Determinisation of a finite automaton with multiple initial states

Submission deadline:	**2023-11-20 12:00:00**
Evaluation:	**5.0000**
Max. assessment:	**5.0000** (Without bonus points)
Submissions:	4 / 50
Advices:	11 / 50

The determinization algorithm of a finite automaton is a key tool in formal language theory. It finds extensive practical application in text processing, data analysis, and various fields of computer science. By converting non-deterministic automata into deterministic ones, it enables efficient language analysis, which is crucial in compilers, text analysis, regular expression matching, network management, database systems, and many other applications. The task is to find a deterministic finite automaton **without useless and unreachable** states such that the language it recognizes is equivalent to the language of the given non-deterministic finite automaton with multiple initial states.

Implement the determinisation as a function in a C++ program, where the function signature is `DFA determinize ( const MISNFA & nfa );` The input and output of the function are instances of finite automata of structures `MISNFA` and `DFA`, representing nondeterministic finite automaton with multiple initial states and deterministic finite automaton, respectively. These structures are defined by the test environment, note the example below. For simplicity, the states of the automata and alphabet symbols are defined as `int` types and `char` types, respectively.

Comparison of your output, i.e., finite automaton given by your solution and the reference solution is through conversion to minimal deterministic finite automaton (mDFA). Your output can be different from the reference output (for example in state label, including/excluding fail state), the important fact is that after your output is minimized it needs to be same as minimized reference automaton (the state names are not important). **The minimization of your automaton is performed by the test environment. It is not necessary to submit minimal automaton.**

The input is always valid nondeterministic finite automaton with multiple initial states, i.e.:

- state set (`MISNFA::m_States`), initial states (`MISNFA::m_InitialStates`), and alphabet symbols (`MISNFA::m_Alphabet`) are nonempty,
- initial and final states z from sets `MISNFA::m_InitialStates` and `MISNFA::m_FinalStates` are at the same time elements of the `MISNFA::m_States` set,
- if there is a state`q` and alphabet symbol `a` combination with undefined transition, the `MISNFA::m_Transitions` map does not contain the key`(q, a)`
- the `MISNFA::m_Transitions` map contains only those values that are specified in state set and symbol set.

Resulting `DFA` must also satisfy automata definition. The same rules as specified for the MISNFA must hold for the DFA (except for details related to differences in the definition of the initial state and transition function).

In case the language accepted by the automaton is empty, the result is a single-state automaton over the same alphabet as the input MISNFA.

Submit a source file with the implementation of the required determnisation as `determinize` function. Further, the source file must include your auxiliary functions which are called from `determinize` function, if any. The `determinize` function is called from the testing environment, thus, it is important to adhere to the required interface. Use the sample code below as a basis for your development, complete `determinize` function and add your required auxiliary functions. There is a sample `main` function with some test in the sample data below. These sample data are used in the basic test. Please note, the header files, `DFA` and `MISNFA` as well as `main` function are nested in a conditional compile blocks (`#ifdef/#endif`). Please keep these conditional compile blocks in place. They are present to simplify the development. When compiling on your computer, the headers and `main` function are present as usual. On the other hand, the header and `main` function "disappear" when compiled by Progtest. Thus, your tests in `main` function does not interfere with the testing environment's `main` function.

You may base your implementation on a file available below in Sample input section. The file also contains some basic tests. However, your output may be different but still valid. One of the reference solutions generated the test cases, you may need to change them to match the results of your solution.

Your implementation is executed in a limited environment. There are limits on both time and memory. The exact limits are shown in the test log of the reference solution.

Sample data:	**Download**

✓ **Reference**

- **Evaluator: computer**
  - Program compiled
  - Test 'Zakladni test': success
    - result: 100.00 %, required: 100.00 %
    - Total run time: 0.005 s (limit: 10.000 s)
    - Mandatory test success, evaluation: 100.00 %
  - Test 'Test meznich hodnot': success
    - result: 100.00 %, required: 100.00 %
    - Total run time: 0.001 s (limit: 9.995 s)
    - Mandatory test success, evaluation: 100.00 %
  - Test 'Test nahodnymi daty': success
    - result: 100.00 %, required: 50.00 %
    - Total run time: 0.943 s (limit: 9.994 s)
    - Mandatory test success, evaluation: 100.00 %
  - Test 'Test nahodnymi daty s memdbg': success
    - result: 100.00 %, required: 50.00 %
    - Total run time: 1.090 s (limit: 10.000 s)
    - Mandatory test success, evaluation: 100.00 %
  - Overall ratio: 100.00 % (= 1.00 * 1.00 * 1.00 * 1.00)
- Total percent: 100.00 %
- Total points: 1.00 * 5.00 = 5.00

		Total	Average	Maximum	Function name
SW metrics:	Functions:	**3**	--	--	--
	Lines of code:	**84**	**28.00 ± 16.97**	**40**	`doSomethingElse`
	Cyclomatic complexity:	**22**	**7.33 ± 4.92**	**13**	`doSomethingElse`

4	**2023-11-16 20:56:28**	**Download**
Submission status:	Evaluated	
Evaluation:	5.0000	

- **Evaluator: computer**
  - Program compiled
  - Test 'Basic test': success
    - result: 100.00 %, required: 100.00 %
    - Total run time: 0.007 s (limit: 10.000 s)
    - Mandatory test success, evaluation: 100.00 %
  - Test 'Borderline test': success
    - result: 100.00 %, required: 100.00 %
    - Total run time: 0.002 s (limit: 9.993 s)
    - Mandatory test success, evaluation: 100.00 %
  - Test 'Random test': success
    - result: 100.00 %, required: 50.00 %
    - Total run time: 1.332 s (limit: 9.991 s)
    - Mandatory test success, evaluation: 100.00 %
  - Test 'Random test with a memdbg': success
    - result: 100.00 %, required: 50.00 %
    - Total run time: 1.674 s (limit: 10.000 s)
    - Mandatory test success, evaluation: 100.00 %
  - Overall ratio: 100.00 % (= 1.00 * 1.00 * 1.00 * 1.00)
- Advices used: 11

- Penalty due to advices: None (11 <= 50 limit)
- Total percent: 100.00 %
- Total points: 1.00 * 5.00 = 5.00

SW metrics:		Total	Average	Maximum	Function name
	Functions:	**6**	--	--	--
	Lines of code:	**169**	**28.17 ± 26.82**	**76**	make_dfa
	Cyclomatic complexity:	**41**	**6.83 ± 7.06**	**19**	make_dfa

3	**2023-11-16 19:35:23**	**Download**
Submission status:	Evaluated	
Evaluation:	0.0000	

- **Evaluator: computer**
  - Program compiled
  - Test 'Basic test': success
    - result: 100.00 %, required: 100.00 %
    - Total run time: 0.007 s (limit: 10.000 s)
    - Mandatory test success, evaluation: 100.00 %
  - Test 'Borderline test': failed
    - result: 47.37 %, required: 100.00 %
    - Total run time: 0.002 s (limit: 9.993 s)
    - Mandatory test failed, evaluation: 0.00 %
    - ✓ Failed (invalid output)

      ```
      Invalid DFA returned from determinize().
      ```

      ✓ Input data **[20 B / 20 B]**

      ```
      MISNFA a b
      > 0 - -
      ```

    - ✓ Failed (invalid output)

      ```
      Invalid DFA returned from determinize().
      ```

      ✓ Input data **[60 B / 60 B]**

      ```
      MISNFA a b
      > 0 0 0
        1 - 2
        2 3 -
       <3 3 3
        4 5 5
       <5 - -
      ```

    - ✓ Failed (invalid output)

      ```
      Invalid DFA returned from determinize().
      ```

      ✓ Input data **[60 B / 60 B]**

      ```
      MISNFA a b
      > 0 0 0
        1 - 2
        2 3 -
        3 3 3
        4 5 5
       <5 - -
      ```

- ☑ Failed (invalid output)

  ```
  Invalid DFA returned from determinize().
  ```

  ☑ Input data **[44 B / 44 B]**

  ```
  MISNFA a b
  > 0 0 0
    1 - 2
    2 3 -
    3 3 3
  ```

- ☑ Failed (invalid output)

  ```
  Invalid DFA returned from determinize().
  ```

  ☑ Input data **[52 B / 52 B]**

  ```
  MISNFA a b
  > 0 1 -
    1 - 2
    2 3 -
    3 - 4
    4 - -
  ```

- ☑ Failed (invalid output)

  ```
  Invalid DFA returned from determinize().
  ```

  ☑ Input data **[38 B / 38 B]**

  ```
  MISNFA a b
  > 0 0|1 0
    1 - 2
    2 2 2
  ```

- ☑ Failed (invalid output)

  ```
  Invalid DFA returned from determinize().
  ```

  ☑ Input data **[22 B / 22 B]**

  ```
  MISNFA a
  > 0 -
   <1 1
  ```

- ☑ Failed (invalid output)

  ```
  Invalid DFA returned from determinize().
  ```

  ☑ Input data **[16 B / 16 B]**

  ```
  MISNFA a
  > 0 -
  ```

  - Overall ratio: 0.00 % (= 1.00 * 0.00)

- Advices used: 3
- Penalty due to advices: None (3 <= 50 limit)
- Total percent: 0.00 %
- Total points: 0.00 * 5.00 = 0.00

		Total	Average	Maximum	Function name
SW metrics:	Functions:	**6**	--	--	--

	Total	Average	Maximum	Function name
Lines of code:	163	27.17 ± 27.61	76	make_dfa
Cyclomatic complexity:	40	6.67 ± 7.18	19	make_dfa

2	**2023-11-16 19:29:35**	**Download**
Submission status:	Evaluated	
Evaluation:	0.0000	

- **Evaluator: computer**
  - Program compiled
  - Test 'Basic test': failed
    - result: 28.57 %, required: 100.00 %
    - Total run time: 0.007 s (limit: 10.000 s)
    - Mandatory test failed, evaluation: 0.00 %
    - Failed (invalid output) **[Unlock advice (155 B)]**
    - Failed (invalid output) **[Unlock advice (95 B)]**
    - Failed (invalid output) **[Unlock advice (135 B)]**
    - Failed (invalid output) **[Unlock advice (243 B)]**
    - Failed (invalid output) **[Unlock advice (200 B)]**
    - Failed (invalid output) **[Unlock advice (324 B)]**
    - Failed (invalid output) **[Unlock advice (422 B)]**
    - Failed (invalid output) **[Unlock advice (613 B)]**
  - Overall ratio: 0.00 %
- Advices used: 2
- Penalty due to advices: None (2 <= 50 limit)
- Total percent: 0.00 %
- Total points: 0.00 * 5.00 = 0.00

SW metrics:

	Total	Average	Maximum	Function name
Functions:	6	--	--	--
Lines of code:	163	27.17 ± 27.61	76	make_dfa
Cyclomatic complexity:	40	6.67 ± 7.18	19	make_dfa

1	**2023-11-16 19:19:14**	**Download**
Submission status:	Evaluated	
Evaluation:	0.0000	

- **Evaluator: computer**
  - Program compiled
  - Test 'Basic test': failed
    - result: 28.57 %, required: 100.00 %
    - Total run time: 0.003 s (limit: 10.000 s)
    - Mandatory test failed, evaluation: 0.00 %
    - ✓ Failed (invalid output)

```
Minimized DFAs are not isomporphic.
```

✓ Input data **[36 B / 36 B]**

```
MISNFA e l
> 0 1 1
 <1 2 -
 <2 0 2
```

✓ Output data **[33 B / 33 B]**

```
DFA e l
  0 1 1
><1 2 -
```

```
 <2 0 2
```

✓ Reference **[33 B / 33 B]**

```
DFA e l
> 1 2 2
 <2 3 -
 <3 1 3
```

- ✓ Failed (invalid output)

```
 Found unreachable or useless state.
```

✓ Input data **[44 B / 44 B]**

```
MISNFA g l
><0 1 2
  1 3 3
 <2 1 0
 <3 - 1
```

✓ Output data **[41 B / 41 B]**

```
DFA g l
 <0 1 2
> 1 3 3
 <2 1 0
 <3 - 1
```

✓ Reference **[41 B / 41 B]**

```
DFA g l
><1 2 3
  2 4 4
 <3 2 1
 <4 - 2
```

- ✓ Failed (invalid output)

```
 Found unreachable or useless state.
```

✓ Input data **[124 B / 124 B]**

```
MISNFA l m
> 0 1 2
 <1 3 -
  2 4 5
 <3 3 -
  4 3 6
  5 7 6
  6 7 8
  7 9 10
  8 6 10
  9 7 8
  10 11 4
  11 12 9
  12 6 10
```

✓ Output data **[121 B / 121 B]**

```
DFA l m
  0 1 2
><1 3 -
  2 4 5
```

```
 <3 3 -
  4 3 6
  5 7 6
  6 7 8
  7 9 10
  8 6 10
  9 7 8
  10 11 4
  11 12 9
  12 6 10
```

✓ Reference **[124 B / 124 B]**

```
DFA l m
> 1 2 3
 <2 4 -
  3 5 6
 <4 4 -
  5 4 7
  6 8 7
  7 8 9
  8 10 11
  9 7 11
  10 8 9
  11 12 5
  12 13 10
  13 7 11
```

        - Failed (invalid output) **[Unlock advice (369 B)]**
        - Failed (invalid output) **[Unlock advice (200 B)]**
        - Failed (invalid output) **[Unlock advice (452 B)]**
        - Failed (invalid output) **[Unlock advice (419 B)]**
        - Failed (invalid output) **[Unlock advice (611 B)]**
    - Overall ratio: 0.00 %
- Total percent: 0.00 %
- Total points: 0.00 * 5.00 = 0.00

		Total	Average	Maximum	Function name
SW metrics:	Functions:	**6**	--	--	--
	Lines of code:	**158**	**26.33 ± 28.01**	**76**	`make_dfa`
	Cyclomatic complexity:	**40**	**6.67 ± 7.18**	**19**	`make_dfa`