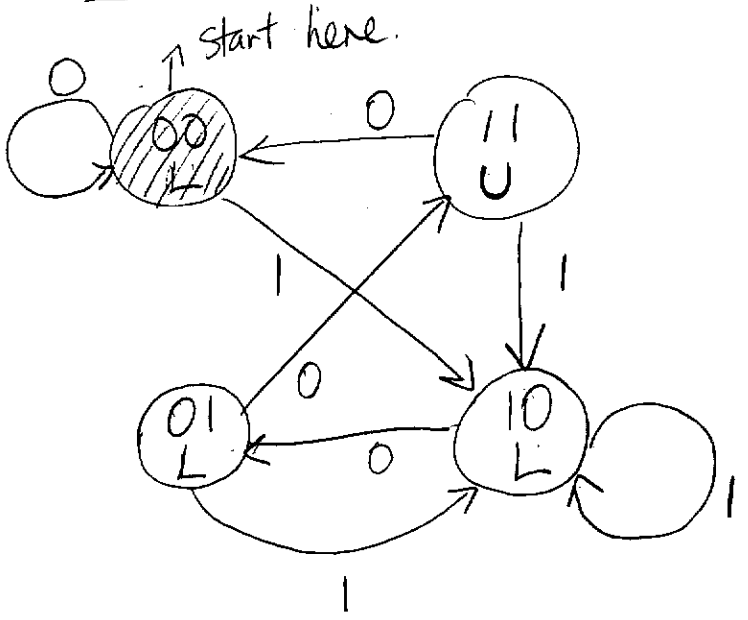
① A digital lock with a combination of 100.

Two buttons 0 , 1 ; two outputs locked or unlocked

100 / 110



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

$S = \{00, 01, 10, 11\}$

S_0 = shaded $\{00\}$

m is the pointers

$F = \{11\}$

100, 110 are called words (denote as x)

The diagram is called state diagram.

A finite state machine is a mathematical model of a system with discrete inputs and outputs. The system may be in any one of a finite number of internal configurations or states. (write title FSM)

Ex: 00, 01, 11, 10 are states.

Formal defn: A FSM over the alphabet Σ is a system $M = (S, m, S_0, F)$ where.

S is a finite set of states

m is state transition function with two variables, current state S, and current input character σ ($\sigma \in \Sigma$) then $m(S, \sigma)$ is the "next" state.

(2) $S_0 \in S$ is the initial state.

$F \subset S$ is the set of accepting states.

Ex:

Let Σ^* be the set of all words over Σ, Ex: {110, 111} (101 1100)

A language is a subset of Σ^*. Ex: {100, 111}.

Def 2: For $M = (S, m, S_0, F)$, $L(M) = \{x \in \Sigma^* : m(S_0, x) \in F\}$

Simple words: L is a subset of Σ^* s.t. the final state is an accepting state.

Ex: {100} is a regular language.

L is regular if $\exists$ a FSM M s.t. $L = L(M)$.

Note: A regular language can have more than one FSM. Intuitively, we can always add some redundant states to change FSM but still define the same language. $L \leftrightarrow$ FSM (one to many)

Def 4: a semigroup is a system $(S, \cdot)$ where S is a nonempty set and $\cdot$ is a binary operation (called multiplication), s.t. if $x, y \in S$, $x \cdot y \in S$. Also the product is associative: if $x, y, z \in S$. $(xy)z = x(yz)$

Ex: S is {all integers > 1} and $\cdot$ is regular mult.

(3) $(S, \cdot)$ is a semigroup.

Def 5: monoid is a semigroup with a unique identity element $e \in S$ with $ex = xe = x$ for all $x \in S$.

ex: replace S by {all integers ≥ 1}, $\cdot$ the same, then $(\mathbb{Z}^+, \cdot)$ is monoid with $e = 1$.

For the rest of talk, I'm going to prove two theorems to show that there exists a unique FSM with minimal states for a regular language.

Ex: Let S be Σ^* with empty word ε, $\cdot$ be the operation of concatenation of words, then $(\Sigma^*, \cdot)$ is a monoid with identity element ε.

Def 6: The substitution property of an equivalence relation: For a semigroup G, and all $x, x', y, y' \in G$

$$x \theta x' \text{ and } y \theta y' \Rightarrow xy \theta x'y'$$

Def:
θ Equivalence class is a collection of element x.

$$x/\theta = \{x' \in S : x' \theta x\}$$

the multiplication of congruence class is

$$x/\theta \cdot y/\theta = (xy)/\theta$$

The set of all θ congruence classes is denoted by

④ Refinement ?

index of an equi. relation is the number of equi. classes

Def 7: A equi. relation θ on a semigroup G called right stable (or left stable respectively) if for all $x, y, z \in G$

$$x \theta y \Rightarrow xz \; \theta \; yz$$

Ex: Let $\Sigma = \{0, 1, 2\} \Rightarrow \Sigma^* = \{0, 1, 2\}^*$ then define an equivalence relation on $(\Sigma^*, \cdot)$

$x \theta y$ iff x and y both begin with a character 0, 1, 2.

∴ There are 3 equi. classes {all words 0---, 1---, 2---.}

This is a right stable relation for $z \in (\Sigma^*, \cdot)$

$$x \theta y \Rightarrow xz \; \theta \; yz$$ keep this

∴ The index is 3.

(5) (Myhill–Nerode Theorem) Theorem 1:

For a set $L \subset \Sigma^*$ of words, the following are equivalent:

1) $L = L(M)$ for some finite state machine M.
(or L is regular)

2) For some right stable equivalence relation θ' on the monoid $(\Sigma^*, \cdot)$ having finite index, L is the union of some θ' equivalence classes.

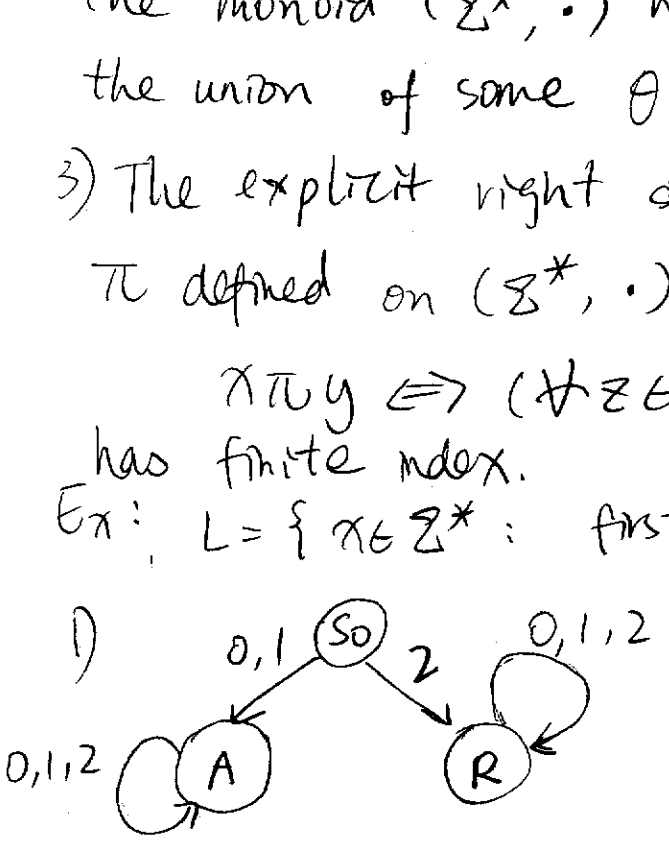
3) The explicit right stable equivalence relation π defined on $(\Sigma^*, \cdot)$ by

$$x \pi y \iff (\forall z \in \Sigma^*)(xz \in L \iff yz \in L).$$

has finite index.

Ex: $L = \{x \in \Sigma^* :$ first characters in x is either 0 or 1$\}$

1)



There exists a FSM for this language.

2) We found (on last page) that the index is 3.
and L is the union of two equi. classes

3) The relation π partition $\{0, 1, 2\}^*$ into 2 classes: L and its complement.
so it has finite index.

⑥ proof. 1)→2) Define θ' on Σ^* by $x\theta' y \Leftrightarrow m(s_0, x) = m(s_0, y)$

Equality is an equivalence relation so we only need to prove right-stability.

$x\theta' y \Rightarrow m(s_0, x) = m(s_0, y) \Rightarrow$

$m(s_0, x \cdot z) = m(m(s, x), z) = m(m(s, y), z)$

$= m(s, y \cdot z)$

$\Rightarrow xz\theta' yz$.

Because we partition the set by states, the number of classes should be at most $|S|$ (which is finite). $\therefore$ index $\theta' \leq |S|$ so it has finite state.

Also, $x \in L(M) \Leftrightarrow m(s_0, x) \in F$. Hence if $x\theta y$ then $m(s_0, y) = m(s_0, x) \in F$ so $y \in L(M)$.

$\therefore L = L(M)$ is the union of θ' equi. classes.

2)→3) suppose L is union of θ' classes for some right-sta. θ' of finite index. Suppose $x\theta' y$ and $xw \in L$ for some w then $xw\,\theta' yw$ by right-stab. L is the union of θ' classes. $\therefore yw \in L$

Thus $x\theta' y \Rightarrow x \pi y$ $\therefore$ θ' is a refinement of π $\therefore$ index $\pi \leq$ index $\theta' =$ finite

Ex: $\theta' =$ words starts with $\{0,1,2\}$ $\pi =$ words starts/end.,

$\pi \Rightarrow \theta'$ $\therefore$ index $\theta' \leq$ index π $3 \leq 9$

⑦

3) → 1) Construct a Fsm $M_\pi(S_\pi, m_\pi, S_{0\pi}, F_\pi)$

s.t. $S_\pi = \{x/\pi : x \in \Sigma^*\}$; $|S|$ = index π (π classes)

$m_\pi = m(x/\pi, \sigma) = (x\sigma)/\pi$ where σ is input character well defined due to right stab.

$S_{0\pi} = \varepsilon/\pi$;

$F_\pi = \{x/\pi : x \in L\} \subset S$. since $L \subset \Sigma^*$

$m_\pi(S_{0\pi}, x) \in F_\pi$

$\Leftrightarrow m_\pi(\varepsilon/\pi, x) \in \{x/\pi : x \in L\}$

$\Leftrightarrow (\varepsilon x)/\pi = x/\pi \in \{x/\pi : x \in L\}$

$\Leftrightarrow x \in L$.

Note: it's easier to think x/π as mod

To establish ① we need to show that for any $x \in L \Leftrightarrow m(S_0, x) \in F$.

Def. 8 For a $M = (S, m, s_0, F)$ to be isomorphic to $M_\pi = (S_\pi, m_\pi, S_{0\pi}, F_\pi)$. ∃ a 1-1 correspondence between the set S of states of M and the set S_π of states of M_π.

a) $s_0 \longmapsto S_{0\pi}$

(8) b) F correspond to states in F_π

c) for all $s \in S$, if $s \mapsto s_\pi$, then for any $\sigma \in \Sigma$, $m(s,\sigma) \mapsto m_\pi(s_\pi, \sigma)$.

Simple words: $\exists$ a 1-1 correspondence between two sets of machine and if we rename the states of one by this correspondence, it becomes the same machine as the other.

Theorem 2: A minimum FSM defining a given regular language is unique up to isomorphism and is described by the machine M_π of theorem 1.

proof: ① prove M_π is minimum.

In theorem 1, for <u>any</u> FSM M for L,

$\exists$ a θ' on Σ^* s.t. $\theta \leq \pi$.

$\therefore$ Index $\pi \leq$ index θ' (for any FSM)

We also show index of $\theta' \leq |S|$, the # of states of M

$\therefore$ Index $\pi = |S_\pi| \leq$ index $\theta' \leq |S|$

$\therefore |S_\pi| \leq |S|$ for any FSM M

(19) 2) prove of uniqueness

$|S| = |S_{\pi}|$

Let M be any other MFSM. States of M = states of M_{π}

Now we need to find a 1-1 correspondence to yield isomorphism.

For any state s in S, $\exists$ an $x \in \Sigma^*$ s.t. $m(s_0, x) = s$

O/W we can delete this m accessible state. (contradicted with minimum)

If $m(s_0, x) = m(s_0, y) = s \Rightarrow x \theta' y$

$\Rightarrow x \pi y$

$\therefore m(s_{0\pi}, x) = m(s_{0\pi}, y)$

For any state of M and any x for which $m(s_0, x) = s$,

$s = m(s_0, x) \longmapsto M_{\pi}(s_{0\pi}, x) = s_{\pi}$

is a well-defined 1-1 function of states of M onto states of M_{π}

$m(s, y) = m(m(s_0, x), y) = m(s_0, xy) \longmapsto$

$M_{\pi}(s_{0\pi}, xy) = M_{\pi}(M_{\pi}(s_{0\pi}, x), y) = M_{\pi}(s_{\pi}, y)$

a) is from. $[m(s_0, \varepsilon) = s_0 \longmapsto M_{\pi}(s_{0\pi}, \varepsilon) = s_{0\pi}$

$\therefore s_0 \longmapsto s_{0\pi}]$ b) from [both machines define

L $\therefore$ $m(s_0, x) \in F \Leftrightarrow M_{\pi}(s_{0\pi}, x) \in F_{\pi}$