PART 1:

Automata:

Finite State Machines (Finite Automata)

Formal Language:

Regular Languages

Non-regular Languages

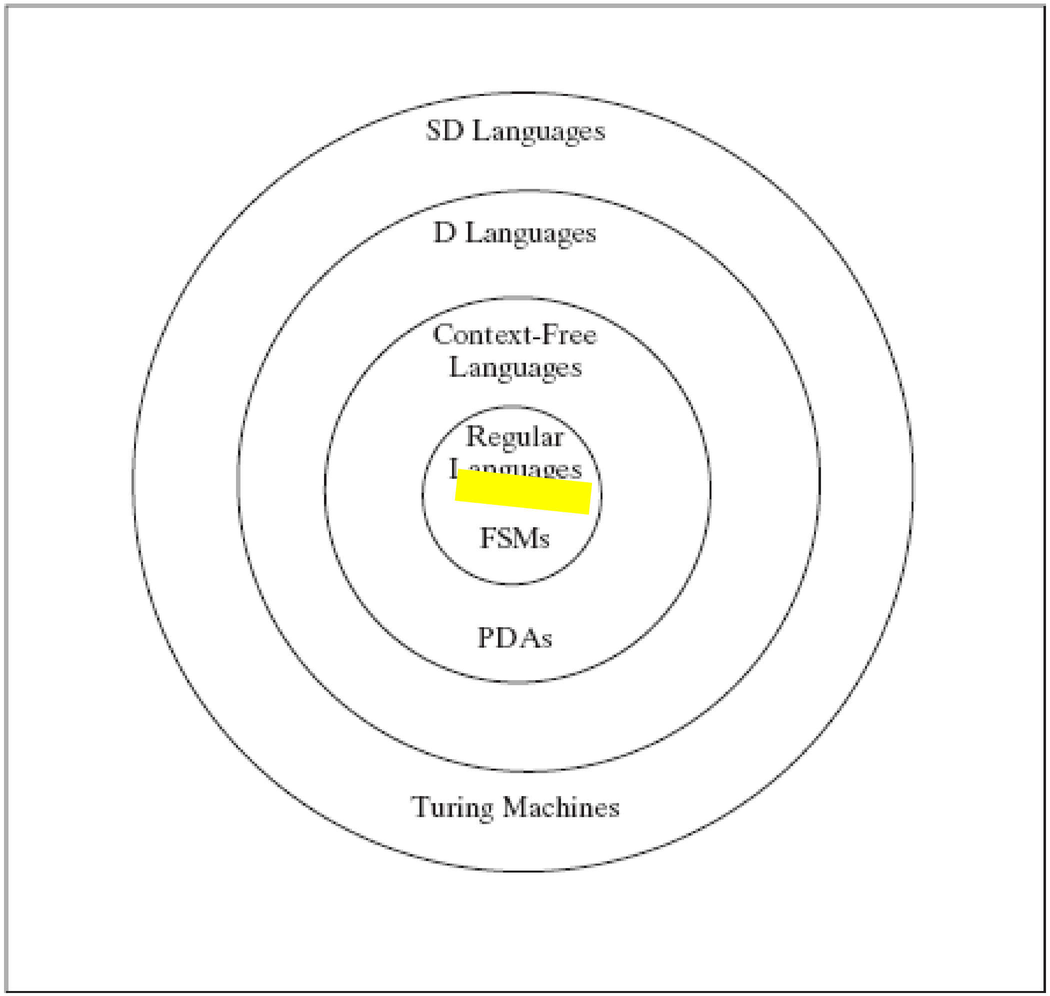
Grammar:

Regular Expressions

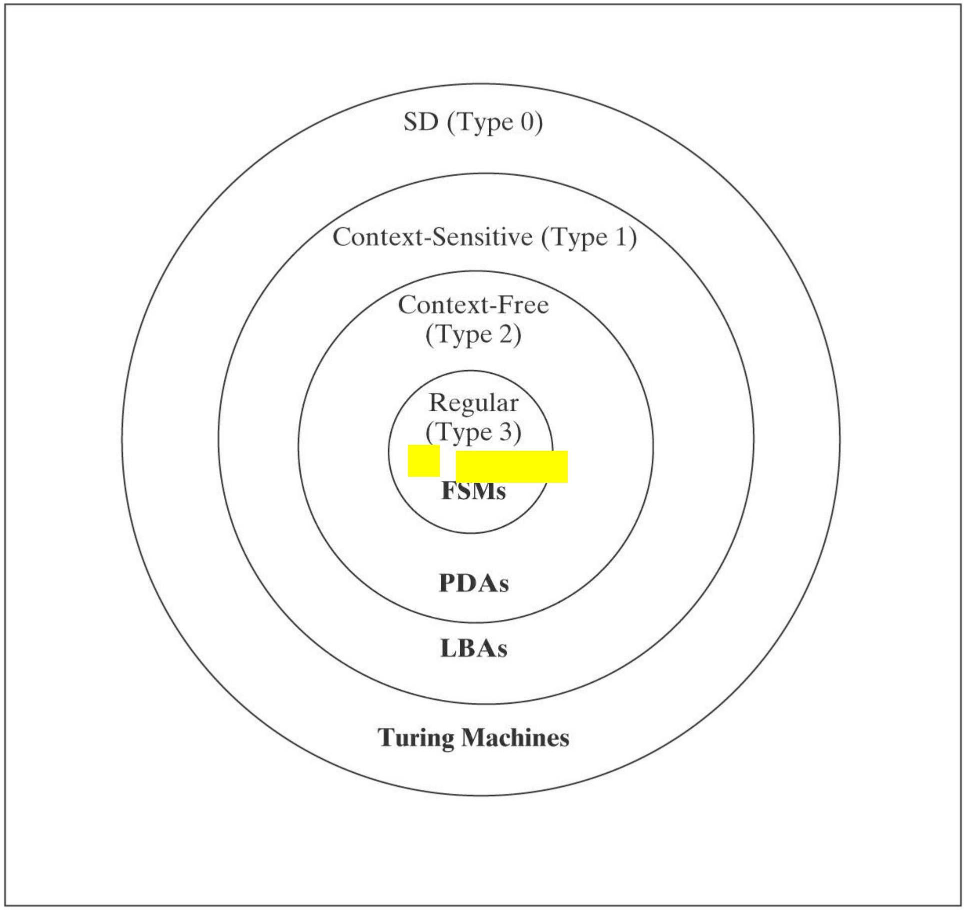
Regular Grammars

Regular Expressions

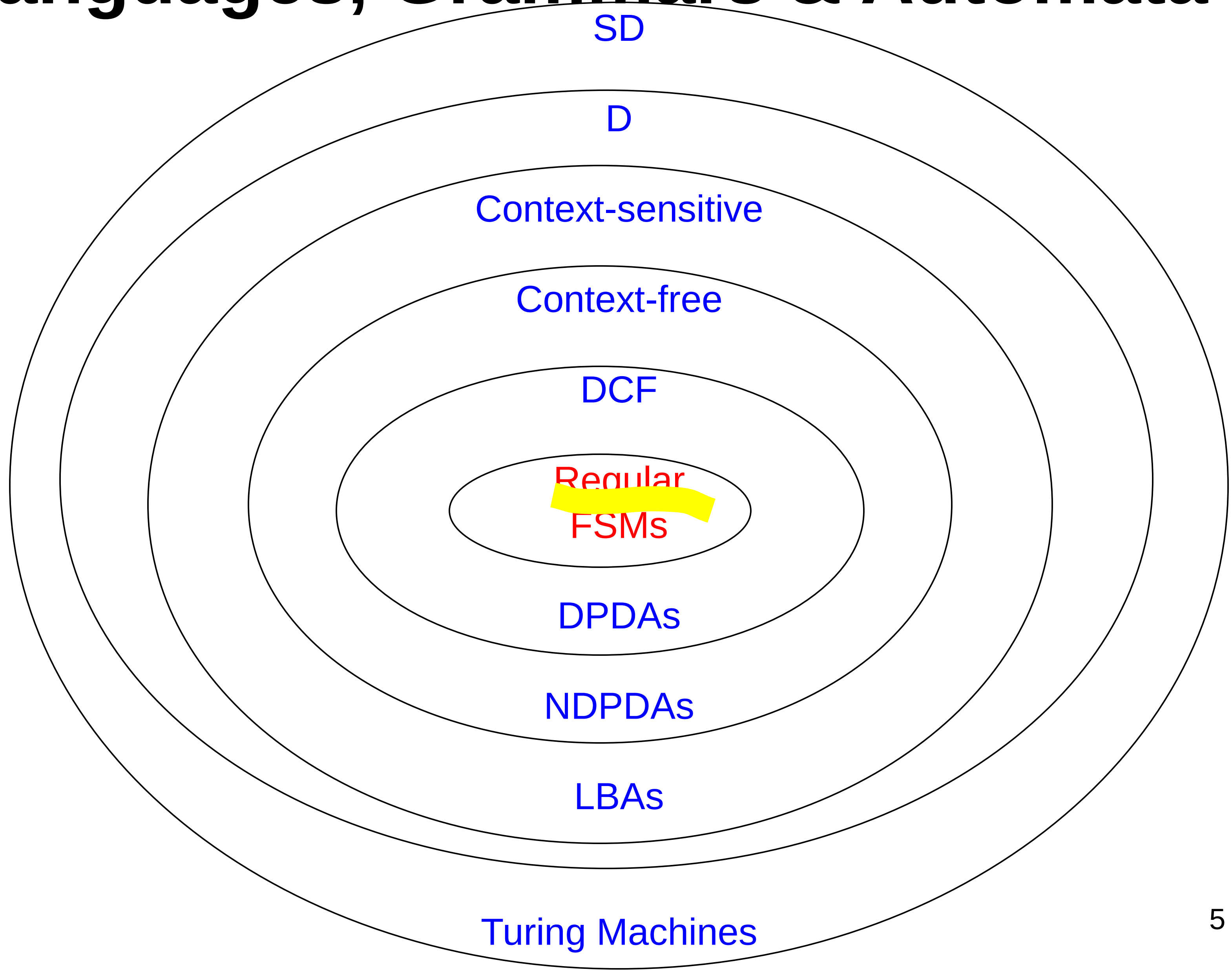
Languages, Grammars & Automata



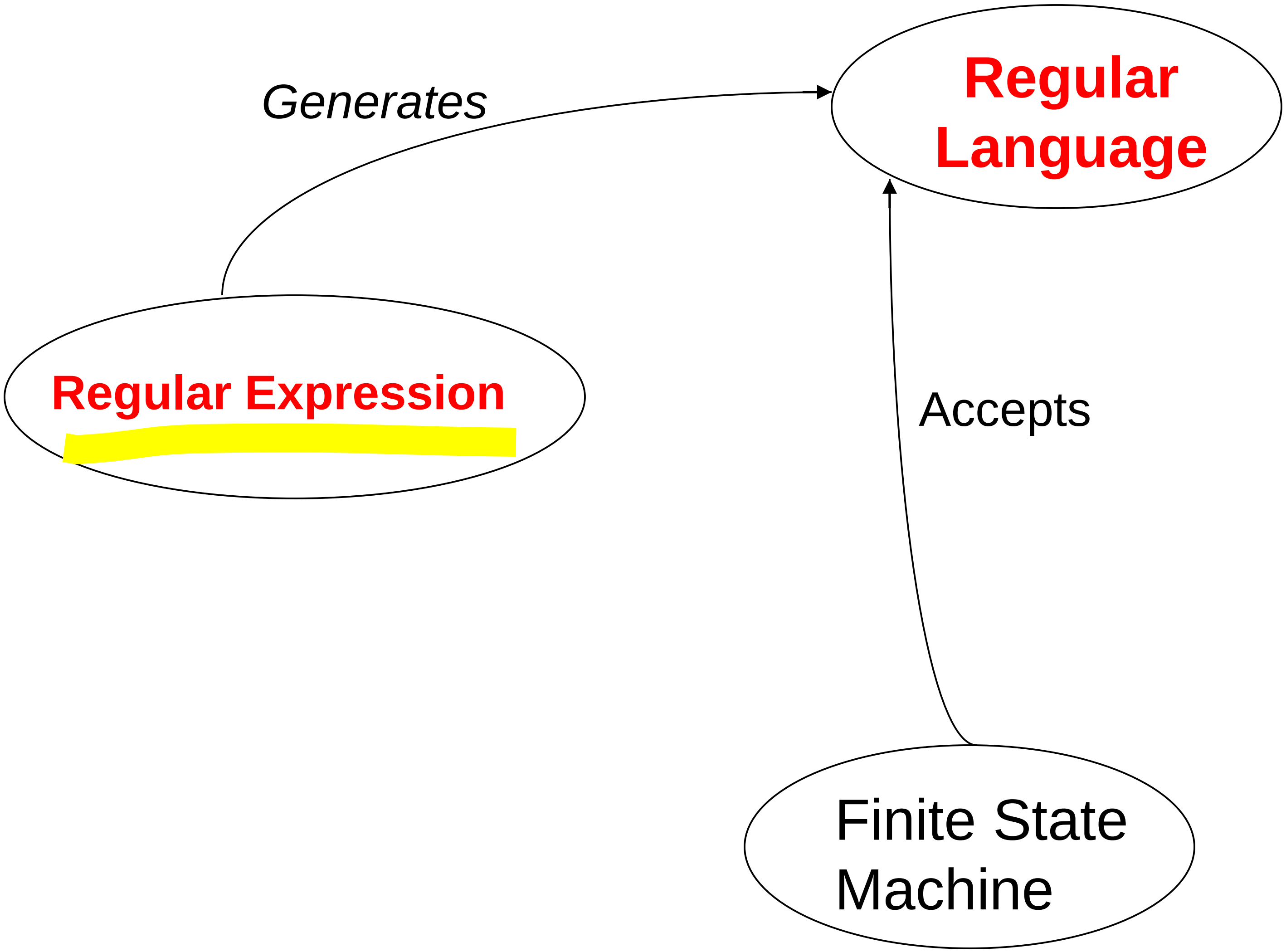
Languages, Grammars & Automata



Languages, Grammars & Automata



Regular Languages



Expressions

- Arithmetic Expressions?
- Regular expressions?

Regular Expressions ?

- An **algebraic expression notation** to describe **regular languages**!

Definition of Regular Expressions

The regular expressions over an alphabet Σ are all and only the strings that can be obtained as follows:

1. $\emptyset$ is a regular expression.

2. ε is a regular expression.

3. Every element of Σ is a regular expression.

Definition of Regular Expressions

4. If α , β are regular expressions, then so is $\alpha\beta$.

5. If α , β are regular expressions, then so is $\alpha\cup\beta$.

6. If α is a regular expression, then so is α^*.

7. α is a regular expression, then so is α^+.

8. If α is a regular expression, then so is (α).

The Role of the Rules

- Rules 1, 3, 4, 5, and 6 give the language its power to define sets.
- Rule 8 has as its only role grouping other operators.
- Rules 2 and 7 appear to add functionality to the regular expression language, but they don't.
  - ✓ $\varepsilon = \emptyset^*$
  - ✓ $\alpha^+ = \alpha\alpha^*$

Example: RE

If $\Sigma = \{\mathtt{a}, \mathtt{b}\}$, the following are regular expressions:

$\emptyset$

ε

$\mathtt{a}$

$(\mathtt{a} \cup \mathtt{b})^*$

$\mathtt{abba} \cup \varepsilon$

$\emptyset^*$

Regular Expressions Define Languages !

Define *L*, a semantic interpretation function for regular expressions:

1. $L(\emptyset) = \emptyset$

2. $L(\varepsilon) = \{\varepsilon\}$

3. $L(c)$, where $c \in \Sigma = \{c\}$

Regular Expressions Define Languages

4. $L(\alpha\beta) = L(\alpha)\ L(\beta)$

5. $L(\alpha \cup \beta) = L(\alpha) \cup L(\beta)$

6. $L(\alpha^*) = (L(\alpha))^*$

7. $L(\alpha^+) = L(\alpha\alpha^*) = L(\alpha)\ (L(\alpha))^*$

8. $L((\alpha)) = L(\alpha)$

Examples: Regular Expressions

$\Sigma = \{a, b\}$:

$L(\emptyset) = \{\ \}$

$L(\varepsilon) = \{\varepsilon\}$

$L(a) = \{a\}$

$L((a \cup b)^*) =$

$L(abba \cup \varepsilon) = \{abba, \varepsilon\}$

$L(\emptyset^*) = \{\varepsilon\}$

Example 6.1

$$
\begin{aligned}
L((\mathtt{a} \cup \mathtt{b})^*\mathtt{b}) &= L((\mathtt{a} \cup \mathtt{b})^*)\ \ L(\mathtt{b}) \\
&= (L((\mathtt{a} \cup \mathtt{b})))^*\ L(\mathtt{b}) \\
&= (L(\mathtt{a}) \cup L(\mathtt{b}))^*\ L(\mathtt{b}) \\
&= (\{\mathtt{a}\} \cup \{\mathtt{b}\})^*\ \{\mathtt{b}\} \\
&= \{\mathtt{a}, \mathtt{b}\}^*\ \{\mathtt{b}\}.
\end{aligned}
$$

L= ?

The set of all strings that end in $\mathtt{b}$.

Example 6.2

$L(((a \cup b) (a \cup b)) \text{ a } (a \cup b) ^*)$

=

$= \{a, b\} \{a, b\} \{a\}\{a, b\}^*$

L= ?

{xay: x and y are strings of a’s and b’s and |x|=2}

The set of all strings st there exists a third character a.

Example 6.3

$L = \{w \in \{\mathtt{a}, \mathtt{b}\}^*: |w| \text{ is even}\}$

RE?

$((\mathtt{a} \cup \mathtt{b})(\mathtt{a} \cup \mathtt{b}))^*$

$(\mathtt{aa} \cup \mathtt{ab} \cup \mathtt{ba} \cup \mathtt{bb})^*$

Example 6.4

L = {w ∈ {a, b}*: w contains an odd number of a's}

RE?

b* (ab*ab*)* a b*

b* a b* (ab*ab*)*

More Examples: RE

$(\alpha \cup \varepsilon)$

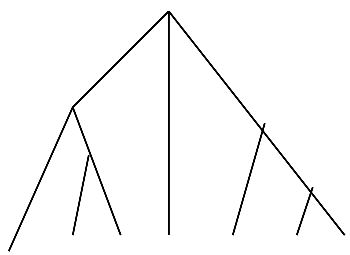
$(a \cup b)^*$

$(a^* \cup b^*)$? $(a \cup b)^*$

$(ab)^*$? $a^* b^*$

Operator Precedence in RE

	Regular Expressions	Arithmetic Expressions
Highest	Kleene star	exponentiation
↕	concatenation	multiplication
Lowest	union	addition



$a\ b^* \cup c\ d^*$ $\qquad$ $x\ y^2 + i\ j^2$

Equivalence of RE and FSM

Finite state machines (FSM) and regular expressions (RE) define the same class of languages!

RE = FSM (DFA & NFA)

RE = RL

Building an FSM from a RE

For every RE, there is an Equivalent FSM.

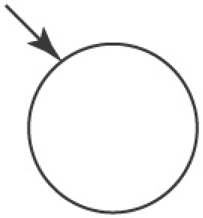
Theorem 6.1 Any language that can be defined with a regular expression can be accepted by some FSM and so is regular.

Proof Idea:
Proof by Construction

For Every Regular Expression There is a Corresponding FSM

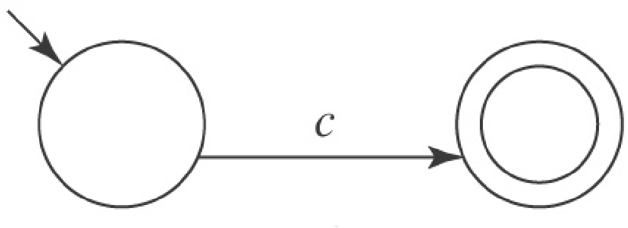
We'll show this by construction. An FSM for:

$\varnothing$:



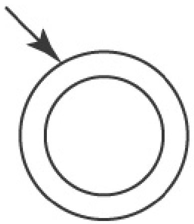
For Every Regular Expression There is a Corresponding FSM

A single element of Σ:



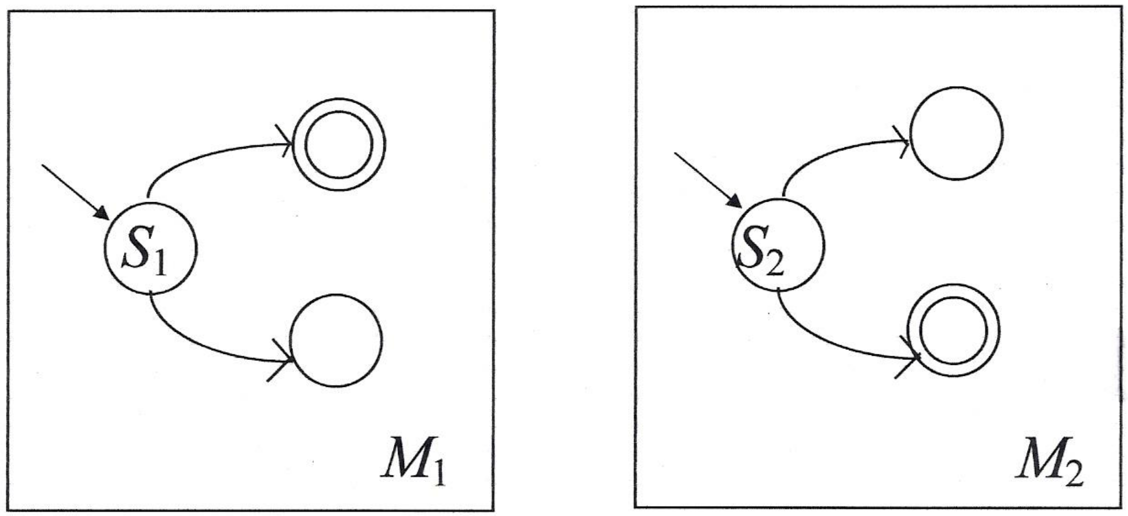
For Every Regular Expression There is a Corresponding FSM

ε (= $\emptyset$*):



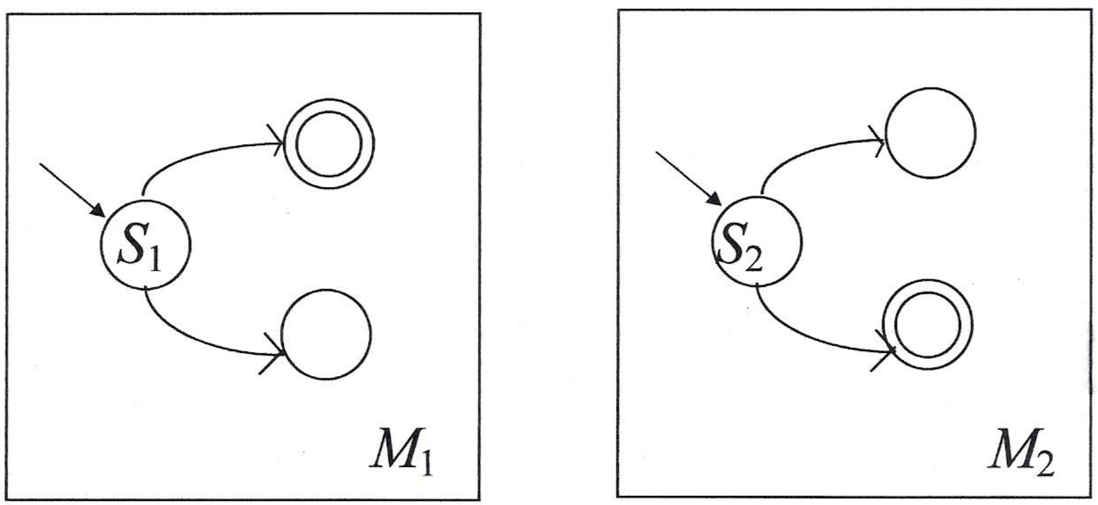
Union

If α is the regular expression $\beta \cup \gamma$ and if both $L(\beta)$ and $L(\gamma)$ are regular:



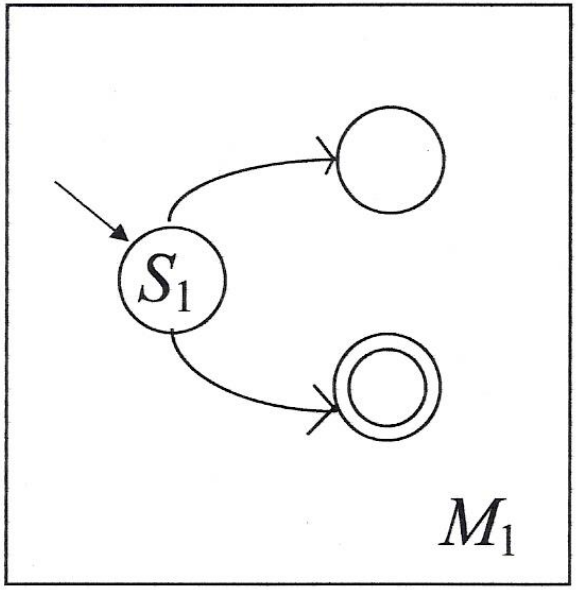
Concatenation

If α is the regular expression $\beta\gamma$ and if both $L(\beta)$ and $L(\gamma)$ are regular:



Kleene Star

If α is the regular expression β^* and if $L(\beta)$ is regular:

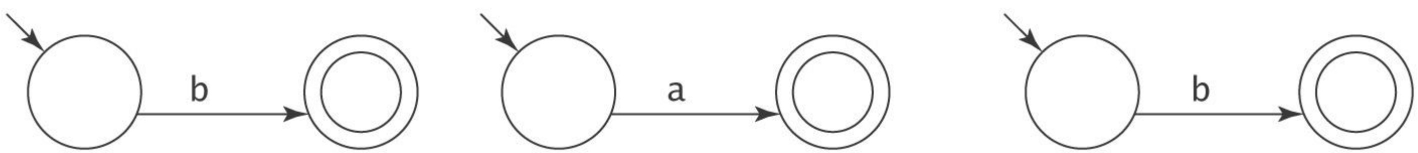


Example 6.5

RE: (b ∪ ab)*

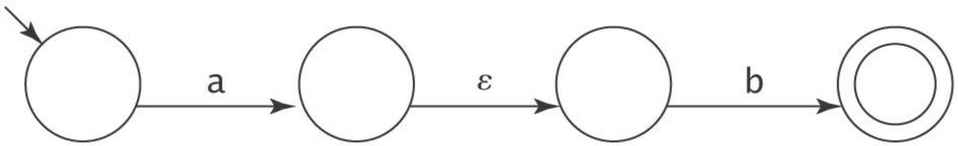
FSM?

An FSM for b An FSM for a An FSM for b



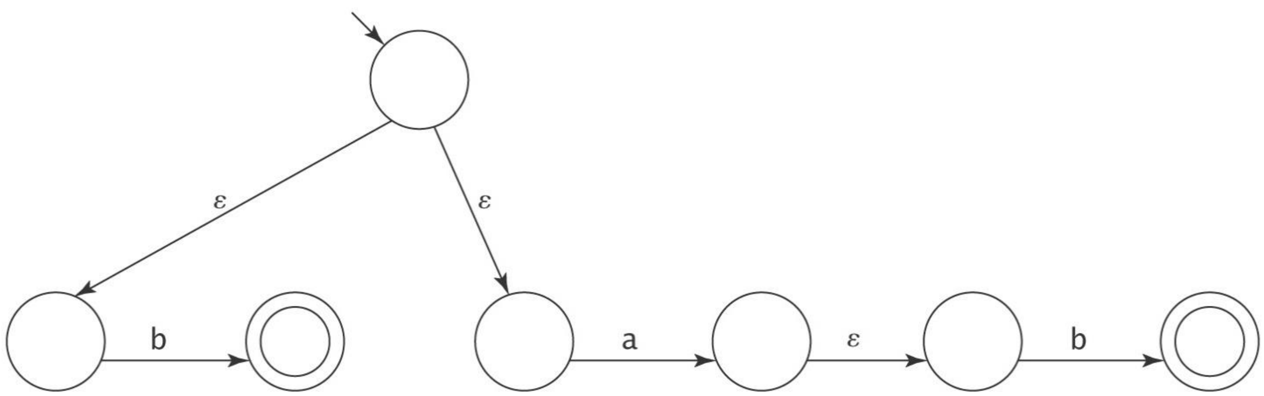
Example 6.5

An FSM for `ab`:



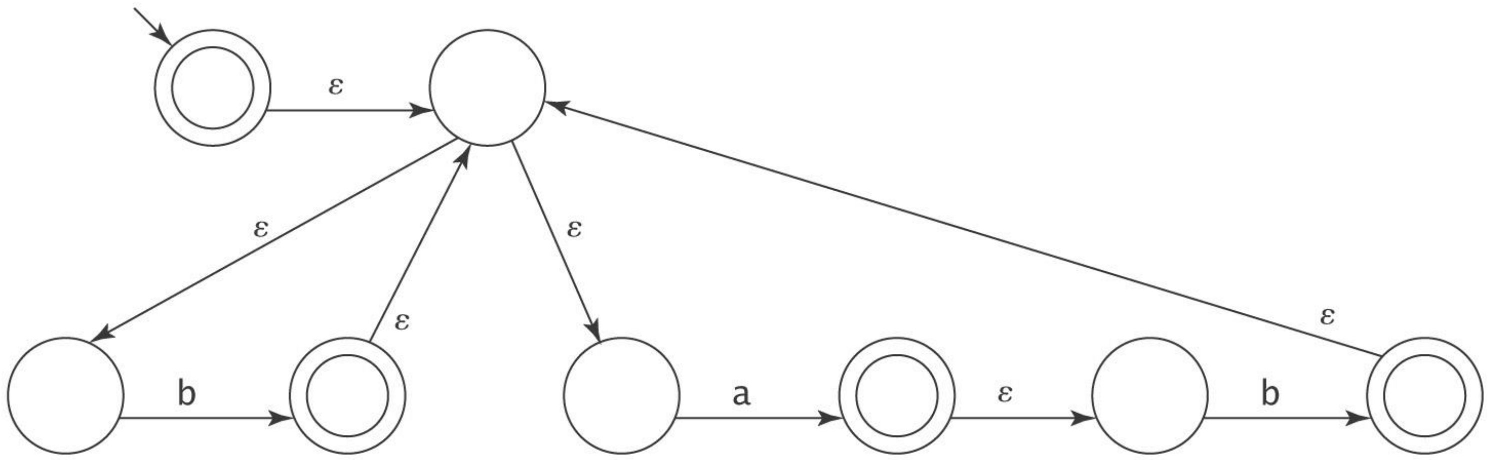
Example 6.5

An FSM for ($\mathtt{b} \cup \mathtt{ab}$):



Example 6.5

An FSM for ($\mathtt{b} \cup \mathtt{ab}$)*:



Building a RE from an FSM

For every FSM, there is an equivalent RE.

Theorem 6.2 Every regular language (i.e., every language that can be accepted by some DFSM) can be defined with a regular expression.

Equivalence of Regular Languages and Regular Expressions

Kleene Theorem

Theorem 6.3 The class of languages that can be defined with regular expressions is exactly the class of regular languages.

RL = RE = FSM

Examples and Designing REs

Example 6.10

$L = \{ w \in \{a, b\}^*$: there is no more than one b}.

RE?

$a^*(b \cup \varepsilon)a^*$

Example 6.11

$L = \{ w \in \{a, b\}^*$: no two consecutive letters are the same$\}$.

RE?

$(b \cup \varepsilon)(ab)^*(a \cup \varepsilon)$

$(a \cup \varepsilon)(ba)^*(b \cup \varepsilon)$

Example 6.12

L = FLOAT = {*w*: w is the string representation of a floating point number}.

RE?

Example 6.14

L = Decimal numbers

RE?

Example 6.15

L = Legal passwords

RE?

Example 6.16

L = IP addresses

RE?

Simplifying Regular Expressions

- Union is commutative: $\alpha \cup \beta = \beta \cup \alpha$
- Union is associative: $(\alpha \cup \beta) \cup \gamma = \alpha \cup (\beta \cup \gamma)$
- $\varnothing$ is the identity for union: $\alpha \cup \varnothing = \varnothing \cup \alpha = \alpha$
- Union is idempotent: $\alpha \cup \alpha = \alpha$

Simplifying Regular Expressions

- Concatenation is associative: $(\alpha\beta)\gamma = \alpha(\beta\gamma)$
- ε is the identity for concatenation: $\alpha\, \varepsilon = \varepsilon\, \alpha = \alpha$
- $\varnothing$ is a zero for concatenation: $\alpha\, \varnothing = \varnothing\, \alpha = \varnothing$
- $(\alpha \cup \beta)\, \gamma = (\alpha\, \gamma) \cup (\beta\, \gamma)$
- $\gamma\, (\alpha \cup \beta) = (\gamma\, \alpha) \cup (\gamma\, \beta)$

Simplifying Regular Expressions

- $\emptyset^* = \varepsilon$
- $\varepsilon^* = \varepsilon$
- $(\alpha^*)^* = \alpha^*$
- $\alpha^*\alpha^* = \alpha^*$
- $(\alpha \cup \beta)^* = (\alpha^*\beta^*)^*$

Example 6.17

Simplifying a RE

Reading Assignment

Chapter 6:

Sections
6.1
6.2
6.3
6.4

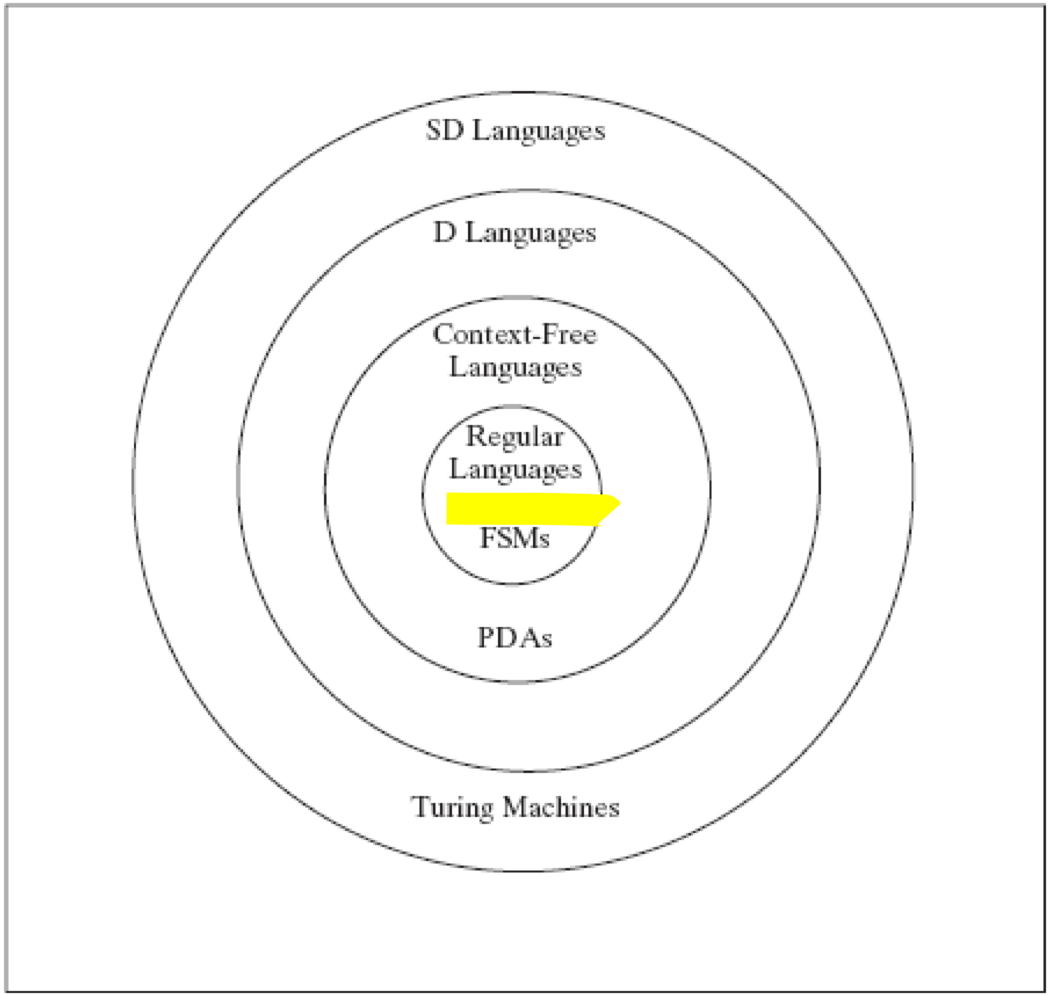
In-Class Exercises

Chapter 6:

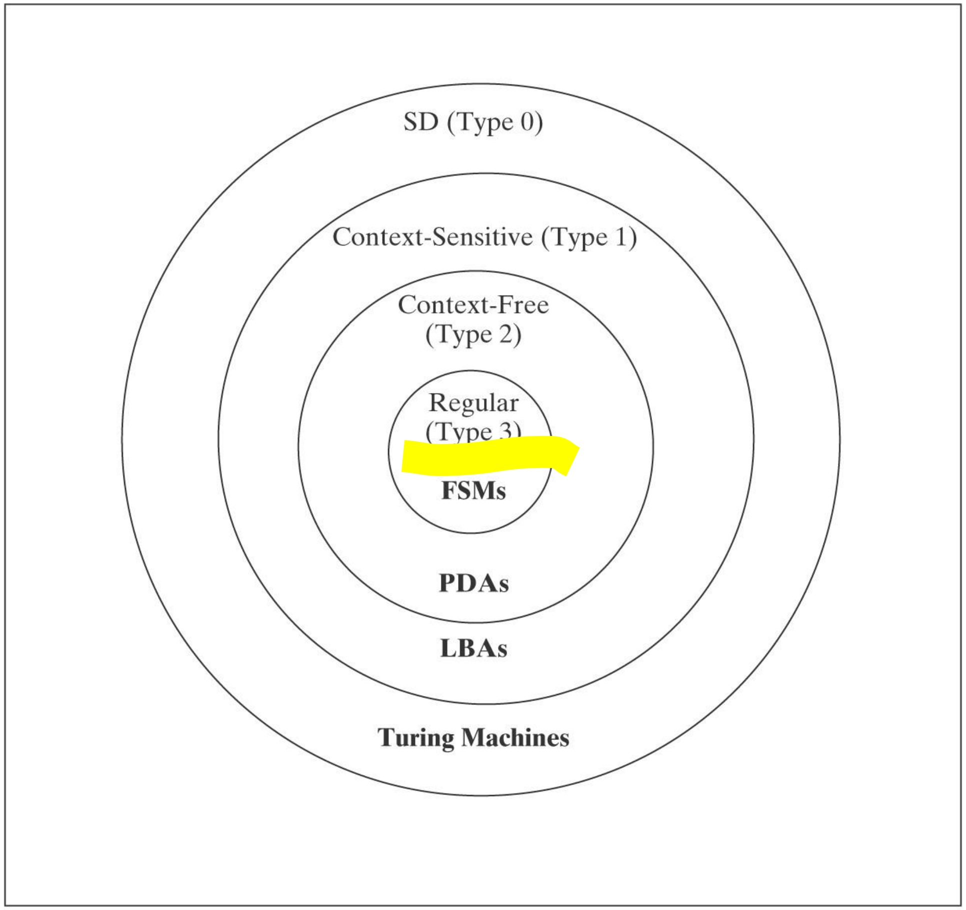
1
2 - g
4
5
8
13 - e

Regular Grammars (Right Linear Grammars)

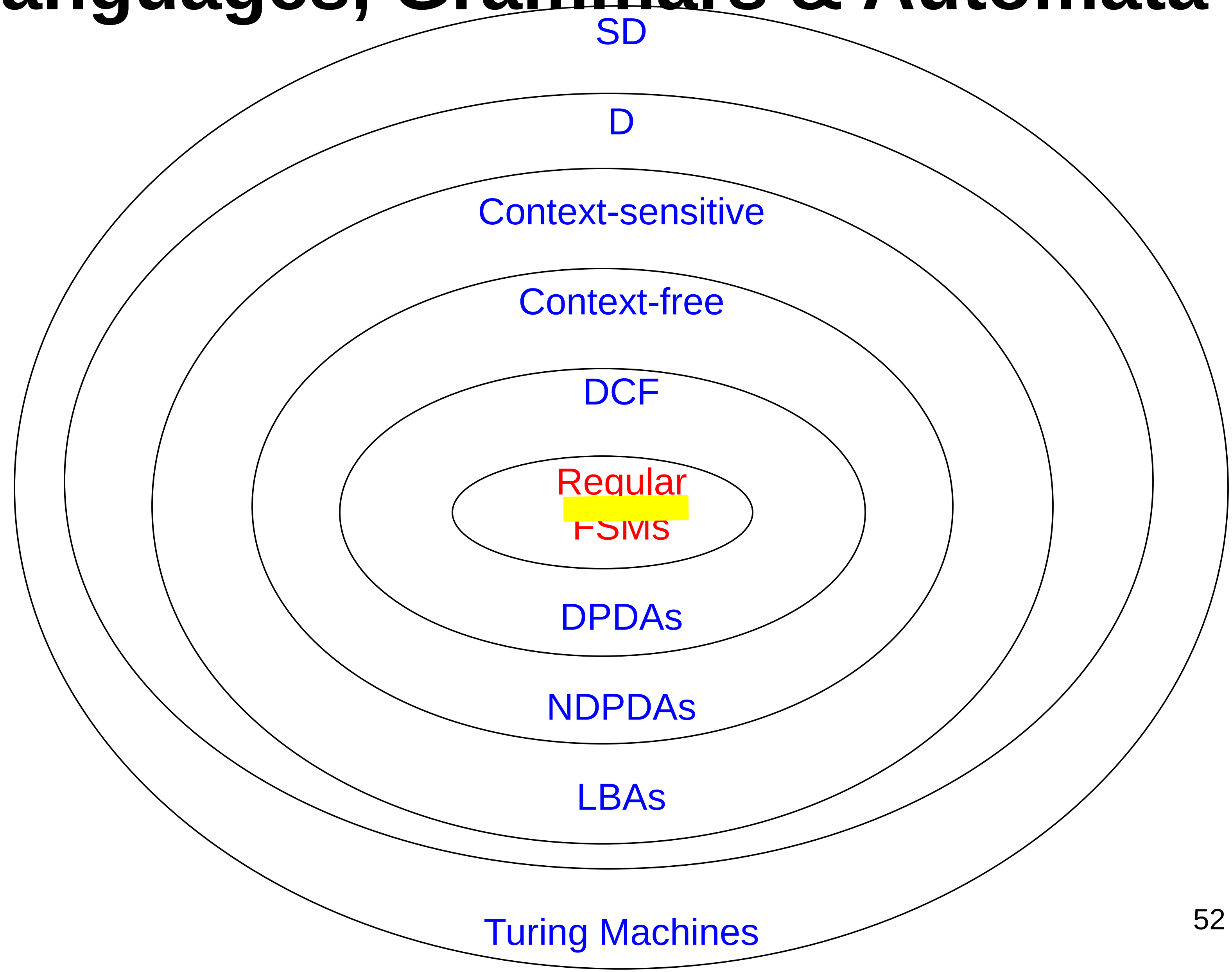
Languages, Grammars & Automata



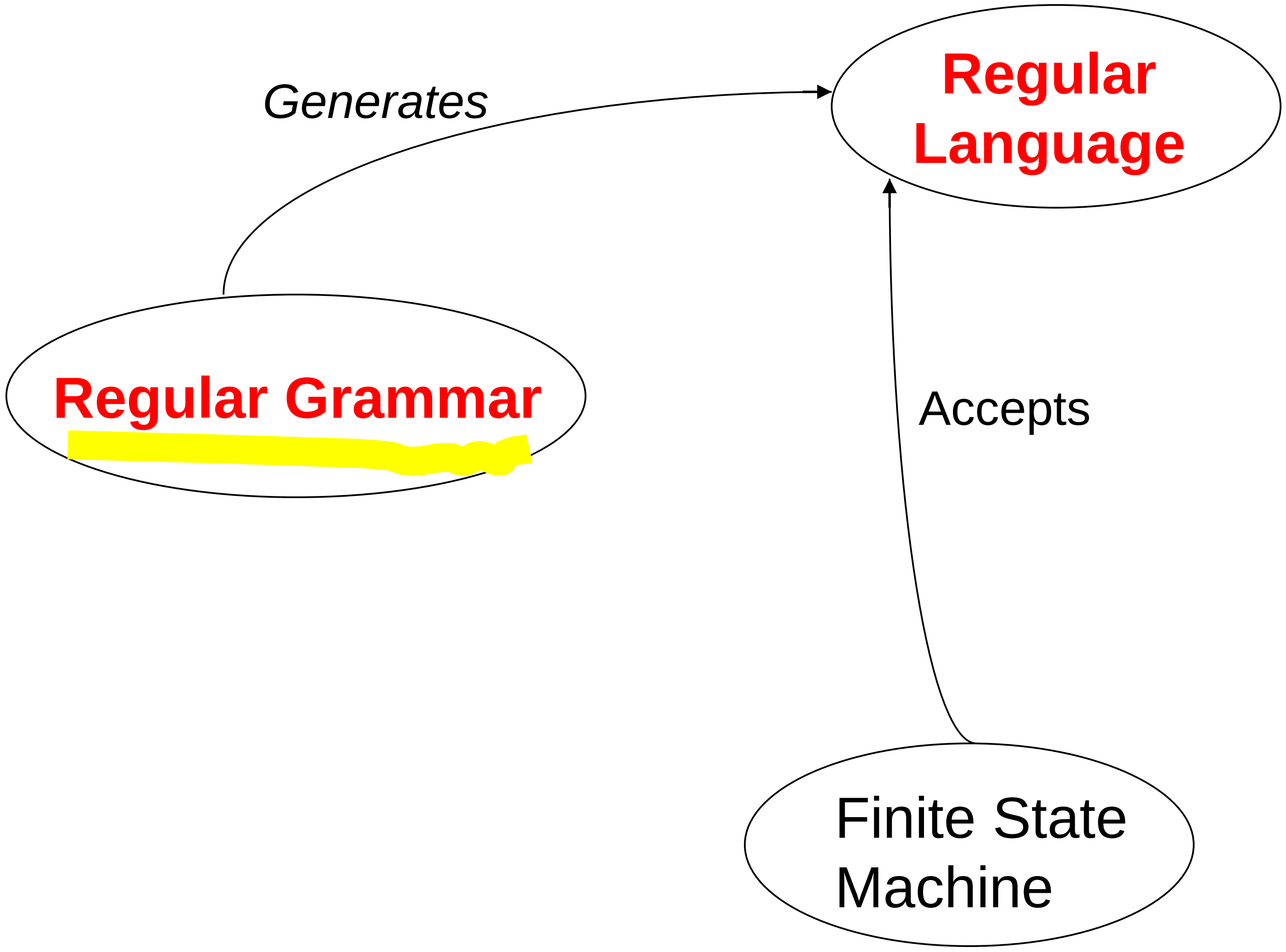
Languages, Grammars & Automata



Languages, Grammars & Automata



Regular Languages



Definition of Regular Grammars

A **regular grammar** or **right-linear grammar** G is a quadruple (V, Σ, R, S), where:

- V is the rule alphabet, which contains nonterminals and terminals,
- Σ (the set of terminals) is a subset of V,
- R (the set of rules) is a finite set of rules of the form $X \rightarrow Y$,
- S (the start symbol) is a nonterminal.

Regular Grammars or Right-Linear Grammars

In a regular grammar, all rules in *R* must:

- have a left hand side that is a single nonterminal
- have a right hand side that is:
  - ε, or
  - a single terminal, or
  - a single terminal followed by a single nonterminal.

Legal: $S \rightarrow \mathtt{a}$, $S \rightarrow \varepsilon$, and $T \rightarrow \mathtt{a}S$
Not legal: $S \rightarrow \mathtt{a}S\mathtt{a}$ and $\mathtt{a}S\mathtt{a} \rightarrow T$

Example: RG

G = {{S,T}, {$\mathtt{a}$, $\mathtt{b}$}, R, S}, where:

R = {

$S \rightarrow \varepsilon$
$S \rightarrow \mathtt{a}T$
$S \rightarrow \mathtt{b}T$
$T \rightarrow \mathtt{a}$
$T \rightarrow \mathtt{b}$
$T \rightarrow \mathtt{a}S$
$T \rightarrow \mathtt{b}S$

}

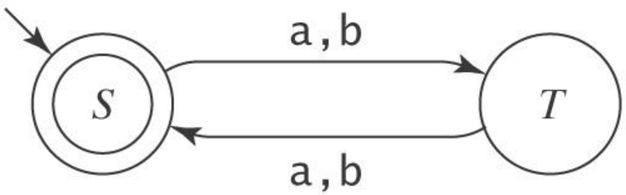
Example 7.1

L = {$w \in$ {a, b}* : $|w|$ is even}

RE?

((aa) ∪ (ab) ∪ (ba) ∪ (bb))*

FSM?



RG?

$S \rightarrow \varepsilon$
$S \rightarrow$ aT
$S \rightarrow$ bT
$T \rightarrow$ aS
$T \rightarrow$ bS

Equivalence of Regular Languages and Regular Grammars

Theorem 7.1 The class of languages that can be defined/generated with regular grammars is exactly the regular languages.

Proof Idea:

Proof by Construction

RL = RG

Regular Languages and Regular Grammars

Regular grammar* $\rightarrow$ *FSM:

grammartofsm(*G* = (*V*, Σ, *R*, S)) =

1. Create in *M* a separate state for each nonterminal in *V*.
2. Start state is the state corresponding to *S* .
3. If there are any rules in *R* of the form $X \rightarrow w$, for some $w \in \Sigma$, create a new state labeled #.
4. For each rule of the form $X \rightarrow w\ Y$, add a transition from *X* to *Y* labeled *w*.
5. For each rule of the form $X \rightarrow w$, add a transition from *X* to # labeled *w*.
6. For each rule of the form $X \rightarrow \varepsilon$, mark state *X* as accepting.
7. Mark state # as accepting.

FSM* $\rightarrow$ *Regular grammar: Similarly.

Example 7.2

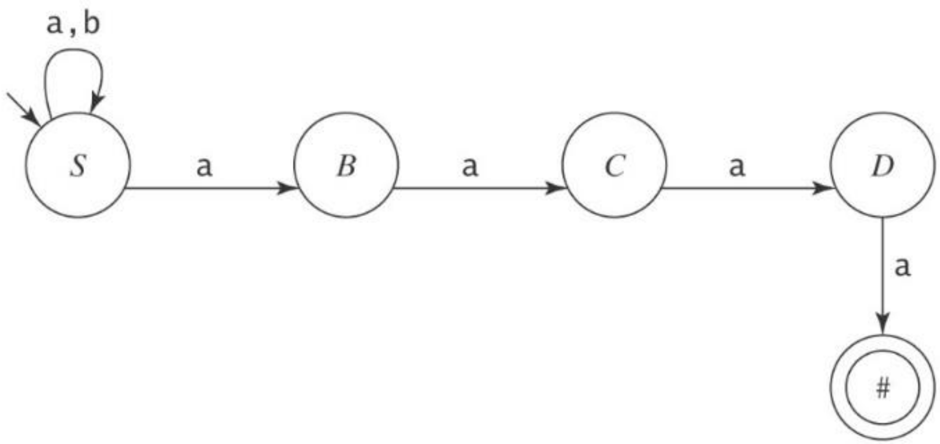
L = {$w \in$ {a, b}* : w ends with the pattern aaaa}.

RE?

(a $\cup$ b)*aaaa

RG: FSM?

$S \rightarrow \mathrm{a}S$
$S \rightarrow \mathrm{b}S$
$S \rightarrow \mathrm{a}B$
$B \rightarrow \mathrm{a}C$
$C \rightarrow \mathrm{a}D$
$D \rightarrow \mathrm{a}$

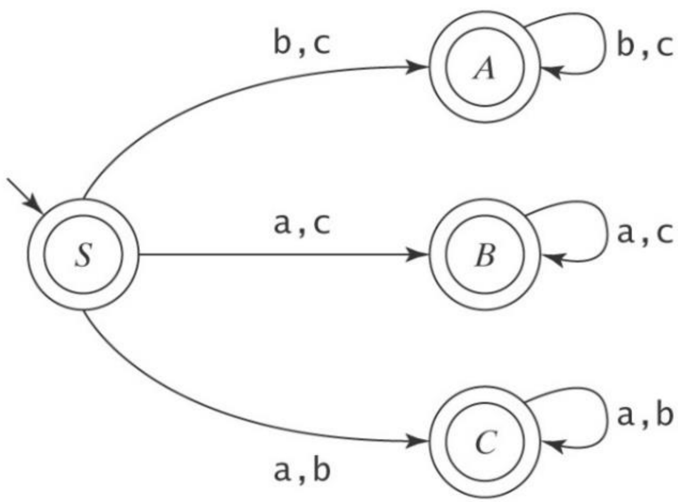


Example 7.3

RG:

$S \rightarrow \varepsilon$	$A \rightarrow \mathtt{b}A$	$C \rightarrow \mathtt{a}C$
$S \rightarrow \mathtt{a}B$	$A \rightarrow \mathtt{c}A$	$C \rightarrow \mathtt{b}C$
$S \rightarrow \mathtt{a}C$	$A \rightarrow \varepsilon$	$C \rightarrow \varepsilon$
$S \rightarrow \mathtt{b}A$	$B \rightarrow \mathtt{a}B$	
$S \rightarrow \mathtt{b}C$	$B \rightarrow \mathtt{c}B$	
$S \rightarrow \mathtt{c}A$	$B \rightarrow \varepsilon$	
$S \rightarrow \mathtt{c}B$		

FSM?



Reading Assignment

Chapter 7:

Sections
7.1
7.2

In-Class Exercises

Chapter 7:

1 – c & e

2 - a

5