Modelovacie formalizmy udalostných systémov

Gabriel Juhás

Názov: Modelovacie formalizmy udalostných systémov

Autor: Gabriel Juhás

Vydalo vydavateľstvo:
RT systems s.r.o., Kopčianska 14, 851 01 Bratislava, v roku 2011

Vytlačili tlačiarne:
ZEPHIROS, a.s., Komárňanská 91, 821 09 Bratislava, v roku 2011

Prvé vydanie

ISBN 978-80-970519-1-4

ISBN 978-80-970519-1-4

9 788097 051914

Obsah

Predslov

Tieto skriptá sú venované popisu základných formalizmov udalostných systémov, známych v literatúre aj ako dynamické systémy diskrétnych udalostí (Discrete Event Dynamic Systems - DEDS). Svojim rozsahom predstavujú nevyhnutný základ pre predmet Modelovanie a simulácia udalostných systémov v 1. ročníku inžinierskeho štúdia študijného programu Aplikovaná informatika na Fakulte elektrotechniky a informatiky Slovenskej technickej univerzity v Bratislave. Na tento základ nadväzujú ako študijná literatúra v uvedenom predmete a v nadväznom predmete Analýza a syntéza udalostných systémov aktuálne vedecké články, ktoré prezentované formalizmy, najmä rôzne rozšírenia Petriho sietí, študujú do výraznejšej hĺbky.

Úvodná kapitola prezentuje stručný prehľad zo všeobecnej teórie systémov, obsahuje zatriedenie DEDS v kontexte s uvedenými poznatkami všeobecnej teórie systémov a stručne zhŕňa obsah nasledujúcich kapitol.

Kapitoly 1-4 sú venované popisu vybraných formalizmov pre logické modelovanie DEDS a ich vzťahu k automatom. Kapitola 1 je venovaná formálnym jazykom a automatom, kapitola 2 Petriho sieťam a farebným Petriho sieťam, kapitola 3 prezentuje definíciu zjednodušenej verzie formalizmu Statechart - tzv. hierarchické automaty (hierarchické stavové stroje) a kapitola 4 popisuje možnosť modelovania DEDS pomocou predikátov na stavovej množine a predikátových transformerov.

Kapitola 5 je venovaná príkladom rozšírení formalizmov udalostných systémov. Vychádza zo štúdia literatúry o systémoch diskrétnych udalostí a ich modelovacích prostriedkoch popísaných v kapitolách 1-4. Prezentuje dva typy rozšírení opisu a modelovania DEDS. Prvá časť tejto kapitoly obsahuje formálnu definíciu DEDS na úrovni stavového opisu spolu so základmi relačného počtu pre modelovanie DEDS. V druhej časti sú načrtnuté možnosti využitia poznatkov kapitoly 2, vedúce k tzv. algebraicky zovšeobecneným Petriho sieťam.

Úvod

V súčasnej dobe sa čoraz naliehavejšie objavuje potreba účinných metód analýzy a syntézy systémov. Pojem systém sa spravidla používa na označenie súboru rôznych častí, určitým spôsobom medzi sebou previäzbených tak, aby tvorili celok a plnili nejakú funkciu.

História obsahu pojmu (všeobecný) systém sa datuje už z čias starých Grékov, keď Aristoleles vyslovil jeho slávny a stále platný výrok, že celok je viac ako súhrn jeho častí, cez, v istom zmysle z pohľadu teórie systémov kontraproduktívny elementaristický prístup, ktorý formoval základ vedeckých metód ešte v 19. storočí, po modernú dobu, keď pojem systému formulovaný rakúskym biológom Ludwigom von Bertalnffy (1928) získal obecnú reputáciu. Pojem systém vychádzal z biológie, ale v rovnakom čase sa objavoval (alebo závery vedúce k nemu) i v iných vedeckých disciplínach. Tento fakt viedol k vytvoreniu novej vednej oblasti nazvanej všeobecná teória systémov (Bertalanffy, 1950, 1968), ktorá sa sústreďuje na určenie zákonitostí a vlastností spoločných pre systémy vo všeobecnosti. Trendy v tejto novej disciplíne sú orientované na vytvorenie matematickej teórie systémov ako i na problémy technológie (automatizácie a zavádzania výpočtovej techniky) a na problémy humanitných disciplín (ekonómia, ekológia, sociológia) a v neposlednom rade na filozofické aspekty teórie systémov. Diskusia o týchto aspektoch a interdisciplinárny charakter teórie systémov v jej najvšeobecnejšom ponímaní môže viesť smerom k unifikácii zdanlivo protichodných oblastí ľudského bádania ako sú prírodné a technické vedy na jednej strane a humanitné vedy na strane druhej.

Predsa však, jednou z neoddeliteľných súčastí systémovej vedy je vybudovať matematické nástroje a metódy umožňujúce opis systémov v ich obecnej povahe. Existujú rôzne druhy teórií (z pohľadu zovšeobecnenia) - od špecifických teórií s vysokou aplikovateľnosťou v špecializovaných oblastiach a nízkou úrovňou všeobecnosti až po matematické teórie abstraktných systémov s vysokou úrovňou všeobecnosti, ktoré sú často príliš všeobecné pre produkciu použiteľnej metodológie pre špecializované oblasti, a preto slúžia zväčša ako návod.

Jednou z rozšírených teórií je prístup vybudovaný na práci Klira (1969) (napr. Orchard 1972; Kotek a kol., 1990). Prístup je založený na definovaní základných rysov systémov, ktoré sú použité pri piatich základných definíciach systémov, ktorých kombináciou je možné získať ľubovoľnú definíciu systému. Predpokladá sa, že systém je definovaný na objekte. Stručne, prístup je vybudovaný nad množinou pozorovaných veličín $X = \{x_1, .., x_n\}$ s hodnotami v príslušných množinách X_i, ktoré spolu s časovou množinou T tvoria úroveň rozlíšenia. Takto, množina stavov, t.j. množina n-tíc $M = \{x_1(t),..,x_n(t) \mid t \in T \wedge \forall\ i \in \{1, .., n\}: x_i \in X_i\}$ reprezentuje aktivitu systému. Na určenie vzťahov medzi pozorovanými veličinami sú použité takzvané vzorkované premenné, ktoré sú definované posunutím pozorovaných veličín v čase a vytvárajú tzv. masku na matici aktivít (napr. pre pohybujúci sa objekt, kde pozorované veličiny sú rýchlosť a zrýchlenie pri uvažovaní diskrétnej úrovne rozlíšenia času vzorkovanými premennými môžu byť okamžitá a posledná minulá hodnota rýchlosti). Množina okamžitých hodnôt vzorkovaných premenných postačujúca k určeniu správania systému predstavuje stav systému. Správanie sa systému je definované ako časovo invariantná relácia na kartezskom súčine množín možných hodnôt vzorkovaných premenných. Aby bolo umožnené generovať rovnaké dáta ako generuje aktivita, je definovaná stavovo-prechodová relácia na stavovej množine. Takzvaná UC-štruktúra pojednáva o dekompozícii systému na subsystémy spojené väzbami (ktoré sú tvorené spoločnými premennými subsystémov).

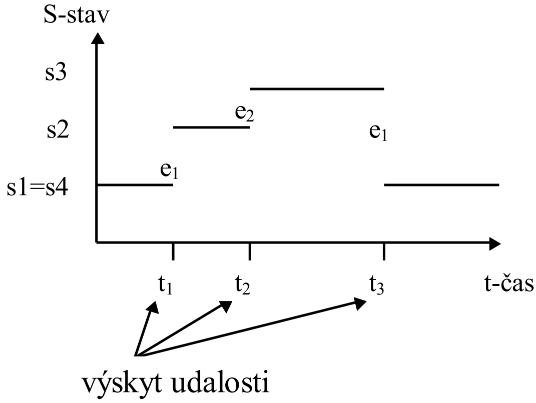
Mesarovic v (1972) definuje systém ako reláciu na abstraktnej množine (podobne s pojmom správania v prístupe Klira (1969)), ďalej špecifikovaný na všeobecný časový systém (ktorý je analogický s aktivitou z (Klir, 1969)). Abstraktný dynamický systém definovaný Mesarovicom je časový systém s danou abstraktnou stavovou množinou a dvoma funkciami: s výstupnou funkciou, ktorá určuje výstup pre každý stav, vstupnú funkciu a čas; a so stavovo-prechodovou funkciou, ktorá určuje pre každý stav v čase t, vstupnú funkciu a čas t' stav v čase t'. Tieto funkcie sú analogické so správaním a stavovo-prechodovou reláciou danou v (Klir 1969).

Jedným z ďalších prístupom je Waymorova teória (Waymore, 1972). Waymore definuje systém ako 6-ticu pozostávajúcu zo stavovej množiny, vstupov, vstupných funkcií (tieto tri sú zhodné s (Mesarovic, 1972)), správania definovaného ako podmnožinu funkcií definovaných na stavovej množine s hodnotami opäť v stavovej množine (Waymorove správanie je odlišné od Klirovho konceptu správania daného v (Klir, 1969), časovej škály a pohybu - čo je funkcia, určujúca správanie pre každú vstupnú funkciu a špecifikovaný čas. Rozdiel medzi pohybom z (Waymore, 1972) a stavovo-prechodovou funkciou z (Mesarovic, 1972) je iba formálny - ich význam je rovnaký.

Ak porovnáme popísané prístupy, získame fakt, že hoci sú formálne rozličné, principiálne sú totožné. Porovnanie taktiež adresuje hlavné problémy všeobecnej teórie systémov - dekompozíciu alebo decentralizáciu, hierarchiu a diskrétnu a spojitú povahu systémov. Taktiež môžeme rozoznať rozdiely v zdôraznení určitých čŕt: určenie pozorovaných veličín pomocou masky vzorkovaných premenných v Klirovom prístupe, dôraz na orientáciu systému v práci Mesarovica (vstupy-výstupy), problémy prípustnosti vstupných funkcií (posunutie a segmentácia) v práci (Waymore, 1972).

Dôležitou otázkou pri definovaní systému je otázka určenia stavu. Heuristická definícia pre dynamický orientovaný deterministický systém definuje stav ako vektor, ktorého hodnota poskytuje úplnú informáciu o histórii systému, ktorú pri známej hodnote vstupu potrebujeme pre stanovenie ďalšieho vývoja systému (Kotek a kol., 1990; Zadeh, 1964). Inými slovami, stavové veličiny majú poskytovať tú minimálnu informáciu o vnútornom stave systému, ktorá je potrebná, aby pomocou nej bolo možné pri známom časovom priebehu vstupných veličín a známych dynamických vlastnostiach systému jednoznačne určiť priebehy výstupných veličín. Zároveň, znalosť hodnôt stavových veličín v určitom okamihu a znalosť nasledujúcich priebehov vstupných veličín a dynamických vlastností systému má byť postačujúca na jednoznačné určenie nasledujúcich stavov (Gvozdiak a kol., 1990).

Dynamické systémy môžu byť rozdelené na spojité a diskrétne. Presnejšie, môžeme uvažovať triedu dynamických systémov spojitých premenných (CVDS), ktoré sú zvyčajne popísané diferenciálnymi rovnicami. Prechod od stavovo-prechodovej funkcie k systému diferenciálnych rovníc je možné nájsť napr. v (Kotek a kol., 1990). Prechod spočíva v rozvoji diferencovateľnej stavovo-prechodovej funkcie do Taylorovho radu, limitnom prechode časovej zmeny k nule a v zanedbaní zvyšku Taylorovho radu (chyba 2. rádu). Trieda CVDSs môže byť ďalej rozdelená podľa úrovne rozlíšenia času na triedu časovo-spojitých CVDSs a triedu časovo-diskrétnych CVDSs, ktorá môže byť popísaná príslušnými diferenčnými rovnicami. Na druhej strane, môžeme uvažovať triedu dynamických systémov diskrétnych udalostí (DEDS) (Ho, 1992; Balemi a kol., 1993), ktoré sú diskrétne vo svojej podstate nezávisle od úrovne časového rozlíšenia. Inými slovami, DEDS sú charakterizované po častiach konštantnými stavovými trajektóriami (obr. 1), keď okamžitý stav je diskrétne menený výskytom diskrétnych udalostí, t j. DEDS sú systémy, v ktorých stavové zmeny nastávajú v dôsledku výskytu diskrétnych udalostí - výskyt udalosti vedie k zmene stavu DEDS - t.j. skokom sa zruší starý stav a systém prechádza do stavu nového. Mnohé oblasti a rôzne aspekty správania sa DEDS vedú k rozvoju variety modelov DEDS. Logické modely DEDS sú charakterizované ignorovaním času výskytu udalostí a uvažovaním iba poradia ich výskytu. Takéto zjednodušenie sa používa pri štúdiu vlastností logickej dynamiky systému.



Obrázok 1: Typická stavová trajektória DEDS.

Aplikačná oblasť systémov diskrétnych udalostí a teda použitia ich modelovacích prostriedkov je veľmi široká: spomeňme za všetky aspoň databázové systémy (Lafortune 1988) a pružné výrobné systémy (Maion a Tadmore, 1986).

Podľa (Brave a Heymann, 1993) najjednoduchším formálnym logickým modelovacím prostriedkom DEDS sú konečné automaty pričom ich správanie môže byť popísané formálnym jazykom generovaným modelujúcim automatom (množinou možných postupností symbolov udalostí (t. j. symbolov vstupnej abecedy), ktoré môžu byť generované automatom). Ak porovnáme štruktúru automatu so štruktúrou dynamického systému napr. z (Mesarovic, 1972) alebo (Kotek a kol., 1990), (Gvozdiak a kol., 1990), zistíme, že sa líšia iba uvažovaním resp. neuvažovaním času.

Táto práca je venovaná niektorým aspektom vybraných logických modelov DEDS s dôrazom na vzťah jednotlivých formalizmov a automatov. Konkrétne, kapitola 1 je venovaná definícii formálnych jazykov (Chomsky, 1966; Hladký, 1975; Preparata a Yeh, 1974; Molnár a kol., 1987; Kotek a kol., 1990) a automatov (Ajzerman a kol., 1963; Birkhoff a Bartee, 1970; Gvozdiak a kol., 1990). Kapitola pojednáva o jazykoch z pohľadu syntaxe, zaoberá sa jazykom, ktorý je chápaný ako množina konečných postupností prvkov ľubovoľnej množiny nazvanej abeceda. Definuje gramatiky ako štruktúry umožňujúce pomocou tzv. prepisovacích pravidiel generovať jazyk. Gramatiky môžu byť podľa typov prepisovacích pravidiel zatriedené do tzv. Chomského hierarchie. Ďalej obsahuje kapitola 1 časť venovanú automatom, sú uvedené definícia nedeterministického automatu, špecifikácia deterministického automatu a konečného automatu. Vo vzťahu k automatom sú rozlišované jazyky generované automatom (t. j.množiny realizovateľných postupností symbolov vstupnej abecedy - prefixovo uzavretý jazyk) a jazyky rozpoznávané automatom (t. j. množiny realizovateľných postupností symbolov vstupnej abecedy končiace v terminálnom stave automatu). Kapitola sa ďalej zaoberá vzťahom konečných automatov a regulárnych jazykov.

Existuje množstvo ďalších formalizmov pre modelovanie DEDS. Jedným z najrozšírenejších sú Petriho siete (Petri, 1962; Peterson, 1981; Reisig, 1985; Murata, 1989) a ich modifikácie, najmä tzv. farebné Petriho siete (Jensen, 1986), ktorým je venovaná kapitola 2. Hlavnými výhodami Petriho sietí sú možnosť grafického vyjadrenia štruktúry modelovaného systému s paralelnými aktivitami a účinného matematického nástroja v podobe algebraických rovníc. Prvá časť kapitoly 2 je venovaná algebraickému pozadiu potrebnému pre ďalšie časti týkajúce sa Petriho sietí. Druhá časť kapitoly 2 ukazuje spôsob akým je možné špecifikáciou stavovej množiny a prechodovej funkcie automatu získať štruktúru poskytujúcu výrazný algebraický aparát pre analýzu modelovaného systému - Petriho sieť. Tretia časť kapitoly 2 je venovaná formálnej definícii Petriho sietí. Petriho sieť môže byť interpretovaná ako bipartitný orientovaný graf, kde jeden typ vrcholov (tzv. pozície) reprezentuje stavové veličiny systému, a druhý typ reprezentuje udalosti (tzv. prechody). Vrcholy sú spojené orientovanými hranami, pričom hrany môžu spájať iba rôzne typy vrcholov. Výskyt udalosti (spustenie prechodu) zmení stavy pozícií - t.j. v súlade s celočíselnými váhami a smerom hrán je z každej pozície pridaný resp. odobraný príslušný počet značiek. Štruktúra grafu môže byť opísaná celočíselnou maticou nazývanou incidenčná matica a označenou **C**, s počtom riadkov a stĺpcov rovným počtu pozícií a prechodov. Potom dynamika siete (zmena stavu - t. j. zmena počtu značiek v pozíciách) je daná tzv. stavovou rovnicou **m**' = **m** + **C** . **Y**, kde vektory **m** a **m**' reprezentujú značkovanie (počet značiek) v jednotlivých pozíciách pred a po spustení postupnosti prechodov s počtom výskytu jednotlivých prechodov určeným vektorom **Y**. Štvrtá časť kapitoly 2 sa zaoberá rozšírením Petriho sietí na tzv. farebné Petriho siete. Stručne, koncept farebných Petriho sietí spočíva v možnosti použiť rôzne typy (rôzne farby) značiek. Takto prechody môžu byť otvárané pre rôzny typ značiek. To umožňuje modelovať niektoré systémy efektívnejšie v porovnaní so štandardnými Petriho sieťami. Posledná časť kapitoly 2 obsahuje jednoduchý príklad (dopravné semafory) ilustrujúci použitie Petriho sietí a ich rozšírenia farebných Petriho sietí. Ako matematický nástroj pre konštrukciu farebných Petriho sietí (a Petriho sietí) sú použité multimnožiny a multimnožinové rozšírenia funkcií.

V mnohých aplikačných oblastiach DEDS pozostáva z veľkého počtu komponentov, čo spôsobuje exponenciálny rast počtu stavov. Potreba redukcie počtu stavov viedla k definícii formalizmu Statechart (Harel, 1987; Harel a kol., 1987), ktorého zjednodušená forma tzv. hierarchický automat (hierarchický stavový stroj) (HSM) (Brave a Heymann, 1993) je popísaná v kapitole 3. Hierarchia stavov je formálne vyjadrená reláciou hierarchie, ktorá každému stavu okrem koreňového stavu určuje bezprostredný nadstav (ktorému je daný stav bezprostredným podstavom). Každý stav HSM je buď AND-stavom (čo znamená, že ak HSM je v danom AND-stave, potom je súčasne vo všetkých jeho bezprostredných podstavoch súčasne), OR-stavom (ak HSM je v danom OR-stave, potom je práve v jednom z jeho bezprostredných podstavov) alebo základným stavom (ktorý nemá žiadne podstavy). Takto, konfigurácia stavu a je, neformálne, daná množinou jeho podstavov, v ktorých HMS môže byť súčasne. Konfigurácia q je úplná, ak pre každý nadstav b stavu a platí, že najnižší spoločný nadstav q a b nie je AND-stavom. Potom je pre každý stav definovaná prechodová funkcia priraďujúca úplným konfiguráciám a udalostiam podmnožinu množiny jeho úplných konfigurácií. HSM je teda ekvivalentný s automatom (bez výstupnej funkcie) so stavovou množinou a prechodovou funkciou danou množinou úplných konfigurácií a prechodovou funkciou koreňového stavu HSM.

Kapitola 4 sa venuje popisu DEDS správania pomocou predikátov a predikátových transformerov, pričom predpokladá DEDS modelovaný automatom. Predikátový opis poskytuje možnosť využitia vlastností systému pri popise správania a takto opäť umožňuje eliminovať stavovú explóziu v rozsiahlych aplikáciách DEDS. Vychádzajúc z (Dijkstra, 1976; Dijkstra a Scholten, 1990; Kumar a kol., 1993) popisuje kapitola 4, že predikáty definované na stavovej množine automatu môžu byť asociované s podmnožinami stavovej množiny, na ktorých sú platné. Taktiež je možné ukázať, že negácia, konjunkcia a disjunkcia predikátov môže byť asociovaná s doplnkom, prienikom a zjednotením množín. Analogicky s reláciou určujúcou, či nejaká množina je podmnožinou inej (alebo či množina obsahuje inú) môže byť definovaná relácia určujúca, či nejaký predikát je silnejší ako iný (alebo či nejaký predikát je slabší ako iný). Ďalej sú definované predikátové transformery, t.j. funkcie priraďujúce každému predikátu definovanému na stavovej množine opäť predikát na stavovej množine. Takto, predikátové transformery nazvané najsilnejšia post-podmienka a najslabšia pred-podmienka sú definované pre každú udalosť automatu ako transformery, ktoré priraďujú danému predikátu predikát platný na množine stavov dosiahnutých spustením danej udalosti resp. danému predikátu *p* predikát *p'* platný na množine stavov pre ktoré výskyt udalosti nie je definovaný alebo vedie do stavu v ktorom je platný predikát *p*. Stavová množina automatu môže byť teda ekvivalentne modelovaná množinou predikátov definovaných na nej a prechodová funkcia môže byť zamenená najsilnejšou post-podmienkou. Takýmto spôsobom je možné použiť pri modelovaní DEDS priamo jeho vlastnosti.

Kapitola 5 je venovaná vybraným príkladom rozšírení základných formalizmov.

Podkapitolu 5.1 tvoria dve logické časti. Prvá z týchto častí vychádza z článku (Juhás, 96a). DEDS je v nej formálne definovaný ako systém diskrétnych stavových zmien (Discrete State Change System - DSCS) vychádzajúc z vnútorného popisu systému v kontexte s poznatkami všeobecnej teórie systémov. Taktiež je formalizovaná podmienka logického vnútorného opisu DSCS (DEDS). Druhá časť prezentuje základy relačného kalkulu pre logické modelovanie DSCS zavedeného v článkoch (Juhás a Kocian, 1996, 1995a), ktoré sú prispôsobené formálnej definícii DSCS. Relačný kalkul pre DEDS vychádza z myšlienky využitia stavovej zmeny v istom zmysle podobne ako CVDS využíva diferenciál. Z pohľadu predikátového počtu možno chápať prezentovaný relačný kalkul ako snahu vyjadriť dynamiku DEDS (detailnú, možnú, požadovanú atď) množinou predikátov na karteziánskom súčine S x S (kde S predstavuje stavovú množinu systému), t.j. vyjadriť dynamiku DEDS pomocou vlastností stavových zmien.

Podkapitola 5.2 prezentuje rozšírenie už existujúceho formalizmu, t.j. Petriho sietí. Zaoberá sa možnosťou algebraického zovšeobecnenia Petriho sietí a farebných Petriho sietí. Vychádza z článkov (Juhás, 1996b,c) a nadväzuje priamo na kapitolu 2. Uvedené zovšeobecnenie umožňuje použiť nielen objekty využívajúce nezápornú časť okruhu celých čísel, ale podmnožinu ľubovoľnej komutatívnej grupy, pričom zachováva základné výhody Petriho sietí. Štandardné Petriho siete môžu byť takto chápané ako špeciálny prípad uvedenej definície.

1 Formálne jazyky a automaty

1. 1 Formálne jazyky

Pod pojmom *jazyk* je možné (Kotek a kol., 1990; podrobnejšie napr. Hladký, 1975) chápať usporiadanú trojicu (L, M, σ), kde L je *množina výrazov* jazyka, M je *množina významov* jazyka a $\sigma \subseteq L \times M$ je relácia nazývaná *sémantika*, priraďujúca jednotlivým výrazom významy. Výrazy ľubovoľnej množiny $X \subseteq L$ takej, že $\exists\ m \in M: \forall\ x \in X: [x, m] \in \sigma$, t.j. výrazy s rovnakým významom, sa nazývajú synonymá. Významy ľubovoľnej množiny $Y \subseteq M$ takej, že $\exists\ l \in L: \forall\ y \in Y: [l, y] \in \sigma$, t.j. významy toho istého výrazu sa nazývajú homonymá. V ďalšom texte sa nebudeme zaoberať sémantikou jazyka, ale iba štúdiom formálnych vlastností množiny výrazov L, t.j. syntaxou jazyka.

Nech A je ľubovoľná množina. Nech množina A^* je množina všetkých konečných postupností prvkov množiny A obsahujúca prázdnu postupnosť označenú ε. Definujme binárnu operáciu na množine A^* označenú ∘ a nazvanú *zreťazenie* nasledovne:

$\circ: A^* \times A^* \to A^*$ taká, že $\forall\ a = \{a_i\}_{i=1}^{k}, b = \{b_i\}_{i=1}^{n} \in A^*: a \circ b = \{x_i\}_{i=1}^{k+n}$, pričom $\forall\ i \leq k: x_i = a_i \wedge \forall\ i > k: x_i = b_{i-k}$.

Tak napr. pre A = {a, b, c, d, e}, *a* = abc a *b* = dea, $a \circ b$ = abcdea.

Je zrejmé, že (A^*, ∘) je monoidom, hovoríme, že je monoidom konečných postupností nad množinou A.

V ďalšom texte budeme niekedy vynechávať operátor zreťazenia, t.j. namiesto $a \circ b$ budeme písať iba *ab*.

Formálnym jazykom L nad abecedou (ľubovoľnou konečnou množinou) V je nazývaná ľubovoľná podmnožina množiny V^*, t.j. $L \subseteq V^*$. Reťazce $v \in L$ sú nazývané *slová* (*výrazy*) jazyka L. Takto napr. abecedou jazyka chemických výrazov sú značky chemických prvkov, čísla a špeciálne symboly, slovami (výrazmi) jazyka chemických výrazov sú chemické vzorce, abecedou prirodzeného jazyka je množina jeho „slov“ nad ktorou je vytvorený prirodzený jazyk, ktorého slovami (výrazmi) sú „vety“.

Prepisovacím systémom nazývame usporiadanú dvojicu (V, P), kde:

- V je abeceda;
- $P \subseteq V^* \times V^*$ je relácia nazývaná *množina prepisovacích pravidiel* (prepisovacie pravidlá, t.j. prvky $[u, v] \in P$ označujeme tiež $u \to v$).

Nech *relácia priameho generovania* $\Rightarrow \subseteq V^* \times V^*$ je taká, že pre všetky $u, v \in V^*$ platí: $[u, v] \in \Rightarrow$ (t.j. $u \Rightarrow v$) práve vtedy, keď existujú $a, b, x, y \in V^*$ také, že $u = a \circ x \circ b \wedge v = a \circ y \circ b \wedge x \to y$,

a nech *relácia generovania* $\underset{*}{\Rightarrow} \subseteq V^* \times V^*$ je taká, že pre všetky $u, v \in V^*$ platí: $[u, v] \in \underset{*}{\Rightarrow}$ (t.j. $u \underset{*}{\Rightarrow} v$) práve vtedy, keď existuje postupnosť $\{u_i\}_{i=1}^{n}$ taká, že $u = u_1 \wedge v = u_n \wedge (\forall\, i \in \{1, \dots, n-1\}: u_i \underset{*}{\Rightarrow} u_{i+1})$.

Ak $u \Rightarrow v$ hovoríme, že reťazec u *priamo generuje* reťazec v v prepisovacom systéme (V, P). Ak $u \underset{*}{\Rightarrow} v$ potom hovoríme, že výraz u *generuje* výraz v v prepisovacom systéme (V, P).

Definicia 1.1.1: *Formálna gramatika* F je štvorica $F = \{V_N, V_T, P, S\}$, kde:

- V_N je *abeceda neterminálnych symbolov* (t. j. množina premenných);
- V_T je *abeceda terminálnych symbolov*;
- $((V_N \cup V_T), P)$ je prepisovací systém s konečnou množinou prepisovacích pravidiel *P* takou, že:

  $\forall\, [u,v] \in P: \exists\, X \in V_N\ \exists\, a,b \in (V_N \cup V_T)^* : u = a \circ X \circ b$ (t.j. u obsahuje $X \in V_N$);

- S je *začiatočný symbol*, $S \in V_N$.

Definícia 1.1.2: Nech $F = \{V_N, V_T, P, S\}$ je formálna gramatika a $\underset{*}{\Rightarrow}$ označuje reláciu generovania v prepisovacom systéme $(V_N \cup V_T, P)$. Potom jazyk $L(F) = \{u \mid S \underset{*}{\Rightarrow} u \wedge u \in V_T^{*}\}$ je nazývaný *jazyk generovaný gramatikou F*.

Všeobecne jazyk generovaný ľubovoľnou gramatikou je nazývaný *rekurzívne vyčísliteľný jazyk* (dôvod pozri napr. v (Birkhoff a Bartee, 1970)).

Jazyk L(F) generovaný gramatikou F je teda tvorený reťazcami z V_T, ktoré získame prepisovaním symbolov z V_N (dosádzaním do premenných z V_N) podľa prepisovacích pravidiel počnúc prepisovaním symbolu S.

Nasledujúci príklad ilustruje generovanie vety anglického jazyka „the man eats the apple“.

Príklad 1.1.1: Nech $F_1 = (V_N, V_T, P, S)$ je gramatika, kde

V_N = {veta, menná_časť, slovesná_časť, určitý_člen, podstatné_meno, sloveso};

V_T = {THE, MAN, APPLE, EATS};

P = { veta → menná_časť slovesná_časť,

menná_časť → určitý_člen podstatné_meno,

slovesná_časť → sloveso menná_časť,

určitý_člen → THE

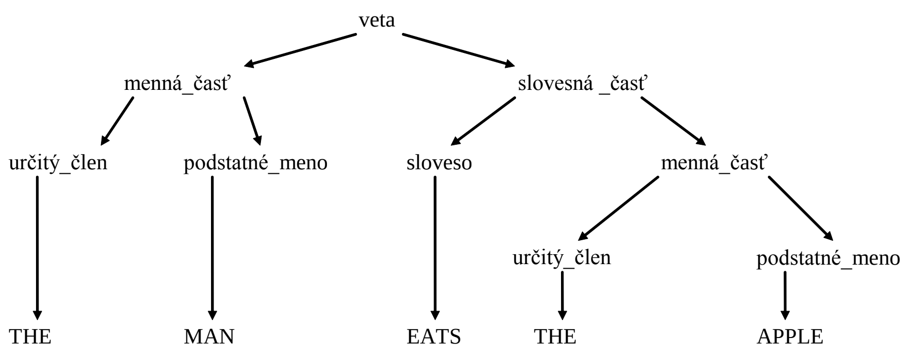
podstatné_meno → APPLE

podstatné_meno → MAN

sloveso $\rightarrow$ EATS};

S = veta.

Uvedenú vetu je možné získať postupným prepisovaním (dosadzovaním) do pravidiel z P znázornenom na obr. 1.1.1.



Obrázok 1.1.1: Derivačný strom popisujúci generovanie anglickej vety „THE MAN EATS THE APPLE“.

Príklad 1.1.2: Nech $F_2 = (V_N, V_T, P, S)$ je gramatika, kde

$V_N = \{A\}$;

$V_T = \{a, b\}$;

$P = \{A \rightarrow b, A \rightarrow aA, A \rightarrow bA\}$;

S = A.

Potom jazyk $L(F_2)$ generovaný gramatikou F_2 sa skladá zo všetkých postupností písmen a, b, ktoré končia písmenom b.

Podľa tvaru prepisovacích pravidiel Chomsky (1966) rozdelil gramatiky do niekoľkých tried.

Definícia 1.1.3: Nech $F = (V_N, V_T, P, S)$ je gramatika.

- Ak platí $\forall\ u \rightarrow v$:

($\exists$ $a, b, x \in (V_N \cup V_T)^*$, $X \in V_N$: $u = aXb \wedge v = axb \wedge x \neq \varepsilon) \vee (u = S \wedge v = \varepsilon)$,

potom gramatiku F nazývame *kontextová gramatika* a jazyk L(F) *kontextový jazyk.*

- Ak platí $\forall\ u \to v$:

  ($\exists$ $x \in (V_N \cup V_T)^*$, $X \in V_N$: $u = X \wedge v = x \wedge x \neq \varepsilon) \vee (u = S \wedge v = \varepsilon)$, potom gramatiku F nazývame *bezkontextová* a jazyk L(F) *bezkontextový jazyk.*

- Ak platí $\forall\ u \to v$:

  ($\exists$ $b \in V_T$, $X, Y \in V_N$: $(u = X \wedge v = b) \vee (u = X \wedge v = bY)) \vee (u = S \wedge v = \varepsilon)$,

  potom gramatiku F nazývame *regulárna* a jazyk L(F) *regulárny jazyk.*

Z predchádzajúcej definície je zrejmé, že pravidlá regulárnej gramatiky spĺňajú podmienky pre bezkontextové gramatiky a tie požiadavky pre kontextové gramatiky. Ak teda označíme množinu rekurzívne vyčísliteľných jazykov $\{L_0\}$, množinu kontextových jazykov $\{L_1\}$, množinu bezkontextových jazykov $\{L_2\}$ a množinu regulárnych jazykov $\{L_3\}$, potom platí $\{L_0\} \supset \{L_1\} \supset \{L_2\} \supset \{L_3\}$ (tzv. Chomského hierarchia).

Napr. jazyk z príkladu 1.1.2 je regulárnym jazykom.

Uveďme na záver definíciu pojmu prefix a definíciu prefixovo uzavretého jazyka.

Nech $v \in V^*$, kde V je ľubovoľná abeceda. *Prefixom* reťazca v nazývame každý reťazec $u \in V^*$ taký, že existuje $x \in V^*$: $ux = v$. *Prefixovo uzavretým jazykom* nazývame jazyk L, pre ktorý je splnená podmienka: $\forall\ v \in L$: $\forall\ u \in V^*$: ak u je prefixom v potom $u \in L$, t.j.

jazyk L, v ktorom pre každý reťazec *v* z L platí, že všetky jeho prefixy tiež patria do L.

1. 2 Automaty

Nech #(A) označuje kardinálne číslo ľubovoľnej množiny A (mohutnosť, početnosť).

Definícia 1.2.1: *Automat* je šestica SM = (S, E, Y, δ, ω, s_0), kde

a) S je ľubovoľná množina nazvaná *množina stavov*;

b) E je ľubovoľná konečná množina nazvaná *vstupná abeceda*;

c) Y je ľubovoľná konečná množina nazvaná *výstupná abeceda*;

d) δ: S x E $\rightarrow$ 2^S je *prechodová funkcia* definovaná na S x E priraďujúca každému stavu a vstupu množinu možných nasledujúcich stavov (2^S je množina všetkých podmnožín množiny S);

e) ω: S x E $\rightarrow$ 2^Y je *výstupná funkcia* automatu definovaná S x E priraďujúca každému stavu a vstupu množinu možných výstupov;

f) $s_0 \in S$ je *počiatočný stav* automatu.

Definícia 1.2.2: *Automat bez výstupnej funkcie* je štvorica SM = (S, E, δ, s_0), kde S, E, δ a s_0 sú dané bodmi a), b), d) a f) v predchádzajúcej definícii.

Automat možno graficky znázorniť ako orientovaný graf nazývaný *stavový diagram*, ktorého vrcholmi sú stavy automatu a orientované hrany predstavujú prechody medzi stavmi (t. j. každá dvojica stavov s, s' taká, že existuje e $\in$ E: s' $\in$ δ(s, e) je spojená hranou označenou dvojicou vstup e, množina výstupov ω(s, e)). Do počiatočného stavu vedie voľná šípka.

Automat je možné chápať ako zariadenie, štartujúce v stave s_0 vykonávajúce prechody medzi stavmi podľa prechodovej funkcie δ a generujúcej výstupy podľa výstupnej funkcie ω.

Definícia 1.2.3: Prázdna postupnosť vstupov je realizovateľná a stav s_0 je dosiahnuteľný realizáciou prázdnej postupnosti. Nech pre stav $s_0 \in S$ a konečnú neprázdnu postupnosť vstupov $e = \{u_i\}_{i=1}^{k}$ existuje postupnosť stavov $\{s_i\}_{i=1}^{k}$ taká, že platí

$$\forall\, i \in \{1 .. k\}: s_i \in \delta(s_{i-1}, u_i).$$

Potom postupnosť vstupov *e* sa nazýva *realizovateľná* z s_0 a stav s_k nazývame *dosiahnuteľný* z s_0 realizáciou postupnosti *e*.

Definícia 1.2.4: Množinu všetkých realizovateľných postupností vstupov (vrátane prázdnej postupnosti) budeme nazývať *jazyk* (vstupných symbolov) *generovaný automatom* SM a označovať $L_G(SM)$.

Je zrejmé, že $L_G(SM) \subseteq E^*$ a zároveň $L_G(SM)$ je prefixovo uzavretý jazyk.

Výstupná funkcia ω automatu SM sa nazýva *značkovacia* ak nezávisí od symbolov vstupnej abecedy a priraďuje každému stavu a vstupu jednoprvkovú množinu, t.j. ak platí:

- $\forall\, s \in S: \forall\, e, e' \in E: \omega(s, e) = \omega(s, e')$;
- $\forall\, [s, e] \in S \times E: \#(\omega(s, e)) = 1$.

Značkovaciu funkciu môžeme nahradiť *redukovanou značkovacou funkciou* ω': $S \rightarrow Y$ takou, že $\forall\, [s, e] \in S \times E: \omega(s, e) = \{\omega'(s)\}$.

Definícia 1.2.5: Nech automat SM = $(S, E, Y, \delta, \omega, s_0)$, ktorého množina výstupov je dvojprvková $Y = \{y_1, y_2\}$ a výstupná funkcia je značkovacia. Potom SM nazývame *akceptor*.

Akceptor môže byť tiež popísaný päticou SM = (S, E, δ, s_0, A), kde (S, E, δ, s_0) je konečný automat bez výstupnej funkcie a $A \subseteq S$ je množina tzv. *možných koncových stavov*.

Vzťah akceptora popísaného automatom s výstupnou značkovacou funkciou a automatom bez výstupnej funkcie je daný nasledovne: S, E, δ a s_0 sú totožné a $A = \{s \mid s \in S \wedge y_1 = \omega'(s)\}$.

Ak pre postupnosť vstupov e akceptora SM existuje stav s dosiahnuteľný realizáciou postupnosti *e* pre ktorý $\omega'(s) = y_1$ (t.j. $s \in A$), potom postupnosť *e* sa nazýva rozoznávaná akceptorom SM. Množinu všetkých rozoznávaných postupností vstupov budeme nazývať *jazyk* (vstupných symbolov) *rozoznávaný akceptorom* SM a označovať $L_R(SM)$.

Z definície 1.2.4 a 1.2.5 je zrejmé, že pre každý akceptor SM platí $L_R(SM) \subseteq L_G(SM)$. Navyše, ak A = S, potom $L_R(SM) = L_G(SM)$. V stavovom diagrame sú koncové stavy znázornené dvojitým kruhom.

1. 2. 1 Konečné automaty a ich vzťah k regulárnym jazykom

Definícia 1.2.1.1: Automat nazývame *konečným*, ak je konečná jeho množina stavov.

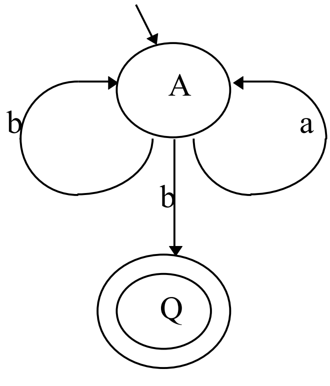
O vzťahu konečných automatov a formálnych jazykov hovorí tzv. Kleenova veta.

Veta 1.2.1.1 (Kleeneova veta): Jazyk je možné rozoznať konečným akceptorom vtedy a len vtedy ak je regulárny.

Ukážeme ako konštruovať ku regulárnej gramatike konečný akceptor, ktorý ju rozoznáva a naopak. Presný formálny dôkaz Kleenovej vety možno nájsť napr. v (Chytil, 1984).

Nech je daná regulárna gramatika $F = (V_N, V_T, P, S_F)$. Nech stavy konečného akceptora $SM = (S_{SM}, E, \delta, s_0, A)$, sú dané množinou V_N a nejakým symbolom Q takým, že $Q \notin V_N \wedge Q \in A$, t.j. $S_{SM} = V_N \cup \{Q\} \wedge Q \notin V_N$; nech množina vstupných symbolov je daná množinou V_T, t.j. $E = V_T$; nech $\forall X \in V_N, \forall b \in V_T,: Q \in \delta(X, b) \Leftrightarrow X \rightarrow b$; nech $\forall X, Y \in V_N, \forall b \in V_T: Y \in \delta(X, b) \Leftrightarrow X \rightarrow bY$; a nech platí $\forall b \in V_T: \delta(Q, b) = \emptyset$; nech s_0 je daný symbolom S_F, t.j. $s_0 = S_F$; a ak $S_F \rightarrow \varepsilon$ potom nech $A = \{S_F, Q\}$, inak $A = \{Q\}$. Potom konečný akceptor SM rozoznáva regulárny jazyk daný gramatikou F, t.j. $L_R(SM) = L(F)$.

Naopak, nech je daný konečný akceptor $SM = (S_{SM}, E, \delta, s_0, A)$. Nech gramatika $F = (V_N, V_T, P, S_F)$ je definovaná nasledovne: množina V_N je daná množinou S_{SM}, t.j. $V_N = S_{SM}$; množina V_T je daná množinou E, t.j. $V_T = E$; nech $\forall X \in S_{SM}, \forall Y \in A, \forall b \in E: X \rightarrow b \Leftrightarrow Y \in \delta(X, b)$; a nech $\forall X, Y \in S_{SM}, b \in E: X \rightarrow bY \Leftrightarrow Y \in \delta(X, b)$; a S_F je daný stavom s_0, t.j. $S_F = s_0$. Potom gramatika F generuje regulárny jazyk rozoznávaný akceptorom SM, t.j. $L(F) = L_R(SM)$.



Obrázok 1.2.1.1. Stavový diagram konečného akceptora rozoznávajúceho regulárny jazyk z príkladu 1.1.2.

Poznámka: Je možné ukázať, že bezkontextové jazyky sú rozoznateľné zásobníkovými automatmi (pozri napr. Hladký, 1975; Chytil, 1984; Birkhoff a Bartee, 1970) a rekurzívne vyčísliteľné jazyky sú rozoznateľné Turingovými strojmi (Hopcroft a Ullman, 1978).

1. 2. 2 Deterministické automaty

Ak pre automat SM platí

$$\exists\, [s, e] \in S \times E: \#(\delta(s, e)) > 1 \vee \#(\omega(s, e)) > 1,$$

potom SM je nedeterministický.

Ak pre automat SM platí

$$\forall\, [s, e] \in S \times E: \#(\delta(s, e)) \leq 1 \wedge \#(\omega(s, e)) \leq 1,$$

teda nový stav a výstup je určený „súčasným“ stavom a vstupom, potom SM je deterministický a definícia 1.2.1 môže byť prepísaná nasledovne.

Definícia 1.2.2.1: *Deterministický automat* je šestica SM = (S, E, Y, δ, ω, s_0), kde

- S je ľubovoľná množina stavov;
- E je ľubovoľná konečná množina nazvaná vstupná abeceda;
- Y je ľubovoľná konečná množina nazvaná výstupná abeceda;
- δ: S x E $\supseteq$ D(δ) $\rightarrow$ S je prechodová funkcia čiastočne definovaná na S x E s hodnotami v S (D(δ) je definičný obor δ);
- ω: S x E $\supseteq$ D(ω) $\rightarrow$ Y je výstupná funkcia čiastočne definovaná na S x E s hodnotami v Y (D(ω) je definičný obor ω);
- $s_0 \in$ S je počiatočný stav automatu.

Vzťah medzi prechodovou funkciou $\delta_{def1221}$ (resp. výstupnou funkciou $\omega_{def1221}$) deterministi/ckého automatu daného definíciou 1.2.2.1 a prechodovou funkciou δ_{def121} (resp. výstupnou funkciou ω_{def121}) toho istého automatu daného definíciou 1.2.1 je daný nasledovne:

1. $\forall$ [s,e] $\in$ S x E:

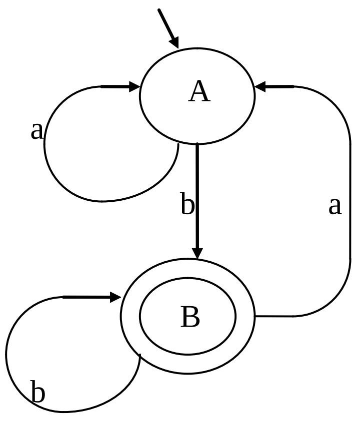
   $[s,e] \in D(\delta_{def1221}) \Leftrightarrow \#(\delta_{def121}(s, e)) = 1$

   (resp. $[s,e] \in D(\omega_{def1221}) \Leftrightarrow \#(\omega_{def121}(s, e)) = 1$);

2. $\forall [s,e] \in D(\delta_{def1221})$: $\delta_{def1221}(s, e) \in \delta_{def121}(s, e)$

   (resp. $\forall [s,e] \in D(\omega_{def1221})$: $\omega_{def1221}(s, e) \in \omega_{def121}(s, e)$);

Je možné dokázať (Hladký, 1975; Hopcroft a Ullman, 1978; Chytil, 1984), že ku každému nedeterministickému konečnému akceptoru existuje deterministický konečný akceptor, ktorý rozoznáva rovnaký regulárny jazyk.



Obrázok 1.2.2.1. Stavový diagram deterministického konečného akceptora rozoznávajúceho regulárny jazyk z príkladu 1.1.2.

Ak v ďalšom texte nešpecifikujeme, či ide o nedeterministický alebo deterministický automat, budeme predpokladať, že automat je deterministický vo forme danej definíciou 1.2.2.1.

2 Petriho siete

2. 1 Algebraické pozadie

Nech $\Gamma = (G, \oplus)$ je ľubovoľná grupa, nech Z označuje celé čísla a nech Z^+ označuje kladné celé čísla a Z^- záporné celé čísla. V nasledujúcom texte:

a) pre ľubovoľnú postupnosť $\{g_i\}_{i=1}^{j}$ ($j \in Z^+$) prvkov množiny G:

$$\sum_{i=1}^{j} {}^{\oplus} g_i = g_1 \oplus .. \oplus g_j \;;$$

b) $\forall\, g, h \in G$: $g -_{\oplus} h = g \oplus h^{-1}$, kde h^{-1} je inverzný prvok k prvku h v grupe Γ;

c) *skalárne násobenie* $\bullet_{\oplus} : Z \times G \to G$ alebo $\bullet_{\oplus} : G \times Z \to G$ prvkov ľubovoľnej grupy $\Gamma = (G, \oplus)$ označuje násobenie skalárom zo Z dané nasledovne:

1. $\forall\, j \in Z^+ : \forall\, g \in G : j \bullet_{\oplus} g = g \bullet_{\oplus} j = \sum_{i=1}^{j} {}^{\oplus} g_i$, kde $g_i = g$ pre všetky $i \in \{1, .., j\}$:;
2. pre $j = 0$: $\forall\, g \in G : 0 \bullet_{\oplus} g = g \bullet_{\oplus} 0 = \varepsilon$, kde ε je neutrálny prvok v grupe Γ;
3. $\forall\, j \in Z^- : \forall\, g \in G : j \bullet_{\oplus} g = g \bullet_{\oplus} j = \sum_{i=1}^{j} {}^{\oplus} g_i$, kde $g_i = g^{-1}$ pre všetky $i \in \{1, .. j\}$ (g^{-1} je inverzný prvok k prvku g v grupe Γ) .

Poznámka: Toto skalárne násobenie má vyššiu prioritu ako operácia $\oplus$.

Potom grupa $(G, \oplus)$ spolu so skalárnym násobením tvoria priestor (modul) nad okruhom $(Z, +, .)$, kde symboly + a . označujú štandardné sčítanie a násobenie celých čísel.

Nech ďalej pre ľubovoľnú komutatívnu grupu (G, $\oplus$) s daným skalárnym násobením pre n-rozmerný vektor $\mathbf{V} = |v_1 \ . \ . \ v_n|^t$ prvkov množiny G (t.j. $\forall$ i $\in$ {1, .. n}: $v_i \in$ G}) je definované pravé násobenie $\bullet_\oplus$ n-rozmerným celočíselným vektorom $\mathbf{X} = | \ x_1 \ . \ . \ x_n \ |^t$ dané nasledovne:

$$\bullet_\oplus : G^n \text{ x } Z^n \rightarrow G \text{ pričom } \bullet_\oplus(\mathbf{V}, \mathbf{X}) = \mathbf{V}^t \bullet_\oplus \mathbf{X} = w = \sum_{i=1}^{n}{}^{\oplus}\, v_i \bullet_\oplus x_i.$$

Nech A a B sú ľubovoľné neprázdne množiny. Potom [A $\rightarrow$ B] označuje množinu všetkých funkcií z A do B, formálne

$[A \rightarrow B] = \{f \mid f\colon A \rightarrow B\}$.

Zaveďme teraz pojem *multimnožina* nad neprázdnou množinou A ako funkciu b: A $\rightarrow Z^+ \cup \{0\}$. Intuitívne, multimnožina je množina, ktorá môže obsahovať násobný výskyt rovnakého prvku. Napríklad {a, b, d, b} je multimnožina nad množinou {a, b, c, d} a je formálne daná ako b(a) = 1, b(b) = 2, b(c) = 0 a b(d) = 1.

Takto každá konečná postupnosť $a = \{u_i\}_{i=1}^{k}$ prvkov z A určuje multimnožinu b_a: A $\rightarrow Z^+ \cup \{0\}$ tak, že $\forall$ a $\in$ A: $b_a(a) = \#(I_a)$, kde $I_a = \{i \mid i \in \{1 .. n\} \wedge u_i = a\}$ a $\#(I_a)$ je kardinálne číslo množiny I_a (mohutnosť, početnosť).

Množinu všetkých multimnožín nad A označujeme ako A_{MS}, formálne $A_{MS} = [A \rightarrow (Z^+ \cup \{0\})]$.

Nech pre ľubovoľné k $\in$ ($Z^+ \cup \{0\}$) multimnožina nad ľubovoľnou množinou A označená $const_k$ je daná nasledovne:

$\forall$ a $\in$ A: $const_k(a) = k$.

Nech (G, $\oplus$) je komutatívna grupa. Funkcia $f \in$ [A $\rightarrow$ G] môže byť rozšírená jednoznačne na lineárnu funkciu $\hat{f} \in [A_{MS} \rightarrow G]$ nazvanú *multimnožinové rozšírenie f* nasledovne:

$$\forall\, b \in A_{MS}\colon\ \hat{f}(b) = \sum_{a \in A}{}^{\oplus} b(a) \bullet_\oplus f(a)\ .$$

Množina všetkých lineárnych funkcií z A_{MS} do G je označená ako $[A_{MS} \to G]_L$, formálne $[A_{MS} \to G]_L = \{ \hat{f} \mid \hat{f} \in [A_{MS} \to G] \wedge$ $(\exists f \in [A \to G]: \forall b \in A_{MS}: \hat{f}(b) = \sum_{a \in A}^{\oplus} b(a) \bullet_{\oplus} f(a))\}$.

Nech A a B sú ľubovoľné neprázdne množiny. Ak A je konečná k-prvková množina ($k \in (Z^{+} \cup \{0\})$) a jej prvky sú indexované ($A = \{a_1, .., a_k\}$), potom pre ľubovoľnú funkciu f: $A \to B$ **f** bude označovať vektor jej funkčných hodnôt: $\mathbf{f} = | f(a_1) \,.\,.\, f(a_k) |^t$.

Nech *CGROUPS* označuje množinu všetkých komutatívnych grúp. Nech A je ľubovoľná neprázdna množina.

Nech pre funkciu $G \in [A \to CGROUPS]$ funkčná hodnota G(a) prvku $a \in A$ je označená ako $G(a) = \Gamma_a = (G_a, \oplus_a)$.

Potom množina $F_G = \{f \mid f \in [A \to \bigcup_{a \in A} G_a] \wedge \forall a \in A: f(a) \in G_a\}$ spolu s binárnou operáciou $\oplus^{F_G}$ na F_G takou, že $\forall f, f' \in F_G : f \oplus^{F_G} f' = f''$, kde $\forall a \in A: f''(a) = f(a) \oplus_a f'(a)$, tvoria komutatívnu grupu.

Ak A je indexovaná množina s k-prvkami, potom tiež množina $\mathbf{F}_G = \{\mathbf{f} \mid f \in F_G\}$ s operáciou $\oplus^{\mathbf{F}_G}$ takou, že $\forall \mathbf{f}, \mathbf{f}' \in F_G : \mathbf{f} \oplus^{\mathbf{F}_G} \mathbf{f}' = \mathbf{f}''$, kde $f'' = f \oplus^{F_G} f'$, tvoria taktiež komutatívnu grupu.

Spolu so skalárnym násobením daným v predchádzajúcej časti tvoria priestory (moduly) nad okruhom celých čísel.

Ak funkcia $G \in [A \to CGROUPS]$ je „konštantná“, t.j. $\forall$ a, a' $\in$ A: $G(a) = G(a') = (G, \oplus)$, potom $\oplus^{F_G}$ je tiež označená ako $\oplus^{[A \to G]}$ a $\oplus^{\mathbf{F}_G}$ je označená ako $\oplus^k$.

Poznámka: Pre k x l rozmerné matice **A**, **A'**, ktorých stĺpce sú tvorené prvkami množiny $\mathbf{F}_G$, operátor $\oplus^{\mathbf{F}_G}$ (resp. $-\oplus^{\mathbf{F}_G}$) označuje operáciu takú, že $\mathbf{A} \oplus^{\mathbf{F}_G} \mathbf{A'} = \mathbf{B}$ (resp. $\mathbf{A} -\oplus^{\mathbf{F}_G} \mathbf{A'} = \mathbf{B}$), kde $\forall i \in \{1, .., l\}$:

$\mathbf{B}_i = \mathbf{A}_i \oplus^{\mathbf{F}_G} \mathbf{A'}_i$ (resp. $\mathbf{B}_i = \mathbf{A}_i -\oplus^{\mathbf{F}_G} \mathbf{A'}_i$) ($\mathbf{A}_i$, $\mathbf{A'}_i$ a $\mathbf{B}_i$ sú i-te stĺpce matíc **A**, **A’** a **B**).

Takto, ($[A \to Z]$, $+^{[A \to Z]}$) (kde + je súčet celých čísel) je komutatívna grupa a spolu so spomenutým skalárnym násobením tvoria priestor (modul) nad okruhom celých čísel.

2.2 Stavová rovnica špecifikovaných automatov

Nech SM = (S, E, δ, s_0) je (deterministický) automat bez výstupnej funkcie, kde

- S je množina stavov;
- E je množina udalostí (vstupná abeceda);
- δ: S x E $\supseteq$ D(δ) $\rightarrow$ S je prechodová funkcia čiastočne definovaná na S x E s hodnotami v S (D(δ) je definičný obor δ);
- q_0 je počiatočný stav automatu;

Ďalej nech pre každé e $\in$ E symbol S(e) označuje množinu stavov v ktorých môže nastať udalosť e, t.j. S(e) = {s | [s, e] $\in$ D(δ)}.

Ak existuje grupa (S', $\oplus$) taká, že S $\subseteq$ S' potom je možné definovať funkciu stavových zmien Δ: S x E $\supseteq$ D(δ) $\rightarrow$ S' tak, že platí:

$$\forall\ [s, e] \in D(\delta): \delta(s, e) = s \oplus \Delta(s, e).$$

Ak navyše Δ je na D(δ) stavovo invariantná, t. j. platí výraz

$\forall$ e $\in$ E $\forall$ a, b $\in$ S(e): Δ(a, e) = Δ(b, e), potom existuje funkcia stavových zmien Δ': E $\rightarrow$ S': $\forall$ [s, e] $\in$ D(δ): Δ(s, e) = Δ'(e)

a môžeme písať:

$$\forall\ [s, e] \in D(\delta): \delta(s,e) = s \oplus \Delta'(e). \qquad (1)$$

Pripomeňme, že ak pre stav $s_0 \in$ S a konečnú postupnosť vstupov *e* = $\{u_i\}_{i=1}^{k}$ existuje postupnosť stavov $\{s_i\}_{i=1}^{k}$ taká, že platí $\forall$ i $\in$ {1 .. k}: $s_i \in \delta$ (s_{i-1}, u_i), potom takúto postupnosť udalostí *e* nazývame realizovateľná z s_0.

Ak teda existuje komutatívna grupa (S', $\oplus$) taká, že S $\subseteq$ S' a platí (1), potom pre každú z $s_0 \in$ S realizovateľnú postupnosť udalostí *e* platí:

$$s_k = s_0 \oplus \sum_{e \in E}{}^{\oplus} \Delta'(e) \bullet_{\oplus} b_e(e) = s_0 \oplus \hat{\Delta}'(b_e) \qquad (2),$$

kde funkcia $\hat{\Delta}'$ je multimnožinovým rozšírením funkcie Δ'. Táto priraďuje multimnožine nad E vyjadrujúcej koľkokrát sa jednotlivé udalosti uskutočnia zmenu stavu, ktorá by bola týmito výskytmi udalostí spôsobená.

Ak E = {e_1, .. e_n} (prvky množiny udalostí indexujeme) potom

$$s_k = s_0 \oplus \left| \Delta'(e_1) \;\; . \;\; . \;\; \Delta'(e_n) \right| \bullet_{\oplus} \begin{vmatrix} b_e(e_1) \\ . \\ . \\ b_e(e_n) \end{vmatrix} = s_0 \oplus \boldsymbol{\Delta}'^{\mathbf{t}} \bullet_{\oplus} \mathbf{b}_{\boldsymbol{e}} = s_k \qquad (3).$$

To znamená, že ak platia uvedené predpoklady, potom pre každé dve z s_0 realizovateľné postupnosti udalostí (*e* a *e'*) s rovnakým počtom výskytu jednotlivých udalostí ($b_e = b_{e'}$) ich realizácia vedie k rovnakému stavu.

Nasledujúci príklad demonštruje, že ak neplatí (1) realizácia takýchto dvoch postupností nemusí viesť k rovnakému stavu.

Príklad 2.2.1:

Nech S = Z, E = {e_1, e_2}, D(Δ) = Z x E a nech $\Delta(s, e_1) = s/2$ a $\Delta(s, e_2) = s - 2$.

Potom pre $s_0 = 4$ realizácia postupností {e_1, e_2} vedie k stavu $s_2 = (4/2) - 2 = 0$ a realizácia postupnosti {e_2, e_1} vedie k stavu $s_2 = (4 - 2)/2 = 1$.

Z rovníc (2) a (3) môžeme formulovať nasledujúce tvrdenia:

Stav s' je dosiahnuteľný zo stavu s vtedy a len vtedy ak existuje zo stavu s realizovateľná postupnosť udalostí *e* pre ktorú multimnožina b_e (vektor $\mathbf{b}_e$) je riešením rovnice $\hat{\Delta}'(x) = \boldsymbol{\Delta}'^{\mathbf{t}} \bullet_{\oplus} \mathbf{x} = s' - s$.

Ak je stav s' dosiahnuteľný zo stavu s potom rovnica

$\hat{\Delta}'(x) = \mathbf{\Delta'^t} \bullet_{\oplus} \mathbf{x} = s' -_{\oplus} s$

má riešenie, t,j, riešenie rovnice (3) je nevyhnutnou podmienkou dosiahnuteľnosti stavu s_k zo stavu s_0.

Príklad 2.2.2:

Nech S = Z, E = {e_1, e_2, e_3}, D(Δ) = Z x E a $\Delta'(e_1) = -2$, $\Delta'(e_2) = 5$, $\Delta'(e_3) = 1$.

Potom napr. pre postupnosti e = {e_1, e_2, e_3, e_2, e_3} a e' = {e_3, e_2, e_1, e_2, e_3} realizovateľné z $s_0 = 0$ ($\mathbf{b}_e = \mathbf{b}_{e'} = [\,1, 2, 2]^T$) ich realizácia vedie k stavu

$$s_5 = 0 + |\, -2 \quad 5 \quad 1 \,| . \begin{vmatrix} 1 \\ 2 \\ 2 \end{vmatrix} = 0 - 2 + 10 + 2 = 10$$

Podmienkou pre dosiahnutie stavu s’ = 10 zo stavu s = 0 je realizácia postupnosti udalostí e pre ktorú vektor $\mathbf{b}_e$ je riešením rovnice

$|\, -2 \quad 5 \quad 1 \,| . \begin{vmatrix} x_1 \\ x_2 \\ x_3 \end{vmatrix} = 10 - 0$, t.j. postupnosti, pre ktorú

$b_e(e_3) = 2 \,.\, b_e(e_1) - 5 \,.\, b_e(e_2) + 10$.

Špecifikáciou stavovej množiny automatu (doplniteľnosť na grupu) je možné získať funkciu stavových zmien. Ak je táto navyše na definičnom obore prechodovej funkcie stavovo invariantná, potom je možné písať rovnice (2) a (3), ktoré pre automat s uvedenými vlastnosťami poskytuje významný aparát pre analýzu správania sa (otázky dosiahnuteľnosti atď.).

V nasledujúcej časti je uvedená formálna definícia Petriho sietí a jej vzťah k automatom použijúc poznatky z predchádzajúcej časti. Ukážeme, že Petriho sieť je špeciálnym prípadom distribuovaného automatu s funkciou stavových zmien invariantnou na definičnom obore prechodovej funkcie na stave (na stave a funkcii stavových zmien je závislý definičný obor prechodovej funkcie), pričom rovnica (3) poskytuje možnosť použitia metód lineárnej algebry pri analýze Petriho siete.

2. 3 Petriho sieť - formálna definícia

Petriho siete sú rozšíreným formalizmom najmä pre ich schopnosť prehľadne graficky vyjadriť štruktúru systému a z dôvodu poskytnutia silného algebraického aparátu v podobe rovnice (4), pričom ponúkajú mnoho ďalších výhod - možnosť modelovať distribuované systémy s paralelným asynchrónnym správaním atď.

Nasledujúce dve definície formálne definujúce Petriho sieť sú ekvivalentné.

V zmysle (Reisig, 1985) a (Murata, 1989) môže byť Petriho sieť definovaná vo forme bipartitného grafu s dvoma typmi vrcholov:

Definícia 2.3.1: *Petriho sieť* je pätica PN = (P, T, F, w, m_0), kde:

- P = $\{p_1, .. , p_m\}$ je konečná neprázdna *množina pozícií*;
- T = $\{t_1, .. , t_n\}$ je konečná neprázdna *množina prechodov*;
- $P \cap T = \emptyset$, $P \cup T \neq \emptyset$ ($\emptyset$ je prázdna množina);
- $F \subseteq (P \times T) \cup (T \times P)$ je *množina hrán* (*relácia toku*);
- w: $F \to (Z^+ \cup \{0\})$ je *váhová funkcia*;
- m_0: $P \to (Z^+ \cup \{0\})$ je *počiatočné značkovanie*.

Pri grafickom znázornení pozície sú reprezentované krúžkami, prechody sú reprezentované obdĺžnikmi, hrany orientovanými oblúkmi nad ktorými sú umiestnené funkčné hodnoty váhovej funkcie. Ak nad hranou h $\in$ F nie je nadpísaná hodnota funkcie w(h), potom sa implicitne predpokladá, že w(h) = 1.

Triviálnou zmenou tejto definície v zmysle maticovej formy podľa Jensena (1986) získame definíciu:

Definícia 2.3.2: *Petriho sieť* je pätica PN = (P, T, *I*, *O*, m_0), kde:

- P = $\{p_1, .. , p_m\}$ je konečná neprázdna *množina pozícií*;
- T = $\{t_1, .. , t_n\}$ je konečná neprázdna *množina prechodov*;
- $P \cap T = \emptyset$, $P \cup T \neq \emptyset$;
- *I*: P x T $\rightarrow$ $(Z^+ \cup \{0\})$ je *vstupná funkcia* (**I** môže byť chápaná ako celočíselná matica rozmeru m x n);
- *O*: T x P $\rightarrow$ $(Z^+ \cup \{0\})$ je *výstupná funkcia* (**O** môže byť chápaná ako celočíselná matica rozmeru n x m);
- m_0: P $\rightarrow$ $(Z^+ \cup \{0\})$ je *počiatočné značkovanie*.

Predpokladá sa ďalej, že žiadna pozícia ani prechod nie sú izolované, t.j.:

$\forall$ p $\in$ P: $\exists$ t $\in$ T: *I*(p, t) $\neq 0 \vee$ *O*(t, p) $\neq 0$

$\forall$ t $\in$ T: $\exists$ p $\in$ P: *I*(p, t) $\neq 0 \vee$ *O*(t, p) $\neq 0$

Často je pre PN definovaná navyše K-funkcia *K*: P $\rightarrow$ $(Z^+ \cup \{\infty\})$ určujúca *kapacitu* pozícií. Ak K-funkcia nie je definovaná, potom predpokladáme nekonečnú kapacitu pre všetky pozície.

Vzťah medzi váhovou funkciou *w* z definície 2.3.1 a funkciami *I* a *O* z definície 2.3.2 je daný výrazom

$\forall$ p $\in$ P $\forall$ t $\in$ T: *O*(t, p) = *w*(t, p) ak [t, p]$\in$ F, inak *O*(t, p) = 0 a *I*(p, t) = *w*(p, t) ak [p, t] $\in$ F, inak *I*(p, t) = 0.

Funkcia *C*: P x T $\rightarrow$ Z daná výrazom $\forall$ [p, t] $\in$ P x T: *C*(p, t) = *O*(t, p) - *I*(p, t) (celočíselná m x n rozmerná matica **C** daná rozdielom $\mathbf{C} = \mathbf{O}^t - \mathbf{I}$, kde $\mathbf{O}^t$ je matica **O** transponovaná je *nazývaná incidenčnou funkciou* (*maticou*) PN.

Zdrojovou (vstupnou) pozíciou prechodu t $\in$ T je nazývaná každá pozícia p $\in$ P, pre ktorú [p, t] $\in$ F. Podobne p $\in$ P je nazývaná *cieľovou* (výstupnou) pozíciou prechodu t $\in$ T ak [t, p] $\in$ F.

Petriho sieť je nazývaná *rýdza* ak neobsahuje žiadnu dvojicu p, t pre ktorú p je zdrojovou i cieľovou pozíciou prechodu t, t.j. ak platí $\forall$ p $\in$ P $\forall$ t $\in$ T: (p, t) $\in$ F $\Rightarrow$ (t, p) $\notin$ F (v zmysle definície 2.3.1) alebo I(p, t) $\neq 0 \Rightarrow O$(t, p) = 0 (v zmysle definície 2.3.2).

Je zrejmé, že pre rýdzu sieť incidenčná matica **C** jednoznačne určuje matice **O** a **I** (funkcie *O* a *I*) z definície 2.3.2 a reláciu toku F a váhovú funkciu *w* z definície 2.3.1.

Stav Petriho siete je vyjadrený *značkovaním*. Je to funkcia *m*: P $\rightarrow Z^+ \cup \{0\}$, ktorá pre každú pozíciu určuje počet značiek (ak je definovaná funkcia kapacity *K*, potom pre značkovanie musí navyše platiť $\forall$ p $\in$ P: $m(p) \leq K(p)$) (je zrejmé, že *m* je multimnožina nad P v zmysle definície z časti 2.1). Pri grafickom znázornení je stav vyjadrený príslušným počtom značiek *m*(p) v krúžku reprezentujúcom pozíciu p $\in$ P.

V ďalšom texte bude značkovanie *m* stotožnené s vektorom jeho funkčných hodnôt $\mathbf{m} = |\ m(p_1) \ .\ .\ m(p_m)\ |^t$.

Zmena stavu PN nastáva uvádzaním prechodov do činnosti (spúšťaním, otváraním prechodov). Prechod t $\in$ T je *spustiteľný* (*otvoriteľný, realizovateľný*) zo značkovania *m* ak platí:

$\forall$ p $\in$ P: $m(p) \geq I(p, t) \wedge K(p) \geq m(p) + O(t, p)$,

t. j. ak pre ktorúkoľvek zdrojovú pozíciu daného prechodu počet značiek v pozícii nie je menší ako váha príslušnej hrany a počet značiek počas spustenia prechodu nepresahuje povolenú kapacitu (pri tejto definícii predpokladáme ľubovoľné poradie spotrebovania a produkovania značiek funkciami *I* a *O* počas spustenia prechodu). *Spustenie otvoriteľného prechodu* t $\in$ T *zo značkovania* m spôsobí zmenu značkovania na m', pre ktoré platí: $\forall$ p $\in$ P: $m'(p) = m(p) + O(p, t) - I(p,t)$.

Príklad 2.3.1:

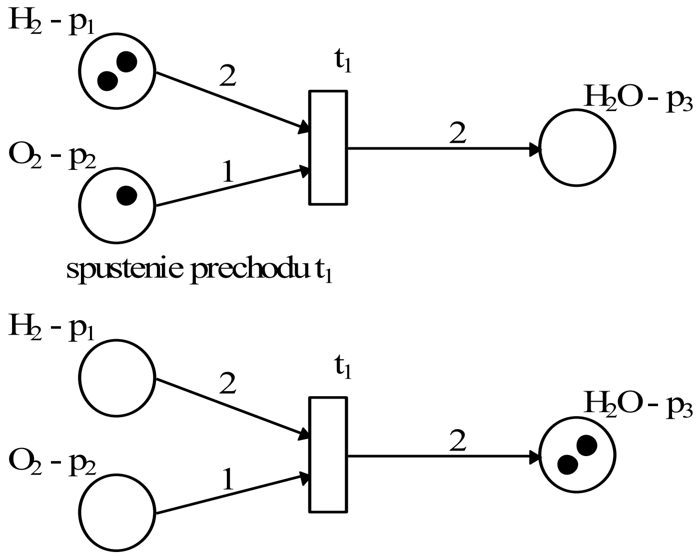
Príklad tohto prechodového pravidla je znázornený na nasledujúcom obrázku použijúc známu chemickú reakciu $2H_2 + O_2 \rightarrow 2H_2O$, prepísanú do PN s tromi pozíciami a jedným prechodom:

$$P = \{p_1, p_2, p_3\}, \qquad T = \{t_1\}, \qquad I = \begin{vmatrix} 2 \\ 1 \\ 0 \end{vmatrix}, \qquad O = |\ 0\ \ 0\ \ 2\ |,$$

inicializačné značkovanie $\mathbf{m}_0 = |\ 2\ 1\ 0\ |^t$,

značkovanie po aktivácii prechodu t_1 je $\mathbf{m}_1 = |\ 0\ 0\ 2\ |^t$.

Značkovanie m_0 znázorňuje stav, keď dve molekuly H_2 a jedna molekula O_2 sú v pozíciách p_1 a p_2, po spustení prechodu t_1 vzniknú dve molekuly H_2O v pozícii p_3 - PN má značkovanie m_1, z ktorého už prechod t_1 nie je otvoriteľný.



Obrázok 2.3.1. Grafická reprezentácia PN z príkladu 2.3.1

Realizácia spustiteľného prechodu $t_i \in T$ zo značkovania *m* vedie teda k značkovaniu *m'* pričom **m'** = **m** + **C** . **Y**, kde **Y** = $|y_1 .. y_n|^t$, $y_i = 1$ a $y_j = 0$ pre všetky $j \in \{1, .., n\}$ rôzne od i.

Podobne ako pre automaty, prázdnu postupnosť prechodov budeme nazývať realizovateľnou z ľubovoľného značkovania. Konečnú neprázdnu postupnosť prechodov $t = \{u_i\}_{i=1}^{k}$, kde $\forall$ i $\in\{1 .. k\}$: $u_i \in T$, budeme nazývať *realizovateľnou zo značkovania* m_a ($a \in Z^+ \cup \{0\}$) ak platí, že $\forall$ i $\in\{1 .. k\}$: prechod u_i je spustiteľný zo značkovania m_{a+i-1}, kde $\forall p \in P$: $m_{a+i}(p) = m_{a+i-1}(p) + O(p, u_i) - I(p, u_i)$.

Spustenie realizovateľnej postupnosti prechodov *t* zo značkovania *m* vedie k značkovaniu *m'*, pričom

$$\mathbf{m}' = \mathbf{m} + \mathbf{C} \,.\, \mathbf{b}_t, \tag{4}$$

kde b_t je multimnožina nad T určená postupnosťou t. Funkciu (multimnožinu nad T) $Y = b_t$ a vektor jej funkčných hodnôt $\mathbf{Y} = \mathbf{b}_t$ nazývame potom *realizovateľné zo značkovania m*.

Ak pre multimnožinu Y nad T platí $\mathbf{I} . \mathbf{Y} \leq \mathbf{m}$, (kde $\leq$ predstavuje porovnanie po zložkách, t. j. pre dva celočíselné m-rozmerné vektory $\mathbf{X}$ a $\mathbf{X}'$ platí: $(\mathbf{X} \leq \mathbf{X}' \Leftrightarrow \forall\ i \in \{1 .. m\}: x_i \leq x_i'))$, potom multimnožinu Y a vektor $\mathbf{Y}$ nazývame *priamo realizovateľné* z *m*. Priama realizovateľnosť vyjadruje skutočnosť, že akákoľvek postupnosť prechodov t, pre ktorú $\mathbf{b}_t = \mathbf{Y}$, je realizovateľná zo značkovania m.

Značkovanie *m'* je *dosiahnuteľné z m* vtedy a len vtedy ak existuje vektor $\mathbf{Y}$ realizovateľný z m taký, že platí $\mathbf{C} .\mathbf{Y} = \mathbf{m}' - \mathbf{m}$.

Ak je značkovanie *m' dosiahnuteľné z m*, potom rovnica $\mathbf{C} .\mathbf{Y} = \mathbf{m}' - \mathbf{m}$ má riešenie, t,j, riešenie rovnice (4) je nevyhnutnou podmienkou dosiahnuteľnosti značkovania *m'* zo značkovania *m*.

Množina všetkých značkovaní dosiahnuteľných z inicializačného značkovania je označená ako *M*.

V súlade s predchádzajúcou časťou kapitoly 2 je možné Petriho sieť chápať ako automat, ktorého stavová množina je daná rovnosťou S = *M*, pričom $M \subseteq [P \rightarrow Z]$ a $([P \rightarrow Z], +^{[P \rightarrow Z]})$ (kde $+^{[P \rightarrow Z]}$ je operácia sčítania funkcií z $[P \rightarrow Z]$) je komutatívna grupa,

množina vstupných symbolov (udalostí) je daná rovnosťou E = T,

prechodová funkcia δ: $M \times T \supseteq D(\delta) \rightarrow M$ a jej definičný obor a sú dané nasledovne:

$\forall\ [m, t] \in M \times T$: $([m, t] \in D(\delta) \Leftrightarrow \forall\ p \in P$: $m(p) \geq I(p, t)) \wedge K(p) \geq m(p) + O(p,t)$ (t.j. t je spustiteľný v m);

$\forall$ [m, t] $\in$ D(δ): $\delta(m, t) = m + \Delta(m, t) = m + \Delta'(t)$, kde funkcia stavových zmien Δ': T $\rightarrow$ [P $\rightarrow$ Z] je taká, že $\forall$ t $\in$ T $\forall$ p $\in$ P: $\Delta'(t)(p) = O(t,p) - I(p,t) = C(p,t)$.

Multimnožinové rozšírenie funkcie stavových zmien $\hat{\Delta}'$: $T_{MS} \rightarrow$ [P $\rightarrow$ Z] potom priradí ľubovoľnej multimnožine $b \in T_{MS}$ vyjadrujúcej koľkokrát sa jednotlivé prechody otvoria zmenu značkovania Δm_b: P $\rightarrow$ Z, ktorá by týmito otvoreniami bola spôsobená, t.j. $\forall$ $b \in T_{MS}$: $\hat{\Delta}'(b) = \Delta m_b$, kde $\mathbf{\Delta m}_b = \mathbf{C} \, . \, \mathbf{b}$ (pričom $\mathbf{\Delta m}_b$ je vektor funkčných hodnôt funkcie Δm_b).

Ak teda stotožníme značkovanie m s vektorom jeho funkčných hodnôt $\mathbf{m}$, potom rovnica (3) je ekvivalentná s rovnicou (4) a $\mathbf{\Delta^t}$ s maticou $\mathbf{C}$.

Manipulácia s maticovými rovnicami vychádzajúcimi z incidenčnej matice $\mathbf{C}$ vedie k vytvoreniu účinných algebraických metód štrukturálnej analýzy Petriho sietí (napr. P a T invarianty).

Definícia 2.3.3: Multimnožina X nad P (vektor $\mathbf{X} = | \, x(p_1) \, . \, . \, x(p_m) \, |^t$) sa nazýva P-*invariantom* Petriho siete ak platí: $\mathbf{C}^t \, . \, \mathbf{X} = \mathbf{0}$ ($\mathbf{0}$ je nulový vektor rozmeru n).

Definícia 2.3.4: Multimnožina Y nad T (vektor $\mathbf{Y} = | \, y(t_1) \, . \, . \, y(t_n) \, |^t$) sa nazýva T-*invariantom* Petriho siete ak platí: $\mathbf{C} \, . \, \mathbf{Y} = \mathbf{0}$ ($\mathbf{0}$ je nulový vektor rozmeru m).

V ďalšom texte sa pod pojmom *support* k-rozmerného ($k \in Z^+$) vektora $\mathbf{V}$ rozumie množina SUPP = $\{i \mid i \in \{1, . . , k\} \wedge v_i \neq 0\}$.

Význam P a T-invariantov Petriho sietí je zrejmý z nasledujúcich viet (podľa Lautenbacha (1986)).

Veta 2.3.1: Ak **X** je P-invariant, potom pre všetky $m, m' \in M$ také, že m' je dosiahnuteľné z m, platí: $\mathbf{X}^t \cdot \mathbf{m}' = \mathbf{X}^t \cdot \mathbf{m}$

Dôsledok: Ak vektor $\mathbf{X} \in \{0, 1\}^m$ je P-invariant, potom platí:

$\forall\ m \in M$: $\sum_{i \in SUPP_X} m(p_i) = \sum_{i \in SUPP_X} m_0(p_i)$, t. j. support takého invariantu určuje množinu pozícií, pre ktoré je v každom dosiahnuteľnom stave súčet značiek týchto pozícií konštantný.

Veta 2.3.2: Nech $m \neq m_0$ je značkovanie dosiahnuteľné z m_0. Ak platí:

$\mathbf{X}^t \cdot \mathbf{m} = \mathbf{X}^t \cdot \mathbf{m}_0$,

potom vektor **X** je P-invariant.

Dôsledok: Nech $m \in M$. Ak existuje taký P-invariant **X**, že platí:

$\mathbf{X}^t \cdot \mathbf{m} \neq \mathbf{X}^t \cdot \mathbf{m}_0$,

potom značkovanie m nie je dosiahnuteľné zo značkovania m_0.

Nutnou podmienkou na to, aby m bolo značkovaním dosiahnuteľným z m_0, je teda podmienka, aby pre všetky P-invarianty **X** danej Petriho siete platila rovnosť $\mathbf{X}^t \cdot \mathbf{m} = \mathbf{X}^t \cdot \mathbf{m}_0$.

Veta 2.3.3: Ak **Y** je T-invariant, potom existuje značkovanie $m_{\mathbf{Y}}$ také, že sa opakuje, ak **Y** je spúšťací vektor.

T-invariant **Y** realizovateľný zo značkovania *m* určuje po koľkých aktiváciach jednotlivých prechodov ($y(t_i)$ určuje koľkokrát je potrebné spustiť i-ty prechod) sa PN zo stavu *m* opäť dostane do stavu *m*.

Manipuláciami s maticovými rovnicami vychádzajúcimi z incidenčnej matice zo zahraničnej literatúry sa zaoberajú mnohé pramene (napr. Memmi a Roucariol, 1980; Martinez a Silva, 1982; Lautenbach, 1986; Kruckeberg a Jaxy, 1986; Silva, 1987; Silva a Colom, 1989), z domácich autorov potom napr. (Juhás, 1993; Juhás a Kocian 1994, 1995b,c).

Existujú rôzne modifikácie Petriho sietí. Významnú skupinu tvoria logické siete a fuzzy siete (Looney, 1988), kde, zjednodušene, značky a hodnoty funkcií *O* a *I* nadobúdajú hodnoty z množiny $\{0, 1\}$, respektíve z intervalu $\langle 0, 1\rangle$ a aritmetické operátory sú nahradené logickými. Tieto siete sa využívajú napr. pri reprezentácii bázy znalostí (Chen a kol., 1990; Čapkovič, 1995a) v súvislosti s riadením DEDS modelovaných obyčajnými PN (Čapkovič, 1995b). V oblasti riadenia DEDS sú PN využité taktiež v prácach (Willson a Krogh, 1990; Ushio, 1990; Hrúz, 1994).

Dôležitým a využívaným rozšírením PN sú tzv. farebné Petriho siete (Jensen, 1986), ktoré sú využité v súvislosti s modelovaním pružných výrobných systémov napr. v článkoch (Frankovič, 1995; Labátová a Frankovič, 1995). Farebným Petriho sieťam je venovaná ďalšia časť kapitoly 2.

2.4 Farebné Petriho siete

Farebné Petriho siete (Jensen, 1986) predstavujú rozšírenie PN, keď značky môžu byť rôzneho druhu (farby). Ich vznik bol priamo inšpirovaný tzv. predikátovo/prechodovými sieťami (PrT sieťami) (Reisig, 1985). Farebná Petriho sieť môže byť podobne ako Petriho sieť definovaná vo forme grafu (ktorý je veľmi blízky PrT sieti) a v maticovej forme.

Definícia 2.4.1 (grafová forma): *Farebná Petriho sieť* je skupina ôsmich prvkov CPN = (P, T, F, V, Φ, w, φ, m_0), kde:

- P = $\{p_1, .. , p_m\}$ je konečná neprázdna *množina* pozícií;
- T = $\{t_1, .. , t_n\}$ je konečná neprázdna množina prechodov;
- $P \cap T = \emptyset$, $P \cup T \neq \emptyset$;
- $F \subseteq (P \times T) \cup (T \times P)$ je *množina hrán* (*relácia toku*);
- V = $\{v_1: D_1, .., v_k: D_k\}$ je konečná neprázdna množina premenných, pričom v_i označuje i-tu premennú a D_i označuje obor hodnôt (typ) premennej v_i pre $i \in \{1 .. k\}$.
- Φ je funkcia definovaná na P, ktorá priradí každej pozícii ľubovoľnú neprázdnu množinu nazvanú *množina farieb značiek*;
- w je funkcia definovaná na F priraďujúca každej hrane $f \in F$ výraz $w(f)$ obsahujúci množinu premenných $V_f = \{ v_{a_1} : D_{a_1}, .., v_{a_r} : D_{a_r} \} \subseteq V$ (kde $\{a_1, .., a_r\}$ je vybraná postupnosť indexov z množiny {1, .., k}, t.j. $\{a_1, .., a_r\} \subseteq \{1, .., k\}$), pričom výraz $w(f\,)$ definuje funkciu z $[D_{a_1} \times .. \times D_{a_r} \rightarrow \Phi(p)_{MS}]$, kde p je zdrojová alebo cieľová pozícia hrany f ($\exists\, t \in T$: $f = [p, t] \vee f = [t, p]$).
- φ je funkcia definovaná na T priraďujúca každému prechodu $t \in T$ predikát nazvaný *guard*, ktorý môže obsahovať iba premenné

vstupných alebo výstupných hrán prechodu t, t.j. premenné množiny $V_t = \bigcup_{f \in F_t} V_f$, kde $F_t = \{f \mid f \in F \wedge \exists\, p \in P: f = [p, t] \vee f = [t, p]\}$.

- m_0 je *počiatočné značkovanie*, t.j. funkcia definovaná na P priraďujúca každej pozícii $p \in P$ multimnožinu z $\Phi(p)_{MS}$, teda $\forall\, p \in P: m_0(p) \in \Phi(p)_{MS}$.

Poznámka: Definíciu grafovej formy je možné doplniť (formálne spresniť) pridaním funkcie *L* definovanej na F priraďujúcej každej hrane $f \in F$ ($f = [p, t] \vee [t, p]$) jazyk $L(f) = (L_f, M_f, \sigma_f)$, kde:

- množina výrazov L_f jazyka $L(f)$ je formálnym jazykom nad nejakou abecedou obsahujúcou premenné z V_f, ktorého slová obsahujú tieto premenné,
- množina významov je tvorená funkciami z $[D_{a_1} \times .. \times D_{a_r} \to \Phi(p)_{MS}]$,
- relácia σ_f je sémantikou jazyka $L(f)$ (z dôvodu jednoznačnosti nech σ_f je funkcia z $[L_f \to M_f]$).

Potom *w* je funkcia definovaná na F priraďujúca každej hrane $f \in F$ výraz $w(f) \in L_f$, pričom výraz $w(f)$ definuje funkciu $w(f)_\sigma$ takú, že $[w(f), w(f)_\sigma] \in \sigma_f$ (t.j. funkcia $w(f)_\sigma$ je významom výrazu $w(f)$).

V súlade s bežným používaním jazyka budeme v nasledujúcom texte pri použití pojmu výraz $w(f)$ rozumieť väčšinou jeho význam, t.j. funkciu $w(f)_\sigma$. Napr. pre nejaké $g' \in D_{a_1} \times .. \times D_{a_r}$ budeme pod pojmom výsledná hodnota výrazu $w(f)$ po dosadení g' za premenné V_f rozumieť výslednú hodnotu významu $w(f)$, t.j. hodnotu funkcie $w(f)_\sigma(g')$, ktorou je multimnožina určujúca pre jednotlivé farby príslušnej pozície počet značiek.

Definícia 2.4.2 (maticová forma): *Farebná Petriho sieť* je skupina šiestich prvkov $CPN = (P, T, \Phi, I, O, m_0)$, kde:

- $P = \{p_1, .. , p_m\}$ je konečná neprázdna *množina pozícií*;

- $T = \{t_1, .., t_n\}$ je konečná neprázdna *množina prechodov*;
- $P \cap T = \emptyset$, $P \cup T \neq \emptyset$;
- Φ je funkcia definovaná na $P \cup T$, ktorá priradí každej pozícii ľubovoľnú neprázdnu množinu nazvanú *množina farieb značiek* a každému prechodu ľubovoľnú neprázdnu množinu nazvanú *množina farieb otvorení*;
- *I* je *vstupná funkcia* definovaná na P x T taká, že $\forall$ [p, t] $\in$ P x T: $I(p, t) \in [\Phi(t)_{MS} \rightarrow \Phi(p)_{MS}]_L$ (**I** môže byť chápaná ako matica rozmeru m x n, ktorej prvkami sú príslušné funkcie);
- *O* je *výstupná funkcia* definovaná na T x P taká, že $\forall$ [t, p] $\in$ T x P: $O(t, p) \in [\Phi(t)_{MS} \rightarrow \Phi(p)_{MS}]_L$ (**O** môže byť chápaná ako matica rozmeru n x m, ktorej prvkami sú príslušné funkcie);
- m_0 je *počiatočné značkovanie*, t.j. funkcia definovaná na P taká, že $\forall$ $p \in P$: $m_0(p) \in \Phi(p)_{MS}$.

Rovnako ako pre štandartné Petriho siete sa predpokladá, že žiadna pozícia ani prechod nie sú izolované.

Nech symboly *i* a *o* označujú funkcie definované na P x T respektíve T x P, pre ktoré platí:

$\forall$ [p, t] $\in$ P x T: $i(p, t) \in [\Phi(t) \rightarrow \Phi(p)_{MS}] \wedge i(p, t) = b \Leftrightarrow I(p, t) = \hat{b}$ respektíve

$\forall$ [t, p] $\in$ T x P: $o(t, p) \in [\Phi(t) \rightarrow \Phi(p)_{MS}] \wedge o(t, p) = b \Leftrightarrow O(t, p) = \hat{b}$, t.j. funkcie, pre ktoré multimnožinové rozšírenia ich funkčných hodnôt sú funkčnými hodnotami vstupnej a výstupnej funkcie.

Spôsob prepisu CPN v grafovej forme na CPN v maticovej forme je možné popísať nasledovne:

- P, T a m_0 sú totožné;

- Zúženie Φ na P v maticovej forme je dané funkciou Φ v grafovej forme;
- Nech pre každé $t \in T$: $V_t = \{ v_{a_1} : D_{a_1}, .., v_{a_s} : D_{a_s} \}$, kde $\{a_1, .., a_s\}$ je vybraná postupnosť indexov z množiny $\{1, .., k\}$.

  Potom pre každé $t \in T$: $\Phi(t) = \{[d_{a_1}, .., d_{a_s}];\ \forall\ i \in \{1, . ., s\}: d_{a_i} \in D_{a_i} \wedge \varphi(t)(d_{a_1}, .., d_{a_s}) = 1$ (TRUE)$\}$;
- Funkcia *I* je definovaná prostredníctvom funkcie *i*, pre ktorú platí:

  $\forall\ [p, t] \in P \times T$:
  - Ak $[p, t] = f \in F$, potom funkcia i: $\Phi(t) \rightarrow \Phi(p)_{MS}$ je definovaná nasledovne: $\forall\ g = (d_{a_1}, .., d_{a_s}) \in \Phi(t)$: $i(p, t)(g) = w(f)_\sigma(g')$, pričom g' je projekcia g na priestor premenných V_f;
  - Ak $[p, t] \notin F$, potom $i(p, t)$ je nulová funkcia;
- Funkcia *O* je definovaná prostredníctvom funkcie *o*, ktorá je definovaná analogicky ako funkcia *i*.

Spôsob prepisu CPN v maticovej forme na CPN v grafovej forme je možné popísať nasledovne:

- P, T a m_0 sú totožné;
- Φ v grafovej forme je dané zúžením Φ na P v maticovej forme;
- Množina hrán F je určená platnosťou nasledujúcich predikátov: $\forall\ [p, t] \in P \times T$: $I(p, t) \neq 0 \Leftrightarrow [p, t] \in F$; $\forall\ [t, p] \in T \times P$: $O(t, p) \neq 0 \Leftrightarrow [t, p] \in F$;
- $V = \{x: \bigcup_{t \in T} \Phi(t)\}$;
- Funkcia *w* priraďujúca hranám výrazy je daná nasledovne: $\forall\ [p, t] \in F$: $w(p, t) = i(p, t)(x)$; $\forall\ [t, p] \in F$: $w(t, p) = o(t, p)(x)$;
- Funkcia φ priraďujúca prechodom predikáty je určená nasledovne: $\forall\ t \in T$: $\varphi(t) = (x \in \Phi(t))$.

Je zrejmé, že zobrazenie *GM* priraďujúce grafu CPN maticovú formu definované uvedenou transformáciou je surjektívne, ale nie je injektívne, a naopak, zobrazenie *MG* definované transformáciou maticovej formy CPN na graf je injektívne ale nie je surjektívne (dôkaz pozri v (Jensen, 1986)).

Grafová forma získaná uvedenou transformáciou maticovej formy je nazývaná tiež normovaná grafická forma CPN.

Stav CPN je vyjadrený *značkovaním*, t.j. funkciou rovnakého typu ako inicializačné značkovanie, ktorá priraďuje každej pozícii počet značiek jednotlivých farieb.

Zmena stavu je spôsobená otváraním prechodov pre dané farby, pričom prechod t $\in$ T je *otvoriteľný zo značkovania m pre* danú *farbu* x $\in$ Φ(t) vtedy a len vtedy ak $\forall$ p $\in$ P: i(p, t)(x) $\leq$ m(p), kde $\leq$ predstavuje porovnanie po zložkách, t. j. pre dve multimnožiny b a b' nad ľubovoľnou množinou A platí: ($b \leq b' \Leftrightarrow \forall$ a $\in$ A: b(a) $\leq$ b'(a)). Ak prechod t $\in$ T je otvoriteľný zo značkovania m pre farbu x $\in$ Φ(t), potom jeho *otvorenie pre* túto *farbu* odoberie z každej pozície p počet značiek jednotlivých farieb určený funkčnou hodnotou i(p,t)(x) a pridá do každej pozície počet značiek jednotlivých farieb určený funkčnou hodnotou o(t,p)(x), t.j. vedie k novému značkovaniu m' danému pre každú pozíciu p $\in$ P rovnosťou

$$m'(\mathrm{p}) = m(\mathrm{p}) -_{+[\Phi(\mathrm{p})\to Z]} i(\mathrm{p}, \mathrm{t})(\mathrm{x}) +^{[\Phi(\mathrm{p})\to Z]} o(\mathrm{t}, \mathrm{p})(\mathrm{x}).$$

Ak pre funkciu Y definovanú na T takú, že $\forall$ t $\in$ T: Y(t) $\in$ $\Phi(\mathrm{t})_{\mathrm{MS}}$ platí výrok

$$\forall\, \mathrm{p} \in \mathrm{P}: \sum_{\mathrm{t}\in \mathrm{T}}^{+[\Phi(\mathrm{p})\to Z]} I(\mathrm{p}, \mathrm{t})(Y(\mathrm{t})) \leq m(\mathrm{p})$$

potom funkciu Y nazývame *priamo realizovateľnou zo značkovania m*. *Uskutočnenie* priamo realizovateľnej funkcie Y zo značkovania m vedie k značkovaniu m', pričom platí $\forall$ p $\in$ P: m'(p) =

$$m(\mathrm{p}) -_{+[\Phi(\mathrm{p})\to \mathrm{Z}]} \sum_{\mathrm{t}\in \mathrm{T}}{}^{+[\Phi(\mathrm{p})\to \mathrm{Z}]} I(\mathrm{p},\mathrm{t})(Y(\mathrm{t})) +^{[\Phi(\mathrm{p}) \to \mathrm{Z}]} \sum_{\mathrm{t}\in \mathrm{T}}{}^{+[\Phi(\mathrm{p})\to \mathrm{Z}]} O(\mathrm{p},\mathrm{t})(Y(\mathrm{t})).$$

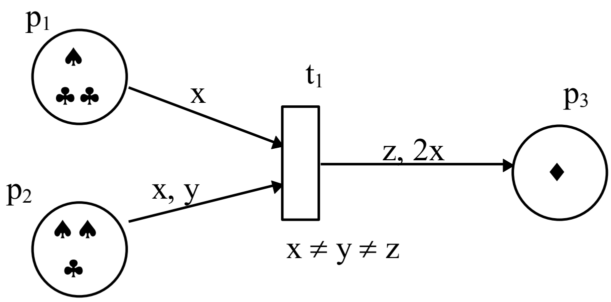
V termínoch grafovej reprezentácie CPN je prechod $t \in T$ otvoriteľný pre tie hodnoty premenných V_t, pre ktoré výrok nazývaný guard je pravdivý, a navyše po dosadení týchto hodnôt do výrazov vstupných hrán výsledné hodnoty týchto výrazov pre jednotlivé farby nie sú väčšie ako počet značiek danej farby v príslušných zdrojových pozíciách prechodu t. To znamená, že otvorenie prechodu $t \in T$ pre skupinu hodnôt premenných vyhovujúcich predchádzajúcim podmienkam spôsobí zmenu značkovania zdrojových pozícií prechodu t zmenšením počtu značiek jednotlivých farieb o výsledné hodnoty získané dosadením do príslušných výrazov vstupných hrán, a zmenu značkovania cieľových pozícií zväčšením počtu značiek jednotlivých farieb o hodnoty získané dosadením do výrazov príslušných výstupných hrán prechodu t.

V nasledujúcich príkladoch budeme predpokladať pri grafovom znázornení CPN pre každú hranu $f = [p, t] \vee f = [t, p]$ výraz $w(f)$ tvorený postupnosťou symbolov premenných z $V_f = \{ v_{a_1} : D_{a_1}, .., v_{a_r} : D_{a_r} \}$ oddelených čiarkami, pričom pred každou premennou v_{a_i} ($i \in \{1, .., r\}$) navyše môže byť reťazec číslic reprezentujúcich celé číslo s_{a_i}. Prázdny reťazec pred premennou v_{a_i} (neprítomnosť reťazca číslic pred premennou v_{a_i}) reprezentuje implicitne číslo $s_{a_i} = 1$. Predpokladáme navyše, že premenné z V_f nadobúdajú hodnoty z $\Phi(p)$. Výraz w(f) potom definuje funkciu $w(f)_\sigma$ danú nasledovne:

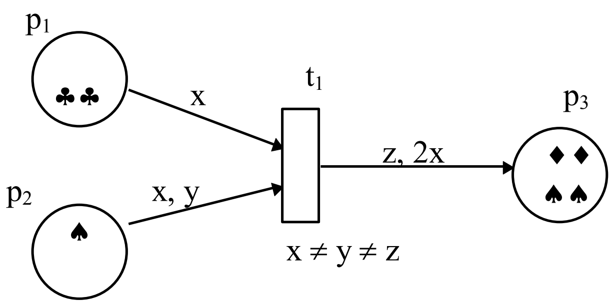
$\forall g' \in D_{a_1} \times .. \times D_{a_r} : w(f)_\sigma(g') = b \in \Phi(p)_{MS}$, pričom

$$\forall c \in \Phi(p): b(c) = \sum_{\substack{i\in\{1,..,r\} \\ \wedge d_{a_i} = c}} s^{a_i}.$$

Ak napr. pre farebnú Petriho sieť na obr. 2.4.1 premenné x, y, z môžu nadobúdať hodnoty z množiny {♠,♣,♦}, potom prechod t_1 nie je otvoriteľný zo značkovania na obr. 2.4.1a napr. pre hodnoty x = ♣, y = ♣, z = ♦ pretože pre ne nie je platný predikát ($x \neq y \neq z$), keďže po dosadení (♣ ≠ ♣ ≠ ♦) = 0 (FALSE); ani pre hodnoty x = ♠, y = ♦, z = ♣, pretože výraz (x, y) nad hranou [p_2, t_1] má po dosadení hodnotu (♠, ♦), ktorá je pre farbu ♦ väčšia ako počet značiek tejto farby v pozícii p_2 (ktorý je 0, keďže v p_2 nie je žiadna značka farby (typu) ♦); avšak je otvoriteľný napr. pre hodnoty x = ♠, y = ♣, z = ♦, pre ktoré jeho otvorenie vedie k značkovaniu na obr. 2.4.1b.



Obrázok 2.4.1a. Ilustrácia jednoduchej farebnej Petriho siete

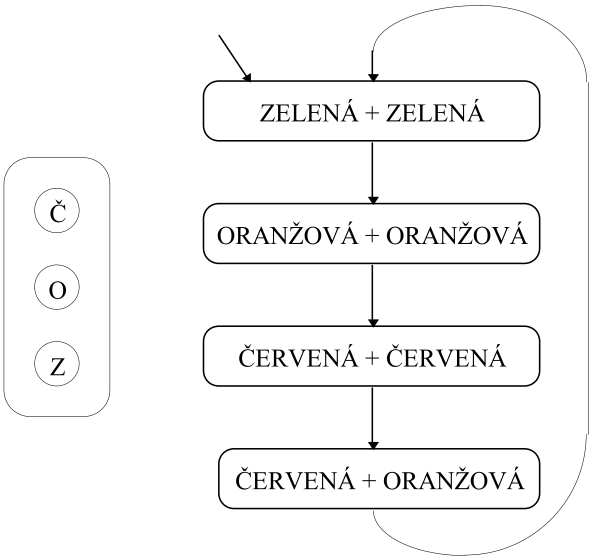


Obrázok 2.4.1b. Farebná Petriho sieť z obr. 2.4.1a po otvorení prechodu pre hodnoty premenných x = ♠, y = ♣, z = ♦.

2.5 Ilustratívny príklad: modelovanie dopravného semaforu Petriho sietou a farebnou Petriho sieťou.

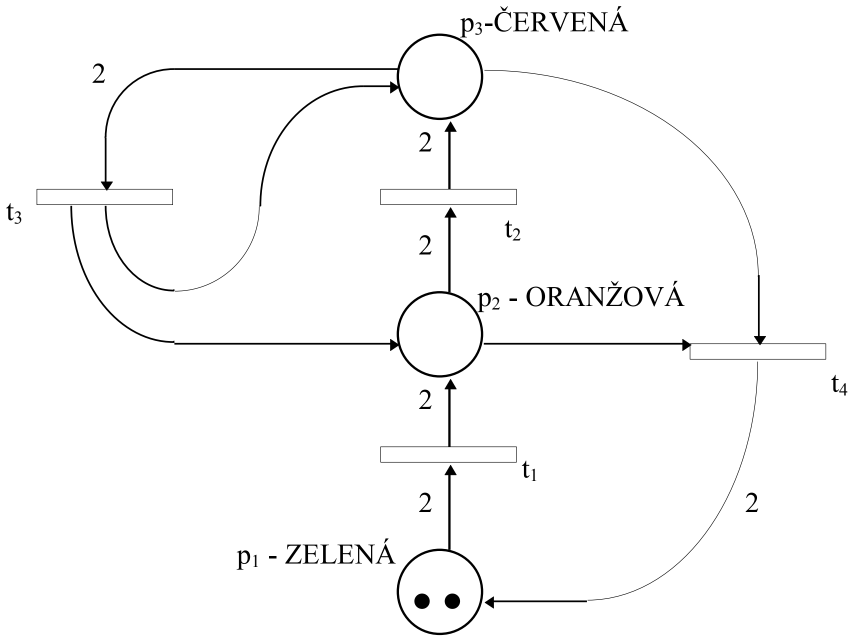
Nasledujúca časť obsahuje jednoduchý príklad (Jensen, 1986) ktorý ilustruje použitie Petriho sietí a farebných Petriho sietí.

Správanie dopravného semaforu s troma farbami svetiel môže byť popísané stavovým diagramom na obrázku 2.4.2.



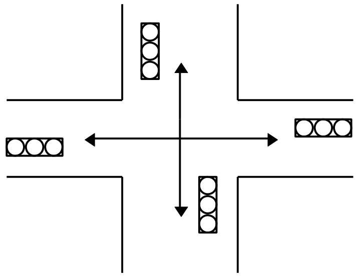
Obrázok 2.4.2. Dopravný semafor a jeho stavový diagram.

Keďže semafor má stav, v ktorom dve svetlá rôznej farby (červená a oranžová) svietia súčasne, stav je reprezentovaný dvojicou svietiacich svetiel. Na obr. 2.4.3 je znázornená Petriho sieť modelujúca semafor na obr. 2.4.2. V sieti sa pohybujú dve značky, ktorých prítomnosť v pozíciách reprezentujúcich farby svetiel znázorňuje fakt, že svetlo danej farby svieti. Ak dve značky sú v jednej pozícii, svieti jedno svetlo. Ak značky sú v rôznych pozíciách svietia dve svetlá súčasne. Inými slovami, svetlo danej farby svieti práve vtedy, ak je v príslušnej pozícii prinajmenšom jedna značka (svetlo nesvieti práve vtedy, ak v príslušnej pozícii nie je žiadna značka). Na začiatku sú obidve značky v pozícii p_1 reprezentujúcej zelenú farbu, t.j. $\mathbf{m}_0 = |2, 0, 0|^t$.



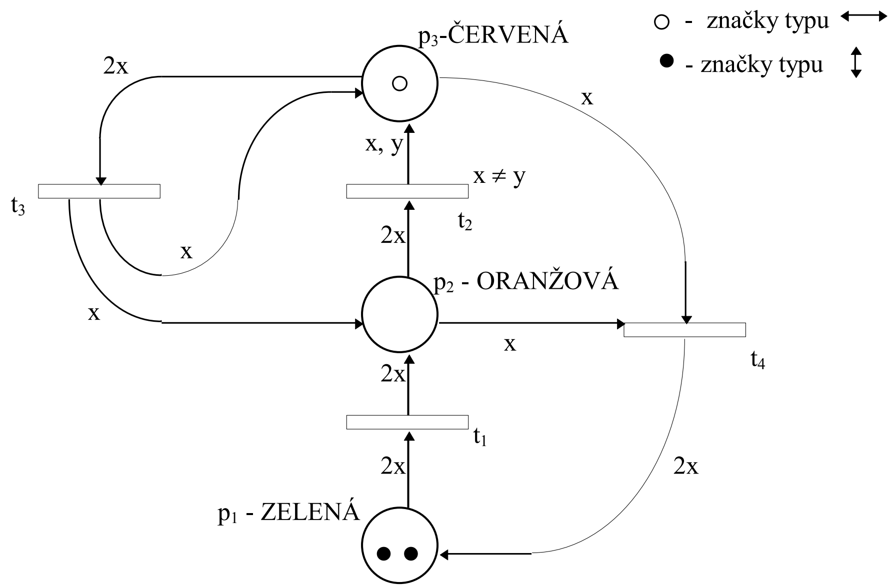
Obrázok 2.4.3. Petriho sieť modelujúca semafor na obr. 2.4.2.

Uvažujme ďalej množinu semaforov synchronizujúcu dopravu na dvoch križujúcich sa cestách znázornených na obr. 2.4.4. Predpokladajme dvojice svetiel umožňujúce dva možné spôsoby (smery) jazdy: spôsob daný šipkou $\longleftrightarrow$ a spôsob daný šipkou $\updownarrow$.



Obrázok 2.4.4. Križovatka s dvoma smermi jazdy

Tento dopravný systém môže byť popísaný farebnou Petriho sieťou na obr. 2.4.5, ktorá má štruktúru rovnakú ako Petriho sieť na obr. 2.4.3.



Obrázok 2.4.5. Farebná Petriho sieť modelujúca svetelnú križovatku na obr. 2.4.4.

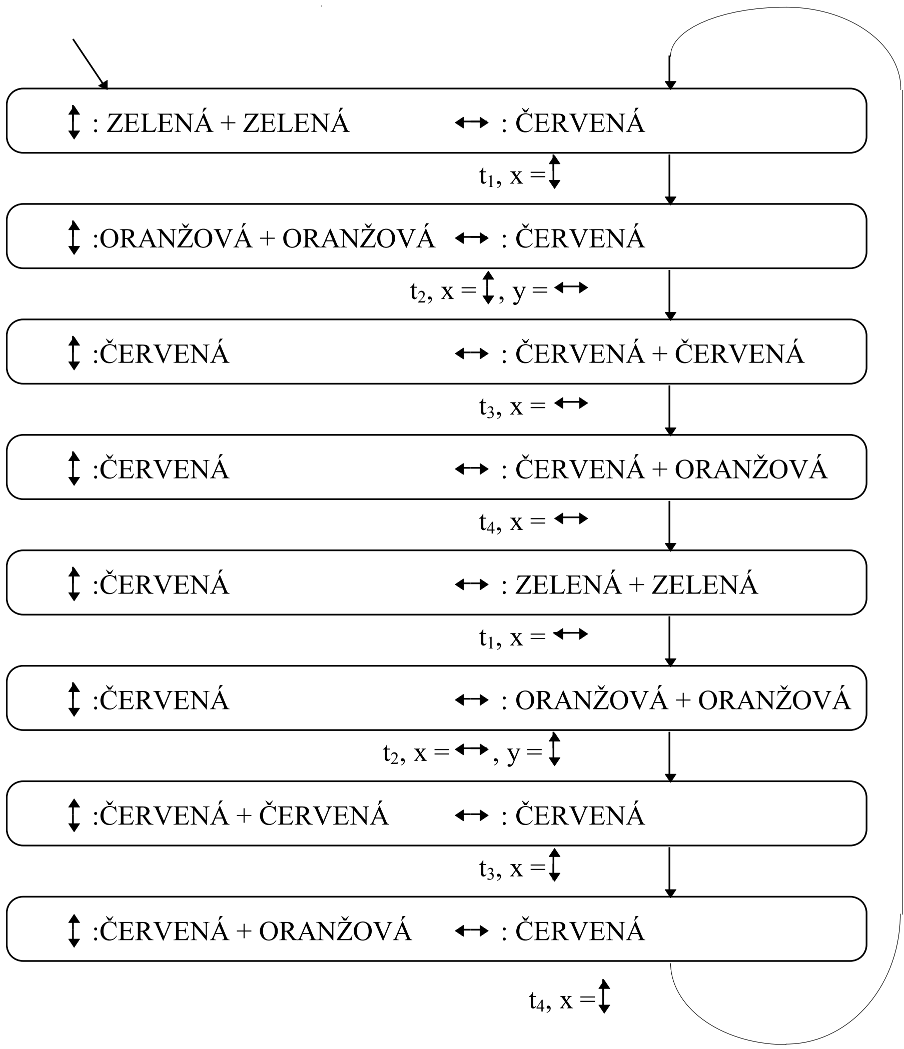
Pre všetky pozície CPN na obr. 2.4.5 množina farieb značiek daná funkciou Φ je zložená zo symbolov $\longleftrightarrow$ a $\updownarrow$ reprezentujúcich dvojice svetiel v príslušnom smere, t.j. $\forall$ p $\in$ P: Φ(p) = { $\longleftrightarrow$, $\updownarrow$ } = SMERY. To znamená, že každá pozícia môže mať dva druhy (farby) značiek: značky typu $\longleftrightarrow$ a značky typu $\updownarrow$. Inými slovami, informácia pripojená k značke reprezentuje v sieti dva rozdielne smery ciest. Presnejšie, svetlo reprezentované pozíciou p svieti v smere $\longleftrightarrow$ respektíve $\updownarrow$ práve vtedy, ak je v pozícii p prinajmenšom jedna značka typu (farby) $\longleftrightarrow$ respektíve $\updownarrow$.

Poznámka: Pripomeňme, že farbou značky sa myslí jedna z hodnôt { $\longleftrightarrow$, $\updownarrow$ }, t.j. smer cesty a nie farba svetla. Tá je reprezentovaná príslušnou pozíciou.

Ku každej hrane je priradený výraz obsahujúci premenné z množiny {x: SMERY, y: SMERY} (t.j. premenné x a y môžu nadobúdať hodnoty z množiny SMERY). Z obrázku je zrejmé, že funkcia *w* priraďuje každej hrane jeden z výrazov „2x“, „x“ a „x, y“. Výraz „2x“ definuje funkciu zobrazujúcu prvok množiny SMERY dosadený za x na multimnožinu priraďujúcu danému prvku číslo 2 a zvyšnému prvku číslo 0. To znamená, že výraz „2x“ reprezentuje zväčšenie respektíve zmenšenie počtu značiek farby (t.j. smeru) dosadenej za x o dve a zmenu počtu značiek zvyšnej farby o žiadnu značku. Podobne výraz „x“ znamená zmenu počtu značiek farby dosadenej za x o jednu, pričom počet značiek zvyšnej farby sa nemení. Výraz „x, y“ znamená zmenu počtu značiek farby dosadenej za x o jednu a zmenu počtu značiek farby dosadenej za y taktiež o jednu.

Predikát $x \neq y$ nazývaný guard prechodu t_2 špecifikuje, že ak sa tento prechod otvorí, potom premenné x a y musia mať rôzne hodnoty, t.j. jedna musí byť $\longleftrightarrow$ a druhá $\updownarrow$. Predikáty ostatných prechodov, t.j. prechodov t_1, t_3 a t_4 sú konštantné predikáty 1 (TRUE).

Počiatočné značkovanie siete je dané dvoma značkami farby $\updownarrow$ v pozícii p_1 a jednou značkou farby $\longleftrightarrow$ v pozícii p_3. Toto značkovanie reprezentuje zelenú pre smer $\updownarrow$ a červenú pre smer $\longleftrightarrow$. Stavový diagram modelovaného dopravného systému je na obr. 2.4.6.



Obrázok 2.4.6. Stavový diagram svetelnej križovatky s dvoma smermi jazdy.

Nech funkcia $id \in [\mathrm{SMERY}_{\mathrm{MS}} \to \mathrm{SMERY}_{\mathrm{MS}}]_{\mathrm{L}}$ je taká, že

$\forall\, b \in \mathrm{SMERY}_{\mathrm{MS}}$: $id(b) = b$,

a funkcia $sum \in [\mathrm{SMERY}_{\mathrm{MS}} \to \mathrm{SMERY}_{\mathrm{MS}}]_{\mathrm{L}}$ je taká, že

$\forall\ b \in \mathrm{SMERY_{MS}}$: $sum(b) = const_k$, kde k = $\sum_{c \in \mathrm{SMERY}} b(c)$.

Potom nasledujúca maticová forma je izomorfná s maticovou formou, ktorá je *GM* obrazom grafu na obr. 2.4.5:

$P = \{p_1, p_2, p_3\}$; $T = \{t_1, t_2, t_3, t_4\}$;

$\forall\ p \in P$: $\Phi(p) = \mathrm{SMERY}$, $\forall\ t \in T$: $\Phi(t) = \mathrm{SMERY}$;

$I(p_1, t_1) = 2id$, $O(t_1, p_2) = 2id$,

$I(p_2, t_2) = 2id$, $O(t_2, p_3) = sum$,

$I(p_3, t_3) = 2id$, $O(t_3, p_2) = id$,

$I(p_2, t_4) = id$, $O(t_3, p_3) = id$,

$I(p_3, t_4) = id$, $O(t_4, p_1) = 2id$;

$m_0(p_1)(\longleftrightarrow) = 0$, $m_0(p_1)(\updownarrow) = 2$,

$m_0(p_2)(\longleftrightarrow) = 0$, $m_0(p_2)(\updownarrow) = 0$,

$m_0(p_3)(\longleftrightarrow) = 1$, $m_0(p_3)(\updownarrow) = 0$.

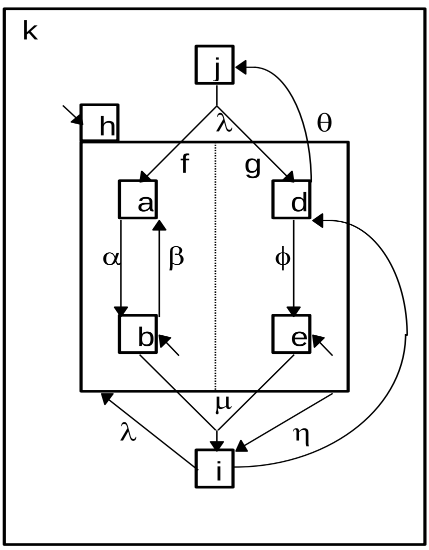
Poznámka: V súlade s podkapitolou 2.1 by bolo presnejšie namiesto 2*id* písať $2 \overset{\bullet}{}_{+[\mathrm{SMERY_{MS}} \to Z]^{[\mathrm{SMERY_{MS}} \to [\mathrm{SMERY} \to Z]]}}\ id$.

3 Hierarchické stavové stroje

Počet stavov v reprezentácii DEDS automatom rastie exponenciálne s počtom paralelných zložiek. Potreba redukcie počtu stavov viedla k definícii formalizmu Statechart (Harel, 1987; Harel a kol., 1987), ktorého zjednodušenou formou je tzv. hierarchický automat (Brave a Heymann, 1993) alebo hierarchický stavový stroj (HSM), ktorý rozširuje automat pridaním hierarchie a ortogonality. Špeciálne:

1) Stavy sú organizované v hierarchii nadstavov a podstavov, čo umožňuje modelovať rôzne úrovne abstrakcie a detailu;
2) Stavy sú komponované ortogonálne, čo umožňuje modelovať stavy systému distribuovane;
3) Prechody sú prípustné na všetkých úrovniach hierarchickej štruktúry, čo umožňuje efektívny popis.

Poznámka: Všetky príklady v nasledovnom texte sa týkajú HSM na obrázku 3.1.



Obrázok 3.1. HSM s alternujúcou štruktúrou.

Graficky sú stavy HSM reprezentované obdĺžnikmi. Hierarchia je reprezentovaná vnútrom obdĺžnikov, t.j. podstav daného stavu je reprezentovaný obdĺžnikom vnútri obdĺžnika znázorňujúceho daný stav. Takto j, h a i sú bezprostredné podstavy stavu k HSM na obrázku 3.1. Stavy f a g sú bezprostredné podstavy stavu h a tiež podstavy stavu k.

Každý stav HSM je buď AND-stavom (čo znamená, že ak HSM je v danom AND-stave, potom je súčasne vo všetkých jeho bezprostredných podstavoch), OR-stavom (ak HSM je v danom OR-stave, potom je práve v jednom z jeho bezprostredných podstavov) alebo základným stavom (ktorý nemá žiadne podstavy). Takto napr. stavy k, f a g sú OR-stavy (ak teda HSM na obr. 3.1 je napr. v stave f, potom je práve v jednom z jeho podstavov a alebo b), stavy a, b, d, e, i, j sú základné stavy a stav h je AND-stavom (ak teda HSM na obr. 3.1 je v stave h, potom je súčasne v stavoch f a g).

Defaultová šípka určuje pre každý OR-stav a bezprostredný defaultový (náhradný) podstav označený ρ(a). Tak napr. na obr. 3.1 určuje defaultová šípka pre stav f defaultový podstav ρ(f)=b.

Konfigurácie HSM sú skupiny (množiny) stavov reprezentujúce ortogonálne stavy, v ktorých HSM môže byť súčasne (napr. ⟨a, g⟩, ⟨b,e⟩ a ⟨i⟩ sú konfigurácie).

Prechodová cesta λ na obr. 3.1 je reprezentovaná trojicou (⟨j⟩, λ, ⟨a, d⟩), kde ⟨j⟩ je zdrojová a ⟨a, d⟩ cieľová konfigurácia. Každá prechodová cesta je združená s práve jedným stavom HSM - t..j. s „najmenším“ OR-stavom obsahujúcim jej zdrojovú a cieľovú konfiguráciu spolu s príslušnou hranou. Takto napr. prechodová cesta označená α patrí stavu f a prechodová cesta označená θ patrí stavu k.

Formálne je teda možné HSM popísať nasledovne (podľa (Brave a Heymann, 1993)).

Definícia 3.1*: Hierarchický automat* je štruktúra HSM = (A, $\vdash$, Σ, T, ρ), kde:

a) A je *množina stavov* taká, že A = $A^+ \cup A^\perp \cup A^{basic}$ a $A^+ \cap A^\perp = A^+ \cap A^{basic} = A^\perp \cap A^{basic} = \emptyset$, kde A^+ je *množina* OR-*stavov*, $A^\perp$ je *množina* AND-*stavov* a A^{basic} je *množina základných stavov* (t.j. množina A je zjednotením troch ľubovoľných disjunktných množín označujúcich AND-stavy, OR-stavy a základné stavy).

b) $\vdash$ je *relácia hierarchie* na množine A ($\vdash \subseteq$ A x A) taká, že:

1. existuje práve jeden stav, nazvaný *koreňový* a označený r = r(HSM), taký, že: $\nexists$ a $\in$ A: a $\vdash$ r ([a, r] $\in \vdash$).
2. $\forall$ (a $\in$ A $\wedge$ a $\neq$ r) $\exists!$ b $\in$ A: b $\vdash$ a. Stav b je nazvaný *bezprostredným nadstavom* stavu a, stav a je *bezprostredným podstavom* stavu b.
3. Stav a $\in$ A nemá žiadny bezprostredný podstav vtedy a len vtedy ak a $\in A^{basic}$.
4. (*Alternujúca štruktúra*) Ak b $\vdash$ a potom buď b $\in A^+ \wedge$ a $\notin A^+$ alebo b $\in A^\perp \wedge$ a $\in A^+$.

Poznámka: Všeobecne HSM nemusí mať alternujúcu štruktúru. Každý HSM možno pomocou jednoduchých pravidiel previesť na HSM s alternujúcou štruktúrou (pozri (Brave Heymann, 93) a obr. 3.2).

Je zrejmé, že pár (A, $\vdash$) definuje *strom hierarchie*. Nech l označuje hĺbku stromu a A_j, $0 \leq j \leq l$ označuje množinu stavov na úrovni j, pričom koreňový stav r má úroveň hĺbky 0. Tranzitívny uzáver relácie $\vdash$ označený $\vdash^+$ určuje pre a $\vdash^+$ b, že b je podstavom (nie nutne bezprostredným) stavu a, pričom b $\neq$ a. Reflexívny a tranzitívny uzáver relácie $\vdash$ označený $\vdash^*$ určuje pre a $\vdash^*$ b, že b je *podstavom* stavu a.

c) Σ je konečná *množina udalostí*.

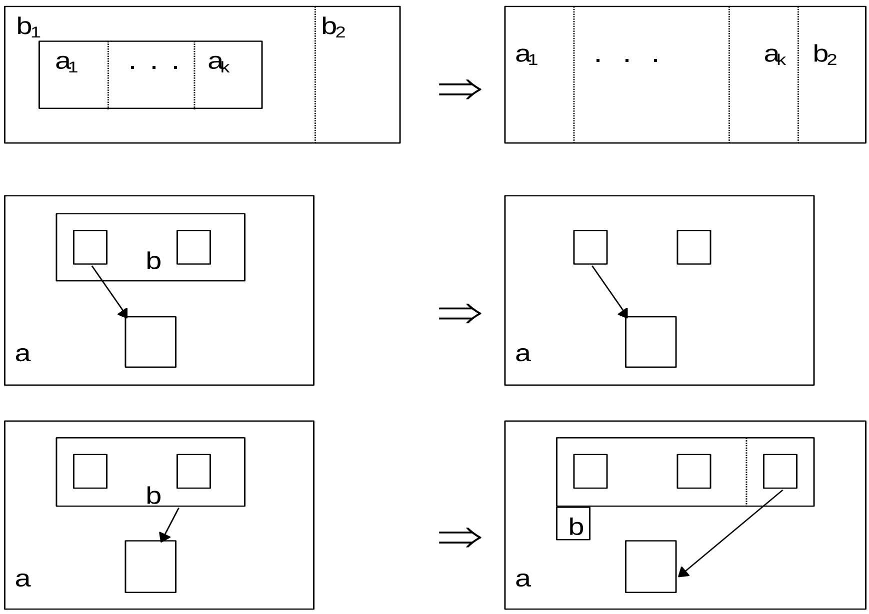
d) Konfigurácia:

Nech q je skupina (množina) stavov (q $\subseteq$ A).

- Stav a je *nadstav skupiny* q ak každý element q je podstavom a (t.j. ak $\forall$ $b \in q$: $a \vdash^* b$), napr. h a k sú nadstavy skupiny $\langle b, e \rangle$.
- *Najnižší nadstav skupiny* q, označený LS(q), je nadstav a skupiny q taký, že pre každý nadstav b skupiny q platí, že $b \vdash^* a$, napr. LS($\langle b, e \rangle$) = h.
- Dva stavy a_1 a a_2 sú *ortogonálne*, označené $a_1 \perp a_2$ ak buď $a_1 = a_2$ alebo žiadny z nich nie je nadstavom druhého a LS($\langle a_1, a_2 \rangle$) $\in A^{\perp}$.
- Skupina stavov q je *ortogonálna* ak každá dvojica stavov $a_1, a_2 \in q$ je ortogonálna. Ortogonálna skupina stavov sa nazýva *konfigurácia*. Napr. $\langle b, e \rangle$ je ortogonálna skupina, t.j. konfigurácia, ale $\langle b, a \rangle$ a $\langle b, i \rangle$ nie. Intuitívne, konfigurácia je skupina stavov v ktorých môže HSM byť súčasne.
- *Základná konfigurácia* je konfigurácia, ktorej všetky prvky sú základné stavy.
- Nech q je konfigurácia a nech a je nadstav q. Potom q je *úplná konfigurácia* stavu $a \in A$ ak platí: $\forall$ $b \in A$: $a \vdash^* b \Rightarrow \langle q, b \rangle$ nie je ortogonálna skupina. Množina všetkých úplných konfigurácií stavu a je označená Q_a. Ak pre konfiguráciu stavu q neplatí predchádzajúca podmienka, potom q je *čiastočná konfigurácia* stavu a. Napr. $\langle b \rangle$ je čiastočná a $\langle b, e \rangle$ úplná konfigurácia stavu h aj stavu k (a teda aj HSM) na obr 3.1.

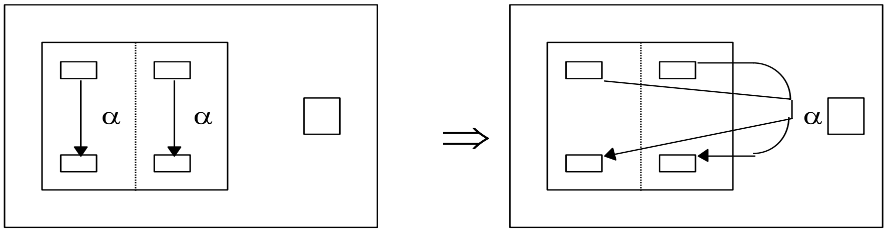
e) T je *množina prechodových ciest* $T = \bigcup_{a \in A^+} T^a$, kde T^a je množina prechodových ciest združená s OR-stavom a. Prechodová cesta OR-stavu a je trojica $t = (u, \sigma, v)_a$, kde u a v sú konfigurácie stavu a nazývané *zdrojová* a *cieľová konfigurácia prechodovej cesty* t a $\sigma \in \Sigma$ je označenie prechodovej cesty. Navyše platí $\forall\, t \in T\ \exists!\, a \in A^+$: $t \in T^a$.

f) ρ je *defaultová funkcia* určujúca pre každý OR-stav $a \in A^+$ *defaultový bezprostredný podstav* označený $\rho(a)$.



Obrázok 3.2. Koncepty konverzií HSM na HSM s alternujúcou štruktúrou

HSM je nazývaný *asynchrónnym* (AHSM) ak platí: $\forall$ a, b $\in A^+$, a $\neq$ b: $(u, \sigma, v) \in T^a \wedge (u', \sigma', v') \in T^b \Rightarrow \sigma \neq \sigma'$, t.j. žiadne dva rozličné stavy nemajú prechodovú cestu označenú rovnakým symbolom. AHSM vylučuje možnosť modelovať interakcie medzi ortogonálnymi zložkami pomocou synchronizačného formalizmu, v ktorom prechodové cesty označené rovnako prebehnú súčasne. Tento typ synchronizácie však môže byť v AHSM modelovaný prechodovými cestami vyššej úrovne, ako je ilustrované na obr. 3.3.



Obrázok 3.3. Konverzia synchronizácie pomocou zdieľaných udalostí na synchronizáciu prechodovými cestami vyššej úrovne.

Definícia 3.2: *Defaultová konfigurácia* stavu $a \in A$, označená $\hat{\rho}(a)$, je definovaná ako základná konfigurácia získaná induktívne nasledovným spôsobom:

- Pre AND-stav a s bezprostrednými podstavmi $a_1, .., a_k$:

  $\hat{\rho}(a) = \langle \hat{\rho}(a_1), .., \hat{\rho}(a_k) \rangle$.

- Pre OR-stav a s bezprostrednými podstavmi $a_1, .., a_k$:

  $\hat{\rho}(a) = \hat{\rho}(a_i) \Leftrightarrow a_i = \rho(a)$.

- Pre základný stav a: $\hat{\rho}(a) = \langle a \rangle$.

Napr. pre HSM na obr. 3.1 $\hat{\rho}(k) = \hat{\rho}(h) = \langle \hat{\rho}(f), \hat{\rho}(g) \rangle = \langle b, e \rangle$.

Definícia 3.3: Nech $t = (u, \sigma, v) \in T^a$, $a \in A^+$, nech sú označené všetky nadstavy elementov konfigurácie v až po stav a a nech sú pridané k $\langle v \rangle$ defaultové konfigurácie všetkých neoznačených bezprostredných podstavov každého označeného AND-stavu. Potom výsledná konfigurácia sa nazýva *explicitná cieľová konfigurácia prechodovej cesty* t.

Na obr. 3.1 uvažujme prechodovú cestu $t = (\langle i \rangle, \phi, \langle d \rangle)$ patriacu stavu k. Ak sa v stave i stane udalosť ϕ, HSM sa dostane do stavu h, a to znamená do stavu f a g. Keďže žiadny podstav stavu f nie je cieľovou konfiguráciou prechodovej cesty t špecifikovaný, je vybraný defaultový bezprostredný podstav stavu f. Takto, explicitná cieľová konfigurácia prechodovej cesty t je konfigurácia $\langle b, d \rangle$.

Definícia 3.4: HSM je v *kanonickej forme* ak:

- HSM má alternujúcu štruktúru.
- Pre každú prechodovú cestu $t = (u, \sigma, v)$ HSM konfigurácie u a v sú základné a cieľová konfigurácia v je daná v explicitnej forme.

Ďalej nech pre skupinu základných stavov q zúženie na stav a, označené $q|_a$, je získané z q vynechaním znakov, ktoré nie sú podstavmi a.

Stav a s množinou bezprostredných podstavov $\{a_1, .., a_k\}$ môže byť s ňou stotožnený zápisom $a \cong a_1 \times .. \times a_k$ pre AND-stav a, a zápisom $a \cong \bigcup_{i=1}^{k} a_i$ pre OR-stav a.

Prechodové cesty AHSM môžu byť interpretované nasledovne.

Poznámka: Pre zjednodušenie predpokladajme ďalej, že AHSM je v kanonickej forme.

Prechod označený $\sigma \in \Sigma$ je definovaný v konfigurácii $q \in Q = Q_r$ vtedy a len vtedy ak existuje prechodová cesta $t = (u, \sigma, v)_b$, $b \in A^+$ taká, že $u \subseteq q$. Výskyt udalosti σ potom spôsobí zmenu konfigurácie q na konfiguráciu $p = (q \setminus q|_b, v)$ (\ označuje rozdiel množín).

Takto prechodová cesta $t = (\langle d \rangle, \theta, \langle j \rangle)_k$ HSM na obrázku 3.1 je definovaná v konfiguráciach $\langle b, d \rangle$ a $\langle a, d \rangle$, nie však v $\langle b, e \rangle$. Výskyt udalosti θ v $q = \langle b, d \rangle$ zmení konfiguráciu na $p = \langle q \setminus q|_k, j \rangle = \langle j \rangle$ (keďže $q = q|_k$).

Takto je možné s každým stavom $a \in A$ asociovať *prechodovú funkciu* δ_a: $Q_a \times \Sigma \to 2^{Q_a}$ takú, že $\forall\, q, p \in Q_a, \forall\, \sigma \in \Sigma$: $p \in \delta_a(q, \sigma) \Leftrightarrow$

$\exists\, b \in A^+$: $a \vdash^* b \wedge \exists\, t = (u, \sigma, v)_b \in T^b$: $u \subseteq q \wedge p = \langle q \setminus q|_b, v\rangle$.

Prechodová funkcia AHSM δ_{AHSM} je daná prechodovou funkciou δ_r koreňového stavu r.

Prechodová funkcia δ_a stavu $a \in A$ môže byť počítaná induktívne (začínajúc základnými stavmi) nasledovným spôsobom:

- Pre základný stav a: $\forall\, \sigma \in \Sigma$: $\delta(\langle a\rangle, \sigma) = \emptyset$.
- Pre OR-stav $a \cong \bigcup_{i=1}^{k} a_i$: $\forall\, q, p \in Q_a, \forall\, \sigma \in \Sigma$: $p \in \delta_a(q, \sigma) \Leftrightarrow$

  $\exists\, i \le k$ také, že $p \in \delta_{a_i}(q, \sigma) \vee \exists\, (u, \sigma, v)_a \in T^a$ taká, že $u \subseteq q \wedge v = p$.
- Pre AND-stav $a \cong a_1 \times .. \times a_k$: $\forall\, q, p \in Q_a, \forall\, \sigma \in \Sigma$: $p \in \delta_a(q, \sigma) \Leftrightarrow$ $\exists\, i \le k$ také, že $p|_{a_i} \in \delta_{a_i}(q|_{a_i}, \sigma) \wedge \forall\, j \ne i$: $p|_{a_j} = q|_{a_i}$.

AHSM = (A, $\vdash$, Σ, T, ρ) môže byť interpretovaný ako zariadenie štartujúce v konfigurácii $q_0 = \hat{\rho}(r)$ vykonávajúce prechody medzi konfiguráciami podľa prechodovej funkcie $\delta = \delta_r$.

AHSM môže byť teda reprezentovaný (nedeterministickým) automatom bez výstupnej funkcie SM = (S, E, δ_{SM}, s_0), ktorého množinu stavov tvorí množina úplných konfigurácií HSM, t.j. S = Q; množina udalostí je totožná, t.j. E = Σ; prechodová funkcia $\delta_{SM} = \delta_{AHSM}$; a počiatočný stav je rovný defaultovej konfigurácii, t. j. $s_0 = q_0 = \hat{\rho}(r)$.

4. Modelovanie DEDS pomocou predikátov a predikátových transformerov.

V tejto kapitole sú uvedené základné definície z teórie predikátového počtu (Dijkstra, 1976; Dijkstra a Scholten, 1990), ktoré umožňujú využiť pri modelovaní DEDS priamo jeho vlastnosti (v stavovom priestore) (Kumar a kol, 1993).

Označme P množinu všetkých predikátov definovaných na ľubovoľnej množine X, t.j. $P = [X \rightarrow \{0, 1\}]$.

S každým predikátom $p \in P$ môže byť združená množina $X_p \subseteq X$ na ktorej predikát p platí, t.j. má hodnotu 1 (TRUE), formálne pre ľubovoľné $p \in P$: $X_p = \{x \mid x \in X \wedge p(x) = 1\}$.

Ak je daná ľubovoľná množina $X' \subseteq X$, potom X' môže byť združená s predikátom $p_{X'} \in P$, kde $\forall\, x \in X$: $p_{X'} = 1 \Leftrightarrow x \in X'$.

Takto, množina predikátov P môže byť združená s množinou 2^X všetkých podmnožín množiny X.

Nech $p \in P$ je predikát na množine X.

Potom jeho *negácia* označená $\neg p$ je predikát definovaný nasledovne: $\forall\, x \in X$: $\neg p(x) = 1 \Leftrightarrow p(x) = 0$. Negáciu môžeme chápať ako unárnu operáciu na P.

Definujme ďalej konjunkciu a disjunkciu na množine P pomocou štandardnej konjunkcie a disjunkcie na množine {0, 1}.

Nech *konjunkcia* $\wedge$: P x P $\rightarrow$ P je binárna operácia na P taká, že $\forall\, p, p' \in P$: $p \wedge p' = p''$, pričom platí $\forall\, x \in X$: $p''(x) = 1 \Leftrightarrow p(x) \wedge p'(x) = 1$. Vidíme, že $X_{p''} = X_{p''} \cap X_{p''}$.

Nech *disjunkcia* $\vee$: P x P $\rightarrow$ P je binárna operácia na P taká, že V p, p' $\in$ P: $p \vee p' = p''$, pričom platí $\forall$ x $\in$ X: $p''(x) = 1 \Leftrightarrow p(x) \vee p'(x) = 1$. Vidíme, že $X_{p''} = X_{p''} \cup X_{p''}$.

Symboly *true, false* označujú predikáty dané nasledovne: $\forall$ x $\in$ X: *true*(x) = 1, $\forall$ x $\in$ X: *false*(x) = 0. Je zrejmé, že *true* = $\neg$*false*. Predikát *true* môže byť združený s množinou X a predikát *false* s prázdnou množinou $\emptyset$.

Definujme binárnu reláciu $\leq$ na P nasledovne: $\forall$ p, p' $\in$ P: $p \leq p' \Leftrightarrow p \wedge p' = p$ (t.j. $p \vee p' = p'$). Ak p $\leq$ p', potom vravíme, že p je silnejší ako p' a tiež, že p' je slabší ako p. Je zrejmé, že $\leq$ je reflexívna, tranzitívna a antisymetrická, t.j. (P, $\leq$) je čiastočne usporiadaná množina.

Štruktúra (P, $\wedge$, $\vee$, $\neg$) je izomorfná s potenčnou množinou 2^X a jej operáciami prieniku, zjednotenia a doplnku, a teda (pozri napr. (Preparata a Yeh, 1974; Bbirkhoff a Bartee, 1970; MacLane a Birkhoff, 1967) tvorí booleovu algebru (t. j. ohraničený, distributívny a komplementárny zväz), pričom relácia $\leq$ na P je izomorfná s reláciou podmnožiny $\subseteq$ na X.

Nech T označuje množinu všetkých predikátových transformerov,

t.j. T = [P $\rightarrow$ P].

Nech $f \in$ T. Potom *negácia f* označená $\neg f$ je transformer definovaný nasledovne: $\forall$ $p \in$ P: $(\neg f)(p) = \neg(f(p))$. Podobným spôsobom je možné definovať operácie *konjunkcie* a *disjunkcie* transformerov:

$\forall f, f' \in$ T: $\forall p \in$ P: $(f \wedge f')(p) = f(p) \wedge f'(p)$, $(f \vee f')(p) = f(p) \vee f'(p)$.

Definícia 4.1: Nech SM = (S, E, δ, s_0) je automat bez výstupnej funkcie. Nech P je množina predikátov definovaných na S. Pre každé e $\in$ E predikátový transformer sp_e: P $\rightarrow$ P je definovaný nasledovne: $\forall$ p $\in$ P: $sp_e(p) = q$, kde množina, na ktorej platí q, $S_q = \{s \mid s \in S \wedge \exists\, s' \in S_p$ taký, že $\delta(s', e) = s\}$. Predikátový transformer sp: P $\rightarrow$ P je daný rovnosťou $sp = \bigvee_{e \in E} sp_e$.

To znamená, že $sp_e(p)$ je predikát, ktrorý platí na množine stavov, ktoré sú dosiahnuté vykonaním udalosti e z niektorého zo stavov, v ktorých platí predikát p.

Definícia 4.2: Nech SM = (S, E, δ, s_0) je automat bez výstupnej funkcie a nech P je množina predikátov definovaných na S. Pre každé e $\in$ E predikátový transformer wlp_e: P $\rightarrow$ P je definovaný nasledovne: $\forall$ p $\in$ P: $wlp_e(p) = q$, kde množina, na ktorej platí q, $S_q = \{s \mid s \in S \wedge \delta(s, e) \in S_p \vee [s, e] \notin D(\delta)\}$. Predikátový transformer wlp: P $\rightarrow$ P je daný rovnosťou $wlp = \bigwedge_{e \in E} wlp_e$.

Predikát $wlp_e(p)$ platí v stavoch, kde buď prechod e nie je definovaný, alebo prechod e z nich vedie do niektorého zo stavov, v ktorom platí predikát p.

Predikátový transformer *sp* je nazývaný *najsilnejšia post-podmienka* (strongest post-condition) a transformer *wlp* je nazývaný *najslabšia liberálna pred-podmienka* (weakest liberal precondition). V (Kumar a kol., 1993) je definovaná dualita transformerov a je ukázané, že predikáty *sp* a *wlp* sú duálne. To znamená, že jeden z predikátov - buď *sp* alebo *wlp* - môže byť definovaný ako základný a zvyšný môže byť z neho odvodený.

Z predchádzajúceho úvodu do predikátového počtu a predikátových transformerov je zrejmé, že DEDS (modelovaný automatom), môže byť reprezentovaný v termínoch predikátov a predikátových transformerov.

Definícia 4. 3: Nech SM = (S, E, δ, s_0) je automat bez výstupnej funkcie. Potom štruktúru (P, E, *sp*, *I*), kde:

- P je množina predikátov definovaných na množine S;
- *sp* je najsilnejšia post-podmienka automatu SM daná definíciou 4.1;
- $I \in P$ je predikát, ktorý platí práve v stave s_0;

nazývame predikátovým modelom automatu SM.

Príklad 4.1: Nech SM = (S, E, δ, s_0), kde $S = R^2$ (R označuje množinu reálnych čísel), E = {e}, $\forall$ s = [x, y] $\in R^2$: δ([x, y], e) = x + y, x - y, a nech predikát $p([x, y]) = (x + y \geq 10 \wedge x > y)$. Vypočítajme hodnotu $sp_e(p)$:

Označme $[x_s, y_s]$ stav pred vykonaním udalosti e a $[x_n, y_n]$ stav po vykonaní udalosti e, t.j. $\delta([x_s, y_s], e) = [x_n, y_n] = [x_s + y_s, x_s - y_s]$ a teda $x_s = (x_n + y_n)/2$, $y_s = (x_n - y_n)/2$.

Predikát $sp_e(p)$ má platiť v každom stave $[x_n, y_n]$ získanom prechodom e zo stavu $[x_s, y_s]$, v ktorom platil predikát *p*, t.j.

$sp_e(p)([x, y]) = sp_e(x_s + y_s \geq 10 \wedge x_s > y_s) = ((x_n + y_n)/2 + (x_n - y_n)/2 \geq 10 \wedge (x_n + y_n)/2 > (x_n - y_n)/2) = (x_n \geq 10 \wedge y_n > 0) = (x \geq 10 \wedge y > 0)$. Inými slovami, pre dvojice reálnych čísel, v ktorých platí predikát $p([x, y]) = (x + y \geq 10 \wedge x > y)$ bude po zmene stavu danej priradeniami x = x + y, y = x - y platiť predikát $sp_e(p)([x, y]) = (x \geq 10 \wedge y > 0)$, čo môžeme zapísať nasledovne:

$sp_e(p) = sp_{x = x + y,\, y = x - y}(x + 10 \geq 10 \wedge x > y) = (x \geq 10 \wedge y > 0)$.

5. Vnútorný popis DEDS, základy relačného modelovania DEDS a algebraické zovšeobecnenie Petriho sietí

Teória logického modelovania systémov diskrétnych udalostí predstavuje prudko sa rozvíjajúcu oblasť. Vychádza z teórie automatov a formálnych jazykov, ktoré sú považované za základný a zároveň najjednoduchší modelovací formalizmus pre opis DEDS. Existuje tiež široká paleta iných modelovacích formalizmov, spomeňme za všetky napr. formalizmy uvedené v predchádzajúcich kapitolách. Spoločnou črtou všetkých týchto formalizmov je snaha umožniť efektívnejší opis a/alebo analýzu a syntézu DEDS. Prirodzene, táto snaha by nemala byť zastavená. Existujú minimálne dva principiálne spôsoby v pokračovaní spomenutého rozvoja. Jedným z nich je snaha nájsť principálne nový opis, nástroje a prostriedky poskytujúce jedinečnú množinu nových výhod. Ďalšou možnosťou je hľadať spôsoby rozšírenia existujúcich formalizmov a nástrojov tak, aby sa zachovali výhody, ktoré tieto formalizmy poskytujú a tak sa umožnila modelovať rovnako efektívne širšia paleta systémov.

Táto kapitola vychádza zo štúdia DEDS a ich modelovacích formalizmov. Prezentuje dve konkrétne možnosti rozvoja oblasti opisu a modelovania DEDS - po jednom z oboch spomenutých spôsobov. Prvý je popísaný v podkapitole 5.1 a druhý v podkapitole 5.2.

Podkapitolu 5.1 tvoria dve logické časti. Prvá z týchto častí vychádza z článku (Juhás, 96a). DEDS je v nej formálne definovaný ako systém diskrétnych stavových zmien (DSCS) vychádzajúc z vnútorného popisu systému v kontexte s poznatkami všeobecnej teórie systémov. Taktiež je formalizovaná podmienka logického vnútorného opisu DSCS (DEDS). Je možné spomenúť, že formálna definícia DEDS často v literatúre chýba. Jednu z mála formálnych definícií DEDS uvedenú v kontexte s všeobecnou teóriou systémov je možné nájsť v (Kozák, 1992), kde je DEDS definovaný na úrovni vonkajšieho opisu (vstupno-výstupného).

Druhá časť podkapitoly 5.1 prezentuje základy relačného kalkulu pre logické modelovanie DSCS zavedeného v článkoch (Juhás a Kocian, 1996, 1995a), ktoré sú prispôsobené formálnej definícii DSCS. Relačný kalkul pre DEDS vychádza z myšlienky využitia stavovej zmeny v istom zmysle podobne ako CVDS využíva diferenciál. Z pohľadu predikátového počtu (pozri kapitolu 4) je možné chápať prezentovaný relačný kalkul ako snahu vyjadriť dynamiku DEDS (detailnú, možnú, požadovanú atď.) množinou predikátov na karteziánskom súčine S x S (kde S predstavuje stavovú množinu systému), t.j. vyjadriť dynamiku DEDS pomocou vlastností stavových zmien.

Podkapitola 5.2 prezentuje pokus o rozšírenie už existujúceho formalizmu, t.j. Petriho sietí. Zaoberá sa možnosťou algebraického zovšeobecnenia Petriho sietí a farebných Petriho sietí. Vychádza z článkov (Juhás, 1996b,c) a nadväzuje priamo na kapitolu 2. Uvedené zovšeobecnenie umožňuje použiť pre stavy jednotlivých pozícií nielen prvky nezápornej časti okruhu celých čísel, ale prvky podmnožiny ľubovoľnej komutatívnej grupy, pričom pri analytickom vyjadrení dynamiky Petriho siete operácia danej grupy nahrádza operáciu sčítania celých čísel. Algebraicky zovšeobecnená definícia zachováva základné výhody Petriho sietí. Štandardné Petriho siete môžu byť takto chápané ako špeciálny prípad uvedenej definície.

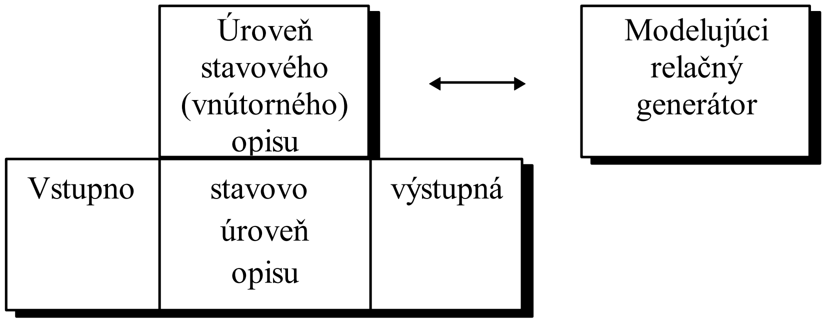
5.1 Vnútorný prístup k popisu systémov a jeho špecifikácia pre relačné modelovanie DEDS

Ako bolo spomenuté v úvodnej kapitole, existuje mnoho formálnych prístupov k popisu všeobecných systémov.

Ako napísal zakladateľ všeobecnej teórie systémov von Bertalanffy v (1972), existujú dva principiálne spôsoby opisu dynamických systémov: vnútorný a vonkajší. Stručne, vnútorný opis definuje systém pomocou stavových premenných a zmeny systému sú vyjadrené trajektóriami, ktoré vytvára pohyb stavových premenných v stavovom priestore, čo je priestor (množina) možných umiestnení (hodnôt) stavových premenných. Pri vonkajšom opise je systém uvažovaný ako "čierna skrinka" - opis systému je daný v termínoch vstupov a výstupov a prenosovej funkcie medzi nimi. Inými von Bertalanffyho slovami, vnútorný popis je "štrukturálny", t.j. pokúša sa popísať správanie systému v pojmoch stavových premenných a ich vzájomného vzťahu, a vonkajší popis je "funkcionálny", správanie systému je dané v pojmoch vzťahu s okolím.

Teórie spomenuté v úvodnej kapitole, podobne ako iné, kombinujú vonkajší a vnútorný popis systému, t.j. používajú (alebo umožňujú použiť) pojmy vstupu, stavu, a výstupu. Prístup prezentovaný v tejto kapitole, je založený na myšlienke, že na určitej úrovni rozlíšenia a riešenia systémových problémov nie je nutné uvažovať vstupy a výstupy, ale iba pozorovať možné stavové trajektórie. Zovšeobecnením tejto myšlienky môžeme získať *viacúrovňový opis systému*, kde na jednotlivých úrovniach sú použité špecifické pojmy a úrovne sú prepojené vzájomnými vzťahmi. Takýto opis môže redukovať zložitosť štruktúry a zdá sa, že určité špecifické problémy môžu byť takto ľahšie riešené.

V prezentovanom prístupe môže byť interpretácia úrovne vnútorného stavového opisu popísaná nasledovne. Pozorovateľ sleduje čo sa deje vnútri v "čiernej skrinke" a s plynúcim časom zapisuje hodnoty stavových premenných na papier. Stopa na papieri reprezentuje stavovú trajektóriu systému. Systém je potom charakterizovaný stavovou množinou a množinou všetkých možných stavových trajektórií. Pozorovateľovi sa systém javí ako spontánny generátor stavových zmien bez vstupov a výstupov.



Obrázok 5.1. Reprezentácia prezentovaného prístupu

Pre DEDS, kde udalosti sú charakterizované ako vstupy, aplikácia takéhoto vnútorného opisu znamená, že udalosti sú nahradené priamymi (bezprostrednými) diskrétnymi stavovými zmenami (skokovými zmenami) a na tejto úrovni pojem udalosť nie je vôbec uvažovaný.

Takto DEDS môžu byť zahrnuté medzi tzv. systémy diskrétnych stavových zmien (DSCS), v ktorých stavové zmeny sú diskrétne. Ako model DSCS správania je použitý relačný generátor stavových zmien, jednoduchá množinová štruktúra daná stavovou množinou a reláciou na tejto množine, ktorá spontánne mení svoje stavy podľa danej relácie. Sú tiež prezentované základné definície a pojmy relačného počtu zavedeného Juhásom a Kocianom v (1995a, 1996), ktoré sú prispôsobené formálnej definícii DSCS. Taktiež sú načrtnuté základné koncepty riešenia otázok riadenia založené na supervízorovej teórii riadenia zavedenej Ramadgeom a Wonhamom (pozri napr. Ramadge a Wonham, 1989). Prístup je ilustrovaný na riešení problému bludiska formulovaného v (Ramadge a Wonham, 1989).

5. 1. 1 Vnútorný popis dynamických systémov

Nech S je ľubovoľná neprázdna množina a nech V je množina všetkých funkcií čiastočne definovaných na R (reálne čísla) s hodnotami v S, formálne $V = \{f \mid f\colon R \supseteq D1(f) \to S\}$, $D1(f)$ je definičný obor funkcie f. Nech α: $2^R \times R \to 2^R$ je funkcia taká, že $\forall\ A \in 2^R, \forall\ c \in R$: $\alpha(A, c) = A' = \{a' \mid a' \in R \wedge a' - c \in A\}$. Nech β: $V \times R \to V$ je taká, že $\forall\ [f, c] \in V \times R$: $\beta(f, c) = g$, kde g: $\alpha(D1(f), c) \to V$ je taká, že $\forall a \in D1(f)$: $f(a) = g(a+c)$.

Množina $B \subseteq V$ je nazývaná *prípustnou množinou funkcií* s hodnotami v S vtedy a len vtedy ak pre $\forall\ f \in B$ platia nasledujúce štyri výroky:

1. $0 \in D1(f)$.

2. $\forall\ c \in R$ platí: ak $0 \in \alpha(D1(f), c)$ potom $\beta(f, c) \in B$.

3. $\forall\ a \geq 0$: $f|_{((-\infty, a\rangle \cap D1(f))} \in B$ (symbol $f|_A$, $A \subseteq D1(f)$ označuje zúženie funkcie f na množinu A ($D1(f|_A) = A$, $\forall\ x \in A$: $f|_A(x) = f(x)$).

4. $S = \bigcup_{f \in B} D2(f)$, kde $D2(f)$ je obor hodnôt funkcie f.

Podmienka v druhom výroku je ekvivalentná s požiadavkou $-c \in D1(f)$).

Definícia 5.1.1.1: *Vnútorný opis dynamického systému* je dvojica $\mathcal{S}$ = (S, B), kde:

S je ľubovoľná neprázdna množina;

B je prípustná množina funkcií s hodnotami v S.

Ak $\mathcal{S}$ = (S, B) je vnútorný opis dynamického systému, potom *stavy* systému sú definované ako prvky množiny S (t.j. S je *stavová množina* systému) a *možné stavové trajektórie* systému sú definované ako prvky množiny B. Množina B je nazývaná *možné správanie* systému.

Parameter stavovej trajektórie $f \in$ B reprezentuje čas. Záporné hodnoty času reprezentujú históriu, nula reprezentuje okamžitý stav a kladné hodnoty reprezentujú budúce správanie sa systému.

Stavové trajektórie $f \in$ B pre ktoré neexistuje žiadna dvojica $[g, a] \in B \times R$ pre ktorú $g|((-\infty, a\rangle \cap D1(g)) = f$

sú tiež nazývané *úplné stavové trajektórie* systému S.

Nech pre každé pevné $f \in$ B a každé pevné t $\in$ D1(f) množina B(f, t) je daná ako $B(f, t) = \{g;\ g \in B \wedge g|((-\infty, t\rangle \cap D1(f)) = f|((-\infty, t\rangle \cap D1(f))\}$. Prvky tejto množiny určujú možný vývoj systému po čase t ak do času t sa systém správal alebo bude správať podľa stavovej trajektórie f.

Prvok s $\in$ S je nazvaný *možným počiatočným stavom* vtedy a len vtedy keď $\exists\ f \in B: \exists\ t \in D1(f): (\ f(t) = s \wedge (\ \forall\ t' \in ((-\infty, t) \cap D(f)): f(t') = f(t)))$. Potom pre každý stav s $\in$ S, ktorý je možným počiatočným stavom môže byť definované *správanie s počiatočným stavom* s ako zjednotenie všetkých stavových trajektórií s počiatočným stavom s, ktoré je označené B_s a formálne dané nasledovne: $B_s = \{\ f, f \in B \wedge \exists\ t \in D1(f): (\ f(t) = s \wedge (\ \forall\ t' \in ((-\infty, t) \cap D(f)): f(t') = f(t)))\}$.

Poznámka: Nech S = (S, B) je vnútorný opis otvoreného orientovaného dynamického systému s opisom (napr. podľa (Mesarovic, 1972; Gvozdiak a kol., 1990; Kotek a kol., 1990)) daným štruktúrou (X, U, Y, U, Y, T, δ, ω) kde X je *množina stavov*, U je *množina vstupov*, Y je *množina výstupov*, U je *(prípustná) množina vstupných funkcií* (u: T $\rightarrow$ U), Y je *(prípustná) množina výstupných funkcií* (y: T $\rightarrow$ Y), T je lineárne usporiadaná množina predstavujúca čas (pre zjednodušenie nech T $\subseteq$ R), δ je *stavovo-prechodová funkcia* δ: X x U x T x T $\rightarrow$ X a ω je *výstupná funkcia* ω: X x U x T $\rightarrow$ Y x T taká, že $\omega(\delta(x, u, t, t'), u, t') = (y, t')$. Potom vzťah medzi týmito opismi je daný výrazmi:

1. $X = S$

2. $\forall f \in B\ \exists u \in U: \forall t, t' \in D1(f): f(t') = \delta(f(t), u, t, t')$;

3. $\forall u \in U: \exists f \in B: \forall t, t' \in D1(f): f(t') = \delta(f(t), u, t, t')$;

Navyše, prezentovaný prístup predpokladá, že $T \subseteq R$ nemusí byť konštantná množina, ale môže byť rôzna pre jednotlivé vstupné funkcie, výstupné funkcie a stavové trajektórie (t.j. u: $R \supseteq D1(u) \rightarrow U$ a vo výrazoch 2. a 3. $D1(f) = D1(u)$).

5. 1. 2 Systémy diskrétnych stavových zmien

Nech Z označuje celé čísla. Nech pre $a, b \in Z \cup \{-\infty, \infty\}$ také, že $a \leq b$ je definovaná množina indexov $INDEX_{ab} = \{ i \mid i \in Z \wedge (a \in Z \Rightarrow i \geq a) \wedge (b \in Z \Rightarrow i \leq b)\}$.

Ďalej nech pre ľubovoľnú množinu X označuje symbol X^* množinu všetkých postupností prvkov z množiny X, $X^* = \{\{x_i\}_{i=a}^{b} \mid a, b \in Z \cup \{-\infty, \infty\} \wedge a \leq b \wedge (\forall i \in INDEX_{ab}: x_i \in X)\}$ a nech symbol X_k^* označuje pre každé pevné $k \in Z$ podmnožinu množiny X^* danú nasledovne:

$X_k^* = \{ \{x_i\}_{i=a}^{b} \mid \{x_i\}_{i=a}^{b} \in X^* \wedge a \leq k \wedge b \geq k\}$.

Poznámka: Na rozdiel od kapitoly 1 nepožadujeme konečnosť postupností ani prítomnosť prázdnej postupnosti v množine X^*.

Nech Φ je množina všetkých reálnych intervalov. Postupnosť $\{\phi_i\}_{i=a}^{b} \in \Phi_0^*$ je nazývaná *prípustnou postupnosťou reálnych intervalov* vtedy a len vtedy keď sú platné nasledujúce štyri výroky:

1. $\bigcup_{i \in INDEX_{ab}} \phi_i = R$ (zjednotenie intervalov tvorí množinu reálnych čísel).

2. $\forall (i, j \in INDEX_{ab} \wedge i < j): \forall t \in \phi_i : \forall t' \in \phi_j : t < t'$ (intervaly sú usporiadané a disjunktné).

3. $0 \in \phi_0$.

4. $\forall \langle c, d\rangle \subseteq R$: #(INTERSECTION$_{\langle c, d\rangle}$) $\in Z$, kde

INTERSECTION$_{\langle c, d\rangle}$ = $\{\phi_i ;\ i \in INDEX_{ab} \wedge \phi_i \cap \langle c, d\rangle \neq \emptyset\}$ ($\emptyset$ označuje prázdnu množinu) (táto podmienka znamená, že každý konečný interval má neprázdny prienik s konečným počtom prvkov postupnosti $\{\phi_i\}_{i=a}^{b}$).

Funkcia *f*: $R \supseteq D1(f) \to S$ je nazývaná *funkciou diskrétnych zmien* (po častiach konštantnou funkciou) vtedy a len vtedy keď existuje prípustná postupnosť intervalov $\{\phi_i\}_{i=a}^{b}$ pre ktorú platia nasledujúce tri predikáty:

1. $\forall i \in INDEX_{ab} : D1(f) \cap \phi_i \neq \emptyset$.

2. $\forall i \in INDEX_{ab}: \forall t, t' \in \phi_i \cap D1(f): f(t) = f(t')$.

3. $\forall (i \in INDEX_{ab} \wedge i < b): \forall t \in \phi_i \cap D1(f): \forall t' \in \phi_{i+1} \cap D1(f): f(t) \neq f(t')$.

Prípustná postupnosť intervalov $\{\phi_i\}_{i=a}^{b}$ ktorá pre funkciu diskrétnych zmien *f*: $R \supseteq D1(f) \to S$ spĺňa predchádzajúce tri podmienky je nazývaná *združenou s f*.

Poznámka: Je zrejmé, že pre každé dve prípustné postupnosti intervalov $\{^1\phi_i\}_{i=a}^{b}$ a $\{^2\phi_i\}_{i=c}^{d}$ združené s nejakou funkciou diskrétnych zmien *f* platí: $a = b \wedge c = d \wedge (\forall i \in INDEX_{ab}: {}^1\phi_i \cap D1(f) = {}^2\phi_i \cap D1(f)$.

Postupnosť $q = \{s_i\}_{i=a}^{b} \in S_0^*$ taká, že $\exists$ prípustná postupnosť intervalov $\{\phi_i\}_{i=a}^{b}$ združená s funkciou diskrétnych zmien *f* pre ktorú platí: $\forall i \in INDEX_{ab}: \exists t \in \phi_i \cap D1(f): f(t) = s_i$ je nazývaná *postupnosťou určenou funkciou diskrétnych zmien f*. Z predchádzajúcej poznámky a podmienky 2 v definícii funkcie diskrétnych zmien je zrejmé, že pre ľubovoľnú funkciu diskrétnych zmien postupnosť ňou určená je jednoznačná.

Ak funkcia *f*: R $\supseteq$ D1(*f*) $\rightarrow$ S je funkcia diskrétnych zmien, potom množina

DC(*f*) = {[t, *f*(t), t', *f*(t')] | ($\{\phi_i\}_{i=a}^{b}$ je prípustná postupnosť intervalov združená z *f*) $\wedge$ t $\in \phi_k \cap$ D1(*f*) $\wedge$ t' $\in \phi_{k+1} \cap$ D1(*f*) $\wedge$ k $\in$ $INDEX_{ab}$ $\wedge$ k < b}

je nazývaná *množina priamych diskrétnych zmien funkcie f*. Každý prvok množiny DC(*f*) vyjadruje fakt, že na $\langle$t, t'$\rangle \cap$ D1(*f*) je práve jedna zmena hodnoty funkcie *f* a táto zmena je skokom z hodnoty *f*(t) na hodnotu *f*(t').

Definícia 5.1.2.1: Systém s vnútorným opisom *S* = (S, B) je nazývaný *systémom diskrétnych stavových zmien* (DSCS) vtedy a len vtedy keď každá stavová trajektória *f* $\in$ B je funkciou diskrétnych zmien.

Ak *S* = (S, B) je vnútorný opis DSCS, potom pre každú funkciu *f* $\in$B množina DC(*f*) je nazývaná množinou *priamych diskrétnych stavových zmien trajektórie f* a jej prvky sú nazývané *priame diskrétne stavové zmeny* (alebo kratšie *priame stavové zmeny*).

Nech pre každé pevné *f* $\in$ B a každé pevné t $\in$ D1(*f*) je definovaná množina možných priamych stavových zmien zo stavu *f*(t) v čase t pre stavovú trajektóriu *f* (označená ako DCB(*f*, t)) nasledovne: DCB(*f*, t) = {[t, s, t', s']; $\exists$ *g* $\in$ B(*f*, t): [t, s, t', s'] $\in$ DC(*g*)}.

Pre t < 0 množina DCB(*f*, t) obsahuje priame stavové zmeny, ktoré boli možné pre trajektóriu *f* v čase t, pre t = 0 množina DCB(*f*, t) označuje priame stavové zmeny, ktoré sú možné ak *f* vyjadruje stavovú históriu systému, a pre t > 0 množina DCB(*f*, t) je tvorená priamymi stavovými zmenami, ktoré budú možné v čase t ak sa systém bude správať do času t podľa stavovej trajektórie *f*.

Potom, pre každé pevné $f \in$ B a každé pevné t $\in$ D1(f) je definovaná *množina nasledovníkov* FOLLOWERS(f, t) nasledovne: FOLLOWERS(f, t) = {s' | [t, s, t', s'] $\in$ DCB(f, t)}.

Prvky tejto množiny sú také stavy, že každý z nich môže (ak t = 0) (mohol (ak t < 0) alebo bude môcť (ak t > 0)) priamo nasledovať za stavom f(t). Inými slovami prvky množiny FOLLOWERS(f, t) sú také stavy, že každý z nich môže byť prvým stavom iným ako stav f(t) ktorý nasleduje po stave f(t) ak do času t sa systém správal (alebo bude správať) podľa trajektórie f.

Definícia 5.1.2.2: Nech S = (S, B) je vnútorný opis DSCS. Funkcia l: B $\to S_0^*$ priraďujúca každej $f \in$ B postupnosť určenú funkciou f, je nazývaná *logické zobrazenie správania DSCS*.

V súlade s definíciou DSCS môžu byť formulovaná nasledovné vety o počiatočných stavoch a správaní nimi danom.

Veta 5.1.2.1: Nech S = (S, B) je vnútorný opis DSCS. Stav s $\in$ S je (možným) počiatočným stavom vtedy a len vtedy keď $\exists\, f \in$ B také, že pre jej logické zobrazenie $l(f) = \{s_i\}_{i=a}^{b}$ platí: $a \in Z \wedge s_a = s$.

Veta 5.1.2.2: Nech S = (S, B) je vnútorný opis DSCS a nech stav s $\in$ S je (možným) počiatočným stavom DSCS. Potom $B_s = \{f \mid f \in B \wedge l(f) = \{s_i\}_{i=a}^{b} \wedge a \in Z \wedge s_a = s\}$ je možné správanie s počiatočným stavom s.

Stručne, priame stavové zmeny sú skokové zmeny stavov a DSCS je systém, kde všetky stavové zmeny sú postupnosťou priamych stavových zmien. Stavové trajektórie DSCS sú funkcie diskrétnych zmien, t.j. po častiach konštantné funkcie.

5.1.3 Logický vnútorný opis DSCS

Aby definícia logického vnútorného opisu DSCS bola použiteľná, je prirodzené požadovať aby pre každé dve stavové trajektórie f, $f' \in$ B, pre ktoré existujú t $\in$ D1(f), t' $\in$ D1(f ') také, že logický obraz trajektórie $f|_{(\langle -\infty, t\rangle \cap D1(f))}$ je rovný logickému obrazu trajektřie $f'|_{(\langle -\infty, t'\rangle \cap D1(f'))}$, množiny nasledovníkov (f, t) a (f ', t') boli totožné, t.j. aby možný logický vývoj z f(t) a f '(t') bol rovnaký (t.j. $D2(l|_{B(f,\ t)}) = D2(l|_{B(f',\ t')})$.

Definícia 5.1.3.1: Nech S = (S, B) je vnútorný opis DSCS s logickým zobrazením l: B $\to S^*_0$ taký, že platí: $\forall f, f' \in B: \forall t \in D1(f), \forall t' \in D1(f')$:

$l(f|_{(\langle -\infty, t\rangle \cap D1(f))}) = l(f'|_{(\langle -\infty, t'\rangle \cap D1(f'))}) \Rightarrow D2(l|_{B(f, t)}) = D2(l|_{B(f', t')})$.

Potom dvojica LS = (S, LB), kde LB = D2(l), je nazývaná *logický vnútorný opis DSCS* s vnútorným opisom S = (S, B).

Poznámka: Ak S = (S, B) je vnútorný opis DSCS s logickým zobrazením l: B $\to S^*_0$ potom predikát ($\forall f, f' \in B: \forall t \in D1(f), \forall t' \in D1(f')$: $l(f|_{(\langle -\infty, t\rangle \cap D1(f))}) =$

$l(f'|_{(\langle -\infty, t'\rangle \cap D1(f'))}) \Rightarrow D2(l|_{B(f, t)}) = D2(l|_{B(f', t')})$) je platný vtedy a len vtedy ak predikát ($\forall f, f' \in B: \forall t \in D1(f), \forall t' \in D1(f')$:

$l(f|_{(\langle -\infty, t\rangle \cap D1(f))}) = l(f'|_{(\langle -\infty, t'\rangle \cap D1(f'))}) \Rightarrow$

FOLLOWERS(f, t) = FOLLOWERS(f ', t')) je platný.

Rovnosť $D2(l|_{B(f,\ t)}) = D2(l|_{B(f',\ t')})$ v implikácii z predchádzajúcej definície je ekvivalentná z rovnosťou FOLLOWERS(f, t) = FOLLOWERS(f ', t').

Ďalej nech *LS* = (S, LB) je logický vnútorný opis DSCS s vnútorným opisom *S* = (S, B). Potom prvky LB sú nazývané *logické stavové trajektórie* DSCS. Množina LB je nazývaná *možné logické správanie*. Ak s $\in$ S je možný počiatočný stav DSCS, potom *možné logické správanie s počiatočným stavom s* je označené LB_s a definované rovnosťou $LB_s = D2(l|_{Bs})$.

Nech l: B $\to$ S_0^* je logické zobrazenie správania DSCS. Ak $f \in$ B je úplná stavová trajektória potom $l(f)$ je nazývaná *úplná logická stavová trajektória*.

Pre každé pevné $q = \{s_i\}_{i=a}^{b} \in$ LB množina LDC(q) = $\{ [s_i, i, s_{i+1}, i + 1] \mid i \in INDEX_{ab} \wedge i < b\}$ je nazvaná množinou *(logických) priamych stavových zmien logickej stavovej trajektórie q*.

Pre každé pevné $q = \{s_i\}_{i=a}^{b} \in$ LB a každé pevné j $\in INDEX_{ab}$ množina LB(q, j) je daná ako LB(q, j) = $\{\{r_i\}_{i=a}^{c} \mid \{r_i\}_{i=a}^{c} \in LB \wedge j \le c \wedge \{r_i\}_{i=a}^{j} = \{s_i\}_{i=a}^{j}\}$.

Prvky tejto množiny určujú možný vývoj systému zo stavu s_j ak sa systém do logického kroku j správal (alebo bude správať) podľa logickej trajektórie q.

Nech pre každé pevné $q = \{s_i\}_{i=a}^{b} \in$ LB a každé pevné j $\in INDEX_{ab}$ je definovaná množina možných priamych stavových zmien zo s_j v kroku j pre logickú stavovú trajektóriu q nasledovne: LDCB(q, j) =

$\{[s , j,s', j+1] \mid s = s_j \wedge \exists\, r \in LB(q, j): [s, j, s', j+1] \in LDC(r)\}$.

Potom, pre každé pevné $q = \{s_i\}_{i=a}^{b} \in$ LB a každé pevné j $\in INDEX_{ab}$ je definovaná *množina logických nasledovníkov* LFOLLOWERS(q, j) = $\{s' \mid [s, j, s', j+1] \in LDCB(q, j)\}$.

Je zrejmé, že ak $\{\phi_i\}_{i=a}^{b}$ je prípustná postupnosť intervalov združených z $f \in$ B, potom $\forall$ j $\in INDEX_{ab}$: $\forall$ t $\in \phi_j$: LB($l(f)$, j) = $D2(l|_{B(f,\ t)})$ a LFOLLOWERS($l(f)$, j) = FOLLOWERS(f, t).

V súlade z predchádzajúcimi definíciami vnutorný opis DSCS môže byť nazvaný tiež *časovaný vnútorný opis* aby sa zdôraznil rozdiel medzi ním a logickým vnútorným opisom.

5.1.4 Logický relačný generátor stavových zmien

Nech P je ľubovoľná množina a L je ľubovoľná binárna relácia na P ($L \subseteq P \times P$). Nech L(p, .) je relácia daná tak, že $L(p, .) = L \cap (\{p\} \times P)$ a nech $L(., p) = L \cap (P \times \{p\})$.

Support binárnej relácie L je daný zjednotením prvého a druhého oboru relácie a je označený ako L_{supp} ($L_{supp} = \{p \mid \exists\, p': [p, p'] \in L \vee [p', p] \in L\}$). Obraz prvku $p \in P$ v relácii L je označený symbolom L(p) ($L(p) = \{p' \mid [p, p'] \in L\}$). Binárna relácia $L^k = \{ [p_1, p_{k+1}] \mid \exists \{p_i\}_{i=1}^{k+1} : [p_i, p_{i+1}] \in L$ pre $\forall\ i \in \{1, ..., k\}\}$ (k je kladné celé číslo), je nazývaná k-ta mocnina relácie L. Symbol $L^{\otimes}$ označuje binárnu reláciu, ktorá je zjednotením všetkých mocnín relácie L ($L^{\otimes} = \bigcup_{k \in Z^+} L^k$), t.j. tranzitívny uzáver relácie L.

Pojem L-*postupnosť* označuje postupnosť $\{p_i\}_{i=a}^{b} \in P^*$, pre ktorú sú splnené nasledujúce dve podmienky:

1. $\forall\ (i \in INDEX_{ab} \wedge i < b): [p_i, p_{i+1}] \in L$
2. $b \neq \infty \Rightarrow L(p_b, .) = \varnothing$.

Postupnosť, pre ktorú platí iba prvá z podmienok, je nazývaná *časť* L-*postupnosti*.

Nech $p = \{p_i\}_{i=a}^{b} \in P^*$ je postupnosť taká, že existuje nejaká L-postupnosť $q = \{p_{k_j}\}_{j=c}^{d} \in P^*$ vybraná z postupnosti *p* a navyše platia podmienky:

1. $a \neq -\infty \Rightarrow k_c = a$
2. $a = -\infty \Rightarrow c = -\infty$

3. $b \neq \infty \Rightarrow k_d = b$

4. $b = \infty \Rightarrow d = \infty$

Potom *p* je označená symbolom (L)-*postupnosť*.

Ak *q* je časť L-postupnosti, potom *p* je nazývaná *časťou* (L)-*postupnosti*.

Nech symbol ${}^{\{p}L$ označuje pre stav $p \in P$ binárnu reláciu ${}^{\{p}L = L(p,.) \cup \{[p', p'']; [p', p''] \in L \wedge [p, p'] \in L^{\otimes}\}$.

Definícia 5.1.4.1: *Logický relačný generátor stavových zmien* je dvojica *G* = (P, F) kde:

P je ľubovoľná neprázdna množina;

F je binárna relácia na P ($F \subseteq P \times P$).

Ak *G* = (P, F) je logický relačný generátor stavových zmien (ďalej krátko generátor), potom: *stavy generátora G* sú definované ako prvky množiny P; F je nazývaná *reláciou priamych stavových zmien generátora G*; prvky relácie F sú nazývané *možné priame stavové zmeny G*; *možné logické správanie G* je definované ako LB = $\{ \{p_i\}_{i=a}^{b} ; \{p_i\}_{i=a}^{b} \in P_0^* \wedge \{p_i\}_{i=a}^{b}$ je časť F-postupnosti$\}$; *Možné logické stavové trajektórie generátora G* sú definované ako prvky množiny LB generátora *G* (t.j. ako časti F-postupností patriacich do P_0^*). Logické stavové trajektórie, ktoré sú F-postupnosti, sú tiež nazývané *úplnými logickými stavovými trajektóriami generátora G*. Pre každý stav $p \in P$ množina $LB_p = \{ \{p_i\}_{i=a}^{b} \mid \{p_i\}_{i=a}^{b} \in LB \wedge a \in Z \wedge p_a = p\}$ je nazývaná *logické správanie generátora G s počiatočným stavom* p. Stav $p' \in P$ je *dosiahnuteľný zo stavu* $p \in P$ v *G* vtedy a len vtedy ak $[p, p'] \in F^{\otimes}$ (t.j. vtedy a len vtedy ak $\exists\, k \in Z^+$: $[p, p'] \in F^k$). Binárna relácia $F^{\otimes}$ je nazývaná *reláciou dosiahnuteľnosti.*

Definícia 5.1.4.2: Nech G = (P, F) je generátor. Stav p'$\in$ P je *živý vzhľadom na stav* p $\in$ P v generátore G vtedy a len vtedy ak pre každú možnú logickú stavovú trajektóriu $\{p_i\}_{i=a}^{b} \in$ LB platí výrok: $\exists\, k \le b$: $p_k = p \Rightarrow \exists\, j, k < j \le b$: $p_j = p'$. Ak p' je živý vzhľadom ku každému stavu z množiny stavov P, potom p' *je živý v G* a ak je v G živý každý stav z P, potom vravíme, že *generátor G je živý*.

Definícia 5.3: Nech P = X x X^*, kde X je ľubovoľná množina. Nech F je binárna relácia na P taká, že pre každú dvojicu [[x, q], [x', q']] $\in$ F platí, že ak $q = \{x_i\}_{i=a}^{b}$ a b $\in$ Z, potom $q' = \{x_i\}_{i=a}^{b+1}$, pričom $x_{b+1} = x'$. Potom generátor G = (P, F) je nazývaný *logický relačný generátor stavových zmien s pamäťou*, množina X je nazývaná *rýdza množina stavov* a množina X^* je nazývaná *pamäťovou stavovou množinou* generátora G. Logická stavová trajektória $\{p_i\}_{i=a}^{b}$ (a $\in$ Z) generátora s pamäťou G taká, že p_a = [x, x] $\in$ X x X je nazývaná *logická stavová trajektória s rýdzim počiatočným stavom* x *a s resetovanou pamäťou*.

5.1.5 Modelovanie a riadenie DSCS pomocou logických relačných generátorov

Z dôvodu rozlíšenia logického správania modelovaného DSCS a modelujúceho generátora budú príslušné množiny označené menom systému, resp. generátora v hornom indexe (napr. logické správanie generátora G = (P, F) bude označené LB^{G}).

Definícia 5.1.5.1: Nech LS = (S, LB) je logický vnútorný opis DSCS a nech G = (P, F) je generátor taký, že existuje injekcia α: S $\rightarrow$ P pre ktorú platí nasledujúci výrok: $(\forall q = \{s_i\}_{i=a}^{b} \in LB^{LS}$: $\forall\, j \in INDEX_{ab}$: LFOLLOWERS($q$, j) = F($\alpha(s_j)$)).

Potom množina trajektórií generátora G daná množinou $\mathrm{MODEL}^{LS} = \{\{\alpha(s_i)\}_{i=a}^{b} \mid \{s_i\}_{i=a}^{b} \in \mathrm{LB}^{LS}\}$ je nazývaná *priamy model logického správania DSCS*.

Nasledujúca veta prezentuje nutnú a postačujúcu podmienku možnosti priameho modelovania DSCS.

Veta 5.1.5.1: Nech LS = (S, LB) je logický vnútorný opis DSCS. Generátor G = (P, F), taký, že existuje injekcia α: S $\rightarrow$ P pre ktorú platí

$(\forall q = \{s_i\}_{i=a}^{b} \in \mathrm{LB}^{LS}\colon \forall\ j \in \mathrm{INDEX}_{ab}\colon \mathrm{LFOLLOWERS}(q, j) = \mathrm{F}(\alpha(s_j)))$,

existuje vtedy a len vtedy ak výrok

$(\forall\ q = \{s_i\}_{i=a}^{b},\ r = \{r_j\}_{i=c}^{d} \in \mathrm{LB}^{LS}\colon \forall\ k, \in \mathrm{INDEX}_{ab}, \forall\ m \in \mathrm{INDEX}_{cd}\colon s_k = r_m \Rightarrow \mathrm{LFOLLOWERS}(q, k) = \mathrm{LFOLLOWERS}(r, m))$

je platný (t.j. vtedy a len vtedy ak pre DSCS s logickým vnútorným opisom LS = (S, LB) pre všetky stavy množiny logických nasledovníkov sú nezávislé na histórii systému).

Tak, pri priamom modelovaní logického správania DSCS je použitý generátor, pričom stavy daného systému sú reprezentované stavmi generátora a možné logické priame stavové zmeny modelovaného systému sú reprezentované prvkami relácie priamych stavových zmien generátora.

Definícia 5.1.5.2: Nech LS = (S, LB) je logický vnútorný opis DSCS a nech G = (P, F), $\mathrm{P} = \mathrm{X} \times \mathrm{X}^*$ je generátor s pamäťou taký, že existuje injekcia α: S $\rightarrow$ X a platí výrok:

$(\forall q = \{s_i\}_{i=a}^{b} \in \mathrm{LB}^{LS}\colon \forall\ j \in \mathrm{INDEX}_{ab}\colon \mathrm{LFOLLOWERS}(q, j) = \mathrm{F}([\alpha(s_j), \{\alpha(s_i)\}_{i=a}^{j}]))$.

Potom množina trajektórií generátora G dané ako $MMODEL^{LS} = \{\{[\alpha(s_j), \{\alpha(s_i)\}_{i=a}^{j}]\}_{j=a}^{b} \mid \{s_j\}_{j=a}^{b} \in LB^{LS}\}$ je nazvaná *pamäťovým modelom logického správania DSCS*.

Evidentne, pamäťové modelovanie môže byť použité ako model logického správania ľubovoľného DSCS, ktorý má logický vnútorný opis. Teda, ak možnosť priamych zmien aktuálneho stavu závisí na stavovej histórii systému (t. j. podmienka priameho modelovania nie je splnená), potom je možné použiť ako model generátor s pamäťou, pričom stavová história je modelovaná stavom pamäti generátora.

Podobne ako v (Ramadge a Wonham, 1989), pri riadení DSCS sa predpokladá, že v prípade potreby určitým priamym stavovým zmenám môže byť zabránené. Prezentovaný prístup k problémom riadenia DSCS s logickým vnútorným popisom vychádza z myšlienky, že na určitej úrovni nie je nutné na úrovni vnútorného opisu systému modelovať supervízor, ale je postačujúce modelovať iba správanie riadeného systému. Ak riadený systém je opäť DSCS, ktorý má logický vnútorný opis (pozri definíciu 5.1.3.1), potom je možné modelovať jeho správanie relačným generátorom. Porovnaním generátorov modelujúcich neriadený a riadený systém je získaná informácia o riadiacich vstupoch (ktoré určujú pre každý aktuálny okamžik (logický krok), ktorým priamym stavovým zmenám musí byť zabránené, aby sa dosiahlo požadované správanie (riadiaca špecifikácia)).

Relácia medzi generátorom (P, F) modelujúcim nejaký systém diskrétnych stavových zmien D s logickým vnútorným opisom *LS* = (S, LB) a generátorom (P, C) modelujúcim riadený systém s logickým vnútorným opisom *LS* = (S', LB'), $S' \subseteq S$, $LB' \subseteq B$ je stručne formulovaná nasledovne: $\forall\, p \in D2(\alpha')$ (kde α' je injekcia $\alpha': S' \rightarrow P$) platí, že $F_u(p) \subseteq C(p) \subseteq F(p)$, kde prvky F_u reprezentujú neriaditeľné priame stavové zmeny systému D.

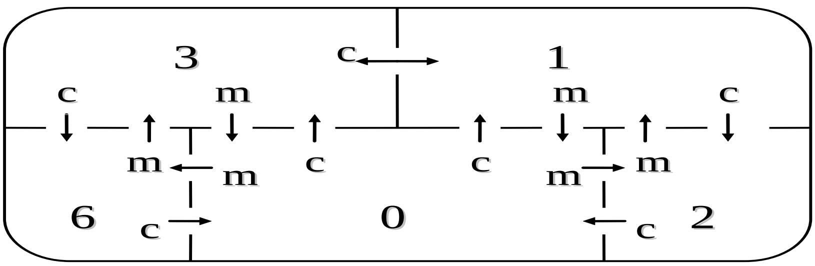
Definícia 5.1.5.3: Nech G = (P, F) je generátor. Nech relácia $F_u \subseteq F$. Nech L, $L \subseteq F$ je binárna relácia na P a nech $p \in L_{supp}$. Vravíme, že L je *riaditeľná v bode* p *vzhľadom k relácii* F_u vtedy a len vtedy ak platí: $(F_u(p,.) \subseteq L(p,.))$. Ak L je riaditeľná v každom bode svojho supportu vzdľadom F_u, potom L je nazvaná *riaditeľná vzhľadom k relácii* F_u.

Definícia 5.1.5.4: Nech G = (P, F) je generátor. Nech relácia $F_u \subseteq F$. Nech p, p' $\in$ P. Stav p' $\in$ P je *riaditeľne dosiahnuteľný vzhľadom k stavu* p $\in$ P *v generátore G* = (P, F) *s* F_u vtedy a len vtedy, keď existuje riaditeľná relácia C vzhľadom k F_u taká, že stavy p', p $\in C_{supp}$ a stav p' je živý vzhľadom k stavu p v generátore (P, C). Ak, navyše, platí že p' je koncový bod C_{supp} (t.j. $C(p') = \emptyset$) potom p' je *riaditeľne stabilizovateľný zo stavu* p *vzhľadom k* F_u *v generátore G*.

5.1.6 Ilustračný Príklad

V nasledujúcej podkapitole je modelovanie logického správania DSCS s vnútorným opisom ilustrované na príklade modelovania možného pohybu mačky a myši v bludisku popísanom v (Ramadge a Wonham, 1989). Tento príklad je často riešený (napr. v (Smedinga, 1993; Čapkovič, 1993; Kozák, 1993; Hrúz, 1994) atď.) čím je poskytnutá možnosť porovnať niektoré aspekty rôznych prístupov k DEDS modelovaniu a riadeniu.

Mačka a myš sú umiestnené v bludisku na obrázku 5.1.6.1. Nech γ je funkcia zobrazujúca miestnosti bludiska na množinu čísel A = {0,1,2,3,6} v súlade s obrázkom 5.1.6.1.



Obrázok 5.1.6.1. Bludisko s mačkou a myšou.

Nech $H = A \cup \{7\}$ a nech $P = A \times H$. Dvojica $[a, h] \in P$ reprezentuje stav, v ktorom:

- mačka je v miestnosti zobrazenej funkciou γ na číslo a;
- voľná myš je v miestnosti zobrazenej funkciou γ na číslo h (ak $h \in A$) alebo mačka chytila myš (ak h = 7).

Predpokladajme, že v počiatočnom stave je mačka umiestnená v miestnosti zobrazenej na číslo 2 a myš v miestnosti zobrazenej na číslo 6.

Nech F_u, F_c a F (F_u, F_c, $F \subseteq P \times P$) sú binárne relácie na P dané nasledovne:

$F_u = \{[[a, h], [a', h']]; (((a = 1 \wedge a' = 3) \vee (a = 3 \wedge a' = 1)) \wedge h' = h \wedge h \in H) \vee (h = a \wedge a' = a \wedge h' = 7 \wedge a \in A)\};$

$F_c = \{ [[a, h], [a', h']]; ([a', h'] = [(a + 1)_{\text{mod } 3}, h] \wedge a \in \{0,1,2\} \wedge h \in H) \vee ([a', h'] = [a, (h - 1)_{\text{mod } 3}] \wedge a \in A \wedge h \in \{0,1,2\}) \vee ([a', h'] = [(a + 3)_{\text{mod } 9}, h] \wedge a \in \{0,3,6\} \wedge h \in H) \vee ([a', h'] = [a, (h - 3)_{\text{mod } 9},] \wedge a \in A \wedge h \in \{0,3,6\})\};$

$F = F_u \cup F_c.$

Potom možný pohyb zvierat môže byť (pri predpoklade asynchrónneho správania sa zvierat) priamo modelovaný správaním generátora G = (P, F) s počiatočným stavom [2,6], pričom prvky relácie F_u reprezentujú priame stavové zmeny, ktorým nemôže byť zabránené (dvere medzi miestnosťami zobrazenými na 1 a 3 nemôžu byť zavreté). Požadovaná špecifikácia správania je vyjadrená nasledovnými troma požiadavkami.

1. Zvieratá počiatočne umiestnené v spomenutých miestnostiach (reprezentovaných dvojicou čísel [2,6]) nebudú nikdy súčasne v rovnakej miestnosti.
2. Vždy bude umožnený návrat mačky a myši do počiatočného stavu.
3. Bude umožnená najväčšia možná voľnosť pohybu zvierat.

Úlohou je určiť, či správanie vyhovujúce týmto požiadavkam môže byť dosiahnuté supervízorovým riadním a ak áno, potom určiť vlastnosti, ktoré musí daný supervízor mať. Po preložení tejto úlohy do termínov modelovacieho prostriedku získame úlohu nájsť binárnu reláciu C na množine P, pre ktorú platia nasledujúce výroky.

1. ${}^{\{[2,6]} C = C \subseteq F$
2. C je riaditeľná
3. $\forall$ [a, a] $\in$ P: [[2, 6], [a, a]] $\notin C^{\otimes}$
4. $\forall$ [a, h] $\in C^{\otimes}$([2, 6]): [[a, h],[2, 6]] $\in C^{\otimes}$
5. $\nexists$ K $\subseteq$ P x P pre ktorú sú platné štyri predchádzajúce výroky a zároveň platí: ${}^{\{[2,6]}C \subset {}^{\{[2,6]}K$.

Relácia C vyhovujúca predchádzajúcim piatim podmienkam je formálne daná nasledovne:

C ={[[a, h], [a’, h’]]; ([a’, h’] = [(a + 1)$_{\text{mod } 3}$, h] $\wedge$ a $\in$ {0,1,2} $\wedge$ h = 6) $\vee$

([a’, h’] = [a, (h - 3)$_{\text{mod } 9}$] $\wedge$ a = 2 $\wedge$ h $\in$ {0,3,6})$\vee$

(a = 0 $\wedge$ a’ = 3 $\wedge$ h’ = h = 6]) $\vee$

(((a = 1 $\wedge$ a’ = 3) $\vee$ (a = 3 $\wedge$ a’ = 1)) $\wedge$ h’ = h = 6)}.

Takto, porovnaním relácie F a relácie C v každom bode p $\in C_{supp}$ sa získa informácia o priamych stavových zmenách zo stavu reprezentovaného bodom p, ktoré musia byť umožnené supervízorom (sú to priame stavové zmeny reprezentované prvkami relácie C(p,.)) a ktorým musí byť supervízorom zabránené (priame stavové zmeny reprezentované prvkami relácie (F(p,.) - C(p,.))). Správanie takéhoto supervízorom riadeného systému môže byť priamo modelované správaním generátora *CG* = (P, C) s počiatočným stavom [2,6].

5.2 Algebraicky zovšeobecnené Petriho siete

5.2.1 Algebraicky zovšeobecnené Petriho siete

V tejto časti sú uvedené dva typy algebraicky zovšeobecnených Petriho sietí (AGPN alebo krátko GPN) podľa (Juhás, 1996) nadväzujúc na poznatky z kapitoly 2. Ako prvá je uvedená definícia jednoduchšieho typu, ktorý umožňuje definovať P aj T invarianty.

Poznámka: Všetky nasledujúce definície zovšeobecnených PN sú uvedené v maticovej forme.

Definícia 5.2.1.1: *Algebraicky zovšeobecnená Petriho sieť* je štruktúra GPN = (P, T, Γ, G', *I*, *O*, m_0), kde:

- P = $\{p_1, .. , p_m\}$ je konečná a neprázdna množina *pozícií*;
- T = $\{t_1, .. , t_n\}$ je konečná a neprázdna množina *prechodov*;
- $P \cap T = \emptyset$, $P \cup T \neq \emptyset$;
- $\Gamma = (G, \oplus)$ je ľubovoľná komutatívna grupa;
- $G' \subseteq G$;
- *I*: $P \times T \rightarrow G'$ je *vstupná funkcia*;
- *O*: $T \times P \rightarrow G'$ je *výstupná funkcia*;
- m_0: $P \rightarrow G'$ je *počiatočný stav*.

Podobne ako PN má funkciu určujúcu kapacitu pozícií, GPN môže mať tiež *K*-funkciu *K*: $P \rightarrow 2^{G'}$ ktorá pre každú pozíciu $p \in P$ určuje *množinu prípustných stavov*. Potom pre počiatočný stav musí platiť výrok $\forall$ p $\in$ P: $m_0 \in K$(p). Ak K-funkcia pre GPN nie je špecifikovaná, potom sa predpokladá, že je daná nasledovne: $\forall$ p $\in$ P: K(p) = G'.

Incidenčná funkcia C: P x T $\rightarrow$ G algebraicky zovšeobecnenej PN je daná (analogicky s PN) výrazom $\forall$ [p, t] $\in$ P x T: C(p, t) = O(t, p) $-_\oplus$ I(p, t) (incidenčná m x n rozmerná matica **C** je daná rovnosťou $\mathbf{C} = \mathbf{O}^t -_{\oplus^m} \mathbf{I}$). *Stav* GPN je vyjadrený funkciou m: P $\rightarrow$ G' (ak funkcia K je definovaná, potom stav GPN je funkcia spĺňajúca výrok $\forall$ p $\in$ P: m(p) $\in K$(p)). Zmena stavu GPN je spôsobená *otvorením prechodu*. Prechod t $\in$ T je *otvoriteľný v stave m* vtedy a len vtedy ak platí: $\forall$ p $\in$ P: $m(p) -_\oplus I(p, t) \in K(p) \wedge m(p) \oplus O(p,t) \in K(p) \wedge m(p) \oplus O(p,t) -_\oplus I(p, t) \in K(p)$. Otvorenie prechodu t $\in$ T v stave m spôsobí zmenu stavu m na stav m', pre ktorý platí: $\forall$ p $\in$ P: $m'(p) = m(p) \oplus O(p, t) -_\oplus I(p,t)$.

Realizácia realizovateľnej postupnosti prechodov t v stave m (definovanej analogicky ako pre PN) vedie do stavu m', pre ktorý platí

$$\mathbf{m}' = \mathbf{m} \oplus^m \mathbf{C} \bullet_{\oplus^m} \mathbf{b}_t. \qquad (5)$$

Nech pre multimnožinu Y nad T a stav m vektor $\mathbf{A} = \mathbf{m} -_\oplus^m \mathbf{I} \bullet_{\oplus^m} \mathbf{Y}$, $\mathbf{A}' = \mathbf{m} \oplus^m \mathbf{O}^t \bullet_{\oplus^m} \mathbf{Y}$ a $\mathbf{B} = \mathbf{m} \oplus^m \mathbf{C}^t \bullet_{\oplus^m} \mathbf{Y}$. Ak $\forall$ i $\in$ {1, .. , m}: $a_i \in K(p_i) \wedge a'_i \in K(p_i) \wedge b_i \in K(p_i)$, potom multimnožina Y a vektor **Y** sú nazývané *priamo realizovateľné* v stave m.

Takto, štandardné Petriho siete sú špeciálnym prípadom algebraicky zovšeobecnených Petriho sietí, kde: Γ = (Z, +), G' = $Z^+ \cup \{0\}$ a vzťah medzi funkciou kapacity K_{PN} Petriho Siete a K_{GPN}-funkciou príslušnej GPN je daný výrazom: $\forall$ p $\in$ P: $K_{GPN}(p) = \{0, .., K_{PN}(p)\}$.

Definícia 5.2.1.2: Multimnožina X nad P (vektor $\mathbf{X} = |\ x(p_1)\ .\ .\ x(p_m)\ |^t$) je nazvaná P-*invariant* GPN vtedy a len vtedy ak platí: $\mathbf{C}^t \bullet_{\oplus^n} \mathbf{X} = \boldsymbol{\varepsilon}$ ($\boldsymbol{\varepsilon} = |\ \varepsilon\ .\ .\ \varepsilon\ |^t$ je n rozmerný vektor, ε je neutrálny prvok v grupe Γ).

Definícia 5.2.1.3: Multi-množina Y nad T (vektor $\mathbf{Y} = |\ y(t_1)\ .\ .\ y(t_n)\ |^t$) je nazývaná T-*invariant* GPN vtedy a len vtedy ak platí: $\mathbf{C} \bullet_{\oplus^m} \mathbf{Y} = \boldsymbol{\varepsilon}$ ($\boldsymbol{\varepsilon} = |\ \varepsilon\ .\ .\ \varepsilon\ |^t$ je m rozmerný vektor, ε je neutrálny prvok v grupe Γ).

GPN môže byť ďalej špecifikovaná požiadavkami na množinu G' a *K*-funkciu.

Definícia 5.2.1.4: *Vyššie algebraicky zovšeobecnená Petriho sieť* je štruktúra HGPN = (P, T, *G*, *G'*, *I*, *O*, m_0), kde:

- $P = \{p_1, .., p_m\}$ je konečná a neprázdna množina *pozícií*;
- $T = \{t_1, .., t_n\}$ je konečná a neprázdna množina *prechodov*;
- $P \cap T = \varnothing$, $P \cup T \neq \varnothing$;
- $G: P \rightarrow CGROUPS$;
- $G': P \rightarrow \bigcup_{p \in P} 2^{G_p}$ je funkcia taká, že $\forall\ p \in P: G'(p) \subseteq G_p$;
- $I: P \times T \rightarrow \bigcup_{p \in P} G'(p)$ je *vstupná funkcia* taká, že $\forall\ p \in P: \forall\ t \in T: I(p,t) \in G'(p)$;
- $O: T \times P \rightarrow \bigcup_{p \in P} G'(p)$ je *výstupná funkcia*, taká, že $\forall\ t \in T: \forall\ p \in P: O(t,p) \in G'(p)$; $m_0: P \rightarrow \bigcup_{p \in P} G'(p)$ taká, že $\forall\ p \in P: m_0(p) \in G'(p)$ je *počiatočný stav*.

Pre HPGN môže byť tiež definovaná K-funkcia (K: P $\rightarrow \bigcup_{p \in P} 2^{G'(p)}$ taká, že $\forall$ p $\in$ P: K(p) $\subseteq$ G'(p)) ktorá pre každú pozíciu p $\in$ P určuje množinu prípustných stavov pozície. Potom pre počiatočný stav musí platiť výrok $\forall$ p $\in$ P: m_0(p) $\in$ K(p). Ak K-funkcia pre HGPN nie je špecifikovaná, potom sa predpokladá, že je daná nasledovne: $\forall$ p $\in$ P: K(p) = G'(p). Zmysel K-funkcie HGPN spočíva v skutočnosti, že umožňuje použiť ľubovoľný prvok G'(p) ako hodnotu vstupnej a/alebo výstupnej funkcie pre pozíciu p a zároveň tento prvok zakázať ako stav pozície p. Incidenčná funkcia C: P x T $\rightarrow$ G HGPN je daná výrazom $\forall$ [p, t] $\in$ P x T: C(p, t) = O(t, p) $-_{\oplus p}$ I(p, t) (incidenčná m x n rozmerná matica **C** je daná rovnosťou $\mathbf{C} = \mathbf{O}^{t} -_{\oplus \mathbf{F}_G} \mathbf{I}$). Stav HGPN je vyjadrený funkciou m: P $\rightarrow \bigcup_{p \in P} G'(p)$ takou, že $\forall$ p $\in$ P: m(p) $\in$ G'(p) (ak je definovaná funkcia K, potom stav HGPN je funkcia, pre ktorú platí $\forall$ p $\in$ P: m(p) $\in$ K(p)). Prechod t $\in$ T je otvoriteľný v stave m vtedy a len vtedy, keď platí: $\forall$ p $\in$ P: m(p) $-_{\oplus p}$ I(p, t) $\in$ K(p) $\wedge$ m(p) $\oplus_p$ O(p,t) $\in$ K(p) $\wedge$ m(p) $\oplus_p$ O(p,t) $-_{\oplus p}$ I(p, t) $\in$ K(p). Otvorenie prechodu t $\in$ T v stave m spôsobí zmenu stavu m na stav m', pre ktorý platí: $\forall$ p $\in$ P: m'(p) = m(p) $\oplus_p$ O(p, t) $-_{\oplus p}$ I(p,t). Realizácia realizovateľnej postupnosti prechodov t (definovanej analogicky ako pre PN) zo stavu m vedie do stavu m', pre ktorý platí

$$\mathbf{m}' = \mathbf{m} \oplus^{\mathbf{F}_G} \mathbf{C} \bullet_{\oplus \mathbf{F}_G} \mathbf{b}_t. \qquad (6)$$

Pre HPGN je možné definovať T-invarianty analogicky s PN, ale P-invarianty je možné definovať spôsobom analogickým s PN iba pre podmnožiny množiny P, pre ktoré je funkcia G konštantná. Inou možnosťou je definovať P-invarianty obdobným spôsobom ako sú definované pre farebné Petriho siete (pozri Jensen, 1986). Je zrejmé, že GPN sú špeciálnym prípadom HGPNs ktorej funkctie G a G' sú konštantné na P.

Z pohľadu výsledkov kapitoly 2 vstupná funkcia *I* a výstupná funkcia *O* HGPN (GPN) môžu byť prepísané na funkcie *I’* a *O’* nasledovným spôsobom:

I': $T \rightarrow [P \rightarrow \bigcup_{p \in P} G'(p)]$ ($T \rightarrow [P \rightarrow G']$) je taká, že $\forall$ t $\in$ T $\forall$ p $\in$ P:

$I'(t)(p) = I(p,t)$;

O': $T \rightarrow [P \rightarrow \bigcup_{p \in P} G'(p)]$ ($T \rightarrow [P \rightarrow G']$) je taká, že $\forall$ t $\in$ T $\forall$ p $\in$ P:

$O'(t)(p) = O(t,p)$.

Potom tiež incidenčná funkcia *C* môže byť prepísaná na formu: C': $T \rightarrow F_G$ ($T \rightarrow [P \rightarrow G]$) je taká, že $\forall$ t $\in$ T: $C'(t) = O'(t) -_{\oplus F_G} I'(t)$ ($C'(t) = O'(t) -_{\oplus [P \rightarrow G]} I'(t)$);

a rovnice (5) a (6) môžu byť nahradené rovnicami:

$$m' = m \oplus^{[A \rightarrow G]} \hat{C}'(b_t) \tag{7}$$

$$m' = m \oplus^{F_G} \hat{C}'(b_t) \tag{8}$$

respektíve, kde $\hat{C}'$ je multimnožinové rozšírenie prepísanej incidenčnej funkcie *C’*.

GPN umožňujú teda použiť ako stavy pozícií nie iba celé čísla a, hlavne, v dynamike Petriho siete nie iba operáciu sčítania celých čísel, ale prvky ľubovoľnej komutatívnej grupy a jej operáciu. Vyššie GPN, navyše, umožňujú použiť rôzne operácie pre jendotlivé pozície (alebo podmnožiny množiny P). Ako motivačný príklad sú uvedené dva typy GPN.

Definícia 5.2.1.5: GPN, kde $\Gamma = (Z_k = \{0, .., k - 1\}, +_{\text{mod } k})$ ($k \in Z^+$, $+_{\text{mod } k}$ je k-modulo sčítanie (k-modulo sčítanie dvoch celých čísel a,b je zvyšok podielu (a + b)/k) pozri napr. (MacLane a Birkhoff, 1968)), je nazývaná *k-modulo Petriho sieť*.

k-Modulo PN je možné využiť napr. v aplikáciách, v ktorých zložky systému majú k-stavov menených cyklicky, t.j. napr. systémy, ktorých zložkami sú posuvné registre s dĺžkou k.

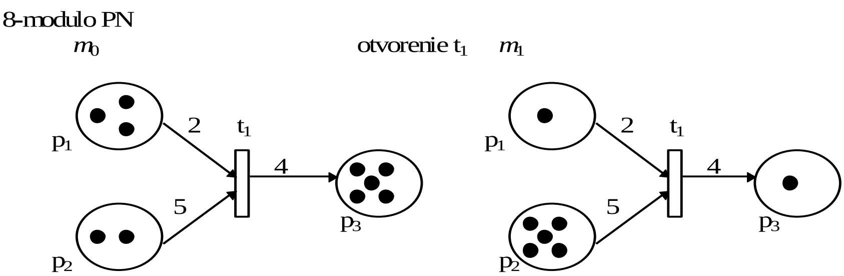
Príklad 5.2.1.1: Na obrázku 5.2.1.1 je znázornená 8-modulo Petriho sieť s troma pozíciami a jedným prechodom:

$\mathbf{I} = | 2\ 5\ 0 |^t$; $\mathbf{O} = | 0\ 0\ 4 |$; $\mathbf{m}_0 = | 3\ 2\ 5 |^t$;

pred a po otvorení prechodu t_1.

Formálne:

$$\mathbf{m}_1 = \mathbf{m}_0 +_{\text{mod } 8^3} \mathbf{C} \bullet_{+_{\text{mod } 8^3}} \mathbf{Y} = |3\ 2\ 5|^t +_{\text{mod } 8^3} |6\ 3\ 4|^t \bullet_{+_{\text{mod } 8^3}} 1 = | 1\ 5\ 1 |^t.$$



Obrázok 5.2.1.1 Grafické znázornenie 8-modulo Petriho siete.

Definícia 5.2.1.6: GPN, kde $\Gamma = (Q, .)$ (Q označuje množinu všetkých racionálnych čísel, . označuje štandardné násobenie) a $G' = Z^+ \cup \{0\}$ je nazývaná *multiplikatívna Petriho sieť*.

Poznámka: V súlade s predchádzajúcimi definíciami štandardná Petriho sieť može byť tiež nazývaná *aditívna Petriho Sieť* (alebo navyše *celočíselná aditívna Petriho sieť*), k-modulo Petriho sieť môže byť nazývaná *k-modulo aditívna Petriho sieť*, a navyše, multiplikatívna Petriho sieť môže byť nazývaná *celočíselná multiplikatívna Petriho sieť*. Všeobecne, meno typu Petriho siete môže byť vytvorené pridaním mena množiny a operácie.

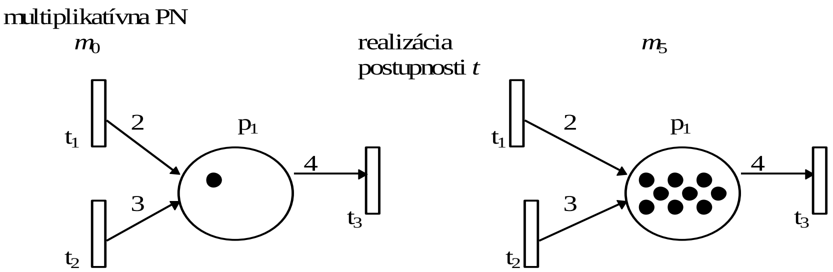
Príklad 5.2.1.2: Na obrázku 5.2.1.2 je znázornená multiplikatívna Petriho sieť s jednou pozíciou a troma prechodmi:

$\mathbf{I} = |\ 1\ 1\ 4\ |$; $\mathbf{O} = |\ 2\ 3\ 1\ |^t$; $\mathbf{m}_0 = |\ 1\ |$;

pred a po realizácii postupnosti prechodov $t = \{t_2, t_2, t_1, t_1, t_3\}$.

Formálne:

$$\mathbf{m}_5 = \mathbf{m}_0 \,.\, \mathbf{C} \bullet_{.} \mathbf{Y} = |\ 1\ |\ .\ |\ 2\ 3\ 1/4\ | \bullet_{.} \begin{vmatrix} 2 \\ 2 \\ 1 \end{vmatrix} = 1\ .\ 2^2\ .\ 3^2\ .\ (1/4) = |\ 9\ |.$$



Obrázok 5.2.1.2 Grafické znázornenie multiplikatívnej Petriho siete.

5.2.2 Algebraicky zovšeobecnené farebné Petriho siete

Definícia 5.2.2.1: *Algebraicky zovšeobecnená farebná Petriho sieť* je štruktúra GCPN = (P, T, Φ, G, G', I, O, m_0), kde:

- P = $\{p_1, .. , p_m\}$ je konečná a neprázdna množina *pozícií*;
- T = $\{t_1, .. , t_n\}$ je konečná a neprázdna množina *prechodov*;
- $P \cap T = \emptyset$, $P \cup T \neq \emptyset$;
- Φ je funkcia definovaná na $P \cup T$, ktorá priradí každej pozícii ľubovoľnú množinu farieb stavových prvkov pozície a každému prechodu ľubovoľnú množinu farieb otvorenia prechodu;
- G je funkcia definovaná na P taká, že $\forall$ p $\in$ P: $G(p) \in [\Phi(p) \rightarrow$ *CGROUPS*];

V súlade s podkapitolou 2.1 venovanou algebraickému pozadiu nech $\forall$ p $\in$ P: $\forall$ c $\in$ Φ(p): $G(p)(c)$ je označená $G(p)(c) = \Gamma_{p,c} = (G_{p,c}, \oplus_{p,c})$;

- G' je funkcia definovaná na P taká, že $\forall$ p $\in$ P: $G'(p) \in [\Phi(p) \rightarrow \bigcup_{c \in (p)} 2^{G_{p,c}}] \wedge \forall$ c $\in \Phi(p)$: $G'(p)(c) \subseteq G_{p,c}$. Funkcia G' určuje pre každú pozíciu p $\in$ P a každú farbu jej stavových prvkov c $\in \Phi(p)$ množinu prípustných hodnôt $G'(p)(c)$;
- I je *vstupná funkcia* definovaná na P x T taká, že $\forall$ [p, t] $\in$ P x T: ($I(p,t) \in [\Phi(t)_{MS} \rightarrow F_{G(p)}]_L \wedge$ funkcia i definovaná na T, pre ktorú platí $\hat{i} = I$ je taká, že $\forall$ x $\in \Phi(t)$: $\forall$ c $\in \Phi(p)$: $i(p, t)(x)(c) \in G'(p)(c)$);
- O je *výstupná funkcia* definovaná na T x P taká, že $\forall$ [t, p] $\in$ T x P: ($O(t,p) \in [\Phi(t)_{MS} \rightarrow F_{G(p)}]_L \wedge$ funkcia o definovaná na T, pre ktorú platí $\hat{o} = O$ je taká, že $\forall$ x $\in \Phi(t)$: $\forall$ c $\in \Phi(p)$: $o(p, t)(x)(c) \in G'(p)(c)$);
- m_0 je funkcia nazývaná *počiatočný stav* definovaná na P taká, že $\forall$ p $\in$ P: $\forall$ c $\in \Phi(p)$: $m_0(p)(c) \in G'(p)(c)$.

Stav GCPN, označený m, je vyjadrený funkciou s rovnakým definičným oborom a oborom hodnôt ako počiatočný stav m_0. Zmena stavu GCPN je spôsobená otváraním prechodov pre jednotlivé farby otvorenia. Prechod $t \in T$ je *otvoriteľný* pre farbu otvorenia $x \in \Phi(t)$ v stave m vtedy a len vtedy, ak platí: $\forall\, p \in P$: $\forall\, c \in \Phi(p)$:

$m(p)(c) -_{\oplus p,c} i(p, t)(x)(c) \in G'(p)(c) \wedge m(p)(c) \oplus_{p,c} o(p,t)(x)(c) \in G'(p)(c) \wedge m(p)(c) \oplus_{p,c} o(p,t)(x)(c) -_{\oplus p,c} i(p, t)(x)(c) \in G'(p)(c)$.

Otvorenie prechodu $t \in T$ pre $x \in \Phi(t)$ v stave m spôsobí zmenu stavu m na stav m', pre ktorý platí:

$\forall\, p \in P$: $m'(p) = m(p) \oplus^{F_{G(p)}} o(p, t)(x) -_{\oplus^{F_{G(p)}}} i(p,t)(x)$.

Ak pre funkciu Y definovanú na T takú, že $\forall\, t \in T$: $Y(t) \in \Phi(t)_{MS}$ platí: $\forall\, p \in P$: $\forall\, c \in \Phi(p)$: $m(p)(c) -_{\oplus p,c} \sum_{t \in T}^{\oplus p,c} I(p, t)(Y(t))(c) \in G'(p)(c) \wedge$ $m(p)(c) \oplus_{p,c} \sum_{t \in T}^{\oplus p,c} O(p, t)(Y(t))(c) \in G'(p)(c) \wedge m(p)(c) -_{\oplus p,c} \sum_{t \in T}^{\oplus p,c} I(p, t)(Y(t))(c) \oplus_{p,c} \sum_{t \in T}^{\oplus p,c} O(p, t)(Y(t))(c) \in G'(p)(c)$,

potom funkcia Y je nazývaná *priamo realizovateľná* v stave m. Realizácia priamo realizovateľnej funkcie Y zo stavu m vedie do stavu m', pre ktorý platí: $\forall\, p \in P$:

$$m'(p) = m(p) -_{\oplus^{F_{G(p)}}} \sum_{t \in T}^{\oplus^{F_{G(p)}}} I(p, t)(Y(t)) \oplus^{F_{G(p)}} \sum_{t \in T}^{\oplus^{F_{G(p)}}} O(p, t)(Y(t)).$$

Špeciálne, ak pre $p \in P$ funkcia $G(p)$ je konštantná s oborom hodnôt tvoreným aditívnou grupou (Z, +), potom získavame štandardnú (aditívnu) farebnú Petriho sieť definovanú v kapitole 2 podľa (Jensen, 1986). Na druhej strane, ak funkcia Φ je konštantná, t.j. jej oborom hodnôt je jednoprvková množina, potom získavame definíciu vyššie algebraicky zovšeobecnenej Petriho siete definovanej v predchádzajúcej podkapitole podľa (Juhás, 1996b).

Príklad 5.2.2.1: Nech GCPN je daná nasledovne:

$P = \{p_1\}$; $T = \{t_1, t_2\}$;

$\Phi(t_1) = \Phi(t_2) = \{*\}$, $\Phi(p_1) = \{\spadesuit, \blacklozenge, \clubsuit\}$;

$G(p_1)(\spadesuit) = (Z,+)$, $G(p_1)(\blacklozenge) = (Q, .)$ a $G(p_1)(\clubsuit) = (Z_4=\{0, .., 3\}, +_{\text{mod } 3})$;

$G'(p_1)(\spadesuit) = (Z^+ \cup \{0\})$, $G'(p_1)(\blacklozenge) = (Z^+ \cup \{0\})$, $G'(p_1)(\clubsuit) = Z_4$;

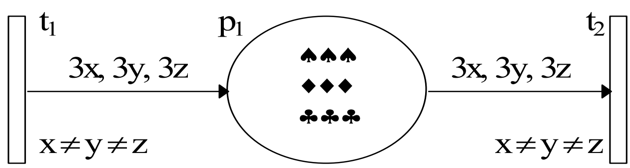
Nech *trans* $\in [\{*\}_{MS} \rightarrow F_{G(p_1)}]_L$ je taká, že $\forall$ b $\in \{*\}_{MS}$:

$trans(b)(\spadesuit) = 3 \,.\, b(*)$, $trans(b)(\blacklozenge) = 3^{b(*)}$, $trans(b)(\clubsuit) = 3 \,._{\text{mod } 3}\, b(*)$ (pre k $\in Z^+$ $._{\text{mod } k}$ je k-modulo násobenie (k-modulo násobenie dvoch celých čísel a,b je zvyšok podielu (a.b)/k);

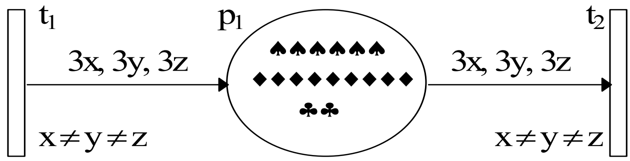
$I(p_1, t_2) = trans$, $O(t_1, p_1) = trans$, $I(p_1, t_1)$ a $O(t_2, p_1)$ sú nulové funkcie;

$m_0(p_1)(\spadesuit) = m_0(p_1)(\blacklozenge) = m_0(p_1)(\clubsuit) = 3$.

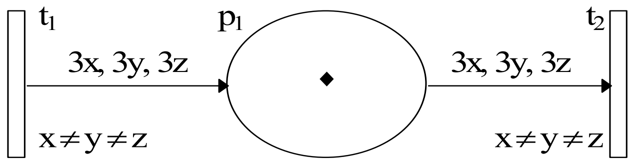
Potom, t_1 aj t_2 sú otvoriteľné v m_0 pre farbu otvorenia *. Otvorenie t_1 v m_0 spôsobí zmenu na stav m', $m'(p_1)(\spadesuit) = 6$, $m'(p_1)(\blacklozenge) = 9$, $m'(p_1)(\clubsuit) = 2$, v ktorom opäť t_1 aj t_2 sú otvoriteľné pre *. Na druhej strane, otvorenie t_2 v m_0 spôsobí zmenu na stav m'' taký, že $m''(p_1)(\spadesuit) = 0$, $m''(p_1)(\blacklozenge) = 1$, $m''(p_1)(\clubsuit) = 0$, v ktorom je otvoriteľný iba prechod t_1 pre farbu otvorenia *.



Obrázok 5.2.2.1a. Grafová forma izomorfná s GCPN z príkladu 5.2.2.1 v počiatočnom stave m_0.



Obrázok 5.2.2.1b. Grafová forma izomorfná s GCPN z príkladu 5.2.2.1 v stave *m*' dosiahnutom otvorením prechodu t_1 pre fabu otvorenia * z požiatočného stavu m_0.



Obrázok 5.2.2.1c. Grafová forma izomorfná s GCPN z príkladu 5.2.2.1 v stave *m*'' dosiahnutom otvorením prechodu t_2 pre fabu otvorenia * z požiatočného stavu m_0.

Poznámka: Grafová forma GCPN môže byť definovaná analogicky s grafovou formou štandardnej CPN danej v kapitole 2 podľa (Jensen, 1986). Poznamenajme, že v grafovej forme pozície, prechody, zúženie funkcie Φ na P a funkcie *G* a *G*' sú definované rovnako ako pre maticovú formu. Ďalej, ku každej hrane je priradený výraz obsahujúci množinu premenných, ktoré určujú funkcie *i* respektíve *o,* a ku každému prechodu je priradený predikát nazvaný guard. Pomocou týchto pojmov je možné interpretovať dynamiku GCPN nasledovným spôsobom. Prechod $t \in T$ je otvoriteľný pre také hodnoty premenných, pre ktoré jeho predikát je pravdivý a po ktorých dosadení do výrazov okolitých hrán, hodnoty stavov pre jednotlivé farby získané interpretáciou týchto výrazov patria do príslušných množín prípustných stavov. V prezentovanom príklade, premenné x, y a z môžu nadobúdať hodnoty z množiny {♠,♦, ♣}.

Záver

Skriptá je možné rozdeliť na dve hlavné časti. V prvej časti, venovanej prehľadu základných prostriedkov, je prezentovaných niekoľko modelovacích formalizmov logických systémov diskrétych udalostí počnúc formálnymi jazykmi a automatmi, cez Petriho siete, farebné Petriho siete a zjednodušenú formu Statechart formalizmu, tzv. hierarchické stavové stroje, po predikátový model DEDS.

Dôraz bol kladený na formálny matematický popis týchto nástrojov a ich formálny vzťah k automatom, pričom jednotlivé prvky popisu sú ilustrované na jednoduchých príkladoch.

Výber prezentovaných formalizmov vychádzal z ich vzťahu k automatom (každý z prezentovaných prostriedkov možno v istom zmysle chápať ako špeciálny typ automatu) a hľadiska rozšírenia ich použitia, pričom ohľad bol braný i na ich použitie v domácich prácach, kde sa využívajú najmä Petriho siete a farebné Petriho siete (pozri kapitolu 2), ale i formalizmus Statechart (napr. (Fogel, Kocian a Sebestyénová, 1994; Fogel, 1995; Sebestyénová, 1995). Vyjadrenie DEDS pomocou predikátov je v zahraničnej literatúre značne rozšírené a rozpracované. Existujú, samozrejme, mnohé ďalšie logické modelovacie prostriedky DEDS ako sú teória komunikujúcich sekvenčných procesov (Cieslak a Varaiya, 1990; Inan, 1993), modely využívajúce temporálnu logiku (Ostroff, 1990), prípadne synchrónne jazyky (Esterel a pod.) (Berry a kol., 1988).

Na tomto mieste spomeňme, že každý z modelovacích prostriedkov DEDS poskytuje určitú jedinečnú množinu výhod. To znamená, že nie je možné všeobecne určiť najefektívnejší prostriedok modelovania DEDS, vhodnosť závisí od aplikačnej oblasti, či typu modelovaného systému.

Vezmúc si na pomoc terminológiu z teórie jazykov, je možné povedať, že práca sa zaoberá syntaxou vybraných prostriedkov a ich vzťahu k automatom, pričom sémantika, t.j. význam jednotlivých čŕt pri modelovaní DEDS všeobecne i pri modelovaní špecializovaných tried

DEDS je spomenutý iba v miere nevyhnutnej na objasnenie ich základného fungovania.

Tento prístup bol zvolený okrem iného preto, že zatiaľ nebola u nás vydaná žiadna publikácia zaoberajúca sa formálnym popisom logických modelovacích prostriedkov systémov diskrétnych udalostí.

V prípade záujmu o sémantiku jednotlivých formalizmov je vhodné obrátiť sa na literatúru, pričom základné pramene sú uvedené i v záverečnom prehľade. Z domácich prác sa v pomerne veľkom rozsahu a logicky ucelene sémantikou jednotlivých prostriedkov a porovnaním ich modelovacej efektivity zaoberá práca (Kocian, 1992).

Druhá časť skrípt (kapitola 5) prezentuje vybrané príklady rozšírenia základných modelovacích formalizmov:

- Definíciu DEDS na úrovni vnútorného opisu ako časť viacúrovňového opisu systému;
- Podmienku logického vnútorného opisu DEDS;
- Základy relačného počtu pre DEDS modelovanie umožňujúceho vyjadriť DEDS množinou predikátov určujúcich reláciu stavových zmien;
- Na algebraických štruktúrach založené zovšeobecnenie Petriho sietí, umožňujúce využívať prvky a operáciu ľubovoľnej komutatívnej grupy zachovávajúc analytické vyjadrenie dynamiky siete.

Dôraz bol kladený na vysvetlenie vzájomných súvislostí medzi matematicky odlišnými spôsobmi popisu udalostných systémov prostredníctvom automatov ako základného modelovacieho formalizmu. Hlavným cieľom skrípt, ktorý ovplyvnil výber formalizmov, je na pomerne kompaktnom rozsahu neprevyšujúcom 100 strán poskytnúť základné nástroje a zároveň ilustrovať rôznorodé prístupy k formalizmom pre modelovanie udalostných systémov a ilustrovať ich možné rozšírenia tak, aby čitateľ získal solídny fundament pre ďalšiu prácu s aktuálnou literatúrou.

Zoznam použitej literatúry

1. Ajzerman, M. A., (1963), *Logika, automaty a algoritmy*. Praha, Academia, 1971; Ruský originál, *Logika, avtomaty, algoritmy.* Moskva, Fizmatgiz, 1963.
2. Balemi, S., Kozák, P. and Smedinga, R. (Ed.), (1993), *Discrete Event Systems: Modelling and Control,. Proc. of a Joint Workshop held in Prague, 1992,* Birkhäuser, 1993.
3. Berry, G., Couronne, P., Gonthier, G., (1988), Synchronous Programming of Reactive Systems: An Introduction to Esterel. In: Fuchi, K. a Nivat, M. (Ed.), *Programming of Future Generation Computers,* Elsevier Science Publisher, 1988, pp. 35-55.
4. Bertalanffy, L. von, (1928), *Modern Theories of Development*. Oxford, Oxford University Press, 1934; German original, *Kritische Theorie der Formbildung.* Berlin, Borntraeger, 1928.
5. Bertalanffy, L. von, (1950), An outline of General System Theory. *British Journal of the Philoshophy of Science*, Vol. 1 (1950), pp. 134-164.
6. Bertalanffy, L. von, (1968), *General System Theory. Foundations, Development, Applications.* New York, George Braziller, 1968.
7. Bertalanffy, L. von (1972) The History and Status of General Systems Theory. In: Klir, G. J (Ed.) *Trends in General Systems Theory*. New York, Wiley, 1972.
8. Birkhoff, G. a Bartee, T. C., (1970), *Aplikovaná algebra*, Bratislava, Alfa, 1981; Anglický originál, *Modern aplied algebra*, New York, McGraw-Hill, 1970.
9. Birkhoff, G. a MacLane, S., (1967), *Algebra*. Bratislava, Alfa, 1974; Anglický originál, *Algebra*. New York, The Macmillan Company, 1968.
10. Brave, Y. a Heymann, M., (1993) Control of Discrete Event Systems Modeled as Hierarchical State Machines. *IEEE tr. on Automatic Control*, Vol. 38 (1993), N. 12, 1803-1819.

11. Čapkovič, F., (1993), Petri Net-based Approach to the Maze Problem Solving. In: Balemi, S., Kozák, P. and Smedinga, R. (Ed.) *Discrete Event Systems: Modelling and Control*, Basel, Birkhäuser, 1993, pp. 173-179.

12. Čapkovič, F., (1994), Computer-Aided Design of Intelligent Control Systems for Discrete Event Dynamic Systems. In: *Proc. of IEEE-IFAC Joint Symposium on Computer-Aided Control System Design*, Tuscon, Arizona, NJ, IEEE Press, 1994, pp. 55-60.

13. Čapkovič, F., (1995) Using fuzzy logic for knowledge representation of control synthesis. *BUSEFAL*, Vol. 63, 1995, pp. 4-9.

14. Cieslak, R. A. a Varaiya, P. P., (1990), Undecidability Results for Deterministic Communicating Sequential Processes, IEEE tr. on AC, Vol. 35 (1990), N. 9, pp. 1032-1039.

15. Dijkstra, E. W., (1976), *A Discipline of Programming*. Englewood Cliffs, NJ, Prentice-Hall, 1976.

16. Dijkstra, E. W. a Scholten, C. S., (1990), *Predicate Calculus and Program Semantics*. New York, Springer Verlag, 1990.

17. Gvozdiak, L., Boršč, M a Vitko, A., (1989), *Základy kybernetiky*. Bratislava, Alfa, 1990.

18. Fogel, J., (1995), A Statecharts Formalism for Modelling Concurrent Object Oriented Systems, In: *Proc. of Workshop on Object-Oriented Programming and Models of Concurrency (within Conference on Application and Theory of Petri Nets*, pp. 211-217.

19. Fogel, J., Kocian, M. a Sebestyénová, J., (1994), An Approach to Object-Oriented Modelling and Control of a DEDS. In: *Preprints of 1st IFAC Workshop on New Trends in Design of Control* Systems 1994, Smolenice, Slovakia, 380-385.

20. Frankovič, B., (1995), The Survey about the Role of Modelling, Simulation, Reasoning and Decision in the Design of FMS Control. In: *Proc. of ISMCR'95*, Smolenice, Slovakia, pp. 51-59.

21. Harel, D., (1987), Statecharts: A visual Formalism for Complex Systems, *Science of Computer Programming,* 8 (1987), pp. 241-274.

22. Harel, D. a kol., (1987), On the Formal Semantics of Statecharts, In: *Proc. of 2nd IEEE Symposium on Logic in Computer Science*, 1987, pp. 54-64.

23. Hladký, M., (1975), *Základy teoretické kybernetiky II., Matematická teorie jazyků*. Skriptum VUT Brno, Praha, SNTL, 1975.

24. Ho, Y. Ch., (Ed.) (1992), *Discrete Event Dynamic Systems.* New York, IEEE Press, 1992.

25. Hopcroft, J.E. a Ullman, J.D., (1978), *Formální jazyky a automaty*. Bratislava, Alfa, 1978.

26. Hrúz, B., (1994), The Supervisory Control Problem Solved Via Petri Nets. In: *Preprints of 1st IFAC Workshop on New Trends in Design of Control* Systems 1994, Smolenice, Slovakia, 386-391.

27. Chen, S. M., Ke, J. S. a Chang, J. F., (1990), Knowledge Representation Using Fuzzy Petri Nets, *IEEE tr.on AC,* Vol. 2 (1990), no. 3, pp. 311-319.

28. Chomsky, N., (1966), *Syntaktické struktury.* Praha, Academia, 1966.

29. Chytil, M., (1984), *Automaty a gramatiky*. Matematický seminár 19, Praha, SNTL, 1984.

30. Inan, K. (1993) Supervisory Control and Formal Methods for Distributed Systems. In: Balemi, S., Kozák, P. and Smedinga, R. (Ed.) *Discrete Event Systems: Modelling and Control,. Proc. of a Joint Workshop held in Prague, 1992,* Basel, Birkhäuser, 1993.

31. Jensen, K., (1986), Coloured Petri Nets. In: *Proceedings of an Advanced Course on Petri Nets,* LNCS 254, (1986), pp.248-299.

32. Juhás, G., (1993), *Štrukturálna analýza Petriho sietí*. Diplomová práca, MFF UK Bratislava, 1993.

33. Juhás, G., (1996a), An Internal Approach to Systems Description and its Specification for DEDS Relation Modelling, In: Trappl, R (ed.): *Cybernetics and Systems'96, Proc. of European Meeting on Cybernetics and Systems Research*, Vienna, 1996, pp. 8-13.

34. Juhás, G., (1996b), An Algebraic Generalisation of Petri Nets, In: *Proc. of 30th International Conference Modelling and Simulation of Systems MOSIS'96*, Krnov, Czech Republic, 1996, pp. 69-74.

35. Juhás, G., (1996c), On Algebraic Structures Based Extension of Coloured Petri Nets. In: *Řízení procesů ŘÍP'96*, Horní Bečva, Česká Republika, 1996.

36. Juhás, G. and Kocian, M., (1994), Generalized T-Invariants of Petri Nets and Control of DEDS. In: *Preprints of 1st IFAC Workshop on New Trends in Design of Control* Systems 1994, Smolenice, Slovakia, 408-413.

37. Juhás, G. a Kocian, M., (1995a), Relation Modelling and Control of Discrete State Change Systems. In: Bubnicki, Z (Ed.): *Proceedings of the International Conference on System Science*, 1995, Wroclaw, Poland, Volume 1, pp. 426-433.

38. Juhás, G. a Kocian, M., (1995b), Invariant Relation Behaviour in Petri Nets. In: Snorek, Sujansky, and Verbraeck (Ed.) *Modelling and Simulation 1995*, ESM95, Prague, A publication of the[SCS], pp. 160-164.

39. Juhás, G. a Kocian, M., (1995c), State Difference Relation in Petri Nets. In: Jávor, A. (Ed.) *Simulation Tools and Applications, Proc. of IMACS European Simulation Meeting*, Győr, Hungary, 1995, pp. 241-247.

40. Juhás, G. a Kocian, M., (1996), Invariant State Progress and Relation Modelling of DEDS. In Doležal, J. and Fidler, J. (Ed.): *Systems Modelling and Optimization*, Proceedings of the IFIP Conference on System Modelling and Optimization 1995, Chapman & Hall, London 1996, pp. 146-154.

41. Klir, G. J., (1969), *An approach to general systems theory*. New York, Van Nostrand Reinhold, 1969.

42. Kocian, M., (1992), *Prostriedky špecifikácie dynamických systémov dyskrétnych udalostí*. Písomná práca k odbornej kandidátskej skúške, Bratislava, ÚTRR SAV, 1992.

43. Kotek, Z., Vysoký, P. a Zdráhal. Z., (1990), *Kybernetika*. Praha, SNTL, 1990.

44. Kozák, P., (1992), Discrete events and general systems theory, *Int. J. of Systems Sci*., Vol. 23 (1992), N. 9, 1403-1422.

45. Kozák, P., (1993), The Cat-and-Mouse Problem with Least Delays. In: Balemi, S., Kozák, P. and Smedinga, R. (Ed.) *Discrete Event Systems: Modelling and Control,. Proc. of a Joint Workshop held in Prague, 1992,* Basel, Birkhäuser, 1993.

46. Kumar, R., V. Garg and Marcus, S. I., (1993), Predicate and predicate transformers for supervisory control of discrete event dynamic systems. *IEEE tr. on AC*, Vol.38 (1993), N. 2, pp. 232-247.

47. Kruckeberg, F. a Jaxy, M., (1986), *Mathematical methods for calculating invariants in Petri nets*, Institute fur Methodische Grundlangen GMD, Germany.

48. Labátová, S. a Frankovič, B., (1995), Decision on the Basis of Reliability Development in the Design of Flexible Manufacturing Systems, In: *Proc. of ISMCR'95*, Smolenice, Slovakia, pp. 513-516.

49. Lafortune, S., (1988), Modelling and Analysis of Transaction Execution in Database Systems. *IEEE tr. on AC*, Vol. 33 (1988), N. 5, pp. 439-447.

50. Lautenbach, K., (1986), Linear algebraic techniques for place/transitions nets. In: *Proceedings of an Advanced Course on Petri Nets*, LNCS 254, pp.142-167.

51. Maimom, O. a Tadmore, G. (1986), Efficient low level control of FMS. Draft Tech. Report LIDS-P-1571, Laboratory for Information and Decision Systems, MIT, Cambridge, MA, 1986.

52. Martinez, J. a Silva, M., (1982), A simple and fast algorithm to obtain all invariants of a generalized Petri nets. *Informatik-Fachberickte 52, Application and Theory of Petri Nets*. New York, Springer-Verlag, 1982, pp. 301-310.

53. Memmi, G. a Roucariol, G., (1980), Linear algebra in the net theory. In: *LNCS*, Vol. 84, [15], 1980, pp. 213-223.

54. Mesarovic, M. D. (1972) A Mathematical Theory of General Systems. In: Klir, G. J (Ed.) *Trends in General Systems Theory*. New York, Wiley, 1972.

55. Molnár, Ľ., Češka, M. a Melichar, B., (1987), *Gramatiky a jazyky*. Bratislava, Alfa, 1987.

56. Murata, T., (1989) Petri nets: Properties, Analysis and Applications. *Proceedings of the IEEE*, Vol. 77 (1989), N. 4, 541-588.

57. Orchard, R. A., (1972), On an Approach to General Systems Theory, Klir, G. J (Ed.) *Trends in General Systems Theory*. New York, Wiley, 1972.

58. Ostroff, J., (1990), A Logic for Real-Time Discrete Event Processes, *IEEE Control Systems Magazine,* Vol. 10 (1990), N. 4, pp. 386-397.

59. Peterson, J. L., (1981), Petri Net Theory and the Modeling of Systems. Englewood Cliffs, NJ, Prentice-Hall, 1981.

60. Petri, C. A., (1962), *Kommunikation mit Automaten*. Schriften des Institutes für Instrumentelle Mathematik, Bonn, 1962.

61. Preparata, F. P., a Yeh, R.T., (1974), *Úvod do teórie diskrétnych matematických štruktúr*. Bratislava, Alfa, 1981; Anglický originál, *Introduction to Discrete Structures*, Addison-Wesley, 1974.

62. Ramadge, P. j. .G. and Wonham, W. M., (1989), The Control of Discrete Event Dynamic Systems. *Proceedings of the IEEE,* Vol. 77 (1989), N. 1, pp. 81-98.

63. Reisig, W., (1985), *Petri Nets*. Berlin, Springer-Verlag, 1985.

64. Sebestyénová, J., (1995), Object-Oriented Framework for Modelling and Control of DEDS-some Concurrency Aspects. In: *Proc. of AMSE Conference SYSTEMS Analysis, Control & Design*, Brno, Czech Republic, 1995, Vol. 2, pp. 128-136.

65. Silva, M., (1987), Toward a synchrony theory for P-T nets. In: *Concurrency and Nets* [25], 1987, pp. 435-460.

66. Silva, M. a Colom, M., (1988), On the computation of structural synchronic invariants in P-T nets In: *Advances in Petri Nets 1988, LNCS,* New York, Springer-Verlag, 1989.

67. Smedinga, R., (1993), The Workshop Exercise Using a Trace Theory Based Setting. In: Balemi, S., Kozák, P. and Smedinga, R. (Ed.) *Discrete Event Systems: Modelling and Control,. Proc. of a Joint Workshop held in Prague, 1992,* Basel, Birkhäuser, 1993.

68. Ushio, T., (1990), Maximally Permissive Feedback and Modular Control Synthesis in Petri Nets with External Input Places, *IEEE tr. on AC*, Vol. 35 (1990), N. 7, pp. 844-848.

69. Waymore, A. W., (1972), A Wattled Theory of Systems. In: Klir, G. J (Ed.) *Trends in General Systems Theory*. New York, Wiley, 1972.

70. Willson, R. G., a Krogh, B. H., (1990), Petri Net Tools for the Specification and Analysis of Discrete Controllers. *IEEE tr. on AC*, Vol. 16 (1990), N. 1, pp. 39-50.

71. Zadeh, L., (1964) The concept of state in system theory. In Mesarovic, M. (Ed.) *Views on general systems theory.* New York, Wiley, 1964.

Názov: Modelovacie formalizmy udalostných systémov

Autor: Gabriel Juhás

Vydalo vydavateľstvo:
RT systems s.r.o., Kopčianska 14, 851 01 Bratislava, v roku 2011

Vytlačili tlačiarne:
ZEPHIROS, a.s., Komárňanská 91, 821 09 Bratislava, v roku 2011

Prvé vydanie

ISBN 978-80-970519-1-4

ISBN 978-80-970519-1-4

9 788097 051914