**Birla Institute of Technology and Science, Pilani**

**First Semester 2013-14**

**CS C351/CS F351 (Theory of Computation), CLOSED BOOK**

**Time: 90 mins MM: 60**

**[Note: Answer all the parts of a question together.]**

**Q1**. Consider the following two languages L1 and L2 over Σ = {a, b} and answer the parts that follow:

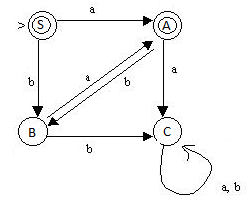
L1 = {w ∈ Σ\* : all a's occur in pairs}

L2 = {w ∈ Σ\* : w contains the sub string bbb}

1. Construct two deterministic DFA's M1and M2 corresponding to L1 and L2.
2. Construct two DFA's M1' and M2', from M1 and M2, that recognize languages (Σ\* - L1) and (Σ\* - L2) respectively.
3. Construct a non-deterministic Finite Automata M3, using the construction for union and the machines M1' and M2', that recognize the language ((Σ\* - L1) ∪ (Σ\* - L2)).
4. Construct machine M4, which is a deterministic machine equivalent to M3.
5. Finally, convert M4 to accept the language L1 ∩ L2.
6. Give a Regular Expression for the language accepted by M4, i.e. L1∩ L2.

[**2x2+1x2+2+4+1+2 = 15M**]

**Q2**. Give the *Regular Grammar* (with rules as per Chomsky classification of languages) for the language accepted by the following DFA. Specify clearly the four tuples of the regular grammar, i.e. (V,Σ, R, S). [**8M**]



**Sol.** The DFA is ({S, A, B, C}, {a, b}, delta, S, {S, A}), where transitions are shown in figure.

The same language is generated by a grammar G = (V, Σ, R, S)

where: V = K U Σ = { S, A, B, C, a, b}

Σ = {a, b}

S = s = S

R = { S → aA | bB | e

A → bB | aC | e

B → aA | bC

C → aC | bC

}

**Q3**. A Context Free Grammar G is in CNF if every rule is of the form A → a, or A → BC, where A, B, C ϵ (V-Σ) and a ϵ Σ. The rule S → e is in G if e ϵ L(G), but in this case we assume that S does not appear on RHS of any rule. Consider a CFG **G =({S, A, B, C, 0, 1, 2}, {0, 1, 2}, R, S)**, where **R** is given by:

**S → BAC**

**B → 0B | e**

**A → 1A | e**

**C → 22C | e**

Convert **G** in CNF.

[**12M**]

**Q4**. Consider a CFG G = ({S, A, B, a, b, c}, {a, b, c}, R, S), where R is given as:

**S** **→ aABb**

**A → bAb | c**

**B → aB | b**

Answer the following questions. For PDA's, assume that # is the end marker symbol for input string and $ is the bottom marker symbol for the stack.

i) The language accepted by G is of the form: **abicbjakbb**. Precisely, what is the language accepted (i.e. give relations (if any) between i, j, and k and constraints (if any) on i, j, and k). [***Do it very carefully as next three parts are based on this.***]

**Sol:** i = j AND i, j, k >= 0.

ii) Give a graphical representation of a deterministic PDA for accepting L(G). This PDA should work without any look ahead symbol in the input string.

**Sol:** The key idea here is that you need not to look at the grammar. Directly construct a PDA (a deterministic one) by looking at the language.

e, e → $

a, $ → $

c, e → e

b, e → b

b, b → e

a, e → e

b, e → e

b, $ → e

b, e → e

a, e → e

iii) Give a graphical representation of a deterministic PDA for accepting L(G). This PDA should work with one look ahead symbol in the input string. Moreover, for any string w ∈ L(G), it is capable of generating the leftmost derivation of the parse tree.

**Sol:**

e, e → $

e, e → S

e, S → aABb

a, a → e

e, A → bAb

b,e → e

e,b → e

c, A → c

e, c → e

b, b → e

e, B → aB

e, a → e

b, B → b

e,b → e

b, b → e

#, $ → e

a,e → e

iv) Using the PDA in (iii) above, generate a top down parsing table for the input string: **abcbabb**. The parsing table has the following columns:

**Sol:** If the above PDA is right then this part is evaluated out of 6M. In case, the PDA is wrong, then this part is evaluated out of 3M only.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S. No.** | **State of PDA** | **Unread Input** | **Stack** | **Rule of G** |
| 1 | R | abcbabb# | - |  |
| 2 | P | abcbabb# | $ |  |
| 3 | q | abcbabb# | S$ |  |
| 4 | q | abcbabb# | aABb$ | **S→ aABb** |
| 5 | q1 | bcbabb# | ABb$ |  |
| 6 | q1b | cbabb# | ABb$ |  |
| 7 | q1b | cbabb# | bAbBb$ | A **→ bAb** |
| 8 | q1 | cbabb# | AbBb$ |  |
| 9 | q1c | babb# | cbBb$ | A **→ c** |
| 10 | q2 | babb# | bBb$ |  |
| 11 | q2 | abb# | Bb$ |  |
| 12 | q2a | bb# | Bb$ |  |
| 13 | q2a | bb# | aBb$ | B **→ aB** |
| 14 | q2 | bb# | Bb$ |  |
| 15 | q2b | b# | bb$ | B **→ b** |
| 16 | q3 | b# | b$ |  |
| 17 | q4 | # | $ |  |
| 18 | qf | - | - |  |
|  |  |  |  |  |
|  |  |  |  |  |

**[3+6+10+6=25M]**

**Note: For a graphical representation of PDA, ((p, a, b), (q, c)) is equivalent to:**

**a , b → c**