

National College of Computer Studies

Paknajol, Kathmandu

**Lab Report 4**

**Submitted by: Submitted to:**

Atullya Maharjan Prashant Gautam

BSc. CSIT 4th Semester NCCS Roll. No: 5

# WAP to design a NFA that accepts string ending with 01.

#include <iostream> #include <vector> using namespace std;

vector<int> states = {0, 1, 2}; vector<vector<pair<char, int>>> transitions = {

{{'0', 0}, {'1', 0}, {'0', 1}},

{{'1', 2}},

{{}}};

bool simulate\_nfa(string input)

{

vector<int> current\_states = {0}; for (char c : input)

{

vector<int> next\_states;

for (int state : current\_states)

{

for (auto transition : transitions[state])

{

if (transition.first == c)

{

next\_states.push\_back(transition.second);

}

}

}

if (next\_states.empty())

{

return false;

}

current\_states = next\_states;

}

for (int state : current\_states)

{

if (state == 2)

{

return true;

}

}

return false;

}

int main()

{

string input;

cout << "Enter a string to check: "; cin >> input;

if (simulate\_nfa(input))

{

cout << "String ends with 01." << endl;

}

else

{

cout << "String does not end with 01." << endl;

}

return 0;

}

OUTPUT



# WAP to design a NFA that accepts string containing substring 101

#include <iostream> #include <vector>

using namespace std;

vector<int> states = {0, 1, 2, 3}; vector<vector<pair<char, int>>> transitions = {

{{'0', 0}, {'1', 0}, {'1', 1}},

{{'0', 2}},

{{'1', 3}},

{{'0', 3}, {'1', 3}}};

bool simulate\_nfa(string input)

{

vector<int> current\_states = {0};

for (char c : input)

{

vector<int> next\_states;

for (int state : current\_states)

{

for (auto transition : transitions[state])

{

if (transition.first == c)

{

next\_states.push\_back(transition.second);

}

}

}

if (next\_states.empty())

{

return false;

}

current\_states = next\_states;

}

for (int state : current\_states)

{

if (state == 3)

{

return true;

}

}

return false;

}

int main()

{

string input;

cout << "Enter a string to check: "; cin >> input;

if (simulate\_nfa(input))

{

cout << "String contains substring 101." << endl;

}

else

{

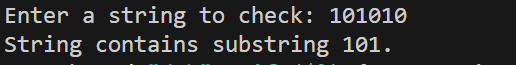
cout << "String does not contain substring 101." << endl;

}

return 0;

}

OUTPUT



# WAP to design a NFA that accepts string starting with 10

#include <iostream> #include <vector>

using namespace std;

vector<int> states = {0, 1, 2}; vector<vector<pair<char, int>>> transitions = {

{{'1', 1}},

{{'0', 2}},

{{'0', 2}, {'1', 2}},

{{}}};

bool simulate\_nfa(string input)

{

vector<int> current\_states = {0};

for (char c : input)

{

vector<int> next\_states;

for (int state : current\_states)

{

for (auto transition : transitions[state])

{

if (transition.first == c)

{

next\_states.push\_back(transition.second);

}

}

}

if (next\_states.empty())

{

return false;

}

current\_states = next\_states;

}

for (int state : current\_states)

{

if (state == 2)

{

return true;

}

}

return false;

}

int main()

{

string input;

cout << "Enter a string to check: "; cin >> input;

if (simulate\_nfa(input))

{

cout << "String starts with 10." << endl;

}

else

{

cout << "String does not start with 10." << endl;

}

return 0;

}

OUTPUT



69