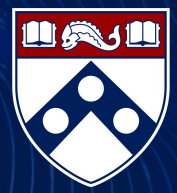


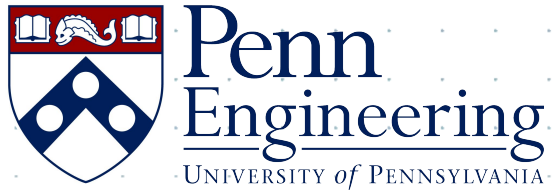
Penn Engineering
University of Pennsylvania

CIS 3990 Recitation 03

NFAs

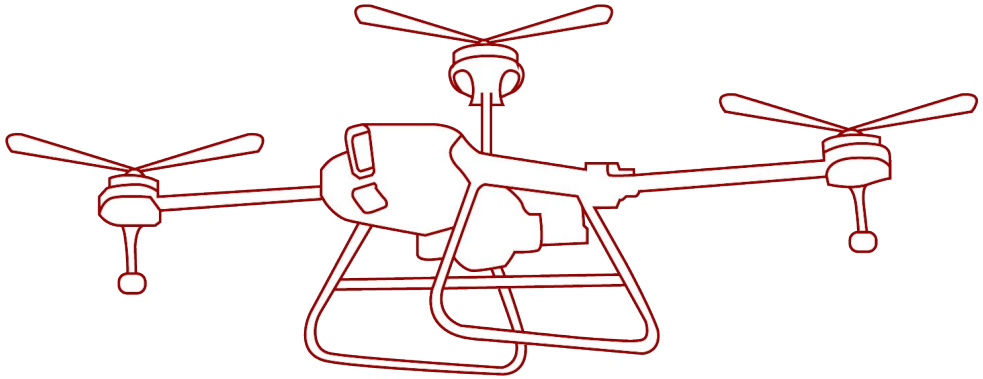
2025-09-18

Agenda

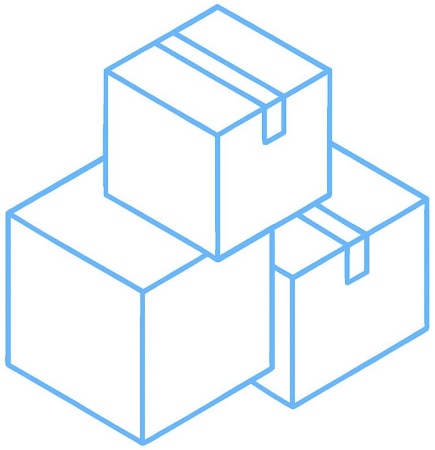


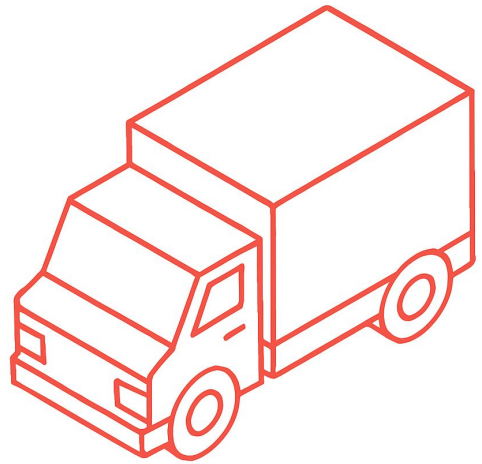
01	Logistics
02	STL Applications
03	NFAs
04	Ed Practice

Penn Engineering
University of Pennsylvania



Logistics

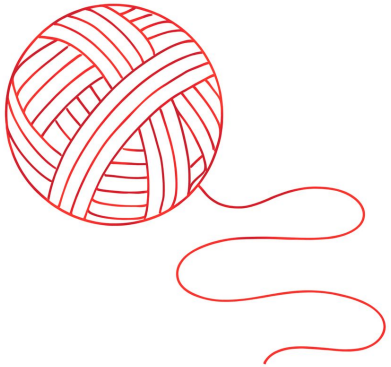




Logistics

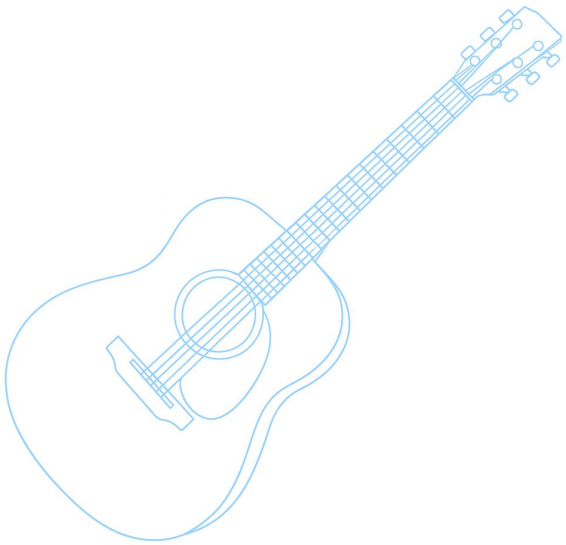
- HW00 through HW02 past-due - can reopen 1 HW per check-in
- HW03 out - due 11:59pm on Tue, Sept 23
- Exam 0 - Oct 22 - Please let us know ASAP if you have conflicts
- Ash's OH - now Mon. 5:30 - 7:30 pm in Levine 269C
  - when in doubt, check the website calendar

Penn Engineering
University of Pennsylvania



STL Applications

Time to use it!



For the Homework

Penn Engineering
University of Pennsylvania

- vector - for “arrayLists” (aka dynamic length)
- algorithm - for sorting
- pair - for grouping values without defining structs
- string - less need for C’s char*
- set, map - ordered by key
- unordered_set, unordered_map - hash-based

Use Cases: (Unsorted) Map

Penn Engineering
University of Pennsylvania

- Tracking unique entities (keys) and its relevant information (values)
  - Library's inventory:
    - **key** = ISBN
    - **value** = a struct containing book title, author, and genre.

- …Especially when the information can be updated
  - Library's inventory:
    - **key** = ISBN
    - **value** = # of books available.

Use Cases: (Unsorted) Map

- "Where was x value last seen?"
  - ex: location of substrings in a given string
    - **key** = substring
    - **value** = index in the string

(Unsorted) Set

- Checking for existence
  - ex: Set of books currently available for checkout
  - I'm not concerned with how many books there are, I just need to know if there is at least one book!

- When we only care about unique items
  - ex: how many different letters are in "Mississippi?"

Pair

- Combines two types into a new type
- Can combine Pair with another data structure
  - Eg: a pentagon is a vector of 5 pairs, a pair represents an (x,y) coordinate

Map of Sets, Maps of Maps?

Penn Engineering
University of Pennsylvania

- Maps of Sets: Grouping data (by **key**) without duplicates
  - ex: given a list of files with unique file names, collect all unique words in each file, keyed on the filename. Which one has the most unique words?

- Maps of Maps: Within a group, make a subgroup
  - ex: for each file in a list of files, provide the word count of each unique word in the file, sorted from highest count to lowest count.

NFAs

Penn Engineering
University of Pennsylvania

chalk & talk:

What is an NFA?

How is it different from a DFA?

NFAs

Penn Engineering
University of Pennsylvania

Walk through together: how to make a DFA that accepts all strings ending with “101” or “010?”

NFAs

Penn Engineering
University of Pennsylvania

Walk through together: how to make a DFA that accepts all strings ending with “101” or “010?”

1. Draw DFA for the “ideal case:” a string that is only “101” or “010”

NFAs

Penn Engineering
University of Pennsylvania

Walk through together: how to make a DFA that accepts all strings ending with “101” or “010?”

1. Draw DFA for the “ideal case:” a string that is only “101” or “010”
2. Start accounting for the other cases

NFAs

Penn Engineering
University of Pennsylvania

Walk through together: how to make a DFA that accepts all strings ending with “101” or “010?”

1. Draw DFA for the “ideal case:” a string that is only “101” or “010”
2. Start accounting for the other cases
    a. strings that are 3 letters and not “101” or “010”

NFAs

Penn Engineering
University of Pennsylvania

Walk through together: how to make a DFA that accepts all strings ending with “101” or “010?”

1. Draw DFA for the “ideal case:” a string that is only “101” or “010”
2. Start accounting for the other cases
   a. strings that are 3 letters and not “101” or “010”
   b. strings longer than 3 letters and end with “101” or “010”

NFAs

Penn Engineering
University of Pennsylvania

Walk through together: how to make a DFA that accepts all strings ending with “101” or “010?”

1. Draw DFA for the “ideal case:” a string that is only “101” or “010”
2. Start accounting for the other cases
   a. strings that are 3 letters and not “101” or “010”
   b. strings longer than 3 letters and end with “101” or “010”
   c. strings longer than 3 letters that contain but do not end with “101” or “010”

NFAs

Penn Engineering
University of Pennsylvania

Walk through together: how to make a DFA that accepts all strings ending with “101” or “010?”

1. Draw DFA for the “ideal case:” a string that is only “101” or “010”
2. Start accounting for the other cases
   a. strings that are 3 letters and not “101” or “010”
   b. strings longer than 3 letters and end with “101” or “010”
   c. strings longer than 3 letters that contain but do not end with “101” or “010”
   d. strings longer than 3 letters that do not contain “101” or “010”

NFAs

Penn Engineering
University of Pennsylvania

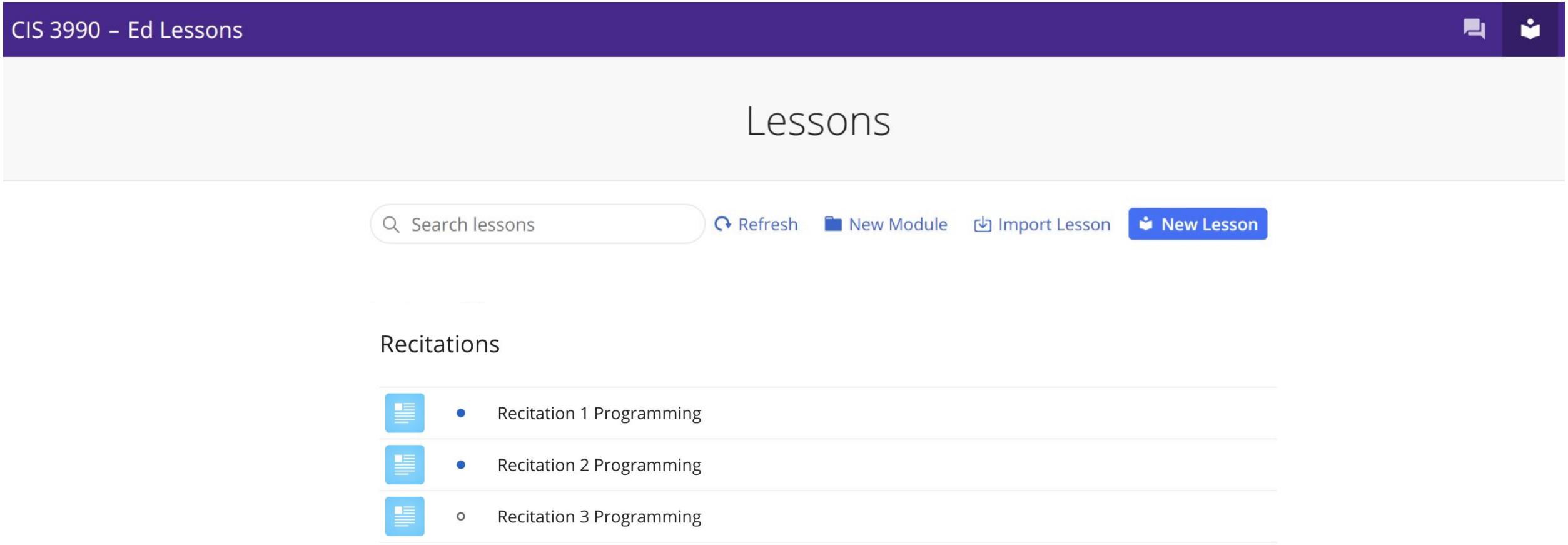
Now, let's rewrite it as an NFA!

NFA = DFA except

1. can have same input go to 2 different states

2. epsilon transition: transition to the next state doesn't consume any input

Ed practice


CIS 3990 – Ed Lessons
Lessons
Search lessons
Refresh
New Module
Import Lesson
New Lesson
Recitations
Recitation 1 Programming
Recitation 2 Programming
Recitation 3 Programming


That's all Folks!