PDA and CFG

Regular and Context-free Languages

CSE 105 Week 5 Discussion

Deadlines and Logistics

- Test 1 next week (2/12 - 2/14)
- Schedule your tests asap on PrairieTest !
- Do review quizzes on PrairieLearn
- Review grades for HW 2

Test 1 topics

Dates: 2/10/25 - 2/16/25. Test 1 in CBTF this week. The test covers material in Weeks 1 through Week 5. In particular, you should make sure you can:

- Distinguish between alphabet, language, sets, and strings
- Translate a decision problem to a set of strings coding the problem
- Use regular expressions and relate them to languages and automata
- Write and debug regular expressions using correct syntax
- Determine if a given string is in the language described by a regular expression
- Use precise notation to formally define the state diagram of DFA, NFA and use clear English to describe computations of DFA, NFA informally.
- State the formal definition of (deterministic) finite automata
- Trace the computation of a finite automaton on a given string using its state diagram
- Translate between a state diagram and a formal definition
- Determine if a given string is in the language described by a finite automaton
- Design and analyze an automaton that recognizes a given language
- Specify a general construction for DFA based on parameters
- Design and analyze general constructions for DFA
- Motivate the use of nondeterminism
- Trace the computation(s) of a nondeterministic finite automaton
- Determine the language recognized by a given NFA
- Design and analyze general constructions for NFA
- Choose between multiple models to prove that a language is regular
- Explain nondeterminism and describe tools for simulating it with deterministic computation.
- Find equivalent DFA for a given NFA
- Convert between regular expressions and automata
- Classify the computational complexity of a set of strings by determining whether it is regular
- Explain the limits of the class of regular languages
- Identify some nonregular sets
- Use the pumping lemma to prove that a given language is not regular.
- Justify why the Pumping Lemma is true
- Apply the Pumping Lemma in proofs of nonregularity
- Use precise notation to formally define the state diagram of PDA and use clear English to describe computations of PDA informally.
- Define push-down automata informally and formally
- Trace the computation of a push-down automaton
- Determine the language recognized by a given PDA
- Design push-down automata to recognize specific languages
- Determine whether a language is recognizable by a (D or N) FA and/or a PDA
- Use context-free grammars and relate them to languages and pushdown automata.
- Identify the components of a formal definition of a context-free grammar (CFG)
- Derive strings in the language of a given CFG
- Determine the language of a given CFG
- Design a CFG generating a given language

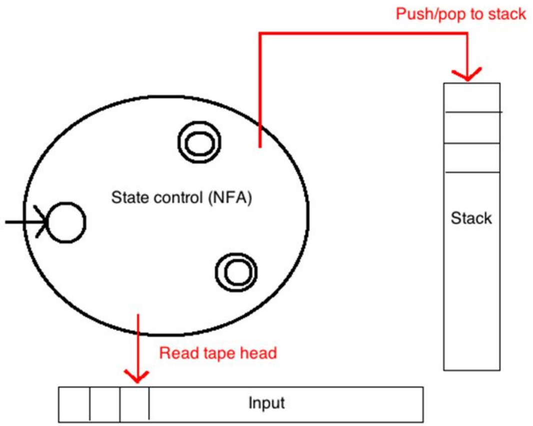
To study for the exam, we recommend reviewing class notes (e.g. annotations linked on the class website, podcast, supplementary video from the class website), reviewing homework (and its posted sample solutions), and in particular *working examples* (extra examples in lecture notes, textbook examples listed in hw, review quizze, discussion examples) and getting feedback (office hours and Piazza).

https://theory-cs.github.io/website/unit6.html

Pushdown Automata (PDA)

Push-Down Automata

- NFA + Stack for (more) powerful computations



Edge label notation (when a, b, c are characters)

- Label a , b ; c or a , b $\longrightarrow$ c means
  - Read an a from the input
  - Pop b from the stack
  - Push c to the stack

Review the formal definition of a PDA

DEFINITION 2.13

A ***pushdown automaton*** is a 6-tuple $(Q, \Sigma, \Gamma, \delta, q_0, F)$, where Q, Σ, Γ, and F are all finite sets, and

1. Q is the set of states,
2. Σ is the input alphabet,
3. Γ is the stack alphabet,
4. $\delta \colon Q \times \Sigma_\varepsilon \times \Gamma_\varepsilon \longrightarrow \mathcal{P}(Q \times \Gamma_\varepsilon)$ is the transition function,
5. $q_0 \in Q$ is the start state, and
6. $F \subseteq Q$ is the set of accept states.

Sipser Definition 2.13, Pg 113

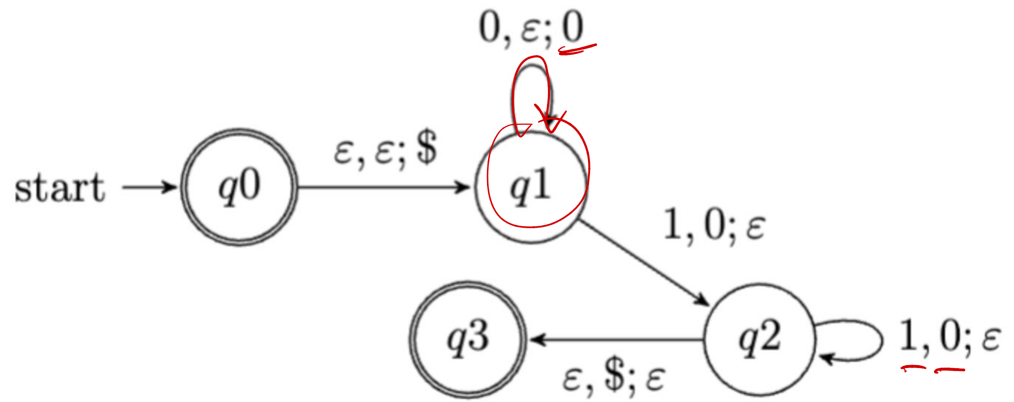
$Q \times \Sigma_\varepsilon \times \Gamma_\varepsilon \to \mathcal{P}(Q \times \Gamma)$

$Q \times \Gamma$

Review the formal definition of a PDA

RQ4.1. Pushdown Automata

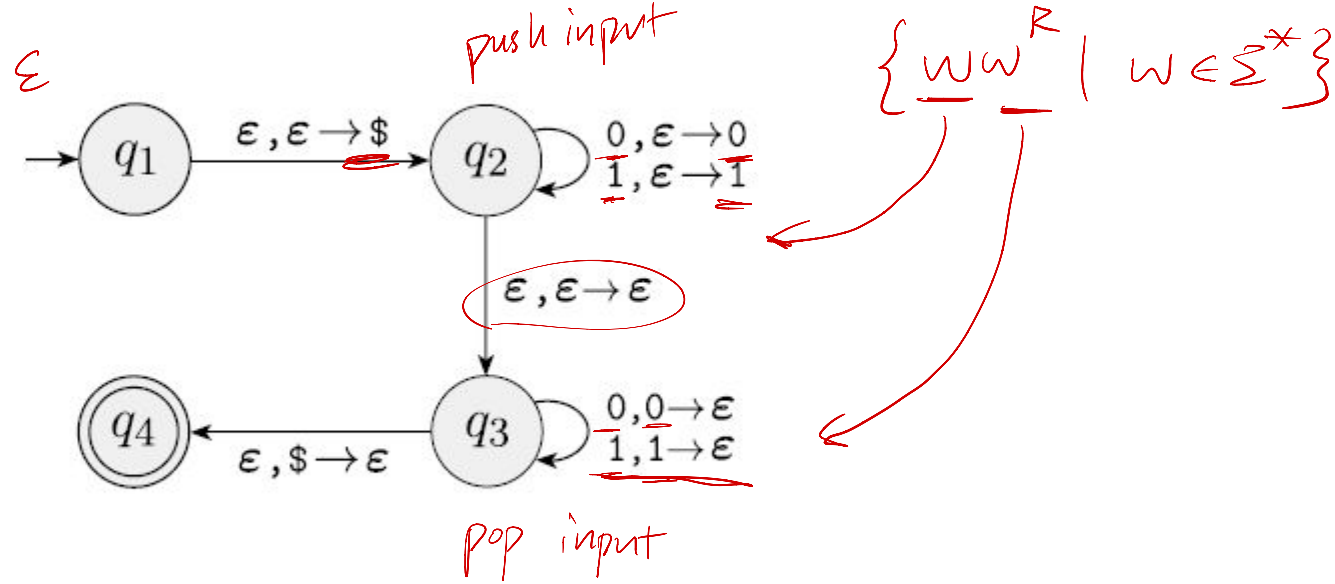
Consider the Pushdown Automaton (PDA), M, with input alphabet $\Sigma = \{0, 1\}$, stack alphabet $\Gamma = \{0, \$\}$ and state diagram:

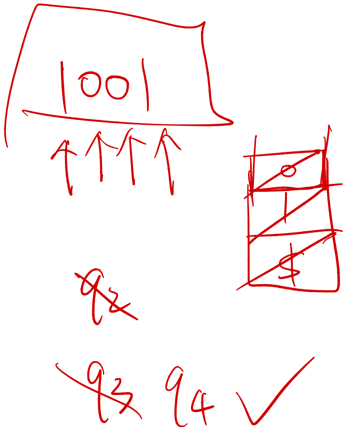


If the transition function of this PDA is called δ, select all and only the true statements about it.

- [ ] $\delta(\ (q2, 1, \varepsilon)\ ) = \{(1, q2)\}$ ✗
- [ ] $\delta(\ (q2, 1, \varepsilon)\ ) = q1$ ✗
- [x] $\delta(\ (q1, 0, \varepsilon)\ ) = \{(q1, 0)\}$
- [x] $\delta(\ (q2, 1, \varepsilon)\ ) = \emptyset$

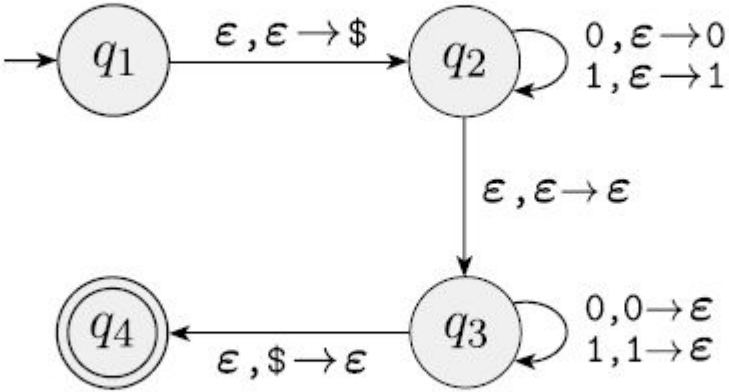
What language does this PDA recognize ?





Sipser Figure 2.19, Pg 116

What language does this PDA recognize ?



$$L(M) = \{ww^R | w \in \{0,1\}^*\}$$

Sipser Figure 2.19, Pg 116

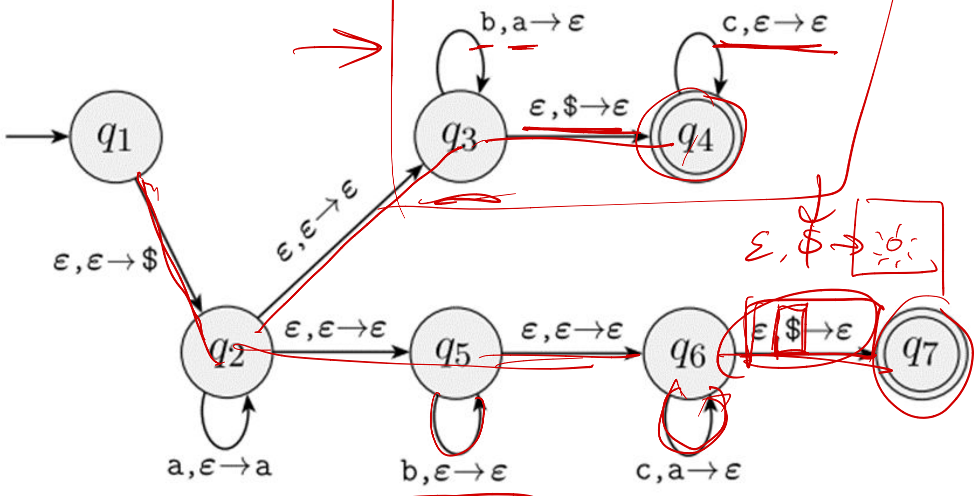
Trace the computation

For each of the following strings, find a computation path that results in the string being accepted. If there is none, draw out the tree of all possible computation paths.

(a) aabcc

(b) abcc

(c) bcc



Sipser Figure 2.17, Pg 116

Context-free Grammars (CFG)

Review the formal definition of a CFG

$\Sigma = \{0,1\}$

$A \to 0$

$L = \{0\}$

$(\{A\}, \{0,1\}, \{A \to 0\}, A\}$

DEFINITION 2.2

A ***context-free grammar*** is a 4-tuple (V, Σ, R, S), where

1. V is a finite set called the ***variables***,
2. Σ is a finite set, disjoint from V, called the ***terminals***,
3. R is a finite set of ***rules***, with each rule being a variable and a string of variables and terminals, and
4. $S \in V$ is the start variable.

Sipser Definition 2.2, Pg 104

A CFG generates a language

If u, v, and w are strings of variables and terminals, and $A \to w$ is a rule of the grammar, we say that uAv ***yields*** uwv, written $uAv \Rightarrow uwv$. Say that u ***derives*** v, written $u \overset{*}{\Rightarrow} v$, if $u = v$ or if a sequence $u_1, u_2, \ldots, u_k$ exists for $k \geq 0$ and

$$u \Rightarrow u_1 \Rightarrow u_2 \Rightarrow \ldots \Rightarrow u_k \Rightarrow v.$$

The ***language of the grammar*** is $\{w \in \Sigma^* | \; S \overset{*}{\Rightarrow} w\}$.

Sipser, Pg 104

CFG practice

RQ4.8. Context-free grammars

Is there a context-free grammar G with $L(G) = \emptyset$?

- [ ] Yes, for example the grammar $(\{S\}, \{0,1\}, \{S \to S\}, S)$.
- [ ] No, because $\emptyset$ is a regular language.
- [ ] Yes, for example the grammar $(\{S,T\}, \{0,1\}, \{S \to \varepsilon\}, T)$.
- [ ] No, because every set of terminals has at least one character.

Select all possible options that apply.

CFG practice

2.3 Answer each part for the following context-free grammar G.

$$
\begin{aligned}
R &\rightarrow XRX \mid S \\
S &\rightarrow \mathtt{a}T\mathtt{b} \mid \mathtt{b}T\mathtt{a} \\
T &\rightarrow XTX \mid X \mid \varepsilon \\
X &\rightarrow \mathtt{a} \mid \mathtt{b}
\end{aligned}
$$

- **a.** What are the variables of G?
- **b.** What are the terminals of G?
- **c.** Which is the start variable of G?
- **d.** Give three strings in $L(G)$.
- **e.** Give three strings *not* in $L(G)$.
- **f.** True or False: $T \Rightarrow \mathtt{aba}$.
- **g.** True or False: $T \overset{*}{\Rightarrow} \mathtt{aba}$.
- **h.** True or False: $T \Rightarrow T$.
- **i.** True or False: $T \overset{*}{\Rightarrow} T$.
- **j.** True or False: $XXX \overset{*}{\Rightarrow} \mathtt{aba}$.
- **k.** True or False: $X \overset{*}{\Rightarrow} \mathtt{aba}$.
- **l.** True or False: $T \overset{*}{\Rightarrow} XX$.
- **m.** True or False: $T \overset{*}{\Rightarrow} XXX$.
- **n.** True or False: $S \overset{*}{\Rightarrow} \varepsilon$.
- **o.** Give a description in English of $L(G)$.

Sipser Exercises 2.3, Pg 155

CFG design

Design CFG to generate the following languages over $\{0, 1\}$:

$$\{ww^{\mathcal{R}} \mid w \in \{0,1\}^*\}$$

$$\{w \mid w = w^{\mathcal{R}}, \text{ that is, } w \text{ is a palindrome}\}$$

Are these languages regular? How to prove that?

Regular and Context-free Languages

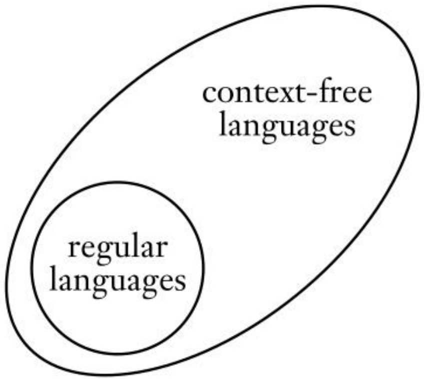
Regular and context-free languages

Over a fixed alphabet Σ, a language L is **regular**

iff it is described by some regular expression
iff it is recognized by some DFA
iff it is recognized by some NFA

Over a fixed alphabet Σ, a language L is **context-free**

iff it is generated by some CFG
iff it is recognized by some PDA



Sipser Figure 2.33, Pg 125