

1. FIRST (aB) = {a}

FIRST (b) = {b}

FIRST (CBB) = {c}

Passed – doesn’t intersect

1. FIRST (aB) = {a}

FIRST (bA) = {b}

FIRST (aBb) = {a}

Fail – Intersection

1. FIRST (aaA) = {a}

FIRST (b) = {b}

FIRST (caB) = {c}

Passed – doesn’t intersect



Call lex(): next lexeme is a

Enter <expr>

Enter <term>

Enter <factor>

Call lex(): next lexeme is +

Exit <factor>

Exit <term>

Call lex(): next lexeme is b

Enter <term>

Enter <factor>

Call lex(): next lexeme is \*

Exit <factor>

Call lex(): next lexeme is c

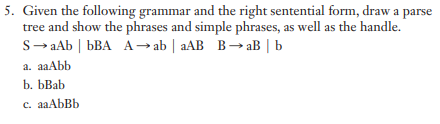
Enter <factor>

Call lex(): next lexeme is EOF

Exit <factor>

Exit <term>

Exit <expr>



Parse tree:

S

/ | \

a A b

/ | \

a A B

\

b

Phrases: aaAbb, aaABb, aAb

Simple Phrase: b

Handles: b, aAB

b.

Parse Tree:

S

/ | \

b B A

/ \

a b

Phrases: bBab, bBA

Simple Phrase: ab

Handles: ab

c.

S

/ | \

a A b

/ | \

a A B

/ \

A b

*Cannot derive from current grammar*