A 🡪 BA’

B 🡪 (A) | a implies ) is in Follow(A)

List all productions with A’ on right hand side.

A🡪 BA’ Members of Follow(A) are in Follow of (A’)

A’🡪+BA’ not adding anything to Follow(A’)

A’🡪-BA’ not adding anything to Follow(A’)

A’🡪\*BA’ not adding anything to Follow(A’)

List all productions with B on the right hand side

A🡪 BA’ (First(A’) – lambda) are in Follow B. Follow(B) = (First(A’) – lambda)

A’ 🡪 λ Concludes that A🡪 B λ which means A 🡪 B that means Follow(A) are in Follow(B)

A’🡪+BA’ Members of Follow (A’) are in Follow B not adding anything as we already did A’ - - lambda

A’🡪-BA’ Doesn’t add anything

A’🡪\*BA’ Doesn’t add anything

Therefore follow(B) = (First(A’) – lambda) U Follow(A)

First(S) = { if i}

First(L) = First( E ) = {i d }

First C = { **λ , }**

**First(E) = {I d}**

**Move on to follow**

All values with S on right side

**S 🡪 if ( E ) S else S; implies Follow (S) in Follow(S).** doesn’t add anything.

**All values with L in right hand side.**

**S 🡪 I (L) implies ) is in Follow(L).**

All values with C in right hand side.

L 🡪 EC

C 🡪 , EC

Members of Follow(L) and Follow( C ) are in Follow(C) ={ ) }

All values with E in the right hand side.

S 🡪 if (E) S else S First(S) are in Follow(E)

L 🡪 EC (First(c) – **λ) are in** Follow(E)

C 🡪, EC First(C) – **λ)** are in Follow (E)

C 🡪 **λ implies** L 🡪 EC and C 🡪 , EC can be L 🡪 E and C 🡪 , E which concludes Follow(L) and Follow(C) are in E

**Thus we have** for follow(E) = First(S) U (First(C) – **λ** ) U (Follow(L)) U Follow(C)

First values on problem 3

Start S:

S 🡪 AB First(S) = First(A)

S 🡪 Cf First(S) = First(C)

A 🡪 **λ** Since A = **λ** , S🡪 AB can be S 🡪 **λB = B**  thus First(S) = First (B)

First(S) = First(A) U First(C) U First(B)

A:

A 🡪 ef

A 🡪 **λ**

First(A) = e, **λ**

B:

B 🡪 hg

First(B) = h

C:

C 🡪 DD

C 🡪 fi

First(C) = First(D) U f

First(C) = g U f = {g f}

D:

D 🡪 g

First(D) = g

Now we move on to Follow

S on righthand side:

S 🡪 AB Since S starts it gets {$}

A on right hand side:

S 🡪 AB First (B) members are in follow A.

B on right hand side:

S 🡪 AB members of Follow(S) are in Follow(B)

C on right hand side:

S 🡪 Cf f is in Follow(C)

D on right hand side:

C 🡪 DD

Follow(D) = First(D) is in Follow(D). + Members of Follow(C) are in Follow(D)

Follow(D) = First(D) U Follow(C)

First set in problem #4.

Start with S.

S:

S 🡪 TU First(S) = First(T)

T 🡪 λ S 🡪 λ U = U S 🡪 U First(S) = First(U)

First(S) = First(T) U First(U)

T:

T 🡪 aVa First(T) = a

T 🡪 λ First(T) = λ

First(T) = { a λ }

U:

U 🡪 bVT First(U) = b

First(U) = b

V:

V 🡪 Ub First(V) = First(U)

V 🡪 d First(V) = d

First(V) = First(U) U d

First(V) = {b d}

Righthand side has S:

S 🡪 TU Follow(S) = { $ }

Follow(S) = {$}

Righthand side T:

S 🡪 TU Follow(T) = First(U)

U 🡪 bVT Follow(T) = Follow(U)

Follow(T) = First(U) U Follow(U)

Righthand side of U:

S 🡪 TU Follow(U) = Follow(S)

V 🡪 Ub Follow(U) = b

Follow(U) = Follow(S) U b

Righthand side of V:

T 🡪 aVa Follow(V) = a

U 🡪 bVT Follow(V) = (First(T) – **λ)** , Follow(V) = Follow(U) when U 🡪 bVλ = bV

Follow(V) = (First(T) – λ) U Follow(U)

Quiz work

**A**→ **A + B**

1. → **A \* B**

**A**→ **B**

1. → **B\*D**

**B**→ **D**

**D**→ **( A)**

**D**→ **a | d**

Remove all left recursions

A 🡪 BA’

A’ 🡪 +BA’

A’ 🡪 \*BA’

A’ 🡪 λ

B’ 🡪 \*DB’

B 🡪 DB’

B’ 🡪 λ

D 🡪 (A)

D 🡪 a | d

First start with A

A 🡪 BA’ First(A) = First(B)

First(A) = First(B)

First(A) = a d (

A’:

A’ 🡪 +BA’ First(A’) = +

A’ 🡪 \*BA’ First(A’) = \*

A’ 🡪 λ First(A’) = λ

First(A’) = + \* λ

B:

B 🡪 DB’

First(B) = First(D)

First(B) = a d (

B’:

B’ 🡪 \*DB’ First(B’) = \*

B’ 🡪 λ First(B’) = λ

First(B’) = \* λ

D:

D 🡪 (A) First(D) = (

D 🡪 a | d First(D) = a d

First(D) = a d (

Then Follow

Righthand side A:

D 🡪 (A) Follow(A) = ) and $

Right hand side A’:

A 🡪 BA’ Follow(A’) = Follow(A)

A’ 🡪 +BA’ Follow(A’) = Follow(A’) doesn’t do anything.

A’ 🡪 \*BA’ Follow(A’) = Follow(A’) doesn’t do anything.

Follow(A’) = Follow(A)

Right hand side of B:

A 🡪 BA’ Follow(B) = (First(A’) – λ)

A’ 🡪 λ since this is possible. A 🡪 B λ = B thus Follow(B) = Follow(A)

Follow(B) = (First(A’) – λ) U Follow(A)

Right hand side of B’:

B 🡪 DB’ follow(B’) = Follow(B)

B’ 🡪 \*DB’ follow(B’) = Follow(B’) redundant.

Follow(B’) = Follow(B)

Righthand side of D:

B 🡪 DB’ Follow(D) = (First(B’) – λ)

B’ 🡪 \*DB’ Follow(D) = (First(B’) – λ) redundant.

B’ 🡪 λ thus it can be possible for B’ 🡪 \*D Follow(D) = Follow(B’)

Thus,

Follow(D) = (First(B’) – λ) U Follow(B’)

Follow(D) = + \* $ )