Budapesti Műszaki és Gazdaságtudományi Egyetem
Méréstechnika és Információs rendszerek Tanszék

Mesterséges intelligencia – VIMIAC10

Bevezetés – követelmények
Mit értünk intelligencia alatt?
Milyen főbb területekről lesz szó?

Dr. Hullám Gábor

MŰEGYETEM 1782

Kurzus információk

Előadás

- A kar vezetésének döntése értelmében a tárgy valós idejű távoktatási formában lesz megtartva

- Az előadások központilag rögzítésre kerülnek, egyedileg rögzítésük nem megengedett!

- Az előadás vágott verziója vagy egy azzal ekvivalens videóanyag lesz megosztva

Kurzus információk

Előadás

- Minden héten: szerda 10:15-12:00
- +páros hetenként: csütörtök 10:15-12:00
- Kivéve:
  - szept.23-án nem lesz előadás sportnap miatt
  - nov.12-én nem lesz előadás TDK miatt

Előadó

- Dr. Hullám Gábor – tárgyfelelős
  hullam.gabor@mit.bme.hu

Demonstrátorok

- D01, D02, D03, D04, D05, D06, D07, D08 (GDPR :-)

Platformok

Teams csoport ✓

- Mesterséges intelligencia - BMEVIMIAC10-HU
- Kommunikáció

MIT Házi feladat portál ✓

- https://hf.mit.bme.hu
- Házi feladatok leadása, automatikus kiértékelése

Moodle ✓

- https://edu.vik.bme.hu
- Tananyagok, videók elérése

Tantárgyi honlap ✓

- http://www.mit.bme.hu/oktatas/targyak/vimiac10
- Tematika, alapinformációk (tartalék)

Követelmények

3 kredit

- 2 ZH
- 1 házi feladat (3 részből áll)
- Összpontszám: **100** = ZH-1 (max. 32p)
  + ZH-2 (max. 32p)
  + házi feladat (max. 36p)
  (1 részfeladat:12p)

- Követelmények:
- Min. 40%-a az elérhető ZH pontoknak: **25**
  - **ZH-nként minimum: 12,5**
- Min. 40%-a az elérhető teljes pontszámnak: **40**

Követelmények

Követelmények:

- Min. 40%-a az elérhető ZH pontoknak: **25**
- **ZH-nként minimum: 12,5**
- Min. 40%-a az elérhető teljes pontszámnak: **40**

Jegyek-ponthatárok

elégséges	40,0-49,0
közepes	49,5-64,0
jó	64,5-79,5
jeles	80,0-

Követelmények

IMSc pontok – összesen 15 pont nyerhető el a tárgyban

- Az IMSc program hallgatói számára emelt szintű fakultatív házi feladatokat kínálunk, továbbá a zh-ban is lesz IMSc plusz feladat.
- Természetesen ezeket bárki megszerezheti, és ezek nem emelik a ponthatárokat.
- A tervezett megoszlás:
  - 2-3 emelt szintű kis hf (5-10 pont /hf)
  - A 2 Zh-n emelt szintű pluszfeladat (5 pont /iMsc feladat)
- A tárgyban elért IMSc pontok= Min(15,megszerzett IMSc pontok)

ZH időrend

Kedd reggeli és péntek délutáni ZH-sáv

- ZH-1: 2020. Október 27. Kedd 8:30-10:00
- pZH-1: 2020. November 6. Péntek 14:15-16:00

- ZH-2: 2020. December 4. Péntek 14:15-16:00
- pZH-2: 2020. December 18. Péntek

Terembeosztás a ZH előtt lesz elérhető Teamsen / a honlapon, ha jelenléti lesz.

Pót-pótZH nincs!!

Házi feladat

- 3 részfeladat Java vagy Python nyelven
  - Kényszerkielégítés/Keresés
  - Valószínűségi hálók
  - Megerősítéses tanulás

- Beadás a HF portálon keresztül:

  https://hf.mit.bme.hu

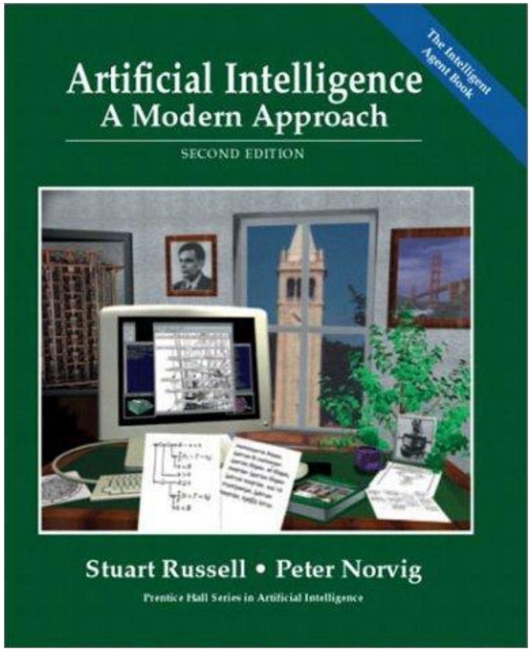
- Minden feladat automatikus értékelésen esik át
- Határidőt követően **plágiumellenőrzés** lesz !!

Házi feladat határidők

  - A tárgy honlapján / TEAMS-en lesznek elérhetők

Jegyzet

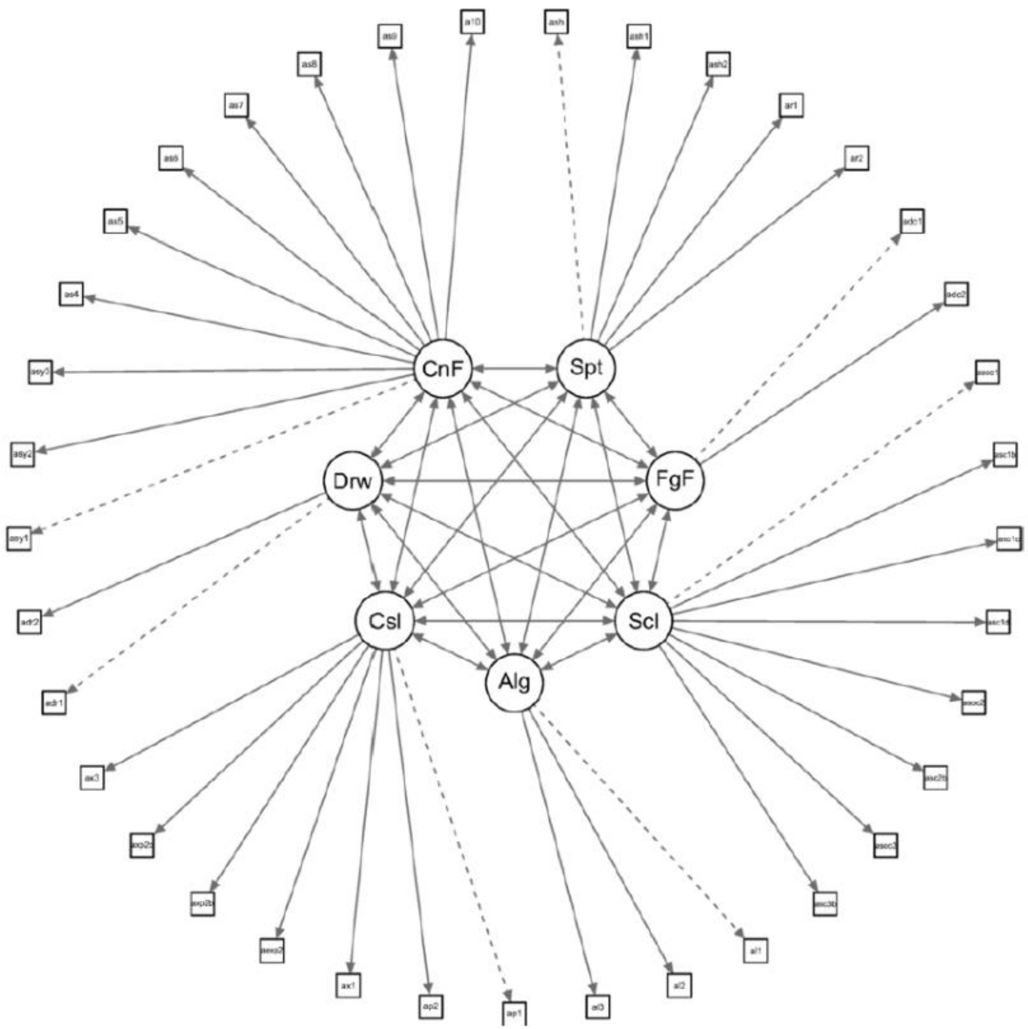
- Russell – Norvig:
  Mesterséges intelligencia modern megközelítésben
- http://mialmanach.mit.bme.hu



Tantárgyi holnapon:

- http://www.mit.bme.hu/oktatas/targyak/vimiac10/eloadasanyagok-jegyzetek

Az intelligencia dimenziói



Spt = spatial ability;
FgF = figural fluency;
Scl = social reasoning;
Alg = algebraic reasoning;
Csl = causal reasoning;
Drw = drawing ability;
CnF = conceptual fluency.

Golino, H.F. and Demetriou, A., 2017. Estimating the dimensionality of intelligence like data using Exploratory Graph Analysis. *Intelligence*.

Miért van szükségünk mesterséges intelligenciára?

- Segít megismerni az emberi kognitív folyamatokat

- Emberi szakértők támogatása, kiegészítése
- Emberi képességek támogatása, kiegészítése, kiterjesztése

- Muszáj: rendelkezésre álló adat és tudás meghaladja az emberi felfogó/feldolgozó képességet

- Olcsóbb, rugalmasabb, tartósabb, mint egy emberi szakértő :)

Miért van szükségünk mesterséges intelligenciára? – mérnöki szempontok

- Megoldható és kivitelezhető problémákkal foglalkozunk
- Hogyan elemezzünk problémákat, melyek mesterséges intelligencia alkalmazását igénylik?
- Hogyan specifikáljuk a problémát?

- Hogyan rögzítsük és tartsuk karban a tudást (formálisan)?
- Milyen architektúrát használjuk a gépi problémamegoldáshoz, ami…
  - Tudás reprezentációt /menedzsmentet…
  - Érzékelést, következtetést…
  - Tanulást…
  - Keresést… igényel?

Lehetséges MI megközelítések

Emberi módon gondolkodó MI	**Racionálisan gondolkodó MI**
Emberi módon cselekvő MI	Racionálisan cselekvő MI

- *1. Emberi ≠ racionális (és ez nem feltétlenül negatív megjegyzés)*
- *2. Nem feltétlenül az emberi vagy a természetben elterjedt mód a legjobb, ha valamilyen célt akarunk elérni* – csapkodó szárnyú repülő.

Lehetséges MI megközelítések

Emberi módon gondolkodó MI	**Racionálisan gondolkodó MI**
Emberi módon cselekvő MI	Racionálisan cselekvő MI

- *Emberi módon gondolkodó MI:* kognitív modellezés
  - (kognitív tudományok, neurobiológia)

- *Emberi módon cselekvő MI:* Turing teszt, chat botok
  - Tudásábrázolás
  - Következtetés
  - Természetes nyelvű kommunikáció („beszédértés”)
  - Tanulás

Lehetséges MI megközelítések

Emberi módon gondolkodó MI	**Racionálisan gondolkodó MI**
Emberi módon cselekvő MI	Racionálisan cselekvő MI

- *Racionálisan gondolkodó MI:* logika, következtetés
  - Probléma: 1) nem minden intelligens viselkedés írható le tisztán logikai kifejezésekkel. 2) Miről kell gondolkodni, mi a célja?

- *Racionálisan cselekvő MI:* a „megfelelő” dolgot teszi a feladat megoldásához
  - A rendelkezésre álló információ alapján maximalizálja a „teljesítményt”
  - Kihívás: 1) rendelkezésre álló információ többnyire nem teljes. 2) A teljesítmény mérése hogyan történjen?

Az MI fázisai

- ~1930 Univerzális számítási modell: Turing-gép (1936), Univerzális Turing-gép, Church-Turing hipotézis, Zuse, Neumann,..: „vezérlő program is adat”:
- 1943 McCulloch & Pitts: Bináris kapcsolati agymodell
- 1950 Turing: "Computing Machinery and Intelligence"
- 1956 Dartmouth találkozó: "Artificial Intelligence” megnevezés elfogadása
- 1950s Korai MI programok: sakk, tételbizonyítás

Számítás (keresés) alapú MI

- A Fizikai Szimbólumrendszer hipotézise: A.Newel&H.A.Simon (1976): „A physical symbol system has the necessary and sufficient means for general intelligent action."
- 1966-73 Számítási komplexitási korlátok a keresésben
Elméleti korlátok a neurális hálózatokban
- 1969-79 Tudásalapú szakértői rendszerek

Tudásalapú MI

- 1986-- Neurális hálózatok újbóli megjelenése
- 1988-- Valószínűségi szakértői rendszerek
- 1995-- Gépi tanulás gyors fejlődése

Adatvezérelt MI (2001-)

Autonóm tanulás alapú MI (2011-)

Az MI jósolt korszakai

Gyenge/szűk mesterséges intelligencia
(Artificial Narrow Intelligence , Weak AI)

- intelligenciát mutat de csak egy speciális területen (sakkozik, arcot felismer stb.)

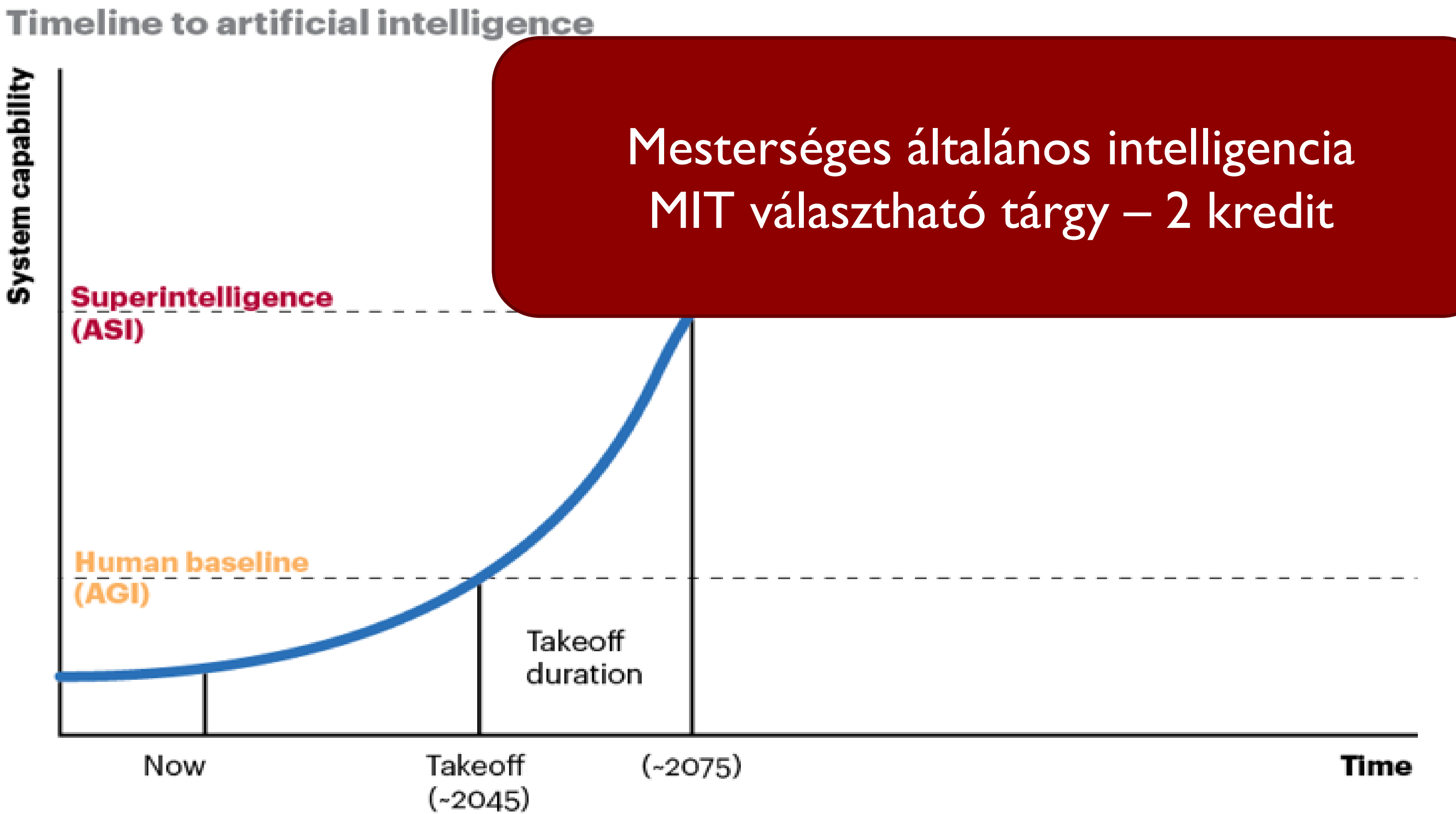
Erős mesterséges intelligencia (Strong Artificial Intelligence)

- Mesterséges általános intelligencia: az élet számos területén intelligensen viselkedik (Artificial General Intelligence)
- Gondolkodik (Human-Level AI): absztrakt gondolkodásra képes, következtet, összetett koncepciókat megért, tanul, általánosít stb.

Szuperintelligencia

- Intelligensebb a legjobb emberi elméknél is, tudásban, kreativitásban, problémamegoldásban…

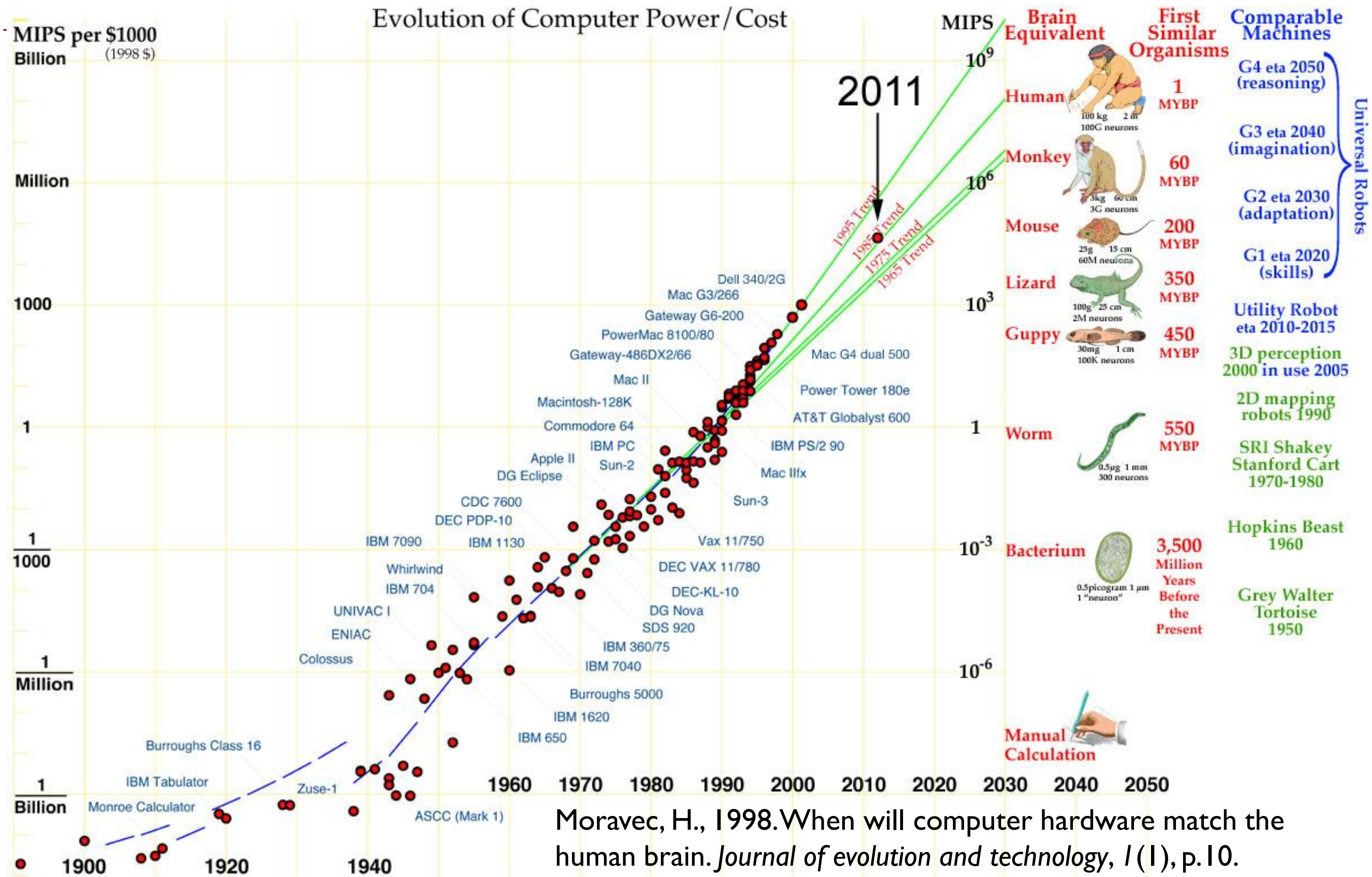
Az MI jósolt korszakai



Mesterséges általános intelligencia
MIT választható tárgy – 2 kredit

Note: AI is artificial intelligence, ASI is artificial superintelligence, and AGI is artificial general intelligence.
Sources: WaitButWhy.com, Nick Bostrom, *Superintelligence: Paths, Dangers, Strategies*; A.T. Kearney analysis

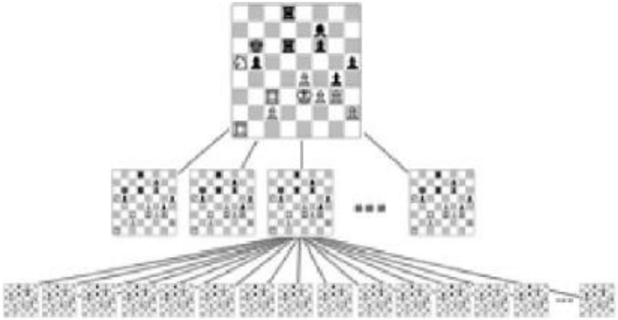
Post-humán intelligens rendszerek

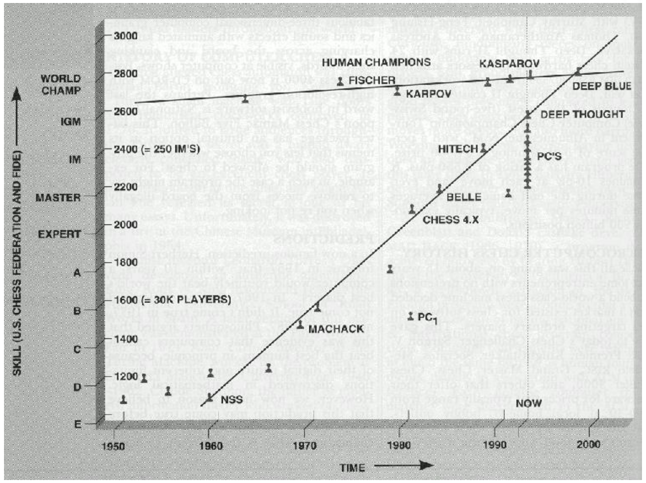


Moravec, H., 1998. When will computer hardware match the human brain. *Journal of evolution and technology*, *1*(1), p.10.

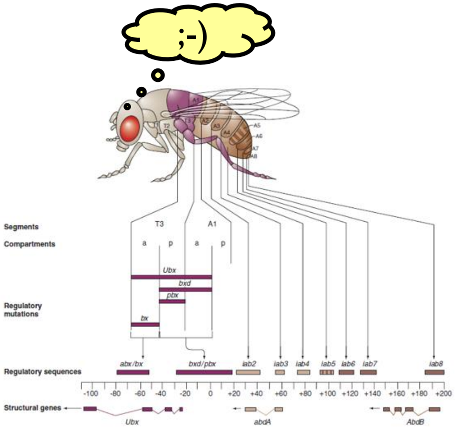
Sakk, az MI kutatások muslicája

J.McCarthy: "Chess as the Drosophila of AI. [Artificial Intelligence]", 1990





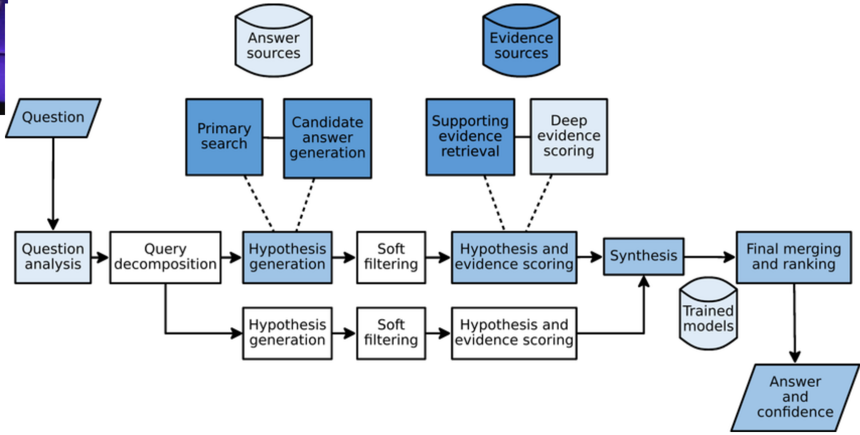
#	Név	Élőpont
1	**SugaR XPrO 1.2 64-bit 4CPU**	**3415**
2	**Komodo 11.2 64-bit 4CPU**	**3402**
3	**Houdini 5.01 64-bit 4CPU**	**3382**
	IBM Deep Blue (1997)	-



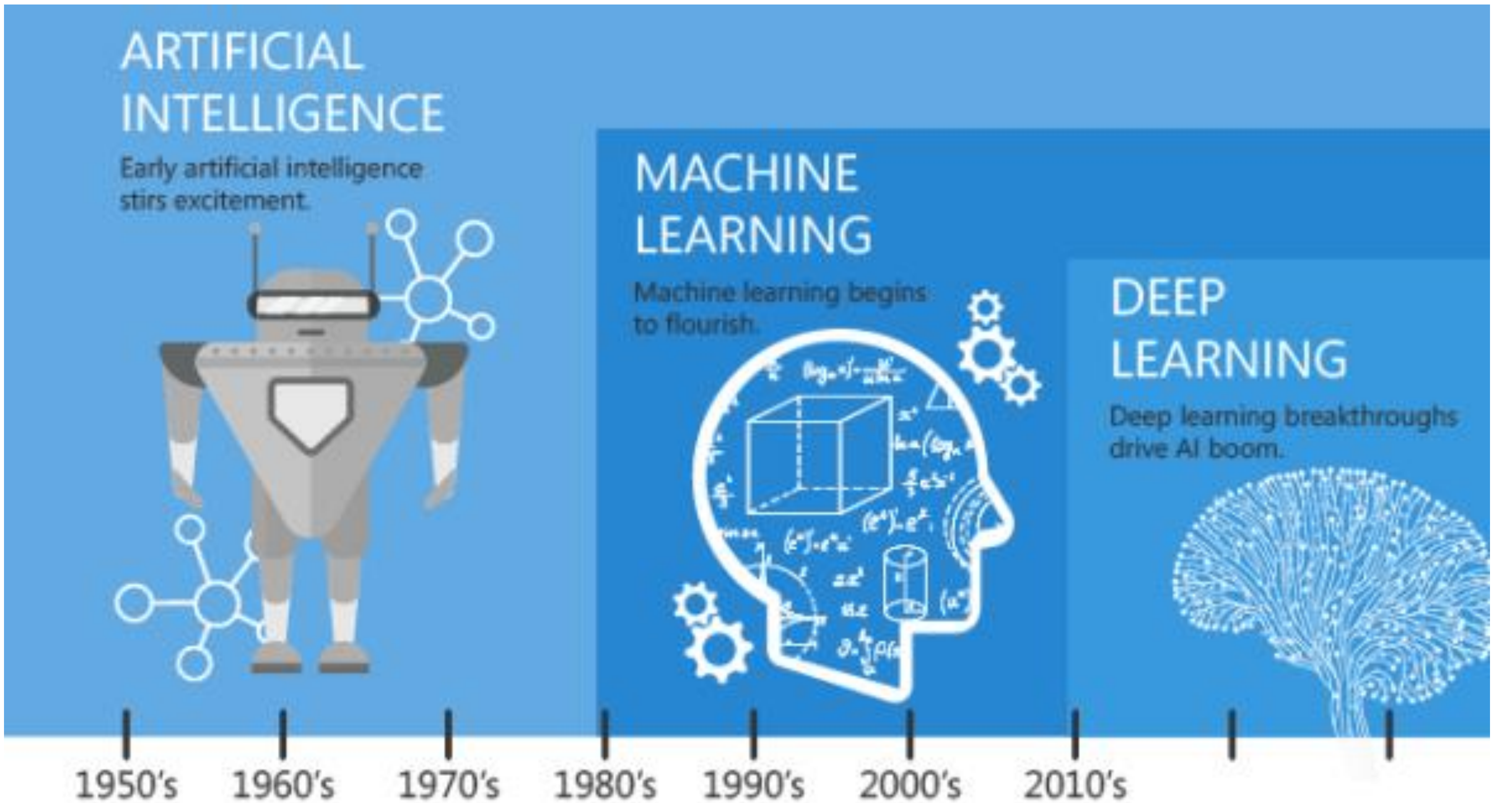
http://www.computerchess.org.uk/ccrl/4040/

IBM Watson (2011): Jeopardy kvízjáték

- **IBM** Grand Challenge
  - 1997: **Deep Blue** legyőzi a G. Kaszparov sakkvilágbajnokot.
  - 1999-2006<: **Blue Gene**, fehérje struktúra predikció
  - 2011: **Watson**
    - Beszédértés
    - Következteté
    - Játék



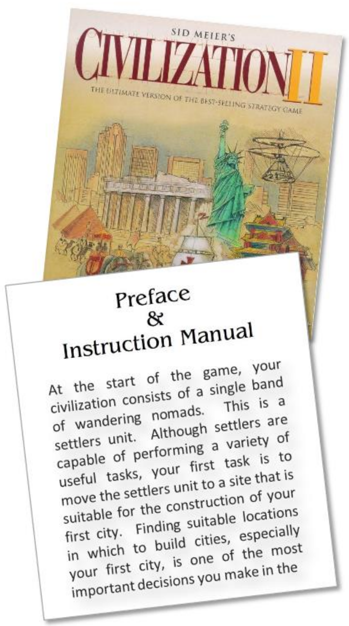
AI – ML - DL

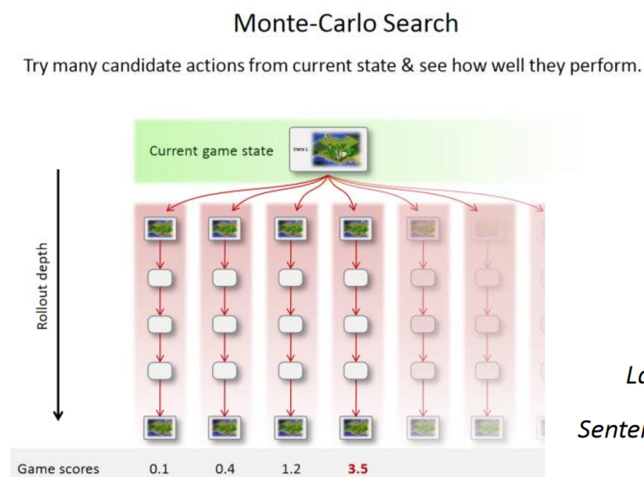


https://shawnennis.com/the-technology-of-machine-learning-with-ai

Civilization: használati útmutató használata

- Hibrid megoldás: nyelvi elemzés és gépi tanulás



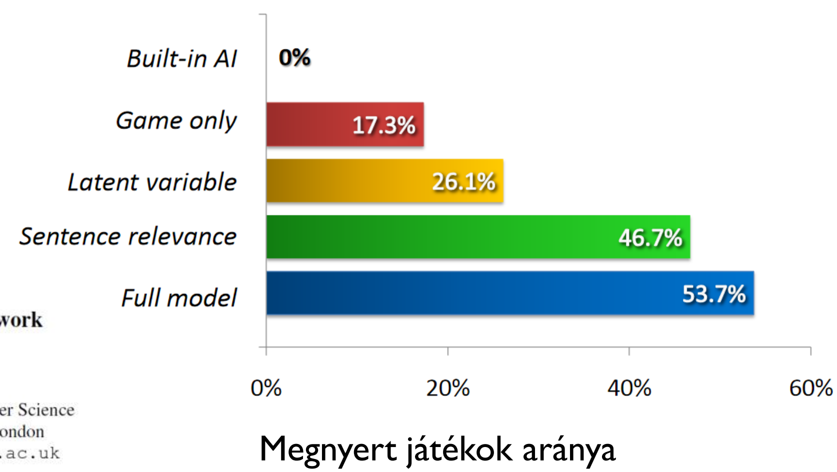


Learning to Win by Reading Manuals in a Monte-Carlo Framework

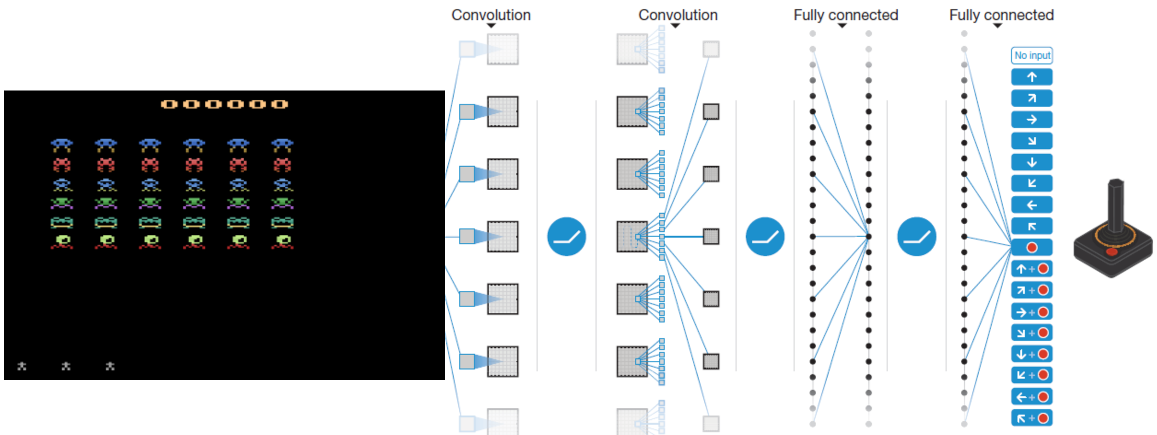
S.R.K. Branavan | David Silver * | Regina Barzilay

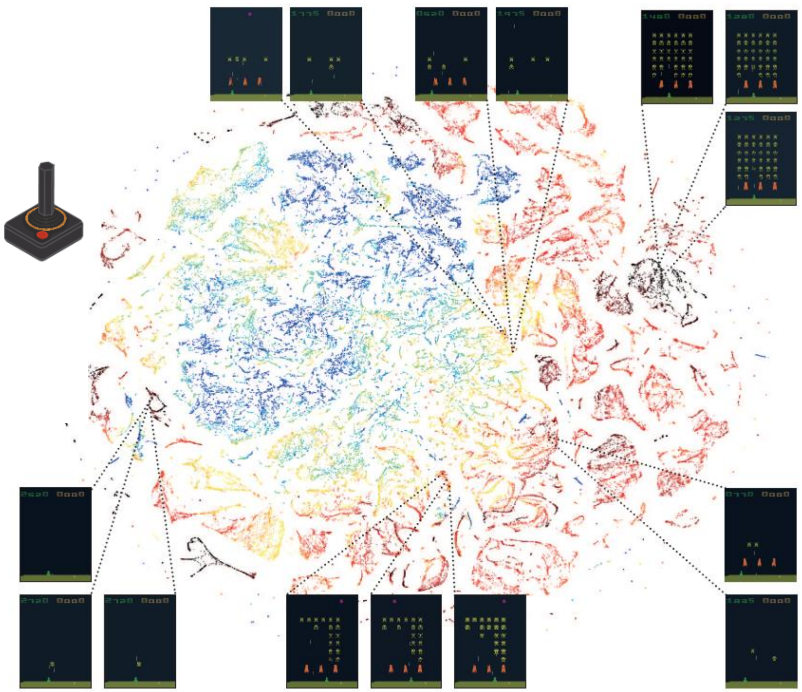
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
{branavan, regina}@csail.mit.edu

* Department of Computer Science
University College London
d.silver@cs.ucl.ac.uk



Számítógépes játékok játszása





LETTER

doi:10.1038/nature14236

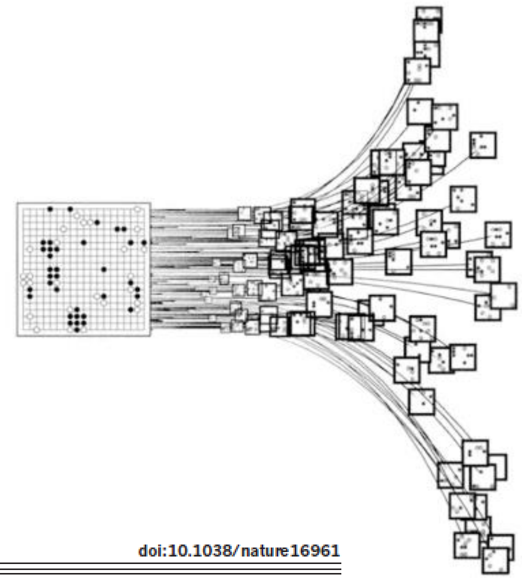
Human-level control through deep reinforcement learning

Volodymyr Mnih[1*], Koray Kavukcuoglu[1*], David Silver[1*], Andrei A. Rusu[1], Joel Veness[1], Marc G. Bellemare[1], Alex Graves[1], Martin Riedmiller[1], Andreas K. Fidjeland[1], Georg Ostrovski[1], Stig Petersen[1], Charles Beattie[1], Amir Sadik[1], Ioannis Antonoglou[1], Helen King[1], Dharshan Kumaran[1], Daan Wierstra[1], Shane Legg[1] & Demis Hassabis[1]

Go:

AlphaGo

- Google DeepMind
- Monte Carlo fakereséses technika
- 2016: 9 dan
- 2017: világbajnok legyőzése



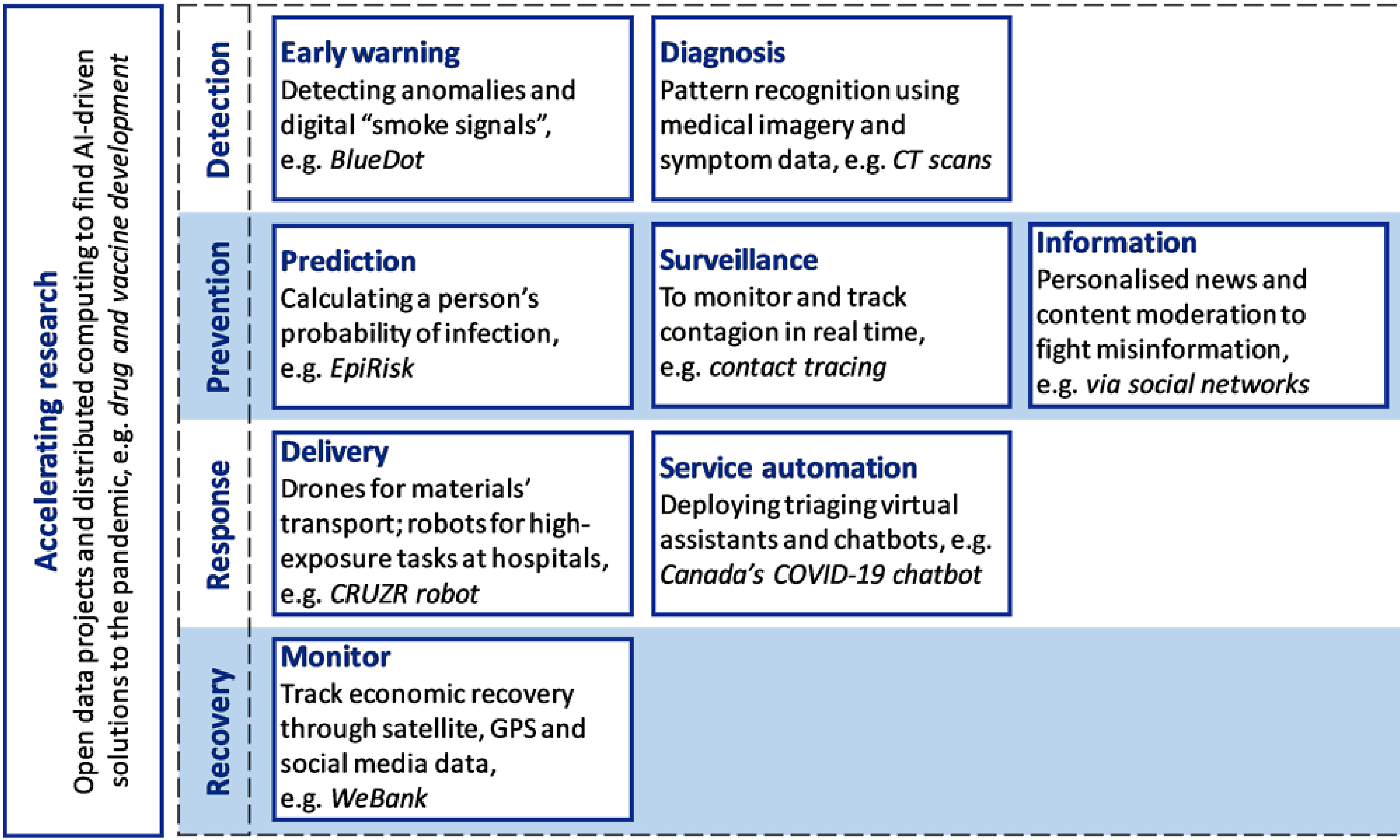
ARTICLE

doi:10.1038/nature16961

Mastering the game of Go with deep neural networks and tree search

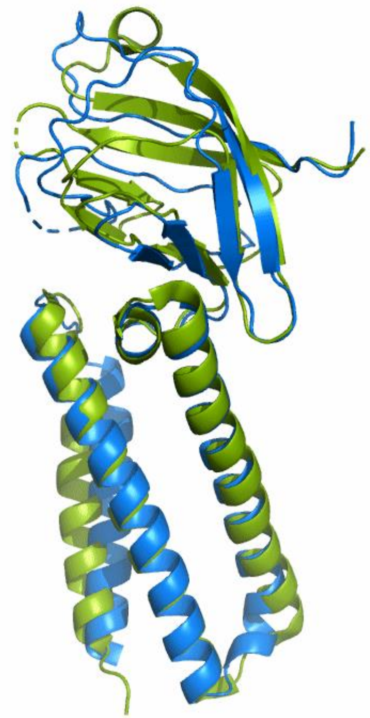
David Silver[1]*, Aja Huang[1]*, Chris J. Maddison[1], Arthur Guez[1], Laurent Sifre[1], George van den Driessche[1], Julian Schrittwieser[1], Ioannis Antonoglou[1], Veda Panneershelvam[1], Marc Lanctot[1], Sander Dieleman[1], Dominik Grewe[1], John Nham[2], Nal Kalchbrenner[1], Ilya Sutskever[2], Timothy Lillicrap[1], Madeleine Leach[1], Koray Kavukcuoglu[1], Thore Graepel[1] & Demis Hassabis[1]

COVID-19 MI alkalmazások



AlphaFold

- John Jumper, Kathryn Tunyasuvunakool, Pushmeet Kohli, Demis Hassabis, and the AlphaFold Team, "Computational predictions of protein structures associated with COVID-19", Version 3, DeepMind website, 4 August 2020, https://deepmind.com/research/open-source/computational-predictions-of-protein-structures-associated-with-COVID-19



Benevolent AI – COVID-19 Drug target

Identifying a potential coronavirus treatment using our Knowledge Graph

1-2 February 2020

Our platform surfaced potential drugs and through a triage process we identified Baricitinib, as a potential treatment to inhibit viral infection.

4 February 2020

Our research was published in The Lancet and we signed the Wellcome Trust pledge to share our research in response to the emerging global health crisis.

28 February 2020

We published a second paper examining the affinity and selectivity of all the approved drugs. This further research indicated that Baricitinib reduces endocytosis and the body's hyperinflammatory response to the virus, thus potentially reducing patient mortality rates.

10 April 2020

Eli Lilly, who owns Baricitinib, announced an agreement with the US National Institute for Allergies and Infectious Diseases (NIAID) to enter Baricitinib into clinical trials in the US, with planned expansion to additional sites in Europe and Asia. Results are expected within the next few months.

15 June 2020

Eli Lilly announced their decision to begin an additional phase 3 clinical trial of Baricitinib to treat COVID-19, planning to enrol 400 patients with data expected in the next few months.

26 June 2020

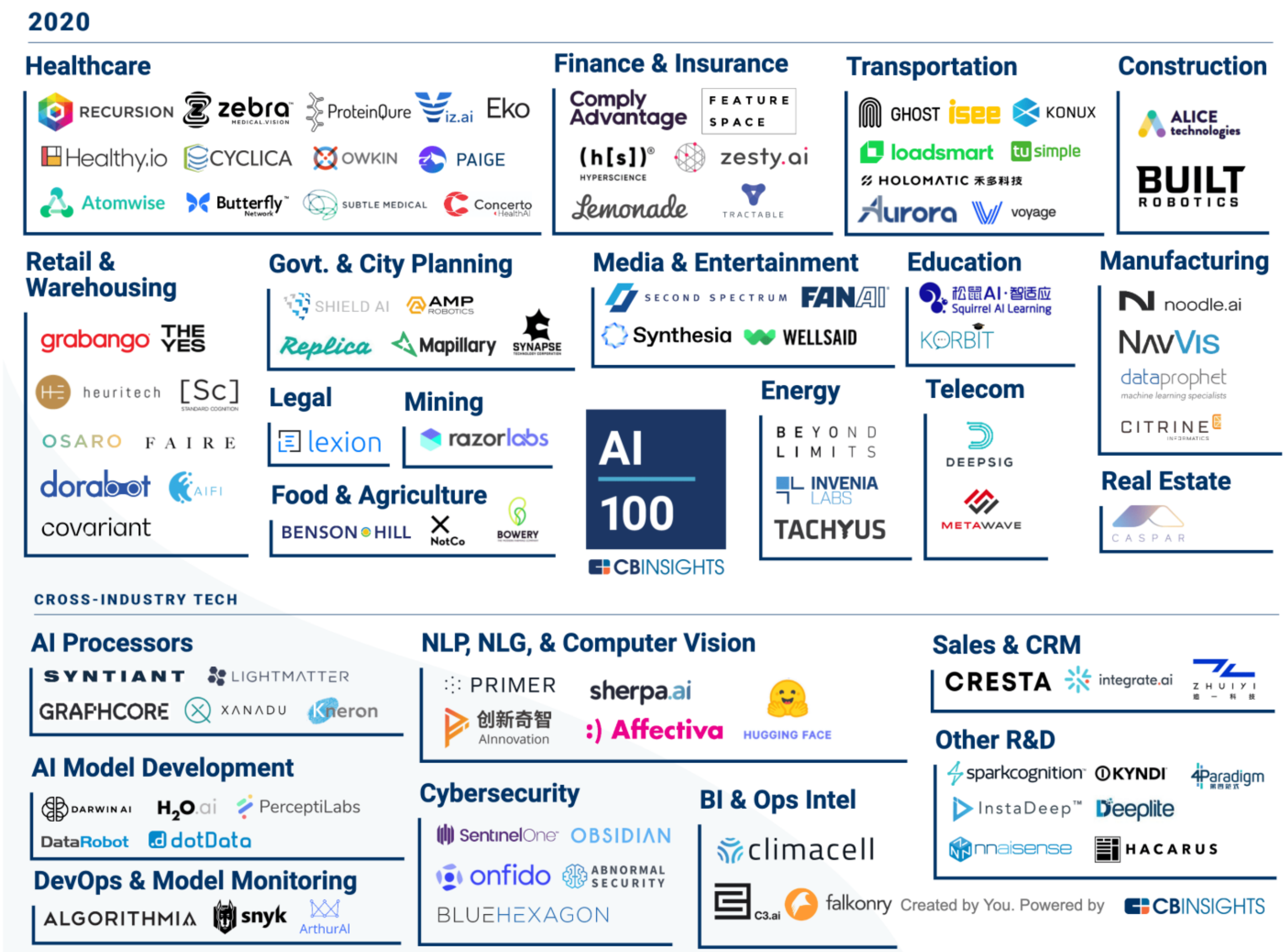
Further experimental data supporting our AI-derived hypothesis and early results from clinical testing were published in the European Molecular Biology Organization Molecular Medicine Journal. This research was conducted in partnership with Eli Lilly.

https://www.benevolent.com/covid-19

COVID-19 „Szövegbányászat”

- AWS launches machine learning enabled search capabilities for COVID-19 dataset
  Taha A. Kass-Hout, MD, MS and Ben Snively
  https://aws.amazon.com/blogs/publicsector/aws-launches-machine-learning-enabled-search-capabilities-covid-19-dataset/
- CORD-19 (COVID-19 Open Research Dataset)
- initially consisted of approximately 24,000 scientific and research sources related to COVID-19, SARS-CoV-2, and coronaviruses.
- CORD-19 dataset has nearly doubled to 47,000 research papers and documents sourced from peer-reviewed publications and pre-print servers.

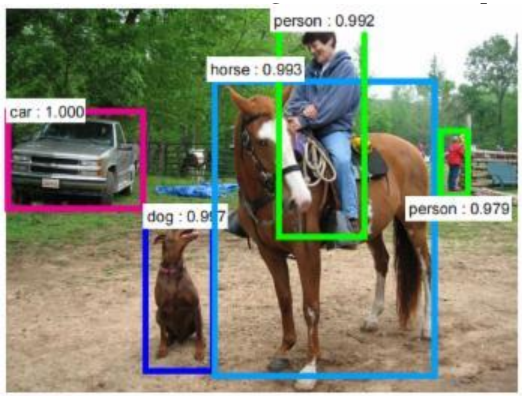
MI startup cégek 2020

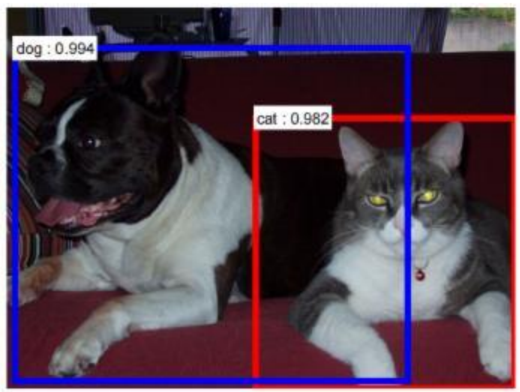


https://www.cbinsights.com/research/artificial-intelligence-top-startups/

Számítógépes látás: YOLO

- YOLO (you only look once): valós idejű, teljes körű

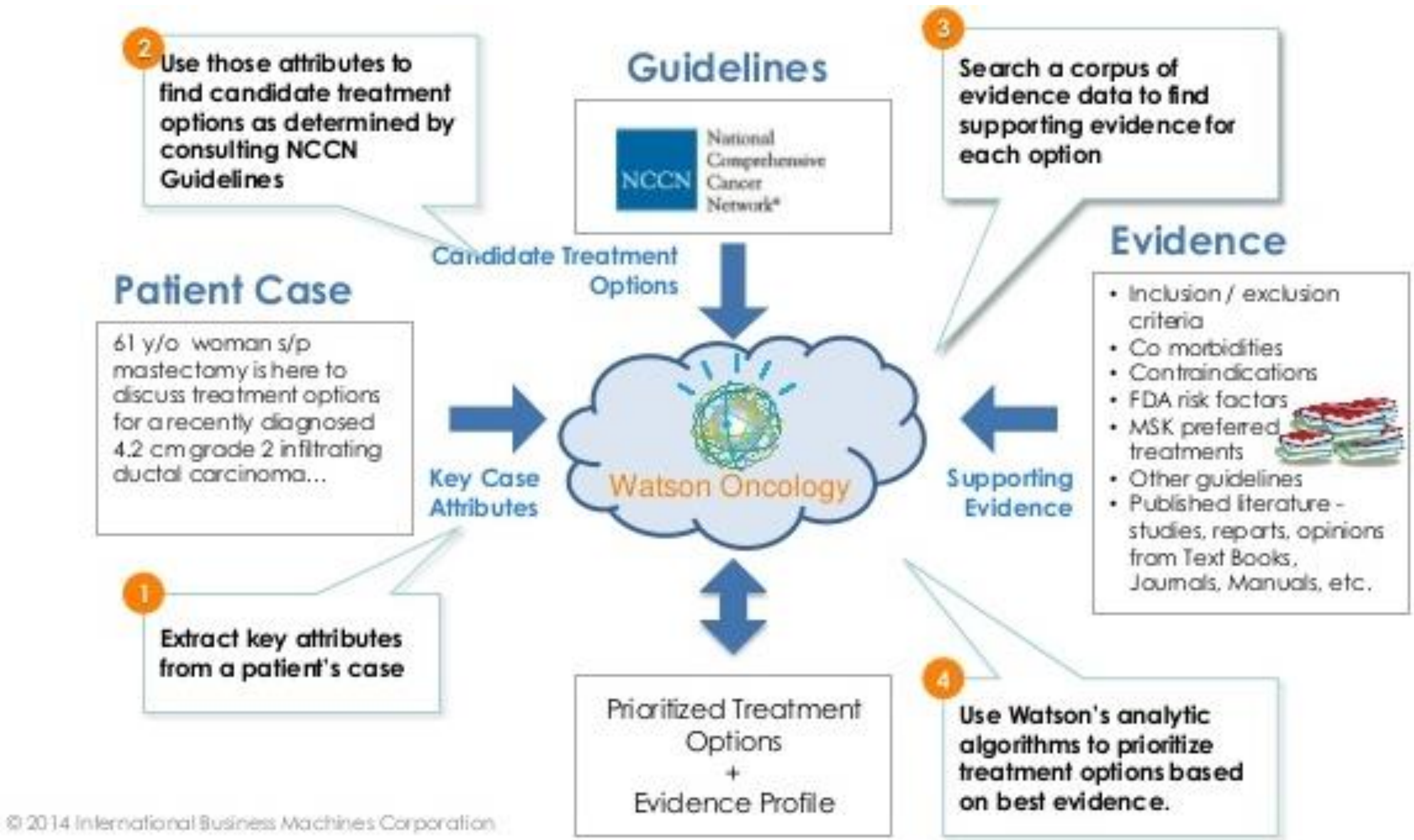




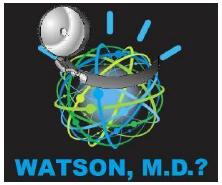
- Mobiltelefonon

https://www.ted.com/talks/joseph_redmon_how_a_computer_learns_to_recognize_objects_instantly#t-409586

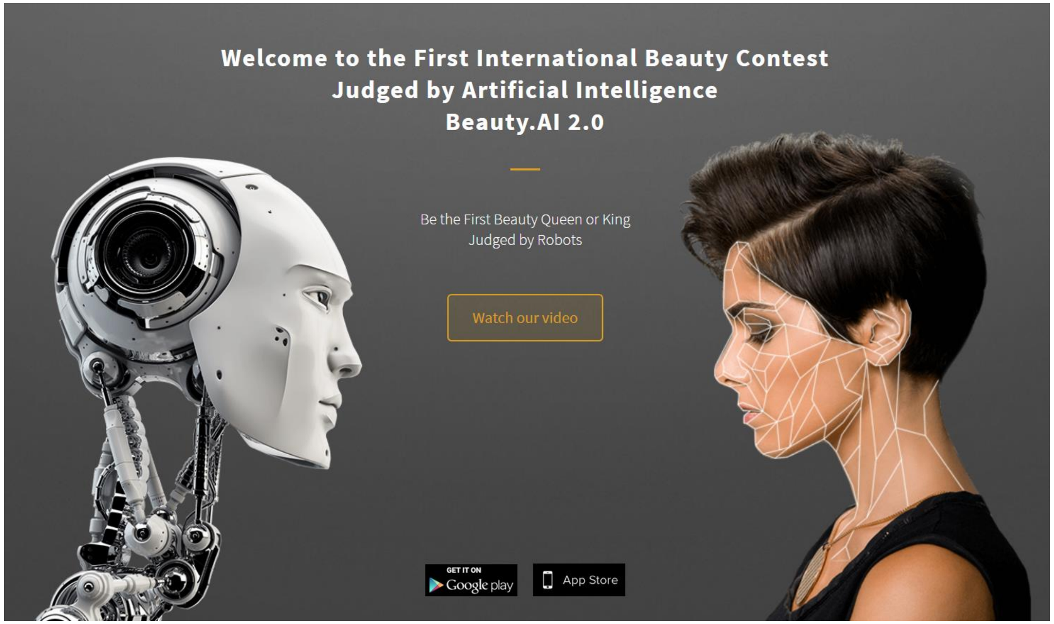
Orvosi döntéstámogató és szakértői rendszerek



Watson for Oncology – assessment and advice cycle
www.avanteoconsulting.com/machine-learning-accelerates-cancer-research-discovery-innovation/



Szépségverseny zsűri: Beauty.AI



http://beauty.ai/

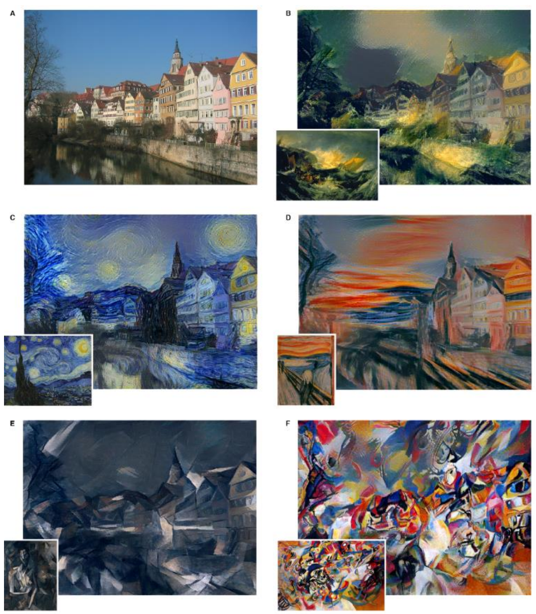
- A beauty contest was judged by AI and the robots didn't like dark skin, Guardian
- Another AI Robot Turned Racist, This Time At Beauty Contest, Unilad

MI beszélgető robot chatbot: Tay

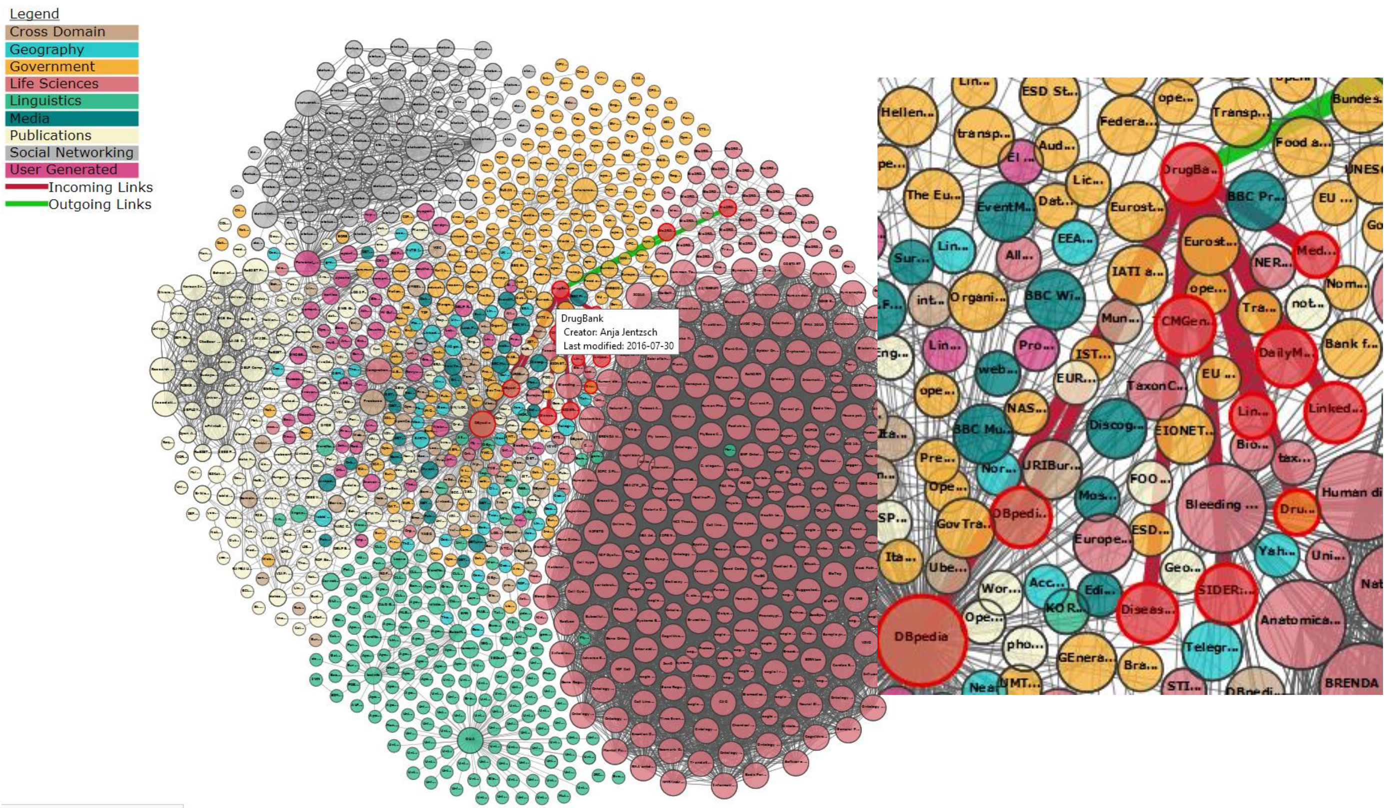
- **Turing-test, Loebner-díj**
- **Tay** was an artificial intelligence chatterbot released by Microsoft Corporation on March 23, 2016. Tay caused controversy on Twitter by releasing inflammatory tweets and it was taken offline around 16 hours after its launch.[1] Tay was accidentally reactivated on March 30, 2016, and then quickly taken offline again.

Festői stílus reprodukciója

- Gatys, L.A., Ecker, A.S. and Bethge, M., 2015. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*.

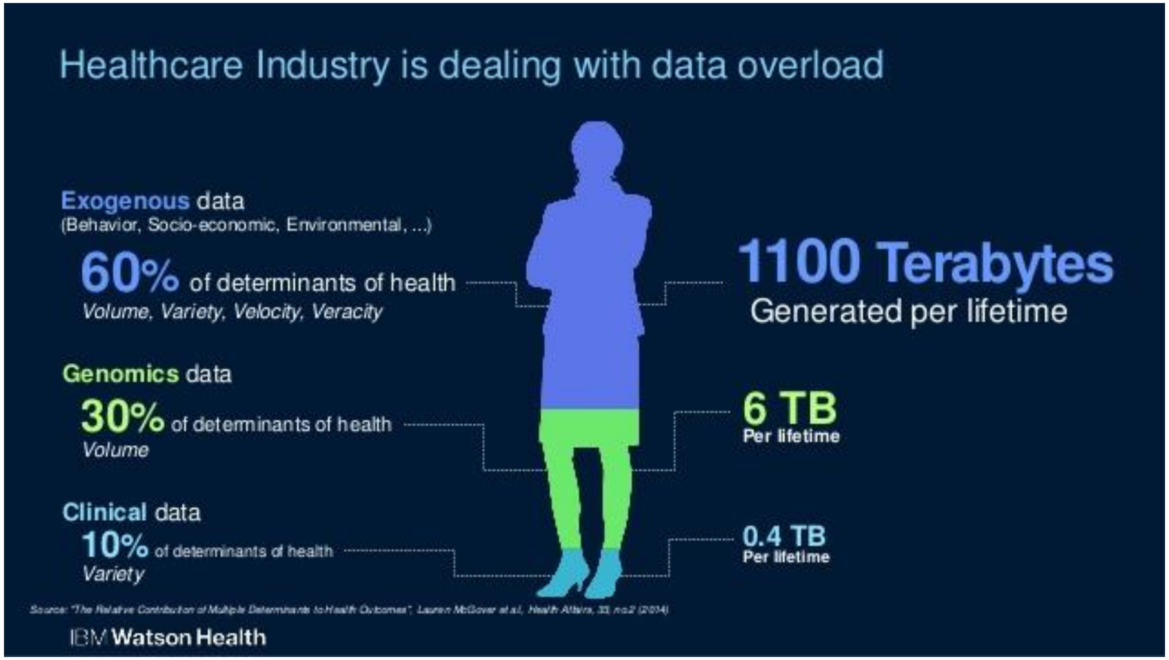


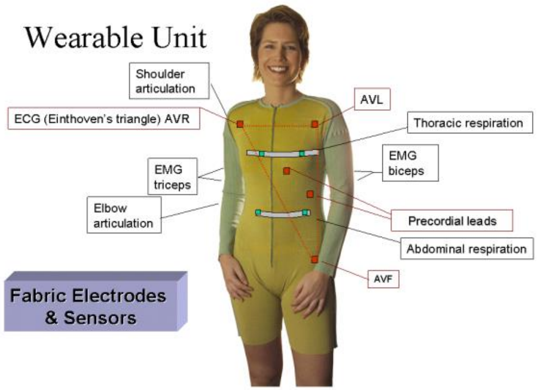
Összekapcsolt nyílt tudás (Linked Open Data)



Linking Open Data cloud diagram 2017, by Andrejs Abele, John P. McCrae, Paul Buitelaar, Anja Jentzsch and Richard Cyganiak.
http://lod-cloud.net/

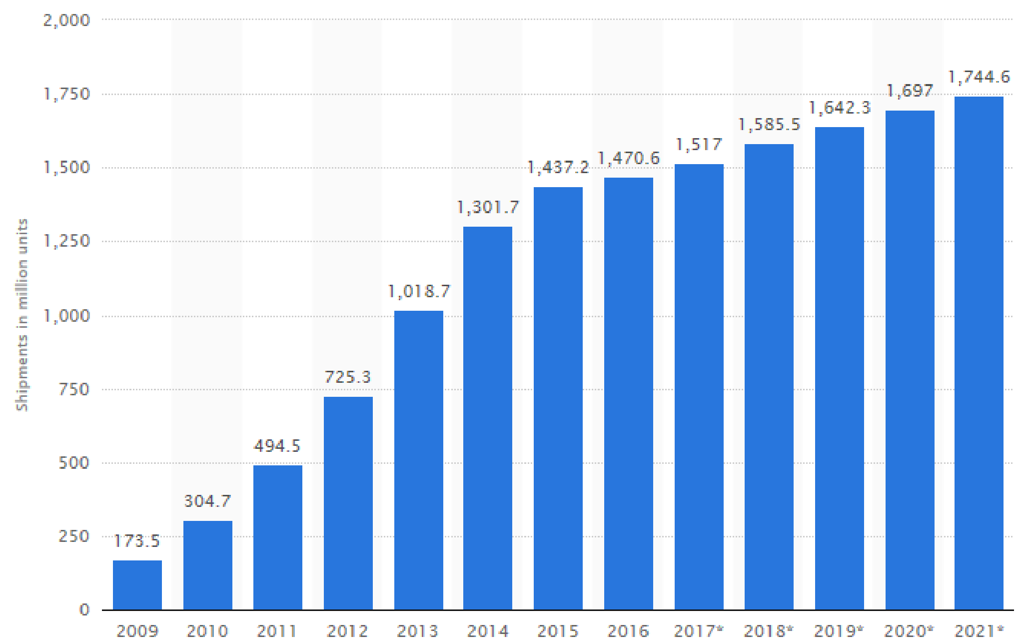
Információ (adat) mennyiség





Intelligens rendszerek a mindennapokban

- Okos-
  - telefon
  - térkép
  - óra
  - otthon
  - mérleg
  - határidőnapló
  - autó
  - társkereső
- Szakértői rendszerek:
  - Ipari szabályozás
  - Auto-pilot: repülő
  - Orvosi diagnosztika
  - gyógyszerkutatás
- „Take X and add intelligence..”



Okostelefonok száma

Témakörök

Problémamegoldás

Tudásreprezentáció
következtetés

Felügyelt
tanulás

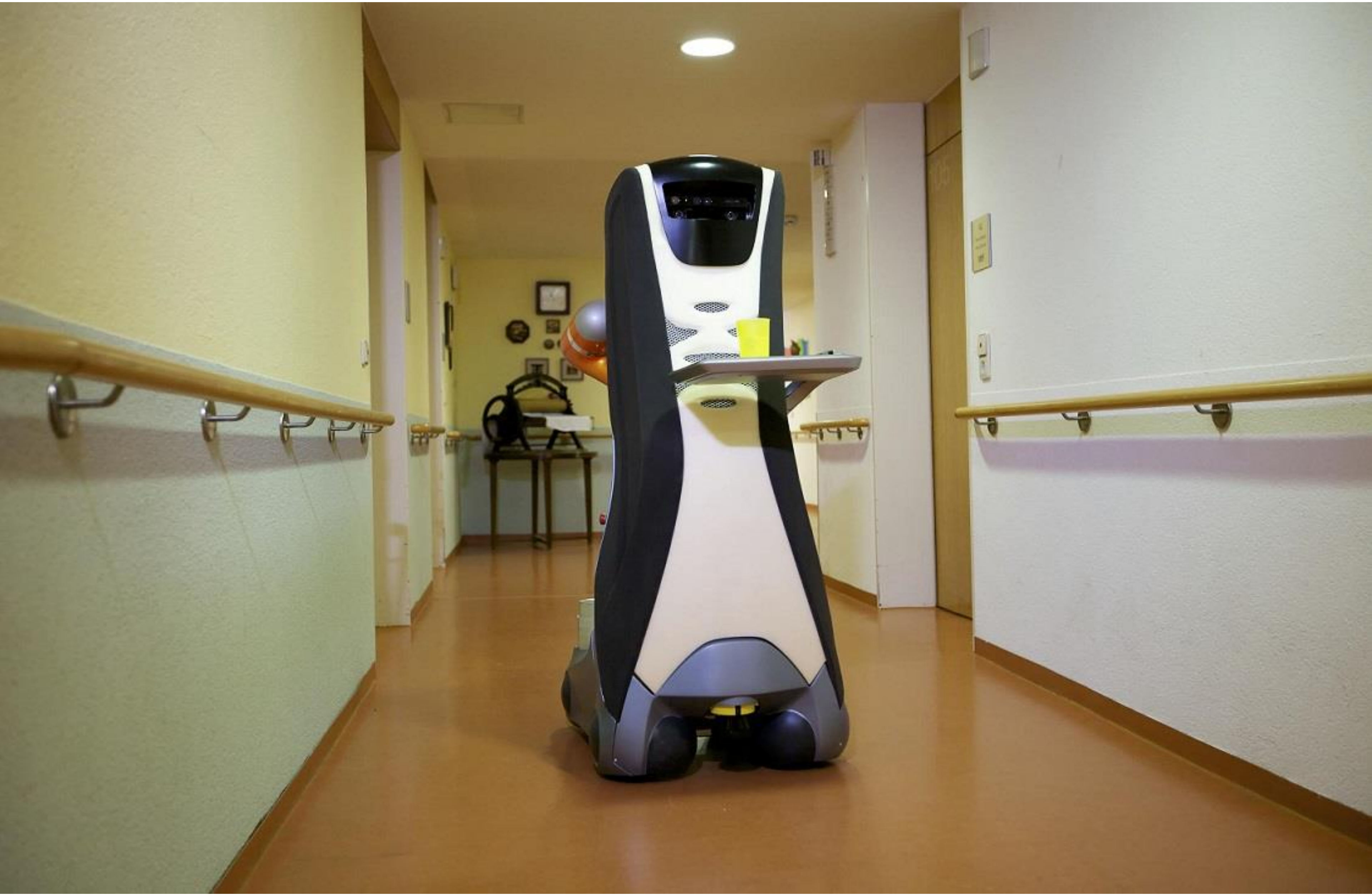
Megerősítéses
tanulás

Ajánló
rendszerek

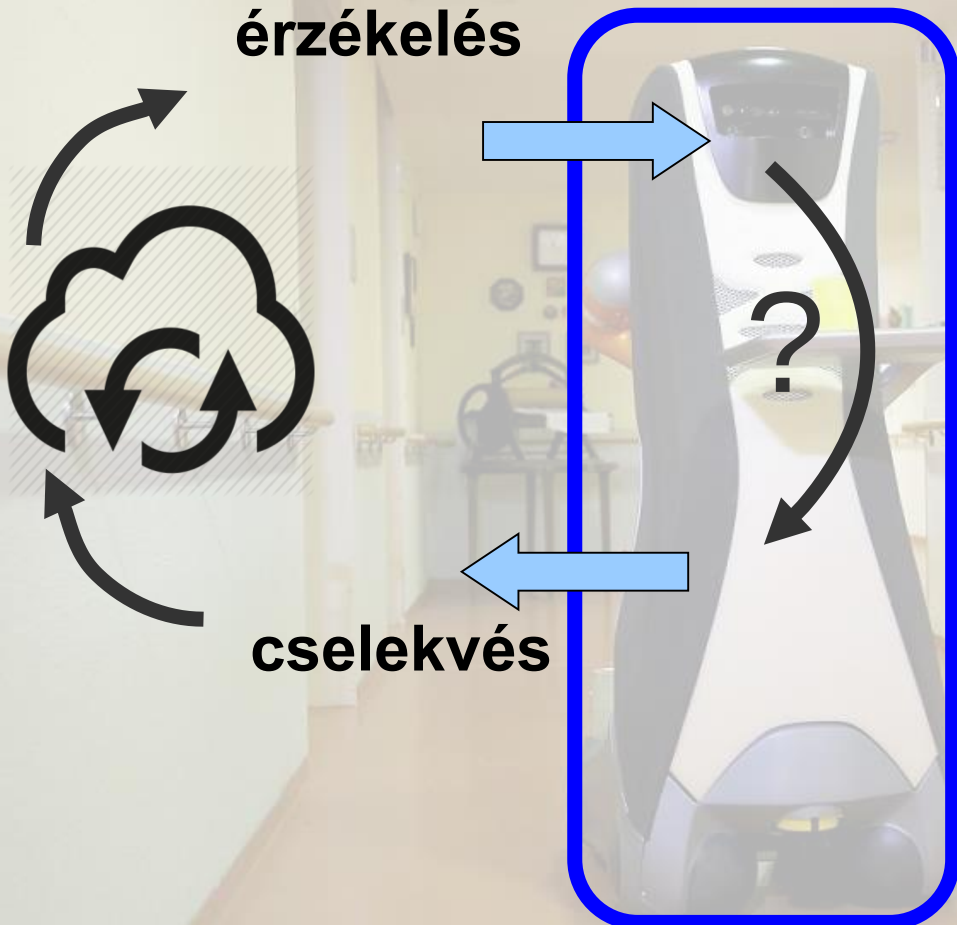
Mi ezekben a rendszerekben a közös?

- http://www.care-o-bot.de/en/care-o-bot-3.html

Fraunhofer Institute for Manufacturing Engineering and Automation IPA



- Ami kívül: az a **környezet** Ami belül: az a **rendszer**

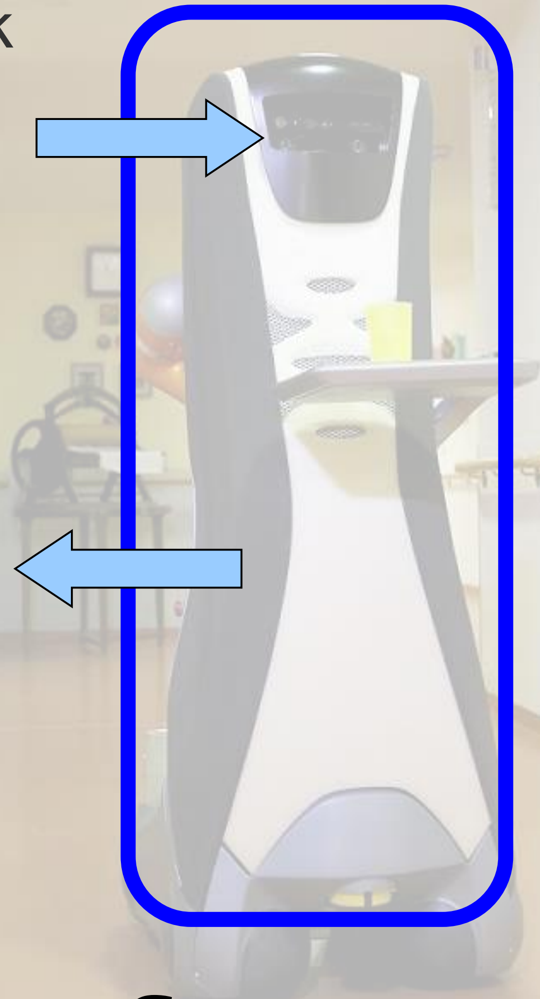


ágens

a környezetébe (fizikailag) ágyazott, vele folyamatos kölcsönhatásban lévő, ...
érzékelőivel érzékeli,
beavatkozóival megváltoztatja ...

Ágens környezetének $s_K(t)$ állapotai vannak

$$s_K(t) \in S_K$$

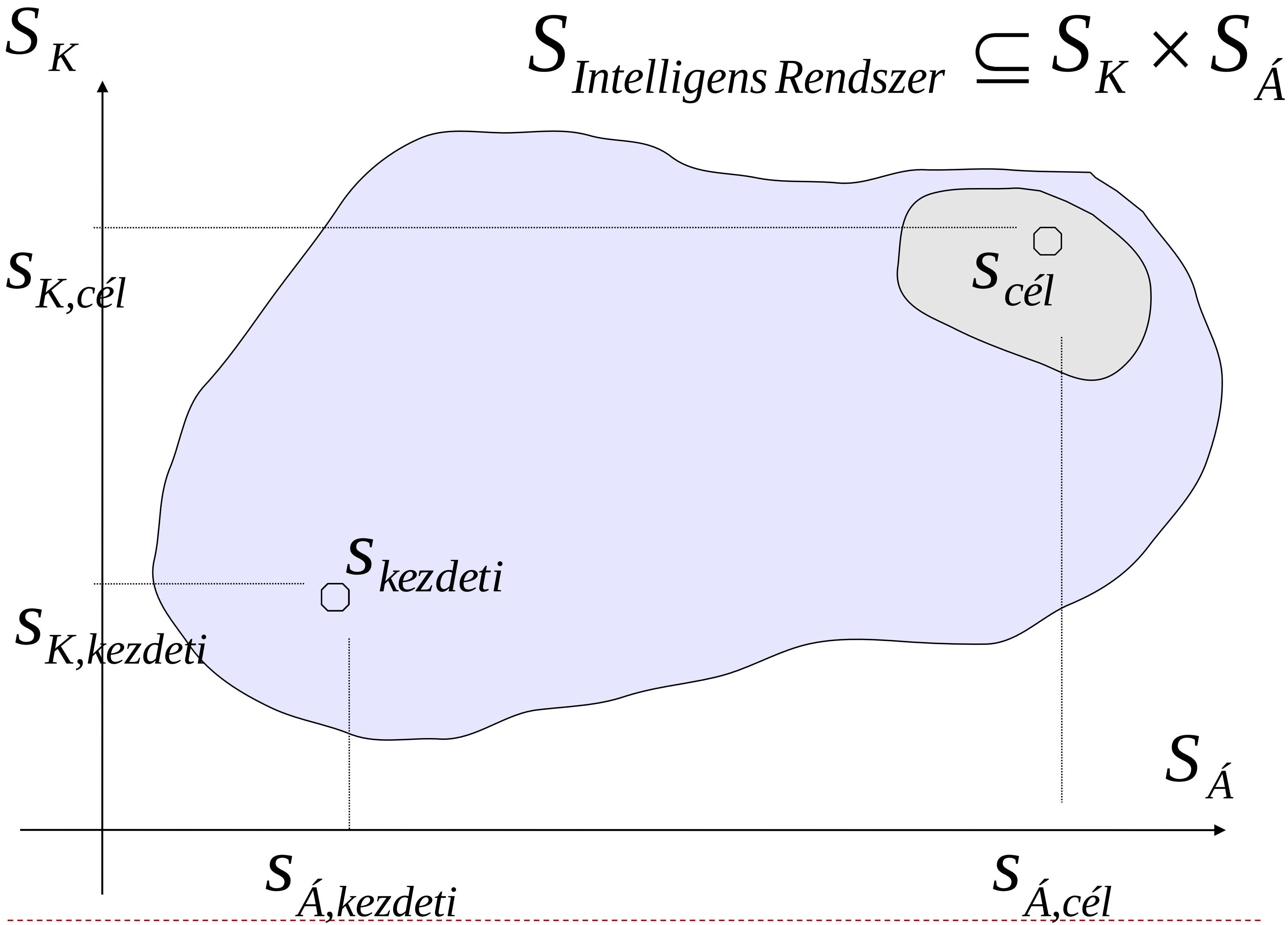


Ágensnek magának is $s_Á(t)$ állapotai vannak

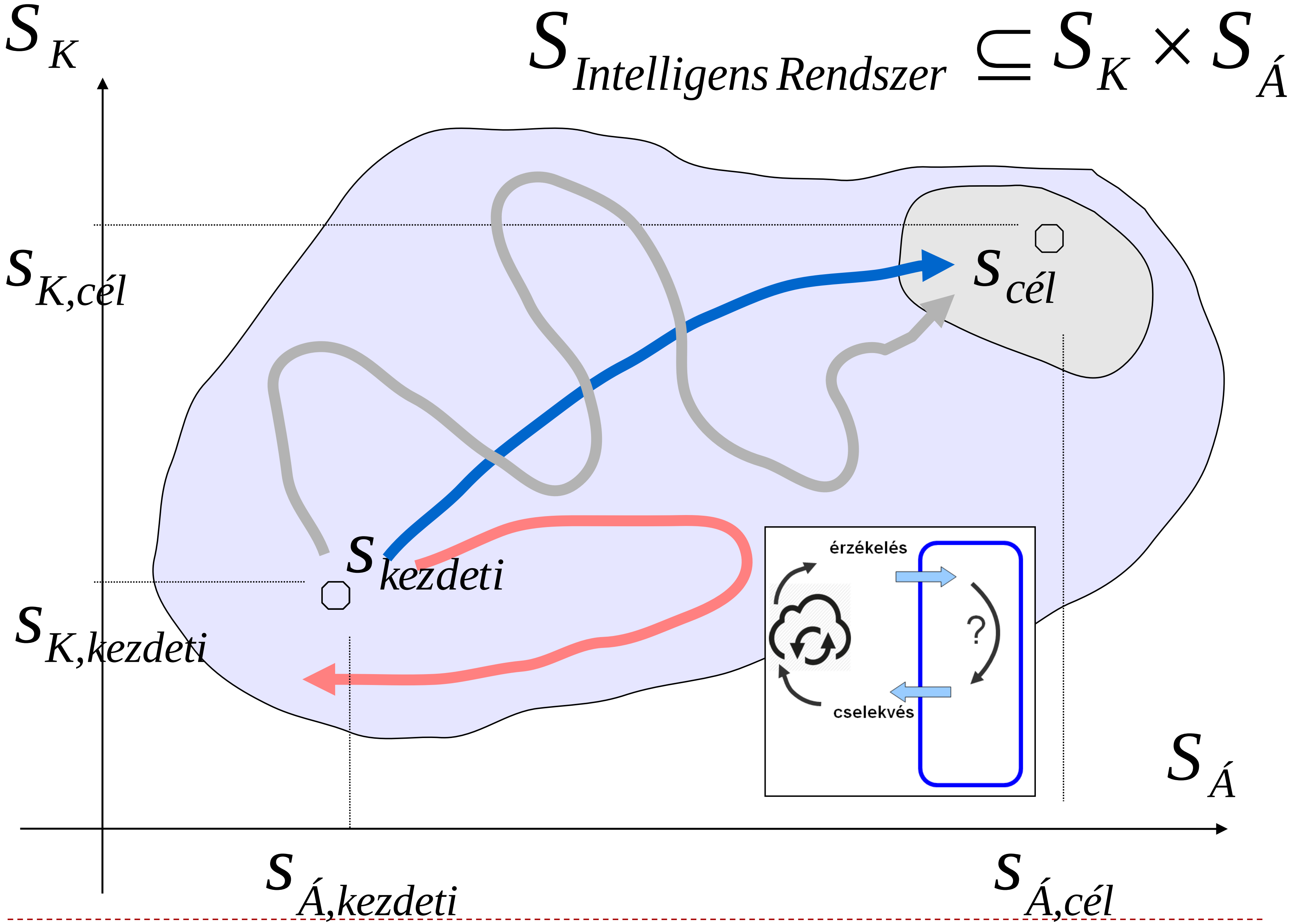
$$s_Á(t) \in S_Á$$

$$S_{Intelligens\ Rendszer} \subseteq S_K \times S_Á$$

$$S_{Intelligens\,Rendszer} \subseteq S_K \times S_Á$$



$$S_{Intelligens\ Rendszer} \subseteq S_K \times S_Á$$



Ágens

- Az ágens egy entitás, ami érzékel és cselekszik

- Absztrakt módon megfogalmazva egy függvényt valósít meg, ami az érzékeléseket képzi le cselekvésre

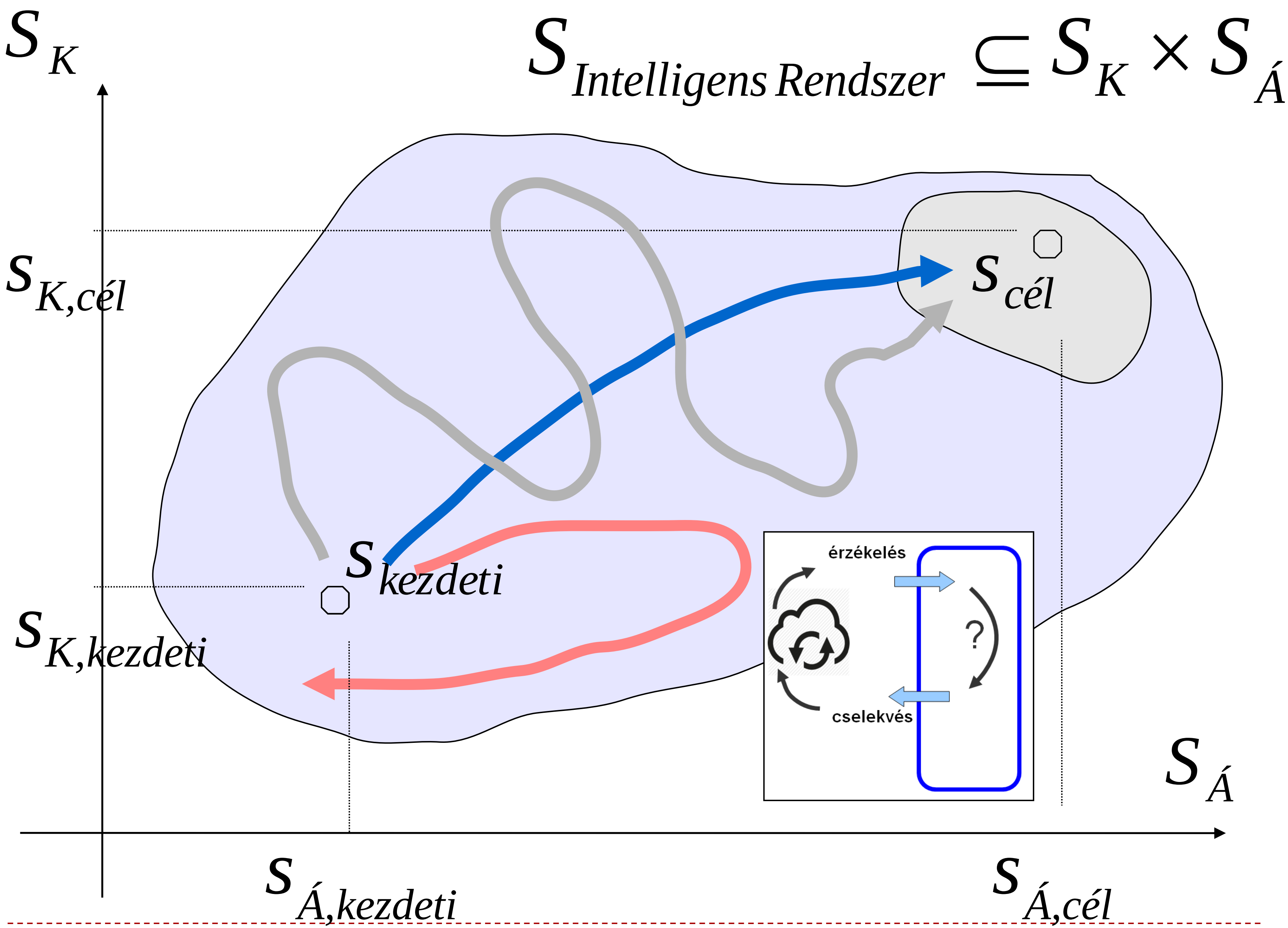
$$[f\text{: } \mathcal{P}^{\star} \rightarrow \mathcal{A}]$$

- Minden környezethez és feladat típushoz a lehető legjobb teljesítményű ágenst keressük.

Cél és trajektória

- Ágens „**célja**” a környezetének egy meghatározott állapotát elérni vagy megvalósítani, ami számára kívánatos: $\boldsymbol{S}_{\textbf{cél}}$
- Érzékelés és cselekvés közben ágens **egy trajektória** mentén halad.

- A trajektóriák nem egyformák.
- Egy „ügyes” ágens, olyan trajektóriát választ, ami „jó” vagy „hatékony” (esetleg más ágenseknél is jobb).


$S_{Intelligens\,Rendszer} \subseteq S_K \times S_Á$
S_K
$S_{K,cél}$
$S_{K,kezdeti}$
$S_{kezdeti}$
$S_{cél}$
érzékelés
?
cselekvés
$S_Á$
$S_{Á,kezdeti}$
$S_{Á,cél}$

Racionális cselekvés – intelligens ágens

- **Racionális cselekvés** = cél felé irányuló cselekvés
- **Intelligens ágens** – racionális módon választja meg a cselekvéseit és a célállapotait sikeresen éri el

- A tökéletes racionalitás lehetetlen, a számítási szükségletek túl nagyok.

- **Korlátozott racionalitás** - megfelelően cselekedni,
- miközben az összes számításra nincs elegendő idő.

Mitől függ és mennyire mérhető az intelligencia?

- Kell egy mechanizmus, hogy jó trajektórián tartsa az ágenst, amíg különbség van a pillanatnyi és a célállapot között.

- Ágens feladata **érzékelésből „kiszámítani” a cselekvést**, de
  - érzékelések függenek az érzékelőktől,
  - cselekvések függenek a beavatkozóktól,
  - a számítás módja függ az ágens felépítésétől.

- A cselekvés kiszámításának „ügyessége” kapcsolatba hozható a rendszer intelligenciájával.

Környezet hatása az intelligenciára

- **A szükséges intelligenciát befolyásolja a környezet**

hozzáférhető	**nem hozzáférhető**
determinisztikus	**nem determinisztikus**
epizódszerű	**nem epizódszerű**
statikus	**dinamikus**
diszkrét	**folytonos**
egy ágens	**több ágens**
kooperatív	**versengő**

- A legnehezebb a **nem hozzáférhető**, **nem epizódszerű**, **dinamikus**, **nem determinisztikus**, és **folytonos, többágenses** környezet.

- „Nehezebb” környezet „összetettebb” intelligenciát igényel

Környezet hatása az intelligenciára

- **A szükséges intelligenciát befolyásolja a környezet**

hozzáférhető	**nem hozzáférhető**
determinisztikus	**nem determinisztikus**
epizódszerű	**nem epizódszerű**
statikus	**dinamikus**
diszkrét	**folytonos**
egy ágens	**több ágens**
kooperatív	**versengő**

- *A valós helyzetek legtöbbje olyan bonyolult, hogy gyakorlati okokból nem determinisztikusként kezelendők.*

- **Ágens „ellenségei”** (amiktől az intelligenciája korlátos, vagy romlandó)
  (1) **véges erőforrásai (rendelkezésre álló időt is beleértve)**
  (2) **információhiány érzékeléskor**
  (3) **a környezet változékonysága**

Összefoglalásként

Informatikában:

- Az intelligencia egy tervezhető és skálázható rendszer-attribútum.

- intelligencia révén igényes és újszerű szolgáltatásokat valósítunk meg.

- egy informatikusnak tudnia kell tervezéskor a rendszer intelligenciájával gazdálkodni.

Mesterséges intelligencia előadássorozat

Az előadás diái az AIMA könyvre épülve (http://aima.cs.berkeley.edu) készültek a University of California, Berkeley mesterséges intelligencia kurzusának anyagainak felhasználásával (http://ai.berkeley.edu).

These slides are based on the AIMA book (http://aima.cs.berkeley.edu) and were adapted from the AI course material of University of California, Berkeley (http://ai.berkeley.edu).

Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Méréstechnika és Információs rendszerek Tanszék

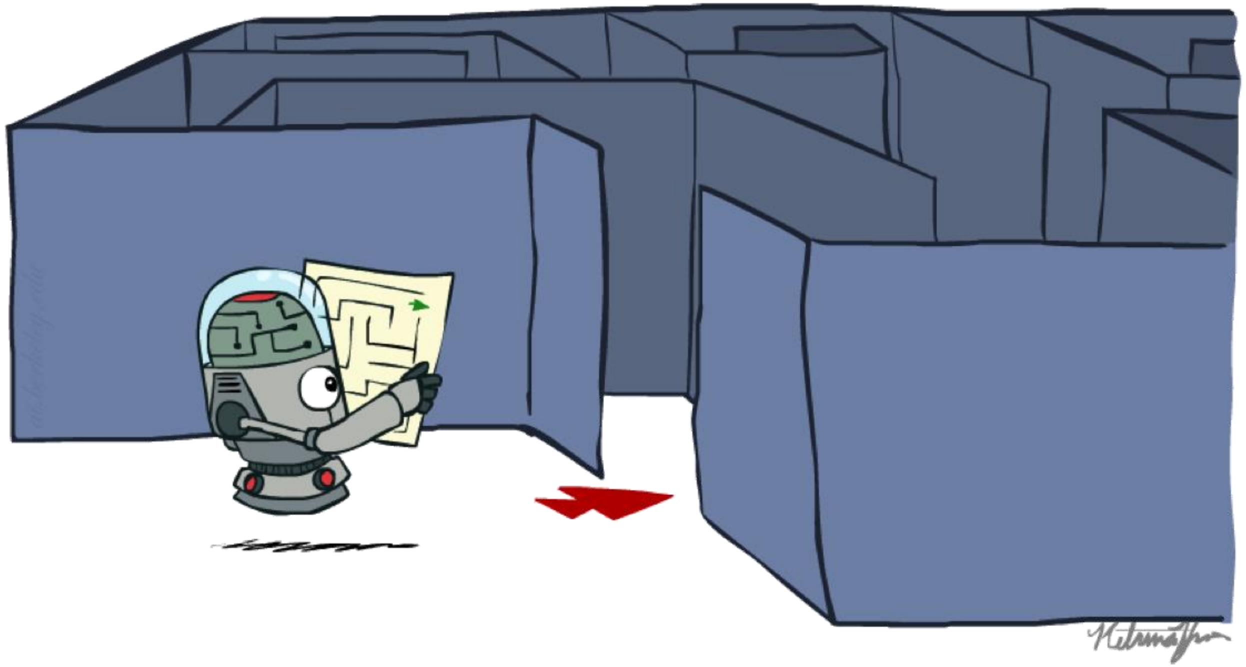
Mesterséges intelligencia

Problémamegoldás kereséssel

Előadó: Dr. Hullám Gábor

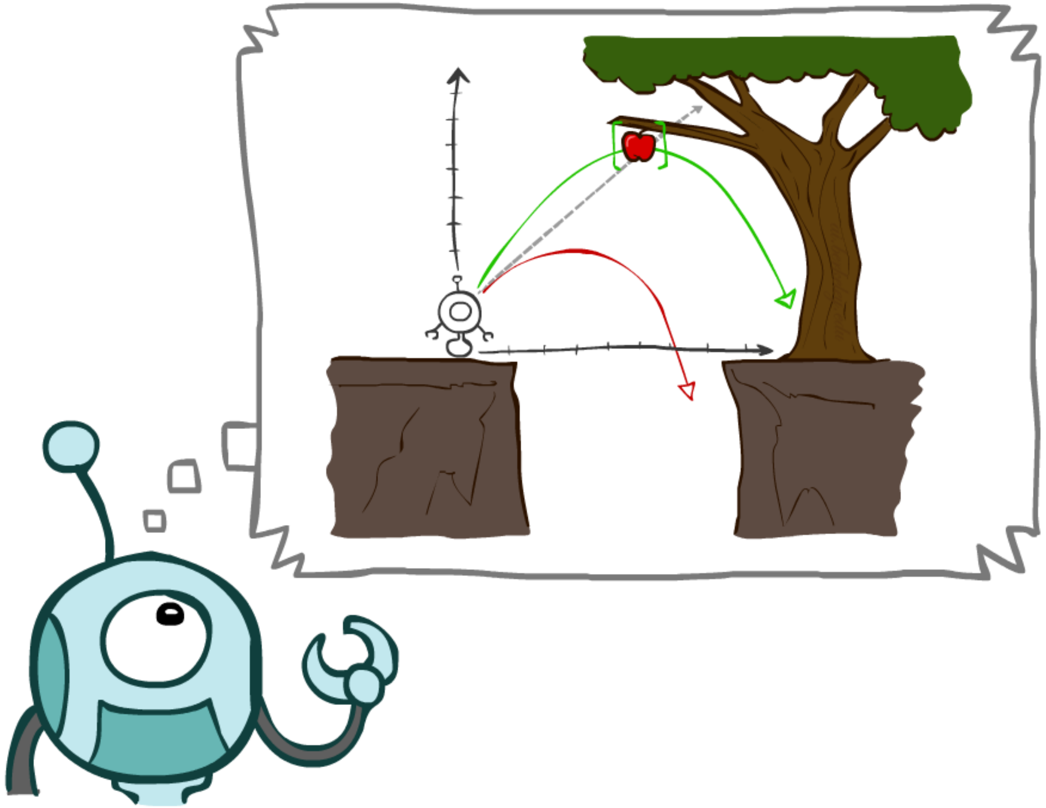
Problémamegoldás kereséssel

Nem informált keresés

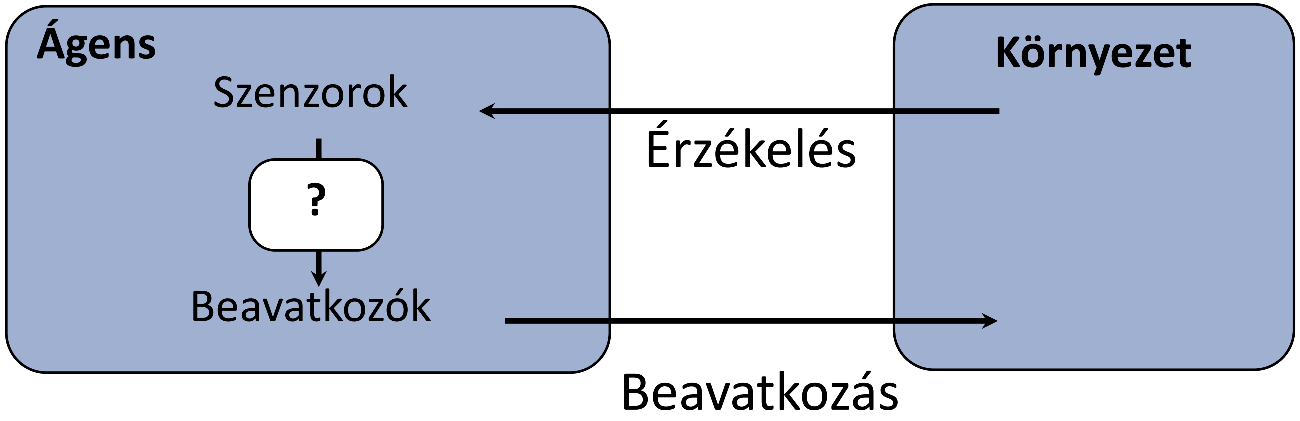


Tartalom

- Ágensek és a tervkészítés
- Keresési problémák
- Nem informált keresési módszerek
  - Mélységi keresés (Depth-First Search)
  - Szélességi keresés (Breadth-First Search)
  - Egyenletes költségű keresés (Uniform-Cost Search)



Ágens és a környezet



- Az ágens ***szenzorokon*** keresztül érzékeli a környezetet és
- ***beavatkozó szerv***eken keresztül cselekszik, azaz befolyásolja a környezetet

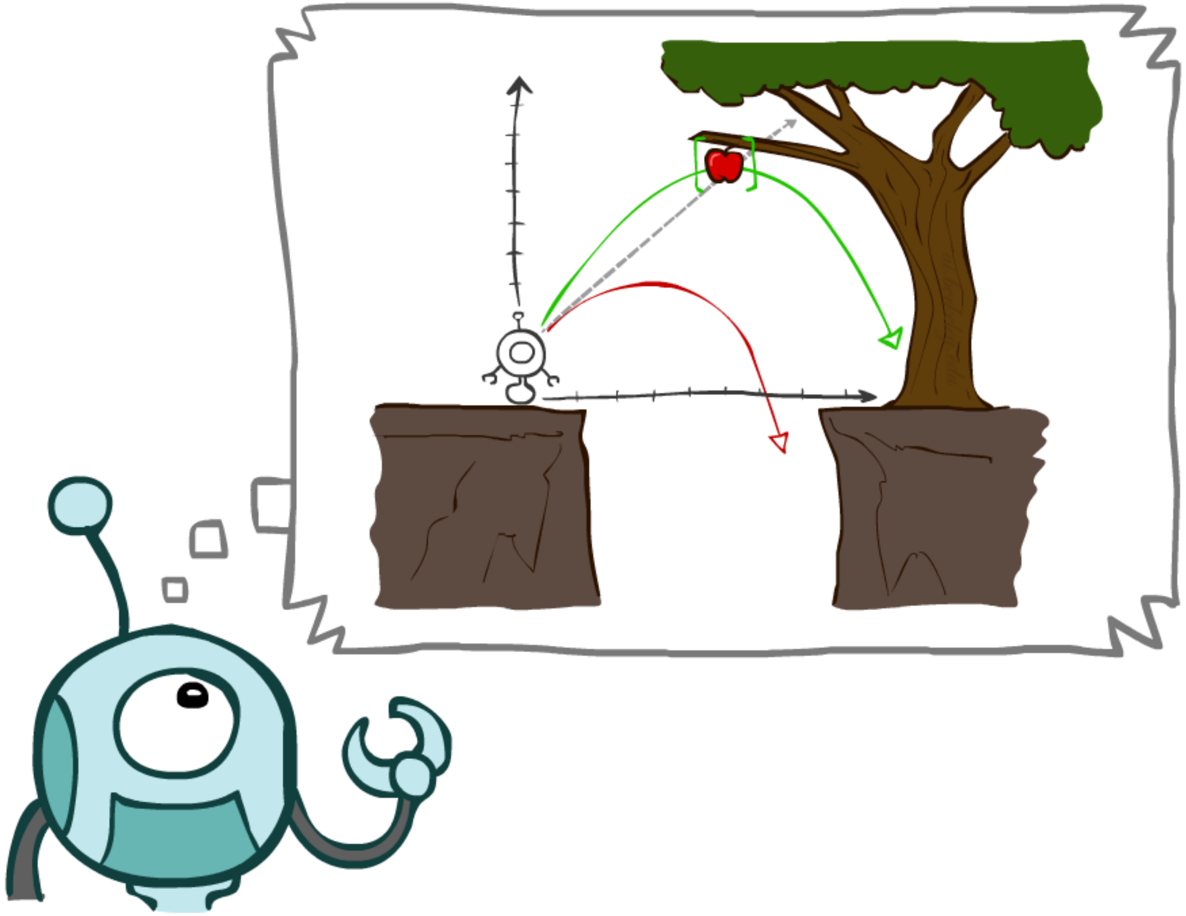
Racionalitás

- A ***racionális ágens*** úgy választ a lehetséges cselekvések közül, hogy általuk maximalizálja a ***várható hasznosságot***
  - Egyszerű eset: az ágens rendelkezik céllal és ismeri a költségeket
    - Például a cél a lehető legalacsonyabb költséggel elérni a célállapotot
  - Komplex eset: az ágens rendelkezik egy hasznosság függvénnyel, ismeri a várható jutalmakat, stb.
    - Az ágens olyan cselekvéseket hajt végre, melyek maximalizálják a teljes jutalmat adott idő alatt (például az elérhető nyereséget)

Ágensek tervezése

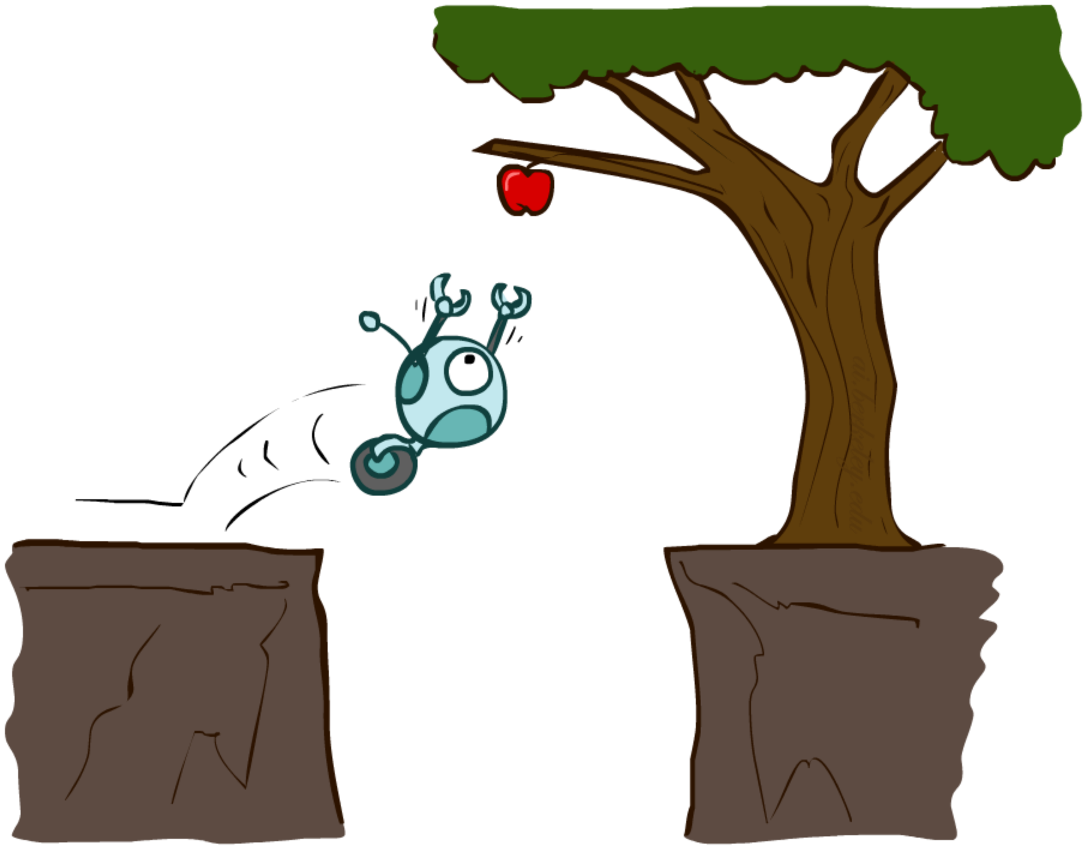
- **A környezet nagy mértékben befolyásolja az alkalmazandó ágens kialakítását, elvárt képességeit**
- **A környezet lehet…**
  - ***Teljesen/részlegesen megfigyelhető***=> az ágensnek szüksége lehet ***memóriára*** (belső állapot nyilvántartására)
  - ***Diszkrét/folytonos*** => az ágens lehet, hogy nem képes az összes lehetséges állapot megkülönböztetésére
  - ***Sztochasztikus/determinisztikus*** => az ágensnek lehet, hogy több lehetséges ***forgatókönyvet*** kell kidolgoznia
  - ***Egyedüli ágens/ Több ágens*** => lehet, hogy az ágensnek ***véletlenszerűen*** kell viselkednie

Tervkészítő ágensek



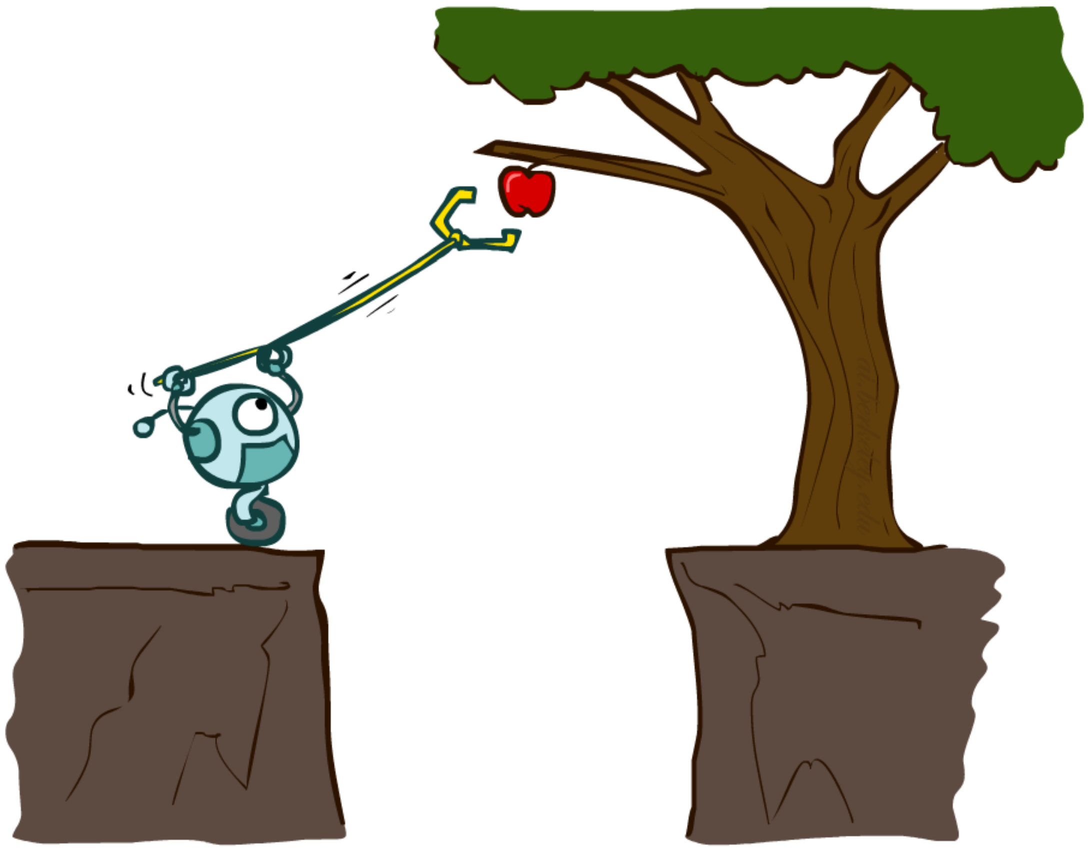
Reflex Ágens

- Reflex ágens:
  - Az aktuális érzékelés (esetleg memória) alapján választ cselekvést
  - A környező világ állapotát nyilvántarthatja a memóriájában vagy egy belső modell segítségével
  - Nem veszi figyelembe a cselekvések jövőbeli következményeit
  - A világot úgy tekinti, ahogy van
- Lehet-e egy reflex ágens racionális?



Tervkészítő ágensek

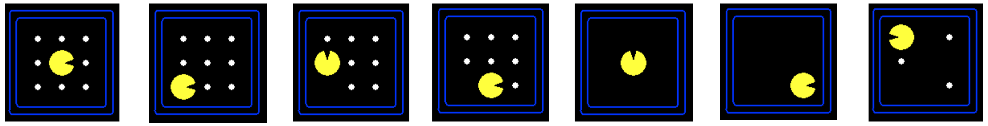
- Tervkészítő ágensek:
  - “Mi lenne ha?” kérdések feltevése
  - Döntést a cselekvések lehetséges következményeinek függvényében hoz.
  - Ehhez rendelkeznie kell egy modellel, ami leírja, hogy a világ hogyan változik a cselekvések hatására
  - Meg kell határoznia egy célt (tesztelhető célfüggvény)
  - Figyelembe veszi, hogy a világ milyen lehet / milyenné válhat



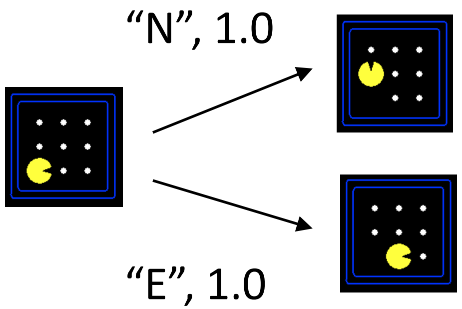
Keresési problémák

Keresési problémák

- Egy keresési probléma az alábbi elemeket tartalmazza:
  - Állapottér



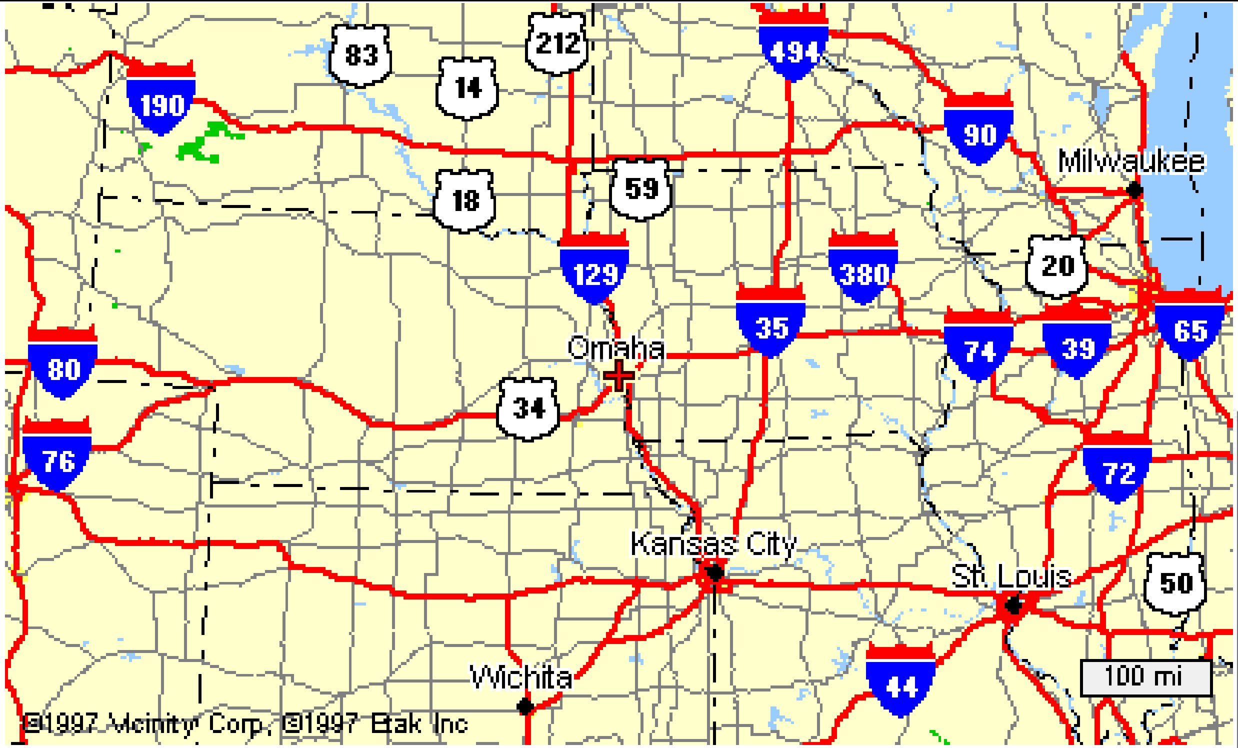
  - Állapotátmenet-függvény (cselekvések, költségek)

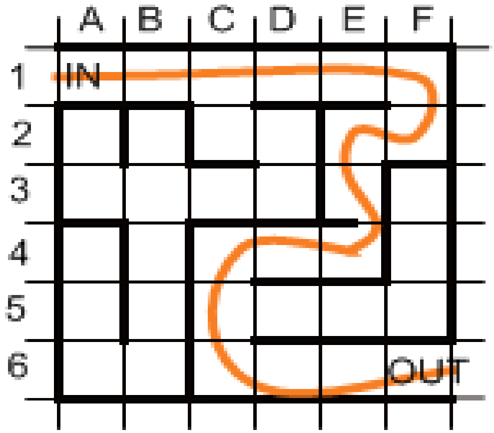


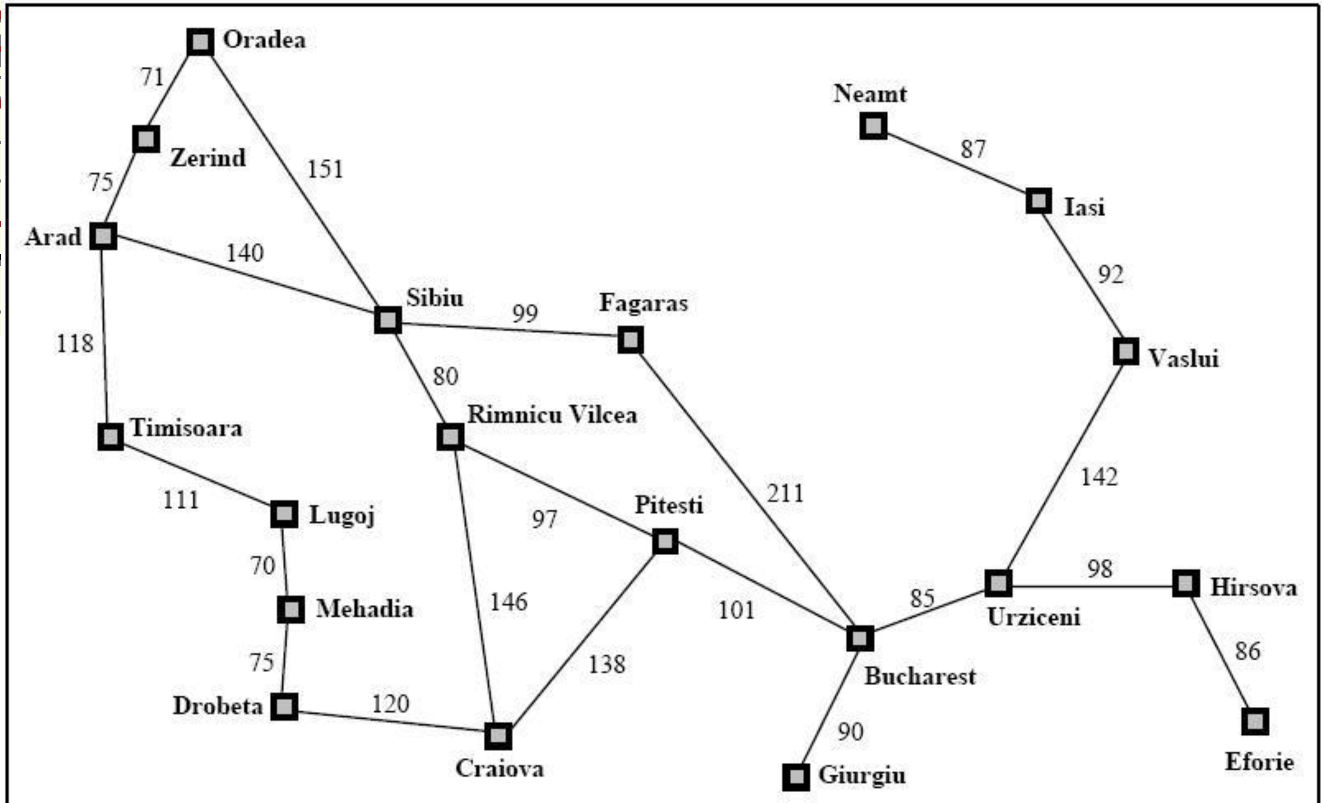
  - Kiindulási állapot és egy célállapot teszt

- A megoldás a cselekvések egy sorozata (egy terv) , amely a kiindulási állapotból eljuttat egy célállapotba (állapottranszformációs lépések révén)

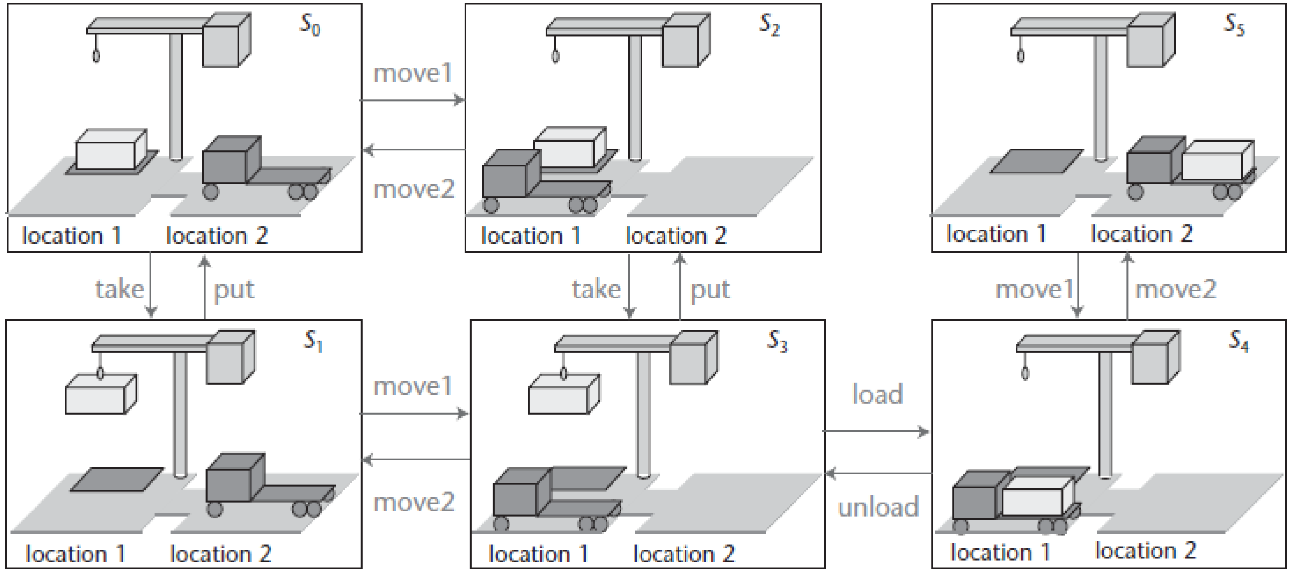
Fizikai tér

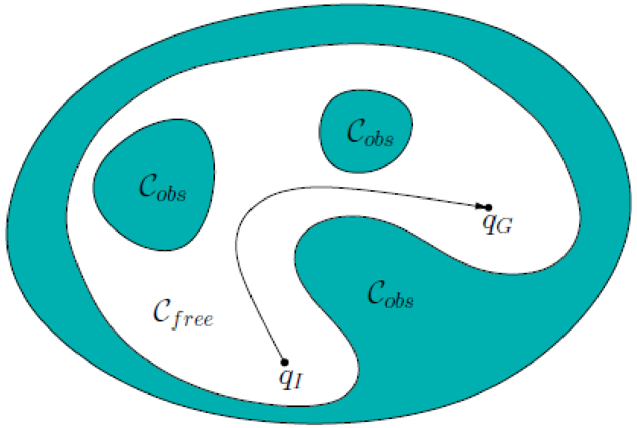

190
83
14
212
494
90
Milwaukee
18
59
129
380
20
35
74
39
65
80
Omaha
34
76
72
Kansas City
St. Louis
50
Wichita
44
100 mi
©1997 Vicinity Corp, ©1997 Etak Inc

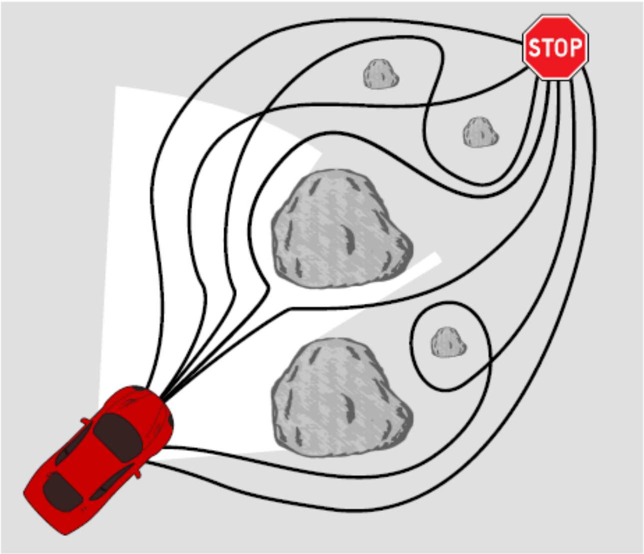

A
B
C
D
E
F
1
2
3
4
5
6
IN
OUT

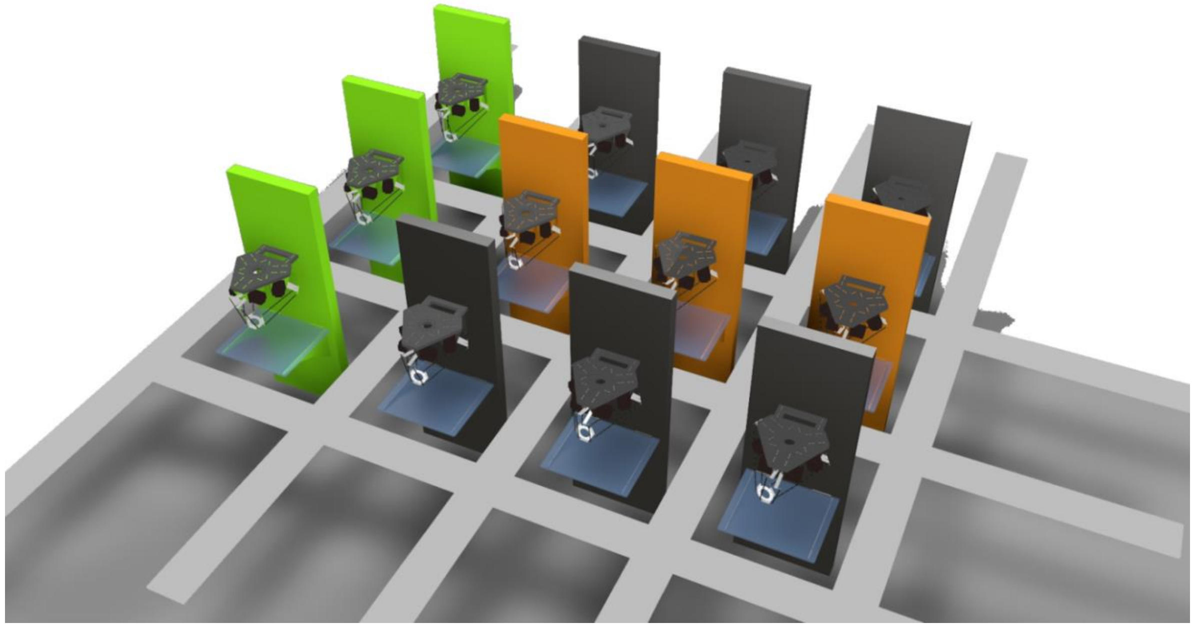

Oradea
71
Zerind
151
75
Arad
140
118
Sibiu
99
Fagaras
80
Rimnicu Vilcea
Timisoara
111
Lugoj
97
Pitesti
211
70
Mehadia
146
101
75
138
Drobeta
120
Craiova
Neamt
87
Iasi
92
Vaslui
142
Urziceni
85
98
Hirsova
86
Bucharest
90
Giurgiu
Eforie

Konfigurációs tér

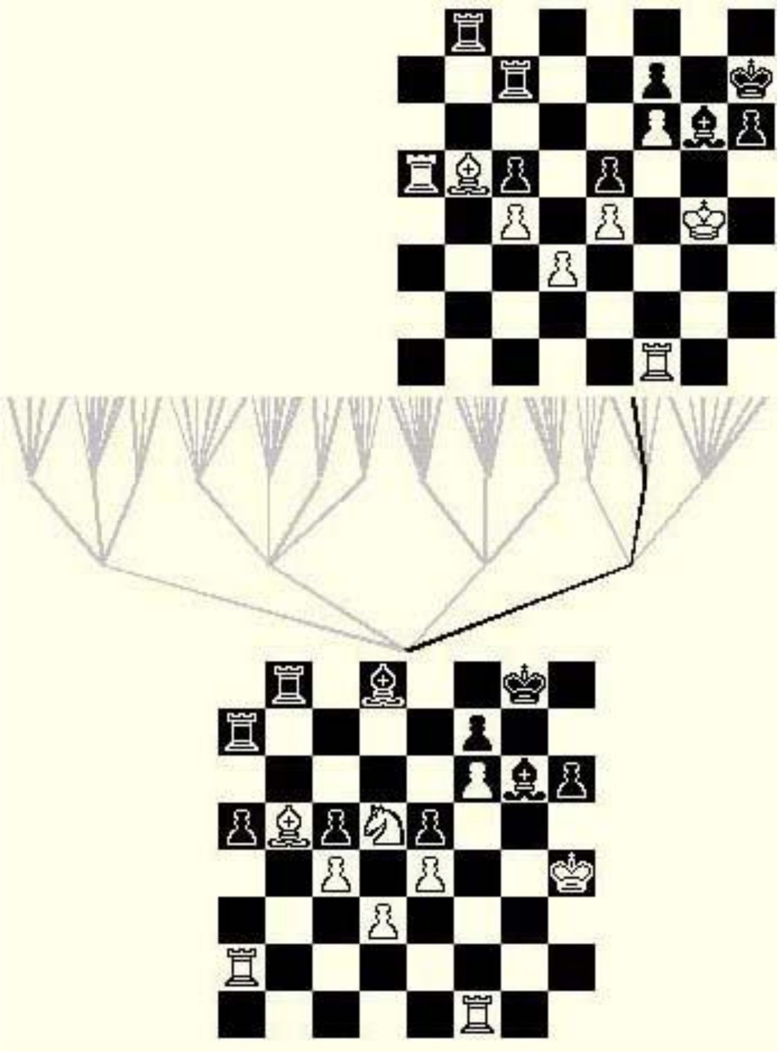


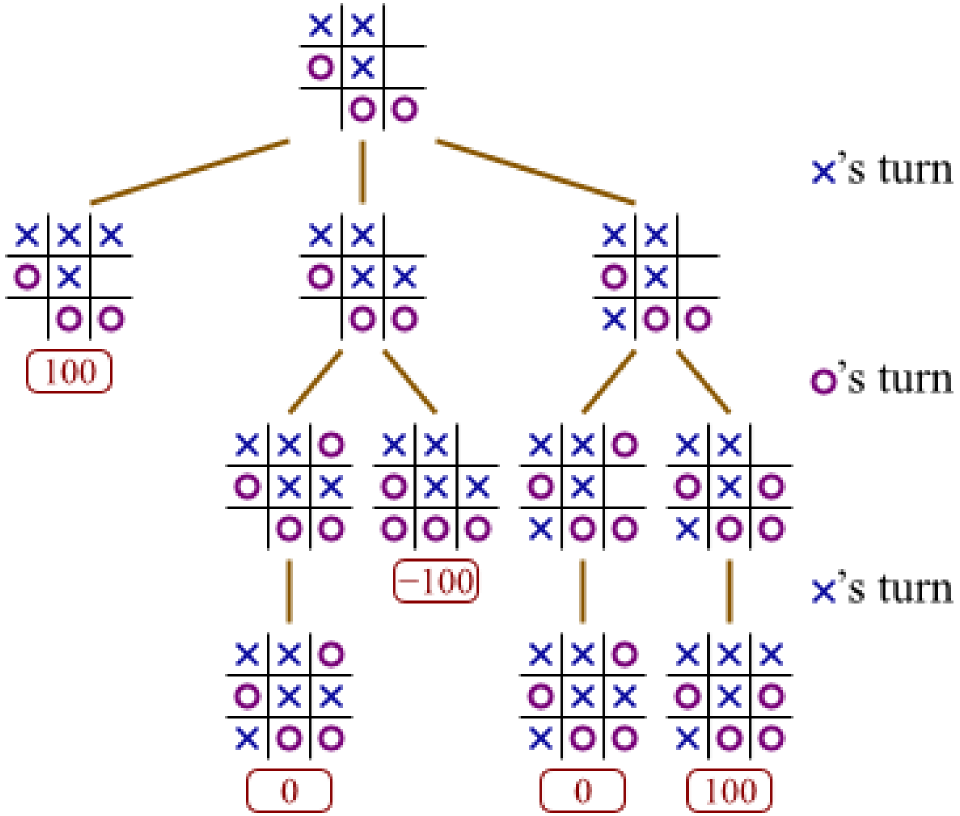


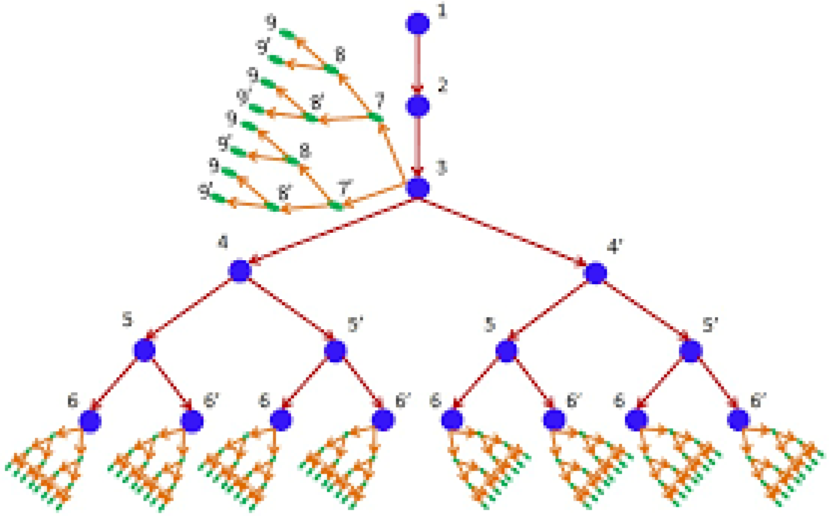




Absztrakt tér

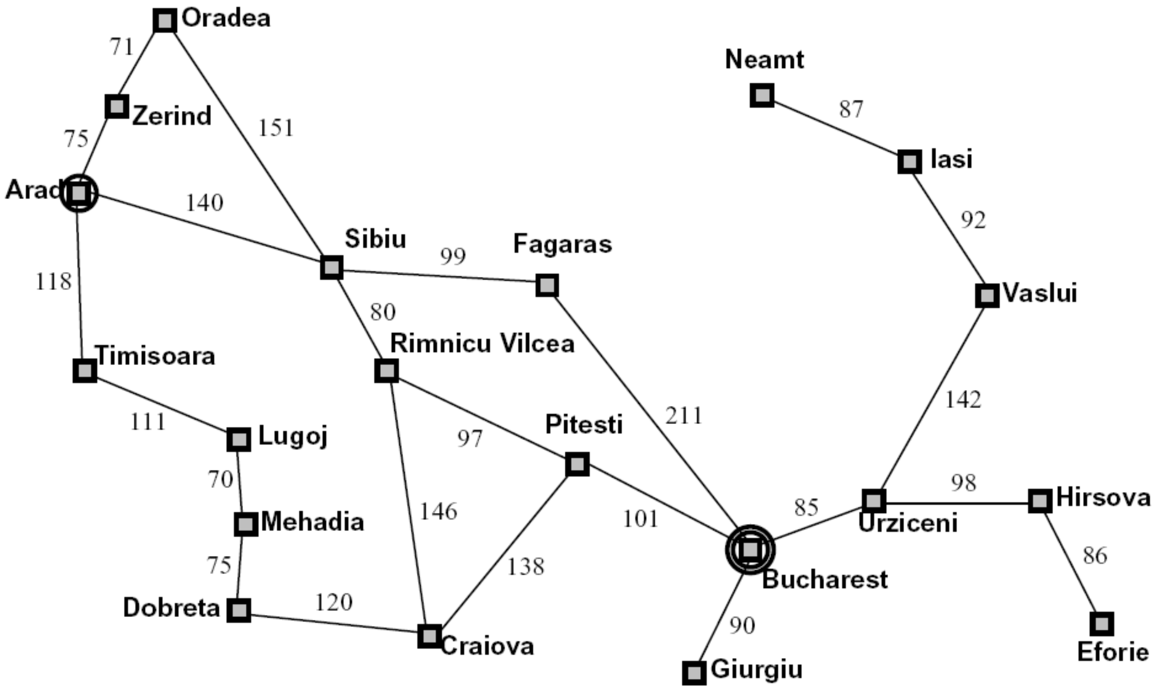






A keresési problémák modellek

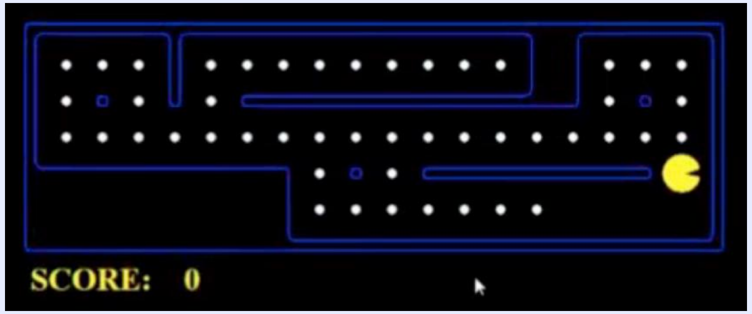
Példa: Útvonaltervezés



- Állapottér:
  - Városok
- Állapotátmenet-függvény:
  - Utak: Utazz a szomszédos városba adott költséggel (= távolság)
- Kiindulási állapot:
  - Arad
- Célállapot tesztelése:
  - Állapot == Bukarest?
- Megoldás?

Mit tartalmaz az állapottér?

A világ állapota a környezet minden részletét tartalmazza



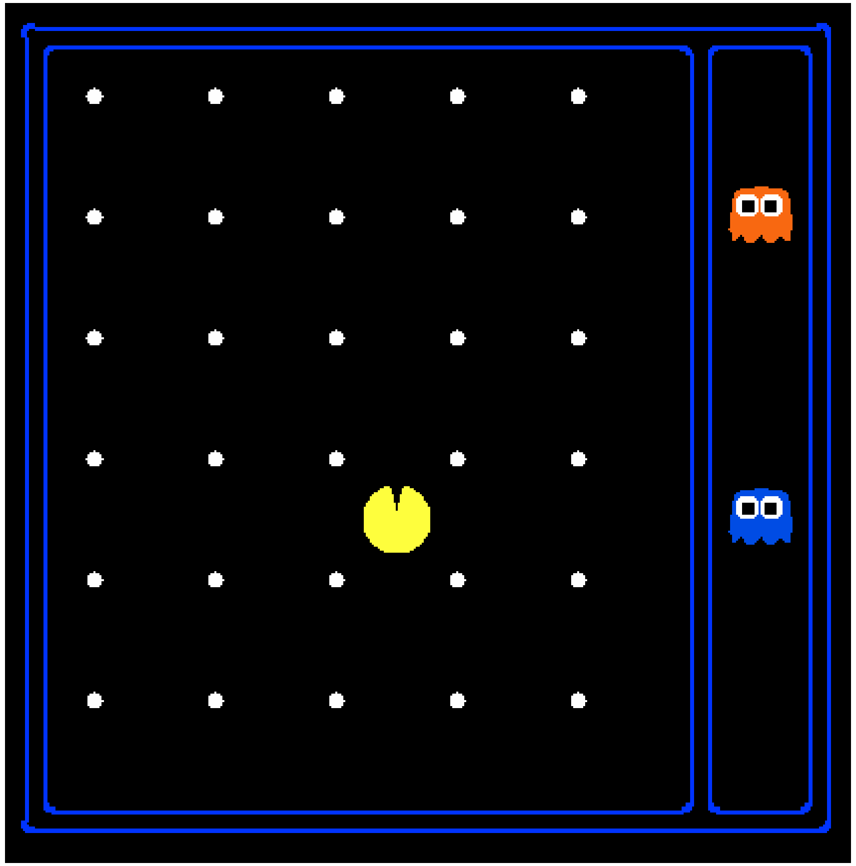
A keresési tér csak azokat a részleteket tartalmazza, melyekre szükség van a tervezéshez (absztrakció)

- 1. Probléma: útvonal
  - Állapotok: (x,y) helyzet
  - Cselekvések: ÉDKNY (mozgás)
  - Állapotátmenet függvény: helyzet frissítése
  - Célteszt: (x,y)=Vége
- 2. Probléma: „pontok megevése”
  - Állapotok: {(x,y), pontok - boolean}
  - Cselekvések: ÉDKNY (mozgás)
  - Állapotátmenet függvény: helyzet frissítése, (ha még ott van) pont jelenlét változó átállítása
  - Célteszt: nincs több pont

Állapotér nagysága

- Világállapotok:
  - Ágens pozíciók: 120
  - Pontok (élelem) száma: 30
  - Szellemek pozíciója: 12
  - Ágens iránya: ÉDKNY

- Nagyságrendek
  - Összes lehetséges állapot: $120 \times (2^{30}) \times (12^{2}) \times 4$
  - Lehetséges útvonaltervezési állapotok: 120
  - Lehetséges pont (élelem) állapotok: $120 \times (2^{30})$



Állapotér nagysága

Pl. a sakk – elég egyszerű szerkezetű állapottér (64 mező, 32 bábú)

1. lépés után (világos nyitólépése): **20** lehetséges állás
2. lépés után (sötét válaszlépése): **400** lehetséges állás
3. lépés után: **8.902**
4. lépés után: **197.281**
5. lépés után: **4.865.609**

stb.

A (jó) sakkozó nem egyforma intenzitással vizsgálja az egyes lehetőségeket!

Ágensek tervezése

- **A környezet nagy mértékben befolyásolja az alkalmazandó ágens kialakítását, elvárt képességeit**
- **A környezet lehet…**
  - ***Teljesen/részlegesen megfigyelhető***=> az ágensnek szüksége lehet ***memóriára*** (belső állapot nyilvántartására)
  - ***Diszkrét/folytonos*** => az ágens lehet, hogy nem képes az összes lehetséges állapot megkülönböztetésére
  - ***Sztochasztikus/determinisztikus*** => az ágensnek lehet, hogy több lehetséges ***forgatókönyvet*** kell kidolgoznia
  - ***Egyedüli ágens/ Több ágens*** => lehet, hogy az ágensnek ***véletlenszerűen*** kell viselkednie

Melyik a jó keresési eljárás?

- Elért eredmény szerint…
  - Megtaláltuk-e a legjobb célállapotot?
  - Találtunk-e legalább egy jó célállapotot?
  - Megtaláltuk-e az összes lehetséges célállapotot?

- Megtalálás költsége szerint…
  - A lehető legkisebb idő/tárhely
  - Melyik számít, számít-e egyáltalán?

- Az egzisztencia bizonyítás – létezik megoldás, de a gyakorlati részletek nem ismertek/nem érdekesek

Melyik a jó keresési eljárás?

1. Teljesség (*completeness*) - ha van megoldás, biztosan megtalálja

2. Időigény (*time complexity*) – mennyi idő a megoldás megtalálása?

3. Tárigény (*space complexity*) – mennyi tárhely kell

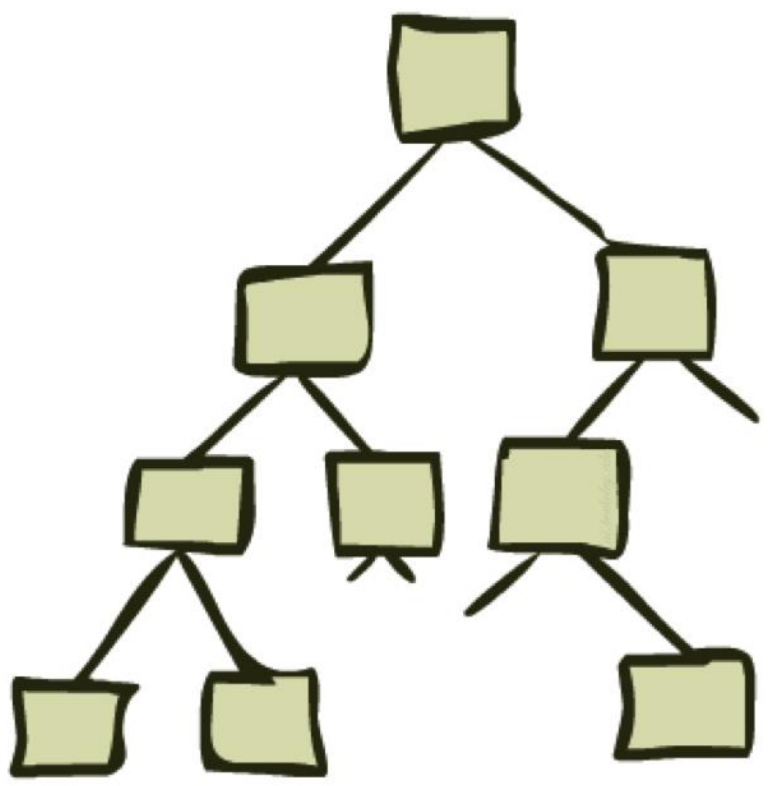
4. Optimalitás (*optimality*) – ha több megoldás van, megtaláljuk-e a legjobbat? (mi a legjobb? – sokszor izgalmas kérdés!)

Mesterséges intelligencia

Problémamegoldás kereséssel

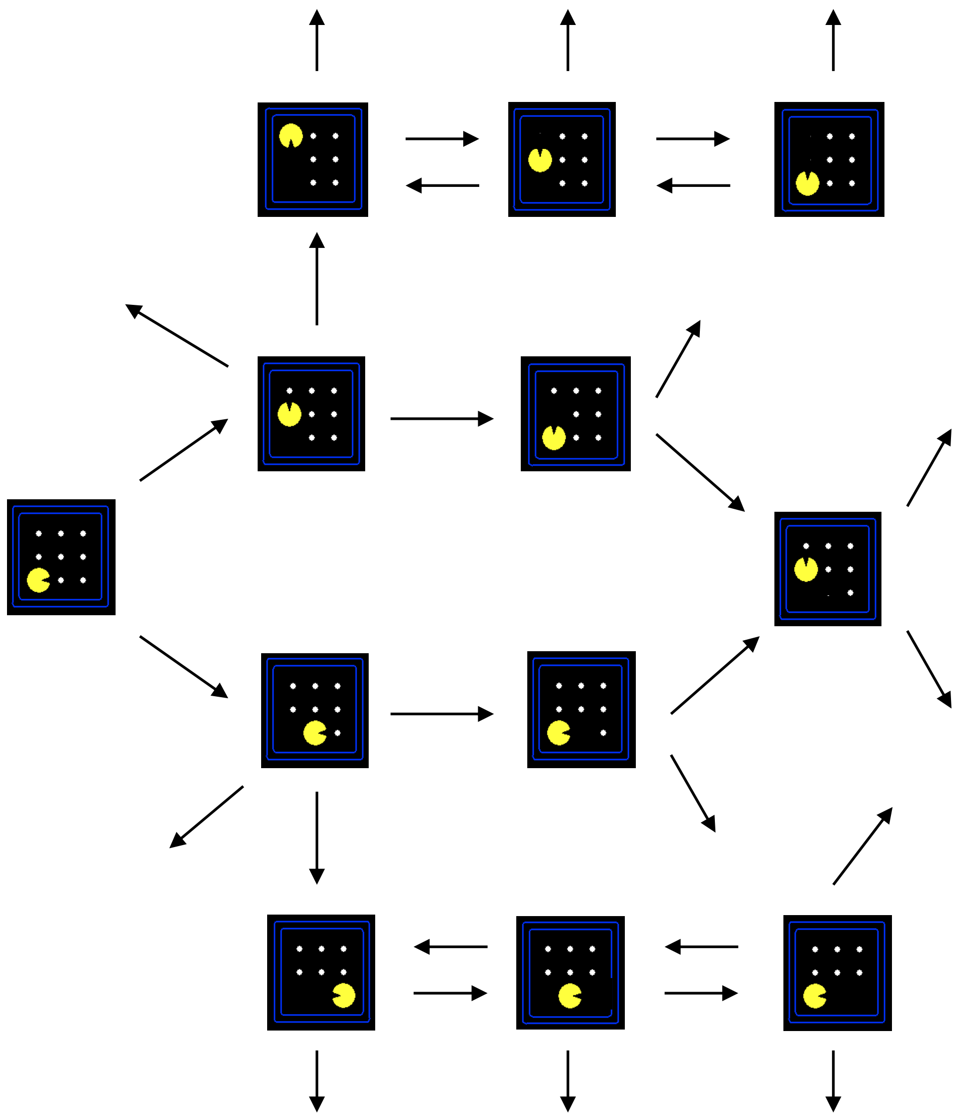
Állapottér-reprezentáció
Egyszerű keresési stratégiák

Állapottérgráfok és keresési fák



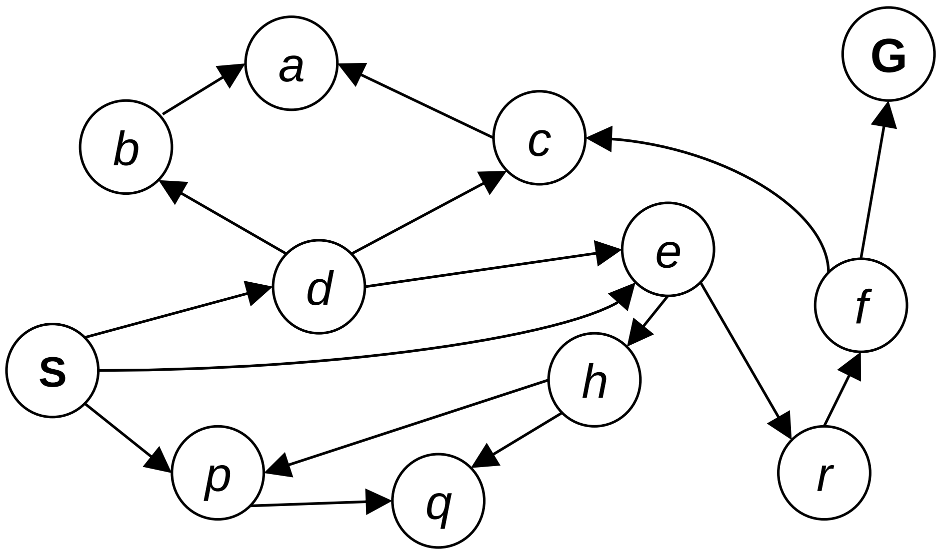
Állapottérgráfok

- Állapottérgráf: A keresési probléma matematikai reprezentációja
  - A csomópontok a világ állapotának absztrakt reprezentációi
  - Az élek az állapotátmenetet jelölik (cselekvések következményeit)
  - A célteszt a célállapot(ok) elérését vizsgálja
- Az állapottérgráfban minden állapot egyszer fordul elő
- Ritkán lehet ezt a gráfot teljes egészében felépíteni a memóriában (tárhelykorlátok miatt)



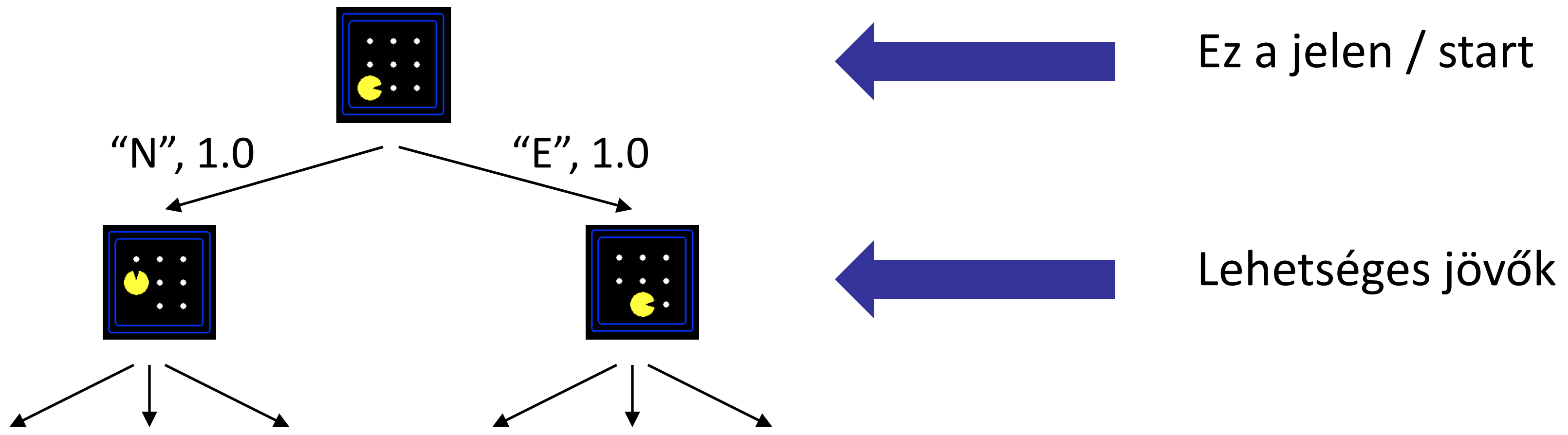
Állapottérgráfok

- Állapottérgráf: A keresési probléma matematikai reprezentációja
  - A csomópontok a világ állapotának absztrakt reprezentációi
  - Az élek az állapotátmenetet jelölik (cselekvések következményeit)
  - A célteszt a célállapot(ok) elérését vizsgálja
- Az állapottérgráfban minden állapot egyszer fordul elő
- Ritkán lehet ezt a gráfot teljes egészében felépíteni a memóriában (tárhelykorlátok miatt)



Kis állapottérgráf egy kis méretű keresési problémához

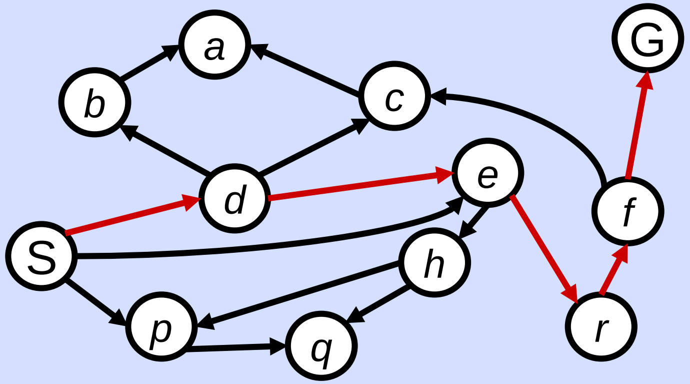
Keresési fák



- A keresési fa:
  - Egy “Mi lenne ha” fa tervekből és azok következményeiből
  - A kezdőállapot a gyökércsomópont
  - A gyermekcsomópontok a követő állapotoknak felelnek meg
  - A csomópontok állapotokat jelölnek és olyan tervekhez tartoznak, melyek elérik/tartalmazzák azokat az állapotokat.
  - Legtöbb probléma esetén nem lehet ténylegesen felépíteni a teljes keresési fát

Állapottérgráfok és keresési fák

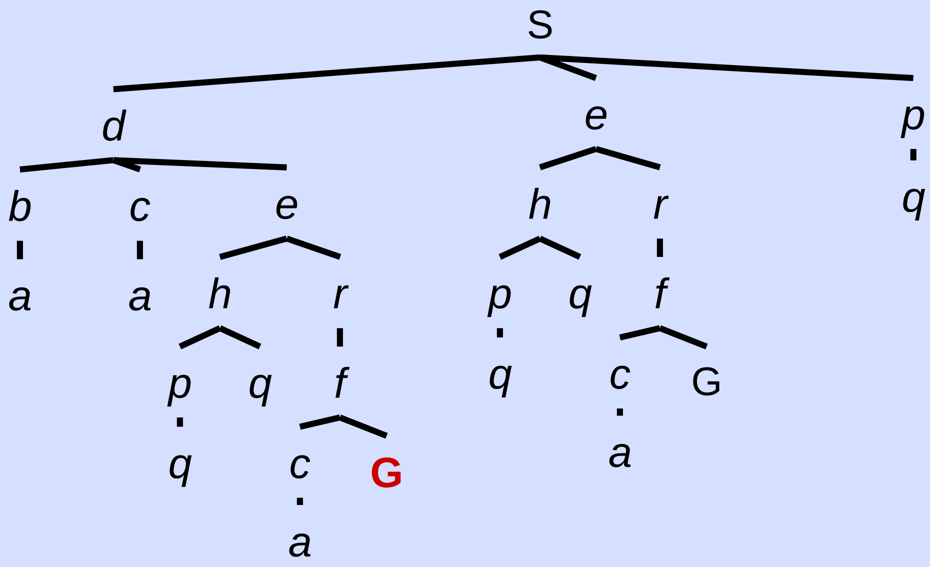
Állapottérgráf



Minden egyes csomópont a keresési fában egy útvonalnak felel meg az állapottér-gráfban.

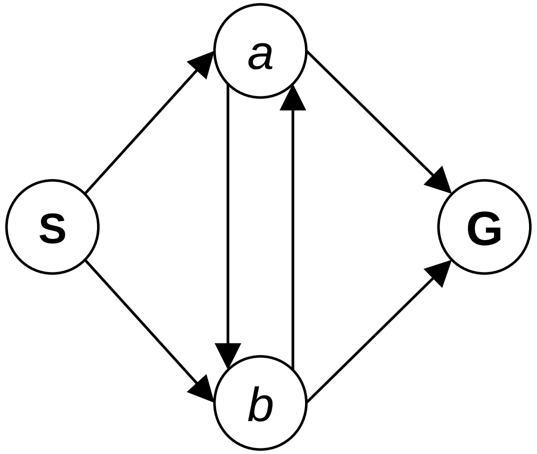
Mindkettőt igény szerint építjük, olyan kis mértékben, amennyire csak lehetséges

Keresési fa



Kvíz: állapottérgráf vs. keresési fa

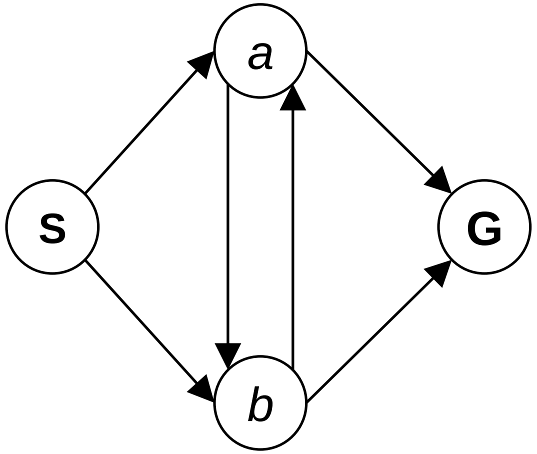
Tekintsük egy 4 állapotú gráfot:



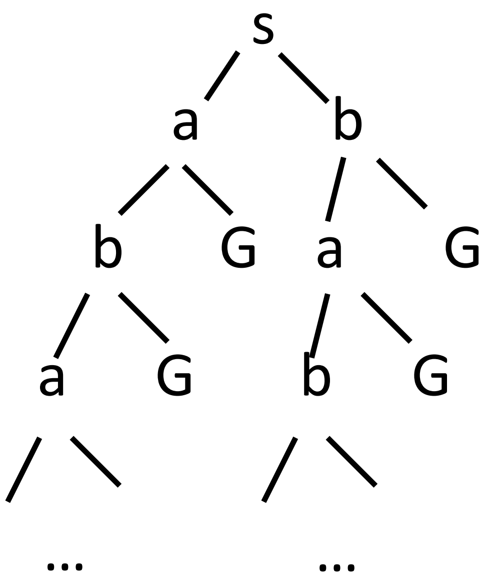
Mekkora az ehhez tartozó keresési fa (S-ből indulva)?

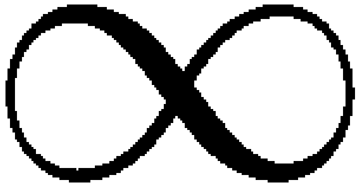
Kvíz: állapottérgráf vs. keresési fa

Tekintsük egy 4 állapotú gráfot:



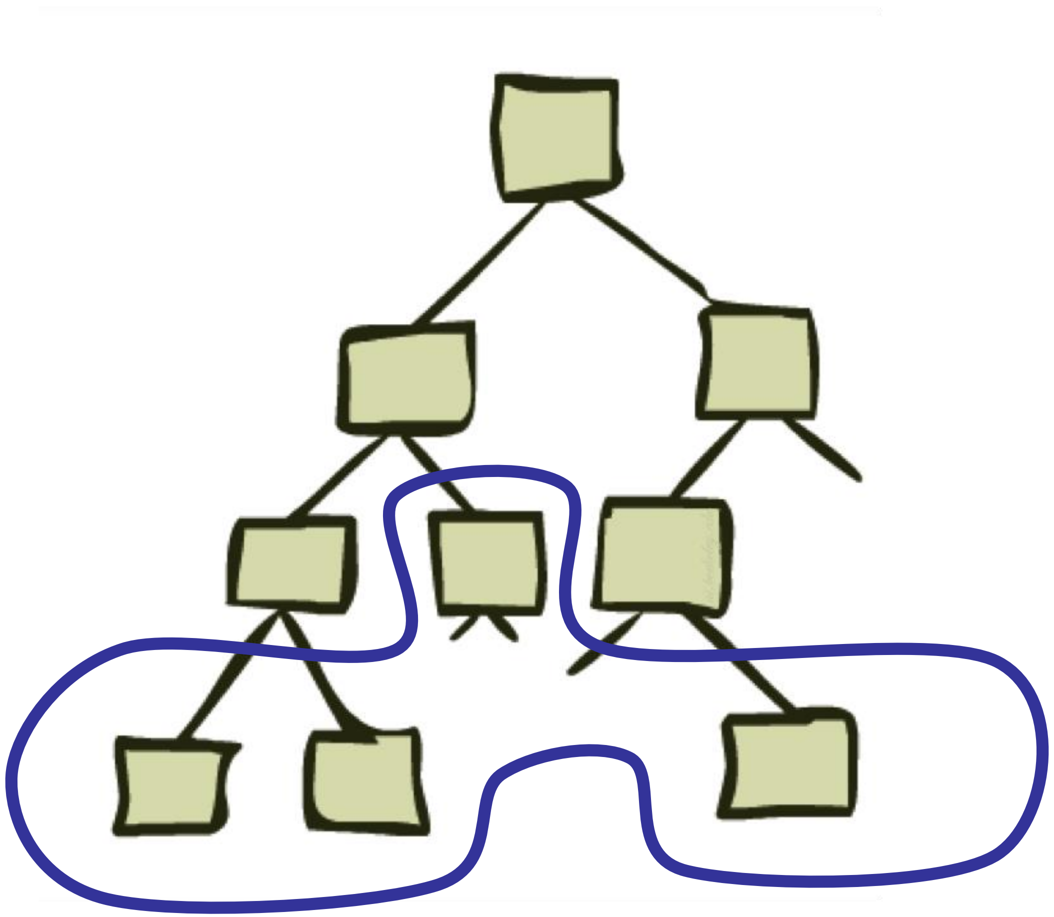
Mekkora az ehhez tartozó keresési fa (S-ből indulva)?



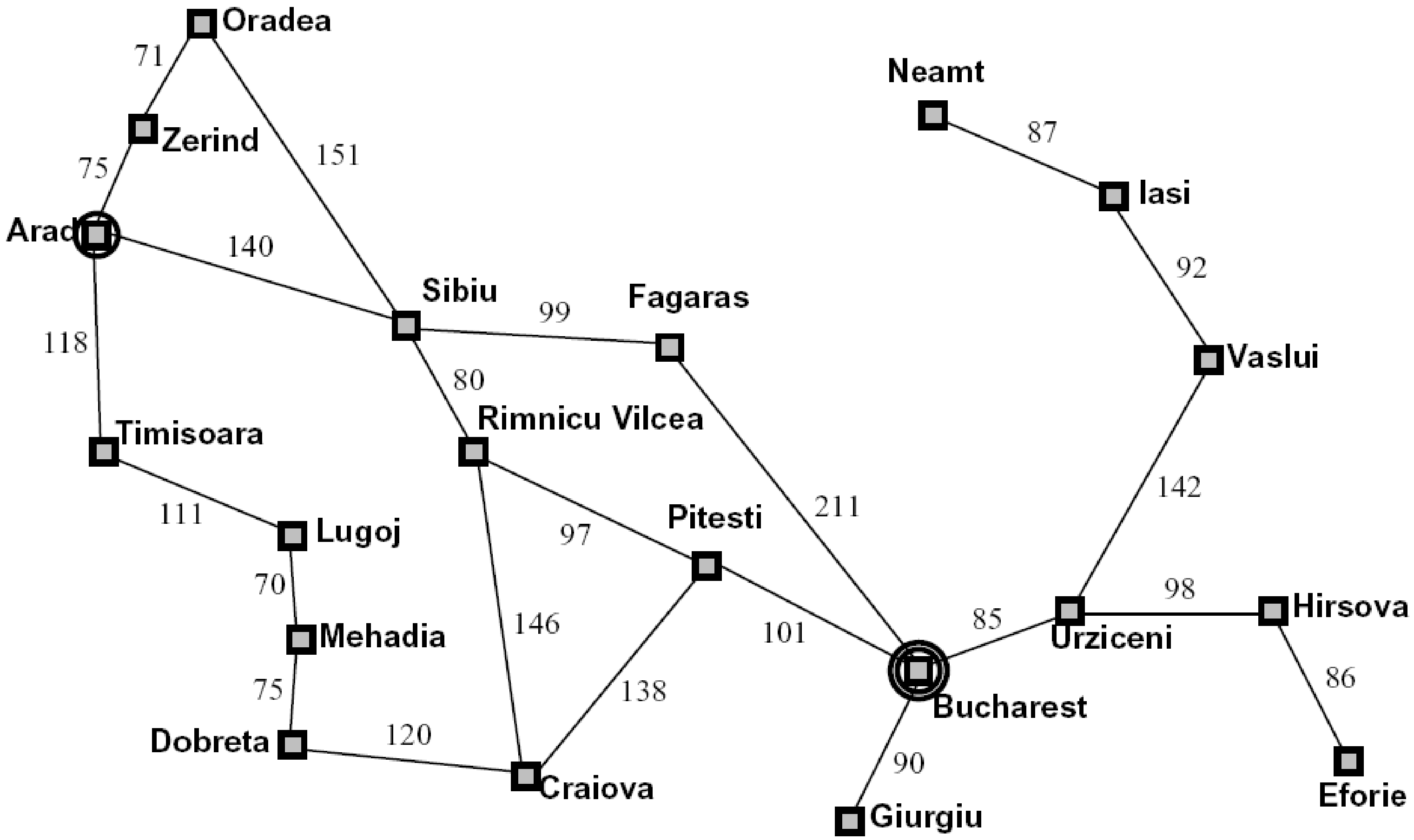


Általában: sok lehet az ismétlődő elem a keresési fában

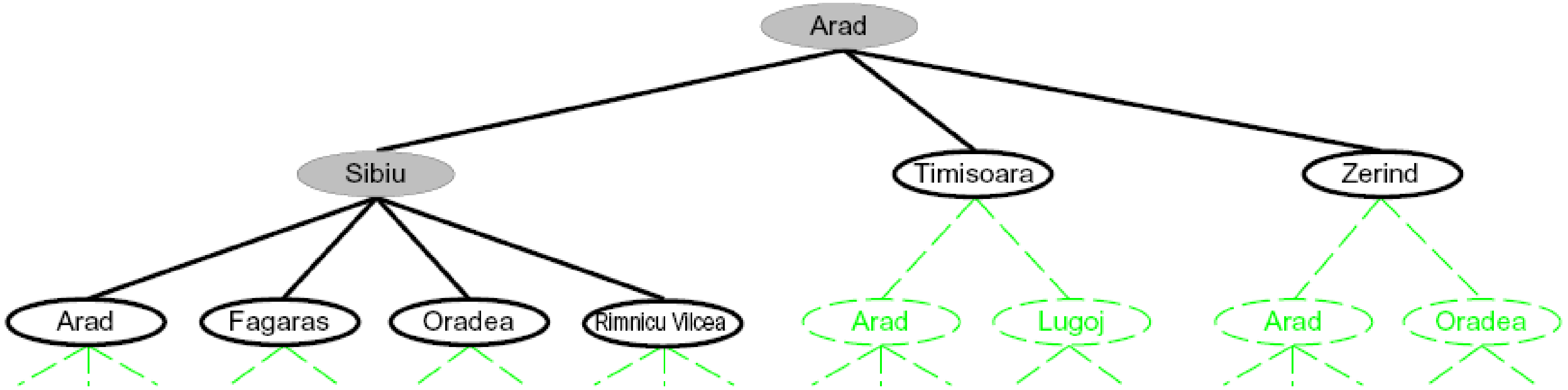
Keresési fa



Keresési példa: útvonaltervezés



Keresés keresési fával



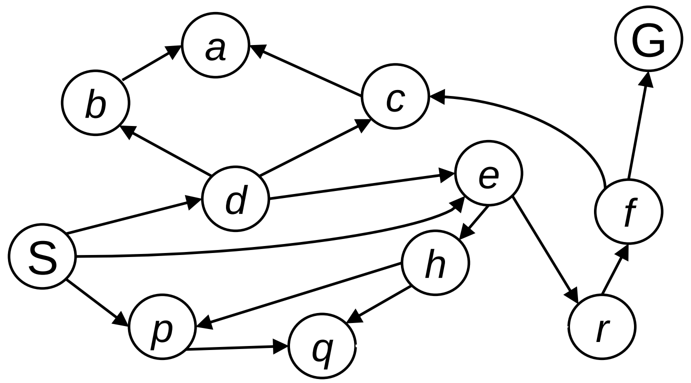
- Keresés:
  - Lehetséges tervek (fa csomópontok) kifejtése
  - „Perem” létrehozása a figyelembe vehető résztervek nyilvántartására
  - Minél kevesebb csomópont kifejtése

Általános fa alapú keresés

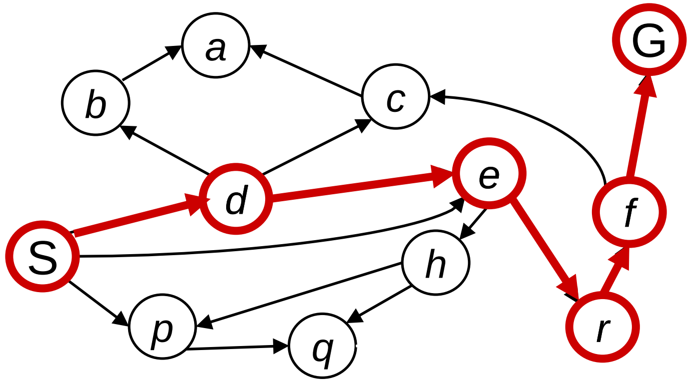
```
function FA-KERESÉS (probléma, stratégia) returns megoldás vagy kudarc
        Keresési fa inicializálása a probléma kiinduló állapotával
        loop do
                If nincs jelölt a kifejtéshez then return kudarc
                Kifejtendő levél csomópont kijelölése a stratégia alapján
                If a csomópont tartalmaz egy célállapotot then return megoldás
                else csomópont kifejtése és az eredmény csomópontok hozzáadása a keresési fához
        end
```

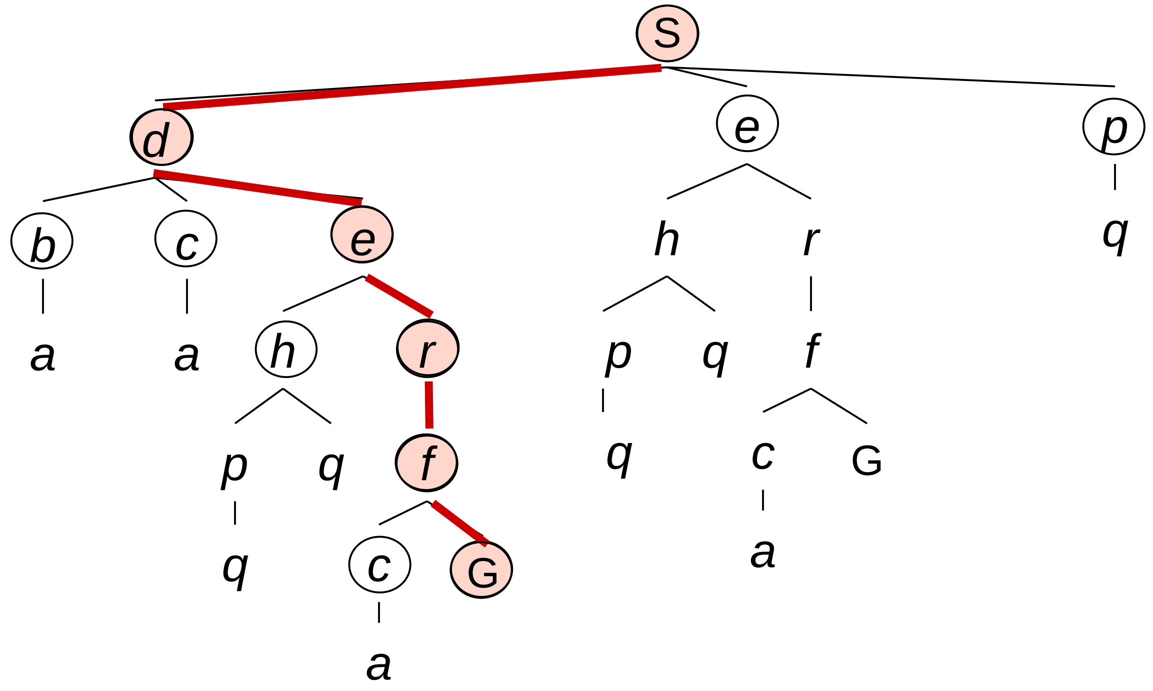
- Lényeges elemek:
  - Perem
  - Kifejtés
  - Felfedezési stratégia
- Központi kérdés: melyik perembeli csomópontot fejtsük ki?

Példa: Fa keresés



Példa: Fa keresés





~~s~~
~~s → d~~
s → e
s → p
s → d → b
s → d → c
~~s → d → e~~
s → d → e → h
~~s → d → e → r~~
~~s → d → e → r → f~~
s → d → e → r → f → c
~~s → d → e → r → f → G~~

Keresési stratégiák – avagy miből gazdálkodhatunk?

- Mire vagyunk képesek? (saját modell)
- Milyen körülöttünk a környezet? (cél-, távolsági modellek)

Nem informált keresések (**gyenge** vagy **vak** keresések)

a. tudjuk: hogy néz ki a célállapot

b. egyáltalán nem: milyen költségű az aktuálisból a célállapotba vezető út

Informált keresések (heurisztikus keresések)

a. tudjuk: hogy néz ki a célállapot

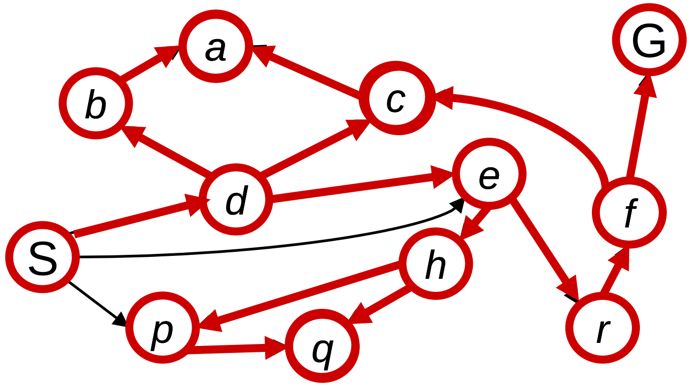
b. (jó) becslésünk van arra, hogy: milyen költségű lehet az aktuális állapotból a célállapotba vezető út

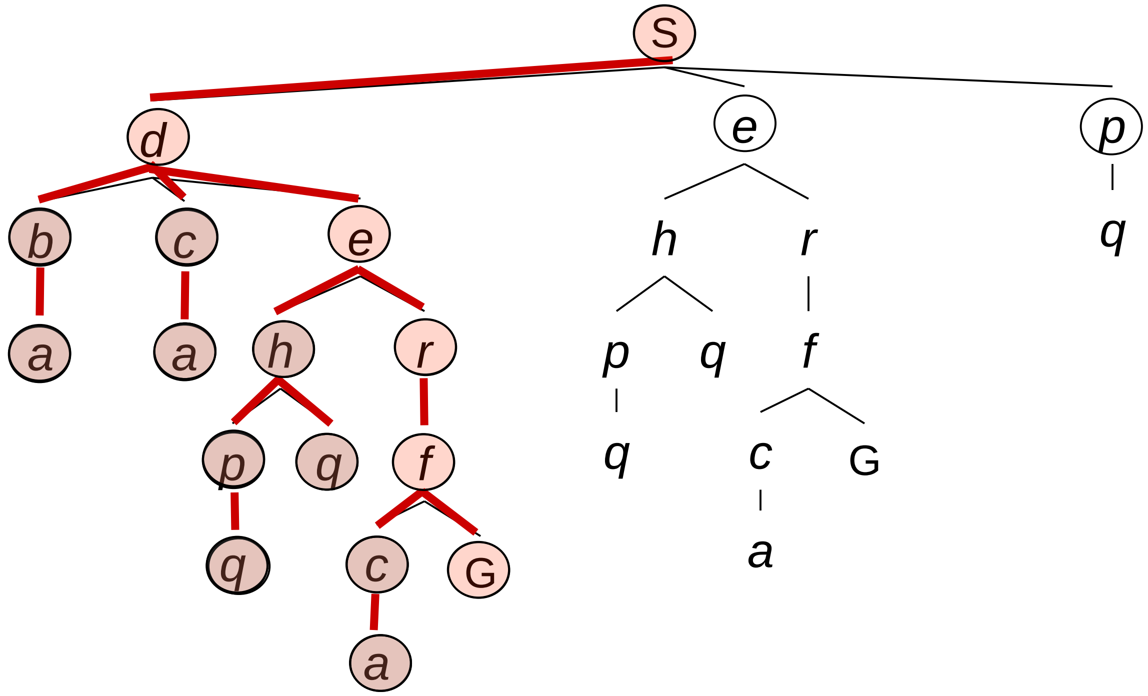
Mélységi keresés

Mélységi keresés

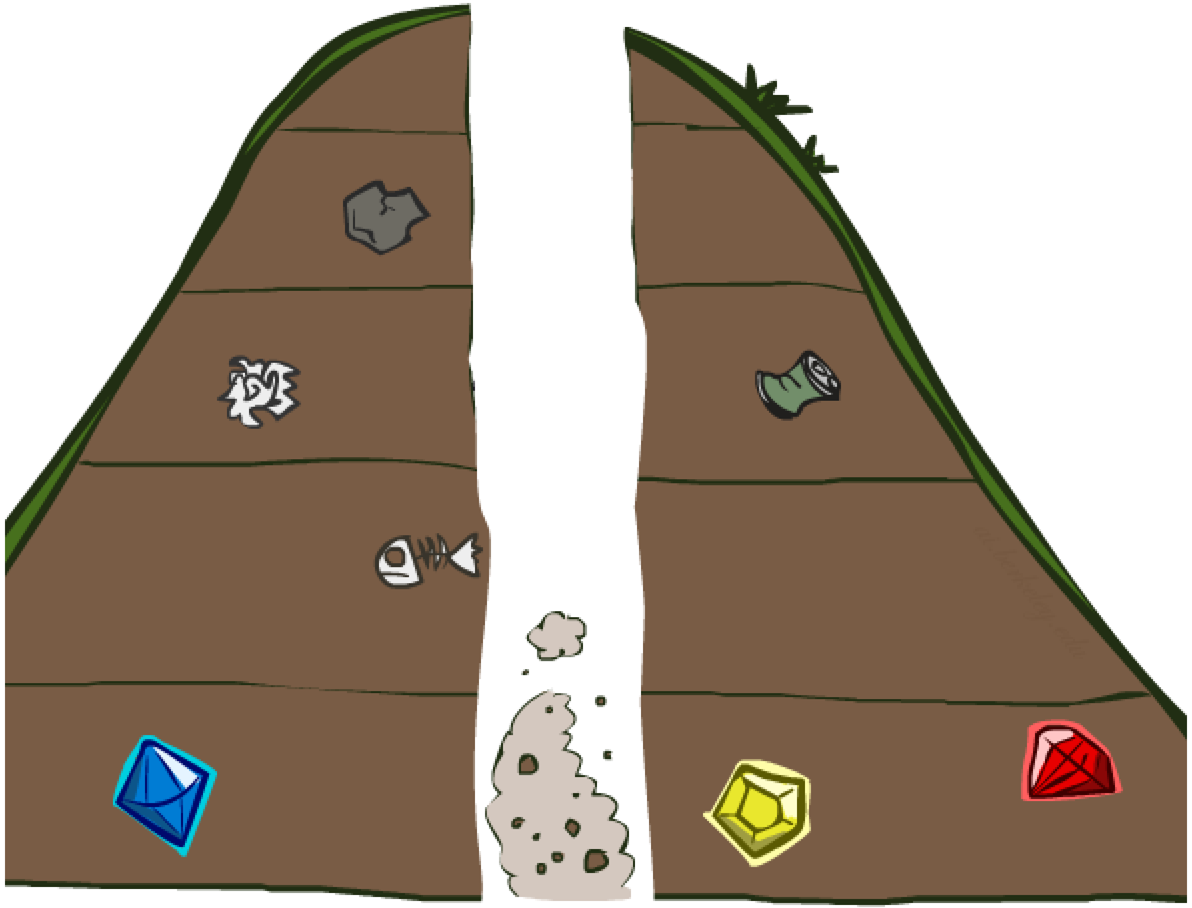
Stratégia: legmélyebb csomópont kifejtése

Megvalósítás: A perem egy LIFO stack





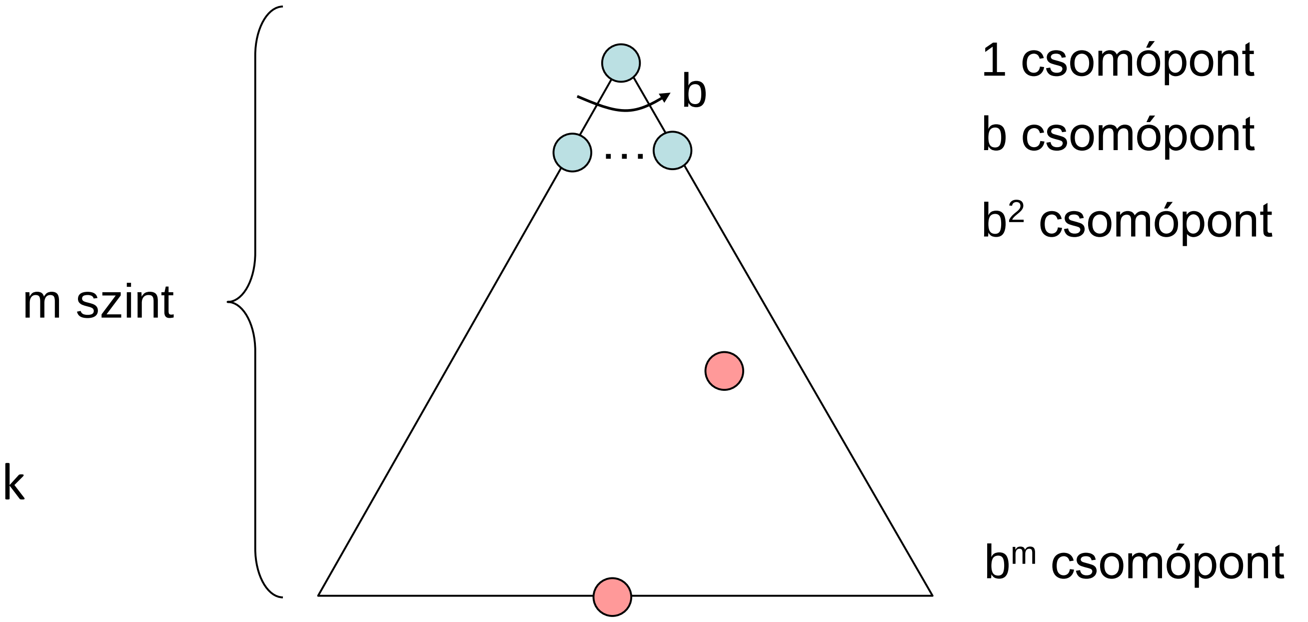
Keresési algoritmusok tulajdonságai



Keresési algoritmusok tulajdonságai

- Teljesség: Garantáltan megtalál-e egy megoldást, ha létezik egy?
- Optimalitás: Garantáltan megtalálja-e a legkisebb költségű utat?
- Időkomplexitás
- Tárkomplexitás

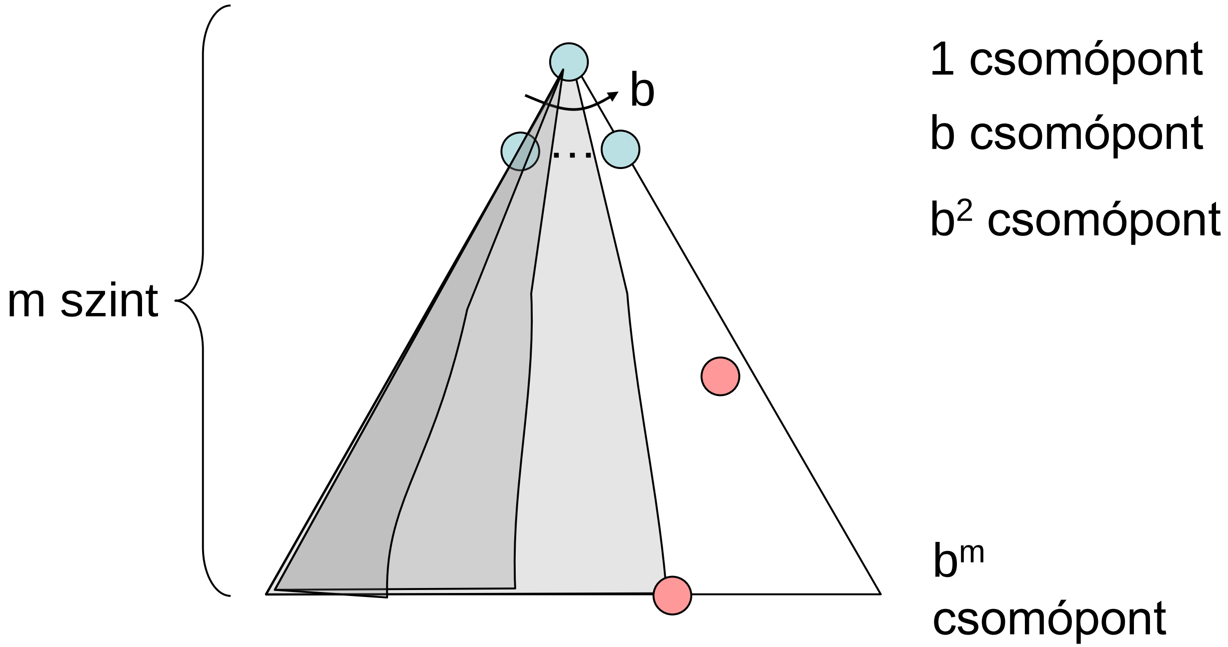
- Keresési fa váza:
  - **b** az elágazási faktor
  - **m** a maximum mélység
  - adott mélységben elérhető megoldások



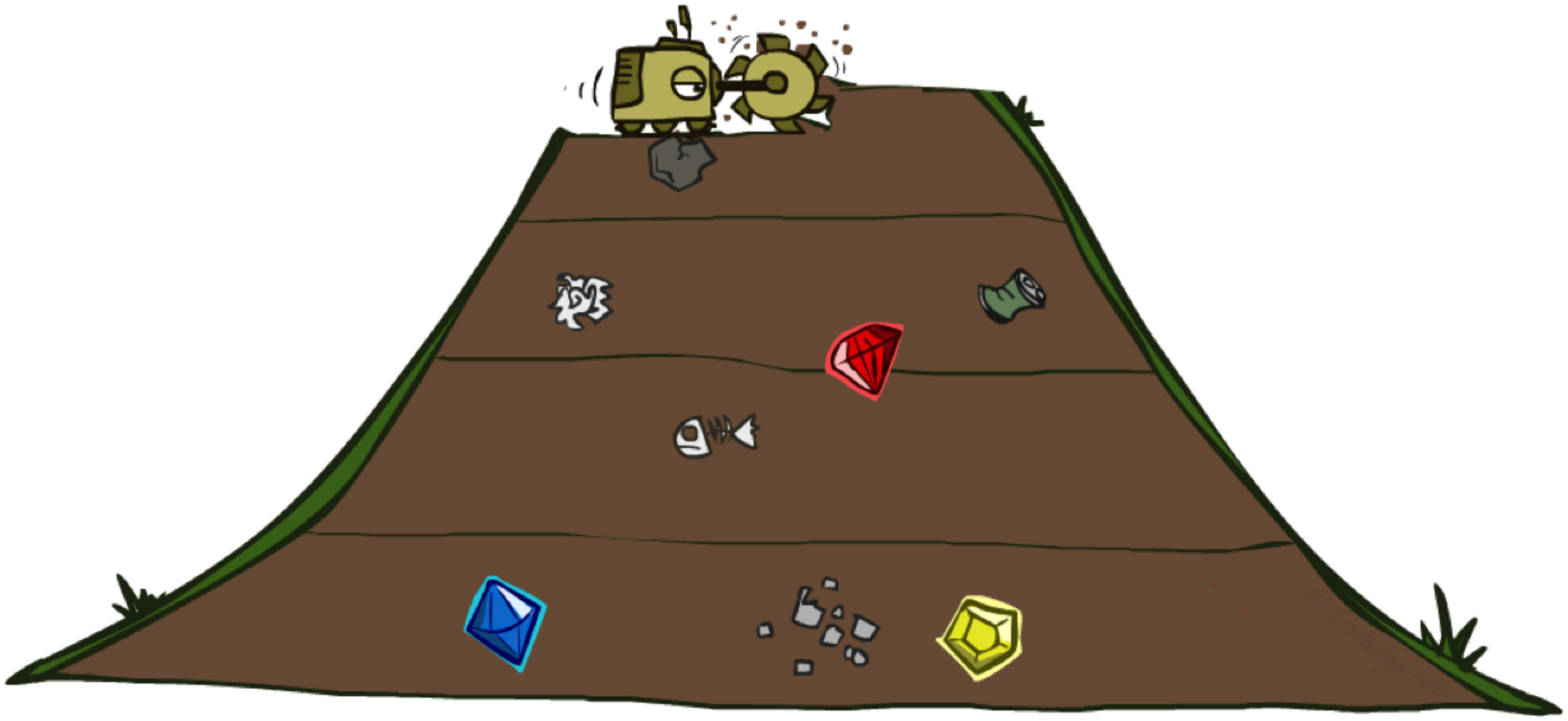
- A teljes fában lévő összes csomópont:
  - $1 + b + b^2 + \ldots b^m = O(b^m)$

Mélységi keresés (DFS) jellemzői

- Milyen csomópontokat fejt ki a mélységi keresés?
  - A keresési fa egy bal oldali prefix részét fejti ki
  - A teljes fát feldolgozhatja
  - Ha **m** véges, O(b^m) időt vesz igénybe
- Mennyi tárhelyet igényel a perem?
  - Csak a gyökércsomóponthoz vezető úton szereplő elágazások számítanak: O(bm)
- Teljes-e?
  - **m** lehetne végtelen, így csak akkor lehet teljes, ha körök kialakulását megakadályozzuk
- Optimális-e?
  - Nem, megtalálja a legbaloldalibb megoldást, függetlenül a mélységtől és a költségtől



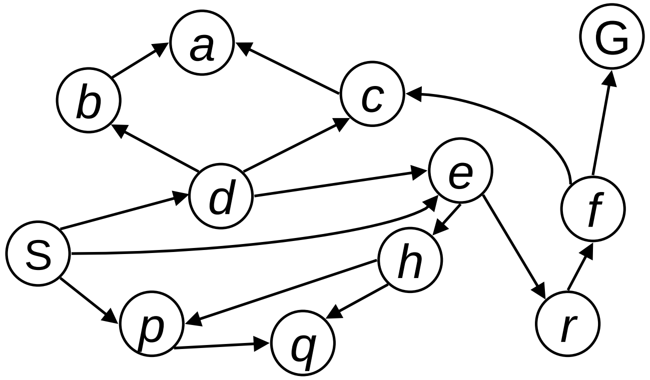
Szélességi keresés

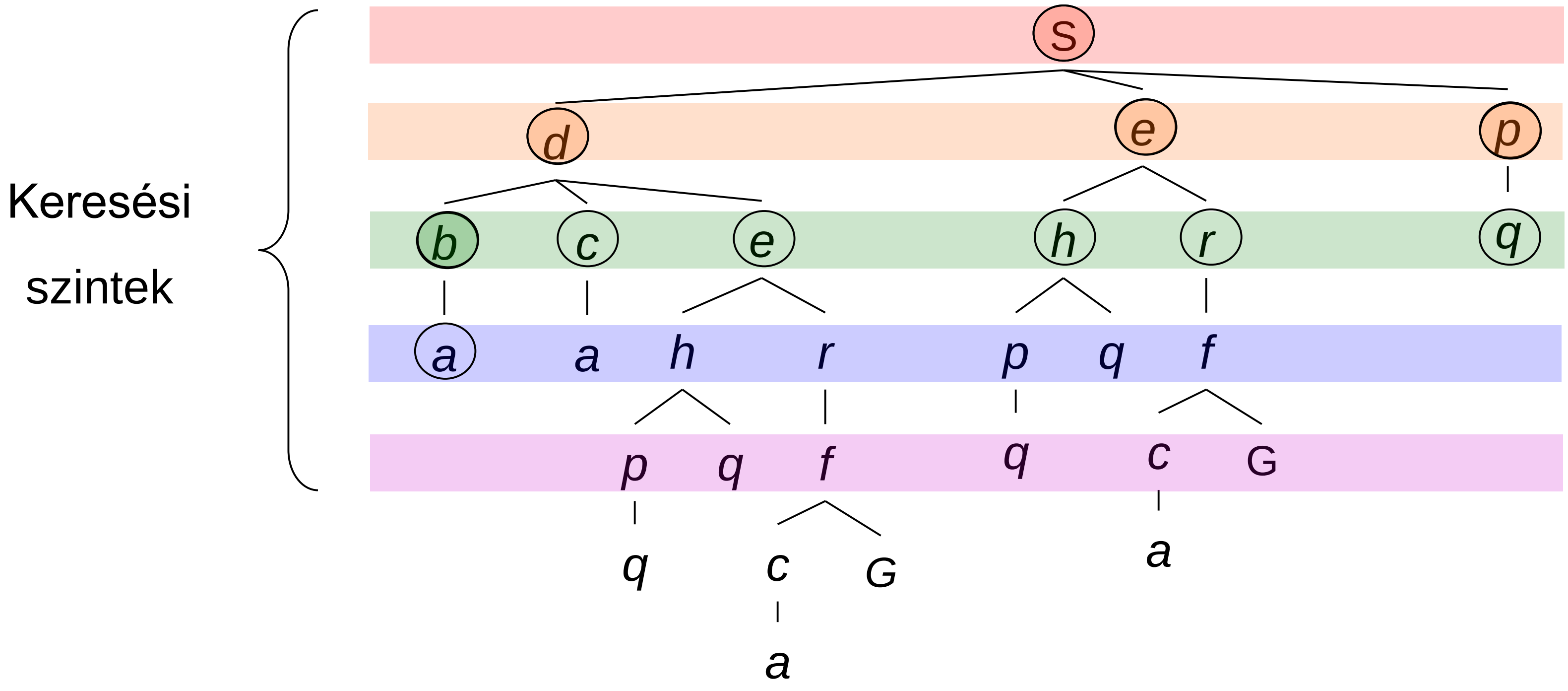


Szélességi keresés

Stratégia: a legkevésbé mélyen lévő csomópont kifejtése

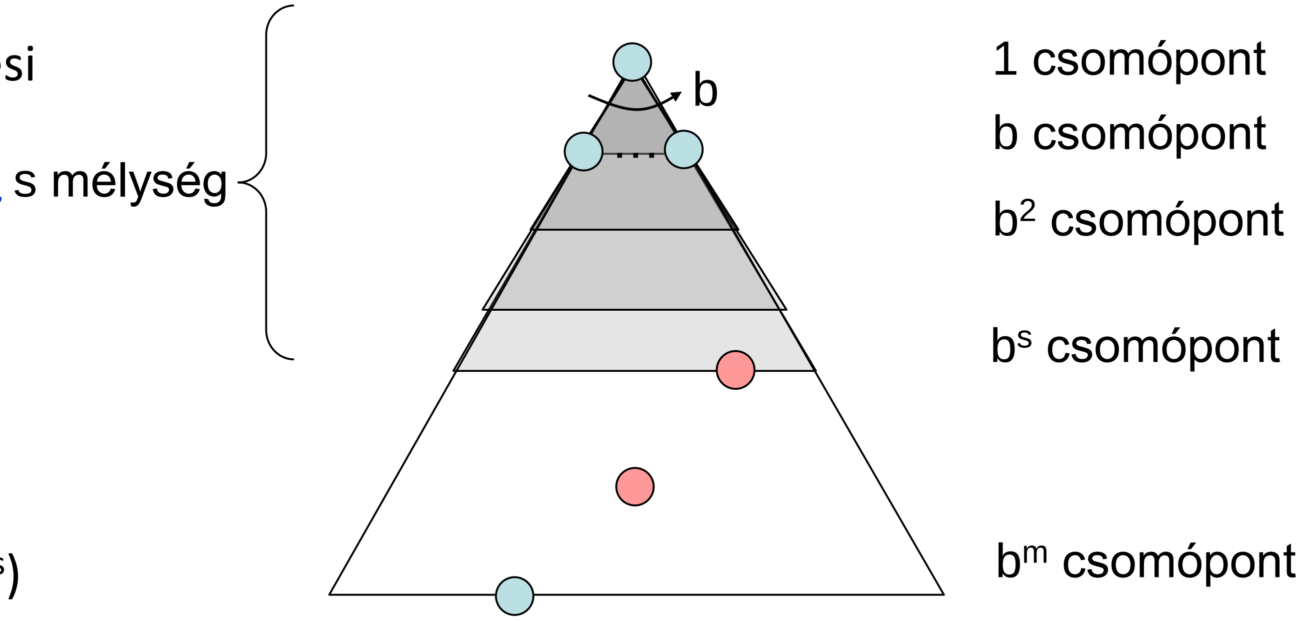
Megvalósítás: A perem egy FIFO sor



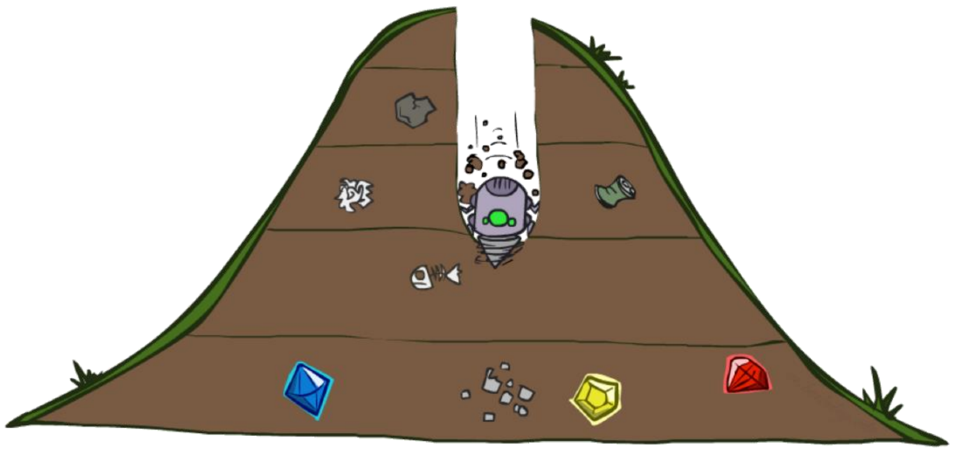


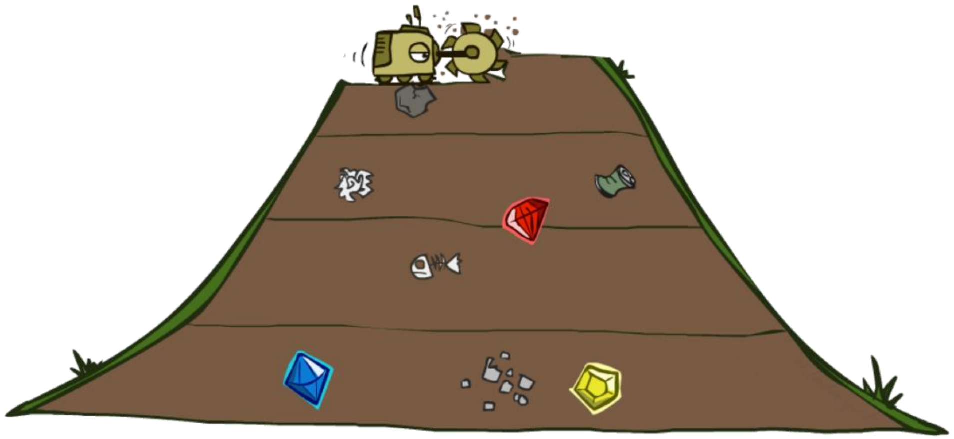
Szélességi keresés (BFS) tulajdonságai

- Milyen csomópontokat fejt ki a szélességi keresés?
  - Feldolgozza az összes csomópontot a keresési fában legsekélyebben fekvő megoldásig
  - Jelölje a legsekélyebb megoldás mélységét ***s***
  - A keresés ideje ekkor: $O(b^s)$
- Mekkora tárhelyet igényel a perem nyilván tartása?
  - Közelítőleg a legutolsó szinttel arányos: $O(b^s)$
- Teljes-e?
  - ***s*** véges, ha létezik megoldás, tehát igen!
- Optimális-e?
  - Csak akkor, ha minden egységesen 1 költségű



Melyik a jobb: a mélységi vagy a szélességi keresés?





Melyik a jobb: a mélységi vagy a szélességi keresés?

- Mikor múlja felül szélességi keresés a mélységit?
- Mikor múlja felül mélységi keresés a szélességit?

Mélységkorlátozott keresés

- Az utak maximális mélységére egy vágási korlátot ad.

- A mélységi levágásnál a keresés visszalép. A megoldást, amennyiben létezik és a mélységkorlátnál sekélyebben fekszik, garantáltan megtaláljuk.

- De semmi garancia nincs arra, hogy a legkisebb költségű (itt: legrövidebb út) megoldást találjuk meg.

- Amennyiben túl kis mélységkorlátot választunk, akkor a mélységkorlátozott keresés még csak teljes sem lesz.

Románia jelen egyszerűsített térképe: 20 város. Ha létezik egy megoldás, az maximálisan 19 lépés hosszú lehet.
Minden város bármelyik városból legfeljebb 9 lépésben elérhető:
az állapottér **átmérője** = jobb mélységkorlát.

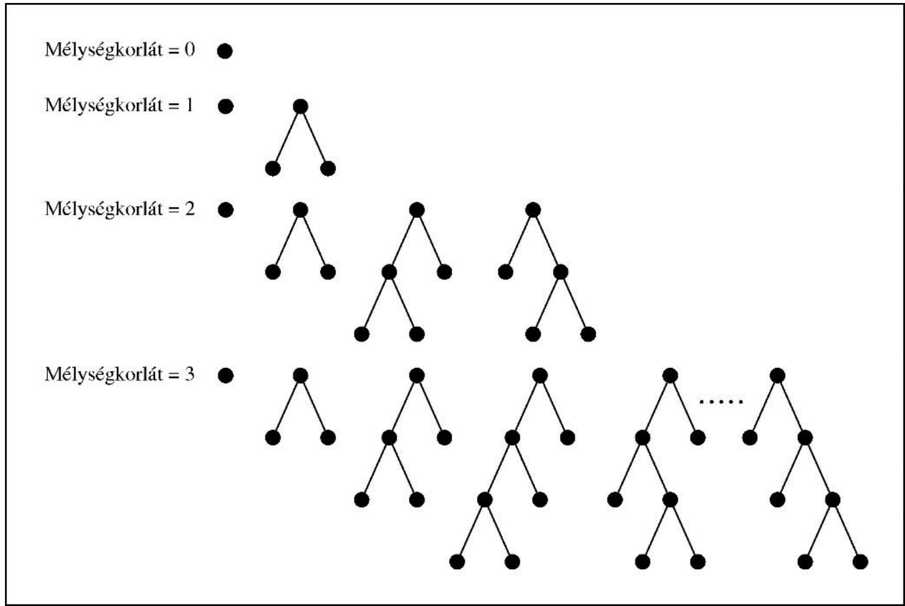
Iteratívan mélyülő keresés

Slate és Atkin (1977): CHESS 4.5 sakkprogram.
 - mélységkorlátozott keresés - egy jó mélységkorlát megválasztása?

A legtöbb esetben azonban mindaddig nem tudunk jó mélységkorlátot adni, amíg meg nem oldottuk a problémát.

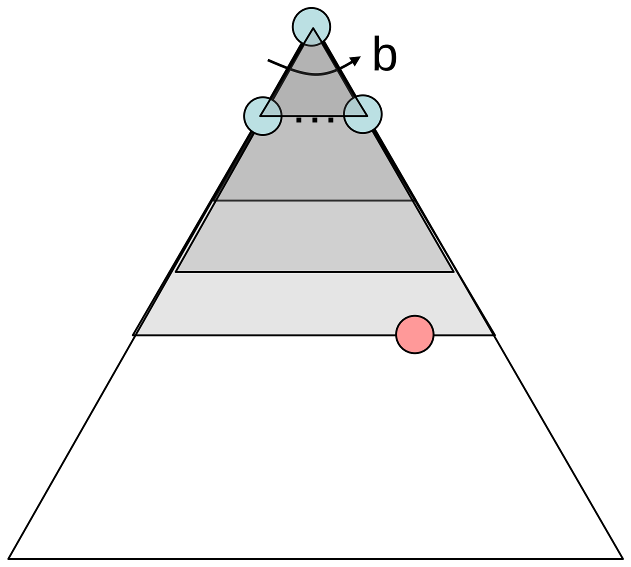
A legjobb mélységkorlát kiválasztása:

- kipróbálja az összes lehetséges mélységkorlátot: először 0, majd 1,
  majd 2, stb.
- mélységkorláttal végez mélységkorlátozott keresést.



Iteratívan mélyülő keresés

- Ötlet: egyesítsük a mélységi keresés relatíve alacsony tárhelyigényét a szélességi keresés relatíve alacsony időigényével
  (legsekélyebben fekvő megoldás megtalálása)
  - Futtassuk a mélységi keresést 1-es mélységi korláttal. Ha nincs megoldás …
  - Futtassuk a mélységi keresést 2-es mélységi korláttal. Ha nincs megoldás …
  - Futtassuk a mélységi keresést 3-es mélységi korláttal. Ha nincs megoldás …



- Nem pazarlóan redundáns ez?
  - A legtöbb „munka” a legalsó keresési szinten van, tehát nem olyan rossz a helyzet!

Iteratívan mélyülő keresés gyakorlatilag ötvözi a szélességi és mélységi keresés előnyös tulajdonságait

- A szélességi kereséshez hasonlóan **optimális** és **teljes**, de csak
- a mélységi keresés **szerény** memória igényével rendelkezik.
- De bizonyos állapotokat az algoritmus többször is kifejt!

Tékozló? - egy exponenciális keresési fában majdnem az összes csomópont a legmélyebb szinten található

d mélységben a megoldás, *b* elágazási tényező
szélességi keresésnél a kifejtések száma:

$1 + b + b^2 + ... + b^{d-2} + b^{d-1} + b^d$

az iteratívan mélyülő keresésnél a kifejtések teljes száma:

$(d+1)1 + (d)b + (d-1)b^2 + ... + 3b^{d-2} + 2b^{d-1} + 1b^d$

Iteratívan mélyülő keresés

- szélességi keresésnél a kifejtések száma:

$1 + b + b^2 + \ldots + b^{d-2} + b^{d-1} + b^d$

pl. b =10 és d = 5 esetén ez a szám: 1 + 10 + 100 + ... + 100000 = 111111

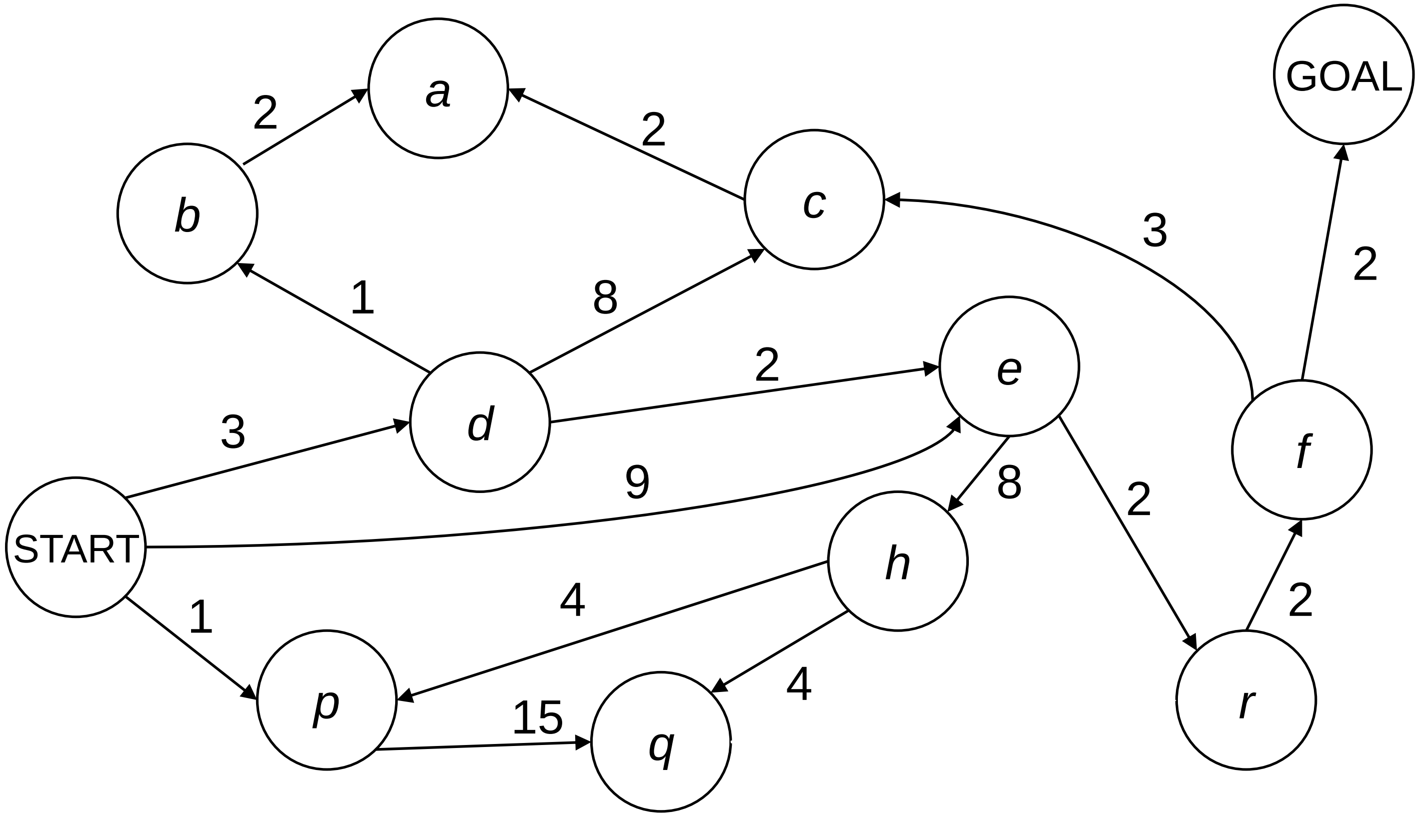
- az iteratívan mélyülő keresésnél a kifejtések teljes száma:

$(d+1)1 + (d)b + (d-1)b^2 + \ldots + 3b^{d-2} + 2b^{d-1} + 1b^d$

b = 10 és d = 5 esetén: 6 + 50 +400 + ... + **100.000** = **123.456** **(+23,5%)**

- minél nagyobb az elágazási tényező, annál kisebb a többletmunka (max. b = 2, kb. **200%**, kb. *kétszerese* a szélességinek!)

Költségérzékeny keresés



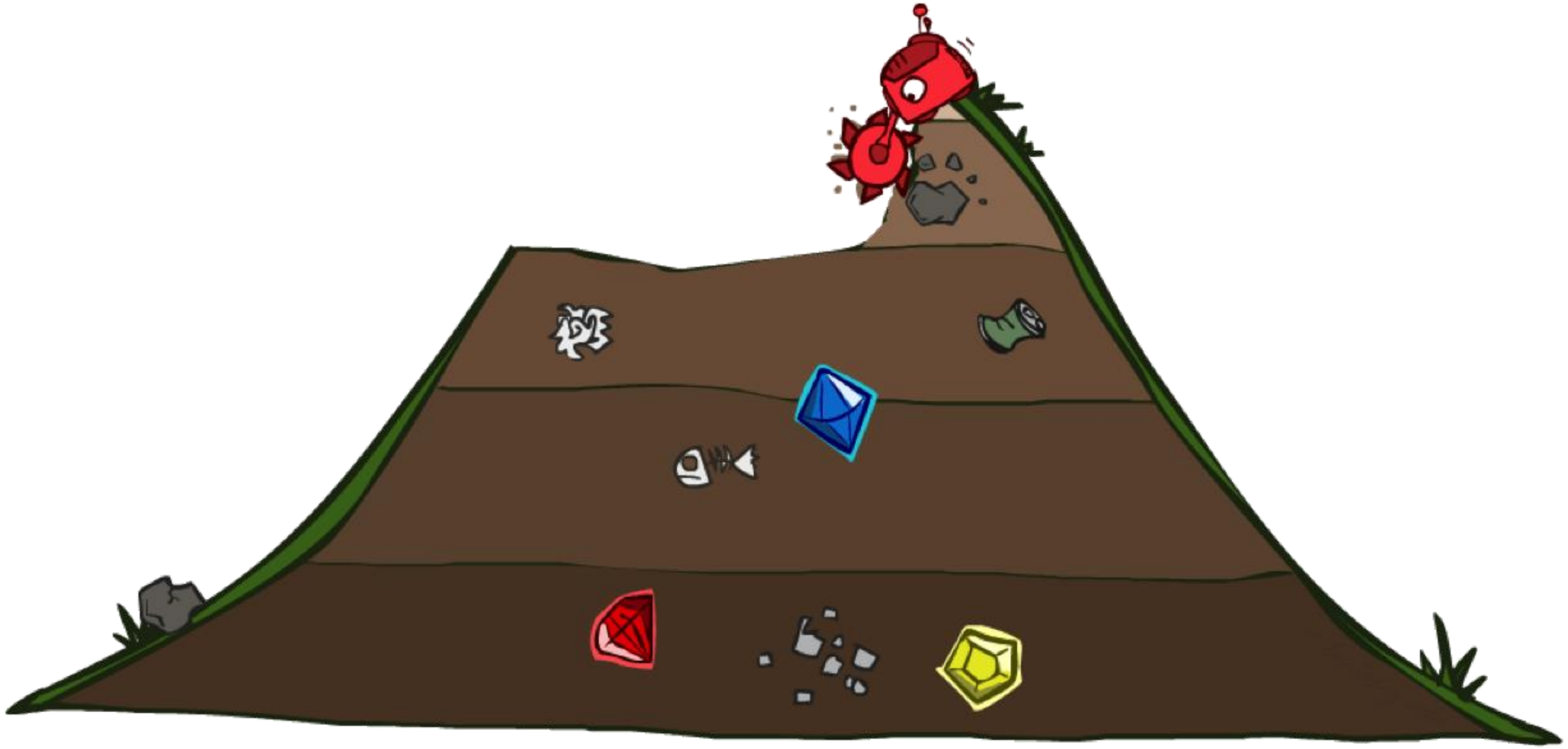
A szélességi keresés megtalálja a cselekvések száma tekintetében legrövidebb utat.
De nem találja meg a legkisebb költségű utat.
Tekintsünk egy hasonló algoritmust, ami viszont megtalálja a legkisebb költségű utat.

Mesterséges intelligencia

Problémamegoldás kereséssel

Költségérzékeny keresés

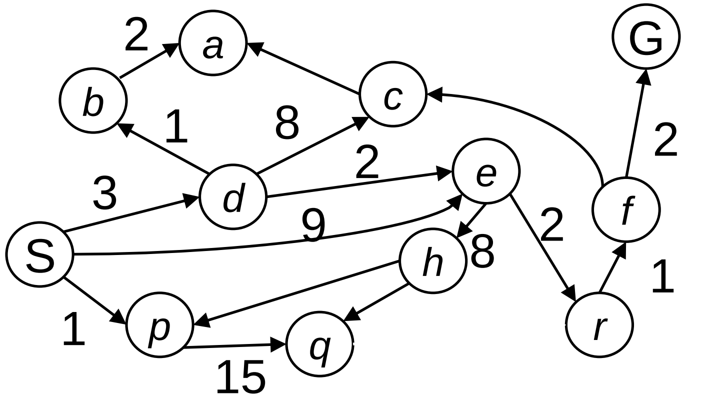
Egyenletes költségű keresés

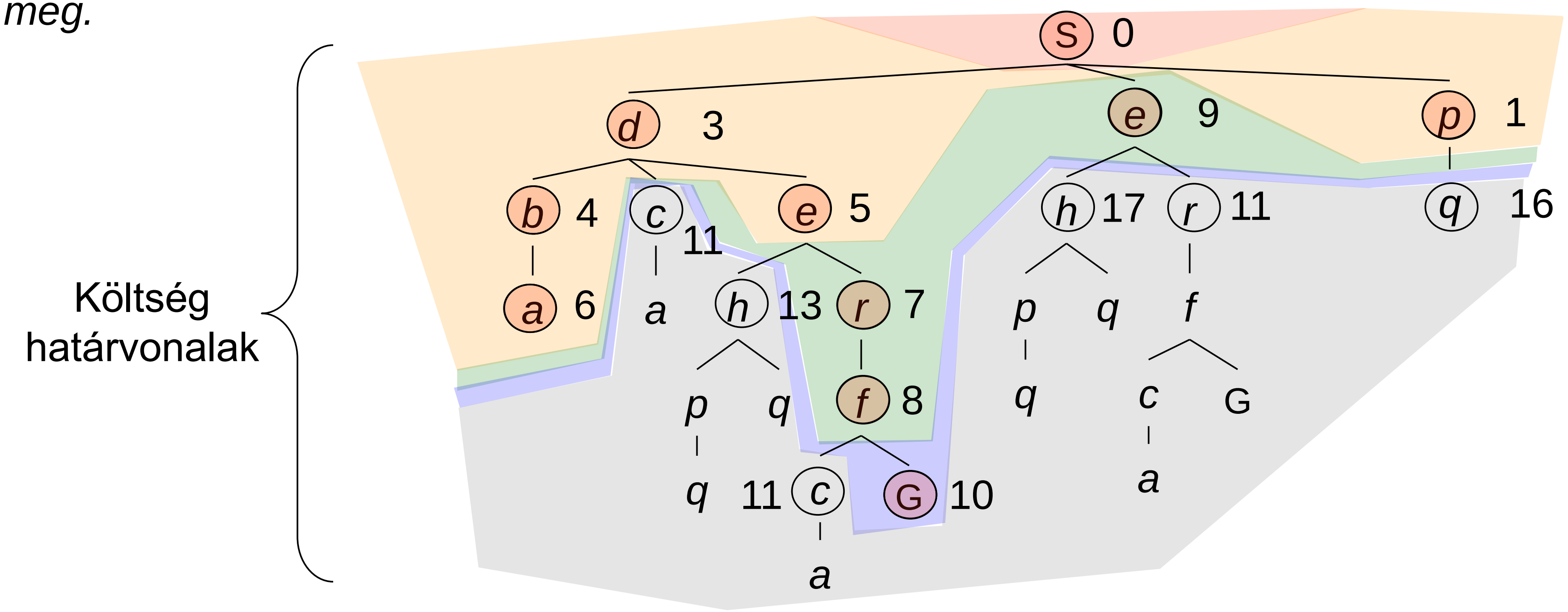


Egyenletes költségű keresés (UCS)

Stratégia: a legkisebb költségű csomópont kifejtése

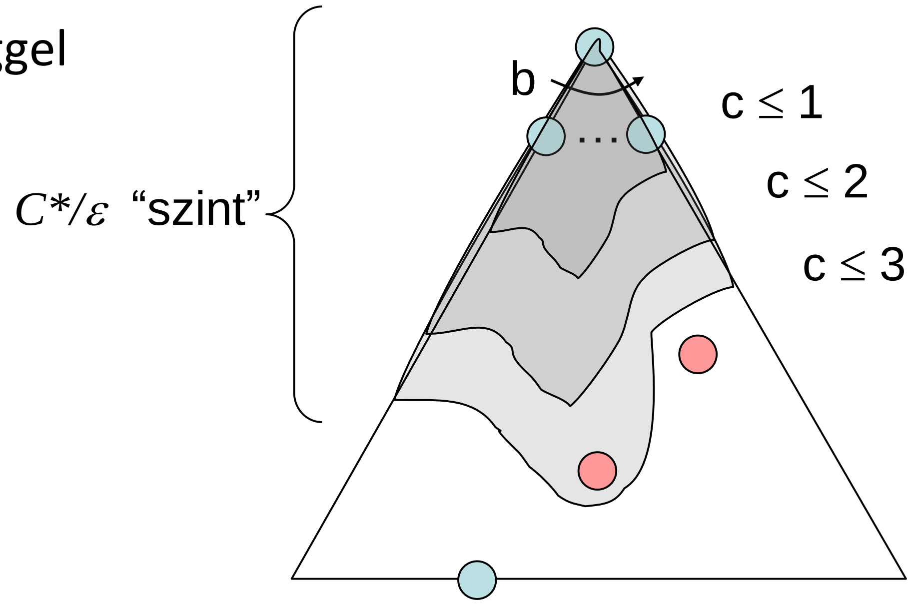
A perem egy prioritásos sor, ahol a prioritás a kumulált költségnek felel meg.





Egyenletes költségű keresés jellemzői

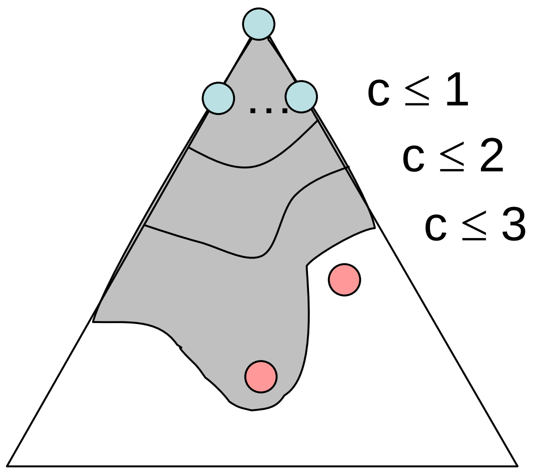
- Melyik csomópontokat fejti ki az egyenletes költségű keresés?
  - Kifejti az összes olyan csomópontot, ami kevesebb költséggel elérhető, mint a legkisebb költségű megoldás
  - Ha a megoldás C^* költségű és az élek legalább ε költségűek, akkor az effektív mélység közelítőleg: C^*/ε
  - A szükséges idő O(b$^{C^*/\varepsilon}$) (az effektív mélységben exponenciális)



- Mennyi tárhelyet igényel a perem?
  - Közelítőleg az utolsó szinttel arányos O(b$^{C^*/\varepsilon}$)
- Teljes-e?
  - Feltéve, hogy a legjobb megoldás véges költségű és a minimum élköltség pozitív, igen!
- Optimális-e?
  - Igen! (Bizonyítás később)

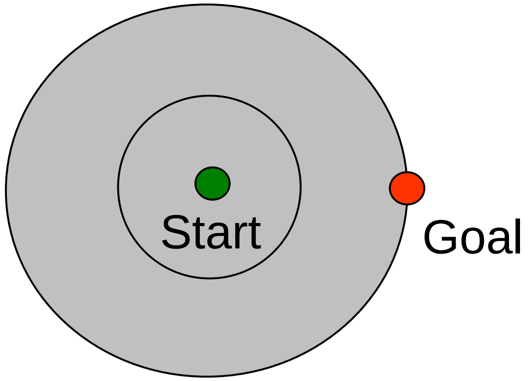
Egyenletes költségű keresés problémái

- Egyre növekvő költségű kontúrokat fejt ki



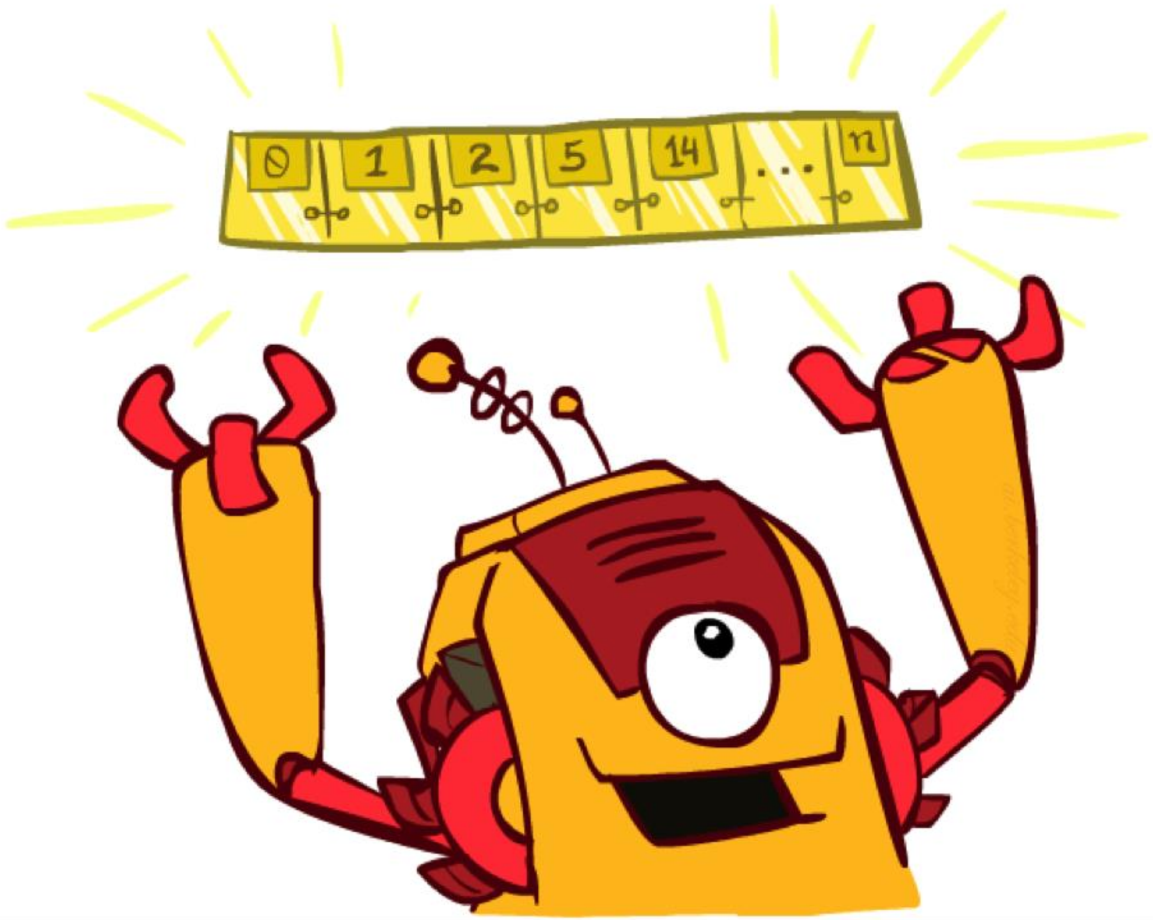
- A jó hír: teljes és optimális!

- A rossz hír:
  - Minden „irányban” feltár megoldási opciókat
  - Nem rendelkezik információval a célállapot helyzetéről



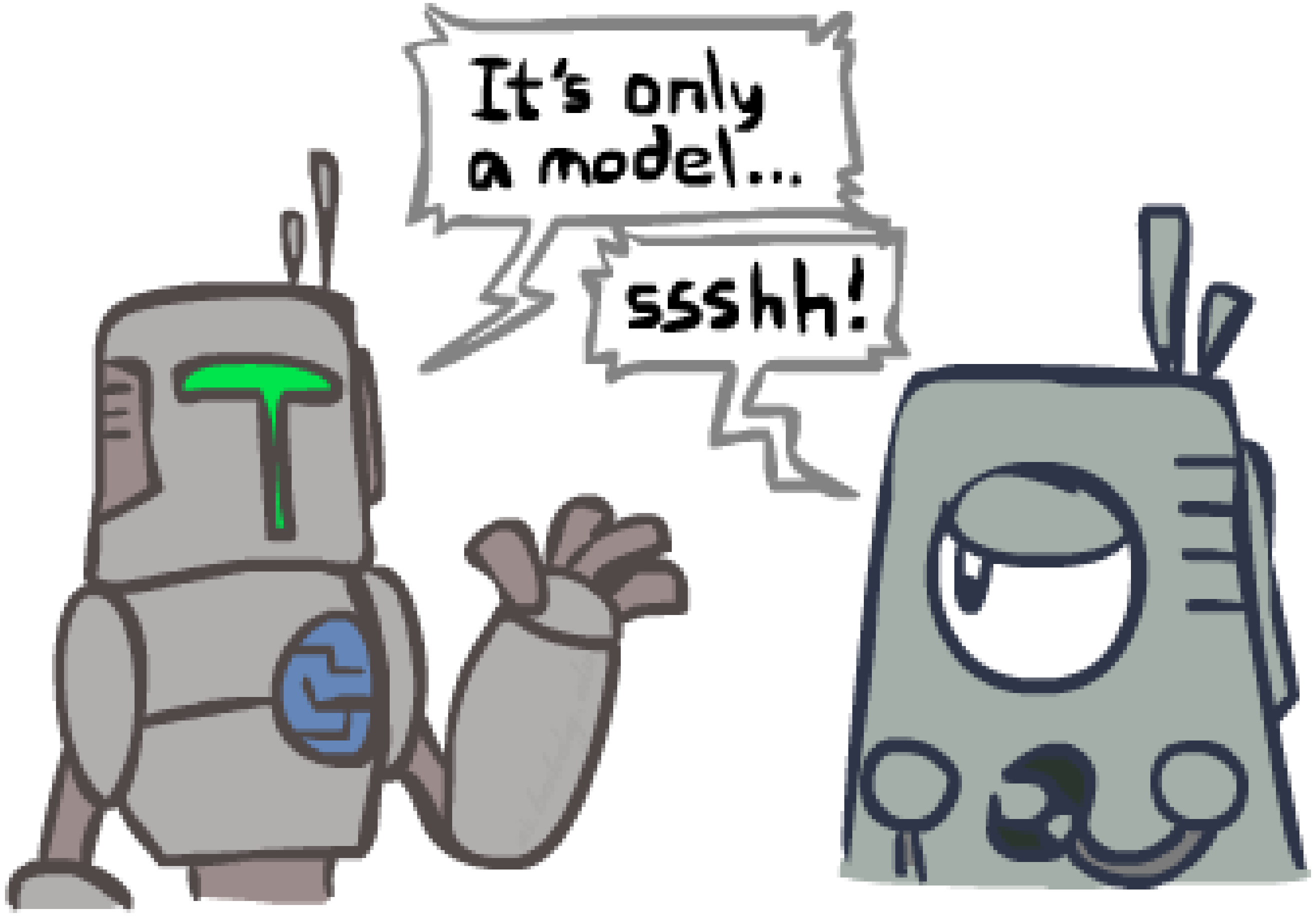
Közös strukturális elem: a sor (queue)

- A bemutatott keresési algoritmusok a perem nyilvántartási stratégiát leszámítva azonosak
  - Lényegében minden perem egy prioritásos sor (vagyis prioritással rendelkező csomópontok gyűjteménye)
  - Gyakorlatban, a mélységi és a szélességi keresés esetén megspórolható a prioritásos sor log(n) számítási igénye, ha (rendre) vermet (stack) és egy egyszerű sort (queue) alkalmazunk
  - Tehát létrehozható egy keretrendszer a keresési algoritmusok megvalósítására, amiben csak a sor megvalósítása változik



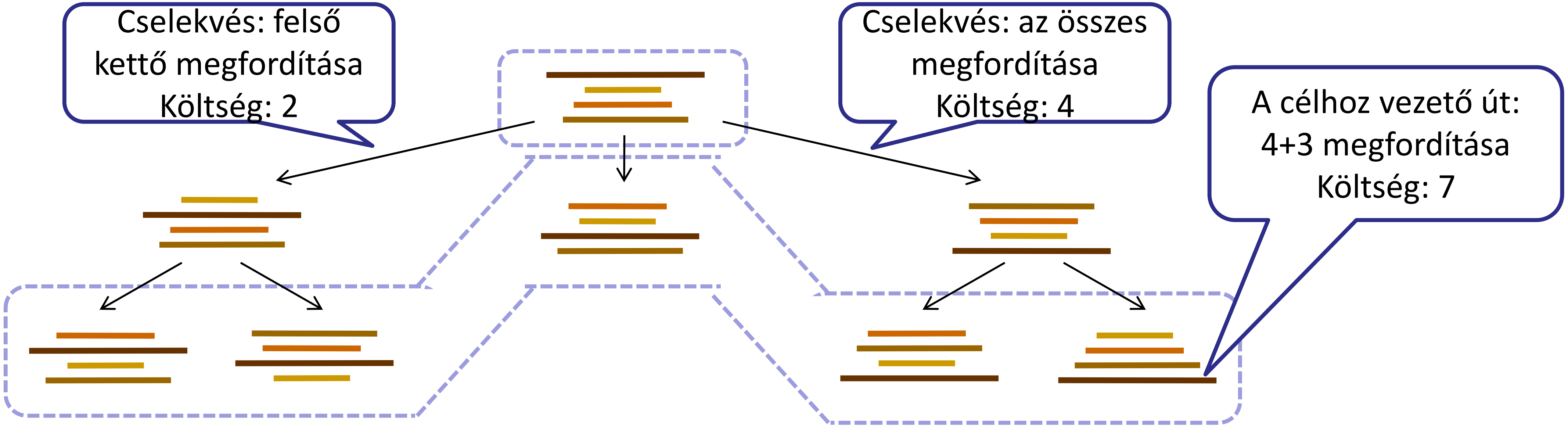
Keresés és a világ modelljei

- A keresés a világ egy modellje alapján történik
  - Az ágens nem próbálja ki az összes lehetséges tervet a valós világban
  - A tervezés „szimulációban történik”
  - A keresés csak annyira lehet jó, mint az alapjául szolgáló modellek



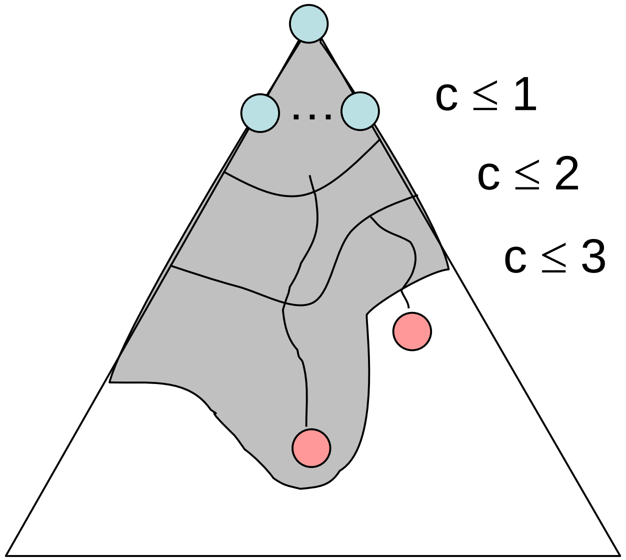
Általános fa keresés

function FA-KERESÉS (*probléma*, *stratégia*) **returns** *megoldás* vagy *kudarc*
    Keresési fa inicializálása a *probléma* *kiinduló állapotával*
    loop do
        If nincs jelölt a kifejtéshez **then return** *kudarc*
        Kifejtendő levél csomópont kijelölése a *stratégia* alapján
        If a csomópont tartalmaz egy *célállapotot* **then return** *megoldás*
        else csomópont kifejtése és az eredmény csomópontok hozzáadása a keresési fához
    end



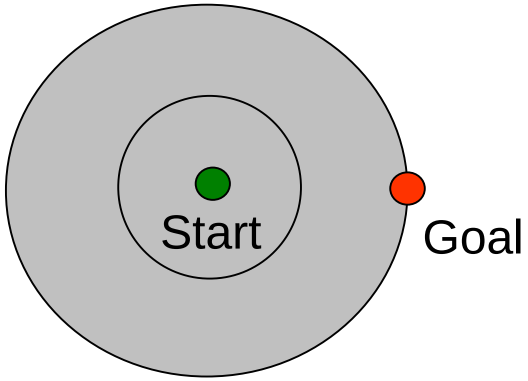
Egyenletes költségű keresés

- Stratégia: legkisebb költségű út kifejtése



- Teljes és optimális!

- A probléma:
  - Minden „irányban” feltár megoldási opciókat
  - Nem rendelkezik információval a célállapot helyzetéről



Kétirányú keresés

- egyszerre előrefelé a kiinduló állapotból, illetve hátrafelé a cél állapotból

- a keresés akkor fejeződik be, ha a két keresés valahol találkozik

$$O(2 \times b^{d/2})=O(b^{d/2})$$

Például: $b = 10$, $d = 6$:

a szélességi keresés = **1.111.111** csomópont,

a kétirányú keresés mindkét irányban 3 mélységnél ér célba = **2.222** csomópontot generál.

Kétirányú keresés

Elméletben nagyon jó, az implementálás nem triviális.
- célállapotból hátrafelé keresni? Az *n* csomópont **előd csomópontjai** azon csomópontok, amelyek követő csomópontja *n*.

- **A hátrafelé keresés** a cél csomópontból indulva az előd csomópontok egymást követő generálását jelenti. Megfordítható-e a folyamat?

- Ha az összes operátor reverzibilis, akkor az előd és követő halmazok azonosak.
- Néhány probléma esetén azonban az elődök meghatározása nagyon nehéz.

Mi van, ha nagyon sok cél állapot létezik?

a cél állapotok egy ***explicit*** listája

a cél állapotok egy ***leírása***

A neminformált keresési stratégiák összehasonlítása

b - elágazási tényező,
m - a keresési fa maximális mélysége,
d - a megoldás mélysége,
l - a mélység korlát.

Jellem-ző	Szélességi keresés	Egyenletes költségű keresés	Mélységi keresés	Mélység korlátozott keresés	Iteratívan mélyülő keresés	Kétirányú keresés
Idő-igény	b^d	b^d	b^m	b^l	b^d	$b^{d/2}$
Tár-igény	b^d	b^d	bm	bl	bd	$b^{d/2}$
Opt.?	Igen (ha…)	Igen	Nem	Nem	Igen	Igen
Teljes?	Igen	Igen	Nem	Igen, ha $l \geq d$	Igen	Igen

Mesterséges intelligencia előadássorozat

Az előadás diái az AIMA könyvre épülve (http://aima.cs.berkeley.edu) készültek a University of California, Berkeley mesterséges intelligencia kurzusának anyagainak felhasználásával (http://ai.berkeley.edu).

These slides are based on the AIMA book (http://aima.cs.berkeley.edu) and were adapted from the AI course material of University of California, Berkeley (http://ai.berkeley.edu).

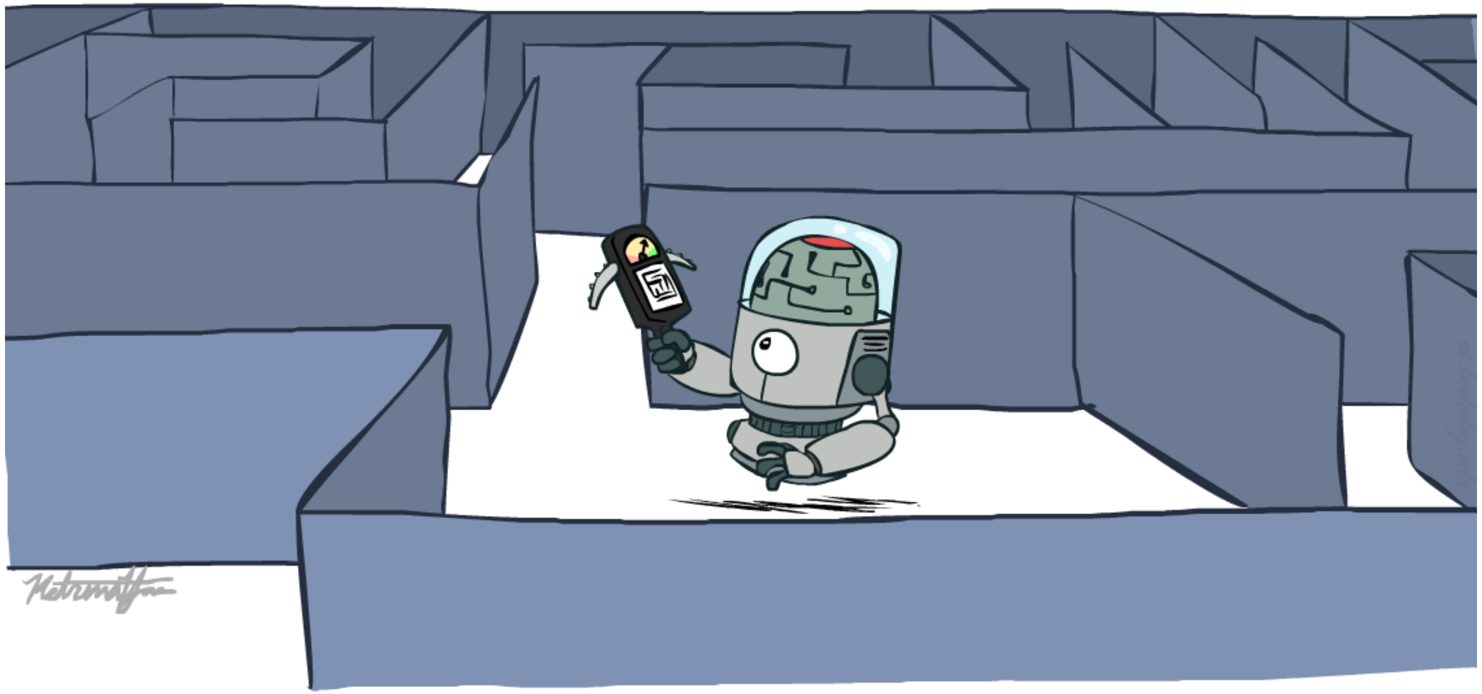
Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Méréstechnika és Információs rendszerek Tanszék

Mesterséges intelligencia

Problémamegoldás kereséssel 2.

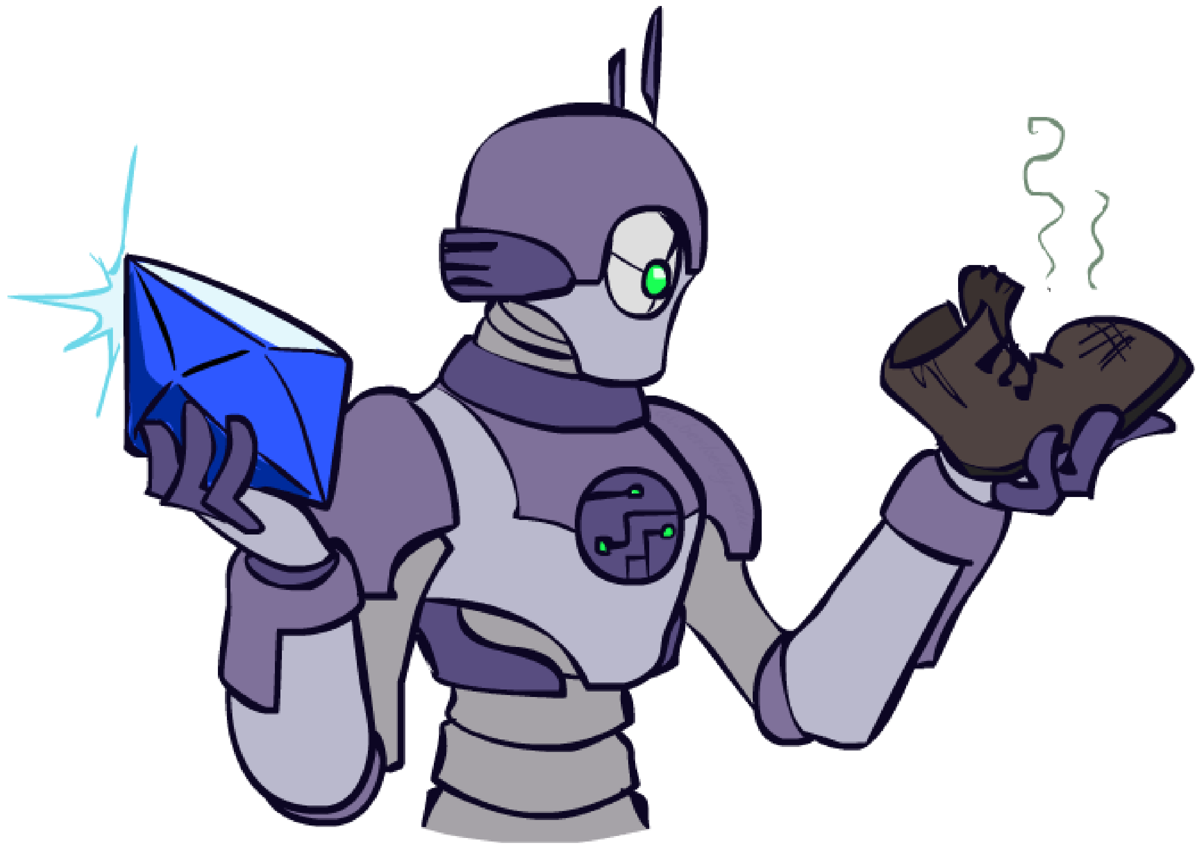
Előadó: Dr. Hullám Gábor

Informált keresés

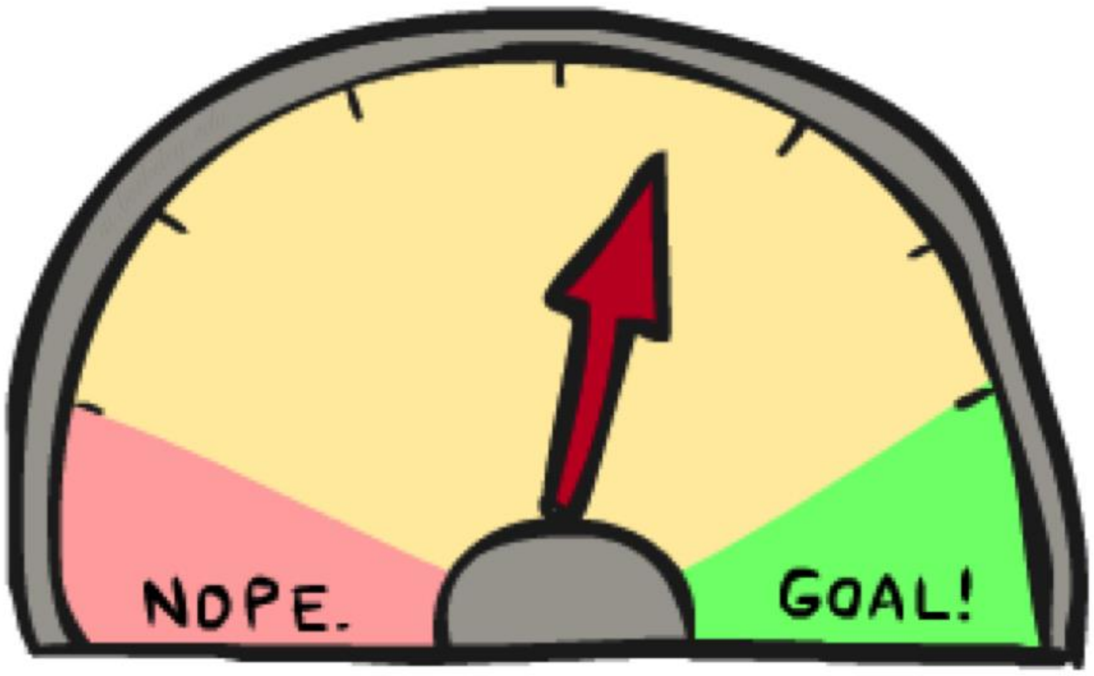


Tartalom

- Informált keresés
  - Heurisztikák
  - Mohó keresés
  - A* keresés
- Gráf keresés

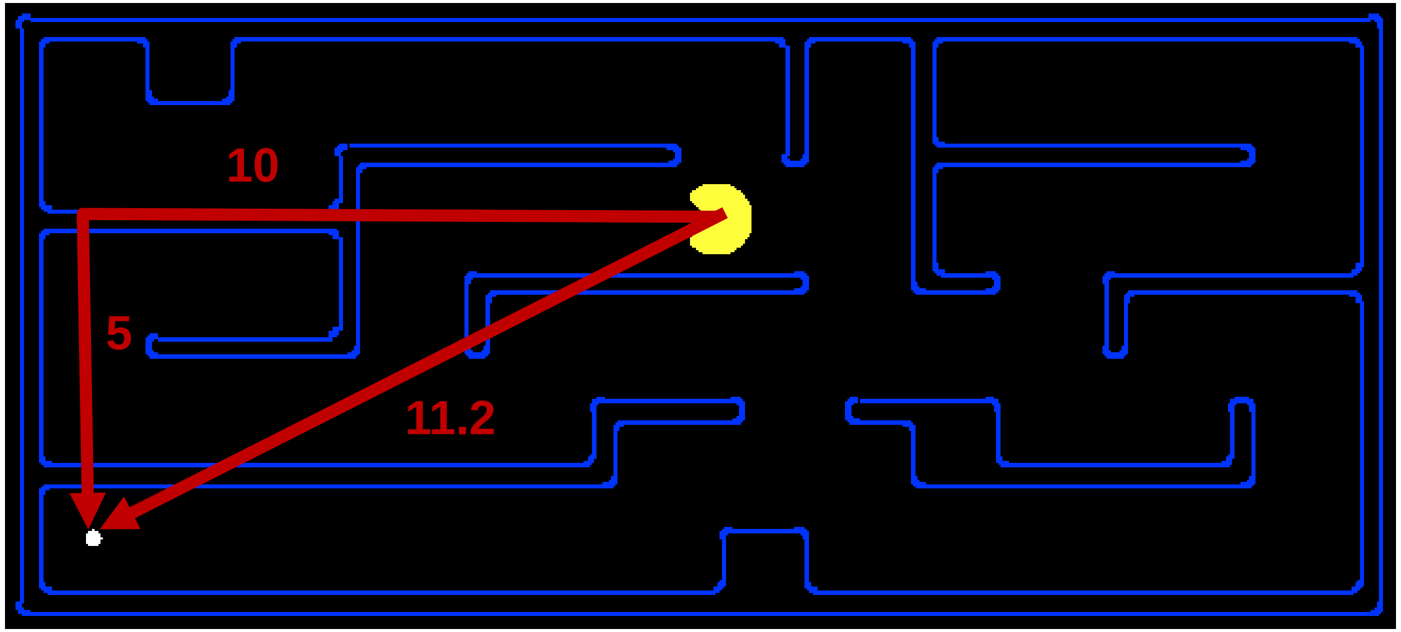


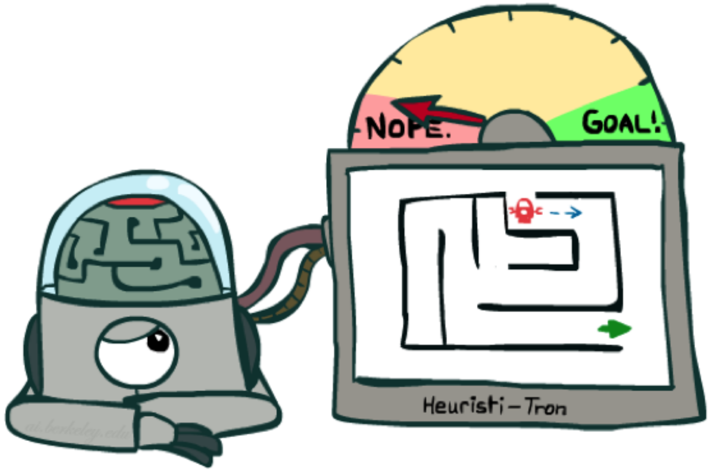
Informált keresés

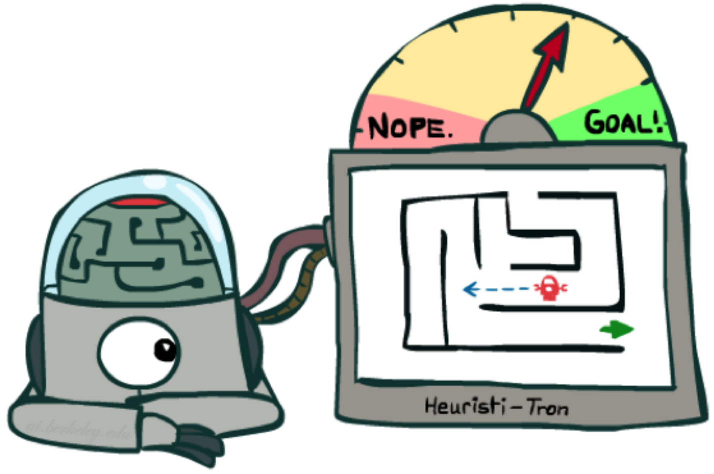


Keresési heurisztika

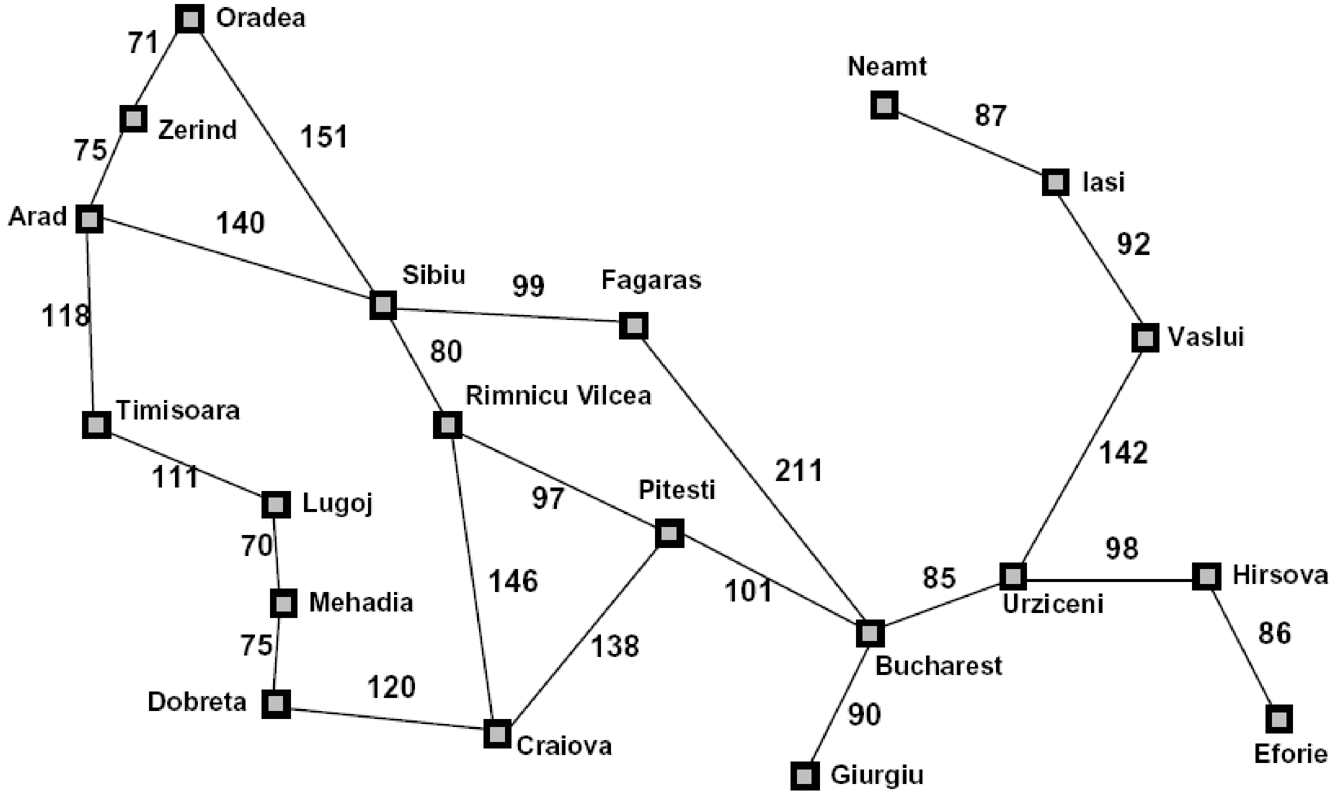
- A heurisztika:
  - Egy függvény, ami becslést ad arra, hogy az adott állapot mennyire van közel a célállapothoz
  - Egy adott keresési problémához kell tervezni
  - Példák: Manhattan távolság, euklideszi távolság (útvonal keresésnél)







Példa heurisztika: légvonalbeli távolság



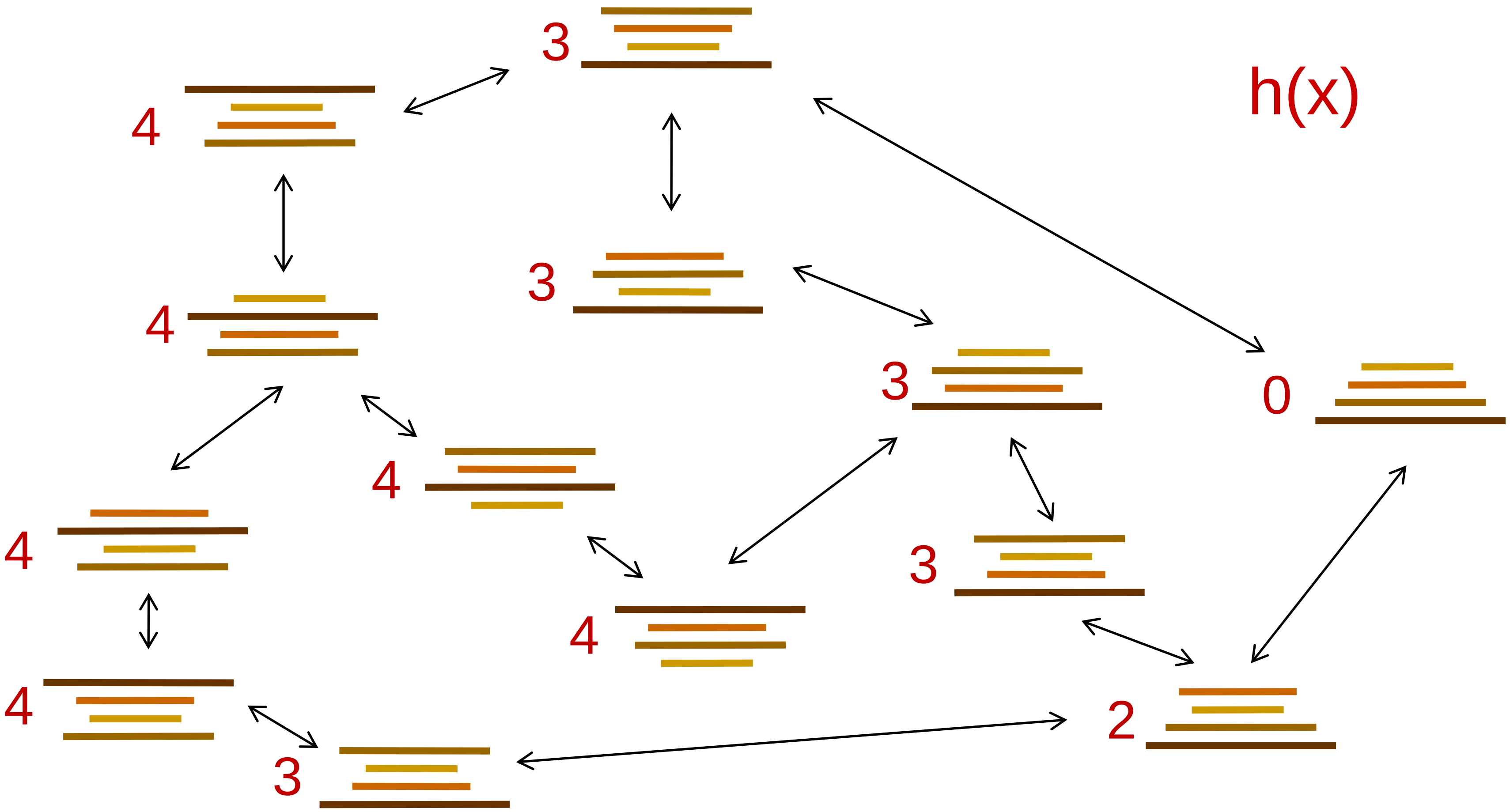
Straight−line distance
to Bucharest

Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

h(x)

Példa heurisztika

Heurisztika: a legnagyobb rossz helyen lévő palacsinta „sorszáma”



Keresési stratégiák

Heurisztikus, vagy más néven **informált keresés**

- Büntessük a hátra (céltól elfele) keresést!
- De ugyanaz, és könnyebb díjazni az előrekeresést a célállapot irányába.

Ehhez kell valami elképzelés, hogy a cél:

- ***merrefelé és,***
- ***nagyjából milyen messzire fekszik*****.**

Ez az információ az ún. **heurisztika**, **heurisztikus függvény h(*n*)**,

Keresési stratégiák – 2

Heurisztika, **heurisztikus függvény h(*n*)**

- a probléma minden *n* állapotára ki kell tudnunk számítani
  kifejezi a **célig előrehaladás becsült költség**ét

- ha pontos, akkor elvben fölöslegessé teszi a keresést

(ha nagyon pontatlan, akkor viszont semmit sem segít)

Ez minden problémára más, problémaspecifikus!

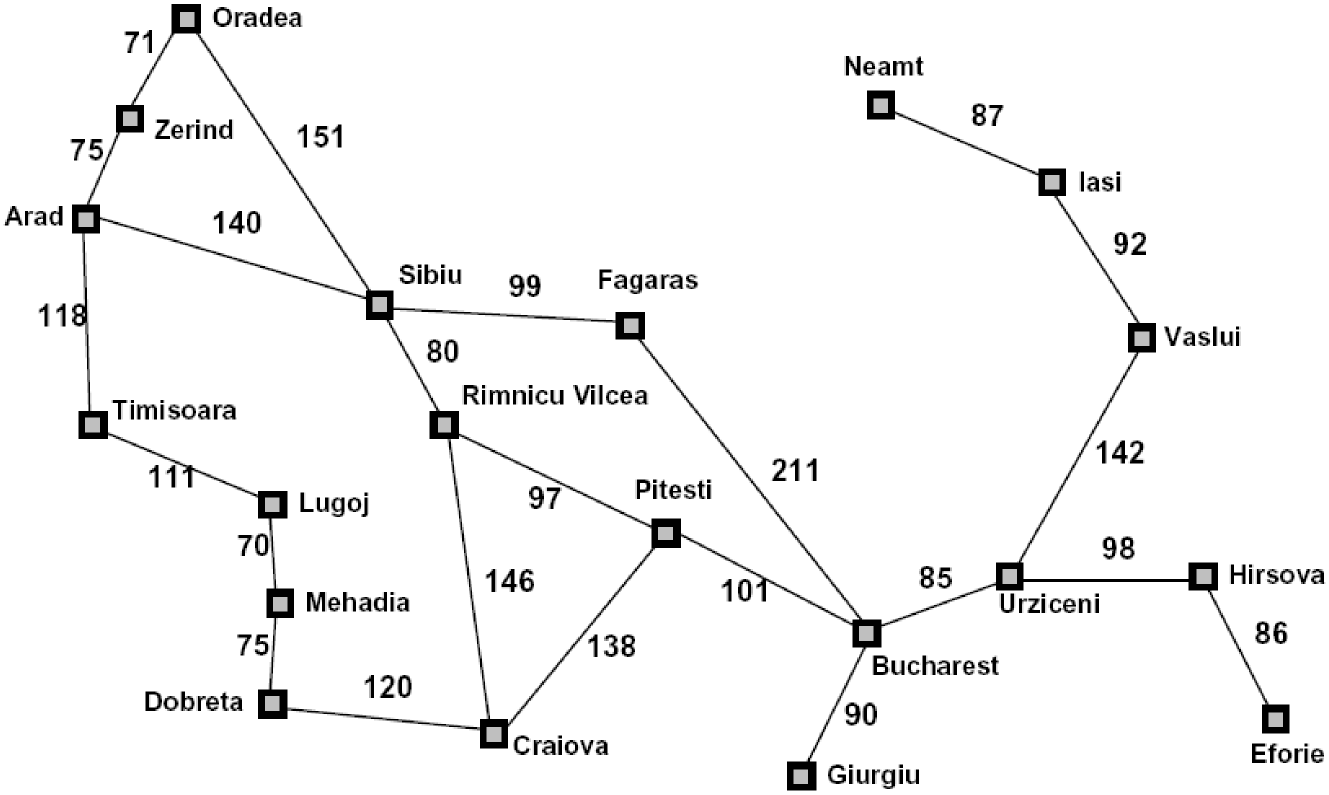
$$f(n) = g(n) + h(n)$$

A heurisztikus függvény, h(*n*) becslést ad arra, hogy mekkora a hátralévő út legkisebb költsége.

- Ezáltal becslést ad a teljes út legjobb költségére is (f(n)).

Mohó keresés

Mohó keresés - heurisztika



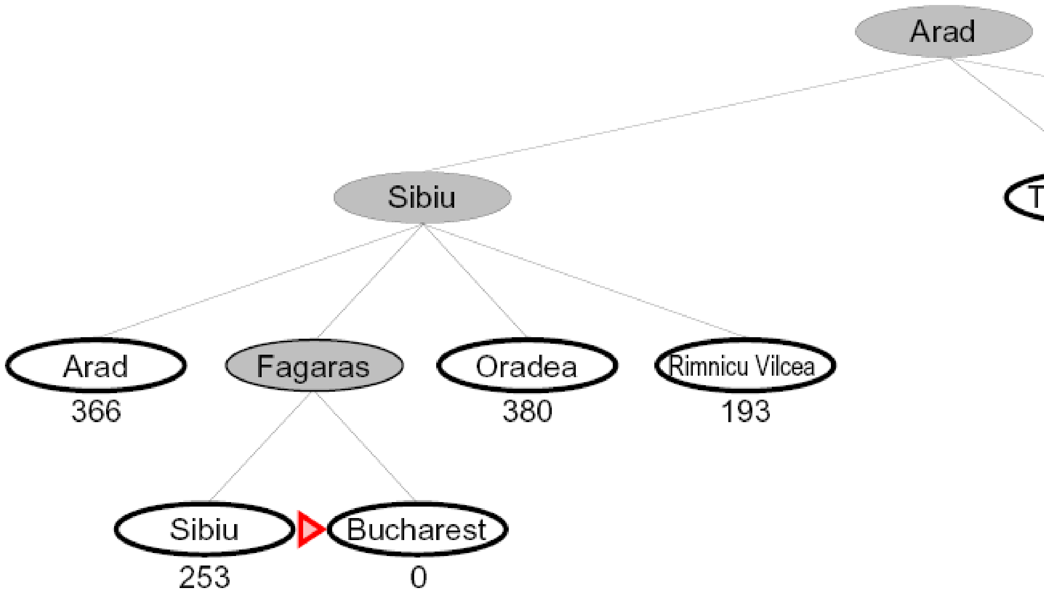
Straight−line distance to Bucharest

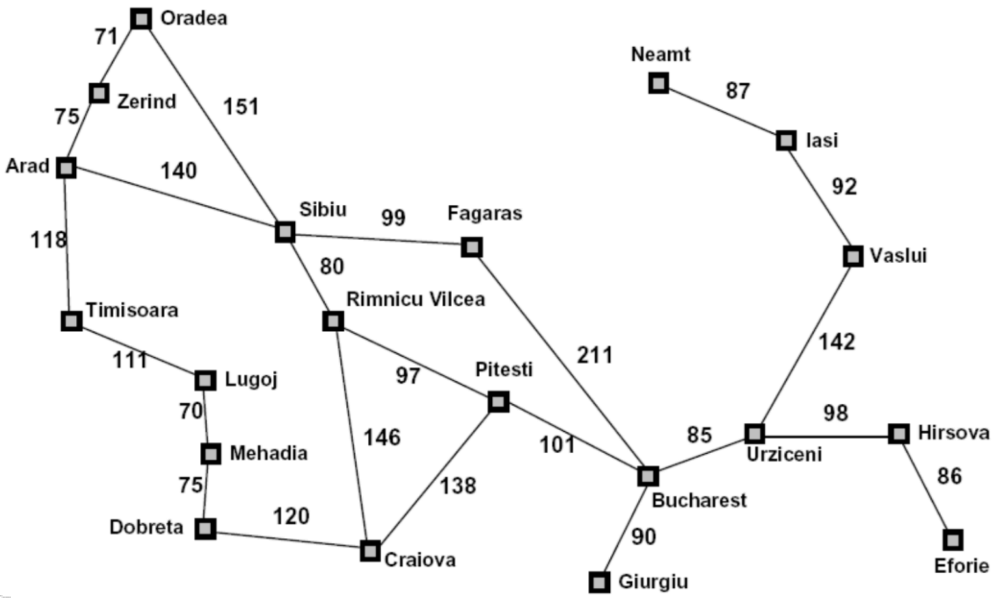
Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

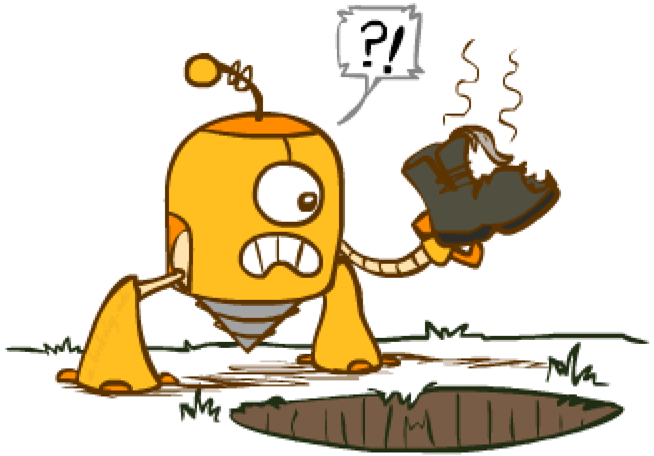
h(x)

Mohó keresés

- A legközelebbi csomópont kifejtése lenne célszerű



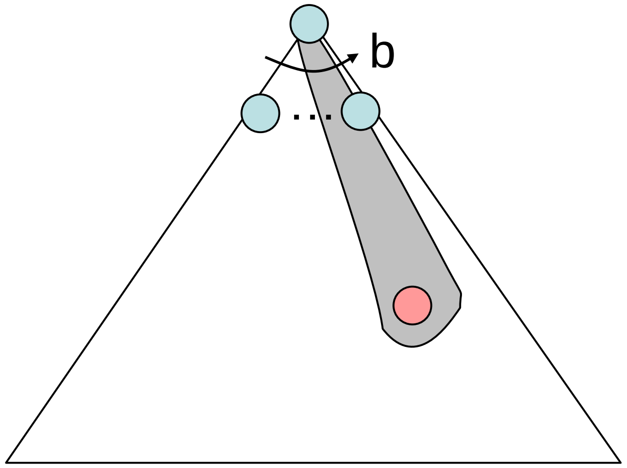




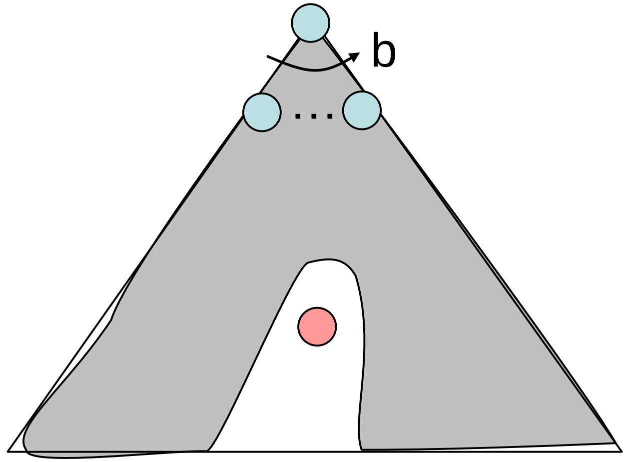
- Mi mehet félre?

Mohó keresés

- Stratégia: fejtsük ki azt a csomópontot, amiről azt gondoljuk, hogy a legközelebb van a célállapothoz
  - Heurisztika: a legközelebbi célállapottól való becsült távolság



- Gyakori esetben:
  - A legjobbat először (mohó) keresés eljuttat egy célállapothoz, ami nem biztos, hogy a legjobb

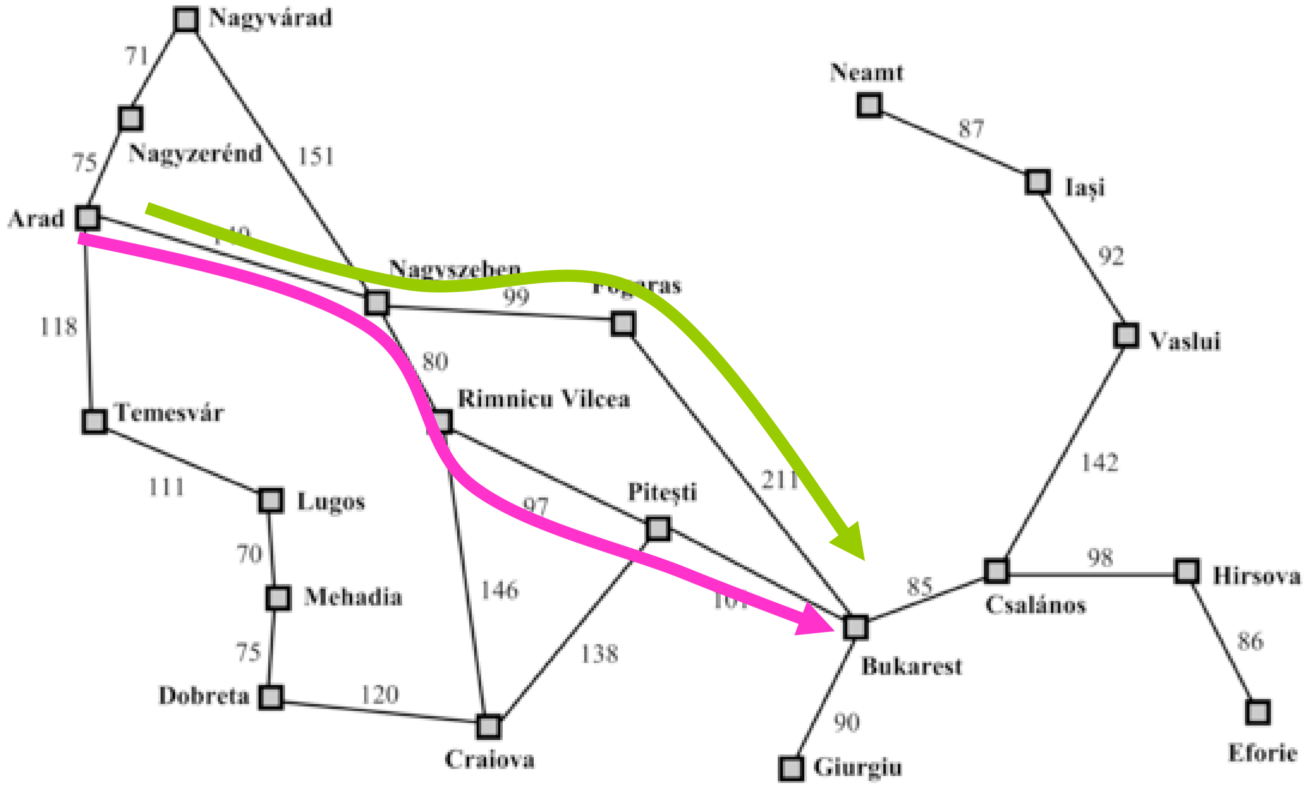


- Legrosszabb esetben:
  - egy rosszul irányított mélységi keresést kapunk

Mohó keresés

A célhoz legközelebbinek tűnő csomópont először (a legjobbnak becsültet először)

<u>Stratégia</u>: a következő lépésben azt a csp-t fejti ki, amelyhez rendelt probléma-állapotot a legközelebbinek ítéli a célállapothoz (legkisebb az n-dik csp-hoz rendelt $h(n)$ heurisztikus érték).

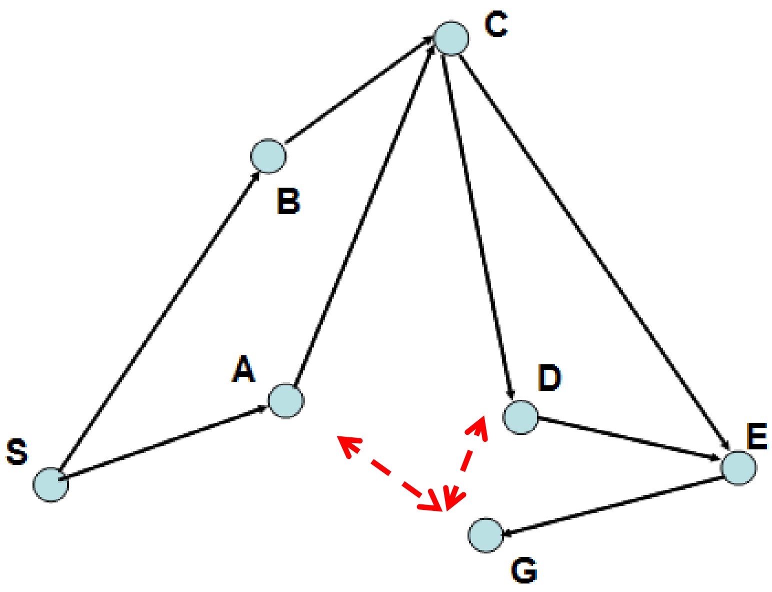


ezt kaptuk

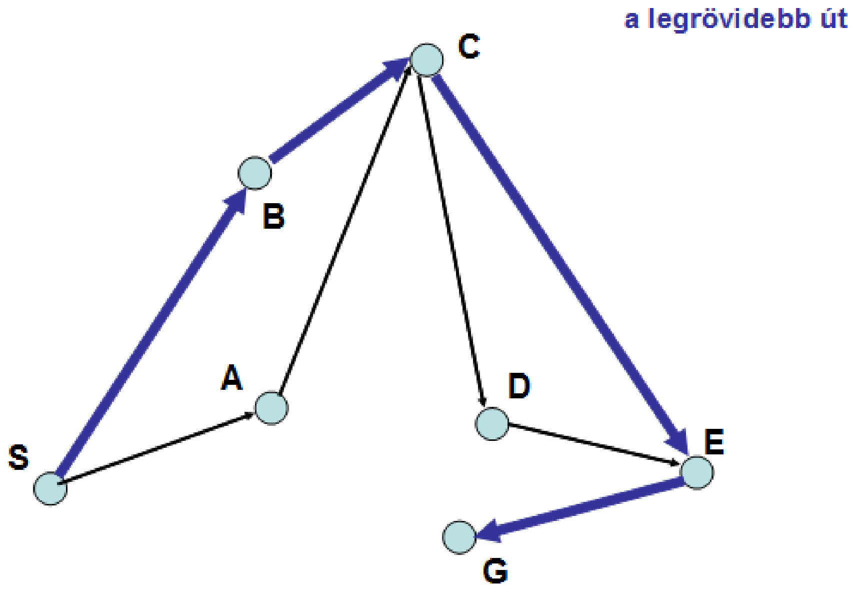
ezt kellene kapni

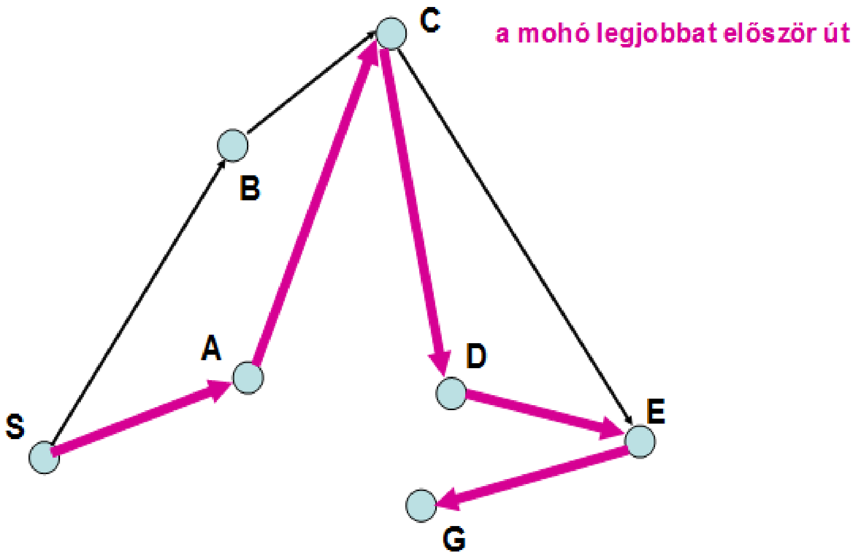
Mohó keresés

Mohó algoritmus általában **gyorsan** megtalálja a megoldást, de **nem mindig az optimális** megoldást találja meg. A mohó keresés érzékeny a hibás kezdő lépésekre is.



Probléma: A és D légvonalban nagyon közel van a G célhoz, de az úthálózaton nem





Mohó keresés

- mélységi keresésre hasonlít, egyetlen út végigkövetését preferálja a célig, zsákutcából visszalép.
- Ua. a problémák, mint a mélységi keresésnél:
  - nem optimális,
  - nem teljes (elindul egy végtelen úton, és ez esetben nem tér vissza új lehetőséget kipróbálni).
- Az összes csomópontot a memóriában tartja: ezért a legrosszabb (worst-case) idő- és tárigény:
  - $O(b^m)$ – ha m a problématér mélysége

Mesterséges intelligencia

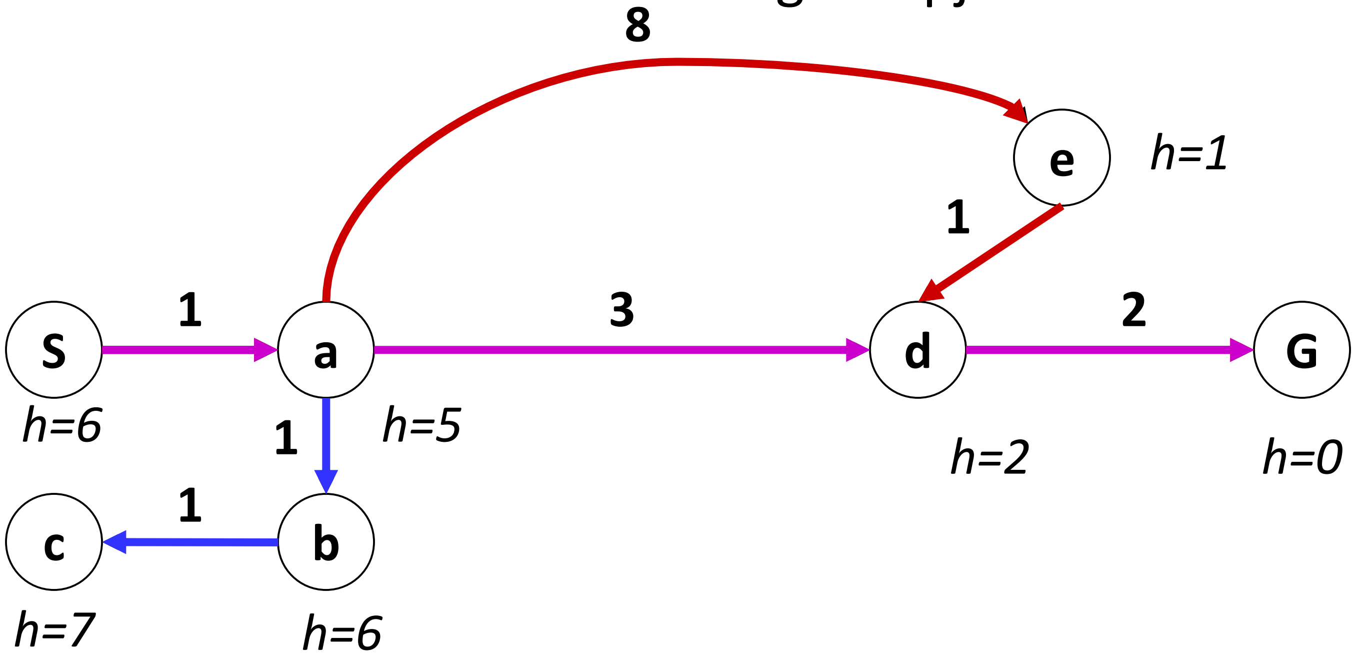
Problémamegoldás kereséssel

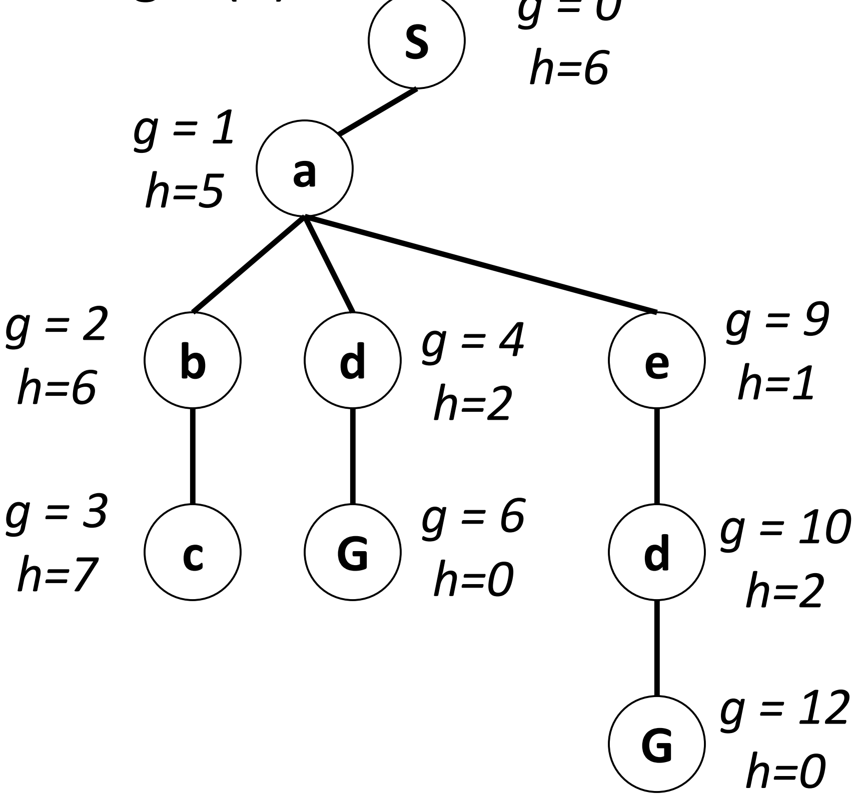
A* keresés

A* keresés

Egyenletes költségű és a mohó keresés egyesítése

- Egyenletes költségű keresés a lehetséges útvonalakat az útköltség alapján rendezi: visszamenőleges költség g(n)
- Mohó keresés a cél közelsége alapján választ: előremutató költség h(n)

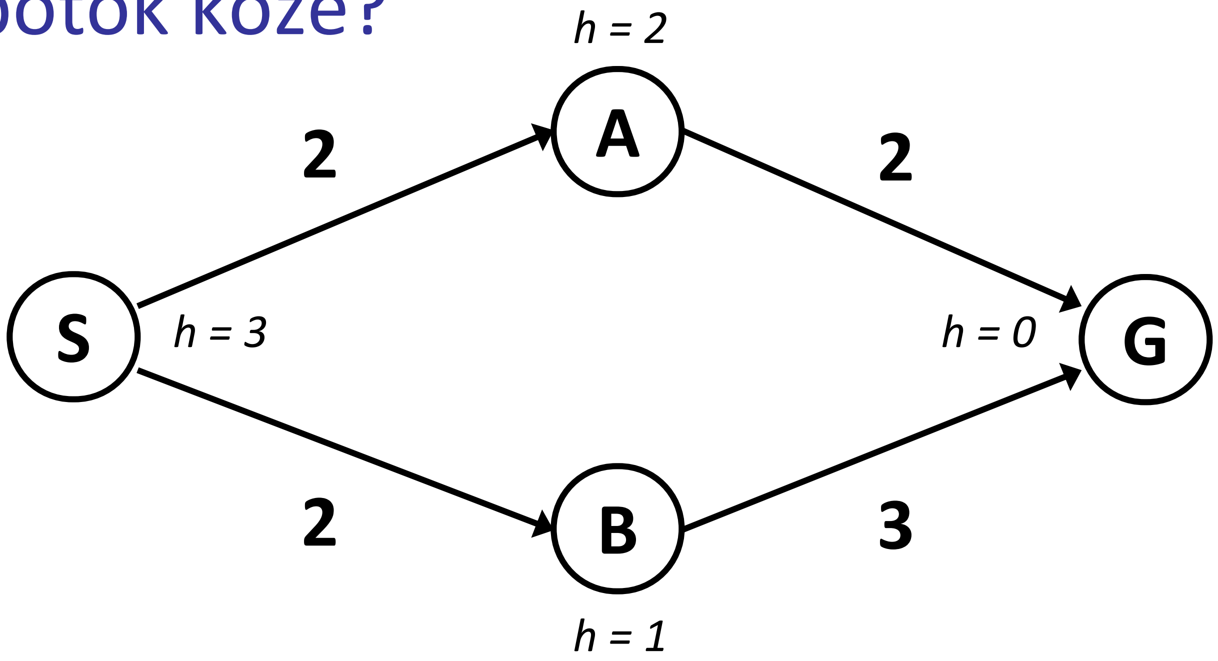




- A* keresés a kétféle költség összege alapján rendez: f(n) = g(n) + h(n)

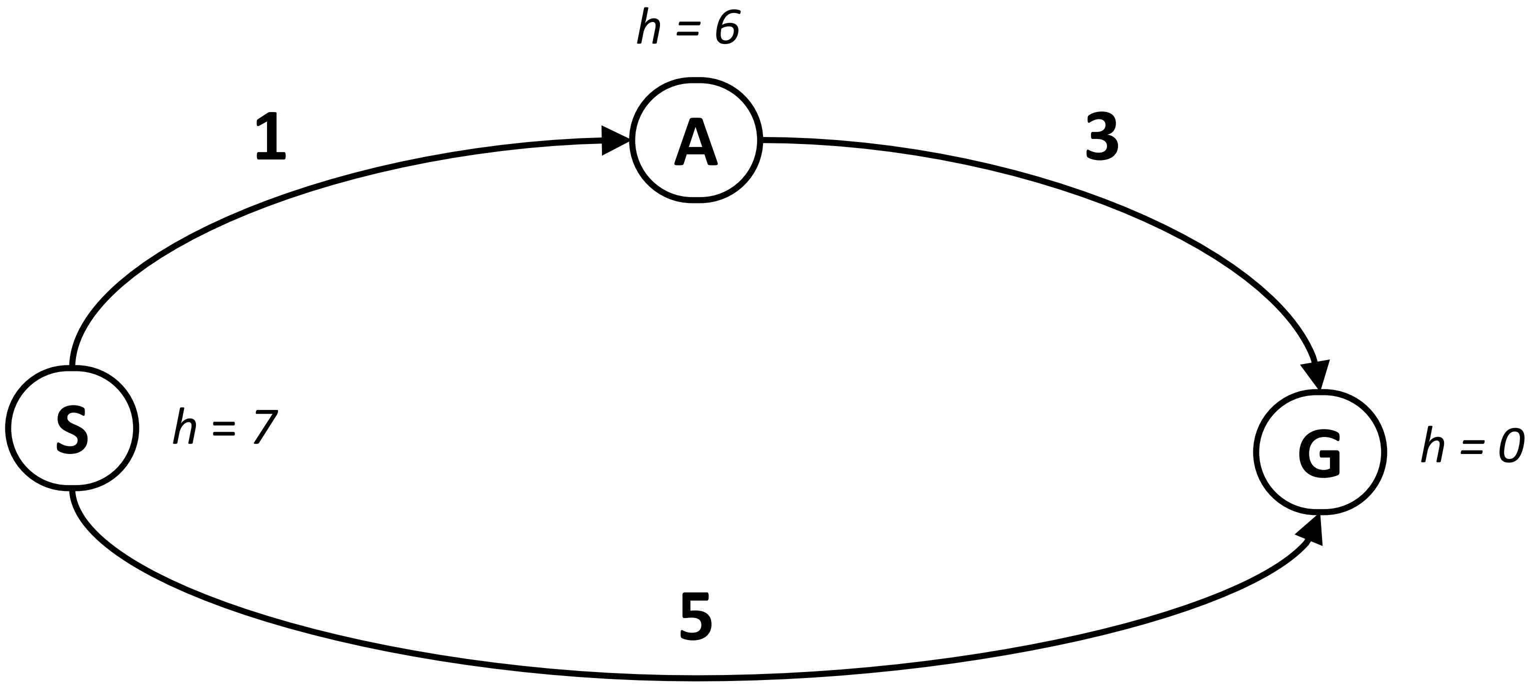
Mikor kell az A* keresésnek megállnia?

- Megállhat-e a keresés, amikor egy célállapot kerül a lehetséges követő állapotok közé?



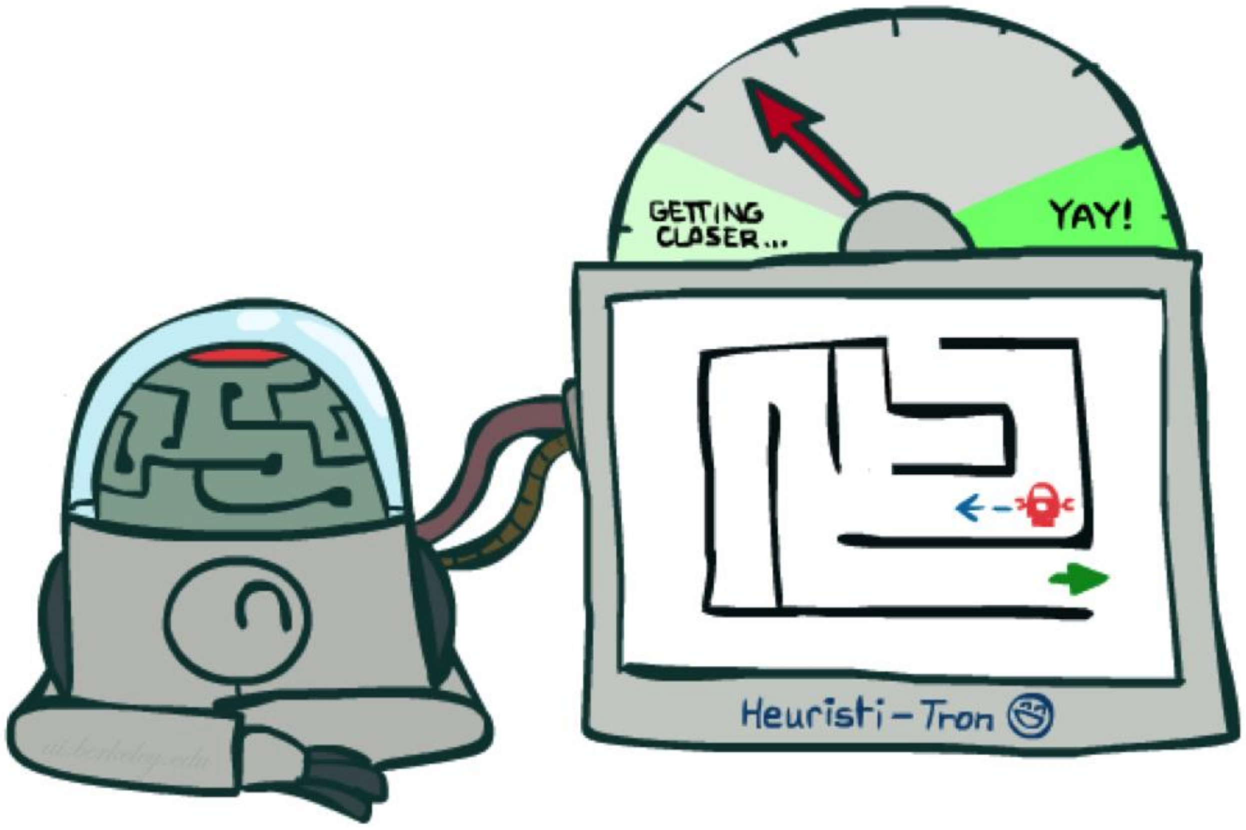
- Nem: csak akkor állhat meg, ha már kikerült az állapotok közül, és kifejtésre került

Optimális-e az A*?

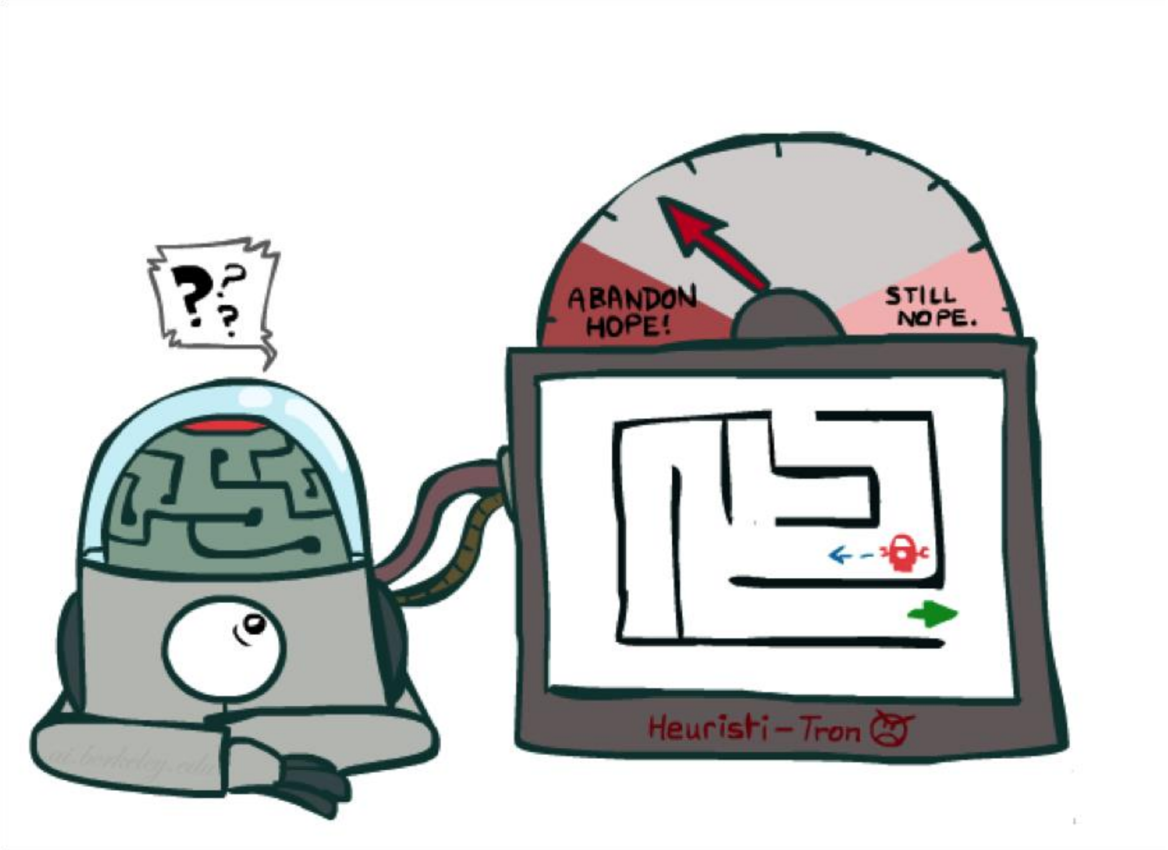


- Mi ment félre?
- A valós „rossz” célköltség < becsült „jó” célköltség
- A becsült költségnek kisebbnek kell lennie a valódi költségnél!

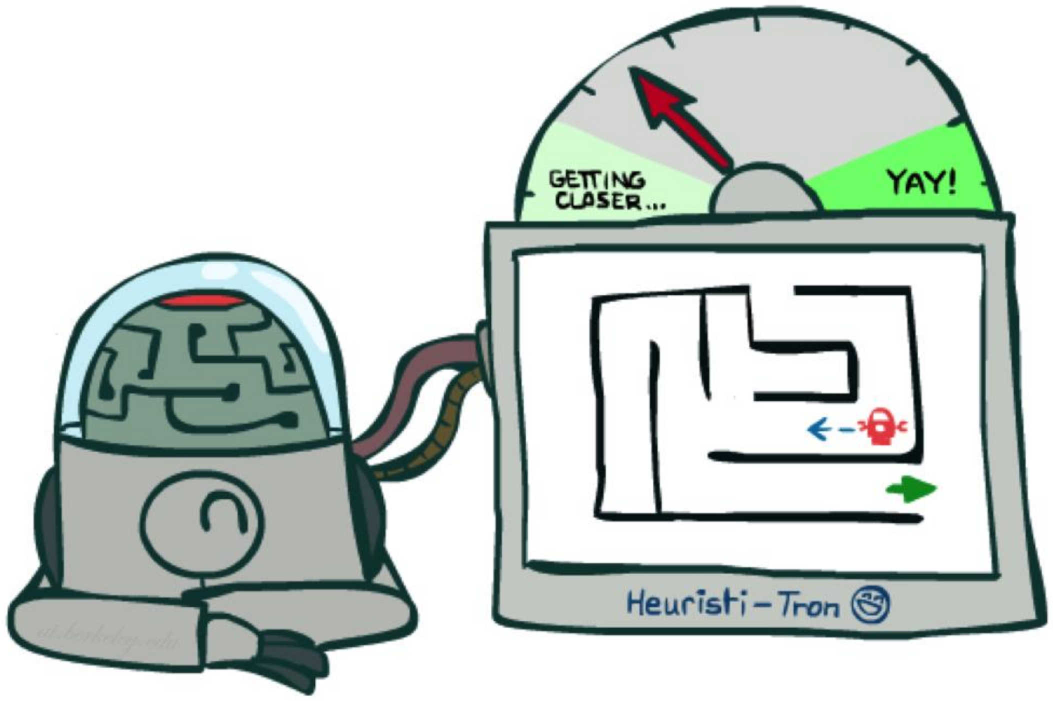
Elfogadható heurisztika



Elfogadhatóság, mint tulajdonság



Nem elfogadható (pesszimista) heurisztika megtöri az optimalitást azzal, hogy jó tervek a peremben maradnak.



Elfogadható (optimista) heurisztika „lelassítja" a rossz terveket, de sosem lép túl a valós költségeken

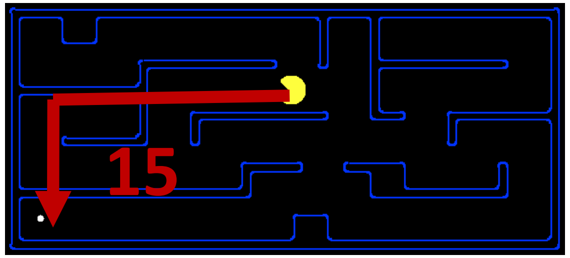
Elfogadható heurisztika

- A *h* heurisztika *elfogadható* (optimista) ha:

$$0 \leq h(n) \leq h^*(n)$$

ahol $h^*(n)$ a valós költség a legközelebbi célállapothoz

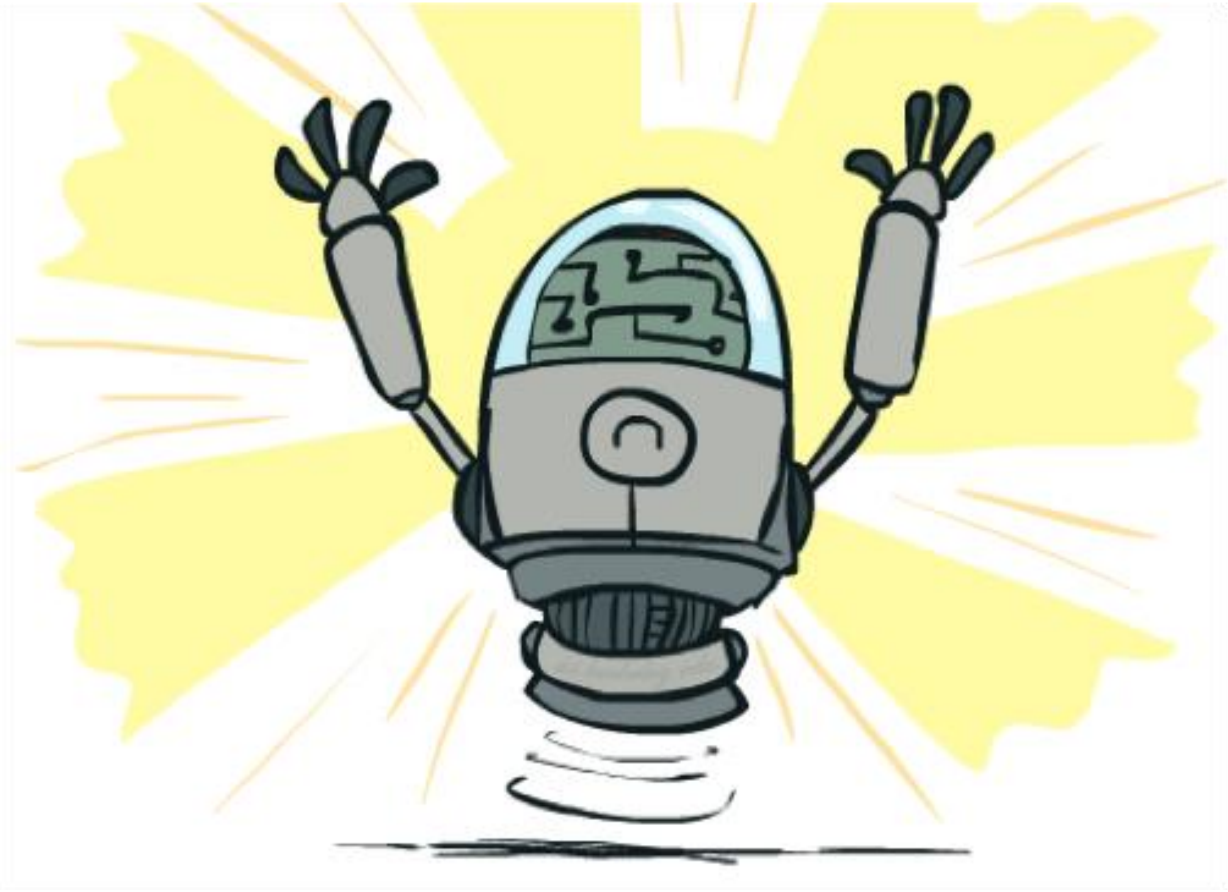
- Példák:



4

- Elfogadható heurisztikák kialakítása az A* keresés egyik lényegi pontja

Az A* keresés optimalitása



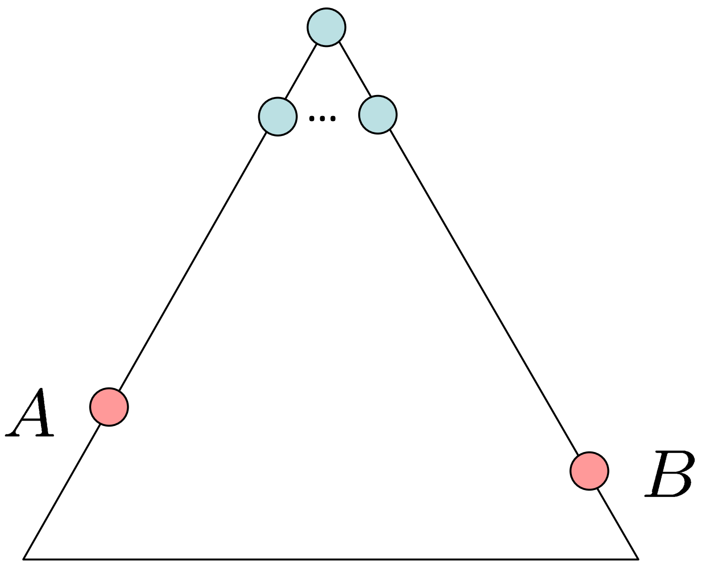
Az A* keresés optimalitása

Feltevések:

- „A” az optimális célcsomópont
- „B” a szuboptimális célcsomópont
- h elfogadható

Állítás:

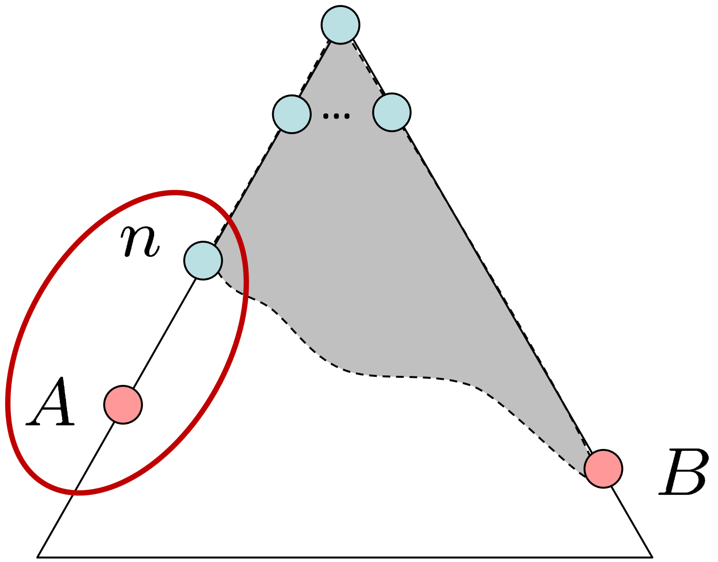
- „A” előbb kerül ki a peremből, mint „B”



Az A* keresés optimalitása: blokkolás

Bizonyítás:

- Tegyük fel, hogy „B” a peremen található
- Továbbá, hogy „A” valamely „n” őse szintén a peremen található (vagy maga „A” is)
- Állítás: „*n*” kifejtésére előbb kerül sor, mint „B” kifejtésére
    1. f(n) kisebb egyenlő, mint f(A)

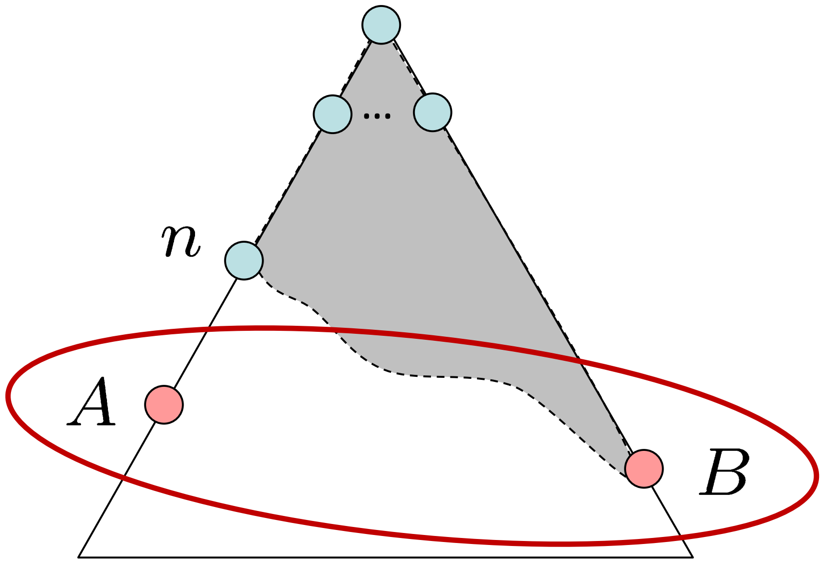


$f(n) = g(n) + h(n)$	f-költség definíció
$f(n) \leq g(A)$	h elfogadhatósága
$g(A) = f(A)$	a célnál h = 0

Az A* keresés optimalitása: blokkolás

Bizonyítás:

- Tegyük fel, hogy „B” a peremen található
- Továbbá, hogy „A” valamely „n” őse szintén a peremen található (vagy maga „A” is)
- Állítás: „*n*” kifejtésére előbb kerül sor, mint „B” kifejtésére
  1. f(n) kisebb egyenlő, mint f(A)
  2. f(A) kisebb, mint f(B)



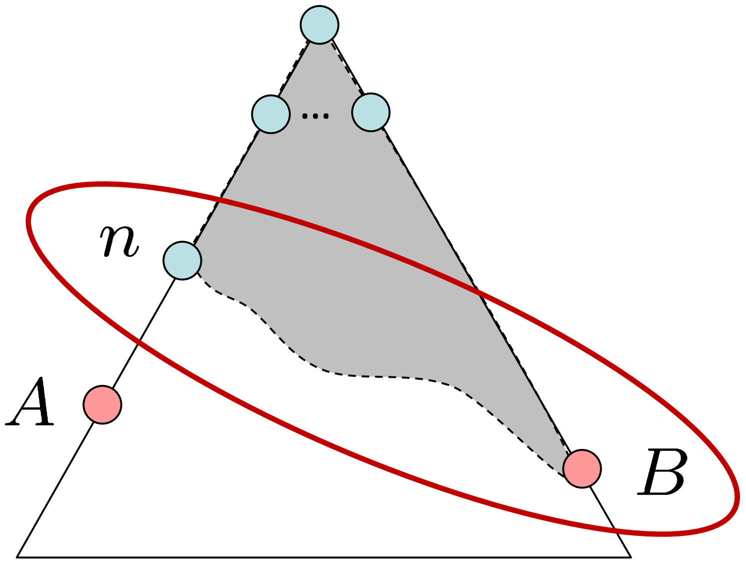
$g(A) < g(B)$ B szuboptimális

$f(A) < f(B)$ h = 0 a célnál

Az A* keresés optimalitása: blokkolás

Bizonyítás:

- Tegyük fel, hogy „B” a peremen található.
- Továbbá, hogy „A” valamely „n” őse szintén a peremen található (vagy maga „A” is).
- Állítás: *„n”* kifejtésére előbb kerül sor, mint „B” kifejtésére.
    1. f(n) kisebb egyenlő, mint f(A)
    2. f(A) kisebb, mint f(B)
    3. *„n”* előbb kerül kifejtésre, mint „B”
- „A” minden őse előbb kerül kifejtésre, mint „B”.
- „A” kifejtésre kerül „B” előtt.
- A* keresés optimális.



$$f(n) \leq f(A) < f(B)$$

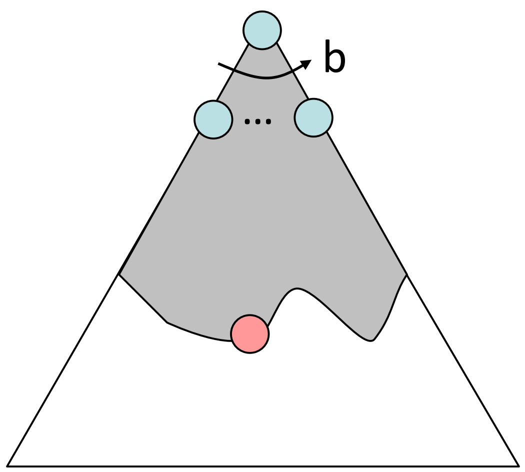
Mesterséges intelligencia

Problémamegoldás kereséssel
Informált keresés

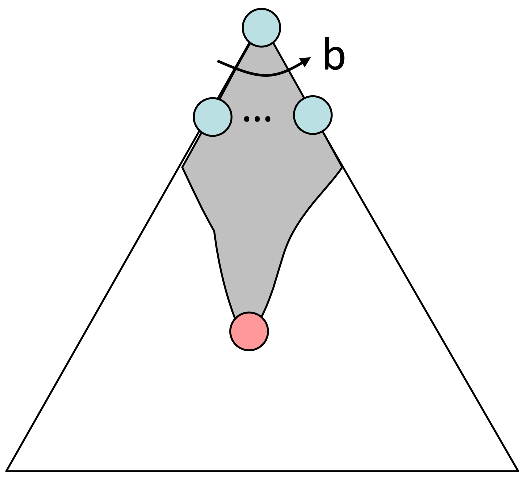
A* keresés további tulajdonságai

A* keresés tulajdonságai

Egyenletes költségű keresés

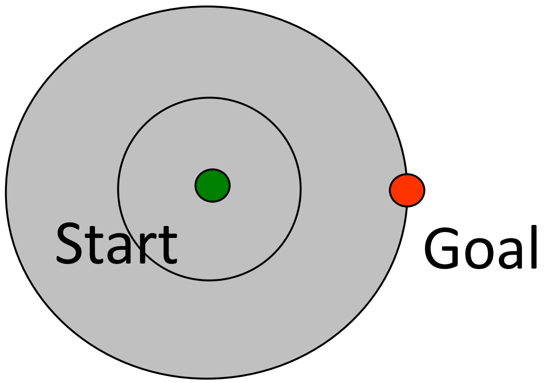


A*

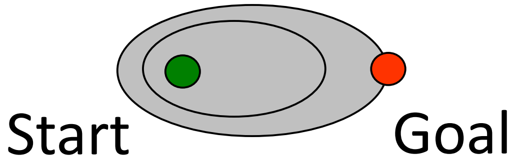


Egyenletes keresés vs A* határvonalak

- Egyenletes költségű keresés minden „irányban” egyenlően végez kifejtést



- A* főleg a cél irányában végez kifejtést



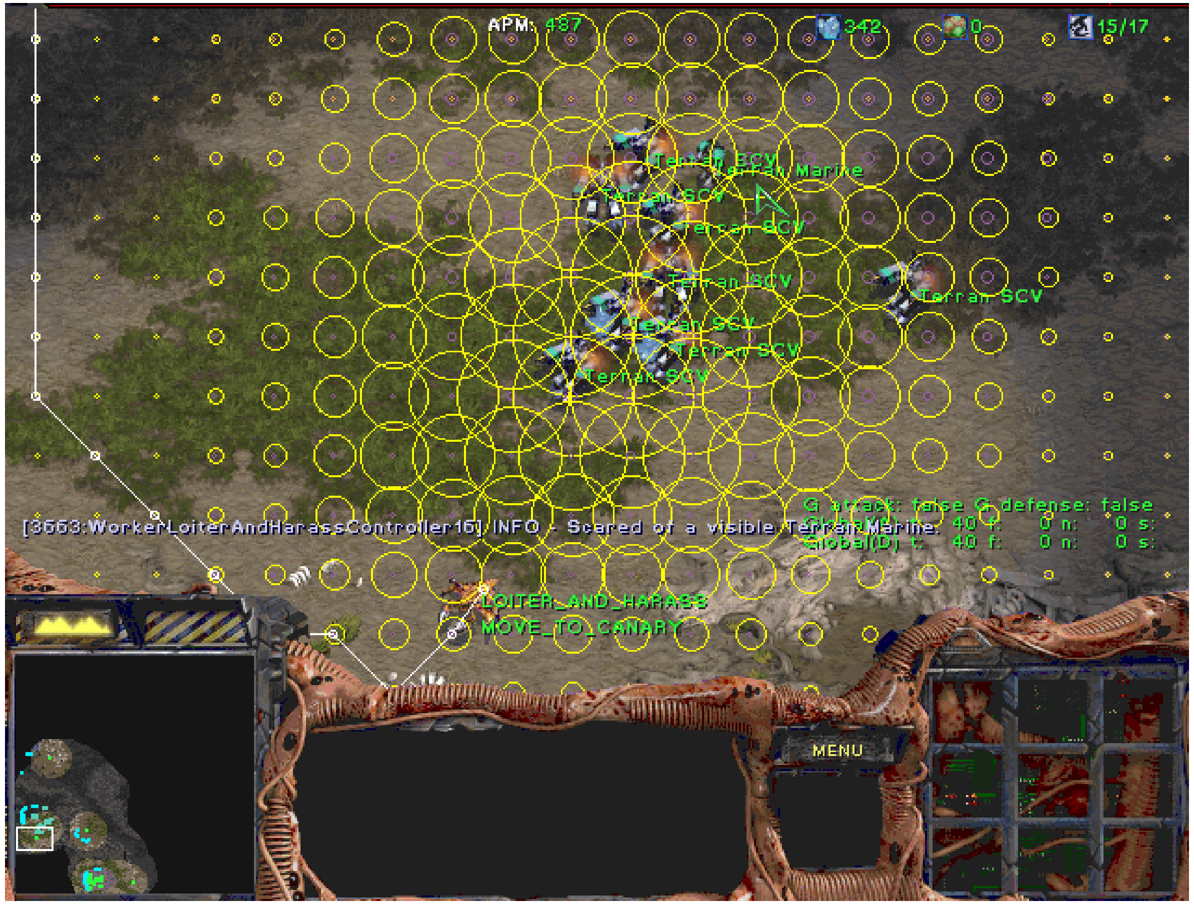
Összehasonlítás

Mohó

Egyenletes költségű keresés

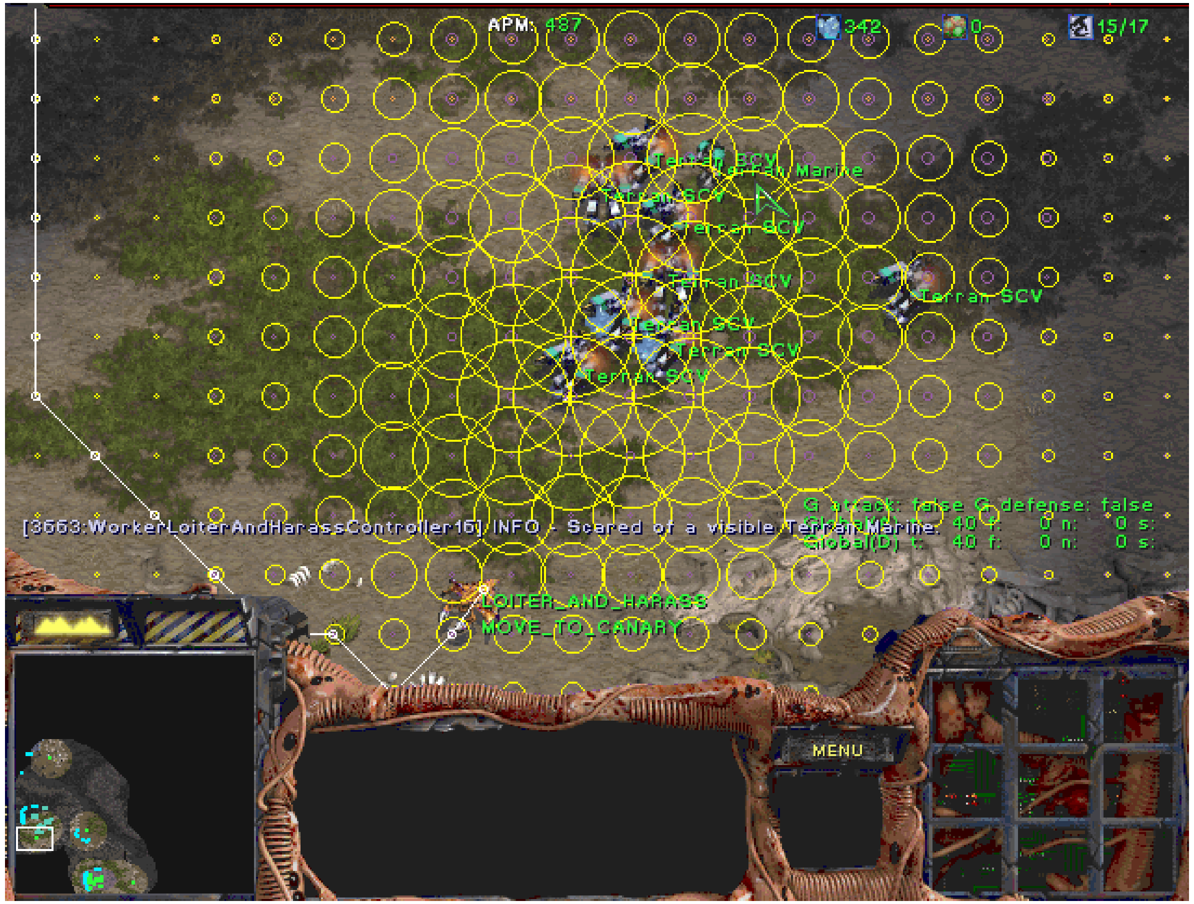
A*

A* alkalmazásai



A* alkalmazásai

- Videójátékok
- Útvonalkeresési problémák
- Erőforrástervezési problémák
- Robot mozgás tervezése
- Nyelvi elemzés
- Gépi fordítás
- Beszédfelismerés

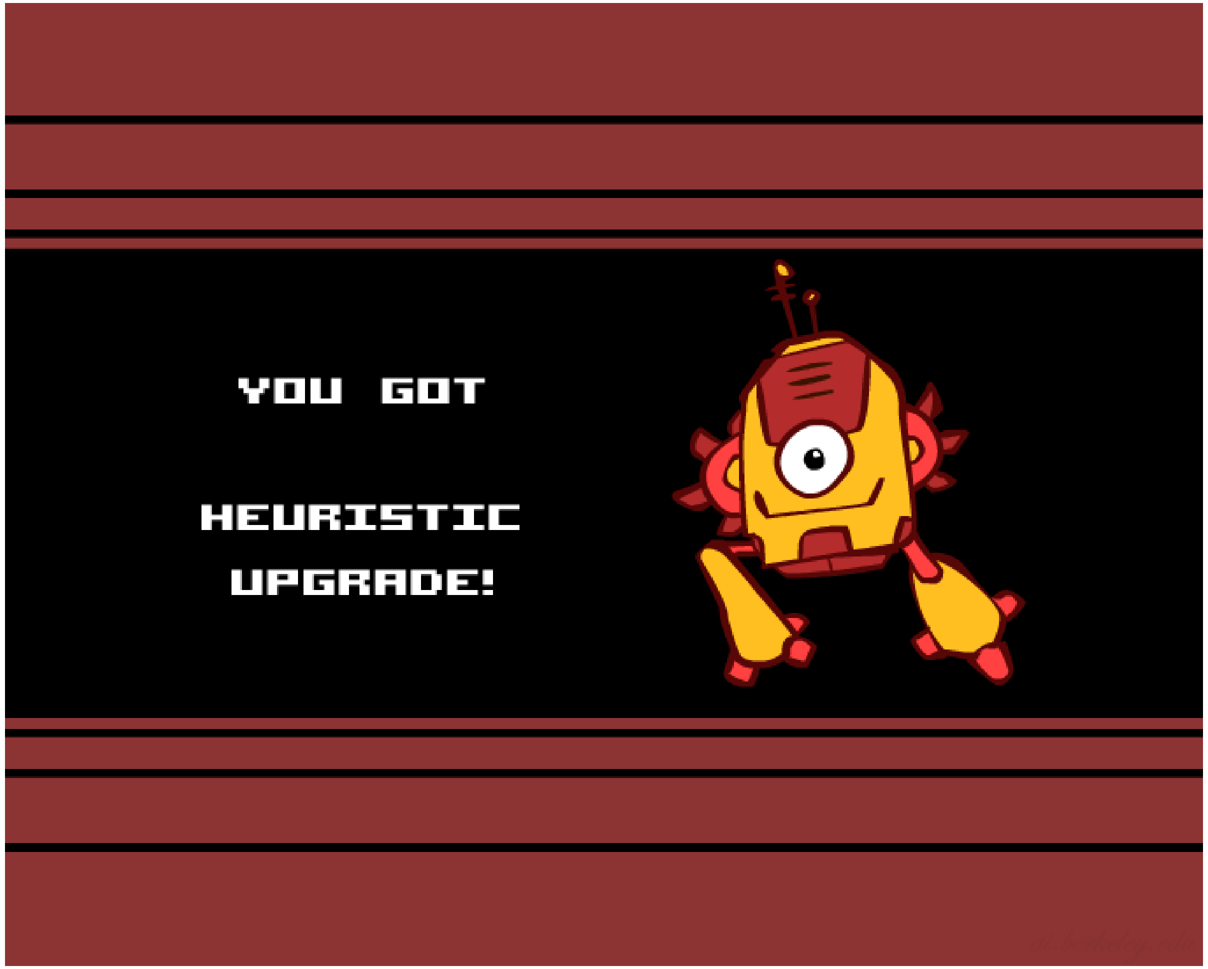


Mesterséges intelligencia

Problémamegoldás kereséssel
Informált keresés

Heurisztikák

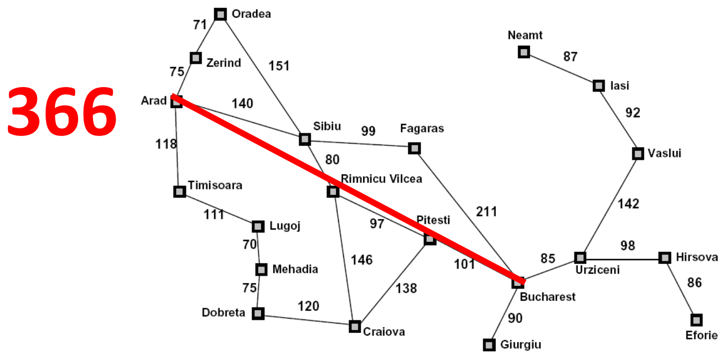
Heurisztikák kialakítása

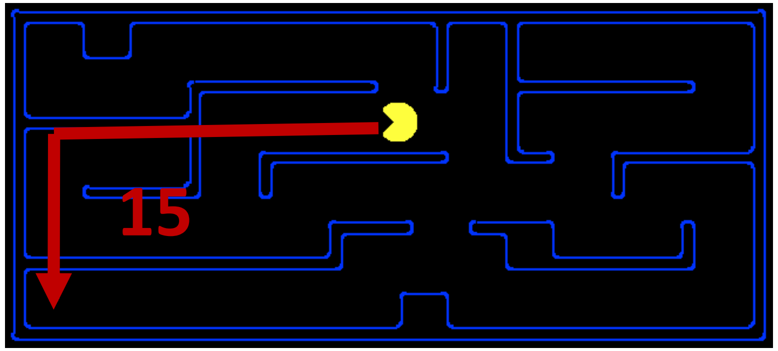


Elfogadható heurisztikák kialakítása

- Nehéz problémák optimális megoldásának a legnehezebb lépése a megfelelő elfogadható heurisztika kialakítása

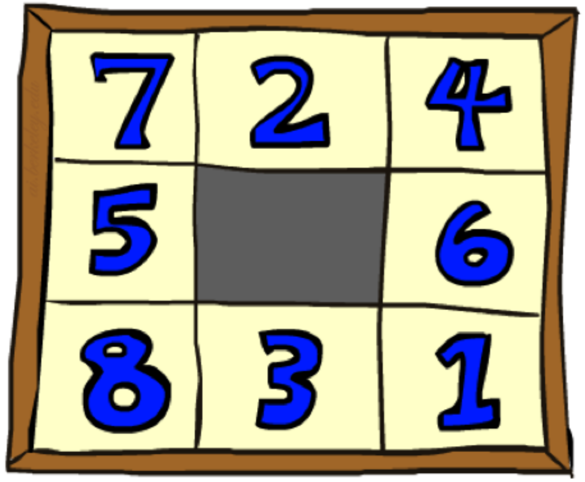
- Az elfogadható heurisztikák gyakran egy relaxált probléma megoldásai, ahol további cselekvések lehetségesek.



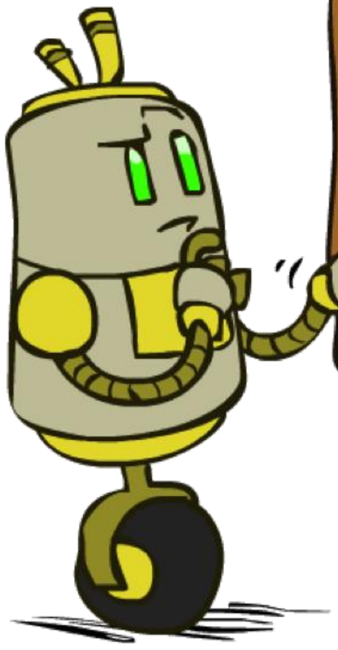


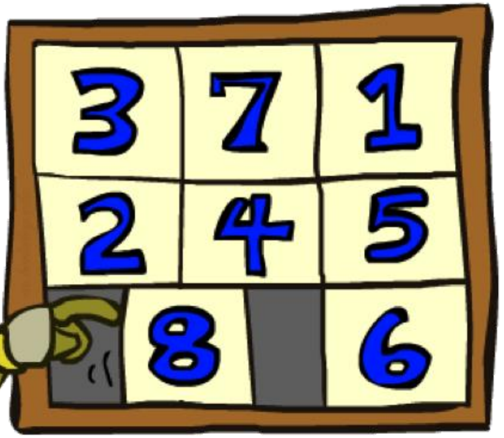
- Nem elfogadható heurisztikák is hasznosak lehetnek esetenként

Példa: 8-as kirakójáték

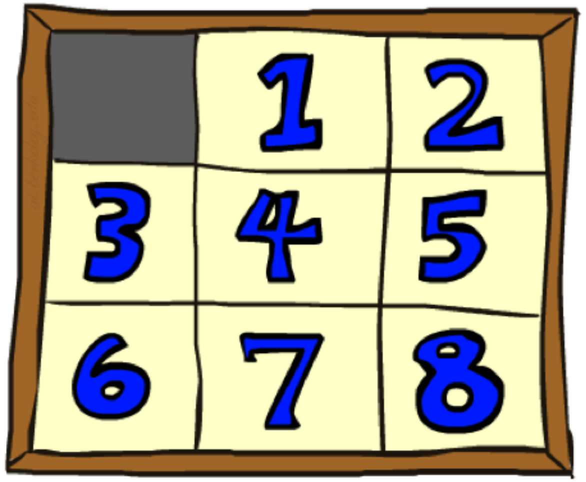


Kiindulási állapot





Cselekvések

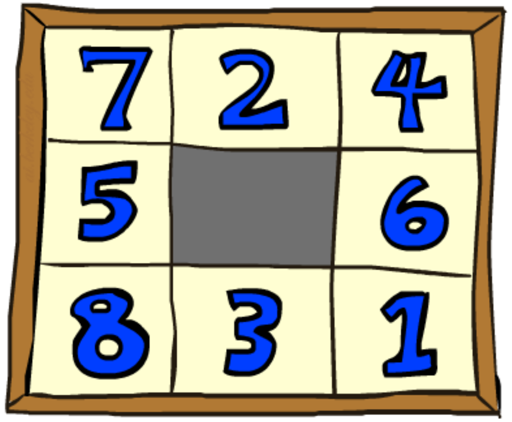


Célállapot

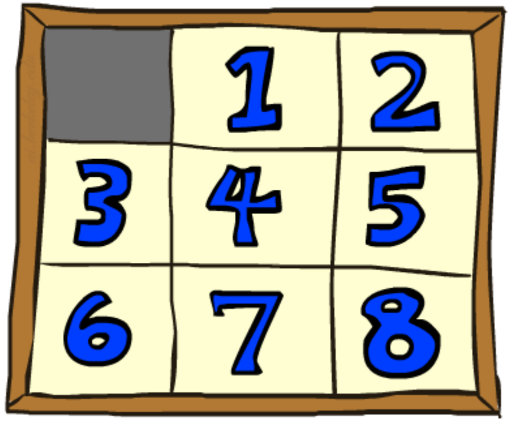
- Melyek a lehetséges állapotok?
- Hány állapot van?
- Melyek a lehetséges cselekvések?
- Hány követőállapot létezik a kiindulási állapotból?
- Mi legyen a költség?

8-as kirakójáték - 1

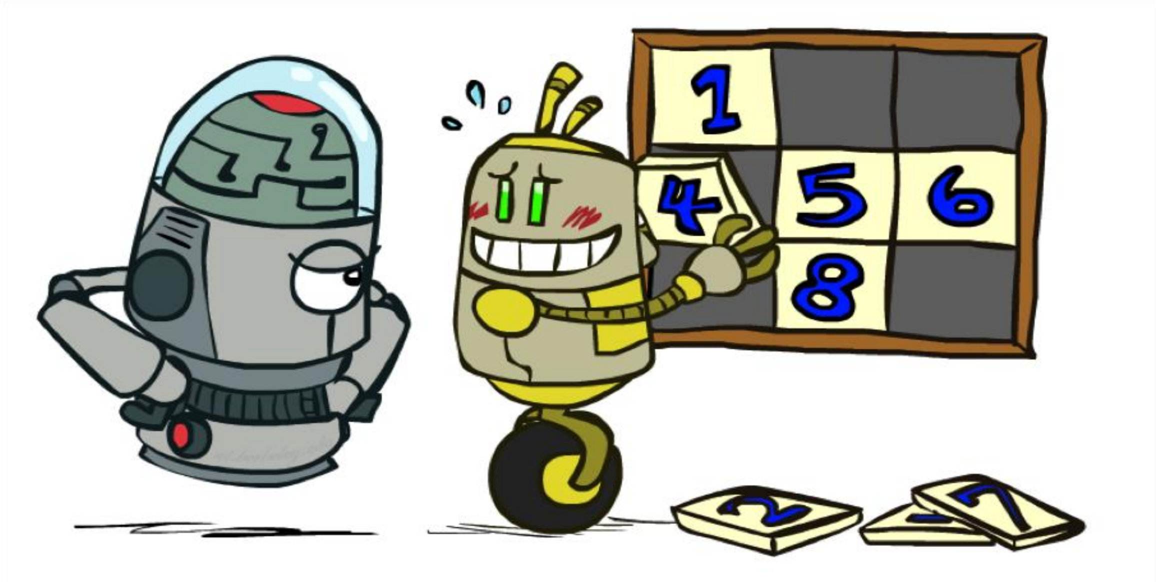
- Heurisztika: a rossz helyen lévő elemek száma
- Ez miért elfogadható?
- h(start) = 8
- Ez egy relaxált probléma alapú heurisztika



Kiindulási állapot



Célállapot



	Átlagos kifejtett csomópontszám, amikor az optimális út tartalmaz…		
	4 lépést	8 lépést	12 lépést
EKK	112	6,300	3.6×10^6
Elemszám	13	39	227

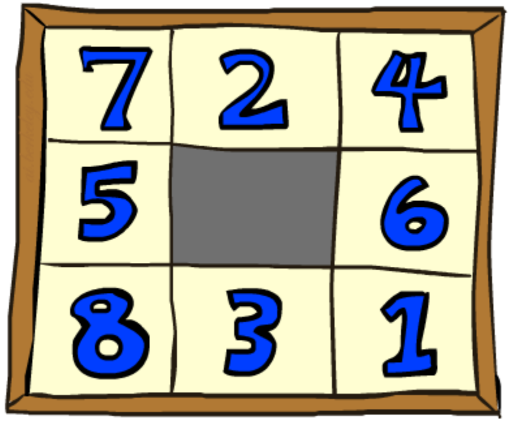
A. Moore

8-as kirakójáték - 2

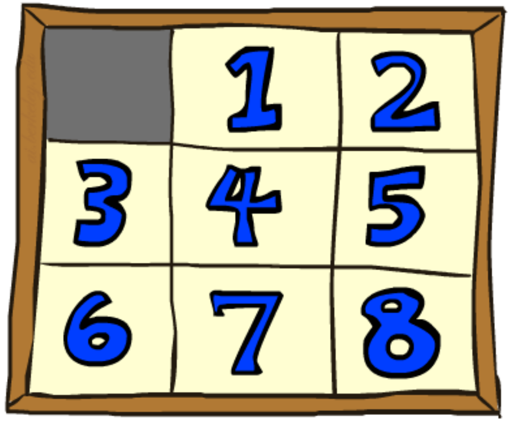
- Mi lenne, ha lenne egy egyszerűbb 8-as kirakó, ahol minden elem minden irányban elmozdulhatna függetlenül a többi elemtől?
- *Manhattan* távolság

- Miért elfogadható?

- h(start) = 3 + 1 + 2 + … = 18



Kiindulási állapot

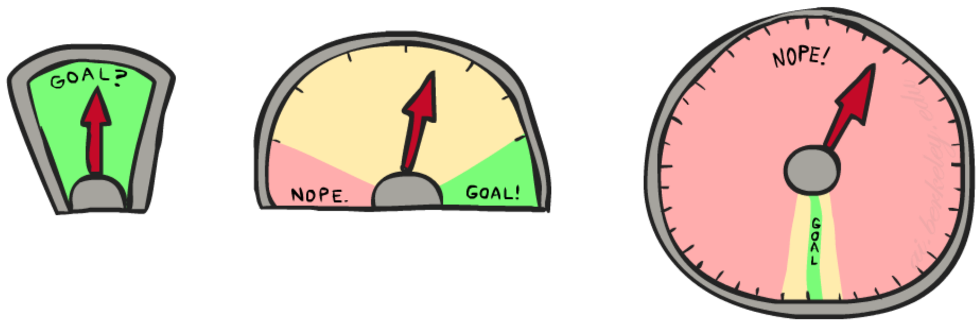


Célállapot

	Átlagos kifejtett csomópontszám, amikor az optimális út tartalmaz…		
	4 lépést	8 lépést	12 lépést
Elemszám	13	39	227
MANHATTAN	12	25	73

8-as kirakójáték - 3

- Mi lenne akkor, ha a tényleges költséget használnák heurisztikának?
  - Elfogadható lenne?
  - Csökkentené a kifejtett csomópontok számát?
  - Mi a probléma vele?



- A*: egyensúlyt teremt a becslés minősége és a csomópontban elvégzendő számítások mennyisége között
  - Ahogy egy heurisztika megközelíti a valós költséget, kevesebb csomópontot kell kifejteni, ugyanakkor a csomópontonként végzendő számítások nagysága nagyobb lesz.

Triviális heurisztikák, dominancia

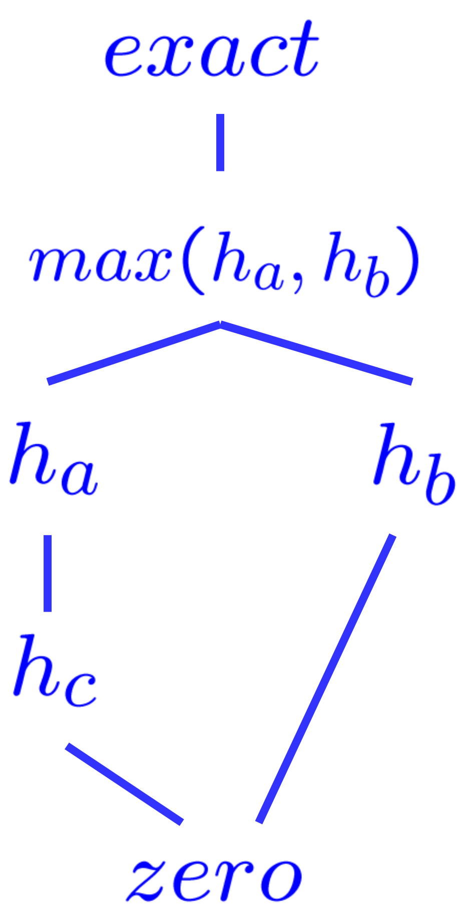
- Dominancia: $h_a \geq h_c$ ha

$$\forall n : h_a(n) \geq h_c(n)$$

- A heurisztikák egy félhálót (semi-lattice) alkotnak (speciális részben rendezett halmazt):
  - A maximuma az elfogadható heurisztikáknak szintén elfogadható

$$h(n) = max(h_a(n), h_b(n))$$

- Triviális heurisztikák
  - A félháló „alja” a zeró heurisztika
  - A teteje pedig az egzakt heurisztika



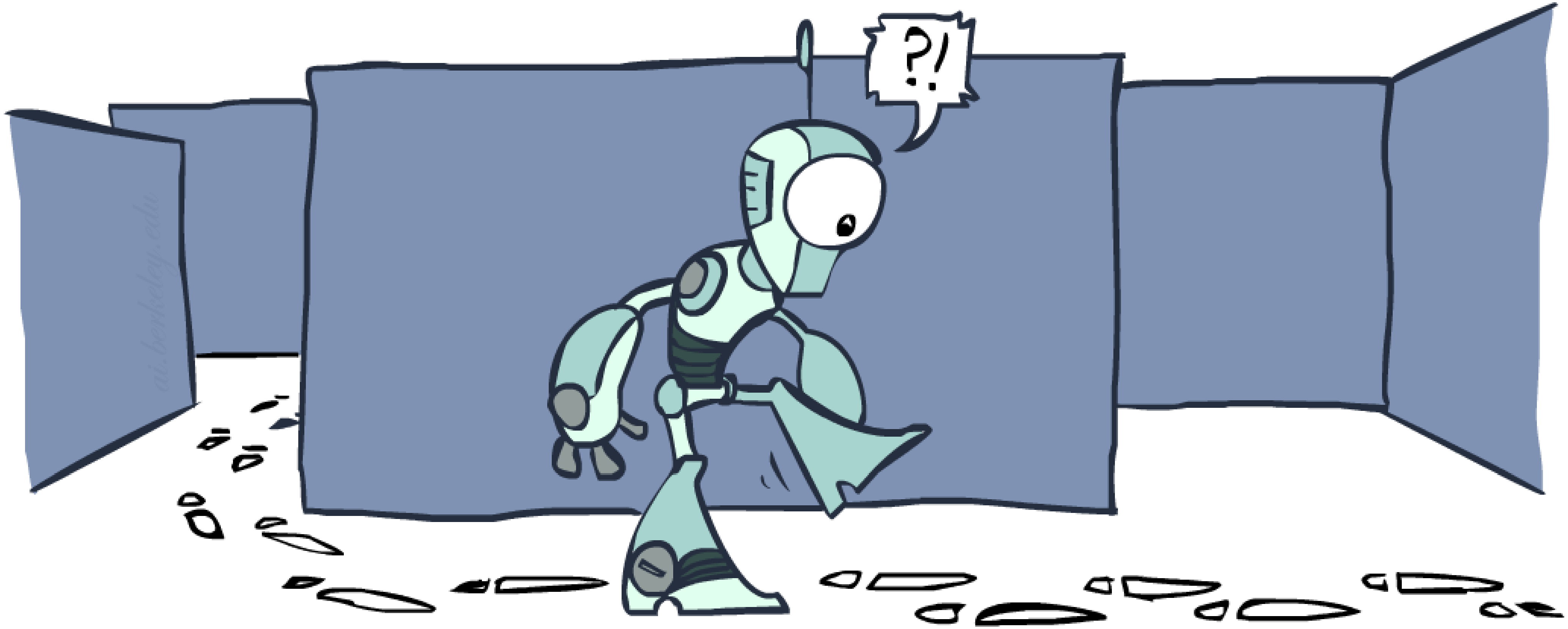
Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Méréstechnika és Információs rendszerek Tanszék

Mesterséges intelligencia

Problémamegoldás kereséssel
Informált keresés

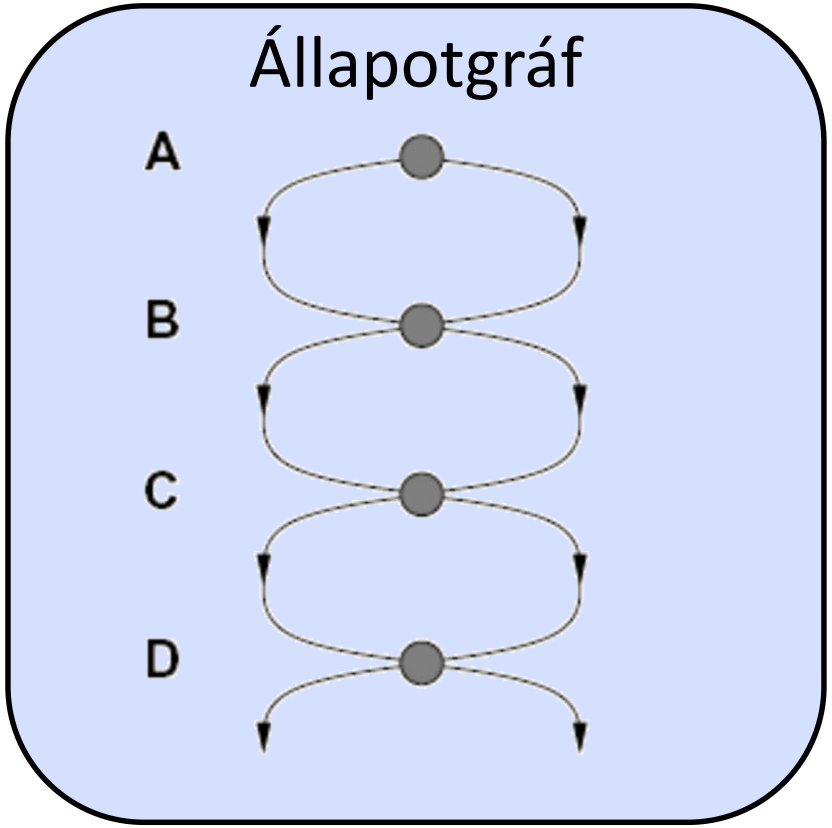
Gráfalapú keresés

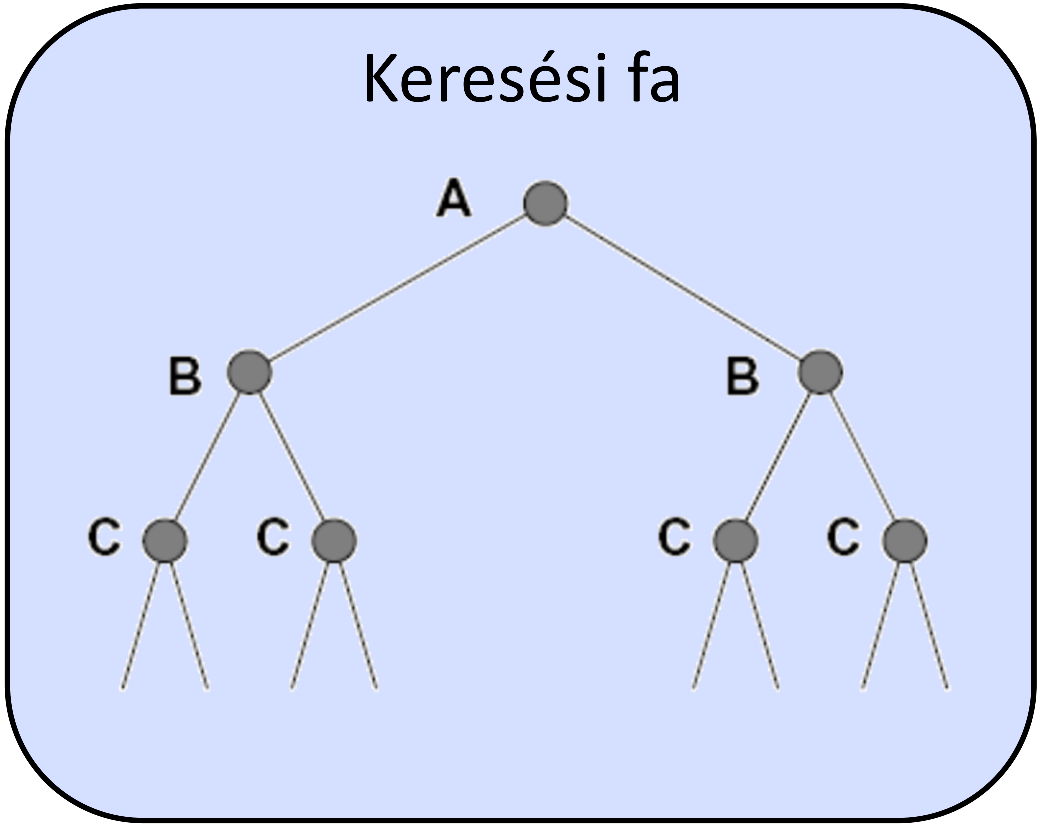
Gráfalapú keresés



A fa alapú keresés többlet munkával jár

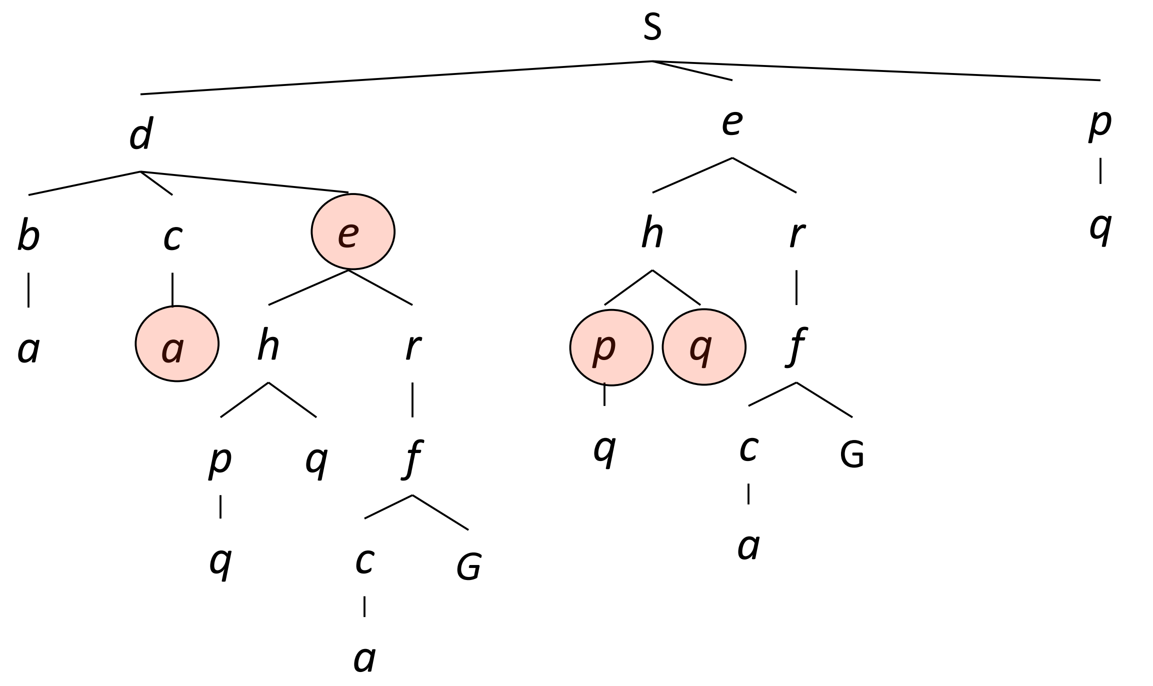
- Ismétlődő állapotok felismerésének a hiánya exponenciálisan több munkát eredményezhet.





Gráf alapú keresés

- Például az alábbi esetben szélességi keresésnél nem kellene foglalkoznunk a bekarikázott csomópontok kifejtésével

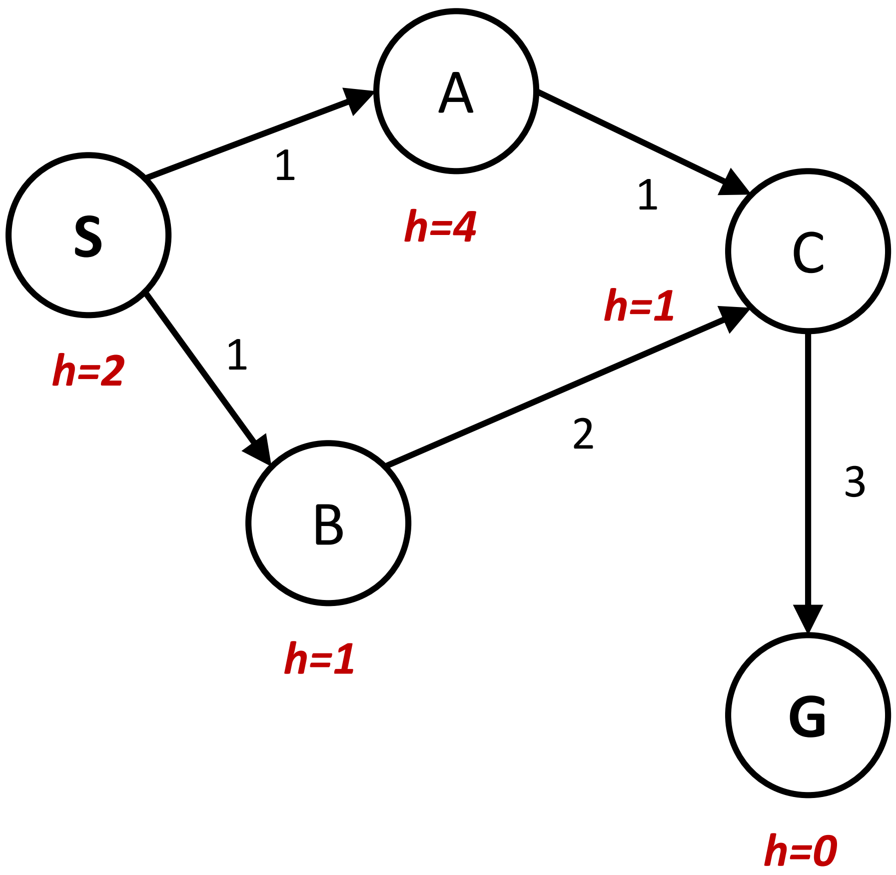


Gráf alapú keresés

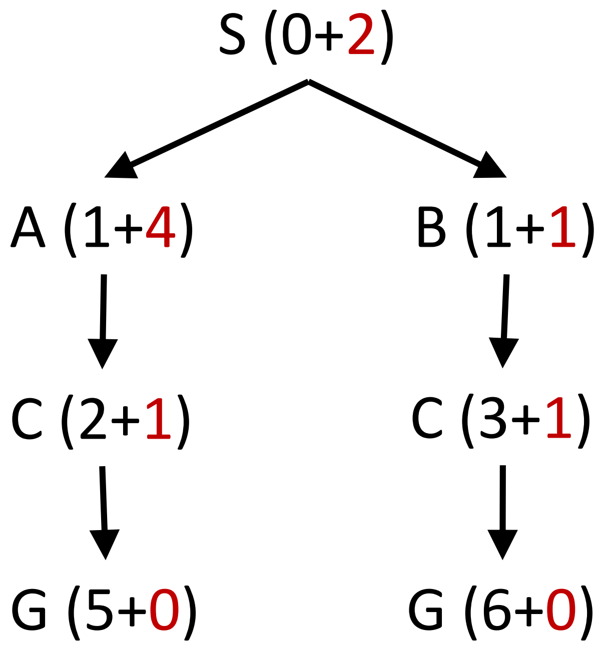
- Alapgondolat: soha ne fejtsük ki ugyanazt az állapotot kétszer
- Megvalósítás:
  - Fa keresés + kifejtett csomópontok halmaza (“zárt halmaz”)
  - Keresési fa kifejtése csomópontonként, de…
  - Kifejtés előtt ellenőrizzük, hogy nem volt-e korábban kifejtve
    - Ha nem új, kihagyjuk,
    - Ha új, hozzáadjuk a zárt halmazhoz.
- Lényeges: a zárt halmazt valóban halmazként tároljuk, ne listaként.
- A gráf alapú keresés befolyásolja-e a teljességet?
- Mi a helyzet az optimalitással?

A* gráf alapú keresés - probléma

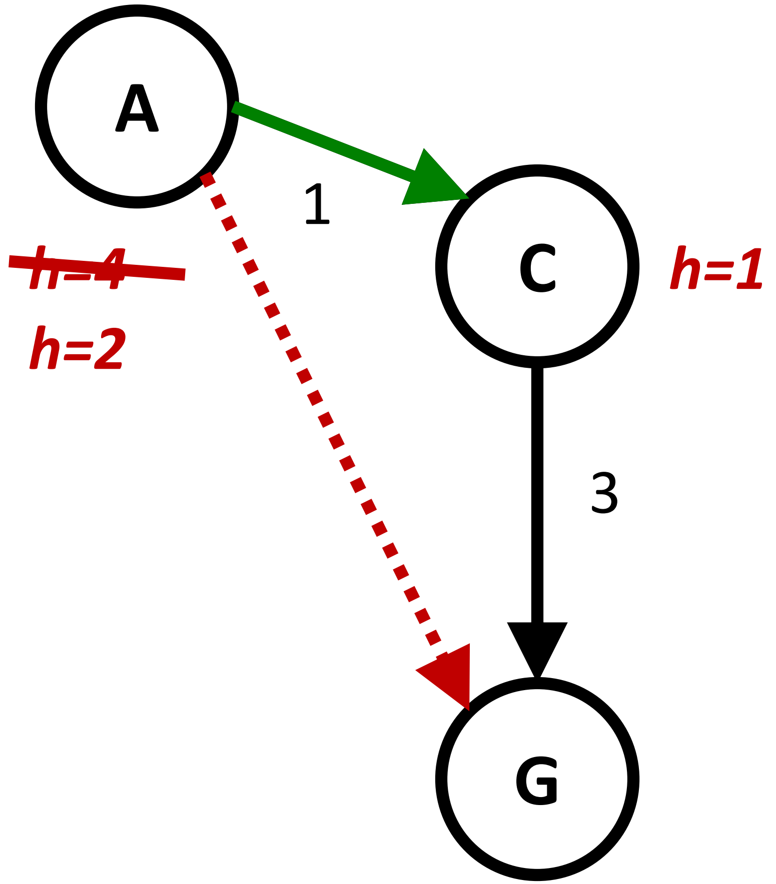
Állapotgráf



Keresési fa



Heurisztikák konzisztenciája



- Központi gondolat:

  becsült heurisztika költsége ≤ valós költség

  - Elfogadhatóság: heurisztika költsége ≤ valós költség a célig

    h(A) ≤ valós költség A-tól G-ig

  - Konzisztencia: heurisztika “élköltsége” ≤ valós élköltség

    h(A) – h(C) ≤ valós költség(A-> C)

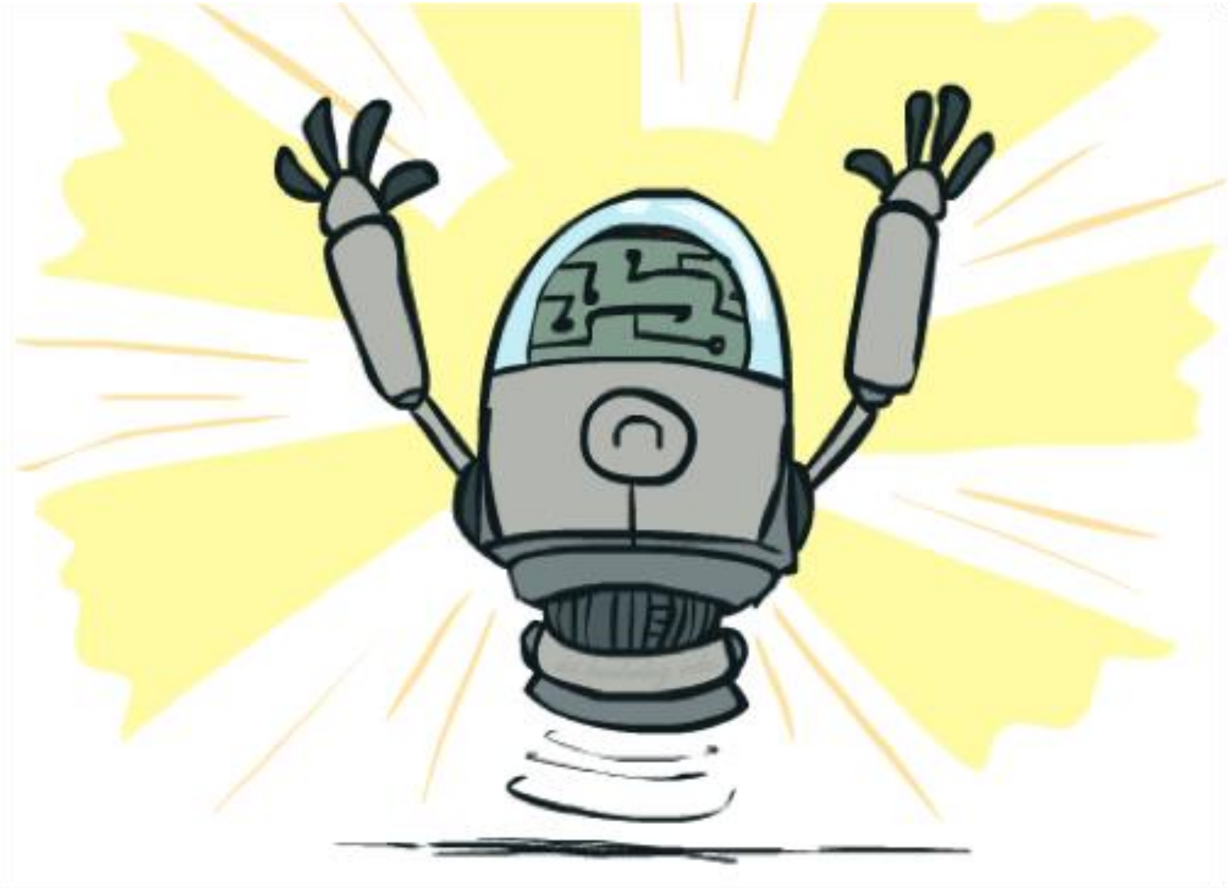
- Konzisztencia következménye:

  - Az f értéke nem csökken az útvonal mentén

    h(A) ≤ valós költség(A-> C) + h(C)

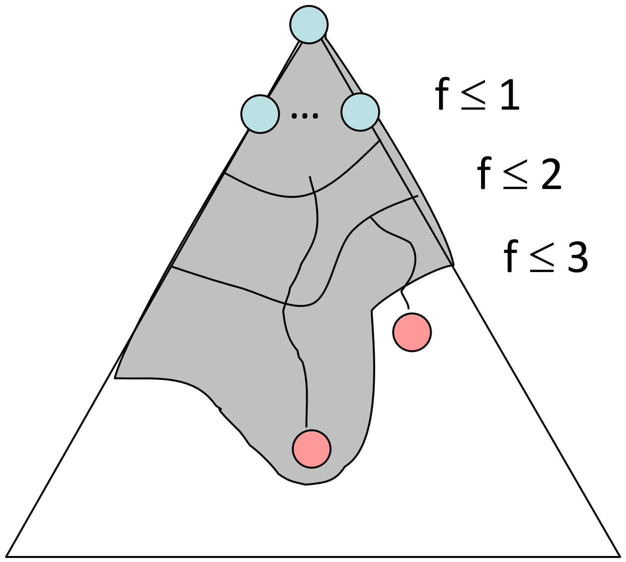
  - A* gráf alapú keresés optimális

A* gráf keresés optimalitása



A* gráf keresés optimalitása

- Vázlat: tekintsük át mit tesz az A* keresés konzisztens heurisztikával:
  - 1. Tény: Fa keresés esetén az A* a csomópontokat növekvő f érték szerint fejti ki. (f-határvonal)
  - 2. tény: Minden *s* állapotra, azokat a csomópontokat, amiken keresztül *s* optimálisan elérhető, hamarabb fejti ki, mint azokat, melyeken keresztül *s* szuboptimálisan elérhető.
  - Eredmény: A* gráf keresés optimális



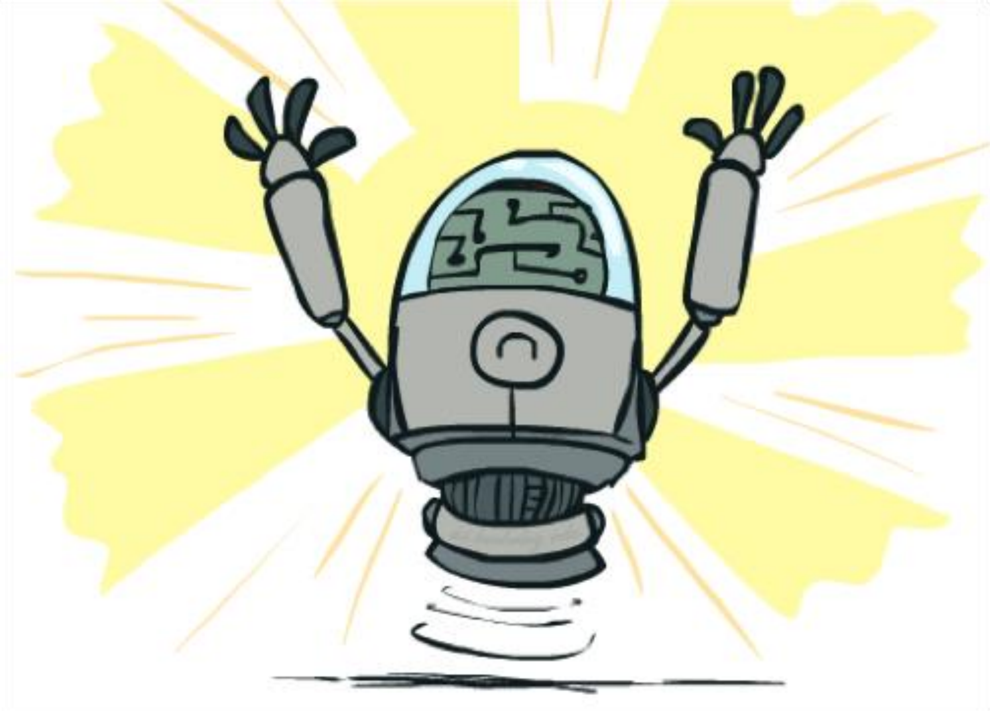
Optimalitás

- Fa alapú keresés:
  - A* optimális, ha a heurisztika elfogadható
  - Az egyenletes költségű keresés ennek egy speciális esete (h = 0)

- Gráf alapú keresés:
  - A* optimális, ha a heurisztika konzisztens
  - Az egyenletes költségű keresés optimális (h = 0 konzisztens)

- Konzisztenciából következik az elfogadhatóság

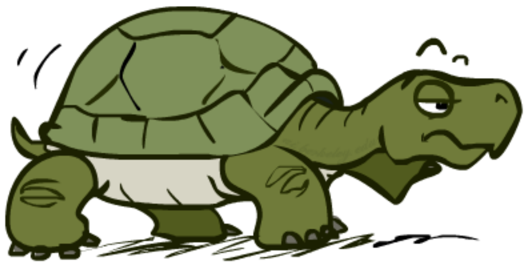
- Általában a legtöbb természetesen adódó, elfogadható heurisztika konzisztens, főleg ha relaxált problémából származnak



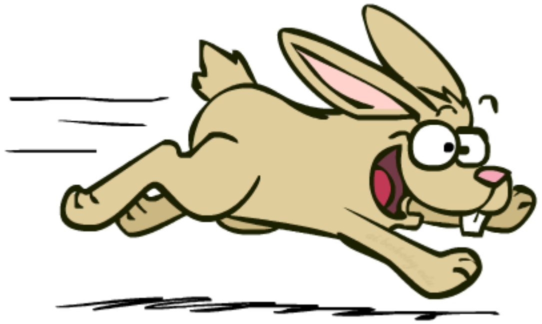
A*: Összefoglalás

A*: Összefoglalás

- A* figyelembe veszi az adott pontig megtett út költségét (backward cost), és az adott pontból a célállapotig tartó út becsült költségét (forward cost)
- A* optimális elfogadható / konzisztens heurisztika alkalmazása esetén
- A heurisztika megtervezése a lényeges pont, gyakran egy relaxált probléma alapján alakítjuk ki



Egyenletes költségű keresés



Mohó keresés

A* keresés

Fa keresés pszeudókódja

```
function FA-KERESÉS (probléma, perem) return egy megoldás vagy kudarc
        perem ← BEILLESZT(ÚJ-CSOMÓPONT(KEZDŐ-ÁLLAPOT[probléma]), perem)
        loop do
                if perem üres then return kudarc
                csomópont ← ELSŐ-ELEM-KIVÁLASZT(perem)
                if CÉLÁLLAPOT-TESZT(probléma, ÁLLAPOT[csomópont]) then return csomópont
                for gyermek-csomópont in KIFEJTÉS(ÁLLAPOT[csomópont], probléma) do
                perem ← BEILLESZT(gyermek-csomópont, perem)
                end
        end
```

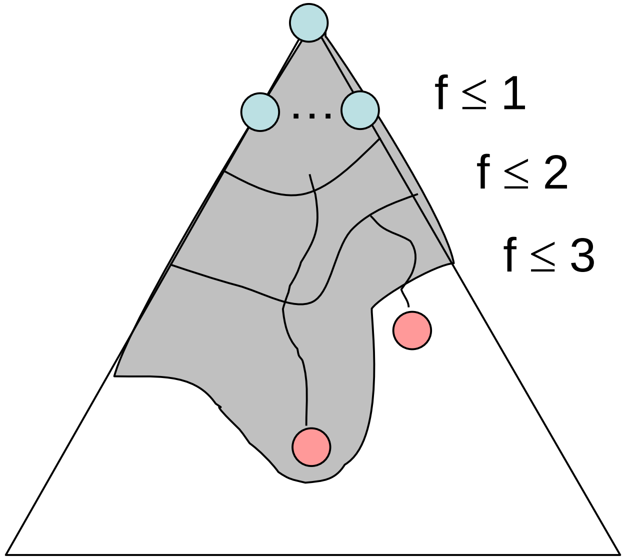
Gráf keresés pszeudókódja

```
function GRÁF-KERESÉS (probléma, perem) return egy megoldás vagy kudarc
    zárt-halmaz ←üres
    perem ← BEILLESZT(ÚJ-CSOMÓPONT(KEZDŐ-ÁLLAPOT[probléma]), perem)
    loop do
        if perem üres then return kudarc
        csomópont ← ELSŐ-ELEM-KIVÁLASZT(perem)
        if CÉLÁLLAPOT-TESZT(probléma, ÁLLAPOT[csomópont]) then return csomópont
        if ÁLLAPOT[csomópont] not in zárt-halmaz then
            zárt-halmaz ← HOZZÁAD(ÁLLAPOT[csomópont], zárt-halmaz)
            for gyermek-csomópont in KIFEJTÉS(ÁLLAPOT[csomópont], probléma) do
                perem ← BEILLESZT(gyermek-csomópont, perem)
            end
    end
```

A* Gráf keresés optimalitása

- Mit tesz az A*:
  - Csomópontokat fejt ki növekvő *f* érték szerint (f-határvonal)
    $f(n) = g(n) + h(n)$ = „n”-be jutás költsége + heurisztika
  - Bizonyítás alapgondolata: Az optimális cél(ok) rendelkeznek a legkisebb *f* értékkel tehát ezeket kell előbb kifejtenie.

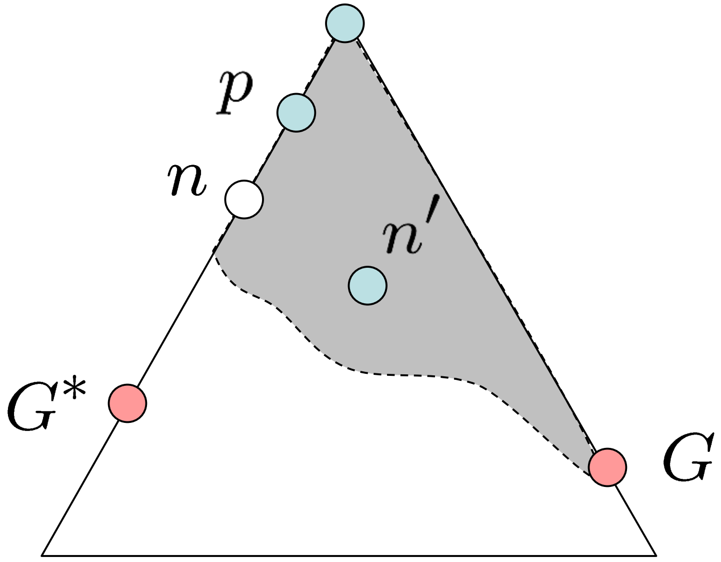
Ez jó lenne, de elég? Élnünk kell-e még további feltételezésekkel?



A* Gráf keresés optimalitása

Bizonyítás:

- Új probléma: a G*-hoz tartó útvonalon egy ***n*** csomópont nincs benne kifejtendő csomópontok sorában, mert egy rosszabb ***n'*** került ki előbb és azt fejtette ki az algoritmus
- Vegyük a legmagasabban lévő ***n*** *-et* a keresési fából
- Legyen ***p*** az ***n*** őse,ami benne volt a sorban, amikor ***n'*** kikerült onnan.
- ***f(p) < f(n)*** a konzisztencia miatt
- ***f(n) < f(n')*** mivel ***n'*** szuboptimális
- ***p*** csomópontot előbb kellett volna kifejtenie, mint ***n'***-et
- Ellentmondásra jutottunk!



Budapesti Műszaki és Gazdaságtudományi Egyetem
Méréstechnika és Információs rendszerek Tanszék

Mesterséges intelligencia – VIMIAC10

Kényszerkielégítési problémák

Előadó: Dr. Hullám Gábor

Előadás anyaga: Dr. Pataki Béla
Dr. Dobrowiecki Tadeusz

MŰEGYETEM 1782

Témakörök

Ajánló rendszerek

Kényszerkielégítési (korlátkielégítési) problémák Constraint Satisfaction Problems (CSP):

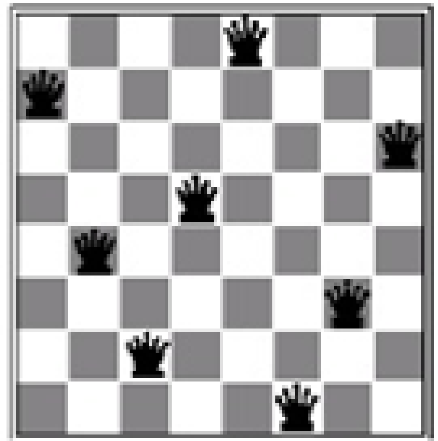
Állapot: Az állapot a leíró változók és a hozzájuk rendelt értékek által definiált.

- Az x_k változók egy-egy D_k érték-tartományból (halmazból) veszik fel értékeiket .

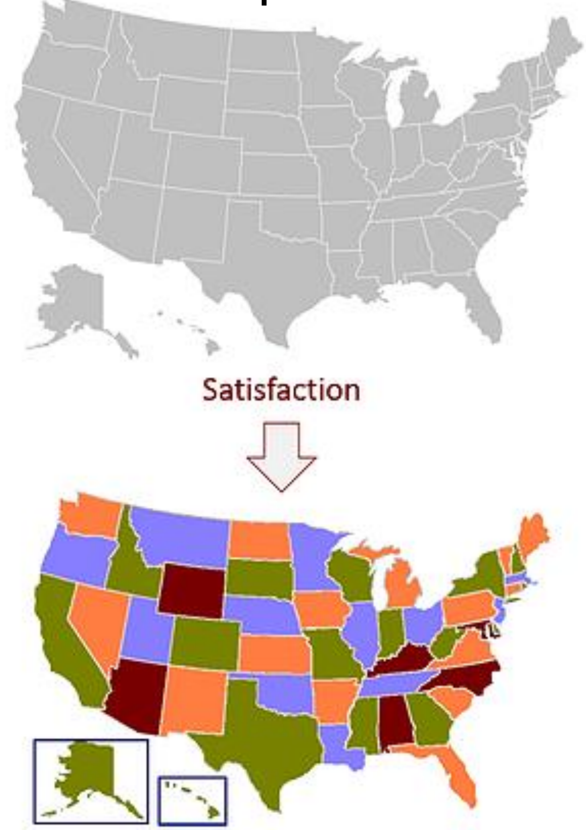
Célállapotteszt:

1. Az összes állapotot leíró változóhoz rendeltünk számára megengedett értéket
2. Az adott korlátok teljes halmazát kielégítettük

		8		5			1	2
			8		9			3
	6		2		4		5	
6		9		2	3	8		
8		5				3		6
		4	9	8		5		1
	3		1		2		8	
2			4		8			
4	8			9		2		



Térképszínezés




```
  T W O
+ T W O
-------
F O U R
```

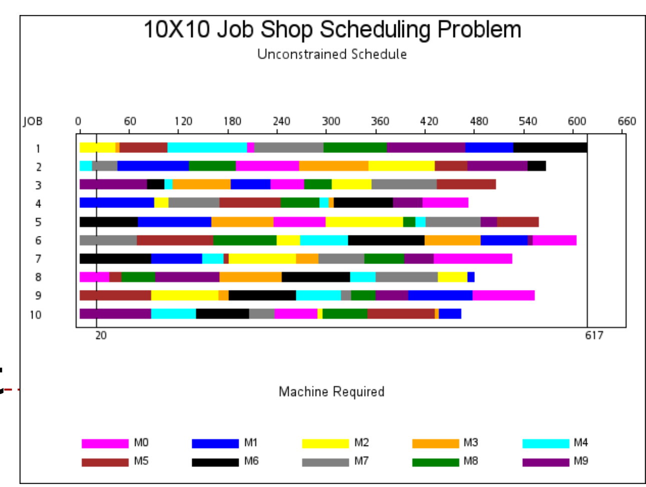
Órarend készítés:

Szerda 10-12, HG,MI-BSc5szem, IB.028

...

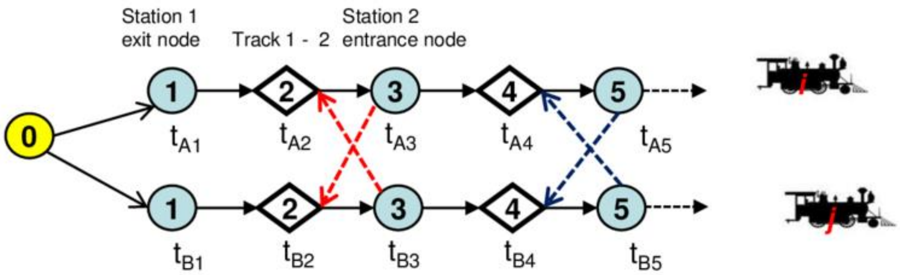
1. egyszerre egy teremben csak egy évfolyam
2. egyszerre egy teremben csak egy oktató,
3. senki se lehet egyszerre két helyen
4. egy évfolyam egyszerre csak egy helyen lehet

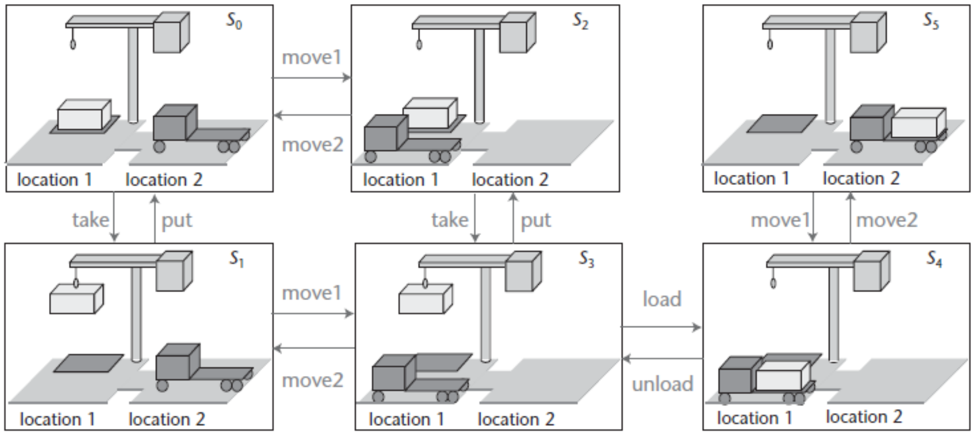
- stb.

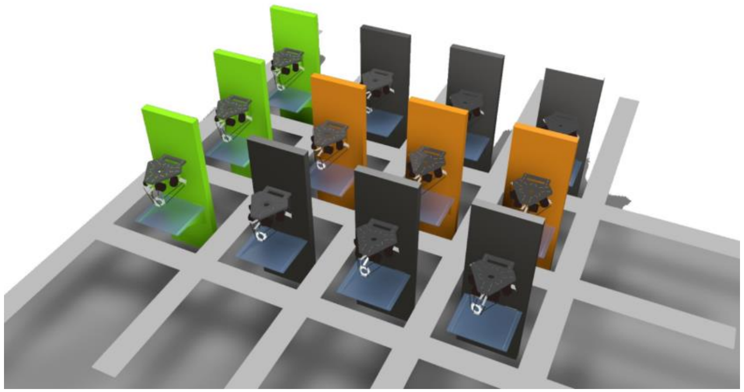


Gyakorlati kényszerkielégítési problémák

- **Hozzárendelési problémák**, pl. ki, mit, hol tanítson
- **Menetrendi problémák**, pl. vasút
- **Szállításütemezés**
- **Gyári ütemezés**







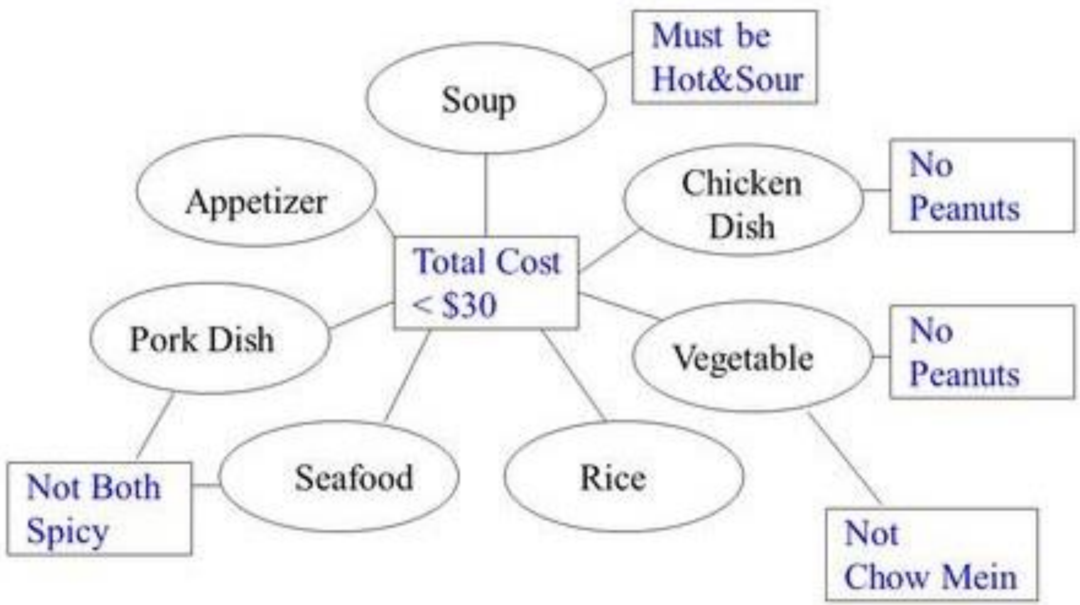
Gyakorlati kényszerkielégítési problémák

Raktár-tervezési probléma

Egy szupermarket-lánc raktárakat szeretne telepíteni a bolti hálózat kiszolgálására. Mit tudunk?

- L_1, ..., L_n lokáció, ahol raktár megépíthető.
- Csak k db. raktárra van szükség ($k < n$).
- Minden lokációra jellemző CP_i kapacitás: hány boltot képes kiszolgálni
- Minden bolthoz rendelni kell egy raktárt.
- S_j bolt ellátása L_i lokációból $P_{i,j}$-be kerül.
- Az összköltséget TC konstans alatt kell tartani.

Menüösszeállítás az étlap alapján



CSP problémák típusai

Változók alapján:

Diszkrét változók

véges értéktartományok:

- pl. Boole-típ. CSPs, Boole-féle kielégítési vizsgálatok (NP-t)

végtelen értéktartományok:

- egész számok, füzérek, stb.
- pl. job scheduling, változók a munkaszakaszok kezdete/vége

Folytonos változók

- megfigyeléseket határoló időpontok,
- fizikai állapotváltozók

CSP problémák típusai

***<u>Kényszerek, korlátok alapján</u>*:**

Unáris korlát: egyetlen egy változóra vonatkozik, pl. SA ≠ green

Bináris korlát: két változó viszonyára vonatkozik, pl. SA ≠ WA

Magasabb-rendű korlát: 3 vagy több változó viszonyára vonatkozik, (pl. oszloponkénti változó korlátok kriptoaritmetikai feladványokban)

Preferecia-kényszer:

Egy érték jobb(an preferált), mint egy másik, pl. a vörös jobb, mint a zöld

Példa: Térképszínezés

Változók: *WA, NT, Q, NSW, V, SA, T*

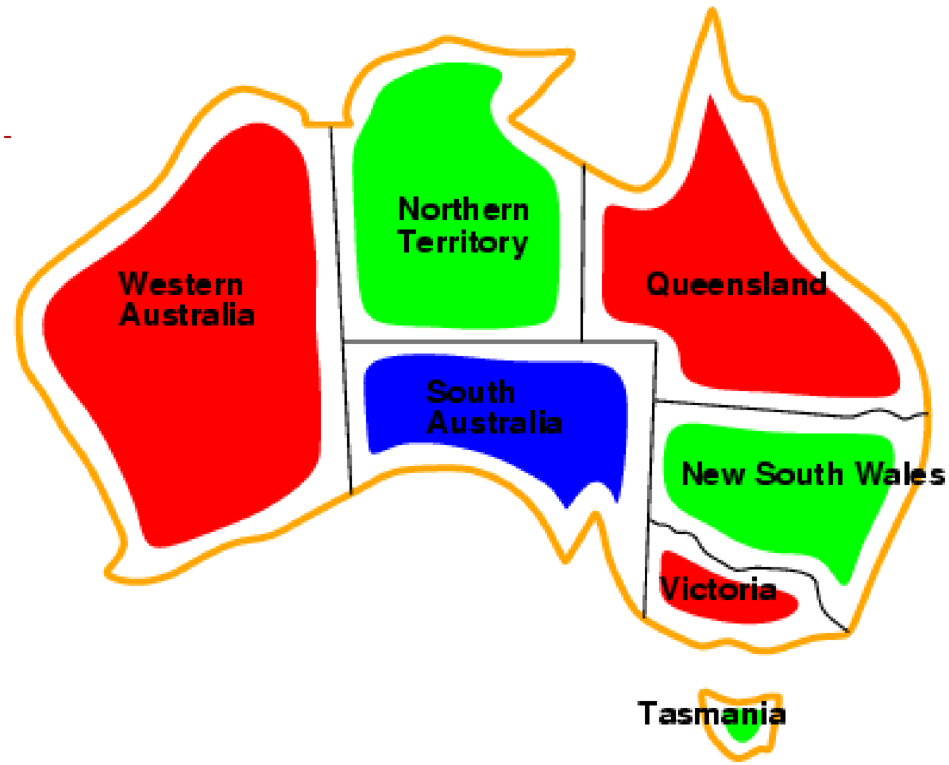
Értéktartományok D_i = {red, green, blue}

Korlátok: szomszédos területek színe legyen eltérő

pl. WA ≠ NT, ill. más megfogalmazásban: (WA,NT) értékét csakis a {(red, green), (red, blue), (green, red), (green, blue), (blue, red), (blue, green)} halmazból vehet fel.

Lehetséges-e?

pl. Térképszínezés

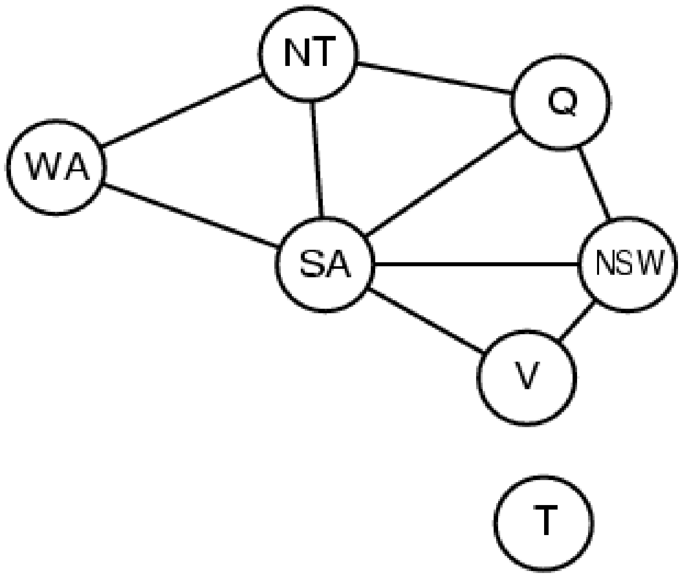


Megoldás: **teljes** és **konzisztens** változó-érték hozzárendelés

pl. WA = red, NT = green, Q = red, NSW = green, V = red, SA = blue, T = green

(T lehet akármi, mert nem határos senkivel - különben is több megoldás van)

Korlátok gráfja



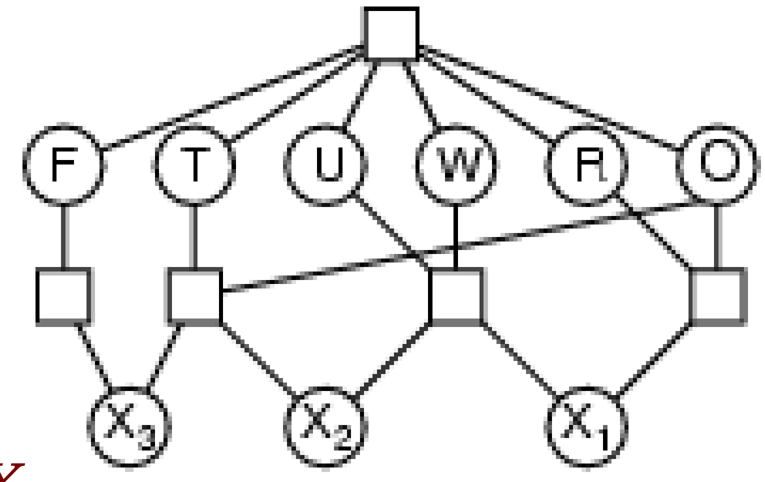
korlátgráf: csomópontjai a változók és élei a korlátok

Itt csak **bináris CSP**-k: egy-egy korlát 2 változót köt össze

…

Másik példa: kriptoaritmetika

```
   T W O
 + T W O
 -------
 F O U R
```



Változók: $F\ T\ U\ W\ R\ O\ X_1\ X_2\ X_3$

Értéktartományok: $\{0,1,2,3,4,5,6,7,8,9\}$, $\{0,1\}$

Korlátok:

1. Mind-eltérő(F, T, U, W, R, O)
2. $O + O = R + 10 \cdot X_1$
3. $X_1 + W + W = U + 10 \cdot X_2$
4. $X_2 + T + T = O + 10 \cdot X_3$
5. $X_3 = F$, $T \neq 0$, $F \neq 0$

A probléma megfogalmazása

Kezdeti állapot: összes változó-hozzárendelés üres { }

Operátor: értéket hozzárendelni egy még nem lekötött változóhoz úgy, hogy az eddigi hozzárendelésekkel ne ütközzön (egyik kényszer se sérüljön)

→ kudarc, ha nincs megengedett hozzárendelés

Célállapotteszt: ha az aktuális hozzárendelés teljes és mindegyik kényszer teljesül

A probléma megfogalmazása

Keresési fa

n db változó esetén minden megoldás *n* mélységben fekszik

→ mi van, ha a **szélességi keresést** használjuk?

Elágazások száma *L* mélységben (*L*=0,1,2,…), ha minden változó *d* számú értéket vehet fel (ez a változó értékkészlete v. más néven doménje)

$b = (n - L)\,d$, (mert *L* változó már értéket kapott)

azaz $\boldsymbol{n!\cdot d^{\,n}}$ levélcsomópont

(miközben $\boldsymbol{d^{\,n}}$ lehetséges hozzárendelés van)

(pl. 8 betűs számjegyaritmetika, n=8, d=10, levelek száma $4\cdot10^{12}$)

Keresés

Változó-hozzárendelés **kommutatív**, azaz például

(WA = red) majd (NT = green)

ugyanaz, mint (NT = green) majd (WA = red)

Egy-egy csomópontban csakis egyetlen egy változó hozzárendelése történhet meg, így:

→ $b = d$ és a fának $d^{\,n}$ levele van

Keresés

- Változó-hozzárendelés **kommutatív**, azaz például
- Egy-egy csomópontban csakis egyetlen egy változó hozzárendelése történhet meg, így:

  → $b = d$ és a fának $d^{\,n}$ levele van

- Alapvető nem informált algoritmus (keresés) CSP problémák megoldására: **visszalépéses keresés**, azaz

  mélységi keresés
  - minden szinten egyetlen egy változó-hozzárendeléssel
  - ha sérül valamelyik kényszer, visszalép
    (egyszer sérült kényszer mélyebben nem jöhet helyre)

Megfogalmazás (modell) hatása a problémamegoldásra

Az n-királynő probléma tanulsága: hasonlítsunk össze három modellt

1. modell, a változók: x_{ij} (*a sakktáblamezők pozíciója*)
értékkészlet: {0, 1} (*van rajta királynő vagy nincs*)
kényszerek (sorok, oszlopok, átlók)

2. modell, a változók: $x_1, \ldots, x_n$ (*egy királynő által elfoglalt mező poziciója*)
értékkészlet: $\{0, 1, 2, \ldots, n^2-1\}$ (*mezőindex*)
kényszerek (sorok, oszlopok, átlók)

3. modell, a változók: $x_1, \ldots, x_n$ (az 1., 2. stb. sorban álló *királynő sorindexe*)
értékkészlet: $\{1, 2, \ldots, n\}$ (*oszlop-index*)
kényszerek (sorok, oszlopok, átlók)

Megfogalmazás (modell) hatása a problémamegoldásra

1. modell, a változók: x_{ij} (*a sakktáblamezők pozíciója*)
értékkészlet: {0, 1} (*van rajta királynő vagy nincs*)

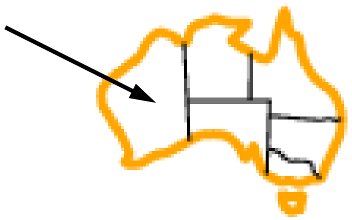
2. modell, a változók: $x_1, \ldots, x_n$ (*egy királynő által elfoglalt mező poziciója*)
értékkészlet: $\{0, 1, 2, \ldots, n^2-1\}$ (*mezőindex*)

3. modell, a változók: $x_1, \ldots, x_n$ (az 1., 2. stb. sorban álló *királynő sorindexe*)
értékkészlet: $\{1, 2, \ldots, n\}$ (*oszlop-index*)

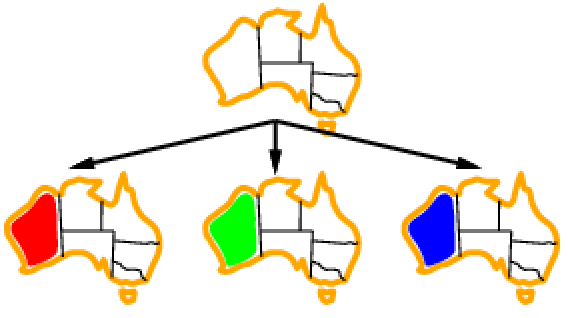
n x *n*-es sakktáblán az *n*-királynő probléma néhány *n*-re

modell	változó	értékkészl.(d)	levelek sz.(d^n)	n = 4	n = 8	n = 20
1.	n^2	2	$(2)^{(n\hat{}2)}$	$6.6 \cdot 10^4$	$1.8 \cdot 10^{19}$	$2.6 \cdot 10^{120}$
2.	n	n^2	$(n^2)^n$	$6.6 \cdot 10^4$	$2.8 \cdot 10^{14}$	$1.1 \cdot 10^{52}$
3.	n	n	n^n	256	$1.6 \cdot 10^7$	$1.0 \cdot 10^{26}$

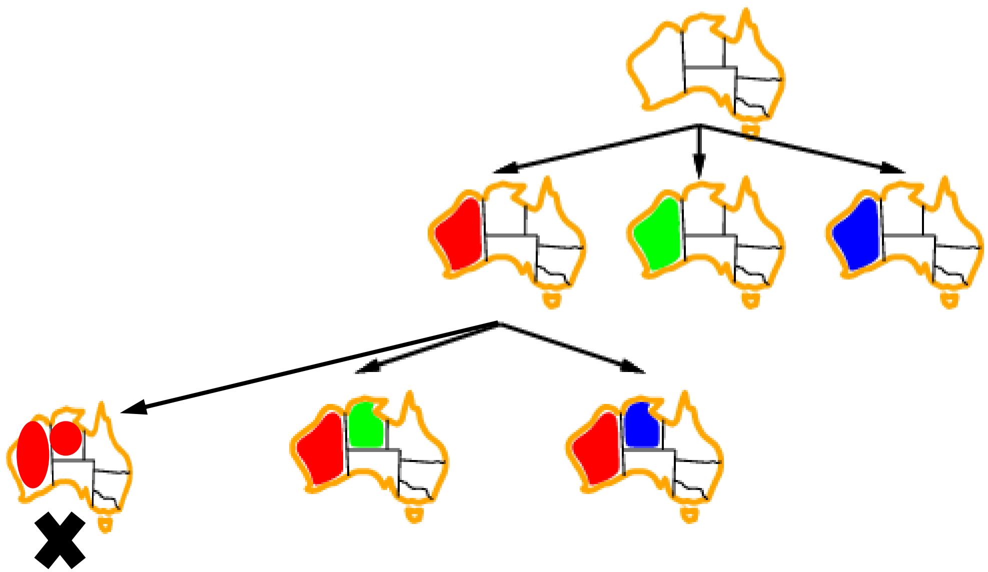
Visszalépéses keresés



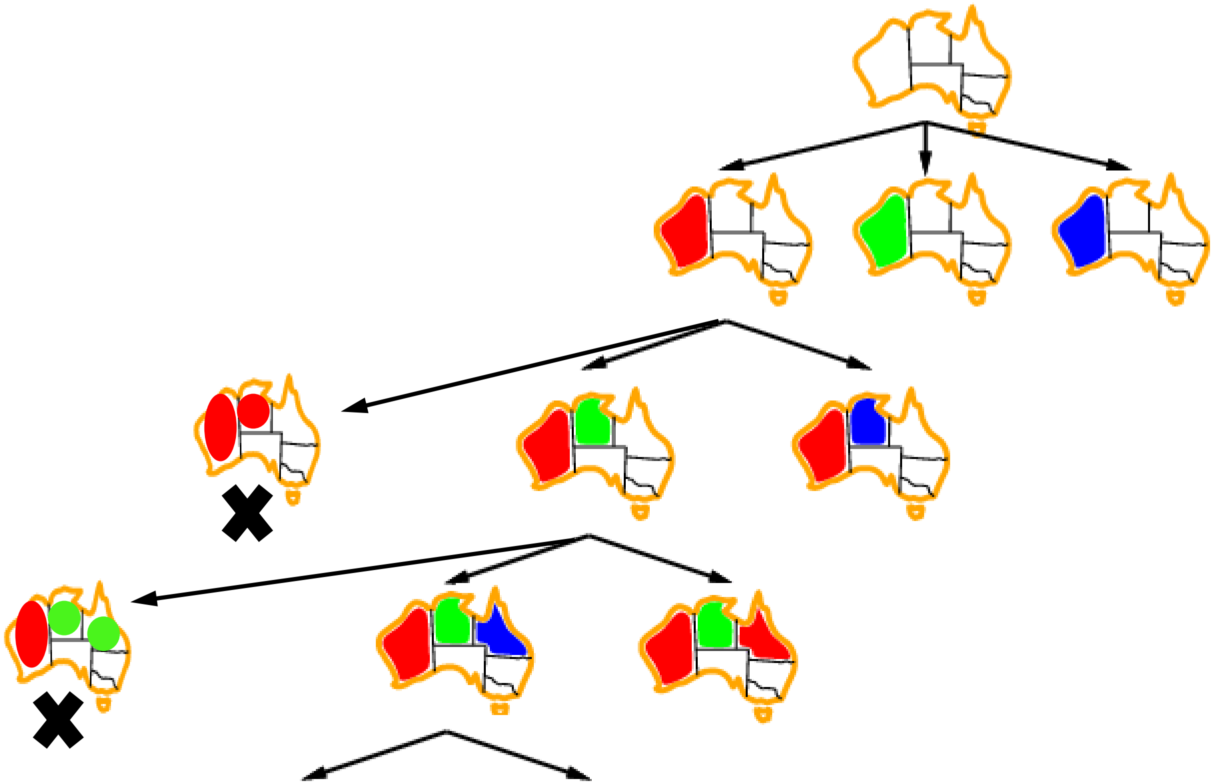
Visszalépéses keresés



Visszalépéses keresés

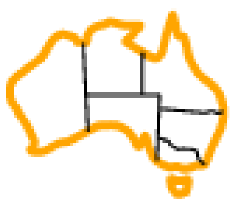


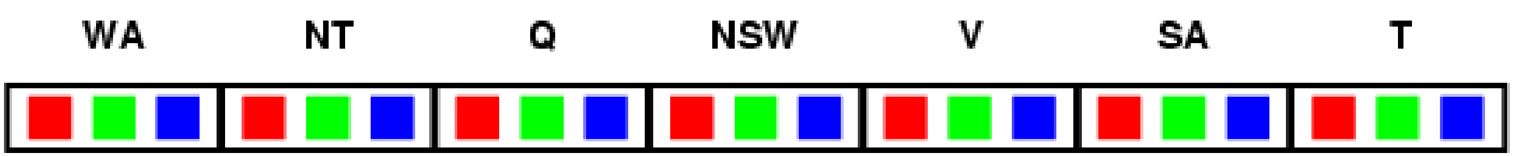
Visszalépéses keresés



Előretekintő ellenőrzés (keresés)

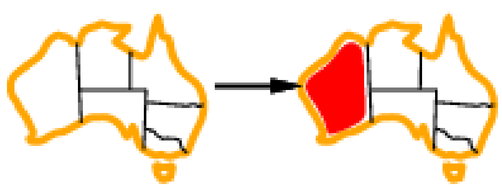
Az előretekintő ellenőrzés minden egyes alkalommal, amikor egy X változó értéket kap, minden, az X-hez kényszerrel kötött, lekötetlen Y-t megvizsgál, és Y tartományából törli az X számára választott értékkel inkonzisztens értékeket.

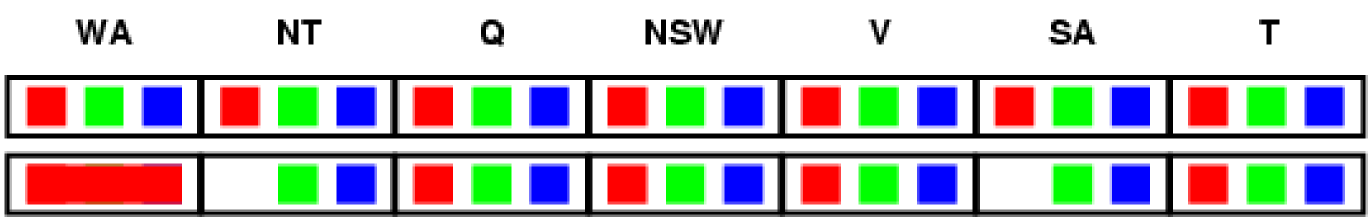




Előretekintő ellenőrzés (keresés)

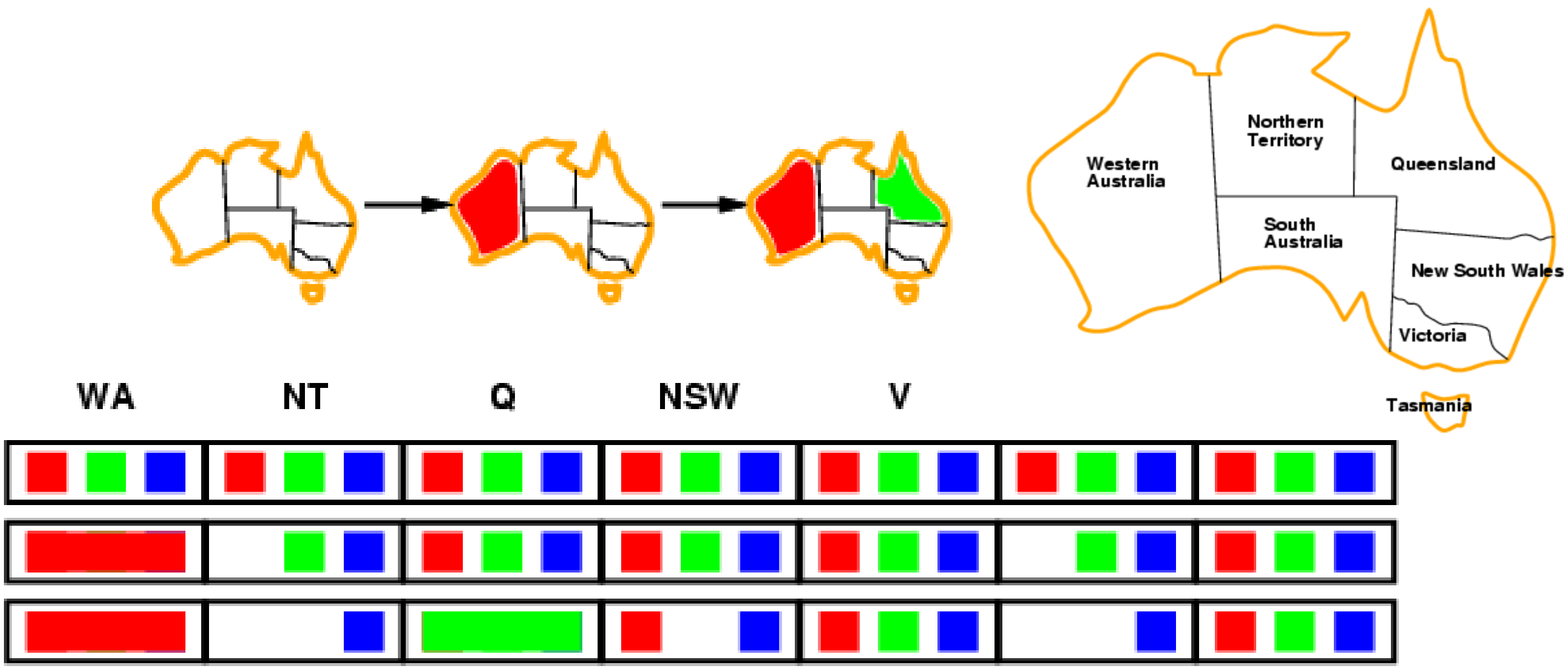
Az előretekintő keresés minden egyes alkalommal, amikor egy X változó értéket kap, minden, az X-hez kényszerrel kötött, lekötetlen Y-t megvizsgál, és Y tartományából törli az X számára választott értékkel inkonzisztens értékeket.





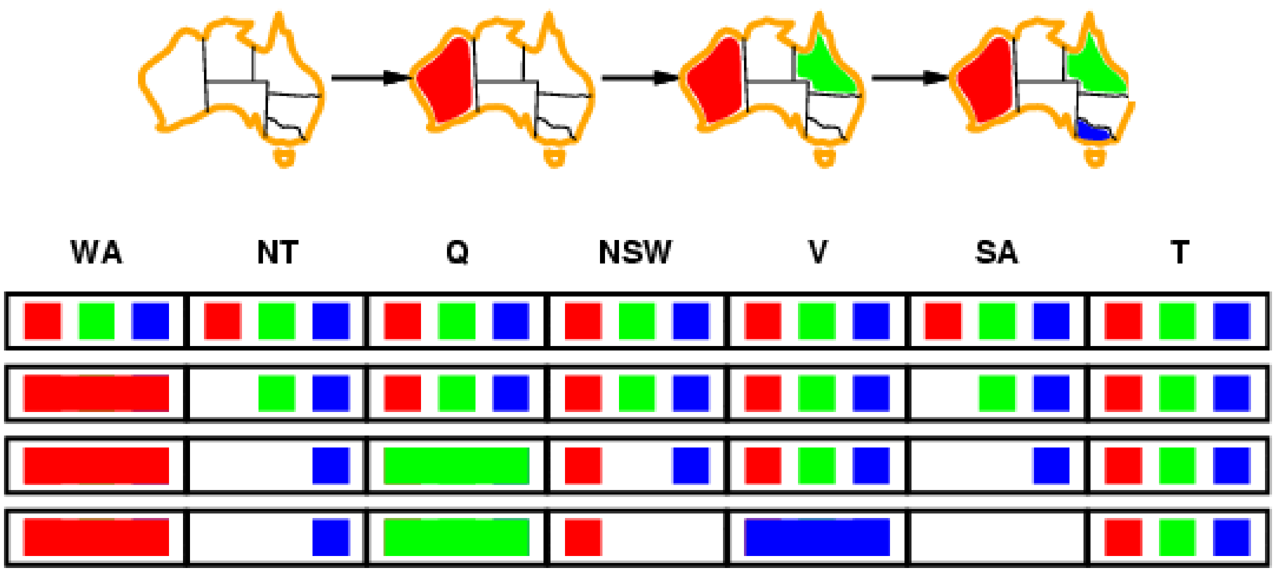
Előretekintő ellenőrzés (keresés)

Az előrenéző ellenőrzés minden egyes alkalommal, amikor egy X változó értéket kap, minden, az X-hez kényszerrel kötött, hozzárendeletlen Y-t megvizsgál, és Y tartományából törli az X számára választott értékkel inkonzisztens értékeket.



Előretekintő ellenőrzés (keresés)

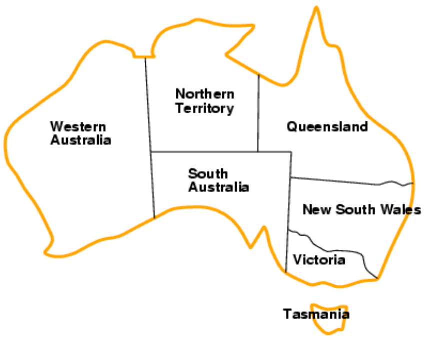
Az előrenéző ellenőrzés minden egyes alkalommal, amikor egy X változó értéket kap, minden, az X-hez kényszerrel kötött, hozzárendeletlen Y-t megvizsgál, és Y tartományábó törli az X számára választott értékkel inkonzisztens értékek

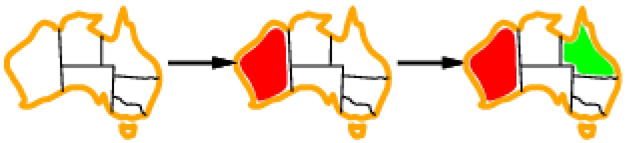


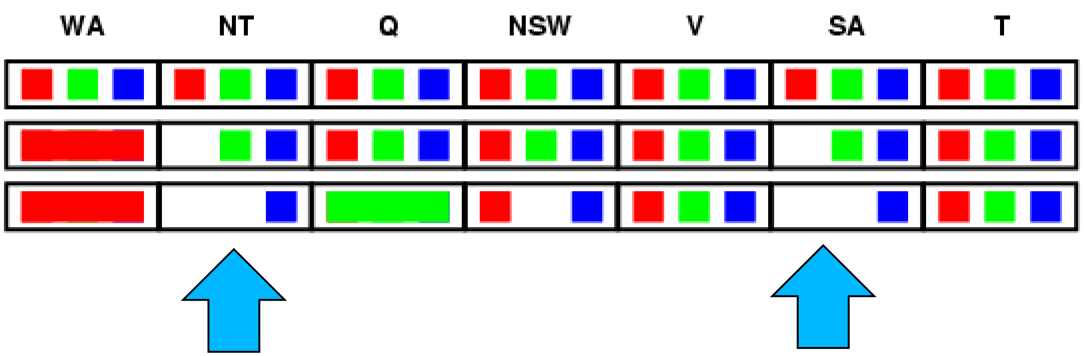
Korlátozás előreterjesztése

Az előretekintő ellenőrzés ugyan sok inkonzisztenciát észrevesz, de nem mindent.

Ráadásul nem látja jól előre a kudarcokat.







NT és SA egyszerre nem lehet kék.

Visszalépéses keresés hatékonyságának növelése (általános heurisztikák CSP-khez)

Általános módszerekkel is komoly gyorsítást el lehet érni:

- Melyik változóval foglalkozzunk a legközelebb?
- Milyen sorrendben vizsgáljuk az értékeit?
- Érzékelhetjük-e jó előre a kudarcokat?

(korai nyesés)

ezek az ún. tárgyterület-független heurisztikák

Melyik változóval foglalkozzunk először, és milyen értéket rendeljünk hozzá?

Kényszerek:

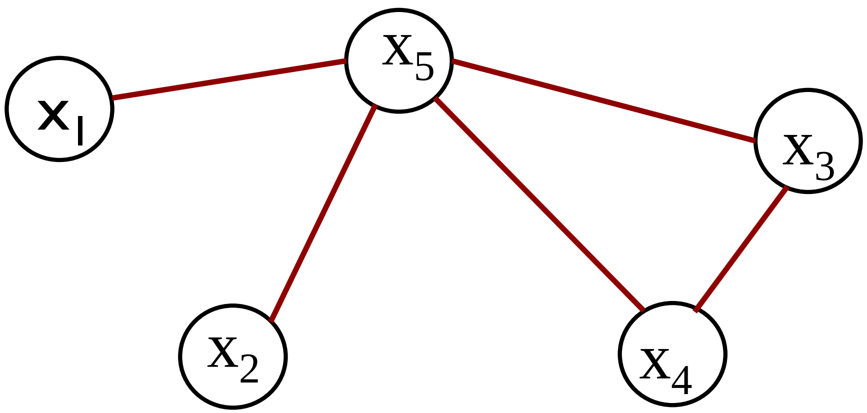
$x_1 > x_5$

$x_2 > 1 + x_5$

$x_3 > 2 \cdot x_5$

$x_4 > x_5 - 1$

$x_3 = 2 + x_4$



Mindegyik változó értékkészlete az egyjegyű nem negatív egészek halmaza: {0,1,2,…,9}

A. $x1 \Leftarrow 5$
B. $x2 \Leftarrow 1$
C. $x3 \Leftarrow 2$
D. $x5 \Leftarrow 0$

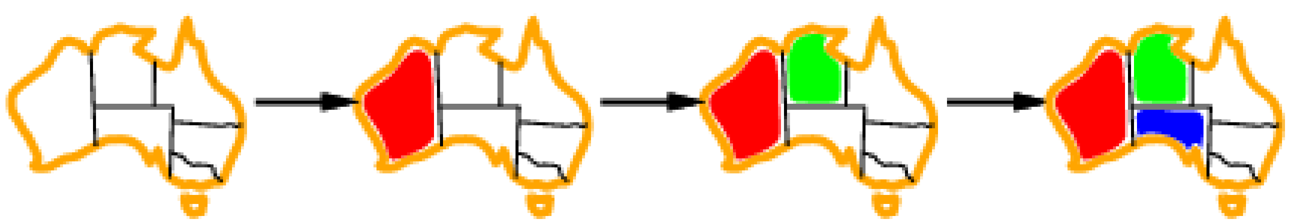
1. ***A legkevesebb fennmaradó érték*** ötlete (melyik <u>változót</u> válasszuk?)

A leginkább korlátozott változó:

a legkisebb számú megengedett értékkel rendelkező változóval kezdjünk, ill. folytassunk (lokálisan kicsi az elágazási tényező!)

= **legkevesebb fennmaradó érték** heurisztika

(min remaining variables, MRV)

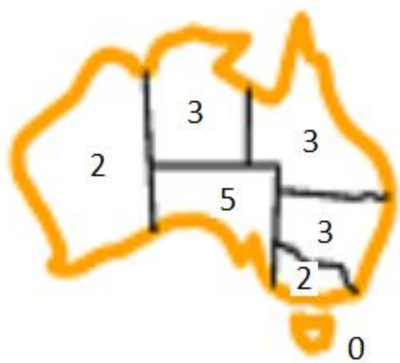


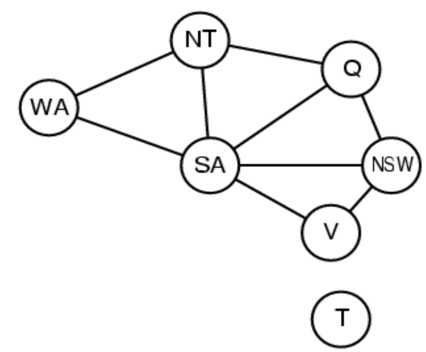
(NT ill. SA csak 2 megengedett érték (piros már nem lehet), minden más 3)

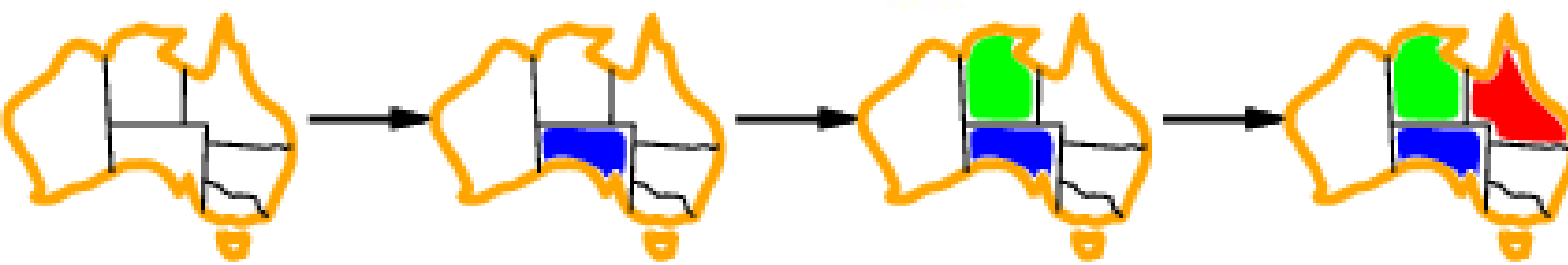
2. ***Fokszám heurisztika*** **ötlete** (melyik <u>változót</u> válasszuk?)

Az MRV-heurisztika semmit sem segít abban, hogy melyik régiót válasszuk ki elsőként Ausztrália kiszínezésekor, mert a kiinduláskor mindegyik régiónak három megengedett színe van.

A *későbbi választások* elágazási tényezőjét csökkentheti, ha azt a változót választjuk ki, amely a legtöbbször szerepel a még hozzárendeletlen változókra vonatkozó kényszerekben.

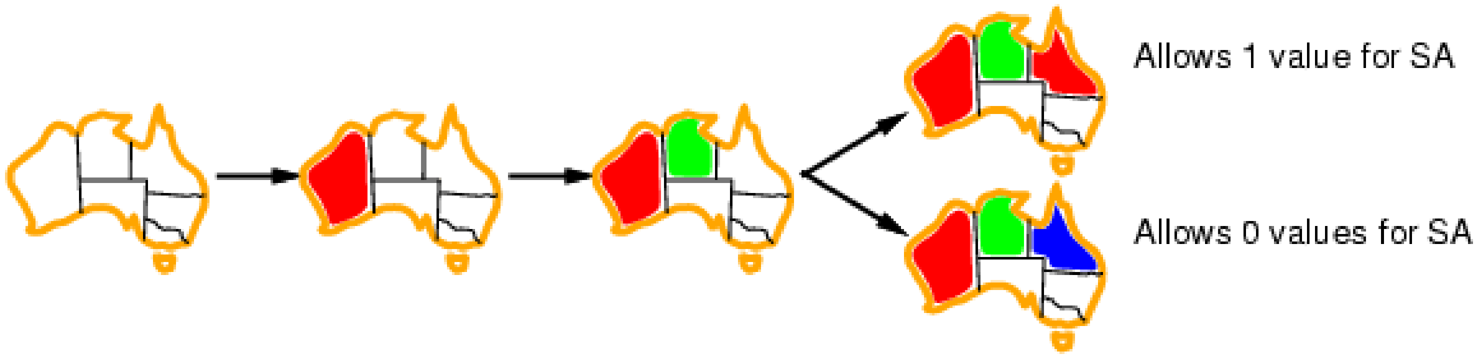






3. ***A legkevésbé korlátozó <u>érték</u>*** ötlete

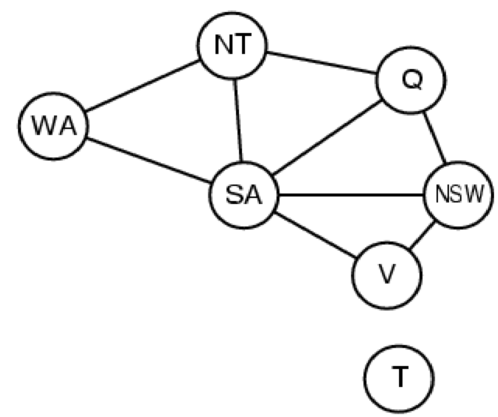
Előnyben részesítjük azt az <u>értéket</u>, amely a legkevesebb választást zárja ki a kényszergráfban a szomszédos változóknál.

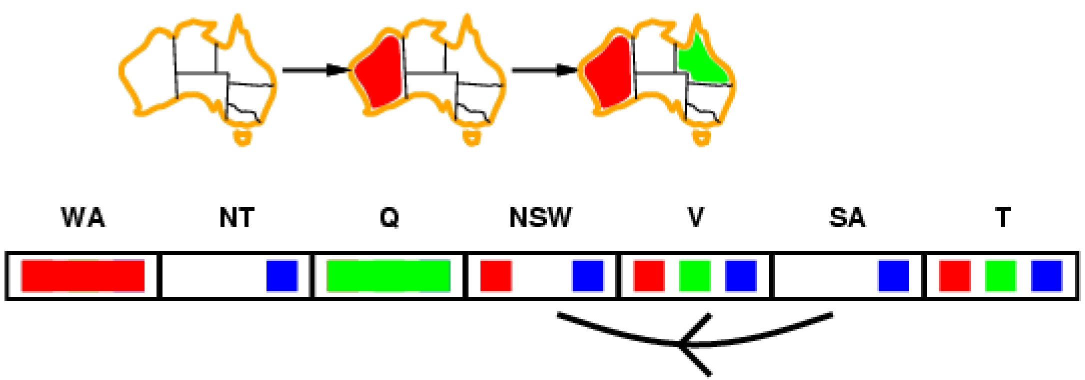


Élkonzisztencia

X → *Y* él konzisztens akkor és csak akkor, ha

X minden x értékére létezik Y-nak valamilyen megengedett y értéke

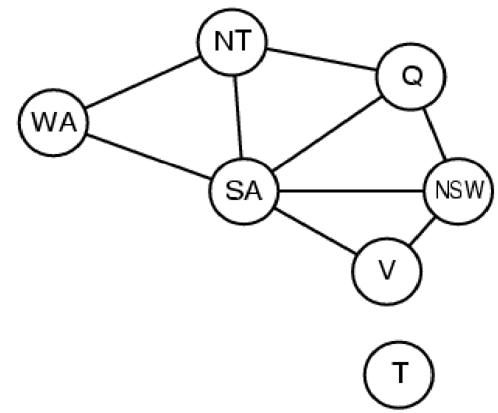


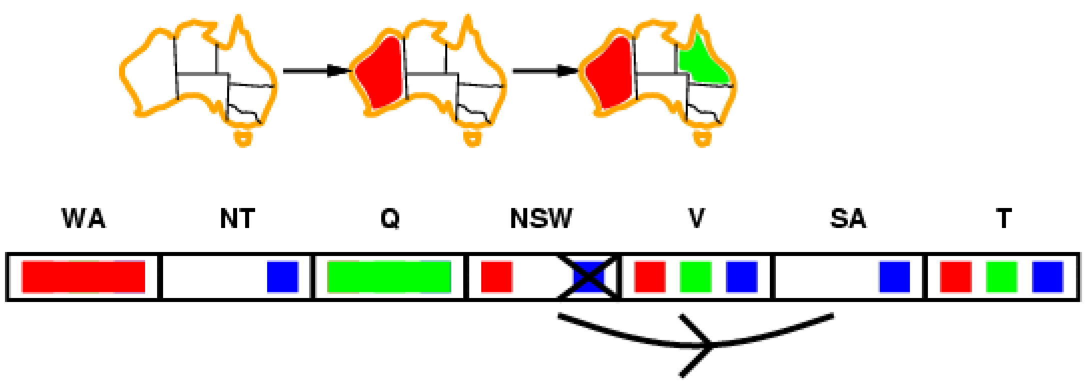


Élkonzisztencia

$X \rightarrow Y$ él konzisztens akkor és csak akkor, ha

X minden x értékére létezik Y-nak valamilyen megengedett y értéke

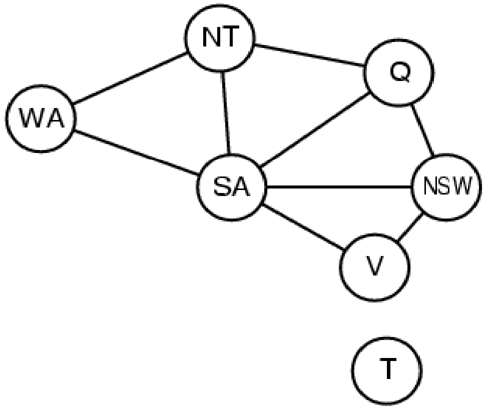


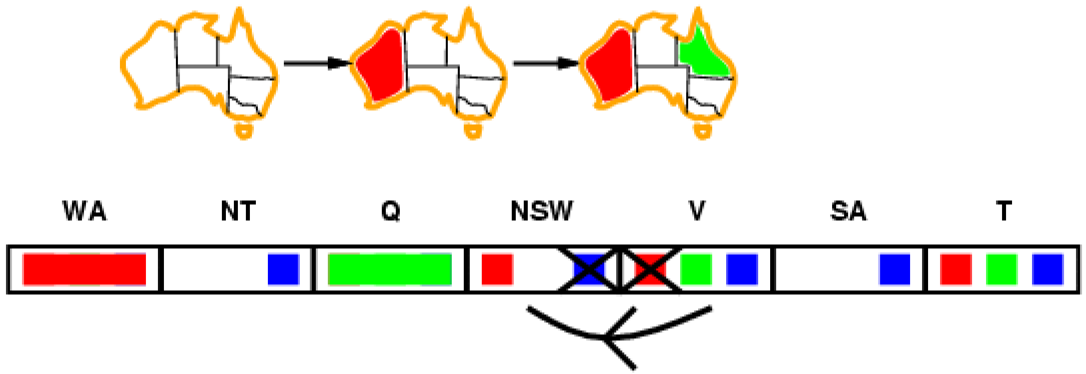


Élkonzisztencia

$X \rightarrow Y$ él konzisztens akkor és csak akkor, ha
X minden x értékére létezik Y-nak valamilyen megengedett y értéke

Ha X értéket veszít, a szomszédjait újra kell ellenőrizni

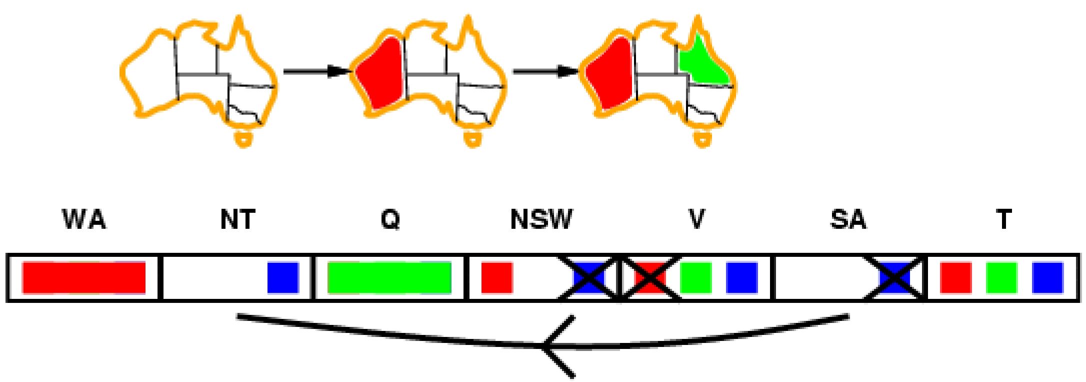


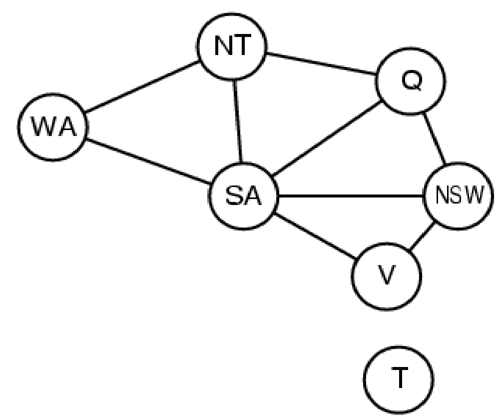


Élkonzisztencia

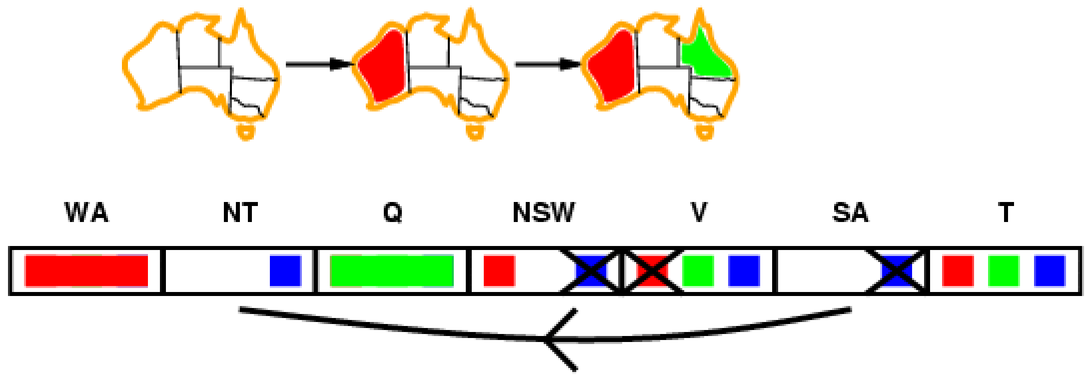
$X \rightarrow Y$ él konzisztens akkor és csak akkor, ha X minden x_j értékére létezik Y-nak valamilyen megengedett y_k értéke.

Ha X értéket veszít, akkor a szomszédjait (akikkel valamilyen kényszerkapcsolatban van) újra kell ellenőrizni





Élkonzisztencia



Az élkonzisztencia-ellenőrzés alkalmazható:

- **előfeldolgozó lépésként** a keresés megkezdése előtt, vagy a
- keresési folyamat minden egyes hozzárendelését követő **terjesztési lépésként**

CSP lokális kereséssel

Kiindulás: teljes állapotleírás = minden változónak van értéke (esetleg rossz, nem teljesült kényszerekkel)

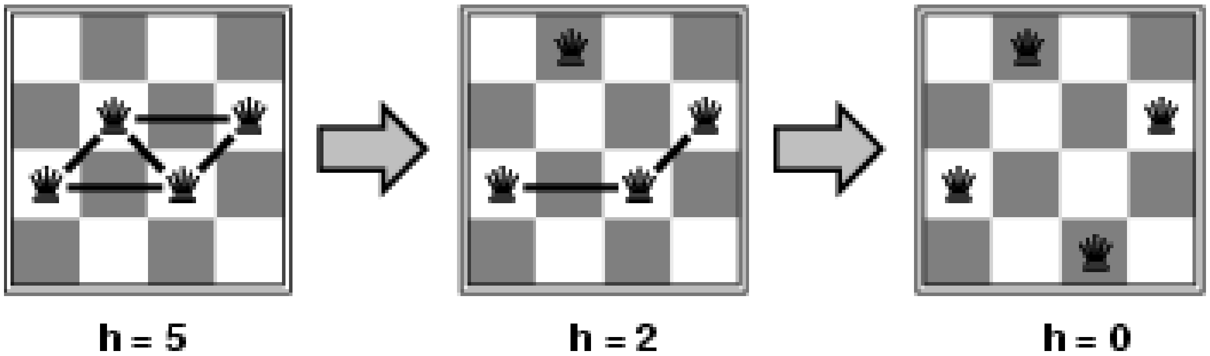
Operátorok: megváltoztatják a változók hozzárendelését, hogy csökkenjen a sérült kényszerek száma

Változó szelekció: véletlen módon, bármely konfliktusban lévő (valamelyik kényszer sérül) változót választhatjuk

CSP lokális kereséssel

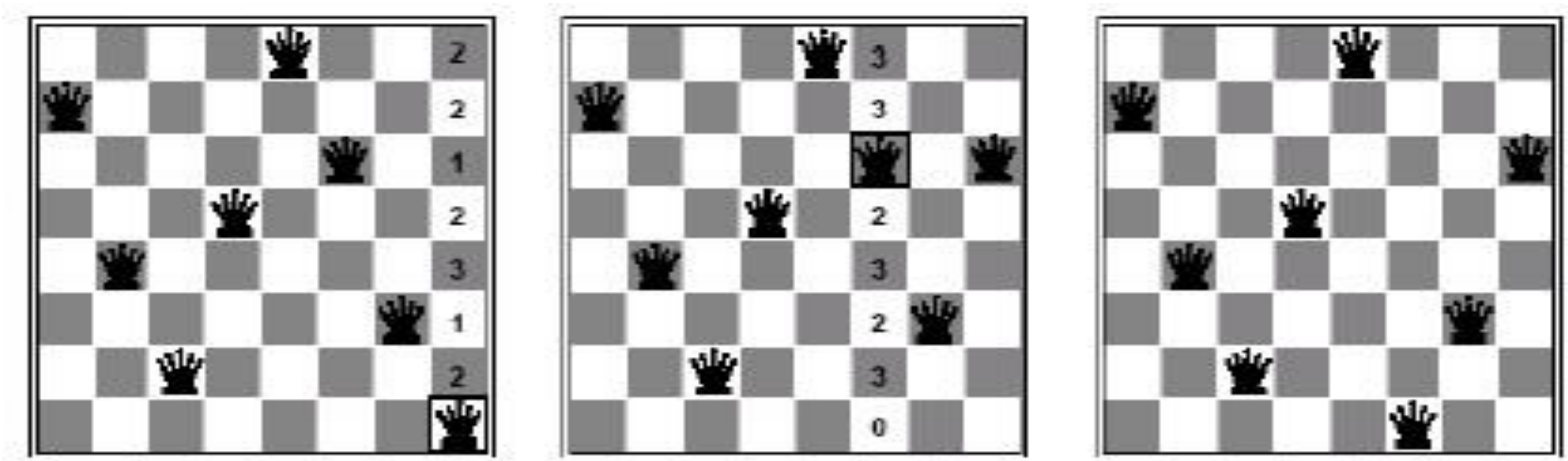
Min. konfliktus heurisztika:

azt az értéket állítjuk be, amely a legkevesebb számú korlátot sérti, pl. hegymászó: *h(n)* = sérült korlátok száma, *h*(*n*) csökkentése a cél (itt most lefele mászunk a völgybe)



h(n) = a támadások száma

CSP lokális kereséssel / Min. konfliktus heurisztika



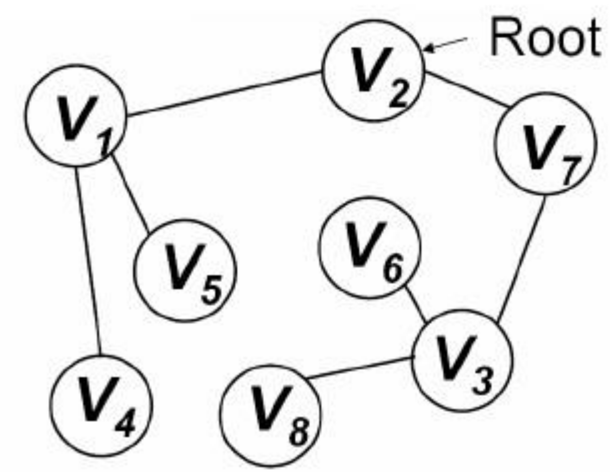
Min. konfliktus heurisztika nagyon hatékony, nagy valószínűséggel gyorsan old meg nagyon nagy problémaeseteket (10 millió királynő!).

CSP struktúrája – miben segíthet?

CSP gráfja: független komponensek … (komplexitás-számítás)

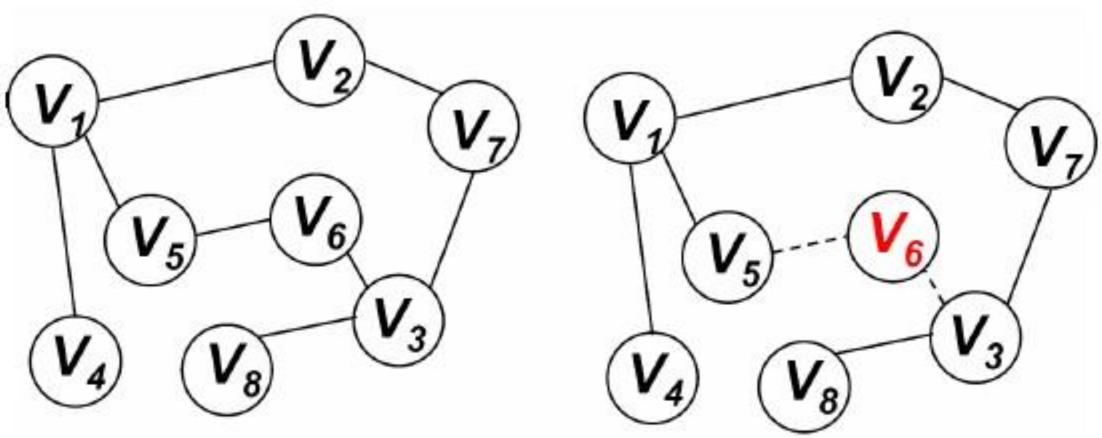
CSP fa gráf: megoldás könnyű, változószámban lineáris

válasszunk egy levelet gyökérnek (lokális keresés kiinduló áll.)
élkonzisztencia-nyesés gyerekektől a szülőik felé



CSP struktúrája – miben segíthet?

CSP hurkos gráf: megoldás általánosságban NP nehéz



Gráf CSP konvertálása fába:
- vágóhalmaz
- fa-**dekompozíció**

Megoldáskeresés
V_6 minden értékére

Mesterséges intelligencia előadássorozat

Az előadás diái az AIMA könyvre épülve (http://aima.cs.berkeley.edu) készültek a University of California, Berkeley mesterséges intelligencia kurzusának anyagainak felhasználásával (http://ai.berkeley.edu).

These slides are based on the AIMA book (http://aima.cs.berkeley.edu) and were adapted from the AI course material of University of California, Berkeley (http://ai.berkeley.edu).

Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Méréstechnika és Információs Rendszerek Tanszék

Mesterséges intelligencia

Keresés ellenséges környezetben és a játékok

Előadó: Dr. Hullám Gábor

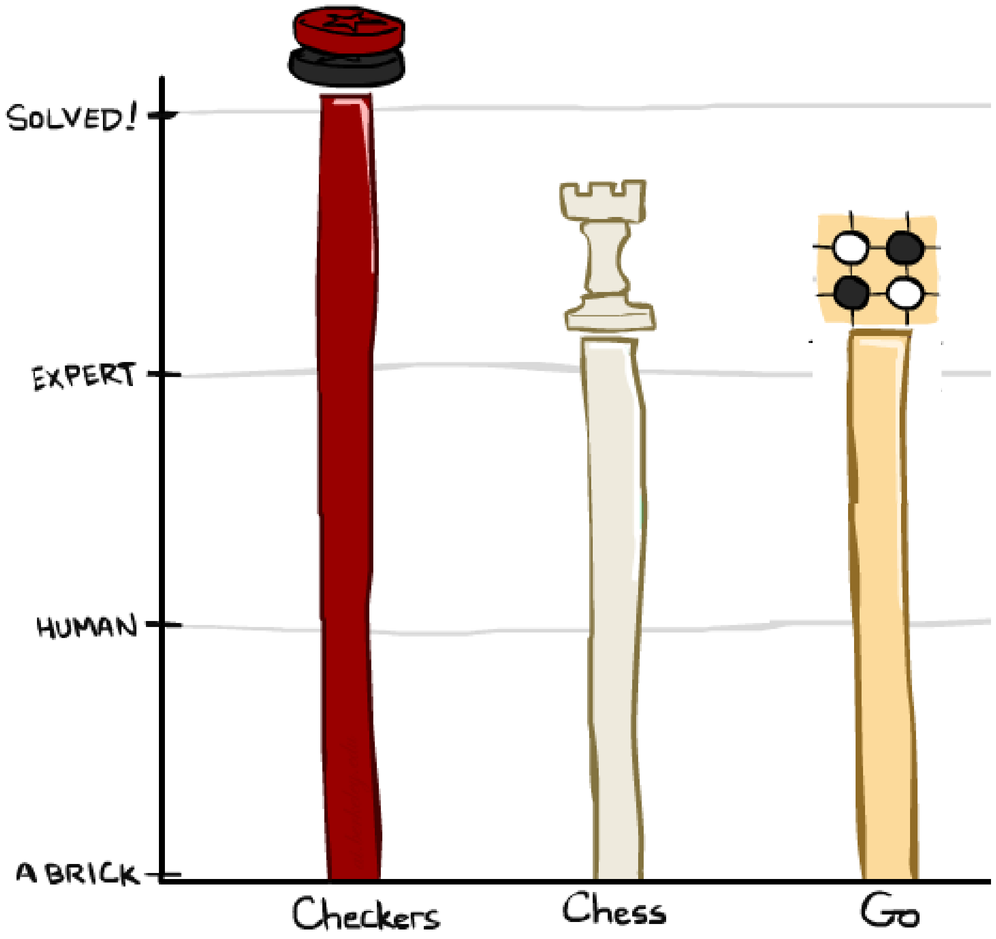
Előadás anyaga: Dr. Gézsi András
Dr. Hullám Gábor

Játékok – a létező legjobb MI játékosok

- **Dámajáték:** 1950: Első számítógépes játékos. 1994: Első számítógépes bajnok: Chinook törte meg az emberi bajnok, Marion Tinsley 40 évig tartó bajnoki címét, majd Tinsley visszavonulása után átvette a bajnoki címet. 2007: Dámajáték megoldása!

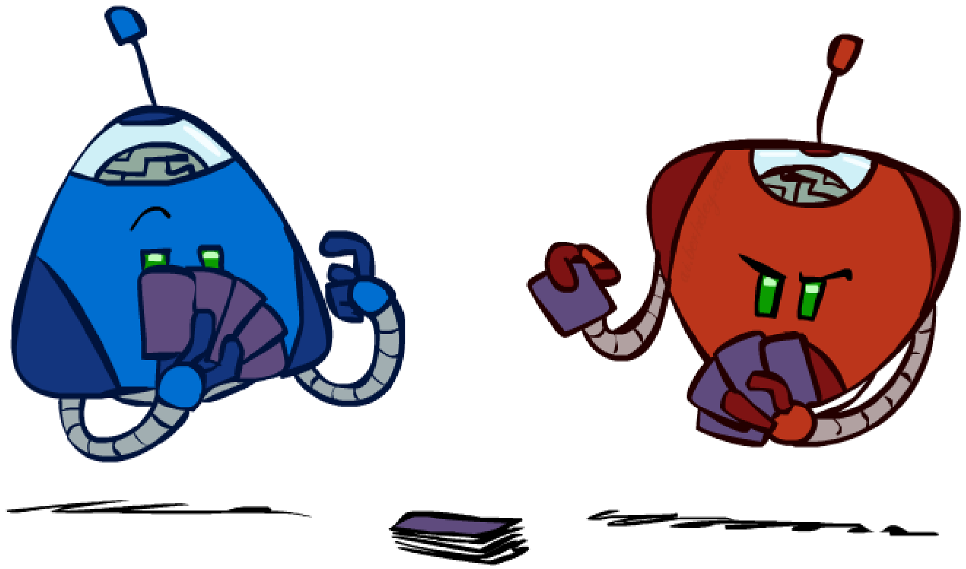
- **Sakk:** 1997: Deep Blue legyőzte az emberi bajnokot, Gary Kasparovot egy hat játékból álló meccsben. Deep Blue 200 millió pozíciót vizsgált meg másodpercenként, rendkívül szofisztikált kiértékelést és egyéb nem nyilvános módszereket használt, amelyekkel a keresést akár 40 lépés mélységig is ki tudta terjeszteni. A jelenlegi programok még jobbak (pl. Stockfish, AlphaZero).

- **Go:** 2015-ig amatőr szint, 2016: AlphaGO legyőzte az emberi bajnokot. 2018: AlphaZero legyőzte AlphaGO-t (Monte Carlo fa keresés (MCTS), megerősítéses tanulás, neurális hálók, önmagával való játék)



Játékok típusai

- Rendkívül sokféle játék létezik

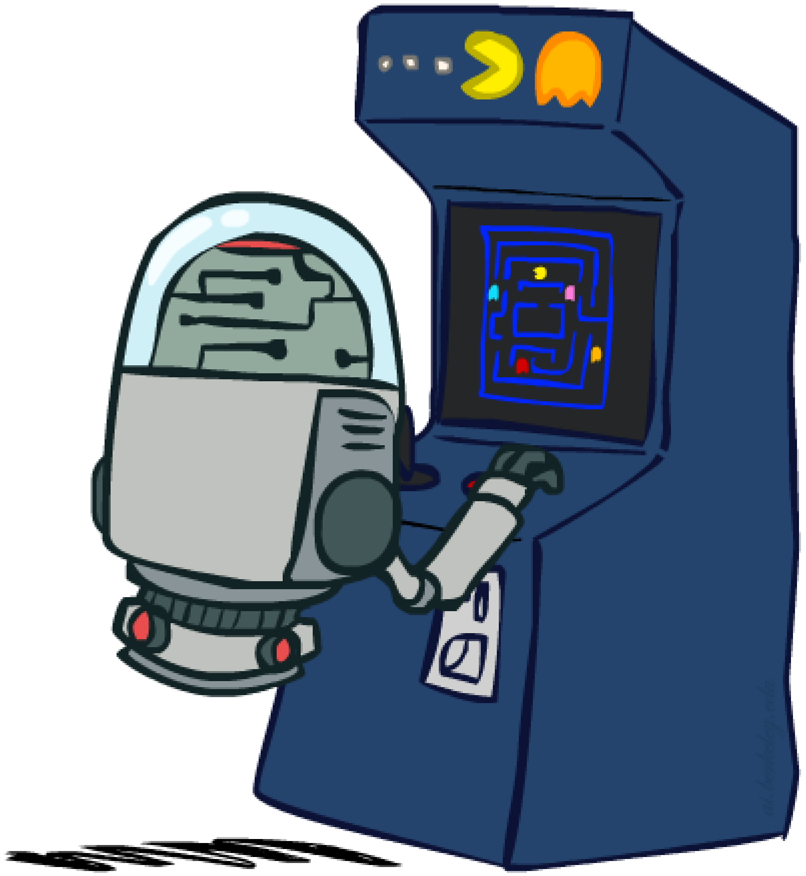


- Tulajdonságok:
  - Determinisztikus vagy sztochasziktus?
  - Egy, kettő vagy több játékos?
  - Zéró összegű játék?
  - Teljes információ (láthatjuk az állapotot)?

- Algoritmus = stratégiát ad
  optimális/jó választ javasol minden egyes állapotban

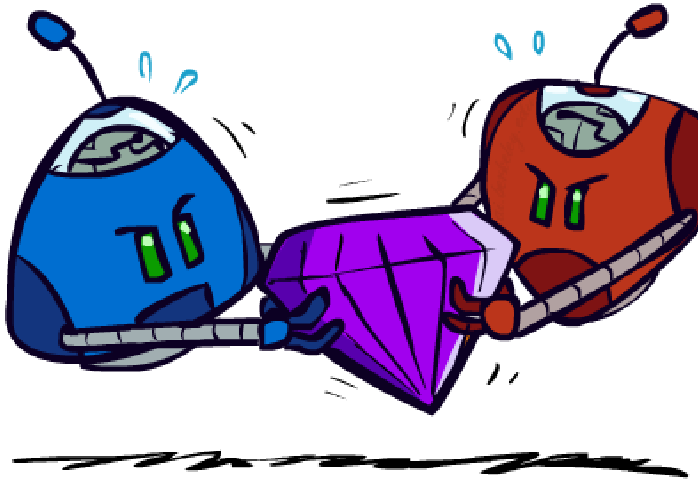
Determinisztikus játékok

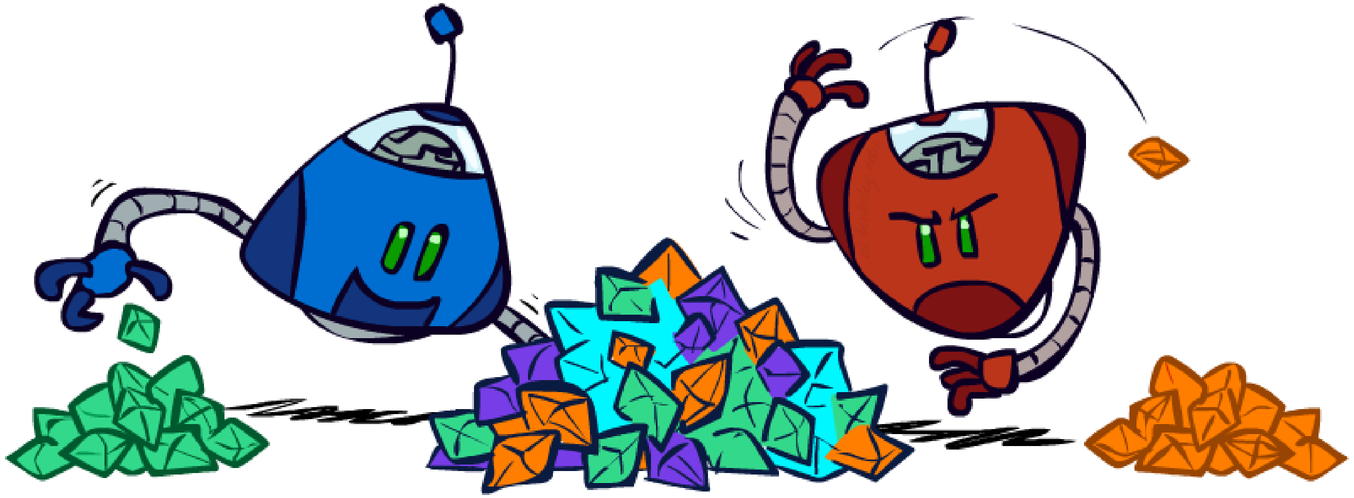
- Egy lehetséges formalizmus:
  - Állapotok: S (s_0 kezdőállapot)
  - Játékosok: P={1...N} (rendszerint felváltva lépnek)
  - Cselekvések: A (a soron következő játékos
    és az aktuális állapot szabja meg)
  - Állapotátmenet függvény: SxA $\rightarrow$ S
  - Célállapot teszt: S $\rightarrow$ {igaz,hamis}
  - Célállapot hasznosság: SxP $\rightarrow$ R



- A megoldás egy játékos számára a **stratégia**: S $\rightarrow$ A

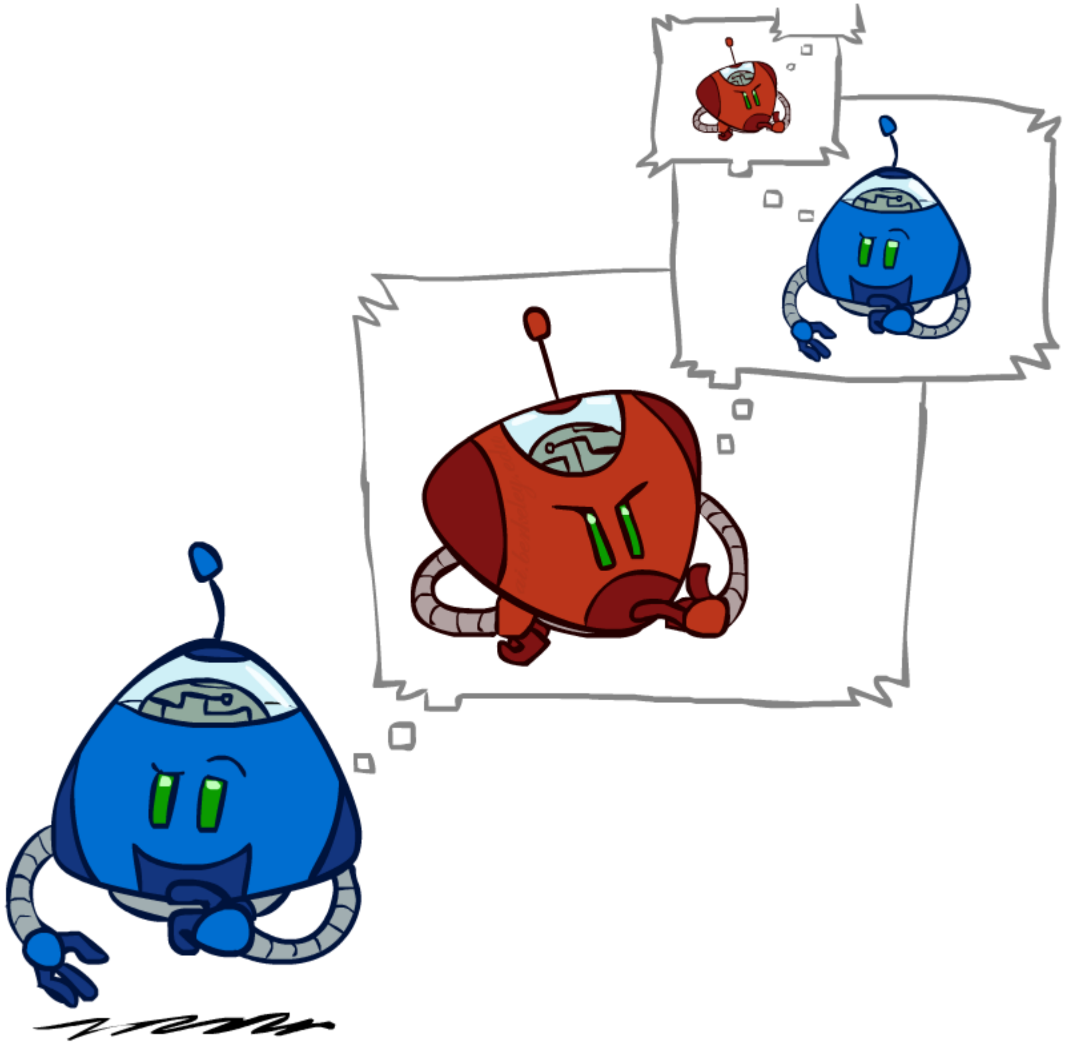
Zéró összegű játékok



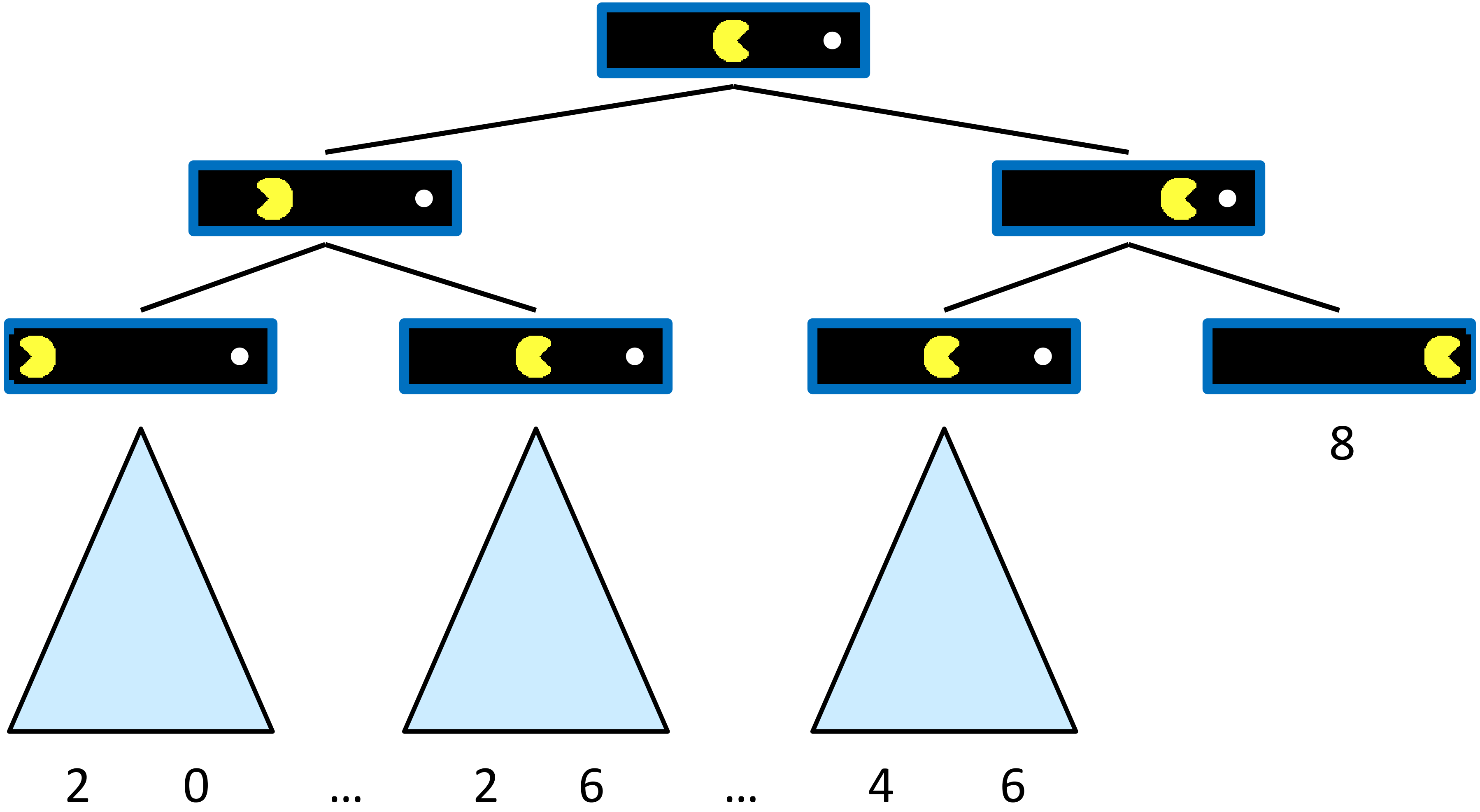


- Zéró összegű játékok
  - Az ágensek hasznossága (a kimenetelek értéke) ellenkező
  - Egyedüli értékként is tekinthető, amelyet az egyik játékos maximalizálni akar, a másik pedig minimalizálni
  - Ellenséges játékos, tiszta versengés
- Általános (nem zéró összegű) játékok
  - Az ágensek hasznossága (a kimenetelek értéke) független
  - Kooperáció, közömbösség, versengés stb. mind elképzelhető

Keresés ellenséges környezetben



Egy-ágenses játékfa

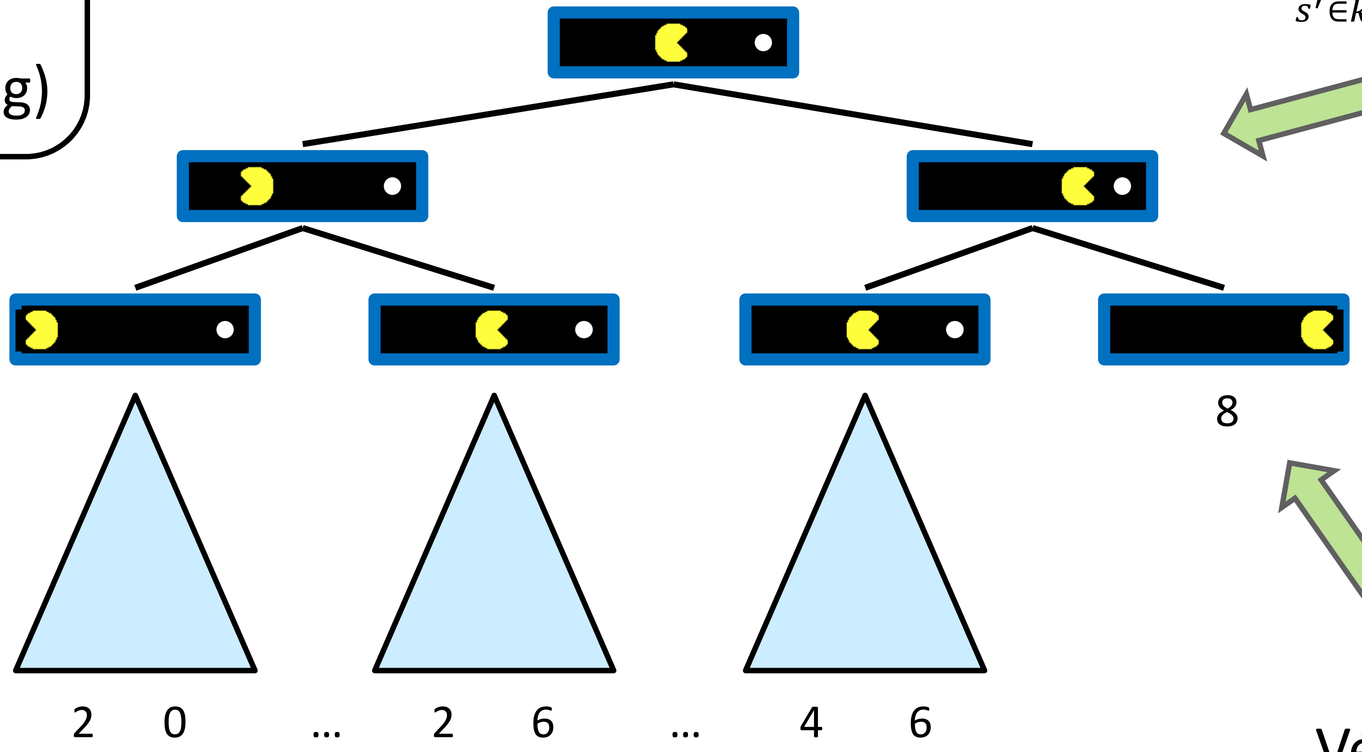


Egy állapot értéke

Egy állapot értéke:
Az adott állapotból
elérhető legjobb
kimenetel (hasznosság)

Nem végállapotok:

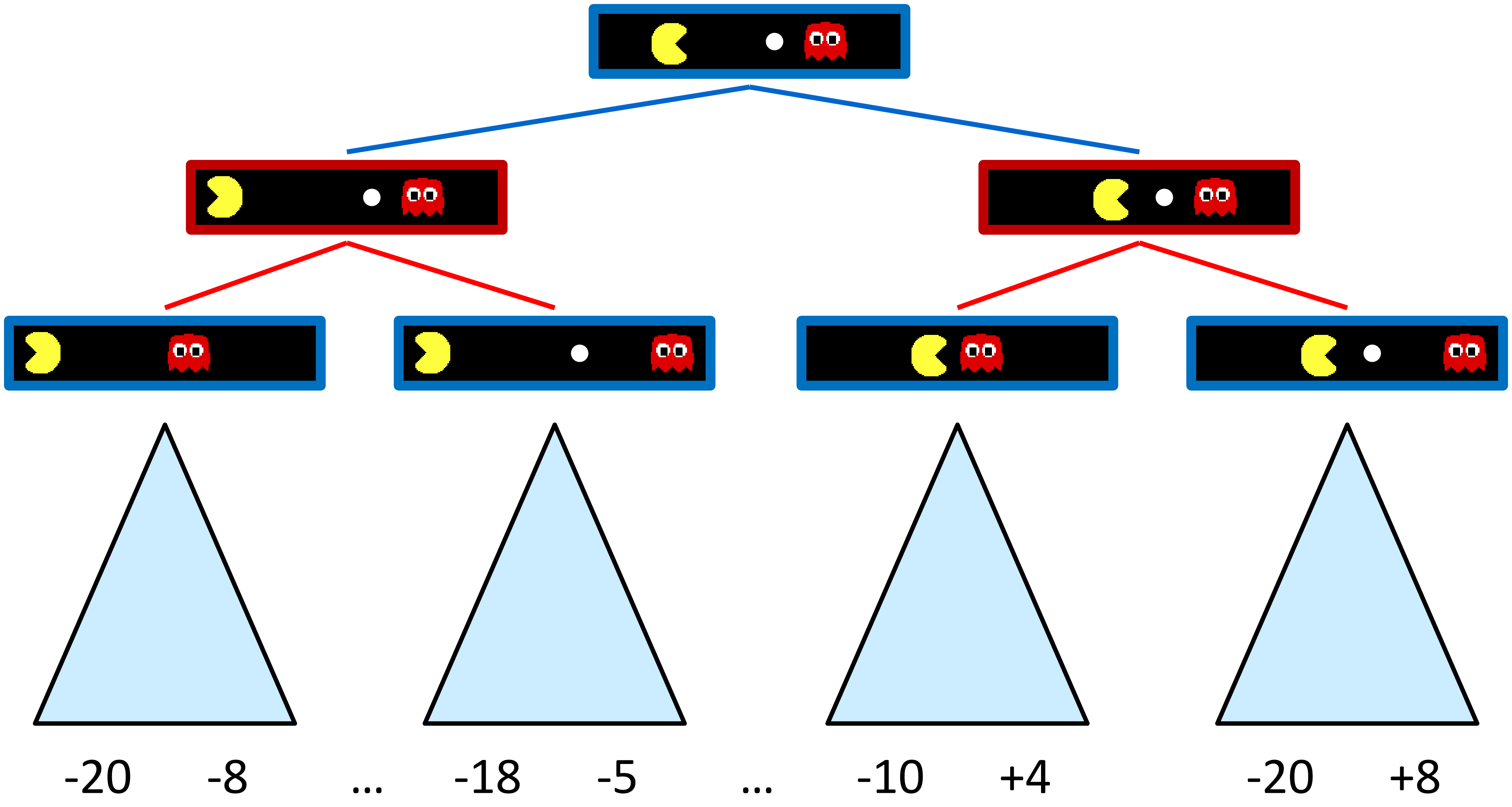
$$V(s) = \max_{s' \in követő-állapot(s)} V(s')$$



Végállapotok:

$$V(s) = ismert$$

Két-ágenses játékfa (Ellenség melletti keresés)



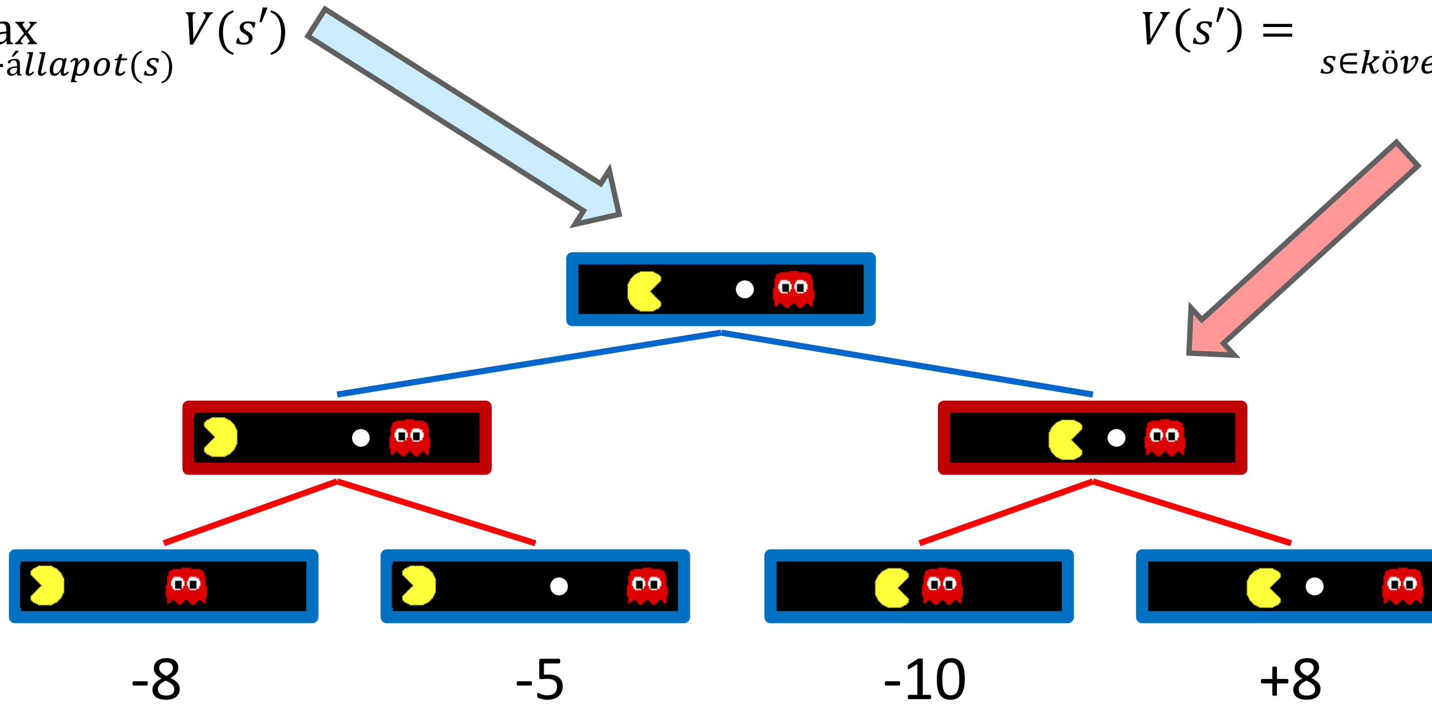
Minimax értékek

Az ágens által kontrollált állapotok:

$$V(s) = \max_{s' \in követő-állapot(s)} V(s')$$

Az ellenfél által kontrollált állapotok:

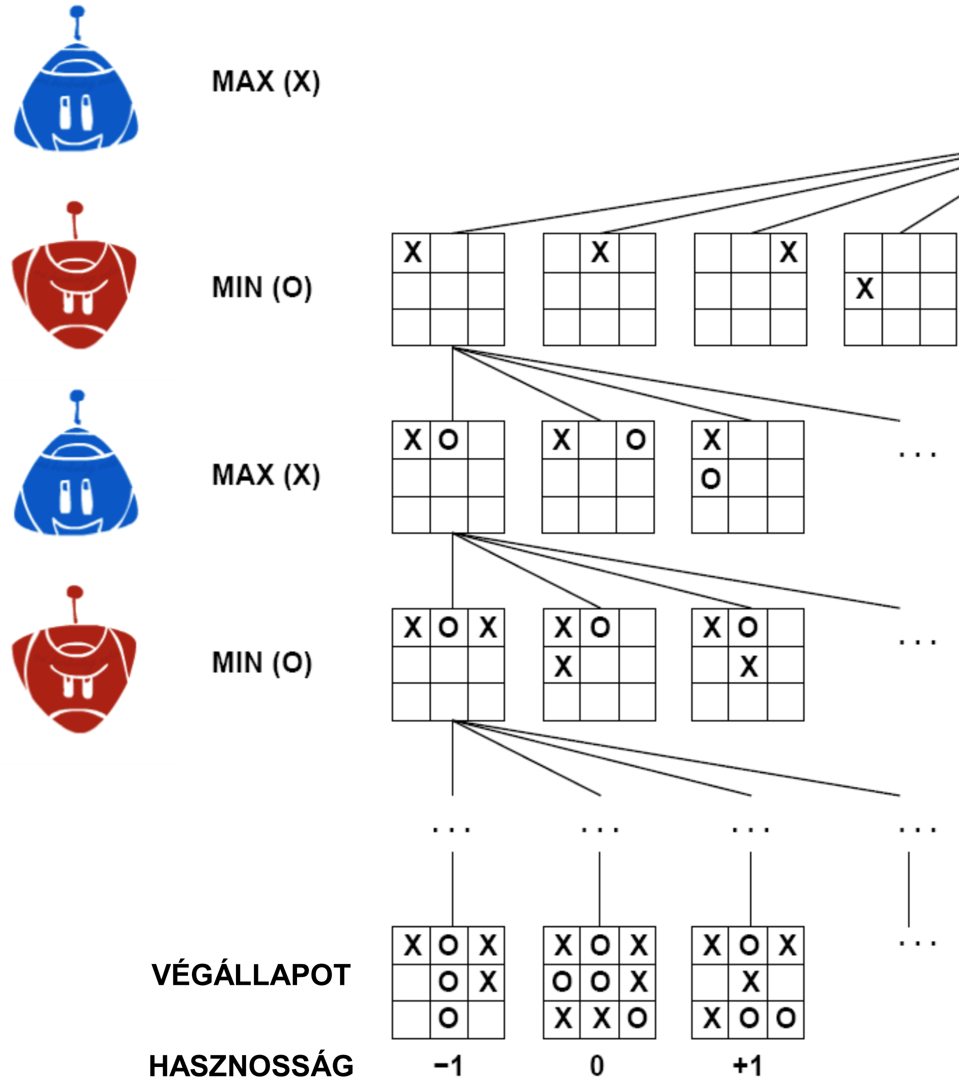
$$V(s') = \min_{s \in követő-állapot(s')} V(s)$$

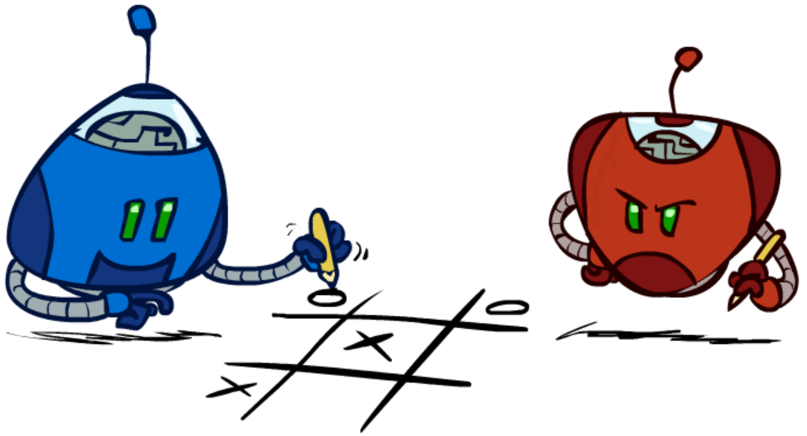


Végállapotok:

$$V(s) = ismert$$

Tic-Tac-Toe (3×3-as amőba) játékfa



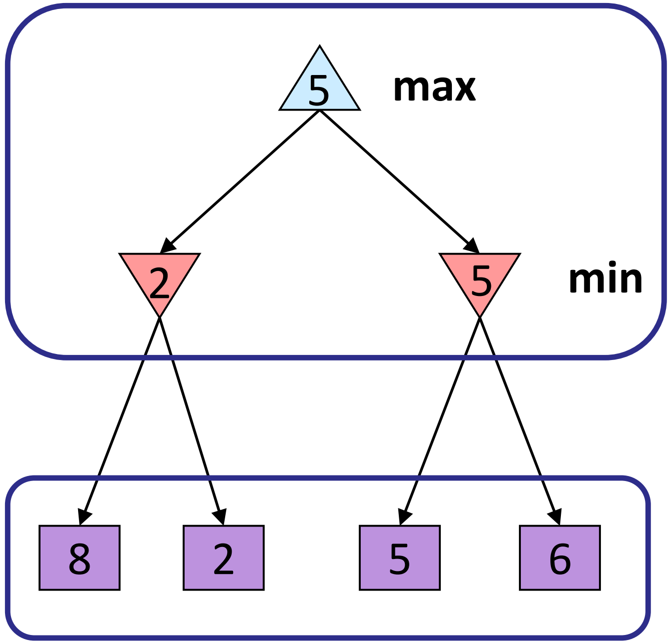


Keresés ellenség mellett (Minimax)

- Determinisztikus, zéró összegű játékok:
  - Tic-tac-toe, sakk, dáma stb.
  - Az egyik játékos maximalizálja az eredményt
  - A másik játékos minimalizálja az eredményt

- Minimax keresés:
  - Keresés állapottér fában (játékfa)
  - A játékosok felváltva lépnek
  - Kiszámítja minden csomópont minimax értékét: a legnagyobb elérhető hasznosság egy racionális (optimálisan játszó) ellenséggel szemben

Minimax értékek: rekurzívan kiszámítva



Végállapotok értéke: a játék szabályaiból kiszámítva

Minimax implementáció

def max-érték(*állapot*):
    inicializál v = -∞
    for each *követő-állapot* **of** *állapot*:
        v = max(v, min-érték(*követő-állapot*))
    return v

def min-érték(*állapot*):
    inicializál v = +∞
    for each *követő-állapot* **of** *állapot*:
        v = min(v, max-érték(*követő-állapot*))
    return v

$$V(s) = \max_{s' \in követő-állapot(s)} V(s')$$

$$V(s') = \min_{s \in követő-állapot(s')} V(s)$$

Minimax implementáció

def érték(*állapot*):

  if *állapot* egy végállapot: **return** az *állapot* hasznossága

  if a következő ágens MAX: **return** max-érték(*állapot*)

  if a következő ágens MIN: **return** min-érték(*állapot*)

def max-érték(*állapot*):

  inicializál v = -∞

  for each *követő-állapot* **of** *állapot*:

    v = max(v, érték(*követő-állapot*))

  return v

def min-érték(*állapot*):

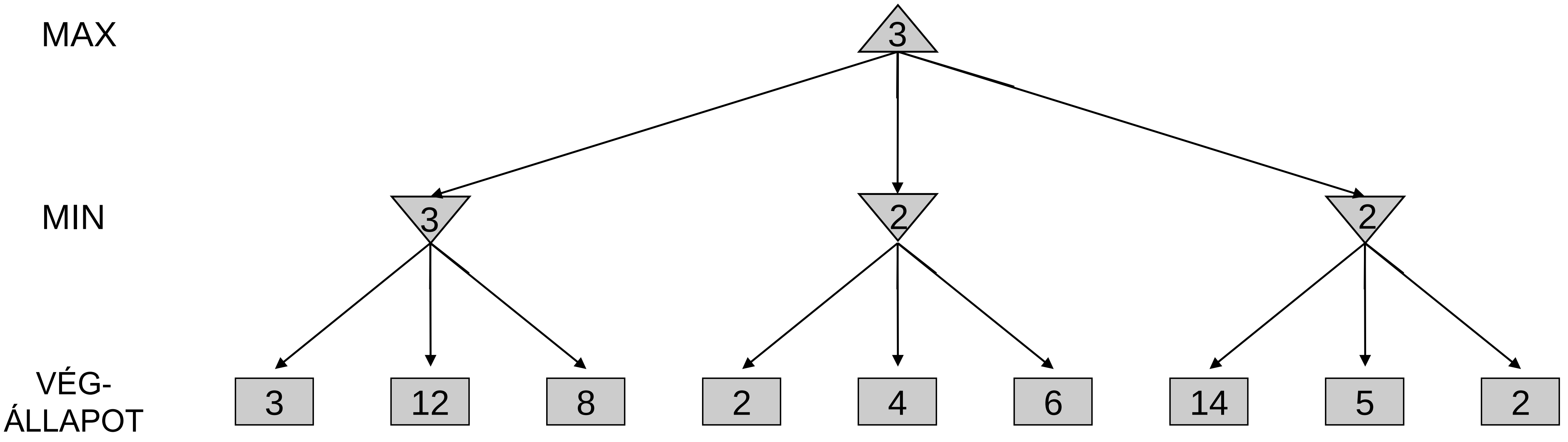
  inicializál v = +∞

  for each *követő-állapot* **of** *állapot*:

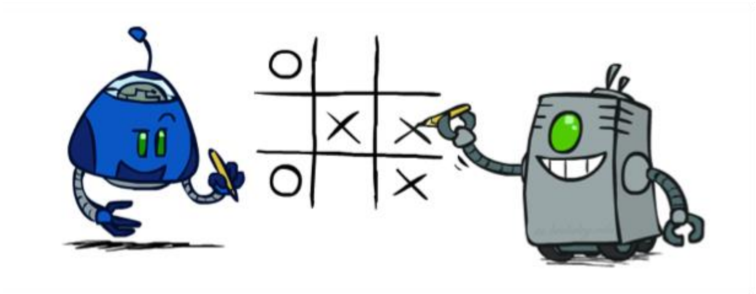
    v = min(v, érték(*követő-állapot*))

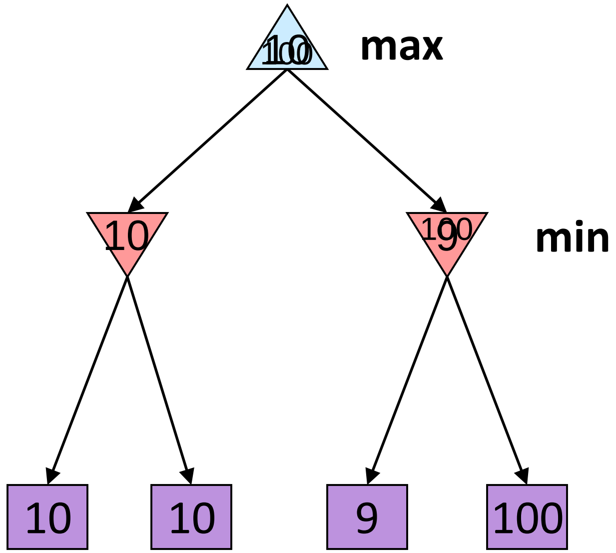
  return v

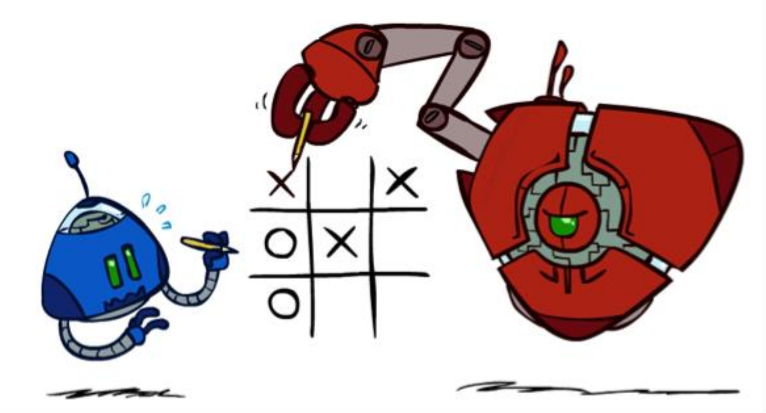
Minimax példa



Minimax tulajdonságok





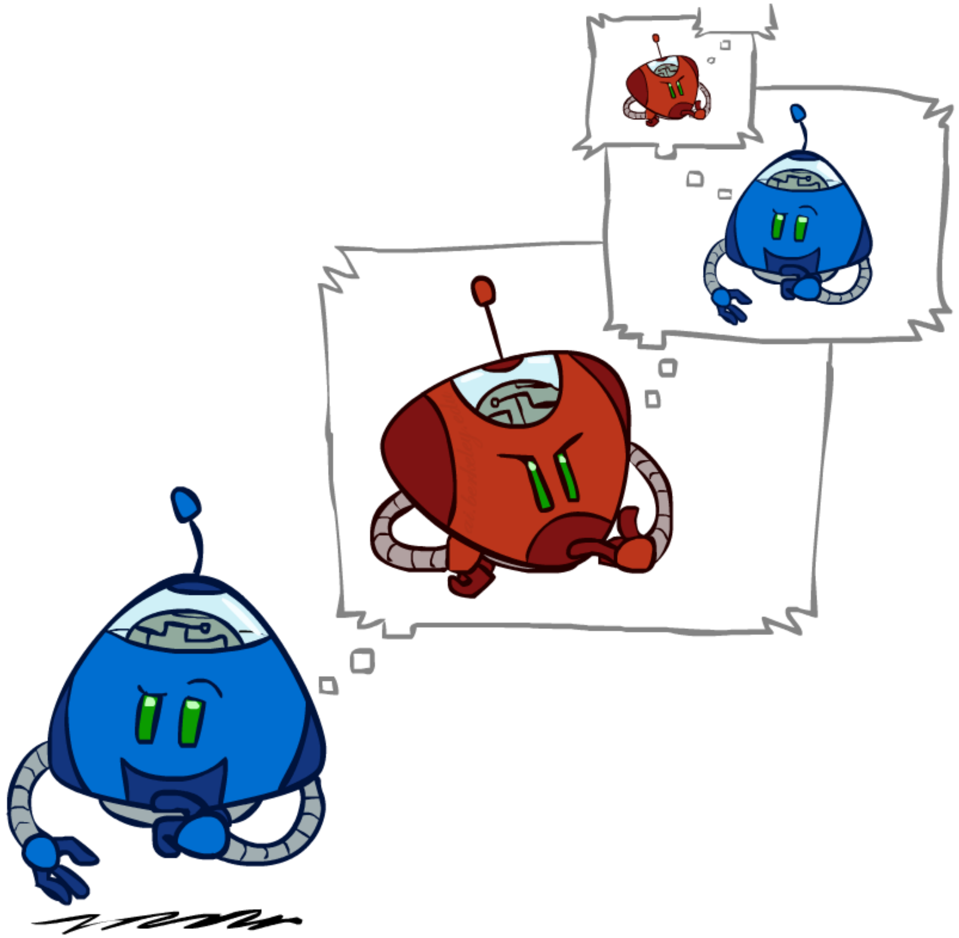


Optimális egy racionális játékossal szemben.
Egyébként?

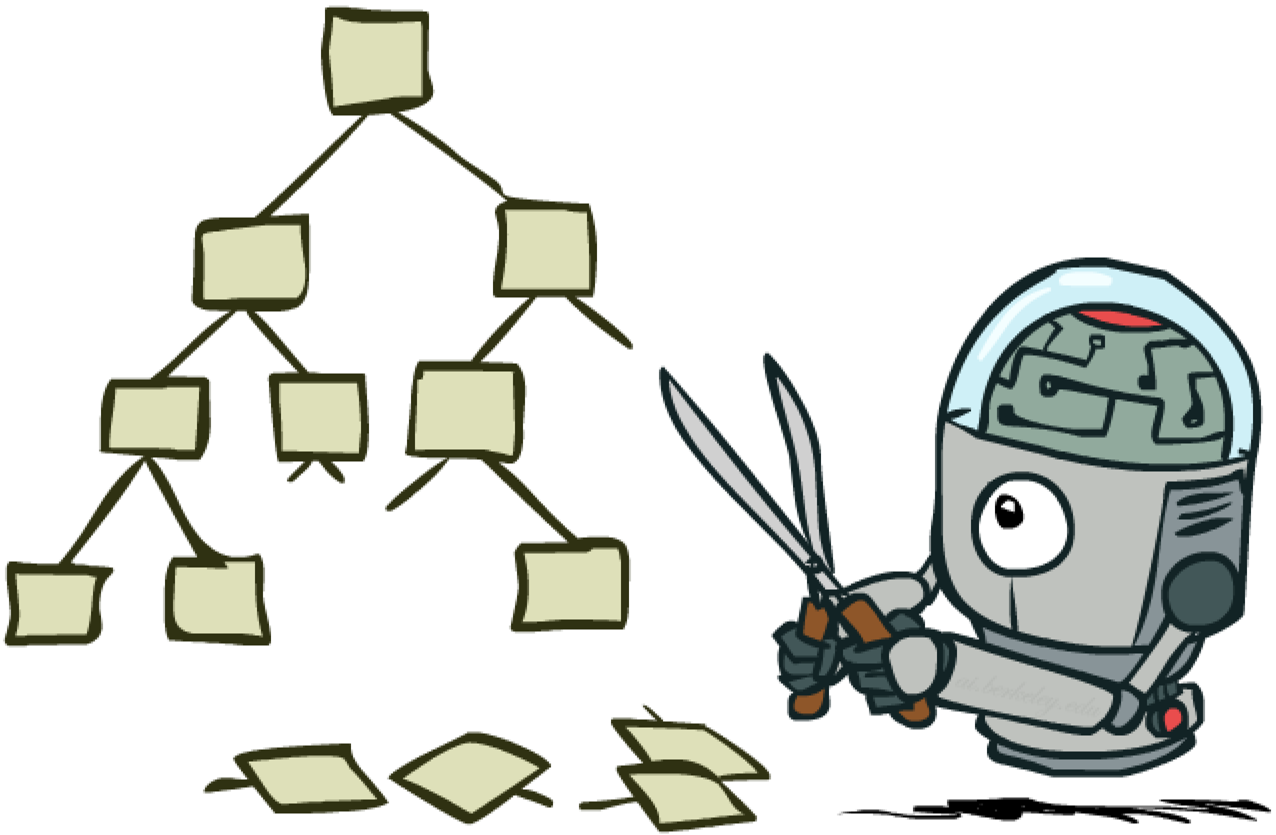
Minimax hatékonysága

- Milyen hatékony a minimax algoritmus?
  - Ugyanannyira, mint a (kimerítő) DFS
  - Idő: $O(b^m)$
  - Tár: $O(bm)$

- Példa: Sakk, $b \approx 35$, $m \approx 100$
  - Egzakt megoldás kivitelezhetetlen
  - Szükséges-e felderítenünk a teljes játékfát?

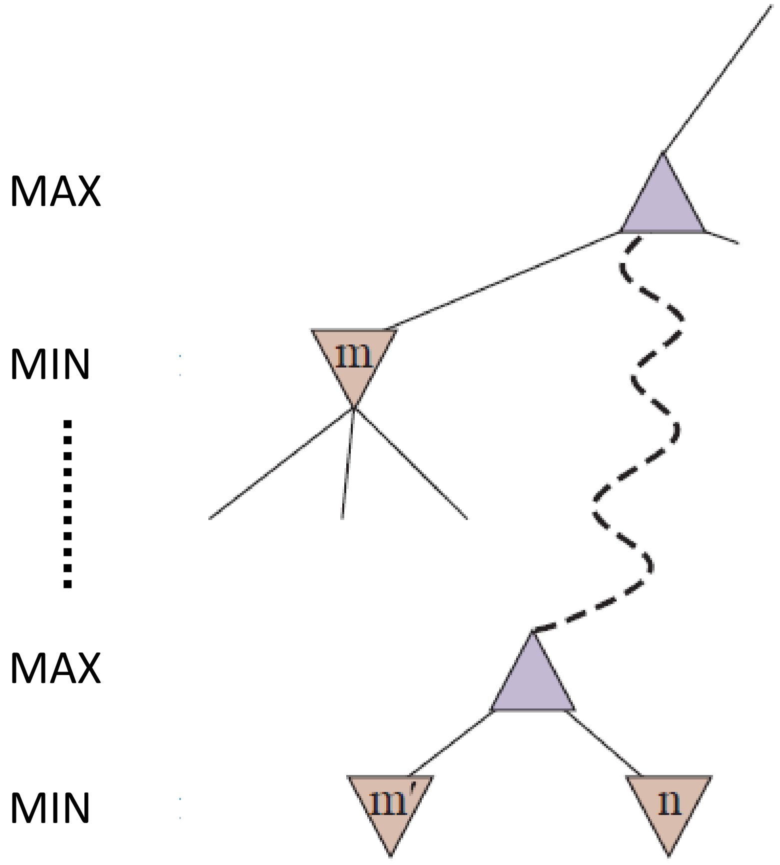


A játékfa nyesése



Alfa-béta nyesés

- Általános eset (MIN szerint)
  - Éppen az *n* csomópont MIN-ÉRTÉKét számítjuk ki
  - Végiglépkedünk *n* gyerekein
  - *n* gyerekeinek minimum értéke egyre csökken(het)
  - Kinek fontos *n* értéke? MAX-nak
  - Legyen *m* és *m’* két másik, már megvizsgált csomópont, amelyeket MAX választhat
  - Ha *n* rosszabbá válik, mint *m* vagy *m’*, akkor MAX el fogja kerülni *n*-t (tudjuk, hogy rossz annyira, hogy MAX nem fogja választani), így nem kell megvizsgálnunk *n* további gyerekeit
- MAX szerint szimmetrikus



Alfa-béta implementáció

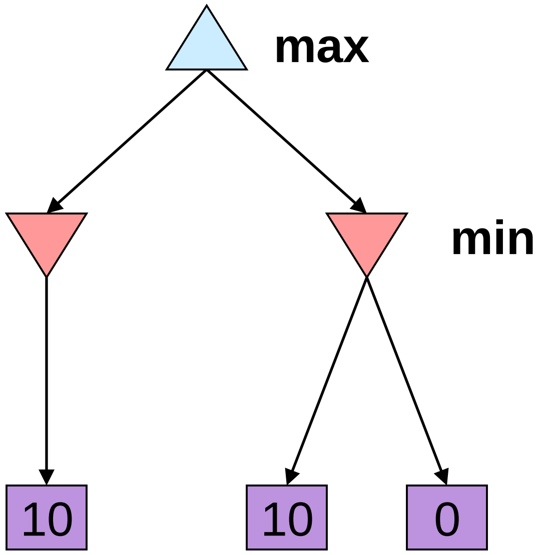
α: MAX legjobb választási lehetősége eddig
β: MIN legjobb választási lehetősége eddig

def max-érték(*állapot*):
    inicializál v = -∞
    for each *követő-állapot* **of** *állapot*:
        v = max(v, érték(*követő-állapot*, α, β))
        if v ≥ β **return** v
        α = max(α, v)
    return v

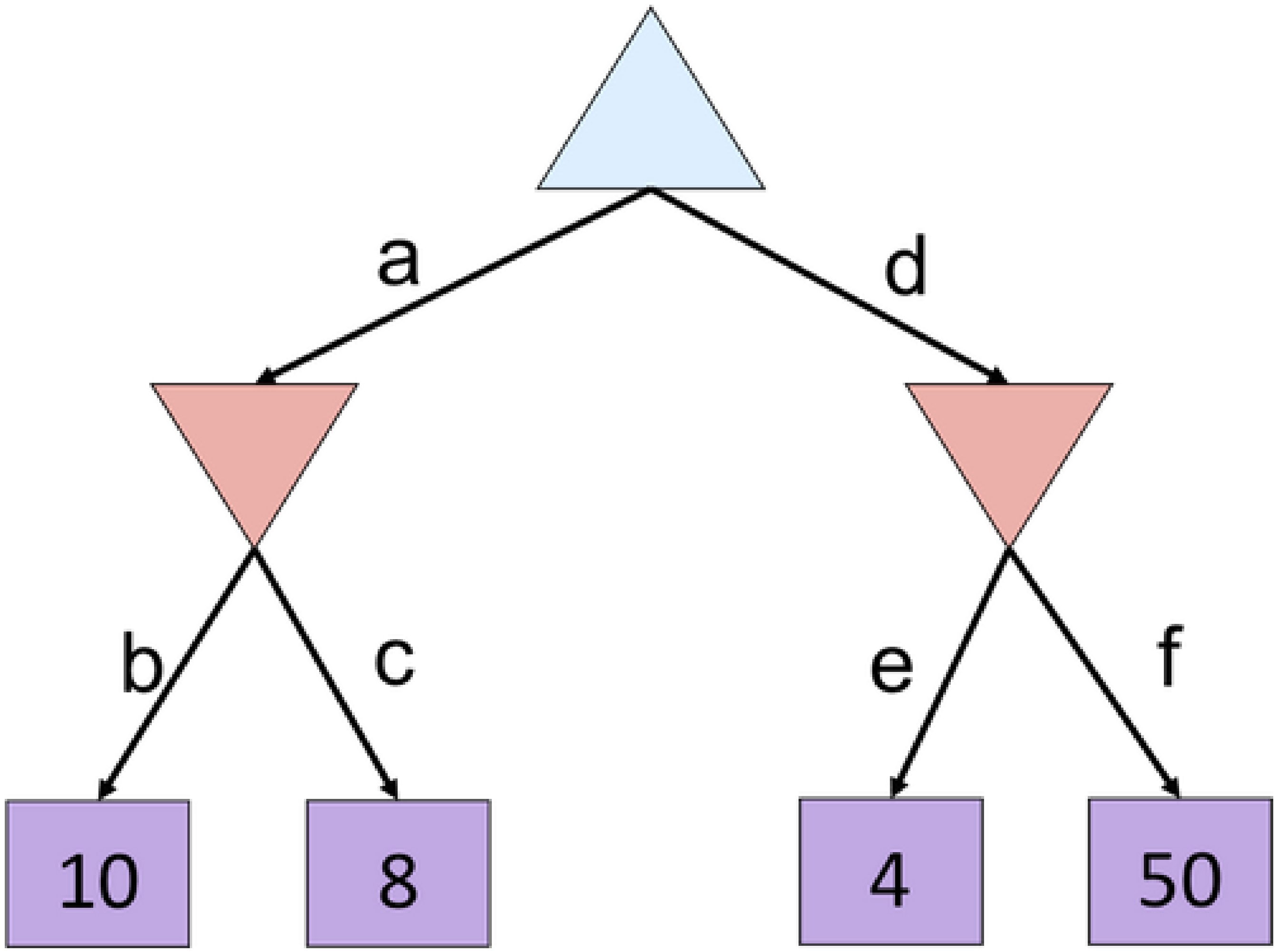
def min-érték(*állapot*):
    inicializál v = +∞
    for each *követő-állapot* **of** *állapot*:
        v = min(v, érték(*követő-állapot*))
        if v ≤ α **return** v
        β = min(β, v)
    return v

Alfa-béta nyesés tulajdonságai

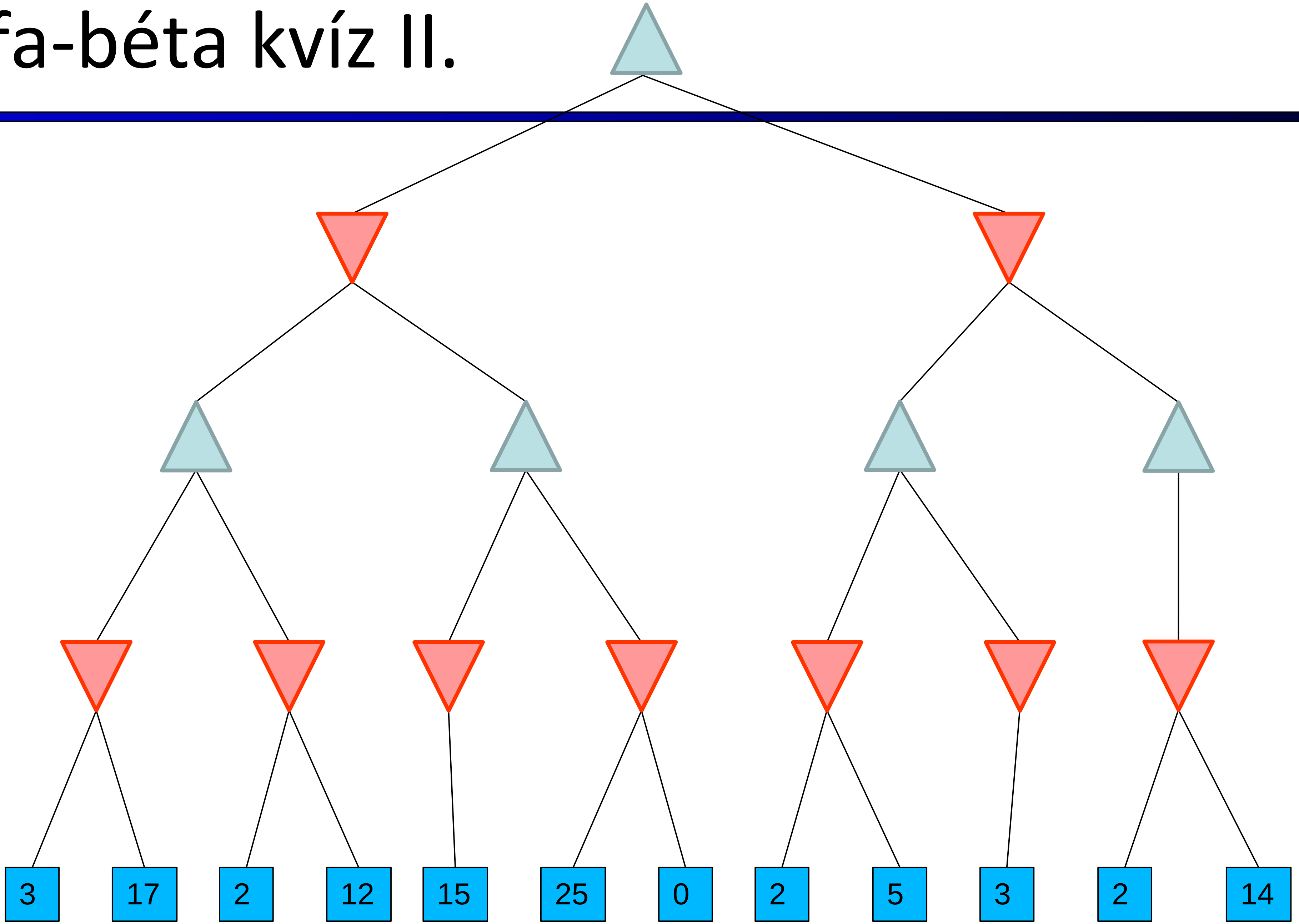
- A nyesés nem befolyásolja a gyökér csomópont minimax értékét!
- Köztes csomópontok értéke eltérhet a valóságtól
- A gyerek csomópontok jó sorrendezése növeli a nyesés hatékonyságát
- „Tökéletes” sorrendezéssel:
  - Az időkomplexitás $O(b^m)$ –ről $O(b^{m/2})$-re csökken
    = az elágazási tényező b-ről $\sqrt{b}$-re csökken
  - Dupla keresési mélység!
  - Sok esetben (pl. sakk) a teljes mélységű keresés továbbra sem kivitelezhető

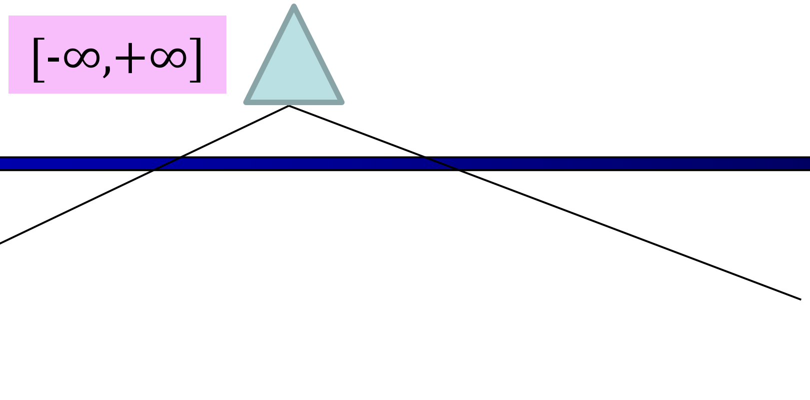


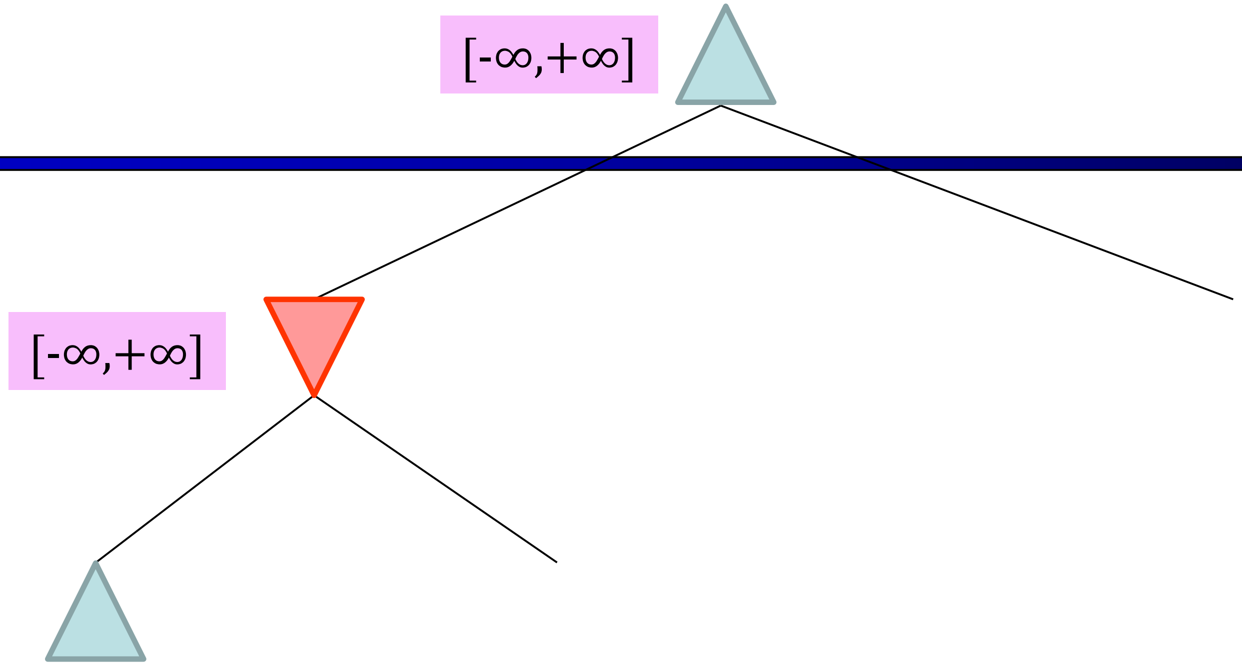
Alfa-béta kvíz I.

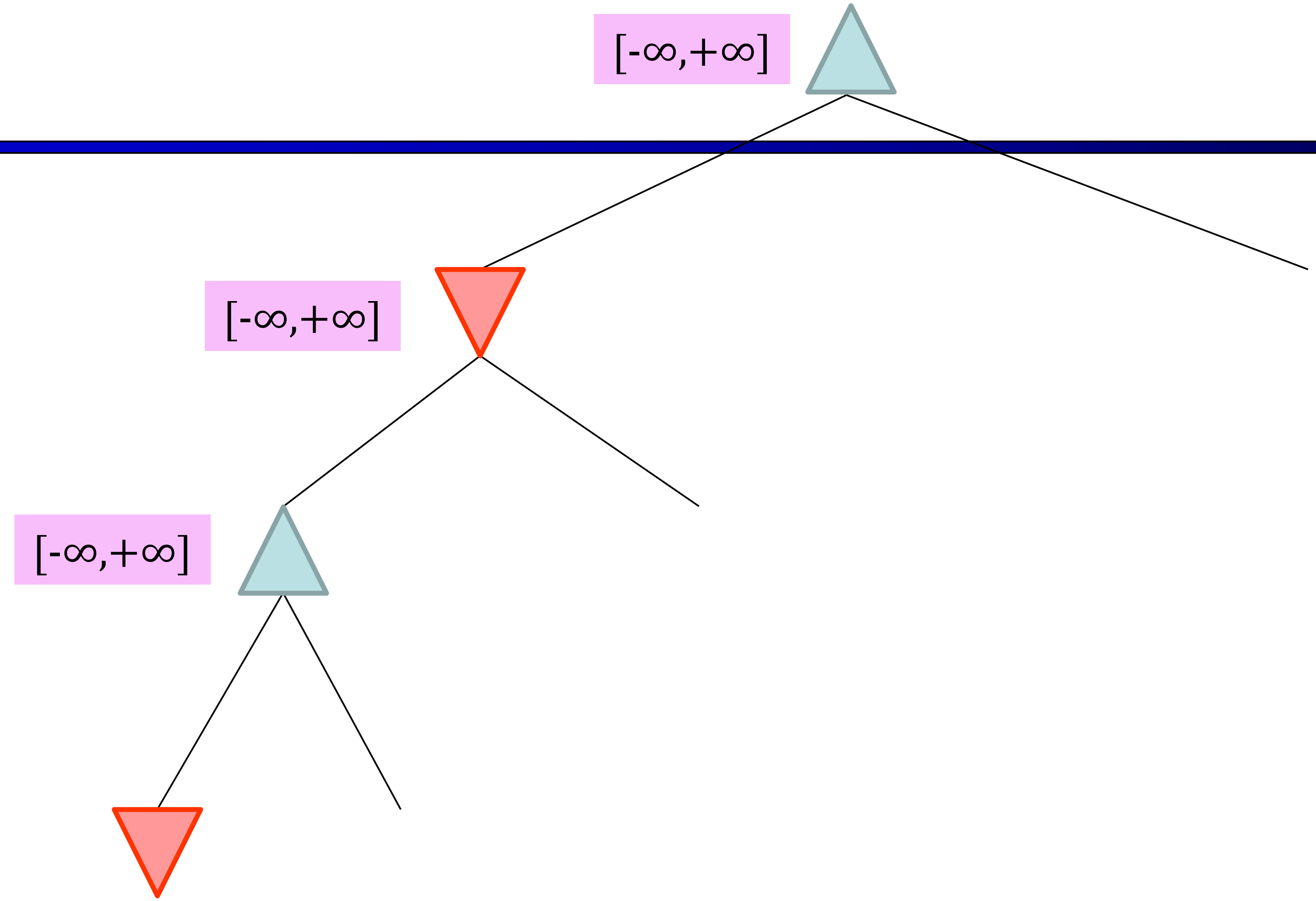


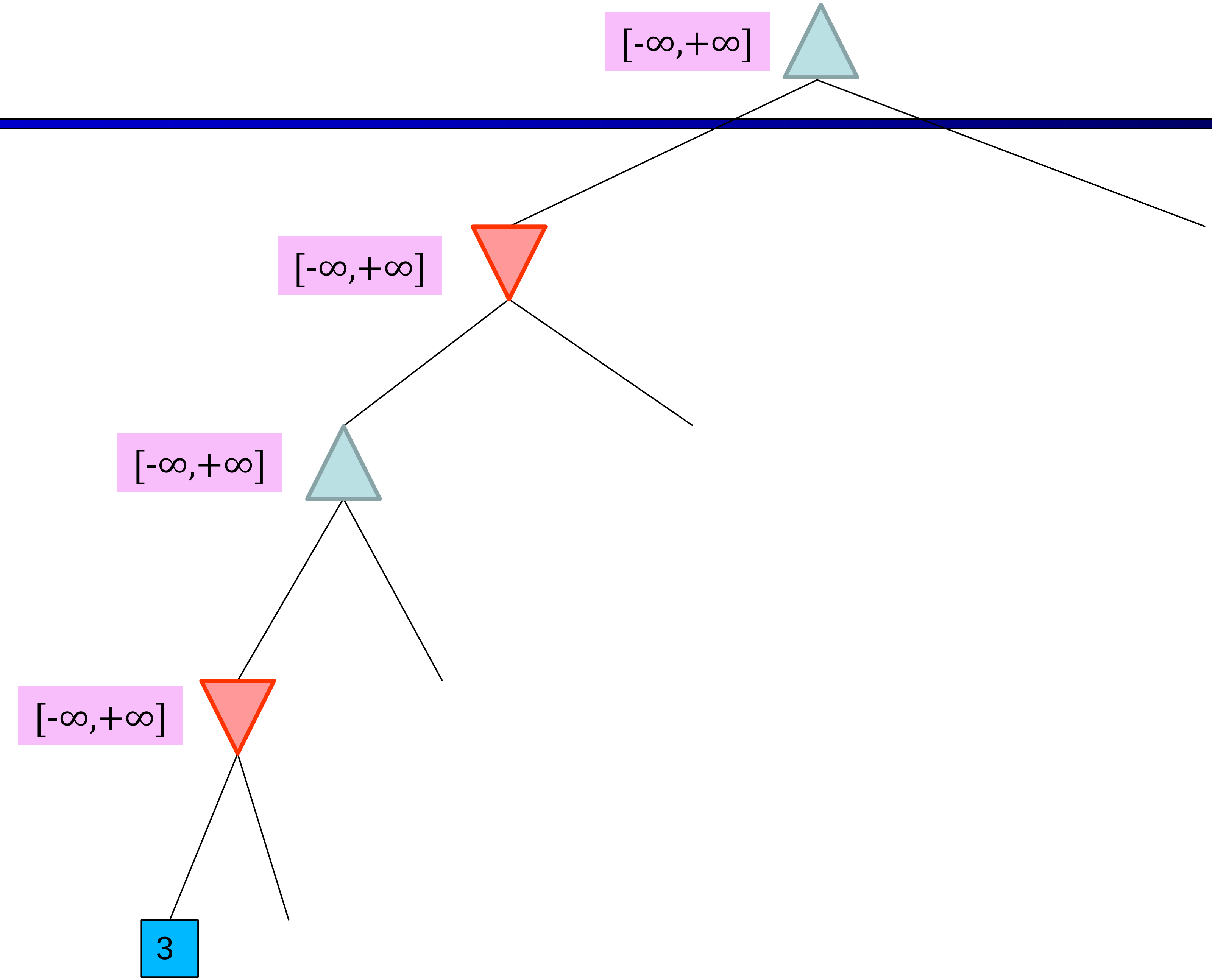
Alfa-béta kvíz II.



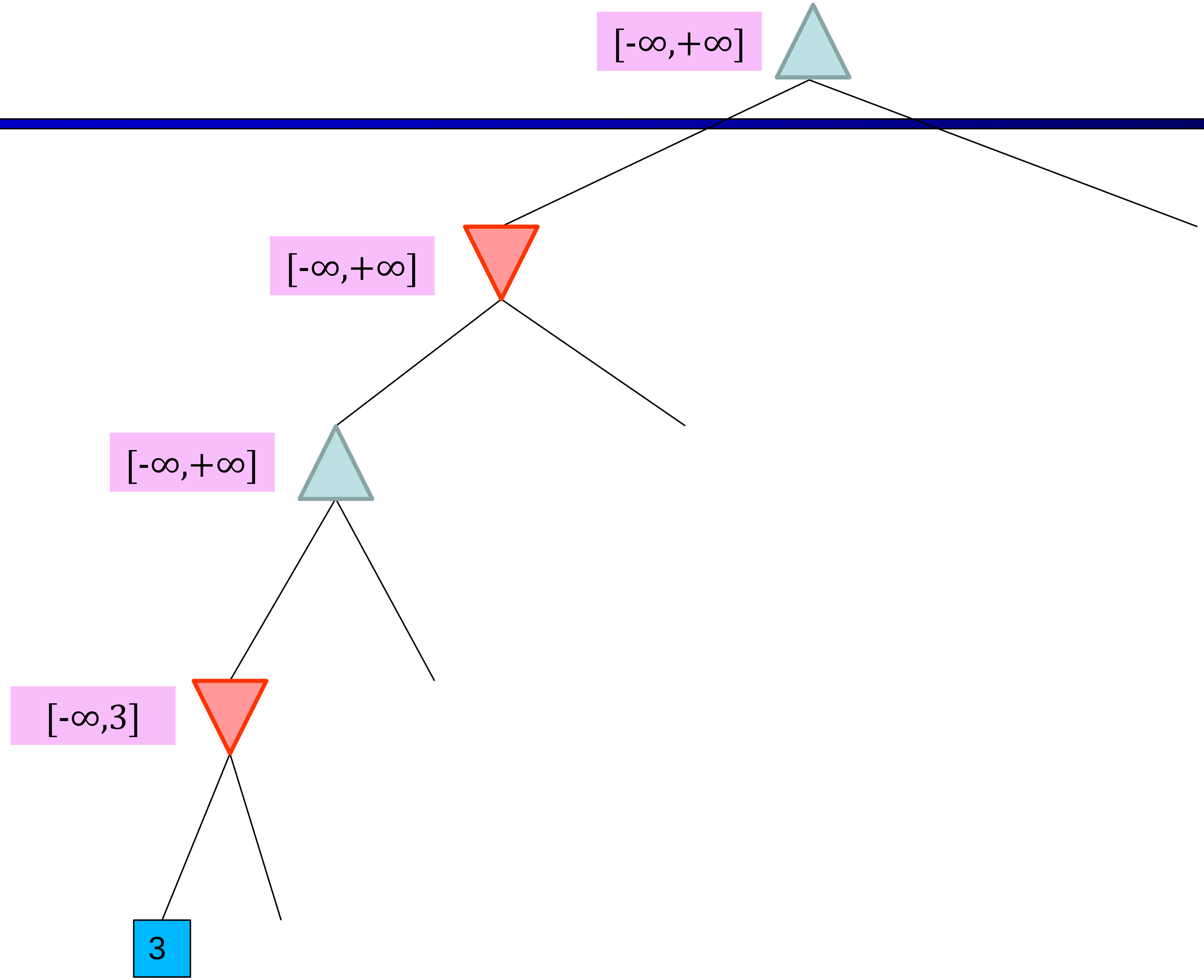


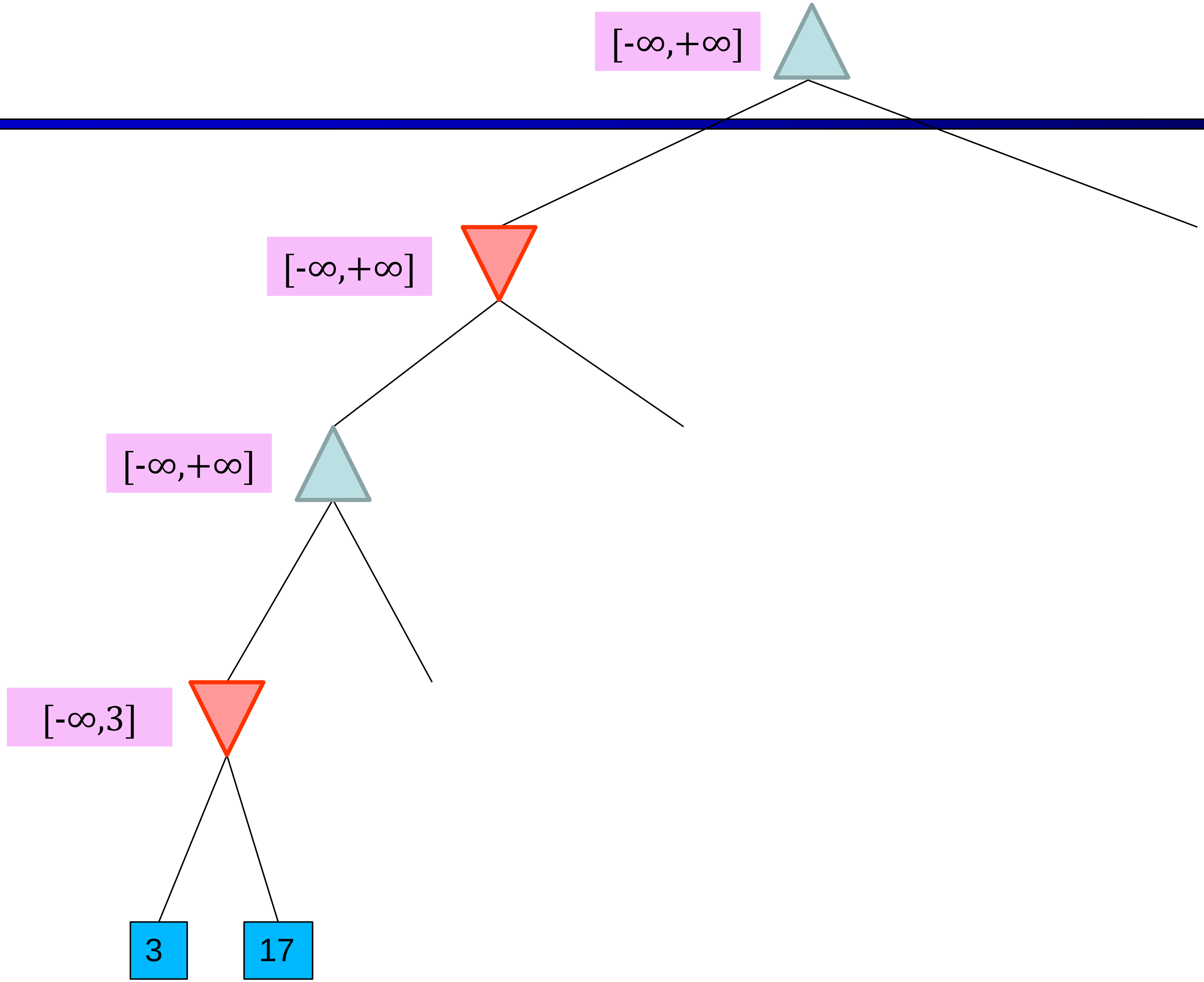


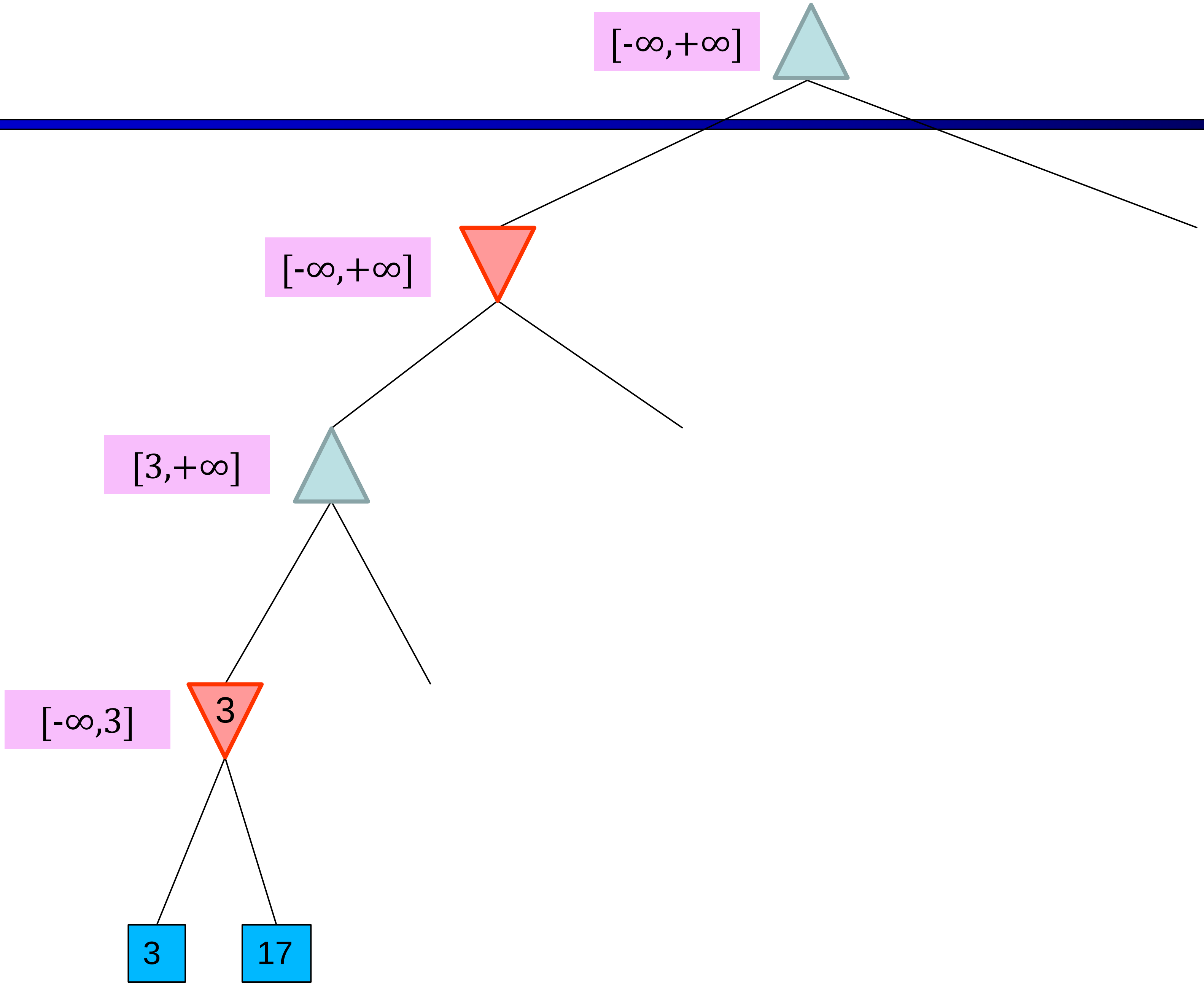




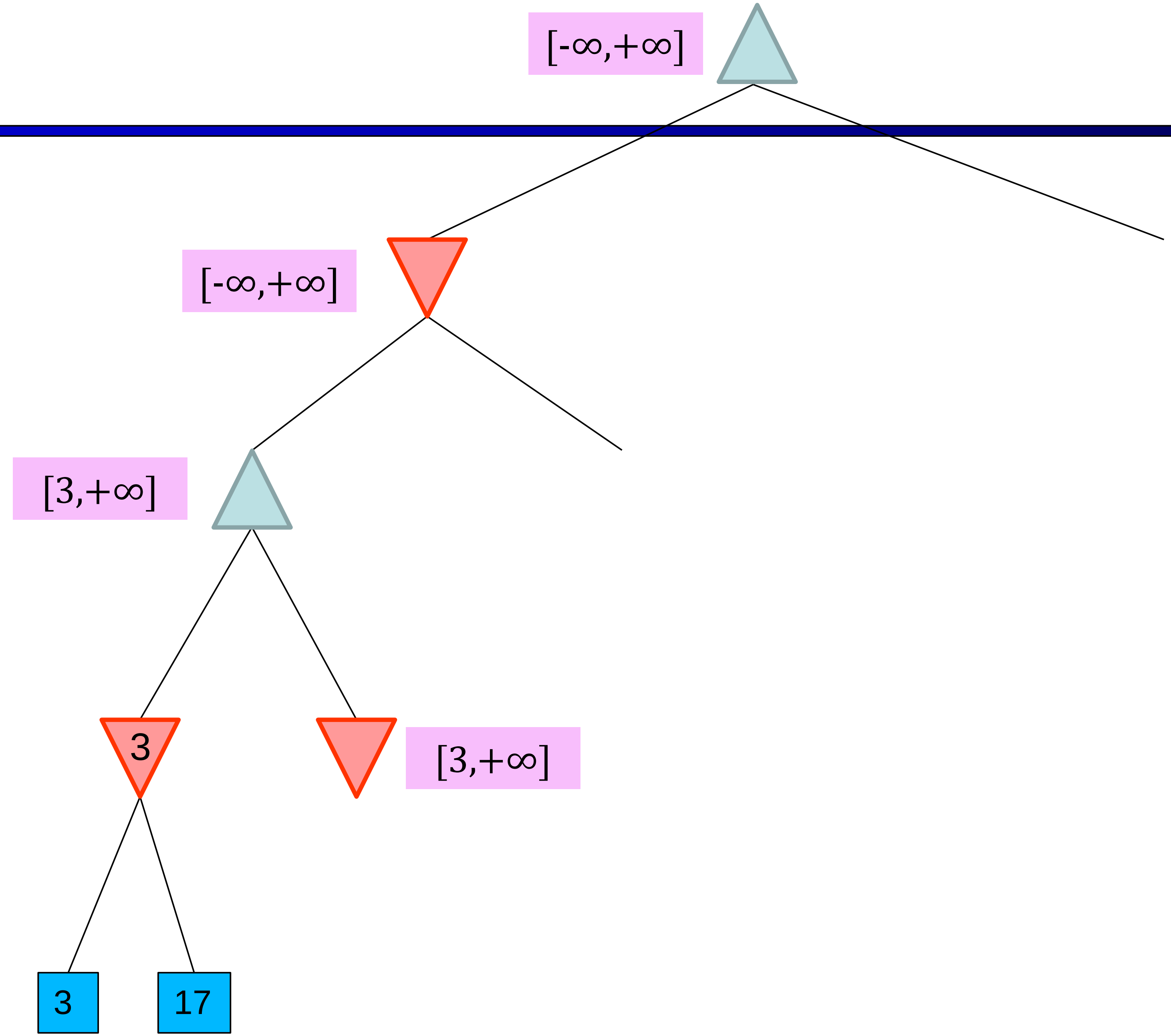
[-∞,+∞]
[-∞,+∞]
[-∞,+∞]
[-∞,+∞]
3


[-∞,+∞]
[-∞,+∞]
[-∞,+∞]
[-∞,3]
3

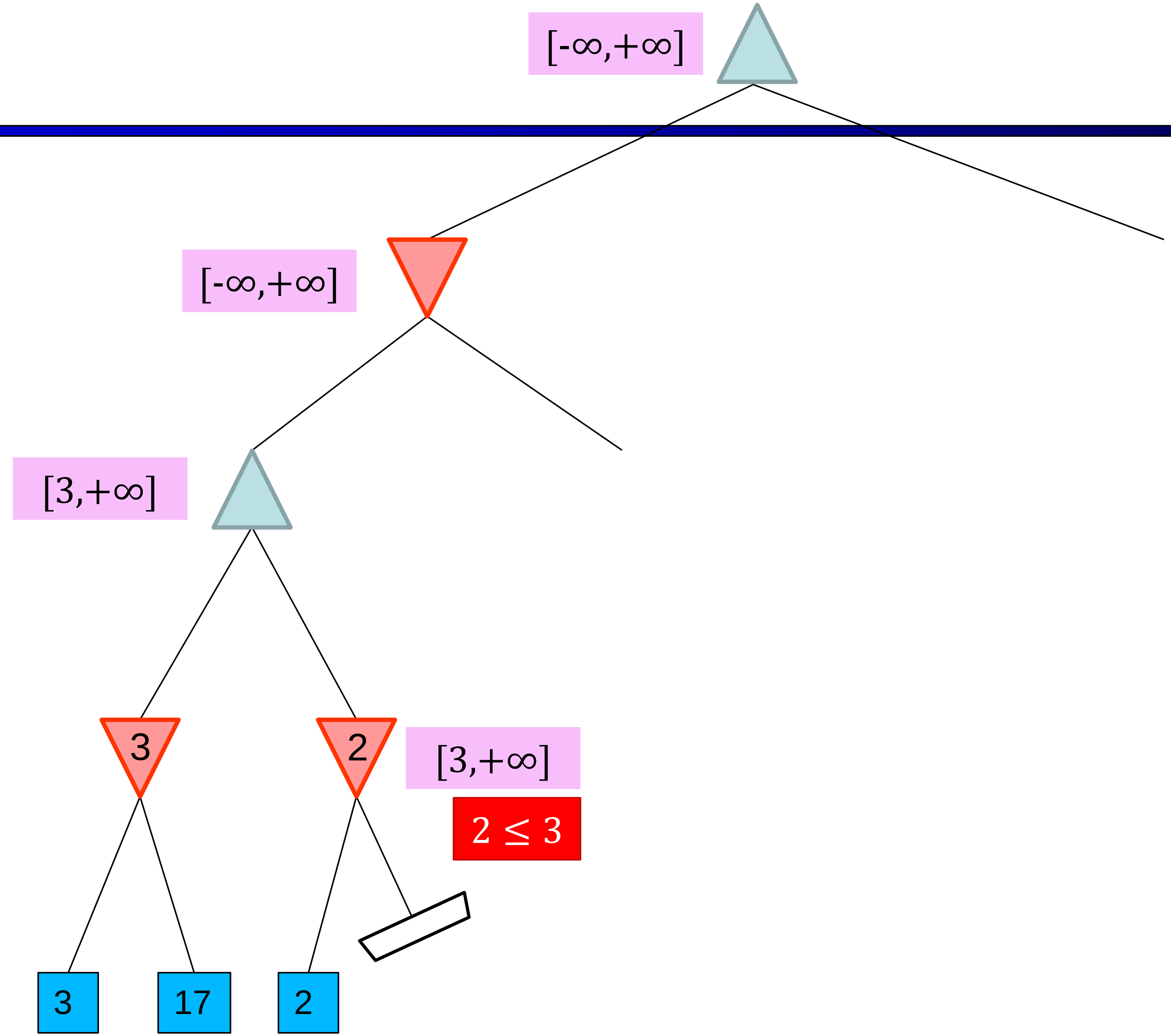

[-∞,+∞]
[-∞,+∞]
[-∞,+∞]
[-∞,3]
3
17



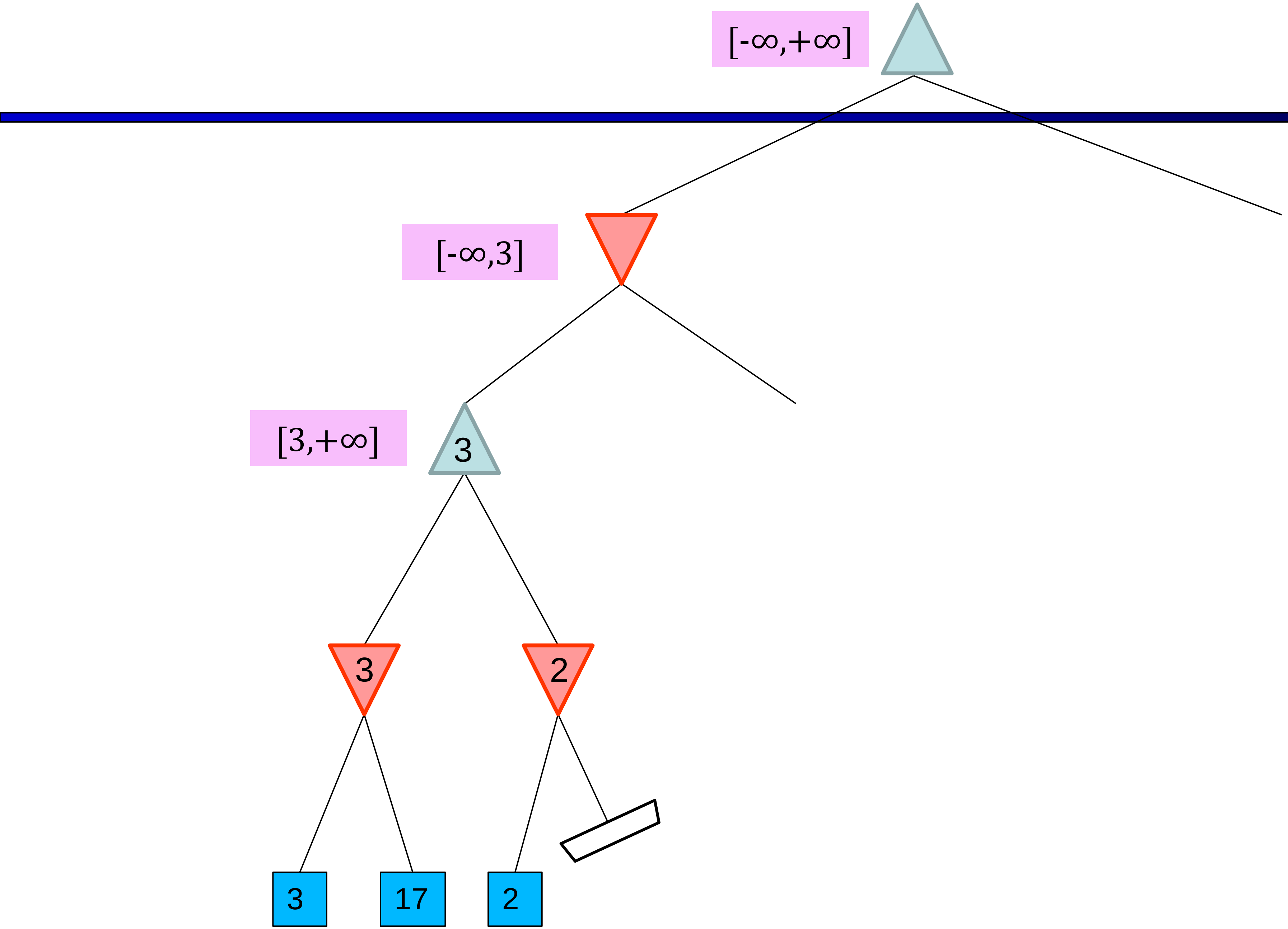
[-∞,+∞]
[-∞,+∞]
[3,+∞]
[-∞,3]
3
3
17



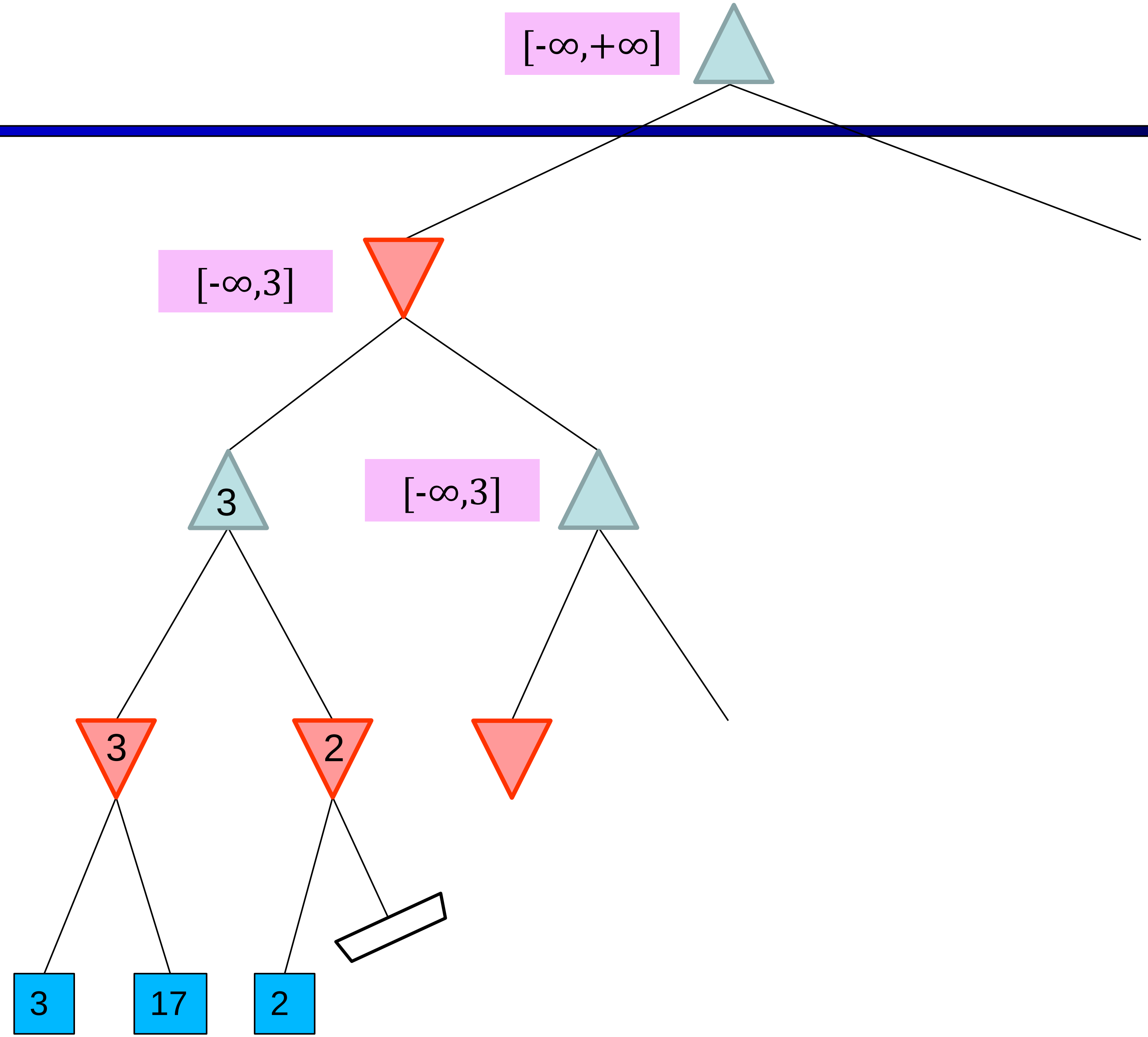
[-∞,+∞]
[-∞,+∞]
[3,+∞]
3
[3,+∞]
3
17



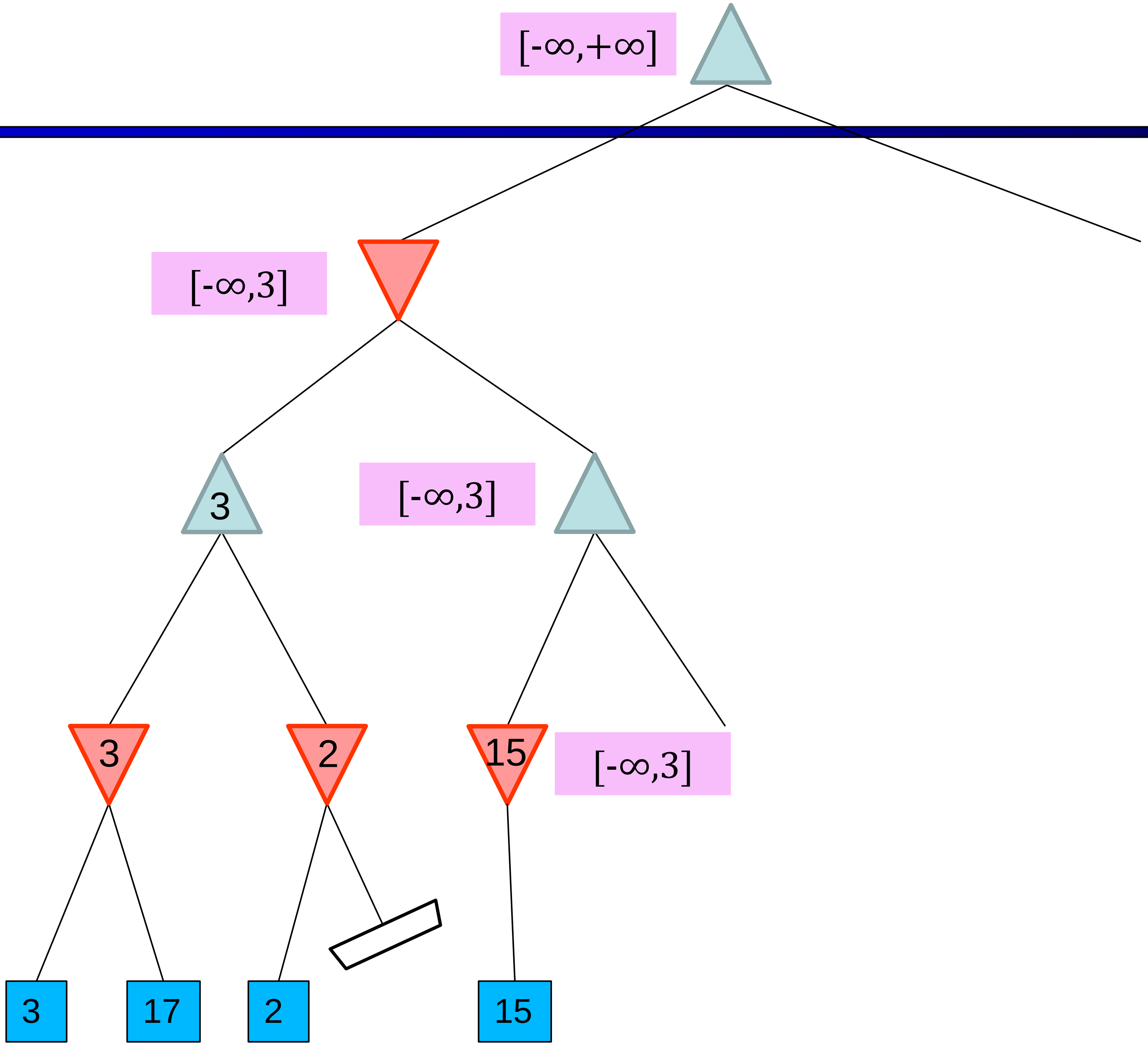
[-∞,+∞]
[-∞,+∞]
[3,+∞]
3
2
[3,+∞]
2 ≤ 3
3
17
2

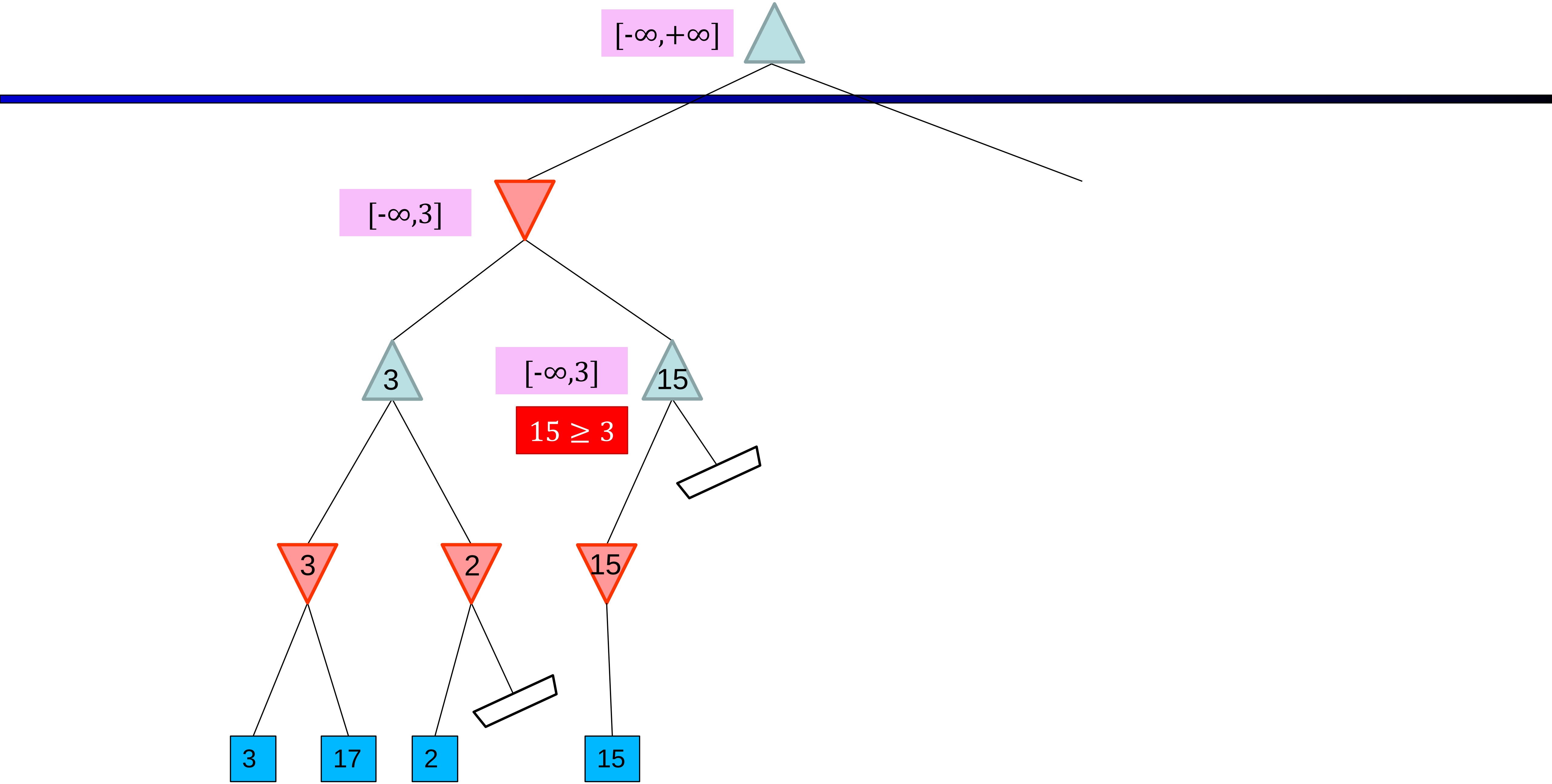


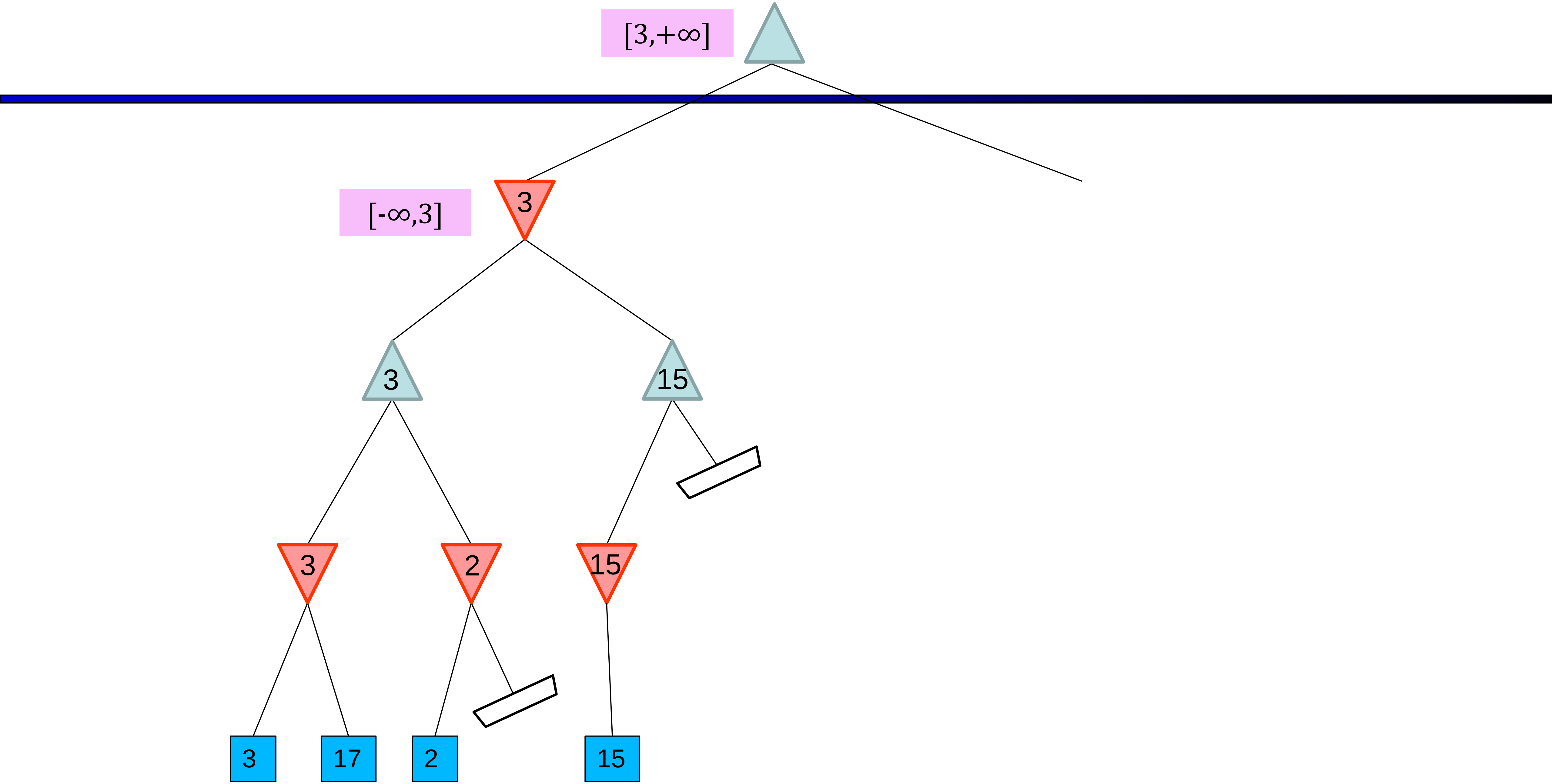
[-∞,+∞]
[-∞,3]
[3,+∞]
3
3
2
3
17
2



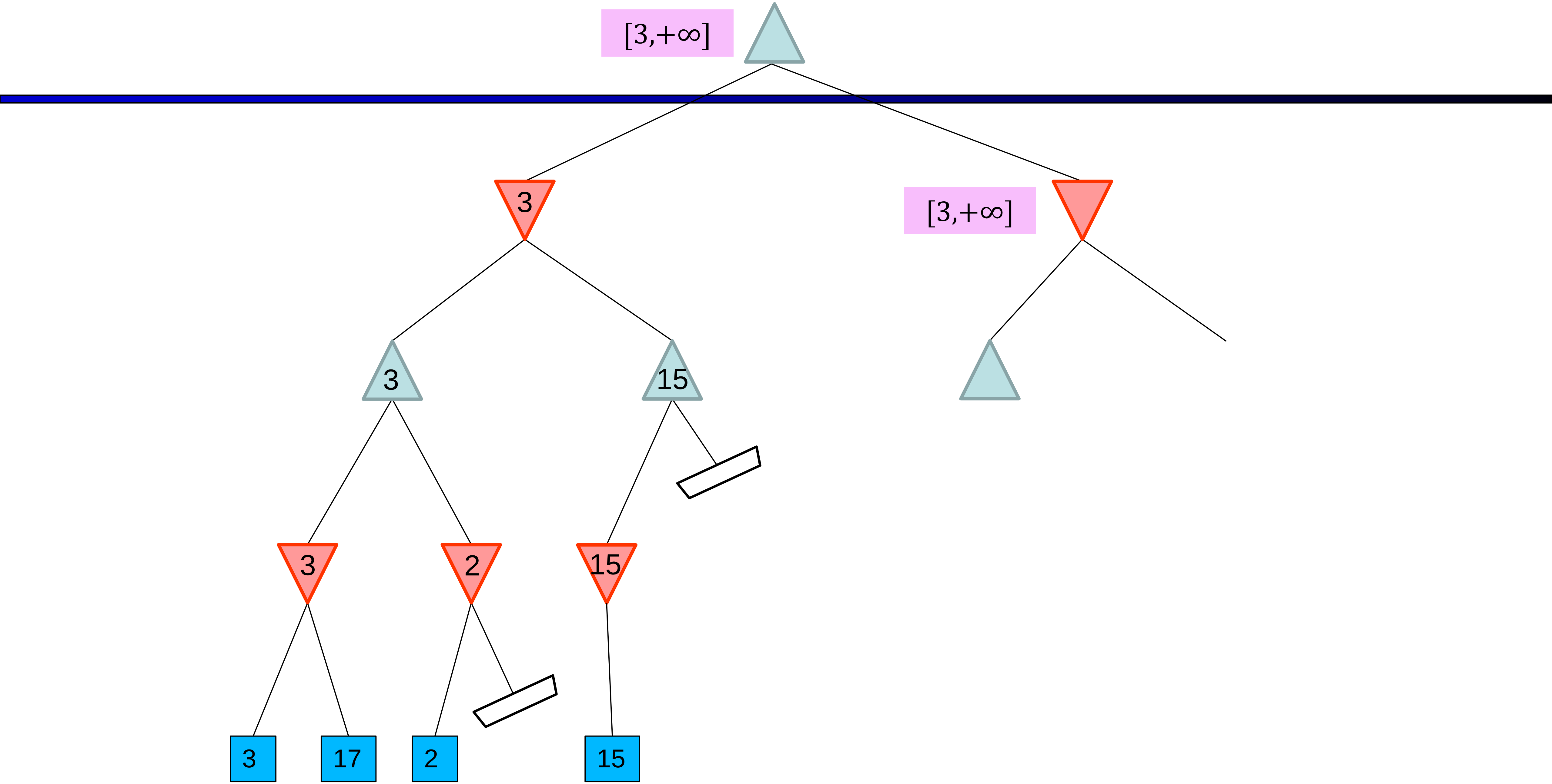
[-∞,+∞]
[-∞,3]
3
[-∞,3]
3
2
3
17
2

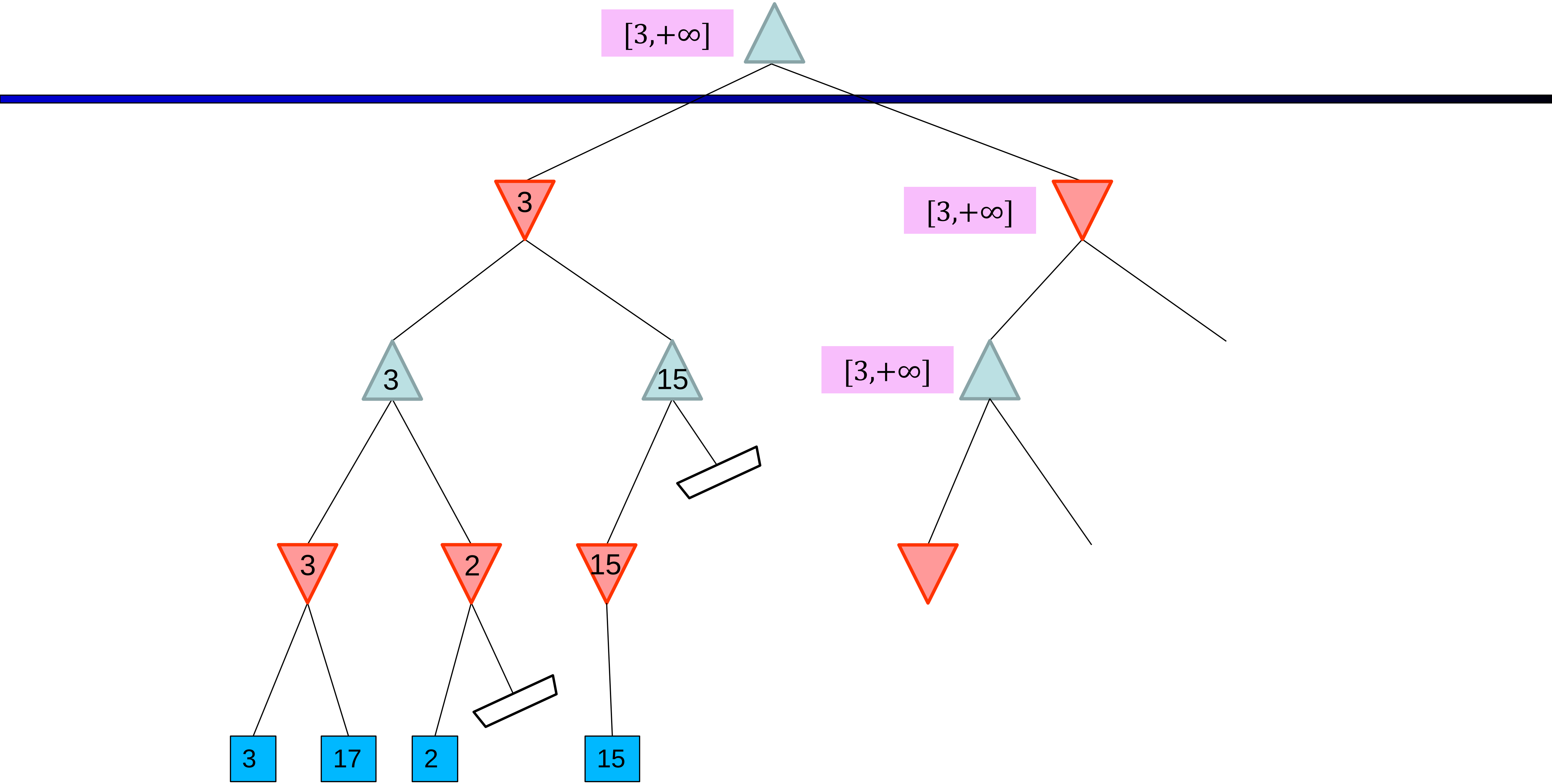

[-∞,+∞]
[-∞,3]
3
[-∞,3]
3
2
15
[-∞,3]
3
17
2
15


[-∞,+∞]
[-∞,3]
3
[-∞,3]
15
15 ≥ 3
3
2
15
3
17
2
15

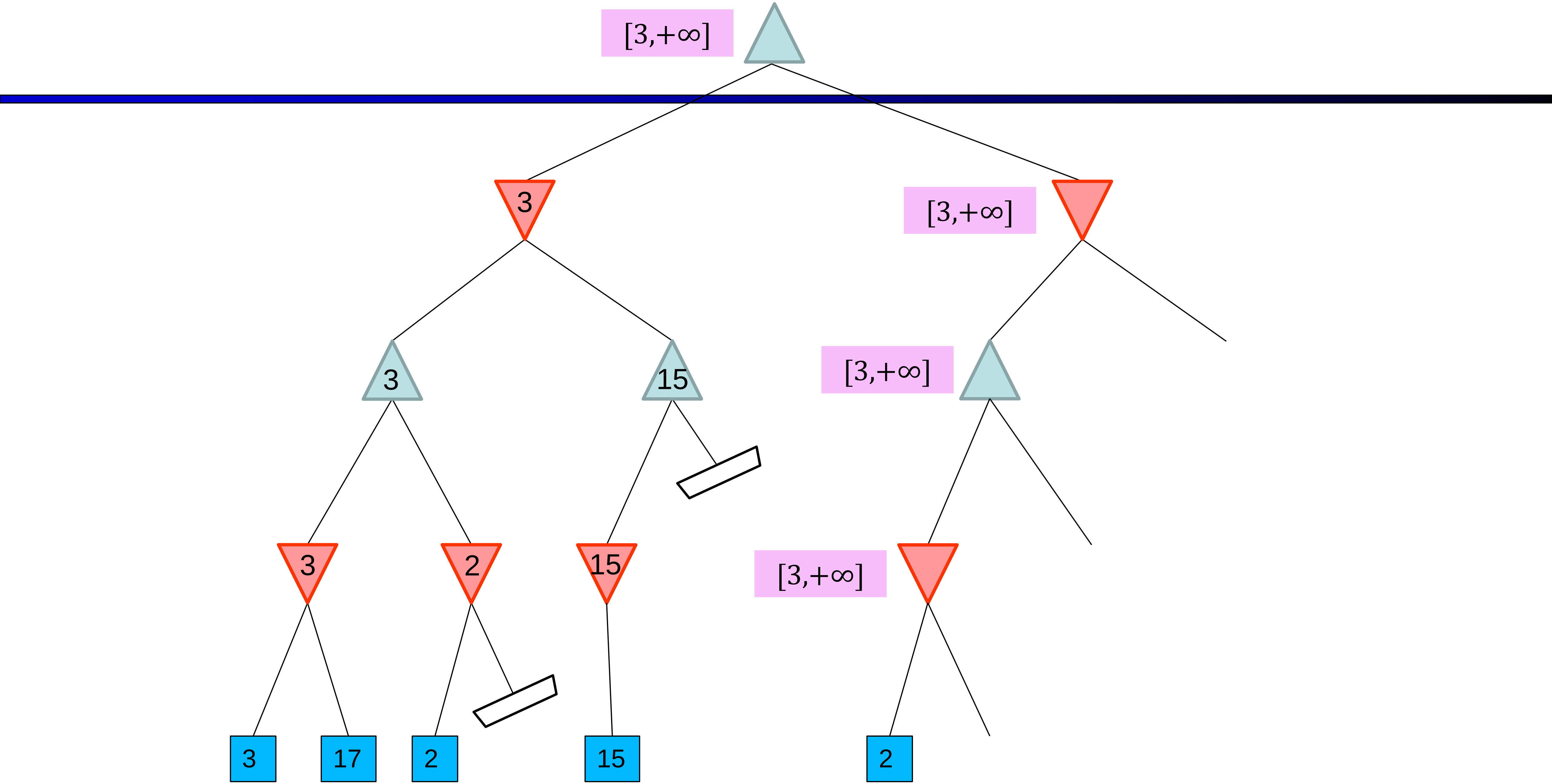


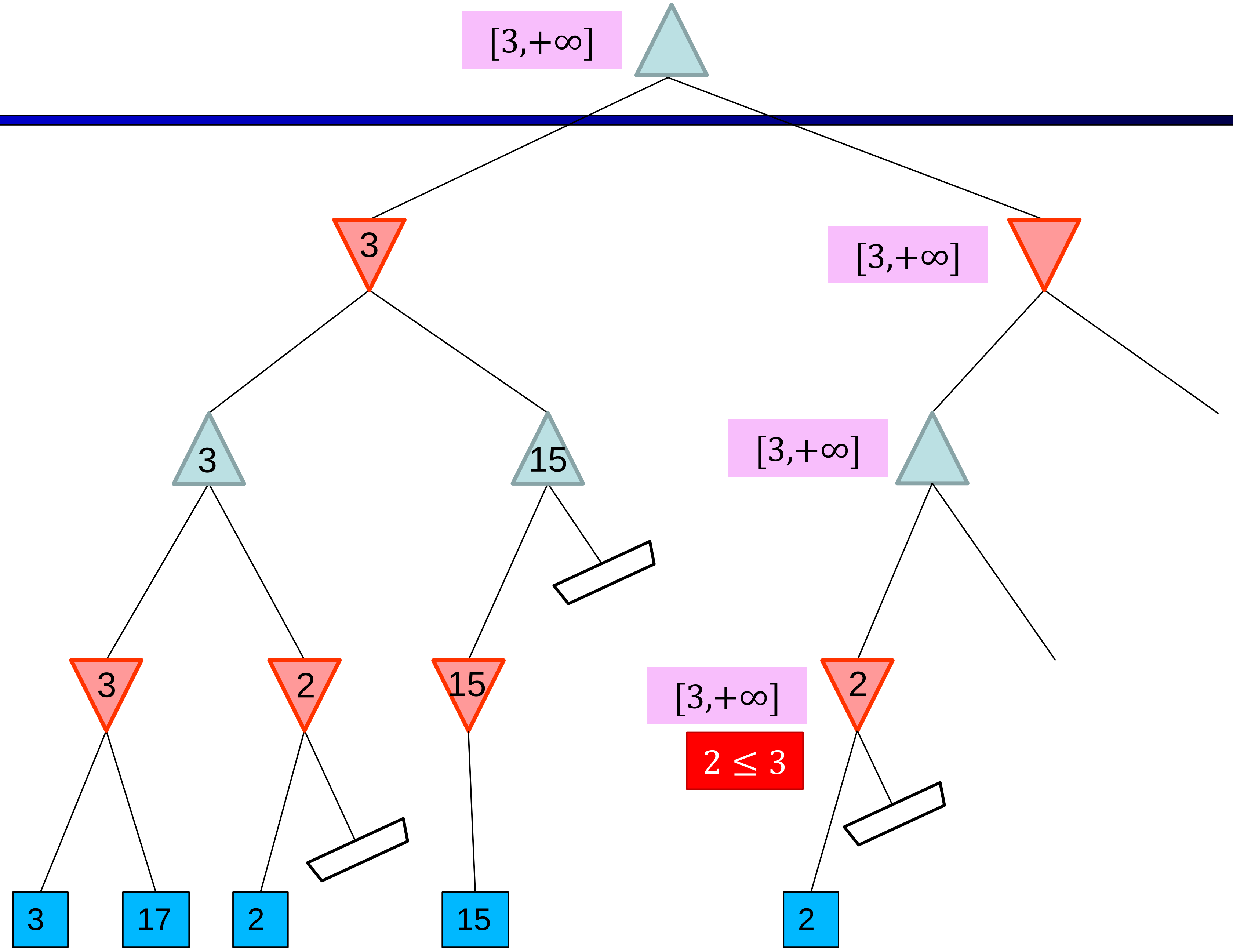
[3,+∞]
[-∞,3]
3
3
15
3
2
15
3
17
2
15


[3,+∞]
3
[3,+∞]
3
15
3
2
15
3
17
2
15

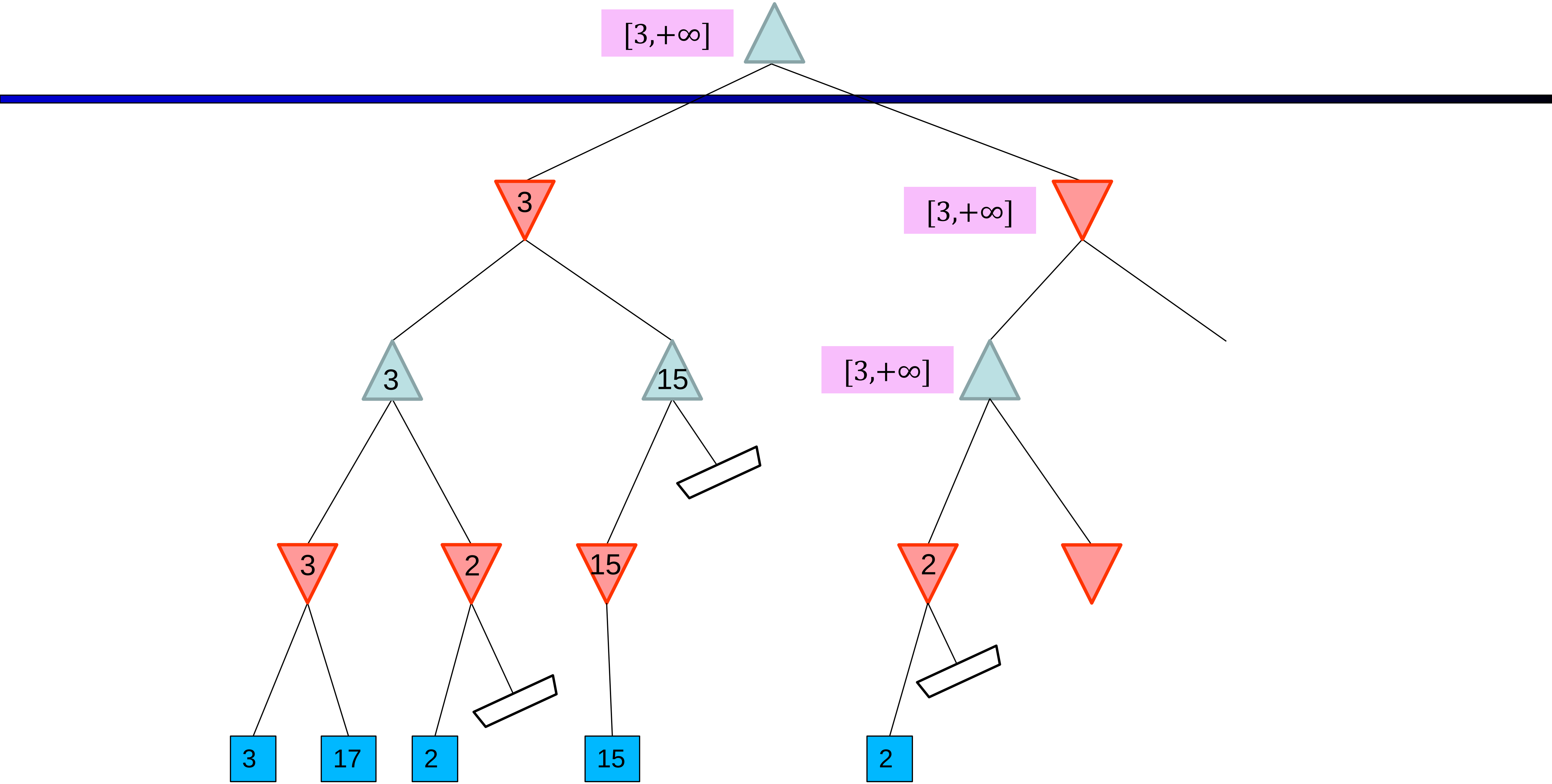


[3,+∞]
3
[3,+∞]
3
15
[3,+∞]
3
2
15
3
17
2
15

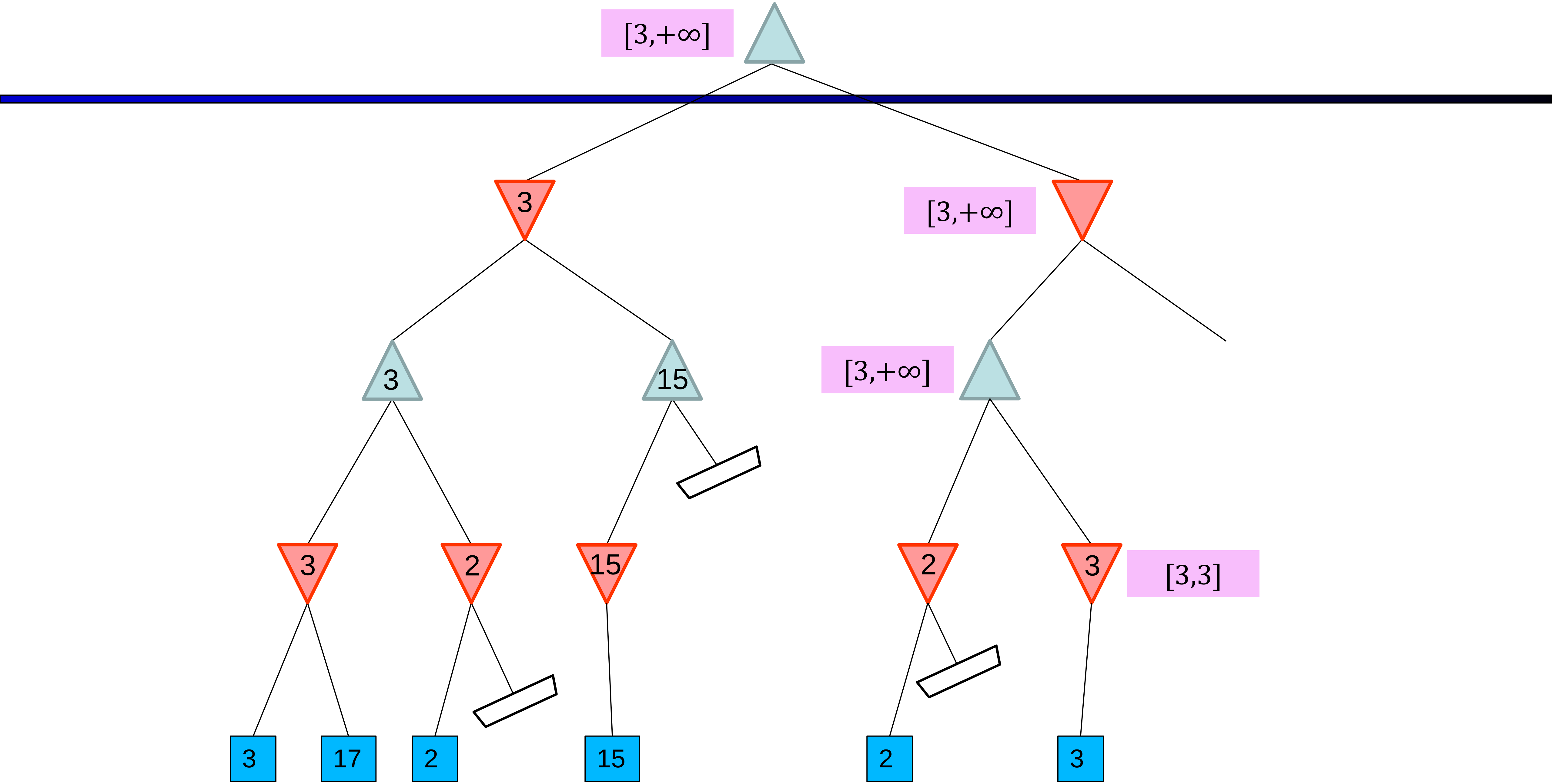

[3,+∞]
3
[3,+∞]
3
15
[3,+∞]
3
2
15
[3,+∞]
3
17
2
15
2



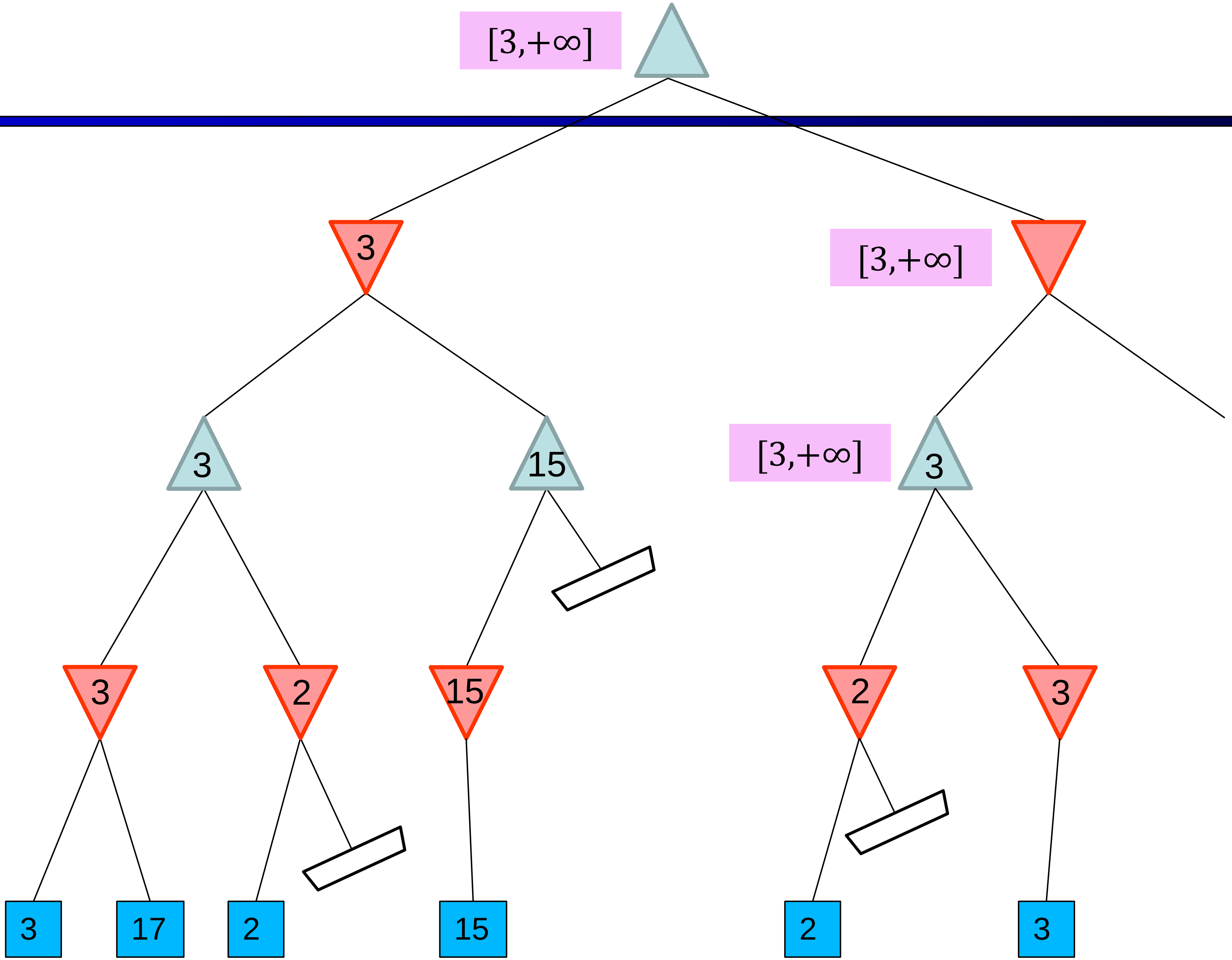
[3,+∞]
3
[3,+∞]
3
15
[3,+∞]
3
2
15
[3,+∞]
2
2 ≤ 3
3
17
2
15
2



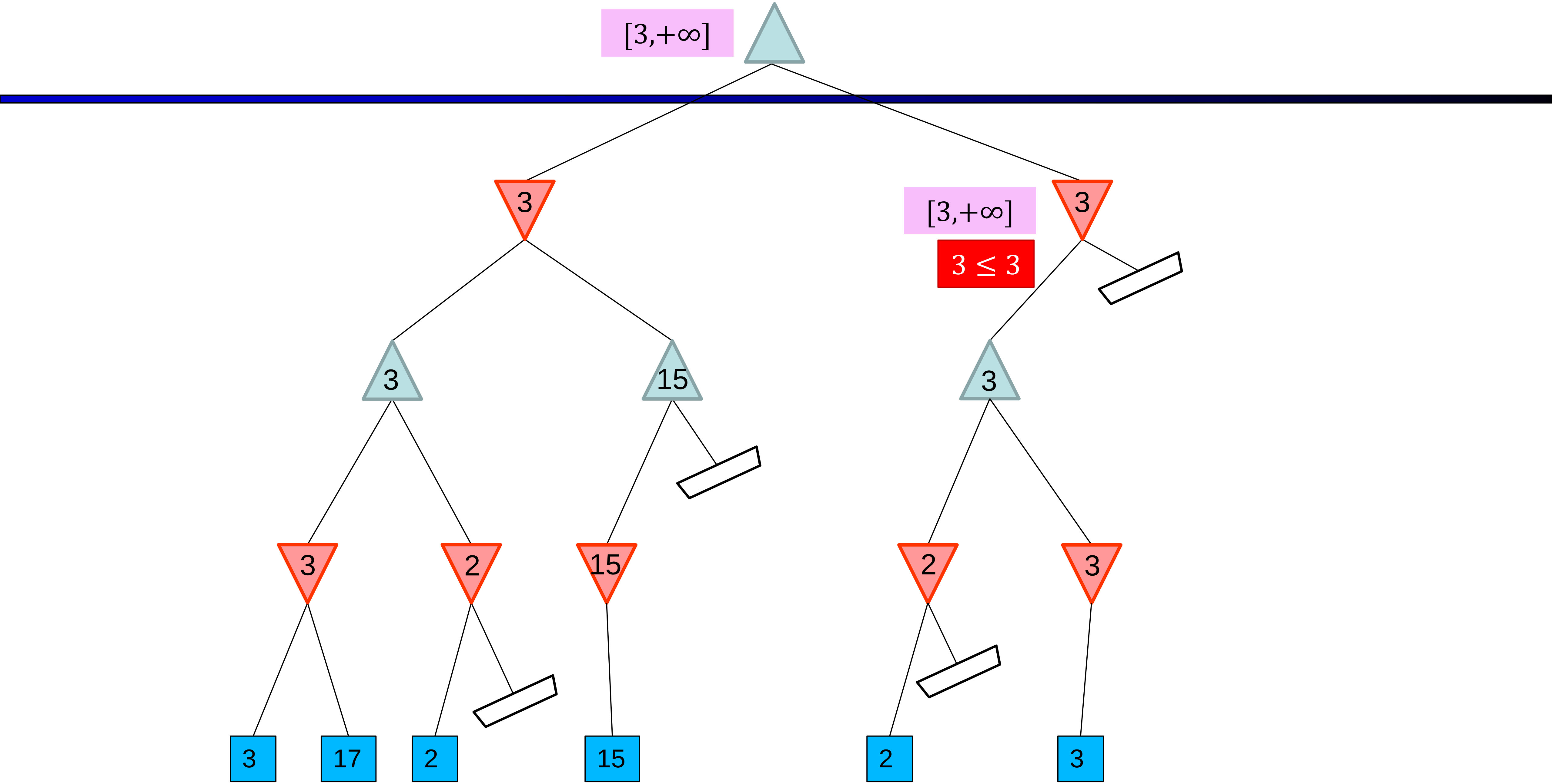
[3,+∞]
3
[3,+∞]
3
15
[3,+∞]
3
2
15
2
3
17
2
15
2

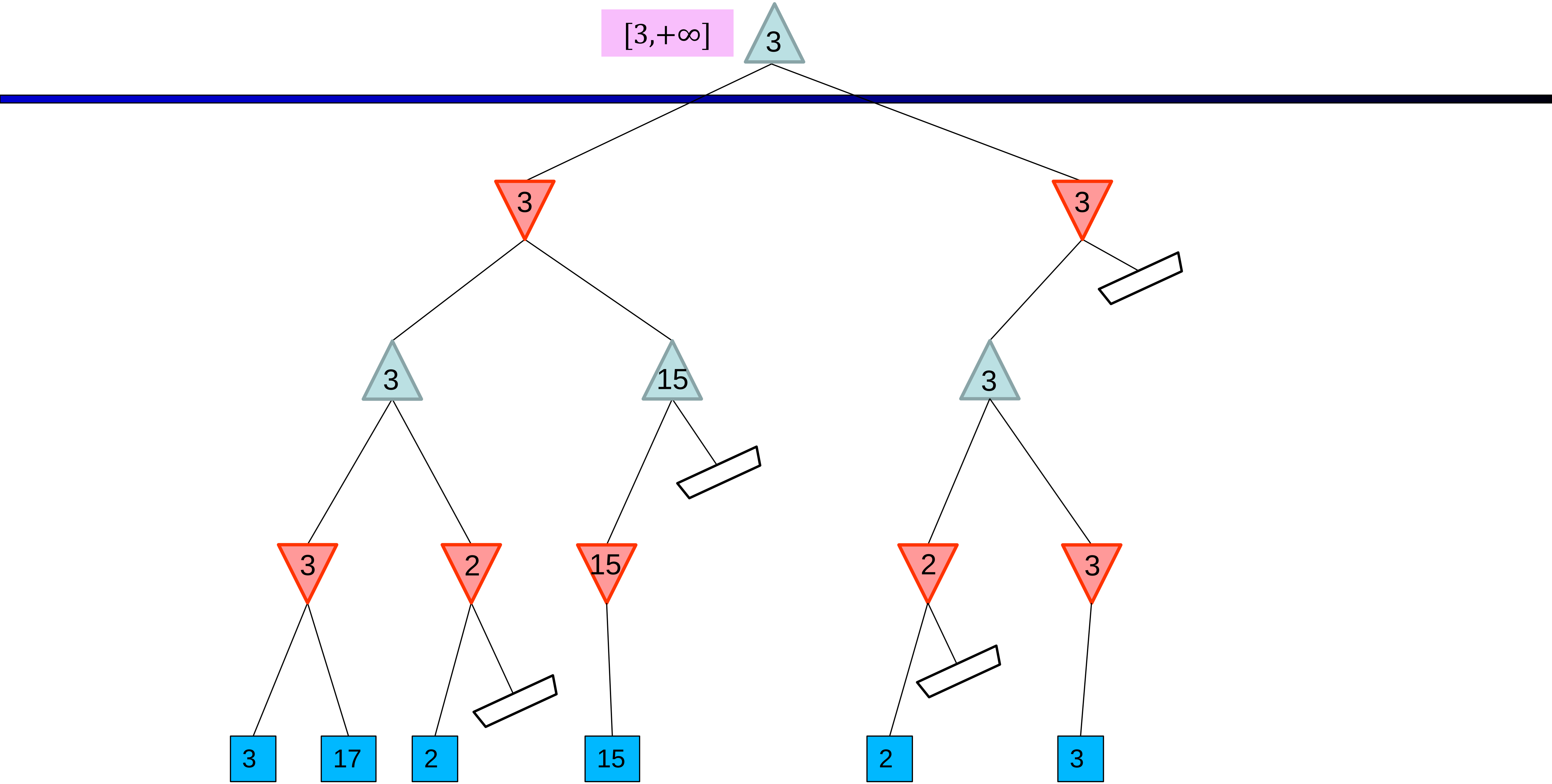


[3,+∞]
3
[3,+∞]
3
15
[3,+∞]
3
2
15
2
3
[3,3]
3
17
2
15
2
3



[3,+∞]
3
[3,+∞]
3
15
[3,+∞]
3
3
2
15
2
3
3
17
2
15
2
3

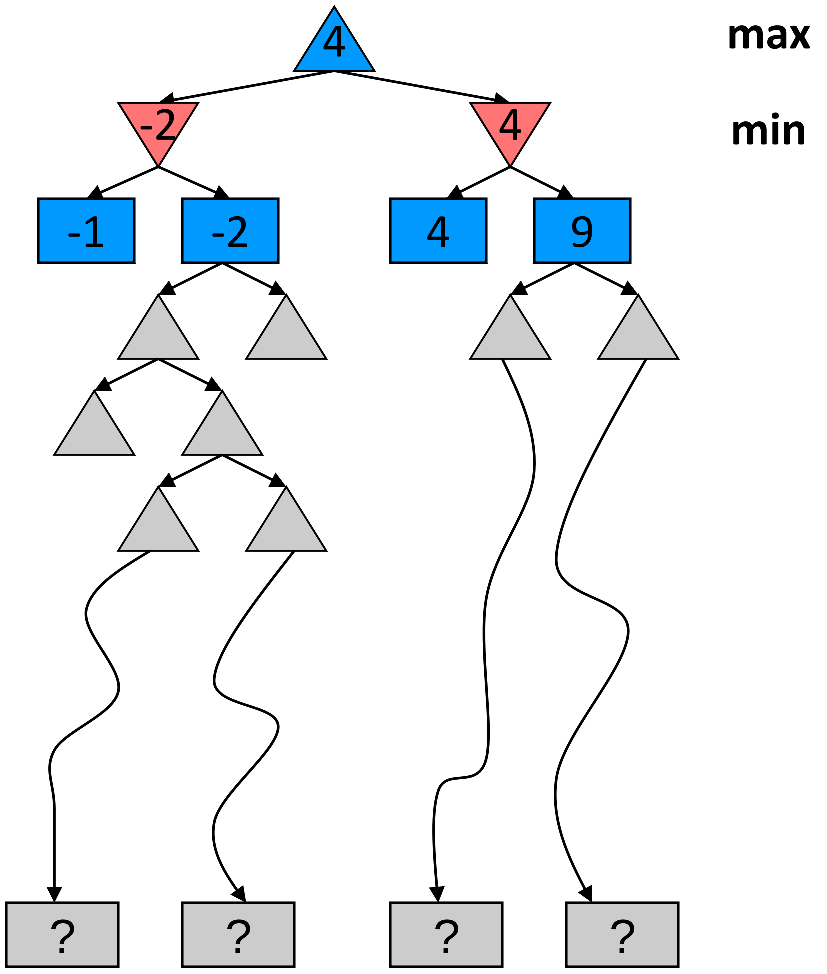

[3,+∞]
3
[3,+∞]
3
3 ≤ 3
3
15
3
3
2
15
2
3
3
17
2
15
2
3


[3,+∞]
3
3
3
3
15
3
3
2
15
2
3
3
17
2
15
2
3

Erőforrások limitációja

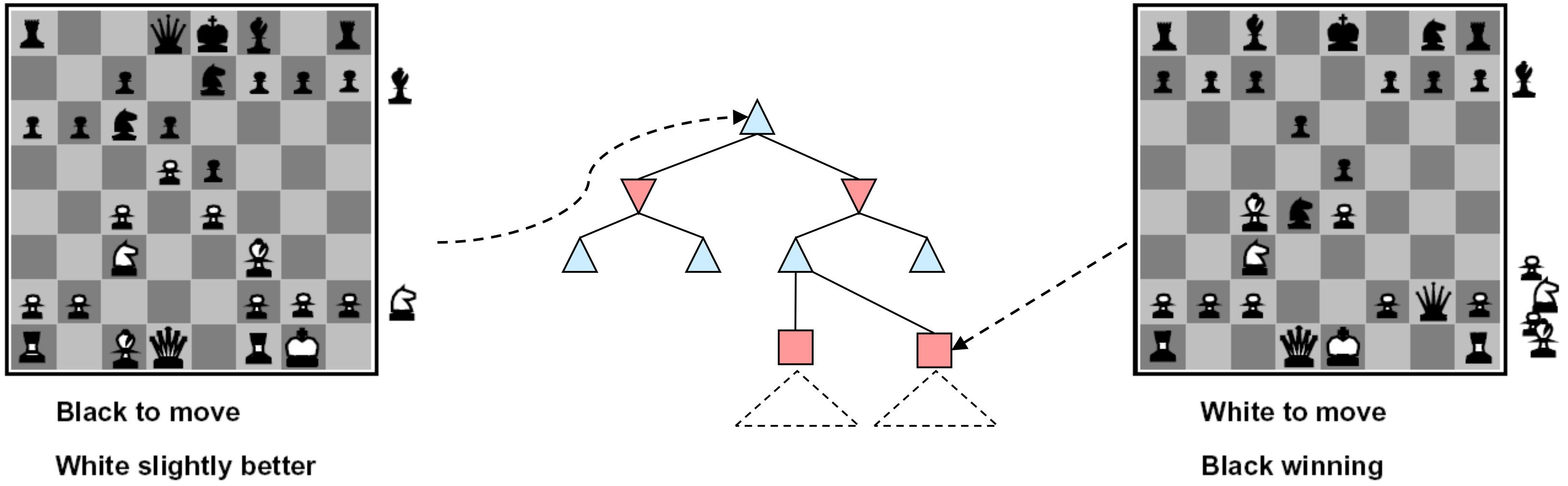
Erőforrások limitációja

- Probléma: a relaisztikus játékokban a keresés nem tud eljutni a levélcsomópontokig!
- Megoldás: Mélységkorlátozott keresés
  - A teljes mélység helyett csak egy korlátos mélységig menjünk le a fában
  - A végállapotok hasznosságát cseréljük le egy kiértékelő függvénnyel (állapot -> szám érték)
- Példa (sakk):
  - Tegyük fel, hogy 100 másodpercünk van minden lépésnél, és 10 ezer csomópontot tudunk megvizsgálni másodpercenként
  - Tehát lépésenként 1 millió állapotot tudunk megvizsgálni
  - α-β nyesés esetén az elérhető mélység: 8 – elég jó sakkprogram
- Az optimális játék már nem garantált
- Ha bármikor le kell tudni állítani: iteratívan mélyülő keresés



Kiértékelő függvények

- A kiértékelő függvények nem végállapot csomópontokat értékelnek ki



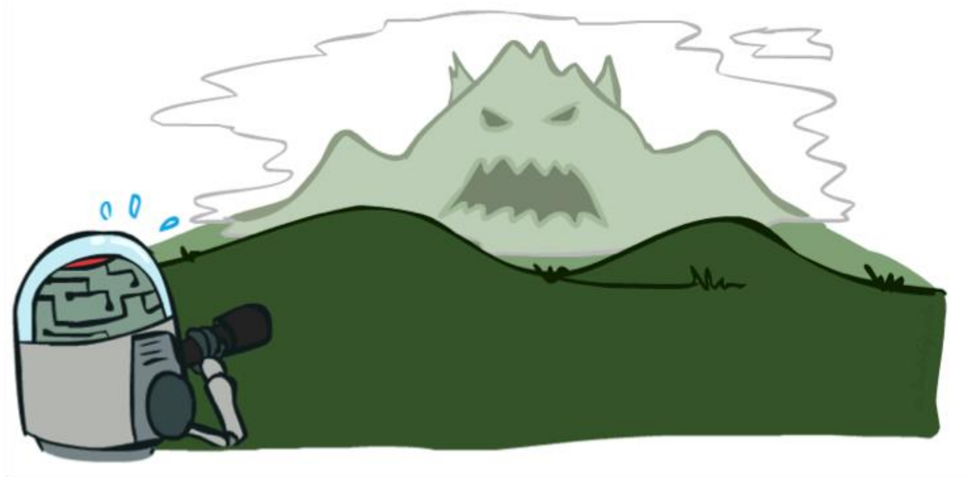
- Ideális függvény: az adott állapot aktuális minimax értékét adja vissza
- A gyakorlatban: tipikusan jegyek súlyozott lineáris kombinációja

$$Eval(s) = w_1 f_1(s) + w_2 f_2(s) + \ldots + w_n f_n(s)$$

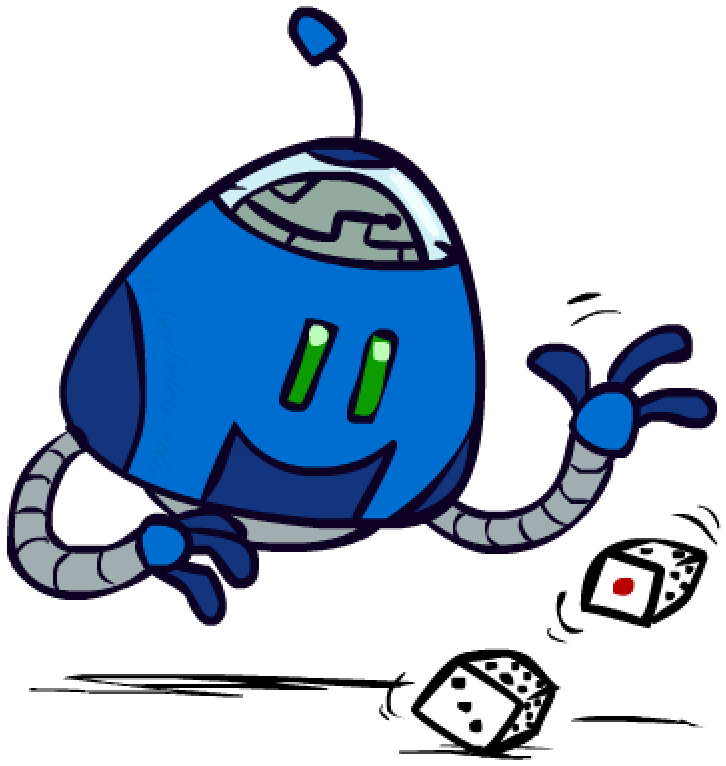
- pl.: $f_1(s)$ = (világos vezérek száma – sötét vezérek száma), stb.

A mélység számít!

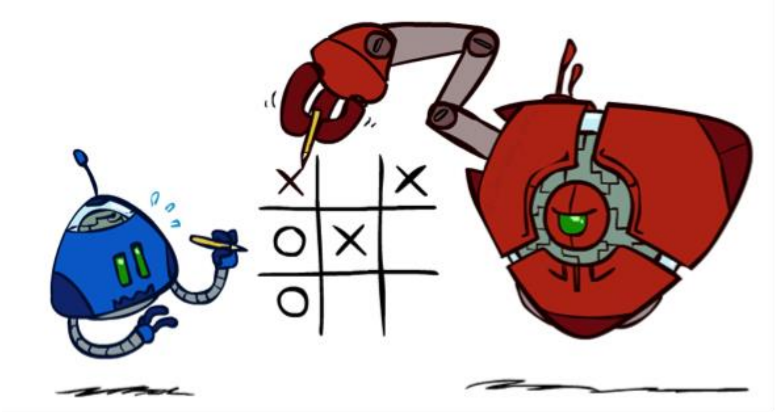
- A kiértékelő függvények mindig pontatlanok
- Minél mélyebben van a kiértékelő függvény, annál kevésbé számít a pontossága
- A kiértékelő függvény komplexitása vs. a keresés (számítás) komplexitása

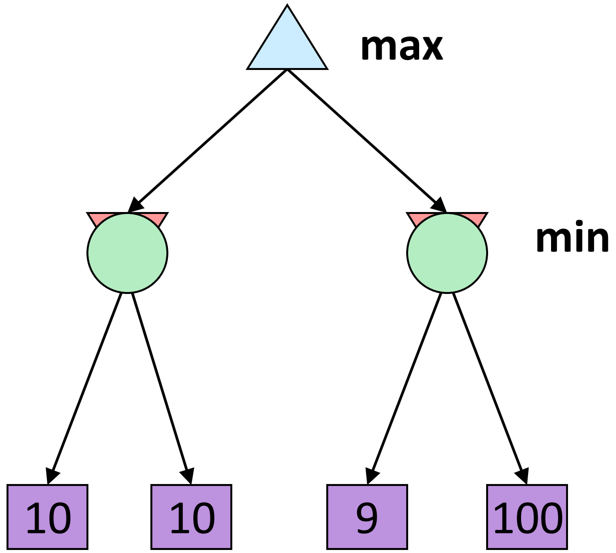


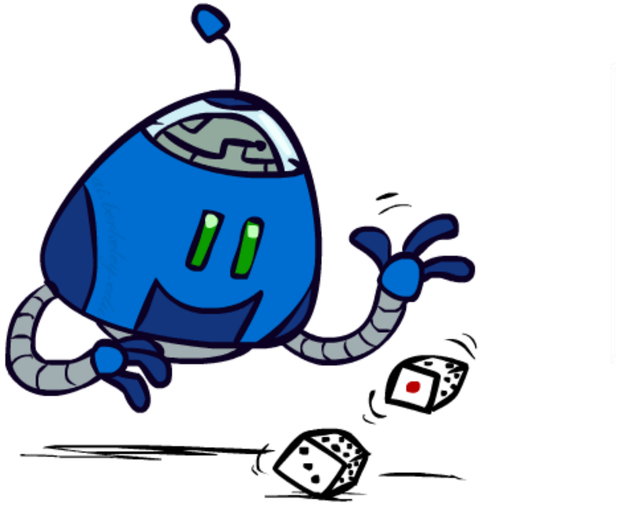
Bizonytalan kimenetelek



Legrosszabb eset vs. átlagos eset





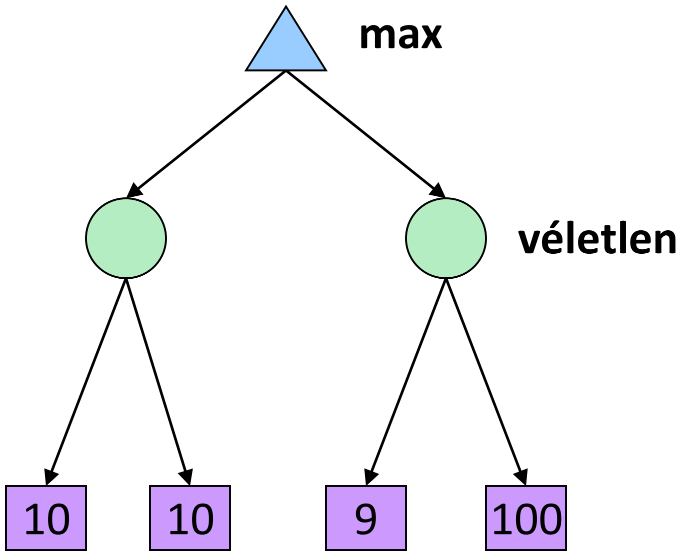


Ötlet: Bizonytalan kimeneteleket a véletlen irányítja, nem egy ellenfél!

Expectimax Keresés

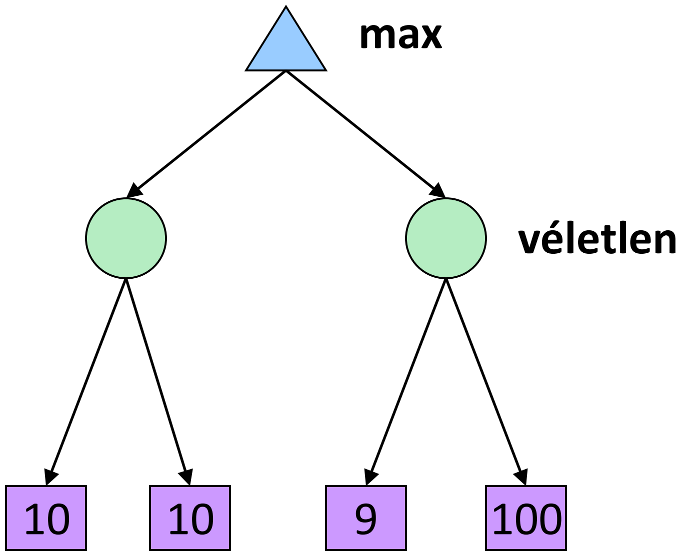
- Miért nem tudjuk (mindig), hogy egy cselekvésnek mi a kimenetele?
  - Explicit véletlen: kockadobás
  - Véletlenszerűen viselkedő ellenfelek: PacMan szellemek véletlen mozgása
  - Cselekvés nem sikerül: robot mozgásánál megcsúszik a kerék

- Az értékeknek az átlagos kimenetelt kellene tükrözniük (expectimax), nem pedig a lehető legrosszabb (minimax) kimenetelt



Expectimax Keresés

- Expectimax keresés: számítsuk ki az átlagos pontszámot (hasznot) optimális játékot feltételezve
  - **Max** csomópontok, mint minimax keresés esetében
  - Véletlenszerű csomópontok, mint **Min** csomópontok, de a kimenetel bizonytalan
  - Számítsuk ki a várható hasznosságot
  - Azaz a gyermekcsomópontok súlyozott átlagát (várhatóértékét)



Expectimax Pszeudokód

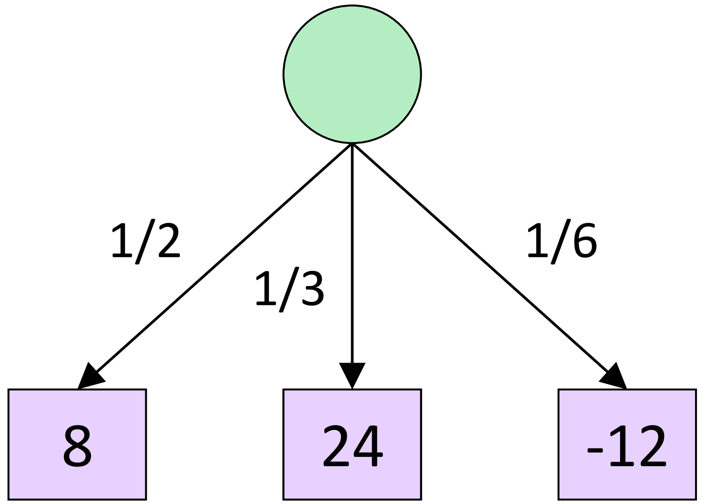
```
def value(state):
    if the state is a terminal state: return the state's utility
    if the next agent is MAX: return max-value(state)
    if the next agent is EXP: return exp-value(state)
```

```
def max-value(state):
    initialize v = -∞
    for each successor of state:
        v = max(v, value(successor))
    return v
```

```
def exp-value(state):
    initialize v = 0
    for each successor of state:
        p = probability(successor)
        v += p * value(successor)
    return v
```

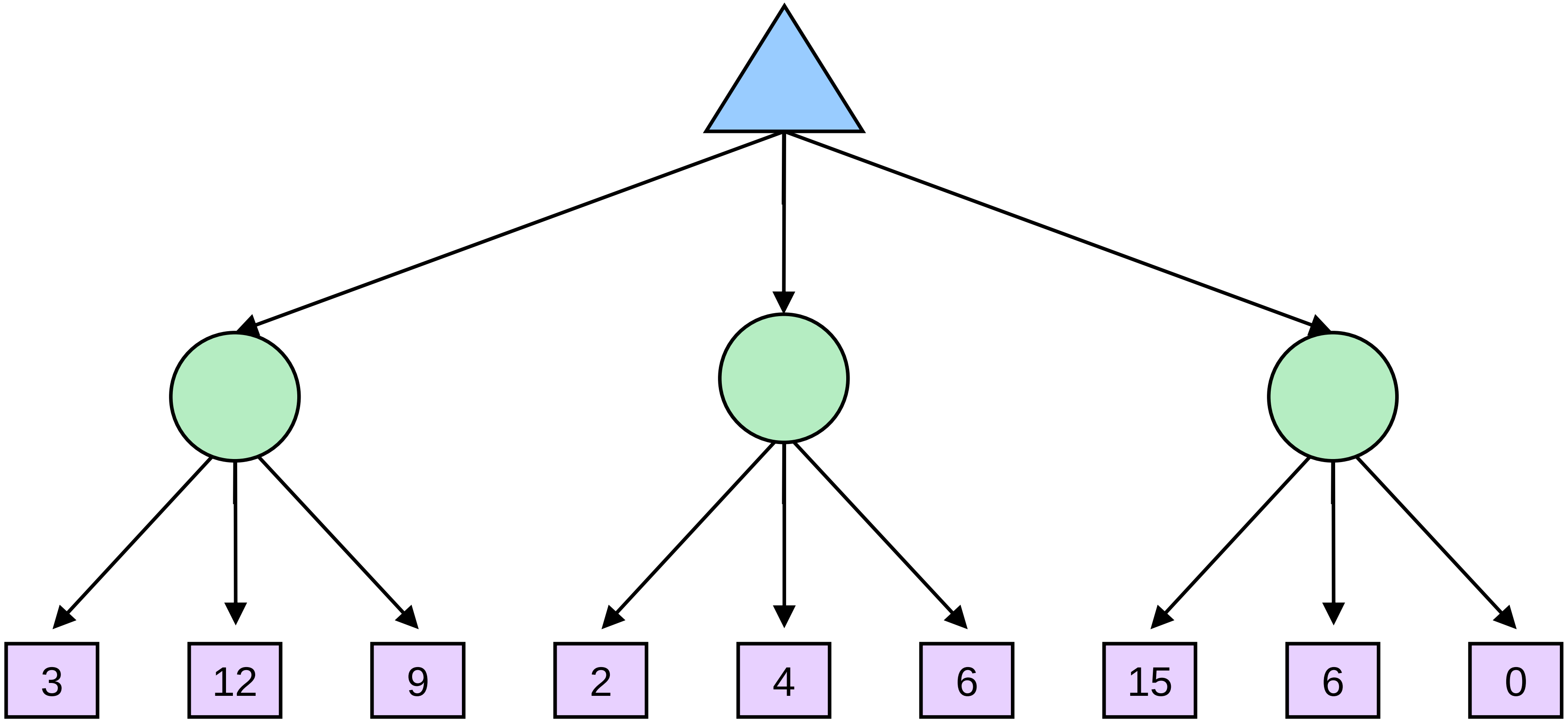
Expectimax Pszeudokód

```
def exp-value(state):
    initialize v = 0
    for each successor of state:
        p = probability(successor)
        v += p * value(successor)
    return v
```

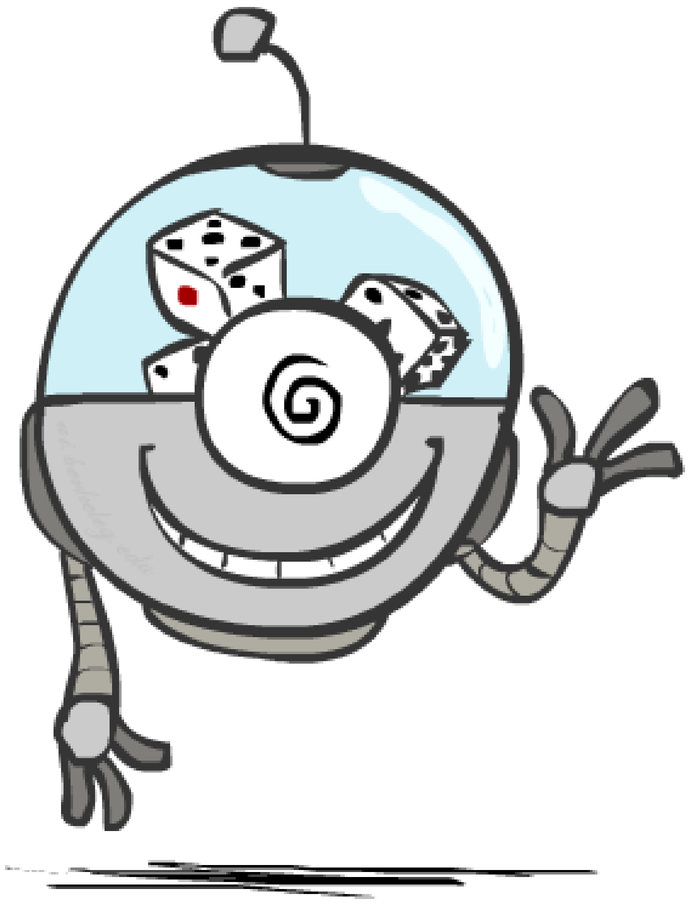


$$v = (1/2)\ (8) + (1/3)\ (24) + (1/6)\ (-12) = 10$$

Expectimax Példa



Valószínűségek



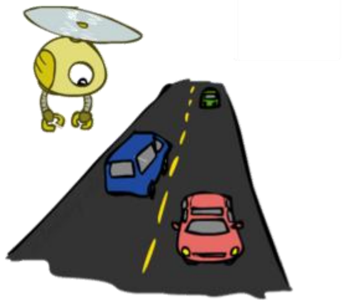
Emlékeztető: valószínűségek, várható érték

- A véletlen változó regy eseményt reprezentál, melynek kimenetele nem ismert
- A valószínűségi eloszlás voltaképpen súlyok hozzárendelése kimenetelekhez

- Példa: Forgalom az autópályán
  - Véletlen változó: T = Forgalom nagysága
  - Kimenetelek (értékek): T in {nincs, kicsi, nagy}
  - Eloszlás: P(T=nincs) = 0.25, P(T=kicsi) = 0.50, P(T=nagy) = 0.25

- Valószínűségi axiómák:
  - A valószínűségek nem negatívak
  - Minden lehetséges kimenetel összesített valószínűsége 1-et ad

0.25

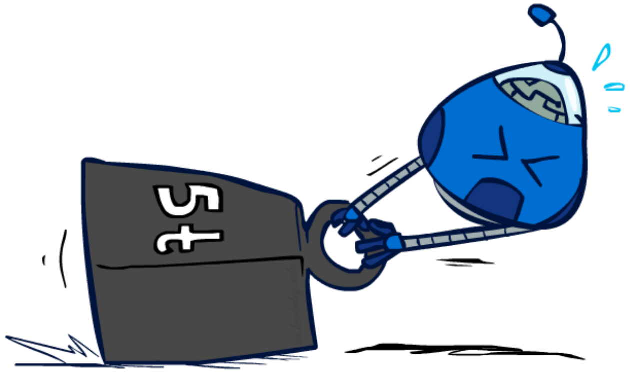


0.50

0.25

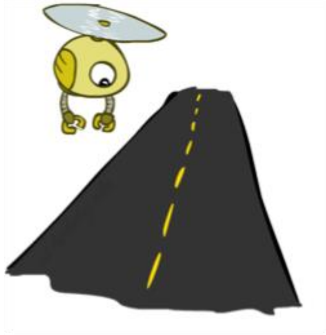
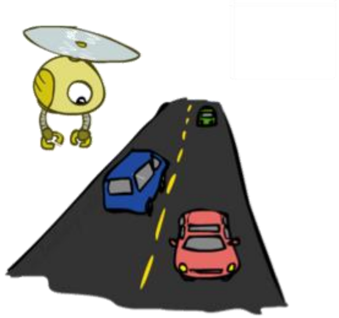
Emlékeztető: valószínűségek, várható érték

- Egy véletlen valószínűségi változó várható értéke a kimenetelek valószínűségi eloszlás szerint súlyozott átlaga



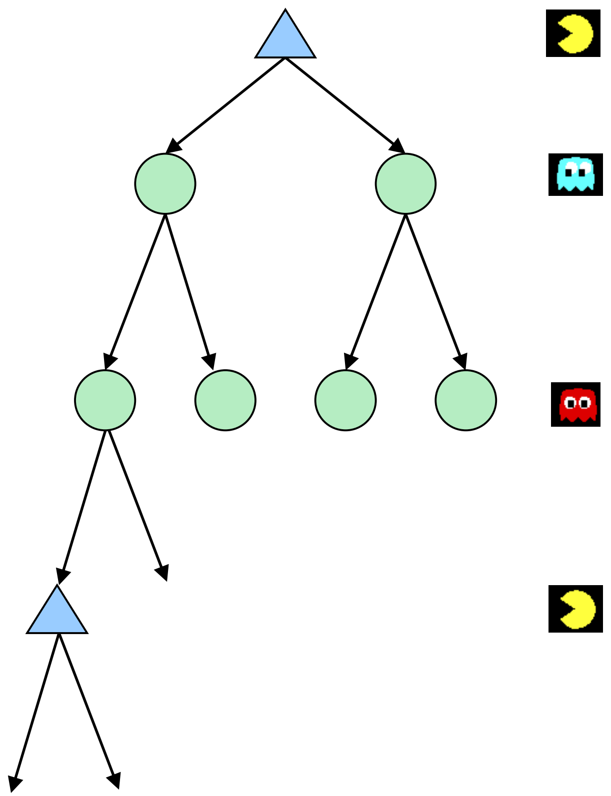
- Példa: Átlagosan mennyi idő kijutni a reptérre?

Idő:	20 min		30 min		60 min		
	x	+	x	+	x	⇒	35 min
Valószínűség:	0.25		0.50		0.25		


Milyen valószínűségeket használjunk?

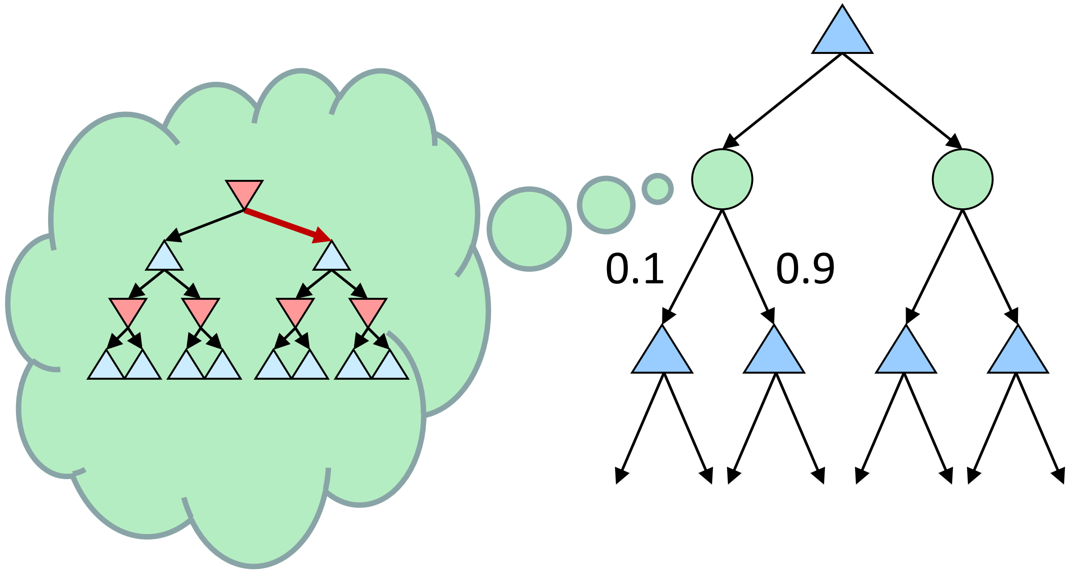
- Az expectimax keresésnél van egy valószínűségi modellünk arról, hogy az ellenfél (vagy a környezet) hogyan fog viselkedni tetszőleges állapotban
  - A modell lehet egy egyszerű egyenletes eloszlás (kockadobás)
  - A modell lehet kifinomult és nagy számításigényű
  - A modellben vannak véletlen csomópontok minden olyan kimenetelre, amelyet nem tudunk kézben tartani (ellenfél vagy környezet)
  - A modell lehetővé teheti az ellenfelek ellenséges akcióit



- Egyelőre tegyük fel, hogy a véletlen csomópont együtt jön létre az kimenetelek eloszlását leíró valószínűségekkel

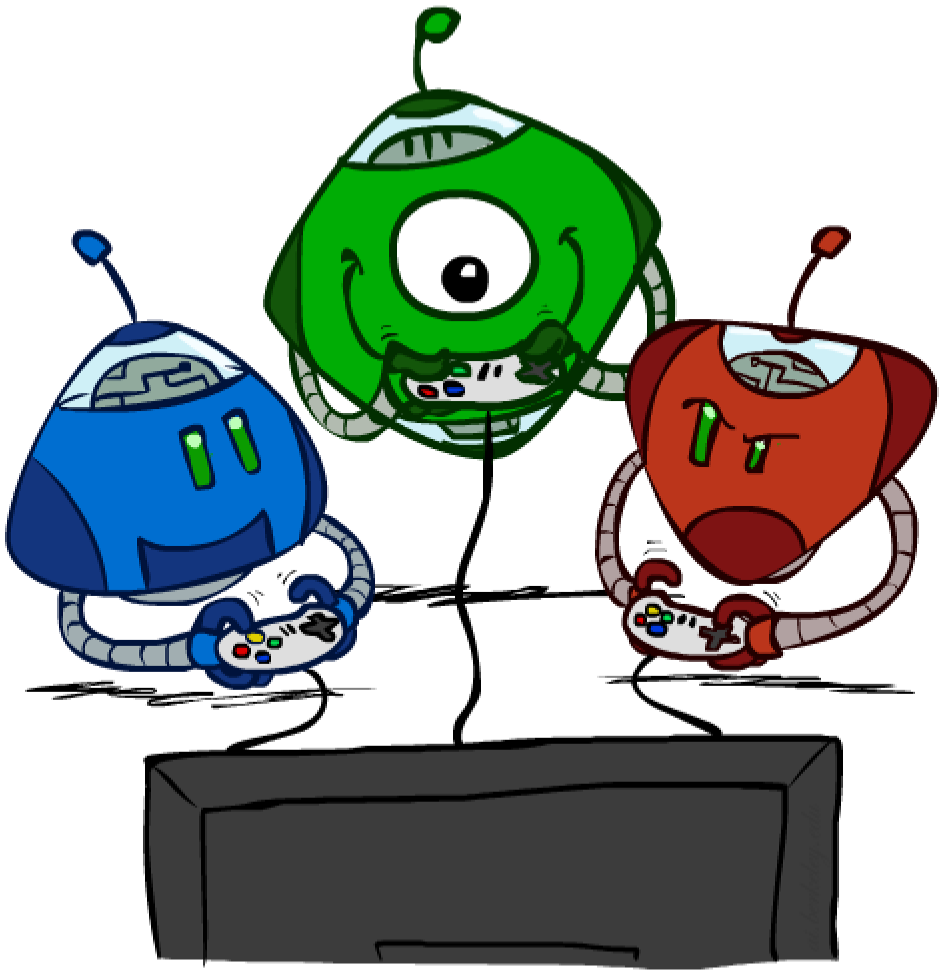
Informált valószínűségek

- Tegyük fel, hogy az ellenfél ténylegesen egy 2 mélységű minimaxot futtat, amelynek eredményét az esetek 80%-ban használja fel, a többiben véletlenszerűen lép
- Kérdés: Milyen fakeresést használjunk ehhez?



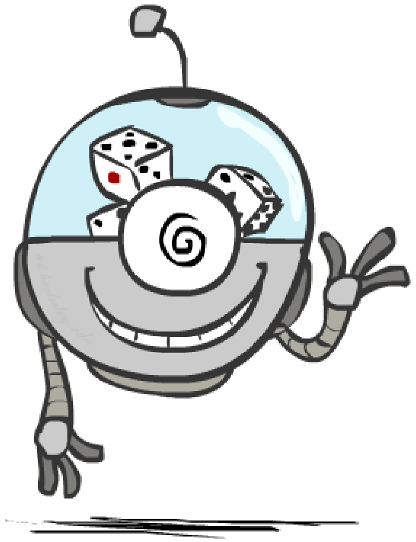
- Válasz: Expectimax!
  - Hogy minden egyes véletlen csomópont valószínűségét ki tudjuk számítani, le kell futtatni az ellenfél szimulációját
  - Ez viszonylag hamar válik nagyon lassúvá
  - Még rosszabb, ha azt is le kell szimulálni, hogy az ellenfél szimulál minket…
  - Kivéve a minimaxot, ami összezsugorodik egyetlen keresési fába

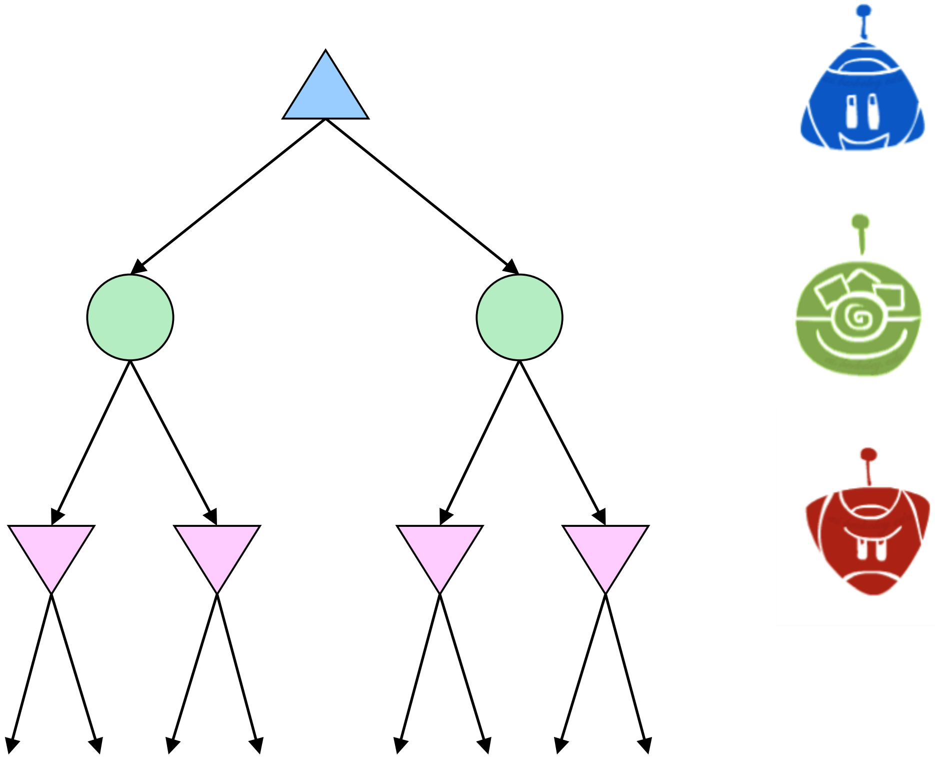
További játéktípusok



Kevert rétegű játék

- Expectiminimax
  - Környezet plusz egy véletlenszerű ágens, aki minden min/max játékos után lép
  - Minden csomópont kiszámítja a ‚megfelelő kombinációját’ a gyermekeinek



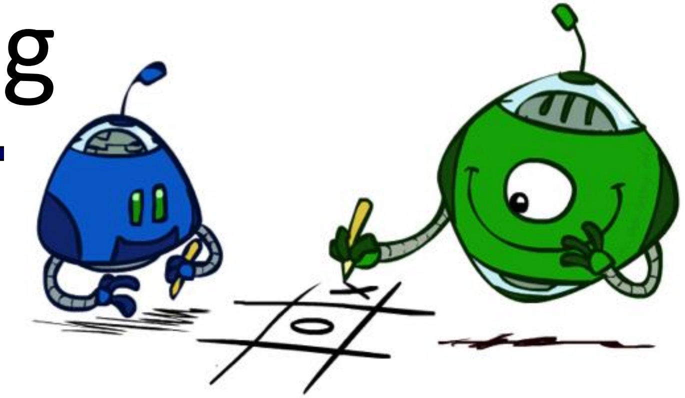


Példa: Ostábla(backgammon)

- Kockadobások növelik a b-t : 21 lehetséges dobás 2 d6 kockával
  - Ostábla(Backgammon) ≈ 20 legális lépés
  - Mélység 2 = 20 x (21 x 20)^3 = 1.2 x 10^9
- Ahogy a mélység növekszik annak a valószínűsége, hogy érintünk egy adott csomópontot csökken
  - A keresés hasznossága csökken
  - Ezért limitálni a mélységet kevésbé problémás
  - A vágás viszont nehéz…
- TDGammon: 2 mélységű keresés + jó kiértékelő függvény + megerősítéses tanulás => világbajnoki szintű játékos

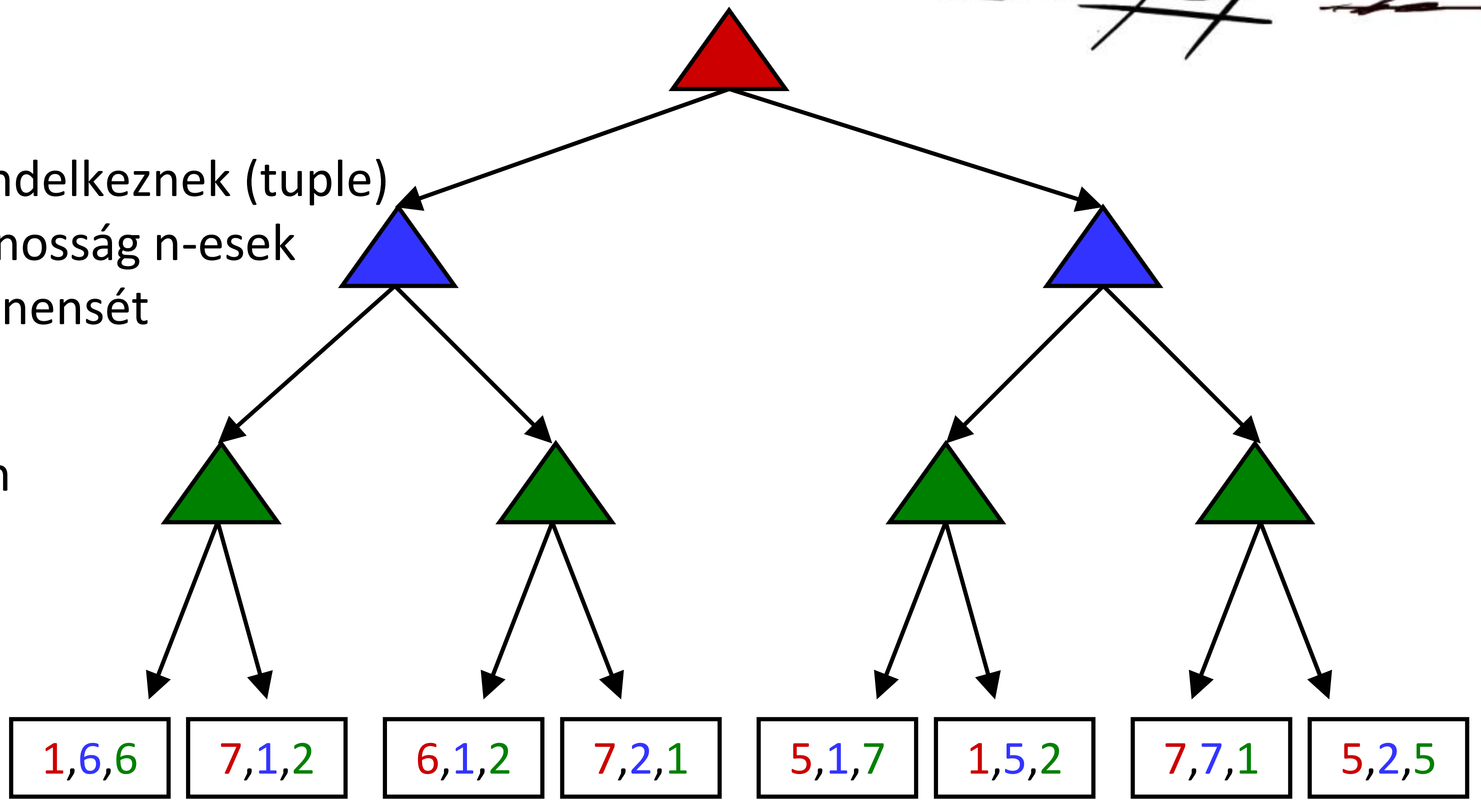
Image: Wikipedia

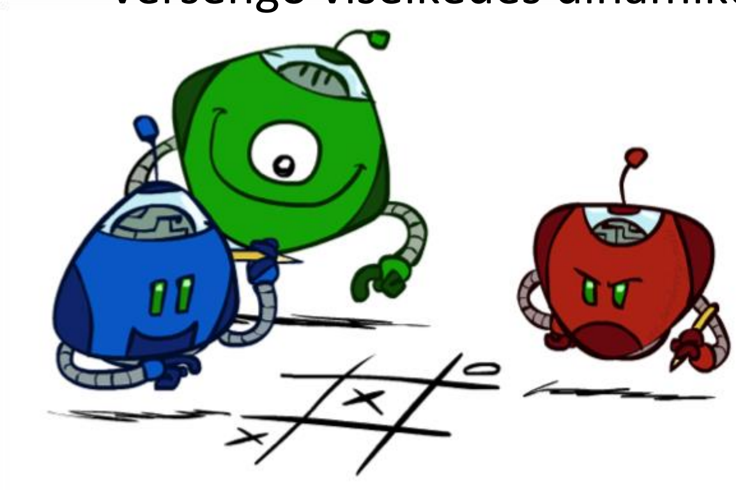
Több ágenses hasznosság



- Mi van abban az esetben, ha nem zéró összegű a játék, hanem több játékos van?

- Minimax általánosítása:
  - Levelek hasznosság n-esekkel rendelkeznek (tuple)
  - Csomópont értékek szintén hasznosság n-esek
  - Mindegyik játékos a saját komponensét maximalizálja
  - Ebből adódhat kooperatív és versengő viselkedés dinamikusan





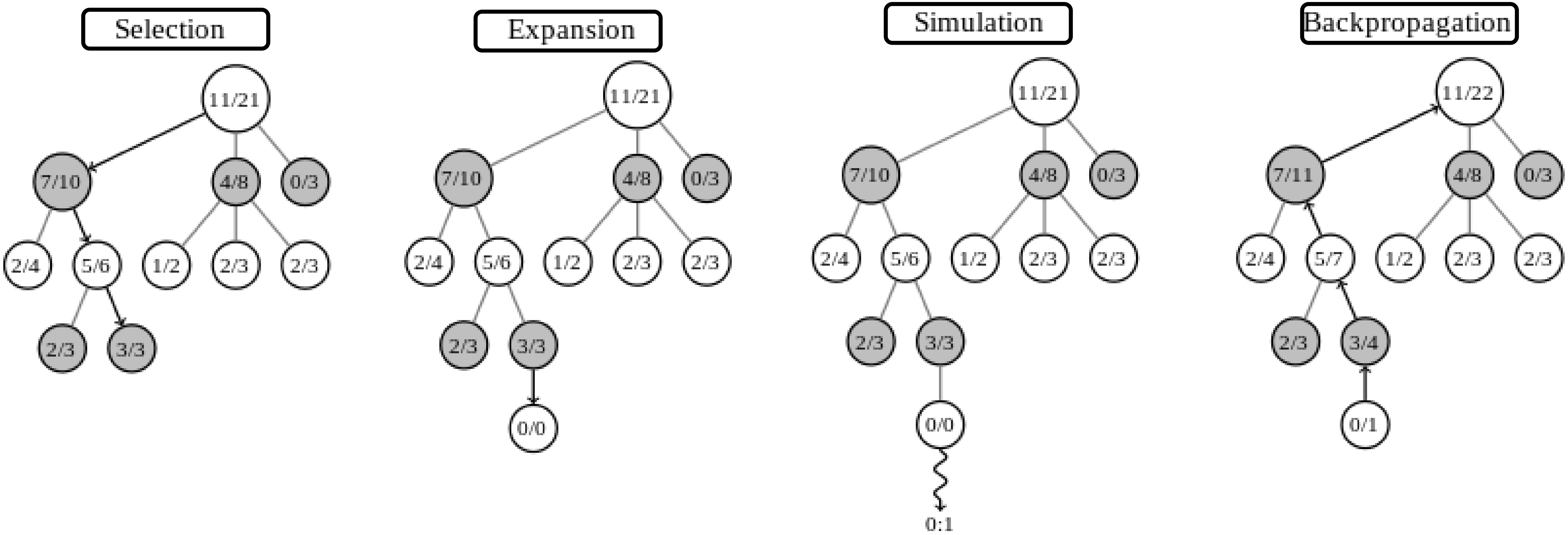
Monte Carlo Tree Search (MCTS)

- Monte Carlo kísérletek :
  - ismételt véletlen mintavétel (egy adott eloszlásból/mintahalmazból)
  - összegyűlt minták gyakorisága (aránya) alapján közelíthetők numerikusan nehezen számítható értékek
- MCTS: Monte Carlo mintavételezést használó fa keresés
- Jól használható komplex döntések meghozatalához
- A legerősebb GO MI-k (Fuego, Pachi, Zen, and Crazy Stone) mind MCTS-en alapulnak

Monte Carlo Tree Search lépései

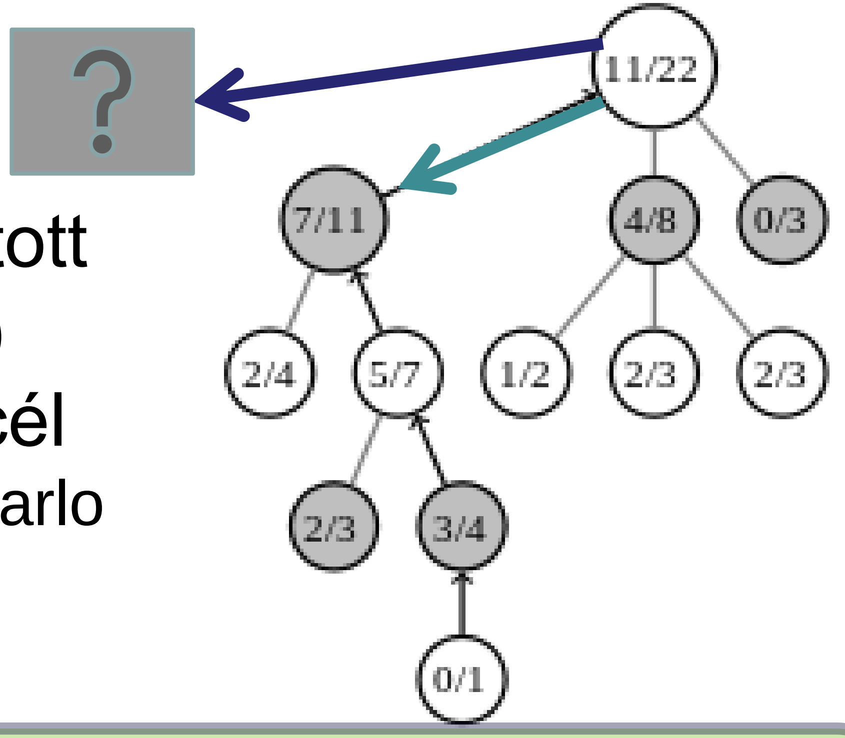
MCTS minden ciklusa az alábbi négy lépésből áll

1. Kiválasztás (selection)
2. Terjeszkedés (expansion)
3. Szimuláció (simulation)
4. Visszaterjesztés (backpropagation)



MCTS – Upper Confidence Bounds (UCB)

- Felfedezés (exploration): új még meg nem látogatott csomópont kiválasztása
- Kizsákmányolás(exploitation): már meglátogatott csomópont kiválasztása (pontosság növelése)
- Felfedezés és kizsákmányolás egyensúlya a cél
  - Kocsis, L. & Szepesvári, C.:Bandit based Monte-Carlo planning (2006) -> UCB
  - Optimális megoldáshoz konvergál



$$\boxed{\frac{w_i}{n_i}} + \boxed{c \cdot \sqrt{\frac{ln(t)}{n_i}}}$$

Exploitation **Exploration**

W_i #győzelmek száma i csomópont meglátogatása után

n_i #i csomópont meglátogatásainak száma

C felfedezési paraméter

t #i csomópont szülei meglátogatásainak száma

Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Méréstechnika és Információs Rendszerek Tanszék

Logikai ágens, szabályalapú rendszerek

Előadó:

Dr. Hullám Gábor

Előadás anyaga:

Dr. Dobrowiecki Tadeusz
Dr. Pataki Béla
Dr. Hullám Gábor

Szakértői rendszerek az MI történetében

- ~1930 Univerzális számítási modell: Turing-gép (1936), Univerzális Turing-gép, Church-Turing hipotézis, Zuse, Neumann,..: „vezérlő program is adat”:
- 1943 McCulloch & Pitts: Bináris kapcsolati agymodell
- 1950 Turing: "Computing Machinery and Intelligence"
- 1956 Dartmouth találkozó: "Artificial Intelligence” megnevezés elfogadása
- 1950s Korai MI programok: sakk, tételbizonyítás
- **Számítás(keresés) alapú MI**
- - A Fizikai Szimbólumrendszer hipotézise: A.Newel&H.A.Simon (1976)
- 1966—73 Számítási komplexitási korlátok a keresésben
  Elméleti korlátok a neurális hálózatokban
- **1969—79 Tudásalapú szakértői rendszerek**
  (szabályalapú szakértői rendszerek)
- **Tudásalapú MI**

Szakértői rendszerek az MI történetében

- **1969—79 Tudásalapú szakértői rendszerek (szabály alapú szakértői rendszerek)**

Tudásalapú MI

- 1986-- Neurális hálózatok újbóli megjelenése
- 1988-- Valószínűségi szakértői rendszerek
- 1995-- Gépi tanulás gyors fejlődése

Adatvezérelt MI (‚data driven’)

- 2001-- ‚Big Data’

Autonóm tanulás alapú MI

- 2011 -- Mély tanulás (deep learning)

Szabály alapú rendszerek = Tudásalapú rendszerek
= Szakértő rendszerek

Mycin, Stanford, ca. 1972, kb. 500 szabály, szakorvos szintje

IF The site of the culture blood, and
The gram stain of the organism is gramneg, and
The morphology of the organism is rod, and
The portal of entry of the organism is urine, and
The patient has not had a genito-urinary manipulative procedure, and
Cystitis is not a problem for which the patient has been treated
THEN There is suggestive evidence (.6) - the identity of the organism is e.coli

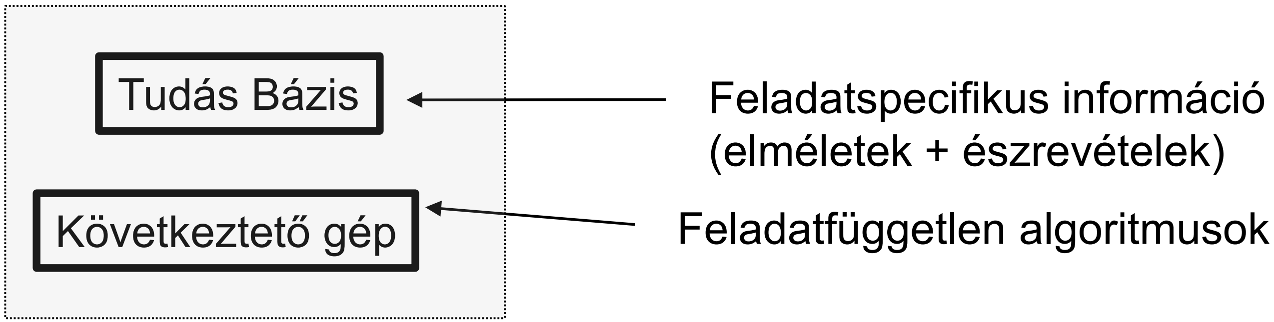
IF The identity of the organism is bacteroides
THEN Recommend therapy chosen from the following drugs:
1. clindamycin (0.99)
2. chloramphenicol (0.99)
3. erthromycin (0.57)
4. tetracycline (0.28)
5. carbenicillin (0.27)

A logika szerepe a komplex MI alkalmazásokban

- Logikai szabályalapú következtetés és beavatkozás

- Egy vagy több feltétel logikai állításként történő megfogalmazása
- Reakciók, beavatkozások megadása logikai szabályok formájában

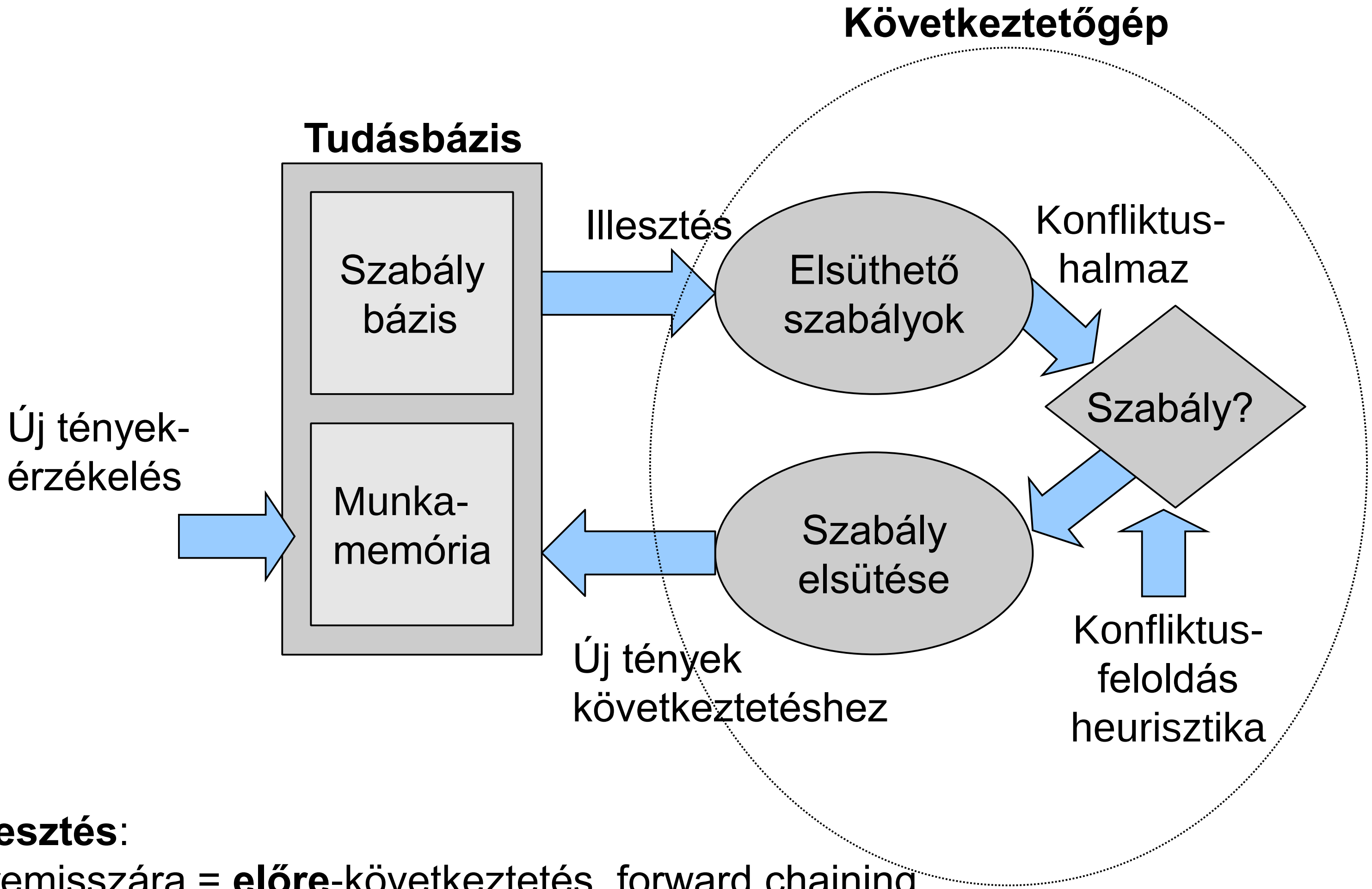
Logika



Mire használhatja az Ágens logikai képességeit?

Hogyan fejezze ki ehhez a tudását?

Milyen algoritmusok lesznek jók, ügyesek, …?



Illesztés:
Premisszára = **előre**-következtetés, forward chaining
Következményre = **hátra**-következtetés, backward chaining

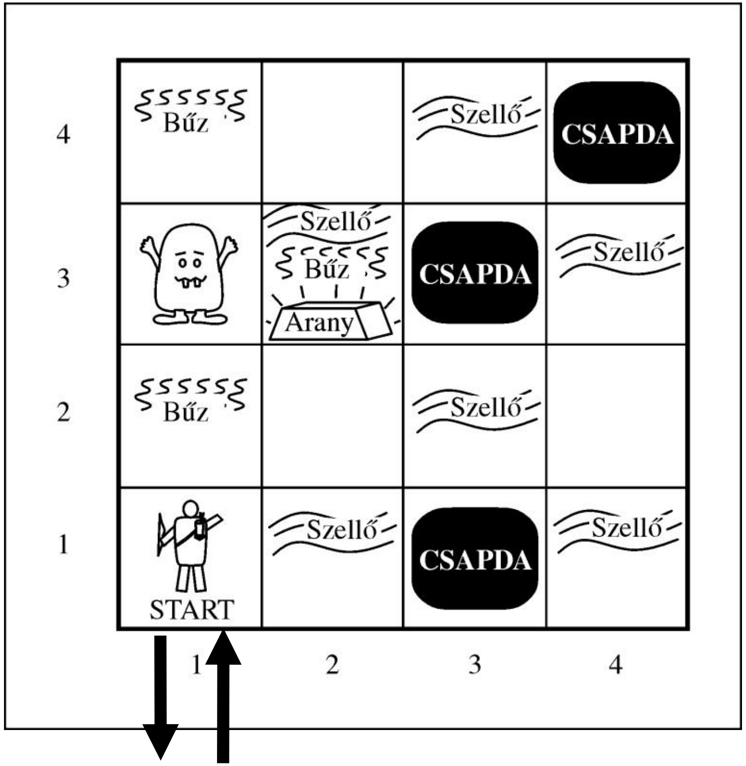
Szabályalapú szakértői rendszerek manapság

- Jogi szakértői rendszerek
- Adó- és pénzügyi tanácsadó rendszerek
- Ügyfélszolgálat és technikai támogató rendszerek

- Természetes nyelvű interfészek
  - Digitális asszisztensek
  - Csetbotok

Logikusan gondolkozó ágens - Mit jelentsen ez?

Esettanulmány - a „Wumpus világ” környezetében (jegyzet)



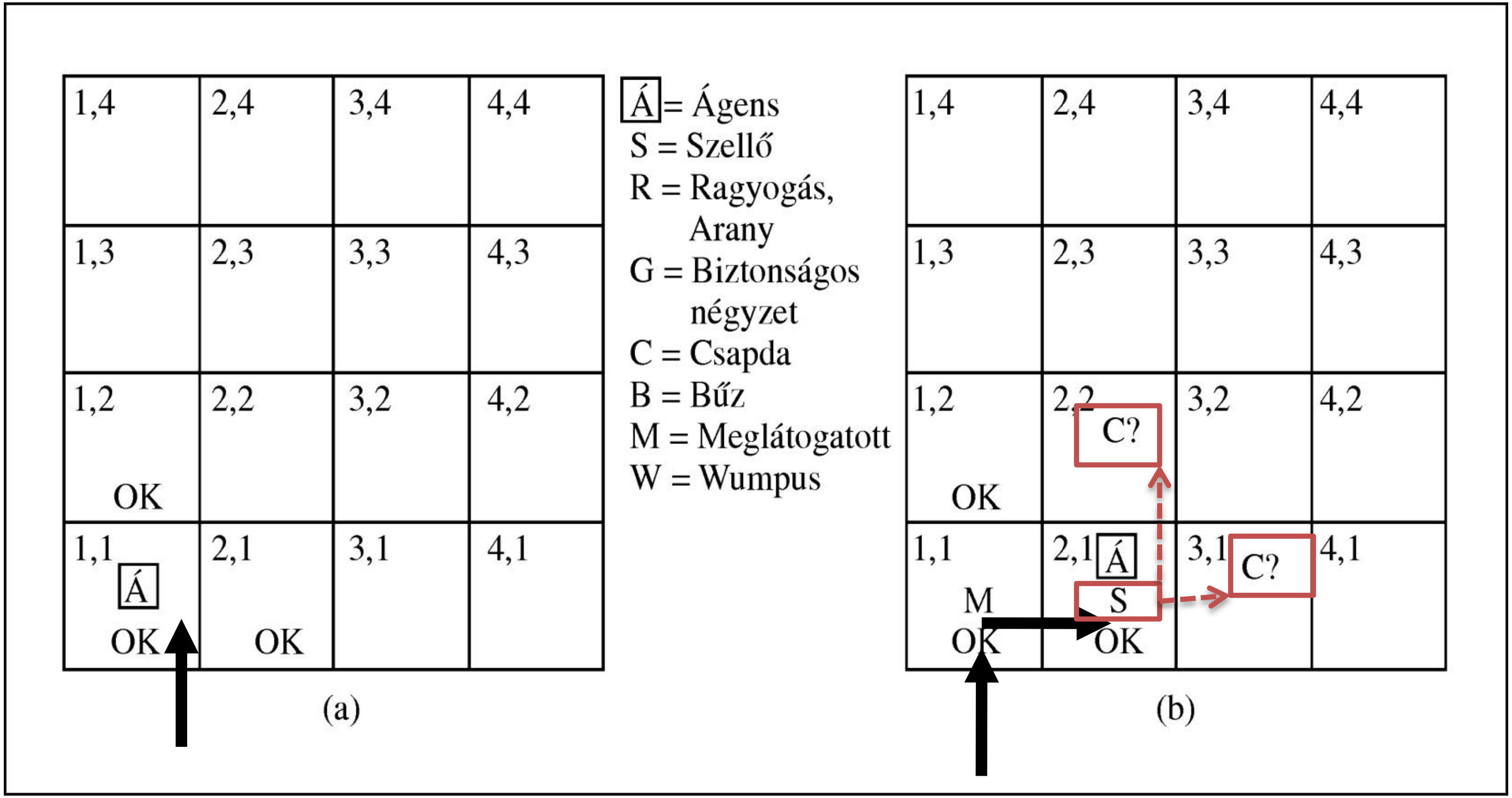
Ha ágens, akkor érzékelések, cselekvések és célok.

Érzet(ek):
[Bűz, Szellő, Csillog, Ütközés, Sikoly]
hol vagyunk, nem tudni.

Cselekvések:
Előre, Jobbra/Balra, Megragad,
Lő, Mászik, … ágens meghal,
ha csapdára vagy Wumpusra rálép.

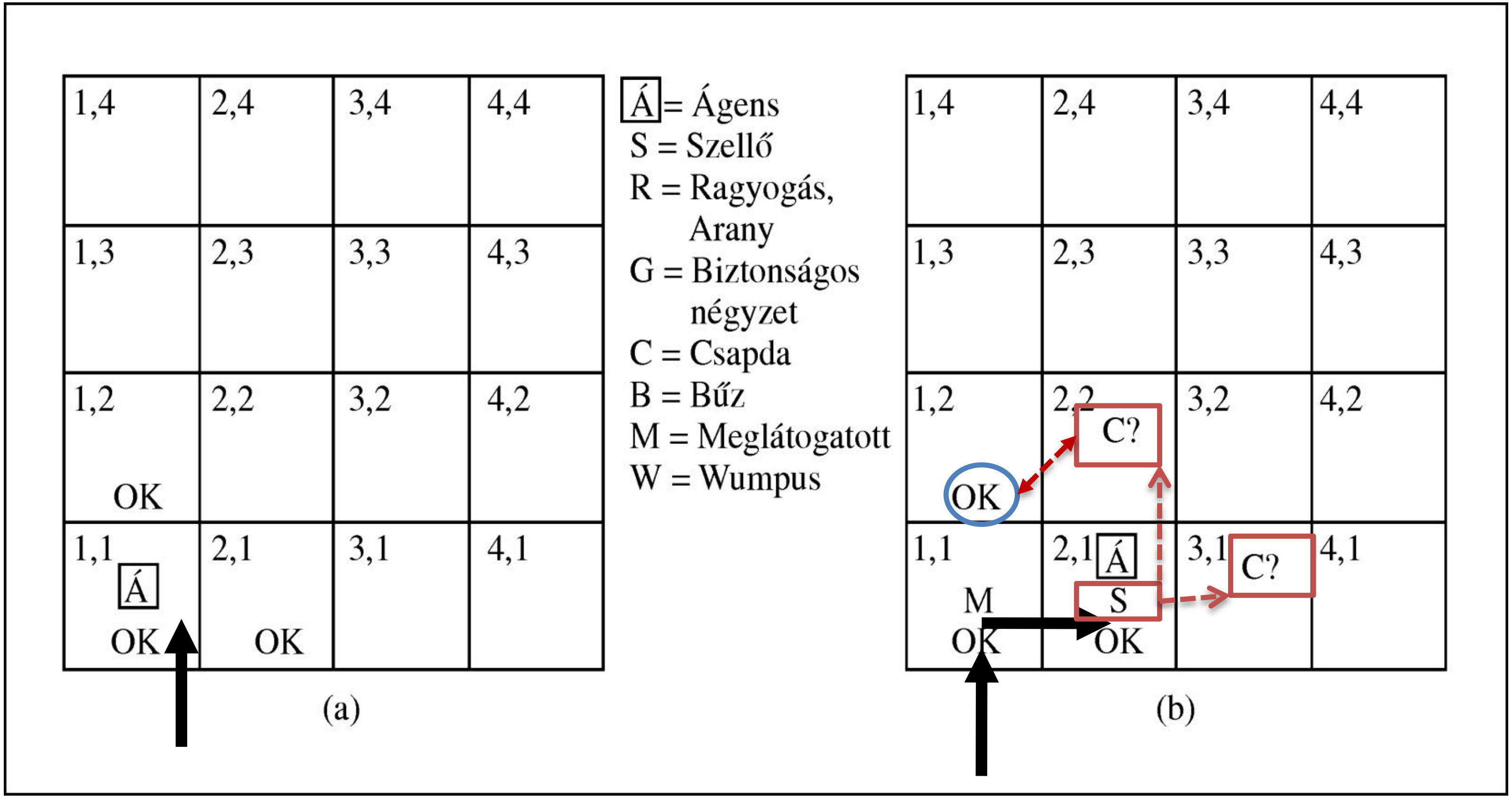
cél: arany - megtalálni és kihozni

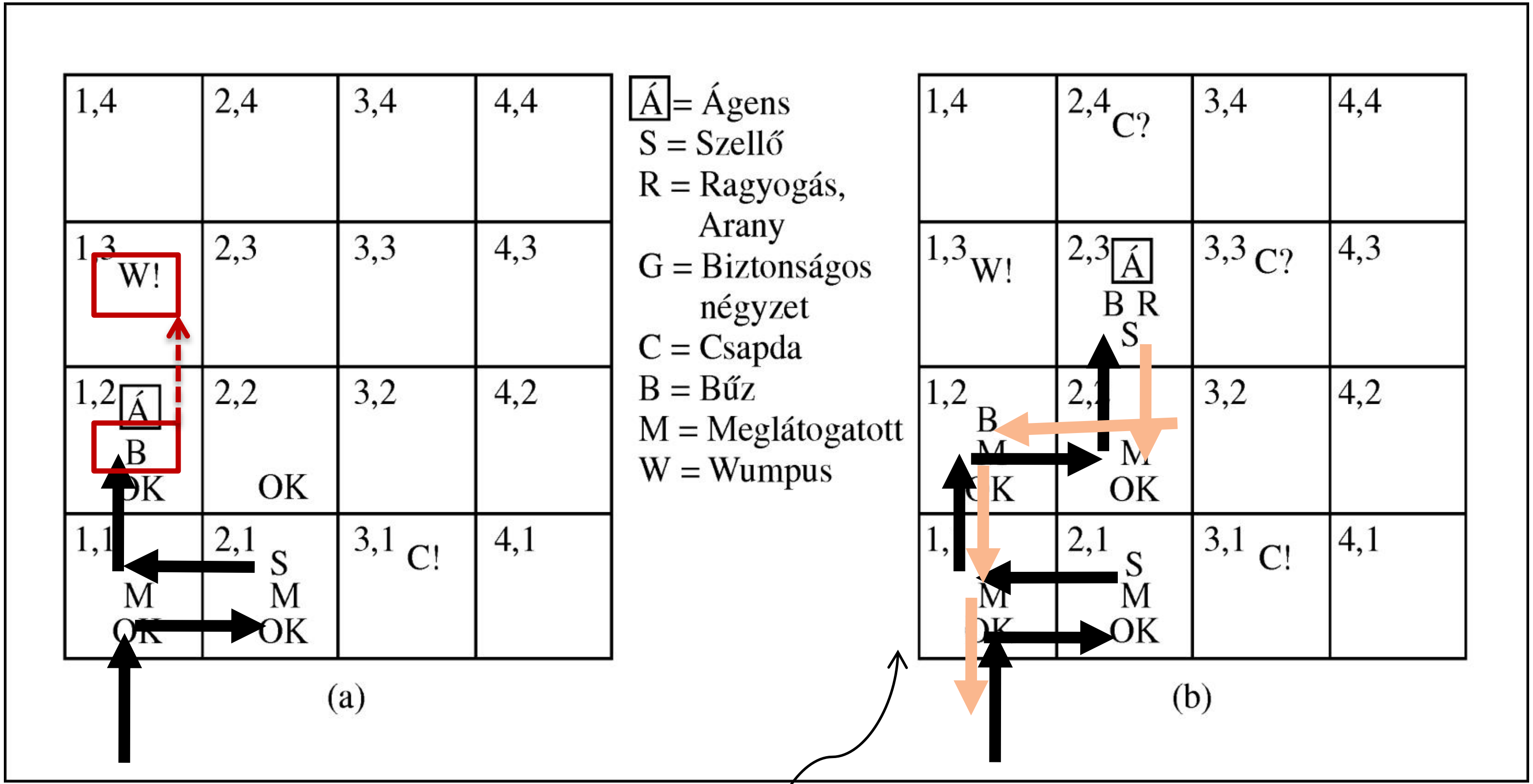
Hogyan biztosítható a siker (logikai gondolkodással)?


1,4
2,4
3,4
4,4
1,3
2,3
3,3
4,3
1,2
2,2
3,2
4,2
OK
1,1
Á
OK
2,1
OK
3,1
4,1
Á = Ágens
S = Szellő
R = Ragyogás, Arany
G = Biztonságos négyzet
C = Csapda
B = Bűz
M = Meglátogatott
W = Wumpus
1,4
2,4
3,4
4,4
1,3
2,3
3,3
4,3
1,2
OK
2,2
C?
3,2
4,2
1,1
M
OK
2,1
Á
S
OK
3,1
C?
4,1

(a)

(b)


1,4
2,4
3,4
4,4
1,3
2,3
3,3
4,3
1,2
OK
2,2
3,2
4,2
1,1
Á
OK
2,1
OK
3,1
4,1
(a)
Á = Ágens
S = Szellő
R = Ragyogás, Arany
G = Biztonságos négyzet
C = Csapda
B = Bűz
M = Meglátogatott
W = Wumpus
1,4
2,4
3,4
4,4
1,3
2,3
3,3
4,3
1,2
OK
2,2
C?
3,2
4,2
1,1
M
OK
2,1
Á
S
OK
3,1
C?
4,1
(b)



Környezeti titkok (nem hozzáférhető!) kiderítése logikai modellezéssel és gondolkodással, majd jó cselekvések eldöntése.

Logika – reprezentáció és manipuláció eszköze

Szintaktika - legálisan létrehozható szimbolikus mondatok
Szemantika - a világ tényei, amikre a szimbolikus mondatok vonatkoznak

Vonzatreláció és a következtetés

Tények neveznek valós dolgokat, amelyek egymással kapcsolatban vannak. Ha az egyik igaz a világban, és a másik szükségszerűen is az:

vonzat: logikai konzekvencia dolgok között.

Vonzatreláció tudásbázis TB és egy α mondat között:

α vonzata TB -nek: **TB |= α** , ha α minden olyan világban igaz, ahol TB is.

Következtetés: a vonzat „kiszámítása” a mondatok formális manipulálásával.
Bizonyítás = a következtetési algoritmus lépéssorozata

α bizonyítható TB -ből: **TB |-- α**

Logika – reprezentáció és manipuláció eszköze

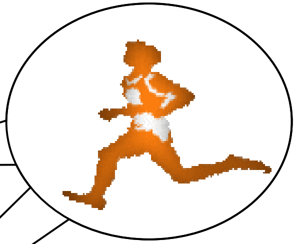
Szintaktika - legálisan létrehozható szimbolikus mondatok
Szemantika - a világ tényei, amikre a szimbolikus mondatok vonatkoznak

Mi egy mondat jelentése?

- A mondat írójának egy interpretációt kell adnia hozzá, kijelentve, hogy milyen tény tartozik hozzá.
- Egy mondat önmagában nem jelent semmit és lehet igaz vagy hamis.
- Egy mondat igaz egy bizonyos interpretációban, ha a dolgok állása ilyen.
- **Az igazság függ mind a mondat interpretációjától, mind a világ aktuális állapotától.**

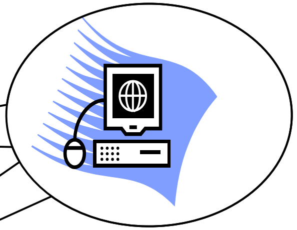
Szemantika és interpretáció

mi lehet pl. a **Fut_Ágens1** mondat jelentése?



Ha „fut” fizikai mozgás, „Ágens1” egy személy, és a világ most: akkor ez a mondat (most) **igaz**.

Ha „fut” nem terminált működés, „Ágens1” egy program, és a világ most ugyanilyen, akkor ez a mondat (most) **hamis**.



Ha „fut” fizikai mozgás, „Ágens1” egy személy és a világ most: akkor ez a mondat (most) **hamis**.

Ha „fut” nem terminált működés, „Ágens1” egy program, és a világ most ugyanilyen, akkor ez a mondat (most) **igaz**.

Modellek

Bármely (formális is) világ, ahol egy mondat igaz egy bizonyos interpretációban. Egy mondatnak számos modellje lehet.

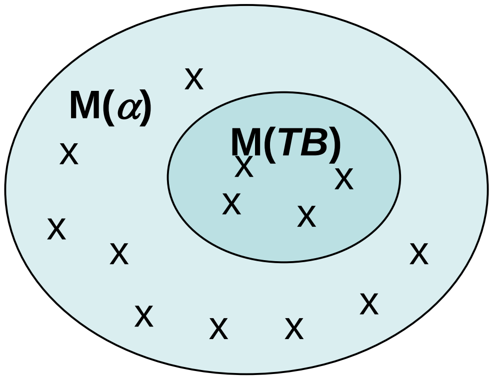
Minél többet állítunk (pl. minél több információt adunk hozzá a tudásbázishoz), annál kevesebb modellünk lesz.

Modellek nagyon fontosak (mert):

egy α mondat vonzata a TB tudásbázisnak, ha a TB modelljei mind modelljei az α-nak is.

Ha ez így van, akkor ha TB igaz, akkor α is igaz

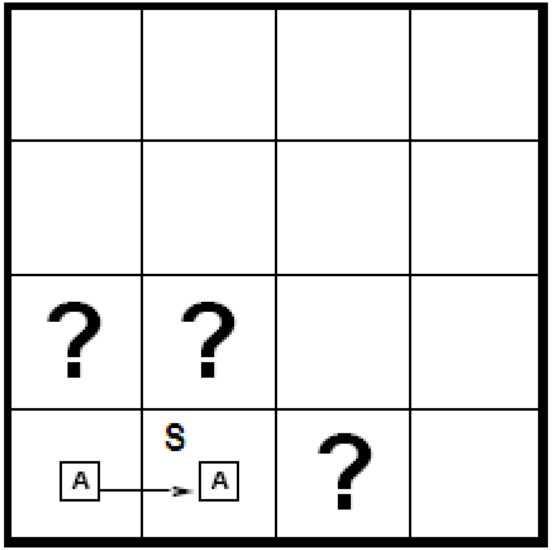
TB |= α akkor és csak akkor, ha M(TB) $\subseteq$ M(α)

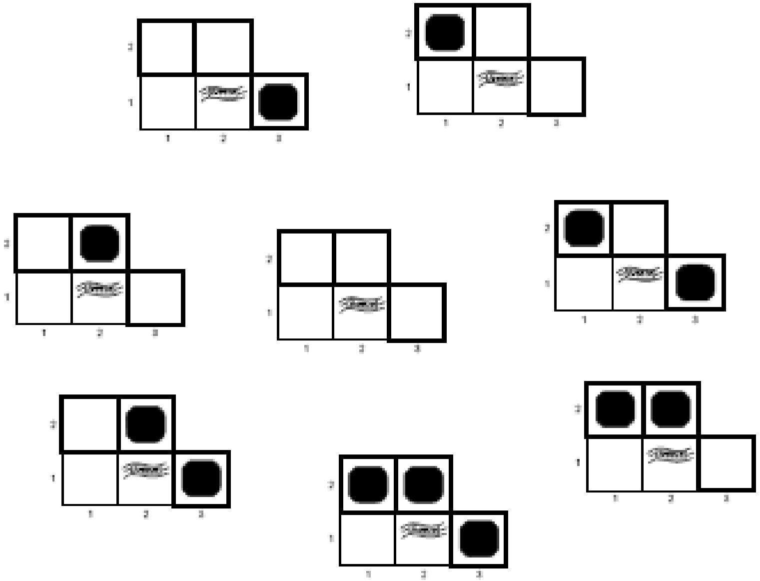


Vonzat és modellek Wumpus világban

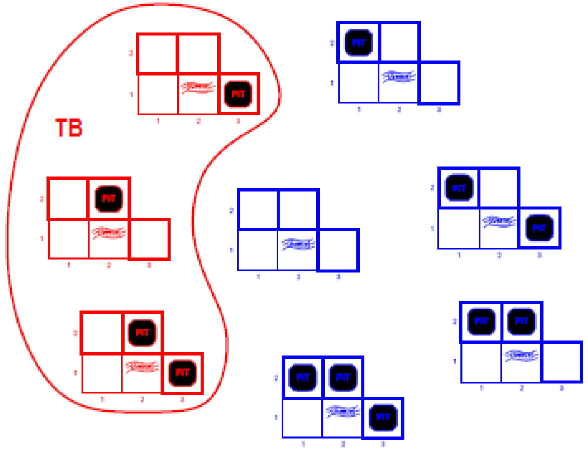
(1,1)-ben semmi érzet, lépés (2,1)-re, szellőt érezni, ...

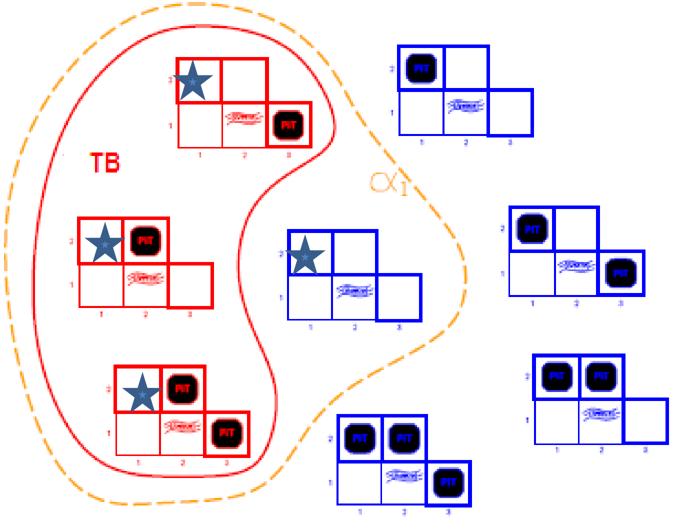
Lehetséges modellek a (?)-re, csakis csapdát feltételezve:
3 db bináris változó, 8 db lehetséges modell

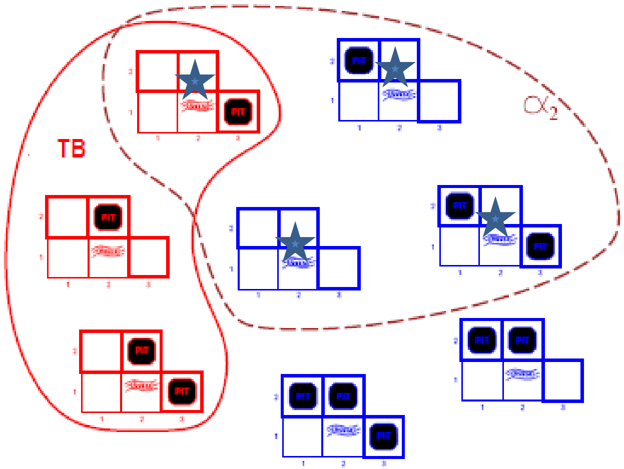




TB = Wumpus világ szabályai + megfigyelések







α1 = „[1,2] biztonságos",★

***TB* |= α1**, modellellenőrzéssel

α2 = „[2,2] biztonságos",★

***TB* |=/= α2**, modellellenőrzéssel

Ha adott egy TB, a **következtetés**:
(1) létrehozhat új α mondatot, amely vonzata a TB-nak
(2) ha adott α mondat, eldöntheti hogy α vonzata-e a TB-nak.

Igazságtartó következtetés (extenzionális), ha csak igazságfüggvényekkel dolgozik (igazságfüggvény logikai értéke csakis a részeinek és a logikai műveletek definíciójától függ, nem függ az interpretációtól).

Formális bizonyítás – igazságtartó következtetési eljárás lépései.

Következtetési eljárás teljes: ha minden vonzatmondathoz talál egy bizonyítást, avagy ami igaz, az bebizonyítható. **A |= B** -ből **A |- B**

Következtetési eljárás helyes: ha minden bizonyított mondat vonzatrelációban áll a felhasznált tényekkel, avagy ami bebizonyított, az igaz is. **A |- B** -ből **A |= B**

Érvényes (analitikus mondat, tautológia): ha minden világban minden lehetséges interpretációja igaz, függetlenül attól, hogy mit akar jelenteni és mi a világ állapota.

Pl. „Van bűz az [1,1]-ben, vagy nincs bűz az [1,1]-ben”

Kielégíthető: ha létezik olyan interpretáció, hogy valamely világban igaz. Ami nem kielégíthető, az **kielégíthetetlen.**

A	B	$A \vee (B \vee \neg B)$	$A \wedge (B \wedge \neg B)$	$A \wedge B$
0	0	1	0	0
0	1	1	0	0
1	0	1	0	0
1	1	1	0	1

Melyik állítás érvényes, kielégíthető, ill. kielégíthetetlen?

Ítéletkalkulus (Arisztotelész, Euklidész, i.e. 300, Leibnitz, Boole, XVII sz.)

Szintaktika:

igaz és **hamis** logikai konstansok, P1, P2, ... ítélet szimbólumok

ítéletmondatok, $\neg$ **P**, **P1 $\wedge$ P2, P1 $\vee$ P2, P1 $\rightarrow$ P2, P1 $\Leftrightarrow$ P2** is ítéletmondatok.

precedencia: $\neg$, $\wedge$, $\vee$, $\rightarrow$ és $\Leftrightarrow$.

Szemantika

Minden modell igaz/hamis értéket rendel minden ítéletszimbólumhoz, pl. P1 = igaz, P2 = hamis, P3 = igaz, ...

Tetszőleges mondat szemantikája - rekurzív elemzés, pl.:

$\neg$P1 $\wedge$ (P2 $\vee$ P3) = hamis $\wedge$ (hamis $\vee$ igaz) = hamis $\wedge$ igaz = hamis

Összekötő jelek igazságtáblái

P1	P2	$\neg$ P	P1 $\wedge$ P2	P1 $\vee$ P2	**P1 $\rightarrow$ P2**	P1 $\Leftrightarrow$ P2
0	0	1	0	0	**1**	1
0	1	1	0	1	**1**	0
1	0	0	0	1	**0**	0
1	1	0	1	1	**1**	1

N argumentumon értelmezett logikai függvény 2^{2^N}

Az ítéletlogika ekvivalencia szabályai

$(a \wedge b) \Leftrightarrow (b \wedge a)$	$\wedge$ kommutativitása
$(a \vee b) \Leftrightarrow (b \vee a)$	$\vee$ kommutativitása
$((a \wedge b) \wedge c) \Leftrightarrow (a \wedge (b \wedge c))$	$\wedge$ asszociativitása
$((a \vee b) \vee c) \Leftrightarrow (a \vee (b \vee c))$	$\vee$ asszociativitása
$\neg(\neg a) \Leftrightarrow a$	dupla negálás eliminálása
$(a \rightarrow b) \Leftrightarrow (\neg b \rightarrow \neg a)$	kontrapozició
$\mathbf{(a \rightarrow b) \Leftrightarrow (\neg a \vee b)}$	**implikáció elimináció**
$(a \Leftrightarrow b) \Leftrightarrow ((a \rightarrow b) \wedge (b \rightarrow a))$	ekvivalencia eliminálás
$\neg(a \wedge b) \Leftrightarrow (\neg a \vee \neg b)$	De Morgan
$\neg(a \vee b) \Leftrightarrow (\neg a \wedge \neg b)$	De Morgan
$(a \wedge (b \vee c)) \Leftrightarrow ((a \wedge b) \vee (a \wedge c))$	$\wedge$ disztributivitása $\vee$ felett **DNF**
$(a \vee (b \wedge c)) \Leftrightarrow ((a \vee b) \wedge (a \vee c))$	$\vee$ $\wedge$ disztributivitása $\wedge$ felett **CNF**

két állítás logikailag ekvivalens, ha ugyanazokban a modellekben igaz:

$A \Leftrightarrow B$ a.cs.a, ha $A \models B$ és $B \models A$

TB $\models \alpha$ a.cs.a., ha (TB $\rightarrow \alpha$) érvényes

TB $\models \alpha$ a.cs.a., ha (TB $\wedge \neg \alpha$) kielégíthetetlen (reductio ad absurdum)

Következtetés (bizonyítások)

Modell-ellenőrzés – mondat kielégíthető-e (SAT satisfiability)

Igazságtábla listázás (exponenciális)

Visszalépéses keresés

Heurisztikus keresés modellek terében (helyes, de nem teljes)
(lokális kereséssel)

Következtetési szabályok alkalmazása

Új mondatok legális generálása régebbi mondatokból.

Bizonyítás: következtetési szabályok sorozata.
speciális operátorok **megszokott** keresési algoritmusokban.

- keresés általános operátorokkal (természetes dedukció)
- teljes keresés teljes operátorral általános logikában (rezolúció, exp.!)
- teljes keresés teljes operátorral redukált logikában
(Horn-klózok, Modus Ponens, redukált komplexitás!)

(más kombinációk nem garantálják a következtetés teljességét)

Egy történet: „Ádám, Béla és Csaba közül valaki egy üvegház ablakát törte ki. Ádám azt állítja, hogy: 'Béla tette, Csaba nem bűnös.‘ Béla azt állítja, hogy: 'Ha Ádám bűnös, akkor Csaba is'. Csaba azt állítja, hogy: 'Nem én tettem; többiek közül valaki'."

Konzisztensek (egyszerre igazak) ezek az állítások?
Ha egyik sem bűnös, van-e, aki hazudik?
Ha mindenki igazat mond, akkor ki a bűnös? Stb.

A történet **ítélet szimbólumai**:
A: Ádám <u>nem</u> bűnös,
B: Béla <u>nem</u> bűnös,
C: Csaba <u>nem</u> bűnös és a mondanivalójuk:

SA $= \neg B \wedge C$ **SB** $= \neg A \rightarrow \neg C$ **SC** $= C \wedge (\neg B \vee \neg A)$

az un. „elmélet” pedig (az igaznak tartott mondatok összessége), ami egyben a TB tartalma: **TB** $= SA \wedge SB \wedge SC$

A	B	C	SA	SB	SC	TB = SA ∧ SB ∧ SC
H	H	H	H	I	H	H
H	H	I	I	H	I	H
H	I	H	H	I	H	H
H	I	I	H	H	I	H
I	H	H	H	I	H	H
I	H	I	I	I	I	I
I	I	H	H	I	H	H
I	I	I	H	I	H	H

felsorolásos bizonyítás,
2^N sor, $O(2^N)$ komplexitás (NP)

Visszalépéses keresés modellek Terében - Davis-Putnam, stb.

A	B	C	SA	SB	SC	TB
~~H~~	~~H~~	~~H~~	~~H~~	~~I~~	~~H~~	~~H~~
~~H~~	~~H~~	~~I~~	~~I~~	~~H~~	~~I~~	~~H~~
~~H~~	~~I~~	~~H~~	~~H~~	~~I~~	~~H~~	~~H~~
~~H~~	~~I~~	~~I~~	~~H~~	~~H~~	~~I~~	~~H~~
~~I~~	~~H~~	~~H~~	~~H~~	~~I~~	~~H~~	~~H~~
I	H	I	I	I	I	I
~~I~~	~~I~~	~~H~~	~~H~~	~~I~~	~~H~~	~~H~~
~~I~~	~~I~~	~~I~~	~~H~~	~~I~~	~~H~~	~~H~~

Keressük TB = SA ∧ SB ∧ SC =
¬B ∧ C ∧ (C ∧ (¬B ∨ ¬A)) ∧ (¬A → ¬C)
modelljét (ahol a TB kielégíthető,
azaz igaz lesz)

¬B ∧ C ∧ (¬B ∨ ¬A) ∧ (¬A → ¬C)
igazához szükséges, hogy ¬B legyen
igaz, azaz **B** legyen **hamis**

C ∧ (¬A → ¬C) igazához szükséges,
hogy **C** legyen **igaz**

¬A → Hamis igazához szükséges,
hogy ¬A legyen hamis, azaz
A legyen **igaz**

(Vajon más kombinációkra is?)

Az ítéletlogika következtetési mintái (operátorok)

Modus Ponens
(Implikáció eliminálása)
néha elég is

$$\frac{A \rightarrow B \qquad A}{B}$$

AND eliminálása

$$\frac{A_1 \wedge A_2 \wedge \ldots \wedge A_n}{A_1, A_2, \ldots, A_n}$$

gépi világhoz

AND bevezetése

$$\frac{A_1, A_2, \ldots, A_n}{A_1 \wedge A_2 \wedge \ldots \wedge A_n}$$

TB = 1 állítás!

OR bevezetése

$$\frac{A_i}{A_1 \vee A_2 \vee \ldots \vee A_n}$$

Dupla negálás eliminálása

$$\frac{\neg\neg A}{A}$$

Elemi (egység)**rezolúció**

$$\frac{A \vee B \qquad \neg B}{A}$$

Rezolúció

$$\frac{B \vee A \qquad \neg B \vee G}{A \vee G}$$

Logikai következtetések fajtái

Tudás átalakítás (manipulálás) modellje

Dedukció (már az ógörögök)

formálisan érvényes következtetés
olyan tények származtatása, amelyek a premisszákból mindenképpen következnek (un. 'igazságtartó' eljárás)

Ha kutya nagy, akkor sokat eszik (feltétel)
kutya nagy (tapasztalat, evidencia)
sokat eszik (következtetés)

Tömörítés, általánosítás: példákból tanulás modellje!

Indukció (középkori arabok, skolasztikusok)

Kutya1 nagy, Kutya2 nagy, …, Kutya1000 nagy (tapasztalat)
Minden kutya nagy (induktív általánosítás)

miért fontos? sok tény helyett egyetlen egy (új) tény
probléma mérete csökken,
exponenciális jelleg kevésbé zavaró,
de formálisan nem igaz! Miért?

Abdukció - 'belátás' folyamata (Charles Pierce, 1839–1914)

ha kutya nagy, akkor sokat eszik (feltétel)
sokat eszik (tapasztalat)
kutya nagy (hipotézis)

Diagnózis, magyarázatadás modellje!

ha kutya nagy, akkor sokat eszik
ha kutya éhes, akkor sokat eszik
sokat eszik (tapasztalat)
éhes, vagy nagy? (melyik hipotézis esélyesebb)

formálisan miért nem igaz? de miért nagyon fontos?

Természetes rendszermodell
(a feladat megfogalmazása)

ha ‘egy rendszer X állapotban van',
akkor 'Y a rendszer megfigyelhető viselkedése'

Természetes kérdés (feladat megoldása)
milyen állapotban van a rendszer, ha konkrét információt (Y) kapok a viselkedéséről?

Tudjuk, hogy a beteg lázas.
Dedukció, vagy abdukció?

ha INFLUENZÁS, akkor LÁZAS
LÁZAS
INFLUENZÁS?

ha LÁZAS, akkor INFLUENZÁS
LÁZAS
INFLUENZÁS?

Ítéletlogika eldönthető és teljes

Minden jól definiált mondat akár **igaz**, akár **hamis** volta belátható véges algoritmussal (a vonzat eldönthető) $\rightarrow$ a **következtetés igazságtábla módszere teljes**, mindig lehetséges kiszámolni a tábla 2^n sorát bármely n ítéletszimbólumot tartalmazó bizonyítás esetében.

A számítási idő azonban n-ben **exponenciális**.

Cook (1971): egy mondathalmaz kielégíthetőségi vizsgálata **NP-teljes**
(de nem minden ítéletlogikai következtetés $O(2^n)$ időt igényel! Ld. előbb)

Ítéletlogika monoton

Logika **monoton:** amikor új mondatokat adunk hozzá a TB-hoz, minden korábban maga után vonzott mondata az eredeti TB-nak továbbra is vonzata marad az új, nagyobb tudásbázisnak.

$$\textbf{ha } TB_1 \models a \text{ , akkor } (TB_1 \cup TB_2) \models a$$

Az igaz mondatok száma csak nőni tud! **Jó ez, vagy nem jó?!**

Jó: amit egyszer nehéz volt bebizonyítani, majd „ingyen van”.
Rossz: a változó világ logikai leírása is változik és ami igaz volt, nem biztos, hogy később annak megtartható.
Σ: amíg a probléma (a szükséges absztrakció szintjén) tökéletesen statikus, a monotonitással nyerünk, különben ráfizethetünk.

Horn klózok:

$P_1 \wedge P_2 \ldots \wedge \ldots P_n \rightarrow Q$
$\neg P_1 \vee \neg P_2 \ldots \vee \ldots \neg P_n \vee Q$

- Legfeljebb egy pozitív literál, pontosan egy pozitív literál = határozott klóz
- Egy hasznos mondattípus, amelyre létezik a TB méretében **lineáris idejű** következtetési eljárás.

Két fontos speciális eset:

$P_1 \wedge P_2 \wedge \ldots P_n \rightarrow$ Hamis, ua., mint $\neg P_i \vee \ldots \neg P_n$
Igaz $\rightarrow Q$ ua., mint Q.

Nem minden TB-beli állítás írható fel Horn klózként, de azoknál, amelyeknél ez megtehető: alkalmazzuk Modus Ponens-t, ameddig marad alkalmazható következtetés.

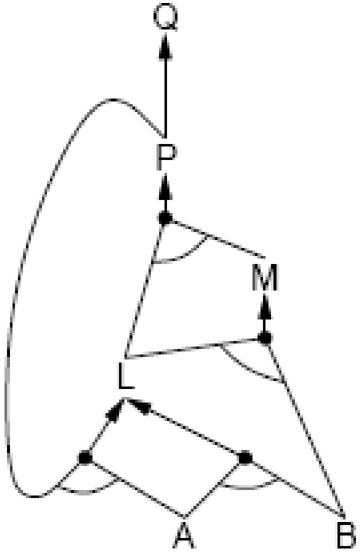
Modus Ponens **teljes** bizonyítási lépés a **Horn klózok** tudásbázisában!

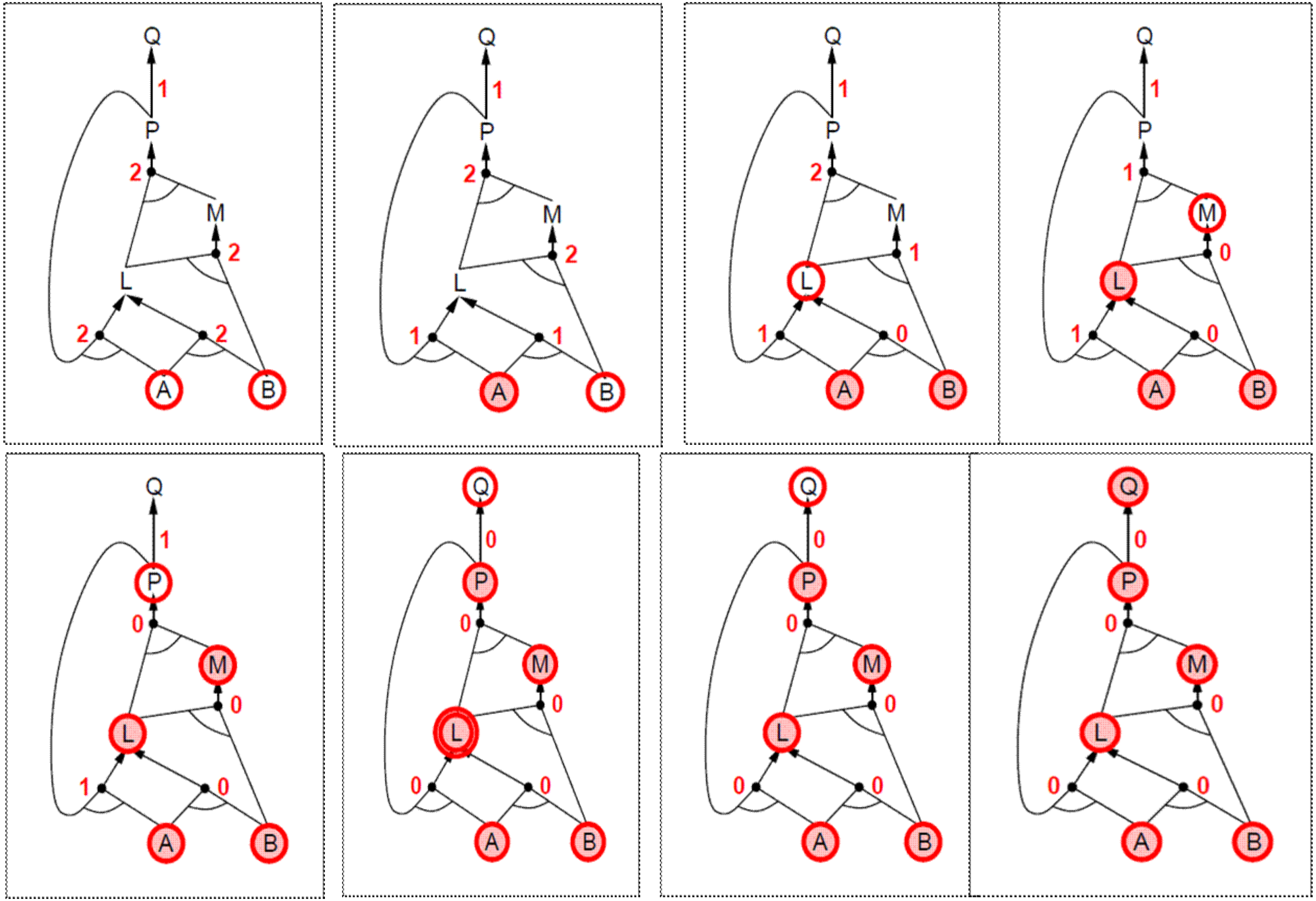
Előrecsatolt, hátracsatolt (Modus Ponens) következtetés

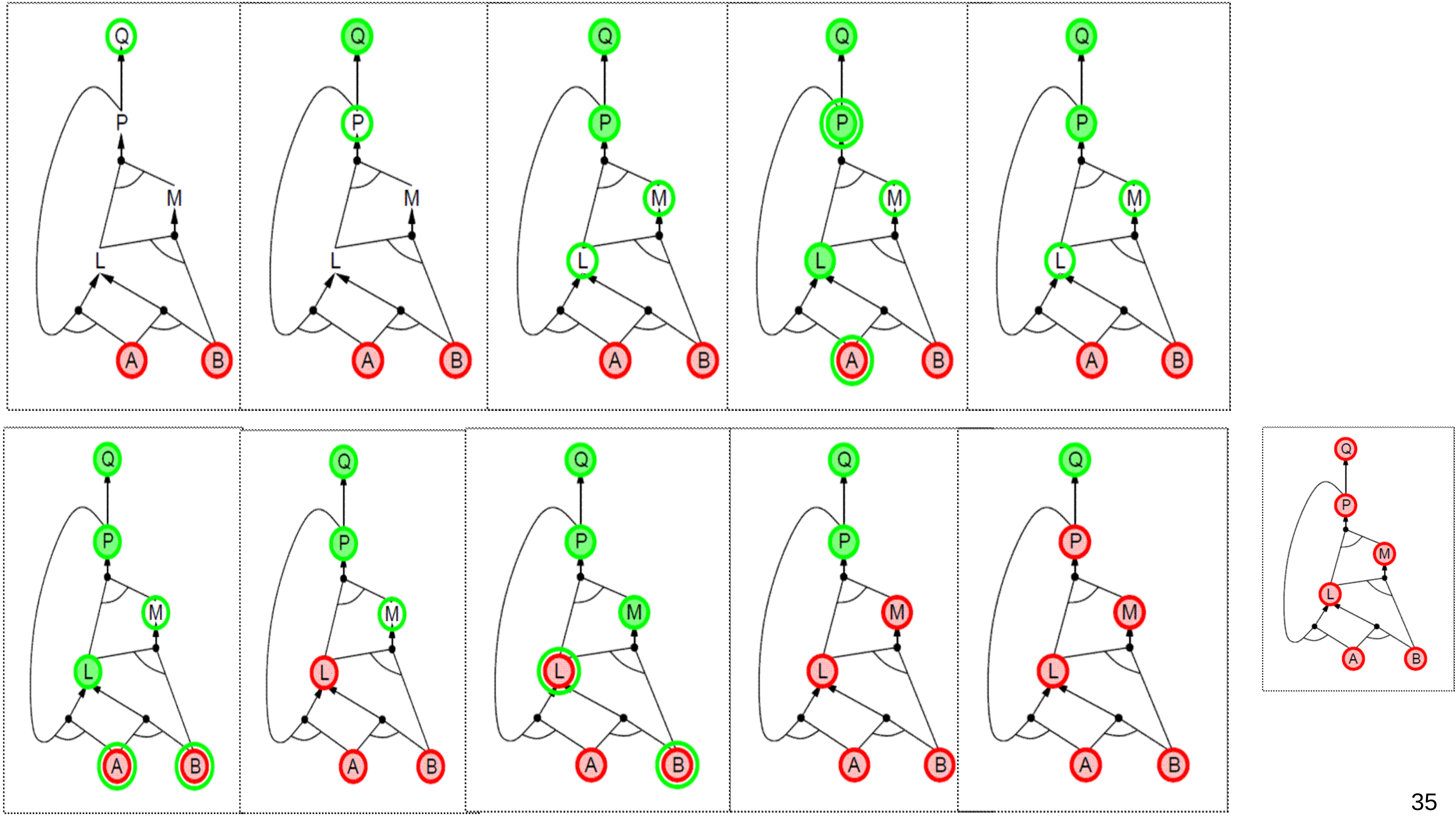
$$P \Rightarrow Q$$
$$L \wedge M \Rightarrow P$$
$$B \wedge L \Rightarrow M$$
$$A \wedge P \Rightarrow L$$
$$A \wedge B \Rightarrow L$$
$$A$$
$$B$$

ECs: Minden szabályt elsütni, melynek premisszája teljesül. Következmény hozzáadása a TB-hoz, amíg a lekérdezendő változó értéket nem kap (fixpont).

HCs: A lekérdezéstől visszafelé:
- ellenőrzés, netán a lekérdezés már igaz,
- elővenni egy szabályt, melynek következménye a lekérdezés és a premisszáit bizonyítani rekurzív Hcs-sal.




Q 1 P 2 M 2 L 2 2 A B
Q 1 P 2 M 2 L 1 1 A B
Q 1 P 2 M 1 L 1 0 A B
Q 1 P 1 M 0 L 1 0 A B
Q 1 P 0 M 0 L 1 0 A B
Q 0 P 0 M 0 L 0 0 A B
Q 0 P 0 M 0 L 0 0 A B
Q 0 P 0 M 0 L 0 0 A B


Q
P
M
L
A
B

Egy ágens a Wumpus világ számára

Tudásbázis ciklus: ágens érzetei → mondatok → tudásbázis
további érvényes mondatok érzet mondatok vonzatai.

tegyük fel:

$B_{1,2}$ = ”bűz van az [1,2]-ben” (de lehetne B12, vagy XYZ)

$S_{1,2}$ = „szellő van az [1,2]-ben", stb.

Tudásbázis: $\neg S_{1,1}$, $\neg B_{1,1}$, $S_{2,1}$, $\neg B_{2,1}$, $\neg S_{1,2}$, $B_{1,2}$, … stb.

amit az ágens tud a környezetéről általánosságban, pl.:

ha valahol nincs bűz, akkor sem ott, sem szomszédban nincs Wumpus.

R_1: $\neg B_{1,1} \rightarrow \neg W_{1,1} \wedge \neg W_{1,2} \wedge \neg W_{2,1}$

R_2: $\neg B_{2,1} \rightarrow \neg W_{1,1} \wedge \neg W_{2,1} \wedge \neg W_{2,2} \wedge \neg W_{3,1}$

R_3: $\neg B_{1,2} \rightarrow \neg W_{1,1} \wedge \neg W_{1,2} \wedge \neg W_{2,2} \wedge \neg W_{1,3}$

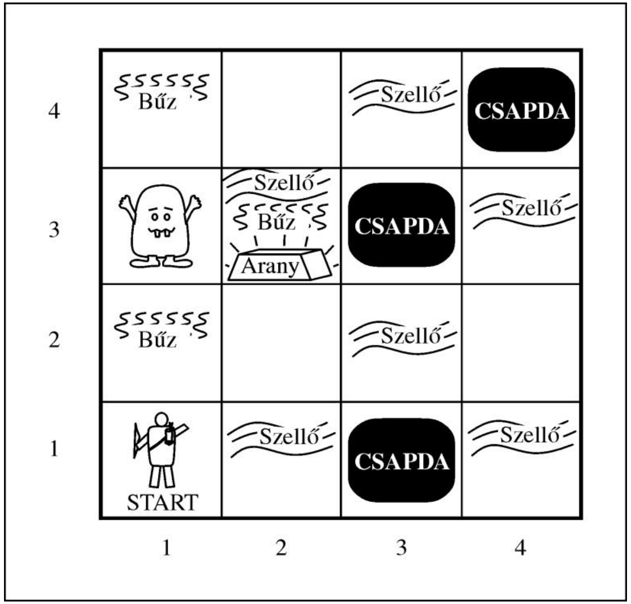
....

ha bűz van az [1,2]-ben, akkor egy Wumpus van ott, vagy egy vagy több szomszédos négyzetben:

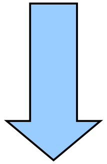
R_4: $B_{1,2} \rightarrow W_{1,3} \vee W_{1,2} \vee W_{2,2} \vee W_{1,1}$

A Wumpus megtalálása - az ágensnek el kellene készítenie a **TB** $\rightarrow$ $\mathbf{W_{1,3}}$ igazságtábláját, hogy megmutassa, hogy ez a mondat **érvényes** (mert akkor a Wumpus(1,3) jelenléte a TB (= ágens tudása) vonzata!).

kb. 12 ítélet szimbólum van: az igazságtáblának 2^{12} = 4096 sora lesz, minden sorban, amelyben a TB mondat igaz, a $\mathbf{W_{1,3}}$ –nak is igaznak kellene lennie. Alkalmazzunk inkább a következtetési mintákat! (de természetes dedukcióval)



$\neg S_{1,1} \neg B_{1,1} \; S_{2,1} \neg B_{2,1} \neg S_{1,2} \; B_{1,2}, \ldots$



$\mathbf{W_{1,3}}$

1. MP: $\neg B_{1,1}$ + R1-re: $\neg \mathbf{W}_{1,1} \land \neg \mathbf{W}_{1,2} \land \neg \mathbf{W}_{2,1}$

$\mathbf{R_1}$: $\neg \mathbf{B}_{1,1} \rightarrow \neg \mathbf{W}_{1,1} \land \neg \mathbf{W}_{1,2} \land \neg \mathbf{W}_{2,1}$

Modus Ponens
(Implikáció eliminálása)

$$\frac{A \rightarrow B \qquad A}{B}$$

$$\frac{\neg \mathbf{B}_{1,1} \rightarrow \neg \mathbf{W}_{1,1} \land \neg \mathbf{W}_{1,2} \land \neg \mathbf{W}_{2,1} \qquad \neg \mathbf{B}_{1,1}}{\neg \mathbf{W}_{1,1} \land \neg \mathbf{W}_{1,2} \land \neg \mathbf{W}_{2,1}}$$

2. És-eliminálás: $\neg \mathbf{W}_{1,1}$, $\neg \mathbf{W}_{1,2}$, $\neg \mathbf{W}_{2,1}$

3. MP: $\neg B_{2,1}$ + R2-re, utána és-elimináció: $\neg \mathbf{W}_{2,2}$, $\neg \mathbf{W}_{2,1}$, $\neg \mathbf{W}_{3,1}$

R_2: $\neg \mathbf{B}_{2,1} \rightarrow \neg \mathbf{W}_{1,1} \wedge \neg \mathbf{W}_{2,1} \wedge \neg \mathbf{W}_{2,2} \wedge \neg \mathbf{W}_{3,1}$

Modus Ponens
(Implikáció eliminálása)

$$\frac{A \rightarrow B \qquad A}{B}$$

$$\frac{\neg \mathbf{B}_{2,1} \rightarrow \neg \mathbf{W}_{1,1} \wedge \neg \mathbf{W}_{2,1} \wedge \neg \mathbf{W}_{2,2} \wedge \neg \mathbf{W}_{3,1} \qquad \neg \mathbf{B}_{2,1}}{\neg \mathbf{W}_{1,1} \wedge \neg \mathbf{W}_{2,1} \wedge \neg \mathbf{W}_{2,2} \wedge \neg \mathbf{W}_{3,1}}$$

4. MP: $B_{1,2}$-re és R4-re: $\mathbf{W_{1,3} \vee W_{1,2} \vee W_{2,2} \vee W_{1,1}}$

R_4: $\mathbf{B_{1,2} \rightarrow W_{1,3} \vee W_{1,2} \vee W_{2,2} \vee W_{1,1}}$

Modus Ponens

(Implikáció eliminálása)

$$\frac{A \rightarrow B \qquad A}{B}$$

$$\frac{\mathbf{B_{1,2} \rightarrow W_{1,3} \vee W_{1,2} \vee W_{2,2} \vee W_{1,1}} \qquad \mathbf{B_{1,2}}}{\mathbf{W_{1,3} \vee W_{1,2} \vee W_{2,2} \vee W_{1,1}}}$$

5. Egység rezolúció: $\mathbf{W_{1,3} \lor W_{1,2} \lor W_{2,2} \lor W_{1,1}}$ és $\mathbf{\neg W_{1,1}}$

Elemi (egység)**rezolúció**

$$\frac{A \lor B \qquad \neg B}{A}$$

$$\frac{\mathbf{W_{1,3} \lor W_{1,2} \lor W_{2,2} \lor W_{1,1}} \qquad \mathbf{\neg W_{1,1}}}{\mathbf{W_{1,3} \lor W_{1,2} \lor W_{2,2}}}$$

az eredmény: $\mathbf{W_{1,3} \lor W_{1,2} \lor W_{2,2}}$

6. Egység rezolúció: $\mathbf{W_{1,3} \lor W_{1,2} \lor W_{2,2}}$ és $\mathbf{\neg W_{2,2}}$
és az eredmény: $\mathbf{W_{1,3} \lor W_{1,2}}$

7. Végül, mégy egy rezolúció: $\mathbf{W_{1,3} \lor W_{1,2}}$ és $\mathbf{\neg W_{1,2}}$
és az eredmény: $\mathbf{W_{1,3}}$

1. MP: $\neg B_{1,1}$ + R1-re: $\neg \mathbf{W}_{1,1} \wedge \neg \mathbf{W}_{1,2} \wedge \neg \mathbf{W}_{2,1}$

2. És-eliminálás: $\neg \mathbf{W}_{1,1}$, $\neg \mathbf{W}_{1,2}$, $\neg \mathbf{W}_{2,1}$

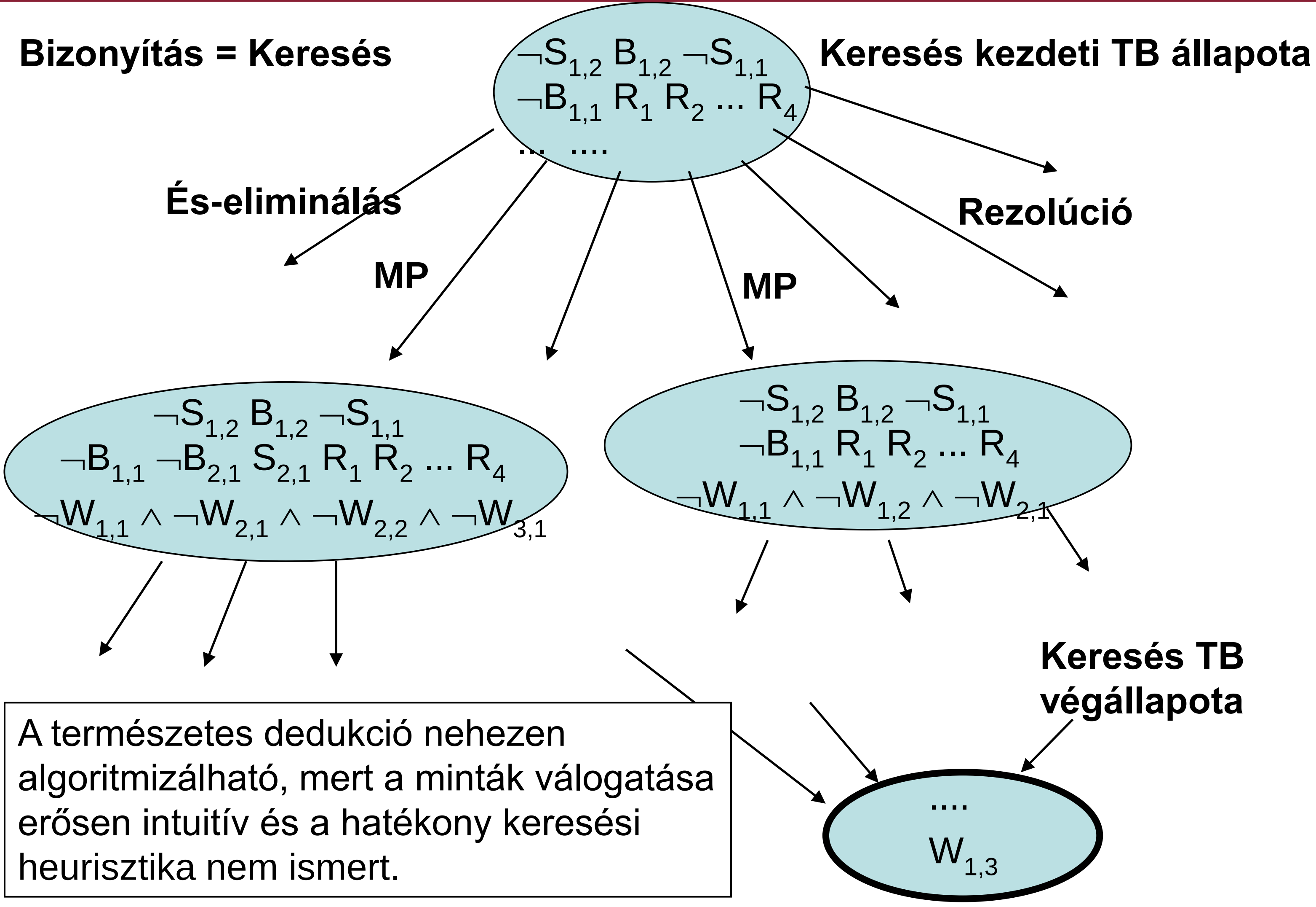
3. MP: $\neg B_{2,1}$ + R2-re, utána és-elimináció: $\neg \mathbf{W}_{2,2}$, $\neg \mathbf{W}_{2,1}$, $\neg \mathbf{W}_{3,1}$

4. MP: $B_{1,2}$-re és R4-re: $\mathbf{W}_{1,3} \vee \mathbf{W}_{1,2} \vee \mathbf{W}_{2,2} \vee \mathbf{W}_{1,1}$

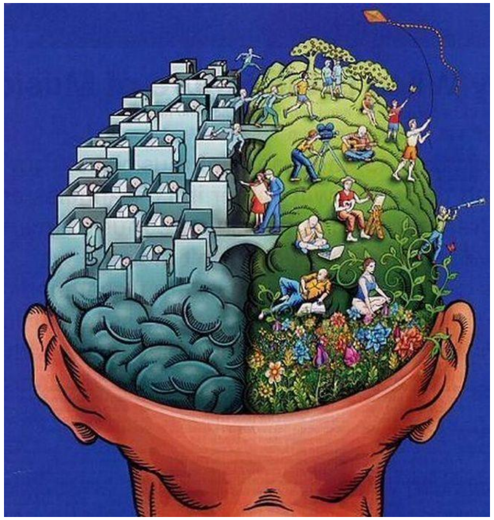
5. Egység rezolúció: $\mathbf{W}_{1,3} \vee \mathbf{W}_{1,2} \vee \mathbf{W}_{2,2} \vee \mathbf{W}_{1,1}$ és $\neg \mathbf{W}_{1,1}$
   az eredmény: $\mathbf{W}_{1,3} \vee \mathbf{W}_{1,2} \vee \mathbf{W}_{2,2}$

6. Egység rezolúció: $\mathbf{W}_{1,3} \vee \mathbf{W}_{1,2} \vee \mathbf{W}_{2,2}$ és $\neg \mathbf{W}_{2,2}$
   és az eredmény: $\mathbf{W}_{1,3} \vee \mathbf{W}_{1,2}$

7. Végül, mégy egy rezolúció: $\mathbf{W}_{1,3} \vee \mathbf{W}_{1,2}$ és $\neg \mathbf{W}_{1,2}$
   és megjelent a kívánt válasz: $\mathbf{W}_{1,3}$
   azaz a Wumpus tényleg az [1,3]-ban van.



A természetes dedukció nehezen algoritmizálható, mert a minták válogatása erősen intuitív és a hatékony keresési heurisztika nem ismert.



Mesterséges Intelligencia - MI

Logikai ágens, elsőrendű logika

Előadó:
Dr. Hullám Gábor

Előadás anyaga:
Dr. Dobrowiecki Tadeusz
Dr. Hullám Gábor

Rezolúció (1963) **Q mondat bizonyítása TB alapján**

Rezolúció „cáfolat-teljes” lépés, garantáltan megtalálja az ellentmondást.
Teljes kereséssel párosítva teljes bizonyítás.

$$\begin{array}{c} B \vee A \\ \neg B \vee G \\ \hline A \vee G \end{array}$$

1. TB átírása klóz formába. (vigyázz, nem Horn-klózok!).

2. Q kérdés negálása. Negált Q kérdés átírása klóz formára.

$$\begin{array}{c} B \\ \neg B \\ \hline \emptyset \end{array}$$

3. Kiterjesztett TB’ = TB U ¬Q tudásbázis létesítése.

4. Rezolúciós lépés ciklikus elvégzése:
   - kilépés üres rezolvensre, akkor TB’ egy ellentmondás, tehát mivel TB igaz, Q igaz kell, hogy legyen.

Rezolúció (1963) **Q mondat bizonyítása TB alapján**

Transzformáció klóz formára:

1. Ekvivalencia elhagyása: $A \Leftrightarrow B = (A \rightarrow B) \wedge (B \rightarrow A)$

2. Implikáció elhagyása: $A \rightarrow B = \neg A \vee B$

3. Negálás atomi formulák szintjére (de Morgan): $\neg (A \vee B) = \neg A \wedge \neg B$

4. Diszjunkciók literálok szintjére (disztributivítás):
$(A \wedge B) \vee C = (A \vee C) \wedge (B \vee C)$ (CNF, Davis, 1960)

5. Konjunkciók elhagyása.
Bontás diszjunktív klózokra (csak $\neg$ és $\vee$ marad)

Példa: $\mathbf{S \vee T \rightarrow Q}$
$\neg (S \vee T) \vee Q$
$(\neg S \wedge \neg T) \vee Q$
$(\neg S \vee Q) \wedge (\neg T \vee Q)$
$\mathbf{\neg S \vee Q}$
$\mathbf{\neg T \vee Q}$

Az eredeti (redundáns) állításforma és a (redundancia mentes) klózforma logikailag ekvivalensek!

TB=axiómák TB’

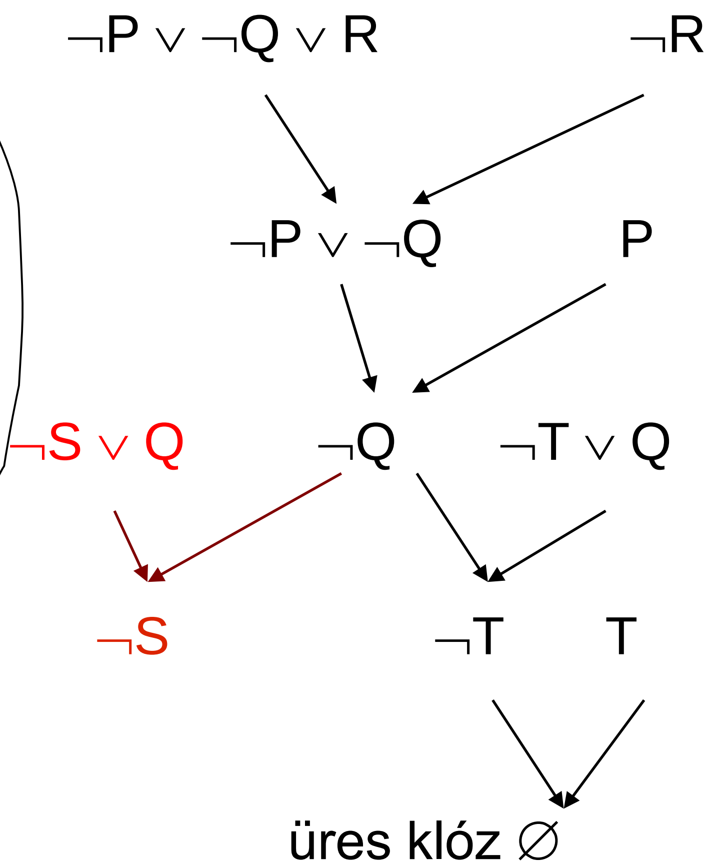
eredeti állítások	klózok
P	P
$P \wedge Q \rightarrow R$	$\neg P \vee \neg Q \vee R$
$S \vee T \rightarrow Q$	$\neg S \vee Q$
	$\neg T \vee Q$
T	T
	$\neg R$

R igaz-e ?

TB |- R ?

TB’ |- $\varnothing$?

rezolúció menete



Egyszerű rezolúciós bizonyítás ítéletkalkulusban

Problémák az ítéletlogikai ágenssel

Általában: - **túl sok ítéletszimbólumot kell kezelni**.
- nehézkes a **változások kezelése**.

Emiatt: **lassú a következtetési eljárás**. (igazságtábla mérete)

De sok fontos gyakorlati alkalmazás (kielégíthetőségi vizsgálattal):
– formális modell-ellenőrzés
– automatikus tesztminta-generálás
– MI tervkészítés
– automatikus tételbizonyítás
– szoftver-verifikálás

Pl. Ipari processzor-verifikáció: 14 ciklusra terjedő viselkedés
kb. 1 millió változó, 30 millió literál, 4 millió klóz, 1.5 millió döntés,
3h futási idő

Elsőrendű logika: milyen a világ valójában?

Objektumok: más objektumoktól megkülönböztetett identitású, **tulajdonságokkal** rendelkező dolgok

Relációk: objektumok közötti kapcsolatok
- n-elemű
- egyelemű (unáris): tulajdonságok

Relációk közül néhány **függvény** – csak egy „érték” egy adott „bemenetre”.

Pl.: **objektumok**: emberek, elméletek, színek, sakkjátszmák, …
relációk: testvére, nagyobb mint, belseje, része, színe, birtokol …
tulajdonságok: piros, kerek, színlelt, páratlan, sokemeletes….
függvények: apja, legjobb barátja, eggyel több mint ….

Ezeket mind külön-külön tudjuk majd ábrázolni és velük következtetni!
(erősen strukturált tudás erősen strukturált ábrázolása jobban megy)

Elsőrendű logika

Szintaktika és szemantika

konstans szimbólumok: X, János, Bodri, K21 (a világ objektumai).

predikátum szimbólumok: kerek, bátyja,… (egy bizonyos reláció).
kerek(X) **bátyja(József, Anna)**

függvény szimbólumok: koszinusz, apja, bal-lába (a relációban szereplő objektum pontosan egy másik objektummal van kapcsolatban).

János = apja(Béla) **apja(János, Béla) = Igaz**

Építőelemek

term: János, Bodri, bal-lába(apja(János)), …, x, f(x)
egy objektumra vonatkozó kifejezés. (konstans a legegyszerűbb term)

atom: bátyja(Géza,Árpád), házas(apja(Richárd),anyja(János)),

atom(i mondat): predikátum szimbólum és az argumentumait jelentő termek
tényeket fejez ki (van igazságértéke).

összetett mondat: bátyja(Misi,János) $\rightarrow$ házas(apja(Misi),anyja(János))

logikai összekötő szimbólumok $\neg \wedge \vee \rightarrow \Leftrightarrow$
a még összetettebb mondatok építésére.

Építőelemek

kvantorok: tulajdonságok, amelyek objektumok halmazaira vonatkoznak, ahelyett hogy megneveznénk minden objektumot a nevével.

Az elsőrendű logika standard kvantorai:

univerzális kvantor ($\forall$) $\forall$x. macska(x) $\rightarrow$ emlős(x)

egzisztenciális kvantor ($\exists$) $\exists$ x. macska(x) $\wedge$ úszik(x)

Egyenlőség**: (term1 = term2)** igaz adott interpretáció mellett, a.cs.a, ha term1 és term2 ugyanarra az objektumra vonatkoznak.

Az $\forall$ és az $\exists$ kapcsolata: $\forall$ x $\neg$ szeret(x, Répa) = $\neg$ $\exists$ x szeret(x, Répa)
$\forall$ x szeret(x, Fagylalt) = $\neg$ $\exists$ x $\neg$ szeret(x, Fagylalt)

Konstans-, predikátum- és függvényszimbólumok megválasztása teljes mértékben a felhasználón múlik. **Ez jó is, de nem is**. De vajon miért?

Fizikai világ: Objektum: Bodri kutya, amely
Tulajdonsága: nagy

Konstans: Bodri
Predikátum:
nagytestű(x)
Egy igaz állítás:
nagytestű(Bodri)

Konstans: Kutya1
Predikátum:
dögnagy(x)
Egy igaz állítás:
dögnagy(Kutya1)

Igaz logikában, hogy: nagytestű(Bodri) ⇔ dögnagy(Kutya1) **Nem!**
Hacsak nem mondjuk ki, hogy: ∀x nagytestű(x) ⇔ dögnagy(x).
Bodri = Kutya1.

azonos jelölés más interpretációt is takarhat
azonos interpretáció más jelöléshez is vezethet

(egy állítás kielégíthető, ha adott interpretációban létezik olyan modell, amiben igaz)
(egy állítás igaz, ha az előbb említett modell a világ jelenlegi állapotát adja meg)

Bizonyítások

Modell alapú következtetés?

Adott TB-hoz felsorolni a modelleket azt jelenti, hogy a modelleket meg kell fogalmazni:

a problématerület objektumainak minden n = 1 ... ∞ száma mellett,

a szókészlet minden k attribútumú *p(x1, ..., xk)* predikátumra,

a szókészlet minden n elemű k-náris relációra,

a szókészlet minden C logikai konstansra,

a C logikai konstans minden problématerület objektum hozzárendelésére n db. objektumból (interpretációk)…

T.i. nem tudjuk, mely kombináció lesz jó?

A legjobb (legegyszerűbb) esetben is legalább **megszámlálható**.

A függvények (relációk) ágyazása a termekben a lényegi nehézség.

Redukció ítéletlogikai következtetésre

Következtetési lépések kiterjesztése

Redukció ítéletlogikai következtetésre

Egy változó nélküli term - **alapterm** (**grounding**)
Változók nélküli predikátum kalkulus = ítélet kalkulus!

kutya(Bodri)	$X1$
nagytestű(Bodri)	$X2$
kutya(Bodri) $\wedge$ *nagytestű(Bodri)* $\rightarrow$ *fél(Béla,Bodri)*	$X1 \wedge X2 \rightarrow X3$

Új TB: minden univerzális mondat minden lehetséges példányosítása.
Egy alapmondat vonzata az új TB-nak, ha vonzata volt az eredeti TB-nak.

Probléma: ha függvények is vannak, akkor ∞-sok alapterm létezik,
pl. *apja(apja ... (apja(János)))*

Herbrand (1930): Ha egy mondat egy FOL TB vonzata, akkor vonzata a példányosított TB egy **véges** részhalmazának is.
Eljárás: minden n = 0, 1, 2, ... példányosított TB létesítése n-mélységű termekkel. A kérdéses mondat vonzata-e?
Probléma: működik jól, ha a vonzat létezik, ∞-hurok, ha nincs.

Predikátum kalkulus tulajdonságai

Teljes: **minden igaz állítás belátható (bebizonyítható)**
(1930, Gödel, egzisztenciális bizonyítás, Herbrand, ítéletlogikai redukció)

A vonzat csak félig eldönthető: **állítás hamis volta nem mutatható ki!**
Turing (1936), **Church** (1936):

- a bizonyítási eljárásnak **igaz állítás** esetén van kilépési pontja,
- **hamis állítás** esetén **nincs** kilépési pontja, és munka közben, logikai alapon, **nem dönthető el**, hogy melyik esettel foglalkozunk.

Gyakorlati következtetés: gépen egy állításhalmaz konzisztenciája elvben nem mutatható ki a **véges erőforrások** (**idő**) miatt!

Predikátum kalkulus tulajdonságai

Teljes: **minden igaz állítás belátható (bebizonyítható)**

A vonzat csak félig eldönthető: **állítás hamis volta nem mutatható ki!**

Gyakorlati következtetés: gépen egy állításhalmaz konzisztenciája elvben nem mutatható ki a **véges erőforrások** (**idő**) miatt!

Az elmélet és az axiómák: matematika és MI

axiómák = TB	kérdés	bizonyítás	és utána?
matematika minimális, redundancia-mentes	érdekes tétel	hosszú	dicsőség
MI maximális, redundáns	ágens feladata	minél rövidebb	cselekvés végrehajtása

Következtetési lépések kibővítése

Modus Ponens:

$$\frac{A \qquad A \rightarrow B}{B} \qquad\qquad \frac{p(A) \qquad \forall x,\ p(x) \rightarrow q(x)}{q(A)}$$

x/A illesztés = **unifikálás (egyesítés)** + **behelyettesítés**

Unifikál(p, q) = θ, ha $p\theta = q\theta$

p	q	θ
ismer(János, x)	ismer(János, Ági)	{x ⁄ Ági}
ismer(János, x)	ismer(y, BME)	{x ⁄ BME, y ⁄ János}
ismer(János, x)	ismer(y, anya(y))	{y ⁄ János, x ⁄ anya(János)}
ismer(János, x)	ismer(x, BME)	kudarc

de további lehetőségek is vannak:

Univerzális kvantor eliminálása:

$$\frac{\forall x,\ p(x, A)}{p(B, A)}$$

Egzisztenciális kvantor eliminálása:

(un. skolemizálás)

$$\frac{\exists x\; q(x, A)}{q(B_S, A)}$$

feltéve, hogy B_S-nek másutt szerepe a TB-ban nincs!

B_S az un. **Skolem konstans**, bizonyos tulajdonságokkal igen, de a **feladatban önálló (interpretált) léttel NEM rendelkező objektum**.

Lehet f(x) **Skolem függvény** is, amely minden *x*-hez egy külön Skolem konstanst rendel hozzá …

$$\frac{\forall x\; \exists y\; q(x, y)}{\forall x\;\; q(x, f_S(x))}$$

$$\frac{\forall ember\; \exists szív.\; van\text{-}szíve(ember, szív)}{\forall\; ember\; van\text{-}szíve(ember, f_{Szív}(ember\;))}$$

$$\frac{\forall x\; p(x, A)}{p(B, A)}$$

ezzel szemben az univerzális kvantor eliminálásánál akármilyen létező objektumot megjelölő konstanst lehetett használni!

Gépi bizonyítás kérdése

Modus Ponens alapú bizonyítás **nem teljes**, de 1' rendű logika **teljes**, avagy igaz tételek bizonyítása létezik (de hogyan?).

Az első jól algoritmizált „hogyan” a **rezolúció** (Robinson, 1963), de a félig eldönthetőség, mint probléma megmarad!

Rezolúció predikátum kalkulusban:

$$\begin{array}{ll} & q(x) \vee p(x, \text{Atti}) \\ & \underline{\neg q(\text{Béla}) \vee r(x)} \\ x/\text{Béla} \qquad & p(\text{Béla}, \text{Atti}) \vee r(\text{Béla}) \end{array}$$

Gépi bizonyítás kérdése

Rezolúció predikátum kalkulusban: $q(x) \vee p(x, \text{Atti})$

$\underline{\neg q(\text{Béla}) \vee r(x)}$

x/Béla $p(\text{Béla}, \text{Atti}) \vee r(\text{Béla})$

Egyesítés (unifikálás) $q(x)$ és $\neg q(y)$, ha x/y

ez viszont akkor lehetséges, ha:

x	y
változó-1	változó-2
változó	konstans
konstans	változó

nem megy azonban, ha:

konstans-1	konstans-2
változó	f(változó)

Transzformáció klóz formára

1. Ekvivalenciát eltüntetni: $A \Leftrightarrow B = (A \rightarrow B) \wedge (B \rightarrow A)$

2. Implikációt eltüntetni: $A \rightarrow B = \neg A \vee B$

3. Negálást az atomi formulák szintjére áthelyezni.

$\neg (A \vee B) = \neg A \wedge \neg B, \quad \neg \forall x\, p(x) = \exists x\, \neg p(x)$

4. Egzisztenciális kvantorokat eltüntetni: Skolemizálás

5. Ha szükséges, a változókat átnevezni.

$\forall x\, p(x) \vee \forall x\, q(x)$ --> $\forall x\, p(x) \vee \forall y\, q(y)$

6. Univerzális kvantorokat balra kihelyezni.

$....\forall x....\forall y.. = \forall x \forall y\, ...x...y..$

A cél:
az eredeti állításforma
és a klózforma
logikailag ekvivalensek!

7. Diszjunkciókat literál szintjére áthelyezni.

$(A \wedge B) \vee C = (A \vee C) \wedge (B \vee C)$ **konjunktív normal forma** (CNF, Davis, 1960)

8. Konjunkciókat eliminálni. Bontás diszjunktív klózokra

9. Ha szükséges, a változókat átnevezni.

10. Univerzális kvantorokat elhagyni. **ami marad: $\neg$, $\vee$, és most?**

Példa: Valaki mindig szereti azt, aki minden állatot szeret.

$\forall$ **x [$\forall$ y állat(y) $\rightarrow$ szeret(x, y)] $\rightarrow$ [$\exists$ y szeret(y, x)]**

$\forall$ x $\neg$ [$\forall$ y $\neg$állat(y) $\vee$ szeret(x, y)] $\vee$ [$\exists$ y szeret(y, x)]

$\forall$ x [$\exists$ y $\neg$ ($\neg$állat(y) $\vee$ szeret(x, y))] $\vee$ [$\exists$ y szeret(y, x)]

$\forall$ x [$\exists$ y $\neg$ $\neg$ állat(y) $\wedge$ $\neg$ szeret(x, y)] $\vee$ [$\exists$ y szeret(y, x)]

$\forall$ x [$\exists$ y állat(y) $\wedge$ $\neg$ szeret(x, y)] $\vee$ [$\exists$ **y** szeret(y, x)]

$\forall$ x [$\exists$ **y** állat(y) $\wedge$ $\neg$ szeret(x, y)] $\vee$ [$\exists$ **z** szeret(z, x)]

$\forall$ **x** [állat(F(x)) $\wedge$ $\neg$ szeret(x, F(x))] $\vee$ szeret(G(x), x)

[állat(F(x)) $\wedge$ $\neg$ szeret(x, F(x))] $\vee$ szeret(G(x), x)

[állat(F(x)) $\vee$ szeret(G(x), x)] $\wedge$ [$\neg$ szeret(x, F(x)) $\vee$ szeret(G(x), x)]

Egy állításból két (lehet több is) klóz:

a. **állat(F(x))** $\vee$ **szeret(G(x), x)**
b. $\neg$ **szeret(x, F(x))** $\vee$ **szeret(G(x), x)** F(x), G(x) Skolem függvények

Példa: Bárkit, aki megöl egy állatot, senki nem szeret

$\forall$ **x [$\exists$ y állat(y)** $\wedge$ **megöli(x, y)]** $\rightarrow$ **[**$\forall$ z $\neg$ **szeret(z, x)]**

$\forall$ x $\neg$ [$\exists$ y állat(y) $\wedge$ megöli(x, y)] $\vee$ [$\forall$ z $\neg$ szeret(z, x)]

$\forall$ x [$\forall$ y $\neg$ (állat(y) $\wedge$ megöli(x, y)] $\vee$ [$\forall$ z $\neg$ szeret(z, x)]

$\forall$ x ,$\forall$ y $\neg$ állat(y) $\vee$ $\neg$megöli(x, y) $\vee$ [$\forall$ z $\neg$ szeret(z, x)]

$\forall$ x ,$\forall$ y, $\forall$ z $\neg$ állat(y) $\vee$ $\neg$megöli(x, y) $\vee$ $\neg$ szeret(z, x)

$\neg$ állat(y) $\vee$ $\neg$megöli(x, y) $\vee$ $\neg$ szeret(z, x)

Példa: Nobby szereti az összes állatot

$\forall$ **x [állat(x)]** $\rightarrow$ **[szeret(Nobby, x)]**

$\neg$ állat(x) $\vee$ szeret(Nobby, x)

Példa: Vagy Nobby vagy a kíváncsiság ölte meg Greebo-t.
Greebo egy macska

macska(Greebo)
megöli(Nobby, Greebo) $\vee$ **megöli(Kíváncsiság, Greebo)**

Kiegészítés:
$\forall$x macska(x) $\rightarrow$ állat(x)
$\neg$ macska(x) $\vee$ állat(x)

A kíváncsiság ölte-e meg Greebo-t?

Kérdés: **megöli(Kíváncsiság, Greebo)**

$\neg$ Kérdés: $\neg$ **megöli(Kíváncsiság, Greebo)**

1.a) állat(F(x)) ∨ szeret(G(x), x)

1.b) ¬ szeret(x, F(x)) ∨ szeret(G(x), x)

2.) ¬ állat(y) ∨ ¬megöli(x, y) ∨ ¬ szeret(z, x)

3.) ¬ állat(x) ∨ szeret(Nobby, x)

4.) macska(Greebo)

5.) megöli(Nobby, Greebo) ∨ megöli(Kíváncsiság, Greebo)

6.) ¬ macska(x) ∨ állat(x)

7.) ¬ **megöli(Kíváncsiság, Greebo)**

Rezolúciós lépések

8: 7+5

megöli(Nobby, Greebo) ∨ megöli(Kíváncsiság, Greebo)

¬ **megöli(Kíváncsiság, Greebo)**

8: megöli(Nobby, Greebo)

1.a) állat(F(x)) $\vee$ szeret(G(x), x)

1.b) $\neg$ szeret(x, F(x)) $\vee$ szeret(G(x), x)

$\boxed{\text{2.) } \neg \text{ állat(y)} \vee \neg \text{megöli(x, y)} \vee \neg \text{ szeret(z, x)}}$

3.) $\neg$ állat(x) $\vee$ szeret(Nobby, x)

4.) macska(Greebo)

5.) megöli(Nobby, Greebo) $\vee$ megöli(Kíváncsiság, Greebo)

6.) $\neg$ macska(x) $\vee$ állat(x)

7.) $\neg$ megöli(Kíváncsiság, Greebo)

$\boxed{\text{8: megöli(Nobby, Greebo)}}$

9: 8+2

megöli(Nobby, Greebo)

$\neg$ állat(y1) $\vee$ $\neg$megöli(x1, y1) $\vee$ $\neg$ szeret(z1, x1)   x1 / Nobby

---   y1/ Greebo

$\neg$ állat(Greebo) $\vee$ $\neg$ szeret(z1, Nobby)

1.a) állat(F(x)) ∨ szeret(G(x), x)

1.b) ¬ szeret(x, F(x)) ∨ szeret(G(x), x)

2.) ¬ állat(y) ∨ ¬megöli(x, y) ∨ ¬ szeret(z, x)

3.) ¬ állat(x) ∨ szeret(Nobby, x)

4.) macska(Greebo)

5.) megöli(Nobby, Greebo) ∨ megöli(Kíváncsiság, Greebo)

6.) ¬ macska(x) ∨ állat(x)

7.) ¬ megöli(Kíváncsiság, Greebo)

8: megöli(Nobby, Greebo)

9: ¬ állat(Greebo) ∨ ¬ szeret(z, Nobby)

10: 9+6

¬ állat(Greebo) ∨ ¬ szeret(z, Nobby)

¬ macska(x2) ∨ állat(x2) x2 / Greebo

¬ macska(Greebo) ∨ ¬ szeret(z, Nobby)

1.a) állat(F(x)) ∨ szeret(G(x), x)

1.b) ¬ szeret(x, F(x)) ∨ szeret(G(x), x)

2.) ¬ állat(y) ∨ ¬megöli(x, y) ∨ ¬ szeret(z, x)

3.) ¬ állat(x) ∨ szeret(Nobby, x)

4.) macska(Greebo)

5.) megöli(Nobby, Greebo) ∨ megöli(Kíváncsiság, Greebo)

6.) ¬ macska(x) ∨ állat(x)

7.) ¬ megöli(Kíváncsiság, Greebo)

8: megöli(Nobby, Greebo)

9: ¬ állat(Greebo) ∨ ¬ szeret(z, Nobby)

10: ¬ macska(Greebo) ∨ ¬ szeret(z, Nobby)

11: 10+4

¬ macska(Greebo) ∨ ¬ szeret(z1, Nobby)

macska(Greebo)

¬ szeret(z1, Nobby)

1.a) állat(F(x)) $\vee$ szeret(G(x), x)

$\boxed{\text{1.b) } \neg \text{ szeret(x, F(x))} \vee \text{szeret(G(x), x)}}$

2.) $\neg$ állat(y) $\vee$ $\neg$megöli(x, y) $\vee$ $\neg$ szeret(z, x)

$\boxed{\text{3.) } \neg \text{ állat(x3)} \vee \text{szeret(Nobby, x3)}}$

11: $\neg$szeret(z1, Nobby)

12: 1b+3

$\neg$ állat(x3) $\vee$ $\boxed{\text{szeret(Nobby, x3)}}$

$\boxed{\neg \text{ szeret(x4, F(x4)}}$) $\vee$ szeret(G(x4), x4) x4 / Nobby

----

$\neg$ állat(F(Nobby)) $\vee$ szeret(G(Nobby), Nobby)

$\boxed{\text{1.a) állat(F(x))} \vee \text{szeret(G(x), x)}}$

1.b) $\neg$ szeret(x, F(x)) $\vee$ szeret(G(x), x)

2.) $\neg$ állat(y) $\vee$ $\neg$megöli(x, y) $\vee$ $\neg$ szeret(z, x)

3.) $\neg$ állat(x3) $\vee$ szeret(Nobby, x3)

11: $\neg$szeret(z1, Nobby)

$\boxed{\text{12: } \neg \text{ állat(F(Nobby))} \vee \text{szeret(G(Nobby), Nobby)}}$

13: 1a+12

$\boxed{\neg \text{ állat(F(Nobby))}}$ $\vee$ szeret(G(Nobby), Nobby)

$\boxed{\text{állat(F(x5))}}$ $\vee$ szeret(G(x5), x5)   x5 / Nobby

----

szeret(G(Nobby), Nobby)

1.a) állat(F(x)) ∨ szeret(G(x), x)
1.b) ¬ szeret(x, F(x)) ∨ szeret(G(x), x)
2.) ¬ állat(y) ∨ ¬megöli(x, y) ∨ ¬ szeret(z, x)
3.) ¬ állat(x3) ∨ szeret(Nobby, x3)

11: ¬szeret(z1, Nobby)

12: ¬ állat(F(Nobby)) ∨ szeret(G(Nobby), Nobby)

13: szeret(G(Nobby), Nobby)

13: 1a+12

szeret(G(Nobby), Nobby)

¬szeret(z1, Nobby) z1 / G(Nobby)

ϕ

Üres rezolvensre jutottunk -> negált kérdés hamis

tehát az eredeti kérdés igaz: **megöli(Kíváncsiság, Greebo)**

Formális

Informális

Rezolúciós bizonyítás procedúrája

Adott állítások halmaza F, a bizonyítandó állítás Q

1. Az F halmaz összes állítását konvertáljuk klóz formába F'.

2. Negáljuk a Q-t és konvertáljuk klóz formába. Adjuk hozzá az F‘-hez.

3. Ismételjük az alábbi ciklust, amíg:
   (a) **ellentmondásra** rá nem futunk,
   (b) az **előrehaladást** már nem tapasztaljuk, vagy
   (c) az erőforrások előre meghatározott mennyiségét ki nem használjuk:

4. **Válasszunk meg** két klózt és alkalmazzunk **rezolúciós** lépést.
5. Ha a rezolvens **üres**, megvan az ellentmondás. Ha nem, adjuk hozza a többi klózhoz és folytatjuk (3.)

Rezolúciós bizonyítás procedúrája

Formális

Informális

…

4. **Válasszunk meg** két klózt és alkalmazzunk **rezolúciós** lépést.

…

Rezolúciós stratégiák (klózok kiválasztási heurisztikái)

1. Egység rezolúció. **Horn-klóz alakú TB-ban az eljárás teljes**, különben nem!

2. 'Set of Support': (egy klóz 'S-of-S'-ból és egy 'külső' klóz), rezolvens vissza 'S-of-S'-ba. **Teljes, ha 'S-of-S'-n kívüli klózok teljesíthetők,** gyakorlatban: 'S-of-S' = a negált kérdés (a többit úgyis elhisszük)

3. Input rezolúció: az egyik klóz mindig az előbbi rezolvens, az első lépésnél viszont a kérdés. **Horn-klóz alakú tudásbázisban az eljárás teljes**, különben nem!

4. Lineáris rezolúció: P és Q rezolválható, ha P benne van az eredeti tudásbázisban, vagy ha P a Q őse a bizonyítási fában. **Lineáris rezolúció egy teljes eljárás.**

Logikai programozás

- Prolog **nem teljes**

Tételbizonyítók

Otter (Organized Techniques for Theorem proving and Effective Research)

rezolúciós bizonyítás támogató halmaz (set of support) stratégiával - **Prover9**

Prolog kiterjesztése – **PTTP Prolog Technology Theorem Prover**

Az elmélet és az axiómák: matematika és MI

axiómák = TB	kérdés	bizonyítás	és utána?	Hol az infó?
Matematika				
minimális redundanciamentes	….	hosszú	pávatoll	üres rezolvens tényében
MI				
maximális redundans	…	minél rövidebb	cselekvés végrehajtása	üres rezolvens tényében **ÉS a behelyettesítésekben**

Ontológia

Cél: A tudás megosztása

- Tárgyterület (domain) részletes leírását valósítja meg

Összetevők:

- Fogalmak
  - osztály
- Tulajdonságok és relációk a fogalmak között
  - taxonómia (fogalmak közötti hierarchia: isA, partOf)
  - reláció, attribútum, szerep (Role)

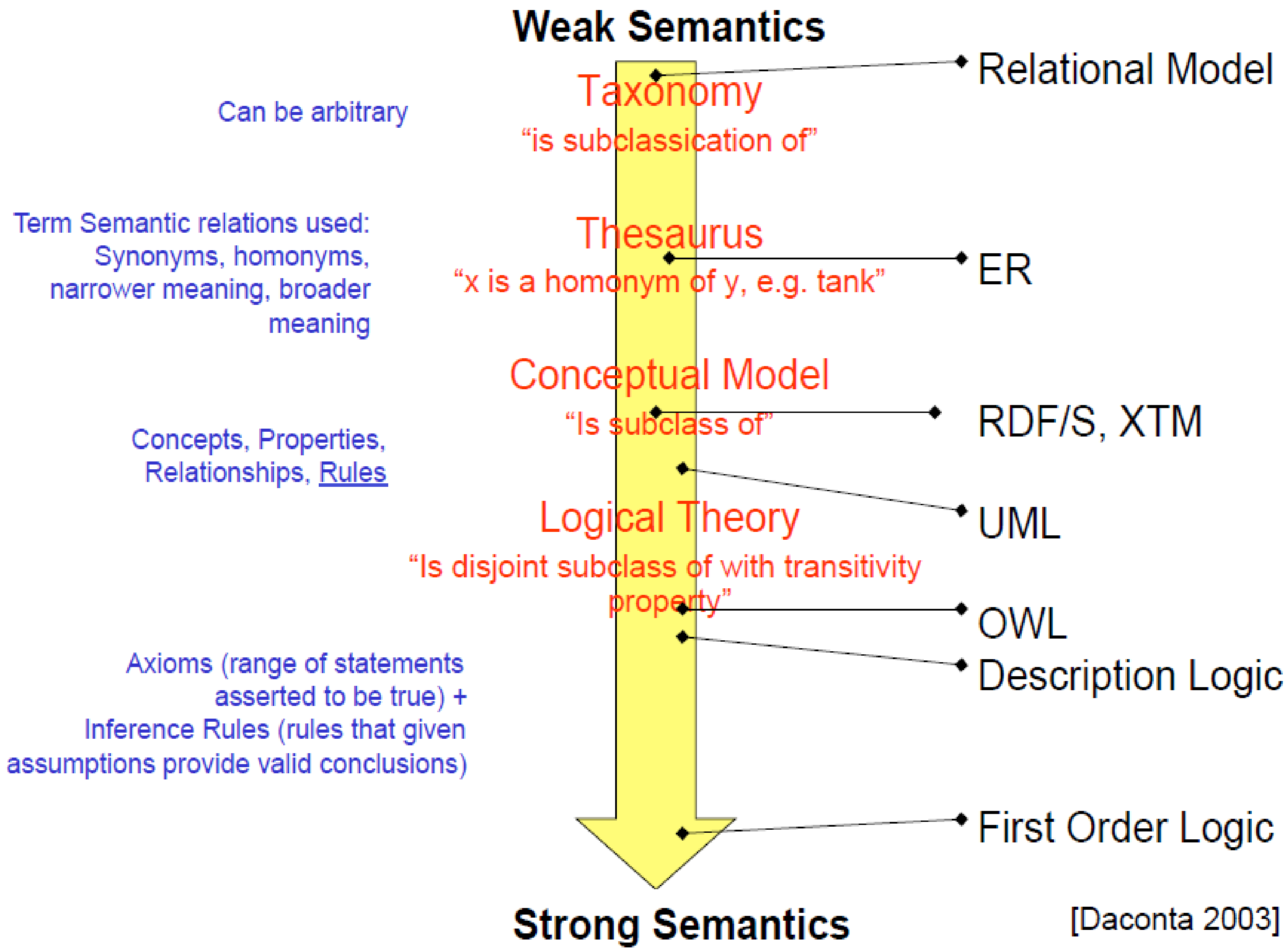
Tárgyterület (domain) részletes leírása

-Megkötések (kényszerek) tulajdonságokra, fogalmakra

- Típus: integer
- Számosság (cardinality): ≥1
- Értéktartomány (range): 0 < Y <= 50
- X nagyobb, mint Y
  - X osztály egyedei nagyobbak, mint Y-éi
- X és Y különböző
  - X és Y osztály diszjunkt, nincs közös egyed

-Konkrét értékek megadása

Ontológiák típusai



O.Conlan: Brief Introduction to Ontology, Trinity College

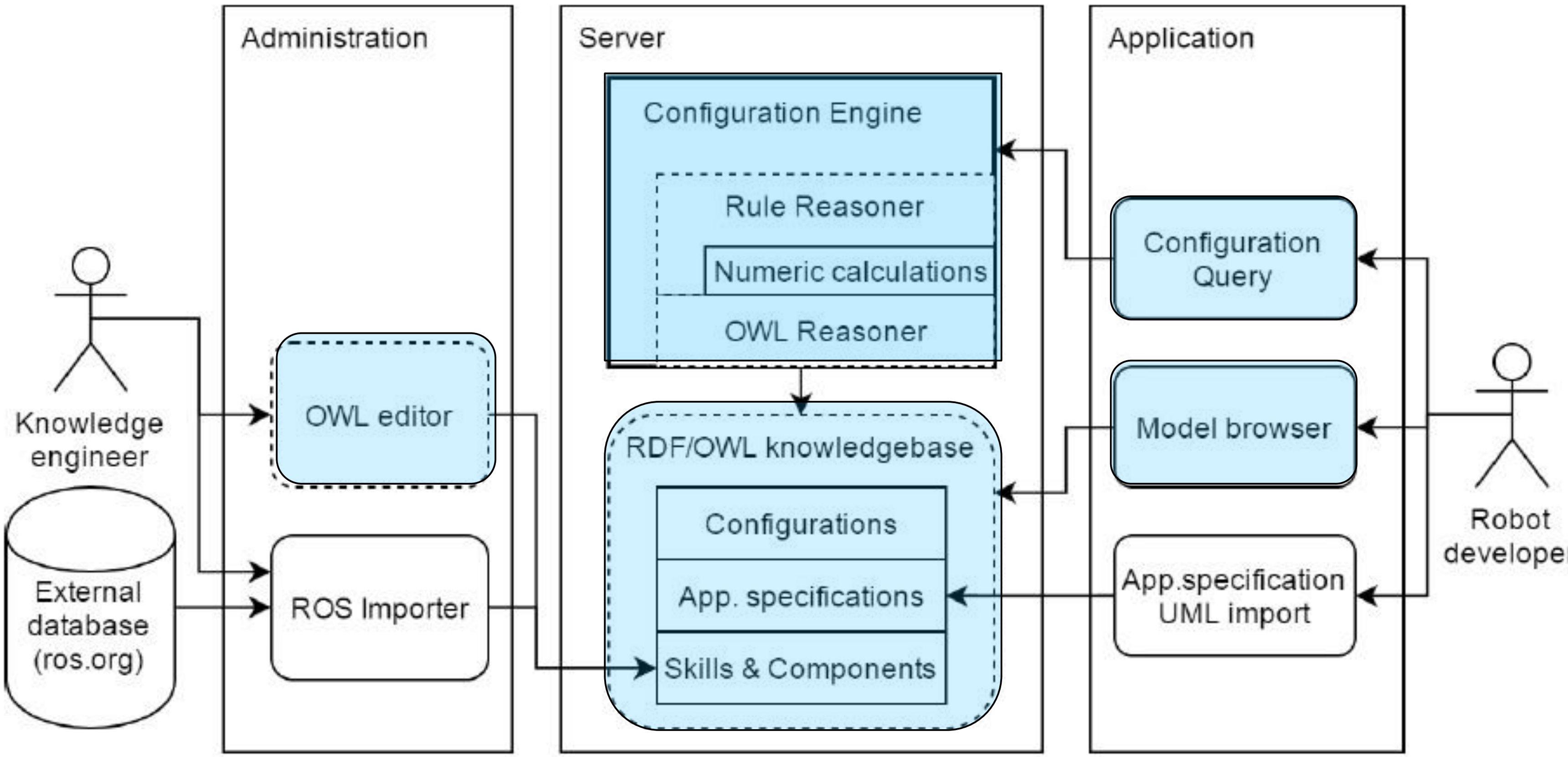
Ontológia vs Relációs séma

- Ontológia:
  - Formálisan definiált kapcsolatok az entitások között
  - Ember és gép által is értelmezhető
- Relációs séma:
  - Relációk implicit kapcsolatot jelentenek entitások között
  - Interpretáció szükséges
  - Adatbázis szemantikájának ismerete nélkül nem használható

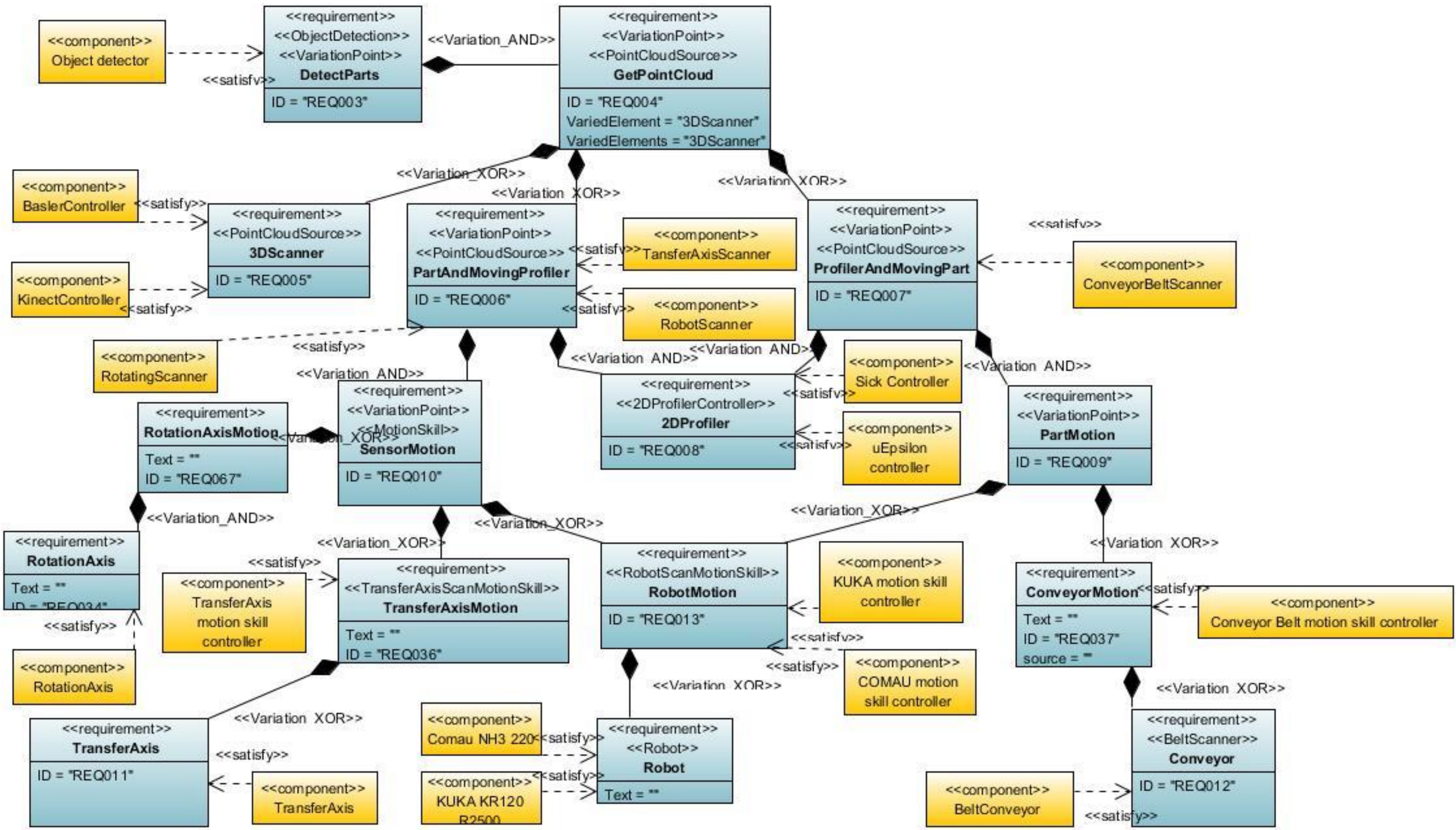
Ontológia vs Tudásbázis

- Ontológia:
  - fogalmak
  - tulajdonságok
  - megkötések
  - értékek
- Tudásbázis:
  - Ontológia
  - példányok

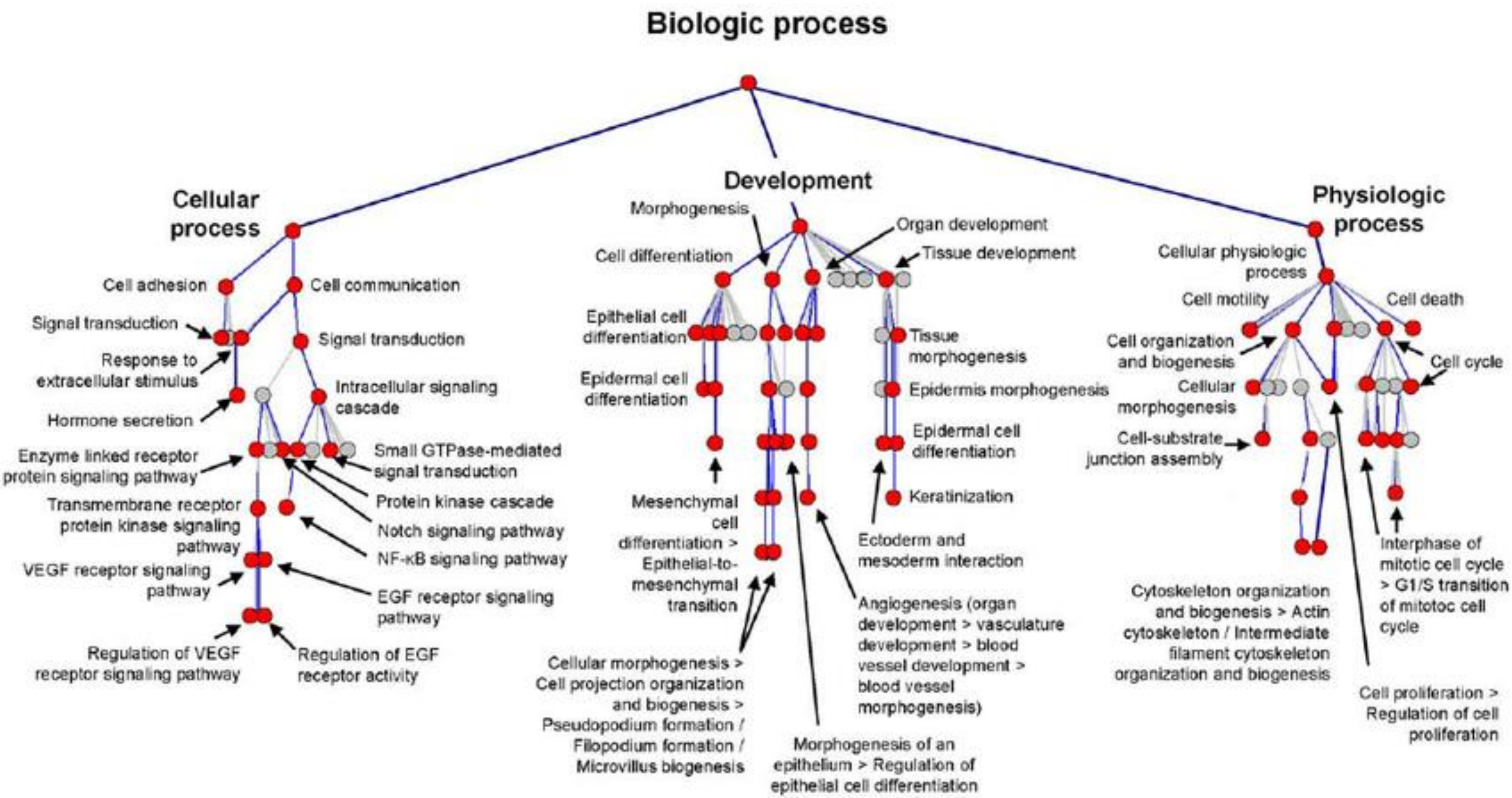
R5COP - Design and development tools



Design and development tools

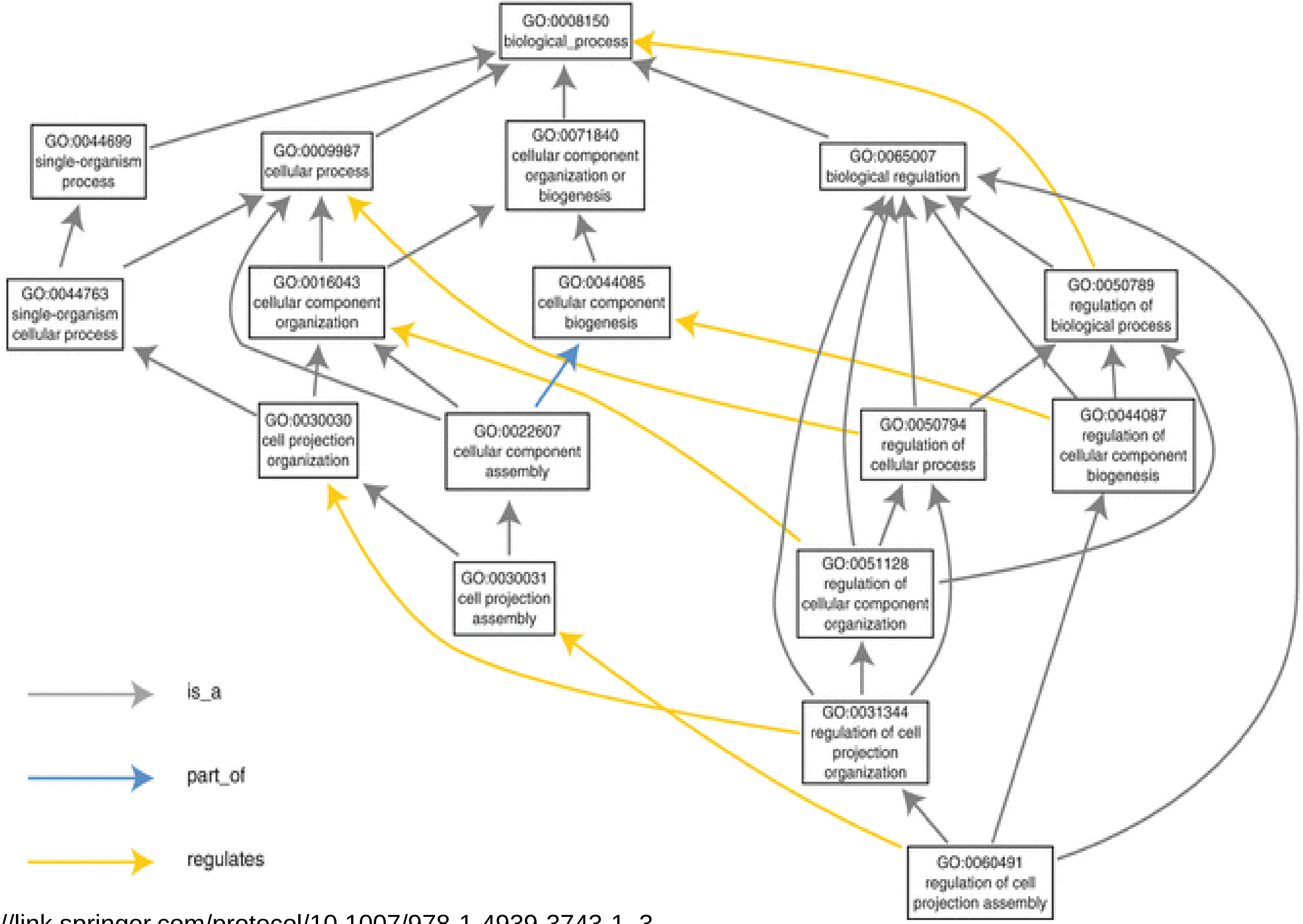


Gene Ontology (GO)



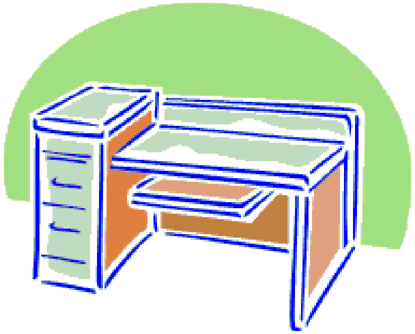
https://link.springer.com/protocol/10.1007/978-1-4939-3743-1_3

Gene Ontology (GO)

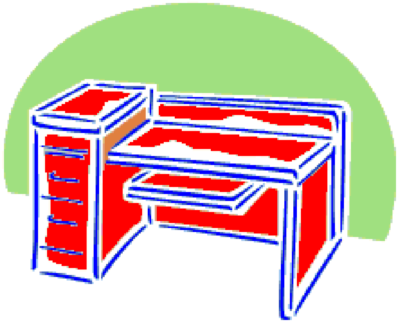


https://link.springer.com/protocol/10.1007/978-1-4939-3743-1_3

A festő ágens problémája



http://technologygads.blogspot
.hu/search/label/robot%20kits



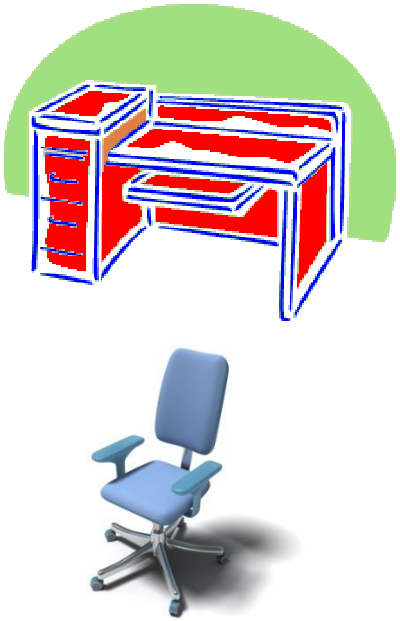
http://www.instructables.com/id/
Pointillist-Painting-Robot-Arm/

Predikátum kalkulus hogyan vethető be a környezetét alakító intelligens rendszer leírására?

Ehhez szükséges:
- Idő múlásának ábrázolása.
- Ágens cselekvéseinek leírása.
- Környezet változása a cselekvés hatására.

A festő ágens problémája

http://technologygads.blogspot
.hu/search/label/robot%20kits

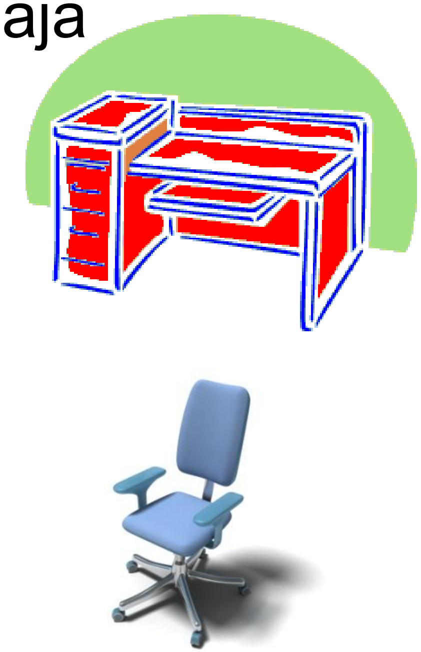


- Az ágens világában egy asztal és egy szék van.
- Ágens bútort kizárólag pirosra festhet.
- Egyik bútor sem piros, de csak az asztalt szeretnénk pirosnak.

Hogyan írjuk le logikával, hogy milyen világgal szembesül az ágens, és milyen világot hagy maga után?

Ráadásul legyen ez a leírás kellően általános is (és értelmesen kezelhető is).

A festő ágens problémája



kék(Szék)
$\neg$ piros(Asztal)

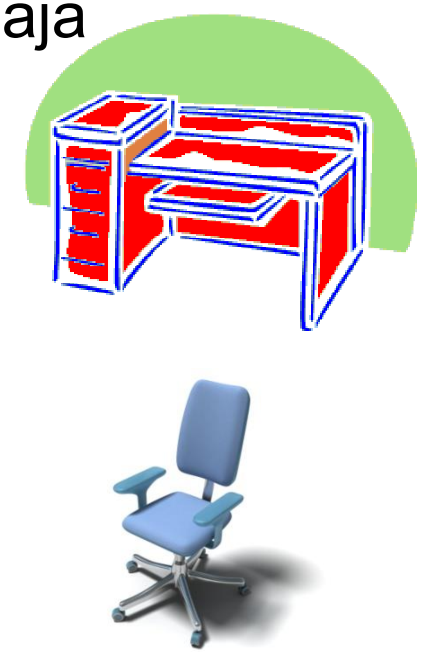
volt?

kék(Szék)
piros(Asztal)

lesz?

Ellentmondás!

A festő ágens problémája



kék(Szék, S1)
¬ piros(Asztal, S1)

kék(Szék, S2)
piros(Asztal, S2)

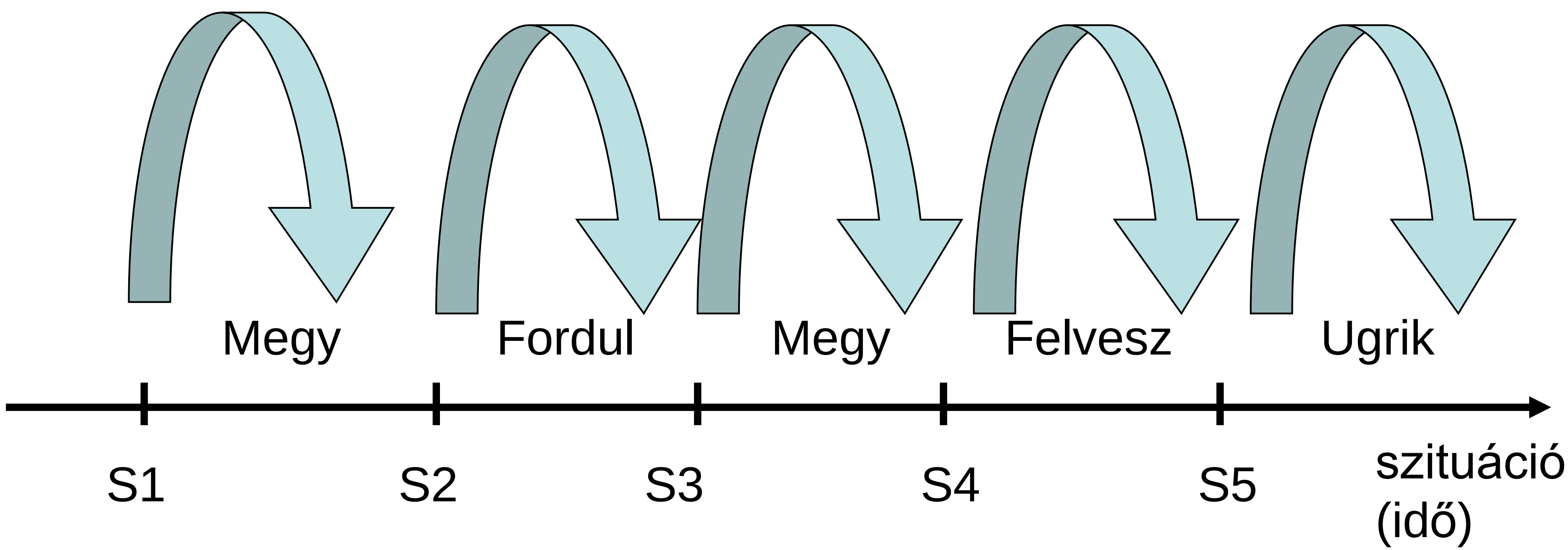
S1 objektum → S2 objektum

S2 = eredmény(Átfest, S1)

Szituáció kalkulus

- a változások leírásának egy módja az elsőrendű logikában.
  - a világ múlása **szituációk** sorozatából áll
  - mindegyike egy „pillanat felvétel” a világ állapotáról
  - egy-egy szituációban egy tény igaz, vagy hamis, **változhat**!

  fluent = „**folyékony esemény**” (pl. piros(Szék, σ))



Hatás axiómák

$\forall$s. szobában(Ágens,s) $\wedge$ $\neg$piros(Asztal,s)
$\rightarrow$ piros(Asztal,eredmény(Megfest,s))

$\forall$s. szobában(Ágens,s) $\rightarrow$ $\neg$szobában(Ágens,eredmény(Kimegy,s))

$\forall$s ott(Arany,s) $\wedge$ vihető(Arany)
$\rightarrow$ birtokol(Arany,eredmény(Megfogás,s))

$\forall$ x, s $\neg$ birtokol (x, eredmény (Elenged, s))

Sajnos ez nem elég! Szükség van még …..

Keret axiómák

$\forall$a, s. $\neg$piros(a,s) $\wedge$ (a $\neq$ Asztal) $\rightarrow$ $\neg$piros(a,eredmény(Megfest, s))

$\forall$ a, x, s. birtokol (x, s) $\wedge$ (a $\neq$ Elenged) $\rightarrow$ birtokol (x, eredmény(a, s))

$\forall$ a, x, s. $\neg$birtokol (x, s) $\wedge$ (a $\neq$ Megfogás $\vee$ $\neg$ (ott(x, s) $\wedge$ vihető(x))
$\rightarrow$ $\neg$ birtokol (x, eredmény (a, s))

Tervkészítés következtetéssel szituációkalkulusban példa

Legyen a feladat nyelvezete:

Ágens szobában van, *S* szituációban: *szoba(Ágens, S)*

asztal színe piros, *S* szituációban: *piros(Asztal, S)*

Mivel más objektum nincs is, le lehet rövidíteni:

ágens szobában van, *S* (kezdeti) szituációban: *szoba*(*S*)

asztal színe piros, *S* (kezdeti) szituációban: *piros(S)*

Ágens cselekvései legyenek: "**Bemegy**", "**Kimegy**", "**Átfest**".

Jelen helyzet: 1. $\neg$*szoba(S)*

2. $\neg$*piros(S)*

azaz az ágens szobán kívül van, szobában az asztal nincs pirosra átfestve.

Probléma: a kívánt helyzet az, hogy az asztal piros legyen és az ágens szobán kívül legyen. Ágensünk képes ezt megvalósítani?

Létezik egyáltalán egy ilyen helyzet?

Hatás axiómák (avagy az ágens kényszercselekedetei)

$\forall \sigma.\ \neg\, szoba(\sigma) \wedge \neg\, piros(\sigma) \rightarrow szoba(eredmény(Bemegy,\ \sigma))$
$\forall \sigma.\ szoba(\sigma) \wedge \neg\, piros(\sigma) \rightarrow piros(eredmény(Atfest,\ \sigma))$
$\forall \sigma.\ szoba(\sigma) \wedge piros(\sigma) \rightarrow \neg\, szoba(eredmény(Kimegy,\ \sigma))$
$\forall \sigma.\ szoba(\sigma) \rightarrow \neg\, szoba(eredmény(Kimegy,\ \sigma))$
$\forall \sigma.\ \neg szoba(\sigma) \rightarrow szoba(eredmény(Bemegy,\ \sigma))$
...

Mitől lesz ilyen? Ha ilyennek megtervezzük és implementáljuk

Keret axiómák (buta formában)

$\forall \sigma.\ szoba(\sigma) \rightarrow szoba(eredmény(Atfest,\ \sigma))$
$\forall \sigma.\ piros(\sigma) \rightarrow piros(eredmény(Bemegy,\ \sigma))$
$\forall \sigma.\ piros(\sigma) \rightarrow piros(eredmény(Kimegy,\ \sigma))$
$\forall \sigma.\ \neg piros(\sigma) \rightarrow \neg\, piros(eredmény(Kimegy,\ \sigma))$
$\forall \sigma.\ \neg piros(\sigma) \rightarrow \neg\, piros(eredmény(Bemegy,\ \sigma))$
…

Probléma: a kívánt helyzet az, …. **létezik egyáltalán?**

$\exists \sigma.\ \neg \boldsymbol{szoba(\sigma)} \wedge \boldsymbol{piros(\sigma)}$?

Legyen egy további rövidítés: $B(s) = eredmény(Bemegy, s)$
$K(s) = eredmény(Kimegy, s)$
$A(s) = eredmény(Átfest, s)$

Ne felejtsük, hogy a beszédes logikai nevek nem a gépnek szólnak, hanem az embereknek, és a gépi eljárásra nincsenek hatással.

Átírással klóz formára:

1. $\neg szoba(S)$
2. $\neg piros(S)$
3. $szoba(\sigma 1) \vee piros(\sigma 1) \vee szoba(B(\sigma 1))$
4. $\neg szoba(\sigma 2) \vee piros(\sigma 2) \vee piros(A(\sigma 2))$
5. $\neg szoba(\sigma 3) \vee \neg piros(\sigma 3) \vee \neg szoba(K(\sigma 3))$
6. $\neg szoba(\sigma 4) \vee szoba(A(\sigma 4))$
7. $\neg szoba(\sigma 5) \vee \neg szoba(K(\sigma 5))$
8. $szoba(\sigma 6) \vee szoba(B(\sigma 6))$
9. $\neg piros(\sigma 7) \vee piros(B(\sigma 7))$
10. $\neg piros(\sigma 8) \vee piros(K(\sigma 8))$
11. $piros(\sigma 9) \vee \neg piros(K(\sigma 9))$
12. $piros(\sigma 10) \vee \neg piros(B(\sigma 10))$
13. $piros(\sigma 11) \vee piros(A(\sigma 11))$
14. $szoba(\sigma 12) \vee \neg piros(\sigma 12)$

Ne felejtsük, hogy a beszédes logikai nevek nem a gépnek szólnak, hanem az embereknek, és a gépi eljárásra nincsenek hatással.

Átírással klóz formára:

1. $\neg y(S)$
2. $\neg x(S)$
3. $y(\sigma 1) \vee x(\sigma 1) \vee y(Z1(\sigma 1))$
4. $\neg y(\sigma 2) \vee x(\sigma 2) \vee x(Z2(\sigma 2))$
5. $\neg y(\sigma 3) \vee \neg x(\sigma 3) \vee \neg y(Z3(\sigma 3))$
6. $\neg y(\sigma 4) \vee y(Z2(\sigma 4))$
7. $\neg y(\sigma 5) \vee \neg y(Z3(\sigma 5))$
8. $y(\sigma 6) \vee y(Z1(\sigma 6))$
9. $\neg x(\sigma 7) \vee x(Z1(\sigma 7))$
10. $\neg x(\sigma 8) \vee x(Z3(\sigma 8))$
11. $x(\sigma 9) \vee \neg x(Z3(\sigma 9))$
12. $x(\sigma 10) \vee \neg x(Z1(\sigma 10))$
13. $x(\sigma 11) \vee x(Z2(\sigma 11))$
14. $y(\sigma 12) \vee \neg x(\sigma 12)$

És az eredmény: Az ágens igenis képes megvalósítani a feladatot, ráadásul $\sigma 12$ állapotban. Mi is ez az állapot?

σ12 = K(A(B(S))) =

eredmeny(**Kimegy**,

eredmeny(**Átfest**,

eredmeny(**Bemegy**, **S**)))

Megvalósítja (akkor meglesz a kívánt állapot), ha:

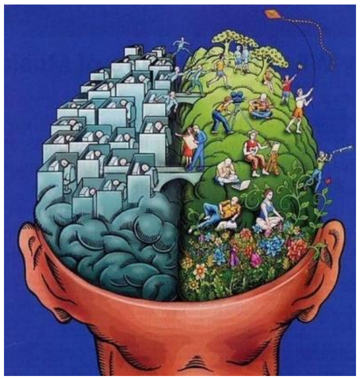
Bemegy → **Átfest** → **Kimegy**

S --------------- S1 ------------- S3 --------------- S4 (szituációk)

cselekvési sorozatot visz véghez.

A szükséges cselekvési sorozatot tehát előre logikailag kitervelte.

Budapesti Műszaki és Gazdaságtudományi Egyetem
Méréstechnika és Információs rendszerek Tanszék



Bizonytalan tudás kezelése

Előadó: Dr. Hullám Gábor

Előadás anyaga: Dr. Dobrowiecki Tadeusz, Dr. Gézsi András

Valószínűségi axiómák

MŰEGYETEM 1782

Témakörök

Keresés
Kényszer kielégítés
Problémamegoldás
Tudásreprezentáció következtetés
Logika
Valószínűségi hálók
Neurális hálók
Döntési fák
Felügyelt tanulás
Megerősítéses tanulás
Q-tanulás
Szekvenciális döntések
Ajánló rendszerek

Bizonytalan tudás

Lehetséges okok

•„Lazaság/lustaság” **-** a részletes kapcsolatok megfogalmazása túl nehéz, a használatuk szintén nehézkes (véges erőforrások)

•Elméleti ismeret hiánya **-** adott problématerületnek az elméleti feltárása még nem zárult le, vagy lezárni soha nem lehet

•Gyakorlati ismeret hiánya **-** nem minden, a szabályokban hivatkozott feltétel ismert a szabályok alkalmazásakor

Bizonytalan tudás

•Példa:

$$\forall p.\, Tünet(p, Fogfájás) \rightarrow Van(p, Fogszuvasodás)$$

$$\forall p.\, Tünet(p, Fogfájás) \rightarrow Van(p, Fogszuvasodás) \vee$$
$$Van(p, Ínysorvadás) \vee Van(p, Bölcsességfognő) \vee \ldots$$

$$\forall p.\, Van(p, Fogszuvasodás) \rightarrow Tünet(p, Fogfájás)$$

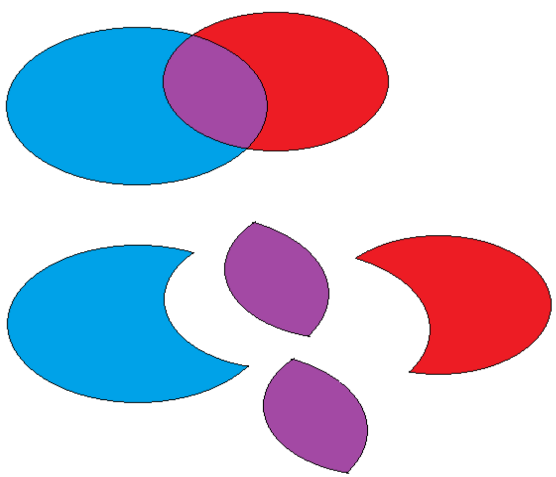
•Fogfájás és fogszuvasodás közötti kapcsolat egyik irányban sem feltétlen logikai következmény.

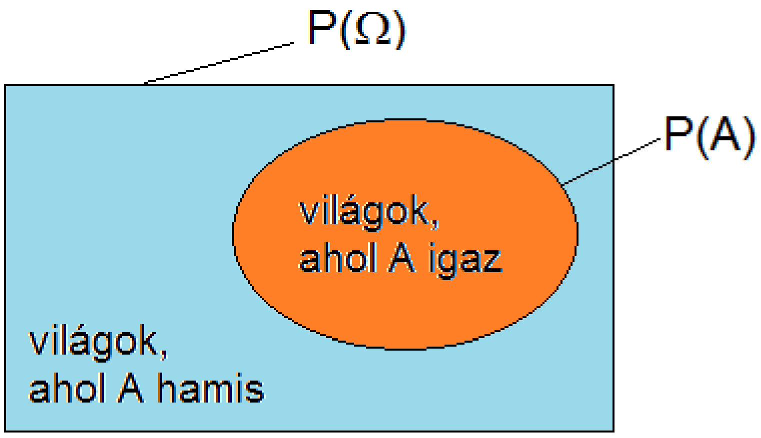
•Az ágens tudása legjobb esetben is csak egy bizonyos mértékű hiedelmet jelent az adott állítással kapcsolatban.

Volt már

Valószínűségi axiómák

1. Minden valószínűség 0 és 1 közé esik $0 \leq P(A) \leq 1$.
2. A biztosan igaz állítás valószínűsége 1,
   a biztosan hamis állításé 0. $P(Igaz) = 1$ $P(Hamis) = 0$
3. Diszjunkció valószínűsége: $P(A \vee B) = P(A) + P(B) - P(A \wedge B)$





Valószínűségi axiómákból bizonyítottuk pl., hogy:

$P(\neg A) = 1 - P(A)$

$P(A) = P(A \wedge B) + P(A \wedge \neg B)$ stb.

Volt már

Valószínűségi állítások

bináris	**Lyuk** = Igaz	P(Lyuk = Igaz) P(Lyuk)
többértékű (kategorikus)	**Időjárás** 1 értéket vesz fel az alábbi 4 lehetségesből {Napos, Esős, Felhős, Havazás}	
		P(Időjárás = Esős)
folytonos változó	**Hőmérséklet** = 22.1 ºC,	**Hőmérséklet** < 22 ºC
		P(Hőmérséklet < 22 ºC)

Feltételes valószínűség: P(A | B) = P(A B) / P(B)

Volt már

Valószínűségi állítások átalakítása

Láncszabály: P(A B C D E) = P(A | B C D E) P(B C D E) =
P(A | B C D E) P(B | C D E) P(C D E) = …
P(A | B C D E) P(B | C D E) P(C | D E) P(D | E) P(E)

$$P(X_1 X_2 X_3 \ldots X_N) = P(X_1 X_2 X_3 \ldots X_{N-1})\, P(X_N \mid X_1 X_2 \ldots X_{N-1})$$
$$= P(X_1 X_2 X_3 \ldots X_{N-2})\, P(X_{N-1} \mid X_1 X_2 \ldots X_{N-2})\, P(X_N \mid X_1 X_2 \ldots X_{N-1})$$
$$= \ldots = \prod_{i=1}^{N} P(X_i \mid X_1 X_2 \ldots X_{i-1})$$

Volt már

Bayes-tétel

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

$$P(A_k|B) = \frac{P(B|A_k)P(A_k)}{\sum_{i=1}^{N} P(B|A_i)P(A_i)}, \quad \bigcup_{i=1}^{N} A_i = \Omega, \quad A_i \cap A_k = \varnothing$$

Miért fontos a Bayes-tétel?

Sokszor rendelkezünk kauzális (ok-okozati) tudással:

Ok Okozat

Ok	Okozat	
Betegség	Tünet	$P(tünet \mid betegség)$
Tűz	Riasztás	$P(riasztás \mid tűz)$

Bayes-tétel

- Viszont sokszor„ellentétes irányban” szeretnék következtetni, tehát evidenciák (következmények) alapján az ok(ok)ra

$P(betegség \mid tünet)$ Betegség Tünet

$P(tűz \mid riasztás)$ Tűz Riasztás

Bayes-tétel jelentőssége

Lehetővé teszi a valószínűségi állítások átalakítását

- Így kiszámíthatóvá válnak nehezen becsülhető mennyiségek
- Oksági irány: általában könnyebb becsülni
- Diagnosztikai irány: általában nehezebb

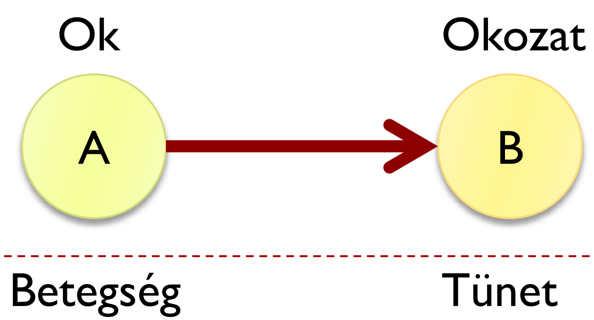
Mi a Tünet létrejöttének valószínűsége a Betegség ismeretében?

Mi a Betegség kialakulásának a valószínűsége?

$$P(Betegség \mid Tünet) = \frac{P(Tünet \mid Betegség)P(Betegség)}{P(Tünet)}$$

Mi a Betegség meglétének valószínűsége a Tünet ismeretében?

Mi a Tünet létrejöttének a valószínűsége?



Bayes-tétel jelentősége

Mi a Tünet létrejöttének valószínűsége a Betegség ismeretében?

Mi a Betegség kialakulásának a valószínűsége?

$$P(Betegség \mid Tünet) = \frac{P(Tünet \mid Betegség)P(Betegség)}{P(Tünet)}$$

Mi a Betegség meglétének valószínűsége a Tünet ismeretében?

Mi a Tünet létrejöttének a valószínűsége?

a posteriori valószínűség (poszterior)

Likelihood

a priori valószínűség (prior)

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

Normalizációs konstans

Bayes-tétel - műveletek

Összetett feltétel

A értéke: $\{v_1, v_2, \ldots v_N\}$

$$P(A = v_k) = P_k$$

$$P(A = v_k) \wedge P(A = v_j) = 0, \quad k \neq j$$

$$P(A = v_1) \vee P(A = v_2) \ldots \vee P(A = v_N) = 1$$

$$P(A = v_1) \vee P(A = v_2) \ldots \vee P(A = v_k) = \sum_{i=1}^{k} P_i$$

$$P(A \mid B) = \frac{P(B \mid A) P(A)}{P(B)} \leftrightarrow P(A \mid \boxed{B \wedge X}) = \frac{P(B \mid \boxed{A \wedge X}) P(A \mid X)}{P(B \mid X)}$$

Bayes-tétel - műveletek

Bővítés:

A értéke: $\{v_1, v_2, \ldots v_N\}$

$P(A = v_k) = P_k$

$$P(A = v_k) \wedge P(A = v_j) = 0, \quad k \neq j$$

$$P(A = v_1) \vee P(A = v_2) \ldots \vee P(A = v_N) = 1$$

$$P(A = v_1) \vee P(A = v_2) \ldots \vee P(A = v_k) = \sum_{i=1}^{k} P_i$$

$$\boxed{P(A \mid B) = \frac{P(B \mid A) P(A)}{P(B)} = \frac{P(B \mid A) P(A)}{\sum_{i=1}^{N} P(B \mid A_i) P(A_i)}}$$

Pl. ha A egy bináris valószínűségi változó:

$$P(A = 1 \mid B = 1) = \frac{P(B = 1 \mid A = 1) P(A = 1)}{P(B = 1 \mid A = 0) P(A = 0) + P(B = 1 \mid A = 1) P(A = 1)}$$

Valószínűség értelmezése
Frekventista vs. bayesi megközelítés

frekventista nézőpont: a valószínűség objektív, események gyakoriságából számítható

Valószínűség (frekventista definíciója): Egy adott ***A*** esemény valószínűsége az a számérték, amely körül az esemény relatív gyakorisága (f_A) ingadozik, ha egyre több kísérletet végzünk.

$$P(A) = \lim_{n \to \infty} f_A$$

P(A) tehát azt mutatja meg, hogy az ***A*** esemény az összes kísérlet mekkora hányadában (%) következik be.

Valószínűség értelmezése
Frekventista vs. bayesi megközelítés

bayesi nézőpont: a valószínűség egy eseménybe vetett hiedelem mértéke.

Valószínűség (bayesi definíciója): Egy adott ***A*** esemény valószínűsége az adott esemény bekövetkezési esélye.

Ez a gyakorlatban azt a hiedelmet fejezi ki, hogy a pillanatnyi tudásunk birtokában, a jelenlegi helyzettől megkülönböztethetetlen esetek mekkora hányadában fog bekövetkezni az adott esemény.

• Prior valószínűségekből (hiedelmekből) indulunk ki és az új evidencia érkezésekor azokat frissítjük (posterior valószínűségekké)

$$P(A) \rightarrow P(A) \rightarrow \ldots \rightarrow P(A) \underset{\substack{\uparrow \\ B}}{\rightarrow} \text{Bayes-tétel} \rightarrow P(A|B)$$

A bayesi következtetés, mint a megismerés általános modellje - gondolatkísérlet

Chile országa 15 régióra van felosztva. Az ország teljes mérete 756 096 km^2. **Mekkora lehet az Atacama régió**? Legyen az …

- A_1 esemény az, hogy Atacama régió kisebb, mint 10 000 km^2,
- A_2 esemény az, hogy 10 000 és 50 000 km^2 között van a mérete,
- A_3 esemény az, hogy 50 000 és 100 000 km^2 között van,
- A_4 esemény az, hogy nagyobb, mint 100 000 km^2.

- Rendeljen valószínűségeket az $A_1, \dots, A_4$ eseményekhez!

Chile…

- az A_1 esemény az, hogy Atacama régió kisebb, mint 10 000 km^2,
- az A_2 esemény az, hogy 10 000 és 50 000 km^2 között van a mérete,
- az A_3 esemény az, hogy 50 000 és 100 000 km^2 között van,
- az A_4 esemény az, hogy nagyobb, mint 100 000 km^2.

- Atacama a negyedik legnagyobb régió.
  Hogyan módosítaná a valószínűségeket?

Chile…

- az A_1 esemény az, hogy Atacama régió kisebb, mint 10 000 km^2,
- az A_2 esemény az, hogy 10 000 és 50 000 km^2 között van a mérete,
- az A_3 esemény az, hogy 50 000 és 100 000 km^2 között van,
- az A_4 esemény az, hogy nagyobb, mint 100 000 km^2.
- Atacama a negyedik legnagyobb régió.

- A legkisebb régió a központi régió, *Santiago Metropolitan*, amelynek mérete 15 403 km^2. Hogyan módosítanád a valószínűségeket?

Chile…

- az A_1 esemény az, hogy Atacama régió kisebb, mint 10 000 km^2,
- az A_2 esemény az, hogy 10 000 és 50 000 km^2 között van a mérete,
- az A_3 esemény az, hogy 50 000 és 100 000 km^2 között van,
- az A_4 esemény az, hogy nagyobb, mint 100 000 km^2.
- Atacama a negyedik legnagyobb régió.
- A legkisebb régió a központi régió, Santiago Metropolitan, amelynek mérete 15 403 km^2.

- A harmadik legnagyobb régió *Aysén del General Carlos Ibanez del Campo*, amely 108 494 km^2.

- Hogyan módosítaná a valószínűségeket?

Chile…

- az A_1 esemény az, hogy Atacama régió kisebb, mint 10 000 km^2,
- az A_2 esemény az, hogy 10 000 és 50 000 km^2 között van a mérete,
- az A_3 esemény az, hogy 50 000 és 100 000 km^2 között van,
- az A_4 esemény az, hogy nagyobb, mint 100 000 km^2.
- Atacama a negyedik legnagyobb régió.
- A legkisebb régió a központi régió, Santiago Metropolitan, amelynek mérete 15 403 km^2.
  A harmadik legnagyobb régió Aysén del General Carlos Ibanez del Campo, amely 108 494 km^2.
- Atacama mérete: 75 176 km^2

Hiedelmek és értelmezésük

Egy adott kijelentéshez rendelt (szubjektív) **0 / 1 valószínűség**: határozott hiedelem, hogy az állítás **hamis / igaz.**

A **0 és 1 közötti** valószínűség a mondat **igazságtartalmában való hiedelem** mértékeinek felel meg. **Az állítás valójában persze vagy igaz vagy hamis.**

A 0.8 (szubjektív) valószínűség nem jelenti, hogy az állítás a „80 %-ban igaz”, hanem egy 80 %-os mértékű hiedelmet, azaz igen erős elvárást az állítás igazságával szemben.

Valószínűségi kijelentés szemantikája: egy **kijelentéshez rendelt valószínűség az addigi észlelésektől** (tény, evidencia, tényállás) **függ**.

Ágens kihúz egy lapot egy megkevert kártyapakliból. Mielőtt ránézne a lapra: P(„a lap pikk ász (lesz)”) = 1/52. Miután megnézte: P(...) = 0, vagy 1.

Orvos páciens megvizsgálása után P1 valószínűségűnek lát egy konkrét betegséget. Labor lelet ezt vagy erősíti (> P1), vagy gyengíti (< P1).

Események – véletlen kísérlet

- **Elemi esemény:** minden lehetséges kimenetel ($e_1, e_2, \ldots, e_n$), amiről egy kísérlet elvégzése után eldönthető, hogy bekövetkezett vagy sem
  - $e_1 \wedge e_2 \wedge \ldots \wedge e_n = \emptyset$
- **Eseménytér:** egy kísérlet összes kimenetele, az összes elemi esemény halmaza (Ω).
  - $e_1 \cup e_2 \cup \ldots \cup e_n = \Omega$
- **Véletlen esemény:** az eseménytér egy részhalmaza
- **Biztos esemény:** a kísérlet során biztosan (minden kimenetelnél) bekövetkezik.
- **Ellentett esemény**: akkor és csak akkor következik be, ha az eredeti esemény nem következik be

Volt már

Együttes valószínűség-eloszlás

- Az együttes valószínűség-eloszlás P(X_1, ... ,X_n) minden egyes elemi eseményhez valószínűséget rendel.

- Ha minden vizsgált valószínűségi változó diszkrét, akkor az együttes valószínűség-eloszlás leírható egy *n*-dimenziós táblázattal

- Egy cella = az adott állapot valószínűsége.

Írjuk le a budapesti időjárást két (egy bináris és egy ternáris) változóval:

időjárás: {napos, felhős},

hőmérséklet: {meleg, közepes, hideg}.

Akkor az együttes eloszlás P(*időjárás*, *hőmérséklet*):

Volt már

Együttes valószínűség-eloszlás

Írjuk le a budapesti időjárást két (egy bináris és egy ternáris) változóval:

időjárás: {napos, felhős},

hőmérséklet: {meleg, közepes, hideg}.

Akkor az együttes eloszlás P(*időjárás*, *hőmérséklet*):

	Meleg	Közepes	Hideg
Napos	.1	.2	.1
Felhős	.05	.35	.2

- Mivel az elemi események egymást kizáróak, ezek együttes bekövetkezése szükségszerűen hamis tény.

- Az axiómákból következően: **a táblázat elemeinek összege 1.**

Volt már

Marginális és más eloszlások

$$P(X,Y) = \sum_{z \in dom(Z)} P(X,Y,Z=z)$$

P(*hőmérséklet*) =

Meleg	Közepes	Hideg
.15	.55	.3

P(*időjárás*) =

Napos	Felhős
.4	.6

	Meleg	Közepes	Hideg
Napos	.1	.2	.1
Felhős	.05	.35	.2

Volt már

Marginális és más eloszlások

P(*hőmérséklet*| *időjárás* = Napos) =

Meleg	Közepes	Hideg
.25	.50	.25

P(*időjárás*| *hőmérséklet*) =

	Meleg	Közepes	Hideg
Napos	.67	.36	.33
Felhős	.33	.64	.67

	Meleg	Közepes	Hideg
Napos	.1	.2	.1
Felhős	.05	.35	.2

Együttes valószínűség-eloszlás

Jó hír: együttes eloszlás birtokában minden kérdésre kapunk választ, ami a benne szereplő véletlen változók viszonyára és tulajdonságaira vonatkozik.

Rossz hír: nemigen megy 10-nél több változót tartalmazó eloszlások megadása

$P(x_1, x_2, \ldots, x_N)$ esetén kell $2^N - 1$ független valószínűségérték.

A diagnózishoz exponenciális számú valószínűség ismerete szükséges.

Rémálom: ha valamelyik valószínűség értéke megváltozik?

A bayesi frissítés

- Egyesével gyűjtjük a tényeket, majd módosítjuk az ismeretlen változóval kapcsolatos korábbi hiedelmi mértéket.

1. Fogfájás:

$$P(Fogsz \mid Ffáj) = P(Fogsz)\frac{P(Ffáj \mid Fogsz)}{P(Ffáj)}$$

2. Lyuk*:* a Bayes-tételt úgy alkalmazzuk, hogy a továbbiakban a *Fogfájás*-t állandó feltételnek tekintjük:

$$P(Fogsz \mid Lyuk) = P(Fogsz)\frac{P(Lyuk \mid Fogsz)}{P(Lyuk)}$$

$$P(Fogsz \mid Ffáj \wedge Lyuk) = P(Fogsz \mid Ffáj)\frac{P(Lyuk \mid Ffáj \wedge Fogsz)}{P(Lyuk \mid Ffáj)}$$

A bayesi frissítés -példa

$$P(Fogsz \mid Ffáj \wedge Lyuk) = P(Fogsz \mid Ffáj)\frac{P(Lyuk \mid Ffáj \wedge Fogsz)}{P(Lyuk \mid Ffáj)}$$

$$P(Fogsz \mid Ffáj) = P(Fogsz)\frac{P(Ffáj \mid Fogsz)}{P(Ffáj)}$$

$$P(Fogsz \mid Ffáj \wedge Lyuk)$$
$$= P(Fogsz)\frac{P(Ffáj \mid Fogsz)}{P(Ffáj)}\frac{P(Lyuk \mid Ffáj \wedge Fogsz)}{P(Lyuk \mid Ffáj)}$$

Feltételes függetlenség

Mind a fogfájásnak, mind a szonda lyukba akadásának közvetlen oka a fogszuvasodás.

Amint **tudjuk**, hogy fogszuvasodás, nem hisszük, hogy a szonda lyukba akadásának valószínűsége a fogfájástól fog függeni.

Hasonlóképpen, a szonda találata nem befolyásolja annak valószínűségét, hogy a szuvasodás fogfájást okoz.

$$P(Lyuk \mid Ffáj \wedge Fogsz) = P(Lyuk \mid Fogsz)$$

$$P(Ffáj \mid Lyuk \wedge Fogsz) = P(Ffáj \mid Fogsz)$$

a *Fogszuvasodás* ténye esetén a *Fogfájás* és *Lyuk* között fennáll a **feltételes függetlenség**.

Feltételes függetlenség - alkalmazás

$$P(Fogsz \mid Ffáj \wedge Lyuk)$$

$$= P(Fogsz)\frac{P(Ffáj \mid Fogsz)}{P(Ffáj)}\frac{\boxed{P(Lyuk \mid Ffáj \wedge Fogsz)}}{P(Lyuk \mid Ffáj)}$$

$$P(Lyuk \mid Ffáj \wedge Fogsz) = P(Lyuk \mid Fogsz)$$

$$P(Ffáj \mid Lyuk \wedge Fogsz) = P(Ffáj \mid Fogsz)$$

$$P(Fogsz \mid Ffáj \wedge Lyuk)$$

$$= P(Fogsz)\frac{P(Ffáj \mid Fogsz)}{P(Ffáj)}\frac{\boxed{P(Lyuk \mid Fogsz)}}{P(Lyuk \mid Ffáj)}$$

Normalizálás

Még mindig kérdéses a: *P*(*Lyuk*|*Fogfájás*)?!

\- várhatóan figyelembe kell venni a tünetek összes lehetséges párosítását (hármasait, stb.), valóságban ez a kifejezés kiesik:

a nevezők szorzata: *P*(*Lyuk*|*Fogfájás*) *P*(*Fogfájás*) = *P*(*Lyuk*)

Ezt az un. **normalizálással** kiküszöbölhetjük, feltéve, hogy pl. a *P*(*Lyuk*|¬*Fogszuvasodás*)-t megbecsüljük.

$$P(Fogsz \mid Lyuk) = \frac{P(Lyuk \mid Fogsz)P(Fogsz)}{P(Lyuk)}$$

$$P(Fogsz \mid Lyuk) \approx P(Lyuk \mid Fogsz)P(Fogsz)$$

Normalizálás

$$P(Fogsz \mid Lyuk) = \frac{P(Lyuk \mid Fogsz)P(Fogsz)}{P(Lyuk)}$$

$$P(Fogsz \mid Lyuk) \approx P(Lyuk \mid Fogsz)P(Fogsz)$$

$$P(Fogsz \mid Lyuk) = \alpha P(Lyuk \mid Fogsz)P(Fogsz) = C_1\alpha$$

$$P(\neg Fogsz \mid Lyuk) = \alpha P(Lyuk \mid \neg Fogsz)P(\neg Fogsz) = C_2\alpha$$

$$P(Fogsz \mid Lyuk) + P(\neg Fogsz \mid Lyuk) = (C_1 + C_2)\alpha = 1$$

$$\alpha = (C_1 + C_2)^{-1}$$

Lehetséges problémák

- Az a priori feltételes és együttes valószínűségek begyűjtése nehéz és költséges

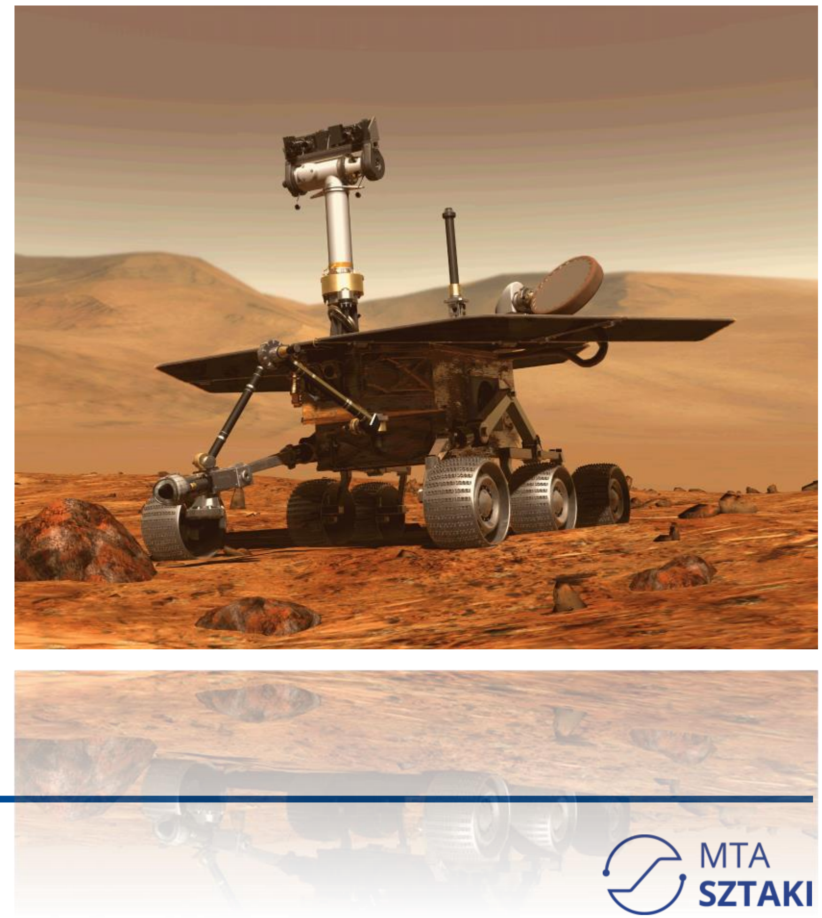
- Az emberek rossz valószínűségbecslők (szubjektív megközelítés esetén)

- A Bayes-szabály sok számítást igényel

Mesterséges Intelligencia
Cselekvéstervezés

Kovács András

MTA SZTAKI, akovacs@sztaki.hu



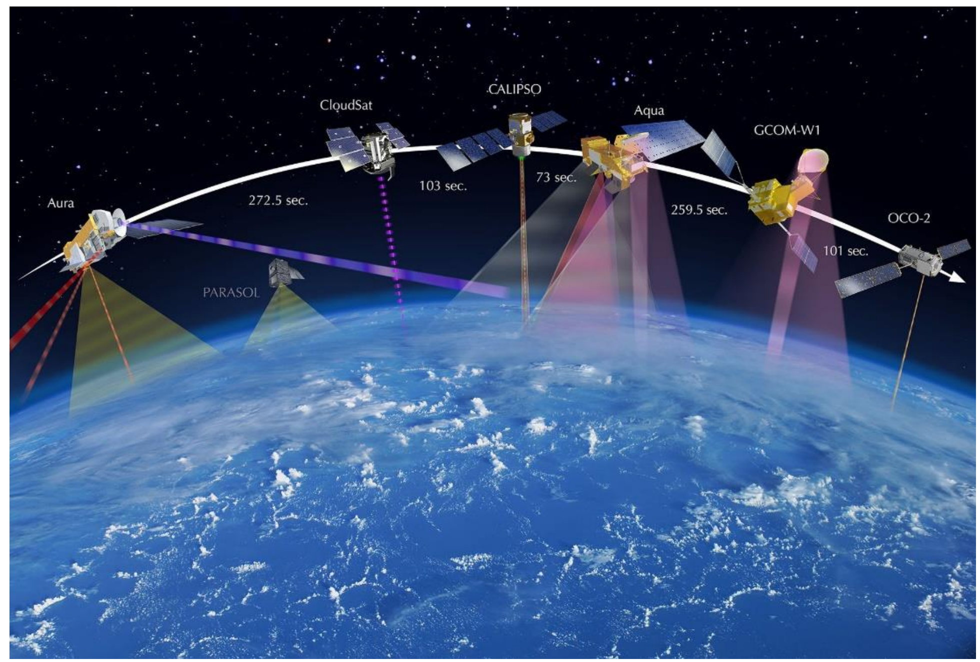
2020.10.15.

Cselekvéstervezési feladatok

- Logisztikai hálózatok
  - Küldeményeket juttatnánk el feladási helyükről célállomásukra
  - Adott járműveket használhatunk (pl. repülő, teherautó)
  - Hogyan csináljuk?

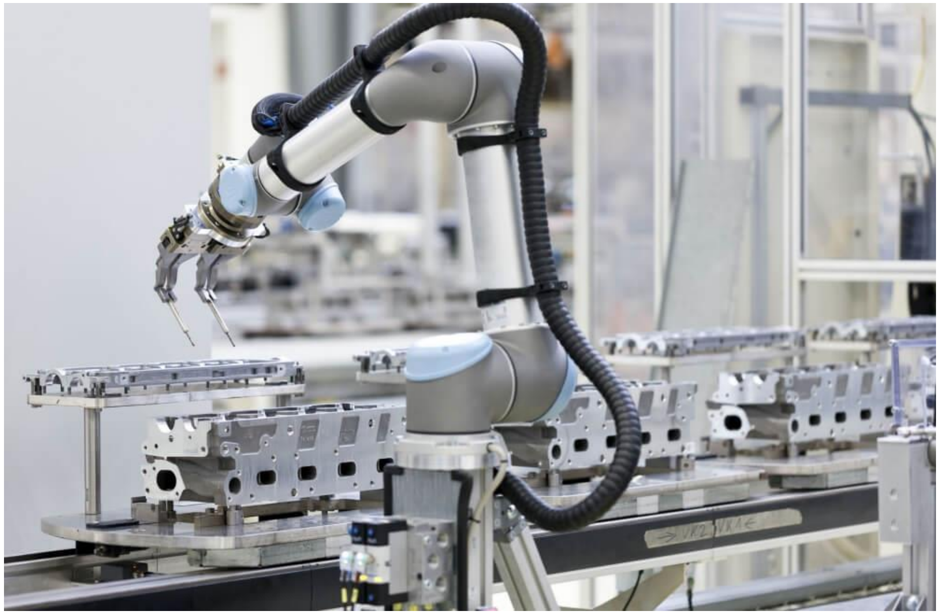
Cselekvéstervezési feladatok (2)

- Műholdas megfigyelések tervezése
  - Adott megfigyelési műveletek (hely, időablak, eszközigény, stb.)
  - Adott műholdak (pálya, eszközök, energiaforrás, stb.)
  - Hogyan csináljuk?



Cselekvéstervezési feladatok (3)

- Szerelési művelettervezés
  - Alkatrészekből szerelnénk össze egy adott terméket
  - Adott eszközökkel (pl. robotok, szerszámok)
  - A technológiai korlátok figyelembe vételével
  - Hogyan csináljuk?



Cselekvéstervezési feladat definíciója

- Olyan cselekvéssorozat meghatározása, amely biztosítja az ágens céljainak elérését
  - Egy adott kiindulási állapotból
  - Egy célállapotba
  - Valamely, az ágens számára elérhető akciók sorozatának végrehajtásával

- Dedikált vs. általános megoldó

- Mi kell ehhez?
  - Egy nyelv, amivel leírjuk a tervezési feladatot
  - Egy algoritmus, ami megoldja

Feltevések

- Determinisztikus cselekvéstervezés
  - A környezet tökéletesen megfigyelhető
  - Az ágens cselekvéseinek kimenete determinisztikus
  - Más ágensek nem hatnak a környezetre
- Tervkészítés *offline*, majd a terv végrehajtása *csukott szemmel*
  - Nincs szükség valós időben érzékelésre és döntéshozatalra
- Zárt világ feltevés *(closed world assumption)*
  - A világ teljes egészében megismerhető
  - Amiről nem tudjuk, hogy igaz, az hamis

A tervezési feladat leírása

- Világállapot leírása
  - Kezdeti állapot: teljesen specifikált
  - Célállapot: teljesen vagy részben specifikált (célfeltétel)

- Lehetséges akciók: cselekvés sémák
  - Milyen feltétele van az alkalmazásuknak?
  - Milyen hatásuk van a világállapotra?

Világállapot leírása

- Atomi elsőrendű logikai kifejezések konjunkciója
  - Kötött és függvénymentes
  - Megengedett: airplane(Plane1) ∧ at(Plane1, BUD) ∧ ...
  - Nem megengedett: at(Plane2, x), at(Plane3,BasisOf(Plane3)), at(Truck3, BUD) ∨ at(Truck3, NAP)
- Ezt a formalizmust használjuk
  - A kezdeti állapot leírására
  - A célfeltétel leírására (kis kiegészítéssel: negált literálok)
  - (Tetszőleges állapot belső reprezentációjára egy állapottérbeli tervező algoritmus által)

Célállapot/célfeltétel

- A célfeltétel egy részlegesen definiált állapot
  - Elsőrendű logikai literálok konjunkciója
  - Ponált és negált literálok megengedettek

- Egy állapot célállapot, ha
  - Tartalmazza a célfeltétel összes ponált literálját, és
  - Nem tartalmazza a cél egyik negált literálját sem

- Pl:
  - Célfeltétel: at(Package1, BUD) ∧ at(Plane2, NAP) ∧ ¬at(Truck3, BUD_BME)
  - Állapot 1: at(Package1, BUD) ∧ at(Plane2, NAP) ∧ at(Package8, CDG) ∧ … ✓
  - Állapot 2: at(Package1, BUD) ∧ at(Package8, CDG) √ ✗
  - Állapot 3: at(Package1, BUD) ∧ at(Plane2, NAP) ∧ at(Truck3, BUD_BME) ∧ … ✗

MTA SZTAKI

Cselekvés leírása

- Cselekvés séma
  - Akció neve
  - Paraméter lista
  - Előfeltételek
  - Hatások

```
(:action fly
    :parameters (?p ?s ?d)
    :precondition (and (airplane ?p)
            (airport ?s)
            (airport ?d)
            (at ?p ?s)
            (not (= ?s ?d)))
    :effect (and (at ?p ?d)
            (not (at ?p ?s))))
```

Cselekvés séma alkalmazása

- Paraméterlista behelyettesítése olyan módon, hogy
  - A cselekvés összes előfeltétele kielégíthető
  - A világállapot leírásában található literálokkal

Világállapot:
(airport BUD) ∧ (airport NAP) ∧ (airport CDG) ∧ (at plane1 BUD) ∧ …

Cselekvés séma:

```
(:action fly
    :parameters (?p ?s ?d)
    :precondition (and (airplane ?p)
            (airport ?s)
            (airport ?d)
            (at ?p ?s)
            (not (= ?s ?d)))
    :effect …
```

Lehetséges cselekvések:
fly(plane1 BUD NAP)
fly(plane1 BUD CDG)
…

Cselekvés séma alkalmazása (2)

- Világállapot frissítése a cselekvés hatásaival
  - A megfelelő literálok hozzáadása/eltávolítása a világállapotból

(airport BUD) ∧ (airport NAP) ∧ (airport CDG) ∧ (at plane1 BUD) ∧ …

fly(plane1 BUD NAP)

(airport BUD) ∧ (airport NAP) ∧ (airport CDG) ∧ (at plane1 NAP) ∧ …

```
(:action fly
    :parameters (?p ?s ?d)
    :precondition (and (airplane ?p)
            (airport ?s)
            (airport ?d)
            (at ?p ?s)
            (not (= ?s ?d)))
    :effect (and (at ?p ?d)
             (not (at ?p ?s))))
```

A teljes tervezési feladat leírása

- Planning Domain Definition Language, PDDL
  - A szakterület (domain) definíciója
  - A feladatpéldány definíciója

- Szakterület definíció
  - Fejléc
  - Predikátumok
  - Cselekvés sémák

```
(define (domain logistics-strips)
  (:requirements :strips :equality)
  (:predicates (obj ?o)
               (truck ?t)
               (at ?o ?l)
               (in ?o ?t)
               (airplane ?p)
               (airport ?s)
               (in-city ?s ?city)
               (city ?c)
               (location ?l))
...
```

Fejléc

Predikátumok

A teljes tervezési feladat leírása (2)

- Szakterület definíció, folyt.
  - Cselekvés sémák: repülő, ill. teherautó mozgatása

```
(:action fly
        :parameters (?p ?s ?d)
        :precondition (and (airplane ?p)
                           (airport ?s)
                           (airport ?d)
                           (at ?p ?s)
                           (not (= ?s ?d)))
        :effect (and (at ?p ?d)
                     (not (at ?p ?s))))
(:action drive
        :parameters (?truck ?s ?d ?city)
        :precondition (and (truck ?truck)
                           (at ?truck ?s)
                           (in-city ?s ?city)
                           (in-city ?d ?city)
                           (not (= ?s ?d)))
        :effect (and (at ?truck ?d)
                     (not (at ?truck ?s)))))
```

A teljes tervezési feladat leírása (2)

- Szakterület definíció, folyt.
  - Cselekvés sémák: repülő, ill. teherautó mozgatása

```
(:action fly
         :parameters (?p ?s ?d)
         :precondition (and (airplane ?p)
                            (airport ?s)
                            (airport ?d)
                            (at ?p ?s)
                            (not (= ?s ?d)))
         :effect (and (at ?p ?d)
                      (not (at ?p ?s))))
(:action drive
         :parameters (?truck ?s ?d ?city)
         :precondition (and (truck ?truck)
                            (at ?truck ?s)
                            (in-city ?s ?city)
                            (in-city ?d ?city)
                            (not (= ?s ?d)))
         :effect (and (at ?truck ?d)
                      (not (at ?truck ?s)))))
```

A teljes tervezési feladat leírása (3)

- Szakterület definíció, folyt.
  - Cselekvés sémák: repülő, ill. teherautó berakodása, kirakodása

```
(:action load-truck
    :parameters (?o ?truck ?loc)
    :precondition (and (obj ?o) (truck ?truck)
                       (at ?o ?loc) (at ?truck ?loc))
    :effect (and (not (at ?o ?loc))
                 (in ?o ?truck)))

(:action load-plane
    :parameters (?o ?p ?loc)
    :precondition (and (obj ?o) (airplane ?p)
                       (at ?o ?loc) (at ?p ?loc))
    :effect (and (not (at ?o ?loc))
                       (in ?o ?p)))
(:action unload
    :parameters (?o ?v ?loc)
    :precondition (and (in ?o ?v)
                            (at ?v ?loc))
    :effect (and (at ?o ?loc)
                       (not (in ?o ?v))))
```

A teljes tervezési feladat leírása (3)

- Szakterület definíció, folyt.
  - Cselekvés sémák: repülő, ill. teherautó berakodása, kirakodása

```
(:action load-truck
        :parameters (?o ?truck ?loc)
        :precondition (and (obj ?o) (truck ?truck)
                           (at ?o ?loc) (at ?truck ?loc))
        :effect (and (not (at ?o ?loc))
                     (in ?o ?truck)))

(:action load-plane
        :parameters (?o ?p ?loc)
        :precondition (and (obj ?o) (airplane ?p)
                           (at ?o ?loc) (at ?p ?loc))
        :effect (and (not (at ?o ?loc))
                     (in ?o ?p)))
(:action unload
        :parameters (?o ?v ?loc)
        :precondition (and (in ?o ?v)
                           (at ?v ?loc))
        :effect (and (at ?o ?loc)
                     (not (in ?o ?v))))
```

A teljes tervezési feladat leírása (4)

- A feladatpéldány definíciója
  - Entitások
  - Kezdeti állapot
  - Célfeltétel

- Pl. egy csomag eljuttatása a Budapesti postáról a nápolyi postára

```
(define (problem log0)
  (:domain logistics-strips)
  (:objects package1 BUD-truck NAP-truck airplane1
            BUD-post NAP-post BUD-airport NAP-airport
            BUD NAP)
  (:init (OBJ package1)
         (TRUCK BUD-truck)
         (TRUCK NAP-truck)
         (AIRPLANE airplane1)
         (LOCATION BUD-post)
         (LOCATION NAP-post)
         (LOCATION BUD-airport)
         (LOCATION NAP-airport)
         (AIRPORT BUD-airport)
         (AIRPORT NAP-airport)
         (CITY BUD)
         (CITY NAP)
         (IN-CITY BUD-post BUD)
         (IN-CITY BUD-airport BUD)
         (IN-CITY NAP-post NAP)
         (IN-CITY NAP-airport NAP)
         (at package1 BUD-post)
         (at airplane1 BUD-airport)
         (at BUD-truck BUD-airport)
         (at NAP-truck NAP-airport))
  (:goal (and (at package1 NAP-post)))
)
```

Statikus predikátumok

Dinamikus predikátumok

A terv

- Cselekvések (részben) rendezett halmaza

(drive BUD-truck BUD-airport BUD-post BUD)

(load-truck package1 BUD-truck BUD-post)

(drive BUD-truck BUD-post BUD-airport BUD)

(unload package1 BUD-truck BUD-airport)

(load-plane package1 airplane1 BUD-airport)

(fly airplane1 BUD-airport NAP-airport)

(unload package1 airplane1 NAP-airport)

(load-truck package1 NAP-truck NAP-airport)

(drive NAP-truck NAP-airport NAP-post NAP)

(unload package1 NAP-truck NAP-post)

Tervkészítő algoritmus: megközelítések

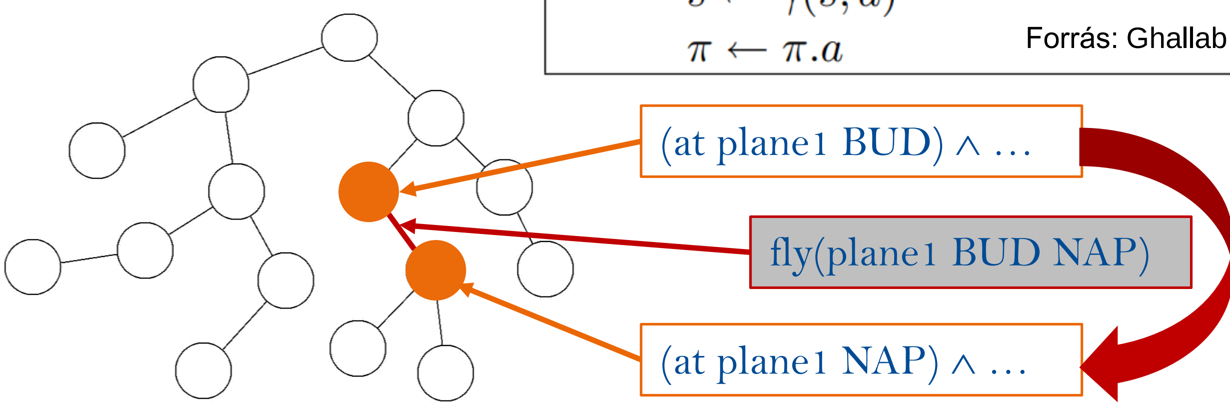
- Szituációkalkulus
  - Az elsőrendű logikai kiterjesztése szituáció változókkal
  - Történetileg érdekes
- Keresés az állapottérben
  - A keresési csomópont egy világállapot
  - Egymást követő állapotok sorozata meghatároz egy tervet !
- Keresés a tervtérben
  - A keresési csomópont egy részleges terv
  - Finomítjuk, mígnem egy teljesen definiált, megengedett tervet kapunk !
- Egyéb technikák, pl.
  - Gráf-alapú tervkészítés (GraphPlan)
  - Fordítás kielégíthetőségi feladatra (SAT)

Keresés állapottérben: előre láncoló keresés

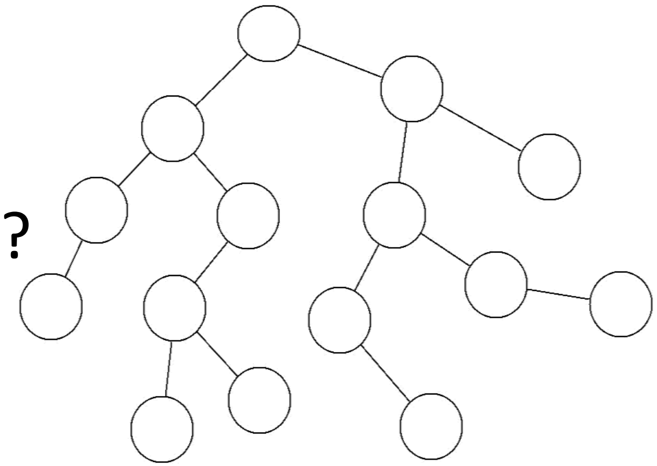
- Általános (nemdeterminisztikus) séma

Forward-search(O, s_0, g)
    $s \leftarrow s_0$
    $\pi \leftarrow$ the empty plan
    loop
        if s satisfies g then return π
        applicable $\leftarrow \{a \mid a$ is a ground instance of an operator in O,
                and $\mathrm{precond}(a)$ is true in $s\}$
        if *applicable* $= \emptyset$ then return failure
        nondeterministically choose an action $a \in$ *applicable*
        $s \leftarrow \gamma(s, a)$
        $\pi \leftarrow \pi.a$

Forrás: Ghallab, Nau, Traverso: Automated planning – Theory and practice



Keresés állapottérben



- Milyen sorrendben járjuk be az állapotteret?
  - Keresési eljárás
  - Heurisztika a kifejtendő csomópont kiválasztására
- Elvileg tetszőleges ismert fa-keresési eljárás alkalmazható
  - Mélységi keresés
  - Szélességi keresés
  - Legjobbat-először keresés
    - Mohó
    - A*
    - …

Keresés állapottérben (2)

- Szélességi és legjobbat-először: helyes és teljes
  - Az exponenciális memóriaigény viszont problémás
- Mélységi, mohó, IDA*
  - A memóriaigény lineáris a terv és az állapotleírás hosszában
  - Teljesek?
    - Az eddig vizsgált tervezési feladatok (AKA klasszikus tervezés) végesek
    - A lehetséges állapotok száma véges
    - A keresési eljárások ciklus-ellenőrzéssel teljessé tehetők

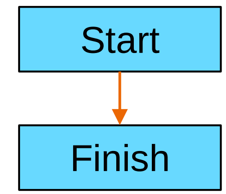
Csomópont választási heurisztikák

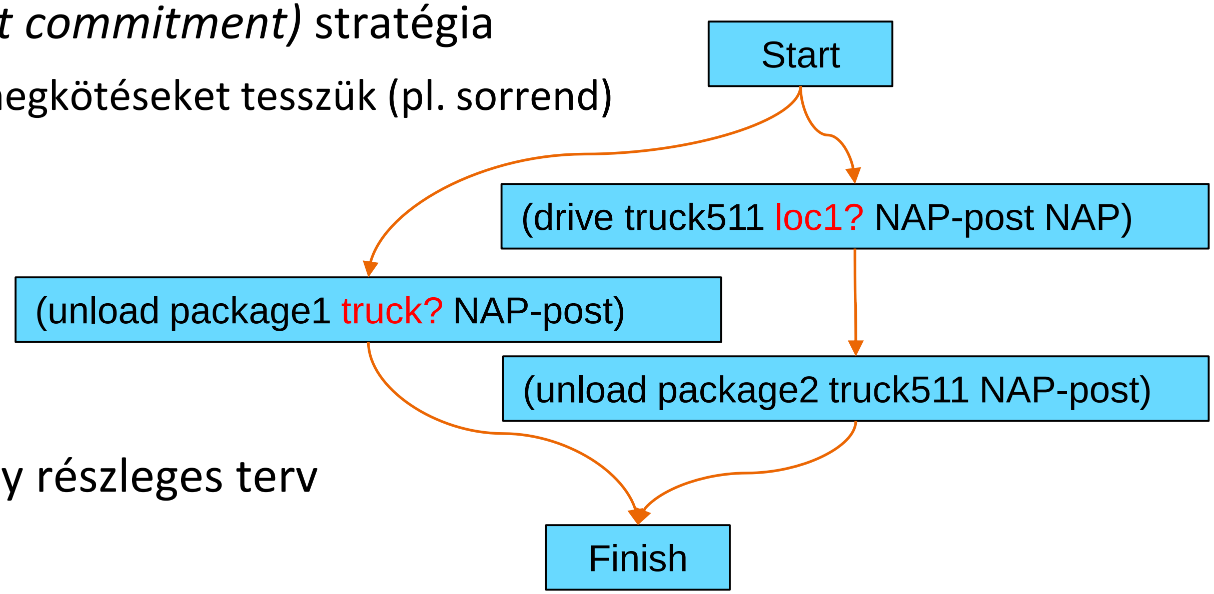
- Kézenfekvő heurisztikák
  - Pl. kielégítetlen célfeltételek száma
  - Alulbecsühet: cselekvések közötti kölcsönhatások
  - Túlbecsülhet: egy cselekvés több célfeltételt előállít

- Egy elfogadható (*admissible*) heurisztika (pl. A*-hoz)
  - Relaxált feladat: töröljük minden cselekvés negált hatásait (Empty-delete-list)
  - Negált cél/előfeltételek esetén új „negált” atom, pl. At(.) → NotAt(.)
  - Egyszerű tervező algoritmus futtatásával számítható (GraphPlan)
  - Előre és hátrafelé láncoló keresésnél is használható
  - Egy legendás cselekvéstervező rendszer alapja: FastForward (FF)

Keresés a tervek terében

- Motiváció
  - Az állapottérben való keresés redundáns
    - A cselekvések minden sorrendjét végigpróbálja
    - Mielőtt rájön, hogy az adott ágon nincs megengedett terv
  - Legkisebb elkötelezettség *(least commitment)* stratégia
    - Csak olyan a legszükségesebb megkötéseket tesszük (pl. sorrend)

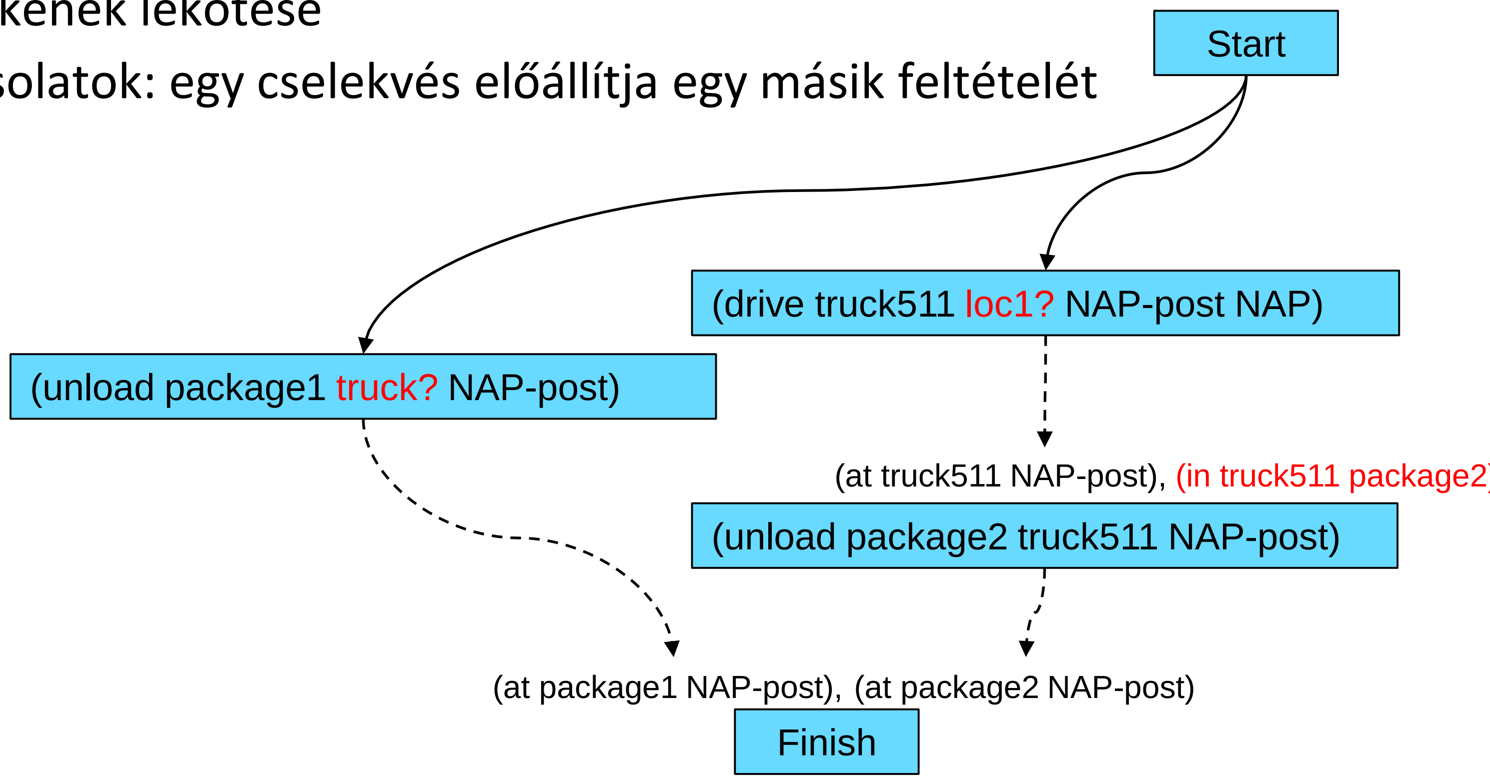




- Keresés a tervek terében
  - Minden keresési csomópont egy részleges terv
  - Kezdetben az üres terv
  - Ezt finomítjuk, mígnem elérünk egy teljesen specifikált tervet

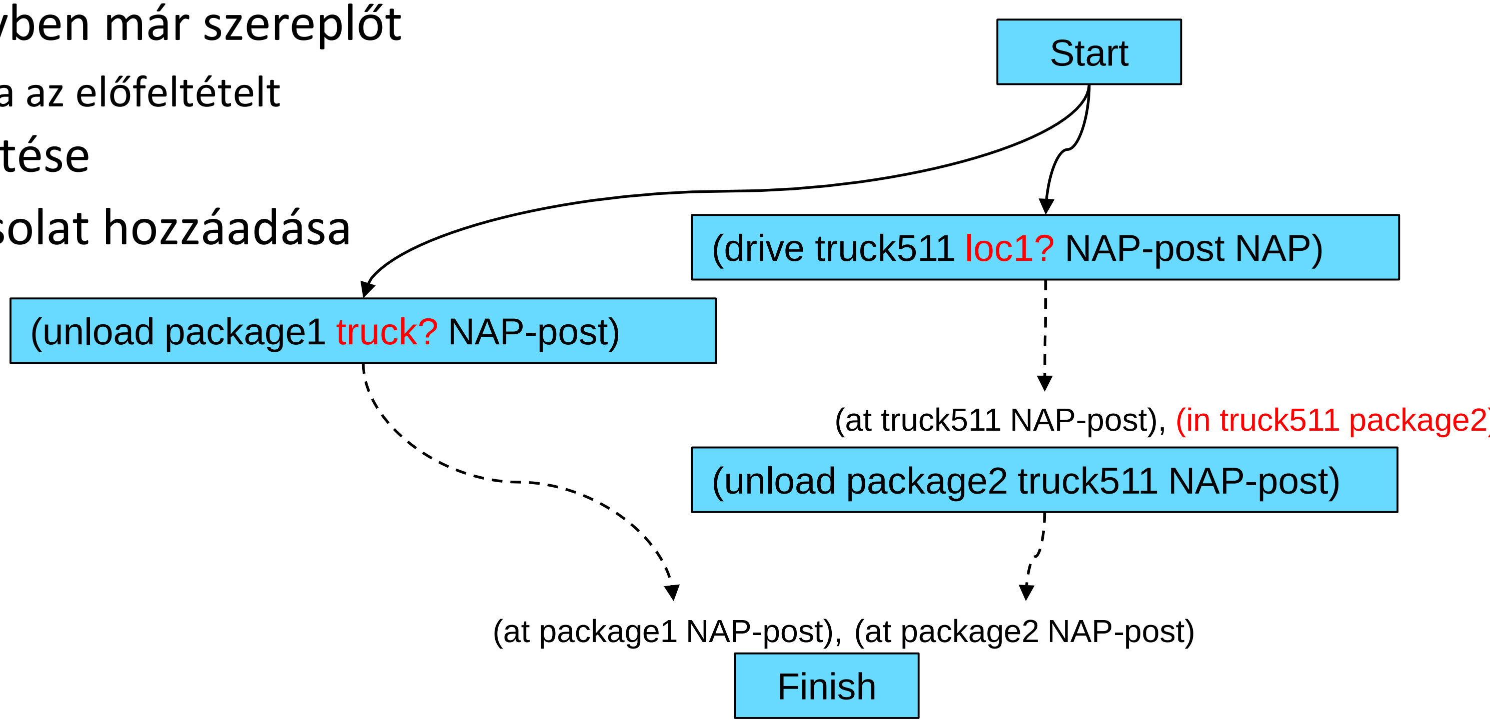
Részben rendezett tervkészítő

- A részben rendezett terv
  - Cselekvések
  - Cselekvés-párok közötti sorrendiség korlát (precedencia)
  - Változók értékének lekötése
  - Okozati kapcsolatok: egy cselekvés előállítja egy másik feltételét



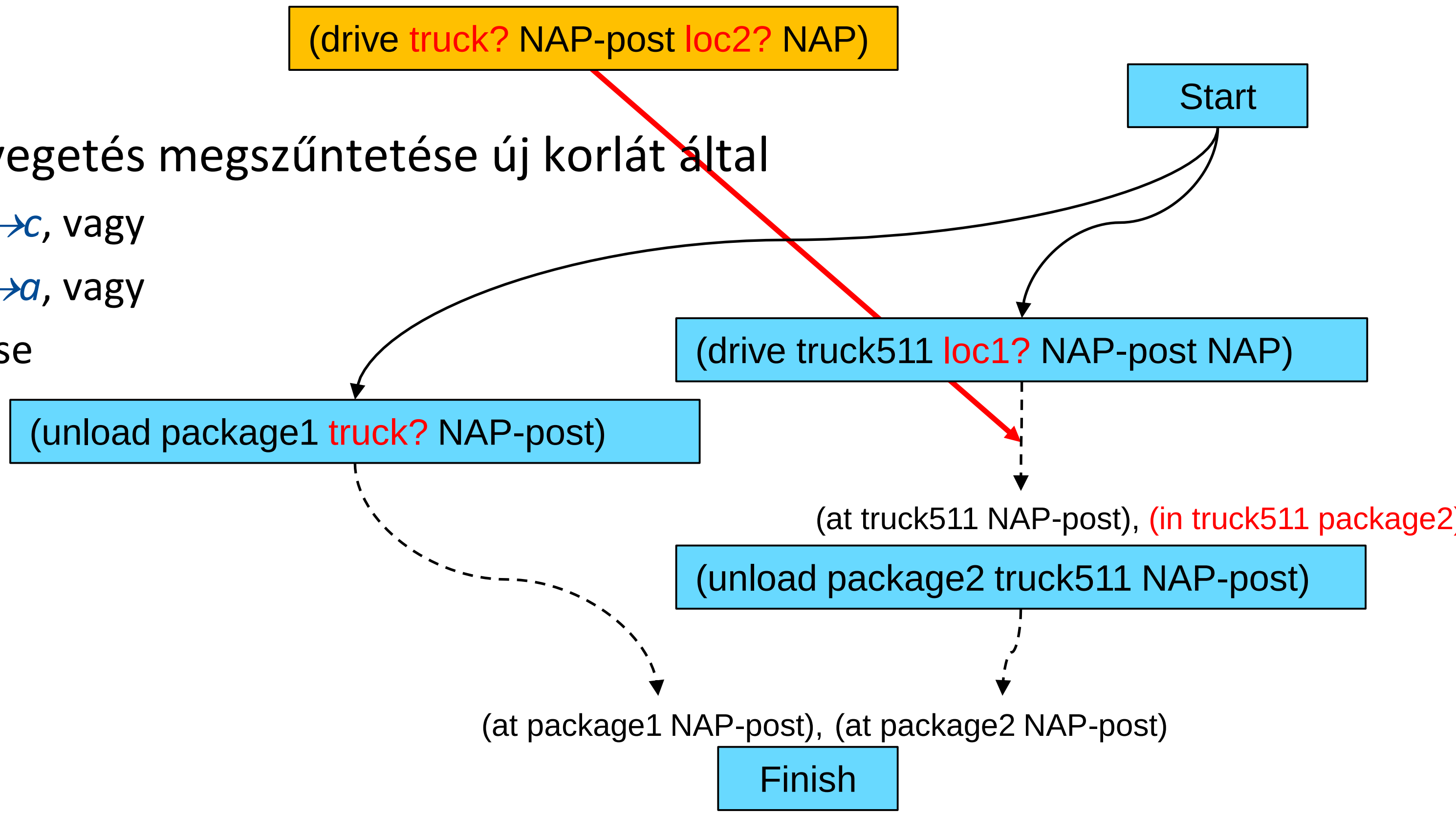
Hiányosságok a részleges tervben

- Hiányosság: kielégítetlen előfeltétel
  - Pl: (in truck511 package2)
- Orvoslása: keressünk egy cselekvést
  - Újat vagy tervben már szereplőt
    - Ami biztosítja az előfeltételt
  - Változók lekötése
  - Okozati kapcsolat hozzáadása



Hiányosságok a részleges tervben

- Hiányosság: okozati kapcsolat fenyegetése
  - Egy *a* cselekvés biztosítja *b* cselekvés *p* előfeltételét
  - Egy cselekvés *c* képes törölni *p*-t
- Orvoslása: fenyegetés megszűntetése új korlát által
  - Precedencia: $b \rightarrow c$, vagy
  - Precedencia: $c \rightarrow a$, vagy
  - Változók lekötése



MTA SZTAKI

Tervtérbeli keresés (PSP)

```
PSP(π)
    flaws ← OpenGoals(π) ∪ Threats(π)
    if flaws = ∅ then return(π)
    select any flaw φ ∈ flaws
    resolvers ← Resolve(φ, π)
    if resolvers = ∅ then return(failure)
    nondeterministically choose a resolver ρ ∈ resolvers
    π′ ← Refine(ρ, π)
    return(PSP(π′))
end
```

- A keresés operátorai *(resolvers)*
  - Cselekvés hozzáadása
  - Precedencia korlát hozzáadása
  - Változó értékének lekötése

PSP példa

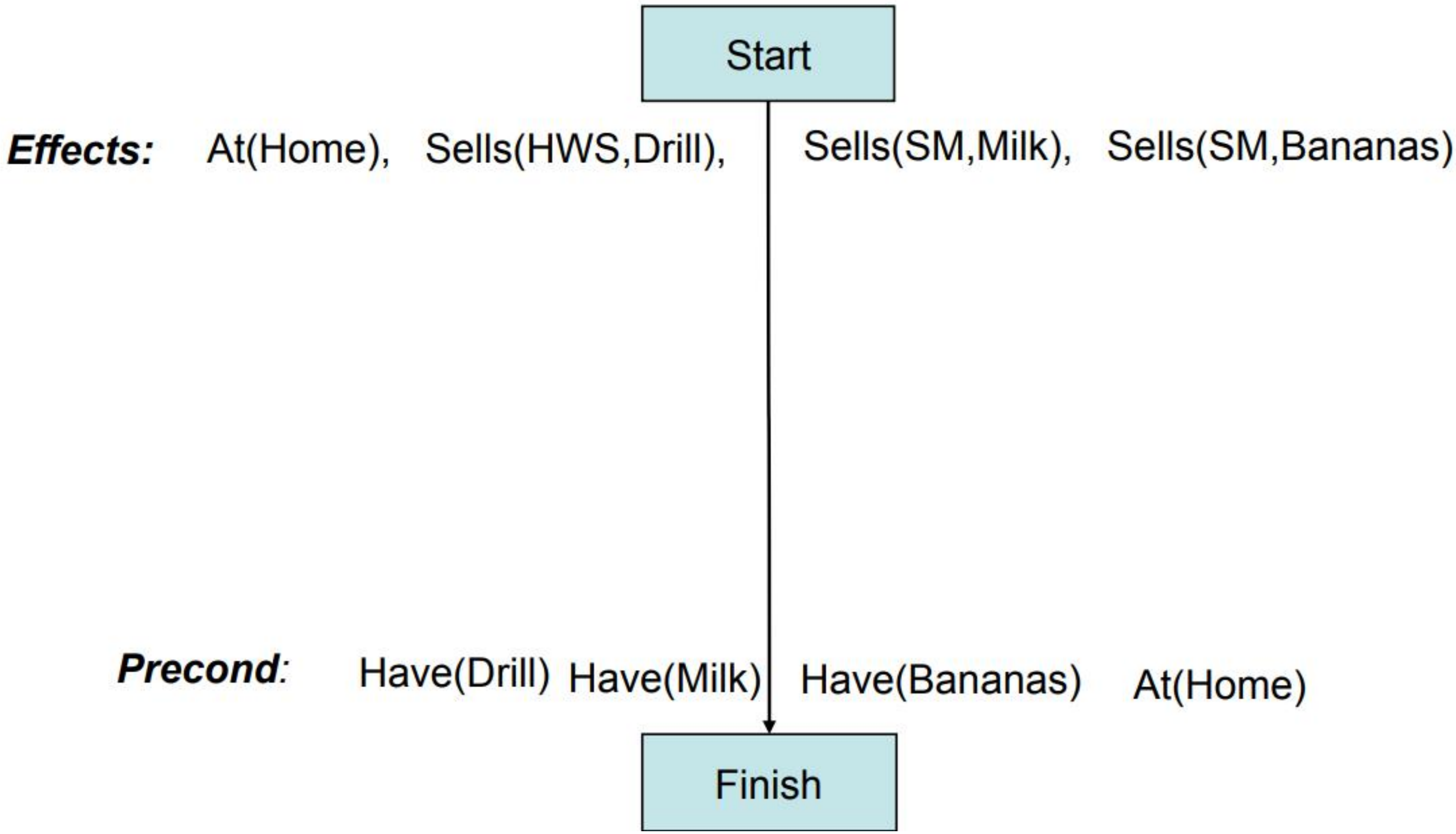
- „Bevásárlás” feladat: szerezzünk fúrógépet (kapható a szerszámboltban, HWS), tejet és banánt (szupermarketben, SM)

- Cselekvések
  - Start
    - Hatás: At(Home), sells(HWS,Drill), Sells(SM,Milk), Sells(SM,Banana)
  - Finish
    - Előfeltétel: Have(Drill), Have(Milk), Have(Banana), At(Home)
  - Go(*l*,*m*)
    - Előfeltétel: At(*l*)
    - Hatás: At(*m*), $\neg$At(*l*)
  - Buy(*p*,*s*)
    - Előfeltétel: At(*s*), Sells(*s*,*p*)
    - Hatás: Have(*p*)

Forrás: Russel & Norvig: AIMA; D. Nau: Automated planning: Theory and practice, UMD

PSP példa (2)

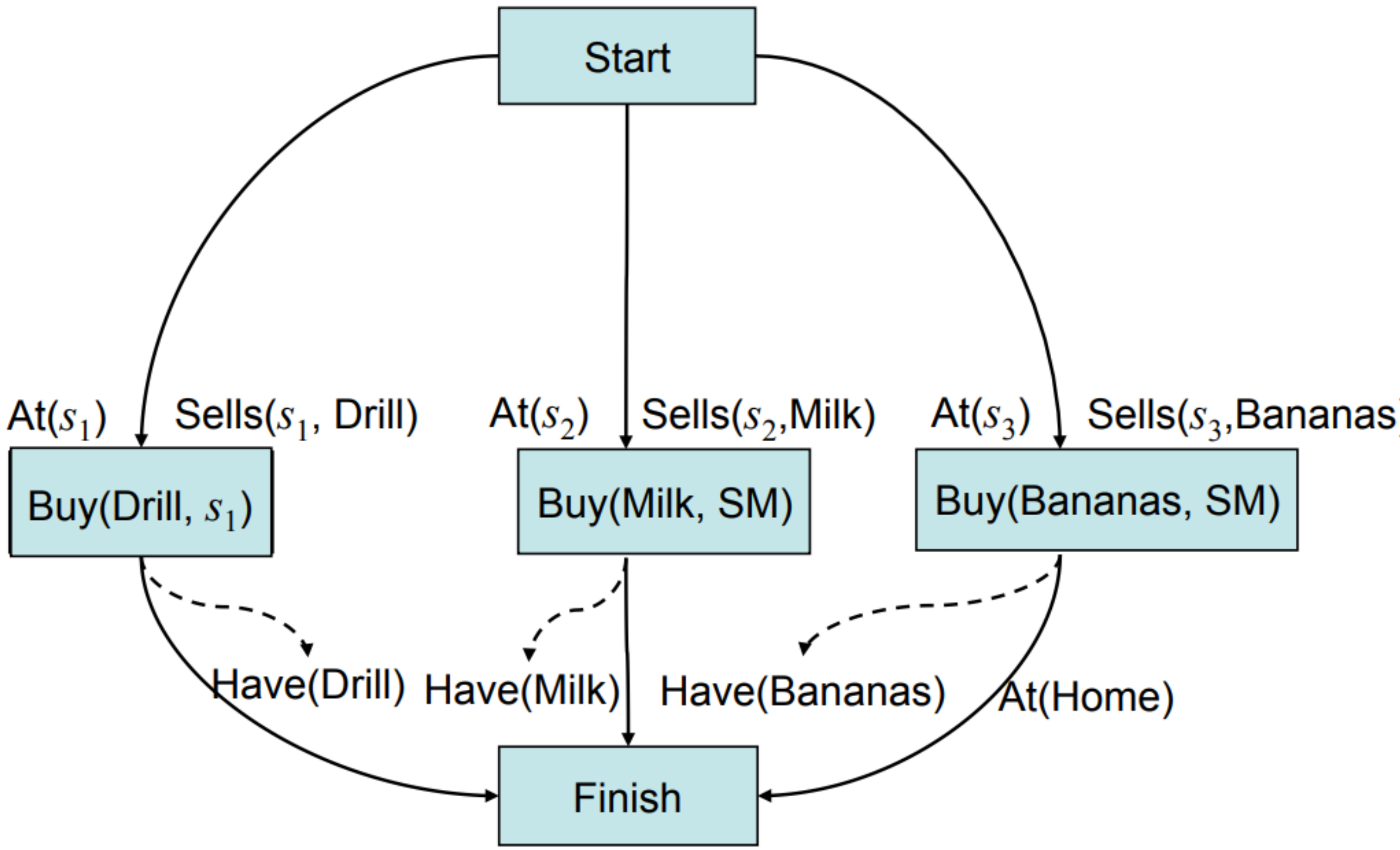
- Kezdeti terv: Start, Finish, köztük precedencia korlát



Forrás: Russel & Norvig: AIMA; D. Nau: Automated planning: Theory and practice, UMD

PSP példa (3)

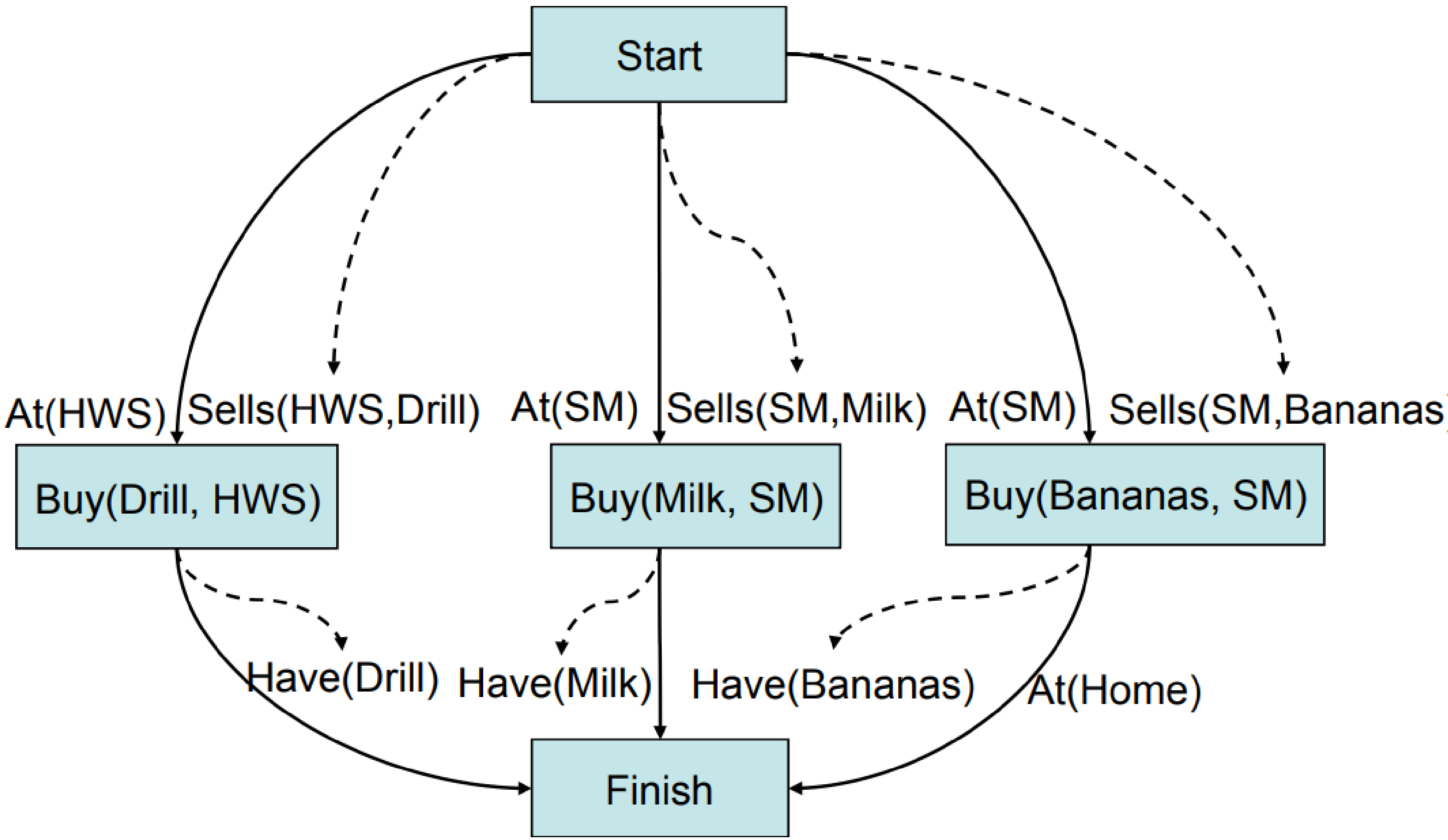
- Finish-ben a Have(.) előfeltételek biztosítása egyértelmű
  - Három lépésben három Buy cselekvés



Forrás: Russel & Norvig: AIMA; D. Nau: Automated planning: Theory and practice, UMD

PSP példa (4)

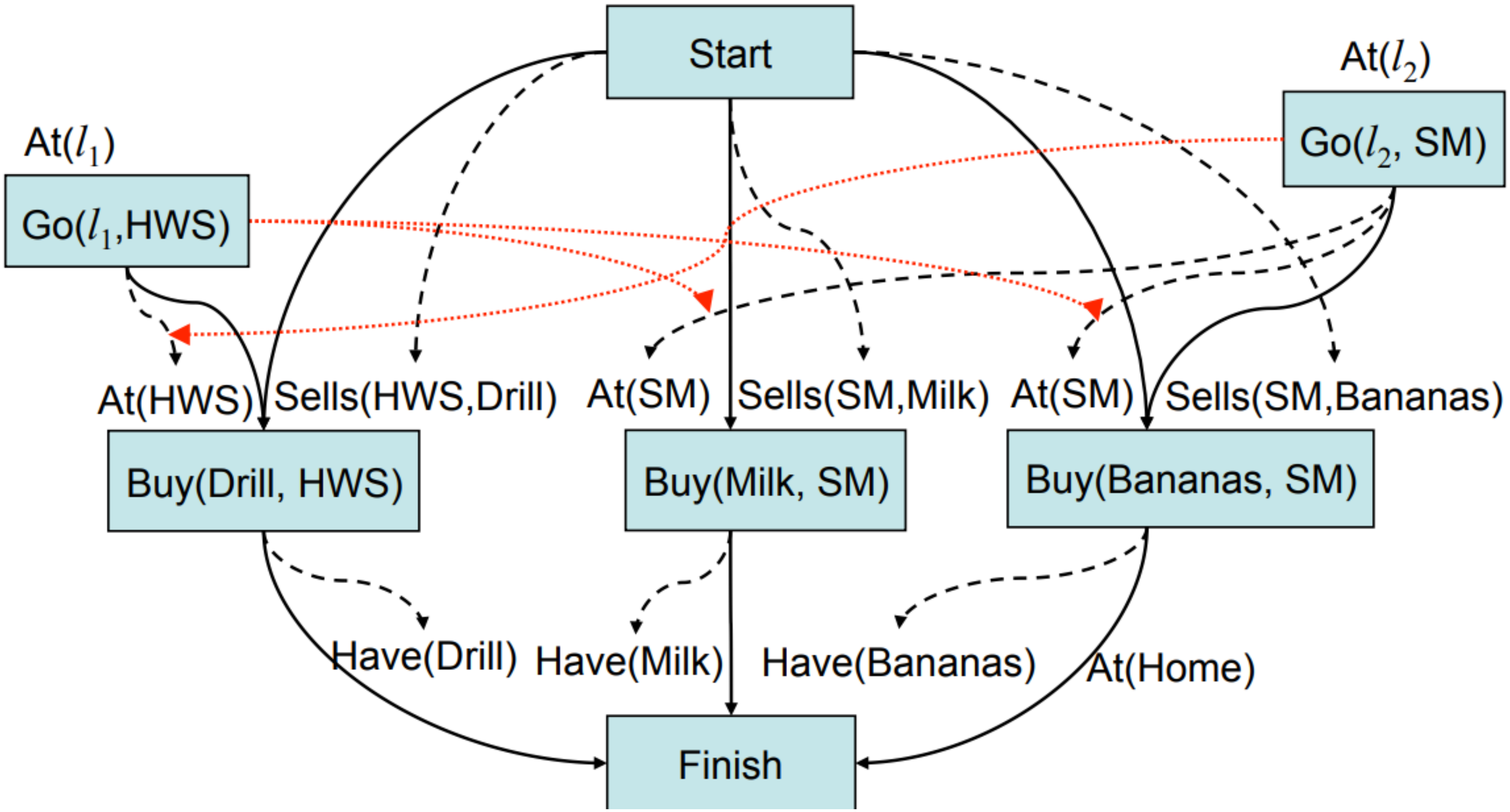
- A Sells előfeltételek csak a Start-ból biztosíthatók



Forrás: Russel & Norvig: AIMA; D. Nau: Automated planning: Theory and practice, UMD

PSP példa (5)

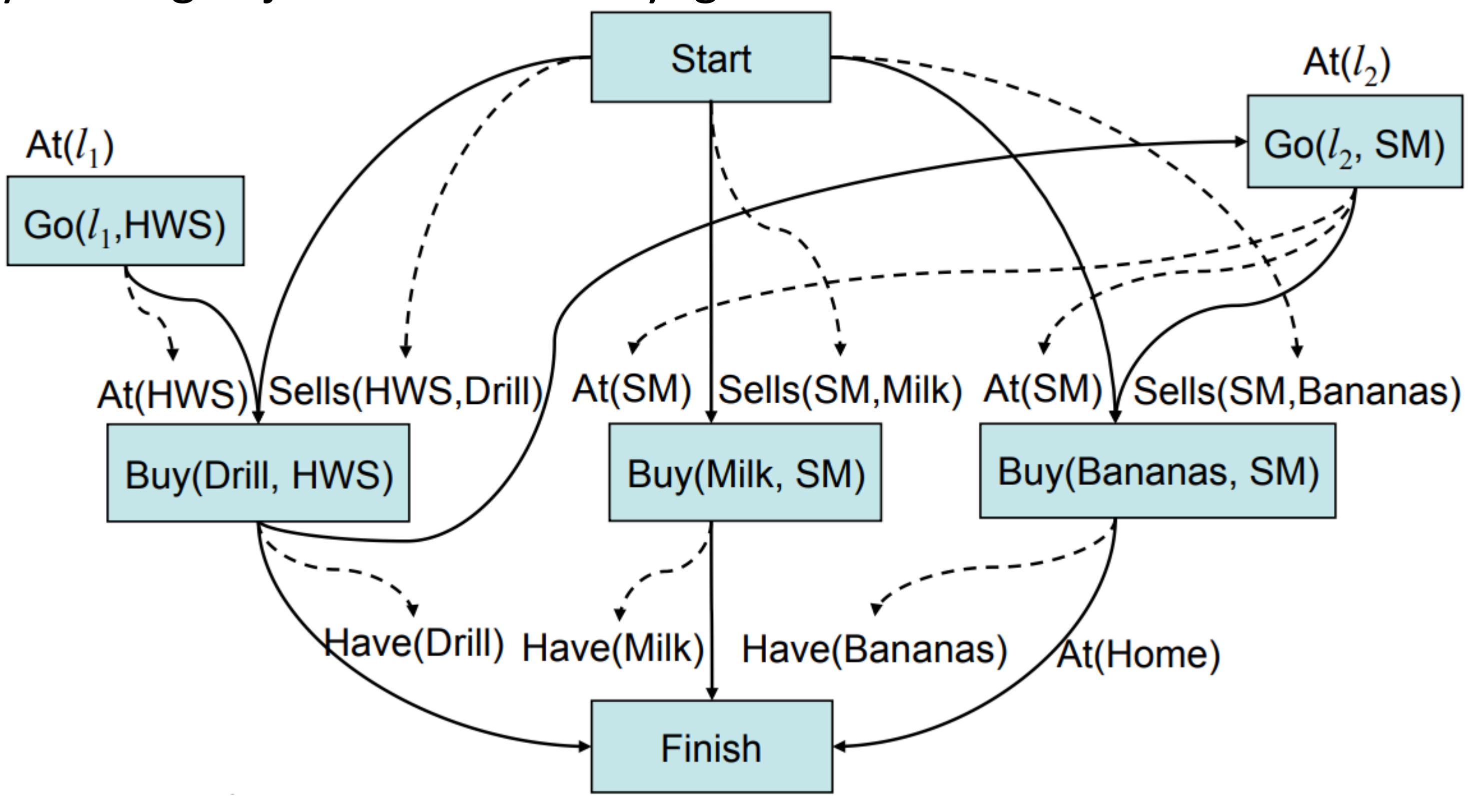
- Az At(HWS) és At(SM) csak Go(.) cselekvéssel biztosítható
  - Több okozati kapcsolat fenyegetve van!



Forrás: Russel & Norvig: AIMA; D. Nau: Automated planning: Theory and practice, UMD

PSP példa (6)

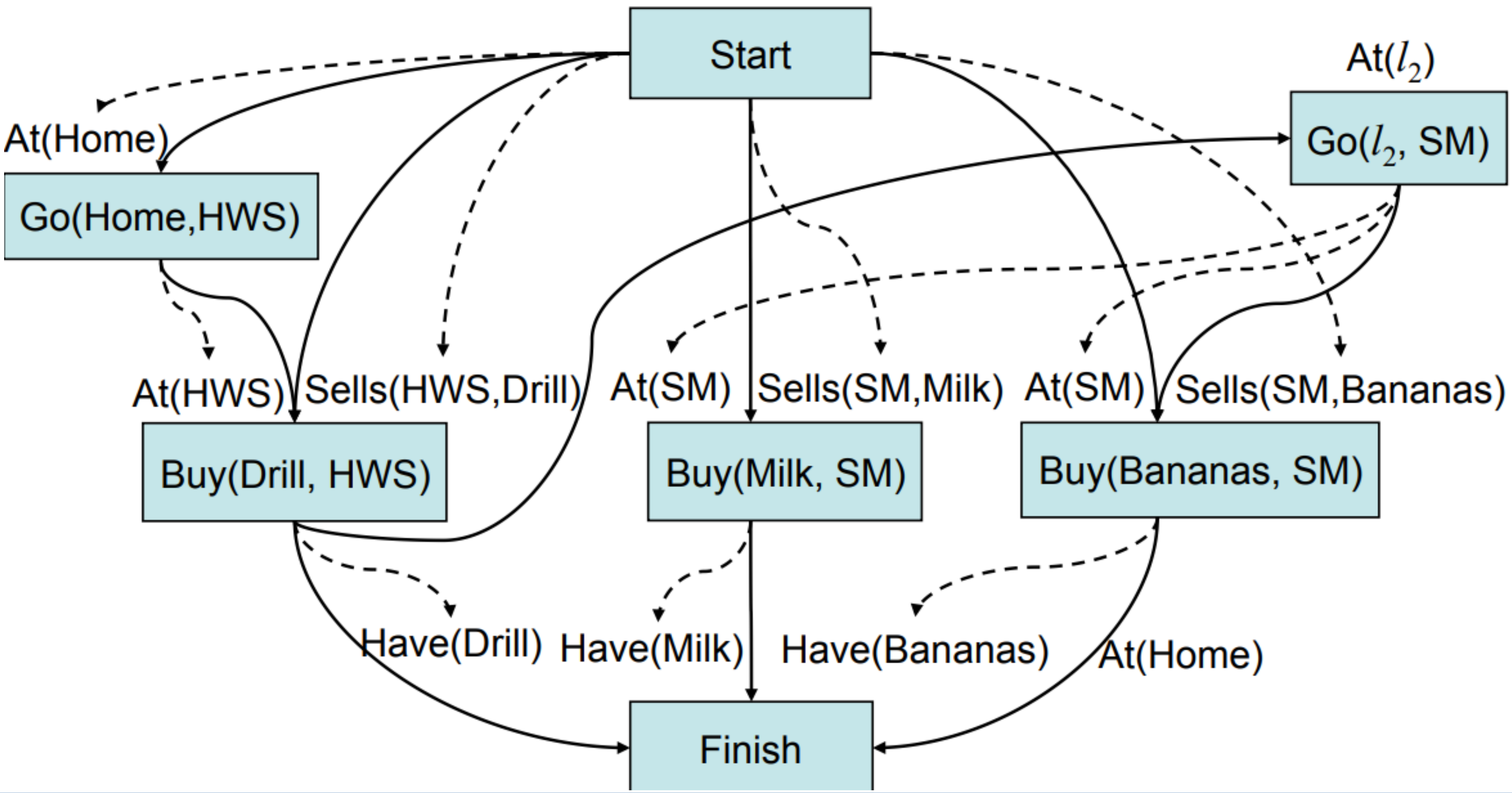
- Hogyan orvosoljuk az At(HWS) fenyegetettségét?
  - Heurisztikus döntés: Buy(Drill) $\rightarrow$ Go(l2, SM) precedencia
  - Ez egyből megoldja a másik két fenyegetést is



Forrás: Russel & Norvig: AIMA; D. Nau: Automated planning: Theory and practice, UMD

PSP példa (7)

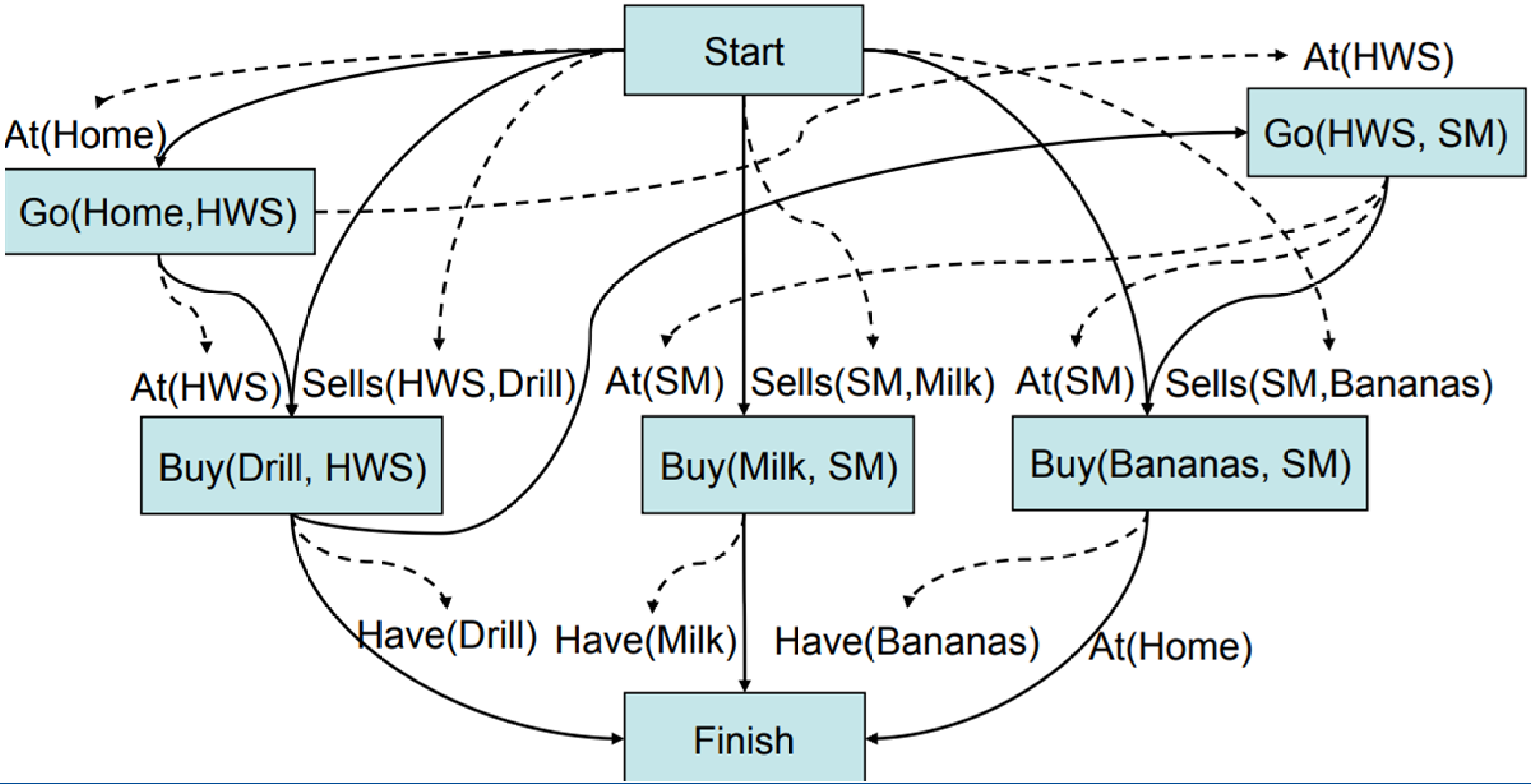
- Hogyan biztosítjuk az At(*l1*) előfeltételt?
  - A Start-ből, *l1*=Home változó lekötés hozzáadásával



Forrás: Russel & Norvig: AIMA; D. Nau: Automated planning: Theory and practice, UMD

PSP példa (8)

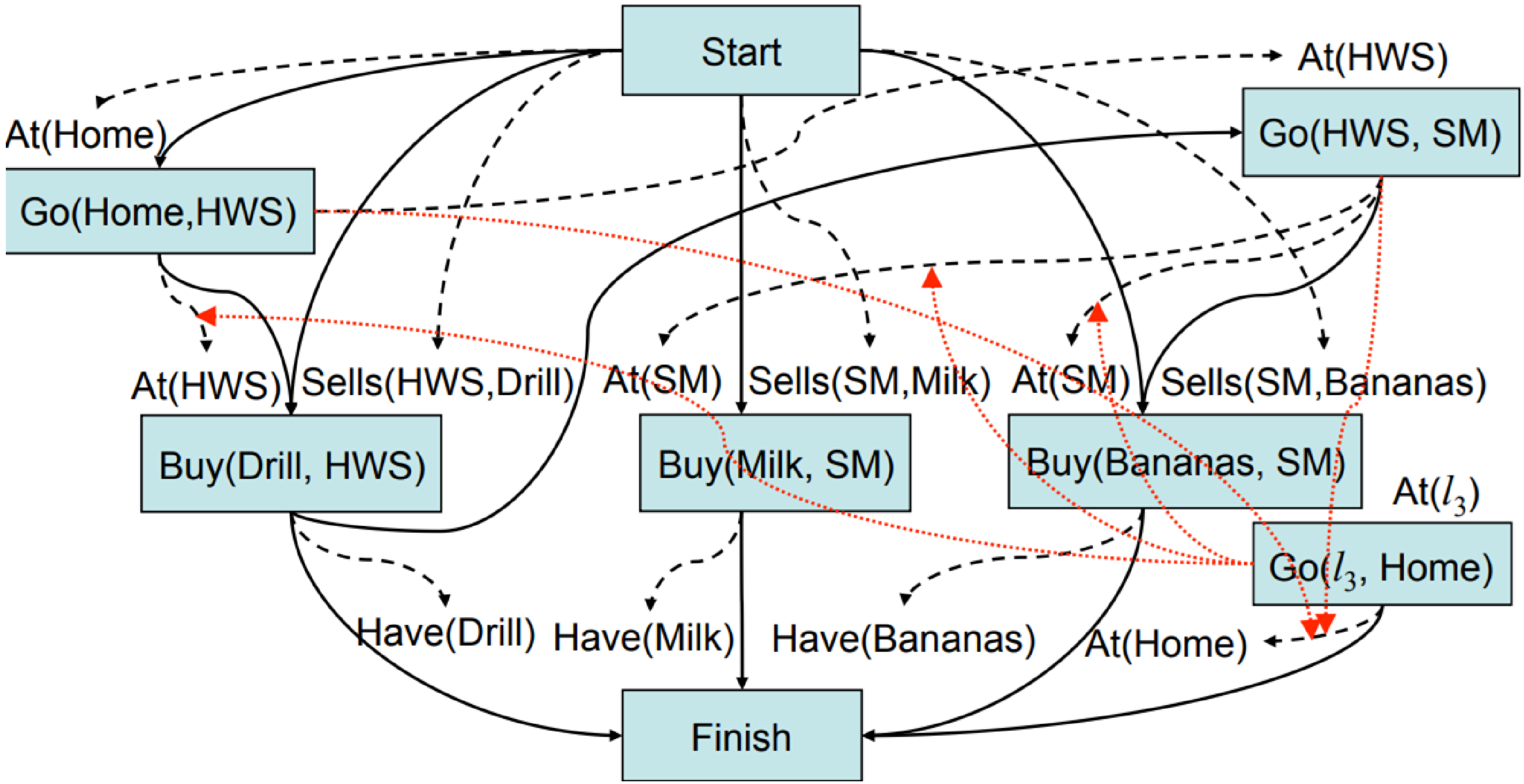
- Hogyan biztosítjuk az At(*l2*) előfeltételt?
  - A Go(Home, HWS)-ből, *l2*=HWS változó lekötés hozzáadásával



Forrás: Russel & Norvig: AIMA; D. Nau: Automated planning: Theory and practice, UMD

PSP példa (9)

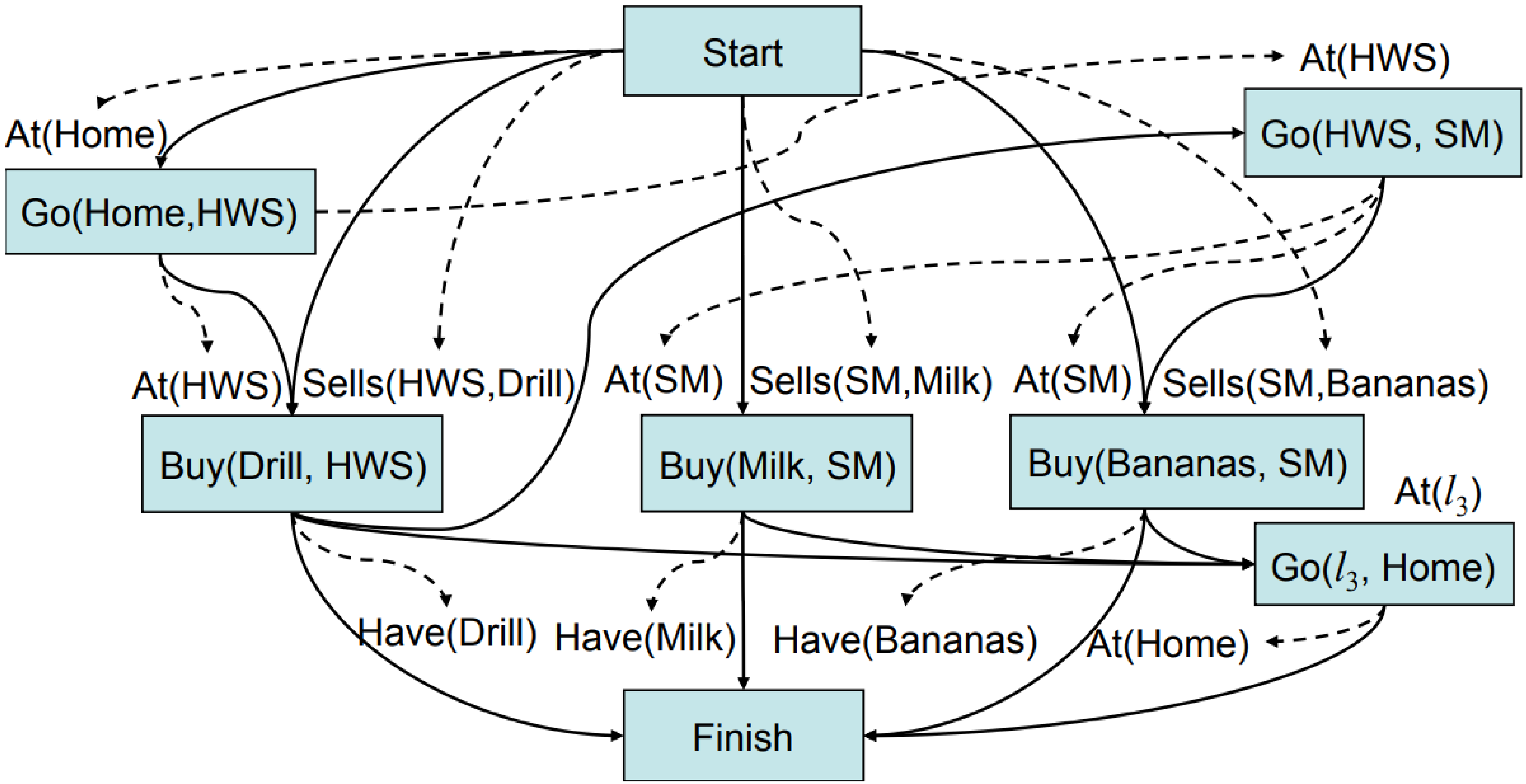
- A Finish At(Home) előfeltételét új Go(.) cselekvéssel biztosítjuk
  - Ez létrehoz néhány fenyegetést



Forrás: Russel & Norvig: AIMA; D. Nau: Automated planning: Theory and practice, UMD

PSP példa (10)

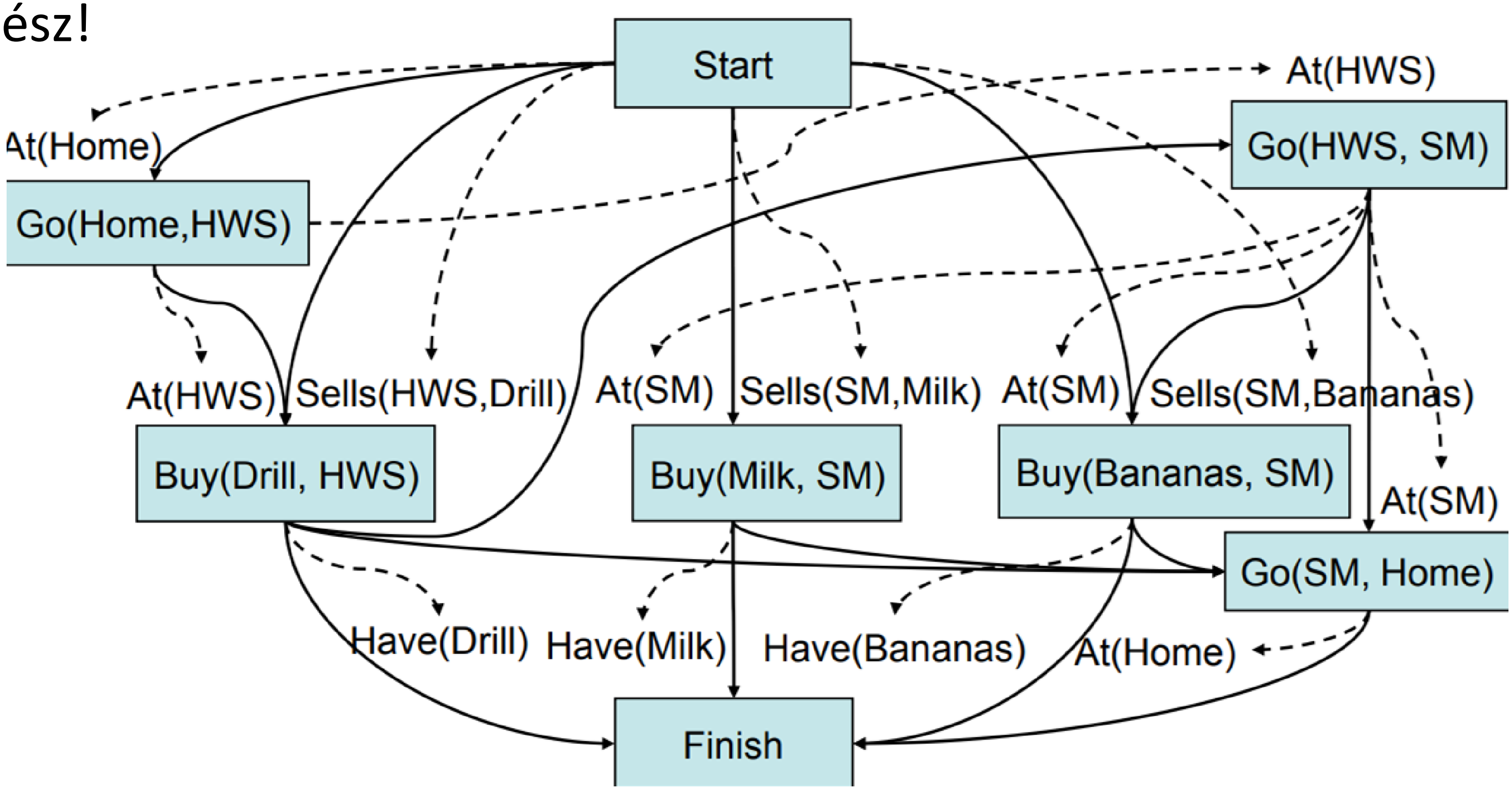
- Az At(HWS) és At(SM) fenyegetettségét új precedenciákkal orvosoljuk



Forrás: Russel & Norvig: AIMA; D. Nau: Automated planning: Theory and practice, UMD

PSP példa (10)

- Az At(l3) előfeltételt az l3=SM lekötés biztosítja a Go(HWS, SM)-ből
- Kész!



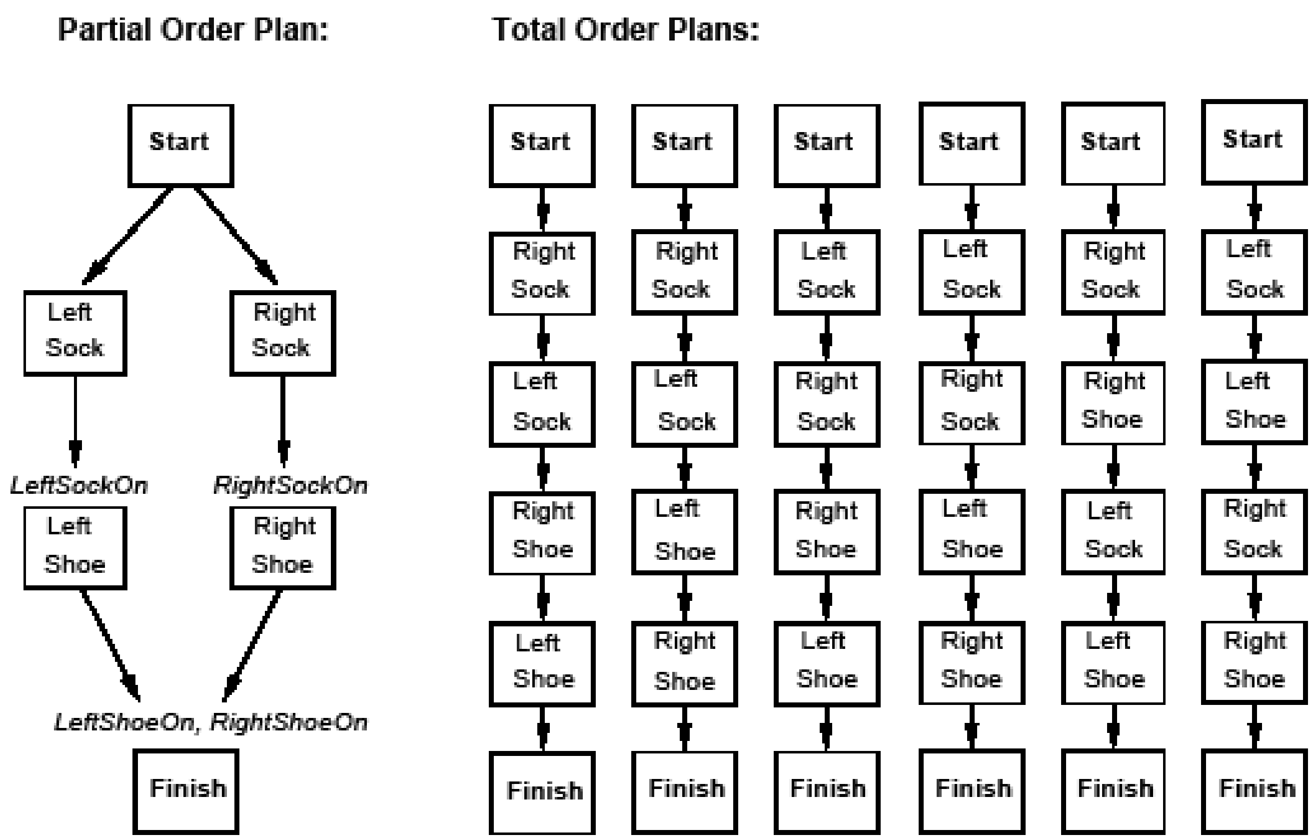
Forrás: Russel & Norvig: AIMA; D. Nau: Automated planning: Theory and practice, UMD

Heurisztikák a tervek terében

- Heurisztikák a tervek terében
  - Nehezebb jó heurisztikát találni, mint állapottérben
  - Egy lehetséges heurisztika: nyitott előfeltételek száma
    - Egy új operátor több előfeltételt kielégíthet (felülbecslés)
    - Interferencia különböző operátorok hatásai között (alulbecslés)
- Tervtér vs. állapottér: a tervezés hatékonysága
  - Sokáig nyitott kérdés: kevesebb redundancia vs. jobb heurisztikák
  - Az utóbbi időben az állapottérbeli tervezés erősen nyerésre áll, a jobb heurisztikák miatt

Részben rendezett tervkészítés

- Tervek a zokni-cipő problémájára



A PDDL cselekvéstervezés korlátai

- Optimalizálás, numerikus képességek
  - Alapvetően a cél egy megengedett terv
  - Korlátozott lehetőségek az optimalizálásra, pl.
    - Klasszikus tervezésben a terv hossza
    - Egyéb egyszerűek, pl. átfutási idő (PDDL :durative-actions)
  - Numerikus változók egyáltalán nem megengedettek
  - Korlátozott lehetőségek az numerikus függvények kezelésére az operátorokban és a célfüggvényben
    - PDDL :fluents kiterjesztés

Found Plan (output)

(fly-airplane apn2 apt7 apt1)
(fly-airplane apn1 apt8 apt1)
(fly-airplane apn1 apt1 apt2)
(fly-airplane apn1 apt2 apt3)
(fly-airplane apn1 apt3 apt4)
(fly-airplane apn1 apt4 apt5)
(fly-airplane apn1 apt5 apt6)
(fly-airplane apn2 apt1 apt2)
(fly-airplane apn2 apt2 apt3)
(fly-airplane apn2 apt3 apt5)
(fly-airplane apn2 apt5 apt6)
(drive-truck tru8 pos8 apt8 cit8)

apt8 → apt1 → apt 2 → apt3 → apt4 → apt5 → apt6 ???!

A PDDL cselekvéstervezés korlátai (2)

- Számítási hatékonyság
  - Komplex, nagyméretű feladatokon önmagában nem hatékony
  - Nehezen integrálható feladat-specifikus heurisztikákkal és keresési technikákkal
  - Ilyen feladathoz jobban megéri dedikált megoldót fejleszteni

- Tervkészítés bizonytalan környezetben
  - A bemutatott tervezők nem kezelik a bizonytalanságot
  - Jelentős eredmények bizonytalan környezet esetére is
    - Eshetőségi tervkészítés, Markov modellek, stb.

Összegzés

- Általános, alkalmazásfüggetlen cselekvéstervezés
  - Egy letisztult, az elsőrendű logikán alapuló nyelven
- Deklaratív megközelítés
  - Nincs szükség algoritmus leprogramozására
- Gyors fejlesztés
  - Egyszerű vagy kisméretű feladatok megoldására
  - „Rapid prototyping”
- De számottevő gyakorlati korlátok
  - A nyelv kifejezőképessége
  - Számítási hatékonyság

Ajánlott eszközök, olvasmányok

Online, felhő-alapú PDDL cselekvéstervező:

http://solver.planning.domains/

Korszerű Linux-os PDDL cselekvéstervező:

http://www.fast-downward.org/

Olvasmány egy legendás heurisztikus cselekvéstervező algoritmusról:

J. Hoffmann, B. Nebel, The FF planning system: Fast plan generation through heuristic search, *Journal of Artificial Intelligence Research* 14:253-302, 2001.

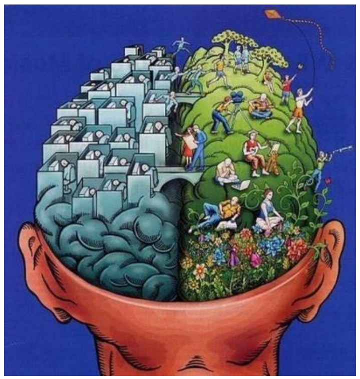
https://arxiv.org/abs/1106.0675

Dana Nau (University of Maryland) cselekvéstervezés kurzusának anyagai:

https://www.cs.umd.edu/~nau/planning/slides/

Kovács András
akovacs@sztaki.hu

Budapesti Műszaki és Gazdaságtudományi Egyetem
Méréstechnika és Információs rendszerek Tanszék



Valószínűségi hálók

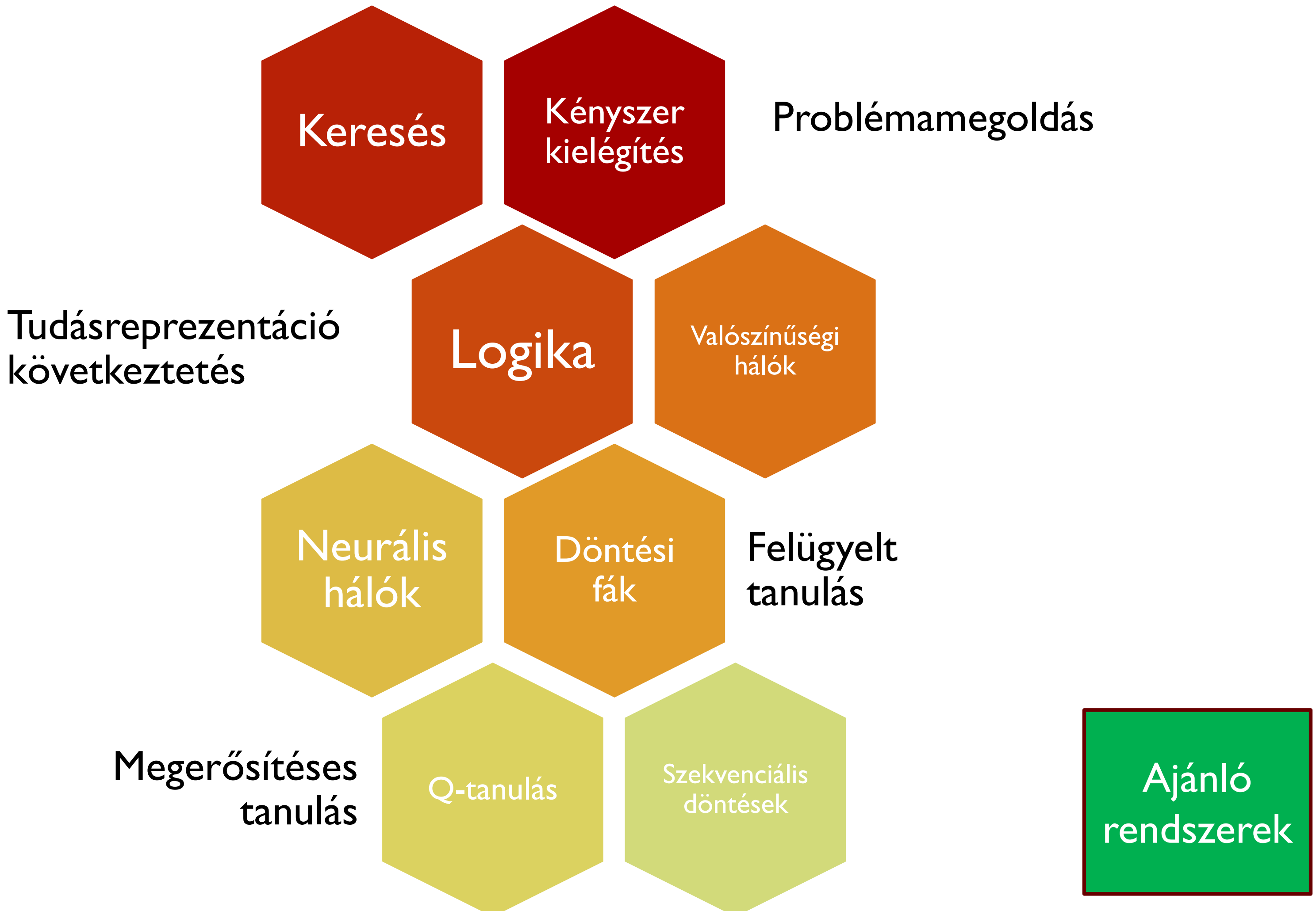
Előadó: Dr. Hullám Gábor

Előadás anyaga: Dr. Dobrowiecki Tadeusz
Dr. Hullám Gábor

Valószínűségi modellezés alapjai

MŰEGYETEM 1782

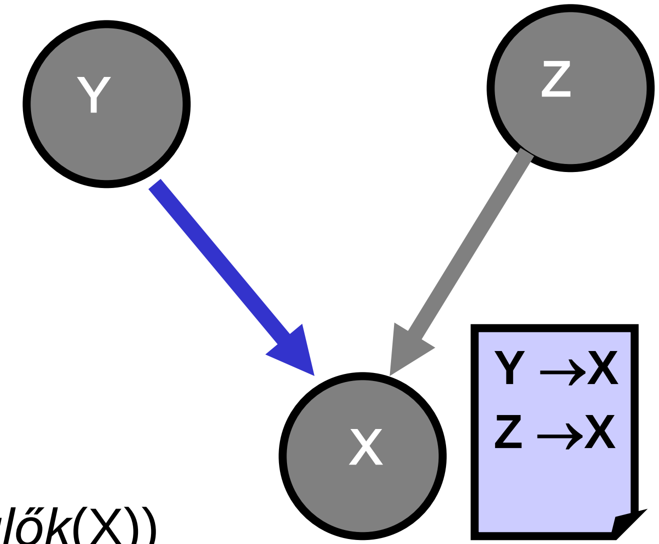
Témakörök



Keresés
Kényszer kielégítés
Problémamegoldás
Tudásreprezentáció következtetés
Logika
Valószínűségi hálók
Neurális hálók
Döntési fák
Felügyelt tanulás
Megerősítéses tanulás
Q-tanulás
Szekvenciális döntések
Ajánló rendszerek

Valószínűségi háló (Bayes-háló) → egy gráf

1. Csomópontok: valószínűségi változók egy halmaza
2. Csomópontok között: irányított élek halmaza:
   Az Y csomópontot az X csomóponttal összekötő nyíl →
   az *Y-nak közvetlen befolyása* van az *X*-re
3. Minden csomópont:
   feltételes valószínűségi tábla →
   szülők hatása a csomópontra P(X | *Szülők*(X))



4. A gráf nem tartalmaz irányított kört (= **irányított, körmentes gráf = DAG**).

Valószínűségi változó = egy állítás a problémáról

Pl. X = HMG szint magas (két érték: I/N), vagy
X = HMG szint (több érték: Magas / Közepes / Alacsony)

Példa

Otthonunkban egy új **riasztót** szereltünk fel. Ez megbízhatóan észleli a **betöréseket**, de kisebb **földrengés** esetén is jelez.
Két szomszédunk, **János** és **Mária**, megígérték, hogy **felhívnak** a munkahelyünkön, ha meghallják a riasztónkat.
János mindig felhív, ha szól a riasztó, de néha összekeveri a telefoncsörgést a riasztó csengésével és ekkor is telefonál.
Mária viszont, mivel szereti hangosan hallgatni a zenét, néha meg sem hallja a riasztót.
Mi tehát a hívások bekövetkezte vagy hiánya alapján szeretnénk megbecsülni a betörés valószínűségét.

(Bináris) változók (tények):

Betörés (megtörtént): Igen/ Nem

Földrengés (megtörtént): Igen/ Nem

Riasztás (megszólalt): Igen/ Nem

János telefonál(t-e): Igen/ Nem

Mária telefonál(t-e): Igen/ Nem

Háló topológiája = egy absztrakt tudásbázis

sok esetben érvényes marad, mivel **általános ok-okozati** folyamatokat ír le az adott problémakörben.

A betöréses háló esetén a topológia azt mutatja, hogy a betörés és a földrengés közvetlenül hat a riasztóra, befolyásolja megszólalásának valószínűségét, ellenben János vagy Mária hívásának bekövetkezése csak magán a riasztón múlik – azaz a háló tartalmazza azt a feltevést, hogy ők közvetlenül nem vesznek észre sem a betöréseket, sem a földrengéseket.

Feltételes valószínűségi tábla – FVT

minden egyes csomópontra.

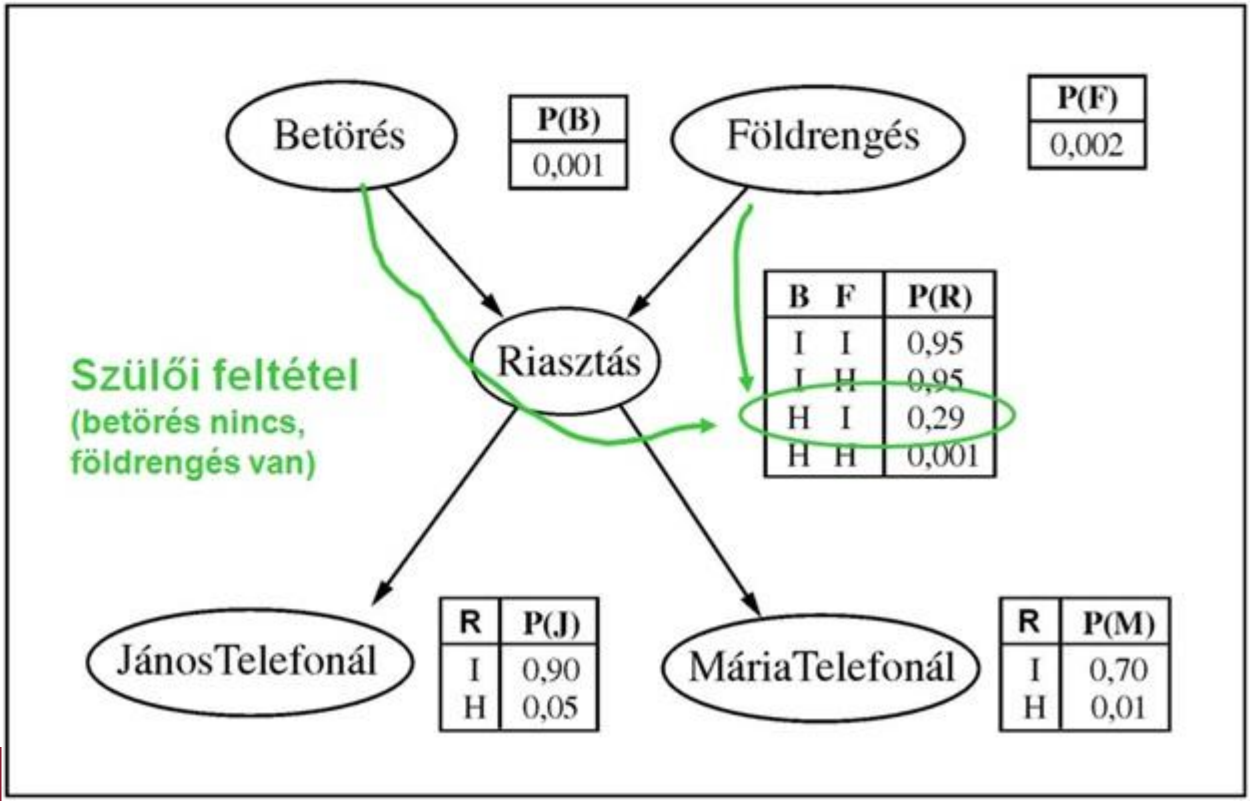
Egy sor a táblázatban az egyes csomóponti értékek feltételes valószínűsége az adott sorhoz tartozó **szülői feltétel** esetén.

Szülői feltétel: a szülő csomópontok értékeinek egy lehetséges kombinációja (**egyfajta elemi esemény a szülők között**).

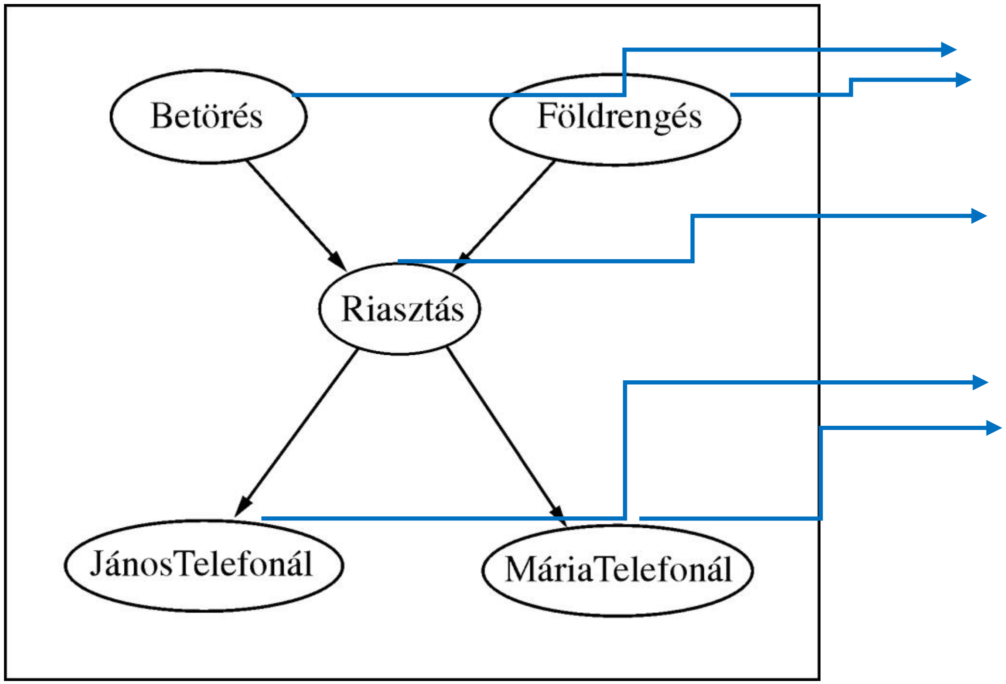
Általánosabban, ha a változó **bináris**, és ha *n* **bináris** szülője van:

2^n -1 valószínűség adható meg a feladat alapján.

Szülő nélküli csomópont – a változó egyes értékeinek a priori valószínűségei (ha bináris, akkor csak egy).



Együttes eloszlás:
5 változó
= 2^5-1 = **31 db** valószínűség



Bayes háló:
2 x 1 db a priori
+
1 x 4 db szülői feltétel
+
2 x 2 db szülői feltétel
= 2 + 4 + 4 = **10 db**

Elég-e?

Általános eset:

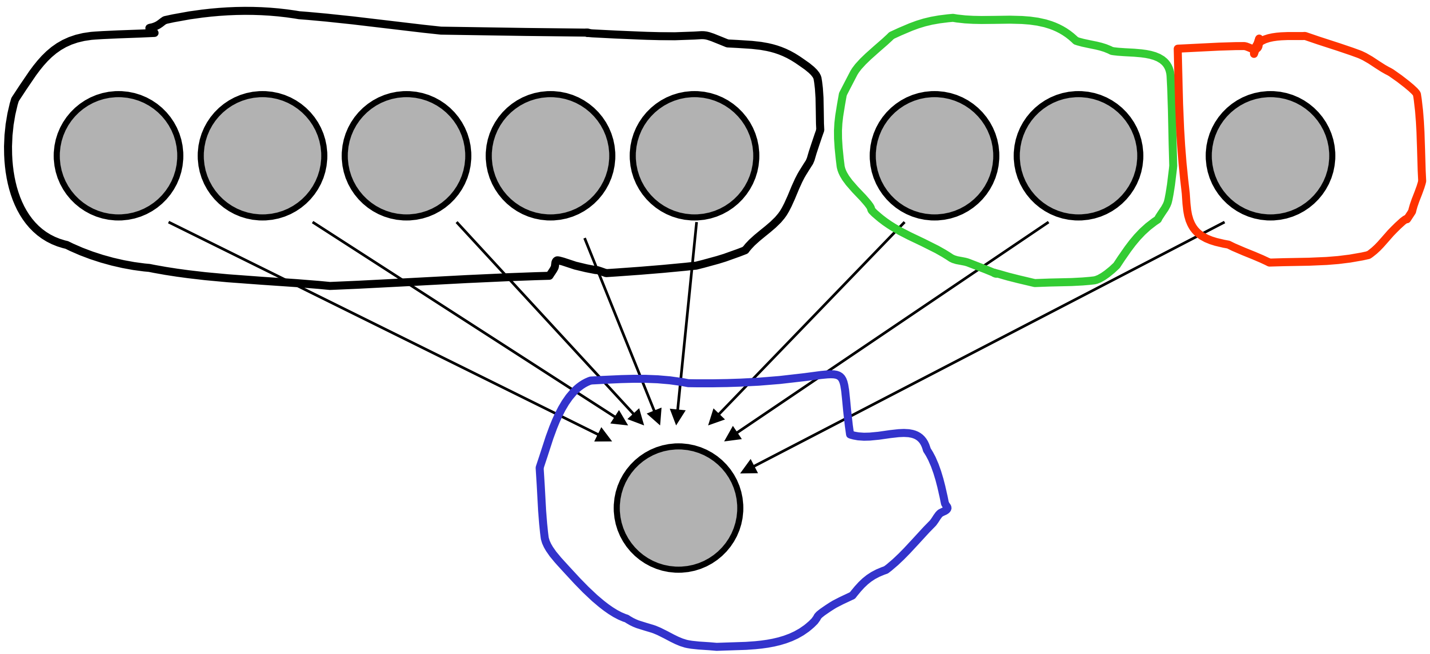
Mi van, pl. ha

5 db szülő csomópont bináris értékű

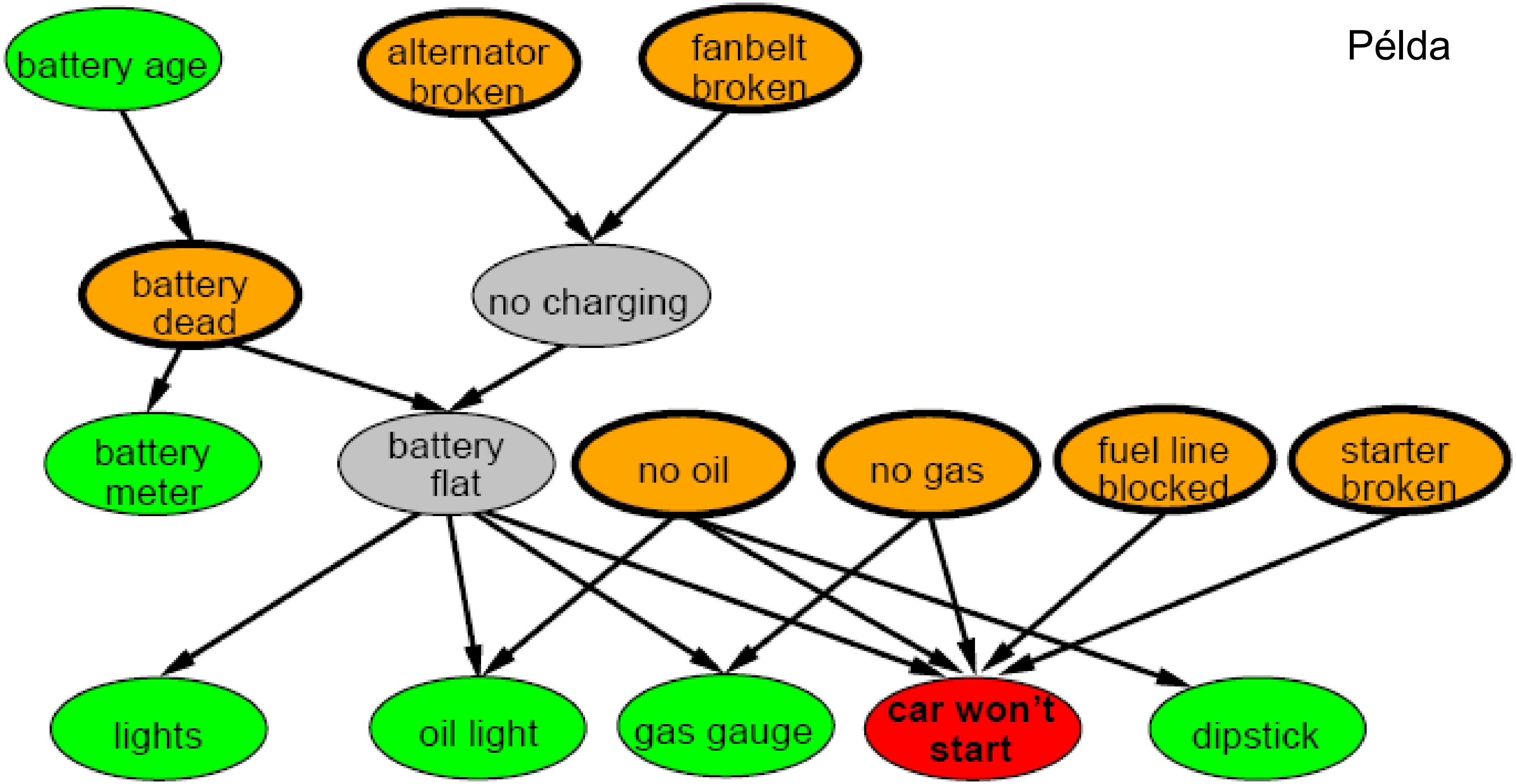
2 db szülő csomópont 3-as értékű

1 db szülő csomópont 4-es értékű és

az eredmény csomópont egy 5-ös értékű változó?

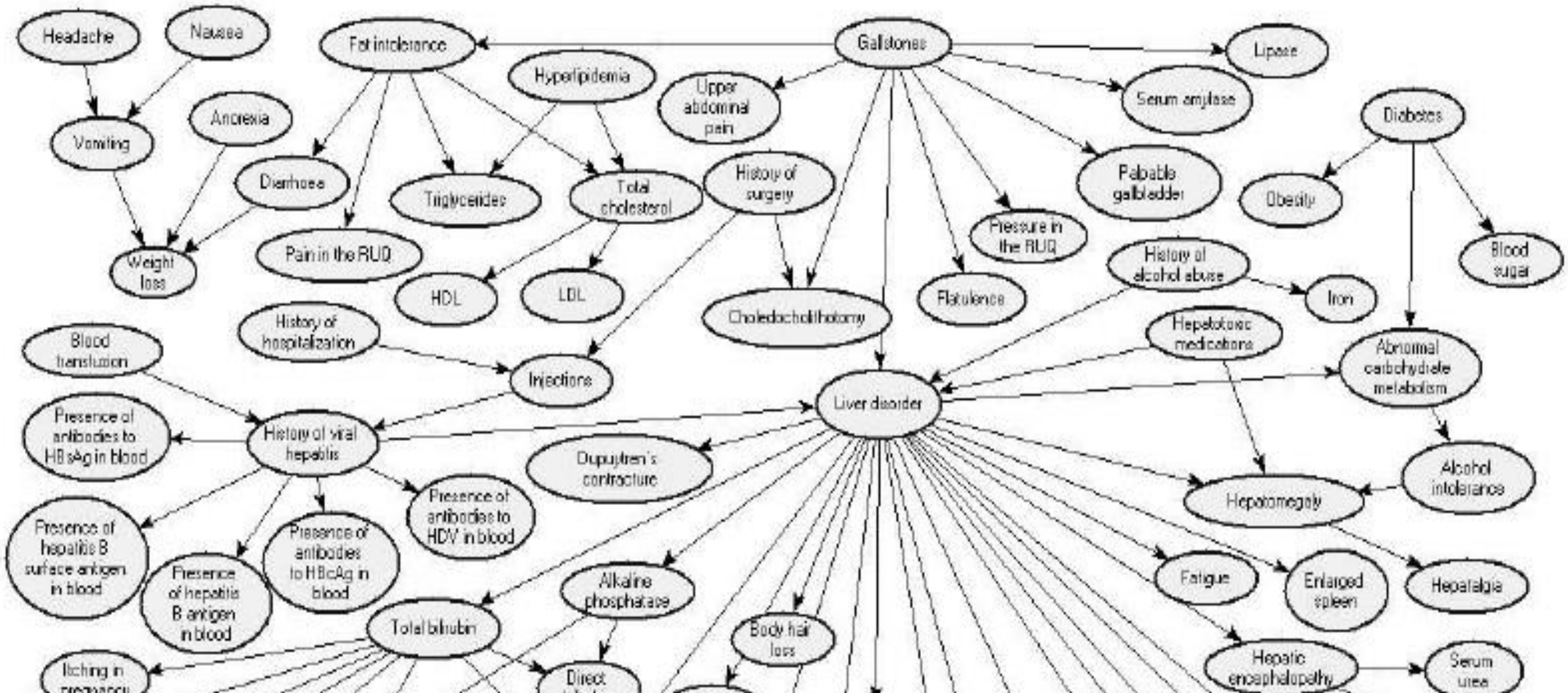


Példa

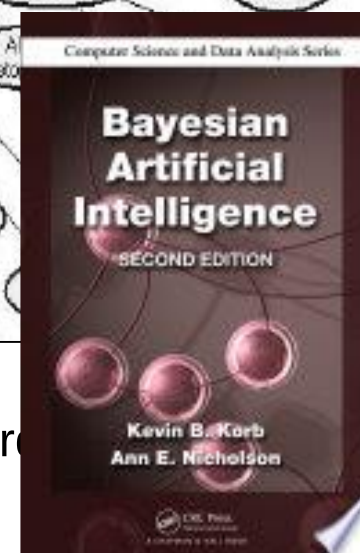


16 (bináris) változó = 2^{16}-1 = 16383 valószínűség

háló = 7 + 4 x 2 + 4 x 4 + 1 x 32 = 63 valószínűség (x 260)



Májdiagnózis
A. Onisko et al., Extension of the HEPAR II Model to Multiple-Disor
Diagnosis (2000)



Biológiai és környezeti folyamatok modellezése

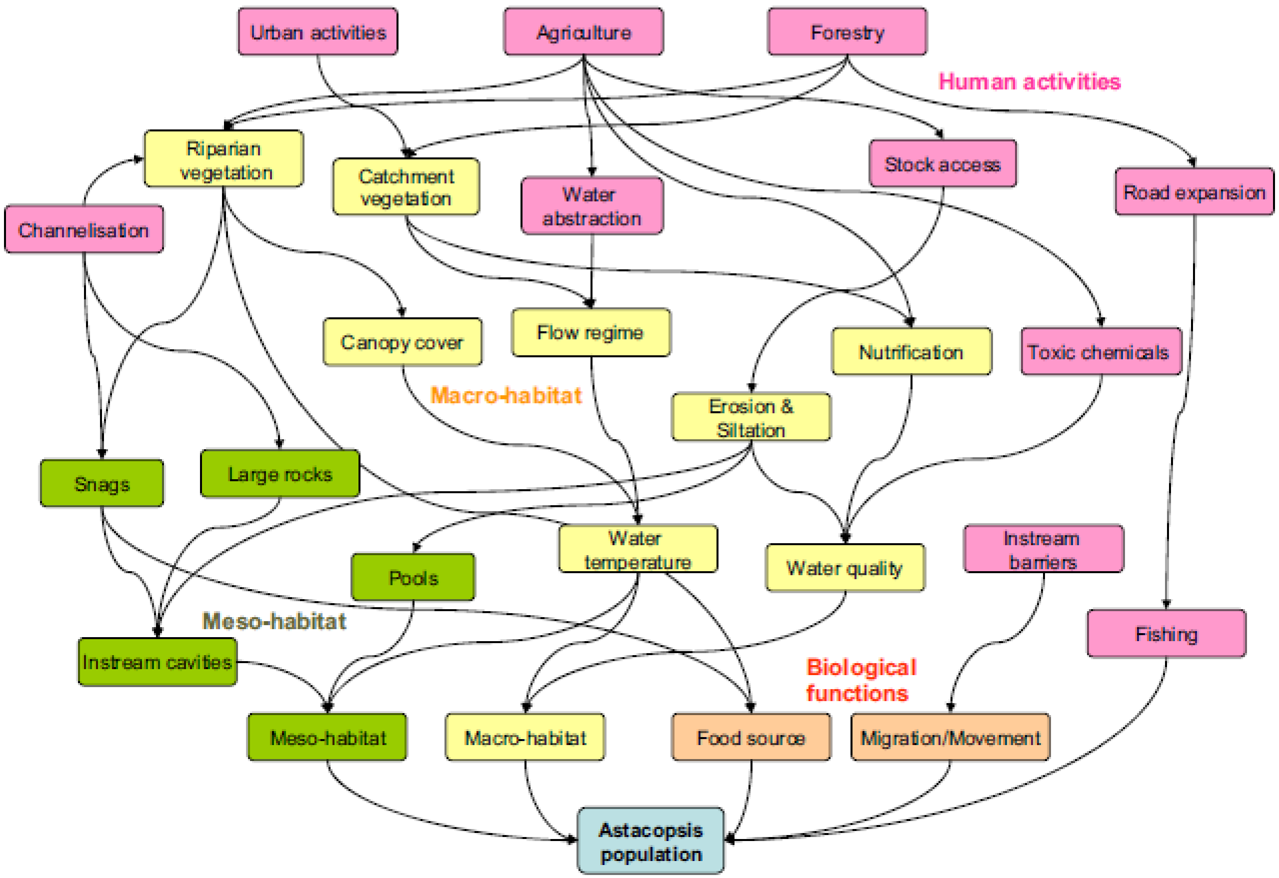
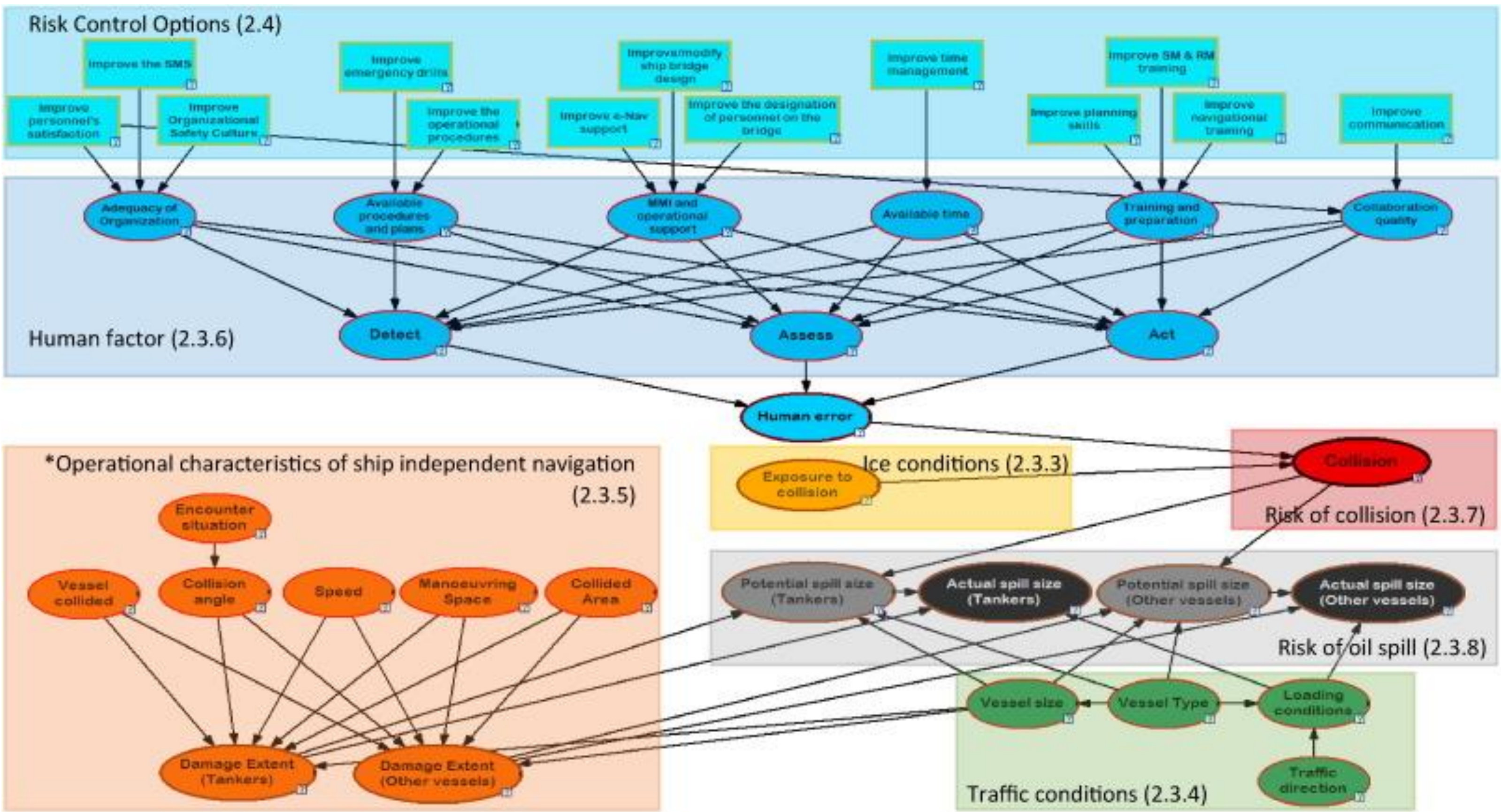


Fig. 2. Conceptual model of the key factors affecting *Astacopsis gouldi* populations.

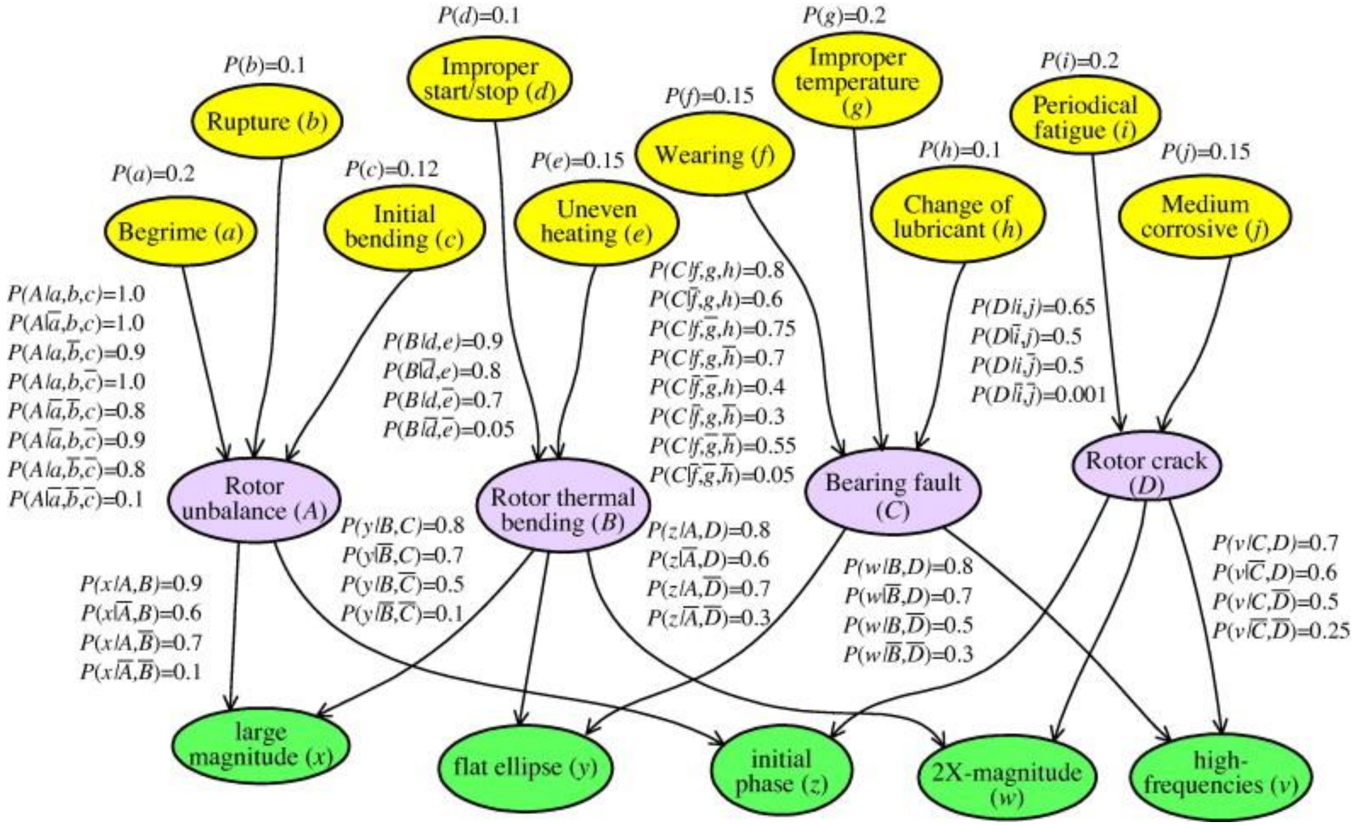
Serena H. Chen & Carmel A. Pollino: Good practice in Bayesian network modelling

Kockázatmodellezés



Osiris A.Valdez et al. : Risk management model of winter navigation operations, Marine Pollution Bulletin,Volume 108, Issues 1–2, 15 July 2016, Pages 242-262

Hibamodellezés és következtetés



- Bin GangXu: Intelligent fault inference for rotating flexible rotors using Bayesian belief network, Expert Systems with Applications,Volume 39, Issue 1, January 2012, Pages 816-822

Modelltípusok - kitérő

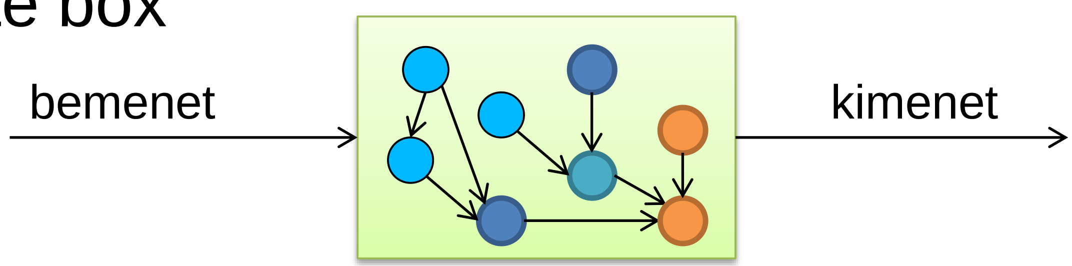
- Black box

Pl.:Neurális hálók

A modell belső működését nem / nehezen lehet értelmezni

- White box

Pl.:Bayes- hálók



A modell belső működése értelmezhető

A valószínűségi hálók szemantikája

(1) a háló = **együttes valószínűségi eloszlás** egy leírása,

(2) a háló = **feltételes függetlenségekről szóló állítások** együttese.

A két szemlélet **ekvivalens**
az első abban segít, hogy hogyan **hozzunk létre** egy hálót
a második abban, hogy hogyan **tervezzünk** következtetési eljárásokat.

Feltételes függetlenség

Ha a szülő (szülői feltétel) ismert, nem érdekes az ős:

$$\mathbf{P}(X_i \mid X_{i-1}, \ldots, X_1) = \mathbf{P}(X_i \mid \mathit{Szülők}(X_i)),\ i=1\ldots n,$$
$$\text{ha } \mathit{Szülők}(X_i) \subseteq \{ X_{i-1},\ldots,X_1 \}.$$

Ez utóbbi könnyen teljesíthető a csomópontok olyan sorszámozásával, ami konzisztens a gráf implicit részleges rendezésével.

$$\mathbf{P}(X_i \mid \mathit{Szülők}(X_i), \mathit{Ősök}(X_i)) = \mathbf{P}(X_i \mid \mathit{Szülők}(X_i))$$

Az együttes valószínűségi eloszlásfüggvény leírása

A valószínűségi hálóban található információk alapján:
az együttes valószínűségi eloszlás bármely bejegyzése kiszámítható.

Egy bejegyzés értéke (globális szemantika):

$$P(X_1 \ldots X_n) = \prod_{i=1 \ldots n} P(X_i \,|\, Szülők(X_i)),\ i=1\ldots n$$

Az együttes valószínűségi eloszlás minden bejegyzése felbontható a FVT megfelelő elemeinek a szorzatára.
FVT-k: az együttes valószínűségi eloszlás dekomponált leírása.

Egy elemi esemény: pl. a riasztó megszólal, de sem betörés, sem földrengés nem volt, azonban János is és Mária is telefonál:

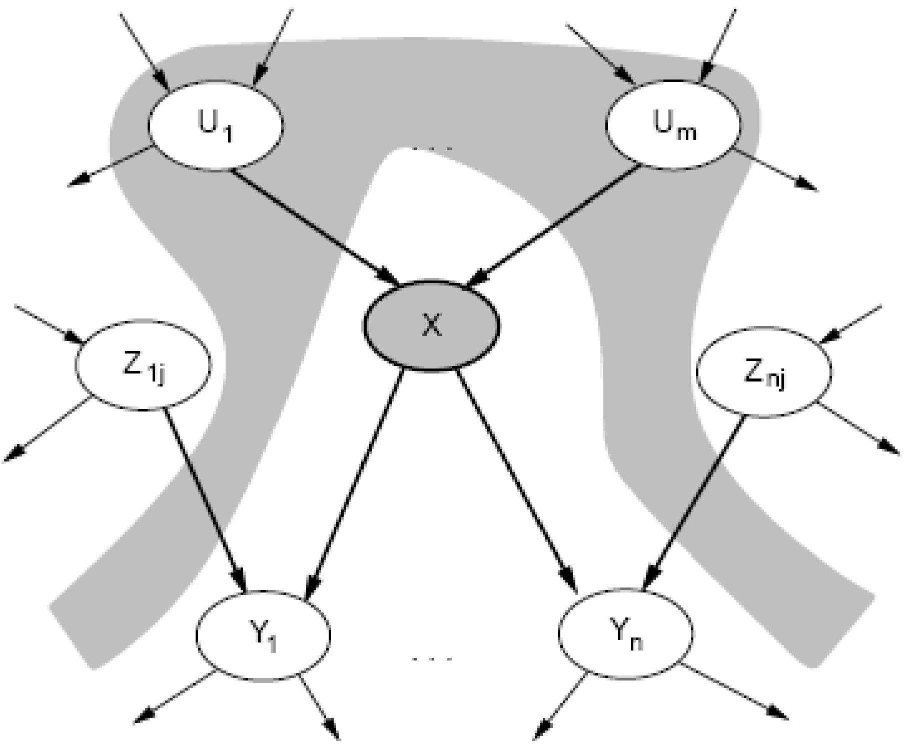
$$\begin{aligned} P(J\ M\ R\ \neg B\ \neg F) &= P(J|\ M\ R\ \neg B\ \neg F)\ P(M\ R\ \neg B\ \neg F) \\ &= P(J|R)\ P(M|R\ \neg B\ \neg F)\ P(R\ \neg B\ \neg F) \\ &= P(J|R)\ P(M|R)\ P(R|\neg B\ \neg F)\ P(\neg B)\ P(\neg F) \\ &= .90 \times .70 \times .001 \times .999 \times .998 = .00062 \end{aligned}$$

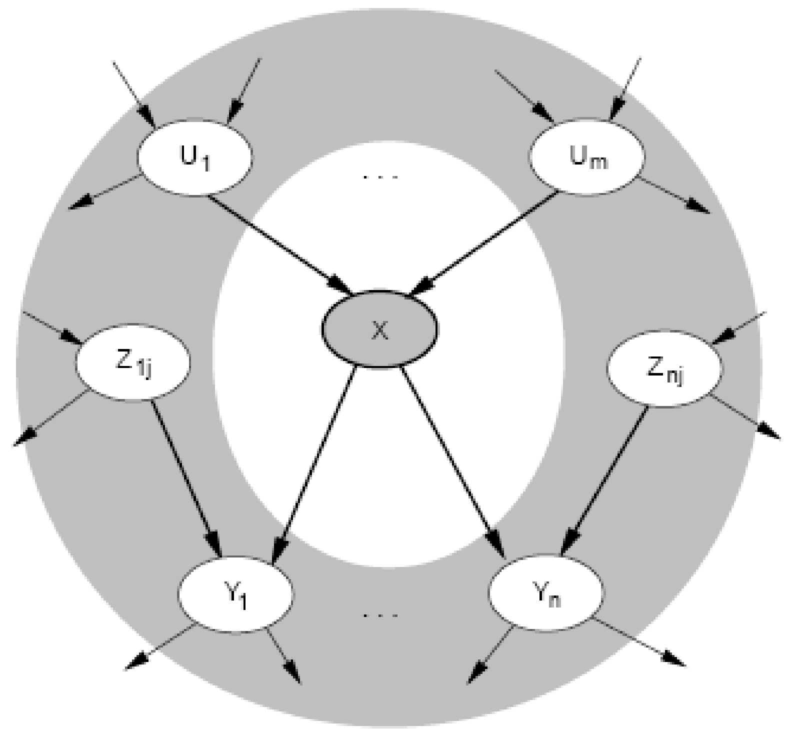
Lokális (topológiai) szemantika:

a. Minden csomópont (X) feltételesen független a nem leszármazottjaitól (Z), ha a **szülői** adottak (U).

b. Minden csomópont (X) feltételesen független minden mástól, ha a **Markov-takarója** adott (szülői+gyerekei+gyerekeinek szülői).

c. **d-elválasztás** …





Az általános eljárás egy háló fokozatos megépítésére:

1. Határozzuk meg a problémát leíró változókat (miről érdemes beszélni).
2. Határozzunk meg **egy sorrendet**.
3. Ameddig maradt még érintetlen változó:

   a) Válasszuk a következő X_i változót és adjunk egy csomópontot a hálóhoz.

   b) Legyen a *Szülők*(X_i) a csomópontok azon **minimális halmaza**, amik már szerepelnek a hálóban és a feltételes függetlenség tulajdonságát teljesítik – húzzuk be a kapcsolatokat.

   c) Definiáljuk X_i csomópont feltételes valószínűségi tábláját.

Mindegyik csomópontot csak korábbi csomóponthoz csatlakoztathatunk
= a háló körmentes lesz.
A valószínűségi háló nem tartalmaz redundáns valószínűségi értékeket, kivéve soronkénti bejegyzéseket a feltételes valószínűségi táblában.

Tömörség és a csomópontok sorrendje

Egy valószínűségi háló gyakran sokkal **tömörebb**, mint az együttes valószínűségi eloszlásfüggvény.

A valószínűségi háló tömörsége a **lokálisan strukturált** (vagy **ritka**) rendszerek egy példája.

Lokálisan strukturált rendszer: egy komponense csak korlátos számú más komponenssel van kapcsolatban közvetlenül, függetlenül a komponensek teljes számától.

Lokális struktúrák: **inkább a lineáris, mint az exponenciális komplexitás** növekedés.

Tömörség és a csomópontok sorrendje

Lokális struktúrák: **inkább a lineáris, mint az exponenciális komplexitás** növekedés.

Valószínűségi háló *n* változóból: legtöbb esetben egy változót csak *k* számú más változó befolyásol, bináris változók = $n \times 2^k$ érték. Együttes valószínűségi eloszlás = $2^n - 1$ érték.

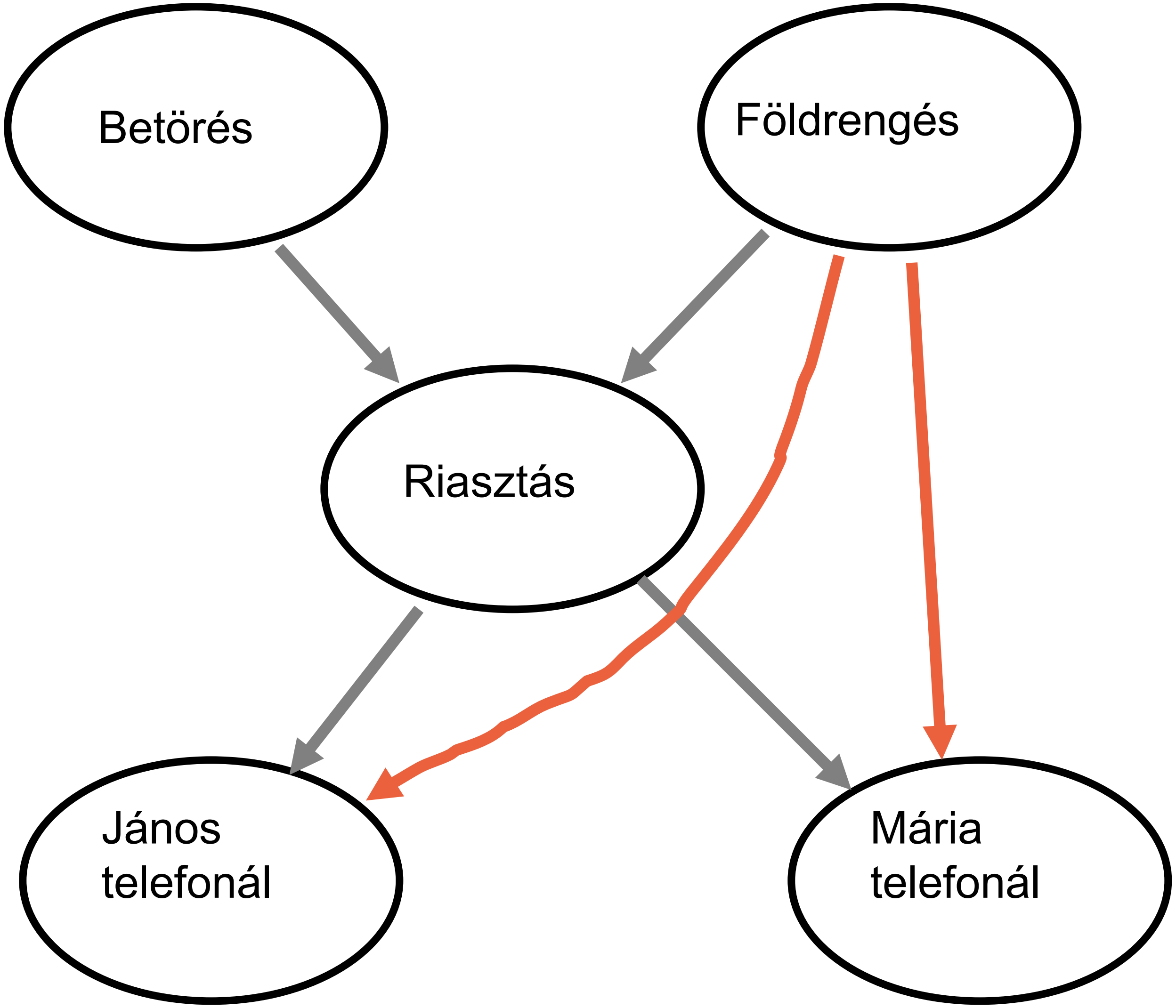
Példa: 20 cs.pont (n = 20), max. 5 szülője legyen (k = 5), valószínűségi háló: 640 érték, együttes valószínűségi eloszlás: > egy millió.

A gyakorlatban előforduló információcsökkenés amiatt lép fel, hogy a valós problémák igen strukturáltak, amit a hálók könnyűszerrel képesek kihasználni.

Tömörség és a csomópontok sorrendje

Bizonyos tárgytartományokban létezhetnek olyan jelentéktelennek tűnő függőségek, amiket feltétlenül modellezni kell egy új kapcsolat felvételével.
De ha ezek a függőségek ténylegesen jelentéktelenek, akkor lehet, hogy nem éri meg a háló komplexitását megnövelni a pontosság kismértékű növelésének érdekében.

Pl.:
hogyha földrengés van, akkor Mária és János akkor sem telefonálna, ha hallanák a riasztót, mivel feltételezik, hogy a földrengés okozta.



2 + 4 + 2 + 2
= 10 (eddig)

2 + 4 + 4 + 4
= 14

Megéri-e?

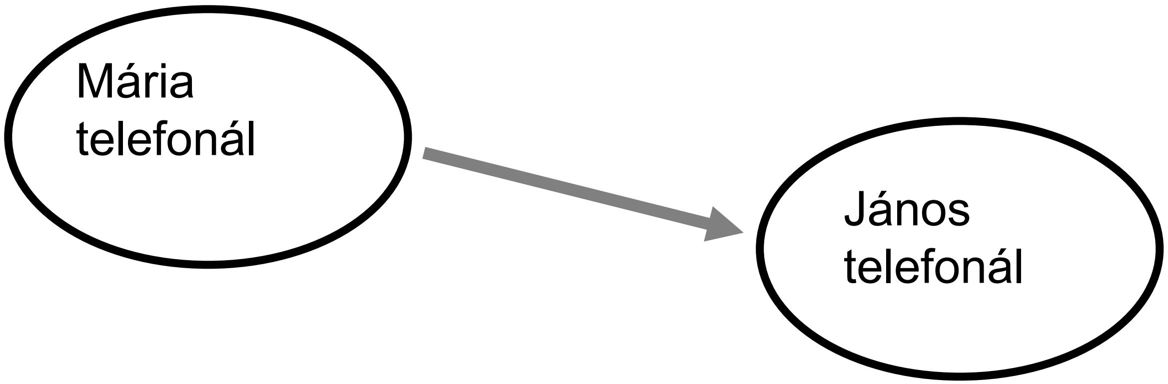
Tömörség és a csomópontok sorrendje

A konstrukciós eljárás működése miatt előbb a „közvetlen befolyásolók”-at kell a hálóhoz adni, ha azt szeretnénk, hogy szülőknek tudjuk őket választani az általuk befolyásolt csomópontnál.

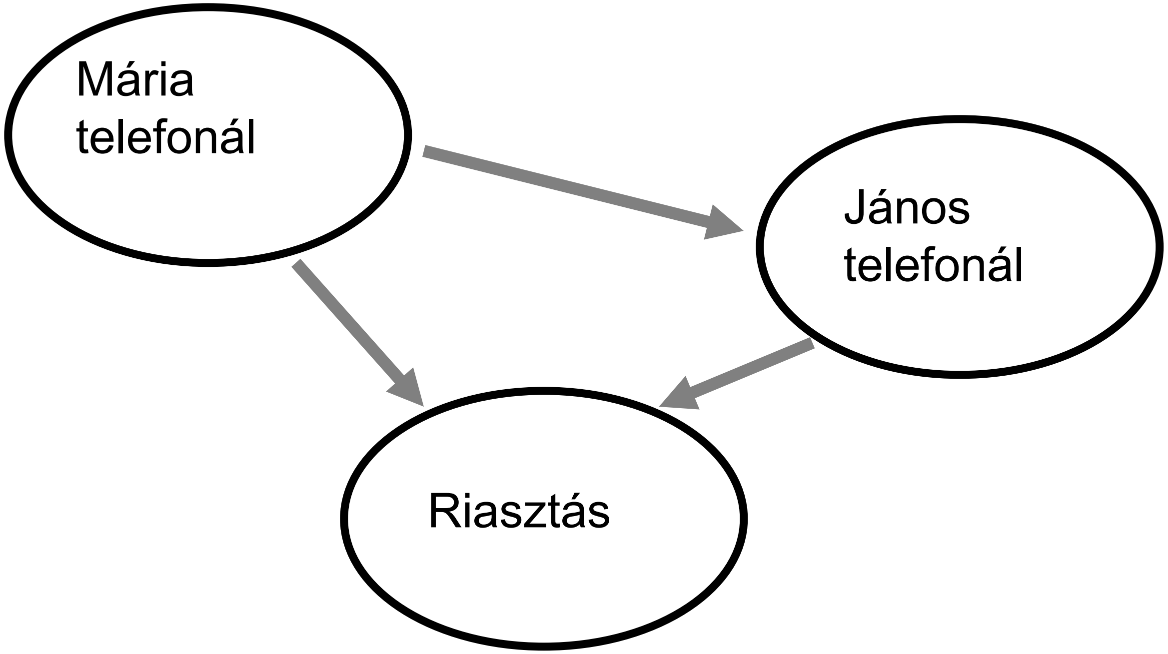
Ezért a helyes sorrend a csomópontok hozzáadásánál:
először az „alapvető okokat” adjuk a hálóhoz, majd a változókat, amiket befolyásolnak, és ezt addig folytatjuk, amíg el nem érjük a „leveleket”, amiknek már nincs közvetlen okozati hatása más változókra (azaz a kauzalitás sorrendjében).

Mi történik, ha történetesen egy rossz sorrendet választunk?
pl.
$$MáriaTelefonál \to JánosTelefonál \to Riasztás \to Betörés \to Földrengés.$$

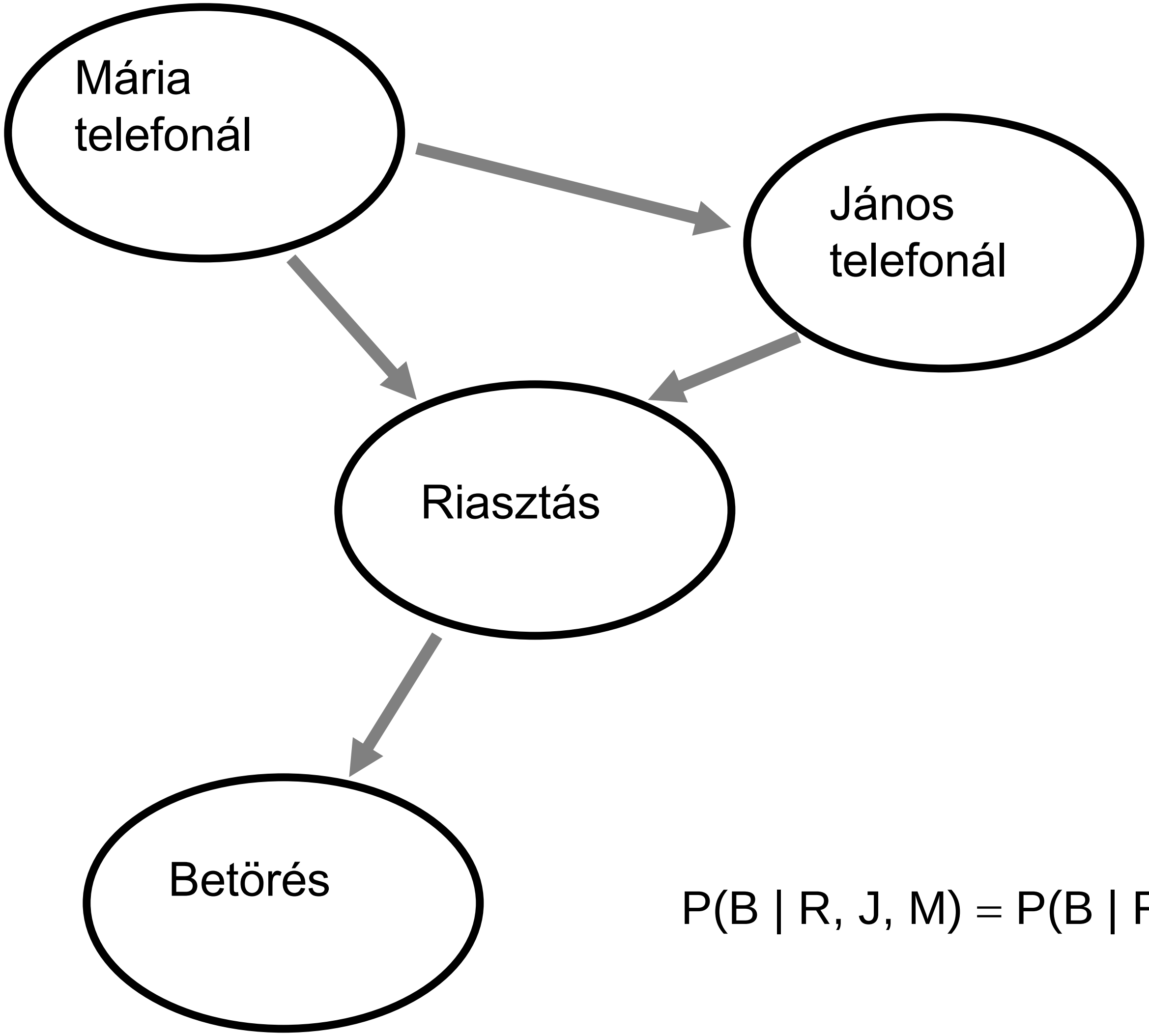

Mária
telefonál



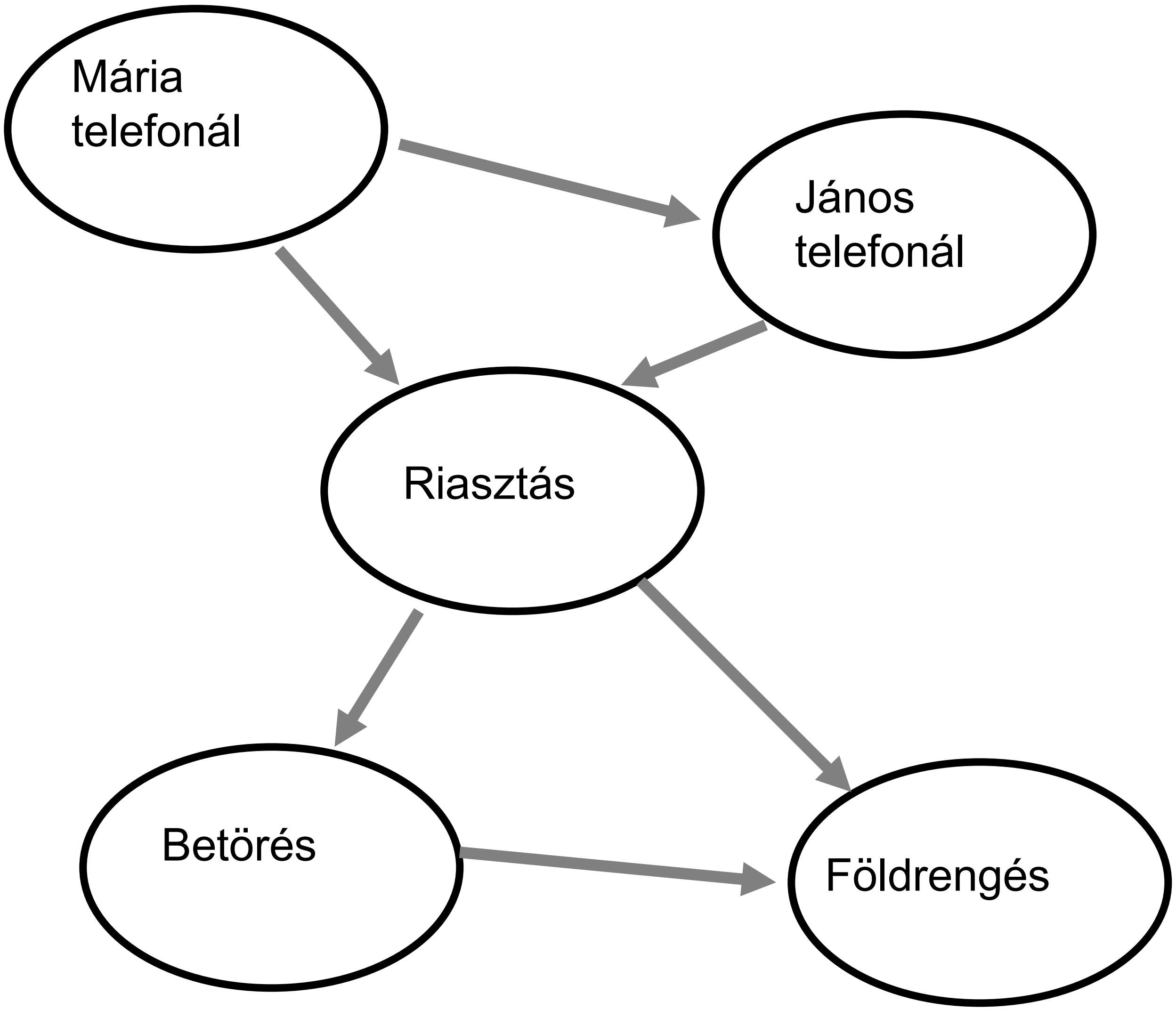
P(J | M) = P(J)? Nem (azaz van nyíl)



P(R | J, M) = P(R | J)? P(R | J, M) = P(R)? Nem



$P(B \mid R, J, M) = P(B \mid R)$?   Igen



$1 + 2 + 4$
$+ 2 + 4$
$= 13$

P(F | B, R, J, M) = P(F | B, R)? = P(F | R)?

Igen Nem

Mi történik, ha történetesen egy nagyon rossz sorrendet választunk?

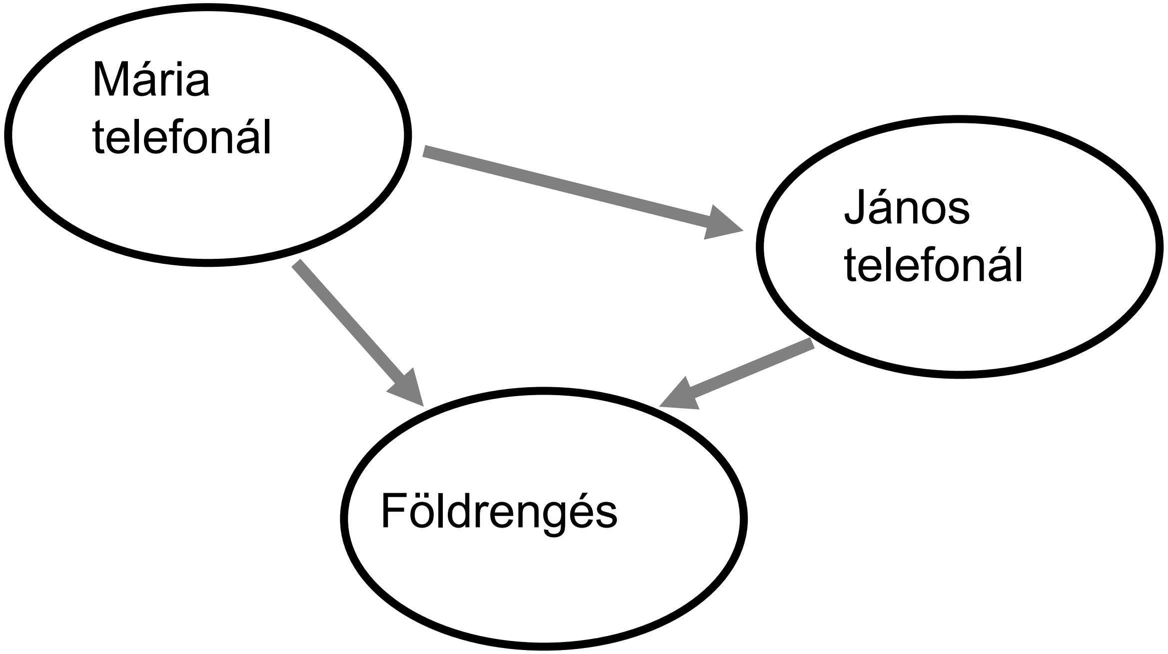
MáriaTelefonál

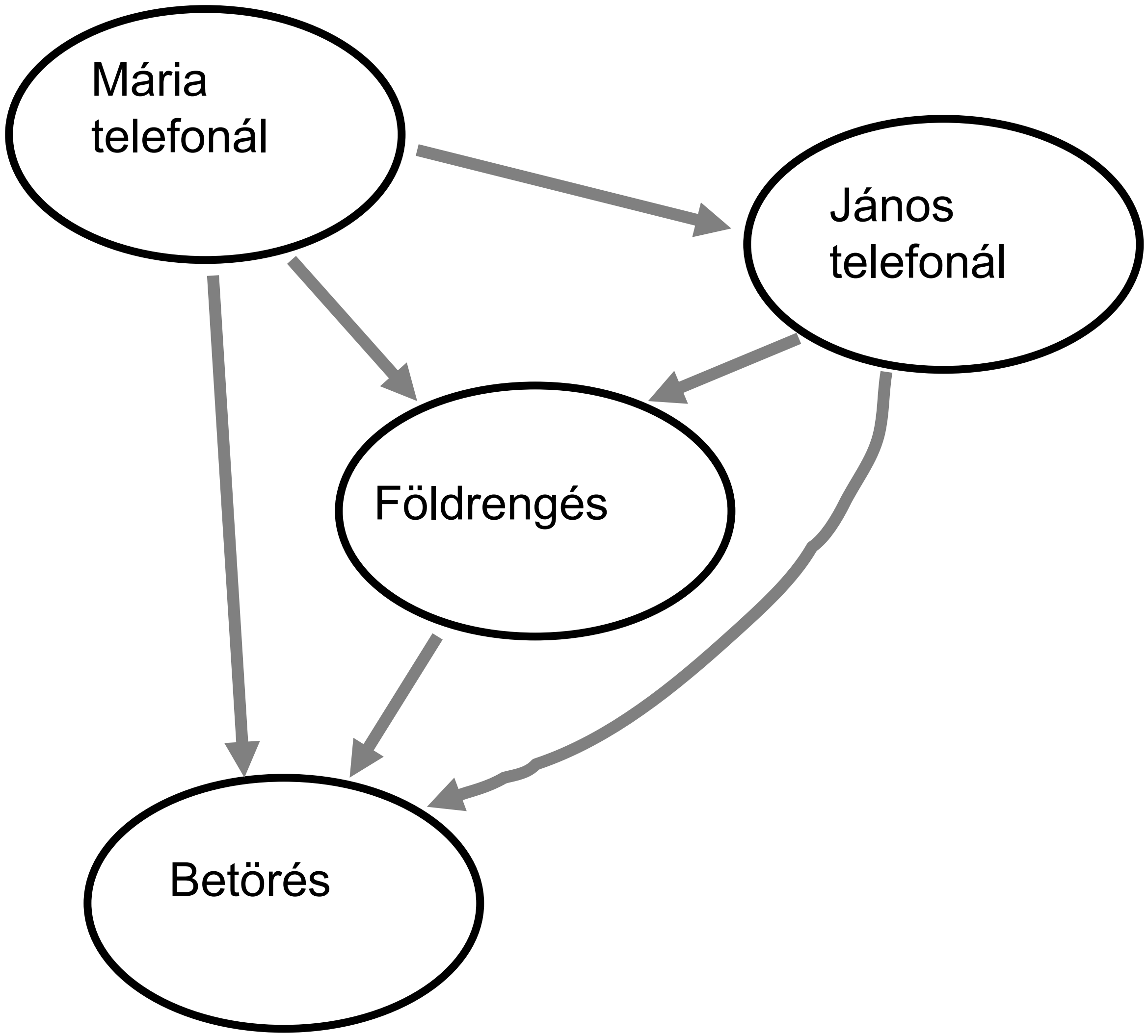
JánosTelefonál

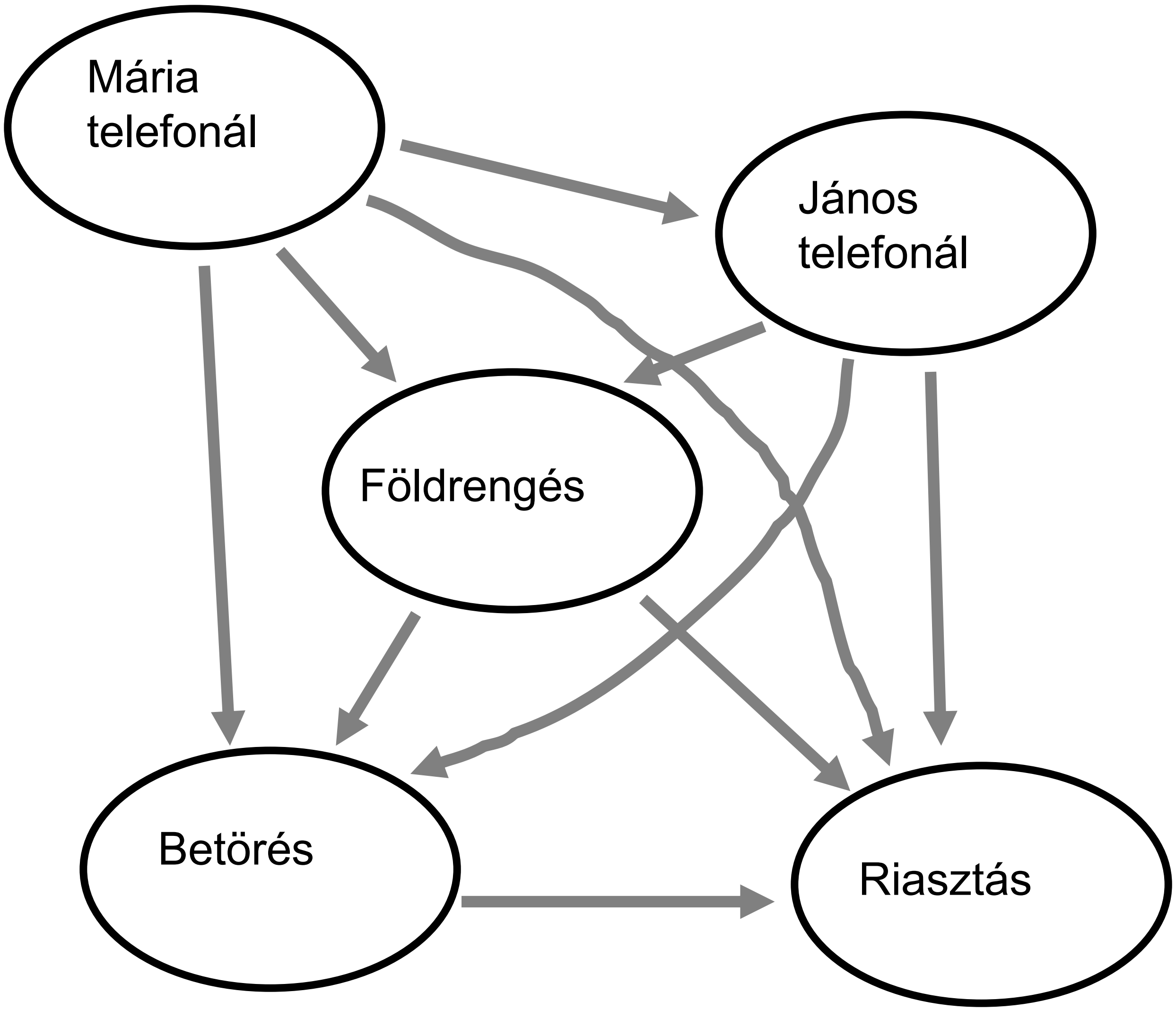
Földrengés.

Riasztás

Betörés


Mária telefonál
János telefonál
Földrengés


Mária telefonál
János telefonál
Földrengés
Betörés



$1 + 2 + 4$
$+ 8 + 16$
$= 31$
$= 2^5 - 1$

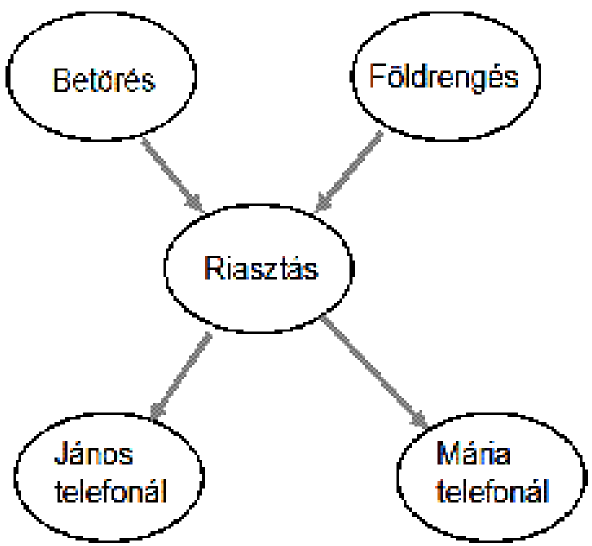
Sok (elméletileg exponenciális számú) **valószínűség**

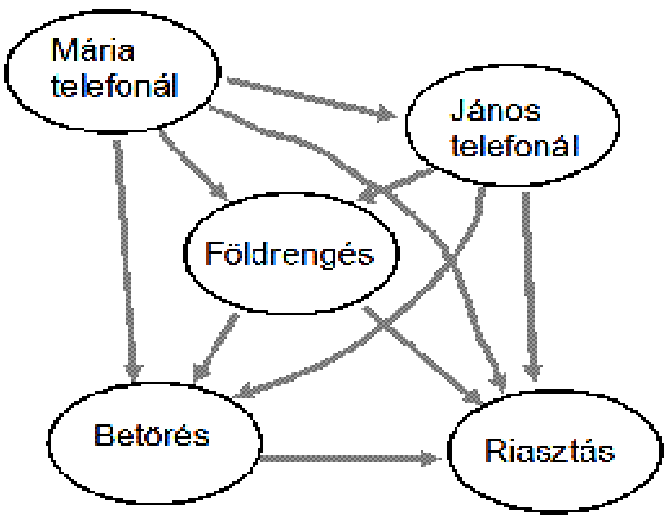
De a legrosszabb következmény:
néhány kapcsolat furcsa viszonyt reprezentál, ami nehéz és **nem természetes valószínűségi ítéleteket igényel**, pl.

P(*Földrengés* | *Mária* …, *János* …) ?

P(*Betörés* | *Földrengés*) ? ... ...

Okozati (kauzális) modell: kevesebb, megbízhatóbb érték, az értékeket gyakran könnyebb elérni





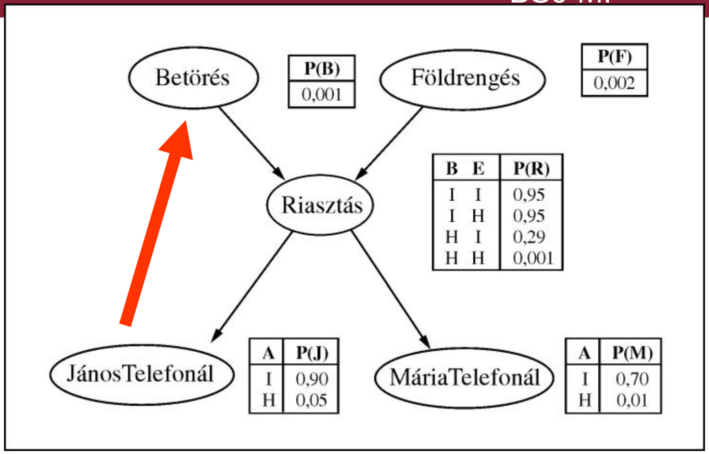
Következtetés valószínűségi hálókban

Az alapvető feladat:
kiszámítani az a posteriori valószínűséget a **lekérdezéses változók**ra, ha a **tény** ill. **bizonyíték (evidencia) változók**nak az értékei adottak:

P(Lekérdezéses | Bizonyíték)

A riasztós példában, pl.:
P(*Betörés* | *MáriaTelefonál* és *JánosTelefonál*)

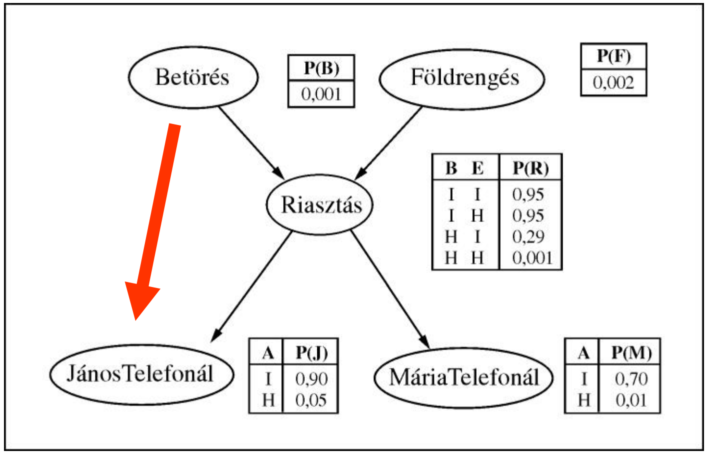
Általában *az ágens az érzékeléséből (vagy egyéb következtetésből) kap értékeket a tény változókhoz, és más változók lehetséges értékeiről kérdez, hogy el tudja dönteni milyen cselekvéseket végezzen*.



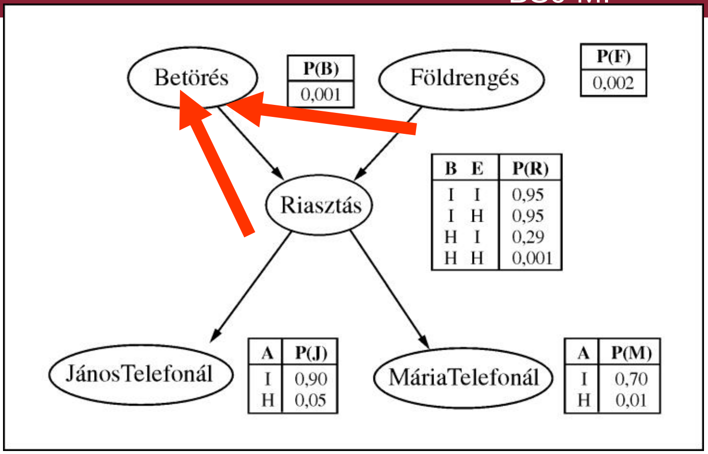
Diagnosztikai következtetés
(hatásról az okra)

Ha adott a *JánosTelefonál,* akkor kiszámíthatjuk pl., hogy
P(*Betörés* | *JánosTelefonál*) = 0,016.

Okozati következtetés
(okról a hatásra)



Ha adott a *Betörés,* akkor kiszámíthatjuk pl., hogy
P(*JánosTelefonál* | *Betörés*) = 0,67.



Okok közötti következtetés

(következtetés egy közös hatás okai között)

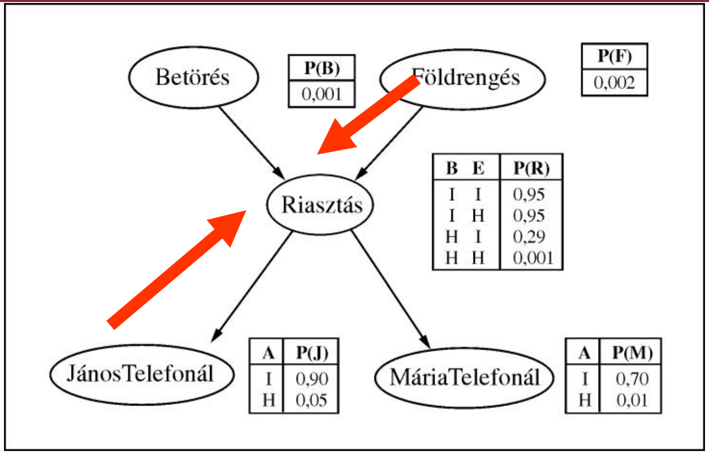
Ha adott a *Riasztás,* akkor *P*(*Betörés* | *Riasztás*) = 0.376.

Ha azonban hozzávesszük azt a tényt is, hogy a *Földrengés* igaz, akkor P(*Betörés* | *Riasztás, Földrengés*) = 0.003.

Bár a betörések és a földrengések függetlenek, az egyik jelenléte a másik valószínűségét csökkenti. Ez a fajta következtetési mód a **kimagyarázás.**

Kevert következtetések
(a fentiek kombinált használata).

Ha a *JánosTelefonál* okozat igaz
és a *Földrengés* ok hamis, akkor



$P(Riasztás \mid JánosTelefonál, \neg Földrengés) = 0{,}03$.

Ez a diagnosztikai és okozati következtetés együttes felhasználása.

Hasonlóan,

$P(Betörés \mid JánosTelefonál, \neg Földrengés) = 0{,}017$.

Ez a diagnosztikai és az okok közötti következtetés kombinálása.

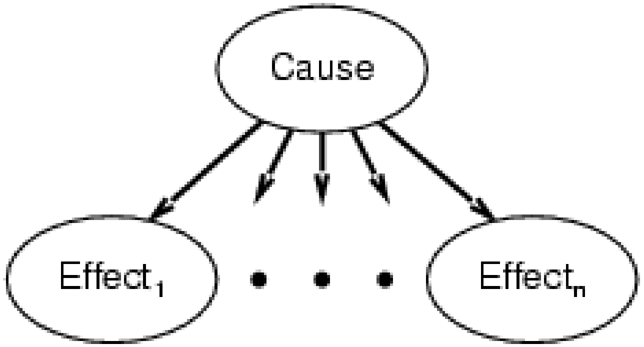
Érzékenységi vizsgálat elvégzése, annak érdekében,
hogy megértsük, hogy a modell mely vonatkozásának van
a legnagyobb hatása a lekérdezéses változókra nézve.

Naiv Bayes-hálók

Feltevések:

1.) Kétféle csomópont lehetséges: a „ok” és „következmény”.

2.) A következmények egymástól feltételesen függetlenek egymástól feltéve az okot.

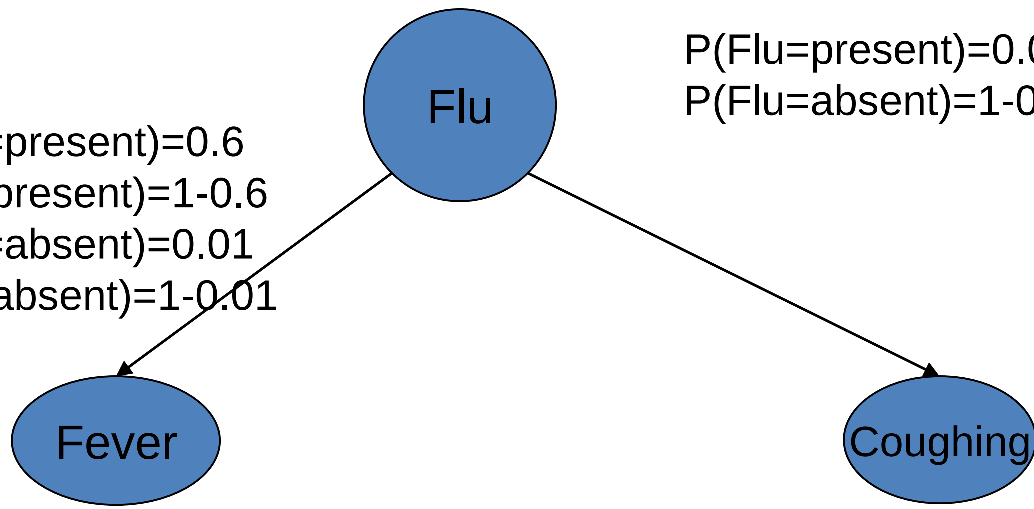


Naiv Bayes-hálók

Változók (csomópontok)

Flu (Influenza): {jelen, nincs jelen}
Fever (Láz): {jelen, nincs jelen}
Coughing (Köhögés): {jelen, nincs jelen}

Modell



P(Flu=present)=0.001
P(Flu=absent)=1-0.001

P(Fever=present|Flu=present)=0.6
P(Fever=absent|Flu=present)=1-0.6
P(Fever=present|Flu=absent)=0.01
P(Fever=absent|Flu=absent)=1-0.01

P(Coughing=present|Flu=present)=0.3
P(Coughing=absent|Flu=present)=1-0.3
P(Coughing=present|Flu=absent)=0.02
P(Coughing=absent|Flu=absent)=1-0.02

Naiv Bayes-hálók

Együttes valószínűség-eloszlás dekompozíciója:

$P(Y,X_1,..,X_n)$ $= P(Y)\prod_i P(X_i,|Y, X_1,..,X_{i-1})$ //lánc szabály miatt

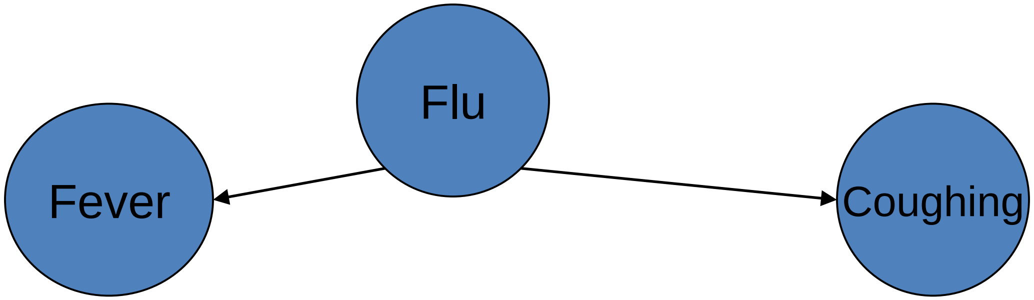
$= P(Y)\prod_i P(X_i,|Y)$ // naiv BN feltevés

2n+1 paraméter

Diagnosztikus következtetés:

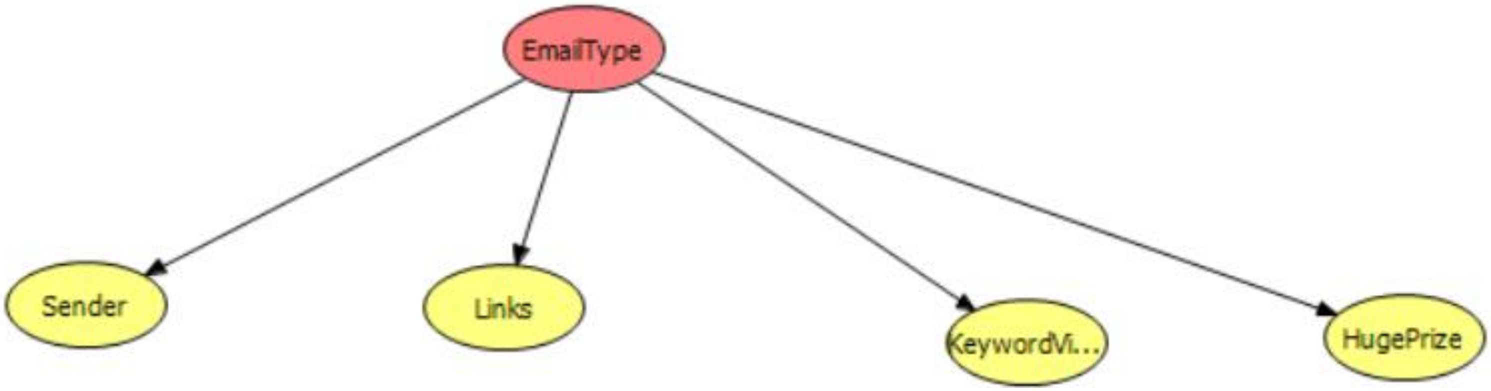
$P(Y|x_{i1},..,x_{ik})$ $= P(Y)\prod_j P(x_{ij},|Y) / P(x_{i1},..,x_{ik})$

$$p(Flu = present \mid Fever = absent, Coughing = present)$$
$$\propto p(Flu = present)\, p(Fever = absent \mid Flu = present)\, p(Coughing = present \mid Flu = present)$$



Gyakorlati példa: SPAM filter

- SPAM filter
  - SPAM: yes/no [suspicious..]
  - Attributes
    - Sender, subject, link, attachment,..



A valószínűségi hálók szemantikája

(1) a háló = **együttes valószínűségi eloszlás** egy leírása,

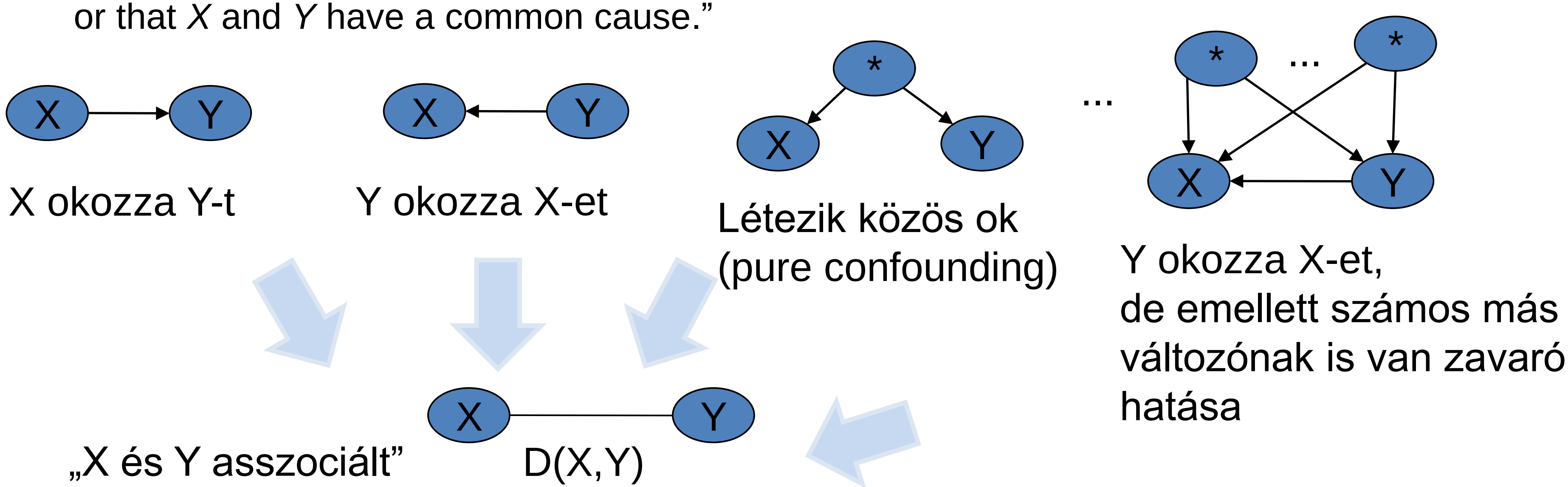
(2) a háló = **feltételes függetlenségekről szóló állítások** együttese.

(3) a háló = **oksági kapcsolatok** együttese.

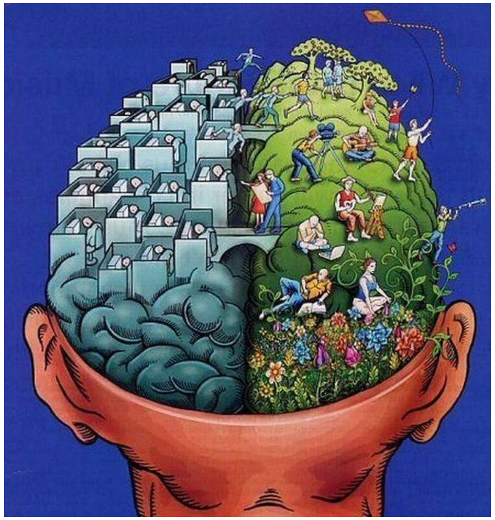
FONTOS: oksági kapcsolat ≠ asszociációs kapcsolat

Reichenbach's Common Cause Principle:

„a correlation between events *X* and *Y* indicates either that *X* causes *Y*, or that *Y* causes *X*, or that *X* and *Y* have a common cause."



Budapesti Műszaki és Gazdaságtudományi Egyetem
Méréstechnika és Információs rendszerek Tanszék



Mesterséges Intelligencia - MI

Következtetés valószínűségi hálókban

Előadó:
Dr. Hullám Gábor

Előadás anyaga:
Dr. Dobrowiecki Tadeusz

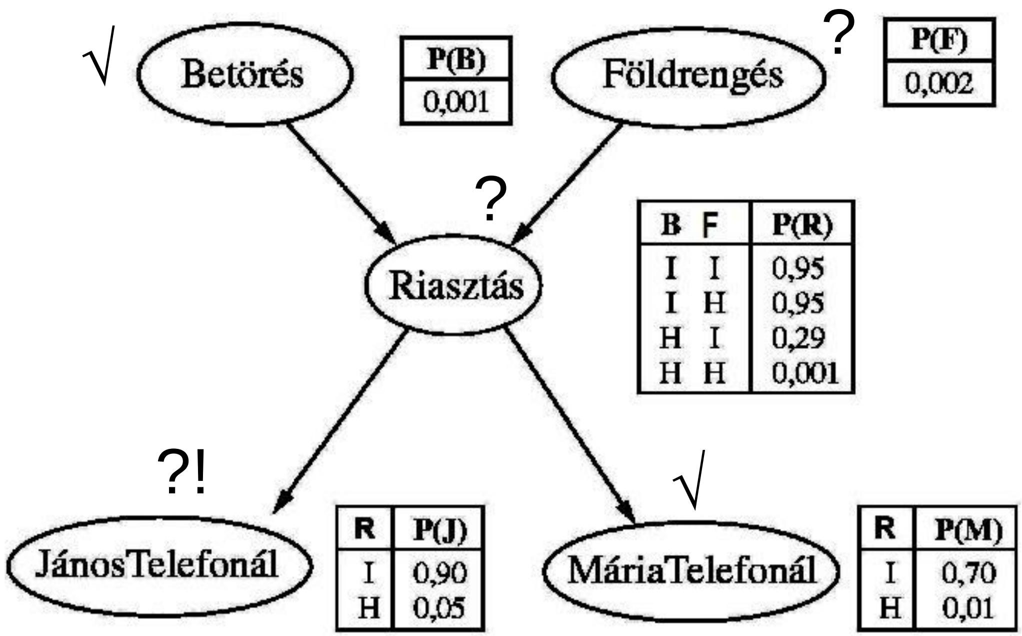
MŰEGYETEM 1782

Következtetés valószínűségi hálókban

P(lekérdezés | bizonyíték) ?

- **egzakt** – „egyszerű” hálónál egyszerű,
  „bonyolult” hálónál bonyolult

- **közelítő** – „bonyolult” háló esetén is egyszerű



P(B)
0,001

P(F)
0,002

B	F	P(R)
I	I	0,95
I	H	0,95
H	I	0,29
H	H	0,001

R	P(J)
I	0,90
H	0,05

R	P(M)
I	0,70
H	0,01

Összegzés rejtett változókra

Pl. $P(J|M\neg B) = ?$

$= P(JM\neg B)/P(M\neg B)$

$P(JM\neg B) =$

$\Sigma_{rf}\, P(JM\neg B\ r\ f)$

$P(M\neg B) =$

$\Sigma_{rfj}\, P(M\neg B\ r\ f\ j)$

$$\left.\begin{array}{l}
P(J\ M\neg B\ \ R\ \ F) = P(J|R)\ P(M|R)\ P(R|\neg BF)\ P(\neg B)\ P(F) = \ldots. \\
P(J\ M\neg B\ \neg R\ \ F) = P(J|\neg R)\ P(M|\neg R)\ P(\neg R|\neg BF)\ P(\neg B)\ P(F) = \ldots. \\
P(J\ M\neg B\ \ R\ \neg F) = P(J|R)\ P(M|R)\ P(R|\neg B\neg F)\ P(\neg B)\ P(\neg F) = \ldots. \\
P(J\ M\neg B\ \neg R\ \neg F) = P(J|\neg R)\ P(M|\neg R)\ P(\neg R|\neg B\neg F)\ P(\neg B)\ P(\neg F) = \ldots.
\end{array}\right\}$$

$$\left.\begin{array}{l}
P(M\neg B\ \ J\ \ R\ \ F) = P(J|R)\ P(M|R)\ P(R|\neg BF)\ P(\neg B)\ P(F) = \ldots. \\
P(M\neg B\ \ J\ \ R\neg F) = P(J|R)\ P(M|R)\ P(R|\neg B\neg F)\ P(\neg B)\ P(\neg F) = \ldots. \\
P(M\neg B\ \ J\neg R\ \ F) = P(J|\neg R)\ P(M|\neg R)\ P(\neg R|\neg BF)\ P(\neg B)\ P(F) = \ldots.. \\
P(M\neg B\ \ J\neg R\neg F) = P(J|\neg R)\ P(M|\neg R)\ P(\neg R|\neg B\neg F)\ P(\neg B)\ P(\neg F) = \ldots. \\
P(M\neg B\ \neg J\ \ R\ \ F) = P(\neg J|R)\ P(M|R)\ P(R|\neg BF)\ P(\neg B)\ P(F) = \ldots.. \\
P(M\neg B\ \neg J\ \ R\neg F) = P(\neg J|R)\ P(M|R)\ P(R|\neg B\neg F)\ P(\neg B)\ P(\neg F) = \ldots.. \\
P(M\neg B\ \neg J\neg R\ \ F) = P(\neg J|\neg R)\ P(M|\neg R)\ P(\neg R|\neg BF)\ P(\neg B)\ P(F) = \ldots.. \\
P(M\neg B\ \neg J\neg R\neg F) = P(\neg J|\neg R)\ P(M|\neg R)\ P(\neg R|\neg B\neg F)\ P(\neg B)\ P(\neg F) = \ldots.
\end{array}\right\}$$

Következtetés felsorolással (balról jobbra)

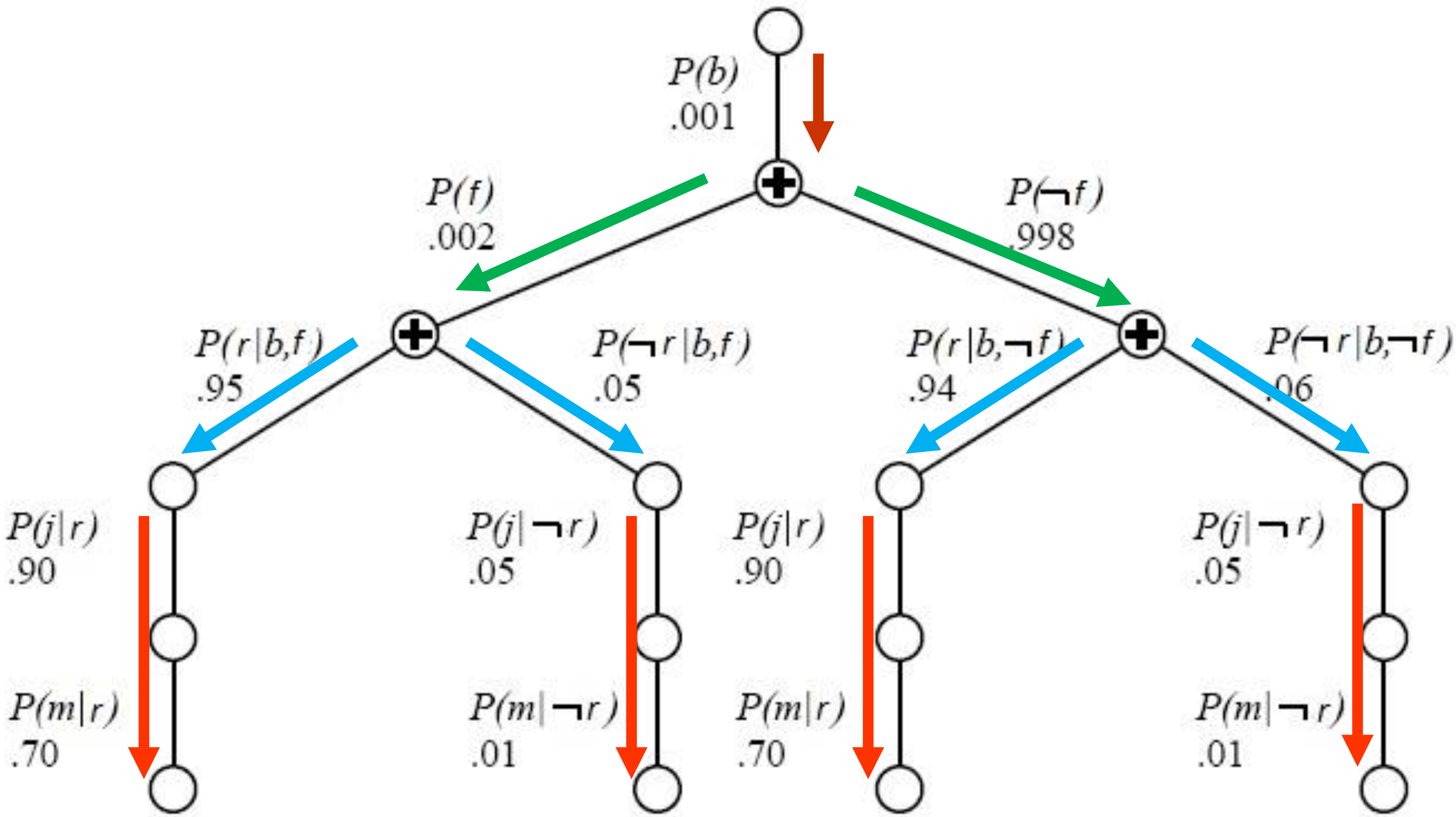
$$P(B \mid J\ M) = P(b{=}\text{Igaz} \mid j{=}\text{Igaz}\ m{=}\text{Igaz}) = ?$$

$$P(B \mid J\ M) = \alpha \sum_f \sum_r P(B\ f\ r\ J\ M)$$
$$= \alpha \sum_f \sum_r P(B)\ P(f)\ P(r \mid B\ f)\ P(J \mid r)\ P(M \mid r)$$
$$= \alpha\ P(B) \sum_f P(f) \sum_r P(r \mid B\ f)\ P(J \mid r)\ P(M \mid r)$$

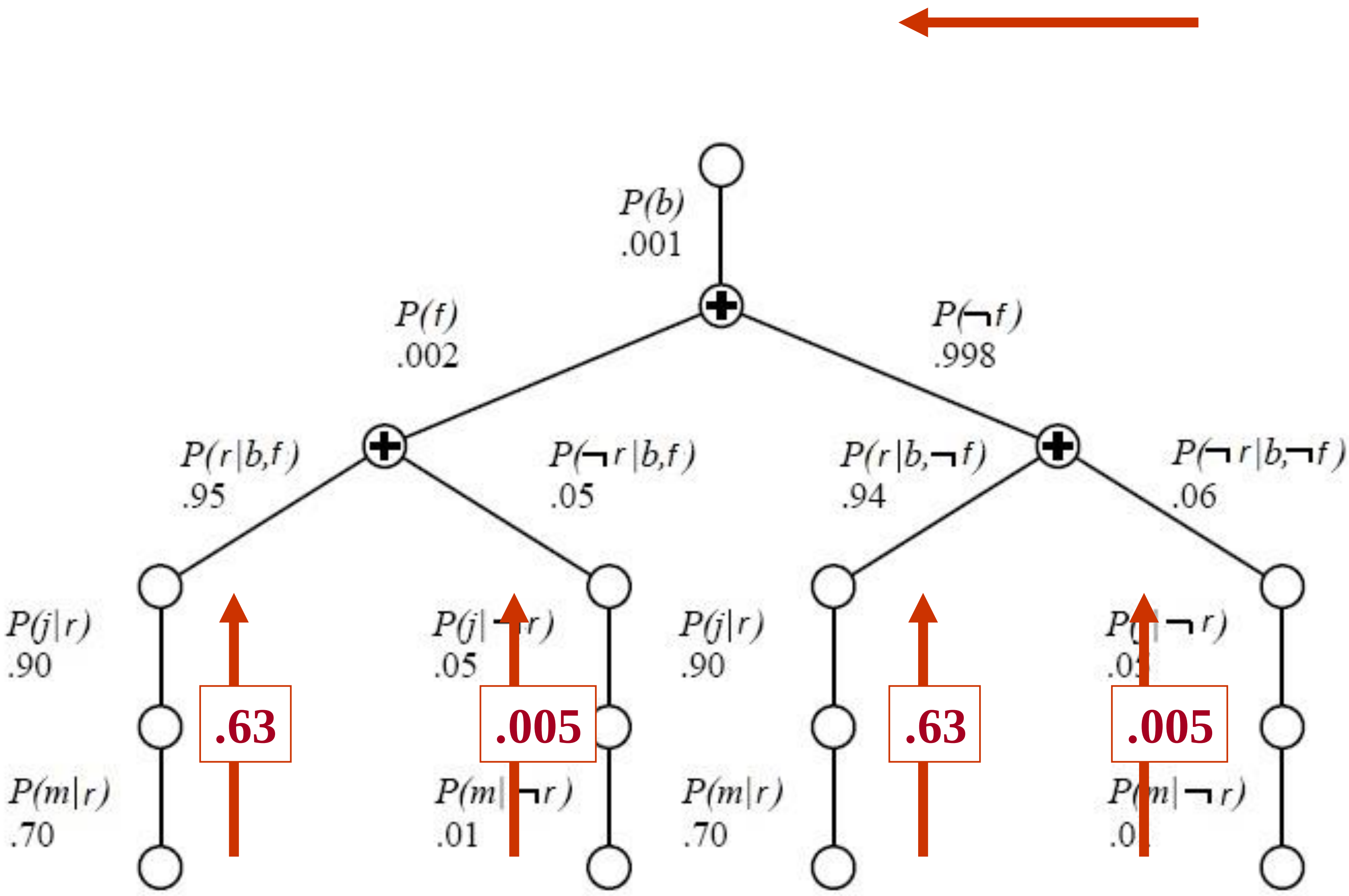
és pl. $P(F \mid J\ M) = \alpha \sum_b \sum_r P(F\ b\ r\ J\ M) = ...$

$$P(B \mid J\ M) \geq P(F \mid J\ M)\ ?$$

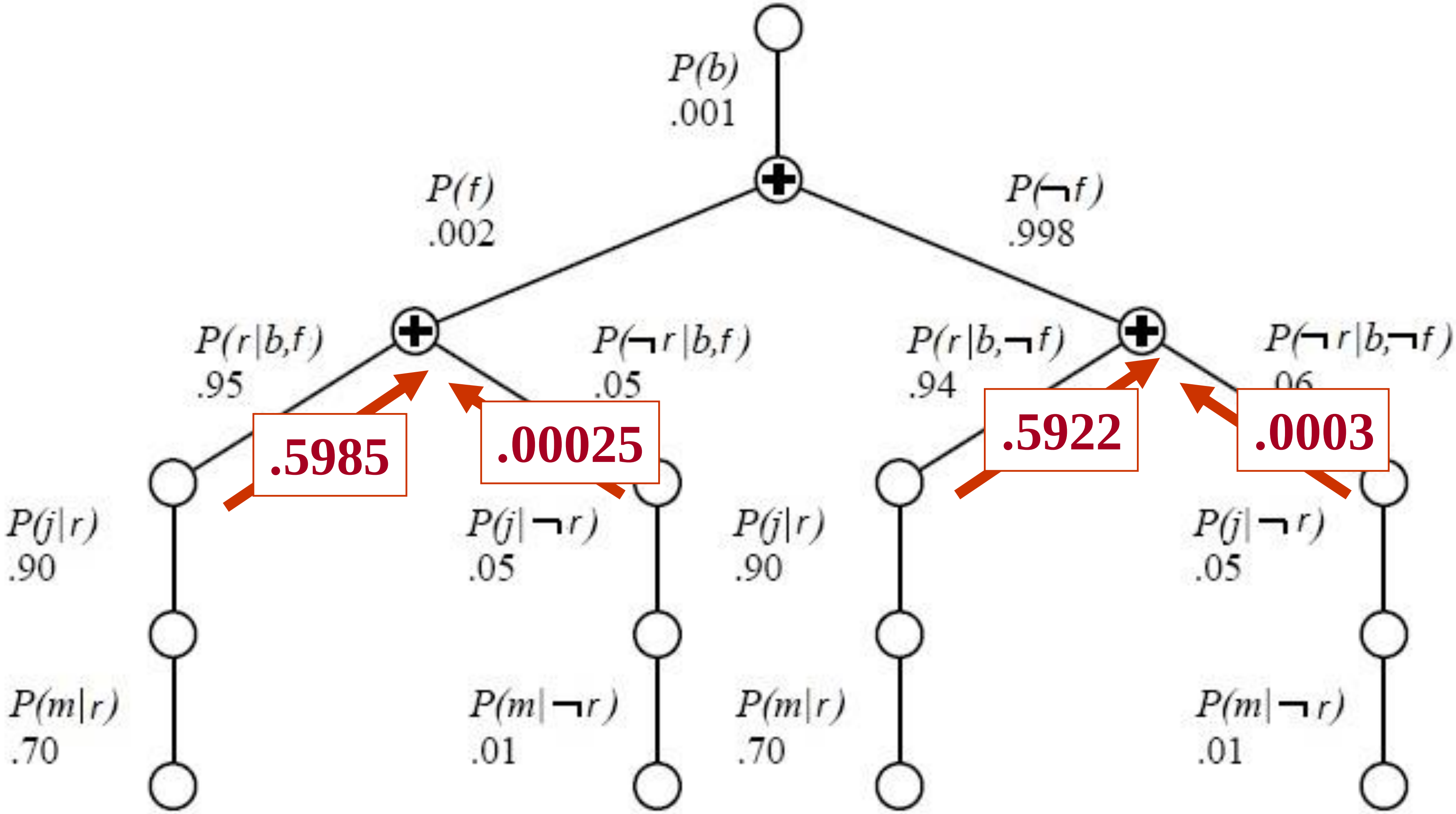
$$P(B \mid J\ M) = \alpha\ P(B)\ \Sigma_f\ P(f)\ \Sigma_r\ P(r \mid B\ f)\ P(J \mid r)\ P(M \mid r)$$



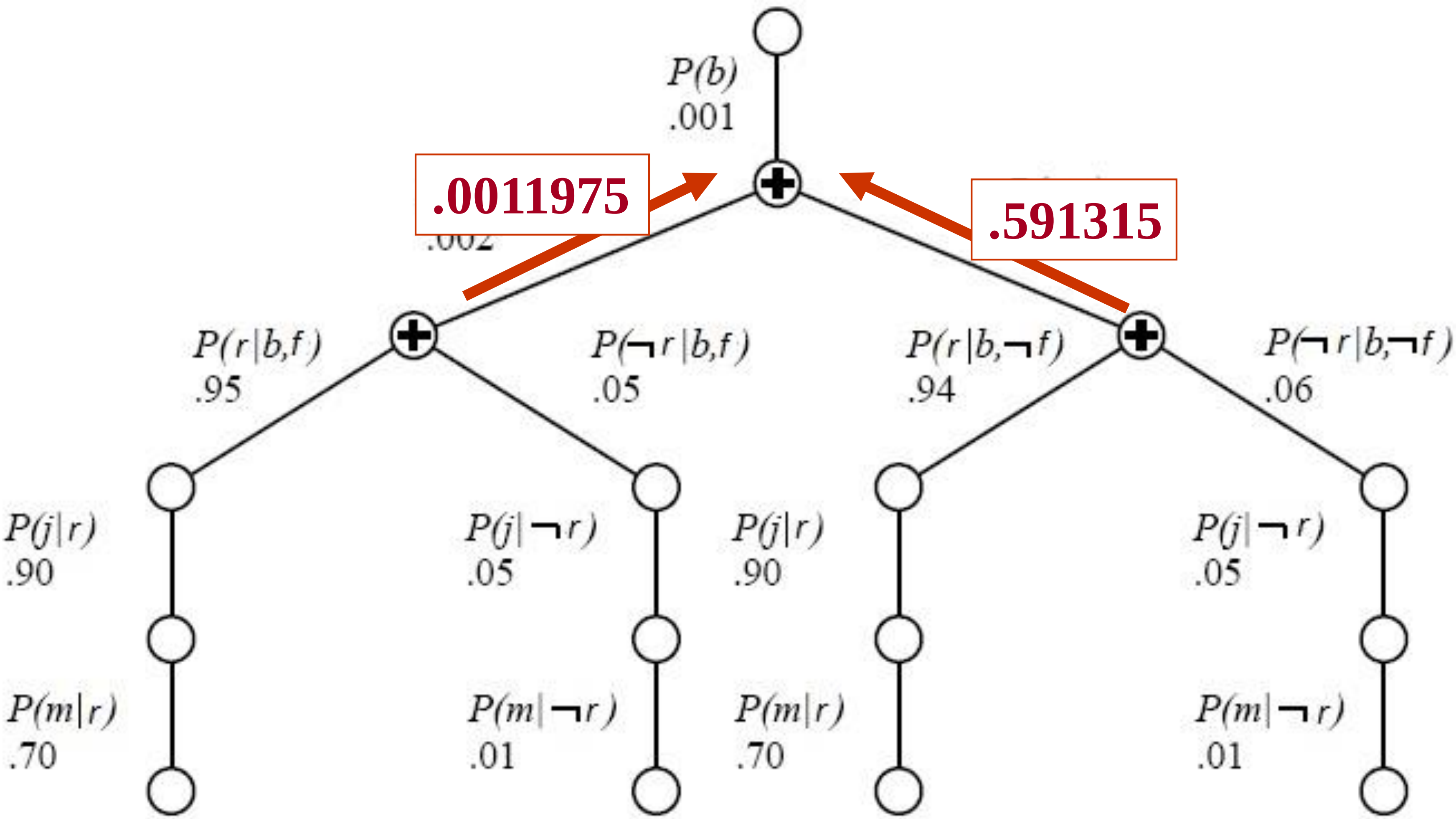
$$P(B \mid J\ M) = \alpha\ P(B)\ \Sigma_f\ P(f)\ \Sigma_r\ P(r \mid B\ f)\ P(J \mid r)\ P(M \mid r)$$



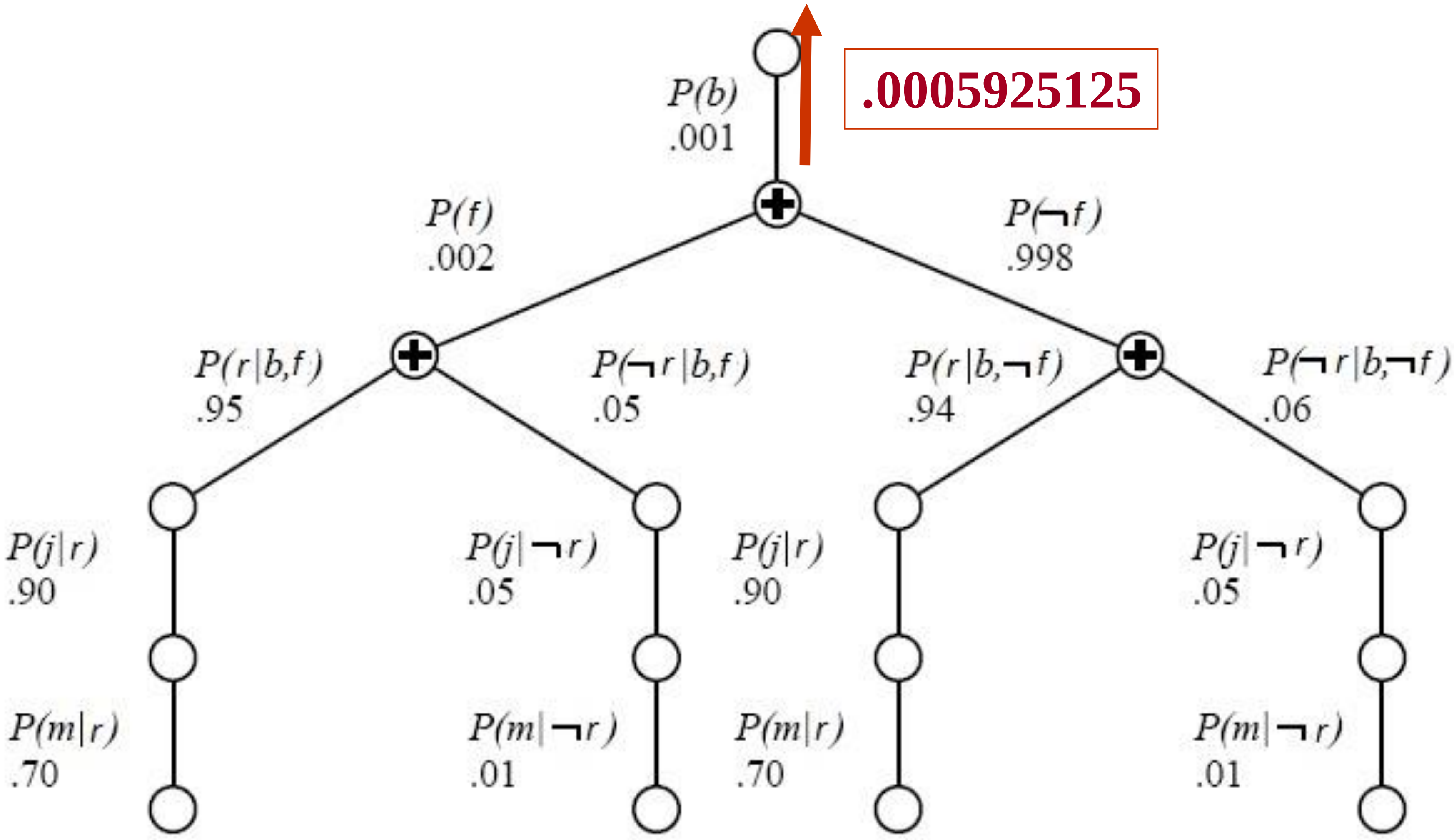
$P(B \mid J\ M) = \alpha\ P(B)\ \Sigma_f\ P(f)\ \Sigma_r\ P(r \mid B\ f)\ P(J \mid r)\ P(M \mid r)$



$$P(B \mid J\ M) = \alpha\, P(B)\, \Sigma_f\, P(f)\, \Sigma_r\, P(r \mid B\ f)\, P(J \mid r)\, P(M \mid r)$$



$$P(B \mid J\ M) = \alpha\ P(B)\ \Sigma_f\ P(f)\ \Sigma_r\ P(r \mid B\ f)\ P(J \mid r)\ P(M \mid r)$$

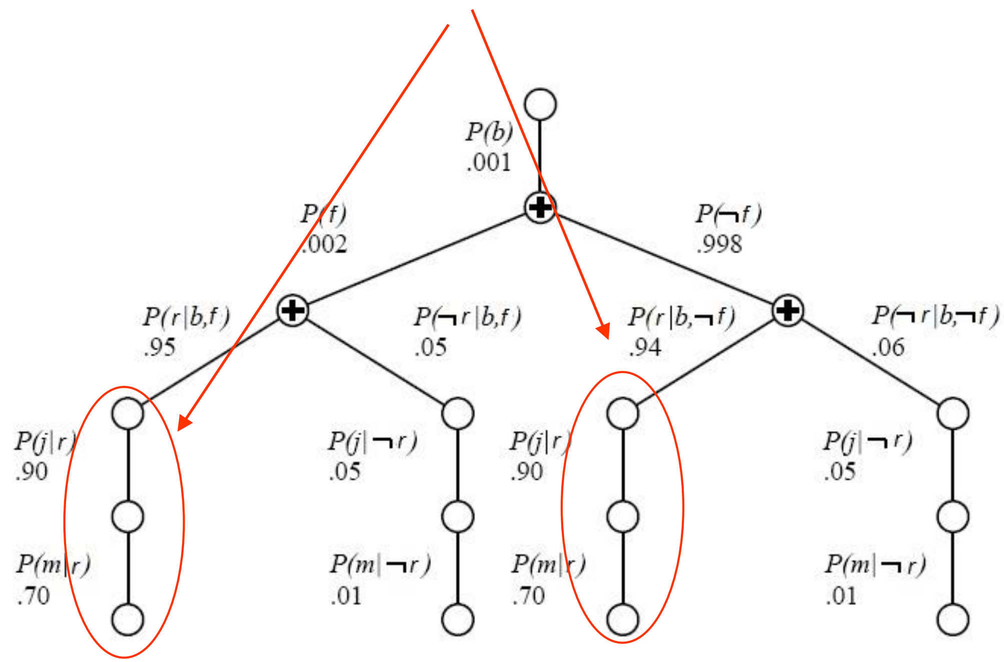


Következtetés változók eliminálásával P(B | J M)

$$P(B \mid J\ M) = \alpha\, P(B)\, \Sigma_f\, P(f)\, \Sigma_r\, P(r \mid B\ f)\, P(J \mid r)\, P(M \mid r)$$

Kiértékelés:
- jobbról balra
- köztes eredmények tárolása

Probléma előbb: bizonyos tagok többszörös kiszámítása, pl.



Irreleváns változók eliminálása

$$P(J \mid B) = \alpha\, P(B)\, \Sigma_f\, P(f)\, \Sigma_r\, P(r \mid B\, f)\, P(J \mid r)\, \Sigma_m\, P(m \mid r)$$

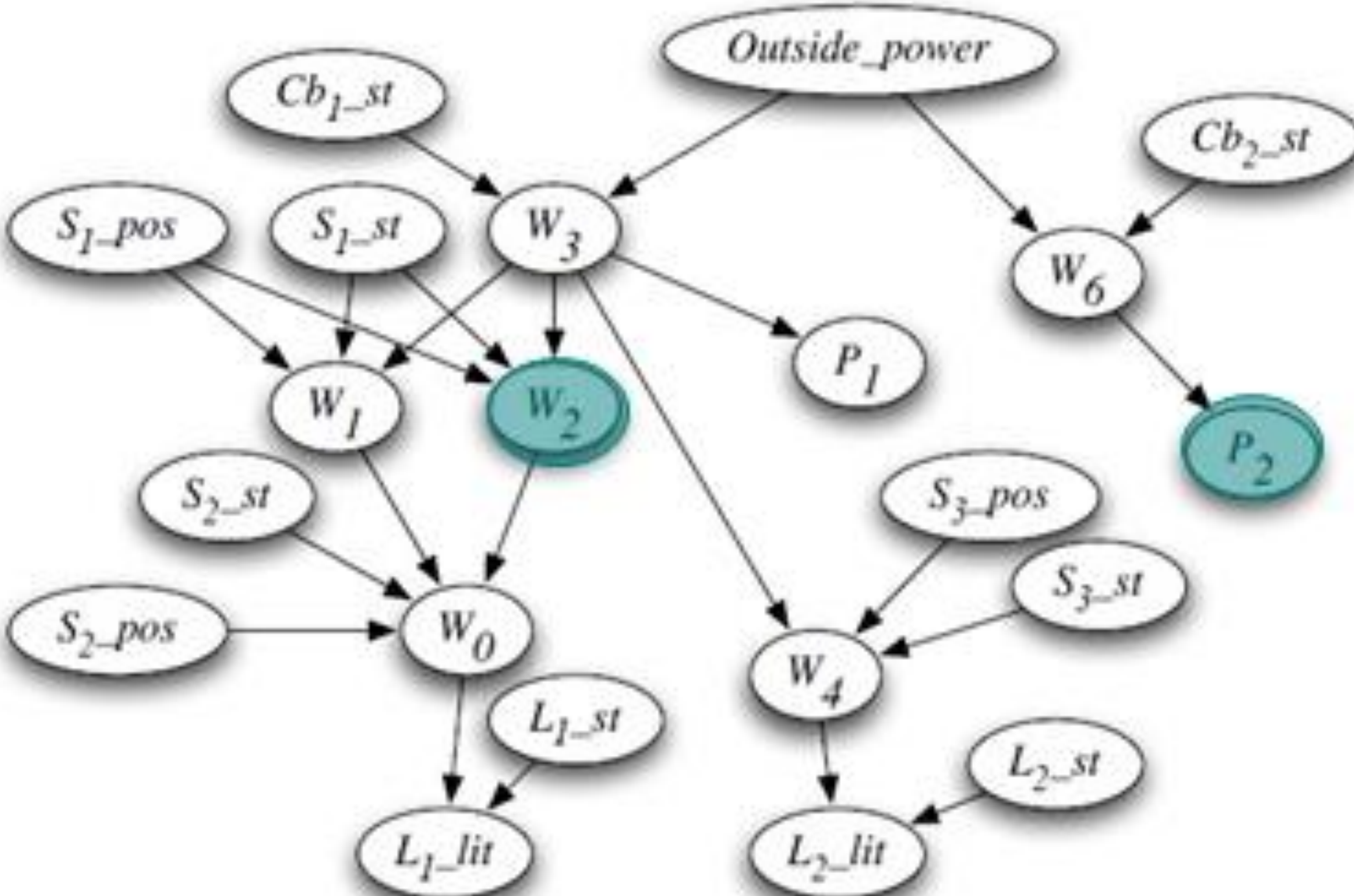
= 1

P(*JánosTelefonál*|*Betörés*) lekérdezés eredményét nem változtatja meg a *MáriaHív* eltávolítása a hálóból.

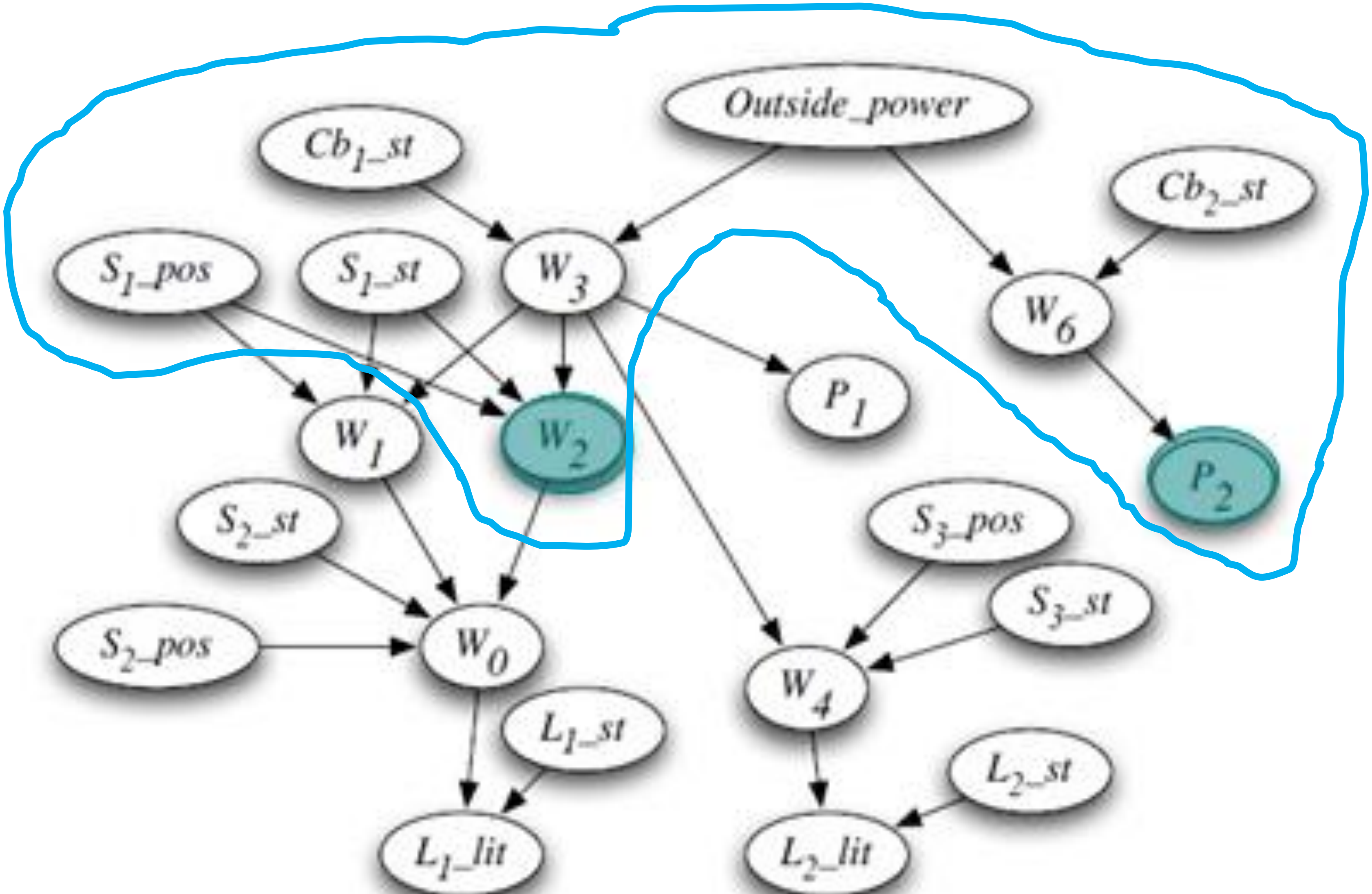
Általában, bármely **levél** csomópontot eltávolíthatunk, ami **nem célváltozó** vagy **nem bizonyíték** változó.
Eltávolítás után lehetnek újabb levélcsomópontok, amelyek szintén irrelevánsak lehetnek.

Minden változó, ami nem őse a célváltozónak, vagy egy bizonyíték változónak irreleváns a lekérdezésre. A változó elimináló algoritmus ezért az összes ilyen változót eltávolíthatja a lekérdezés kiértékelése előtt.

Mi a helyzet a *P(Földrengés* |*Betörés*) esetben?



Outside_power
Cb_1_st
Cb_2_st
S_1_pos
S_1_st
W_3
W_6
P_1
W_1
W_2
P_2
S_2_st
S_3_pos
S_3_st
S_2_pos
W_0
W_4
L_1_st
L_2_st
L_1_lit
L_2_lit



Outside_power
Cb_1_st
Cb_2_st
S_1_pos
S_1_st
W_3
W_6
W_1
W_2
P_1
P_2
S_2_st
S_3_pos
S_3_st
S_2_pos
W_0
W_4
L_1_st
L_2_st
L_1_lit
L_2_lit

Következtetés változók eliminálásával

$P(B \mid J\ M)$

- kiértékelés jobbról-balra
- matrixos faktorok FVT helyett
- speciális algebrai műveletek
- lényegesen egyszerűbb implementáció

$$P(B \mid J\ M) = \alpha\, P(B)\, \Sigma_f\, P(f)\, \Sigma_r\, P(r \mid B\ f)\, P(J \mid r)\, P(M \mid r)$$

$$\begin{aligned}
P(B \mid J\ M) &= \alpha\, P(B)\, \Sigma_f\, P(f)\, \Sigma_r\, P(r \mid B\ f)\, P(J \mid r)\, P(M \mid r) \\
&= \alpha\, P(B)\, \Sigma_f\, P(f)\, \Sigma_r\, P(r \mid B\ f)\, P(J \mid r)\, f_M(r) \\
&= \alpha\, P(B)\, \Sigma_f\, P(f)\, \Sigma_r\, f_r(r\ B\ f) \times f_J(r) \times f_M(r) \\
&= \alpha\, P(B)\, \Sigma_f\, P(f)\, f_{\mathbf{f}JM}(B\ f) \\
&= \alpha\, P(B)\, \Sigma_f\, f_f(f) \times f_{fJM}(B\ f) && \text{(mátrixos) tényezők} \\
&= \alpha\, P(B)\, f_{JM}(B) && \text{speciális szorzattal} \\
&= \alpha\, f_B(B) \times f_{JM}(B)
\end{aligned}$$

Feladat faktorai (FVT – faktor megfeleltetés):

P(b) $\times\Sigma$ P(f) $\times\Sigma$ P(r | b f) $\times$ P(j | r) $\times$ P(m | r)

b	f_b
0	0.999
1	0.001

$\times\Sigma$

f	f_f
0	0.998
1	0.002

$\times\Sigma$

b	f	r	f_{bfr}
0	0	0	0.999
0	0	1	0.001
0	1	0	0.71
0	1	1	0.29
1	0	0	0.06
1	0	1	0.94
1	1	0	0.05
1	1	1	0.95

$\times$

r	j	f_{rj}
0	0	0.95
0	1	0.05
1	0	0.1
1	1	0.9

$\times$

r	m	f_{rm}
0	0	0.99
0	1	0.01
1	0	0.3
1	1	0.7

Műveletek faktorokkal

- behelyettesítés
- változó eliminálása kiösszegzéssel
- faktorok szorzata

f(j,r) … f(j=Igaz,r)

r	j	f_{rj}
0	0	0.95
0	1	0.05
1	0	0.1
1	1	0.9

$\Rightarrow$

r	f_{rJ}
0	0.05
1	0.9

$$\Sigma_f\, f_{rbf}(r,b,f) = f_{rbf}(r,b,f=Hamis) + f_{rbf}(r,b,f=Igaz) = f_{rb}(r,b)$$

$$f_{rj}(r,j) \text{ x } f_{rm}(r,m) = f_{rjm}(r,j,m)$$

r	j	f_{rj}
0	0	0.95
0	1	0.05
1	0	0.1
1	1	0.9

$\times$

r	m	f_{rm}
0	0	0.99
0	1	0.01
1	0	0.3
1	1	0.7

$=$

r	j	m	f_{rjm}
0	0	0	0.95 x 0.999
0	0	1	0.95 x 0.01
0	1	0	0.05 x 0.99
0	1	1	0.05 x 0.01
1	0	0	0.1 x 0.3
1	0	1	0.1 x 0.7
1	1	0	0.9 x 0.3
1	1	1	0.9 x 0.7

$$P(B \mid J\ M) = \alpha\ P(B)\ \Sigma_f\ P(f)\ \Sigma_r\ P(r \mid B\ f)\ P(J \mid r)\ P(M \mid r)$$

b	f_b
0	0.999
1	$\boxed{0.001}$

$\times \Sigma$

f	f_f
0	0.998
1	0.002

$\times \Sigma$

b	f	r	f_{bfr}
0	0	0	0.999
0	0	1	0.001
0	1	0	0.71
0	1	1	0.29
1	0	0	0.06
1	0	1	0.94
1	1	0	0.05
1	1	1	0.95

$\times$

r	j	f_{rj}
0	0	0.95
0	1	0.05
1	0	0.1
1	1	0.9

$\times$

r	m	f_{rm}
0	0	0.99
0	1	0.01
1	0	0.3
1	1	0.7

$$P(B \mid J\ M) = \alpha\ P(B)\ \Sigma_f\ P(f)\ \Sigma_r\ P(r \mid B\ f)\ P(J \mid r)\ P(M \mid r)$$

f_B
0.001

$\times\Sigma$

f	f_f
0	0.998
1	0.002

$\times\Sigma$

f	r	f_{Bfr}
0	0	0.06
0	1	0.94
1	0	0.05
1	1	0.95

$\times$

r	f_{rJ}
0	0.05
1	0.9

$\times$

r	f_{rM}
0	0.01
1	0.7

f_B
0.001

$\times\Sigma$

f	f_f
0	0.998
1	0.002

$\times\Sigma$

f	r	f_{Bfr}
0	0	0.06
0	1	0.94
1	0	0.05
1	1	0.95

$\times$

r	f_{rJM}
0	0.05 x 0.01
1	0.9 x 0.7

f_B
0.001

$\times\Sigma$

f	f_f
0	0.998
1	0.002

$\times\Sigma$

f	r	f_{BJMfr}
0	0	0.06 x 0.05 x 0.01
0	1	0.94 x 0.9 x 0.7
1	0	0.05 x 0.05 x 0.01
1	1	0.95 x 0.9 x 0.7

$$P(B \mid J\,M) = \alpha\, P(B)\, \Sigma_f\, P(f)\, \Sigma_r\, P(r \mid B\, f)\, P(J \mid r)\, P(M \mid r)$$

f_B
0.001

$\times \Sigma$

f	f_f
0	0.998
1	0.002

$\times \Sigma$

f	r	f_{BJMfr}
0	0	0.06 x 0.05 x 0.01
0	1	0.94 x 0.9 x 0.7
1	0	0.05 x 0.05 x 0.01
1	1	0.95 x 0.9 x 0.7

.0005925125

f_B
0.001

$\times \Sigma$

f	f_f
0	0.998
1	0.002

$\times$

f	f_{BJMf}
0	0.06 x 0.05 x 0.01 + 0.94 x 0.9 x 0.7
1	0.05 x 0.05 x 0.01 + 0.95 x 0.9 x 0.7

f_B
0.001

$\times \Sigma$

f	f_{BJMf}
0	0.998 x (0.06 x 0.05 x 0.01 + 0.94 x 0.9 x 0.7)
1	0.002 x (0.05 x 0.05 x 0.01 + 0.95 x 0.9 x 0.7)

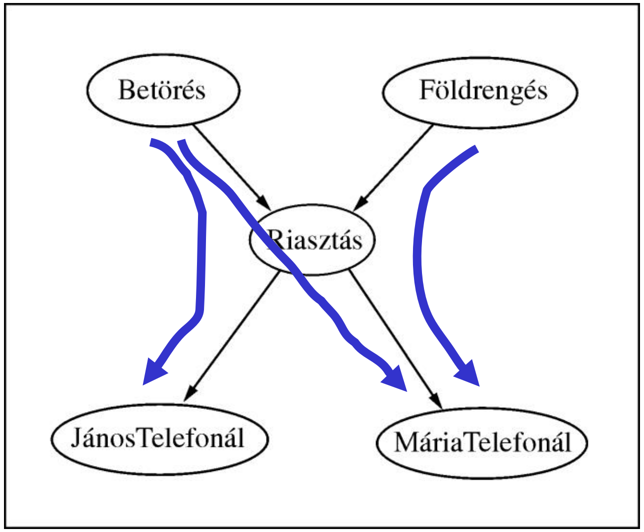
f_B
0.001

$\times$

f_{BJM}
0.998 x (0.06 x 0.05 x 0.01 + 0.94 x 0.9 x 0.7) + 0.002 x (0.05 x 0.05 x 0.01 + 0.95 x 0.9 x 0.7)

.001 x (.998 x (.06 x .05 x .01 + .94 x .9 x .7) +0.002 x (.05 x .05 x .01 + .95 x .9 x0.7))

Az egzakt következtetés komplexitása

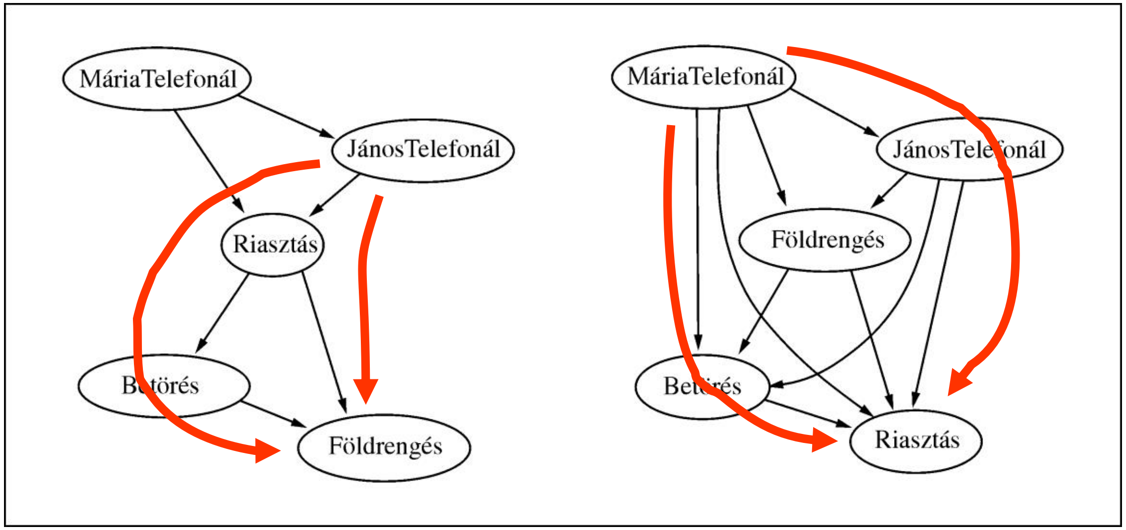


egyszeresen összekötött,

bármely két csp. között legfeljebb egyetlen egy irányítatlan út van

többszörösen összekötött

bármely két csomópont között több az út



Algoritmusok lekérdezések megválaszolására

A.) egyszeresen összekötött, fa gráf (polytree)

- létezik lineáris komplexitású algoritmus

B.) többszörösen összekötött

- nem létezik zárt alakú (rekurzív) algoritmus

➢ vagy a komplexitás megugrik (l. előbb), vagy csak közelítő módszerek maradnak

A számítás rekurzív hívásokból áll, X-ből indulva,
a hálóban minden lehetséges úton haladva (fel/le).

A rekurzió leáll a:

tény csomópontokon,
gyökércsomópontokon (szülő nélküli csomópont) és
levélcsomópontokon (gyermek nélküli csomópont).

Az algoritmus lineáris a háló csomópontjainak számában.

Ez csak azért lehetséges, mert a háló fa gráf.

Következtetés többszörösen összekötött hálókban

Előbbi általános egzakt módszerek ...

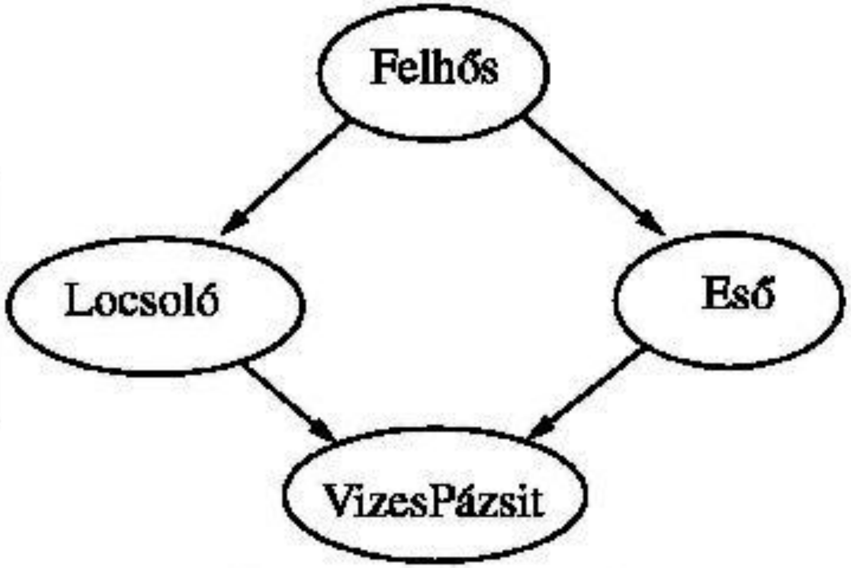
Csoportosító, összevonó eljárások:

- Átalakítják a hálót a valószínűségek szempontjából ekvivalens (de más topológiájú) fa gráffá, a nem megfelelő csomópontokat összevonva (majd lineáris).

Sztochasztikus szimulációs eljárások:

- A tárgytartomány nagyon nagy számú konkrét modelljét generálják le, ami konzisztens a valószínűségi háló által definiált eloszlással.
- Ez alapján az egzakt eredmények közelítését adják.

$P(F) = 0,5$



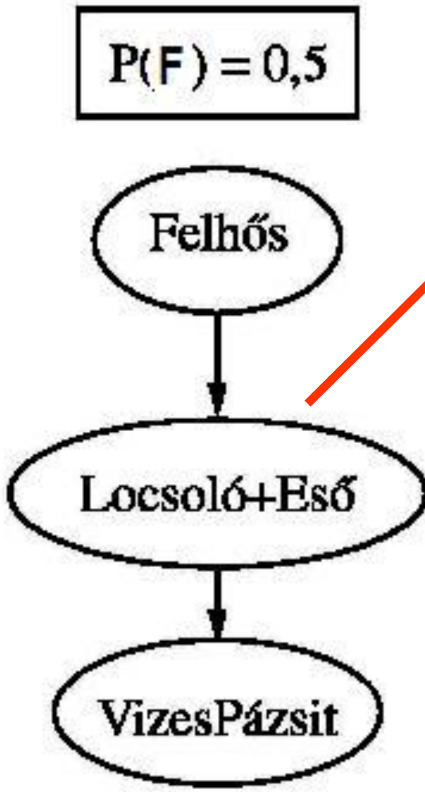
F	P(L)
I	0,10
H	0,50

F	P(E)
I	0,80
H	0,20

L	E	P(V)
I	I	0,99
I	H	0,90
H	I	0,90
H	H	0,00

(9 érték)

Összevonás



(Összevonás növeli értékek számát)

F	P(L+E=x)			
	II	IH	HI	HH
I	0,08	0,02	0,72	~~0,18~~
H	0,10	0,40	0,10	~~0,40~~

(11 érték)

L+E	P(V)
I I	0,99
I H	0,90
H I	0,90
H H	0,00

Következtetés egyszerűbb, de az összevonás exponenciális is lehet, ügyes összevonás komplexitásba kerül.

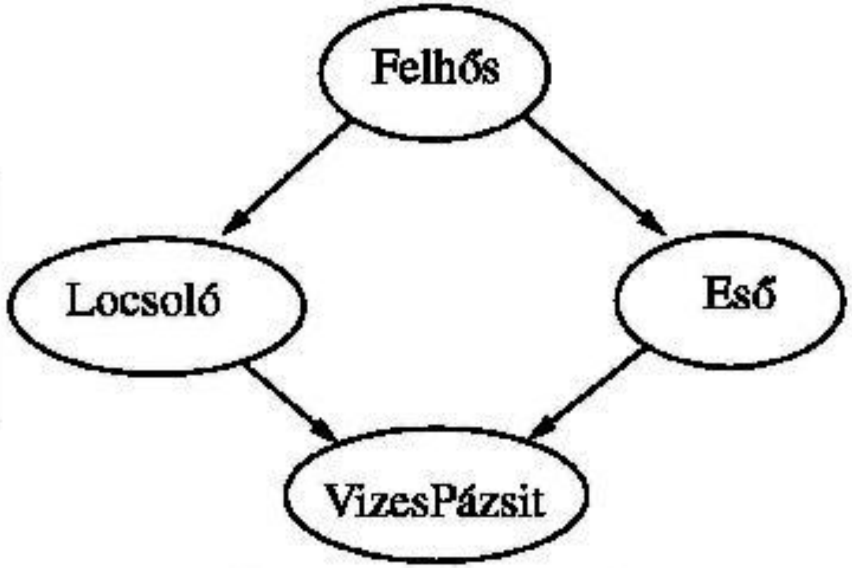
Közvetlen logikai mintavételezés: sztochasztikus szimuláció

Ismételten leszimuláljuk a háló által leírt világot
(háló = véletlen számgenerátor, elemi eseményeket sorsol)
a kérdéses valószínűségeket a megfelelő események előfordulási gyakoriságával becsüljük meg.

Minden szimulációs ciklusban:
- a gyökér csomópontok értékeinek véletlen sorsolása az a priori valószínűségeinek megfelelően
- ez szülői feltétel a gyerek csomópontok számára, a gyerek sorsolásához a valószínűséget a szülői feltételhez rendelt FVT-ból választjuk
- ugyanezt tesszük lefelé haladva a hálóban

A P(X) megbecsüléséhez a folyamatot sokszor megismételjük, és kiszámítjuk az esemény relatív frekvenciáját.

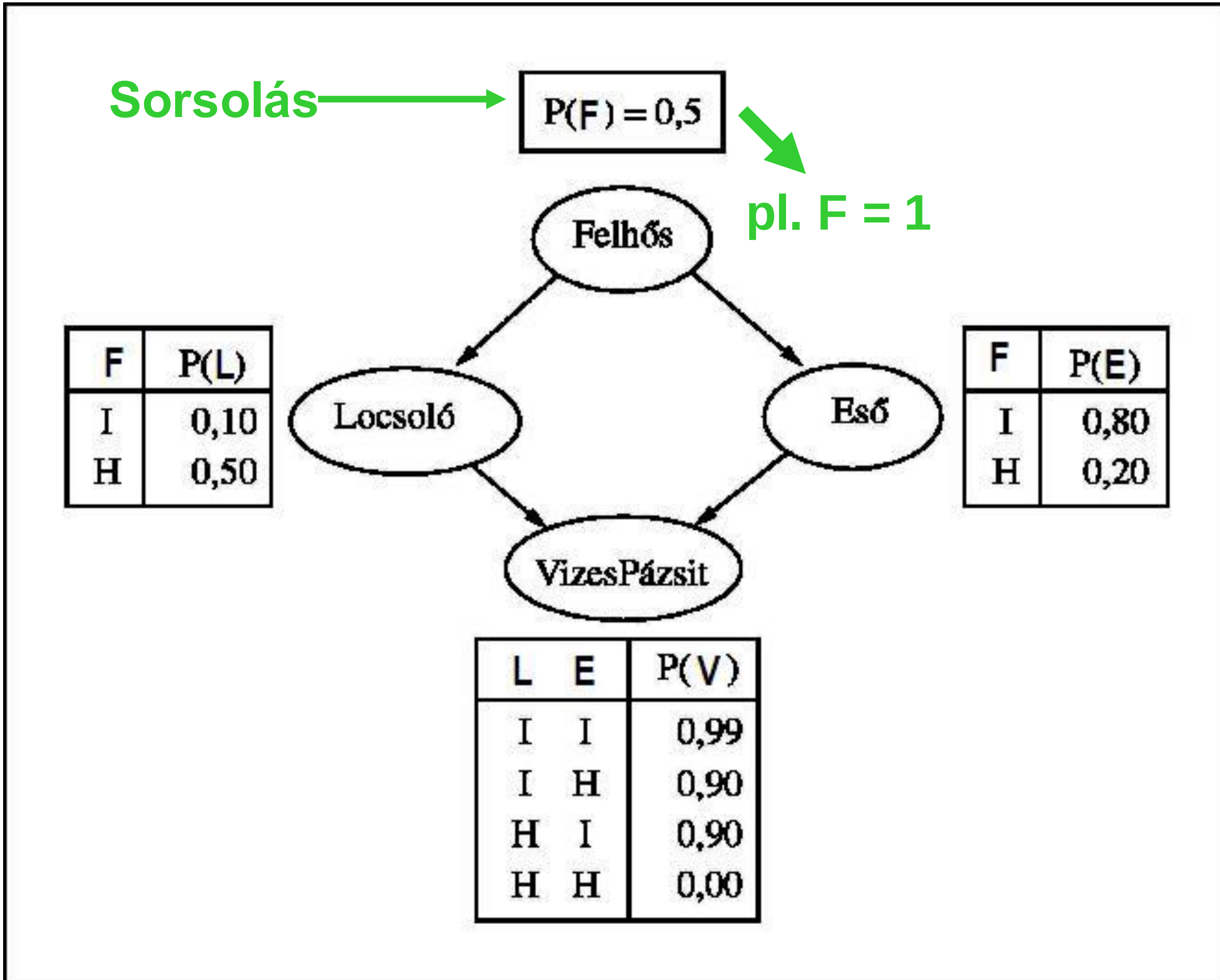
Sorsolás → $P(F) = 0{,}5$



F	P(L)
I	0,10
H	0,50

F	P(E)
I	0,80
H	0,20

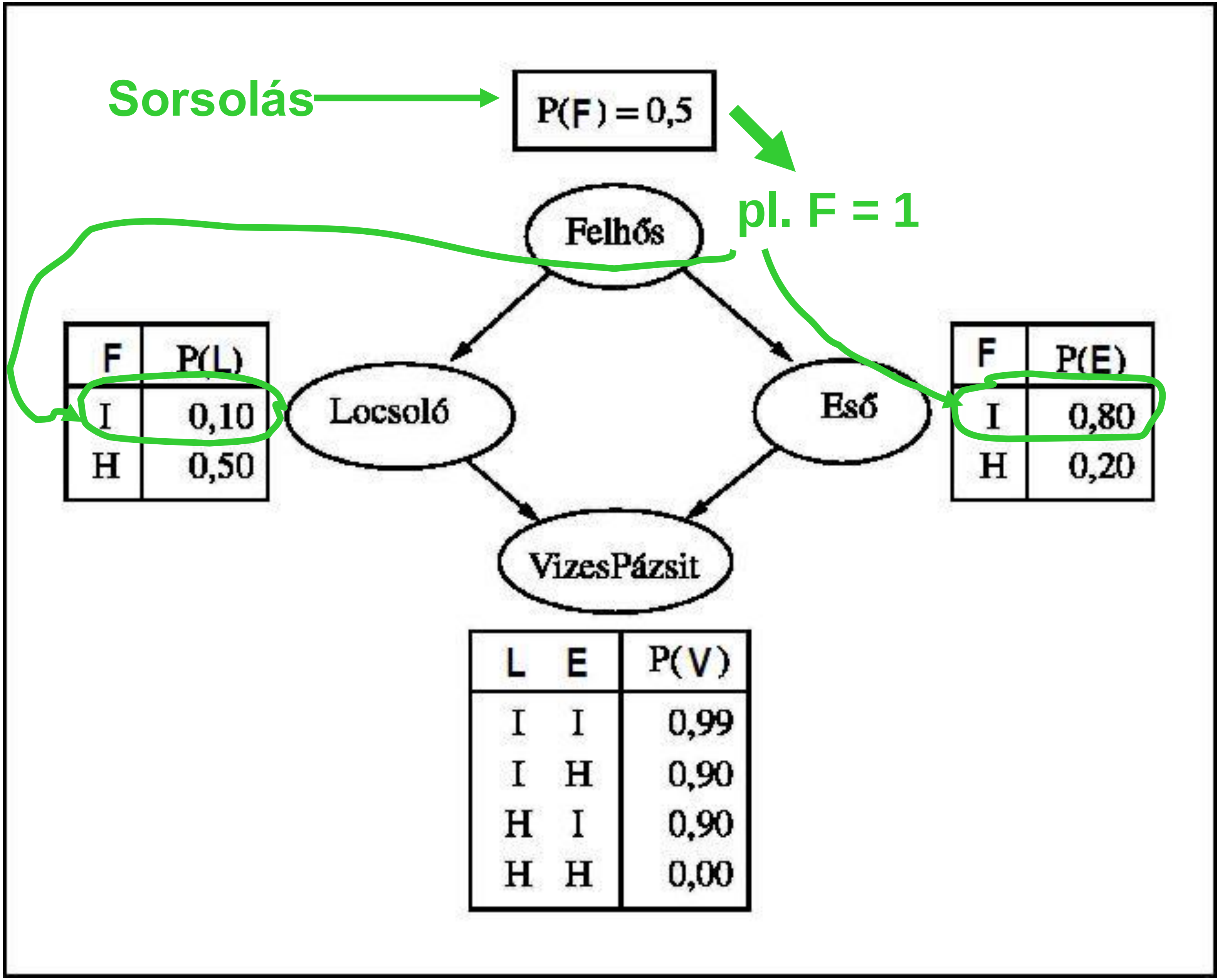
L	E	P(V)
I	I	0,99
I	H	0,90
H	I	0,90
H	H	0,00



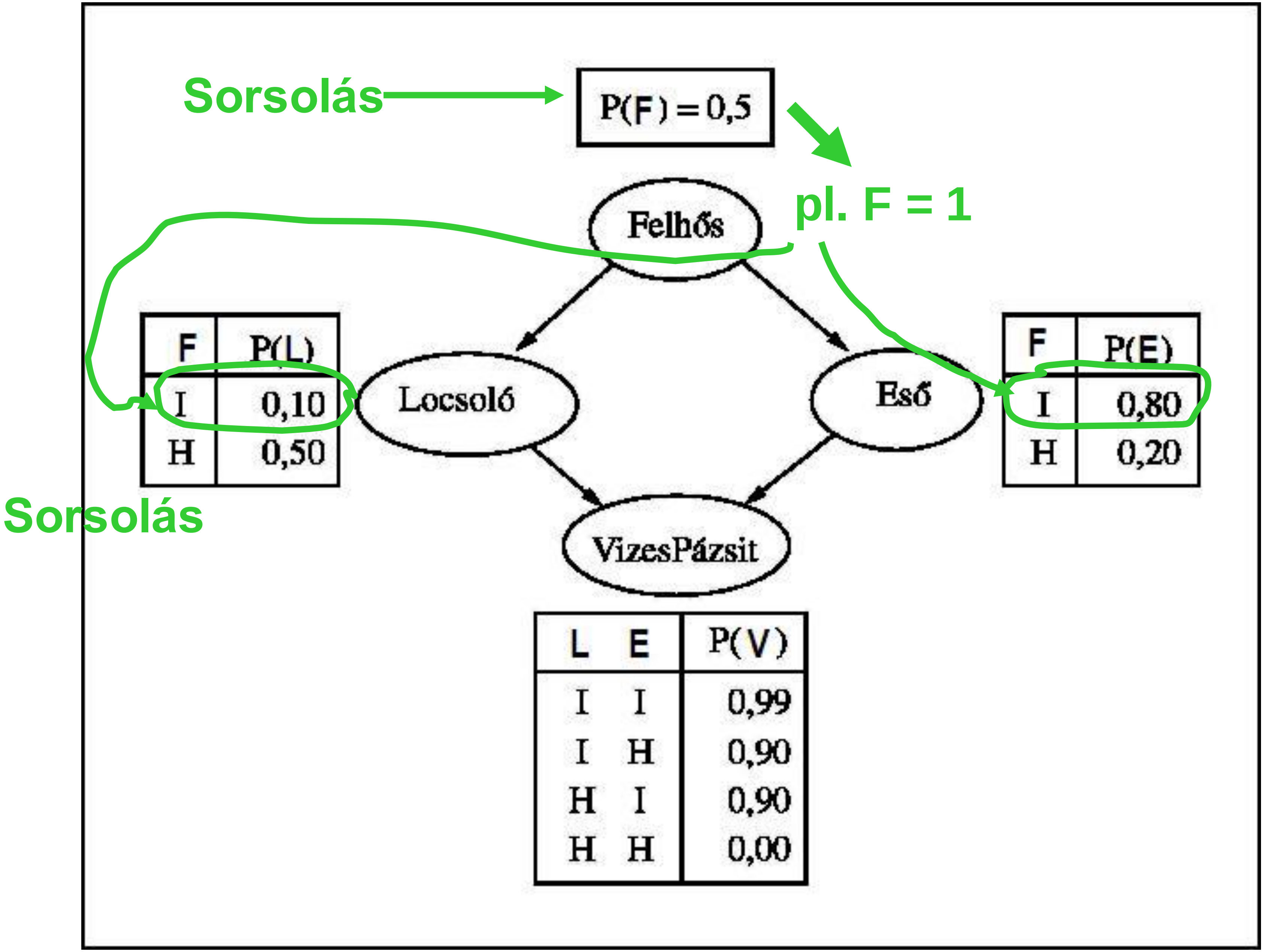
F	P(L)
I	0,10
H	0,50

F	P(E)
I	0,80
H	0,20

L	E	P(V)
I	I	0,99
I	H	0,90
H	I	0,90
H	H	0,00



Sorsolás
P(F) = 0,5
pl. F = 1
Felhős
Locsoló
Eső
VizesPázsit
F P(L)
I 0,10
H 0,50
F P(E)
I 0,80
H 0,20
L E P(V)
I I 0,99
I H 0,90
H I 0,90
H H 0,00

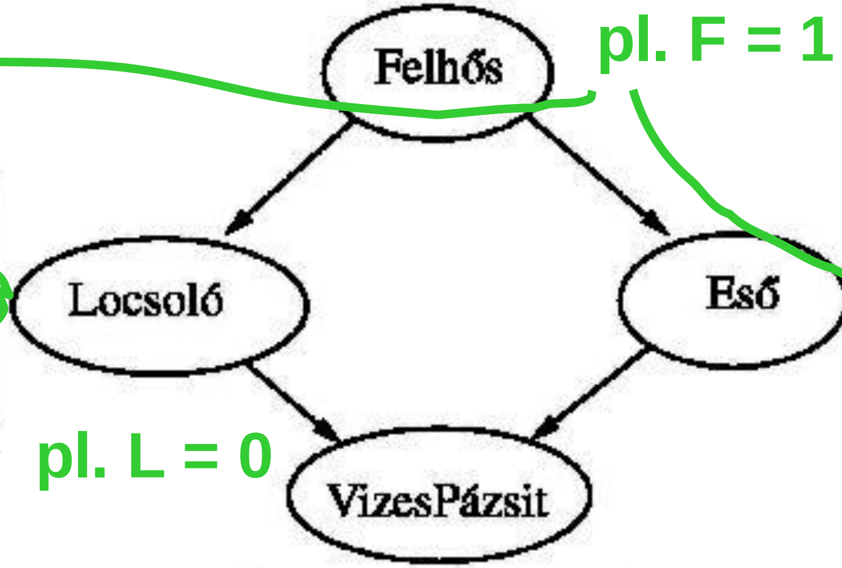


Sorsolás
P(F) = 0,5
pl. F = 1
Felhős
F P(L)
I 0,10
H 0,50
Locsoló
Eső
F P(E)
I 0,80
H 0,20
Sorsolás
VizesPázsit
L E P(V)
I I 0,99
I H 0,90
H I 0,90
H H 0,00

Sorsolás

P(F) = 0,5

pl. F = 1

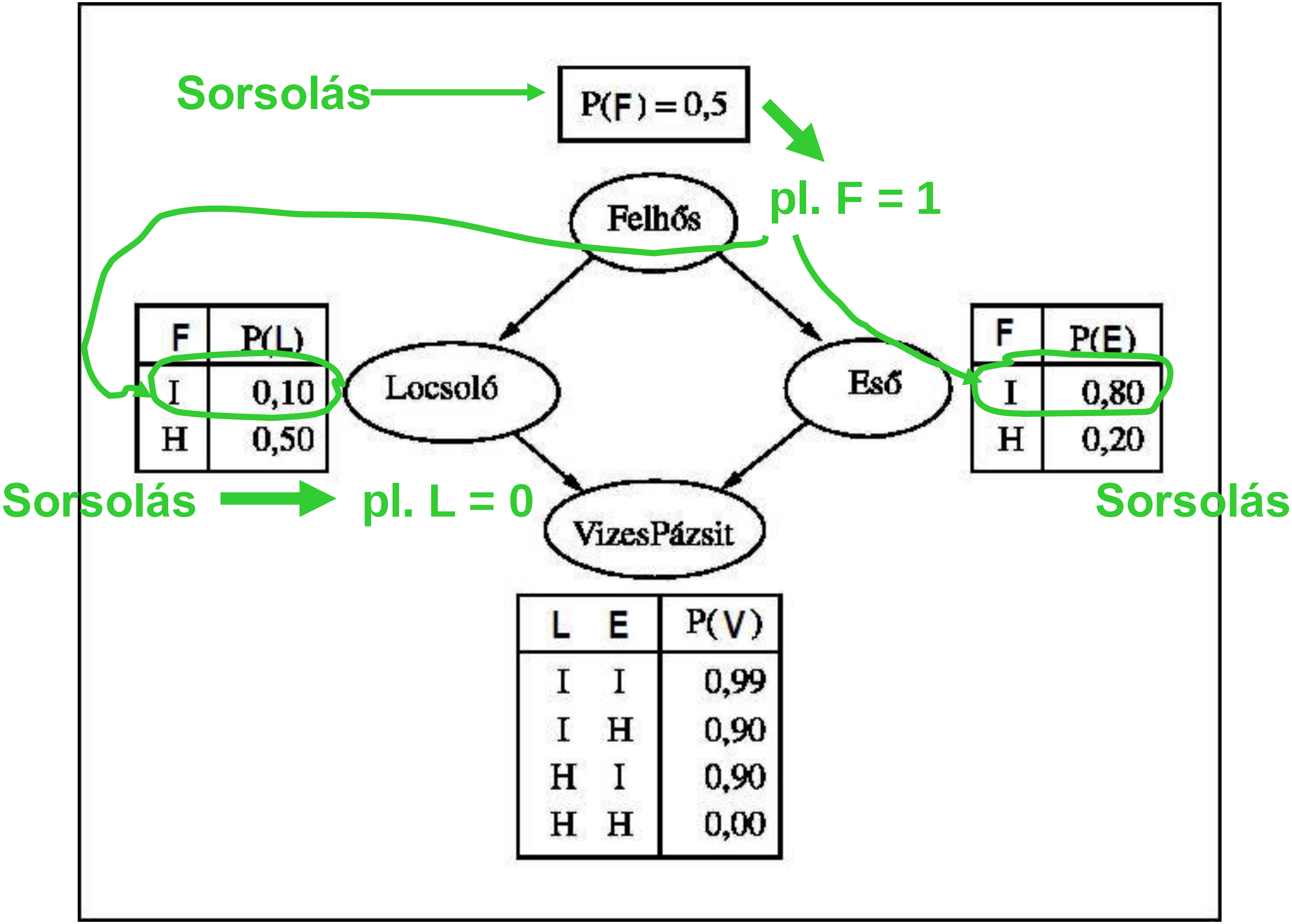


F	P(L)
I	0,10
H	0,50

F	P(E)
I	0,80
H	0,20

Sorsolás pl. L = 0

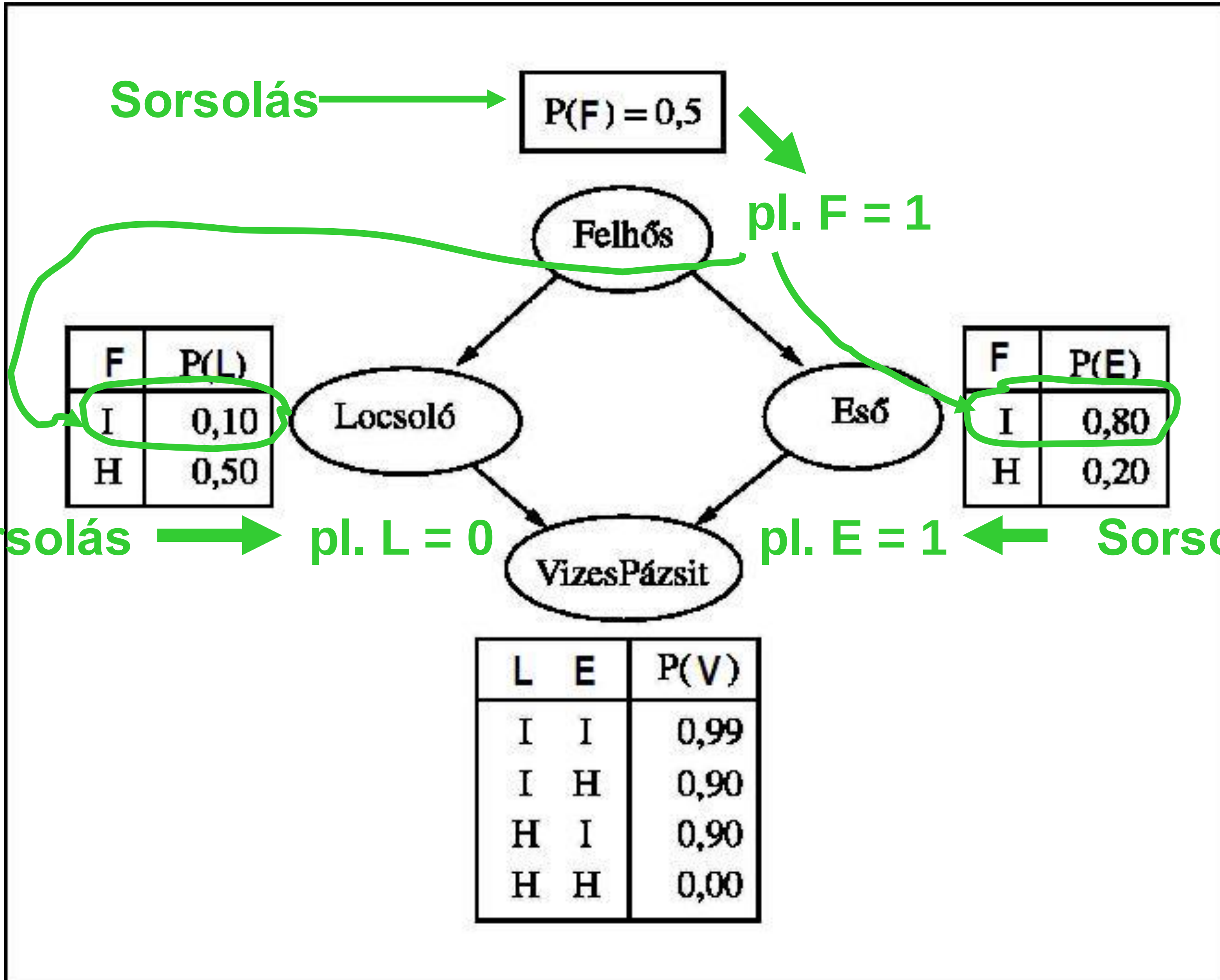
L	E	P(V)
I	I	0,99
I	H	0,90
H	I	0,90
H	H	0,00



F	P(L)
I	0,10
H	0,50

F	P(E)
I	0,80
H	0,20

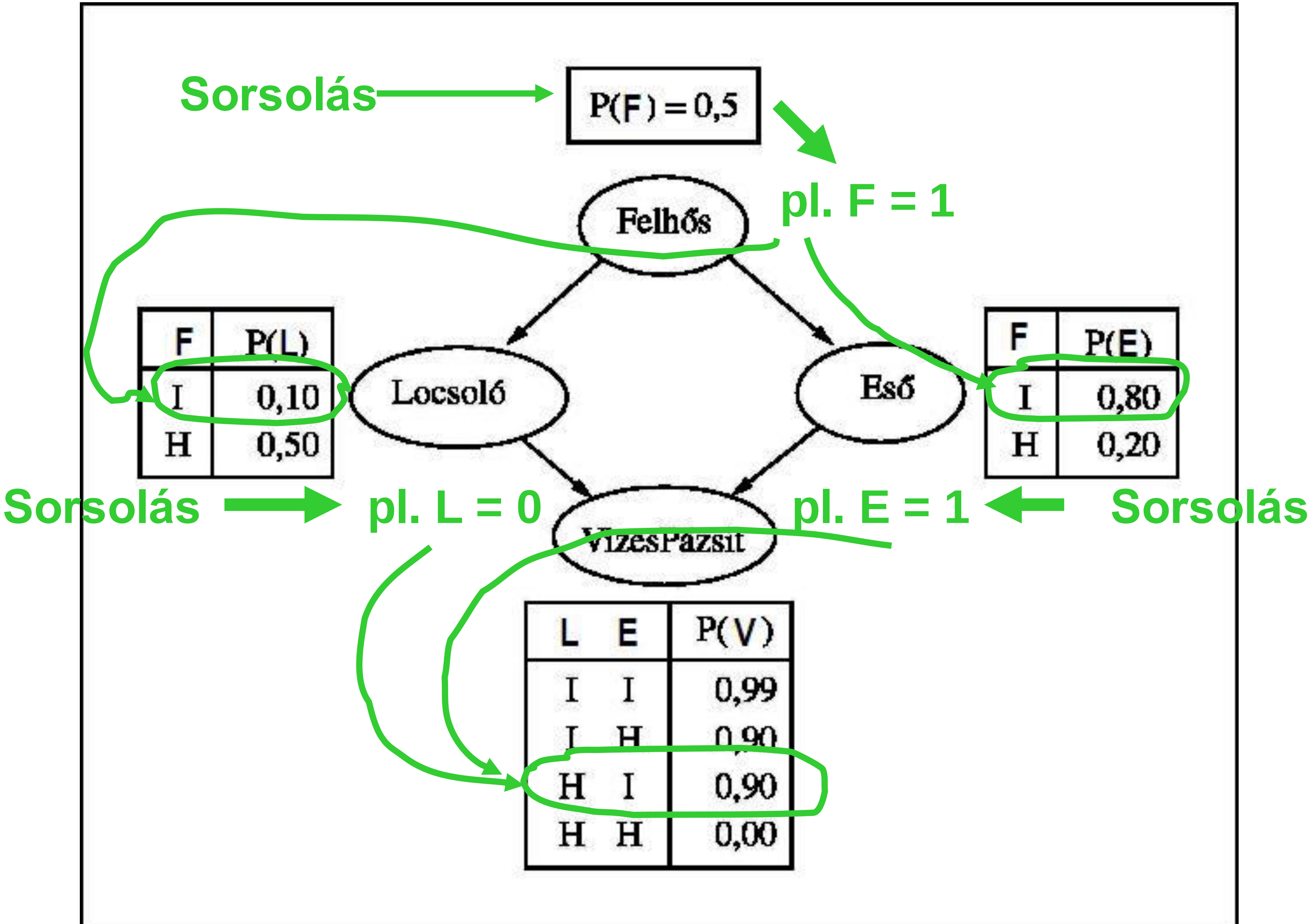
L	E	P(V)
I	I	0,99
I	H	0,90
H	I	0,90
H	H	0,00



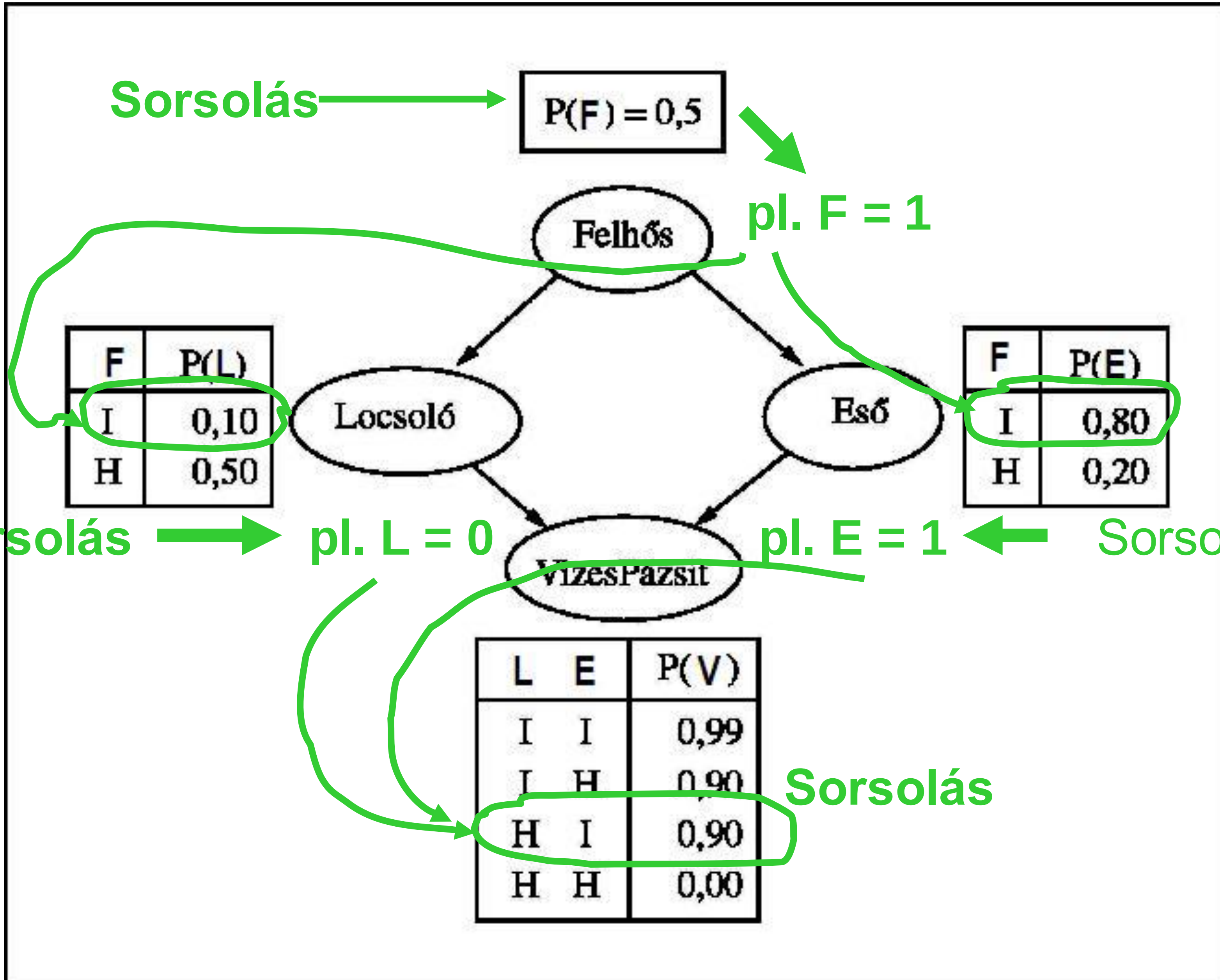
F	P(L)
I	0,10
H	0,50

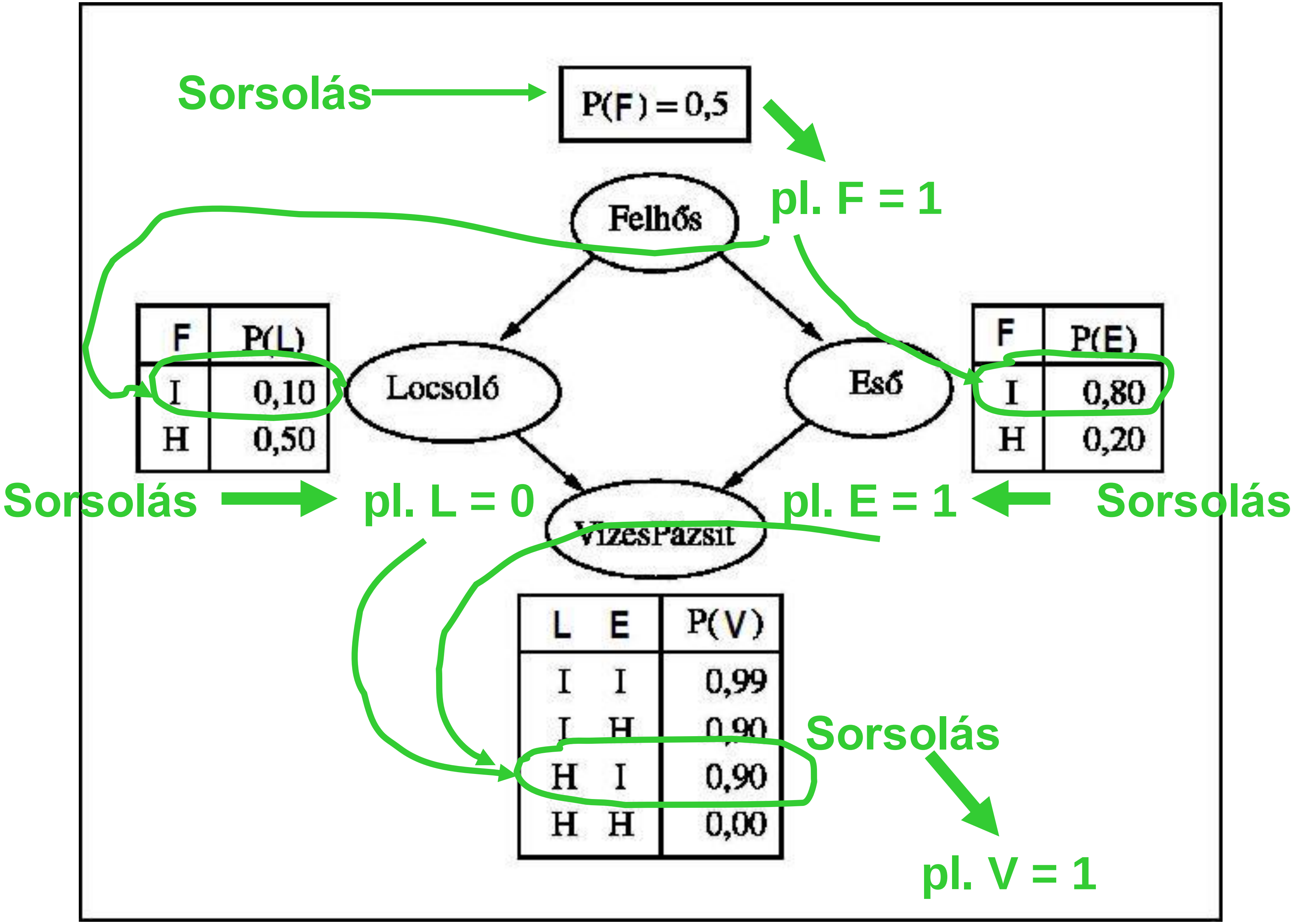
F	P(E)
I	0,80
H	0,20

L	E	P(V)
I	I	0,99
I	H	0,90
H	I	0,90
H	H	0,00



Sorsolás
P(F) = 0,5
pl. F = 1
Felhős
F P(L)
I 0,10
H 0,50
Locsoló
Eső
F P(E)
I 0,80
H 0,20
Sorsolás
pl. L = 0
pl. E = 1
Sorsolás
VizesPázsit
L E P(V)
I I 0,99
I H 0,90
H I 0,90
H H 0,00


Sorsolás
P(F) = 0,5
pl. F = 1
Felhős
F
P(L)
I
0,10
H
0,50
Locsoló
Eső
F
P(E)
I
0,80
H
0,20
Sorsolás
pl. L = 0
pl. E = 1
Sorsolás
VizesPázsit
L
E
P(V)
I
I
0,99
I
H
0,90
H
I
0,90
H
H
0,00
Sorsolás



Sorsolás
P(F) = 0,5
pl. F = 1
Felhős
F P(L)
I 0,10
H 0,50
Locsoló
Eső
F P(E)
I 0,80
H 0,20
Sorsolás
pl. L = 0
pl. E = 1
Sorsolás
VizesPázsit
L E P(V)
I I 0,99
I H 0,90
H I 0,90
H H 0,00
Sorsolás
pl. V = 1

Elutasító logikai mintavételezés

először a háló által megadott a priori eloszlásból generál mintákat, majd elutasítja azokat, amelyek nem illeszkednek a bizonyítékhoz.

Sorsolások száma = *Locsoló* = *Igaz* N_L (nincs elutasítva)
Locsoló = *Igaz* és *Eső* = *Igaz* N_{LE}

P(*Eső* | *Locsoló*) $\leftarrow$ N_{LE} / N_L

Az elutasító mintavétel legnagyobb hibája, hogy nagyon sok mintát utasít el.

Az **e** bizonyítékkal konzisztens minták aránya exponenciálisan egyre kevesebb, ahogy a bizonyítékváltozók száma nő (**több feltétel = egyre ritkább esemény**), így az eljárás egyszerűen használhatatlan komplex problémákban.

A fő probléma: ha az érdekes események igen ritkán fordulnak elő.

Pl. a következő értéket szeretnénk tudni:

$$P(\mathit{Eső} \mid \mathit{VizesPázsit} \wedge \mathit{Locsoló})$$

Mivel az $\mathit{VizesPázsit} \wedge \mathit{Locsoló}$ ritkán fordul elő a világban, a szimuláció során a legtöbb menetben más értékeket kapunk ezekre a tény változókra, és ezeket az eseteket nem tudjuk felhasználni.

Valószínűségi súlyozás

amikor tény változóhoz érünk, ahelyett, hogy véletlenül választanánk egy értéket (a feltételes valószínűségek alapján), mindig a tény változó adott kívánt értékét választjuk

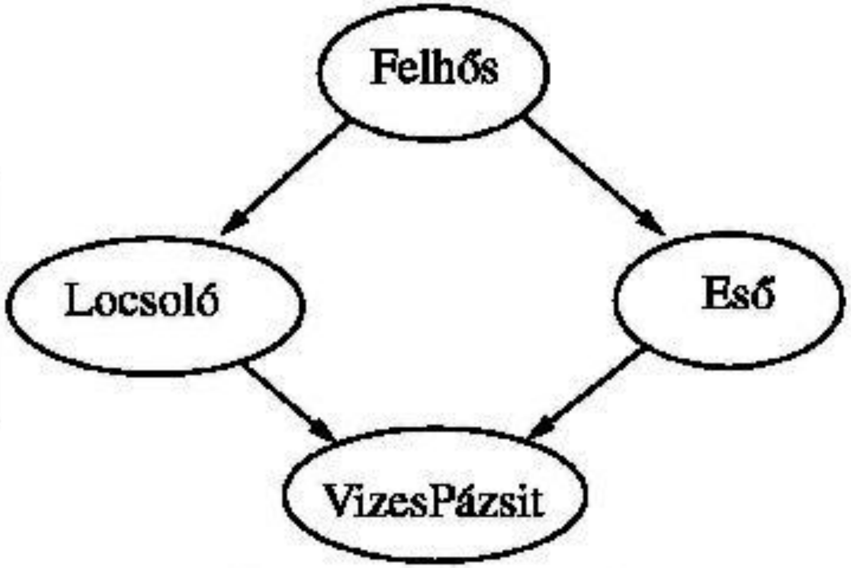
(= a ritka esemény **mindig** forduljon elő!)

de felhasználjuk a feltételes valószínűségét, hogy az mennyire is valószínű.

(az így kapott hozzájárulás a relatív frekvenciához nem 1, hanem kis valószínűség, a likelihood értékek szorzata)

Példa: *P(Eső | VizesPázsit, Locsoló)*

Sorsolás → $P(F) = 0,5$



F	P(L)
I	0,10
H	0,50

F	P(E)
I	0,80
H	0,20

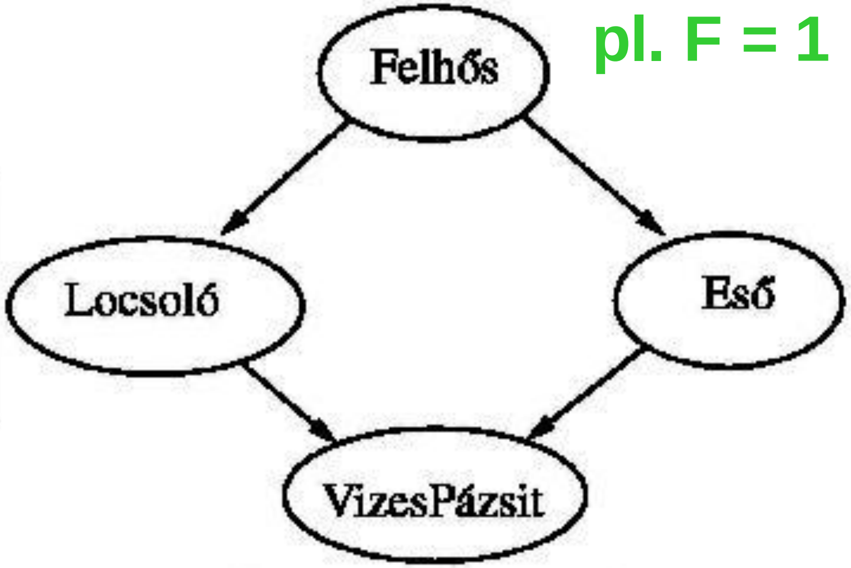
L	E	P(V)
I	I	0,99
I	H	0,90
H	I	0,90
H	H	0,00

W = 1

Példa: *P(Eső | VizesPázsit, Locsoló)*

Sorsolás → $P(F) = 0{,}5$

pl. F = 1



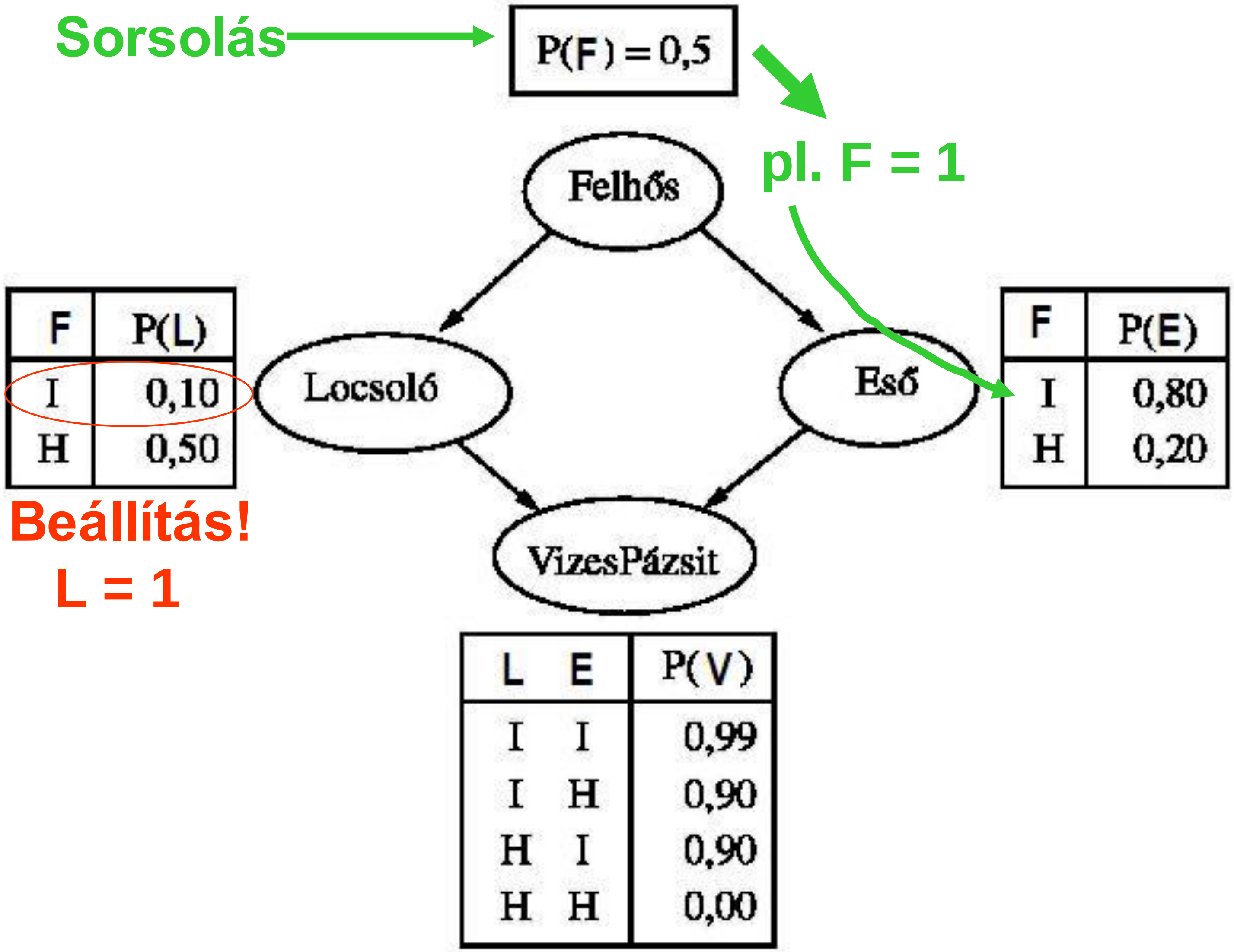
F	P(L)
I	0,10
H	0,50

F	P(E)
I	0,80
H	0,20

L	E	P(V)
I	I	0,99
I	H	0,90
H	I	0,90
H	H	0,00

W = 1

Példa: *P(Eső | VizesPázsit, Locsoló)*



F	P(L)
I	0,10
H	0,50

F	P(E)
I	0,80
H	0,20

L	E	P(V)
I	I	0,99
I	H	0,90
H	I	0,90
H	H	0,00

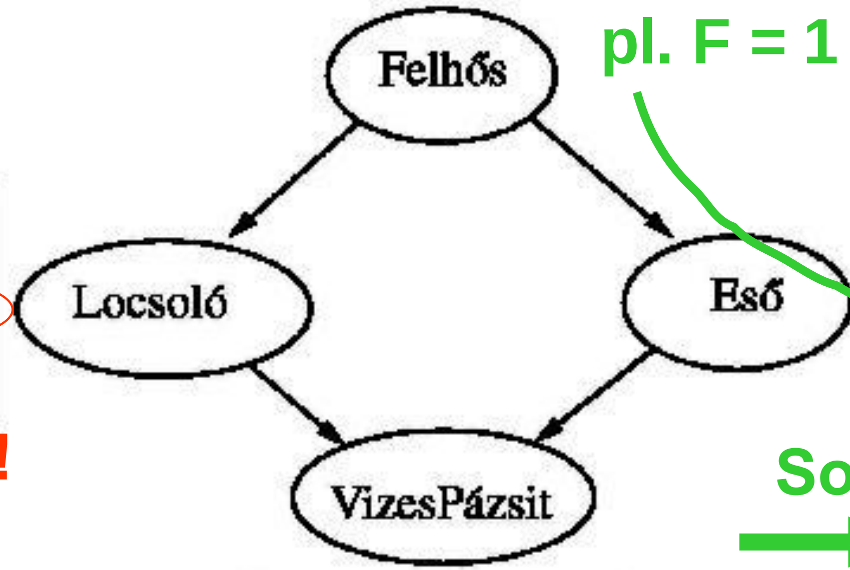
$W \leftarrow W \times P(Locsoló{=}igaz|Felhős{=}igaz) = 0,1.$

Példa: *P(Eső | VizesPázsit, Locsoló)*

Sorsolás

$P(F) = 0{,}5$

pl. F = 1



F	P(L)
I	0,10
H	0,50

Beállítás!
L = 1

F	P(E)
I	0,80
H	0,20

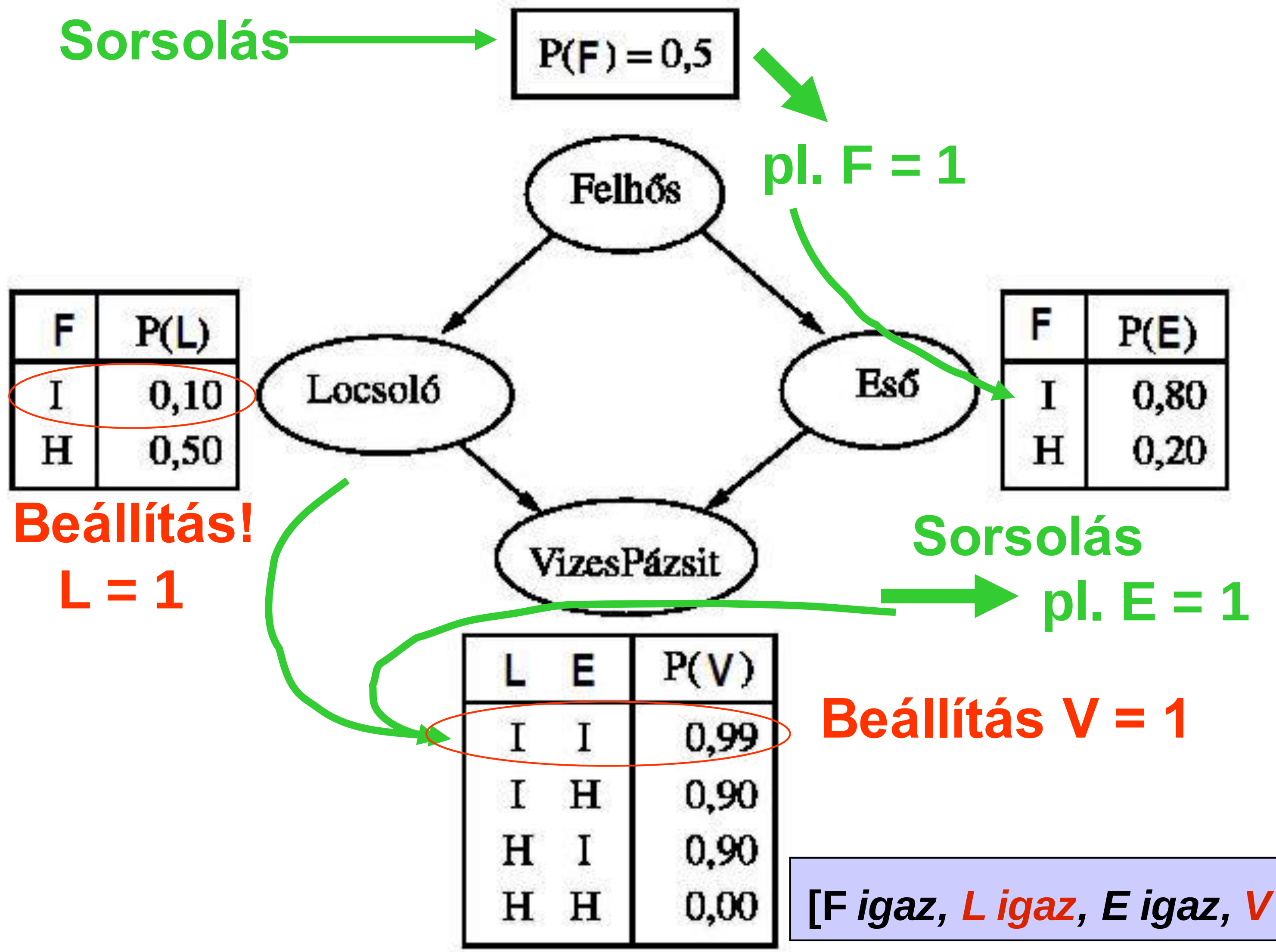
Sorsolás

pl. E = 1

L	E	P(V)
I	I	0,99
I	H	0,90
H	I	0,90
H	H	0,00

W ← W × P(Locsoló=igaz|Felhős=igaz) = 0,1.

Példa: *P(Eső | VizesPázsit, Locsoló)*



F	P(L)
I	0,10
H	0,50

F	P(E)
I	0,80
H	0,20

L	E	P(V)
I	I	0,99
I	H	0,90
H	I	0,90
H	H	0,00

[F *igaz*, *L igaz*, *E igaz*, *V igaz*]

W←W × P(VizesPázsit=igaz|Locsoló=igaz, Esős=igaz) = 0,099.

Valószínűségi súlyozás

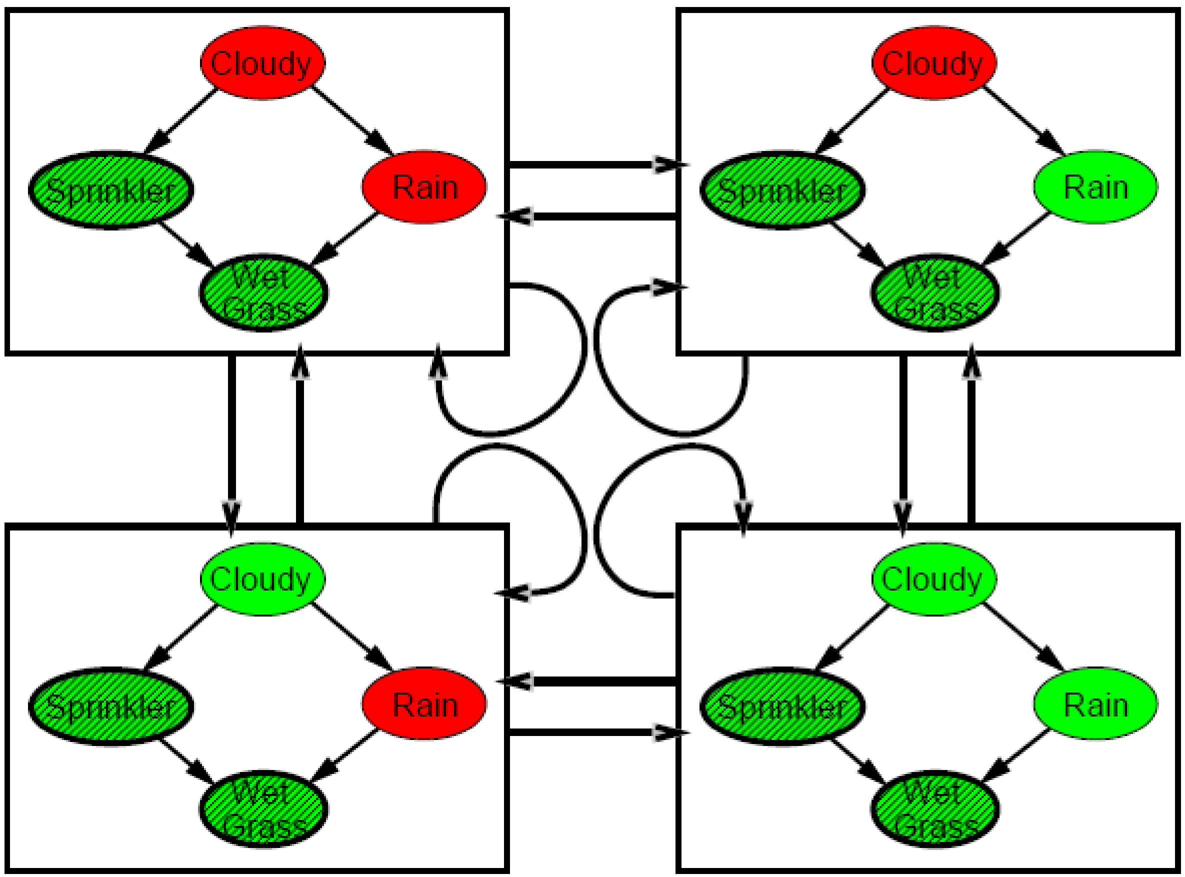
- valószínűségi súlyozás az összes generált mintát felhasználja
- a teljesítménye leromlik, amint a bizonyíték változók száma növekszik (a legtöbb mintának nagyon kis súlya lesz és így a súlyozott becslést főként a minták azon töredéke határozza meg, amelyek egy elenyésző valószínűségnél jobban illeszkednek a bizonyítékokhoz)

- **gyorsabban konvergál**, mint a logikai mintavételezés
- **igen nagy méretű** valószínűségi hálókat is képes kezelni
- probléma: pontos valószínűség kis valószínűségű eseményekre = **hosszú idő**
- egy adott pontossághoz szükséges futási idő **fordítottan arányos** az esemény valószínűségével

Kritikus események: … egy nukleáris reaktor leolvadása egy adott napon $P \cong 0$, de nagy különbség van 10^{-5} és 10^{-10} között, így ekkor is pontos értékeket kell kapnunk

Markov Chain Monte Carlo (MCMC) mintavételezés (változók értékkészletén)

With $Sprinkler = true, WetGrass = true$, there are four states:

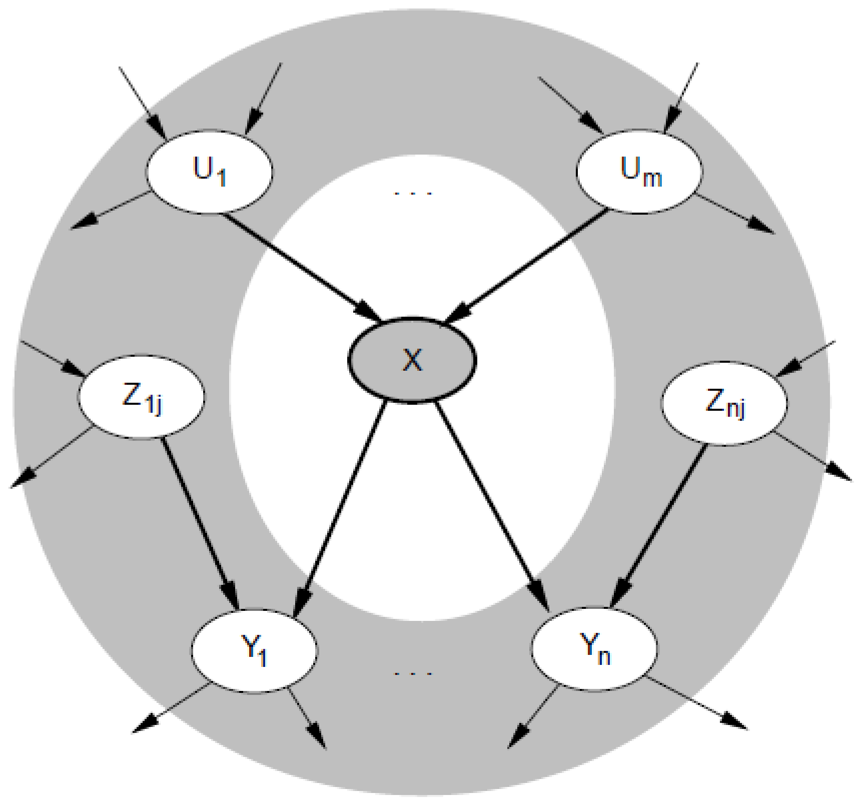


Wander about for a while, average what you see

Adapted from AIMA

MCMC mintavételezés Markov-takaró alapján (közelítő következtetés)

Each node is conditionally independent of all others given its Markov blanket: parents + children + children's parents



Adapted from AIMA

MCMC mintavételezés Markov-takaró alapján (közelítő következtetés)

"State" of network = current assignment to all variables. Generate next state by sampling one variable given Markov blanket. Sample each variable in turn, keeping evidence fixed

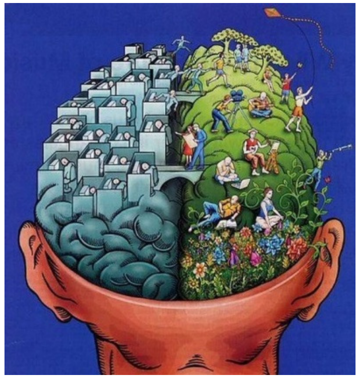
function MCMC-ASK($X, \mathbf{e}, bn, N$) **returns** an estimate of $P(X|\mathbf{e})$
  local variables: $\mathbf{N}[X]$, a vector of counts over X, initially zero
    $\mathbf{Z}$, the nonevidence variables in bn
    $\mathbf{x}$, the current state of the network, initially copied from $\mathbf{e}$

  initialize $\mathbf{x}$ with random values for the variables in $\mathbf{Y}$
  for $j = 1$ to N **do**
    for each Z_i in $\mathbf{Z}$ **do**
      sample the value of Z_i in $\mathbf{x}$ from $\mathbf{P}(Z_i|mb(Z_i))$
        given the values of $MB(Z_i)$ in $\mathbf{x}$
      $\mathbf{N}[x] \leftarrow \mathbf{N}[x] + 1$ where x is the value of X in $\mathbf{x}$
  return NORMALIZE($\mathbf{N}[X]$)

Can also choose a variable to sample at random each time

Adapted from AIMA

Budapesti Műszaki és Gazdaságtudományi Egyetem
Méréstechnika és Információs rendszerek Tanszék



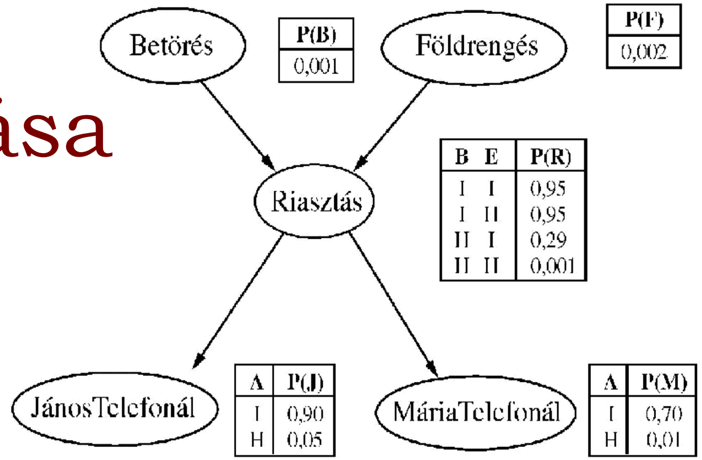
Bayes-hálók tanulása

Előadó: Dr. Hullám Gábor

Előadás anyaga: Dr. Dobrowiecki Tadeusz
Dr. Antal Péter

MŰEGYETEM 1782

Valószínűségi hálók tanítása

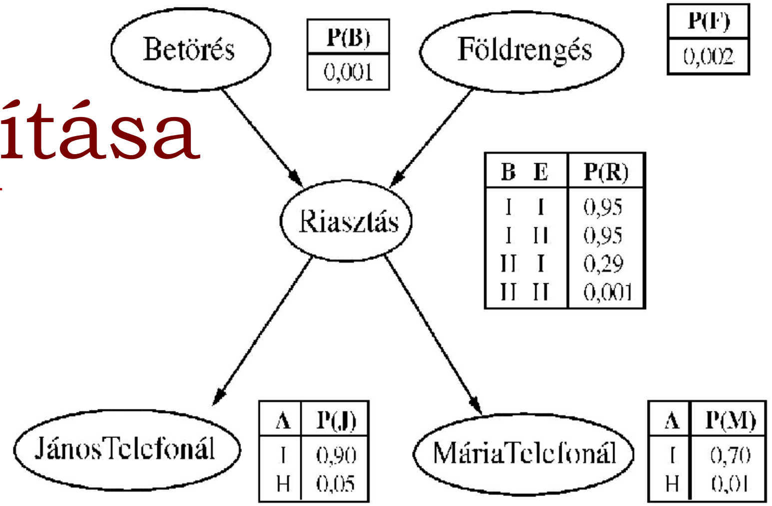


Ismert struktúra, teljesen megfigyelhető változók:

- tanulandó: a feltételes valószínűségek táblázata, közvetlenül becsülhető a példahalmaz alapján

paramétertanulás

Valószínűségi hálók tanítása



Ismeretlen struktúra, teljesen megfigyelhető változók:

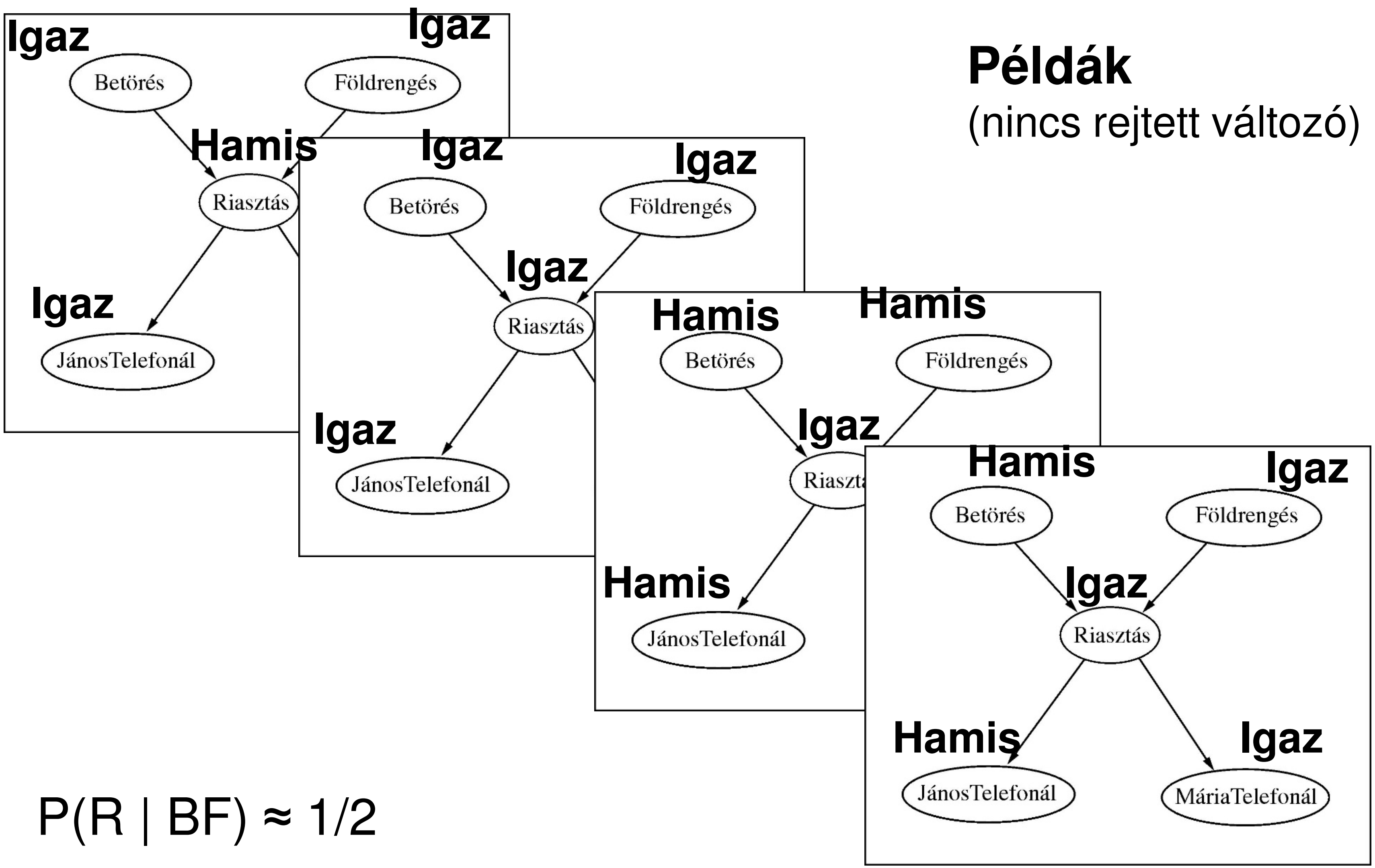
tanulandó: a háló topológiája

- struktúrák terében való keresés, a keresést az egyes struktúráknak az adatok modellezési képessége irányítja

struktúratanulás

Példák

(nincs rejtett változó)

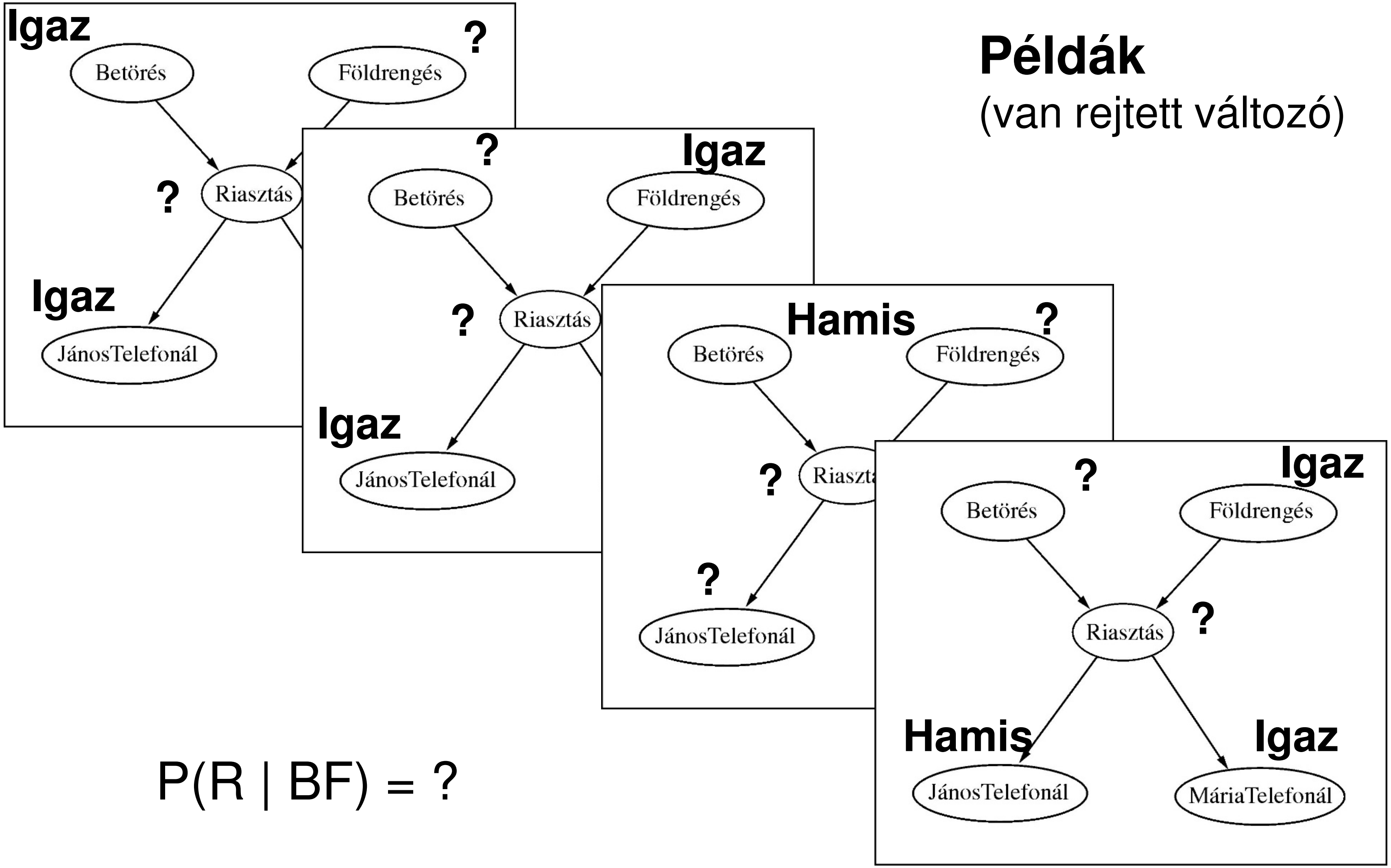


P(R | BF) ≈ 1/2

• • •

Példák

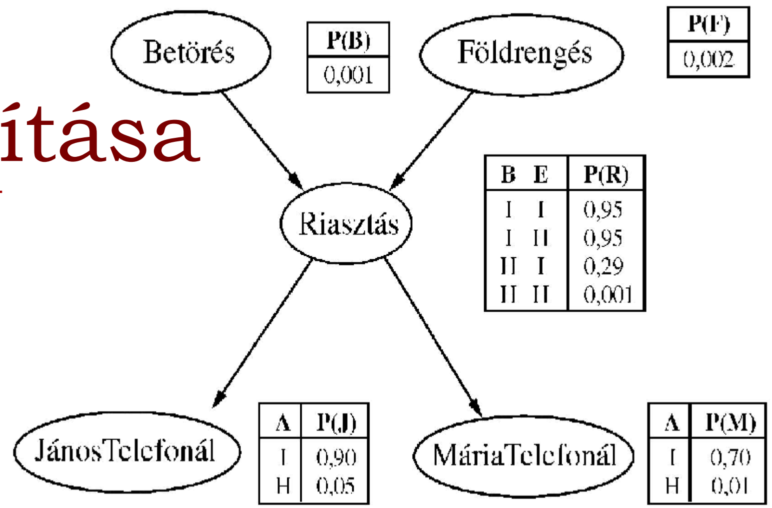
(van rejtett változó)



$P(R \mid BF) = ?$

…

Valószínűségi hálók tanítása



Ismert struktúra, rejtett változók:

- EM (expectation maximization) algoritmus

Ismeretlen struktúra, rejtett változók:

- Strukturális EM
- Egyéb közelítő megoldások

Gépi tanulás (MSc)

Valószínűségi következtető és döntéstámogató rendszerek (MSc)

Bayes tanulás

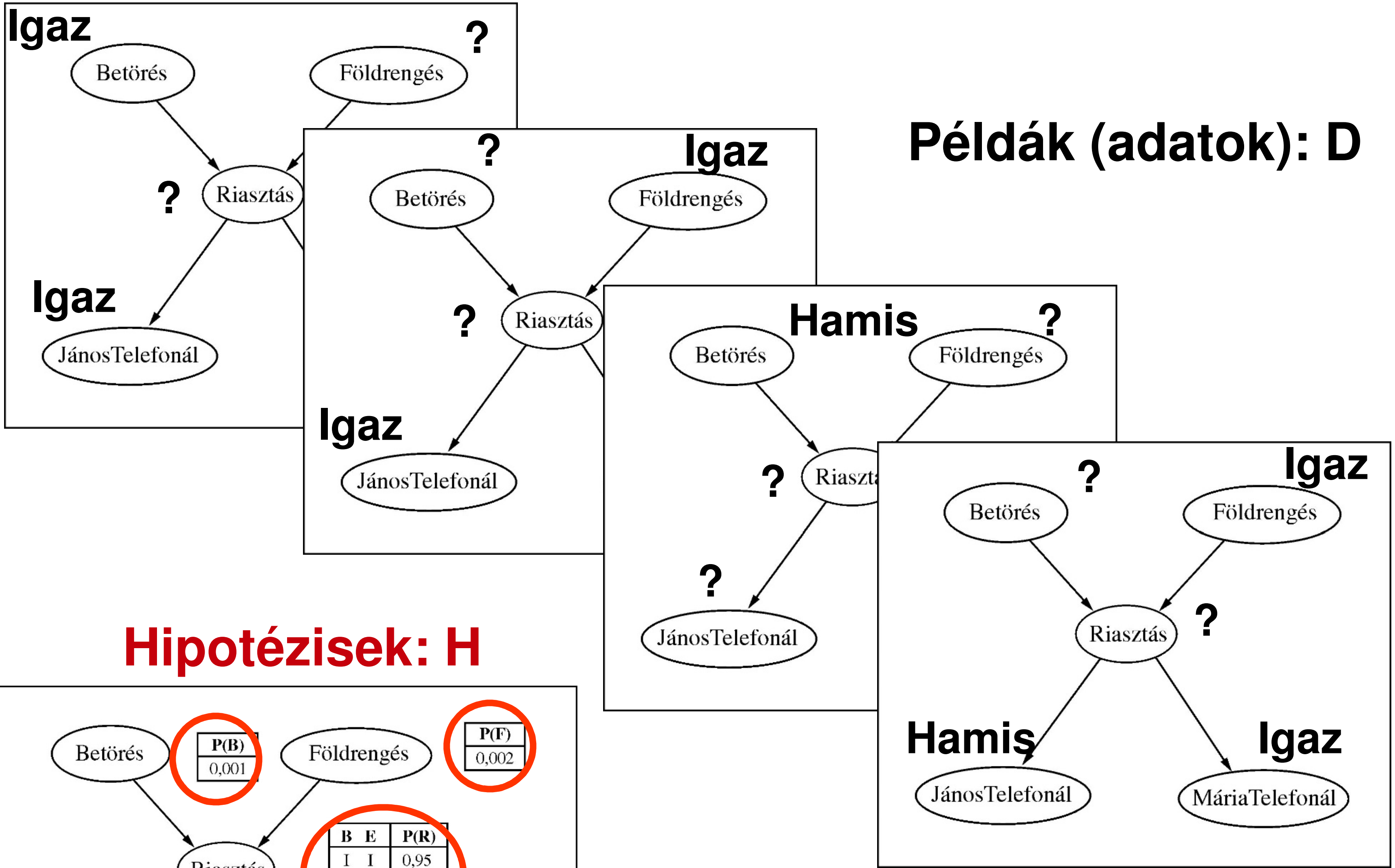
Általános predikció - hipotézis konstruálása adatok alapján:

1. az összes hipotézis valószínűségét megbecsüljük az adatokból.
2. a hipotézisek alapján predikciókat készítünk.
3. a predikciókat a hipotézisek a posteriori valószínűségével súlyozzuk.

D adatok, $\boldsymbol{H_1}$, $\boldsymbol{H_2}$,... hipotézisek, ismeretlen ***X*** predikciója
- minden egyes $\boldsymbol{H_i}$ az ***X*** egy teljes eloszlását specifikálja

$$P(X) = \sum_i P(X \mid H_i) P(H_i)$$

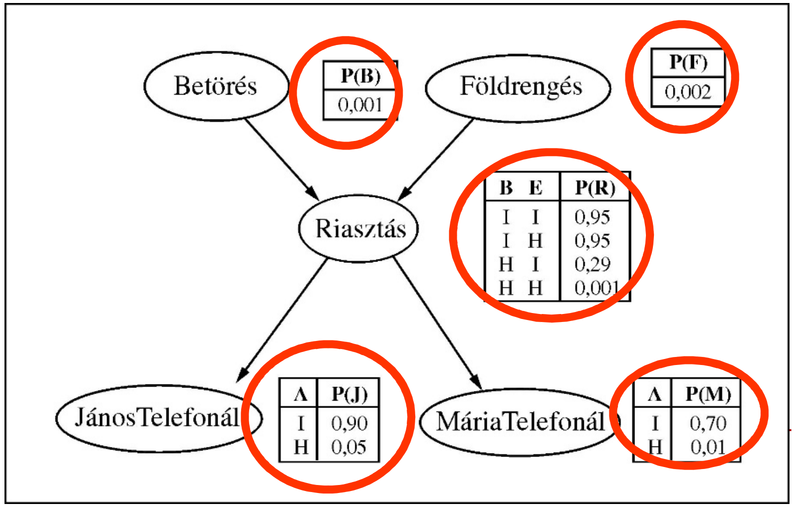
$$P(X \mid D) = \sum_i P(X \mid D, H_i) P(H_i \mid D) = \sum_i P(X \mid H_i) P(H_i \mid D)$$



Példák (adatok): D

• • •

Hipotézisek: H



Bayes-tanulás

D adatok, $\boldsymbol{H_1}$, $\boldsymbol{H_2}$,... hipotézisek, ismeretlen ***X*** predikciója
- minden egyes $\boldsymbol{H_i}$ az ***X*** egy teljes eloszlását specifikálja

$$P(X) = \sum_i P(X \mid H_i) P(H_i)$$

$$P(X \mid D) = \sum_i P(X \mid D, H_i) P(H_i \mid D) = \sum_i P(X \mid H_i) P(H_i \mid D)$$

- $P(H_i|D) = \alpha \cdot P(D|H_i) \cdot P(H_i)$

posterior likelihood prior

Teljes Bayes tanulás: P(H_i | D) kiszámítása az összes H_i-re.
A legtöbb esetben kezelhetetlen, de nincs jobb módja az optimalis predikció készítésnek.

Teljes bayesi tanulás közelítése

$$P(X) = \sum_i P(X \mid H_i) P(H_i)$$

$$P(X \mid D) = \sum_i P(X \mid D, H_i) P(H_i \mid D) = \sum_i P(X \mid H_i) P(H_i \mid D)$$

A legelterjedtebb közelítés - a **legvalószínűbb hipotézis:** max P(H_i | D)

P(H_{MAP} | D) ≈ 1, P($H_{más}$ | D) ≈ 0

Maximum a posteriori (**MAP**) hipotézis melletti predikció:

$$P(X \mid D) \approx P(X \mid H_{MAP})$$

A probléma a H_{MAP} megtalálása.

MAP hipotézis:

$$\max\left(P(H_i \mid D) = \frac{P(D \mid H_i) P(H_i)}{P(D)}\right)$$

MAP és ML tanulás

Maximálizálás: ‚probléma' az ***a priori*** valószínűség

$$P(H_i|D) = \frac{P(D|H_i) \cdot P(H_i)}{P(D)}$$

- *a priori* valószínűségeket rendeljünk az egyes hipotézisekhez, összegük 1
- bizonyos esetekben: **egyenletes** *a priori* eloszlás
  - tudás hiánya, de néha jobb nem felhasználni korábbi eredményeket

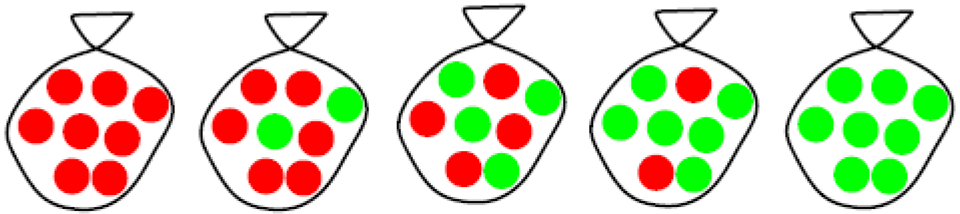
$$\max_i \mathrm{P}(H_i \mid D) = \max_i \mathrm{P}(D \mid H_i)$$

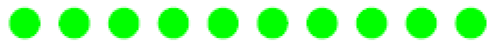
maximum likelihood (ML) hipotézis: $\boldsymbol{H_{ML}}$

Bayesian Model Averaging example

Suppose there are five kinds of bags of candies:

10% are h_1: 100% cherry candies
20% are h_2: 75% cherry candies + 25% lime candies
40% are h_3: 50% cherry candies + 50% lime candies
20% are h_4: 25% cherry candies + 75% lime candies
10% are h_5: 100% lime candies

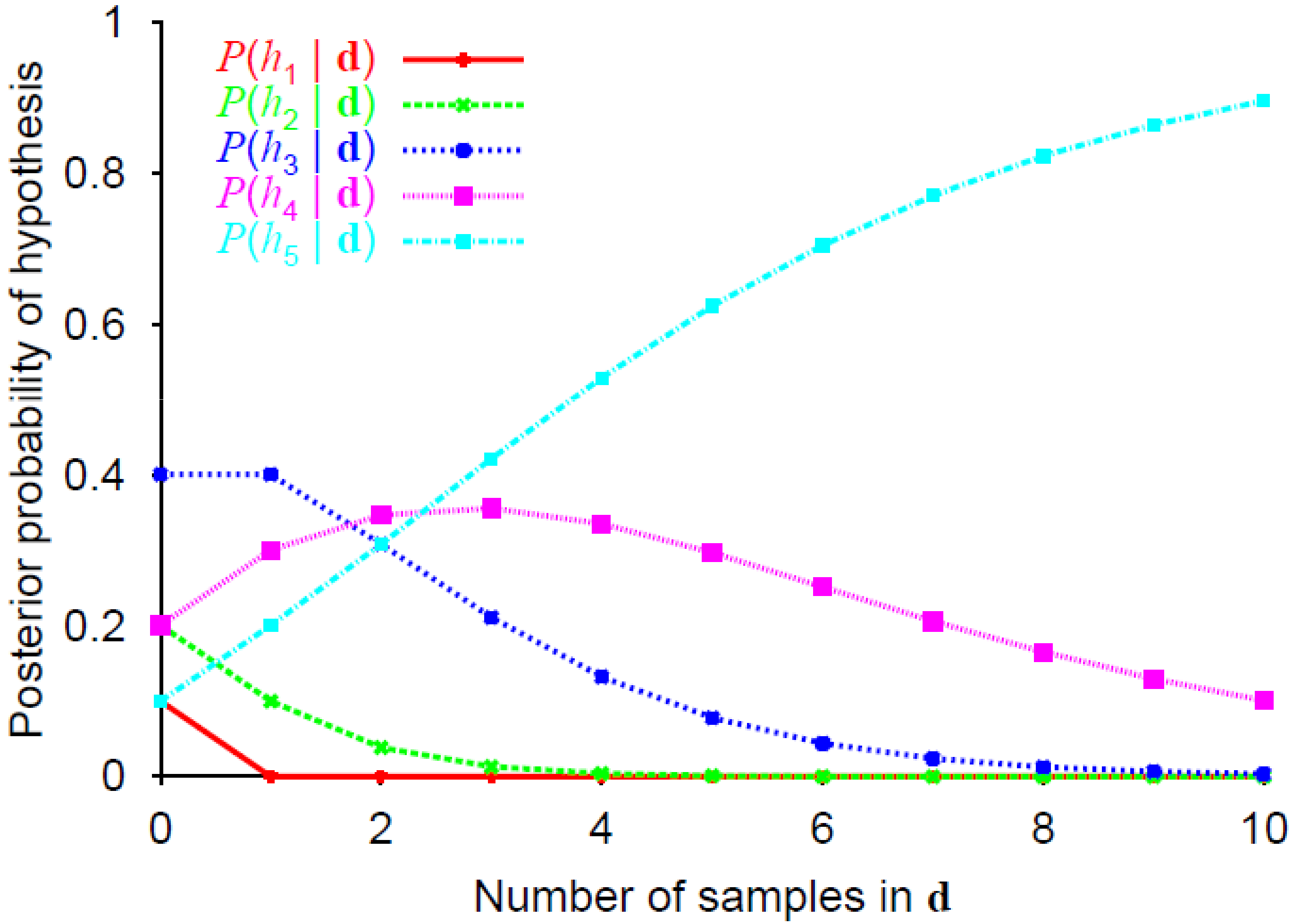


Then we observe candies drawn from some bag: 

What kind of bag is it? What flavour will the next candy be?

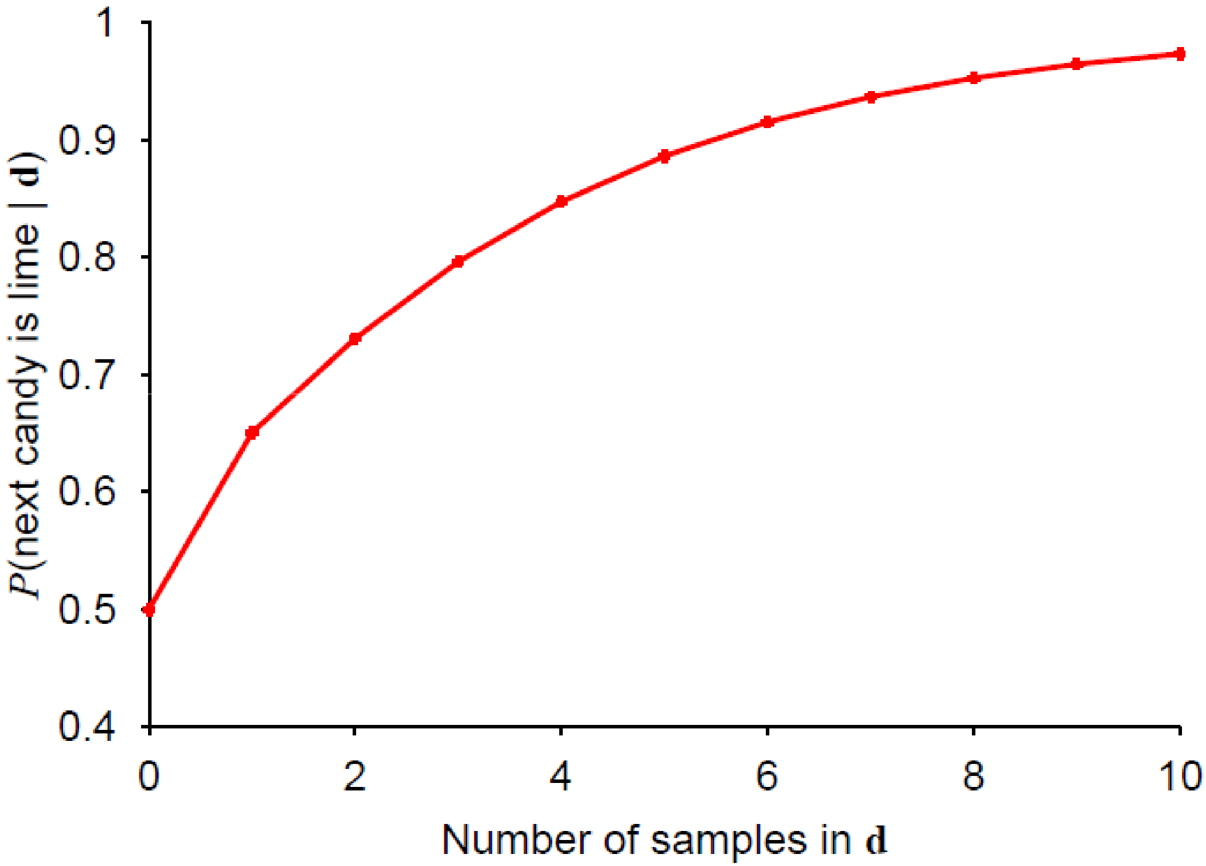
Russel&Norvig:Artificial intelligence

Az egyes modellek predikciójának a posteriori valószínűsége



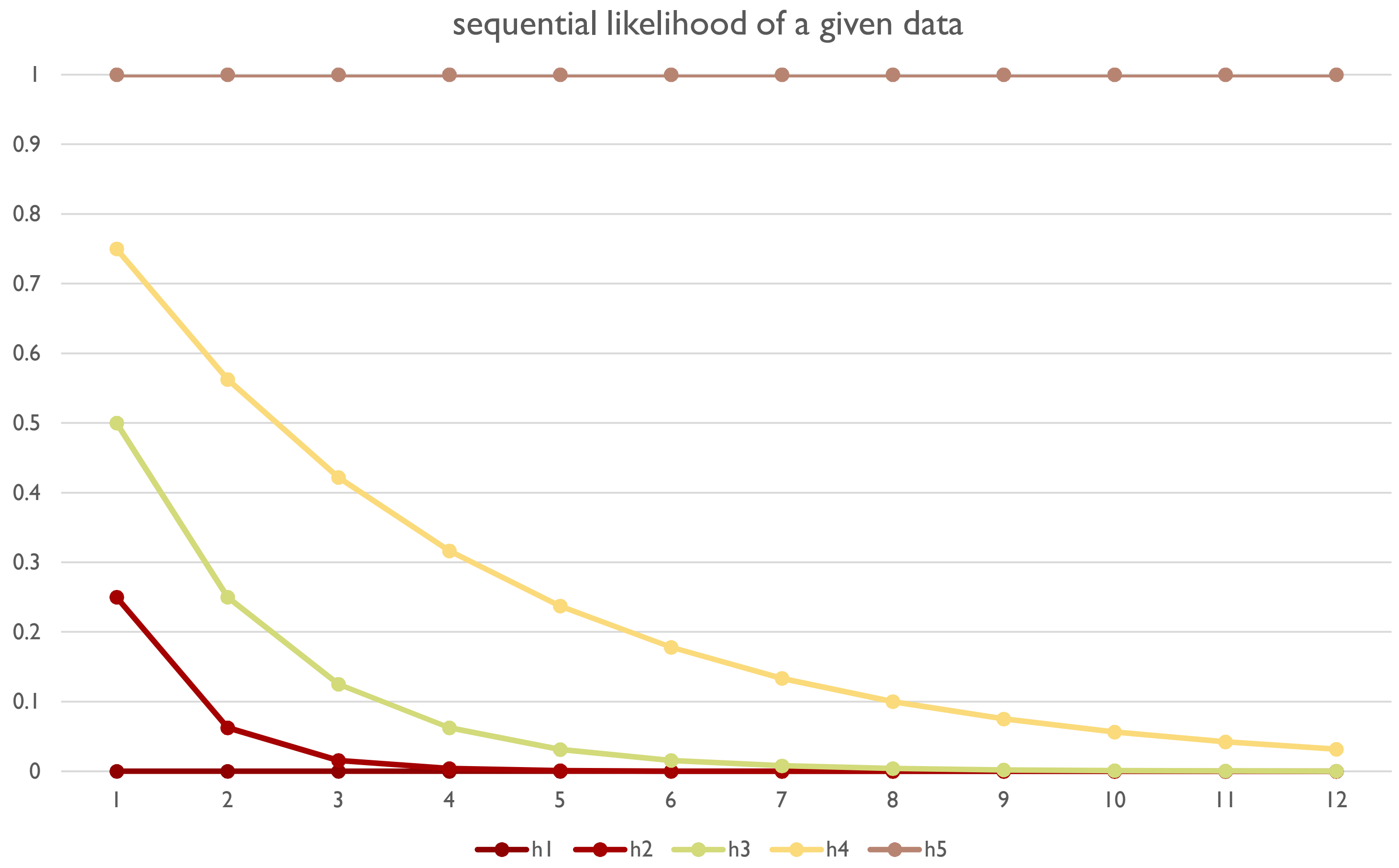
Russel&Norvig: Artificial intelligence

A bayesi módon átlagolt modell predikciójának *a posteriori* valószínűsége



Russel&Norvig: Artificial intelligence

Maximum likelood model selection



ML paramétertanulás

A cukorkás példa esetében

$$P(D|H_\theta) = \prod_{j=1}^{N} P(d_j|H_\theta) = \theta^c \cdot (1-\theta)^l$$

- Sokszor előnyös a log-likelihooddal számolni

$$L(D|H_\theta) = \log P(D|H_\theta)$$

$$= \sum_{j=1}^{N} P(d_j|H_\theta) = c \cdot \log\theta + l \cdot \log(1-\theta)$$

- Maximum számítás: deriválással

$$\frac{dL(D|H_\theta)}{d\theta} = \frac{c}{\theta} - \frac{l}{1-\theta} = 0 \rightarrow \theta = \frac{c}{c+l} = \frac{c}{N}$$

Struktúratanulás

- Irányított aciklikus gráfok n csomópontra: DAG(n)
- Csomópontok/változók sorrendezése: $\prec$
- Szülői halmazok száma: Π
- Kardinalitás (számosság):

$$f(n)=\sum_{i=1}^{n}(-1)^{i+1}2^{i(n-1)}f(n-i)\ with\ f(0)=1$$

n	$\lvert DAG(n)\rvert$	$\lvert G_{\prec}\rvert$	$\lvert G_{\prec}^{\lvert\pi\rvert\leq 4}\rvert$	$\lvert G_{\prec}^{\lvert\pi\rvert\leq 2}\rvert$	$\lvert\prec\rvert$	$\lvert\pi^{\prec}\rvert$	$\lvert\lvert\pi^{\prec}\rvert\leq 4\rvert$	$\lvert\lvert\pi^{\prec}\rvert\leq 2\rvert$
5	2.9e+004	1e+003	1e+003	6.2e+002	1.2e+002	30	30	24
6	3.8e+006	3.3e+004	3.2e+004	9.9e+003	7.2e+002	62	61	40
7	1.1e+009	2.1e+006	1.8e+006	2.2e+005	5e+003	1.3e+002	1.2e+002	62
8	7.8e+011	2.7e+008	1.8e+008	6.3e+006	4e+004	2.5e+002	2.2e+002	91
9	1.2e+015	6.9e+010	2.9e+010	2.3e+008	3.6e+005	5.1e+002	3.8e+002	1.3e+002
10	4.2e+018	3.5e+013	7.5e+012	1.1e+010	3.6e+006	1e+003	6.4e+002	1.7e+002
15	2.4e+041	4.1e+031	2.1e+027	3.1e+019	1.3e+012	3.3e+004	4.9e+003	5.7e+002
35	2.1e+213	1.3e+179	1.8e+109	8.5e+068	1e+040	3.4e+010	3.8e+005	7.2e+003

Bayes-háló struktúratanulása NP-nehéz

- **Tétel:** Legyen *V* a változók egy halmaza, melynek együttes valószínűségeloszlása p(*V*). Tegyük fel, hogy létezik egy orákulum, amely *O(1)* időben megválaszolja, hogy egy függetlenségi állítás igaz p-ben.
- Legyen $0 < k \leq |V|$ és $s = \frac{1}{2}n(n-1) - \frac{1}{2}k(k-1)$
- Ekkor annak eldöntése az orákulum segítségével, hogy létezik-e vagy sem egy olyan nem minimális Bayes-háló, ami reprezentálja *p*-t kisebb vagy egyenlő, mint *s* éllel, NP-nehéz

= Megfelelő Bayes-háló megtalálása a megfigyelési adatokhoz NP-nehéz

Bayes-háló struktúratanulása NP-nehéz

- **Tétel:** Legyen V a változók egy halmaza D_N a teljes adathalmaz $S(G,D_N)$ egy pontszámfüggvény (scoring function) és c egy valós szám.
- Ekkor annak eldöntése, hogy létezik-e egy V változók felett definiált G_0 Bayes-háló, amelynél minden csomópont legfeljebb $1<k$ szülővel rendelkezik, úgy hogy $c \leq S(G_0, D_N)$, az NP-nehéz

= A legjobb illeszkedési pontszámú Bayes-háló megtalálása NP-nehéz

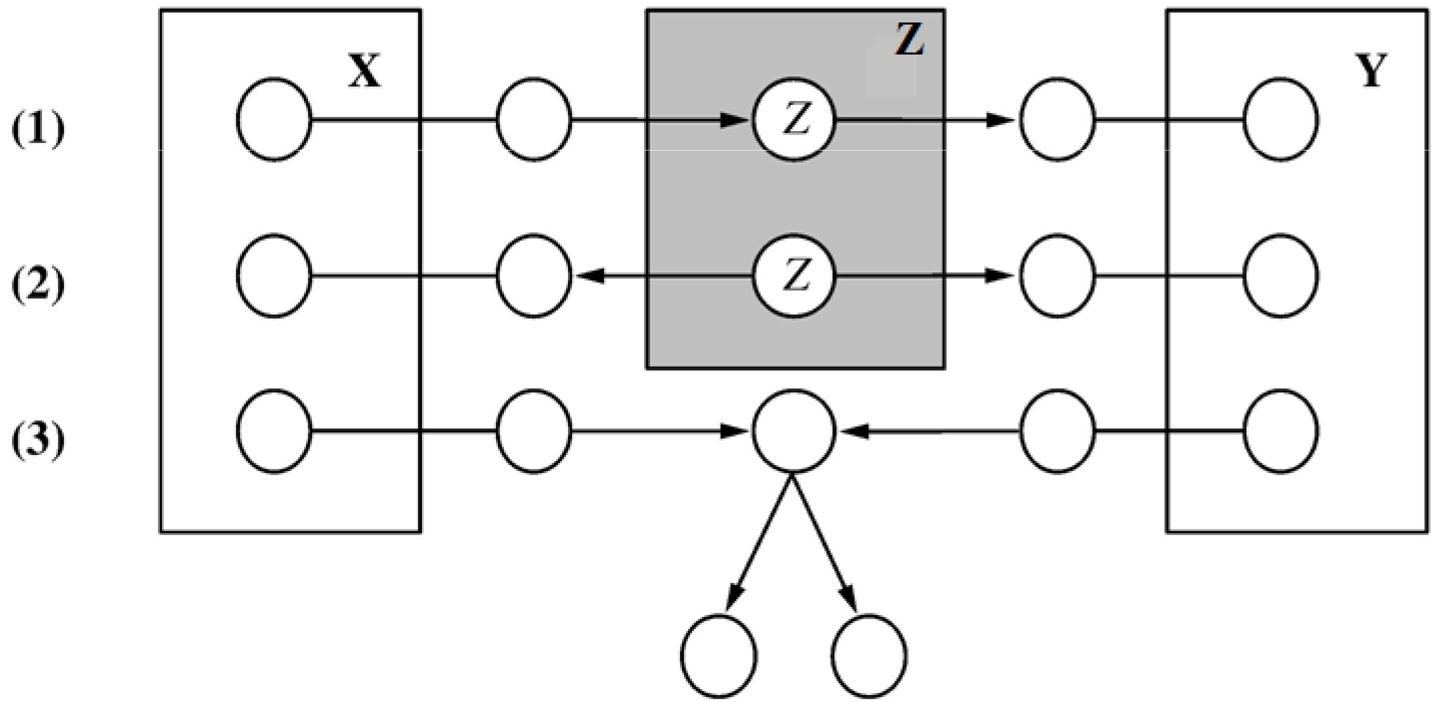
Bayes-háló - függőségek

- **Bayes-háló =** feltételes függetlenségek (függőségek) térképe

- A kérdés: egy adott struktúrából milyen feltételes függetlenségeket/függőségeket lehet kiolvasni?
- A válasz: **d-szeparációs szabályok** szerint
  - Directed separation ~ d-elválasztás

Bayes-háló: D-szeparáció

- Jelölje $I_G(X,Y|\mathbf{Z})$ hogy *X* d-szeparált *Y*-tól feltéve ***Z***-t a *G* irányított gráfban.



D-szeparáció – globális Markov-feltétel

Definition 7 *A distribution $P(X_1, \ldots, X_n)$ obeys the global Markov condition w.r.t. DAG G, if*

$$\forall\, X, Y, Z \subseteq U \; (X \perp\!\!\!\perp Y|Z)_G \Rightarrow (X \perp\!\!\!\perp Y|Z)_P, \tag{9}$$

where $(X \perp\!\!\!\perp Y|Z)_G$ denotes that X and Y are d-separated by Z, that is if every path p between a node in X and a node in Y is blocked by Z as follows

1. *either path p contains a node n in Z with non-converging arrows (i.e. $\rightarrow n \rightarrow$ or $\leftarrow n \rightarrow$),*
2. *or path p contains a node n not in Z with converging arrows (i.e. $\rightarrow n \leftarrow$) and none of its descendants of n is in Z.*

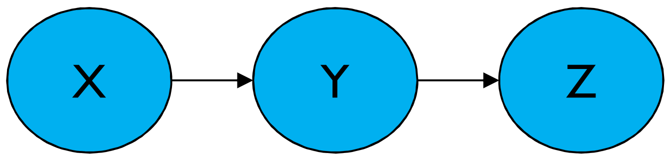
Feltételes függetlenségi modell

- A *P* eloszlás *M* függetlenségi térképe (modellje) az érvényes feltételes függetlenségeket jelölő tripletek halmaza:

$$M_P=\{I_{P,1}(X_1,Y_1|\mathbf{Z}_1),\ldots,\ I_{P,k}(X_k,Y_k|\mathbf{Z}_k)\}$$

- Ha *P(X,Y,Z)* egy Markov-lánc, akkor

$$\mathrm{M_P}=\{\mathrm{D(X,Y),\ D(Y,Z),\ I(X,Z|Y),\ és\ D(X,Z)}^*\},$$



- ahol D(…) = ¬ I(..)

*normális esetben D(X,Z), kivételesen lehet I(X,Z)

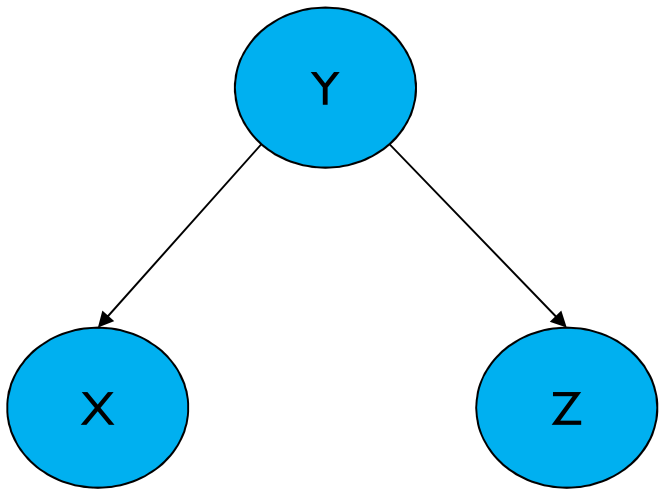
Feltételes függetlenségi modell - NBN

- Ha *P(X,Y,Z)* egy naiv Bayes-háló (NBN), akkor

  M_P={D(X,Y), D(Y,Z), I(X,Z|Y), és D(X,Z)*},

- ahol D(…) = ┐I(..)

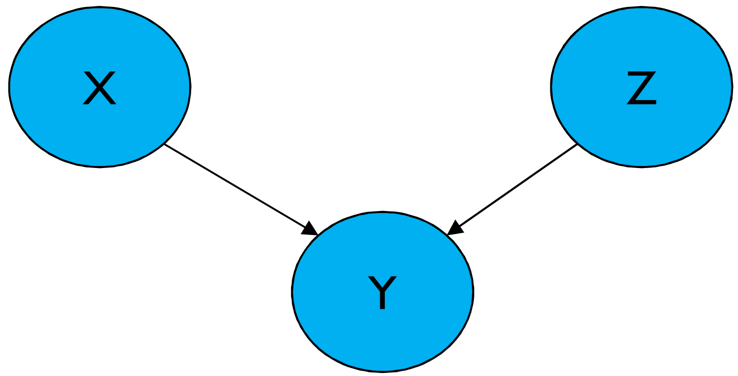
*normális esetben D(X,Z), kivételesen lehet I(X,Z)



Feltételes függetlenségi modell – V-struktúra

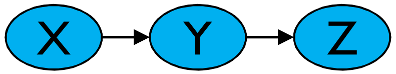
- Ha *P(X,Y,Z)* egy V-struktúra, akkor

$$M_P = \{D(X,Y),\ D(Y,Z),\ I(X,Z),\ \text{és}\ D(X,Z|Y)\},$$

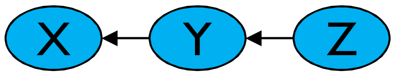


Feltételes függőségeket/függetlenségeket kódoló tripletek

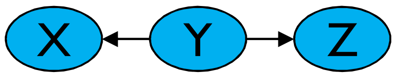
$p(X), p(Y|X), p(Z|Y)$



$p(X|Y), p(Y|Z), p(Z)$

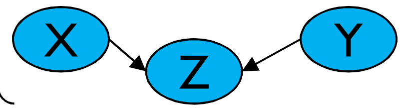


$p(X|Y), p(Y), p(Z|Y)$



"tranzitív" M ≠ „intranzitív" M

$p(X), p(Z|X,Y), p(Y)$



„V-struktúra"

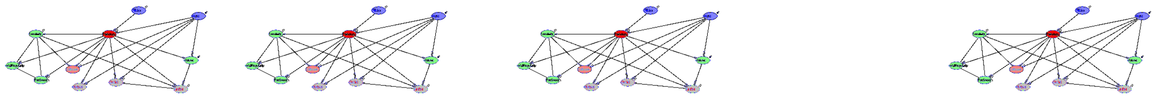
$M_P=\{D(X,Z), D(Z,Y), D(X,Y), I(X,Y|Z)\}$

Gyakran: a jelenlegi állapot (tudás) ismerete a jövőbeli állapotokat feltételesen függetlenné teszi a korábbiaktól

Oksági modellek megfigyelési ekvivalenciája

Oksági modellek:

J.Pearl:
~„3D objects”



Passzív megfigyelések:

$P(X_1,\ldots, X_n)$

$M_P=\{I_{P,1}(X_1;Y_1|Z_1),\ldots, I_{P,K}(X_K;Y_K|Z_K)\}$

„2D projection”

- Különböző oksági modelleknek lehet ugyanaz a függetlenségi térképe
- Jellemzően az oksági modelleket nem lehet tökéletesen megtanulni/meghatározni kizárólag passzív megfigyelési adatokból
- Ezek egy része **megfigyelés-ekvivalens**

Markov-lánc, mint oksági modell

Oksági modell:

$X_1 \rightarrow X_2 \rightarrow X_3 \rightarrow X_4$ $\qquad$ $X_1 \leftarrow X_2 \leftarrow X_3 \leftarrow X_4$

$P(X_1,\ldots)$

$M_P=\{I(X_{i+1},X_{i-1}|X_i)\}$

„elsőrendű Markov-tulajdonság”

Időbeliség meghatározható?

Megfigyelési ekvivalencia – V-struktúrák

Definition 11 *Two DAGs G_1, G_2 are* observationally equivalent *, if they imply the same set of independence relations (i.e. $(X \perp\!\!\!\perp Y|Z)_{G_1}) \Leftrightarrow (X \perp\!\!\!\perp Y|Z)_{G_2}$).*

The implied equivalence classes may contain $n!$ number of DAGs (e.g. all the full networks representing no independencies) or just 1.

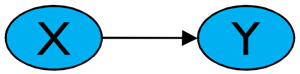
Theorem 2 *Two DAGs G_1, G_2 are* observationally equivalent *, iff they have the same skeleton (i.e. the same edges without directions) and the same set of* v-structures *(i.e. two converging arrows without an arrow between their tails).*

Definition 12 *The* essential graph *representing observationally equivalent DAGs is a partially oriented DAG (PDAG), that represents the identically oriented edges called* compelled edges *of the observationally equivalent DAGs (i.e. in the equivalence class), such a way that in the common skeleton only the compelled edges are directed (the others are undirected representing inconclusiveness).*

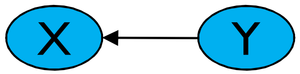
Reichenbach közös ok elve

Reichenbach's Common Cause Principle:

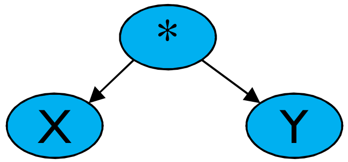
A korreláció *X* és *Y* esemény között azt jelenti, hogy vagy *X* okozza *Y-t*, vagy *Y* okozza *X-et*, vagy *X* és *Y* –nak közös oka van.



X okozza Y-t

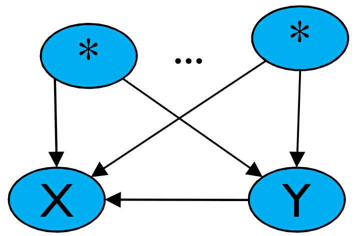


Y okozza X-et



Van egy közös ok (pure confounding)

...

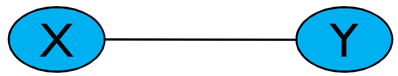


Egy ok-okozati kapcsolatot „zavarnak" más hatások

Passzív megfigyelésekből:

P(X,Y)

$M_P=\{D(X;Y)\}$



„X és Y asszociált"

Kényszeralapú Bayes-háló tanulás

The Inductive Causation algorithm (assuming a stable distribution P):

1. *Skeleton:* Construct an undirected graph (skeleton), such that variables $X, Y \in \mathbf{V}$ are connected with an edge iff $\forall S (X \perp\!\!\!\perp Y | S)_P$, where $S \subseteq \mathbf{V} \setminus \{X, Y\}$.
2. *v-structures:* Orient $X \rightarrow Z \leftarrow Y$ iff X, Y are nonadjacent, Z is a common neighbour and $\neg\exists S$ that $(X \perp\!\!\!\perp Y | S)_P$, where $S \subseteq \mathbf{V} \setminus \{X, Y\}$ and $Z \in S$.
3. *propagation:* Orient undirected edges without creating new v-structures and directed cycle.

Theorem

The following four rules are necessary and sufficient.

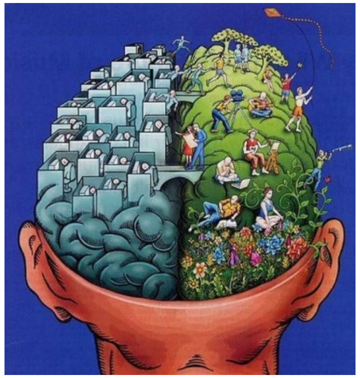
R_1 $\textit{if } (a \not\rightarrow c) \wedge (a \rightarrow b) \wedge (b - c), \textit{ then } b \rightarrow c$

R_2 $\textit{if } (a \rightarrow c \rightarrow b) \wedge (a - b), \textit{ then } a \rightarrow b$

R_3 $\textit{if } (a - b) \wedge (a - c \rightarrow b) \wedge (a - d \rightarrow b) \wedge (c \not\rightarrow d), \textit{ then } a \rightarrow b$

R_4 $\textit{if } (a - b) \wedge (a - c \rightarrow d) \wedge (c \rightarrow d \rightarrow b) \wedge (c \not\rightarrow b) \wedge (a - d), \textit{ then } a \rightarrow b$

Budapesti Műszaki és Gazdaságtudományi Egyetem
Méréstechnika és Információs rendszerek Tanszék



Racionalitás, hasznosság, döntés
Markov döntési folyamat

Előadó: Dr. Hullám Gábor

Előadás anyaga: Dr. Dobrowiecki Tadeusz

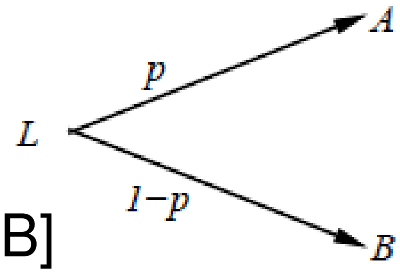
MŰEGYETEM 1782

Preferenciák

Egy ágens választásai
A, B, … determinisztikus tételek,
ill. bizonytalan kimenetelű sorsjátékok

$A > B$: A preferált B-hez képest
$A \sim B$: nincs preferencia A és B között
$A \geq B$: B nem preferált A-val szemben

Sorsjáték:
$L = [p, A; (1\text{-}p), B]$



Sorrendezhetőség

$(A > B) \vee (B > A) \vee (A \sim B)$

Tranzitivitás

$(B > A) \wedge (A > C) \rightarrow (B > C)$

Folytonosság

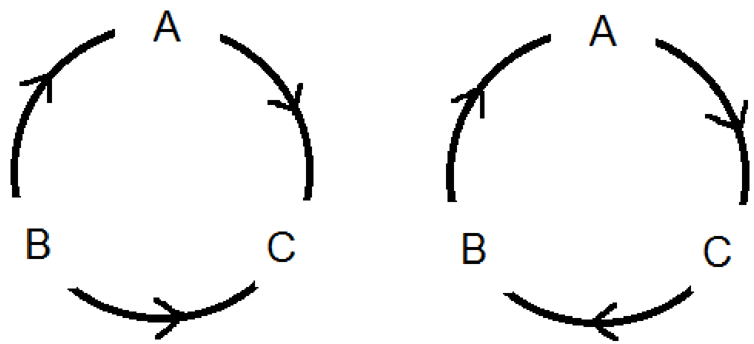
$A > B > C \rightarrow \exists p.\ [p, A; 1\text{-}p, C] \sim B$

Helyettesíthetőség

$A \sim B \rightarrow [p, A; 1\text{-}p, C] \sim [p, B; 1\text{-}p, C]$

Monotonitás

$A > B \rightarrow (p \geq q \leftrightarrow [p, A; 1\text{-}p, B] \geq [q, A; 1\text{-}q, B])$



Várható hasznosság maximalizálása

A korlátokat teljesítő preferenciákhoz létezik olyan valós értékű U(x) függvény, hogy (Ramsey, 1931, Neumann és Morgenstern, 1944):

$$U(A) \geq U(B) \leftrightarrow A \geq B$$

$$U([p_1, S_1; \ldots ; p_n, S_n] = \Sigma_i\, p_i\, U(S_i)$$

$$EU(A \mid E) = \sum_k P(Eredmény_k(A) \mid Tesz(A), E)\, U(Eredmény_k(A))$$

azt maximáló cselekvés megválasztása

Hasznosságok modellezése

Egy A állapot $\leftrightarrow$ standárt sorsolás:

a lehető legjobb díj - u_{max} p valószínűséggel

a lehető legnagyobb katasztrófa - u_{min} 1-p valószínűséggel

p módosítása, amíg: $A \sim L_p$

Hasznossági skálák

Normált: $u_{max} = 1$, $u_{min} = 0$

Mikromort: haláleséły/1000000, kb. 50 USD (2009)

pl. Mt.Everest: 39427 mm/ megmászás

....

A pénz hasznossága és az emberi (ir)racionalitás

Nem szabályos hasznosság! Ha L egy sorsjáték, aminek várható pénzbeli nyeresége EMV(L), akkor általában U(L) < U(EMV(L))

Hasznossági görbe: milyen p valószínűség esetén indifferens az x díj és a [p, M; 1-p, 0] sorsjáték értéke között, nagyon nagy M-re?

Tegyük fel, hogy nyert egy TV játékban. A műsorvezető most választásra kéri fel: elviheti az 1 milliós díjat, vagy felteheti azt egy pénzfeldobásos hazárdjátékon. Ha fej, nem kap semmit, ha írás, akkor kap 3 milliót. Ha hasonló a többi emberhez, akkor vonakodna játszani, és zsebre vágná a milliót. Ez irracionális volna?

1 millió < EMV(L) = 1.5 millió

De mi van, ha már van valami pénze (S_k)?

$U(S_k)$	5	5.0
$U(S_{k+1M})$	9	5.1
$U(S_{k+3M})$	11	5.3

$$EU(Elfogad) = \frac{1}{2}U(S_k) + \frac{1}{2}U(S_{k+3M})$$

$$EU(Elutasít) = U(S_{k+1M})$$

Grayson (1960): a pénz hasznossága majdnem teljesen arányos a mennyiségének logaritmusával (először Bernoulli, 1783).

A pénz hasznossága és az emberi (ir)racionalitás

Nyereségekre: (kockázatkerülő)

U(L) < U(EMV(L) biztos kifizetése)

Veszteségekre: (kockázatkereső)

U(L) > U(EMV(L) biztos kifizetése)

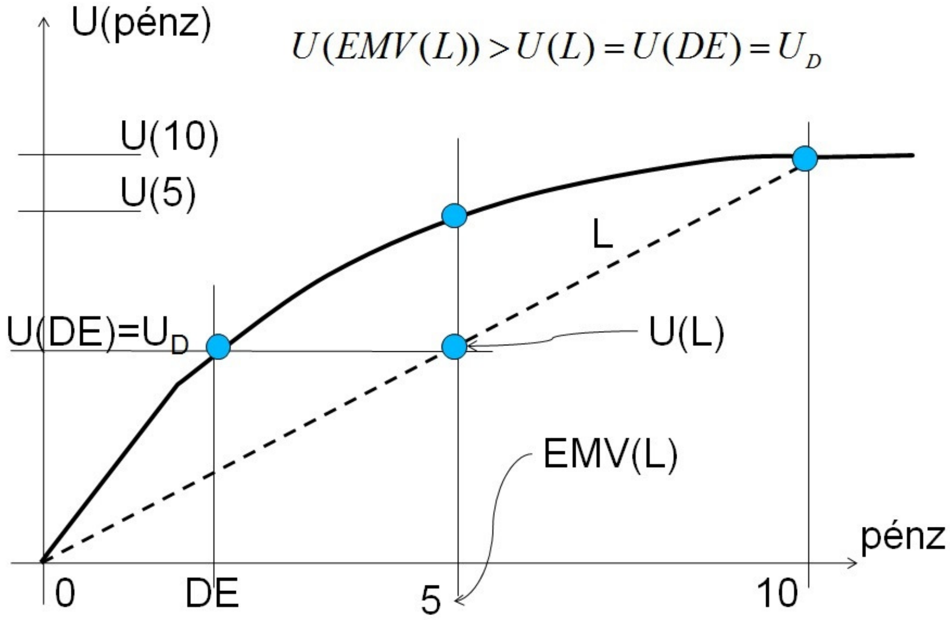
Kis értékek szakasza lineáris

- kockázat-semleges

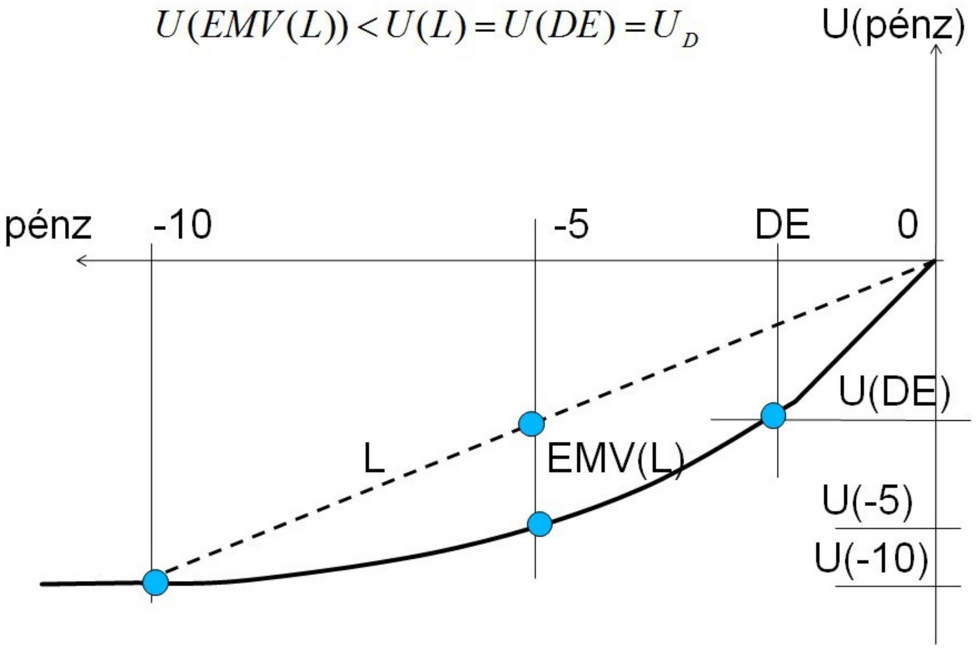
Sorsjáték determinisztikus ekvivalense

DE (játék helyett fogad el)

$U(EMV(L)) > U(L) = U(DE) = U_D$



$U(EMV(L)) < U(L) = U(DE) = U_D$



Többváltozós hasznosságfüggvények

U(Halálesetek, Zaj, Költség)?

$U(x_1, x_2, \ldots, x_n) = ?$ (1) teljes körű beazonosítás
(2) függetlenségek, kanonikus alakok

Additív értékfüggvény $U = k_1 U_1 + k_2 U_2 + k_3 U_3$

Pl. U(Zaj, Költség, Halálesetek) =
– Zaj[dB] × 10^4 – Költség[mFt] – Halálesetek[mikromort] × 10^{12}

Multiplikatív értékfüggvény

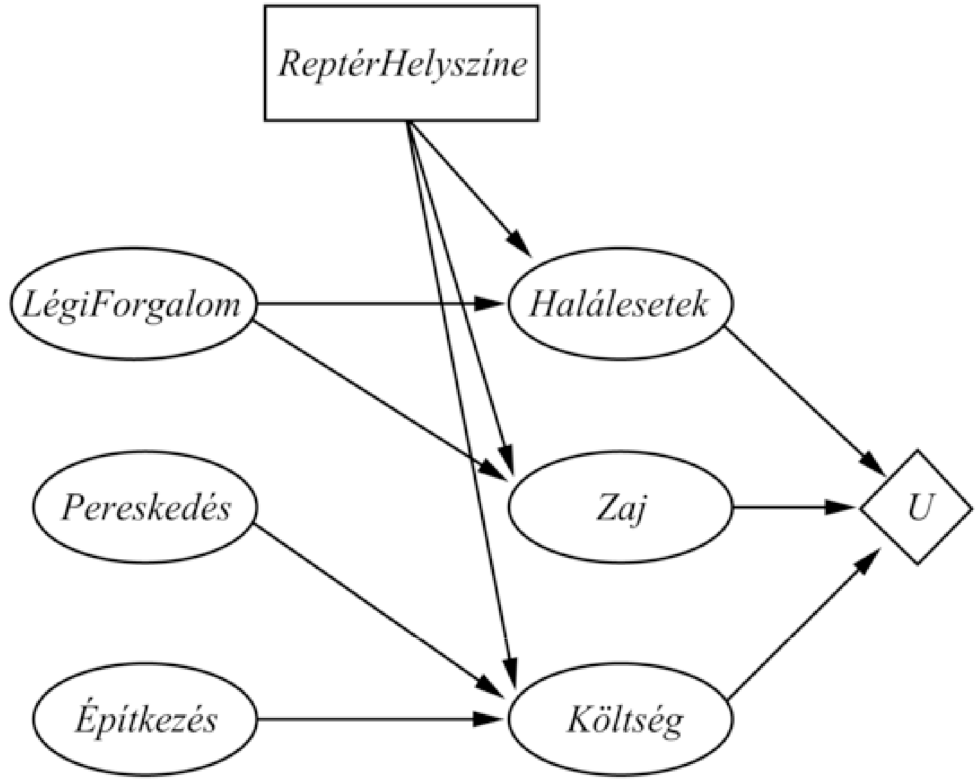
$$U = k_1 U_1 + k_2 U_2 + k_3 U_3 + k_1 k_2 U_1 U_2 + k_2 k_3 U_2 U_3 + k_3 k_1 U_3 U_1 + k_1 k_2 k_3 U_1 U_2 U_3$$

csak 3 paraméter

stb.

Döntési hálók

véletlen csomópontok
  FVT
döntési csomópontok
  döntési lehetőségek
hasznosság csomópontok
  hasznosságok leírása
  cselekvéshasznosság
  táblák



Következtetés:
- evidencia változók beállítása
- a döntési csomópont minden egyes értékére:
  - állítsuk be a döntési csomópontot erre az értékre
  - **számítsuk ki az a posteriori valószínűségeket a hasznosság-csomópont szüleire** (szabványos valószínűségi hálós következtetés)
  - számítsuk ki a cselekvések hasznosságát
- ? a legnagyobb hasznosságértékű cselekvés

Információ hasznossága

Legyen a meglévő evidencia E, az aktuális legjobb cselekvés α, melynek lehetséges kimenetelei $Eredmény_i$, az új lehetséges evidencia E_j.
A pillanatnyi legjobb cselekvés értéke:

$$EU(\alpha \mid E) = \max_A \sum_k P(Eredmény_k(A) \mid Tesz(A), E)\, U(Eredmény_k(A))$$

A pillanatnyi legjobb cselekvés értéke új evidencia után:

$$EU(\alpha_{E_j} \mid E, E_j) = \max_A \sum\nolimits_i U(Eredm_i(A)) P(Eredm_i(A) \mid Tesz(A), E, E_j)$$

A teljes információ értéke (TIÉ) (az előre még nem ismert új evidencia értékeire vett átlag):

$$TIÉ_E(E_j) = \left(\sum\nolimits_k P(E_j = e_{jk} \mid E) EU(\alpha_{e_{jk}} \mid E, E_j = e_{jk})\right) - EU(\alpha \mid E)$$

Racionális ágensek tranzitív preferenciáiról

Három ágens preferenciái:

(Ág1) Körte > Szőlő > Alma
(Ág2) Szőlő > Alma > Körte
(Ág3) Alma > Körte > Szőlő

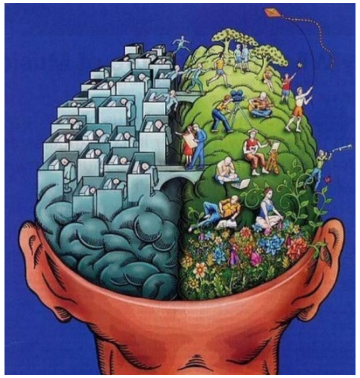
Mi a csoport véleménye, a csoport preferenciasora?

Legyen annak kifejezője a többségi választás (itt 2 az 1 ellen):

Körte > Szőlő
Szőlő > Alma
Alma > Körte

Egyenként racionális (tranzitív), együtt már nem?

Budapesti Műszaki és Gazdaságtudományi Egyetem
Méréstechnika és Információs rendszerek Tanszék



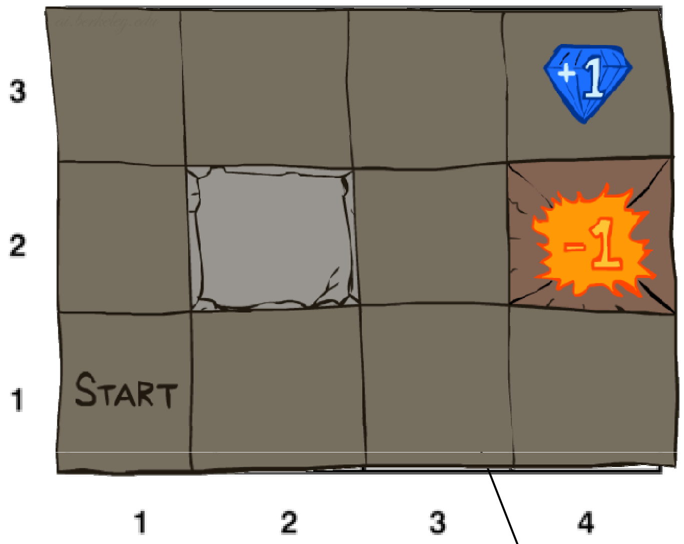
Markov döntési folyamat

Előadó: Dr. Hullám Gábor

Előadás anyaga: Dr. Dobrowiecki Tadeusz

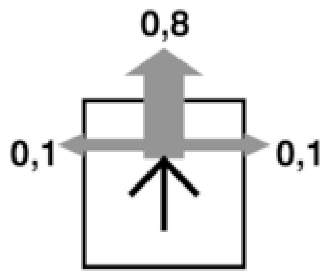
MŰEGYETEM 1782

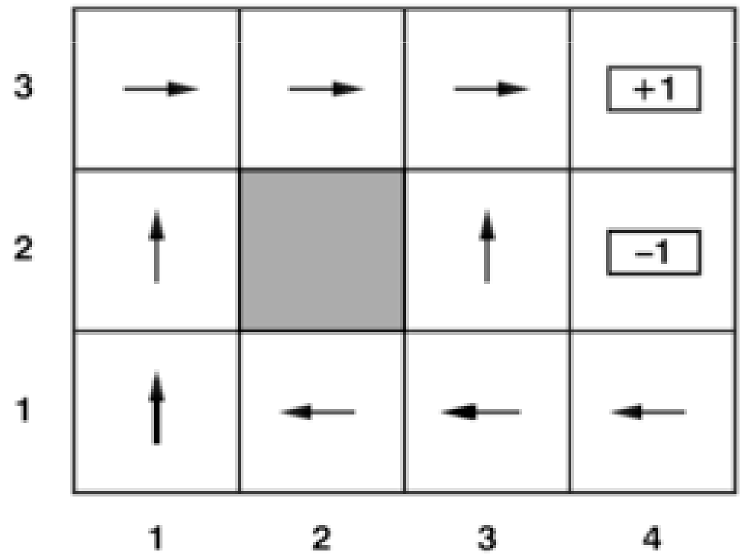
Szekvenciális döntési probléma



(a)

- 0.04





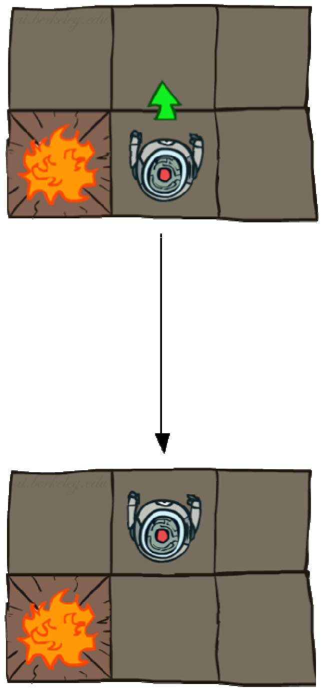
(b)

Fix út: fel, fel, jobbra, jobbra, jobbra?

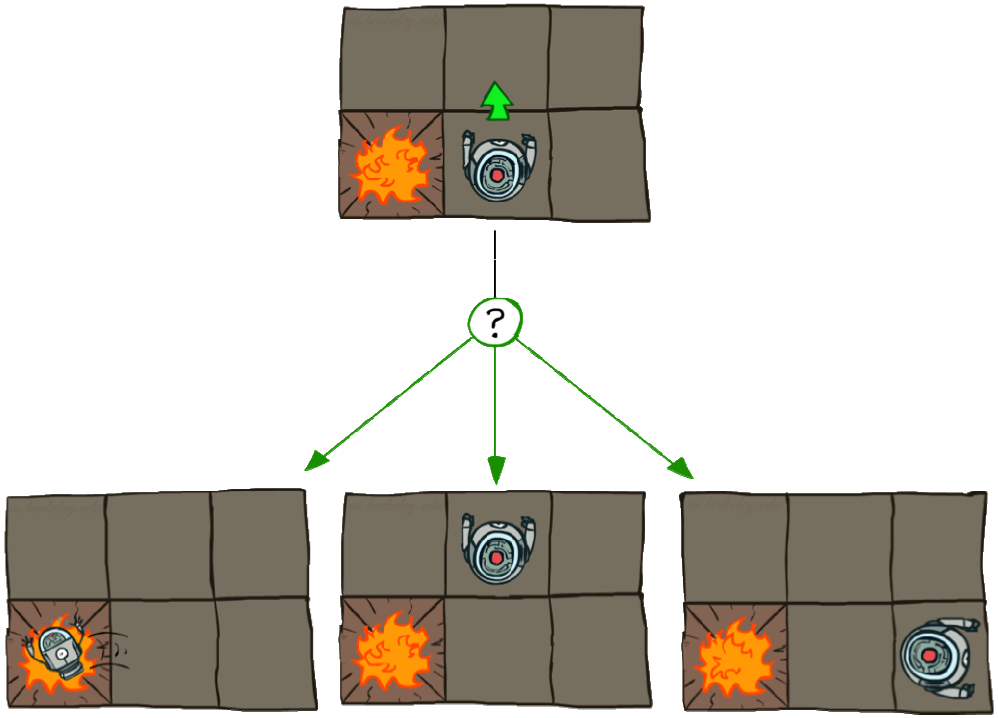
(optimális) Eljárásmód: $\pi(s) = a$

Grid World Cselekvések

Determinisztikus Grid World



Sztochasztikus Grid World



Szekvenciális döntési probléma

Markov döntési folyamat

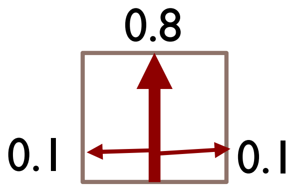
Kezdőállapot:	S0
Állapotátmenet-modell:	T(s, a, s′)
Jutalomfüggvény:	R(s), vagy R(s, a, s′)

Optimális eljárásmód = optimális mozgás, döntés cselekvés megválasztására, de nem elég egyszer, folyamatosan kell, amíg nincs vége (a problémának).

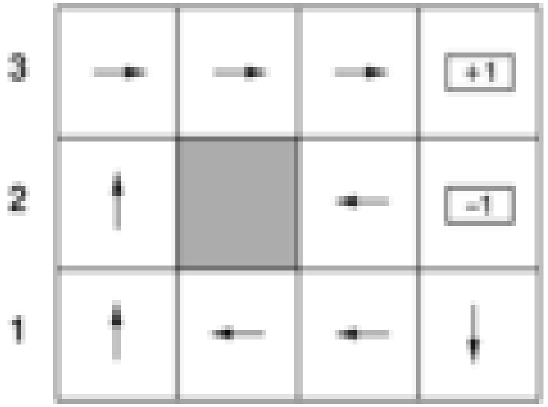
$$\pi(s) = a$$

$$\pi^*(s) = a$$

Szekvenciális döntési probléma

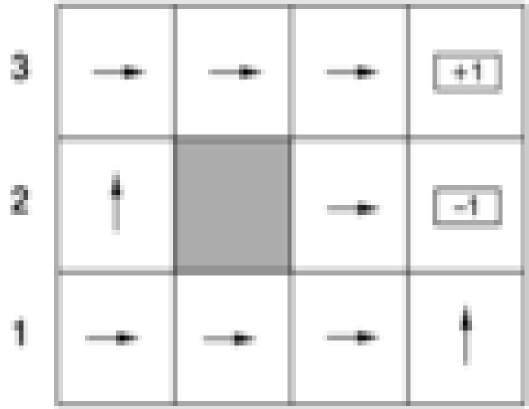


– 0,0221 < R(s) < 0



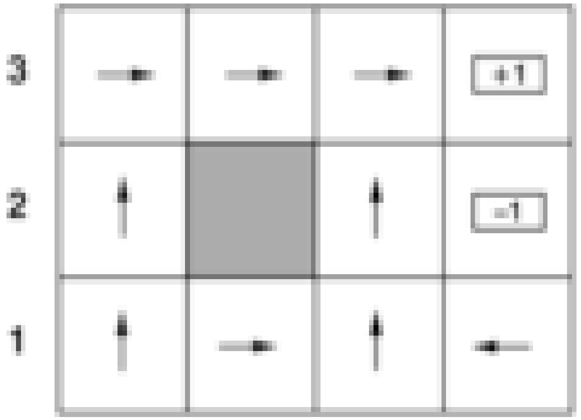
az élet csak kevéssé bánatos,
ne legyen kockázat!

R(s) ≤ –1,6284



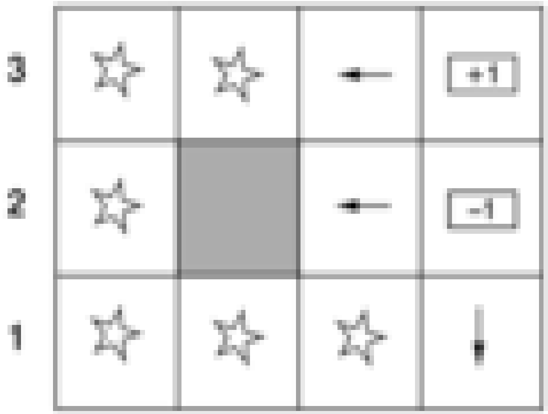
élet elviselhetetlen, ki!

– 0,4278 ≤ R(s) ≤ – 0,0850



élet kellemetlen; +1 állapot,
–1 kockázattal

R(s) > 0



élet kifejezetten élvezhető,
az ágens benn akar maradni

Szekvenciális döntési probléma

Optimalis szekvenciális döntési probléma

	optimális eljárásmód
végtelen horizont	stacionárius
véges horizont	nem-stacionárius

Többattribútumú hasznosságelmélet:
ágens preferenciái az állapotsorozatok között stacionáriusok

additív jutalmak $U_h([s_0, s_1, s_2, \ldots]) = R(s_0) + R(s_1) + R(s_2) + \ldots$

leszámított jutalmak

$$U_h([s_0, s_1, s_2, \ldots]) = R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \ldots$$

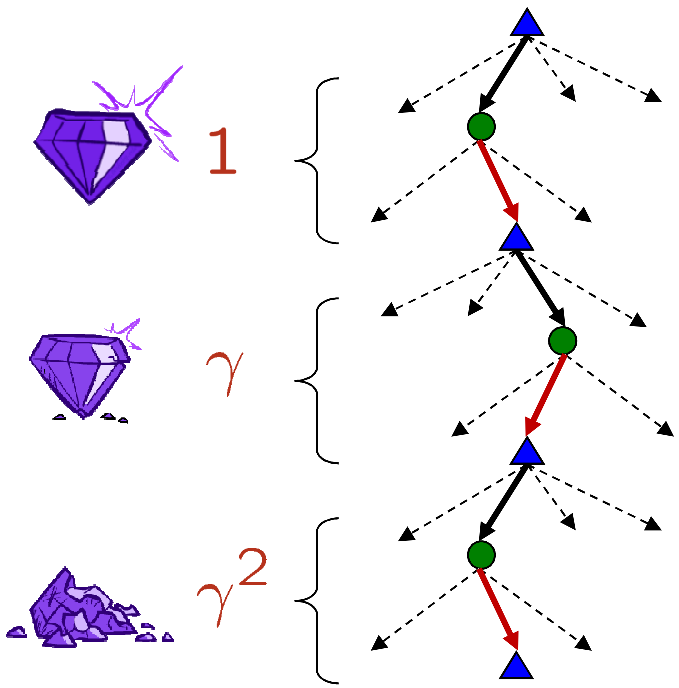
Leszámítolás

- Ésszerű a jutalmak maximalizálására törekedni
- Ésszerű egy jelenlegi jutalmat preferálni egy jövőbeli jutalom helyett
- Egy megoldás: a jutalmak értékei exponenciálisan csökkennek

1	γ	γ^2
Hasznosság most	Hasznosság a következő lépésben	Hasznosság 2 lépés múlva

Leszámítolás

- Ésszerű a jutalmak maximalizálására törekedni
- Ésszerű egy jelenlegi jutalmat preferálni egy jövőbeli jutalom helyett
- Egy megoldás: a jutalmak értékei exponenciálisan csökkennek



Szekvenciális döntési probléma

Leszámítolt jutalmak, egy végtelen sorozat hasznossága

$$U_h([s_0, s_1, s_2, \ldots]) = \Sigma_{t=0\ldots\infty} \gamma^t R(s_t) =< \Sigma_{t=0\ldots\infty} \gamma^t R_{\max} = R_{\max} / (1-\gamma)$$

Alternatív esetek:

1.) Ha van végállapot, és ha garantált, hogy az ágens végül bele kerül, akkor nincs szükség végtelen sorozatok összehasonlítására.
Egy eljárásmód, ami garantáltan végállapotba juttat, véges eljárásmód, $\gamma = 1$

2.) Időegységenkénti átlagjutalom

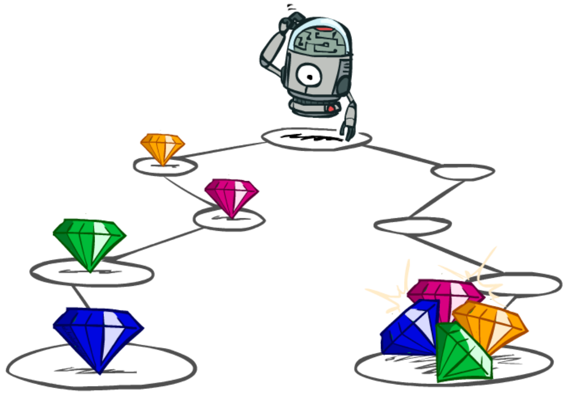
Szekvenciális döntési probléma

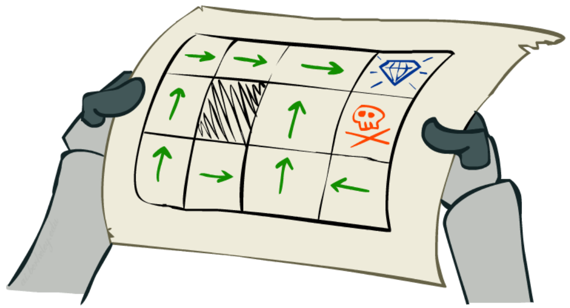
Leszámítolt jutalmak, egy végtelen sorozat hasznossága

$$U_h([s_0, s_1, s_2, \ldots]) = \Sigma_{t=0\ldots\infty} \gamma^t R(s_t) =< \Sigma_{t=0\ldots\infty} \gamma^t R_{\max} = R_{\max} / (1-\gamma)$$

Optimális eljárásmód

$$\pi^* = \arg\max_{\pi} E[\Sigma_{t=0\ldots\infty} \gamma^t R(s_t) \mid \pi]$$





Optimális eljárásmód meghatározása - Értékiteráció

Egy **állapot hasznossága –** a belőle kiinduló állapotsorozatok várható hasznossága

Az állapotsorozatok függnek a végrehajtott eljárásmódtól, így elsőként egy adott *π* eljárásmódra definiáljuk a hasznosságot:

$$U^{\pi}(s) = E\left[\sum_{t=0}^{\infty} \gamma^{t} R(s_t) \,|\, \pi, s_0 = s\right]$$

Optimális eljárásmód

$$\pi^{*}(s) = \arg\max_{a} \sum_{s'} T(s,a,s')\,U(s')$$

Optimális eljárásmód meghatározása - Értékiteráció

Az **állapot hasznossága** - az állapotban tartózkodás közvetlen jutalmának és a következő állapot várható leszámított hasznosságának az összege, feltéve, hogy az ágens az optimális cselekvést választja

$$U(s) = R(s) + \gamma \max_{a} \sum_{s'} T(s,a,s')U(s')$$

Bellman egyensúlyi **egyenlet**

Legyen $\gamma = 1$ és a nem végállapotoknál $R(s) = -0{,}04$

Nézzük meg a 4×3-as világ Bellman-egyenleteinek egyikét.

Az (1, 1) állapothoz tartozó egyenlet:

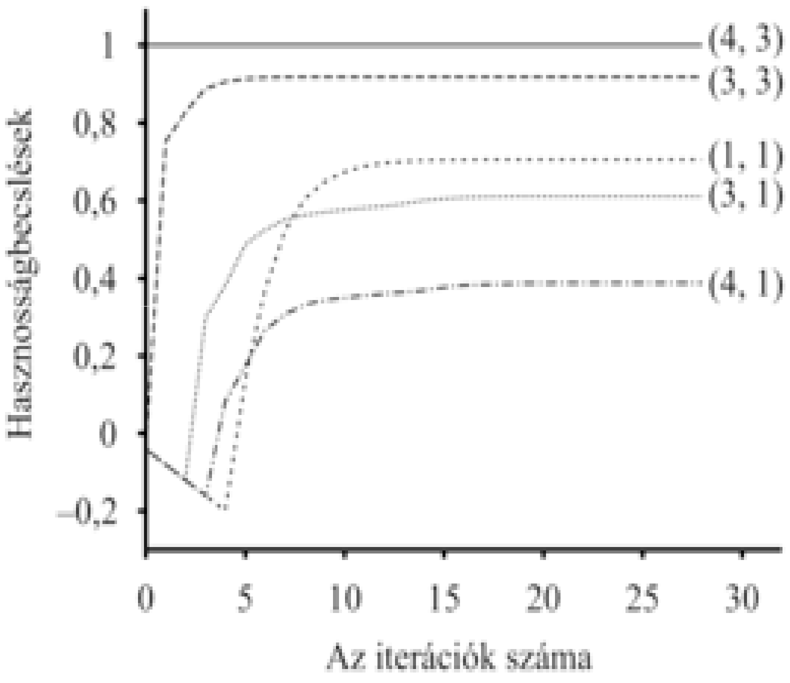
$$
\begin{aligned}
U(1, 1) = &-0{,}04 + \\
\gamma \max\{&0{,}8\, U(1, 2) + 0{,}1\, U(2, 1) + 0{,}1\, U(1, 1) && \text{(Fel)}, \\
&0{,}8\, U(2, 1) + 0{,}1\, U(1, 2) + 0{,}1\, U(1, 1) && \text{(Jobbra)}, \\
&0{,}9\, U(1, 1) + 0{,}1\, U(1, 2) && \text{(Balra)}, \\
&0{,}9\, U(1, 1) + 0{,}1\, U(2,1)\} && \text{(Le)}
\end{aligned}
$$

Sajnos nemlineáris – iteráció!

3	0,812	0,868	0,918	+ 1
2	0,762		0,660	–1
1	0,705	0,655	0,611	0,388
	1	2	3	4

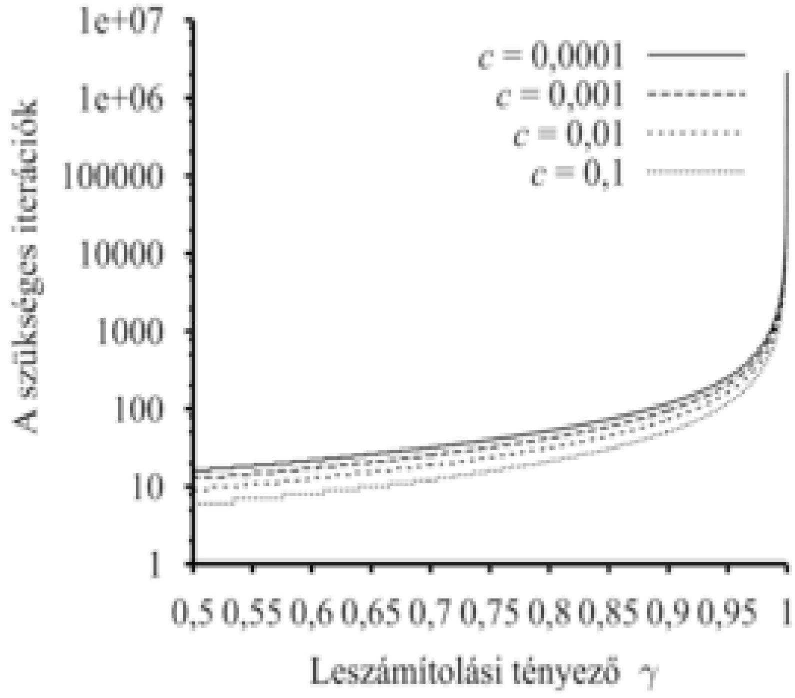
Bellman-frissítés:

$$U_{t+1}(s) = R(s) + \gamma \max_a \sum_{s'} T(s,a,s') U_t(s')$$



(a)

A hasznosságok fejlődése



(b)

A szükséges értékiterációk száma, hogy a hiba garantáltan legfeljebb $\varepsilon = c\ R_{max}$ legyen

Az értékiteráció konvergenciája – **kontrakció** $U_{i+1} = B\, U_i$

$$\|U\| = \max_s |U(s)|$$

$\|BU_i - U^*\| \leq \gamma\|U_i - U^*\|$, U^* az igazi: $B(U^*) = U^*$

Az összes állapot hasznossága korlátos $\pm R_{max}/(1-\gamma)$ értékkel
A maximális kezdeti hiba $\|U_0 - U^*\| \leq 2R_{max}/(1-\gamma)$

Ha $\|U_{i+1} - U_i\| < \varepsilon(1-\gamma)/\gamma$, akkor $\|U_{i+1} - U^*\| < \varepsilon$

Eljárásmód-iteráció

(optimális eljárásmódot is kaphatunk, ha a hasznosság becslése pontatlan - ha egy cselekvés egyértelműen jobb, akkor a releváns állapotok pontos hasznosságát nem szükséges precízen tudnunk)

$$U(s) = R(s) + \gamma \max_a \sum_{s'} T(s,a,s') U(s')$$

Eljárásmód-értékelés

Egy adott π_i eljárásmódnál számítsuk ki $U_i = U^{\pi_i}$-t, az egyes állapotok hasznosságát mintha π_i volna végrehajtva.

lineáris!

$$U_t(s) = R(s) + \gamma \sum_{s'} T(s,\pi_t(s),s') U_t(s')$$

$$U_{t+1}(s) \leftarrow R(s) + \gamma \sum_{s'} T(s,\pi_t(s),s') U_t(s')$$

Módosított eljárásmód-iteráció

Eljárásmód-javítás

for each s állapotra **in** S **do**

if $\max_a \sum_{s'} T(s,a,s')U[s'] > \sum_{s'} T(s,\pi[s],s')U[s']$ **then**

$\pi[s] \leftarrow \mathrm{argmax}_a \sum_{s'} T(s,a,s')U[s']$

Részlegesen megfigyelhető Markov döntési folyamat

Kezdőállapot: S0
Állapotátmenet-modell: T(s, a, s′)
Jutalomfüggvény: R(s), v. R(s, a, s′)
Megfigyelési modell,
az s állapotban az o megfigyelés érzékelésének a valószínűsége
O(s, o)

Hiedelmi állapot = *b*(*s*) = eloszlás állapotok felett

0,111	0,111	0,111	$\boxed{0{,}000}$
0,111		0,111	$\boxed{0{,}000}$
0,111	0,111	0,111	0,111

$$\left\langle \frac{1}{9},\frac{1}{9},\frac{1}{9},\frac{1}{9},\frac{1}{9},\frac{1}{9},\frac{1}{9},\frac{1}{9},\frac{1}{9},0,0 \right\rangle$$

$$b'(s') = \alpha O(s',o)\sum_{s} T(s,a,s')\,b(s)$$

(szűrés)

$$P(o \mid a,b) = \sum_{s'} P(o \mid a,s',b)\, P(s' \mid a,b)$$

$$= \sum_{s'} O(s',o) P(s' \mid a,b) = \sum_{s'} O(s',o) \sum_{s} T(s,a,s') b(s)$$

$$\tau(b,a,b') = P(b' \mid a,b) = \sum_{o} P(b' \mid o,a,b)\, P(o \mid a,b)$$

$$= \sum_{o} P(b' \mid o,a,b) \sum_{s'} O(s',o) \sum_{s} T(s,a,s') b(s)$$

$$\rho(b) = \sum_{s} b(s) R(s)$$

RMMDF megoldása a fizikai (véges) állapottérben redukálható egy MDF megoldására a hozzá tartozó hiedelmi állapot térben (val. eloszlások folytonos terében)

Szekvenciális döntési problémából indulunk

Markov döntési folyamat (MDF)

S0
T(s, a, s′)
R(s), v. R(s, a, s′)

$$U(s) = R(s) + \gamma \max_a \sum_{s'} T(s,a,s')U(s')$$

$$\pi(s) \leftarrow U(s)$$

$$U_{i+1}(s) \leftarrow R(s) + \gamma \max_a \sum_{s'} T(s,a,s')U_i(s') \qquad \pi(s) \leftarrow U_i(s)$$

$$\pi^*(s) = \arg\max_a \sum_{s'} T(s,a,s')U(s')$$ Optimális eljárásmód

Megerősítéses tanulás

Kezdőállapot:	ismert
Állapotátmenet-modell:	**nem ismert, legfeljebb a tapasztalat**
Jutalomfüggvény:	**nem ismert, legfeljebb ahogy jön**
Optimális eljárásmód	**megtanulni, mindennek ellenére?!**

Megerősítéses tanulás

Pl. sakkjáték: tanító nélkül is van visszacsatolás a játék végén:
nyert vagy **vesztett** = +/- **jutalom**, azaz **megerősítés**.

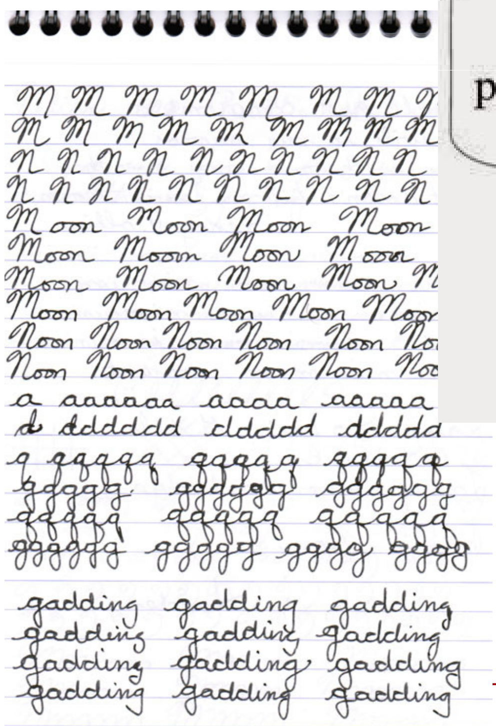
... a játék végén = a cselekvés szekvencia végén

Megerősítés: milyen gyakran? milyen erős? milyen értékű?

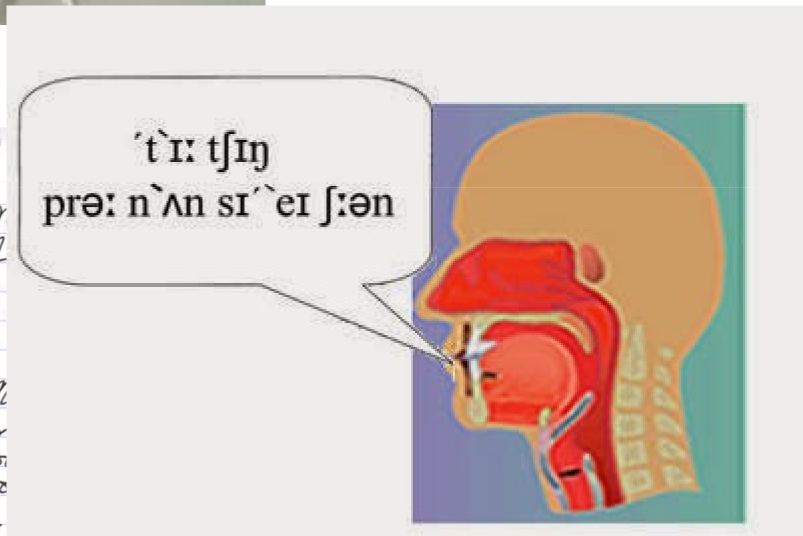
A feladat:

(ritka) jutalmakból megtanulni egy sikeres ágens-függvényt (optimális eljárásmód: melyik állapot, melyik cselekvés hasznos).

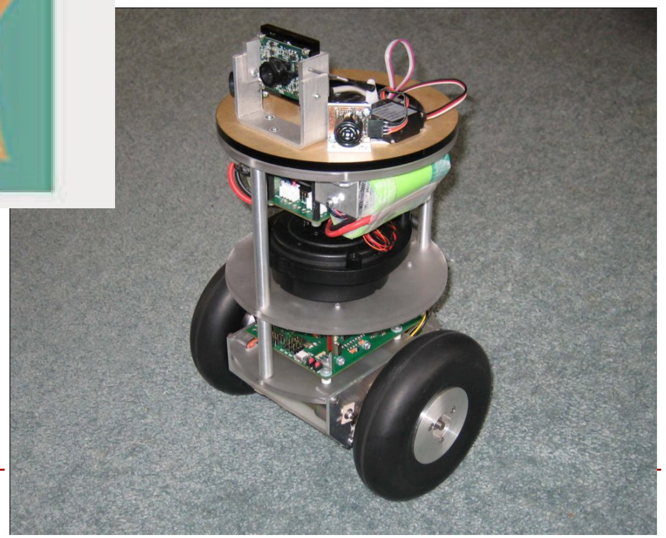
Nehéz: az információhiány miatt az ágens sohasem tudja, hogy **mik a jó lépések**, azt sem, **melyik jutalom melyik cselekvésből ered.**



Moon Moon Moon Moon
Noon Noon Noon Noon
a aaaaa aaaa aaaaa
d ddddd ddddd dddd
gadding gadding gadding



´t`ɪː tʃɪŋ
prəː n`ʌn sɪ´`eɪ ʃːən



Ágens tudása:

induláskor **tudja már** a környezetet és a cselekvéseinek hatását, vagy pedig még ezt is **meg kell tanulnia**.

Megerősítés:

csak a **végállapotban**, vagy menet közben **bármelyikben**.

Ágens:

passzív tanuló: figyeli a világ alakulását és tanul.

aktív tanuló: a megtanult információ birtokában cselekednie is kell.

felfedező …

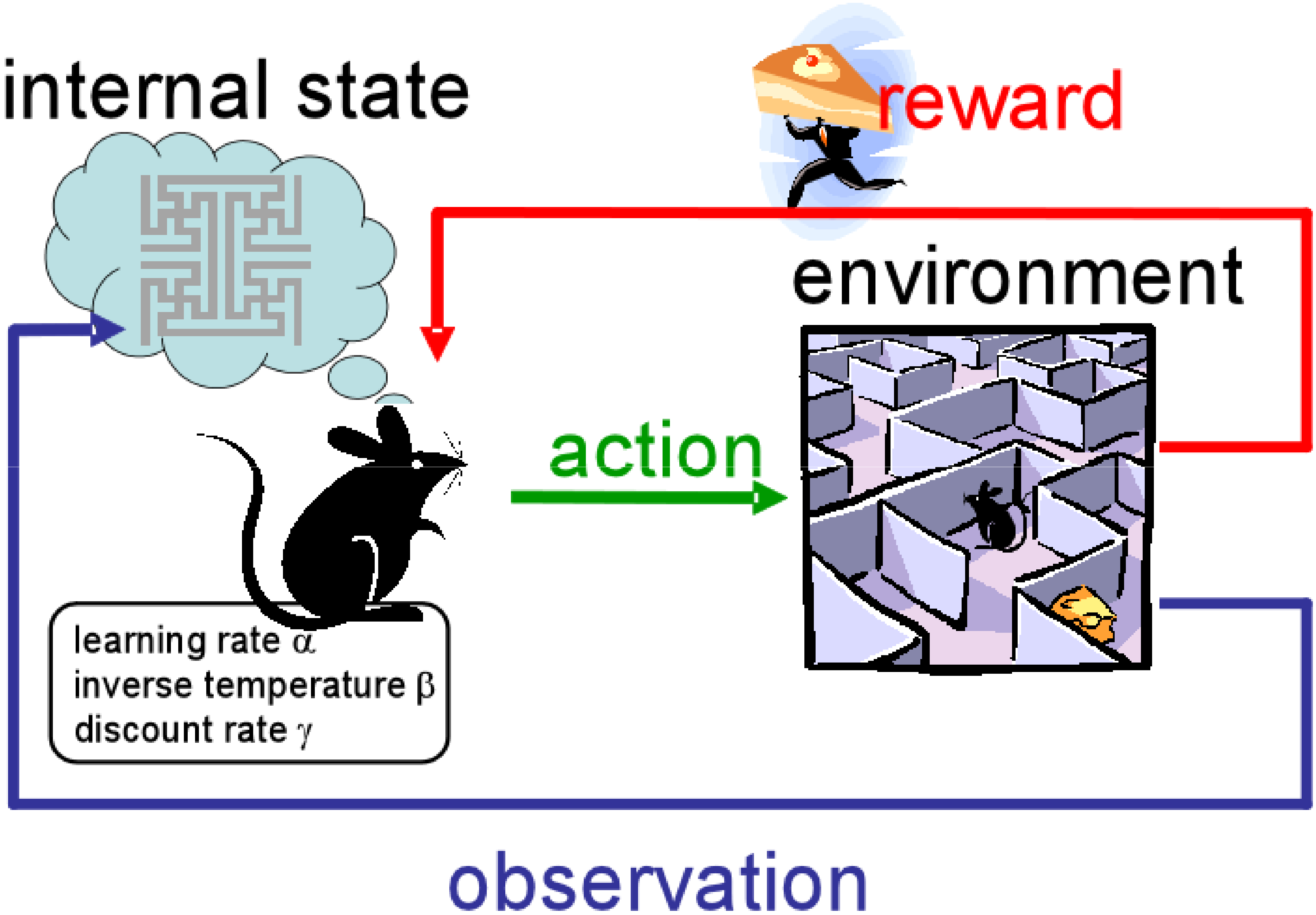
Ágens kialakítása: megerősítés → hasznosság leképezés

U(s) **hasznosság** függvény tanulása, ennek alapján a cselekvések eldöntése, hogy az elérhető hasznosság várható értéke max legyen

(környezet/ágens modellje **szükséges**)

Q(a, s) **cselekvés érték** függvény (állapot-cselekvés párok) tanulása, valamilyen várható hasznot tulajdonítva egy adott helyzetben egy adott cselekvésnek (környezet/ágens modell **nem szükséges**, közben tanult)

– **Q tanulás (model-free)**

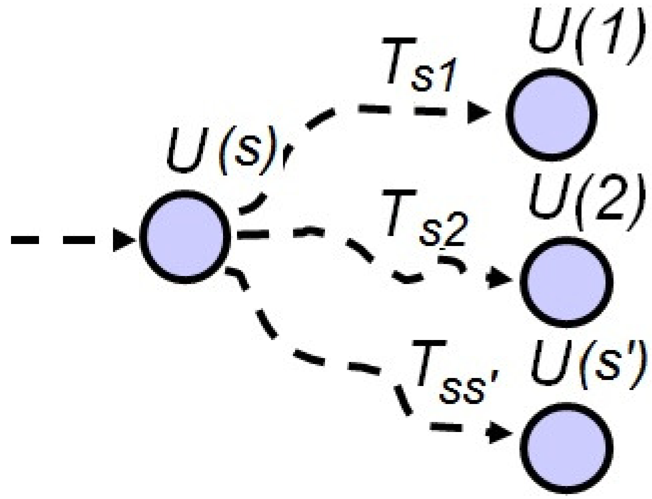

internal state
reward
environment
action
learning rate α
inverse temperature β
discount rate γ
observation

Fontos egyszerűsítés (ld. a MDF): a sorozat hasznossága
= a sorozat állapotaihoz rendelt hasznosságok összege.
= a hasznosság **additív**

Az állapot **hátralevő-jutalma** (**reward-to-go**):
azon jutalmak összege, amelyet akkor kapunk, ha az adott állapotból valamelyik végállapotig eljutunk.

Egy állapot várható hasznossága = **a hátralevő-jutalom várható értéke**

$$U^{\pi}(s) = E[\sum_t \gamma^t R(s_t) \mid \pi, s_0 = s]$$



$$U^{\pi}(s) = R(s) + \gamma \sum_{s'} T(s, \pi(s), s') U^{\pi}(s')$$

Adaptív Dinamikus Programozás

Az állapotátmeneti valószínűségek T*ij* a megfigyelt gyakoriságokkal becsülhetők.
Amint az ágens megfigyelte az összes állapothoz tartozó jutalom értéket, a hasznosság-értékek a következő **egyenletrendszer megoldásával** kaphatók meg:

$$U^{\pi}(s) = R(s) + \gamma \sum_{s'} T(s,\pi(s),s')U^{\pi}(s')$$

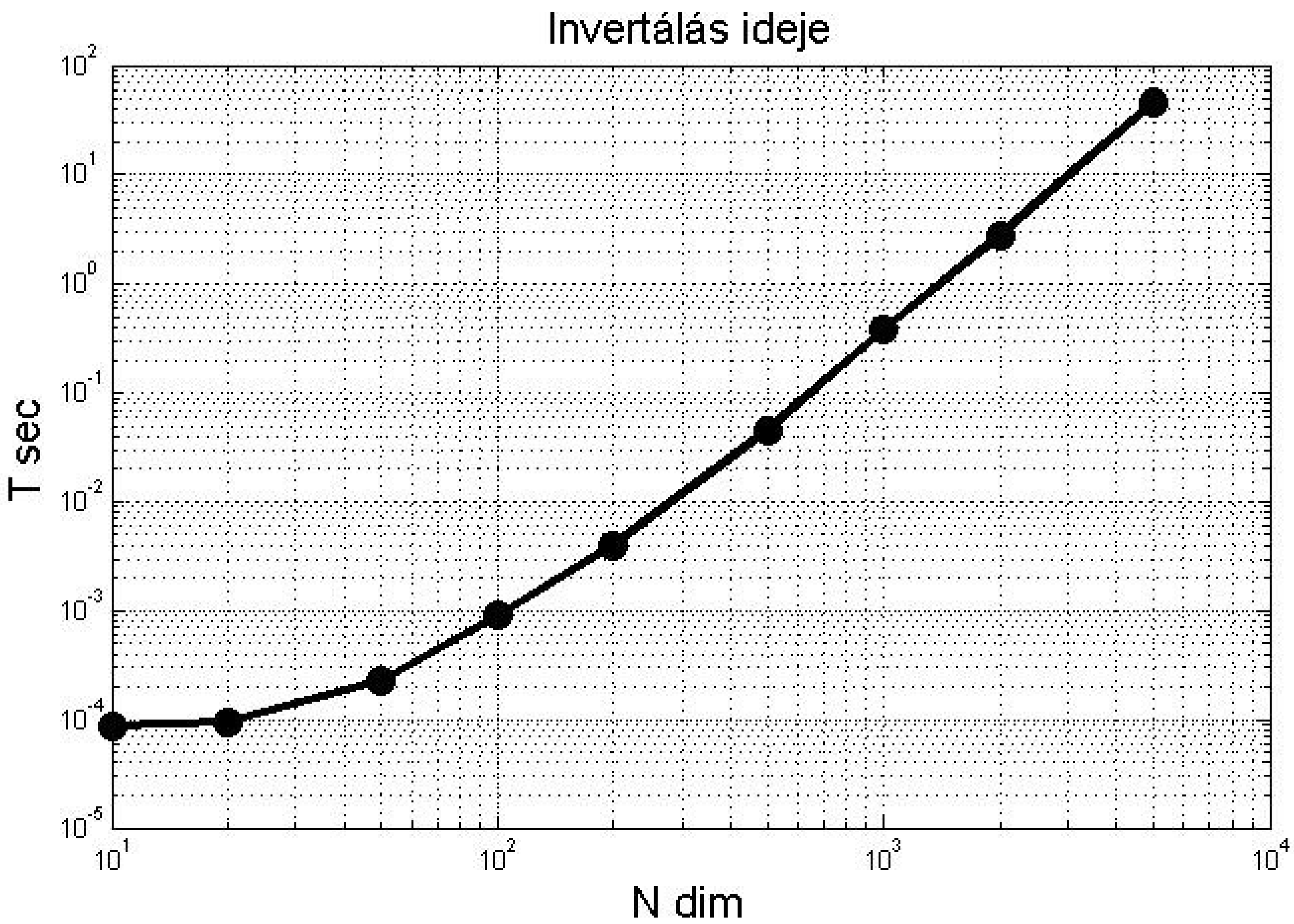
megfigyelt

megtanult

$$\begin{pmatrix} U(1) \\ U(2) \\ U(3) \\ \dots \\ U(N) \end{pmatrix} = \begin{pmatrix} R(1) \\ R(2) \\ R(3) \\ \dots \\ R(N) \end{pmatrix} + \gamma \begin{pmatrix} T_{11} & T_{12} & T_{13} & \dots & T_{1N} \\ T_{21} & T_{22} & T_{23} & \dots & T_{2N} \\ T_{31} & T_{32} & T_{33} & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ T_{N1} & T_{N2} & \dots & \dots & T_{NN} \end{pmatrix} \begin{pmatrix} U(1) \\ U(2) \\ U(3) \\ \dots \\ U(N) \end{pmatrix}$$

$$\mathbf{U} = \mathbf{R} + \gamma \mathbf{T}\mathbf{U} \qquad \mathbf{U} = (\mathbf{I} - \gamma \mathbf{T})^{-1}\mathbf{R}$$

és ha nem megy?



Invertálás ideje
T sec
N dim
10^{2}
10^{1}
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{1}
10^{2}
10^{3}
10^{4}

Az időbeli különbség (IK) tanulás (TD – Time Difference)

$$U(s) \leftarrow U(s) + \alpha(R^{\circ}(s) - U(s))$$

Monte Carlo mintavétel: használjuk fel a megfigyelt állapotátmeneteket a megfigyelt visszaterjesztett megerősítésekkel
- ki kell várni az epizód végét.

Alapötlet: használjuk fel a megfigyelt állapotátmeneteknél a jutalom becsült értékét:

$$R(s) + \gamma U(s')$$

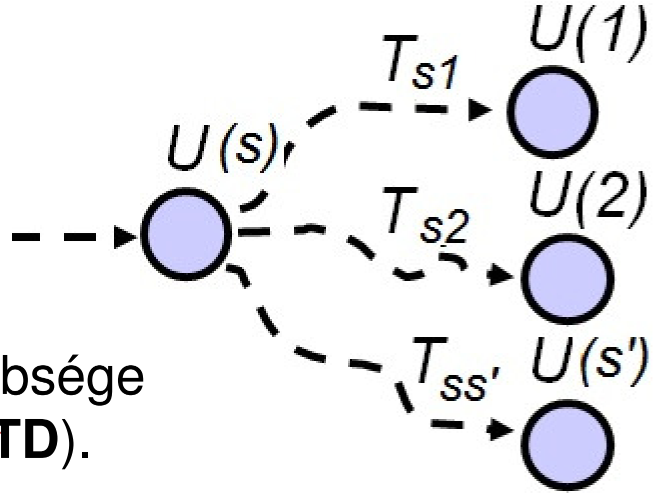
TD(0)-különbség (a jutalom becslése)
TD(0) -hiba:

$$\delta(s) = R(s) + \gamma U(s') - U(s)$$

$$U(s) \leftarrow (1-\alpha)U(s) + \alpha\delta(s) =$$

$$U(s) + \alpha(R(s) + \gamma U(s') - U(s))$$

α - **bátorsági faktor,** tanulási tényező
γ - **leszámoltatási** tényező (ld. MDF)



Az egymást követő állapotok hasznosság-különbsége **időbeli különbség** (**IK**) (**temporal difference**, **TD**).

Az összes időbeli különbség eljárásmód alapötlete

1. korrekt hasznosság-értékek esetén **lokálisan fennálló feltételek rendszere**:

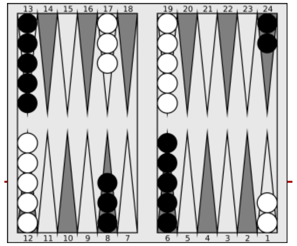
$$U^{\pi}(s) = R(s) + \gamma \sum_{s'} T(s, \pi(s), s') U^{\pi}(s')$$

2. **módosító egyenlet**, amely a becsléseinket ezen „**egyensúlyi" egyenlet irányába** módosítja:

$$U(s) \leftarrow U(s) + \alpha(R(s) + \gamma U(s') - U(s))$$

3. Csak a **(A)DP** módszer használta fel a **modell teljes** ismeretét.
Az **IK** az állapotok közt fennálló kapcsolatokra vonatkozó információt használja, de csak azt, amely az **aktuális tanulási sorozatból származik**.

Az **IK** változatlanul **működik előzetesen ismeretlen környezet esetén is**.



Ismeretlen környezetben végzett aktív tanulás

Döntés: melyik cselekvés? a cselekvésnek mik a kimeneteleik?
hogyan hatnak az elért jutalomra?

A környezeti modell: a többi állapotba való **átmenet valószínűsége egy adott cselekvés esetén**.

$$U(s) = R(s) + \gamma \max_a \sum_{s'} T(s,a,s')U(s')$$

Az állapottér felderítése - milyen cselekvést válasszunk? **Nehéz probléma**

Helyes-e azt a cselekvést választani, amelynek a jelenlegi hasznosság-becslés alapján legnagyobb a várható hasznossága?

Ez figyelmen kívül hagyja a **cselekvésnek a tanulásra gyakorolt hatását**.

Döntésnek kétféle hatása van:

1. Jutalmat eredményez a **jelenlegi** szekvenciában.
2. Befolyásolja az észleléseket, és ezáltal az ágens **tanulási képességét** – így jutalmat eredményezhet a **jövőbeni** szekvenciákban.

Az állapottér felderítése

Kompromisszum: a **jelenlegi jutalom**, amit a pillanatnyi hasznosság-becslés tükröz, és a **hosszú távú előnyök** közt.

Két megközelítés a cselekvés kiválasztásában:

„**Hóbortos, Felfedező**„: véletlen módon cselekszik, annak reményében, hogy végül is felfedezi az egész környezetet
„**Mohó**„: a jelenlegi becslésre alapozva maximalizálja a hasznot.

Hóbortos: képes jó hasznosság-becsléseket megtanulni az összes állapotra. Sohasem sikerül fejlődnie az optimális jutalom elérésében.
Mohó: gyakran talál egy jó utat. Utána ragaszkodik hozzá, és soha nem tanulja meg a többi állapot hasznosságát.

Ágens addig legyen hóbortos, amíg kevés fogalma van a környezetről, és legyen mohó, amikor a valósághoz közeli modellel rendelkezik.

Létezik-e optimális felfedezési stratégia?

Ágens súlyt kell adjon azoknak a cselekvéseknek, amelyeket még nem nagyon gyakran próbált, a kis hasznosságúnak gondolt cselekvéseket elkerülje.

felfedezési függvény

$$f(u,n) = \begin{cases} R^{+} & \text{ha } n < N_e \\ u & \text{különben} \end{cases}$$

mohóság ↔ kíváncsiság

$f(u,n)$ u-ban monoton növekvő, n-ben monoton csökkenő

$$U^{+}(s) \leftarrow R(s) + \gamma \max_a f\left(\sum_{s'} T(s,a,s')U^{+}(s'), N(a,s)\right)$$

$U^{+}(i)$: az i állapothoz rendelt hasznosság optimista becslése
$N(a,i)$: az i állapotban hányszor próbálkoztunk az a cselekvéssel

R^{+} a tetszőleges állapotban elérhető legnagyobb jutalom optimista becslése

Az a cselekvés, amely felderítetlen területek *felé* vezet, nagyobb súlyt kap.

ε-mohóság: az ágens ε valószínűséggel véletlen cselekvést választ, ill. 1-ε valószínűséggel mohó

Boltzmann-felfedezési modell
egy a cselekvés megválasztásának valószínűsége egy s állapotban:

$$P(a,s) = \frac{e^{hasznosság(a,s)/T}}{\sum_{a'} e^{hasznosság(a',s)/T}}$$

a T „hőmérséklet” a két véglet között szabályoz.
Ha T → ∞ , akkor a választás tisztán (egyenletesen) véletlen,
ha T → 0 , akkor a választás mohó.

A cselekvés-érték függvény tanulása

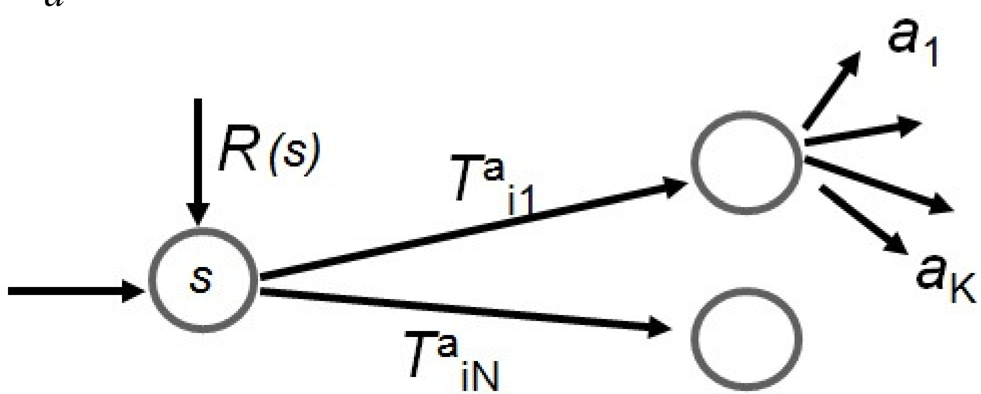
A cselekvés-érték függvény egy adott állapotban választott adott cselekvéshez egy várható hasznosságot rendel: **Q-érték**

$$U(s) = \max_a Q(a,s)$$

A Q-értékek fontossága:
a feltétel-cselekvés szabályokhoz hasonlóan lehetővé teszik a döntést **modell használata** nélkül,
ellentétben a feltétel-cselekvés szabályokkal, **közvetlenül** a jutalom visszacsatolásával **tanulhatók**.

Mint a hasznosság-értékeknél, felírhatunk egy kényszer egyenletet, amely **egyensúlyi állapotban**, amikor a **Q-értékek korrektek**, fenn kell álljon:

$$Q(a,s) = R(s) + \gamma \sum_{s'} T(s,a,s') \max_{a'} Q(a',s')$$



A cselekvés-érték függvény tanulása

Ezt az egyenletet közvetlenül felhasználhatjuk egy olyan iterációs folyamat módosítási egyenleteként, amely egy adott modell esetén a pontos Q-értékek számítását végzi.

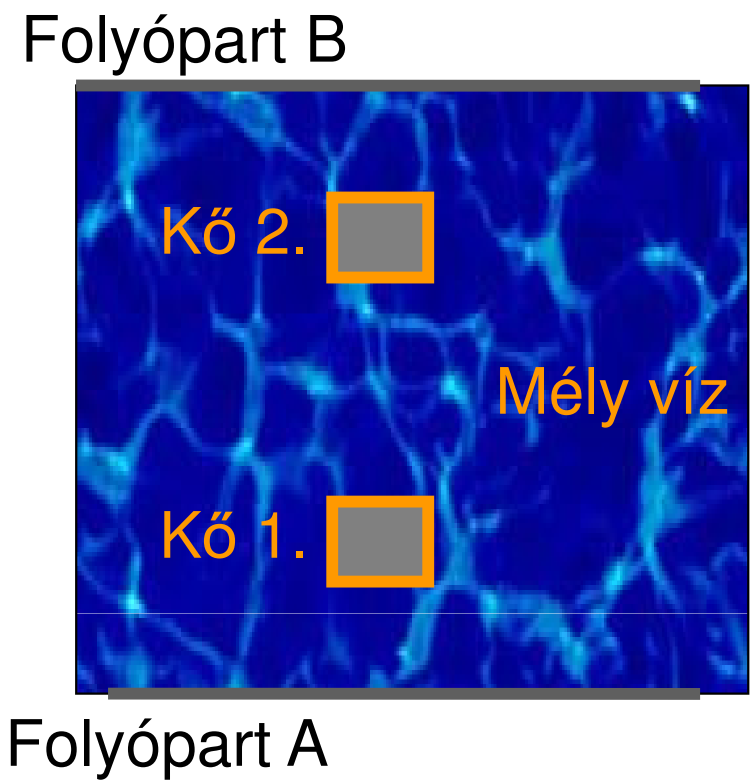
Az **időbeli különbség** viszont nem igényli a modell ismeretét. Az **IK** módszer **Q-tanulásának** módosítási egyenlete:

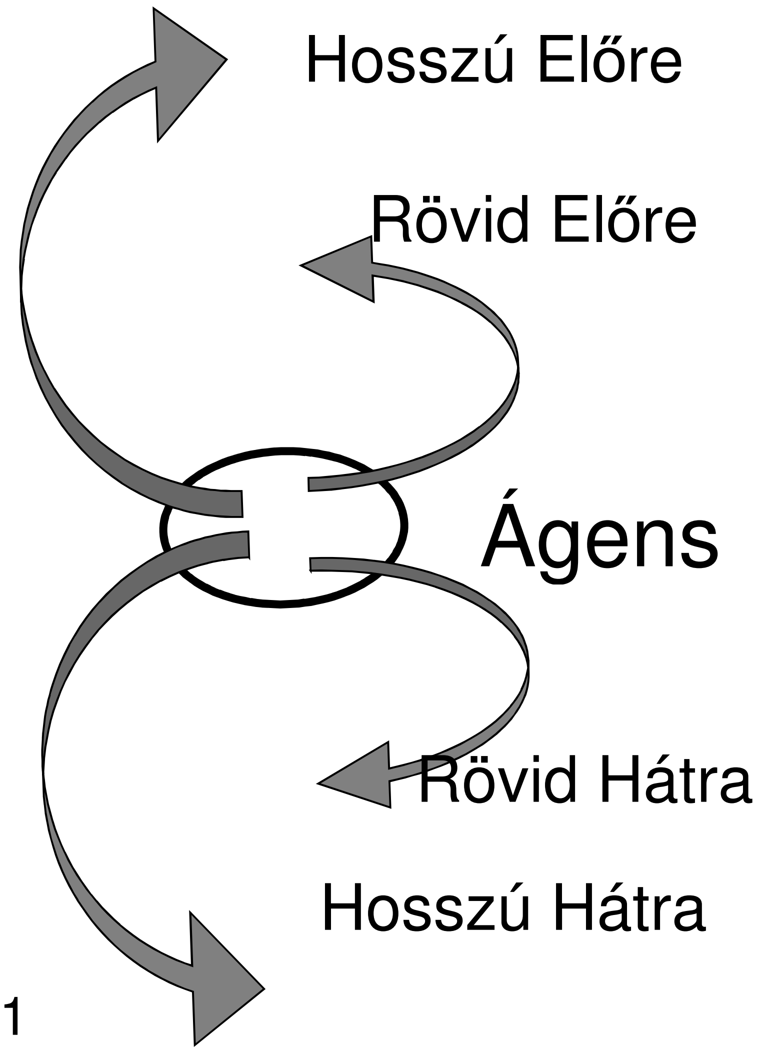
$$Q(a,s) \leftarrow Q(a,s) + \alpha\left(R(s) + \gamma \max_{a'} Q(a',s') - Q(a,s)\right)$$

SARSA Q-tanulás (State-Action-Reward-State-Action)

$$Q(a,s) \leftarrow Q(a,s) + \alpha[R(s) + \gamma Q(a',s') - Q(a,s)]$$

(a' megválasztása pl. Boltzmann felfedezési modellből)





Tanuló szekvencia:

RE→ HE→ RE→ +1 (száraz lábbal át)

RE→ HE→ HH→ HE→ HH→ RH→ HE→ -1

RE→ HE→ RH→ -1 (megfürdött)

....

Part B

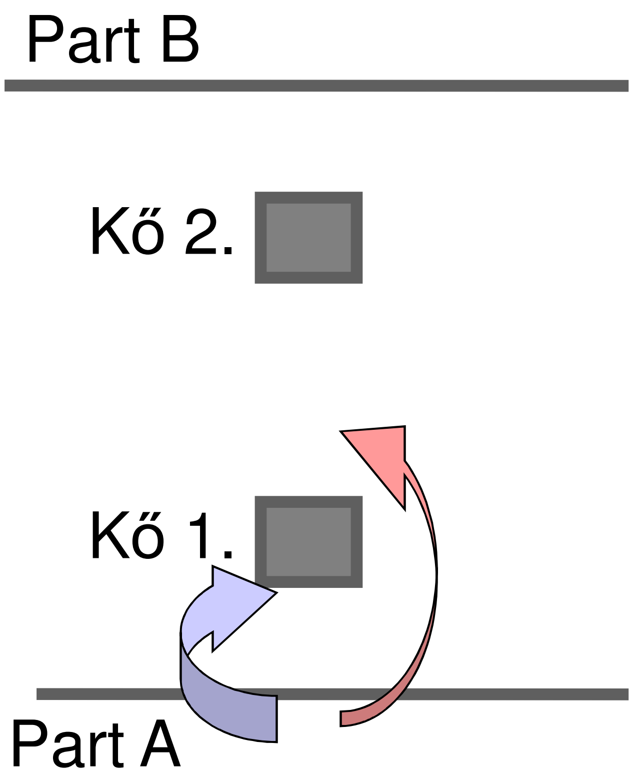
Kő 2.

$$Q(a,s) \leftarrow Q(a,s) + \alpha\left(R(s) + \gamma \max_{a'} Q(a',s') - Q(a,s)\right)$$

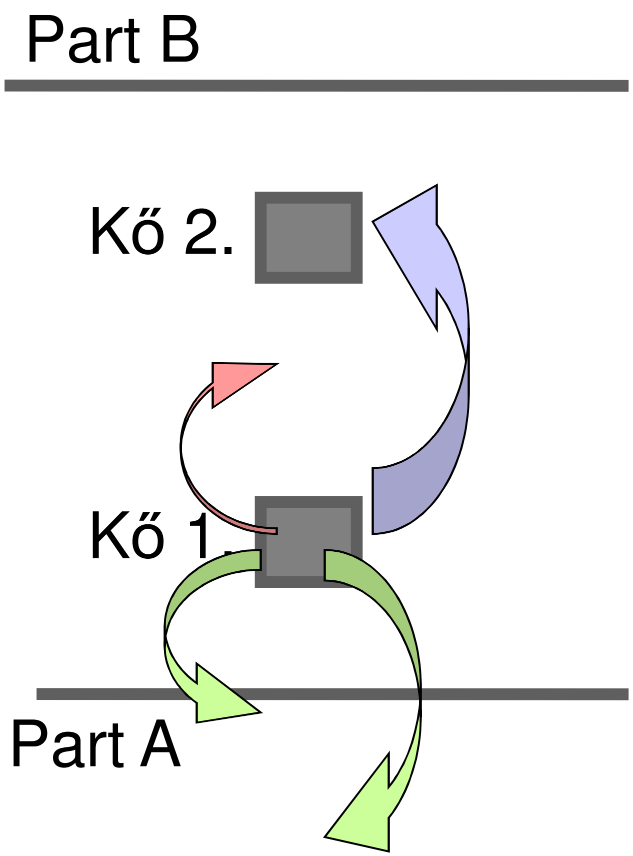
Kő 1.

Part A

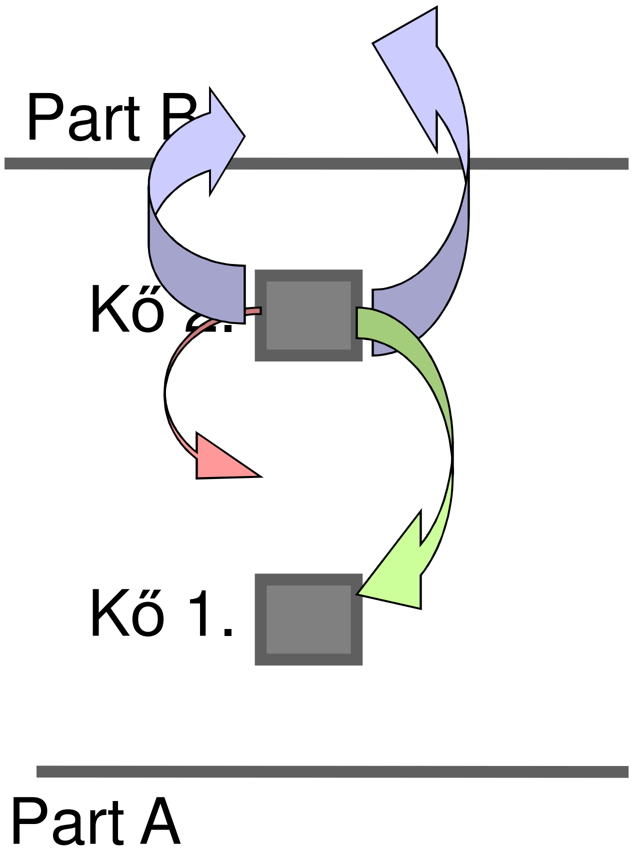
		Cselekvés			
		HH	RH	RE	HE
Állapot	Part A	0	0	-0.035	-0.877
	Kő 1.	-0.520	-0.550	-0.835	0.555
	Kő 2.	-0.204	-0.897	1.180	1.158



	HH	RH	RE	HE
Part A	0	0	-0.035	-0.877
Kő 1.	-0.520	-0.550	-0.835	0.555
Kő 2.	-0.204	-0.897	1.180	1.158



	HH	RH	RE	HE
Part A	0	0	-0.035	-0.877
Kő 1.	-0.520	-0.550	-0.835	0.555
Kő 2.	-0.204	-0.897	1.180	1.158



	HH	RH	RE	HE
Part A	0	0	-0.035	-0.877
Kő 1.	-0.520	-0.550	-0.835	0.555
Kő 2.	-0.204	-0.897	1.180	1.158

A megerősítéses tanulás általánosító képessége

Eddig: ágens által tanult hasznosság: **táblázatos** formában reprezentált
= **explicit reprezentáció**

Kis állapotterek: elfogadható, de a konvergencia idő és (ADP esetén) az iterációnkénti idő gyorsan nő a tér méretével.

Gondosan vezérelt közelítő ADP: 10000, vagy ennél is több.
De a való világhoz közelebb álló környezetek szóba se jöhetnek.
(a sakk stb. - állapottere 10^{50}-10^{120} nagyságrendű)
(az összes állapotot látni, hogy megtanuljunk játszani?!)

Egyetlen lehetőség: a függvény **implicit reprezentációja**:

Pl. egy játékban a táblaállás tulajdonságainak valamilyen halmaza $f_1,\ldots f_n$.
Az állás becsült hasznosság-függvénye:

$$\hat{U}_\theta(s) = \theta_1 f_1(s) + \theta_2 f_2(s) + \ldots \theta_n f_n(s)$$

C. Shannon, **Programming a Computer for Playing Chess**, Philosophical Magazine, 7th series, 41, no. 314 (March 1950): 256-75
A.L. Samuel, **Some Studies in Machine Learning Using the Game of Checkers. B.II-Recent Progress**, IBM. J. 3, 211-229 (1959), kb. 20 tulajdonság
Deep Blue (Feng-Hsiung Hsu) kb. 8000 tulajdonság, spec. sakk-csip

A hasznosság-függvény *n* értékkel jellemezhető, pl. 10^{120} érték helyett. Egy átlagos sakk kiértékelő függvény kb. 10 súllyal épül fel, tehát *hatalmas* tömörítést értünk el.

Az implicit reprezentáció által elért tömörítés teszi lehetővé, hogy a tanuló ágens általánosítani tudjon a már látott állapotokról az eddigiekben nem látottakra.

Az implicit reprezentáció legfontosabb aspektusa nem az, hogy kevesebb helyet foglal, hanem az, hogy lehetővé teszi a **bemeneti állapotok induktív általánosítását**. Az olyan módszerekről, amelyek ilyen reprezentációt tanulnak, azt mondjuk, hogy **bemeneti általánosítást** végeznek.

4 x 3 világ

$$E_j(s) = \frac{1}{2}\left(\hat{U}_\theta(s) - u_j(s)\right)^2$$

$$\hat{U}_\theta(x, y) = \theta_0 + \theta_1 x + \theta_2 y$$

Jósolt és kísérleti tényleges

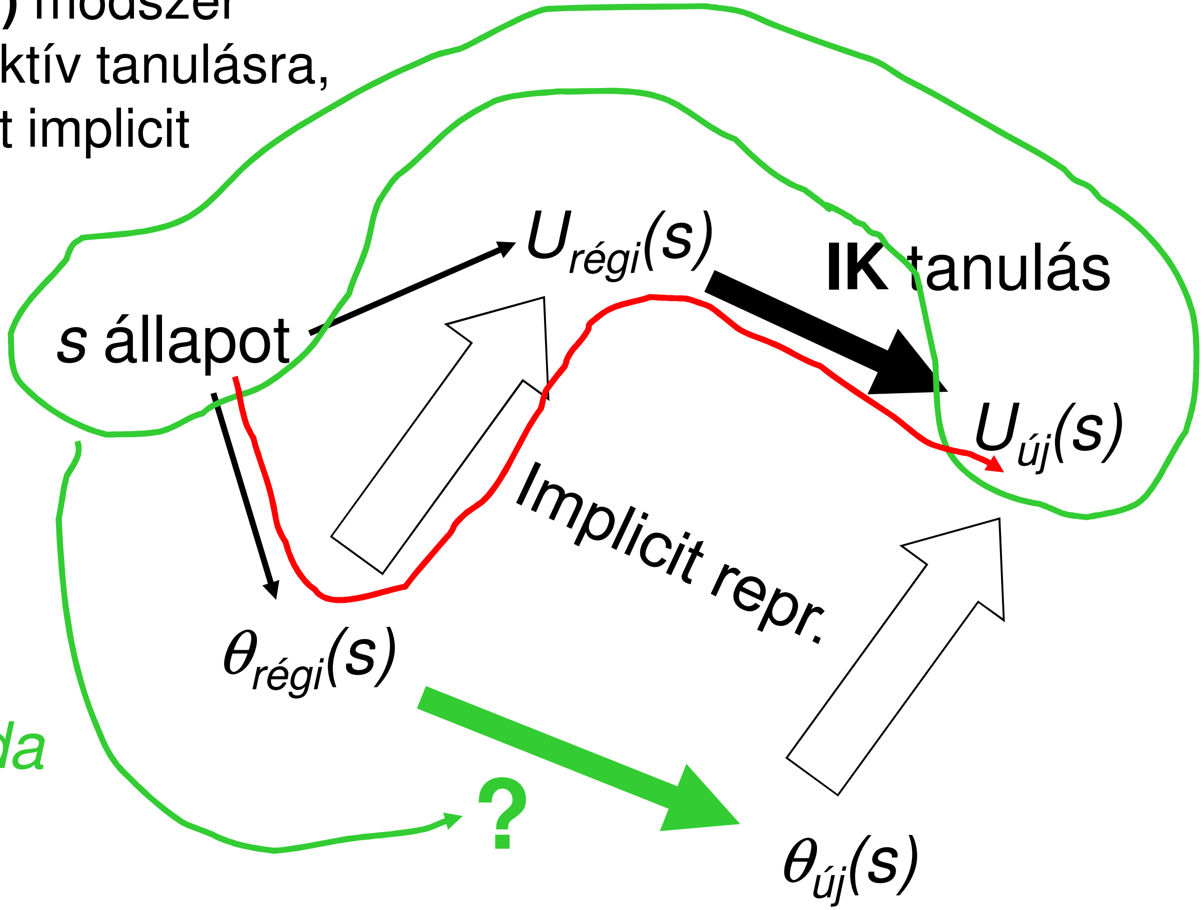
$$\theta_i \leftarrow \theta_i - \alpha \frac{\partial E_j(s)}{\partial \theta_i} = \theta_i + \alpha\left(u_j(s) - \hat{U}_\theta(s)\right)\frac{\partial \hat{U}_\theta(s)}{\partial \theta_i}$$

$$\theta_0 \leftarrow \theta_0 + \alpha\left(u_j(s) - \hat{U}_\theta(s)\right)$$

$$\theta_1 \leftarrow \theta_1 + \alpha\left(u_j(s) - \hat{U}_\theta(s)\right)x$$

$$\theta_2 \leftarrow \theta_2 + \alpha\left(u_j(s) - \hat{U}_\theta(s)\right)y$$

Az **IK (időbeli különbség)** módszer szintén alkalmazható induktív tanulásra, ha az *U*, vagy *Q* táblázatot implicit reprezentációra cseréltük (pl. neurális háló).

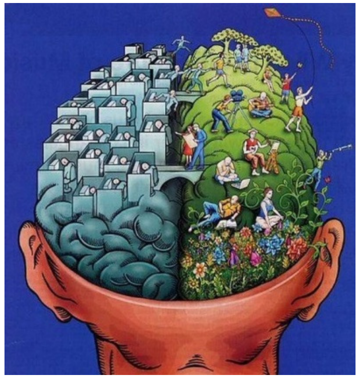


U/*Q* tanuló gradiensek

$$\theta_i \leftarrow \theta_i + \alpha\left[R(s) + \gamma \hat{U}_\theta(s') - \hat{U}_\theta(s)\right]\frac{\partial \hat{U}_\theta(s)}{\partial \theta_i}$$

$$\theta_i \leftarrow \theta_i + \alpha\left[R(s) + \gamma \max_{a'} \hat{Q}_\theta(a',s') - \hat{Q}_\theta(a,s)\right]\frac{\partial \hat{Q}_\theta(a,s)}{\partial \theta_i}$$

Budapesti Műszaki és Gazdaságtudományi Egyetem
Méréstechnika és Információs rendszerek Tanszék



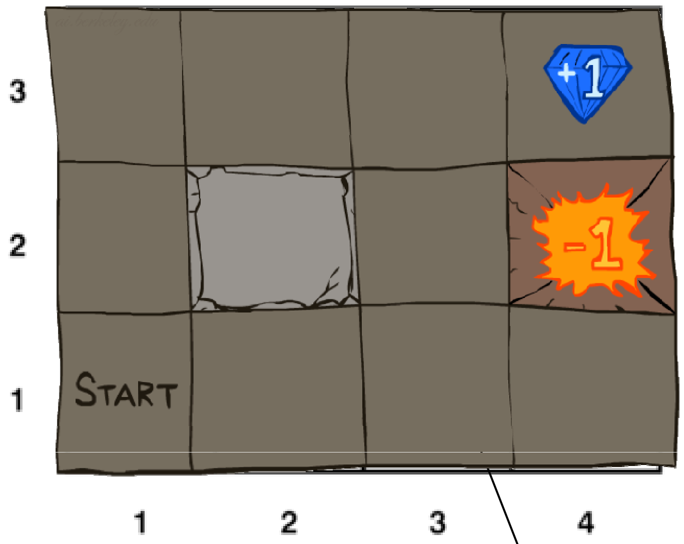
Markov döntési folyamat

Előadó: Dr. Hullám Gábor

Előadás anyaga: Dr. Dobrowiecki Tadeusz

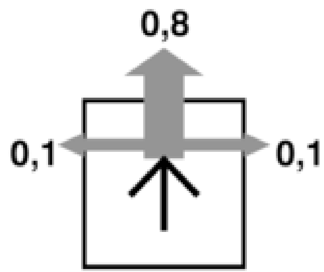
MŰEGYETEM 1782

Szekvenciális döntési probléma

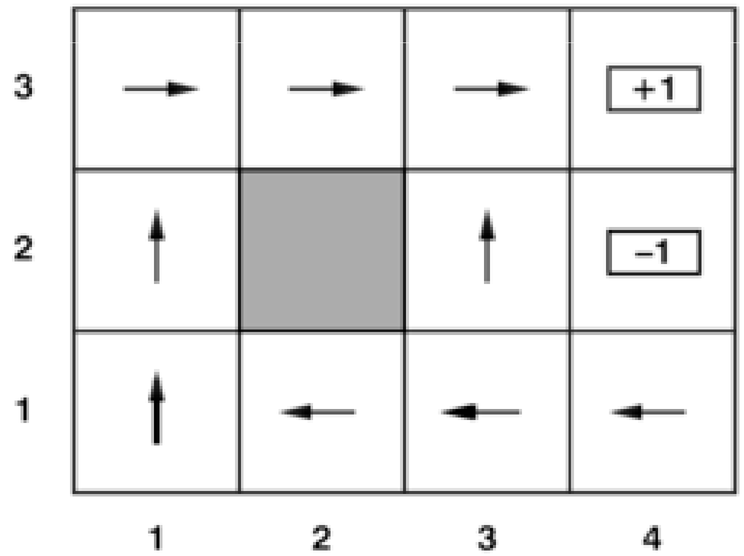


(a)

- 0.04



(b)

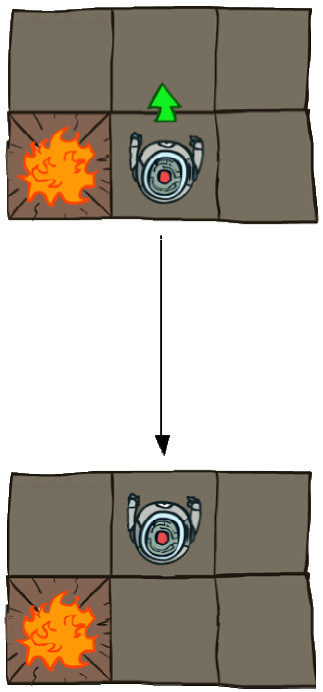


Fix út: fel, fel, jobbra, jobbra, jobbra?

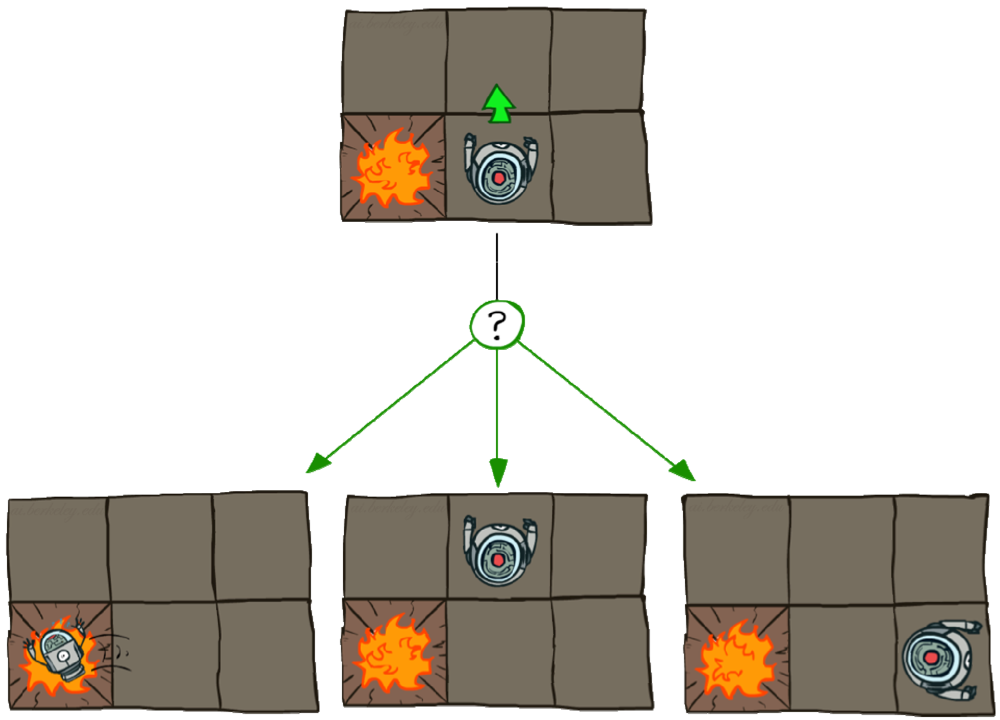
(optimális) Eljárásmód: $\pi(s) = a$

Grid World Cselekvések

Determinisztikus Grid World



Sztochasztikus Grid World



Szekvenciális döntési probléma

Markov döntési folyamat

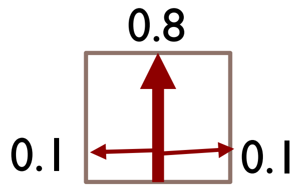
Kezdőállapot:	S0
Állapotátmenet-modell:	T(s, a, s′)
Jutalomfüggvény:	R(s), vagy R(s, a, s′)

Optimális eljárásmód = optimális mozgás, döntés cselekvés megválasztására, de nem elég egyszer, folyamatosan kell, amíg nincs vége (a problémának).

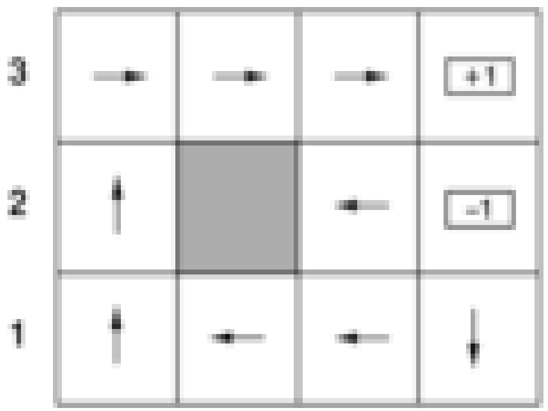
$\pi(s) = a$

$\pi^*(s) = a$

Szekvenciális döntési probléma

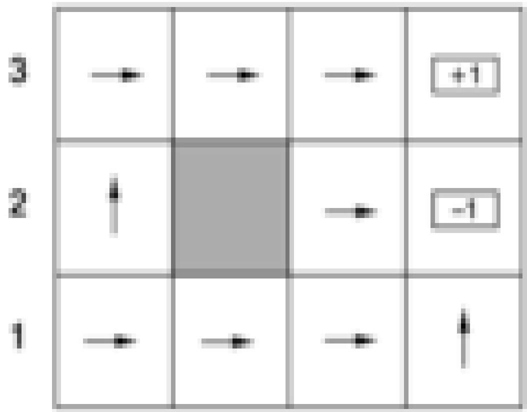


– 0,0221 < R(s) < 0



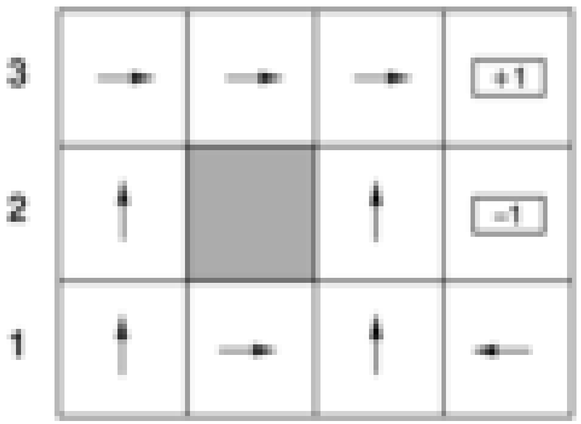
az élet csak kevéssé bánatos,
ne legyen kockázat!

R(s) ≤ –1,6284



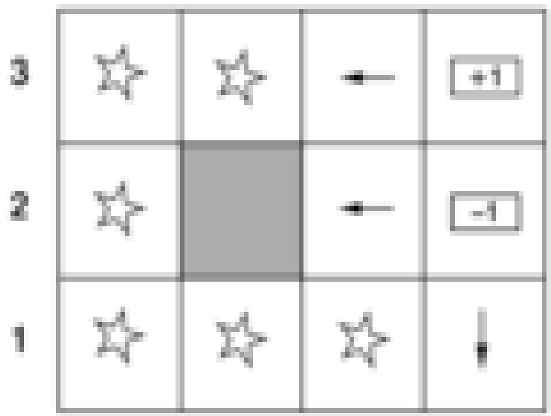
élet elviselhetetlen, ki!

– 0,4278 ≤ R(s) ≤ – 0,0850



élet kellemetlen; +1 állapot,
–1 kockázattal

R(s) > 0



élet kifejezetten élvezhető,
az ágens benn akar maradni

Szekvenciális döntési probléma

Optimalis szekvenciális döntési probléma

	optimális eljárásmód
végtelen horizont	stacionárius
véges horizont	nem-stacionárius

Többattribútumú hasznosságelmélet:
ágens preferenciái az állapotsorozatok között stacionáriusok

additív jutalmak $U_h([s_0, s_1, s_2, \ldots]) = R(s_0) + R(s_1) + R(s_2) + \ldots$

leszámítolt jutalmak

$$U_h([s_0, s_1, s_2, \ldots]) = R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \ldots$$

Leszámítolás

- Ésszerű a jutalmak maximalizálására törekedni
- Ésszerű egy jelenlegi jutalmat preferálni egy jövőbeli jutalom helyett
- Egy megoldás: a jutalmak értékei exponenciálisan csökkennek

1

Hasznosság most

γ

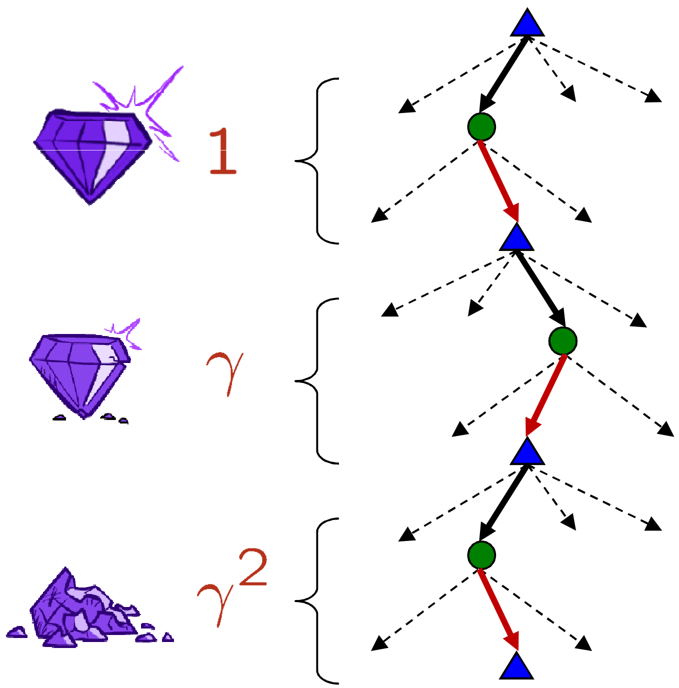
Hasznosság a következő lépésben

γ^2

Hasznosság 2 lépés múlva

Leszámítolás

- Ésszerű a jutalmak maximalizálására törekedni
- Ésszerű egy jelenlegi jutalmat preferálni egy jövőbeli jutalom helyett
- Egy megoldás: a jutalmak értékei exponenciálisan csökkennek



Szekvenciális döntési probléma

Leszámítolt jutalmak, egy végtelen sorozat hasznossága

$$U_h([s_0, s_1, s_2, \ldots]) = \Sigma_{t=0\ldots\infty} \gamma^t R(s_t) =< \Sigma_{t=0\ldots\infty} \gamma^t R_{\max} = R_{\max} / (1-\gamma)$$

Alternatív esetek:

1.) Ha van végállapot, és ha garantált, hogy az ágens végül bele kerül, akkor nincs szükség végtelen sorozatok összehasonlítására.
Egy eljárásmód, ami garantáltan végállapotba juttat, véges eljárásmód, $\gamma = 1$

2.) Időegységenkénti átlagjutalom

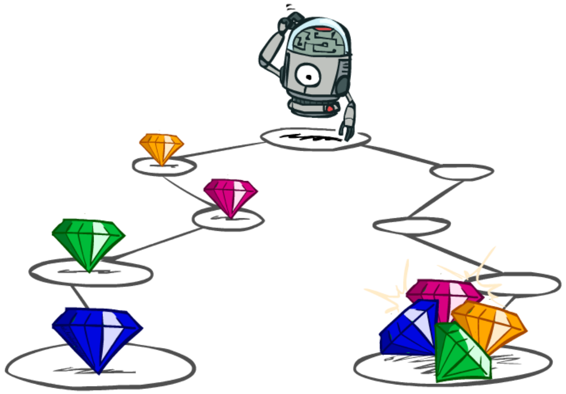
Szekvenciális döntési probléma

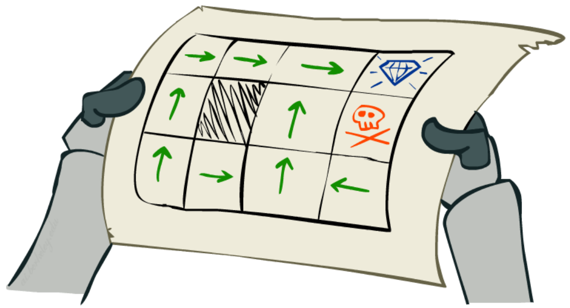
Leszámítolt jutalmak, egy végtelen sorozat hasznossága

$$U_h([s_0, s_1, s_2, \ldots]) = \Sigma_{t=0\ldots\infty} \gamma^t R(s_t) =< \Sigma_{t=0\ldots\infty} \gamma^t R_{\max} = R_{\max} / (1-\gamma)$$

Optimális eljárásmód

$$\pi^* = \arg\max_{\pi} E[\Sigma_{t=0\ldots\infty} \gamma^t R(s_t) \mid \pi]$$





Optimális eljárásmód meghatározása - Értékiteráció

Egy **állapot hasznossága –** a belőle kiinduló állapotsorozatok várható hasznossága

Az állapotsorozatok függnek a végrehajtott eljárásmódtól, így elsőként egy adott π eljárásmódra definiáljuk a hasznosságot:

$$U^{\pi}(s) = E\left[\sum_{t=0}^{\infty} \gamma^{t} R(s_t) \mid \pi, s_0 = s\right]$$

Optimális eljárásmód

$$\pi^{*}(s) = \arg\max_{a} \sum_{s'} T(s,a,s') U(s')$$

Optimális eljárásmód meghatározása - Értékiteráció

Az **állapot hasznossága** - az állapotban tartózkodás közvetlen jutalmának és a következő állapot várható leszámított hasznosságának az összege, feltéve, hogy az ágens az optimális cselekvést választja

$$U(s) = R(s) + \gamma \max_a \sum_{s'} T(s,a,s') U(s')$$

Bellman egyensúlyi **egyenlet**

Legyen $\gamma = 1$ és a nem végállapotoknál R(s) = – 0,04

Nézzük meg a 4 × 3-as világ Bellman-egyenleteinek egyikét.

Az (1, 1) állapothoz tartozó egyenlet:

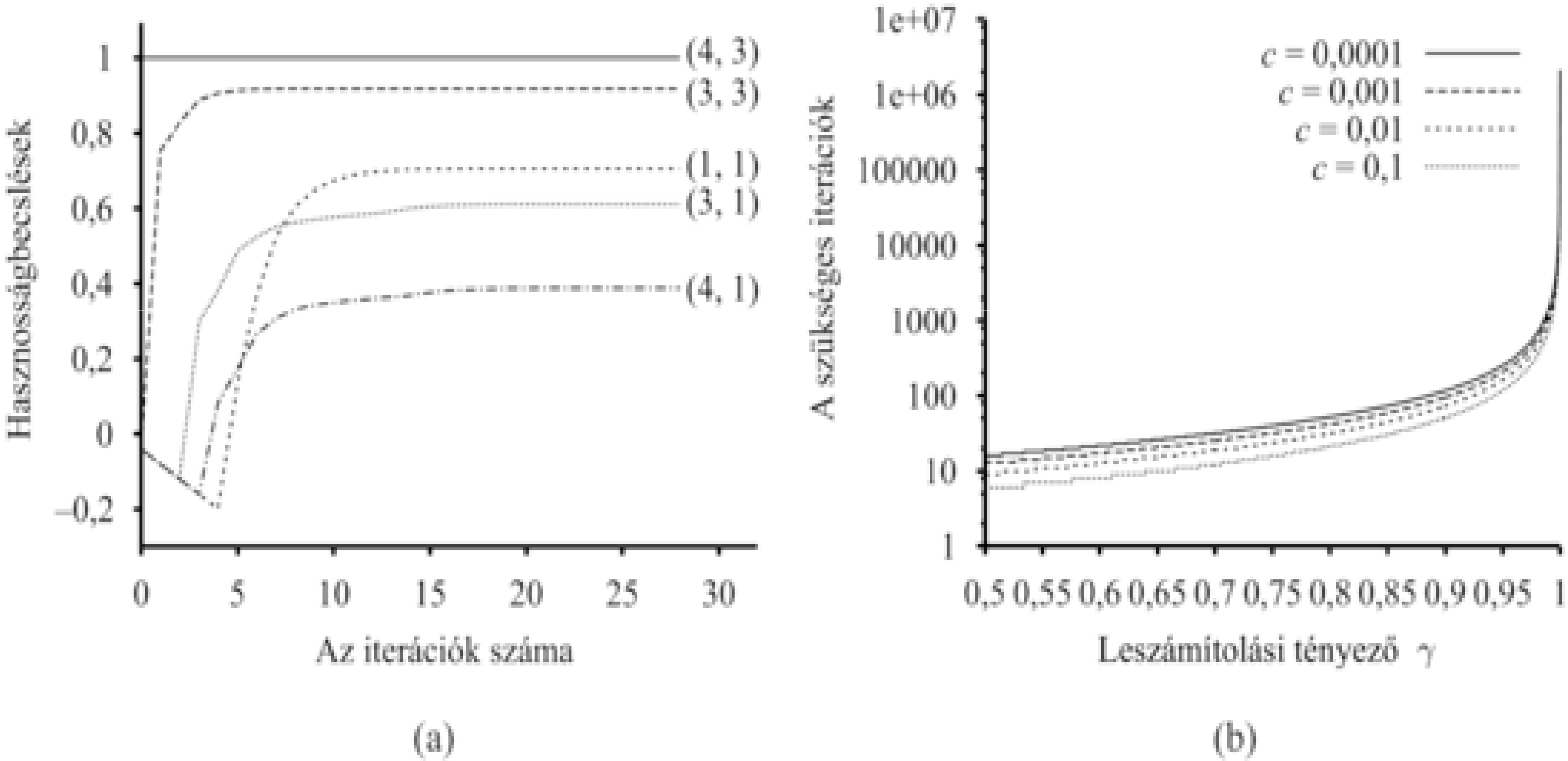
$$
\begin{aligned}
U(1, 1) = &-0{,}04 + \\
\gamma \max\{&0{,}8\, U(1, 2) + 0{,}1\, U(2, 1) + 0{,}1\, U(1, 1) && \text{(Fel)}, \\
&0{,}8\, U(2, 1) + 0{,}1\, U(1, 2) + 0{,}1\, U(1, 1) && \text{(Jobbra)}, \\
&0{,}9\, U(1, 1) + 0{,}1\, U(1, 2) && \text{(Balra)}, \\
&0{,}9\, U(1, 1) + 0{,}1\, U(2,1)\} && \text{(Le)}
\end{aligned}
$$

Sajnos nemlineáris – iteráció!

	1	2	3	4
3	0,812	0,868	0,918	+ 1
2	0,762		0,660	–1
1	0,705	0,655	0,611	0,388

Bellman-frissítés:

$$U_{t+1}(s) = R(s) + \gamma \max_a \sum_{s'} T(s,a,s') U_t(s')$$



A hasznosságok fejlődése

A szükséges értékiterációk száma, hogy a hiba garantáltan legfeljebb $\varepsilon = c\, R_{max}$ legyen

Az értékiteráció konvergenciája – **kontrakció** $U_{i+1} = B\, U_i$

$$\|U\| = \max_s |U(s)|$$

$\|BU_i - U^*\| \leq \gamma \|U_i - U^*\|$, U^* az igazi: $B(U^*) = U^*$

Az összes állapot hasznossága korlátos $\pm R_{max}/(1-\gamma)$ értékkel
A maximális kezdeti hiba $\|U_0 - U^*\| \leq 2R_{max}/(1-\gamma)$

Ha $\|U_{i+1} - U_i\| < \varepsilon(1-\gamma)/\gamma$, akkor $\|U_{i+1} - U^*\| < \varepsilon$

Eljárásmód-iteráció

(optimális eljárásmódot is kaphatunk, ha a hasznosság becslése pontatlan - ha egy cselekvés egyértelműen jobb, akkor a releváns állapotok pontos hasznosságát nem szükséges precízen tudnunk)

$$U(s) = R(s) + \gamma \max_a \sum_{s'} T(s,a,s')U(s')$$

Eljárásmód-értékelés

Egy adott π_i eljárásmódnál számítsuk ki $U_i = U^{\pi_i}$-t, az egyes állapotok hasznosságát mintha π_i volna végrehajtva.

lineáris!

$$U_t(s) = R(s) + \gamma \sum_{s'} T(s,\pi_t(s),s')U_t(s')$$

$$U_{t+1}(s) \leftarrow R(s) + \gamma \sum_{s'} T(s,\pi_t(s),s')U_t(s')$$

Módosított eljárásmód-iteráció

Eljárásmód-javítás

for each s állapotra **in** S **do**

if $\max_a \sum_{s'} T(s,a,s')U[s'] > \sum_{s'} T(s,\pi[s],s')U[s']$ **then**

$\pi[s] \leftarrow \mathrm{argmax}_a \sum_{s'} T(s,a,s')U[s']$

Részlegesen megfigyelhető Markov döntési folyamat

Kezdőállapot: S0
Állapotátmenet-modell: T(s, a, s′)
Jutalomfüggvény: R(s), v. R(s, a, s′)
Megfigyelési modell,
az s állapotban az o megfigyelés érzékelésének a valószínűsége
O(s, o)

Hiedelmi állapot = *b*(*s*) = eloszlás állapotok felett

0,111	0,111	0,111	$\boxed{0,000}$
0,111		0,111	$\boxed{0,000}$
0,111	0,111	0,111	0,111

$$\left\langle \frac{1}{9},\frac{1}{9},\frac{1}{9},\frac{1}{9},\frac{1}{9},\frac{1}{9},\frac{1}{9},\frac{1}{9},\frac{1}{9},0,0 \right\rangle$$

$$b'(s') = \alpha O(s',o) \sum_{s} T(s,a,s') b(s)$$

(szűrés)

$$P(o \mid a,b) = \sum_{s'} P(o \mid a,s',b)\, P(s' \mid a,b)$$

$$= \sum_{s'} O(s',o) P(s' \mid a,b) = \sum_{s'} O(s',o) \sum_{s} T(s,a,s') b(s)$$

$$\tau(b,a,b') = P(b' \mid a,b) = \sum_{o} P(b' \mid o,a,b)\, P(o \mid a,b)$$

$$= \sum_{o} P(b' \mid o,a,b) \sum_{s'} O(s',o) \sum_{s} T(s,a,s') b(s)$$

$$\rho(b) = \sum_{s} b(s) R(s)$$

RMMDF megoldása a fizikai (véges) állapottérben redukálható egy MDF megoldására a hozzá tartozó hiedelmi állapot térben (val. eloszlások folytonos terében)

Szekvenciális döntési problémából indulunk

Markov döntési folyamat (MDF)

S0
T(s, a, s′)
R(s), v. R(s, a, s′)

$$U(s) = R(s) + \gamma \max_a \sum_{s'} T(s,a,s')U(s')$$

$$\pi(s) \leftarrow U(s)$$

$$U_{i+1}(s) \leftarrow R(s) + \gamma \max_a \sum_{s'} T(s,a,s')U_i(s') \qquad \pi(s) \leftarrow U_i(s)$$

$$\pi^*(s) = \arg\max_a \sum_{s'} T(s,a,s')U(s')$$ Optimális eljárásmód

Megerősítéses tanulás

Kezdőállapot:	ismert
Állapotátmenet-modell:	**nem ismert, legfeljebb a tapasztalat**
Jutalomfüggvény:	**nem ismert, legfeljebb ahogy jön**
Optimális eljárásmód	**megtanulni, mindennek ellenére?!**

Megerősítéses tanulás

Pl. sakkjáték: tanító nélkül is van visszacsatolás a játék végén:
nyert vagy **vesztett** = +/- **jutalom**, azaz **megerősítés.**

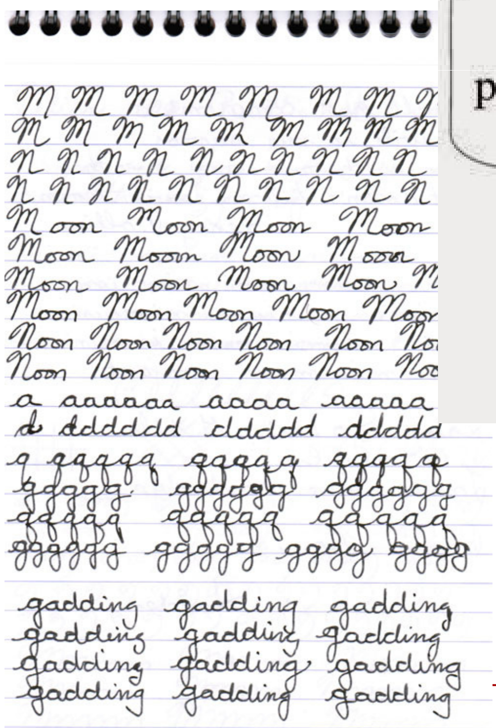
... a játék végén = a cselekvés szekvencia végén

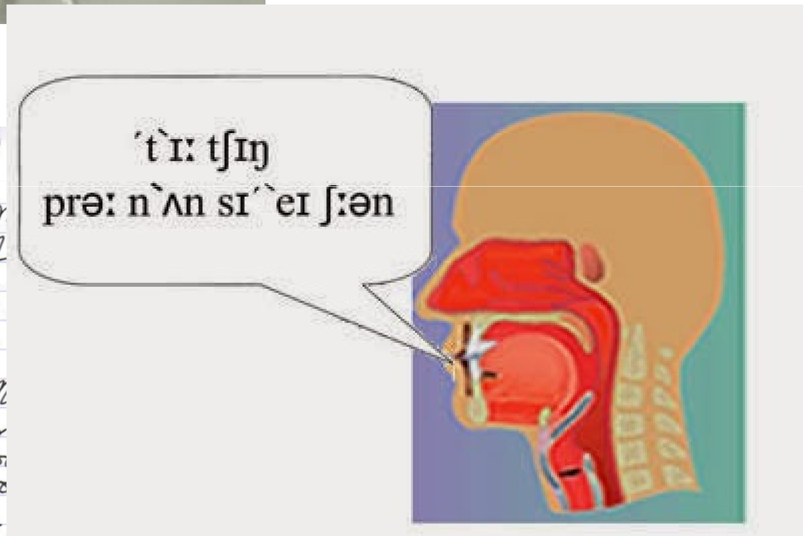
Megerősítés: milyen gyakran? milyen erős? milyen értékű?

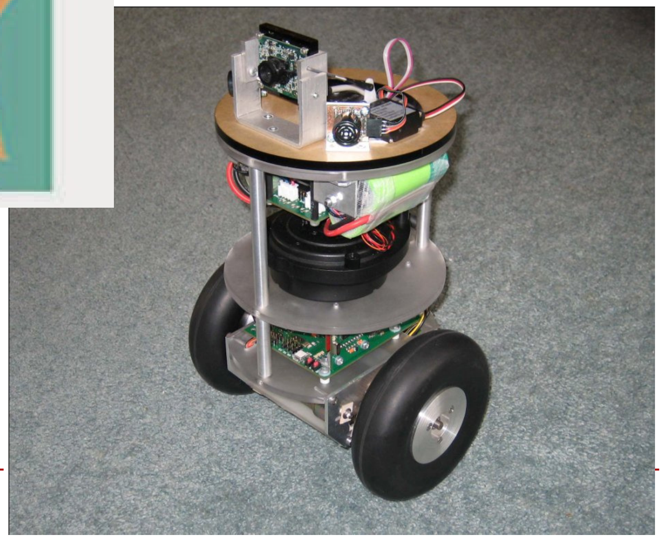
A feladat:

(ritka) jutalmakból megtanulni egy sikeres ágens-függvényt (optimális eljárásmód: melyik állapot, melyik cselekvés hasznos).

Nehéz: az információhiány miatt az ágens sohasem tudja, hogy **mik a jó lépések**, azt sem, **melyik jutalom melyik cselekvésből ered.**


Moon Moon Moon Moon
Noon Noon Noon Noon
a aaaaa aaaa aaaaa
d ddddd ddddd dddd
gadding gadding gadding


´t`ɪː tʃɪŋ
prəː n`ʌn sɪ´`eɪ ʃːən



Ágens tudása:

induláskor **tudja már** a környezetet és a cselekvéseinek hatását, vagy pedig még ezt is **meg kell tanulnia**.

Megerősítés:

csak a **végállapotban**, vagy menet közben **bármelyikben**.

Ágens:

passzív tanuló: figyeli a világ alakulását és tanul.

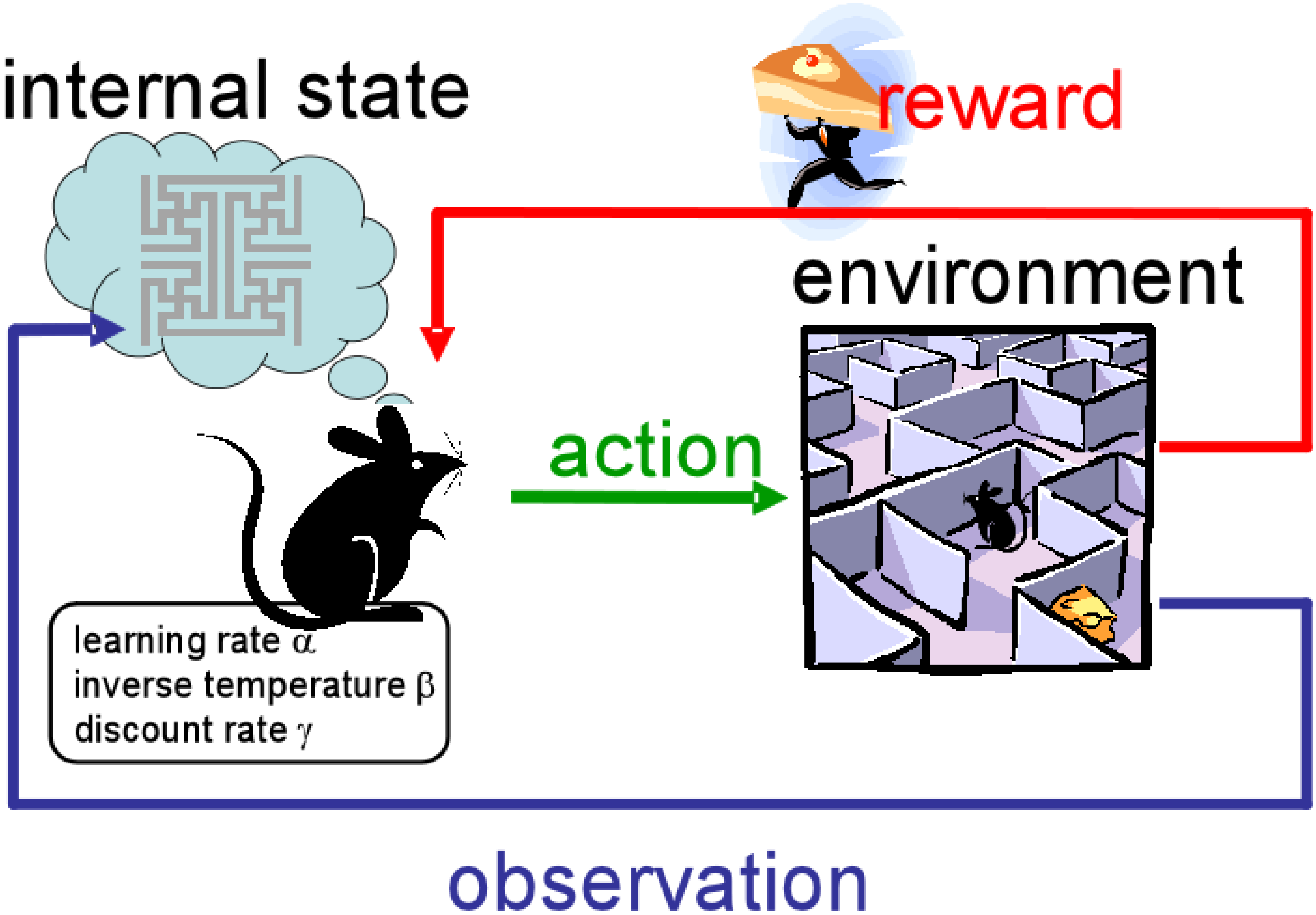
aktív tanuló: a megtanult információ birtokában cselekednie is kell.

felfedező …

Ágens kialakítása: megerősítés → hasznosság leképezés

U(s) **hasznosság** függvény tanulása, ennek alapján a cselekvések eldöntése, hogy az elérhető hasznosság várható értéke max legyen
(környezet/ágens modellje **szükséges**)

Q(a, s) **cselekvés érték** függvény (állapot-cselekvés párok) tanulása, valamilyen várható hasznot tulajdonítva egy adott helyzetben egy adott cselekvésnek (környezet/ágens modell **nem szükséges**, közben tanult)
– **Q tanulás (model-free)**

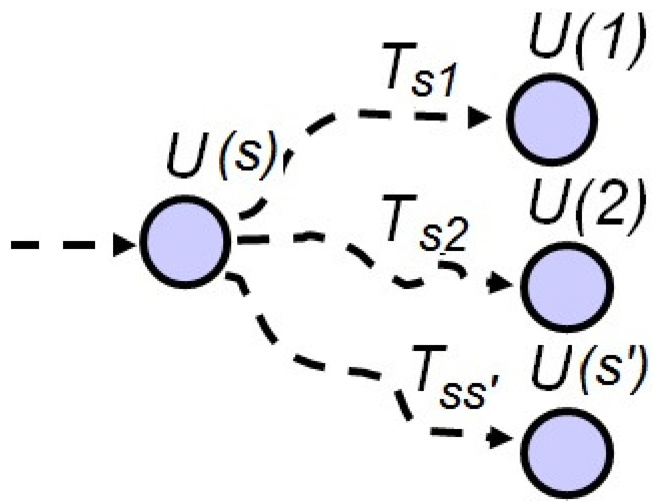

internal state
reward
environment
action
learning rate α
inverse temperature β
discount rate γ
observation

Fontos egyszerűsítés (ld. a MDF): a sorozat hasznossága
= a sorozat állapotaihoz rendelt hasznosságok összege.
= a hasznosság **additív**

Az állapot **hátralevő-jutalma** (**reward-to-go**):
azon jutalmak összege, amelyet akkor kapunk, ha az adott állapotból valamelyik végállapotig eljutunk.

Egy állapot várható hasznossága = **a hátralevő-jutalom várható értéke**

$$U^{\pi}(s) = E[\sum_t \gamma^t R(s_t) \,|\, \pi, s_0 = s]$$



$$U^{\pi}(s) = R(s) + \gamma \sum_{s'} T(s, \pi(s), s') U^{\pi}(s')$$

Adaptív Dinamikus Programozás

Az állapotátmeneti valószínűségek T*ij* a megfigyelt gyakoriságokkal becsülhetők.
Amint az ágens megfigyelte az összes állapothoz tartozó jutalom értéket, a hasznosság-értékek a következő **egyenletrendszer megoldásával** kaphatók meg:

$$U^{\pi}(s) = R(s) + \gamma \sum_{s'} T(s,\pi(s),s')U^{\pi}(s')$$

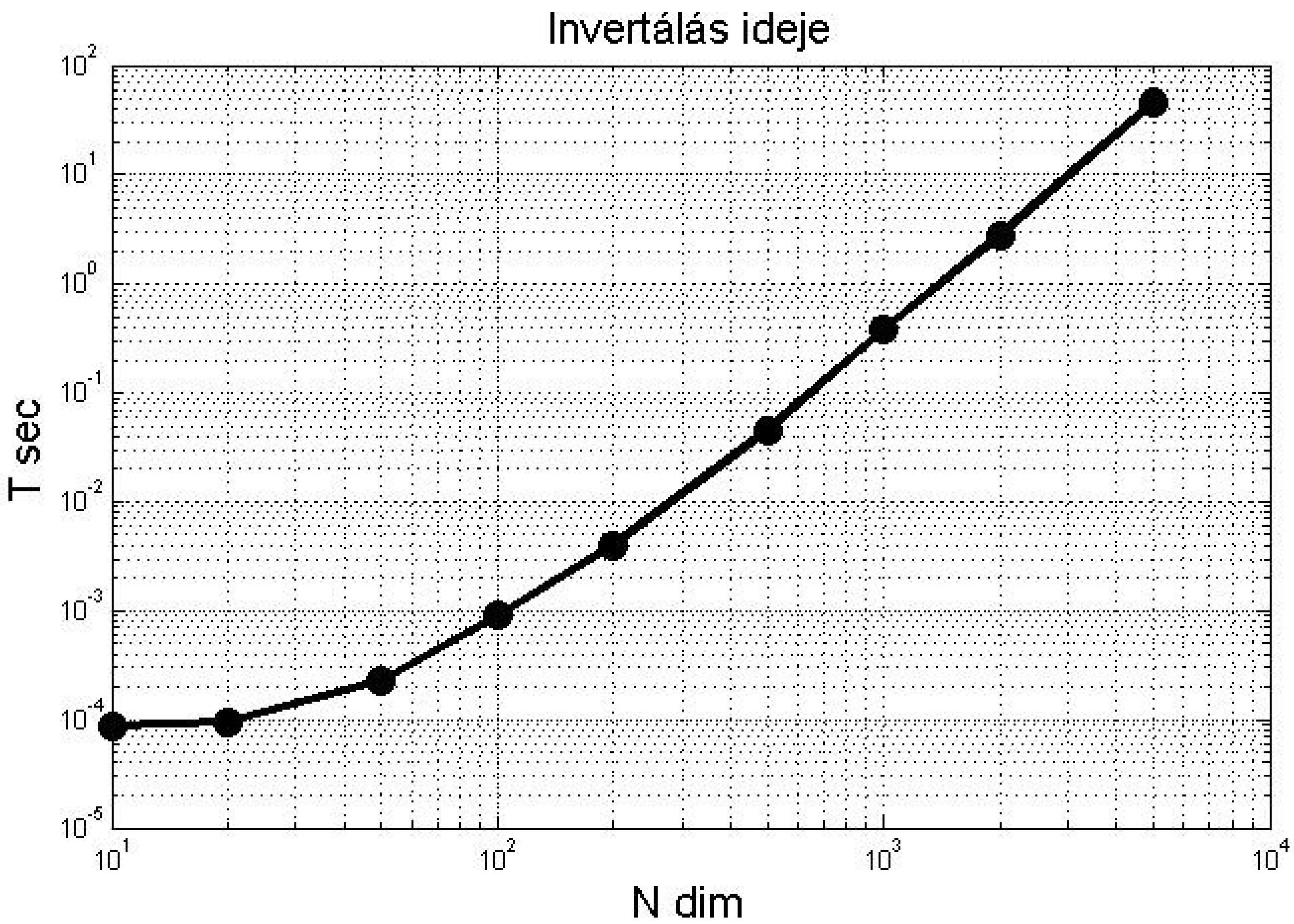
megfigyelt

megtanult

$$\begin{pmatrix} U(1) \\ U(2) \\ U(3) \\ \dots \\ U(N) \end{pmatrix} = \begin{pmatrix} R(1) \\ R(2) \\ R(3) \\ \dots \\ R(N) \end{pmatrix} + \gamma \begin{pmatrix} T_{11} & T_{12} & T_{13} & \dots & T_{1N} \\ T_{21} & T_{22} & T_{23} & \dots & T_{2N} \\ T_{31} & T_{32} & T_{33} & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ T_{N1} & T_{N2} & \dots & \dots & T_{NN} \end{pmatrix} \begin{pmatrix} U(1) \\ U(2) \\ U(3) \\ \dots \\ U(N) \end{pmatrix}$$

$$\mathbf{U} = \mathbf{R} + \gamma \mathbf{T} \mathbf{U} \qquad \mathbf{U} = (\mathbf{I} - \gamma \mathbf{T})^{-1} \mathbf{R}$$

és ha nem megy?


Invertálás ideje
T sec
N dim
10^{2}
10^{1}
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{1}
10^{2}
10^{3}
10^{4}

Az időbeli különbség (IK) tanulás (TD – Time Difference)

$$U(s) \leftarrow U(s) + \alpha(R^{\circ}(s) - U(s))$$

Monte Carlo mintavétel: használjuk fel a megfigyelt állapotátmeneteket a megfigyelt visszaterjesztett megerősítésekkel
- ki kell várni az epizód végét.

Alapötlet: használjuk fel a megfigyelt állapotátmeneteknél a jutalom becsült értékét:

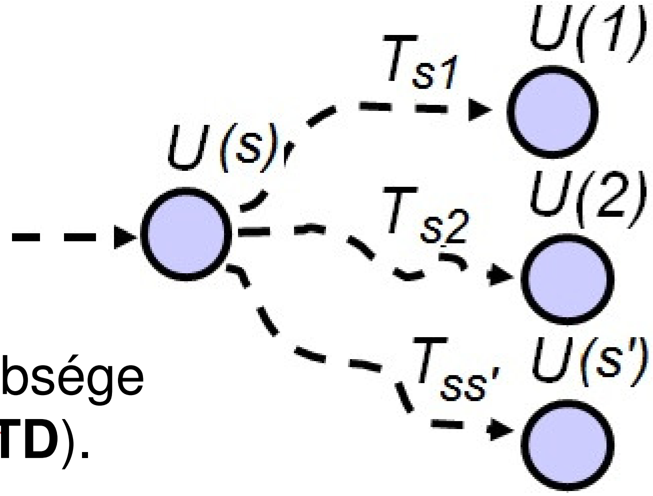
$$R(s) + \gamma U(s')$$

TD(0)-különbség (a jutalom becslése)
TD(0) -hiba:

$$\delta(s) = R(s) + \gamma U(s') - U(s)$$

$$U(s) \leftarrow (1-\alpha)\, U(s) + \alpha\,\delta(s) =$$

$$U(s) + \alpha(R(s) + \gamma U(s') - U(s))$$



α - **bátorsági faktor,** tanulási tényező
γ - **leszámoltatási** tényező (ld. MDF)

Az egymást követő állapotok hasznosság-különbsége **időbeli különbség** (**IK**) (**temporal difference**, **TD**).

Az összes időbeli különbség eljárásmód alapötlete

1. korrekt hasznosság-értékek esetén **lokálisan fennálló feltételek rendszere**:

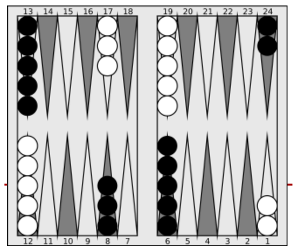
$$U^{\pi}(s) = R(s) + \gamma \sum_{s'} T(s, \pi(s), s') U^{\pi}(s')$$

2. **módosító egyenlet**, amely a becsléseinket ezen „**egyensúlyi" egyenlet irányába** módosítja:

$$U(s) \leftarrow U(s) + \alpha(R(s) + \gamma U(s') - U(s))$$

3. Csak a **(A)DP** módszer használta fel a **modell teljes** ismeretét.
Az **IK** az állapotok közt fennálló kapcsolatokra vonatkozó információt használja, de csak azt, amely az **aktuális tanulási sorozatból származik**.

Az **IK** változatlanul **működik előzetesen ismeretlen környezet esetén is**.



Ismeretlen környezetben végzett aktív tanulás

Döntés: melyik cselekvés? a cselekvésnek mik a kimeneteleik?
hogyan hatnak az elért jutalomra?

A környezeti modell: a többi állapotba való **átmenet valószínűsége egy adott cselekvés esetén**.

$$U(s) = R(s) + \gamma \max_a \sum_{s'} T(s,a,s')U(s')$$

Az állapottér felderítése - milyen cselekvést válasszunk? **Nehéz probléma**

Helyes-e azt a cselekvést választani, amelynek a jelenlegi hasznosság-becslés alapján legnagyobb a várható hasznossága?

Ez figyelmen kívül hagyja a **cselekvésnek a tanulásra gyakorolt hatását**.

Döntésnek kétféle hatása van:

1. Jutalmat eredményez a **jelenlegi** szekvenciában.
2. Befolyásolja az észleléseket, és ezáltal az ágens **tanulási képességét** – így jutalmat eredményezhet a **jövőbeni** szekvenciákban.

Az állapottér felderítése

Kompromisszum: a **jelenlegi jutalom**, amit a pillanatnyi hasznosság-becslés tükröz, és a **hosszú távú előnyök** közt.

Két megközelítés a cselekvés kiválasztásában:

„**Hóbortos, Felfedező**„: véletlen módon cselekszik, annak reményében, hogy végül is felfedezi az egész környezetet
„**Mohó**„: a jelenlegi becslésre alapozva maximalizálja a hasznot.

Hóbortos: képes jó hasznosság-becsléseket megtanulni az összes állapotra. Sohasem sikerül fejlődnie az optimális jutalom elérésében.
Mohó: gyakran talál egy jó utat. Utána ragaszkodik hozzá, és soha nem tanulja meg a többi állapot hasznosságát.

Ágens addig legyen hóbortos, amíg kevés fogalma van a környezetről, és legyen mohó, amikor a valósághoz közeli modellel rendelkezik.

Létezik-e optimális felfedezési stratégia?

Ágens súlyt kell adjon azoknak a cselekvéseknek, amelyeket még nem nagyon gyakran próbált, a kis hasznosságúnak gondolt cselekvéseket elkerülje.

felfedezési függvény

$$f(u,n) = \begin{cases} R^+ & \text{ha } n < N_e \\ u & \text{különben} \end{cases}$$

mohóság ↔ kíváncsiság

f(*u*,*n*) *u*-ban monoton növekvő, *n*-ben monoton csökkenő

$$U^+(s) \leftarrow R(s) + \gamma \max_a f\left(\sum_{s'} T(s,a,s')U^+(s'), N(a,s)\right)$$

$U^+(i)$: az *i* állapothoz rendelt hasznosság optimista becslése
$N(a,i)$: az *i* állapotban hányszor próbálkoztunk az *a* cselekvéssel

R^+ a tetszőleges állapotban elérhető legnagyobb jutalom optimista becslése

Az a cselekvés, amely felderítetlen területek *felé* vezet, nagyobb súlyt kap.

ε-mohóság: az ágens ε valószínűséggel véletlen cselekvést választ, ill. 1-ε valószínűséggel mohó

Boltzmann-felfedezési modell
egy a cselekvés megválasztásának valószínűsége egy s állapotban:

$$P(a,s) = \frac{e^{hasznosság(a,s)/T}}{\sum_{a'} e^{hasznosság(a',s)/T}}$$

a T „hőmérséklet” a két véglet között szabályoz.
Ha T → ∞ , akkor a választás tisztán (egyenletesen) véletlen,
ha T → 0 , akkor a választás mohó.

A cselekvés-érték függvény tanulása

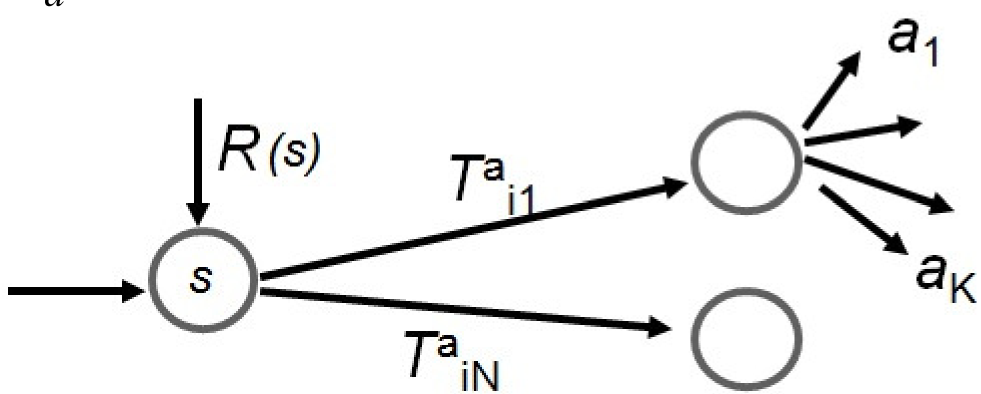
A cselekvés-érték függvény egy adott állapotban választott adott cselekvéshez egy várható hasznosságot rendel: **Q-érték**

$$U(s) = \max_a Q(a,s)$$

A Q-értékek fontossága:
a feltétel-cselekvés szabályokhoz hasonlóan lehetővé teszik a döntést **modell használata** nélkül,
ellentétben a feltétel-cselekvés szabályokkal, **közvetlenül** a jutalom visszacsatolásával **tanulhatók**.

Mint a hasznosság-értékeknél, felírhatunk egy kényszer egyenletet, amely **egyensúlyi állapotban**, amikor a **Q-értékek korrektek**, fenn kell álljon:

$$Q(a,s) = R(s) + \gamma \sum_{s'} T(s,a,s') \max_{a'} Q(a',s')$$



A cselekvés-érték függvény tanulása

Ezt az egyenletet közvetlenül felhasználhatjuk egy olyan iterációs folyamat módosítási egyenleteként, amely egy adott modell esetén a pontos Q-értékek számítását végzi.

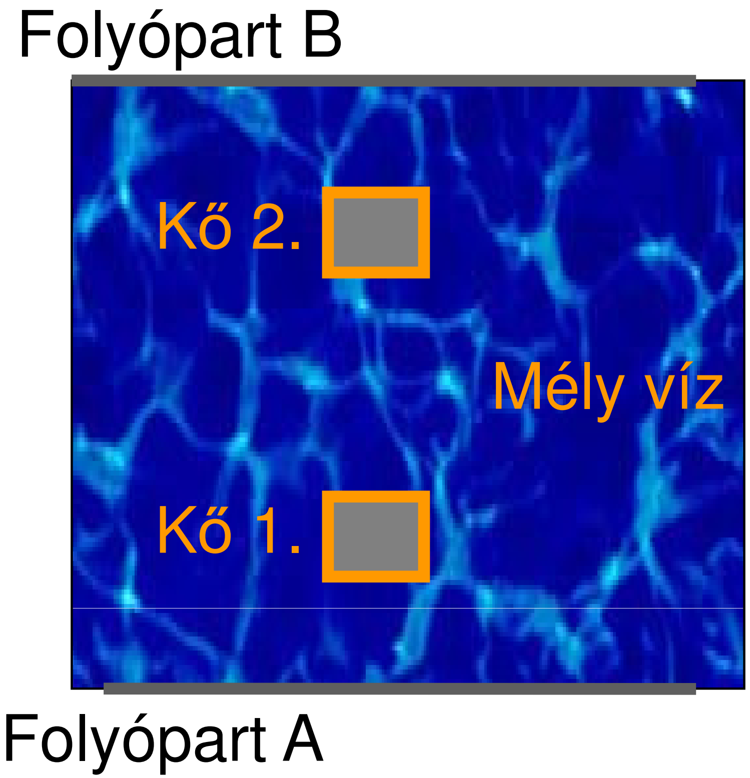
Az **időbeli különbség** viszont nem igényli a modell ismeretét. Az **IK** módszer **Q-tanulásának** módosítási egyenlete:

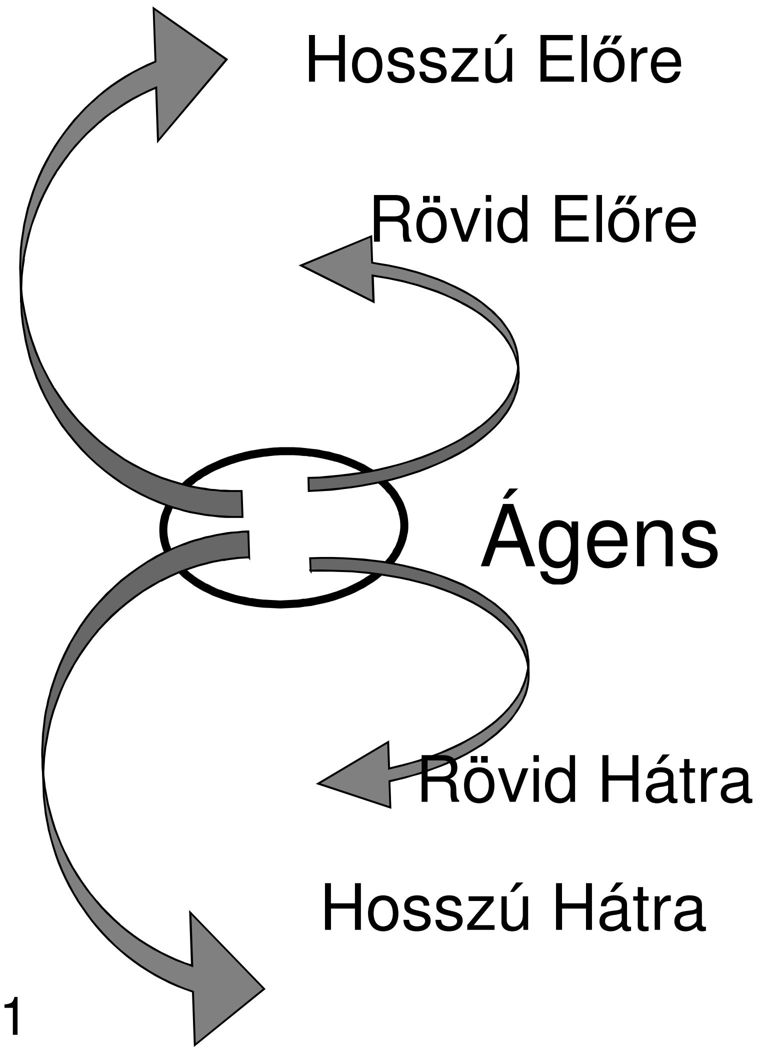
$$Q(a,s) \leftarrow Q(a,s) + \alpha \left(R(s) + \gamma \max_{a'} Q(a',s') - Q(a,s)\right)$$

SARSA Q-tanulás (State-Action-Reward-State-Action)

$$Q(a,s) \leftarrow Q(a,s) + \alpha [R(s) + \gamma Q(a',s') - Q(a,s)]$$

(a' megválasztása pl. Boltzmann felfedezési modellből)





Tanuló szekvencia:

RE→ HE→ RE→ +1 (száraz lábbal át)

RE→ HE→ HH→ HE→ HH→ RH→ HE→ -1

RE→ HE→ RH→ -1 (megfürdött)

…………

Part B

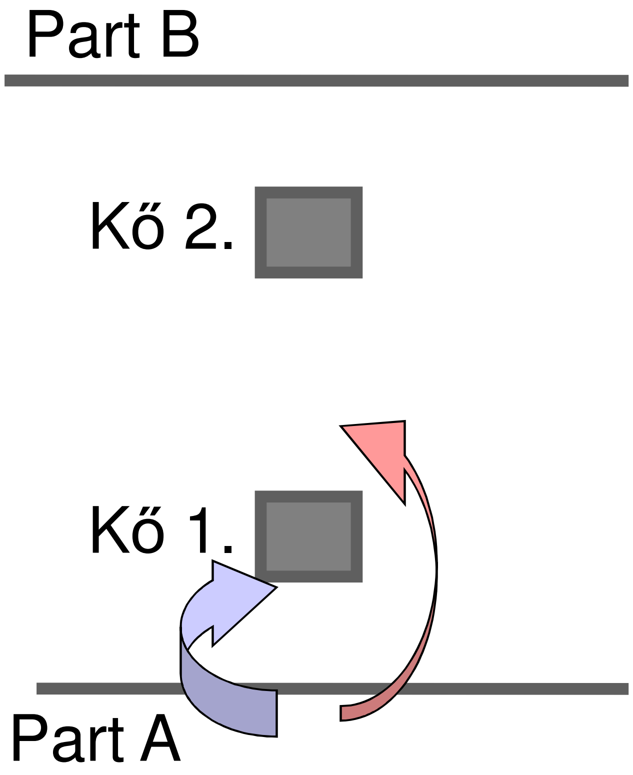
Kő 2.

$$Q(a,s) \leftarrow Q(a,s) + \alpha\left(R(s) + \gamma \max_{a'} Q(a',s') - Q(a,s)\right)$$

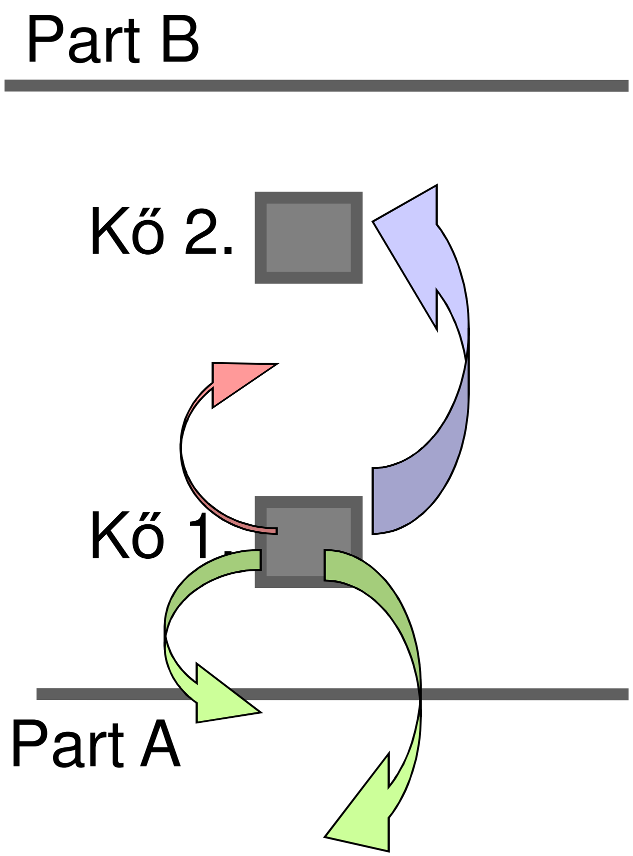
Kő 1.

Part A

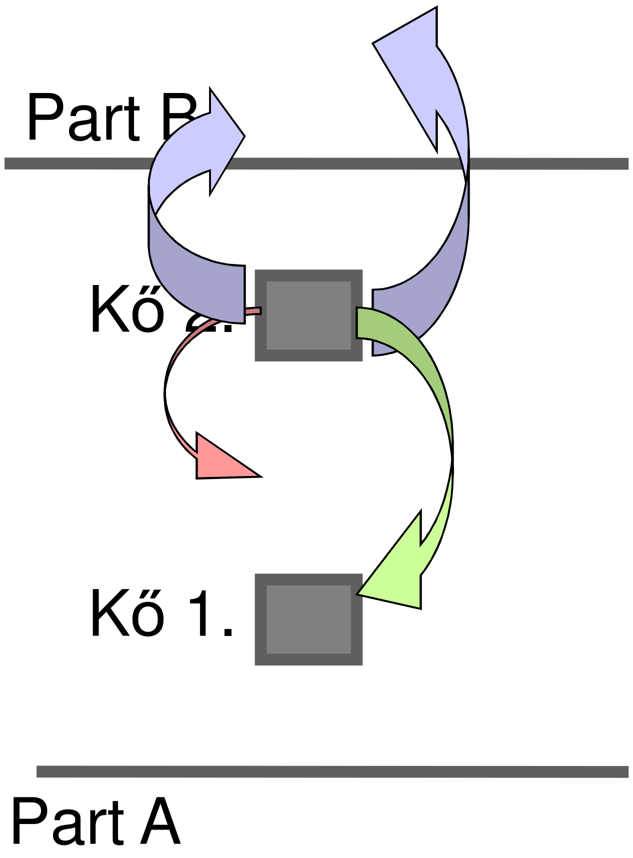
		Cselekvés			
		HH	RH	RE	HE
Állapot	Part A	0	0	-0.035	-0.877
	Kő 1.	-0.520	-0.550	-0.835	0.555
	Kő 2.	-0.204	-0.897	1.180	1.158



	HH	RH	RE	HE
Part A	0	0	-0.035	-0.877
Kő 1.	-0.520	-0.550	-0.835	0.555
Kő 2.	-0.204	-0.897	1.180	1.158



	HH	RH	RE	HE
Part A	0	0	-0.035	-0.877
Kő 1.	-0.520	-0.550	-0.835	0.555
Kő 2.	-0.204	-0.897	1.180	1.158



	HH	RH	RE	HE
Part A	0	0	-0.035	-0.877
Kő 1.	-0.520	-0.550	-0.835	0.555
Kő 2.	-0.204	-0.897	1.180	1.158

A megerősítéses tanulás általánosító képessége

Eddig: ágens által tanult hasznosság: **táblázatos** formában reprezentált
= **explicit reprezentáció**

Kis állapotterek: elfogadható, de a konvergencia idő és (ADP esetén) az iterációnkénti idő gyorsan nő a tér méretével.

Gondosan vezérelt közelítő ADP: 10000, vagy ennél is több.
De a való világhoz közelebb álló környezetek szóba se jöhetnek.
(a sakk stb. - állapottere 10^{50}-10^{120} nagyságrendű)
(az összes állapotot látni, hogy megtanuljunk játszani?!)

Egyetlen lehetőség: a függvény **implicit reprezentációja**:

Pl. egy játékban a táblaállás tulajdonságainak valamilyen halmaza $f_1,\dots f_n$.
Az állás becsült hasznosság-függvénye:

$$\hat{U}_\theta(s) = \theta_1 f_1(s) + \theta_2 f_2(s) + \dots \theta_n f_n(s)$$

C. Shannon, **Programming a Computer for Playing Chess**, Philosophical Magazine, 7th series, 41, no. 314 (March 1950): 256-75
A.L. Samuel, **Some Studies in Machine Learning Using the Game of Checkers. B.II-Recent Progress**, IBM. J. 3, 211-229 (1959), kb. 20 tulajdonság
Deep Blue (Feng-Hsiung Hsu) kb. 8000 tulajdonság, spec. sakk-csip

A hasznosság-függvény *n* értékkel jellemezhető, pl. 10^{120} érték helyett. Egy átlagos sakk kiértékelő függvény kb. 10 súllyal épül fel, tehát *hatalmas* tömörítést értünk el.

Az implicit reprezentáció által elért tömörítés teszi lehetővé, hogy a tanuló ágens általánosítani tudjon a már látott állapotokról az eddigiekben nem látottakra.

Az implicit reprezentáció legfontosabb aspektusa nem az, hogy kevesebb helyet foglal, hanem az, hogy lehetővé teszi a **bemeneti állapotok induktív általánosítását**. Az olyan módszerekről, amelyek ilyen reprezentációt tanulnak, azt mondjuk, hogy **bemeneti általánosítást** végeznek.

4 x 3 világ

$$E_j(s) = \frac{1}{2}\left(\hat{U}_\theta(s) - u_j(s)\right)^2$$

$$\hat{U}_\theta(x, y) = \theta_0 + \theta_1 x + \theta_2 y$$

Jósolt és kísérleti tényleges

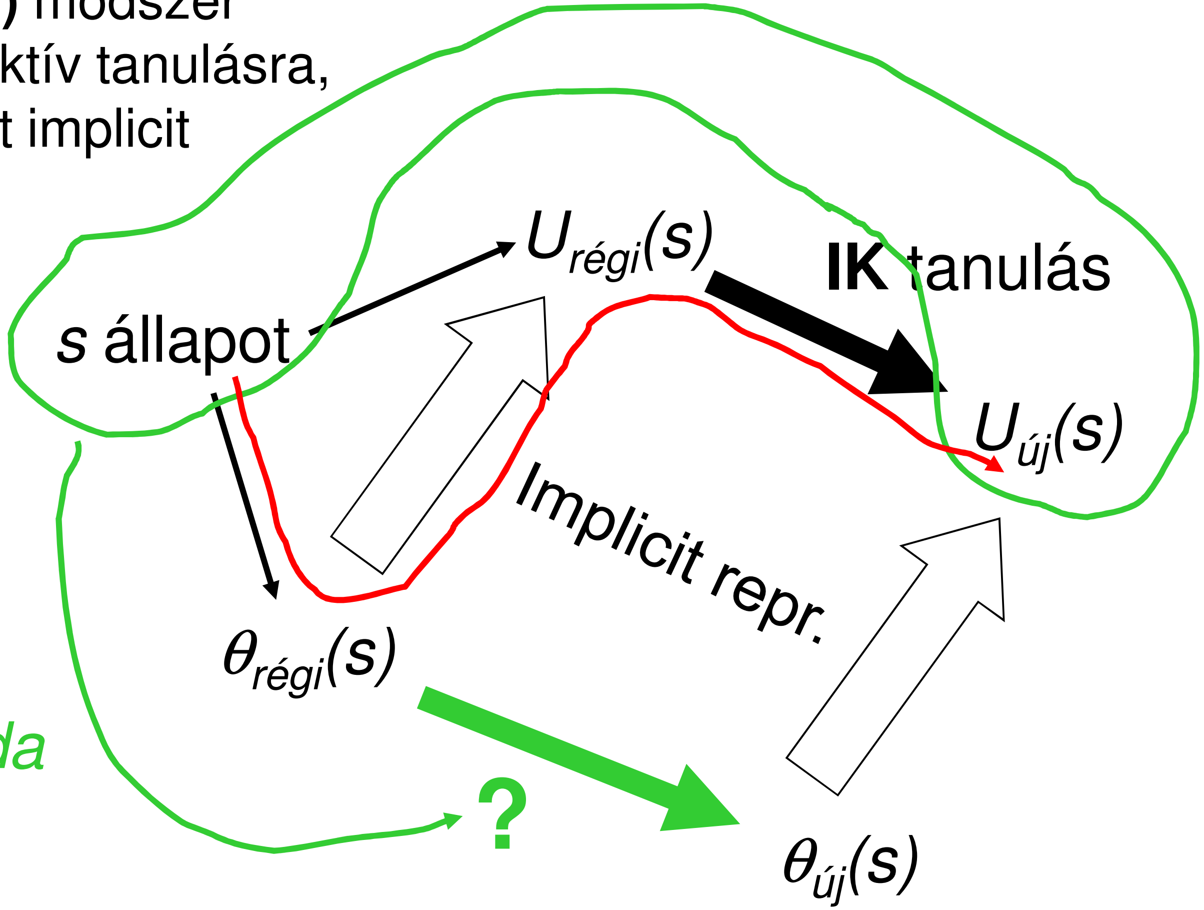
$$\theta_i \leftarrow \theta_i - \alpha \frac{\partial E_j(s)}{\partial \theta_i} = \theta_i + \alpha\left(u_j(s) - \hat{U}_\theta(s)\right)\frac{\partial \hat{U}_\theta(s)}{\partial \theta_i}$$

$$\theta_0 \leftarrow \theta_0 + \alpha\left(u_j(s) - \hat{U}_\theta(s)\right)$$

$$\theta_1 \leftarrow \theta_1 + \alpha\left(u_j(s) - \hat{U}_\theta(s)\right)x$$

$$\theta_2 \leftarrow \theta_2 + \alpha\left(u_j(s) - \hat{U}_\theta(s)\right)y$$

Az **IK (időbeli különbség)** módszer szintén alkalmazható induktív tanulásra, ha az *U*, vagy *Q* táblázatot implicit reprezentációra cseréltük (pl. neurális háló).

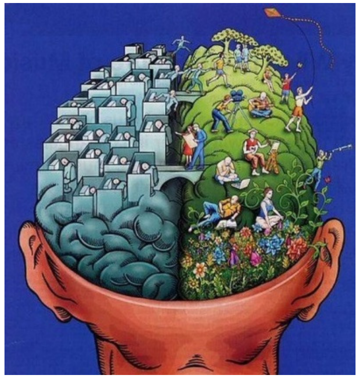


U/*Q* tanuló gradiensek

$$\theta_i \leftarrow \theta_i + \alpha\left[R(s) + \gamma \hat{U}_\theta(s') - \hat{U}_\theta(s)\right]\frac{\partial \hat{U}_\theta(s)}{\partial \theta_i}$$

$$\theta_i \leftarrow \theta_i + \alpha\left[R(s) + \gamma \max_{a'} \hat{Q}_\theta(a',s') - \hat{Q}_\theta(a,s)\right]\frac{\partial \hat{Q}_\theta(a,s)}{\partial \theta_i}$$

Budapesti Műszaki és Gazdaságtudományi Egyetem
Méréstechnika és Információs rendszerek Tanszék



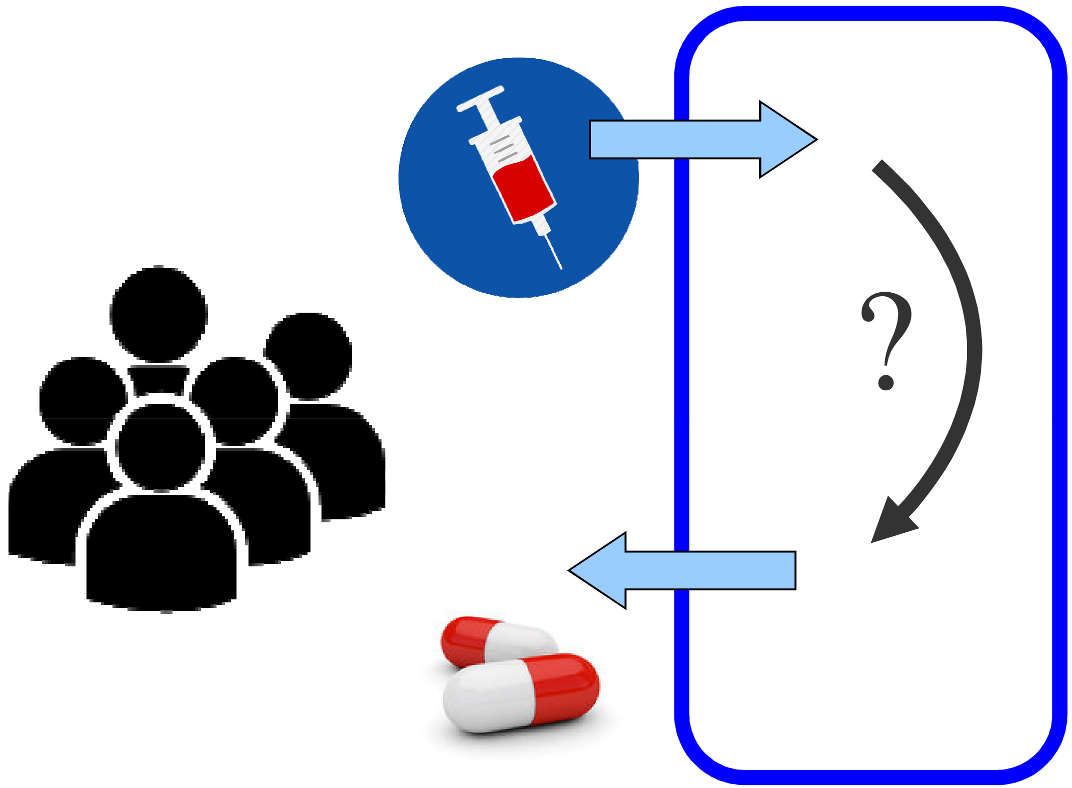
Tanulás, döntés és minősítés

Előadó: Dr. Hullám Gábor

Előadás anyaga: Dr. Dobrowiecki Tadeusz
Dr. Pataki Béla

MŰEGYETEM 1782

Az észleléseink alapján térjünk át injekcióról tablettára!
Ez jó döntés?



Jó döntést szeretnénk hozni.

De mi a jó döntés?

- <u>Általában nincs tökéletesen jó döntés</u>, csak az egyik jobb, a másik rosszabb!
- Ahhoz, hogy össze tudjunk hasonlítani két alternatívát (döntési módszert), kell legyen a döntés jóságát jellemző (skalár) mérőszámunk
- **Először a döntések minősítésével, értékelésével foglalkozunk**
- Alapvetően bináris (jó/rossz, igaz/hamis, beteg/egészséges, ártatlan/bűnös) döntéseken mutatjuk be a módszereket

A döntések értékelése: melyik a jó döntés?

Kétértékű függvény ⇒ osztályozás = bináris döntés

igazi állapot f(x) tény	feltételezett állapot h(x) hipotézis, döntés	helyzet
paciens **beteg** $f(x) = \mathbf{I}$	**felismerjük** $h(x) = \mathbf{I}$	kezeljük, helyesen teszünk **Valós Pozitív, True Positive TP**
paciens **egészséges** $f(x) = \mathbf{H}$	**felismerjük** $h(x) = \mathbf{H}$	nem kezeljük, most is jól teszünk, **Valós Negatív, True Negative TN**
paciens **beteg** $f(x) = \mathbf{I}$	**nem ismerjük fel** $h(x) = \mathbf{H}$	**nem kezeljük, nem járunk el jól, Hamis Negatív, False Negative FN,** („elnézett támadás”, **2. tip. hiba**)
paciens **egészséges** $f(x) = \mathbf{H}$	**nem ismerjük fel** $h(x) = \mathbf{I}$	**kezeljük fölöslegesen, nem járunk el jól, Hamis Pozitív, False Positive FP,** („hamis riadó”, **1. tip. hiba**)

true positive rate (**TPR**)
valódi pozitív arány, érzékenység
(hit rate, recall, sensitivity)

TPR = TP/P = TP / (TP+FN)

true negative rate (**TNR**)
valódi negatív arány, specificitás
(specificity)

TNR = TN/N = TN / (TN+FP)

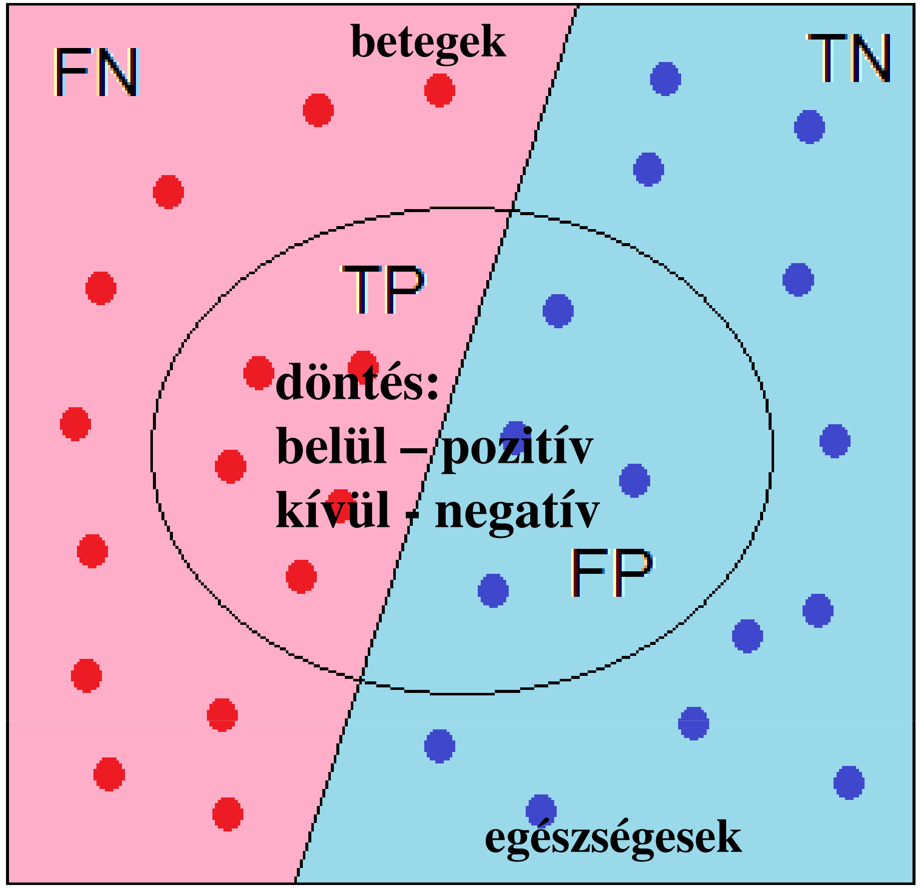
false positive rate (**FPR**)
hamis pozitív arány
(false alarm rate, fall-out)

FPR = FP/N = FP / (FP+TN)

és sok egyéb más mérőszám …

<u>Az ábrán vázolt esetben:</u>

Döntéseink: TP = 5, TN = 11,
FP = 4, FN = 10
Tények: P = 15, N = 15

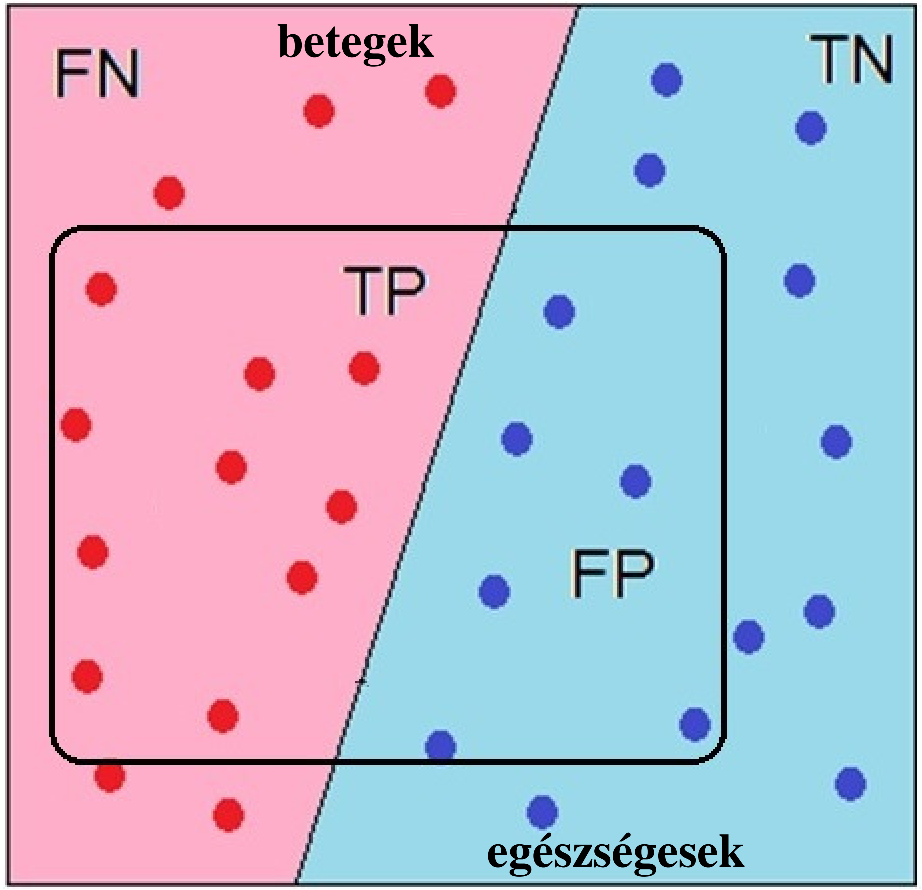


TPR = TP / (TP+FN) = 5/15 = .33

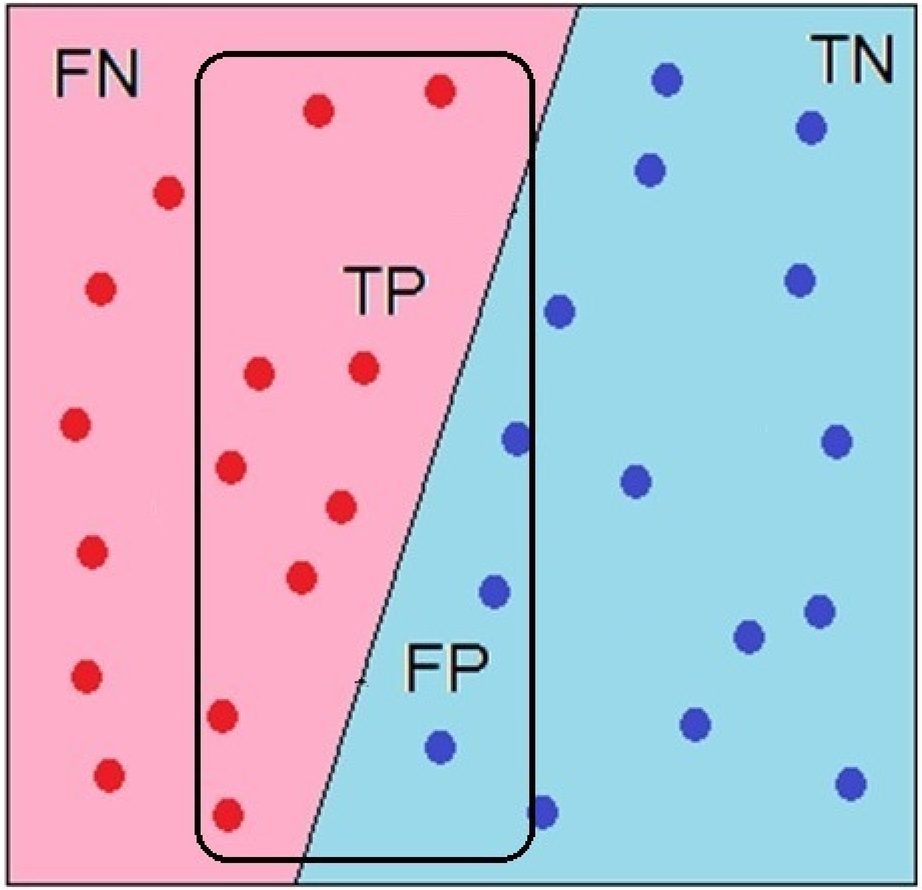
FPR = FP / (FP+TN) = 4/15 = .26

ACC = (TP+TN)/(P+N) = 1/2 = .5

Néhány más javaslatra is kipróbáltuk… **Melyik jobb?**

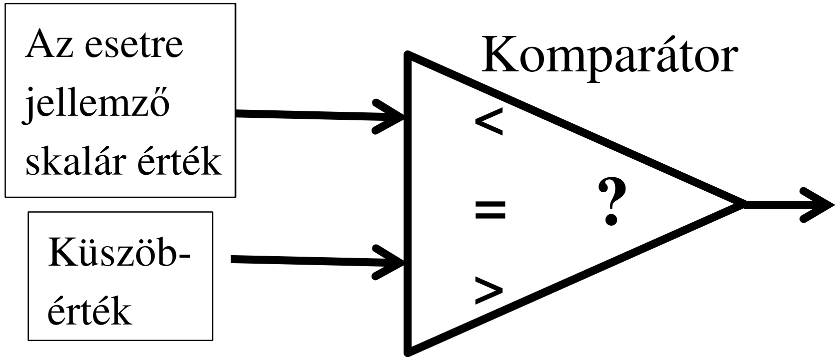


TP = 10, TN = 9, FP = 6, FN = 5



TP = 9, TN = 12, FP = 3, FN = 6

A döntésünk modellje



például (lehetne fordítva is):

ha **>** , akkor kimenet=1

pozitív esetnek vesszük

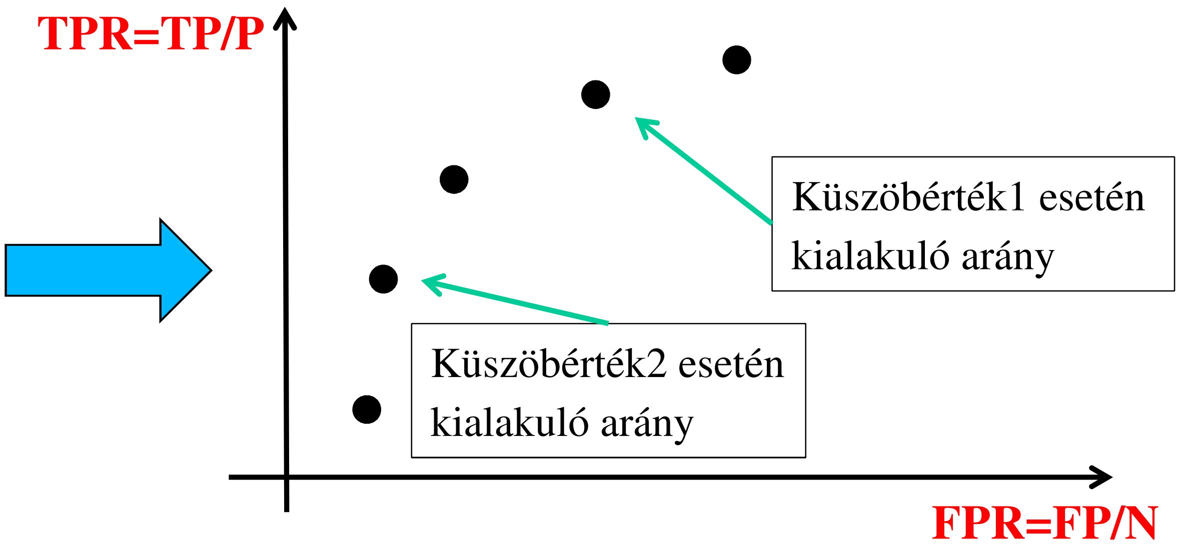
ha **<=**, akkor kimenet= 0

negatív esetnek vesszük

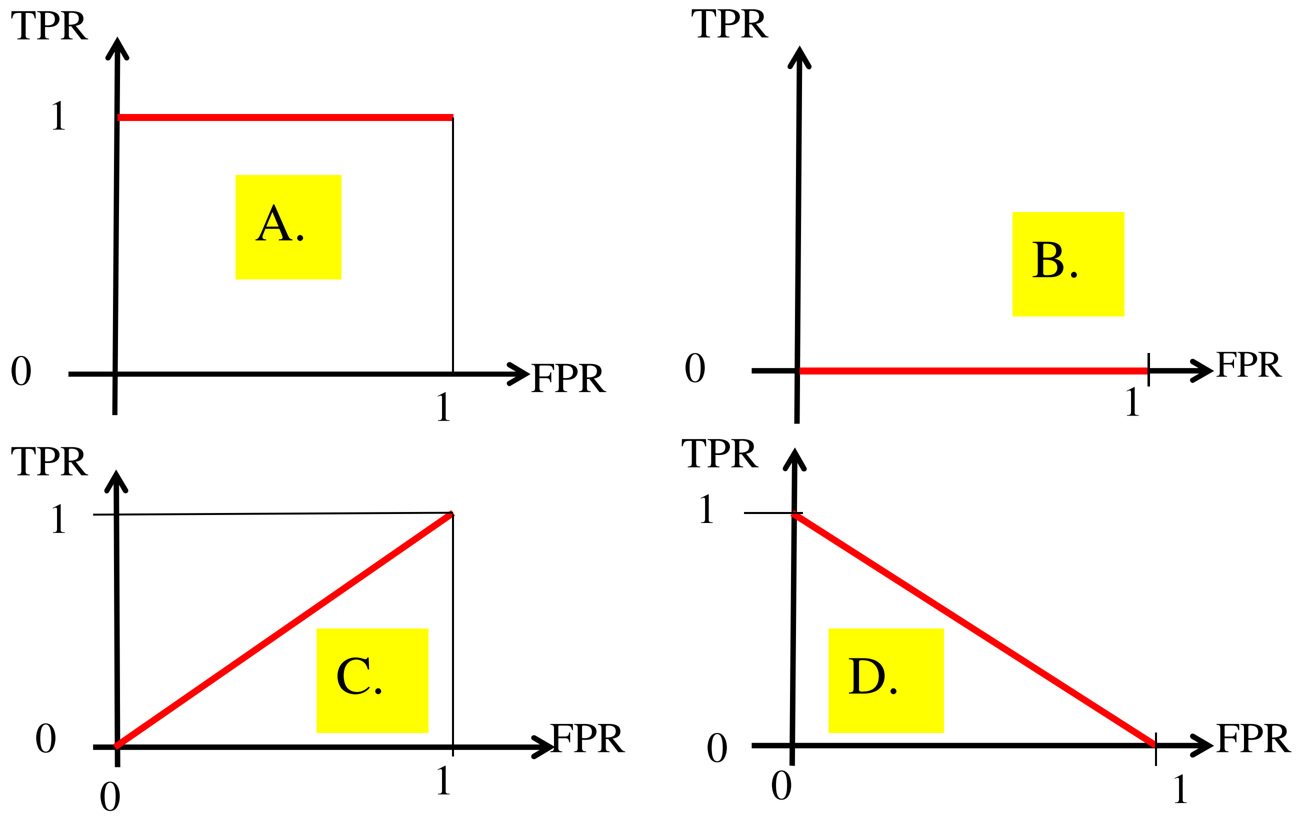
Példa:

Ha a mért testhőmérséklet 38°C küszöbnél nagyobb, akkor láz (betegség), ha kisebb vagy egyenlő, akkor nem lázas (egészséges).

Ez a ROC görbe (Receiver Operating Characteristic curve)

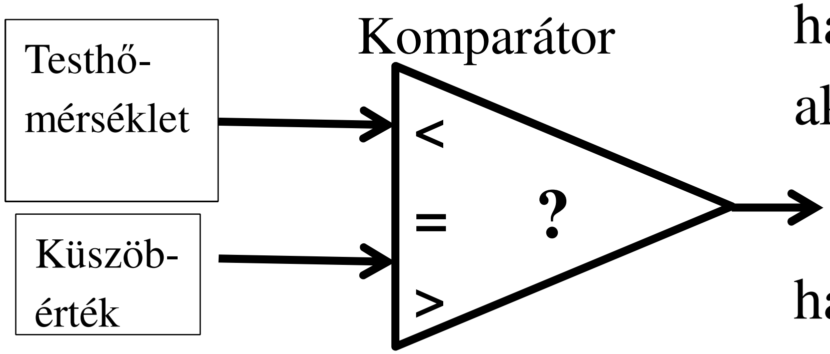


Ha érdektelen (irreleváns) paraméter alapján döntünk, akkor nagyszámú mintán melyik ROC görbe alakul ki?



Gyk. az ábrákon piros vonallal jelöltük a ROC görbét

Mi van, ha egy érdektelen (irreleváns) paraméterre alapozzuk a döntésünket?



ha hőmérséklet **>** küszöb,
akkor szőke hajú

ha hőmérséklet **<=** küszöb,
akkor barna hajú

Megvizsgáltunk 6396 embert, köztük 2460 volt szőke, a többi barna.

Testhőm. [ºC]	35,0… 36,0	36,0… 37,0	37,0… 38,0	38,0… 39,0	39,0… 40,0	40,0… 41,0	41,0… 42,0	42,0… 43,0	43,0… 44,0
SZŐKE	5	1000	800	600	40	10	5	0	0
BARNA	8	1600	1280	960	64	16	8	0	0

küszöb1

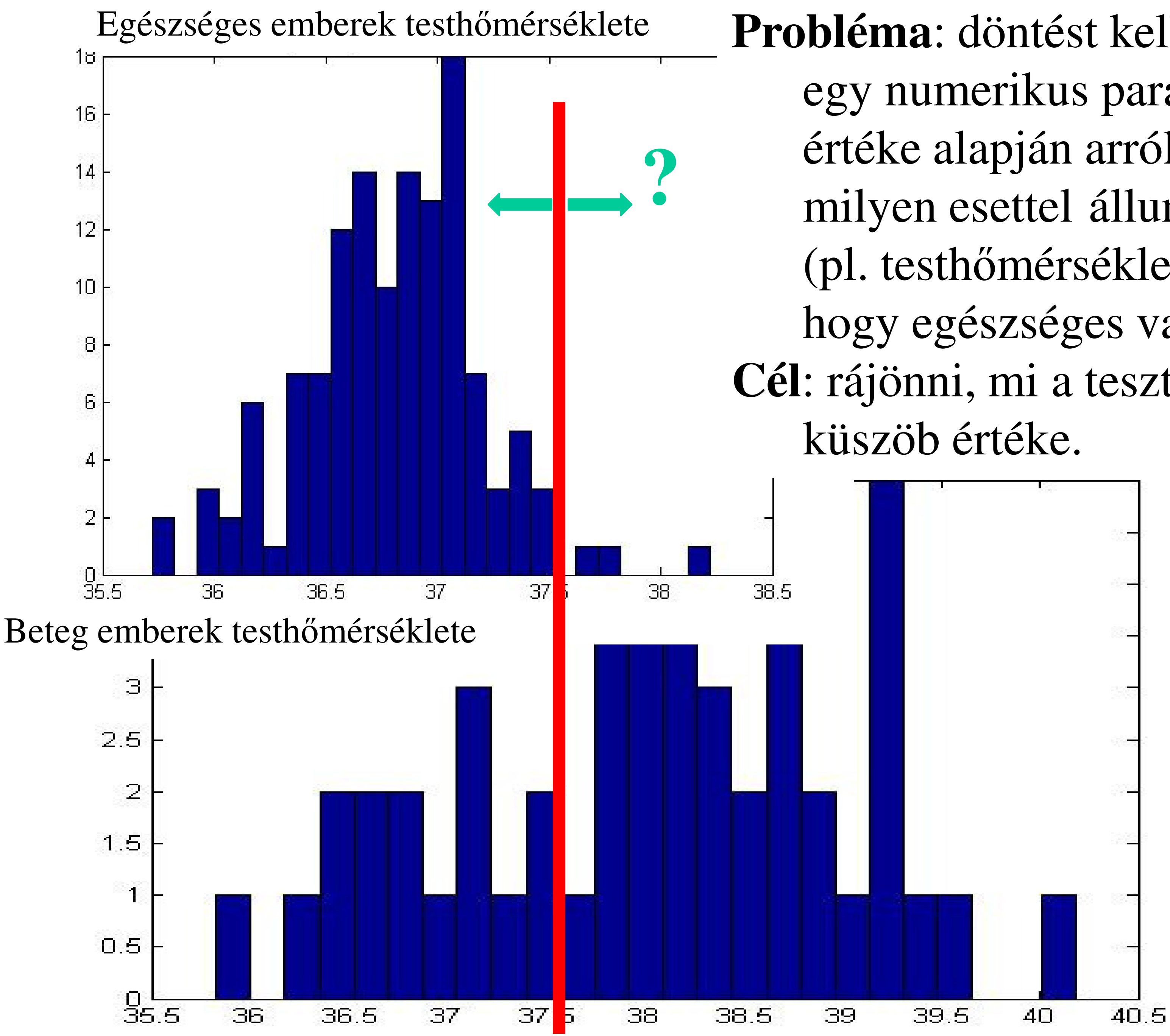
küszöb2

• • • •

küszöb9

küszöb10

Milyen ROC görbe alakul ki?

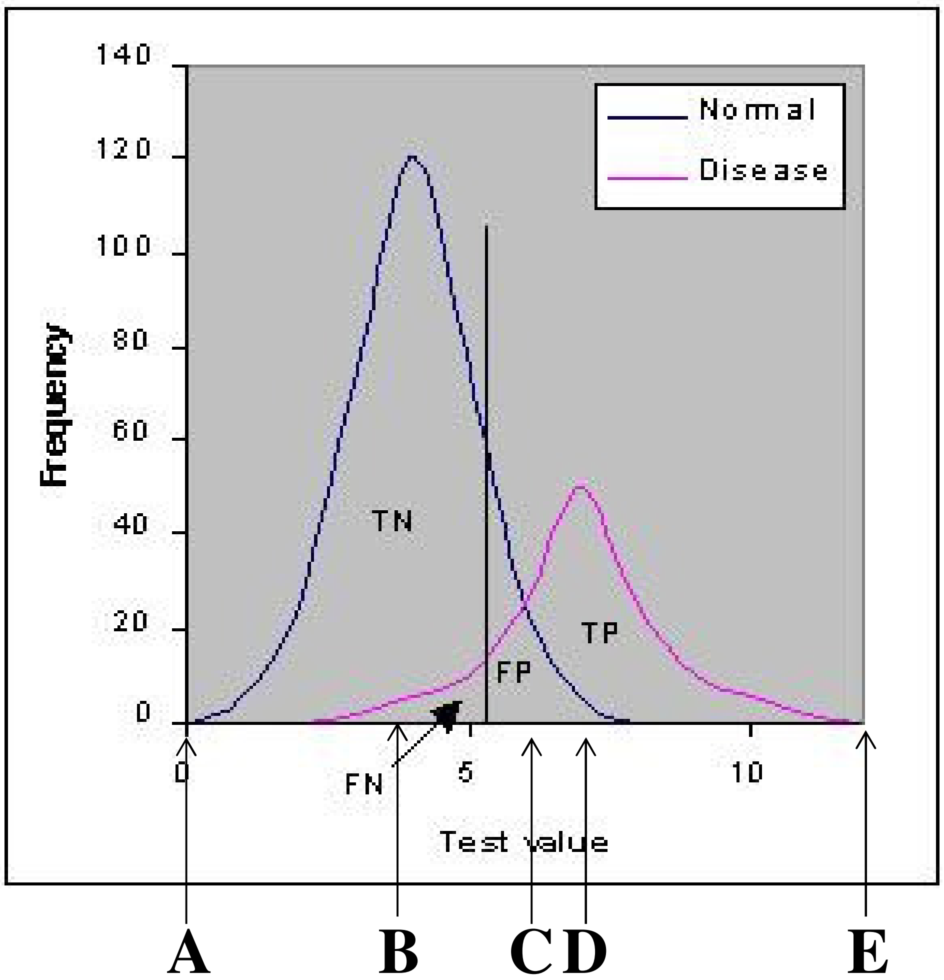


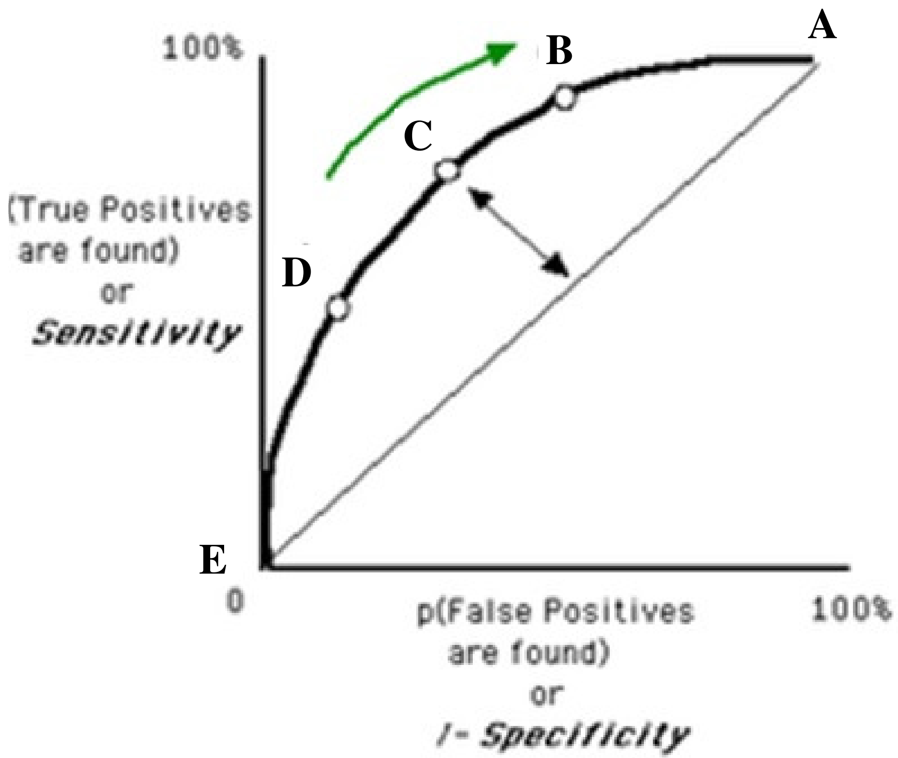
Probléma: döntést kell hoznunk, egy numerikus paraméter értéke alapján arról, hogy milyen esettel állunk szemben. (pl. testhőmérséklet alapján, hogy egészséges vagy beteg?)

Cél: rájönni, mi a teszt (optimális) küszöb értéke.

ROC: Vevő működési karakterisztika

Receiver Operating Characteristic





A küszöb fölött betegnek tekintjük, alatta egészségesnek

Hiba (confusion) mátrix

true positive rate (**TPR**)
TPR = TP/P = TP / (TP+FN)

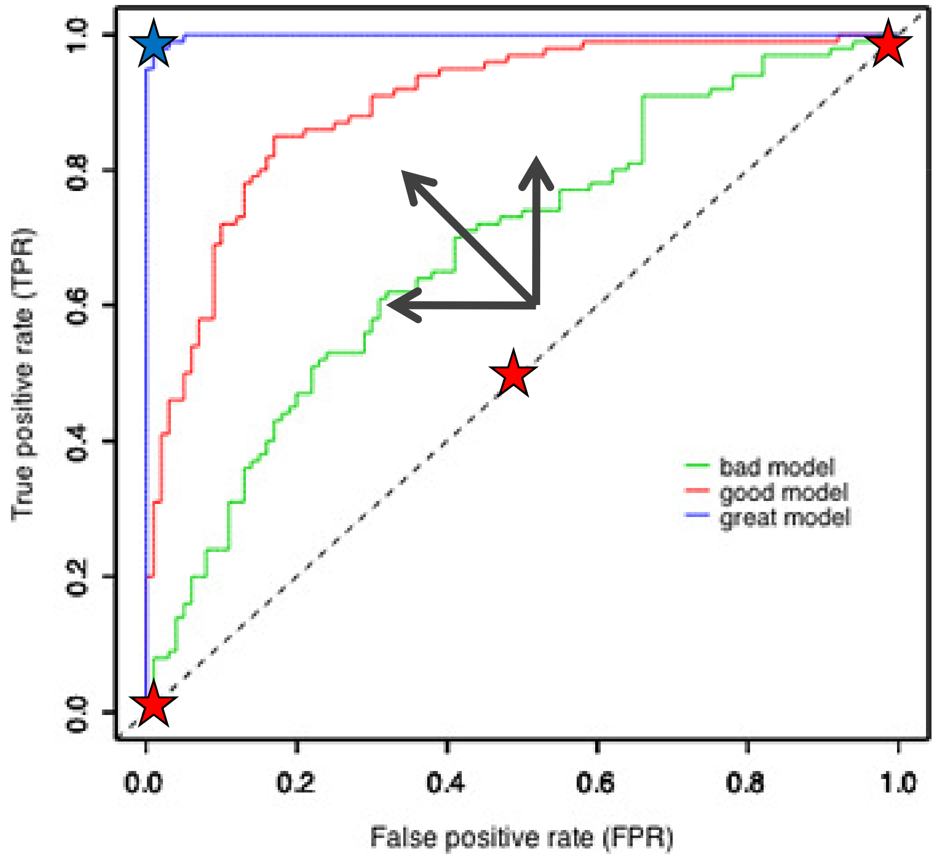
false positive rate (**FPR**)
FPR = FP/N = FP / (FP+TN)

ROC: Vevő működési karakterisztika
Receiver **O**perating **C**haracteristic (curve)

AUC: Görbe alatti terület
Area **U**nder **C**urve

Tények

Döntés	Beteg	Egész-séges
Beteg	TP	FP
Egész-séges	FN	TN



Mi a helyzet, ha a tévedéseknek különböző a költsége?

1. **típusú hiba hamis negatív (FN)**
   – azt hisszük egészséges (H0), pedig beteg (T1)
   - azt hisszük barát pedig ellenség
   költség: C_{01} (későn vesszük észre a bajt, bonyolult műtét vagy kezelés kell, azt hittük barát, pedig ellenség és lebombázza a hidat/gyárat)

2. **típusú hiba hamis pozitív (FP)**
   – azt hisszük beteg (H1), pedig egészséges (T0)
   - azt hisszük ellenség pedig barát
   költség: C_{10} (feleslegesen kezeljük vagy vizsgáljuk az egészségest, a baráti gépet lelőjük vagy vadászrepülőket ezt-azt küldünk rá)

legtöbbször a hamis negatív a rosszabb: $C_{01} >> C_{10}$

Ráadásul a helyes döntéseknek is van költsége!

valós negatív (TN)

– azt hisszük egészséges (H0), valóban az (T0)

- azt hisszük barát, és tényleg

költség: $\mathbf{C_{00}}$ vizsgálatot kell végeznünk

– orvosi szűrés, program teszt, katonai radar

valós pozitív (TP)

– azt hisszük beteg (H1), valóban az (T1)

- azt hisszük ellenség, és tényleg

költség: $\mathbf{C_{11}}$ vizsgálatot kell végeznünk + kezelnünk kell

– orvosi szűrés, program teszt, katonai radar

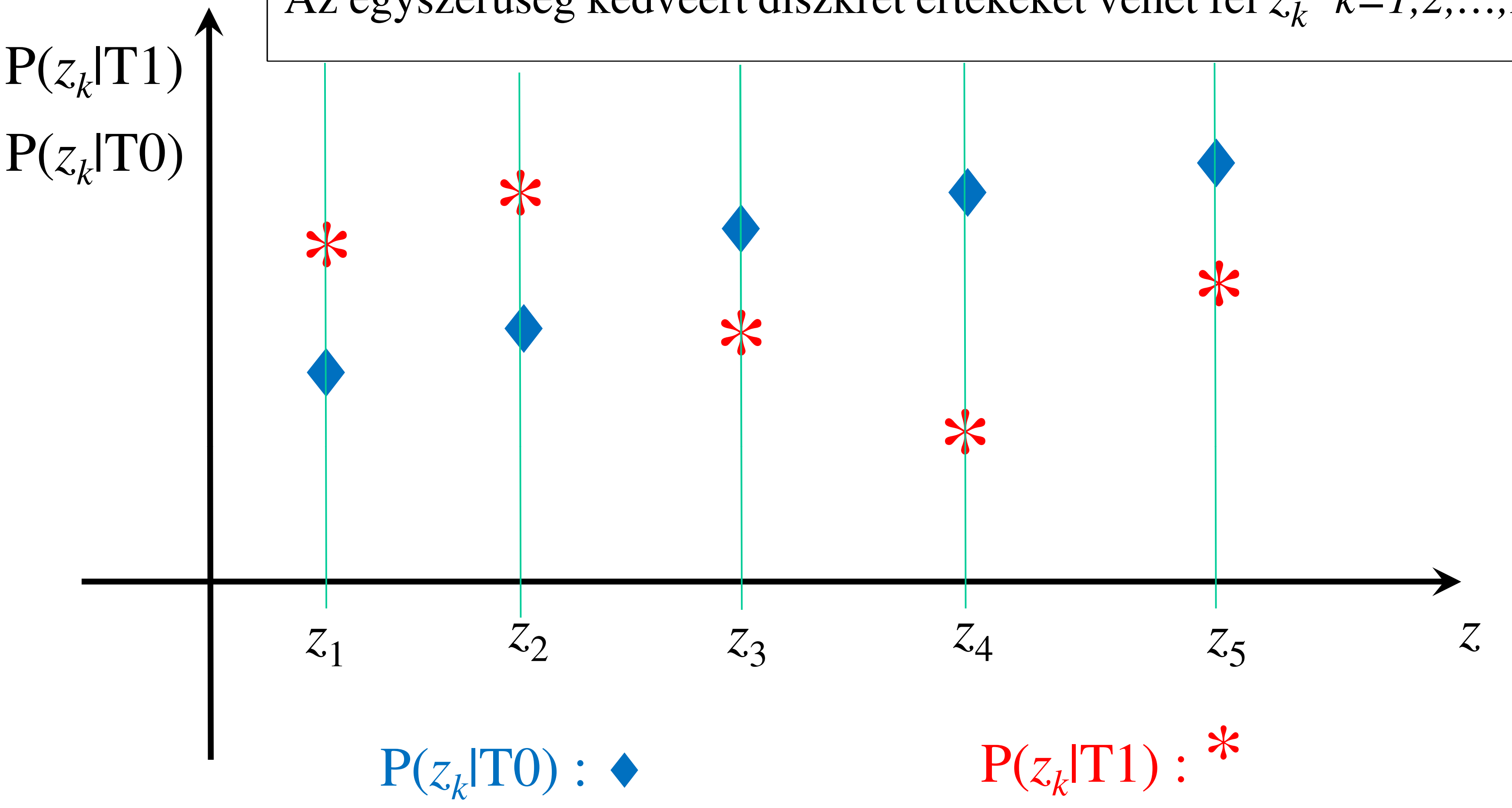
\+ kezelés, légvédelmi rakéta, hibajavítás

tehát négyféle költség befolyásolja a döntést:

$\mathbf{C_{00}}$, $\mathbf{C_{11}}$, $\mathbf{C_{01}}$, $\mathbf{C_{10}}$

Skalár mért vagy származtatott paraméter, ami alapján döntünk : z

Az egyszerűség kedvéért diszkrét értékeket vehet fel z_k $k=1,2,\ldots,K$



Hogyan becsülhetjük meg ezeket az arányokat?
Egy tesztet végzünk (vagy az eddigi adatokból összegyűjtünk egy megfelelően nagy mintát).

Tudnunk kell, hogy milyen arányban produkálják a negatív és pozitív esetek az egyes lehetséges z értékeket.
Ezért tesztelünk N_T esetet, ezek közül N_N tartozik a negatív, N_P a pozitív csoportba ($N_N + N_P = N_T$) Fontos: a tesztben sokszor nem ugyanaz a negatív és pozitív esetek aránya, mint a teljes népességben (utóbbiak később: P0 és P1).

k	**1**	**2**	**3**	**4**	**5**	**Σ**
z_k	z_1	z_2	z_3	z_4	z_5	
z_k-t produkál az negatív esetek közül	N_{N1} (pl. 1200)	N_{N2} (pl. 3100)	N_{N3} (pl. 4500)	N_{N4} (pl. 2700)	N_{N5} (pl. 1820)	$N_{N1}+N_{N2}+\ldots$ $\ldots+N_{N5} =$ $= N_N$ (N_N=13320)
$P(z_k\|T0)$	$\frac{N_{N1}}{N_N}$ $(=\frac{1200}{13320})$	$\frac{N_{N2}}{N_N}$	$\frac{N_{N3}}{N_N}$	$\frac{N_{N4}}{N_N}$	$\frac{N_{N5}}{N_N}$	1
z_k-t produkál a pozitív esetek közül	N_{P1}	N_{P2}	N_{P3}	N_{P4}	N_{P5}	$N_{P1}+N_{P2}+\ldots$ $\ldots+N_{P5} =$ $= N_P$
$P(z_k\|T1)$	$\frac{N_{P1}}{N_P}$	$\frac{N_{P2}}{N_P}$	$\frac{N_{P3}}{N_1}$	$\frac{N_{P4}}{N_P}$	$\frac{N_{P5}}{N_P}$	1

Ha a populáció N elemű, és ebből a negatívok aránya $P_0 = \frac{N_0}{N}$, akkor a negatív esetek száma $N_0 = P_0 \cdot N$.
Az N_0 negatív esetből z_k értéket produkál $N_0 \cdot \mathrm{P}(z_k|\mathrm{T0})$ eset.

A pozitív esetek aránya $P_1 = \frac{N_1}{N}$, akkor a pozitív esetek száma $N_1 = P_1 \cdot N$.
Az N_1 pozitív esetből z_k értéket produkál $N_1 \cdot \mathrm{P}(z_k|\mathrm{T1})$ eset.

Ha úgy döntünk, hogy z_k esetén pozitívnak véleményezzük az ismeretlen esetet, akkor a költségünk:

$$\mathrm{K1} = \mathrm{C}_{10} \cdot \mathrm{N}_0 \cdot \mathrm{P}(z_k|\mathrm{T0}) + \mathrm{C}_{11} \cdot N_1 \cdot \mathrm{P}(z_k|\mathrm{T1}) =$$
$$= \mathrm{C}_{10} \cdot P_0 \cdot N \cdot \mathrm{P}(z_k|\mathrm{T0}) + \mathrm{C}_{11} \cdot P_1 \cdot N \cdot \mathrm{P}(z_k|\mathrm{T1})$$

Ha úgy döntünk, hogy z_k esetén negatívnak véleményezzük az ismeretlen esetet, akkor a költségünk:

$$\mathrm{K2} = \mathrm{C}_{00} \cdot \mathrm{N}_0 \cdot \mathrm{P}(z_k|\mathrm{T0}) + \mathrm{C}_{01} \cdot N_1 \cdot \mathrm{P}(z_k|\mathrm{T1}) =$$
$$= \mathrm{C}_{00} \cdot P_0 \cdot N \cdot \mathrm{P}(z_k|\mathrm{T0}) + \mathrm{C}_{01} \cdot P_1 \cdot N \cdot \mathrm{P}(z_k|\mathrm{T1})$$

A kérdés tehát az, hogy melyik esetben kisebb a költség?

Ha K1 < K2 azaz

$$\mathbf{C_{10} \cdot \boldsymbol{P_0} \cdot \boldsymbol{N} \cdot P(\boldsymbol{z_k}|T0) + C_{11} \cdot \boldsymbol{P_1} \cdot \boldsymbol{N} \cdot P(\boldsymbol{z_k}|T1) < C_{00} \cdot \boldsymbol{P_0} \cdot \boldsymbol{N} \cdot P(\boldsymbol{z_k}|T0) + C_{01} \cdot \boldsymbol{P_1} \cdot \boldsymbol{N} \cdot P(\boldsymbol{z_k}|T1)}$$

átrendezve:

$$\mathbf{(C_{10} - C_{00}) \cdot \boldsymbol{P_0} \cdot \boldsymbol{N} \cdot P(\boldsymbol{z_k}|T0) < (C_{01} - C_{01}) \cdot \boldsymbol{P_1} \cdot \boldsymbol{N} \cdot P(\boldsymbol{z_k}|T1)}$$

egyszerűsítve:

$$\mathbf{(C_{10} - C_{00}) \cdot \boldsymbol{P_0} \cdot P(\boldsymbol{z_k}|T0) < (C_{01} - C_{01}) \cdot \boldsymbol{P_1} \cdot P(\boldsymbol{z_k}|T1)}$$

akkor minden ilyen esetet **(**z_k-t produkál**)** **pozitívnak** veszünk.

****

Ha K1 > K2, akkor minden ilyen esetet **(**z_k-t produkál**)** **negatívnak** érdemes tekinteni.

****

A döntésünk nagy mértékben azon múlik, hogy jól becsüljük-e meg az egyes költségeket!!!
Ha C_{01} nagy, ettől félünk, akkor egész más alakul ki, mint ha C_{10}-tól félünk, azt vesszük nagynak.

Például a jog: attól félünk jobban , hogy egy ártatlant elítélnek, vagy attól, hogy egy bűnös megússza (esetleg később újra elköveti)?

Például a terrorista kérdés: attól félünk, hogy valahogy besurran egy terrorista, vagy attól, hogy jogtalanul vádolunk valakit?

Például az egészségügy: attól félünk-e jobban, hogy valakinek nem vesszük észre a betegségét, vagy attól, hogy feleslegesen ijesztgetjük, visszarendeljük vizsgálatra?

A következőkben olyan módszereket vizsgálunk, amikor minták, **mintapéldák alapján akarjuk kialakítani a döntési rendszerünket**.

Nagyon gyakran a számítógép tanulása azt jelenti, hogy van egy csomó mintapéldánk, és ez hordozza az információt. Nincsenek előre kialakított szabályaink a feladatra, csak a mintáink. Persze ilyenkor is valamilyen struktúrában igyekszünk felhasználni a minták hordozta tudást.

Tehát a módszerünk – tehát az, hogy hogyan használjuk fel a mintákat – bizonyos mértékben problémafüggetlen.

A hétköznapi életben nagyon gyakori a minta alapján való tanulás, a klasszikus számítógépes megoldásokban a humán fejlesztő szabályokat alkotott, és ezeket programozta be.

Összefoglalva: hogyan lehet egy döntést (ágensfüggvényt) megvalósítani? A következőkben a (2)-vel fogllakozunk

(1) Analitikus tervezés:
Begyűjteni az analitikus (fizikai, kémiai stb. összefüggésekből felépített) modelleket az adott problémára.
Megtervezni analitikusan a konkrét mechanizmust, és azt algoritmusként implementálni.

(2) Megtervezni a tanulás mechanizmusát, és azt algoritmusként implementálni.
Majd alkalmazásával megtanulni vele a minták alapján a döntés tényleges mechanizmusát, és azt algoritmusként implementálni.

Tanulás alapvető fajtái:

felügyelt tanulás egy komponensnek mind a bemenetét, mind a kimenetét észlelni tudjuk (bemeneti minta + kívánt válasz)

megerősítéses tanulás az ágens az általa végrehajtott tevékenység csak bizonyos értékelését kapja meg, esetleg nem is minden lépésben (jutalom, büntetés, **megerősítés**)

felügyelet nélküli tanulás semmilyen információ sem áll rendelkezésünkre a helyes kimenetről (az észlelések közötti összefüggések tanulása)

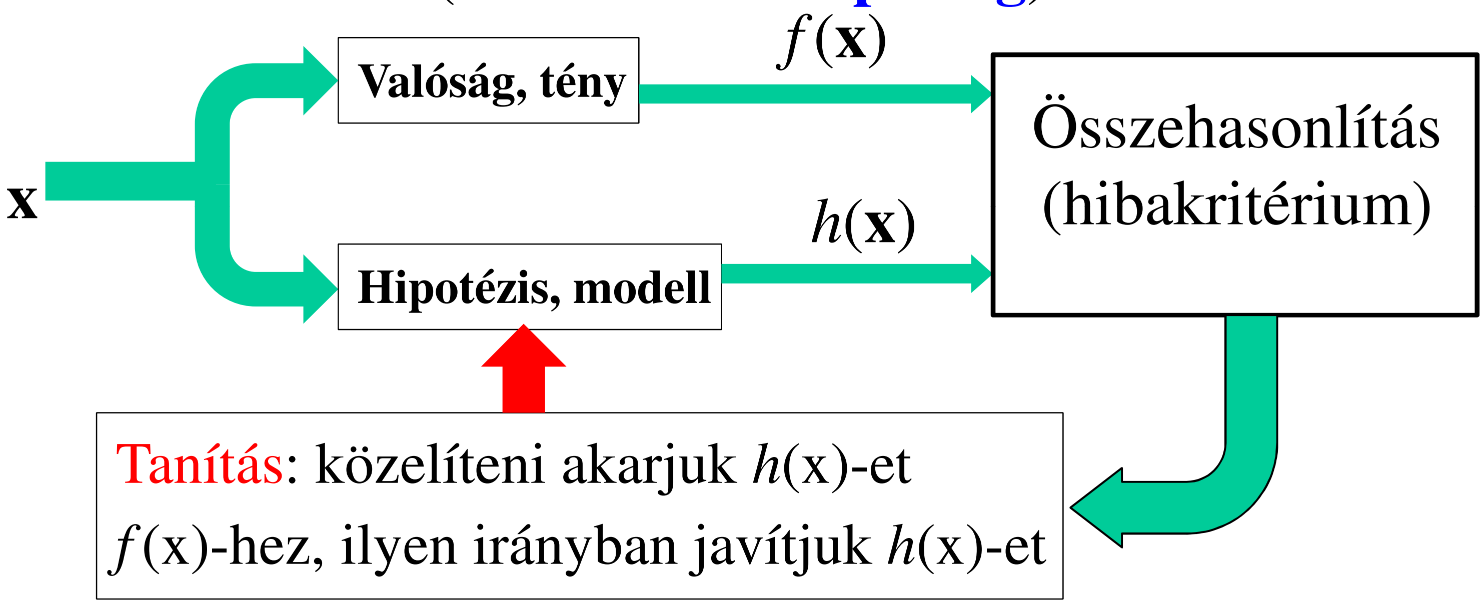
Felügyelt tanulás (pl. induktív következtetés):

tanulási példa: $(\mathbf{x}, f(\mathbf{x}))$ adatpár, ahol $f(\mathbf{x})$ ismeretlen

tanulás célja: $f(\mathbf{x})$ értelmes közelítése egy $h(\mathbf{x})$ hipotézissel

$h(\mathbf{x}) = f(\mathbf{x})$, $\mathbf{x}$ – ismert példákon

$h(\mathbf{x}') \cong f(\mathbf{x}')$, **$\mathbf{x}'$ – a tanulás közben még nem látott eset (általánosító képesség)**



Feladat: *az ismeretlen **f**-re vonatkozó példák egy halmaza alapján (**tanítóhalmaz**), adjon meg egy olyan **h** függvényt (**hipotézist**), amely tulajdonságaiban jól közelíti az **f**-et (amit **teszthalmazon** verifikálunk).*

Kifejezőképesség és hatékonyság

A legfontosabb döntés a tanuló ágens tervezője számára:
mi legyen a kívánt függvény (és a hipotézisek) **reprezentációja**?

Kompromisszum: (cél: legyen tömör és jól általánosító)
a **kifejezőképesség** és a **hatékonyság** között.

Kifejezőképesség: a kívánt függvény jól (egyáltalán) ábrázolható-e.

Hatékonyság: a tanulási probléma jól kezelhető lesz-e egy adott reprezentáció választása esetén.
(tanulási algoritmus tár-, és időkomplexitása)

Összefoglaló táblázat – különböző szokásos paraméterek

true positive (TP)

eqv. with hit

true negative (TN)

eqv. with correct rejection

false positive (FP)

eqv. with false alarm, Type I error

false negative (FN)

eqv. with miss, Type II error

sensitivity or true positive rate (TPR)

eqv. with hit rate, recall

$$TPR = \frac{TP}{P} = \frac{TP}{TP + FN}$$

specificity (SPC) or true negative rate (TNR)

$$SPC = \frac{TN}{N} = \frac{TN}{FP + TN}$$

precision or positive predictive value (PPV)

$$PPV = \frac{TP}{TP + FP}$$

negative predictive value (NPV)

$$NPV = \frac{TN}{TN + FN}$$

fall-out or false positive rate (FPR)

$$FPR = \frac{FP}{N} = \frac{FP}{FP + TN} = 1 - SPC$$

false discovery rate (FDR)

$$FDR = \frac{FP}{FP + TP} = 1 - PPV$$

miss rate or false negative rate (FNR)

$$FNR = \frac{FN}{P} = \frac{FN}{FN + TP} = 1 - TPR$$

accuracy (ACC)

$$ACC = \frac{TP + TN}{P + N}$$

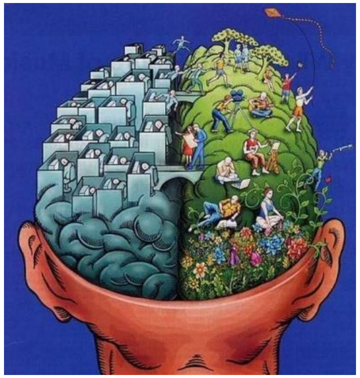
F1 score

is the harmonic mean of precision and sensitivity

$$F1 = \frac{2TP}{2TP + FP + FN}$$

https://en.wikipedia.org/wiki/Confusion_matrix

Budapesti Műszaki és Gazdaságtudományi Egyetem
Méréstechnika és Információs rendszerek Tanszék



Felügyelt tanulás, döntési fák

Előadó: Dr. Hullám Gábor

Előadás anyaga: Dr. Dobrowiecki Tadeusz
Dr. Pataki Béla

MŰEGYETEM 1782

A következőkben olyan módszereket vizsgálunk, amikor minták, **mintapéldák alapján akarjuk kialakítani a döntési rendszerünket**.

Nagyon gyakran a számítógép tanulásához rendelkezésünkre áll egy csomó mintapélda, és ez hordozza az információt. Nincsenek előre kialakított szabályaink a feladatra, csak a mintáink. Persze ilyenkor is valamilyen struktúrában igyekszünk felhasználni a minták hordozta tudást.

Tehát a módszerünk – az, hogy hogyan használjuk fel a mintákat – bizonyos mértékben problémafüggetlen.

A hétköznapi életben nagyon gyakori a minta alapján való tanulás, míg a klasszikus (számítógépes) megoldásokban a humán fejlesztő szabályokat alkotott, és ezeket programozta be.

Egy algoritmust, pl. egy döntést (ágensfüggvényt) alapvetően két módon lehet megvalósítani?

(1) Analitikus tervezés:

Begyűjteni az analitikus (fizikai, kémiai stb. összefüggésekből felépített) modelleket az adott problémára.

Megtervezni analitikusan a konkrét mechanizmust, és azt algoritmusként implementálni.

(2) **Tanulás:**

Megtervezni a tanulás mechanizmusát, és azt algoritmusként implementálni.

Majd alkalmazásával megtanulni vele a minták alapján a döntés tényleges mechanizmusát, és azt algoritmusként implementálni.

A következőkben a (2)-vel foglalkozunk

Tanulás alapvető fajtái:

felügyelt tanulás egy esetnél mind a bemenetet, mind a kimenetet észlelni tudjuk (bemeneti minta + kívánt válasz)

megerősítéses tanulás az ágens az általa végrehajtott tevékenység csak bizonyos értékelését kapja meg, esetleg nem is minden lépésben (jutalom, büntetés, **megerősítés**)

felügyelet nélküli tanulás semmilyen információ sem áll rendelkezésünkre a helyes kimenetről (az észlelések közötti összefüggések tanulása)

féligellenőrzött tanulás a tanításra használt esetek egy részénél mind a bemenetet, mind a kimenetet észlelni tudjuk (bemeneti minta + kívánt válasz), a másik – tipikusan nagyobb – részénél csak a bemeneti leírás ismert

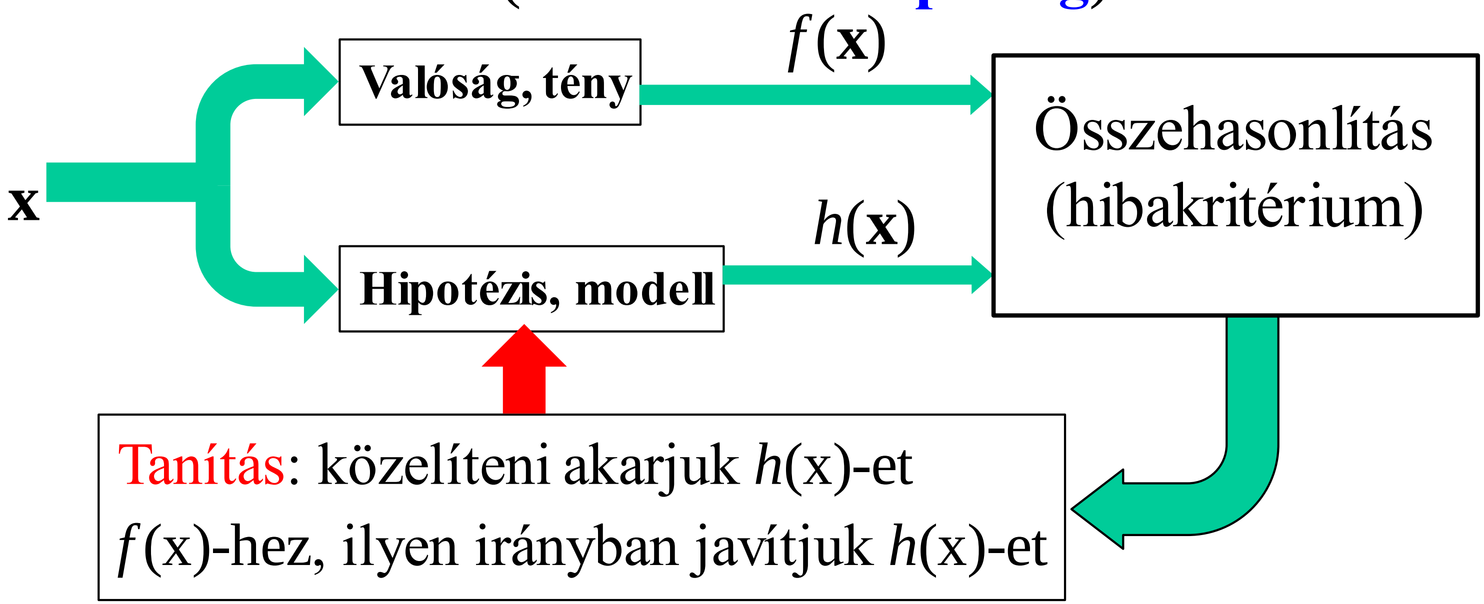
Felügyelt tanulás (pl. induktív következtetés):

tanulási példa: $(\mathbf{x}, f(\mathbf{x}))$ adatpár, ahol $f(\mathbf{x})$ ismeretlen

tanulás célja: $f(\mathbf{x})$ értelmes közelítése egy $h(\mathbf{x})$ hipotézissel

$h(\mathbf{x}) = f(\mathbf{x})$, $\mathbf{x}$ – ismert példákon gyakran teljes pontosság

$h(\mathbf{x}') \cong f(\mathbf{x}')$, **$\mathbf{x}'$ – a tanulás közben még nem látott esetek (általánosító képesség)**



<u>Feladat</u>: *az ismeretlen* ***f****-re vonatkozó példák egy halmaza alapján (****tanítóhalmaz****), adjon meg egy olyan* ***h*** *függvényt (****hipotézist****), amely tulajdonságaiban jól közelíti az* ***f****-et (amit* ***teszthalmazon*** *verifikálunk).*

Sokféle tanítható eszközt kitaláltak:

- neurális hálók
- Bayes-hálók
- kernelgépek
- döntési fák
- legközelebbi szomszéd osztályozók
- stb. stb.

A következőkben először a döntési fákkal foglalkozunk – működésük az emberi gondolkodás számára jóval jobban áttekinthető, mint pl. a mesterséges neurális hálóké. (white box ⇔ black box)
Vannak regressziós döntési fák is, de mi az osztályozással foglalkozunk.

Példaprobléma: döntés, hogy egy adott étteremben várjunk-e asztalra.

Cél: előállítani a döntés logikai függvényét

A problémát tulajdonságokkal vagy **attribútumokkal** jellemezzük.

1. *Alternatíva*: van-e a környéken más megfelelő étterem. (I/N)
2. *Bár*: van-e az étteremben kényelmes bár, ahol várhatunk. (I/N)
3. *Pén/Szom*: igaz pénteken- és szombatonként. (I/N)
4. *Éhes*: éhesek vagyunk-e. (I/N)
5. *Kuncsaft*: hányan vannak már benn (*Senki*, *Néhány* és *Tele*).
6. *Ár*: mennyire drága (*Olcsó, Közepes, Drága*)
7. *Eső*: esik-e kint (I/N)
8. *Foglalás*: foglalni kell-e előzetesen az asztalt (I/N)
9. *Típus*: az étterem típusa (*Francia*, *Olasz*, *Thai v. Burger*)
10. *BecsültVár*: a pincér által bemondott várakozási idő (0-10 perc, 10-30, 30-60, >60 perc)

A **döntés eredményét** a döntési ágensfüggvény adja meg:

VárniFog = *f(Alternativa*, *Bár*, *Pén/Szom*, *Éhes*, *Kuncsaft*, *Ár*, *Eső*, *Foglalás*, *Típus*, *BecsültVár*, …)

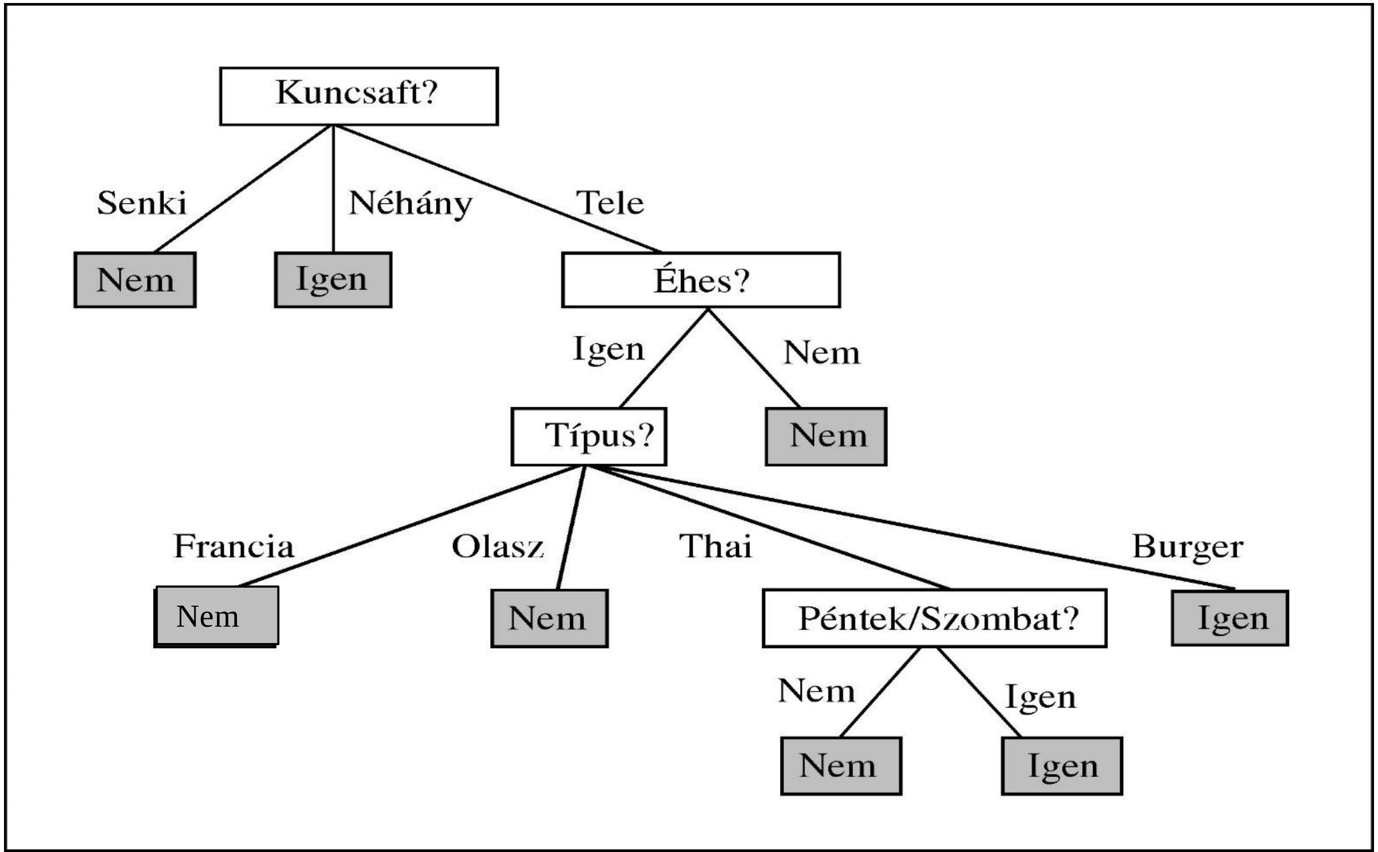
Az eddigi kiruccanásaink tapasztalata - tanítóminták

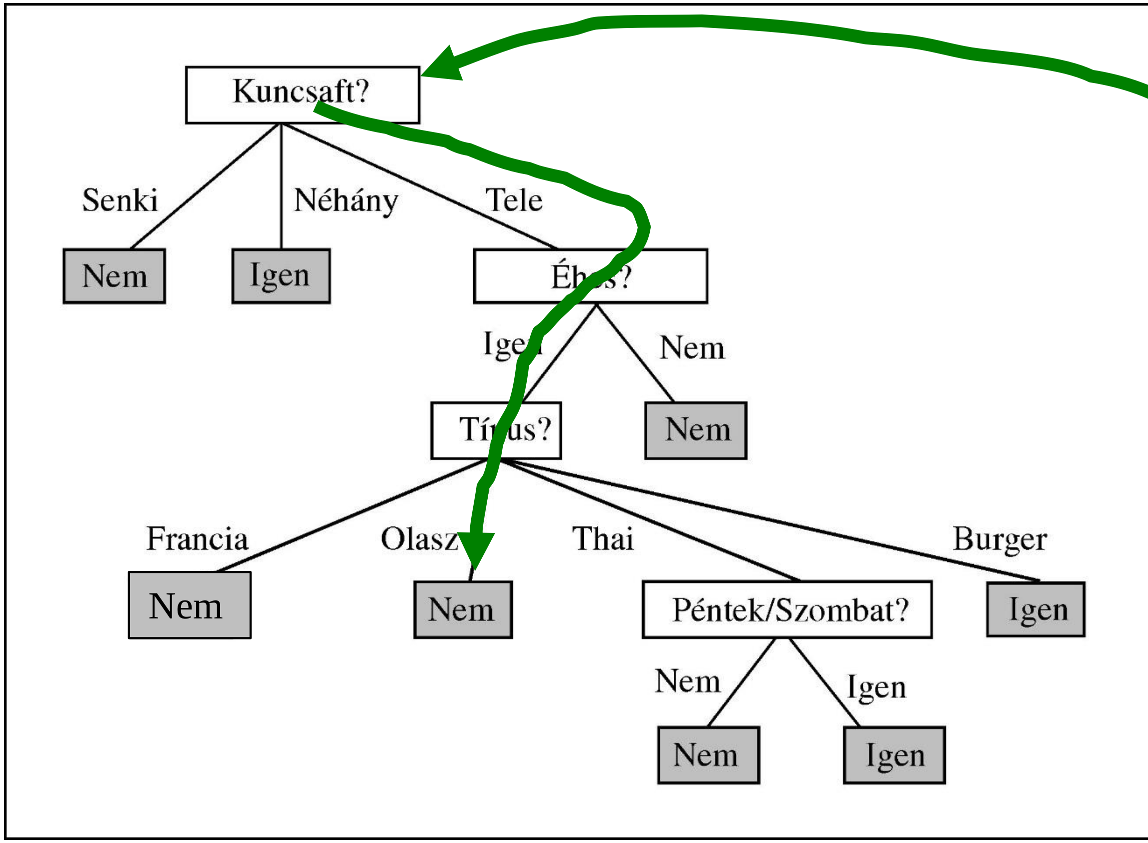
Pl.	**Attribútumok**										**Cél**
	Altern	*Bár*	*Pént*	*Éhes*	*Kuncs*	*Ár*	*Eső*	*Fogl*	*Típus*	*Becs*	*VárniFog*
X_1	*Igen*	*Nem*	*Nem*	*Igen*	*Néhány*	*Drága*	*Nem*	*Igen*	*Francia*	*0–10*	*Igen*
X_2	*Igen*	*Nem*	*Nem*	*Igen*	*Tele*	*Olcsó*	*Nem*	*Nem*	*Thai*	*30–60*	*Nem*
X_3	*Nem*	*Igen*	*Nem*	*Nem*	*Néhány*	*Olcsó*	*Nem*	*Nem*	*Burger*	*0–10*	*Igen*
X_4	*Igen*	*Nem*	*Igen*	*Igen*	*Tele*	*Olcsó*	*Nem*	*Nem*	*Thai*	*10–30*	*Igen*
X_5	*Igen*	*Nem*	*Igen*	*Nem*	*Tele*	*Drága*	*Nem*	*Igen*	*Francia*	*>60*	*Nem*
X_6	*Nem*	*Igen*	*Nem*	*Igen*	*Néhány*	*Közep*	*Igen*	*Igen*	*Olasz*	*0–10*	*Igen*
X_7	*Nem*	*Igen*	*Nem*	*Nem*	*Senki*	*Olcsó*	*Igen*	*Nem*	*Burger*	*0–10*	*Nem*
X_8	*Nem*	*Nem*	*Nem*	*Igen*	*Néhány*	*Közep*	*Igen*	*Igen*	*Thai*	*0–10*	*Igen*
X_9	*Nem*	*Igen*	*Igen*	*Nem*	*Tele*	*Olcsó*	*Igen*	*Nem*	*Burger*	*>60*	*Nem*
X_{10}	*Igen*	*Igen*	*Igen*	*Igen*	*Tele*	*Drága*	*Nem*	*Igen*	*Olasz*	*10–30*	*Nem*
X_{11}	*Nem*	*Nem*	*Nem*	*Nem*	*Senki*	*Olcsó*	*Nem*	*Nem*	*Thai*	*0–10*	*Nem*
X_{12}	*Igen*	*Igen*	*Igen*	*Igen*	*Tele*	*Olcsó*	*Nem*	*Nem*	*Burger*	*30–60*	*Igen*

Mi benne a viselkedésünk mintája? Van-e egyáltalán egy konzisztens - minden tanítópéldának megfelelő – viselkedési mintánk?
(9216 lehetséges példa van, 12 alapján tanulunk)

Döntési fák tanulása

döntési fa = **egy logikai függvény**
bemenete: egy tulajdonsághalmazzal leírt objektum vagy szituáció
kimenete: egy igen/nem „döntés/cselekvés"
belső csomópont = valamelyik tulajdonság tesztje
él = a teszt lehetséges értéke
levél = logikai érték, amelyet akkor kell kiadni, ha ezt a levelet elértük





Egy konkrét eset:

Kuncsaft = Tele és
Éhes és
Típus = Olasz

VárniFog = (Kuncsaft = Néhány) V
((Kuncsaft = Tele) ∧ Éhes ∧ (Típus = Thai) ∧ Pént/Szomb) V
((Kuncsaft = Tele) ∧ Éhes ∧ (Típus = Burger))

Döntési fák kifejezőképessége

teljes - **minden (ítélet)logikai függvény felírható döntési faként**.
(az igazságtábla minden sora = a fa egy bejárása,
de így az igazságtábla mérete = a fa mérete = exponenciális!)

Döntési fák kialakítása példák alapján

példa: (attribútumok értékei, célpredikátum értéke)

példa **besorolása**: a célpredikátum értéke

pozitív/negatív példa: a célpredikátum értéke igaz / hamis

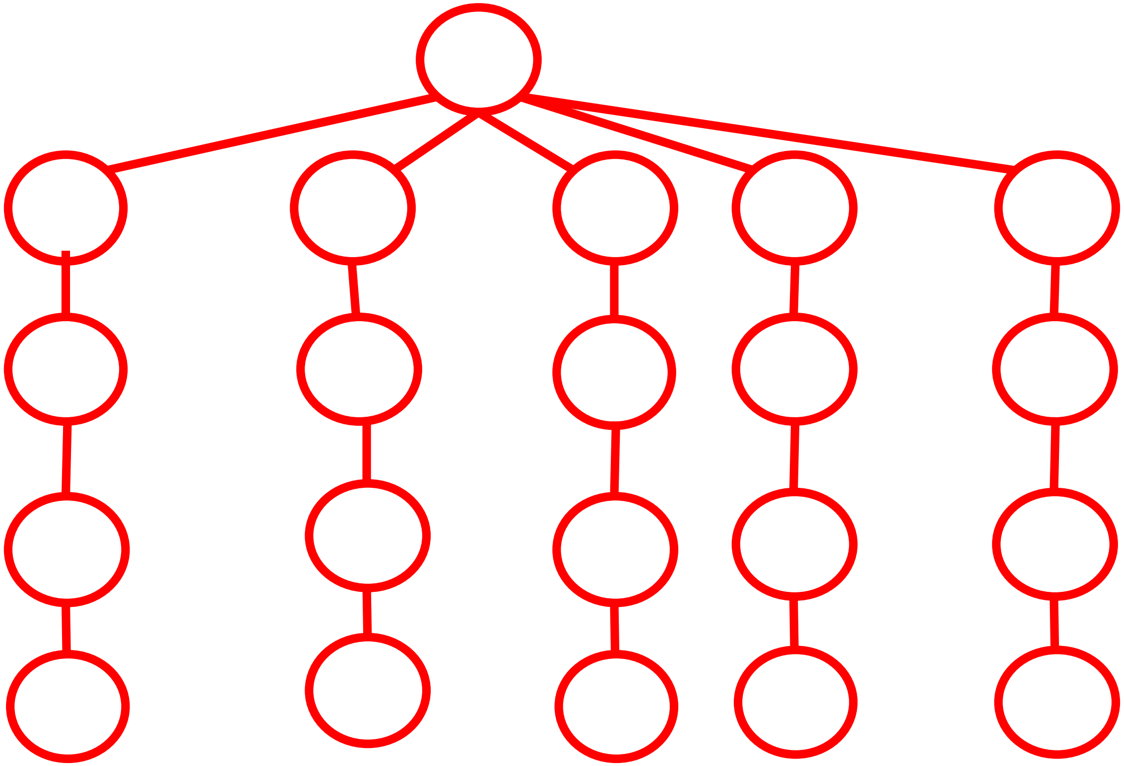
tanító halmaz: a teljes példahalmaz

triviális fa, könnyű - mindegyik példához egy önálló bejárási út

a levél a példa besorolását adja.

a fa egyszerűen memorizálja a példát, nem alakít ki jellegzetes mintákat

a példákból **nem általánosít** (nem tud ismeretlen példákra extrapolálni)

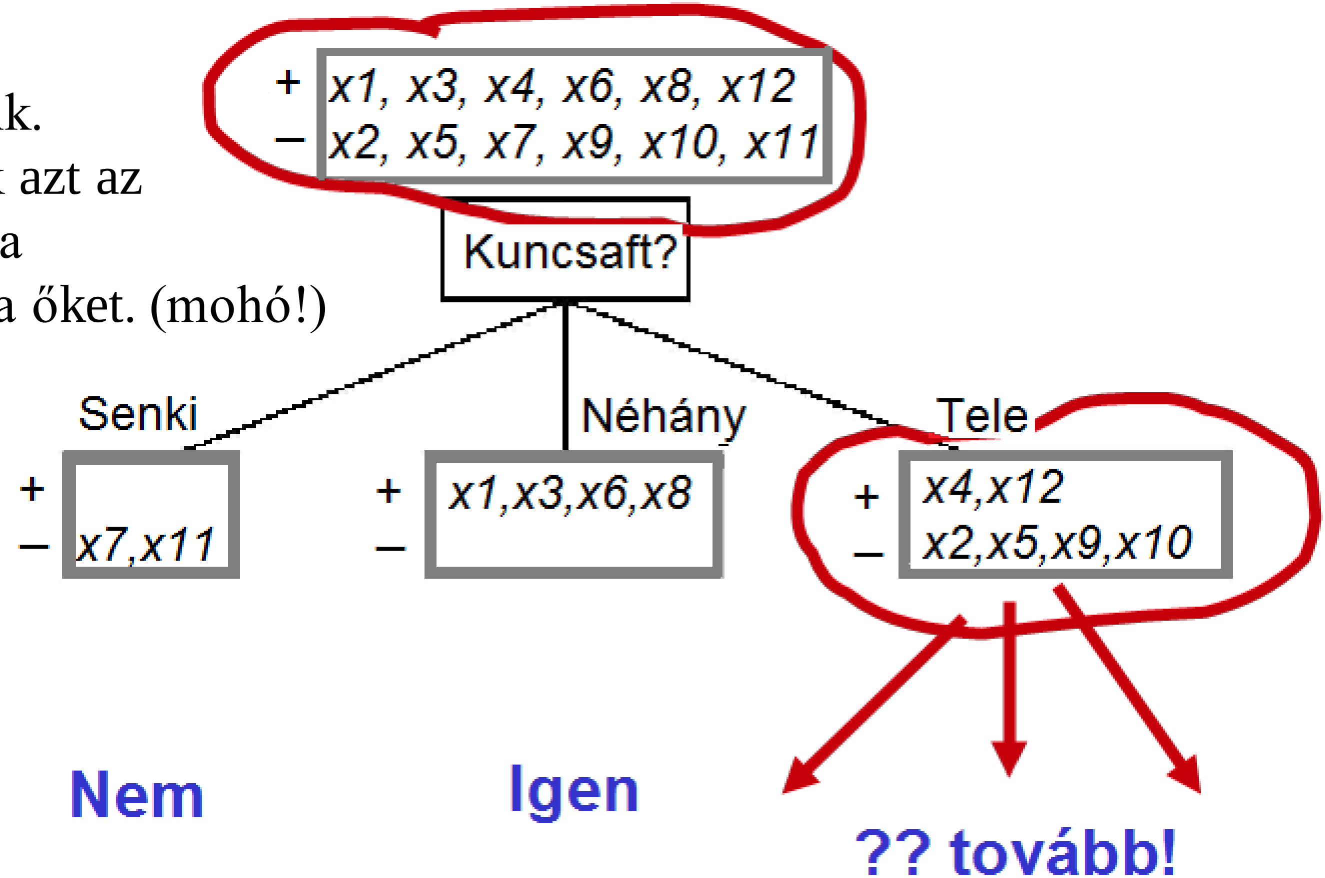


„**igazi**" fa - a jellegzetes minták kinyerése,

a módszer képes nagyszámú esetet **tömör** formában leírni, és így az egyelőre ismeretlen példákra is **általánosítani**

A döntési fa építése – általános jelenségek

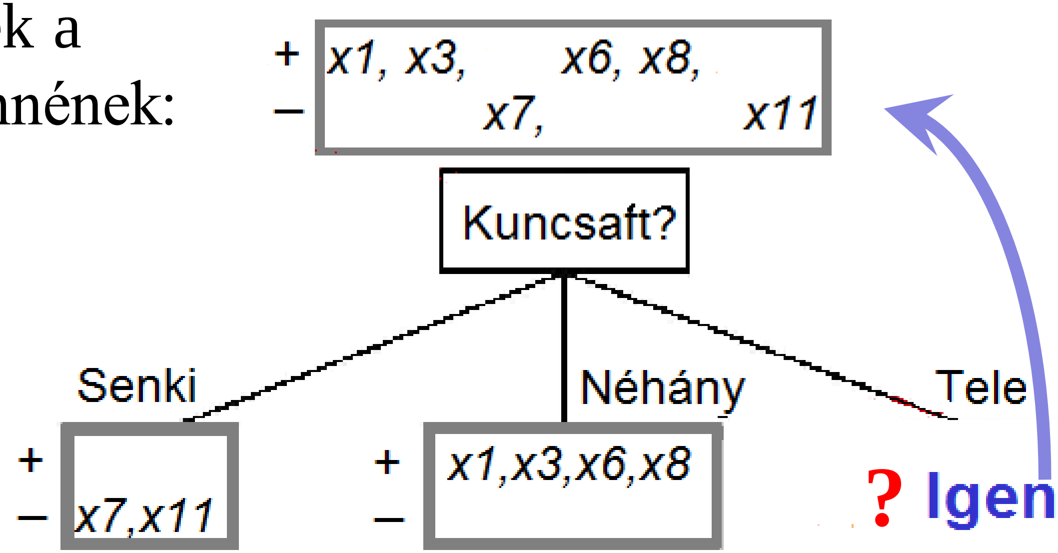
Van néhány pozitív és
néhány negatív példánk.
Heurisztika: válasszuk azt az attribútumot, amelyik a legjobban szétválasztja őket. (mohó!)



Ha az összes megmaradt eset pozitív, (vagy az összes negatív), akkor készen vagyunk azzal az ággal: a válasz **Igen** vagy **Nem**.

A döntési fa építése – általános jelenségek

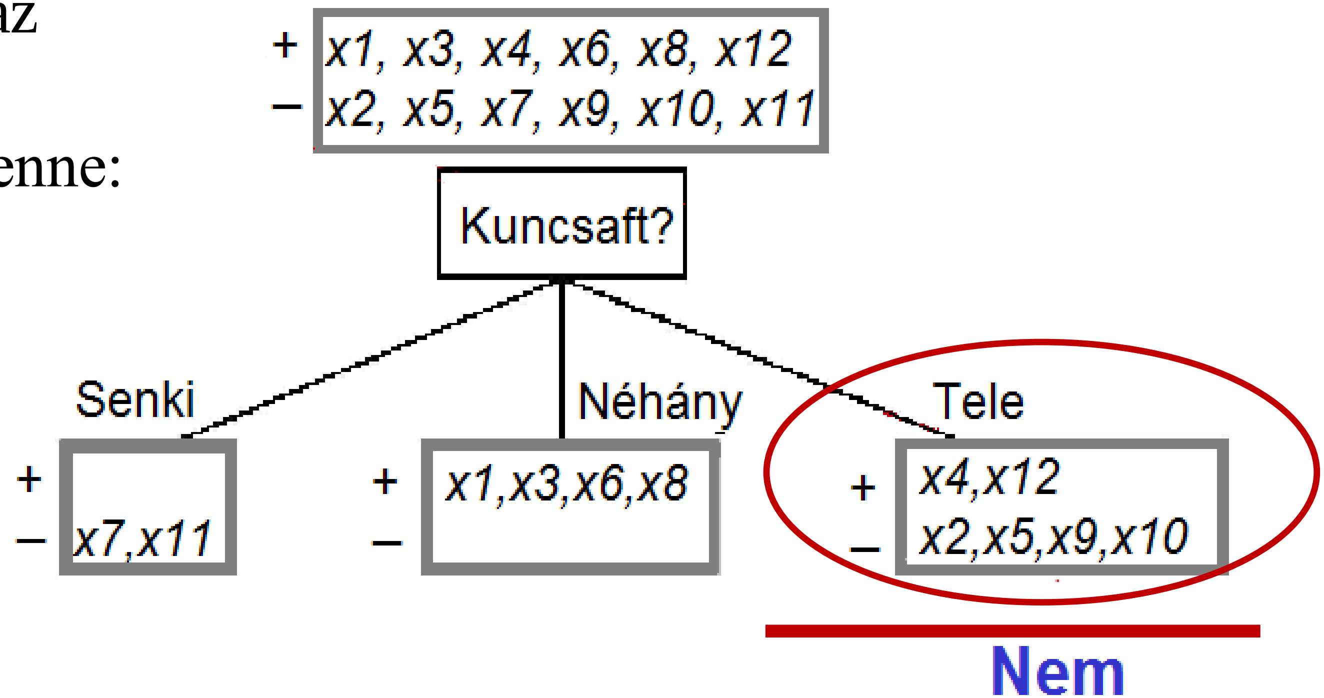
Ha csak ezek a példáink lennének:



Ha nem maradt egyetlen példa sem, ez azt jelenti, hogy ilyen példát nem figyeltünk meg eddig…., de a jövőben mégis jelentkezhet. Ilyenkor valamilyen alapértéket adunk vissza, amelyet rendszerint a csomópont szülőjének többségi válaszából származtatunk.

A döntési fa építése – általános jelenségek

Ha csak ez az egyetlen attribútum lenne:



Nem maradt már teszteletlen attribútum, de maradtak még pozitív és negatív példák. Baj van.

Ezeknek a példáknak pontosan megegyezik a leírása, de különböző a besorolásuk.

Néhány adat tehát nem korrekt: a **zaj** torzítja az adatokat – vagy hiányzik még attribútum (infó).

Megoldás? Pl. a **többségi szavazás** használata.

Döntési fák kialakítása példák alapján

példa: (attribútumok értékei, célpredikátum értéke)
példa **besorolása**: a célpredikátum értéke
pozitív/negatív példa: a célpredikátum értéke igaz / hamis
tanító halmaz: a teljes példahalmaz

A *legkisebb döntési fa* megtalálása - általánosságban nem megoldható

Heurisztika: **mohóság** - egy meglehetősen egyszerű (jó) fa is jó lesz!
Az alapötlet: először a „legfontosabb” attribútumot teszteljük.

„**legfontosabb**" = a **legnagyobb eltérést** okozza példák besorolásában

Elvárás: kisszámú példa alapján korrekt besoroláshoz jussunk:
a **bejárási utak rövidek** legyenek, és így az **egész fa kicsi** (tömör) legyen.

Döntési fák kialakítása példák alapján

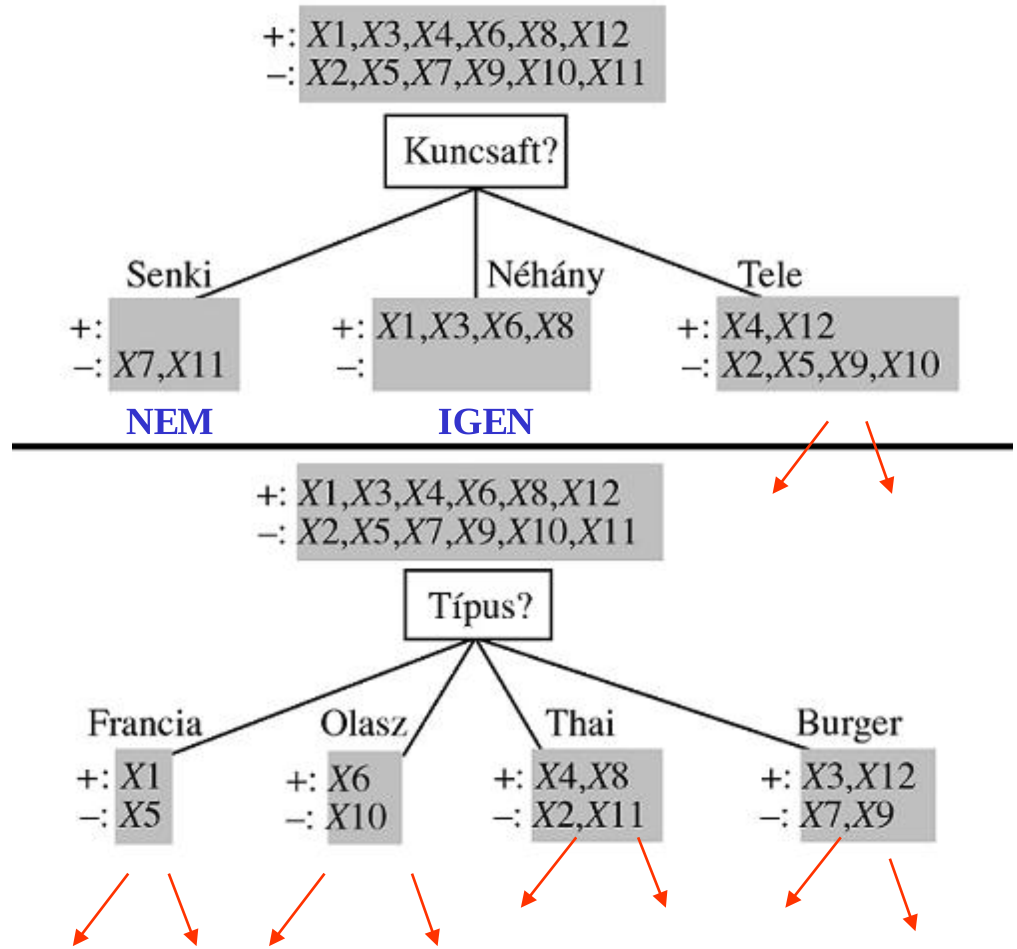
A **tökéletes attribútum** a példákat két csoportra bontja,
az egyikbe **csak pozitív**,
a másikba **csak negatív** példák tartoznak.

Ezzel be is lehetne fejezni a fa tanítását – de tökéletes attribútum általában nincs!

Olyan attribútumot válasszunk, amely a lehető legmesszebbre jut a pontos besorolás biztosításában.

A *Kuncsaft* attribútum nem tökéletes, de elég jó.
Ha a Kuncsaft tesztelése után döntünk, várhatóan legfeljebb 2-t hibázunk a 12 példából.

A *Típus* attribútum ehhez képest? (vacak…)
Ha a Típus tesztelése után döntünk, várhatóan 6-ot hibázunk a 12 példából. Éppúgy, mint a teszt előtt!



Egy nagymértékben haszontalan attribútum, mint pl. a *Típus* olyan példahalmazokat hoz létre, amelyek nagyjából ugyanolyan arányban tartalmaznak pozitív és negatív példákat, mint az eredeti halmaz.

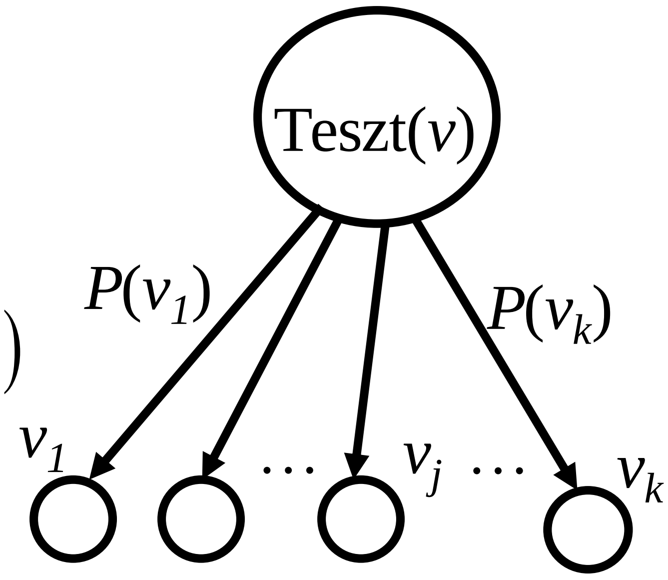
„Elég jó?" „Nagymértékben haszontalan?„

A mérték legyen **maximális**: a **tökéletes** attribútumra,
minimális: olyan attribútumra, aminek egyáltalán **nincs értéke** számunkra.

Egy megfelelő mérték: egy helyzetben az információszükségletünk a helyes megoldáshoz, az attribútum által adott információ várható értéke, információ átlagos tartalma, entrópia, …

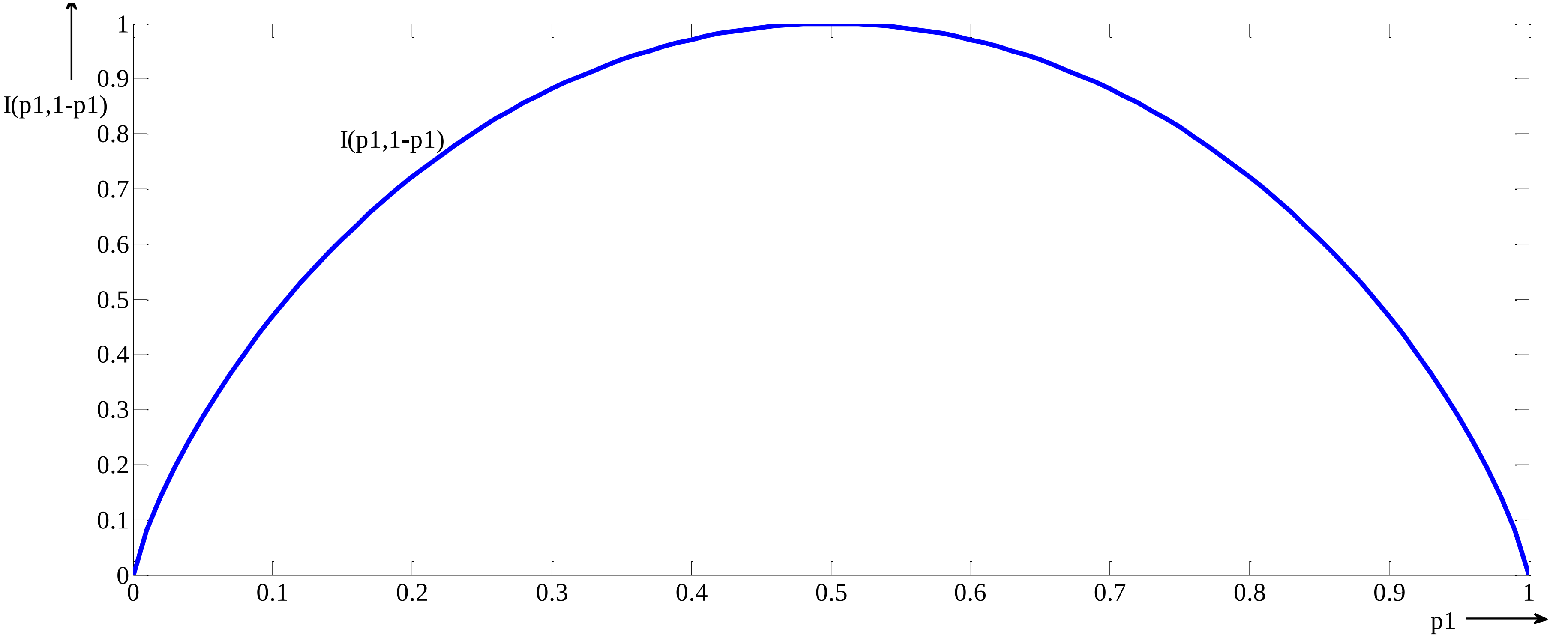
Ha a lehetséges v_j válaszok valószínűsége $P(v_j)$, akkor az adott kiinduló halmaz esetén a jó döntéshez szükséges információszükséglet:

$$I(P(v_1), P(v_2), \ldots, P(v_k)) = \sum_{j=1}^{k} -P(v_j) \cdot \log_2\left(P(v_j)\right)$$



pl. Fogadunk 1000 Ft-ba egy szabályos pénzérme dobása esetén, hogy „fej” jön ki.

- Mennyit ér nekünk az infó, ha valaki elárulja, hogy melyik fog kijönni? Mennyi infóra van szükségünk?
- Mennyit ér az infó, ha tudjuk, hogy az érme aszimmetrikus, és 80%-ban „fej” jön ki?
- Mennyit ér nekünk az infó, ha az érme mindkét oldalán „fej” van, tehát 100%-ban ez jön ki?



A döntési fa információtartalma - a tanítóhalmazban található pozitív és negatív példák aránya alapján

A tanítóhalmaz ***p*** pozitív és ***n*** negatív példát tartalmaz. A két válasz: v_1 (pozitív), v_2 (negatív), és a valószínűségük becslése:

$$P\left(v_1\right) \cong \frac{p}{p+n}, \qquad P\left(v_2\right) \cong \frac{n}{p+n}$$

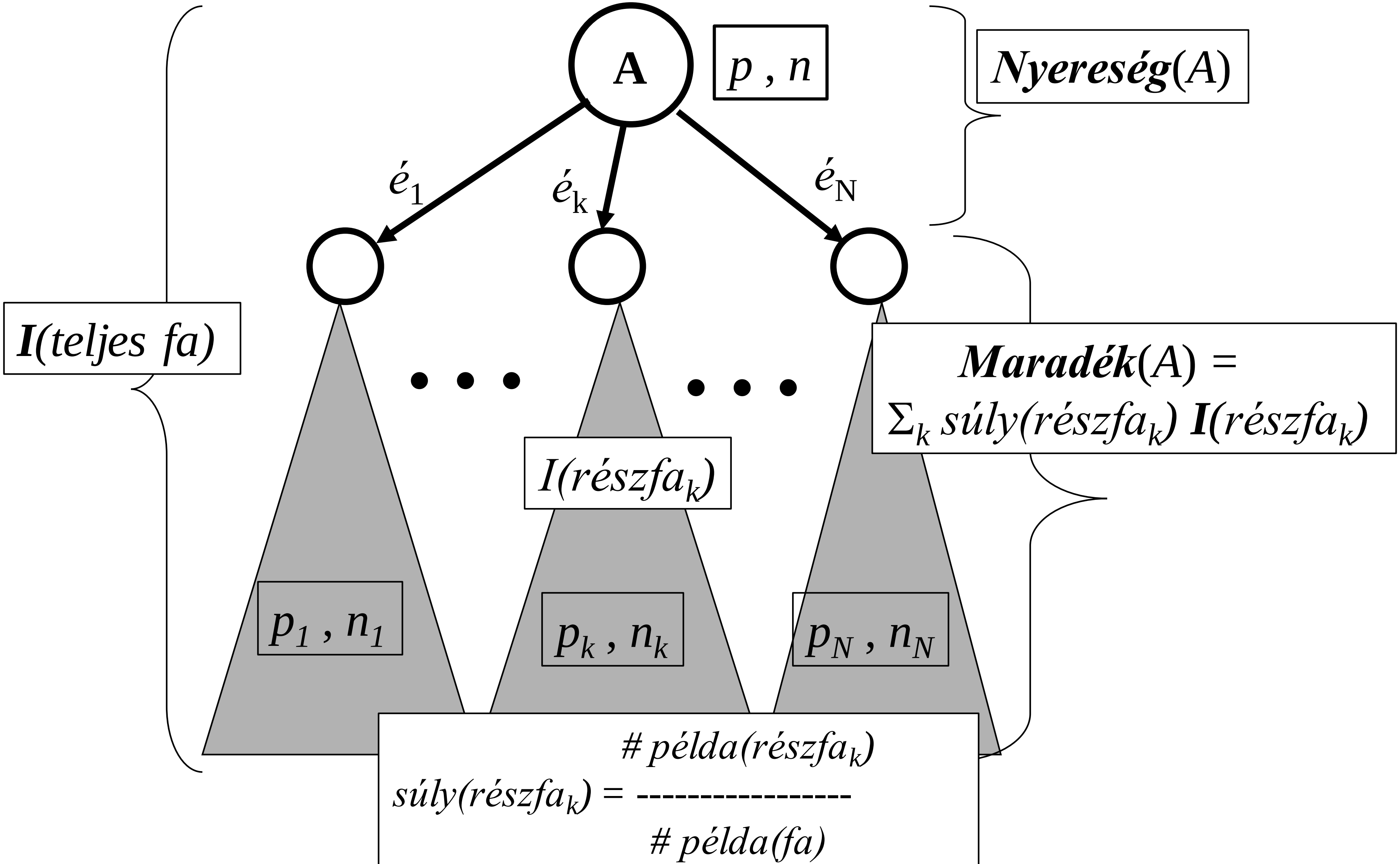
Ekkor a **fa információtartalmának** becslése:

$$I(P(v_1), P(v_2)) \cong I\left(\frac{p}{p+n}, \frac{n}{p+n}\right) = -\frac{p}{p+n} \cdot \log_2\left(\frac{p}{p+n}\right) - \frac{n}{p+n} \cdot \log_2\left(\frac{n}{p+n}\right)$$

Hogyan válasszuk ki, hogy melyik attribútumot teszteljük?
Mennyi új információt ad nekünk egy attribútum tesztelése?

Mennyi információra volt szükségünk az attribútum tesztelése **előtt**, és mennyire van **még** szükségünk az attribútum tesztelése **után**? A kettő különbségét nyertük meg a teszttel!

Bármelyik ***A*** attribútum az ***E*** tanító halmazt $\boldsymbol{E}_1, \boldsymbol{E}_2, \ldots, \boldsymbol{E}_N$ részhalmazokra bontja az ***A*** tesztjére adott válaszoknak megfelelően, ha ***A*** tesztje *N* különböző választ ad.



Hogyan válasszunk attribútumot?

$$Maradék(A)=\sum_{k=1}^{N}\frac{p_k+n_k}{p+n}\cdot I\left(\frac{p_k}{p_k+n_k},\frac{n_k}{p_k+n_k}\right)$$

Az attribútum tesztjének **információnyeresége** az **eredeti** információigény és a **teszt utáni** új információigény **különbsége**:

$$Nyereség(A)=I\left(\frac{p}{p+n},\frac{n}{p+n}\right)-Maradék(A)$$

Válasszuk a pillanatnyilag legnagyobb nyereséget adó attribútumot!
(Mohó eljárás)

Közbevetés: mi az információszükséglet, ha 100%-os az egyik (és 0%-os a másik) eset?

$$I(0,1) = ?$$

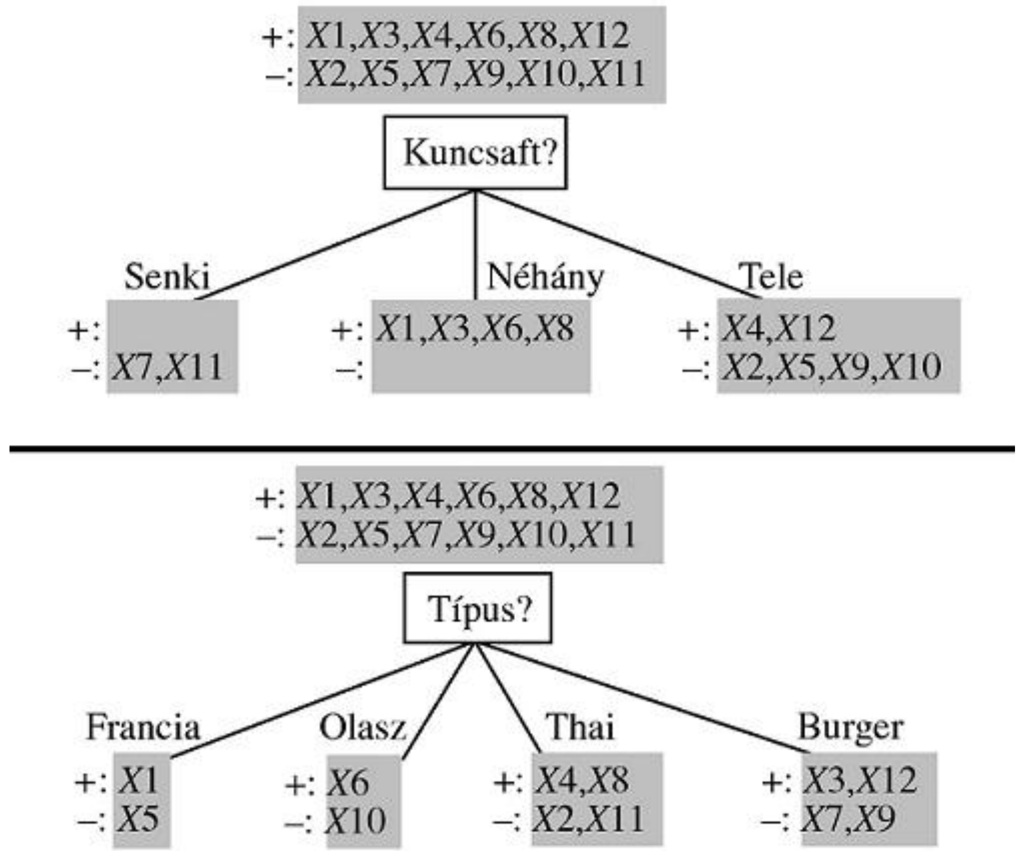
$\log_2()$ helyett használhatunk ln()-t, hiszen csak egy konstanssal való szorzásban különböznek: $\log_2(x) = \log_2(e).\ln(x)$

$$0 \cdot \ln(0) = \lim_{x \to 0} \frac{\ln(x)}{1/x}$$

helyette a deriváltak hányadosa vizsgálható (l′Hopital szabály)

$$0 \cdot \ln(0) = \lim_{x \to 0} \frac{\ln(x)}{1/x} = \lim_{x \to 0} \frac{1/x}{-1/x^2} = \lim_{x \to 0}(-x) = 0$$

Nézzük meg a *Kuncsaft* és a *Típus* attribútumokat



$$Nyereség(Kuncsaft) = 1 - \left[\frac{2}{12}\cdot I(0,1) + \frac{4}{12}\cdot I(0,1) + \frac{6}{12}\cdot I(\frac{2}{4},\frac{4}{4})\right] \approx 0,54 \text{ bit}$$

$$Nyereség(Típus) = 1 - \left[\frac{2}{12}\cdot I(\frac{1}{2},\frac{1}{2}) + \frac{2}{12}\cdot I(\frac{1}{2},\frac{1}{2}) + \frac{4}{12}\cdot I(\frac{2}{4},\frac{2}{4}) + \frac{4}{12}\cdot I(\frac{2}{4},\frac{2}{4})\right] = 0 \text{ bit}$$

Több attribútumra is kellene a számítás, de a Kuncsaft kimagaslóan jó.

(**Kuncsaft = Tele**) ágon még fennmaradó példák

Példák	**Attribútumok**									**Cél**
	Altern	*Bár*	*Pént*	*Éhes*	*Ár*	*Eső*	*Fogl*	*Típus*	*Becs*	***VárniFog***
X_2	*Igen*	*Nem*	*Nem*	*Igen*	*Olcsó*	*Nem*	*Nem*	*Thai*	*30–60*	***Nem***
X_5	*Igen*	*Nem*	*Igen*	*Nem*	*Drága*	*Nem*	*Igen*	*Francia*	*>60*	***Nem***
X_9	*Nem*	*Igen*	*Igen*	*Nem*	*Olcsó*	*Igen*	*Nem*	*Burger*	*>60*	***Nem***
X_{10}	*Igen*	*Igen*	*Igen*	*Igen*	*Drága*	*Nem*	*Igen*	*Olasz*	*10–30*	***Nem***

X_4	*Igen*	*Nem*	*Igen*	*Igen*	*Olcsó*	*Nem*	*Nem*	*Thai*	*10–30*	***Igen***
X_{12}	*Igen*	*Igen*	*Igen*	*Igen*	*Olcsó*	*Nem*	*Nem*	*Burger*	*30–60*	***Igen***

Részfában: $p_1 = 2$, $n_1 = 4$, $p_1/(p_1+n_1) = 1/3$, $n_1/(p_1+n_1) = 2/3$

(**Kuncsaft = Tele**) ág

I = I(részfa) = I(2/6, 4/6) = .9183 (2 pozitív és 4 negatív példa)

Ha pl. **Alternatíva-**t teszteljük, akkor a nyereség:

$$\mathrm{Ny(Altern)} = I - [\underbrace{5/6\ I(2/5,\ 3/5)}_{\text{Altern = Igen}} + \underbrace{1/6\ I(0,\ 1)}_{\text{Altern = Nem}}]$$

$$\left.\begin{array}{ll} p2 = 2,\ p3 = 0 & \\ n2 = 3 & n3 = 1 \end{array}\right\} \text{6 példa}$$

$$\mathrm{Ny(Altern)} = I - [5/6\ I(2/5,\ 3/5) + 1/6\ I\cancel{(0,\ 1)}]$$

$$= .92 - .81 \approx .11$$

(**Kuncsaft = Tele**) ág

Ny(Altern) = I – [5/6 I(2/5, 3/5) + 1/6 I(0,1)] = .92 - .81 ≈ .11

Ny(Bár) = I – [1/2 I(1/3, 2/3) + 1/2 I(1/3, 2/3)] = .92 - .92 = 0

Ny(Péntek) = I – [5/6 I(2/5, 3/5) + 1/6 I(0,1)] = .92 - .81 ≈ .11

Ny(Éhes) = I – [4/6 I(1/2, 1/2) + 2/6 I(0,1)] = .92 - .66 ≈ .25 ***Válasszuk pl. ezt!***

Ny(Ár) = I – [4/6 I(1/2, 1/2) + 2/6 I(0,1)] = .92 - .66 ≈ .25

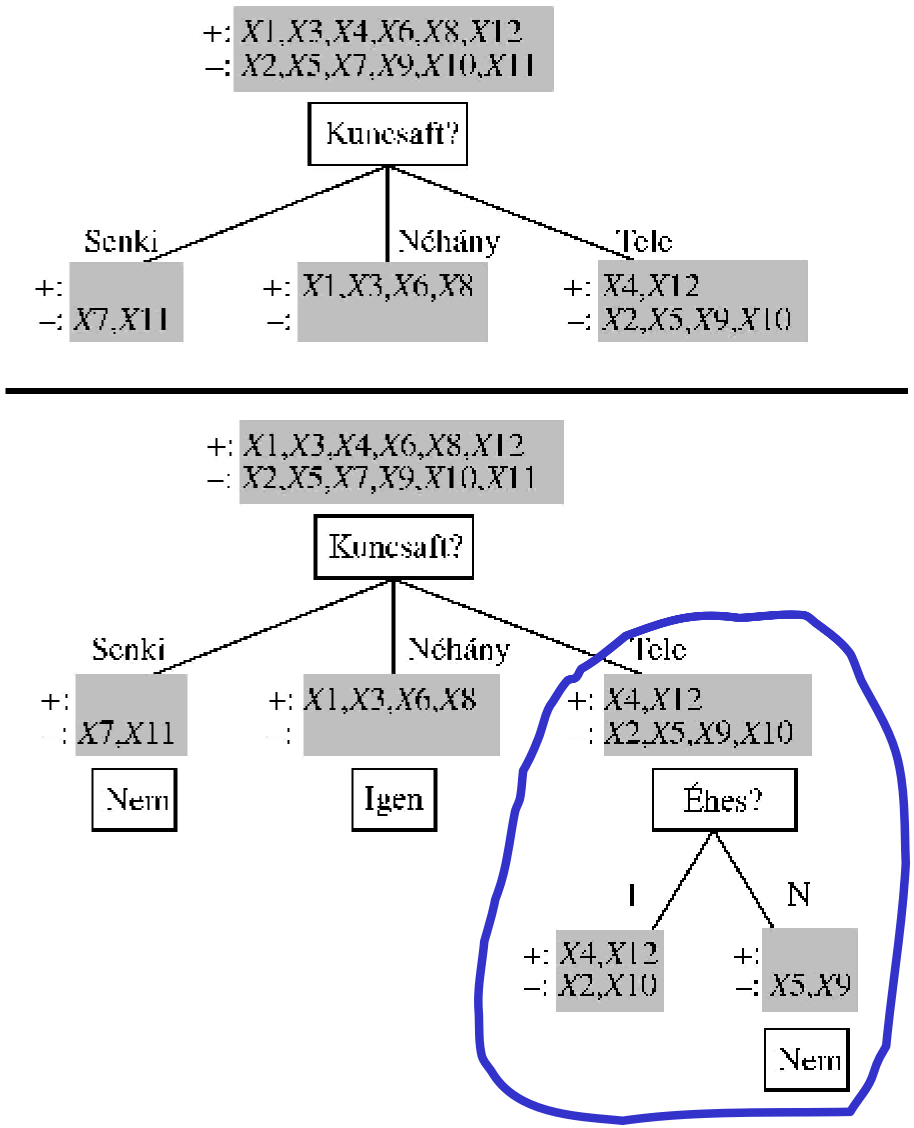
Ny(Eső) = I – [5/6 I(2/5, 3/5) + 1/6 I(0,1)] = .92 - .81 ≈ .11

Ny(Foglalt) = I – [2/6 I(0,1) + 4/6 I(1/2,1/2)] = .92 - .66 ≈ .25

Ny(Típus) = I –
[2/6 I(1/2, 1/2) + 1/6 I(0,1) + 2/6 I(1/2, 1/2) + 1/6 I(0,1)] = .92 - .66 ≈ .25

Ny(Becs) = I –
[2/6 I(1/2, 1/2) + 2/6 I(0,1) + 2/6 I(1/2, 1/2)] = .92 - .66 ≈ .25

(Kuncsaft = Tele) ÉS Teszt(Éhes?)



(**Kuncsaft = Tele**) ÉS **(Éhes?=Igen)** ágon még fennmaradó példák

Példa	**Attribútumok**								**Cél**
	Alt	*Bár*	*Pént*	*Ár*	*Eső*	*Fogl*	*Típus*	*Becs*	***VárniFog***
X_2	*Igen*	*Nem*	*Nem*	*Olcsó*	*Nem*	*Nem*	*Thai*	*30–60*	***Nem***
X_{10}	*Igen*	*Igen*	*Igen*	*Drága*	*Nem*	*Igen*	*Olasz*	*10–30*	***Nem***
X_4	*Igen*	*Nem*	*Igen*	*Olcsó*	*Nem*	*Nem*	*Thai*	*10–30*	***Igen***
X_{12}	*Igen*	*Igen*	*Igen*	*Olcsó*	*Nem*	*Nem*	*Burger*	*30–60*	***Igen***

Részfában: $p_1 = 2$, $n_1 = 2$, $p_1/(p_1+n_1) = ½$, $n_1/(p_1+n_1) = ½$

Információszükséglet a következő teszt előtt ezen az ágon: 1 bit!

Különböző attribútumok nyeresége a (**Kuncsaft = Tele**) ÉS **(Éhes?=Igen)**

Ny(Alt) = I – [1/1 I(1/2, 1/2) + 0] = 1 - 1 = 0

Ny(Bár) = I – [1/2 I(1/2, 1/2) + 1/2 I(1/2, 1/2)] = 1 - 1 = 0

Ny(Péntek) = I – [1/4 I(0,1) + 3/4 I(1/3,2/3)] = 1 - .92 ≈ .08

Ny(Ár) = I – [1/4 I(0,1) + 3/4 I(1/3,2/3)] = 1 - .92 ≈ .08

Ny(Eső) = I – [1/1 I(1/2, 1/2) + 0] = 1 - 1 = 0

Ny(Foglalt) = I – [1/4 I(0,1) + 3/4 I(1/3,2/3)] = 1 - .92 ≈ .08

Ny(Típus) = I – [1/4 I(0,1) + 1/4 I(0,1) + 1/2 I(1/2, 1/2)] = .5 ***Válasszuk pl. ezt!***

Ny(Becs) = I – [1/2 I(1/2, 1/2) + 1/2 I(1/2, 1/2)] = 1 - 1 = 0

(Kuncsaft = Tele) ÉS **(Éhes?=Igen) ÉS (Típus = Thai)**
ágon még fennmaradó példák

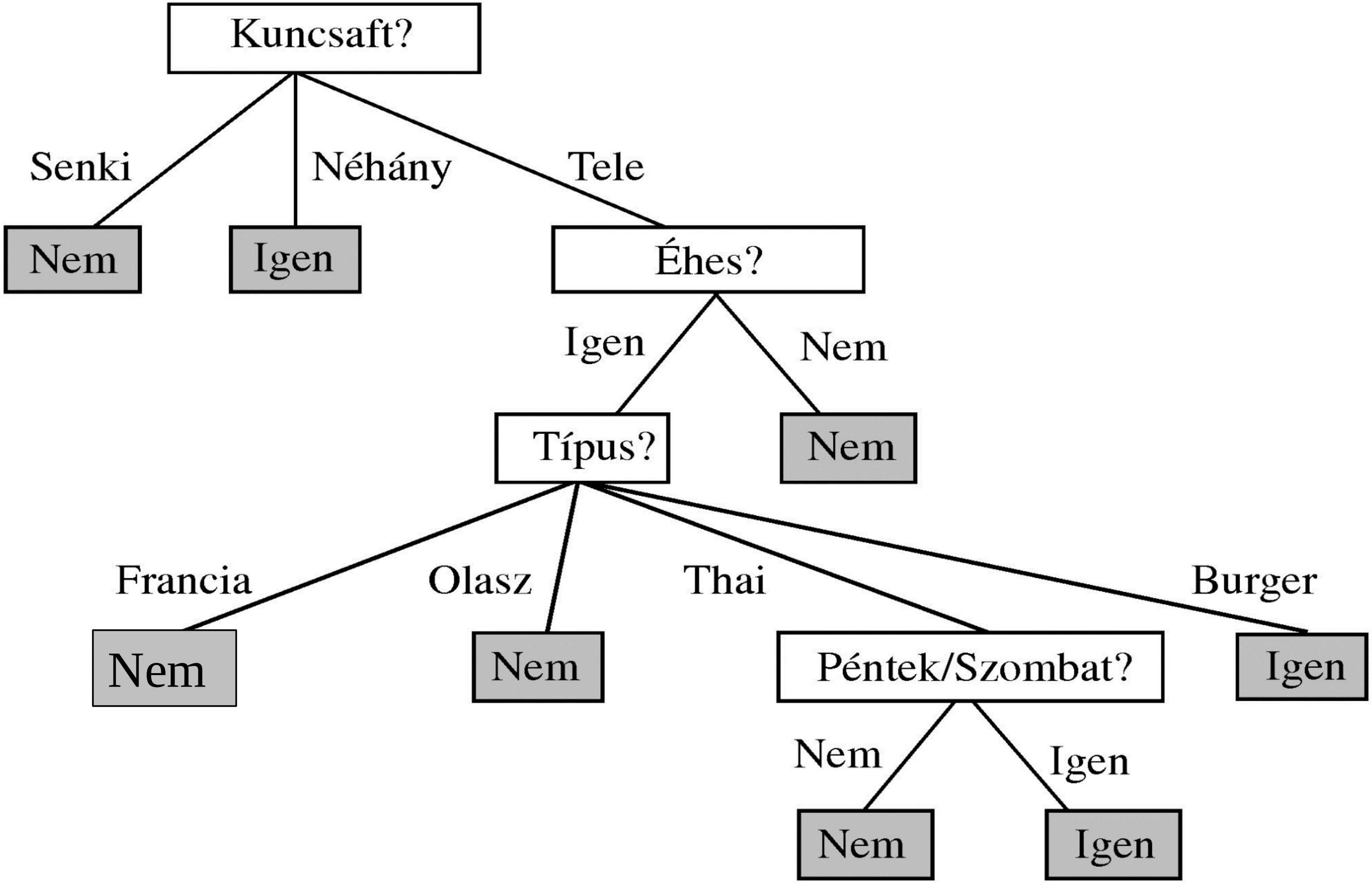
Példa				**Attribútumok**				**Cél**
	Alt	*Bár*	*Pént/Szo*	*Ár*	*Eső*	*Fogl*	*Becs*	***VárniFog***
$\boldsymbol{X_2}$	*Igen*	*Nem*	*Nem*	*Olcsó*	*Nem*	*Nem*	*30–60*	***Nem***
$\boldsymbol{X_4}$	*Igen*	*Nem*	*Igen*	*Olcsó*	*Nem*	*Nem*	*10–30*	***Igen***

A többi nem jó jellemző

Részfában: $p_1 = 1$, $n_1 = 1$, $p_1/(p_1+n_1) = 1/2$, $n_1/(p_1+n_1) = 1/2$

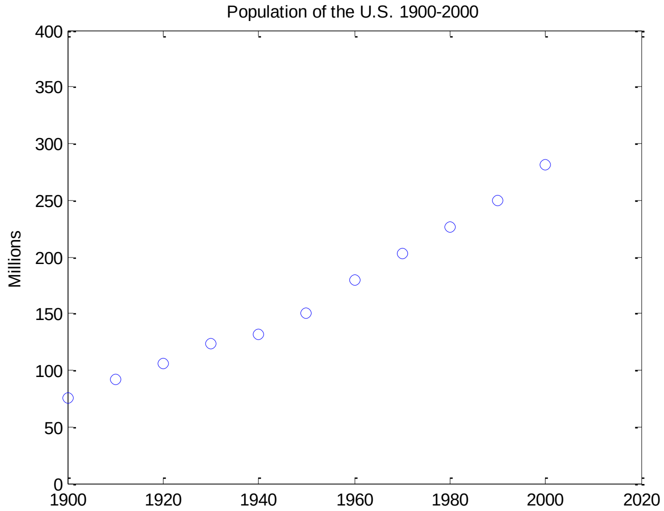
És ez ugyanúgy marad pl. a *Bár* = *Nem*, vagy *Eső* = *Nem*, vagy *Ár* = *Olcsó*, stb. mentén (azaz ezekbe az irányokba nem érdemes építeni a fát).

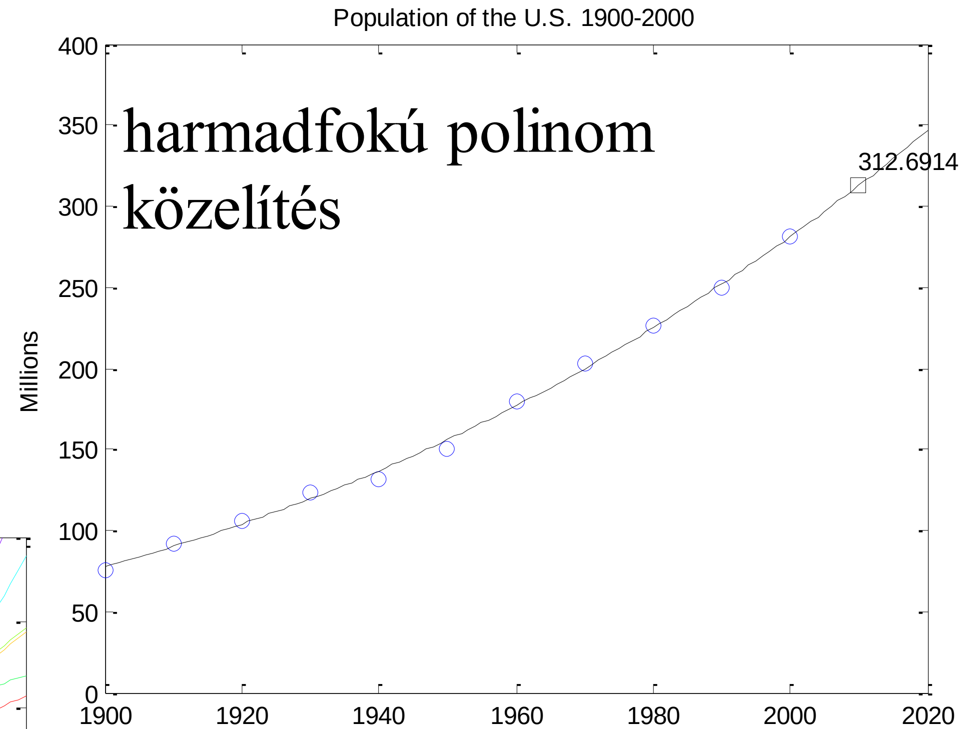
A kialakított döntési fa

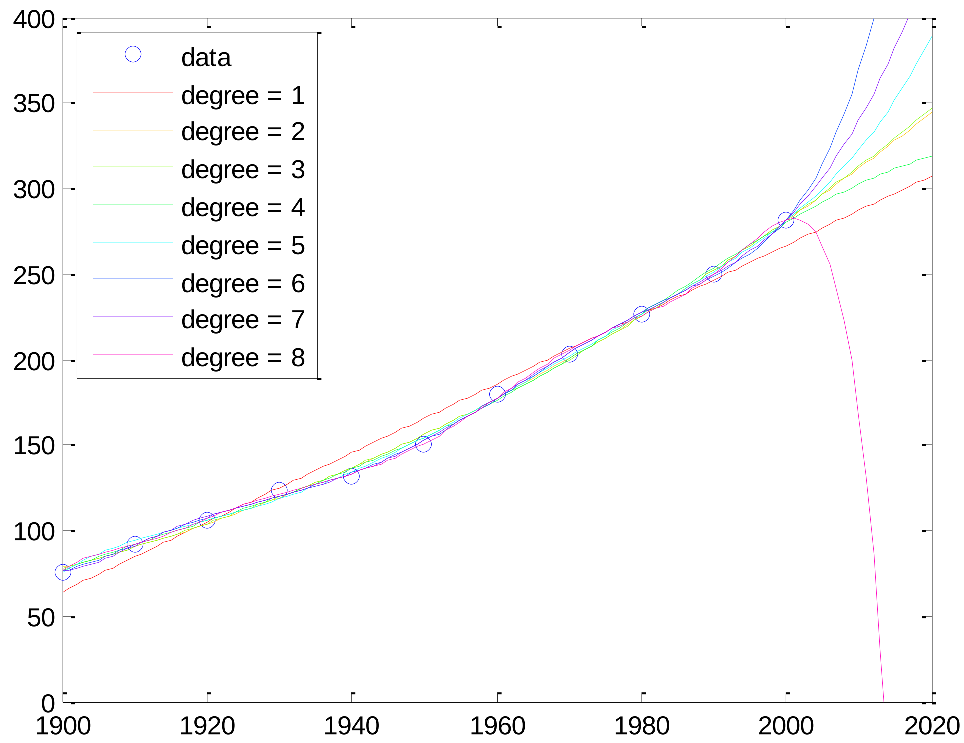


Ha pl. a Pént/Szomb helyett a Becs attribútumot választjuk – ugyanúgy **konzisztens** a kialakuló fa mindegyik példánkkal, **DE máshogy általánosít**!

A mintákat torzító zaj hatása – túltanulást okozhat!
(a faék egyszerűségű gondolkodás dicsérete! ☺)







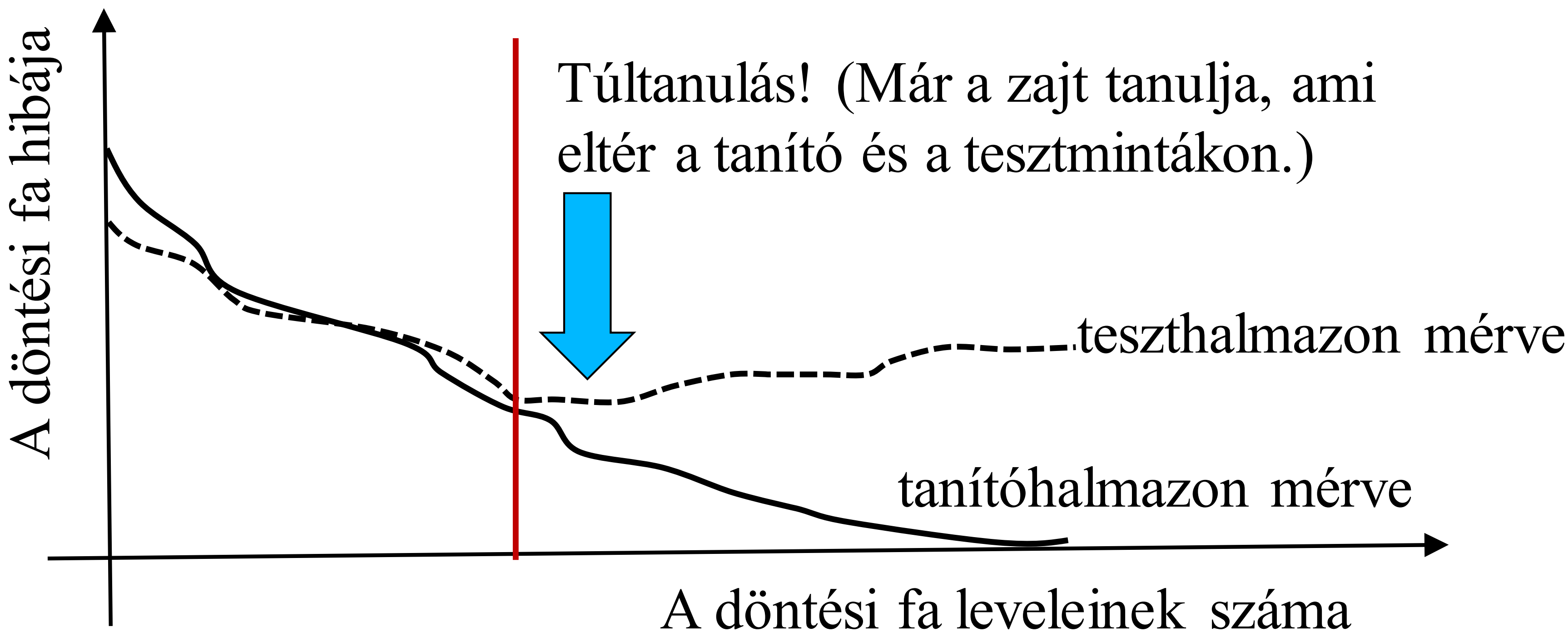
Az 1900-2000 közti népszámlálási adatokból becsüljük 2010-es USA népességet. (Regressziós feladat, de a jelenség a döntéseknél hasonló hatást okoz.)

A nagyon „okos” = túl komplex eszköz a zajt is megtanulja, rosszabbul becsül, mint az egyszerűbb!

Mikor álljunk le a döntési fa növesztésével?

Alapvetően 2 stratégia:

1. ***Korai leállás***



Ahogy növesztjük a fát, egyre több levele van.

(A levelekkel mérhetjük, hogy mekkorára nőtt a fa, mennyire komplex.)

Döntési fa nyesése (*pruning*):

(1) Beállított max. mélység (mint a mélységkorlátozott keresésnél), ha elérjük, akkor ezen a szinten mindenképpen leállunk, nem fejtjük ki tovább az adott ágat.

(2) Vizsgáljuk meg az elvégzett teszteket, ismerjük fel a nem releváns (semmitmondó) felhasznált attribútumot, és az adott ágat ott fejezzük be (metsszük vissza a fát).

Döntési fa nyesése

2. ***<u>Engedjük túlnőni, majd visszavágjuk, visszanyessük</u>***

Első lehetőség

Ismerjük fel ha nem releváns attribútumot használtunk, és az adott ágat ne fejtsük ki tovább, sőt messük le a kialakított részfát.

Irreleváns attribútum: példahalmaz kettévágásánál a kapott részhalmazok kb. azonos arányban tartalmaznak pozitív és negatív példát, mint az eredeti halmaz.
Az információ nyereség ilyenkor közel 0.

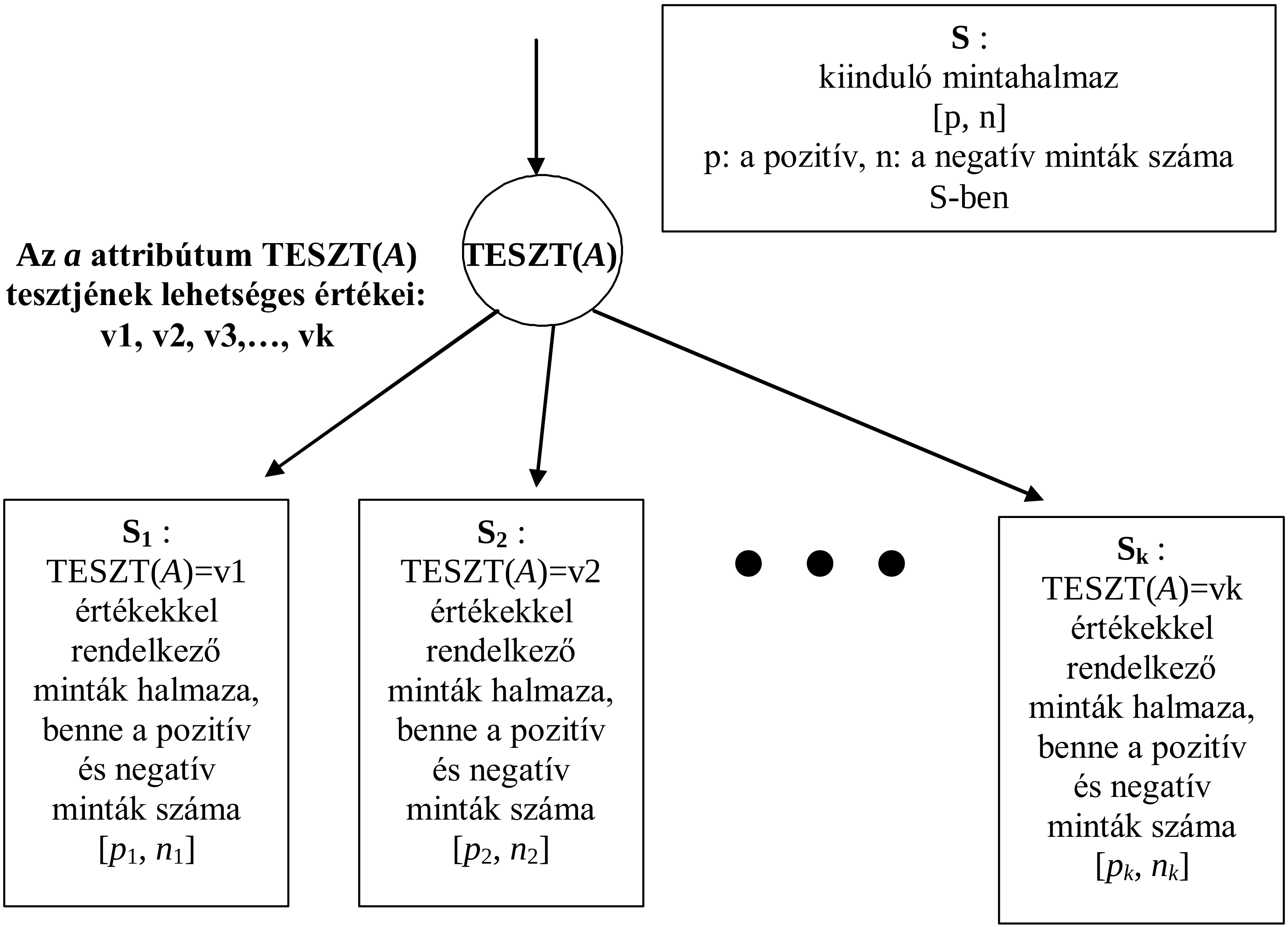
Az információ nyereség hiánya az irrelevancia jó jelzése.
Milyen nagy legyen az információ nyereség, hogy egy attribútumot mégis felhasználjunk a példahalmaz megosztására?

<u>Statisztikai tesztet végzünk</u>: **feltesszük, hogy nincs jellegzetes minta** = ún. **nullhipotézis**.

Statisztikai teszt – fordítva gondolkodunk, feltesszük, hogy nincs a halmazban jellegzetes minta = ún. **nullhipotézis**. Utána ezt megpróbáljuk cáfolni –vagy legalább belátni,hogy nagyon valószínűtlen!

Ha a nullhipotézis igaz, akkor elvileg 0 lenne az információnyereség, az ettől való eltérés csak a véletlen műve.

Ha az eltérés **nagy** (jóval nagyobb, mint 0) $\Rightarrow$ **nem valószínű**, hogy az észlelt eltérés véletlen ingadozásból következik $\Rightarrow$
a nullhipotézis nem valószínű: **vélhetőleg lényeges minta** van az adatokban. (Valamilyen valószínűséggel cáfolni tudjuk a nulla hipotézist.)


S :
kiinduló mintahalmaz
[p, n]
p: a pozitív, n: a negatív minták száma
S-ben
Az *a* attribútum TESZT(*A*)
tesztjének lehetséges értékei:
v1, v2, v3,…, vk
TESZT(*A*)
$\mathbf{S_1}$:
TESZT(*A*)=v1
értékekkel
rendelkező
minták halmaza,
benne a pozitív
és negatív
minták száma
[p_1, n_1]
$\mathbf{S_2}$:
TESZT(*A*)=v2
értékekkel
rendelkező
minták halmaza,
benne a pozitív
és negatív
minták száma
[p_2, n_2]
$\mathbf{S_k}$:
TESZT(*A*)=vk
értékekkel
rendelkező
minták halmaza,
benne a pozitív
és negatív
minták száma
[p_k, n_k]

$$\hat{p}_i = \frac{p_i + n_i}{p + n}\, p \qquad \hat{n}_i = \frac{p_i + n_i}{p + n}\, n$$

$$D = \sum_{i=1}^{k} \left(\frac{(p_i - \hat{p}_i)^2}{\hat{p}_i} + \frac{(n_i - \hat{n}_i)^2}{\hat{n}_i} \right)$$

A *D* változó χ^2 eloszlást követ. Adott *k* (lehetséges teszteredmények száma) mellett a táblázat azt mutatja, hogy mi annak a valószínűsége, hogy a kiszámított relatív eltérés (*D*) eléri a megadottat, bár valójában irreleváns dolgot teszteltünk, ez csupán a véletlen műve:

SzF = *k*-1	$P_{0,5}$	$P_{0,2}$	$P_{0,1}$	$P_{0,05}$	$P_{0,01}$	$P_{0,005}$	$P_{0,001}$
1	0,46	1,64	2,71	3,84	6,64	7,88	10,83
2	1,39	3,22	4,61	5,99	9,21	10,60	13,82
3	2,37	4,64	6,25	7,82	11,35	12,84	16,27
4	3,36	5,99	7,78	9,49	13,28	14,86	18,47
5	4,35	7,29	9,24	11,07	15,09	16,75	20,52
6	5,35	8,56	10,65	12,59	16,81	18,55	22,46
7	6,35	9,80	12,02	14,07	18,48	20,28	24,32
8	7,34	11,03	13,36	15,51	20,09	21,96	26,12
9	8,34	12,24	14,68	16,92	21,67	23,59	27,88
10	9,34	13,44	15,99	18,31	23,21	25,19	29,59

Például: 400-at dobtunk pénzérmékkel és ebből 192 fej lett, 208 írás. Kiderült, hogy

- ebből 200-at egy adott százforintossal dobtunk, ebből 88 a fej, 112 az írás,
- utána adott húszforintossal dobtunk 200-at, ebből 104 a fej, 96 az írás.

Állíthatjuk-e 99% biztonsággal, hogy az érmék különbözőek (a fej/írás kimenetel szempontjából), számít, hogy melyikkel dobunk?

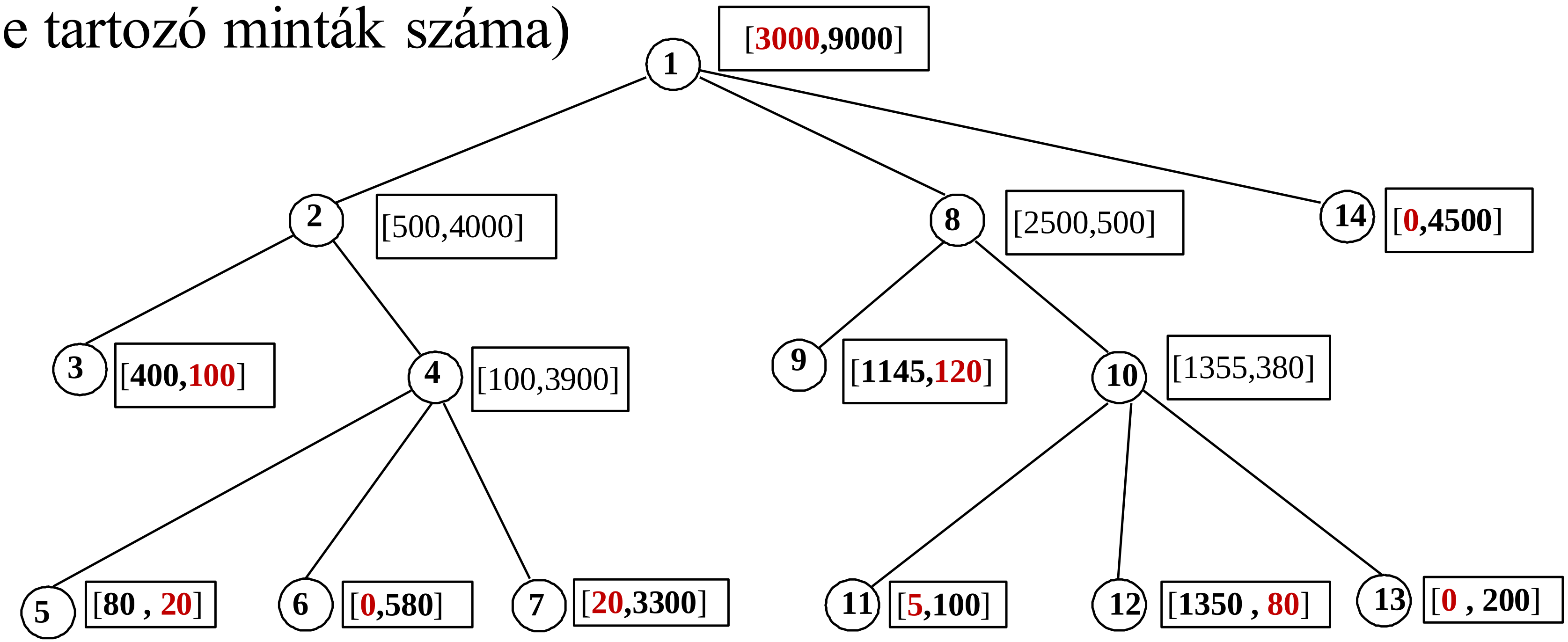
$\Rightarrow D$=2,56

Döntési fa nyesése: ***második lehetőség***

Hibaarány – komplexitás kompromisszumon alapuló metszés

Egy kétosztályos (bináris) osztályozási példa kapcsán

(Baloldali szám a C1 osztályba tartozó minták száma, jobboldali a C2-be tartozó minták száma)



Komplexitás – legyen a levelek száma (lehetne más is), a példánkban: a gyökérnél 1, a kifejtett fánál 9

Hibaarány:

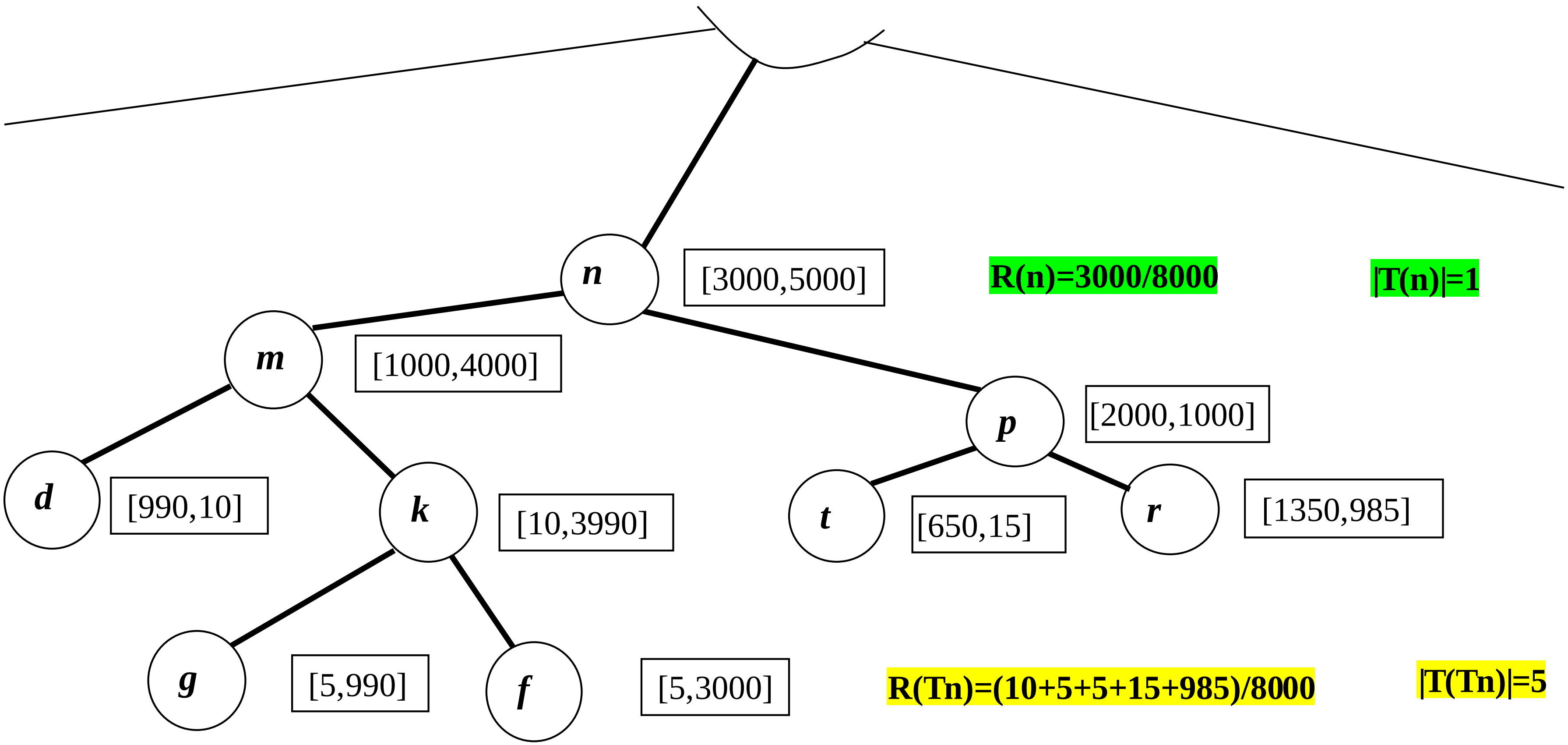
a gyökérnél 3000/12000 (hiszen a jobb válaszunk C2 lenne),

a kifejtett fánál: (100+20+0+20+120+5+80+0+0)/12000

Hibaarány – komplexitás kompromisszumon alapuló metszés

Számozzuk a csomópontokat, jelölés:

- az *n*-dik csomópontot önmagában (mintha levél lenne) jelölje *n*
- az *n*-edik csomópontból kiinduló részfa T*n*



Hogyan hasonlítsuk össze a komplexitást a hibával?!?

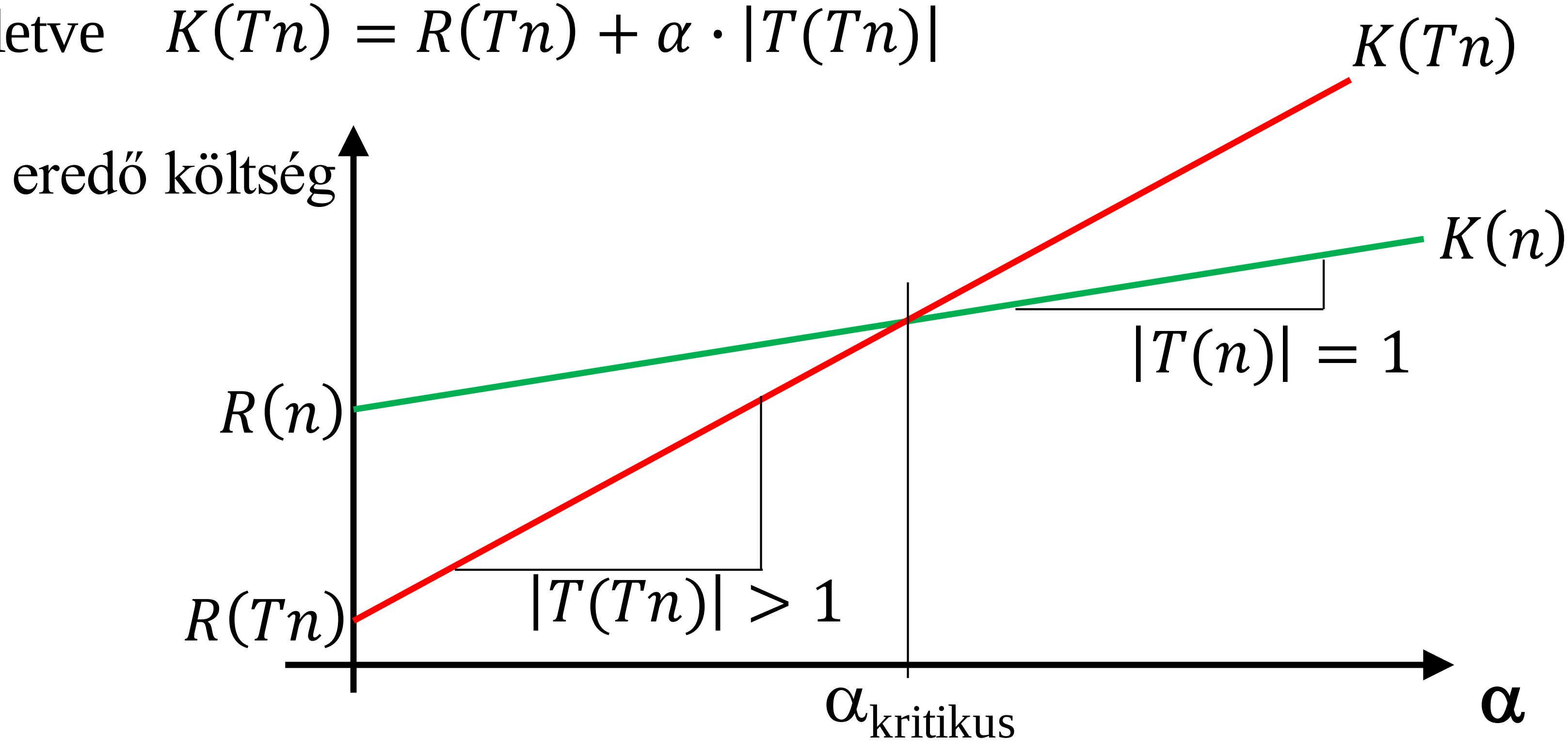
Legyen az összetett, eredő költség (pl. az n-dik csomópontra):

$$K^*(n) = EgységnyiHibaKöltsége \cdot R(n) + EgységnyiKomplexitásKöltsége \cdot |T(n)|$$

Legyen a két költség aránya: $\alpha = \dfrac{EgységnyiKomplexitásKöltsége}{EgységnyiHibaKöltsége}$

Így $K(n) = \dfrac{K^*(n)}{EgységnyiHibaKöltsége} = R(n) + \alpha \cdot |T(n)|$

illetve $K(Tn) = R(Tn) + \alpha \cdot |T(Tn)|$



$\alpha_{kritikus}$ amikor mindegy, hogy levágjuk-e a részfát, az összetett költség azonos az *n*-edik csomópontra, mint levélre, és az *n*-dik csomópontból kiinduló *Tn* részfára

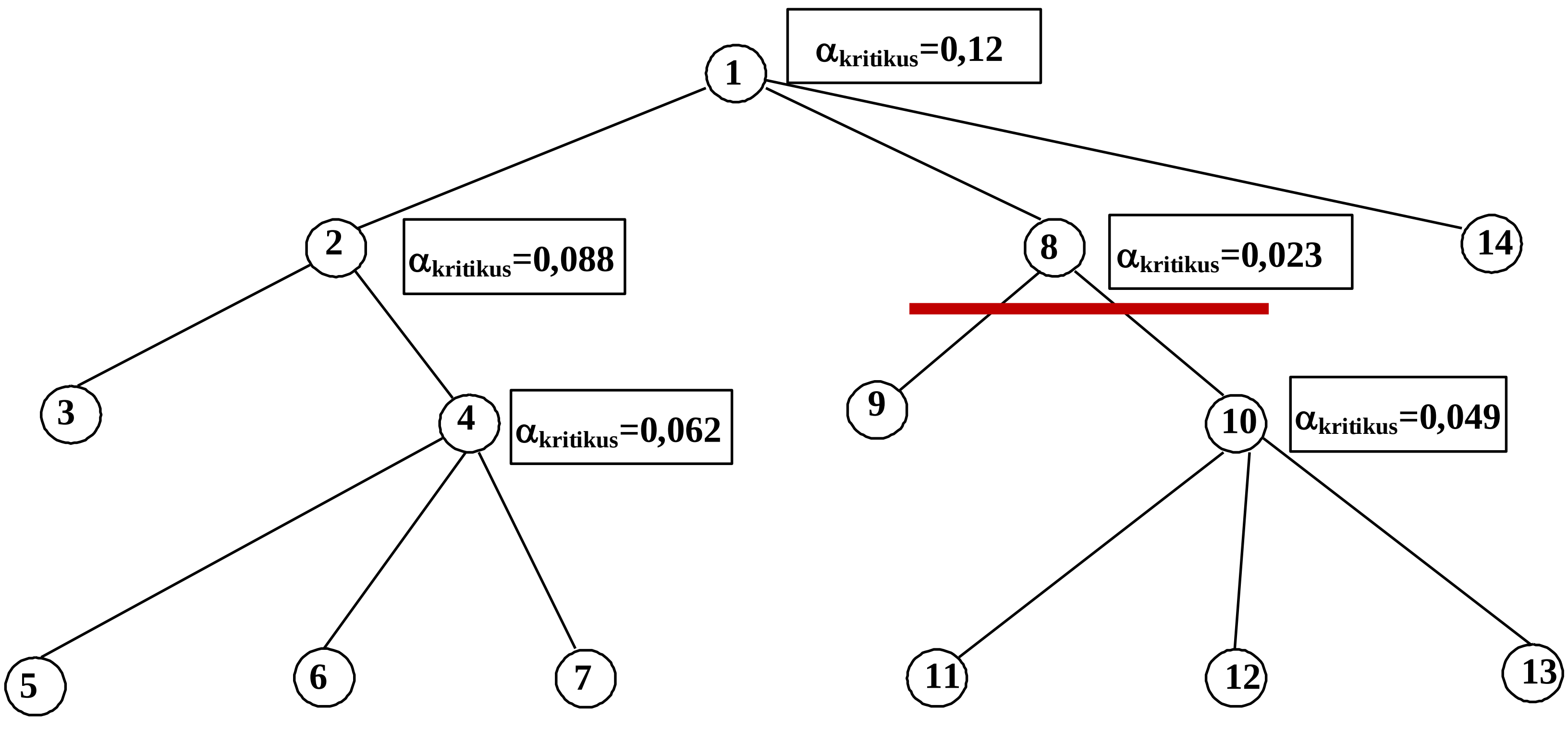
$$R(n) + \alpha_{kritikus} \cdot |T(n)| = R(Tn) + \alpha_{kritikus} \cdot |T(Tn)|$$

$$\alpha_{kritikus} = \frac{R(n) - R(Tn)}{|T(Tn)| - |T(n)|} = \frac{R(n) - R(Tn)}{|T(Tn)| - 1}$$

1. Ha $\alpha=0$, akkor a komplexitásnak nincs ára, soha nem érdemes visszametszeni
2. $0 < \alpha < \alpha_{kritikus}$, akkor még mindig jobb nem visszametszeni
3. Ha $\alpha = \alpha_{kritikus}$, akkor mindegy, hogy visszavágjuk-e
4. Ha $\alpha_{kritikus} < \alpha$, akkor érdemes visszavágni, olcsóbb abbahagyni az *n*-dik levélnél, nem kifejteni az innen induló részfát

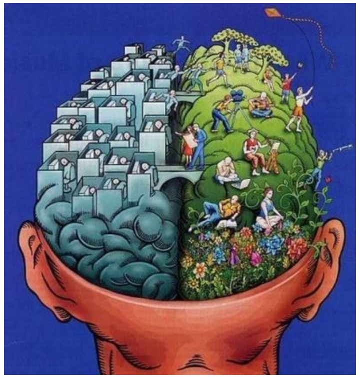
Gondoljuk végig, hogy mi történik, ha **α**–t folyamatosan növeljük 0-tól indulva!

amelyik csomópont $\boldsymbol{\alpha}_{\mathbf{kritikus}}$ értékét először érjük el, ott érdemes *leginkább* metszeni a fát! (Persze ha a hiba nem nő túl nagyra)



A számok légből kapottak (csak a példa kedvéért)

Budapesti Műszaki és Gazdaságtudományi Egyetem
Méréstechnika és Információs rendszerek Tanszék

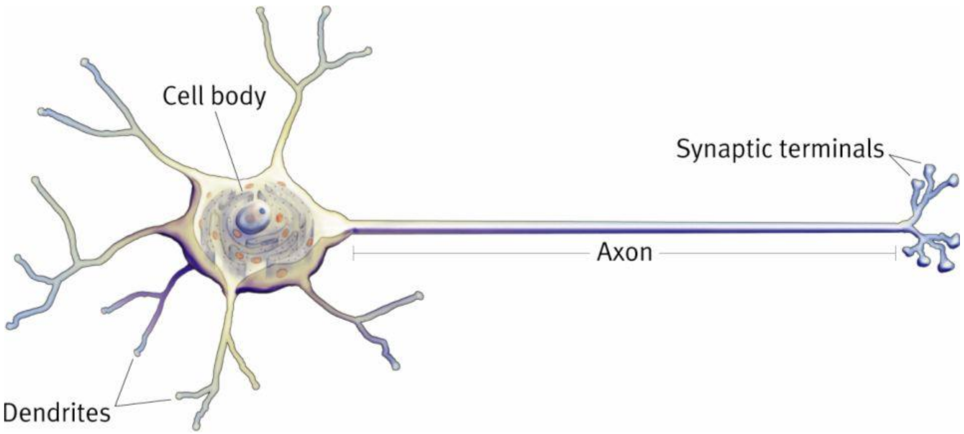


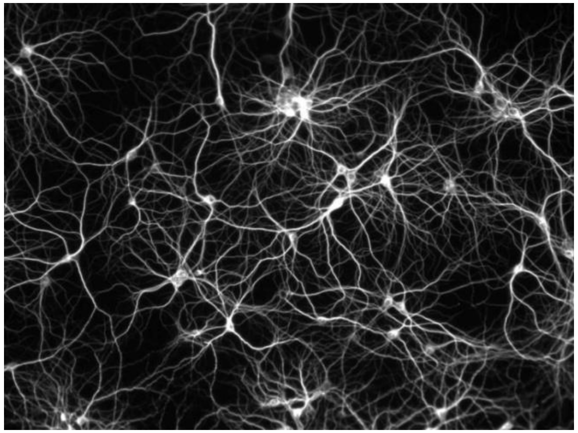
Neurális hálók

Előadó:	Dr. Hullám Gábor
Előadás anyaga:	Dr. Dobrowiecki Tadeusz Dr. Pataki Béla

MŰEGYETEM 1782

Neurális hálózat - neuronok



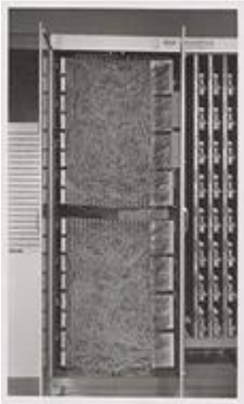


- Neuron doktrina: S. Ramón y Cajal (1852-1934)
- Mesterséges neuron: W. McCulloch and W. Pitts, 1943
- Tanulás: D. Hebb, 1949
- SNARC (Stochastic Neural Analog Reinforcement Calculator): M. Minsky, 1951
- Perceptron: F. Rosenblatt, 1957

•Neuronok becsült száma az emberi agyban: F. Azevedo et al., 2009
86.1 +/- 8.1 milliárd

Perceptron - történelem

- 1957 - Frank Rosenblatt: Perceptron
  - A perceptron algoritmus első implementációja a Mark I Perceptron gép



- 1969: Minsky és Papert: Perceptrons c. könyve
  - A perceptronnak súlyos korlátai vannak a képességeit tekintve
  - Hatására kb 10 évig nem történt kutatás a témakörben (AI Winter)

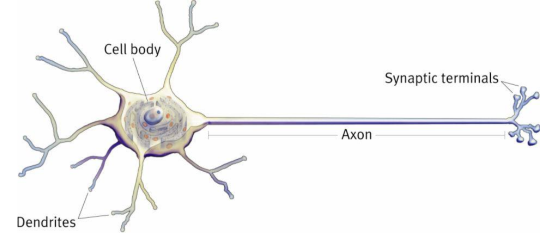
Klasszikus neurális hálózatok

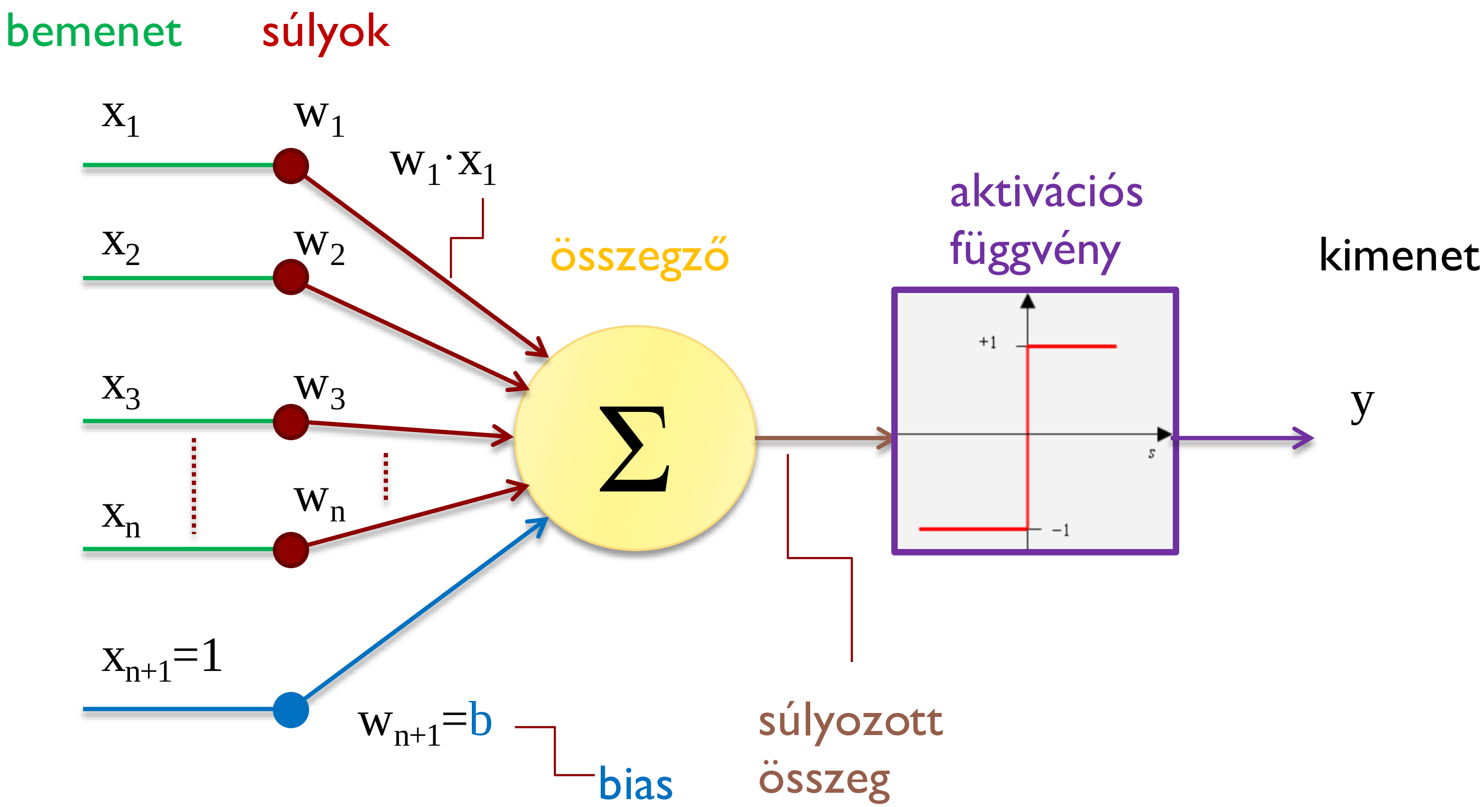
- 1980 - 1986 - David E. Rumelhart, Geoffrey E. Hinton and Ronald J. Williams: "Learning representations by back-propagating errors"
  - Hibavisszaterjesztés algoritmusa
  - Újra beindította a kutatást a témakörben
- 1995 - Vapnik: Support-vector networks
  - Számos problémára sokkal jobb eredményt ad, kevésbé nehézkes a tanítása
  - Hatásásra a neurális hálózatok kutatása (ismét) alábbhagyott

Mély neurális hálózatok

- 2000-es évek közepén a neurális hálózatok témakörben alig lehetett cikket elfogadtatni
- 2004 – G. Hinton - CIFAR (Canadian Institute for Advanced Research)
  - Új csomagolás a neurális hálózatoknak: **deep learning**
- 2006 - Hinton, Osindero, Yee-Whye Teh: A fast learning algorithm for deep belief nets
  - Új áttörés: mély hiedelem hálók rétegenkénti tanítása,
  - Újra beindítja a kutatást

Perceptron





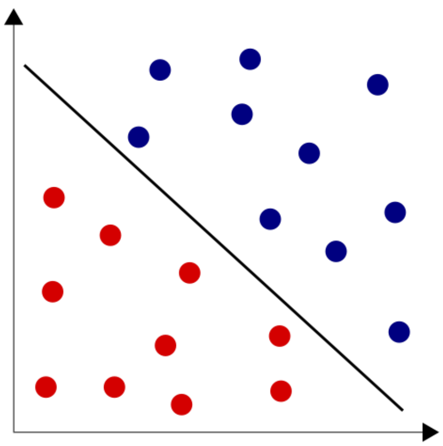
$$y = \mathrm{sgn}\Big(\sum_{i=1}^{n+1} w_i \; x_i\Big) = \mathrm{sgn}\Big(\sum_{i=1}^{n} w_i \; x_i + w_{n+1}\Big) = \mathrm{sgn}(\mathbf{x}^T \mathbf{w} + b)$$

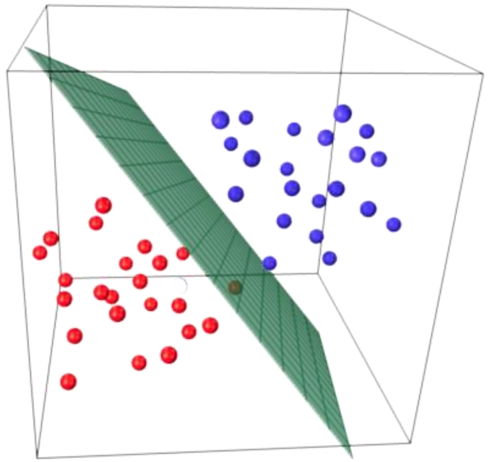
A perceptron képessége

$$y = \mathrm{sgn}(\sum_{i=1}^{n+1} w_i \ x_i) = \mathrm{sgn}(\sum_{i=1}^{n} w_i \ x_i + w_{n+1}) = \mathrm{sgn}(\mathbf{x}^T \mathbf{w} + b)$$

$$\mathbf{x}^T \mathbf{w} + b > 0$$

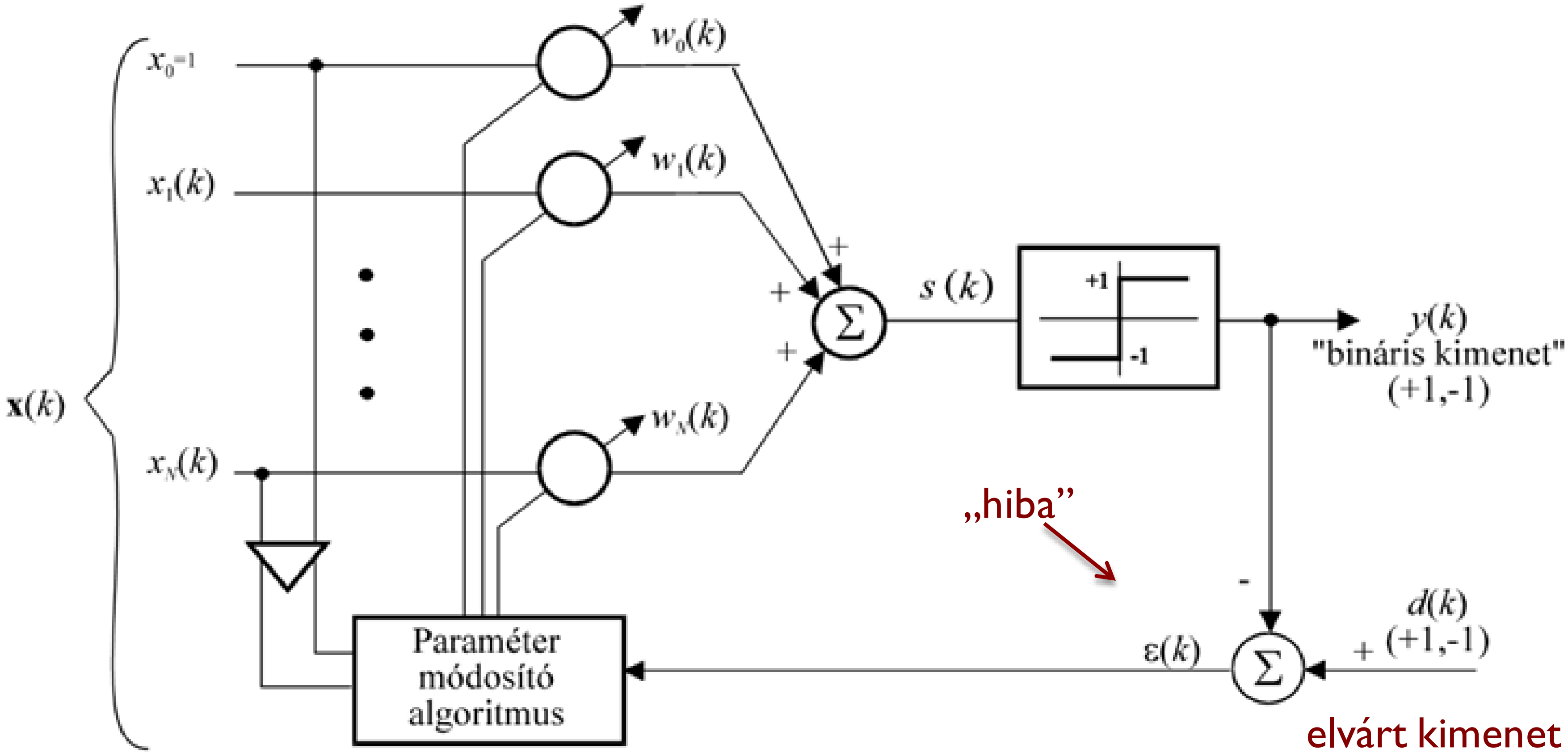
$$\mathbf{x}^T \mathbf{w} + b < 0$$





lineárisan (hipersíkkal) szeparálható függvényt képes reprezentálni

Perceptron tanítása 1.



$s(k) = \mathbf{w}^T\mathbf{x}(k)$ $\quad y(k) = \mathrm{sgn}(s(k))$ $\quad \varepsilon(k) = d(k) - y(k)$

$$\mathbf{w}(k) = \mathbf{w}(k-1) + \alpha\,\varepsilon(k)\,\mathbf{x}(k)$$

α - **bátorsági faktor,**
tanulási tényező

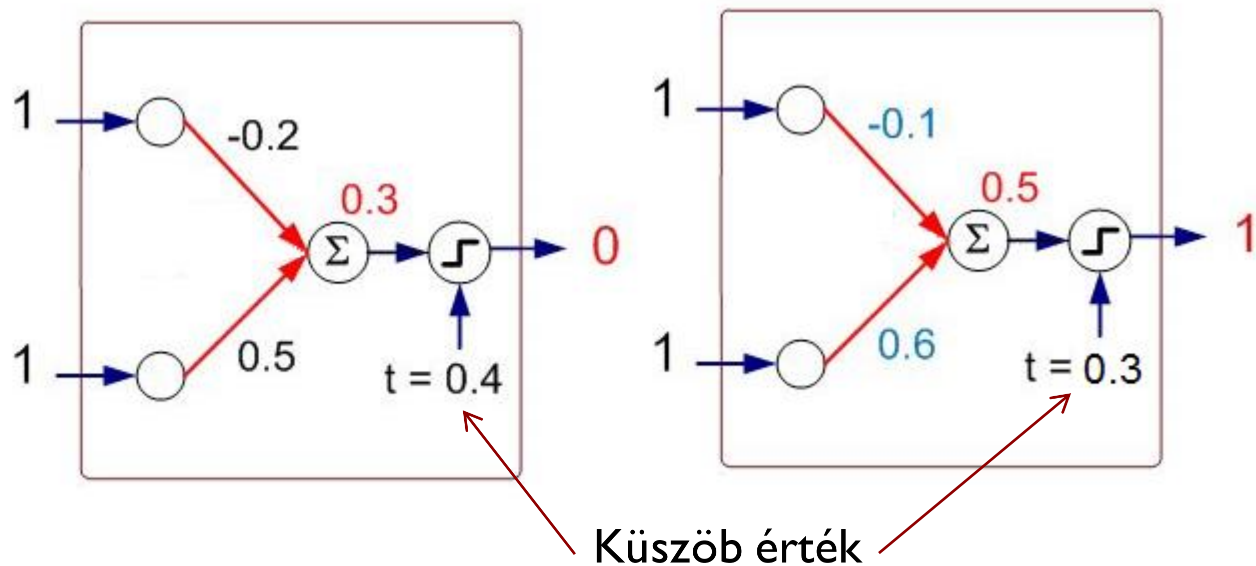
Perceptron tanítása 2.

Tanítás konvergens, ha
- A tanító példák lineárisan szeparálhatók
- Bátorsági tényező elegendően kicsi

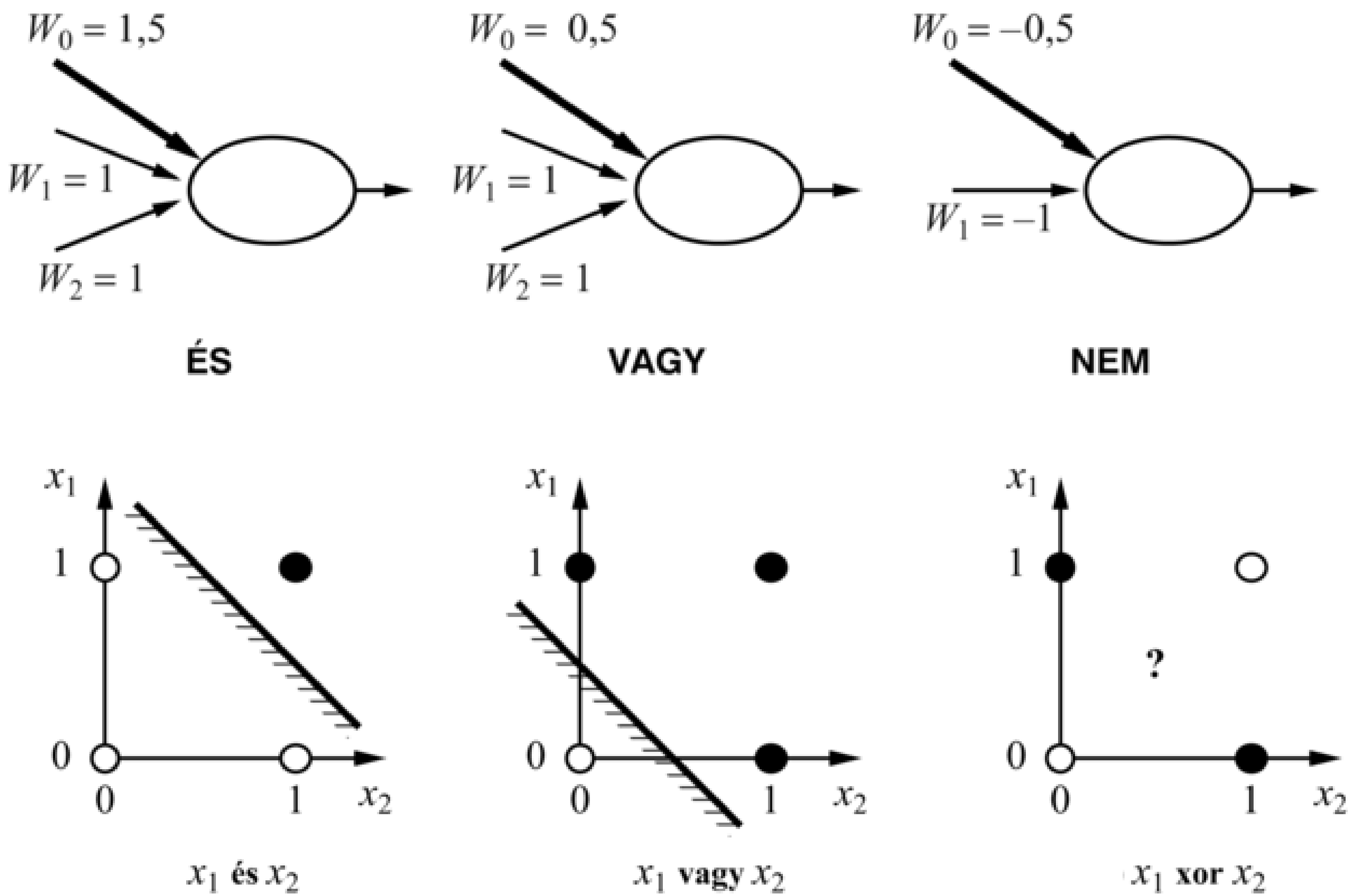
$$\mathbf{w}(k) = \mathbf{w}(k-1) + \alpha\,\varepsilon(k)\,\mathbf{x}(k)$$

$$\eta = 0.1$$

X_1	X_2	Y
1	1	1

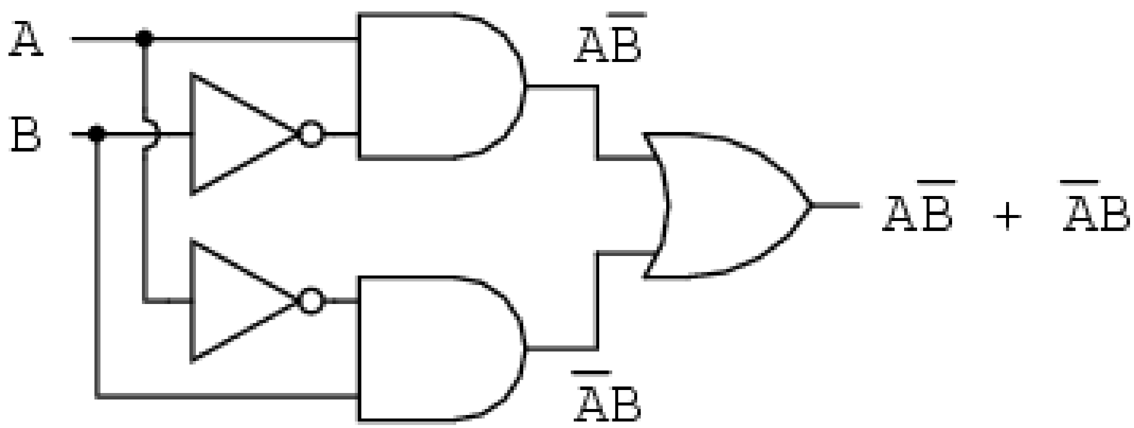


Perceptron és a logikai függvények



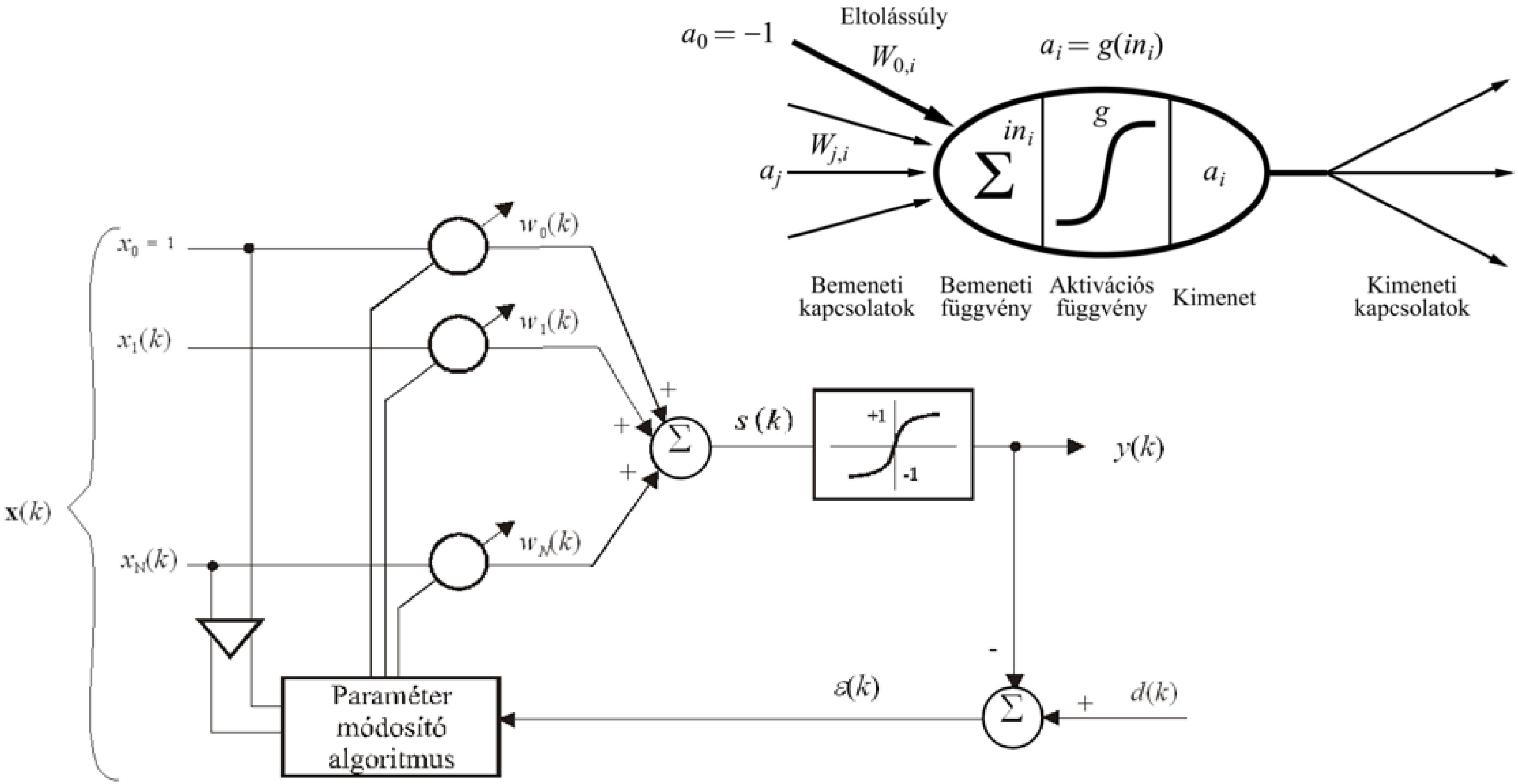
XOR (és hasonlóan lineárisan nem szeparálhatók) problémája

- több réteg kellene
- tanulás (preceptronnál) csak 1 rétegre
  - hiba értelmezése a kérdés



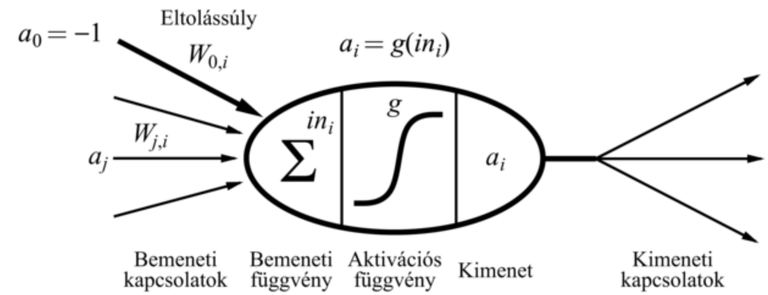
$$\mathbf{A \oplus B = A\overline{B} + \overline{A}B}$$

Mesterséges neuron

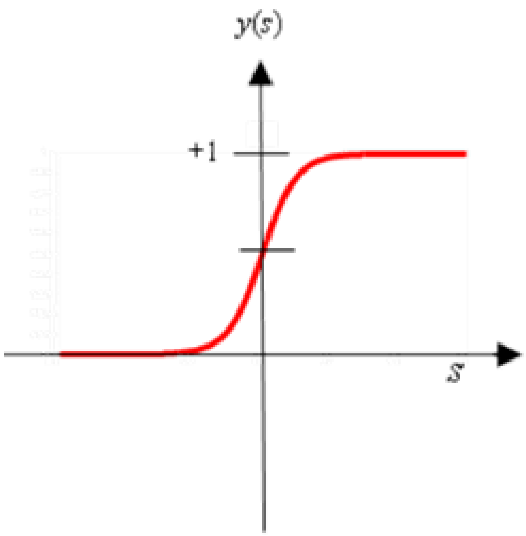


$s(k) = \mathbf{w}^T\mathbf{x}(k)$ $\quad y(k) = \text{sgm}(s(k))$ $\quad \varepsilon(k) = d(k) - y(k)$

Mesterséges neuron – aktivációs függvények

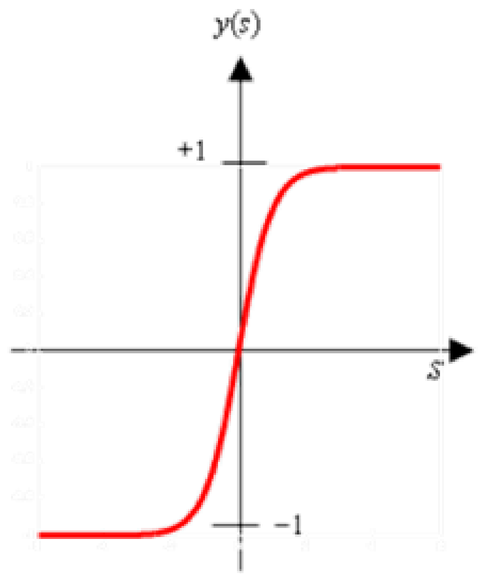


Szigmoid

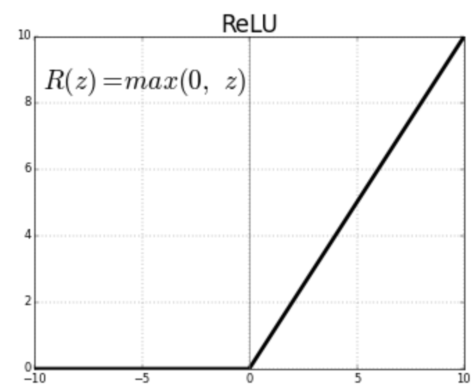


$$y = \frac{1}{1+e^{-Ks}}; \quad K > 0$$

Tanh

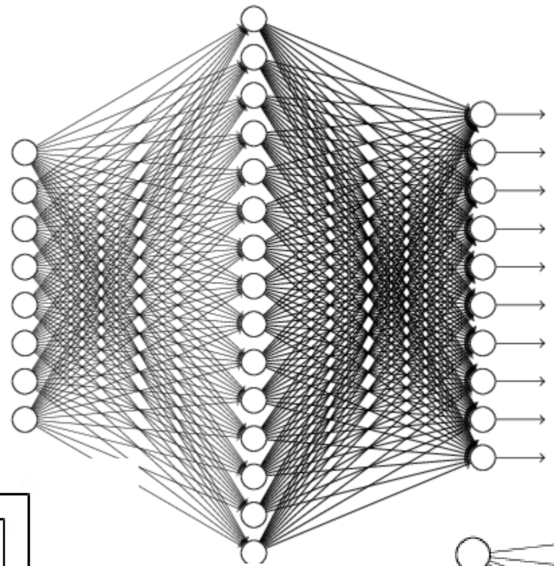


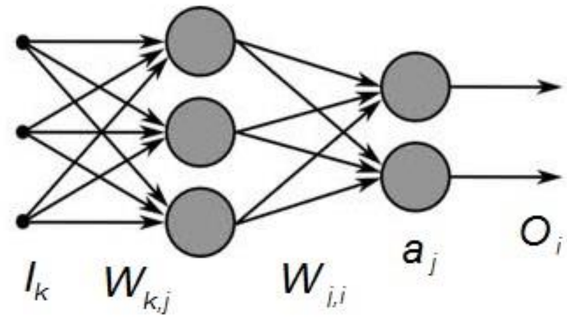
$$y = \frac{1-e^{-Ks}}{1+e^{-Ks}}; \quad K > 0$$

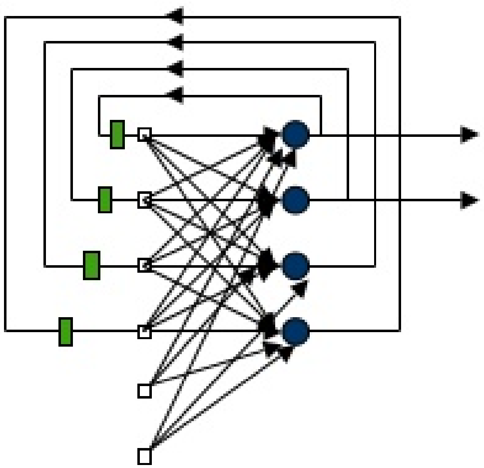


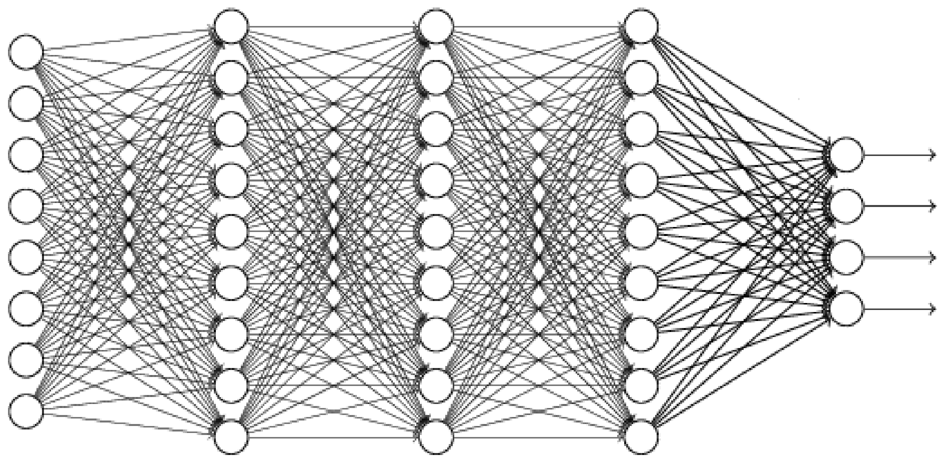
Mesterséges Neurális Háló (Artificial Neural Network)

nemlineáris approximációt megvalósító, induktív tanulási algoritmussal tanítható matematikai struktúra.

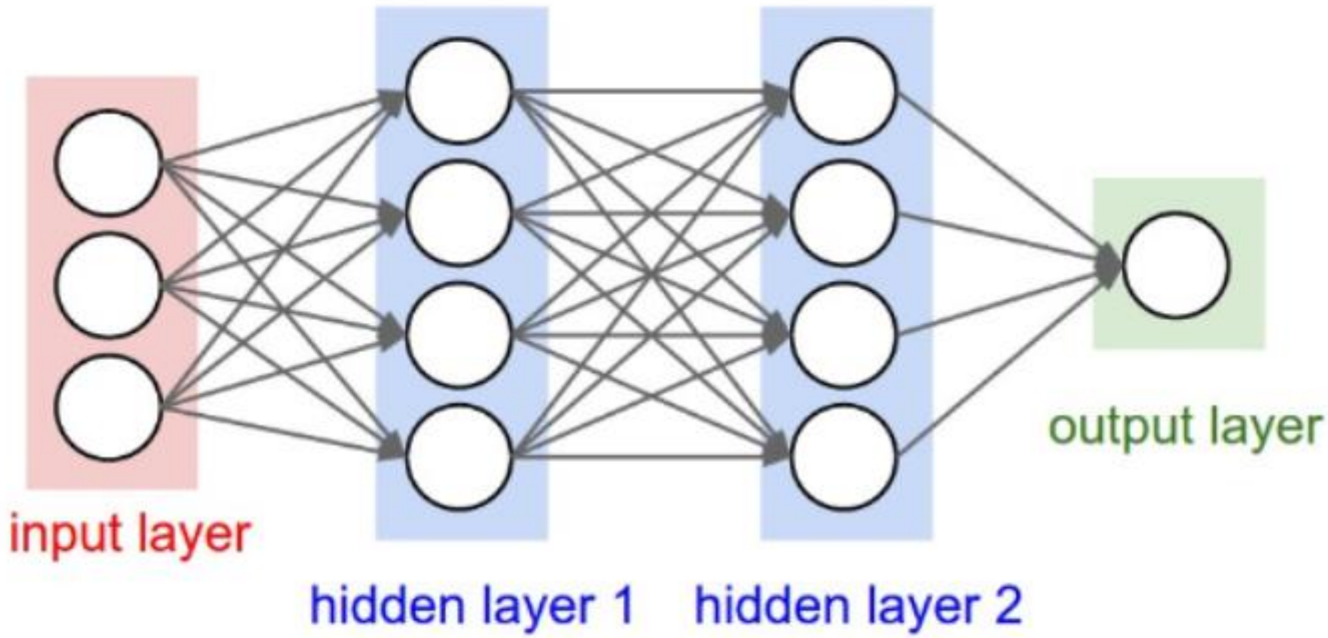








Mesterséges neurális háló felépítés

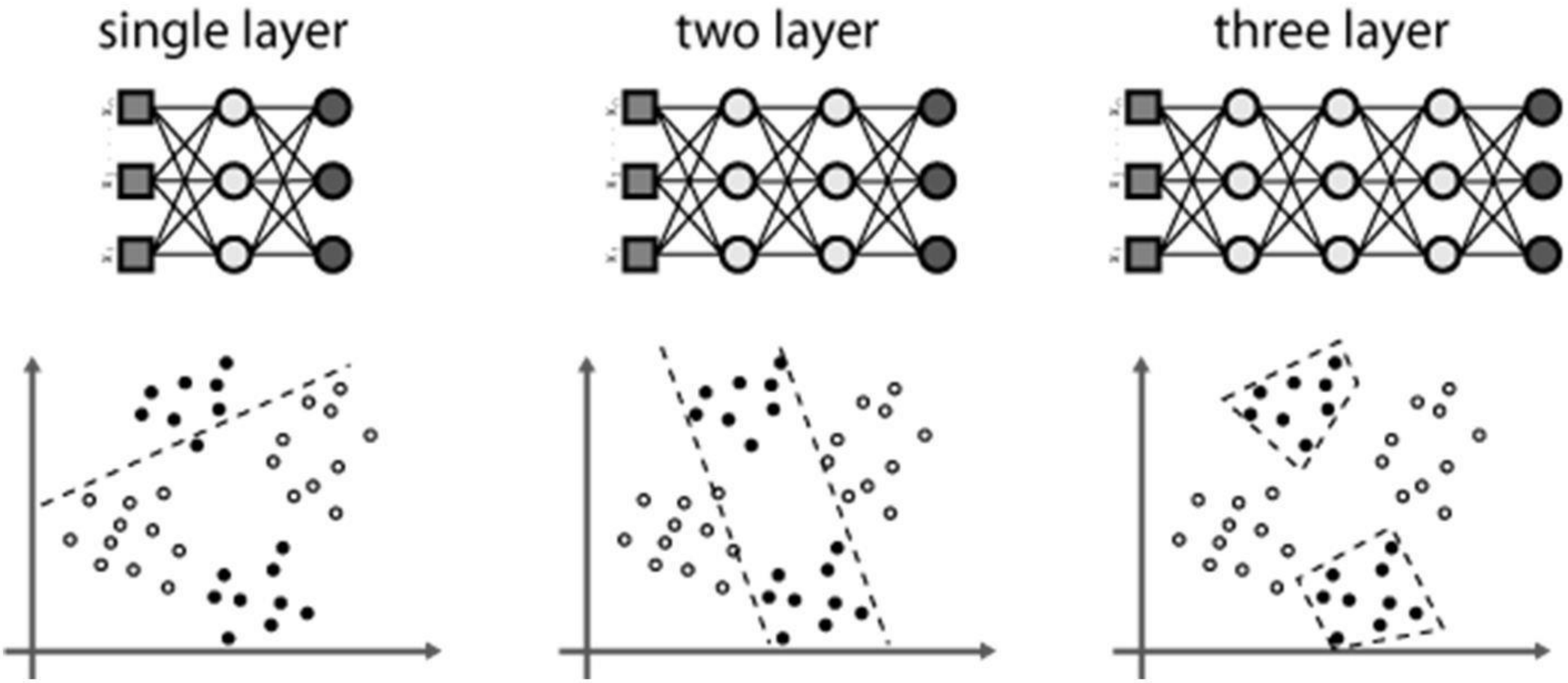


Bemeneti réteg

Rejtett rétegek

Kimeneti réteg

Mesterséges neurális háló



Mesterséges neurális háló reprezentációs képessége

D. Hilbert (1900) 23 matematikai problémája/sejtése:
13. probléma: "Bizonyítsuk be, hogy az $x^7+ax^3+bx^2+cx+1=0$ hetedfokú egyenlet nem oldható meg pusztán kétváltozós függvények segítségével!„

…

"Mutassuk meg, hogy van olyan háromváltozós folytonos függvény, mely nem írható fel véges számú kétváltozós folytonos függvény segítségével!„

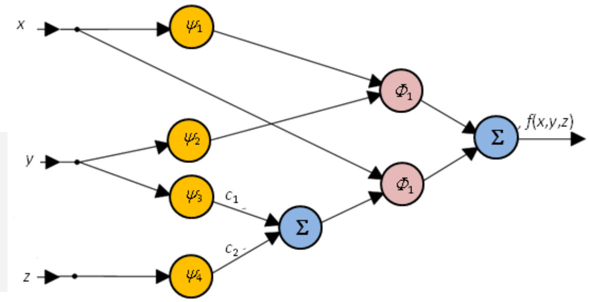
…

A. Kolmogorov (1957): nem csupán minden háromváltozós függvény, hanem tetszőleges *N*-változós folytonos függvény felírható csupán **egy**változós függvények és az összeadás segítségével.

…

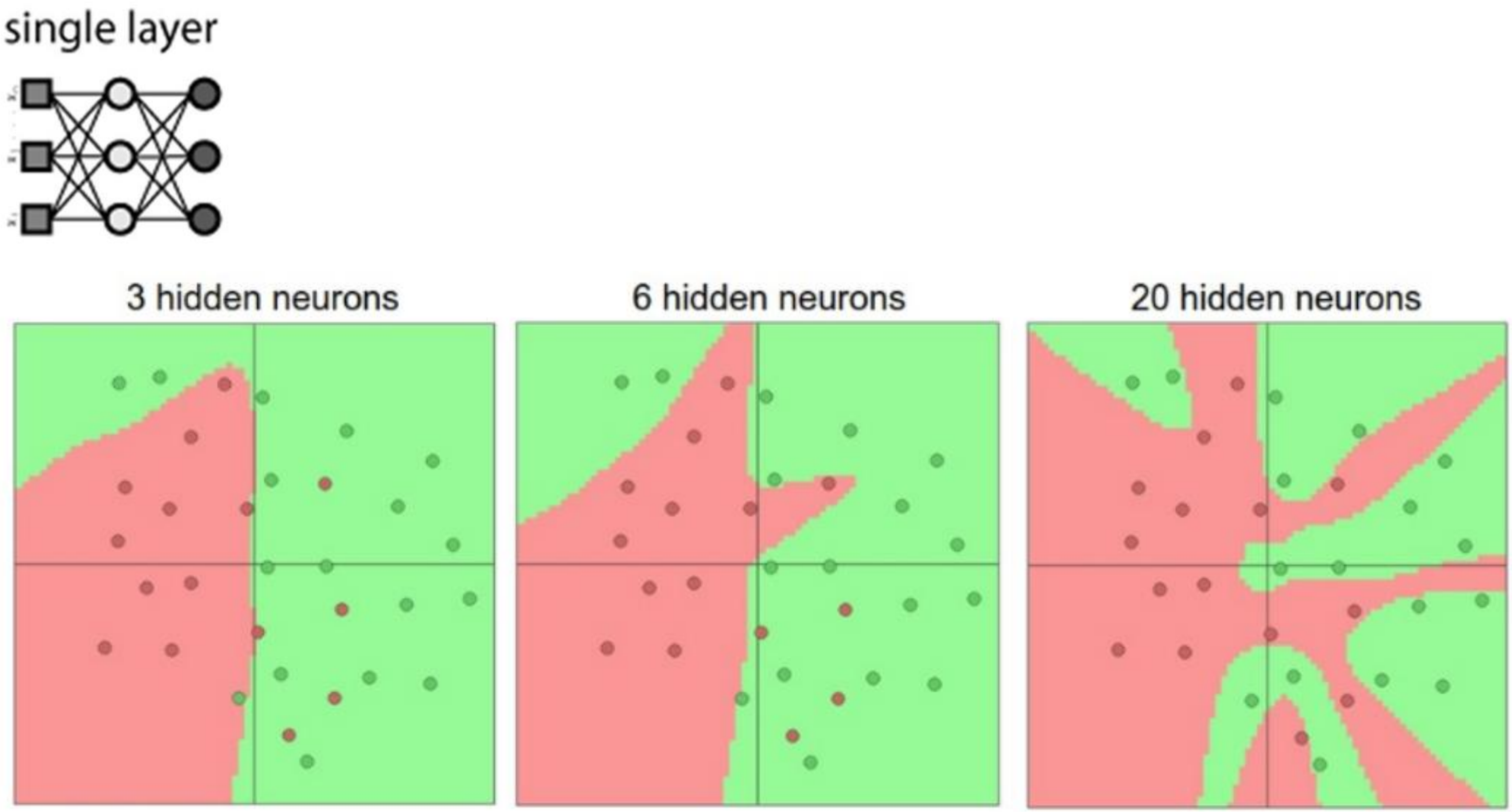
http://mialmanach.mit.bme.hu/neuralis/ch01s04

$$f(x_1,x_2,\ldots,x_N)=\sum_{q=1}^{2N+1}\phi_q\left(\sum_{p=1}^{N}\psi_{pq}(x_p)\right)$$



Mesterséges neurális háló

- Már egy rejtett réteggel is univerzális approximátor (bármely függvényt képes reprezentálni)



S.Sun, Univ.Toronto

Mesterséges neurális háló tanítása

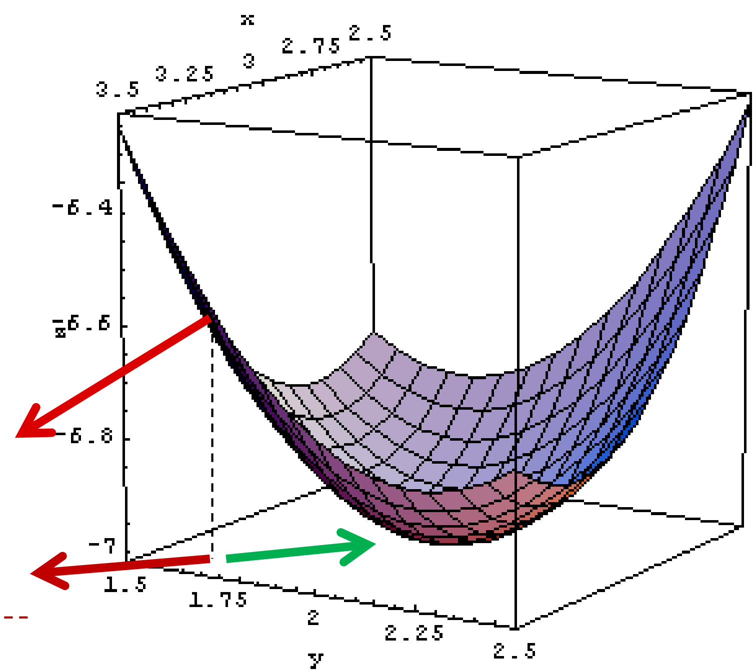
Többrétegű előrecsatolt háló tanítása (elemi alapok)

Hibavisszaterjesztés gradiens módszerrel

- példa bemeneteket mutatunk a hálónak,
- ha hiba lép fel (a kimenet és a kívánt érték eltér), a súlyokat úgy módosítjuk, hogy a hiba csökkenjen.

A trükk a hiba megállapítása és a hibának a hibát létrehozó súlyok közti szétosztása.

$$W \leftarrow W - \alpha \frac{\partial E}{\partial W}$$



Hiba-/költség-/veszteség-függvény választás

- Négyzetes hiba – regresszió

$$E(\underline{d}, \underline{y}) = \frac{1}{2} \sum_i (d_i - y_i)^2$$

- Keresztentrópia – osztályozás

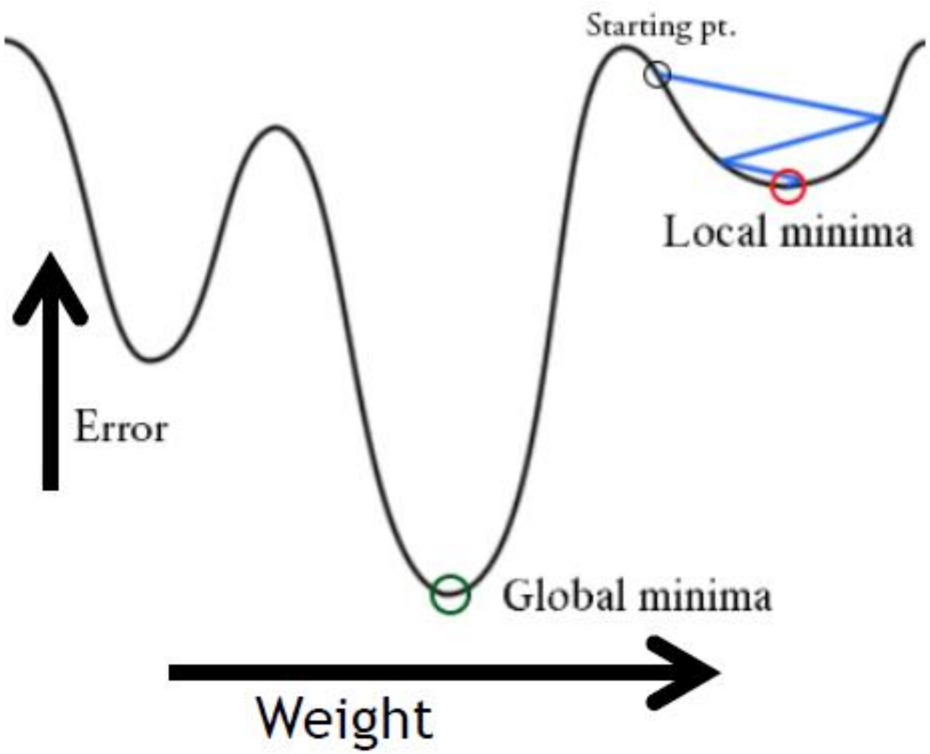
$$E(\underline{d}, \underline{y}) = -\sum_i d_i \log y_i$$

- A cél azon súlyok megtalálása, melyek mellett minimális a hiba

$$w^* = \arg\min_w \sum_{i=1}^{N} E(d_i, y_i)$$, ahol $\underline{y} = f(\underline{x}, \underline{w})$

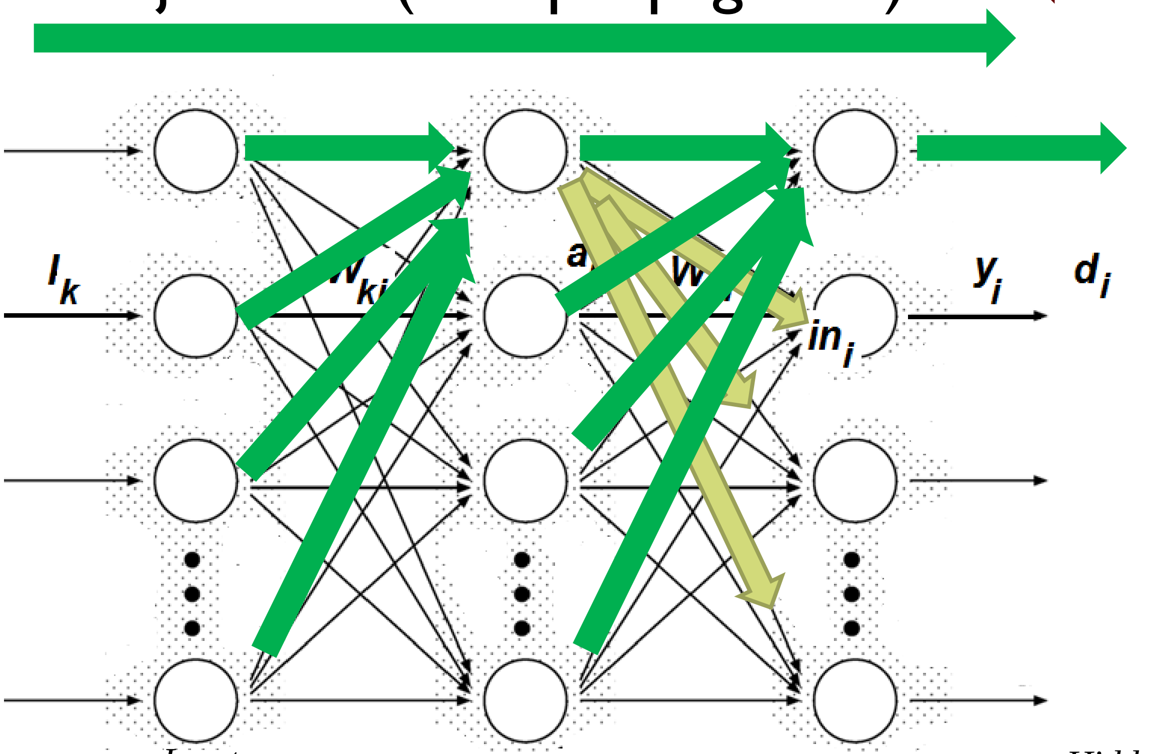
A tanítás nehézségei

- Kihívás: a neurális hálók tanítása egy nem konvex optimalizálási feladat



A tanítás lépései

- Kimenet számítása (forward pass)
- Hiba visszaterjesztése (backpropagation)

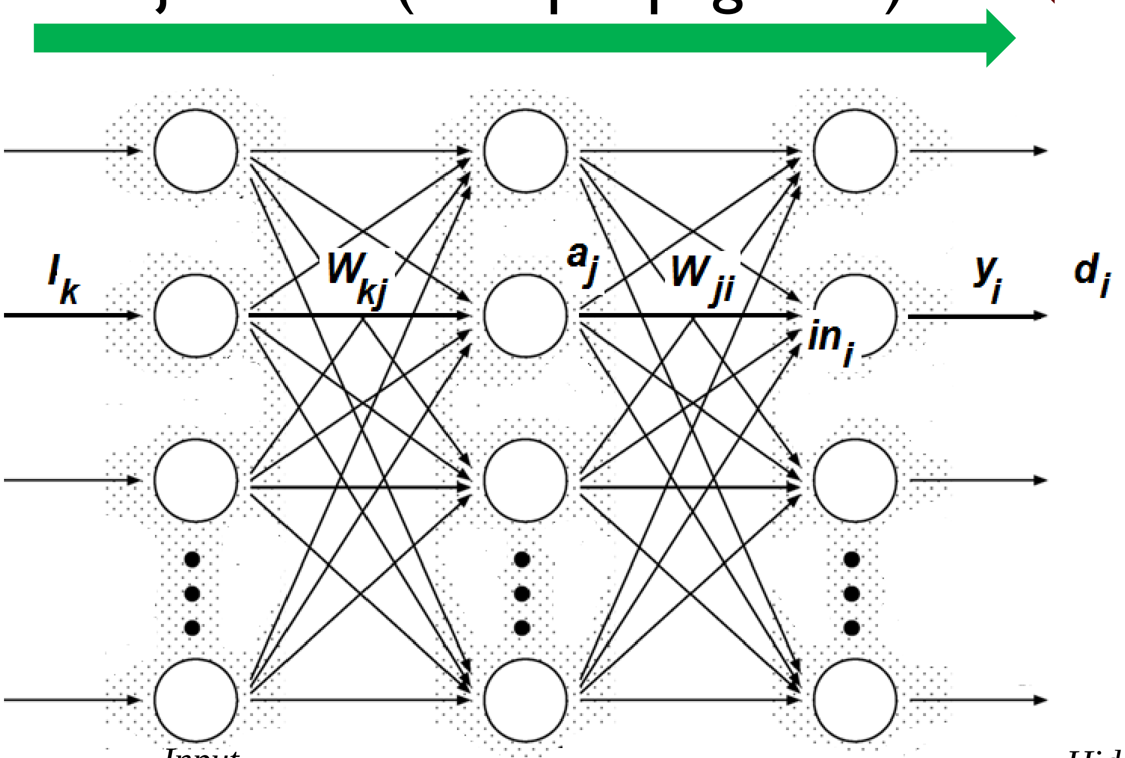


$$a_j(\underline{x}) = g(w_{k0} + \sum_{k=1}^{Input} x_k \cdot w_{kj})$$

$$y_i(\underline{x}) = g^*(w_{j0} + \sum_{j=1}^{Hidden} a_j(\underline{x}) \cdot w_{ji})$$

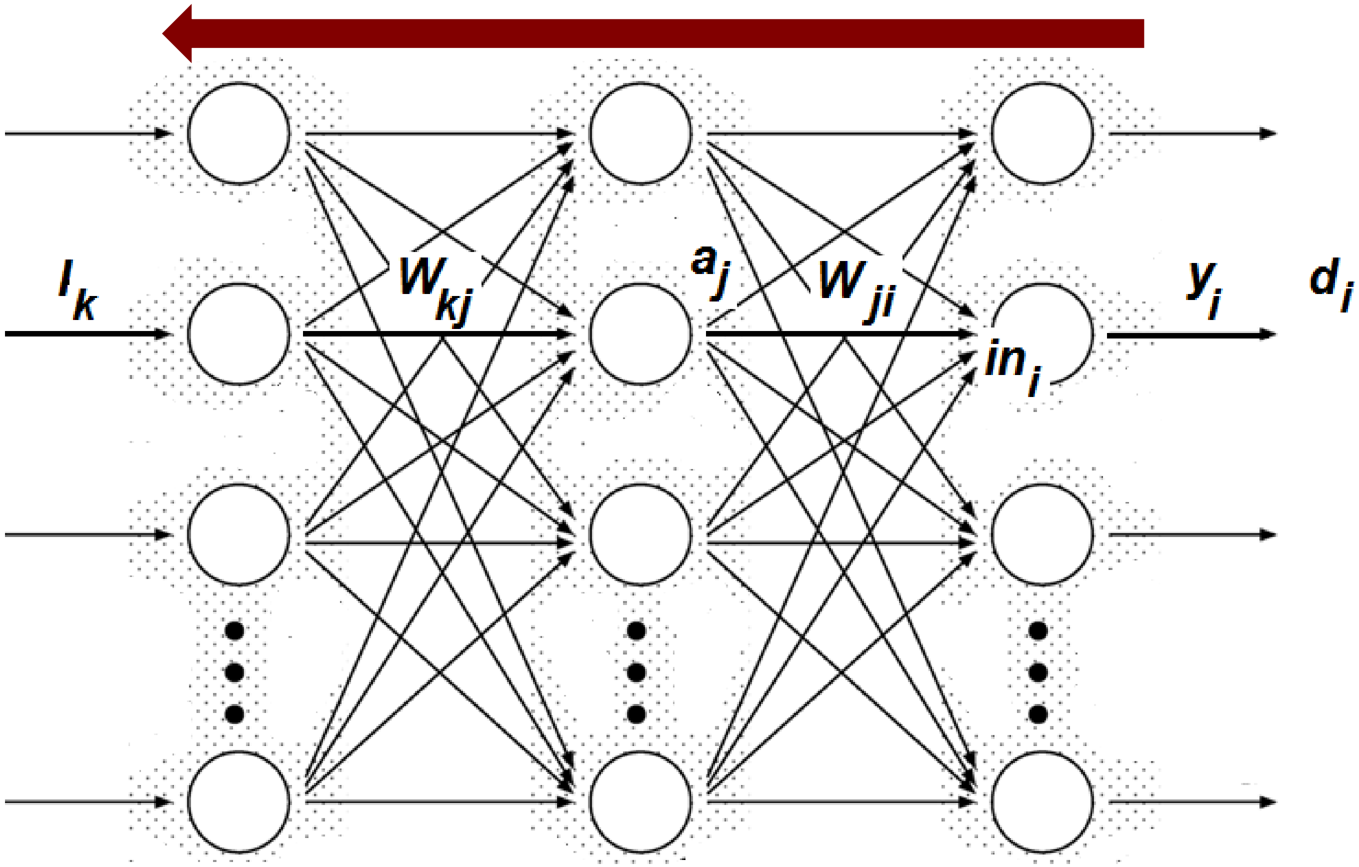
A tanítás lépései

- Kimenet számítása (forward pass)
- Hiba visszaterjesztése (backpropagation)



$$a_j(\underline{x}) = g(w_{k0} + \sum_{k=1}^{Input} x_k \cdot w_{kj}) \qquad y_i(\underline{x}) = g^*(w_{j0} + \sum_{j=1}^{Hidden} a_j(\underline{x}) \cdot w_{ji})$$

Backpropagation



$$W_{k,j} \leftarrow W_{k,j} - \alpha \frac{\partial E}{\partial W_{k,j}} \qquad W_{j,i} \leftarrow W_{j,i} - \alpha \frac{\partial E}{\partial W_{j,i}} \qquad E = \frac{1}{2} \sum_i (d_i - y_i)^2$$

Mesterséges Neurális Háló – hibavisszaterjesztés, alapok

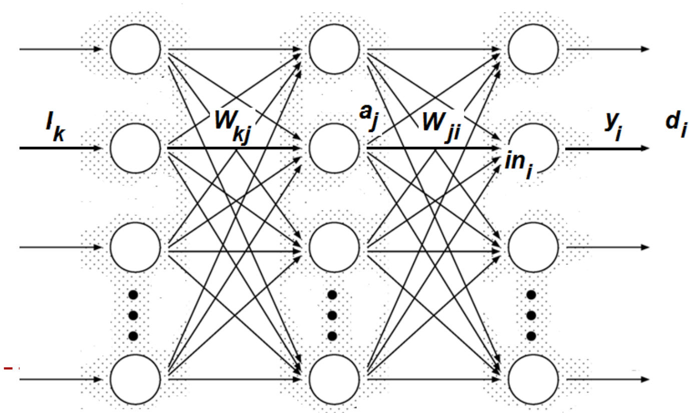
$$E = \frac{1}{2}\sum_i (d_i - y_i)^2 = \frac{1}{2}\sum_i Err_i \qquad W_{j,i} \leftarrow W_{j,i} - \alpha \frac{\partial E}{\partial W_{j,i}}$$

$$E(\mathbf{W}) = \frac{1}{2}\sum_i (d_i - g(\sum_j W_{j,i}\, a_j))^2 = \frac{1}{2}\sum_i (d_i - g(\sum_j W_{j,i}\, g(\sum_k W_{k,j}\, I_k)))^2$$

$$\frac{\partial E}{\partial W_{j,i}} = -a_j (d_i - y_i) g'(\sum_j W_{j,i}\, a_j) = -a_j (d_i - y_i) g'(in_i) = -a_j\, \Delta_i$$

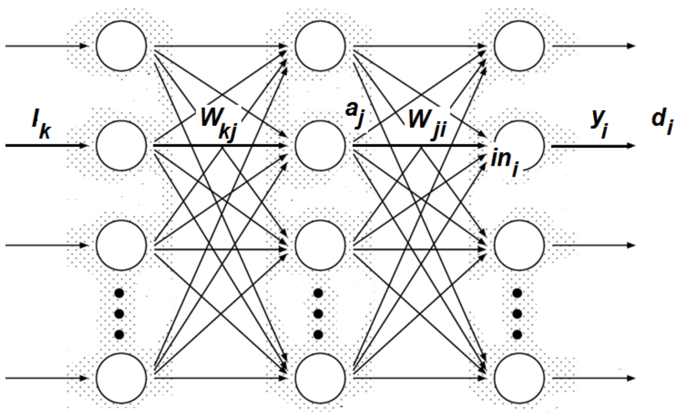
$$W_{j,i} \leftarrow W_{j,i} - \alpha \frac{\partial E}{\partial W_{j,i}} = W_{j,i} + \alpha\, a_j\, Err_i\, g'(in_i)$$

$$\Delta_i = Err_i\, g'(in_i)$$



$$W_{j,i} \leftarrow W_{j,i} + \alpha\, a_j\, \Delta_i$$

Mesterséges Neurális Háló – hibavisszaterjesztés, alapok



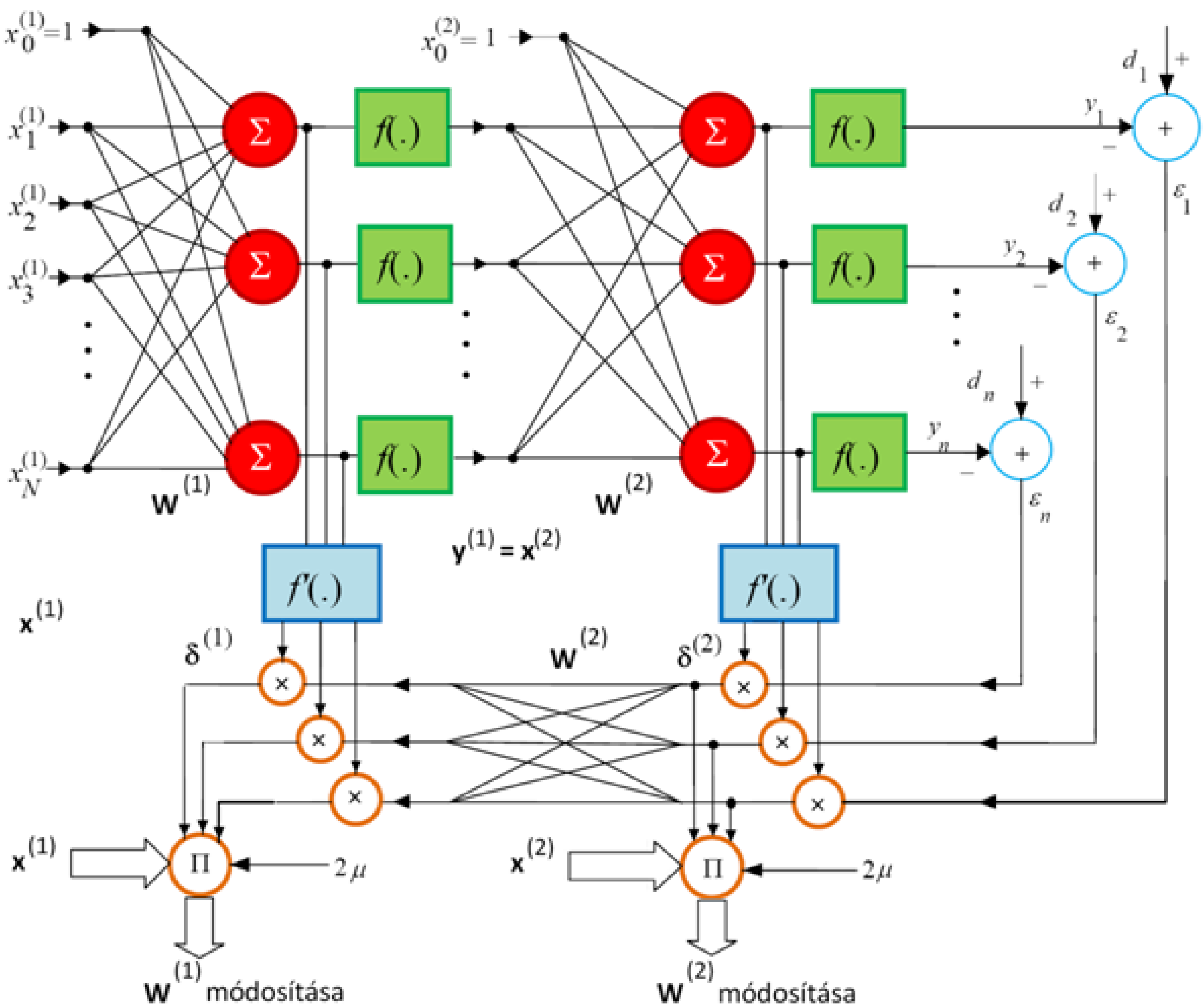
$$\frac{\partial E}{\partial W_{j,i}} = -a_j (d_i - y_i) g'(\sum_j W_{j,i}\, a_j) = -a_j (d_i - y_i) g'(in_i) = -a_j\, \Delta_i$$

$$\frac{\partial E}{\partial W_{k,j}} = -I_k\, \Delta_j$$

$$W_{k,j} \leftarrow W_{k,j} - \alpha \frac{\partial E}{\partial W_{k,j}}$$

$$W_{k,j} \leftarrow W_{k,j} + \alpha\, I_k\, \Delta_j \qquad \Delta_j = g'(in_j) \sum_i W_{j,i}\, \Delta_i$$

Mesterséges Neurális Háló



http://mialmanach.mit.bme.hu/neuralis/index

Neurális háló - tulajdonságok

Aktivációs függvény

- A hibavisszaterjesztést alkalmazó (sekély) hálókban általában a szigmoid függvényt vagy annak valamelyik változatát használjuk.
- A szigmoidnak megvan az a tulajdonsága, hogy deriváltja

$$g' = g(1-g) \qquad g(x) = \frac{1}{1+e^{-x}}$$

Neurális háló - tulajdonságok

Kifejezőképesség

- Nem rendelkeznek az általános logikai reprezentációk kifejezőképességével.
- A többrétegű hálók osztálya egészében, mint osztály az attribútumok bármely függvényének reprezentációjára képes.

Számítási hatékonyság

- Legrosszabb esetben a szükséges epochok száma a bemenetek számának, *n*-nek, még exponenciális függvénye is lehet.
- A hibafelület lokális minimumai problémát jelenthetnek.

Neurális háló - tulajdonságok

Általánosító képesség: jó általánosításra képesek.

Zajérzékenység: alapvetően nemlineáris regresszió, nagyon jól tolerálják a bemeneti adatok zajosságát.

Átláthatóság: alapvetően fekete doboz.

A priori ismeret felhasználása: nem könnyű (még aktívan kutatott terület)

A mesterséges neurális háló kérdései

- Mekkora (hány réteg, rétegenként hány processzáló elem) hálózatot válasszunk?
- Hogyan válasszuk meg a tanulási tényező, α értékét?
- Milyen kezdeti súlyértékeket állítsunk be?
- Hogyan válasszuk meg a tanító és a tesztelő minta készletet?
- Hogyan használjuk fel a tanító pontokat, milyen gyakorisággal módosítsuk a hálózat súlyait?

- http://mialmanach.mit.bme.hu/neuralis/index

Kérdések 1.

- Mekkora hálózatot válasszunk?(hány réteg, rétegenként hány processzáló elem)

- Nincs egzakt számítás ennek meghatározásához.

1.) Kiindulás egy ‘nagyobb’ hálózatból, majd a hálózatban megmutatkozó redundancia csökkentése minimális értékre.

- pruning technikák (http://mialmanach.mit.bme.hu/neuralis/ch04s03)
- regularizáció

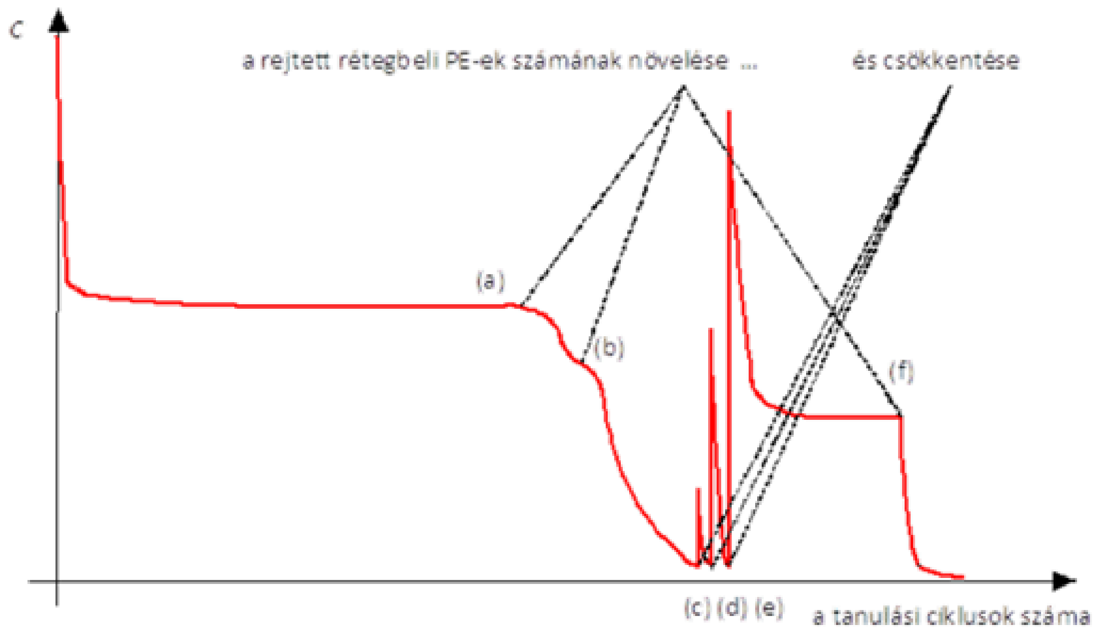
$$C_T(\mathbf{w}) = C(\mathbf{w}) + \lambda \sum_{i,j} |w_{ij}|$$

2.) Kiindulás egy ‘kisebb’ méretű hálózatból, majd fokozatos bővítéssel (processzáló elemekkel, ill. rétegekkel) vizsgáljuk meg tudja-e oldani a feladatot.

A két megközelítés nem feltétlenül vezet azonos eredményre!

Kérdések 1.

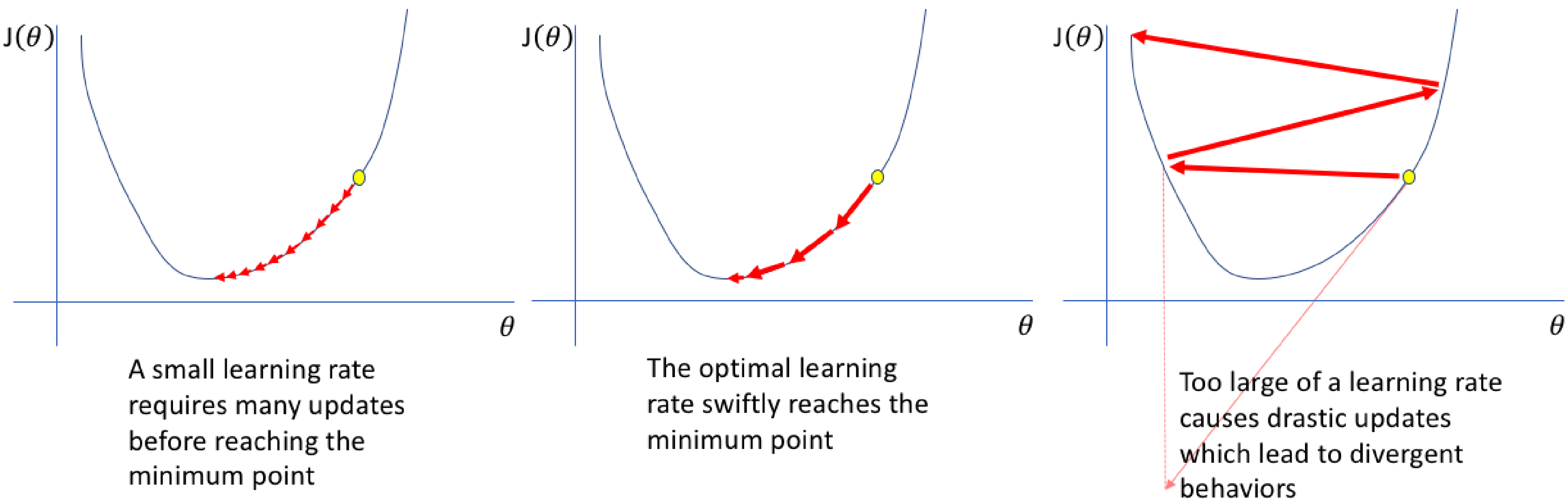
2.) Kiindulás egy ‘kisebb’ méretű hálózatból, majd fokozatos bővítéssel (processzáló elemekkel, ill. rétegekkel) vizsgáljuk meg tudja-e oldani a feladatot.



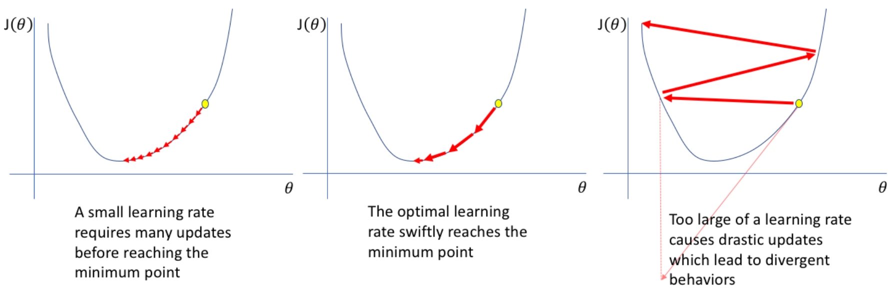
http://mialmanach.mit.bme.hu/neuralis/ch04s03

Kérdések 2.

- Hogyan válasszuk meg a tanulási tényező értékét?



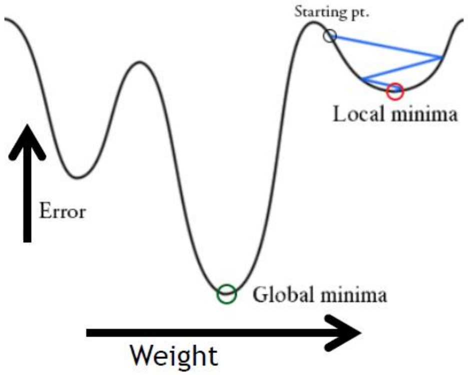
Kérdések 2.



- nincs egyértelmű módszer *α* (=μ) meghatározására
- változó, lépésfüggő *α* alkalmazása
  - kezdeti nagy értékből kiindulva m lépésenként csökkentjük *α* -t.
- adaptív *α* alkalmazása
  - *α* változtatását attól tesszük függővé, hogy a súlymódosítások csökkentik-e a kritériumfüggvény értékét

Kérdések 3.

- Milyen kezdeti súlyértékeket állítsunk be?



- Nincs egyértelmű módszer
- A priori ismeret hiányában a véletlenszerű súlybeállítás a leginkább megfelelő
  - az egyes súlyokat egy egyenletes eloszlású valószínűségi változó különböző értékeire választhatjuk
- Szigmoid aktiváció függvénynél kerüljük el, hogy már a tanítás elején telítéses szakaszra kerüljön a rejtett rétegek kimenete
  - Minél nagyobb a hálózat, annál kisebb véletlen értékek választása célszerű

Kérdések 4.

- Hogyan válasszuk meg a tanító és a tesztelő minta készletet?

Általánosan:

- Tanító minta : felhasználjuk a tanításhoz
- Tesztelő minta: nem használjuk a tanításhoz csak a minősítéshez

Kérdések 4.

- Hogyan válasszuk meg a tanító, kiértékelő és tesztelő minta készletet?

Neurális hálóknál:

- Tanító minta : felhasználjuk a tanításhoz
- Kiértékelő (validációs) minta: nem használjuk súlymódosításhoz, de ahhoz igen, hogy mikor álljon le a tanítás
- Tesztelő (minősítő) minta: nem használjuk a tanításhoz csak a minősítéshez

- http://mialmanach.mit.bme.hu/neuralis/ch04s03

Kérdések 4.

- Hogyan válasszuk meg a tanító, kiértékelő és tesztelő minta készletet?

- Nagy számú mintánál nem jelent problémát akár három egyenlő részre osztani az adathalmazt

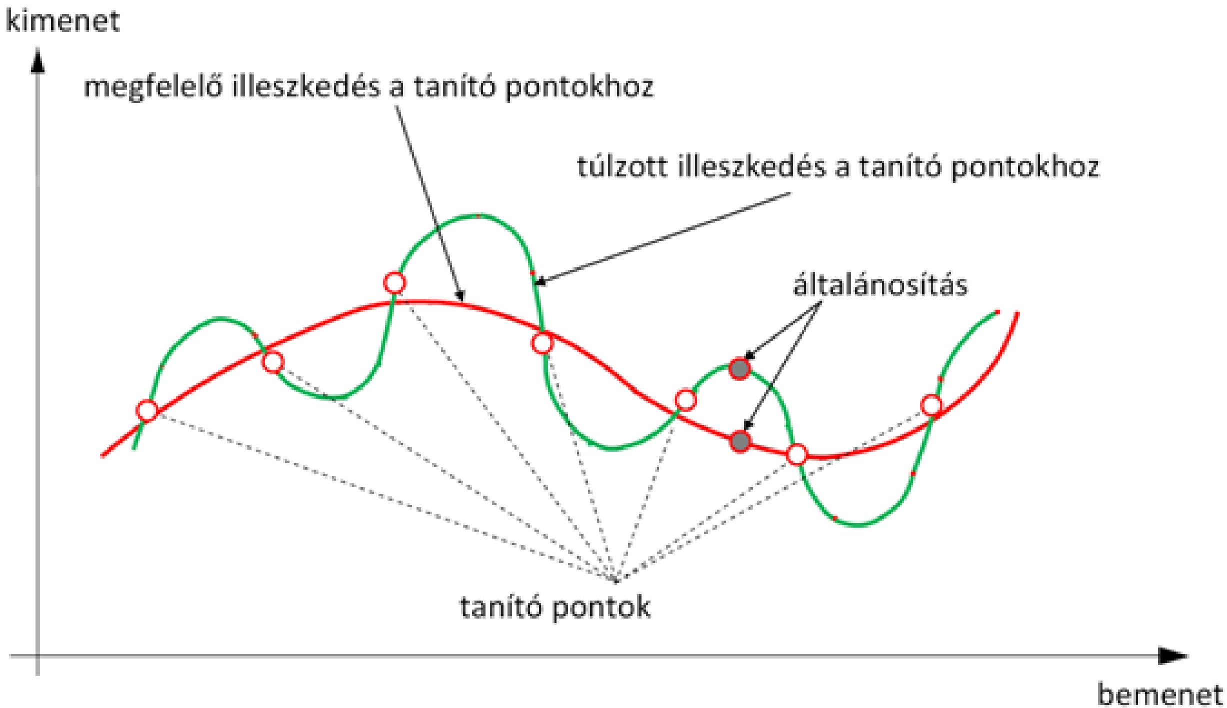
- A legtöbb valós probléma esetén azonban „sosem elég” a minta
  - Léteznek elméleti és gyakorlati korlátok a szükséges tanítópontok számára a háló mérete és teszt készleten elvárt hiba alapján
  - Kereszt-kiértékelési (cross validation) eljárások alkalmazása

http://mialmanach.mit.bme.hu/neuralis/ch04s03

Kérdések 5.

Hogyan védekezzünk a túltanulással szemben?

- Akkor lép fel (overfitting), ha a tanító készlet mintáira már nagyon kis hibájú válaszokat kapunk, miközben a kiértékelő készletre egyre nagyobb hibával válaszol a hálózat.



Kérdések 5.

Hogyan védekezzünk a túltanulással szemben?

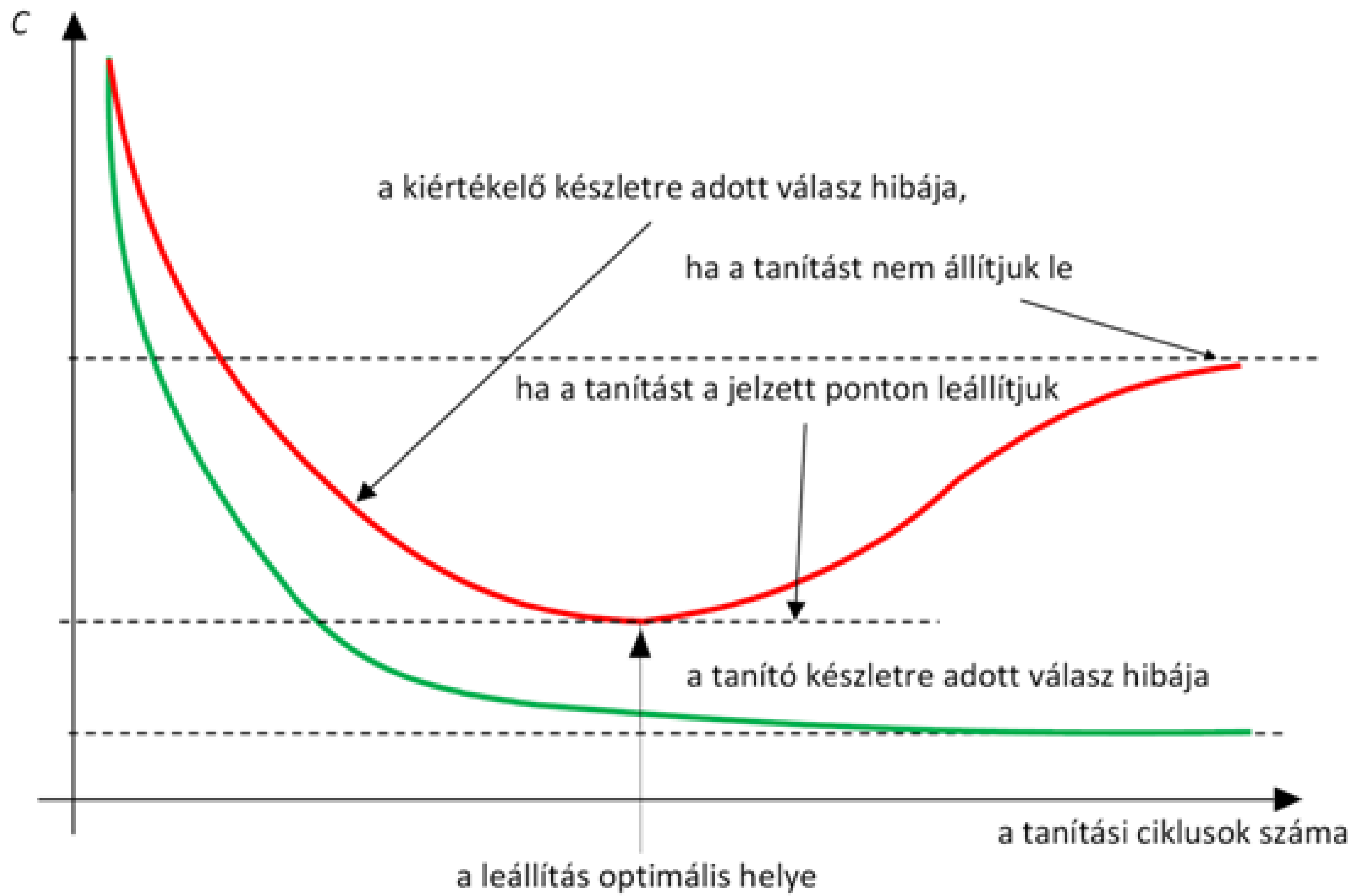
- A hálózat válaszai túlzottan illeszkednek a véges számú tanító pont által megadott leképezéshez
- Ok: a hálózat mérete (szabadságfoka) a tanító pontok számához viszonyítva túl nagy

Miért kell védekezni ellene?

- Túltanulással romlik az általánosító képesség.

Kérdések 5.

- Meddig tanítsuk a hálózatot?



Kérdések 6.

- Hogyan használjuk fel a tanító pontokat, milyen gyakorisággal módosítsuk a hálózat súlyait?

- Pontonként
  - Súlymódosítás tanítópontonként

- Kötegelt (batch) eljárás szerint
  - Súlymódosítás k db tanítópontonként vagy
  - Csak a teljes tanító készlet (epoch) felhasználása után