Mesterséges intelligencia

Gregorics Tibor
people.inf.elte.hu/gt/mi

Szakirodalom

- Könyvek
  - Russel, J. S., Norvig, P.: Artificial Intelligence: A Modern Approach (3rd Edition), Prentice Hall, 2009. ISBN 0-13-604259-7
  - Fekete István - Gregorics Tibor - Nagy Sára: Bevezetés a mesterséges intelligenciába, LSI Kiadó, Budapest, 1990, 1999. ELTE-Eötvös Kiadó, Budapest, 2006.
  - Futó Iván (szerk): Mesterséges intelligencia, Aula Kiadó, Budapest, 1999.
- Internet
  - people.inf.elte.hu/gt/mi

Bevezetés

1. AZ MI FOGALMA

mesterséges intelligencia – MI (artificial intelligence - AI)

sokan sokfélét értenek alatta

Nem egy rész-területe az informatikának, hanem egy szemléletmód, amely az informatika fejlődését szolgálja: olyan problémákra keres számítógépes megoldásokat, amelyek megoldásában az ember jobbnak tűnik.

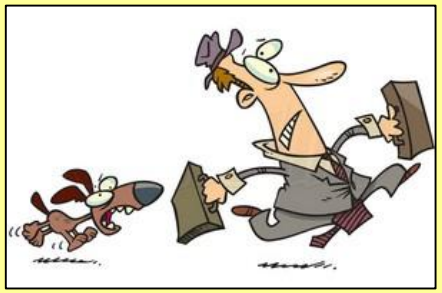
Erős MI
Cél: az emberi gondolkodás számítógéppel történő reprodukálása.

MI szkeptikusok
A számítógép soha nem lesz okosabb az embernél.

Gyenge MI
Cél: Azon elméletek és módszerek kutatása, fejlesztése, rendszerezése, amelyekkel az emberi intelligencia számára is érdekes és nehéz problémákra adhatunk számítógépes megoldásokat.

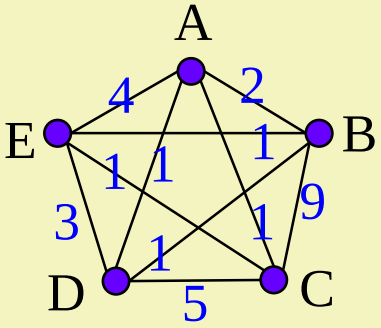
Miről ismerhető fel egy szoftverben az MI?

- Megoldandó feladat: nehéz
- Szoftver viselkedése
- Felhasznált technológiák



Utazó ügynök problémája

Adott n város a közöttük vezető utak költségeivel. Melyik a legolcsóbb olyan útvonal, amely az A városból indulva mindegyik várost egyszer érintve visszatér az A városba?



n	(n-1)!
5	24
50	$6 \cdot 10^{62}$

lehetséges utak:

ABCDEA

ACBDEA ABDCEA ABCEDA

ABDECA

ADBECA …

problématér

Miről ismerhető fel egy szoftverben az MI?

- Megoldandó feladat: nehéz
  - A feladat problématere hatalmas,
  - szisztematikus keresés helyett intuícióra, kreativitásra (azaz heurisztikára) van szükségünk ahhoz, hogy elkerüljük a kombinatorikus robbanást.
- Szoftver viselkedése
- Felhasznált technológiák

7-11

$x=?,\ y=?,\ z=?,\ t=?$

$$\begin{cases} x+y+z+t=7.11 \\ x\cdot y\cdot z\cdot t=7.11 \end{cases}$$

$$x, y, z, t \in \{1, \dots, 708\}$$

$$\begin{cases} x+y+z+t=711 \\ x\cdot y\cdot z\cdot t=711\cdot 10^6=2^6\cdot 3^2\cdot 5^6\cdot 79 \end{cases}$$

$x=79$ $\quad x=2\cdot 79$ $\quad \dots$ ~~$x=7\cdot 79$~~ $\quad x=8\cdot 79$ $\quad$ ~~$x=9\cdot 79=711$~~

$$\begin{cases} y+z+t=632 \\ y\cdot z\cdot t=2^6\cdot 3^2\cdot 5^6 \end{cases} \qquad \begin{cases} y+z+t=553 \\ y\cdot z\cdot t=2^5\cdot 3^2\cdot 5^6 \end{cases} \qquad \dots \qquad \begin{cases} y+z+t=79 \\ y\cdot z\cdot t=2^3\cdot 3^2\cdot 5^6 \end{cases}$$

Miről ismerhető fel egy szoftverben az MI?

- Megoldandó feladat: nehéz
  - A feladat problématere hatalmas,
  - szisztematikus keresés helyett intuícióra, kreativitásra (azaz heurisztikára) van szükségünk ahhoz, hogy elkerüljük a kombinatorikus robbanást.
- Szoftver viselkedése: intelligens
  - Turing teszt
- Felhasznált technológiák

ELIZA

Minta-válasz szabályok:

Úgy érzem, hogy ön mostanában engem un.

<a> *ön* <b> *engem* <c>.

1. Miért gondolja, hogy ön <a> én <b> <c>?
2. Tegyük fel, hogy én <b> önt <c>. Mit változtat ez a dolgokon?

Szóismétlés szabály:

„Miért ismételgeti ugyanazt újra és újra?”

Folytatási szabályok:

Igen, értem. Kérem folytassa. Ez nagyon érdekes.
Még miről szeretne beszélgetni?

Miről ismerhető fel egy szoftverben az MI?

Intelligens szoftver jellemzői
- megszerzett ismeret tárolása
- automatikus következtetés
- tanulás
- term. nyelvű kommunikáció

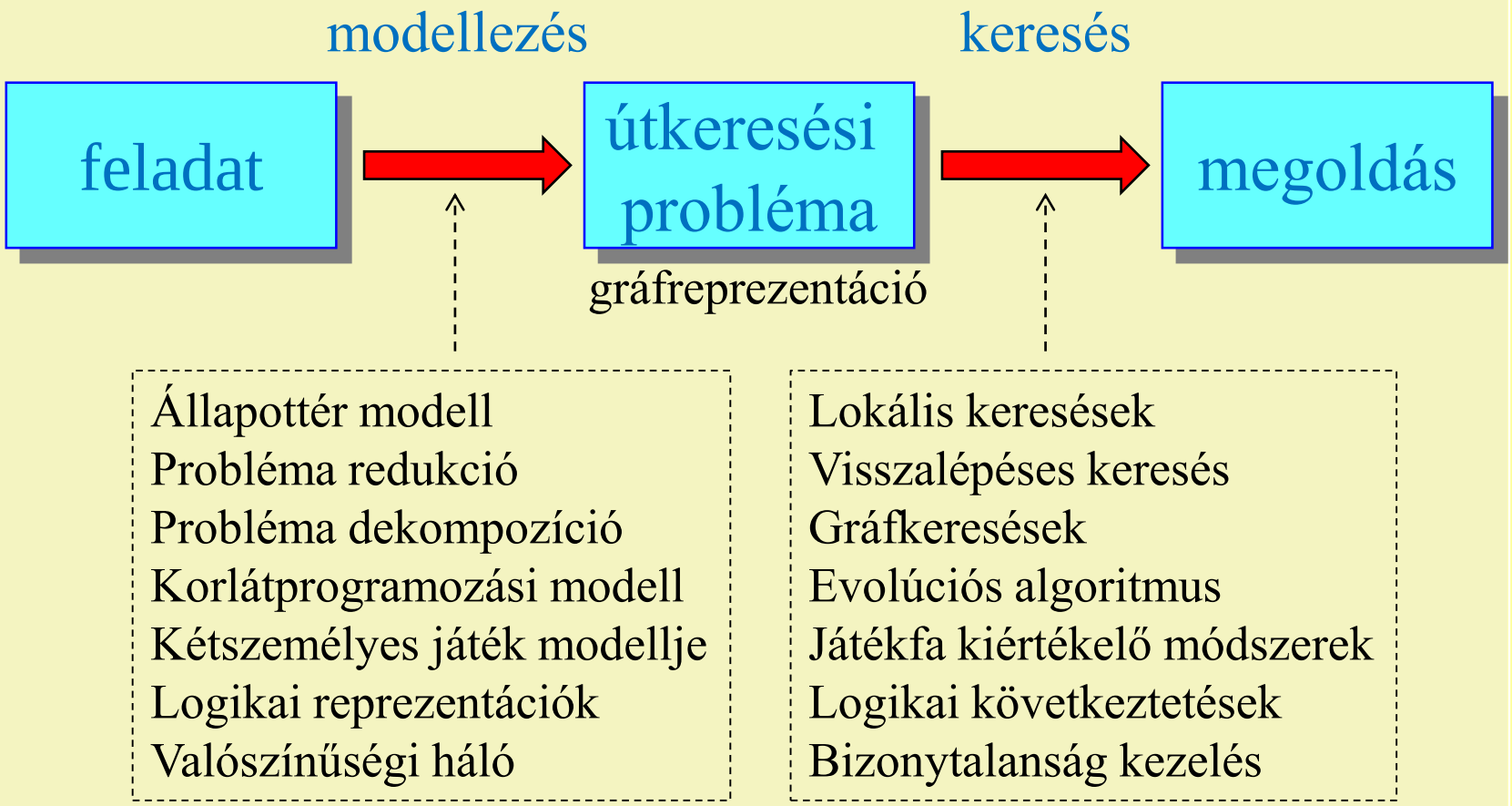
+ gépi látás, gépi cselekvés

- Megoldandó feladat: nehéz
  - A feladat problématere hatalmas,
  - szisztematikus keresés helyett intuícióra, kreativitásra (azaz heurisztikára) van szükségünk ahhoz, hogy elkerüljük a kombinatorikus robbanást.
- Szoftver viselkedése: intelligens

„mesterjelölt szintű

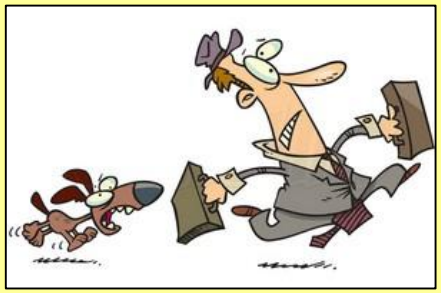
  - Turing teszt vs. kínai szoba elmélet
  - Nem multifunkcionális, nem multidiszciplináris
- Felhasznált technológiák: sajátosak
  - speciális reprezentáció a feladat modellezéséhez
  - heurisztikával megerősített hatékony algoritmusok
  - gépi tanulás módszerei

2. MODELLEZÉS & KERESÉS



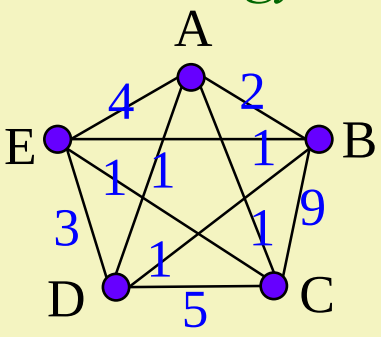
Mire kell a modellezésnek fókuszálni

- ***Problématér elemei***: probléma lehetséges válaszai
- ***Cél***: egy helyes válasz (megoldás) megtalálása
- ***Keresést segítő ötletek*** (heurisztikák):
  - Problématér hasznos elemeinek elválasztása a haszontalanoktól.
  - Kiinduló elem kijelölése.
  - Az elemek szomszédsági kapcsolatainak kijelölése, hogy a probléma tér elemeinek szisztematikus bejárását segítsük.
  - Adott pillanatban elérhető elemek rangsorolása.



Utazó ügynök problémája

Adott n város a közöttük vezető utak költségeivel. Melyik a legolcsóbb olyan útvonal, amely az A városból indulva mindegyik várost egyszer érintve visszatér az A városba?



~~AEDCBA~~

~~AEDBCA~~

~~ACEDBA~~

felesleges ~~ACEBDA~~

szomszédos elempár cseréje

start: ABCDEA 2+9+5+3+4=23

ACBDEA 1+9+1+3+4=18

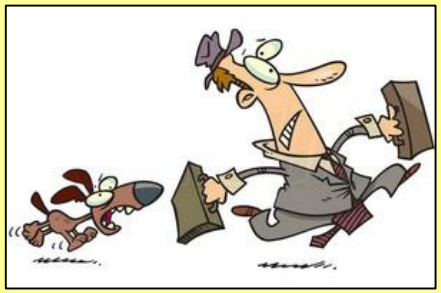
ABDCEA 2+1+5+1+4=13

ABCEDA 2+9+1+3+1=16

ABDECA 2+1+3+1+1=8

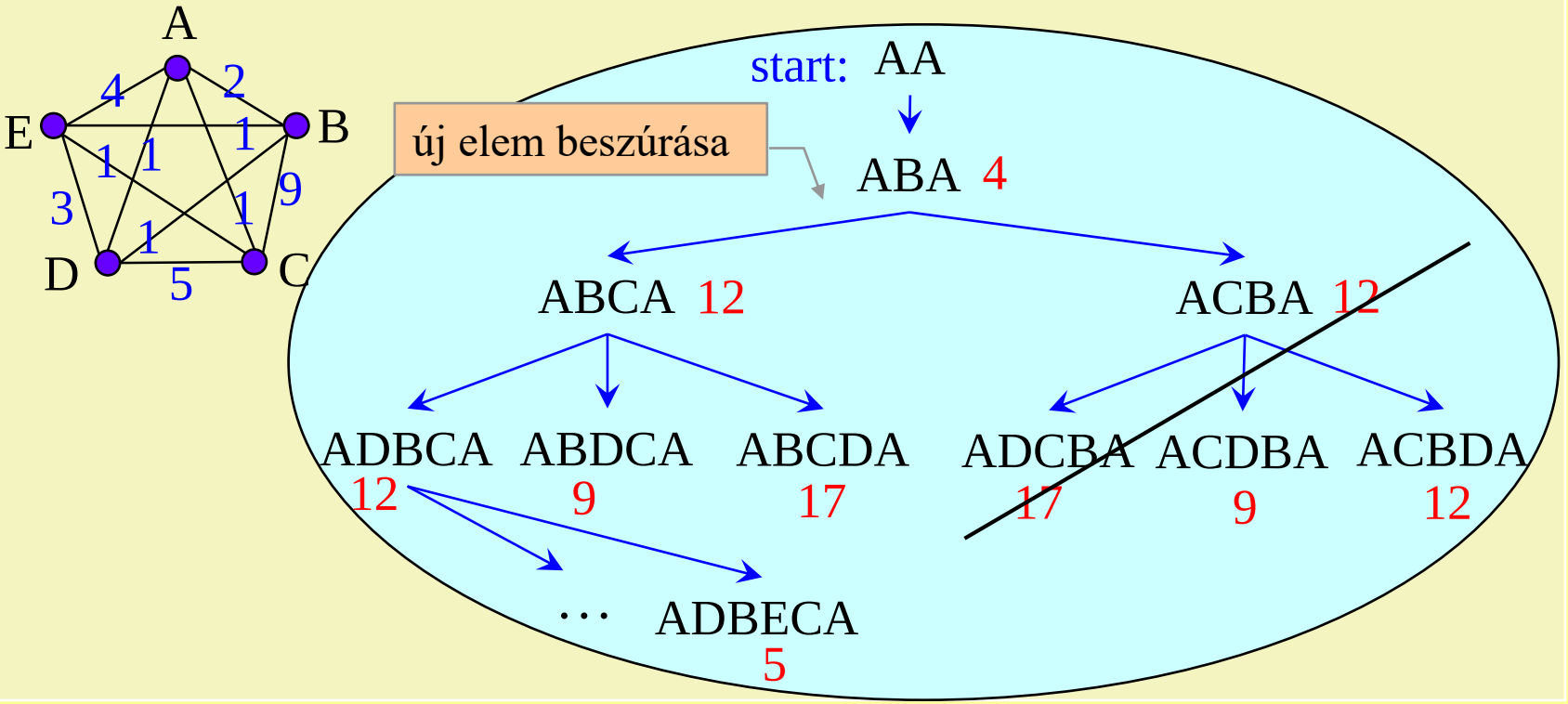
ADBECA 1+1+1+1+1=5

…



Utazó ügynök problémája

Adott n város a közöttük vezető utak költségeivel. Melyik a legolcsóbb olyan útvonal, amely az A városból indulva mindegyik várost egyszer érintve visszatér az A városba?



Útkeresési probléma

- Útkeresési probléma az, amelynek megoldása megfeleltethető egy élsúlyozott irányított gráfbeli
  - csúcsnak (célcsúcs), vagy még inkább
  - útnak (startcsúcsból célcsúcsba, esetleg a legolcsóbb)

Számos olyan modellező módszert ismerünk, amely a kitűzött feladatot útkeresési problémává fogalmazza át.

- Ez a gráf (δ-gráf) lehet végtelen nagy, de
  - csúcsainak kifoka véges, és
  - élei súlyának (költségének) van egy konstans globális pozitív alsó korlátja (δ).

Gráf fogalmak 1.

csúcsok, irányított élek	N, $A \subseteq N \times N$ (végtelen számosság)
él n-ből m-be	$(n,m) \in A$ $(n,m \in N)$
n utódai	$\Gamma(n) = \{m \in N \mid (n,m) \in A\}$
n szülei	$\pi(n) \in \Pi(n) = \{m \in N \mid (m,n) \in A\}$
irányított gráf	$R=(N,A)$
véges sok kivezető él	$\lvert \Gamma(n) \rvert < \infty \quad (\forall n \in N)$
élköltség	$c: A \to \mathbb{R}$
δ-tulajdonság ($\delta \in \mathbb{R}^+$)	$c(n,m) \geq \delta > 0 \quad (\forall (n,m) \in A)$
δ-gráf	δ-tulajdonságú,véges sok kivezető élű, élsúlyozott irányított gráf

Gráf fogalmak 2.

- irányított út

  $\alpha = (n,n_1),(n_1,n_2),\ldots,(n_{k-1},m)$
  $= \langle n,n_1,n_2,\ldots,n_{k-1},m\rangle$

  δ-gráfokban ez végtelen sok út esetén is értelmes.

  Értéke ∞, ha nincs egy út se.

- út hossza

  az út éleinek száma: $|\alpha|$

- út költsége

  $c(\alpha)=c^{\alpha}(n,m) := \Sigma_{i=1..k}\, c(n_{i-1},n_i)$
  ha $\alpha = \langle n=n_0,n_1,n_2,\ldots,n_{k-1},m=n_k\rangle$

- opt. költség

  $c^*(n,m) := \min_{\alpha\in\{n\to m\}} c^{\alpha}(n,m)$
  $c^*(n,M) := \min_{\alpha\in\{n\to M\}} c^{\alpha}(n,m)$

- opt. költségű út

  $n\to^* m := \min_c\{\alpha \mid \alpha\in\{n\to m\}\}$
  $n\to^* M := \min_c\{\alpha \mid \alpha\in\{n\to M\}\}$

Gráfreprezentáció fogalma

- Minden útkeresési probléma rendelkezik egy (a probléma modellezéséből származó) gráfreprezentációval, ami egy (R,s,T) hármas, amelyben
  - $R=(N,A,c)$ δ-gráf az ún. reprezentációs gráf,
  - az $s \in N$ startcsúcs,
  - a $T \subseteq N$ halmazbeli célcsúcsok.
- és a probléma megoldása:
  - egy $t \in T$ cél megtalálása, vagy
  - egy $s \longrightarrow T$, esetleg $s \longrightarrow^* T$ optimális út megtalálása

$s \longrightarrow T$: *s*-ből *T* egyik csúcsába vezető irányított út

$s \longrightarrow^* T$: *s*-ből *T* egyik csúcsába vezető legolcsóbb irányított út

Keresés

- Egy útkeresési probléma megoldásához a reprezentációs gráfjának nagy mérete miatt speciális (nem-determinisztikus, heurisztikus) útkereső algoritmusra van szükség, amely
  - a startcsúcsból indul (kezdeti aktuális csúcs);
  - minden lépésben nem-determinisztikus módon új aktuális csúcso(ka)t választ a korábbi aktuális csúcs(ok) segítségével (gyakran azok gyerekei közül);
  - tárolja a már feltárt reprezentációs gráf egy részét;
  - megáll, ha célcsúcsot talál vagy nyilvánvalóvá válik, hogy erre semmi esélye.

Kereső rendszer (KR)

Procedure *KR*
1. ADAT := *kezdeti érték*
2. **while** ¬*terminálási feltétel*(ADAT) **loop**
3. SELECT SZ FROM *alkalmazható szabályok*
4. ADAT := SZ(ADAT)
5. **endloop**

end

globális munkaterület
tárolja a keresés során megszerzett és megőrzött ismeretet (egy részgráfot)
(*kezdeti érték ~ start csúcs, terminálási feltétel ~ célcsúcs*)

keresési szabályok
megváltoztatják a globális munkaterület tartalmát
(*előfeltétel, hatás*)

vezérlési stratégia
alkalmazható szabályok közül kiválaszt egy „megfelelőt”
(*általános elv + heurisztika*)

Kereső rendszerek vizsgálata

- helyes-e (azaz korrekt választ ad-e)
- teljes-e (minden esetben választ ad-e)
- optimális-e (optimális megoldást ad-e)
- idő bonyolultság
- tár bonyolultság

3. GÉPI TANULÁS

- Egy algoritmus tanul, ha egy feladat megoldása során olyan változások következnek be a működésében, hogy később ugyanazt a feladatot vagy hasonló feladatokat jobban (eredmény, hatékonyság) képes megoldani, mint korábban.

- A tanulás sokszor a megoldandó probléma néhány konkrét esetének, a tanító példáknak segítségével történik. A tanulás lehet:
  - felügyelt, ha a tanító példák input-output párok
  - felügyelet nélküli, ha a tanító példák csak inputok
  - megerősítéses, ha a tanító példák input-hasznosság párok