(//www.sanfoundry.com)

Home (http://www.sanfoundry.com) About (http://www.sanfoundry.com/about/)

Bangalore Training (http://www.sanfoundry.com/coursesfees/)

Online Training (http://www.sanfoundry.com/online-training-san-linux-kernel-device-drivers/)

Developers (http://www.sanfoundry.com/salary-50l/) Jobs (http://www.sanfoundry.com/jobs/)

Internship (http://www.sanfoundry.com/internship/) Contact (http://www.sanfoundry.com/contact/)

Search …

Questions & Answers

C Interview Questions (http://www.sanfoundry.com/c-interview-questions-answers/)
C++ Questions (http://www.sanfoundry.com/cplusplus-interview-questions-answers/)
Linux MCQs (http://www.sanfoundry.com/technical-interview-questions/)
C# Quiz (http://www.sanfoundry.com/csharp-questions-answers/)
Java MCQs (http://www.sanfoundry.com/java-questions-answers-freshers-experienced/)
JavaScript MCQs (http://www.sanfoundry.com/1000-javascript-questions-answers/)
SAN Questions (http://www.sanfoundry.com/san-storage-mcqs-freshers-experienced/)
PHP Questions (http://www.sanfoundry.com/php-questions-answers/)
Python Quiz (http://www.sanfoundry.com/1000-python-questions-answers/)

Computer Science Questions

Operating System Quiz (http://www.sanfoundry.com/operating-system-questions-answers/)
Computer Architecture MCQs (http://www.sanfoundry.com/1000-computer-organization-architecture-questions-answers/)
Software Architecture MCQs (http://www.sanfoundry.com/software-architecture-design-questions-answers/)
Software Engineering MCQs (http://www.sanfoundry.com/software-engineering-questions-answers/)
Artificial Intelligence MCQs (http://www.sanfoundry.com

« Prev Page (http://www.sanfoundry.com/cpp-program-implement-levenshtein-distance-computing-algorithm/)
Next Page (http://www.sanfoundry.com/cpp-program-perform-cryptography-using-transposition-technique/) »

C++ Program to Construct DFA from NFA

This is a C++ Program to convert NFA to DFA. A DFA (Deterministic Finite Automaton) is a finite state machine where from each state and a given input symbol, the next possible state is uniquely determined. On the other hand, an NFA (Non-Deterministic Finite Automaton) can move to several possible next states from a given state and a given input symbol. However, this does not add any more power to the machine. It still accepts the same set of languages, namely the regular languages. It is possible to convert an NFA to an equivalent DFA using the powerset construction.
The intuition behind this scheme is that an NFA can be in several possible states at any time. We can simulate it with a DFA whose states correspond to sets of states of the underlying NFA.

Here is source code of the C++ Program to Construct DFA from NFA. The C++ program is successfully compiled and run on a Linux system. The program output is also shown below.

```
#include <cstdio>
#include <fstream>
#include <iostream>
#include <bitset>
#include <vector>
#include <cstring>
#include <cstdlib>
#include <algorithm>
#include <queue>
#include <set>
#define MAX_NFA_STATES 10
#define MAX_ALPHABET_SIZE 10
using namespace std;
// Representation of an NFA state
class NFAstate
{
  public:
    int transitions[MAX_ALPHABET_SIZE][MAX_NFA_STATES];
    NFAstate()
    {
      for (int i = 0; i < MAX_ALPHABET_SIZE; i++)
        for (int j = 0; j < MAX_NFA_STATES; j++)
          transitions[i][j] = -1;
    }
}*NFAstates;
// Representation of a DFA state
struct DFAstate
{
    bool finalState;
```

/artificial-intelligence-questions-answers/)
LISP Programming MCQs (http://www.sanfoundry.com/lisp-programming-questions-answers/)
Database Management MCQs (http://www.sanfoundry.com/1000-database-management-system-questions-answers/)
Computer Network MCQs (http://www.sanfoundry.com/computer-network-questions-answers/)
Microprocessor MCQs (http://www.sanfoundry.com/microprocessors-questions-answers/)

C Programming Examples

Simple C Programs (http://www.sanfoundry.com/simple-c-programs/)
C - Arrays (http://www.sanfoundry.com/c-programming-examples-arrays/)
C - Matrix (http://www.sanfoundry.com/c-programming-examples-matrix/)
C - Strings (http://www.sanfoundry.com/c-programming-examples-strings/)
C - Bitwise Operations (http://www.sanfoundry.com/c-programming-examples-bitwise-operations/)
C - Linked Lists (http://www.sanfoundry.com/c-programming-examples-linked-list/)
C - Stacks & Queues (http://www.sanfoundry.com/c-programming-examples-stacks/)
C - Searching & Sorting (http://www.sanfoundry.com/c-programming-examples-searching-sorting/)
C - Trees (http://www.sanfoundry.com/c-programming-examples-on-trees/)
C - Strings (http://www.sanfoundry.com/c-programming-examples-strings/)
C - File Handling (http://www.sanfoundry.com/c-programming-examples-file-handling/)
C - Mathematical Functions (http://www.sanfoundry.com/c-programming-examples-mathematical-functions/)
C - Puzzles & Games (http://www.sanfoundry.com/c-programming-examples-on-puzzles-games/)
C Programs - Recursion (http://www.sanfoundry.com/c-programming-examples-recursion/)
C Programs - No Recursion

```
    bitset<MAX_NFA_STATES> constituentNFAstates;
    bitset<MAX_NFA_STATES> transitions[MAX_ALPHABET_SIZE];
    int symbolicTransitions[MAX_ALPHABET_SIZE];
};
set<int> NFA_finalStates;
vector<int> DFA_finalStates;
vector<DFAstate*> DFAstates;
queue<int> incompleteDFAstates;
int N, M; // N -> No. of stattes, M -> Size of input alphabet
// finds the epsilon closure of the NFA state "state" and stores it into "cl
void epsilonClosure(int state, bitset<MAX_NFA_STATES> &closure)
{
    for (int i = 0; i < N && NFAstates[state].transitions[0][i] != -1; i++)
        if (closure[NFAstates[state].transitions[0][i]] == 0)
        {
            closure[NFAstates[state].transitions[0][i]] = 1;
            epsilonClosure(NFAstates[state].transitions[0][i], closure);
        }
}
// finds the epsilon closure of a set of NFA states "state" and stores it in
void epsilonClosure(bitset<MAX_NFA_STATES> state,
        bitset<MAX_NFA_STATES> &closure)
{
    for (int i = 0; i < N; i++)
        if (state[i] == 1)
            epsilonClosure(i, closure);
}
// returns a bitset representing the set of states the NFA could be in aft
// from state X on input symbol A
void NFAmove(int X, int A, bitset<MAX_NFA_STATES> &Y)
{
    for (int i = 0; i < N && NFAstates[X].transitions[A][i] != -1; i++)
        Y[NFAstates[X].transitions[A][i]] = 1;
}
// returns a bitset representing the set of states the NFA could be in aft
// from the set of states X on input symbol A
void NFAmove(bitset<MAX_NFA_STATES> X, int A, bitset<MAX_NFA_STAT
{
    for (int i = 0; i < N; i++)
        if (X[i] == 1)
            NFAmove(i, A, Y);
}
int main()
{
    int i, j, X, Y, A, T, F, D;
    // read in the underlying NFA
    ifstream fin("NFA.txt");
    fin >> N >> M;
    NFAstates = new NFAstate[N];
    fin >> F;
    for (i = 0; i < F; i++)
    {
        fin >> X;
        NFA_finalStates.insert(X);
    }
    fin >> T;
    while (T--)
    {
        fin >> X >> A >> Y;
        for (i = 0; i < Y; i++)
        {
            fin >> j;
            NFAstates[X].transitions[A][i] = j;
        }
    }
    fin.close();
    // construct the corresponding DFA
    D = 1;
    DFAstates.push_back(new DFAstate);
    DFAstates[0]->constituentNFAstates[0] = 1;
    epsilonClosure(0, DFAstates[0]->constituentNFAstates);
    for (j = 0; j < N; j++)
        if (DFAstates[0]->constituentNFAstates[j] == 1 && NFA_finalStates.f
                j) != NFA_finalStates.end())
        {
            DFAstates[0]->finalState = true;
            DFA_finalStates.push_back(0);
```

(http://www.sanfoundry.com/c-programming-examples-without-using-recursion/)

Java Algorithms

Java - Numerical Problems (http://www.sanfoundry.com/java-programming-examples-numerical-problems-algorithms/)
Java - Combinatorial Problems (http://www.sanfoundry.com/java-programming-examples-combinatorial-problems-algorithms/)
Java - Graph Problems (http://www.sanfoundry.com/java-programming-examples-graph-problems-algorithms/)
Java - Hard Graph Problems (http://www.sanfoundry.com/java-programming-examples-hard-graph-problems-algorithms/)
Java - Computation Geometry (http://www.sanfoundry.com/java-programming-examples-computational-geometry-problems-algorithms/)
Java - Sets & Strings (http://www.sanfoundry.com/java-programming-examples-set-string-problems-algorithms/)
Java - Data-Structures (http://www.sanfoundry.com/java-programming-examples-data-structures/)
Java - Collection API Problems (http://www.sanfoundry.com/java-programming-examples-collection-api/)

C++ Algorithms

C++ - Numerical Problems (http://www.sanfoundry.com/cpp-programming-examples-numerical-problems-algorithms/)
C++ - Combinatorial Problems (http://www.sanfoundry.com/cpp-programming-examples-combinatorial-problems-algorithms/)
C++ - Graph Problems (http://www.sanfoundry.com/cpp-programming-examples-graph-problems-algorithms/)
C++ - Hard Graph Problems (http://www.sanfoundry.com/cpp-programming-examples-hard-graph-problems-algorithms/)
C++ - Computation Geometry (http://www.sanfoundry.com/cpp-programming-examples-computational-geometry-problems-algorithms/)
C++ - Sets & Strings (http://www.sanfoundry.com/cpp-programming-examples-set-string-problems-algorithms/)
C++ - Data-Structures (http://www.sanfoundry.com/cpp-programming-examples-data-structures/)
C++ - STL Library (http://www.sanfoundry.com/cpp-programming-examples-stl/)

```
            break;
        }
    incompleteDFAstates.push(0);
    while (!incompleteDFAstates.empty())
    {
        X = incompleteDFAstates.front();
        incompleteDFAstates.pop();
        for (i = 1; i <= M; i++)
        {
            NFAmove(DFAstates[X]->constituentNFAstates, i,
                    DFAstates[X]->transitions[i]);
            epsilonClosure(DFAstates[X]->transitions[i],
                    DFAstates[X]->transitions[i]);
            for (j = 0; j < D; j++)
                if (DFAstates[X]->transitions[i]
                        == DFAstates[j]->constituentNFAstates)
                {
                    DFAstates[X]->symbolicTransitions[i] = j;
                    break;
                }
            if (j == D)
            {
                DFAstates[X]->symbolicTransitions[i] = D;
                DFAstates.push_back(new DFAstate);
                DFAstates[D]->constituentNFAstates
                        = DFAstates[X]->transitions[i];
                for (j = 0; j < N; j++)
                    if (DFAstates[D]->constituentNFAstates[j] == 1
                            && NFA_finalStates.find(j) != NFA_finalStates.end())
                    {
                        DFAstates[D]->finalState = true;
                        DFA_finalStates.push_back(D);
                        break;
                    }
                incompleteDFAstates.push(D);
                D++;
            }
        }
    }
    // write out the corresponding DFA
    ofstream fout("DFA.txt");
    fout << D << " " << M << "\n" << DFA_finalStates.size();
    for (vector<int>::iterator it = DFA_finalStates.begin(); it
            != DFA_finalStates.end(); it++)
        fout << " " << *it;
    fout << "\n";
    for (i = 0; i < D; i++)
    {
        for (j = 1; j <= M; j++)
            fout << i << " " << j << " "
                    << DFAstates[i]->symbolicTransitions[j] << "\n";
    }
    fout.close();
    return 0;
}
```

Output:

C Algorithms

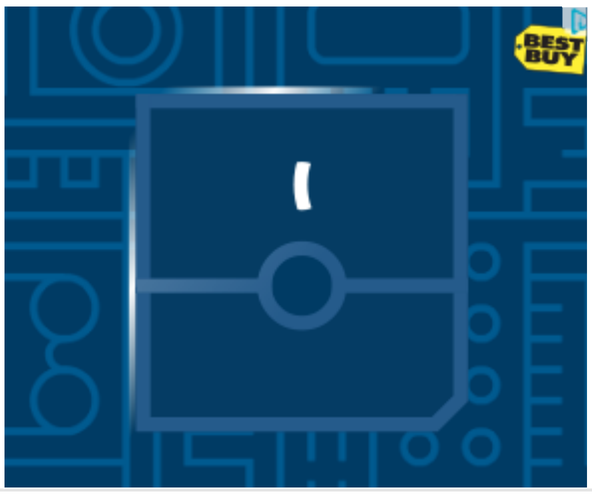
C - Numerical Problems (http://www.sanfoundry.com/c-programming-examples-numerical-problems-algorithms/)
C - Combinatorial Problems (http://www.sanfoundry.com/c-programming-examples-combinatorial-problems-algorithms/)
C - Graph Problems (http://www.sanfoundry.com/c-programming-examples-graph-problems-algorithms/)
C - Hard Graph Problems (http://www.sanfoundry.com/c-programming-examples-hard-graph-problems-algorithms/)
C - Computation Geometry (http://www.sanfoundry.com/c-programming-examples-computational-geometry-problems-algorithms/)
C - Sets & Strings (http://www.sanfoundry.com/c-programming-examples-set-string-problems-algorithms/)
C - Data-Structures (http://www.sanfoundry.com/c-programming-examples-data-structures/)

```
$ g++ NFAtoDFA.cpp
$ a.out

Input file
NFA.txt
4 2
2 0 1
4
0 1 2 1 2
1 1 2 1 2
2 2 2 1 3
3 1 2 1 2

Output file
DFA.txt
4 2
3 0 1 3
0 1 1
0 2 2
1 1 1
1 2 3
2 1 2
2 2 2
3 1 1
3 2 2

------------------
(program exited with code: 0)
Press return to continue
```

Sanfoundry Global Education & Learning Series – 1000 C++ Programs.

Here’s the list of Best Reference Books in C++ Programming, Data Structures and Algorithms. (http://www.sanfoundry.com/best-reference-books-c-plusplus-programming-data-structures-algorithms/)

« Prev Page - C++ Program to Implement Levenshtein Distance Computing Algorithm (http://www.sanfoundry.com/cpp-program-implement-levenshtein-distance-computing-algorithm/)
» Next Page - C++ Program to Perform Cryptography Using Transposition Technique (http://www.sanfoundry.com/cpp-program-perform-cryptography-using-transposition-technique/)



Deep Dive @ Sanfoundry:

1. **Automata Theory Questions and Answers (http://www.sanfoundry.com/1000-automata-theory-questions-answers/)**
2. **Java Program to Construct an Expression Tree for an Infix Expression (http://www.sanfoundry.com/java-program-construct-expression-tree-infix-expression/)**
3. **C++ Program to Construct an Expression Tree for a Postfix Expression (http://www.sanfoundry.com/cpp-program-construct-expression-tree-postfix-expression/)**
4. **Automata Theory Questions and Answers – Mealy Machine-II (http://www.sanfoundry.com/automata-theory-questions-answers-mealy-machine-1/)**
5. **Automata Theory Questions and Answers – Epsilon**

Closures (http://www.sanfoundry.com/automata-theory-questions-answers-epsilon-closures/)

6. **C++ Program to Illustrate the usage of bitset (http://www.sanfoundry.com/cpp-program-illustrate-bitset-usage/)**
7. **C++ Program to Construct Transitive Closure Using Warshall's Algorithm (http://www.sanfoundry.com/cpp-program-construct-transitive-closure-using-warshalls-algorithm/)**
8. **Compilers Questions and Answers – The NFA with epsilon- Moves-2 (http://www.sanfoundry.com/compilers-questions-answers-experienced/)**
9. **Automata Theory Questions and Answers – Uses of Epsilon-Transitions (http://www.sanfoundry.com/automata-theory-questions-answers-uses-epsilon-transitions/)**
10. **Theory of Computation – Nondeterministic Finite Automata with Null Transitions (http://www.sanfoundry.com/automata-theory-nfa-null-transitions/)**

Manish Bhojasia (http://www.sanfoundry.com/about/), a technology veteran with 20+ years @ Cisco & Wipro, is Founder and CTO at Sanfoundry. He is Linux Kernel Developer and SAN Architect and is passionate about competency developments in these areas. He lives in Bangalore and delivers focused training sessions to IT professionals in Linux Kernel, Linux Debugging, Linux Device Drivers, Linux Networking, Linux Storage & Cluster Administration, Advanced C Programming, SAN Storage Technologies, SCSI Internals and Storage Protocols such as iSCSI & Fiber Channel. Stay connected with him below:
LinkedIn (http://www.linkedin.com/in/manishbhojasia) | Facebook (http://www.facebook.com/sanfoundry) | Twitter (http://www.twitter.com/sanfoundry) | Google+ (https://plus.google.com/104408026570656234343/posts)

Subscribe Sanfoundry Newsletter and Posts

Name*

Email*

Subscribe

Best Careers

Developer Tracks (http://www.sanfoundry.com/salary-50l/)
SAN Developer (http://www.sanfoundry.com/san-storage-developer-training-courses/)
Linux Kernel Developer (http://www.sanfoundry.com/linux-kernel-developer-training-courses-jobs/)
Linux Driver Developer (http://www.sanfoundry.com/linux-device-driver-developer-training/)
Linux Network Developer (http://www.sanfoundry.com/linux-network-developer-training/)

Live Training Photos (http://www.sanfoundry.com/sanfoundry-classes/)
Mentoring (http://www.sanfoundry.com/professional-mentoring-coaching-career-guidance-cto/)
Software Productivity (http://www.sanfoundry.com/programming-discipline-and-software/)
GDB Assignment (http://www.sanfoundry.com/gdb-example-tutorial/)

*Sanfoundry is **No. 1** choice for Deep Hands-ON Trainings in **SAN, Linux & C, Kernel Programming**. Our Founder has trained employees of almost all Top Companies in India such as VMware, Citrix, Oracle, Motorola, Ericsson, Aricent, HP, Intuit, Microsoft, Cisco, SAP Labs, Siemens, Symantec, Redhat, Chelsio, Cavium, ST-Micro, Samsung, LG-Soft, Wipro, TCS, HCL, IBM, Accenture, HSBC, Mphasis, Tata-Elxsi, Tata VSNL, Mindtree, Cognizant and Startups.*

Best Trainings

- SAN I - Technology (http://www.sanfoundry.com/san-storage-area-networks-training/)
- SAN II - Admin (http://www.sanfoundry.com/san-administration-training-course/)
- Linux Fundamentals (http://www.sanfoundry.com/linux-administration-training/)
- Advanced C Training (http://www.sanfoundry.com/advanced-c-programming-training/)
- Linux-C Debugging (http://www.sanfoundry.com/training-on-linux-debugging-techniques/)
- System Programming (http://www.sanfoundry.com/training-on-linux-internals-systems/)
- Network Programming (http://www.sanfoundry.com/training-socket-network-programming/)
- Linux Threads (http://www.sanfoundry.com/training-multithreaded-parallel/)
- Kernel Programming (http://www.sanfoundry.com/linux-kernel-internals-training/)
- Kernel Debugging (http://www.sanfoundry.com/linux-kernel-debugging-training/)
- Linux Device Drivers (http://www.sanfoundry.com/training-on-linux-device-drivers/)

Testimonials

It was a wonderful course, a thorough study through the bits and pieces of Storage Devices and the other advancements in the SAN technology which was very informative for me. Well prepared slides, practical thoughts and fully interactive class.
— Milind - Microsoft

The depth of C coverage, the common pitfalls, the presentation by Manish and his pace were all remarkable. Gave a clear picture of the inner working of C language design.
— Shivanshu - Netapp

The course was well designed and coverage was satisfactorily deep. The linux programming assignments were designed to get a feel of real time issues and covered lot of areas of uncertainty. The project work suggested at the end of the course was invaluable.
— Somenath - EMC

Best Reference Books

Computer Science Books (http://www.sanfoundry.com/best-reference-books-computer-science-engineering/)
Algorithm & Programming Books (http://www.sanfoundry.com/best-reference-books-programming-hot-technologies)
Electronics Engineering Books (http://www.sanfoundry.com/best-reference-books-electrical-electronics-engineering/)
Electrical Engineering Books (http://www.sanfoundry.com/best-reference-books-electrical-electronics-engineering/)
Chemical Engineering Books (http://www.sanfoundry.com/best-reference-books-chemical-engineering/)
Civil Engineering Books (http://www.sanfoundry.com/best-

Questions and Answers

1000 C Questions & Answers (http://www.sanfoundry.com/c-interview-questions-answers/)
1000 C++ Questions & Answers (http://www.sanfoundry.com/cplusplus-interview-questions-answers/)
1000 C# Questions & Answers (http://www.sanfoundry.com/1000-csharp-questions-answers/)
1000 Java Questions & Answers (http://www.sanfoundry.com/java-questions-answers-freshers-experienced/)
1000 Linux Questions & Answers (http://www.sanfoundry.com/technical-interview-questions/)
1000 Python Questions (http://www.sanfoundry.com/1000-python-questions-answers/)

India Internships

Computer Science Internships (http://www.sanfoundry.com/internships-computer-science-engineering/)
Instrumentation Internships (http://www.sanfoundry.com/internships-instrumentation-engineering/)
Electronics Internships (http://www.sanfoundry.com/internships-electronics-electrical-engineering/)
Electrical Internships (http://www.sanfoundry.com/internships-electronics-electrical-engineering/)
Mechanical Internships (http://www.sanfoundry.com/internships-mechanical-engineering/)
Industrial Internships (http://www.sanfoundry.com

About Sanfoundry

About Us (http://www.sanfoundry.com/about/)
Copyright (http://www.sanfoundry.com/copyright/)
TOS & Privacy (http://www.sanfoundry.com/tos-privacy-policy/)
Jobs (http://www.sanfoundry.com/jobs/)
Bangalore Training (http://www.sanfoundry.com/coursesfees/)
Online Training (http://www.sanfoundry.com/online-training-san-linux-kernel-device-drivers/)
SAN Training (http://www.sanfoundry.com/san-storage-area-networks-training/)

reference-books-civil-engineering/)
Mechanical Engineering Books (http://www.sanfoundry.com/best-reference-books-mechanical-engineering/)
Industrial Engineering Books (http://www.sanfoundry.com/best-reference-books-industrial-engineering-operations-research/)
Instrumentation Engg Books (http://www.sanfoundry.com/best-reference-books-instrumentation-engineering/)
Metallurgical Engineering Books (http://www.sanfoundry.com/best-reference-books-metallurgical-engineering/)
All Stream Best Books (http://www.sanfoundry.com/best-reference-books-tech-engineering-sciences/)

1000 PHP Questions & Answers (http://www.sanfoundry.com/1000-php-questions-answers/)
1000 Hadoop Questions (http://www.sanfoundry.com/1000-hadoop-questions-answers/)
Cloud Computing Questions (http://www.sanfoundry.com/1000-cloud-computing-questions-answers/)
Computer Science Questions (http://www.sanfoundry.com/computer-science-questions-answers/)
All Stream Questions & Answers (http://www.sanfoundry.com/)

/internship-industrial-engineering/)
Systems Internships (http://www.sanfoundry.com/internships-systems-control-engineering/)
Chemical Internships (http://www.sanfoundry.com/internships-chemical-engineering/)
Civil Internships (http://www.sanfoundry.com/internships-civil-engineering/)
IT Internships (http://www.sanfoundry.com/internship/)
All Stream Internships (http://www.sanfoundry.com/internship/)

Developers Track (http://www.sanfoundry.com/salary-50l/)
Mentoring Sessions (http://www.sanfoundry.com/professional-mentoring-coaching-career-guidance-cto/)
Contact Us (http://www.sanfoundry.com/contact/)
Sitemap (http://www.sanfoundry.com/sitemap_index.xml)