MODUL II

TEORI BAHASA DAN AUTOMATA

Tujuan :

Mahasiswa memahami Finite State Automata (FSA) dan dapat mengeksekusi suatu mesin otomata

Materi :

- FSA dan Implemetasi FSA
- Deterministic Finite Automata (DFA)
- Non Deterministic Finite Automata (NFA)

FINITE STATE AUTOMATA
DAN IMPLEMENTASINYA

Finite State Automata (FSA) adalah suatu mesin abstrak yang digunakan untuk merepresentasikan penyelesaian suatu persoalan dari suatu sistem diskrit. Sebagai sebuah mesin maka FSA akan bekerja jika diberikan suatu masukan. Hasil proses adalah suatu nilai kebenaran diterima atau tidaknya masukan yang diberikan.
FSA memiliki state yang banyaknya berhingga, jika diberikan suatu simbol input maka dapat terjadi suatu perpindahan dari sebuah state ke state lainnya. Perubahan state tersebut dinyatakan oleh suatu simbol transisi. Mekanisme FSA tidak memiliki memori sehingga selalu mendasarkan prosesnya pada posisi state “saat ini”. Misalnya pada mekanisme kontrol pada sebuah lift, selalu didasari pada posisi lift saat itu pada suatu lantai, pergerakan ke atas atau ke bawah dan sekumpulan permintaan yang belum terpenuhi.

Finite State Automata merupakan suatu tool yang berguna untuk merancang sistem nyata.

Sebagai contoh pada penyelesaian kasus: seorang petani dengan seekor serigala, kambing dan seikat rumput berada pada suatu sisi sungai. Tersedianya hanya sebuah perahu kecil yang hanya dapat dimuati dengan petani tersebut dengan salah satu serigala, kambing atau rumput. Petani tersebut harus menyeberangkan ketiga bawaannya kesisi lain sungai. Tetapi jika petani meninggalkan serigala dan kambing pada suatu saat, maka kambing akan dimakan serigala. Begitu pula jika kambing ditinggalkan dengan rumput, maka rumput akan dimakan oleh kambing. Mungkinkah ditemukan suatu cara untuk melintasi sungai tanpa menyebabkan kambing atau rumput dimakan.

Masalah ini dapat dimodelkan dengan memperhatikan mereka yang menempati setiap sisi sungai. Misalnya kita akan menggunakan kombinasi pada setiap sisi sungai sebagai keadaan/state yang mungkin.

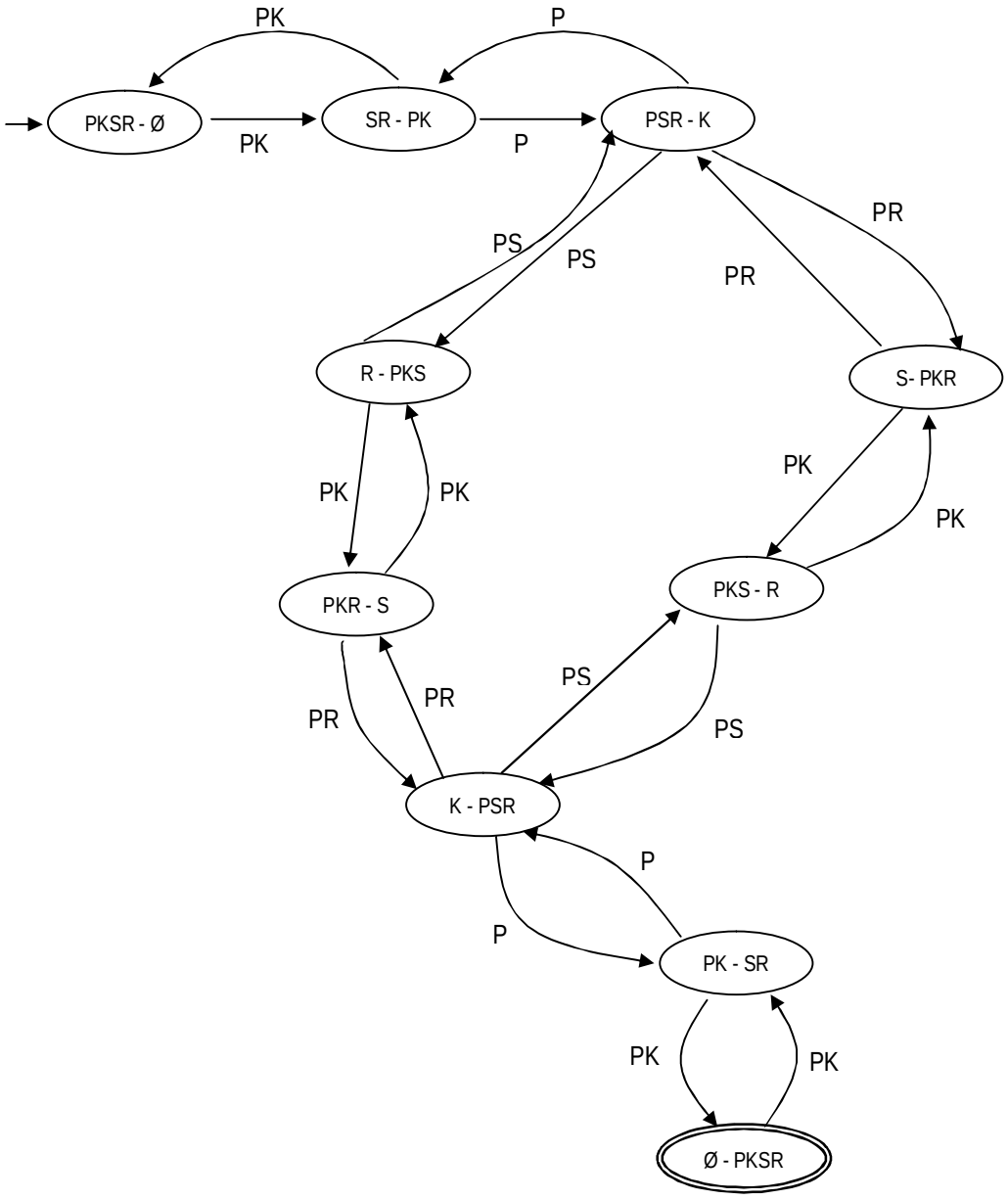
Terdapat 16 kombinasi state untu Petani (P), serigala (S), kambing (K) dan rumput (R) yaitu :

Sisi kiri	Sisi Kanan	Simbol State
PSKR	Ø	PSKR – Ø
SR	PK	SR – PK
SK	PR	SK – PR
KR	PS	KR – PS
PSR	K	PSR – K
PSK	R	PSK – R
PKR	S	PKR – S
PK	SR	PK – SR
PR	SK	PR – SK
PS	KR	PS – KR
K	PSR	K – PSR
R	PSK	R – PSK
S	PKR	S – PKR
SKR	P	SKR – P
P	SKR	P – SKR
Ø	PSKR	Ø – PSKR

Dari 16 kombinasi state tersebut, hanya 10 state yang mungkin, yaitu :

Sisi kiri	Sisi Kanan	Simbol State
PSKR	Ø	PSKR – Ø
SR	PK	SR – PK
PSR	K	PSR – K
PSK	R	PSK – R
PKR	S	PKR – S
PK	SR	PK – SR
K	PSR	K – PSR
R	PSK	R – PSK
S	PKR	S – PKR
Ø	PSKR	Ø – PSKR

Berdasarkan kemungkinan state tersebut, dapat digambarkan diagram transisi dari persoalan tersebut dengan mesin automata, sbb :



Pada diagram diatas, arti bentuk-bentuk adalah sebagai berikut :

- lingkaran merepresentasikan kedudukan (state),
- label pada lingkaran adalah nama state tersebut.
- Busur menyatakan transisi
- Label pada busur adalah masukan / input
- Lingkaran didahului dengan sebuah busur tanpa label adalah state awal
- Lingkaran ganda menyatakan state akhir (final)

Secara formal FSA didefinisikan dengan 5 tuple : M = (Q, ∑, δ, S, F), dimana :

Q : himpunan state/kedudukan

∑ : himpunan simbol input

∂ : fungsi transisi

S : State awal (initial state)

F : himpunan state akhir (Final State)

Untuk kasus petani dengan bawaanya, dapat didefinisikan FSA sebagai berikut :

Q = { PSKR–Ø, SR–PK, PSR–K, PSK–R, PKR–S, PK–SR, K–PSR, R–PSK, S–PKR, Ø–PSKR}

∑ = {P, PK, PR, PS }

δ =

	P	PK	PR	PS
PSKR – Ø	SR - PK	-	-	-
SR – PK	PSR - K	PSKR–Ø	-	-
PSR – K	SR - PK	-	S - PKR	R - PKS
PSK – R	-	S - PKR	-	K - PSR
PKR – S	-	R - PKS	K - PSR	-
PK – SR	K - PSR	Ø-PSKR		
K – PSR	PK - SR	-	PKR - S	PKS – R
R – PSK	-	PKR - S	-	PSR - K
S – PKR	-	PKS - R	PSR - K	-
Ø – PSKR	-	PK – SR	-	-

S = PSKR – Ø

F = { Ø – PSKR }

DETERMINISTIC FINITE AUTOMATA (D F A)

Apa yang dimaksud DFA?

A *deterministic finite automaton* (DFA) M = (Q, ∑, δ, S, F), dimana :

Q : himpunan state/kedudukan

∑ : himpunan simbol input

∂ : fungsi transisi, dimana ∂ ∈ Q x ∑ → Q

S : State awal (initial state)

F : himpunan state akhir (Final State)

Sebagai contoh mesin otomata berikut

$$B = (\{q_0, q_1, q_2\}, \{0, 1\}, \delta_B, q_0, \{q_2\})$$

dimana

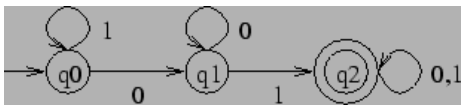
$$\delta_B = \{((q_0, 0), q_1), ((q_0, 1), q_0), ((q_1, 0), q_1), ((q_1, 1), q_2), ((q_2, 0), q_2), ((q_2, 1), q_2)\}$$

FSA tersebut dikatakan DFA, jika selalu memenuhi tabel transisi berikut :

	0	1
→ q_0	q_1	q_0
q_1	q_1	q_2
$*q_2$	q_2	q_2

The table represents the function δ, i.e. to find the value of $\delta(q, x)$ we have to look at the row labelled q and the column labelled x. The initial state is marked by an $\rightarrow$ and all final states are marked by $*$.

Yet another, optically more inspiring, alternative are transition diagrams:



There is an arrow into the initial state and all final states are marked by double rings. If $\delta(q, x) = q'$ then there is an arrow from state q to q' which is labelled x.

We write Σ^* for the set of words (i.e. sequences) over the alphabet Σ. This includes the empty word which is written ϵ. I.e.

$$\{0,1\}^* = \{\epsilon, 0, 1, 00, 01, 10, 11, 000, \ldots\}$$

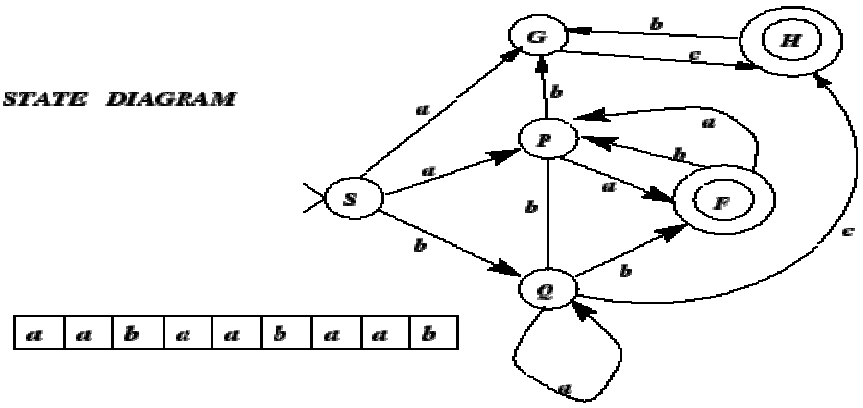
NON-DETERMINISTIC FINITE AUTOMATA

A **non-deterministic finite automaton** is a 5-tuple $M = (Q, \Sigma, \delta, q_0, F)$ where

Q	a finite set of states
Σ	an alphabet
$q_0 \in Q$	the initial state
$F \subseteq Q$	a set of final states
δ	the transition function, $\delta : Q \times \{\Sigma \cup \epsilon\} \rightarrow \mathcal{P}(Q)$.

(ϵ is the empty string; $\mathcal{P}(Q)$ is the set of all subsets of Q.)

STATE DIAGRAM



a	a	b	a	a	b	a	a	b

A **configuration** of a finite automaton $(Q, \Sigma, \delta, q_0, F)$ is an element of $Q \times \Sigma^*$.

A configuration (q, w) **yields** in one step configuration (q',w') if there is $u \in \Sigma^*$ such that $w = uw'$ and $q' \in \delta(q, u)$.

$$(q, w) \mapsto_M (q', w')$$

$\mapsto_M^*$ is the reflexive, transitive closure of $\mapsto_M$.

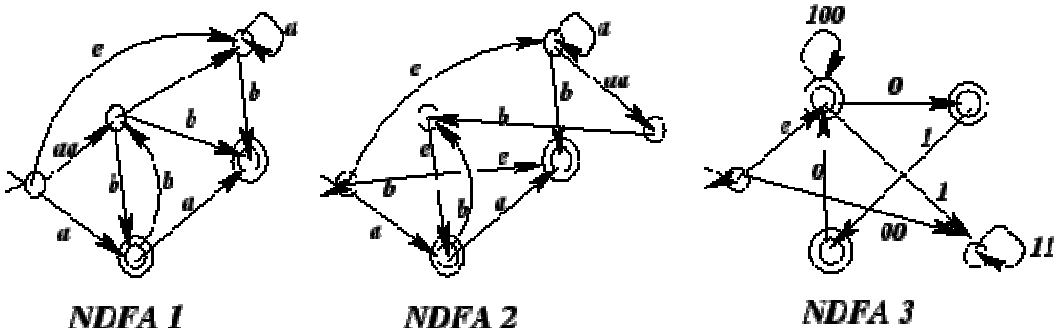
A string $w \in \Sigma$ is **accepted** by *M* if there is a path labeled *w* from the initial state q_0 to a final state, or equivalently

$$\exists q \in F \text{ such that } (q_0, w) \mapsto_M^* (q, e).$$

$$L(M) = \{w : w \text{ is accepted by } M\}.$$

Problem :
Which of the following strings are accepted by these non-deterministic finite automata?

aa, *aba*, *abb* ,*ab*, *abab* (NDFA 1); *ba*, *ab*, *bb*, *b*, *bba* (NDFA 2); 00, 01001, 10010, 000, 0000 (NDFA 3).

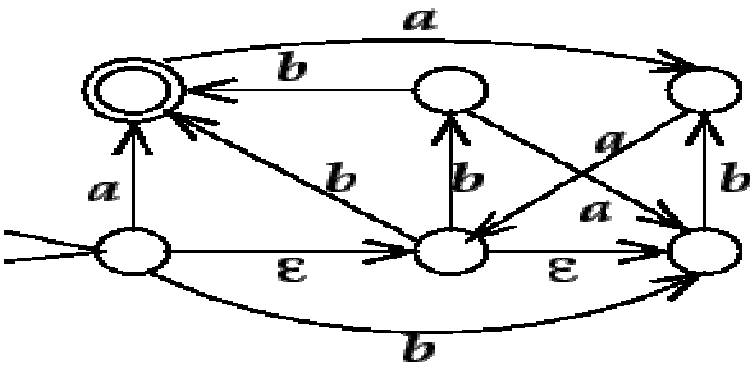


Find strings different from the above which are accepted (not accepted) by the corresponding automata.

Problem 2.2 Draw state diagrams for non-deterministic finite automata accepting these languages:

(a) $(ab)^*(ba)^* \cup aa^*$;
(b) $((ab \cup aab)^*a^*)^*$;
(c) $((a^*b^*a^*)^*b)^*$;
(d) $(ba \cup b)^* \cup (bb \cup a)^*$.

Problem 2.3 Which of the following strings are accepted by nondeterministic finite automata: *a*; *b*; *a b*; *bb*; *b a b b*?


a
b
a
b
b
a
b
b
ε
ε