Regular Language

A collection S of objects is **closed** under operation *f* if applying *f* to members of S always returns an object still in S.

Regular languages are indeed closed under the regular operations (e.g. union, concatenation, star ⋯)

如果将f应用于S的成员总是返回一个仍然在S中的对象，那么对象的集合S在操作f下是关闭的。

规则语言在规则操作（例如联合，连接，星⋯）下确实是关闭的

Regular Languages Closed Under Union

The set of regular languages is closed under the union operation.

i.e. A and B are regular languages over the same alphabet Σ, then A∪B is also a

regular language.

规则语言集在联合操作下被关闭。

即A和B是同一字母表Σ上的规则语言，那么A∪B也是规则语言。

1. Since A and B are regular languages, there are finite automata M1 = $(Q1, \Sigma, \delta1, q1, F1)$ and M2 = $(Q2, \Sigma, \delta2, q2, F2)$ that accept A and B, respectively.
2. In order to prove that A ∪ B is regular, we have to construct a finite automaton M that accepts A ∪ B. In other words, M must have the property that for every string $w \in \Sigma^*$:

由于A和B是正则语言，因此有有限自动机M1 =（Q1、Σ、δ1、q1、F1）和M2 =（Q2、Σ、δ2、q2、F2）分别接受A和B。

为了证明一个A∪B是正则的，我们必须构造一个接受一个A∪B的有限自动机M。换句话说，M必须具有对于每个字符串w∈Σ *：

DFA:

Proof

Given M_1 and M_2 that $\mathrm{A} = \mathrm{L}(\mathrm{M}_1)$ and $\mathrm{B} = \mathrm{L}(\mathrm{M}_2)$, we can define $\mathrm{M} = (Q, \Sigma, \delta, q, F)$:

- $Q = Q_1 \times Q_2 = \{(q_1, q_2): q_1 \in Q_1 \text{ and } q_2 \in Q_2\}$
- Σ is same as the alphabet of A and B
- $q = (q_1, q_2)$
- $F = \{(q_1, q_2): q_1 \in Q_1 \text{ or } q_2 \in Q_2\}$
- $\delta : Q \times \Sigma \rightarrow Q$

  $\delta((q_1, q_2), a) = (\delta(q_1, a), \delta(q_2, a)), a \in \Sigma$

Proof

- $\delta^*((q_1, q_2), w) = (\delta^*(q_1, w), \delta^*(q_2, w))$
- $\delta^*((q_1, q_2), w) \in F \Leftrightarrow \delta^*(q_1, w) \in F_1 \text{ or } \delta^*(q_2, w) \in F_2$
- M accepts $w \Leftrightarrow \delta^*(q_1, w) \in F_1 \text{ or } \delta^*(q_2, w) \in F_2$
- M accepts $w \Leftrightarrow \mathrm{M}_1$ accepts w or M_2 accepts w

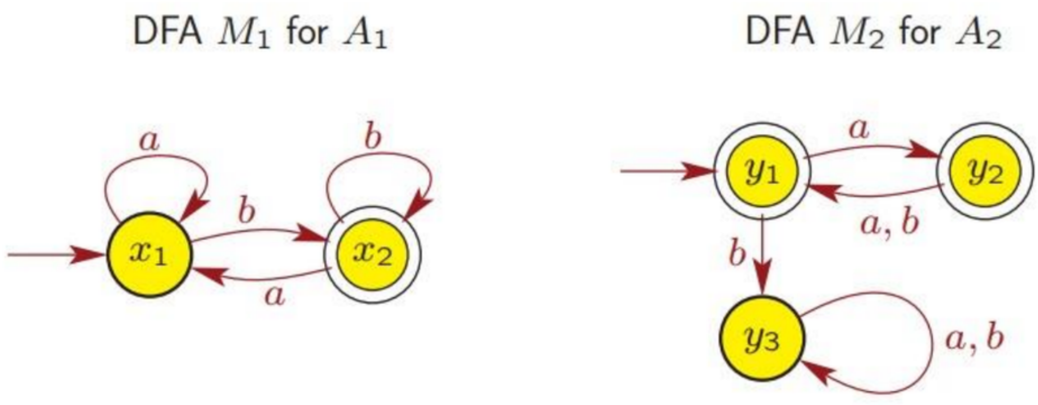
Proved

Example

Consider the following DFAs and languages over $\Sigma = \{a, b\}$:

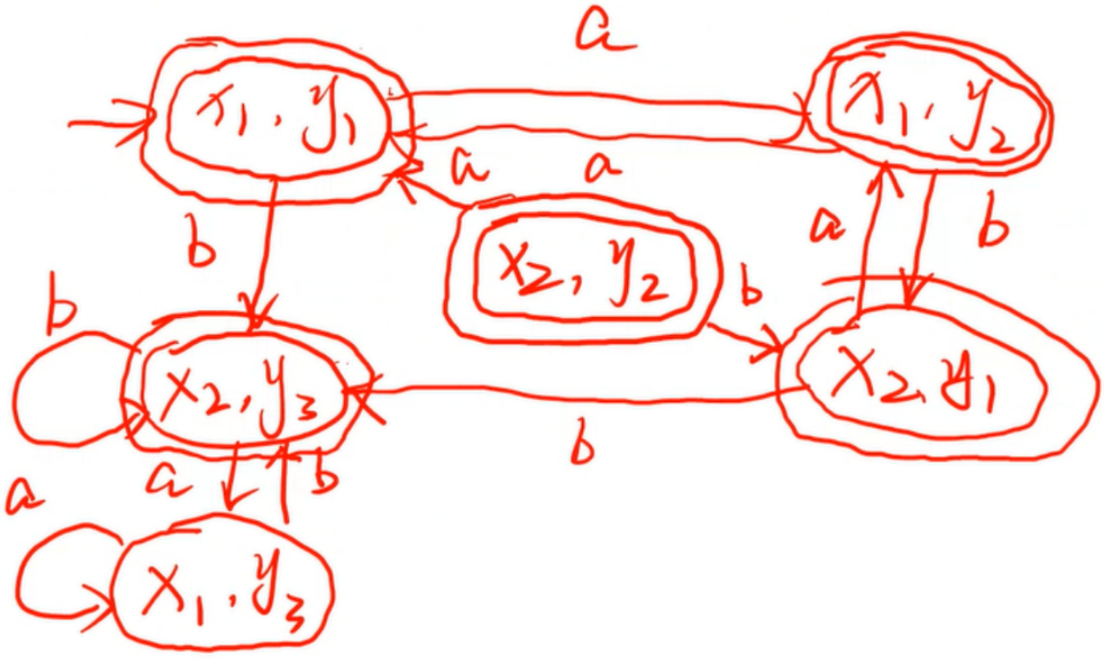
DFA M_1 recognizes $A_1 = L(M_1)$

DFA M_2 recognizes $A_2 = L(M_2)$



DFA M for $A_1 \cup A_2$?

Example



NFA:

Proof

Consider the following NFAs:

NFA $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ recognizes $A_1 = L(M_1)$

NFA $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ recognizes $A_2 = L(M_2)$

We will construct an NFA $M = (Q, \Sigma, \delta, q, F)$

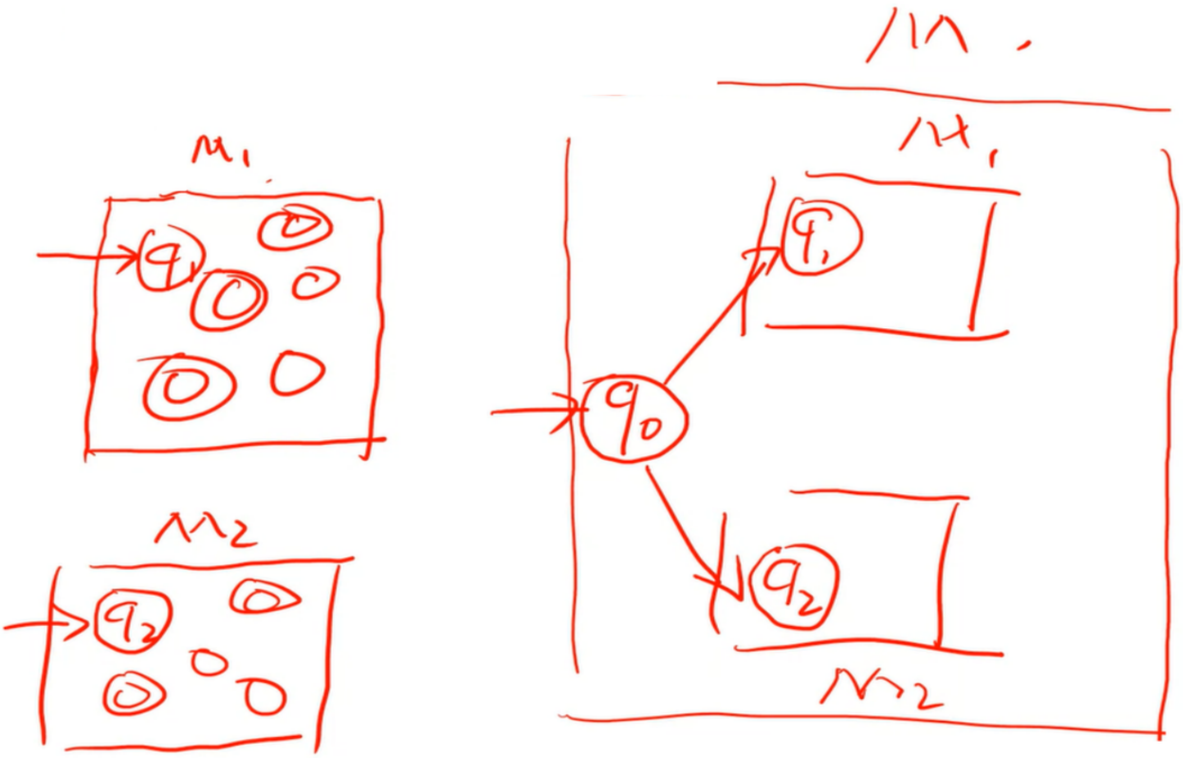
Regular Languages Closed Under Union

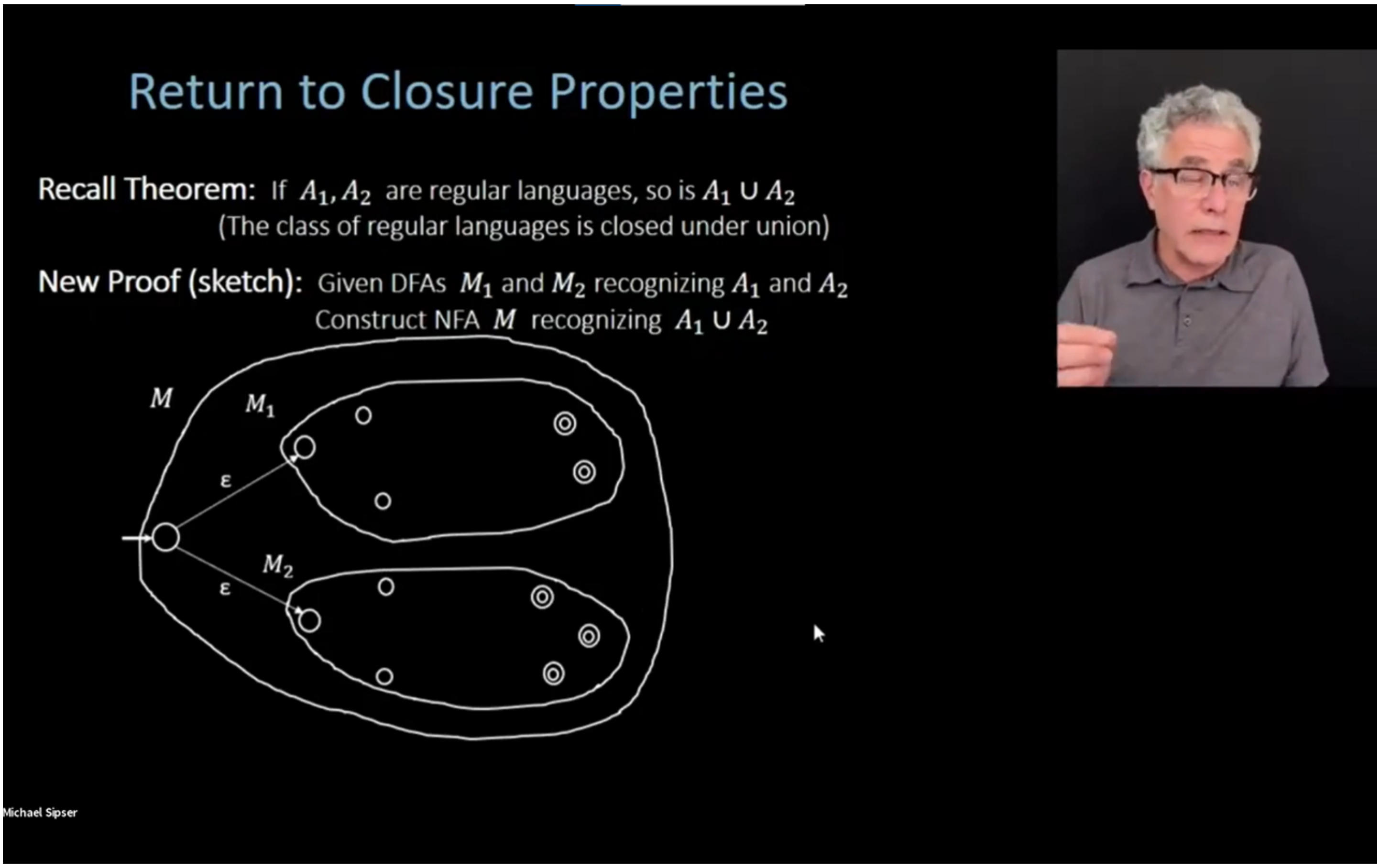
Proof

- $Q = \{q_0\} \cup Q_1 \cup Q_2$
- q_0 is the start state of M
- $F = F_1 \cup F_2$
- $\delta : Q \times \Sigma_\epsilon \rightarrow P(Q)$ is defined as: For any $r \in Q$ and for any $a \in \Sigma_\epsilon$

$$\delta(r, a) = \begin{cases} \delta_1(r, a), & \text{对 } r \in Q_1, \\ \delta_2(r, a), & \text{对 } r \in Q_2 \\ \{q_1, q_2\} & \text{对 } r = q_0 \text{ and } a = \epsilon \\ \emptyset & \text{对 } r = q_0 \text{ and } a \neq \epsilon \end{cases}$$

Proof





Regular Languages Closed Under Concatenation

NFA:

Regular Languages Closed Under Concatenation

The concatenation of A_1 and A_2 is defined as:

$$A_1 A_2 = \{ww' : w \in A_1 \text{ and } w' \in A_2\}$$

Proof

Consider the following NFAs:

NFA $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ recognizes $A_1 = L(M_1)$

NFA $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ recognizes $A_2 = L(M_2)$

We will construct an NFA $M = (Q, \Sigma, \delta, q, F)$ for $A_1 A_2$

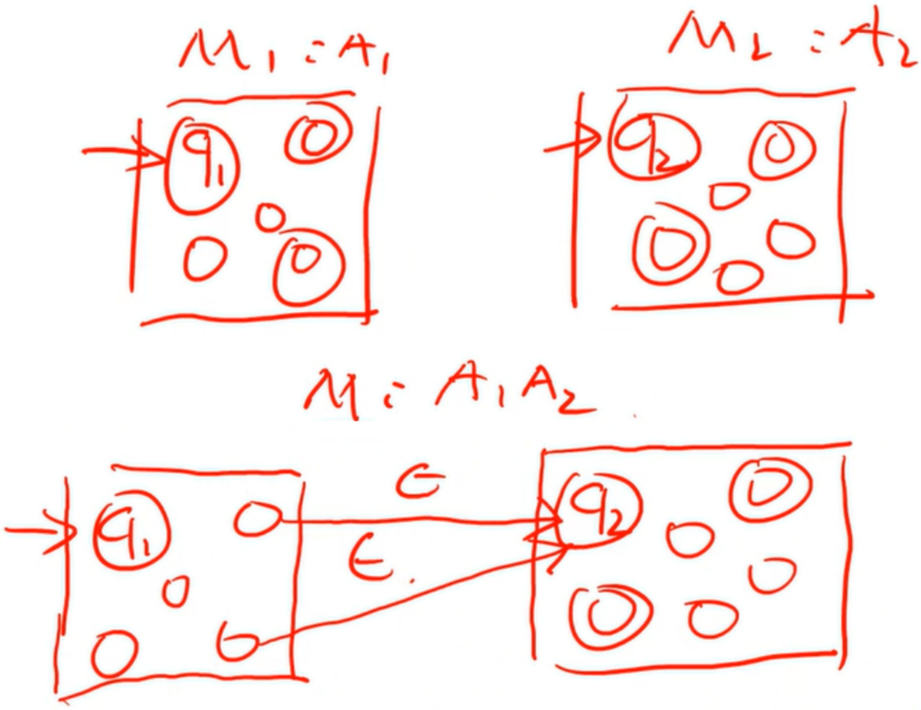
Regular Languages Closed Under Concatenation

Proof

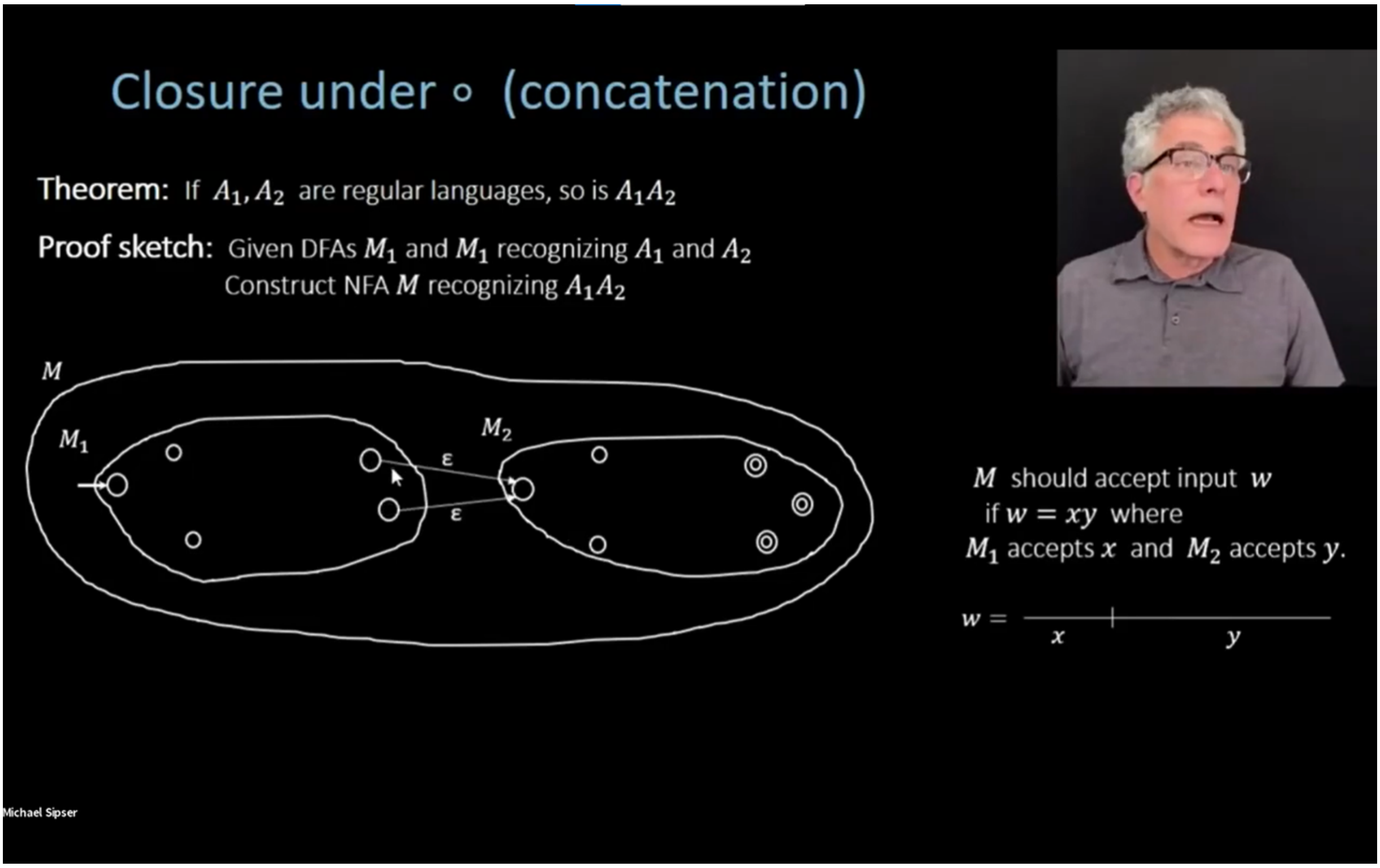
- $Q = Q_1 \cup Q_2$
- M has the same start state as M_1 : q_1
- Set of accept states of M is same as M_2 : F_2
- $\delta : Q \times \Sigma_\epsilon \rightarrow P(Q)$ is defined as: For any $r \in Q$ and for any $a \in \Sigma_\epsilon$

$$\delta(r,a) = \begin{cases} \delta(r,a) & \text{if } r \in Q_1 \text{ and } r \notin F_1 \\ \delta(r,a) & \text{if } r \in F_1 \text{ and } a \neq \epsilon \\ \delta(r,a) \cup (q_2) & \text{if } r \in F_1 \text{ and } a = \epsilon \\ \delta(r,a) & \text{if } r \in Q_2 \end{cases}$$

Proof



① Convert the acceptance states of M_1 ⇒ middle state



Regular Languages Closed Under Kleene star

NFA:

The star of A is defined as:

$$A^* = \{u_1 u_2 \ldots u_k : k \geq 0 \text{ and } u_i \in \mathbf{A} \text{ for all } i = 1, 2, \ldots, k\}$$

Proof

Consider the following NFA:

NFA $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ recognizes $A = L(M_1)$

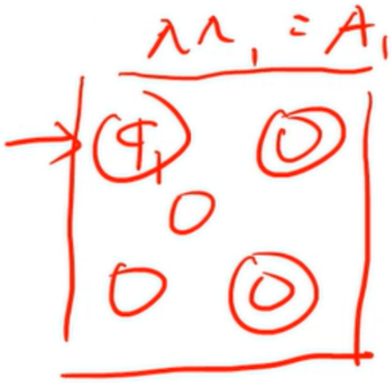
We will construct an NFA $M = (Q, \Sigma, \delta, q, F)$ for A^*

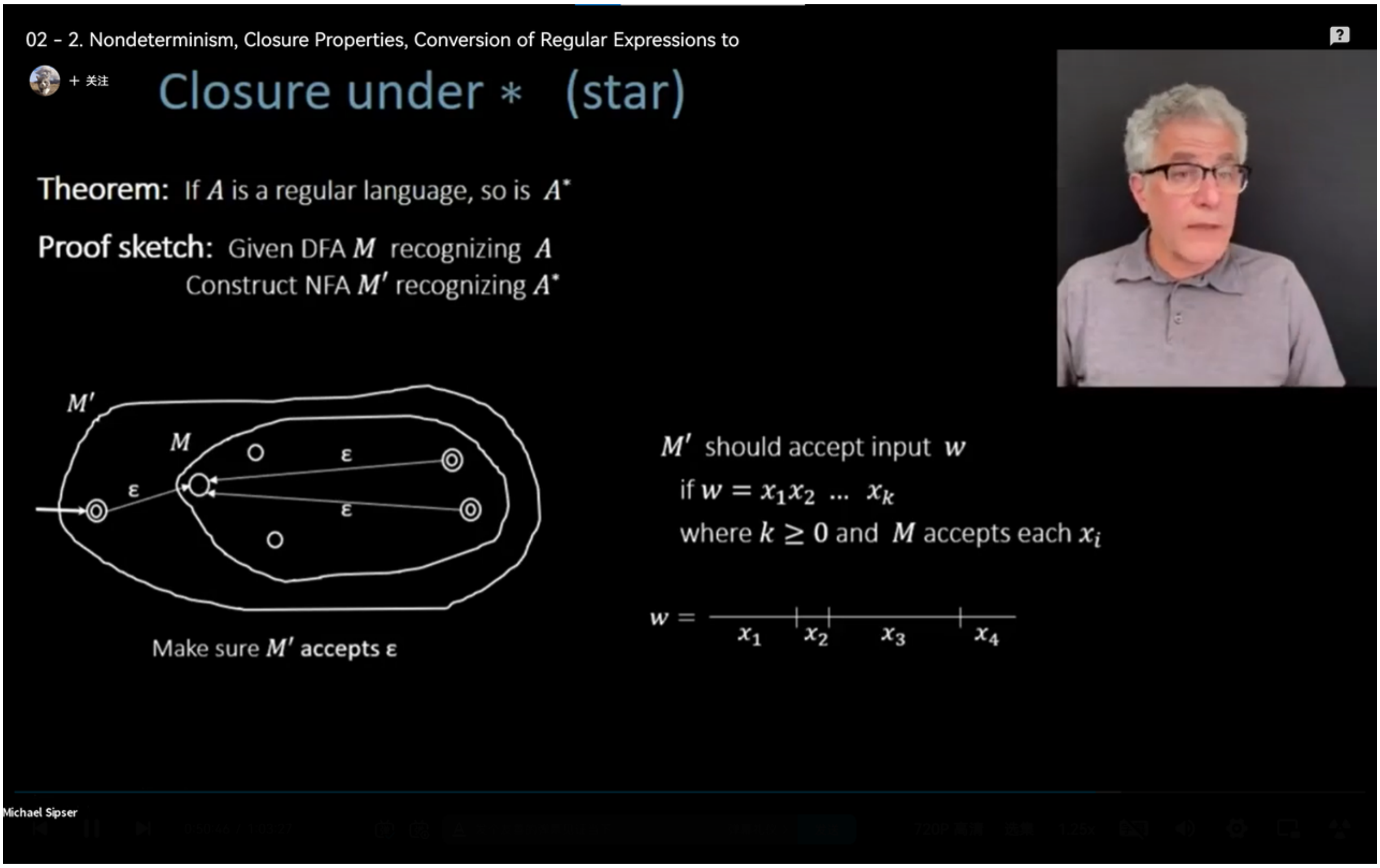
Proof

- $Q = \{q_0\} \cup Q_1$
- q_0 is the start state of M
- $F = \{q_0\} \cup F_1$
- $\delta : Q \times \Sigma_\epsilon \rightarrow P(Q)$ is defined as: For any $r \in Q$ and for any $a \in \Sigma_\epsilon$

$$\delta(r,a) = \begin{cases} \delta(r,a), & \text{if } r \in Q_1 \text{ and } r \notin F_1 \\ \delta(r,a) & \text{if } r \in F_1 \text{ and } a \neq \epsilon \\ \delta(r,a) \cup \{q_1\} & \text{if } r \in F_1 \text{ and } a = \epsilon \\ \{q_1\} & \text{if } r = q_0 \text{ and } a = \epsilon \\ \phi & \text{if } r = q_0 \text{ and } a \neq \epsilon \end{cases}$$

Proof





Regular Languages Closed Under Complement and Interaction

The set of regular languages is closed under the complement and interaction operations:

- If A is a regular language over the alphabet Σ, then the complement:

$$\overline{\mathrm{A}} = \{w \in \Sigma^*: w \notin \mathrm{A}\}$$

  is also a regular language.

- If A_1 and A_2 are regular languages over the same alphabet Σ, then the interaction:

$$\mathrm{A}_1 \cap \mathrm{A}_2 = \{w \in \Sigma^*: w \in \mathrm{A}_1 \text{ and } w \in \mathrm{A}_2\}$$

  is also a regular language.

Regular Expressions

Regular expressions are means to describe certain languages.

正则表达式是描述某些语言的方法。

Example

Consider the expression:

$$(0 \cup 1)01^*$$

The language described by this expression is the set of all binary strings satisfy:

- that start with either 0 or 1 (this is indicated by $(0 \cup 1)$),
- for which the second symbol is 0 (this is indicated by 0),
- that end with zero or more 1s (this is indicated by 1^*).

Example

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

The language $\{w : w \text{ contains exactly two } 0\text{s}\}$ is described by the expression:

$1^*01^*01^*$

The language $\{w : w \text{ contains at least two } 0\text{s}\}$ is described by the expression:

$(0 \cup 1)^*0(0 \cup 1)^*0(0 \cup 1)^*$

The language $\{w : 1011 \text{ is a substring of } w\}$ is described by the expression:

$(0 \cup 1)^*1011(0 \cup 1)^*$

Formal Definition of regular expressions

Let Σ be a non-empty alphabet.

1. ☒ is a regular expression.

☒是一个正则表达式。

2. ∅ is a regular expression.

∅是一个正则表达式。

3. For each $a \in \Sigma$, a is a regular expression.

对于每个a∈Σ，a都是一个正则表达式

4. If $R1$ and $R2$ are regular expressions, then $R1 \cup R2$ is a regular expression.
5. If $R1$ and $R2$ are regular expressions, then $R1\ R2$ is a regular expression.
6. If R is a regular expression, then R⊠ is a regular expression

Formal Definition of regular expressions

If R is a regular expression, then L(R) is the **language** generated (or described or

defined) by R.

Let Σ be a non-empty alphabet.

1. The regular expression ⊠ describes the language {⊠}.
2. The regular expression ∅ describes the language ∅.
3. For each $a \in \Sigma$, the regular expression a describes the language {a}.
4. Let $R1$ and $R2$ be regular expressions and let L1 and L2 be the languages described by them, respectively. The regular expression $R1 \cup R2$ describes the language L1 ∪ L2 .
5. Let $R1$ and $R2$ be regular expressions and let L1 and L2 be the languages described by them, respectively. The regular expression $R1R2$ describes the language L1L2 .
6. Let R be a regular expression and let L be the language described by it. The regular expression R⊠ describes the language L⊠ .

1.正则表达式⊠描述了语言{⊠}。

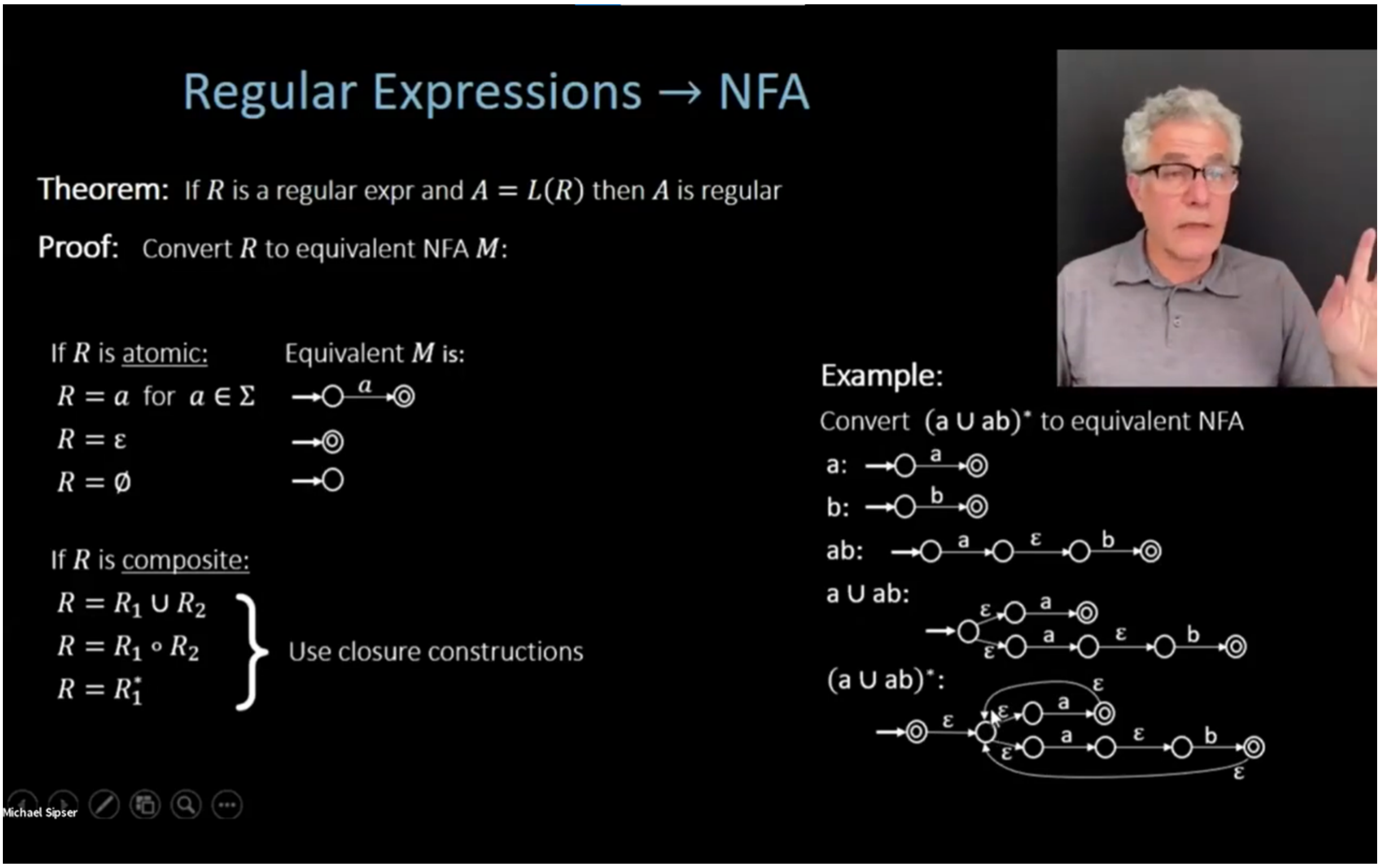
2.正则表达式∅描述了语言∅。

3.对于每个∈Σ，正则表达式a描述了语言{a}。

4.设r1和r2是正则表达式，设l1和l2分别是它们所描述的语言。正则表达式R1∪R2描述了语言L1∪L2。

5.设r1和r2是正则表达式，设l1和l2分别是它们所描述的语言。正则表达式R1R2描述了语言L1L2。

6.设R是一个正则表达式，设L是它所描述的语言。正则表达式R⊠描述了语言L⊠。



Example

Given $(0 \cup 1)^* 101 (0 \cup 1)^*$, prove it is a regular expression (note: $\Sigma = \{0, 1\}$).