MATEMATICKO-GRAFICKÁ FORMALIZÁCIA ČINNOSTI RIADIACEHO OBVODU VÝHYBKY
MATHEMATIC – GRAPHICAL FORMALIZATION OF SWITCH POINT CONTROL CIRCUIT FUNCTION

Juraj Ždánsky, Karol Rástočný

Katedra riadiacich a informačných systémov, Elektrotechnická fakulta, Žilinská univerzita, Univerzitná 1, 010 26 Žilina

Abstrakt: V článku je uvedený postup, ktorý bol navrhnutý autormi a umožňuje matematicko-grafickú formalizáciu funkčnej špecifikácie systému. Výsledkom tejto činnosti je algebraický systém - konečný automat, ktorý je zapísaný tabuľkou prechodov. Tabuľku prechodov možno následne prepísať do grafickej formy (stavový diagram) alebo do matematickej formy (prechodová a výstupná funkcia). Tento postup je vysvetlený na príklade riadiaceho obvodu výhybky.

Summary: This article describes authors designed method then enables mathematic – graphical formalization of system's functional specification. The result of this method is algebraic system – finite automata that is written in transition table. This transition table is possible to overwrite to graphic form (state diagram) or to mathematic form (transition and output function). This method is described by example of switch point control circuit.

1. ÚVOD

Pre bezpečnostne kritické systémy je dôležité, aby na báze formálnych prípadne poloformálnych metód bol v procese analýzy požiadaviek vytvorený prehľadný a zrozumiteľný model, ktorý umožní odstrániť prípadné nejasnosti alebo protirečenia v neformálnej špecifikácii a umožní preskúšať komplexnosť a bezchybnosť špecifikácie. Použitie vhodného formalizmu na opis správania riadiaceho systému (vytvorenie modelu) výrazne zefektívni prácu programátora a minimalizuje systematické chyby v programe.

2. NEFORMÁLNA ŠPECIFIKÁCIA FUNKČNÝCH POŽIADAVIEK NA RIADIACI OBVOD VÝHYBKY

Požiadavky na riadiaci obvod elektricky ovládaného prestavníka výhybky sú uvedené v norme [1]. Podľa tejto normy, riadiaci obvod prestavníka výhybky musí:

a) znemožniť vykonanie príkazu na prestavenie výhybky, ak sú splnené tieto podmienky :
   - výhybka je uzavretá v jazdnej ceste (záver výhybky alebo príslušný úsek);
   - príslušný úsek výhybky nie je voľný a nebol daný príkaz na núdzové prestavenie výhybky;
b) v ktoromkoľvek okamihu prestavovania umožniť zmenu pohybu prestavnej tyče výhybkového prestavníka za predpokladu, že sú rešpektované podmienky podľa predchádzajúceho bodu a);
c) nedovoliť prerušenie započatého pohybu prestavnej tyče výhybkového prestavníka pri strate informácie o voľnosti výhybkového úseku;
d) po každom násilnom prestavení pohyblivých častí výhybky železničným koľajovým vozidlom (rozreze) znemožniť ďalšie ústredné ovládanie prestavníka až do záznamu jej rozrezu.

Koncová poloha výhybky sa vyhodnotí, ak sú splnené tieto podmienky:

a) všetky snímače koncovej polohy pohyblivých častí výhybky hlásia požadovanú polohu;
b) nie sú hlásené obidve koncové polohy pohyblivých častí výhybky;
c) nezaznamenal sa rozrez výhybky;
d) nedáva sa príkaz na prestavenie výhybky do opačnej polohy.

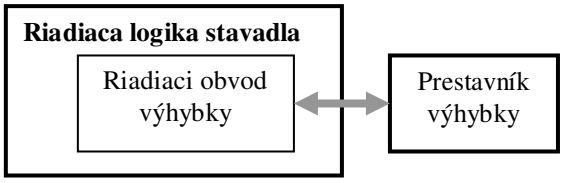
Rozrez výhybky sa vyhodnotí, ak sú splnené tieto podmienky:

a) nedáva sa príkaz na prestavenie výhybky;
b) nie je hlásená koncová poloha výhybky;
c) príslušný výhybkový úsek je obsadený.

3. FORMÁLNA ŠPECIFIKÁCIA FUNKČNÝCH POŽIADAVIEK NA RIADIACI OBVOD VÝHYBKY

3.1. Analýza vstupných a výstupných informácií

Nech riadiaci obvod výhybky je samostatný modul riadiacej logiky stavadla. Vstupné informácie pre riadiaci obvod výhybky prichádzajú od spolupracujúcich modulov riadiacej logiky stavadla a od výhybkového prestavníka (obr. 1.)



Obr.1. Rozhranie riadiacej logiky
Fig.1. Control logic interface

Nech tieto vstupné informácie (vstupné slová) majú takto priradené logické úrovne:

- Informácia o uzavretí (závere) výhybky v jazdnej ceste:
  - log. 1 – výhybka je uzavretá v jazdnej ceste;
  - log. 0 – výhybka nie je uzavretá v jazdnej ceste.

- Informácia o voľnosti úseku prislúchajúceho výhybke:
  - log. 1 – výhybkový úsek je voľný;
  - log. 0 – výhybkový úsek je obsadený.
- Príkaz na núdzové prestavovanie výhybky:
  - log. 1 – bol daný príkaz na núdzové prestavenie výhybky;
  - log. 0 – nebol daný príkaz na núdzové prestavenie výhybky.
- Informácia o plusovej (koncovej) polohe výhybky:
  - log. 1 – výhybka je v plusovej polohe;
  - log. 0 – výhybka nie je v plusovej polohe.
- Informácia o mínusovej (koncovej) polohe výhybky:
  - log. 1 – výhybka je v mínusovej polohe;
  - log. 0 – výhybka nie je v mínusovej polohe.
- Požiadavka na prestavenie výhybky do plusovej polohy:
  - log. 1 – existuje požiadavka na prestavenie výhybky do plusovej polohy;
  - log. 0 – neexistuje požiadavka na prestavenie výhybky do plusovej polohy.
- Požiadavka na prestavenie výhybky do mínusovej polohy:
  - log. 1 – existuje požiadavka na prestavenie výhybky do mínusovej polohy;
  - log. 0 – neexistuje požiadavka na prestavenie výhybky do mínusovej polohy.

Výstupné informácie (výstupné slová) riadiaceho obvodu výhybky sú určené pre spolupracujúce moduly riadiacej logiky stavadla a pre výhybkový prestavník. Nech tieto výstupné informácie majú takto priradené logické úrovne:

- Povel na prestavenie výhybky do plusovej polohy:
  - log. 1 – existuje povel na prestavenie výhybky do plusovej polohy;
  - log. 0 – neexistuje povel na prestavenie výhybky do plusovej polohy.
- Povel na prestavenie výhybky do mínusovej polohy:
  - log. 1 – existuje povel na prestavenie výhybky do mínusovej polohy;
  - log. 0 – neexistuje povel na prestavenie výhybky do mínusovej polohy.
- Informácia o rozreze výhybky:
  - log. 1 – výhybka je rozrezaná;
  - log. 0 – výhybka nie je rozrezaná.

Predpokladajme, že vytvorený stavový diagram bude opisovať synchrónny sekvenčný obvod (pre riadenie bezpečnostne kritických procesov je synchrónny sekvenčný obvod výhodnejší ako asynchrónny sekvenčný obvod [3]).

3.2. Postup pri formalizácii funkčných požiadaviek

Postup pri formalizácii sa dá zhrnúť do nasledujúcich bodov:

- prepis požiadaviek neformálnej špecifikácie do tabuľkovej formy; kontrola tabuľky na úplnosť a čiastočné doplnenie špecifikácie;
- označenie vstupných slov, výstupných slov a stavov systému;
- vytvorenie matice identifikátorov vstupných slov;
- vytvorenie prvotnej prechodovej tabuľky;
- upresňovanie špecifikácie (zlučovanie stavov, odstránenie nezlučiteľných stavov, doplnenie prechodovej tabuľky o nedefinované prechody);
- minimalizácia prechodovej tabuľky;
- prepis prechodovej tabuľky do grafickej alebo matematickej formy.

Pri prepise požiadaviek neformálnej špecifikácie do tabuľkovej formy (tab. 1) ide o sekvenčný zápis, preto každá zmena vstupnej informácie musí byť zaznamenaná v novom riadku tabuľky. Do stĺpca pomenovaného Výstupy sa zapisujú výstupné informácie, ktoré zodpovedajú určitému stavu systému a vstupným informáciám systému. V stĺpci pomenovanom Stav sa uvádzajú k príslušným vstupným a výstupným informáciám niektoré významné stavy systému, ktoré sú buď už pomenované v špecifikácii, alebo si pomenovanie zvolí autor tabuľky. Za významný stav treba považovať predovšetkým počiatočný stav a tiež všetky stavy systému, z ktorých existuje prechod do viac ako jedného nasledujúceho stavu, čo však v tejto fáze nemusí byť jednoznačné. Rozhodnutie o tom, či stav je alebo nie je významný závisí na autorovi tabuľky. Toto rozhodnutie ovplyvňuje prehľadnosť a veľkosť tabuľky, ale nemá vplyv na správnosť a zložitosť výsledného matematického modelu. Počiatočný stav je stav, do ktorého sa systém dostane po inicializácii. Ak nie je priamo opísaný v neformálnej špecifikácii, treba ho definovať na základe konzulácie so zadávateľom, v zhode s bezpečnostnými požiadavkami na systém; prakticky ide o definovanie vstupných a výstupných informácií, ktoré sa viažu k tomuto stavu.

Takto zostavená tabuľka obsahuje informácie, ktoré sú obsiahnuté v neformálnej špecifikácii. Nevyplnené bunky tabuľky v stĺpcoch Vstupy a Výstupy upozorňujú na neúplnosť alebo nejednoznačnosť neformálnej špecifikácie. Tento problém treba dôsledne vyriešiť, pretože nejednoznačnosť a neúplnosť špecifikácie funkčných požiadaviek je zdrojom systematických chýb, ktoré sa spravidla prejavia až v záverečných fázach procesu vývoja systému (prípadne až v prevádzke). Odstránenie takýchto chýb je potom časovo aj finančne náročné.

V tab. 1, ktorá obsahuje prepis neformálnej špecifikácie funkčných požiadaviek pre riadiaci obvod výhybky sú nevyplnené bunky v riadkoch, ktoré prináležia stavu rozrez. Vzhľadom na bezpečnosť systému je vhodné, aby tieto bunky boli doplnené o všetky možné kombinácie hodnôt

príslušných vstupných informácií. Takto doplnená tabuľka (nie je v článku uvedená) ešte nemusí byť úplná. Tabuľka je úplná vtedy, ak pre každé vstupné slovo (počet vstupných slov je 2^k, kde k je počet vstupov) je definovaný prechod medzi stavmi resp. zotrvanie v stave. Úplnosť v tomto prípade neznamená ešte jednoznačnosť.

Tab.1 Tabuľka neformálnej špecifikácie
Tab.1 Table of informal specification

Režim	Vstupy							Výstupy			Stav
				koncová poloha		požiadavka na prestavenie		príkaz na prestavenie			
	záver	voľnosť	núdzové prestavenie	poloha +	poloha -	poloha +	poloha -	do polohy +	do polohy -	rozrez	
normálny +-	0	1	0	1	0	1	0	0	0	0	počiatočný +
	0	1	0	1	0	0	1	0	1	0	
	0	1	0	0	0	0	1	0	1	0	
	0	1	0	0	1	0	1	0	0	0	počiatočný -
normálny -+	0	1	0	0	1	0	1	0	0	0	počiatočný -
	0	1	0	0	1	1	0	1	0	0	
	0	1	0	0	0	1	0	1	0	0	
	0	1	0	1	0	1	0	0	0	0	počiatočný +
núdzový +-	0	0	1	1	0	1	0	0	0	0	počiatočný +
	0	0	1	1	0	0	1	0	1	0	
	0	0	1	0	0	0	1	0	1	0	
	0	0	1	0	1	0	1	0	0	0	počiatočný -
Núdzový -+	0	0	1	0	1	0	1	0	0	0	počiatočný -
	0	0	1	0	1	1	0	1	0	0	
	0	0	1	0	0	1	0	1	0	0	
	0	0	1	1	0	1	0	0	0	0	počiatočný +
reverzácia ++	0	1	0	1	0	1	0	0	0	0	počiatočný +
	0	1	0	1	0	0	1	0	1	0	
	0	1	0	0	0	0	1	0	1	0	
	0	1	0	0	0	1	0	1	0	0	
	0	1	0	1	0	1	0	0	0	0	počiatočný +
reverzácia - -	0	1	0	0	1	0	1	0	0	0	počiatočný -
	0	1	0	0	1	1	0	1	0	0	
	0	1	0	0	0	1	0	1	0	0	
	0	1	0	0	0	0	1	0	1	0	
	0	1	0	0	1	0	1	0	0	0	počiatočný -
núdz_rev ++	0	0	1	1	0	1	0	0	0	0	počiatočný +
	0	0	1	1	0	0	1	0	1	0	
	0	0	1	0	0	0	1	0	1	0	
	0	0	1	0	0	1	0	1	0	0	
	0	0	1	1	0	1	0	0	0	0	počiatočný +
núdz_rev - -	0	0	1	0	1	0	1	0	0	0	počiatočný -
	0	0	1	0	1	1	0	1	0	0	
	0	0	1	0	0	1	0	1	0	0	
	0	0	1	0	0	0	1	0	1	0	
	0	0	1	0	1	0	1	0	0	0	počiatočný -
rozrez z+	0	1	0	1	0	1	0	0	0	0	počiatočný +
		0		0	0	1	0	0	0	1	rozrez
rozrez z-	0	1	0	0	1	0	1	0	0	0	počiatočný -
		0		0	0	0	1	0	0	1	rozrez

V tab. 2 sú uvedené všetky vstupné slová vyplývajúce zo špecifikácie činnosti riadiaceho obvodu výhybky a k nim priradené kombinácie logických hodnôt na jednotlivých vstupoch. Index označujúci vstupné slovo, zodpovedá desiatkovému ekvivalentu binármej kombinácie logických hodnôt na jednotlivých vstupoch. Tie vstupné slová, ktoré nie sú v tabuľke uvedené, sa počas bezporuchovej prevádzky nevyskytujú. Ich výskyt môže byť znakom poruchy. Problém nedefinovaných vstupných slov je riešený neskoršie pri úprave prvotnej prechodovej tabuľky.

V tab. 3 sú uvedené všetky výstupné slová vyplývajúce zo špecifikácie činnosti riadiaceho obvodu výhybky a k nim priradené kombinácie logických hodnôt na jednotlivých výstupoch.

Maximálny počet výstupných slov je 2^l, kde l je počet výstupov. Index označujúci výstupné slovo, zodpovedá desiatkovému ekvivalentu binármej kombinácie logických hodnôt na jednotlivých výstupoch.

Tab. 2 Priradenie vstupov vstupným slovám
Tab. 2 Assignment of inputs to input words

Vstupy							Vstupné slovo
záver	voľnosť	núdz. prestav.	konc. poloha		požiadav. na prest.		
			+	-	+	-	
0	0	0	0	0	0	1	X_1
0	0	0	0	0	1	0	X_2
0	0	1	0	0	0	1	X_{17}
0	0	1	0	0	1	0	X_{18}
0	0	1	0	1	0	1	X_{21}
0	0	1	0	1	1	0	X_{22}
0	0	1	1	0	0	1	X_{25}
0	0	1	1	0	1	0	X_{26}
0	1	0	0	0	0	1	X_{33}
0	1	0	0	0	1	0	X_{34}
0	1	0	0	1	0	1	X_{37}
0	1	0	0	1	1	0	X_{38}
0	1	0	1	0	0	1	X_{41}
0	1	0	1	0	1	0	X_{42}
1	0	0	0	0	0	1	X_{65}
1	0	0	0	0	1	0	X_{66}
1	0	1	0	0	0	1	X_{81}
1	0	1	0	0	1	0	X_{82}

Tab. 3 Priradenie výstupov výstupným slovám
Tab. 3 Assignment of outputs to output words

Výstupy			Výstupné slovo
príkaz na prestavenie		rozrez	
do polohy +	do polohy -		
0	0	0	Y_0
0	0	1	Y_1
0	1	0	Y_2
1	0	0	Y_3

Aplikáciou označenia vstupných a výstupných slov, ktoré je uvedené v tab. 2 a v tab. 3, možno tab. 1 upraviť do prehľadnejšieho tvaru (tab. 4). V tab. 4 sú už označené všetky stavy systému. Stavy, ktoré sú v tab. 1 rovnako slovne pomenované, možno považovať za identické. To znamená, že každému riadku tab. 1 zodpovedá nový stav systému, okrem stavov s rovnakým slovným pomenovaním.

Nech $s_{k(t)}$ je identifikátor výskytu stavu S_k v čase t a nech $s_{k(t+t_0)}$ je identifikátor výskytu stavu S_k v čase $t+t_0$. Nech binárne usporiadanie týchto identifikátorov je nasledovné:

- $s_{k(t)} = 0$, ak sa systém v čase t nenachádza v stave S_k;
- $s_{k(t)} = 1$, ak sa systém v čase t nachádza v stave S_k;
- $s_{k(t+t_0)} = 0$, ak sa systém v čase $t+t_0$ nenachádza v stave S_k;
- $s_{k(t+t_0)} = 1$, ak sa systém v čase $t+t_0$ nachádza v stave S_k.

Tab. 4 Upravená tabuľka neformálnej špecifikácie
Tab. 4 Modified table of informal specification

Režim	Vstupné slovo	Výstupné slovo	Stav	Označenie stavu
normálny +-	X_{14}	Y_0	počiatočný +	S_0
	X_{13}	Y_2		S_1
	X_9	Y_2		S_2
	X_{11}	Y_0	počiatočný -	S_3
normálny -+	X_{11}	Y_0	počiatočný -	S_3
	X_{12}	Y_3		S_4
	X_{10}	Y_3		S_5
	X_{14}	Y_0	počiatočný +	S_0
núdzový +-	X_8	Y_0	počiatočný +	S_0
	X_7	Y_2		S_6
	X_3	Y_2		S_7
	X_5	Y_0	počiatočný -	S_3
núdzový -+	X_5	Y_0	počiatočný -	S_3
	X_6	Y_3		S_8
	X_4	Y_3		S_9
	X_8	Y_0	počiatočný +	S_0
reverzácia ++	X_{14}	Y_0	počiatočný +	S_0
	X_{13}	Y_2		S_{10}
	X_9	Y_2		S_{11}
	X_{10}	Y_3		S_{12}
	X_{14}	Y_0	počiatočný +	S_0
reverzácia - -	X_{11}	Y_0	počiatočný -	S_3
	X_{12}	Y_3		S_{13}
	X_{10}	Y_3		S_{14}
	X_9	Y_2		S_{15}
	X_{11}	Y_0	počiatočný -	S_3
núdz_rev ++	X_8	Y_0	počiatočný +	S_0
	X_7	Y_2		S_{16}
	X_3	Y_2		S_{17}
	X_4	Y_3		S_{18}
	X_8	Y_0	počiatočný +	S_0
núdz_rev - -	X_5	Y_0	počiatočný -	S_3
	X_6	Y_3		S_{19}
	X_4	Y_3		S_{20}
	X_3	Y_2		S_{21}
	X_5	Y_0	počiatočný -	S_3
rozrez z +	X_{14}	Y_0	počiatočný +	S_0
	X_2, X_{16}, X_{18}, X_4	Y_1	rozrez	S_{22}
rozrez z -	X_{11}	Y_0	počiatočný -	S_3
	X_1, X_{15}, X_{17}, X_3	Y_1	rozrez	S_{22}

Nech

$$\underline{\underline{S}}_{(t)} = \left\{ s_{0(t)}, s_{1(t)}, \ldots, s_{n-1(t)} \right\}^T \qquad (1)$$

je vektor identifikátorov výskytu stavu systému v čase t.

Nech

$$\underline{\underline{S}}_{(t+t_0)} = \left\{ s_{0(t+t_0)}, s_{1(t+t_0)}, \ldots, s_{n-1(t+t_0)} \right\}^T \qquad (2)$$

je vektor identifikátorov výskytu stavu systému v čase $t + t_0$.

Nech $x_{r(t)}$ je identifikátor výskytu vstupného slova X_r v čase *t*. Nech binárne usporiadanie tohto identifikátorov je nasledovné:

- $x_{r(t)} = 0$, ak v čase *t* sa nenachádza na vstupoch systému vstupné slovo X_r;
- $x_{r(t)} = 1$, ak v čase *t* sa nachádza na vstupoch systému vstupné slovo X_r.

Nech $\mathbf{X}_{(t)}$ je matica rozmeru *n* x *n*, ktorej prvkami sú identifikátory výskytu vstupných slov. Táto matica formálne zodpovedá transponovanej inverznej prechodovej matici, pričom na rozdiel od nej neobsahuje vstupné slová, ale identifikátory príslušných vstupných slov systému. Prvky matice identifikátorov vstupných slov **X** sú definované takto:

- $x_{ij} = 0$, ak nie je možný prechod zo stavu S_j do stavu S_i;
- $x_{ij} = x_{r(t)}$, ak v čase *t* je možný prechod zo stavu S_j do stavu S_i pôsobením vstupného slova X_r.

$i, j = 0,1,...,n-1.$

Za týchto predpokladov platí, že

$$\vec{S}_{(t+t_0)} = \mathbf{X}_{(t)}.\vec{S}_{(t)}. \qquad (3)$$

Použitím vzťahu (3) možno zostaviť prechodovú tabuľku konečného automatu opisujúceho definovaný systém. Počet riadkov tabuľky je daný počtom stavov systému a počet stĺpcov tabuľky je daný počtom vstupných slov tabuľky. Prechodová tabuľka reprezentuje zápis konečného automatu, v ktorom bunky vo vnútri tabuľky predstavujú stavy v čase $t + t_0$. V konkrétnej bunke tabuľky, ktorá sa nachádza na priesečníku *i*-tého riadku a *j*-tého stĺpca tabuľky, je zapísaný ten stav systému v čase $t + t_0$, do ktorého sa systém dostane, ak sa v čase *t* nachádza v stave prislúchajúcom uvažovanému riadku tabuľky a na vstupoch systému je v čase *t* vstupné slovo prislúchajúce uvažovanému stĺpcu tabuľky.

Do uvažovanej vnútornej bunky prechodovej tabuľky sa zapíše ten stav S_k (v tomto okamihu riešenia môže jedna bunka obsahovať aj viac stavov – nedeterministický systém), ktorého identifikátor vyhovuje podmienke vyplývajúcej s rovnice (3):

$$s_{k(t+t_0)} = x_{ij}.s_{r(t)} = 1. \qquad (4)$$

Pozícia tejto bunky je daná priesečníkom *j*-tého riadku ($s_{r(t)}$ je identifikátor výskytu stavu S_r, ktorý prislúcha *j*-tému riadku) a toho stĺpca, ktorému je priradený vstupný vektor, ktorého indikátor sa nachádza na pozícii prvku x_{ij} v matici $\mathbf{X}_{(t)}$.

Ak platí, že

$$s_{k(t+t_0)} = x_{ij}.s_{r(t)} = 0, \qquad (5)$$

potom príslušná bunka prechodovej tabuľky bude prázdna.

Tab.5 Prechodová tabuľka
Tab.5 Transition table

	X_1	X_2	X_{17}	X_{18}	X_{21}	X_{22}	X_{25}	X_{26}	X_{33}	X_{34}	X_{37}	X_{38}	X_{41}	X_{42}	X_{65}	X_{66}	X_{81}	X_{82}	Výstupné slovo
S_0	S_0	S_{14}	S_0	S_{14}	S_0	S_0	S_6	S_0	S_0	S_0	S_0	S_0	S_1	S_0	S_0	S_{14}	S_0	S_{14}	Y_0
S_1	S_1	S_1	S_1	S_1	S_1	S_1	S_1	S_1	S_2	S_1	S_1	S_1	S_1	S_1	S_1	S_1	S_1	S_1	Y_2
S_2	S_2	S_2	S_2	S_2	S_2	S_2	S_2	S_2	S_2	S_{10}	S_3	S_2	S_2	S_2	S_2	S_2	S_2	S_2	Y_2
S_3	S_{14}	S_3	S_{14}	S_3	S_3	S_8	S_3	S_3	S_3	S_3	S_3	S_4	S_3	S_3	S_{14}	S_3	S_{14}	S_3	Y_0
S_4	S_4	S_4	S_4	S_4	S_4	S_4	S_4	S_4	S_4	S_5	S_4	S_4	S_4	S_4	S_4	S_4	S_4	S_4	Y_4
S_5	S_5	S_5	S_5	S_5	S_5	S_5	S_5	S_5	S_{11}	S_5	S_5	S_5	S_5	S_0	S_5	S_5	S_5	S_5	Y_4
S_6	S_6	S_6	S_7	S_6	S_6	S_6	S_6	S_6	S_6	S_6	S_6	S_6	S_6	S_6	S_6	S_6	S_6	S_6	Y_2
S_7	S_7	S_7	S_7	S_{12}	S_3	S_7	S_7	S_7	S_7	S_7	S_7	S_7	S_7	S_7	S_7	S_7	S_7	S_7	Y_2
S_8	S_8	S_8	S_8	S_9	S_8	S_8	S_8	S_8	S_8	S_8	S_8	S_8	S_8	S_8	S_8	S_8	S_8	S_8	Y_4
S_9	S_9	S_9	S_{13}	S_9	S_9	S_9	S_9	S_0	S_9	S_9	S_9	S_9	S_9	S_9	S_9	S_9	S_9	S_9	Y_4
S_{10}	S_{10}	S_{10}	S_{10}	S_{10}	S_{10}	S_{10}	S_{10}	S_{10}	S_{10}	S_{10}	S_{10}	S_{10}	S_{10}	S_0	S_{10}	S_{10}	S_{10}	S_{10}	Y_4
S_{11}	S_{11}	S_{11}	S_{11}	S_{11}	S_{11}	S_{11}	S_{11}	S_{11}	S_{11}	S_{11}	S_3	S_{11}	S_{11}	S_{11}	S_{11}	S_{11}	S_{11}	S_{11}	Y_2
S_{12}	S_{12}	S_{12}	S_{12}	S_{12}	S_{12}	S_{12}	S_{12}	S_0	S_{12}	S_{12}	S_{12}	S_{12}	S_{12}	S_{12}	S_{12}	S_{12}	S_{12}	S_{12}	Y_4
S_{13}	S_{13}	S_{13}	S_{13}	S_{13}	S_3	S_{13}	S_{13}	S_{13}	S_{13}	S_{13}	S_{13}	S_{13}	S_{13}	S_{13}	S_{13}	S_{13}	S_{13}	S_{13}	Y_2
S_{14}	S_{14}	S_{14}	S_{14}	S_{14}	S_{14}	S_{14}	S_{14}	S_{14}	S_{14}	S_{14}	S_{14}	S_{14}	S_{14}	S_{14}	S_{14}	S_{14}	S_{14}	S_{14}	Y_1

Takto zostavený konečný automat je spravidla nedeterministický. Preto treba na základe podmienok o zlučiteľnosti stavov [2] zlúčiť tie stavy S_f a S_h ($f \neq h$), ktorých identifikátory spĺňajú podmienku:

$$s_{f(t+t_0)} = s_{h(t+t_0)} = 1, \tag{6}$$

pričom pre všetky vstupné slová X_i, ak $i = 1,2, ..., k$, platí, že

$$V_{(S_f, X_{i(t)})} = V_{(S_h, X_{i(t)})}, \tag{7}$$

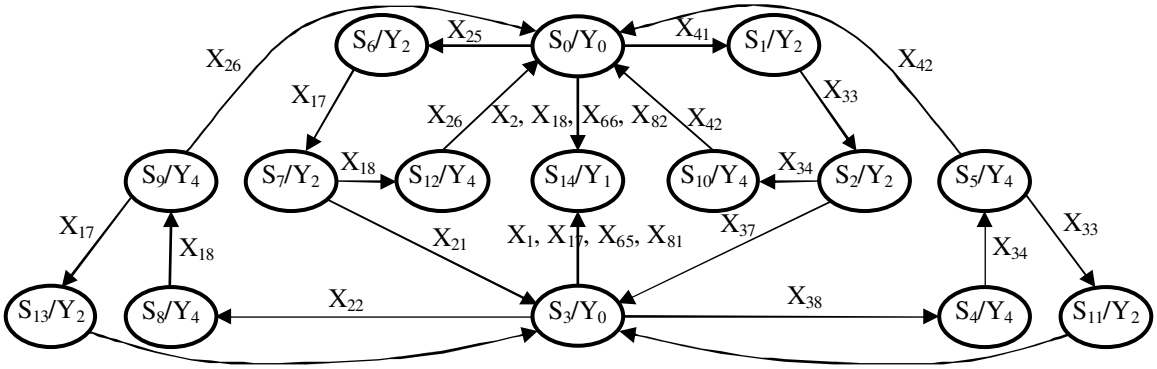
kde $V_{(S_f, X_{i(t)})}$ je výstupné slovo systému v stave S_f a v čase $t + t_0$, do ktorého sa systém dostal pôsobením vstupného slova X_i v čase t.

Zlučovanie stavov v prechodovej tabuľke treba opakovať dovtedy, pokiaľ nenastane jedna z dvoch možností:

- v prechodovej tabuľke už nie sú zlučiteľné stavy (t. j. v žiadna bunka tabuľky už neobsahuje dva stavy);
- stavy nachádzajúce sa v jednej bunke nie sú zlučiteľné (ich výstupné slová sú navzájom odlišné); tento fakt znamená, že v neformálna špecifikácia obsahuje rozporné tvrdenia; tieto rozporné tvrdenia sa nachádzajú v tých bodoch špecifikácie, ktoré zodpovedajú príslušným stavom v tab. 4.

Prázdne bunky prechodovej tabuľky poukazujú na neúplnosť špecifikácie. Tieto bunky zodpovedajú stavu, pre ktorý nie je definované správanie sa systému po príchode vstupného slova. Pre systémy na riadenie bezpečnostne kritických procesov je nutné, aby ich správanie sa bolo v každom stave jednoznačne definované pre všetky vstupné slová. Odstránenie nedostatkov v špecifikácii (doplnenie prechodovej tabuľky a odstránenie nezlučiteľných stavov) možno urobiť len na základe konzultácie sa zadávateľom. V tab. 5(prechodová tabuľka pre obvod riadenia výhybka) sú uvedené prechody medzi stavmi systému len pre vstupné slová zavedené v tab. 2. Pre následné spracovanie takto definovaného konečného automatu podľa [3] platí, že pre vstupné slová neuvedené v tab. 5 systém svoj stav nemení.

Konečný automat vytvorený uvedeným postupom a zapísaný prechodovou tabuľkou môže obsahovať nadbytočné stavy. Minimalizáciu počtu stavov konečného automatu možno uskutočniť postupom uvedeným v [2].



Obr. 2. Stavový diagram
Fig. 2. State diagram

Na obr. 2. je stavový diagram vytvorený na základe prechodovej tabuľky (tab. 5).

4. ZÁVER

Uvedený postup formalizácie funkčných požiadaviek na systém je algoritmizovateľný (okrem úvodného prepisu špecifikácie do tabuľkovej formy), čo dáva predpoklady na automatické generovanie stavového diagramu a následne na automatické generovanie kódu na základe pravidiel uvedených v [3].

V prípade zložitých riadiacich systémov treba dekomponovať systém na moduly. Dôležité je, aby bola vhodne zvolená úroveň dekompozície systému. Všeobecne platí, že čím sú moduly menšie, tým sú jednoduchšie, ale na druhej strane je zložitejšie riadenie ich vzájomnej koordinácie. Moduly treba vytvoriť tak, aby modul bol jasný a zrozumiteľný, aby zodpovedal špecifickej funkcii, aby mal čo najmenší počet vstupov a výstupov, aby rozhrania medzi modulmi boli presne definované a výmena informácií medzi modulmi bola čo najmenšia.

Článok bol spracovaný za podpory výskumnej úlohy 46/604 Dopravná telematika a nástroje na zvyšovanie jej kvality.

LITERATÚRA

[1] STN 342620: *Predpisy pre staničné zabezpečovacie zariadenia*. 1996

[2] FRIŠTACKÝ, N. - KOLESÁR, M. - KOLENIČKA, J. - HLAVATÝ, J.: *Logické systémy*. Vydavateľstvo Alfa, Bratislava, 1986

[3] RÁSTOČNÝ, K. – ŽDÁNSKY, J.: *Použitie konečného automatu pri programovaní PLC*. AEEE No. 1 Vol.3/2004. ŽU v Žiline, s. 45 - 49, ISSN 1336-1376