



INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

TENTO PROJEKT JE SPOLUFINANCOVÁN EVROPSKÝM SOCIÁLNÍM FONDEM A STÁTNÍM ROZPOČTEM ČESKÉ REPUBLIKY.

# UMĚLÁ INTELIGENCE

**EVA VOLNÁ  
MARTIN KOTYRBA**



PODPORA TERCIÁRNÍHO VZDĚLÁVÁNÍ  
STUDENTŮ SE SPECIFICKÝMI  
VZDĚLÁVACÍMI POTŘEBAMI  
NA OSTRAVSKÉ UNIVERZITĚ V OSTRAVĚ

**CZ.1.07/2.2.00/29.0006**

**OSTRAVA, ČERVENEC 2013**

**Studijní opora je jedním z výstupu projektu ESF OP VK.**

<b>Číslo Prioritní osy:</b>	7.2
<b>Oblast podpory:</b>	7.2.2 – Vysokoškolské vzdělávání
<b>Příjemce:</b>	Ostravská univerzita v Ostravě
<b>Název projektu:</b>	Podpora terciárního vzdělávání studentů se specifickými vzdělávacími potřebami na Ostravské univerzitě v Ostravě
<b>Registrační číslo projektu:</b>	CZ.1.07/2.2.00/29.0006
<b>Délka realizace:</b>	6.2.2012 – 31.1.2015
<b>Řešitel:</b>	PhDr. Mgr. Martin Kaleja, Ph.D.

**Tento projekt je spolufinancován Evropským sociálním fondem a státním rozpočtem České republiky.**

**Název:** Umělá inteligence  
**Autor:** doc. RNDr. PaedDr. Eva Volná, PhD.  
RNDr. Martin Kotyrba, Ph.D.

Studijní opora k inovovanému předmětu: Umělá inteligence (KIP/UMINT)

Jazyková korektura nebyla provedena, za jazykovou stránku odpovídá autor.

**Recenzent:** RNDr. Martin Žáček, Ph.D.  
Ostravská univerzita v Ostravě

© Eva Volná, Martin Kotyrba  
© Ostravská univerzita v Ostravě  
ISBN 978-80-7464-330-9

## OBSAH:

Úvod.....	6
<b>1 Úvod do umělé inteligence .....</b>	<b>9</b>
1.1 Historie umělé inteligence.....	9
1.2 Umělá inteligence, jako pojem.....	13
1.3 Turingův test .....	14
Shrnutí kapitoly.....	15
<b>2 Řešení úloh a prohledávání stavového prostoru.....</b>	<b>17</b>
2.1 Stavový prostor a jeho prohledávání.....	18
2.2 Neinformované (slepé) metody prohledávání .....	20
2.3 Informované (heuristické) metody prohledávání .....	22
Shrnutí kapitoly.....	25
<b>3 Expertní systémy .....</b>	<b>27</b>
3.1 Charakteristika expertního systému .....	27
3.2 Struktura expertního systému.....	28
3.3 Typy expertních systémů .....	31
3.4 Využití expertních systémů.....	32
Shrnutí kapitoly.....	33
<b>4 Multiagentové systémy.....</b>	<b>35</b>
4.1 Agent.....	36
4.2 Architektura agenta .....	37
4.3 Multiagentové systémy .....	41
Shrnutí kapitoly.....	43
<b>5 Umělý život.....</b>	<b>45</b>
5.1 Umělý život.....	45
5.2 Celulární automaty .....	46
5.3 Historie CA .....	47
5.4 Hra života .....	49
Shrnutí kapitoly.....	51
<b>6 Deterministický chaos – principy a aplikace .....</b>	<b>53</b>
6.1 Popis deterministického chaosu .....	54
6.2 Turbulence.....	55
6.3 Atraktory .....	55
6.4 Motýlí efekt.....	59
6.5 Řízení deterministického chaosu.....	60
Shrnutí kapitoly.....	61
<b>7 Fraktály.....</b>	<b>65</b>
7.1 Úvod do fraktálů.....	65
7.2 Co je fraktál.....	66
7.3 Klasické fraktály .....	67
7.4 Typy fraktálů .....	70
7.5 Výskyt a využití fraktálů nejen v přírodě.....	71
Shrnutí kapitoly.....	79

<b>8</b>	<b>Reprezentace znalostí.....</b>	<b>83</b>
8.1	Reprezentace znalostí .....	83
8.2	Produkční systémy .....	85
8.3	Sémantické sítě.....	89
8.4	Rámcová reprezentace.....	91
	Shrnutí kapitoly .....	93
<b>9</b>	<b>Robotika .....</b>	<b>95</b>
9.1	Charakteristika robotů .....	96
9.2	Definice a klasifikace robotů.....	97
9.3	Příklady robotů .....	98
	Shrnutí kapitoly .....	100
<b>10</b>	<b>Zpracování přirozeného jazyka .....</b>	<b>101</b>
10.1	Jak mohou stroje slyšet a mluvit? .....	101
10.2	Rozpoznávání hlasu a řeči.....	105
	Shrnutí kapitoly .....	108
<b>11</b>	<b>Počítačové vidění .....</b>	<b>109</b>
11.1	Princip počítačového vidění .....	109
11.2	Využití počítačového vidění.....	112
	Shrnutí kapitoly .....	114
<b>12</b>	<b>OCR.....</b>	<b>115</b>
12.1	Optické rozpoznávání znaků .....	115
12.2	Části OCR programu .....	117
12.3	OCR algoritmy .....	119
12.4	Praktické využití OCR .....	120
	Shrnutí kapitoly .....	121
<b>13</b>	<b>Vybrané aplikace umělé inteligence .....</b>	<b>123</b>
13.1	Umělá inteligence v počítačových hrách .....	123
13.2	Umělá inteligence v dalších aplikacích .....	126
	Shrnutí kapitoly .....	129

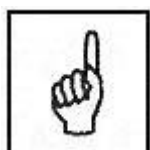
## Vysvětlivky k používaným symbolům



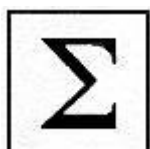
**Průvodce studiem** – vstup autora do textu, specifický způsob kterým se studentem komunikuje, povzbuzuje jej, doplňuje text o další informace.



**Příklad** – objasnění nebo konkretizování problematiky na příkladu ze života, z praxe, ze společenské reality apod.



**K zapamatování**



**Shrnutí** – shrnutí předcházející látky, shrnutí kapitoly.



**Literatura** – použitá ve studijním materiálu, pro doplnění a rozšíření poznatků.



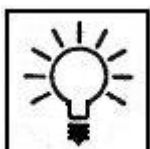
**Kontrolní otázky a úkoly** – prověřují, do jaké míry studující text a problematiku pochopil, zapamatoval si podstatné a důležité informace a zda je dokáže aplikovat při řešení problémů.



**Úkoly k textu** – je potřeba je splnit neprodleně, neboť pomáhají k dobrému zvládnutí následující látky.



**Korespondenční úkoly** – při jejich plnění postupuje studující podle pokynů s notnou dávkou vlastní iniciativy. Úkoly se průběžně evidují a hodnotí v průběhu celého kurzu.



**Otázky k zamyšlení**



**Část pro zájemce** – přináší látku a úkoly rozšiřující úroveň základního kurzu. Pasáže i úkoly jsou dobrovolné.

# Úvod

Tento text má sloužit pro potřeby výuky výběrového předmětu umělá inteligence na katedře informatiky a počítačů. Nepředpokládá se žádná předchozí znalost problematiky. Předmět má rysy kurzu, ve kterém student získá ucelenější pohled na problematiku umělé inteligence.

Cílem tohoto textu je seznámit studenty se základními problémy umělé inteligence a vymezit umělou inteligenci jako obor i jako soubor unikátních problémů a přístupů k řešení. Absolvent by měl získat potřebný nadhled pro další studium v hlouběji zaměřených předmětech.

## Po prostudování textu budete znát:

- základní problematiku z oblasti umělé inteligence,
- metody prohledávání stavového prostoru,
- úvod do problematiky expertních a multiagentových systémů,
- úvod do umělého života,
- úvod do problematiky reprezentace znalostí,
- úvod do robotiky,
- vybrané aplikace umělé inteligence.

## Získáte:

- ucelený pohled na základní problematiku umělé inteligence,
- potřebný nadhled pro další studium v hlouběji zaměřených předmětech.

## Korespondenční úkoly.

Vyberte si a vypracujte DVA korespondenční úkoly z kapitoly 1. - 13. této studijní opory.

Řešení obou korespondenčních úkolů zašlete do konce semestru (nejpozději před zkouškou) na adresu [eva.volna@osu.cz](mailto:eva.volna@osu.cz).

Pokud máte jakékoliv věcné nebo formální připomínky k textu, kontaktujte autory ([eva.volna@osu.cz](mailto:eva.volna@osu.cz), [martin.kotyrba@osu.cz](mailto:martin.kotyrba@osu.cz)).

# 1 Úvod do umělé inteligence

**V této kapitole se dozvíte:**

- Co se rozumí pojmem umělá inteligence
- Jaká je historie tohoto oboru
- Princip Turingova testu

**Po jejím prostudování byste měli být schopni:**

- Vysvětlit největší milníky z historie umělé inteligence
- Co samotná umělá inteligence obnáší
- Charakterizovat pojem umělá inteligence

**Klíčová slova kapitoly:** Umělá inteligence, DFKI, alternativní definice.

## Průvodce studiem

*Na zvládnutí této kapitoly budete potřebovat asi 1 hodinu, tak se pohodlně usadte a nenechte se nikým a ničím rušit. Tento učební text by vám měl především otevřít obzor do celé škály problémů, postupů a otázek, které spadají do oblasti umělé inteligence. I když existuje mnoho definic umělé inteligence, je velice těžké vymezit ji jako disciplínu s přesnými hranicemi jako je teoretická informatika, numerická matematika a podobně. Proto se nejprve pokusíme podívat se do historie tohoto oboru.*

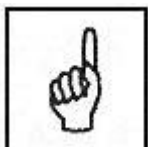


Inteligence je vlastností některých živých organismů. Vznikla a vyvíjela se v průběhu dlouhého časového intervalu a dnes umožňuje některým živým organismům efektivně reagovat na složité projevy prostředí a aktivně je využívat ve svůj prospěch a k dosažení svých cílů (Mikulecký, Ponce, 1996).

## 1.1 Historie umělé inteligence

Zařazení umělé inteligence jako oboru je vcelku obtížné. Lze na ni pohlížet jako na matematickou disciplínu s aplikacemi, nebo také jako na technický

obor. Závratný růst tohoto oboru ovlivňovalo mnoho faktorů, jako např. zvyšování požadavků v oblastech automatizovaného řízení, průzkumu nedosažitelných míst a řada dalších činností, kde je přítomnost člověka z technických či zdravotních důvodů vyloučena.



Umělá inteligence (UI) jako vědní disciplína se postupně formuje v posledních třiceti letech jako průsečík několika disciplín, jakými jsou např. psychologie, neurologie, kybernetika, matematická logika, teorie rozhodování, informatika, teorie her, lingvistika atd. Její vývoj není zdaleka ukončen. Dnes jí však již nikdo neupírá právo na samostatnou existenci. Přesto má však mezi ostatními vědními disciplínami poněkud specifické postavení, a to ze dvou důvodů. Za prvé, dosud neexistuje všeobecně přijímaná definice umělé inteligence; za druhé, umělá inteligence neposkytuje jednotící teorie - spíše volně sdružuje různorodé teorie, metody a techniky, které lze úspěšně používat k počítačovému řešení některých složitých úloh rozhodování, plánování, diagnostiky apod.

Počítačová UI byla ve větším měřítku použita poprvé v NASA, a to k řízení a kontrole letu dálkových družic. Asi všem je jasné, že základem UI je počítač. Bez počítače by nebylo UI a podle toho, jakou roli hraje počítač v existenci umělé inteligence, rozdělujeme UI na slabou - tam je počítač jen užitečným nástrojem a silnou - počítač není pouhým nástrojem, nýbrž přímo myslí a pomocí programu chápe. Možná někdy v budoucnu se objeví tak sofistikovaný počítač, který zvládne myslet sám o sobě.

Rok 1941, poprvé byl použit elektronický počítač. Bylo to v Německu. Počítač zabíral obrovské místnosti a byl chlazen leteckými motory. Převážně byl využíván pro vojenské účely. Jeho programování bylo pro programátory hotová noční můra.

Počátky roku 1950 Norbert Wiener přišel na to, že vlastně všechna inteligentní rozhodnutí jsou založena na principu zpětné vazby (feedback). Princip zpětné vazby použil už James Watt ve svém parním stroji, i když on asi netušil, že tento objev velmi ovlivní počátky vývoje UI.



Později v roce 1955, Newell a Simon vyvinuli The Logic Theorist, mnohými považovaný za první UI program. Tento program, který reprezentoval každý problém jako tree (stromový) model, se pokoušel vyřešit ho tak, že našel větev, jejíž výsledek byl s největší pravděpodobností ten správný. Dopad Logic Theorist na veřejnost a na pole UI, měl také rozhodující vliv na vývoj UI.

V roce 1956 John McCarthy, který je považován za otce UI, zorganizoval konferenci pro všechny zajímající se o strojovou inteligenci. Pozval je do Vermontu na The Dartmouth summer research project on artificial intelligence. Od této chvíle se oboru začalo říkat UI. I přes trochu rozpačitý průběh dartmouthská konference přivedla k sobě všechny objevitele UI a stala se výchozím bodem pro budoucnost UI.

Sedm let poté se pole UI dalo znovu do pohybu. Ideje z konference byly přezkoumány a přepracovány. Centra výzkumu se zformovala na Carnegie Mellon a MIT. Byla stanovena nová výzva: Vytvořit systém, který by efektivně řešil problémy, podobně jako Logic Theorist. A za druhé, vytvořit systém, který by se dokázal sám učit.

V roce 1957 se testovala první verze programu The General Problem Solver (GPS) . Program vytvořili ti samí lidé jako Logic Theorist. GPS byl rozšířením principu zpětné vazby a byl schopen řešit velké množství běžných problémů. Pár let nato, IBM vytvořilo tým na výzkum umělé inteligence. Herbert Gelerner strávil 3 roky práce na řešení geometrických teorémů.

V roce 1958 John McCarthy uvedl svůj nový objev, jazyk LISP, který se užívá dodnes. LISP ( LIST Processing - zpracování seznamů) se stal brzy jazykem UI vývojářů.

Dalším mezníkem ve vývoji UI byl rok 1963, kdy MIT dostala grant v hodnotě 2.2 milionů dolarů od Ministerstva obrany USA, aby zajistila technologickou výhodu proti Sovětskému svazu. Projekt posloužil k zvyšování tempa vývoje UI a k jeho financování.

V dalších letech vznikalo velké množství programů. Koncem šedesátých let vznikl třeba program STUDENT, který dokázal řešit algebraické problémy, nebo program SIR, který rozuměl jednoduchým anglickým větám. Výsledkem těchto programů bylo zdokonalení v porozumění jazykům a logice.

Sedmdesátá léta znamenala příchod expertních systémů. Expertní systémy předvídají pravděpodobnost řešení ve stanovených podmínkách. V těchto letech vzniklo mnoho nových metod. Například David Marr předložil novou teorii o strojovém vidění. Týkala se rozeznávání obrazů na základě odstínu, základních informací o tvarech, barvě, hranách a texturách. Další novinkou v roce 1972 byl jazyk PROLOG.

Osmdesátá léta znamenaly pro UI zrychlení vývoje. Zvýšila se poptávka po expertních systémech pro jejich účinnost. Podniky jako Digital Electronics užívaly XCON, expertní systém navržený pro programování na velkých VAX počítačích. Také firmy DuPont, General Motors a Boeing se silně spoléhaly na expertní systémy.

Samozřejmě že zároveň s vývojem se začalo využívat UI i v praxi. Například práce Minského a Marra měla za následek vznik kamer a počítačů pro kontrolu kvality na výrobních linkách. Jak jinak, také armáda měla zájem o UI. Ministerstvo obrany USA financovalo projekt na vývoj smart truck. Cílem bylo vytvořit robota, který by dokázal provést mnoho bitevních úkolů. Projekt byl velkým rozčarováním a proto Pentagon brzy jeho financování zastavil.



I přes některé neúspěchy se UI pomalu vyvíjela dál. Byly objeveny fuzzy logika a neuronové sítě a začalo se ukazovat, že UI se dá použít i v reálném životě (mimo jiné, UI systémy byly použity také ve válce během Písečné bouře - raketové systémy, heads-up-displays a jiné novinky byly prověřeny v praxi.) Dnes se umělá inteligence infiltrovala prakticky do všech vědních disciplín od ekonomie až po lékařství.

## 1.2 Umělá inteligence, jako pojem

V literatuře můžeme najít různé definice umělé inteligence. Pokusme se zde uvést alespoň nejdůležitější a nejmýstižnější z nich.

Marvin Minsky tvrdí, že „... umělá inteligence je věda o vytváření strojů nebo systémů, které budou při řešení určitého úkolu užívat takového postupu, který – kdyby ho dělal člověk – bychom považovali za projev jeho inteligence.“

(1967)

Tato definice vychází z Turingova testu.

Vyplývá z ní, že úlohy jsou tak složité, že i u člověka by vyžadovaly použití inteligence. Otázkou však je, jaké vlastnosti má složitost a ‚inteligentní‘ řešení? Složitost lze ohodnotit počtem všech řešení, které připadají v úvahu. Ale hledání řešení pouhým prohledáváním stavového prostoru možných řešení není možné u složitějších úloh ani prostřednictvím superrychlých počítačů. Tento postup navíc není možno nazvat inteligentním. Je tedy nutno omezit velikost množiny prohledávaných řešení, což se děje na základě využívání znalostí.

E. Richová se domnívá, že „... umělá inteligence se zabývá tím, jak počítačově řešit úlohy, které dnes zatím zvládají lidé lépe.“ (1991)

Tato definice se bezprostředně váže na aktuální stav v oblasti počítačových věd a je možno očekávat, že se v budoucnosti bude ohnisko této vědy posouvat a měnit. Nevýhodou této – jinak velmi výstižné – definice je fakt, že nezahrnuje úlohy, které dosud neumí řešit počítače, ale ani člověk.

### 1.2.1 Alternativní definice

- 1 UI je označení uměle vytvořeného jevu, který dostatečně přesvědčivě připomíná přirozený fenomén lidské inteligence.
- 2 UI označuje tu oblast poznávání skutečnosti, která se zaobírá hledáním hranic a možností symbolické, znakové reprezentace poznatků a procesů jejich nabývání, udržování a využívání.
- 3 UI se zabývá problematikou postupů zpracování poznatků – osvojováním a způsobem použití poznatků při řešení problémů.



Umělá inteligence je interdisciplinární vědou, která nemá pevně vymezený předmět zkoumání ani teoretický základ – jde spíše o soubor metod, teoretických přístupů a algoritmů, sloužících k řešení velmi složitých úloh.

### 1.3 Turingův test

Turingův test (pojmenovaný podle svého tvůrce Alana Turinga, který jej prezentoval roku 1950) je pokus, který má za cíl prověřit, jestli se nějaký systém umělé inteligence opravdu chová inteligentně. Jelikož inteligence je pojem, který lze jen těžko definovat, tím hůře testovat, používá Turingův test porovnání s člověkem.

#### Průběh testu

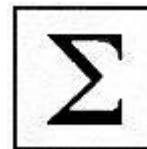
Turingův test probíhá tak, že do oddělených místností umístíme jednak testujícího, jednak předmět zkoumání (např. počítač s příslušným programem) a nějakého dalšího člověka. Testující poté klade otázky v přirozené řeči a předává je do druhé místnosti, kde je zodpoví buď počítač, nebo druhý člověk (což se rozhodne náhodně). Odpovědi jsou předávány zpět testujícímu (samozřejmě v nějaké neutrální podobě, např. vytištěné na papíře). Pokud testující nedokáže rozpoznat, jestli komunikuje se strojem, nebo s člověkem, pak tato umělá inteligence splňuje Turingův test.

#### Historie a překonávání testu

Turingův test byl dlouho považován za základní měřítko schopností uměle inteligentní entity, avšak zdaleka nepokrývá všechny aspekty, které jsou od inteligentních entit očekávány. Jeden z nedostatků tohoto testu ilustruje argument čínského pokoje. Jedním ze zásadních zlomů nastal v roce 1966, kdy byl představen program ELIZA Josepha Weizenbauma, který test částečně splnil, byť se nejednalo o inteligentní entitu, ale o program, který dnes nazýváme chatterbot. Navzdory tomu Loebnerova cena 100 000 dolarů pro první počítač, jehož odpovědi budou od člověka zcela nerozlišitelné, dosud nebyla udělena.

## Shrnutí kapitoly

- Předem je důležité, aby si student uvědomil především základní definici umělé inteligence a následně i její široké možnosti použití. Umělá inteligence se jako obor infiltrovala prakticky do všech vědních disciplín, např. jako podpůrné softwarové vybavení. Pro člověka, který má nutkání vše idealisticky zjednodušovat je využití inteligentních principů téměř všední věcí.



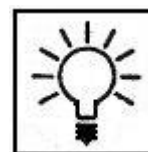
## Kontrolní otázky a úkoly:

1. Co je umělá inteligence?
2. Co je voxel?
3. Existuje více definic umělé inteligence?
4. Kde se využívá inteligentních principů?
5. Proč je vhodné využít inteligentní princip?



## Otázky k zamyšlení:

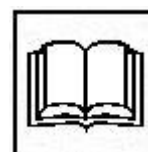
1. Zjistěte, jak se využívá umělá inteligence v automobilovém průmyslu a v ekonomii.
2. Víte co je to DeepBlue a jak využívá inteligence?



## Citovaná a doporučená literatura

Habiballa, H. *Umělá inteligence*. Texty pro distanční studium. Ostravská univerzita. Ostrava 2004.

Mařík, V., Štěpánková, O., Lažanský, J. *Umělá inteligence (4)*, Academia, Praha 2003.





## 2 Řešení úloh a prohledávání stavového prostoru

**V této kapitole se dozvíte:**

- Co je to stavový prostor.
- Jaké jsou metody prohledávání stavového prostoru.

**Po jejím prostudování byste měli být schopni:**

- objasnit pojem stavový prostor,
- charakterizovat neinformované (slepé) metody prohledávání stavového prostoru.
- charakterizovat informované (heuristické) metody prohledávání stavového prostoru.

**Klíčová slova kapitoly:** Stavový prostor, neinformované prohledávání, informované prohledávání, heuristika, algoritmus A-Star, rozklad úlohy na podproblémy.

### **Průvodce studiem**

*Stavový prostor si můžeme představit jako orientovaný graf (strom), jehož uzly představují jednotlivé stavy úloh a jehož hrany představují přechody mezi těmito stavy. Cesta z počátečního stavu do některého cílového stavu je řešením úlohy. Metody jak najít řešení ve stavovém prostoru jsou v principu nastíněny v této kapitole.*

*Na zvládnutí této kapitoly budete potřebovat asi 4 hodiny.*



Je-li dán počáteční model prostředí a požadovaný koncový model prostředí, pak je úkolem inteligentních strojů a systémů vyhledat vhodnou posloupnost akcí, jejichž aplikací lze přejít od počátečního modelu k cílovému. Taková posloupnost se nazývá *plán* a metody vytváření plánů se označují jako *metody řešení úloh*. Každému modelu odpovídá jistý stav prostředí, množina všech stavů tvoří *stavový prostor*.

## 2.1 Stavový prostor a jeho prohledávání



Stavový prostor je určen:  $SP = \langle S, O \rangle$

- Konečnou neprázdnou množinou stavů  $S$  (ve zcela obecném případě i nekonečnou).
- Neprázdnou konečnou množinou operátorů  $O$ , kde každý operátor je partiální funkcí na stavovém prostoru (nemusí být definován všude)  
 $\alpha: S \rightarrow S$ .

Úloha ve stavovém prostoru  $SP$  je  $\langle s_0, C \rangle$ , kde

- $s_0$  je počáteční stav,
- $C$  je množina požadovaných cílových stavů.

Plán (řešení)  $P$  pro danou úlohu  $U$  je taková posloupnost operátorů  $\langle \alpha_1, \alpha_2, \dots, \alpha_n \rangle$ , která splňuje následující podmínky

$$s_1 = \alpha_1(s_0)$$

$$s_2 = \alpha_2(s_1) \dots$$

$$s_n = \alpha_n(s_{n-1}) = \alpha_{n-1}(\alpha_{n-2}(\dots(\alpha_2(\alpha_1(s_0))))),$$

přičemž  $s_n$  je definováno a je prvkem množiny cílových stavů  $C$ .



**Příklad** (Habiballa 2004):

Jsou dány dvě nádoby, větší  $A$  o obsahu  $a$  litrů, a menší  $B$  obsahu  $b$  litrů:

- na začátku jsou obě prázdné (= počáteční stav)
- cílem je stav, kdy nádoba  $A$  je prázdná a v  $B$  je přesně  $2 \cdot (a - b)$  litrů vody

K dispozici je neomezený zdroj vody a nádoby nemají označené míry.

Naplňte nádobu  $B$  (= nalezněte posloupnost akcí, že bude splněn daný cíl).

*Reprezentace stavů:*

Zvolení vhodné reprezentace je jednou z důležitých podmínek efektivního vyřešení konkrétního problému. Stavů jsou v naprosté většině případů generovány v průběhu algoritmu

- Stav: dvojice  $\langle c_A, c_B \rangle$  množství vody v nádobách  $A, B$ 
  - počáteční stav  $\langle 0, 0 \rangle$
  - jediný cílový stav  $\langle 0, 2 \cdot (a - b) \rangle$
- přechody (= operátory):



- vylití  $A$ :  $\langle c_A, c_B \rangle \rightarrow \langle 0, c_B \rangle$
- vylití  $B$ :  $\langle c_A, c_B \rangle \rightarrow \langle 0, c_{BA}, 0 \rangle$
- naplnění  $A, B$
- přelití  $A$  do  $B$
- $\langle c_A, c_B \rangle \rightarrow \langle \max(c_A - (b - c_B), 0), \min(c_A + c_B, b) \rangle$
- možné jsou samozřejmě i další operátory

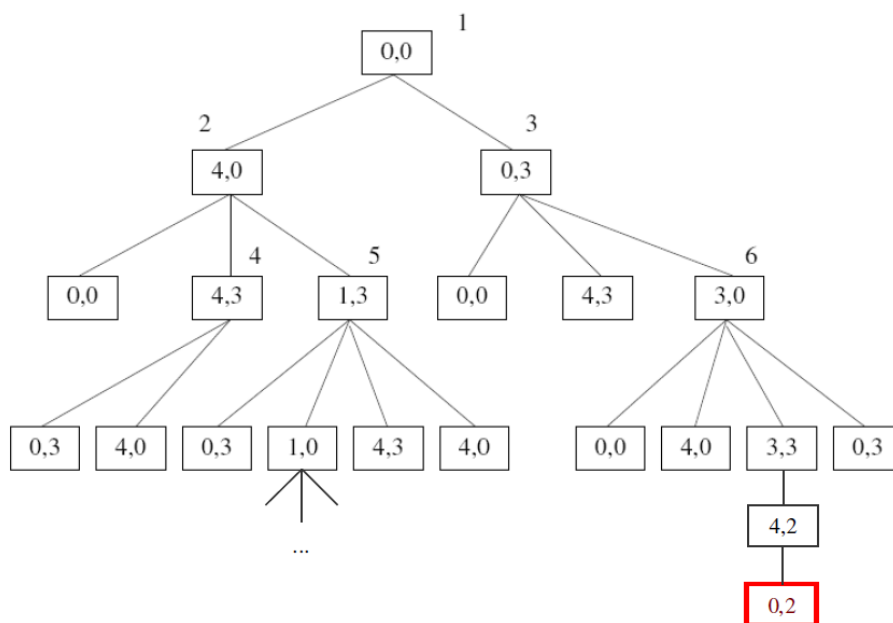
### Prohledávání stavového prostoru

Stavový prostor lze reprezentovat orientovaným grafem  $G$  (stavovým grafem, stromem)  $G = (V; E)$ ,  $V$ : vrcholy (uzly),  $E$ : hrany

- uzel reprezentuje stav,
- hrana reprezentuje přechod mezi stavy.

Řešení úloh pak lze formulovat jako hledání přijatelné cesty v orientovaném grafu z počátečního uzlu do některého z cílových uzlů (viz obr. 2.1).

*Pozn.:* k jednotlivým hranám je často přiřazena i cena vykonání daného přechodu.



Obrázek 2.1: Stavový prostor úlohy přelévání vody (převzato z

<https://cw.felk.cvut.cz>)

### Způsoby prohledávání stav. prostoru

1) Slepé (neinformované prohledávání):

- do šířky – uzly k expanzi řadíme do fronty, náročné na paměť,

- do hloubky – uzly řadíme do zásobníku; nutno ošetřit cykly.

2) Heuristické (informované prohledávání) - pořadí prohledávání na základě "další" informace, odhadu vzdálenosti stavu od cíle, vyjádřeného tzv. heuristikou  $h(\text{uzel})$ .

- paprskové (beam search),
- gradientní (hill-climbing),
- uspořádané (best-first),
- A-Star (A\*).

### Dle čeho hodnotit jednotlivé algoritmy:

Pokud má být algoritmus výběru cesty úspěšný musí splňovat dvě základní vlastnosti:

- Musí vést k prohledávání, což znamená, že se musí procházet stavovým prostorem a přitom se vyhnout cyklům.
- Musí být systematický.

### Při rozlišování mezi jednotlivými algoritmy se uplatňují zejména tyto kritéria:

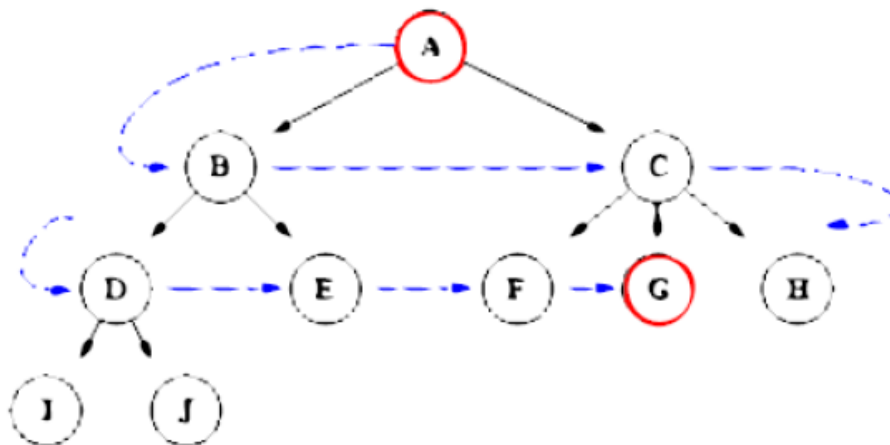
- *Úplnost*: garantuje algoritmus nalezení řešení (pokud toto existuje).?
- *Časová složitost*: jak dlouho to potrvá – asymptotická reprezentace složitosti.
- *Paměťová náročnost*: kolik paměti je zapotřebí.
- *Optimalita*: nalezne algoritmus to nejlepší řešení v případě existence více řešení.

## 2.2 Neinformované (slepé) metody prohledávání

Princip prohledávání do šířky spočívá v tom, že na každé úrovni vždy vygenerujeme všechny možnosti a ty teprve řešíme. Cena (ohodnocení) uzlu se rovná jeho hloubce. Uzly se expandují v pořadí, v jakém byly vygenerovány. Metoda nalezne vždy nejkratší cestu k cíli, pokud tato cesta existuje.

**Prohledávání do šířky** (obr. 2.2)

- 1)  $CLOSED = \{ \}$ ,  $OPEN = \{s_0\}$ , kde  $s_0$  je počáteční stav. Je-li  $s_0$  současně i cílový stav, pak ukonči prohledávání.
- 2) Je-li  $OPEN$  prázdný, pak řešení neexistuje  $\rightarrow$  ukonči prohledávání.
- 3) Vyber a současně vymaž první stav ze seznamu  $OPEN$ , označ jej  $s_i$  a zapiš jej do seznamu  $CLOSED$ .
- 4) Expanduj stav  $s_i$ . Pokud stav  $i$  nemá následovníky nebo všichni následovníci byli již expandováni (tj. jsou v seznamu  $CLOSED$ ), jdi na krok 2.
- 5) Zapiš všechny následovníky stavu  $i$ , kteří nejsou v seznamu  $CLOSED$  na konec seznamu  $OPEN$ .
- 6) Pokud některý z následovníků stavu je cílovým stavem, tj. řešení bylo právě nalezeno, ukonči prohledávání, jinak pokračuj krokem č. 2.



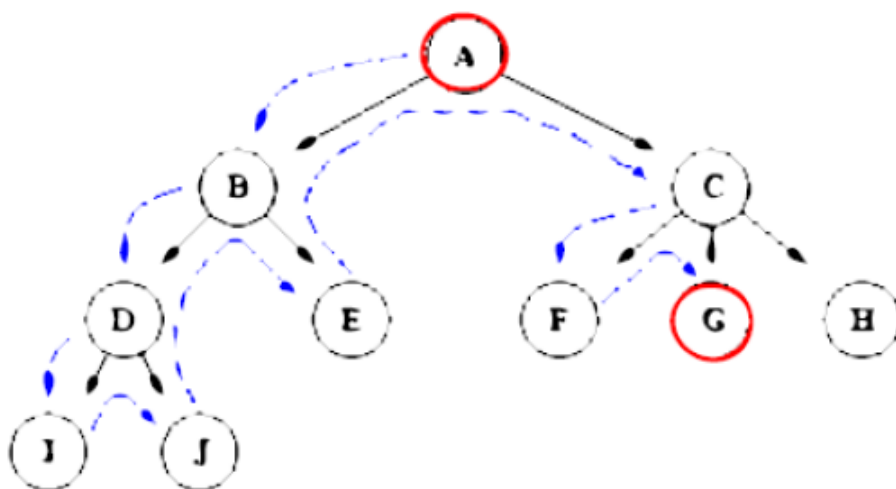
Obrázek 2.2: Prohledávání do šířky (převzato z <https://cw.felk.cvut.cz>)

**Prohledávání do hloubky** (s omezením max na délku větve), obr. 2.3

- 1)  $CLOSED = \{ \}$ ,  $OPEN = \{s_0\}$ , kde  $s_0$  je počáteční stav. Je-li  $s_0$  současně i cílový stav, pak ukonči prohledávání.
- 2) Je-li  $OPEN$  prázdný, řešení neexistuje  $\rightarrow$  ukonči prohledávání.
- 3) Vyber a současně vymaž první stav ze seznamu  $OPEN$ , označ jej  $s_i$  a zapiš jej do seznamu  $CLOSED$

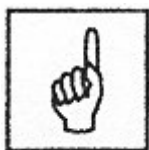


- 4) Pokud se hloubka uzlu  $i$  rovná maximální přípustné hloubce  $max$ , pokračuj krokem 2.
- 5) Expanduj stav  $s_i$ . Pokud stav  $s_i$  nemá následovníky nebo všichni následovníci byli již expandováni (tj. jsou v seznamu CLOSED), jdi na krok 2.
- 6) Zapiš všechny následovníky stavu  $s_i$ , kteří nejsou v seznamu CLOSED na začátek seznamu OPEN.
- 7) Pokud některý z následovníků stavu je cílovým stavem, tj. řešení bylo právě nalezeno, ukonči prohledávání, jinak pokračuj krokem č. 2



Obrázek 2.3: Prohledávání do hloubky (převzato z <https://cw.felk.cvut.cz>)

### 2.3 Informované (heuristické) metody prohledávání



Informované hledání je založeno na strategii, která využívá specifické znalosti pro daný problém, a nalezení cíle je mnohem efektivnější, včetně lepší možnosti dosáhnout optimálního řešení. Tyto znalosti se nazývají heuristické znalosti nebo také **heuristiky**.

Heuristické znalosti můžeme využít dvěma způsoby (Habiballa 2004):

- První způsob je zřejmější a spočívá v zahrnutí těchto znalostí do pravidel.
- Druhý způsob je založen na doplnění ohodnocující funkce o předpokládanou cenu k cíli:

$$f^*(i) = g^*(i) + h^*(i),$$

$f^*(i)$  je ohodnocení uzlu  $i$

$g^*(i)$  je dosavadní cena cesty z počátečního uzlu do uzlu  $i$

$h^*(i)$  je předpokládaná cena cesty z uzlu  $i$  do cílového uzlu

**Algoritmus A-Star** (Russel, Norvig 2003)

Algoritmus A-Star je grafový algoritmus sloužící k nalezení „nejkratší“ cesty v ohodnoceném grafu vedoucí ze zadaného počátečního uzlu do uzlu cílového. Jeho vstupem je ohodnocený graf, počáteční uzel a cílový uzel, výstupem je nejkratší cesta z počátečního uzlu do cílového, nebo zpráva o tom, že žádná taková cesta neexistuje.

Algoritmus je v principu shodný s prohledáváním do šířky s tím rozdílem, že namísto obyčejné fronty používá frontu prioritní, ve které jsou cesty seřazeny podle hodnoty speciální funkce  $f$ . Tato funkce je definována pro každou cestu  $p$  a je součtem tzv. heuristické funkce ( $h$ ) posledního uzlu cesty  $p$  a její délky ( $g$ ). Čím je hodnota funkce  $f(p)$  nižší, tím vyšší má daná cesta  $p$  prioritu. Dále se předpokládá, že cesty ve frontě neobsahují kružnice a pro každý cílový uzel se v ní nachází nanejvýš jedna cesta, a to ta nejkratší doposud nalezená.

Kroky algoritmu A-Star:

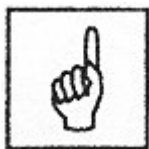
- Vytvoř prázdnou množinu cest  $F$ .
- Do množiny  $F$  vlož cestu nulové délky obsahující počáteční uzel  $s$ .
- Dokud není množina  $F$  prázdná, opakuj:
  - Z množiny  $F$  vyber nejkratší cestu  $p$  (s nejnižší hodnotou  $f(p)$ ) a odeber ji.
  - Končí-li cesta v cílovém uzlu, vrať ji a ukonči výpočet.
  - Vytvoř nové cesty použitím všech možných operátorů na koncový uzel cesty  $p$ , které neobsahují smyčky.
  - Jestliže dvě cesty končí ve stejném uzlu, odstraň všechny kromě té nejkratší (s nejnižší hodnotou  $f(x)$ ).
  - Přidej cestu  $p$  do množiny  $F$ .
- Je-li množina  $F$  prázdná, oznam, že žádná cesta z počátečního do cílového uzlu neexistuje.



Asymptotická složitost algoritmu A-Star silně závisí na použité heuristické funkci, která ovlivňuje množství cest uvažovaných během výpočtu.

Asymptotická složitost vyjadřuje, jak roste náročnost algoritmu (doba výpočtu

nebo potřebná paměť) s rostoucím množstvím vstupních dat. Obecně se však dá říci, že asymptotická složitost algoritmu A-Star nikdy nebude horší než složitost prohledávání do šířky.



### Rozklad úlohy na podproblémy

Rozklad úlohy na podproblémy lze opět znázornit grafem, ale na rozdíl od předcházející metody neodpovídají nyní uzly grafu stavům úlohy, ale podúlohám (podproblémům). Každý podproblém se opět rozkládá na jednodušší podproblémy až konečně listy grafu odpovídají buď elementárním úlohám nebo úlohám neřešitelným.

Druhý rozdíl od grafů stavových prostorů spočívá v tom, že rozeznáváme dva typy bezprostředních následníků – uzly AND a uzly OR (větve k následníkům typu AND se spojují obloučkem). Uzel je řešitelný pokud jsou řešitelní všichni jeho bezprostřední následníci typu AND, resp. je-li řešitelný alespoň jeden bezprostřední následník typu OR. V opačném případě je uzel neřešitelný. Metody řešení rozkladem na podproblémy jsou vlastně metody prohledávání AND/OR grafů a můžeme je opět rozdělit na slepé (do šířky a do hloubky) a heuristické.



### Příklad:

Rozklad problému na podproblémy ilustrujme na příkladu hry NIM.

Pravidla:

- hráč zadá počet zápalek ( např. od 15 do 35 )
- pak se střídá se strojem v odebírání; odebrat lze 1, 2 nebo 3 zápalky,
- prohraje ten, kdo odebere poslední zápalku.

Dílčí podproblémy:

- zadání počtu zápalek
- odebrání zápalek hráčem
- odebrání zápalek strojem

Hrubé řešení:

```
int pocet;
```

```
boolean stroj = false;
```

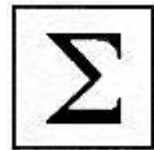
```
“zadani_pocetu_zapalek“
```

```
do {  
    if ( stroj )  
        „odebrani_zapalek_strojem“  
    else  
        „odebrani_zapalek_hracem“  
    stroj = !stroj;  
} while ( pocet>0 );  
if ( stroj ) „vyhral_stroj“  
else „vyhral_hrac“
```

Podproblémy „*zadani\_poctu\_zapalek*“, „*odebrani\_zapalek\_strojem*“ a „*odebrani\_zapalek\_hracem*“ budeme realizovat metodami. Proměnné *pocet* a *stroj* pro ně budou statickými (čili nelokálními) proměnnými.

## Shrnutí kapitoly

V této kapitole jsme se věnovali problematice řešení úloh a prohledávání stavového prostoru. Stavový prostor si můžeme představit jako orientovaný graf (strom), jehož uzly představují jednotlivé stavy úloh a jehož hrany představují přechody mezi těmito stavy. Cesta z počátečního stavu do některého cílového stavu je řešením úlohy. Dále byly popsány neinformované (slepé) a informované (heuristické) metody prohledávání stavového prostoru.



### Kontrolní otázky a úkoly:

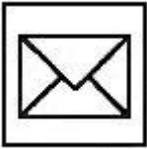
1. Co je to stavový prostor?
2. Jaké jsou metody prohledávání stavového prostoru?
3. Jaký je princip A-Star algoritmu?
4. Charakterizujte rozklad úlohy na podproblémy.



### Úkoly k textu

Nalezněte na Internetu další algoritmy prohledávání stavového prostoru a porovnejte je.





### **Korespondenční úkoly**

Navrhněte, vytvořte a odlaďte program pro řešení jednoduchého problému pomocí zvoleného algoritmu prohledávání stavového prostoru.

Příklady:

- Mějme hrací plochu o 9 polích, obsahující kameny s čísly 1 až 8. Kameny jsou na hrací ploše náhodně rozmístěny a úkolem je posunem kamenů dosáhnout zvoleného cílového stavu.
  - Cesta bludištěm.
  - Problém obchodního cestujícího.
  - Hanojské věže.
- apod.



### **Citovaná a doporučená literatura**

Russel, S. Norvig, P., *Artificial Intelligence: A Modern Approach*. 2. vyd. New Jersey, USA : Prentice Hall, 2003. ISBN 0-13-790395-2.

Habiballa, H. *Umělá inteligence*. Texty pro distanční studium. Ostravská univerzita. Ostrava 2004.



## 3 Expertní systémy

**V této kapitole se dozvíte:**

- Co je to expertní systém
- K čemu jsou expertní systémy určeny
- Jaká je struktura expertních systémů
- Jaké typy úloh mohou expertní systémy řešit

**Po jejím prostudování byste měli být schopni:**

- Definovat pojem expertní systém
- Znat základní rozdělení a charakteristiku expertních systémů
- Vědět, kde se expertní systémy uplatňují

**Klíčová slova kapitoly:** Expertní systém, inferenční mechanismus, báze znalostí.

### Průvodce studiem

*V této kapitole se podíváme na problematiku expertních systémů. Dozvíte se co je to expertní systém, kde se s ním můžete v praxi setkat, jaké typy úloh zpracovává, jaká je jeho vnitřní struktura a jaké typy expertních systémů existují. Na zvládnutí této kapitoly budete potřebovat asi 2 hodiny, tak se pohodlně usadte a nenechte se nikým a ničím rušit.*



### 3.1 Charakteristika expertního systému

Feigenbaum definuje expertní systém jako počítačový program, simulující rozhodovací činnost experta při řešení složitých úloh a využívající vhodně zakódovaných, explicitně vyjádřených znalostí, převzatých od experta, s cílem dosáhnout ve zvolené problémové oblasti kvality rozhodování na úrovni experta.

Expertní systémy se vyznačují následujícími charakteristickými rysy (typickým je první rys, další rysy jsou žádoucí, ale nemusejí být vždy přítomny):

(1) oddělení znalostí a mechanismu jejich využívání (tím se expertní systémy

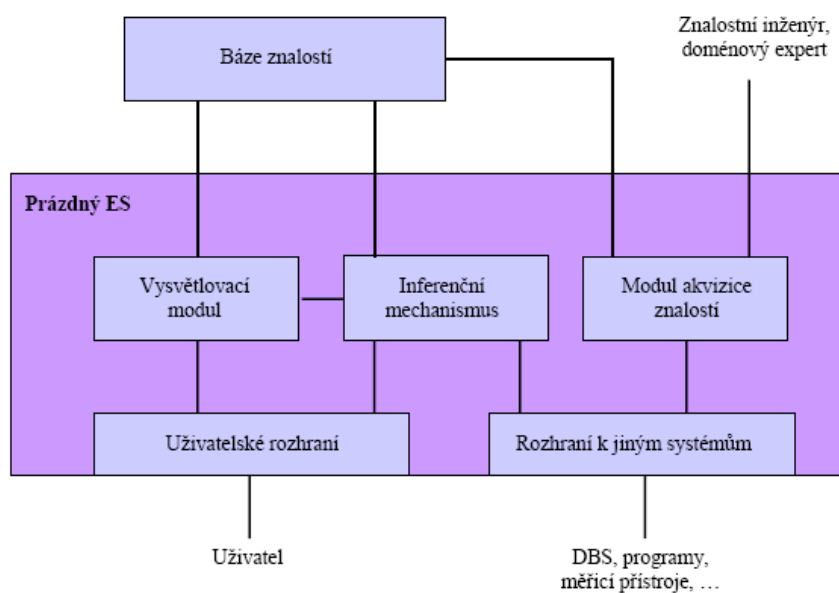
odlišují od klasických programů), (2) schopnost rozhodování za neurčitosti, (3) schopnost vysvětlování.

V literatuře se můžeme setkat také s pojmem znalostní systém (*knowledge-based system*), který je podle staršího pojetí obecnější než pojem expertní systém. Expertní systém tedy lze chápat jako zvláštní typ znalostního systému, který se vyznačuje používáním expertních znalostí a některými dalšími rysy, jako je např. vysvětlovací mechanismus. V poslední době však dochází ke stírání rozdílů mezi těmito pojmy.

### 3.2 Struktura expertního systému

Expertní systém obsahuje tyto základní složky (jejich vzájemné vazby ukazuje obr. 3.1):

- bázi znalostí,
- inferenční mechanismus,
- I/O rozhraní (uživatelské, vývojové, vazby na jiné systémy),
- vysvětlovací modul,
- modul pro akvizici (získávání) znalostí.



Obrázek 3.1: Architektura expertního systému

#### 3.2.1 Báze znalostí

Zatímco u konvenčních programů jsou znalosti specialistů nejrůznějším způsobem „roztroušeny“ v jednotlivých instrukcích programu, které se aplikují v předem stanoveném pořadí, u expertních systémů jsou znalosti experta

vyjádřeny naprosto explicitně, v podobě tzv. báze znalostí a předem je dána pouze strategie využívání znalostí z této báze – inferenční (řídící) mechanismus. Striktní oddělení znalostí a řídící strategie pro jejich využívání, tj. existence zřetelně oddělených samostatných komponent báze znalostí a řídícího mechanismu, je typické právě pro architekturu expertních systémů. Tato vlastnost je základem vysokého stupně znouvopoužitelnosti jednou vyvinutého a odzkoušeného řídícího mechanismu (a dalších podpůrných komponent) (Poper, 1988).

Báze znalostí obsahuje znalosti experta (expertů) potřebné k řešení zvoleného problému. Znalosti mohou být nejrůznějšího charakteru:

- od nejobecnějších znalostí až po znalosti vysoce specifické
- od „učebnicových“ znalostí až po znalosti „soukromé“ (tj. takové, jaké by si expert ani nedovolil publikovat)
- od exaktně prokázaných znalostí až k nejistým heuristikám
- od jednoduchých znalostí až po metaznalosti (tj. znalosti o znalostech)

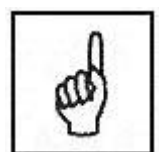
Znalosti experta nemají statický charakter, nýbrž se postupně (a obvykle nikoliv pomalým tempem) vyvíjejí a rozrůstají. Přirozeným omezením, kladeným na bázi znalostí, je proto požadavek její vysoké modularity tak, aby bylo možné jednoduše zahrnovat přírůstky nových znalostí. Báze znalostí musí být též transparentní, čitelná pro experta (aby ji mohl upravovat a rozšiřovat), ale i pro další odborné pracovníky, kteří se jejím prostřednictvím mohou učit, doučovat či doškoloovat (Pokorný, 2004).

Báze znalostí popisuje znalosti z dané oblasti. Má tedy charakter obecného rozhodovacího pravidla (či systému rozhodovacích pravidel). Řešit konkrétní případ znamená „dosadit“ data o daném případě do obecně formulovaných znalostí z báze znalostí.

U některých expertních systémů je dialog s uživatelem (částečně nebo úplně) nahrazen přímým měřením údajů na reálných objektech či jejich vyhledáváním v databankách. Množinu všech údajů k danému případu nazýváme – na rozdíl od báze znalostí – **bází dat** (Pokorný, 2004).

Pro reprezentaci znalostí jsou nejčastěji používány tyto prostředky:

- matematická logika,
- pravidla (rules),
- rozhodovací stromy (decision trees)



- sémantické sítě (semantic nets),
- rámce a scénáře (frames and scripts),
- objekty (objects).



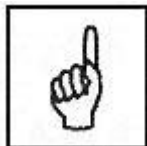
Znalosti můžeme klasifikovat jako:

- *mělké* (shallow knowledge), které jsou založeny na empirických a heuristických znalostech.
- *hluboké* (deep knowledge), založené na základních strukturách, funkcích a chování objektů.

Obdobně můžeme i usuzování charakterizovat jako mělké (shallow reasoning) nebo hluboké (deep reasoning). Pro hluboké usuzování jsou užitečnými modely rámce a objekty.

V průběhu řešení konkrétního problému se vytváří báze faktů, která obsahuje data k řešenému problému (vstupní údaje a postupně odvozované výsledky).

### 3.2.2 Inferenční mechanismus



Inferenční mechanismus obsahuje obecné (doménově nezávislé) algoritmy schopné řešit problémy na základě zadaných faktů pomocí manipulace se znalostmi z báze znalostí. Typický inferenční mechanismus je založen na inferenčním pravidle pro odvozování nových poznatků z existujících znalostí, strategii prohledávání báze znalostí.

V (Giarratano a Riley 1998) je uveden následující přehled metod inference:

- *Dedukce* – logické usuzování, při němž závěry musejí vyplývat z předpokladů.
- *Indukce* – postup od specifického případu k obecnému.
- *Abdukce* – usuzování směřující ze správného závěru k předpokladům, které jej mohly způsobit.
- *Heuristiky* – pravidla „zdravého rozumu“ založená na zkušenostech.
- *Generování a testování* – metoda pokusů a omylů.
- *Analogie* – odvozování závěru na základě podobnosti s jinou situací.
- *Defaultní inference* – usuzování na základě obecných znalostí v případě absence znalostí specifických.

- *Nemonotonní inference* – je možná korekce resp. ústup od dosavadních znalostí na základě nového pozorování.
- *Intuice* – obtížně vysvětlitelný způsob usuzování, jehož závěry jsou možná založeny na nevědomém rozpoznání nějakého vzoru. Tento typ usuzování zatím nebyl v ES implementován a snad by se k němu mohlo blížit usuzování neuronových sítí (Dvořák, 2004).

Důležitou schopností inferenčního mechanismu je zpracování neurčitosti.

Neurčitost v expertních systémech se může vyskytovat jednak v bázi znalostí a jednak v bázi faktů. Zdroji neurčitosti jsou: nepřesnost, nekompletnost, nekonzistence dat, vágní pojmy, nejisté znalosti.

Neurčitost může být reprezentována a zpracovávána např. pomocí následujících přístupů a prostředků: pravděpodobnostní (Bayesovské) přístupy, faktory jistoty, Dempster-Shaferova teorie, fuzzy logiky.

### 3.3 Typy expertních systémů

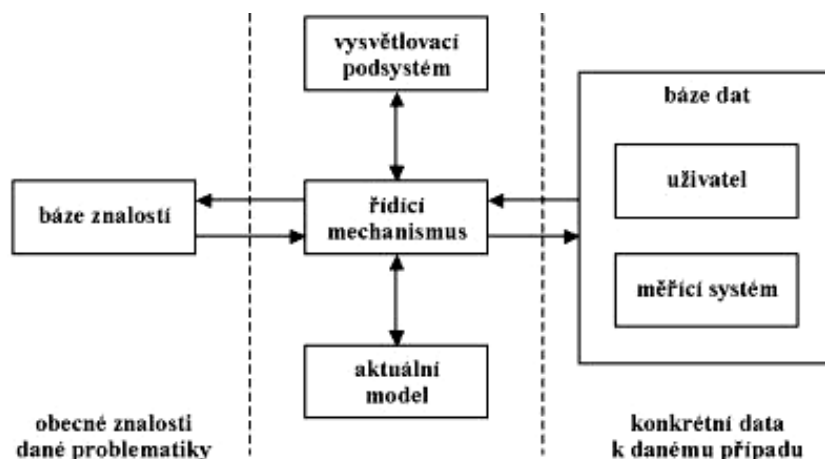
Expertní systémy můžeme klasifikovat podle různých hledisek. Podle obsahu báze znalostí můžeme expertní systémy rozdělit na:

- *problémově orientované*, jejichž báze znalostí obsahuje znalosti z určité domény.
- *prázdné* (shells), jejichž báze znalostí je prázdná.

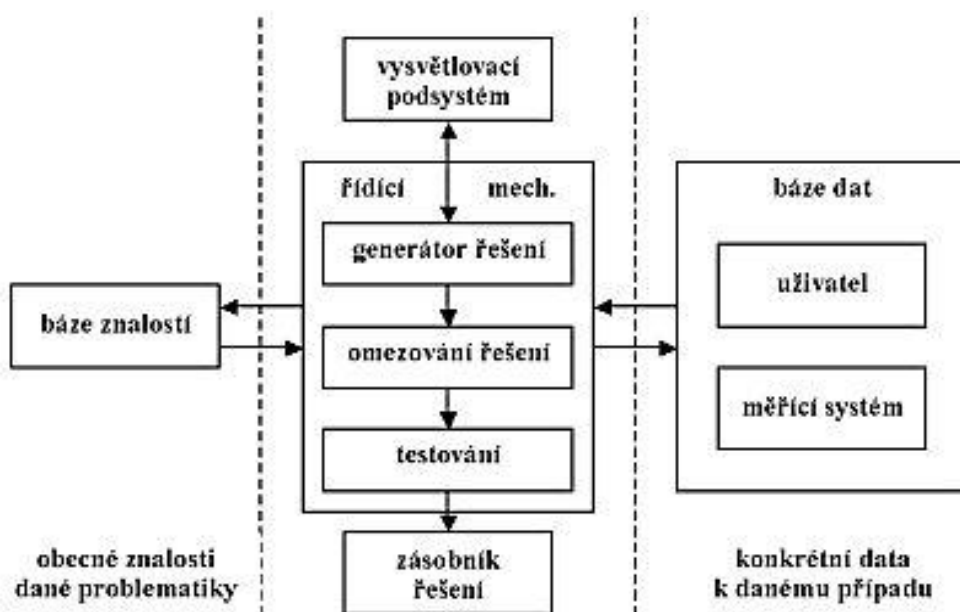


Podle charakteru řešených problémů můžeme expertní systémy rozdělit na :

1. *diagnostické*, jejichž úkolem je určit, která z hypotéza z předem definované konečné množiny cílových hypotéz nejlépe koresponduje s daty týkajícími se daného konkrétního případu (Dvořák, 2004).



Obrázek 3.2: Blokové schéma diagnostického expertního systému - *plánovací*, které obvykle řeší takové úlohy, kdy je znám cíl řešení a počáteční stav a je třeba s využitím dat o konkrétním řešeném případě nalézt posloupnost kroků, kterými lze cíle dosáhnout.



Obrázek 3.3: Blokové schéma plánovacího expertního systému

### 3.4 Využití expertních systémů



**Monitorování** – ohodnocování chování systému na základě porovnání s daty očekávanými a daty pozorovanými.

**Diagnostika** – nalezení důvodů chybného chodu systému na základě výsledků pozorování (diagnostika strojů a zařízení, lékařská diagnostika).

**Interpretace** – objasnění pozorovaných dat (stanovení struktur sloučenin a interpretace chemických dat).

**Plánování** – předepsání sledu činností pro získání žádaného výsledku (vojenská strategie a taktika).

**Konfigurace** – zkompletování patřičných komponent systému vhodným způsobem.

**Prognózování** – předpovídání pravděpodobných důsledků zadaných situací (ekonomické odhady, meteorologie).

**Učení** – inteligentní výučování, studující smí klást otázky typu: proč, kdy, co kdyby (výuka a školení) apod.

**Ladění** – vytvořit ustanovení pro odstranění závad systému.

**Řízení** – regulace procesů (může zahrnovat plánování, diagnostiku, monitorování, interpretaci, prognózování a ladění) (Dvořák, 2004).

Příklady aplikací expertních systémů je možno najít např. v knize (Turban, 1992).

*Výhody expertních systémů:*

- schopnost řešit složité problémy,
- dostupnost expertíz a snížené náklady na jejich provedení,
- trvalost a opakovatelnost expertíz,
- trénovací nástroj pro začátečníky,
- uchování znalostí odborníků odcházejících z organizace.

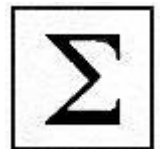


*Nevýhody expertních systémů:*

- nebezpečí selhání ve změněných podmínkách,
- neschopnost poznat meze své použitelnosti.

## Shrnutí kapitoly

- Feigenbaum definuje expertní systém jako počítačový program, simulující rozhodovací činnost experta při řešení složitých úloh a využívající vhodně zakódovaných, explicitně vyjádřených znalostí, převzatých od experta, s cílem dosáhnout ve zvolené problémové oblasti kvality rozhodování na úrovni experta. Báze znalostí obsahuje znalosti experta (expertů) potřebné k řešení zvoleného problému. Báze znalostí popisuje znalosti z dané oblasti. Má tedy charakter obecného



rozhodovacího pravidla (či systému rozhodovacích pravidel). Řešit konkrétní případ znamená „dosadit“ data o daném případě do obecně formulovaných znalostí z báze znalostí. Inferenční mechanismus obsahuje obecné (doménově nezávislé) algoritmy schopné řešit problémy na základě zadaných faktů pomocí manipulace se znalostmi z báze znalostí. Typický inferenční mechanismus je založen na inferenčním pravidle pro odvozování nových poznatků z existujících znalostí a strategii prohledávání báze znalostí.



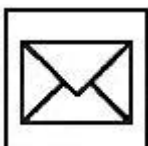
### Kontrolní otázky a úkoly:

1. Co je expertní systém?
2. K čemu slouží báze znalostí?
3. Co je to inferenční mechanismus?
4. Kde se využívají expertní systémy?



### Otázky k zamyšlení:

1. Kde by jste mohli ve svém životním stylu využít expertní systém, k čemu by sloužil a jakého typu a charakteru by expertní systém byl?



### Korespondenční úkoly

1. V systému LMPS vytvořte plnohodnotný expertní systém. Vhodné téma prokonzultujte s vyučujícím.



### Citovaná a doporučená literatura

(Beckman, 1997) Beckman, T. A., *Methodology for Knowledge Management*, IASTED AI and Software Computing Conference, Banff, Canada, 1997.

(Dvořák, 2004) Dvořák, J. *Expertní systémy*. Ústav automatizace a informatiky, 2004. 92 s. Skripta. Vysoké učení technické v Brně.

(Pokorný, 2004) POKORNÝ, Miroslav. *Expertní systémy*. Ostravská univerzita v Ostravě, 2004. 103 s. Skripta. Ostravská univerzita v Ostravě.

(Turban, 1992) TURBAN, E. - *Expert Systems and applied Artificial Intelligence*, Macmillan Publishing Company, 1992. ISBN: 0-02-421665-8.

(Poper, 1988) POPPER, M. - KELEMEN, J.: *Expertné systémy*. Alfa, Bratislava 1988. 360 str., ISBN 80-05-00051-0.



## 4 Multiagentové systémy

**V této kapitole se dozvíte:**

- co se rozumí pod pojmem agent,
- jaké typy agentů rozlišujeme,
- jak probíhá koordinace, kooperace a komunikace agentů v multiagentovém systému.

**Po jejím prostudování byste měli být schopni:**

- vysvětlit pojmy agent a multiagentový systém,
- charakterizovat jednotlivé typy agentů a jejich interakce v multiagentovém systému.

**Klíčová slova kapitoly:** Distribuovaná umělá inteligence, multiagentové systémy, koordinace, kooperace a komunikace agentů.

### Průvodce studiem

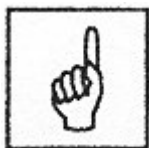
*Tato kapitola představuje úvod do problematiky multiagentových systémů. Seznámíte se zde s pojmem agent, s architekturou agenta a se základní typologií agentů. Dále bude pozornost zaměřena na systémy, ve kterých pracuje současně více agentů a které se proto označují multiagentové systémy. Zatímco problematiku jednotlivých agentů bylo možné zjednodušeně vnímat jako problém strojového učení a řešení úloh, problematika multiagentových systémů přinesla do umělé inteligence sociální témata týkající se vytváření skupin na základě společných zájmů. Aby agenti byli schopni spolupráce, musí mít prostředky pro vzájemnou komunikaci. To znamená, že musí mít společný jazyk, společné vnímání pojmů a musí mít dána pravidla vzájemné komunikace. Na zvládnutí této kapitoly budete potřebovat asi 3 hodiny.*



Problematika agentů a multiagentových systémů vychází z oblasti zvané *distribuovaná umělá inteligence*, ze které se postupně vyděluje jako samostatná disciplína, opírající se o výsledky výzkumu v oblasti počítačových věd i z

ostatních částí umělé inteligence. Distribuované řešení úloh reprezentuje tu třídu problémů, která se zabývá hledáním možností, jak pro pevně zvolenou úlohu vhodně rozdělit práci na jejím řešení mezi více modulů (uzlů) a jak následně mohou jednotlivé moduly sdílet informace o postupu vzniku cílového řešení. Studiem chování skupin volně propojených autonomních systémů (agentů), které spolupracují v zájmu nějakého společného cíle se zabývají *multiagentové systémy*.

## 4.1 Agent



Agent je entita zkonstruovaná za účelem kontinuálně a do jisté míry autonomně plnit své cíle v adekvátním prostředí na základě vnímání prostřednictvím senzorů a prováděním akcí prostřednictvím aktuátorů. Agent zároveň ovlivňuje podmínky prostředí tak, aby se přibližoval k plnění cílů (Kubík 2000).

O agentovi obecně platí, že:

- může posílat a přijímat informace od jiných agentů za použití vhodných protokolů.
- může zpracovávat přijaté informace a uvažovat o nich (tj. provádět odvozování, syntézu i analýzu).
- má soubor schopností provádět akce, které se mohou i dynamicky měnit (tj. akce charakterizují úkoly, které dokáže agent provádět).

*Inteligentní agent* je entita, která je zodpovědná za rozhodování, zda a jak reagovat na externí podněty.

Inteligentní agent má schopnost plnit cíle s využitím logické dedukce, tj. navíc umí:

- uvažovat o svých schopnostech a schopnostech ostatních agentů.
- generovat cíle nebo plány pro sebe a jiné agenty.
- účastnit se složitých interakcí s ostatními (např. za účelem vyjednávání a delegování úkolů), dynamicky se zapojovat do skupin či organizací, které mohou například sdružovat agenty s podobnou funkcí, nebo také skupiny opouštět.

- získávat informace a používat jejich zdroje a udržovat explicitní modely důvěry pro sebe a ostatní agenty.

Agenty lze rozdělit následovně (Kubík 2000):

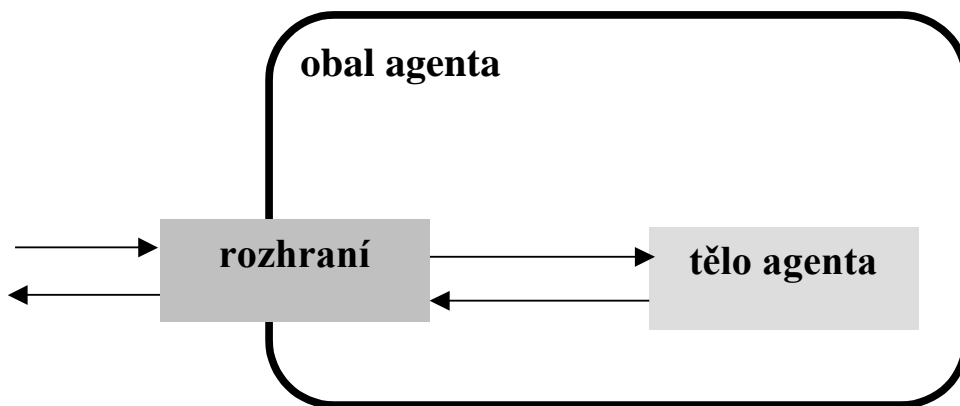
1. Biologičtí agenti – lidé.
2. Techničtí agenti – roboti.
3. Programoví agenti – softboti
  - agenti v počítačových hrách,
  - počítačové viry,
  - agenti pro specifické úkoly,
  - agenti – entity umělého života.

Funkcionalita agenta má svá ohraničení a každý agent je schopen jenom takového chování, které lze dosáhnout přizpůsobením jeho senzorů (např. kamery u robotů nebo příkazy pro zjištění obsahu adresáře u softwarových agentů) a aktuátorů (např. kolečka nebo ramena u robotů nebo příkaz na smazání souborů u softwarových agentů) prostředí.

## 4.2 Architektura agenta

Agent má obvykle následující strukturu (viz obr. 4.1):

- *Obal*, který je zodpovědný za plánování a realizaci sociálních interakcí a tvoří jej:
  - komunikační vrstva;
  - model sociálního chování.
- *Vlastní tělo*, jež nemá informace o komunitě.

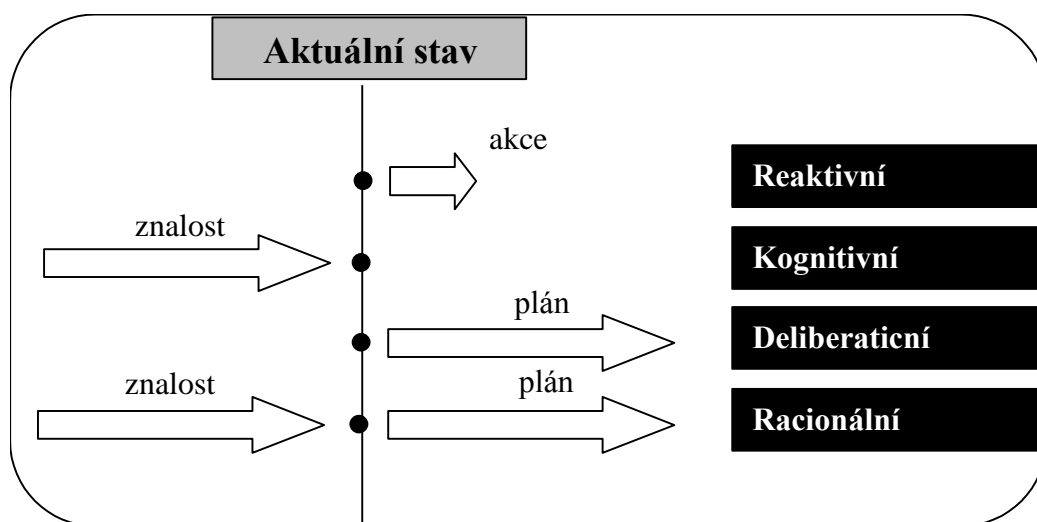


Obrázek 4.1: Architektura agenta (Netrvalová 2004).

Individuální agent může být charakterizován podle toho, zda a na jaké úrovni je schopen zvažovat rozličné varianty řešení svého cíle.

Základní charakteristiky agenta jsou (Kubík 2000):

- **autonomie**  
tj. vlastnost agenta, spočívající v samostatnosti rozhodování o svém chování v rámci daného systému, bez implicitní závislosti na jakýchkoliv jiných prvcích tohoto systému. Autonomie je pravděpodobně jediným styčným bodem mezi mnoha přístupy k pojmu agent.
- **reaktivita**  
tj. vlastnost agenta, spočívající v jeho reakci na změny prostředí tak, aby dosáhl cíle, pro který byl navržen.
- **intencionalita.**  
tj. agenti jsou schopni mít uloženy v paměti dlouhodobé cíle, organizace chování k dosahování těchto cílů, formulace vlastních plánů a využití svých úsudků.
- **schopnost sociálního chování.**  
tj. agenti jsou schopni spolupráce pro dosažení společných cílů, udržování informace o jiných agentech a vytváření úsudků o nich, sdružování do koalic a týmů, od nichž očekávají vzájemný prospěch.



Obrázek 4.2: Základní rozdělení agentů (Netrvalová 2004).

Agenty lze dále rozlišit na následující typy (viz obr. 4.2):

- **agent reaktivní**

Tento agent bezprostředně reaguje na jisté změny prostředí (nebo své změny vůči prostředí), aniž by měl vnitřní reprezentaci znalostí o tomto prostředí. Jeho reakce nejsou výsledkem výpočtů či dedukcí na základě znalostí, ale pouze reakcemi na podněty. Reaktivita je vedle autonomie druhou významnou vlastností agentů.

- **agent deliberativní (rozvážný)**

Na rozdíl od reaktivního agenta má deliberativní (rozvážný) agent schopnost plánovat postup svých akcí vedoucích k dosažení zvolených nebo zadaných záměrů/cílů. To znamená, že agent musí mít schopnost různých výpočtů, což bude později v textu chápáno jako druh vnitřní činnosti agenta. K dosažení svých záměrů pak agent ovlivňuje okolní prostředí tak, aby získal nějakou výhodu. Toto jednání je další z často uváděných vlastností agentů a nazývá se proaktivita.

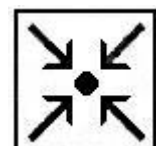
- **agent kognitivní**

Kognitivní agent má schopnost vyvozovat logické závěry ze svých pozorování okolního prostředí. Takový agent musí především být schopen se učit a vytvářet si svou vlastní bázi znalostí. Do ní si během svého působení ukládá informace získané interakcí s okolím nebo znalosti získané dedukcí. Kognitivní agent nemusí mít nutně deliberativní schopnosti. Pak provádí pouze vnitřní akce, například analyzuje scénu, provádí překlad, nebo získává znalosti (dolování znalostí z dat).

- **agent racionální**

Racionální agent má všechny výše uvedené vlastnosti a jeho struktura obsahuje jak plánovací jednotku, tak i kognitivní jednotku včetně báze znalostí. Je to agent, který je na základě svých poznatků schopen se učit a pak plánovat svoji činnost tak, aby dosáhl svých cílů racionálním způsobem. Stojí nejvýše na pomyslné hierarchii uvedených agentů (viz obr. 4.2).

Příkladem *reaktivního* agenta je mechanická beruška, která má senzory – tykadla a aktuátory – kolečka. Prostředím berušky je rovná plocha stolu.



Jediným relevantním stimulem je zachycení okraje desky stolu, kdy se poloha jednoho z tykadel sníží, čímž se malé kolečko dostane na plochu stolu a způsobí otočení berušky doprava. Beruška se tak vyhne pádu přes okraj stolu. Toto chování je v jejím jednoduchém prostředí racionální, i když o tom sama beruška nemá „představu“ a nevyužívá žádné reprezentace prostředí k jeho dosažení.

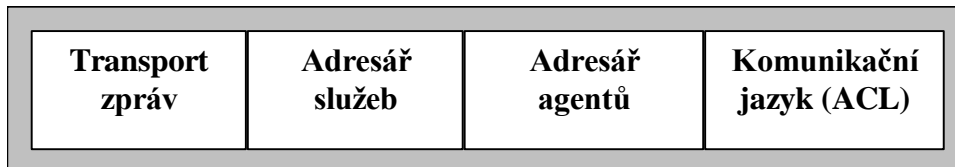


Problematikou budování inteligentních systémů zdola nahoru, tj. od jednodušších agentů ke stále komplikovanějším se jako první zabývala M. Matric (Mataric 1994). Nejprve sestrojila jednoduchého robotnického agenta, vybaveného sonary rozloženými po obvodu jeho válcovitého těla pro vnímání prostředí a s přiměřenou motorikou, který se dokázal pohybovat podél stěn a vyhýbal se při svém pohybu překážkám. Takto vzniklý systém doplnila tak, aby byl schopen zapamatovat si určité charakteristiky stavů prostředí, ve kterých se právě nacházel a následně rozeznal body, které již dříve navštívil. Postupně by měl agent z takovýchto bodů sestavovat mapu svého prostředí, kterou by si měl pamatovat, tj. postupně si vytvářet vnitřní reprezentaci svého prostředí, přičemž využíval tuto mapu k efektivnější navigaci. Robot, který byl vybudován na základě této architektury byl nazván Toto. Robot Toto byl svým chováním o krok blíže k autonomnímu chování lidí. Dokázal cíleně sledovat vytyčenou cestu na základě zaznamenání historie rozložení objektů v prostoru a také její dynamické modifikace. Tuto reprezentaci si vytvářel sám, avšak její jednoduchost na úrovni symbolických značek záchytných bodů v místnostech mu neumožňovala projevovat se jinak, než bezkolizním pohybem v komplexu místnosti. Dále nebylo ani vyřešeno, jak ale zabezpečíme, aby se agent dokázal rozhodnout, kterou část úkolu bude právě plnit, jaké prostředky na splnění cílů použije a jak bude postupovat při plnění cílů. Nelze také *deliberativního* agenta naučit zlepšovat své chování a funkcionalitu.



Foundation for Intelligent Physical Agents (FIPA) je organizace, která se zabývá sjednocením výzkumu v oblasti softwarových agentů a multiagentních systémů a standardizací jednotlivých technik. *FIPA architektura agenta* předpokládá realizaci agenta jako internetovou službu. Abstraktní FIPA architektura však není šablonou pro vytváření konkrétních agentů. Zahrnuje

sice principy společné pro většinu vytvořených systémů agentů, ale nezmiňuje například vůbec problematiku plánování a racionálního jednání. Na obrázku 4.3 jsou ukázány základní bloky specifikace správy agenta FIPA. V rámci jedné platformy existuje jedna specifikace správy agenta a každý agent musí v ní být registrován. Agenti mohou využívat vlastnosti systému pro registraci svých služeb v adresáři služeb a svých lokací v adresáři agentů a komunikovat s jinými agenty prostřednictvím jazyka ACL a transportu zpráv.



Obrázek 4.3: Specifikace správy agenta FIPA (Netrvalová 2004).

### 4.3 Multiagentové systémy

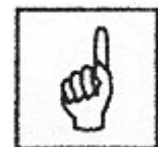
Multiagentové systémy (MAS Multi-Agent Systems,) jsou takové systémy, kde se v prostředí pohybuje více než jeden agent. Musí tudíž nutně docházet ke spolupráci agentů a je proto zapotřebí vytvořit prostředky pro vzájemnou komunikaci.

V multiagentových systémech musí mít agenti společné následující schopnosti a zájmy:

- vnímání pojmů,
- pravidla vzájemné komunikace,
- jazyk.

Agenti mohou navzájem ovlivňovat svoje chování, tj. svou činnost mohou provádět v souladu s ostatními nebo pouze nenarušovat činnost ostatních nebo působit proti ostatním. Dále můžeme agenty dělit na (Netrvalová 2004):

- **kooperativní agenty**  
mající společné cíle, tj. každá akce, která je vykonána jedním agentem je v souladu se záměry druhého agenta.
- **kompetitivní agenty**  
mající protichůdné cíle, tj. snaha jednoho agenta k dosažení jeho cíle jde přímo proti snaze agenta druhého.



- **kolaborativní agenty**

tj. navzájem spolupracující agenty.

Častější jsou však případy, kdy agenti mají své partikulární zájmy, které sice nejsou stejné, ale nejsou ani v přímém rozporu se zájmy ostatních agentů. Během činnosti agentů pak může docházet k souladu nebo kolizím jejich záměrů.



*Koordinace* je proces probíhající v multiagentových systémech, kterým se dosahuje propojení jednotlivých komponent v systému za účelem řešení problému. Koordinační procesy lze rozdělit do dvou základních kategorií na procesy s centralizovaným a decentralizovaným řízením. Pokud je společenství agentů řízeno jedním centrálním prvkem nebo malou skupinou částečně decentralizovaných komponent, vzniká v něm *kooperace*. Každý agent má přesně určenou roli, kterou musí plnit a také vztahy s ostatními agenty k tomu, aby bylo dosaženo globálního cíle. Říkáme, že agenti mají protokol pro kooperaci. Protokolu se využívá za účelem efektivní *komunikace*, ke které v rámci kooperace dochází. Jedná se o přesně stanovené sledy zpráv, které určují, jak lze odpovědět na přijatou zprávu. Agenty mohou komunikovat i bez takto specifikovaného protokolu, což je sice mnohem sofistikovanější, ale také mnohem složitější způsob komunikace.

Kooperace je vyšší proces než koordinace, protože kooperaci nelze bez procesu koordinace dosáhnout, ale zároveň koordinace mezi agenty nemusí nutně vést ke kooperaci. Komunikace je nutným předpokladem kooperace, jejímž speciálním případem je kolaborace s pevnějšími vazbami a závazky mezi agenty.



*Strategie agenta* udává, která akce z báze akcí bude vykonána jako reakce na aktuální stav prostředí. U reaktivního agenta vyplývala následná akce automaticky z ohodnocení prostředí, kdežto v případě multiagentového systému je strategie agenta volena z množiny strategií podle jeho vlastního rozhodnutí. Strategie skupiny agentů je množina strategií jednotlivých agentů. Nejjednodušší situace nastává, když si každý agent volí svou strategii bez ohledu na strategie, které si zvolili ostatní agenti. Pak je zřejmě jeho záměrem



zvolit takovou strategii, která je pro něho optimální. Pokud existuje strategie, která je pro agenta nejlepší nezávisle na strategiích zvolených ostatními agenty, je tato strategie nazývána dominantní strategií. Racionální agent pak vždy volí tuto dominantní strategii. Dominantní strategii lze nalézt například v problému známém pod názvem věžňovo dilema (Prisoner's dilemma).

**Příklad** (Netrvalová 2004).

Dva agenti A (např. Alice) a B (např. Bob) byli chyceni nedaleko místa loupeže a oběma hrozí za tento čin deset let vězení. Oba jsou vyslýcháni zvlášť a je jim řečeno, že když se přiznají, bude jim přiznána polehčující okolnost a dostanou jen pět let. Když se nepřiznají, dostanou stejně rok vězení za to, že u nich byl nalezen ukradený prsten. Pokud se však přizná jen jeden z nich a bude svědčit proti druhému, bude za to propuštěn. Následující tabulka udává možnosti pro oba agenty.



Tabulka 4.1: Věžňovo dilema.

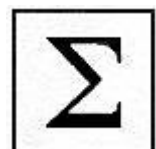
	Alice se přizná	Alice se nepřizná
Bob se přizná	A = -5; B = -5	A = -10; B = 0
Bob se nepřizná	A = 0; B = -10	A = -1; B = -1

Je zřejmé, že výsledek strategie jednoho agenta nezávisí na zvolené strategii druhého agenta (pro Boba je výhodnější se přiznat, nezávisle na tom, co udělá Alice a totéž platí pro Alici) a proto v tomto případě existuje dominantní strategie pro oba obviněné.

## Shrnutí kapitoly

V této kapitole jste se seznámili s pojmem agent, s architekturou agenta a se základní typologií agentů. Dále jsme se věnovali multiagentovým systémům. Aby agenti byli schopni kooperace, musí svá jednání koordinovat, a proto spolu musí komunikovat a mít i vůli ve prospěch celé skupiny.

Kooperace je vyšší proces než koordinace, protože kooperaci nelze bez procesu koordinace dosáhnout a zároveň nutným předpokladem kooperace je komunikace.



**Kontrolní otázky a úkoly:**

1. Co se rozumí pod pojmem agent?
2. Jaké typy agentů rozlišujeme?
3. Jak probíhá koordinace, kooperace a komunikace agentů v multiagentovém systému?

**Korespondenční úkoly**

1. Nalezněte na Internetu další možné charakteristiky a typy agentů. Na toto téma pojednání v rozsahu alespoň 5 stran
2. Nalezněte např. na Internetu možné příklady použití multiagentových systémů při řešení úloh. Téma zpracujte jako seminární práci v rozsahu alespoň 5 stran.

**Citovaná a doporučená literatura**

Kubík, A., *Agenty a multiagentové systémy*, Slezská univerzita v Opavě, 2000.

Mataric, M., *Interaction and Intelligent Behavior, MIT PhD thesis*, MIT Press, Cambridge, Mass., 1994

Netrvalová, A. Úvod do problematiky multiagentních systémů. 2004. [Cit. 2013-07-16]. Dostupné online <http://www.kiv.zcu.cz/~netrvalo/phd/MAS.pdf>.

## 5 Umělý život

### V této kapitole se dozvíte:

- Co je umělý život
- Jaké jsou projevy umělého života
- Jak lze umělý život modelovat

### Po jejím prostudování byste měli být schopni:

- Charakterizovat definici umělého života
- Chápat pojem celulární automat
- Modelovat umělý život

**Klíčová slova kapitoly:** Celulární automat, life.

#### Průvodce studiem

*Stejně tak jako definice umělé inteligence naráží na problém neexistence všeobecně přijatelné definice inteligence přirozené, naráží i definice umělého života na nejednotnost definice života existujícího na Zemi. Navíc otázky života vyvolávají v porovnání s otázkami inteligence ještě více pochybností etických, filozofických i náboženských. V této kapitole se pokusíme ozřejmit, co považujeme za umělý život a jaké jsou jeho projevy. Na zvládnutí této kapitoly budete potřebovat asi 1,5 hodiny, tak se pohodlně usad'te a nenechte se nikým a ničím rušit.*



### 5.1 Umělý život

Umělý život je všeobecná metoda, podstatou které je generovat z jednoduchých mikroskopických spolupracujících prvků takové chování na úrovni makroskopické, které je možno interpretovat jako projev života.

Základní vlastnosti umělého života jsou následující:



- Podstatou umělého života je *informace* a ne materiální forma, která slouží k jejímu uchování a zpracování. Umělý život v podstatě představuje systém pro zpracování informací.
- Život vyžaduje jistou míru *složitosti*. Struktura po dosažení určité složitosti je schopna vytvářet svoje identické kopie a také složitější potomky.
- Informace má dvojitou podobu:  
neinterpretovaná - *genotyp* - slouží zejména k rozmnožování;  
interpretovaná - *fenotyp* - podle něj se vytváří struktura nového jedince.
- Prostředkem *evoluce* tj. vývoje směrem ke složitějším a dokonalejším strukturám je samoreprodukce, mutace a selekce.
- Syntetický proces vývoje života probíhá zásadně směrem zdola nahoru, tj. od elementárních primitiv k složitým strukturám vykazujícím složité chování.
- Vzájemné lokální spolupůsobení primitiv vyvolává na globální úrovni úplně nové fenomény – tento jev je označován jako *emergence*. Vše přitom probíhá bez jakéhokoliv centrálního řízení.
- Jednou z podmínek *emergence* je nelineární chování elementárních primitiv, tedy neplatnost principu superpozice. *Princip superpozice* předpokládá že chování celku lze odvodit složením dílčích chování primitiv. (Codd, 1968), (Gardner, 1971).

## 5.2 Celulární automaty



Celulární automat (zkratka CA) je souhrnné označení pro určitý typ fyzikálního modelu reálné situace, ať již v podobě reálného přístroje či mnohem častěji počítačového algoritmu (programu). Slouží k časové i prostorové diskrétní (nespojité) idealizaci (ideální modelaci) fyzikálních systémů, kde hodnoty veličin nabývají pouze diskrétních hodnot v průběhu času. Celulární automat (dále jen CA) je dynamický systém, který je diskrétní v prostoru i čase. Je tvořený pravidelnou strukturou buněk v  $n$ -rozměrném prostoru. Nejčastěji je  $n=2$ , tj. buňky tvoří čtvercovou mřížku. Každá buňka může nabývat jeden z  $K$  možných *stavů*. Často se jedná pouze o dva stavy:  $0$  - mrtvá buňka,  $1$  - živá buňka; v tomto případě se občas stav  $1$  označuje jako buňka a  $0$  jako prázdné políčko (mřížky). Hodnoty stavů buněk v dalším časovém kroku (v následujících

generaci) se vypočítávají *paralelně* na základě *lokální přechodové funkce* (stejně pro všechny buňky). Argumenty této funkce jsou aktuální hodnoty stavů uvažované buňky a všech sousedních buněk v jejím okolí. V případě, že  $n=1$ , je *okolí* tvořené tzv. poloměrem - počtem sousedů po obou stranách uvažované buňky; v případě, že  $n=2$ , tvoří okolí čtyři přilehlé buňky k dané buňce (tzv. *neumannovské okolí*) nebo se do něj zahrnou i čtyři další sousední buňky, které se dotýkají vyšetřované buňky jen v rozích (tzv. *úplné okolí*). Využívá se v teorii systémů, matematice a teoretické biologii. (Codd, 1968), (Gardner, 1971) .

### 5.3 Historie CA

První nápady na CA se objevily již ve 40. letech dvacátého století, kdy se John von Neumann snažil navrhnout stroj, který by kontroloval i opravoval sám sebe. Jeho cílem bylo najít logickou strukturu sebereprodukcujícího se automatu bez nutného vztahu k biologickým procesům. Spolu s S. Ulamem rozdělili celý prostor na jednotlivé buňky (cells), přičemž každá buňka je na začátku charakterizována počátečním stavem. Tento počáteční stav se podle evolučního pravidla mění vždy po jednotlivých krocích, tzv. iteracích. Evoluční pravidlo nemusí být stejné pro všechny buňky, ale je vždy funkcí stavů buněk v okolí buňky, kterou právě zkoumáme. V padesátých letech se začaly CA používat k automatickému zpracování obrazů. Byla vyvinuta speciální pravidla pro úpravu šumu, odhad velikosti a počtu objektů na obraze pořízeném mikroskopem.

V roce 1970 našel John Conway jednoduché pravidlo vedoucí ke komplexnímu chování. Nazval je Game of Life (více níže). V sedmdesátých letech byl vytvořen HPP model mřížového plynu trojicí Hardy, Pomeau a de Pazzis. Je to jeden z prvních CA, který zachovává kromě počtu částic také hybnost. Tímto modelem se otevřela cesta k simulování pohybu tekutin, nebo zrnitých látek pomocí CA. V osmdesátých letech se Stephen Wolfram začal zabývat jednodimenzionálními CA, u kterých prokázal jejich vztah ke statistické fyzice.

V roce 1986 přišel na svět FHP model pánů Frische, Hasslachera a Pomeaua[5], který nezávisle na nich objevil také Stephen Wolfram. Provedou-

li se u něj příslušné limity, ukazuje se, že se jedná o obdobu Navierových-Stokesových rovnic, což je nepopiratelný důkaz schopnosti CA modelovat reálný fyzikální problém.

Celulární automat je dynamický systém, diskrétní v hodnotách, prostoru i čase. Je tvořen pravidelnou strukturou buněk v  $N$ -rozměrném prostoru (nejčastěji je  $N=2$ , tzv. 2D CA, kde buňky tvoří čtvercovou mřížku). Každá buňka může nabývat jeden z  $K$  možných stavů. Často jde pouze o dva stavy: 0-mrtvá buňka, 1-živá buňka; v tomto případě se občas stav 1 označuje jako buňka a 0 jako prázdné políčko (mřížky).

Hodnoty stavů buněk v dalším časovém kroku (v následující generaci) se vypočtou synchronně na základě lokální přechodové funkce (stejně pro všechny buňky). Argumenty této funkce jsou aktuální hodnoty stavů vyšetřované buňky a všech sousedů (buněk v jejím okolí).

V případě 1D CA je okolí charakterizováno tzv. poloměrem, tj. počtem sousedů po obou stranách vyšetřované buňky; v případě 2D CA tvoří okolí čtyři přilehlé buňky (tzv. neumanovské okolí), a nebo se do okolí zařadí i čtyři další sousedi, dotýkajících se vyšetřované buňky jen v rozích (tzv. úplné okolí). Zpravidla se očekává, že struktura buněk je nekonečná. V praktických realizacích se buď předpokládají okrajové buňky identicky nulové (prázdné), nebo jsou okraje „propojeny“ a tvoří v případě 1D smyčku a v případě 2D anuloid. Některé z  $K$  možných stavů jsou označovány za „klidové“; když buňka v klidovém stavu má ve svém okolí také jenom buňky v klidovém stavu, potom se hodnota jejího stavu v další generaci nemění. (web 1,2)

Někdy je účelná širší koncepce, ve které jsou pro CA charakteristické tři klíčové vlastnosti:

- paralelismus (výpočet nových hodnot všech stavů probíhá současně, na běžných sériových počítačích se musí tento postup simulovat)
- lokalita (nový stav prvku závisí jen na jeho původním stavu a na původních stavech prvků z jeho okolí)
- homogenita (pro všechny prvky platí stejná lokální přechodová funkce)



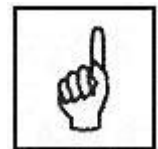
Nevyžaduje se při tom nutně pravidelná prostorová struktura (proto byl v uvedené charakteristice použit všeobecnější pojem „prvek“ místo „buňka“).

## 5.4 Hra života

Hra života nebo také život (anglicky Game of Life či pouze Life) je dvoustavový, dvourozměrný celulární automat, který svým chováním připomíná vývoj společenství živých organismů. Odehrává se na matici buněk, jejíž stav předurčuje podobu hry v následujícím kroku. Uživatel pouze určí výchozí konfiguraci a dále již hra běží automaticky podle předem daných pravidel. Čas je ve hře diskrétní, při každém uplatnění pravidel se posune o jednotku. Hra života se od podobných celulárních automatů liší v pravidlech pro zrod nových buněk a jejich přežití. Kořeny hry života sahají do roku 1940, kdy maďarský matematik John von Neumann přišel s konceptem celulárních automatů. Hru života jako takovou však vymyslel až britský matematik John Horton Conway v roce 1970. V průběhu hry vznikají různé tvary, které se dělí do kategorií jako jsou zátiší, oscilátory, děla a další. Hra života je příkladem systému, kde z jednoduchých pravidel vzniká komplexní chování - objevují se zde tzv. emergentní struktury. (web 1,2)

### 5.4.1 Popis a pravidla hry

Dvourozměrná hrací plocha Hry života se znázorňuje jako čtvercová síť. Její políčka se nazývají buňky (proto název celulární neboli buněčný automat). Každá buňka může nabývat dvou stavů, může být živá či mrtvá. Buňka sousedí s dalšími buňkami, se kterými se dotýká hranou (ortogonálně) či vrcholem (diagonálně) a tvoří tak Moorovo okolí. Nový stav buňky určuje přechodová funkce podle stavu této buňky a jejího okolí v předchozím kroku. Stav všech buněk se mění naráz. Celulární automaty, které po vzoru Hry života splňují výše uvedené skutečnosti tvoří třídu celulárních automatů založených na Hře života (Life-like cellular automata). Sestavení živých a neživých buněk na hrací ploše se říká konfigurace nebo vzor. Jednotkou času je ve Hře života generace. Jedna generace představuje jedno provedení přechodové funkce. Podle toho, jaká pravidla pro změnu stavu buněk přechodová funkce uplatňuje, se hovoří o konkrétním názvu life-like celulárního automatu. (web 1,2)



Jako Hra života se označuje celulární automat jehož přechodová funkce používá množinu pravidel, kterou na počátku 70. let definoval britský matematik John Conway takto:



- Každá živá buňka s méně než dvěma živými sousedy zemře.
- Každá živá buňka se dvěma nebo třemi živými sousedy zůstává žít.
- Každá živá buňka s více než třemi živými sousedy zemře.
- Každá mrtvá buňka s právě třemi živými sousedy oživne.

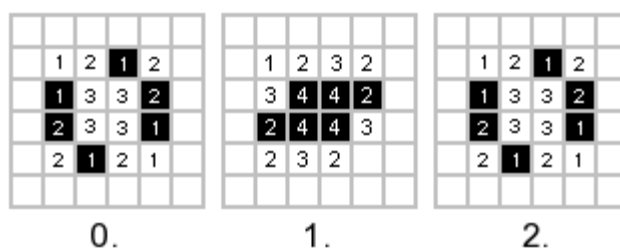
Tato konkrétní pravidla jsou označována jako S23/B3. Číslo před lomítkem říká kolik sousedů musí buňka mít, aby přežila, čísla za lomítkem říkají kolik sousedů potřebuje mrtvá buňka k oživení. Tato pravidla byla vybrána vědomě s cílem zajistit, aby populace živých buněk rostla nepředvídatelně.



Konkrétně jde o splnění požadavků, že by:

- neměl existovat žádný počáteční vzor, pro který by šlo snadno dokázat, že populace může růst nade všechny meze,
- měly existovat počáteční vzory zdánlivě nade všechny meze rostoucí
- a že by měl existovat takový počáteční vzor, který se určitou dobu vyvíjí než úplně vymizí, přejde do stabilní podoby anebo do oscilační fáze.

Možných pravidel pro Life-like buněčné automaty je celkem  $2^{18}$ . Pro definici počtu buněk nutných k přežití či oživení lze totiž použít pro každou z nich celkem 9 možností (0-8 sousedů) a buňka se nachází ve dvou stavech. Při některých pravidlech všechny vzory vymizí, v některých dochází k explozivnímu růstu. (web 1,2)

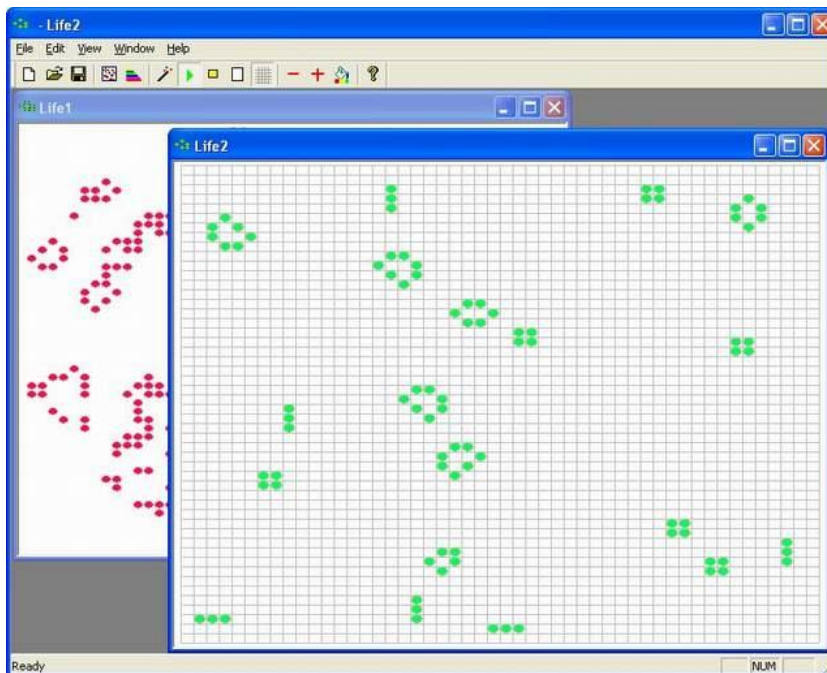


Obrázek 5.1: Vývoj CA v průběhu tří generací

Obrázek znázorňuje vývoj vzoru v průběhu tří generací. Černě vybarvená buňka je ve stavu živém, bílá ve stavu neživém. Číslo uvnitř buňky označuje počet živých buněk v jejich okolí. Přechodová funkce se řídí pravidly



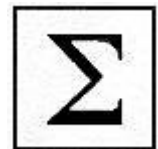
S23/B3.V nulté generaci se vlevo dole nachází živá buňka s právě jedním sousedem. Tato buňka po aplikaci pravidel zemře a v příští generaci se stane neživou. Vlevo a vpravo se nachází živé buňky se dvěma sousedy. Tyto buňky přežijí do příští generace. Čtyři neživé buňky uprostřed, které mají každá tři živé sousedy, v příští generaci ožijí. V první generaci se uprostřed nachází čtyři buňky každá se čtyřmi sousedy. Tyto buňky podle pravidel zemřou a v příštím kole budou neživé. Pravidla se uplatní i na ostatní buňky a výsledkem je generace 2. V tomto případě je druhá generace stejná jako nultá. Tento tvar spadá do skupiny oscilátorů s periodou dvě generace. (web 1,2)



Obrázek 5.2: Ukázka reálné aplikace s hrou života

## Shrnutí kapitoly

- V této kapitole jde hlavně o pochopení základních principů umělého života ve formě celulárních automatů. Hlavním a nejpodstatnějším představitelem je hra LIFE, která je v této kapitole vysvětlena. V rámci video tutoriálů, které vznikají, jako podpůrný materiál k výuce bude hra LIFE reálně na aplikaci představena a demonstrována.

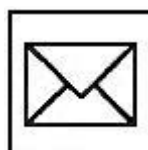


**Kontrolní otázky a úkoly:**

1. Co je celulární automat
2. Jaké má celulární automat základní charakteristické vlastnosti
3. Na jakém principu pracuje hra LIFE

**Úkoly k textu**

1. Nalezněte na Internetu další simulátory umělého života a vyzkoušejte si je na vlastní aplikaci.

**Korespondenční úkoly**

1. Nalezněte např. na Internetu další možné aplikace umělého života. Téma zpracujte jako seminární práci v rozsahu nejméně tří stran.

**Citovaná a doporučená literatura**

(Codd, 1968) Codd E.F. *Cellular Automata*. Academic Press, New York 1968.

(Gardner, 1971) Gardner M. „On cellular automata, self-reproduction, the Garden of Eden and the game life“. *Scientific American* 224 (1971), pp. 112-117.

(web2): [http://cs.wikipedia.org/wiki/Celul%C3%A1rn%C3%AD\\_automat/](http://cs.wikipedia.org/wiki/Celul%C3%A1rn%C3%AD_automat/),  
(listopad, 2012).

(web1): [http://cs.wikipedia.org/wiki/Hra\\_%C5%BEivota](http://cs.wikipedia.org/wiki/Hra_%C5%BEivota), (říjen 2012)

## 6 Deterministický chaos – principy a aplikace

**V této kapitole se dozvíte:**

- Co je to deterministický chaos
- Jak souvisí chaos a determinismus
- Co je to turbulentní jev a samoorganizace
- Jak se dá chaos řídit...

**Po jejím prostudování byste měli být schopni:**

- Vysvětlit pojem chaosu
- Charakterizovat metody řízení
- Rozumět pojmům jako samoorganizace

**Klíčová slova kapitoly:** Atraktor, samoorganizace, deterministický chaos..

### **Průvodce studiem**

*Tato kapitola je zaměřena na problematiku řízení deterministického chaosu. Vzhledem k šíři dané problematiky je kapitola zaměřen spíše do úrovně informativní s cílem podat studentovi alespoň nástin základních principů chaosu a jeho řízení. Současně s „klasickou“ teorií řízení chaosu jsou zde nastíněny současné trendy v oblasti řízení chaotických systémů a rozloženými parametry. Na zvládnutí této kapitoly budete potřebovat asi 2 hodin/y, tak se pohodlně usadte a nenechte se nikým a ničím rušit.*



Když se vysloví slovo „chaos“, obvykle se u lidí, kteří se touto disciplínou nezabývají, vybaví proces který je čistě nahodilý a bez vnitřních zákonitostí. Málokdo si uvědomuje že „být chaotický“, znamená striktně se řídit přesně danými pravidly, v nichž mnohdy není pro náhodu místo. Chaos je disciplínou, která dostala své jméno až ve dvacátém století, nicméně jeho kořeny sahají až do 19 století, kdy bylo zjištěno, že už jednoduché problémy generují velmi složité a nepředpověditelné chování.

V poslední době vznikl v souvislosti s deterministickým chaosem směr zvaný řízení deterministického chaosu. Je to relativně nový vědecký interdisciplinární směr, v němž dochází k symbióze více oborů jako je např. fyzika, chemie, biologie či elektronika. To vše pak zastřešuje kybernetika, která využívá poznatků přispívajících oborů k syntéze toho, čemu se říká "řízení deterministického chaosu". Navzdory faktu, že v samotném názvu se vyskytují dvě vzájemně odporující si pojmy (řízení a chaos), je pravdou, že se jedná o rigorózní vědu řídicí se striktními pravidly. Cílem této kapitoly je podat nejen nástin toho co znamená řízení chaosu, ale také toho co to vlastně deterministický chaos znamená, jak se rodí a samozřejmě také kdo se podílel na jeho vzniku či spíše objevení.

## 6.1 Popis deterministického chaosu

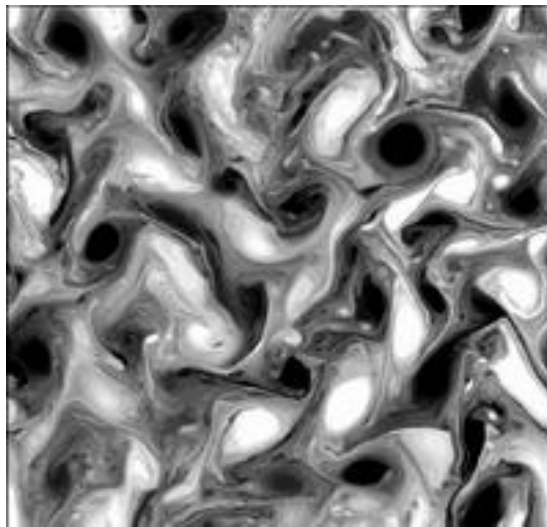
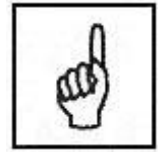


Pod pojmem chaos se všeobecně rozumí takové chování, které je projevem absolutní a čisté náhody, není v něm tedy místo pro působení zákonitostí. Takovéto chování by bylo možno nazvat absolutně nekoherentní, kdy neexistují zákonité vazby mezi sousedními stavy (jak v prostoru tak v čase). Chaotické chování skutečných systémů v přírodě však charakterizuje termín "deterministický chaos". Jedná se o proces samoorganizace složitých systémů, kdy vznikají soustavy koherentních struktur chovajících se v souladu s přírodními zákony. Z hlediska jedné určité struktury má vývoj systému prvky náhodnosti, systém jako celek se však vyvíjí zcela zákonitě a tedy deterministicky. Tento jev lze nalézt v přírodních systémech všech možných forem - od fyzikálních, chemických (např. chemické reakce), ekonomických (např. Elliottova vlna na kapitálových trzích), tak i v biologických systémech (např. chování kolonie mravenců).

O některých projevech chaosu se ví již poměrně dlouhou dobu, většinou však byly v minulosti považovány za neužitečnou anomálii a proto také tabu pro výzkum. Dnes na chaos pohlížíme jako na způsob chování vlastní všem přírodním dynamickým systémům, které se nalézají dostatečně daleko od jejich statické rovnovážné polohy. Za těchto podmínek totiž vzniká zcela zákonitě chaotické chování v souvislosti s uplatněním nelinearit.

## 6.2 Turbulence

Typickým příkladem relativně dobře prozkoumaného systému, chovajícího se podle zákonů deterministického chaosu, je turbulentní proudění. Struktura vyvinutého turbulentního proudění je charakterizována vírovými koherentními strukturami jejichž velikost je dána jistými zákonitostmi, okamžitá poloha a orientace konkrétního víru v prostoru je však náhodná. Turbulentní proudění lze pozorovat při experimentech, lze jej však také modelovat pomocí numerické simulace rovnic popisujících pohyb tekutiny. Matematický model proudící tekutiny je dán rovnicemi bilance hybnosti, jedná se o tzv. Navier-Stokesovy rovnice spolu s rovnicí kontinuity (zachování hmoty). Navier-Stokesovy rovnice jsou parciální diferenciální, nelineární rovnice. V každém případě se jedná o deterministický matematický model. Ukazuje se však, že za určitých podmínek může dojít k extrémnímu zesilování poruch určitého charakteru v proudovém poli. Systém tedy funguje jako filtr, který některé poruchy potlačuje, jiné zesiluje. Tento proces, který je zpočátku lineární, vede po určitém čase, kdy dojde k zesílení poruch nad jistou mez, k masivnímu uplatnění nelinearity a k přechodu systému do chaotického stavu.



Obrázek 6.1: Turbulentní jev hladiny kapaliny

## 6.3 Atraktory

*Co to je vlastně atraktor? Jak se definuje? Proč se spojuje s pojmem fázový prostor? Jak se atraktory dělí? Co znamená, když se řekne podivný atraktor? Tak se dozvíte, který atraktor je nejznámější.*



Podle definice je to konečný stav systému. Atraktor můžeme celkem jednoduše zobrazit na grafech, tedy pokud nemá systém více jak tři rozměry. Zde bychom rádi zavedli pojem fázový prostor. Pro pohyb hmotného bodu si vystačíme s třemi souřadnicemi a jestliže přidáme i všechny 3 složky vektoru jeho rychlosti dostaneme 6 souřadnic - fázový prostor. Problém nastane tehdy, když chceme popsat chování (např. pohyb) systému, který se nedá definovat pomocí jednoho hmotného bodu - pohyb pružných, kapalných těles nelze jednoduše zanášet do grafu, ale matematicky se chování těchto systémů dá v principu zvládnout. Jak se traktory dělí? Začneme bodem. Vysvětlení najdete v následujícím příkladu.



Fyzikální kyvadlo (podléhající tření) se postupně zastavuje, a když se zcela zastaví, dosáhne svého atraktoru. Zde se situace zjednodušuje tím, že kyvadlo je de facto hmotný bod. Jak můžeme zobrazit jeho chování? Například máme dvojrozměrný graf, osa x představuje čas, a na osu y nanášíme aktuální výchylku. Zpočátku dostaneme graf podobný funkci  $\cos$ , ale s postupujícím časem se křivka vyhlazuje, až se z ní stane přímka - kyvadlo se zastavilo. Můžeme ještě přidat třetí prostor - aktuální rychlost kyvadla - a jsme u fázového prostoru. Nebo to můžeme udělat jinak - ignorujeme čas a na graf vykresluje jen aktuální výchylku a aktuální rychlost. Pro nebržděné kyvadlo se nám vykreslí kružnice a pro bržděné se vykreslí spirála - atraktorem je bod uprostřed. Nemusíme dokonce ani kreslit graf - stačí kyvadlo v podobě dřevě nádoby s pískem. Ten se stále vysypává a kde je hromádka největší, tam je atraktor (za předpokladu, že se kyvadlo zastaví dříve, než dojde písek). Výhoda tohoto způsobu je v tom, že se kyvadlo může kývat ve dvou rozměrech - tedy nejen tam a zpátky, ale zase nevidíme názorně aktuální rychlost kyvadla. Kyvadlo lze popsat poměrně jednoduchými vzorci a tak nás ani nepřekvapí jednoduchost atraktoru. Některé systémy se nespokojí s bodem, ale s cyklicky se opakující křivkou.



Lepším příkladem jsou planety. Podmínka cyklu je ovšem splněna jen tehdy, když nehmotný bod obíhá kolem hmotného bodu - planety mají zanedbatelnou hmotnost. Jejich atraktor je obyčejná elipsa. Ve skutečnosti to není úplně přesné (planety si Slunce trošičku přitahují a tím porušují uzavřenou elipsu), také jsme zanedbali zakřivení prostoru.

Podobný atraktor má i mlýnské kolo. Právě na něm si vyznačíme bod a pokud stále přitéká voda, jeho atraktorem je kruh. Vraťme se ale zpátky do vesmíru. Jak je dobře známo, většina hvězd jsou vlastně dvojhvězdy. Předpovídat polohu planety ve Sluneční soustavě je triviální, ale co u dvojhvězdy? To je problém. Představte si ale jinou situaci. Tři srovnatelně hmotná tělesa. To je také velký problém. Tento je ovšem neřešitelný, neboť koncem předminulého století H. Poincaré dokázal, že analytické řešení neexistuje (důkaz stále platný)! Není tedy možné žádným alespoň trochu přesným způsobem předpovědět chování tří blízkých hvězd! Pro podobný úkaz ale nemusíme létat do vzdáleného vesmíru. V pásu asteroidů mezi Marsem a Jupiterem obíhá skupina tří velkých asteroidů, které se jmenují Trojané. Zde je situace ještě komplikovaná přítomností menších těles a prachu. Jaký je tedy atraktor této soustavy? Atraktor této soustavy je opravdu podivný. Toto je chápáno jako pojem, ne jako přirovnání. Jednotlivé body se na grafu mohou objevovat zdánlivě náhodně a vytvoří nekonečnou křivku (nikde se neprotíná) - podivný atraktor. To, že neexistuje analytické řešení, by mohlo svádět k domněnce, že tuto soustavu nelze vyjádřit rovnicemi. To sice lze, ale míra vzájemného ovlivňování je velmi vysoká (což je vyjádřeno značnou nelinearitou soustavy). Proto je systém extrémně nestabilní i při malé změně počátečních podmínek.



Atraktor dynamického systému je stav, do kterého systém směřuje v nekonečném čase. Je to tedy množina, do které se dostane stavový vektor, když je systém v nekonečném čase. V případě planety, pohybující se kolem Slunce je tímto atraktorem buď kružnice, nebo elipsa čili hladké uzavřené křivky o konečné délce, takže v nich je planeta trvale vázána. Atraktory rozdělujeme do několika tříd: (web2).

- atraktorem jsou pevné body
- atraktorem jsou periodické body
- atraktorem jsou kvaziperiodické body
- atraktor je chaotický
- podivný atraktor



Jsou-li atraktorem dynamického systému pevné body, jde o nejjednodušší případ. Systém se v nekonečném čase ustálil v nějakém stabilním stavu a v

podstatě už nejde o dynamický systém. Příkladem může být kyvadlo, které se vlivem odporu vzduchu a odporu ložisek zastaví v nejnižším bodě své dráhy. Jsou-li atraktorem periodické (resp. kvaziperiodické) body, jde také o jednoduchý případ. Systém se ustálí tak, že osciluje mezi několika stavy. Příkladem je těleso, které se na své cestě vesmírem dostane do blízkosti velmi hmotného tělesa. Po určitém čase se pohyb tohoto tělesa ustálí na eliptické dráze.



Je-li atraktor chaotický, znamená to, že výsledný atraktor nelze v podstatě nijak předpovědět. To je způsobeno tím, že je systém velmi citlivý na počáteční podmínky. Chaotičnost v tomto případě neznamena náhodnost, protože se hovoříme o deterministických systémech. Příkladem může být koule postavená na vrcholku jehlanu. Jakýkoliv vnější podnět způsobí, že koule tento stav opustí a dostane se do některého atraktoru (místo pod jehlanem). Tento atraktor nelze předpovědět, protože nemůžeme bez zásahu do měření zjistit počáteční podmínky. V kvantové fyzice existuje takzvaný princip neurčitosti, který říká, že pozorovatel měřením nějakého systému tento systém více či méně ovlivňuje. Čím přesnější má být měření, tím více je systém ovlivněn. Tato chyba se nedá nijak obejít a její velikost je rovna Planckově konstantě  $h$ . Lze říci, že chaotické atraktory jsou z velké části fraktály, však ne všechny fraktály jsou chaotické. Kritériem „fraktálnosti“ je soběpodobnost, tj. nezávislost tvaru na velikosti měřítka, což ani nevyklučuje ani nevyžaduje pravidelnost. Avšak soběpodobnost chaotických fraktálů je jistým druhem řádu, který je znakem deterministického chaosu. Fraktální struktura se objevuje i u tzv. diagramů bifurkací, což je větvení vlastností rovnic při změně parametrů, je to extrémní případ nestability, kdy jedna a tatáž situace má dvě různá, rozbíhající se řešení (Kukal, 2009).

Podivný atraktor je nejzajímavějším případem atraktoru. Tento typ atraktoru vzniká, je-li systém popsán minimálně třemi diferenciálními rovnicemi. Takový systém může mít velmi komplikovaný atraktor, který sice bude chaotický, ale přesto bude vykazovat určité pravidelnosti. Termín podivný atraktor není přesně matematicky definován, ale považujeme za něj takový atraktor, který vykazuje stejné vlastnosti, jaké mají fraktály (podivný atraktor je tedy fraktálem). Tento termín (strange attractor) poprvé zavedl ve své práci dr. Ruelle v roce 1970. První dynamický systém, který měl podivný atraktor,



vytvořil Ed Lorenz v roce 1963. Jednalo se o jednoduchý systém se třemi diferenciálními rovnicemi, které však ve svém důsledku vytvořily při simulaci chaotický atraktor, který měl fraktální strukturu. Na těchto rovnicích bylo také vidět velkou citlivost na počátečních podmínkách. (Gonzales, 2004)

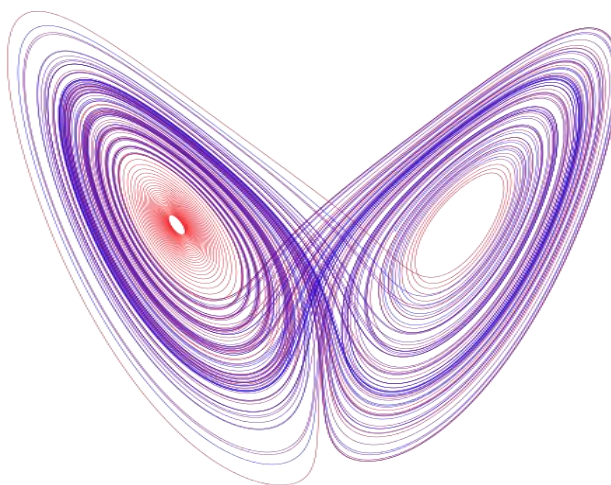
## 6.4 Motýlí efekt

Edward Lorenz působil začátkem 60. let minulého století na Massachusetts Institute of Technology, kde vytvořil jednoduchý matematický model zemské atmosféry, na kterém se pokoušel studovat počasí, konkrétně vynucenou konvekci v atmosféře. K simulacím použil z dnešního pohledu primitivní, ve své době však špičkový číslicový počítač, jednalo se o počítač Royal-McBee LGP-30 s 16kB paměti, který vypočetl 60 násobení za sekundu. Jeho výpočty byly s přesností na 6 platných číslic. Lorenz provedl zaokrouhlení počáteční podmínky na 3 platné číslice a očekával, že toto zaokrouhlení nebude mít vliv na výsledky, při-tom narazil na nestabilní chování matematického modelu. Postupně zjednodušil svůj matematický model, který měl původně 12 dimenzí až na známý třírozměrný Lorenzův systém z roku 1963.

Tento matematický model zachycuje základní vlastnosti konvektivního proudění atmosféry, která je zahřívána povrchem ze spodu a ochlazována z vrchu. Vzniká tak rotační pohyb částic vzduchu, kdy ohřátá částice stoupá, tím se ochlazuje a začne klesat, aby se opět zahřála a stoupala. Tento jev je známý jako Rayleigh-Bénárdova nestabilita. Okrajové podmínky jsou poněkud idealizovány: proudění v horní oblasti je považováno bez smykového napětí místo realističtější podmínky stejných rychlostí, v příčném směru je uvažována periodická okrajová podmínka místo omezení stěnami a celý případ je modelován jako rovinný místo prostorového. Jedná se o tzv. Reygleigh-Bénárdovu buňku, která se periodicky opakuje v příčném směru.

Dynamické systémy jsou charakterizovány limitním stavem - atraktorem, který nastává po určitém přechodovém čase, který závisí na počátečních podmínkách. Tento limitní, konečný stav může být zobrazen ve fázovém prostoru buďto jako bod - konečný stav klidu ke kterému systém spěje nebo jako limitní cyklus - uzavřená křivka, který odpovídá periodickému pohybu. Atraktor příslušející Lorenzovu systému viz. obrázek 6.2 za určitých podmínek

vybočuje z tohoto konceptu, nedojde k jeho ustálení ani po velmi dlouhém čase, vzniká nekonvergující křivka.



Obrázek 6.2: Model Lorenzova atrakatoru



Tento atraktor je prvním z tzv. "podivných atraktorů" charakterizujících chaotické chování dynamického systému, který byl podroben zevrubnému systematickému zkoumání. Tento atraktor má některé vskutku podivné vlastnosti:

Je tvořen spojitou křivkou v prostoru, která obecně začíná v jistém počátečním bodě, může však mít nekonečně velkou délku. Přitom vyplňuje jistý přesně vymezený podprostor ve fázovém prostoru, ze kterého nikdy nevybíhá;

- Nikdy neprotíná sám sebe, nekříží se ani se neopakuje;
- Má vlastnost fraktálů, tj. jeho struktura se opakuje na různých měřítkách;
- Jeho průběh v prostoru je náhodný, chaotický, nepředpověditelný.

## 6.5 Řízení deterministického chaosu

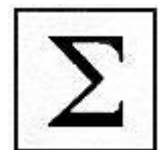
Řízením deterministického chaosu se rozumí takové působení na daný chaotický systém tak, aby se z režimu chaotického dostal systém do režimu periodického či neperiodického ustáleného stavu. To je základní myšlenka řízení deterministického chaosu. Jsou zde samozřejmě ještě i opačné trendy, tzn. takové řízení, jenž by daný systém přivedlo právě do chaotického režimu. Ačkoliv to zní absurdně, je faktem že některé děje mají optimální průběh právě při chaotickém chování (např. chemické reakce).

Podle Andrievského a Fradkova je řízení s otevřenou smyčkou založeno na perturbaci. Perturbace je funkce času nezávislá na řízeném procesu. Princip řízení perturbací nebo "řízení signálem programu," je vytváření řídicího signálu jako časové funkce neohledně na hodnoty kontrolovaného procesu, který je založen na rozlišném chování nelineárního systému akcí předurčeného externím vstupem  $u(t)$ , který může být buď jistá fyzická akce systému jako např. síla nebo pole nebo variace ("modulace") nějakého parametru kontrolovaného systému. Hlavní výhoda vychází z faktu, že řídicí proces je nezávislý na řízeném procesu. Tento přístup splní podmínku jednoduchosti, protože se obejde bez měření či senzorů. To je obzvláště důležité pro řízení super rychlých procesů působících například na molekulární nebo atomové úrovni, kde systémový stav nemůže být změřen (přinejmenším v reálném čase). Možnost variace dynamiky systému periodickou excitací je známa již dlouhou dobu. Například, jak bylo demonstrováno v první polovině minulého století, vysokofrekvenční excitace může stabilizovat kyvadlo v nejistém stavu. V mnoha systémech nejsou vždy všechny řídicí parametry zcela známy. V takovém případě si systém žádá adaptivní řízení, které je založeno na algoritmických statistických metodách nejmenších čtverců a maximální pravděpodobnosti odhadů. Je velmi důležité si uvědomit, že neexistuje žádná "nejlepší" metoda řízení. To znamená, že každý systém je individuální a reaguje jinak na různé metody řízení, kdy každá má svá pozitiva ale i negativa.



## Shrnutí kapitoly

- Teorie řízení chaosu je velmi široká interdisciplinární problematika, k níž bylo vypracováno od dob jejího vzniku více jak 1000 prací. Ty se zabývají jak řízením systémů se soustředěnými parametry, tak systémů s rozprostřenými parametry, což jsou systémy vykazující tzv. časoprostorový chaos. Význam řízení deterministického chaosu spočívá v možnosti řídit chaotické chování vhodné třídy systémů s tím, že lze pomocí tohoto řízení (vhodně spočítaných perturbací) převést systém do stabilního režimu v klasickém slova smyslu. Stejně tak lze systém převést to chaotického režimu je-li to nutné. Řízení obou typů systémů



má svůj význam a stává se součástí moderní teorie řízení. Lze očekávat, že řízení chaosu a speciálně chaosu časoprostorového bude nalézat stále širší aplikační uplatnění tak, jak to naznačují současné práce z tohoto oboru



### Kontrolní otázky a úkoly:

1. Co je chaos?
2. Co znamená, že je chaos deterministický?
3. Který atraktor simuluje konvektivní proudění v atmosféře?
4. Jaké základní charakteristické vlastnosti má křivka Lorenzova atraktoru?



### Otázky k zamyšlení:

1. Jak souvisí chaos s fraktály, pokuste se zamyslet.
2. Má model Lorenzova traktoru fraktální charakter?



### Citovaná a doporučená literatura

(Zelinka, 2006) Zelinka I., Čandík M. Fraktální geometrie – principy a aplikace. BEN – technická literatura, Praha 2006.

(Zelinka, 2003) Zelinka I. Řízení deterministického chaosu, Automatizace, 5, 2003, 310-315., ISSN 0005-125X.

(Zelinka, 2003) Zelinka I. Umělá inteligence. Hrozba nebo naděje? BEN – technická literatura, Praha 2003.

(web2): <http://www.root.cz/serialy/fraktaly-v-pocitacove-grafice/>, autor: Tišnovský P., fraktály v počítačové grafice, (listopad, 2009).

(web4): <http://www.fit.vutbr.cz/~tisnovpa/publikace/diplomka/doc/node1.html>, autor: Tišnovský P., Fraktály, (prosinec, 2009)

(web5): <http://amber.zine.cz/AZOld/Dimenze/fraktaly12.htm>, autor: Vachtl P., Fraktály a chaos, (prosinec, 2006).

(web8): <http://chaos.host.sk/>, autor: Tuleja S., Hanč J, Teorie chaosu a fraktály, (leden, 2010).

(Kukal, 2009) Kukal, J.: Dimenze a míry. Automatizace, ročník 52, číslo 5 (2009) pp. 292-293. ISSN 0005-125X.

(Mandelbrot, 1982) Mandelbrot, B. B. The Fractal Geometry of Nature. New York: W. H. Freeman and Co., 1982. ISBN 0-7167-1186-9

(Horák, 1996) Horák J., Krlín L. Deterministický chaos, Academia, ISBN 80-200-0416-5, 1996.

(Jick, 1996) Jick J. Chaos, Ando publishing, ISBN 80-86047-04-0, 1996.



## 7 Fraktály

**V této kapitole se dozvíte:**

- Základní pojmy
- Definici fraktálů
- Dělení fraktálů
- Využití fraktálů

**Po jejím prostudování byste měli být schopni:**

- Definovat fraktál
- Umět tvořit základní typy fraktálů
- Pochopit rozdíl mezi matematickým a přírodním fraktálem

**Klíčová slova kapitoly:** Fraktál, fraktální geometrie, soběpodobnost, soběpříbuznost, Kochova křivka, Sierpinského trojúhelník, Cantorova množina..

### Průvodce studiem

*Cílem této kapitoly je seznámení se základními pojmy fraktálů a fraktální geometrie. Tato kapitola by měla přinést lehký úvod a přehled ve fraktálech. Úvodem se v této první části budeme zabývat podstatou fraktálů od samého vzniku. Povíme si, jaký je rozdíl mezi fraktální a euklidovskou geometrií, dále si řekneme, jak fraktály vznikly, jak se dělí. Samozřejmě si některé základní typy fraktálů ukážeme a jednoduchým postupem se některé naučíme tvořit. V neposlední řadě se podíváme na využití fraktálů. Na zvládnutí této kapitoly budete potřebovat asi 2 hodiny, tak se pohodlně usadte a nenechte se nikým a ničím rušit.*



### 7.1 Úvod do fraktálů

Eukleidovská geometrie, která se učí na školách je vlastně naprosto primitivní. Učí nás pár vzorečků pro nemnoho objektů či tvarů a je už jedno kolika rozměrných. Každý si dovede představit krychli, kruh, čtverec a případně je i nakreslit a dokonce narýsovat. Staří Řekové jí popsali Platónský svět idejí,

statický a vlastníci oněch jednoduchých pár tvarů. Přesto tvrdili, že se tak dá popsat celý svět. Podobné tendence zaznamenáváme za těch 2500 let i v umění (obrazy, hudba- zejména v baroku a klasicismu a to přesto, že vyjádřit míru determinismu v hudbě je velice těžký úkol) a obecně ideálem krásy se většinou stane nesymetrický eukleidovský objekt. Zkusme se podívat třeba z okna skrz záclony a představit si je, jako soustavu půlválců a křivek (např. bezierových, pojem bezierovy křivky musíme znát například z počítačové grafiky). Patrně se nám to nikdy nepodaří naprosto přesně. Když totiž budeme zjemňovat měřítko, uvidíme další a další podrobnosti (např. otřepy na nitích). Musíme se tedy spokojit s aproximací, která je ovšem ztrátová. Nakonec se ukazuje, že v přírodě je velice těžké na každém kroku nenarazit na fraktální útvar. K popisu těchto útvarů se používá fraktální geometrie, bez níž by některé přírodní úkazy, nebo útvary byly nepochopitelné (např. turbulence). Slovo „fraktál“ pochází z latinského slova „*fractus*“, což znamená zlomený. Mandelbrot zvolil toto slovo ve svých pracích jako název pro objekty příliš nepravidelné pro běžnou matematiku.

Fraktály mají ovšem širší využití. Predikce (předpověď) počasí je toho nejlepším dokladem. Dále predikce v oblasti přírodě velice vzdálené, a to v oblasti ekonomiky.

## 7.2 Co je fraktál



*Fraktál je geometrický objekt, který po rozdělení na menší části vykazuje tvarovou podobnost s těmito částmi (Zelinka, 2006).*

*Fraktál je množina či geometrický útvar, jejíž Hausdorffova dimenze je (ostře) větší než dimenze topologická. (Mandelbrot, 1982)*

Fraktálními objekty se zabývá samostatná vědní disciplína nazývaná fraktální geometrie. Tato disciplína je intenzivně rozvíjena zhruba od šedesátých let minulého století. Za jejího zakladatele je dnes považován matematik Benoit B. Mandelbrot, který jako první matematicky definoval pojem fraktál. I před zavedením pojmů fraktál a fraktální geometrie se vědci a umělci zabývali geometrickými útvary, které dnes nazýváme fraktály, jako například sněhovou vločku Kochovu (Koch snowflake) nebo Sierpinského trojúhelník (Sierpinsky triangle).



Protože velká část fraktálů je využívána v počítačové grafice a fraktály lze nejlépe popsat jako geometrické objekty, můžeme fraktál nejjednodušeji definovat jako nekonečně členitý útvar. Opakem nekonečně členitého útvaru je geometricky hladký útvar, který lze popsat klasickou Euklidovskou geometrií.

Fraktály se dělí na:

- *Soběpodobné*

Kterákoliv část fraktálu je přesnou kopií původního motivu. Vyskytuje se jen u čistě matematických struktur, protože jednak jsme v přírodě omezeni velikostí částic (některé se zdají být nedělitelné) a dále těžko v přírodě vznikne takto dokonalý fraktál.

- *Soběpříbuzné*

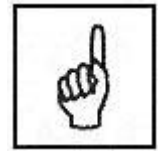
Je to určité zobecnění soběpodobnosti. Kterákoliv část fraktálu je velmi podobná, ne však zcela shodná s původním motivem.

Soběpříbuznost lze nalézt v reálné přírodě málem na každém kroku: je základní vlastností všech přírodních struktur a systémů, vesmírem počínaje, strukturou listu a tvarem mraků, profilem krajiny a větvením žil v lidském těle konče.

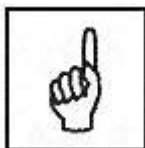
### 7.3 Klasické fraktály

Jedny z prvních fraktálů vznikly jako pokus o nalezení hranic matematických pojmů. Slavní matematikové Georg Cantor, Giuseppe Peano, David Hilbert, Niels Fabian Helge von Koch, Waclaw Sierpiński, Gaston Julia či Felix Hausdorff vymysleli různé matematické objekty vyhovující matematickým definicím, ale svými vlastnostmi byly velmi podivné. Například Kochova křivka, která je spojitá, avšak nikde nemá ani první derivaci. Tyto objekty byly považovány spíše za výjimky, za „matematická monstra“. Uveďme některé z těchto objektů.

Začneme asi nejznámější a také nejjednodušeji zkonstruovatelnou množinou, a to Cantorovou množinou. Cantorova množina je množina bodů z uzavřeného intervalu  $< 0 ; 1 >$ . Nejjednodušší definicí této množiny je popis, jak ji získat. Interval  $< 0 ; 1 >$  rozdělíme na tři shodné a otevřený interval  $(1/3 ; 2/3)$  vyjmeme. Čísla  $1/3$  a  $2/3$  nám tedy zůstanou v množině. Takto jsme získali dva uzavřené intervaly  $< 0 ; 1/3 >$  a  $< 2/3 ; 1 >$  o délce  $1/3$ . Nyní opakujeme tento postup na nové intervaly, tj. Z obou intervalů vyjmeme prostředek. To provádíme až do nekonečna. Body, které zůstanou, nám definují Cantorovu

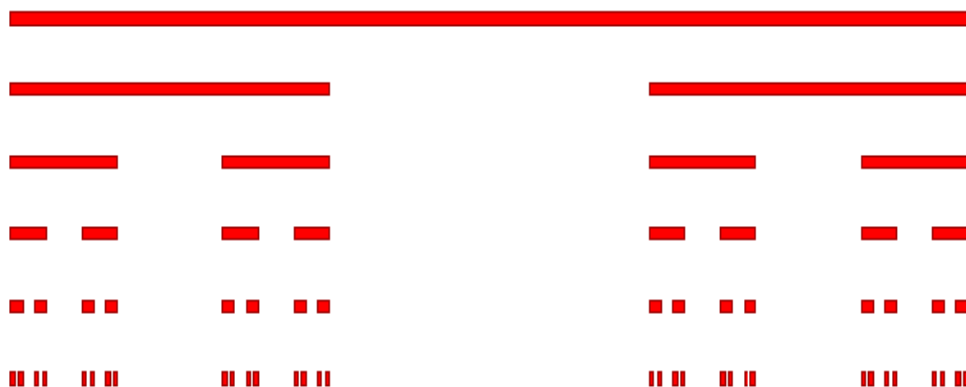


množinu.



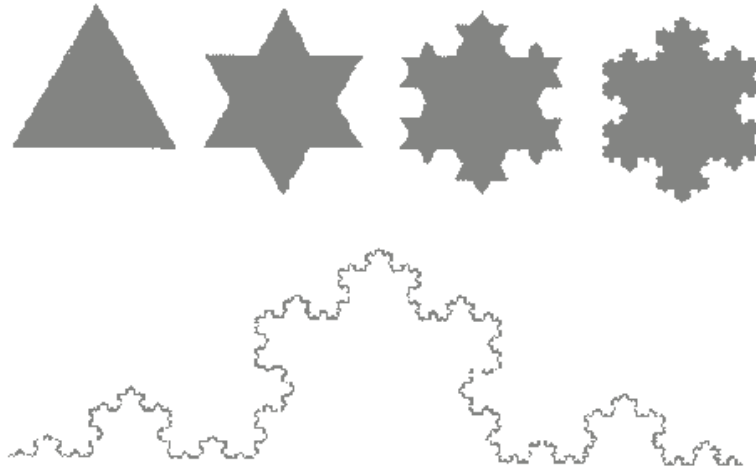
*Cantorova množina je množina bodů z uzavřeného intervalu  $\langle 0 ; 1 \rangle$*

Které body tedy patří do množiny? Jistě to jsou  $0, 1, 1/3, 2/3, 1/9, 2/9, 7/9, 8/9, \dots$ , tedy krajní body všech intervalů, kterých je spočetně mnoho. Nejsou to ale všechny body. Cantorova množina je nespočetná, zbývá tedy ještě nespočetně mnoho bodů, které náleží množině. Dále je třeba zmínit, že Cantorova množina je soběpodobná. Už z konstrukce je zřejmé, že např. části množiny v intervalech  $\langle 0 ; 1/3 \rangle$  a  $\langle 2/3 ; 1 \rangle$  jsou geometricky podobné celé množině, jen jsou zmenšeny v měřítku  $1/3$ . Kdybychom chtěli body patřící do této množiny popsat běžným způsobem, tak bychom narazili na problém. Nelze ji popsat několika podmínkami ani zapsat jako řešení nějaké jednoduché rovnice.



Obrázek 7.1: Konstrukce Cantorovy množiny

Velmi známá je i Kochova křivka, někdy je v upraveném tvaru nazývána Kochova vločka, nebo Kochův ostrov. Jak vlastně vzniká? Představme si trojúhelník, k jehož každé straně "přilepíme" k prostřední třetině další trojúhelník o třetinu menší. A tento postup budeme opakovat i na tento trojúhelníček. Po mnoha opakováních vznikne křivka s několika velice zajímavými vlastnostmi, které nalezneme i u dalších fraktálů. Tato křivka nikdy neprotne sebe sama, protože nové trojúhelníky jsou příliš malé, než aby si "překážely". Každá iterace křivku o malý kousek prodlouží, ale plocha zůstává na rozdíl od křivky konečná.



Obrázek 7.2: Kochova vločka

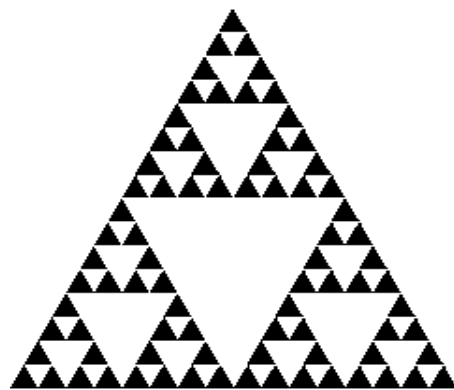
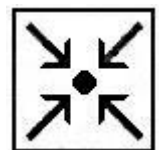
Mezi dvě základní vlastnosti Kochovy křivky patří:

- 1, Tato křivka nikdy neprotne sebe sama.
- 2, Iterace křivku o malý kousek prodlouží, ale plocha zůstává na rozdíl od křivky konečná.

Pomocí principu soběpodobnosti a iterací lze vytvořit mnoho různých fraktálů různých vlastností, tak i různých dimenzí. Za zmínku stojí ještě další dva klasické fraktály, které zavedl polský matematik Waław Sierpiński. Nejprve je uveden Sierpinského trojúhelník. Vytvoření tohoto fraktálu není složité, pouze se aplikují tyto metody:

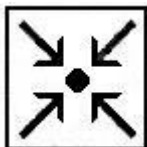
Příklad 1

1. Narýsujeme libovolný trojúhelník.
2. Vyznačíme střední příčky.
3. Vyjmeme prostřední trojúhelník, na zbylé tři opět aplikujeme body 2 a 3.



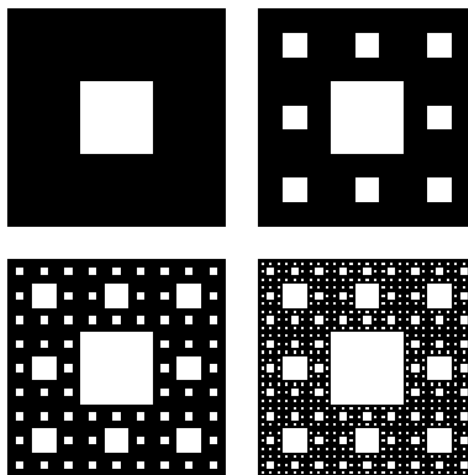
Obrázek 7.3: Sierpinského trojúhelník

Druhým jeho známým fraktálem je Sierpinského koberec. Konstruuje se následovně:



Příklad 2

1. vezmeme jednotkový čtverec, ten rozdělíme na 9 čtverců o straně  $1/3$
2. prostřední vyjme, takto můžeme postupovat do nekonečna.



Obrázek 7.4: Sierpinského koberec

## 7.4 Typy fraktálů

Tak, jako v každé vědní disciplíně, i ve fraktální geometrii se po určitém čase začaly rozlišovat jednotlivé typy fraktálů. Toto rozlišení není zavedeno jen kvůli systematickosti, ale i proto, že jednotlivé typy fraktálů jsou vhodné pro řešení určitého okruhu problémů, které se vyskytují v různých oborech. Také způsob vytváření (generování) se liší podle typu fraktálu (*Zelinka, 1999*), (*Zelinka, 2006*), (*Mandelbrot, 2003*), (web1,2,3).



Nejhrubší rozlišení fraktálů dle jejich generování je následující:

1. L-systémy
2. Systémy iterovaných funkcí IFS
3. Dynamické systémy
4. Nepravidelné fraktály

**L-systémy**-(Lindenmayerovy systémy) jsou skupinou fraktálů definovaných pomocí přepisovacích gramatik. Podstatou tvorby L-systémů je přepisování řetězců podle určitých pravidel. Každý symbol v řetězci má přiřazen jistý geometrický význam, například transformaci či generování objektu. S pomocí L-systémů lze generovat fraktály, které se podobají rostlinám, stromům a dalším přírodním útvarům. Poslední aplikace také směřují k využití těchto fraktálů při generování textur. Těmto fraktálům se také někdy říká graftály.

**Systémy IFS**- Jsou to generativní metody pro tvoření fraktálů. Tyto metody jsou vhodné jak pro generování fraktálů, tak i pro kompresi bitmapových obrazů, zavedeme-li tzv. *inverzní úlohu*. Zajímavé je, že tato metoda může být jak deterministická, tak i náhodná. Oboje však paradoxně vede ke stejnému výslednému fraktálu, použijeme-li dostatečný počet iterací.

**Dynamické systémy**- Dynamické systémy jsou pravděpodobně tím typem fraktálů, který má v technické praxi nejširší uplatnění. Dynamický systém je matematický model, který je závislý na nějaké proměnné, většinou to bývá na čase. Dynamický systém vychází z počátečních podmínek a je jimi v čase determinován. Existují dynamické systémy, které se po určitém čase neustálí v pevném stavu, ale ani nedivergují. Tento případ, který připomíná iracionální čísla, má většinou fraktální dynamiku a označuje se termínem deterministický chaos. Za dynamický systém v komplexní rovině lze považovat Mandelbrotovu množinu

**Nepřavidelné fraktály**- Další skupinou fraktálů jsou nepřavidelné fraktály. Zatímco všechny předchozí skupiny fraktálů byly v určitém smyslu symetrické, nepřavidelné fraktály vnáší při generování fraktálu do algoritmu náhodu. Tento typ fraktálů také umožňuje nejlepší popis přírodních objektů.

## 7.5 Výskyt a využití fraktálů nejen v přírodě

Přírodní fraktály oproti matematickým nejsou nikdy striktně soběpodobné, proto přírodní útvary nejsou při zvětšení přesně identické. Dnes již klasickým

případem jsou pobřeží ostrovů a břehy říčních toků, které je možné modelovat pomocí stochastických fraktálů. Dalšími příklady přírodních fraktálů jsou pohoří, oblaka, blesky, sněhové vločky, stromy i listí, cévní systémy živočichů, komůrky v plicích, DNA, buněčné kolonie, dále rozložení hmoty v galaxiích, hvězdokupy, ale i povrch, tvar trhlin a vad v obrobku či nástroji, časový vývoj akcií na burzách, časové změny inflace, zadlužení států i vývoj kurzu měny. Dá se říci, že nás fraktály obklopují a jsou všude, aniž bychom to tušili.

U přírodních fraktálů neznáme „algoritmus generování“ útvaru. Vlastnosti těchto fraktálů můžeme jen odhadovat. Přírodní fraktály jsou často multifraktály. Tento pojem značí, že má fraktál v určitých měřítcích odlišnou složitost. To je dáno různou vahou vlivů působících na vznik fraktálu v různých měřítcích. Multifraktály mohou být také generovány „uměle“, lze se tak setkat i s matematickými multi-fraktály.

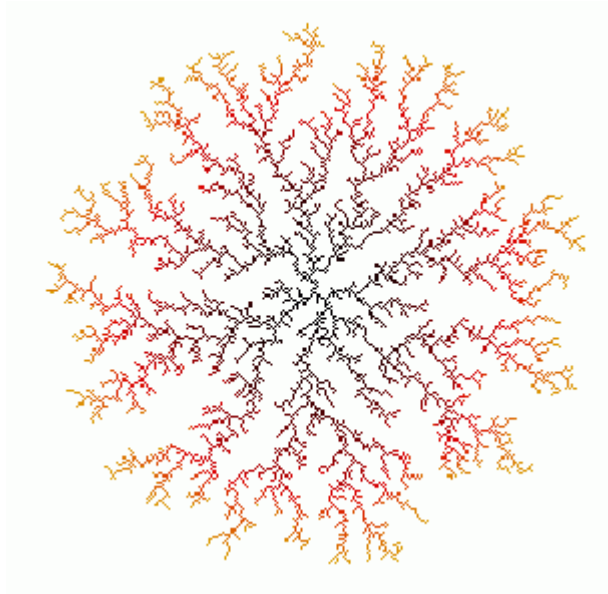
(Mandelbrot, 1982) a (Barnsley, 1989) se zabývají fraktální geometrií, soběpodobností a vztahem mezi řádem a chaosem.

Názvy těchto knižních titulů *Fractal Geometry of Nature* a *Fractals Everywhere* v žádném případě nepřehánějí – mnoho přírodních objektů vykazuje fraktální vlastnosti, například soběpodobnost či soběpříbuznost. A nejenom to: s fraktály se můžeme setkat i ve fyzice, chemii, biologii a mnoha dalších vědních disciplínách, které nemají se studiem geometrických objektů zdánlivě nic společného. Některé jevy, ve kterých lze nalézt fraktály, si nyní ukážeme.

### 7.5.1 Difúze

Difúze je (velmi zjednodušeně řečeno) přírodní jev, při kterém jsou pevné, kapalně či plynné částičky jedné hmoty postupně rozšiřovány v jiné pevné, kapalně či plynné hmotě. Pohyb těchto částiček, který se nazývá *Brownovým pohybem*, vzniká vlivem náhodných nárazů molekul jedné hmoty a hmoty druhé – viz obrázek 7.5. Poprvé byl Brownův pohyb popsán již v 19. století jako pohyb mikroskopických částiček pylu ve vodě. Zdůvodnění fyzikální podstaty tohoto pohybu však bylo podáno až Einsteinem v roce 1905. Základem pro studium různých rozšíření a zobecnění Brownova pohybu je takzvaný *Weinerův skalární Brownův pohyb* – ten lze, jakožto nejjednodušší formu Brownova pohybu, snadno analyzovat. Na difúzi se můžeme dívat jako

na stochastický fraktální objekt, na kterém je mnohdy už na první pohled patrná soběpříbuznost. Současně se jedná o typický příklad takzvaného *větévkatého fraktálu*. (web2)

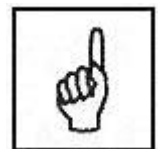


Obrázek 7.5: Difúze vizualizovaná počítačem

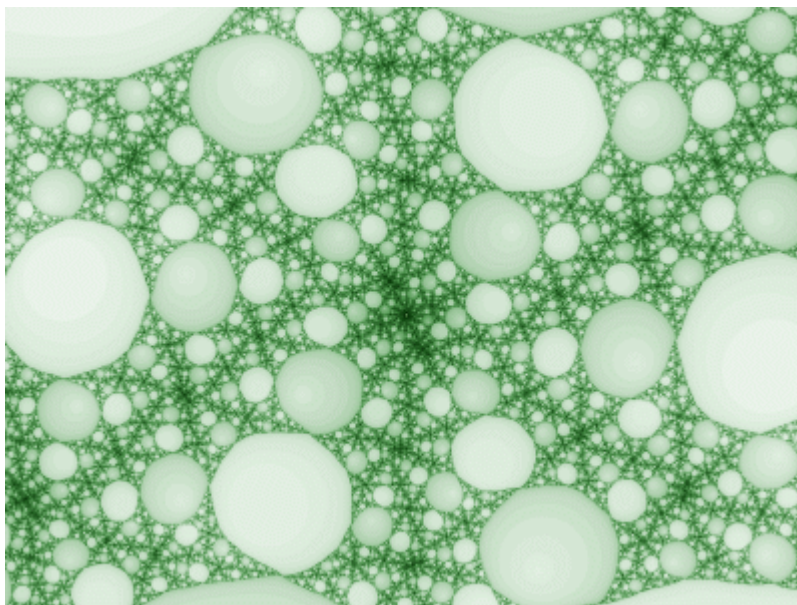
### 7.5.2 Magnetismus

Další vědní oblastí (na pomezí fyziky a chemie), ve které byly objeveny a využity fraktály, je zkoumání změny stavu některých materiálů. Podrobně byla studována například změna stavu magnetického materiálu na materiál nemagnetický (a naopak). Materiál je uspořádán z elementárních magnetů, které v závislosti na teplotě určují, zda je dané těleso jako celek magnetické či nikoliv. Při nízkých teplotách je materiál jako celek magnetický, ale při mikroskopickém pohledu existují jisté lokální fluktuace, kde nejsou elementární magnety uspořádané. Naopak, za dostatečně vysoké teploty je materiál jako celek nemagnetický, ale existují malé domény, ve kterých jsou elementární magnety uspořádány.

Vlastnost magnetismu je tedy závislá na měřítku pohledu na materiál. Při kritické teplotě (takzvaná *Curierova teplota*) materiál přechází z magnetického stavu do stavu nemagnetického a naopak. Jestliže je materiál zahříván na kritickou teplotu, vypadají jeho magnetické domény stejně (či podobně) ve všech měřítkách (zvětšení) a přitom mají fraktální strukturu. Při této teplotě



také nemůžeme jednoznačně určit, zda je materiál magnetický či nikoli. Objev fraktální podstaty magnetických domén při kritické teplotě některé vědce překvapil, protože panovala domněnka, že přechod z magnetického materiálu na materiál nemagnetický je v podstatě skokovou záležitostí – „rozpad“ orientace magnetů v nějaké části by se pak podle těchto domněnek měl velkou rychlostí šířit všemi směry.



Obrázek 7.6: Fraktály typu Magnet vykreslované v komplexní rovině nám přibližují možné rozložení magnetických dipólů při Courierově teplotě

### 7.5.3 Blesky

Tvary blesků, ať už přírodních či uměle vytvořených, je možné považovat za jednu z forem *větévkatých fraktálů*, podobně jako plošné či trojrozměrné obrázky difúze. Blesk totiž není tvořen pouze jednosměrným uspořádaným proudem elektronů (klasická klikatá čára). Na začátku a konci blesku nastává rozdělení celého proudu do několika menších větví, které se dále dělí podle vodivosti země a také podle tvaru (ostré předměty narušují homogenitu pole, proto blesky také více „přitahují“). Blesk jako celek je z těchto důvodů nápadně podobný ostatním větévkatým fraktálům a samozřejmě vykazuje i běžnou fraktální charakteristiku – soběpříbuznost (jde totiž o stochastický fraktál). Existují však i sofistikovanější modely, které používají například energetici. Na obrázku 18 můžete vidět přírodní bouři, která vytváří bleskové fraktální struktury. (*web4*)





Obrázek 7.7: Bouře vytvářející bleskové struktury

#### 7.5.4 Fraktály v lidském těle a medicíně

Fraktály a dynamické systémy se také používají při studiu chaosu – ostatně na fraktály se můžeme dívat jako na objekty ležící na hranici mezi řádem a (deterministickým či stochastickým) chaosem. Chaotické chování vykazuje signál srdečního rytmu - EKG signál, který vzniká při elektrické činnosti srdce a popisuje procesy šíření vzruchu srdeční tkáně. Také EEG signál je generován chaotickým systémem konečného řádu. Jednotlivé neurony v mozku nepracují samostatně, ale mají určité kolektivní chování – samoorganizaci. Při normální činnosti mozku je řád popisující jeho chování dosti vysoký a snižuje se při útlumu mozkové činnosti. Stejně postupy, které byly použity na signály srdečního rytmu, na EKG a EEG signál se dají aplikovat i na jiné signály snímané z lidského těla, chaotického chování vykazuje také stabilografický signál.

Posledních několik let můžeme vidět snahu využít fraktály a fraktální geometrii i v biologii, biochemii a medicíně. Například při zkoumání některých vlastností krve se měří Hausdorffova dimenze jejích částic. V lidském těle se fraktální struktury vyskytují v sítích cév, nervech a kanálcích, viz obrázek 20 (Holčík, 2001). Většina podrobně studovaných fraktálů v těle je systém trubic, např. těch, které transportují vzduch do plic a zase zpátky. West a Goldberger provedli detailní měření délek a průměrů trubic v této nepravidelné síti vzdušných kanálů a porovnali tato měření z částí plic lidí a některých jiných

savců. Navzdory jemným rozdílům zjistili, že typ měřítka předpovídá dimenzi fraktálu. Hans van Beek a James B. Bassingthwaighe použili fraktální geometrii k vysvětlení anomálií krevního toku v modelu zdravého srdce. Přerušení tohoto krevního toku může vyvolat infarkt myokardu. Fraktální struktura v srdci se nalézá také ve vrstvách tkáňových vláken připojených k srdci. Pokud se struktura těchto tkání poškodí, tak může být krev vyvrhována zpět z komor do síně a následuje překrvení srdce a srdeční porucha. Fraktální architekturu má také větvení Hisova - Purkyňova systému, který slouží k převodu elektrického signálu ze síní do komor. Z jakého důvodu vykazuje tak mnoho orgánových systémů fraktální strukturu? Objevuje se několik anatomických a fyziologických vysvětlení. Fraktální větve nebo lomy výrazně zvětšují povrchovou plochu vhodnou pro absorpci (například střevní výstelka), rozkládání a skládání (pomocí krevních cév, plicních trubic a žlučvodů) a zpracování informace (pomocí nervů). Fraktální struktury jsou díky své redundanci a nepravidelnosti robustní a odolné vůči zranění. Srdce může plynule pumpovat s relativně minimální mechanickou dysfunkcí navzdory velkému poškození Hisova – Purkyňova převodního systému. Fraktální struktury vznikají v lidském těle z pomalé dynamiky embryonálního vývoje a evoluce. K jejich generování je potřeba minimum informací a přitom přinášejí maximální biologickou odolnost organismu. Můžeme se proto domnívat, že fraktální struktura prošla evolučním vývojem jako nejvhodnější kandidát pro stabilní, dynamicky přizpůsobivou a prostředí odolnou metodu stavby určitých orgánů. Fraktální stavbu vykazují i keře a stromy a měření ukázalo, že každý druh má svou identickou fraktální dimenzi, kterou dodržuje při větvení, tloušťkách kmen a rozložení menších větví, stejně tak jako rozložení kořenů, kořínků až po drobné vlasečnice.

#### 7.5.5 Další využití

##### **Fraktální antény**

Velmi zajímavou aplikací fraktálů v technické praxi jsou takzvané *fraktální antény*. Jedná se o speciálně zkonstruované antény o různé vlnové délce, jejichž tvar je odvozen od L-systému (například křivky Helge von Kocha).

Tyto antény se vyznačují širokým pásmovým rozsahem a přitom jsou malé a kompaktní, což je předurčuje k využití například v mobilní technice nebo v armádě.

### **Architektura**

Fraktální geometrie (zejména L-systémy) byla použita při návrhu základní struktury velkých měst. Výsledkem tohoto využití byl návrh hierarchicky členěných ulic. Celkový půdorys města byl rozdělen do několika oblastí čtvercového tvaru o straně několika kilometrů. Hrany těchto čtverců tvořily největší komunikace, co se dopravní kapacity týká, čili čtyř a více pruhové dálnice. Každý čtverec se následně rozdělil do menších čtverců, kde opět hrany tvořily tentokrát již užší silnice s menší kapacitou. Toto dělení pokračovalo až do stavu, kdy byly ty nejmenší čtverce (tvořící parcely pro blok několika domů) rozděleny ulicemi s nejmenší dopravní kapacitou - typicky se jednalo o období našich obytných zón.

### **Kompresce dat**

Systémy iterovaných funkcí tvoří velmi důležitou skupinu lineárních deterministických fraktálů. V současnosti je nejvíce používána **fraktální komprese dat**, jelikož jedná transformace je zapsána pouze pomocí osmi čísel (šest čísel na zapsání transformace a dvě čísla na změnu kontrastu a jasu).

Metodu IFS komprimace využívá mimo jiné i grafický formát FIF. V porovnání s ostatními grafickými formáty dosahuje většinou lepších výsledků, nebo výsledků alespoň srovnatelných se ztrátovou metodou JPEG (kompresce pomocí IFS je také ztrátová metoda). Grafický formát FIF se příliš nerozšířil, protože proces komprimace je stále příliš náročný na výpočetní výkon (komprimace je až o několik řádů pomalejší než zpětná dekomprimace) a navíc algoritmus komprese je patentovaný firmou Iteration Systems, která také dodává hardwarové akcelerátory ve formě zásuvných karet do počítačů.

Dekompresní algoritmy jsou dostatečně rychlé pro programovou implementaci a jsou volně dostupné, například pro HTML prohlížeč Netscape a jeho následovníky existuje zásuvný modul určený pro zobrazování rastrových obrázků komprimovaných metodou FIF. Velkou výhodou tohoto formátu je možnost zobrazení obrázku ve velkém zvětšení bez viditelné ztráty kvality,

neboť fraktální komprimace obrazů může v některých případech (například u fotek většiny přírodních objektů) nabízet lepší poměr kvality obrazu k objemu dat, než diskrétní kosinová transformace (*DCT – discrete cosine transformation*) nebo transformace *WT – wavelet transformation*, jež jsou dnes masivně rozšířeny.

### **Komplexní síť**

Modelování a simulace komplexních systémů je velmi aktuální, důležité a zajímavé téma. Modelováním se lidé zabývají dlouho, ale teprve posledních 20 let umožňují výkonné počítače simulovat opravdu komplexní systémy - tedy takové systémy, které mají mnoho podsoučástí a vztahů, v nichž fungují složité interakce a jejichž studium jde napříč tradičními vědeckými obory. Mezi komplexní systémy patří zásadní problémy současnosti, jako jsou například změny klimatu, epidemie AIDS, stabilita ekonomických systémů nebo využití (ne)obnovitelných zdrojů. Pomocí modelování a simulace můžeme tyto systémy lépe uchopit a učit se o nich uvažovat. Sledovat simulace výpočetních modelů je navíc díky jejich přitažlivému vizuálnímu vzhledu velmi zajímavé. Dobré modely nás dokáží zaujmout a změnit náš pohled na svět. Dobrý model nereprodukuje skutečnost co nejděle, ale naopak mnoho věcí zjednodušuje a tím nám pomáhá pochopit alespoň část komplikované skutečnosti. Tato spojitost mezi modelováním a uměním není náhodná.

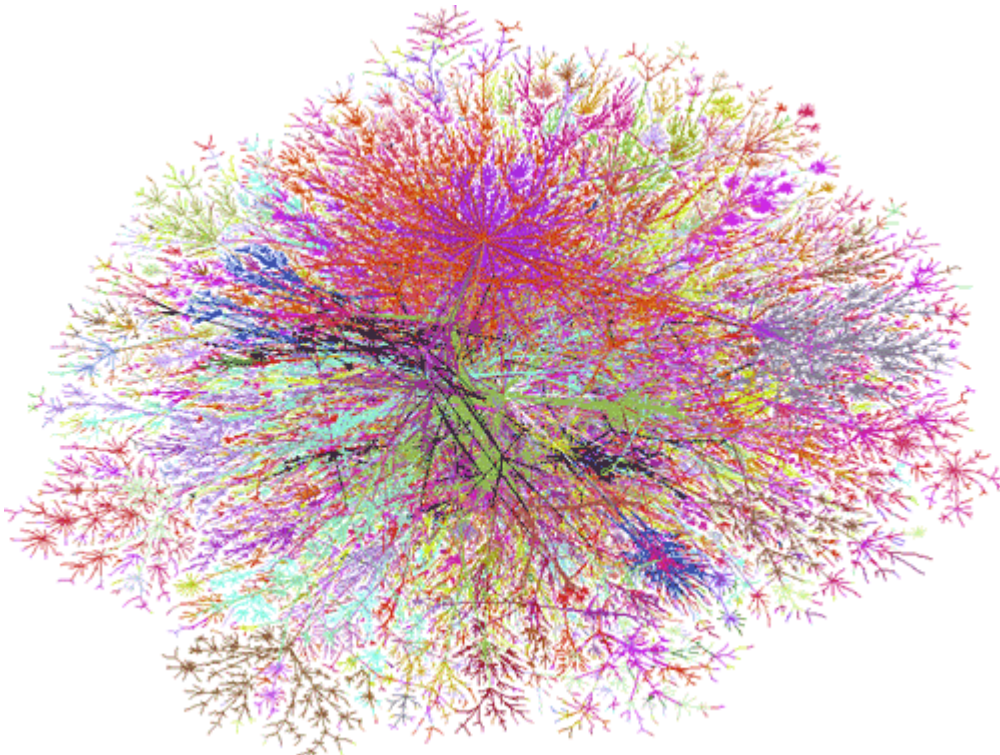
Komplexní síť jsou dalším typem aplikace sloužícím pro modelování a simulace. Nejsou určeny k tomu, aby modelovaly konkrétní systém, ale modelují obecný jev nebo obecnou vlastnost společnou pro mnoho subsystémů. Za komplexní síť můžeme považovat např. modely sociálních sítí nebo Internetu, regulačních sítí apod. Jde o velmi složitý a rozsáhlý graf, který má určité společné vlastnosti, např. vzdálenosti nebo stupně vrcholů.

Komplexní síť je ve skutečnosti kolekce prvků, které jsou propojeny odkazy, jako například:

- Lidé, kteří jsou přátelé,
- počítače, které jsou propojeny do sítě,
- neurony spojené synapsí,

- interakce proteinů atd.

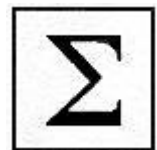
Sítě obecně jsou ideálním případem pro studium, protože jsou všude okolo nás a lze jimi modelovat čím dál více systémů. Proto se dnes zabýváme otázkou, jaké nástroje pro práci s rozsáhlými sítěmi vyvinout. Komplexní síť má také do určité míry fraktální charakter na základě soběpříbuznostní charakteristické vlastnosti, kdy můžeme rozsáhlé grafy zvětšovat a budeme nacházet podobné struktury v jejích útrobách. Na obrázku 7.8 můžete vidět strukturu komplexní sítě zobrazující internetem propojené vědecké centra.



Obrázek 7.8: Komplexní síť

## Shrnutí kapitoly

- Protože velká část fraktálů je využívána v počítačové grafice a fraktály lze nejlépe popsat jako geometrické objekty, můžeme fraktál nejjednodušeji definovat jako nekonečně členitý útvar. Opačem nekonečně členitého útvaru je geometricky hladký útvar, který lze popsat klasickou Euklidovskou geometrií. Fraktál je tedy geometrický objekt, který po rozdělení na menší části vykazuje určitou tvarovou podobnost s původním motivem. Mezi dvě základní vlastnosti Kochovy křivky patří: 1, Tato křivka nikdy neprotne sebe sama. 2, iterace křivku o malý kousek prodlouží, ale plocha zůstává na rozdíl od křivky



konečná. Podstatou tvorby L-systémů je přepisování řetězců podle určitých pravidel. Každý symbol v řetězci má přiřazen jistý geometrický význam. IFS jsou generativní metody pro tvoření fraktálů vhodné pro generování fraktálů a kompresi bitmapových obrazů. Dynamický systém vychází z počátečních podmínek a je jimi v čase determinován. Nepravidlené fraktály vnáší při generování fraktálu do algoritmu náhodu. Tento typ fraktálů také umožňuje nejlepší popis přírodních objektů.



### Příklad:

Fraktální struktury a samotné fraktály jsou v přírodě pravdu nejvíce patrné a jsou součástí studií univerzit po celém světě. Fraktály se vyskytují téměř ve všech vědních disciplínách, jako fyzika, ekonomie, ekologie, materiálové inženýrství, artware, oděvní průmysl, automobilový průmysl a mnoho jiného. Jak tedy souvisí fraktál a umělá inteligence?



### Kontrolní otázky a úkoly:

1. Co je fraktál?
2. Jaký je základní rozdíl mezi soběpodobností a soběpříbuzností?
3. Co je základní charakteristická vlastnost IFS?
4. Který typ fraktálů je z hlediska generování nejrozšířenější?
5. Jak vzniká Kochova křivka?
6. Jak souvisí fraktály a antény
7. Kde se všude můžeme s fraktály setkat



### Úkoly k textu

Pokuste se zamyslet a uveďte příklad, kde jste se v životě setkali s fraktálem. Dále se pokuste laicky vysvětlit co je fraktál a zamyslete se, proč se používá hojně fraktálů v počítačové grafice.



### Citovaná a doporučená literatura

(Zelinka, 2006) Zelinka I., Čandík M. *Fraktální geometrie – principy a aplikace*. BEN – technická literatura, Praha 2006.

- (Mandelbrot, 1982) Mandelbrot, B. B. *The Fractal Geometry of Nature*. New York: W. H. Freeman and Co., 1982. ISBN 0-7167-1186-9
- (Zelinka, 1999) Zelinka, I.: *Aplikovaná informatika*. VUT v Brně, Brno 1999.
- (web1): <http://www.fi.muni.cz/~sochor/PA010/Slajdy/Fraktaly.pdf>, autor: Sochor K., Fraktály, (listopad, 2009).
- (web2): <http://www.root.cz/serialy/fraktaly-v-pocitacove-grafice/>, autor: Tišnovský P., fraktály v počítačové grafice, (listopad, 2009).
- (web3): <http://www.elec.qmul.ac.uk/research/projects/pfd.htm>, autor: Lindsay A., Kontrola chaosu v plazmě, (leden, 2010).
- (Barnsley, 1989) Barnsley Michael *Fractals Everywhere*, Academic Press Inc., 1989, ISBN 0-12-079062-9.
- (web4): <http://www.fit.vutbr.cz/~tisnovpa/publikace/diplomka/doc/node1.html>, autor: Tišnovský P., Fraktály, (prosinec, 2009)
- (Holčík, 2001) Holčík, J., Fojt, O. *Modelování biologických systémů*. VUT v Brně, 2001. ISBN 80-214-2023-5.





## 8 Reprezentace znalostí

### V této kapitole se dozvíte:

- Co je reprezentace znalostí.
- Co jsou produkční systémy.
- K čemu slouží rámcová reprezentace.
- Co jsou to sémantické sítě.

### Po jejím prostudování byste měli být schopni:

- Vědět, co je reprezentace znalostí.
- Definovat pojem produkční systém včetně jeho vlastností.
- Znat problematiku rámcové reprezentace.
- Vědět k čemu slouží sémantické sítě.

**Klíčová slova kapitoly:** Reprezentace znalostí, rámce, sémantická síť, produkční systém.

### Průvodce studiem

*V této kapitole se dozvíte vše důležité o problematice reprezentace znalostí. Budou zde řešeny základní přístupy k reprezentaci znalostí postavené na sémantických sítích, rámcové reprezentaci nebo produkčních systémech. Vše podstatné bude vysvětleno na příkladech. Na zvládnutí této kapitoly budete potřebovat asi 3 hodiny, tak se pohodlně usad'te a nenechte se nikým a ničím rušit.*



### 8.1 Reprezentace znalostí

Reprezentace znalostí je jednou z nejdůležitějších a zároveň nejobtížnějších oblastí umělé inteligence. V souvislosti s rozvojem umělé inteligence se o znalostech začalo živěji hovořit od 80. let na základě potřeby vyvíjet efektivnější systémy. Kládl se důraz na to, co má být znalostní aplikací řešeno a až pak, jak to má být řešeno. Znalost se tak stala klíčovým prvkem znalostně orientovaných aplikací, např. expertních systémů. Na tvůrcích znalostního systému je zvolit vhodný způsob zviditelnění (zachycení, reprezentace)

znalostí, tzv. reprezentační schéma. Takové schéma si lze představit jako soubor pravidel a postupů, které mají být dodržovány pro zachycení znalostí. Schéma by mělo být pro člověka dostatečně srozumitelné, univerzální, resp. schopné zachytit širokou škálu znalostí – od obecnějších po specifitější. Dále by mělo podporovat snadnou modularitu (rozšiřování) znalostí. Jedním z ústředních problémů umělé inteligence je, zda existují takové struktury symbolů, které umožňují zachytit uvedené atributy poznatků. Tento problém se nazývá problém reprezentace znalostí.

Obvykle se při hledání vhodného reprezentačního formalismu postupuje tak, že se nejprve vytvoří hypotéza o tom, jak symboly vhodně organizovat do struktur, pomocí nichž by bylo možné zachytit alespoň některé z atributů poznatků. Poté se vytvořená představa implementuje v některém programovacím jazyce tak dlouho, dokud výsledný produkt nesplňuje požadavky na formalismus kladené.

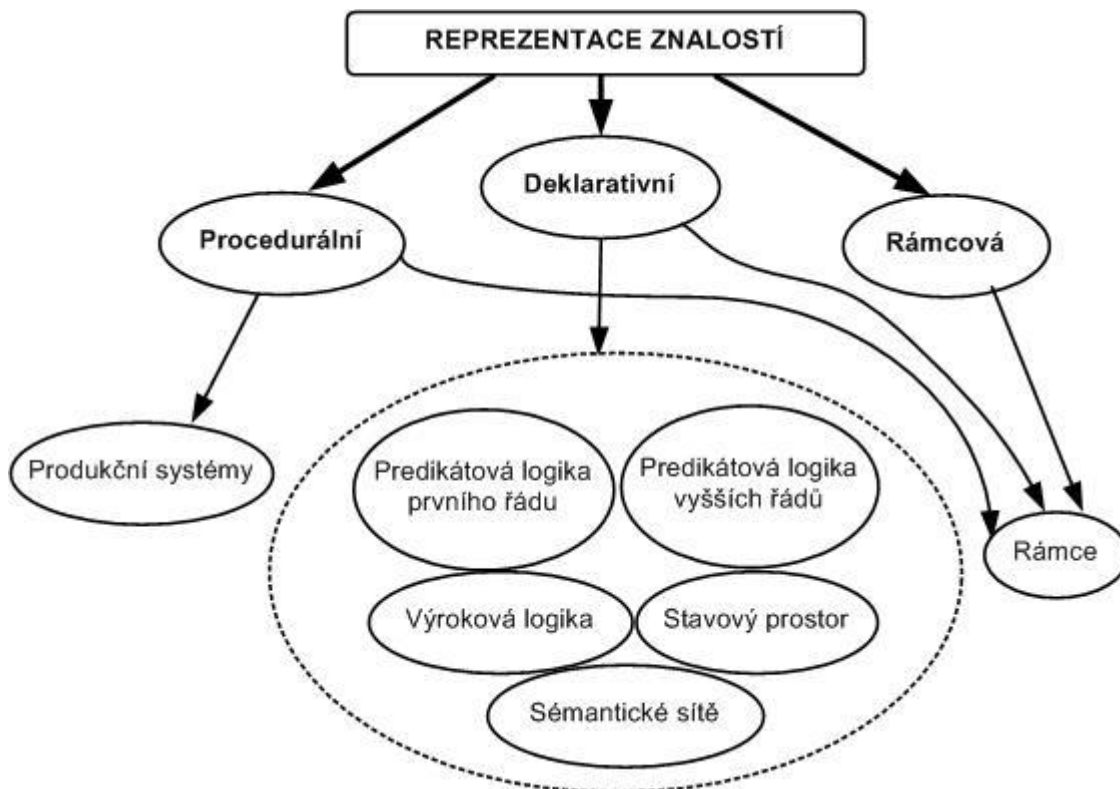
Některé prostředky reprezentace poznatků:

1. Deklarativní metody reprezentace - vyplynuly z atributu vyjádřitelnosti  
Z nich vycházejí produkční (pravidlové) systémy. Základní struktura pravidel vypadá takto:  
 $a \Rightarrow b$ , tj. pravidlo se skládá z předpokladové (podmínkové) části ( $a$ ) a uzávěrové (důsledové, akční) části ( $b$ ).
2. Asociativní metody reprezentace - kladou důraz na začlenitelnost  
Poznatky se reprezentují tak, aby se do nich explicitně promítly i souvislosti mezi nimi.
3. Procedurální reprezentace - dominuje zde atribut použitelnosti  
Význam reprezentovaných poznatků se definuje vymezením procedurálního kontextu, ve kterém se mohou reprezentované položky uplatnit. Defínuje se tedy, "co musím dělat, abych zjistil s danými předpoklady, zda daný objekt 'něco' je."
4. Rámcová reprezentace - spojuje v sobě všechny tři atributy  
Umožňuje práci s očekáváním (Sowa, 2000).

Problémem je, **jaký** způsob reprezentace použít pro zpracování **konkrétního** problému z reálného světa. M. Minsky tvrdí, že "... neexistuje jediný 'nejlepší'



způsob reprezentace znalostí. Každý typ problémů vyžaduje vhodné typy myšlení a uvažování - a vhodné typy reprezentace."



Obrázek 8.1: Struktura reprezentace znalostí

## 8.2 Produkční systémy

Na všechny produkční systémy můžeme na abstraktní úrovni pohlížet jako na "transformační procesy" - procesy, které transformují zdroje na užitečné výrobky a služby. Transformační proces používá k docílení změny obvyklé zdroje jako je práce, kapitál (na stroje a zařízení, materiály apod.) a prostor (pozemky, budovy apod.). Ekonomové nazývají tyto zdroje "výrobními faktory" a obvykle o nich mluví jako o práci, kapitálu a pozemcích. Výrobní manažeri je často nazývají "pěti M" (z anglického *men* - lidé, *machines* - stroje, *methods* - metody, *materials* - materiály a *money* - peníze). Pokud se na něj díváme jako na operační systém, může být produkční systém dále charakterizován toky (kanály pohybu) v procesu: jak fyzickými toky materiálů, rozpracovaných výrobků a hotových výrobků, tak toků informací a nevyhnutelných dokumentů, které doprovázejí fyzické toky. Fyzické toky jsou omezeny kapacitou produkčního systému, který omezuje rovněž schopnost systému vyhovět požadavkům na výstupy. Měřítkem výstupu systému může



podobně být kanál produkčního systému pro zpracování informací. Řízení informačních toků nebo plánování a řízení systému tak, aby bylo dosaženo přijatelných výstupů, je důležitým úkolem výrobního manažera (Kyonka, 2004).



Obrázek 8.2: Zobecněný produkční systém

I když je hlavním určujícím faktorem, jestli lze splnit nároky kladené na výstup a kapacita systému, za omezující faktor lze třeba považovat kvalitu. Kvalita produktu, měřená vzhledem k určitému objektivnímu standardu, zahrnuje vzhled, provozní charakteristiky, trvanlivost, praktičnost a jiné fyzické charakteristiky, dále pak včasnost dodávky, cenu, vhodnou dokumentaci a podpůrné materiály a podobně. Kvalita je důležitou částí definice systému.

### 8.2.1 Typy produkčních systémů



Produkční systémy dělíme na tři základní typy: **dávkový systém, průběžný systém a kusový systém**. V dávkovém systému jsou používána obecně použitelná zařízení a metody k produkci malých výstupních dávek se specifikacemi, které se od jedné dávky ke druhé výrazně liší. Dané množství produktů prochází jako dávka jedním nebo více kroky a na konci produkčního cyklu se současně objeví všechny produkty. Jako příklad uveďme systémy pro výrobu specializovaného nářadí nebo vysoce zatěžovaná stavební zařízení, speciální chemikálie a potravinářské výrobky, nebo v sektoru služeb, systém pro likvidaci škod ve velké pojišťovně. Dávkovým produkčním systémům často říkáme "práce na zakázku". V průběžném systému procházejí zpracovávané jednotky řadou kroků nebo operací, které jsou společné pro většinu ostatních zpracovávaných jednotek. Protože se očekávají velká výstupní množství, jsou pro to, aby se dosáhlo nízkých nákladů, často používána specializovaná zařízení a metody. Činnosti vykonávané jednotlivými pracovníky jsou často rozděleny na relativně malé úseky, které

Ize rychle ovládnout a efektivně vykonávat. Jako příklad můžeme uvést systémy pro montáž automobilových motorů nebo celých automobilů. Průběžné produkční systémy se často nazývají montážními linkami a jak je uvedeno níže, jsou běžné v hromadné výrobě (Křiláková, 2007). Doposud zmíněné dva typy systémů se často kombinují. Například při výrobě integrovaných obvodů pro elektroniku se v tuctech nebo dokonce stovkách kroků v dávkách zpracovávají tisíce obvodů na několika velkých řezech křemíkového krystalu. Malé obvody, z nichž každý měří pouze několik milimetrů, jsou pak odděleny a individuálně sestaveny s dalšími prvky obvodů na průběžné lince, a tak vzniká konečný produkt. Třetím typem produkčního systému je kusový systém. Pro jediný produkt daného typu, např. budovu nebo loď nebo prototyp letadla nebo velkého počítače se shromáždí prostředky pouze jednou. Díky speciální podstatě těchto systémů byly vyvinuty speciální metody řízení, aby se náklady udržely na přijatelné úrovni.

### 8.2.2 Úvahy k produkčním systémům

Po schválení obecných specifikací produkčního systému včetně přesné definice potřebných zdrojů a požadavků na výstup je třeba učinit tři důležitá rozhodnutí. Za prvé musí průmysloví inženýři, výrobní manažeři a jiní specialisté vybrat a vyprojektovat technologii, která bude používána. Jejich rozhodnutí musí zahrnovat volbu zařízení a nástrojů, rozdělení výrobních prostor a rozmístění zařízení, výběr pracovníků a pracovních postupů a mnohé jiné aspekty projektu procesu. Tyto volby je třeba provádět pečlivě; chyby v tomto časném stadiu mohou mít za následek ztrátu konkurenceschopnosti podniku nebo ztrátu jeho schopnosti udržet si výhodné postavení na trhu (Křiláková, 2007).

Dále musí být při zvolené technologii zvolena kapacita systému. Kapacita projektovaného systému je funkcí množství dostupného kapitálu, předpovědi požadavků na výstup a mnoha dalších méně významných faktorů. Rozhodnutí je opět třeba provádět uvážlivě. Stanovením příliš velké kapacity příliš brzo může firmu zatížit nadměrnými náklady a neefektivním provozem. Při příliš malé kapacitě může být obtížné a nákladné zvýšení výstupu v budoucnosti, pokud dojde k rychlému nárůstu trhu; to může způsobit firmě značné problémy



pokud konkurenti, kteří disponují větším zařízením, vyrábějí výrobek za nižší cenu nebo s lepší kvalitou (Mařík, 2003).

A nakonec musí být při určené základní kapacitě učiněna rozhodnutí o reakcích objemu výstupů na nevyhnutelné změny v požadavcích trhu, se kterými se firma setká. Kapacita většiny produkčních systémů se upravuje najmutím nebo propuštěním pracovníků, naplánováním přesčasových hodin nebo zkrácením pracovní doby, přidáním nebo zastavením strojů nebo celých oddělení nebo změnou rychlosti výroby v rozumných mezích. Účinnost každého z těchto mechanismů úpravy ve značné míře závisí na technologických omezeních samotného procesu, ekonomice odvětví a konkurenci. V některých odvětvích je úprava kapacity velice obtížným úkolem. Na příklad montážní linky se specializovaným zařízením jsou nejefektivnější při provozu na určité rychlosti, kterou nelze snížit nebo přerušovat bez značných ekonomických ztrát. V takových případech je pro celkový úspěch podniku zásadní velká péče při základním projektu produkčního systému.

### 8.2.3 Báze produkčních pravidel

Báze produkčních pravidel obsahuje položky, které nazýváme pravidla. Pravidla jsou grafickým zápisem poznatků. Skládají se ze dvou částí - předpokladové (podmínkové) a důsledkové (akční) - které bývají odděleny zpravidla šipkou (Mařík, 2003).

Výpočet probíhá tak, že systém prochází bázi faktů a kontroluje, zda jsou splněny všechny předpoklady pravidla (tj. zda se všechny prvky podmínkové části nacházejí v bázi faktů) a - je-li tomu tak - přidá důsledkovou část pravidla do báze faktů (což znamená, že systém ji tímto odvodil a přijal za platnou).



Schematicky je možno pravidlo zobrazit takto:

*předpoklady => důsledek (akce)*

nebo lépe:

*Jestliže  $P_1$  &  $P_2$  & ... &  $P_n$  pak  $D_1$  &  $D_2$  & ... &  $D_m$*

Při splnění předpokladů  $P_1$  až  $P_n$  vykoná systém akce z důsledkové části pravidla ( $D_1$  až  $D_m$ ). Obě části pravidel mohou mít různou vnitřní strukturu. Je

možné i to, že předpokladovou část tvoří soubor procedur, které na základě obsahu báze faktů počítají nové údaje, které se používají jako vstupy procedur, tvořících akční část pravidel. Ta potom počítá výslednou datovou strukturu. Obvykle se ale tato forma pravidel nepoužívá.

### 8.3 Sémantické síť

Sémantická síť byla navržena jako statická struktura pojmů z určité problémové domény, které jsou spojeny vazbami podle jejich významových souvislostí. Sémantické síť jsou v oblasti znalostního inženýrství často používány a jsou touto vědní disciplínou považovány za formu reprezentace znalosti. Sémantické síť jsou grafové struktury, které se skládají z množiny pojmenovaných vrcholů a množiny pojmenovaných orientovaných hran, spojujících *některé* dvojice vrcholů - každý uzel grafu odpovídá nějakému objektu z reality, každá hrana odpovídá binární relaci mezi objekty. Používají se na zviditelnění reprezentačních prostředků, umožňujících explicitní reprezentaci asociativních vazeb mezi reprezentovanými položkami (Gonzales, 1993).

#### 8.3.1 Vlastnosti znalostí

Přestože v současné době převládá názor, že znalost není možné definovat, že lze pouze vymezit její vlastnosti, většina autorů se shoduje na třech základních vlastnostech znalosti:

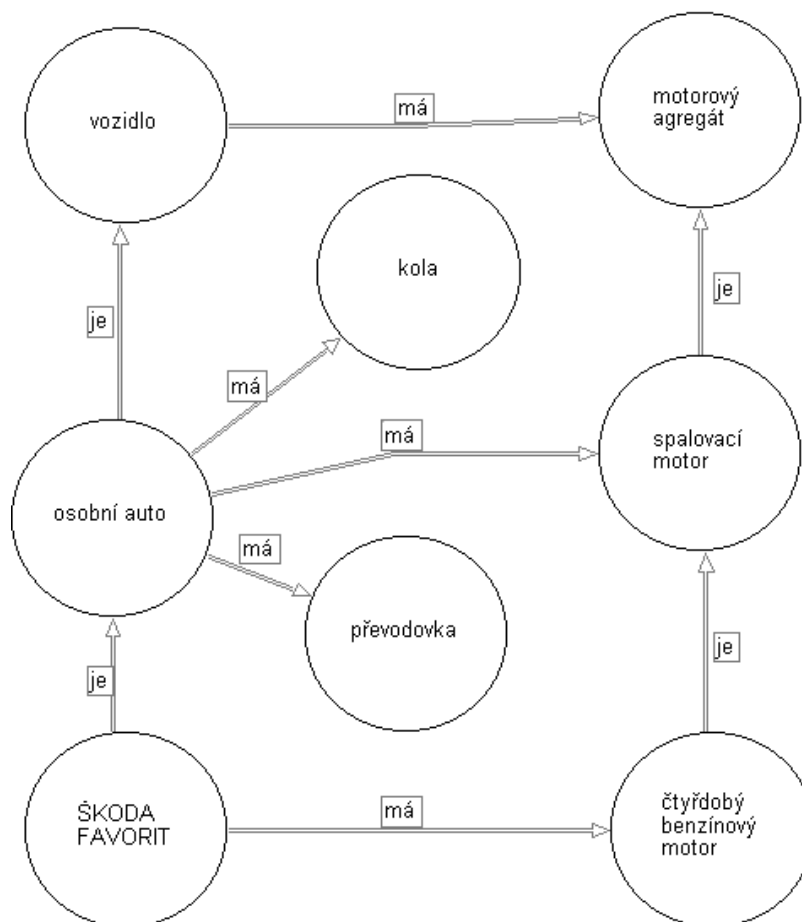
- základem znalosti je informace, která je nějakým způsobem obohacená, rozvinutá, transformovaná,
- znalost je dynamická, resp. zachycuje dynamiku, pohyb, postup,
- znalost je spjata s řešením nějakého problému.

Tyto tři atributy znalosti jsou klíčové pro vymezení pojmu znalost pohledem systémového inženýrství. Z hlediska této disciplíny je proto obecná sémantická síť jako forma reprezentace znalosti nevyhovující, neboť nemusí vždy zachycovat dynamiku a zejména vazby objektů sítě na problém a jeho úspěšné řešení (Gonzales, 1993).



### 8.3.2 Praktická ukázka sémantické sítě

Asociativní sítě se konstruují z vět v přirozeném jazyce pomocí predikátové logiky 1. řádu. Mají přirozenou grafickou reprezentaci:



Obrázek 8.3: Ukázka sémantické sítě

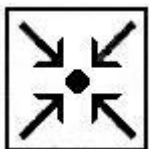
Předcházejícímu schématu odpovídají tyto věty, popisující realitu:

Osobní auto je vozidlo. Vozidlo má motorový agregát. Spalovací motor je motorový agregát. Osobní auto má spalovací motor. Osobní auto má kola a převodovku. Škoda Favorit je osobní auto. Škoda Favorit má čtyřdobý benzínový motor (který je spalovacím motorem a tedy motorovým agregátem) atd...

Tyto poznatky lze vyjádřit pomocí predikátové logiky 1. řádu takto:

$JE(x, \text{osobní auto}) \Rightarrow JE(x, \text{vozidlo})$

$JE(x, \text{vozidlo}) \Rightarrow MÁ(x, \text{motorový agregát})$





JE(x, spalovací motor) => JE(x, motorový agregát)

atd...

V sémantických sítích lze velmi pohodlně kódovat vztahy množinové inkluze a příslušnosti k množině, čímž lze vyjadřovat hierarchické uspořádání. Rozlišuje se mezi typem (Škoda Favorit) a instancí objektu (moje Škoda Favorit OPK 54-67). V hierarchii lze některé vlastnosti získávat přímo, jiné prostřednictvím dědění. Například fakt, že Škoda Favorit má benzínový motor, je v síti reprezentována přímo. Další vlastnosti lze získat odvozením. Škoda Favorit je osobní auto, má tedy kola, převodovku a spalovací motor, je zároveň vozidlem, má tedy motorový agregát atd.

Problémem sémantických sítí je nemožnost efektivně reprezentovat předpokládané (default) vlastnosti. Máme-li v síti reprezentován fakt, že Škoda Favorit má karburátor, nastane problém, když moje Škoda Favorit OPK 54-67 bude mít vstřikování paliva. Prostřednictvím dědění se totiž odvodí oba, vzájemně se vylučující fakty. Práci s default položkami umožňují reprezentace pomocí rámců. Sémantické sítě se používají v expertních systémech založených na dialogu s klientem, např. při rezervaci letenek (Jiroušek, 1995).

## 8.4 Rámcová reprezentace

Člověk používá svoje poznatky tak, jak to neumožňuje žádná z reprezentačních schémat. Reprezentace, kterou pravděpodobně používá lidská mysl, integruje ve stejné míře všechny atributy poznatků a podle situace, ve které se mysl ocitne, zdůrazňuje interpretační mechanismy vždy nejprůměrnější atribut.

Rámcová reprezentace poznatků vznikla v polovině 70. let. Její motivací bylo:

- 1.) syntéza existujících schémat,
- 2.) rozšíření dosavadních schémat o psychologickou motivaci.

Rámec je speciální datová struktura na reprezentaci stereotypních situací.

Dříve, než přijdeme do styku s nějakou konkrétní situací, máme už s ní spojeny nějaké očekávání a vědomosti. Podle Minského rámec obsahuje i více informací o tom, za jakých okolností a jak ho je možné použít. Rámec si můžeme představit jako formulář, tedy soubor rubrik, které jsou vyplněny



položkami rozličných typů. Údaje v rubrikách mohou být opět názvy jiných rámců, čím se jednotlivé rámce mohou spojovat do hierarchických sítí.

#### 8.4.1 Rámcové zobrazení struktury

Rámcovou strukturu reprezentace poznatků je možno znázornit ve formě tabulky nebo formuláře. Formulář má svoje *jméno*, které se může vyskytnout i v jiných formulářích - tak vznikají vazby mezi jednotlivými formuláři, vyjadřují se nadřazené skupiny či členství ve skupině. (Jiroušek, 1995) Každou tabulku je možné rozdělit na několik položek, které se nazývají *rubriky*. Počet rubrik v tabulce není fixovaný. Do rubrik se zapisují *položky*. Položky jsou buď povinné (musí být vyplněny), nebo nepovinné. Některé položky mohou vyjadřovat různé očekávání (angl. defaults). Při konkrétní aktivaci rámce se tyto očekávané hodnoty konfrontují se skutečností a v případě potřeby se nahradí skutečnými hodnotami. Například víme, že většina ptáků létá. Defaultová hodnota položky *létá* je *ANO*. Ale je známo, že tučňák ani pštros nelétají. Při aktivaci rámce Tučňák se proto nastaví do položky *létá* hodnota *NE* (Křiláková, 2007).



Tabulka 8.1: Zobrazení rámcových struktur

Jméno rámce	OSOBNÍ AUTO
Nadřazený rámec	Automobily
Podřazené rámce	Škoda Favorit Porsche 911 Ferrari Maranello

Tabulka 8.2: Zobrazení rámcových struktur 2

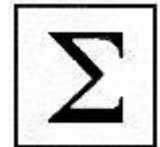
Jméno rámce	ŠKODA FAVORIT
Nadřazený rámec	Osobní auto
Podřazené rámce	Škoda Favorit OPK 54-67
Motor	Čtyřdobý benzínový
Převodovka	Manuální
Karburátor	Jikov

Chápání rámců jako specifických struktur údajů vedlo ke vzniku rámcových programovacích jazyků, jako jsou FRL (Frame Representation Language),

KRL (Knowledge Representation Language), OWL (One World Language) a jiné. Rámcová schémata jsou kombinací procedurálního a deklarativního přístupu. Deklarativnost spočívá ve způsobu zachycování informací o objektech reality pomocí slotů (vlastnosti rámce) a facet (hodnoty vlastností rámce). Procedurálnost je obsažena v procedurách, které mohou být součástí rámcové struktury.

## Shrnutí kapitoly

- Na všechny produkční systémy můžeme na abstraktní úrovni pohlížet jako na "transformační procesy" - procesy, které transformují zdroje na užitečné výrobky a služby. Produkční systémy dělíme na tři základní typy: **dávkový systém, průběžný systém a kusový systém**. Báze produkčních pravidel obsahuje položky, které nazýváme pravidla. Pravidla jsou grafickým zápisem poznatků. Skládají se ze dvou částí - předpokladové (podmínkové) a důsledkové (akční) - které bývají odděleny zpravidla šipkou. Sémantická síť byla navržena jako statická struktura pojmů z určité problémové domény, které jsou spojeny vazbami podle jejich významových souvislostí. Asociativní sítě se konstruují z vět v přirozeném jazyce pomocí predikátové logiky 1. řádu. Mají přirozenou grafickou reprezentaci. Rámcovou strukturu reprezentace poznatků je možno znázornit ve formě tabulky nebo formuláře. Formulář má svoje **jméno**, které se může vyskytnout i v jiných formulářích - tak vznikají vazby mezi jednotlivými formuláři, vyjadřují se nadřazené skupiny či členství ve skupině.



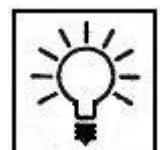
## Kontrolní otázky a úkoly:

1. Co je to reprezentace znalostí?
2. Jak by jste definoval sémantickou síť?
3. K čemu slouží produkční systémy?
4. Co je to rámcová reprezentace a čím se vyznačuje?



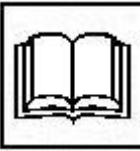
## Otázky k zamyšlení:

1. Pokuste se pomocí rámce zobrazit strukturu své rodiny.



**Citovaná a doporučená literatura**

- (Jiroušek, 1995) Jiroušek, R. *Metody reprezentace a zpracování znalostí v umělé inteligenci*. Praha, 1995. 104 s. Skripta. Vysoká škola ekonomická.
- (Gonzales, 1993) Gonzalez, A. J. and Daniel, D. D. *The Engineering of Knowledge-based Systems: Theory and Practice*. Englewood Cliffs, NJ: Prentice-Hall, 1993. ISBN 0-13-276940-9.
- (Mařík, 2003) Mařík, V., Štěpánková, O., Lažanský, J. *Umělá inteligence (4)*, Academia, Praha 2003.
- (Křiláková, 2007) Specialni, metody [online]. 2007-2010 [cit. 2010-03-21]. Vendulka.zcu.cz. Dostupné z WWW:  
<[vendulka.zcu.cz/Download/Free/spec\\_metody.doc](http://vendulka.zcu.cz/Download/Free/spec_metody.doc)>.
- (Sowa, 2000) SOWA, J.: *Knowledge Representation: Logical, Philosophical, and Computational Foundations*, Brooks/Cole Publishing Co., Pacific Grove, CA, 2000, ISBN 0-534-94965-7.



## 9 Robotika

### V této kapitole se dozvíte:

- jaká je charakteristika robotů,
- jaké jsou definice robotů,
- jaké je použití robotů.

### Po jejím prostudování byste měli být schopni:

- vysvětlit jak je chápán robot v USA a v Japonsku,
- uvést příklady robotů.

**Klíčová slova kapitoly:** Robot, manipulátor, samoříditelné vozidlo.

#### Průvodce studiem

*Již od doby Karla Čapka máme pořádně zamotanou hlavu z robotů v jeho dramatu R.U.R. Hovoříme-li o robotech, pak vlastně vyjadřujeme touhu po virtuálních lidech (vypadají jako lidé, ale nejsou to skuteční lidé) nebo po modelech lidí (zachovány jsou pouze podstatné vlastnosti). Tato kapitola se snaží vnést jasno do chápání pojmu robot tak, aby nešlo jen o formální vnější rysy (pohyb, ruka, oko), ale zejména o styl řešení úloh (záměr, působnost).*

*Na zvládnutí této kapitoly budete potřebovat asi 2 hodiny.*

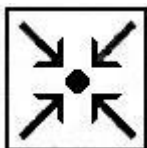


Robot je stroj vystavěný na principech umělé inteligence. Podle současného stavu a trendů je jasné, že robotika jako taková se stane nedílnou součástí naší civilizace. Její aplikace lze očekávat téměř na všech úrovních. V domácnosti jako domácí sluhý, v lékařství jako experty na různé speciální zákroky, ale také jako diagnostické systémy v podmínkách, kde nebude lékař. Ani mořské hlubiny nezůstanou výjimkou. I pro toto prostředí jsou dnes vyvíjeny roboty, které mají tvar ryby. Tyto roboty mají samozřejmě výzkumný charakter a slouží k odhalování tajemství mořských hlubin. To je vcelku významný úkol, vezme-li se v úvahu fakt, že právě moře by mohlo pomoci při surovinové a potravinové krizi, která velmi pravděpodobně lidstvo čeká.

## 9.1 Charakteristika robotů

Robot je mechanický nebo virtuální umělý systém, který na základě svého vzhledu nebo pohybu vytváří dojem, že má svůj vlastní účel nebo působnost. Slovo robot se může vztahovat jak k fyzikálním robotům, tak i k virtuálním softwarovým agentům. Zatímco se stále diskutuje o tom, které stroje spadají do kategorie robotů, je zřejmé, že typický robot bude muset mít (i když ne nezbytně) následující vlastnosti (Hornýak, 2006):

- nemá „přirozený“ původ, tzn. je uměle vytvořený;
- vnímá své životní prostředí a manipuluje s předměty v něm nebo je s nimi v interakci;
- má určitou schopnost provádět výběr podle daného prostředí, často s použitím automatického řízení nebo předem na- programované sekvence;
- je programovatelný;
- pohybuje se v jedné nebo více rotačních nebo translačních rovinách;
- provádí pohotové koordinované pohyby;
- jeví se, jako by měl záměr nebo působnost.

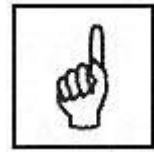


Pro robotické inženýry je fyzický vzhled stroje méně důležitý než způsob řízení jeho činnosti. Čím je patrnější, že systém řízení má působnost sám o sobě, tím spíše lze stroj nazvat robotem. Důležitým rysem působnosti je schopnost provádět výběr. Tedy čím více je stroj schopen skutečně zvolit něco jiného, o to má větší působnost (Lee 2005):

- vozidlo s hodinovým strojem nebude nikdy robot;
- dálkově řízené vozidlo lze někdy považovat za robota;
- vozidlo s palubním počítačem, které je řízeno programovatelnou sekvencí, lze nazvat robotem;
- samoříditelné vozidlo (např. vozidla bez řidiče, která vnímají své prostředí a rozhodují se na základě této informace) pravděpodobně lze nazvat robotem;
- vnímavé vozidlo, které se může rozhodovat, je volně navigovatelné a plynně komunikuje s člověkem, je běžně považováno za robota.

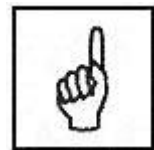
## 9.2 Definice a klasifikace robotů

Existuje řada různých definic robotů. Ve snaze poskytnout všeobecně přijatelnou definici stanovila Mezinárodní organizace pro standardizaci definici robota v normě ISO 8373, kde je robot definován jako „*automaticky řízený, opětovně programovatelný, víceúčelový manipulátor pro činnost ve třech nebo více osách, který může být buď upevněn na místě nebo mobilní k užití v průmyslových automatických aplikacích*“. Tato definice se používá při porovnávání počtu robotů v různých zemích.



Oproti definici ISO mají země, jako např. USA a Japonsko, odlišné definice robotů. Japonsko má velmi mnoho částečných robotů, protože zde je považováno za roboty více strojů. Vzhledem k tomu, že Japonsko i USA jsou důležitými subjekty ve vývoji robotů, uvádíme definice používané v těchto zemích.

Americký institut pro robotiku (RIA) definuje robota jako „*přeprogramovatelný vícefunkční manipulátor určený k přemísťování materiálu, součástek, nástrojů nebo specializovaných přístrojů pomocí různě naprogramovaných pohybů za účelem provádění různých úkolů*“.



RIA rozlišuje čtyři třídy robotů:

- manipulační zařízení s ručním řízením;
- automatické manipulační zařízení s předem určenými cykly;
- programovatelné, řízené servoroboty pohybující se od bodu k bodu po spojitě trajektorii;
- roboty, které k inteligentnímu pohybu vyžadují informace z prostředí.

Japonské robotické sdružení (JARA) rozděluje roboty do šesti tříd:

- ruční manipulační zařízení uváděné do chodu operátorem;
- robot s pevnou sekvencí;
- robot s proměnlivou sekvencí snadno měnitelnou řídicí sekvencí;
- přehrávací robot (playback), který může zaznamenat pohyb pro pozdější opakování;
- číslicově řízený robot s pohybovým programem a ručním ovládním;



- inteligentní robot, který rozumí prostředí a je schopen dokončit úkol navzdory změnám v provozních podmínkách.

Neexistuje žádná definice robota, která by vyhovovala všem, a tak mnoho lidí má svou vlastní definici (Peňázová, 2008), např. Joseph Engelberger, průkopník průmyslové robotiky, jednou poznamenal: „*Neumím definovat robota, ale poznám ho, když ho uvidím.*“

### 9.3 Příklady robotů

Použití umělé inteligence se v budoucnosti bude vyvíjet dvěma hlavními směry, a to v samostatných systémech a v systémech buď s *centrálně umístěnou inteligencí* nebo s tzv. *distribuovanou umělou inteligencí*. Pod pojmem samostatný systém lze rozumět např. plně samostatného robota, který se bude umět samostatně chovat v jakémkoliv prostředí bez pomoci vzdáleného supervizora, ať už jím bude člověk či nadřazený systém. Takováto zařízení jsou vyvíjena již dnes pro různé lidské činnosti, například roboty pro výzkum vulkánů či práci v jiných životu nebezpečných prostředích. Rovněž jsou používány v armádě pro průzkum nepřátelského prostředí ze vzduchu či země. Roboti dnes jsou převážně prototypy z větší míry ovládané ze země lidskou obsluhou. Nicméně podíl umělé inteligence na jejich řízení a rozhodování stále roste, protože v případě ztráty spojení musí být takovéto zařízení schopno samostatné činnosti. Není možné, aby se robot po ztrátě spojení zřítíl na zem nebo spadl do vulkánu. Navíc případné spojení se základnou může vést k jeho odhalení, a proto i z důvodů utajení by měly být roboti samostatní a bez jakéhokoliv spojení s kýmkoliv.



#### Roboti s centrálně umístěnou inteligencí

Ukázkou robota s centrálně umístěnou inteligencí, který je dnes jako prototyp zkoušen v námořnictvu USA, může být např. malý robotek – kamikadze, jež má za úkol vyhledávat miny a svým výbuchem je likvidovat. Má tvar šestinožného brouka a převrátí-li se, jednoduše si „překloubí“ nohy a pokračuje v cestě.

Další dnes již funkční ukázkou je robot pro hlídání vzdušného prostoru vyráběný ve Velké Británii, který je plně samostatný. Má zásobník



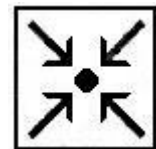
protiletadlových střel a samostatně pomocí radaru identifikuje objekt a rozhoduje o jeho zařazení (tj. zda je viděné letadlo nepřítele). Robot musí nejprve získat obraz letadla (radarový odraz), zpracovat jej („vytažení“ relevantních informací ze signálu), oklasifikovat (rozhodnout, co je daný objekt zač) a rozhodnout o zásahu (tj. zda daný letoun sestřelil nebo ignoroval). Ani jiné roboty nejsou výjimkou. Například prvky umělé inteligence v sobě obsahovalo i vozítko Sojourner ze sondy PathFinder. Pokud by sonda operující na Marsu či jiné vzdálené planetě neměla samostatnou, byť velmi primitivní inteligenci, která by umožnila samostatné jednání a orientaci v prostoru musela by při každém problému čekat dlouhé minuty na povel ze Země, což je nejen nevhodné, ale i nebezpečné (např. zpráva o průrvě + povel „stop“ ze Země).

### **Roboti s distribuovanou umělou inteligencí**

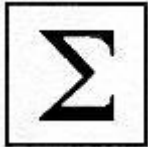
Systémy s distribuovanou umělou inteligencí jsou hierarchicky stupňovány tak, že nejvyšší inteligence je na nejvyšším stupni řízení a rozhodování a s klesajícími stupni „slábne“. Klasickým příkladem může být úsek továrny řízený počítači, které rozhodují o chodu jako celku a „nestarají“ se o činnost jednotlivých částí. Avšak s poklesem stupně řízení do nižších úrovní mizí význam celkového chodu systému a do popředí vystupuje význam činnosti jednotlivých jeho prvků.

Tyto systémy naleznou uplatnění nejen v továrnách, ale i jako samostatné jednotky. Např. robot – mozek, který se bude pohybovat se svým týmem robotů vykonavatelů v daném prostředí (cizí planety, nepřátelská území, doly ... ) a vykonávat daný úkol. Celý systém robotů je samozřejmě výhodnější než jeden samostatný robot, např. při havárii v případě samostatného robota – mozku bychom přišli o jednoho vykonavatele, jehož funkci by převzal jiný robot a mise by nebyla ohrožena. To je jistě jednodušší a levnější než v případě robota jako samostatného systému.

Lze rovněž očekávat, že nastane věk robotů miniaturizace. Již dnes existují roboty, které si svou velikostí nezdají s ptáky či obry hmyzí říše. Jejich využití je především v oblasti špionáže, ale lze je používat kdekoliv, kam se člověk nedostane. Od čištění nepřístupných míst až po superjemné opravy různých mikrostruktur. Mohou se podílet na likvidaci hmyzích škůdců, nebo na



likvidaci ekologických katastrof. Další z oblastí využití počítačového vidění je dálkový průzkum země (lesnictví, geodézie, meteorologie, archeologie, ...), lékařské aplikace (rozpoznávání rakovinných buněk, texturní analýza myokardu v echokardiografii, ...) a aplikace ve výrobě (počítadlo lahví, obsluha textilních strojů).



### Shrnutí kapitoly

Tato kapitola představuje stručný úvod do robotiky a snaží se objasnit chápání pojmu robot. Existuje řada různých definic robotů. Zde jsou uvedeny definice robota používané v USA a Japonsku. Závěr kapitoly je věnován příkladům použití robotů.



### Kontrolní otázky a úkoly:

1. Jaká je charakteristika robotů?
2. Vysvětlete jak je chápán robot v USA a v Japonsku.
3. Jaké je uplatnění robotů?



### Korespondenční úkoly

1. Vyberte si jednu oblast použití robotů a napište pojednání v rozsahu alespoň 5 stran.
2. Zpracujte historii robotiky v rozsahu alespoň 5 stran.

### Citovaná a doporučená literatura



Hornyak, T. N., *Loving the Machine. The Art and Science of Japanese Robots*. New York : Kodansha International, 2006.

Lee, D. G., *Axiomatic Design and Fabrication of Composite Structures*. Oxford : University Press, 2005. ISBN 0195178777.

Peňázová, M. Když se řekne robot. *Automatizace*. Vol. 51, No. 7-8 pp. 488-493. 2008

## 10 Zpracování přirozeného jazyka

**V této kapitole se dozvíte:**

- Jak probíhá zpracování přirozeného jazyka.
- Jak probíhá morfologická analýza.

**Po jejím prostudování byste měli být schopni:**

- objasnit princip rozpoznávání hlasu a řeči,
- charakterizovat textový korpus.

**Klíčová slova kapitoly:** Textový korpus, morfologická analýza, počítačová lematizace.

### **Průvodce studiem**

*K rozpoznávání lidské řeči pomocí počítače lze přistupovat ze dvou rozdílných hledisek, které plynou ze dvou hlavních aplikací této technologie. Bud' se rozpoznávají hlasy různých řečníků od sebe nebo je nezávisle na řečníkovi zjišťován obsah (význam) jeho promluvy. Při rozpoznávání jednotlivých řečníků dochází vlastně k identifikaci daného řečníka. Tato aplikace je zejména používána v identifikačních a autorizačních systémech. Rozpoznávání obsahu lidské řeči se zase využívá hlavně při komunikaci s počítačem nebo s automatizovanými systémy (telefonní systémy, systémy pro sběr dat, systémy řízení) jako rozhraní pro vstup dat.*

*Na zvládnutí této kapitoly budete potřebovat asi 2 hodiny.*



### **10.1 Jak mohou stroje slyšet a mluvit?**

V podstatě existují dva způsoby generování lidské řeči. *První* způsob spočívá v tom, že se vytvoří zvuková databáze předem namluvených slov, která se aktivují a přehrávají v závislosti na tom, jak jsou obsaženy v textu, jenž má být přečten. Tento způsob je dnes používán zejména u různých slovníků a překladačů. *Druhý* způsob spočívá opět ve vytvoření databáze, která

neobsahuje zvukové vzory slov, ale jejich matematické charakteristiky (dá-li se to tak říct) a ty pak při čtení využívá ke generování slov ve vhodném zvukovém generátoru. Zvukovým generátorem je zde software, hardware, nebo jejich vzájemná kombinace. Na rozdíl od prvního způsobu má výhodu v tom, že není tak náročný na objem dat. Malý objem dat vychází z faktu, že každé slovo je reprezentováno jistou matematickou charakteristikou a ne jeho celým zvukovým záznamem. V praxi se využívají oba způsoby generování lidské řeči.



Automatická analýza přirozeného jazyka počítačem vyžaduje rozdělit práci na několik menších, dobře definovaných podproblémů, které pak řešíme nezávisle. V oblasti zpracování přirozeného jazyka se mluví o tzv. *rovinách popisu jazyka*. Pro účely analýzy jazyka jsou tyto roviny uspořádány zdola nahoru, tj. od roviny nejjednodušší (zabývající se akustickou stránkou věci) po rovinu nejsložitější - rovinu významu. Každá rovina má své vlastní jednotky popisu pro definice vztahů uvnitř roviny a bezprostředně navazuje na rovinu vyšší (tzn. výstup z nižší roviny je vstup do následující vyšší roviny). Obvykle se hovoří o pěti až šesti rovinách (Hajíčova, Sgall 1985)

1. **Fonetická rovina:**

Vstupem do fonetické roviny je akustický signál, výstupem pak posloupnost fónů (zvuků — vektorů různých charakteristik).

2. **Fonologická rovina:**

Vstupem je posloupnost znaků čili písmen, výstupem pak posloupnost slov a interpunkce. Předzpracování můžeme rozdělit na roviny: pravopis, fonologie, morfonologie.

3. **Morfologická rovina:**

Vstupem je jak se dá předpokládat posloupnost slov a výstupem dvojice [lemma, značka]. Morfologické analýze se budu podrobněji věnovat v další části.

4. **Syntaktická (povrchová) rovina:**

Vstupem je posloupnost dvojic [lemma, značka].

Výstupem větná struktura (strom) s označením větných vztahů.

5. **Sémantická (tektogramatická, hloubková) rovina:**

Vstupem je větná struktura (strom) s pojmenováním vztahů.

Výstupem je rovněž stromová struktura, ale jsou odstraněna pomocná slova (hloubkové struktura věty).

#### 6. Pragmatická (logická) rovina:

Vstupem je hloubková struktura věty (propozice).

Výstupem je logická forma, která může být vyhodnocena (pravda/nepravda).

Někdy je vhodné některé roviny dále rozdělit, nebo naopak sloučit či přeskočit. To závisí na faktorech řešeného problému, např. jaký jazyk zpracováváme; nebo zda se jedná zpracování mluvené řeči; nebo jen překladu z jednoho jazyka do druhého atd.

#### Textový korpus

Textový korpus se většinou definuje jako rozsáhlý vnitřně strukturovaný a ucelený soubor textů daného jazyka elektronicky uložený a zpracováváný. Korpusy mohou vznikat pouhým převodem dostupných textů v elektronické podobě, což je velice levné řešení. Pro rozsáhlejší korpusy, které kladou velký důraz na reprezentativnost, existují poměrně rozsáhlá kritéria, co a jak se má do korpusu vložit. Například se setkáváme s problémem duplicit textů, nebo problémem zastaralých textů apod. Všechny větší korpusy většinou obsahují kromě čistě psaného textu také texty vzniklé přepisem mluvené řeči. Existují i mluvené korpusy obsahující výhradně mluvené slovo, u kterých je primární složkou zvukový záznam promluv. Takové korpusy slouží ke studiu výslovnosti nebo jako trénovací resp. testovací data pro syntézu a analýzu řeči. Každý rozsáhlejší systém, který má být snadno použitelný, musí vytvářet nějakou abstraktní úroveň s dostatečně jednoduchými prvky. V korpusu by takový základní prvek mohlo být slovo. Texty ovšem obsahují i jiné řetězce znaků: čísla a interpunkce (tečky, čárky, uvozovky apod.). Základním stavebním prvkem by tedy mělo být něco, co může být slovem, číslem nebo nějakým znaménkem. Tento prvek se označuje jako pozice. Korpus tedy tvoří posloupnost pozic.





### Morfologická analýza

Morfologická (tvaroslovná) analýza vstupuje do hry až v okamžiku, kdy jsou ve vstupním textu identifikována slova, mezery, interpunkce začátky vět, konce vět apod. Tento předpoklad je i z praktického hlediska nepříliš omezující, neboť většina existujících textových korpusů je již takto upravena. Lze říci, že úkolem „morfologické analýzy slova“ je určit morfologické kategorie příslušející danému slovu v textu. Při počítačovém zpracování je však situace třeba definovat a popsat mnohem přesněji. Především je třeba jasně rozlišovat mezi morfologickou kategorií a její hodnotou. V češtině rozlišujeme např. následující: slovní druh, rod, číslo, pád, osobu, čas, slovesný rod apod.. Hodnotami kategorie jsou např. čísla 1 až 7 pro české pády, "aktivní" a "pasivní" pro slovesný rod apod. (Hajič 2001). Morfologická analýza pracuje bez ohledu na kontext, tj. zpracovává izolovaně vždy jen jedno slovo (slovní tvar).

Pro počítačové zpracování se zavádí tzv. množina morfologických značek, z nichž každá reprezentuje hodnoty morfologických kategorií pro jeden slovní tvar. Pro vlastní zpracování se používá několik typů notací, z nichž nejrozšířenější je tzv. poziční notace, v níž se každé kategorii přiřadí pozice ve značce a každé hodnotě jeden znak, který se zapisuje na příslušnou pozici.

Pro každý slovní tvar určí všechny možnosti kombinací hodnot morfologických kategorií, které mu přísluší. Počítačová morfologická analýza však musí řešit ještě jeden problém, tzv. problém lematizace, jež určuje pro každý slovní tvar jeho základní podobu. Je nutno rozlišovat mezi slovy, která jsou v základním tvaru homonymní - např. stát (jako státní útvar) a stát (jako sloveso). Počítačová lematizace proto ještě navíc tato slova rozlišuje a jednoznačně identifikuje (např. připojením číselného indexu k základnímu tvaru slova, např. stát-1, stát-2 atd.). Morfologická analýza je v počítači realizována jako výpočetní procedura. Základní datovou strukturou pro daný přirozený jazyk je jeho morfologický slovník, který je používán vlastním algoritmem morfologické analýzy (v zásadě pak již na jazyce nezávislým).

### Příklad (Habiballa, 2004):

Tzv. "přímá" analýza slovních tvarů je založena na vyčerpávající analýze slova z hlediska možné segmentace na kmen a koncovku. Příkladem může být slovo



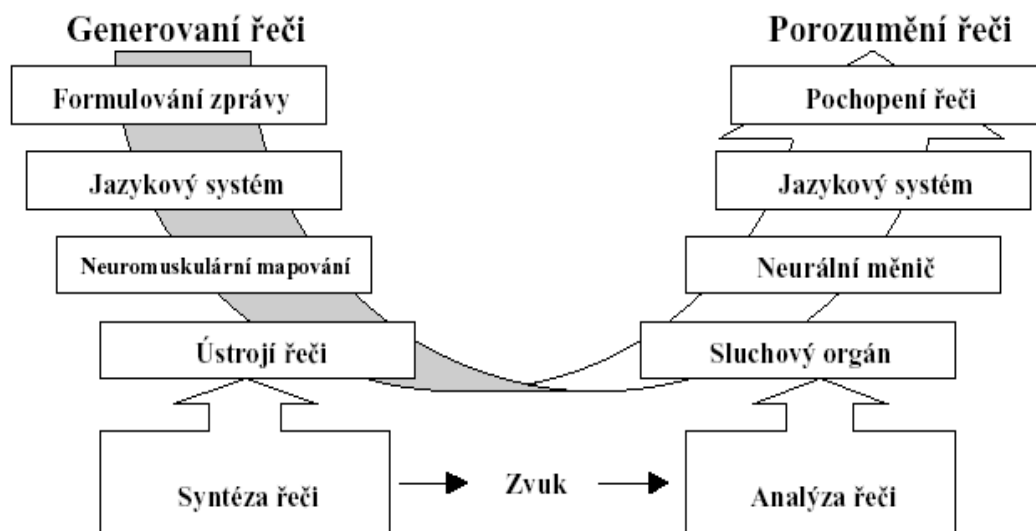
(slovní tvar) housky. Toto slovo je možno rozdělit na kmen housky + nulovou koncovku, nebo na housk + y, nebo na hous + ky, atd. až k h + ousky (kmen nulové délky se nepřipouští). Z těchto možností nakonec bude správná jen možnost hous + ky, neboť ve slovníku je neměnná část základu (zde jen hous, neboť 2. p. mn. čísla je hous+ek). Koncovky y, sky, a nulová koncovka jsou sice ve slovníku koncovek uvedeny také, ale kmen housk (hou) je nepřipouští (resp. nejsou uvedeny v seznamu koncovek pro vzor příslušný danému kmeni).

Pro zjednoznačnění výstupu morfologické analýzy se může využít značkování. Dnes se téměř výhradně používají pro značkování metody statistické, založené na strojovém učení (Hajič 2001). Počítač se tedy naučí, že po určitých předložkách následují jen některé pády, že na začátku věty nalezneme nejspíše pád první než jakýkoliv jiný apod. Nyní se naskýtá otázka: Jak se může počítač takovou věc naučit? Potřebuje k tomu (alespoň v dosud nejlépejších metodách) předem ručně označovaný korpus. Takový korpus je samozřejmě velmi pracnou záležitostí; pro spolehlivé naučení, kdy procento chyb klesá (pro češtinu) pod 5%, bylo třeba označovat přes 1.5 miliónu výskytů slov v textu. Označované korpusy jsou proto velmi cenným zdrojem lingvistických informací. Učení z ručně označovaného korpusu (takovému korpusu se říká trénovací data) může probíhat několika způsoby. Velmi jednoduchý a účinný (a dosud prakticky nepřekonaný) je postup, při kterém se spočítají relativní četnosti značek následujících po dvojici bezprostředně předcházejících značek v textu. Pro každou dvojici značek se tak vytvoří menší či větší tabulka, ve které jsou uvedeny relativní četnosti značek po ní následujících v trénovacích datech. Jakkoli je tento systém lingvisticky jasně neadekvátní, značkování založené na efektivním algoritmu aplikace těchto tabulek dává na kontinuální text velmi dobré výsledky: pro angličtinu se dosahuje i méně než 3% chyb na prakticky libovolném textu, pro češtinu pak okolo 5%.

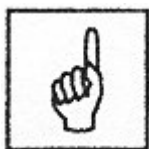
## 10.2 Rozpoznávání hlasu a řeči

Základní jednotkou popisu řeči jako zvukové formy je elementární zvukový segment. Volba jeho délky ovlivňuje další zpracování řeči. Segment musí být dostatečně krátký, musí být schopen tvořit další jednotky tzv. řetězením. Nutná je znalost vztahu segmentu k jazyku jako lingvistické formě. Při zpracování

řeči lze kombinovat segmenty o různých velikostech. Základní řečová jednotka je foném, jež označuje minimální fonetickou jednotku identifikující jednotlivé primitivní zvuky. Počet fonémů v existujících v českém jazyce je 36. Fonémy se skládají do větších celků – slabik. Libovolná promluva je potom vlastně opakování různých slabik.



Obrázek 10.1: Přenos informací pomocí řeči (Habiballa 2004).



Přenos informací řeči se pro komunikaci člověk-člověk rozděluje na dvě části. První část je *generování* řeči (vytváření), druhá část je *rozpoznání* řeči (porozumění) viz obrázek 10.1. Mezi těmito dvěma celky vystupuje řeč (zvuk) jako médium pro přenos informace (informační kanál).

### Faktory ovlivňující hlasový vstup

Hlasový vstup je ovlivněn mnoha faktory. První je řečová složitost vstupu. Na aplikace pro zadávání příkazů a ovládání jsou kladeny nejmenší nároky. Jedná se o přístup, kdy člověk ovládá zařízení pouze jednoduchými (jedno nebo více slovními) příkazy. Tyto příkazy se rozpoznají ze slovníkové databáze čítající několik desítek slov, na jejich základě se provedou případné povely a pokud to bylo vyžádáno, tak se v posloupnosti slov či v jednoduchých větách podá uživateli zpráva o výsledku příkazu. Vyšší nároky jsou kladeny na aplikace pro diktát. Tyto aplikace se musejí vyrovnat s plynulou řečí a nezávislosti na mluvčím. Nejvyšší nároky jsou na aplikace pro diktát s integrovaným



porozuměním mluvenému jazyku. Další hledisko je prostředí použití. Nejjednodušší je použití v tichém prostředí a pro jednoho mluvčího. Obtížnost stoupá s použitím plynulé řeči, nezávislosti na prostředí a na mluvčím.

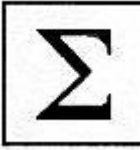
### **Charakteristika akustického signálu**

Informace obsažená v mluvené řeči je obvykle získávána mikrofonem. Pro další zpracování je zapotřebí tyto analogové kmity převést do číslicových údajů. Tento proces se nazývá *digitalizace* a zahrnuje provedení dvou kroků, a to vzorkování a kvantizaci s kódováním. Důležitým předpokladem při zpracování akustického signálu je fakt, že se jeho vlastnosti v průběhu času mění pomalu, což umožňuje dále signál zpracovávat pomocí metod krátkodobé analýzy, při níž se úseky řeči vydělují a zpracovávají tak, jako by to byly krátké, na sobě nezávislé zvuky. Výsledkem analýzy je potom číslo, či vektor čísel, který jistým způsobem charakterizuje daný zvuk, který se dalšími metodami vylepšuje, zpracovává a nakonec se realizuje jeho rozpoznání. Metody rozpoznávání signálu vychází z matematických metod, jsou to např. metody syntézy řeči. Syntezátory řeči můžeme rozdělit na tři skupiny. Jsou to syntezátory založené na zpracování řečového signálu v kmitočtové oblasti (vychází z modelování procesu vytváření řeči - matematický model artikulačního ústrojí) a v časové oblasti.

Při syntéze řeči v časové oblasti je spojitý akustický signál převeden na digitální kód s možností jeho rekonstrukce v určitém časovém okamžiku. Tento způsob syntézy je relativně jednoduchý, technicky nenáročný a produkuje kvalitní syntetickou řeč. V mnoha případech lze zachovat charakteristiky mluvčího (nejsou-li editovány). Řeč je kódována, digitalizována a vzorky jsou uloženy v paměti. Po provedení syntézy řetězením je zpětně dekodována na analogový tvar. Nevýhoda syntézy založené na řetězení izolovaně promluvených a zakódovaných slov je především velikost slovníku, tj. metoda je vhodná pro méně rozsáhlé slovníky (např. pro slovníky profesně zaměřené), pro povely, příkazy a hlášení.

## Shrnutí kapitoly

Tato kapitola představuje stručný úvod do strojového zpracování přirozeného jazyka. K této problematice lze přistupovat ze dvou rozdílných hledisek. Buď se rozpoznávají hlasy různých řečníků od sebe nebo je nezávisle na řečníkovi zjišťován obsah (význam) jeho promluvy.



### Kontrolní otázky a úkoly:

1. Objasněte princip rozpoznávání hlasu a řeči.
2. Co je to textový korpus?
3. Jak probíhá morfologická analýza?



### Korespondenční úkoly

Nalezněte konkrétní aplikace zpracování přirozeného jazyka a napište na toto téma pojednání v rozsahu alespoň 5 stran.



### Citovaná a doporučená literatura

Hajič, J., Statistické modelování a automatická analýza přirozeného jazyka (morfologie, syntax, překlad). Jarošová, A. (ed.) *Slovenčina a čeština v počítačovom spracovaní*. VEDA, vydavateľstvo SAV, Bratislava. 2001.

Habiballa, H.: *Umělá inteligence. Texty pro distanční studium*. Ostravská univerzita. Ostrava 2004.

Hajíčková, E., Sgall, P., Towards an automatic identification of topic and focus, *ACL Proceedings, Second European Conference*, s.263-7, 1985

## 11 Počítačové vidění

**V této kapitole se dozvíte:**

- jaké jsou principy počítačového vidění,
- jaké jsou techniky zpracování 2D obrazů,
- jaké jsou oblasti použití počítačového vidění.

**Po jejím prostudování byste měli být schopni:**

- vysvětlit principy počítačového vidění,
- objasnit jaké jsou techniky zpracování 2D obrazů.

**Klíčová slova kapitoly:** Segmentace obrazu. digitalizace, prahování, klasifikace.

### Průvodce studiem

Počítačové vidění může být označeno za technologii, ale také za vědní obor, který je dnes ve velké míře spojován s oborem umělé inteligence. Počítačové vidění jako vědní obor zkoumá teoretické možnosti, pro vytvoření umělých systémů schopných vidět nebo porozumět obrazům. Jako technická disciplína bývá počítačové vidění označováno za kombinaci počítače, snímacího zařízení a softwaru, který je schopný snímanou předlohu zpracovat nebo klasifikovat.

Oblastmi využití počítačového vidění je dálkový průzkum země (lesnictví, geodézie, meteorologie, archeologie, ...), lékařské aplikace (rozpoznávání rakovinných buněk, texturní analýza myokardu v echokardiografii, ...) a aplikace ve výrobě (počítadlo lahví, obsluha textilních strojů).

Na zvládnutí této kapitoly budete potřebovat asi 2 hodiny.



### 11.1 Princip počítačového vidění

Úlohou počítačového vidění je napodobit pomocí technických prostředků schopnosti lidského oka. Jde nejen o samotnou schopnost zpracování obrazu, ale také o rozpoznání toho, co obraz představuje. V tom hraje podstatnou roli

inteligence člověka a jeho předchozí zkušenosti. Počítačové vidění lze rozdělit do dvou oblastí: rozpoznávání 2D obrazu (např. textu a znaků na papíře) a rozpoznávání 3D obrazu (např. rozpoznávání součástky pro její následné uchopení).



Důležitou částí počítačového vidění jsou techniky zpracování 2D obrazů.

(Šonka, Hlaváč 1992):

1. Získání digitálního obrazu.
2. Předzpracování digitálního obrazu.
3. Segmentace obrazu - rozložení digitálního obrazu na objekty.
4. Popis objektů digitálního obrazu.
5. Klasifikace objektů digitálního obrazu.

### **Získání digitálního obrazu**

Snímání digitálního obrazu je proces, kdy se používají zařízení typu scanner či videokamera k získání vstupní spojitě fyzikální veličiny a jejich převod na spojitý elektrický signál. Vstupní informací může být jas, intenzita rentgenového záření, intenzita úměrná teplotě v termovizi apod. Snímat můžeme v jednom nebo více spektrálních pásmech.

*Digitalizací* obrazu rozumíme převod vstupního spojitého signálu na diskretní tvar. Vstupní analogový signál je popsán funkcí  $f(i,j)$  představující souřadnice v obrazu. Funkční hodnota odpovídá jasu (intenzitě). Digitalizace se provádí tak, že se příslušný obraz zaostří na plochu čidel, a ta generují elektrický signál úměrný intenzitě obrazu. Dochází tak k diskretnímu převodu jak plošnému, tak amplitudovému, tj. signál se vzorkuje a kvantuje. Při digitalizaci bohužel vždy dojde ke ztrátě určité informace. *Plošná diskretizace* (vzorkování) je proces, kdy se původně spojitý obrazový signál převede na nespojitý a to tak, že se plocha nejprve rozdělí na body, kde se umístí čidla, která snímají velikost elektrického signálu obrazu (pokud se daný signál v rámci čidla změní, pak se obvykle vezme jen jeho střední hodnota). *Amplitudová diskretizace* (kvantování) je převod spojitého signálu na diskretní a to tak, že se úroveň signálu popisuje binárními čísly (což je v podstatě totéž jako jeho náhrada schodovou funkcí). Kvantováním se tedy spojitě hodnotě jednoho vzorku přidělí diskretní hodnota, obvykle jedna z 256 možných jasových úrovní. Po

digitalizaci obrazu dostaneme matici přirozených čísel popisujících obraz. Získání digitálního obrazu významně ovlivňuje všechny ostatní procesy, neboť úspěšnost dalších kroků záleží především na jeho kvalitě (tj. pokud získáme obraz v nežádoucí kvalitě, v němž nebudou díky horšímu snímacímu zařízení zachyceny jisté detaily, pak také nebudou nalezeny a zpracovány v dalších krocích). Špatná kvalita získaného obrazu může mít nepříjemný dopad např. pro zařízení zpracovávající špionážní snímky, řízení automobilu atd.

### **Předzpracování digitálního obrazu**

Úpravou digitálního obrazu v podstatě rozumíme použití různých filtrů a opravných mechanismů k odstranění poruch obrazu (tj. různých šumů přidaných k obrazu před vstupem do kamery nebo během jeho digitalizace). Během tohoto procesu se nejen odstraňuje šum, ale také zlepšuje kvalita některých částí obrazu, které jsou podstatné pro další zpracování (např. vyostřování hran, potlačení či zvýraznění některých vlastností obrazu pomocí filtrace apod.) Odstranění šumu je poměrně důležitý krok pro získání kvalitní informace.

### **Segmentace obrazu**

Po ukončení procesu úpravy digitálního obrazu, nastupuje proces, který umožňuje rozpoznávat jednotlivé objekty typu auto, jablko, lidskou postavu, papilární otisk prstu atd. Tento proces se jmenuje *segmentace*. Při segmentaci se snažíme nalézt v obrazu nepřekrývající se oblasti odpovídající objektům (jde například o hledání černých písmen na bílém podkladě). Segmentaci rozdělujeme na úplnou, kdy se zcela povede přiřadit oblasti objektům a částečnou, kdy některé nalezené části neodpovídají objektům. V průběhu tohoto procesu sice dochází k ohraničení objektů, ale také k redukci dat. Nejjednodušší segmentací je tzv. *prahování*, při kterém se vychází ze znalosti odrazivosti povrchů jednotlivých těles a lze ho používat, pokud se hledané objekty odlišují výrazně svým jasnem od pozadí. Určuje se tzv. *jasová konstanta* a přes daný filtr se propustí jen ty části obrazu, které mají odpovídající nebo vyšší jas. Existuje mnoho dalších algoritmů segmentace, nicméně výsledkem každé z nich je zvýraznění objektu v daném obraze.

### Popis objektů digitálního obrazu

Dalším krokem je popis objektů v digitalizovaném obraze. To znamená, že objekty musí být popsány nějakým vhodným a jednoznačným způsobem, který by umožnil také jejich jednoznačnou klasifikaci. To lze udělat několik způsoby. Jedním je například tzv. *řetězový kód*. Jde o vyjádření pozic jednotlivých pixelů pomocí čísel, které odpovídají pozici daného pixelu, jasu, atd. Cílem každé popisné metody je provést pokud možno takový popis, aby stroj mohl rozpoznat, co je to za objekt a na základě toho mohl měnit své chování.

### Klasifikace objektů digitálního obrazu

Vlastní klasifikace objektů digitálního obrazu se provádí tzv. *klasifikátory*, které určují, do jakých tříd patří právě vyšetřovaný objekt. Takovýmto klasifikátorem mohou být různé algoritmy včetně neuronových sítí či fuzzy logiky. Ať už se použije kterýkoliv z nich, jde o zařazování objektů do příslušných tříd (například „auto“ do třídy „motorová vozidla“, „jablko“ do třídy „ovoce“ atd.). Obecně lze říci, že klasifikace objektů digitálního obrazu znamená interpretaci jeho dat, o kterých se předem nic nepředpokládalo.

## 11.2 Využití počítačového vidění

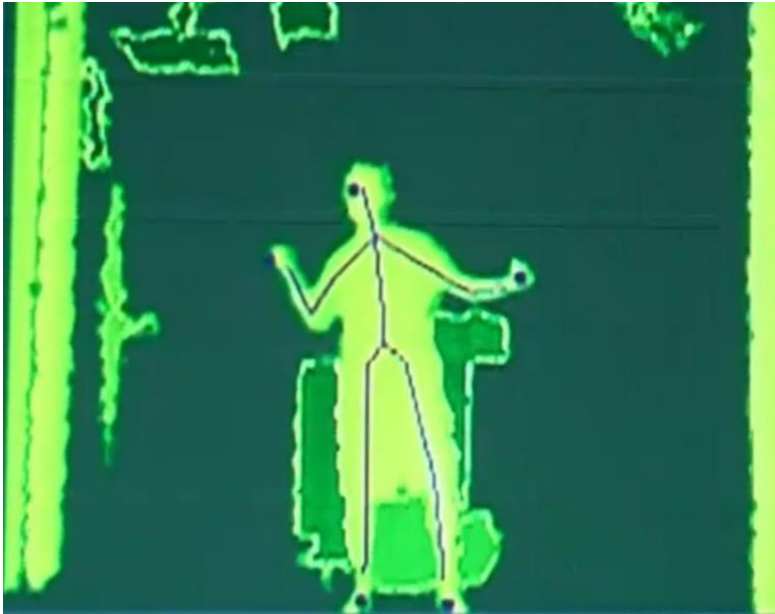
Mezi základní úlohy pro využití počítačového vidění patří:

- detekce událostí – jedná se například o vizuální kontrolu výrobků ve výrobním procesu,
- modelování objektů nebo prostředí – průmyslové inspekce, topografické modelování, lékařské obrazové analýzy,
- řízení procesů, systémů – průmyslové roboty, hydraulická ramena, samočinná vozidla a podobně,
- organizace událostí – indexace databází obrázků, obrázkových sekvencí.



Zajímavým použitím je využití počítačového vidění k interakci (komunikaci) mezi člověkem a strojem, počítačem. Pokud například k dnes již běžnému rozpoznávání hlasových příkazů připojíme také schopnost počítače porozumět gestikulaci a mimice člověka, tak získáme mocný nástroj, který je schopen porozumět také kontextu a významu celé konverzace.

Určitě důležitým využitím počítačového vidění je zábavní průmysl. Zábavnost interaktivních her dnes prezentuje například Microsoft s řešením Kinect, které dovoluje konzoli XBOX sledovat hráče a umožnit mu ovládání her na základě pohybů jeho těla. Na obrázku č. 11.1 je uveden příklad, jak v takovémto případě počítač (konzole XBOX® Kinect®) vidí - vyhodnocuje postavu, která je před snímač postavena.



Obrázek 11.1: Jak "vidí" postavu systém XBOX Kinect (zdroj zive.cz)

#### **Jednoduchý příklad** (Habiballa 2004)

Jednoduchá automatická analýza transparentního obrazu buněčného preparátu pozorovaného optickým mikroskopem. Cílem je spočítat buňky a rozřadit je podle tvaru buněčných jader na podlouhlé a ostatní. Úloha je z principu dvourozměrná. Předzpracováním se odstraní případný šum a pro segmentaci lze využít skutečnosti, že buněčná jádra jsou mnohem tmavší než zbytek. Pak už jen zbývá popsat body obrazu, příslušné každému buněčnému jádru, číselnými charakteristikami a podle nich buňky třídit.

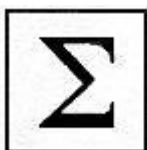


#### **Složitý systém - porozumění obrazu používané v autonomních vozidlech** (Habiballa 2004)

Cílem je vybavit navigací terénní vozidlo pohybující se v blíže neomezené krajině bez řidiče. Poměrně dokonalé experimentální verze již byly vytvořeny. Systém analýzy obrazu je ale jen jeden z mnoha senzorů, na jejichž základě se



řídící systém vozidla rozhoduje. Jinými zdroji informací jsou např. radar, laserový dálkoměr, informace o absolutní poloze vozidla odvozená z gyroskopu, digitální mapa. Scéna je trojrozměrná a navíc v čase proměnná. Obrazový systém obvykle pracuje s informací ze dvou kamer, aby pomocí techniky stereovidění mohl odhadovat hloubku objektů ve scéně od pozorovatele. Vozidlo si musí vytvářet a průběžně modifikovat vnitřní model okolního světa.



### Shrnutí kapitoly

Tato kapitola představuje stručný úvod počítačového vidění. Jako technická disciplína bývá počítačové vidění označováno za kombinaci počítače, snímacího zařízení a softwaru, který je schopný snímanou předlohu zpracovat nebo klasifikovat. Důležitou částí počítačového vidění jsou techniky zpracování 2D obrazů, jimž se věnujeme v úvodu této kapitoly. V závěru kapitoly jsou zmíněny oblasti využití počítačového vidění.



### Kontrolní otázky a úkoly:

4. Jaké jsou principy počítačového vidění?
5. Jaké jsou techniky zpracování 2D obrazů?
6. Jaké jsou oblasti použití počítačového vidění?



### Korespondenční úkoly

Vyberte si jednu oblast použití počítačového vidění a napište na toto téma pojednání v rozsahu alespoň 5 stran.



### Citovaná a doporučená literatura

Šonka, M. a Hlaváč, V. *Počítačové vidění*. Praha : Grada a.s., 1992. ISBN 08-85424-67-3.

Habiballa, H. *Umělá inteligence*. Texty pro distanční studium. Ostravská univerzita. Ostrava 2004.



## 12 OCR

### V této kapitole se dozvíte:

- jak pracují OCR programy,
- jaké jsou metody vektorizace a skeletonizace,
- jaké jsou typické oblasti použití OCR programů.

### Po jejím prostudování byste měli být schopni:

- vysvětlit princip OCR,
- charakterizovat preprocesing a postprocesing v OCR programech.

**Klíčová slova kapitoly:** OCR (Optical Character Recognition), scanner, vektorizace, skeletonizace.

#### Průvodce studiem

*OCR (z anglického Optical Character Recognition) je metoda, která pomocí scanneru umožňuje digitalizaci tištěných textů, s nimiž pak lze pracovat jako s normálním počítačovým textem. Počítačový program převádí obraz buď automaticky nebo se musí naučit rozpoznávat znaky. Převedený text je téměř vždy v závislosti na kvalitě předlohy třeba podrobit důkladné korektuře, protože OCR program nerozezná všechna písmena správně.*

*Na zvládnutí této kapitoly budete potřebovat asi 2 hodiny*



### 12.1 Optické rozpoznávání znaků

OCR neboli optické rozpoznávání znaků je technologie převodu textu uloženého v bitmapovém formátu do formátu textového. OCR je speciálním případem vektorizace (Optical Character Recognition), tedy rozpoznávání písma. Text uložený v bitmapě není chápán jako text, je to jen sada tmavých a světlých bodů v obrázku. OCR program tedy musí identifikovat v bitmapě různé tvary a porovnat je s předlohou a rozhodnout jaké písmenko, ten který shluk představuje. Situace je navíc zkomplikována tím, že texty bývají napsány

v různých fontech a dokumenty bývají často nekvalitní. Zvláště xeroxované dokumenty bývají „zašpiněné“, tzn. obsahují rozmazaná písmenka a šmouhy. Program se tedy musí snažit i určit zda tečka poblíž identifikovaného písmenka „c“ je háček a nebo jen nějaké smítko. Novější trasovací programy pracují tak, že dokument procházejí několikrát za sebou a při posledních průchodech už spolupracují se *spell-checkerem*. Mnohé programy se umí „učit“. Takže když chcete převést do textového formátu sadu dokumentů psaných na jednom psacím stroji, můžete OCR program naučit, že dotyčnému stroji ustřelovalo písmenka „z“ a „k“ bylo trochu rozmazané.

### Vektorizace



Vektorizací rozumíme převod dat z rastrového formátu do formátu vektorového. Jedná se o úlohu obtížnou, neboť informací uložených v rastrovém formátu je méně, než informací uložených ve formátu vektorovém, a tak je potřeba nové informace automaticky generovat, nebo je ručně do dat doplnit.

Při vektorizaci jsou používány tři základní metody: (Habiballa 2004):

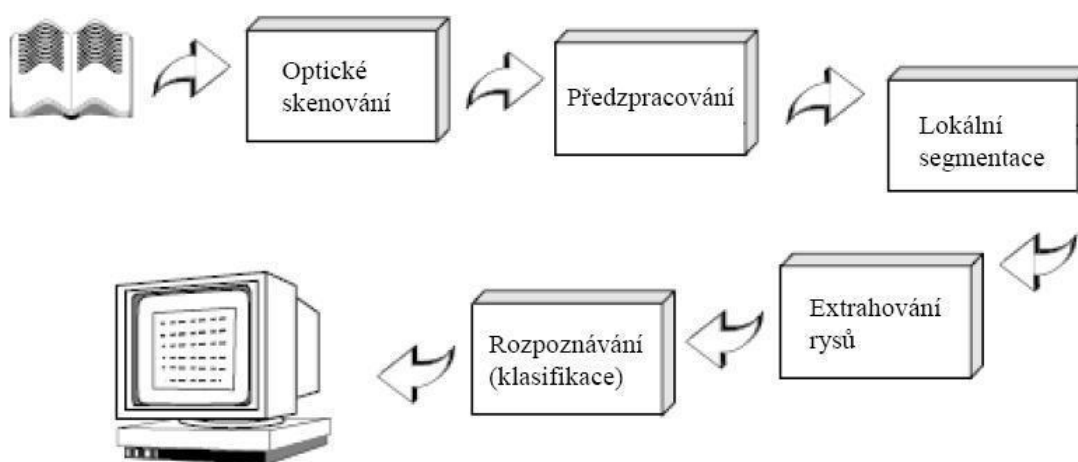
**Ruční** - obsluhovaná uživatelem. Na zobrazovacím zařízení (monitoru) se zobrazí rastrový obrázek a podle něj uživatel zadává pomocí vstupního digitalizačního zařízení (tablet nebo myš) jednotlivé vektorové entity. Ruční vektorizace je nejpřesnější, ale velmi zdoluhavá a náročná. Zvláštním příkladem ruční vektorizace může být přímá digitalizace papírové předlohy digitizérem bez použití mezistupně rastrového obrázku.

**Automatická** - obsluhovaná programem. Program musí být schopen na základě informací o barvě jednotlivých pixelů určit základní entity, ze kterých se obraz skládá. V mnoha případech se pro takový postup používají metody příbuzné umělé inteligenci. Automatickou vektorizaci lze v současné době použít jen u jednoduchých a zřetelných předloh.

**Poloautomatická** vektorizace je dnes nejčastější. Samotnou vektorizaci provádí program, který je v případě sporných situací korigován a opravován uživatelem. Kvalita vektorizace a její rychlost závisí na stupni automatizace.

## 12.2 Části OCR programu

V současné době se rozpoznávání ručně psaných znaků potýká s řadou problémů. Zejména je to odstraňování pozadí, korekce sklonu a velikosti písma, digitální způsob přemýšlení, který se liší od lidského. Některé z těchto problémů se podařilo do jisté míry odstranit tzv. preprocessing and postprocessingem, které se vykonávají před nebo po samotném OCR. Je rozumné oddělovat fáze „předprocesní“ a „poprocesní“ od algoritmů OCR a to z důvodu, že většina algoritmů OCR umí pracovat jen s černými znaky na bílém pozadí. Na obrázku 12.1 je uvedena typická činnost OCR programu.

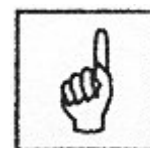


Obrázek 12.1: Části OCR programu (převzato z

[http://geo3.fsv.cvut.cz/vyuka/kapr/SP/2008\\_2009/vymetalek\\_viktora/index.html](http://geo3.fsv.cvut.cz/vyuka/kapr/SP/2008_2009/vymetalek_viktora/index.html))

### Preprocessing

Typickým příkladem algoritmů pro předzpracování v OCR je odstraňování vzorů na pozadí, vypreparování textu a srovnání šikmo psaného textu, korekce velikosti a sklonu znaků. Text je vlastně často psán na vzorovaném pozadí (papíře), například při posílání dopisů lidé často používají pěkně graficky zpracované obálky, proto by dobrý systém měl umět rozeznat text od okrasné grafiky. Lidé toto samozřejmě zvládnou díky přirozené intuici, ale podíváme-li se na tento problém z hlediska počítače, je to složité. Není výjimkou, že se celý blok textu rozpozná jako grafika a je tak vyloučen z rozpoznávací procedury. Jednou z možných metod je tzv. *thresholding*, jedná se o algoritmus na získání prahové úrovně šedi, kdy každému pixelu na pozadí se přiřadí "0" a pixelu na



popředí hodnota "1". Takto připravená data mohou přistoupit k dalšímu kroku v preprocesní fázi, kterým je segmentace a následně skeletonizace.

Algoritmus *segmentace* hledá na obrázku sloupcovité skrumáže bílých pixelů, které znamenají oddělení jednotlivých znaků.

Vypreparování textu (znaků), tzv. *skeletonizace*, je velice důležitou úlohou, jelikož takto získaná informace o tvaru znaků je základní vstupní informací pro algoritmus OCR. Skeletonizace může pomoci odstranit nepravidelnosti ve znacích, a tím zjednodušit celý rozpoznávací algoritmus, který pak uvažuje jen znakové úhozy, jež jsou pouhý pixel široké. Rapidně se taktéž sníží paměťové nároky na uchovávané informace o vstupních znacích. Toto je jedna z hojně užívaných metod protože mimo jiné, zkracuje čas samotného rozpoznávání. Kostry (skeletony) se získávají tzv. ztenčovacími metodami nebo transformacemi vzdáleností.

Zjednodušení algoritmu OCR můžeme dosáhnout také tím, že algoritmus bude považovat všechna vstupní data za normalizovaná na standardní rozložení (míněno, že text není "do kopce" ani "z kopce"), velikost a sklon. Existuje řada úspěšných algoritmů, které umožňují vytvořit horizontální rozložení, otáčet znaky a vytvářet konzistentní velikost. Algoritmy OCR pak mohou dostávat normalizované tvary znaků. Jiným příkladem normalizace je případ, kdy může být znak psán více způsoby.

### **Postprocessing**



Postprocesní zpravování dat je velice důležité, protože slouží k napravování chyb, kterých se případně dopustí algoritmus OCR. Nejtypičtějším případem postprocesního zpracování je kontrola pravopisu (spell checking), která automaticky opraví drobné chybičky a u větších chyb se zeptá uživatele na správný tvar. V mnoha případech se raději přímo přistupovalo k přepisování dokumentů než k použití OCR a následné manuální opravě, protože množství chyb, jichž se OCR systémy dopouštěly, bylo příliš velké. Musíme si uvědomit, že postprocesní systémy by neměly znehodnotit původně správně

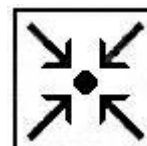
rozpoznaná data. Poprocesní systém opravující chyby OCR algoritmu, je možné napojit na jeho výstup.

### 12.3 OCR algoritmy

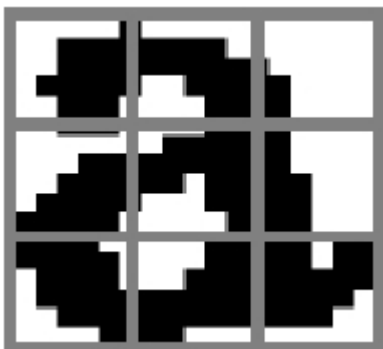
Nejběžněji používané techniky OCR jsou založeny na Markovových modelech, neuronových sítích, srovnávacích metodách tvarů a vzorů (stromové struktury vzorů jsou úspěšné při rozpoznávání netradičních fontů), polygonálních odhadech (vstupní data jsou převedena na polygonální reprezentaci), filtrech na extrakci tahů a bodů (on-line rozpoznávání). Markovovy modely zaznamenaly velký úspěch v rozpoznávání řeči, jsou to vlastně stavové stroje, které využívají kontextové informace. Počítače obecně nemají problémy s rozpoznáváním dobře napsaných znaků, které se moc neliší od daných vzorů, ale psané znaky jsou mnohdy víceznačné a nečitelné i pro člověka. Lidé jsou schopni číst slova s nečitelnými znaky, a tak by tomu mělo být i u počítačů. Algoritmy založené na principu rozpoznávání znaku po znaku by na takovém slově neuspěly, ukazuje se, že kontextová informace je nejen užitečná, ale často i nutná.

#### Extrakce příznaků

Klasifikace vzorů je nejproblematictější částí z celého OCR (Suchý 2007). Jejím úkolem je získání základních charakteristik každého znaku. Většina metod se snaží popsat znak přímo ze skenovaného obrázku, jiné zase získávají specifické rysy, které jednotlivé znaky charakterizují. Prvně jmenovaná metoda popsání znaku přímo ze skenovaného obrázku je založená na rozložení bodů v mřížce. Tato metoda má dva zástupce a jsou jimi: rozdělení do pásem a průsečíky.

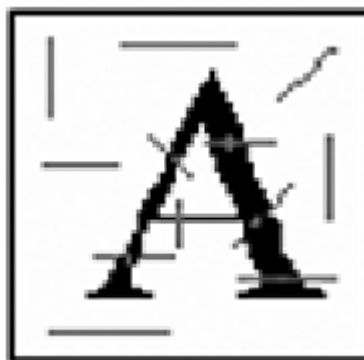


*Rozdělení do pásem.* Políčko s lokalizovaným znakem je rozděleno na několik oblastí a zkoumá se histogram tmavých míst v jednotlivých oblastech znaku, jak je vidět na obrázku 12.2.. Histogramy se pak porovnávají s rysy jednotlivých znaků, které vzejdou z tzv. trénovacích dat.



Obrázek 12.2: Rozdělení do pásem.

*Průsečíky.* Tato metoda je založena na počtu průsečíků předem zvolených vektorů v políčku se znakem. Názorně je to vidět na obrázku 12.3. Metoda rozpoznávající na základě specifických rysů je nazývána strukturální analýzou, kdy jsou jednotlivé znaky popisovány geometrickou a topologickou strukturou znaků. Tato metoda je však ještě předmětem aktivního výzkumu.



Obrázek 12.3: Průsečíky.

## 12.4 Praktické využití OCR



V zásadě existují dva velcí uživatelé systémů na rozpoznávání ručně psaných znaků. Jedná se o sektory *bankovních a poštovních* služeb. Finanční instituce využívají těchto systémů zejména v souvislosti se šeky, jelikož šek obsahuje značné množství položek, které je třeba řádně ověřit, aby byla zajištěna bezpečnost bankovních operací. Systém by například mohl kontrolovat, zdali souhlasí částka uvedená číslicemi a částka uvedená slovy, dále by mohl kontrolovat, jestli jméno plátce souhlasí s účtem, ze kterého má být částka odečtena, avšak první věcí, která by nás v souvislosti s šeky napadla, by bylo ověřování podpisu. Když vezmeme v úvahu množství šeků, které projde každý den bankovním systémem, a i kdyby takový systém rozpoznával pouze

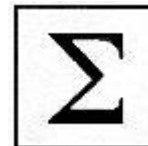
polovinu z nich, znamenal by značnou úsporu mravenčí práce. Francouzská pošta zavedla počátkem devadesátých let podobný systém, který měl mít 0,01% chybovost a umožňoval, aby až 50% šeků prošlo bez nutnosti předání k ručnímu zpracování.

Již mnoho poštovních úřadů zavedlo systémy OCR, které rozeznávají samostatné číslice, tím napomáhají k automatickému třídění zásilek na základě poštovních směrovacích čísel. Mnoho výzkumů potvrdilo obrovskou výhodu používání výlučně číslic v poštovních směrovacích číslech (USA, Česká republika atd.), jelikož číslice lze relativně nejsnadněji automatizovaně rozpoznávat. Systémy na poštách však mohou jít mnohem dál. Bylo by je možné používat k rozpoznávání celých adres a třídění obálek jen na jejich základě, tedy bez PSC.

Jinou oblastí, kde se využívá ručně psaného vstupu dat, je vyplňování formulářů, psaní si poznámek, korektura dokumentů a udržování záznamů. Celkem pěknou aplikací by se mohl stát systém, jež by přijímal data z digitálního pera, např. při přejímání dodávek zboží, kde by se pak data přenášela do centrálního počítače. To by ale vyžadovalo, aby se ve větší míře rozšířily elektronické podpisy. Když bychom se přenesly ze skladu do kanceláří, pak by nás určitě potěšil systém, jenž by automaticky četl faxové zprávy. S tím souvisí i podpora pro nevidomé, která jednoznačně ukazuje, že rozpoznávání ručně psaných znaků se nevyužívá výhradně pro komerční nebo humanitární aplikace, ale obě úrovně se navzájem překrývají.

## Shrnutí kapitoly

Tato kapitola představuje stručný úvod OCR neboli optického rozpoznávání znaků (z anglického Optical Character Recognition. OCR je metoda, která pomocí scanneru umožňuje digitalizaci tištěných textů, s nimiž pak lze pracovat jako s normálním počítačovým textem. Nejproblematičtější částí OCR programů je klasifikace vzorů, proto je jejich nezbytnou součástí preprocessing



a postprocessing zpracovávaných dat. V závěru kapitoly jsou zmíněny typické oblasti využití OCR.



### **Kontrolní otázky a úkoly:**

1. Jak pracují OCR programy?
2. Jaké znáte metody vektorizace a skeletonizace?
3. jaké jsou typické oblasti použití OCR programů?



### **Korespondenční úkoly**

1. Vyberte si jednu oblast použití OCR a napište na toto téma pojednání v rozsahu alespoň 5 stran.
2. Napište pojednání v rozsahu alespoň 5 stran na téma „OCR algoritmy“.



### **Citovaná a doporučená literatura**

Habiballa, H. *Umělá inteligence*. Texty pro distanční studium. Ostravská univerzita. Ostrava 2004.

Suchý, V. *Rozpoznávání textu v obraze*, Brno 2007.



## 13 Vybrané aplikace umělé inteligence

**V této kapitole se dozvíte:**

- Jaké jsou možné oblasti použití umělé inteligence.

**Po jejím prostudování byste měli být schopni:**

- Vysvětlit kde lze používat umělou inteligenci,
- Objasnit čím je významná počítačová hra Black & White,
- Uvést další možnosti využití umělé inteligence

**Klíčová slova kapitoly:** Boti, navigační síť,

### **Průvodce studiem**

*V této kapitole jsou zmíněny některé možnosti použití umělé inteligence v počítačových hrách a dalších oblastech informatiky. Samozřejmě, že existují i mnohé další oblasti použití umělé inteligence. Na zvládnutí této kapitoly budete potřebovat asi 2 hodiny.*

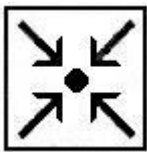


### **13.1 Umělá inteligence v počítačových hrách**

Herní umělá inteligence je v současné době jedním z nejbouřlivěji se rozvíjejících odvětví interaktivního zábavního průmyslu. Až do druhé poloviny 90. let byla umělá inteligence v počítačových hrách považována za okrajovou záležitost; herní vývojáři měli v tu dobu jinou prioritu – dokonalejší a realističtější grafiku. To se projevilo nejen ve velmi malém podílu na výkonu procesoru, který měla herní umělá inteligence k dispozici, ale také ve spěchu a nekoncepčnosti, se kterými byla do her na poslední chvíli implementována. Situace se začala měnit až s rozmachem grafických akceleračních na sklonku 90. let, jejichž nasazení výrazně odlehčilo hlavnímu procesoru, a uvolnilo tak výkon pro algoritmy umělé inteligence. Herní vývojáři si pomalu začali uvědomovat, že atraktivitu hry už nelze stavět pouze na realistické grafice, a zaměřili svou pozornost i na umělou inteligenci. Dnes v sobě herní umělá

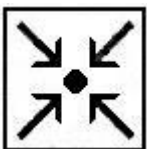
inteligence zahrnuje herní programování, herní design, ale také psychologii, drama a samozřejmě akademickou umělou inteligenci. Od té se ale výrazně odlišuje už samotným cílem – herní umělou inteligenci zajímá skutečná inteligence totiž jen do té míry, do které přispívá k tomu, aby hra hráče více (a případně déle) bavila. Proto se herní umělá inteligence někdy označuje také termínem zábavní inteligence (entertainment intelligence) - musí být realistická a nesmí obsahovat prvky, které by logice herního světa odporovaly.

### Navigační síť



Každý hráč 3D akčních her typu Quake se střetnul s botem, tedy inteligentním agentem nahrazujícím ve hře lidského protivníka. Ať už v roli protivníka nebo spoluhráče si dnešní boti vedou v boji velmi dobře, přestože zatím většinou nebývá problém odlišit je od živého hráče. Základním problémem, který musí každý bot řešit, je navigace a pohyb v herním prostředí. Bot si za tímto účelem musí nejen pamatovat pozici významných prvků v herní mapě (protivník, zbraň/střelivo, vlajka nepřítele apod.), ale musí být i kdykoliv schopen najít do vybraného místa pokud možno nejkratší cestu. O to se stará tzv. *pathfinding*, dnes v naprosté většině her založený na algoritmu heuristického prohledávání grafu. Pro jeho efektivní nasazení je potřeba k dané úrovni nejdříve zkonstruovat tzv. navigační síť, tedy graf, který popisuje odkud kam je možno se v dané úrovni dostat. Kromě informací týkající se pohybu může navigační síť obsahovat i speciální značky označující místa výhodná pro obranu či naopak pro číhanou na protivníka. V poslední době se proto objevuje snaha tvořit navigační síť automaticky pomocí nástrojů umělé inteligence, což však představuje nelehký úkol.

### Řídicí systém

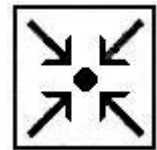


Zatímco navigaci lze považovat za podpůrný systém, skutečným mozkiem bota je jeho řídicí systém. Na základě aktuálního stavu bota, jeho záměru a situace v nejbližším okolí volí akce, které bot v příštích okamžicích provede. Tyto akce lze většinou rozdělit do několika úrovní – hovoříme proto o hierarchickém řídicím systému. Na nejvyšší úrovni jsou akce, které se váží k dlouhodobějším a strategičtějším rozhodnutím – bot si (ve hře určitého typu) může vybrat, zda bude prozkoumávat aktuální úroveň, doplňovat střelivo, útočit či naopak

ustupovat z boje. O úroveň níž bot rozhoduje např. o tom, kterým směrem se v mapě při svém průzkumu vydá, na jaký konkrétní cíl zaútočí a jakou přitom použije zbraň. Nejnižší úroveň se pak týká konkrétních atomických úkonů, tj. např. kdy vystřelit, kam zamířit, kudy uhnout střele protivníka apod. Zdaleka nejčastějším způsobem implementace řídicího systému je určitá obdoba tzv. hierarchického stavového automatu, pravidlové systémy či cílově orientované dynamické plánování.

### **Black & White**

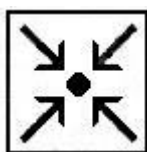
Jestliže s herními boty se hráč setká velmi často, umělá inteligence ve hře Black & White (Lionhead Studios, 2001) zůstává dodnes výjimečným počinem. Black&White je od základů postavena na schopnosti herních postav učit se, a je proto často považována z pohledu umělé inteligence za nejinnovativnější hru vůbec. Každému hráči je zde přidělen tzv. titán, prostřednictvím kterého hru hraje. Největším lákadlem Black&White je možnost titánovu osobnost zásadním způsobem formovat. Kromě přímých rozkazů hráče a jeho zpětné vazby (trest nebo odměna), se titán dokáže zdokonalovat i z pozorování ostatních postav herního světa a z vlastního průzkumu prostředí. Základem titánovy tvárnosti je jeho schopnost *učit se*. Vzhledem k různým způsobům učení v Black & White zvolili autoři i několik různých modelů *reprezentace znalostí*. Fakta o jednotlivých objektech jsou reprezentovány jednoduchým seznamem atributů. Znalost o tom, jak jsou které předměty vhodné k uspokojování titánových potřeb, je reprezentována pomocí rozhodovacích a regresních stromů. Závislost mezi stavem agenta a jeho přáními je pak reprezentována pomocí *umělé neuronové sítě*. Lze tak mít titána, který začne hledat kořist, až když má opravdu velký hlad, a naopak titána, který žere pořád, obzvlášť je-li v depresi. Zásadní je, že tyto reprezentace jsou vytvářeny a dále modifikovány dynamicky až během hry pomocí *algoritmů strojového učení*. Black & White lze považovat za ukázkou, jak může vhodné použití technik umělé inteligence vést nejen ke kvalitativnímu skoku v herní umělé inteligenci.



## 13.2 Umělá inteligence v dalších aplikacích

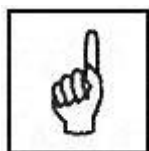
### Intelligentní vyhledávání

Určitě to znáte: když potřebujete na internetu něco najít, spustíte internetový vyhledávač Google, do příslušného políčka napíšete odpovídající výraz, chvíli počkáte a nakonec projdete výsledky vyhledávání. O co lepší by však bylo, kdyby za vás počítač udělal i již zmíněné prohledávání ve výsledcích hledání, a tak vám připravil skutečně pouze to, co vás zajímá.



### Vyhledávání na internetu pomocí hlasu

Přesně na výše zmíněném úkolu pracují vědci a studenti ve výzkumné oblasti vytváření inteligentního rozhraní pro uživatele v německém výzkumném centru pro umělou inteligenci (DFKI). Jejich projekt se jmenuje Smart Web (<http://www.smartweb-projekt.de>). Jeho cílem je vytvořit mobilního průvodce, který je bezdrátově připojen k internetu a který uživateli prostřednictvím hlasového nebo textového výstupu automaticky podává informace. Na otázku typu „Kdo byl William Shakespeare?“, kterou uživatel položí, přístroj odpoví (a to rovněž hlasem) „Významný anglický básník a dramatik“. Uživatel tak vůbec nemusí prohledávat žádné výsledky vyhledávání.



### Vyhledávání obrázků podle motivu

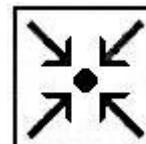
Program Imagesorter 2.0.1 třídí obrázky na pevném disku podle jejich podobnosti, v závislosti na obsahu. Nástroj je zdarma a najdete jej , popřípadě na internetové adrese <http://mmk.f4.fhtw-berlin.de> jako soubor IMAGESORTERV2\_XP.ZIP o velikosti 7,80 MB. Další funkcí programu je možnost označit obrázek a zobrazit všechny, co mu jsou podobné. Tímto způsobem se tedy dají velmi snadno najít kupříkladu všechny obrázky Eiffelovy věže, které máte na svém počítači. Utilita pracuje tak, že provádí analýzu obsahu obrázku. Prostřednictvím matematických algoritmů popisuje fotografie jako vektory, popřípadě body ve vícerozměrném vektorovém prostoru.

## Zpracování řeči

Mezi počítačové programy, které se pokouší napodobovat lidskou řeč, patří aplikace pro strojový překlad, rozpoznávání řeči a hlasový výstup. Čím více se v daném programu skrývá inteligence, tím lepší jsou logicky výsledky, které jsou jako výstup poskytovány uživateli.

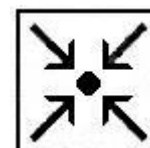
## Strojový překlad

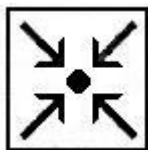
Jedním z těchto algoritmů je k patentování připravený neuronální přenos, jenž se uplatňuje v programu Linguatex Personal Translator 2008. Tento program pro překlad textů si můžete za 49 eur zakoupit na internetové adrese <http://www.linguatex.net>. Tato metoda dokáže podle kontextu správně přeložit slova s více významy. Správný význam slova rozpoznává program i na základě dalších vět. Pokud si sami chcete udělat obrázek o tom, jak výkonná je aplikace Personal Translator 2008 od firmy Linguatex a kde asi leží hranice překladu vytvořeného pomocí programu využívajícího umělou inteligenci, navštivte internetovou stránku [www.linguatex.net/onlineservices/pt](http://www.linguatex.net/onlineservices/pt). Zde můžete zadat pro ukázkový překlad až 500 znaků dlouhý text, který lze přeložit do několika jazyků.



## Rozpoznávání textu

Při využívání technologie pro rozpoznávání textu (OCR) jde víceméně o to, aby program převedl oskenovaný nebo ofocený dokument do podoby citovatelného textu. Zní to velmi jednoduše, ale praxe je daleko složitější. Pro počítač není oskenovaný list papíru nic jiného než obrázek obsahující černé, bílé či barevné body. Všechny tyto principy kupříkladu využívá program pro rozpoznávání textu s názvem Abbyy Finereader 9.0. Tento program najdete jako 15denní zkušební verzi, a to jako soubor FR90PE\_ESD.EXE o velikosti 126 MB, popřípadě si jej můžete stáhnout na internetové stránce <http://www.abbyy.com>. Program vyhledává v naskenovaném dokumentu prvky, které svou strukturou a uspořádáním vypadají jako text (uplatnění principu jednotnosti).





### **Rozpoznávání řeči**

Program Dragon Naturally Speaking ve své verzi 9 slibuje 99procentní úspěšnost při rozpoznání řeči, a to aniž by ho bylo nutné na váš hlas nějak trénovat. Mluvit přitom můžete tempem, jakým jste zvyklí. Cena verze Standard je včetně náhlavní soupravy stanovena na 99 eur. Podrobnější informace naleznete na internetové stránce programu na adrese <http://www.nuance.de>. Přesnost 99 procent znamená u typické stránky s 500 slovy 5 špatně zapsaných slov, nicméně program se ze svých chyb neustále učí, takže se přesnost rozpoznávání časem trvale zlepšuje. Program bohužel nepodporuje češtinu, ale pouze angličtinu a několik dalších světových jazyků.

### **Hlasový výstup**

Hlasový výstup (používá se i pojem syntéza hlasu) se stále více blíží lidské řeči. Jeho základ tvoří jednotlivými mluvčími namluvená slova či věty. Výrobci programů pro syntézu hlasu využívají textů namluvených a nahraných ve studiu jednotlivými mluvčími. Při vývoji programů se pak tento zvukový záznam rozdělí na části, s nimiž se dále pracuje.



### **Rozpoznávání obličeje**

Rozpoznávání obličeje se dá využít dvěma způsoby: buď pro identifikaci určité osoby na obrázku (obličej této osoby však musí být uložen v databázi), nebo pro interpretaci mimiky osob. Identifikace osoby podle obličeje řeší například program X-Login 1.0 vám umožní pracovat na počítači pouze tehdy, pokud před ním bude sedět určitá dříve zadaná osoba. Obraz se do programu snímá obyčejnou webovou kamerou. Programy tohoto druhu mají sklon pracovat spíše jednodušeji, neboť si pamatují pouze umístění osoby a vzdálenost partií obličeje. Ochrana počítače tímto softwarem se tedy dá svým způsobem obejít tak, že použijete fotografii oprávněné osoby, kterou umístíte před webovou kameru.

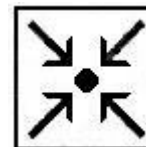
### **Rozpoznávání pocitů**

Jedná se o program Real Time Face Detector 4.05, který najdete jako 60denní demoverzi, popřípadě ji můžete stáhnout na internetové adrese

<http://www.iis.fraunhofer.de/bf/bv/kognitiv/biom/dd.jsp> jako soubor

RTFACEDETECT\_SETUP\_V405\_TCM97-78258.EXE o velikosti 7 MB.

I některé digitální fotoaparáty dokáží rozpoznat lidské obličej a zaměřit na ně objektiv. Forma Sony jde se svými modely DSC-T70 a DSC-T200 dokonce ještě dále. V jejich případě můžete nastavit, aby přístroj pořídil snímek až tehdy, když na tváři fotografované osoby objeví úsměv. 8 0006/TJi ?

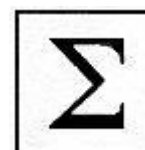


### Čtení myšlenek

Bude v nejbližší době nějaký program tak chytrý, že dokáže číst naše vlastní myšlenky? Tak docela nemožné to zase není – alespoň to naznačují aktuální výsledky dosažené ve výzkumu v této oblasti. Při vlastních testech se hlava testované osoby zasune do tomografu využívajícího magnetické rezonance. Toto zařízení využívá elektromagnetického vlnění k měření aktivity neuronů. Tak se zjistí, ve které části mozku mají v daný okamžik neurony největší aktivitu. Kvůli přesnějšímu určení dělí výzkumní pracovníci mozek do několika různých oblastí, pro které používají pojem voxel. Testované osobě pak ukazují několik obrázků stejného druhu, ale z několika oblastí – například nejprve obrázky koní a poté třeba obrázky ptáků. Výsledky ukazují, že při předložení obrázků z různých oblastí jsou vždy aktivní jiné voxely mozku. Po provedení důkladnějšího testování pak dokáže speciální software s vysokou přesností určit, zda testovaná osoba právě myslela na koně nebo na ptáka. Jiné myšlenky se (zatím) ještě zachytit nedají. O něco snazší to mají výzkumní pracovníci v případě, pokud mají určit, zda někdo mluví pravdu nebo lže. V obou případech jsou totiž aktivní různé části mozku. Nicméně doba, kdy budou detektory lži pracující na zmiňovaném principu uvedeny na trh nebo kdy se budou běžně využívat při soudních procesech, je ještě velmi vzdálená

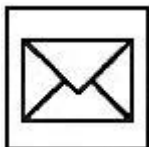
### Shrnutí kapitoly

Obsahem této kapitoly byly příklady možného umělé inteligence v počítačových hrách a dalších oblastech využití. Samozřejmě, že existují i mnohé další oblasti použití umělé inteligence.



**Kontrolní otázky a úkoly:**

Jaké jsou možnosti použití umělé inteligence?

**Korespondenční úkoly**

Nalezněte další možné oblasti použití umělé inteligence a napište na toto téma pojednání v rozsahu alespoň 5 stran.

**Citovaná a doporučená literatura**

Mařík, V. a kol.: Umělá inteligence 1,2,3. Academia, Praha, 1993, 1995, 2001