Not all sets of strings can be formed by regular expressions. For example:

- The set of palindromes - {E, 0, 1, 00, 11, 010, …}
- A string of 0s followed by an equal-length string of 1's

$$\{0^k 1^k : k \geq 0\}$$

- The set of all binary strings in which each 0-block is at least as long as the 1-block that immediately follows it

These are not *rational languages*

Instead of regular expressions, we can represent these using recursive decompositions/expressions

Recursive Decomposition

A recursive decomposition/expression R is either

$$[\text{regex}] \cup \{\varepsilon, 0, 1\}$$

or one or more expressions involving $\cup$, concatenation, or $*$ over $\{\varepsilon, 0, 1\}$ and R itself

Ex. $R = \varepsilon \cup R(0 \cup 1)$: set of binary strings

$R = \varepsilon \cup 0R0 \cup 1R1 \cup 0 \cup 1$: set of even-length palindromes

$0[\varepsilon$ OR $0R0$ OR $1R1$ OR 0 OR $1]$ 0

every option here produces a palindrome
repeat recursion indefinitely

$R = \varepsilon \cup 0R1$; $\{0^k 1^k : k \geq 0\}$

The set of strings produced by a recursive expression R is the *union* of:

- All regular expressions on the RHS not involving R
- All expressions involving R, with every occurrence of R replaced by a string that has already been produced; repeated until no more new strings are produced
  - May repeat a countably infinite number of times

Ex. $S = 0^* \cup S1$

$\Rightarrow 0^* \cup [0^* \cup S1]1$

$\Rightarrow 0^* \cup [0^* \cup [0 \cup S1]1]1$

$\vdots$

$\Rightarrow 0^* 1^*$

Ex. 2. $R = \underbrace{\varepsilon \cup 0 \cup 1}\, \cup RR$

$\{\varepsilon, 0, 1\} \cup RR$:

$\{\varepsilon, 0, 1, 0, 1, 01, 10 \ldots\}$: for R_1 in S:
for R_2 in S:
(R_1, R_2)

$= \{0, 1\}^*$

Ex. 3. $R = \varepsilon \cup 0 \cup 1 \cup 0R1 \cup 1R0$: set of all strings which equal themselves when flip + reverse order

<u>Ambiguity</u>

A recursive decomposition is unambiguous if there is exactly one way in which any given string in the set is produced

$S = 0^* \cup S1$ is unambiguous

$R = \varepsilon \cup 0 \cup 1 \cup RR$ is ambiguous : $0 = (\varepsilon, 0)$ or (0)

$R = \varepsilon \cup 0 \cup 1 \cup 0R1 \cup 1R0$ is unambiguous:

$01 \stackrel{?}{=} (0, 1)$ ✓ or $(0, \varepsilon, 1)$ ✗

Unambiguous -> can make a generating series

Ex. $R = \varepsilon \cup R(0 \cup 1)$

$$\Phi_R(x) = \Phi_\varepsilon(x) + \Phi_R(x) \cdot \Phi_{0\cup 1}(x)$$

$$\Rightarrow \Phi_R(x) = 1 + \Phi_R(x) \cdot (x + x)$$

$$\Rightarrow \Phi_R(x) = \frac{1}{1-2x}$$

Ex. Find a recursive decomposition for

$S = \{0^j 1^l : j \geq l \geq 1\}$

$S = 0[0^{j-1} 1^{l-1}]1, \quad j-1 \geq l-1 \geq 0$

Recursion stops when $l = 1$ (since $j \geq l$), so:

Base case: $l = 1; \ j \geq 1 \rightarrow 00^*1$

Recursive case: $j, l \geq 2 \rightarrow 0S1$

$\Rightarrow S = 0S1 \cup 00^*1$

Proving that this is unambiguous:

let $s = 0^j 1^l \in S, \quad j \geq l \geq 1$

Case 1: $j \geq l \geq 2$

$s \in 0s'1$

$s' = 0^{j-1}1^{l-1}$: $j-1 \geq l-1 > 1$

$s' \in S \rightarrow s \in 0S1$

So each s can be formed uniquely

Case 2: 00^*1 is unambiguous

Generating series:

$$\Phi_S(x) = x^2 \Phi_S(x) + \frac{1}{1-x}$$

Then solve for $\Phi_S(x)$

Ex.2. Find T = set of strings where if a 1-block follows a 0-block, then the length of the 0-block is greater than the length of the 1-block

$(\underbrace{\varepsilon \text{ or } 1\text{-block}}_{1^*})(\underbrace{0\text{-block or } 1\text{-block}}_{S^*})(\underbrace{0\text{-block or } \varepsilon}_{0^*})$

$$\Phi_T(x) = \Phi_{1^*} \cdot \Phi_{S^*} \cdot \Phi_{0^*}$$

<u>Excluded Substrings</u>

A non-empty string S where k is not a substring of S

Let A = set of strings that exclude 100

Every string in A can only end with 10, 01, or 10, if the rest of the string does not contain 100

It may also end in 00, but only if the rest of the string doesn't contain a 1

$A(0 \cup 1)$ can end with 100 if A ends with 10 and we pick 1 from $(0 \cup 1)$

So: $\epsilon \cup A(0 \cup 1) = A \cup B$

where B is the set of all strings that have 100 only at the end $\rightarrow B \subseteq A100$

$$\Rightarrow \underbrace{1}_{\epsilon} \underbrace{+}_{\cup} \underbrace{[\Phi_A(x)]}_{A} \underbrace{(x+x)}_{(0 \cup 1)} = \Phi_A(x) + \Phi_B(x)$$

$$\Rightarrow 1 + [\Phi_A(x)](2x) = \Phi_A(x) + x^3 \Phi_B(x)$$

Ex.2. Let A = strings that exclude 10010

$\epsilon \cup A(0 \cup 1) = A \cup B$

where B has 10010 only at the end

This is the form for all excluded substrings: adding a 0 or a 1 to an excluded substring A either produces something in A, or something with the excluded substring *only at the end*

$B = A10010$

C - 2 instances of $10010 \rightarrow 10010010$ $\therefore$ added 010

$$\Rightarrow \epsilon \cup A(0 \cup 1) = A \cup B \rightarrow 1 + 2x\Phi_A = \Phi_A + \Phi_B$$
$$A10010010 = B \cup C$$
$$B010 = C$$