

CISS445: Programming Languages
Quiz q3104

Name: nweadick1@cougars.ccis.eduScore:

Open `main.tex` and enter answers (look for `answercode`, `answerbox`, `answerlong`). Turn the page for detailed instructions. To rebuild and view pdf, in bash shell execute `make`. To build a gzip-tar file, in bash shell execute `make s` and you'll get `submit.tar.gz`.

Q1. T or F or M. Design a DFA M that accepts strings made up of 0's and 1's that contains exactly one substring with a strictly positive even number of 0's (of course "strictly positive even number" means 2 or 4 or 6 or ...). Here are some strings in $L(M)$:

00, 0000, 100111, 1100, 111001111

Here are some strings not in $L(M)$:

ϵ , 010, 000, 100000, 11110111, 1001001, 1100001110000

Here's an example of now you should specify your DFA below:

```
S = ["0", "1"]
Q = ["q0", "q1", "q2", "q3"]
start = "q0"
F = ["q1"]
transitions = {("q0","0"): "q1",
               ("q0","1"): "q2",
               ("q1","0"): "q0",
               ("q1","1"): "q1",
               ("q2","0"): "q2",
               ("q2","1"): "q3",
               ("q3","0"): "q0",
               ("q3","1"): "q0",
               }
```

In this case, the set of symbols is S which is made up of "0" and "1". The states are in Q which in this case is made up of "q0", ..., "q3". The start state is "q0". The accept states are in F – in this case where's only one accept states "q1". The transitions are in `transitions`. For instance at state "q0", if the character to be processed is "0", then the DFA goes into state "q1". (See the first entry for transition.) If at state "q0", the character to be processes is "1", then the DFA goes into state "q2".

Etc. The above is (of course) not the answer. It's provided just to show you how to describe the DFA to me. Make sure you follow the format exactly.

ANSWER:

```
S = ["0", "1"]
Q = ["q0", "q1"]
start = "q0"
F = ["q1"]
transitions = {("q0","00"): "q0",
                ("q0","1"): "q1",
                ("q1","0"): "q0",
                ("q1","1"): "q1",
                }
```

(This DFA is easy enough for you to design it as a DFA. But if you can't get the design right away, then describe the regex, then the NFA, and then convert to DFA.)

INSTRUCTIONS

In `main.tex` change the email address in

```
\renewcommand\AUTHOR{jdoe5@cougars.ccis.edu}
```

yours. In the bash shell, execute “`make`” to recompile `main.pdf`. Execute “`make v`” to view `main.pdf`. Execute “`make s`” to create `submit.tar.gz` for submission.

For each question, you’ll see boxes for you to fill. You write your answers in `main.tex` file. For small boxes, if you see

```
1 + 1 = \answerbox{}
```

you do this:

```
1 + 1 = \answerbox{2}
```

`answerbox` will also appear in “true/false” and “multiple-choice” questions.

For longer answers that needs typewriter font, if you see

```
Write a C++ statement that declares an integer variable name x.
\begin{answercode}
\end{answercode}
```

you do this:

```
Write a C++ statement that declares an integer variable name x.
\begin{answercode}
int x;
\end{answercode}
```

`answercode` will appear in questions asking for code, algorithm, and program output. In this case, indentation and spacing is significant. For program output, I do look at spaces and newlines.

For long answers (not in typewriter font) if you see

```
What is the color of the sky?
\begin{answerlong}
\end{answerlong}
```

you can write

```
What is the color of the sky?
\begin{answerlong}
The color of the sky is blue.
\end{answerlong}
```

For students beyond 245: You can put \LaTeX commands in `answerbox` and `answerlong`.

A question that begins with “T or F or M” requires you to identify whether it is true or false, or meaningless. “Meaningless” means something’s wrong with the statement and it is not well-defined. Something like “ $1+_2$ ” or “ $\{2\}^{\{3\}}$ ” is not well-defined. Therefore a question such as “Is $42 = 1+_2$ true or false?” or “Is $42 = \{2\}^{\{3\}}$ true or false?” does not make sense. “Is $P(42) = \{42\}$ true or false?” is meaningless because $P(X)$ is only defined if X is a set. For “Is $1 + 2 + 3$ true or false?”, “ $1 + 2 + 3$ ” is well-defined but as a “numerical expression”, not as a “proposition”, i.e., it cannot be true or false. Therefore “Is $1 + 2 + 3$ true or false?” is also not a well-defined question.

When writing results of computations, make sure it’s simplified. For instance write 2 instead of $1 + 1$. When you write down sets, if the answer is $\{1\}$, I do not want to see $\{1, 1\}$.

When writing a counterexample, always write the simplest.

Here are some examples (see `instructions.tex` for details):

1. T or F or M: $1 + 1 = 2$ T

2. T or F or M: $1 + 1 = 3$ F

3. T or F or M: $1+_2 =$ M

4. $1 + 2 =$ 3

5. Write a C++ statement to declare an integer variable named **x**.

```
int x;
```

6. Solve $x^2 - 1 = 0$.

Since $x^2 - 1 = (x - 1)(x + 1)$, $x^2 - 1 = 0$ implies $(x - 1)(x + 1) = 0$. Therefore $x - 1 = 0$ or $x = -1$. Hence $x = 1$ or $x = -1$.

7. Which is true? C

(A) $1 + 1 = 0$

(B) $1 + 1 = 1$

(C) $1 + 1 = 2$

(D) $1 + 1 = 3$

(E) $1 + 1 = 4$