Michael Smith

CSE 570

Spring 2018

Problem Sets

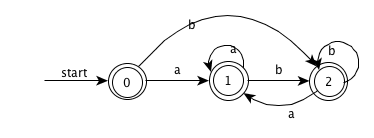
**3.3.6**

Write character classes for the following sets of characters:

1. The first ten letters (up to "j") in either upper or lower case.
   1. [A-Ja-j]
2. The lowercase consonants.
   1. [bcdfghjklmnpqrstvwxyz]
3. The "digits" in a hexadecimal number (choose either upper or lower case for the "digits" above 9).
   1. [0-9a-f]
4. The characters that can appear at the end of alegitimate English sentence (e.g. , exclamation point)
   1. [.?!]

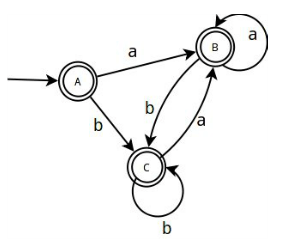
**3.4.2**

((ε|a)b\*)\*

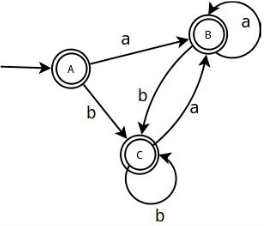


**3.9.3**

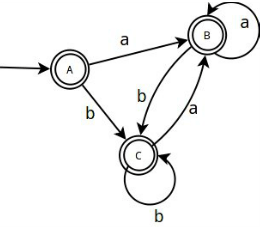
(a|b)\*



(a\*|b\*)\*



((ε|a)|b\*)\*



**4.2.3**

Design grammars for the following languages:

1. The set of all strings of 0s and 1s such that every 0 is immediately followed by at least one 1.
   1. S -> (0?1)\*
2. ! The set of all strings of 0s and 1s that are palindromes; that is, the string reads the same backward as forward.
   1. S -> 0S0 | 1S1 | 0 | 1 | ε
3. ! The set of all strings of 0s and 1s with an equal number of 0s and 1s.
   1. S -> 0S1S | 1S0S | ε
4. !! The set of all strings of 0s and 1s with an unequal number of 0s and 1s.
   1. S -> 0S1 | 1S0 | SS | λ
5. ! The set of all strings of 0s and as in which 011 does not appear as a substring.
   1. S -> 1\*(0+1?)\*

**5.2.3**

Suppose that we have a production A -> BCD. Each of the four nonterminals A,B,C,D have two attributes. S is a synthesized attribute, and I is an inherited attribute. Four each of the sets of rules below, tell whether 1) rules are consistent with an S attributed definition 2) the rules are consistent with an L attributed definition and 3) whether the rules are consistent with any evaluation order at all?

1. A.s = B.i + C.s
   1. No contains inherited attribute
   2. Yes from above or from the left
   3. Yes L attributed so no cycles
2. A.s = B.i + C.s , D.i = A.i + B.s
   1. No contains inherited attributes
   2. Yes from above or from the left
   3. Yes L attributed so no cycles
3. A.s = B.s + D.s
   1. Yes all are synthesized
   2. Yes all are systhesized
   3. Yes both S and L attributed
4. ! A.s = D.i , B.i = A.s + C.s , C.i = B.s , D.i = B.i + C.i
   1. No contains inherited attributes
   2. No B.i uses A.s which depends on D.i which then depends on B.i
   3. No cycle means no evaluation orders.

**6.1.1**

Construct the DAG for the expression

((x+y) - ((x+y) \* (x-y))) + ((x+y) \*(x-y))

* 1. a + a \* (b-c) + (b-c) \* d

**8.4.1**

B1

i = 0

B2

if i >= n goto(13)

B3

j = 0B4

if j >= n goto(11)

B5

t1 = n \* i

t2 = t1 + j

t3 = t2 \* 8

c[t3] = 0.0

j = j + 1

goto(4)

B6

i = i + 1

goto(2)

B7

i = 0

B8

if i >= n goto(40)

B9

j = 0

B10

if j >= n goto(38)

B11

k = 0B12

if k >= n goto(36)

B13

t4 = n \* i

t5 = t4 + j

t6 = t5 \* 8

t7 = c[t6]

t8 = n \* i

t9 = t8 + k

t10 = t9 \* 8

t11 = a[t10]

t12 = n \* k

t13 = t12 + j

t14 = t13 \* 8

t15 = b[t14]

t16 = t11 \* t15

t17 = t7 + t16

c[t6] = t17

k = k + 1

goto(18)

B14

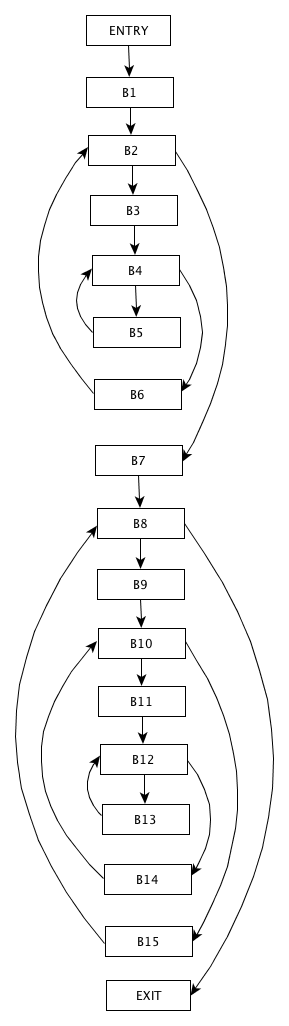
j = j + 1

goto(16)

B15

i = i + 1

goto(14)



{B2, B3, B4, B6}

{B4, B5}

{B8, B9, B10, B15}

{B10, B11, B12, B14}

**8.5.1**

Construct the DAG for the basic block

d = b \* c

e = a + b

b = b \* c

a = e – d

