**Assignment 5 (20 points), SE 421 due: Wednesday, 9/29/2021**

**Name (Last, First): Ogbondah, Chimzim**

**Submission Requirement**: Submit a PDF named HW5-lastname-firstname. Include the top two lines with your last and the first name. Include the problem statement followed by your answer.

**Pre-requisite**:

1. Review Lecture Notes 6.1, 6.2, 7.1, 7.2, 8, 9, 10., 10.2
2. Review Atlas Demo Slides.
3. Read all the problems carefully and ask questions if you have doubts. Try to do so by Monday.

**Problem 1 (5 points):** Map XINU and execute the following commands in Atlas shell. Include the screenshots as mentioned in the comments. Answer the following:

1. Explain precisely but in as short as possible why show(a) and show(b) produce the same result.
   1. You are constructing or getting to the same graph via the selected callsite by either going up the edges or down the children
2. Give number of nodes and containment edges in the result.
   1. Nodes = 5
   2. Edges = 4

x1 = functions("dswrite")

x2 = x1.children.nodes(XCSG.ControlFlow\_Node)

x3 = x2.children.nodes(XCSG.SimpleCallSite)

show(x3)

\\* From the graph produced by show(x3), select the call site node for call to dskenq() \*\

\\* Ensure that you have indeed selected the call

site node by checking its XCSG tag in Element Detail View (EDV) \*\

\\* Include the screen shot of EDV in your answer \*\

Graphical user interface, application

Description automatically generated

var a = edges(XCSG.Contains).reverse(selected)

show(a)

\\* Include the screen shot of the graph produced by show(a) \*\



show(x2)

\\* Select the control flow node which has call to dskenq() \*\

\\* Check the XCSG tag of the selected node in Element Detail View \*\

var b = selected.children.nodes(XCSG.SimpleCallSite)

show(b)

\\* Include the screen shot of the graph produced by show(b)\*\

Graphical user interface, application, chat or text message

Description automatically generated

**Problem 2 (7 points):** Use the *enumeration algorithm* taught in class.

1. List the first *four* paths in the specific order they are generated by the algorithm. (2pts.).

Path 1: C1, C2, D, F, G

Path 2: C1, C2, C4, D, F, G

Path 3: C1, C2, C4, F, G

Path 4: C1, C3, C4, D, F, G

1. What is the total number paths? (1pt.)
   1. 6
2. To generate all paths, how many times the *forward pass* would set Bf to 1? (1pt.)

Numbers for C1: **3** C2: **1** C3: **2** C4: **2**

1. To generate all paths, how many times the *backward pass* would set Bf to 0? (1pt.)

Numbers for C1: **3** C2: **2**  C3: **1** C4: **2**

1. Path = C1, C3, C4, D, F, G; the control goes to the backward pass. Give both the passes for the remaining iterations of the algorithm. Follow the format shown in the class. (2pts.)
   1. C1 C3 C4 F
   2. C1 C3 E F G

G

1

0

1

0

0

D

F

C1

C3

C2

1

E

C4

1

0

**Problem 3 (2 points)**: Use the *enumeration algorithm* taught in class. Path = C1, C2, C4, F; the control goes to the *backward pass*. Give the next three passes of the enumeration algorithm. Follow the format shown in the lecture note.

1. Path 1: C1, C2, C4, G
2. Path 2: C1, C3, C4, F
3. Path 3: C1, C3, C4, G

G

1

0

1

0

0

D

F

C1

C3

C2

1

E

C4

0

1

**Problem 4 (4 points)**: Use the *multiplicity algorithm* taught in class. Give all the forward and backward iterations needed to compute the multiplicities of the leaf nodes F and G. Show only those flags Bf(C1), Bf(C2), Bf(C4), Bf(C5), Rf(C4), or Rf(F) that are changed by an iteration. Show only those multiplicities M(C4) or M(F) that are changed by an iteration. Follow the format shown in the lecture note.

G

1

0

1

0

0

D

F

C1

C3

C2

1

E

C4

0

1

|  |  |  |
| --- | --- | --- |
| **Path:** | **Flag** | **Multiplicity** |
| **F1:** C1-C2-D-F | Bf(C1) = 1 | M(F) = 1 |
| **B1:** C1-C2-C4 | Bf(C1) = 1, Bf(C2) = 2,  Bf(C4) = 1, Rf(C4) = 3 | M(C4) = 1, M(F) = 1 |
| **F1:** C1-C3-C4 | Bf(C1) = 0,  Bf(C4) = 2, Rf(C4) = 1 | M(C4) = 2, M(F) = 1 |
| **B1:** C1-C3-E- C4-G | Bf(C1) = 2,  Bf(C4) = 2, Rf(C4) = 0 | M(C4) = 3, M(F) = 1 |
| **F1:** C1-C3-C4-G | Bf(C1) = 0, Bf(C3) = 1, Bf(C4) = 0, | M(C4) = 3, M(F) = 1 |
| **B1:** C1-C2-C4-G | Bf(C1) = 1, Bf(C2) = 1, Bf(C4) = 0, | M(C4) = 3, M(F) = 1 |
| **F1:** C1-C2-C4-F | Bf(C1) = 1, Bf(C2) = 1, Bf(C4) = 2, | M(C4) = 3, M(F) = 4 |
| **B1:** C1-C3-C4-F  **F1:** C1-C3-E-C4-F | Bf(C1) = 0, Bf(C3) = 1, Bf(C4) = 1, Bf(C1) = 0, Bf(C3) = 0, Bf(C4) = 1, | M(C4) = 3, M(F) = 4  M(C4) = 3, M(F) = 4 |

**Problem 5 (2 points)**: Refer to the above Graph in Problem 4.

1. How many times the edge (C4 F) will be appended by the *enumeration* algorithm?
   1. 3
2. How many times the edge (C4 F) will be appended by the *multiplicity* algorithm?
   1. 3
3. How many times the flag Bf(C4) will be set to 1 by the *enumeration* algorithm?
   1. 3
4. How many times the flag Bf(C4) will be set to 1 by the *multiplicity* algorithm?
   1. 1