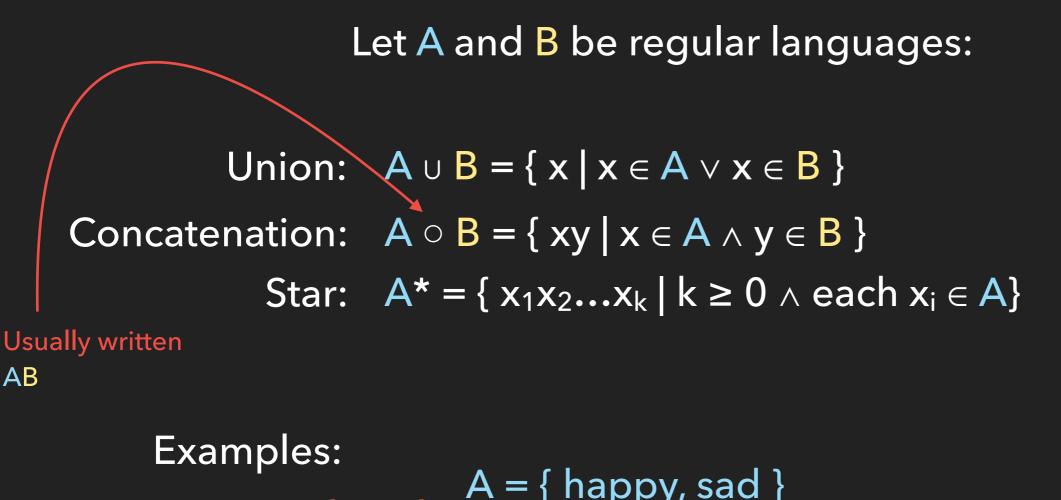
REGULAR EXPRESSIONS

CSC 240

- A language is called a regular language if some finite automaton recognizes it.
- Every nondeterministic finite automaton has an equivalent deterministic finite automaton.
- A language is regular if and only if some nondeterministic finite automaton recognizes it.

OPERATIONS ON REGULAR LANGUAGES (THE REGULAR OPERATIONS)



 $\Sigma = \{a...z\} A = \{happy, sad \}$ $B = \{cat, dog \}$ $A \cup B = \{happy, sad, cat, dog \}$ $A \cup B = \{happy, sad, cat, dog \}$ $A \circ B = \{happycat, sadcat, happydog, saddog\}$ $A^* \{\varepsilon, happy, sad, happyhappy, sadsad, happysad, sadhappy, sadsadhappy, sadhappy, sadhappysad, ... \}$

 $\Sigma = \{a...z, A...Z\}$

Article = { A, The }

```
Noun = { Cat, Dog, Moose, Cow }
```

```
Verb = { Eats, Chases, Kicks, Loves }
```

```
ArticleNounVerbArticleNoun = {
ACatEatsTheDog, TheMooseChasesACow,
ACowLovesACat, ... }
```

If Article, Noun, and Verb are regular languages, is ArticleNountVerbArticleNoun a regular language?

 $\Sigma = \{a...z\}$

 $A = \{a, b\}$

 $AA = \{aa, ab, ba, bb\}$

AAA = { aaa, aab, aba, baa, bba, bab, abb, bbb }

 $\begin{array}{l} A^{0} = \{\epsilon\} \\ A^{1} = A \\ A^{2} = AA \\ A^{2} = AA \\ A^{*} = \{ x_{1}x_{2}...x_{k} \mid k \geq 0 \land each \; x_{i} \in A \} \\ A^{+} = \{ x_{1}x_{2}...x_{k} \mid k \geq 1 \land each \; x_{i} \in A \} \\ A^{+} = \{ x_{1}x_{2}...x_{k} \mid k \geq 1 \land each \; x_{i} \in A \} \\ \end{array}$

 $\mathbb{N} = \{1, 2, 3, ...\}$

 \mathbb{N} is *closed* under multiplication:

$$\forall x. (x \in \mathbb{N} \rightarrow \forall y. (y \in \mathbb{N} \rightarrow x \cdot y \in \mathbb{N}))$$

For every x and y in the set of natural numbers, the product of x and y is a natural number.

Regular Languages are closed under the Union operation. Regular Languages are closed under the Concatenation operation. Regular Languages are closed under the Star operation.

- If we can create a DFA that recognizes L.
- If we can create an NFA that recognizes L.
- If L can be formed from other regular languages using regular operations.

Regular Expressions are expressions built from regular operations and are used to describe a language.

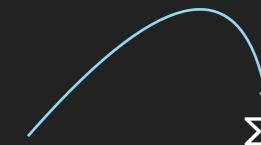
R is a regular expression if R is:

a for some a in Σ ϵ

Ø

 $(R_1 \cup R_2)$ Where R_1 and R_2 are regular expressions $(R_1 \circ R_2)$ Where R_1 and R_2 are regular expressions (R_1^*) Where R_1 is a regular expression

$\Sigma = \{0, \, 1\}$



0*10* = {w | w contains a single 1}

 $\Sigma^*1\Sigma^* = \{w \mid w \text{ contains at least a single 1}\}$

Means "star operator on the alphabet"

 $\Sigma^* 001\Sigma^* = \{w \mid w \text{ contains the substring } 001\}$

1*(01+)* = {w | every 0 in w is followed by at least a single 1}

 $(\Sigma\Sigma)^* = \{w \mid w \text{ is a string of even length}\}\$

 $01 \cup 10 = \{01, 10\}$

 $0\Sigma^*0 \cup 1\Sigma^*1 \cup 0 \cup 1 = \{w \mid w \text{ starts and ends with the same symbol}\}$ $(0 \cup \varepsilon)1^* = 01^* \cup 1^*$

REGULAR EXPRESSION IDENTITIES

$\mathsf{R} \cup \varnothing = \mathsf{R}$

Adding the empty language to another language doesn't change it.

 $\mathsf{R} \circ \varepsilon = \mathsf{R}$

Concatenating the empty string to another string doesn't change it.

However:

 $R \cup \varepsilon \neq R$ If R = 0, then L (R) = {0}, but L (R \cup \varepsilon) = {0, \varepsilon} }
R \cip \varnothing \nothing R
If R = 0, then L (R) = {0}, but L (R \cup \varnothing) = \varnothing

 $D = \{0...9\}$ $\Sigma = \{D, +, -, .\}$ $(+ \cup - \cup \varepsilon) (D^+ \cup D^+.D^* \cup D^*.D^*)$

Positive or negative numbers with or without decimals.

 $W = \{a..., A..., Z\}$ $\Sigma = \{W, .., @\}$ $W^{+}(.W^{+})^{*}@W^{+}(.W^{+})^{+}$

Valid email addresses