , b }

Grammar/Production	FIRST( )	FOLLOW( )
S → AaAb / BbBa	{a,b}	{$}
A → ∈	{∈}	{a,b}
B → ∈	{∈}	{a,b}

5. Calculate the first and follow functions for the given grammar-

E → E + T / T

T → T x F / F

F → (E) / id

Answer-5. The first and follow functions are as follows-

We have-

- The given grammar is left recursive.
- So, we first remove left recursion from the given grammar.

After eliminating left recursion, we get the following grammar-

$E \rightarrow TE'$

$E' \rightarrow + TE' / \in$

$T \rightarrow FT'$

$T' \rightarrow x FT' / \in$

$F \rightarrow (E) / id$

Now, the first and follow functions are as follows-

First Functions-

- First(E) = First(T) = First(F) = { (, id }
- First(E') = { + , ∈ }
- First(T) = First(F) = { (, id }
- First(T') = { x , ∈ }
- First(F) = { (, id }

Follow Functions-

- Follow(E) = { $,) }
- Follow(E') = Follow(E) = { $,) }
- Follow(T) = { First(E') – ∈ } ∪ Follow(E) ∪ Follow(E') = { + , $,) }
- Follow(T') = Follow(T) = { + , $,) }
- Follow(F) = { First(T') – ∈ } ∪ Follow(T) ∪ Follow(T') = { x , + , $,) }

Grammar/Production	FIRST( )	FOLLOW( )
E → TE'	{(,id}	{$,)}
E' → + TE' / ∈	{+,∈}	{$,)}
T → FT'	{(,id}	{+,$,)}
T' → x FT' / ∈	{x,∈}	{+,$,)}
F → (E) / id	{(,id}	{x,+,$,)}

6. Calculate the first and follow functions for the given grammar-

S → ACB / CbB / Ba

A → da / BC

B → g / ∈

C → h / ∈

Answer-6. The first and follow functions are as follows-

First Functions-

- First(S) = { First(A) – ∈ } ∪ { First(C) – ∈ } ∪ First(B) ∪ First(b) ∪ { First(B) – ∈ } ∪ First(a) = { d , g , h , ∈ , b , a }
- First(A) = First(d) ∪ { First(B) – ∈ } ∪ First(C) = { d , g , h , ∈ }
- First(B) = { g , ∈ }
- First(C) = { h , ∈ }

Follow Functions-

- Follow(S) = { $ }
- Follow(A) = { First(C) – ∈ } ∪ { First(B) – ∈ } ∪ Follow(S) = { h , g , $ }
- Follow(B) = Follow(S) ∪ First(a) ∪ { First(C) – ∈ } ∪ Follow(A) = { $, a , h , g }

- Follow(C) = { First(B) – ∈ } ∪ Follow(S) ∪ First(b) ∪ Follow(A) = { g , $, b , h }

Grammar/Production	**FIRST( )**	**FOLLOW( )**
S → ACB / CbB / Ba	{a,b,d,g,h,∈}	{$}
A → da / BC	{d,g,h,∈}	{h,g,$}
B → g / ∈	{g,∈}	{$,h,g,a}
C → h / ∈	{h,∈}	{b,g,$,h}

****

****

****