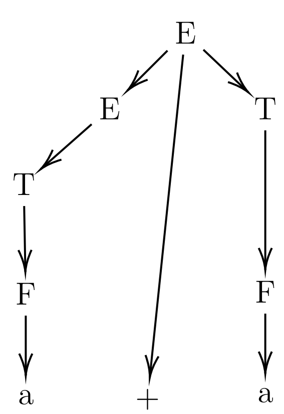
COMP.3040 Homework 1

June 22, 2019

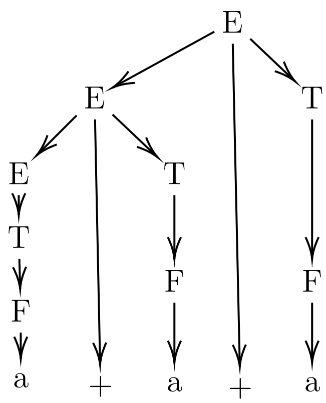
Problem 2.1.

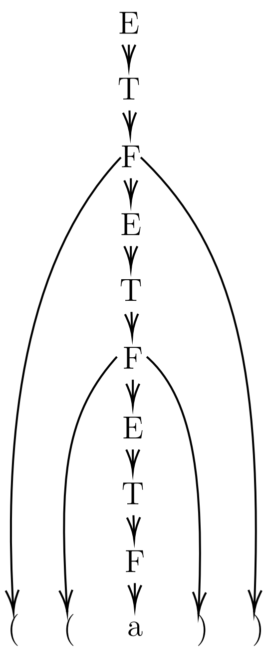
(a)

(b)



(c)





(d)

Problem 2.4.

(b) $G_b = (\{S, S_0\}, \{0,1\}, R, S)$
R:
$S \to 0S_00 \mid 1S_01$
$S_0 \to 0S_0 \mid 1S_0 \mid 0 \mid 1$

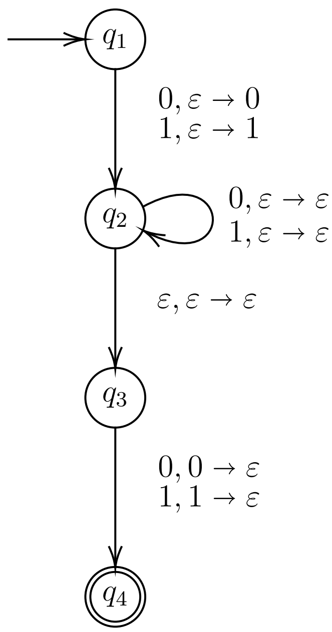
(c) $G_c = (\{S, S_0\}, \{0,1\}, R, S)$
R:
$S \to 0S_0 \mid 1S_0$
$S_0 \to 00S_0 \mid 01S_0 \mid 10S_0 \mid 11S_0 \mid 00 \mid 01 \mid 10 \mid 11$

(e) $G_e = (\{S\}, \{0,1\}, R, S)$
R:
$S \to 0S0 \mid 1S1 \mid 1 \mid 0 \mid 00 \mid 11$

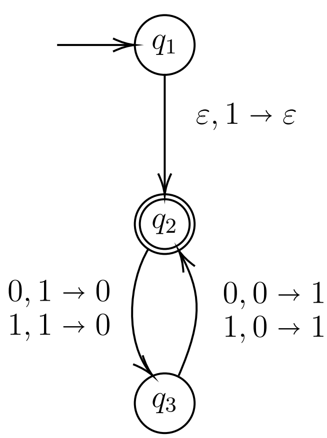
(f) $G_b = (\{S\}, \{0,1\}, R, S)$
R:
$S \to S$

Problem 2.5.

(b) 1. Read first input and push to stack.
2. Skip all following input till reads the last input. If last input equals to symbol at top of stack, accept input.

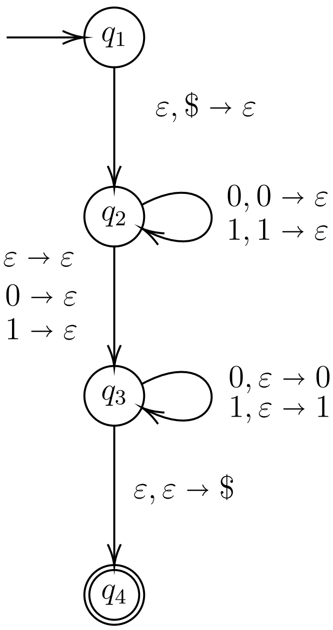


(c) 1. Push 1 to stack.
2. Read input. If number in stack is 1 accept input. Pop stack and push inverse symbol to stack.
3. Repeat Step 2.

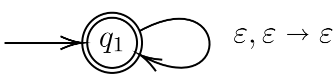


(e) 1. Keep reading input and push to stack.
2. When reach to middle of string, if there is a center symbol, skip it.

3. Read input and pop stack. If input does not equal to popped symbol or input is finished before, reject input.
4. When empty stack and finish input, accept input.



(f) Ignore all input.



Problem 2.6.

(b) $G_b = (\{S, S_a, S_{ab}, S_{aba}, S_b, S_{ba}, S_{bab}, S_x\}, \{a, b\}, R, S)$
R:
$S \to ba \mid aS_a \mid bS_b$
$S_a \to aS_a \mid aS_{ab}$
$S_{ab} \to bS_{ab} \mid bS_{aba}$
$S_{aba} \to aS_x \mid a$
$S_b \to bS_b \mid bS_{ba}$
$S_{ba} \to aS_{ba} \mid aS_{bab}$
$S_{bab} \to bS_x \mid b$
$S_x \to aS_x \mid bS_x \mid a \mid b$

(d) $G_d = (\{S, A, B\}, \{a, b\}, R, S)$
R:
$S \to A\#B\#A$
$A \to aA \mid bA \mid \#A \mid a \mid b$
$B \to aBa \mid bBb \mid aa \mid bb \mid a \mid b \mid \#A\#$

Problem 2.9.

$G_b = (\{S, A_1, A_2, C_1, C_2\}, \{a, b, c\}, R, S)$
R is
$S \to A_1 \mid C_1 \mid \varepsilon$
$A_1 \to aA_1 \mid a \mid bA_2c \mid bc$
$A_2 \to bA_2c \mid bc$
$C_1 \to C_1c \mid c \mid aC_2b \mid ab$
$C_2 \to aC_2b \mid ab$

There are two ways to generate string that are $i = j = k$, so it is ambiguous.

Problem 2.10.

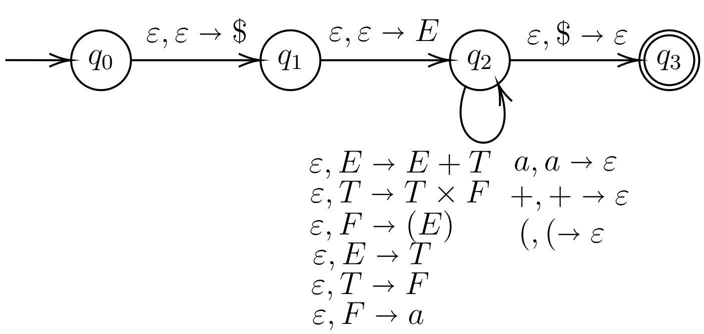
There are two cases:
Case 1 (i = j):

1. Each time reads an a from input, push a to stack.
2. When input changes to b, pop stack each time input read.
3. When input changes to c and stack is empty, input is accepted.

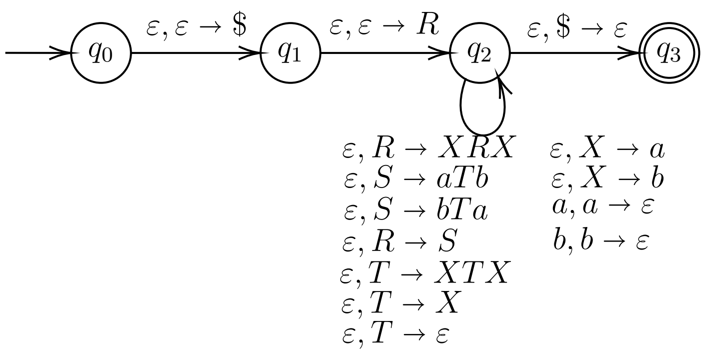
Case 2 (j = k):

1. Do nothing when input is a.
2. When input changes to b, push b to stack each time input read.
3. When input changes to c, pop stack each time input read.
4. If stack is empty when finish reading all inputs, accept input.

Problem 2.11.



Problem 2.12.



Problem 2.13.

(a) There are two cases.
Case 1: N 0s and 2N 0s are separated by a # where N is non negative integers.
Case 2: Three sets of 0s are separated by 2 # where number of 0s in each set is any non negative integers.

(b) In case 1 mentioned above, number N need to be remembered to check if number of N in second set of 0s is correct. N can be infinite; therefore, the gammer is not regular.

Problem 2.14.

$S \to BA_1 \mid BA \mid AB \mid 00 \mid \varepsilon$
$A \to BA_1 \mid BA \mid AB \mid 00$
$A_1 \to AB$
$B \to 00$

Problem 2.26.

If $n = 1$, steps required is 1 where $2 \times 1 - 1 = 1$ which is correct because single element takes one step to derive. Let w_k be a string with k length and requires $2k-1$ steps to derive. Increase length of w_k by 1 by functions $A \to BC$, $B \to b$ and $C \to c$. Steps required is also increased by 2. w_{k+1} takes $2k - 1 + 2 = 2k + 1$ steps. According to the proving function, length of w_{k+1} is $2(k + 1) - 1 = 2k + 1$ which is same as the result above.

According to mathematical induction, exactly $2n-1$ steps are required for any derivation of w where n is length of the string.

Problem 2.30.

(a) Assume the grammer is context free and let p be pumping length. Consider $s = 0^p1^p0^p1^p$ is a member of the grammer. By the pumping lemma, we can write s as $uvxyz$.

Suppose that v or y contains both 0s and 1s, order of 0s and 1s in string uv^ixy^iz can not be correct.

Suppose that v and y only contains single type of numbers, in string uv^ixy^iz, numbers each set of 0s and 1s cannot be the same.

(d) Assume the grammer is context free and let p be pumping length. Consider $s = a^pb^p\#a^pb^p$ is a member of the grammer. By the pumping lemma, we can write s as $uvxyz$.

Suppose x contains #, in uv^ixy^iz, all a are in sequence after #, but a are separated by bs before #. Thus, x cannot contain #.

Suppose v or y contains #, uv^ixy^iz contains more than one #, which is not member of s.

Suppose u or z contains #, v and y are at same side of #, which means there are more symbols on one side of #. In $a^pb^p\#a^pb^p$, there are same number of symbols in both side. u or z cannot contain #.

Problem 2.31.

Assume the grammer is context free and let p be pumping length. Consider $s = 0^p1^{2p}0^p$ is a member of the grammer. By the pumping lemma, we can write s as $uvxyz$.

Suppose v and y only contain one type of number, in uv^ixy^iz, either number of 1s and 0s are not equal or number of left and right 0s are not the same.

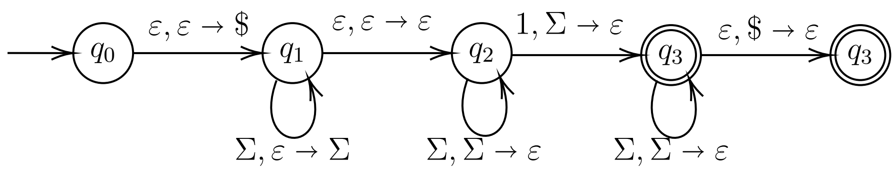
Suppose v or y contains both 0s and 1s, in uv^ixy^iz, order of numbers cannot be correct. Thus it is not member of s.

Problem 2.32.

Assume the grammer is context free and let p be pumping length. Consider $s = 1^p3^q2^p4^q$ is a member of the grammer. By the pumping lemma, we can write s as $uvxyz$, $|vxy| \leq p$ and uv^ixy^iz is member of s. To keep number of 1s and 2s equal and number of 3s and 4s equal, v and y must contains 1 and 2 or 3 and 4 and so does vxy. Since $|vxy| \leq p$, vxy can only contains 1,3 or 3,2 or 2,4. There is no way to have equal number of 1s and 2s and equal number of 3s and 4s in uv^ixy^iz. Therefore, such grammer is not context free.

Problem 2.47.

(a)



(b) $G = (\{S, S_1, S_2\}, \{0,1\}, R, S)$
R is
$S \to \Sigma S \Sigma \mid S_1$
$S_1 \to \Sigma S_2 1 \mid \Sigma 1$
$S_2 \to \Sigma S_2 \Sigma \mid \Sigma S_2 \mid \Sigma\Sigma \mid \Sigma$