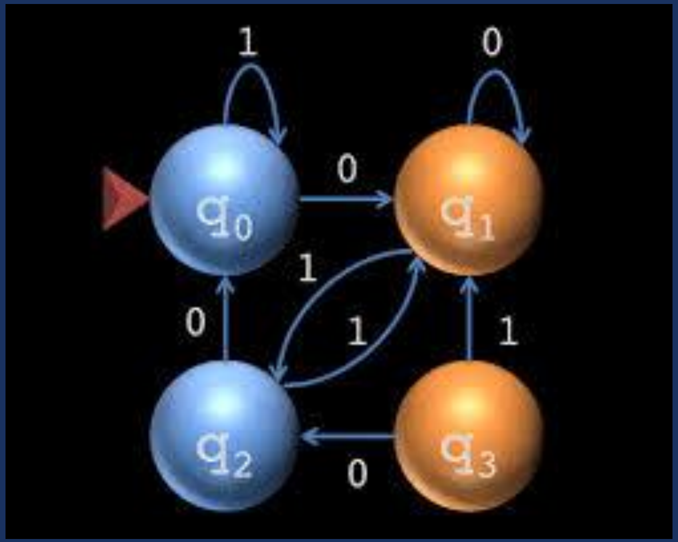
CS313 –THEORY OF AUTOMATA

LECTURE 5: CHAPTER 4: REGULAR EXPRESSION

WEEK 2



Asst. Prof. Syed Faisal Ali
sfaisal@uit.edu
Computer Science E & F

SPRING 2025

DEFINING LANGUAGES BY ANOTHER NEW METHOD

- We wish now to be very careful about the phrases we use to define languages. We defined L1 in Chapter 2 by the symbols:

L1 = $\{x^n$ for n = 1 2 3 ... $\}$ and we presumed that we all understood exactly which values n could take .

- We might even have defined the language L2 by the symbols:
- L2 = $\{x^n$ for n = 1 3 5 7 . . . $\}$ and again we could presume that we all agree on what words are in this language .
- We might define a language by the symbols:
- L5 = $\{x^n$ for n = 1 4 9 16 . . . $\}$ but now the symbols are becoming more of an IQ test than a clear definition .
- What words are in the language? L6 = {x" for n = 3 4 8 22 . . . } ages of the sisters of Louis XIV.

DEFINING LANGUAGES BY ANOTHER NEW METHOD

- We presented one method for indicating this set as the closure of a smaller set.
- Let S = {x} . Then L4 = S*.
- As shorthand for this, we could have written L4 = {x}*.
- We now introduce the use of the Kleene star applied not to a set, but directly to the letter .r and written as a superscript as if it were an exponent: **x***
- The simple expression **x*** will be used to indicate some sequence of x's (maybe none at all).
- This x is intentionally written in boldface type to distinguish it from an alphabet character.
- x^* = Lambda λ or x or x^2or x^3 or x^5 $\ldots$ = x^n for some n = 0 1 2 3 4 …

DEFINING LANGUAGES BY ANOTHER NEW METHOD

- We can think of the star as an unknown power or undetermined power. That is, x^* stands for a string of x's, but we do not specify how many. It stands for any string of x's in the language L4.
- The star operator applied to a letter is analogous to the star operator applied to a set. It represents an arbitrary concatenation of copies of that letter (maybe none at all). This notation can be used to help us de fine languages by writing

  L4 = language(x^*) Since x^* is any string of x's,

- L4 is then the set of all possible is then the set of all possible strings of x's of any length (including Lambda λ)

DEFINING LANGUAGES BY ANOTHER NEW METHOD

- Suppose that we wished to describe the language L over the alphabet $\sum = \{a\ b\}$

  where L = {a ab abb abbb abbbb ...}

- We could summarize this language by the English phrase "all words of the form one a followed by some number of b's (maybe no b's at all)."
- Using our star notation and boldface letters , we may write

  L = language(a b*) or without the space L = language(ab*)

- Whether we put a space inside ab* or not is only for the clarity of reading: it does not change the set of strings this represents.

DEFINING LANGUAGES BY ANOTHER NEW METHOD

- No string can contain a blank unless a blank is a character in the alphabet ∑.
- If we want blanks to be in the alphabet, we normally introduce some special symbol to stand for them, as blanks themselves are invisible to the naked eye.
- The reason for putting a blank between a and b* in the product above is to emphasize the point that the star operator is applied to the b only.
- We have now used a boldface letter without a star as well as with a star.
- We can apply the Kleene star to the whole string ah if we want, as follows:
- (ab)* = Lambda λ or ab or abab or ababab …

DEFINING LANGUAGES BY ANOTHER NEW METHOD

- Parentheses are not letters in the alphabet of this language, so they can be used to indicate factoring without accidentally changing the words.
- Since the star represents some kind of exponentiation, we use it as powers are used in algebra, where by universal understanding the expression xy^2 means $x(y^2)$, not $(xy)^2$.

DEFINING LANGUAGES BY ANOTHER NEW METHOD

- If we want to define the language L1 this way, we may write

$$L1 = \text{language}(xx^*).$$

- This means that we start each word of L1 by writing down an x and then we follow it with some string of x's (which may be no more x's at all).
- Or we may use the + notation from Chapter 2 and write L1 = language(x^+) meaning all words of the form x to some positive power (i.e., not x^0 = Lambda λ).
- The + notation is a convenience, but is not essential since we can say the same thing with *'s alone.

SOLVING xx^*

x	x^*	Out come
x	λ	x
x	x	xx
x	xx	xxx
x	xxx	xxxx
x	x^n	x^{n+1}
x constant x	$x^{infinity}$	$x^{infinity}$

EXAMPLE

- The language L1 can be defined by any of the expressions below:
- xx^* x^+ xx^*x^* x^*xx^* x^+x^* x^*x^+ $x^*x^*x^*xx^*$
- Remember, x^* can always be Lambda λ.

EXAMPLE

- The language defined by the expression ab^*a is the set of all strings of a's and h's that have at least two letters, that begin and end with a's, and that have nothing but b's inside (if anything at all).

$$\text{Language}(ab^*a) = \{\text{ aa aba abba abbba abbbba } \ldots\}$$

- It would be a subtle mistake to say only that this language is the set of all words that begin and end with an a and have only b's in between, because this description may also apply to the word a, depending on how it is interpreted. Our symbolism eliminates this ambiguity.

EXAMPLE

- The language of the expression $a^* b^*$contains all the strings of a's and b's in which all the a's (if any) come before all the b's (if any).

$$\text{Language}(a^* b^*) = \{\lambda \text{ a b aa ab bb aaa aab abb bbb aaaa } \ldots\}$$

- Notice that ba and aba are not in this language.
- Notice also that there need not be the same number of a's and b's.
- Here we should again be very careful to observe that

$$a^* b^* \neq (ab)^*$$

- Since the language defined by the expression on the right contains the word abab, whereas the language defined by the expression on the left does not.
- This cautions us against thinking of the * as a normal algebraic exponent.

EXAMPLE

- The language defined by the expression a*b*a* contains the word baa since it starts with zero a's followed by one b followed by two a's.

EXAMPLE

- The following expressions both define the language L2 = $\{x^{odd}\}$

(xx)*x or x(xx)*

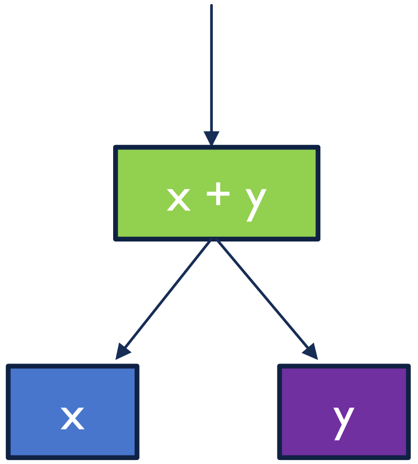
- but the expression

x*xx*

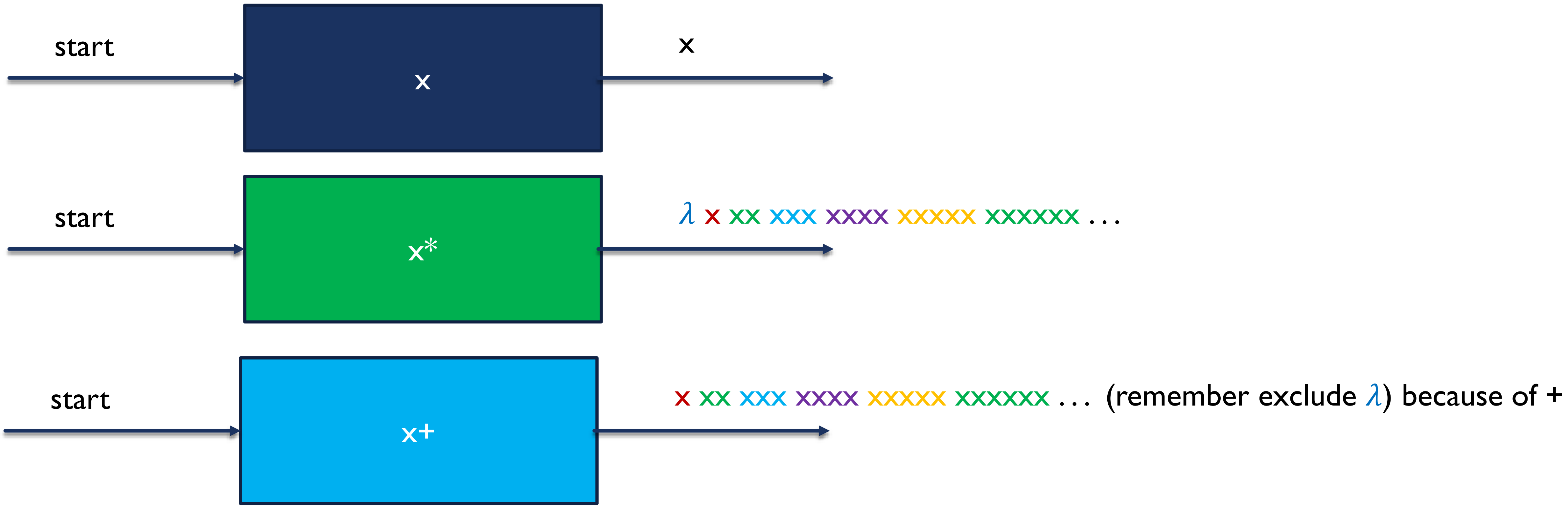
- does not since it includes the word (xx) x (x).

EXTENDED KLEEENE (+) AND OPTION OF EITHER (+)

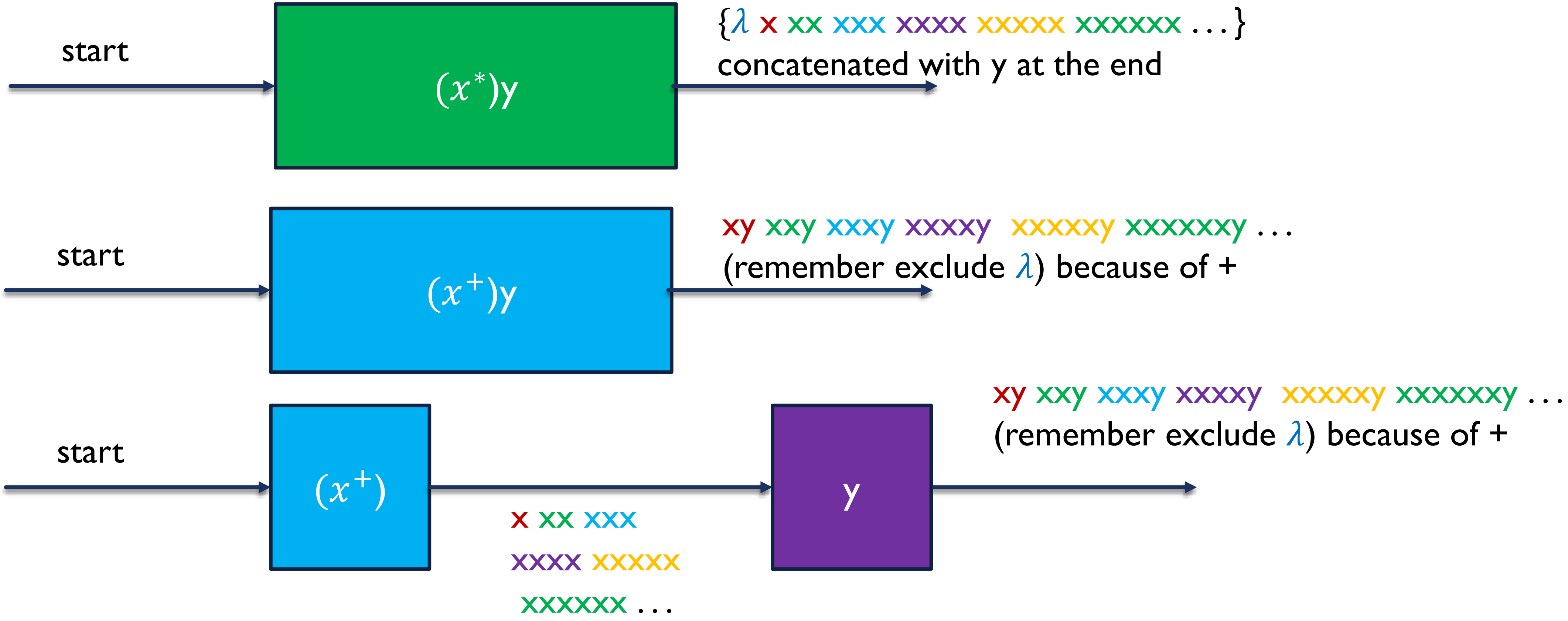
- We now introduce another use for the plus sign.
- By the expression x + y where x and y are strings of characters from an alphabet, we mean "either x or y."
- This means that x + y offers a choice, much the same way that x* does.
- Care should be taken so as not to confuse this with + as an exponent.



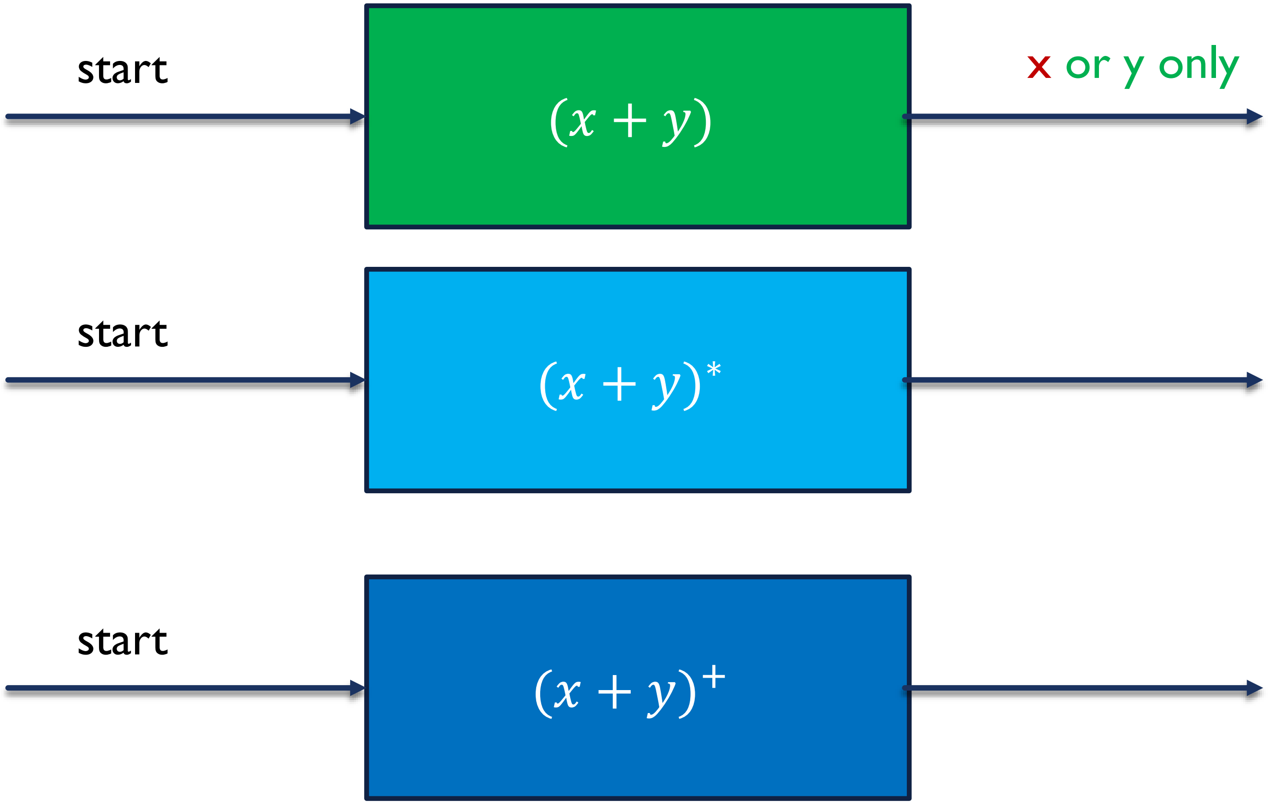
HOW DOES THIS WORKS?



HOW DOES THIS WORKS?



HOW DOES THIS WORKS?



λ x xx xxx xxxx xxxxx xxxxxx . . .
λ y yy yyy yyyy yyyyy yyyyyy . . .
λ xy xyy xyyy xyyyy xyyyyy xyyyyyy . . .
λ yx yxx yxxx yxxxx yxxxxx yxxxxxxx . . .
And all possible combinations.

x xx xxx xxxx xxxxx xxxxxx . . .
y yy yyy yyyy yyyyy yyyyyy . . .
xy xyy xyyy xyyyy xyyyyy xyyyyyy . . .
yx yxx yxxx yxxxx yxxxxx yxxxxxxx . . .
And all possible combinations without λ

All possible combinations = *This means that any alphabet can come first, between and last at any iteration.*

HOW DOES THIS WORKS?

start → $(x+y)(x+y)(x+y)$ → xxx, xxy, xyx, xyy, yxx, yxy, yyx, yyy

start → $(x)(x+y)(y)$ → xxy, xyy

start → (x)(x+y)*(y) → xλy = xy

λ x xx xxx xxxx xxxxx xxxxxx . . .
λ y yy yyy yyyy yyyyy yyyyyy . . .
λ xy xyy xyyy xyyyy xyyyyy xyyyyyy . . .
λ yx yxx yxxx yxxxx yxxxxx yxxxxxxx . . .
And all possible combinations. Can be Inserted in between x and y.

PRACTICE

(0*10*)

(0 + 10*)

(0 + ε)(1 + ε)

(a+b)*

(0*1110*)

(0*10*)+(0 + 10*)

(0*10*) + (11) + (1 + 0)*

READING ASSIGNMENT

- Read what you have been taught today.
- Download book and read Chapter 4 from Daniel I A Cohen book 2nd Edition.
- Solve all the above cases discussed in class by yourself.
- Try to write program for the regular expressions learn today. Do not use any library that supports regex.

END OF LECTURE