Bharatiya Vidya Bhavan's

SARDAR PATEL INSTITUTE OF TECHNOLOGY

(Autonomous Institute Affiliated to University of Mumbai)
Munshi Nagar, Andheri (W), Mumbai – 400 058.

Department of Computer Engineering

ISE 1 – Computer Programming Exam

Date – 12 Feb 2021 **Time – 09:15 AM to 10:45 AM**

AY 2020-21

Class: BTech (Computer Engineering) Semester: VI

Course Name: System Programming and Compiler Construction Course Code: CE61

Total Marks: 20

Note - (1) Attempt all questions.
(2) Assume suitable data wherever **required but justify same.**
(3) Write the details namely your name, roll no, date of examination, course name etc. as comment in the beginning of the programming file.
(4) Take picture of manually created predictive parsing table. Upload this image of predictive parsing table and the programming file on the SPIT moodle under ISE 1 Section of SPCC Course.

Each student with roll no $\boldsymbol{r}$ has to use a *context free grammar* $\boldsymbol{G_r} = (\boldsymbol{V_r}, \boldsymbol{T_r}, \boldsymbol{P_r}, \boldsymbol{S_r})$. Each element of this grammar is defined as follows. The set $\boldsymbol{P_r}$ of four number of productions is all the production with index $\boldsymbol{i}$ in **Table 1** calculated using the function $(\mathbf{r} + \mathbf{2^i})\ \mathbf{mod}\ \mathbf{11}$ where $\boldsymbol{r}$ and $\boldsymbol{i}$ are the roll no of the student and $\boldsymbol{i^{th}}$ production (denoted by$\boldsymbol{P_r^i}$) respectively. The 4th production (denoted by $\boldsymbol{P_r^4}$) is $\boldsymbol{P_r^4} \rightarrow \boldsymbol{P_r^1} \mid \boldsymbol{P_r^2} \mid \boldsymbol{P_r^3}$. The set of non-terminal symbols $\boldsymbol{V_r}$ is all non-terminals $\boldsymbol{P_r^i}$ (for i=1 to 4) by replacing all the right side non-terminal symbols with respective left side non-terminal symbol i.e. $\boldsymbol{P_r^i}$ (for i=1 to 3) of the productions. The set of terminal symbols $\boldsymbol{T_r}$ is all terminals on right side of the productions of $\boldsymbol{P_r^i}$ (for i=1 to 3). The start symbol $\boldsymbol{S_r}$ is $\boldsymbol{P_r^4}$.

---Table 1 ---

Index	Right Side of Production
0	aSa | bSb | a | b | ε
1	ε | SS | bSa | aSb
2	S+S | S*S | a
3	S-S | S/S | b

Index	Right Side of Production
4	a | b
5	aSbb | aSbbb | ε
6	Saab | aSab | aaSb | aabS | a
7	aSb | bSa | SS | ε

Index	Right Side of Production
8	aaSb | ε
9	() | SS | (S)
10	a | aS | bS

For example – the student with roll no. 72 has index set ={8,10,3} using the index formula $(\mathbf{r} + \mathbf{2^i})\ \mathbf{mod}\ \mathbf{11}$. Then, the set of productions, $\boldsymbol{P_{72}} = \{\boldsymbol{P_{72}^1} \rightarrow \boldsymbol{aaSb} \mid \boldsymbol{\varepsilon}\,, \boldsymbol{P_{72}^2} \rightarrow \boldsymbol{a} \mid \boldsymbol{aS} \mid \boldsymbol{bS}, \boldsymbol{P_{72}^3} \rightarrow \boldsymbol{S} - \boldsymbol{S} \mid \boldsymbol{S/S} \mid \boldsymbol{a}, \boldsymbol{P_{72}^4} \rightarrow \boldsymbol{P_{72}^1} \mid \boldsymbol{P_{72}^2} \mid \boldsymbol{P_{72}^3}\}$. Then the right side Non-terminal is replaced by respective left side non-terminal symbol. Then, the revised set of productions, $\boldsymbol{P_{72}} = \{\boldsymbol{P_{72}^1} \rightarrow \boldsymbol{aa}\ \boldsymbol{P_{72}^1 b} \mid \boldsymbol{\varepsilon}\,, \boldsymbol{P_{72}^2} \rightarrow \boldsymbol{a} \mid \boldsymbol{aP_{72}^2} \mid \boldsymbol{bP_{72}^2}, \boldsymbol{P_{72}^3} \rightarrow \boldsymbol{P_{72}^3} - \boldsymbol{P_{72}^3} \mid \boldsymbol{P_{72}^3/P_{72}^3} \mid \boldsymbol{a}, \boldsymbol{P_{72}^4} \rightarrow \boldsymbol{P_{72}^1} \mid \boldsymbol{P_{72}^2} \mid \boldsymbol{P_{72}^3}\}$. The set of non-terminal symbols = $\boldsymbol{V_r} = \{\boldsymbol{P_{72}^1}, \boldsymbol{P_{72}^2}\,, \boldsymbol{P_{72}^3}, \boldsymbol{P_{72}^4}\}$. The set of terminals = $T_r = \{a, b, -, /, \varepsilon\}$ and finally the start symbol is $\boldsymbol{S_{72}}$ is $\boldsymbol{P_{72}^4}$**.**

Perform following:

1) Find the predicting parsing table manually and then store the same using any data structure.

[Marks 10]

*2) Write a menu driven program for Non-recursive Predictive Parsing in C/C++/Java/Python using the grammar $\boldsymbol{G_r} = (\boldsymbol{V_r}, \boldsymbol{T_r}, \boldsymbol{P_r}, \boldsymbol{S_r})$. The menus should be able i) to recognize a string input by the user, ii) to print predictive parsing table and iii) to exit from program. This menu should be available as many times as required.

[Marks 10]

***Note – Marks of part (B) is given only if part (A) is submitted and correct.**

Best of luck….