

# Foundations of Computing

## Lecture 11

Arkady Yerukhimovich

February 20, 2024

- 1 Lecture 10 Review
- 2 The CFG Pumping Lemma
- 3 Midterm Review

# Lecture 10 Review

- $\text{CFG} == \text{PDA}$ 
  - Construct PDA from CFG
  - Construct CFG from PDA
- CFG Pumping Lemma

# Lecture 10 Review

- CFG  $\equiv$  PDA
  - Construct PDA from CFG
  - Construct CFG from PDA
- CFG Pumping Lemma

## Today

- Midterm review

# Outline

- 1 Lecture 10 Review
- 2 The CFG Pumping Lemma
- 3 Midterm Review

# The CFG Pumping Lemma

## Theorem

If  $L$  is a CFL, then there exists a pumping length  $p$  s.t. for any  $s \in L$ , with  $|s| \geq p$ ,  $s$  can be divided into 5 pieces  $s = uvxyz$  satisfying:

- 1 For each  $i \geq 0$ ,  $uv^i xy^i z \in L$
- 2  $|vy| > 0$
- 3  $|vxy| \leq p$

- Last week we saw how to use this to prove languages not context-free

# The CFG Pumping Lemma

## Theorem

If  $L$  is a CFL, then there exists a pumping length  $p$  s.t. for any  $s \in L$ , with  $|s| \geq p$ ,  $s$  can be divided into 5 pieces  $s = uvxyz$  satisfying:

- 1 For each  $i \geq 0$ ,  $uv^i xy^i z \in L$
- 2  $|vy| > 0$
- 3  $|vxy| \leq p$

- Last week we saw how to use this to prove languages not context-free
- But, we did not explain why this lemma is true

# Proving the CFG Pumping Lemma (Intuition)



# Outline

- 1 Lecture 10 Review
- 2 The CFG Pumping Lemma
- 3 Midterm Review**

## ① DFA

- Know what it means for a DFA to accept a string
- Know what it means for DFA to accept/recognize a language

- ① DFA
  - Know what it means for a DFA to accept a string
  - Know what it means for DFA to accept/recognize a language
- ② Building DFA
  - Given description of  $L$  build a DFA for it

## ① DFA

- Know what it means for a DFA to accept a string
- Know what it means for DFA to accept/recognize a language

## ② Building DFA

- Given description of  $L$  build a DFA for it

## ③ Regular Languages

- Know what they are
- Recall closure properties of regular languages (complement, union, intersection, concatenation,  $*$  closure)

## ① DFA

- Know what it means for a DFA to accept a string
- Know what it means for DFA to accept/recognize a language

## ② Building DFA

- Given description of  $L$  build a DFA for it

## ③ Regular Languages

- Know what they are
- Recall closure properties of regular languages (complement, union, intersection, concatenation,  $*$  closure)

## ④ NFAs

## ① DFA

- Know what it means for a DFA to accept a string
- Know what it means for DFA to accept/recognize a language

## ② Building DFA

- Given description of  $L$  build a DFA for it

## ③ Regular Languages

- Know what they are
- Recall closure properties of regular languages (complement, union, intersection, concatenation,  $*$  closure)

## ④ NFAs

- Know when an NFA accepts a string/language
- Know when it doesn't accept a string/language

## ① DFA

- Know what it means for a DFA to accept a string
- Know what it means for DFA to accept/recognize a language

## ② Building DFA

- Given description of  $L$  build a DFA for it

## ③ Regular Languages

- Know what they are
- Recall closure properties of regular languages (complement, union, intersection, concatenation,  $*$  closure)

## ④ NFAs

- Know when an NFA accepts a string/language
- Know when it doesn't accept a string/language
- Remember the definition of non-determinism – only need there to exist a part to accept

## ① DFA

- Know what it means for a DFA to accept a string
- Know what it means for DFA to accept/recognize a language

## ② Building DFA

- Given description of  $L$  build a DFA for it

## ③ Regular Languages

- Know what they are
- Recall closure properties of regular languages (complement, union, intersection, concatenation,  $*$  closure)

## ④ NFAs

- Know when an NFA accepts a string/language
- Know when it doesn't accept a string/language
- Remember the definition of non-determinism – only need there to exist a part to accept
- Be able to build an NFA from a language description



## 1 DFA

- Know what it means for a DFA to accept a string
- Know what it means for DFA to accept/recognize a language

## 2 Building DFA

- Given description of  $L$  build a DFA for it

## 3 Regular Languages

- Know what they are
- Recall closure properties of regular languages (complement, union, intersection, concatenation,  $*$  closure)

## 4 NFAs

- Know when an NFA accepts a string/language
- Know when it doesn't accept a string/language
- Remember the definition of non-determinism – only need there to exist a part to accept
- Be able to build an NFA from a language description
- NFA to DFA using the finger method

## ⑤ Regular Expressions

## ⑤ Regular Expressions

- Be able to build an RE for a language

## ⑤ Regular Expressions

- Be able to build an RE for a language
- RE to NFA
- NFA to RE

## 5 Regular Expressions

- Be able to build an RE for a language
- RE to NFA
- NFA to RE

## 6 Regular Language Pumping Lemma

## ⑤ Regular Expressions

- Be able to build an RE for a language
- RE to NFA
- NFA to RE

## ⑥ Regular Language Pumping Lemma

- Remember statement as sequence of quantifiers

## ⑤ Regular Expressions

- Be able to build an RE for a language
- RE to NFA
- NFA to RE

## ⑥ Regular Language Pumping Lemma

- Remember statement as sequence of quantifiers
- Understand why it is true (state of NFA must repeat)

## 5 Regular Expressions

- Be able to build an RE for a language
- RE to NFA
- NFA to RE

## 6 Regular Language Pumping Lemma

- Remember statement as sequence of quantifiers
- Understand why it is true (state of NFA must repeat)
- Understand how to use it.



## 5 Regular Expressions

- Be able to build an RE for a language
- RE to NFA
- NFA to RE

## 6 Regular Language Pumping Lemma

- Remember statement as sequence of quantifiers
- Understand why it is true (state of NFA must repeat)
- Understand how to use it.
- Also know how to prove languages not regular using closure properties

## 7 PDAs

## 7 PDAs

- Know what strings/languages are accepted
- Know what strings/languages are not

## 7 PDAs

- Know what strings/languages are accepted
- Know what strings/languages are not
- Be able to build a PDA given a language description

- 7 PDAs
  - Know what strings/languages are accepted
  - Know what strings/languages are not
  - Be able to build a PDA given a language description
- 8 Context-free Grammars (CFG)

- ⑦ PDAs
  - Know what strings/languages are accepted
  - Know what strings/languages are not
  - Be able to build a PDA given a language description
- ⑧ Context-free Grammars (CFG)
  - Remember what this means

## 7 PDAs

- Know what strings/languages are accepted
- Know what strings/languages are not
- Be able to build a PDA given a language description

## 8 Context-free Grammars (CFG)

- Remember what this means
- Be able to construct one from language description

## 7 PDAs

- Know what strings/languages are accepted
- Know what strings/languages are not
- Be able to build a PDA given a language description

## 8 Context-free Grammars (CFG)

- Remember what this means
- Be able to construct one from language description
- Remember what a derivation is and what a parse tree is



## 7 PDAs

- Know what strings/languages are accepted
- Know what strings/languages are not
- Be able to build a PDA given a language description

## 8 Context-free Grammars (CFG)

- Remember what this means
- Be able to construct one from language description
- Remember what a derivation is and what a parse tree is
- $\text{PDA} \implies \text{CFG}$  (at a high level)

## 9 CFL pumping lemma

- There will not be any questions on the CFL pumping lemma on the exam
- But, there will be on the next homework

# Exam Format

- 7 questions – most have multiple parts
- Covers most of the material outlined above
- 2 questions requiring proofs, the rest are more constructive
- Some yes/no questions

## Don't Forget

- Exam is in class on Thursday 11:10-12:25, don't be late!
- You can bring two  $8.5 \times 11$  piece of paper

# Any Questions?