

INTELLIGENT DECISION SUPPORT SYSTEM DALAM MENDETEKSI BEHAVIOUR SIRKUIT LOGIKA

Wiwini Suwarningsih

Pusat Penelitian Informatika LIPI

Jl. Sangkuriang No.21/154D (komplek LIPI) Cisu Bandung 40135, Indonesia

e-mail: wiwin@informatika.lipi.go.id

ABSTRAKSI

Dalam tulisan ini akan diuraikan mengenai intelligent decision support system (IDSS) yang akan digunakan sebagai alat bantu dalam mendeteksi finite automata behaviour. Finite automata behaviour yang akan diteliti yaitu sirkuit logika yang merepresentasikan AND-gate (\cdot), OR-gate ($+$) dan inverter (\sim). Dan sebagai masukan sirkuit logika tersebut adalah digit 1 (satu) dan 0 (nol).

Langkah awal penelitian ini adalah menganalisa sirkuit logika dengan menggunakan metoda finite automata, hasil analisa akan menghasilkan formula untuk sirkuit logika tersebut. Kemudian IDSS akan mendapatkan masukan dari rule atau formula tersebut untuk menghasilkan keluaran berupa alternatif keputusan yang sifatnya intelligent.

Hasil akhir dari penelitian ini dapat memberikan gambaran alternative keputusan, untuk sirkuit logika tersebut sehingga dapat digunakan untuk analisa dan bahan perbandingan untuk finite automata tingkah laku lainnya.

Kata kunci: Intelligent decision support system, finite state automata behavioral

1. PENDAHULUAN

Finite automata tingkah laku (*finite automaton behaviour*) merupakan bentuk aplikasi dari finite automata yang merepresentasikan suatu tingkah laku atau kebiasaan pola kerja dari suatu system atau sub system.

Finite automata membantu dalam menganalisa tingkah laku dengan cara mengidentifikasi setiap 'move' dalam suatu notasi yang telah digunakan pada finite automata. Karena terkadang dengan bentuk yang lebih sederhana misalnya dalam bentuk diagram transisi, suatu masalah dapat diselesaikan dengan cepat dan tepat.

Dalam tulisan ini finite automata tingkah laku yang akan diteliti adalah finite automata sirkuit logika yang akan dianalisa dengan bantuan *intelligent decision support system*. Dimana ini merupakan variasi dari perancangan perangkat lunak dari masalah yang sederhana dengan konversi secara otomatis dalam notasi ekspresi regular untuk menjadi lebih efisien dalam implementasi dari finite automata konspondensi.

Makalah ini akan membahas perancangan perangkat lunak IDSS dalam membantu analisa finite automata tingkah laku.

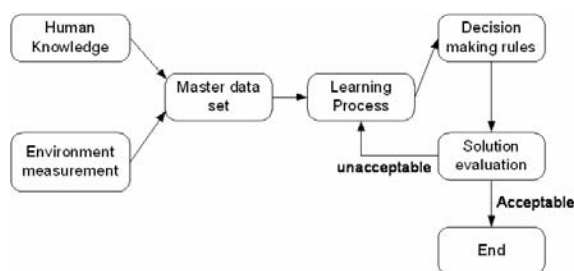
2. TEORI PENUNJANG

2.1 Intelligent Decision Support System (IDSS)

IDSS merupakan sistem pendukung keputusan yang menambah komponen berupa basis pengetahuan dengan tujuan untuk membuat DSS menjadi pintar (*intelligent*) [2]. Berbeda dengan sistem pakar, IDSS tetap menghasilkan alternatif solusi yang dijadikan sebagai alat bantu pengambilan keputusan dimana hasil akhir bukan

merupakan hasil yang final yang tidak dapat dipertimbangkan kembali. Sedangkan sistem pakar hanya sebuah proses memindahkan kepakaran seseorang pada sebuah program komputer dengan keluaran berupa hasil akhir yang pasti.

Pada gambar 1 dapat dilihat diagram alir IDSS dalam menentukan solusi keputusan dengan masukan berupa pengetahuan manusia dan pengukuran lingkungan yang dihimpun menjadi data master atau data utama. Data tersebut akan mengalami proses pembelajaran sebagai dasar pembuatan rules atau aturan pembuatan keputusan. Kemudian dilakukan evaluasi solusi terhadap rules tersebut untuk dijadikan sebagai alternatif keputusan.



Gambar 1. Diagram Alir IDSS [5]

Bila solusi tersebut tidak diterima maka proses selanjutnya akan mengalami proses pembelajaran sampai mengulang kembali untuk menghasilkan solusi yang diterima. Bila solusi diterima maka proses selesai.

2.2 Finite Automata (FA)

Finite automata adalah model matematika suatu system dengan input dan output yang diskret.

Dalam finite automata, perlu diketahui tentang finite state system (FSS). Contoh FSS banyak dijumpai dalam ilmu computer, salah satunya adalah switching circuit sebagai unit control computer. Switching circuit tersebut disusun oleh gerbang-gerbang logika yang berisi bilangan 0 atau 1. Contoh FSS yang lain, adalah mekanisme kendali pada elevator.

FA merupakan salah satu pengenalan bahasa (*recognizer*) yang paling sederhana. FA juga merupakan mesin yang dapat membuktikan apakah suatu string termasuk regular language atau tidak.

Suatu FA beroperasi dengan membuat urutan 'move'. Suatu 'move' dibatasi oleh state yang aktif dari finite state control dan input symbol di scan oleh input head. 'Move' berisi control perubahan state dan input head berpindah satu konteks (elemen) ke kanan.

Finite automata[1] terdiri dari dua jenis yaitu deterministic finite automata dan nondeterministic finite automata. Berikut ini bentuk formula dari kedua finiter tersebut:

a. Deterministic Finite Automata (DFA)

DFA adalah bentuk FA yang 'move' nya hanya satu arah saja. Dan state akhir yang dituju hanya satu.

$$DFA = (Q, \Sigma, \delta, q_0, F)$$

dimana:

Q = himpunan terbatas 'state'

Σ = himpunan terbatas input symbol

δ = mapping $Q \times \Sigma$ terhadap $p(Q)$ atau fungsi transisi state

$q_0 = q_0 \in Q$ adalah state awal (dalam FSC)

F = $F \subseteq Q$ state akhir

b. NonDeterministic Finite Automata (NFA)

DFA adalah bentuk FA yang 'move' nya lebih dari satu arah. Dan state akhir yang dituju dapat lebih dari satu

$$DFA = (Q, \Sigma, \delta, q_0, F)$$

dimana:

Q = himpunan terbatas 'state'

Σ = himpunan terbatas input symbol

δ = mapping $Q \times \Sigma$ terhadap $p(Q)$ atau fungsi transisi state

$q_0 = q_0 \in Q$ adalah state awal (dalam FSC)

F = $F \subseteq Q$ himpunan state akhir

2.3 Diagram Transisi pada Finite Automata

Diagram transisi adalah graph berarah, dimana vertexnya merupakan state dari FA. Jika terdapat transisi dari state q ke state p dengan input a, maka busur dari q ke p adalah a. FA menerima string x, jika urutan sekumpulan transisi ke symbol x yang dimulai dari state awal ke state akhir. Simbol-simbol yang digunakan dalam diagram transisi adalah:

Tabel 1. Tabel symbol diagram transisi

Symbol	Arti
	State awal
	State antara
	State Akhir
	Arah transisi

3. HASIL PENELITIAN

3.1 Pendefinisian Human Knowledge dan Environment Measurement

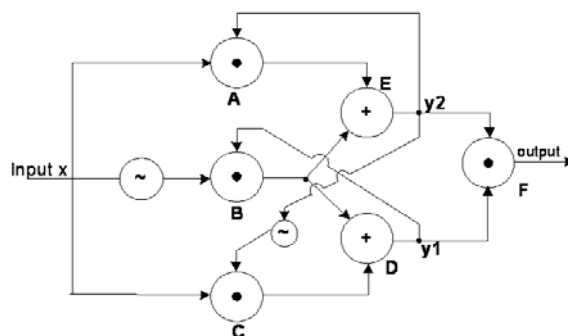
Dua tahap ini merupakan tahap awal yang akan dijadikan sebagai masukan pada proses master data set.

3.1.1 Human Knowledge

Human Knowledge (pengetahuan manusia) adalah tahap untuk mendefinisikan pemahaman terhadap permasalahan yang akan diteliti pada tulisan ini. Pemahaman tersebut berupa analisa terhadap pola kerja sirkuit logika (lihat gambar 2) yang akan dijadikan sebagai bahan penelitian.

Pola kerja sirkuit logika tersebut adalah sebagai berikut:

- Input x dapat berupa digit biner 0 dan 1, kemudian akan masuk gerbang A, B, C, D, E, F yang merupakan gerbang AND(+) dan gerbang OR(•).
- Nilai x ini akan bergantung pula terhadap kombinasi digit yang diberikan oleh y1 dan y2 (lihat table.2 yang merupakan tahap pemasukan digit kombinasi yang diberikan oleh y1 dan y2).
- Output dihasilkan dari hasil akhir setelah nilai x dan kombinasi y1 dan y2 diproses secara circular, sehingga gerbang D dan gerbang E akan menghasilkan nilai yang kemudian akan masuk gerbang F yang merupakan output dari proses sirkuit logika tersebut



Gambar 2. Sirkuit Logika

3.1.2 Environment Measurement

Environment Measurement (lingkungan pengukuran) disini lebih menitikberatkan pada aturan dari finite automata yang akan digunakan. Pada penelitian ini formula yang digunakan adalah Deterministik Finite Automata.

3.2 Master Data Set

Tahap ini merupakan inisialisasi digit kombinasi dari y1 dan y2 (lihat gambar 2 posisi y1 dan y2). Nilai y1 akan mempengaruhi gerbang A dan C, sedangkan y2 akan mempengaruhi gerbang B. Sedangkan gerbang D merupakan hasil dari A + B dan Gerbang E dihasilkan dari B + C.

Gerbang dan Notasi yang digunakan:

OR (+), AND (•), inverter (~)

Jika $y1 = 0, y2 = 0$
maka $A = 0 \bullet X = 0$
 $B = 0 \bullet \sim X = 0$
 $C = \sim 0 \bullet X = X$
 $D = A + B = 0 + 0 = 0$
 $E = B + C = 0 + X = X$

Jika $y1 = 0, y2 = 1$
maka $A = 0 \bullet X = 0$
 $B = 1 \bullet \sim X = \sim X$
 $C = \sim 0 \bullet X = X$
 $D = A + B = 0 + \sim X = \sim X$
 $E = B + C = \sim X + X = 1$

Jika $y1 = 1, y2 = 0$
maka $A = 1 \bullet X = X$
 $B = 0 \bullet \sim X = 0$
 $C = \sim 1 \bullet X = 0$
 $D = A + B = X + 0 = X$
 $E = B + C = 0 + 0 = 0$

Jika $y1 = 1, y2 = 1$
maka $A = 1 \bullet X = X$
 $B = 1 \bullet \sim X = \sim X$
 $C = \sim 1 \bullet X = 0$
 $D = A + B = X + \sim X = 1$
 $E = B + C = \sim X + 0 = \sim X$

Hasil paparan diatas secara ringkas dapat dilihat pada tabel 2. Untuk hasil yang bernilai X ini berarti bahwa nilai tersebut tergantung pada nilai input X.

Tabel 2. Kombinasi digit y1 dan y2

y1	y2	A	B	C	D	E
0	0	0	0	X	0	X
0	1	0	~X	X	~X	1
1	0	X	0	0	X	0
1	1	X	~X	0	1	~X

Kombinasi y1y2 yang terdiri dari pasangan digit biner selanjutnya dinotasikan menjadi himpunan terbatas 'state', sehingga menjadi:

Tabel 3. Notasi state untuk y1y2

y1y2	state
00	q0
01	q1
10	q2
11	q3

Penentuan atau pemberian nama untuk state dapat menggunakan symbol apa saja. Tetapi dalam penelitian ini penulis menggunakan symbol dengan q0 sampai dengan q3 dengan alasan untuk memudahkan melihat 'move' yang terjadi dari satu state ke state yang lain dan karena dalam aturan formula finite state automata state awal dinotasikan dengan q0 maka untuk state selanjutnya juga menggunakan notasi qn.

3.3 Learning Process

Setelah mendapat master data maka selanjutnya adalah proses pembelajaran sampai menghasilkan formula dan diagram transisi yang sesuai dengan tingkah laku sirkuit logika tersebut.

Tahap 1: Memberi nilai Input X

Dari tabel 2 tahap selanjutnya adalah memberi nilai Input X dengan digit 0 dan 1. Bila kita memberi nilai input X maka sudut pandang kita pada posisi kanan (lihat gambar 2) yaitu nilai y1 berasal dari gerbang D dan y2 berasal dari gerbang E. Maka tabel 2 akan berubah seperti tabel 4.

Tabel 4. Memasukan Nilai Input X

state y1y2	Gerbang					X=0		X=1	
	A	B	C	D	E	y1	y2	y1	y2
q0	0	0	X	0	X	0	0	0	1
q1	0	~X	X	~X	1	1	1	0	1
q2	X	0	0	X	0	0	0	1	0
q3	X	~X	0	1	~X	1	1	1	0

Tahap 2: Membuat mapping state dan Input X

Dari proses memasukan nilai input X, selanjutnya adalah menyederhanakan tabel 4 menjadi suatu bentuk mapping atau pemetaan dalam bentuk tabel. Dengan melihat ketentuan state pada tabel 3, maka tabel mappingnya dapat dilihat pada tabel 5.

Tabel 5. Mapping State dan Input X

state y1y2	X=0		X=1	
	y1	y2	y1	y2
q0	0	0	0	1
q1	1	1	0	1
q2	0	0	1	0
q3	1	1	1	0

Tahap 3: Menyederhanakan mapping

Menyederhanakan bentuk mapping dengan mengelompokkan state dan kombinasi y1y2. Bentuk sederhana dari tabel 5 dapat dilihat pada tabel 6

Tabel 6. Penyederhanaan mapping input X

	0	1
s		
t	q ₀	00
a	q ₁	11
t	q ₂	00
e	q ₃	11

Tahap 4: Membuat Mapping Finite State Automata

Hasil akhir tabel mapping dengan menggunakan notasi state sesuai dengan ketentuan dari tabel 3, maka mapping dari sirkuit logika adalah sebagai berikut (tabel 7).

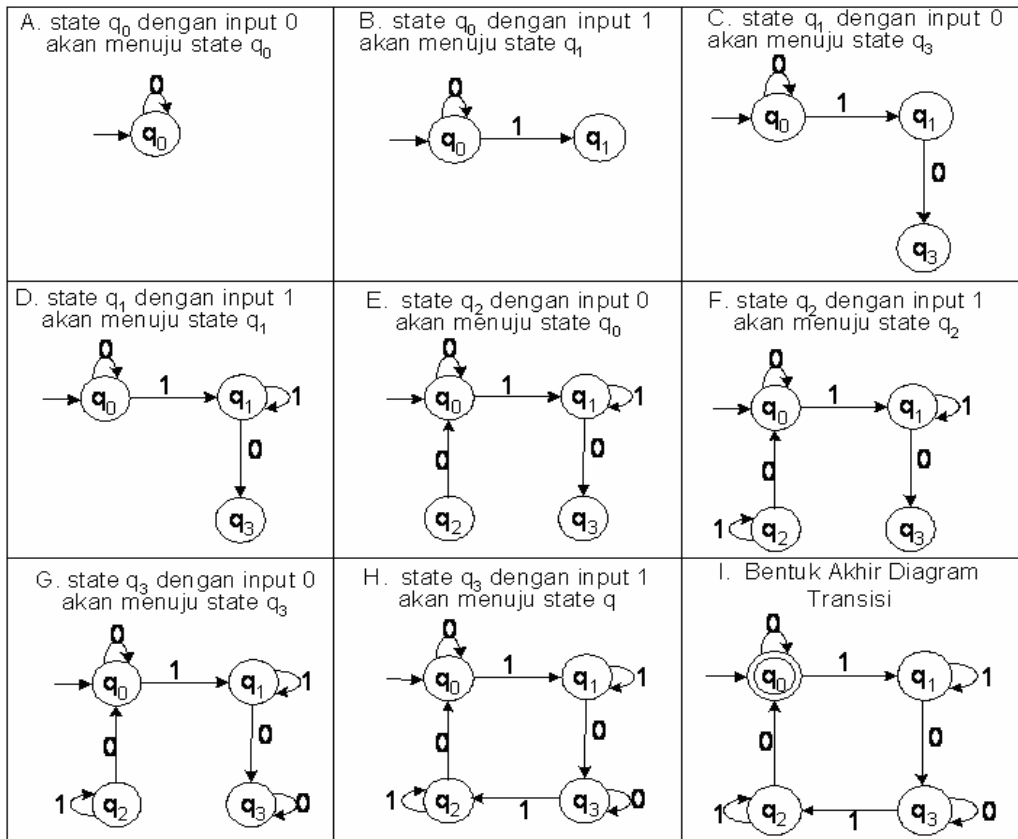
Tabel 7. Mapping FSA dari sirkuit Logika

	0	1
q ₀	q ₀	q ₁
q ₁	q ₃	q ₁
q ₂	q ₀	q ₂
q ₃	q ₃	q ₂

Tahap 5. Membuat Diagram Transisi

Tahap selanjutnya adalah mengubah mapping DFA dalam bentuk tabel ke dalam bentuk diagram transisi. Diagram transisi ini untuk memudahkan penentuan state awal dan state akhir serta perubahan yang terjadi dari satu state ke state yang lain dengan kombinasi input (digit 0 dan 1) yang dibaca oleh DFA. Uraian penjelasan perubahan dari mapping tabel ke dalam bentuk diagram transisi dapat dilihat pada gambar 3.A sampai dengan 3.H.

Hasil Akhir penggabungan tahap A sampai dengan H dapat dilihat pada gambar 3.I. Dimana State awal dan state akhir terdapat di state q₀.



Gambar 3. Tahap Pembentukan Diagram Transisi Sirkuit Logika

3.4 Decision Making Rules

Tahap ini merupakan tahap pembuatan aturan untuk FA tingkah laku yang dihasilkan dari tahap learning process yang berasal dari diagram transisi pada gambar 3.

Formula standar dari FSA deterministic adalah:

$$FA = (Q, \Sigma, \delta, q_0, F)$$

Formula FSA dari sirkuit logika adalah sebagai berikut:

$$FSA-SL = (\{q_0, q_1, q_2, q_3\}; \{0,1\}, \delta, q_0, q_0) \dots (3)$$

dimana δ :

$$\delta(q_0, 0) = \{q_0\}$$

$$\delta(q_0, 1) = \{q_1\}$$

$$\delta(q_1, 0) = \{q_3\}$$

$$\delta(q_1, 1) = \{q_1\}$$

$$\delta(q_2, 0) = \{q_0\}$$

$$\begin{aligned} \delta(q_2, 1) &= \{q_2\} \\ \delta(q_3, 0) &= \{q_3\} \\ \delta(q_3, 1) &= \{q_2\} \end{aligned}$$

3.5 Solution Evaluation

Evaluasi solusi ini dilakukan untuk menguji apakah rule yang dihasilkan telah memenuhi kondisi dari finite automata atau tidak memenuhi. Cara evaluasinya adalah dengan memasukan beberapa contoh string (word) ke dalam formula. Bila string tersebut dipetakan terus menerus sesuai mapping dari formula (3) dan mencapai state akhir dengan string akhir empty, maka solusi tersebut *accepted* (diterima) oleh formula. Jika mencapai state akhir dan string tidak empty maka solusi tersebut *not accepted* (ditolak).

Bentuk kondisi awal untuk mapping: (q_0, w)

Bentuk kondisi akhir yang harus dicapai: (q_0, e)

q_0 = state awal dan state akhir

w = string

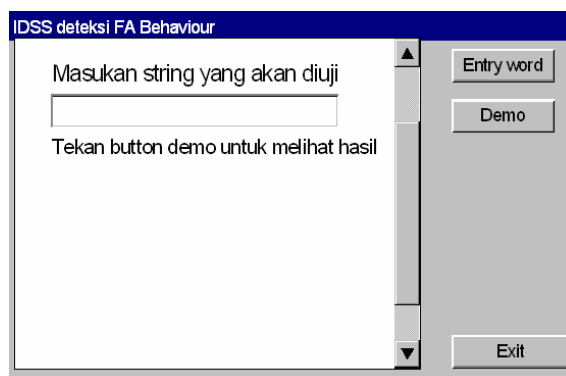
e = empty atau kosong

Tabel 8. Evaluasi Solusi

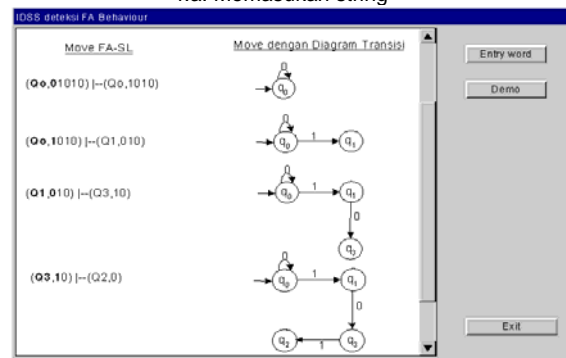
String (word)	Move mapping
0011010	$(q_0, 0011010) \rightarrow (q_0, 011010)$ $\rightarrow (q_0, 11010)$ $\rightarrow (q_1, 1010)$ $\rightarrow (q_1, 010)$ $\rightarrow (q_3, 10)$ $\rightarrow (q_2, 0)$ $\rightarrow (q_0, e)$ Karena mencapai state akhi yaitu q_0 dan string akhir empty(e), maka solusi dengan string = 0011010 ini diterima
010110	$(q_0, 010110) \rightarrow (q_0, 10110)$ $\rightarrow (q_1, 0110)$ $\rightarrow (q_3, 110)$ $\rightarrow (q_2, 10)$ $\rightarrow (q_2, 0)$ $\rightarrow (q_0, e)$ Karena mencapai state akhi yaitu q_0 dan string akhir empty(e), maka solusi dengan string = 010110 ini diterima
1100110	$(q_0, 1100110) \rightarrow (q_1, 100110)$ $\rightarrow (q_1, 00110)$ $\rightarrow (q_3, 0110)$ $\rightarrow (q_3, 110)$ $\rightarrow (q_2, 10)$ $\rightarrow (q_2, 0)$ $\rightarrow (q_0, e)$ Karena mencapai state akhi yaitu q_0 dan string akhir empty(e), maka solusi dengan string = 1100110 ini diterima
1110010	$(q_0, 1110010) \rightarrow (q_1, 110010)$ $\rightarrow (q_1, 10010)$ $\rightarrow (q_1, 0010)$ $\rightarrow (q_3, 010)$ $\rightarrow (q_3, 10)$ $\rightarrow (q_2, 0)$ $\rightarrow (q_0, e)$ Karena mencapai state akhi yaitu q_0 dan string akhir empty(e), maka solusi dengan string = 1100110 ini diterima

Dari empat solusi diatas hasil akhirnya mencapai state akhir yaitu q_0 dan string dalam kondisi empty. Karena solusi di terima oleh formula, maka proses IDSS ini selesai. Dimana formula(3) merupakan formula yang berlaku untuk sirkuit logika yang menghasilkan lebih dari satu solusi yang merupakan ciri dari IDSS.

Evaluasi dengan menggunakan program dapat dilihat pada gambar4. Gambar 4.a. awal proses dengan memasukan string yang akan diuji. sedangkan gambar 4.b. proses moving baik dengan notasi move-fa dan notasi dengan diagram transisi.



4.a. Memasukan string



4.b. Proses moving

Gambar 4. Tampilan program IDSS deteksi FA behaviour

4. KESIMPULAN

Dari hasil penelitian IDSS untuk mendeteksi finite state automata behaviour, maka dapat disimpulkan bahwa:

- Dari rule yang dibuat dapat mengasilkan solusi string yang dapat diterima oleh formula tersebut. sehingga stardar daris uatu dss dapat dipenuhi.
- Finte state automata behaviour merupakan bentuk kelakuan finite automata yang dapat membantu analisa input dan otput dari suatu sirkuit logika.

PUSTAKA

- [1] Hopcroft, Ullman, "Introduction to Automata Theory, languages and computation", Addison Wesley Publishing Company, California, 1979.

- [2] Dipankar, “*An Intelligent Decision Support System for Intrusion Detection and Response*”, Memphis, 2001, url: issrl.cs.memphis.edu/papers/issrl/2001/mm-sub.pdf
- [3] Mauro Dell'Orco, “*Intelligent Decision Support Tools For Optimal Planning of Rail Track Maintenance*”, 2003, url: <http://www.iasi.cnr.it/ewgt/13conference>
- [4] Aho, Ullman, “*The Theory of parsing and Translation*”, McGraw-Hill International, New York, 1979.
- [5] Cohen, Daniel, “*Introduction to automata theory*”, Addison Wesley Publishing Company, California, 1986.
- [6] Josh Bongard, “Active Coevolutionary Learning of Deterministic Finite Automata”, *Journal of Machine Learning Research* 6, 2005.
- [7] Chiu-Che Tseng, Piotr J. Gmytrasiewicz, 2003, *Real Time Decision Support System for Portfolio Management*, Department of Computer Science Engineering, University of Texas at Arlington 300 Nedderman Hall Arlington, TX. 76011.
- [8] Cummings, 1999, *Automation Bias in Intelligent Time Critical Decision Support Systems*, Massachusetts Institute of Technology, Cambridge, MA 02319.
- [9] Mauro Dell'Orco, 2003, *Intelligent Decision Support Tools For Optimal Planning of Rail Track Maintenance*, Italy. url: <http://www.iasi.cnr.it/ewgt/13conference>.
- [10] Power, 2002, *Decision support systems: concepts and resources for managers*. Quorum Books.
- [11] Sprague, R. H., 1982, *Building Effective Decision Support System*, Grolier, New Jersey.
- [12] M. P. Singh and M. N. Huhns, 2005. *Service-Oriented Computing*. Wiley.