

COMPILER PROJECT

Name Student	Number ID
Aliah Abdulrhman Alhameed	
Shamaa Khaled Ben Elewa	
Sara Ebrahim Almashharawi	
Neehal Saleh Almuways	
Renad Saud Al Hussain	

Instructor: Dr. Dania Alomar

Section 371

Submission Date: 5 Nov 2022

Table of Contents

Topic	Page
Part 1	3
1.1 Builds an NFA from a given regular expression.....	4
1.2 Converts a giving NFA into a DFA	5
1.3 Builds a DFA from a given regular expression directly.....	6
Part 2	37
2.1 Modify the grammar	39
2.2 Re-write the grammar such that it can be parsed by an LL(1) parser.....	40
2.3 Use the re-written grammar as a base to implement LL(1) parser.....	42
2.4 Test your code with correct and incorrect input code.....	48
References	52

PART 1

1. Part one:

Implement a program to do the following tasks:

1.1 Builds an NFA from a given regular expression

The screenshot shows a Visual Studio Code interface with two open files: `Comiler.java` and `NFAToDFA.java`. The `Comiler.java` file contains Java code for a GUI application, including a constructor that creates and displays a form. The `NFAToDFA.java` file contains code for a regular expression parser and converter.

A modal dialog titled "Compiler Project Part 1" is displayed over the code editor. It has a text input field "Enter RE: (a|b)*abb" and three buttons: "NFA", "NFAToDFA", and "DFA". Below these buttons are two tables: "States" and "Transaction".

The "States" table lists states from state 0 to state 10. The "Transaction" table lists transitions:

From State	Transition	To State
0	E	1
0	E	7
1	E	2
2	a	3
3	E	4
4	b	5
5	E	6
6	E	7
6	E	1
7	a	8
8	b	9
9	b	10

The "Start State" is set to 0 and the "Accept State" is set to 10.

The terminal window shows the command line output of the Java compiler:

```
PS E:\src> & 'C:\Program Files\Eclipse Foundation\jdk-8.0.302.8-hotspot\bin\java.exe' '-cp' 'C:\Users\USER\AppData\Roaming\Code\User\workspaceStorage\b2114a5611a7bc1adffdb5f1b29 96ad7\redhat\java\jdt_ws\src_51987c3e\bin' 'GUT.Comiller'
new RE is: (a|b)*.a.b.b
the postfix form ab|*a.b.b.
```

The status bar at the bottom right indicates: In 259, Col 1 | Spaces: 4 | UTF-8 | LF | Java | Run: Comiller.

1.2 Converts a giving NFA into a DFA

The screenshot shows a Visual Studio Code interface with two main panes. The left pane displays the Java code for a compiler, specifically the part related to NFA to DFA conversion. The right pane shows a graphical user interface titled "Compiler Project Part 1" for performing the conversion.

Code (Left Pane):

```
File Edit Selection View Go Run Terminal Help
GUI > Comilerjava 1 x J NFAtoDFA.java
248     java.util.logging.Logger.getLogger(Comiler.class.getName()).log(ja
249 }
250 //</editor-fold>
251
252 /* Create and display the form */
253 java.awt.EventQueue.invokeLater(new Runnable() {
254     public void run() {
255         new Comiler().setVisible(b: true);
256     }
257 });
258
259 // Variables declaration - do not modify//GEN-BEGIN:variables
260 private javax.swing.JButton DFAButton;
261 private javax.swing.JButton NFAButton;
262 private javax.swing.JButton NFAtoDFAButton;
263 private javax.swing.JTextField REText;
264 private javax.swing.JTextPane acceptState;
265 private javax.swing.JLabel jLabel1;
266 private javax.swing.JLabel jLabel2;
267 private javax.swing.JLabel jLabel3;
```

Graphical Interface (Right Pane):

The graphical interface has three tabs at the top: "NFA" (highlighted in yellow), "NFAtoDFA" (selected), and "DFA".

- Enter RE:** A text input field containing the regular expression `(abb)*abb`.
- States:** A list of states for the DFA:
 - state A
 - state B
 - state C
 - state D
 - state E
- Transaction:** A list of transitions for the DFA:
 - A->3->B
 - A->0->C
 - B->3->B
 - B->0->D
 - C->3->B
 - C->0->C
 - D->3->B
 - D->0->E
 - E->0->B
 - E->0->C
- Start State:** A box labeled "A".
- Accept State:** A box labeled "E".

At the bottom of the interface, there is a "Run: Comiler" button.

Bottom Status Bar:

Activate Windows
Go to Settings to activate Windows.
Ln 259, Col 1 · Spaces: 4 · UTF-8 · LF · Java

1.3 Builds a DFA from a given regular expression directly

The screenshot shows the Visual Studio Code interface with the 'CompilerJava' project open. A pop-up window titled 'Compiler Project Part 1' is displayed, showing the conversion of the regular expression $(a|b)^abb$ into a DFA. The window includes fields for entering the RE, buttons for 'NFA' and 'DFA', and sections for 'States', 'Transaction', 'Start State', and 'Accept State'. The 'States' section lists state A, B, C, and D. The 'Transaction' section shows transitions: A $\xrightarrow{a} B$, A $\xrightarrow{b} A$, B $\xrightarrow{a} B$, B $\xrightarrow{b} C$, C $\xrightarrow{a} B$, C $\xrightarrow{b} D$, D $\xrightarrow{a} B$, and D $\xrightarrow{b} A$. The 'Start State' is A and the 'Accept State' is D. Below the pop-up, the terminal output shows the generated DFA details:

```
[1, 2, 3] ---- b ----> [1, 2, 3]
[1, 2, 3, 4] ---- a ----> [1, 2, 3, 4]
[1, 2, 3, 4] ---- b ----> [1, 2, 3, 5]
[1, 2, 3, 5] ---- a ----> [1, 2, 3, 4]
[1, 2, 3, 5] ---- b ----> [1, 2, 3, 6]
[1, 2, 3, 6] ---- a ----> [1, 2, 3, 4]
[1, 2, 3, 6] ---- b ----> [1, 2, 3]
Initial state:[1, 2, 3]
final state:[1, 2, 3, 6]
DFA states: [A, B, C, D]
initial state is :A
final states are :[D]
```

Activate Windows
Go to Settings to activate Windows.

In 259, Col 1 Spaces: 4 UTF-8 LF Java

Code NFA and NFA to DFA:

GUI:

The screenshot shows a Visual Studio Code interface with two tabs open: 'Comiler.java 1' and 'NFAToDFA.java'. The 'Comiler.java 1' tab is active, displaying Java code for a GUI application. The code includes imports for javax.swing, java.awt, and other Java packages. It defines a class 'Comiler' that extends 'JFrame'. The constructor initializes components like labels, text fields, and buttons. A comment indicates that the code was generated by the Form Editor. The right side of the screen shows a vertical sidebar with various icons and a preview area.

```
1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3   * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template
4   */
5  package GUI;
6
7
8  import DFADir.*;
9
10 import NFAandDFA.*;
11 public class Comiler extends javax.swing.JFrame {
12
13     NFA nfamain;
14
15     public Comiler() {
16         initComponents();
17     }
18
19     /**
20      * This method is called from within the constructor to initialize the form.
21      * WARNING: Do NOT modify this code. The content of this method is always
22      * regenerated by the Form Editor.
23      */
24     @SuppressWarnings("unchecked")
25 // <editor-fold defaultstate="collapsed" desc="Generated Code">/GEN-BEGIN:initComponents
26     private void initComponents() {
27
28         jLabel1 = new javax.swing.JLabel();
29         jLabel2 = new javax.swing.JLabel();
30         RText = new javax.swing.JTextField();
31         NFAButton = new javax.swing.JButton();
32         DFAButton = new javax.swing.JButton();
33         jLabel3 = new javax.swing.JLabel();
34         jScrollPane1 = new javax.swing.JScrollPane();
35         resultTrans = new javax.swing.JTextPane();
36
37         setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
38
39         jLabel1.setText("NFA");
40         jLabel2.setText("DFA");
41
42         NFAButton.setText("NFA");
43         DFAButton.setText("DFA");
44
45         jScrollPane1.setViewportView(resultTrans);
46
47         javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
48         getContentPane().setLayout(layout);
49         layout.setHorizontalGroup(
50             layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
51             .addGroup(layout.createSequentialGroup()
52                 .addComponent(jLabel1)
53                 .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
54                 .addComponent(jLabel2)
55                 .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
56                 .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 375, javax.swing.GroupLayout.PREFERRED_SIZE)
57                 .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
58                 .addComponent(NFAButton)
59                 .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
60                 .addComponent(DFAButton)
61                 .addContainerGap())
62         );
63         layout.setVerticalGroup(
64             layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
65             .addGroup(layout.createSequentialGroup()
66                 .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
67                     .addComponent(jLabel1)
68                     .addComponent(jLabel2)
69                     .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 375, javax.swing.GroupLayout.PREFERRED_SIZE)
70                     .addComponent(NFAButton)
71                     .addComponent(DFAButton))
72                 .addContainerGap())
73         );
74
75         pack();
76     }// </editor-fold>
77
78     // Variables declaration - do not modify//GEN-BEGIN:variables
79     private javax.swing.JButton DFAButton;
80     private javax.swing.JButton NFAButton;
81     private javax.swing.JTextField RText;
82     private javax.swing.JLabel jLabel1;
83     private javax.swing.JLabel jLabel2;
84     private javax.swing.JScrollPane jScrollPane1;
85     private javax.swing.JTextPane resultTrans;
86     // End of variables declaration//GEN-END:variables
87
88     /**
89      * @param args the command line arguments
90      */
91     public static void main(String args[]) {
92         /* Set the Nimbus look and feel */
93         //
94         /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
95          * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
96          */
97         try {
98             for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
99                 if ("Nimbus".equals(info.getName())) {
100                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
101                    break;
102                }
103            }
104        } catch (ClassNotFoundException ex) {
105            java.util.logging.Logger.getLogger(Comiler.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
106        } catch (InstantiationException ex) {
107            java.util.logging.Logger.getLogger(Comiler.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
108        } catch (IllegalAccessException ex) {
109            java.util.logging.Logger.getLogger(Comiler.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
110        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
111            java.util.logging.Logger.getLogger(Comiler.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
112        }
113
114        //
115        //
116        //
117        //
118        //
119        //
120        //
121        //
122        //
123        //
124        //
125        //
126        //
127        //
128        //
129        //
130        //
131        //
132        //
133        //
134        //
135        //
136        //
137        //
138        //
139        //
140        //
141        //
142        //
143        //
144        //
145        //
146        //
147        //
148        //
149        //
150        //
151        //
152        //
153        //
154        //
155        //
156        //
157        //
158        //
159        //
160        //
161        //
162        //
163        //
164        //
165        //
166        //
167        //
168        //
169        //
170        //
171        //
172        //
173        //
174        //
175        //
176        //
177        //
178        //
179        //
180        //
181        //
182        //
183        //
184        //
185        //
186        //
187        //
188        //
189        //
190        //
191        //
192        //
193        //
194        //
195        //
196        //
197        //
198        //
199        //
200        //
201        //
202        //
203        //
204        //
205        //
206        //
207        //
208        //
209        //
210        //
211        //
212        //
213        //
214        //
215        //
216        //
217        //
218        //
219        //
220        //
221        //
222        //
223        //
224        //
225        //
226        //
227        //
228        //
229        //
230        //
231        //
232        //
233        //
234        //
235        //
236        //
237        //
238        //
239        //
240        //
241        //
242        //
243        //
244        //
245        //
246        //
247        //
248        //
249        //
250        //
251        //
252        //
253        //
254        //
255        //
256        //
257        //
258        //
259        //
260        //
261        //
262        //
263        //
264        //
265        //
266        //
267        //
268        //
269        //
270        //
271        //
272        //
273        //
274        //
275        //
276        //
277        //
278        //
279        //
280        //
281        //
282        //
283        //
284        //
285        //
286        //
287        //
288        //
289        //
290        //
291        //
292        //
293        //
294        //
295        //
296        //
297        //
298        //
299        //
300        //
301        //
302        //
303        //
304        //
305        //
306        //
307        //
308        //
309        //
310        //
311        //
312        //
313        //
314        //
315        //
316        //
317        //
318        //
319        //
320        //
321        //
322        //
323        //
324        //
325        //
326        //
327        //
328        //
329        //
330        //
331        //
332        //
333        //
334        //
335        //
336        //
337        //
338        //
339        //
340        //
341        //
342        //
343        //
344        //
345        //
346        //
347        //
348        //
349        //
350        //
351        //
352        //
353        //
354        //
355        //
356        //
357        //
358        //
359        //
360        //
361        //
362        //
363        //
364        //
365        //
366        //
367        //
368        //
369        //
370        //
371        //
372        //
373        //
374        //
375        //
376        //
377        //
378        //
379        //
380        //
381        //
382        //
383        //
384        //
385        //
386        //
387        //
388        //
389        //
390        //
391        //
392        //
393        //
394        //
395        //
396        //
397        //
398        //
399        //
400        //
401        //
402        //
403        //
404        //
405        //
406        //
407        //
408        //
409        //
410        //
411        //
412        //
413        //
414        //
415        //
416        //
417        //
418        //
419        //
420        //
421        //
422        //
423        //
424        //
425        //
426        //
427        //
428        //
429        //
430        //
431        //
432        //
433        //
434        //
435        //
436        //
437        //
438        //
439        //
440        //
441        //
442        //
443        //
444        //
445        //
446        //
447        //
448        //
449        //
450        //
451        //
452        //
453        //
454        //
455        //
456        //
457        //
458        //
459        //
460        //
461        //
462        //
463        //
464        //
465        //
466        //
467        //
468        //
469        //
470        //
471        //
472        //
473        //
474        //
475        //
476        //
477        //
478        //
479        //
480        //
481        //
482        //
483        //
484        //
485        //
486        //
487        //
488        //
489        //
490        //
491        //
492        //
493        //
494        //
495        //
496        //
497        //
498        //
499        //
500        //
501        //
502        //
503        //
504        //
505        //
506        //
507        //
508        //
509        //
510        //
511        //
512        //
513        //
514        //
515        //
516        //
517        //
518        //
519        //
520        //
521        //
522        //
523        //
524        //
525        //
526        //
527        //
528        //
529        //
530        //
531        //
532        //
533        //
534        //
535        //
536        //
537        //
538        //
539        //
540        //
541        //
542        //
543        //
544        //
545        //
546        //
547        //
548        //
549        //
550        //
551        //
552        //
553        //
554        //
555        //
556        //
557        //
558        //
559        //
560        //
561        //
562        //
563        //
564        //
565        //
566        //
567        //
568        //
569        //
570        //
571        //
572        //
573        //
574        //
575        //
576        //
577        //
578        //
579        //
580        //
581        //
582        //
583        //
584        //
585        //
586        //
587        //
588        //
589        //
590        //
591        //
592        //
593        //
594        //
595        //
596        //
597        //
598        //
599        //
600        //
601        //
602        //
603        //
604        //
605        //
606        //
607        //
608        //
609        //
610        //
611        //
612        //
613        //
614        //
615        //
616        //
617        //
618        //
619        //
620        //
621        //
622        //
623        //
624        //
625        //
626        //
627        //
628        //
629        //
630        //
631        //
632        //
633        //
634        //
635        //
636        //
637        //
638        //
639        //
640        //
641        //
642        //
643        //
644        //
645        //
646        //
647        //
648        //
649        //
650        //
651        //
652        //
653        //
654        //
655        //
656        //
657        //
658        //
659        //
660        //
661        //
662        //
663        //
664        //
665        //
666        //
667        //
668        //
669        //
670        //
671        //
672        //
673        //
674        //
675        //
676        //
677        //
678        //
679        //
680        //
681        //
682        //
683        //
684        //
685        //
686        //
687        //
688        //
689        //
690        //
691        //
692        //
693        //
694        //
695        //
696        //
697        //
698        //
699        //
700        //
701        //
702        //
703        //
704        //
705        //
706        //
707        //
708        //
709        //
710        //
711        //
712        //
713        //
714        //
715        //
716        //
717        //
718        //
719        //
720        //
721        //
722        //
723        //
724        //
725        //
726        //
727        //
728        //
729        //
730        //
731        //
732        //
733        //
734        //
735        //
736        //
737        //
738        //
739        //
740        //
741        //
742        //
743        //
744        //
745        //
746        //
747        //
748        //
749        //
750        //
751        //
752        //
753        //
754        //
755        //
756        //
757        //
758        //
759        //
760        //
761        //
762        //
763        //
764        //
765        //
766        //
767        //
768        //
769        //
770        //
771        //
772        //
773        //
774        //
775        //
776        //
777        //
778        //
779        //
780        //
781        //
782        //
783        //
784        //
785        //
786        //
787        //
788        //
789        //
790        //
791        //
792        //
793        //
794        //
795        //
796        //
797        //
798        //
799        //
800        //
801        //
802        //
803        //
804        //
805        //
806        //
807        //
808        //
809        //
810        //
811        //
812        //
813        //
814        //
815        //
816        //
817        //
818        //
819        //
820        //
821        //
822        //
823        //
824        //
825        //
826        //
827        //
828        //
829        //
830        //
831        //
832        //
833        //
834        //
835        //
836        //
837        //
838        //
839        //
840        //
841        //
842        //
843        //
844        //
845        //
846        //
847        //
848        //
849        //
850        //
851        //
852        //
853        //
854        //
855        //
856        //
857        //
858        //
859        //
860        //
861        //
862        //
863        //
864        //
865        //
866        //
867        //
868        //
869        //
870        //
871        //
872        //
873        //
874        //
875        //
876        //
877        //
878        //
879        //
880        //
881        //
882        //
883        //
884        //
885        //
886        //
887        //
888        //
889        //
890        //
891        //
892        //
893        //
894        //
895        //
896        //
897        //
898        //
899        //
900        //
901        //
902        //
903        //
904        //
905        //
906        //
907        //
908        //
909        //
910        //
911        //
912        //
913        //
914        //
915        //
916        //
917        //
918        //
919        //
920        //
921        //
922        //
923        //
924        //
925        //
926        //
927        //
928        //
929        //
930        //
931        //
932        //
933        //
934        //
935        //
936        //
937        //
938        //
939        //
940        //
941        //
942        //
943        //
944        //
945        //
946        //
947        //
948        //
949        //
950        //
951        //
952        //
953        //
954        //
955        //
956        //
957        //
958        //
959        //
960        //
961        //
962        //
963        //
964        //
965        //
966        //
967        //
968        //
969        //
970        //
971        //
972        //
973        //
974        //
975        //
976        //
977        //
978        //
979        //
980        //
981        //
982        //
983        //
984        //
985        //
986        //
987        //
988        //
989        //
990        //
991        //
992        //
993        //
994        //
995        //
996        //
997        //
998        //
999        //
1000    //
```

File Edit Selection View Go Run Terminal Help Compiler.java - src - Visual Studio Code

J Comiler.java 1 X J NFAtoDFA.java

GUI > J Comiler.java > \$ Comiler > initComponents()

```

69     public void actionPerformed(java.awt.event.ActionEvent evt) {
70         DFAButtonActionPerformed(evt);
71     }
72 }
73
74 jLabel3.setFont(new java.awt.Font(name: "Segoe UI", style: 0, size: 18)); // NOI18N
75 jLabel3.setText(text: "Compiler Project Part 1");
76
77 jScrollPane1.setViewportView(resultTrans);
78
79 jScrollPane2.setViewportView(resultState);
80
81 jLabel4.setText(text: "States");
82
83 jLabel5.setText(text: "Transaction");
84
85 NFAtoDFAButton.setText(text: "NFAtoDFA");
86 NFAtoDFAButton.addActionListener(new java.awt.event.ActionListener() {
87     public void actionPerformed(java.awt.event.ActionEvent evt) {
88         NFAtoDFAButtonActionPerformed(evt);
89     }
90 });
91
92 jLabel6.setText(text: "Start State");
93
94 jScrollPane3.setViewportView(startState);
95
96 jLabel7.setText(text: "Accept State");
97
98 jScrollPane4.setViewportView(acceptState);
99
100 javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
101 getContentPane().setLayout(layout);
102 layout.setHorizontalGroup(
103     layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

```

Activate Windows
Go to Settings to activate Windows.

Ln 103, Col 51 Spaces: 4 UTF-8 LF { Java

File Edit Selection View Go Run Terminal Help Compiler.java - src - Visual Studio Code

J Comiler.java 1 X J NFAtoDFA.java

GUI > J Comiler.java > \$ Comiler > initComponents()

```

103     layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
104         .addGroup(layout.createSequentialGroup())
105             .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
106                 .addGroup(layout.createSequentialGroup())
107                     .addComponent(jLabel1)
108                     .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup())
109                         .addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED_SIZE, pref: 156, javax.swing.GroupLayout.PREFERRED_SIZE)
110                         .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
111                             .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, pref: 147, javax.swing.GroupLayout.PREFERRED_SIZE))
112                         .addComponent(jLabel2)
113                         .addComponent(jLabel3)
114                         .addComponent(jLabel4)
115                         .addComponent(jLabel5)
116                         .addComponent(jLabel6)
117                         .addComponent(jLabel7)
118                         .addGroup(layout.createSequentialGroup())
119                             .addComponent(jLabel8)
120                             .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, resizable: false)
121                                 .addComponent(jScrollPane3)
122                                 .addComponent(jScrollPane4)
123                                 .addComponent(NFAButton, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
124                                 .addComponent(NFAtoDFAButton, javax.swing.GroupLayout.DEFAULT_SIZE, pref: 114, Short.MAX_VALUE)
125                                 .addComponent(DFAButton, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
126                             .addGroup(layout.createSequentialGroup())
127                             .addComponent(jLabel9)
128                             .addGroup(layout.createSequentialGroup())
129                                 .addComponent(jLabel10)
130                                 .addComponent(jLabel11)
131                                 .addGroup(layout.createSequentialGroup())
132                                     .addComponent(jLabel12)
133                                     .addComponent(jLabel13)
134                                     .addComponent(jLabel14)
135                                     .addComponent(jLabel15)
136                                     .addComponent(jLabel16)

```

Activate Windows
Go to Settings to activate Windows.

Ln 103, Col 82 Spaces: 4 UTF-8 LF { Java

File Edit Selection View Go Run Terminal Help Comiler.java - src - Visual Studio Code

J Comiler.java 1 X J NFAToDFA.java

GUI > J Comiler.java > Comiler > initComponents()

```

137     .addComponent(jLabel16)
138     .addGap(min: 76, pref: 76, max: 76)
139     .addGroup(layout.createSequentialGroup()
140       .addGap(min: 170, pref: 170, max: 170)
141       .addComponent(jLabel13)
142       .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
143   );
144   layout.setVerticalGroup(
145     layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
146     .addGroup(layout.createSequentialGroup()
147       .addGap(min: 44, pref: 44, max: 44)
148       .addComponent(jLabel13)
149       .addGap(min: 51, pref: 51, max: 51)
150       .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
151         .addGroup(layout.createSequentialGroup()
152           .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
153             .addComponent(RETText, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
154             .addComponent(jLabel12)
155             .addComponent(jLabel11)
156             .addComponent(NFAButton)
157             .addComponent(NFAToDFAButton))
158           .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
159           .addComponent(jLabel10)
160           .addComponent(jLabel9)
161           .addComponent(jLabel8)
162           .addComponent(DFAButton)
163           .addGap(min: 12, pref: 12, max: 12)
164           .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
165             .addComponent(jLabel14)
166             .addComponent(jLabel15)
167             .addComponent(jLabel16)
168             .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
169             .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
170               .addComponent(jLabel17)
171               .addComponent(jLabel18)
172               .addComponent(jLabel19)
173               .addComponent(jLabel20)
174               .addComponent(jLabel21)
175               .addComponent(jLabel22)
176               .addComponent(jLabel23)
177               .addComponent(jLabel24)
178               .addComponent(jLabel25)
179               .addComponent(jLabel26)
180             )
181             .pack();
182           )// </editor-fold> //GEN-END:initComponents
183         private void RETTextActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_RETTextActionPerformed
184           // TODO add your handling code here:
185         } //GEN-LAST:event_RETTextActionPerformed
186
187         private void NFAButtonActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_NFAButtonActionPerformed
188
189           NFA nfa = new NFA();
190           nfa.main=nfa.compile(RETText.getText());
191           resultState.setText(nfa.printNFAState(nfamain));
192           resultTrans.setText(nfa.printNFATrans(nfamain));
193           startState.setText(nfa.printStartState(nfamain));
194           acceptState.setText(nfa.printFinalState(nfamain));
195         } //GEN-LAST:event_NFAButtonActionPerformed
196
197         private void NFAToDFAButtonActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_NFAToDFAButtonActionPerformed
198           NFAToDFA dfa=new NFAToDFA(nfamain);
199           dfa.compile();
200           DFA dfal=new DFA(dfa);
201           resultState.setText(dfal.printDFAState());
202           resultTrans.setText(dfal.printTrans(dfa));
203           startState.setText(dfal.printStartState());
204         } //GEN-LAST:event_NFAToDFAButtonActionPerformed
205

```

Activate Windows
Go to Settings to activate Windows.

In 137, Col 39 Spaces: 4 UTF-8 LF Java

File Edit Selection View Go Run Terminal Help Comiler.java - src - Visual Studio Code

J Comiler.java 1 X J NFAToDFA.java

GUI > J Comiler.java > Comiler > initComponents()

```

171     .addComponent(jLabelScrollPane3, javax.swing.GroupLayout.PREFERRED_SIZE, pref: 111, javax.swing.GroupLayout.PREFERRED_SIZE)
172     .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
173     .addComponent(jLabel17)
174     .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
175     .addComponent(jLabelScrollPane4)
176     .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
177       .addComponent(jLabelScrollPane2, javax.swing.GroupLayout.PREFERRED_SIZE, pref: 246, javax.swing.GroupLayout.PREFERRED_SIZE)
178       .addComponent(jLabelScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, pref: 246, javax.swing.GroupLayout.PREFERRED_SIZE))
179     .addGap(min: 0, pref: 30, Short.MAX_VALUE)
180   );
181
182   pack();
183 } // </editor-fold> //GEN-END:initComponents
184
185 private void RETTextActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_RETTextActionPerformed
186   // TODO add your handling code here:
187 } //GEN-LAST:event_RETTextActionPerformed
188
189 private void NFAButtonActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_NFAButtonActionPerformed
190
191   NFA nfa = new NFA();
192   nfa.main=nfa.compile(RETText.getText());
193   resultState.setText(nfa.printNFAState(nfamain));
194   resultTrans.setText(nfa.printNFATrans(nfamain));
195   startState.setText(nfa.printStartState(nfamain));
196   acceptState.setText(nfa.printFinalState(nfamain));
197 } //GEN-LAST:event_NFAButtonActionPerformed
198
199 private void NFAToDFAButtonActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_NFAToDFAButtonActionPerformed
200   NFAToDFA dfa=new NFAToDFA(nfamain);
201   dfa.compile();
202   DFA dfal=new DFA(dfa);
203   resultState.setText(dfal.printDFAState());
204   resultTrans.setText(dfal.printTrans(dfa));
205   startState.setText(dfal.printStartState());
206

```

Activate Windows
Go to Settings to activate Windows.

In 137, Col 39 Spaces: 4 UTF-8 LF Java

File Edit Selection View Go Run Terminal Help Comiler.java - compiler5 - Visual Studio Code

```
J Comiler.java 2 x
src > GUI > J Comiler.java > D Comiler > D DFAButtonActionPerformed(ActionEvent)
205     startState.setText(dfa1.printStartState());
206     acceptState.setText(dfa1.printFinalState());
207
208 } //GEN-LAST:event_NFAtoDFAButtonActionPerformed
209
210 private void DFAButtonActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_DFAButtonActionPerformed
211
212     BinaryTree r= new BinaryTree();
213     String re=RText.getText()+"#";
214     r.getSymbols(re);
215     SyntaxTree st = new SyntaxTree(re);
216     Node root= st.getRoot();
217     RToDFA df=new RToDFA(st);
218     df.compile();
219     DFADir dfa1=new DFADir(df);
220     resultState.setText(dfa1.printDFAstate());
221     resultTrans.setText(dfa1.printTrans(df));
222     startState.setText(dfa1.printStartState());
223     acceptState.setText(dfa1.printFinalState());
224 } //GEN-LAST:event_DFAButtonActionPerformed
225
226 /**
227 * @param args the command line arguments
228 */
229 Run|Debug
230 public static void main(String args[]) {
231     /* Set the Nimbus look and feel */
232     //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
233     /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
234     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
235     */
236     try {
237         for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
238             if ("Nimbus".equals(info.getName())) {
239                 javax.swing.UIManager.setLookAndFeel(info.getClassName());
240             }
241         }
242     } catch (ClassNotFoundException ex) {
243         java.util.logging.Logger.getLogger(Comiler.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
244     } catch (InstantiationException ex) {
245         java.util.logging.Logger.getLogger(Comiler.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
246     } catch (IllegalAccessException ex) {
247         java.util.logging.Logger.getLogger(Comiler.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
248     } catch (javax.swing.UnsupportedLookAndFeelException ex) {
249         java.util.logging.Logger.getLogger(Comiler.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
250     }
251 //</editor-fold>
252
253     /* Create and display the form */
254     java.awt.EventQueue.invokeLater(new Runnable() {
255         public void run() {
256             new Comiler().setVisible(true);
257         }
258     });
259 }
260
261 // Variables declaration - do not modify//GEN-BEGIN:variables
262 private javax.swing.JButton DFAButton;
263 private javax.swing.JButton NFAButton;
264 private javax.swing.JButton NFAtoDFAButton;
265 private javax.swing.JTextField RText;
266 private javax.swing.JTextPane acceptState;
267 private javax.swing.JLabel jLabel1;
268 private javax.swing.JLabel jLabel2;
269 private javax.swing.JLabel jLabel3;
270 private javax.swing.JLabel jLabel4;
271 private javax.swing.JLabel jLabel5;
272 private javax.swing.JLabel jLabel6;
273 private javax.swing.JLabel jLabel7;
```

Activate Windows
Go to Settings to activate Windows.

Ln 218, Col 22 Spaces: 4 UTF-8 LF {} Java ⚙️

File Edit Selection View Go Run Terminal Help Comiler.java - compiler5 - Visual Studio Code

```
J Comiler.java 2 x
src > GUI > J Comiler.java > D Comiler > D main(String[])
239         break;
240     }
241 }
242 } catch (ClassNotFoundException ex) {
243     java.util.logging.Logger.getLogger(Comiler.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
244 } catch (InstantiationException ex) {
245     java.util.logging.Logger.getLogger(Comiler.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
246 } catch (IllegalAccessException ex) {
247     java.util.logging.Logger.getLogger(Comiler.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
248 } catch (javax.swing.UnsupportedLookAndFeelException ex) {
249     java.util.logging.Logger.getLogger(Comiler.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
250 }
251 //</editor-fold>
252
253 /* Create and display the form */
254 java.awt.EventQueue.invokeLater(new Runnable() {
255     public void run() {
256         new Comiler().setVisible(true);
257     }
258 });
259 }
260
261 // Variables declaration - do not modify//GEN-BEGIN:variables
262 private javax.swing.JButton DFAButton;
263 private javax.swing.JButton NFAButton;
264 private javax.swing.JButton NFAtoDFAButton;
265 private javax.swing.JTextField RText;
266 private javax.swing.JTextPane acceptState;
267 private javax.swing.JLabel jLabel1;
268 private javax.swing.JLabel jLabel2;
269 private javax.swing.JLabel jLabel3;
270 private javax.swing.JLabel jLabel4;
271 private javax.swing.JLabel jLabel5;
272 private javax.swing.JLabel jLabel6;
273 private javax.swing.JLabel jLabel7;
```

Activate Windows
Go to Settings to activate Windows.

Ln 227, Col 34 Spaces: 4 UTF-8 LF {} Java ⚙️

```

file Edit Selection View Go Run Terminal Help
Comiler.java - src - Visual Studio Code
J Comiler.java < J NFAToDFA.java
GUI > J Comiler.java > Comiler
248     java.util.logging.Logger.getLogger(Comiler.class.getName()).log(java.util.logging.Level.SEVERE, msg: null, ex);
249 }
250 //</editor-fold>
251
252 /* Create and display the form */
253 java.awt.EventQueue.invokeLater(new Runnable() {
254     public void run() {
255         new Comiler().setVisible(b: true);
256     }
257 });
258 }
259
// Variables declaration - do not modify//GEN-BEGIN:variables
260 private javax.swing.JButton DFAButton;
261 private javax.swing.JButton NFAButton;
262 private javax.swing.JButton NFAToDFAButton;
263 private javax.swing.JTextField REText;
264 private javax.swing.JTextPane acceptState;
265 private javax.swing.JLabel jLabel1;
266 private javax.swing.JLabel jLabel2;
267 private javax.swing.JLabel jLabel3;
268 private javax.swing.JLabel jLabel4;
269 private javax.swing.JLabel jLabel5;
270 private javax.swing.JLabel jLabel6;
271 private javax.swing.JLabel jLabel7;
272 private javax.swing.JScrollPane jScrollPane1;
273 private javax.swing.JScrollPane jScrollPane2;
274 private javax.swing.JScrollPane jScrollPane3;
275 private javax.swing.JScrollPane jScrollPane4;
276 private javax.swing.JScrollPane jScrollPane5;
277 private javax.swing.JScrollPane jScrollPane6;
278 private javax.swing.JTextPane resultState;
279 private javax.swing.JTextPane resultTrans;
280 private javax.swing.JTextPane startState;
281 // End of variables declaration//GEN-END:variables
282 }
283

```

Activate Windows
Go to Settings to activate Windows.

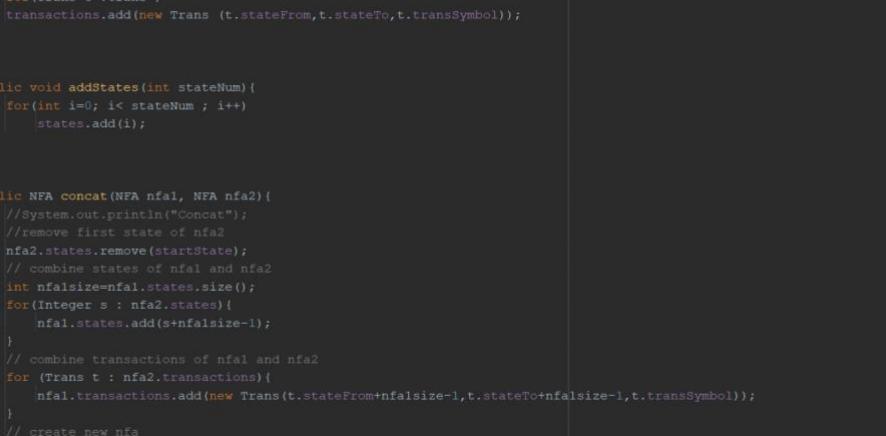
Ln 259, Col 1 Spaces: 4 UTF-8 LF () Java 🔍 🔒

- Class NFA:

```

Source History
5 package NFAandDFA;
6 import java.util.Stack;
7 import java.util.ArrayList;
8 import java.util.Scanner;
9 import java.util.Collections;
10
11 public class NFA {
12     ArrayList<Integer> states;
13     ArrayList<Trans> transactions;
14     int startState, finalState;
15
16     public NFA() {
17         states = new ArrayList <Integer> ();
18         transactions = new ArrayList <Trans> ();
19         startState = 0;
20         finalState = 0;
21     }
22     // if alpha construct nfa
23     public NFA(char symbol) {
24         states=new ArrayList<Integer>();
25         transactions= new ArrayList<Trans>();
26         addStates(2);
27         startState=0;
28         finalState=1;
29         transactions.add(new Trans (0,1,symbol));
30
31     }
32     // used in concatenation and alternation methods
33     public NFA(ArrayList<Integer> st, ArrayList<Trans> trans){
34         states=new ArrayList<Integer>();
35         transactions= new ArrayList<Trans>();
36         for(Integer s : st )

```



The screenshot shows a Java code editor with the following code:

```
Source History [File] [Edit] [Search] [View] [Tools] [Help] [Exit]
37     states.add(s);
38     startState=0;
39     finalState=states.size()-1;
40     for(Trans t :trans )
41         transactions.add(new Trans (t.stateFrom,t.stateTo,t.transSymbol));
42     }
43 }
44
45 public void addStates(int stateNum){
46     for(int i=0; i< stateNum ; i++)
47         states.add(i);
48 }
49
50
51 public NFA concat(NFA nfal, NFA nfa2){
52     //System.out.println("Concat");
53     //remove first state of nfa2
54     nfa2.states.remove(startState);
55     // combine states of nfal and nfa2
56     int nfalsize=nfal.states.size();
57     for(Integer s : nfa2.states){
58         nfal.states.add(s+nfalsize-1);
59     }
60     // combine transactions of nfal and nfa2
61     for (Trans t : nfa2.transactions){
62         nfal.transactions.add(new Trans(t.stateFrom+nfalsize-1,t.stateTo+nfalsize-1,t.transSymbol));
63     }
64     // create new nfa
65     NFA newNFA=new NFA(nfal.states,nfal.transactions );
66     //printNFA(newNFA);
67     return newNFA;
68 }
```

```
69
70
71     public NFA star(NFA nfa){
72         //System.out.println("Star");
73         // main states and transaction to move the new satates and transactions in
74         ArrayList<Integer> newStates= new ArrayList<Integer>();
75         ArrayList<Trans> newTrans=new ArrayList<Trans>();
76         // add first state
77         newStates.add(0);
78         // s0--E-->s1
79         newTrans.add(new Trans(0,nfa.startState+1,'E'));
80         //s0--E-->last (( for empty case ))
81         newTrans.add(new Trans(0,nfa.states.size()+1,'E'));
82
83         //move the original states and trans to the new arrays
84         for(Integer s : nfa.states)
85             newStates.add(s+1);
86
87         for(Trans t : nfa.transactions)
88             newTrans.add(new Trans (t.stateFrom+1,t.stateTo+1,t.transSymbol));
89
90         // add final state
91         newStates.add(newStates.size());
92         // sn-1--E--> sn (sn is last state)
93         newTrans.add(new Trans(newStates.size()-2,newStates.size()-1,'E'));
94         // trans for the loop with E
95         newTrans.add(new Trans(newStates.size()-2,newStates.size()-nfa.states.size()-1,'E'));
96
97
98         NFA newNFA=new NFA(newStates,newTrans );
99         //printNFA(newNFA);
100        return newNFA;
```

```

101     }
102
103     public NFA alternate(NFA nfa1, NFA nfa2) {
104         //System.out.println("OR");
105         // main states and transaction to move the new states and transactions in
106         ArrayList<Integer> newStates= new ArrayList<Integer>();
107         ArrayList<Trans> newTrans=new ArrayList<Trans>();
108         // s0 and its transaction for first branch (nfa1)
109         newStates.add(0);
110         newTrans.add(new Trans(0,nfa1.startState+1,'E'));
111
112         for(Integer s : nfa1.states)
113             newStates.add(s+1);
114
115         for(Trans t : nfa1.transactions)
116             newTrans.add(new Trans (t.stateFrom+1,t.stateTo+1,t.transSymbol));
117
118         // s0 and its transaction for second branch (nfa2)
119         newTrans.add(new Trans(0,newStates.size(),'E'));
120
121         int size = newStates.size()-1;
122         for(Integer s : nfa2.states)
123             newStates.add(s+size+1);
124
125         for(Trans t : nfa2.transactions)
126             newTrans.add(new Trans (t.stateFrom+size+1,t.stateTo+size+1,t.transSymbol));
127         // add last state
128         newStates.add(newStates.size());
129         // combine the 2 branches with the final state
130         newTrans.add(new Trans(nfa1.states.size(),newStates.size()-1,'E'));
131         newTrans.add(new Trans(newStates.size()-2,newStates.size()-1,'E'));
132
133     NFA newNFA=new NFA(newStates,newTrans );
134     return newNFA;
135 }
136
137     public String printNFATrans(NFA newNFA){
138
139         String s="Transaction for the NFA :\n";
140         //System.out.println("Transaction for the NFA : ");
141         for(Trans d: newNFA.transactions)
142             s+= d.print()+"\n";
143
144         Collections.sort(newNFA.transactions);
145         return s;
146     }
147
148     public String printNFAState(NFA newNFA){
149         String s="States for the NFA :\n ";
150         for(Integer d: newNFA.states)
151             s+="state "+d+"\n ";
152         return s;
153     }
154
155     public String printStartState(NFA newNFA){
156
157         return Integer.toString(newNFA.startState);
158     }
159
160     public String printFinalState(NFA newNFA){
161
162         return Integer.toString(newNFA.finalState);
163     }

```

```

165     public NFA compile(String re){
166         Stack<NFA> stack = new Stack<NFA>();
167         //Scanner input = new Scanner(System.in);
168         //System.out.println("Enter RE note :\n"
169         //    + "use '|' for alternation\nuse '*' for repetition");
170         // it has to check the validity of RE and convert it from infix to postfix expression
171         // example re= a*b.a
172         //String re=input.next();
173         RE eval=new RE(re);
174         String postRE= eval.RE;
175         System.out.println("the postfix form "+postRE);
176         // if alpha:
177         // 1- convert alpha into nfa
178         //2- push nfa into stack
179         // if operator: pop the stack and call appropriate method
180         for (int i=0; i<postRE.length();i++){
181             if (isAlpha(postRE.charAt(i))){
182                 NFA nfa = new NFA(postRE.charAt(i));
183                 stack.push(nfa);
184             }
185
186             else if (postRE.charAt(i)=='.'){
187                 NFA nfa2=stack.pop();
188                 NFA nfa1=stack.pop();
189                 stack.push(concat(nfa1,nfa2));
190             }
191
192             else if (postRE.charAt(i)=='|'){
193                 NFA nfa2=stack.pop();
194                 NFA nfa1=stack.pop();
195                 stack.push(alternate(nfa1,nfa2));
196             }
197             else{
198                 NFA nfa=stack.pop();
199                 stack.push(star(nfa));
200             }
201         }
202
203
204         NFA nfa=stack.pop();
205
206         return nfa;
207     }
208
209     public boolean isAlpha(char c){
210         return c>='a' && c<='z' || c=='E';
211     }
212
213
214
215
216
217
218
219
220
221 }

```

- Class Trans:

```

package NFAandDFA;

// this class to make transaction for nfa states

public class Trans implements Comparable<Trans>{
    int stateFrom, stateTo;
    char transSymbol;

    public Trans(int state_from, int state_to, char trans_symbol) {
        this.stateFrom = state_from;
        this.stateTo = state_to;
        this.transSymbol = trans_symbol;
    }

    // method to sort transaction
    @Override
    public int compareTo(Trans compare) {
        int compareage=((Trans)compare).stateFrom;
        /* For Ascending order*/
        return this.stateFrom-compareage;
    }

    public String print(){
        return (stateFrom+ " ---- "+transSymbol+" ----> "+stateTo);
    }
}

```

- Class RE:

```

7 import java.util.ArrayDeque;
8 import java.util.Deque;
9 // this class used to convert the RE from infix format to postfix format
10 public class RE {
11     String RE;
12     Deque<Character> stack = new ArrayDeque<Character>();
13
14     public RE(String RE) {
15         if (isValid(RE)) {
16             this.RE = toPostfix(add_join_symbol(RE));
17         } else System.out.println("RE is not valid ");
18     }
19
20     // check if the RE is valid
21     public boolean isValid(String RE){
22         if (RE.isEmpty())
23             return false;
24         else {
25             for ( char c : RE.toCharArray())
26                 if ( !isOperator(c) && !isAlpha(c) )
27                     return false;
28             }
29
30             return true;
31         }
32
33         // method to add . in re
34         public String add_join_symbol(String re) {
35
36             if(re.length()==1) {
37
38

```



```
103         newRE += stack.peek();
104         stack.pop();
105     }
106     return newRE;
107 }
108 
109 public int prec(char c){
110     switch (c) {
111         case '|':
112             return 1;
113 
114         case '.':
115             return 2;
116         case '*':
117             return 3;
118     }
119     return -1;
120 }
121 
122 }
123 }
124 }
```

- Class NFAToDFA:

```
6 
7 import java.util.ArrayList;
8 import java.util.Collections;
9 import java.util.Stack;
10 import java.util.Queue;
11 import java.util.LinkedList;
12 
13 public class NFAToDFA {
14     // this class to find the set states of DFA
15     ArrayList<ArrayList<Integer>> setStates;
16     ArrayList<TransDFA> transactions;
17     NFA nfa;
18     ArrayList<Character> symbols;
19     ArrayList<Integer> initialState;
20     ArrayList<ArrayList<Integer>> finalState;
21 
22     public NFAToDFA(NFA nfa) {
23         this.nfa = nfa;
24         transactions=new ArrayList<TransDFA> ();
25         setStates=new ArrayList<ArrayList<Integer>> ();
26         symbols=new ArrayList<Character> ();
27         recognizesSymbol(nfa);
28 
29         initialState=new ArrayList<Integer> ();
30         finalState=new ArrayList<ArrayList<Integer>> ();
31     }
32     // to find the inputs
33     public void recognizesSymbol(NFA nfa) {
34         for( Trans c : nfa.transactions)
35             if(!symbols.contains(c.transSymbol) && c.transSymbol!='E' )
36                 symbols.add(c.transSymbol);
37     }
38 }
```

```
37
38 // find E-cluser for a specific set
39 public ArrayList<Integer> e_cluser(int state){
40     Stack<Integer> stack = new Stack<Integer>();
41     ArrayList<Integer> states =new ArrayList<Integer> ();
42     stack.push(state);
43     while(!stack.empty()){
44         int x=stack.pop();
45         for (Trans s : nfa.transactions){
46             if(!states.contains(x))
47                 states.add(x);
48             if(x==s.stateFrom && s.transSymbol=='E')
49                 stack.add(s.stateTo);
50         }
51         Collections.sort(states);
52     }
53     return states;
54 }
55 // method to find transaction for set when the input is symbol
56 public ArrayList<Integer> move( ArrayList<Integer> state, char symbol)
57 {
58     ArrayList<Integer> x= new ArrayList<Integer>();
59     for(Integer i : state){
60         for(Trans s : nfa.transactions)
61             if (i==s.stateFrom && symbol==s.transSymbol){
62                 x.add(s.stateTo);
63                 //System.out.println(i);
64             }
65         Collections.sort(x);
66     }
67     return x;
68 }
69 // method to check on the states and find the transactions
```

```
Source History         

```
// method to check on the states and find the transactions
public void compile(){
 Queue<ArrayList<Integer>> queue=new LinkedList <ArrayList<Integer>>();
 setStates.add(e_cluser(0));
 queue.add(e_cluser(0));
 initialState=queue.peek();
 while(!queue.isEmpty()){
 ArrayList<Integer> q = new ArrayList<Integer>();
 q=queue.remove();
 for (Character c : symbols){
 ArrayList<Integer> moveResult = new ArrayList<Integer>();
 moveResult=move(q, c);
 ArrayList<Integer> ecluserforMove = new ArrayList<Integer>();
 for(Integer i : moveResult){
 ecluserforMove=combine(ecluserforMove,e_cluser(i));
 }
 if (ecluserforMove.size()!=0 && !setStates.contains(ecluserforMove)){
 setStates.add(ecluserforMove);
 queue.add(ecluserforMove);
 }
 }
 if (ecluserforMove.size()!=0)
 transactions.add(new TransDFA(q,ecluserforMove,c));
 }
 System.out.println("States for DFA :" +setStates);
 printDFA();
}
public void printDFA(){
```


```

```

100     System.out.println("Transaction for the DFA : ");
101     for(TransDFA d: this.transactions)
102         d.print();
103     System.out.println("Initial state:"+ this.initialState);
104     this.finalState=finalstates(sets);
105     System.out.println("final state:"+ this.finalState);
106
107 }
108
109
110
111
112     public ArrayList<Integer> combine(ArrayList<Integer> real, ArrayList<Integer> res){
113         for (Integer r:res){
114             if(!real.contains(r))
115                 real.add(r);
116         }
117         return real;
118     }
119
120     public ArrayList< ArrayList<Integer>> finalstates(ArrayList<ArrayList<Integer>> sets){
121         ArrayList<ArrayList<Integer>> fs=new ArrayList<ArrayList<Integer>> ();
122         for (ArrayList<Integer> s : sets){
123             for(Integer i : s){
124                 if(i== nfa.finalState)
125                     if(!fs.contains(s))
126                         fs.add(s);
127             }
128         }
129         return fs;
130     }
131

```

- Class TransDFA:

```

5     package NFAandDFA;
6
7     import java.util.ArrayList;
8
9     // class to store the transactions of DFA states
10    public class TransDFA {
11
12        ArrayList<Integer> stateFrom = new ArrayList<Integer>();
13        ArrayList<Integer> stateTo = new ArrayList<Integer>();
14        char symbol;
15
16        public TransDFA(ArrayList<Integer> stateFrom, ArrayList<Integer> stateTo, char symbol) {
17
18            this.stateFrom = stateFrom;
19            this.stateTo = stateTo;
20            this.symbol = symbol;
21        }
22
23
24        public void print(){
25            if(stateTo.size() !=0)
26                System.out.println(stateFrom.toString()+" ---- "+symbol+" ----> "+stateTo.toString());
27        }
28
29    }
30

```

- Class DFA:

Output of class NFAToDFA ((before convert it into char)):

```

new RE is: (a|b)*.a.b.b
the postfix form ab*ab.b.

States for DFA : [[0, 1, 2, 3, 4, 5, 6, 7], [1, 2, 3, 4, 6, 7, 8], [1, 2, 4, 5, 6, 7], [1, 2, 4, 5, 6, 7, 9], [1, 2, 4, 5, 6, 7, 10]]

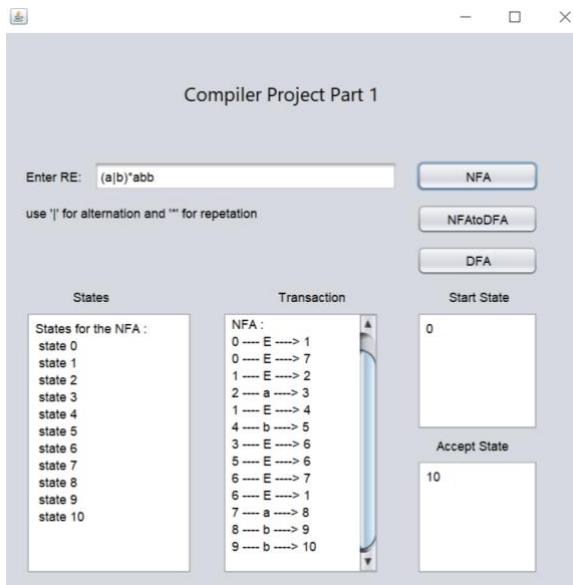
Transaction for the DFA :
[0, 1, 2, 4, 7] ---- a ----> [1, 2, 3, 4, 6, 7, 8]
[0, 1, 2, 4, 7] ---- b ----> [1, 2, 4, 5, 6, 7]
[1, 2, 3, 4, 6, 7, 8] ---- a ----> [1, 2, 3, 4, 6, 7, 8]
[1, 2, 3, 4, 6, 7, 8] ---- b ----> [1, 2, 4, 5, 6, 7, 9]
[1, 2, 4, 5, 6, 7] ---- a ----> [1, 2, 3, 4, 6, 7, 8]
[1, 2, 4, 5, 6, 7] ---- b ----> [1, 2, 4, 5, 6, 7]
[1, 2, 4, 5, 6, 7, 9] ---- a ----> [1, 2, 3, 4, 6, 7, 8]
[1, 2, 4, 5, 6, 7, 9] ---- b ----> [1, 2, 4, 5, 6, 7, 10]
[1, 2, 4, 5, 6, 7, 10] ---- a ----> [1, 2, 3, 4, 6, 7, 8]
[1, 2, 4, 5, 6, 7, 10] ---- b ----> [1, 2, 4, 5, 6, 7]

Initial state:[0, 1, 2, 4, 7]
final state:[1, 2, 4, 5, 6, 7, 10]]

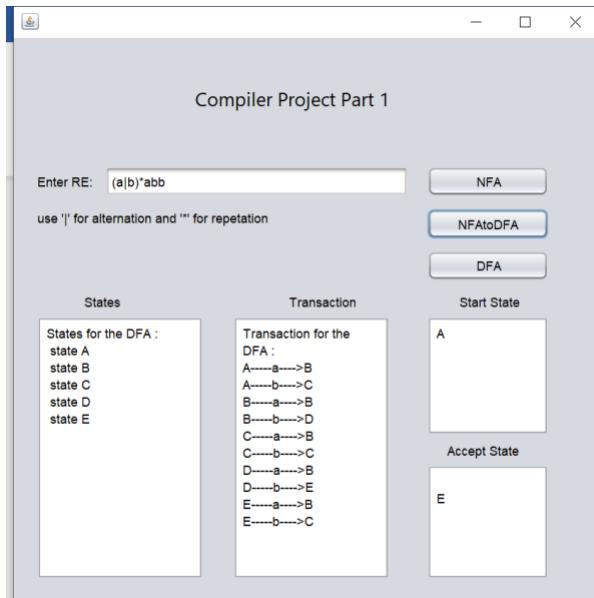
```

GUI output:

NFA:



NFAToDFA ((after convert it into char)):



Code DFA from a given regular expression directly:

- Class BinaryTree:

File Edit Selection View Go Run Terminal Help

BinaryTree.java - compiler5 - Visual Studio Code

J BinaryTreeNode.java ●

```
src > DFADir > J BinaryTree.java > BinaryTreeNode.java > leafNodeID
1
2 package DFADir;
3
4 import java.util.Arrays;
5 import java.util.HashMap;
6 import java.util.HashSet;
7 import java.util.Set;
8
9 import java.util.*;
10
11
12 public class BinaryTree {
13
14     /**
15      ***
16      | (a|b)*a => creating binary syntax tree:
17      |   .
18      |   / \
19      |   *   a
20      |   /
21      |   / \
22      |   a   b
23      ***
24
25     */
26
27     public static int leafNodeID = 0;
28
29     // Stacks for symbol nodes and operators
30     public Stack<Node> stackNode = new Stack<>();
31     private Stack<Character> operator = new Stack<Character>();
32
33     // Set of inputs
34     private Set<Character> input = new HashSet<Character>();
35 }
```

Activate Windows
Go to Settings to activate Windows.

The screenshot shows a Java file named `BinaryTree.java` open in Visual Studio Code. The code implements a class `BinaryTree` that generates a binary tree from a regular expression. It uses two helper classes: `leafNodeID` and `Character`. The `getSymbols` method extracts symbols from the regular expression, and the `generateTree` method constructs the tree. The code is annotated with comments explaining its purpose and logic.

```
src > DFADir > J BinaryTree.java > BinaryTree > leafNodeID
 36 private ArrayList<Character> op = new ArrayList<>();
 37
 38     private Set<String> Char; //set of characters
 39
 40     private HashMap<Integer, String> symbNum; //set of Integer
 41
 42     //this method is to get symbols
 43     public void getSymbols(String re) {
 44
 45         /**
 46          * op is a set of characters have operational meaning for example '*'
 47          * could be a closure operator
 48         */
 49
 50         Set<Character> op = new HashSet<>();
 51         Character[] ch = {'(', ')', '*', '|', '&', '.', '[' , ']' , '+'};
 52         op.addAll(Arrays.asList(ch));
 53
 54         Char = new HashSet<>();
 55         symbNum = new HashMap<>();
 56         int num = 1;
 57         for (int i = 0; i < re.length(); i++) {
 58             char charAt = re.charAt(i);
 59             if (op.contains(charAt)) {
 60                 Char.add("'" + charAt);
 61                 symbNum.put(num++, "'" + charAt);
 62             }
 63         }
 64     }
 65
 66
 67     // Generates tree using the regular expression and returns it's root
 68     public Node generateTree(String regular) {
 69
 70         Character[] ops = {"*", '|', '.'};
```

BinaryTree.java

```
src > DFADir > J BinaryTree.java > Binary Tree > leafNodeID
71 op.addAll(Arrays.asList(ops));
72
73 // Only inputs available
74 Character ch[] = new Character[26 + 26];
75 for (int i = 65; i <= 90; i++) {
76     ch[i - 65] = (char) i;
77     ch[i - 65 + 26] = (char) (i + 32);
78 }
79 Character integer[] = {'0', '1', '2', '3', '4', '5', '6', '7', '8', '9'};
80 Character others[] = {'#', '\\', '=', '_', '^', '*', '/', '+', '-', '(', ')'};
81 input.addAll(Arrays.asList(ch));
82 input.addAll(Arrays.asList(integer));
83 input.addAll(Arrays.asList(others));
84
85 // Generate regular expression with the concatenation
86 regular = AddConcat(regular);
87
88 // Cleaning stacks
89 stackNode.clear();
90 operator.clear();
91
92 // Flag which is true when there is something like: \( or \* or etc
93 boolean isSymbol = false;
94
95 for (int i = 0; i < regular.length(); i++) {
96
97     if (regular.charAt(i) == '\\') {
98         isSymbol = true;
99         continue;
100    }
101    if (isSymbol || isInputCharacter(regular.charAt(i))) {
102        if (isSymbol) {
103            //create a node with "\{symbol}" symbol
104            pushStack("\\"+Character.toString(regular.charAt(i)));
105        }
106    }
107    else{
108        pushStack(Character.toString(regular.charAt(i)));
109    }
110    isSymbol = false;
111 } else if (operator.isEmpty()) {
112     operator.push(regular.charAt(i));
113
114 } else if (regular.charAt(i) == '(') {
115     operator.push(regular.charAt(i));
116
117 } else if (regular.charAt(i) == ')') {
118     while (operator.get(operator.size() - 1) != '(') {
119         doOperation();
120     }
121
122     // Pop the '(' left parenthesis
123     operator.pop();
124
125 } else {
126     while (!operator.isEmpty()
127           && Priority(regular.charAt(i), operator.get(operator.size() - 1)) {
128         doOperation();
129     }
130     operator.push(regular.charAt(i));
131 }
132
133 // Clean the remaining elements in the stack
134 while (!operator.isEmpty()) {
135     doOperation();
136 }
137
138 // Get the complete Tree
139 Node completeTree = stackNode.pop();
```

BinaryTree.java

```
src > DFADir > J BinaryTree.java > Binary Tree > leafNodeID
105 }
106 else{
107     pushStack(Character.toString(regular.charAt(i)));
108 }
109 isSymbol = false;
110 } else if (operator.isEmpty()) {
111     operator.push(regular.charAt(i));
112
113 } else if (regular.charAt(i) == '(') {
114     operator.push(regular.charAt(i));
115
116 } else if (regular.charAt(i) == ')') {
117     while (operator.get(operator.size() - 1) != '(') {
118         doOperation();
119     }
120
121     // Pop the '(' left parenthesis
122     operator.pop();
123
124 } else {
125     while (!operator.isEmpty()
126           && Priority(regular.charAt(i), operator.get(operator.size() - 1)) {
127         doOperation();
128     }
129     operator.push(regular.charAt(i));
130 }
131
132 // Clean the remaining elements in the stack
133 while (!operator.isEmpty()) {
134     doOperation();
135 }
136
137 // Get the complete Tree
138 Node completeTree = stackNode.pop();
```

```
File Edit Selection View Go Run Terminal Help • BinaryTree.java - compiler5 - Visual Studio Code
J BinaryTree.java ●
src > DFADir > J BinaryTree.java > Binary Tree > leafNodeID
139     Node completeTree = stackNode.pop();
140     return completeTree;
141 }
142
143 private boolean Priority(char first, Character second) {
144     if (first == second) {
145         return true;
146     }
147     if (first == '*') {
148         return false;
149     }
150     if (second == '*') {
151         return true;
152     }
153     if (first == '.') {
154         return false;
155     }
156     if (second == '.') {
157         return true;
158     }
159     if (first == '|') {
160         return false;
161     }
162     return true;
163 }
164
165 // Do the desired operation based on the top of stackNode
166 private void doOperation() {
167     if (this.operator.size() > 0) {
168         char charAt = operator.pop();
169
170         switch (charAt) {
171             case '|':
172                 union();
173                 break;
174
175             case '.':
176                 concatenation();
177                 break;
178
179             case '*':
180                 star();
181                 break;
182
183             default:
184                 System.out.println(">>" + charAt);
185                 System.out.println("Unknown Symbol !");
186                 System.exit(status: 1);
187                 break;
188         } } }
189
190 // Do the star operation
191 private void star() {
192     // Retrieve top Node from Stack
193     Node node = stackNode.pop();
194
195     Node root = new Node(symbol: "*");
196     root.setLeft(node);
197     root.setRight(right: null);
198     node.setParent(root);
199
200     // Put node back in the stackNode
201     stackNode.push(root);
202 }
203
204 // Do the concatenation operation
205 private void concatenation() {
206     // retrieve node 1 and 2 from stackNode
207     Node node2 = stackNode.pop();
```

```
File Edit Selection View Go Run Terminal Help • BinaryTree.java - compiler5 - Visual Studio Code
J BinaryTree.java ●
src > DFADir > J BinaryTree.java > Binary Tree > leafNodeID
173         break;
174
175     case ('.'):
176         concatenation();
177         break;
178
179     case ('*'):
180         star();
181         break;
182
183     default:
184         System.out.println(">>" + charAt);
185         System.out.println("Unknown Symbol !");
186         System.exit(status: 1);
187         break;
188     } } }
189
190 // Do the star operation
191 private void star() {
192     // Retrieve top Node from Stack
193     Node node = stackNode.pop();
194
195     Node root = new Node(symbol: "*");
196     root.setLeft(node);
197     root.setRight(right: null);
198     node.setParent(root);
199
200     // Put node back in the stackNode
201     stackNode.push(root);
202 }
203
204 // Do the concatenation operation
205 private void concatenation() {
206     // retrieve node 1 and 2 from stackNode
207     Node node2 = stackNode.pop();
```

File Edit Selection View Go Run Terminal Help

BinaryTree.java

```

src > DFADir > J BinaryTree.java > Binary Tree > leafNodeID
207     Node node2 = stackNode.pop();
208     Node node1 = stackNode.pop();
209
210     Node root = new Node(symbol: ".");
211     root.setLeft(node1);
212     root.setRight(node2);
213     node1.setParent(root);
214     node2.setParent(root);
215
216     // Put node back to stackNode
217     stackNode.push(root); }

218
219 // Makes union of sub Node 1 with sub Node 2
220 private void union() {
221     // Load two Node in stack into variables
222     Node node2 = stackNode.pop();
223     Node node1 = stackNode.pop();
224
225     Node root = new Node(symbol: "|");
226     root.setLeft(node1);
227     root.setRight(node2);
228     node1.setParent(root);
229     node2.setParent(root);
230     // Put Node back to stack
231     stackNode.push(root); }

232
233 // Push input symbol into stackNode
234 private void pushStack(String symbol) {
235     Node node = new LeafNode(symbol, ++leafNodeID);
236     node.setLeft(left: null);
237     node.setRight(right: null);
238
239     // Put Node back to stackNode
240     stackNode.push(node); }

241
242 // add "." when is concatenation between to symbols that: "-" -> "&"
243 // concatenates to each other
244 private String AddConcat(String regular) {
245     String newRegular = new String(original: "");
246
247     for (int i = 0; i < regular.length() - 1; i++) {
248         /*
249             *# consider a , b are characters in the E
250             *# and the set: {', ', '*', '+', '&', '['} are the operators
251             *# then, if '&' is the concat symbol, we have to concatenate such expressions:
252             *# a & b
253             *# a & (
254             *# ) & a
255             *# * & a
256             *# * & (
257             *# ) & (
258         */
259         if (regular.charAt(i) == '\\\' && isInputCharacter(regular.charAt(i + 1))) {
260             newRegular += regular.charAt(i);
261         } else if (regular.charAt(i) == '\\\' && regular.charAt(i + 1) == '(') {
262             newRegular += regular.charAt(i);
263         } else if ((isInputCharacter(regular.charAt(i)) || (regular.charAt(i) == '(' && i > 0 && regular.charAt(i - 1) == '\\\\')) && isInputCharacter(regular.charAt(i + 1))) {
264             newRegular += regular.charAt(i) + "\\\"";
265
266         } else if ((isInputCharacter(regular.charAt(i)) || (regular.charAt(i) == '(' && i > 0 && regular.charAt(i - 1) == '\\\\')) && regular.charAt(i + 1) == ')') {
267             newRegular += regular.charAt(i) + ".";
268
269         } else if (regular.charAt(i) == '*' && isInputCharacter(regular.charAt(i + 1))) {
270             newRegular += regular.charAt(i) + ".";
271
272         } else if (regular.charAt(i) == '*' && regular.charAt(i + 1) == '(') {
273
274     }

```

Activate Windows
Go to Settings to activate Windows.

Ln 14, Col 7 Spaces:4 UTF-8 CRLF {} Java

File Edit Selection View Go Run Terminal Help

BinaryTree.java

```

src > DFADir > J BinaryTree.java > Binary Tree > leafNodeID
241 // add "." when is concatenation between to symbols that: "-" -> "&"
242 // concatenates to each other
243 private String AddConcat(String regular) {
244     String newRegular = new String(original: "");
245
246     for (int i = 0; i < regular.length() - 1; i++) {
247         /*
248             *# consider a , b are characters in the E
249             *# and the set: {', ', '*', '+', '&', '['} are the operators
250             *# then, if '&' is the concat symbol, we have to concatenate such expressions:
251             *# a & b
252             *# a & (
253             *# ) & a
254             *# * & a
255             *# * & (
256             *# ) & (
257             *#
258         */
259         if (regular.charAt(i) == '\\\' && isInputCharacter(regular.charAt(i + 1))) {
260             newRegular += regular.charAt(i);
261         } else if (regular.charAt(i) == '\\\' && regular.charAt(i + 1) == '(') {
262             newRegular += regular.charAt(i);
263         } else if ((isInputCharacter(regular.charAt(i)) || (regular.charAt(i) == '(' && i > 0 && regular.charAt(i - 1) == '\\\\')) && isInputCharacter(regular.charAt(i + 1))) {
264             newRegular += regular.charAt(i) + "\\\"";
265
266         } else if ((isInputCharacter(regular.charAt(i)) || (regular.charAt(i) == '(' && i > 0 && regular.charAt(i - 1) == '\\\\')) && regular.charAt(i + 1) == ')') {
267             newRegular += regular.charAt(i) + ".";
268
269         } else if (regular.charAt(i) == '*' && isInputCharacter(regular.charAt(i + 1))) {
270             newRegular += regular.charAt(i) + ".";
271
272         } else if (regular.charAt(i) == '*' && regular.charAt(i + 1) == '(') {
273
274     }

```

Activate Windows
Go to Settings to activate Windows.

Ln 14, Col 7 Spaces:4 UTF-8 CRLF {} Java

```
BinaryTree.java
src > DFADir > J BinaryTree.java > Binarytree > leafNodeID
269     } else if (regular.charAt(i) == ')' && isInputCharacter(regular.charAt(i + 1))) {
270         newRegular += regular.charAt(i) + ".";
271
272     } else if (regular.charAt(i) == '*' && isInputCharacter(regular.charAt(i + 1))) {
273         newRegular += regular.charAt(i) + ".";
274
275     } else if (regular.charAt(i) == '^' && regular.charAt(i + 1) == '(') {
276         newRegular += regular.charAt(i) + ".";
277
278     } else if (regular.charAt(i) == ')' && regular.charAt(i + 1) == '(') {
279         newRegular += regular.charAt(i) + "";
280
281     } else {
282         newRegular += regular.charAt(i);
283     }
284     newRegular += regular.charAt(regular.length() - 1);
285     return newRegular;
286 }
287 // Return true if is part of the automata Language else is false
288 private boolean isInputCharacter(char charAt) {
289
290     if (op.contains(charAt)) {
291         return false;
292     for (Character c : input) {
293         if ((char) c == charAt && charAt != '(' && charAt != ')') {
294             return true;
295         }
296     }
297     return false;
298 }
299 //to get number of leafs
300 public static int getNumberOfLeafs(){
301     return leafNodeID;
302 }
303 }
```

Activate Windows
Go to Settings to activate Windows.

Ln 14, Col 7 Spaces:4 UTF-8 CRLF Java

- Class SyntaxTree:

SyntaxTree.java - compiler5 - Visual Studio Code

```

File Edit Selection View Go Run Terminal Help SyntaxTree.java 1 ×
src > DFADir > J SyntaxTree.java > bt SyntaxTree > bt
1
2 package DFADir;
3
4 import java.util.*;
5
6
7 public class SyntaxTree {
8
9     public String regex;
10    public static BinaryTree bt;
11    private Node root; //the head of raw syntax tree
12    private int numOfeleafs; // the number of leafs
13    ArrayList<ArrayList<Integer>> fol; // the set of follow pos
14    ArrayList<Node> leafnode; //the leaf node
15
16    public SyntaxTree(String regex) {
17        this.regex = regex;
18        bt = new BinaryTree();
19        fol= new ArrayList<ArrayList<Integer>>();
20        /**
21         * generates the binary tree of the syntax tree
22         */
23        root = bt.generateTree(regex);
24
25        numOfeleafs = bt.getNumberOfeleafs();
26
27        for(int i=0; i< numOfeleafs ;i++ )
28            fol.add(new ArrayList<Integer>());
29        this.leafnode = new ArrayList<Node>();
30        leaf( root );
31
32        generateNullable(root);
33        generateFirstposLastPos(root);
34        generateFollowPos(root);
35    }

```

Activate Windows
Go to Settings to activate Windows.

SyntaxTree.java - compiler5 - Visual Studio Code

```

File Edit Selection View Go Run Terminal Help SyntaxTree.java 1 ×
src > DFADir > J SyntaxTree.java > bt SyntaxTree > bt
35
36    public void generateNullable(Node node) {
37        if (node == null) {
38            return;
39        }
39        if (!(node instanceof LeafNode)) {
40            Node left = node.getLeft();
41            Node right = node.getRight();
42            generateNullable(left);
43            generateNullable(right);
44            switch (node.getSymbol()) {
45                //if "|" we will take nullable c1 OR nullable c2
46                case "|":
47                    node.setNullable(left.isNullable() | right.isNullable());
48                    break;
49                    //if "," we will take nullable c1 AND nullable c2
50                case ",":
51                    node.setNullable(left.isNullable() & right.isNullable());
52                    break;
53                    //if "*" will be true
54                case "*":
55                    node.setNullable(nullable: true);
56                    break;
57            }
58        }
59        public void generateFirstposLastPos(Node node) {
60            if (node == null) {
61                return;
62            }
63            if (node instanceof LeafNode) {
64                LeafNode lnode = (LeafNode) node;
65                node.addToFirstPos(lnode.getNum());
66                node.addToLastPos(lnode.getNum());
67            } else {
68                Node left = node.getLeft();
69                Node right = node.getRight();
                generateFirstposLastPos(left);

```

Activate Windows
Go to Settings to activate Windows.

SyntaxTreeJava - compiler5 - Visual Studio Code

```

File Edit Selection View Go Run Terminal Help
J SyntaxTree.java 1 ×
src > DFADir > J SyntaxTreeJava > SyntaxTree > bt
69     generateFirstposLastPos(left);
70     generateFirstposLastPos(right);
71
72     switch (node.getSymbol()) {
73         case "|":
74             // for both first and last pos for case "|" we will take c1&c2
75             node.addAllToFirstPos(left.getFirstPos());
76             node.addAllToFirstPos(right.getFirstPos());
77             node.addAllToLastPos(left.getLastPos());
78             node.addAllToLastPos(right.getLastPos());
79
80             break;
81         case ".":
82             //if c1 is nullable, first pos will be c1&c2
83             if (left.isNullable()) {
84                 node.addAllToFirstPos(left.getFirstPos());
85                 node.addAllToFirstPos(right.getFirstPos());
86             } else {
87                 //if c1 is not nullable, first pos will be c1
88                 node.addAllToFirstPos(left.getFirstPos());
89             }
90             //if c2 is nullable, last pos will be c1&c2
91             if (right.isNullable()) {
92                 node.addAllToLastPos(left.getLastPos());
93                 node.addAllToLastPos(right.getLastPos());
94             } else {
95                 //if c1 is not nullable, last pos will be c2
96                 node.addAllToLastPos(right.getLastPos());
97             }
98             break;
99
100        case "*":
101            //for both first and last pos we will take (c1)
102            node.addAllToFirstPos(left.getFirstPos());
103            node.addAllToLastPos(left.getLastPos());

```

Activate Windows
Go to Settings to activate Windows.

Ln 10, Col 33 Spaces:4 UTF-8 CRLF ⓘ Java ⚙

SyntaxTreeJava - compiler5 - Visual Studio Code

```

File Edit Selection View Go Run Terminal Help
J SyntaxTree.java 1 ×
src > DFADir > J SyntaxTreeJava > SyntaxTree > bt
103         node.addAllToLastPos(left.getLastPos());
104         break; }
105     System.out.println(node.getSymbol()); } }

106     public void generateFollowPos(Node node) {
107         if (node == null) {
108             return;
109         }
110     }

111     switch (node.getSymbol()) {
112         //first we will take last pos(c1) and first pos(c2)
113         //in case "." the follow pos(c1) =c2
114         case ".":{
115             ArrayList<Integer> k = new ArrayList<Integer>();
116             k.addAll(node.getRight().getFirstPos());
117             for (Integer i : node.getLeft().getLastPos() ){
118                 k=combine(fol.get(i-1),k);
119                 System.out.println("i="+i + " fol"+ k); }
120             break;
121         //in case "*" the follow pos(c2) =c1
122         case "*":{
123             ArrayList<Integer> k = new ArrayList<Integer>();
124             k.addAll(node.getLastPos());
125             for (Integer i : node.getFirstPos() ){
126                 k=combine(fol.get(i-1),k);
127                 System.out.println("i="+i + " fol"+ k); }
128             break; }

129         generateFollowPos(node.getLeft());
130         generateFollowPos(node.getRight()); }

131     public ArrayList<Integer> combine(ArrayList<Integer> real, ArrayList<Integer> res){
132         for (Integer r:res){
133             if(!real.contains(r))
134                 real.add(r); }


```

Activate Windows
Go to Settings to activate Windows.

Ln 10, Col 33 Spaces:4 UTF-8 CRLF ⓘ Java ⚙

SyntaxTree.java - compiler5 - Visual Studio Code

```
File Edit Selection View Go Run Terminal Help SyntaxTree.java 1 X
src > DFADir > J SyntaxTree.java > bt SyntaxTree > bt
137     |     real.add(r); }
138     |     Collections.sort(real);
139     |     return real;
140
141     //method to get root
142     public Node getRoot() {
143         return this.root;
144
145     //this method is to print the first pos and last pos
146     public static void printTree(Node node) {
147         //base case
148         if (node == null) {
149             return;
150
151         //first left child
152         if (node.getLeft() != null || node.getRight() != null)
153             { System.out.println(node.getSymbol() + " "); // to print the data of the node
154             System.out.println("The First pos is: " + node.getFirstPos().toString());
155             System.out.println("The Last pos is: " + node.getLastPos().toString());
156             System.out.println();
157         else if (node.getLeft() == null && node.getRight() == null)
158             { System.out.println(node.getSymbol() + " "); // to print the data of the node
159             System.out.println("The First pos is: " + node.getFirstPos().toString());
160             System.out.println("The Last pos is: " + node.getLastPos().toString());
161             System.out.println();
162         // now right child
163         printTree(node.getLeft());
164         printTree(node.getRight());
165
166     public void leaf(Node node) {
167         //base case
168         if (node == null) {
169             return;
170         //first left child
171         else if (node.getLeft() == null && node.getRight() == null)
172             { if (!node.getSymbol().equals(anObject: "#"))
173                 | this.leafnode.add(node);
174         // now right child
175         leaf(node.getLeft());
176         leaf(node.getRight()); }
```

SyntaxTree.java - compiler5 - Visual Studio Code

```
File Edit Selection View Go Run Terminal Help SyntaxTree.java 1 X
src > DFADir > J SyntaxTree.java > bt SyntaxTree > bt
143     |     return this.root;
144
145     //this method is to print the first pos and last pos
146     public static void printTree(Node node) {
147         //base case
148         if (node == null) {
149             return;
150
151         //first left child
152         if (node.getLeft() != null || node.getRight() != null)
153             { System.out.println(node.getSymbol() + " "); // to print the data of the node
154             System.out.println("The First pos is: " + node.getFirstPos().toString());
155             System.out.println("The Last pos is: " + node.getLastPos().toString());
156             System.out.println();
157         else if (node.getLeft() == null && node.getRight() == null)
158             { System.out.println(node.getSymbol() + " "); // to print the data of the node
159             System.out.println("The First pos is: " + node.getFirstPos().toString());
160             System.out.println("The Last pos is: " + node.getLastPos().toString());
161             System.out.println();
162         // now right child
163         printTree(node.getLeft());
164         printTree(node.getRight());
165
166     public void leaf(Node node) {
167         //base case
168         if (node == null) {
169             return;
170         //first left child
171         else if (node.getLeft() == null && node.getRight() == null)
172             { if (!node.getSymbol().equals(anObject: "#"))
173                 | this.leafnode.add(node);
174         // now right child
175         leaf(node.getLeft());
176         leaf(node.getRight()); }}
```

- Class LeafNode:

LeafNode.java

```

src > DFADir > J LeafNode.java > LeafNode
1
2 package DFADir;
3
4 import java.util.HashSet;
5 import java.util.Set;
6
7
8 public class LeafNode extends Node {
9
10    private int num;
11    public static Set<Integer> followPos;
12
13    public LeafNode(String symbol, int num) {
14        super(symbol);
15        this.num = num;
16        followPos = new HashSet<>();
17    }
18
19    /**
20     * @return the num
21     */
22    public int getNum() {
23        return num;
24    }
25
26    /**
27     * @param num the num to set
28     */
29    public void setNum(int num) {
30        this.num = num;
31    }
32
33    public void addToFollowPos(int number){
34        followPos.add(number);
35    }

```

Activate Windows
Go to Settings to activate Windows.

LeafNode.java

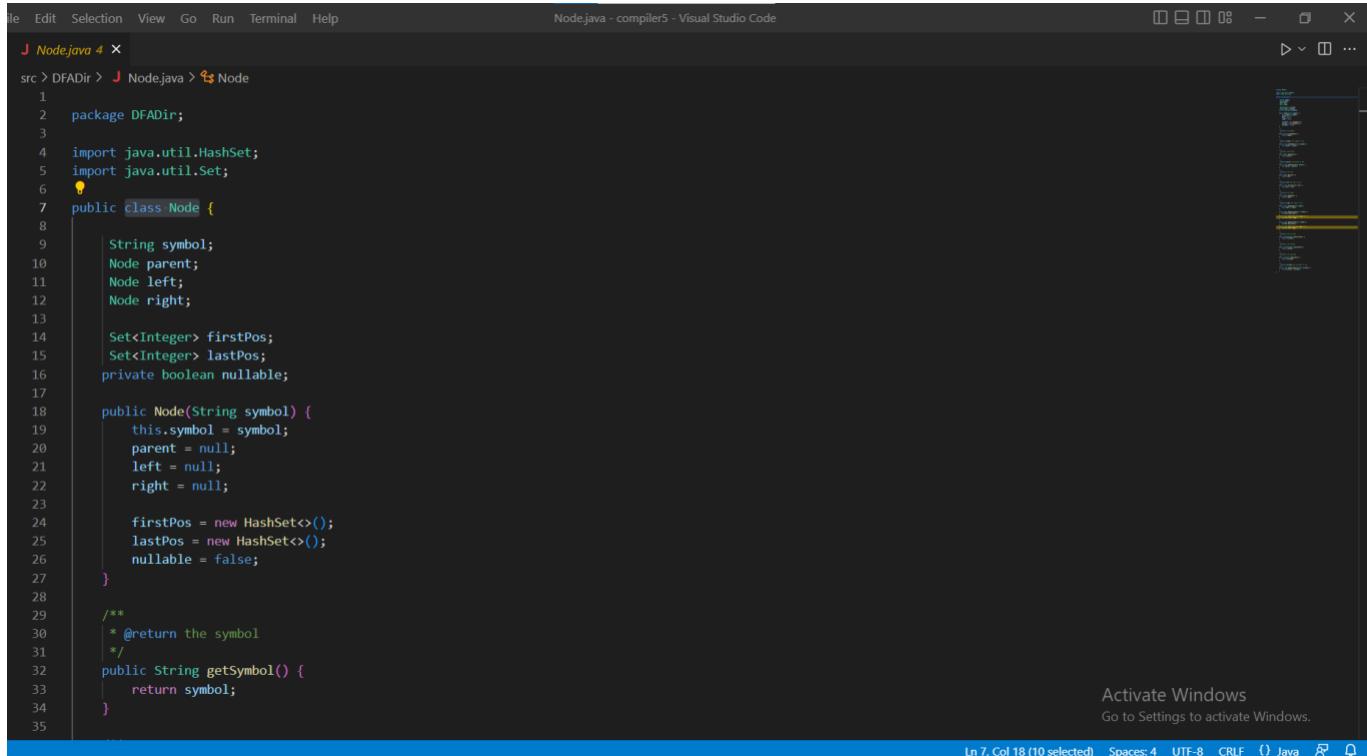
```

src > DFADir > J LeafNode.java > LeafNode
18
19 /**
20  * @return the num
21  */
22 public int getNum() {
23     return num;
24 }
25
26 /**
27  * @param num the num to set
28  */
29 public void setNum(int num) {
30     this.num = num;
31 }
32
33 public void addToFollowPos(int number){
34     followPos.add(number);
35 }
36
37 /**
38  * @return the followPos
39  */
40 public static Set<Integer> getFollowPos() {
41     return followPos;
42 }
43
44 /**
45  * @param followPos the followPos to set
46  */
47 public void setFollowPos(Set<Integer> followPos) {
48     LeafNode.followPos = followPos;
49 }
50 }

```

Activate Windows
Go to Settings to activate Windows.

- Class Node:



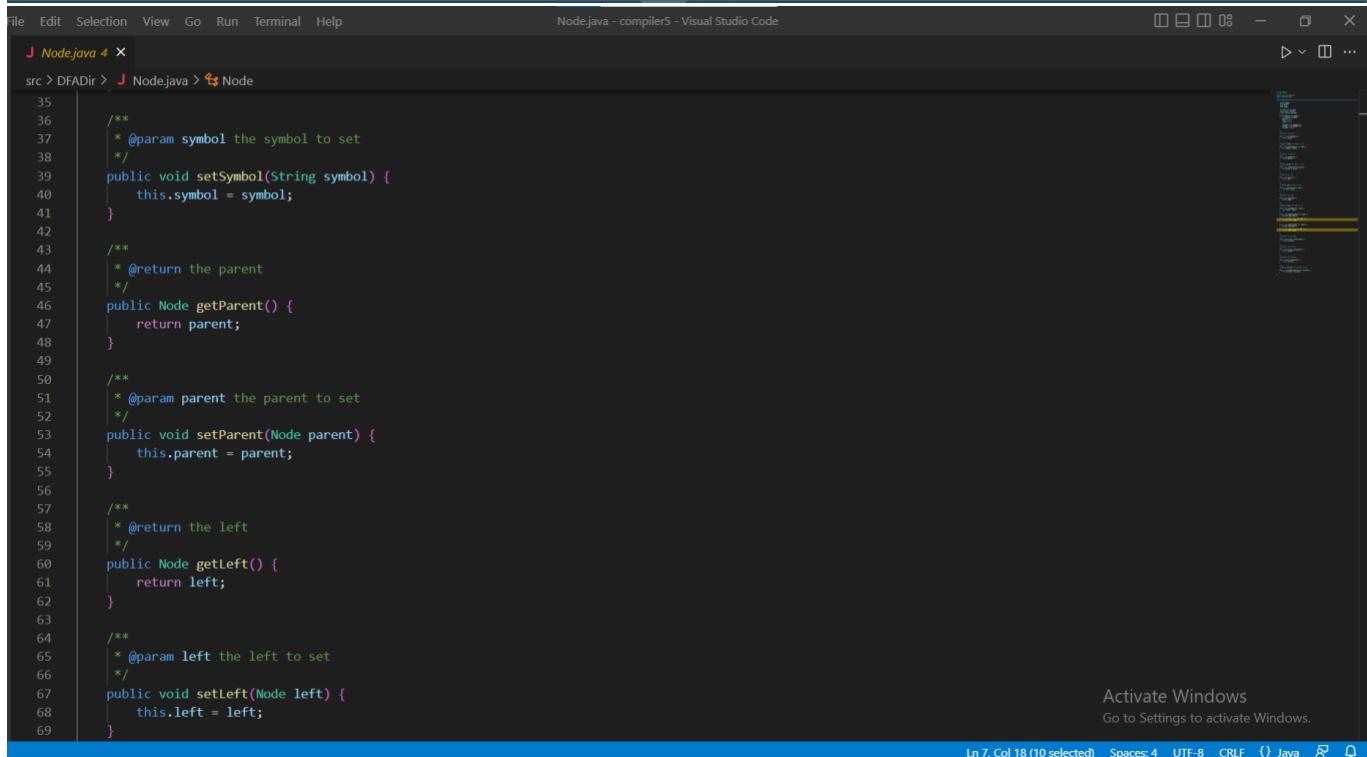
```

File Edit Selection View Go Run Terminal Help Node.java - compiler5 - Visual Studio Code
J Node.java 4 x
src > DFADir > J Node.java > Node
1
2 package DFADir;
3
4 import java.util.HashSet;
5 import java.util.Set;
6
7 public class Node {
8
9     String symbol;
10    Node parent;
11    Node left;
12    Node right;
13
14    Set<Integer> firstPos;
15    Set<Integer> lastPos;
16    private boolean nullable;
17
18    public Node(String symbol) {
19        this.symbol = symbol;
20        parent = null;
21        left = null;
22        right = null;
23
24        firstPos = new HashSet<>();
25        lastPos = new HashSet<>();
26        nullable = false;
27    }
28
29 /**
30  * @return the symbol
31 */
32 public String getSymbol() {
33     return symbol;
34 }
35

```

Activate Windows
Go to Settings to activate Windows.

Ln 7, Col 18 (10 selected) Spaces: 4 UTF-8 CRLF () Java ⚙



```

File Edit Selection View Go Run Terminal Help Node.java - compiler5 - Visual Studio Code
J Node.java 4 x
src > DFADir > J Node.java > Node
35
36 /**
37  * @param symbol the symbol to set
38 */
39 public void setSymbol(String symbol) {
40     this.symbol = symbol;
41 }
42
43 /**
44  * @return the parent
45 */
46 public Node getParent() {
47     return parent;
48 }
49
50 /**
51  * @param parent the parent to set
52 */
53 public void setParent(Node parent) {
54     this.parent = parent;
55 }
56
57 /**
58  * @return the left
59 */
60 public Node getLeft() {
61     return left;
62 }
63
64 /**
65  * @param left the left to set
66 */
67 public void setLeft(Node left) {
68     this.left = left;
69 }

```

Activate Windows
Go to Settings to activate Windows.

Ln 7, Col 18 (10 selected) Spaces: 4 UTF-8 CRLF () Java ⚙

File Edit Selection View Go Run Terminal Help Node.java - compiler5 - Visual Studio Code

```
J Node.java 4 x
src > DFADir > J Node.java > Node
69     }
70
71     /**
72      * @return the right
73      */
74     public Node getRight() {
75         return right;
76     }
77
78     /**
79      * @param right the right to set
80      */
81     public void setRight(Node right) {
82         this.right = right;
83     }
84
85     public void addToFirstPos(int number) {
86         firstPos.add(number);
87     }
88     public void addAllToFirstPos(Set<Set> set) {
89         firstPos.addAll(set);
90     }
91
92     public void addToLastPos(int number) {
93         lastPos.add(number);
94     }
95     public void addAllToLastPos(Set<Set> set) {
96         lastPos.addAll(set);
97     }
98
99     /**
100      * @return the firstPos
101      */
102     public Set<Integer> getFirstPos() {
103         return firstPos;
104     }
105
106     /**
107      * @return the lastPos
108      */
109     public Set<Integer> getLastPos() {
110         return lastPos;
111     }
112
113     /**
114      * @return the nullable
115      */
116     public boolean isNullable() {
117         return nullable;
118     }
119
120     /**
121      * @param nullable the nullable to set
122      */
123     public void setNullable(boolean nullable) {
124         this.nullable = nullable;
125     }
126 }
```

Activate Windows
Go to Settings to activate Windows.

Ln 7, Col 18 (10 selected) Spaces: 4 UTF-8 CRLF () Java ⚙️ 🔍

File Edit Selection View Go Run Terminal Help Node.java - compiler5 - Visual Studio Code

```
J Node.java 4 x
src > DFADir > J Node.java > Node
95     public void addAllToLastPos(Set<Set> set) {
96         lastPos.addAll(set);
97     }
98
99     /**
100      * @return the firstPos
101      */
102     public Set<Integer> getFirstPos() {
103         return firstPos;
104     }
105
106     /**
107      * @return the lastPos
108      */
109     public Set<Integer> getLastPos() {
110         return lastPos;
111     }
112
113     /**
114      * @return the nullable
115      */
116     public boolean isNullable() {
117         return nullable;
118     }
119
120     /**
121      * @param nullable the nullable to set
122      */
123     public void setNullable(boolean nullable) {
124         this.nullable = nullable;
125     }
126 }
```

Activate Windows
Go to Settings to activate Windows.

Ln 7, Col 18 (10 selected) Spaces: 4 UTF-8 CRLF () Java ⚙️ 🔍

- Class DFADir:

File Edit Selection View Go Run Terminal Help DFADir.java - compiler5 - Visual Studio Code

```

J DFADir.java ×
src > DFADir > J DFADir.java > DFADir
1
2 package DFADir;
3
4 import java.util.ArrayList;
5
6 public class DFADir {
7     ArrayList<Character> setStates = new ArrayList<Character>();
8     char initialState;
9     ArrayList<Character> finalState;
10
11    public DFADir(REtoDFA dfa) {
12        finalState=new ArrayList<Character>();
13        setDfaStetes(dfa);
14        finalstate(dfa);
15        System.out.println("DFA states: "+setStates);
16        initialState=setStates.get(dfa.setStates.indexOf(dfa.initialState));
17        printTrans(dfa);
18        System.out.println("initial state is :"+ initialState);
19        System.out.println("final states are :" + finalState );
20    }
21}
22
23 public void setDfaStetes(REtoDFA dfa){
24     char c='A';
25     for(int i=0; i<dfa.setStates.size();i++ ){
26         this.setStates.add(c);
27         c++;
28     }
29 }
30
31
32 public void finalstate(REtoDFA dfa){
33     for(ArrayList<Integer> s: dfa.finalState)
34         this.finalState.add(setStates.get(dfa.setStates.indexOf(s)));
35 }
```

Activate Windows
Go to Settings to activate Windows.

Ln 6, Col 20 (6 selected) Spaces: 4 UTF-8 CRLF { Java  

File Edit Selection View Go Run Terminal Help DFADir.java - compiler5 - Visual Studio Code

```

J DFADir.java ●
src > DFADir > J DFADir.java > DFADir > printTrans(REtoDFA)
36
37
38    public String printDFAState(){
39        String s="States for the DFA :\n";
40        for(Character d: setStates )
41            s+=("state "+d+"\n ");
42        return s;
43    }
44
45    public String printStartState(){
46
47        return Character.toString(initialState);
48    }
49
50    public String printFinalState(){
51        String s="\n";
52        for(Character d: finalState )
53            s+=d+"\n ";
54        return s;
55    }
56
57
58    public String printTrans(REtoDFA dfa){
59        String st= "Transaction for the DFA :\n";
60        for(TransDFA s: dfa.transactions){
61            int posFrom=dfa.setStates.indexOf(s.stateFrom);
62            int posTo=dfa.setStates.indexOf(s.stateTo);
63
64            if( setStates.get(posTo) != -1)
65                st+=(setStates.get(posFrom)+"----"+s.symbol+"---->"+
66                setStates.get(posTo)+"\n");
67        }
68    }
69 }
```

Activate Windows
Go to Settings to activate Windows.

Ln 68, Col 6 Spaces: 4 UTF-8 CRLF { Java  

Class REtoDFA :

```

File Edit Selection View Go Run Terminal Help
REtoDFA.java •
src > DFADir > REtoDFA.java > {} DFADir
1 package DFADir;
2
3 import java.util.ArrayList;
4 import java.util.Collections;
5 import java.util.Queue;
6 import java.util.LinkedList;
7
8 public class REtoDFA {
9
10     ArrayList<ArrayList<Integer>> setStates;
11     ArrayList<TransDFA> transactions;
12     SyntaxTree tree;
13     ArrayList<Character> symbols;
14     ArrayList<Integer> initialState;
15     ArrayList<ArrayList<Integer>> finalState;
16
17     public REtoDFA(SyntaxTree tree) {
18         this.tree = tree;
19         transactions=new ArrayList<TransDFA> ();
20         setStates=new ArrayList<ArrayList<Integer>> ();
21         symbols=new ArrayList<Character>();
22         recognizeSymbol(tree.regex);
23         initialState=new ArrayList<Integer> ();
24         finalState=new ArrayList<ArrayList<Integer>> ();
25     }
26
27     public void recognizeSymbol(String re){
28         for( Character c : re.toCharArray()){
29             if( isAlpha(c))
30                 if(symbols.contains(c) )
31                     symbols.add(c); }
32
33     public boolean isAlpha(char c){
34         return c>='a' && c<='z' ; }
35

```

Activate Windows
Go to Settings to activate Windows.

```

File Edit Selection View Go Run Terminal Help
REtoDFA.java •
src > DFADir > REtoDFA.java > {} DFADir
35
36     public void compile(){
37
38         System.out.println(symbols);
39
40         Queue<ArrayList<Integer>> queue=new LinkedList <ArrayList<Integer>>();
41
42         ArrayList<Integer> first= new ArrayList<Integer>();
43         first.addAll(tree.getRoot().getFirstPos());
44         setStates.add(first);
45
46         queue.add(first);
47         initialState=queue.peek();
48         while(!queue.isEmpty()){
49             ArrayList<Integer> q = new ArrayList<Integer>();
50             q=queue.remove();
51             for (Character c : symbols){
52
53                 ArrayList<Integer> moveResult = new ArrayList<Integer>();
54                 moveResult=move(q, c);
55                 //here
56                 ArrayList<Integer> followPosSet = new ArrayList<Integer>();
57                 for(Integer i : moveResult){
58                     //her2
59                     followPosSet=combine(followPosSet,tree.fol.get(i-1)); }
60
61                 if (followPosSet.size() !=0 && !setStates.contains(followPosSet)){
62                     setStates.add(followPosSet);
63                     queue.add(followPosSet);
64                 }
65
66                 if(followPosSet.size()!=0)
67                     transactions.add(new TransDFA(q,followPosSet,c));
68             }
69         }

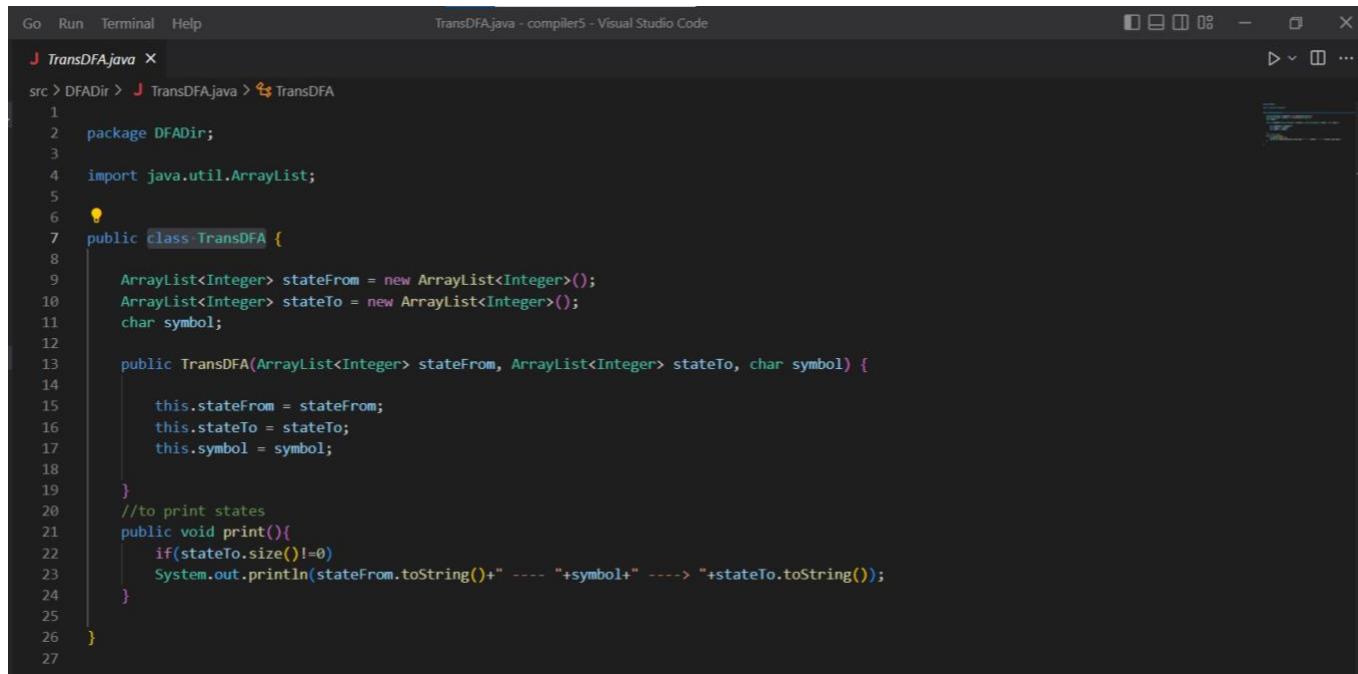
```

Activate Windows
Go to Settings to activate Windows.

```
File Edit Selection View Go Run Terminal Help • REtoDFAJava - compiler5 - Visual Studio Code
J REtoDFAJava •
src > DFADir > J REtoDFAJava > {} DFADir
69
70 System.out.println("States for DFA :" + setStates);
71
72     printDFA();
73
74     public ArrayList<Integer> combine(ArrayList<Integer> real, ArrayList<Integer> res){
75         for (Integer r : res){
76             if(!real.contains(r))
77                 real.add(r);
78         }
79         return real;
80     }
81
82     public ArrayList<Integer> move( ArrayList<Integer> state, char symbol){
83         // to find index with same input
84         ArrayList<Integer> x = new ArrayList<Integer>();
85         for(Integer i : state){
86             for (Node n : tree.leafnode)
87                 if (n.symbol.charAt(index: 0) == symbol && n.getFirstPos().contains(i) )
88                     x.add(i);
89         }
90         Collections.sort(x);
91         return x;
92     }
93
94     public void printDFA(){
95
96         System.out.println(x: "Transaction for the DFA : ");
97         for(TransDFA d : this.transactions)
98             d.print();
99         System.out.println("Intial state:" + this.initialState);
100        this.finalState=finalstates(setStates);
101        System.out.println("final state:" + this.finalState);
102    }
103    public ArrayList< ArrayList<Integer>> finalstates(ArrayList<ArrayList<Integer>> sets){
104
105
106
107
108
109
110
111
112
113
114 }
```

```
File Edit Selection View Go Run Terminal Help • REtoDFAJava - compiler5 - Visual Studio Code
J REtoDFAJava •
src > DFADir > J REtoDFAJava > {} DFADir
81
82     public ArrayList<Integer> move( ArrayList<Integer> state, char symbol){
83         // to find index with same input
84         ArrayList<Integer> x = new ArrayList<Integer>();
85         for(Integer i : state){
86             for (Node n : tree.leafnode)
87                 if (n.symbol.charAt(index: 0) == symbol && n.getFirstPos().contains(i) )
88                     x.add(i);
89         }
90         Collections.sort(x);
91         return x;
92     }
93
94     public void printDFA(){
95
96         System.out.println(x: "Transaction for the DFA : ");
97         for(TransDFA d : this.transactions)
98             d.print();
99         System.out.println("Intial state:" + this.initialState);
100        this.finalState=finalstates(setStates);
101        System.out.println("final state:" + this.finalState);
102    }
103    public ArrayList< ArrayList<Integer>> finalstates(ArrayList<ArrayList<Integer>> sets){
104        ArrayList<ArrayList<Integer>> fs=new ArrayList<ArrayList<Integer>> ();
105        for (ArrayList<Integer> s : sets){
106            for(Integer i : s){
107                if(i== (tree.leafnode.size()+1))
108                    if(!fs.contains(s))
109                        fs.add(s);
110            }
111        }
112        return fs;
113    }
114 }
```

- Class TransDFA:

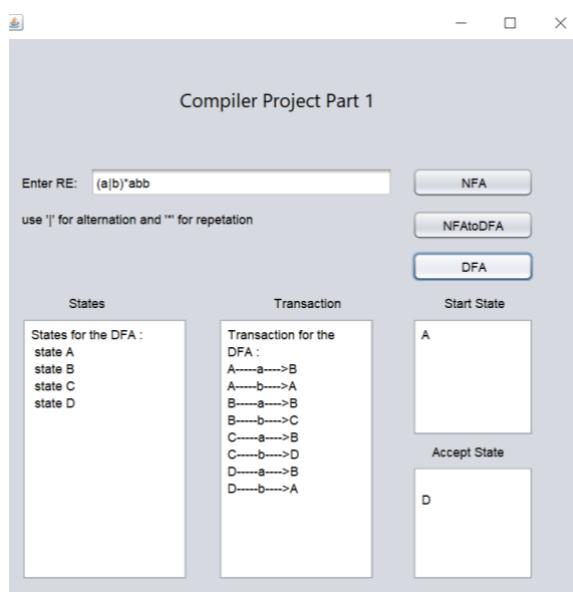


```

1  package DFADir;
2
3  import java.util.ArrayList;
4
5
6  public class TransDFA {
7
8      ArrayList<Integer> stateFrom = new ArrayList<Integer>();
9      ArrayList<Integer> stateTo = new ArrayList<Integer>();
10     char symbol;
11
12     public TransDFA(ArrayList<Integer> stateFrom, ArrayList<Integer> stateTo, char symbol) {
13
14         this.stateFrom = stateFrom;
15         this.stateTo = stateTo;
16         this.symbol = symbol;
17     }
18
19     //to print states
20     public void print(){
21         if(stateTo.size() != 0)
22             System.out.println(stateFrom.toString() + " ---- " + symbol + " ----> " + stateTo.toString());
23     }
24
25 }
26
27

```

DFA output GUI:



PART 2

2. Part two:

PROGRAM → STMTS

STMTS → STMT | STMT ; STMTS

STMT → id = EXPR

EXPR → EXPR + TERM | EXPR - TERM | TERM

TERM → TERM * FACTOR | TERM / FACTOR | FACTOR

FACTOR → (EXPR) | id | integer

Consider the grammar above which is an extension to what we had in class. Where {PROGRAM, STMTS, STMT, EXPR, TERM, FACTOR} are non-terminals and { ;, =, +, -, *, ,, (,), id, integer} are terminals. PROGRAM is the start non terminal, and the terminal id is defined using the regular expression: letter(letter digit)*.

2.1 Modify the grammar to add the power operator (^) and unary sign operators + and -

PROGRAM → STMTS

STMTS → STMT | STMT ; STMTS

STMT → id = EXPR

EXPR → EXPR + TERM | EXPR - TERM | TERM

TERM → TERM * POW | TERM / POW | POW

POW → UNARY ^ POW | UNARY

UNARY → - UNARY | + UNARY | FACTOR

FACTOR → (EXPR) | id | INTEGER

2.2 Re-write the grammar such that it can be parsed by an LL(1) parser

1) eliminate left recursion:

PROGRAM → STMTS

STMTS → STMT | STMT ; STMTS

STMT → id = EXPR

EXPR → TERM EXPR'

EXPR' → + TERM EXRPR' | - TERM EXRPR' | E

TERM → POW TERM'

TERM' → * POW TERM' | / POW TERM' | E

POW → UNARY ^ POW | UNARY

UNARY → - UNARY | + UNARY | FACTOR

FACTOR → (EXPR) | id | INTEGER

2) left factoring:

PROGRAM → STMTS

STMTS → STMT STMTS'

STMTS' → E | ; STMTS

STMT → id = EXPR

EXPR → TERM EXPR'

EXPR' → + TERM EXRPR' | - TERM EXRPR' | E

TERM → POW TERM'

TERM' → * POW TERM' | / POW TERM' | E

POW → UNARY POW'

POW' → ^ POW | E

UNARY → - UNARY | + UNARY | FACTOR

FACTOR → (EXPR) | id | INTEGER

So the final grammar is:

$\text{PROGRAM} \rightarrow \text{STMTS}$
 $\text{STMTS} \rightarrow \text{STMT } \text{STMTS}'$
 $\text{STMTS}' \rightarrow E \mid ; \text{STMTS}$
 $\text{STMT} \rightarrow \text{id} = \text{EXPR}$
 $\text{EXPR} \rightarrow \text{TERM } \text{EXPR}'$
 $\text{EXPR}' \rightarrow + \text{ TERM } \text{EXRPR}' \mid - \text{ TERM } \text{EXRPR}' \mid E$
 $\text{TERM} \rightarrow \text{POW } \text{TERM}'$
 $\text{TERM}' \rightarrow * \text{ POW } \text{TERM}' \mid / \text{ POW } \text{TERM}' \mid E$
 $\text{POW} \rightarrow \text{UNARY } \text{POW}'$
 $\text{POW}' \rightarrow ^\wedge \text{ POW} \mid E$
 $\text{UNARY} \rightarrow - \text{ UNARY} \mid + \text{ UNARY} \mid \text{FACTOR}$
 $\text{FACTOR} \rightarrow (\text{ EXPR }) \mid \text{id} \mid \text{INTEGER}$

- Table 1: the grammar after replacement it to letters and we assume the eleponline E in grammar is symbol (@).

$\text{PROGRAM} \rightarrow \text{STMTS}$	A->S
$\text{STMTS} \rightarrow \text{STMT } \text{STMTS}'$	S->TM
$\text{STMTS}' \rightarrow E \mid ; \text{STMTS}$	M->@ ;S
$\text{STMT} \rightarrow \text{id} = \text{EXPR}$	T->i=E
$\text{EXPR} \rightarrow \text{TERM } \text{EXPR}'$	E->RX
$\text{EXPR}' \rightarrow + \text{ TERM } \text{EXRPR}' \mid - \text{ TERM } \text{EXRPR}' \mid E$	X->+RX -RX @
$\text{TERM} \rightarrow \text{POW } \text{TERM}'$	R->PK
$\text{TERM}' \rightarrow * \text{ POW } \text{TERM}' \mid / \text{ POW } \text{TERM}' \mid E$	K->*PK /PK @
$\text{POW} \rightarrow \text{UNARY } \text{POW}'$	P->UW
$\text{POW}' \rightarrow ^\wedge \text{ POW} \mid E$	W->^P @
$\text{UNARY} \rightarrow - \text{ UNARY} \mid + \text{ UNARY} \mid \text{FACTOR}$	U->-U +U F
$\text{FACTOR} \rightarrow (\text{ EXPR }) \mid \text{id} \mid \text{INTEGER}$	F->(E) i n

2.3 Use the re-written grammar as a base to implement LL(1) parser

- Class LL1:

```
1 import re
2 import string
3 #Parser Class
4 ...
5 the grammar is:
6 PROGRAM → STMTS
7 STMTS → STMT STMTS'
8 STMTS' → E | ; STMTS
9 STMT → id = EXPR
10 EXPR → TERM EXPR'
11 EXPR' → + TERM EXPR' | - TERM EXPR' | E
12 TERM → POW TERM'
13 TERM' → * POW TERM' | / POW TERM' | E
14 POW → UNARY POW'
15 POW' → ^ POW | E
16 UNARY → - UNARY | + UNARY | FACTOR
17 FACTOR → ( EXPR ) | id | INTEGER
18
19 After replace to char
20 A->S
21 S->TM
22 M->@|;S
23 T->i=E
24 E->RX
```

```
25 X->+RX|-RX|@
26 R->PK
27 K->*PK|/PK|@
28 P->UW
29 W->^P|@
30 U->+U|+U|F
31 F->(E)|i|n
32 ...
33 class LL1Parser:
34     start_sym=""
35     productions={}
36     first_table= {}
37     follow_table= {}
38     table={}
39     def __init__(self,file='g3'):
40         grammar = open(file, "r")
41         self.Create_Productions(grammar)
42         for nt in self.productions:
43             self.first_table[nt] = self.Cal_first(nt)
44             self.follow_table[self.start_sym] = set('$')
45             for nt in self.productions:
46                 self.follow_table[nt] = self.Cal_follow(nt)
47             self.Create_Table()
```

```
49     #Check if sym is Non-Terminal
50     def isNonterminal(self,sym):
51         if sym.isupper():
52             return True
53         else:
54             return False
55     #Create Productions Dict From Given Grammar
56     def Create_Productions(self,grammar):
57         for production in grammar: # save the grammar in dictionary , the key is non term
58             #print(production)
59             lhs,rhs=re.split(">",production)
60             #print(lhs,rhs)
61             rhs = re.split("\|\|\n",rhs) # for every ninterm , split the rules
62             #print(rhs)
63             self.productions[lhs]=set(rhs)-{''}
64             if not self.start_sym:
65                 self.start_sym = lhs
66     #Calculate First Set Of Given Symbol
67     def Cal_first(self,sym):
68         if sym in self.first_table:
69             return self.first_table[sym];
70         if self.isNonterminal(sym):
71             first = set()
72             for x in self.productions[sym]:

```

```
73                 if x == '@':
74                     first = first.union('@')
75                 else:
76                     for i in x:
77                         fst = self.Cal_first(i)
78                         if i != x[-1]:
79                             first = first.union(fst-{ '@'})
80                         else:
81                             first = first.union(fst)
82                         if '@' not in fst:
83                             break
84             return first;
85         else:
86             return set(sym)
87     #Calculate Follow Set Of Given Symbol
88     def Cal_follow(self, sym):
89         if sym not in self.follow_table: # if it is not discovered yet
90             self.follow_table[sym] = set() # create a set for it to store the follow
91         for nt in self.productions.keys(): # bring all the nonterm and for each nonterm enters the loop
92             #print(nt,self.productions.keys())
93             #print('nt', nt)
94             #print('sym',sym)
95             for rule in self.productions[nt]: # bring all the production for a nonterm
96                 #print(self.productions[nt])
```

```

95     for rule in self.productions[nt]: # bring all the production for a nonterm
96         #print(self.productions[nt])
97         pos = rule.find(sym) # is the sym is exixt in any production
98         #print(pos,sym,rule)
99         if pos!=-1: # if yes
100             if pos == (len(rule)-1): # nonterm pos in the end of the production
101                 if nt != sym and nt in self.follow_table: #no recursion in the follow and no follow for undetrained nt yet
102                     #print(nt,sym)
103                     self.follow_table[sym] = self.follow_table[sym].union( self.follow_table[nt])
104             else: # if its not the end and comes after it non term bring its first
105                 #print(nt, sym)
106                 first_next = set()
107                 for next in rule[pos+1:]:
108                     fst_next = self.cal_first(next)
109                     first_next = first_next.union(fst_next-{ '@' })
110                     if '@' not in fst_next:
111                         break
112                     if '@' in fst_next:
113                         if nt != sym:
114                             self.follow_table[sym] = self.follow_table[sym].union(self.cal_follow(nt))
115                             self.follow_table[sym] = self.follow_table[sym].union(first_next) - { '@' }
116             else:
117                 self.follow_table[sym] = self.follow_table[sym].union(first_next)
118         return self.follow_table
119     #Create Parsing Table

```

```

119     #Create Parsing Table
120     def Create_Table(self):
121         for nt in self.productions:
122             for rule in self.productions[nt]:
123                 first_rule = set()
124                 for sym in rule:
125                     fst_sym = self.cal_first(sym)
126                     first_rule = first_rule.union(fst_sym-{ '@' })
127                     if '@' not in fst_sym:
128                         break
129                     if '@' in fst_sym:
130                         first_rule.add('@')
131                     for element in first_rule:
132                         self.table[nt,element] = rule
133                     if '@' in first_rule:
134                         for element in self.follow_table[nt]:
135                             self.table[nt,element] = rule
136     #Print First,Follow Sets and Parsing Table
137     def PrintDetails(self):
138         import pandas as pd
139         print("!! First Sets !!")
140         for nt in self.productions:
141             print(nt+": "+str(self.first_table[nt]))
142             print("\n")

```

```
143     print("!! Follow Sets !!")
144     for nt in self.productions:
145         print(nt+":"+str(self.follow_table[nt]))
146         print("\n")
147     print("!! Parsing Table !!")
148     temp_table={}
149     for key in self.table:
150         temp_table[key[1]]={}
151     for key in self.table:
152         temp_table[key[1]][key[0]] = (key[0]+"->"+self.table[key])
153     print(pd.DataFrame(temp_table).fillna('Error'))
154     print("\n")
155     #Parse A Given String According to grammar
156     def Parse(self,input):
157         print("input is ", input)
158         input=input+"$"
159         stack = []
160         f = 0
161         stack.append("$")
162         stack.append(self.start_sym)
163         input_length = len(input)
164         cur_index = 0
165         print(f'{Stack' : <20}{'Input' : ^20}{Action' : >20}')
166         while len(stack) > 0:
```

Reader Mode | ✓

```
167     TOS = stack[len(stack)-1]
168     current_input = input[cur_index]
169     if TOS == current_input and TOS == '$':
170         f=1
171         break
172     elif TOS == current_input:
173         print(f'{''.join(stack)) : <20}{input[cur_index:] : ^20}{Match' : >20}")
174         stack.pop()
175         cur_index+=1
176     else:
177         key = TOS,current_input
178         if key not in self.table:
179             f=0
180             break
181         rule=self.table[key]
182         action = str(TOS)+"->"+str(rule)
183         print(f'{''.join(stack)) : <20}{input[cur_index:] : ^20}{action : >20}")
184         if rule != '@':
185             rule = rule[::-1]
186             stack.pop()
187             for sym in list(rule):
188                 stack.append(sym)
189             else:
190                 stack.pop()
```

Reader Mode | ✓

```
191     if f == 0:
192         print("String Not Accepted!!")
193     else:
194         print("String Accepted!")
195
196     ##Main Function
197     if __name__ == "__main__":
198         parser = LL1Parser()
199         parser.PrintDetails()
200
201         parser.Parse("i=i+n")
202         parser.Parse("i=-n/n^n*n")
203         parser.Parse("i=-n;i=n+n")
204         parser.Parse("-(n)+n^i")
205         parser.Parse("i-i+i")
206         parser.Parse("-(i)=n")
```

- Class main:

```

e Edit Selection View Go Run Terminal Help
cod > main.py • LL1.py • main.py - cod2 - Visual Studio Code
cod > main.py ...
1  from tkinter import *
2  from tkinter.messagebox import *
3  import tkinter.filedialog as filedialog
4  import tkinter as tk
5  from tkinter import ttk
6  from LL1 import *
7  window = Tk()
8  window.title('Part(2)')
9  lbl = tk.Label(text='Welcome :)')
10 window.config(background='lightblue')
11 lbl.pack()
12
13 n2= Label(text = "Enter your input please ")
14 n2.pack(fill=X)
15
16 topFrame = Frame(window)
17 topFrame.pack()
18
19 no=Entry(window,width=60)
20 no.pack()
21
22 def button1():
23     parser = LL1Parser()
24     value=no.get()
25     parser.parse(value)
26
27 bt2=Button(window,text='Check the Input',width='12',command=button1)
28 bt2.place(x=150,y=300)
29 bt2.pack()
30
31 def buClick():
32     print(no.get())
33     no.delete(0,END)
34
35 window.mainloop()
36

```

Activate Windows
Go to Settings to activate Windows.

Ln 26, Col 5 Spaces: 4 UTF-8 CRLF Python 3.10.5 64-bit

Output in terminal:

```

!! First Sets !!
A:{'i'}
S:{'i'}
M:{';', '@'}
T:{'i'}
E:{'n', '+', '(' , 'i' , ')'}
X:{'@', '+', '-'}
R:{'n', '+', '(' , 'i' , ')'}
K:{'/', '@', '*'}
P:{'n', '+', '(' , 'i' , ')'}
W:{'^', '@'}
U:{'n', '+', '(' , 'i' , ')'}
F:{'i', 'n', '('}

```

```
!! Follow Sets !!
A:{'$'}
S:{'$'}
M:{'$'}
T:{';', '$'}
E:{';', '$', ')'}
X:{';', ')', '$'}
R:{'+', ')', '-+', ';', '$'}
K:{'+', ')', '-+', ';', '$'}
P:{'+', ')', '$', ';', '/', '-+', '*'}
W:{';', '+', ')', '$', '/', '-+', '*'}
U:{'+', ')', '$', '^', ';', '/', '-+', '*'}
F:{'+', ')', '$', '^', ';', '/', '-+', '*'}
```

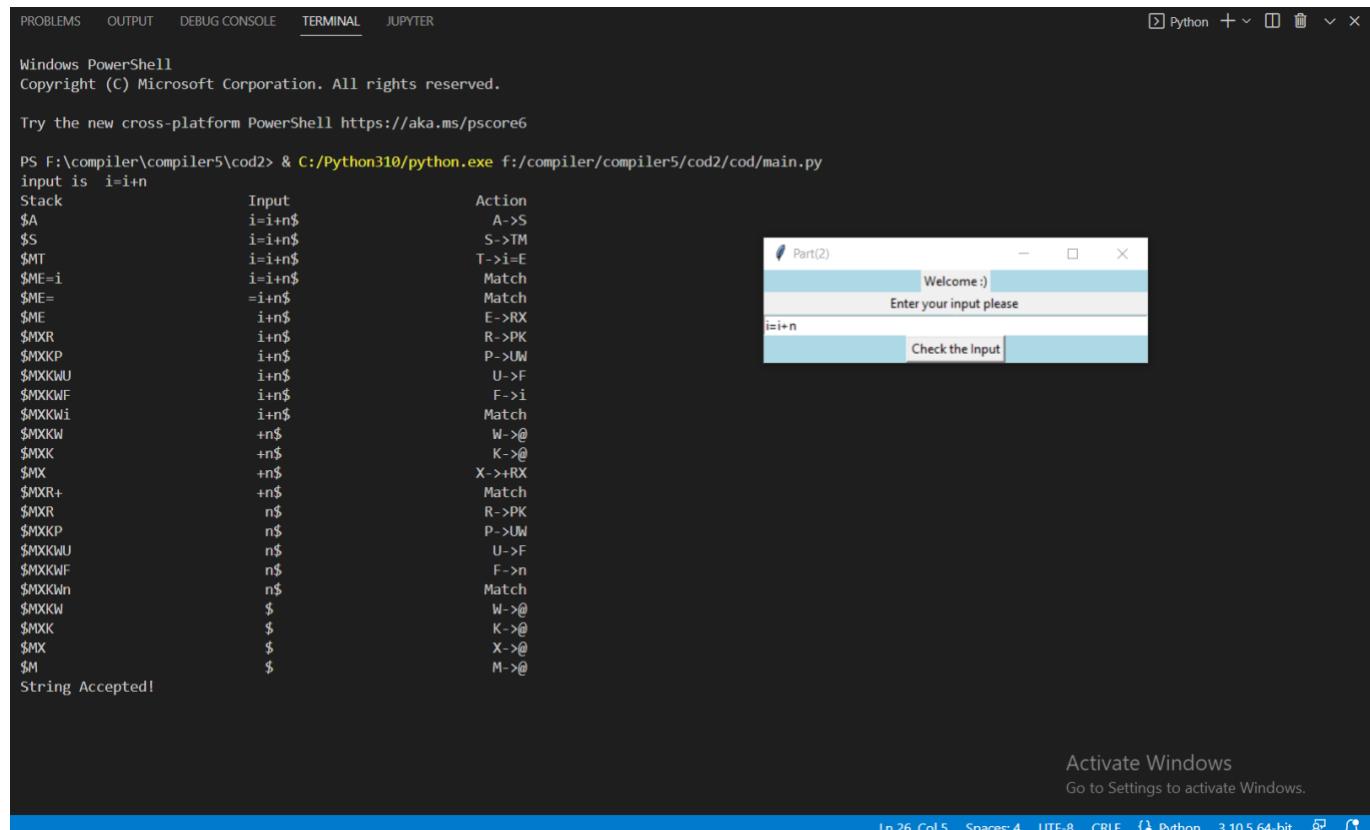
```
!! Parsing Table !!
      i   ;   @   $   n   ...   -   )   *   /   ^
A  A->S  Error  Error  Error  Error  ...  Error  Error  Error  Error  Error  Error
S  S->TM  Error  Error  Error  Error  ...  Error  Error  Error  Error  Error  Error
T  T->i=E  Error  Error  Error  Error  ...  Error  Error  Error  Error  Error  Error
E  E->RX  Error  Error  Error  E->RX  ...  E->RX  Error  Error  Error  Error  Error
R  R->PK  Error  Error  Error  R->PK  ...  R->PK  Error  Error  Error  Error  Error
P  P->UW  Error  Error  Error  P->UW  ...  P->UW  Error  Error  Error  Error  Error
U  U->F  Error  Error  Error  U->F  ...  U->U  Error  Error  Error  Error  Error
F  F->i  Error  Error  Error  F->n  ...  Error  Error  Error  Error  Error  Error
M  Error  M->S  M->@  M->@  Error  ...  Error  Error  Error  Error  Error  Error
X  Error  X->@  X->@  X->@  Error  ...  X->-RX  X->@  Error  Error  Error  Error
K  Error  K->@  K->@  K->@  Error  ...  K->@  K->@  K->*PK  K->/PK  Error
W  Error  W->@  W->@  W->@  Error  ...  W->@  W->@  W->@  W->@  W->^P
```

[12 rows x 12 columns]

2.4 Test your code with correct and incorrect input code

Valid input:

1) $i=i+n$



Stack	Input	Action
\$A	i=i+n\$	A->S
\$S	i=i+n\$	S->TM
\$MT	i=i+n\$	T->i=E
\$ME=i	i=i+n\$	Match
\$ME=	=i+n\$	Match
\$ME	i+n\$	E->RX
\$MXR	i+n\$	R->PK
\$MXKP	i+n\$	P->UW
\$MXKNU	i+n\$	U->F
\$MXKWF	i+n\$	F->i
\$MXKWi	i+n\$	Match
\$MXKW	+n\$	W->@
\$MXK	+n\$	K->@
\$MX	+n\$	X->+RX
\$MXR+	+n\$	Match
\$MXR	n\$	R->PK
\$MXKP	n\$	P->UW
\$MXKNU	n\$	U->F
\$MXKWF	n\$	F->n
\$MXKWi	n\$	Match
\$MXKW	\$	W->@
\$MXK	\$	K->@
\$MX	\$	X->@
\$M	\$	M->@

String Accepted!

Activate Windows
Go to Settings to activate Windows.

In 26, Col 5 Spaces: 4 UTF-8 CRLF { Python 3.10.5 64-bit

2) $i = n/n^n * n$

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

PS F:\compiler\compiler5\cod2> & C:/Python310/python.exe f:/compiler/compiler5/cod2/cod/main.py

```

input is i=n/n^n*n
Stack           Input          Action
$A              i=n/n^n*n$    A->S
$S              i=n/n^n*n$    S->TM
$MT             i=n/n^n*n$    T->i-E
$ME=i          i=n/n^n*n$    Match
$ME=          =n/n^n*n$    Match
$ME             -n/n^n*n$    E->RX
$MKR            -n/n^n*n$    R->PK
$MKKP           -n/n^n*n$    P->UW
$MKKNU          -n/n^n*n$    U->U
$MKKNU-         -n/n^n*n$    Match
$MKKNU          /n/n^n*n$    U->F
$MKKNUF         /n/n^n*n$    F->n
$MKKNUn        /n/n^n*n$    Match
$MKKKW          /n/n^n*n$    W->@
$MKKK            /n/n^n*n$    K->/PK
$MKKP/          /n/n^n*n$    P->UW
$MKKP            /n/n^n*n$    P->UW
$MKKKU          n/n^n*n$    U->F
$MKKKNU         n/n^n*n$    F->n
$MKKKWF         n/n^n*n$    Match
$MKKKWn        n/n^n*n$    W->@
$MKKK            *n$          K->*PK
$MKKK*          *n$          Match
$MKKP            *n$          P->UW
$MKKKU          n*n$n        U->F
$MKKKWF         n*n$n        F->n
$MKKKWn        n*n$n        Match
$MKKK            $             W->@
$MKKK*          $             K->@
$MX              $             X->@
```

Part(2)

Welcome :) Enter your input please

i=n/n^n*n Check the Input

Activate Windows
Go to Settings to activate Windows.

In 26, Col 5 Spaces:4 UTF-8 CRLF Python 3.10.5 64-bit

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

PS F:\compiler\compiler5\cod2> & C:/Python310/python.exe f:/compiler/compiler5/cod2/cod/main.py

```

$A              i=-n/n^n*n$    A->S
$S              i=-n/n^n*n$    S->TM
$MT             i=-n/n^n*n$    T->i-E
$ME=i          i=-n/n^n*n$    Match
$ME=          =-n/n^n*n$    Match
$ME             -n/n^n*n$    E->RX
$MKR            -n/n^n*n$    R->PK
$MKKP           -n/n^n*n$    P->UW
$MKKNU          -n/n^n*n$    U->U
$MKKNU-         -n/n^n*n$    Match
$MKKNU          /n/n^n*n$    U->F
$MKKNUF         /n/n^n*n$    F->n
$MKKNUn        /n/n^n*n$    Match
$MKKKW          /n/n^n*n$    W->@
$MKKK            /n/n^n*n$    K->/PK
$MKKP/          /n/n^n*n$    P->UW
$MKKP            /n/n^n*n$    P->UW
$MKKKU          n/n^n*n$    U->F
$MKKKNU         n/n^n*n$    F->n
$MKKKWF         n/n^n*n$    Match
$MKKKWn        n/n^n*n$    W->@
$MKKK            *n$          K->*PK
$MKKK*          *n$          Match
$MKKP            *n$          P->UW
$MKKKU          n*n$n        U->F
$MKKKWF         n*n$n        F->n
$MKKKWn        n*n$n        Match
$MKKK            $             W->@
$MKKK*          $             K->@
$MX              $             X->@
```

Part(2)

Welcome :) Enter your input please

i=-n/n^n*n Check the Input

Activate Windows
Go to Settings to activate Windows.

In 26, Col 5 Spaces:4 UTF-8 CRLF Python 3.10.5 64-bit

3) $i = -n; i = n + n$

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

PS F:\compiler\compiler5\cod2> & C:/Python310/python.exe f:/compiler/compiler5/cod2/cod/main.py

input is i=-n;i=n+n

Stack	Input	Action
\$A	i=-n;i=n+n\$	A->S
\$S	i=-n;i=n+n\$	S->TM
\$MT	i=-n;i=n+n\$	T-i=E
\$ME=i	i=-n;i=n+n\$	Match
\$ME=	=n;i=n+n\$	Match
\$ME	-n;i=n+n\$	E->RX
\$MXR	-n;i=n+n\$	R->PK
\$MKKP	-n;i=n+n\$	P->UN
\$MKKNU	-n;i=n+n\$	U->U
\$MKKNU-	-n;i=n+n\$	Match
\$MKKNU	n;i=n+n\$	U->F
\$MKKWF	n;i=n+n\$	F->n
\$MKKWh	n;i=n+n\$	Match
\$MKKWh	j;i=n+n\$	W->@
\$MKX	j;i=n+n\$	K->@
\$MX	j;i=n+n\$	X->@
\$M	j;i=n+n\$	M->;S
\$S;	j;i=n+n\$	Match
\$S	i=n+n\$	S->TM
\$MT	i=n+n\$	T-i=E
\$ME=i	i=n+n\$	Match
\$ME=	=n+n\$	Match
\$ME	n+n\$	E->RX
\$MXR	n+n\$	R->PK
\$MKKP	n+n\$	P->UN
\$MKKNU	n+n\$	U->F
\$MKKWF	n+n\$	F->n
\$MKKWh	n+n\$	Match
\$MKKWh	+n\$	W->@
\$MKX	+n\$	K->@
\$MX	+n\$	X->RX
\$MXR+	+n\$	Match
\$MXR	n\$	R->PK
\$MKKP	n\$	P->UN
\$MKKNU	n\$	U->F
\$MKKWF	n\$	F->n

Part(2) Welcome :) Enter your input please i=-n;i=n+n Check the Input

Activate Windows Go to Settings to activate Windows.

Go Run Terminal Help main.py - cod2 - Visual Studio Code

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

Python + - 🌐 📁

\$ME=	=-n;i=n+n\$	Match
\$ME	-n;i=n+n\$	E->RX
\$MXR	-n;i=n+n\$	R->PK
\$MXKP	-n;i=n+n\$	P->UW
\$MXKWU	-n;i=n+n\$	U->-U
\$MXGWU-	-n;i=n+n\$	Match
\$MXGWU	n;i=n+n\$	U->F
\$MXKWF	n;i=n+n\$	F->n
\$MXKWh	n;i=n+n\$	Match
\$MXK	;i=n+n\$	W->@
\$MXK	;i=n+n\$	K->@
\$MX	;i=n+n\$	X->@
\$M	;i=n+n\$	M->S
\$S;	;i=n+n\$	Match
\$S	i=n+n\$	S->TM
\$MT	i=n+n\$	T->i=E
\$ME=i	i=n+n\$	Match
\$ME=	=n+n\$	Match
\$ME	n+n\$	E->RX
\$MXR	n+n\$	R->PK
\$MXKP	n+n\$	P->UW
\$MXKWU	n+n\$	U->F
\$MXKWF	n+n\$	F->n
\$MXKWh	n+n\$	Match
\$MXK	+n\$	W->@
\$MXK	+n\$	K->@
\$MX	+n\$	X->RX
\$MXR+	+n\$	Match
\$MXR	n\$	R->PK
\$MXKP	n\$	P->UW
\$MXGWU	n\$	U->F
\$MXKWF	n\$	F->n
\$MXKWh	n\$	Match
\$MXK	\$	W->@
\$MXK	\$	K->@
\$MX	\$	X->@
\$M	\$	M->@

String Accepted!

Activate Windows
Go to Settings to activate Windows.

Invalid input:

1) -(n)+n^i

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER 

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell <https://aka.ms/pscore6>

```
PS F:\compiler\compiler5\cod2> & C:/Python310/python.exe f:/compiler/compiler5/cod2/cod/main.py
input is -(n)+n^i
Stack           Input           Action
String Not Accepted!!
```



The window has a title bar 'Part(2)'. Inside, there's a 'Welcome :)' message, an 'Enter your input please' input field containing '-(n)+n^i', and a 'Check the Input' button.

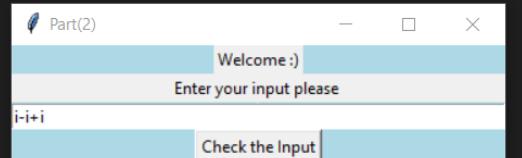
2) i-i+i

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell <https://aka.ms/pscore6>

```
PS F:\compiler\compiler5\cod2> & C:/Python310/python.exe f:/compiler/compiler5/cod2/cod/main.py
input is i-i+i
Stack           Input           Action
$A              i-i+i$         A->S
$S              i-i+i$         S->TM
$MT             i-i+i$         T->i=E
$ME=i          i-i+i$         Match
String Not Accepted!!
```



The window has a title bar 'Part(2)'. Inside, there's a 'Welcome :)' message, an 'Enter your input please' input field containing 'i-i+i', and a 'Check the Input' button.

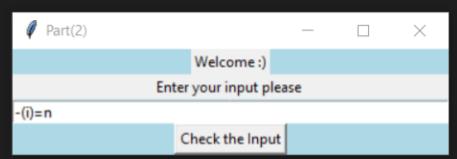
3) -(i)=n

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER 

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell <https://aka.ms/pscore6>

```
PS F:\compiler\compiler5\cod2> & C:/Python310/python.exe f:/compiler/compiler5/cod2/cod/main.py
input is -(i)=n
Stack           Input           Action
String Not Accepted!!
```



The window has a title bar 'Part(2)'. Inside, there's a 'Welcome :)' message, an 'Enter your input please' input field containing '-(i)=n', and a 'Check the Input' button.

References

- <https://github.com/priyatelj/thompsonconstruction/blob/master/Thompson.java>
- <https://medium.com/swlh/visualizing-thompsons-construction-algorithm-for-nfas-step-by-step-f92ef378581b>
- <https://www.geeksforgeeks.org/convert-infix-expression-to-postfix-expression/>
- https://github.com/pranshu-02/CD_LL1Parser
- https://pages.cs.wisc.edu/~fischer/cs536.s08/course.hold/html/NOTE_S/3.CFG.html
- https://github.com/alirezakay/RegexToDFA/tree/master/RegexToDfa_By_SyntaxTree/src/regextodfa