Monash University
Faculty of Information Technology

# Lecture 9
# (A) Closure properties.
# (B) Pumping Lemma for Regular Languages.

Slides by David Albrecht (2011), revisions and additions by Graham Farr (2013, 2018).

FIT2014 Theory of Computation

# Overview

- Closure properties of regular languages
- Circuits in FAs
- Pumping Lemma
- Non-regular Languages

# Closure properties of regular languages

If doing some operation on regular languages *always* produces another regular language, then we say that regular languages are ***closed*** under that operation.

We will see that regular languages are closed under:

complement, union, intersection, concatenation

**Theorem**

The complement of a regular language is regular.

We prove this using Kleene's Theorem.

# Closure properties of regular languages

**Theorem**

The complement of a regular language is regular.

***Proof*** (outline):

Suppose we have a Regular Language.

There must be a regular expression that defines it.

So, by Kleene's Theorem, there is a Finite Automaton (FA) that defines this language.

We can convert this FA into one that defines the complement of the language.

So, by Kleene's Theorem, there is a regular expression that defines the complement. Q.E.D.

# Closure properties of regular languages

**Theorem**

The union of two regular languages is regular.

***Proof***

- Suppose $\mathbf{L_1}$ and $\mathbf{L_2}$ are regular.
- By definition of "regular language", there exist regular expressions $\mathbf{R_1}$ and $\mathbf{R_2}$ that describe $\mathbf{L_1}$ and $\mathbf{L_2}$, respectively.
- Then $\mathbf{R_1} \cup \mathbf{R_2}$ is a regular expression that describes $\mathbf{L_1} \cup \mathbf{L_2}$.

This uses part 3(iii) of the inductive definition of regular expressions in Lecture 5.

- So $\mathbf{L_1} \cup \mathbf{L_2}$ is regular. Q.E.D.

# Closure properties of regular languages

**Theorem**

The intersection of two regular languages is regular.

We can't just mimic the proof that regular languages are closed under union, since there is no $\cap$ operation on regular expressions.

# Closure properties of regular languages

**Theorem**

The intersection of two regular languages is regular.

***Proof***

- Suppose $L_1$ and $L_2$ are regular.
- We know that their complements $\overline{L_1}$ and $\overline{L_2}$ are regular, and the union of these, $\overline{L_1} \cup \overline{L_2}$, is therefore regular, by the previous Theorem.
- Its complement, $\overline{\overline{L_1} \cup \overline{L_2}} = L_1 \cap L_2$, must also be regular.

Q.E.D.

# Closure properties of regular languages

**Exercises**

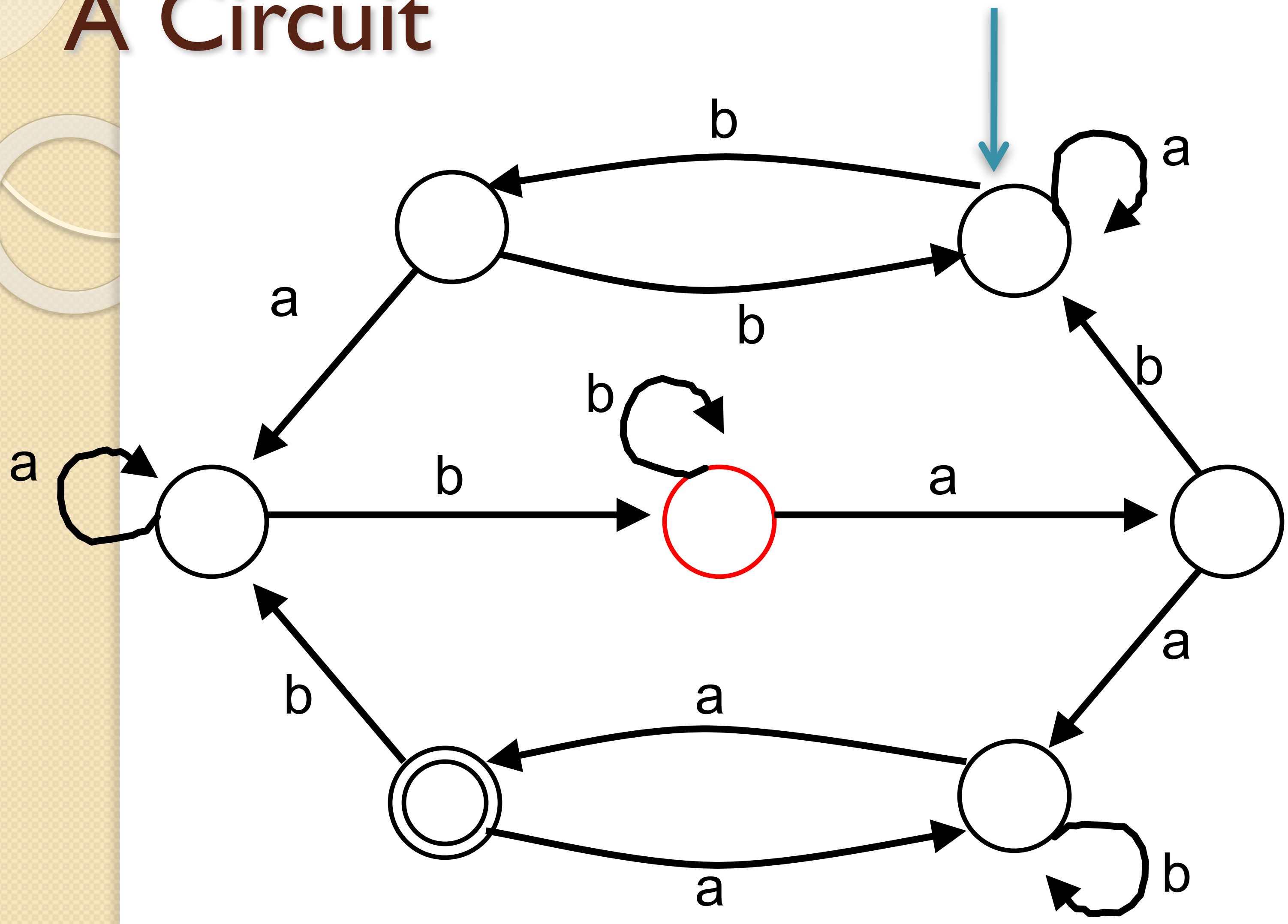
Prove that regular languages are closed under concatenation.

$$L_1L_2 = \{ xy : x \text{ is in } L_1,\ y \text{ is in } L_2 \}$$

Prove that regular languages are closed under *symmetric difference*. (You can use the closure results we've already proved.)

$$L_1 \Delta L_2 = \{ \text{strings in } L_1 \text{ but not in } L_2, \text{ or in } L_2 \text{ but not in } L_1 \}$$

# A Circuit



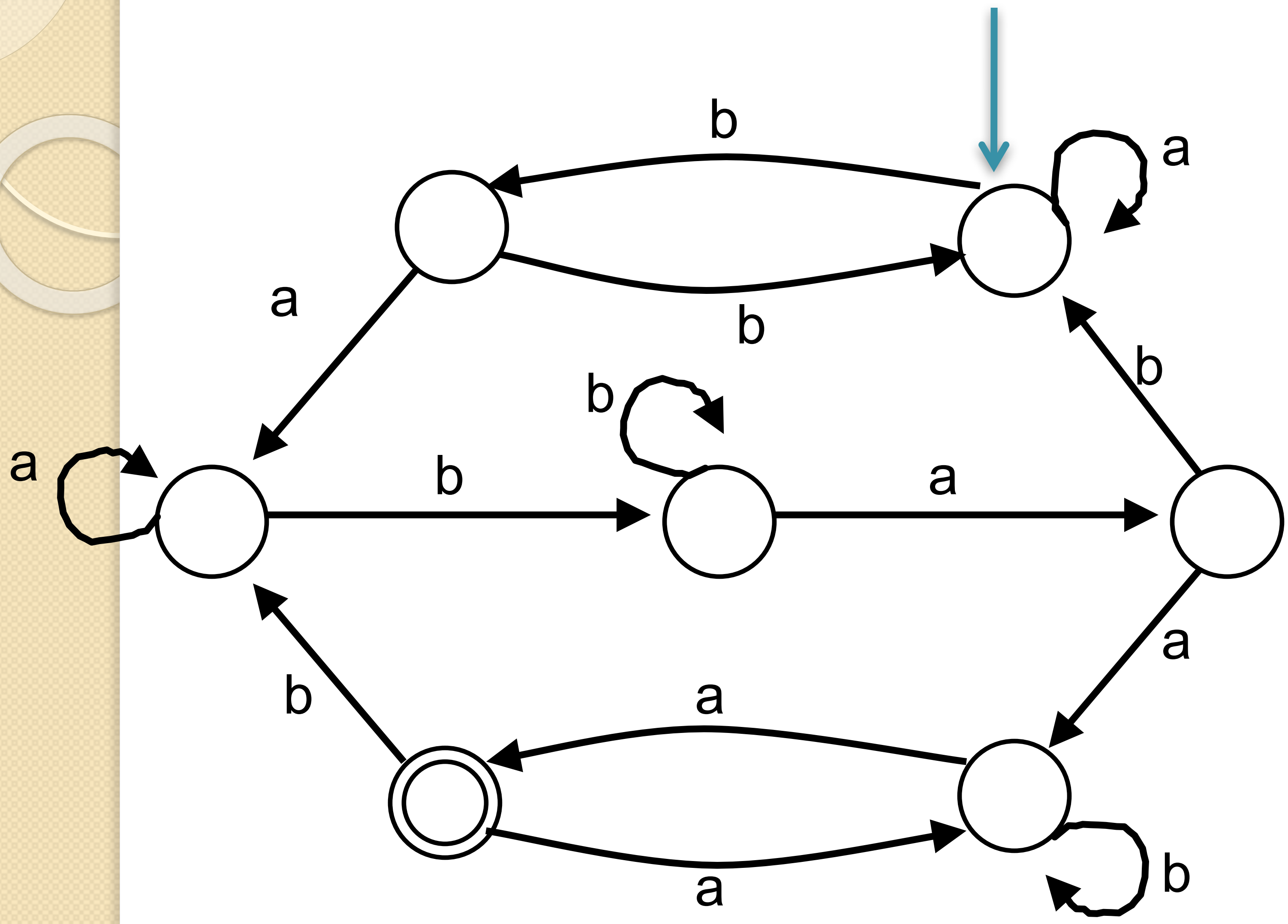

**… abbab ...**

# Definitions

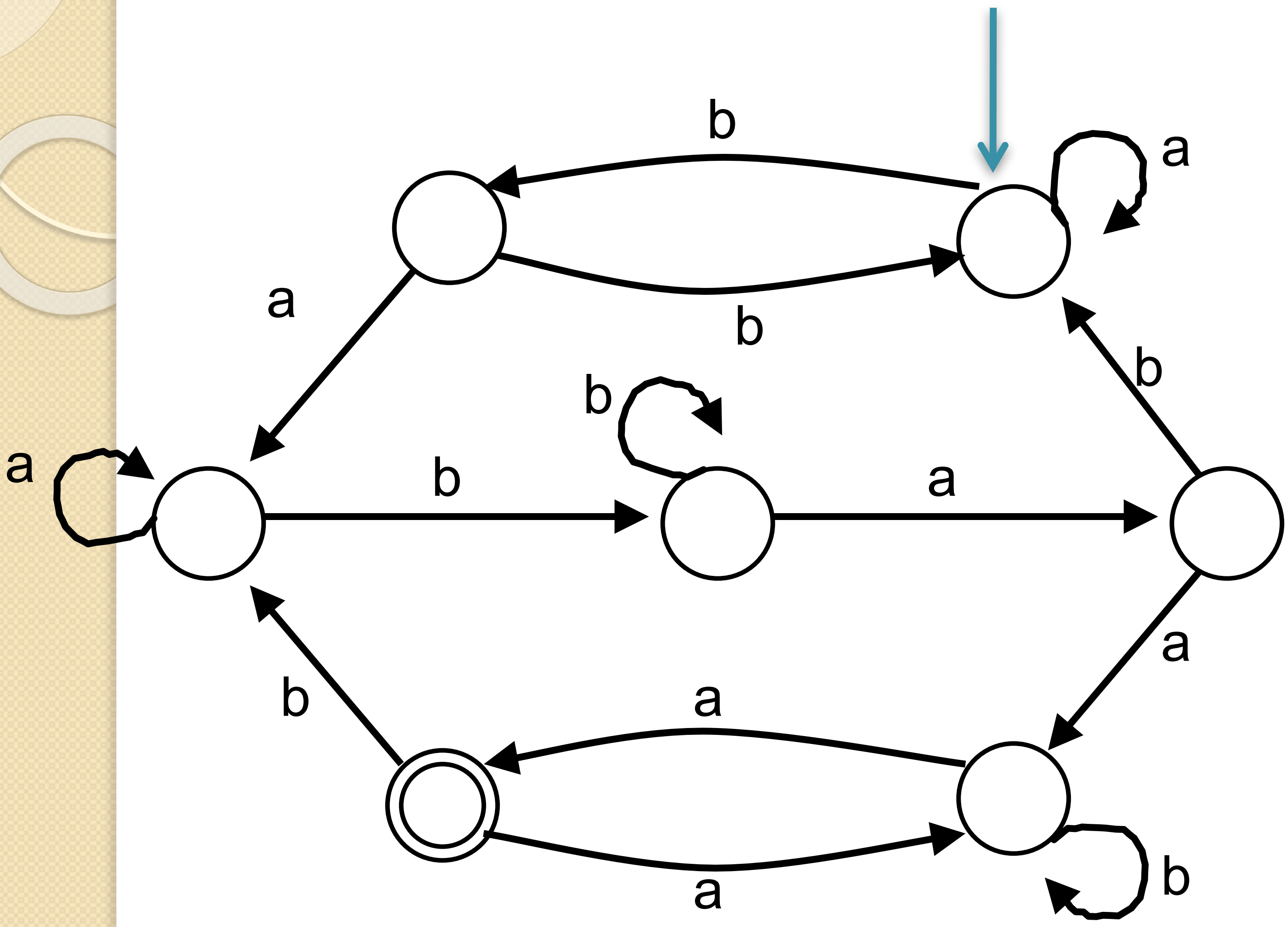
- A **circuit** is a path which starts and ends at the same state.
- The **length** of a circuit is the number of edges in the path.

# Observation

- Take any Finite Automaton.
- Take any string with has more letters than there are states in that Finite Automaton.
- Then the path taken when this string is used as input must contain a circuit.

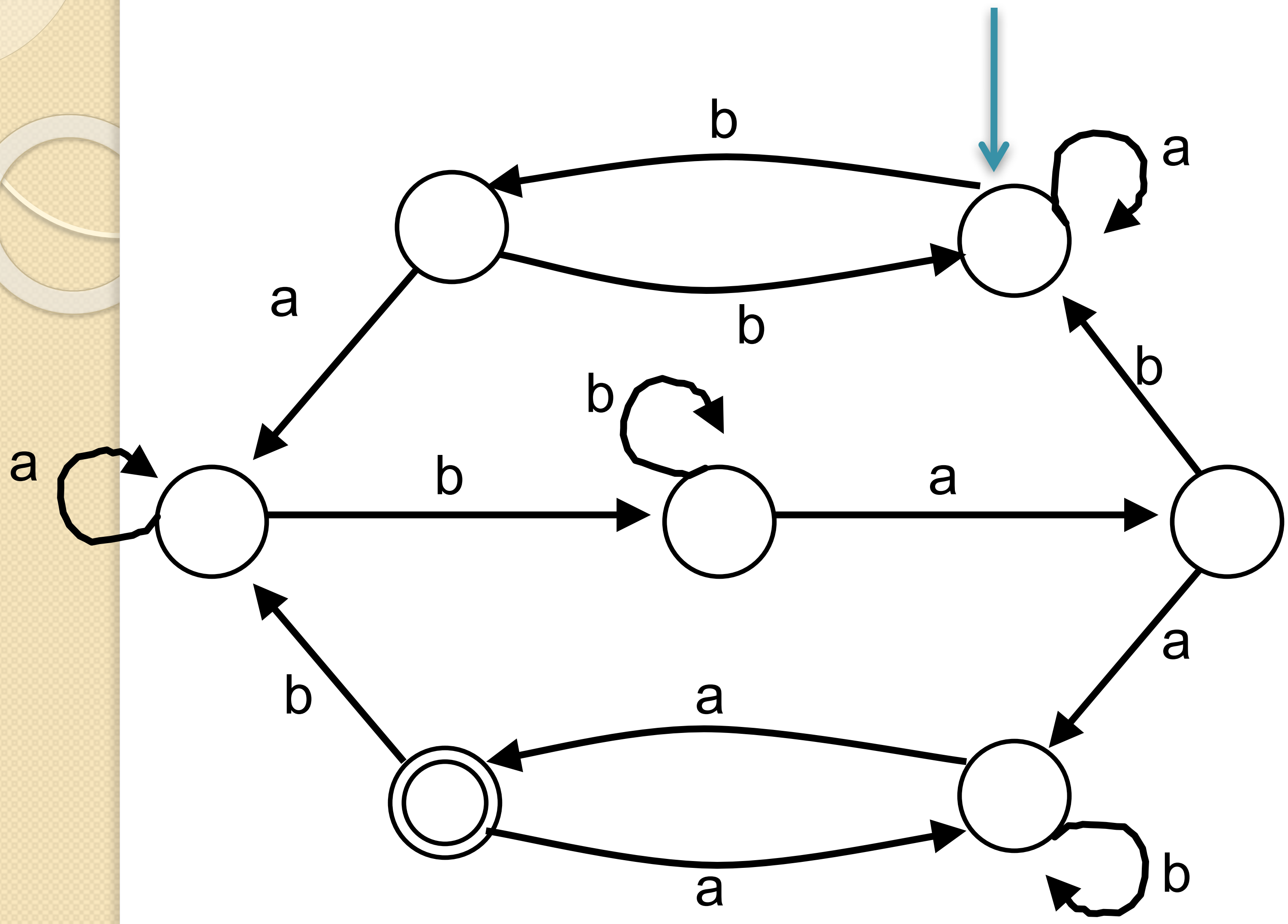



**babaaabbab**



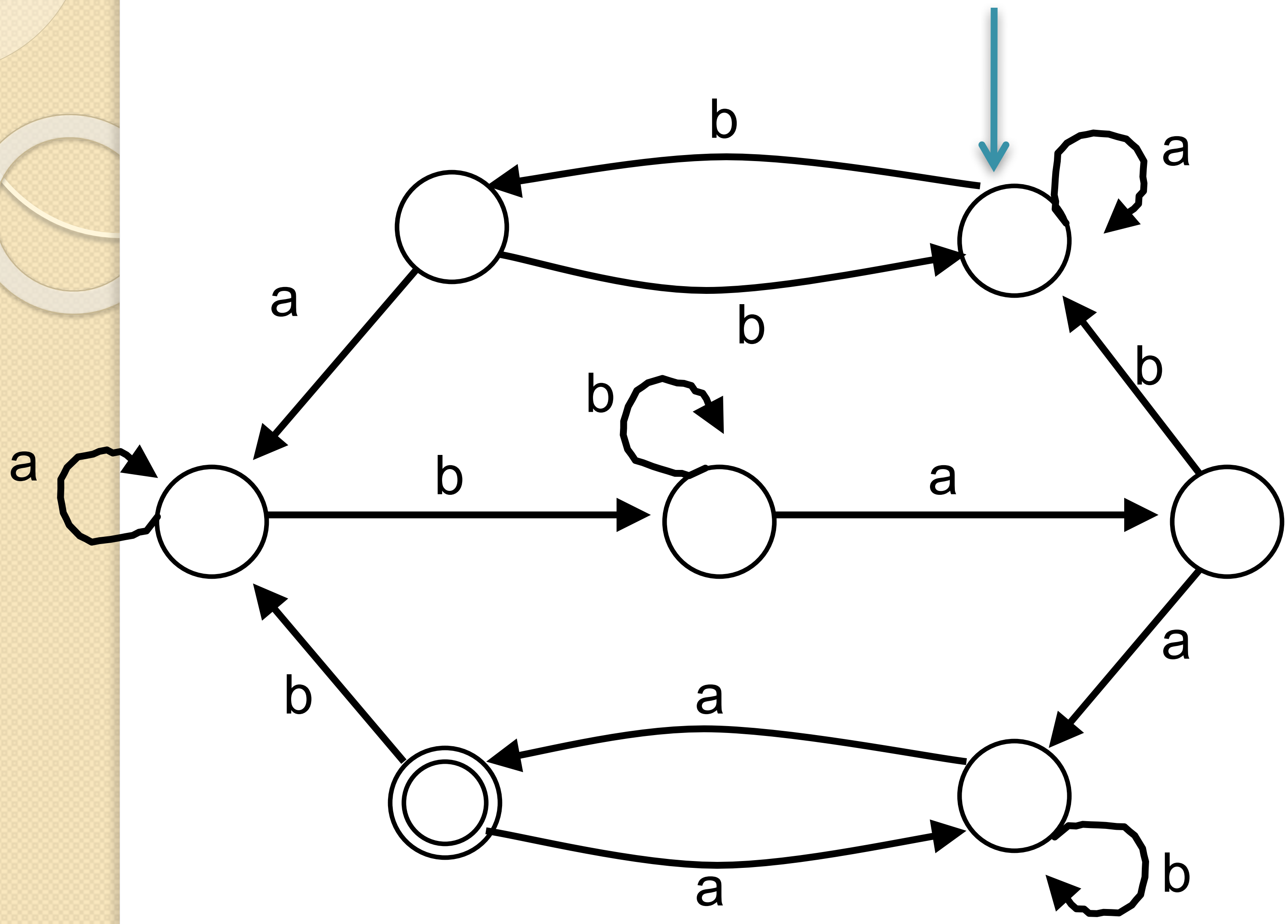

**ba baaab bab**

**x y z**

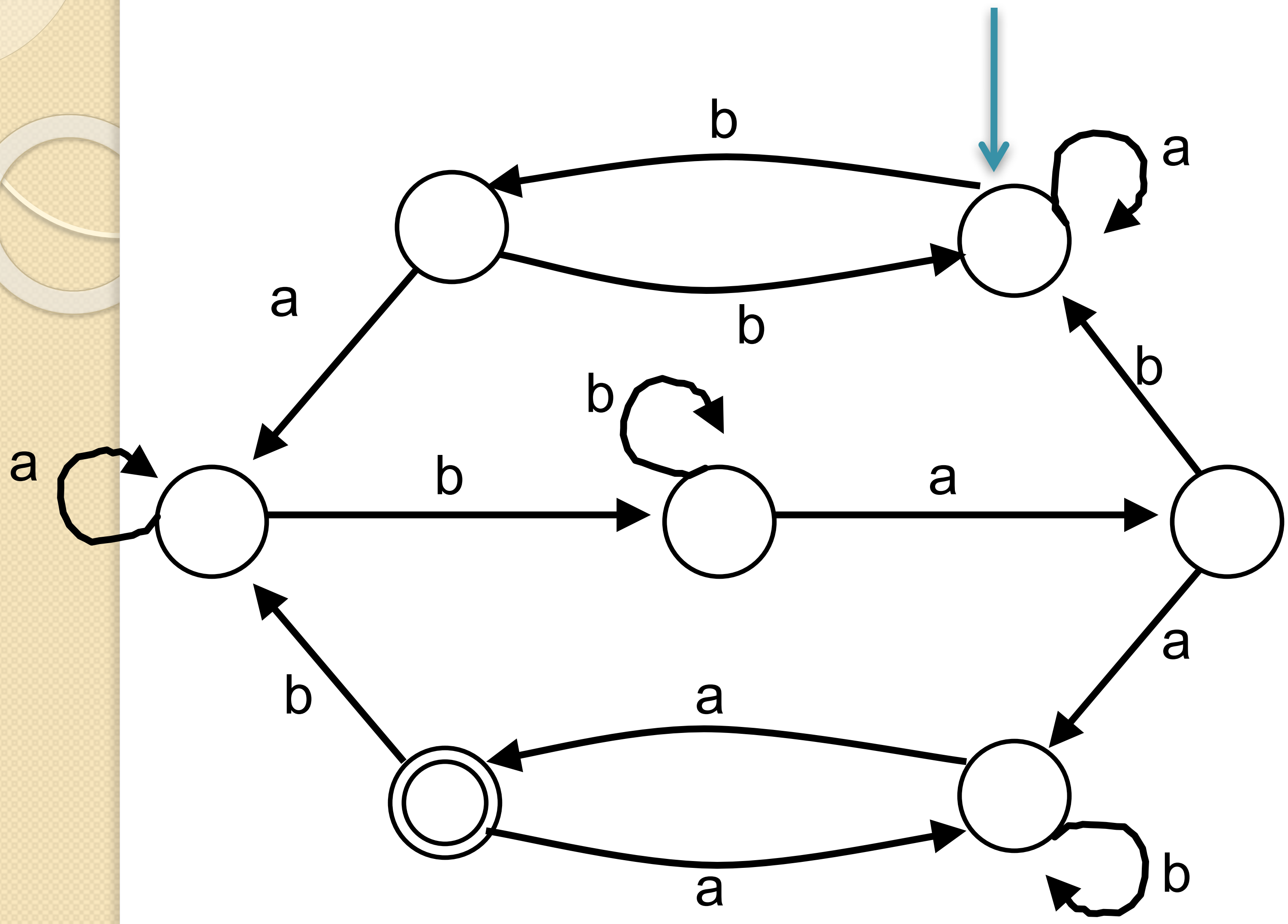



**ba baaab baaab bab**

**x y y z**

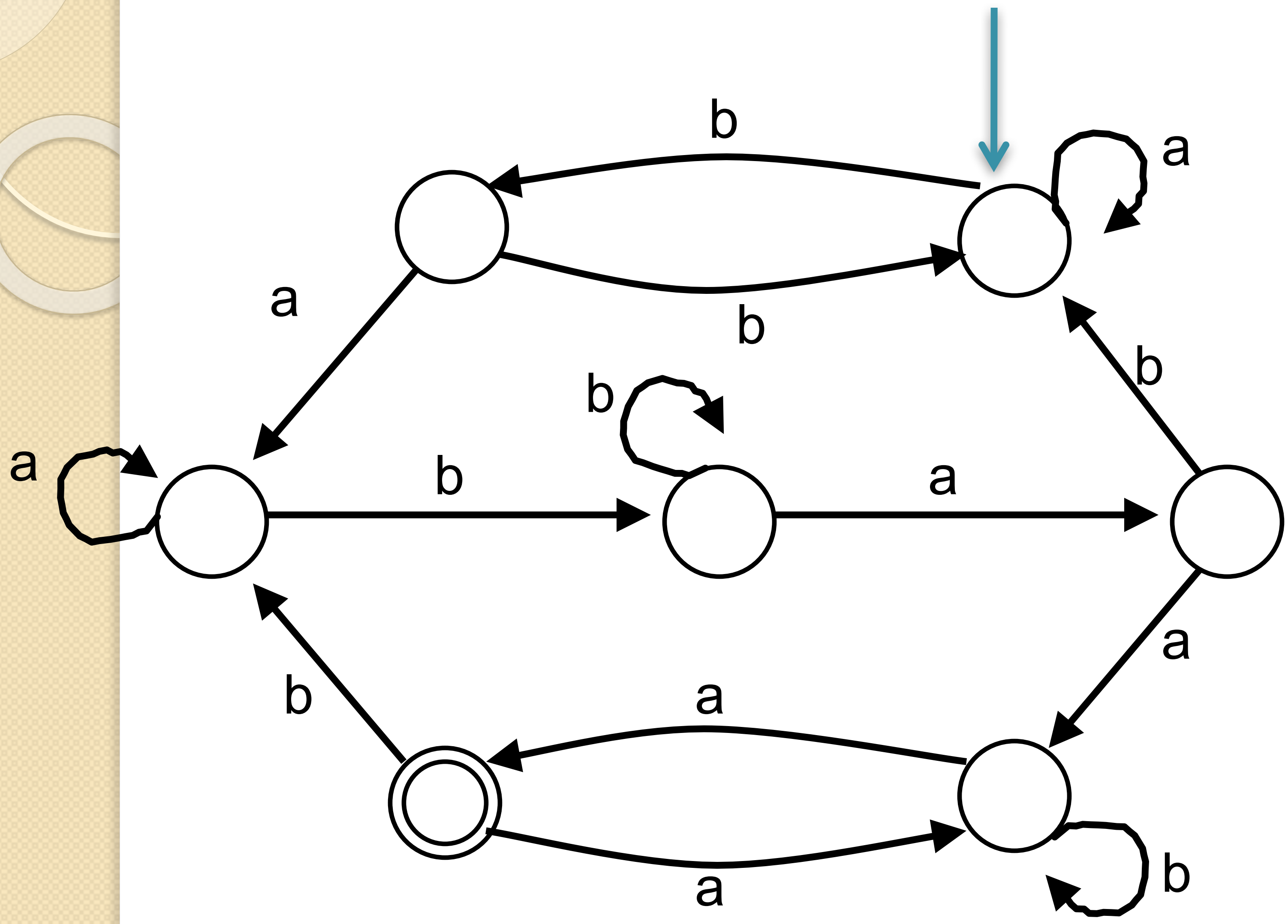



**babbabbababaaba**




**ba bbabba baaba**

**x y z**




**ba bbabba bbabba baaba**

**x y y z**

# Theorem (Pumping Lemma)

- Let **L** be a regular language with an infinite number of words, accepted by a FA with **N** states.
- Then <u>for all</u> words **w** in **L** with more than **N** letters,

<u>there exist</u> strings **x**, **y, z**, with **y** $\neq$ **ε,** such that

- **w = xyz**
- **length(x) + length(y) ≤ N**
- <u>for all</u> $i \geq 0$, $xy^iz$ is in **L**.

**i.e.,**

- $xyz, xyyz, .., xy^nz$ are words in L

# Proof

- Take any word **w** in **L** with > **N** letters.
- Let
  - **x** be the letters up to the first circuit.
  - **y** be the letters corresponding to the circuit.
  - **z** be the remaining letters.
- **w = xyz** by construction.
- **length(x) + length(y) ≤ N**, since the FA reads **xy** without repeating any state.

…

# Proof (continued)

- Since **w = xyz** is in **L**, and **y** starts and finishes at endState(**x**), and **z** goes from endState(**x**) to a Final State, we can repeat y any number of times (or none) and still we end up at the same Final State.

Q.E.D.

# Consequence

- Using the Pumping Lemma we can show there are non-regular languages.
- Method
  - Assume **L** is regular
  - Then, by Kleene's Theorem, it is recognised by some FA.
  - Let **N** be the number of states in this FA.
  - Choose a suitable word **w** in **L**, of length > **N**.
  - Show that, for any **x**, **y** $\neq$ **ε**, and **z** s.t. **w** = **xyz** and |**xy**| $\leq$ **N** …
  - … there exists **N** $\geq 0$ s.t. $\mathbf{xy^Nz}$ is not a word in **L**.
  - Contradiction.

Compare quantifiers above with those in Pumping Lemma

# $L = \{a^n b^n : n \geq 0\}$ HALF-AND-HALF

$L = \{\varepsilon, ab, aabb, aaabbb, \ldots\}$

**Claim**: **L** is not regular.

***Proof*** (by contradiction):

- Assume **L** is regular. Let **N** = # states in an FA for it.
- Choose **w** to be a word in **L** of length > **N**.
- Consider *any* **x**, **y** ≠ **ε**, and **z** s.t. **w** = **xyz**

  *Think:* are **xyz, xyyz, …,** $\mathbf{xy^Nz}$**,…** all in **L**?
- Case 1: **y** is all **a**’s
- Case 2: **y** is all **b**’s
- Case 3: **y** contains an **ab**
- Now consider **xyyz.**
- Contradiction. Q.E.D.

# $L = \{a^n b^n : n \geq 0\}$ HALF-AND-HALF

$L = \{\varepsilon,\ ab,\ aabb,\ aaabbb,\ \ldots\}$

**Claim**: **L** is not regular.

***Shorter Proof*** (by contradiction):

- Assume **L** is regular. Let **N** = # states in an FA for it.
- Choose $\mathbf{w = a^N b^N}$
- Consider *any* **x**, **y** $\neq$ **ε**, and **z** s.t. **w** = **xyz** …

  … and $|\mathbf{xy}| \leq \mathbf{N}$
- ***How many cases now?***
- Now consider **xyyz.**
- Contradiction. Q.E.D.

# EQUAL

- All words which have an equal number of **a**'s as **b**'s.

{**ε, ab, ba, aabb, abab, abba, baba, …** }

- $\{a^n b^n\}$ = **EQUAL** $\cap$ **a*b***
- $\{a^n b^n\}$ is non-regular (as just shown)
- **a*b*** is regular
- Therefore, by closure of regular expressions under intersection,
- **EQUAL** is non-regular

# PALINDROME

- All the strings which are the same if they are spelt backwards

- E.g.

  **ε, a, b, aa, bb, aaa, aba, bab, bbb**

# PALINDROME is non-regular

***Proof*** (by contradiction):

- Assume **PALINDROME** is regular.
- Then exists a FA with **N** states which accepts **PALINDROME**.
- Let $\mathbf{w = a^N b a^N}$
- Consider all strings $\mathbf{x}, \mathbf{y} \neq \boldsymbol{\varepsilon}$, and $\mathbf{z}$ s.t.
  - $\mathbf{w = xyz}$
  - **length(x) + length(y) ≤ N**
- Consider **xyyz**.
- Contradiction Q.E.D.

# *Revision*

- Know the closure properties of regular languages.
- Know what the Pumping Lemma is used to show.
- Know some examples of non-regular languages.

- Sipser, Section 1.4, pp 77-82.

# Preparation

- Read

  M. Sipser, , "***Introduction to the Theory of Computation***", Chapter 2.