

```

Start here X LAB Exam Final Term.cpp X
1  #include <iostream>
2  #include <vector>
3  #include <set>
4  #include <map>
5  #include <queue>
6  #include <string>
7  #include <algorithm>
8
9  using namespace std;
10
11  const int STATES = 11;
12
13
14  const int SYMBOLS = 2;
15
16
17  const int MAX_TRANSITIONS = 5;
18
19  string inputSymbols[SYMBOLS] = {"a", "b"};
20
21
22  string epsilon = "e";
23
24  string transitions[STATES][SYMBOLS + 1][MAX_TRANSITIONS];
25
26
27  int transitionCount[STATES][SYMBOLS + 1];
28
29  void addTransition(int from, string symbol, string to) {
30
31      int index;
32
33

```

```

Start here X LAB Exam Final Term.cpp X
31
32      int index;
33
34
35      if (symbol == "a") index = 0;
36
37
38      else if (symbol == "b") index = 1;
39
40
41      else index = 2; // epsilon
42
43
44      int& count = transitionCount[from][index];
45
46
47      if (count < MAX_TRANSITIONS) {
48
49          transitions[from][index][count++] = to;
50
51
52      }
53
54
55
56  }
57
58  void setupENFA() {
59
60      addTransition(0, "e", "1");
61
62
63

```

Logs & others

C:\Sahanaj Mam\Final\LAB Exam Final Term.cpp

Start here X LAB Exam Final Term.cpp X

```
61     addTransition(0, "e", "1");
62
63
64     addTransition(0, "e", "7");
65
66
67     addTransition(1, "e", "2");
68
69
70     addTransition(1, "e", "4");
71
72
73     addTransition(3, "e", "6");
74
75
76     addTransition(5, "e", "6");
77
78
79     addTransition(6, "e", "1");
80
81
82     addTransition(6, "e", "7");
83
84
85     addTransition(2, "a", "3");
86
87
88     addTransition(4, "b", "5");
89
90
91     addTransition(7, "a", "8");
92
93
```

Start here X LAB Exam Final Term.cpp X

```
94     addTransition(8, "b", "9");
95
96
97     addTransition(9, "b", "10");
98
99
100 }
101
102 set<int> epsilonClosure(int state) {
103
104
105     set<int> closure;
106
107
108     queue<int> q;
109
110
111     closure.insert(state);
112
113
114     q.push(state);
115
116     while (!q.empty()) {
117
118
119         int current = q.front(); q.pop();
120
121
122         for (int i = 0; i < transitionCount[current][2]; ++i) {
123
124
125             int next = stoi(transitions[current][2][i]);
126
```

Start here X LAB Exam Final Term.cpp X

```
94     addTransition(8, "b", "9");
95
96
97     addTransition(9, "b", "10");
98
99
100 }
101
102 set<int> epsilonClosure(int state) {
103
104
105     set<int> closure;
106
107
108     queue<int> q;
109
110
111     closure.insert(state);
112
113
114     q.push(state);
115
116     while (!q.empty()) {
117
118
119         int current = q.front(); q.pop();
120
121
122         for (int i = 0; i < transitionCount[current][2]; ++i) {
123
124
125             int next = stoi(transitions[current][2][i]);
```

Start here X LAB Exam Final Term.cpp X

```
124
125         int next = stoi(transitions[current][2][i]);
126
127
128         if (closure.find(next) == closure.end()) {
129
130
131             closure.insert(next);
132
133
134             q.push(next);
135
136
137         }
138
139     }
140
141
142 }
143
144
145
146 return closure;
147
148 }
149
150
151 set<int> epsilonClosure(const set<int>& states) {
152
153
154     set<int> result;
```

```

Start here X LAB Exam Final Term.cpp X
154     set<int> result;
155
156
157     for (int s : states) {
158
159
160         set<int> closure = epsilonClosure(s);
161
162
163         result.insert(closure.begin(), closure.end());
164
165
166     }
167
168
169     return result;
170
171 }
172
173
174 set<int> move(const set<int>& states, int symbolIndex) {
175
176
177     set<int> result;
178
179
180     for (int state : states) {
181
182
183         for (int i = 0; i < transitionCount[state][symbolIndex]; ++i) {
184
185
186             int next = stoi(transitions[state][symbolIndex][i]);

```

```

319         string target = dfaTransitionTable[name][inputSymbols[i]];
320
321         cout << (target.empty() ? "-" : target) << "\t";
322
323     }
324
325     cout << stateSetToString(state) << endl;
326
327 }
328
329 int main() {
330     setupENFA();
331     convertToDFA();
332     return 0;
333 }

```

Start here X: LAB Exam Final Term.cpp X

```
313
314     cout << name << "\t";
315
316
317     for (int i = 0; i < SYMBOLS; ++i) {
318
319         string target = dfaTransitionTable[name][inputSymbols[i]];
320
321         cout << (target.empty() ? "-" : target) << "\t";
322
323
324     }
325
326     cout << stateSetToString(ata) << endl;
327
328 }
329
330
331
332
333
334
335
336
337 int main() {
338
339     setupENFA();
340
341     convertToDFA();
342
343
344
345 }
```

```

280         )
281     }
282     dfaTransitionTable[currentName][inputSymbols[i]] = stateNames[nextClosure];
283
284 }
285
286 // Print DFA Table
287
288 cout << "\nDFA Transition Table:\n";
289
290 cout << "State\t";
291
292 for (int i = 0; i < SYMBOLS; ++i)
293
294     cout << inputSymbols[i] << "\t";
295
296 cout << "\nDFA States\n";
297
298 for (const auto& state : dfaStates) {
299     string name = stateNames[state];
300
301 }
302
303 }
304
305 }
306
307 }
308
309 }
310
311 }
312

```

```

250
251 while (!q.empty()) {
252
253     set<int> current = q.front(); q.pop();
254
255     string currentName = stateNames[current];
256
257     for (int i = 0; i < SYMBOLS; ++i) {
258
259         set<int> nextMove = move(current, i);
260
261         set<int> nextClosure = epsilonClosure(nextMove);
262
263         if (nextClosure.empty()) continue;
264
265         if (stateNames.find(nextClosure) == stateNames.end()) {
266
267             stateNames[nextClosure] = "D" + to_string(stateId++);
268
269             dfaStates.push_back(nextClosure);
270
271             q.push(nextClosure);
272
273         }
274     }
275 }
276
277 }
278
279 }
280
281 }
282

```

Start here X LAB Exam Final Term.cpp X

```
217     return res;
218
219
220
221 }
222
223 void convertToDFA() {
224
225     map<set<int>, string> stateNames;
226
227     vector<set<int>> dfaStates;
228
229     map<string, map<string, string>> dfaTransitionTable;
230
231     int stateId = 0;
232
233     set<int> start = epsilonClosure(0);
234
235     queue<set<int>> q;
236
237     q.push(start);
238
239     stateNames[start] = "D" + to_string(stateId++);
240
241     dfaStates.push back(start);
```

Start here X LAB Exam Final Term.cpp X

```
187
188     result.insert(next);
189
190
191 }
192
193
194
195 }
196
197     return result;
198
199
200
201 }
202
203 string stateSetToString(const set<int>& s) {
204
205     string res = "{";
206
207     for (int state : s) res += to_string(state) + ",";
208
209     if (!s.empty()) res.pop_back();
210
211     res += "}";
212
213     return res;
214
215
216
217
218
219 }
```



"D:\Semestar-7\CD(Sahanaj N



DFA Transition Table:

State	a	b	NFA States
D0	D1	D2	{0,1,2,4,7}
D1	D1	D3	{1,2,3,4,6,7,8}
D2	D1	D2	{1,2,4,5,6,7}
D3	D1	D4	{1,2,4,5,6,7,9}
D4	D1	D2	{1,2,4,5,6,7,10}

Process returned 0 (0x0) execution time : 0.102 s
Press any key to continue.

|