

Vážení zákazníci,

dovolujeme si Vás upozornit, že na tuto ukázkou knihy se vztahují autorská práva, tzv. copyright.

To znamená, že ukáзка má sloužit výhradně pro osobní potřebu potenciálního kupujícího (aby čtenář viděl, jakým způsobem je titul zpracován a mohl se také podle tohoto, jako jednoho z parametrů, rozhodnout, zda titul koupí či ne).

Z toho vyplývá, že není dovoleno tuto ukázkou jakýmkoliv způsobem dále šířit, veřejně či neveřejně např. umístováním na datová média, na jiné internetové stránky (ani prostřednictvím odkazů) apod.

redakce nakladatelství BEN – technická literatura
redakce@ben.cz



4.3.8 Impulzní výstupy nespojitých regulátorů

V dosavadních příkladech řešení nespojitých regulátorů platilo, že vyčíslené hodnoty výstupů byly platné, dokud nedošlo k jejich změně. Pokud je např. programem regulátoru opakovaně nastavována jedničková hodnota akční veličiny, stav binárního výstupu PLC se nemění. Tomuto způsobu ovládání se někdy říká *hladinové*. Protikladem je *impulzní ovládání výstupů*. Obvykle je využíváno při aktivaci programu v pevném časovém rastru, ale není to podmínkou. Například pro třístavový regulátor s přírůstkovým charakterem výstupů (*obr. 4.8c*) mohou být místo hladinových výstupů v každém výpočetním kroku generovány impulzy typu *přidávej – ubírej*. Mohou mít pevně zadanou délku. Pokud je jejich délka úměrná akční veličině spočtené podle algoritmu PID přiblížíme se vlastnostem spojitých regulátorů – jde o šířkovou modulaci nespojitých výstupů.

4.4 Smíšený systém s dvěma a více vstupy

4.4.1 Motivace

Dosud uváděné příklady (kap. 4.2 a kap. 4.3) byly určeny především jako ilustrace možných způsobů vyhodnocení číslicové informace ve smíšených logických systémech. Proto byly záměrně zjednodušeny na jednorozměrné případy, kdy byla vyhodnocována jediná číselná (jazyková) proměnná. V praxi se ale obvykle vyskytují smíšené systémy se dvěma nebo více jazykovými proměnnými, jejichž chování je určováno vztahy mezi jejich termy, popsanými souborem pravidel, logických výrazů, tabulkami apod. Vedle číslicových proměnných se v reálných systémech vyskytují i samostatné binární logické proměnné, které ovlivňují jejich chování. Často se jedná o systémy, které řeší diagnostiku, regulaci, rozhodování a podobné typy logických úloh.

4.4.2 Systémy se dvěma vstupy

Příklad 4.8 Jednoduchý diagnostický systém

Pro ilustraci uvažujme příklad jednoduchého diagnostického systému (např. ložisek), který vyhodnocuje vibrace a teplotu a stanovuje diagnózu: *OK – varování1 – varování2 – alarm – porucha*. Předpokládejme, že vibrace jsou vyhodnocovány stejně jako v příkladu 4.1 a nad číselnými hodnotami intenzity vibrací jsou definovány dvouhodnotové logické proměnné (vstupní termy) *v_nízké*, *v_zvýšené*, *v_vysoké* (*obr. 4.2*). Podobně jsou pro teplotní pásma definovány vstupní termy *normální*, *zvýšená*, *přehřátí*, jako v příkladu 4.2 (*obr. 4.3*).

Řešení 1: binární logický systém

Situaci můžeme interpretovat tak, že nad množinou hodnot teplotní osy (vstupní jazykové proměnné *teplota*) a nad množinou hodnot intenzity vibrací (vstupní jazykové proměnné *vibrace*) definujeme po třech dvouhodnotových proměnných (vstupních ter-

mech), z nichž každá nabývá pravdivostní hodnoty 1 nebo 0 (hodnota vstupní jazykové proměnné je, anebo není v odpovídajícím pásmu). Podobně lze interpretovat diagnostické stavy, jako pět stejnojmenných výstupních logických proměnných (termů), jejichž pravdivostní hodnota (1 nebo 0) nese informaci o odpovídajícím diagnostickém stavu. Obvykle jsou definovány souborem logických pravidel, např.:

```
jestliže v_nízké AND normální pak OK
jestliže v_nízké AND zvýšená pak varování1
jestliže v_nízké AND přehřátí pak varování2
jestliže v_zvýšené AND normální pak varování1
jestliže v_zvýšené AND zvýšená pak varování2
jestliže v_zvýšené AND přehřátí pak alarm
jestliže v_vysoké AND normální pak varování2
jestliže v_vysoké AND zvýšená pak alarm
jestliže v_vysoké AND přehřátí pak porucha
```

nebo rovnocenně:

```
jestliže (vibrace jsou nízké) AND (teplota je normální)
pak (diagnóza je OK)
jestliže (vibrace jsou nízké) AND (teplota je zvýšená)
pak (diagnóza je varování1)
```

.....

Pravidla mají formu implikace, ale pro booleovskou logiku je lze přepsat do ekvivalentní formy logických výrazů, v nichž pravdivost důsledku je rovna pravdivosti podmínky (v našem případě součinu dvou termů):

```
OK = v_nízké AND normální (4.1)
varování1 = v_nízké AND zvýšená
varování2 = v_nízké AND přehřátí
varování1 = v_zvýšené AND normální
varování2 = v_zvýšené AND zvýšená
alarm = v_zvýšené AND přehřátí
varování2 = v_vysoké AND normální
alarm = v_vysoké AND zvýšená
porucha = v_vysoké AND přehřátí
```

Pro zápis pravidel je výhodné (a praktické) použít pravdivostní tabulku nebo mapu. Zdánlivě by pro šestici vstupních proměnných měla mít rozsah $2^6 = 64$ položek. To by ale platilo, pokud by všechny vstupní proměnné byly navzájem nezávislé – všechny jejich kombinace by byly možné. Ve skutečnosti se ale termy nad společnou jazykovou proměnnou navzájem vylučují, takže pouze (právě) jeden z nich může být pravdivý. Dostupných je pak jen $3 \times 3 = 9$ kombinací. Soubor pravidel nebo logických výrazů tak lze názorně zobrazit pravdivostní mapou (maticí, dvojrozměrnou tabulkou) na obr. 4.15. Každému z políček odpovídá logický součin (AND) odpovídajících vstupních termů. Řádkům odpovídají termy nad jazykovou proměnnou *teplota* (*normální*, *zvýšená*, *přehřátí*) a sloupcům termy nad jazykovou proměnnou *vibrace* (*v_nízké*, *v_zvýšené*,

v_vysoké). V každém z políček je uvedeno jméno výstupního termu, jehož pravdivost je definována odpovídajícím logickým součinem. Obsah mapy je rovnocenný uvedenému souboru logických výrazů. Výhodou mapy je její prostorová úspornost a názornost zápisu pravidel.

Je vidět, že některé termy se vyskytují ve více políčkách tabulky a rovnocenně na levé straně několika logických výrazů – jsou „osloveny“ více podmínkovými součiny. Každý z odpovídajících součinů lze chápat jako příspěvek k výsledné pravdivosti osloveného výstupního termu. Proto se jejich pravdivosti logicky počítají

```
OK = v_nízké AND normální           (4.2)
varování1 = (v_nízké AND zvýšená) OR (v_zvýšené AND normální)
varování2 = (v_nízké AND přehřátí) OR (v_zvýšené AND zvýšená)
           OR (v_vysoké AND normální)
alarm = (v_zvýšené AND přehřátí) OR (v_vysoké AND zvýšená)
porucha = v_vysoké AND přehřátí
```

Podobně je soubor pravidel se společným důsledkem, např.:

```
jestliže v_nízké AND přehřátí pak varování2
jestliže v_zvýšené AND zvýšená pak varování2
jestliže v_vysoké AND normální pak varování2
```

rovnocenný jedinému pravidlu, v jehož podmínkové části se vyskytuje logický součet dílčích podmínek:

```
jestliže (v_nízké AND přehřátí) OR (v_vyšší AND zvýšená)
           OR (v_vysoké AND normální)
```

pak varování2

Logické výrazy (4.1) a (4.2) jsou rovnocenné a oba soubory mohou být východiskem pro zápis programu PLC. Pro jazyk strukturovaného textu má pro výrazy (4.2) program téměř shodnou a liší se jen drobnými syntaktickými detaily,

```
OK := v_nízké AND normální
varování1 := (v_nízké AND zvýšená) OR (v_zvýšené AND normální)
varování2 := (v_nízké AND přehřátí) OR (v_zvýšené AND zvýšená)
           OR (v_vysoké AND normální)
alarm := (v_zvýšené AND přehřátí) OR (v_vysoké AND zvýšená)
porucha := v_vysoké AND přehřátí
```

při použití výrazů (4.1) je třeba sečíst pravdivosti všech dílčích podmínek:

```
OK := v_nízké AND normální
varování1 := v_nízké AND zvýšená
varování2 := v_nízké AND přehřátí
varování1 := varování1 OR (v_zvýšené AND normální)
varování2 := varování2 OR (v_zvýšené AND zvýšená)
alarm := v_zvýšené AND přehřátí
varování2 := varování2 OR (v_vysoké AND normální)
alarm := alarm OR (v_vysoké AND zvýšená)
```

```
porucha := v_vysoké AND přehřátí
```

Program v jazyku mnemokódů lze získat rutinní úpravou výchozích výrazů pro soubor (4.1):

```
ld v_nízké
and normální
wr OK
ld v_nízké
and zvýšená
wr varování1
ld v_nízké
and přehřátí
wr varování2
ld v_zvýšené
and normální
set varování1
ld v_zvýšené
and zvýšená
set varování2
ld v_zvýšené
and přehřátí
wr alarm
ld v_vysoké
and normální
set varování2
ld v_vysoké
and zvýšená
set alarm
ld v_vysoké
and přehřátí
wr porucha
```

K přičítání příspěvků k pravdivosti již vyčíslených výstupních termů je použita instrukce SET, která je sice určena k nastavení paměťové funkce typu RS, ale lze ji též interpretovat jako operaci logického přičtení obsahu střádače k obsahu adresovaného místa. Pokud jsou jako výchozí použity výrazy (4.2) bude mít program tvar:

```
ld v_nízké
and normální
wr OK
ld v_nízké
and zvýšená
ld v_zvýšené
and normální
or
wr varování1
```

```

ld v_nízke
and přehřátí
ld v_zvýšené
and zvýšená
or
ld v_vysoké
and normální
or
wr varování2
ld v_zvýšené
and přehřátí
ld v_vysoké
and zvýšená
or
wr alarm
ld v_vysoké
and přehřátí
wr porucha

```

Ve všech programech jsme mlčky předpokládali, že proměnné s pravdivostmi vstupních termů mají bitový charakter (bit, BOOL) a jejich pravdivosti byly vyčísleny některým ze způsobů uvedených v příkladech 4.1 a 4.2 (zde již nejsou uváděny). Analogicky předpokládáme, že i proměnné s hodnotami výstupních termů jsou opět deklarovány jako bitové.

Řešení 2: Tříhodnotové operandy, binární výstupy

Situaci z předchozího případu lze rovnocenně interpretovat jako logický systém ve tříhodnotové logice. Trojici intervalů nad vstupní jazykovou proměnnou *teplota* s významem *normální*, *zvýšená*, *přehřátí* lze přiřadit tři hodnoty pravdivosti tříhodnotové logické proměnné *přehřátí* (např. 0 – 0,5 – 1). Podobně lze intervalům hodnot intenzity vibrací přiřadit pravdivosti tříhodnotové logické proměnné *vysoké*. Mapu pravidel z obr. 4.15 pak lze upravit podle obr. 4.16 a chápat ji jako sdruženou mapu pravdivosti pětice dvouhodnotových termů pro dva tříhodnotové operandy *přehřátí* a *vysoké* – jako zobecněnou Karnaughovu mapu (K-mapu) pro dvě tříhodnotové proměnné. Místo prav-

teplota \ vibrace	nízke	zvýšené	vysoké
	normální	OK	varování 1
zvýšená	varování 1	varování 2	alarm
přehřátí	varování 2	alarm	porucha

Obr. 4.15

Mapa pravidel pro diagnózu ložisek

divostních hodnot operandů (zde 0 – 0,5 – 1) lze používat přímo pořadová čísla (indexy) odpovídajících intervalů hodnot vstupních jazykových proměnných (obr. 4.17). Je to výhodné zejména při přímé realizaci logické funkce pravdivostní tabulkou. Mapy na obr. 4.16 a obr. 4.17 definují pěťici výstupních logických proměnných. Jsou tedy sdruženými pravdivostními mapami, vzniklými sloučením pěti pravdivostních map pro pravdivost každého z výstupních termů.

přehřátí	vysoké	0	0,5	1
	0	OK	varování 1	varování 2
0,5		varování 1	varování 2	alarm
1		varování 2	alarm	porucha

Obr. 4.16

Sdružená pravdivostní mapa pěti binárních funkcí s tříhodnotovými operandy

přehřátí	vysoké	0	1	2
	0	OK	varování 1	varování 2
1		varování 1	varování 2	alarm
2		varování 2	alarm	porucha

Obr. 4.17

Sdružená pravdivostní mapa pěti binárních funkcí jejichž operandy suplují indexy vstupních proměnných

přehřátí	vysoké	OK	varování 1	varování 2	alarm	porucha
0 (0)	0 (0)	1	0	0	0	0
0 (0)	0,5 (1)	0	1	0	0	0
0 (0)	1 (2)	0	0	1	0	0
0,5 (1)	0 (0)	0	1	0	0	0
0,5 (1)	0,5 (1)	0	0	1	0	0
0,5 (1)	1 (2)	0	0	0	1	0
1 (2)	0 (0)	0	0	1	0	0
1 (2)	0,5 (1)	0	0	0	1	0
1 (2)	1 (2)	0	0	0	0	1

Obr. 4.18

Pravdivostní tabulka pěti binárních funkcí s tříhodnotovými operandy

Pravdivostní tabulka je jednorozměrným ekvivalentem pravdivostní mapy. V terminologii datových struktur je tabulka vektorem (jednorozměrným polem), zatímco mapa je formálně maticí (dvourozměrným polem). Stejně jako mapa, definuje i pravdivostní tabulka hodnoty výstupní funkce (skupiny výstupních funkcí) pro všechny kombinace operandů. Odpovídající pravdivostní tabulku získáme podobným postupem jako při převodu vývojové tabulky stavového automatu na tabulku pro program. Obdobně, jako u pravdivostních tabulek pro dvouhodnotové operandy, jsou pravdivostní tabulky pro vícehodnotové operandy uspořádány tak, že v jejich záhlaví se vyskytují všechny kombinace proměnných v narůstajícím pořadí. Na *obr. 4.18* je uvedena jedna varianta pravdivostní tabulky dvou tříhodnotových proměnných. Vznikla z tabulky na *obr. 4.16* a *obr. 4.17* čtením (rozmitáním) po řádcích. Výsledný index lze spočítat podle vztahu:

$$\text{index_hodnoty} := 3 * \text{ind_přehřátí} + \text{ind_vysoké}$$

Nejjednodušším způsobem realizace je přímé využití tabulky jako datové struktury v programu:

```
# table byte tab_termů = %10000, %01000, %00100,
    %01000, %00100, %00010,
    %00100, %00010, %00001
ld ind_přehřátí
mul 3
add ind_vysoké
lrb tab_termů
wr termy
```

V programu je předpokládáno, že indexy pásem hodnot vibrací a teploty byly již vyčísleny některým z postupů, uvedených v příkladech 4.1 a 4.2 a jsou uloženy v proměnných *ind_vysoké* a *ind_přehřati*. Obsah tabulky *tab_termu* je opsán z *obr. 4.18*. Z této pozice je instrukcí LTB vybrána odpovídající položka a je uložena na adresu *termy*. Umístění termů je shodné s tabulkou na *obr. 4.18* (OK na pozici pátého bitu *termy:4* a porucha na nejnižší pozici *termy:0*). Rovnocenně lze vytvořit pravdivostní tabulku z mapy (*obr. 4.16* a *obr. 4.17*) rozkladem po sloupcích. Index hodnoty pak spočteme podle vzorce:

$$\text{index_hodnoty} := 3 * \text{ind_vysoké} + \text{ind_přehřátí}$$

Poznámka – rovnocennost booleovské a vícehodnotové logiky.

Lze dokázat, že vícehodnotové logiky a dvouhodnotová logika jsou stejně mohutný aparát pro popis a řešení problémů a že jsou navzájem převoditelné. Je to ilustrováno na řešení 1 až 4 příkladu 4.8.

Řešení 3: Tříhodnotové operandy, pětihodnotový výstup

Pravdivostní hodnoty pěti výstupních termů (dvouhodnotových logických proměnných) můžeme využít např. k ovládní pěti signálů s varovnými nápisy nebo jiných indikátorů stavu stroje. Lze ale předpokládat, že údržbář dá přednost jediné výstupní proměnné *riziko_poruchy* s odstupňovanými hodnotami, která vyjadřuje naléhavosti stavu. Můžeme ji opět považovat za vícehodnotovou logickou proměnnou – zde pětihodnotovou. Je

přirozené, abychom její pravdivostní hodnoty pravidelně odstupňovali v řadě 0 – 0,25 – 0,5 – 0,75 – 1, kde nulové hodnotě odpovídá zaručeně bezchybný stav, zatímco jedničkové odpovídá jistota poruchy. Obvyklé je i vyjádření pravdivosti v procentech: 0 % – 25 % – 50 % – 75 % – 100 %. Na *obr. 4.19* je uvedena pravdivostní mapa této funkce. Lze ji realizovat programem ve formě posloupnosti podmíněných příkazů

```

if v_nízke AND normální then riziko_poruchy := 0
if (v_nízke AND zvýšená) OR (v_zvýšené AND normální)
    then riziko_poruchy := 25
if (v_nízke AND přehřátí) OR (v_zvýšené AND zvýšená) OR
    (v_vysoké AND normální) then riziko_poruchy := 50
if (v_zvýšené AND přehřátí) OR (v_vysoké AND zvýšená)
    then riziko_poruchy := 75
if v_vysoké AND přehřátí then riziko_poruchy := 100

```

nebo s využitím tabulky – program zůstane stejný, jako v řešení 2 tohoto příkladu, jen se změní obsah tabulky

```
# table byte tab_riziko = 0, 25, 50, 25, 50, 75, 50, 75, 100
```

V našem případě, kdy je pravdivostní mapa (*obr. 4.19*) symetrická podle své hlavní úhlopříčky (není to podmínkou – jen jsme si tak zvolili své řešení), lze pravdivostní hodnotu přímo spočítat z hodnot indexů obou operandů

```
riziko_poruchy := (ind_přehřátí + ind_vysoké)*25
```

Nic nám ale nebrání v tom, abychom si počet pravdivostních hodnot výstupní logické proměnné *riziko_poruchy* a jejich odstupňování zvolili podle vlastního uvážení, např. pět hodnot v řadě 0,15 – 0,35 – 0,5 – 0,7 – 0,9 (15 % – 35 % – 50 % – 70 % – 90 %) nebo pro každé z políček jinou hodnotu.

		vysoké		
	přehřátí	0	1	2
0		0	0,25	0,5
1		0,25	0,5	0,75
2		0,5	0,75	1

Obr. 4.19

Pravdivostní mapa pětihodnotové logické funkce

Poznámka k počtu stavů

Skutečnost, že v našem příkladu jsou oba operandy právě tříhodnotovými logickými proměnnými není významná. Vyplývá jen ze zadání zjednodušeného příkladu. Obecně mohou mít operandy libovolný počet stavů (pravdivostních hodnot) a navíc každý může mít jiný počet stavů – vše je jen otázkou naší volby v závislosti na povaze řešené úlohy. Můžeme si např. představit obdobný diagnostický systém, který při vyhodnocení teploty rozlišuje pět stavů, jako v příkladu 4.3. Stejně tak může být i počet výstupních termů

libovolný a nezávislý na počtu stavů operandů logické funkce. Existuje jen praktické omezení na přijatelnou komplikovanost a přehlednost řešení. Psychologové uvádějí, že „magické číslo“ 7 je prahem kapacity krátkodobé paměti při našem myšlení. Podržit v mysli a současně zpracovávat větší počet myšlenek, než 7, promýšlet souvislosti mezi nimi se proto stává neúměrně komplikovanějším problémem a je zde zvýšené riziko chyb a zjednodušení. Ani náročnost reálných úloh obvykle nevyžaduje překračování této meze rozlišení.

Řešení 4: Výpočet číselných hodnot z pravdivosti termů

Pokud jsme už nějakým způsobem získali pravdivostní hodnoty souboru termů nad výstupní jazykovou proměnnou (v našem příkladu nad proměnnou *riziko_poruchy*), můžeme z nich vyčíst číselnou hodnotu této výstupní proměnné. Nejobvyklejší je posloupnost podmíněných příkazů

```
if OK then riziko_poruchy := 15
if varování1 then riziko_poruchy := 35
if varování2 then riziko_poruchy := 50
if alarm then riziko_poruchy := 70
if porucha then riziko_poruchy := 90
```

nebo smíšený aritmeticko-logický výraz

```
riziko_poruchy := (OK AND 15) + (varování1 AND 35) +
+ (varování2 AND 50) + (alarm AND 70) + (porucha AND 90)
```

nebo v jazyce mnemokódů

```
ld 15
and OK
ld 35
and varování1
add ; nebo OR
ld 50
and varování2
add ; nebo OR
ld 70
and alarm
add ; nebo OR
ld 90
and porucha
add ; nebo OR
wr riziko_poruchy
nebo rovnocenně
ld 0
wr riziko_poruchy
ld 15
and OK
```

```

set riziko_poruchy
ld 35
and varování1
set riziko_poruchy
ld 50
and varování2
set riziko_poruchy
ld 70
and alarm
set riziko_poruchy
ld 90
and porucha
set riziko_poruchy

```

Za předpokladu, že je pravdivý právě jeden z termů (v našem příkladu), jsou všechny uvedené programy rovnocenné. Místo aritmetického sčítání (symbol „+“ nebo instrukce ADD) lze použít i logický součet (instrukci OR nebo SET). Pokud může být několik termů současně nulových, lze použít již jen operaci aritmetického sčítání. Pokud může nastat situace, kdy jsou všechny termy nulové („díra“ ve formulaci pravidel, tj. neúplný soubor pravidel nebo logických výrazů) poskytuje každý z programů jiné výsledky – chybné, nebo přinejmenším diskutabilní. Program s podmíněnými příkazy (první) nevyčísluje aktuální hodnotu výsledku, která tak zůstává nezměněna od doby, kdy byla naposledy aktualizována nebo inicializována. Programy s aritmeticko-logickými výrazy (druhý až čtvrtý) v tomto případě dávají nulový výsledek, což obvykle neznamená že riziko poruchy je nulové, pouze jsme se nad touto alternativou zapomněli zamyslet a popsat ji v pravidlech.

Řešení 5: Výpočet polohy těžiště

Postup z předchozího řešení lze graficky interpretovat způsobem, který je pro náš případ poněkud násilný, je vhodným východiskem pro zobecnění obdobné situace pro fuzzy logiku. Předpokládejme, že nad výstupní jazykovou proměnnou *riziko_poruchy* je definováno pět termů s obdélníkovými průběhy podle *obr. 4.20a*. V závislosti na pravdivosti termů, vyčíslených v některém ze způsobů v řešení 1 nebo 2 tohoto příkladu bude mít pouze jeden z obdélníků jedničkovou výšku, ostatní budou nulové. Výslednou číselnou hodnotu proměnné *riziko_poruchy* můžeme vypočítat jako vodorovnou souřadnici těžiště tohoto obdélníka. Můžeme si představit, že obdélníky jsou vystřiženy z plechu a připevněny k pevnému rameni zanedbatelné hmotnosti.

Stejný výsledek bychom získali i pro libovolné užší obdélníky se stejnými polohami těžiště, např. podle *obr. 4.20b*. V mezním případě se obdélníky stanou tenkými hmotnými tyčkami, jejichž šířku můžeme zanedbat (považovat za nulovou) a výška odpovídá pravdivosti odpovídajícího termu (*obr. 4.20c*). V matematické abstrakci odpovídá tomuto případu *diracův impulz*, který má sice jedničkovou plochu, ale přitom je nekonečně úzký tenký (má nulovou šířku) a nekonečně vysoký. V praxi se obvykle nazývá singleton – název je odvozen z podobnosti se spektrem jediného harmonického tónu. Můžeme

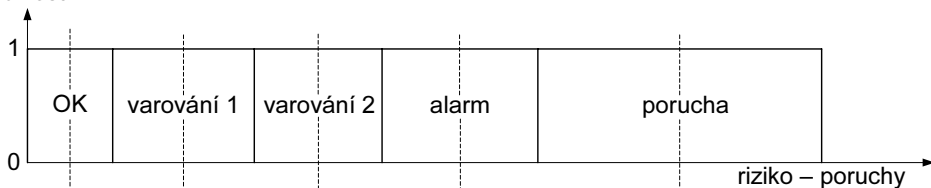
si představit, jakoby v *singletonech* byla soustředěna hmotnost obdélníků. Výsledná číselná hodnota výstupní proměnné je rovna poloze nenulového singletonu.

Popsaný postup je použitelný i v případě, kdy by bylo více výstupních termů současně pravdivých. Pak by ovšem bylo nutné výslednou hodnotu spočítat jako polohu *těžiště singletonů* podle vzorce:

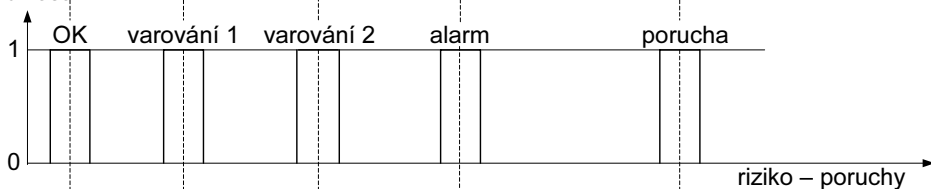
$$\text{riziko_poruchy} := \frac{(p_1 \times \text{OK} + p_2 \times \text{varování1} + p_3 \times \text{varování2} + p_4 \times \text{alarm} + p_5 \times \text{porucha})}{(\text{OK} + \text{varování1} + \text{varování2} + \text{alarm} + \text{porucha})}$$

kde p_1, p_2, p_3, p_4, p_5 , jsou polohy singletonů pro termy *OK*, *varování1*, *varování2*, *alarm*, *porucha*, s jejichž pravdivostními hodnotami zde zacházíme jako s čísly velikosti 0 a 1. Součet poloh nenulových singletonů je dělen jejich počtem. Postup lze považovat za výpočet průměrné hodnoty.

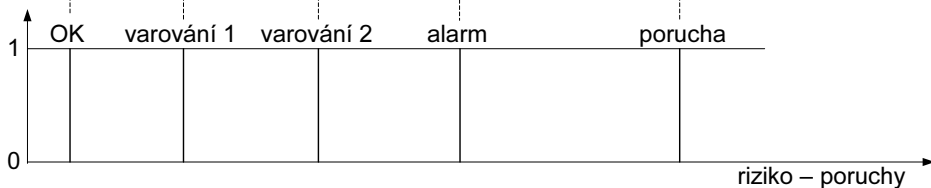
a) pravdivost



b) pravdivost



c) pravdivost



Obr. 4.20

Výpočet těžiště výstupních termů: a) ve tvaru dotýkajících se obdélníků, b) úzkých obdélníků, c) singletonů

4.4.3 Úplnost souboru pravidel a „logické díry“

Významnou výhodou používání mapy pravidel je její názornost, přehlednost a plošná úspornost – informaci, nesenou obsáhlým souborem pravidel lze umístit na plochu nevelikého obdélníku. Každému pravidlu odpovídá údaj jediného políčka, jehož význam je okamžitě patrný a není nutné jej hledat v „záplavě balastních slov“.

Ještě významnější je ale skutečnost, že mapa pravidel nebo pravdivostní tabulka je účinným prostředkem kontroly úplnosti souboru pravidel. Její vyplňování nutí řešitele, aby vyplnil všechna políčka a zamyslel se nad situací, která každému políčku odpovídá. Při formulování souboru pravidel nebo logických výrazů postupujeme intuitivně a vycházíme z požadavků na činnost systému v typických situacích (které jsme si sami vytvořili nebo nám byly sděleny zadavatelem). Mnohdy si neuvědomíme, že v zadání jsou „logické díry“, že jsme zapomněli formulovat chování systému pro některé méně obvyklé (ale reálně se vyskytující) situace.

Praxe nám obvykle dokáže, že „nemožné“ se až příliš často stává skutečností a to v těch nejméně vhodných situacích. Výskyt mnohých poruch a havárií je zdánlivě zcela nepředvídatelný, jejich mechanismus je mnohdy kuriózní a zdánlivě se vymyká „zdravému rozumu“ (do té doby, než nastanou – pak již jsou logicky vysvětlitelné a ukáže se, že mohly být v předstihu předpovězeny). Při formulování slovního zadání, souboru pravidel nebo logických výrazů se obvykle zaměříme na popis žádoucí činnosti systému v obvyklých pracovních situacích opomeneme, že se mohou vyskytnout neobvyklé, mezní nebo havarijní situace. V nich je správná činnost řídicího systému klíčová pro odvrácení zráta a havárií – a on právě v nich selhává, přestává řídit nebo vydává chybné povely v důsledku našeho neúplného nebo chybného popisu.

Nutnost vyplnit mapu pravidel nás nutí zamyslet se nad významem každého políčka, nad jemu odpovídající situací a zodpovědně jej vyplnit. Vyplatí se vyplnit i políčka, odpovídající zdánlivě nesmyslným a fyzikálně nedosažitelným situacím – třeba tak, že pro tyto nestandardní stavy doplníme nové diagnostické proměnné, indikující vzniklou závadu. Dále je třeba si uvědomit, že o stavu reálného procesu je řídicí nebo diagnostický systém informován prostřednictvím čidel, která také mohou selhat. Rozpoznání některého z neobvyklých, mezních nebo „fyzikálně nedostupných“ stavů může být upozorněním na závadu v měření nebo v komunikaci.

Prázdné políčko v mapě pravidel může mít několik důvodů a může být různě interpretováno (námi i programem). Stejně jako při vyplňování pravdivostních map a tabulek v booleovských systémech, je i ve smíšených systémech obvyklé (a výhodné) přijmout úmluvu, že prázdnému políčku je přiřazena nulová nebo jiná implicitně stanovená hodnota, případně jiné standardní ošetření, např. „nic nedělej“, „opakuji dosavadní akci“, „indikuj nepředvídaný stav a riziko chyby“, „zastav stroj a indikuj poruchu“. Prázdnému políčku je tedy cílevědomě přiřazen určitý implicitní obsah, který jen není do políčka vepsán (z důvodu přehlednosti nebo úspory práce). Pokud existuje několik možných interpretací pro prázdná políčka, je třeba je rozlišit dohodnutými symboly, např. „X“ pro paměťovou funkci („opakuji dosavadní akci“), „E“ pro jednotné ošetření chybových stavů („indikuj nepředvídaný stav a riziko chyby“, „zastav stroj a indikuj poruchu“) a nevyplněné políčko pro nulovou