I. Bevezetés

1. A mesterséges intelligencia (MI) fogalma
2. Probléma modellezés
3. Kereső rendszerek az MI-ben

1

1. AZ MI FOGALMA

1956 nyár. Darthmouth College-i konferencia

Kezdeti cél:

Az emberi gondolkodás számítógép segítségével történő reprodukálása.

2

Erős mesterséges intelligencia (EMI)

szélsőséges redukcionizmus

⬇

Church-Turing tézis

⬇

Neumann-elvű számítógép

⬇

Az emberi gondolkodás reprodukálható egy működő programmal

3

Gyenge mesterséges intelligencia

Cél:

Az MI az emberi gondolkodás számítógépes reprodukálása szempontjából hasznos elveket, módszereket, technikákat kutatja, rendszerezi, fejleszti.

Az MI az informatikának a gondolkodási-tudományos előőrse.

4

MI története

- 60-as évek: általános feladatok, általános technikák
- 70-es évek: feladatosztályok, speciális módszerek
- 80-as évek: konkrét feladatok, szakértő rendszerek
- 90-es évek: újrafelfedezett módszerek, hibrid rendszerek, matematikai háttér

5

Első szakasz (60-as évek)

- Eredmények: kétszemélyes játékok (dáma, sakk), beszélgető program (ELIZA,1966)

6

ELIZA

- Illesztési szabályok
  - <**a**> *ön* <**b**> *engem* <**c**>**.**
  - *Miért gondolja, hogy ön* <**a**> *én* <**b**> <**c**>?
    - Úgy érzem, hogy *ön* mostanában *engem* un.
    - *Miért gondolja, hogy ön* úgy érzi, hogy *én* mostanában unom?
- Emlékezési szabályok
- Folytatási szabályok

7

Első szakasz (60-as évek)

- Eredmények: kétszemélyes játékok (dáma, sakk), beszélgető program (ELIZA,1966)
- Módszerek, eszközök: GPS, rezolúció (1966), LISP(1958), mesterséges neuronhálózatok (1969), evolúciós algoritmusok (1959)
- Kudarcok: DOCTOR-PARRY, nyelvi fordítók, kombinatorikus robbanás

8

Második szakasz (70-es évek)

- Eredmények: SHRDLU (1972), BACON, AM, EURISKO
- Módszerek, eszközök: Prolog, heurisztikus keresési technikák, tudásábrázolási módszerek (kognitív modellek)
- Kudarcok: MI fejlődési trendje, meseíró program

9

Harmadik szakasz (80-as évek)

- Eredmények: DENDRAL (1969-78), MYCIN(1976), PROSPECTOR(1979), XCON (1982)
- Módszerek, eszközök: tudásalapú szakértő rendszerek, shell-ek, módszertanok, nem klasszikus logikák, bizonytalanság kezelése
- Kudarcok: rendszerek elkészítése lassabb, mint a gyorsan változó programozási környezet

10

Negyedik szakasz (90-as évektől)

- Eredmények: logisztika, űrkutatás, Deep Blue, döntés támogató rendszerek, nyelvi fordítók, robotika (beszélgetés, gépi látás, tervgenerálás, gépi tanulás)
- Módszerek, eszközök: elosztott tudás reprezentálása (mesterséges neuron háló, evolúciós algoritmus, ágens szemlélet), döntéselmélet (valószínűségi hálók), beszédfelismerés (rejtett Markov modellek)

11

MI helye

- *A MI egy műszaki tudomány:*
  - adott működés minél jobb minőségű számítógépes reprodukálása.
- *Az MI-vel szemben a kognitív pszichológia:*
  - emberi gondolkodás természetének megismeréséhez tervez számítógépes modelleket.

12

Miről ismerhető fel a mesterséges intelligencia?

- A megoldandó feladatról
  - A számítógéppel nem megoldható feladatokat akarunk számítógéppel megoldani ☺
  - *Példa:* orvosi diagnózis, sakk játék, természetes nyelv megértése
  - *Ellenpélda:* telefonkönyv nyilvántartó program
- A megoldó algoritmusról
  - Turing teszt

13

MI tárgya

- Azon feladatok számítógépes megoldása, amelyek
  - megoldása nehéz
  - az embertől is kellő szakértelmet, kreativitást és intuíciót kíván (szemléletmód váltások)
  - megoldásukban ma többnyire az ember a jobb

/medve/

14

- a probléma tere (lehetséges válaszok száma) nagy, az összes lehetőség kipróbálása szisztematikus úton nem lehetséges,

/n-királynő/

15

- a válasz sokszor elemi tevékenységek sorozatával írható le,
- amely előre nem rögzíthető, hanem több lehetséges sorozat közül kell kiválasztani.
- irányított keresésre van szükség.

/ Rubik kocka /

16

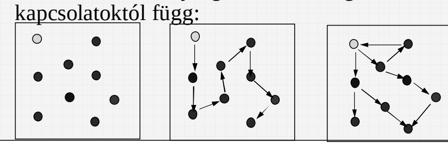
2. PROBLÉMA MODELLEZÉS

- Problématér: összes <u>lehetséges</u> válasz halmaza
- Cél: a problémára adható <u>helyes</u> választ (megoldást) kell megtalálni a problématérben
- A problématér szűkítése
  - <u>hasznos</u> válaszok kijelölése
  - szomszédsági kapcsolatok definiálása
  - kiinduló pont rögzítése
- Kiértékelő függvény segítheti a válaszok közötti válogatást

17

Útkeresési probléma válaszok - irányított gráfbeli csúcsok

- Feleltessük meg a válaszokat egy irányított gráf csúcsainak, a köztük levő szomszédsági kapcsolatokat egy-egy irányított élnek.
- A helyes választ az annak megfeleltetett célcsúcs megkeresésével lelhetjük meg.
- A keresés hatékonysága szomszédsági kapcsolatoktól függ:



18

Speciális útkeresési probléma

- A lehetséges válasz elemi lépések sorozata.
  - Két ilyen sorozat (válasz) között szomszédsági kapcsolatot definiál az, hogy a kezdő lépéseik milyen hosszan egyeznek meg.
- Ilyenkor egy <u>másféle</u> irányított gráffal is lehet szemléletetni a problémát
  - Az elemi lépések, mint irányított élek definiálnak egy másikat, amelyben a válaszokat egy rögzített csúcsból kiinduló irányított utak szimbolizálják.

19

Gráf fogalmak 1.

csúcsok, ir. élek	N, $A \subseteq N \times N$ (számosság)
él n-ből m-be	$(n,m) \in A$ $(n,m \in N)$
n utódai, szülei	$\Gamma(n)$, $\Pi(n)$, $\pi(n)$
irányított gráf	$R=(N,A)$
σ-tulajdonság	$\lvert\{(n,m) \in A \mid m \in N\}\rvert < \sigma \ \forall n \in N$
élköltség	$c: A \to \mathbf{R}$, $c(n,m)$ $\forall (n,m) \in A$
δ-tulajdonság	$c(n,m) \geq \delta > 0$ $\forall (n,m) \in A$
δ-gráf	δ, σ -tulajdonságú élsúlyozott irányított gráf

20

Gráf fogalmak 2.

irányított út	$\alpha=(n,n_1),(n_1,n_2),\ldots,(n_{k-1},m)$ $(n,n_1,n_2,\ldots,n_{k-1},m)$, $n \xrightarrow{\alpha} m$, $n \to m$
n-ből kiinduló utak	$\{n \to m\}$, $\{n \to M\}$
ir. út hossza	$\lvert \alpha \rvert$
ir. út költsége	$c^{\alpha}(n,m) := \Sigma_i \, c(n_{i-1},n_i)$
opt. költség	$c^{*}(n,m) := \min_{\alpha \in \{n \to m\}} c^{\alpha}(n,m)$
opt. költségű út	$n \xrightarrow{*} m$, $n \xrightarrow{*} M$

21

Gráf fogalmak 3.

- irányított kör
- s gyökerű irányított fa
- irányított gráf s gyökerű irányított feszítőfája
- irányított gráf s gyökerű optimális irányított feszítőfája

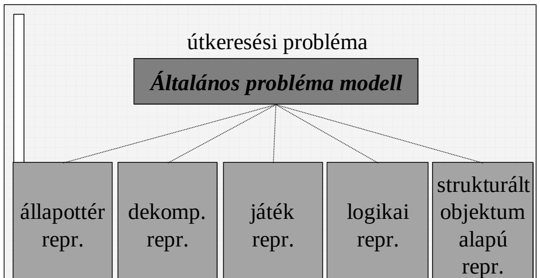
22

Gráfreprezentáció fogalma

- Egy útkeresési probléma gráfreprezentációja egy (R,s,T) hármas, amelyben
  - $R=(N,A,c)$ δ-gráf az un. reprezentációs gráf, amely a probléma teret szimbolizálja,
  - az $s \in N$ startcsúcs a kiinduló pont,
  - a $T \subseteq N$ halmazbeli célcsúcsok.
- A feladat megoldása:
  - többnyire egy $s \to T$, esetleg egy $s \xrightarrow{*} T$ optimális út megtalálása
  - néha egy $t \in T$ cél megtalálása

23

Reprezentációs modellek



24

Állapottér-reprezentáció

- Ez egy széles körben (nemcsak a MI-ban) használt modell, amely segítségével egy problémát specifikálhatunk.
- Jellegzetessége, hogy a problémák megoldását műveletek sorozataként fogalmazza meg, ennél fogva az állapottér-reprezentáció alkalmazása egy útkeresési problémát (többnyire speciális útkeresési problémát) ír le.

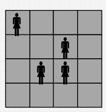
25

Állapottér-reprezentáció modellje

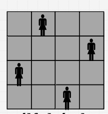
- Állapottér (domináns típusérték-halmaz)
  - invariáns
- Műveletek (elemi lépés az állapottérben)
  - előfeltétel, hatás
- Kezdő állapot(ok) vagy előfeltétel
- Célállapot(ok) vagy utófeltétel

26

n-királynő probléma 1.



általános állapot



célfeltétel= …

Állapottér: ***Tábla*** *= mátrix([1..n,1..n];{K, SZ})*

invariáns: „királynők száma"= *n*

Művelet: ***Áthelyez****(x,y,u,v)*:***Tábla→Tábla*** (*a*:***Tábla***)

HA (*a*[*x,y*]=*K*) *és* (*a*[*u,v*]=*SZ*)

AKKOR *a*[*x,y*]↔ *a*[*u,v*]

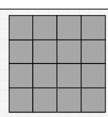
27

Állapottér-reprezentáció állapot gráfja

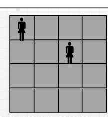
állapot	csúcs
művelet hatása	irányított él
művelet költsége	élköltség
kezdő állapot	startcsúcs
célállapotok	célcsúcsok
műveletsorozat hatása	irányított út

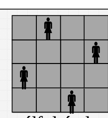
28

n-királynő probléma 2.



kezdőállapot





célfeltétel = …

Állapottér: ***Tábla*** *= mátrix([1..n,1..n];{K, SZ})*

invariáns: „királynők száma"≤ *n*

Művelet: ***Elhelyez****(sor,oszlop)*:***Tábla→Tábla*** (*a*:***Tábla***)

HA *a*[*sor,oszlop*]=*SZ és*

„királynők száma"< *n*

AKKOR *a*[*sor,oszlop*]← *K*

29

Megjegyzés

- Mindkét eddig látott reprezentáció - az első egy általános, a második egy speciális - útkeresési problémát definiál:

- 1. reprezentáció
  - Célcsúcsot keresünk
  - Nincs megadva startcsúcs
  - problématér = állapottér
    - válasz = csúcs
    - szomszéd = él
- 2. reprezentáció
  - Célcsúcsba vezető utat keresünk a startcsúcsból indulva
  - problématér ≠ állapottér
    - válasz = *n* hosszú út
    - szomszéd = közös kezdő részút

30

n-királynő probléma 3a.

Állapottér:

Tábla = *mátrix*([1..*n*,1..*n*];{*K*, *SZ*})
invariáns: „királynők száma”≤ *n*
első valahány sorban, de
egy sorban csak egy királynő

Művelet:

Helyez(oszlop):Tábla→Tábla (*a*:***Tábla***)

HA a *sor* az első olyan sor, ahol nincs *K*
AKKOR *a*[*sor*,*oszlop*]← *K*

31

n-királynő probléma 3b.

Állapottér:

Tábla =*rec*(*m*:*mátrix*([1..*n*,1..*n*];{*K*, *SZ*}), ***sor*:N**)
invariáns: *sor* ≤ *n*
1-től *sor*-ig egy-egy királynő
kezdő állapotnál: sor = 0

Művelet:

Helyez(*oszlop*):***Tábla→Tábla*** (*a*:***Tábla***)

HA *a.sor* < *n*
AKKOR *a.sor* ← *a.sor+1*
a.m[*a.sor*,*oszlop*]← *K*

32

Megjegyzés

- Sok modellje lehet ugyanannak a feladatnak.
- Művelet előfeltételének (szomszédsági kapcsolatnak) megváltoztatásával csökken az állapottér mérete, azaz az utak (a megengedett válaszok) száma.
- A műveleti előfeltétel ellenőrzése ügyesebb reprezentáció mellett - például az állapottér átalakításával - hatékonyabb lehet.

33

n-királynő probléma 4a.

Állapottér:

Tábla =*rec*(*m*:*mátrix*([1..*n*,1..*n*];{*K*, *SZ*}), *sor*:**N**)
invariáns: **nincs ütés** és *sor* ≤ *n*
1-től *sor*-ig egy-egy királynő
kezdőáll.: sor = 0 *céláll.:* ***sor = n***

Művelet:

Helyez(*oszlop*):***Tábla→Tábla*** (*a*:***Tábla***)

HA *a.sor* < *n* és **„az új királynő nem üt”**
AKKOR *a.sor* ← *a.sor+1*
a.m[*a.sor*,*oszlop*]← *K*

34

n-királynő probléma 4b.

Állapottér:

Tábla= *rec*(*m*:*mátrix*([1..*n*,1..*n*];{*K*, **Ü**, *SZ* }), *sor*:**N**)
invariáns: nincs ütés és *sor* ≤ *n*
1-től *sor*-ig egy-egy királynő
kezdőáll.: sor = 0 *céláll.:* ***sor = n***

Művelet:

Helyez(*oszlop*):***Tábla→Tábla*** (*a*:***Tábla***)
HA *a. sor* < *n* *és* ***a.m*[*a.sor+1, oszlop*]=*SZ***
AKKOR *a.sor* ← *a.sor+1*
a.m[a.sor,oszlop] ← *K*
minden megfelelő *i,j* -re: *a.m*[*i,j*] ← *Ü*

35

n-királynő probléma C++ osztálya

```
class Table {
   int   n;
   int** matrix; // 0-Sz; 1-K; 2-Ü
   int   row;

public:

   Table();      // matrix=0, row=-1

   // copy constructor, destructor, etc.

   bool Put(const int col);
   bool Goal() const {return row==n-1;}

};
```

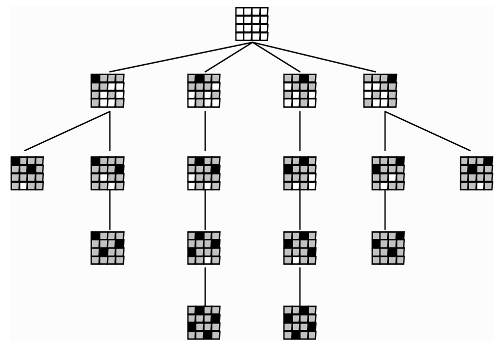
36

n-királynő probléma műveletének C++ metódusa

```
bool Table::Put(const int col){
   if( !(col>=0 && col<=n-1    &&    row<n-1
      &&  matrix[row+1][col]==0 ) )return false;
   row++;
   matrix[row][col] = 1;
   for(int i=row+1;i<n;i++)
         matrix[i][col] = 2;
   for(int i=row+1;i<n && col-row+i<n;i++)
         matrix[i][col-row+i] = 2;
   for(int i=row+1;i<n && col-(i-row)>=0;i++)
         matrix[i][col-(i-row)] = 2;
   return true;
}
```

37

Állapot gráf



Tologató játék

2	8	3
1	6	4
7		5

⇒

1	2	3
8		4
7	6	5

Állapottér: ***Tábla****=rec(mátrix, üreshely pozíciója)*

Művelet: ***Tol****(irány):****Tábla→Tábla*** (*a:****Tábla***)

HA *a.üreshely+irány* egy érvényes pozíció

AKKOR *a.mátrix[a.üreshely]↔ a.mátrix[a.üreshely+irány]*

a.üreshely←a.üreshely+irány

39

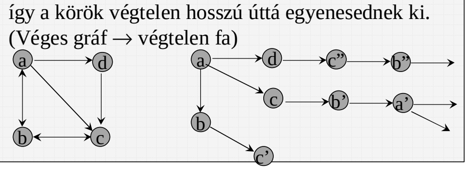
Megjegyzés

- Az útkeresés hatékonysága a reprezentációs gráf (itt állapot gráf) bonyolultságát (csúcsok, élek számát, körök gyakoriságát, körök hosszát) meghatározó reprezentáción múlik.
- A reprezentációs gráfnak csak a startcsúcsból elérhető része érdekel bennünket.
- Ha egy keresés nem végez körfigyelést, csak a megelőző csúcsba történő visszalépést zárja ki, akkor tulajdonképpen nem az eredeti gráfon, hanem annak úgynevezett fává egyenesített változatában dolgozik.

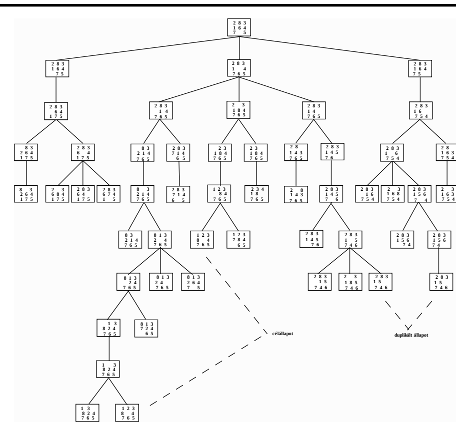
40

Irányított gráf fává egyenesítése

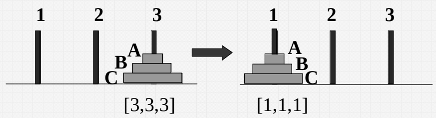
- Elméletben el is készíthetjük ezt a fát, amely a startcsúcsból kivezető utakat tartalmazza
  - Oda-vissza irányuló élek közül a start csúcs felé irányulót elhagyjuk. (kettő hosszú körök törlődnek).
  - Egy több úton is elérhető csúcsot megsokszorozzuk, így a körök végtelen hosszú úttá egyenesednek ki. (Véges gráf → végtelen fa)



41



Hanoi tornyai probléma



Állapottér: **Állás**=*vektor*([*A*,*B*,*C*];{*1,2,3*})

Művelet: ***Rak****(honnan, hová)*:**Állás→Állás** (*a*:**Állás**)

HA *a*-ban a *honnan* nem üres, és
a *hová* üres vagy a *hová* felső korongja nagyobb,
mint *honnan* felső korongja

AKKOR *a[honnan felső korongja]← hová*

43

Implementáció

Művelet:

Rak*(honnan, hová)*:**Állás→Állás** (*a*:**Állás**)

l1,i=***linker****(A..C, a[i]=honnan)* *i*-t akarjuk mozgatni

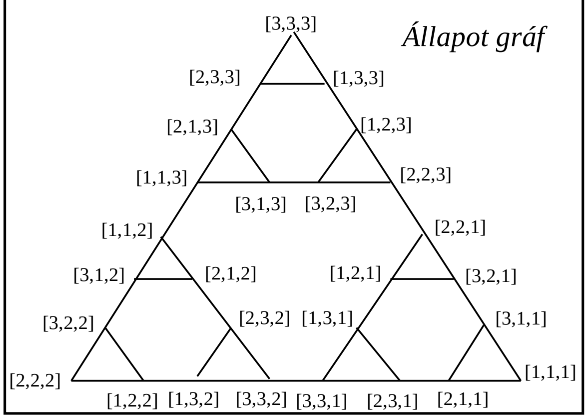
l2,j=***linker****(A..C, a[j]=hová)* *j*-re akarunk rakni

HA *l1 és (¬l2 vagy i<j)*

AKKOR *a[i]← hová*

44

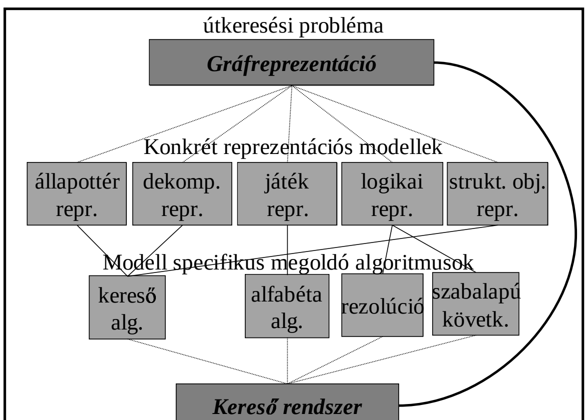
Állapot gráf



3. KERESŐ RENDSZEREK

- A különféle megoldó algoritmusok egy <u>közös algoritmus-sémából</u> származtathatók, hiszen a problémák többnyire ugyanarra az alapfeladatra vezethetők vissza: keresünk egy irányított utat a reprezentációs gráfban
- A fő különbség az egyes kereső algoritmusok között az, hogy általános vagy speciális útkeresési problémát oldanak-e meg, illetve hogy a keresett megoldást optimalizálják-e valamilyen szempontból.

46



Kereső rendszer sémája

<u>Procedure</u> *KR*
1. ADAT ← *kezdeti érték*
2. <u>while</u> ¬ *terminálási feltételt*(ADAT) <u>loop</u>
3. <u>select</u> SZ <u>from</u> *alkalmazható* szabályok
4. ADAT ← SZ(ADAT)
5. <u>endloop</u>

<u>end</u>

A reprezentációs gráf feletti keresés, amely a gráf egy részét (*ADAT*) látja, azt változtatja meg (*SZ*) az általa meghatározott (*select*) módon.

48

A KR részei

globális munkaterület
ADAT

a keresés során megszerzett és megőrzött (deklaratív) ismeret
kezdeti érték, terminálási feltétel

kereső rendszer szabályai
SZ

globális munkaterületet megváltoztató operátorok (procedurális ismeret)
hatás, értelmezési tartomány

vezérlési stratégia
select

végrehajtható szabályt kiválasztó *általános elv* + a konkrét feladattól származó *vezérlési ismeret*

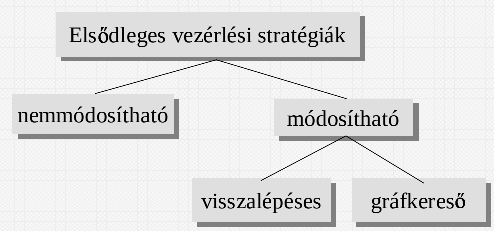
49

Vezérlési vagy keresési stratégia

- Elsődleges stratégia: teljesen független a feladattól.
- Másodlagos stratégia: a reprezentációs modelltől függ, de a modellben rögzített konkrét ismeretektől nem.
- Vezérlési ismeret (heurisztika): A feladattal kapcsolatos, a reprezentációban nem rögzített konkrét ismeret, amelytől a jó eredményt, és a megfelelő hatékonyságot várjuk.
  - Sorrendi: alkalmazható szabályok közül választ
  - Vágó: alkalmazható szabályok számát szűkíti

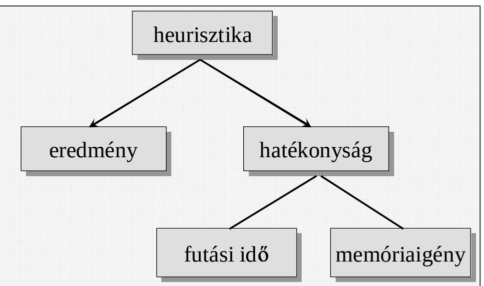
50

Elsődleges stratégiák osztályozása



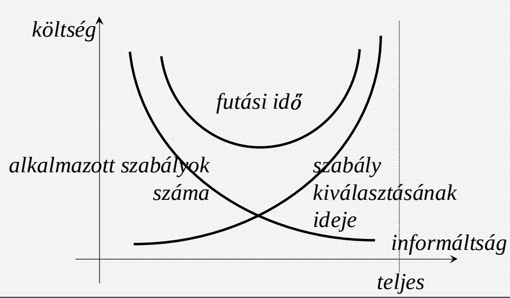
51

A heurisztika hatása a KR működésére



52

A KR futási ideje



53