

Ex C2 Part 1 "CF"

- What is the most important property of "context-freeness"?
- Why the constraint $|LHS| = 1$ leads to a CFG (cf. unrestricted grammars)?
- What kind of strings cannot be specified by CFGs?
- What are the differences between top-down and bottom-up parsing?
- Why can PDAs capture all of CFLs?
- Why is the class of DCFLs smaller than that of CFLs?

CSC460 C3

1

Review

Languages/Grammars/Automata

- Language: Set of strings ~ problems
- Grammar: Convenient way of specifying languages
 - E.g., $L(G)$ for a CFG G
- Automata: Abstract model of processing languages
 - E.g., $L(G)$, where G is a CFG, needs a PDA
- Chomsky hierarchy: about languages

CSC460 C3

2

Ex C2 Part 2 Graphing Tool

- Expression is a series of Terms connected with + or -
 - $Exp \rightarrow Term \quad Exp \rightarrow Term + Exp \quad Exp \rightarrow Term - Exp$
 - Note: $Exp \rightarrow \epsilon$
- Term is a series of Factors connected with * or /
 - $Term \rightarrow Factor \quad Term \rightarrow Factor * Term \quad Term \rightarrow Factor / Term$
- Factor is Const, Factorial, Log, or Power
 - $Factor \rightarrow Const \mid Factorial \mid Log \mid Power$
- Const
 - $Const \rightarrow Digits \quad Const \rightarrow Digits "." Digits \quad Digits \rightarrow Digit$
 - $Digits \rightarrow Digit Digits \quad Digit \rightarrow 0 \mid 1 \mid \dots \mid 9$
- Factorial $\rightarrow "n!"$
- Log
 - $Log \rightarrow "logn" \mid "log" Const "n" \mid "log(" Exp ")" \mid "log" Constant "(" Exp ")"$
 - Note: "log" Const Const is illegal
 - Note: not including "log(" Constant ")"(" Exp ")")
- Power
 - $Power \rightarrow ("n" \mid "(" Exp ")" \mid Const) ["*" ("n" \mid "(" Exp ")" \mid Const)]$

CSC460 C3

3

Ex C2 Part 3 Foxtrot Question C

- Possible to represent the same language with a regular grammar?

```

Foxtrot   → Basic
Basic     → promenadePosition naturalWeave threeStep heelTurn Basic
Basic     → featherFinish WeaveNext
WeaveNext → weave WeaveNext
WeaveNext → threeStep heelTurn [ Basic ]
    
```

CSC460 C3

4

Top-down Parsing

Demo

- Example: Recursive-descent parsing
 - Keep the current input and the remaining part of the rule on stack
 - Expand nonterminals
 - Check terminals against the input

- $S \rightarrow A$
- $A \rightarrow a A b \mid a b$

CSC460 C3

5

Bottom-up Parsing

Demo

- Example: Shift-reduce parsing
 - If the RHS of a rule matches a part of the input, **reduce** it to the LHS symbol
 - Otherwise, push the leftmost symbol onto the stack and repeat (**shift**)

<http://www.tcnj.edu/~komagata/ShiftReduce/>

CSC460 C3

6

Main CF Property

- Matching growth to the left and right
 - E.g., $(\underbrace{\dots}_{h_1} \underbrace{\dots}_{h_2})$, $0^n 1^n$, $a_1 \dots a_n a_n \dots a_1$
- Characterized by *balanced* rules
 - E.g., $A \rightarrow (A)$, $A \rightarrow 0A1$, $A \rightarrow a_i A a_i$
- Characterized by the use of stack
 - E.g., pushing '(' and later popping it and matching with ')'

$0^n 1^n 2^k 3^k$, $0^n 2^k 3^k 1^n$, $0^i 2^k 3^k 0^{(n-i)} 1^n$

CSC460 C3

7

Group Exercise 1 (CF)

- Which of the following languages is context-free? Why?
 1. $0^n 1^n 2^k 3^k$
 2. $0^n 2^k 3^k 1^n$
 3. $0^i 2^k 3^k 0^{(n-i)} 1^n$

CSC460 C3

8

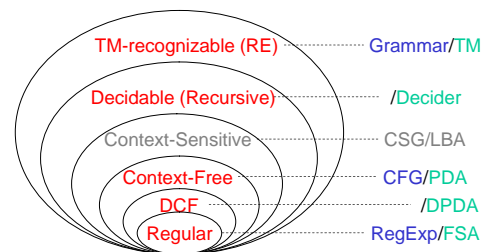
Unit C3: Overview

- Review examples of regular languages
 - Review grammars/automata for regular languages
- Understand the effects of nondeterminism for different automata
- Preview Exercise C3 “Regular”

CSC460 C3

9

Extended Chomsky Hierarchy



CSC460 C3

10

Initial survey problem

Vending Machine

- Imagine a vending machine that accepts nickels, dimes, and quarters and sells Vodka by a paper cup for 35 cents. Draw a diagram of a finite-state machine/automaton that would represent the state of the vending machine. Make additional assumptions if necessary.

CSC460 C3

11

Password Screening

- Requirement 1
 - Your password must be non-TM-recognizable.
- Requirement 2
 - Your password must be a palindrome.
- Requirement 3
 - Your password must be at least 6 characters long, contain at least one uppercase letter, *and* at least one symbol or a numeral.

CSC460 C3

12

Java Comments

- End-of-line comment: `// ...` (until EOL)
- Traditional comment: `/* ... */`
 - Note: No nesting allowed

CSC460 C3

13

Binary Arithmetic

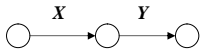
- Do we need a TM?
- Do we need a PDA?

CSC460 C3

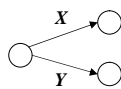
14

FA Components

- Sequence



- Alternative



- Repetition (zero or more times)

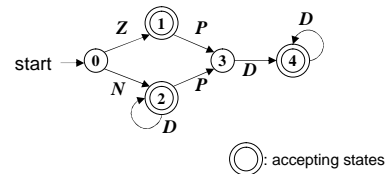


CSC460 C3

15

Finite-State Automata (FSA)

- A machine that consists of the three components shown on the previous slide
- Example



CSC460 C3

16

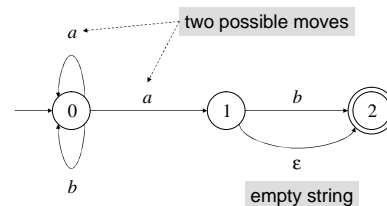
Deterministic FA (DFA)

- $\mathbf{M} = (Q, \Sigma, \delta, q_0, F)$
 - Q : set of states
 - Σ : set of input symbols
 - δ : transition function
 $Q \times \Sigma \rightarrow Q$
 - q_0 : initial state $\in Q$
 - F : set of final states $\subseteq Q$

CSC460 C3

17

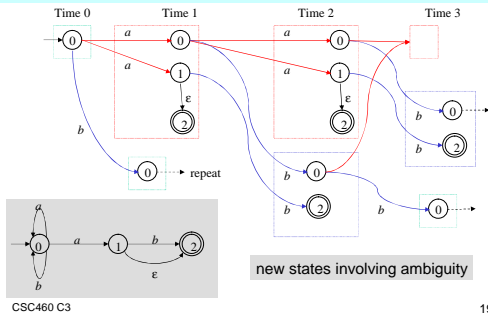
Nondeterministic Example



CSC460 C3

18

Tracing Example 1



Observation

- Due to nondeterminism and ϵ -transitions, the result of a move can be a combination of possible next states.
- For a set of states Q , such a combination is a member of the power set of Q , i.e., $P(Q)$.
 - E.g., $P(\{1, 2\}) = \{\emptyset, \{1\}, \{2\}, \{1, 2\}\}$
- Naturally, any DFA is an NFA.

CSC460 C3

20

Nondeterministic FA (NFA)

- $\mathbf{M} = (Q, \Sigma, \delta, q_0, F)$
 - Q : set of states
 - Σ : set of input symbols
 - δ : transition function
 - $Q \times (\Sigma \cup \{\epsilon\}) \rightarrow P(Q)$ **cf. transition relation**
 - q_0 : initial state $\in Q$
 - F : set of final states $\subseteq Q$

CSC460 C3

21

Nondeterminism

- Possibility of multiple transitions from a state on a single input
- Instead of a single state, considering a set of states
 - S as the set of all states
 - $P(S)$ as the power set of S
 - The next state $\in P(S)$
- Power set ~ nondeterminism
 - exponential growth**

CSC460 C3

22

Effects of Nondeterminism

- TMs: No effects on power (i.e., the set of languages that can be recognized)
- PDAs: Affects the power (i.e., PDAs \neq DPDAs)
- FSAs: No effects on the power (i.e., the set of languages that can be recognized)
- Efficiency/speed: exponential slow down

CSC460 C3

23

Regular Grammar

- $\mathbf{G} = (N, T, R, S)$
 - N : finite set of nonterminals [upper case]
 - T : finite set of terminals [lower case]
 - R : finite set of rules $A \rightarrow \alpha B$ or $A \rightarrow B \alpha$ where
 - α is a string made up of the elements of T
 - $B \in N$
 - S : start symbol $\in N$
- **Regular languages** = $\{\mathbf{L(G)} \mid \mathbf{G}$ is a regular grammar $\}$

CSC460 C3

24

Regular Expression (RegExp)

- Practical alternative to regular grammar
- Expression components (with sets X, Y)
 - Sequence: XY Notation: $X^n = \underbrace{X \dots X}_n$
 - Alternative: $X | Y$
 - Repetition (zero or more times): X^* Kleene closure

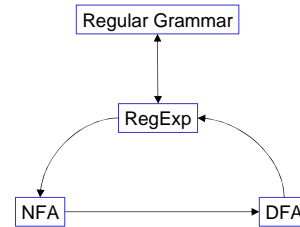
A single regular expression represents a **set**.
Convention: Singleton $\{a\}$ abbreviated as **a**

- Equivalence of RegExp and regular grammars

CSC460 C3

25

Equivalence



CSC460 C3

26

Main *Regular* Property

- Language: Uncoordinated repetition
 - E.g., $0^m 1^n$, “a **very very** long noun phrase”
- Grammar: Characterized by recursion *at edge*
 - E.g., $A \rightarrow aA, A \rightarrow Aa$
 - Equivalently, Kleene closure, X^*
- Automaton: Characterized by finite states
 - E.g., looping represented by a **cycle**

CSC460 C3

27

More Examples

- FP numbers
- URL
- Spanish spelling (cf. English spelling)
- Noun phrase in English (except for CF components)
- Musical code sequence (?)
- Your daily tasks

CSC460 C3

28

Unit Summary

- Main property of regular languages: *uncoordinated* repetition, cf. CF
- RegExp is a practical way of specifying a broad range of simple languages.
- FA is a simple and fast model for processing regular languages.

CSC460 C3

29

Summary Question

- How far have you been thinking Exercise C5 (Comprehensive Exercise)? Explain. If not yet, what is your current thought?
- Questions/Comments/Suggestions

CSC460 C3

30