River Kelly

CSCI-305

Chapter 4 Homework

**1. Perform the pairwise disjointness test for the following grammar rules.**

**a.) S → aSb | bAA**

*Step 1.*

*FIRST*(aSb) = {a}

*FIRST*(bAA) = {b}

*Step 2.*

*FIRST*(aSb) ∩ *FIRST*(bAA) = ∅

Is the same as,

{a} ∩ {b} = ∅

*Answer*: **Passes the test**

**b. A → B{aB} | a**

*Step 1.*

*FIRST*(B{aB}) = {b}

*FIRST*(a) = {a}

*Step 2.*

*FIRST*(B{aB}) ∩ *FIRST*(a) = ∅

Is the same as,

{b} ∩ {a} = ∅

*Answer*: **Passes the test**

**c. B → aB | a**

*Step 1.*

*FIRST*(aB) = {a}

*FIRST*(a) = {a}

*Step 2.*

*FIRST*(aB) ∩ *FIRST*(a) = {a} ≠ ∅

Is the same as,

{a} ∩ {a} = {a} ≠ ∅

*Answer*: **Fails the test**

**2. Show a trace of the recursive descent parser given in Section 4.4.1 (in the book) for the string a + b \* c .**

Next token is: 11 Next lexeme is a

Enter <expr>

Enter <term>

Enter <factor>

Next token is: 21 Next lexeme is +

Exit <factor>

Exit <term>

Next token is: 11 Next lexeme is b

Enter <term>

Enter <factor>

Next token is: 23 Next lexeme is \*

Exit <factor>

Next token is: 11 Next lexeme is c

Enter <factor>

Next token is: -1 Next lexeme is EOF

Exit <factor>

Exit <term>

Exit <expr>

**3. Given the following grammar and the right sentential form, draw a parse tree and show the phrases and simple phrases, as well as the handle.**

|  |  |  |
| --- | --- | --- |
| S→aAb | bBA | A→ab | aAB | B→aB | b |

**a. aaAbb**

|  |  |
| --- | --- |
| **Phrases:** aaAbb, aaABb, aAb  **Simple Phrases:** b  **Handle:** b |  |

**b. bBab**

|  |  |
| --- | --- |
| **Phrases:** bBab, bBA  **Simple Phrases:** ab  **Handle:** ab |  |

**c. aaAbBb**

aaAbBb → aSBb → aSBB → X

The last string cannot be derived from the given grammar, therefor the phrase, simple phrase or handle cannot be calculated.

**4. Design a state diagram to recognize the floating-point literals of your favorite programming language. Similar to the state diagram from the book that I used numerous times in the lecture videos.**

