Informált keresési stratégiák absztrakció alapú modellellenőrzésben

Vörös Asztrik

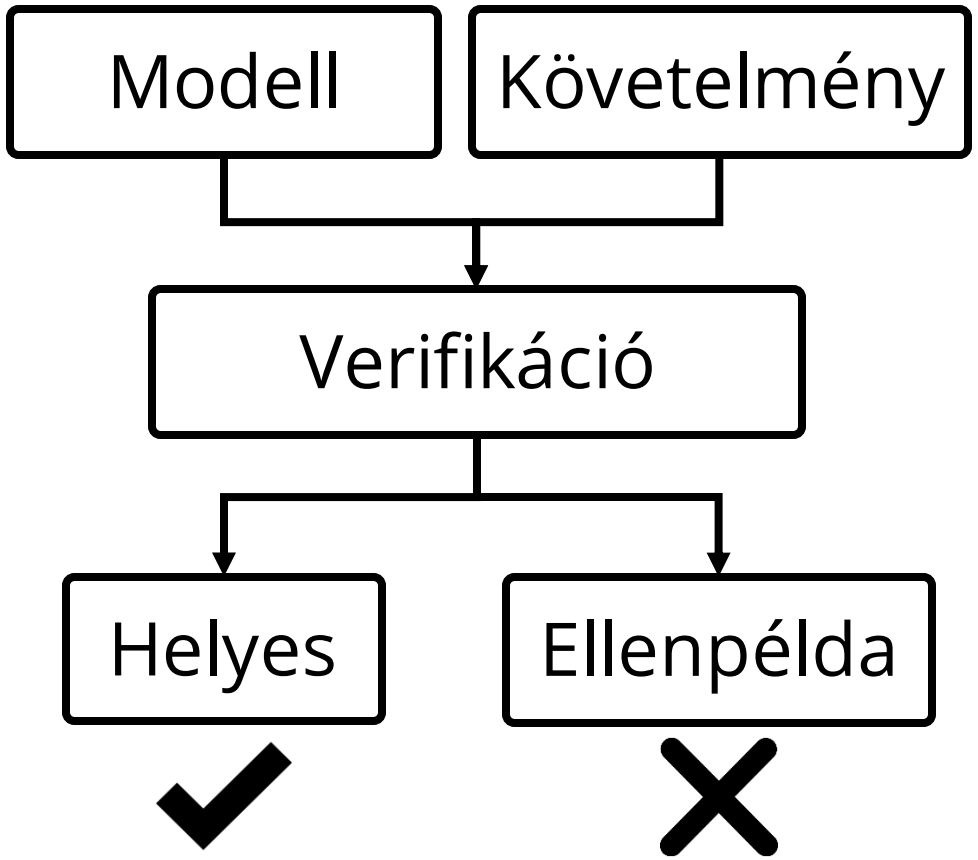
Konzulensek
Szekeres Dániel
dr. Vörös András
dr. Molnár Vince

Budapest University of Technology and Economics
Department of Measurement and Information Systems
Critical Systems Research Group

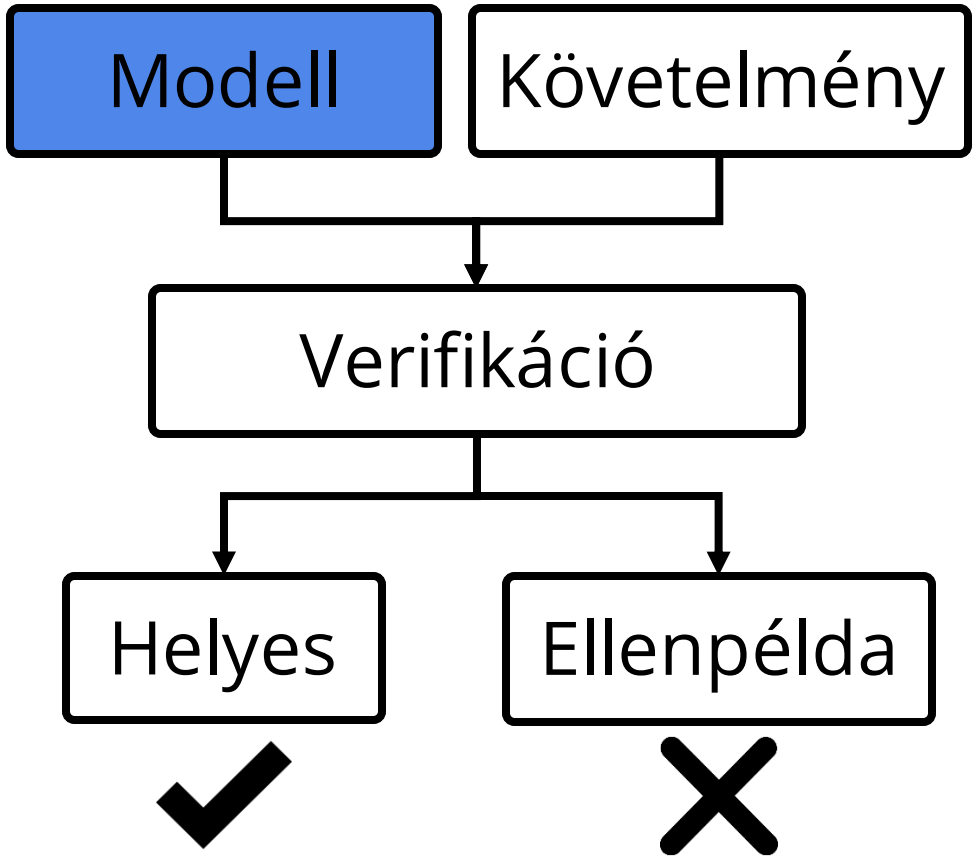
Kritikus beágyazott rendszerek

- Helyesség biztosítása fontos
- Nem elég egyes viselkedések vizsgálata
- Formális verifikáció: minden lehetséges viselkedés ellenőrzése

Formális verifikáció



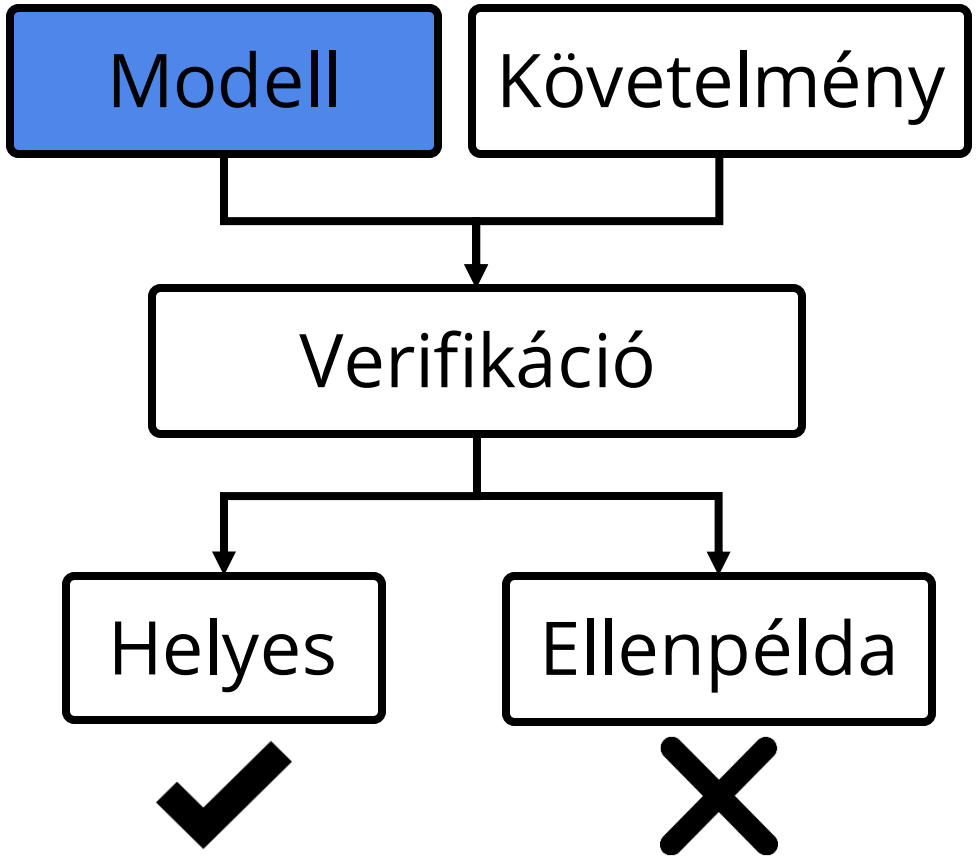
Formális verifikáció



C nyelvű forráskód

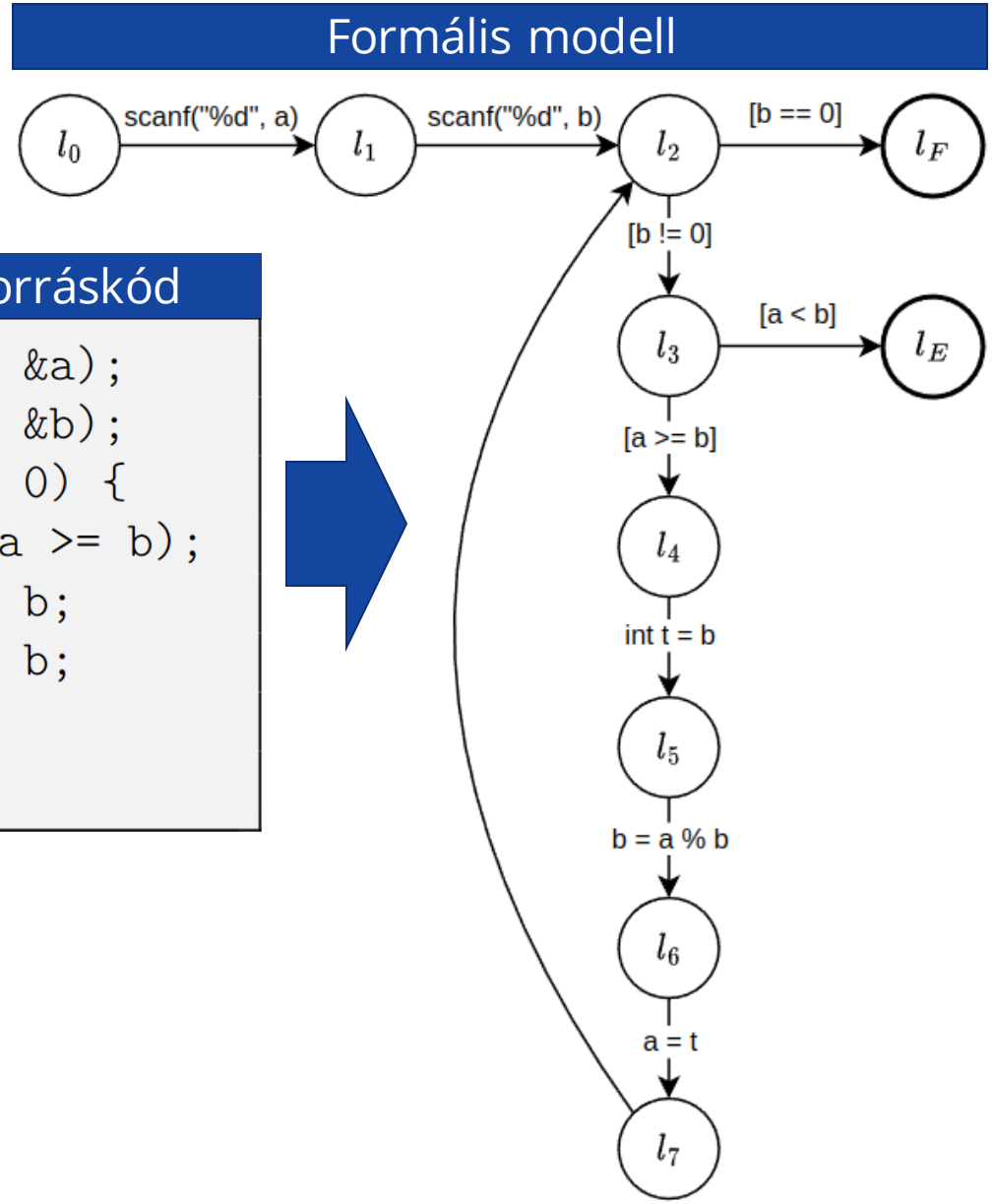
```
scanf("%d", &a);
scanf("%d", &b);
while (b != 0) {
    assert(a >= b);
    int t = b;
    b = a % b;
    a = t;
}
```

Formális verifikáció

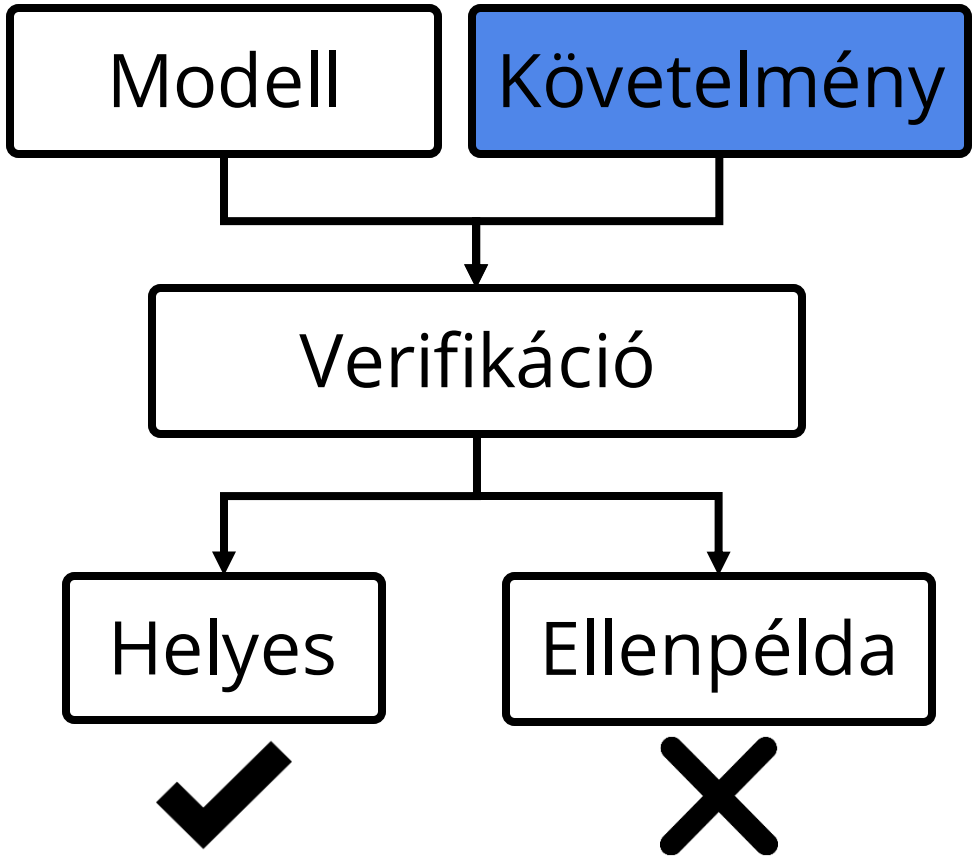


C nyelvű forráskód

```
scanf("%d", &a);
scanf("%d", &b);
while (b != 0) {
    assert(a >= b);
    int t = b;
    b = a % b;
    a = t;
}
```



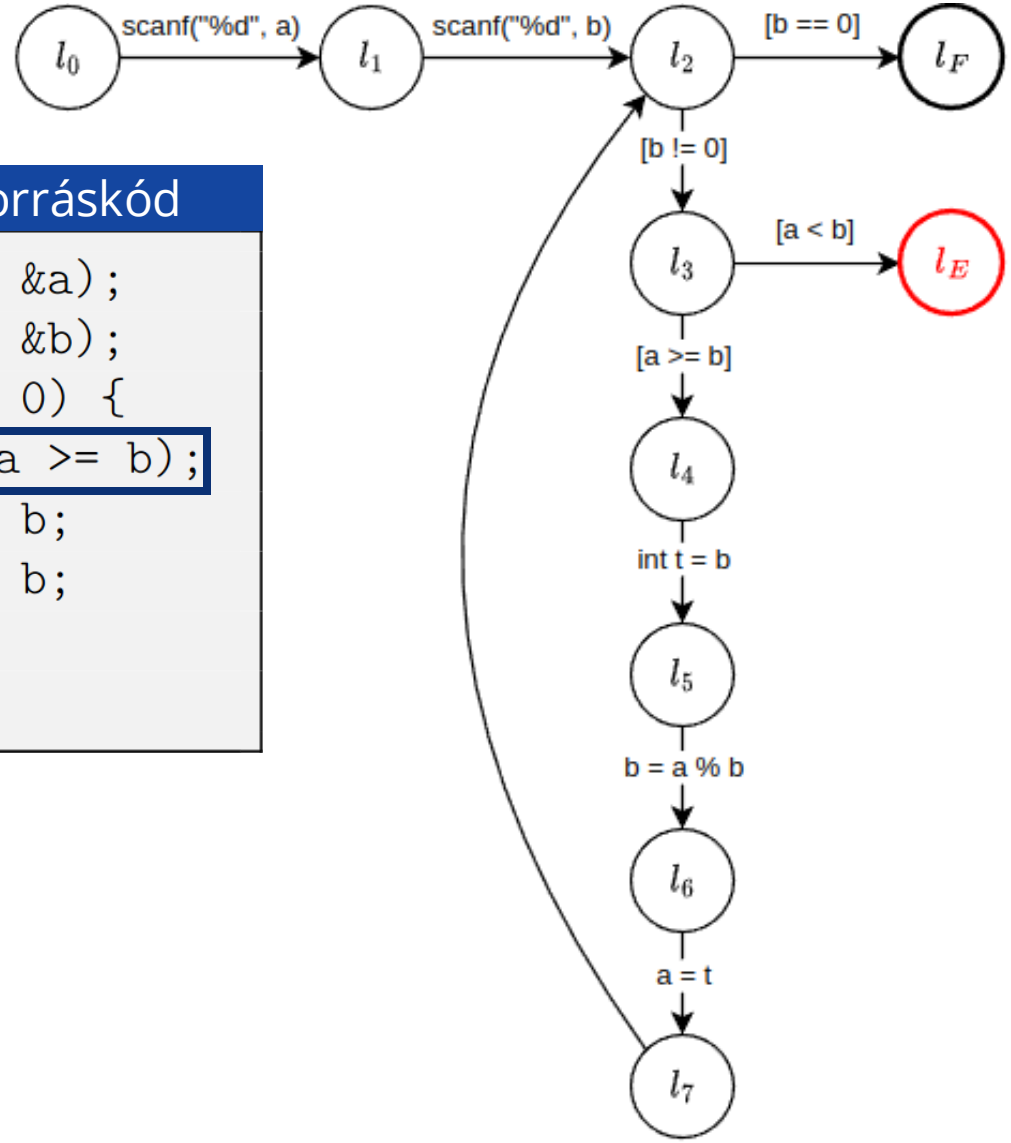
Formális verifikáció



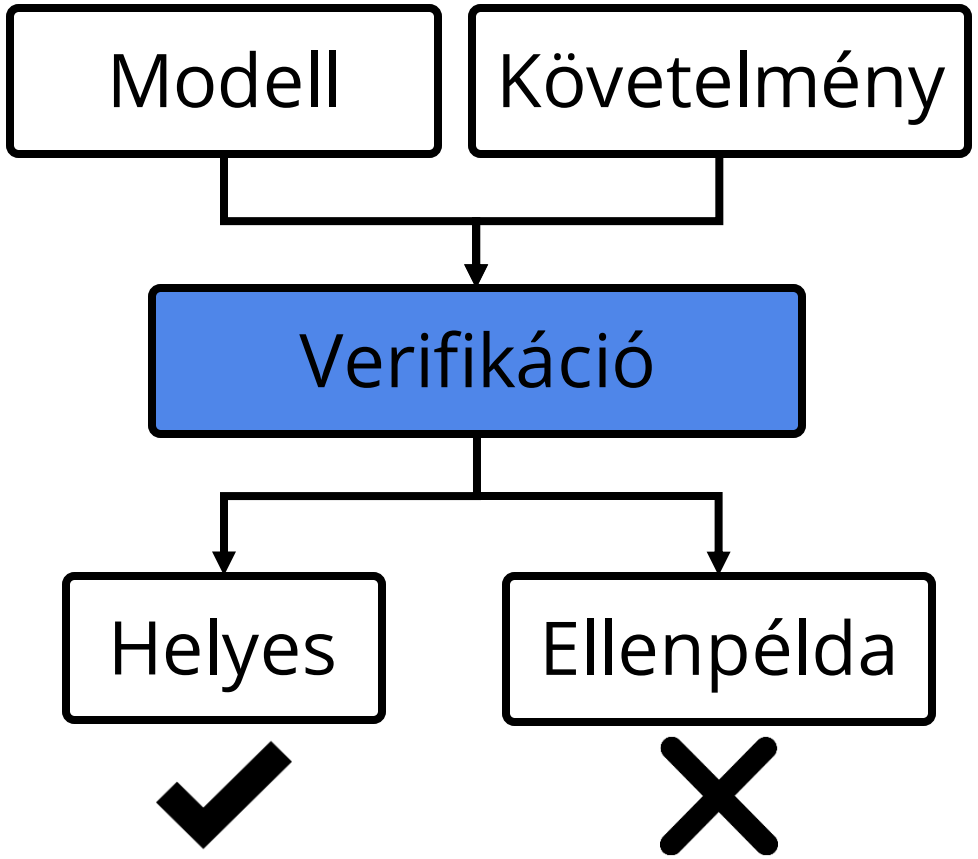
C nyelvű forráskód

```
scanf("%d", &a);
scanf("%d", &b);
while (b != 0) {
    assert(a >= b);
    int t = b;
    b = a % b;
    a = t;
}
```

Formális modell



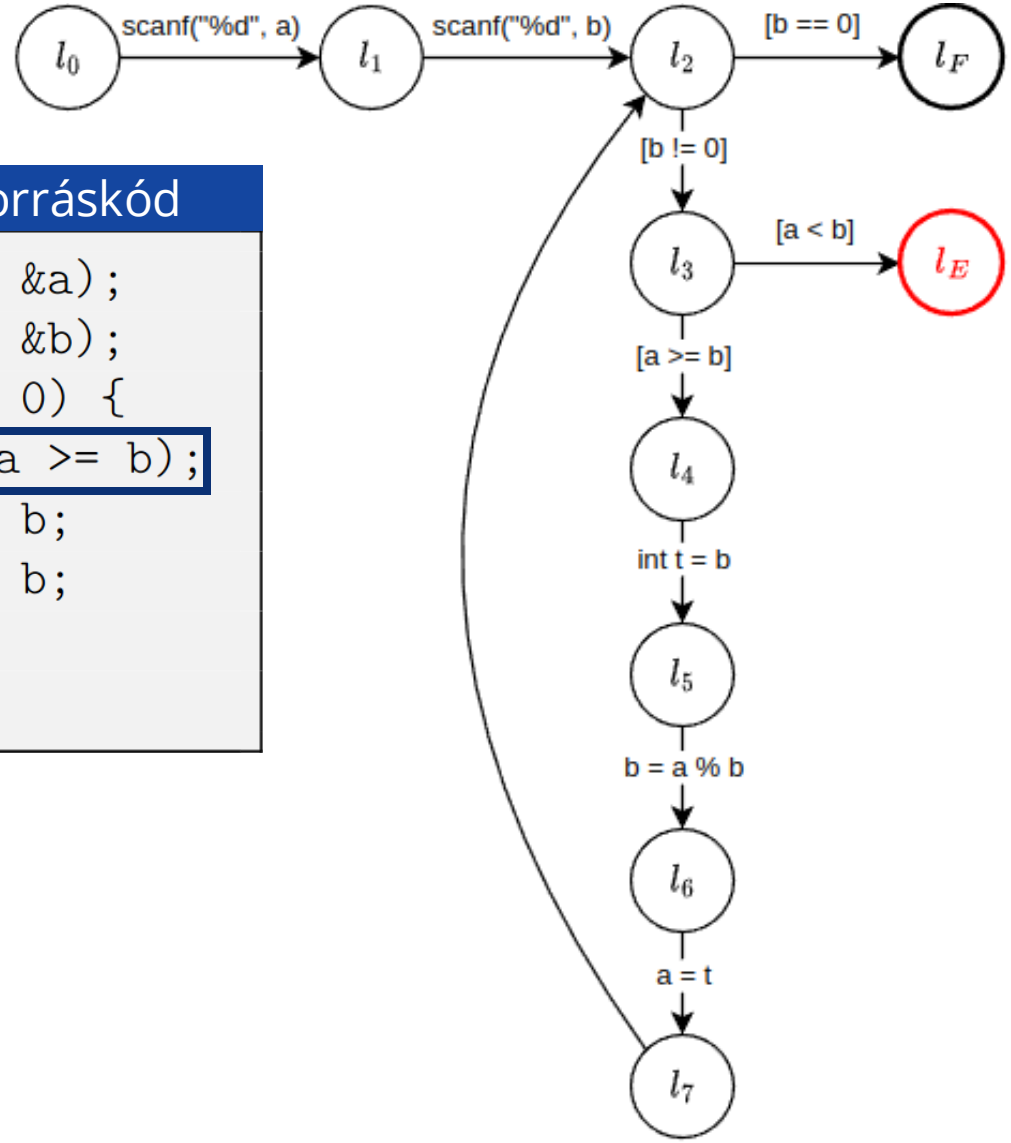
Formális verifikáció



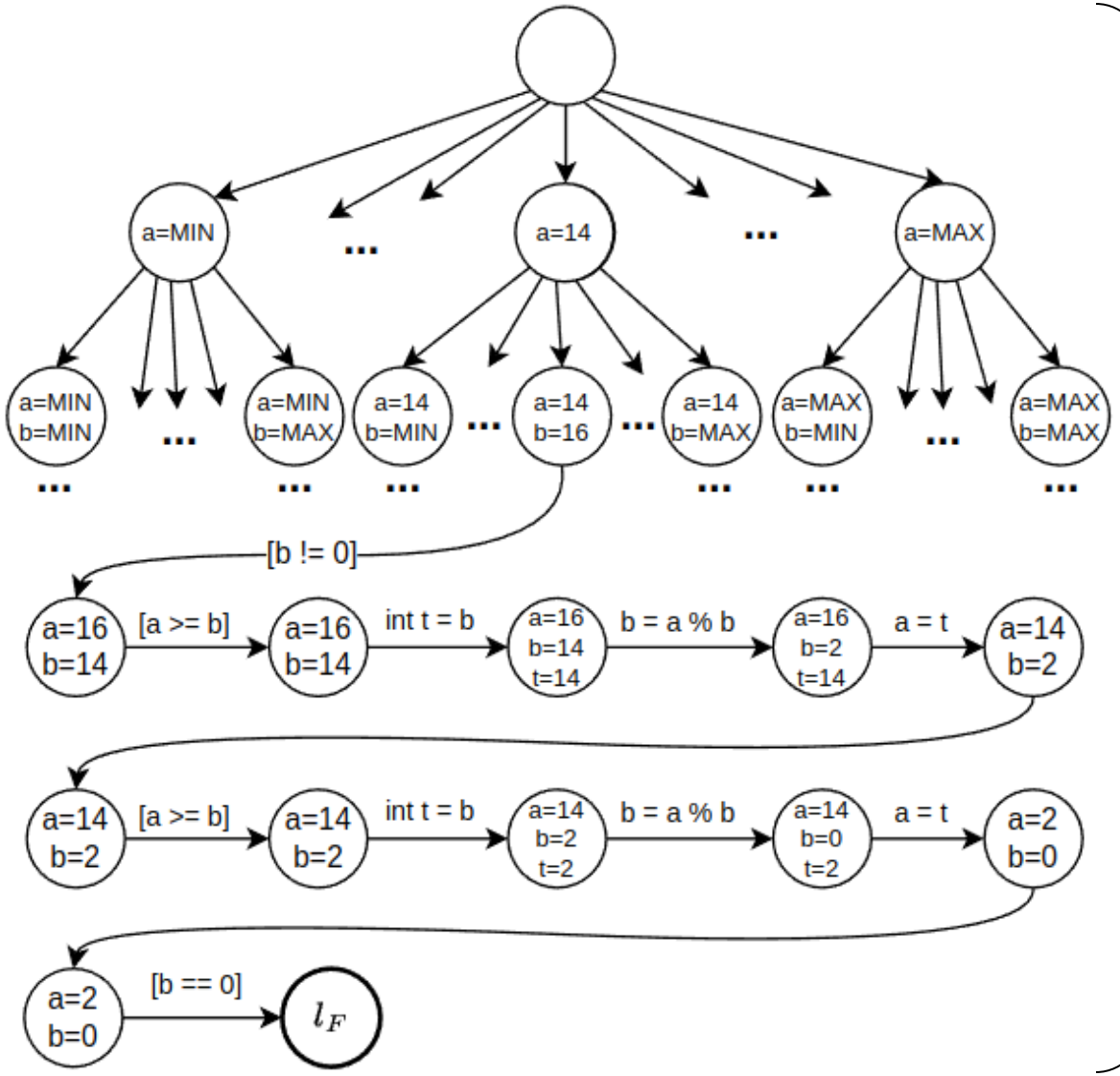
C nyelvű forráskód

```
scanf("%d", &a);
scanf("%d", &b);
while (b != 0) {
    assert(a >= b);
    int t = b;
    b = a % b;
    a = t;
}
```

Formális modell

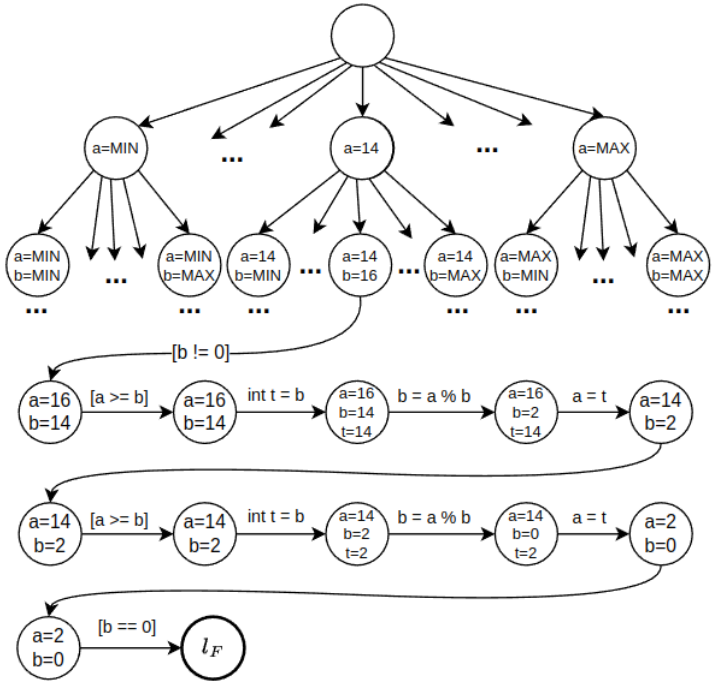


Verifikáció: Állapottér bejárás



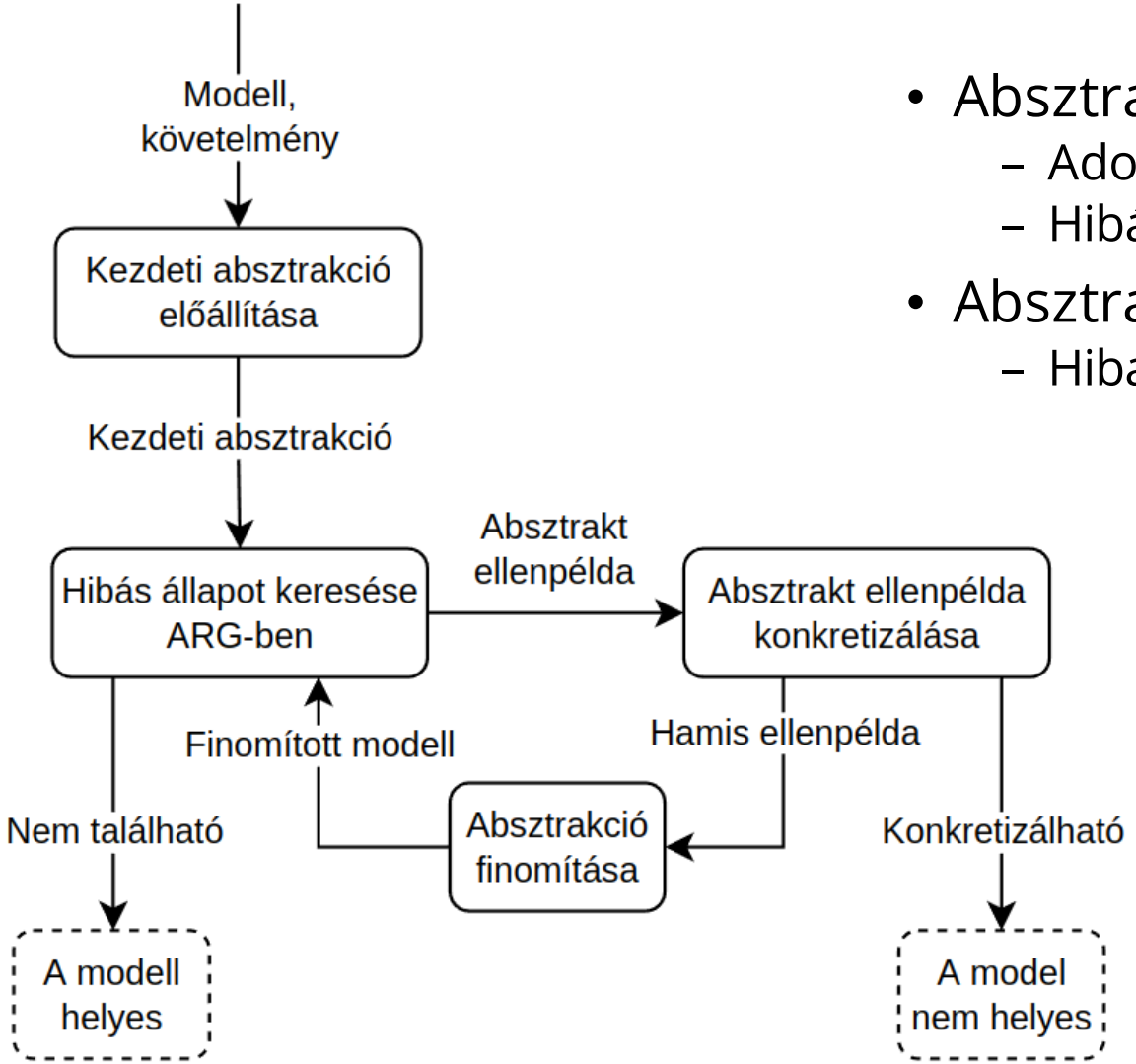
Hibás állapotok keresése

Verifikáció: Állapottér bejárás

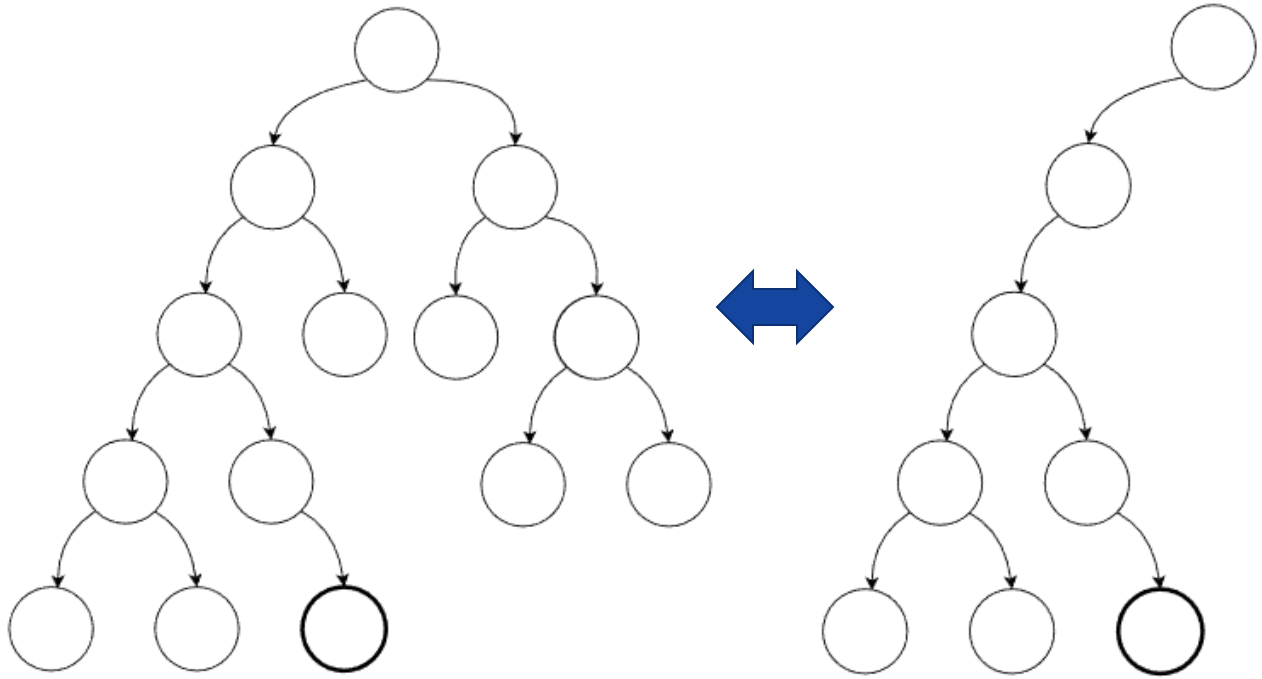


- Állapottér-robbanás
  - Szoftverek adat változói sokféle kombinációban előállhatnak
  - Állapottér mérete exponenciálisan nő a változók számában
- Megoldás: absztrakció alapú formális verifikáció

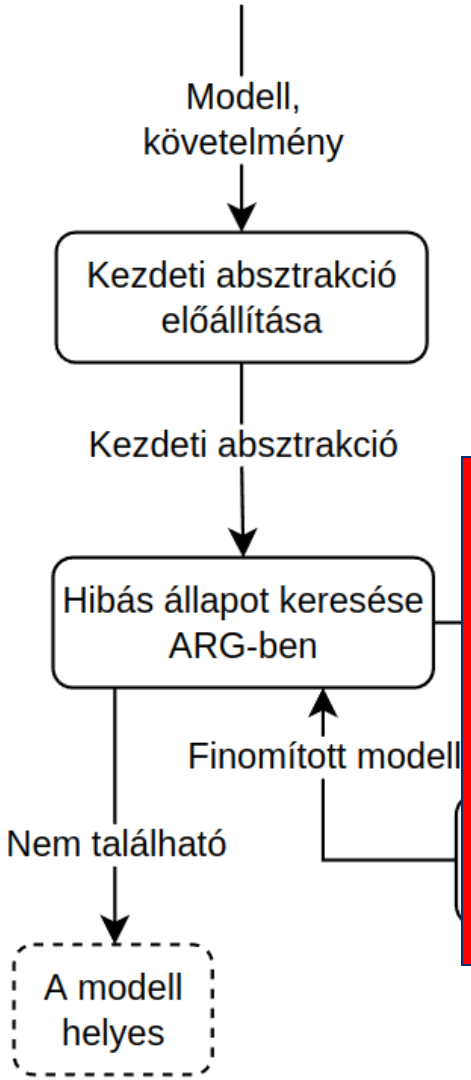
CEGAR – Ellenpélda-vezérelt absztrakció finomítás



- Absztrakt elérhetőségi gráf - ARG
  - Adott absztrakt állapottér tömör reprezentációja
  - Hibás állapotok keresése: BFS, DFS, heurisztikák
- Absztrakt állapottér a lehetséges lefutásokat felülbecsli
  - Hibaállapot nem érhető el → Helyes a rendszer



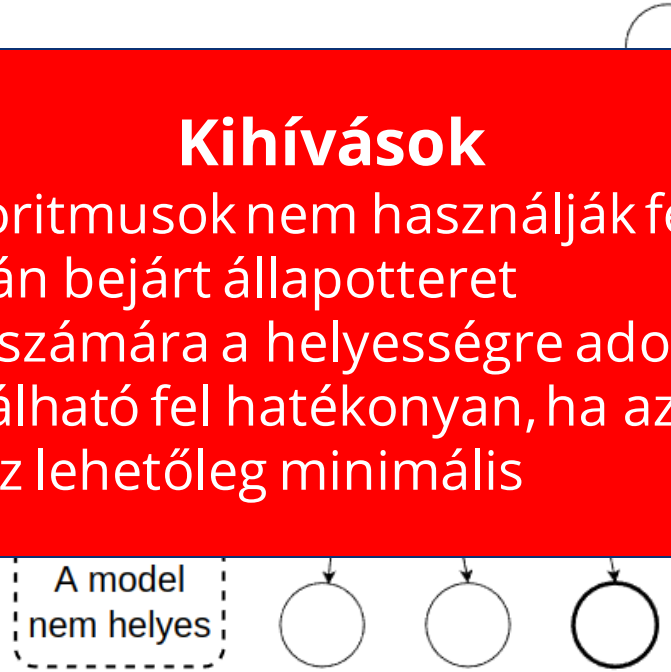
CEGAR – Ellenpélda-vezérelt absztrakció finomítás

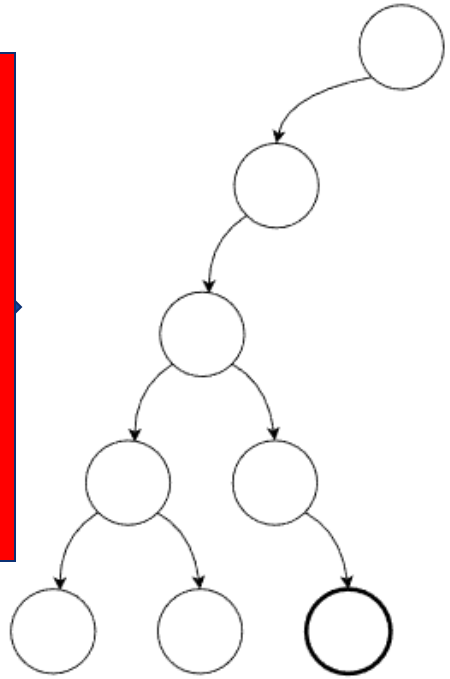


- Absztrakt elérhetőségi gráf - ARG
  - Adott absztrakt állapottér tömör reprezentációja
  - Hibás állapotok keresése: BFS, DFS, heurisztikák
- Absztrakt állapottér a lehetséges lefutásokat felülbecsli
  - Hibaállapot nem érhető el → Helyes a rendszer

Kihívások

1. Meglévő algoritmusok nem használják fel a korábbi iterációk során bejárt állapotteret
2. A mérnökök számára a helyességre adott ellenpélda akkor használható fel hatékonyan, ha az a hibára fókuszál, azaz lehetőleg minimális

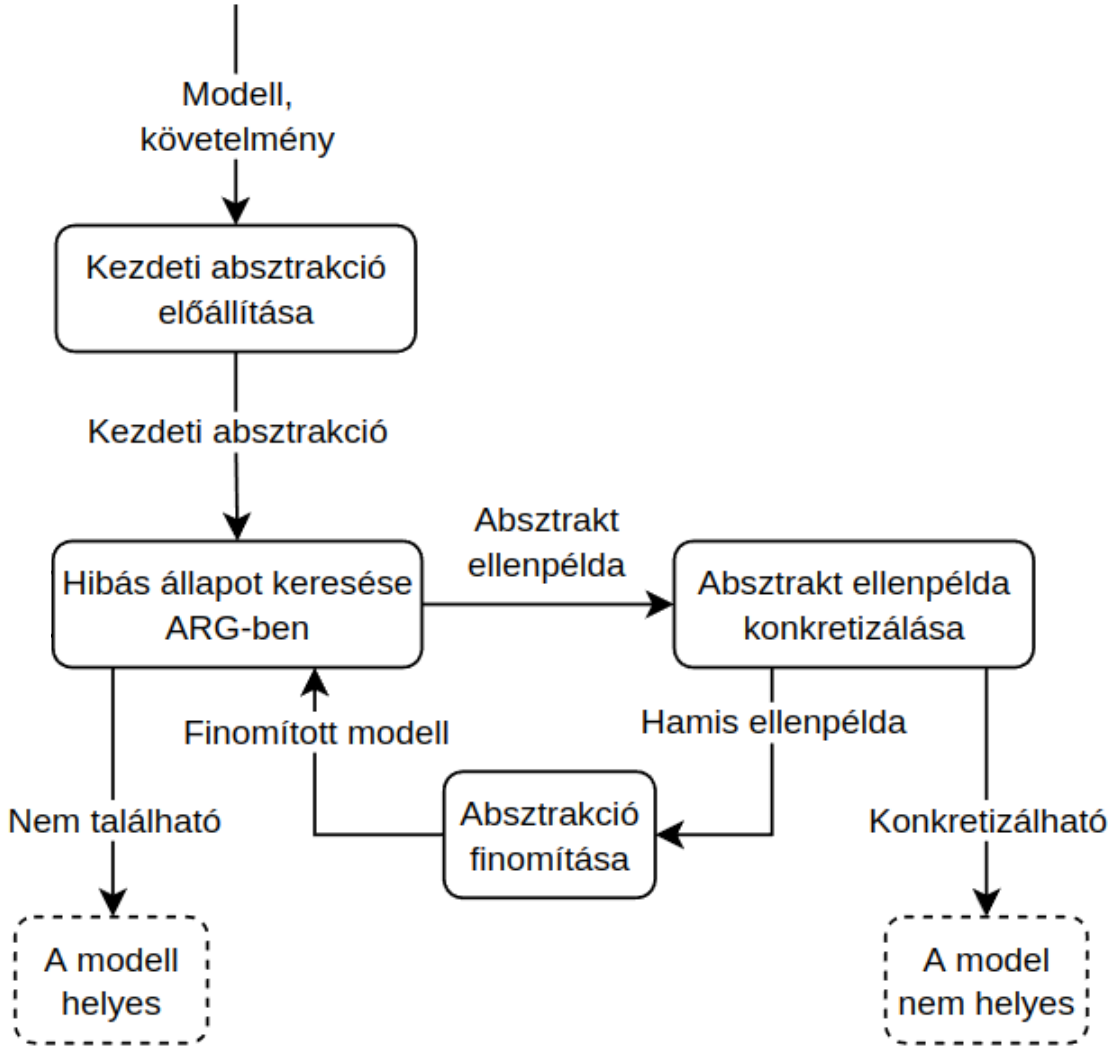




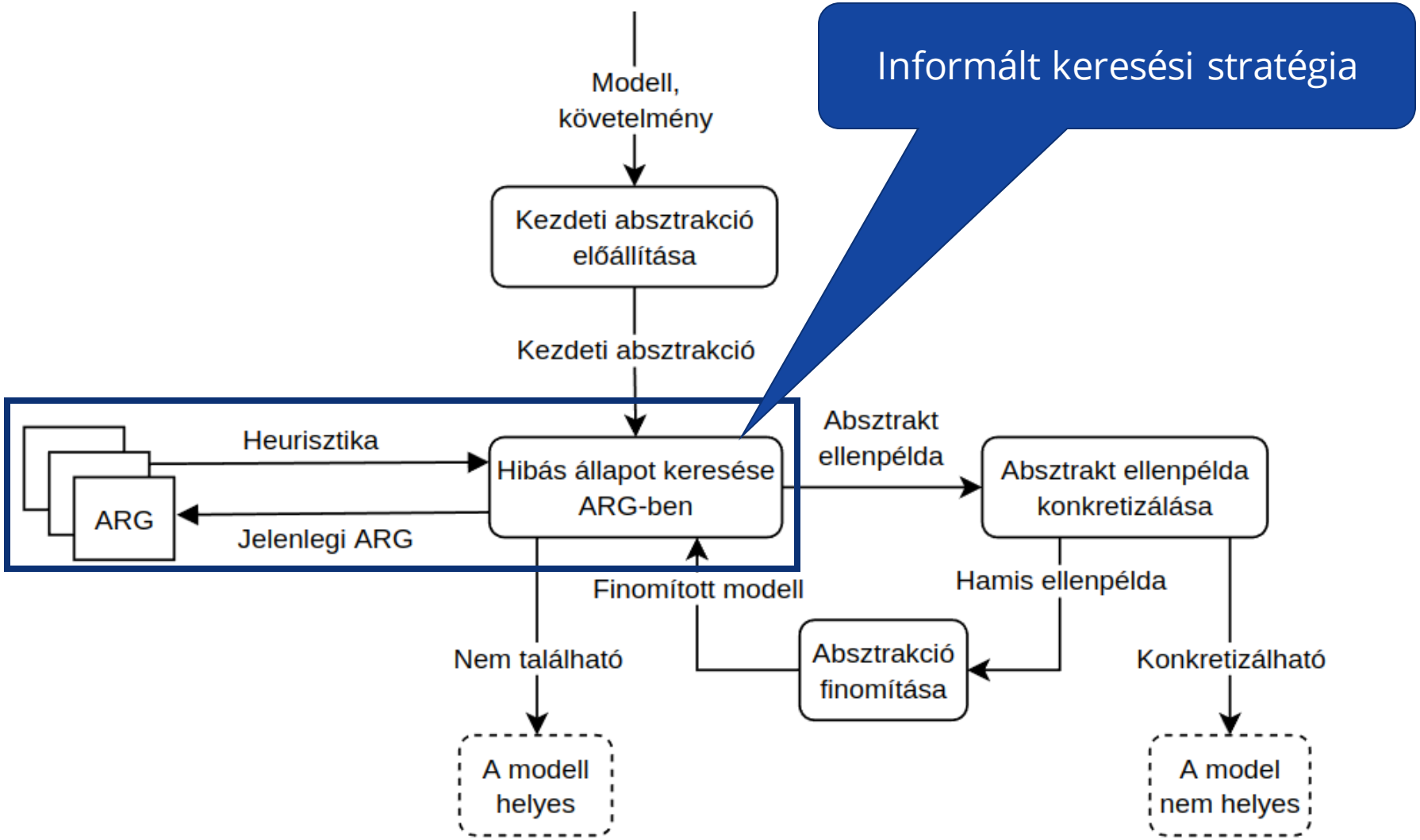
Célom: hatékony formális verifikáció

- Hatékony keresési algoritmusok implementálása
  - Gyorsabb helyesség bizonyítás
  - Hibás rendszer: rövidebb, minimális hosszúságú ellenpélda
- A* keresés az absztrakt állapottérben
  - CEGAR iterációk során nyert információk elmentése
  - Absztrakt elérhetetőségi gráfokból távolság információ származtatása
  - A* keresés az állapottér reprezentációkban az aktuális keresés támogatására

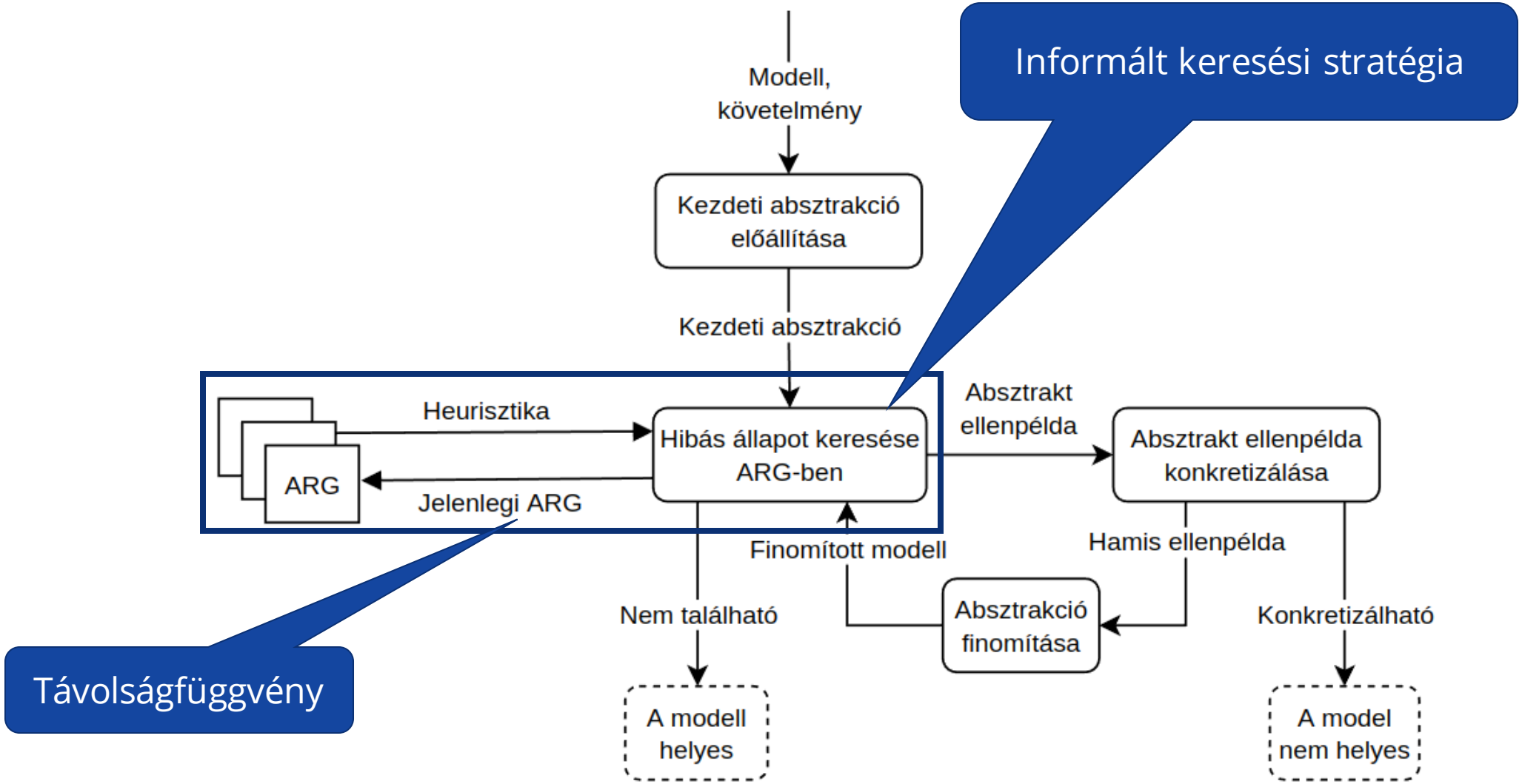
A* kereséssel támogatott CEGAR



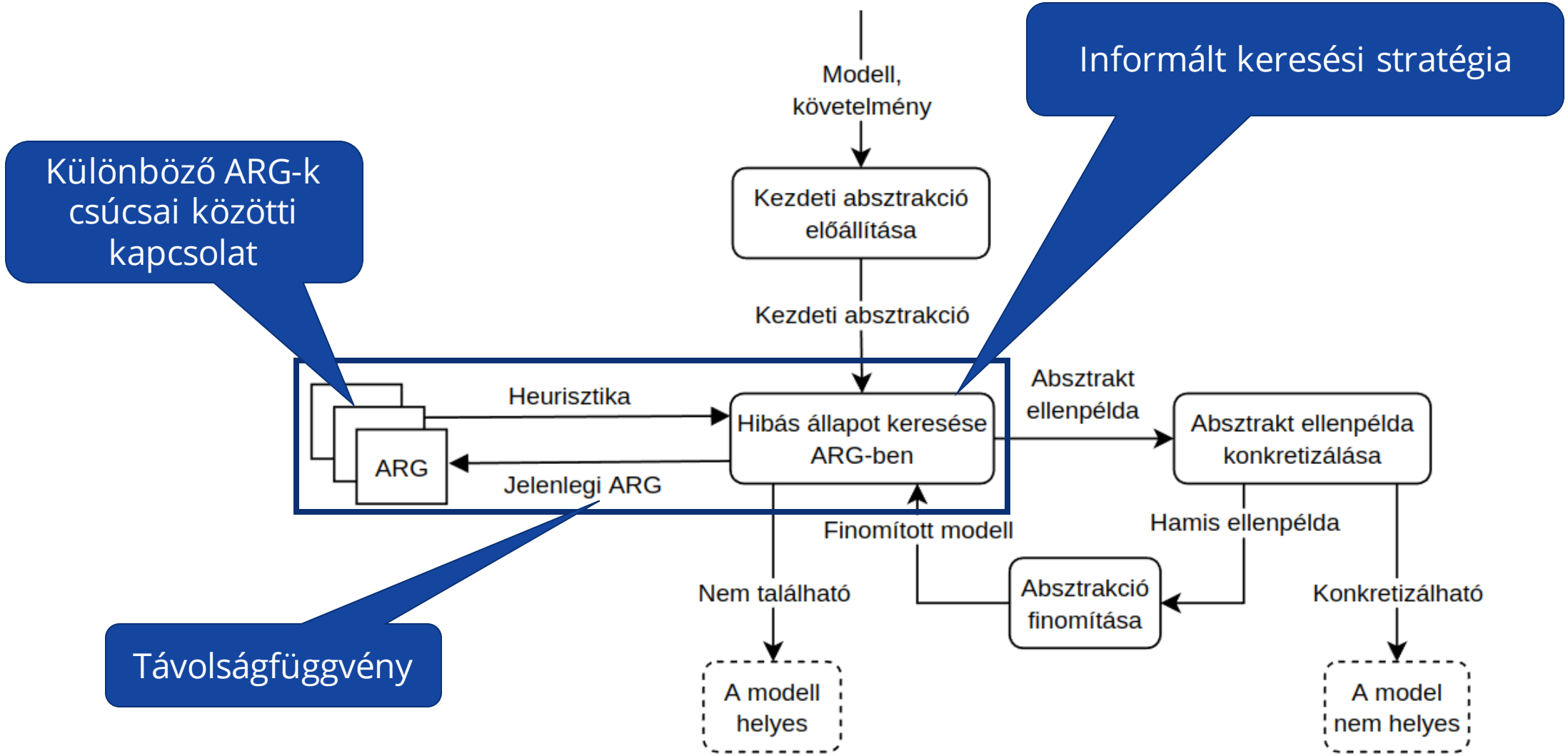
A* kereséssel támogatott CEGAR



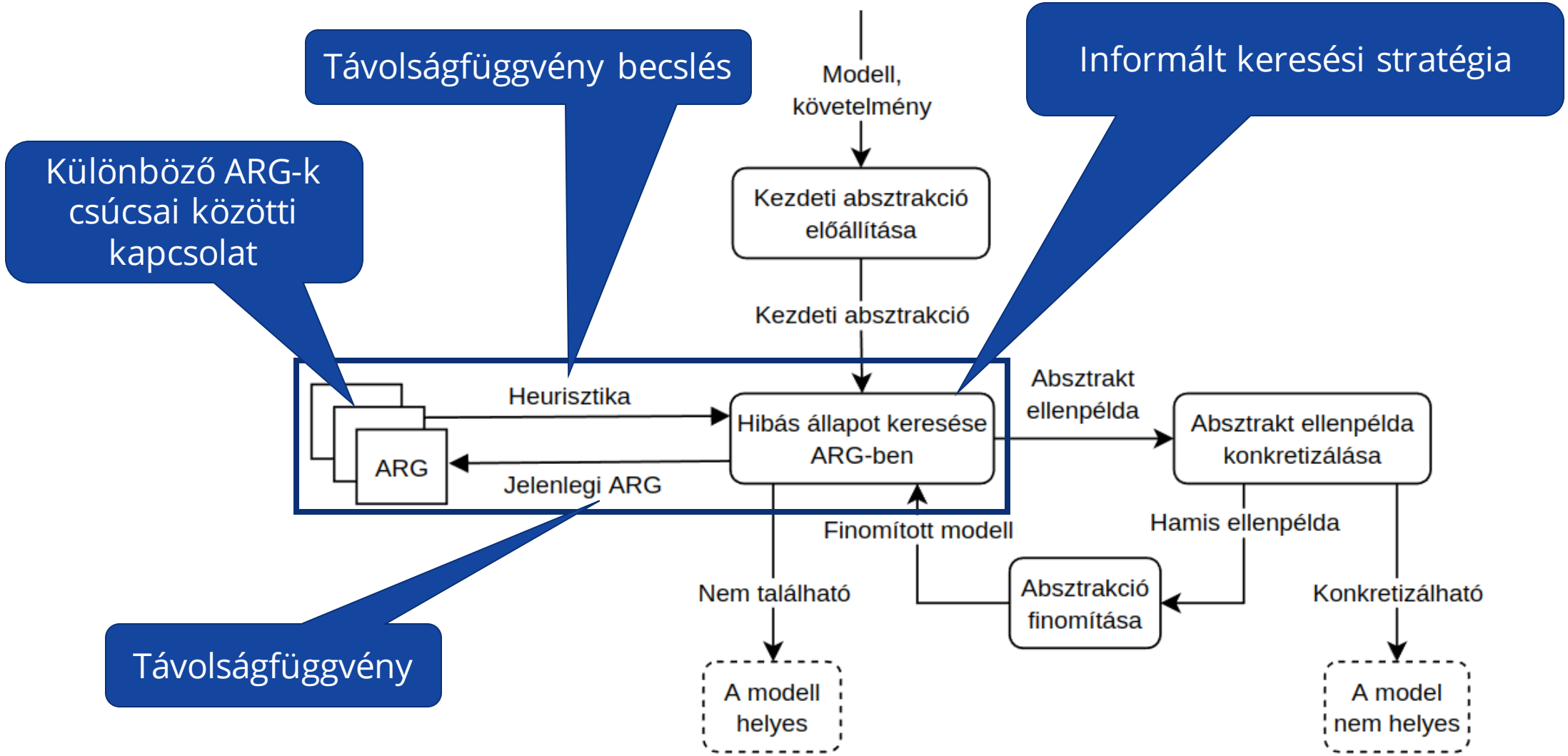
A* kereséssel támogatott CEGAR



A* kereséssel támogatott CEGAR



A* kereséssel támogatott CEGAR



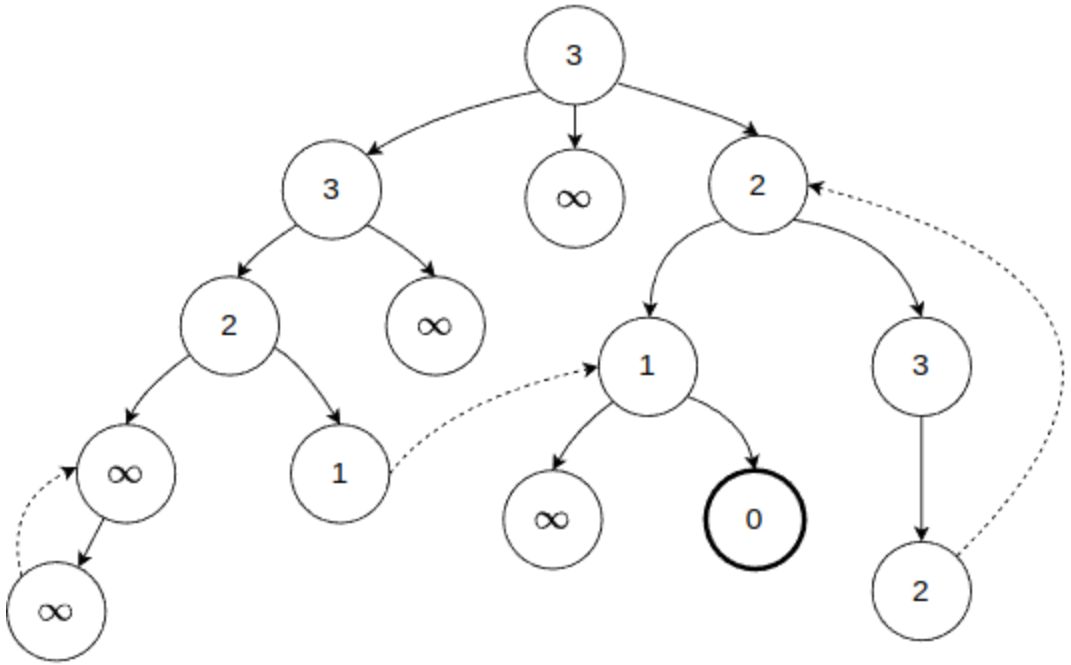
Keresési algoritmusok – A* keresés

- Irányított keresés: haladás a hibás állapot irányába
- Az A* működéséhez szükségünk van heurisztikára:
  - Konzervatív: szomszédos csúcsok hibás állapottól vett becsült távolságának a különbsége konzisztens
  - Minél pontosabban becsüljük a távolságot → annál hatékonyabb a keresés

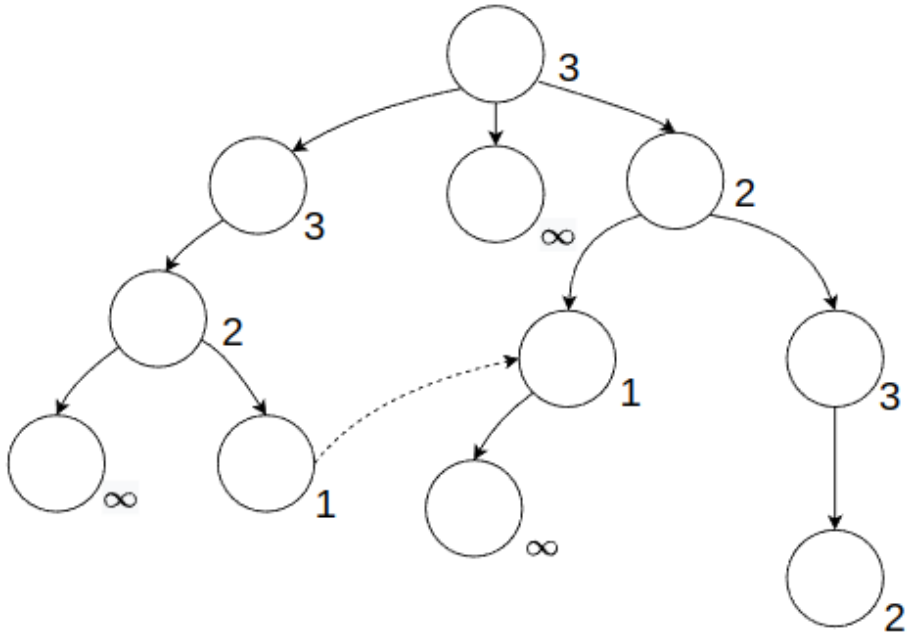
Munkám célja: hatékony A* alapú heurisztikák fejlesztése

Hierarchikus A* - Teljes ARG kifejtéssel

Nem csak a hibás állapothoz megtalálásához
szükséges csúcsokat látogatjuk meg



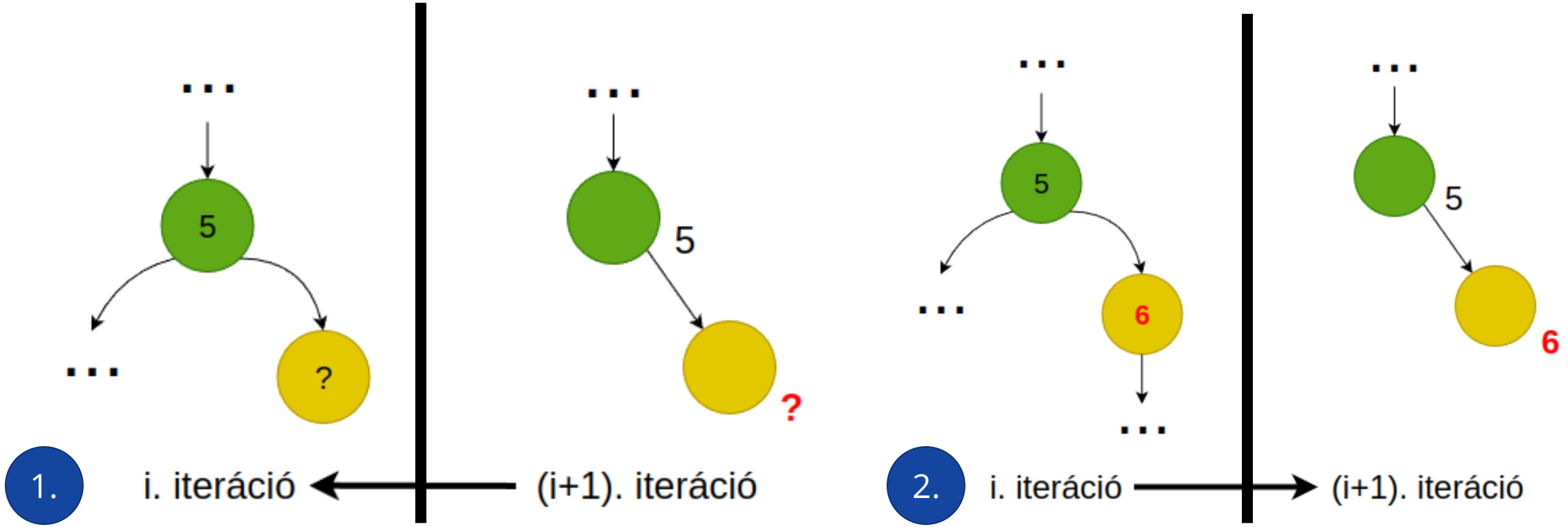
i. iteráció



(i+1). iteráció

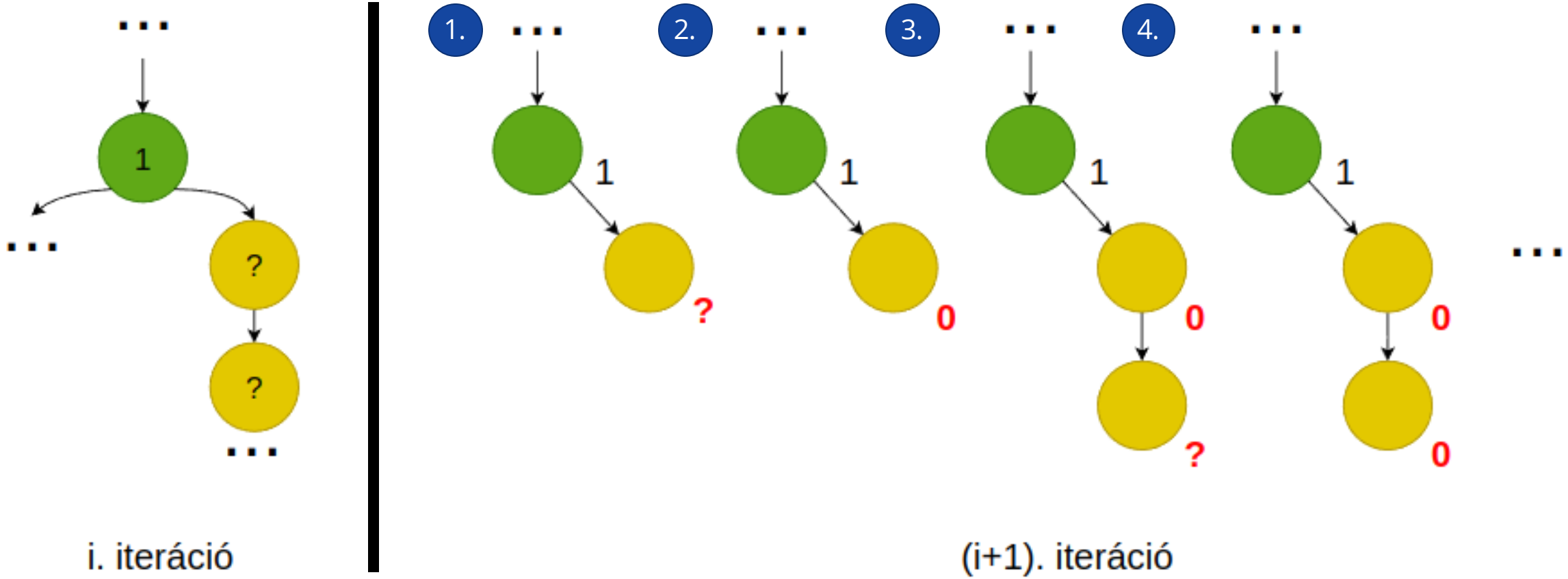
Hierarchikus A* - Igény szerinti ARG kifejtéssel

- Adott iterációban: csak első hibás állapotig bejárás
- Ha a korábbi ARG csúcsának távolsága (= heurisztika) nem ismert:
  - új A* keresés indítása a csúcstól annak távolságának megismerése érdekében



Hierarchikus A* - Csökkentéssel

- Heurisztika = { absztraktabb csúcs távolsága — ha az ismert
  MAX(szülő heurisztikája – 1, 0) — egyébként

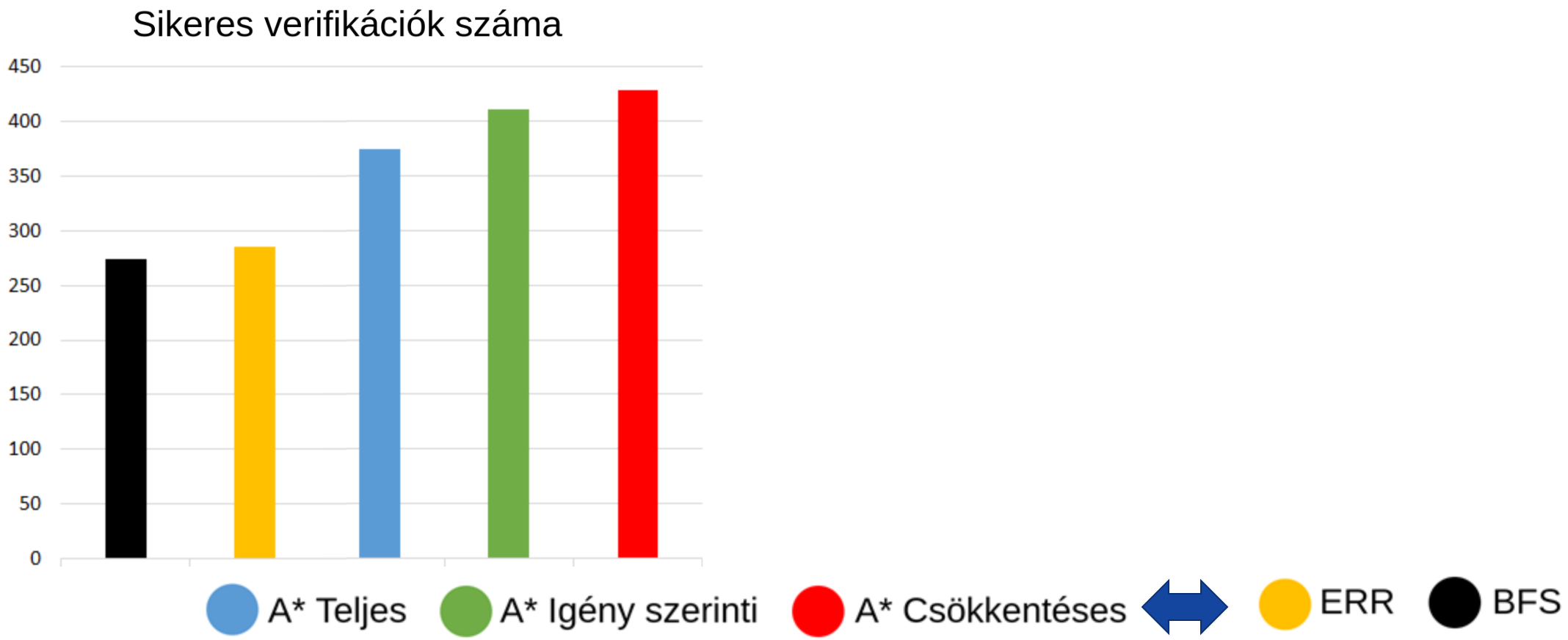


Mérések

- A* változatok **nyílt forráskódú** Thetába implementálása
- Ipari modellek verifikálása, különböző konfigurációk mellett:
  - Keresési algoritmusok
    - Hierarchikus A* változatok
    - BFS
    - ERR: egyszerű heurisztika modell szerkezete alapján
  - Absztrakciók
  - …

Mérések: szoftverek verifikálása

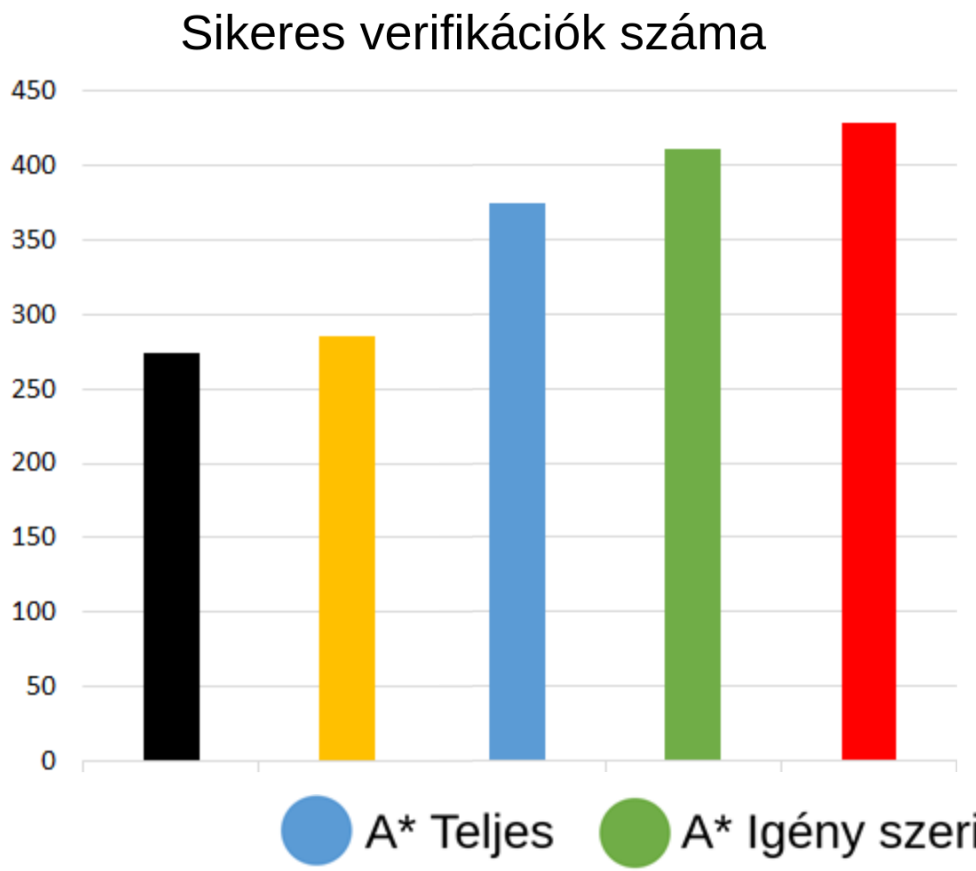
Az egyik részkonfiguráció mellett mindegyik A* változat jelentősen jobban teljesít

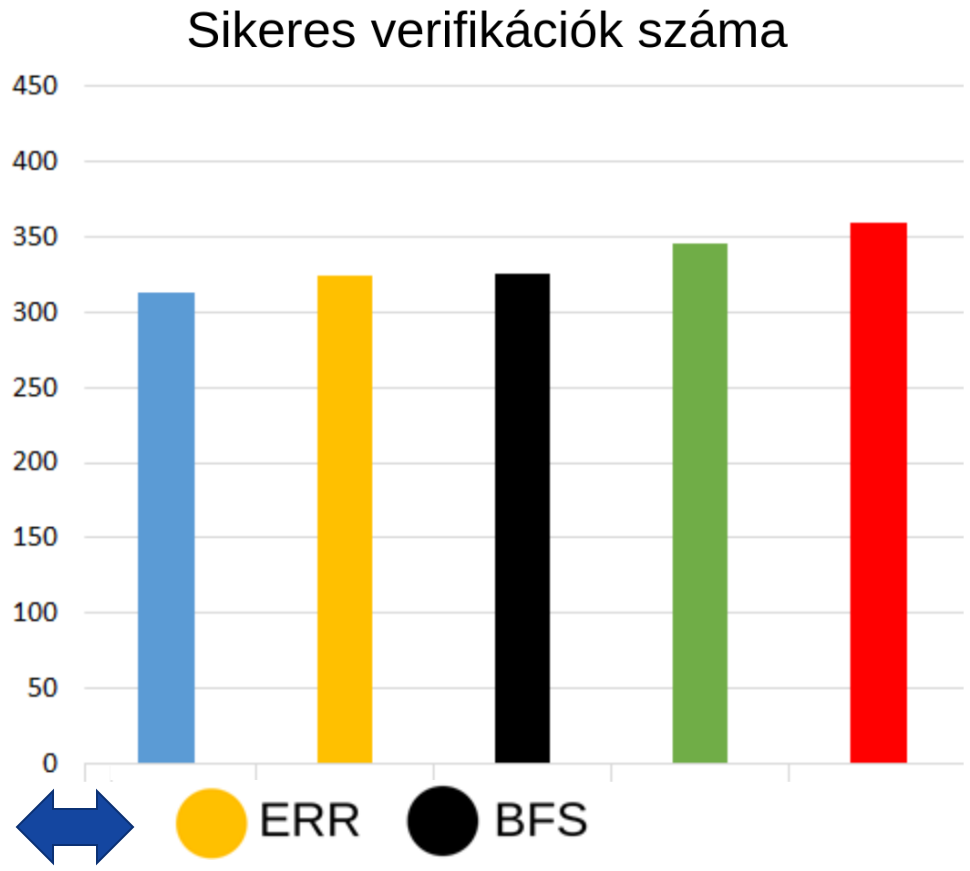


Mérések: szoftverek verifikálása

Az egyik részkonfiguráció mellett mindegyik A* változat jelentősen jobban teljesít

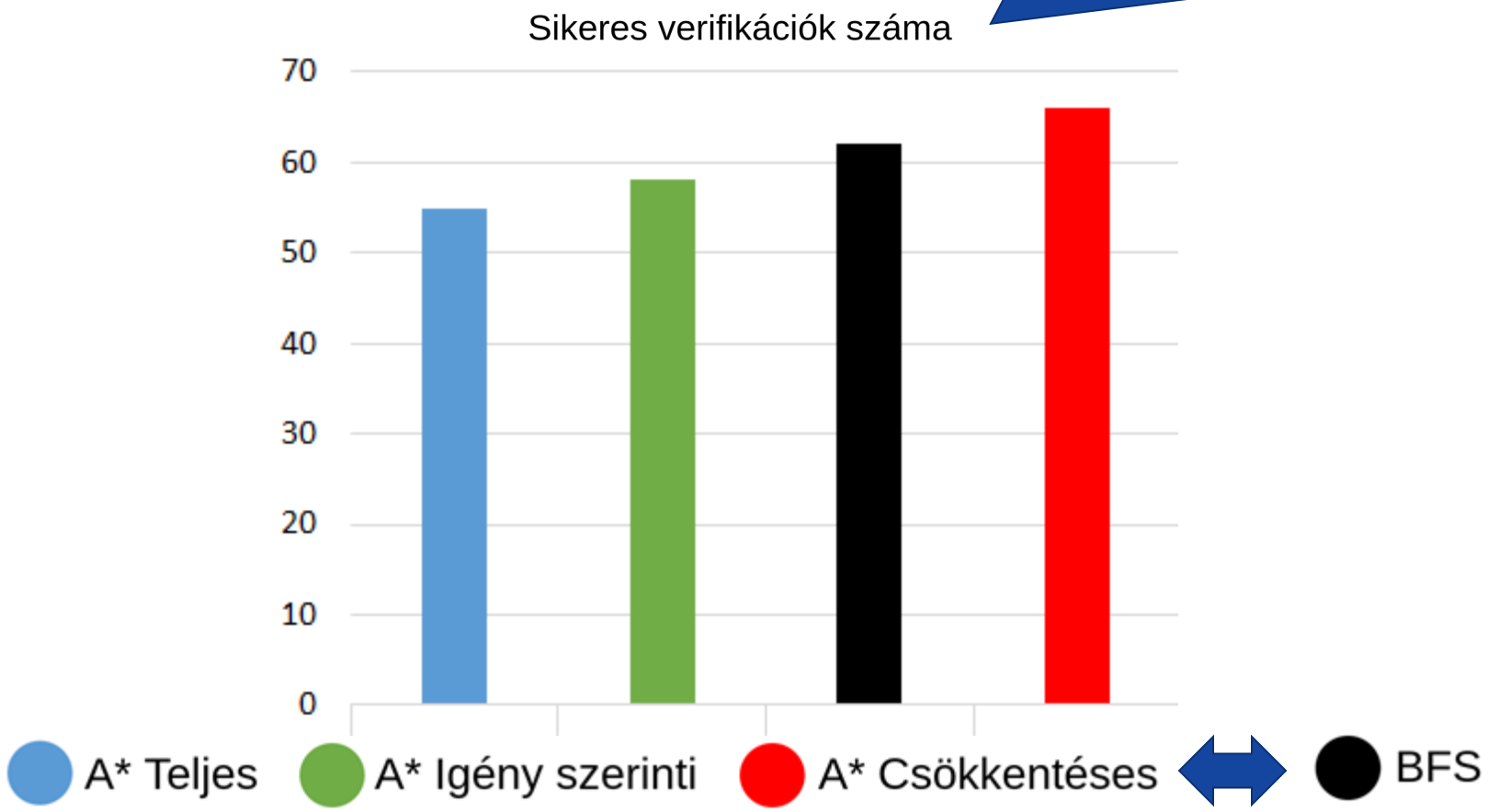
Míg a másik részkonfiguráció mellett csak az Igény szerinti és a Csökkentéses teljesít jobban





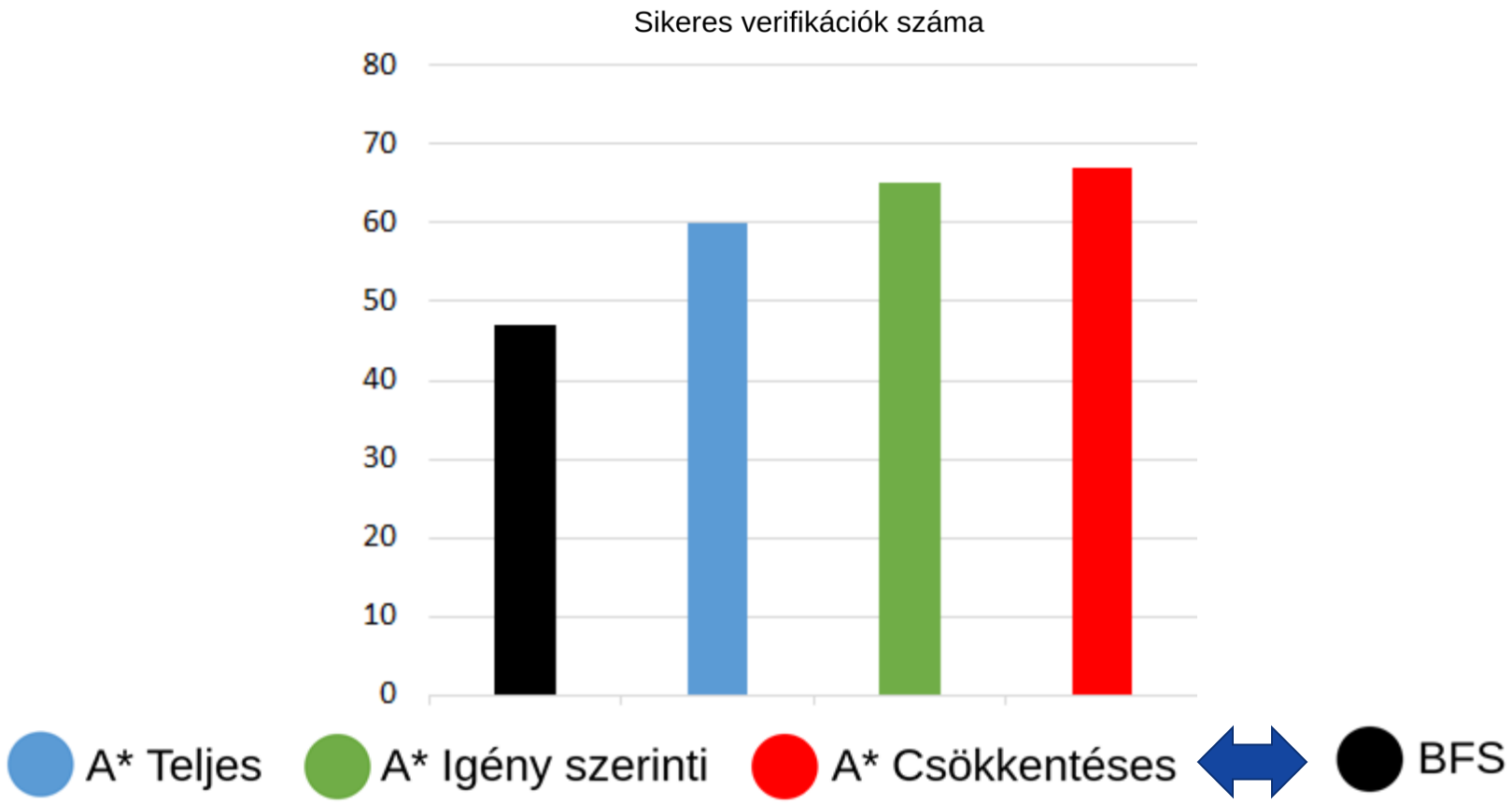
Mérések: állapottérképek verifikálása

Bár ebben a konfigurációban a Teljes és az Igény szerinti kevesebb modellt képes verifikálni, azonban a Csökkentéses változat még mindig jobban tud teljesíteni



Mérések: állapottérképek verifikálása

Az állapottérképek verifikálása során is van olyan konfiguráció, ahol az összes A* változat tud jelentősen gyorsítani a verifikáción



Összefoglalás

- Elméleti eredményeim:
  - Hierarchikus A* algoritmus elméleti kidolgozása és a legrövidebb ellenpélda garantálásának matematikai bizonyítása
  - Különböző hierarchikus A* algoritmus változatok kidolgozása
    - Teljes ARG kifejtés
    - Igény szerinti ARG kifejtés
    - Csökkentésen alapuló

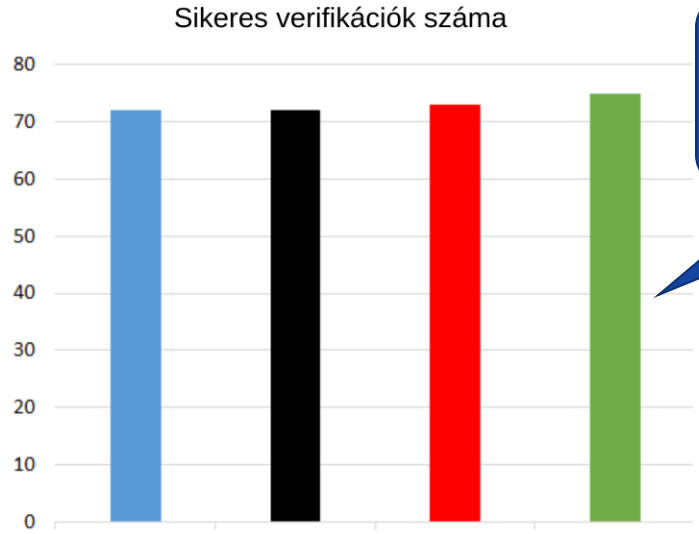
- Gyakorlati eredményeim:
  - **Nyílt forráskódú** implementáció a **Theta** modellellenőrző keretrendszerben
  - Teljesítmény kiértékelése benchmark mérések futtatásával

Bírálói kérdések

1. Különböző absztrakciós módszerekre az összehasonlításra került módszerek teljesítménye jelentősen eltér. Tud erre esetleg valamilyen magyarázatot adni?

Olyan kérdéskör, amit a jövőben alaposan meg kell vizsgálnom.

Mindegyik A* változat jobban teljesít



Nem mutat semelyik keresési stratégia se eltérést

Bírálói kérdések

2. Az időlimit alatt megoldott tesztesetek száma szerepel a mérési eredményeken.
   - Honnan tudható, hogy a megoldott tesztesetek száma elegendő?
   - Mit jelent egyáltalán az elegendő?
   - Van-e egy minimum érték, vagy csak az egymáshoz való összehasonlíthatóság miatt van jelentősége?

   A tesztesetek az egyes verifikálandó rendszereket jelentik, melyeknek ismert a helyessége. Sajnos a dolgozatban kicsit pongyolán használtam a kifejezést, ez okozhatta a félreértést. Minden megoldott “teszteset” sikeres verifikációt jelent.
   Minél több tesztesetet oldunk meg, az annál több rendszer sikeres verifikációját jelenti.

Bírálói kérdések

3. Mikor tudható, hogy egy adott vizsgálat lefedte az összes esetet?

   Mivel az absztrakció felülbecsli a lehetséges lefutásokat, így ha egy CEGAR iterációban nem található hibás állapot, akkor a finomabb lefutásokban sem, így a nem absztrakt lefutások között sem.

Összefoglalás

- Elméleti eredményeim:
  - Hierarchikus A* algoritmus elméleti kidolgozása és helyességének matematikai bizonyítása
  - Különböző hierarchikus A* algoritmus változatok kidolgozása
    - Teljes ARG kifejtés
    - Igény szerinti ARG kifejtés
    - Csökkentésen alapuló

- Gyakorlati eredményeim:
  - **Nyílt forráskódú** implementáció a **Theta** modellellenőrző keretrendszerben
  - Teljesítmény kiértékelése benchmark mérések futtatásával