

Základy umělé inteligence

(26 hod. přednášky, 13 hod. projekty)

Hodnocení předmětu: 20/20/60 (test/projekty/zkouška)

Zápočet: minimálně 15 bodů z projektů a testu

Zkouška: minimálně 25 bodů

Přednášky:

1. Úvod, definice umělé inteligence (UI), typy UI úloh, přehled metod řešení těchto úloh.
2. Metody řešení úloh prohledáváním stavového prostoru.
3. Metody řešení úloh rozkladem na dílčí úlohy.
4. Metody řešení optimalizačních úloh algoritmy inspirovanými přírodou.
5. Základní metody hraní her.
6. Logika a UI, rezoluční metoda a její využití při řešení úloh a plánování.
7. Jazyk PROLOG a jeho použití v UI.

8. Strojové učení.
9. Klasifikace a rozpoznávání.
10. Principy expertních systémů.
11. Principy počítačového vidění.
12. Principy zpracování přirozeného jazyka.
13. Úvod do agentních systémů.

- 1. Úvod, definice umělé inteligence (UI),
typy UI úloh, přehled metod řešení
těchto úloh.**

Intelligence

Intelligence je slovo odvozené z latinského slova intelligentia, které znamená schopnost rozlišovat, racionálně poznávat a chápat.

Intelligence byla dlouhou dobu přisuzována pouze lidem, nicméně v poslední době je studována také u zvířat a dokonce i u některých rostlin.

Pojmem umělá intelligence (UI), resp. Artificial Intelligence (AI) je pak označována intelligence neživých subjektů - výpočetních systémů a příslušných programů.

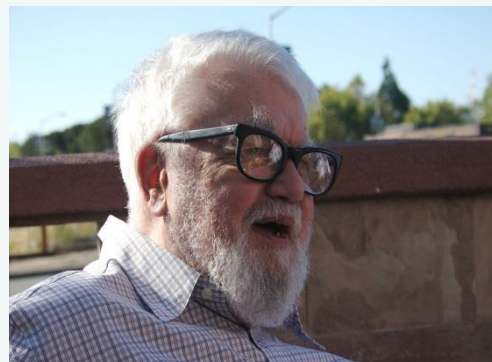
Stručný pohled do historie

- 1943 Model neuronu (W. McCulloch, W. Pitts)
- 1949 Učení neuronu (D. Hebb)
- 1950 Turingův test (A. Turing)
- 1951 Neuronová síť (M. Minsky, D. Edmonds)
- 1955 Program pro hraní dámy (A. Samuel)
- A jak jsme na tom byli v té době my (v ČSR) ?
- 1956:

The Dartmouth Summer Research Project on **Artificial Intelligence**

Autor pojmu AI:

John McCarthy (o padesát let později):



1956 – 1968 (Počáteční elán, velká očekávání)

- 1956 Logic Theorist (A. Newel, H. A. Simon, J. C. Shaw)
- 1958 LISP (List Processor, J. McCarthy)
- 1960 GPS (General Problem Solver, A. Newell)
- 1961 Program pro symbolickou integraci (J. Slagle)
- 1962 Šachový program (A. Newell, J. C. Shaw, H. A. Simon)
- 1965 Rezoluční metoda (J. A. Robinson)
- 1966 ELIZA (J. Weizenbaum)
- 1968 Algoritmus A* (P. E. Hart, N. J. Nilsson, B. Raphael)

1968 – 1973 (Návrat k realitě, skepse, první „zimní“ období UI)

- Velmi omezená výpočetní síla tehdejších počítačů
- Velké problémy s kombinatorickou explozí (exponenciální časová nebo/i paměťová náročnost)
- Problémy s monotónností tradičních logik
- Problémy s řešením problémů bez znalostí okolního světa, resp. kontextu. Například zcela selhaly pokusy o strojové překlady založené pouze na slovnících:

The spirit is willing but the flesh is weak

The vodka is good but the meat is rotten

- Na rozdíl od předpokladů se ukázalo řešení high-level problémů (například hraní her) jednodušší, než řešení low-level problémů (především senzomotorických)

1971 – 1981 (Obnovení zájmu o UI, znalostní systémy)

- 1971 STRIPS (R. Fikes, N. J. Nilsson)
- 1971 DENDRAL (E. A. Feigenbaum)
- 1972 PROLOG (PROgramming in LOGic, A. Colmerauer)
- 1976 MYCIN (E. H. Shortliffe)
- 1978 PROSPECTOR (R. Duda, P. Hart)
- 1980 HEARSAY II (L. D. Erman)
- 1981 Japonský projekt počítačů páté generace založený na jazyku PROLOG (Fifth Generation, 1981 – 1991)

1981- 2000 (První větší úspěchy UI)

- 1982 R1, expertní systém pro konfiguraci počítačů firmy DEC údajně přinášel této firmě úsporu 40 milionů dolarů ročně
- 1986 Renesance neuronových sítí, metoda backpropagation (D. Rumelhart, G. Hinton, R. Williams)
- 1988 Počítač Hitech (Carnegie-Mellon University, H. J. Berliner) porazil počítač šachového velmistra (A. Denkera)
- 1990 Pravděpodobnostní uvažování, Bayesovské sítě (J. Pearl)
- 1991 Počátek výzkumu inteligentních agentů
- 1992 Posilované učení, program TD-gammon (G. Tesauro)
- 1997 RoboCup – 1. ročník mezinárodních soutěží/turnajů autonomních robotů v kopané, pokračuje doposud
- 1997 Počítač DeepBlue (IBM, F. Hsu) porazil v sérii šesti zápasů poměrem 4:2 světového šachového šampiona G. Kasparova

1987- 1993 (Krátké druhé „zimní“ období UI)

- Neúspěch japonského projekt počítačů páté generace
- Nenaplnila se očekávání přínosů expertních systémů
- V roce 1987 zapříčinil prudký vývoj stolních počítačů Apple a IBM náhlé zhroucení trhu s UI hardware, především s tzv. LISPovskými počítači – uvádí se, že celý tento průmysl v hodnotě půl miliardy dolarů byl zničen prakticky přes noc

2000 - doposud (UI se stává součástí běžného života)

- 2000 Počátek vývoje komerčních chytrých zařízení: telefonů, praček, fotoaparátů, navigací, hodinek, apod.
- 2006 Počátek vývoje komerčních servisních robotů (pomocníků a společníků v domácnostech)
- 2009 První samostatná jízda autonomního auta (Google) po dálnici
- 2011 IBM Watson (systém dotaz-odpověď) zvítězil v americké vědomostní televizní soutěži Jeopardy
- 2015 Boom hlubokého učení (klasifikace obrazů)
- 2016 Počítač (AlphaGo, Google) zvítězil poměrem 4:1 v sérii pěti zápasů nad osmnáctinásobným světovým mistrem Lee-Se-dolem ve hře Go

Umělá intelligence – současný stav

- Klasická umělá intelligence (řešení úloh, učení, klasifikace, rozpoznávání, aplikační oblasti: Expertní systémy, počítačové vidění, zpracování přirozeného jazyka)
- Distribuovaná umělá intelligence:
 - Multiagentní systémy
 - Intelligence davu/roje/hejna (Swarm Intelligence)
- Soft computing:
 - Neuronové sítě
 - Genetické algoritmy
 - Fuzzy množiny a fuzzy logika
 - Pravděpodobnostní usuzování
 - Hrubé (Rough) množiny
 - Chaos

Klasická umělá inteligence

řešení úloh, učení, klasifikace, rozpoznávání

Stavový prostor a definice úlohy

Stavový prostor

$$(S, O)$$

Množina stavů úlohy

$$S = \{s_i\}, \quad i = 1, 2, \dots$$

Množina operátorů

$$O = \{o_j\}, \quad j = 1, 2, \dots$$

Úloha

$$(s_0, G)$$

Počáteční stav

$$s_0 \in S$$

Množina cílových stavů

$$G = \{s_{g1}, s_{g2}, \dots\}, \quad G \subset S$$

Řešení úlohy

$$s_1 = o_1(s_0),$$

$$s_2 = o_2(s_1),$$

$$\dots,$$

$$s_n = o_n(s_{n-1})$$

$$s_n \in G$$

Stavový prostor a definice úlohy

Aplikace operátorů na jednotlivé stavy je prakticky vždy omezena nějakými podmínkami, které jsou závislé na konkrétní úloze.

V některých úlohách není posloupnost operátorů vedoucí k nalezení „cesty“ z počátečního do cílového stavu podstatná, záleží pouze na nalezení cílového stavu při splnění předem daných omezujících podmínek. Takové úlohy se nazývají CSPs (Constraint Satisfaction Problems). Stavy v CSPs jsou obvykle definovány množinou proměnných, kterým se přiřazují hodnoty z množin přípustných hodnot pro tyto proměnné.

Typy UI úloh

1. Umělé úlohy (hříčky)

- Úloha dvou džbánů
- Úloha Loydovy osmičky/patnáctky
- Úloha osmi dam (CSP)
- Úloha hanojských věží
- Úloha balančních vah

a řada dalších (například Rubikova kostka, Sudoku (CSP), kryptoaritmetické úlohy (CSPs), apod.)

Typy UI úloh

2. Reálné úlohy/problémy

- Problém nalezení optimální cesty
- Problém obchodního cestujícího
- Problém barvení map (CSP)

a řada dalších (například problémy rozmístování prvků složitých obvodů, problémy navigace robotů apod.)

Úloha dvou džbánů (4 litry, 3 litry)

Stav $s = (\text{obsah většího džbánu}, \text{obsah menšího džbánu})$.

Klasická/původní úloha je dána takto:

$$s_0 = (0, 0)$$

$$G = \{s_{g1}, s_{g2}\} = \{(2, 0), (0, 2)\}$$

$$O = \{\rightarrow V, \rightarrow M, V \rightarrow, M \rightarrow, M \rightarrow V, V \rightarrow M\}$$

Operátory postupně značí: Naplnění většího džbánu, naplnění menšího džbánu, vyprázdnění většího džbánu, vyprázdnění menšího džbánu, přelití (části) obsahu menšího do většího džbánu (menší džbán je pak buď prázdný, nebo/a větší džbán je plný), přelití (části) obsahu většího do menšího džbánu (větší džbán je pak buď prázdný, nebo/a menší džbán je plný).

Úloha Loydovy osmičky

Stav $s =$ (matice s čísly kamenů na jednotlivých políčkách).

Úloha může být zadána různě, například takto:

$$s_0 = \begin{bmatrix} 1 & 4 & 2 \\ 8 & & 3 \\ 7 & 6 & 5 \end{bmatrix}$$

$$G = \{s_g\} = \begin{bmatrix} 1 & 2 & 3 \\ 8 & & 4 \\ 7 & 6 & 5 \end{bmatrix}$$

Operátory:

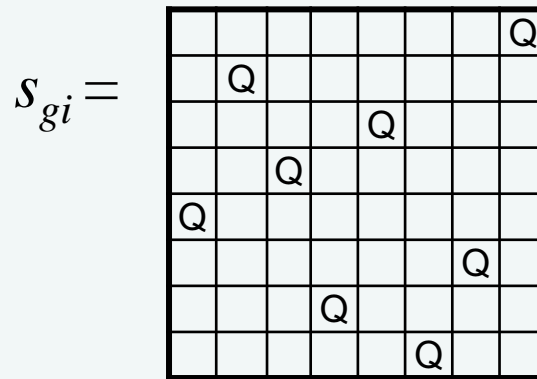
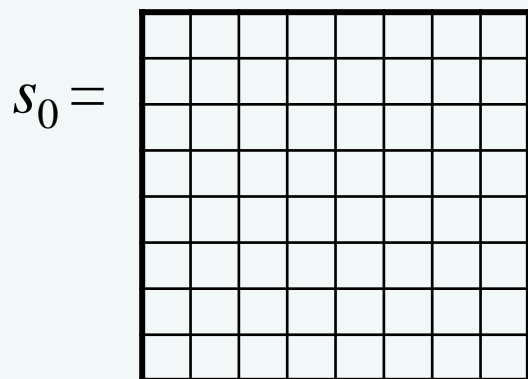
- Pohybuje-li se jednotlivými kameny, pak $O = \{o_1, o_2, \dots, o_8\}$, kde $o_i = \{\uparrow, \rightarrow, \downarrow, \leftarrow\}$, tj. tah příslušným kamenem nahoru, doprava, dolů a doleva.
- Pohybuje-li se prázdným políčkem, pak $O = \{\uparrow, \rightarrow, \downarrow, \leftarrow\}$.

Pozn.: Operátory z obou uvedených množin jsou ekvivalentní (například pohyb kamene č. 4 z počátečního stavu dolů odpovídá pohybu prázdného políčka nahoru).

Úloha osmi dam (CSP)

Stav s = seznam pozic dam (čísel řádků) v jednotlivých sloupcích, tj.
 $s = (Q_1, Q_2, Q_3, Q_4, Q_5, Q_6, Q_7, Q_8)$, $Q_j \in \langle 1, 8 \rangle$.

Úloha je zadána takto:

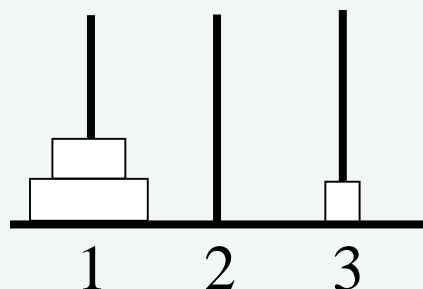


G je seznam cílových stavů, ve kterých žádná dáma neohrožuje jinou dámu, $s_{gi} \in G$.

Operátory o_j , $j = 1 \dots 8$, vedou k postavení dámy Q_j na libovolnou pozici ve sloupci j .

Úloha hanojských věží

Stav je seznam pozic kotoučů (nejlépe od největšího k nejmenšímu), pak například seznam $s = (1,1,3)$ značí stav



Úloha je dána takto: $s_0 = (1,1,1)$, $G = \{s_g\}$, $s_g = (2,2,2)$.

Operátor o_{ij} ($i, j = 1, 2, 3$, $i \neq j$) provádí přesun kotoučku z j -té tyčky na i -tou tyčku, s respektováním podmínky, že nikdy nesmí ležet větší kotouček na menším kotoučku.

Úloha balančních vah

Úlohou je nalézt minci s mírně odlišnou vahou od ostatních mincí pomocí vážení na balančních vahách. V klasické úloze je mincí celkem 12.

Stav $s = (M, M_S, M_{SL}, M_{ST})$, kde značí

- M množinu mincí s dosud nerozhodnutou váhou,
- M_S množinu mincí se stejnými vahami,
- M_{SL} množinu mincí, které jsou buď všechny stejné, nebo jedna z nich je lehčí
- M_{ST} množinu mincí, které jsou buď všechny stejné, nebo jedna z nich je těžší

Úloha balančních vah - pokračování

$$s_0 = (\{m_1, m_2, \dots, m_{12}\}, \{\}, \{\}, \{\})$$

$$s_{g1} = (\{\}, \{m_1, m_2, \dots, m_{12}\} - \{m_i\}, \{m_i\}, \{\})$$

$$s_{g2} = (\{\}, \{m_1, m_2, \dots, m_{12}\} - \{m_i\}, \{\}, \{m_i\})$$

Operátory provedou přesuny mincí mezi příslušnými množinami jednotlivých stavů na základě výsledků vážení (v miskách na obou stranách balanční váhy musí být samozřejmě stejné počty mincí, tj. 1 - 1, 2 - 2, ..., 6 - 6):

- Pokud klesne levá strana balanční váhy, pak se přesunou všechny mince z levé misky, které dosud byly v množině M , do množiny M_{ST} , všechny mince z pravé misky, které dosud byly v množině M , do množiny M_{SL} a všechny zbývající mince z množiny M do množiny M_S .

Úloha balančních vah - pokračování

- Pokud klesne pravá strana balanční váhy, pak se přesunou všechny mince z pravé misky, které dosud byly v množině M , do množiny M_{ST} , všechny mince z levé misky, které dosud byly v množině M , do množiny M_{SL} a všechny zbývající mince z množiny M do množiny M_S .
- Pokud zůstane váha ve vodorovné poloze, pak se přesunou všechny mince z obou misek do množiny M_S .

Problém nalezení optimální cesty

Úkolem je nalézt optimální cestu z počátečního do cílového stavu/místa za předpokladu, že přechodům mezi jednotlivými stavy úlohy je přiřazena cena. Touto cenou může být vzdálenost, kvalita přechodu, rychlost průjezdu, apod.

Problém nalezení optimální cesty je NP-úplný problém, protože pro n -míst teoreticky existuje $(n - 1)!$ různých možných řešení.

Problém obchodního cestujícího (TSP – Travelling Salesman Problem)

Úkolem je nalézt optimální cestu z počátečního místa přes všechna jiná místa zpět do počátečního místa s tím, že každé jiné místo se může navštívit pouze jednou. Přechodům mezi jednotlivými místy je přiřazena cena (vzdálenost).

TSP je také NP-úplný problém, pro n -míst zřejmě existuje

$$(n - 1)!/2$$

různých možných řešení.

Problém barvení map (CSP)

Úkolem je nalézt obarvení jednotlivých ploch mapy barvami z dané množiny barev takové, aby sousední plochy měly jiné barvy.

Metody řešení úloh

- Metody založené na prohledávání stavového prostoru (*State Space Search Methods*)
- Metody založené na rozkladu úloh na dílčí úlohy (*Problem Reduction Methods*)
- Metody řešení optimalizačních úloh algoritmy inspirovanými přírodou (*Genetic Algorithms, Ant Colony Optimization, Particle Swarm Optimization, etc.*)