CSE 311 Section 08

Induction, Regular Expressions, CFGs

Administrivia

Announcements & Reminders

- Homework 6 was due Wednesday (2/21)
- Midterm grades have been released!
    - Regrade requests are open
    - Concerns about grades: Read Robbie's post on Ed!
- Check your section participation grade on gradescope
    - If it different than what you expect, let your TA know

Recursively Defined Sets

Recursive Definition of Sets

Define a set S as follows:

Basis Step:
Describe the basic starting elements in your set
ex: $0 \in S$

Recursive Step:
Describe how to derive new elements of the set from previous elements
ex: If $x \in S$ then $x + 2 \in S$.

Exclusion Rule: Every element of S is in S from the basis step (alone) or a finite number of recursive steps starting from a basis step.

Problem 3 – Recursively Defined Sets

For each of the following, write a recursive definition of the sets satisfying the following properties. Briefly justify that your solution is correct.

a) Binary strings of even length.

a) Binary strings not containing 10.

a) Binary strings not containing 10 as a substring and having at least as many 1s as 0s.

a) Binary strings containing at most two 0s and at most two 1s.

Work on this problem with the people around you.

Problem 3 – Recursively Defined Sets

For each of the following, write a recursive definition of the sets satisfying the following properties. Briefly justify that your solution is correct.

a) Binary strings of even length.

Generate accepted and rejected strings first!

Problem 3 – Recursively Defined Sets

For each of the following, write a recursive definition of the sets satisfying the following properties. Briefly justify that your solution is correct.

a) Binary strings of even length.

Generate accepted and rejected strings first!

Step 1: Write out basic cases and more intricate cases

Problem 3 – Recursively Defined Sets

For each of the following, write a recursive definition of the sets satisfying the following properties. Briefly justify that your solution is correct.

a) Binary strings of even length.

Generate accepted and rejected strings first!

Step 1: Write out basic cases and more intricate cases

Accepted Strings	Rejected Strings
ε	0
11	1
10101010	101010**1**
10101011	**0**10101011

Problem 3 – Recursively Defined Sets

For each of the following, write a recursive definition of the sets satisfying the following properties. Briefly justify that your solution is correct.

a) Binary strings of even length.

Generate accepted and rejected strings first!

Step 1: Write out basic cases and more intricate cases

Step 2: Find a pattern!

Accepted Strings	Rejected Strings
ε	0
11	1
10101010	101010**1**
10101011	**0**10101011

Problem 3 – Recursively Defined Sets

For each of the following, write a recursive definition of the sets satisfying the following properties. Briefly justify that your solution is correct.

a) Binary strings of even length.

Generate accepted and rejected strings first!

Step 1: Write out basic cases and more intricate cases

Accepted Strings	Rejected Strings
ε	0
11	1
10101010	101010**1**
10101011	**0**10101011

Step 2: Find a pattern!

All even-length strings can be generated from **a series of substrings** of **length 2!**

All possible substrings of length 2 are:
10, 01, 11, 00

Problem 3 – Recursively Defined Sets

For each of the following, write a recursive definition of the sets satisfying the following properties. Briefly justify that your solution is correct.

a) Binary strings of even length.

Step 3: Write out Basis and Recursive step

Problem 3 – Recursively Defined Sets

For each of the following, write a recursive definition of the sets satisfying the following properties. Briefly justify that your solution is correct.

a) Binary strings of even length.

Step 3: Write out Basis and Recursive step

Basis: $\varepsilon \in S$

Recursive Step: If $x \in S$, then $x00, x01, x10, x11 \in S$

Problem 3 – Recursively Defined Sets

For each of the following, write a recursive definition of the sets satisfying the following properties. Briefly justify that your solution is correct.

a) Binary strings of even length.

Step 3: Write out Basis and Recursive step

Basis: $\varepsilon \in S$

Recursive Step: If $x \in S$, then $x00, x01, x10, x11 \in S$

Step 4: check that you cannot build the rejected strings and only build accepted strings with the recursive step

Problem 3 – Recursively Defined Sets

For each of the following, write a recursive definition of the sets satisfying the following properties. Briefly justify that your solution is correct.

a) Binary strings of even length.

Step 3: Write out Basis and Recursive step

Basis: $\varepsilon \in S$

Recursive Step: If $x \in S$, then $x00, x01, x10, x11 \in S$

Step 4: check that you cannot build the rejected strings and only build accepted strings with the recursive step

Accepted Strings	Rejected Strings
ε	0
11	1
10101010	101010**1**
10101011	**0**10101011

Problem 3 – Recursively Defined Sets

For each of the following, write a recursive definition of the sets satisfying the following properties. Briefly justify that your solution is correct.

b) Binary strings not containing 10.

Problem 3 – Recursively Defined Sets

For each of the following, write a recursive definition of the sets satisfying the following properties. Briefly justify that your solution is correct.

b) Binary strings not containing 10.

Step 1: Write out basic cases and more intricate cases

Problem 3 – Recursively Defined Sets

For each of the following, write a recursive definition of the sets satisfying the following properties. Briefly justify that your solution is correct.

b) Binary strings not containing 10.

Step 1: Write out basic cases and more intricate cases

Accepted Strings	Rejected Strings
1	010
0	10
ε	100
111	1110
00001	100001

Problem 3 – Recursively Defined Sets

For each of the following, write a recursive definition of the sets satisfying the following properties. Briefly justify that your solution is correct.

b) Binary strings not containing 10.

Step 1: Write out basic cases and more intricate cases

Step 2: Find a pattern!

Accepted Strings	Rejected Strings
1	010
0	10
ε	100
111	1110
00001	100001

Problem 3 – Recursively Defined Sets

For each of the following, write a recursive definition of the sets satisfying the following properties. Briefly justify that your solution is correct.

b) Binary strings not containing 10.

Step 1: Write out basic cases and more intricate cases

Accepted Strings	Rejected Strings
1	010
0	10
ε	100
111	1110
00001	100001

Step 2: Find a pattern!

0's and 1's cannot be in the same string **unless** 0's **come first** and 1's **come second**

0's should be **built from the left** (0x)
1's should be **built from the right** (x1)

such strings that have 1's and 0's can only look like: 000…1111

Problem 3 – Recursively Defined Sets

For each of the following, write a recursive definition of the sets satisfying the following properties. Briefly justify that your solution is correct.

b) Binary strings not containing 10.

Step 3: Write out Basis and Recursive step

Problem 3 – Recursively Defined Sets

For each of the following, write a recursive definition of the sets satisfying the following properties. Briefly justify that your solution is correct.

b) Binary strings not containing 10.

Step 3: Write out Basis and Recursive step

If the string does not contain 10, then the first 1 in the string can only be followed by more 1s. Hence, it must be of the form $0^m 1^n$ for some $m, n \in \mathbb{N}$.

Basis: $\varepsilon \in S$

Recursive Step: If $x \in S$, then $0x \in S$ and $x1 \in S$

Step 4: check that you cannot build the rejected strings and only build accepted strings with the recursive step :)

Problem 3 – Recursively Defined Sets

For each of the following, write a recursive definition of the sets satisfying the following properties. Briefly justify that your solution is correct.

c) Binary strings not containing 10 as a substring and having at least as many 1s as 0s.

Problem 3 – Recursively Defined Sets

For each of the following, write a recursive definition of the sets satisfying the following properties. Briefly justify that your solution is correct.

c) Binary strings not containing 10 as a substring and having at least as many 1s as 0s.

Step 1: Write out basic cases and more intricate cases

Problem 3 – Recursively Defined Sets

For each of the following, write a recursive definition of the sets satisfying the following properties. Briefly justify that your solution is correct.

c) Binary strings not containing 10 as a substring and having at least as many 1s as 0s.

Step 1: Write out basic cases and more intricate cases

Accepted Strings	Rejected Strings
1	0**10**
01	**0**
ε	100
111	11**10**
00001111	**00001**

Problem 3 – Recursively Defined Sets

For each of the following, write a recursive definition of the sets satisfying the following properties. Briefly justify that your solution is correct.

c) Binary strings not containing 10 as a substring and having at least as many 1s as 0s.

Step 1: Write out basic cases and more intricate cases

Step 2: Find a pattern!

Accepted Strings	Rejected Strings
1	0**10**
01	**0**
ε	100
111	11**10**
00001111	**00001**

Problem 3 – Recursively Defined Sets

For each of the following, write a recursive definition of the sets satisfying the following properties. Briefly justify that your solution is correct.

c) Binary strings not containing 10 as a substring and having at least as many 1s as 0s.

Step 1: Write out basic cases and more intricate cases

Accepted Strings	Rejected Strings
1	010
01	**0**
ε	100
111	1110
00001111	**00001**

Step 2: Find a pattern!

From part (b) we know:
0's should be **built from the left** (0x)
1's should be **built from the right** (x1)

New restriction for adding a 0:
for every 0 we add, there must be **at least** an additional 1 accompanying it so we always have **# 1's ≥ # 0's**

So lets change: 0x to 0x1

Problem 3 – Recursively Defined Sets

For each of the following, write a recursive definition of the sets satisfying the following properties. Briefly justify that your solution is correct.

c) Binary strings not containing 10 as a substring and having at least as many 1s as 0s.

Step 3: Write out Basis and Recursive step

Problem 3 – Recursively Defined Sets

For each of the following, write a recursive definition of the sets satisfying the following properties. Briefly justify that your solution is correct.

c) Binary strings not containing 10 as a substring and having at least as many 1s as 0s.

Step 3: Write out Basis and Recursive step

These must be of the form $0^m 1^n$ for some $m, n \in \mathbb{N}$ with $m \leq n$. We can ensure that by pairing up the 0s with 1s as they are added:

Basis: $\varepsilon \in S$.

Recursive Step: If $x \in S$, then $0x1 \in S$ and $x1 \in S$.

Step 4: <u>check</u> that you cannot build the rejected strings and only build accepted strings with the recursive step :)

Regular Expressions

Regular Expressions

Basis:

- ε is a regular expression. The empty string itself matches the pattern (and nothing else does).
- $\varnothing$ is a regular expression. No strings match this pattern.
- a is a regular expression, for any $a \in \Sigma$ (i.e. any character). The character itself matching this pattern.

Recursive:

- If A, B are regular expressions then $(A \cup B)$ is a regular expression. matched by any string that matches A or that matches B [or both]).
- If A, B are regular expressions then AB is a regular expression. matched by any string x such that $x = yz$, y matches A and z matches B.
- If A is a regular expression, then $A*$ is a regular expression. matched by any string that can be divided into 0 or more strings that match A.

Regular Expressions

A regular expression is a recursively defined set of strings that form a language.

A regular expression will generate all strings in a language, and won't generate any strings that ARE NOT in the language

Hints:

- Come up with a few examples of strings that ARE and ARE NOT in your language
- Then, after you write your regex, check to make sure that it CAN generate all of your examples that are in the language, and it CAN'T generate those that are not

Problem 1 – Regular Expressions

a) Write a regular expression that matches base 10 numbers (e.g., there should be no leading zeroes).

b) Write a regular expression that matches all base-3 numbers that are divisible by 3.

c) Write a regular expression that matches all binary strings that contain the substring "111", but not the substring "000".

d) Write a regular expression that matches all binary strings that do not have any consecutive 0's or 1's.

e) Write a regular expression that matches all binary strings of the form $1^{k}y$, where $k \geq 1$ and $y \in \{0,1\}^*$ has at least k 1's.

Work on this problem with the people around you.

Problem 1 – Regular Expressions

a) Write a regular expression that matches base 10 numbers (e.g., there should be no leading zeroes).

base-10 numbers:

Our everyday numbers!
Notice we have 10 symbols (0-9) to represent numbers.

256: (**2** * 10^2) + (**5** * 10^1) + (**6** * 10^0)

base-2 numbers: (binary)

10: (**1** * 2^1) + (**0** * 2^0)

Problem 1 – Regular Expressions

a) Write a regular expression that matches base 10 numbers (e.g., there should be no leading zeroes).

Representing numbers all possible *strings* using numbers 0-9:

Problem 1 – Regular Expressions

a) Write a regular expression that matches base 10 numbers (e.g., there should be no leading zeroes).

Representing numbers all possible *strings* using numbers 0-9:

$$(0 \cup 1 \cup 2 \cup 3 \cup 4 \cup 5 \cup 6 \cup 7 \cup 8 \cup 9)^*$$

Problem 1 – Regular Expressions

a) Write a regular expression that matches base 10 numbers (e.g., there should be no leading zeroes).

Representing numbers all possible *strings* using numbers 0-9:

$(0 \cup 1 \cup 2 \cup 3 \cup 4 \cup 5 \cup 6 \cup 7 \cup 8 \cup 9)^*$

⚠️ "<u>0</u>101" or "<u>0</u>91" **are not** Base-10 numbers

Problem 1 – Regular Expressions

a) Write a regular expression that matches base 10 numbers (e.g., there should be no leading zeroes).

Representing numbers all possible *strings* using numbers 0-9:

$(0 \cup 1 \cup 2 \cup 3 \cup 4 \cup 5 \cup 6 \cup 7 \cup 8 \cup 9)^*$

⚠️ “0101” or “091” **are not** Base-10 numbers

All possible *strings* using numbers 0-9 that never start with 0

Problem 1 – Regular Expressions

a) Write a regular expression that matches base 10 numbers (e.g., there should be no leading zeroes).

Representing numbers all possible *strings* using numbers 0-9:

(0 U 1 U 2 U 3 U 4 U 5 U 6 U 7 U 8 U 9)*

⚠️ "0101" or "091" **are not** Base-10 numbers

All possible *strings* using numbers 0-9 that never start with 0

(1 U 2 U 3 U 4 U 5 U 6 U 7 U 8 U 9)(0 U 1 U 2 U 3 U 4 U 5 U 6 U 7 U 8 U 9)*

Problem 1 – Regular Expressions

a) Write a regular expression that matches base 10 numbers (e.g., there should be no leading zeroes).

Representing numbers all possible *strings* using numbers 0-9:

$(0 \cup 1 \cup 2 \cup 3 \cup 4 \cup 5 \cup 6 \cup 7 \cup 8 \cup 9)^*$

⚠️ “<u>0</u>101” or “<u>0</u>91” **are not** Base-10 numbers

All possible *strings* using numbers 0-9 that never start with 0

$(1 \cup 2 \cup 3 \cup 4 \cup 5 \cup 6 \cup 7 \cup 8 \cup 9)(0 \cup 1 \cup 2 \cup 3 \cup 4 \cup 5 \cup 6 \cup 7 \cup 8 \cup 9)^*$

⚠️ “<u>0</u>” **is** a Base-10 number not considered

Problem 1 – Regular Expressions

a) Write a regular expression that matches base 10 numbers (e.g., there should be no leading zeroes).

Representing numbers all possible *strings* using numbers 0-9:

$(0 \cup 1 \cup 2 \cup 3 \cup 4 \cup 5 \cup 6 \cup 7 \cup 8 \cup 9)^*$

⚠️ "0101" or "091" **are not** Base-10 numbers

All possible *strings* using numbers 0-9 that never start with 0

$(1 \cup 2 \cup 3 \cup 4 \cup 5 \cup 6 \cup 7 \cup 8 \cup 9)(0 \cup 1 \cup 2 \cup 3 \cup 4 \cup 5 \cup 6 \cup 7 \cup 8 \cup 9)^*$

⚠️ "0" **is** a Base-10 number not considered

All possible *strings* using numbers 0-9 that never start with 0 *or* is 0

Problem 1 – Regular Expressions

a) Write a regular expression that matches base 10 numbers (e.g., there should be no leading zeroes).

Representing numbers all possible *strings* using numbers 0-9:

$(0 \cup 1 \cup 2 \cup 3 \cup 4 \cup 5 \cup 6 \cup 7 \cup 8 \cup 9)^*$

⚠️ "0101" or "091" **are not** Base-10 numbers

All possible *strings* using numbers 0-9 that never start with 0

$(1 \cup 2 \cup 3 \cup 4 \cup 5 \cup 6 \cup 7 \cup 8 \cup 9)(0 \cup 1 \cup 2 \cup 3 \cup 4 \cup 5 \cup 6 \cup 7 \cup 8 \cup 9)^*$

⚠️ "0" **is** a Base-10 number not considered

All possible *strings* using numbers 0-9 that never start with 0 *or* is 0

$0 \cup ((1 \cup 2 \cup 3 \cup 4 \cup 5 \cup 6 \cup 7 \cup 8 \cup 9)(0 \cup 1 \cup 2 \cup 3 \cup 4 \cup 5 \cup 6 \cup 7 \cup 8 \cup 9)^*)$

Problem 1 – Regular Expressions

a) Write a regular expression that matches base 10 numbers (e.g., there should be no leading zeroes).

Representing numbers all possible *strings* using numbers 0-9:

$(0 \cup 1 \cup 2 \cup 3 \cup 4 \cup 5 \cup 6 \cup 7 \cup 8 \cup 9)^*$

⚠️ "<u>0</u>101" or "<u>0</u>91" **are not** Base-10 numbers

All possible *strings* using numbers 0-9 that never start with 0

$(1 \cup 2 \cup 3 \cup 4 \cup 5 \cup 6 \cup 7 \cup 8 \cup 9)(0 \cup 1 \cup 2 \cup 3 \cup 4 \cup 5 \cup 6 \cup 7 \cup 8 \cup 9)^*$

⚠️ "<u>0</u>" **is** a Base-10 number not considered

All possible *strings* using numbers 0-9 that never start with 0 *or* is 0

$0 \cup ((1 \cup 2 \cup 3 \cup 4 \cup 5 \cup 6 \cup 7 \cup 8 \cup 9)(0 \cup 1 \cup 2 \cup 3 \cup 4 \cup 5 \cup 6 \cup 7 \cup 8 \cup 9)^*)$

✅ Generates only all possible Base-10 numbers

Problem 1 – Regular Expressions

b) Write a regular expression that matches all base-3 numbers that are divisible by 3.

Problem 1 – Regular Expressions

b) Write a regular expression that matches all base-3 numbers that are divisible by 3.

Write a regular expression that matches all base-3 numbers

Problem 1 – Regular Expressions

b) Write a regular expression that matches all base-3 numbers that are divisible by 3.

Write a regular expression that matches all base-3 numbers

$0 \cup ((1 \cup 2)(0 \cup 1 \cup 2)*)$

✅Generates only all possible Base-3 numbers

Problem 1 – Regular Expressions

b) Write a regular expression that matches all base-3 numbers that are divisible by 3.

Write a regular expression that matches all base-3 numbers

$$0 \cup ((1 \cup 2)(0 \cup 1 \cup 2)*)$$

Generates only all possible Base-3 numbers

…divisible by 3

Problem 1 – Regular Expressions

b) Write a regular expression that matches all base-3 numbers that are divisible by 3.

Write a regular expression that matches all base-3 numbers

$$0 \cup ((1 \cup 2)(0 \cup 1 \cup 2)*)$$

Generates only all possible Base-3 numbers

…divisible by 3

Hint: you know that Base-<u>10</u> numbers are divisible by <u>10</u> when <u>they end in 0</u> (10, 20, 30, 40…)

Problem 1 – Regular Expressions

b) Write a regular expression that matches all base-3 numbers that are divisible by 3.

Write a regular expression that matches all base-3 numbers

$$0 \cup ((1 \cup 2)(0 \cup 1 \cup 2)*)$$

Generates only all possible Base-3 numbers

…divisible by 3

Hint: you know that Base-10 numbers are divisible by 10 when <u>they end in 0</u> (10, 20, 30, 40…)

$$0 \cup ((1 \cup 2)(0 \cup 1 \cup 2)*\mathbf{0})$$

✅all possible Base-3 numbers divisible by 3

Problem 1 – Regular Expressions

c) Write a regular expression that matches all binary strings that contain the substring “111”, but not the substring “000”.

Problem 1 – Regular Expressions

c) Write a regular expression that matches all binary strings that contain the substring “111”, but not the substring “000”.

all binary strings that contain the substring “111”

Problem 1 – Regular Expressions

c) Write a regular expression that matches all binary strings that contain the substring “111”, but not the substring “000”.

all binary strings that contain the substring “111”

$(0 \cup 1)^* \, 111 \, (0 \cup 1)^*$

⚠️ The Kleene-star has us generating any number of 0’s

Problem 1 – Regular Expressions

c) Write a regular expression that matches all binary strings that contain the substring “111”, but not the substring “000”.

all binary strings that contain the substring “111”

$(0 \cup 1)^* \, 111 \, (0 \cup 1)^*$

⚠️ The Kleene-star has us generating any number of 0’s

…without the substring “000”

Use careful case-work to modify this and produce only 0,1,or two 0’s

Problem 1 – Regular Expressions

c) Write a regular expression that matches all binary strings that contain the substring “111”, but not the substring “000”.

all binary strings that contain the substring “111”

$(0 \cup 1)^* \, 111 \, (0 \cup 1)^*$

⚠️ The Kleene-star has us generating any number of 0’s

…without the substring “000”

Use careful case-work to modify this and produce only 0,1,or two 0’s

$(0 \cup 00 \cup \varepsilon)\,(1)^*\,111\,(0 \cup 00 \cup \varepsilon)\,(1)^*$

Problem 1 – Regular Expressions

c) Write a regular expression that matches all binary strings that contain the substring “111”, but not the substring “000”.

all binary strings that contain the substring “111”

$(0 \cup 1)^* \; 111 \; (0 \cup 1)^*$

⚠️ The Kleene-star has us generating any number of 0’s

…without the substring “000”

Use careful case-work to modify this and produce only 0,1,or two 0’s

$(0 \cup 00 \cup \varepsilon)\,(1)^* \; 111 \; (0 \cup 00 \cup \varepsilon)\,(1)^*$

⚠️ Cannot produce 1’s with “0” or “00” like “<u>1</u>011101<u>1</u>”

Problem 1 – Regular Expressions

c) Write a regular expression that matches all binary strings that contain the substring “111”, but not the substring “000”.

all binary strings that contain the substring “111”

$(0 \cup 1)^* \, 111 \, (0 \cup 1)^*$

⚠️ The Kleene-star has us generating any number of 0’s

…without the substring “000”

Use careful case-work to modify this and produce only 0,1,or two 0’s

$(0 \cup 00 \cup \varepsilon)\,(1)^*\,111\,(0 \cup 00 \cup \varepsilon)\,(1)^*$

⚠️ Cannot produce 1’s with “0” or “00” like “<u>1</u>011101”

$(0 \cup 00 \cup \varepsilon)\,(01 \cup 001 \cup 1)^*\,111\,(0 \cup 00 \cup \varepsilon)\,(01 \cup 001 \cup 1)^*$

Problem 1 – Regular Expressions

c) Write a regular expression that matches all binary strings that contain the substring “111”, but not the substring “000”.

all binary strings that contain the substring “111”

$(0 \cup 1)^*\ 111\ (0 \cup 1)^*$

⚠️ The Kleene-star has us generating any number of 0’s

…without the substring “000”

Use careful case-work to modify this and produce only 0,1,or two 0’s

$\mathbf{(0 \cup 00 \cup \varepsilon)}\ (1)^*\ 111\ \mathbf{(0 \cup 00 \cup \varepsilon)}\ (1)^*$

⚠️ Cannot produce 1’s with “0” or “00” like “<u>1</u>011101<u>1</u>”

$(0 \cup 00 \cup \varepsilon)\ \mathbf{(01 \cup 001 \cup 1)^*}\ 111\ (0 \cup 00 \cup \varepsilon)\ \mathbf{(01 \cup 001 \cup 1)^*}$

⚠️ Generates “000” like “<u>00</u> <u>01</u> 111”

Problem 1 – Regular Expressions

c) Write a regular expression that matches all binary strings that contain the substring “111”, but not the substring “000”.

all binary strings that contain the substring “111”

$(0 \cup 1)^*\ 111\ (0 \cup 1)^*$

⚠️ The Kleene-star has us generating any number of 0’s

…without the substring “000”

Use careful case-work to modify this and produce only 0,1,or two 0’s

$(0 \cup 00 \cup \varepsilon)\ (1)^*\ 111\ (0 \cup 00 \cup \varepsilon)\ (1)^*$

⚠️ Cannot produce 1’s with “0” or “00” like “<u>1</u>011101”

$(0 \cup 00 \cup \varepsilon)\ (01 \cup 001 \cup 1)^*\ 111\ (0 \cup 00 \cup \varepsilon)\ (01 \cup 001 \cup 1)^*$

⚠️ Generates “000” like “<u>00</u> <u>01</u> 111”

$(01 \cup 001 \cup 1)^*\ (0 \cup 00 \cup \varepsilon)\ 111\ (01 \cup 001 \cup 1)^*\ (0 \cup 00 \cup \varepsilon)$

✅ all binary strings with “111” and without “000”

Problem 1 – Regular Expressions

c) Write a regular expression that matches all binary strings that contain the substring “111”, but not the substring “000”.

all binary strings that contain the substring “111”

$(0 \cup 1)^*\ 111\ (0 \cup 1)^*$

⚠️ The Kleene-star has us generating any number of 0’s

…without the substring “000”

Use careful case-work to modify this and produce only 0,1,or two 0’s

$\mathbf{(0 \cup 00 \cup \varepsilon)}\ (1)^*\ 111\ \mathbf{(0 \cup 00 \cup \varepsilon)}\ (1)^*$

⚠️ Cannot produce 1’s with “0” or “00” like “<u>1</u>011101<u>1</u>”

$(0 \cup 00 \cup \varepsilon)\ \mathbf{(01 \cup 001 \cup 1)^*}\ 111\ (0 \cup 00 \cup \varepsilon)\ \mathbf{(01 \cup 001 \cup 1)^*}$

⚠️ Generates “000” like “<u>00</u> <u>01</u> 111”

$(01 \cup 001 \cup 1)^*\ (0 \cup 00 \cup \varepsilon)\ 111\ (01 \cup 001 \cup 1)^*\ (0 \cup 00 \cup \varepsilon)$

✅ all binary strings with “111” and without “000”

$$\mathbf{(01 \cup 001 \cup 1)^*\ (0 \cup 00 \cup \varepsilon)\ 111\ (01 \cup 001 \cup 1)^*\ (0 \cup 00 \cup \varepsilon)}$$

Problem 1 – Regular Expressions

d) Write a regular expression that matches all binary strings that do not have any consecutive 0’s or 1’s.

Problem 1 – Regular Expressions

d) Write a regular expression that matches all binary strings that do not have any consecutive 0’s or 1’s.

Step 1: Write out basic and more intricate cases

Problem 1 – Regular Expressions

d) Write a regular expression that matches all binary strings that do not have any consecutive 0's or 1's.

Step 1: Write out basic and more intricate cases

Accepted Strings	Rejected Strings
ε	00
1	11
10101	1010**1**1
0101	010**0**

Problem 1 – Regular Expressions

d) Write a regular expression that matches all binary strings that do not have any consecutive 0's or 1's.

Step 1: Write out basic and more intricate cases

Accepted Strings	Rejected Strings
ε	00
1	11
10101	1010**11**
0101	01**00**

Step 2: Find a pattern!

Problem 1 – Regular Expressions

d) Write a regular expression that matches all binary strings that do not have any consecutive 0's or 1's.

Step 1: Write out basic and more intricate cases

Accepted Strings	Rejected Strings
ε	00
1	11
10101	1010**11**
0101	01**00**

Step 2: Find a pattern!

strings can be generated from **either a series** of **"01" or "10" substrings**

(1) Using the "01" substring, one <u>additional 0 can be added</u>

(1) Using the "10" substring, one <u>additional 1 can be added</u>

Problem 1 – Regular Expressions

d) Write a regular expression that matches all binary strings that do not have any consecutive 0's or 1's.

Step 3: Write out the expression with the two cases we found

Problem 1 – Regular Expressions

d) Write a regular expression that matches all binary strings that do not have any consecutive 0's or 1's.

Step 3: Write out the expression with the two cases we found

$$((01)^* (0 \cup \varepsilon)) \cup ((10)^* (1 \cup \varepsilon))$$

Problem 1 – Regular Expressions

e) Write a regular expression that matches all binary strings of the form $1^k y$, where $k \geq 1$ and $y \in \{0,1\}^*$ has at least k 1's.

Problem 1 – Regular Expressions

e) Write a regular expression that matches all binary strings of the form $1^k y$, where $k \geq 1$ and $y \in \{0,1\}^*$ has at least k 1's.

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

Explanation: While it may seem like we need to keep track of how many 1's there are, it turns out that we don't. Convince yourself that strings in the language are exactly those of the form 1x, where x is any binary string with at least one 1. Hence, x is matched by the regular expression $(0 \cup 1)*1(0 \cup 1)*$

Context-Free Grammars

Context-Free Grammars

A context free grammar (CFG) is a finite set of production rules over:

- An alphabet Σ of “terminal symbols”
- A finite set V of “nonterminal symbols”
- A start symbol (one of the elements of V) usually denoted S

A production rule for a nonterminal $A \in V$ takes the form

- $A \rightarrow w1 \mid w2 \mid \ldots \mid wk$

Where each $wi \in V \cup \Sigma^*$ is a string of nonterminals and terminals.

Problem 2 – CFGs

Write a context-free grammar to match each of these languages.

a) All binary strings that start with 11.

b) All binary strings that contain at most one 1.

c) All strings over 0, 1, 2 with the same number of 1s and 0s and exactly one 2.

Work on this problem with the people around you.

Problem 2 – CFGs

a) All binary strings that start with 11.

Problem 2 – CFGs

a) All binary strings that start with 11.

Thinking back to regular expressions…

Problem 2 – CFGs

a) All binary strings that start with 11.

Thinking back to regular expressions…

$11\ (0 \cup 1)^*$

Problem 2 – CFGs

a) All binary strings that start with 11.

Thinking back to regular expressions…

$11\ (0 \cup 1)^*$

Now generate the CFG…

Problem 2 – CFGs

a) All binary strings that start with 11.

Thinking back to regular expressions…

$11\ (0 \cup 1)^*$

Now generate the CFG…

$\mathbf{S} \rightarrow 11\mathbf{T}$
$\mathbf{T} \rightarrow 1\mathbf{T} \mid 0\mathbf{T} \mid \varepsilon$

Problem 2 – CFGs

b) All binary strings that contain at most one 1.

Problem 2 – CFGs

b) All binary strings that contain at most one 1.

Thinking back to Regular expressions…

Problem 2 – CFGs

b) All binary strings that contain at most one 1.

Thinking back to Regular expressions…

$0^* (1 \cup \boldsymbol{\varepsilon})\, 0^*$

Problem 2 – CFGs

b) All binary strings that contain at most one 1.

Thinking back to Regular expressions…

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

Now generate the CFG…

Problem 2 – CFGs

b) All binary strings that contain at most one 1.

Thinking back to Regular expressions…

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

Now generate the CFG…

$S \rightarrow ABA$
$A \rightarrow 0A \mid \varepsilon$
$B \rightarrow 1 \mid \varepsilon$

Problem 2 – CFGs

b) All binary strings that contain at most one 1.

Thinking back to Regular expressions…

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

Now generate the CFG…

$$\begin{aligned} S &\rightarrow ABA \\ A &\rightarrow 0A \mid \varepsilon \\ B &\rightarrow 1 \mid \varepsilon \end{aligned}$$

Alternative solution:

$$S \rightarrow 0S \mid S0 \mid 1 \mid 0 \mid \varepsilon$$

Problem 2 – CFGs

c) All strings over 0, 1, 2 with the same number of 1s and 0s and exactly one 2.

Problem 2 – CFGs

c) All strings over 0, 1, 2 with the same number of 1s and 0s and exactly one 2.

~~$S \rightarrow 01S \mid 10S \mid 0S1 \mid 1S0 \mid S01 \mid S10 \mid 2$~~

Counter example: 001121100

Correct Answer:

$S \rightarrow 2T \mid T2 \mid ST \mid TS \mid 0S1 \mid 1S0$

$T \rightarrow TT \mid 0T1 \mid 1T0 \mid \varepsilon$

That's All, Folks!

Thanks for coming to section this week!
Any questions?