

CS 160 Compilers

Lecture 6: Regular Expressions and Finite State Machine

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

Yu Feng
Fall 2021

Outline

- Last time: Specifying lexical structure using **regular expressions**
- Today: How to recognize strings matching regular expressions using finite automata.
Assignment Project Exam Help
<https://tutorcs.com>
- We will see deterministic finite automata (DFAs) and non-deterministic finite automata (NFAs)
WeChat: tutorcs
- High-level story: RegEx -> NFA -> DFA -> Table

Finite automata

- Regular Expressions \Leftrightarrow Specification

- Finite Automata \Leftrightarrow Implementation

Assignment Project Exam Help

- A finite automata formally consists of:

<https://tutorcs.com>

- An input alphabet Σ WeChat: cstutorcs

- A set of states S

- A start state n

- A set of accepting states $F \subseteq S$

- A set of transitions $\text{state} \xrightarrow{\text{input}} \text{state}$

Finite automata

- Transition $S_1 \xrightarrow{\alpha} S_2$
- This means: In state S_1 and input character α , go to state S_2
- If end of input and in accepting state \Rightarrow accept
- Otherwise \Rightarrow reject

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

Finite Automata as State Graphs

A state:

The start state:

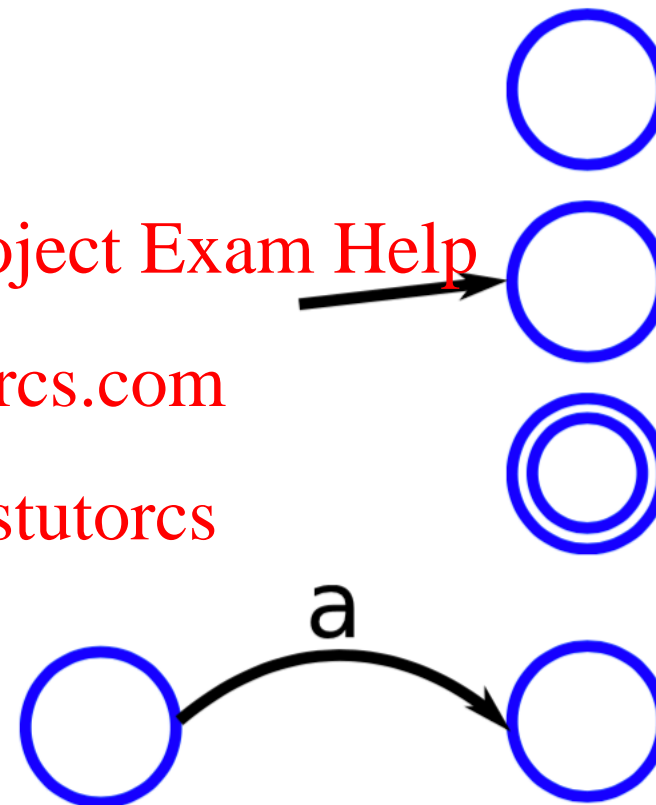
An accepting state:

A transition:

Assignment Project Exam Help

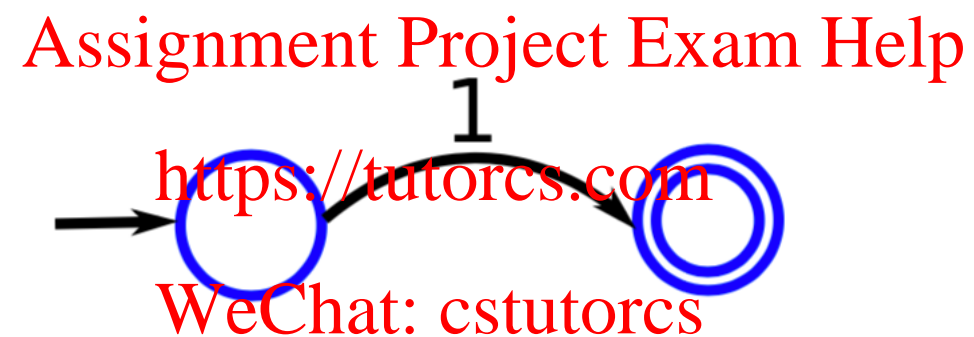
<https://tutorcs.com>

WeChat: cstutorcs



A simple example

- Here is an automaton that only accepts the string "1":



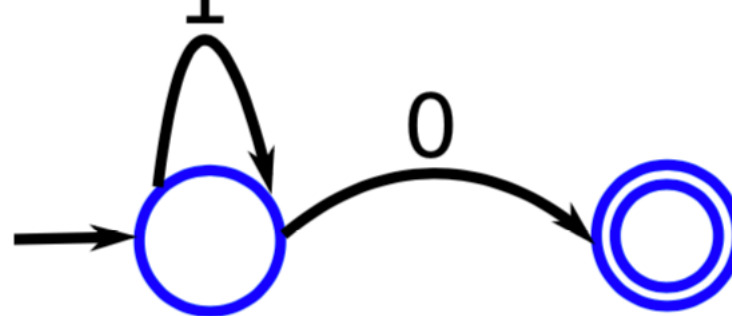
Another simple example

- A finite automaton accepting any number of 1's followed by a single 0
- Alphabet: $\{0,1\}$

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs



Epsilon transitions

- A special kind of transition: ϵ -transitions
- Machine can move from state A to B without reading any input



Deterministic and Nondeterministic Automata

- Deterministic Finite Automata (DFA)
 - At most one transition per input on any state
 - No ϵ moves
- Nondeterministic Finite Automate (NFA)
 - Can have multiple transitions for one input in a given state
 - Can have ϵ -moves

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

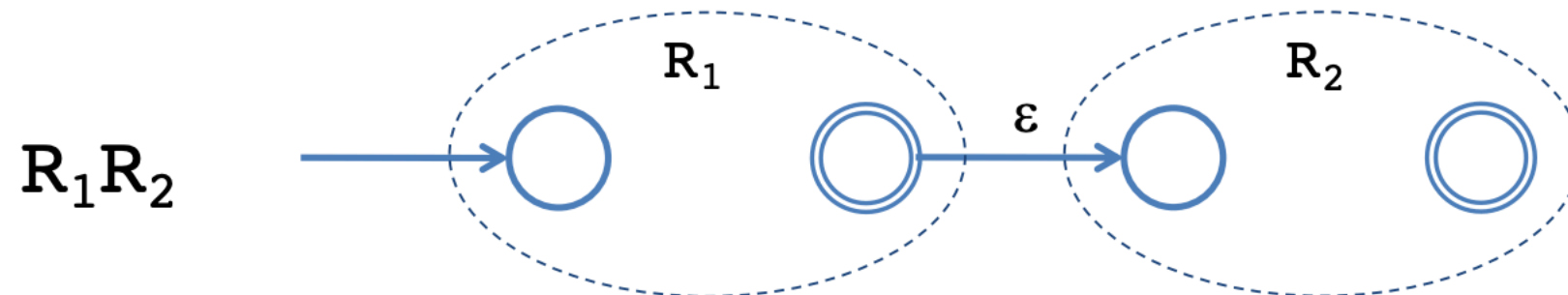
RE to NFA

- Can we build a finite automaton for every regular expression
- Strategy: consider every possible regular expression (by induction on the structure of the regular expressions)

Assignment Project Exam Help

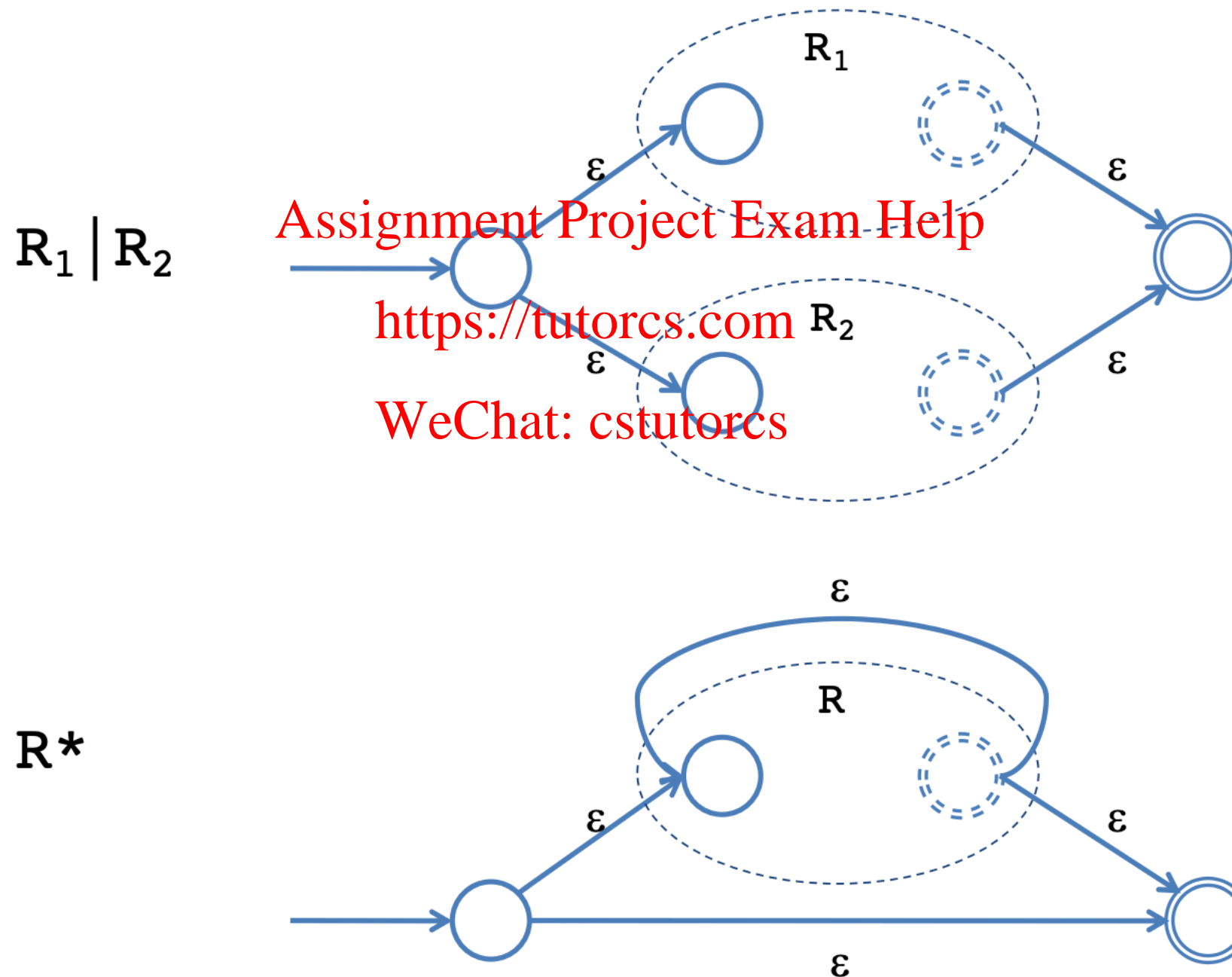
<https://tutorcs.com>

WeChat: cstutorcs



RE to NFA

- Can we build a finite automaton for every regular expression



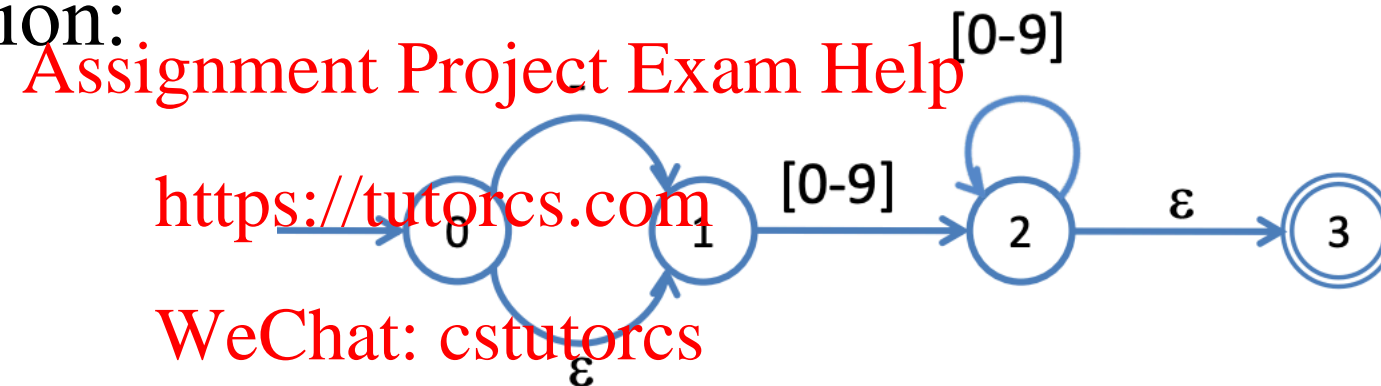
NFA to DFA: The Trick

- **Insight:** Simulate the NFA
- At any given time, the NFA is in a **set of states**
Assignment Project Exam Help
- **State** in the DFA \Rightarrow all **reachable** subsets of states in the NFA
https://tutorcs.com
WeChat: cstutorcs
- **Start State:** the set of states reachable through ϵ moves from the NFA start state
- Add transition $A \xrightarrow{\alpha} B$ to DFA iff:
 - B is in the set of states reachable from any state in A after seeing input α , considering ϵ moves as well

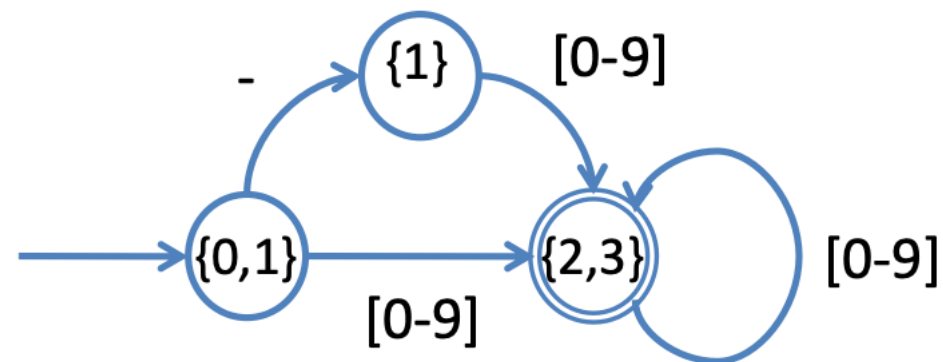
NFA to DFA: Example

- Consider: $-?[0-9]^+$

- NFA representation:



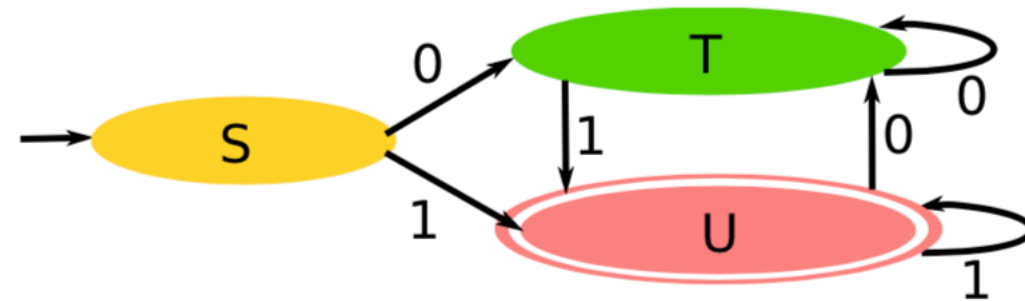
- DFA representation:



DFA: Implementation

- A DFA can be implemented by a 2D table T
 - One dimension is “states”
Assignment Project Exam Help
 - Other dimension is “input symbols”
<https://tutorcs.com>
WeChat: cstutorcs
 - For every transition $A \xrightarrow{c} B$ define $T[A,c]=B$
- DFA “execution”: If in state A and input c, read $T[A,c] = B$ and switch to state B
- Very efficient

Implementation of a DFA



Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

	0	1
S	T	U
T	T	U
U	T	U

Translation from NFA to the table implementation is handled by modern lexer

TODOs by next lecture

- Hw2 will be out. Get familiar with the Patina language
- Come to the discussion session if you have questions

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs