Model of Fail-Safe Self-Modifying Finite Automata

Vencislav G. Trifonov[1] and Ivailo I. Atanasov[2]

Abstract – **The paper presents a model of fail-safe finite state machine with self-modifying functions. Self-modifying functions are fail-safe and change automata states, ordered by a safety set of conditions.**

Keywords – **Fail-safe finite machines, Self-modifying automata, Discrete Event Systems**

I. Introduction

Finite automata (FA, FM) are one of methods for mathematical description of discrete event systems. Fail-safe finite machines (FSFM) are a subclass of main automata class. [1-3] This class includes two types of function:

— normal (conventional, work) functions;

— fail-safe functions.

Fail-safe functions used to achieve a high-level of safety and security it case of hazardous fault in conventional operations.

Definition 1: Finite machine (FM) is a 6-tulpe

$$FM = \langle X, Y, Q, Int(\bullet), Out(\bullet)\rangle \qquad (1)$$

where $X : \{x_1, ..., x_n\}$ is input alphabet of FM, $Y : \{y_1, ..., y_n\}$ is output alphabet of FM. Q" $\{q_1, ..., q_n\}$ is internal set of automata states. $Int(\bullet)$ is automata states function and $Out(\bullet)$ is automata output function.

Formal description of automata behavior is given:

$$\left| \begin{array}{l} Q(t+1) = Int(Q(t), X(t)) \\ Y(t+1) = Out(Q(t), X(t)) \end{array} \right. \qquad (2)$$

Fig. 1 presents an abstract FM.

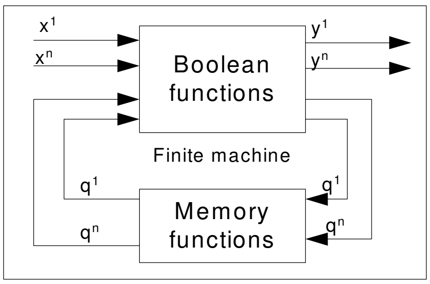


Fig. 1. Finite state machine

Definition 2: **Fail-safe behavior** is each automata transaction that changes current state to safe state after hazardous fault detection [1].

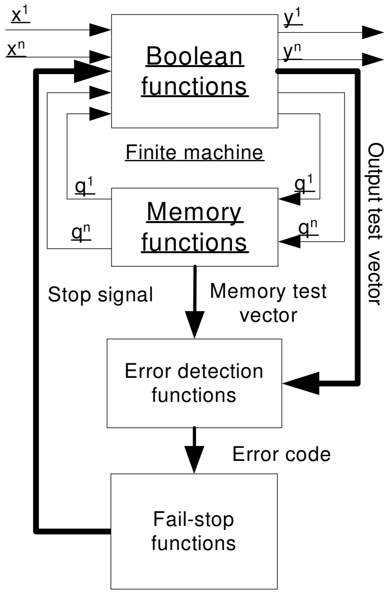


Fig. 2. Fail-safe finite machine

Definition 3: **Hazardous failure** is a failure that changes current state to a dangerous state [2].

Definition 4: **Dangerous input sequence** is each sequence of input alphabet that puts automata to a dangerous state [2].

For successfully recognize a kind of automata states should by defined criteria for fail-safe behavior and criteria for safety fault classification, described in detail in [1-3].

To achieve fail-safe behavior FM abstract structure on Fig. 1 extends to FSFM structure Fig. 2

II. States Ordering by Fail-Safe Degrees

The paper presents another point of view about the automata states. For most of fail-safe finite machines have possibility to define a linear ordered relation between their states [4,5].

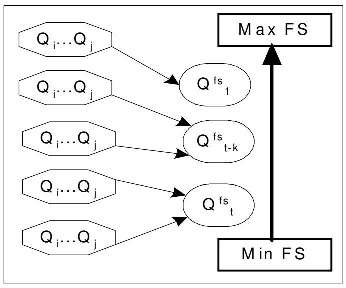


Fig. 3. Fail-safe state order

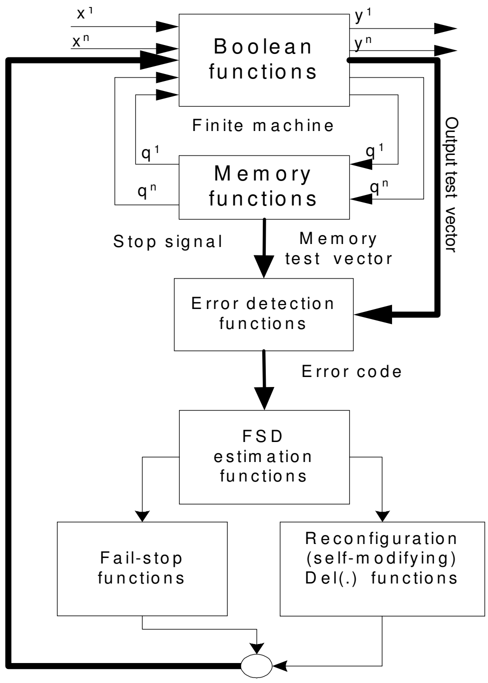
[1]Vencislav G. Trifonov is with Telecom Department at Technical University of Sofia, “Kl. Ohridsky” Blvd. 8, 1756 Sofia, Bulgaria, e-mail: vgt@vmei.acad.bg

[2]Ivailo I. Atanasov is with Telecom Department at Technical University of Sofia, “Kl. Ohridsky” Blvd. 8, 1756 Sofia, Bulgaria, e-mail: iia@vmei.acad.bg

Each set of automata states should by presents as:

$$Class(Q) = \sum_{1}^{j} Subc; ass_j(Q_i) \tag{3}$$

where: $Class(Q)$ is a set of all FM states and $Sublass_j(Q_i)$ is a set of FM states with equivalent fail-safe degree, j : $\{1, ...n\}$ where $n = Gard\{Subclass_i\}$ and $i : \{1, ..., m\}$ is number of all internal states with same fail-safe degree [4,5]. The automata states should by present as follow (Fig. 3):



Let FM A is a reverse counter with input alphabet X : $\{a, b, c\}$; output alphabet Y : $\{0, 1\}$; and set of internal states: $Q\{1, 2, ..., N\}$. FM model is presents in Table 1.

Table 1. Reverse counter

	a	**b**	**c**
1	*2*	*1*	*1*
2	*3*	2	*1*
N	*N*	*N*	*N-1*

FM recognizes each valid input sequence with a set of final functional states

$$Final\_set(Q_i) = \{2, 4, 6, 8, ..., n/2\}.$$

A sample distribution to subclasses is:

$$\left| \begin{array}{l} Subclass\ (1) : \{2, 4, 6\}; \\ Subclass\ (2) : \{8, 14, 18, 34, 98\}; \\ \dots\dots\dots\dots\dots\dots\dots \\ Subclass\ (n) : \{q_i, ..., q_k\}. \end{array} \right. \tag{4}$$

For each subclass defines fail-safe degree (FSD). A fail-safe relation orders FSD as follow [4]:

$$FSD_{\min} < FSD_1 < \cdots < FSD_n < FSD_{\max}. \tag{5}$$

For Subclass set defines a map function to FSD set:

$$\left| \begin{array}{l} Subclass\ (1) \in FSD_1; \\ Subclass\ (2) \in FSD_2; \\ \dots\dots\dots\dots\dots\dots \\ Subclass\ (n) \in FSD_k. \end{array} \right. \tag{6}$$

III. Fail-Safe Reconfiguration Functions

Structure of a Fail-Safe Automata with State Restriction is present on Fig. 4. To realize new self-modifying function **Del()** should by defines it as follows:

$$\begin{array}{l} D(g) : Class(Q_{old}) \to Class(Q_{new}); \\ Gard(Class(Q_{new})) > Gard((Class(Q_{old})). \end{array} \tag{7}$$

Definition 5. Del(.) function is fail-safe function if:

1. $Class(Q_{new})$ is FSD ordered set;
2. $D(t) : Class(Q_{old}(t - 1)) \to Class(Q_{new}(t))$;
3. Each possible input sequence FSFM maps as follow:
   – if $X(i)$ is valid input word then
   $FSFM : \{x_1, ..., x_n\} \to Final\_set(Class_{new}(Q))$;
   – else:
   $FSFM : \{x_1, ..., x_n\} \to Final\_stop\_set(Class_{new}(Q))$
4. After **Del(.)** contraction, subtraction subclasses are total inaccessible.

IV. Conclusion

The paper presents a model for self-modifying fail-safe finite state machine and definition for fail-safe self-modifying function.

Self-modifying property based on automata states contraction, witch subtract dangerous states for each detected hazardous fault.

References

[1] H. Hristov, V. Trifonov , "New problems in fail-safe automata", II Международная научно-техническа конференция "Актуалные проблеми развития железодорожного транспорта" Россия, Москва, 24-25 сентября 1996 г. стр. 125.

[2] Trifonov V, "Algoritmus fur Syntese des ungefarlichen Endautomat bei geanderte Verteilung des Eingangraums nach der Absage" TRANSCOM'97, University of Zilina, Slowak Republik, June 25-26, 1997 Volume 2, стр. 73.

[3] Hristo Hristov, Trifonov Vencislav, "A method for fail-safe degradation of final state machines", The 4 th INTERNATIONAL SCIENTIFIC CONFERENCE OF RAILWAYS EXPERT, Yugoslawia, Vrnjacka Banja, Oktober 2-4 1997 г., стр. 328.

[4] V. Trifonov, "Method for synthesis a fail-safe automata with restricted degradation", The 5 th INTERNATIONAL SCIENTIFIC CONFERENCE OF RAILWAY EXPERTS - JUЖEL, Yugoslvia, Vrnjcka Banja, October 28-30, 1998 г. стр. 91.

[5] Joze Eyzell, Jose Cury, Exploiting Symmetry in the Synthesis of Supervisor for Discrete Event Systems, American Control Conference 1998.