

Prof. Ing. Ján Kolenička, PhD

# ÚVOD K INFORMATIKE



---

Banská Bystrica, 2004

V roku 2003 bola na istej celoslovenskej konferencii vyslovená výzva kolegov z humanitnej fakulty na napísanie materiálu, ktorý by im umožnil nielen povrchné vnímanie niektorých frekventovaných pojmov z informatiky, ale aj ich výklad na dostatočne zrozumiteľných – a napriek tomu pokiaľ možno presných – formuláciách.

Rovnako nás oslovujú aj niektorí kolegovia z prírodovedných katedier, ktorí vo väzbe na svoje predmety aplikujú pojmy informatiky, často sa na nás obracajú a žiadajú objasniť, vysvetliť či potvrdiť presný význam pojmov z informatiky, aby ich práca mohla byť precíznejšia. Napokon, aj pre mnohých pracovníkov v oblasti záujmovej činnosti, alebo aj pre bežného človeka žijúceho v prudko sa rozvíjajúcej informačnej spoločnosti, môže byť prospešné poznať jasnejší význam aspoň najzákladnejších používaných pojmov informatiky.

V akademickom roku 2001/2002 sme na Katedre informatiky zaviedli v prvom ročníku štúdia informatiky nový predmet *Úvod k informatike*. Vznikol z podnetu učiteľov informatiky z praxe, ktorí žiadali, aby sme zvýšili tlak na používanie presných odborných termínov v slovnom i písomnom vyjadrovaní pracovníkov, a to najmä z oblasti informatiky. Aj naše skúsenosti z praxe škôl potvrdzujú opodstatnenosť takýchto požiadaviek. Študenti, prichádzajúci k nám na štúdium z rozličných škôl, si nesú so sebou aj nesystematickosť a nepresnosť pri používaní odborných pojmov a termínov, ktorú im zrejme spôsobila ich nie dosť dôsledná stredná škola.

Preto vznikla táto kniha. Definuje skutočne len základné pojmy z informatiky. Vracia sa pritom ku koreňom informatiky, aby bolo zrejmé, ako a prečo pojmy historicky vznikali, ako sa menili a aktualizovali a ako sa používajú v súčasnosti. Nejde pritom o encyklopedický slovník, napriek tomu že sa tu prezentuje vyše dvesto termínov (hesiel). Každý čitateľ s maturitou zvládne na dostatočnej úrovni obsah aj formu predloženého textu, bohato ilustrovaného obrázkami.

Autor

---

Lektorovali: Doc. RNDr. Bohuslav Sivák, CSc  
Mgr. Vladimír Siládi  
Recenzoval PhDr. Igor Šimo

## 1. INFORMÁCIA A INFORMATIKA

Ťažko si dokážeme predstaviť modernú civilizáciu bez spotreby ohromného množstva energie. Tá prúdi nielen do oceliarní, automobilového priemyslu, do poľnohospodárstva, do dopravných zariadení, atď., ale napríklad aj do nemocníc, do tlačiarň kníh, do rozhlasových a televíznych vysieláčov, telefónnych sietí a prichádza za človekom i do jeho domácností alebo pracovní, kde svieti, hreje, vykonáva rôzne práce a napr. prostredníctvom rádia či televízoru nás baví, vzdeláva, alebo informuje o dani v spoločnosti. Presvedčujeme sa tak o tom, že energia neslúži len materiálnym potrebám človeka, ale i potrebám jeho ducha. Predstavme si, ako by asi vyzeral život v krajine či meste, keby sa raz minuli všetky zásoby energie a človek by musel vystačiť so svojimi holými rukami. Potom ľahko pochopíme, že energia je pre ľudskú spoločnosť životne dôležitá.

Práve tak sú preto rovnako dôležité aj prostriedky a spôsoby, ktorými sa dosahuje čo najvyššia *účinnosť* využitia energie v najširšom slova zmysle. Možno to formulovať aj tak, že v živote treba aplikovať také spôsoby a preferovať také postupy (*procesy*), ktoré pri minime spotrebovanej energii, alebo s minimálnou námahou, rýchle vedú k žiadanému cieľu. Jedna z oblastí ľudskej činnosti, ktorá to napomáha je *informatika*.

Ak chceme poznať, sledovať, upravovať alebo riadiť nejaký proces, udalosť či objekt, musíme mať o ňom isté signály. Pre každú takúto sústavu (stroj alebo živý organizmus) je charakteristická jedna všeobecná vlastnosť, ktorá je jednotná pre všetky sústavy: prvky takýchto sústav sú navzájom prepojené tak, aby si mohli pomocou signálov navzájom odovzdávať správy o procesoch (dejoch), ktoré v nich prebiehajú. Ale energetické pochody, ktoré sprevádzajú prenos signálov nie sú pritom podstatné a nehrajú principiálnu úlohu, podstatný je *signál*. Pochopme to: *televízor síce potrebuje na svoju činnosť energiu, ale nevytvára ju, podáva nám však informáciu, správu, a to vo forme vhodne spracovaného signálu.*

Pojem *informácia* je veľmi široký. Prenáša sa telefónom, rádiom, televíziou, počítačmi, uchováva sa na magnetických i kompaktných diskoch, v knihách, fotografiách a iných písomných podobách. Náš zrak, sluch, hmat nás informujú o stave v našom okolí, chuť, čuch nám hovorí o kvalite potravy, počítačové siete nám umožnia vyhľadávať informáciu vo svete, počítače dokážu informácie spracovať, transformovať, na základe nich riadiť určité procesy, atď.

Informácia je všetko, čo môžeme vnímať o nejakom fakte, jave, úkone alebo procese, ktorý sa udial alebo sa stane. Zhŕňame to do pojmu udalosť: *Z hľadiska informatiky nazývame*

*udalosťou výskyt nejakého javu, procesu, úkazu alebo faktu, ktorý nás zaujíma, a je istým spôsobom využiteľný.*

Väčšinou autori uvádzajú takúto definíciu informácie [11]:

#### **Informácia je**

- **oznam, správa, podávaná ústne, písomne alebo inak, o nejakej veci, alebo situácii,**
- **oznam, úzko spojený s procesom riadenia,**
- **odraz rôznostvárnosti medzi fubovoľnými objektmi a procesmi v živej a neživej prírode,**
- **oznam, ktorý znižuje stupeň neurčitosti u príjemcu.**

Vznik, prenos, pamätanie si, využitie a najmä *spracovanie informácií* prebieha v strojoch i živých organizmoch. Pravidlá, ktorými sa spracovanie informácií riadi nazývame *algoritmy spracovania informácií*.

Zamyslime sa teraz nad týmto faktom: fotografia, obraz, novinová stránka, disketa, hlasový opis atď., nemajú ako *fyzikálne objekty* vôbec nič spoločného s udalosťou, o ktorej sa ich prostredníctvom dozvedáme, existujú samostatne v priestore a čase a majú svoju vlastnú fyzikálnu povahu. Napriek tomu tento „obraz“ pôsobí za určitých podmienok na pozorovateľa tak, ako keby na neho priamo pôsobila samotná pôvodná situácia. Signál je teda tou *fyzikálnou veličinou*, ktorá vzniká pri nejakej udalosti, činnosti, jave. Má samostatnú fyzikálnu podstatu a existenciu, nezávislú na podstate udalosti samotnej. Existuje len v nejakom organizovanom systéme a je vždy viazaný na nejaký hmotný objekt alebo pochod. Preto môže byť signál nielen pozorovateľný, ale aj zaznamenaný (uchovaný) na vhodnom nosiči, alebo tiež prenášaný na vzdialené miesta, a to prostredníctvom prenosového kanálu, vytvoreného medzi zdrojom signálu a jeho príjemcom. To umožňuje pochopiť a uviesť ďalšiu dôležitú definíciu: **Signálom nazývame akúkoľvek fyzikálnu veličinu, ktorá nesie informáciu** (pozri kapitolu Signál).

Je zrejmé, že *významový* (sémantický) obsah pojmu informácia je **totožný s izomorfným vzájomným priradením** nejakej udalosti a signálu. Ak máme na jednej strane určitý jav alebo udalosť a množinu stavov tohto javu, a na druhej strane signál a množinu jeho stavov, *môžeme potom prvky týchto dvoch množín k sebe navzájom priradiť*. Informácia v *abstrahovanom zmysle* (bez konkrétnej náplne správy) je **výber** niektorého prvku z množiny možných prvkov. Každému stavu sledovaného javu môžeme *priradiť* určitý stav fyzikálneho procesu reprezentujúceho signál.

Už dávno bolo stanovené, že jednotkou, ktorou určujeme **mieru množstva informácie** je **bit** (**binary digit**). Vyššou informačnou jednotkou je  **bajt** (B, píše sa aj Byte), t.j. 8 bitov, ďalej **kilobajt** (KB) t.j.  $2^{10} = 1024$  bitov, **megabajt** (MB), t.j.  $1024 \text{ KB} = 1\,048\,576$  bitov, **gigabajt** (GB) t.j.  $1024 \text{ megabajtov} = 1\,073\,741\,824$  bitov, **terabajt** (TB), t.j.  $1024$

gigabajtov = 1024 krát 1024 MB. Sú aj inak označované jednotky, napr. **blok** (spravidla 128 B) alebo **sektor** (spravidla 512 B).

Univerzálnou pracovnou informačnou jednotkou vo vzťahu k signálu je *slovo*. **Slovo predstavuje usporiadanú štruktúru určeného počtu bitov** (štandardom je napr. 16, 32, 64 bitov), alebo je tento pojem používaný aj celkom voľne vo vzťahu k počtu bitov.

Špecifickou podobou informácie je *údaj*. **Údaj predstavuje výber jedného prvku z množiny prvkov definičného oboru, ktorý je údaju priradený** (v databázach tejto množine hovoríme *doména*). Údajom je napr. konkrétny dátum, hodnota nejakej premennej, meno mesta a pod. V počítači je údaj reprezentovaný *slovom alebo zoskupením slov* v zmysle predchádzajúceho odseku.

### 1.1 Informačná entropia

Vráťme sa však ešte k pojmu informácia. Je informácia užitočná? Do akej miery ovplyvňuje chod spoločnosti, konanie jednotlivca alebo správanie sa nejakého systému? Odpoveď na prvú otázku je zrejme: áno. Tak, ako je nemožná existencia spoločnosti a jej elementov bez energie, nie je možný ani jej pohyb a rozvoj bez informácií, viažúcich sa na procesy v tejto spoločnosť.

Aby sme mohli odpovedať na druhú otázku, predstavme si seba ako príjemcu informácie na jednom konci prenosového kanálu a prijímame rôzne správy. Intuitívne veríme tomu, že všetky správy nemajú pre nás rovnaký subjektívny význam. Niektoré sú pre nás nové a upútajú našu pozornosť, lebo sú neočakávané, iné sme už predtým počuli, takže nám neprinášajú nič nové. Pri týchto druhých dostávame množstvo informácie rovné nule.

Jednotlivé správy sú isto zaujímavé a poskytnú nám „skutočnú“ informáciu iba vtedy, ak je pre nás táto informácia nečakaná, matematicky povedané, ak je *náhodná*. Táto definícia je úplne objektívna i presná a vylučuje akúkoľvek intuitívnosť hodnotenia novosti a neočakávanosti. V tejto súvislosti bol definovaný pojem **informačná entropia**, ktorej hodnota vyjadruje **mieru neurčitosti** (neočakávanosti) výskytu nejakej správy.

Rozložme všetky možné správy  $S_i$  v nejakom poradí a priradíme každej správe pravdepodobnosť  $P_i$ , ktorá je rovná pravdepodobnosti jej výskytu pri skutočnom prenose. Tak dostaneme tzv. *konečnú schému*:

$$\begin{pmatrix} S_1 S_2 \dots S_i \dots S_n \\ P_1 P_2 \dots P_i \dots P_n \end{pmatrix},$$

zostavenú z úplnej množiny správ tak, že ak sa vyskytne jedna správa, nesmie sa súčasne vyskytnúť iná a ku každej správe sa priradí pravdepodobnosť jej výskytu. Pri každom prenose sa vyberá jeden z prvkov  $S_i$  v konečnej schéme. Určite bude vybraná niektorá správa (inak by prenos nemal zmysel) a teda súčet pravdepodobností musí byť rovný jednej:

$$\sum_{i=1}^n P_i = P_1 + P_2 + \dots + P_n = I,$$

pričom niektoré  $P_i$  môžu byť nulové.

Každá konečná schéma predstavuje istú neurčitosť výberu jej prvkov, ktorá zaniká až po vlastnom výbere. Stupeň neurčitosti je rôzny pre rôzne schémy. Napr. schéma

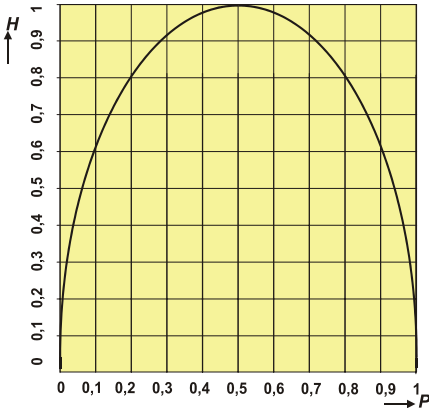
$$\begin{pmatrix} S_1 & S_2 \\ 1/2 & 1/2 \end{pmatrix}$$
 má vyššiu neurčitosť ako schéma

$$\begin{pmatrix} S_1 & S_2 \\ 0,01 & 0,99 \end{pmatrix},$$
 lebo v prvom prípade sú rovnako pravdepodobné obe správy, zatiaľ

čo podľa druhej schémy je takmer isté, že príde správa  $S_2$ .

Neurčitosť sa v takýchto prípadoch odstráni, keď jedna z pravdepodobností bude

rovná jednej a všetky ostatné budú nulové.



Miera neurčitosti v danej konečnej schéme je

vlastne zároveň mierou množstva informácií

pre jeden výber, lebo v okamihu výberu sa

neurčitosť odstraňuje. Miera neurčitosti

výberu, t.j. miera množstva informácie pre

jeden výber musí byť funkciou všetkých

pravdepodobností  $P_i$ , musí sa meniť spojitou

pri spojitých zmenách  $P_i$  a musí nadobudnúť

maximálnu hodnotu práve vtedy, keď sú

všetky pravdepodobnosti rovnaké.

Obr. 1.1 Priebeh entropie  $H$  v závislosti na pravdepodobnosti  $P$  pri výbere z dvoch možností s pravdepodobnosťami  $P$  a  $P - I$ .

Claude Shannon (jeden zo zakladateľov teórie informácie) ukázal [3], že riešeniu tohoto problému vyhovuje funkcia

$$H = -k \sum_{i=1}^n P_i \log P_i, \quad (1.1)$$

kde konštanta  $k$  určuje jednotku merania, veličina  $H$  sa nazýva **entropiou** súboru pravdepodobností  $P_i$ . Za jednotku množstva informácie alebo entropie sa berie jednotlivý

výber z dvoch rovnako pravdepodobných možností. Práve vtedy je  $H = 1$ , za základ logaritmu sa volí dvojka a vtedy je konštanta  $k = 1$ .

$$H(1/2; 1/2) = -(1/2 \log_2 1/2 + 1/2 \log_2 1/2) = 1. \quad (1.2)$$

Ak sa správy  $S_1$  a  $S_2$  môžu objavovať tak, že súčet pravdepodobností ich výskytov je rovný jednej, a ak výskyt prvej sa objavuje s pravdepodobnosťou blízkou nule, je temer isté, že možno očakávať objavenie sa správy  $S_2$ . V takomto prípade je entropia tiež blízka nule, správa  $S_2$  nezapôsobí ako neočakávaná. Veľkosť entropie je teda zároveň istou mierou množstva informácie, ktorú konkrétna informácia prináša a znižuje tak mieru neurčitosti u príjemcu. Potom je priebeh entropie daný obr. 1.1 nakreslený pre výpočty analogické vzťahu (1.2), pre meniace sa pravdepodobnosti (ktorých súčet však musí vždy byť rovný jednej).. Východzí vzťah (1.1) určuje entropiu jednotlivého výberu z množiny  $N$  správ (alebo symbolov) tj. predstavuje *entropiu na symbol* prenášanej správy. Túto veličinu nazývame aj *obsažnosť jednotlivej správy*.

Niektorí Shannonovi súčasníci sa domnievali, že spôsob výberu správ z množiny správ je problém rýdzo intuitívny, psychologický a teda nie matematický. Zásluhou Shannona sa intuitívnosť nahradila matematickou štatistikou, čo viedlo k tomu, že vzťah (1.1) bol nahradený entropiou

$$H = \sum_{i=1}^n P_i (-\log P_i), \quad (1.3)$$

čo však nie je nič iné, iba stredná hodnota záporného logaritmu zo vzťahu (1.1). Zdôraznime tiež, že veľkosť entropie sa nevzťahuje na jednotlivú správu, ktorá už bola vybraná, ale k jednotlivému výberu z danej množiny pri danom rozložení prevdepodobností. Inak povedané, *entropia* necharakterizuje to, čo bolo v danom okamihu vyslané, ale to, čo by mohlo byť vyslané. Preto sa entropia označuje tiež ako *miera voľnosti výberu* alebo *miera apriórnej nevedomosti*.

Uvedené myšlienky predstavujú „matematickú“ odpoveď na obe, v úvode kapitoly o entropii, položené otázky. Ak pri nejakom jave, procese alebo systéme získame informáciu o zmene jeho stavu (t.j. došlo teda k zníženiu stupňa neurčitosti), treba na túto zmenu reagovať tak, aby systém vykazoval žiadané (t.j. napr. optimálne, bezporuchové, bezproblémové, nenákladné, zdravé, atď.) správanie sa, potom je takáto informácia dôležitá a musíme ju zužitkovať. Človek to robí na úrovni svojho poznania, na základe skúseností, ktoré získal a neustále získava. Využíva pritom aj kolektívny rozum, ktorým spoločnosť vytvára a

aplikuje tie formy spätného pôsobenia, ktoré existujú v danej etape vývoja a hľadá a experimentuje s novými.

Informácia je takou formou existencie hmoty, ktorá pôsobí v procese vytvárania, udržiavania a zlepšovania podmienok života v prírode a spoločnosti, v celej rozmanitosti, v akej sa takéto univerzum prejavuje.

## 1.2 Informatika

Informatika je oblasť vedy, ktorá sa viaže na pojem informácie a zaoberá sa informáciami z hľadiska ich vzniku, ich štruktúry, spracúvania (transformácie), uchovávanía, prenosu a ich využítia.

Cieľom, účelom (i poslaním) informatiky je hľadať, vytvárať a aplikovať účinné a efektívne technické i netechnické prostriedky, metódy, postupy a teórie, ktoré človeku uľahčia prácu s informáciami alebo mu pomôžu prácu lepšie organizovať či významne redukovať. Tieto prostriedky spravidla umožnia zvýšiť či optimalizovať účinnosť alebo výkonnosť každého existujúceho systému. *Informatika je teda široká oblasť poznania a poznávania.*

Rovnaký cieľ má tak isto aj *výpočtová technika*, ba aj *kybernetika*. Čím sa *informatika* od nich líši? Odlišnosti nemusíme považovať za významné a vyjadrujú v skutočnosti len vplyv histórie na uvedené pojmy. Slovo *informatika* je termín, ktorý sa ustálil a bežne sa v Európe používa. Termín *výpočtová technika* je starší pojem, odvodený od amerického *computer science*. Použijeme ho vtedy, keď chceme zdôrazniť tú stránku informatiky, ktorá sa viaže na signál, t.j. na realizáciu počítačov, pamätí, atď., číslicovými obvodmi: zvyrazňujeme teda hardvérovú stránku prístupov (nie však izolovane od stránky softvérovej).

Termín *kybernetika* (je definovaná ako veda o riadení v živých organizmoch a strojoch) je dnes podstatne menej frekventovaný v reči bežného informatika, ale použijeme ho, ak chceme poukázať na dôležitosť problémov riadenia, alebo zdôrazniť význam riadiacich procesov v nejakom systéme.

Vzťah informatiky k ostatným vedám je obdobný ako vzťah filozofie (alebo kybernetiky) k nim: dotýka sa a ovplyvňuje všetky vedné odbory a oblasti ľudskej činnosti. Preto aj nástroje a prostriedky, ktorými informatika napĺňa svoj účel sú rôznorodé a je ich mnoho. Za základné považujeme

- technické prostriedky (hardvér),
- programové prostriedky (softvér),
- abstraktné prostriedky (vytváranie a aplikácia vhodných teórií),
- informačné technológie.

Nemožno povedať, že by toto rozdelenie predstavovalo navzájom nesúvisiace prvky. Ukazuje sa, že sa vzájomne ovplyvňujú, v niektorých momentoch prelínajú, často dokonca vzájomne podmieňujú a takmer vždy spoločne zvyšujú svoju kvalitu. Toto tvrdenie možno dokumentovať historickým vývojom výpočtovej techniky a informatiky.

Informatiku možno teda definovať aj takto: ***Informatika je systém, ktorého prvky sú prepojené informačnými väzbami a jeho správanie sa je procesom transformácie informácií a ich efektívneho zužitkovania.***

### **1.3 Historické pozadie informatiky**

Do súčasného stavu privedlo informatiku úsilie človeka vytvoriť také pomôcky, ktoré by zmenšili alebo odstránili námahu pri akejkolvek jeho činnosti, vrátane duševnej práce. Prvé myšlienky o konštrukcii počítača sa spájajú s menami *B. Pascala*, *Ch. Babbageho*, *G. Leibniza* (17–18. storočie). Skutočnými predchodcami dnešných počítačov sú však až stroje zo 40. rokov minulého storočia. Ide napr. o počítač *Mark 1* (1943), ktorý skonštruoval profesor Hardwaradskej univerzity *Aiken*, na ktorý neskôr nadviazal releový počítač *Mark 2* (programoval sa pomocou prepojovacieho panelu). Za praotca dnešných počítačov sa považuje elektronický počítač *ENIAC* (**E**lectronic **N**umerical **I**ntegrator **A**nd **C**alculator), vyvinutý *Johnom von Neumannom* na pensylvánskej univerzite (1942 - 1946). Práve pri tomto počítači sa prvýkrát použila spoločná pamäť pre inštrukcie programu i údaje, čo predstavuje revolučný prínos pri konkrétnej realizácii procesu spracovania programu.

Rozvíjajúc myšlienky *Von Neumanna* i teoretické práce *A. Turinga* vznikajú ďalej počítače *EDVAC* (**E**lectronics **D**iscrete **V**ariable **A**utomatc **C**omputer) a v novej verzii *UNIVAC*. Tento bol navrhovaný so zreteľom na potreby spracúvania komerčných agiend, pričom sa kládli vysoké nároky na výkonnosť a spoľahlivosť systému.

V Česko-Slovensku – ako reakcia na prudký rozvoj elektronických počítačov vo svete – vzniká v 50. rokoch (minulého storočia) releový počítač *SAP0* (**S**amočinný **P**očítač) a elektrónkový počítač *EPOS*. V nich sa uplatňujú nové pôvodné myšlienky, napr. použitie paritných kódov, dekadického aritmetiky, kódu zvyškových tried, vnútorného priradenia času a i prvkov multiprogramovania. Preto – najmä neskoršia tranzistorová verzia *EPOSU* – bola zaradovaná do svetovej špičky.

Centier, v ktorých sa vo svete vyvíjali počítače bolo niekoľko, okrem Ameriky napr. aj v Anglii a Nemecku. Aby sa mohli odlíšiť jednotlivé typy i stupne vývoja počítačov z hľadiska ich architektúry, prvkovej základne, používateľského prístupu, programového

vybavenie, jazykových prostriedkov a aplikačného určenia, vzniká na prelome 60. a 70. rokov pojem *generácia počítačov*. Obvykle sa rozlišuje päť generácií vývoja počítačov:

**I. generácia** (obdobie zhruba do konca 50. rokov minulého storočia). Z hľadiska technických prostriedkov ju charakterizujú relé, elektrónky, operačnú pamäť predstavuje magnetický bubon, konštrukcia je masívna, vyžaduje sa veľký príkon, spoľahlivosť je nízka. Úroveň *programovacích prostriedkov* je tu určená úrovňou strojového jazyka, prípadne úrovňou jednoduchého jazyka symbolických adries, užívateľ musí aplikovať individuálny prístup k počítaču.

**II. generácia** ( päťdesiate roky minulého storočia) sa viaže na vynález tranzistora a využitie feritových operačných pamätí. Účinná miniaturizácia umožňuje konštruovať rozsiahle a špecializované podoby počítačov: hovoríme už o veľkých počítačoch pre vedecké úlohy, o počítačoch špecializovaných na spracovanie dát, ale aj o počítačoch *prechodných*, ktorých rysy prechádzajú do 3. generácie.

Programovacie prostriedky druhej. generácie charakterizuje vznik vyšších programovacích jazykov (FORTRAN, ALGOL...) a tiež vznik jednoduchých operačných systémov. Začína sa uplatňovať filozofia prideľovania času a jednoduchých prerušovacích systémov, spôsob spracovania programov je dávkový (postupne jedna úloha za druhou).

**III. generácia** (šesťdesiate a sedemdesiate roky minulého storočia) je z hľadiska technických prostriedkov charakterizovaná tranzistorovými mikromodulmi a nižším či stredným stupňom hustoty integrácie. Okrem feritovej operačnej pamäte sa už používa aj polovodičová, čo má najväčší vplyv na zvýšenie operačnej rýchlosti počítača, a to rádovo na  $10^6$  operácií za sekundu. Rastie zároveň i kapacita operačnej pamäte na hodnotu 0,1 až 10 MB. Spracovanie informácií sa riadi mikroprogramovo a vznikajú rodiny počítačov (najmä IBM), čo vedie k modulárnej podobe počítačových zostáv. V oblasti programovacích jazykov vynikajú ďalšie, aj pre simuláciu vhodné jazyky, rozvíjajú sa prostriedky multiprogramovania, a to aj pre prácu v reálnom čase.

**IV. generácia.** Sem spravidla zahrnujeme počítače vznikajúce od polovice 70. rokov minulého storočia. Z hľadiska technických prostriedkov ich charakterizujú prvky vysokej a veľmi vysokej integrácie a použitie mikroprocesorov. Operačná pamäť je polovodičová a uplatňujú sa aj nové princípy holografie, laserovej techniky a pod., kapacita pamätí dosahuje 100 MB až 10 GB. Prudko rastie operačná rýchlosť na hodnoty až 100 miliónov operácií za sekundu. Okrem osobných počítačov vznikajú aj *viacprocesorové systémy, paralelné systémy a superpočítače*, príkon pritom klesá, miniaturizácia prudko akceleruje, automatizuje sa indikovanie chýb a ich odstraňovanie.

V oblasti programových prostriedkov vznikajú vysokošpecializované a konverzačné jazyky maximálne zjednodušujúce styk používateľa s počítačom, operačný systém začal obsahovať prvky pevnej logiky, veľký počet používateľov vyžaduje vytvárať podobu *distribovaných počítačových systémov*. Najzaujímavejší je fakt, že už v sedemdesiatych rokoch vznikajú rozsiahle *počítačové siete*.

V súvislosti s touto kategóriou počítačov sa začali používať aj ďalšie pojmy charakterizujúce stupeň vývoja prostriedkov výpočtovej techniky. Ide napr. o pojem *sálové počítače*. Boli charakteristické sústredením technických prostriedkov „do jednej miestnosti“ (vo výpočtovom stredisku) a v podstate boli prejavom vyzretosti počítačového priemyslu i úrovne spracovania informácií v zodpovedajúcej etape organizácie práce. V skutočnosti to však boli len rýchlejšie a lacnejšie verzie počítačov 3. generácie.

Koniec sedemdesiatych rokov bol obdobím ohromného rastu a inovácií najmä na poli *minipočítačov*. Rozličné výrobné série vyústili do podoby *mikroprocesorov a mikropočítačov*, z ktorých najznámejšie boli určené pre jedného používateľa: hovoríme o *osobných počítačoch*.

**V. generácia.** V skutočnosti ťažko nájdeme ucelenú definíciu počítačov piatej generácie; trauje sa, že sem patria počítače, ktorých vývoj začal v 80. rokoch minulého storočia a sú vyrábané od 90. rokov. Hovoríme, že ide o *moderné počítače*, ktoré obsahujú okolo  $10^6$  tranzistorov na jeden čip, operačná rýchlosť dosahuje rádovo  $10^{12}$  operácií za sekundu. Vznikajú triedy *paralelných počítačov* s extrémnym počtom procesorov, *relačné a databázové počítače* a nové typy počítačových a komunikačných sietí. *Programové a systémové prostriedky* charakterizuje systém strojového prekladu, spracúvanie *znalostných informácií* (riadenie bázy vedomostí, riešenie problémov a mechanizmy uvažovania), inteligentný medzistyk (porozumenie reči, obrazu), využitie *umelej inteligencie* na riešenie úloh (neurónové siete, genetické algoritmy ...) a *kvantové počítače*.

Názorným ukazovateľom kvality hardvéru vyrobených počítačov a s tým súvisiacou ich *miniaturizáciou* je stupeň integrácie číslcových prvkov, ktorý sa vyjadruje počtom tranzistorov (prípadne počtom hradieľ, t.j. základných logických členov) na jednotku plochy čipu. Ide teda o vyjadrenie určitej *hustoty prvkov* na štvorcový milimeter. **Čip je tenká kremíková doštička s rádovo milimetrovými rozmermi, na ktorej sa vytvárajú obvodové prvky s tranzistorami a prepojeniami.** Tak ako sa menili generácie počítačov, rástla aj hustota prvkov a tým aj stupeň integrácie. V stupni, ktorý sa označuje **SSI** (malá integrácia – small scale integration) bolo na čipe len niekoľko tranzistorov. V stupni **MSI** (stredná integrácia – middle scale integration) to už bolo niekoľko desiatok prvkov. Ďalší stupeň bol **LSI** (veľká

integrácia – large scale integration) so stovkami prvkov a **VLSI** (very large scale integration) už s tisíckami prvkov na čípe. V súčasnosti už technológia založená na kremíku dosahuje v stupni **ELSI** (extrémne rozsiahla integrácia – extra large scale integration) svoje hranice, s počtom desiatok miliónov prvkov na čípe.

Zhruba od sedemdesiatych rokov minulého storočia pojem generácie vo vzťahu k počítačom prestáva byť často používaný. Časová zložka hodnotenia počítača sa viac sústreďuje na jeho najpodstatnejší vnútorný podsystém a tým je **procesor**. Nasledujúca stručná chronológia ukazuje nesmierne dynamický rast kvality a schopností takýchto počítačových systémov, ktoré menia nielen svoju podobu, ale aj samotné prostredie, v ktorom sú aplikované:

**1971 – Mikroprocesor 4 004.** Bol to prvý mikroprocesor spoločnosti Intel a ako revolučný objav tvoril jadro kalkulačiek *Busicom*, pričom otvoril cestu vkladania inteligencie do osobných počítačov a iných číslicových zariadení. Obsahoval *asi 2300 tranzistorov s pracovnou frekvenciou 108 kHz*.

**1974 – Procesor 80 80.** Tento procesor bol mozgom prvého osobného počítača Altair, ktorý bol vraj pomenovaný podľa cieľa vesmírnej lode Enterprise z televízneho seriálu Star Trek a počítačoví fanatici si mohli sami taký počítač lacno zostaviť ako stavebnicu. Oficiálni výrobcovia v priebehu pár mesiacov predali desiatky tisíc počítačov zostrojených na báze uvedeného procesora. *Počet tranzistorov 6000, frekvencia 2 MHz*.

**1978 – Procesory 80 86, 80 88.** Firma IBM, v snahe vytvoriť novú divíziu osobných počítačov, použila procesor 80 88 ako mozog nového hitu, počítača IBM PC. Úspech procesoru 80 88 vyzdvihol Intel medzi spoločnosti rebríčku Fortune a časopis Fortune označil spoločnosť za jeden z podnikateľských triumfov sedemdesiatych rokov. Procesor obsahoval *29 000 tranzistorov a pracoval na frekvenciách 5 MHz, 8 MHz a 10 MHz*.

**1982 – Procesor 286.** Tento procesor, označovaný aj **80 286** bol prvým procesorom spoločnosti Intel, ktorý mohol pracovať aj so softvérom vytvoreným pre jeho predchodcu. Je veľmi dôležité, že kompatibilita softvéru je zachovaná pri všetkých mikroprocesoroch Intel. Tiež je pozoruhodné, že počas šiestich rokov od uvedenia na trh, sa predalo 15 miliónov počítačov s týmto procesorom. Viacnásobne vzrástol počet tranzistorov na *143 000 kusov v ňom, pracovná frekvencia 6 MHz, 8 MHz, 10 MHz a 12,5 MHz*.

**1985 – Intel 386.** Bol to *32-bitový* čip s podporou tzv. multiaskingu, čo znamená, že mohol spracovávať niekoľko programov naraz. Umožnil použiť prostredie *Windows* a tým aj *WYSIWYG*, čo už predtým umožňovali počítače *Apple* na báze procesoru *Motorola*.

Vyššie stonásobne vzrástol počet tranzistorov ( v porovnaní s mikroprocesorom 4 004) na **275 000 ks., pracovná frekvencia 16 MHz, 20 MHz, 25 MHz a 33 MHz.**

**1989 – Procesor Intel 486 DX.** Generácia procesorov 486 umožnila plynulý prechod z počítača, používajúceho príkazový riadok, na počítač s grafickým rozhraním. Požívateľ tak mohol po prvý raz pracovať s farebným počítačom a pracovať aj s rozsiahlymi textovými a obrazovými aplikáciami v rozumne prijateľnej rýchlosti. Procesor 486 bol ako prvý obohatený o *matematický koprocesor*, ktorý prevzal od centrálného procesora na seba realizáciu matematických funkcií, čím sa významne zrýchlil celý výpočtový proces. Počet *tranzistorov 1,2 miliónov ks, pracovná frekvencia 25 MHz, 33MHz, 50 MHz.*

**1993 – Procesor Pentium.** Názov Pentium sa mimoriadne rýchlo po uvedení stal bežne používaným pojmom a práve tak rýchlo sa na staršie procesory zabúdalo. Tento procesor umožnil počítačom jednoduchšie a ľahšie pracovať s faktormi zo skutočného sveta, ako je napr. reč, zvuk, písané písmo a fotografické snímky. Počet tranzistorov *dosahuje číslo 3,1 milióna a pracovná frekvencia je 60 MHz a 66 MHz.*

**1997 – Pentium II.** Tento procesor bol vytvorený v *technológii Intel MMX*, ktorá je určená na účinné spracovanie videa, zvuku a grafických dát. Procesor navyše umožňuje používateľom spracúvať digitálne fotografie aj na diaľku prostredníctvom internetu, vkladať do domáceho videa titulky, dodatkové zvuky, hudbu a v spojení s videotelefónom dokáže tiež posielat' video po štandardných telefónnych linkách i po internete. Počet *tranzistorov 7,5 milióna, rýchlosť zodpovedá 200 MHz, 233 MHz, 266 MHz a 300 MHz.*

**1999 – Celeron.** Tento procesor bol vyvinutý pre uplatnenie v počítačoch dostupných pre domáce použitie, pre hry a aj pre vzdelávací softvér. Uvádza sa niekoľko generácií tohto procesora: prvá mala 7,5 milióna tranzistorov (266 MHz), druhá 19 miliónov a dnešný Celeron – s taktovacou frekvenciou 1,1 GHz – má 27 miliónov tranzistorov. *Verzia s frekvenciou 1,2 GHz má dokonca 44 miliónov tranzistorov.* Dnešné verzie pracujú na frekvenciách od 500 MHz do 1,2 GHz.

**1999 – Pentium III.** Tento procesor sa líši predovšetkým v tom, že obsahuje sedemdesiat nových inštrukcií, ktoré zvyšujú výkon progresívnych grafických a 3D aplikácií, aplikácií pre plynulý prenos zvuku a videa, a programov pre rozpoznávanie hlasu. Predovšetkým však má podstatne skvalitniť používanie internetu. Pri jeho vytvorení bola použitá tzv. 0,25 mikrónová technológia. Obsahuje *9,5 milióna tranzistorov, rýchlosť určujú frekvencie 650 MHz až 1,2 GHz.*

**2000 – Pentium 4.** Ide o ďalšie zdokonalenie, ktoré umožňuje vyrábať filmy v profesionálnej kvalite, prenášať cez internet video v televíznej kvalite, komunikovať

v reálnom čase pomocou hlasu a videa. Technológia spojov dosahuje hrúbku 0,18 mikrónu, **počet tranzistorov je 42 miliónov, rýchlosť predstavuje frekvencia 1,3 GHz, 1,4 GHz, 1,5 GHz, 1,8 GHz a 2 GHz.**

#### 1.4 Spracovanie informácií

Povedali sme už, že prenos, pamätanie si, využitie a najmä **spracovanie informácií** prebieha v strojoch i živých organizmoch. Pravidlá, ktorými sa spracovanie informácií riadi nazývame **algoritmy spracovania informácií**.

Algoritmus je jeden z najstarších matematických pojmov, napr. niektoré algoritmy na riešenie numerických problémov poznali už Babylončania okolo roku 1800 pred n.l. Samotné slovo *algoritmus* je vraj odvodené od mena perzského matematika Muhammada Músá Al-Chvárizmího, ktorý žil okolo roku 825 n.l. v Chorezme (dnešný Uzbekistan).

Existuje celý rad definícií algoritmu, no i tak sa v nich podáva skôr vysvetlenie toho, čo pod tým pojmom rozumieme, než rigorózne slovné vyjadrenie: algoritmus sa nedá totiž rozložiť na jednoduchšie pojmy a tak ho pokladáme za taký základný pojem ako *množinu, zobrazenie* a pod. Predsa však, vzhľadom na nové pojmy, uvedieme najrozšírenejšiu Markovovu definíciu [8]:

**Algoritmus je presný opis, definujúci výpočtový proces, ktorý vedie od zmeniteľných východísk (vstupných) údajov až k žiadaným výsledkom.**

Pod pojmom *výpočtový proces* rozumieme vykonávanie stanovených operácií nad stanovenými operandami. Inak tiež hovoríme, že *výpočtový proces transformuje* vstupné informácie na výstupné. Nejde teda len o jednoduché počítanie (rátanie), ale o všeobecne chápané **operácie** najrozličnejších typov (triedenie, usporiadanie, písanie, kreslenie, aritmetické operácie, prenos, tlač, operácie so zvukom, videom, atď.).

Každý algoritmus musí vyhovovať týmto požiadavkám:

1. Musí byť **presný a zrozumiteľný**, t.j. v žiadnej etape výpočtového procesu nesmie pripustiť pochybnosti o tom, ako a kam postúpiť v procese ďalej. Toto je požiadavka **determinovanosti** algoritmu.
2. Musí vychádzať **z meniteľných východiskových údajov**, t.j. nemôže to byť opis jedinej úlohy, ale *celej skupiny príbuzných úloh* líšiacich sa len východzími údajmi. Toto je tzv. požiadavka *masovosti* alebo **hromadnosti** algoritmu.
3. Musí byť zameraný na **získanie určitých hľadaných výsledkov**. Z tejto tzv. požiadavky *rezultatívnosti* tiež plynie, že *počet krokov v algoritme musí byť konečný*.

Formulujme ešte jeden pojem: *Skupinu veľkého počtu úloh rovnakého typu, pri ktorých pre vstupné údaje hľadáme výsledky, t.j. výstupné údaje, nazývame algoritmičky*

**problém.** Chceme tým naznačiť, že existujú aj úlohy, ktoré nemožno riešiť algoritmom, dokonca takých úloh je väčšina. No na riešenie algoritmického problému sa snažíme nájsť jednotný postup, teda algoritmus.

Konkrétnu formuláciu algoritmu založenú na konkrétnej reprezentácii a štruktúre údajov nazývame **program** [9]. Pri vykonávaní, t.j. pri realizácii algoritmu (programu) prebieha istý dej – prebieha **proces**. Preto **vykonávateľ algoritmu sa nazýva procesor**.

Vytvorenie (pokiaľ možno najlepšieho) algoritmu pre riešenie daného algoritmického problému je vo všeobecnosti veľmi zložitý problém a robia to špecialisti. Proces vytvárania algoritmu a jeho zápisu vo forme **programu** prostriedkami **formálneho programovacieho jazyka** sa nazýva **programovanie**.

Programovací jazyk, ktorý vychádza zo štruktúry počítača (vo funkcii vykonávateľa algoritmu či programu) a ovláda priamo jeho podsystémy a prvky hardvéru sa **nazýva strojový jazyk** alebo **strojový kód**. Všetky vyššie programovacie jazyky, ktoré informatika poskytla programátorom, umožňujú rýchlejšiu, jednoduchšiu, ale aj univerzálnejšiu tvorbu programov, no vlastný beh programu *sa vždy v počítači realizuje v strojovom kóde*. Preto realizácia programu sa deje v dvoch etapách: Najprv **kompilátor** (prekladač) preloží program z vyššieho programovacieho jazyka do strojového jazyka a potom prebehne vlastná realizácia programu. Dnes sa však dajú bežne kúpiť už skompilované programy pre konkrétny operačný systém, takže prvá etapa odpadá.

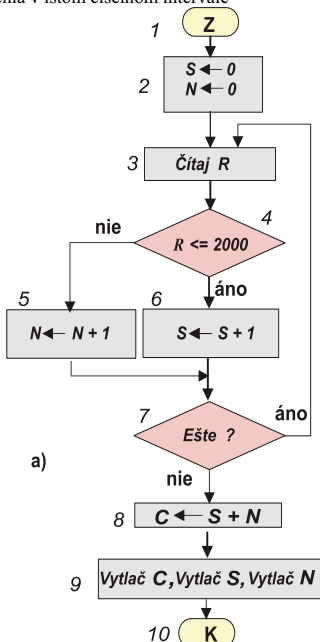
Celková **doba realizácie programu** (výpočtu) a **kapacita pamäte** procesoru, potrebná pre uvažovaný program a ním spracovávané údaje, sú hlavnými charakteristikami **kvality algoritmu** či programu.

## Jazyk zápisu algoritmov a programov

Tento odsek nemá za cieľ zaoberať sa algoritmicizáciou či programovaním, má len pripomenúť bežné ale dôležité relevantné pojmy, súvisiace so spracovaním informácie. Na obr. 1.2 a je ukážka vývojového diagramu, ako najnázornejšieho jazyka zápisu algoritmu. V princípe je to ohodnotený orientovaný multigraf (kap. 5), ktorého vrcholy (uzly) majú spravidla podobu obdĺžnika alebo kosoštvorca a spojnice tých vrcholov sú orientované hrany, vyjadrujúce vždy konkrétny postup v danom výpočtovom procese. Obdĺžnik je tzv. *operátorový (operačný) vrchol*, kosoštvorec považujeme za *podmienkový vrchol*, v ktorom sa podľa stanovenej podmienky algoritmus vetví, alebo prechádza na cyklus.

Uvedený algoritmus je veľmi jednoduchý: z nejakej množiny rokov  $R$  narodenia nejakých ľudí sa má nájsť počet  $S$  tých, čo sa narodili v minulom storočí a ďalej počet  $N$  ľudí narodených v novom storočí. Tento príklad isto nepotrebuje zvláštny komentár, ak sa dohodneme, že roky  $R$  mimo uvedené intervaly neexistujú.

Poznámka.: Ide o jednoduchý príklad na počítanie udalostí, pričom udalosťou sa tu rozumie výskyt roku narodenia v istom číselnom intervale



```

begin
  S := 0; N := 0;
Pokr: read R ;
  if R <= 2000 then S := S + 1
    else N := N + 1 ;
  if ESTE = true then Goto Pokr
  else begin
    C := A + B;
    write (C);
    write (S);
    write (N);
  end;
end;
    
```

b)

c)

KÓD OPERÁCIE	Adresová časť	
	A1	A2

d)

KÓD OPERÁCIE	Adresa	Priamy operand

Obr. 1.2 Zázpis algoritmu a) vývojovým diagramom., b) programovacím jazykom, c) inštrukcia s adresami, d) inštrukcia s priamym operandom

Z hľadiska potrebných pojmov však upozorníme na tento dôležitý fakt: Zázpis algoritmu v obr. 1.2 a jednoznačne vyjadruje, že použité veličiny (v matematike ich nazývame *premenne* alebo aj *operands* = prvky, ktoré vstupujú do operácie) v priebehu výpočtového procesu menia svoje hodnoty a namiesto *starých* hodnôt nadobúdajú *hodnoty nové*, podľa **operácie**, v ktorej sa uplatňujú. Tak napr. v bloku 2 vývojového diagramu veličina *S* nadobudla hodnotu 0 (nula), zatiaľ čo v kroku 5 tá istá veličina nadobudne novú hodnotu, o jedna väčšiu než bola stará hodnota. Ak sa na *S* dívame ako na jednoduché písmenko (symbol), potom ľahko pochopíme, že jednému a tomu istému písmenku (t.j. jednej a tej istej premennej) priradíme v rôznych etapách výpočtového procesu rozličné hodnoty. Z toho ale plynie, že ak je procesorom (t.j. realizátorom algoritmu) počítač, musí v ňom takému písmenku prislúchať **jedno a to isté miesto**, do ktorého sa postupne ukladajú meniace sa hodnoty. Preto je pri algoritimizácii a programovaní vhodnejšie a presnejšie používať termín **identifikátor premennej** (namiesto rýdzo matematického označenia *premenná*).

Miest, kam možno ukladať hodnoty identifikátorov je v počítači veľmi veľa; predovšetkým sú to miesta v *operačnej pamäti počítača*, ďalej sú to *pracovné a pomocné registre* rôznych dĺžok (čo do počtu bitov), ktoré slúžia na krátkodobé alebo inak určené

doby uchovávaní hodnôt pri spracúvaní programov. K takýmto miestam prístupujeme spravidla pomocou adresy. **Adresa je vo všeobecnosti špecifický údaj, ktorý určuje polohu toho miesta v systéme počítača, do ktorého chceme definované informačné slovo vložiť, alebo z ktorého chceme to slovo vybrať.** Z procesného hľadiska je lepšie hovoriť o zápise **do** resp. čítaní **z** pamäťového miesta.

Adresami sú spravidla čísla určujúce riadok v slovne organizovanej pamäti (každé slovo, ktoré predstavuje hodnotu pamätaného identifikátora, má svoj samostatný riadok), alebo sú to **súradnice** určujúce riadok a stĺpec v maticovo organizovanej pamäti, pričom tieto súradnice určujú polohu **pamäťovej bunky** takejto pamäte. Často bývajú adresami aj **mená** pracovných registrov v príslušných podsystémoch počítača. Z tohoto pohľadu možno teda aj *identifikátor premennej* považovať za adresu, prostredníctvom ktorej sa dostaneme k jeho hodnote. Adresa je teda pojem, ktorý sa jednoznačne viaže na štruktúru, t.j. na tie hardvérové prvky procesora, ktoré sa podieľajú na fyzickej realizácii príslušnej operácie.

Napr. blok 8 vývojového diagramu predstavuje *operáciu aritmetického súčtu*, podobne je aritmetickou operáciou v bloku 5 alebo 6 *pripočítanie jednotky* (tzv. *inkrementácia*), iným typom operácie je napr. v bloku 2 *priradenie* konštanty 0 (nula) identifikátoru *S* a celkom odlišným typom je napr. v bloku 9 *operácia tlače* hodnoty, ktorú nadobudol niektorý identifikátor. Posledne menovaný prípad je vlastne istou transformáciou dvojkového (bitového) zobrazenia hodnoty identifikátora v procesore, do podoby arabských číslíc, vytlačených na papieri alebo monitore.

Vo všeobecnosti teda operáciu možno definovať aj takto: **operácia je transformácia, ktorá z definovaných východných hodnôt vytvára cieľové hodnoty či tvary.** Ako vidíte, ide tu o zhodu pojmov, použitých už pri definícii algoritmu ako výpočtového procesu. Dodajme ešte, že podsystém počítača, v ktorom sa realizujú operácie sa nazýva **operačná jednotka**. Pri aritmetických alebo logických operáciách hovoríme, že ich realizuje **aritmeticko – logická jednotka**. Pamäť, ktorá spolupracuje s operačnou jednotkou sa nazýva **hlavná** alebo **operačná pamäť**. V nej je vždy uložený celý aktuálne prebiehajúci program aj so všetkými údajmi, ktoré sú takýmto programom spracúvané.

Jednoduchý usporiadaný súpis operácií vo vytvorenom algoritme nie je sám o sebe realizáciou výpočtového procesu. Realizáciou sa stane iba v tom prípade, ak **každú operáciu v algoritme jednoznačne združíme** (do dvojice) **s príkazom na jej vykonanie**. Tak vznikne nová tvár algoritmu v podobe presného *sledu príkazov*, ktorý treba vložiť do procesora, aby sa algoritmus skutočne v tomto procesore realizoval a človek tak našiel riešenie svojho algoritmického problému v podobe, ktorú si želal. Medzi *operáciou* a *príkazom na jej vykonanie* je teda *izomorfný* vzťah. **Usporiadaný sled príkazov sa nazýva program**. Príkladom programu v istom programovacom jazyku je obr. 1.2 b, kde každý riadok predstavuje jeden príkaz. Často sa za príkaz považuje aj skupina príkazov nachádzajúcich sa medzi k sebe patriacimi tzv. zátvorkami **begin** a **end** (v jazyku Pascal),

resp.  $\{ \}$  (v jazyku C). Takúto podobu nazývame **zložený príkaz**. Príkladom zloženého príkazu vo vývojovom diagrame môže byť aj zápis napr. v bloku 2 vývojového diagramu.

Slovo *príkaz* je isto dostatočne zrozumiteľné, ale s ohľadom na pojem *identifikátor*, mu treba ešte pridať ďalší prívlastok a definovať tzv. **prirad'ovací príkaz**. Uvažujme napr. jednoduchú rovnicu  $C = A + B$ , kde  $A, B, C$  sú nejaké premenné, napr. sú to celé čísla. Je zrejmé, že po vyčíslení výrazu na pravej strane rovnice **nadobudne** premenná  $C$  hodnotu, ktorá je aritmetickým súčtom hodnôt  $A$  a  $B$ .

Teraz pre zmenu uvažujme zápis s premennými  $S = S + I$ . Z matematického hľadiska ide o hrubú nepravdu. Ak ale  $S$  je identifikátorom (t.j. adresou alebo menom miesta, do ktorého sa ukladá jeho hodnota), potom uvedený zápis možno interpretovať (a to pravdivo) takto: zober starú hodnotu identifikátora  $S$ , zväčši ju o jednotku a **výsledok prirad' identifikátoru  $S$  ako jeho novú hodnotu**. Takáto interpretácia sa v algoritmoch zapisuje takto:  $S \leftarrow S + I$  alebo  $S := S + I$ . Znak  $\leftarrow$  alebo  $:=$  je **znakom priradenia** a príslušný príkaz sa nazýva **prirad'ovací príkaz**.

Stretávame sa aj s názornejším (starším) spôsobom zápisu, ktorý v uvedenom príklade je takýto:  $S \leftarrow \langle S \rangle + I$ . Tu  $S$  predstavuje adresu a  $\langle S \rangle$  indikuje, že ide o obsah tejto adresy, t.j. o hodnotu identifikátora  $S$ .

Opäť tu priam cítiť väzbu prirad'ovacieho príkazu na príslušný hardvér, ktorý prirad'ovací príkaz realizuje v prvkoch svojej štruktúry. John von Neuman označil príkaz ovládajúci prvky štruktúry procesora slovom **inštrukcia**. Inštrukcia je príkaz, ktorý je prvkom (riadkom) programu zapísaného v strojovom kóde. **Potom program, zapísaný v strojovom jazyku, je usporiadaným sledom inštrukcií**. Takýto program sa ukladá do operačnej pamäte počítača v rovnako usporiadanom slede adries prislúchajúcich jednotlivým inštrukciám. Objav inštrukcie bol geniálnym krokom, ktorý viedol na taký režimu spracovania programu, ktorý označujeme ako **samočinný**.

**Inštrukcia** je teda informačné slovo, (so štandardným počtom bitov v zodpovedajúcej štruktúre procesora), ktoré **sa skladá z dvoch častí** (obr. 1.2 c): prvá časť predstavuje tzv. **kód operácie** (alebo *operačný znak*) a druhá je tzv. **adresová časť**, ktorá obsahuje jednu alebo niekoľko adries, na ktorých sú uložené hodnoty tých operandov (identifikátorov), ktoré vstupujú do operácie určenej kódom operácie v tejto inštrukcii. Bežné sú inštrukcie dvojadresové, ktoré spravidla na počiatku obsahujú na adresách  $A1$  a  $A2$  hodnoty príslušných dvoch operandov a po vykonaní takej inštrukcie sa *výsledok operácie priradí* identifikátoru s adresou  $A1$ , čím sa prepíše bývalá hodnota prvého operandu výsledkom operácie.

Pozrime sa ešte raz na príkazy napr. v bloku 2 alebo 5. Z hľadiska vyššie uvedeného textu sú zaujímavé tým, že sa v nich pracuje s konštantami (ich hodnoty sú  $0$  resp.  $1$  v tomto príklade) a o nich sme zatiaľ nepovedali, kde sú v počítači uložené. Dúfam však, že je už zrejmé, že niekde byť uložené musia, práve tak, ako hodnota ľubovoľného iného

identifikátora premennej. *Prvé* riešenie problému *čo s konštantami* je teda analogické: uložíme ich na vopred určené adresy v pamäti a príslušné adresy potom použijeme v príslušných inštrukciách.

*Druhé* riešenie možno dosiahnuť použitím takej inštrukcie, ktorá umožňuje zapísať namiesto adresy v jej adresovej časti priamo hodnotu konštanty alebo operandu. Taká inštrukcia je naznačená na *obr.1.2. d* a je v praxi bežne využívaná. Zrejme aj príslušný kód operácie v takejto inštrukcii musí rešpektovať fakt, že adresová časť neobsahuje len adresy operandov.

Osobitným typom inštrukcií sú tzv. **skokové inštrukcie**. Sú to inštrukcie, ktoré síce pracujú s bežnou predstavou adresy (t.j. číslo určujúce miesto v pamäti ako napr. v *obr. 1.2 c*), no na tom mieste nie je uložená hodnota nejakého operandu, ale je tam adresa tej inštrukcie programu, ktorá sa má vykonať ako ďalšia v poradí. Ak napr. operačný kód v takejto inštrukcii vyjadruje tzv. **podmienенý skok**, testuje sa splnenie istej podmienky. Ak podmienka splnená **je**, potom program bude bezprostredne pokračovať realizáciou inštrukcie ktorá je uložená na adrese *A1*, ak splnená **nie je**, program bude bezprostredne pokračovať inštrukciou, ktorá je uložená na adrese *A2*. Z toho jednoznačne plynie, že tak, ako sa do pamäte ukladajú hodnoty identifikátorov na určené adresy, ukladajú sa aj jednotlivé inštrukcie programu do pamäte na jednotlivé adresy v takom usporiadanom slede, v akom bol program vytvorený.

Praktickejšie a rýchlejšie riešenie podmieneného skoku má takú podobu inštrukcie, v ktorej je na mieste adresy *A1* zapísané meno nejakého registra, napr. *Z* a spomínaný test skúma napr. znamienko čísla zapísaného v *Z*. Ak je v *Z* číslo so znamienkom *minus*, program bude pokračovať inštrukciou uloženou na adrese *A2* (skok ako v predchádzajúcom prípade), ale ak ide o znamienko *+*, potom sa bezprostredne bude realizovať tá inštrukcia, ktorá leží hneď na najbližšej vyššej adrese v pamäti vzhľadom k adrese práve realizovanej skokovej inštrukcie.

Jednoadresovú podobu má aj inštrukcia *nepodmieneného* tzv. **tvrdého skoku**. Jej operačný kód vyjadruje príkaz **prejdi na** tú inštrukciu, ktorej adresa je uvedená v adresovej časti inštrukcie. Nejde teda o žiaden test, program bude bezprostredne pokračovať inštrukciou uloženou na určenej adrese.

Každý počítač poskytuje používateľovi – programátorovi desiatky až stovky rozličných inštrukcií, aby mohol spracúvať čo najširšiu triedu úloh. Vzhľadom na izomorfizmus *operácia – inštrukcia* to znamená, že je k dispozícii dostatočná množina operácií realizovateľných hardvérovými podsystemami počítača. Ak sme operáciu definovali ako istú *transformáciu* (pozri vyššie), nemusi pri jej realizácii ísť o jedinú elementárnu činnosť. Napr. operácia **SPOČÍTAJ**  $a + b$  sa v procesore v skutočnosti realizuje takto (skrátaná podoba):

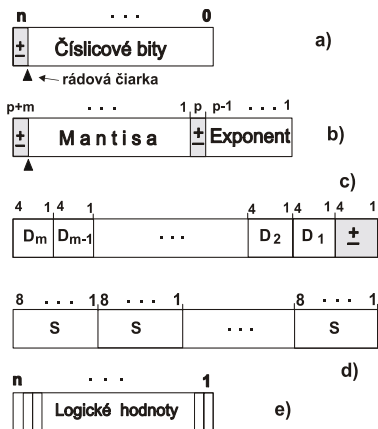
1. vynuluj sčítací register (zhromažďovač)
2. vyber z pamäte *a*
3. vlož ho do sčítacieho registra

4. vyber z pamäte  $b$
5. pripočítaj ho k obsahu sčítacieho registra
6. ulož výsledok do pamäte
7. ukončí operáciu.

Vidno, že uvažovaná operácia SPOČÍTAJ sa v toto príklade rozpadá na sedem tzv. elementárnych činností. Vo všeobecnosti hovoríme, že **operácia sa realizuje prostredníctvom mikrooperácií**, t.j. že jej prislúcha istý **mikroprogram**, ktorý je usporiadaným sledom tzv. **mikroinštrukcií**. Týmto problémom sa však tu nebudeme naďalej zaoberať.

### Zobrazenie údajov v počítači

V súvislosti s termínom *slovo* sme z hľadiska samočinnosti procesora (hardvéru) definovali vyššie *slovo typu inštrukcia*. Z rovnakého dôvodu definujeme kvôli úplnosti ešte ďalšie typy informačných slov viazúcich sa na zobrazenie v počítači: *slovo typu číslo* a *slovo typu text*, vo všeobecnosti hovoríme o *numerickej* (číselných) a *nenumerickej* (nečíselných) údajoch [2].



Základnou formou vyjadrenia číselných hodnôt sú **dvojkové čísla**. Číslo  $N$  s **pevnou rádovou čiarkou** sa zobrazuje formátom podľa obr. 1.3a. Rádová čiarka je umiestnená za znamienkovým bitom  $n$ , ktorý (v prípadoch 1.3a,b) svojou nulovou hodnotou indikuje číslo *kladné* a ak je v ňom hodnota 1, potom ide o číslo *záporné*. Ostatné bity patria mantise zobrazeného čísla. Rozsah zobraziteľných čísel je daný počtom bitov mantisy. (Ďalšie podrobnosti o zobraziteľnosti záporných čísel prislúchajúcimi kódmi tu uvádzať nebudeme).

Obr.1.3 Zobrazenie numerickej a nenumerickej slov

Číslo  $N$  zobraziteľné vo formáte podľa obr.1.3a patria do číselného rozsahu  $-(1-2^{-n}) \leq N \leq 1-2^{-n}$ .

Číslo z iného rozsahu treba do tohoto formátu transformovať a robí sa to takzvanou zmenou mierky.

Formát podľa obr.1.3b predstavuje princíp zobrazenia dvojkového čísla s **pochyblivou rádovou čiarkou**. Slovo je rozdelené na dve časti: prvá patrí mantise čísla a znamienku čísla, druhá exponentu a jeho znamienku. Mantisa čísla so svojim znamienkom tu zodpovedá podobe ako v pevnej radovej čiarky, ale exponent so svojim znamienkom určuje skutočnú

veľkosť zobrazeného čísla. Nejde o nič iné, len o vyjadrenie hodnoty čísla  $N$  v známom semilogaritmickom tvare, ktorý možno zapísať takto:

$$N = \pm M \cdot Z^{\pm E}$$

kde  $M$  je mantisa čísla,  $Z$  je základ číselnej sústavy a  $E$  je exponent čísla.

Kvôli názornosti uvažujme desiatkovú sústavu ( $Z=10$ ) na tomto príklade: máme číslo  $N = +0,25783 \times 10^{+4}$ . Zrejme ho môžeme zapísať aj takto:  $N = +2578,3$ . Jasne vidno, že čísla sú zhodné, líšia sa len vo forme zápisu z ktorého je zrejme, že sa rádová čiarka formálne presunula (je teda „pohyblivá“). Základ sústavy sa vo formáte podľa *obr. 1.3b* ani nevyjadruje (v procesore je implicitne daný).

Prečo sa však čísla s pohyblivou rádovou čiarkou zaviedli? Výpočtové procesy s reálnymi číslami sú oveľa častejšie než s číslami celými a ďalej, rozsah čísel v pevnej rádovej čiarky je daný počtom bitov mantisy, čo vedie k radikálnej redukcii veľkosti množiny použiteľných čísel. Riešením je práve prechod k zobrazeniu čísel formátom s exponentom, lebo napr. pre exponent  $E = \pm 256$  môžu do výpočtového procesu vstúpiť čísla z rozsahu od  $2^{-255}$  do  $2^{+255}$  (pre dvojkovú číselnú sústavu),  $10^{-255}$  do  $10^{+255}$  (pre desiatkovú sústavu), čo je zrejme podstatne širší rozsah než pri pevnej rádovej čiarky. A pritom, keby sme exponent v štruktúre slova podľa *obr. 1.3b* vyjadrili dvojkovo, zabral by len 8 bitov ( $2^8 = 256$ ).

Uvedomme si, že zobrazenie čísel v počítači sa jasne viaže na použitý počet bitov v danom slove (ktorý je vždy nejaký limitovaný), a preto (na rozdiel od matematiky,) číselná os v počítači nie je nekonečná ani spojitá ale je „deravá“. Dve susedné počítačové  $n$ -bitové čísla (mantisy) sa totiž líšia o hodnotu vyjadrenú najnižším bitom, t.j. o číslo  $2^{-n}$ , ktoré je teda najmenším zobraziteľným číslom v počítači. Dĺžka počítačovej číselnej osi je určená rozsahom zobraziteľnosti čísel pre použitý formát. Pre pohyblivú rádovú čiarku platí, že počet bitov mantisy je spravidla menší než pri pevnej rádovej čiarky. Z toho plynie, že hoci je rozsah zobraziteľnosti čísel oveľa väčší, príslušná číselná os počítača je „deravejšia“ ako pri pevnej rádovej čiarky.

**Čísla v desiatkovej sústave** sa najčastejšie zobrazujú niektorým z desiatkových štvorbitových kódov. Preto sa zobrazujú formátom *s premenným počtom segmentov* podľa *obr. 1.3c*. Desiatkové číslo  $N$  sa zobrazí v tvare  $N = D_{m+1}D_m \dots D_2D_1$ , kde  $D_i$  je štvorbitový kódový obraz desiatkovej číslice  $0, 1, \dots, 9$ . Štvorica bitov číslice  $D_{m+1}$  býva umiestnená na konci slova a označujeme ju ako **znamienkové miesto**. Bez podrobností uvedme teraz len toľko, že číslice  $0$  až  $4$  v znamienkovom mieste spravidla vyjadrujú, že číslo  $N$  v slove je

kladné, číslice 5 až 9 určujú záporné číslo. **Dĺžka slova** podľa formátu v *obr. 1.3c* býva najčastejšie z intervalu 1 až 31.

Pozrime sa teraz na formáty slov pre **nečíselné údaje**. Patria sem predovšetkým **textové reťazce**, v ktorých každý prvok (písmeno, symbol, číslica, znak) je štandardne kódovaný 8 bitmi, t.j. rozsahom jedného **bajtu**. Reťazec **R** symbolov je reprezentovaný postupnosťou bajtov v tvare

$$R = SS...S, \quad \text{kde } S \text{ je 8 bitový kód symbolu.}$$

Aj tu je reťazec zobrazovaný *formátom s premenlivou dĺžkou* (*obr. 1.3d*), ktorá najčastejšie nadobúda hodnotu z intervalu 1 až 256 bajtov.

Osobitným druhom údajov sú **slová s logickými hodnotami**. Každá logická premenná nadobúda hodnotu 0 alebo 1, takže pre ňu postačuje jeden bit v slove, napr. podľa formátu na *obr.1.3e*. Dĺžka takého slova býva štandardná a počet jeho bitov sa najčastejšie zhoduje s počtom bitov čísla v pevnej rádovej čiarky. (Znamienko zrejme v prípade logických hodnôt stráca zmysel).

## Vlastné spracovanie

Lahko sa dá zistiť, že súbežne s vývojom technických prostriedkov dochádza a priori aj k zmenám v spôsobe spracovania informácií – jednoznačne to dokazuje vývoj programových prostriedkov a *operačných systémov (OS)*. Staršie operačné systémy typu DOS (diskový operačný systém) sa nahrádzajú, alebo niekde už aj nahradili novými systémami typu *Windows, UNIXC alebo Linux*. Existuje viacero definícií operačného systému, ktoré sa odvodzujú napr. od hardvéru, iné od programov, ďalšie od používateľského prostredia. Tu si pripomeňme definíciu operačného systému podľa [10], ktorá je dostatočne výstižná i všeobecná. **Operačný systém je množina softvérových produktov, ktorá spoločne a jednotne riadi systémové zdroje i procesy, využívajúce tieto zdroje v počítačovom systéme.** Inými slovami, *operačný systém umožňuje* konfigurovať, spájať a kombinovať hardvérové prostriedky počítačového systému, rozsiahle svoje softvérové produkty a ich zložky, i vlastné používateľské programy do jednotného systému, ktorým možno *efektívne a optimálne realizovať* samotný *proces spracovania informácií* [33].

Konkrétne podoby spracovania tu však nebudeme podrobne uvádzať, sú na to vo vyučovaní samostatné predmety. Poukážeme len na dve základné podoby: dávkové

spracovanie a spracovanie v reálnom čase, pričom samotný beh programov sa viaže na použitý operačný systém.

**Dávkové spracovanie** úloh v počítači je spôsob, pri ktorom sa niekoľko programov uložených vo vonkajšej pamäti počítača postupne (po dávkach) premiestňuje do operačnej pamäte počítača a spracúva sa v procesore (základnej jednotke) tohoto počítača, ak sú k dispozícii všetky prostriedky a údaje vyžadované týmto programom. Dávkový režim spracovania programov bol charakteristický pre sálové (strediskové) počítače, ktoré v princípe mohli byť, ale v skutočnosti neboli veľmi efektívne využívané. Potrebný časový fond používateľa v etape prípravy, ladenia a realizácie programu v tomto systéme je totiž značný, a preto veľmi vyhovujúci. Ľahko si pri dávkovom spôsobe spracovania možno predstaviť prípad, keď prebiehajúci program v sálovom počítači čaká na údaje, ktoré práve zbiera používateľ programu na svojom pracovisku mimo výpočtového strediska. Ak takéto úvahy zovšeobecníme, dospejeme k poznaniu, že pre efektívne riešenie je nevyhnutné, aby **údaje** pre spracovanie **vstúpili do počítača v čase svojho vzniku a v mieste svojho vzniku**, alebo boli k dispozícii v mieste ich spracovania ! Preto klasické výpočtové strediská strácali postupne svoj zmysel a nastala nová etapa zavádzania a využívania osobných počítačov.

Osobitné spracovanie informácií, určené vonkajšími vplyvmi a realizované v časovom súlade (v synchronizácii) s procesmi, ktoré sú vysielacími údajov na spracovanie a prijímačmi jeho výsledkov, sa nazýva **spracovanie v reálnom čase**. Takéto spracovanie sa využíva najmä na riadenie technologických procesov [32], ale aj všade tam, kde odozva počítača na prijaté informácie musí prísť skôr, ako by prestala byť aktuálna. Vzhľadom na to, že tento systém spracovania vyhovuje rozsiahlej množine úloh z hľadiska efektívneho využívania času, prostriedkov a ekonomiky, pracujú v súčasnosti v tomto režime rôzne druhy informačných systémov v širokom zábere života spoločnosti: v technických systémoch, v obchode, vo výrobe, vo vzdelávaní, v sociálnych systémoch, v zdravotníctve, vede, kultúre, štatistike, ekonomike a pod. Svoju rolu tu majú aj špecializované procesory.

V tejto súvislosti je veľmi dôležitý **vzťah komunikácie a počítačovej siete**. Význam komunikácie medzi počítačmi sa zvyšoval z jednej generácie do druhej. Dnes už je množstvo počítačov pripojených k sieti alebo sú počítače priamo prvkami komunikačných sietí. Hranica medzi komunikačnou a počítačovou technológiou už nie je veľmi zrejmalá a v budúcnosti sa môže celkom rozplynúť. S tým sa spája aj problém vhodnej distribúcie informácií v priestore, ich obnovovania, zrkadlenia a koexistencie.

## 1.5 Využitie informatiky v praxi

Povedali sme už, že informatika hľadá, vytvára a aplikuje účinné a efektívne technické i netechnické prostriedky, metódy, postupy a teórie, ktoré človeku uľahčia jeho činnosť alebo mu pomôžu prácu lepšie organizovať, a tak zvýšiť jeho účinnosť alebo výkonnosť. Veľmi zaujímavé je, že samotný názov **informatika** vznikol vo Francúzsku na pomenovanie predmetného vedného odboru, ktorého hlavnou úlohou bolo urýchliť prenos výsledkov vedy do praxe. Zdôrazňuje sa tým skutočnosť, že výsledky vedy majú mimoriadny význam pre ekonomiku, že množstvo výsledkov vedy veľmi rýchlo narastá a ich prenos do praxe treba preto urýchliť. Tak isto je nevyhnutné zvládať a rozvíjať metódy a prostriedky tohto nového vedného odboru tak, aby sa spoločnosť – ak nechce stagnovať – preorientovala z *priemyselno-technickej* na **informačnú**. Informatika, ako mladý vedný odbor v ekonomickom procese spoločnosti môže vystupovať ako **predmet**, ako **vzdelávanie** a ako **prostriedok**.

Ako *predmet* sa realizuje v dvoch oblastiach: formou technických prostriedkov a programových prostriedkov. Aj keď istý čas mali prioritné postavenie technické prostriedky, pre spojenie s praxou je dôležitejšie vedieť ovládať a používať aj programové prostriedky.

Informatika ako *predmet vzdelávania* nadobudla taký význam, že hovoríme o **druhej gramotnosti**. V tejto oblasti je stále čo robiť, lebo počet používateľov informatiky bude naďalej narastať.

Najdôležitejšia je však informatika ako *prostriedok* budovania informačnej spoločnosti. Možno dokonca povedať, že informatika ako prostriedok v ekonomike **pridáva hodnotu**. Pridáva hodnotu vo vede, vzdelávaní, rozhodovacom i výrobnom procese. Dokonca vytvára nové ekonomické prostredia. Ide však o oveľa viac: ide o novú filozofiu, v ktorej kľúčovú úlohu hrajú informácie, ich **zber, prenos, uchovávanie, spracovanie a najrôznejšia forma prezentácie**. Problém budovania informačných systémov nadobudol celosvetový charakter, a preto hovoríme o **globálnej informačnej spoločnosti**.

Pojem *informačná spoločnosť* opisuje morálnu a sociálnu víziu spoločnosti, v ktorej je povinnosťou ľudí vymieňať si informácie a tak isto ich aj chrániť. Prvoradým cieľom štátu by malo byť túto výmenu informácií umožniť a prostriedky na infraštruktúru umožňujúcu tok informácií by mal tiež poskytovať štát.

## 1.6 Informačné technológie

Nemožno pochybovať o tom, že v súčasnosti sa už široko rozvinulo mnohotvárne používanie informačných technológií (**IT**) a to už nielen vo vedeckej komunite, ktorá ich

začala vyvíjať ani nie pred šestnástimi rokmi, a ktoré sa už významne prejavujú v mnohých aspektoch fungovania systémov. Predpokladá sa nárast dôležitosti tých hospodárstiev (knowledge-based economies), ktoré budú založené na vedeckých poznatkoch. Pritom tieto hospodárstva sú samy o sebe zdrojom nových poznatkov, (ktoré možno odhaliť výskumom) a tiež aj hlavným činiteľom vo vzdelávaní a príprave vedcov i výkonných pracovníkov. Deje sa to v široko chápanom kontexte rozvoja globálnej (svetovej) informačnej infraštruktúry (GII – global information infrastructure) a vzniku globálnej informačnej spoločnosti (GIS – global information society), ktorá reprezentuje vzrastajúcu dôležitosť politiky vlád v tejto oblasti.

Čo sú to teda informačné technológie, čo si pod týmto – dnes tak často frekventovaným slovným spojením – treba predstaviť? Uveďme nasledujúcu definíciu[ 4 ]:

**Informačné technológie sú systémom metód, programov a aktivít, ktorými sa realizuje maximálne využitie blízkyh i vzdialených zdrojov informácií uchovávaných na širokej triede informačných médií, alebo tvorba takýchto zdrojov, a to prostredníctvom komunikácie v počítačových a telekomunikačných sieťach, s cieľom nájsť optimálne riešenie stanovených problémov a úloh, alebo dosiahnuť svoje zámery, či uspokojiť svoje potreby.**

Táto definícia skutočne vystihuje celú šírku uvažovanej problematiky. Napríklad v oblasti priemyselných a hospodárskych aplikácií sa informačné technológie uplatnia pri hľadaní (celosvetovo) najlepších know-how pre technologické postupy, manažment, riadiace procesy; možno sa kontaktovať s najlepšimi poradcami v ľubovoľnej sfére. V oblasti výroby už výrobcovia začali inzerovať a ponúkať svoje výrobky, mnohí používatelia si robia meno naplňaním svojich **www** stránok informáciami o sebe a svojich aktivitách, cez sieť sa začali obchodné aktivity (tzv. e-business). V sociálnej oblasti a oblasti zamestnanosti umožňujú informačné technológie sledovať široký trh práce v asociovaných spoločenských, a využívať ho pre naplnenie svojich cieľov. Lahko možno nájsť adresy i personálne obsadenie rôznych úradov, inštitúcií, skladbu riadiacich orgánov v oblasti ich pôsobenia, sú dostupné všetky normy, predpisy a nariadenia v širokej škále zameraní a záujmov technického, ekonomického, vojenského, sociálneho, spoločenského, kultúrneho i vedeckého charakteru.

Častá je už dnes i skratka **IKT** pre **informačné a telekomunikačné technológie**, čo zvyčajne postavenie komunikácie pri aplikáciách informačných technológií. Každý účastník sa môže spojiť s hocikým, kto kontakt neodmieta a v tomto styku môže zdokonaľovať svoju osobnosť. Dnes sú bežné diskusné skupiny a kluby, vzdelávacie krúžky apod., rozoberajúce

najrôznejšie témy, pričom ich účastníci môžu ponúknuť svoje názory a inšpirovať sa názormi iných. Sú však ešte krajiny, ktoré obmedzujú kontakty v internete.

V oblasti vzdelávania je tiež situácia veľmi priaznivá: môžem si vybrať ľubovoľnú učebnicu vytvorenú na multimedialnom médiu a v časových intervaloch, ktoré mi vyhovujú, naberať nové vedomosti zvukom, obrazom, v statickom, alebo dynamickom zobrazení. Internetom sú dostupné takmer všetky knižnice sveta, s nepreberným množstvom dostupných prameňov. Môžem sa dívať na mapy ľubovoľnej krajiny doplnené obrazmi krajiny, miest, prírodných úkazov, cestovať internetom za poznaním doteraz neznámeho. Môžem sa oboznamovať so školskými systémami v zahraničí, komunikovať nielen s ich tvorcami, ale aj s ich obeťami. V oblasti kultúry sa sprístupňuje cesta k najlepším prácam tvorcov umeleckých diel najširšieho zamerania v rozsiahlej škále audiovizuálnej podoby, od obrazov až po operné, alebo iné bohaté scénické predstavenia. Kuriozitou je fakt, že pri cestovaní v internete možno aj zablúdiť. Treba sa teda naučiť správne sa v ňom orientovať.

Zaujímavý je aj prínos informačných technológií pre využitie voľného času. Mnoho firiem vo svete ponúka nielen počítačové hry, ale i prospekty na dovolenkové cesty, dodávané do počítačovej siete vo veľmi atraktívnych audiovizuálnych podobách. Atraktívny je i svet virtuálnej reality, ktorá je v tejto etape rozšíreným módnym hitom najmä u mladšej generácie v kategórii hier. (Treba však povedať, že technológia virtuálnej reality má mimoriadny dopad i v praktických použitíach napr. vo sfére medicíny, letectva a ďalších, kde sa významne uplatňuje v simulačných procesoch, trénažeroch, pri nácvikoch zručností a testovaní ľudí, materiálov, výrobkov i postupov.)

Už z tohoto stručného vymenovania je zrejmé, že informačné technológie zasahujú do všetkých sfér života spoločnosti, rodiny i jednotlivca. Vo všetkých vyspelých krajinách venujú problémom takejto informačnej spoločnosti veľa pozornosti. Napr. Európska únia sa v súčasnosti zameriava na riešenie týchto problémov:

- teleworking – viac a nových prác pre mobilnú spoločnosť,
- dištančné vzdelávanie – permanentné vzdelávanie v meniacej sa spoločnosti,
- spájanie vedeckých centier – prepojenie mozgových kapacít,
- telematické služby – podpora pokroku a zamestnanosti,
- manažment cestnej premávky – zvýšenie kvality,
- riadenie leteckej premávky – elektronizácia,
- zdravotnícka sieť – efektívnejšie zdravotníctvo,
- transeurópska verejná sieť – zefektívnenie verejnej administratívy,
- urbanistické informačné superdiaľnice.

Ako vidno, informatika má a stále bude mať mimoriadny význam pre rozvoj spoločnosti. Dopad informatiky ďaleko prevyšuje ekonomický rozvoj spoločnosti, a preto sa jej problematika stáva veľmi zaujímavou pre jednotlivé vlády. Existujú už štáty, ktoré majú zodpovedajúci rezort riadený samostatným ministerstvom pre informačné technológie.

Definíciu informačných technológií však možno chápať aj takto: informačné technológie predstavujú aplikáciu informatiky v konkrétnych úlohách bežnej praxe ale i vedy, ba dokonca niektorí autori kladú medzi aplikovanú informatiku a informačné technológie znamienko rovnosti [7]. Väčšina vedcov sa však zhoduje v tom, že *informačné technológie sú nástrojmi informatiky*, a preto sú jej súčasťou.

## 2. SIGNÁL

K tomu, aby človek získal nejakú informáciu o istej udalosti, netreba aby bol pri nej fyzicky prítomný. Ba dokonca môže o nej získať omnoho detailnejšiu informáciu ako tí, ktorí sú priami účastníci uvažovanej udalosti. Spomeňte si napr. na závod formuly 1: pri televízore poznáte omnoho viac súvislostí o stave závodu ako tí, ktorí sú priamo na tribúne pretekov. Ďalej je tiež zrejmé, že správy o nejakých javoch, príhodách a skutočnostiach dostávame často dávno potom, čo sa takéto príhody stali. Z toho plynie, že po niektorých udalostiach zostávajú isté stopy, ktoré existujú krátko po udalosti alebo i dlho po tom, čo samotná udalosť už zanikla.

Ako sme už povedali, *udalosťou rozumieme výskyt nejakého javu, procesu, úkazu alebo faktu, ktorý nás zaujíma, alebo je istým spôsobom využiteľný.*

**Udalosť** môže zanechať stopy v pamäti účastníka, môže sa uchovať v podobe písomného záznamu, obrázku, fotografie, fonozáznamu či videozáznamu. Väčšinou takáto podoba **opisu** (záznamu) nejakkej udalosti viac či menej „súhlasí“ so skutočnou udalosťou. Zmysel slov *opis súhlasí s udalosťou* intuitívne celkom dobre chápeme – ide o to, či je záznam udalosti *pravdivý*. V skutočnosti je otázka „súhlasu“ trochu zložitejšia: fotografia jazdca v cieľi pretekov predsa **nijak fyzicky nesúvisí** s jeho dojazdom do cieľa; je to len papier s nanesenou vrstvou striebra, ktoré vytvára napr. v odtieňoch šedej farby takú podobu rozloženia svetla, aká bola pri dopade na citlivý film v okamžiku záberu. Fotografia (ako spôsob záznamu udalosti) nesúhlasí s udalosťou úplne presne – napr. rozdelenie intenzity svetla je zachytené len približne, snímka nevyjadruje priestorovosť atď. Chceme tým povedať, že žiaden opis, snímka, obraz a pod., nevyjadruje sledovanú udalosť absolútne presne.

Zhrňme to všetko stručne: **snímka, písomný opis, ústny opis, film apod. , ako fyzikálne objekty nemajú nič spoločné s udalosťou, o ktorej sa prostredníctvom nich čosi dozvedáme.** Tieto záznamy majú celkom samostatnú fyzikálnu podstatu a samostatne existujú v priestore a čase. Skutočnosť, že snímka *súhlasí* s tým, čo zobrazuje, spočíva v tom, že za istých podmienok pôsobí na pozorovateľa rovnako, ako by na neho pôsobila samotná zobrazená situácia. Takto sme dopracovali k všeobecnému dôležitému pojmu **signál**.

**Signál vzniká pri určitej udalosti, činnosti, fakte alebo jave. Má samostatnú fyzikálnu podstatu a existenciu, nezávislú od podstaty udalosti samotnej [1].** Signál

existuje vždy v nejakej organizovanej sústave (systéme) a vždy sa viaže na určitý **hmotný objekt alebo proces**. Preto môže byť **zaznamenaný** (uchovateľný), existovať po dlhú dobu a možno ho prenášať na potrebné vzdialenosti z jedného miesta na druhé.

**Signál môže** vyvolať istý dej, t.j. **pôsobiť** na živého pozorovateľa alebo neživý stroj a vyvolať v nich určitú reakciu: tá môže byť taká istá, akú by vyvolala udalosť sama, keby pôsobila bezprostredne.

Zdôrazníme, že nezávislá existencia signálu umožňuje signálu byť neaktívnym obsahom pamäti po požadovanej dobe, a aktivovať ho možno práve vtedy, keď to treba. Rovnako možno signály kombinovať s inými signálmi a vytvárať tak signály nové, alebo možno signály transformovať do inej fyzikálnej podstaty a pod. Z toho tiež plynie, že samostatnosť existencie signálu treba chápať relatívne, t.j. tak, že **po** vzniku signálu už jeho existencia nezávisí len na udalosti, pri ktorej vznikol. Pojmy *signál* a *informácia* sú teda zviazané: informácia nemôže existovať inak, než vo forme signálu. Fyzický objekt, na ktorom možno pozorovať a vyhodnocovať signál, nazývame **informačné médium**.

Presnejšie povedané, nedá sa rozlíšiť, či istá udalosť pôsobí priamo, alebo je pôsobenie sprostredkované, lebo *každé* pôsobenie udalosti či faktu na pozorovateľa alebo stroj, sa deje *prostredníctvom* signálu.

Pôsobenie udalosti vo vzťahu k signálu je interaktívne: ak máme s potrebnou presnosťou spracovaný vhodný opis nejakej udalosti alebo objektu, môžeme podľa neho udalosť alebo objekt spätne zrekonštruovať.

Zhrňme tieto úvahy: **medzi udalosťou a signálom existuje za určitých okolností vzájomný vzťah a vzájomne jednoznačné priradenie** (v medziach, ktoré sú určené stupňom podrobnosti opisu). Signál je teda **izomorfným zobrazením** istých stránok nejakého faktu alebo udalosti.

Na povahe signálu nie je podstatná fyzikálna sústava v ktorej je signál realizovaný, ale skutočnosť, že **signál je množina rôznych stavov** tejto **fyzikálnej sústavy** alebo procesu. Množiny stavov rozličných objektov si môžu izomorfne navzájom zodpovedať a tak možno vhodným *meničom* preniesť signál z jedného objektu na druhý. Pritom sa dá meniť aj fyzikálna podstata, ktorá nesie signál. Zdôrazníme, že **fyzikálna forma ani energia signálu nemajú** pritom **rozhodujúci význam**. V spojitosti so signálom je podstatná skutočnosť, že je to proces, ktorý zodpovedá nejakej udalosti, alebo ktorému zodpovedá nejaká reakcia.

Je teda zrejmé, že **významový (sémantický) obsah pojmu informácia je totožný s izomorfným vzájomným priradením nejakej udalosti a signálu**. Ak existuje na jednej

strane určitý jav alebo udalosť a množinu stavov tohto javu a na druhej strane signál a množina jeho stavov alebo symbolov, môžeme prvky týchto množín navzájom priradiť. Tým určíme každému symbolu v signále určitý význam, t.j. konkrétny stav opisovaného javu, stav, ktorý bude tomuto symbolu zodpovedať. Vyslanie určitého signálu je vlastne voľba jedného symbolu alebo prvku z celej množiny stavov signálu, ktorému príslušný symbol zodpovedá.

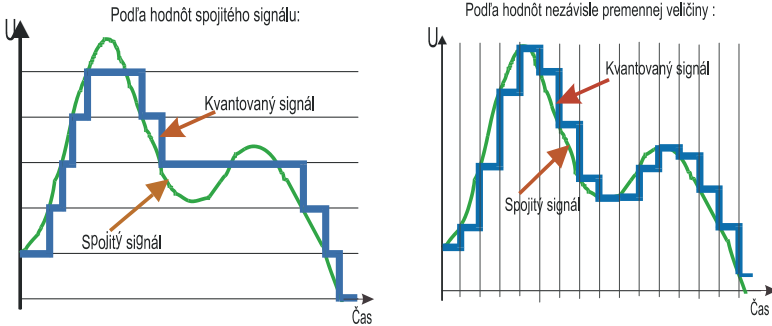
Z toho všetkého vyplýva, že **signál je nositeľom informácie, informácia bez signálu neexistuje**. Doplňme, že tá istá informácia môže byť prenášaná rôznymi signálmi a naopak, jeden signál môže prenášať rozličné informácie.

## **2.1 Spojité a nespojité signály, kvantovanie**

Signály ako fyzikálne veličiny môžu spojito nadobúdať všetky možné hodnoty (v medziach daných technológiou) – potom hovoríme o **spojitých signáloch**, alebo nadobúdajú len isté presne dané hodnoty, preto hovoríme o **nespojitéch** alebo **diskrétnych** signáloch. Spojito sa mení napr. množstvo vody v rieke, dráha letu muchy v priestore, zvuk huslí (a jemu zodpovedajúci elektrický prúd v mikrofone) a pod., naproti tomu počet ľudí v miestnosti, počet mincí v peňaženke alebo čísla s konečným počtom desiatinných miest sú signálmi diskrétnymi. Transformácia (prevod) spojitéch signálov na diskrétne (a naopak) je možná a v praxi častá. Zmeny stavov veličín môžu prebiehať spojito alebo skokovo a tomu musí zodpovedať aj signál. Spojité signály vedú k analógovému zobrazeniu informácie, a to, z hľadiska dnešného pohľadu na informatiku, nie je už dominantné.

Diskrétne veličiny sa na spojité menia pomerne ľahko, lebo diskrétna veličina sa môže prirodzene v istých bodoch zhodovať so spojitou. Naopak je to zložitejšie. Tomuto prevodu hovoríme **kvantovanie**. Kvantovanie sa robí tak, že spojitá fyzikálna veličina sa nahradí konečným počtom svojich hodnôt, ktoré potom reprezentujú veličiny diskrétne. Sú dva spôsoby kvantovania: **podľa hodnôt** spojitej veličiny a **podľa času** (všeobecnejšie: **podľa nezávisle premennej** veličiny). *Obr. 2.1* znázorňuje oba princípy. Je zrejmé, že pri kvantovaní spojitého signálu dochádza k prirodzeným odchýlkam hodnôt a teda k istým chybám v presnosti, čo sa považuje za prirodzený **šum**. Okrem neho však sa prejavuje aj iný veľmi nepriaznivý vplyv a to je šum z okolia. Takýto šum vytvárajú náhodné poruchy, ktoré sa interferenčne pripoja k signálu a nemožno ich jednoducho od signálu oddeliť. Šum teda ovplyvňuje kvalitu každého signálu a môže aj zhoršiť spoľahlivosť príslušného systému.

Bolo už povedané, že jednotkou miery množstva informácie je *bit* (**binary digit**). Ak teda informácia, viažúca sa na nejaký stav, určuje hodnotu takého bitu, totiž nulu (0) alebo jednotku (1), mus priradený signál tiež nadobúdať dve, jednoznačne rozlišiteľné hodnoty



Obr. 2. 1 Kvantovanie spojitkej veličiny

hodnoty fyzikálnej veličiny, ktorá tento signál realizuje. Existuje mnoho spôsobov ktorými možno realizovať takéto dva navzájom **rozlišiteľné** stabilné stavy, napríklad: *dierka a nedierka* v papierovej páske, relé s kontaktami v polohe *zopnuté-nezopnuté*, alebo magnetický prstenec s magnetickým tokom orientovaným *ľavotočivo* či *pravotočivo*. Tieto princípy sa v praxi skutočne využívali. Pre dnešnú rutinnú potrebu sa však historicky ustálila taká podoba signálu, ktorá sa viaže na elektrický prúd, elektrické napätie, prejavujúce sa v nejakom elektronickom (integrovanom) obvode najčastejšie tak, že jednotke (1) je priradené napätie  $U_1$  (spravidla 5 voltov) a nule (0) napätie  $U_0$  (spravidla 0 voltov). Signál, ktorý má takúto "dvojhodnotovú" podobu sa nazýva *elementárny signál*. Pretože 0 a 1 sú hodnotami dvojkových číslíc a sú tiež hodnotami premenných logických funkcií, nazývame tento signál tiež **číslicový signál** alebo **logický signál**. Potom prvky, v ktorých sa uplatňuje, označujeme ako **číslicové** alebo **logické prvky** či **obvody**. Na obr. 2.4 a 2.5 sú uvedené rozličné používané elementárne signály, ktoré sa v histórii i praxi skutočne používali či používajú.

Povedali sme tiež, že vyššou informačnou jednotkou je **bajt** (B), t.j. 8 bitov, ďalej **kilobajt** (KB) t.j.  $2^{10} = 1024$  bitov, **megabajt** (MB), t.j.  $2^{20}$  bitov atď., prípadne **blok** (spravidla 128 B) alebo **sektor** (spravidla 512 B). Univerzálnou pracovnou informačnou jednotkou vo vzťahu k signálu je **slovo**. Predstavuje usporiadanú štruktúru niekoľkých bitov (štandardom je napr. 16, 32, 64 bitov), alebo je tento pojem používaný aj celkom voľne vo vzťahu k počtu bitov.

Elementárny signál teda musí zobrazit' slová rôznej dĺžky. Ak pritom každému bitu takého slova je priradený samostatný elementárny signál (vedený samostatným drôtom), hovoríme o **paralelnom zobrazení informácie**. Naopak, ak všetkým bitom priradíme jediný

signál (tj. jediný drôt) tak, že hodnoty informácie v jednotlivých bitoch rozložíme do usporiadaného časového sledu, potom ide o **sériové zobrazenie informácie**.

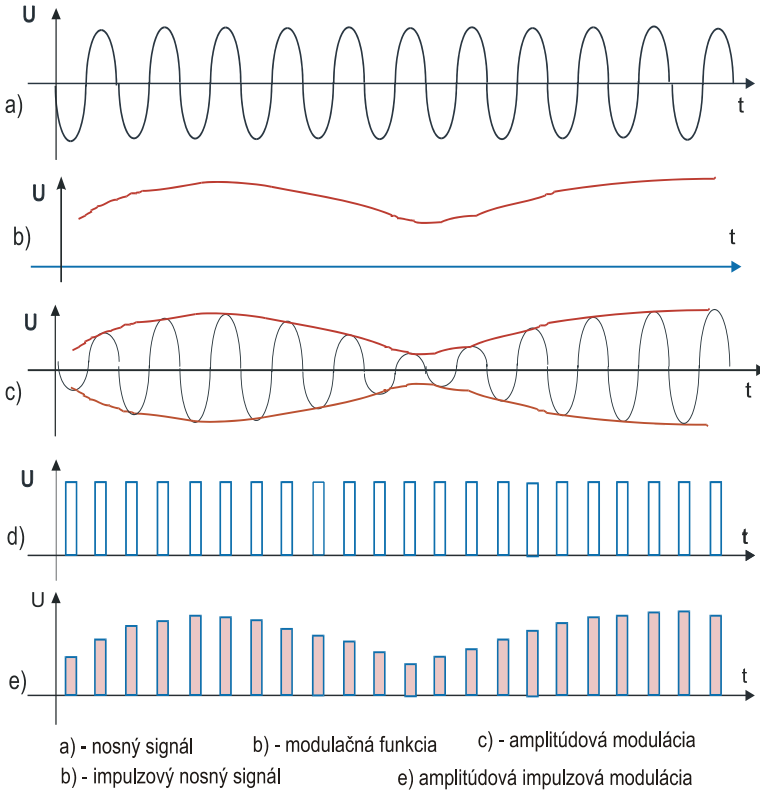
Z toho, čo sme tu naznačili, vyplýva, že *signál* ako nositeľ informácie je dôležitý najmä z hľadiska spracovania *informácie*. Kvalita signálu (najmä jeho frekvenčné spektrum) je určená vlastnosťami (elektronických) prvkov, v ktorých sa signál uplatňuje. Používaná technológia a úroveň výroby číslicových prvkov veľmi silno ovplyvňuje aj kvalitu, rozsah a rýchlosť samotného procesu spracovania informácie v strojoch, v ktorých sa takéto spracovanie realizuje. Problémy viažuce sa na technológiu výroby číslicových prvkov a zariadení sú často určujúce a sprevádzajú vývoj strojov na spracovanie informácie, t.j. počítačov a počítačových systémov, už od ich vzniku. Podstatne pritom ovplyvňujú aj rozvoj informatiky samotnej.

## **2.2 Informačný prenosový kanál**

Každé prostredie alebo zariadenie, ktoré sprostredkováva prenos signálu na určitú vzdialenosť sa nazýva **prenosový kanál**. Môže to byť napr. obyčajný drôt spájajúci dva blízko osadené tranzistory, na druhej strane to môže byť medzikontinentálny optický kábel, medziplanetárny priestor, ktorým sa šíri elektromagnetické vlnenie rádiového signálu z kozmickej rakety na zem, alebo aj medziplanetárny priestor, ktorým sa šíri svetelný lúč zo vzdialenej galaxie do nejakého astronomického observatória, apod. Z bežných prenosových kanálov možno uviesť telefónne vedenie, televízny systém alebo *spojenie počítačov v internete*.

Pozrime sa na informačný prenosový kanál z hľadiska jeho štruktúry. Je znázornená na obr. 2.7. Prenosový kanál preberá informáciu na svojom vstupe, tj. na *vysielacej strane* z nejakého **zdroja informácií** v podobe signálov. Tie prejdú kanálom na jeho koniec, k jeho výstupu, tj. k **prijímaču** informácií, kde ich preberá *adresát*. Ak množinu stavov signálov na vstupe kanála považujeme za určitú **abecedu**, potom abeceda na výstupe nemusí byť totožná s abecedou signálu na vstupe: vo všeobecnosti môže totiž prenosový kanál obsahovať určité kódovacie zariadenia (prevodníky, dekodéry), ktoré vyhovujú nosiču signálu na vstupe, ale nehodia sa napr. pre signál na výstupe kanála. Príkladom takéhoto usporiadania je napr. *sledovanie teploty* čidlom na vstupe kanála a jej *kódovanie* nejakým číselným kódom, potom na výstupe možno napr. kresliť graf priebehu sledovanej teploty po *dekódovaní* prenesených čísel na *polohu* kresliaceho pera na papieri v nejakej grafickej sústave.

Nebudeme sa podrobne zaoberať technikou prenosu signálu a fyzikálnymi záležitosťami, sprostredkujúcimi prenos informácie v priestore a čase, ale uvedieme tu len niektoré pojmy, ktoré z hľadiska informatiky považujeme za základné. Predovšetkým je to



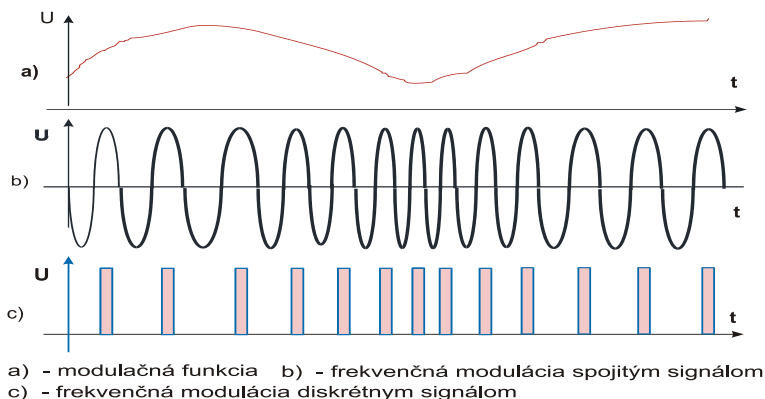
Obr. 2. 2 Amplitúdová modulácia

modulácia. Vo všeobecnosti pojmom **modulácia** označujeme určitú **riadenú zmenu stavu signálu**.

Bolo zmerané, že frekvencia ľudskej reči sa pohybuje medzi 60 Hz a 9 kHz, teda **frekvenčné spektrum** ľudskej reči obsahuje frekvencie ležiace medzi uvedenými medzami. Predstavme si prípad, že v tom istom čase prenáša zodpovedajúci signál prenosovým kanálom naraz viacero ľudí s tým istým frekvenčným spektrom. Dôsledok je veľmi zlý: signály od jednotlivých ľudí sa miešajú, (dochádza k tzv. *interferencii*) a prenesený signál na

strane príjemcu je takmer úplne nezrozumiteľný. Cieľ preniesť informáciu sa nedosiahol.

Jedna z ciest, ako zvýšiť spoľahlivosť prenosu a umožniť prenášať značný počet informácií (správ) súčasne, spočíva v tom, že každý individuálny signál zodpovedajúci



Obr. 2. 3 Frekvenčná modulácia

informácii od vysielajúceho až k adresátovi zviazať s nejakým ďalším, pomocným signálom tak, aby k spomínanému vzájomnému rušeniu nedochádzalo. To sa dosahuje moduláciou.

**Moduláciou** v tomto slova zmysle najčastejšie **rozumieme** vytvorenie určitého nosného signálu (nosnej vlny), ktorého frekvencia je pre každý individuálny zdroj informácie rôzna, pričom veľkosť (t.j. amplitudu) týchto nosných kmitov treba meniť tak, aby „kopirovala“ tvar a podobu signálu zodpovedajúceho prenášanej pôvodnej informácii. Takto sa interferencia pôvodných (na rovnakom pásme frekvencií informácií) znemožní tým, že sa zariadi, aby frekvenčný odstup jednotlivých nosných vln bol dostatočne veľký na to, aby sa navzájom neovplyvňovali. . Potom stačí na strane príjemcu urobiť inverzný proces, tj. demodulovať prenesenú „združenú dvojicu signálov“, odstrániť z nej pomocnú nosnú vlnu a využiť prenesenú správu podľa potrieb prijímateľa. Tak sa naplní princíp prenosu informácie prenosovým kanálom.

*Poznámka:* Zariadenie, ktoré realizuje moduláciu sa nazýva **modulátor** a zariadenie pre demoduláciu je **demodulátor**. V obojsmerných prenosových kanáloch sa takéto zariadenia realizujú ako integrovaný celok zvaný **modem** (modulátor – demodulátor)..

Spôsob takto opísanej modulácie je znázornený na obr. 2.2. Priebeh 2.2a predstavuje pomocný nosný signál, 2.2b je signál ktorý reprezentuje prenášanú informáciu (hovorí sa mu

aj **modulačná funkcia**), 2.2c je už modulovaná informácia postupujúca prenosovým signálom od zdroja k prijímaču. Je to typický prípad predstavujúci tzv. **amplitúdovú moduláciu (AM)** spojitým sínusovým signálom, ktorá je bežná v rádiovom vysielaní. Analogická je aj amplitúdová modulácia diskretným signálom podľa *obr. 2.2 d,e*, ktorá je častejšia pri číslicových signáloch.

V úvode sme už spomínali, že pri prechode kanálom pôsobia na signál rozličné rušivé vplyvy, ktoré spôsobujú vznik **šumu** alebo **poruchy**. Môžeme konštatovať, že každý signál sa objavuje spolu s šumom, ktorý vznikol už pri kvantovaní signálu alebo sa pridá v procese prenosu. Najslabší signál (z hľadiska prijatého výkonu), ktorý ešte možno rozoznať od šumu, sa nazýva **prahový signál**. Stane sa, že adresát na prijímacom konci kanálu prijme niečo, čo sa líši od vyslanej podoby. Mohli by sme dokázať, že ak považujeme v tomto prípade výstup kanálu za zdroj informácií, je entropia väčšia na výstupe ako na vstupe. „Množstvo“ informácie teda navonok vzrástlo, ale jej obsah sa v dôsledku šumu nezväčšil. Na vznik šumu má vplyv aj spôsob modulácie. Je známe, že amplitúdová modulácia je veľmi citlivá na šumové signály vznikajúce napr. pri búrkach (elektrické výboje v atmosfére), naproti tomu takéto javy takmer nemajú vplyv pri použití **frekvenčnej modulácie (FM)**. Jej príklad je na *obr. 2.3a*, ktorý predstavuje modulačnú funkciu (t.j. signál zodpovedajúci prenášanej informácii), *b* je pomocný, nosný signál (v zmysle vyššie uvedeného odseku), ktorý však modulačná funkcia nemení vo veľkosti amplitúdy, ale mení ho podľa veľkosti modulačnej funkcie v šírke periódy nosných kmitov, t.j. mení frekvenciu nosného signálu. *Obr. 2.3c* je analogiou frekvenčnej modulácie pre diskretný signál. Frekvenčná modulácia sa uplatňuje v prenosoch na VKV a v televíznych systémoch. Sú aj iné spôsoby modulácie, zmienime sa o nich ešte v časti o elementárnom (logickom) signále.

### **2.3 Prenosová kapacita kanálu**

Fyzikálne procesy prebiehajúce v prenosovom kanále sú isto podstatné, ale nemožno prehliadnuť najdôležitejší dôvod, pre ktorý sa prenosový systém vytvoril, a to je informácia, jej množstvo prenášané kanálom vôbec a najmä množstvo za jednotku času. Číselným hodnotením množstva informácie sa zaoberá vedný odbor **teória informácií**. Rieši problémy priepustnosti prenosových kanálov, problémy optimálneho kódovania informácií a zaoberá sa analýzou podmienok pre dosiahnutie maximálneho množstva informácií, ktoré môžu byť prenesené za jednotku času. Takéto množstvo však nemôže byť väčšie než je **priepustnosť** kanálu alebo inak **prenosová kapacita kanálu**.

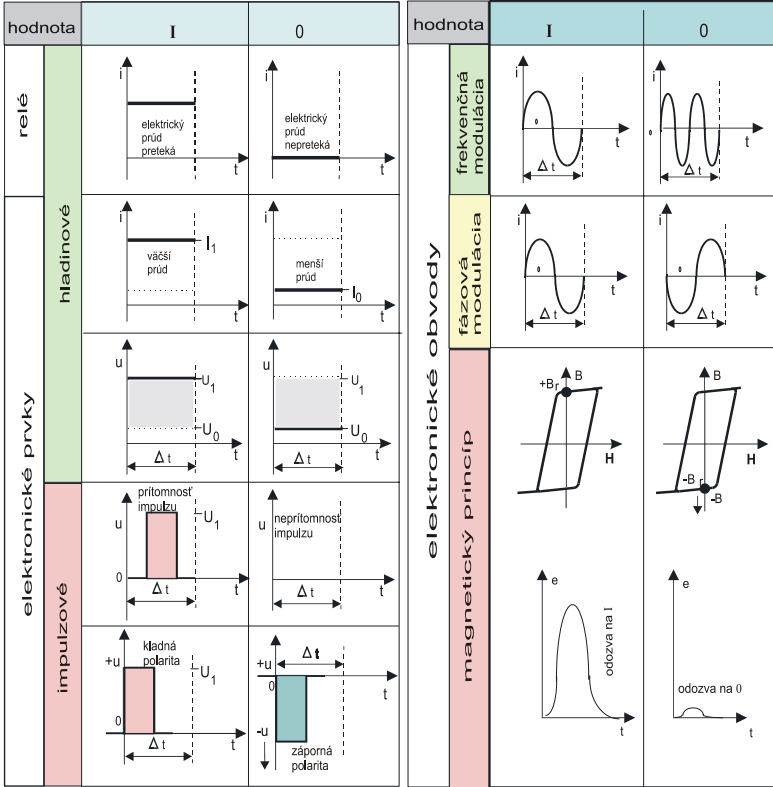
Ak si odmyslíme konkrétny obsah udalosti, javu, objektu, o ktorom sa prenášajú alebo zaznamenávajú informácie, zostane nám vždy len **istá množina navzájom odlišných stavov**. Tieto stavy nikdy nedokážeme opísať (informáciou) ako jeden celok, lebo každý objekt môže byť v každom okamžiku iba **v jednom zo všetkých možných stavov**. (Množina stavov môže byť spojitá alebo diskretná, ale môže byť tiež tvorená jedinou veličinou alebo súborom mnoho veličín). Z toho plynie, že vlastný opis udalosti je v podstate len údaj, v ktorom z tých možných stavov opisovaný objekt je.

O informácii sme už povedali, že predstavuje výber niektorého prvku z množiny možných prvkov. Vzhľadom na *izomorfizmus medzi informáciou a signálom* potom vydanie akejkoľvek informácie (z množiny jej možných stavov) sa realizuje ako fyzikálny výber určitého stavu z množiny stavov signálu. Používa sa v tejto súvislosti takáto terminológia: množina všetkých stavov signálov sa nazýva **abecedou signálu**, jeden prvok z abecedy signálu označujeme ako **symbol** alebo **písmeno** z danej abecedy. Akúkoľvek kombináciu symbolov či písmen nazývame **slovo** (pozri analógiu v 1.kapitole). Príkladom najjednoduchšej abecedy je abeceda obsahujúca iba dva symboly:  **$\theta$**  a  **$I$** . Slovo v tejto abecede možno zapísať napr. takto :  *$\theta I \theta \theta I \theta I \theta$* .

Na *obr. 2.4 a 2.5* sú príklady signálovej podoby symbolov  **$\theta$**  a  **$I$** , ktoré sú používané v minulosti i dnes pre prácu s informáciami v číslicovej technike či logických systémoch. *Obr. 2.6 a,b* je príkladom vyjadrenia sedembitového slova **statickou** a **dynamickou** podobou dvojhodnotového signálu vyjadrujúceho dané slovo.

*Obr.2.6c* sa viaže na problematiku šumu. Definuje napät'ovú hladinu  $U_I$  priradenú hodnote  **$I$**  a napät'ovú hladinu  $U_\theta$  priradenú hodnote  **$\theta$** . Tieto signálové hladiny nemožno vo všeobecnosti lacno udržať na ideálnych hodnotách, preto patria do tzv. **pásma jednotkovej resp. nulovej úrovně**. Pokiaľ sa istý uvažovaný signál nachádza vo vnútri pásma, je šum v kanále takpovediac „neškodný“. Tak, ako nemožno udržať ideálne pásma pre jednotkový a nulový signál, nemožno stanoviť ani presnú hodnotu napätia, ktorá obe pásma oddeľuje: preto medzi obe uvažované pásma treba vložiť tzv. **zakázané pásmo**. Ak niektorý zo signálov nadobudne hodnotu napätia zo zakázaného pásma, nemožno o príslušnom signále rozhodnúť, či zobrazuje nulu alebo jednotku. *V tomto prípade šum znehodnocuje informáciu*. (Ak však zariadíme, aby sa napr. v čase prechodu z jednej hladiny na druhú informácia nemohla považovať za platnú, potom takéto šum opäť nemusí škodiť. To sa často používa v tzv. synchronných (riadených) systémoch).

Ak poznáme abecedu signálu a dobu trvania prenosu (čo v prenesenom zmysle slova znamená vlastne *dĺžku slov*) môžeme ľahko vypočítať počet možných správ, z ktorých robíme výber pri každom prenose. Už v r. 1928 navrhol Hartley [1] hodnotiť informačnú



$\Delta t$  - doba jedného bitu

Obr. 2. 4 Elementárne signály hladinové

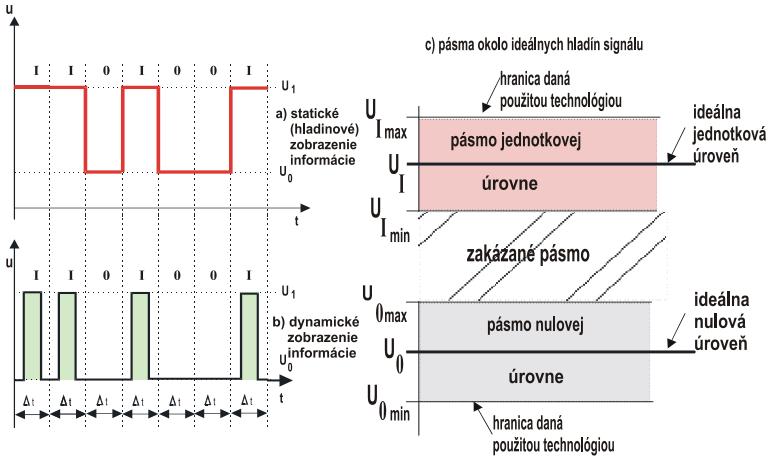
Obr.2.5 Elementárne signály dynamické

kapacitu systému logaritmom počtu možných stavov tohoto systému. Ak má systém  $N$  rôznych navzájom nezávislých stavov, je jeho informačná kapacita daná vzťahom

$$C_I = \log N \quad (2.1)$$

Ak spojíme dva takéto systémy, budú mať zrejme  $N^2$  možných stavov, lebo každému prvku prvého systému možno priradiť  $N$  stavov druhého systému. Informačná kapacita takéhoto spojenia dvoch systémov bude

$$C_2 = \log N^2 = 2 \cdot \log N = 2 \cdot C_1 \quad (2.2)$$



Obr. 2.6 Pásma signálu

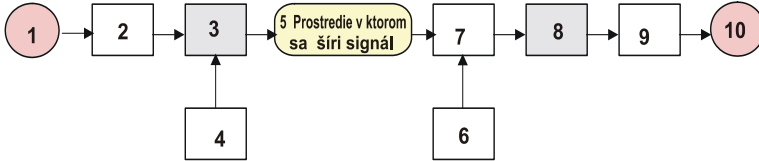
Z toho možno odvodiť, že spojením  $k$  systémov dosiahneme informačnú kapacitu  $k$ -krát väčšiu, z čoho plynie, že informačná kapacita má aditívnu vlastnosť. Táto linearita zjednodušuje výpočty a odôvodňuje použitie logaritmickej miery kapacity. Treba však vhodne vybrať základ logaritmu. Tu nám pomôžu vyššie spomínané úvahy o šume: prax ukázala, že sa pomerne ľahko realizujú prvky zodpovedajúce dvom spoľahlivo odlišným stavom (s možnými širokými dovolenými pásmami dvoch napätových hladín) a takéto dva stavy takých prvkov sú dostatočne stabilné. To viedlo k tomu, že sa základ logaritmu vo vzťahoch (2.1 a 2.2) berie číslo 2, ide teda o dvojkový logaritmus.

**Příklad 2.1.** Uvažujme, že prenosový kanál má preniesť 100 rôznych symbolov signálovej abecedy, teda  $N = 100$ . Ak túto hodnotu dosadíme do vzťahu (2.1), dostávame pri základe logaritmu 10 informačnú kapacitu  $C = 2$ . Výsledok je správny,

lebo ak označíme jednotlivé symboly signálovej abecedy číslami 00, 01, ..., 99, skutočne v desiatkovej sústave na to stačia dve desiatkové číslice.

**Príklad 2.2.** Predpokladajme, že prenosový kanál má preniesť 64 rôznych symbolov signálovej abecedy. Pri základe logaritmu rovnom 2 podľa vzťahu (2.2) je potom informačná kapacita uvažovaného kanálu

$$C = \log_2 64 = 6, \text{ lebo } 2^6 = 64.$$



- |   |                            |
|---|----------------------------|
| 1 - zdroj informácií                          | 6 - zdroj šumu             |
| 2 - kódovacie zariadenie, snímač a vysielateľ | 7 - prijímač               |
| 3 - modulátor                                 | 8 - demodulátor            |
| 4 - generátor nosnej frekvencie               | 9 - dekódovacie zariadenie |
| 5 - prostredie, ktorým sa signál šíri         | 10 - výkonný orgán         |

Obr. 2.7 Prenosový kanál

Inými slovami môžeme povedať aj to, že v tomto prípade stačí použiť šesť bitov na to, aby sme spoľahlivo uvažovaných 64 symbolov preniesli informačným kanálom.

**Príklad 2.3.** Uvažujme, že treba preniesť 1030 symbolov. Zo vzťahu (2.2) vyplýva aj toto: vzhľadom na to, že  $2^{10} = 1024$ , (čo je menej ako 1030), musí byť informačná kapacita potrebného kanálu rovná 11, lebo  $2^{11} = 2048$ , čo je síce o 1018 stavov viac a tieto nadbytočné stavy nebudú využitú. Takýto prípad je častý a je spravidla vyvolaný požiadavkami na vysokú spoľahlivosť pri prenose informácie.

Ukázali sme, ako sa exaktne vyjadruje priepustnosť či prenosová kapacita kanálu. Jednoznačne ju určujú vzťahy (2.1) resp. (2.2). Tieto vzťahy však nevyjadrujú časové a frekvenčné súvislosti prenosu informácií. Slovo „priepustnosť“ zväzda položiť aj otázky:

- prenesie kanál ľubovoľný počet správ ?
- naozaj prenosový kanál prenesie všetky informácie bez ohľadu na fyzikálne vlastnosti modulovaného signálu?
- aká je prenosová rýchlosť kanálu?

Na prvú otázku je odpoveď ľahká: ak je dostatok času, prenesie aj kanál s nízkou priepustnosťou dostatočne veľký počet správ radených postupne za sebou.

Druhá otázka je ťažšia, ale dá sa aj na ňu dostatočne stručne odpovedať. Nie každé prostredie, v ktorom sa signál šíri (*obr.2.7*), obsiahne všetky frekvencie, s ktorými je signál spojený. Je ľahké pochopiť, že **počet nosných signálov** použitých v súvislosti s moduláciou signálu **nie je nekonečný**. Je ohraničený predovšetkým tým, aký je rozsah frekvencií, ktoré cez nosné prostredie signálu prejdú bez straty kvality. Ak nosné prostredie má **medznú frekvenciu** napr. *1000 KHz* (kilohertz), potom pri odstupe frekvencií nosných signálov *100 KHz* sa v rozsahu *100 KHz až 1000 KHz* spracuje **desať nosných vln súčasne** (tj. *100, 200, 300, 400, 500, 600, 700, 800, 900 a 1000 KHz*). Inak povedané, môže súčasne prebiehať prenos desiatich nezávislých správ („hovorov“) tým istým prenosovým kanálom. Prenos sa teda spája s pojmom **frekvenčné spektrum kanála**, a to vyjadruje rozsah použiteľných frekvencií, ktoré kanál spoľahlivo prenesie. Frekvenčné spektrum je charakteristickou veličinou materiálu, ktorým sa signál šíri. V súčasnosti je najvhodnejším prostredím na šírenie signálu **sklené vlákno**, ktorého spektrum dosahuje tisícky megahertzov, takže súčasne môže prebiehať prenos až niekoľko desiatok tisíc nezávislých správ prostredníctvom optického signálu.

To, čo tu spomíname, súvisí s neutíchajúcou tendenciou zvyšovať rýchlosť procesu spracovania informácií, ich prenosy nevynímajúc: kto má rýchlejší počítač či číslicový systém, je úspešnejší aj na trhu s takouto technikou. Zostáva teda odpovedať na poslednú z položených otázok: Vo všeobecnosti je **prenosová rýchlosť kanálu určená počtom bitov, ktoré prejdú kanálom za jednotku času**, jej rozmer je teda **bit/s**, alebo vyššie odvodené jednotky množstva informácie za jednotku času.

### 3. SYSTÉM

Postúpme k ďalším základným pojmom týmito myšlienkami: žiadna informácia o nejakom objekte, jave či udalosti nie je celkom úplná a nevystihuje všetky vlastnosti, ktoré sa naň viažu. Objekty nemožno študovať v celej ich rozmanitosti. Signál zachytí len tie stránky javu, ktoré sú preň podstatné. Vo všeobecnosti totiž nemôžeme definovať všetky stavy, ktoré tento objekt nadobúda (alebo stavy, ktoré ho bezo zvyšku charakterizujú), lebo ich môže byť napríklad nespočítateľný počet. Z toho vyplýva, že na nejakom objekte (jave, udalosti, ktorá nás zaujíma) treba definovať *system*. Systém je taká podoba opisu nejakého objektu nášho záujmu, ktorá s postačujúcou (alebo dohodnutou) presnosťou charakterizuje ten objekt (t.j. jeho stavy, jeho správanie sa, jeho využiteľnosť). Možno tiež povedať, že *system* predstavuje istý *model* originálu, pomocou ktorého možno originál exaktne či štatisticky skúmať, riadiť alebo inak s ním zaobchádzať. Inými slovami, *system* je nástrojom poznávania a ovládania jemu zodpovedajúcej reality.

Každý systém má svoje **okolie**. Vzhľadom na to, že sa obmedzujeme len na konečný počet determinovaných vzťahov medzi systémom a okolím, hovoríme o **podstatnom okolí**. Systém a okolie sú vo vzájomnej interakcii. Systémy, pri ktorých uvažujeme všetky možné vplyvy okolia na systém a systému na okolie sú **otvorené**. Systémy tohoto druhu sú často nazývané aj **objekty**. Také systémy, pri ktorých neberieme na zreteľ interakciu s okolím nazývame **absolútne uzatvorené**.

Také systémy, pri ktorých nie sú presne stanovené cesty, ktorými okolie pôsobí na systém a podobne nie sú stanovené ani cesty ktorými systém pôsobí na okolie nazývame **relatívne uzatvorené**. Takéto systémy si tu všimneme bližšie : budeme uvažovať, že sú uzatvorené z hľadiska energetickej výmeny alebo látkovej výmeny, ale sú **otvorené z hľadiska informačnej výmeny**.

Dôležitými pojmami, ktoré sa tu používajú sú **vstupy a výstupy systému**. Informácia z okolia , ktorú považujeme za **podnet**, vstupuje v podobe signálu cez **vstupné uzly** (kanály) do systému, a v ňom sa určitým spôsobom uplatní: hovoríme, že systém *reaguje* na vstupný podnet. Bez toho, že by sme to teraz bližšie špecifikovali, hovoríme, že na vstupný podnet vo všeobecnosti **zmení systém svoj vnútorný stav**.

Podobne, vplyvy systému na okolie nazveme **odozvami**. Odozvy sa prejavja prostredníctvom **výstupných uzlov** (kanálov) systému. Ak teda hovoríme o systéme otvorenom z hľadiska informačnej výmeny, potom na výstupných uzloch sa objaví informácia o zmene vnútorného stavu systému, vyvolanej vstupným podnetom. Súboru všetkých vstupných, resp. výstupných uzlov môžeme jednoducho hovoriť **vstup** resp. **výstup**. V niektorých prípadoch - ak to nebude na škodu zrozumiteľnosti - použijeme aj označenie pre dielčí vstup napr. *vstup č. 3*, *r-tý vstup*, *m výstupov* a pod.

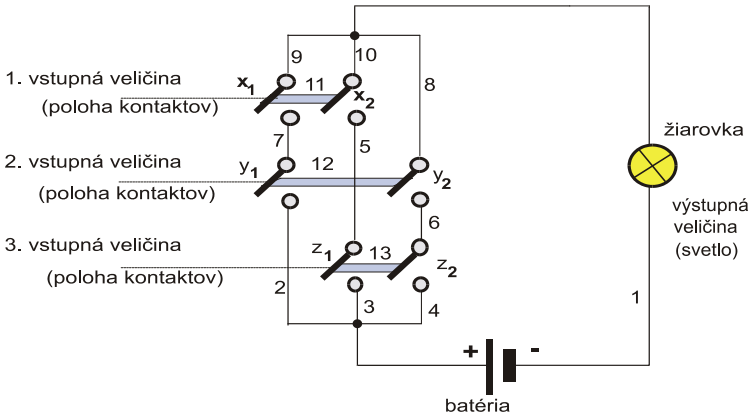
### 3.1 Vlastnosti systému

Každý systém možno charakterizovať dvoma základnými vlastnosťami:

1. **správaním sa systému a**
2. **štruktúrou systému.**

Správanie sa systému sa definuje ako **závislosť medzi odozvami a podnetmi**. Štruktúrou systému rozumieme **vnútornú organizáciu systému**, (t.j. usporiadanie vzájomných väzieb medzi prvkami systému aj väzieb na vstup či výstup) a tiež správanie sa týchto prvkov.

Uveďme teraz príklad jednoduchého systému: Na *obr. 3.1* je zobrazený systém, ktorý stráži tri fyzikálne veličiny nejakého technologického procesu, napr. tlaku, teploty a vlhkosti. Úlohou systému je zapnúť žiarovku (ako znamenie poplachu) vtedy, keď aspoň dve zo sledovaných veličín prekročia určené kritické hranice. Možno teda povedať: žiarovka svieti práve vtedy, keď je súčasne prekročená kritická hodnota tlaku a teploty, alebo tlaku a vlhkosti, alebo teploty a vlhkosti, alebo tlaku, teploty a vlhkosti. Okolím tohoto systému je sledovaný technologický proces, (stroje, zariadenia, obsluha) a jeho konkrétne vymedzenie nazývame **podstatné okolie**. Jeho súčasťou sú snímače, sledujúce hodnoty tlaku, teploty a vlhkosti tak, že ak dôjde k prekročeniu kritickej hodnoty veličiny, spoja sa príslušné kontakty v systéme, inak zostanú rozpojené. Vidno, že ak sa zopne  $x_1$ , zopne sa súčasne aj  $x_2$  a to platí aj pre ostatné kontakty. **Polohu** každej dvojice kontaktov chápeme ako veličinu, ktorá môže nadobúdať dve navzájom odlišné hodnoty, budeme považovať za **vstupnú veličinu**. Žiarovku budeme považovať za výstupný prvok, a za dôležité považujeme len to, či svieti alebo nie. Tak opäť dostávame dvojhodnotovú **výstupnú veličinu**.



Obr. 3. 1 Systém s kontaktami

Systém podľa uvedeného obrázku obsahuje ešte ďalší prvok: **batériu**. O nej v tejto chvíli povedzme len to, že jej parametre stačia na to, aby činnosť systému bola správna. Zhráme to: opísaný systém obsahuje tri vstupné veličiny

- polohu kontaktov  $x_1, x_2$
- polohu kontaktov  $y_1, y_2$
- polohu kontaktov  $z_1, z_2$ .

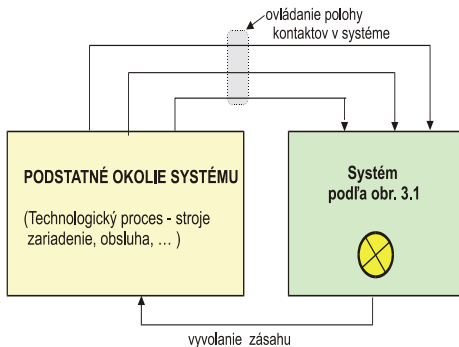
jednu veličinu výstupnú, stav žiarovky, t.j. či svieti alebo nie, a ešte celkovo osem prvkov

- kontakty  $x_1, x_2, y_1, y_2, z_1, z_2$ ,
- žiarovku a
- batériu.

Okrem toho sú v systéme pozorovateľné isté pevne dané väzby medzi jeho prvkami. Napr. záporný pól batérie je zviazaný s jednou svorkou žiarovky (označme to ako väzba č. 1), kladný pól batérie je zviazaný s nepohyblivou časťou kontaktu  $y_1$  (väzba č. 2), pohyblivá časť kontaktu  $x_2$  je zviazaná s druhou svorkou žiarovky (väzba č. 10), atď. Je tu celkovo 10 väzieb elektrických (galvanických) a tri väzby mechanické (č. 11, 12, 13). Všetky väzby vyjadrujú vzájomný vzťah medzi prvkami systému a aj ich usporiadanie. Môžeme povedať, že vyjadrujú **štruktúru** daného **systému**. Pozrime sa teraz na jeho správanie sa.

Všimnime si najprv vzťah systému k jeho okoliu. Obr.3.2 naznačuje, že z technologického procesu, t.j. z podstatného okolia prichádzajú na vstup podnety, ktoré môžu zmeniť polohu kontaktov systému: ak sú splnené podmienky pre rozsvietenie žiarovky,

Obr. 3. 2 Okolie systému



systém na podnety reaguje tak, že žiarovku rozsvieti, a tým vyvolá spätný zásah obsluhy do technologického procesu. Takto sa teda prejaví vplyv systému na svoje okolie.

Ak chceme systematicky opísať správanie sa nášho systému, musíme určiť isté hľadisko či hĺbku rozlišovacej úrovne, na ktorej nás systém zaujíma. *Priestorovú rozlišovaciu úroveň* môžeme v tomto prípade chápať tak, že sme určili veličiny, ktoré môžu nadobúdať iba dve hodnoty (kontakt zopnutý - rozopnutý, žiarovka svieti – nesvieti). Tak isto musíme rozhodnúť o *časovej rozlišovacej úrovni*. Spravidla sa v takýchto úlohách uvažujú iba tie časové okamžiky, v ktorých sú veličiny, ktoré nás zaujímajú, už ustálené. Nevšímajme si teda to, ako sa správa systém v dobe, keď sa mení poloha kontaktov a ďalej uvažujme, že sa pôsobením vstupného podnetu nemení štruktúra systému a že na rovnaký podnet prichádzajúci v rozličných časových okamžikoch reaguje systém tiež rovnako. (Takto definujeme pojem istej *stacionarity* systému). Za týchto predpokladov potom môžeme zistiť aká je odozva systému (t.j. či žiarovka svieti či nesvieti) pre ľubovoľnú kombináciu vstupných veličín.

Urobme takéto priradenie : kontakty  $x_1$  a  $x_2$  sú zopnuté vtedy a len vtedy, keď je prekročená kritická hodnota tlaku, a analogicky to urobme pre teplotu a vlhkosť. Potom môžeme jednoznačne zapísať všetky možné prípady polôh kontaktov a ku každému prípadu z obr.3.1 vyjadriť stav žiarovky. Dostaneme tak *Tab .3.1*, ktorá **definuje závislosť medzi vstupmi a výstupmi** a hovoríme, že **opisuje správanie sa systému**.

Vráťme sa ešte k pojmu *rozlišovacia úroveň*. Na systém v príklade sme pozerali z istého, nami nejako definovaného hľadiska, neuvažovali sme napr. konkrétne vlastnosti batérie (napr. napätie, kapacita), ani kontaktov (spínacia doba, zaťažiteľnosť), ani žiarovky (výkon, farba svetla), neuvažovali sme ani veľkosť celkovej potrebnej energie, nezohľadnili sme ani do akých klimatických podmienok sa systém hodí, nezaujímal nás vzhľad, atď. Ale my sami sme sa rozhodli zanedbať isté okolnosti, isto so systémom súvisiace, my sme si

Poradové číslo	Kontakty $x_1, x_2$ zopnuté	Kontakty $y_1, y_2$ zopnuté	Kontakty $z_1, z_2$ zopnuté	Žiarovka svieti
1	nie	nie	nie	nie
2	nie	nie	áno	nie
3	nie	áno	nie	nie
4	nie	áno	áno	áno
5	áno	nie	nie	nie
6	áno	nie	áno	áno
7	áno	áno	nie	áno
8	áno	áno	áno	áno

Tab. 3.1 Tabuľka správania sa systému

zvolili náš pohľad, našu **rozišovací úroveň na ktorej chceme systém pozorovať alebo skúmať**. Je to v praxi bežné, že volíme takú rozišovací úroveň, ktorá je optimálna pre konkrétny typ úlohy. Niektoré systémy skúmame detailnejšie, iné menej podrobne. Je zrejme, že aj na jeden a ten istý systém sa môžeme pozerat' s rôznou hĺbkou úrovne. *Zvolená rozišovacia úroveň zároveň určí aj to, pre ktoré prístupy je daný systém otvorený či uzavretý.*

### 3.2 Definícia systému

Na základe uvedených myšlienok pristúpme teraz k formálnej definícii systému [ 5,6 ] :

Nech systém  $\mathcal{S}$  obsahuje prvky  $a_1, a_2, \dots, a_n$ , a nech  $a_0$  je okolie tohoto systému. Nech množina  $\mathcal{A} = \{ a_1, a_2, \dots, a_n \}$  a množina  $\mathcal{B} = \{ a_0, a_1, a_2, \dots, a_n \}$ , t.j. množinu  $\mathcal{A}$  tvoria len prvky systému  $\mathcal{S}$ , a množinu  $\mathcal{B}$  tvoria nielen tieto prvky, ale i okolie systému, ktoré sa považuje za osobitný prvok  $a_0$ .

Nech každý prvok množiny  $\mathcal{B}$  je charakterizovaný množinou vstupných veličín a množinou výstupných veličín. Označme symbolom  $r_{ij}$  závislosti vstupných veličín prvku  $a_j$  na výstupných veličinách prvku  $a_i$ , ktoré sú dané vzťahmi medzi týmito veličinami. Množinu všetkých závislostí  $r_{ij}$  ( $i, j = 0, 1, \dots, n$ ) označme  $\mathcal{R}$ . Potom systém definujeme takto:

$$\text{Systémom je každá usporiadaná dvojica } \mathcal{S} = \{ \mathcal{A}, \mathcal{R} \}. \quad (3.1)$$

Ako vyplýva z definície, systémom nazývame súbor určitých prvkov, medzi ktorými existujú isté vzťahy a to aj vzťahy systému k jeho okoliu. Množinu  $A$  nazývame **univerzum** systému a množinu  $R$  **charakteristikou** systému.

### Správanie sa systému

Formulujme teraz presnejšie pojem *správania sa systému*. Už sme naznačili, že tento pojem predstavujú závislosti medzi vstupmi a výstupmi systému. Predpokladajme teda, že každý relatívne uzavretý systém má **vstup** a **výstup**. Ako sme už povedali vyššie, *vstupom* (resp. *výstupom*) rozumieme množinu dielčích vstupov (*výstupov*), t.j. vstupných (resp. *výstupných*) **uzlov** alebo **bodov**. Najjednoduchšie je uvažovať, že každým uzlom „prechádza“ iba jedna veličina a okamžitú hodnotu veličiny na vstupnom (výstupnom) uzle budeme nazývať čiastkovým podnetom (čiastkovou odozvou) a označíme ju takto:

$x_1, x_2, \dots, x_n$  sú čiastkové podnety,  $n$  je počet vstupných uzlov systému,  
 $y_1, y_2, \dots, y_m$  sú čiastkové odozvy,  $m$  je počet výstupných uzlov systému.

Čiastkové podnety možno považovať za zložky vektoru v  $n$ -rozmernom priestore, t.j.

$\mathbf{x} = (x_1, x_2, \dots, x_n)$  a nazveme ho *vstupným vektorom*. Podobne zavedieme aj *výstupný vektor*  $\mathbf{y} = (y_1, y_2, \dots, y_m)$  v  $m$ -rozmernom priestore. Potom **správanie sa systému definujeme ako transformáciu  $T$**  vektoru  $\mathbf{x}$  do vektoru  $\mathbf{y}$ .

$$\mathbf{y} = T(\mathbf{x}). \quad (3.2)$$

kde  $T$  je operátor transformácie.

V tých prípadoch, keď každému podnetu (t.j. každému **stavu vstupného vektoru**) zodpovedá vždy *len jedna* odozva, (t.j. jeden **stav výstupného vektoru**), hovoríme o jednoznačnej transformácii. Slovo *stav* tu zámerne zdôrazňujeme, lebo je spojovníkom k už spomínaným pojmom vstupná, výstupná abeceda, symbol ako prvok abecedy apod., (pozri kapitolu o signále). Môžeme teda písať:

$$\begin{aligned} y_1 &= f_1(x_1, x_2, \dots, x_n) \\ y_2 &= f_2(x_1, x_2, \dots, x_n) \\ &\vdots \\ &\vdots \\ y_m &= f_m(x_1, x_2, \dots, x_n) \end{aligned} \quad (3.3)$$

kde  $f_1, f_2, \dots, f_m$  sú funkcie  $n$  nezávisle premenných  $x_1, x_2, \dots, x_n$ .

Vzťahy (3.3) vyjadrujú tzv. kombinačné správanie sa systému, a takýto systém nazývame **kombinačný systém**. Taký je každý systém, v ktorom výstupné hodnoty sú jednoznačne určené iba okamžitými vstupnými hodnotami.

V takých prípadoch, keď jednému stavu vstupného vektora zodpovedá viac než jeden stav výstupného vektora (viacero odoziev) hovoríme o **viacznačnej transformácii**. Ak je pritom možno presne stanoviť všetky odozvy zodpovedajúce všetkým postupnostiam podnetov, hovoríme o sekvenčnom správaní sa a systémy tohoto typu nazývame **sekvenčné systémy**.

Z hľadiska vzťahu k používateľovi (alebo pozorovateľovi) je každý systém charakteristický svojou štruktúrou a správaním sa. Štruktúra je daná prvkami z ktorých sa skladá a jeho správanie možno chápať ako realizáciu funkcie, ktorá určuje hodnoty odozvy na výstupe systému (do okolia), v závislosti od hodnôt podnetu, prichádzajúceho z okolia na vstup systému. Táto funkcia určuje aj väzby medzi prvkami systému a je *funkciou času*. Určuje, ako sa v časovej postupnosti mení *stav systému* a *stav na výstupe* v závislosti na *vstupnom stave*. V systéme teda vo všeobecnosti prebieha **proces**, ktorý reprezentuje správanie sa systému.

Kombinačné a sekvenčné správanie sa nazývame *determinovaným správaním* resp. **determinovaným systémom**.

Ak transformáciu (3.2) možno určiť len štatisticky, potom správanie sa systému označujeme ako náhodné. Takéto systémy nebudú predmetom nášho záujmu.

## Hierarchia systémov

Definícia systému hovorí, že systém je usporiadaná dvojica prvkov a vzťahov medzi nimi. Čo je to však prvok? Pojmom prvok systému označujeme istý relatívne uzatvorený systém nižšieho rádu vzhľadom na uvažovaný systém, **prvok je teda istý podsystém**. Každý prvok  $a_i$  systému  $S$  považujeme z hľadiska príslušnej uvažovanej rozlišovacej úrovne za nedeliteľný celok, ktorého štruktúru považujeme za známu a nechceme alebo nepotrebujeme ju nejako rozoberať. Tým sme sa dostali k pojmu **rád systému**. Je to určitý ukazovateľ, vzťahujúci sa na určitú referenčnú (porovnávaciu) úroveň. Pritom pojmu *systém* zodpovedá pohľad na jednej konkrétnej rozlišovacej úrovni. Pojmu **nadsystém** zodpovedá pohľad,

v ktorom nie sú uvažované tie rozlišovacie úrovne, ktoré nie sú pozorovateľovi dostupné, alebo, na ktorých ho daný systém nezaujíma. Takto je teda definovaná **hierarchia systémov**. Systém sa definuje vždy na nejakej potrebnej alebo dohodnutej úrovni detailizácie, v ktorej môžeme odhaliť hierarchiu systémov, a v nej potom hovoríme o podsystémoch alebo nadsystémoch vzhľadom na nejaký systém.

Podľa charakteru prvkov a zložitosti väzieb v systémoch hovoríme o systémoch *energetických, dopravných, sociálnych, biologických, pedagogických, ekonomických, matematických*, a mnohých ďalších, ktoré sa v spoločnosti objavujú. Veľmi zaujímavé sú systémy, ktoré využívajú prvky pracujúce na báze číslcového signálu a uplatňujú sa aj v riadiacich a rozhodovacích procesoch človeka či stroja, preto ich často označujeme ako *logické* alebo *číslcové systémy, niekedy aj kybernetické systémy*.

Systémy, ktorých vnútorné prvky a ich vzájomné vzťahy majú informačný charakter (sú teda previazané informačnými väzbami) a vzťahy k okoliu sa tiež viažu na spracovanie informácie, sú *informatické* alebo *informačné systémy*. Takéto systémy sú dnes predmetom rozsiahleho záujmu spoločnosti, a preto dokonca hovoríme, že už žijeme v informačnej spoločnosti. Vznikli napríklad podľa rezortov alebo istých združení, a tak existuje informačný systém školstva, bankový informačný systém, mestský informačný systém, geografický informačný systém a mnohé ďalšie. Chápeme ich ako *špecializované informačné systémy*.

## Základné úlohy

Existujú tri základné úlohy v spojitosti so systémom. Sú to

- analýza systému
- syntéza systému a
- problém čiernej skrinky.

Analýza systému je chápaná takto: Systém je zadaný tým, že poznáme jeho štruktúru, ale nepoznáme jeho správanie sa. Úlohou je teda pre zadanú štruktúru nájsť opis jeho správania sa a to v nejakej vhodnej, napr. tabuľkovej podobe. Pri syntéze je to naopak: je zadané požadované správanie sa, a treba nájsť, t.j. zostrojiť, vytvoriť štruktúru, ktorá bude požadované správanie realizovať. Pre obe úlohy bol vypracovaný celý rad metód, ktoré sa v praxi úspešne uplatňujú.

Problém čiernej skrinky je takýto: máme systém, o ktorom nevieme ani ako sa správa, a ani jeho štruktúru nemožno priamo zistiť. Ak ho chceme poznať, získať vedomosti o jeho

správaní sa a o jeho štruktúre, treba použiť postup, ktorému hovoríme **metóda pokusov a omylov**. Spočíva v tom, že so systémom experimentujeme, sledujeme odozvy na zvolené podnety a po istom počte pokusov sa nám môže podariť niečo o systéme odhaliť do tej miery, že to bude postačujúce pre naše potreby.

### 3.3 Model

V celom rade praktických úloh v rôznych odboroch ľudskej činnosti pracujeme s pojmom *model, modelovanie*. V čom sú korene týchto slov? Človek už dávno spozoroval významné podobnosti medzi niektorými dvojicami systémov a tak v niektorých prípadoch môže považovať jeden zo systémov z daného hľadiska za model druhého a pomocou systému modelujúceho skúmať vlastnosti systému modelovaného. Pojem podobnosti je chápaný širšie, (napr. ide o podobnosť tvarovú, podobnosť v zložitosti apod.) , ale slovo *model* má užší a presne stanovený význam. Modelovaním rozumieme len určitý, determinovaný druh podobnosti medzi systémami.

Systém sme charakterizovali štruktúrou a správaním sa. Analogicky hovoríme aj o *modeloch štruktúry a modeloch správania sa*.

Ak vykazuje systém  $S_2$  rovnaké správanie sa ako systém  $S_1$ , hovoríme, že systém  $S_2$  je **modelom správania sa** systému  $S_1$ . Toto tvrdenie platí zrejme aj naopak a pritom môžu byť štruktúry modelovaného systému  $S_1$  (*originálu*) i modelu  $S_2$  celkom odlišné.

Ak hovoríme o modeloch štruktúry, máme na mysli určité podobnosti medzi štruktúrami systému modelovaného i modelujúceho. Z podobnosti štruktúr vyplývajú tiež podobnosti v správaní sa, a preto každý model štruktúry je zároveň modelom správania sa. Takýto model nazývame jednoducho **model systému**.

Ak modelujeme štruktúru určitého systému  $S_1$  systémom  $S_2$  , **musíme zachovať izomorfný vzťah**, ktorý je daný takto:

1. každému prvku  $a_i$  systému  $S_1$  je priradený jediný prvok  $a_i'$  systému  $S_2$  a naopak,
2. každý prvok  $a_i'$  systému  $S_2$  je modelom správania sa zodpovedajúceho prvku  $a_i$  systému  $S_1$  a naopak,

3. každej väzbe medzi dvoma prvkami  $a_i$ ,  $a_j$  systému  $S_1$  je priradená rovnaká väzba medzi zodpovedajúcou dvojicou prvkov  $a_i'$ ,  $a_j'$  systému  $S_2$  a naopak.

Treba sa teraz zmeniť aj o **abstraktnom modeli**. Ak chceme poznať nejaký systém, zaujímame sa predovšetkým o jeho správanie sa pričom za známe považujeme len také správanie, ktoré je jednoznačne nejakým opísané. Podobne, pri vysvetlení toho, akým procesom v systéme dochádza k určitému správaniu sa, opäť žiadame opis príslušného procesu. Tieto myšlienky sa viažu aj na štruktúru s tým, že správanie sa možno vysvetliť na základe štruktúry. Ak je teda systém nejakým vhodným jazykom (spôsobom) opísaný, hľadáme v opise súvislosti o vzťahoch v systéme a tieto súvislosti sa potom snažíme vyjadriť pomocou slov, obrázkov, schém, matematickej symboliky, a pod. Prídeme zrejme k záveru, že pod pojmom **opis** sa vlastne rozumie vytvorenie rovnakých súvislostí, aké sa objavujú v reálnom systéme (origináli), lenže sú vyjadrené inak, napr. systémom rovníc, systémom výrokov, systémom obrázkov, grafov, iných predstáv a pod. V zhode s definíciou modelu potom takýto nový systém nazveme **abstraktným modelom systému**.

Z toho, čo sme doteraz povedali možno usúdiť, že ak systém  $S_2$  modeluje systém  $S_1$  potom vstupný podnet  $x_1$  systému  $S_1$  môže byť iný (má napríklad inú fyzikálnu podstatu) ako vstupný podnet  $x_2$  systému  $S_2$ . Z toho dôvodu treba realizovať určité **zobrazenie podnetov modelovaného systému** na podnety systému modelujúceho. Podobne a z rovnakých dôvodov to treba urobiť aj na strane výstupov, kde hovoríme o **zobrazení odoziev modelujúceho systému** na odozvy systému modelovaného. Zobrazenie na vstupe systému  $S_2$  nazveme **vstupným zobrazením** a podobne na výstupe hovoríme o **výstupnom zobrazení**. Situáciu znázorňuje obr. 3.3.

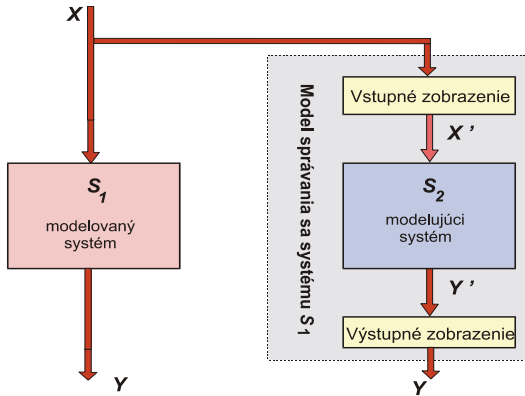
Niekedy sa spôsob akým sa zobrazenie robí vyjadruje formou rovníc, ktoré potom nazývame **vstupné**, resp **výstupné zobrazovacie rovnice**. To umožňuje nakoniec uviesť nasledujúce formálne definície modelu správania sa a modelu štruktúry.

1. *Modelom správania sa* systému  $S_1$  (originálu) nazveme každý taký systém  $S_2$  s príslušným vstupným a výstupným zobrazením, v ktorom všetky vstupné podnety systému  $S_1$  sa po vykonaní vstupného zobrazenia transformujú, a po vykonaní výstupného zobrazenia prejavia tak, že odozvy sú rovnaké ako odozvy originálu  $S_1$ .

2. Modelom systému  $S_1 = \{A_1, R_1\}$  je izomorfný systém  $S_2 = \{A_2, R_2\}$  s takým zobrazením  $R_1$  na  $R_2$ , že na rovnaký podnet okolia na obidva systémy vykazujú sebe zodpovedajúce si prvky v  $A_1$  a  $A_2$  rovnaké odozvy.

Zo všetkých prostriedkov, ktorými možno systémy modelovať, majú základný význam dva druhy: technické a abstraktné:

**Technické prostriedky modelovania** sú predovšetkým rozličné fyzikálne, napr. elektrické alebo elektronické systémy a to buď jednoduché (určené na riešenie špeciálnych prípadov), alebo zložitejšie, napr. číslicové počítače. Práve modelovaní pomocou počítačov (počítačové modelovanie) má dnes mimoriadnu dôležitosť.



Obr. 3.3 Model správanía sa systémú

Veľmi významné sú však aj **abstraktné prostriedky modelovania**. Uplatňujú sa vo všetkých úlohách spojených so systémami: v analýze, syntéze, ale i pri riešení problémov čiernej skrinky.

Za abstraktný model možno považovať **slovný opis systému**, ktorý vo vhodnom jazyku vyjadruje, aké sú odozvy na výstupoch pri uvažovaných vstupných podnetoch, ďalej, aké sú prvky systémov a ich správanie sa, aké sú vzťahy medzi prvkami systému a vzťahy systému k okoliu. Iným typom abstraktných modelov sú **rozličné tabuľky**, ktoré často zlepšujú znížení presnosť slovného opisu (pozri napr. Tab 3.1).

Pre sekvenčné systémy (pri ktorých je dôležité vyjadriť zmeny stavov systému v čase a v závislosti na podnetoch z okolia) je veľmi názorným **abstraktným modelom graf správanía sa** (pozri ďalej).

Ďalším typom abstraktného modelu je **algebraický model** a za takýto model považujeme rozličné sústavy algebraických, diferenciálnych, integrálnych a iných rovníc.

Najrozšírenejším a veľmi prehľadným a názorným abstraktným modelom (modelom štruktúry) je symbolická bloková schéma. Každý prvok je zobrazený dohovoreným symbolom (štvorček, koliesko, alebo iná dohovorená ( dokonca aj normovaná) značka, čím sa znázorní a lebo naznačí správanie sa tohoto prvku a nakreslením spojnic medzi symbolmi sa znázornia (napr. čiarami) všetky aktívne alebo aj pasívne väzby medzi prvkami systému. Príkladom takéhoto systému je napr. elektrická schéma televízoru, alebo schéma plynového rozvodu v byte apod. Modely tohoto druhu sa skutočne vyznačujú veľkou názornosťou, a sú preto omnoho častejšie používané ako modely algebraické alebo maticové.

### 3. 4 Logické systémy

Už sme naznačili, že predmetom nášho záujmu sú predovšetkým systémy, otvorené z hľadiska informačnej výmeny. Tieto systémy sa okrem iného vyznačujú tým, že ich štruktúry obsahujú *logické obvody*. **Logický obvod** je taký fyzikálny determinovaný systém, v ktorom každá veličina na vstupe aj výstupe nadobúda v ustálenom stave (s predpísanou presnosťou) iba jednu z dvoch možných hodnôt, a ktorý obsahuje také prvky, ktorých vstupné a výstupné veličiny nadobúdajú tiež iba jednu z dvoch možných hodnôt.

Príkladom takéhoto obvodu je systém na obr. 3.1, v ktorom skutočne všetky veličiny sú dvojhodnotové. Definícia vychádza z toho, že každú dvojhodnotovú veličinu možno vyjadriť jednoduchým výrokom, ktorý je základným pojmom logiky. Výroku sa priradzuje pravdivostná hodnota ( $0$  či  $1$ ), alebo mu možno priradiť určitú zodpovedajúcu logickú premennú  $x$ , ktorá nadobúda logickú hodnotu z množiny  $\{0, 1\}$ , t.j.  $x \in \{0, 1\}$ .

#### Kombinačný systém

Zostaňme ešte pri obr. 3.1. Správanie sa bolo vyjadrené takto: „svetlo svieti“ práve vtedy, keď

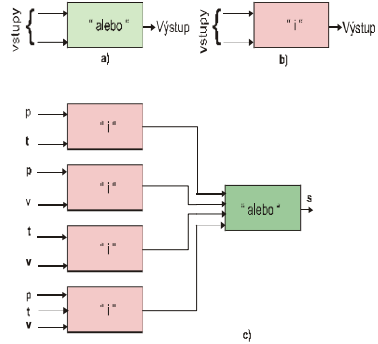
1. „je súčasne prekročená kritická hodnota tlaku i teploty,“ alebo
2. „je súčasne prekročená kritická hodnota tlaku i vlhkosti,“ alebo
3. „je súčasne prekročená kritická hodnota teploty i vlhkosti,“ alebo
4. „sú súčasne prekročené kritické hodnoty tlaku, teploty i vlhkosti.“

Vety v uvodzovkách sú jednoduché výroky, ktorým priradíme pravdivostné hodnoty, alebo rovno vstupné logické premenné (napr.)  $p, t, v \in \{0, 1\}$  a výstupnú logickú

premennú  $s \in \{0, 1\}$ . Potom správanie sa uvažovaného systému je dané Tab. 3.2, ktorá je inou podobou Tab. 3.1, a to tzv. *pravdivostnou tabuľkou*, vyjadrujúcou logickú funkciu  $f$  viažúcu sa na uvažovaný systém. Premenné  $p, t, v$  sú nezávislé logické premenné,  $s$  je závislá logická premenná (funkčná hodnota) danej logickej funkcie  $f$ . Platí

$$s = f(p, t, v) \quad (3.4)$$

a tento vzťah je analogickým vyjadrením



Obr. 3.4 Logická sieť

vzťahu (3.3), ktorý – vzhľadom na to, že výstup systému je jednoznačne určený vstupom - opisuje **kombinačné správanie sa** daného systému. Vo všeobecnosti možno povedať, že **logický systém realizuje (modeluje) logické funkcie**. Každý systém vyhovujúci formálnej podobe vzťahu (3.4) nazývame **kombinačný logický systém**.

Predpokladajme, že máme k dispozícii prvky systému také, že istý prvok realizuje výrok „*súčasne platí*“ (tzv. prvok „*i*“) a ďalej tiež prvok, ktorý realizuje výrok „*alebo*“ podľa obr. 3.4 a,b. Potom je ľahké zostrojiť štruktúru na obr. 3.4 c, ktorá predstavuje tzv. **logickú štruktúru systému** z obr.3.1. Tento obrázok predstavuje istú podobu dohovorených značiek použitých prvkov v systéme, spojených

Vstupy			Výstup
p	t	v	s
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Tab. 3.2 Log. funkcia vyjadrujúca

spojnicami, ktoré vyjadrujú reálne väzby použitých

správanie sa systému

prvkov, a preto je táto podoba zároveň *abstraktným modelom* skutočného fyzikálneho realizovaného logického obvodu z obr. 3.1. **Abstraktné modely logických obvodov** ( napr. v obr. 3.4 ) nazývame **logická sieť**.

Kombinačný logický systém ( obvod ) vo všeobecnosti realizuje viacero logických funkcií ( a nie iba jednu ako v obr. 3.4 c). Preto jeho značka má podobu podľa obr.3.5 a a svojou podobou viac pripomína zápis vzťahov (3.3).

Pozrime sa ešte na logický systém - ktorého správanie sa určuje logická funkcia definovaná tabuľkou Tab.3.2 - z hľadiska vstupnej a výstupnej abecedy. Ľahko sa

presvedčíme, že vstupnú abecedu  $X$  tvorí množina ôsmich symbolov, t.j. vstupných stavov  $X_1, X_2, \dots$  až  $X_8$ , teda  $X = \{ X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8 \}$  a sú to tieto stavy logických premenných  $p, t, v$ :  $X_1 = 0 0 0$ ,  $X_2 = 0 0 I$ ,  $X_3 = 0 I 0$ ,  $X_4 = 0 I I$ ,  $X_5 = I 0 0$ ,  $X_6 = I 0 I$ ,  $X_7 = I I 0$ ,  $X_8 = I I I$ .

Výstupné stavy sú dva, teda výstupná abeceda obsahuje dva symboly:  $Y_1$  a  $Y_2$ , takže  $Y = \{ Y_1, Y_2 \}$ , a sú to tieto stavy logickej premennej  $s$ :  $Y_1 = 0$ ,  $Y_2 = I$ . Môžeme písať napr. vektorové vyjadrenie kombinačného správania sa daného logického systému takto:

$$Y = F(X), \quad \text{čo je len iná podoba vzťahu (3.4), } F \text{ je logická funkcia.}$$

### Sekvenčný systém

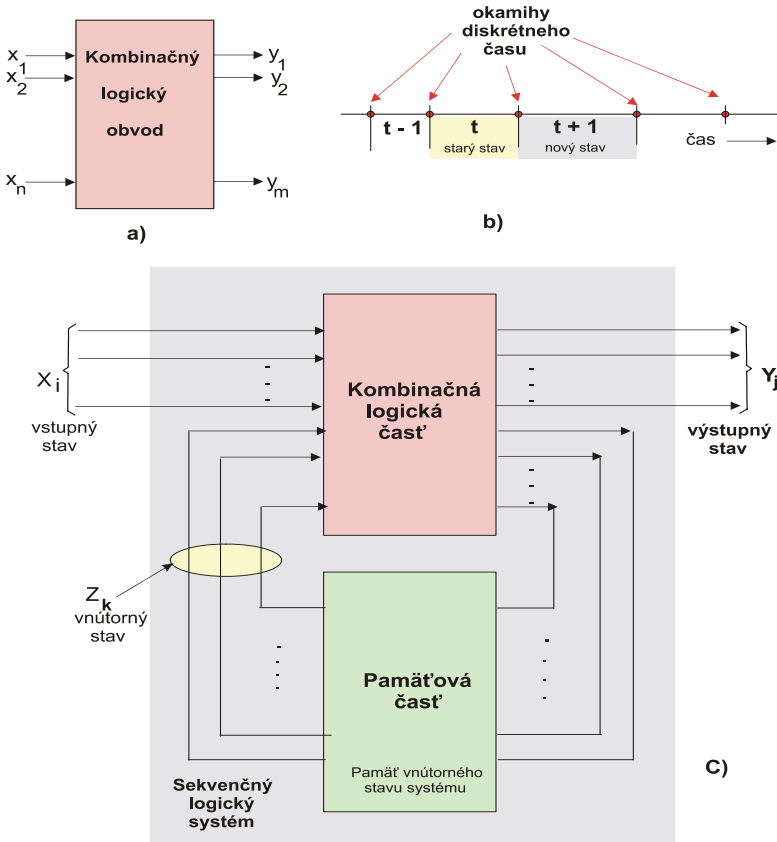
Sekvenčný logický systém je výborným prostriedkom na objasnenie transformácie podľa vzťahu (3.2) pre sekvenčné správanie sa systému. Povedali sme už, že množina všetkých podnetov tvorí *vstupnú abecedu*, a množina všetkých odoziev tvorí *výstupnú abecedu*. **Symbol** zo vstupnej abecedy predstavuje jeden zo stavov na vstupe systému, podobne, jeden z výstupných stavov je symbolom z výstupnej abecedy. (Slovo *symbol* sa niekedy označuje aj ako *písmeno* z danej abecedy). Podnet na vstupe spôsobí predovšetkým zmenu vnútorného stavu systému a tým aj zmenu na jeho výstupe. Preto sa definuje aj **množina vnútorných stavov systému**, pričom množina všetkých vnútorných stavov sa nazýva **vnútorná abeceda** systému. Dôležité je uvedomiť si, že na vstupe systému *nemôže byť v tom istom okamžiku viac než jeden vstupný symbol*, a platí to aj pre výstup. Rovnako nemôže v tom istom čase vnútorný stav systému predstavovať viac než jeden symbol z vnútornej abecedy.

Pretože v systéme prebieha proces zmien jednotlivých stavov, treba zohľadniť aj časové hľadisko. Z toho dôvodu zavedieme pojem **diskrétny čas**. To je čas, ktorý nevnímame bežne ako spojitú veličinu, ale len v istých určených okamihoch. Taký bod na časovej osi v ktorom dochádza k zmene stavu sledovanej veličiny nazveme **okamih diskrétno času** (obr. 3. 5 b).

Doba medzi dvoma okamihmi diskrétno času predstavuje ustálený stav veličiny, ktorú v tom intervale považujeme za nemennú. Interval naľavo od okamihu diskrétno času predstavuje minulý, **starý stav** veličiny, interval napravo od okamihu diskrétno času považujeme za budúci, **nový stav** sledovanej veličiny a zapisujeme  $Z_{i,t}$  resp.  $Z_{i,t+1}$ , kde  $Z_i$  je

napr. niektorý symbol z abecedy vnútorných stavov systému a  $t$  predstavuje interval starého stavu,  $t+1$  predstavuje interval viažúci sa na nový stav.

Predpokladajme, že vstupná abeceda sekvenčného systému je  $X = \{X_1, X_2, \dots, X_n\}$ , výstupná abeceda  $Y = \{Y_1, Y_2, \dots, Y_m\}$ , a nech abeceda vnútorných stavov (ktoré sekvenčný logický systém dosahuje v okamihoch diskrétného času) je  $Z = \{Z_1, Z_2, \dots, Z_r\}$ . Správanie sa sekvenčného obvodu, najmä spomínaná nejednoznačnosť transformácie  $T$  vo vzťahu (3.2), umožňuje takýto výklad: Na jeden a ten istý podnet na vstupe systému (vstupný symbol) nemožno očakávať jednu a tú istú odozvu (výstupný symbol) na výstupe,



Obr. 3. 5 Kombinačný a sekvenčný logický systém

lebo tá je určená nielen vstupným stavom, ale aj okamžitým vnútorným stavom systému. Rovnako platí, že *nový* vnútorný stav systému do ktorého systém prejde v okamžiku diskrétného času, nie je určený len okamžitým vstupným stavom, ale aj okamžitým vnútorným (starým) stavom, v ktorom sa systém nachádza. Z toho plynie, že **sekvenčný systém musí mať vo svojej štruktúre pamäť, t.j. pamäťové prvky**, ktoré uchovávajú vždy jeden symbol z abecedy vnútorných stavov. Na *obr. 3.5 c* je načrtnutá štruktúra sekvenčného logického obvodu, ktorá sa skladá z kombinačnej a pamäťovej časti. Formálny zápis predchádzajúcich slov vyjadruje vzťah (3.5), určujúci  $i$ -tý symbol výstupnej abecedy  $Y_i$  v závislosti od vstupného symbolu  $X_j$  a okamžitého vnútorného stavu  $Z_k$

$$Y_i = F(X_j, Z_k), \quad (3.5)$$

kde  $F$  je logická funkcia, ktorá tú závislosť definuje a hovoríme jej **výstupná funkcia**.

Podobne možno vyjadriť závislosť (3.6) nového vnútorného stavu  $Z_{k,t+1}$  od okamžitého vstupu  $X_j$  v čase  $t$  a od starého stavu  $Z_k$  v čase  $t$

$$Z_{k,t+1} = G(X_{j,t}, Z_{k,t}), \quad (3.6)$$

kde  $G$  je tzv. **prechodová funkcia** (vyjadruje prechod zo stavu do stavu) a je to logická funkcia.

Výstupná a prechodová funkcia odstraňujú spomínanú nejednoznačnosť transformácie  $T$  v (3.2) a presne definujú správanie sa sekvenčného systému.

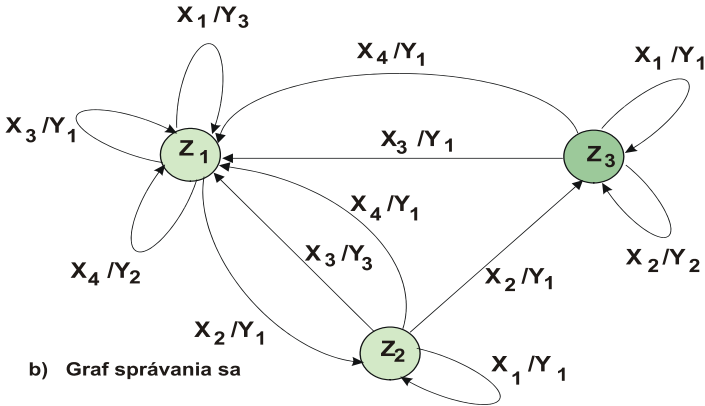
*Poznámka.* Uvedomme si, že symboly  $X_j, Y_i, Z_k$  nie sú logické premenné, reprezentujú však príslušné stavy logických premenných (napr.  $x, y, z, \dots \in \{0, 1\}$ ), ktorými sa  $X_j, Y_i, Z_k$  musia kódovať, aby bol logický systém fyzikálne realizovateľný.

### **Príklad 3.1**

Na *obr. 3.6 a, b* sú znázornené dva spôsoby opisu správania sa sekvenčného logického systému. Prvý z nich je tabuľkový, pričom prvá časť tabuľky sa viaže na určenie nového stavu v závislosti od okamžitého vstupu a okamžitého (starého) stavu. Ak použijeme symboliku podľa vyššie uvedeného označenia, nový stav je zapísaný vždy na priesečníku príslušného riadku a stĺpca pre všetky definované prechody (ale nie všetky možné musia byť vždy určené). Táto časť tabuľky teda určuje *prechodovú* funkciu  $G$  vo vzťahu (3.6) Druhá časť tabuľky sa formálne zhoduje s prvou, vyjadruje však už *výstupnú* funkciu  $F$  vo vzťahu (3.5). Preto sa obe časti často kreslia v jedinej spoločnej tabuľke a do príslušného políčka sa píše napr. dvojica  $X_i/Y_j$ .

Starý stav	Okamžitý vstup	Okamžitý vstup	
	Nový stav	$X_1X_2X_3X_4$	$X_1X_2X_3X_4$
$Z_1$	$Z_k, t+1$	$Z_1Z_2Z_1Z_1$	$Y_3Y_1Y_1Y_2$
$Z_2$		$Z_2Z_3Z_1Z_1$	$Y_1Y_1Y_3Y_1$
$Z_3$		$Z_3Z_3Z_1Z_1$	$Y_1Y_2Y_1Y_1$

a) Tabuľka správania sa



Obr. 3.6 Správanie sa sekvenčného systému

Obr. 3.6 b je oveľa názornejší spôsob opisu správania sa sekvenčného systému. Je to **orientovaný ohodnotený multigraf** (pozri kapitolu Grafy), ktorého **vrcholy (uzly)** sú ohodnotené symbolmi vnútorných stavov a orientované **hrany** (spojnice vrcholov) sú ohodnotené dvojicou  $X_i/Y_j$  čo môžeme interpretovať napr. pre hranu vedúcu z vrcholu  $Z_2$  do vrcholu  $Z_3$  takto: systém prejde zo starého vnútorného stavu  $Z_2$  do nového stavu  $Z_3$  pôsobením symbolu  $X_2$  na vstupe, a pritom sa na výstupe objaví výstupný symbol  $Y_1$ . Podobne to možno interpretovať pre všetky prechody v zhode s tabuľkou v obr. 3.6 a.

Oba spôsoby sú rovnocenné, no pre veľký počet stavov sa môže graf stať menej prehľadným. Tabuľka je najmä nástrojom syntézy systému, lebo umožňuje analogicky zapísať aj podobu po kódovaní symbolov všetkých abecied a tak umožní aj presný zápis, potrebný pre tvorbu kombinačnej časti sekvenčného systému.

## 4 KÓDY

### 4.1 Kódovanie desiatkových číslic

Aby bol styk počítača s človekom pružnejší, počítače sa konštruujú tak, že človek zadáva údaje priamo v desiatkovej sústave, aj keď vnútorné usporiadanie je v podstate dvojkové. Desiatkové číslice sa preto kódujú.

**Kód je zobrazenie (predpis), ktoré symbolom jednej abecedy priraduje symboly inej abecedy.** V desiatkovej sústave sa symboly z abecedy  $\{0, 1, 2, \dots, 9\}$  zobrazujú číslicami z abecedy  $\{0, 1\}$ . Na vyjadrenie desiatich číslic potrebujeme pre kódový obraz najmenej štvoricu bitov, tj. najmenej štvormiestne dvojkové číslo (pozri definíciu informačnej kapacity). Príklady takýchto kódov sú v *Tab. 4. 1*.

Skôr než prejdeme k jednotlivým kódom, uveďme najprv určité teoretické poznatky, ktoré formuloval profesor Harvardskej univerzity Aiken, vyjadrujúc tak isté požiadavky, ktoré by kódy pre vyjadrenie desiatkových číslic mali spĺňať. Hovoríme im **Aikenove podmienky pre kódy**.

**1. Podmienka jednoznačnosti:** Odlišným dekadickým čísliciam musia zodpovedať odlišné dvojkové čísla. Táto podmienka je celkom samozrejmá: nemôžno predsa kódovať jednu a tú istú desiatkovú číslicu dvoma rôznymi dvojkovými číslami, alebo dve rôzne desiatkové číslice tým istým dvojkovým obrazom – to by kódovanie stratilo zmysel.

**2. Podmienka usporiadosti:** Desiatkovej číslici s vyššou hodnotou musí zodpovedať kód, ktorý tiež predstavuje vyššie dvojkové číslo. Aj táto podmienka má prirodzené opodstatnenie – slúži na usporiadúvanie desiatkových čísel podľa veľkosti, čo sa v praxi často vyskytuje v celom rade bežných úloh. Je len logické, že táto podmienka môže byť interpretovaná aj naopak.

**3. Podmienka párnosti – nepárnosti:** párnym desiatkovým čísliciam majú zodpovedať párne dvojkové obrazy, respektíve naopak. V dvojkovom čísle sa párnosť či nepárnosť prejavuje hodnotou 0 resp. 1 v najnižšom bite. Opodstatnenosť tejto podmienky kódovania nie je priamo viditeľná, ale má zmysel pri operáciách zaokrúhľovania čísel a pri delení.

#### **4. Podmienka deviatkového doplnku:**

Nech sú dané dve desiatkové číslice *ALFA* a *BETA* také, že

$$ALFA + BETA = 9, \quad x_i \in \{0, 1\}.$$

Ak číslici *ALFA* zodpovedá štvoricu bitov  $(x_1 x_2 x_3 x_4)$ , potom číslici *BETA* musí zodpovedať štvoricu  $(\bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4)$ .

Táto podmienka sa označuje aj ako **pravidlo inverzie**: zo zapísaného totiž vyplýva, že *ALFA* dostaneme z *BETA* jednoduchou inverziou bitov v *BETA*, resp. naopak. To veľmi urýchľuje operácie so zápornými číslami, v ktorých sa odčítanie realizuje pričítaním doplnkov.

**5. Podmienka váhovania:** Takáto podmienka je dôležitá pri „odkódovaní“ čísel. Musia existovať také konštantné celé čísla  $W_1, W_2, W_3, W_4$ , ktoré obvykle nazývame **váhy**, že ak niektorej desiatkovej číslici *ALFA* zodpovedá štvorica bitov  $(x_1 x_2 x_3 x_4)$ ,  $x_i \in \{0, 1\}$ , potom musí platiť

$$ALFA = x_1.W_1 + x_2.W_2 + x_3.W_3 + x_4.W_4 .$$

Dá sa ukázať, že nie každý kód vyhovuje všetkým uvedeným podmienkam, ale len niektorým z nich. Napríklad kódy, ktoré vyhovujú podmienke váhovania, nazývame *váhové kódy*, iné kódy vyhovujúce napr. aritmetickým operáciám označujeme ako *kódy pre aritmetiku*, ďalšie kódy sú určené *pre spoľahlivý prenos informácie*, iné sú *kódy pre prevod analogových veličín na čísla alebo naopak*, atď.

Desiatková číslica <i>N</i>	Kód BCD Váhy 8421	Rubinoffov kód 84-2-1	Kód $3n+2$	Kód $8421+3n+3$	Kód 5421 pre násobičky	Aikenov kód 2421
0	0000	0000	00010	0011	0000	0000
1	0001	0111	00101	0100	0001	0001
2	0010	0110	01000	0101	0010	0010
3	0011	0101	01011	0110	0011	0011
4	0100	0100	01110	0111	0100	0100
5	0101	1011	10001	1000	1000	1011
6	0110	1010	10100	1001	1001	1100
7	0111	1001	10111	1010	1010	1101
8	1000	1000	11010	1011	1011	1110
9	1001	1111	11101	1100	1100	1111
<i>N</i>	Bikvinárny kód 5043210	Kód dva z päť (2 z 5)	Johnsonov kód	Grayov kód	Kód jeden z desať (1 z 10)	
0	0100001	11000	00000	0000	000000001	
1	0100010	00011	00001	0001	000000010	
2	0100100	00101	00011	0011	000000100	
3	0101000	00110	00111	0010	0000001000	
4	0110000	01001	01111	0110	0000010000	
5	1000001	01010	11111	0111	0000100000	
6	1000010	01100	11110	0101	0001000000	
7	1000100	10001	11100	0100	0010000000	
8	1001000	10010	11000	1100	0100000000	
9	1010000	10100	10000	1000	1000000000	

Tab. 4.1 Kódy na zobrazenie číslic desiatkovej sústavy

Všimnime si aspoň v náznakoch (podrobnosti nie sú cieľom tejto práce) niektoré vlastnosti kódov. Kód **BCD**, uvedený v tab.4.1, patrí do skupiny tzv. váhových kódov, v ktorých je každému bitu priradená určitá váha (významnosť). V kóde BCD sú to váhy 8, 4, 2, 1 (čo zodpovedá mocnine dvoch — ako sme zvyknutí z vyjadrenia čísla mnohočlenom). Hodnotu desiatkovej číslice vyjadrenej v kóde **BCD** určíme potom ľahko tak, že spočítame tie váhy, ktorým v kódovej kombinácii zodpovedá jednotka. Napríklad kódová kombinácia 0101 (s váhami 8421) obsahuje váhy 4 a 1, takže táto kombinácia zobrazuje desiatkovú číslicu päť(5). V *Tab. 4.1* sú uvedené aj ďalšie najčastejšie používané kódy, napr. **Aikenov kód**, ďalej **Rubinoffov kód**, ktorý má aj záporné váhy, alebo **kód  $3n + 2$** , v ktorom je desiatková číslica  $n$  zobrazená dvojkovým číslom, ktoré dostaneme vyčíslením výrazu  $3n + 2$ .

Kódy uvedené v hornej časti tabuľky sa používajú predovšetkým pre spracovanie čísel v aritmetických operáciách, keď často vyžadujeme jednoduchú tvorbu desiatkových alebo deviatkových doplnkov (a tým aj jednoduchšiu štruktúru príslušnej operačnej jednotky). Kódy typu **dva z piatich** sa uplatňujú pri konštrukcii čiarových kódov (pozri ďalej), ale aj ako kódy na spoľahlivý prenos informácie prenosovými kanálmi, tak ako kód **bikvinárny** alebo **jeden z desiatich**. Tieto kódy majú totiž určitú (a čím viac bitov tak vyššiu) redundanciu, ktorá je vhodná na to, aby sme prípadnú chybu pri prenose kódu odhalili, alebo dokonca opravili.

**Johnsonov** alebo **Grayov** kód majú pozoruhodnú zvláštnosť: kódy dvoch susedných čísel sa **líšia** vždy len **v jednom ráde** (v jednom bite). To ich predurčuje na to, aby boli vhodným prostriedkom na tvorbu analógovo-číslícových prevodníkov, najmä pri prevodoch polohy alebo uhla natočenia na číslo; podrobnosťami sa tu ďalej zaoberať nebudeme.

### Kód zvyškových tried

Nech je dané  $n$  prirodzených čísel  $z_1, z_2, \dots, z_n$  pričom každé z nich je väčšie ako 1. Označme tieto čísla ako **základy**. Ak  $A$  je celé číslo, bude jeho obrazom v sústave zvyškových tried so základmi  $z_1, z_2, \dots, z_n$  usporiadaná  $n$ -tica

$$(a_1, a_2, \dots, a_n), \text{ kde} \quad (4.1)$$

$$a_i = |A|_{z_i} \text{ pre } i = 1, 2, \dots, n \quad (4.2)$$

predstavuje teda celočíselný zvyšok po delení čísla  $A$  príslušným základom. Čísla  $a_i$  nazývame jednoducho **číslícami** a každá z číslic  $a_i$  je tzv. **hlavnou hodnotou** príslušnej

zvyškovej triedy obsahujúcu  $A$ . Ide tu o nepolyadickú číselnú sústavu, ktorú nazývame *sústava zvyškových tried* [ 15]. (Polyadické sústavy – pozri Prílohu A)

Číslica  $a_i$  je teda celé číslo, ktoré vyhovuje podmienke  $0 \leq a_i < z_i$  (4. 3)

a môžeme písať aj  $a_i = A \pmod{z_i}$  (4. 4)

Ako príklad uvažujme tri triedy so základmi 3, 5 a 7 a zobrazme napr. číslo  $A=28$  v takej sústave. Podľa (4. 4) dostaneme - ako obraz čísla 28 - usporiadanú trojicu (4, 3, 0).

Podobne ako v bežnej polyadickej sústave bol zavedený pojem *rád čísla (rádové miesto)*, aj tu možno toto označenie použiť. Ak je napr.  $n$ -tíca (4.1) obrazom čísla v sústave zvyškových tried so základmi  $z_1, z_2, \dots, z_n$ , hovoríme, že  $a_1, a_2, \dots, a_n$  sú postupne za sebou číslice v rádoch 1, 2, ...,  $n$  a môžeme hovoriť o nižších alebo vyšších rádoch. (V skutočnosti poloha číslic v  $n$ -tici je určená *poradím* v zápise základov tried, ktoré musí byť nemenné).

Zo vzťahu (4. 2) vyplýva, že v sústave zvyškových tried sú jednoznačne zobraziteľné všetky celé čísla. Preto môžeme využiť tento fakt napríklad aj na zobrazenie čísl desiatkovej sústavy. V tomto prípade sa používajú základy 2, 3 a 5. Uvedme, že  $P = z_1 \cdot z_2 \cdot \dots \cdot z_n$ , t.j. súčin všetkých základov tried, je tzv. *modulom* alebo *periodou zobraziteľnosti*, ktorá udáva, že obrazy čísel rovných alebo väčších než  $P$  sa periodicky opakujú ako v číslach 1 až  $P-1$  t.j. s modulom  $P$ . Ak sústavu zvyškových tried využijeme na kódovanie, potom celkom prirodzene hovoríme aj o kódoch zvyškových tried.

Číslica N	Kód zvyškových tried			Binárne kódovanie zvyškov		
	Základ $z_1 = 2$	Základ $z_2 = 3$	Základ $z_3 = 5$	$a_1$	$a_2$	$a_3$
	$a_1$	$a_2$	$a_3$			
0	0	0	0	0I	00I	0000I
1	1	1	1	I0	0I0	000I0
2	0	2	2	0I	I00	00I00
3	1	0	3	I0	00I	0I000
4	0	1	4	0I	0I0	I0000
5	1	2	0	I0	I00	0000I
6	0	0	1	0I	00I	000I0
7	1	1	2	I0	0I0	00I00
8	0	2	3	0I	I00	0I000
9	1	0	4	I0	00I	I0000

Tab. 4.2 Kód zvyškových tried pre desiatkové číslice N

Pre príklad s triedami o základoch 2, 3, 5 je  $P=30$ , a teda obrazy číslic 0, 1, 2, ..., 9 apriori vyhovujú požiadavke jednoznačnosti (determinovanosti) kódu. Tab. 4. 2 je tabuľkou kódu zvyškových tried s uvedenými základmi, a ten bol úspešne použitý v prvých česko-slovenských počítačoch SAPO a EPOS v päťdesiatych a šesťdesiatych rokoch dvadsiateho

storočia. Takéto kódovanie sa ukázalo ako zaujímavé: napr. súčet dvoch čísel väčší ako 10, povedzme  $8 + 7 = 15$ , pričom číslo 15 je samo o sebe zobraziteľné v tom istom kóde ako trojica (1 0 0). Ale číslica desiatkového prenosu v tomto obraze nie je priamo „viditeľná“, lebo súčet je menší ako modul zobrazenia. To umožnilo vytvárať sčítačky istým výhodnejším obvodomým riešením. Určité výhody kódu sa ukázali aj pri operácii násobenia. Binárne kódovanie číslic  $a_i$  v podobe ako udáva Tab.4.2 má tiež svoj zmysel a vlastne je obdobou kódu 1 zo  $z_i$ , ktorý tu opäť umožňuje jednoduchšie riešenie logických obvodov pre aritmetiku. Detaily takýchto prístupov tu však rozoberať nebudeme.

## 4.2 Alfnumerické kódy

Prenos informácií medzi číslicovými systémami prebieha prirodzene v dvojkovej forme, pričom „spoločný jazyk“ predstavuje najčastejšie **kód ASCII** (*American Standard Code for Information Interchange*). V tomto kóde sú definované nielen zobraziteľné znaky, napríklad veľké a malé písmená, číslice a osobitné znaky ( -, \$, %, @, ...), ale aj tzv. riadiace znaky. Tie sú určené na riadenie prenosu dát, na riadenie periférnych zariadení atď.

Okrem toho existuje ešte celý rad rozšírených znakových sád, z ktorých niektoré sa stali štandardnými [17]. Jednou z najpoužívanejších je americká osembitová rozšírená ASCII sada IBM, označovaná ako **kódová stránka 437** a jej podobu nájdete v Tab 4.4a. V pripojenej Tab. 4.4b je uvedená **kódová stránka 852**, ktorá obsahuje aj symboly slovens-

Znakové sady v UNICODE	Jazyk	Kód krajiny	Kód klávesnice	Základná znaková sada	Alternatívna znaková sada
znaky ASCII	nemčina	049	gr	850	437
Latin 1	angličtina britská	044	uk	437	850
rozšírený Latin	angl. medzinár.	061	-	437	850
grécke znaky	francúzština	033	fr	850	437
azbuka	italčina	039	it	850	437
arabské znaky	slovinčina	038	yu	852	850
hebrejské znaky	poľština	048	pl	852	850
ázijské znaky	slovenčina	042	sl	852	850
geometrické tvary	španielčina	034	sp	850	437
rôzne symboly	švédčina	046	sv	850	437
číselné znaky	čeština	042	cz	852	850
matematic.symboly	fínsky jazyk	358	su	850	437

Tab. 4. 3a,b Ukážky niektorých sád a kódov



kej abecedy. Uvádza ju tu nielen pre potreby slovenského informatika, ale hlavne preto, aby sme si uvedomili, že vo svete existuje veľké množstvo špecifických písmen a znakov, ktoré štandardný kód ASCII vôbec neuznáva: ide napr. o latinské abecedy, arabské, hebrejské, ruské, čínske a iné abecedy. Na základe takto chápanej univerzálnosti vznikol kód s názvom UNICODE, vybudovaný ako *Universal Multiple-Octet Coded Character Set*, tj. UCS, v ktorom prvých 128 znakov predstavuje štandardný kód ASCII a pre ďalšie jazyky sa s ohľadom na určité spoločné rysy vytvorili ďalšie **zjednocujúce znakové sady**. Základný (nie však úplný) prehľad udáva *Tab. 4.3a,b*. Prvá z nich je ukážkou niektorých príbuzných sád, druhá uvádza kódy a znakové sady niektorých krajín. Poznamenajme, že v časoch železnej opony sa u nás používali naše kódy *Kamenických*, *Latin2* alebo kód Rady vzájomnej hospodárskej pomoci *KOI-8 (Kod dlja obrotky informacij)*.

### 4.3 Čiarové kódy

Čiarové kódy sa využívajú v systémoch automatickej identifikácie prvkov, vyskytujúcich sa v najrozmanitejších oblastiach života spoločnosti, pri ich agende, evidencii, manipulácii s nimi. Tieto kódy významne zvyšujú produktivitu práce a spoľahlivosť pri spracovaní tak rozsiahleho druhu informácií napr. v priemysle, obchode, doprave a iných i spoločenských a sociálnych systémoch [12,13].

Označiť čiarovým kódom sa dá všeličo: etiketa s čiarovými kódmi môže byť papierová, plastová, kovová, keramická, atď. Od objavu čiarových kódov v r. 1949 sa vytvorilo a uplatňuje sa viac ako 200 rôznych čiarových kódov, a ani magnetické, indukčné či radiofrekvenčné metódy identifikácie nie sú doteraz tak rozšírené ako práve čiarové kódy.

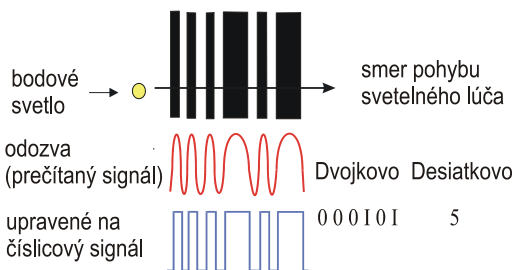
Je zrejmé, že o automatickej identifikácii sa dá hovoriť len v súvislosti s nasadením počítačov do praxe. Čítanie čiarových kódov umožňujú rozličné snímacie perá, alebo najčastejšie laserové skenery, pripojené k počítaču väčšinou prostredníctvom štandardného rozhrania (RS 232, RS 485, Alt 0133, ...), alebo v prípadoch rozsiahlych dátových systémov aj automatizovanými zbernými systémami.

Základným porovnávacím kritériom kódov je ich kódovacia tabuľka. Podľa toho, ktoré znaky obsahuje, hovoríme o kódoch *numerických*, *numerických so špeciálnymi znakmi*, *alfanumerických* a *úplných alfanumerických*. Druhým porovnávacím kritériom je *počet znakov* v kódovom obraze, a tretím môže byť *oblasť použitia* kódu. Stručný prehľad niektorých najuvádzanejších kódov je v *Tab. 4.5*.

Čiarový kód je postupnosť medzier a čiar definovanej hrúbky a výšky, ktorá sa optoelektronicky sníma - spravidla laserovým svetelným lúčom, pričom sa generujú elektronické impulzy zodpovedajúce skladbe medzier a čiar, napr. podľa obr. 4.1. Veľkosť poľa čiar a medzier je v kódovom obrazci vždy definovaná istými charakteristickými veličinami, podľa obr. 4.2. Kód môže byť zostrojený v rozličných veľkostiach a hustote podľa prostredia, pre ktoré je určený, vždy však musí byť zachovaný potrebný kontrast medzi farbou medzery a čiary, aby snímacie zariadenie mohlo spoľahlivo kód vyhodnotiť.

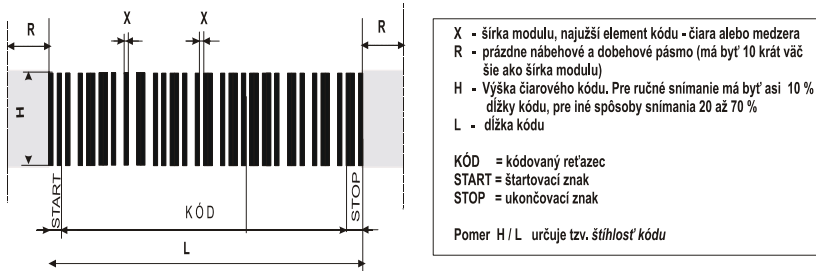
Kód	Typ znakov	Počet znakov	Dĺžka kódového obrazca	Oblasť použitia
UPC A	numerický	10	pevná (12 znakov)	obchod
UPC E	numerický	10	pevná (8 znakov)	obchod
EAN 8	numerický	10	pevná (8 znakov)	obchod
EAN 13	numerický	10	pevná (13 znakov)	obchod
Code 2/5	numerický	10	premenná	technika
Codabar	num. a špec. znaky	16	premenná	zdravníctvo
Code 39	num., špec. znaky, veľká abeceda	43	premenná	všeobecné, elektrotechnika
Code 128	ASCII 128	ASCII 128	premenná	technika, farmácia, medicína, ...

Tab. 4.5 Porovnanie niektorých vlastností čiarových kódov



Obr. 4.1 Reprezentácia čiarového kódu

Kódy UPC (Universal Product Code) sú rozšírené v Kande, kódy EAN (European Article Numbering) sú typické najmä pre Európu a ostané krajiny. Nebudeme tu však opisovať celú množinu kódov, uvedieme len niektoré príklady.



Obr. 4. 2 Charakteristické veličiny kódového poľa



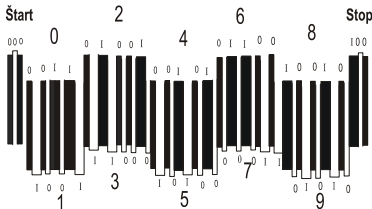
Obr. 4. 3 Kód 2/5 Industrial

## Kód 2/5 Industrial

Vznikol v r. 1968 vo firme *Identicon Corporation* ako kód s variabilnou dĺžkou, znakmi 0 až 9 a znakmi Štart a Stop. Je to päťčiarový kód (dva z piatich), ktorého definičnú podobu možno ľahko odhaliť pomocou príkladu na obr. 4.3.

## Kód 2/5 Prekrývaný

Snaha podstatne zvýšiť hustotu kódu viedla k vzniku tzv. prekrývaného (interleaved) päťčiarového kódu (dva z piatich), v ktorom sa číselný význam nepriradzuje len čiaram, ale aj medzerám - hustota je teda dvojnásobná. Konkrétnu podobu kódu ľahko odvodíme z ukážky na obr. 4.4.



Obr. 4. 4 Prekrývaný kód 2/5



Obr. 4. 5 Kód Codabar

V skupine 2/5 je viacero kódov: napr. kód 2/5 invertovaný, *Datalogic*, *Matrix* apod. Podobne sa členia aj kódy zo skupiny **Codabar**, ktoré boli vyvinuté v r.1972 pre označovanie cien v maloobchode. Codabar je numerický kód s variabilnou dĺžkou. Obsahuje symboly 0 až 9, špeciálne znaky (+, -, :, /, ., \$) a štyri identické znaky Start/Stop. Ukážka je na obr. 4. 5. Ďalší kód tejto skupiny je kód ABC-Codabar.

Prvý (1974) so skupiny kódov **Code 39** bol štandardný kód s alfanumerickými znakmi a premenlivou dĺžkou. Obsahoval špeciálne znaky \*, /, ., \$, +, znaky 0 až 9, a veľké písmená A až Z. Ďalší bol *Rozšírený Code 39 (FULL ASCII)*, ktorý už umožňoval prácu s úplnou tabuľkou ASCII znakov.

Skupina Code 93 bola vyvíjaná od r. 1982. Prvý kód **Code 93** bol alfanumerický s variabilnou dĺžkou, so znakmi Start/Stop, znakmi 0 až 9, znakmi veľkej abecedy A až Z, so špeciálnymi znakmi (+, -, :, /, ., \$, %) a riadiacimi znakmi (+), (/), (\$) ,(%), spolu 43 symbolov a 4 riadiace znaky. Nasledoval *Rozšírený Code 93 (FULL ASCII)* ktorý tiež obsahuje všetky znaky ASCII.

Kód **Code 128** vyvinutý v r. 1981 pracuje s alfanumerickými znakmi, s premenlivou dĺžkou a má 128 ASCII znakov, 4 špeciálne znaky, 4 riadiace znaky, 3 štartovacie znaky a jeden znak Stop. Zaujímavé je, že má dokopy tri sady znakov, označených *A, B, C*. Sada *A* obsahuje numerické znaky, znaky veľkej abecedy, riadiace a špeciálne znaky. Sada *B* obsahuje navyše ešte aj znaky malej abecedy. Sada *C* obsahuje dvojice numerických znakov od 00 do 99, riadiace a špeciálne znaky. Pomocou tejto sady možno kódovať numerické údaje s dvojnásobnou hustotou.

## Kódy EAN

Takto označované kódy sú dnes už svetovým štandardom a riadi ho Medzinárodná organizácia IANA *EAN (International Article Numbering Association)*. Kódy EAN 13

i EAN 8 sú numerické s pevnou dĺžkou. Definované sú tri kódovacie tabuľky: *A* pre nepárnu paritu, *B* a *C* pre párnú paritu. Na kódovanie numerických údajov je k dispozícii 30 rozličných

Pozície číslic							
13	12	11	10	9	8	7	6 až 1
0	A	A	A	A	A	A	C
1	A	A	B	A	B	B	C
2	A	A	B	B	A	B	C
3	A	A	B	B	B	A	C
4	A	B	A	A	B	B	C
5	A	B	B	A	A	B	C
6	A	B	B	B	A	A	C
7	A	B	A	B	A	B	C
8	A	B	A	B	B	A	C
9	A	B	B	A	B	A	C

ČÍSLICE	ZNAKOVÁ SADA		
	A	B	C
0			
1			
2			
3			
4			
5			
6			
7			
8			
9			
Okrajové značky			

Tab. 4.4 Tabuľka výberu sady

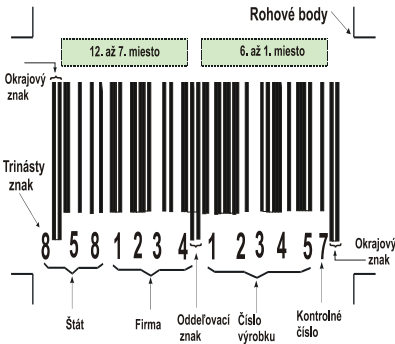
Obr. 4.6 Kódovanie znakov v sadách

kombinácií. Ktorú z nich treba vybrať, to určuje číslica, ktorá je na 13. pozícii. Zdôvodňovať konštrukcie kódu nebudeme, na ilustráciu postačuje nasledujúci príklad. Všimnime si kód EAN 13 s dĺžkou 13 číslic a uvažujme kódovanie reťazca **8 5 8 1 2 3 4 1 2 3 4 5 7**.

Obrázok 4.6 určuje, ktorá sada bude vybraná pre kódovanie číslice podľa jej pozície v kódovanom reťazci. V našom prípade je na 13. pozícii číslica **8**, preto pre jednotlivé číslice určuje riadok 8 v Tab.4.4 nasledujúce sady takto:

Pozícia	13. znak	okraj. znak	12	11	10	9	8	7	oddeľ. znak	6	5	4	3	2	1	okraj. znak
Číslica	<b>8</b>		<b>5</b>	<b>8</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>		<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>7</b>	
Sada	A		A	B	A	B	B	A		C	C	C	C	C	C	

Obrázok 4.6 určuje, ako kódovať jednotlivé číslice podľa príslušnej sady, a tak výsledná podoba kódu je na obr. 4.7. V tomto obrázku sme uviedli aj niektoré dodatkové informácie: vyznačili sme napríklad, čomu slúžia určené skupiny čísel (napr. skupina **858** označuje Slovensko ako štát vyrábajúci príslušný výrobok, na ktorom bude kód vyznačený), rohové značky naznačujú veľkosť kódového poľa, čo sú to okrajové značky, apod.



Obr. 4.7 Príklad kódu EAN 13



Obr. 4.8 Kód EAN s dodatkom

Niektorým používateľom nestačí trinásť miest na vyjadrenie ich potrieb, a preto siahnu po tzv. **dotatkových kódoch EAN**. Takéto kódy sa používajú napr. v časopisoch a knihách ako prídavná informácia k číslam ISSN a ISBN, a to najmä pre presnejšie označenie vydania.

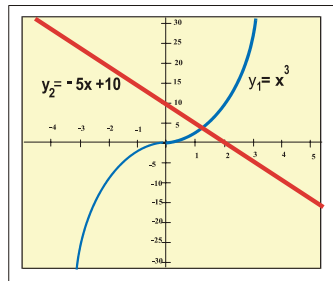
Na obr. 4.8 je príklad **dvojmiestneho** dodatkového kódu EAN, u ktorého pri čítaní oboch častí dostaneme reťazec zložený z oboch kódov. Pravidlá pre vytváranie dodatkového kódového obrazu sú podobné tým, ktoré sme uviedli vyššie, ale nebudeme ich tu opisovať. Pripomeňme len, že okrem dvojmiestneho dodatkového kódu sa používa aj **päťmiestny** dodatkový kód.

## 5. GRAFY

### 5.1 Základné predstavy

**Graf** je v odbornej verejnosti - ale mnohokrát aj v bežnej praxi - často používané slovo. Najčastejšie predstavuje istý útvar v rovine, ktorý dovoľuje prehľadne znázorniť mnohé vzťahy a súvislosti, ktoré by inak (napr. v podobe rozsiahlych číselných tabuliek) boli možno neprehľadné alebo ťažšie pochopiteľné. Príklady takejto podoby grafov sú na nasledovných obrázkoch a je zrejmé, že názornosť ktorú poskytujú, je mimoriadne cenná.

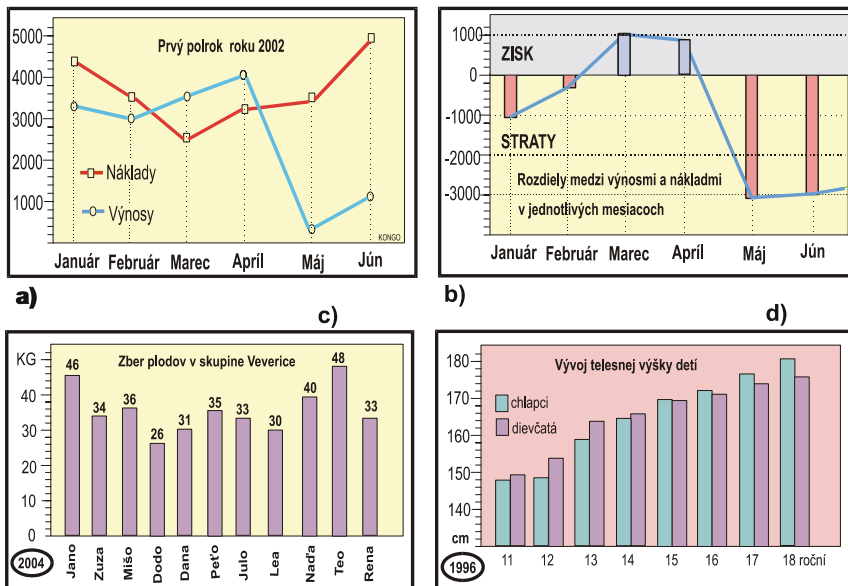
Obrázok 5.1 predstavuje takú grafickú štruktúru, ktorú nazývame *graf matematickej funkcie* alebo *priebeh matematickej funkcie*. Sú tam zobrazené dve funkcie:  $y_1 = x^3$  (modrá krivka) a  $y_2 = -5x + 10$  (červená priamka). Vodorovná os prislúcha nezávisle premennej  $x$  a zvislá os prislúcha funkčným hodnotám, ktoré možno odčítať podľa naznačenej stupnice. Stupeň presnosti odčítania hodnôt je daný kvalitou kreslenia a má v podstate orientačný charakter postačujúci na to, aby sme mohli vyslovovať tvrdenia o vlastnostiach zobrazených funkcií. Dá sa teda poznať, že krivka  $y_1$  je tzv. kubická parabola a priamka  $y_2$  je stále klesajúcou funkciou.



Obr.5.1 Graf matematických funkcií

V celom rade praktických úloh sa uplatňujú analogicky ďalšie princípy, ktoré umožňujú hodnotiť procesy, na ktoré sa viažu svojou grafickou podobou. Ide napríklad o grafické znázornenie priebehu telesnej teploty pacienta, vývoj cien v čase apod.

Na obr.5.2a je znázornený časový priebeh nákladov a výnosov v nejakom výrobnom podniku za isté obdobie. Sledované parametre sa zobrazia ako body v príslušnom obrázku. Ak body *nepospájame* čiarami, vytvorili sme tzv. **bodový graf**. Ak však susedné body pospájame úsečkami, dostaneme podobu, ktorú označujeme ako **čiarový graf** alebo **čiarový diagram**. Ten názornejšie zobrazuje **časovú závislosť** parametrov i možný trend ich zmien, a poslúži tak riadiacim pracovníkom v ich rozhodovacom procese. Z obrázku sa dá usúdiť, že na prelome februára a marca začala byť výroba efektívna, no od mája opäť náklady prevýšili výnos, takže treba vykonať nejaké optimalizačné zásahy do sledovanej výroby.



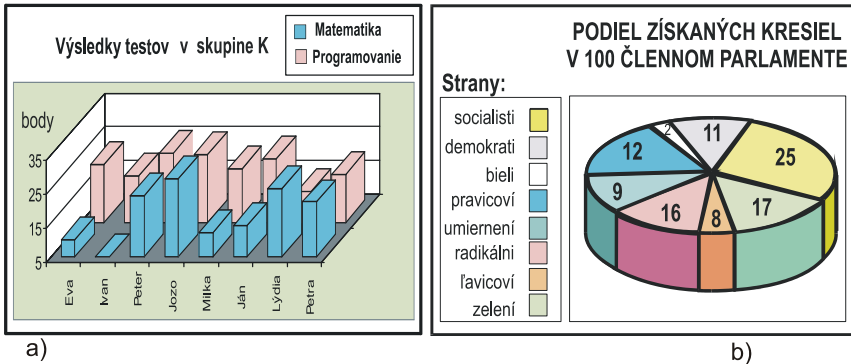
Obr.5.2 Čiarové diagramy a histogramy

Niekedy treba graficky zvýrazniť istý vzájomný pomer sledovaných parametrov. Príkladom je obr. 5.2b. Je to opäť čiarový diagram, v tomto prípade odvodený z obr.5.2a tak, že vyjadruje rozdiel medzi nákladmi a výnosmi ako funkciu času. Tento rozdiel nadobúda kladnú a zápornú hodnotu (čo je v grafe okrem lomenej čiary vyznačené aj príslušnou výškou stĺpcov), preto rozdeľuje obrázok na oblasť zisku a straty. Táto podoba je oveľa názornejšia ako v grafe predchádzajúcom. Graf, ktorý vyjadruje vzájomnú súvislosť parametrov sa označuje aj ako *diagram súvislosti* a je vhodným nástrojom riadiacej práce.

Vodorovná os grafu však nemusí vyjadrovať len časové závislosti. Je mnoho procesov, v ktorých sa vyjadrujú hodnoty parametrov pre diskkrétne (nespojité) prvky, ako napr. pre konkrétnych ľudí, pre konkrétne body, pre konkrétne skupiny, mestá apod. V takýchto prípadoch sa používa graf, ktorý nazývame **stĺpcový diagram** alebo *histogram*. Na obr.5.2c je príklad takého histogramu, a – ako sa dá usúdiť – znázorňuje výsledky práce niekoľkých detí pri zbere istých plodov. Nazbierané množstvo (napríklad v kilogramoch) je vyznačené výškou stĺpca pri mene dieťaťa a celý obrázok potom predstavuje isté hodnotenie práce

zodpovedajúcej skupiny detí i jednotlivcov v nej. Príklad isto nepotrebuje podrobnejší komentár, zmysel je celkom zrejмый: prezentovať či zviditeľniť isté pracovné výkony.

Histogram je vhodným prostriedkom aj na vyjadrenie istých previazaností sledovaných parametrov, ktoré vyjadrujú rovnakú vlastnosť pre príbuzný prvok na vodorovnej osi, napr. podľa obr.5.2d. Graf udáva vývoj telesnej výšky pre chlapcov a dievčatá v istom rozmedzí ich veku, pričom zodpovedajúce stĺpce sú združené vždy do dvojice. Ľahko tak možno sledovať ich vzájomný súvis a jeho zmeny. Takýto graf nazývame *združený histogram*. Združujú sa aj viacej než dva stĺpce, nemá ich byť však toľko, aby sa stratila názornosť.



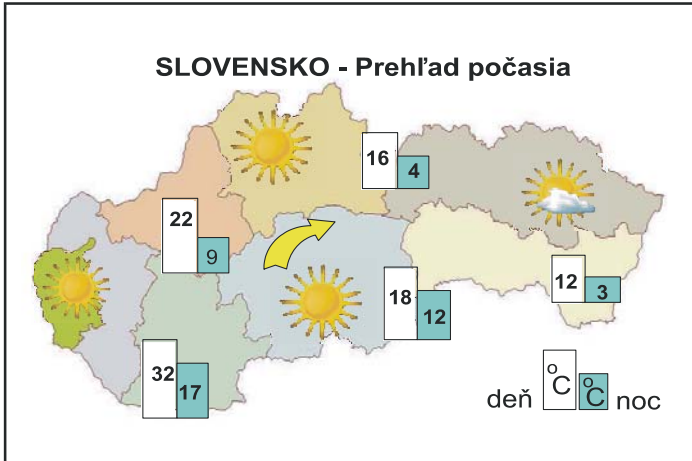
Obr. 5.3 Priestorový (3D) diagram a koláčový graf

Moderné programovacie prostriedky umožňujú kresliť združené histogramy aj v priestorovej podobe tzv. **trojdimenzionálneho** (3D) grafu. Príklad takéhoto grafu je na obr.5.3a a isto nevyžaduje podrobnejší komentár.

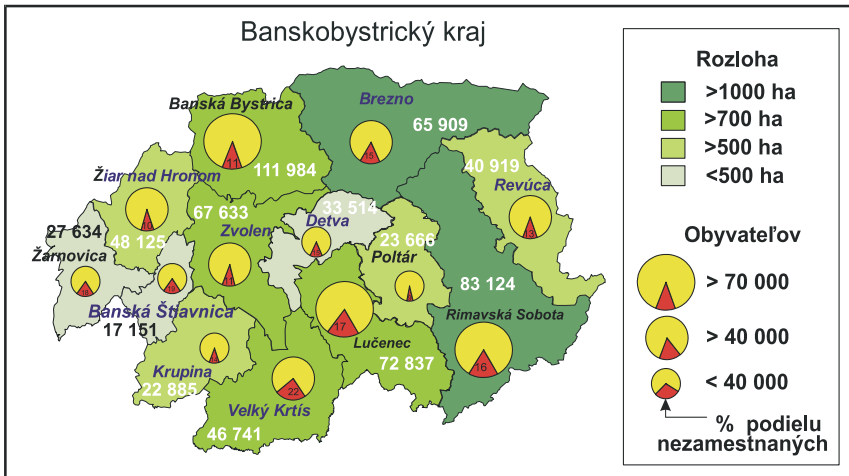
Ak chceme vyjadriť rozdelenie nejakého celku na dielčie skupiny podľa istého stanoveného kritéria, je vhodné použiť veľmi názorný *koláčový graf*. Príkladom je graf na obr.5.3b, ktorý zobrazuje podiel jednotlivých politických strán na rozdelení kresiel v stočlennom parlamente nejakého štátu. Veľkosť príslušnej kruhovej výseče v uvedenom obrázku môže tak pozorovateľovi naznačiť význam alebo silu príslušných strán v politickom živote. Pripomeňme ešte, že aj tento obrázok naznačuje priestorovú podobu, často sa však kreslí len ako kruh rozdelený na príslušne veľké **výseče**. Uvedený príklad je tiež dost názorný, a preto nebudeme uvádzať ďalšie podrobnosti.

S rozvojom geografických informačných systémov sa značne rozvinulo zobrazovanie mnohých závislostí a vplyvov na život v spoločnosti pomocou **kartodiagramov**. *Kartodiagramy* predstavujú obľúbený a obsažný grafický prostriedok uľahčujúci vytváranie

si názoru na javy a procesy, ktoré zaujímajú nielen odborníkov, ale aj verejnosť, predovšetkým prostredníctvom najrôznejších médií. Nasledujúce obrázky sú príkladom takých kartodiagramov. Prvý z nich, *obr.5.4a*, možno v naznačenej podobe sledovať denne pri predpovedi počasia. Ako vidno, je na mapku Slovenska – nad regióny na ktoré sa viaže – vložený zdužený histogram denne a nočnej teploty, prípadne ďalšie indikátory (slniečko,



a)



b)

Obr.5.4 Príklady kartodiagramov

mraky, smer vetra... ). Poskytuje to pozorovateľovi toľko informácií, že sa v podstate môže podľa nich riadiť. Druhý príklad na obr.5.4b znázorňuje určitý stav vybraných parametrov v Banskobystrickom kraji. Prvý parameter je rozloha príslušného okresu, ktorú vyjadruje sýtosť farby a zaraďuje tak okresy do štyroch veľkostných skupín. Ďalším parametrom je počet obyvateľov v okrese – okrem konkrétneho čísla ho vyjadruje aj veľkosť koláčového grafu umiestneného na území príslušného okresu, a v ňom kruhová výseč ukazuje percento podielu počtu nezamestnaných vzhľadom na počet obyvateľov. Takýto graf teda predstavuje istý integrovaný pohľad na stav sledovaných vecí a umožňuje prijímať také závery alebo robiť také aktivity, ktoré treba na dosiahnutie optimálneho stavu.

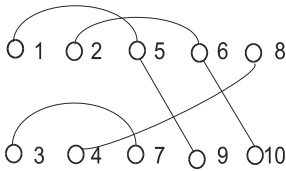
Na záver konštatujeme, že grafických prostriedkov znázorňujúcich a zvyrazňujúcich potrebné pojmy, javy, fakty alebo súvislosti, je celý rad. To čo sme tu ukázali sú ich základné podoby. V praxi sa všelijako kombinujú, obmieňajú, spájajú a modifikujú tak, aby ich názornosť bola nástrojom rýchleho vnímania problémov a následne ich optimálneho riešenia.

## 5. 2 Neorientované grafy

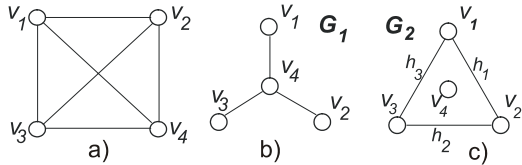
Teraz chceme ukázať, že slovo graf sa v matematike a nadväzujúcich disciplínach používa v inom význame než bolo povedané v predchádzajúcom texte, a takýto matematický náhľad budeme prezentovať v ďalšom texte s prijateľným stupňom stručnosti.

Uvažujme množinu čísel  $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$  a skúsme vhodne vyjadriť napr. fakt, že sa číslo v tejto množine líši od ostatných čísel v nej práve o 4. Možno to urobiť napr. tak, že uvažované čísla zobrazíme v rovine ako body (krúžky), a tie čísla, pre ktoré platí vyslovené tvrdenie, spojíme čiarami (spojnicami). Dostaneme tak útvar uvedený na obr.5.5, ktorý má 6 spojnic a 10 bodov. Tento – isto názorný príklad - ukazuje, že sa dá hovoriť o množine bodov (krúžkov), tú budeme označovať  $V$ , a o množine spojnic (čiar), ktorú označíme  $H$ . Pre prvok množiny  $V$  sa ujal názov *vrchol*, a spojnicu dvoch vrcholov nazývame *hrana* [18,19].

Poznamenajme, že vo všeobecnosti sa hrany môžu navzájom krížiť, čo však nie je dôležité pre vyjadrenie podstaty toho, čo takýto útvar predstavuje. A podobne, zaujímajú nás iba tie body, ktoré sme nazvali vrcholmi; budeme ich graficky zobrazovať v podobe krúžkov. Pripomeňme však, že *vrchol* sa v niektorých prameňoch nazýva aj *uzol*. V ďalšom texte však budeme používať názov **vrchol**.



Obr.5.5 Neorientovaný graf



Obr.5.6 Úplný graf a komplementárne grafy

Formulujme teraz často v literatúre udávanú matematickú definíciu neorientovaného grafu:

**Def. 5.1:** Neorientovaný graf je trojica

$$G = [V, H, \varepsilon] \quad (5.1)$$

tvorená konečnou množinou  $V$ , ktorej prvky nazývame vrcholmi, konečnou množinou  $H$ , ktorej prvky nazývame hranami, a zobrazením  $\varepsilon : H \rightarrow V \times V$ , ktoré nazývame vzťahom incidencie, a ktoré priradzuje každej hrane  $h \in H$  jednoprvkovú alebo dvojprvkovú množinu vrcholov.

Tieto vrcholy nazývame aj *krajné vrcholy hrany*  $h$ .

Vrcholy grafu sa obvykle označujú symbolmi malej abecedy, napr.  $v_1, v_2, \dots, v_i, \dots$  a hrany sú vlastne dvojprvkové podmnožiny množiny  $V$ , takže napr. hranu medzi vrcholmi  $v_i$  a  $v_j$  môžeme zapísať takto:  $\{v_i, v_j\}$ , jednoduchšie  $v_i v_j$ , alebo najjednoduchšie tak, že dvojicu  $v_i v_j$  nahradím symbolom  $h_s$ . Potom takúto hranu  $h_s \in H$ , ( $s = 1, 2, 3, \dots$ ) chápeme ako prvok z množiny hrán  $H$ . Pripomíname, že v takýchto grafoch nezáleží na tvare spojnice, teda hrana môže byť priamkou, oblúkom alebo inou spojitou čiarou, lebo jej existencia nevyjadruje nič iné, len to že vrcholy ktoré spája, viaže určitá spoločná vlastnosť.

Hovoríme, že hrana  $v_i v_j$  *inciduje* s vrcholmi  $v_i$  a  $v_j$  (alebo je *incidentná* s vrcholmi  $v_i$  a  $v_j$ ), pričom tieto vrcholy sú *koncovými vrcholmi* hrany  $v_i v_j$ . (Platí aj opak : **vrchol je incidentný** s hranou). Ak sú dva vrcholy spojené hranou, nazývame ich *susedné vrcholy*. Ak hrana  $v_i v_j$  *inciduje* s vrcholmi  $v_i$  a  $v_j$  pričom  $v_j = v_i$ , potom ju nazývame *sľučka*.

Analogicky sa možno vyjadriť aj o rozmiestnení vrcholov: nezáleží na tom, kde na ploche niektorý vrchol leží - vo vzťahu k polohe ostatných vrcholov. Je však prirodzené, že *žiadnen vrchol nemôže ležať na inom vrchole*, lebo graf by prestal naplňovať svoj zmysel a došlo by k nejednoznačnosti.

Ak je množina vrcholov prázdna, zrejme je prázdna i množina hrán. Potom sa takýto graf tiež nazýva *prázdny* graf. Naopak, ak  $H$  je množina *všetkých* dvojprvkových častí množiny  $V$ , potom sa graf  $G = [V, H]$  nazýva **úplný**. Příklad úplného grafu je na obr. 5.6 a.

Ak v neorientovanom grafe  $G = [V, H]$  je množina  $V$  konečná, potom hovoríme, že ide o **konečný neorientovaný graf**. Opačne, pre nekonečnú množinu  $V$  je graf nekonečný.

Všimnime si niektoré vlastnosti, ktoré sa viažu na vrchol. Je to predovšetkým počet hrán, ktoré s vrcholom incidujú. Ak je takých hrán konečný počet, hovoríme, že *vrchol je konečného stupňa*, a ak sú v grafe všetky vrcholy konečného stupňa, potom ide aj o graf konečného stupňa. (Grafmi nekonečného stupňa sa tu nebudeme zaoberať.) Hovoríme: ak je  $v_i$  vrchol konečného stupňa v danom grafe  $G$ , potom *počet hrán* ktoré sú incidentné s vrcholom  $v_i$  nazývame *stupňom vrcholu*  $v_i$ .

Na obr. 5.6 b,c sú dva grafy  $G_1$  a  $G_2$ . Lahko poznáme, že v grafe  $G_1$  vrcholy  $v_1, v_2$  a  $v_3$  sú stupňa 1, vrchol  $v_4$  stupňa 3, v grafe  $G_2$  zase vrcholy  $v_1, v_2$  a  $v_3$  sú stupňa 2 a vrchol  $v_4$  stupňa 0. Vrchol, ktorého stupeň je 0 (neinciduje so žiadnou hranou) je *izolovaný*.

Obrázok 5.6 b,c umožňujú zaviesť ďalšie pojmy. Nech platí, že dva grafy  $G_1$  a  $G_2$  sú také, že  $G_1 = [V, H_1]$  a  $G_2 = [V, H_2]$ , (t.j. majú spoločnú množinu vrcholov). Ďalej nech prienik  $H_1 \cap H_2 = \emptyset$ , (t.j. žiadna hrana sa nevyskytuje v oboch množinách hrán grafov  $G_1$  a  $G_2$ ) a nech zjednotenie množín hrán vedie na úplný graf  $[V, H_1 \cup H_2]$ . Potom hovoríme, že grafy  $G_1$  a  $G_2$  sú *navzájom komplementárne*. Skutočne, grafy na obr. 6.6 b,c vyhovujú daným predpokladom, a sú teda navzájom komplementárne.

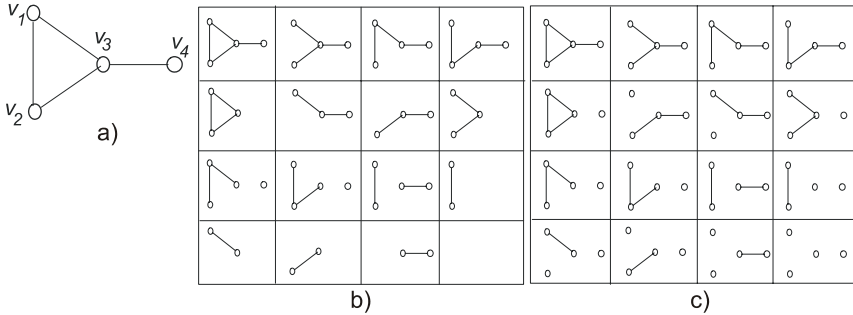
Ak sú dané dva grafy  $G_1 = [V_1, H_1]$  a  $G_2 = [V_2, H_2]$  také, že  $V_1 \cap V_2 = \emptyset$  (prienik oboch množín vrcholov je *prázdna* množina), potom hovoríme, že grafy  $G_1$  a  $G_2$  sú *disjunktné grafy*.

Ak sú dané dva grafy  $G_1 = [V_1, H_1]$  a  $G_2 = [V_2, H_2]$  také, že  $V_1 = V_2$  a súčasne platí  $H_1 = H_2$ , potom hovoríme, že grafy  $G_1$  a  $G_2$  *sú si rovné* a píšeme  $G_1 = G_2$ . Naopak, ak neplatí  $G_1 = G_2$ , potom sú grafy *rôzne*, čo zapisujeme  $G_1 \neq G_2$ .

### Podgrafy daného grafu

Nech sú dané dva grafy  $G_1 = [V_1, H_1]$  a  $G_2 = [V_2, H_2]$  také, že  $V_1 \subset V_2$  (množina vrcholov prvého grafu je podmnožinou vrcholov druhého) a súčasne platí  $H_1 \subset H_2$  (množina hrán prvého grafu je podmnožinou hrán druhého), potom hovoríme, že graf  $G_1$  je *podgrafom* grafu  $G_2$ . Ak súčasne neplatí  $V_1 = V_2$  a  $H_1 = H_2$ , potom  $G_1$  nazývame *vlastným podgrafom* grafu  $G_2$ .

Na obr. 5.7a je príklad grafu so 4 vrcholmi a ak určíme všetky jeho podgrafy, ktoré neobsahujú žiaden izolovaný vrchol, dostaneme obr. 5.7b. Nie sú v ňom však všetky možné podgrafy k danému grafu.



Obr.5. 7 Podgrafy a faktory grafu

Ukázalo sa, že osobitný význam majú **podgrafy**, ktoré majú rovnakú množinu vrcholov ako základný graf. Ak je daný graf  $G = [V, H]$  a aj jeho podgraf  $G_0 = [V, H_0]$ , potom hovoríme, že  $G_0$  je *faktorom grafu*  $G$ . (Faktor teda obsahuje tú istú množinu vrcholov ako pôvodný graf. Z toho sa dá tiež usúdiť, že každý graf je svojim vlastným faktorom. Ukážka všetkých faktorov grafu z predchádzajúceho príkladu je na obr. 5.7c.

### Súvislosť grafov

Predpokladajme teraz, že je daný nejaký graf  $G = [V, H]$  a v ňom zvolíme vrcholy  $v_0$  a  $v_n$ . Ak sa dá zostrojiť konečná postupnosť vrcholov a hrán grafu  $G$  v tvare

$$v_0, v_0v_1, v_1, v_1v_2, v_2, \dots, v_{n-1}, v_{n-1}v_n, v_n, \quad (5.2)$$

alebo

$$v_0, h_1, v_1, h_2, v_2, \dots, v_{n-1}, h_n, v_n,$$

potom sa táto postupnosť nazýva *sled* medzi vrcholmi  $v_0$  a  $v_n$ . Hovoríme tiež, že tento **sled začína** vo vrchole  $v_0$  a **končí** vo vrchole  $v_n$ . Ostatné vrcholy tohoto sledu sú *vnútorné*. Číslo  $n$  je dôležitým údajom, lebo znamená *počet hrán* v uvažovanom slede, a preto hovoríme, že  $n$  je *dĺžka sledu* medzi vrcholmi  $v_0$  a  $v_n$ .

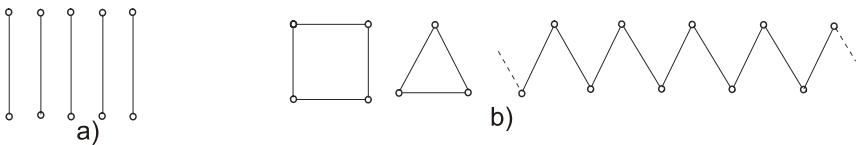
Pozrime sa na *obr. 5. 6c*. Je zrejme že medzi vrcholom  $v_1$  a vrcholom  $v_3$  existuje sled  $S_1$  dĺžky 2 a to  $v_1, h_1, v_2, h_2, v_3$ , ale tiež aj sled  $S_2 = v_1, h_3, v_3$  dĺžky 1. Na druhej strane, medzi vrcholom  $v_1$  a vrcholom  $v_4$  toho istého grafu žiaden sled neexistuje.

Vo všeobecnosti teda medzi dvomi vrcholmi buď neexistuje žiaden sled, alebo medzi nimi existuje nekonečne mnoho sledov. Zápis sledu (5.2) totiž neobmedzuje počet opakovaní pri priechodoch niektorými vrcholmi, takže pre graf z *obr. 5. 6c* je sledom aj postupnosť  $S_3 = v_1, h_1, v_2, h_2, v_3, h_3, v_1, h_1, v_2, h_2, v_3$  dĺžky 5 (a všetky sledy analogicky odvodené). Isto nás zaujíma sled *s najmenšou dĺžkou*, t.j. ten, čo obsahuje najmenší počet hrán. Označme túto dĺžku symbolom  $d_{\min}$ . Toto číslo sa ľahko nájde, ale musíme pripustiť, že môže existovať niekoľko sledov s najkratšou dĺžkou. Preto zavedieme ďalšie obmedzenia a zamerajme sa na istý špeciálny druh sledu. *Nech v grafe  $G$  existuje sled  $C$ , v ktorom sa každý vrchol grafu vyskytuje nanajvýš raz. Potom sa takýto sled  $C$  nazýva cesta. Sled, v ktorom sa žiadna hrana neopakuje, sa nazýva **Ľah**.*

Ak existuje sled vedúci z vrcholu  $v_i$  do vrcholu  $v_j$ , potom hovoríme, že vrchol  $v_j$  je *dostupný* z vrcholu  $v_i$

Nakoniec definujeme súvislý graf: *Graf  $G$  sa nazýva súvislý, ak medzi každou dvojicou jeho vrcholov  $v_i$  a  $v_j$  existuje aspoň jeden sled.*

Existujú isto aj grafy, ktoré nie sú súvislé a skladajú sa zo zložiek, ktoré nazývame **komponenty**. Definujeme komponent  $G(v)$  grafu  $G$  takto: *Komponent  $G(v)$  je podgraf grafu  $G$  taký, že do  $G(v)$  patria práve všetky tie vrcholy a hrany grafu  $G$ , ktoré ležia aspoň v jednom slede grafu  $G$ , ktorý začína vo vrchole  $v$ . Inak tiež hovoríme, že  $G(v)$  je maximálny súvislý podgraf grafu  $G$ , ktorý obsahuje vrchol  $v$ . (Slovom „maximálny“ tu chceme vyjadriť fakt, že  $G(v)$  nie je vlastným podgrafom v žiadnom súvislom podgrafe grafu  $G$ .) Na *obr.5. 8 b* je ukážka grafu pozostávajúceho z troch komponent.*



Obr.5.8 Pravidelné grafy

Všimnime si opäť pojem *vzdialenosť* dvoch vrcholov v ceste. Uvažujme ľubovoľný konečný súvislý graf  $G$  a pre každé jeho vrcholy  $v_i, v_j$  definujme číslo  $d(v_i, v_j)$  ako dĺžku najkratšieho sledu medzi  $v_i$  a  $v_j$ . V súvislom grafe  $G$  platí, že existuje prirodzené číslo  $m$  také, že pre každé dva vrcholy  $v_i, v_j$  grafu, platí  $d(v_i, v_j) < m$ . Najväčšia z hodnôt  $d(v_i, v_j)$  sa

potom nazýva *priemer grafu*. **Priemer grafu** je teda určený počtom hrán medzi najvzdialenejšími vrcholmi v grafe. Ak ide o nekonečný graf, je priemer tiež nekonečný.

### Pravidelné grafy

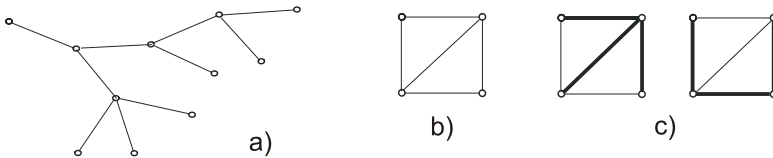
Zaujímavou skupinou sú také grafy, ktorých vrcholy sú rovnakého stupňa  $s$ , a preto aj graf  $s$  touto vlastnosťou nazývame *pravidelný graf  $s$  - tého stupňa*. Na obr.5. 8a je ukážka pravidelného grafu 1. stupňa. Obrázok 5.8b predstavuje príklad **nekonečného** grafu druhého stupňa a ako vidno, tento graf sa skladá z troch komponentov: prvý obsahuje 4 vrcholy, druhý 3 vrcholy a tretí je nekonečný, ale všetky obsahujú vrcholy 2. stupňa.

Pre niektoré typické grafy sa ujali názvy z geometrie. Napr. konečný súvislý graf druhého stupňa sa nazýva *kružnica*. Počet vrcholov, z ktorých sa **kružnica** skladá, sa nazýva *dĺžka kružnice*. Prvý komponent grafu na obr. 5. 8b je kružnicou, má dĺžku 4 a nazýva sa aj *štvoruholník*. Druhý komponent toho grafu je kružnica dĺžky 3 a označuje sa aj ako *trojuholník*.

Všimnime si ešte ďalšie vlastnosti predovšetkým konečných grafov. Ide najmä o určité grafy, ktoré majú tú vlastnosť, že neobsahujú podgraf typu kružnica.

### Strom a kostra grafu

*Konečný súvislý graf, ktorý neobsahuje ako podgraf žiadnu kružnicu, sa nazýva **strom***. Názov je zrejme odvodený od podoby obrázku grafu, ktorá pripomína vetvenie kmeňa a konárov skutočného stromu. Príklad grafu typu strom je na obr.5.9a. Ak v uvedenej definícii stromu vypustíme požiadavku súvislosti, potom taký graf nazývame *les*. (Možno si to predstaviť aj tak, že **les** tvoria komponenty grafu, ktoré sú typu strom.)

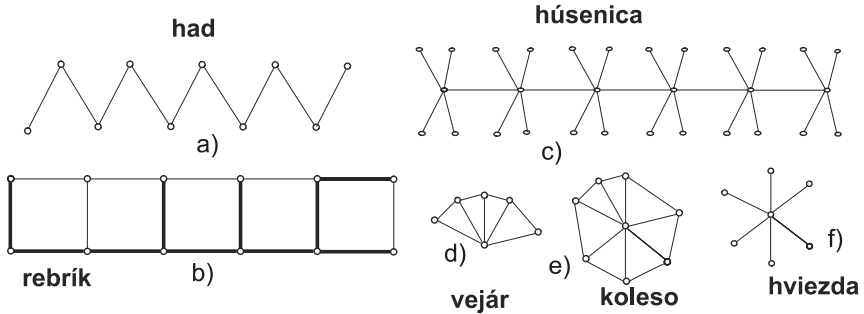


Obr.5.9 Strom a kostra grafu

Platí takáto veta: Nech je daný ľubovoľný súvislý graf  $G = [V, H]$ . Potom existuje jeho faktor  $G_1 = [V, H_1]$  ktorý je strom. Takýto faktor sa nazýva *kostra* grafu  $G$ . Teda podgraf určitého grafu, ktorý je stromom, sa nazýva *kostra*. Zavedenie tohoto pojmu je isto prospešné, ale nebudeme sa mu venovať podrobnejšie. Pripomeňme len, že kostier

k niektorému grafu môže byť viacero, napr. na obr. 5.9c sú silnejšími čiarami vyznačené dve z množstva možných kostier grafu podľa obr. 5.9b. Podobne je naznačená jedna z možných kostier v grafe podľa obr. 5.10b.

Na obr.5.10 sú ukážky niektorých grafov, ktoré získali svoje mená z bežnej ľudskej reči a podoby vecí: *had*, *rebrík*, *húsenica*, *vejár*, *koleso* či *hviezda*. Takéto pomenovanie často pomáha rýchlo si vytvoriť vizuálny model grafu.



Obr.5.10 Rozličné typy grafov

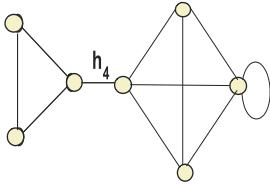
### Mosty grafu

Skôr ako prejdeme k orientovaným grafom, zastavme sa ešte pri ďalších pojmoch. Najprv si všimnime hranu grafu nazvanú *most*. Ide o takú hranu, ktorá nepatrí do žiadnej kružnice v ňom. Formulujme to takto: Zvoľme v grafe  $G$  jeho hranu  $h_s$ . Ak neexistuje v grafe  $G$  žiadna kružnica, ktorá obsahuje hranu  $h_s$ , potom hovoríme, že hrana  $h_s$  je *most* grafu  $G$ . Situáciu ilustruje obr. 5.11. Na ňom je nakreslený súvislý graf, ktorý nie je stromom; jeho hrana  $h_4$  ako vidno, nie je obsiahnutá v žiadnej kružnici a je teda mostom tohoto grafu (**most** spája napríklad dva komponenty grafu). Dalo by sa dokázať, že každá kostra konečného súvislého grafu obsahuje všetky mosty toho grafu.

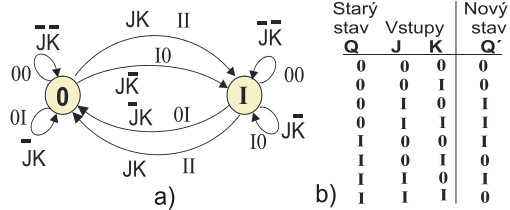
### Ohodnotený graf

Použitie grafov sa často viaže na riešenie praktických úloh, pri ktorých je vhodné pracovať s tzv. *ohodnoteným grafom*. Ten sa od predchádzajúceho typu líši tým, že sa ku každej hrane grafu priradí napr. nejaké číslo, alebo určitá hodnota (ktorá má pre používateľa istý žiadaný zmysel); potom hovoríme o *hranovo ohodnotenom grafe*. Podobne, ak v rovnakom zmysle priradíme aj vrcholom grafu nejaké číslo alebo istú hodnotu, potom analogicky hovoríme o *vrcholovo ohodnotenom grafe*. Ak sú ohodnotené aj vrcholy aj hrany grafu, ide

o **ohodnotený graf**. Príklady sú veľmi jednoduché: napr. ak graf predstavuje mapu miest nejakej krajiny, potom vrcholy takého grafu môžu byť ohodnotené menami miest na takejto mape, a hrany takého grafu, ktoré si možno predstaviť ako cesty spájajúce jednotlivé mestá, môžu byť ohodnotené číslami, ktoré predstavujú vzdialenosť v kilometroch medzi príslušnou dvojicou miest.



Obr.5.11 Súvislý graf



Obr.5.12 Príklad ohodnoteného grafu

Príklad ohodnoteného grafu je na obr. 5.12a. Tento graf predstavuje určitý automat (so vstupmi  $J, K$  – sú to logické premenné), ktorý má dva stavy  $0$  a  $1$ , pričom týmito hodnotami sú ohodnotené oba vrcholy grafu. Všetky možné prechody z jedného do druhého stavu (definované tabuľkou správania sa na obr.5.12b) sú v grafe zobrazené hranami, ktoré sú tiež ohodnotené veličinami, ktoré daný prechod vyvolávajú. Nie je ťažké presvedčiť sa pomocou tabuľky prechodov o tom, kedy a za akých podmienok sa uskutoční prechod zo „starého“ do „nového“ stavu, t.j. prechod z *počiatočného* vrcholu niektorej hrany do *koncového* vrcholu tej hrany. Slová *počiatočný* a *koncový* nás privádzajú k pojmu **orientovaná hrana**, ktorá šípkou graficky jasne vyznačuje smer postupu v grafe. V spomínanom obrázku sme ohodnotenie hrany urobili z dôvodu vyššej názornosti v dvoch podobách: k hrane sme priradili nielen algebraický výraz ale aj číselné hodnoty premenných (v algebraickom výraze). Napríklad hrana, ktorá zodpovedá tretiemu riadku tabuľky správania sa z obr.5.12b, vyjadruje prechod z vrcholu ohodnoteného *nulou* do vrcholu ohodnoteného *jednotkou*, ktorý sa uskutoční práve vtedy, keď má platnosť algebraický výraz  $\overline{JK}$ , alebo – inak povedané – vtedy, keď  $J = 1$  a  $K = 0$ . Oba tieto zápisy sme teda k príslušnej hrane pripísali (a tým sme hranu *ohodnotili*).

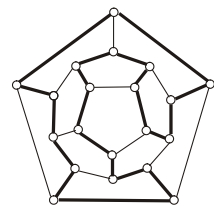
Všimnite si aj ďalšiu pozoruhodnosť: medzi dvoma vrcholmi v grafe nemusí existovať len jediná hrana, ale aj viacero hrán a to v závislosti od toho, aký proces daný graf vyjadruje. Tento fakt je natoľko významný, že *graf, v ktorom existuje medzi niektorými dvoma susednými vrcholmi viacero hrán, nazývame multigraf*.

Ďalší príklad ohodnoteného grafu je na obr. 3.6 (v kapitole Systém), kde sme vrcholom grafu správanía sa systému priradili isté stavy systému ( $Z_1, Z_2, Z_3$ ) a hrany sme ohodnotili symbolmi príčin, ktoré vyvolajú prechod z jedného do druhého stavu. Uvedené podoby ohodnoteného grafu sú potom veľmi dobrým prostriedkom na syntézu systému, ktorého správanie sa je grafom elegantne vyjadrené. A ešte dodajme, že aj vývojový diagram opisujúci nejaký výpočtový proces, napr. obr.1.2 (v prvej kapitole), je grafom v uvedenom zmysle: jeho vrcholy sú ohodnotené buď symbolmi istej definovanej operácie (takéto vrcholy majú podobu obdĺžnika), alebo rozhodovania (tie majú podobu kosoštvorca), a hrany potom určujú príslušné prechody na nadväzujúce vrcholy od počiatočného po koncový.

### Hamiltonovská kružnica

Všimnime si ešte jeden zaujímavý pojem, ktorý sa niekedy používa v súvislosti s teóriou hier, a to je hamiltonovská kružnica. W.R. Hamilton okolo roku 1859 vymyslel takú hru, že na vrcholy dvanásťstenu umiestnil mená niektorých svetových veľkomiest. Bolo treba – počnúc niektorým mestom – prejsť po hranách toho telesa všetky mestá tak, aby sme každým mestom prešli práve jedenkrát a vrátili sa do východzieho mesta. Zaujímavá je tu práve súvislosť s teóriou grafov: v jej terminológii to znamená, že treba nájsť v príslušnom grafe uzatvorený **faktor grafu typu kružnica**, ktorá obsahuje všetky vrcholy pôvodného grafu (a hrany v nej teda incidujú vždy len s dvoma vrcholmi). Môžeme formulovať toto: *Súvislý faktor druhého stupňa (tzv. kvadratický faktor) sa nazýva hamiltonovská kružnica.*

Príklad takej kružnice je v grafe na obr.5.13. Samotná hamiltonovská kružnica je vyznačená hrubšími čiarami. Možno sa presvedčiť, že sa dá nájsť viacero hamiltonovských kružnic v danom grafe (graf je totiž symetrický) a dá sa tiež určiť, koľko ich tu vôbec existuje.



Obr.5.13

Problém existencie hamiltonovských kružnic v grafoch je z všeobecného hľadiska pomerne zložitý a nebudeme ho tu riešiť.

Povedzme však, že ak graf obsahuje hamiltonovskú kružnicu, potom sa stručne nazýva *hamiltonovský*. Kvôli ucelenému náhľadu uvedme z rozličných bádání autorov aspoň Diracovu vetu o existencii hamiltonovskej kružnice [24]:

*Nech  $G$  je graf o  $n$  vrcholoch ( $n \geq 3$ ). Nech stupeň každého vrcholu je aspoň  $n/2$ . Potom  $G$  obsahuje hamiltonovskú kružnicu.*

## Orientované grafy

Teraz sa zamerajme na orientované grafy. Orientované im hovoríme preto, že hranám v takomto grafe pridáme orientáciu. Tu je definícia.

*Def. 5. 2. Orientovaný graf je trojica  $G = [V, H, \varepsilon]$  tvorená konečnou množinou  $V$ , ktorej prvky nazývame vrcholmi, konečnou množinou  $E$ , ktorej prvky nazývame orientované hrany, a zobrazením  $\varepsilon : H \rightarrow V \times V$ , ktoré nazývame vzťahom incidencie, a ktoré priraduje každej hrane  $h \in H$  usporiadanú dvojicu vrcholov  $\varepsilon(h) = (v_i, v_j) \in V \times V$ .*

Vrchol  $v_i$  nazývame *počiatočným vrcholom hrany  $h$* . Vrchol  $v_j$  nazývame *koncovým vrcholom hrany  $h$* . O oboch týchto vrcholoch hovoríme, že sú *incidentné s hranou  $h$*  a naopak, hrana  $h$  je incidentná s vrcholmi  $v_i, v_j$ . Obidva takéto vrcholy sú zároveň *krajnými vrcholmi hrany  $h$* . Ak hrana  $h$  inciduje s vrcholmi  $v_i, v_j$ , pričom  $v_i = v_j$  (ide o ten istý vrchol), potom hranu  $h$  nazývame *orientovanou slučkou*. Príklady takýchto **slučiek** opäť nájdeme v obr.5.12a aj 3. 6. Vrchol, ktorý nie je incidentný so žiadnou hranou, nazývame *izolovaný vrchol*.

Uvedomme si, že ak je **hrana orientovaná**, t.zn. že v nejakom vrchole  $v_i$  začína a vo  $v_j$  končí, potom zápis hrany v podobe  $v_i v_j$  predstavuje inú orientovanú hranu ako zápis  $v_j v_i$ . Preto sa stretne aj s takou podobou zápisu, ktorá používa šípky, napr.  $\overrightarrow{v_i v_j}$ , čo názornejšie vyjadruje orientáciu hrany. Z toho dôvodu sa tiež stretne s doplnením symbolu grafu pomocou šípky:  $\overrightarrow{G}$  potom značí, že ide o orientovaný graf. Analogicky ako pri neorientovaných grafoch zavedieme príslušné pojmy aj pre orientované grafy.

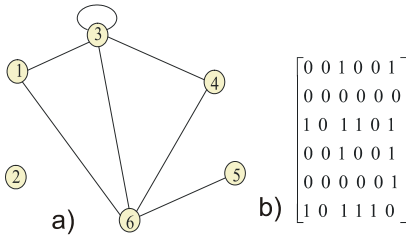
Počet hrán orientovaného grafu, ktoré vychádzajú z vrcholu  $v_i$ , t.j. počet hrán s počiatočným vrcholom  $v_i$ , sa nazýva *výstupný stupeň vrcholu  $v_i$* . Analogicky, počet hrán orientovaného grafu ktoré vstupujú do vrcholu  $v_i$ , t.j. počet hrán s koncovým vrcholom  $v_i$ , sa nazýva *vstupný stupeň vrcholu  $v_i$* . Súčet vstupného a výstupného stupňa vrcholu  $v_i$ , sa potom rovná počtu všetkých hrán incidentných s vrcholom  $v_i$ , pričom slučky počítame dvakrát. Počet hrán s počiatočným vrcholom  $v_i$  a koncovým vrcholom  $v_j$  sa nazýva *násobnosť orientovanej hrany*. Graf typu *strom*, ktorého všetky vrcholy majú vstupný stupeň rovný 1 (vstupuje do nich jediná hrana), sa nazýva *koreňový strom*. Jeho prvý vrchol je *koreň*.

Predpokladajme, že je daný orientovaný graf  $\overrightarrow{G} = [V, H, \varepsilon]$  a v ňom zvolíme vrcholy  $v_0$  a  $v_n$ . Ak existuje konečná postupnosť vrcholov a orientovaných hrán grafu  $\overrightarrow{G}$  ktorá má tvar

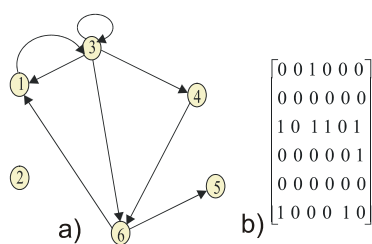


prvkov, ktoré určíme podľa takejto dohody: ak v grafe existuje hrana  $v_i v_j$ , potom nech prvok matice  $a_{ij} = 1$ , ak taká hrana neexistuje, nech  $a_{ij} = 0$ . Pre príklad grafu z obr.5.14a je takto vytvorená matica susednosti v obr. 5.14b. Možno sa presvedčiť, že matica je v takýchto prípadoch vždy symetrická vzhľadom na hlavnú diagonálu.

Presvedčte sa, že číslo  $a_{ij}$  v podstate predstavuje počet hrán, ktoré spájajú vrchol  $i$  a vrchol  $j$  (pre neorientované grafy je to priamo počet týchto hrán). Súčet nenulových prvkov v jednom riadku matice je rovný počtu hrán incidentných s príslušným vrcholom a je to teda *stupeň* tohoto vrcholu. Ďalej si uvedomte, že matica susednosti nepodáva o incidencii vrcholov a hrán žiadnu novú informáciu, je však zrejmé, že jednotková hodnota na hlavnej diagonále predstavuje slučku, vychádzajúcu a končiacu v tom istom vrchole. Je tiež jasné, že matica susednosti je počítačovou podobou príslušného grafu.



Obr.5.14 Graf a matica susedností



Obr.5.15 Orientovaný graf s maticou

Teraz uvažujme orientovaný konečný graf  $\vec{G} = [V, H]$ , v ktorom je poradie vrcholov ľubovoľne (ale pevne) zvolené, napr.:  $v_1, v_2, \dots, v_n$ . Vytvoríme štvorcovú maticu  $\mathbf{A}_2$  ( $m \times m$ ) prvkov, ktoré určíme podľa inej dohody: ak v grafe existuje orientovaná hrana  $\overrightarrow{v_i v_j}$  taká, že  $v_i$  je počiatočným vrchol a  $v_j$  koncovým vrcholom hrany, potom nech prvok matice  $a_{ij} = 1$ . Ak taká hrana neexistuje, nech  $a_{ij} = 0$ . Pre príklad grafu z obr.5.15a je takto vytvorená matica susednosti v obr. 5.15b. Počet jednotiek v jednom riadku matice udáva počet orientovaných hrán vychádzajúcich z príslušného vrcholu. Matica – ako vidno – stratila svoju symetričnosť voči hlavnej diagonále, napriek tomu je počítačovou reprezentáciou príslušného orientovaného grafu.

*Pozn.* Ak majú dva grafy rovnakú maticu susedností, sú navzájom izomorfné.

Do tretice uvažujme orientovaný konečný graf bez slučiek  $\vec{G} = [V, H]$  v ktorom množina vrcholov  $V = \{v_1, v_2, v_3, \dots, v_m\}$  a množina hrán  $H = \{h_1, h_2, \dots, h_n\}$  (sú ľubovoľne

–ale pevne– usporiadané). Potom možno vytvoriť maticu  $\mathbf{A}_3$  ( $m \times n$ ) prvkov, ktoré určíme takýmto predpisom:

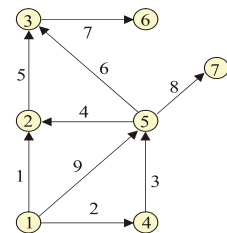
$$a_{ij} = \begin{cases} 1, & \text{ak } v_i \text{ je počiatočným vrcholom hrany } h_j, \\ -1, & \text{ak } v_i \text{ je koncovým vrcholom hrany } h_j, \\ 0, & \text{v ostatných prípadoch.} \end{cases} \quad (5.5)$$

Takúto maticu nazývame *incidenčná matica*. Na obr. 5.16 je príklad orientovaného grafu, pre ktorý vytvoríme **incidenčnú maticu** podľa stanoveného predpisu (5.5). Ukážme však názornejšiu podobu matice v podobe tabuľky, v ktorej vidno čísla vrcholov

hrany vrcholy	1	2	3	4	5	6	7	8	9
1	1	1	0	0	0	0	0	0	-1
2	-1	0	0	-1	1	0	0	0	0
3	0	0	0	0	-1	-1	1	0	0
4	0	-1	1	0	0	0	0	0	0
5	0	0	-1	1	0	1	0	1	1
6	0	0	0	0	0	0	-1	0	0
7	0	0	0	0	0	0	0	-1	0

Tab. 5.1 Incidenčná matica  $\mathbf{A}_3$  k príkladu

(v záhlaví riadkov) a čísla hrán (v záhlaví stĺpcov), preto je vizuálna orientácia v nej pohodlnejšia. Matica pre uvažovaný príklad je v Tab. 5.1. Nie je štvorcová a možno ľahko zistiť, že súčet kladných jednotiek v niektorom riadku je rovný počtu orientovaných hrán, ktoré vychádzajú z príslušného vrcholu, zatiaľ čo súčet záporných jednotiek v jednom riadku, je – až na znamienko – zhodný s počtom hrán, ktoré do príslušného vrcholu vstupujú.



Uvedené počítačové reprezentácie grafov nie sú jediné, existuje mnoho ďalších. Všetky však slúžia na využitie algoritmov, Obr.5.16 Graf k príkladu ktoré s rozličnou mierou účinnosti uľahčujú spracovanie úloh, v ktorých je graf prostriedkom analýzy či syntézy určitých systémov, ktoré by človek manuálne vôbec časovo či objemovo nezvládal. Rýchlosť a pamäťové nároky takých algoritmov však v tejto práci nie sú predmetom nášho záujmu.

## 6. HRY A ICH STRATÉGIA

Je zaujímavé, že niektoré stránky konfliktných situácií (ktorých je v bežnom živote veľa), sa dajú pomerne dobre modelovať a potom analyzovať a skúmať prostredníctvom hier. To je asi dôvodom toho, že pre matematickú teóriu konfliktných situácií sa zaužíval názov teória hier, ktorá analyzuje a využíva vlastnosti matematického modelu. Vo všeobecnosti matematickým modelom nazývame formálny opis niektorého javu, procesu alebo vzťahu reálneho sveta matematickými prostriedkami [23 ] (t.j. rovnicami, grafmi a pod.). Povedali sme to už v kapitole o systéme a jeho modeli.

**Teória hier** je teda matematická disciplína, ktorá skúma také rysy hier, ktoré možno opísať matematickým modelom. Riešením hry rozumieme určenie najvýhodnejšieho spôsobu hry každého účastníka z hľadiska jeho záujmov a cieľov.

**Hra je proces**, v ktorom sú charakteristické tieto rysy:

1. Je určený počet účastníkov hry. Tých nazývame **hráči**.
2. Všetci hráči poznajú a rešpektujú **pravidlá hry**, ktoré určujú začiatok a koniec hry a tiež určujú vyhodnotenie možných výsledkov hry pre každého hráča.
3. Hráč môže v priebehu hry **vybrať jednu z niekoľkých možností** konania, ale bez informácie o tom, ako budú konať protivníci.
4. Niektorí (prípadne všetci) hráči sa snažia **dosiahnuť svoj cieľ** (vyhrať).
5. Ciele hráčov sú (čiastočne alebo úplne) **protikladné**.

Ak niektorý z uvedených rysov hry neexistuje, (napr. niektorý hráč nedodržuje pravidlá alebo inak narušuje priebeh hry), stráca hra svoj zmysel a v skutočnosti prestáva byť hrou.. Z hľadiska dosahu však žiadna všeobecná charakteristika hry nevyčerpá mnohostrannosť každého konkrétneho prípadu, no uvedené vety možno úspešne považovať za prijateľnú definíciu hry pre väčšinu bežných hier. Poznamenajme ešte, že za hru v uvedenom zmysle možno považovať aj vojnové stretnutie, v ktorom každej z bojujúcich strán ide o dosiahnutie čo najväčšej výhody (napr. získanie nepriateľského územia). Podobne sa možno dívať aj na súdny proces, v ktorom stojí proti sebe obžaloba a obhajoba. Takýchto príkladov možno uviesť mnoho, ale zovšeobecňme to do tejto podoby: *Konfliktnou situáciou alebo hrou v širokom slova zmysle nazveme každú situáciu alebo proces, ktoré vyhovujú uvedenej definícii.*

Každá hra má svoj priebeh, t.j. má svoj **začiatok** i svoj **koniec**. Konkrétnu realizáciu priebehu od začiatku hry po niektoré z možných ukončení hry nazývame **partia** hry. Konkrétny výber niektorej z možností volieb hráča v priebehu partie nazývame **ťah hráča**.

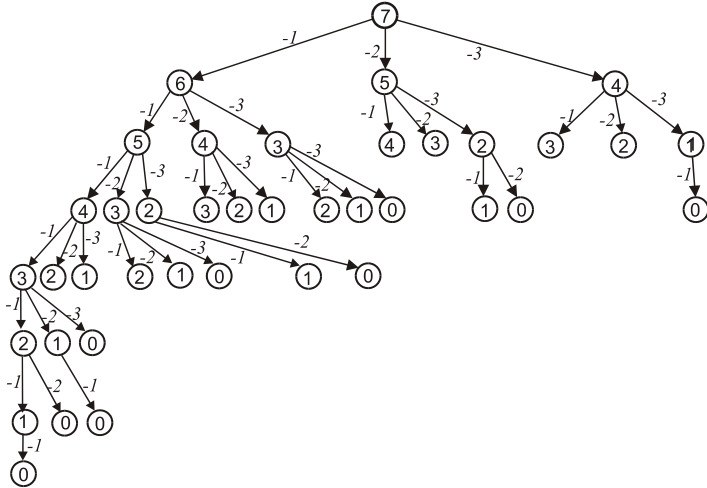
Pri dosahovaní svojho cieľa hráč spravidla musí v partii uvažovať o rôznych možnostiach konania účastníkov hry. *Premyslený postup hráča na dosiahnutie svojho cieľa* (vo všeobecnosti treba tento postup realizovať v závislosti od postupu iných hráčov) sa nazýva **stratégia**. Hráč si obvykle môže ešte pred začatím partie (t.j. *základnej jednotky hry*) premyslieť rôzne stratégie a vybrať si z nich tú, ktorú považuje za najvhodnejšiu pre konkrétne ťahy hráča z hľadiska jeho cieľov. Často sa podarí určiť takú stratégiu, ktorá zaručuje hráčovi najpriaznivejší výsledok (aj pri „najzlomyseľnejšej“ stratégii ostatných hráčov) zo všetkých možných stratégií. Takúto stratégiu potom nazývame **optimálna stratégia**. Pri určovaní optimálnych stratégií používajú najmä metódy teórie grafov, klasickej algebry a aritmetiky, dnes sa už ale využívajú aj počítače.

Viacere hry závisia od parametrov, ktoré môžu nadobúdať ľubovoľne veľké celočíselné hodnoty. Problém určenia optimálnej stratégie potom patrí medzi tzv. **hromadné problémy**. Z teórie je známe, že takéto problémy môžu byť **algoritmicky neriešiteľné** alebo (v lepšom prípade) **NP-úplné**. To posledné znamená, že na ich riešenie asi nikdy nenájdeme dostatočne rýchly algoritmus. V prvom prípade určite nenájdeme vôbec nijaký algoritmus. Aj keď sa nám nepodarí nájsť optimálnu stratégiu, netreba celkom rezignovať. Pre mnohé zaujímavé, i keď zložité hry sa podarilo objaviť aspoň nejaké rozumné taktiky, ktoré môžu byť prospešné pri hľadaní východísk v riešení konfliktných situácií.

Pre hľadanie optimálnej stratégie je názorným prostriedkom orientovaný graf typu strom, tzv. **strom hry**. Je to koreňový strom, v ktorom každý vrchol predstavuje konkrétny okamih hry (stav hry, **pozíciu**) a prislúchajúca orientovaná ohodnotená hrana zobrazuje prechod do nasledujúceho stavu hry podľa toho, akú voľbu hráč urobí. Zdôraznime, že nás tu zaujímajú len **konečné hry**, ktoré sú zobraziteľné konečným stromom.

Na ilustráciu uvažujme známu jednoduchú hru **NIM** so siedmimi zápalkami. Hrajú dvaja hráči: začína *Prvý*, pokračuje *Druhý* a ďalej sa pravidelne striedajú. Hráč ťahá tak, že zo spoločnej kôpky zápaliiek odoberie najmenej 1 a najviac 3 zápalky. *Vyhráva hráč, ktorý odoberie poslednú zápalku*. Na obr. 6. 1 je strom tejto hry. Vrcholy sú ohodnotené číslom, ktoré udáva počet zápaliiek na kôpke, takže najvyššie položený vrchol 7 predstavuje začiatok hry. *Prvý* hráč môže odobrať jednu, dve alebo tri zápalky, t.j. z vrcholu 7 vedú tri hrany,

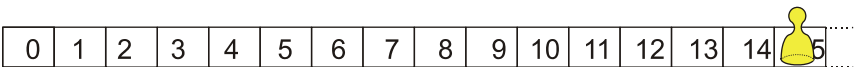
ohodnotené číslom, ktoré vyjadruje počet odobratých zápaličiek a tak možno v grafe prejsť do vrcholov 6, resp. 5, resp. 4 podľa voľby *Prvého*. Vyjadríme slovné, že sa takto *Druhý* môže dostať do *pozície* 6, resp. *pozície* 5, resp. do *pozície* 4. V grafe ľahko poznáme, do akých pozícií dostane potom *Druhý* po svojom ťahu *Prvého*. Ak je napr. priebeh hry taký, že



Obr. 6.1 Strom hry NIM7

prestavuje sled vrcholov a hráč 7,-3, 4, potom *pozícia* 4 je pre *Druhého* prehrávajúca, lebo *Prvý* sa nevyhnutne dostane do pozícií 1, resp. 2, resp. 3, ktoré sú pre neho vyhrávajúcimi pozíciami pri ľubovoľnom počte ním odobratých zápaličiek (v zmysle pravidiel).

Na obr.6.1 sme nemuseli kresliť úplný graf, lebo niektoré jeho podgrafy by sa zrejme opakovali, čo by mohlo byť na pohľad príliš neprehľadné. Napriek tomu možno pomocou uvedeného obrázku odvodiť priebehy všetkých možných partií hry. (Každá cesta od počiatočného po koncový vrchol v grafe predstavuje jednu partiu). Pomerne ľahko zistíme, že *pozície* 4 a samozrejme 0 (*nula*) sú prehrávajúce, ostatné možno považovať za vyhrávajúce, pokiaľ hráč neurobí chybu. Nespome je strom hry dobrým prostriedkom hľadania vhodnej stratégie hry. Uvedomme si však, že napriek jednoduchosti uvedenej hry je jej strom dosť zložitý. Čím zložitejšia je hra, tým prudšie rastie aj zložitosť príslušného grafu, takže v mnohých prípadoch (napríklad pri šachu) nemožno strom hry vôbec rozumne použiť.



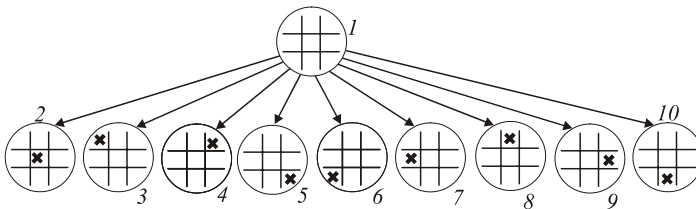
Obr. 6.2 Lištová podoba hry

Aby sme naznačili, že sa stratégia dá uplatniť aj inak, použijeme tzv. lištovú podobu uvedenej hry NIM. Na obr.6. 2 je nakreslená lišta, ktorej políčka sú očíslovaná číslami pozícií v zmysle predchádzajúceho stromu hry. Uvažujme, že „zápaliek môže byť i viac než sedem“ (aby sme hru trochu zovšeobecnil). Tentoraz predpokladáme, že namiesto zápaliek, stojí na príslušnej pozícii na lište figúrka. Hra začína polohou figúrky na pravom konci lišty (na najväčšom čísle pozície) a obaja hráči striedavo posúvajú figúrku smerom doľava o toľko políčok, koľko zápaliek by odobrali vo verzii so zápalkami – t.j. o 1, 2 alebo tri. Dá sa dokázať, že za týchto predpokladov možno označiť pozície ...16, 12, 8, 4, 0, ako pozície jasne prehrávajúce. (To samozrejme nemôže byť na lište zviditeľnené – lebo hra by stratila herný náboj). Ďalej uplatňujeme takéto pravidlá:

1. ak sa z pozície  $X$  dá ťahať iba do pozícií vyhraných pre súpera, tak  $X$  je prehrávajúca pozícia,
2. ak sa z pozície  $X$  dá ťahať aspoň do jednej pozície prehranej pre súpera, tak pozícia je vyhrávajúca.

Vidno, že lištová podoba hry umožňuje oveľa ľahšie hľadanie stratégie hry, je ale zrejmé, že rozbor a dôkaz sa principiálne opäť viaže na príslúchajúci strom hry. Obe podoby hry sú izomorfné (grécky: *iso* = rovnaký, *morfé* = tvar).

Uvedieme teraz príklad ďalšej hry dvoch hráčov, nazývanej *Tri v rade* alebo *Tic-tac-toe*. Ide o to, aby v štvorci (v mriežke) s deviatimi políčkami, do ktorých dvaja hráči striedavo umiestňujú svoje symboly (napr. jeden hráč *krížik* a druhý *krúžok*), vznikol rad (*riadok*,



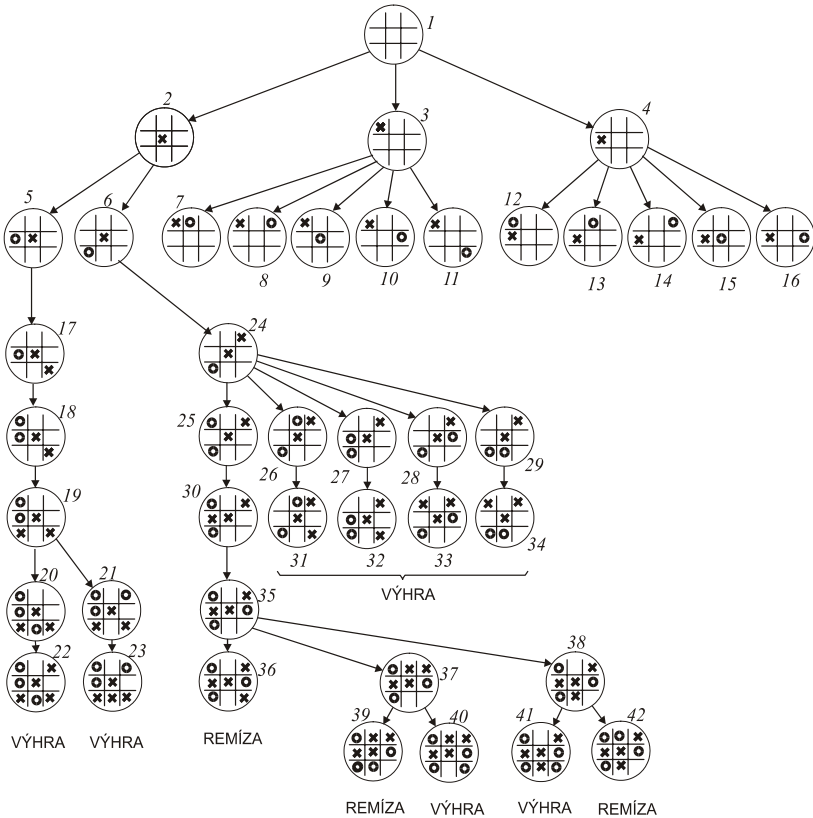
Obr. 6.3 Začiatok stromu hry

*stĺpec, uhlopriečka*) s tromi rovnakými symbolmi. Vyhráva ten hráč, ktorý to urobí prvý.

Na obr.6.3 je nakreslený začiatok stromu tejto hry. Vrcholy tohoto grafu sú ohodnotené priamo stavom obsadenia jednotlivých častí v mriežke s deviatimi políčkami, takže sledovanie je veľmi názorné a ľahko kontrolovateľné, ak si ešte vrcholy grafu viditeľne očísľujeme. Vrchol 1 ukazuje prázdnu mriežku a je zrejmé, že ten hráč, ktorý hru začína

napríklad s krížikom, má deväť možností jeho položenia do mriežky. Z vrcholu 1 vychádza teda deväť hrán do nadväzujúcich vrcholov grafu. To naznačuje, že úplný strom hry bude zrejme rozsiahly. Úspešne ho však možno redukovať už od začiatku rozboru hry, ak si uvedomíme, že mriežka je vlastne symetrický útvar, v ktorom stačí uvažovať toto:

1. krížik je uprostred mriežky (vrchol 2),
2. krížik je v rohu mriežky (vrcholy 3,4,5,6),
3. krížik je uprostred okrajového riadku alebo stĺpca mriežky (vrcholy 7,8,9,10).



Obr. 6.4 Významná časť stromu

Případy 2 resp. 3 predstavujú – vzhľadom na symetriu mriežky – len jej pootočené, takže z hľadiska rozboru hry pri hľadaní stratégie stačí z vrcholu 1 grafu uvažovať len tri prechody, ako to naznačuje horná časť obr.6. 4. Ostatné sú v nich už implicitne obsiahnuté.

Všimnime si ďalej priebeh hry od vrcholu 4. Protihráč bude ukladať *krížok* a má na to osem možností. V grafe však z vrcholu 1 vedie len päť hrán do vrcholov 12 až 16. To preto, že opäť „funguje“ symetria: napr. vrchol 12 platí aj pre symetrický prípad s *krížkom* v ľavom dolnom rohu, vrchol 13 aj pre prípad s *krížkom* uprostred spodného riadku, vrchol 14 aj pre prípad s *krížkom* v pravom dolnom rohu. Nakreslené jednoduchšie riešenie teda postačuje.

Podobne pre prechod z vrcholu 3. Protihráč má opäť možnosť položiť *krížok* do ôsmich pozícií, ale v grafe je opäť len päť hrán. To preto, že vrchol 7 predstavuje aj symetrickú situáciu s *krížkom* uprostred ľavého stĺpca mriežky, vrchol 8 zase symetrickú situáciu s *krížkom* v dolnom ľavom rohu, vrchol 10 aj symetrický stav s *krížkom* uprostred spodného riadku mriežky. Logická úvaha opäť vedie k redukcii počtu vrcholov stromu, čím sa graf ďalej zjednodušil. Nebudeme však pokračovať v uvedených vetvách (ponecháme to na čitateľa), zameriame sa na priebeh hry od vrcholu 2, a chceme ukázať, aké stratégie treba uplatniť, aby sme pri takomto postupe hry neprehrali.

Skúsme teraz, či na dosiahnutie nášho cieľa nestačí rozbor jednoduchšej časti stromu hry, ktorá vznikne po umiestnení prvého *krížika* a prvého *krížku*, t.j. rozbor časti stromu, nadväzujúcej na vrchol 2. Lahko sa presvedčíme, že ak nebude v tomto prípade *krížok* umiestnený do rohu mriežky (vrchol 5), bude mať začínajúci hráč možnosť vyhrať, ak umiestni druhý *krížik* do pravého spodného rohu, pozri vrchol 17, od ktorého nasleduje postupnosť vynútených ťahov. Analogicky možnorobiť rozbor aj v druhom prípade, keď je *krížok* umiestnený do rohu (vrchol 6), čo sa viaže na trochu zložitejšiu časť stromu. Začínajúci hráč si tu môže zabezpečiť to, že neprehrá.

Podarilo sa nám nájsť riešenie, ktoré udáva, ako má začínajúci hráč postupovať, aby nikdy neprehral. Pracovali sme však len s vybranou časťou stromu, takže nemôžeme zatiaľ povedať, že to platí aj pre úplný strom tejto hry. Nemožno preto tvrdiť, že pre túto hru vždy existuje pre niektorého z hráčov stratégia, ktorá mu zaistí víťazstvo. Z hľadiska dosiahnutého výsledku sme síce našli vhodnú stratégiu hry začínajúceho hráča, ale bez analýzy úplného stromu alebo nejakého vhodného dôkazu nevieme, či nájdenná stratégia je v uvedenom zmysle najlepšia. *Analogické tvrdenia platia aj pre mnohé iné hry.*

Ak sme optimálnou stratégiou nazvali premyslený postup hráča, potom v uvedenej hre možno rozborom stromu nájsť viacero stratégií, podľa vetiev grafu, ktoré vedú k žiadanému cieľu. Ak by sa hráčovi podarilo postupovať v hre tak, že prejde od vrcholu 1 grafu cez vrcholy 2,5,17,18,19,21 po vrchol 23, potom takáto stratégia je víťazná. Podobne

sled 1, 2, 6, 24, 29, 34 je realizáciou víťaznej stratégie. Sled 1, 2, 6, 24, 25, 30, 35, 36 vedie len k remíze.

Domnievame sa, že je už zrejmé, prečo sa robí rozbor stromu hry: treba odhaliť vhodnú stratégiu, zapamätať si ju a použiť v pravý čas v konkrétnej situácii. To isto nie je jednoduché. Rozbor stromu, ktorý nám pomáha pri rozhodovaní v danej pozícii, je zrejme založený na myšlienkových pokusoch, pri ktorých si predstavujeme dôsledky rozhodnutí: vyberieme v danej pozícii jednu z možností a pozíciu, ktorá takto vznikla, hodnotíme z hľadiska protihráča rovnakými úvahami, ako pri svojej pozícii. Tento kolobeh sa opakuje, pokiaľ neprídeme do koncovej pozície.

### Skupiny počítačových hier

V súvislosti s realizáciou hier počítačmi môžeme hovoriť o takýchto skupinách hier:

- **Simulátory.** Ide o hry, ktoré simulujú činnosť nejakého prístroja. Vzhľad obrazovky je prispôsobený tak, že zodpovedá vizuálnej podobe, s ktorou sa v skutočnosti stretávame pri práci s takýmto prístrojom. Tento typ hier úzko súvisí s činnosťami, ktoré sa realizujú na trenažéroch, aké sa v praxi skutočne používajú. Sú to simulátory automobilové, letecké, riadiace apod.
- **Akčné hry.** Podstata uplatnenia tejto skupiny programov je v akcii a protiakcii. Uvádza sa niekoľko druhov takýchto hier. *Bojové hry* simulujú boj niekedy tak verne, že sprievodné krvavé scény vôbec nie sú vhodné pre deti a mládež. Iná časť hier založených na bojových situáciách je zas z prostredia východných bojových umení. Ďalšia skupina programov uplatňujúcich bojové prvky je spojená s príbehom, počas ktorého musí hlavná postava prekonávať určité prekážky, alebo (v niektorých hrách tohto typu) sa charakter postavy mení v závislosti od obtiažnosti prekážok, ktoré sa naučí prekonávať.
- **Logické hry.** Tie sú založené na logických postupoch, ktoré treba aplikovať. Takéto hry rozvíjajú logické myslenie, nútia hráča tvorivo rozmyšľať nad každým nasledujúcim krokom.
- **Strategické hry.** Do tejto skupiny zaraďujeme hry, v ktorých hráč musí postupovať podľa určitej stratégie. Dej hry sa napríklad odohráva v neznámom prostredí, akoby v reálnom čase (napríklad simulácia života vznikajúceho mesta). Prostredie zobrazované v hre má svoje zákonitosti, počítač simuluje náhodné deje, aké sa môžu

vyskytnúť v tomto prostredí, a tiež situácie, ktoré zákonite vznikajú ako reakcie na kroky hráča.

Iná skupina strategických hier je založená na reakcii na ťah hráča a odvetve protihráča, alebo počítača. Sem patria aj šachy, ktoré majú vo svete počítačových hier svoje nezastupiteľné miesto.

Medzi strategické hry zaradujeme aj hry založené na simulácii reálnych ekonomických stratégií.

- **Adventúry.** Sú to hry, ktoré rozvíjajú nejaký príbeh. Niektoré sú spracované podľa deja opísaného v dobrodružnej literatúre a ak hráč pozná kľúčový dej, vie sa v hre orientovať. Takmer vždy v pozadí adventúry prebieha príbeh s dobrodružným dejom a hráč je v úlohe ústrednej postavy. Časť týchto hier je úzko spojená s bojovými situáciami a pri niektorých nevieme presne určiť, či patria do skupiny akčných hier alebo adventúr.
- **Vzdelávacie hry.** Sú to hry, pomocou ktorých sa hráč vzdeláva v širokej oblasti odborov, napr. hry na výučbu jazykov, alebo hry simulujúce isté fyzikálne zákonitosti, či hry na počítanie matematických príkladov, hry zemepisné a mnohé ďalšie.

Poznamenajme znovu, že hľadanie stratégie akejkoľvek hry nie je vo všeobecnosti jednoduché, v mnohých prípadoch je až nemožné. V nasledujúcich jednoduchých príkladoch naznačíme možné formalizmy, ktoré dajú aspoň orientačné údaje pre rozhodovanie hráčov.

## 6. 1 Solitérne hry

Takéto hry sú vlastne matematické hlavolamy, ktoré sa snaží riešiť jediný hráč. Úspešné vyriešenie hlavolamu sa interpretuje ako výhra tohoto jediného hráča.

Z teórie algoritmov je známe, že nemôže existovať algoritmus na riešenie všetkých matematických hlavolamov. Preto pre každú solitérnu hru treba hľadať riešenie špecifickou metódou.

### 6.1.1 Loydova „hra 15“

Americký šachový skladateľ a výtvarník Samuel Loyd [22] bol známy ako autor mimoriadne originálnych šachových úloh, ale tiež ako konštruktér rôznych nešachových hračiek, hádaniek, matematických a mechanických hlavolamov, ktoré si dokonca dával aj patentovať.

V solitérnej hre **15** (pre jedného hráča) sa v škatuľke  $4 \times 4$  políček nachádza 15 očíslovaných (od 1 do 15) predmetov tvaru štvorcíka  $1 \times 1$  a jedno políčko je teda prázdne. Na prázdne pole môžeme zo susedného poľa presunúť očíslovaný predmet. Cieľom hry je spravidla dosiahnutie konfigurácie znázornenej na diagrame 1.

Na začiatku hry sú očíslované predmety v škatuľke rozložené hocijakým spôsobom. Potom sa snažíme dovoľenými presunmi vytvoriť pozíciu na diagrame 1. Svoj postup môžeme aj jednoducho zapisovať: každý presun zaznamenáme číslom presúvaného predmetu.

**Príklad 6.1.** Majme konfiguráciu znázornenú na diagrame 2. V tejto pozícii možno vykonať jeden zo 4 ťahov: 3, 6, 10 alebo 14. Dobre zapamätateľná stratégia funguje tak, že najprv presunieme č. 1, 2, 3 a 4 (v tomto poradí) na ich cieľové polia. To sa dosiahne sekvenciou ťahov 14, 7, 15, 1, 12, 4, 8, 2, 10 (pozri diagram 3), 14, 7, 15, 1, 11, 3, 10 (pozri diagram 4), 2, 8, 4, 12, 11, 3, 10 (pozri diagram 5), 2, 8, 4, 12, 10 (pozri diagram 6), ďalej

Diagram 1

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

Diagram 2

9	6	5	13
10		14	7
2	3	11	15
8	4	12	1

Diagram 3

9	6	5	13
	14	7	15
10	3	11	1
2	8	4	12

Diagram 4

9	6	5	13
14	7	15	1
	10	3	11
2	8	4	12

Diagram 5

9	6	5	13
14	7	15	1
2		10	3
8	4	12	11

Diagram 6

9	6	5	13
14	7	15	1
8	2		3
4	12	10	11

2, 8, 4, 12, 10, 11, 3, 2 (diagram 7), 8, 4, 14, 9, 6, 5, 13, 1, 2, 3 (diagram 8), 11, 8, 4, 7, 5, 13, 1, 2, 3 (diagram 9), 4, 7, 5, 13, 1, 2, 3, 4 (diagram 10), 7, 5, 14, 9, 6, 1, 2, 3, 4 (diagram 11).

Diagram 7

9	6	5	13
14	7	15	1
4	8		2
12	10	11	3

Diagram 8

6	5	13	1
9	7	15	2
14	4	8	3
12	10	11	

Diagram 9

6	13	1	2
9	5	15	3
14	7	4	
12	10	8	11

V ďalšej etape analogicky presunieme č. 5, 6, 7 a 8 na ich cieľové políčka bez porušenia polohy č. 1, 2, 3 a 4. To sa dosiahne sekvenciou ťahov 15, 13, 6, 9, 14, 5, (pozri

Diagram 10

6	1	2	3
9	13	15	4
14	5	7	
12	10	8	11

Diagram 11

1	2	3	4
6	13	15	
9	14	5	7
12	10	8	11

Diagram 12

1	2	3	4
9	6	13	15
14	5		7
12	10	8	11

diagram 12), 13, 6, 5, 14, 9, 5, 6 (diagram 13), 15, 7, 13, 8, 11, 13, 8, 15, 7, 8 (diagram 14). Takýmito systematickými presunmi sa dá z každej začiatočnej pozície vytvoriť buď cieľová pozícia z diagramu 1, alebo pozícia z diagramu 15. S využitím bežného algebraického aparátu (počet inverzií permutácie) sa však dá dokázať, že naša základná pozícia (diagram 2, ale aj diagram 15) bola neriešiteľná. My tento dôkaz nebudeme robiť, zdôraznime však, že v tejto hre je pravdepodobnosť úspechu asi 0,5. Príklad je teda aj ukázkou toho, že pri hre nemusí vždy „existovať“ výhra.

Diagram 13

1	2	3	4
5	6		15
9	14	13	7
12	10	8	11

Diagram 14

1	2	3	4
5	6	7	8
9	14	15	
12	10	11	13

Diagram 15

1	2	3	4
5	6	7	8
9	10	11	12
13	15	14	

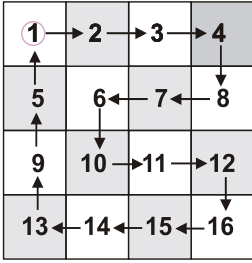
### 6.1.2. Hry s aplikáciou Hamiltonovej kružnice

Veľmi veľa hlavolamov sa viaže na šachovnicu [26]. Uvedme niektoré ukážky. Na obr. 6.5a je matica  $4 \times 4$  očíslovaných políčok, a treba nájsť také spojenia políčok, aby sme prešli od vybraného počiatočného políčka všetky políčka iba raz, a skončili opäť v počiatočnom políčku. To je typická úloha pre hľadanie hamiltonovskej kružnice v grafe. Isto vám nerobí problém predstaviť si jednotlivé políčka matice ako vrcholy grafu a spojnice políčok považovať za hrany. Ľahko teda zistíte, že riešenie na uvedenom obrázku je korektné a hamiltonovská kružnica tu existuje. Rovnako vás netreba presviedčať, že riešenie je tu celý rad, lebo už len počiatočných vrcholov môže byť šesťnásť.

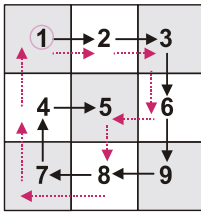
Modifikáciou uvedenej úlohy je prípad, keď počiatočný a koncový vrchol sú rôzne. Asi najvtipnejší hlavolam takéhoto typu publikoval vyššie spomínaný S. Loyd pod názvom *Marťanské kanály* v [ 22 ], spolu s množstvom iných zaujímavých hlavolamov.

**Príklad 6.2.** Úlohou je prejsť uzavretou cestou šachovnicu  $3 \times 3$ . Vrcholmi sú políčka šachovnice, hrany spájajú políčka so spoločnou stranou.

*Tentoraz však Hamiltonova kružnica neexistuje.* Dôkaz je jednoduchý: Keby existovalo riešenie hlavolamu (t.j. keby existovala Hamiltonova kružnica), muselo by pozostávať z 9 ťahov (podobne ako v predchádzajúcom príklade to bolo 16 ťahov). Predstavme si šachovnicu  $3 \times 3$  ofarbenú bežným spôsobom ako v obr.6.5b. Každý ťah vedie z bieleho poľa na čierne alebo naopak. Preto cesta z 9 ťahov začínajúca napríklad na bielom poli musí končiť na čiernom poli. Existuje tu ale tzv. **Hamiltonova cesta** (keď počiatočný a koncový vrchol môžu byť rôzne), napríklad  $1-2-3-6-9-8-7-4-5$ . Takýchto ciest existuje viac, ale každá z nich začína a končí na čiernom poli (podľa rohov šachovnice) a pozostáva z 8 ťahov. V obrázku je čiarkovane nakreslený aj príklad chybného riešenia (cesta je uzavretá, ale neprechádza políčkom 9).



a)



b)

c)

50	59	48	33	22	31	12	5
41	34	51	58	13	6	23	30
60	49	40	47	32	21	4	11
35	42	57	52	7	14	29	24
56	61	46	39	20	25	10	3
43	36	53	64	15	8	17	28
62	55	38	45	26	19	2	9
37	44	63	54	1	16	27	18

Obr. 6.5 Hlavolamy

**Príklad 6.3.** Zostaňme ešte pri šachovnici. Na obr. 6.5c je nakreslená bežná šachovnica so 64 políčkami. Úlohou je prejsť postupne všetky políčka ťahmi šachového jazdca tak, aby sa v ceste vyskytlo každé políčko len raz a aby cesta bola uzavretá. Zase je to úloha na hľadanie hamiltonovskej kružnice v zodpovedajúcom grafe. Úloha má viacero riešení, jedno je naznačené na obr. 6.5c a je zaujímavé tým, že políčka (chápané ako vrcholy zodpovedajúceho grafu) sú očíslované tak, aby bolo vidno postup v príslušnej ceste a pritom bola zachovaná názornosť ťahov jazdcom. Hustota čiar je výrazná, preto obrázok neobsahuje celé riešenie, čitateľ si môže doplniť chýbajúce hrany a skúsiť si tak hľadať ďalšie časti kružnice.

### 6.1.3. Hra SOKOBAN

Táto hra je typickou počítačovou verziou hlavolamu. Hracia doska má tvar štvorca alebo obdĺžnika ľubovoľných rozmerov a v bežne dostupnej počítačovej verzii je to obdĺžnik 19x16 políčok zaberajúci celú obrazovku. Počítačový program SOKOBAN majú k dispozícii mnohé slovenské základné školy a záujmové krúžky. Program umožňuje používateľovi editovanie vlastných pozícií, overovanie nových pozícií aj riešenie hotových pozícií. Na doske sú rozmiestnené objekty nasledujúcich typov:

- a) aktívna figúrka – je iba jedna, jej polohu v poli možných pozícií ovládame kurzorom klávesnice počítača,
- b) múr – nehybný objekt, slúži ako prekážka pre pohyblivé objekty,
- c) škatuľa – pohyblivý objekt, uvádzaný do pohybu tým že ho pred sebou tlačí aktívna figúrka,
- d) cieľové polia – rafinovane umiestnené políčka, na ktoré treba škatule postupne dotlačiť vhodným obchádzaním prekážok.

Aktívna figúrka môže ťahať 4 smermi (doľava, doprava, nahor alebo nadol), a to buď „naprázdno“, alebo s tlačeníím jednej škatule. Cieľom hry je *popresúvať všetky škatule na vopred vyznačené cieľové polia*. V zadaní môže byť cieľové pole prázdne, ale môže na ňom stáť aktívna figúrka alebo škatuľa (samozrejme nie múr!).

Písomný spôsob opisu priebehu hry a jej jednotlivých ťahov je veľmi nenázorný a nevystihuje dynamiku počítačovej hry *Sokoban* – pripomína zložitú a neprehľadnú vizualizáciu opisu hry **15**. To najmä preto, že v základnom poli existuje veľa prázdnych pozícií, ktoré spôsobia vznik „falošných“ ciest, ale by viedli aj k zložitým podobám grafu hry. Navyše ide o NP-problém. Preto nie je cieľom tejto kapitoly venovať sa ďalej

podrobnostiam spomínanej hry. Odporúčame čitateľovi skúsiť si zahrať hru *Sokoban* naživo s počítačom a hľadať tak rozumné taktiky, aby sa aspoň vyhol typickým chybám.

## 6.2 Hry typu NIM

Túto skupinu hier tu uvádzame ako typických reprezentantov tzv. **antagonistických strategických hier dvoch hráčov** s konečnou množinou možných pozícií. Slovo *antagonistický* tu znamená, že jeden hráč môže vyhrať len za cenu, že druhý prehrá. Už v úvode 6. kapitoly sme hru NIM použili na ilustráciu základných pojmov a postupov, umožňujúcu názorne predviesť metódy určovania, prípadne aj grafickej reprezentácie optimálnych stratégií. Preto tu uvedieme iba niektoré formalizmy pri definovaní hry (a okrem iného) pre pomenovanie prvkov na kope použijeme všeobecnejšie označenie **kamene**.

### Jednokopový NIM

Nech  $k, n$  sú celé kladné čísla. Hra *NIM I*( $k, n$ ) znamená toto: máme jednu kopy, na ktorej je  $k$  kameňov. Hrajú dvaja hráči: začína A, pokračuje B a ďalej sa pravidelne striedajú. Hráč ťahá tak, že z kopy odoberie najmenej 1 a najviac  $n$  kameňov. Kopa je „spoločná“! Vyhráva hráč, ktorý odoberie posledný kameň. Rozbor tejto hry sme urobili v úvode kapitoly pre hru *NIM I*(7,3) a nebudeme ho tu opakovať. Pripomeňme však, že sme tam zaviedli aj pojem *pozícia* (pozri napr. obr. 6.1, resp. 6.2) ako číslo  $K(z)$ , kde  $z$  je aktuálny počet zostávajúcich kameňov na kope. Pre hru *NIM I*(\*,3) sme odhalili pozície  $K(0), K(4), K(8), K(12), \dots$  ako pozície jasne prehrávajúce. (Hviezdička \* tu označuje ľubovoľný základný počet použitých hracích kameňov).

**Poznámka.** Pravidlá hry *NIM I* by sme mohli modifikovať tak, že hráč, ktorý odoberie posledný kameň, prehráva. Takéto modifikácie bývajú pri hrách typu *NIM* spravídla rozumné a zaujímavé. V riešení však táto modifikácia nedáva nič nového, len „posunutie stratégie o 1“ – napríklad pre modifikáciu predošlého príkladu *NIM I*(\*,3) dostaneme prehrané *pozície*  $K(1), K(5), K(9), K(13), \dots$ .

### Dvojkopové a viackopové hry NIM

Dvojkopová hra  $NIM 2(k_1, k_2, n)$  znamená, že na dvoch kopách je  $k_1$ , resp.  $k_2$  kameňov. Hráči  $A, B$  sa striedajú na ťahu ako vyššie. Hráč ťahá tak, že si vyberie kopu a z nej odoberie najmenej 1 a najviac  $n$  kameňov. Vyhráva ten hráč, ktorý odoberie posledný kameň.

Ako vidno, princíp je rovnaký ako vyššie, a preto nebudeme tu takéto hry zoširoka opisovať a to ani pre viackopové hry, chceme však využiť fakt, že dvojkopová hra  $NIM 2(*, *, I)$  dovoľuje ukázať tzv pseudohru.

Pri tejto hre zistíme, že žiadny hráč nemôže ovplyvniť sumárny počet kameňov po svojom ťahu (nanajvýš ich rozdelenie medzi obe kopy). Ak by sme zobrazili pozície tejto hry, tak sa ukáže, že pozícia  $K(z_1, z_2)$  je teda prehrávajúca práve vtedy, ak číslo  $z_1 + z_2$  je párne. Výsledok je daný začiatočným počtom kameňov v kopách a hráči ho (dokonca ani keby spolupracovali) nemôžu vôbec ovplyvniť. Preto  $NIM 2(*, *, I)$  je tzv. **pseudohra**. Pri pseudohrách nemá zmysel hľadať stratégie. Niekedy však až v procese hľadania stratégie zistíte, že ide o pseudohru. Viackopové hry  $NIM$  spravidla nie sú pseudohry a ich stratégie sa opisujú rozmanitými prostriedkami.

### 6.3 Maticové hry

Sú to hry dvoch hráčov, pri ktorých ide o to, že obaja hráči majú nezávisle na sebe možnosť výberu z niekoľkých postupov (ani jeden hráč nemá informácie o postupe súpera) a podľa tzv. **výplatnej matice** (tiež sa používa pojem **výplatná tabuľka**) sa určí výhra, resp. prehra hráčov v práve uskutočnenej partii. Úplná teória maticových hier je matematicky náročná a vedie až k poznatkom z lineárneho programovania. Tu sa chceme zaoberať iba prípadom tzv. **hier s nulovým súčtom**, pri ktorých výhra jedného hráča znamená rovnakú (v absolútnej hodnote) prehru druhého hráča a naopak.

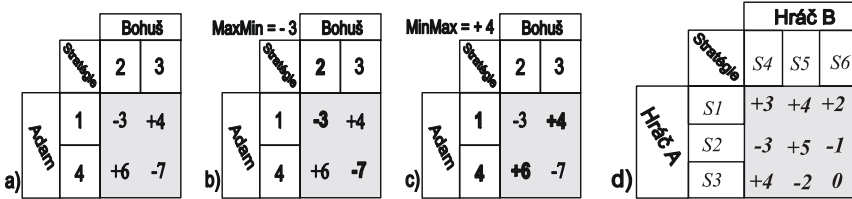
Matica sa zostavuje tak, že riadky zodpovedajú možným postupom prvého hráča a stĺpce zodpovedajú možným postupom druhého hráča. Ak sa v matici vyskytne napríklad hodnota 4, bude to znamenať výhru  $+4$  pre prvého hráča a  $-4$  pre druhého hráča.

**Poznámka.** Hry s nulovým súčtom automaticky znamenajú, že pri každej partii „usporiadateľ“ buď získa alebo musí doplatiť rozdiel od nulového súčtu. Potom však už nejde o serióznu matematickú hru, lebo hráči sa môžu dopredu dohodnúť na postupe, ako usporiadateľ a hry čo najviac „ozbijať“. Pri hrách s nulovým súčtom to nehrozí, lebo hráči

môžu vyhrať iba na škodu svojich súperov. Preto sme už vyššie označili takéto hry ako **antagonistické**.

Začnime jednoduchým príkladom nasledujúcej hry: Hru budú hrať dvaja hráči *Adam* a *Bohuš*. Na pokyn rozhodcu **napíšu** nezávisle od seba na kus papiera (alebo tabuľu a pod., dôležité je, aby si na napísané navzájom nevideli) **jedno číslo**. *Adam* môže napísať číslo *1* alebo *4*. *Bohuš* môže napísať číslo *2* alebo *3*. Ak je súčet napísaných čísel párny, vyhráva *Adam*. V opačnom prípade vyhráva *Bohuš*. **Výška výhry je rovná súčtu napísaných čísel**.

Ukážme, ako pre túto hru zostaviť výplatnú maticu. Pre názornosť napíšeme maticu v takej názornej podobe, aby bolo vidno aj záhlavia riadkov a stĺpcov pre oboch hráčov, ako na *obr. 6.6a*. V záhlaví riadkov budú postupy hráča *Adama*, (v našom prípade *1* alebo *4*), v záhlaví stĺpcov budú postupy hráča *Bohuša* (v našom prípade *2* alebo *3*). V ďalšom budeme hovoriť aj takto: ak hráč *Adam* zvolí číslo *1* alebo *4*, povieme, že zvolil *stratégiu1* resp. *stratégiu4* (a zapíšeme to *S1* resp. *S4*), analogicky to budeme hovoriť aj o *Bohušovej stratégiu2* resp. *stratégiu3* (a zapíšeme *S2* resp. *S3*). Prvok matice potom do nej zapíšeme tak, že výhra hráča *Adama* (t.j. súčet čísel zvolených hráčmi) má *kladné* znamienko + ( a *záporné* znamienko -, ak vyhrá *Bohuš*.):



Obr. 6.6 Výplatné matice

*Adam* má na výber dve tzv. čisté stratégie: *stratégiu1* a *stratégiu4*. Nevie však, akú stratégiu zvolí *Bohuš*. Ak bude vždy počítať s tou možnosťou, ktorá je pre neho najmenej výhodná, tak *stratégia1* mu dáva výhru  $-3$  a *stratégia4* mu dáva výhru  $-7$ . Analogicky však môže uvažovať aj *Bohuš* (jeho výhry sú rovné prehrám hráča *Adama*). *Bohušova stratégia2* dáva výhru  $-6$  (*Adam stratégiu4* vyhráva  $+6$ ) a *stratégiu3* vyhráva  $-4$ . Tento príklad názorne ukazuje, že hráč *Adam* si musí všímať najmenšie čísla v riadkoch a z nich hľadať najväčšie (teda maximum z miním), zatiaľ čo hráč *Bohuš* si musí všímať najväčšie čísla po stĺpcoch a z nich hľadať najmenšie (teda minimum z maxim). V našom príklade to bude pre *Adama* vyzerat' tak, ako ukazuje *obr. 6.6b*. Hodnota maxima z miním je **MaxMin = -3**.

Pre *Bohuša* je situácia zobrazená na obr. 6.6c. Minimum z maxim je **MinMax** = + 4.

Na prvý pohľad to vyzerá čudne, lebo obaja hráči prehrávajú (hodnota min max = +4 znamená, že hráč *Bohuš* vyhráva -4).

V nadväznosti na uvedené pojmy sa zavádza pojem **sedlový bod**, podľa nasledujúcej definície:

**Definícia 6.1.** Hovoríme, že matica má **sedlový bod**, ak pre ňu platí **MaxMin** = **MinMax**.

*Poznámka.* Dá sa matematicky dokázať (tzv. veta o minimaxe), že pre každú maticu platí

$$\text{MaxMmin} \leq \text{MinMax}. \quad (6.1)$$

O sedlovom bode teda hovoríme vtedy, ak v uvedenej nerovnosti nastáva rovnosť. Sedlový bod predstavuje tiež tzv. *rovnovážnu situáciu* alebo rovnovážny stav.

Ukážme teraz aspoň formou príkladu, ako možno nájsť stratégie pre **hry so sedlovým bodom**.

**Príklad 6.4.** Zamerajme sa na hru ktorej výplatná matica je na obr. 6.6d.:

Pre hráča *A* dostaneme  $\text{MaxMin} = \max \{+2, -3, -2\} = +2$ , pre hráča *B* podobne

$$\text{MinMax} = \min \{+4, +5, +2\} = +2, \text{ z čoho plynie, že výplatná matica}$$

má sedlový bod. Dostávame ho v riadku *S1* a v stĺpci *S6*. To znamená, že optimálna stratégia hráča *A* je *S1* a optimálna stratégia hráča *B* je *S6*. Hráč *A* má zaručenú výhru +2 (pri opustení stratégie by riskoval -3, resp -2). Hráč *B* má zaručenú výhru -2 (pri opustení stratégie by riskoval -5, resp. -4). Takto jednoznačne vychádzajú optimálne stratégie vždy, keď výplatná matica hry má sedlový bod.

Pre **hry bez sedlového bodu** je to trochu zložitejšie, ale bez zachádzania do matematických podrobností prezradíme, že ak výplatná matica hry nemá sedlový bod, optimálna stratégia má formu **náhodného výberu** z čistých stratégií pri určitom konkrétnom pomere pravdepodobnosti výberu jednotlivých čistých stratégií. Pritom niektoré čisté stratégie v skutočnosti vôbec nemusia byť použité. Vzhľadom na rozsah takej úlohy a pravdepodobnostné výpočty, nebudeme v tomto texte podrobnejšie rozoberať tento problém a odkážeme čitateľa na odporúčanú literatúru.

V živote sa väčšinou stretávame s problémami, ktoré nemajú antagonistický charakter, a preto môže byť užitočné uviesť tu aj príklad **neantagonistických hier**, ktorými sa situácia dá zobrazovať či modelovať.

**Príklad 6.5.** Uvažujme o dvoch hráčoch *A* a *B*. Nech každý z nich má dve stratégie *S* (spolupracovať) a *Z* (zradiť). Hra sa nazýva **väzňova dilemma** [25], lebo v jej pôvodnej forme šlo o dvoch navzájom izolovaných väzňov obvinených zo spoločného zločinu, ktorí stoja

pred rozhodnutím, či spolupáchatel'a prezradiť, alebo mlčať. Každý musí voliť svoju stratégiu bez toho, aby poznal stratégiu spoluhráča. Bez ohľadu na to, čo urobí druhý hráč, prináša zrada väčšiu výhodu ako spolupráca. Pritom však zrada obidvoch vedie k horšiemu výsledku ako spolupráca obidvoch.

Pomocou týchto matic ľahko zistíme, že dvojica tzv. čistých stratégií [ Z,Z ] tvorí rovnovážnu situáciu. Ak hráč B uplatňuje stratégiu Z, potom prechodom od stratégie Z k stratégii S by si hráč A zhoršil svoju výplatu z 2 na 0. Ak hráč A uplatňuje

		Výplaty hráča A		Výplaty hráča B	
		B		B	
Stratégie		S	Z	S	Z
		A	S	5 0	5 9
Z	9 2		0 2		

Obr.6.7 Matice výplat oboch hráčov

stratégiu Z, tak aj hráč B by si prechodom od stratégie Z k stratégii S zhoršil svoju výplatu z 2 na 0. Rovnako ľahko zistíme, že dvojica [ S,S ] nie je rovnovážnou situáciou. Z matice výplat hráča A zistíme, že pri stratégii S hráča B si hráč A prechodom od stratégie S k stratégii Z zvýši svoju výplatu z 5 na 9. Z druhej matice výplat zistíme, že aj hráč B by si prechodom od stratégie S k stratégii Z zvýšil svoju výplatu z 5 na 9. Z obidvoch matic je však zároveň zrejmé, že obaja hráči dávajú prednosť nerovnovážnej situácii [ S,S ] pred rovnovážnou [ Z,Z ], lebo dvojica stratégií [ S,S ] vedie pre obidvoch k väčšej výplate ako dvojica stratégií [ Z,Z ].

Tento príklad naznačuje, že teória neantagonistických hier (čo len dvoch hráčov) je oveľa zložitejšia ako teória antagonistických hier. Dokazuje to napríklad fakt, že vo všeobecnosti sa rovnovážne situácie nedajú považovať za riešenie v neantagonistických hrách. Príklad ďalej ukazuje, že pokiaľ pravidlá hry umožňujú spoluprácu, mali by sa hráči o ňu usilovať, napr. preto, aby sa hráč A vyhol situácii [ S,Z ], hráč B situácii [ Z,S ] a obidvaja situácii [ Z,Z ].

Uzavríme uvedený výklad nasledujúcou poznámkou: Ak sa v neantagonistickej hre pripúšťa spolupráca hráčov, hovoríme o **kooperatívnych hrách**, v opačnom prípade hovoríme o **nekooperatívnych hrách**.

### 6.4 Hry proti prírode

Hra proti prírode sa obvykle definuje ako maticová hra dvoch účastníkov s nulovým súčtom, v ktorej jeden účastník **nekoná cieľavedome**. Poznamenajme však, že za účastníka takejto hry možno považovať aj prírodu alebo nejaký nevedomý systém. O hre proti

prírode hovoríme aj vtedy, ak je účastníkom hry mysliaci hráč, ktorý je z nejakého dôvodu *lahostajný* k výsledku hry.

Zdôraznime, že vyššie uvedený *minimaxový princíp* riešenia hry predpokladá cieľavedomé konanie obidvoch účastníkov, takže vo väčšine prípadov v hre proti prírode, nie je vhodným nástrojom na určenie takého konania účastníka. Môžeme ho však použiť vtedy, ak chceme nájsť stratégiu, ktorá zaručí minimálne riziko, tj. najlepšiu stratégiu hráča proti preňho najhoršej stratégii prírody.

**Príklad 6.6.** K chirurgovi privezú pacienta, ktorý mal ťažký úraz a silno krváca. Chirurg musí preveriť celkový zdravotný stav pacienta a pre náš príklad predpokladajme, že existujú dva stupne jeho zdravotného stavu : označme ich  $S_1$  a  $S_2$ . Ďalej uvažujme, že lekár môže použiť dva rôzne postupy operácie, tie označme  $P_1$  a  $P_2$ . Potom možno zostaviť

		Stav	
		$S_1$	$S_2$
LEKÁR	$P_1$	0,7	0,5
	$P_2$	0,8	0

maticu, ktorú sme vyššie označovali ako *výplatnú*. Je na obr. 6.8. Prvky matice však tu nie sú hodnoty výplat, ale *pravdepodobnosti*, s ktorými bude daný spôsob operácie úspešný pri zdravotnom stave  $S_1$  alebo  $S_2$  pacienta. Potom maticu výplat nazývame aj *matica pravdepodobností*. Otázkou je, kto určil alebo vypočítal konkrétne hodnoty použitých

Obr. 6.8. Matica pravdepodobností

pravdepodobností. Zrejme sú uvedené hodnoty určené na základe dlhodobu získavaných skúseností a pokusov lekárov a uvádza ich najmä príslušná vedecká lekárska literatúra. Ktorú alternatívu operačného zákroku zvolí v opísanej situácii chirurg ? Chirurg sa chce zabezpečiť proti najhoršej možnosti, riziko tu predstavuje smrť človeka. Preto zvolí čistou stratégiu  $P_1$  podľa *minimaxového* princípu (matica výplat má sedlový bod  $(P_1, S_2)$ ).

**Príklad 6.7.** Majiteľ bufetu v horskej chate objednáva mlieko pre detských návštevníkov. Reálne množstvo predaného mlieka závisí od počasia v onen deň. Ak je počasie ideálne, predá maximálne množstvo mlieka a zinkasuje maximálny zisk. Ak je počasie chladné, ale neprší, predá časť maximálneho množstva mlieka, ale zvyšok mu skysne a tým jeho zisk klesne. Ak dokonca prší, nepredá takmer nič a pri objednávke na maximálne množstvo by utrpel značnú stratu. Čo má robiť majiteľ bufetu?

Samozrejme, vždy môže majiteľ vychádzať z pesimistického odhadu, že bude pršať, čiže objednať minimálne množstvo mlieka. Ak však bude počasie priaznivé, bude mať zbytočne nízku tržbu, lebo mlieko rýchlo vypredá a ďalších záujemcov už nebude môcť uspokojiť. Tu sa hráč  $A$  (majiteľ bufetu) môže pridržať rôznych stratégií ako pri maticových hrách, podľa vzoru predchádzajúceho príkladu. Na orientáciu môže použiť napríklad ľudové pranostiky, predpovede počasia a podobne, aby aspoň približne odhadol príslušné hodnoty pravdepodobností do výplatnej matice. Ani jeden z takých postupov nie je spoľahlivý, a preto by hráč  $A$  mal vyskúšať rôzne stratégie a vyhodnocovať svoje skúsenosti.

## 7. KRIMINALITA V INFORMATIKE

Problematika počítačovej kriminality začína byť tak závažná, že sa už častejšie organizujú stretnutia a odborne orientované konferencie v tejto oblasti [28,29]. Témy sú stále vysoko aktuálne a zaujímavé, a preto ich tu v stručnosti uvedieme. Cieľom tejto kapitoly je poukázať na niektoré pojmy a myšlienky, ktoré majú podnietiť zvýšenie záujmu o túto oblasť.

Termíny *zneužívanie počítačov*, alebo *nesprávne používanie počítačov*, sú z hľadiska vzťahu ku kriminalite podstatne širšie, ako termín *počítačová kriminalita*. Ale ani pojmy *počítačová kriminalita* (computer crime), kriminalita týkajúca sa počítačov (computer related crime), *kriminalita v automatizovanom spracovaní dát* (ASD) a *kriminalita v informačných technológiách* nie sú doteraz presne definované, a často sa chápu ako synonymá. *Počítačová kriminalita* ako pojem, sa považuje za hospodársku kriminalitu, ale napriek tomu nie sú počítačové trestné činy iba čisto ekonomickou kriminalitou. Pojem počítačovej kriminality sa používa najmä v kriminalistických kontextoch, ale je užitočný v celom trestnoprávnom sektore. V niektorých krajinách sa pojem kriminalita v informačných technológiách používa aj pre trestné činy týkajúce sa uchovávaní a prenosu programov a dát.

Jednoduchá definíciu vytvorila skupina expertov OECD už v r. 1983 v podobe pojmu *zneužitie počítačov*, čím rozumeli každé nelegálne, neetické alebo neoprávnené konanie vo vzťahu k automatizovanému spracovaniu a prenosu dát. Ako vidno, je to široko chápaná definícia a má takpovediac medzinárodný význam.

Ukázalo sa ako dôležité rozlišovať medzi prísne právnou definíciou počítačovej kriminality a definíciou vytvorenou pre potreby výskumu – tá je užitočná pre systemizovanie a identifikáciu problémov vzťahujúcich sa najmä na trestné činy týkajúce sa počítačov. Stanoviť dobrú všeobecnú definíciu je teda ťažké, lebo každá doterajšia definícia sa vždy viazala v istom zmysle na konkrétny čin. Napr. vo Fínsku sa doteraz za *trestný čin v informačných technológiách považoval každý nelegálny čin, v ktorom je informačný systém, vrátane hardvéru, nástrojom, objektom alebo miestom trestného činu, a spáchanie ktorého vyžaduje expertné znalosti informačných technológií* [28]. Je to zaujímavá definícia, ktorá predpokladá u páchatel'a vysokú odbornosť, ale udalosti z praxe ukázali, že v mnohých prípadoch sa podvod podaril aj úplnému laikovi.

Globálna definícia teda ešte nebola vytvorená a Európsky výbor pre problémy kriminality pracujúci pri Rade Európy ponechal príslušným krajinám možnosť rozhodnúť o modifikácii funkčných klasifikácií v súlade so svojimi právnymi systémami a historickými

tradiíciami. S rozvojom informačných technológií uplatňovaných v informačnej spoločnosti rastie aj počet trestných činov viazucich sa na informačné siete. Preto by žiadna spoločnosť nemala zabúdať aj na právne úpravy a ošetrenia proti narušaniu jej infraštruktúry, a to nielen z hľadiska počítačovej kriminality, ale aj z hľadiska zneužívania sietí. Nové informačné technológie a ich aplikácie v praxi so sebou priniesli aj nové typy zneužívania, ktoré by mali byť naďalej adekvátne sledované a trestané stále aktualizovanou legislatívou.

### **7.1 Klasifikácia počítačových trestných činov**

Trestno-právny systém mnohých krajín je založený na ochrane právnych záujmov napadnutého. Preto väčšina kriminalistických a právnych štúdií o počítačovej kriminalite rozlišuje

- ◆ hospodárske trestné činy súvisiace s počítačmi a
- ◆ počítačové narušenia súkromia a útoky na iné právne chránené záujmy.

Pokiaľ ide o trestné činy v informačných technológiách, možno ich podľa ochraňovaných právnych záujmov zahrnúť do štyroch celkov:

- *ochrana súkromia,*
- *ochrana proti ekonomickým trestným činom,*
- *ochrana duševného vlastníctva a*
- *ochrana proti nelegálnemu a škodlivému obsahu.*

V období rokov 1970 –1980 sa prudko rozvíjali možnosti pre zhromažďovanie, uchovávanie a prenos dát pomocou nových technológií, preto sa nutne objavujú aj občianskoprávne a trestnoprávne normy, ktoré sú neustále novelizované, s cieľom chrániť právo občanov na súkromie. Podobne bola regulovaná aj ochrana osobných údajov v súvislosti s rozvojom telekomunikácií, najmä z hľadiska uchovania dôverností v elektronickej pošte. Analýzou rôznych prístupov trestnoprávných ustanovení v rôznych európskych krajinách možno stanoviť štyri hlavné kategórie trestnoprávných porušení:

- a) porušenie hmotných práv na súkromie,
  - b) porušenie formálnych právnych požiadaviek,
  - c) porušenie práv na prístup k informáciám a
  - d) zanedbanie bezpečnostných opatrení.
- a) *Porušenie hmotných práv na súkromie* predstavujú takéto činy:
- nelegálne zverejnenie, rozširovanie, získanie údajov a/alebo prístup k údajom,
  - nezákonné použitie údajov,

- nelegálne vkladanie, upravovanie a/alebo falšovanie údajov s úmyslom spôsobiť škodu,
  - zber, zaznamenávanie a/alebo uchovávanie údajov, ktoré sú z hmotných dôvodov nelegálne,
  - uchovávanie nesprávnych údajov.
- b) Porušenia formálnych požiadaviek predstavujú takéto činy:
- porušenia právnych požiadaviek na začatie spracovania osobných údajov (napr. registrácia, upozornenie, žiadosť o registráciu,
  - odmietnutie podať informáciu alebo uvedenie falošnej informácie nadriadeným orgánom,
  - marenie výkonu súdneho príkazu,
  - nevymenovanie kontrolóre ochrany údajov v spoločnosti,
  - nezaznamenanie dôvodov alebo prostriedkov rozširovania osobných údajov
- c) *Porušenie práv na prístup k informáciám* sa viaže na tzv. slobodu informácií, keď žiadateľ dostane nepravdivú informáciu aj keď má na pravdivú a úplnú informáciu nárok, alebo nedostane odpoveď na žiadosť.
- d) *Zanedbanie bezpečnostných opatrení* sa z hľadiska trestnoprávnych akcií začalo sledovať len pomerne nedávno, pričom zmyslom sledovania je vyvodzovať dôsledky z neplnenia nielen preventívnych opatrení.

## **7.2 Kriminalita vzťahujúca sa na počítače**

Zhruba pred dvadsiatimi rokmi ochraňovala trestná legislatíva predovšetkým hmotné a viditeľné objekty, no chrániť bolo treba aj netradičné formy (napr. peniaze v elektronickej podobe) aj nehmotné objekty, akými sú napr. počítačové programy a aj novovzniknuté funkcie a činnosti, (napr. nelegálna manipulácia s obsahom počítača). Teda nešlo napr. len o porušenie súkromia, ale priamo aj o ekonomickú kriminalitu týkajúcu sa počítačov. Také trestné činy sa podľa typu zatrieďujú do nasledujúcich skupín:

- ◆ počítačové hackovanie,
- ◆ počítačovú špionáž,
- ◆ počítačová sabotáž,
- ◆ softvérové pirátstvo a iné formy pirátstva produktov,
- ◆ počítačové falšovanie a podvody

## 7.2.1 Hackovanie

Neoprávnený prístup k programom alebo dátam uloženým v počítačovom systéme nazývame **hackovanie**. Hacker to robí len z potešenia, pre radosť z prelomenia ochranných bariér, pričom nemá úmysel poškodiť informačný systém alebo robiť špionáž. Ak by sa taký úmysel prejavil, ide o **crackera** a hovoríme o crackovaní. Rozdiel v oboch prístupoch môže byť značný, v prvom prípade ide len o narušenie formálnej sféry tajnosti a systém nebol poškodený. Ale aj takéto preniknutie do počítačového systému bolo už kvalifikované ako trestný čin.

Je zaujímavé, že niektoré krajiny považujú za trestné iba preniknutie do počítačového systému „zvonka“, t.j. z iných organizácií či z iných lokalít. Je však pravdou, že aj „domáci“ pracovník, ktorý má systém bežne prístupný, môže pri istých príležitostiach neoprávnene využiť informačný počítačový systém tak, že sa javí ako nelegálne a neoprávnené „kradnutie času“ zamestnancami „zvnútra“.

Popri tradičných spôsoboch hackovania, preniknutia do systému alebo zistenia hesla využitím používateľov, ktorí sú neopatrní pri narábaní s heslami, vznikli aj nové, agresívne procedúry prieniku do počítačových systémov. Bohužiaľ sa to netýka len klasických počítačových prostriedkov, ale aj stále sa rozvíjajúcich telekomunikačných prostriedkov a služieb.

## 7.2.2 Počítačová špionáž

Hlavným cieľom útokov pri **počítačovej špionáži** sú počítačové programy, výskumné a obranné diela, údaje komerčného účtovníctva a klientske informácie. Preto je počítačová špionáž mimoriadne nebezpečná, aj keď v najjednoduchšej podobe ide len o jednoduché kopírovanie. Uplatňuje sa však aj krádež médií s dátami, dokumentov, odpočúvanie a monitorovanie elektronických prenosových kanálov. Krádežou sa tu nemyslí len odnesenie príslušných fyzických prvkov, ale hlavne kopírovanie, alebo nedovolené zapožičanie, ktoré sa nepodarí vždy odhaliť.

## 7.2.3 Softvérové pirátstvo a iné formy pirátstva produktov

Nelegálne použitie a kopírovanie programov sa tiež nazýva krádežou softvéru alebo **softvérové pirátstvo**. Zväčša sa týka kopírovania takých programov, ktoré sú pre firmu dôležité najmä z hľadiska jej vnútornej činnosti, takže sa prejavuje vlastne aj ako počítačová špionáž. Stále je aktuálne pirátske kopírovanie štandardných programov na osobné počítače

a kvôli vysokej hodnote informácií dochádza tiež k nelegálnemu používaniu rozličných databáz. Ide tiež o neautorizované nahrávky zvukových kaziet a o videonahrávky.

Takéto trestné činy sa prejavujú vlastne ako porušenia práv duševného vlastníctva, ktorého využitie nie je povolené žiadnou licenciou. Na internete je kontrola výkonu práv na duševné vlastníctvo takmer nemožná a o jeho porušení sa vlastník často ani nedozvie. Poskytovanie služieb, používatelia a iní účastníci internetu sa - vzhľadom na celosvetový dopad internetu – môžu dopustiť aj niektorých ďalších trestných činov. V oblasti útokov na firemný majetok sa ich sledovanie dá robiť napr. zintenzívnením kontrol na hraniciach, ale kde sú hranice štátov v internete ?

#### 7.2.4 Počítačová sabotáž

Čo je predmetom počítačovej sabotáže ? Je to predovšetkým hardvér, softvér, dôležité dáta v elektronickej forme a informačné siete. Tu predstavuje počítačová sabotáž mimoriadne nebezpečenstvo, lebo k nej nerozlučne patrí úloha **spôsobiť fyzickú a logickú škodu**. Stále sa vyskytujú rôzne viac či menej škodiace programy – nazývané najčastejšie **vírusy** – (rôzne utility, trójske kone, červy a iné ničivé programy resp. využívajúce tzv. „bugy“, t.j. chyby hardvéru a softvéru, ktoré sa takmer nekontrolovateľne šíria počítačovými sieťami.

Sabotáž v spomínanej súvislosti môže viesť k priamemu ohrozeniu firmy, ba dokonca pri napadnutí rozsiahlych dátových súborov v štátnych, najmä centrálnych inštitúciách môže ohroziť bezpečnosť samotného štátu. Preto treba legislatíve, ošetrujúcej problematiku takéhoto typu sabotáže, venovať neustálu pozornosť a inovovať príslušné zákony v zhode s meniacimi sa podmienkami v praxi.

Stíhanie páchatel'ov, ktorí rozširujú na internete nelegálny a škodlivý obsah je však vždy komplikované. Zložitosť spočíva v tom, že páchatelia obvykle pôsobia v zahraničí a mechanizmy medzinárodnej spolupráce pri ich odhaľovaní nie sú vždy pružné. Najčastejšími prípadmi sú programy s prezentáciou pornografie, najmä detskej, ďalej prejavy podporujúce nenávisť a nactiufhačstvo, ba dokonca aj sexuálne obťažovanie. Ak sa k tomu pridá faktor anonymity, ktorý je v internete dosť výrazný, stíhanie páchatel'ov je takmer nemožné. Výmaz alebo blokovanie stránok s nelegálnym či škodlivým obsahom je v širokom merítku prakticky (technicky) nerealizovateľné.

#### 7.2.5 Počítačové podvody a falšovanie

V niektorých krajinách sa v tradičných ustanoveniach o falšovaní interpretoval fakt, že sa predpokladala vizuálna čitateľnosť informácií v dokumente. To zrejme nemožno

dosiahnuť pri elektronických formách zápisu dát či ukladania dokumentov (napr. obsah diskety sa priamo očami upretými na disketu čítať nedá), preto zákony musia takéto dokumenty chrániť rovnako, ako sú chránené tlačené dokumenty. Z tohoto dôvodu ustanovenia o integrite a správnosti informácií patria k najdôležitejším ustanoveniam o falšovaní.

Vo vyššie uvedených prípadoch išlo prevažne o trestné činy, v ktorých išlo o možnú ekonomickú škodu. Ale treba odlíšiť aj druhý prípad, keď sú používané falošné informácie s úmyslom napadnúť právne chránené záujmy inej osoby. V takýchto prípadoch hovoríme o **počítačovom podvode**. Najrozšírenejším druhom takéhoto podvodu je manipulácia s faktúrami, platby účtov, iných výplat, manipulácia s účtovnými výkazmi a bankovými operáciami, pričom takéto manipulácie možno – s ohľadom na prepojenia v sieťach – robiť „zvonku“ organizácie, ktorá sa stane obeťou podvodu.

Do tohoto druhu podvodov patria aj rozličné formy zneužívania kreditných kariet a analogických platobných prostriedkov, a ochrana správnosti a autenticity nových foriem peniaží sa stala súčasťou počítačového falšovania. Patria sem aj falošné inzeráty alebo zneužívanie telefónnych sietí. Právne zaujímavý je prípad, keď páchatel podvedie počítač.

### **7.3 Zhrnutie**

Väčšinu počítačových trestných činov môže páchatel urobiť bez toho, aby musel opustiť svoju kanceláriu alebo svoj domov, a – vzhľadom na medzinárodný charakter internetu - počítačová kriminalita potom môže prekračovať nielen hranice pracovísk, organizácií, mesta, regiónu, ale aj štátne a medzinárodné hranice, je teda počítačovou kriminalitou medzinárodnou. Jej vyšetrovanie je preto veľmi náročné a to nielen kvôli rozdielnym právnym úpravám v jednotlivých krajinách. Na úspešný boj s počítačovou kriminalitou treba organizovať účinnú a masívnu medzinárodnú spoluprácu, ktorá predpokladá vzájomné porozumenie problémov a ich možných riešení.

A tak už niekoľko rokov dochádza k mobilite príslušných informácií medzi zainteresovanými krajinami a k harmonizácii a koordinácii príslušných zákonov. Najväčšiu prácu v tejto oblasti vykonali OECD (Organizácia pre hospodársku spoluprácu a rozvoj), Rada Európy a Európska únia, OSN, a Organizácia svetového obchodu. Tento proces pokračuje a Slovensko je jeho účinným faktorom.

## PRÍLOHA A. ČÍSELNÉ SÚSTAVY

Prirodzených čísiel je nekonečne veľa, a preto nemožno každé z nich označiť samostatným symbolom. Každé číslo sa zapisuje pomocou dohovorených symbolov alebo znakov, ktoré vznikli historicky a nazývame ich **číslice**. V desiatkovej sústave poznáme **číslice 0, 1, 2, 3, 4, 5, 6, 7, 8, 9**.

Nech je dané číslo 5 689. Môžeme o ňom povedať, že je to prirodzené číslo, zapísané číslicami 5, 7, 8, 9. Predstavuje 5 tisícok, 7 stoviek, 8 desiatok a 9 jednotiek, pričom si ani neuvedomujeme, že vo vyjadrení čísla 5 789 je skrytý zápis výrazu

$$5 \cdot 1\,000 + 7 \cdot 100 + 8 \cdot 10 \cdot 1 + 9 \cdot 1$$

Pre mocniny desať platí

$$1\,000 = 10^3$$

$$100 = 10^2$$

$$10 = 10^1$$

$$1 = 10^0.$$

Dané číslo môžeme teda zapísať takto:  $5\,789 = 5 \cdot 10^3 + 7 \cdot 10^2 + 8 \cdot 10^1 + 9 \cdot 10^0$ .

Výraz tohoto typu nazývame **mnohočlen**. Číslice pri mocninách nazývame **koefficienty**, číslo **10** je **základ číselnej sústavy**. Základ tiež určuje aj názov sústavy: Ak je základom čísla *desať*, hovoríme o *desiatkovej (dekadickej) sústave* [16].

Všeobecne teda každé prirodzené číslo  $N$  môžeme v desiatkovej sústave zapísať takto:

$$N = a_n \cdot 10^n + a_{n-1} \cdot 10^{n-1} + \dots + a_1 \cdot 10^1 + a_0 \cdot 10^0$$

kde  $a_i$  ( $i = 0, 1, \dots, n$ ) je koefficient (v desiatkovej sústave je to jedna z číslic 0, 1, 2, 3, 4, 5, 6, 7, 8, 9),

$10$  - základ sústavy,

$i$  - hodnota, ktorá udáva poradie zodpovedajúceho koefficienta, alebo poradie miesta.

Poradie miesta určuje teda poloha číslice v danom čísle (nazývame rád  $i$  alebo  $i$ -tý rád). Napríklad v čísle 89 765 má číslica 9 rád 3.

Zápis v tvare mnohočlena je dlhý, a preto sa bežne nepoužíva. Obyčajte sa čísla píše iba zápisom koefficientov, pričom zapisujeme aj nulové hodnoty koefficientov. V skrátenom zápise musia byť koefficienty usporiadané podľa klesajúcich mocnín v zodpovedajúcom mnohočlene.

Doteraz sme hovorili o celých číslach. Rovnakým spôsobom sa však vyjadrujú aj desatinné čísla. Napríklad o čísle 0,3745 môžeme povedať, že predstavuje 3 desatiny, 7 stotín, 4 tisíciny a 5 desaťtisíciny.

To umožňuje zapísať číslo  $0,3745$  ako mnohočlen

$$0,3745 = 3 \cdot 10^{-1} + 7 \cdot 10^{-2} + 4 \cdot 10^{-3} + 5 \cdot 10^{-4}$$

Exponenty pri mocninách  $10$  sú záporné, preto hovoríme o *záporných rádoch* čísiel.

*Desatinná* (alebo všeobecnejšie *rádová*) *čiarka* oddeľuje záporné rády od nezáporných. V niektorých programovacích jazykoch alebo v anglosaských štátoch sa namiesto čiarky píše rádová bodka.

Celkovo môžeme povedať, že ľubovoľné číslo  $N$  s  $n + 1$  kladnými a  $m$  záporným rádmi môžeme v číselnej sústave so základom  $z$  zapísať ako mnohočlen

$$N = a_n \cdot z^n + a_{n-1} \cdot z^{n-1} + \dots + a_1 \cdot z^1 + a_0 \cdot z^0 + a_{-1} \cdot z^{-1} + \dots + a_m \cdot z^{-m} \quad (\text{A.1})$$

kde  $a_i$  ( $i = n, n-1, n-2, \dots, 1, 0, -1, -2, \dots, -m$ ) sú **koeficienty**.

V skrátenej tvare môžeme takéto číslo zapísať zápisom koeficienta (čísllice) takto:

$$a_n a_{n-1} a_{n-2} \dots a_1 a_0 , a_{-1} a_{-2} \dots a_m \quad (\text{A.2})$$

Číselné sústavy, ktoré možno vyjadriť podľa vzťahu (A.1), nazývame polyadické. Pre tieto sústavy platí

$$0 \leq a_i < z \quad (\text{A.3})$$

Povedali sme už, že hodnota základu sústavy určuje aj názov sústavy. Ak je základ sústavy  $z = 2$ , hovoríme o dvojkovej (binárnej) sústave, ak je  $z = 10$ , ide o desiatkovú (dekadickú) sústavu, ak je  $z = 8$ , ide o osmičkovú (oktálovú) sústavu a ak je  $z = 16$ , ide o šestnástkovú (hexadecimálnu) sústavu. V niektorých počítačoch sa používali aj sústavy so základom  $3$ , základom  $5$  alebo s iným základom.

### **A.1 Dvojková sústava**

*Dvojková sústava* je základom pre mnohé aplikácie v číselných počítačoch a je vhodná aj pre uplatnenie logických systémov pri realizácii počítačových operácií.

Podľa vzťahu (A.3) dosahujú koeficienty dvojkovej sústavy menšie hodnoty, ako je základ, teda  $a$  môže byť iba  $0$  alebo  $1$ . Tieto čísllice sa preto nazývajú dvojkové a z anglického názvu binary digit pre ne vznikol názov **bit**.

Dvojkové čísla sa zapisujú podobne ako desiatkové - zápisom koeficientov čísllic. Napríklad  $1\ 010,01$  je dvojkové číslo, ktoré možno vyjadriť mnohočlenom takto:

$$1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 + 0 \cdot 2^{-1} + 1 \cdot 2^{-2}$$

**Príklad A1.** Určite hodnotu mnohočlena pre dvojkové číslo  $10110,011$ . V akej sústave je táto hodnota vyjadrená?

Riešenie:  $1 \cdot 2^4 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^{-2} + 1 \cdot 2^{-3} = 16 + 4 + 2 + 0,25 + 0,125 = 22,375.$

Táto hodnota je vyjadrená v desiatkovej sústave. Môžeme teda povedať, že číslo *10110,011* je dvojkovým vyjadrením desiatkového čísla *22,375* a opačne.

**Príklad A2.** Rozhodnite, v akej sústave je zapísané číslo *11,01*.

Riešenie je v podstate nejednoznačné. Vzhľadom na to, že číslice *0* a *1* sú taktiež číslice z desiatkovej sústavy, (ale aj každej polyadickej sústavy so základom  $z \geq 2$ ), bez dodatočnej informácie nemožno určiť, v akej sústave je zapísané dané číslo. Preto sa pomocou indexu zhodného so základom sústavy vyjadruje v prípade nejasností pri každom čísle, v akej sústave je dané číslo zapísané. Pre príklad *A1* môžeme písať

$$10110,011_2 = 22,375_{10}.$$

Zaoberajme sa teraz problémom prevodu čísel medzi polyadickými sústavami s rôznym základom. Jednoduchý je prevod čísla zo sústavy s ľubovoľným základom do desiatkovej sústavy. Vykonáva sa tak, ako sme už naznačili: výpočtom hodnoty mnohočlena v desiatkovej sústave.

Zložitejší (ale nie ťažší) je opačný prevod. Teraz opíšeme prevod čísla z desiatkovej sústavy do dvojkovej sústavy. Na riešenie tejto úlohy existuje niekoľko metód. Najnázornejšia z nich je metóda odpočítania mocnín dvoch.

**Príklad A3.** Urobte prevod čísla *9* do dvojkovej sústavy.

V riešení hľadáme najbližšiu nižšiu mocninu dvoch, a to je  $2^3 = 8$ . V hľadanom dvojkovom čísle bude teda člen polynómu  $1 \cdot 2^3$ . Potom odpočítame nájdenú mocninu od daného desiatkového čísla a dostaneme rozdiel  $9 - 8 = 1$ , ku ktorému opäť hľadáme najbližšiu nižšiu mocninu dvoch. V našom prípade je rozdiel *1*, t.j.  $2^0$ , teda v hľadanom dvojkovom čísle bude člen polynómu  $1 \cdot 2^0$ . Po opakovanom vykonaní rozdielu  $1 - 1 = 0$  je výsledok nulový, prevod je teda ukončený. Mocniny  $2^2$  a  $2^1$  sa nevyskytli, teda hľadané číslo je vyjadrené mnohočlenom

$$1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0, \text{ ktorý skrátene zapíšeme } 1001. \text{ Platí } 9_{10} = 1001_2.$$

Rýchlejší je prevod **postupným delením** transformovaného čísla základom sústavy a zápisom zvyškov, ktoré priamo určujú koeficienty hľadaného čísla. Postup ukážeme na príklade.

**Príklad A4.** Urobte prevod čísla *19* do dvojkovej sústavy.

Riešenie:

$$\begin{array}{ccccccc}
 19 : 2 = 9 ; & 9 : 2 = 4 ; & 4 : 2 = 2 ; & 2 : 2 = 1 & \leftarrow & \text{posledný podiel} \\
 1 & 1 & 0 & 0 & & \text{je menší} \\
 \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \\
 a_0 & a_1 & a_2 & a_3 & a_4 & \\
 & & & & & \text{ako základ} \\
 & & & & & \text{sústavy}
 \end{array}$$

Môžeme písať :  $19_{10} = 10011_2$

Pri tomto riešení dostávame hodnoty koeficientov v opačnom poradí ako pri predchádzajúcom postupe. Číslu najvyššieho rádu dostaneme ako poslednú.

V uvedených príkladoch sme do dvojkovej sústavy transformovali iba celé čísla. Postup pri prevode desatinných čísiel je iný. Ukážeme ho na príklade.

**Príklad A5.** Urobte prevod čísla  $0,375_{10}$  do dvojkovej sústavy.

Riešenie urobíme nasledujúcim spôsobom. Na rozdiel od celých čísel teraz budeme postupne dané číslo **násobiť** základom sústavy. Po vynásobení rozdelíme číslo na celú a desatinnú časť. Formálny zápis môže byť napr. takýto:

0	, 3 7 5 × 2	Násobíme vždy len desatinnú
0	, 7 5 0 × 2	časť, takže číslu pred rádovou
1	, 5 0 0 × 2	čiarkou určujú priamo hľadané
1	, 0 0 0	dvojkové číslo.

samé nuly indikujú ukončenie prevodu

Môžeme teda napísať :  $0,375_{10} = 0,011_2$  .

## A.2 Osmičková sústava

Číslo  $N$  sa v osmičkovej sústave vyjadruje v zhode so vzťahom (A.1), príp. (A.2). Základ je  $z = 8$ . Prevod z osmičkovej sústavy do desiatkovej sústavy je jednoduchý a spočíva opäť len vo vyčíslení hodnoty mnohočlena.

**Príklad A6.** Urobte prevod čísla  $0, 1, 2, 3, 4, 5, 6, 7, 10, 11, 12$  z osmičkovej sústavy do desiatkovej. Riešenie je v Tab. A1.

Prevod z desiatkovej sústavy do osmičkovej sústavy je podobný ako prevod z desiatkovej sústavy do dvojkovej. Rozdiel je iba v tom, že pracujeme s mocninami ôsmich (pri desatinných číslach). Pracujeme v tej sústave, z ktorej robíme prevod, t.j. v desiatkovej sústave.

**Príklad A7.** Urobte prevod čísla  $139_{10}$  do osmičkovej sústavy. *Riešenie:*

	$139 : 8 = 17,$	$17 : 8 = 2$	← posledný podiel je menší ako základ sústavy
zvyšok po delení	3	1	
	↓	↓	
koeficient:	$a_0$	$a_1$ $a_2$	

Môžeme teda písať :  $139_{10} = (a_2 a_1 a_0)_8 = 213_8$  .

z = 2	z = 3	z = 5	z = 8	z = 10	z = 16
0	0	0	0	0	0
1	1	1	1	1	1
10	2	2	2	2	2
11	10	3	3	3	3
100	11	4	4	4	4
101	12	10	5	5	5
110	20	11	6	6	6
111	21	12	7	7	7
1000	22	13	10	8	8
1001	100	14	11	9	9
1010	101	20	12	10	A
1011	102	21	13	11	B
1100	110	22	14	12	C
1101	111	23	15	13	D
1110	112	24	16	14	E
1111	120	30	17	15	F
10000	121	31	20	16	10

Tab. A1. Vyjadrenie niektorých čísiel v rôznych číselných sústavách

### A. 3 Šestnástková sústava

Šestnástková sústava je pre človeka dosť nezvyčajná, ale v počítačoch je bežná. Formálne vyjadrenie ľubovoľného čísla vychádza rovnako ako pri predchádzajúcich sústavách zo vzťahov (A.1) a (A.2). Ak je však základ sústavy  $z = 16$ , koeficienty nadobúdajú podľa vzťahu (A.3) hodnoty  $0, 1, 2, \dots, 15$ . Číslice  $0$  až  $9$  budú v šestnástkovej sústave rovnaké ako v desiatkovej sústave. Každú zo šestnástkových číslic  $10, 11, 12, \dots, 15$  označíme jediným písmenom - v poradí  $A$  až  $F$  ( $A=10, B=11, C=12, D=13, E=14, F=15$ ).

Ak zapíšeme napr. číslo  $3AD6,19$ , znamená to, že zodpovedajúci mnohočlen je

$$3 \cdot 16^3 + A \cdot 16^2 + D \cdot 16^1 + 6 \cdot 16^0 + 1 \cdot 16^{-1} + 9 \cdot 16^{-2}$$

alebo  $3 \cdot 16^3 + 10 \cdot 16^2 + 13 \cdot 16^1 + 6 \cdot 16^0 + 1 \cdot 16^{-1} + 9 \cdot 16^{-2}$ .

Po vyčíslení tohoto mnohočlena môžeme písať

$$3 \cdot 4096 + 10 \cdot 256 + 13 \cdot 16 + 6 \cdot 1 + 1 \cdot 0,0625 + 9 \cdot 0,003962 = \\ = 12\,288 + 2\,560 + 208 + 6 + 0,0625 + 0,0351558$$

Z toho vyplýva  $3AD6,19_{16} = 15\,062,0976558_{10}$ .

Uvedený príklad zároveň ilustruje spôsob prevodu zo šestnástkovej sústavy do desiatkovej. Postup prevodu je rovnaký ako pri predchádzajúcom zápise. Tak isto sa nelíši

prevod celého čísla z desiatkovej sústavy do šestnástkovej. Postupným delením šestnástimi dostávame zvyšky (menšie ako 16), ktoré v poradí od rádu 0 (t.j. od koeficienta  $a_0$ ) určujú priamo hodnotu príslušného koeficienta. Rozdiel je iba v tom, že koeficienty väčšie ako 9 nahradzujeme dohovorenými symbolmi (A až F). Pri prevode desatinných čísel násobíme šestnástimi.

**Příklad A8.** Urobte prevod čísla 7 194 do šestnástkovej sústavy. Riešenie je toto:

	$7\ 194 : 16 = 449$	$449 : 16 = 28$	$28 : 16 = 1$	
zvyšok :	10	1	12	↓
symbol :	A	1	C	↓
	↓	↓	↓	↓
koeficient:	$a_0$	$a_1$	$a_2$	$a_3$
Platí teda	$7\ 194_{10} = ICIA_{16}$			

Doteraz sme vykonávali prevody iba medzi desiatkovou sústavou a ostatnými. Na rovnakom princípe možno však pretransformovať čísla medzi dvojkovou a osmičkovou sústavou, osmičkovou a šestnástkovou sústavou a pod. Dôležité je uvedomiť si, ktorá sústava je východisková, ktorá cieľová a ktorá sústava je nám bližšia (t.j. v ktorej vieme alebo chceme počítať). Podľa zvyklostí sa uvedené postupy aplikujú obyčajne v tej sústave, z ktorej transformujeme. Ak je to sústava s menším základom ako sústava do ktorej chceme pretransformovať, musíme zvyšky po delení (alebo celé časti po násobení) transformovať samostatne. Všeobecne však pri prevodoch potrebujeme poznať vykonávanie aritmetických operácií v týchto sústavách. Preto najprv preberieme tieto operácie.

### A. 4 Aritmetické operácie

#### Spočítanie

Pri spočítaní postupujeme v každej polyadickej sústave rovnako ako v bežnej desiatkovej sústave. Ak je pri spočítaní dvoch číslíc (koeficientov)  $i$ -tého rádu súčet väčší ako najväčšia číslica v danej sústave, zapíšeme do  $i$ -tého rádu nižšiu číslicu súčtu a pripočítame k vyššiemu rádu  $i+1$  jednotku. Takejto jednotke hovoríme **prenos do vyššieho rádu**. Presvedčte sa, že v žiadnej sústave nie je pri súčte dvoch sčítancov prenos väčší ako 1.

**Příklad A9.**  $8_{10} + 7_{10} = 15_{10}$

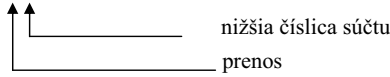
$\begin{array}{c} \uparrow \uparrow \\ \text{nižšia číslica súčtu} \\ \text{prenos} \end{array}$

$9_{16} + 6_{16} = 15_{10} = F_{16}$  ← nižšia číslica súčtu (prenos nevznikol)

$4_8 + 5_8 = 9_{10} = 11_8$

$\begin{array}{c} \uparrow \uparrow \\ \text{nižšia číslica súčtu} \\ \text{prenos} \end{array}$

$$I_2 + I_2 = 2_{10} = 10_2$$



**Príklad A10.** Spočítajte dvojkovo: 1 0 1 1 1 0 0 0

$$\begin{array}{r} 0\ 0\ 1\ 1\ 1\ 0\ 0\ 1\ 1 \\ \underline{\phantom{0\ 0\ 1\ 1\ 1\ 0\ 0\ 1\ 1}} \\ 1\ 1\ 1\ 1\ 1\ 0\ 0\ 1\ 1 \end{array}$$

← vzniknuté prenoso

Výsledok : 1 1 1 1 0 0 1 1

Ak chceme spočítanie uľahčiť a zrýchliť, vytvoríme si prehľadné tabuľky súčtov. Takéto tabuľky vyjadrujú pravidlá na spočítavanie v uvedených sústavách a sú v *Tab.A2*.

Dvojková sústava			Osmičková sústava														
+	0	1	+	0	1	2	3	4	5	6	7						
0	0	1	0	0	1	2	3	4	5	6	7						
1	0	10	1	1	2	3	4	5	6	7	10						
			2	2	3	4	5	6	7	10	11						
			3	3	4	5	6	7	10	11	12						
			4	4	5	6	7	10	11	12	13						
			5	5	6	7	10	11	12	13	14						
			6	6	7	10	11	12	13	14	15						
			7	7	10	11	12	13	14	15	16						

Šestnástková sústava																
+	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10
2	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11
3	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12
4	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13
5	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14
6	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15
7	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16
8	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17
9	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18
A	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19
B	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A
C	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B
D	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C
E	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D
F	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E

Tab. A2. Súčty čísel v rôznych číselných sústavách

### Odpočítanie

Aj keď sa v počítači vykonáva odpočítanie prostredníctvom spočítania doplnkových alebo inverzných obrazov operandov, vhodné je vedieť odpočítaj aj v inej sústave ako

v desiatkovej. Postup je rovnaký ako v desiatkovej sústave: ak si uvedomíme hodnotu základu sústavy a pravidlo, že pri odpočítaní väčšej číslice od menšej si z vyššieho rádu vypožičiame jednotku, treba ju hneď po určení číslice rozdielu pripočítať k vyššiemu rádu.

### Násobenie

Pri násobení v každej polyadickej sústave sa postupuje rovnako ako v bežnej desiatkovej sústave. Uplatňuje sa pritom postupné pričítanie určeného násobku a posuv.

**Príklad A11.** Vynásobte v dvojkovej sústave  $101,101 \times 11,01$ . Sledujte pozorne riešenie, v ktorom zámerne nepoužívame šikmé písmo:

$$\begin{array}{r}
 \underline{101,101} \times \underline{11,01} \\
 101101 \\
 000000 \\
 \hline
 0101101 \quad \text{čiastkový súčet 1} \\
 \underline{101101} \\
 11100001 \quad \text{čiastkový súčet 2} \\
 \underline{101101} \\
 10010,01001 \quad \text{Výsledok (3 + 2 = 5 miest za rádovou čiarkou)}
 \end{array}$$

Násobenie prebieha tak, že sa násobenec násobí postupne jednotlivými číslicami násobiteľa, čiastkové násobky sa posunú o 1 miesto vľavo a pre názornosť sme v príklade hneď vytvorili čiastkový súčet. Nulové násobky násobenca sa nemusia zapisovať, rešpektuje sa však pritom posun o jedno miesto doľava. V počítači sa potom násobenie uskutočňuje kombinovaním posunov a spočítania.

V *tab. A3* ukazujeme tabuľky násobkov dvoch číslíc v dvoch číselných sústavách.

Dvojková sústava			Osmičková sústava								
X	0	1	X	0	1	2	3	4	5	6	7
0	1	0	0	0	0	0	0	0	0	0	0
	0	1	1	0	1	2	3	4	5	6	7
1	0	1	2	0	2	4	6	10	12	14	16
	1	0	3	0	3	6	11	14	17	22	25
			4	0	4	10	14	20	24	30	34
			5	0	5	12	17	24	31	36	43
			6	0	6	14	22	30	36	44	52
			7	0	7	16	25	34	43	52	61

*Tab. A3* Tabuľka násobkov v dvojkovej a šestnástkovej číselnej sústave

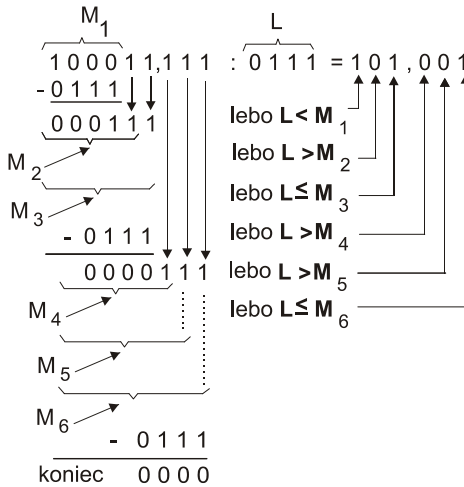
**Delenie**

Delenie je zložitá operácia, lebo pracuje s odhadom. V každom kroku odhadujeme, koľkokrát je deliteľ  $L$  obsiahnutý v „časti“ delenca (označme ho  $M$ ) alebo vo zvyšku. Tento odhad je obťažný aj v desiatkovej sústave a oveľa obťažnejší je v iných sústavách.

Relatívne jednoduchšie je delenie v dvojkovej sústave, kde vlastne iba zisťujeme, či platí  $L > M$  (potom hľadaná číslica podielu je 0) alebo  $L \leq M$  (hľadaná číslica podielu je 1)

**Príklad A12.** Deľte v dvojkovej sústave  $00011,111 : 0111$ . Sledujte pozorne riešenie.

Delenie realizuje postupným odpočítaním deliteľa od delenca a postupným priberaním číslice delenca. Je to názorný príklad „ručného“ riešenia, v počítači sú algoritmy obmenené.



**A.5 Prevody medzi sústavami**

Ukázalo sa, že medzi jednotlivými číselnými sústavami existujú isté vnútorné vzťahy. Uvidíme to na nasledujúcom príklade, ktorým chceme demonštrovať možnosť aplikácie takého prevodu medzi číselnými sústavami, ktorý vedie k rýchlejšiemu nájdeniu hľadaného riešenia úlohy.

Ukážme najprv prevod medzi sústavami, ktorý nevyužíva sprostredkovateľskú funkciu bežnej desiatkovej sústavy. Postup je rovnaký ako v príkladoch uvedených napríklad v príkladoch A3 až A5. Používa sa tu postupné delenie najčastejšie v *tej sústave, z ktorej sa vykonáva prevod.*

**Príklad A13** . Urobte prevod čísla  $110011$  do osmičkovej sústavy. Nasledujúce riešenie je dosť názorné a inštruktívne.

Dané číslo budeme postupne deliť ôsmimi, budeme to však robiť v dvojkovej sústave. Pritom platí  $8_{10} = 1000_2$  .

$$\begin{array}{r}
 110011 : 1000 = 110 \\
 \hline
 - 1000 \quad \downarrow \downarrow \\
 01001 \quad \downarrow \\
 \hline
 - 1000 \quad \downarrow \\
 \hline
 \text{zvyšok} \quad 00011 = a_0 = 3_8, \quad a_1 = 6_8 \quad (\text{menší ako } 8, \text{ prevod teda skončil.})
 \end{array}$$

Platí teda :  $110011_2 = 63_8$  .

Všimneme si zaujímavú skutočnosť: ak rozdelíme dané číslo sprava po trojiciach, tak každá trojica predstavuje priamo *osmičkovú číslicu* (v našom prípade  $\underline{110}$ ,  $\underline{011}$ ).



Naopak, každú osmičkovú číslicu možno zapísať priamo ako dvojkovú trojicu. Vyplýva to z toho, že medzi základmi oboch sústav platí vzťah  $8 = 2^3$  . Môžeme ho využívať na rýchlejší prevod napr. z desiatkovej sústavy do dvojkovej sústavy cez osmičkovú sústavu.

Analogický je vzťah medzi šestnástkovou a dvojkovou sústavou. Tentoraz sú však všetky šestnástkové číslice daného čísla zobraziteľná štvoricami dvojkových čísel. Napríklad

$$\begin{array}{cccccccc}
 B7FA52_{16} & = & 1011 & 0111 & 1111 & 1010 & 0101 & 0010_2 \\
 & & B & 7 & F & A & 5 & 2.
 \end{array}$$

## PRÍLOHA B. ZÁKLADNÉ LOGICKÉ FUNKCIE

Predpokladajme dve nezávislé logické premenné  $x_1$  a  $x_2$ , ktorým zodpovedajú celkovo štyri rôzne stavy:  $00$ ,  $0I$ ,  $I0$  a  $II$ . Týmto štyrom stavom možno priradiť hodnotu  $0$  alebo  $I$  dokopy šestnástimi spôsobmi (dá sa vytvoriť celkom  $16$  rôznych zobrazení), možno teda vytvoriť  $16$  rôznych logických funkcií, ktoré označíme postupne  $f_1, f_2, \dots, f_{16}$ . Tieto logické funkcie uvádza *Tab. B1*. Niektoré z nich si všimneme bližšie. Každá z uvedených funkcií má svoj názov a zodpovedajúce symbolické označenie. Ukážeme aj, že jednotlivé funkcie nie sú navzájom nezávislé, ale naopak, platia medzi nimi isté logické vzťahy [16,30].

**Logická funkcia je zobrazenie, ktoré stavom nezávisle logických premenných priraduje hodnoty závisle logických premenných.**

Funkcia  $f_1 = x_1 + x_2$  sa nazýva *logický súčet* (disjunkcia). Pre ilustráciu uveďme nasledujúcu interpretáciu s pomocou výrokového počtu: Považujme nezávislé logické premenné  $x_1$  a  $x_2$  za jednoduché výroky a hodnoty  $0, I$  týchto premenných za **pravdivostné hodnoty** tých výrokov. Potom funkčnú hodnotu funkcie  $f_1$  môžeme považovať za pravdivostnú hodnotu zloženého výroku, ktorý je formulovaný takto:  $f_1$  nadobúda hodnotu  $I$  vtedy a len vtedy, ak  $x_1$  alebo  $x_2$ , alebo obidve premenné nadobúdajú hodnotu  $I$ . Spojka **alebo** vyjadruje zároveň operáciu, t.j. postup, ktorým sa jednoznačne získa funkčná hodnota pre jednotlivé stavy nezávisle premenných. Znamienko plus (+) je symbolom tejto operácie. Vo všeobecnosti symbol operácie nazývame *operátor*.

Funkciu  $f_2 = x_1 \cdot x_2$  nazývame *logický súčin* (konjunkcia) a nadobúda hodnotu  $I$  vtedy a len vtedy, ak  $x_1$  aj  $x_2$  (súčasne) nadobúdajú hodnotu  $I$ . Symbolom tejto operácie je bodka, ktorá sa však často vynecháva.

Dôležité miesto medzi logickými funkciami má funkcia  $f_9$ . Je to funkcia negácie a je to funkcia *jednej* nezávisle premennej (t.j. unárna funkcia). V tabuľke vidno, že ak  $x_1$  nadobúda hodnotu  $0$ , je  $f_9 = I$ , ak  $x_1 = I$ , je  $f_9 = 0$ . Operáciu negácie označujeme čiarou nad príslušnou premennou, teda pre funkciu  $f_9$  môžeme písať  $f_9 = \bar{x}_1$  (čítaj  $\bar{x}_1$  non).

Významná je funkcia  $f_6$ , tzv. Shefferova funkcia, ktorá nadobúda hodnotu  $I$  pre všetky stavy nezávisle premenných okrem stavu, v ktorom  $x_1$  a súčasne aj  $x_2$  nadobúda hodnotu  $I$ . Zodpovedajúcu operáciu zapisujeme pomocou šípky smerom hore:  $f_6 = x_1 \uparrow x_2$ . Porovnaním stĺpcov funkčných hodnôt  $f_6$  a  $f_2$  zistíme, že *Shefferova funkcia je negáciu funkcie logického súčimu*, t.j.  $f_6 = \overline{f_2}$  alebo  $x_1 \uparrow x_2 = \overline{x_1 \cdot x_2}$ .

Logické premenne	$f_1$ logický súčet	$f_2$ logický súčin	$f_3$ implikácia	$f_4$ ekvivalencia
$x_1 \quad x_2$	$x_1 + x_2$	$x_1 \cdot x_2$	$x_1 \rightarrow x_2$	$x_1 \equiv x_2$
0 0	0	0	I	I
0 I	I	0	I	0
I 0	I	0	0	0
I I	I	I	I	I
Logické premenne	$f_5$ nonekvivalencia	$f_6$ Shefferova funkcia	$f_7$ Peirceova funkcia	$f_8$ inhibícia
$x_1 \quad x_2$	$x_1 \neq x_2$	$x_1 \uparrow x_2$	$x_1 \downarrow x_2$	$x_1 + \rightarrow x_2$
0 0	0	I	I	0
0 I	I	I	0	0
I 0	I	I	0	I
I I	0	0	0	0
Logické premenne	$f_9$ negácia $x_1$	$f_{10}$ negácia $x_2$	$f_{11}$ asercia $x_1$	$f_{12}$ asercia $x_2$
$x_1 \quad x_2$	$\overline{x_1}$	$\overline{x_2}$	$x_1$	$x_2$
0 0	I	I	0	0
0 I	I	0	0	I
I 0	0	I	I	0
I I	0	0	I	I
Logické premenne	$f_{13}$ spätná implikácia	$f_{14}$ spätná inhibícia	$f_{15}$ nulová funkcia	$f_{16}$ jednotková funkcia
$x_1 \quad x_2$	$x_1 \leftarrow x_2$	$x_1 \leftarrow + x_2$	0	I
0 0	I	0	0	I
0 I	0	I	0	I
I 0	I	0	0	I
I I	I	0	0	I

Tab. B1 Prehľad základných logických funkcií

Analogická situácia je aj pri funkcii  $f_7$ , ktorú symbolicky označujeme šípkou dolu a nazývame Pierceovou funkciou, vo vzťahu k funkcii logického súčtu. Platí  $f_7 = \overline{f_1}$ , alebo tiež  $x_1 \downarrow x_2 = \overline{x_1 + x_2}$ .

Poznamenajme ešte, že funkcii, ktorá nadobúda funkčnú hodnotu I bez ohľadu na stav nezávisle premenných, hovoríme **tautológia**.

Práve sme naznačili, že jednu funkciu môžeme vyjadriť pomocou inej funkcie. Na základe analýzy tohoto problému možno z množiny základných funkcií vybrať také súbory funkcií, ktorými možno vyjadriť ľubovoľnú inú logickú funkciu. Súbory, ktoré majú túto

vlastnosť nazývame *úplné súbory funkcií*. Spomenieme tu tri najdôležitejšie úplné súbory funkcií:

1. logický súčet, logický súčin, negácia,
2. Shefferova funkcia,
3. Pierceova funkcia.

Pre prvý z uvedených úplných súborov bola vypracovaná algebra, ktorá je účinným prostriedkom pre analýzu aj syntézu logických obvodov. Jej autorom je *George Boole* a je zrozumiteľnou a často uplatňovanou metódou opisu i transformácie logických funkcií.

Ostatné dva úplné súbory funkcií sú jednoprvkové a neuvádzame ich tu náhodou. Poznamenajme najprv, že *každú zo základných funkcií* uvedených v troch opísaných úplných súboroch funkcií *možno realizovať fyzicky* (aj fyzikálne) základným logickým obvodom, ktorý nazývame *základný logický člen*. Prax ukázala, že *pre metodiku* analýzy a syntézy logických sietí (abstraktný model) je vhodný *booleovský súbor*, ale *pre fyzikálnu realizáciu* logických obvodov sú *vhodnejšie Shefferove*, resp. Peirceove členy. Pripomeňme tiež, že logickú premennú, ktorá nadobúda dve definované hodnoty, nazývame aj *boolovská* premenná.

## Tabuľkový, algebraický a mapový zápis logickej funkcie

Každú logickú funkciu môžeme vyjadriť (zapísať) rozličnými spôsobmi; uvedieme tu tie, ktoré sa používajú najčastejšie : tabuľkové, algebraické a mapové. Hovoriť budeme nielen o zápise základných funkcií s dvoma nezávislými premennými (pozri napr. *Tab.B1*), ale budeme predpokladať aj väčší počet nezávisle premenných.

Ak je každému stavu  $x_1 x_2 \dots x_n$  nezávisle premenných nejakej logickej funkcie  $y = F(x_1, x_2, \dots, x_n)$  priradená funkčná hodnota, ide o *úplne určenú logickú funkciu*  $F$ . Ak chýba priradenie funkčnej hodnoty hoci len jednému stavu nezávisle premenných, ide o *neurčitú funkciu* (neúplne určenú) a taký stav nazývame *neurčený stav*. Namiesto funkčnej hodnoty  $0$ , resp.  $1$  píšeme potom symbol  $\times$ .

## Tabuľky

Tabuľkový spôsob zápisu logickej funkcie  $y = F(x_1, x_2, \dots, x_n)$  je univerzálny a nezávisí od druhu použitej logickej algebry. Jednotlivé  $n$ -tice (stavy) nezávisle logických premenných  $x_1 x_2 \dots x_n$  sú zapísané postupne pod sebou (v istom usporiadaní podľa tzv. stavového indexu) a ku každému stavu (ak ide o úplne určenú funkciu) je pripísaná hodnota logickej závisle premennej  $y$ , t.j. funkčná hodnota  $0$  alebo  $1$ .

Stavový index je desiatkové číslo  $s$ , ktoré dostaneme, ak transformujeme stav nezávisle premenných, ktorý považujeme za dvojkové číslo, do desiatkovej sústavy.

V Tab. B2a je príklad tabuľkového zápisu logickej funkcie  $y_1 = F(x_1, x_2, x_3)$ , v ktorom sa objavuje aj stĺpec zodpovedajúcich stavových indexov. Takéto tabuľky nazývame **pravdivostné tabuľky** (alebo *tabuľky pravdivostných hodnôt*). Stav, ktorému je priradená funkčná hodnota  $I$ , nazývame *jednotkový* stav nezávisle logických premenných, stav  $s$  funkčnou hodnotou  $0$  je *nulový* stav. (Lahko zistíte, že uvedená tabuľka definuje takto definovanú logickú funkciu: stav nezávisle premenných  $x_1, x_2, x_3$  obsahuje aspoň dve jednotky.)

Niekedy, pri zadávaní príkladov alebo pri precvičovaní, sa funkcie zadávajú zjednodušene iba množinou tých stavových indexov, pre ktoré má príslušná funkcia funkčnú hodnotu  $I$ . Pri neurčitej funkcii sa stavové indexy neurčených stavov zapíšu napr. do zátvorky. Pre funkciu podľa Tab.B2b možno teda zjednodušene písať:  $y_2 = I$  pre  $s = I, 4, 6, 7, (2, 5)$ .

$s$	$x_1 x_2 x_3$	$y_1$
0	0 0 0	0
1	0 0 1	0
2	0 1 0	0
3	0 1 1	I
4	1 0 0	0
5	1 0 1	I
6	1 1 0	I
7	1 1 1	I

a)

$s$	$x_1 x_2 x_3$	$y_2$
0	0 0 0	I
1	0 0 1	0
2	0 1 0	×
3	0 1 1	0
4	1 0 0	I
5	1 0 1	×
6	1 1 0	I
7	1 1 1	I

b)

$s$	$x_1 x_2 x_3$	$y_3 = ((x_1 \equiv x_2) \rightarrow x_3) \downarrow (x_1 + x_3)$
0	0 0 0	0
1	0 0 1	I
2	0 1 0	0
3	0 1 1	0
4	1 0 0	0
5	1 0 1	0
6	1 1 0	0
7	1 1 1	I

c)

Tab. B2 Príklady zápisu logických funkcií

Logická funkcia zapísaná v Tab.B2c sa formálne nelíši od predchádzajúcich dvoch, rozdiel je len v tom, ako sme priradili funkčné hodnoty jednotlivým stavom nezávisle premenných. V prípade  $y_1$  sme ich priradili v súlade so slovnou definíciou logickej funkcie, v prípade  $y_2$  sme funkčné hodnoty zvolili náhodne, len aby sme ukázali príklad neúplne určenej logickej funkcie, ale v treťom prípade boli funkčné hodnoty vyčíslené na základe logického výrazu, ktorý je zapísaný v záhlaví príslušného stĺpca a určuje  $y_3$ .

### Algebraické podoby logických výrazov a funkcií

Zápis logickej funkcie algebraickými podobami logických výrazov je už závislý od druhu použitej logickej algebry. Logický výraz možno vo všeobecnosti definovať ako

algebraickú formulu, ktorá obsahuje symboly logických premenných, konštanty 0, 1, operátory (t.j. symboly logických operácií ktoré určujú, ako spolu súvisia príslušné logické premenné) a prípadne aj zátvorky. Príklady jednoduchých logických výrazov nájdeme v tom riadku Tab.B1, ktorý je napísaný vždy pod slovným názvom každej základnej funkcie. Treba povedať, že algebraický výraz, obsahujúci v sebe napr. operácie implikácie, Shefferovej, Pierceovej funkcie, inhibície atď., je pre človeka síce nezvyčajný, ťažko "čitateľný", ale pomerne ľahko vyčísliteľný. Máme tým na mysli proces, ktorým zistíme, akú logickú hodnotu predstavuje daný výraz pre konkrétne hodnoty logických premenných, ktoré obsahuje. Hovoríme, že logický výraz vyhodnocujeme.

**Príklad B1.** Na ilustráciu jednotlivých krokov vyhodnoťme výraz určujúci funkčné hodnoty  $y_3$  v Tab.B2c. Najlepšie je v takýchto prípadoch pracovať s tabuľkou a rozkladať počítačový výraz na jednoduchšie výrazy, obsahujúce - pokiaľ možno - vždy dvojicu logických premenných viazanú operátorom. To nám umožní - pomocou definičných tabuliek jednotlivých základných funkcií - postupne určovať pre každý riadok hľadanú logickú hodnotu zodpovedajúceho výrazu. Postup opakujeme tak dlho, kým nie je vyhodnotený celý pôvodný výraz. Postup je zrejmy z Tab.B3.

	$A$	$\overline{A}$	$B$	$C$	
$x_1 x_2 x_3$	$(x_1 \equiv x_2)$	$\overline{x_3}$	$A \rightarrow \overline{x_3}$	$\overline{x_1 + x_3}$	$B \downarrow C$
0 0 0	1	1	1	1	0
0 0 1	1	0	0	0	1
0 1 0	0	1	1	1	0
0 1 1	0	0	1	0	0
1 0 0	0	1	1	0	0
1 0 1	0	0	1	0	0
1 1 0	1	1	1	0	0
1 1 1	1	0	0	0	1

Tab. B3 Vyhodnotenie logického výrazu

Spomínaná ťažkopádnosť pri práci so všeobecnými logickými výrazmi a funkciami spôsobila, že sa v praxi pre vyjadrenie ľubovoľnej logickej funkcie takmer výlučne využíva boolovský úplný súbor funkcií, t.j. logický súčin (*and*), logický súčet (*or*) a negácia (*not*). Algebra, založená na tomto súbore je veľmi blízka ľudskému mysleniu a vedie k nasledujúcim definíciám: *Logický výraz* obsahujúci okrem logických premenných a konštant 0, 1 výlučne operátory logického súčinu, logického súčtu a negácie nazývame boolov výraz (prípadne *boolovský výraz*) alebo krátko *B- výraz*.

Logickú funkciu vyjadrenú boolovými výrazmi nazývame boolovou funkciou, alebo krátko *B- funkciou*. *B- funkcia*  $f(x_1, x_2, \dots, x_n)$   $n$  nezávisle premenných  $x_1, x_2, \dots, x_n$  je

zobrazenie, ktoré priraduje každej  $n$ -tici hodnôt týchto premenných ( $n$ -tíc je  $2^n$ ) *funkčnú hodnotu* 0 alebo 1. Vidíte, že definícia boovej funkcie je zhodná s definíciou všeobecnej logickej funkcie, a rozdiel spočíva len v algebraickom vyjadrení.

Akúkoľvek  $B$ - funkciu  $y = f(x_1, x_2, \dots, x_n)$  možno vždy vyjadriť (zapísať) *úplnou súčtovou normálnou formou*

$$y = \sum_{s=0}^{2^n-1} y_s \cdot K_s \tag{B.1}$$

kde  $\Sigma$  znamená logický súčet,  $\cdot$  logický súčin,  $s$  je stavový index,  $y_s$  funkčná hodnota zodpovedajúcu stavovému indexu  $s$  a  $K_s$  je základná konjunkcia.

*Základný súčin* (tiež *minterm* alebo *základná konjunkcia*) logickej funkcie je súčinový výraz  $x_1 x_2 \dots x_n$ , v ktorom symbol  $x_i$  ( $i = 1, 2, \dots, n$ ) je bez negácie, ak je na zodpovedajúcom mieste v príslušnom stave v tabuľke hodnota 1, a naopak,  $\bar{x}_i$  je negované, ak na zodpovedajúcom mieste je hodnota 0.

**Príklad B2.** Tabuľka B4 vyjadruje  $B$ - funkciu troch premenných a v jej treťom stĺpci sú zapísané príslušné mintermy.

$s$	$x_1 x_2 x_3$	mintermy $K_s$	$y_s$
0	0 0 0	$\bar{x}_1 \bar{x}_2 \bar{x}_3$	I
1	0 0 1	$\bar{x}_1 \bar{x}_2 x_3$	0
2	0 1 0	$\bar{x}_1 x_2 \bar{x}_3$	0
3	0 1 1	$\bar{x}_1 x_2 x_3$	I
4	1 0 0	$x_1 \bar{x}_2 \bar{x}_3$	I
5	1 0 1	$x_1 \bar{x}_2 x_3$	0
6	1 1 0	$x_1 x_2 \bar{x}_3$	I
7	1 1 1	$x_1 x_2 x_3$	I

Tab. B4. Príklad k vyjadreniu  $B$ -funkcie úplnou súčtovou normálnou formou

Ak do vzťahu (B.1) dosadíme za  $y_s$  a  $K_s$  jednotlivé hodnoty a výrazy z uvedenej tabuľky, dostaneme pre zadanú funkciu túto úplnú súčtovú normálnu formu :

$$y = \bar{x}_1 \bar{x}_2 \bar{x}_3 + \bar{x}_1 \bar{x}_2 x_3 + x_1 \bar{x}_2 \bar{x}_3 + x_1 \bar{x}_2 x_3 + x_1 x_2 \bar{x}_3 + x_1 x_2 x_3. \tag{B.2}$$

Inak povedané, *úplná súčtová (disjunktná) normálna forma* zodpovedá logickému súčtu tých mintermov (základných logických súčinov), ktorým prislúcha funkčná hodnota 1 závisle premennej. Označujeme ju ÚDNF, pričom slovo *úplná* znamená, že *obsahuje všetky*

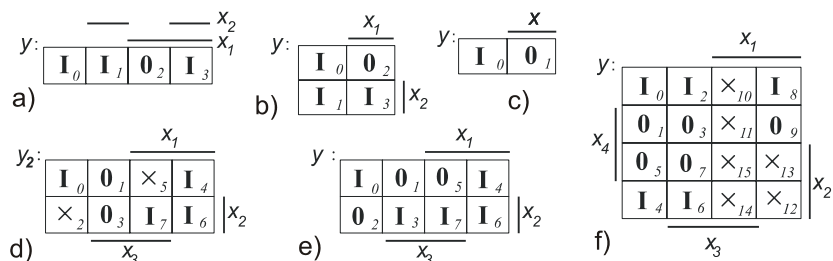
jednotkové mintermy a každý minterm obsahuje *všetkých*  $n$  symbolov nezávisle premenných. *Disjunktná* (alebo *súčtová forma*) preto, že mintermy viaže spolu operácia logického súčtu a *normálna forma* preto, že sa to chápe ako dohovorený normatívny tvar. Vidíte, že všetky logické výrazy v tejto forme obsahujú len boolovské operátory.

Treba povedať, že ÚDNF nie je optimálnou formou z hľadiska zložitosti logického obvodu, ktorým sa daná funkcia má realizovať. Na základe toho, čo sme už v 3. kapitole povedali o kombinačnom logickom **systeme** je zrejmé, že vyjadrenie funkcie algebraickými výrazmi určuje štruktúru príslušného logického obvodu. Každému základnému logickému výrazu v algebraickom vyjadrení logickej funkcie totiž možno priradiť jeden základný logický člen. Z toho plynie, že čím viac výrazov forma obsahuje (a čím sú tieto výrazy dlhšie), tým zložitejší (a teda aj drahší) je obvod, ktorý danú funkciu realizuje. Preto sa - pokiaľ možno - hľadajú formy *skrátene* či *minimálne* (MDNF). Procesu hľadania minimálnych foriem *hovoríme minimalizácia normálnych foriem* a uplatňuje sa pri nich Boolova algebra a jej zákony.

### Mapový zápis logickej funkcie

Účinným nástrojom hľadania minimálnych, alebo aspoň skrátenej normálnych foriem je zápis funkcie do **logickej mapy**. Mapou rozumieme obdĺžnik alebo štvorec, priradený logickej funkcii tak, že pre  $n$  nezávisle premenných uvažovanej funkcie je rozdelený na  $2^n$  políčok, pričom každé políčko mapy prislúcha jedinému stavu nezávisle premenných, (t.j. jednému riadku príslušnej pravdivostnej tabuľky).

Aby bolo vidno ktorému konkrétne, vyznačia sa čiarami okolo mapy oblasti, v ktorých príslušná nezávislá premenná nadobúda hodnotu 1 a tam kde čiara nie je, platí, že



Obr. B1 Logické mapy pre funkcie 1,2,3 a 4 nezávisle premenných.

tam tá premenná nadobúda hodnotu 0. Na obr. B1a je ukážka takej mapy pre dve nezávislé logické premenné a je v nej zapísaná základná funkcia  $f_3$  (implikácia) z tab. B1. Do políčok mapy sme pripísali aj im prislúchajúce stavové indexy, takže sa dá veľmi ľahko zistiť, že

mapa je vlastne iba špeciálnou podobou pravdivostnej tabuľky. Poznamenajme, že autorom mapy s uvedeným čiarovým priradením bol známy československý vedec prof. Antonín Svoboda. Jeho priradenie čiar mapy (hovoríme tomu priradenie premenných mapy) sa vyznačovalo tým, že čiary „kopírovali“ rozloženie núl a jednotiek v stĺpcoch nezávisle premenných príslušnej pravdivostnej tabuľky.

Karnaugh ukázal, že priradenie čiar mapy podľa iných pravidiel (*obr. B1a – f*), (v ktorých sú čiary súvislé a „nepretrhnuté“) spôsobí to, že dve políčka ležiace tesne vedľa seba v riadku alebo stĺpci patria takým stavom nezávisle premenných, ktoré líšia iba v hodnote jedinej nezávisle premennej (túto vlastnosť majú aj políčka mapy na opačných koncoch toho istého riadku alebo toho istého stĺpca).

Uvádžame ukážky týchto Karnaughových máp pre rôzne počty nezávisle premenných vo funkciách. Tak *obr. B1b* vyjadruje tú istú funkciu ako v prípade Svobodovej mapy z obrázku *a*. Ďalší *obr. B1c* je mapou pre funkciu jednej nezávisle premennej a zobrazuje funkciu **negácie**:  $y = \bar{x}$ .

*Obr. B1d* je mapou vytvorenou pre funkciu  $y_2$  z *Tab.B2,b*. Ďalší *obr.B1e* je zápisom funkcie z *tab. B4*, pre ktorú sme našli úplnú súčtovú normálnu formu (B.2). A konečne, *obr. f* je zápisom istej logickej funkcie, ktorú by ste mohli odhaliť aj sami: ide zrejme o funkciu, ktorá nadobúda hodnotu 1 práve vtedy, keď je **desiatková číslica 0 – 9** (vyjadrená štvoricou bitov  $x_1x_2x_3x_4$  v kóde *BCD*) párna. Svedčí o tom tiež fakt, že stavy 10 – 15 nezávisle premenných nie sú v kóde definované, takže sú im v mape priradené neurčené funkčné hodnoty.

Bez toho, že by sme chceli zachádzať do podrobností, ukončíme náš výklad vyslovením dôvodov, prečo je Karnaughova mapa taká preferovaná. Využime na to len jeden zo zákonov Booleovej algebry (pozri Prílohu C):

$$A \cdot x + A \cdot \bar{x} = A, \quad (\text{B.3})$$

kde  $A$  je nejaký logický výraz a  $x$  je logická premenná. Vzťah (B.3) umožňuje tvrdiť, že ak sa v súčtovej normálnej forme niektoré dva výrazy, t.j.  $Ax$  a  $A\bar{x}$  líšia v hodnote jedinej premennej (t.j. v hodnote  $x$ ), potom oba pôvodné výrazy možno nahradiť jediným výrazom  $A$  a premenná  $x$  sa stráca. To je jeden z princípov minimalizácie normálnych foriem, ktorá vo svojich dôsledkoch vedie k jednoduchším a teda lacnejším realizáciám logických obvodov

Karnaughova mapa priradením premenných mapy dáva pre každú dvojicu susedných políčok (ktoré ležia vedľa seba tak ako sme povedali vyššie), možnosť uplatniť pravidlo (B.3). To sa však využije len vtedy, ak sú v oboch takýchto políčkach jednotkové funkčné

hodnoty. Uvedomme si totiž, že do úplnej súčtovej normálnej formy sme brali len tie mintermy, ktoré sa viažu na jednotkové funkčné hodnoty danej funkcie. Proces vyhľadávania a vylučovania premenných (ktoré z výrazu vypadnú v zmysle pravidla (B.3)) je teda procesom hľadania minimálnych normálnych foriem a je vizuálne ľahko sledovateľný i kontrolovateľný. Preto v praxi pre menšie počty nezávisle premenných veľmi zvyšuje efektivitu a produktivitu práce pri syntéze logických obvodov. Záujemcov o hlbšie štúdium odkážeme napr. na práce [16,30,31].

## PRÍLOHA C. BOOLEOVA ALGEBRA

*Booleova algebra je množina, ktorá obsahuje logické premenné (napr.  $x, y, z, \dots$ ), konštanty  $0$  a  $1$  a operácie logického súčtu ( $+$ ), logického súčinu ( $\cdot$ ) a negácie ( $\bar{\phantom{x}}$ ). Platia pre ňu tieto zákony:*

Komutatívny zákon

$$x + y = y + x \quad (C1)$$

$$x \cdot y = y \cdot x$$

Asociatívny zákon

$$(x + y) + z = x + (y + z) \quad (C2)$$

$$(x \cdot y) \cdot z = x \cdot (y \cdot z)$$

Distributívny zákon

$$(x + y) \cdot z = x \cdot z + y \cdot z \quad (C3)$$

$$xy + z = (x + z) \cdot (y + z)$$

Zákon o vylúčení tretieho

$$x + \bar{x} = 1 \quad (C4)$$

$$x \cdot \bar{x} = 0$$

Zákon neutrálnosti hodnoty  $0$

$$x + 0 = x \quad (C5)$$

Zákon neutrálnosti hodnoty  $1$

$$x \cdot 1 = x \quad (C6)$$

Zákon agresivity hodnoty  $0$

$$x \cdot 0 = 0 \quad (C7)$$

Zákon agresivity hodnoty  $1$

$$x + 1 = 1 \quad (C8)$$

Zákon o idempotencii prvkov

$$x + x = x \quad (C9)$$

$$x \cdot x = x$$

Zákon absorpcie

$$x + x \cdot y = x \quad (C10)$$

$$x \cdot (x + y) = x$$

Zákon absorpcie negácie

$$x + \bar{x} \cdot y = x + y \quad (C11)$$

$$x \cdot (\bar{x} + y) = x \cdot y$$

Zákon dvojitej negácie

$$\bar{\bar{x}} = x \quad (C12)$$

De Morganove zákony

$$\begin{array}{l} \overline{\overline{x} \cdot \overline{y}} = \overline{\overline{x+y}} \\ \overline{\overline{x+y}} = \overline{\overline{x} \cdot \overline{y}} \end{array} \quad (C13)$$

Zákony uvedené vzťahmi (C1), (C3),(C4) a (C5) sa považujú za axiómy, ostatné sú z nich odvodené.

**Příklad C1.** Dokážte platnosť zákona (C10).

Podľa zákona o neutrálnosti hodnoty  $I$  možno písať:

$$x + x \cdot y = I \cdot x + x \cdot y,$$

podľa distributívneho zákona platí

$$I \cdot x + x \cdot y = x \cdot (I + y),$$

podľa zákona agresivity hodnoty  $I$  je

$$I + y = I,$$

teda

$$x + x \cdot y = x.$$

Nie je cieľom tejto prílohy robiť dôkazy či aplikácie zákonov Booleovej algebry, chceme len poskytnúť čitateľovi prehľad pre prípad, že ho bude potrebovať. Odkazy na literárne pramene sú rovnaké, ako v prílohe B.

## Literatúra

- [ 1 ] Poletajev, I. A.: *Kybernetika*. SNTL Praha, 1961
- [ 2 ] Blatný, J., Křišťoufek, K., Pokorný,Z., Kolenička,J.: *Číslicové počítače* . Vysokoškolská učebnica, SNTL Praha, ALFA Bratislava, 1.vyd.1982, 2. vyd. 1989.
- [ 3 ] Shannon,C.,E.: *A Mathematical Theory of Communication*. The Bell System Technical Journal,1948
- [ 4 ] Kolenička, J.: *Informačné technológie a spoločnosť*. In: Zborník „Trendy ekonomického rozvoja Slovenskej republiky 1998“. Trenčianska univerzita v Trenčíne, 8.-9. júla 1998, str. 143 – 150, ISBN 80- 88914- 01- 9.
- [ 5 ] Klír, J., Valach, M.: *Kybernetické modelování*.SNTL Praha, 1965
- [ 6 ] Kolenička, J.: *Logické systémy. Část I*. Skriptum VUT Brno, 99 str. SNTL Praha, 1976, ISBN 05 –101- 76
- [ 7 ] Návrat,P., Šafařík, J.: *Informatika v kontexte vied o informáciách*. In. Proceedings of the Computer Science Education Workshop. Košice – Herľany, October 25.- 26.2001, str.7.
- [ 8 ] Markov, A.: *Teoriya algorifmov*. Trudy matem. instituta AN ZSSR V.A Steklova, vol.42, Izd. AN ZSSR, 1954.
- [ 9 ] Wirt, N.: *Systematické programovanie*. ALFA – SNTL, Bratislava, 1989
- [ 10 ] *A Dictionary of Computing*. Oxford University Press, Fourth edition 1996, ISBN 0-19-280046-9
- [ 11 ] Drlík,P., Hvorecký,J.: *Informatika – náčrt didaktiky*. Skriptum Pedagogickej fakulty v Nitre, 1992, ISBN 80–85183–81–1
- [ 12 ] Benadiková,A., Mada,Š., Weinlich,S.: *Čárové kódy – automatická identifikace*. GRADA, Praha, 1994, ISBN 80-85623-66-8
- [ 13 ] Bushnell, R. D., Meyers, R.R.: *Getting started with bar codes*. Bushnell Consulting Group,Inc. 1993
- [ 14 ] Sobota, B.: *ABC o PC ' 96*. ibis PUBLISHING, Košice, 1996. ISBN 80-967063-8-1.
- [ 15 ] Pluháček, A.: *Aritmetika a kódy*. Ediční středisko ČVUT Praha, 1981
- [ 16 ] Kolenička, J., Boltík, J.: *Technika počítačov I*. Alfa-pres Bratislava, 1995, 3.vydanie. ISBN 80-88811-20-1
- [ 17 ] Dembowski, K.: *PC v tabulkách*.(Podrobný průvodce osobními počítači.) UNIS Publishing Brno, 1996.
- [ 18 ] Sedláček, J.: *Úvod do teorie grafů*. Academia Praha, 1977, 510-21-826

- [19] Demel, J.: *Grafy*. Nakladatelství technické literatury Praha, 1988. 04-006-88
- [20] Dudeney, H.E.: *The Canterbury Puzzles and other Curious Problems*. Ruský preklad: Djudeni, G.E.: Kanterberijskije golovolomki. Izd. Mir, Moskva 1979.
- [21] Gatial, J., Hecht, T., Hejný, M.: *Hry takmer matematické*. Mladá fronta, Praha 1982.
- [22] Loyd, S.: *Matematical Puzzles*. Ruský preklad: Matematičeskaja mozaika. Izd. Mir, Moskva, 1980.
- [23] Turnovec, F., Chobot, M.: *Teória hier*. SPN, Bratislava 1967.
- [24] Dirac, G. A.: Proc. London Math. Soc. 2 (1952), 69–81.
- [25] Valach, M.: *Seminár a cvičenia z matematiky (Teória hier)*. Skriptum, SPN, 1989, 067 – 268 – 89.
- [26] Gik, E. Ja.: *Matematika na šachmatnoj doske*, izd. Nauka, Moskva 1976.
- [27] Kolenička, J.: *Čiarové kódy názornejšie*. Informatika v škole. č.25/2003. str.20., 1/3 AH, ISSN 1335 – 616X.
- [28] *Konferencia o počítačovej kriminalite* organizovaná Americkou komorou právnikov, Právnou iniciatívou pre strednú a východnú Európu, v spolupráci s US Department of Justice, Federal Bureau of Investigation, Úradom vlády SR, Ministerstvom spravodlivosti SR, Generálnou prokuratúrou SR a Ministerstvom vnútra SR. Hotel Bôrik, Bratislava, 24.- 25. januára 2001.
- [29] Kolenička, J.: *Kriminalita v informatike*. Zborník „Computer Science Education Workshop 2002“, vydaný Žilinskou univerzitou pri rovnomennej konferencii. Žilina – Súľov, 7.-8. nov. 2002, ISBN 80-8070-016-8.
- [30] Klír, J., Seidl, L.K.: *Syntéza logických obvodů*. SNTL Praha, 1966.
- [31] Frištacký, N., Kolesár, M., Kolenička, J., Hlavatý, J.: *Logické systémy*. Celoštátna vysokoškolská učebnica - II. vydanie. (40,99 AH) Alfa /SNTL Bratislava, 1990 .
- [32] Škrinárová, J.: *Neural network for support of robot force control*. In: International Scientific Conference on Electronics Computers and Informatics Proceedings 2004, Herľany. ISBN 80-8073-150-0, s. 371–375
- [33] Škrinárová, J.: *Operačné systémy*. Skriptum FPV UMB Banská Bystrica, 72 str. 1.vydanie 1994.

## Register

### A

abeceda, na vstupe, 32,54  
abeceda, na výstupe, 32, 55  
abeceda, signálu, 36  
abeceda, vnútorných stavov, 55  
adresa, 17  
Aiken, 9  
algoritmický problém, 14  
algoritmus, 14  
analýza systému, 48

### B

Babbage, Ch., 9  
bajt, byte,  
bit, 4  
Boolova algebra, 130

### C

cesta, 785  
cesta, orientovaná, 84  
Codabar, 67  
Code 2/5, 66,67  
Code( 39, 128), 65,67  
cracker, 108  
cyklus, 84  
časová závislosť, 70  
čiarový diagram, 70  
číp, 11  
číselná sústava, 111  
číslo s pevnou čiarkou, 20  
číslo s pohyblivou čiarkou, 20

### D

diagram súvislostí, 71  
diagram, stĺpcový, 2  
dĺžka sledu, 77  
dĺžka slova, 5,36  
dostupnosť vrcholu, 78

### E

ENIAC, 9  
entropia na symbol, 7  
entropia, 6,7  
EPOS, 9

### F

faktor grafu, 77  
frekvencia, medzná, 40

funkcia, logická, 121  
funkcia, neurčitá, 123  
funkcia, prechodová, 56  
funkcia, výstupná, 54,56

### G

generácie počítačov, 10  
graf, konečný, 76  
graf, 3D, 72  
graf, 70  
graf, bodový, čiarový, 70  
graf, koláčový, 72  
graf, komplementárny, 75  
graf, konečný, 76  
graf, neorientovaný, 74  
graf, ohodnotený, 80,81  
graf, orientovaný, 83  
graf, pravidelný, 79  
graf, prázdny, 76  
graf, správania sa, 57,82  
graf, stĺpcový, 71  
graf, súvislý, 77  
graf, trojdimenzionálny, 72  
graf, úplný, 76  
grafy, disjunktné, 76  
grafy, komplementárne, 76  
grafy, rovné, 76  
grafy, rôzne, 76

### H

hacker, 108  
had, 80  
histogram, 70, 72  
histogram, združený, 72  
hra "15", 94  
hra, definícia, 87  
hráč, 87  
hrana, 74  
hrana, incidentná, 75  
hrana, ohodnotená, 80  
hrana, orientovaná, 80, 83  
hry, antagonistické, 99  
hry, kooperatívne, 103  
hry, maticové, 100  
hry, neantagonistické, 102

hry, počítačové, 93,98  
hry, proti prírode, 103  
hry, s nenulovým súčtom, 100  
hry, s nulovým súčtom, 100  
hry, so sedlovým bodom, 101  
hry, solitérne, 94  
húsenica, 80  
hviezda, 80

### I

identifikátor premennej, 16  
informačné technológie, 25  
informatika, 8,9  
inštrukcia, 16  
inštrukcia, skoková, 19  
Intel, 13  
interferencia, 33

### J

John von Neuman, 9

### K

kanál, prenosový, 32  
kapacita, informačná, 37  
kapacita, prenosová, 35  
kartodiagram, 72  
kód, 1z 10, 59  
kód, 2 z 5, 59  
kód,  $3n+2$ , 59  
kód, 5421,59  
kód, Aikenov, 59  
kód, ASCII, 62  
kód, BCD, 59  
kód, bikvinárny, 59  
kód, čiarový, 64  
kód, dodatkový, 69  
kód, Grayov, 59  
kód, Industrial, 66  
kód, Johnsonov, 59  
kód, KOI 8, 64  
kód,  $n+3$ , 59  
kód, prekrývaný, 67  
kód, Rubinoff, 59  
kód, zvyškových tried, 60  
kódová stránka 437, 63

kódy UPC, 65  
kódy, alfanumerické, 62  
kódy, desiatkové, 59  
kódy, EAN, 65, 67, 69  
koleso, 80  
komponent grafu, 78  
konečná schéma, 5  
kostra, 79  
kružnica, 79  
kružnica, hamiltonovská, 82,97  
kvantovanie signálu, 31  
lišta hry, 86

## L

Leibniz, G., 9  
les, 79  
lišta hry, 89  
Lloyd, S., 91,93

## M

mapa, logická, 127  
Mark I, 9  
matica susedností, 85  
matica, incidenčná, 86  
matica, susedností, 85  
matica, výplatná, 100  
médiu, informačné, 29  
miera apriórnej nevedomosti, 7  
miera neurčitosti, 5  
mikroinštrukcia, 20  
mikrooperácia, 20  
mikroprocesory, 12,13  
miniaturizácia, 11  
model, abstraktný, 50  
model, algebraický, 51,56  
model, správania sa, 49  
model, systému, 51  
modelovanie, 51  
modem, 34  
modulácia, amplitúdová, 33  
modulácia, frekvenčná, 34  
most, 100  
multigraf, 57,81

## eN

nadsystém, 47  
NIM, 88,99  
nosný signál, 40

## O

objekt, 41  
obvod, logický, 52  
okolie systému, 41  
operácia, 16,17  
operačná jednotka, 17  
operačný systém, 23  
operačný vrchol, uzol, 15  
operands, 16

## P

parabola, kubická, 70  
partia hry, 88  
Pascal, B., 9  
pásma logického signálu, 38  
pásmo, zakázané, 36,38  
Pentium, 13  
podgraf, vlastný, 76  
podmienkový vrchol, uzol, 15  
podsystem, 47  
pravdivostná tabuľka, 53,123  
pribeh funkcie, 70  
priemer grafu, 79  
priepustnosť kanálu, 35  
príkaz, priradovací, 18  
problém čiernej skrinky, 48  
procesor, 15  
program, 15

## R

rebrík, 80  
rozlišovacia úroveň, 44  
rýchlosť, prenosová, 40

## S

SAPO, 9  
sedlový bod, 102  
Shannon, C., 6  
sieť, logická, 53  
signál, 28  
signál, číslícový, 31  
signál, dvojhodnotový, 31  
signál, elementárny, 31  
signál, jednotková úroveň, 34  
signál, logický, 31  
signál, nosný, 34,39  
signál, nulová úroveň, 36

signál, sériový, 32  
signál, statický, 36, 38  
signály, paralelné, 31  
skok, podmienený, 19  
skok, tvrdý, 19  
sled, 77  
sled, orientovaný, 84  
sled, uzatvorený, 84  
slovo, logické, 22  
slovo, typu text, 22  
stupeň vrcholu, 76  
stupne integrácie obvodov, 11  
súvislosť grafov, 77, 78  
Svoboda, A., 128  
symbol z abecedy, 36  
syntéza systému, 48  
systém, definícia, 41, 45  
systém, kombinačný, 47,53  
systém, logický, 52  
systém, operačný, 23  
systém, sekvenčný, 47, 54  
systém, správanie sa, 42,45  
systém, štruktúra, 42,43  
systém, základné úlohy, 48  
systémy, hierarchia, 47  
systémy, informačné, 9, 48,  
štíhlosť kódu, 67

## T

tabuľka výplatná, 100  
tabuľka, pravdivostná, 124  
ťah, 78  
ťah, orientovaný, 84  
ťah, uzatvorený, 84  
tautológia, 122  
teória hier, 87  
Tic-tac-toe, 90  
Tri v rade, 90  
Turing, A., 9

## U

údaj, 5  
udalosť, 4  
UNICODE, 64  
uzol, 74  
uzol, vstupný, 41  
uzol, výstupný, 41

## **V**

vážňova dilema, 102  
vejár, 80  
vrchol, 74  
vrchol, incidentný, 75  
vrchol, izolovaný, 76  
vrchol, koncový, 75  
vrchol, konečného stupňa, 76

vrchol, krajný, 75  
vrchol, ohodnotený, 80  
vrchol, počiatočný, 81  
vrchol, susedný, 75  
vstup systému, 41  
výplatná tabuľka, 100  
výraz, algebraický, 125  
výraz, boolovský, 125

výraz, logický, 122  
výstup systému, 38  
vývojový diagram, 16  
vzťah incidencie, 72

## **Z**

zobrazenie, 78  
zobrazenie údajov, 20

## OBSAH

<b>1. INFORMÁCIA, INFORMATIKA</b>	<b>3</b>
1.1 Informačná entropia	5
1.2 Informatika	8
1.3 Historické pozadie informatiky	9
1.4 Spracovanie informácií	14
1.5 Využitie informatiky v praxi	24
1.6 Informačné technológie	24
<b>2. SIGNÁL</b>	<b>28</b>
2.1 Spojité a nespojité signály, kvantovanie	30
2.2 Informačný prenosový kanál	32
2.3 Prenosová kapacita kanálu	35
<b>3. SYSTÉM</b>	<b>41</b>
3.1 Vlastnosti systému	42
3.2 Definícia systému	45
3.3 Model	49
3.4 Logické systémy	52
<b>4. KÓDY</b>	<b>58</b>
4.1 Kódovanie desiatkových číslíc	58
4.2 Alfanaumerické kódy	62
4.3 Čiarové kódy	64
<b>5. GRAFY</b>	<b>70</b>
5.1 Základné predstavy	70
5.2 Neorientované grafy	74
5.3 Orientované grafy	83
5.4 Reprezentácia grafu v počítači	84
<b>6. HRY A ICH STRATÉGIA</b>	<b>87</b>
6.1 Solitérne hry	94
6.2 Hry typu NIM	99
6.3 Maticové hry	100
6.4 Hry proti prírode	103
<b>7. KRIMINALITA V INFORMATIKE</b>	<b>105</b>
7.1 Klasifikácia počítačových trestných činov	106
7.2 Kriminalita vzťahujúca sa na počítače	107
7.3 Zhrnutie	110

<b>PRÍLOHA A. ČÍSELNÉ SÚSTAVY</b>	<b>111</b>
A1. Dvojková sústava	112
A2. Osmičková sústava	114
A3. Šestnástková sústava	115
A4. Aritmetické operácie	116
A5. Prevody medzi sústavami	119
<b>PRÍLOHA B. ZÁKLADNÉ LOGICKÉ FUNKCIE</b>	<b>121</b>
<b>PRÍLOHA C. BOOLEOVA ALGEBRA</b>	<b>130</b>
LITERATÚRA	132
REGISTER	134
OBSAH	137

Autor Prof. Ing. Ján Kolenička, PhD  
Názov ÚVOD K INFORATIKE

Počet strán 138  
Počet obrázkov 48  
Počet tabuliek 16

AH 8,62

Vydala Katedra informatiky FPV UMB Banská Bystrica

Rok vydania 2004

Náklad 60ks

ISBN 80 – 8055 – 917 –1

EAN 9788080559175