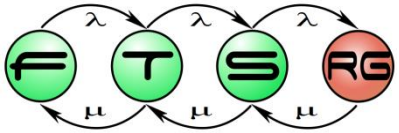
CEGAR-alapú modellellenőrzés

Hajdu Ákos

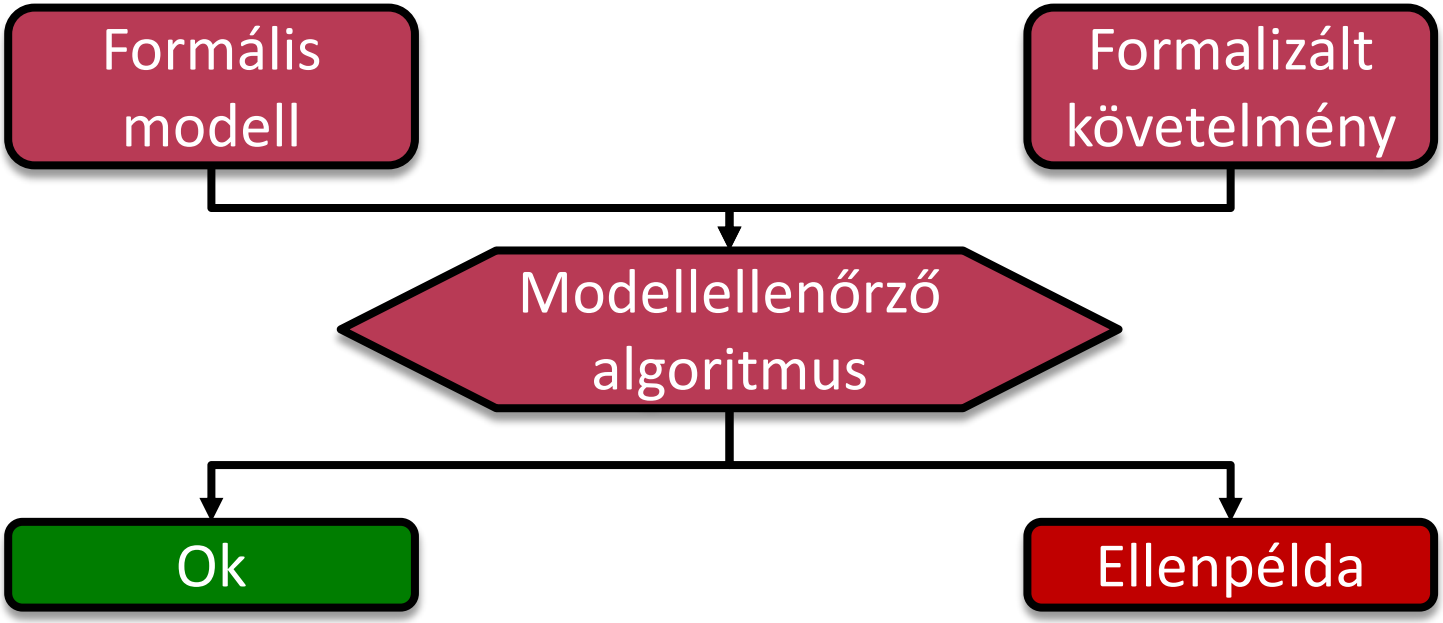
akos.hajdu@inf.mit.bme.hu

Szoftver verifikáció és validáció

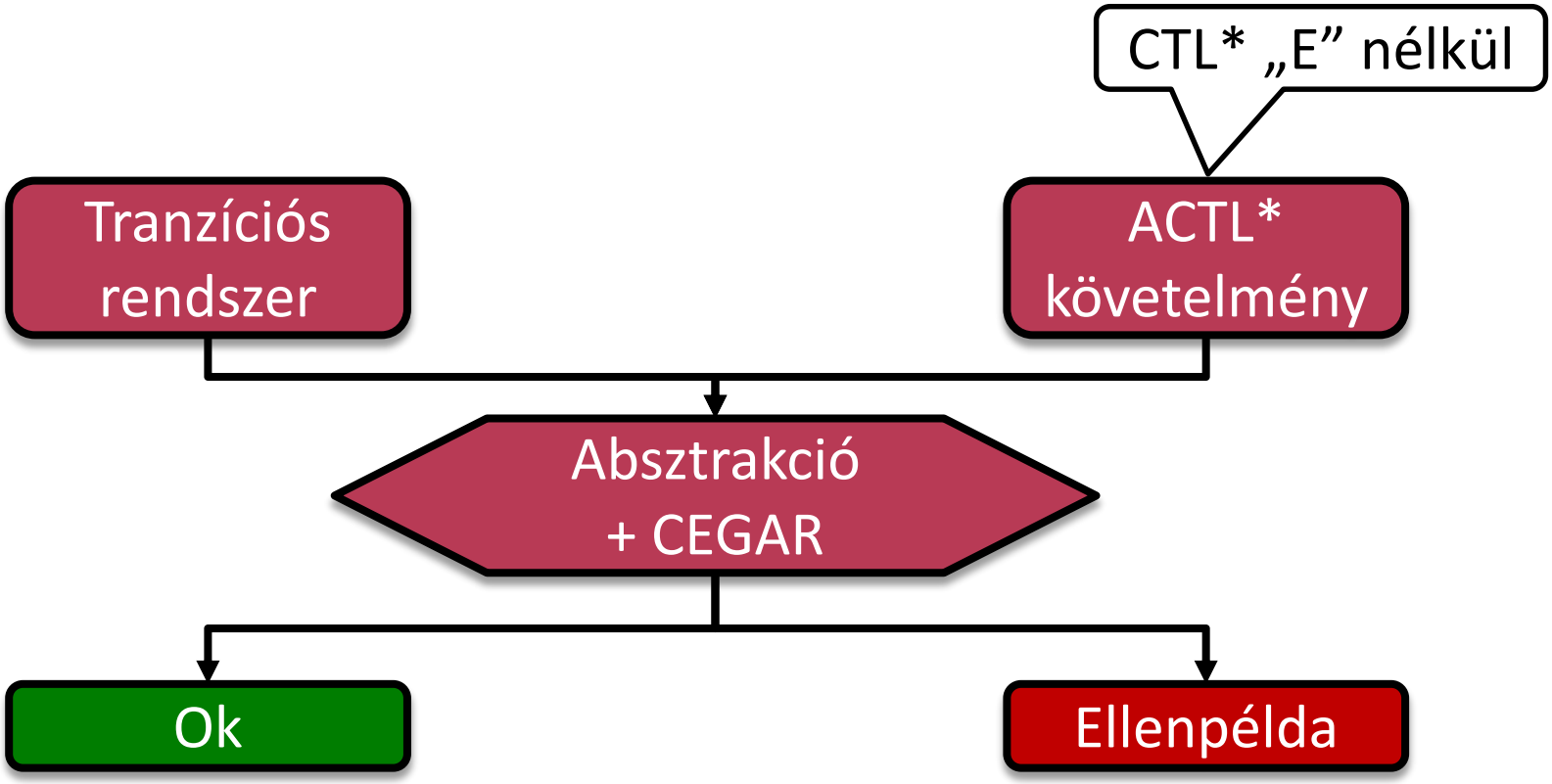
2015.11.04.



Ismétlés - modellellenőrzés

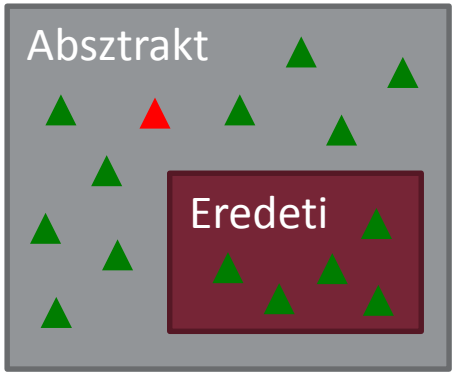


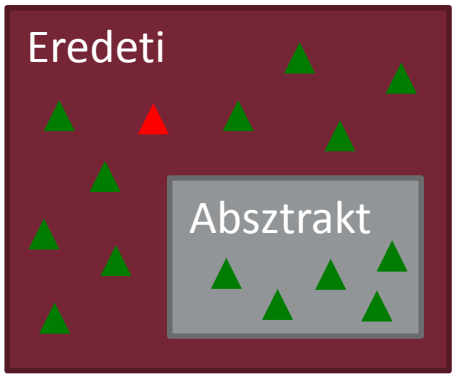
Az előadás témája



Absztrakció

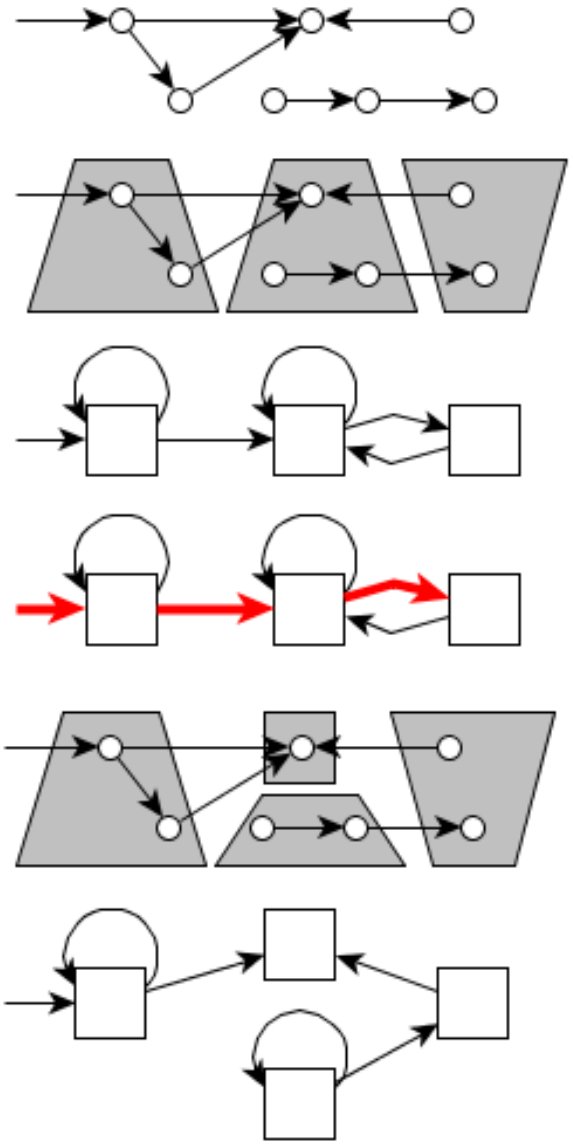
- Absztrakció
  - Általános matematikai eszköz
  - Egyszerűsítés részletek elhagyásával
    - „Megvan az ára”
- Absztrakció elterjedt típusai
  - Felülbecslés *(egzisztenciális absztrakció)*
    - Lazítás a megkötéseken → új viselkedések
    - „False negatives”
  - Alulbecslés *(univerzális absztrakció)*
    - Viselkedések eltávolítása
    - „False positives”
  - Pontos eredmény
    - Eldönthetetlen problémák





Absztrakció

- Egzisztenciális absztrakció
  - Absztrakt állapot = több konkrét állapot
    - Leképezés sokféle lehet (pl. predikátum)
  - Minden viselkedést megtart, de újakat hozhat be
    - Követelmény teljesül → eredetileg is
  - Ellenpélda → le kell ellenőrizni az eredeti modellben
    - Eredeti modell egy részének bejárása
    - Hamis ellenpélda → absztrakciófinomítás
  - CEGAR: CounterExample Guided Abstraction Refinement



Tranzíciós rendszerek (TTMC)

```
property safe : {

    local var reset : integer
    local var x : integer
    local var y : integer

    invariant reset >= 0
    invariant reset <= 1
    invariant x >= 0
    invariant x <= 2
    invariant y >= 0
    invariant y <= 2

    initial reset = 0
    initial x = 0
    initial y = 1

    transition reset' >= 0 and reset' <= 1
    transition x' = (
        if reset = 1 then 0
        else if x < y then x + 1
        else if x = y then 0
        else x
    )
    transition y' = (
        if reset = 1 then 0
        else if x = y and not y = 2 then y+1
        else if x = y then 0
        else y
    )
} models G (x < y or reset = 1)
```

Változók

Tranzíciós reláció

Nemdet.

Invariánsok

Függőségek

Kezdeti értékek

Követelmény

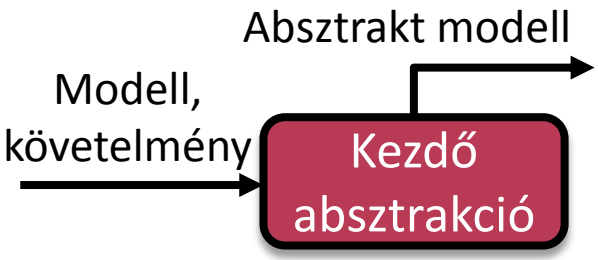
PREDIKÁTUMABSZTRAKCIÓ

Kezdő absztrakció

- Predikátumabsztrakció
  - Két konkrét állapot egy absztrakt állapotban van, ha a predikátumok alapján nem megkülönböztethetők
- Absztrakció kiszámolása
  - Konkrét állapotok felsorolása és összevonása
  - Példában 3x3 konkrét állapot → 5 absztrakt
  - Állapottér robbanás ☹

Változók:
x, y; $D_x = D_y = \{0, 1, 2\}$
Predikátumok:
(x=y), (x<y), (y=2)

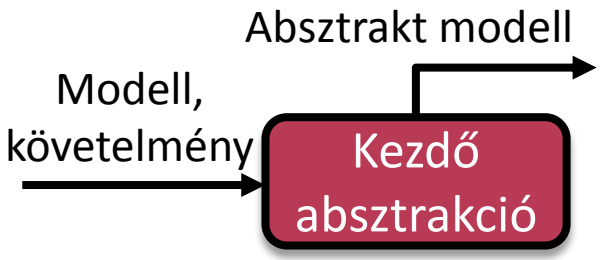
y\x	0	1	2
0	(x=y)		
1	(x<y)	(x=y)	
2	(x<y) (y=2)	(x<y) (y=2)	(x=y) (y=2)



Kezdő absztrakció

- Absztrakció kiszámolása (más módszer)
  - Csak az absztrakt állapotok felsorolása
  - $2^{|P|}$, ahol $|P|$ a predikátumok száma
  - Példa:
    - 3 predikátum → $2^3 = 8$ absztrakt állapot
    - Nem mind lehetséges, kiszűrhető SMT solver-rel
    - Így is megkapjuk az 5 állapotot

	x=y	x<y	y=2
1	X	X	X
2	X	X	✓
3	X	✓	X
4	X	✓	✓
5	✓	X	X
6	✓	X	✓
7	✓	✓	X
8	✓	✓	✓



Kezdő absztrakció

- Absztrakció kiszámolása (más módszer)
  - Absztrakt tranzíció: legalább egy konkrét
    - SMT: $P_0(x_0, y_0) \wedge R(x_0,y_0,x_1,y_1) \wedge P_1(x_1,y_1)$

Forrásállapot predikátumai

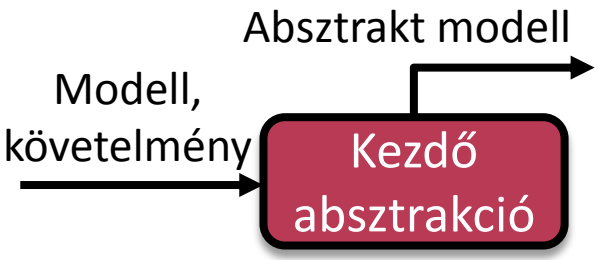
Tranzíciós reláció

Célállapot predikátumai

  - Absztrakt kezdőállapot: legalább egy konkrét
    - SMT: $P_0(x_0, y_0) \wedge I(x_0,y_0)$

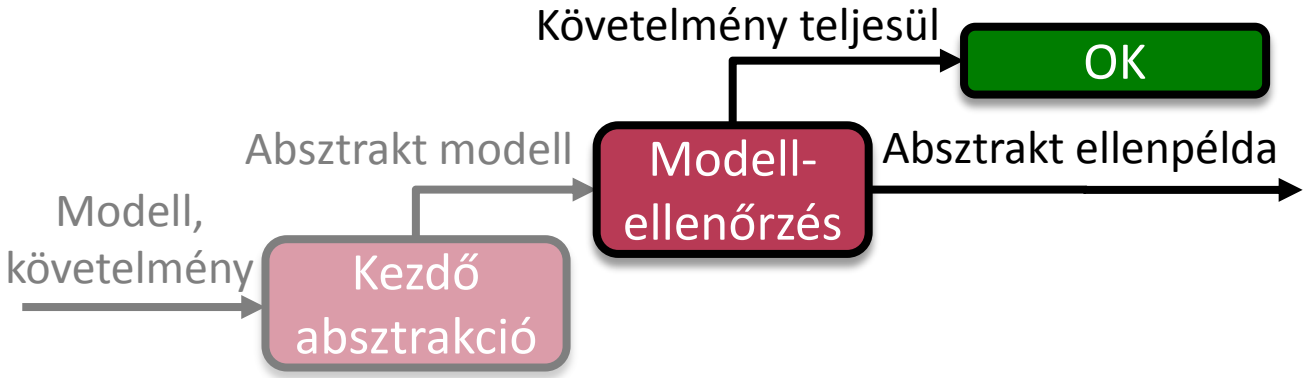
Kezdőállapot formula

	x=y	x<y	y=2
1	X	X	X
2	X	X	✓
3	X	✓	X
4	X	✓	✓
5	✓	X	X
6	✓	X	✓
7	✓	✓	X
8	✓	✓	✓



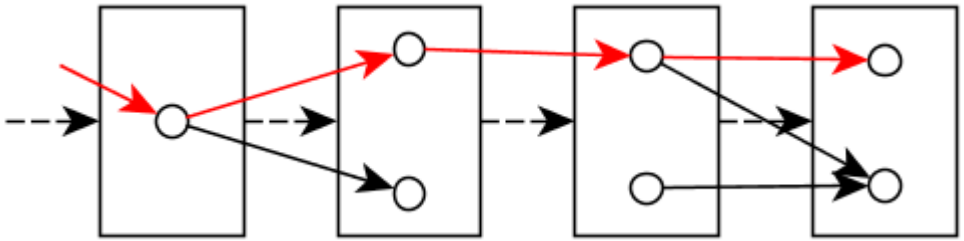
Modellellenőrzés

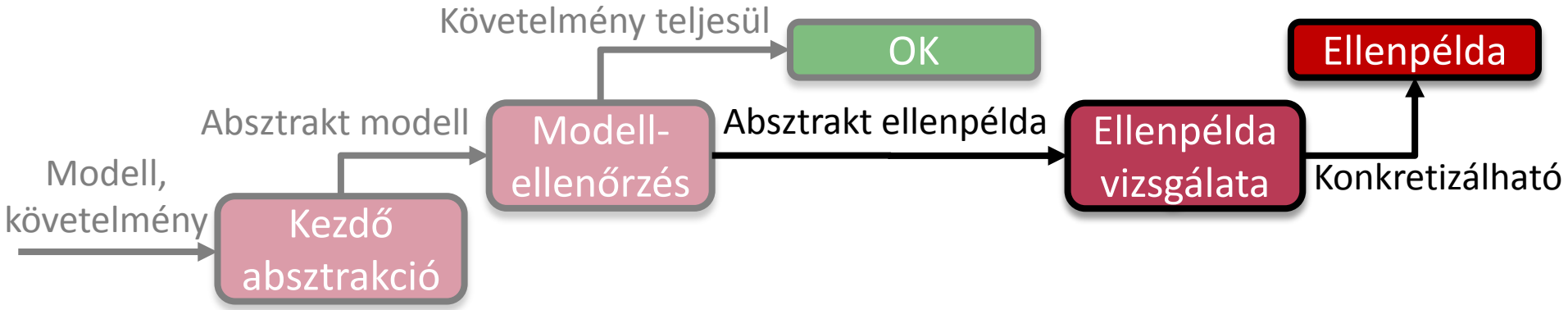
- ACTL* követelmény vizsgálata
  - CTL*, de csak univerzális (A) operátor
  - Előny: ellenpélda létezése
- Példa:
  - AG(ϕ) ellenpélda: olyan állapotba vezető út, ahol ϕ nem igaz
  - AF(ϕ) ellenpélda: végtelen (ciklikus) út, ahol ϕ sosem igaz



Ellenpélda vizsgálata

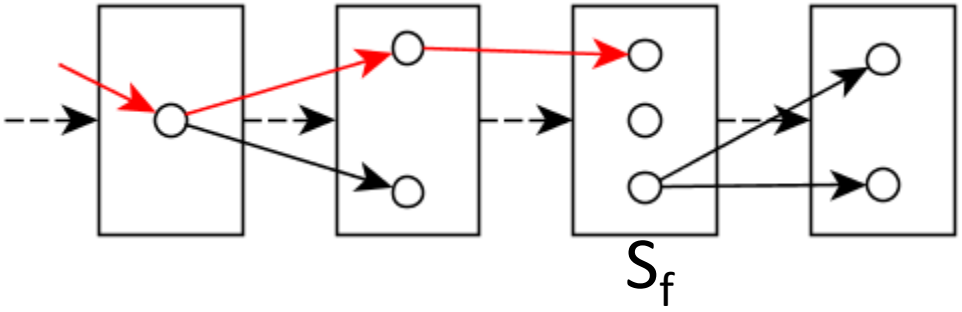
- Konkrét állapottér egy részének bejárása
  - Ciklus esetén kihajtogatás (polinom sokszor)
- Ha van konkrét állapotsorozat → valódi ellenpélda

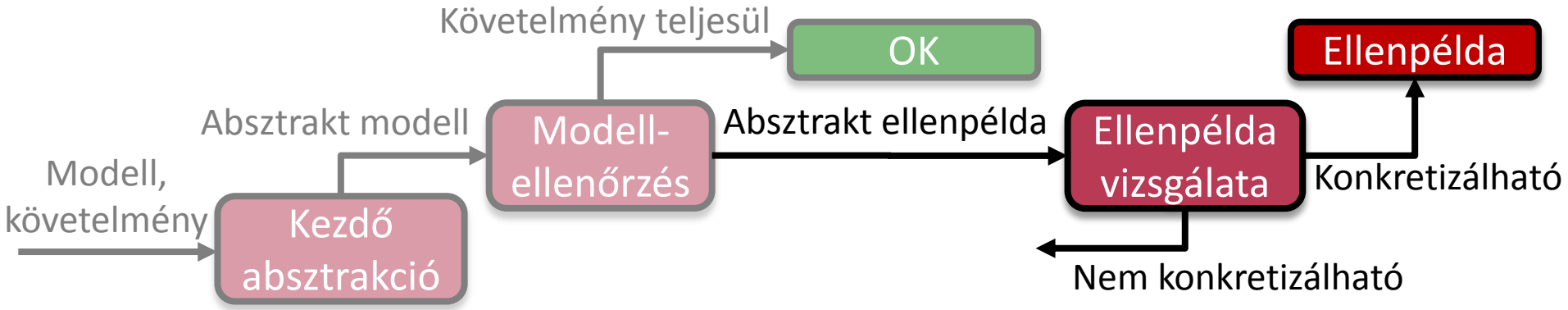




Ellenpélda vizsgálata

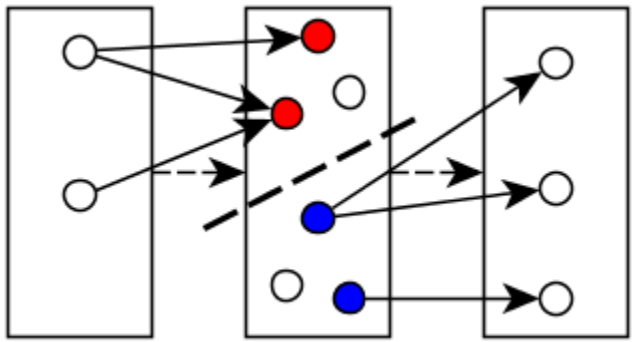
- Ha nincs konkrét állapotsorozat
  - Egy S_f állapotig van út és (S_f, S_{f+1}) között is, de ezek különálló konkrét utak (S_f: „failure” állapot)
  - Absztrakció finomítására van szükség

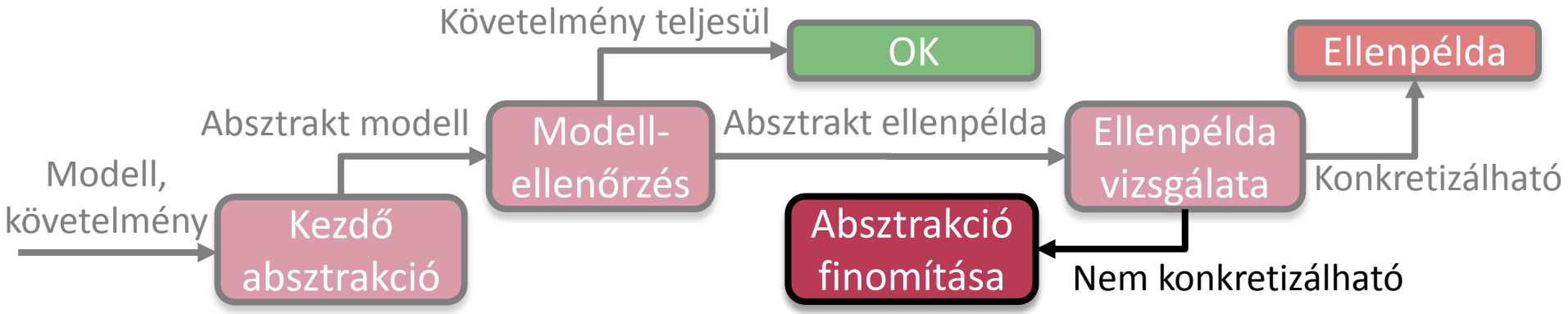




Absztrakció finomítása

- Konkrét állapotok csoportosítása a „failure” állapoton belül
  - D = „Dead-end”: elérhető
  - B = „Bad”: következő állapotra lép
  - IR = „Irrelevant”: többi
- Cél: D és B szétválasztása
  - Konkrét állapotok felsorolása nélkül?

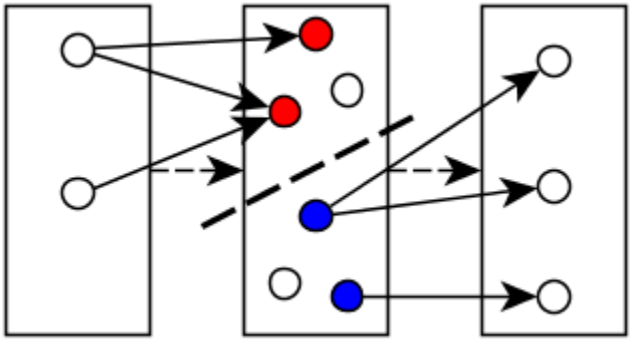


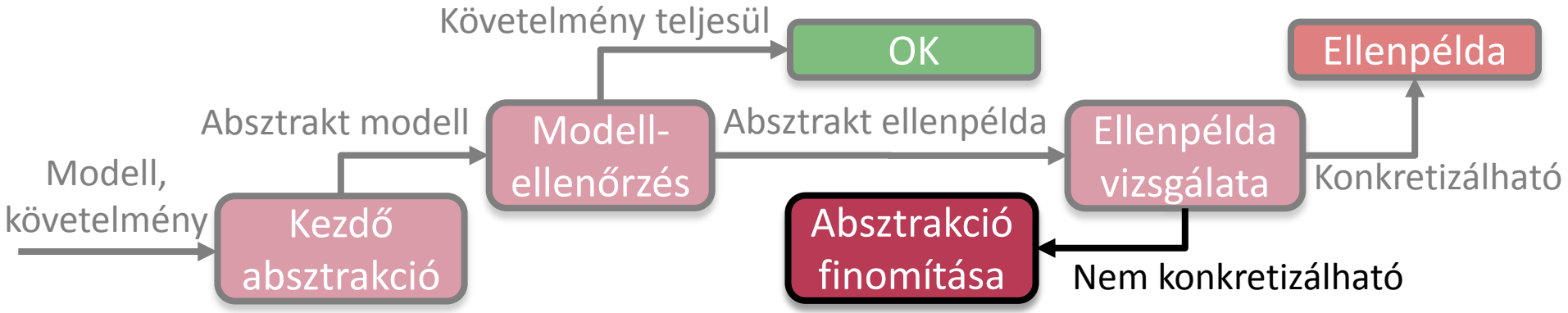


Absztrakció finomítása

- D és B szétválasztása
  - Karakterizálás formulákkal
  - D: $I(s_0) \wedge P_0(s_0) \wedge R(s_0,s_1) \wedge P_1(s_1) \wedge \ldots \wedge R(s_{f-1},s_f) \wedge P_f(s_f)$
  - B: $P_f(s_f) \wedge R(s_f,s_{f+1}) \wedge P_{f+1}(s_{f+1})$
  - D ∧ B kielégíthetetlen

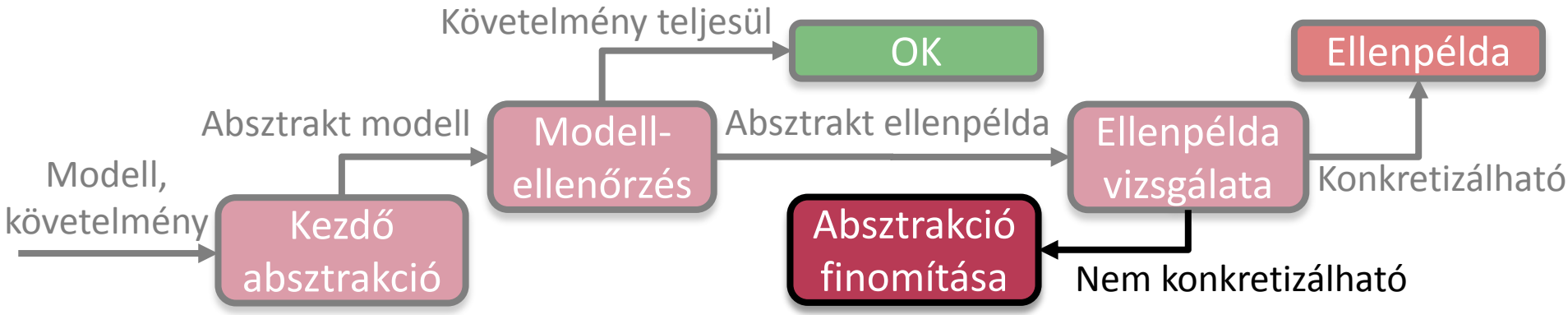
Kezdőállapot: $I(s_0) \wedge P_0(s_0)$
Átmenet: $P_0(s_0) \wedge R(s_0,s_1) \wedge P_1(s_1)$





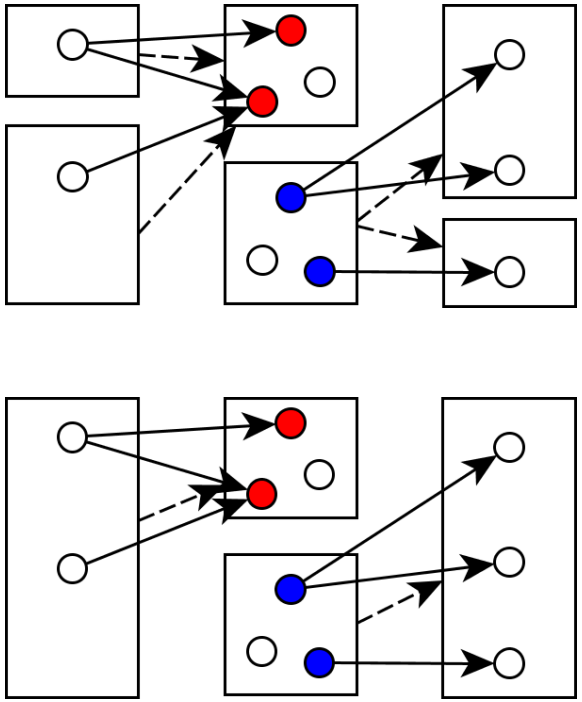
Absztrakció finomítása

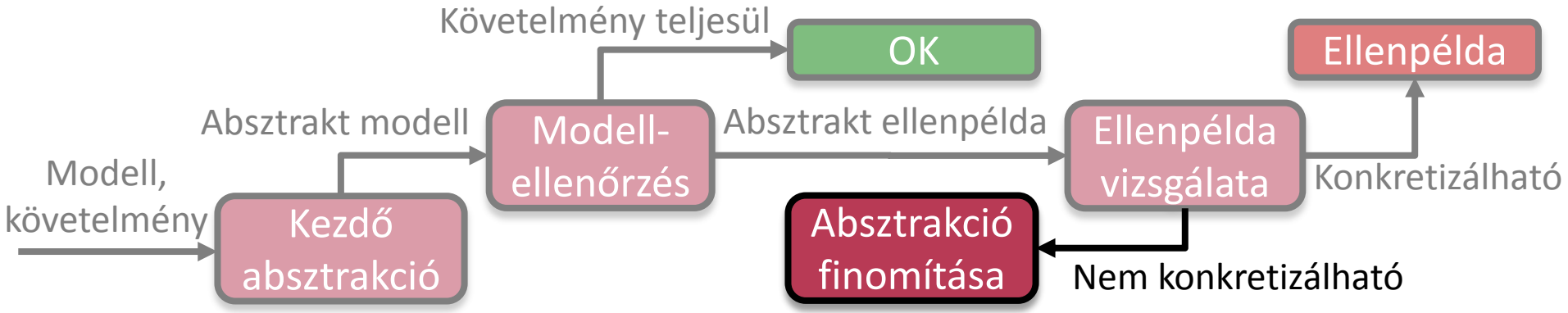
- Craig interpoláció
  - D ∧ B kielégíthetetlen → létezik ϕ formula, amely:
    - D → ϕ „Dead-end”-ek benne vannak
    - ϕ ∧ B kielégíthetetlen „Bad”-ek nincsenek benne
    - ϕ csak D és B közös szimbólumait tartalmazza Csak s_f -re vonatkozik



Absztrakció finomítása

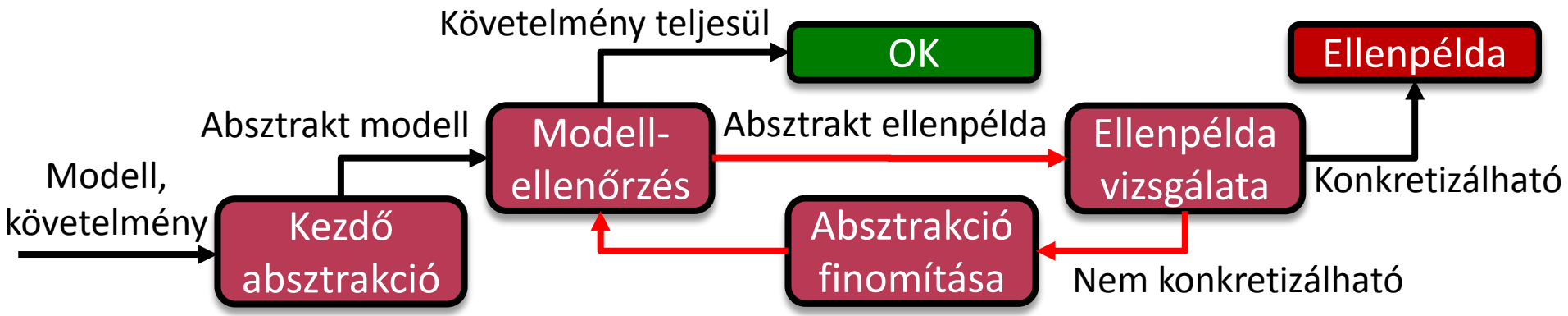
- Absztrakciófinomítás ϕ alapján
  - Predikátumhalmaz bővítése
    - P := P U {ϕ}
    - Állapotok kétszereződése
  - Egy állapot vágása
    - P_f U {ϕ} és P_f U {¬ϕ}
    - „Lazy abstraction”





CEGAR algoritmus

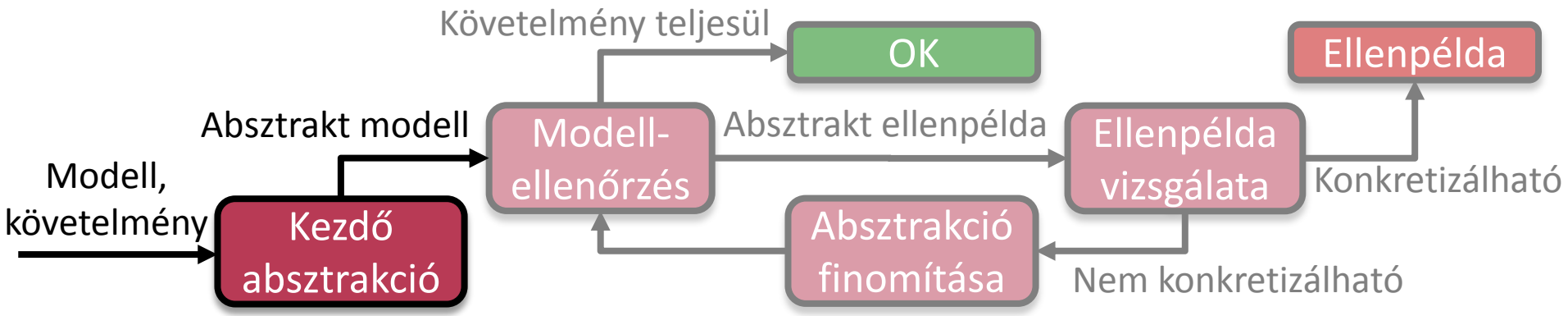
- Absztrakciófinomítási ciklus ismétlése amíg…
  - … a követelmény nem teljesül
  - … konkrét ellenpéldába nem futunk
- Véges értékkészletű változók → algoritmus terminálódik



LÁTHATÓ VÁLTOZÓK

Kezdő absztrakció

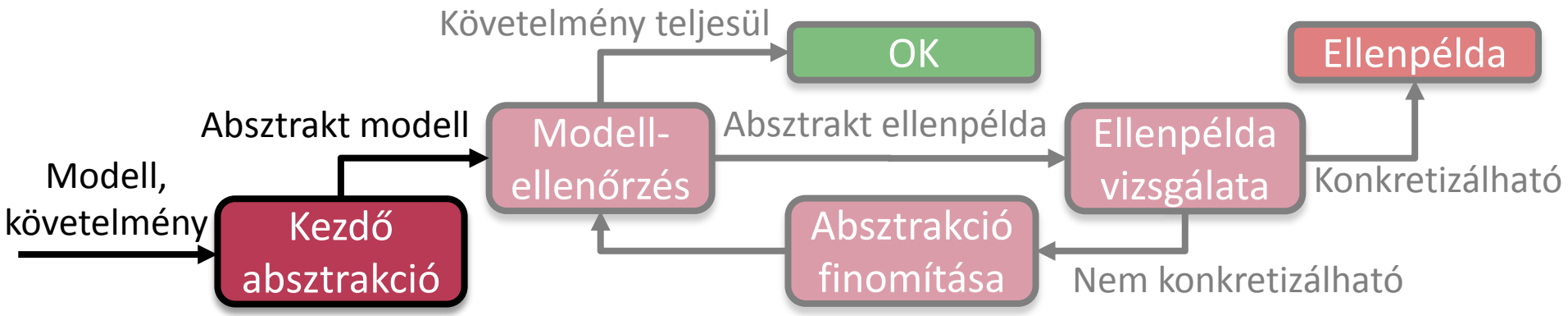
- Változók felosztása
  - Látható (Vis) és nem látható (Inv)
  - Két konkrét állapot egy absztrakt állapotban van, ha a látható változókon megegyeznek
  - Vis := követelményben szereplő változók



Kezdő absztrakció

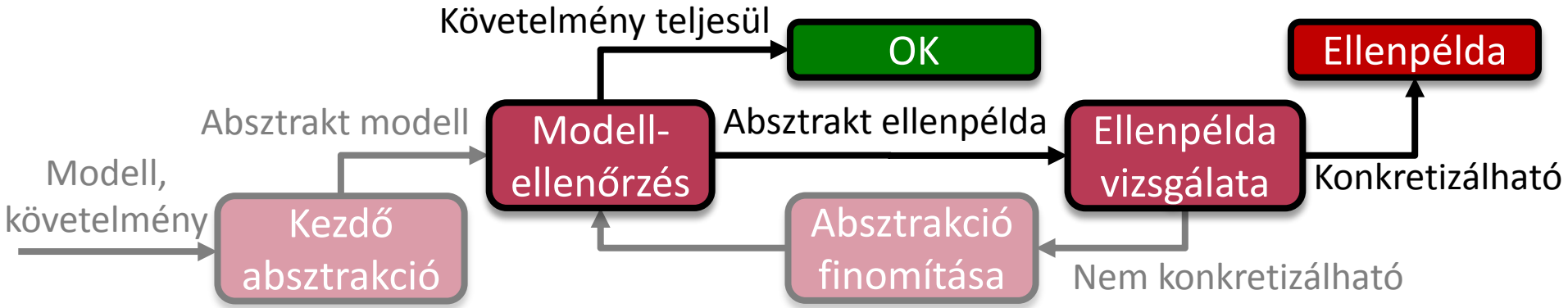
- Példa
  - $D_x = \{0, 1\}$, $D_y = \{0, 1, 2\}$, $D_z = \{0, 1\}$
  - Kezdetben Vis = {x, y}

	x=0	x=1
y=0	(x=0, y=0, **z=0**) (x=0, y=0, **z=1**)	(x=1, y=0, **z=0**) (x=1, y=0, **z=1**)
y=1	(x=0, y=1, **z=0**) (x=0, y=1, **z=1**)	(x=1, y=1, **z=0**) (x=1, y=1, **z=1**)
y=2	(x=0, y=2, **z=0**) (x=0, y=2, **z=1**)	(x=1, y=2, **z=0**) (x=1, y=2, **z=1**)



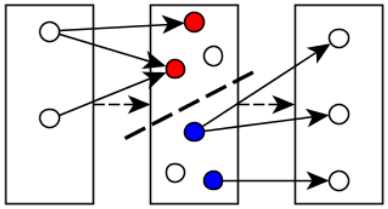
Modellellenőrzés & ellenpélda vizsgálata

- Úgy, mint predikátumabsztrakció esetén
  - Modellellenőrzés
    - Követelmény teljesül / ellenpélda
  - Ellenpélda vizsgálata
    - „Failure” állapot keresése
    - „Dead-end”, „Bad”, „Irrelenvant” kategorizálás



Absztrakció finomítása

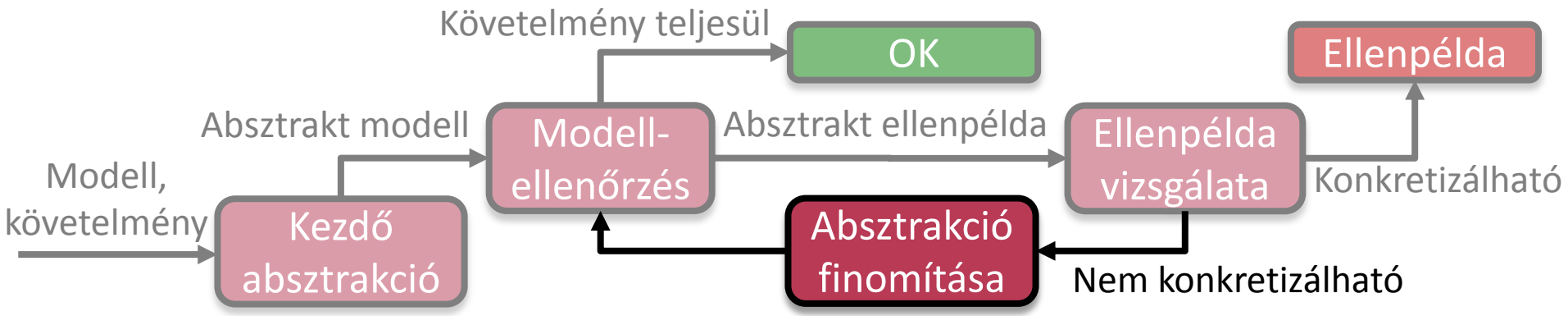
- Új változó amivel a „failure” állapot vágható
- Példa:



Vis = {x}
Inv = {y, z}
S_f: (x=0)

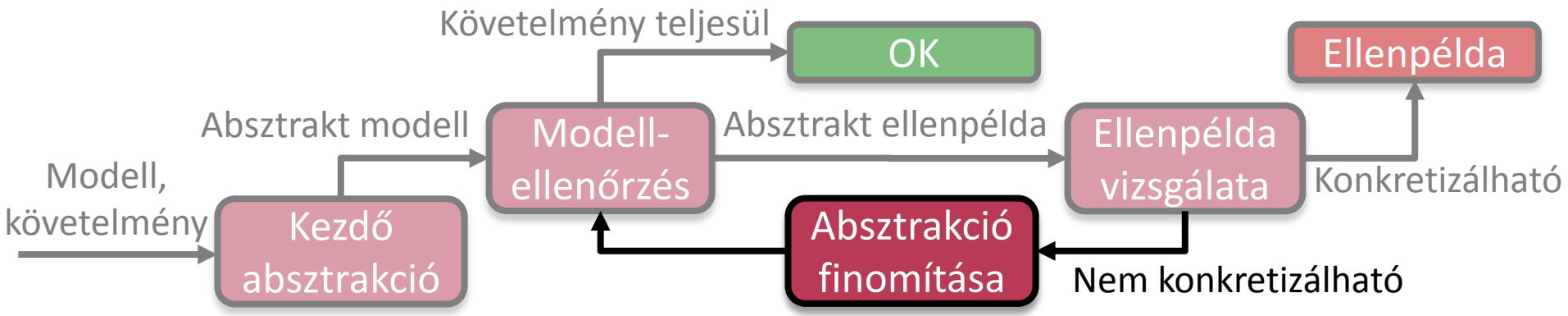
Dead-end	(x=0, y=1, z=1) (x=0, y=0, z=1)
Irrelevant	(x=0, y=1, z=0) (x=0, y=2, z=1)
Bad	(x=0, y=0, z=0) (x=0, y=2, z=0)

Vis = {x, z}
Inv = {y}



Absztrakció finomítása

- Új változók megtalálása
  - Interpolánsból
  - Egészértékű lineáris programozással (ILP)
  - Döntési fa tanulással (Decision tree learning)



Hivatkozások

- Clarke, Edmund, et al. **"Counterexample-guided abstraction refinement."** *Computer aided verification*. Springer Berlin Heidelberg, 2000.
- Clarke, Edmund M., Anubhav Gupta, and Ofer Strichman. **"SAT-based counterexample-guided abstraction refinement."** *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on* 23.7 (2004): 1113-1123.
- McMillan, Kenneth L. **"Applications of Craig interpolants in model checking."** *Tools and Algorithms for the Construction and Analysis of Systems*. Springer Berlin Heidelberg, 2005. 1-12.
- Beyer, Dirk, and Stefan Löwe. **"Explicit-state software model checking based on CEGAR and interpolation."** *Fundamental Approaches to Software Engineering*. Springer Berlin Heidelberg, 2013. 146-162.
- Ermis, Evren, Jochen Hoenicke, and Andreas Podelski. **"Splitting via interpolants."** *Verification, Model Checking, and Abstract Interpretation*. Springer Berlin Heidelberg, 2012.