



evropský
sociální
fond v ČR



EVROPSKÁ UNIE



MINISTERSTVO ŠKOLSTVÍ,
MLÁDEŽE A TĚLOVÝCHOVY



OP Vzdělávání
pro konkurenceschopnost

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Expertní systémy

Cíl kapitoly:

Úkolem této kapitoly je pochopení významu expertních systémů, umět rozpoznat expertní systémy od klasicky naprogramovaných systémů a naučit se jejich tvorbu a základní vlastnosti. Pokračováním této kapitoly jsou programovací jazyky pro expertní systémy. Ty ovšem už nepatří do této skupiny kurzů.

Klíčové pojmy:

Abdukce, báze faktů, báze poznatků, dopředné řetězení, expertní systémy, heuristika, interferenční mechanismus, metaznalosti, predikátová logika, produkční pravidla, znalostní systémy, zpětné řetězení,

1. Úvod k expertním systémům

Znalostní systémy, mezi něž patří i expertní systémy, se liší od klasických systémů, kde činnost počítače je dána přesnou posloupností úkonů, algoritmů, nebo definicí objektů a metodami jejich zpracování. Znalostní systémy nemají přesně definované postupy, hlavním vodítkem je výsledek. Jsou představiteli tzv. evolučního programování – pokud problém je analogický s problémem úspěšně vyřešeným metodou M, pak i tento problém se začne řešit metodou M. Princip tkví v naučení rozpoznat, které zkušenosti lze použít a o nejvhodnější analogii.

Intelligence expertních systémů nespočívá v programu (programovém kódu), ale je určena znalostmi, závěry expertů, úspěšnými výpočty, které tvoří vstupní databáze systému. Program má za úkol vyhodnocovat stav, do kterého se v daném okamžiku systém dostal. Výsledkem může být hledaný závěr, rada, doporučení a nebo požadavek na rozšíření databáze znalostí, tedy pomoc experta.

2. Popis expertních systémů

2.1 Charakteristika

Expertní počítačový program má za úkol řešit složité úkoly tak, jako by je řešil tým expertů v daném oboru. Tedy se musí rozhodovat na základě znalostí vytvořených expertem s cílem dosáhnout výsledku odpovídajícího kvalitou výsledkům znalce. Z tohoto vyplývají charakteristické rysy expertních systémů

- Oddělení znalostníchází od vlastního vyhodnocovacího programu. (Toto je zásadní odlišnost od klasických programů.)
- Schopnost systému rozhodovat i v případech neurčitostí a nejasností (klasické programy neurčitosti nepodporují vyjma fuzzy neuronových sítí)
- Dokládat svoje výsledky podklady a vysvětlením.

2.2 Struktura expertního systému

Struktura expertních systémů se liší od klasických. U klasických systémů se zpracovává množina dat podle přesně stanoveného programu a tedy jednoznačně převádí vstupní data na výstupní. Má tedy dvě složky, která spolu souvisí.

Znalostní systémy a tedy i expertní systémy (rozdíl mezi nimi se postupně stírá) tvoří základní složky

- **Báze znalostí**, kterou dále dělíme na
 - **Báze faktů** – , tj všeobecně známé poznatky o dané oblasti. Tyto fakta lze dobře popsat a jednoznačně - **deterministicky** definují postup řešení určitého problému. Báze je uložena v nonvolatilních pamětech počítačových systémů. Odpovídá souboru vstupních dat klasického programu. Tato báze se vytváří a mění během řešení. V podstatě každé vyhodnocovací pravidlo **produkuje nový fakt a tím i nový počáteční stav** pro další krok řešení.
 - **Báze poznatků** – vědomosti - tj. vzájemně propojené (měnitelné, doplnitelné) struktury souvisejících faktů. Tato část se také nazývá **HEURISTIKA** (způsob hledání optimálního řešení) - tj postupy, metody řešení, pravidla sloužící k nalezení cíle. Tato část je právě vyjádřením **vědomostí a zkušeností experta**. Tyto znalosti jsou všeobecně **nedostupné**,

těžce sdělitelné a zdůvodnitelné. Heuristika má podstatně větší cenu než fakta. Heuristiky ale **nezaručují** nalezení **optimálního řešení**, někdy v případech, kdy nelze popsat dostatečně řešení, ho nenaleznou vůbec. Tato báze je uložena ve vnitřní paměti počítače. Může měnit i báti faktů. Mezi tyto znalosti patří i metaznalosti (znalosti o znalostech), logika, pravidla, rozhodovací stromy apod.

- **Interferenční mechanismus** – obsahuje myšlenkové, usuzovací, odvozovací algoritmy - vlastní program, který na základě znalostí modifikuje bázi faktů tak, aby byl schopen najít požadované výsledky. Interferenční mechanismus umožňuje
 - Odvozování nových poznatků
 - Prohledávání báze znalostí
 - Zpracování neurčitostí – nepřesností, nekompletností, nekonzistenci dat, vágní pojmy, nejisté znalosti
 - Dedukce
 - Indukce
 - Abdukce – ze správných závěrů hledá předpoklady, které k těmto závěrům vedly
 - Heuristiky – závěry na zkušenostech
 - Intuice
- **Vysvětlovací modul**
- **Modul pro získávání znalostí**
- **Modul pro vazby na jiné systémy**

2.3 Postup při výpočtu

ES (expertní systém) je **nedeterministická** posloupnost aktivování jednotlivých pravidel. To znamená, že

- není známo pořadí kroků postupu, nebo
- není možno jednoznačně určit výsledek podmínky.

Pokud interferenční mechanismus - IM nalezne **více** než jedno **pravidlo**, pak tato pravidla vytvoří **konfliktní množinu**. Tato množina je právě příčinou nedeterministického chování ES. Neexistuje jednoznačný návod, který by určoval, které pravidlo zvolit. Pokud ale

- zvolené pravidlo a tím i další cesta nevede k cíli,
- neexistuje pravidlo
- narazí na protichůdná tvrzení (je věcí expertů, aby stanovili hlavní a podružné rysy, a tím zabránili tomuto konfliktu).
- odhalí jednodušší postup

vrátí se IM na pomyslné **rozcestí (backtracking)**.

Řešení problému pomocí ES lze tedy vnímat jako **trajektorii průchodu stavovým prostorem** tvořeným množinou stavů produkovaných jednotlivými pravidly. Jednotlivé postupy nelze hodnotit podle žádné metriky, tj. ani podle přiblížení se k cíli nebo podle odchylky od správného směru řešení.

Během řešení systém vyhodnocuje splnění pravidel pomocí fuzzy logiky.

Řešení konfliktních množin

Při absenci analytických, deterministických, algoritmických postupů je nutné postupovat podle nedeterministických heuristických postupů. Těchto postupů v daném místě je několik. Nelze tyto množiny řešit zkusmo, to je vyzkoušet všechny možnosti. Proto postupy jsou následující

- **Dekompozice** velkých problémů na množinu malých. Tím se rozdělí i stavový prostor. Na malých prostorech lze aplikovat metody **slepého prohledávání**.
- **Hierarchizace** dělí prostor podle různých **úrovní podrobností**. Nejdříve hrubě, pak další dělení podle větších a větších detailů.
- **Nahrazení chybějící metriky** stavového prostoru. Metrikou označujeme veličiny, které definují vzdálenost cíle a směr zpracování. Proto tyto metody každému stavu **přirazují** určité **ohodnocení**, pomocí kterého lze nahradit chybějící údaje o „vzdálenosti“ cíle a „směru“ postupu nebo o **pravděpodobnosti** úspěšnosti.
- **Systém generuj a testuj**. Tento systém generuje cesty **ze všech pravidel** do určité **hloubky**. Pak výsledky **vyhodnotí** a určí další cestu.

Každé pravidlo neposkytuje všechny možné případy. Proto pravidlo platí s určitou **pravděpodobností**. Důsledek jednoho pravidla však podmiňuje použití dalšího neurčitého pravidla. Dochází tak k **šíření nejistoty** v průběhu inference celého problému. Lze dospět i k tomu, že identický závěr bude dosažen různými cestami s různou pravděpodobností.

3 Strategie

3.1 Usuzování

Při řešení úloh expertní systémy používají následující strategie

- **Dopředné řetězení** – forward chaining – jedná se o rozhodování řízení daty, tj. rozhoduje se na základě získaných dat. Příkladem je diagnostika, kdy hledáme nejvhodnější hypotézu pro cílové stavy se známých vstupních dat.
- **Zpětné řetězení** – backward chaining – jedná se o rozhodování podle požadovaného cíle, tj. vybere možný závěr a snaží se dokázat jeho platnost

vhodným výběrem vstupních dat. Zpětné řetězení se používá tam, kde je mnohem více vstupů než možných závěrů. Do této strategie patří plánování, kdy je znám cíl a hledáme vhodný počáteční stav.

3.2 Predikátová logika

Predikátová logika dovoluje odvozovat pravdivé formule z axiomů nebo již ověřených pravdivých formulí

3.3 Produkční pravidla

Základní produkční pravidlo zní: Jestliže platí předpoklady, pak tehdy a jen tehdy platí i důsledek. Produkční pravidla vyžadují správné nastavení počátečních parametrů. Dojde-li k rozporu nebo k chybě pravidel správný expertní systém tyto chyby najde a informuje o nich experty. Vzhledem k odděleným bázím od interferenčního mechanismu lze rychle vyhodnocovací mechanismus opravit a není třeba systém přeprogramovávat jak je to u klasického programování.

Shrnutí a doplnění:

Úlohy pro systémy umělé inteligence, mezi ní patří i znalostní a expertní systémy, mají různý charakter. Každá úloha je vhodná pro určitý typ systému. Proto

- Úlohy vyžadující přesný algoritmus řešení nejsou pro umělou inteligenci vhodné.
- Vývojové (evoluční) systémy – druh systému umělé inteligence - zpracovávají úlohy, kde je velké množství alternativ a neexistují výběrová pravidla pro nejvhodnější. Tyto systémy poskytují ohodnocení jednotlivých alternativ. Využití u optimalizačních systémů.
- Učící se systémy postavené na neuronových sítích využívají vysoký paralelismus u zpracování dat. Jsou vhodné pro předvídání, předpovědi počasí, řízení procesů apod.
- Expertní systémy používáme tam, kde je zapotřebí se rozhodovat a nalézt řešení a kdy neznáme přesná pravidla nebo vstupní data jsou neúplná.

Použití expertních systémů má smysl jen tehdy, pokud náklady na expertní systém nepřevýší využití úspory.

Výhody a nevýhody ES

Mezi výhody patří schopnost řešit složité problémy, nepřesně zadané problémy, dostupnost expertíz a jejich opakované využití, uchovávání znalostí na delší dobu.

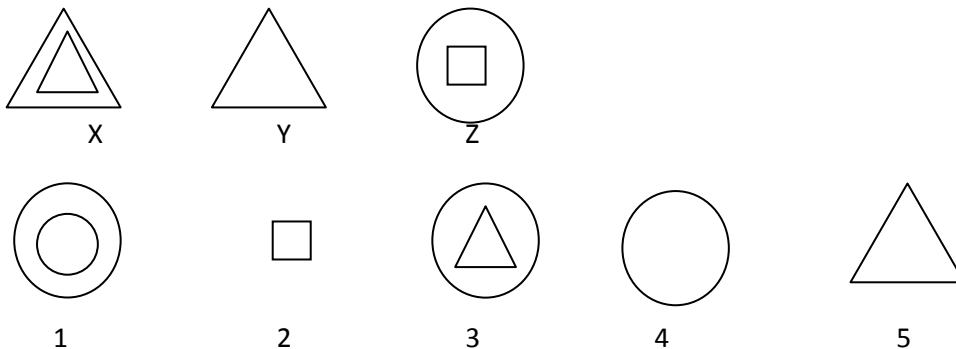
Jako nevýhodu považujeme možnost nesprávné interpretace ve změněných podmínkách a poznat meze své použitelnosti.

Příklad postupu expertního systému

Na následujícím příkladu si uvedeme logický postup při řešení logické úlohy.

Příklad:

Z obrázků 1 až 5 vyberte ten, který má stejný vztah k obrázci Z jako X k Y.



Popis obrázků:

X: UVNITŘ (malý ∇ , velký ∇)

Y: UVNITŘ (\emptyset , velký ∇)

Z: UVNITŘ (malý \square , velká O)

1: UVNITŘ (malá O, velká O)

2: UVNITŘ (malý \square , \emptyset)

3: UVNITŘ (malý ∇ , velká O)

4: UVNITŘ (\emptyset , velká O)

5: UVNITŘ (\emptyset , velký ∇)

Dále je nutné definovat množinu operací, pro porovnávání

O - odstranit

P - ponechat

N - nahradit

a označit objekty

v - vnitřní objekt

w - vnější objekt

Porovnávání

X-Y: $O(v) \cap P(w)$

Z-1: $N(v_i, v_k) \cap P(w)$

Z-2: $P(v) \cap O(w)$

Z-3: $N(v_i, v_k) \cap P(w)$

Z-4: $O(v_i) \cap P(w)$

Z-5: $O(v) \cap N(w_i, w_k)$

Porovnáním výsledků dostaneme výslednou relaci Z-4.

Kontrolní otázky:

- Co je to expertní systém?
- Proč pro expertní systémy nejsou vhodné úlohy z finanční sféry?
- Čím je dána inteligence expertních systémů?
- Jaká je role experta u expertních systémů?
- Příklad: rébus

JANO
_____ IVAN
KAROL

Najděte pravidla pro řešení uvedeného rébusu

- Vyjmenujte 3 charakteristické rysy expertních systémů.
- Jaký je rozdíl mezi bází poznatků a bází faktů?
- Co je to interferenční mechanismus?
- Vysvětlete pojmy dedukce, indukce, abdukce.
- Vysvětli rozdíl mezi dopředným a zpětným řetězením.

Literatura:

Cs. [Wikipedia.org/wiki/expertní systémy](https://cs.wikipedia.org/wiki/expertní_systémy)

Popper: Expertné systémy

[Programujte.com/clanek/2008020100-expertni-systemy-i](https://programujte.com/clanek/2008020100-expertni-systemy-i)

Jiří Dvořák: Expertní systémy

Kostal: sylaby pro SŠ