

# 1. Regular Expressions & CFGs

Let  $\Sigma = \{1, 2, 3, 4\}$  and let  $L$  be the language containing all strings  $w$  where  $w$  is **nondecreasing**.

We define a string  $w$  as **nondecreasing** if for all  $k$ , the character at index  $k$  is greater than or equal to all the numbers to the left, meaning the number at index  $k$  is greater than or equal to each of the numbers at indices  $0$  to  $k-1$ .

Some example **nondecreasing** strings are:

- $\varepsilon$  (this string vacuously meets the definition of nondecreasing)
- 13 ( $3 \geq 1$ )
- 11223334

An example on an invalid string (one not in the language) is 132, as  $3 > 2$ .

- (a) Write a regular expression that matches  $L$ . (No explanation required).

**Solution:**

$(1)^*(2)^*(3)^*(4)^*$

- (b) Write a CFG that generates  $L$ .

Be sure to tell us which symbol is the start symbol; also include a sentence or two of explanation of how your CFG works.

**Solution:**

**S** is the start symbol. If the string is nondecreasing then all 1's come before any 2's which come before 3's which come before 4's. So we split the string into those sections (any or all of which can be the empty string).

$S \rightarrow ABCD \mid \varepsilon$   
 $A \rightarrow 1A \mid \varepsilon$   
 $B \rightarrow 2B \mid \varepsilon$   
 $C \rightarrow 3C \mid \varepsilon$   
 $D \rightarrow 4D \mid \varepsilon$

## 2. DFAs & NFAs

Let  $\Sigma = \{0, 1\}$  for all problems on this page. For the following questions, show how to describe exactly the language specified using the given formalism (regular expression, DFA, NFA or CFG).

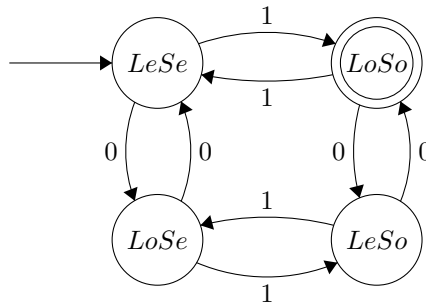
- (a) Write a **regular expression** that matches all binary strings that do not end in 11.

Briefly (1-2 sentences) explain your solution. **Solution:**

$(1^*0^*0)^*(1 \cup \varepsilon)$

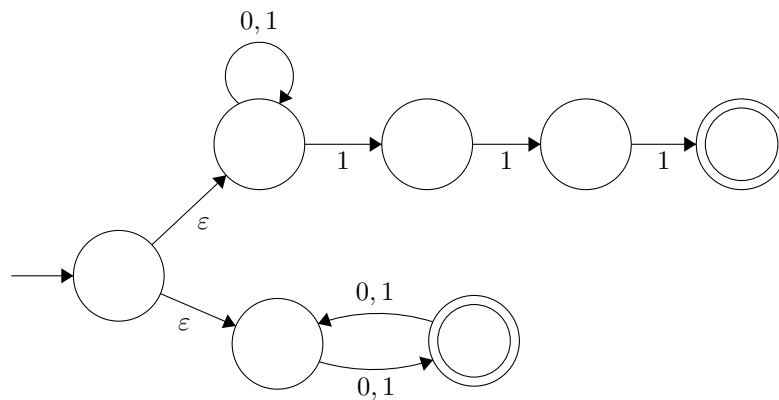
- (b) Draw a **DFA** that accepts all binary strings that meet both of the following conditions: the string has an odd length **and** the digits of the string sum to an odd number.

Briefly (1-2 sentences) explain your solution. **Solution:**



- (c) Draw an **NFA** that accepts all binary strings with an odd length **or** end in 111 (or both).

Briefly (1-2 sentences) explain your solution. **Solution:**



- (d) Write a context-free grammar that generates all binary strings that meet **both** of the following conditions:

- The string contains strictly more 0s than 1s
- In the string, all 0s appear before 1s **or** all 1s appear before 0s.

For example, 0, 00, 100, 001 should be in the language but  $\varepsilon$ , 1, 01, 010 should not be in the language.

If your grammar has multiple non-terminal symbols, describe the purpose of each non-terminal (1 sentence per non-terminal symbol), and tell us which is one is the start symbol.

**Solution:**

$$S \rightarrow A \mid B$$
$$A \rightarrow 1A0 \mid A0 \mid 0$$
$$B \rightarrow 0B1 \mid 0B \mid 0$$

$S$  splits into cases on the **or** condition: either having all 1s first or all 0s first.

$A$  generates all binary strings with more 1s than 0s where all 1s appear before 0s.

$B$  generates all binary strings with more 1s than 0s where all 0s appear before 1s.

### 3. Treeshake

We define simple binary trees as the recursive set  $\mathcal{B}$ :

**Basis Step:**  $\bullet \in \mathcal{B}$ .

**Recursive Step:** If  $L, R \in \mathcal{B}$ , then  $(L, \bullet, R) \in \mathcal{B}$ .

Define the following functions on simple binary trees:

$$\begin{aligned} \text{edges}(t) &= \begin{cases} 0 & \text{if } t = \bullet \\ 2 + \text{edges}(L) + \text{edges}(R) & \text{if } t = (L, \bullet, R) \end{cases} \\ \text{degree}(t) &= \begin{cases} 1 & \text{if } t = \bullet \\ 3 & \text{if } t = (L, \bullet, R) \end{cases} \\ \text{sum}(t) &= \begin{cases} \text{degree}(t) & \text{if } t = \bullet \\ \text{degree}(t) + \text{sum}(L) + \text{sum}(R) & \text{if } t = (L, \bullet, R) \end{cases} \end{aligned}$$

Prove that for all  $t \in \mathcal{B}$ ,  $\text{sum}(t) = 2 \cdot \text{edges}(t) + 1$ .

You must use structural induction for this problem.

**Solution:**

Let  $P(t) := \text{sum}(t) = 2 \cdot \text{edges}(t) + 1$ . We will prove  $P(t)$  holds for all  $t \in \mathcal{B}$  by structural induction.

**Basis Step:** From the definitions,

$$\text{sum}(\bullet) = \text{degree}(\bullet) = 1$$

and

$$2 \cdot \text{edges}(\bullet) + 1 = 2(0) + 1 = 1$$

Since both are 1, the base case is satisfied.

**Inductive Hypothesis:** Suppose  $P(L)$  and  $P(R)$  for some arbitrary simple binary trees  $L, R$ .

**Inductive Step:** Let  $t = (L, \bullet, R)$ . We will show  $P(t)$ .

$$\begin{aligned} \text{sum}(t) &= \text{degree}(t) + \text{sum}(L) + \text{sum}(R) \\ &= 3 + \text{sum}(L) + \text{sum}(R) \\ &= 3 + 2 \cdot \text{edges}(L) + 1 + 2 \cdot \text{edges}(R) + 1 && \text{by IH} \\ &= 5 + 2 \cdot \text{edges}(L) + 2 \cdot \text{edges}(R) \\ &= 1 + 2(2) + 2 \cdot \text{edges}(L) + 2 \cdot \text{edges}(R) \\ &= 1 + 2(2 + \text{edges}(L) + \text{edges}(R)) \\ &= 1 + 2 \cdot \text{edges}(t) \end{aligned}$$

This proves  $P(t)$ .

**Conclusion:** Therefore,  $P(t)$  holds for all  $t \in \mathcal{B}$  by structural induction.

## 4. Training Wheels

Let your domain of discourse be integers. Define the following predicates

- $\text{PerfectSquare}(x)$  returns true if and only if  $x$  is a perfect square (that is  $x = z^2$  for some  $z \in \mathbb{Z}$ ).
- $\text{Divides}(x, y)$  returns true if and only if  $x|y$ .
- $\text{Even}(x)$  returns true if and only if  $x$  is even.

(a) Translate this sentence into predicate logic notation.

If  $x$  is even and  $x$  is a perfect square then  $4|x$ .

**Solution:**

$$\forall x([\text{Even}(x) \wedge \text{PerfectSquare}(x)] \rightarrow \text{Divides}(4, x))$$

(b) Translate this sentence into predicate logic notation.

Every even number is divisible by two different integers.

**Solution:**

$$\forall x \exists y \exists z (\text{Even}(x) \rightarrow [\text{Divides}(y, x) \wedge \text{Divides}(z, x) \wedge y \neq z])$$

(c) Negate the following sentence (leave your answer as symbols; negations must be applied to single predicates, but you do not need to reformat your answer to show domain restriction).

$$\forall x \forall y ([\text{Even}(x) \wedge \text{Even}(y)] \rightarrow [\text{Divides}(x, y) \vee \text{Divides}(y, x)])$$

**Solution:**

$$\exists x \exists y (\text{Even}(x) \wedge \text{Even}(y) \wedge \neg \text{divides}(x, y) \wedge \neg \text{Divides}(y, x))$$

(d) State **in English** the contrapositive of the **original** statement in the last part (not your answer). Your English contrapositive must use domain restriction. **Solution:**

For all  $x$  and  $y$  if  $x$  does not divide  $y$  and  $y$  does not divide  $x$  then at least one of  $x$  and  $y$  is not even.

(e) The (original) statement in (c) is [1 point]

True

False **Solution:**

True

(f) The contrapositive of the statement in (c) [i.e., the correct answer to part (d)] has: [1 point]

The same truth value as part (e).

The negation of the truth value in part (e).

It depends on what  $x$  and  $y$  are. **Solution:**

Same truth value.

*Additional space to work on problems*