Section 9

CSE 311 - Sp 2022

Administrivia

Announcements and Reminders

- HW7
  - Due yesterday, Wednesday 5/18 @ 10pm
  - Late due date Saturday 5/28 @ 10pm
- HW8
  - Last homework!
  - Out now, due next Wednesday 6/1 @ 10pm
- Final Exam Info:
  - In-person on Monday 6/6 @ 12:30 pm
  - Majority of students in Kane 120, some students in smaller extra location for increased distancing
  - Ed post with more info + form to fill out to request which room you prefer
- If you have any questions or concerns about your grade:
  - Reach out to Robbie to schedule a quick grade chat!
  - Reach out to your TAs for extra help if you need it - we are here to help!

References

- How to LaTeX
  - https://courses.cs.washington.edu/courses/cse311/22sp/assignments/HowToLaTeX.pdf
- Logical Equivalences
  - https://courses.cs.washington.edu/courses/cse311/22sp/resources/reference-logical_equiv.pdf
- Inference Rules
  - https://courses.cs.washington.edu/courses/cse311/22sp/resources/InferenceRules.pdf
- Set Definitions
  - https://courses.cs.washington.edu/courses/cse311/22sp/resources/reference-sets.pdf
- Modular Arithmetic Definitions and Properties
  - https://courses.cs.washington.edu/courses/cse311/22sp/resources/reference-number-theory.pdf
- Induction Templates
  - https://courses.cs.washington.edu/courses/cse311/22sp/resources/induction-templates.pdf

Warm-Up with
Context-Free Grammars

CFGs

- This is another way we can describe a language using recursive rules.

- We can think of CFGs as *generating* strings:

  1. Start from the start symbol S.
  2. Choose a nonterminal in the string, and a production rule $A \to w_1 \mid w_2 \mid \dots \mid w_k$ replace that copy of the nonterminal with w_i
  3. If no nonterminals remain, you're done! Otherwise, goto step 2.

- A string is in the language of the CFG iff it can be generated starting from S.

- All regular expressions can be written as context-free grammars, but not all context-free grammars can be written as regular expressions!

Context-Free Grammars

A context free grammar (CFG) is a finite set of production rules over:
An alphabet Σ of "terminal symbols"
A finite set V of "nonterminal symbols"
A start symbol (one of the elements of V) usually denoted S.

A production rule for a nonterminal $A \in V$ takes the form
$A \rightarrow w_1 \mid w_2 \mid \ldots \mid w_k$
Where each $w_i \in V \cup \Sigma^*$ is a string of nonterminals and terminals.

Problem 1 - CFGs

Write a context-free grammar to match each of these languages.

(a) All binary strings that start with 11.

(b) All binary strings that contain at most one 1.

(c) All strings over 0, 1, 2 with the same number of 1s and 0s and exactly one 2.

Try to finish one or two parts of this problem with the people around you, and then we'll go over it together!

Problem 1 - CFGs

(a) All binary strings that start with 11.

(b) All binary strings that contain at most one 1.

(c) All strings over 0, 1, 2 with the same number of 1s and 0s and exactly one 2.

Problem 1 - CFGs

(a) All binary strings that start with 11.

S → 11**T**
T → 1**T** | 0**T** | ε

(b) All binary strings that contain at most one 1.

(c) All strings over 0, 1, 2 with the same number of 1s and 0s and exactly one 2.

Problem 1 - CFGs

(a) All binary strings that start with 11.

S → 11**T**
T → 1**T** | 0**T** | ε

(b) All binary strings that contain at most one 1.

S → **ABA**
A → 0**A** | ε
B → 1 | ε

(c) All strings over 0, 1, 2 with the same number of 1s and 0s and exactly one 2.

Problem 1 - CFGs

(a) All binary strings that start with 11.

S → 11**T**
T → 1**T** | 0**T** | ε

(b) All binary strings that contain at most one 1.

S → **ABA**
A → 0**A** | ε
B → 1 | ε

(c) All strings over 0, 1, 2 with the same number of 1s and 0s and exactly one 2.

S → 2**T** | **T**2 | **ST** | **TS** | 0**S**1 | 1**S**0
T → **TT** | 0**T**1 | 1**T**0 | ε

Relations

Properties of Relations

For a binary relation R on a set $\boldsymbol{S}$:

- It is “reflexive” iff:
  - for all $\boldsymbol{a} \in \boldsymbol{S}$, [$(\boldsymbol{a}, \boldsymbol{a}) \in \boldsymbol{R}$]

- It is “transitive” iff:
  - for all $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c} \in \boldsymbol{S}$, [$(\boldsymbol{a}, \boldsymbol{b}) \in \boldsymbol{R} \land (\boldsymbol{b}, \boldsymbol{c}) \in \boldsymbol{R} \rightarrow (\mathbf{a}, \mathbf{c}) \in \boldsymbol{R}$]

- It is “symmetric” iff:
  - for all $\boldsymbol{a}, \boldsymbol{b} \in \boldsymbol{S}$, [$(\boldsymbol{a}, \boldsymbol{b}) \in \boldsymbol{R} \rightarrow (\mathbf{b}, \mathbf{a}) \in \boldsymbol{R}$]

- It is “anti-symmetric” iff:
  - for all $\boldsymbol{a}, \boldsymbol{b} \in \boldsymbol{S}$, [$(\boldsymbol{a}, \boldsymbol{b}) \in \boldsymbol{R} \land \boldsymbol{a} \neq \boldsymbol{b} \rightarrow (\mathbf{b}, \mathbf{a}) \notin \boldsymbol{R}$]

Problem 2 - Relations

(a) Consider the relation $R = \{(x, y) : x = y + 1\}$ on $\mathbb{N}$. Is R reflexive? Transitive? Symmetric? Anti-symmetric?

(b) Consider the relation $S = \{(x, y) : x^2 = y^2\}$ on $\mathbb{R}$. Prove that S is reflexive, transitive, and symmetric

Work on this problem with the people around you, and then we'll go over it together!

Problem 2 - Relations

(a) Consider the relation $R = \{(x, y) : x = y + 1\}$ on $\mathbb{N}$. Is R reflexive? Transitive? Symmetric? Anti-symmetric?

Problem 2 - Relations

(a) Consider the relation $R = \{(x, y) : x = y + 1\}$ on $\mathbb{N}$. Is R reflexive? Transitive? Symmetric? Anti-symmetric?

It isn't reflexive, because $1 \neq 1 + 1$; so, $(1, 1) \notin R$.

It isn't symmetric, because $(2, 1) \in R$ (because $2 = 1 + 1$), but $(1, 2) \notin R$, because $1 \neq 2 + 1$.

It isn't transitive, because note that $(3, 2) \in R$ and $(2, 1) \in R$, but $(3, 1) \notin R$.

It is anti-symmetric, because consider $(x, y) \in R$ such that $x \neq y$. Then, $x = y + 1$ by definition of R. However, $(y, x) \notin R$, because $y = x - 1 \neq x + 1$.

Problem 2 - Relations

(b) Consider the relation $S = \{(x, y) : x^2 = y^2\}$ on $\mathbb{R}$. Prove that S is reflexive, transitive, and symmetric.

Problem 2 - Relations

(b) Consider the relation $S = \{(x, y) : x^2 = y^2\}$ on $\mathbb{R}$. Prove that S is reflexive, transitive, and symmetric.

Consider $x \in S$. Note that by definition of equality, $x^2 = x^2$; so, $(x, x) \in S$. so, S is reflexive.

Consider $(x, y) \in S$. Then, $x^2 = y^2$. It follows that $y^2 = x^2$; so, $(y, x) \in S$. So, S is symmetric.

Suppose $(x, y) \in S$ and $(y, z) \in S$. Then, $x^2 = y^2$, and $y^2 = z^2$. Since equality is transitive, $x^2 = z^2$. So, $(x, z) \in S$. So, S is transitive.

Deterministic Finite Automata (DFA)

DFAs

- A DFA is a finite-state machine that accepts or rejects a given string of symbols, by running through a state sequence uniquely determined by the string.

- In other words:
  - Our machine is going to get a string as input. It will read one character at a time and update "its state."
  - At every step, the machine thinks of itself as in one of the (finite number) vertices. When it reads the character it follows the arrow labeled with that character to its next state.
  - Start at the "start state" (unlabeled, incoming arrow).
  - After you've read the last character, accept the string if and only if you're in a "final state" (double circle).

- Every machine is defined with respect to an alphabet Σ

- Every state has exactly one outgoing edge for every character in Σ

- There is exactly one start state; can have as many accept states (aka final states) as you want – including none

Problem 3 - DFAs, Stage 1

Construct DFAs to recognize each of the following languages.
Let $\Sigma = \{0, 1, 2, 3\}$.

(a) All binary strings.

(b) All strings whose digits sum to an even number.

(c) All strings whose digits sum to an odd number.

Work on this problem with the people around you, and then we'll go over it together!

Problem 3 - DFAs, Stage 1

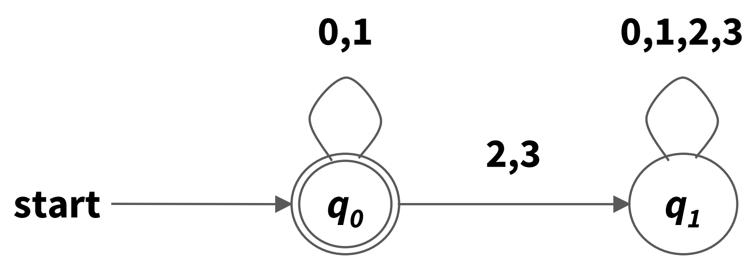
Let $\Sigma = \{0, 1, 2, 3\}$.

(a) All binary strings.

Problem 3 - DFAs, Stage 1

Let Σ = {0, 1, 2, 3}.

(a) All binary strings.



q_0: binary strings

q_1: strings that contain a character which is not 0 or 1

Problem 3 - DFAs, Stage 1

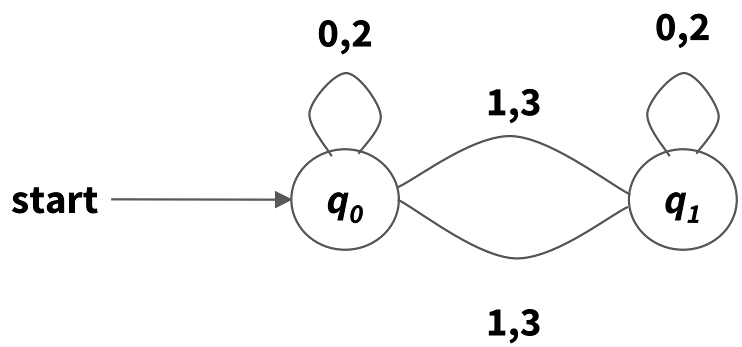
Let $\Sigma = \{0, 1, 2, 3\}$.

(b) All strings whose digits sum to an even number.

Problem 3 - DFAs, Stage 1

Let Σ = {0, 1, 2, 3}.

(b) All strings whose digits sum to an even number.



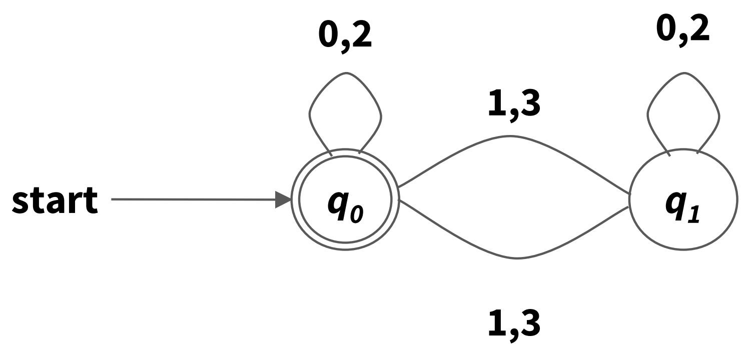
q_0: strings whose sum of digits is even

q_1: strings whose sum of digits is odd

Problem 3 - DFAs, Stage 1

Let $\Sigma = \{0, 1, 2, 3\}$.

(b) All strings whose digits sum to an even number.



q_0: strings whose sum of digits is even

q_1: strings whose sum of digits is odd

Problem 3 - DFAs, Stage 1

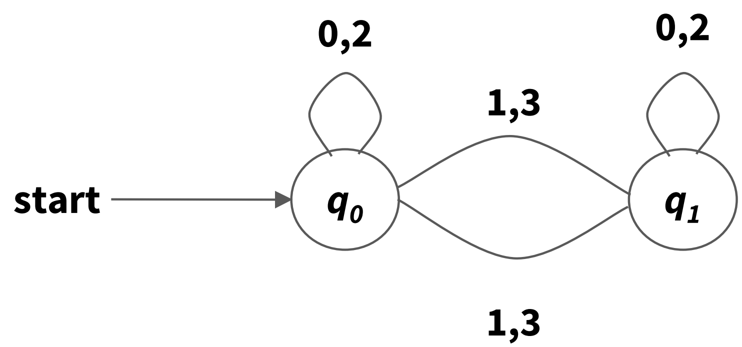
Let $\Sigma = \{0, 1, 2, 3\}$.

(c) All strings whose digits sum to an odd number.

Problem 3 - DFAs, Stage 1

Let $\Sigma = \{0, 1, 2, 3\}$.

(c) All strings whose digits sum to an odd number.



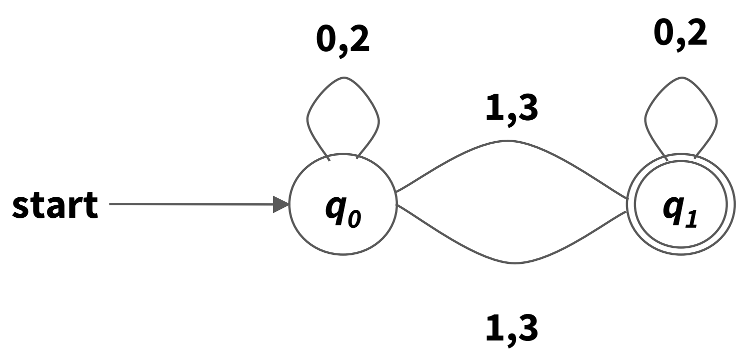
q_0: strings whose sum of digits is even

q_1: strings whose sum of digits is odd

Problem 3 - DFAs, Stage 1

Let Σ = {0, 1, 2, 3}.

(c) All strings whose digits sum to an odd number.



q_0: strings whose sum of digits is even
q_1: strings whose sum of digits is odd

Nondeterministic Finite Automata (NFA)

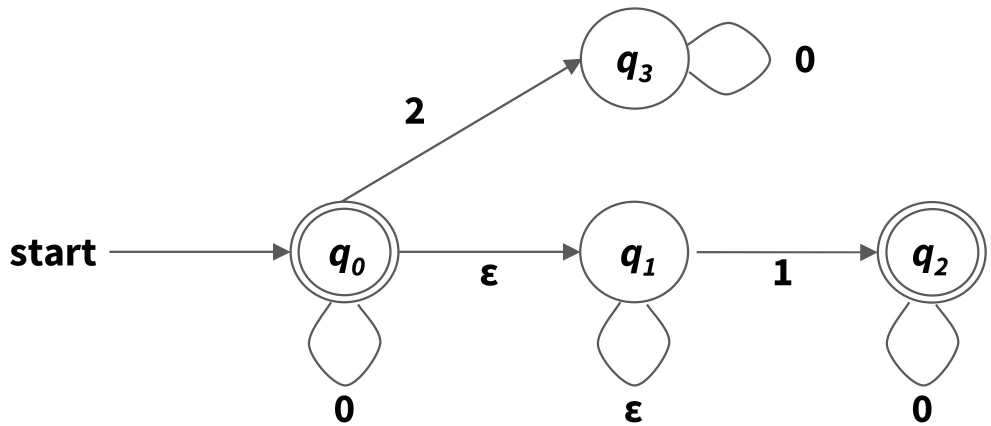
NFAs

- Similar to DFAs, but with less restrictions.
  - From a given state, we'll allow any number of outgoing edges labeled with a given character. (In a DFA, we have only 1 outgoing edge labeled with each character).
  - The machine can follow any of them.
  - We'll have edges labeled with "ε" – the machine (optionally) can follow one of those without reading another character from the input.
  - If we "get stuck" i.e. the next character is a and there's no transition leaving our state labeled a, the computation dies.

- An NFA still has exactly one start state and any number of final states.
- The NFA accepts x if there is some path from a start state to a final state labeled with x.
- From a state, you can have 0,1, or many outgoing arrows labeled with a single character. You can choose any of them to build the required path.

Problem 5 - NFAs

(a) What language does the following NFA accept?

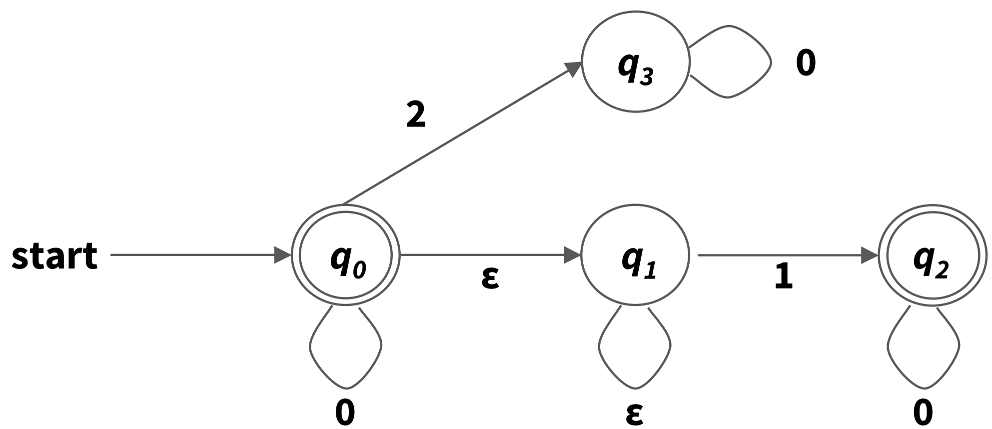


(b) Create an NFA for the language “all binary strings that have a 1 as one of the last three digits”.

Work on this problem with the people around you, and then we’ll go over it together!

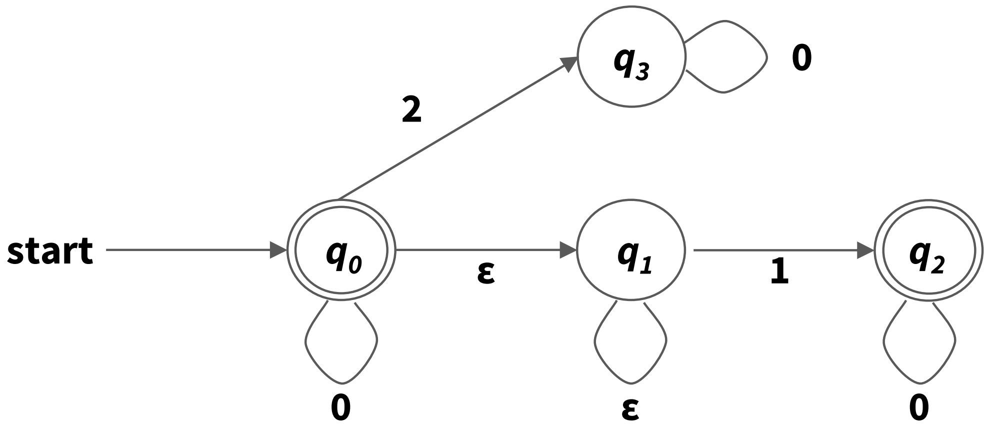
Problem 5 - NFAs

(a) What language does the following NFA accept?



Problem 5 - NFAs

(a) What language does the following NFA accept?



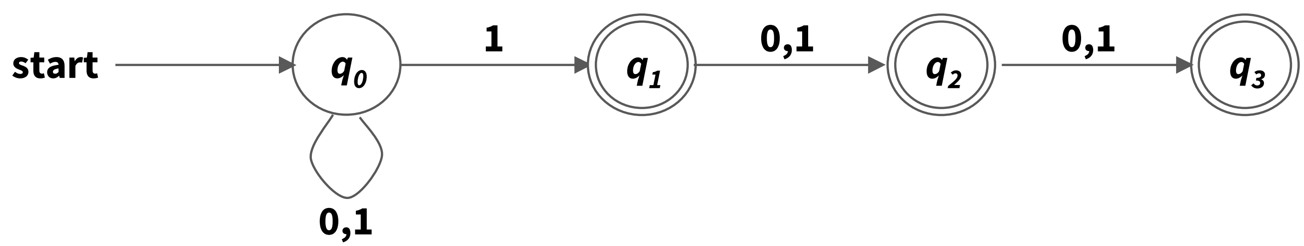
All strings of only 0's and 1's, not containing more than one 1.

Problem 5 - NFAs

(b) Create an NFA for the language “all binary strings that have a 1 as one of the last three digits”.

Problem 5 - NFAs

(b) Create an NFA for the language “all binary strings that have a 1 as one of the last three digits”.



That’s All, Folks!

Any questions?