

What can we do with a DFA?

- check that a # in binary is div. by k
- check that length of string is div. k
- search for any substring
- inexact pattern matching (insert any's)

Regular Languages

- sequencing $A \cdot B$
- Branching $A + B$
- Repetition A^*

Context-free

- everything from reg. lang.
- recursion

What can't we do w/ a DFA?

- check that a string is a palindrome
- check that the # of 0s and 1s is equal
- check that a # is prime

A language is context-free if it can be generated by a context-free grammar.

ex

$S \rightarrow A$

$S \rightarrow B$

$A \rightarrow OA$

$A \rightarrow OC$

$B \rightarrow B1$

$B \rightarrow C1$

$C \rightarrow \epsilon$

$C \rightarrow OC1$

S

A

OA

OOA

OOOA

OOOOC

OOOOC1

OOOOC1

OOOOC1

Σ : symbols/terminals/alphabet $\{0,1\}$

Γ : non-terminals

$\{S, A, B, C\}$

$\Sigma \cap \Gamma = \emptyset$

R: production rules of form $D \rightarrow w$,
where $D \in \Gamma$ and $w \in (\Sigma \cup \Gamma)^*$

$S \in \Gamma$: starting non-terminal

$L(C)$ = set of string generated by C

$L(C) = \{0^n 1^n : n \geq 0\}$

$$G = (\Sigma, \Gamma, R, S)$$

$$L(G) = L(S)$$

$(01)^*$

the language of grammar G

$$S \rightarrow A | B$$

$$A \rightarrow 0A | 0C$$

$$B \rightarrow B1 | C1$$

$$C \rightarrow \epsilon | 0C1$$

could do:

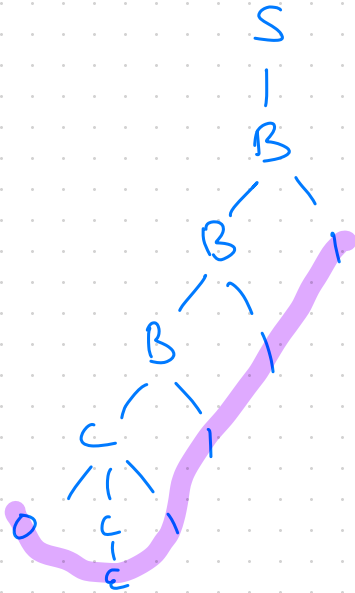
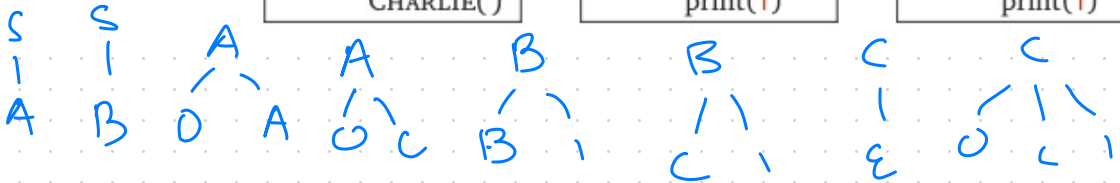
$$S \rightarrow AB | AA$$

```
START():
  if █
    ALPHA()
  else
    BRAVO()
```

```
ALPHA():
  if █
    print(0)
    ALPHA()
  else
    print(0)
    CHARLIE()
```

```
BRAVO():
  if █
    BRAVO()
    print(1)
  else
    CHARLIE()
    print(1)
```

```
CHARLIE():
  if █
    return
  else
    print(0)
    CHARLIE()
    print(1)
```



Q: 57:

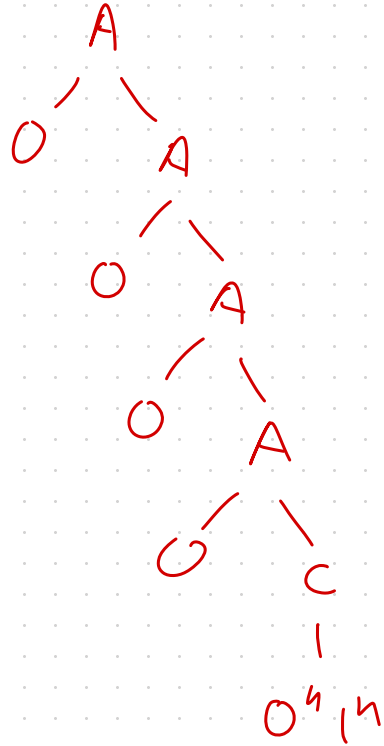
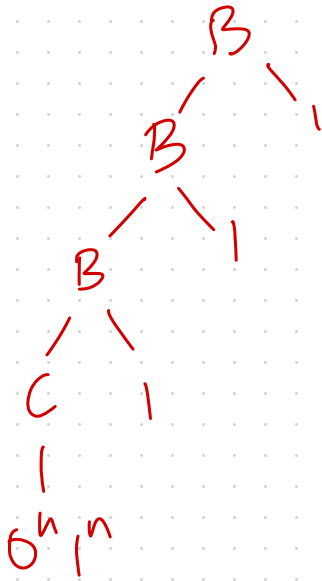
What is $L(G)$? $\{0^m 1^n : n \neq m\}$

$S \rightarrow A | B \rightarrow \{0^n 1^m : n \neq m\}$

$A \rightarrow 0A | 0C \rightarrow \{0^n 1^m : n > m\}$

$B \rightarrow B1 | C1 \rightarrow \{0^n 1^m : m > n\}$

$C \rightarrow \varepsilon | \underline{0C1} \rightarrow \{0^n 1^n : n \geq 0\}$



$$0^* 1^* \setminus \{0^n 1^m : n \neq m\} = \{0^n 1^n : n \geq 0\}$$

Strings w/ equal # of 0s and 1s.

1010 0011 1001

$S \rightarrow \epsilon \mid 0S1 \mid 1S0 \mid SS$

with partner

balanced brackets $[[[]]]$

$[[[]][[]][[]]]][[]][[]]$

$S \rightarrow \epsilon \mid [S] \mid SS$

⟨sentence⟩ → ⟨noun phrase⟩⟨verb phrase⟩⟨noun phrase⟩

⟨noun phrase⟩ → ⟨adjective phrase⟩⟨noun⟩

⟨adj. phrase⟩ → ⟨article⟩ | ⟨possessive⟩ | (⟨adjective phrase⟩⟨adjective⟩)

⟨verb phrase⟩ → ⟨verb⟩ | ⟨adverb⟩⟨verb phrase⟩

⟨noun⟩ → dog | trousers | daughter | nose | homework | time lord | pony | ...

⟨article⟩ → the | a | some | every | that | ...

⟨possessive⟩ → ⟨noun phrase⟩'s | my | your | his | her | ...

⟨adjective⟩ → friendly | furious | moist | green | severed | timey-wimey | little | ...

⟨verb⟩ → ate | found | wrote | killed | mangled | saved | invented | broke | ...

⟨adverb⟩ → squarely | incompetently | barely | sort of | awkwardly | totally | ...

