

# Szoftver-modellellenőrzés absztrakciós módszerekkel

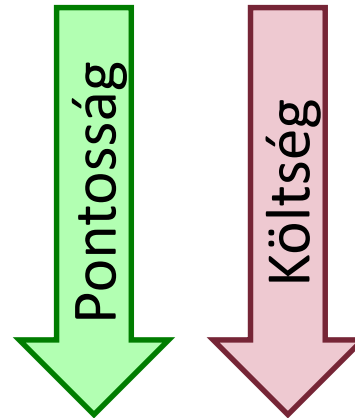
Dr. Hajdu Ákos, dr. Majzik István

BME Méréstechnika és Információs Rendszerek Tanszék

# Bevezető

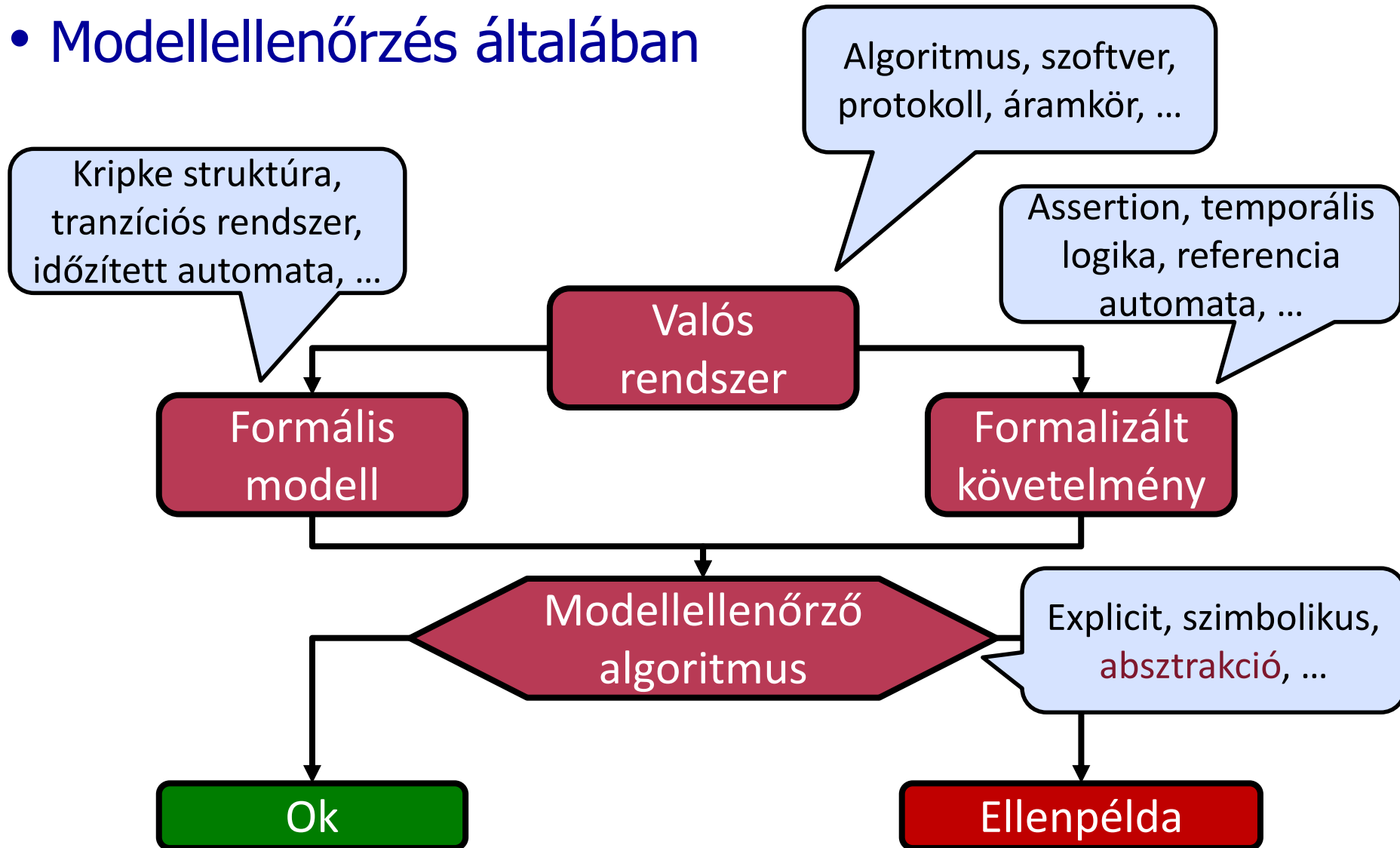
# Célok

- Motiváció
  - Forráskód közvetlen ellenőrzése
  - „Gombnyomásra” működjön
    - Komoly háttérismeretek nélkül
- Jellegzetes szoftverellenőrzési technikák
  - Statikus analízis
    - Hibaminta kereső
    - Absztrakt interpretáció
  - Dinamikus analízis
    - Modellellenőrzés



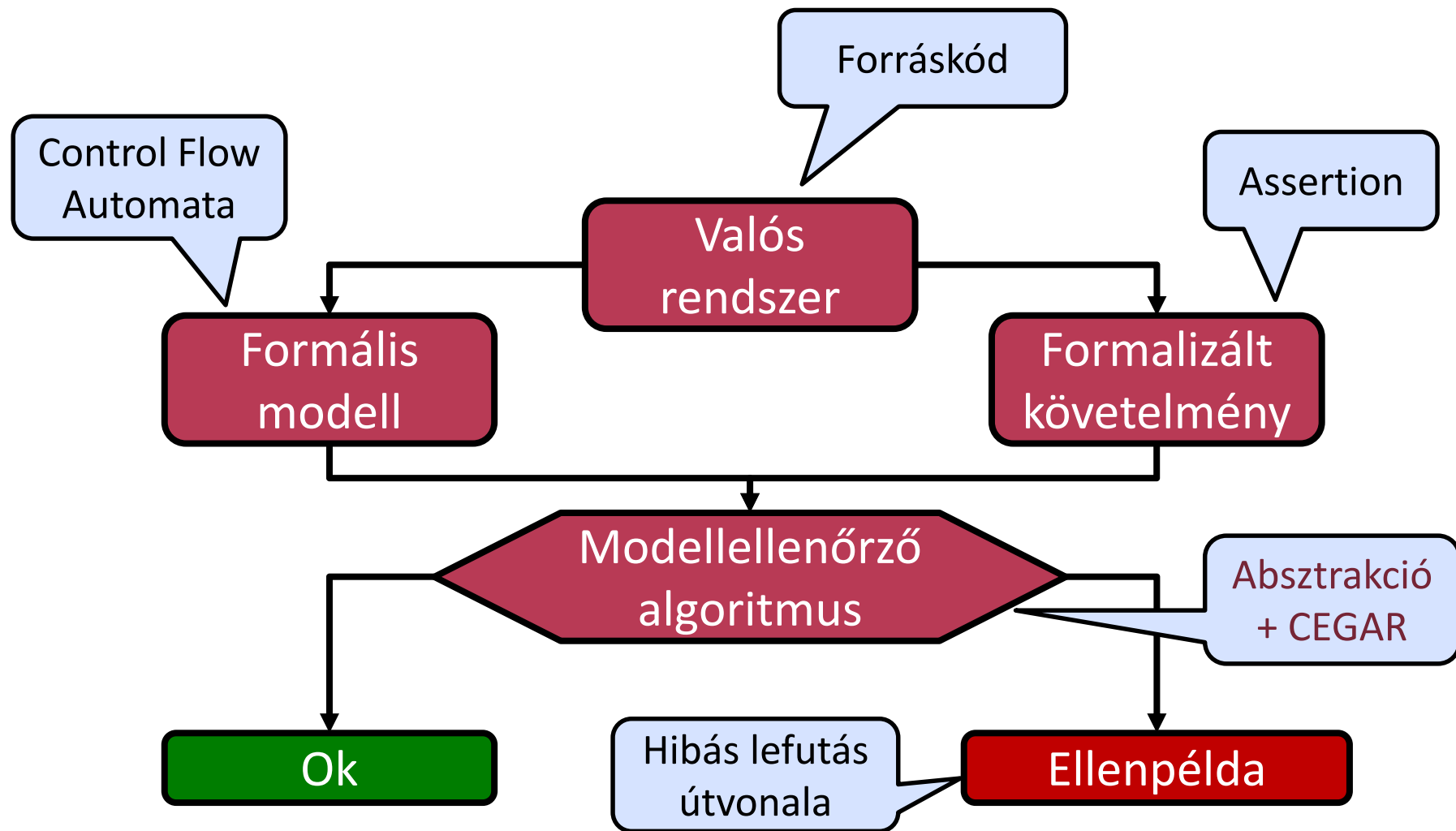
# Modellellenőrzés általában

- Modellellenőrzés általában



# Modellellenőrzés szoftvereken

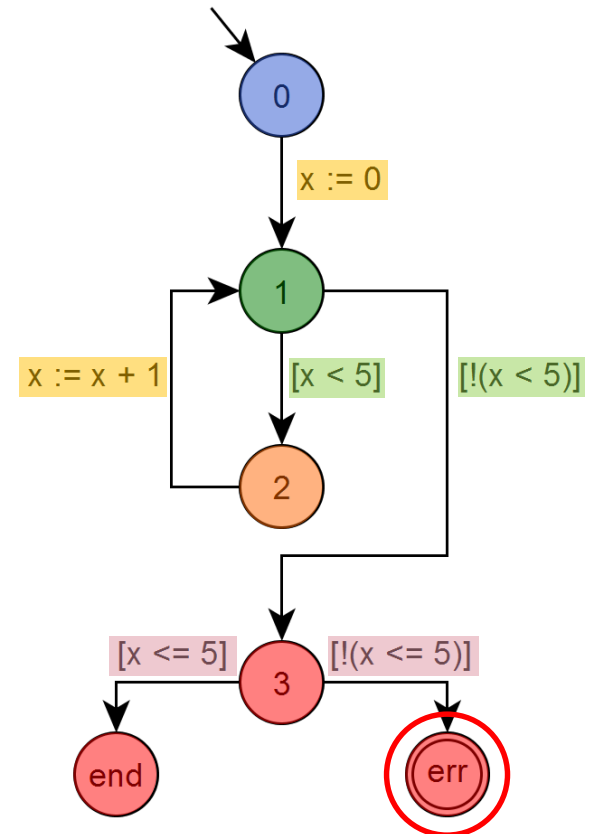
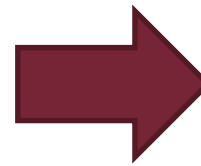
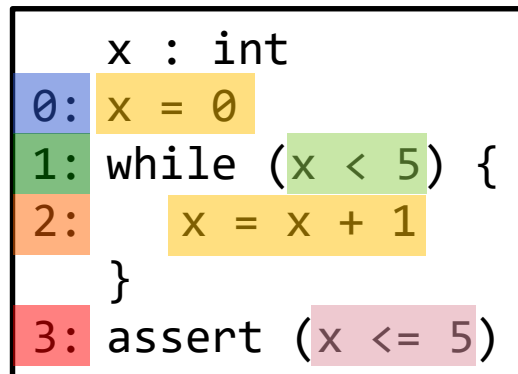
- Célok: Szoftver ellenőrzése absztrakcióval



# Szoftvermodell és követelmény

- Control-Flow Automata (CFA)

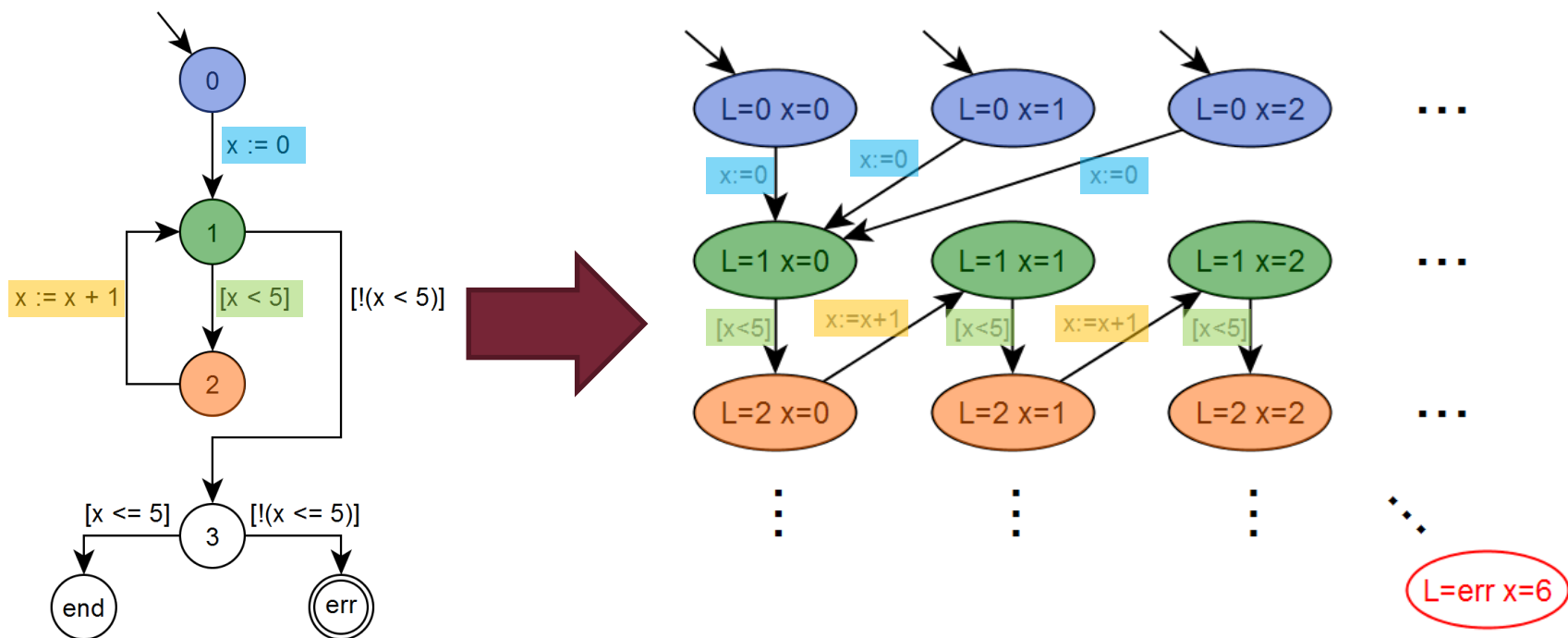
- Vezérlési helyek:  $L$  halmaz ( $l_0, l_1, \dots$ )
- Élek:  $G$  halmaz
  - Örfeltétel, értékadás, assertion a változók felett



- Tipikus követelmény: „error” hely (jelölés:  $l_E$ ) ne legyen elérhető
  - Itt: assertion megsértését reprezentálja

# Állapotok és átmenetek

- **Állapot:** vezérlési hely + változók értékei ( $L, x_1, x_2, \dots, x_n$ )
- **Átmenet:** művelet vagy feltétel
- **Probléma:** **Állapottér** robbanás az adatváltozók miatt
  - Pl.: 10 vezérlési hely, 2 db 32 bites int  $\rightarrow 10 \cdot 2^{32} \cdot 2^{32}$  lehetséges állapot
- **Cél:** Állapottér reprezentáció méretének csökkentése absztrakcióval



# Háttér: Matematikai logika alkalmazása

- **Propozicionális logika (nulladrendű)**

- Boole-logikai változók és operátorok
- SAT probléma: Formula kielégíthető-e
  - Példa: Korlátos modellellenőrzés
- Kifejezőerő nem mindig elégséges

$$\neg p \wedge (p \vee q)$$

- **Elsőrendű logika**

- Függvények, predikátumok, kvantorok
- Általános esetben nem eldönthető

$$\forall x, y \exists z: p(f(x, y), g(z))$$

- **Satisfiability Modulo Theories (SMT)**

- Elsőrendű logikai formulák korlátozva
- Interpretált szimbólumok
  - Pl. egész aritmetika

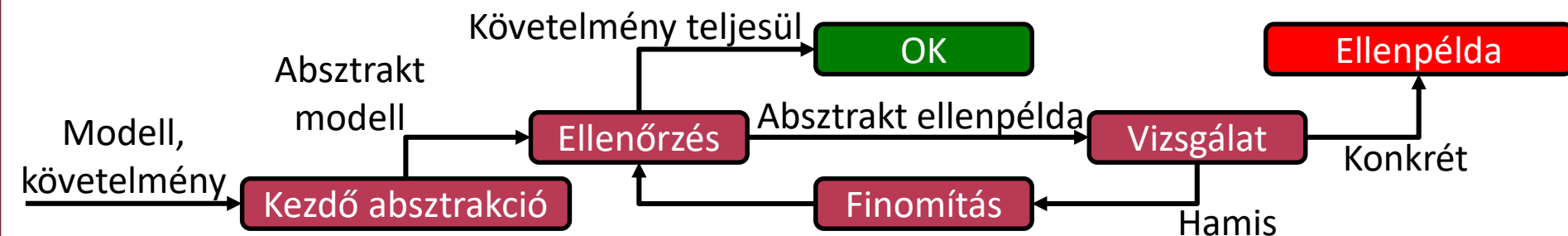
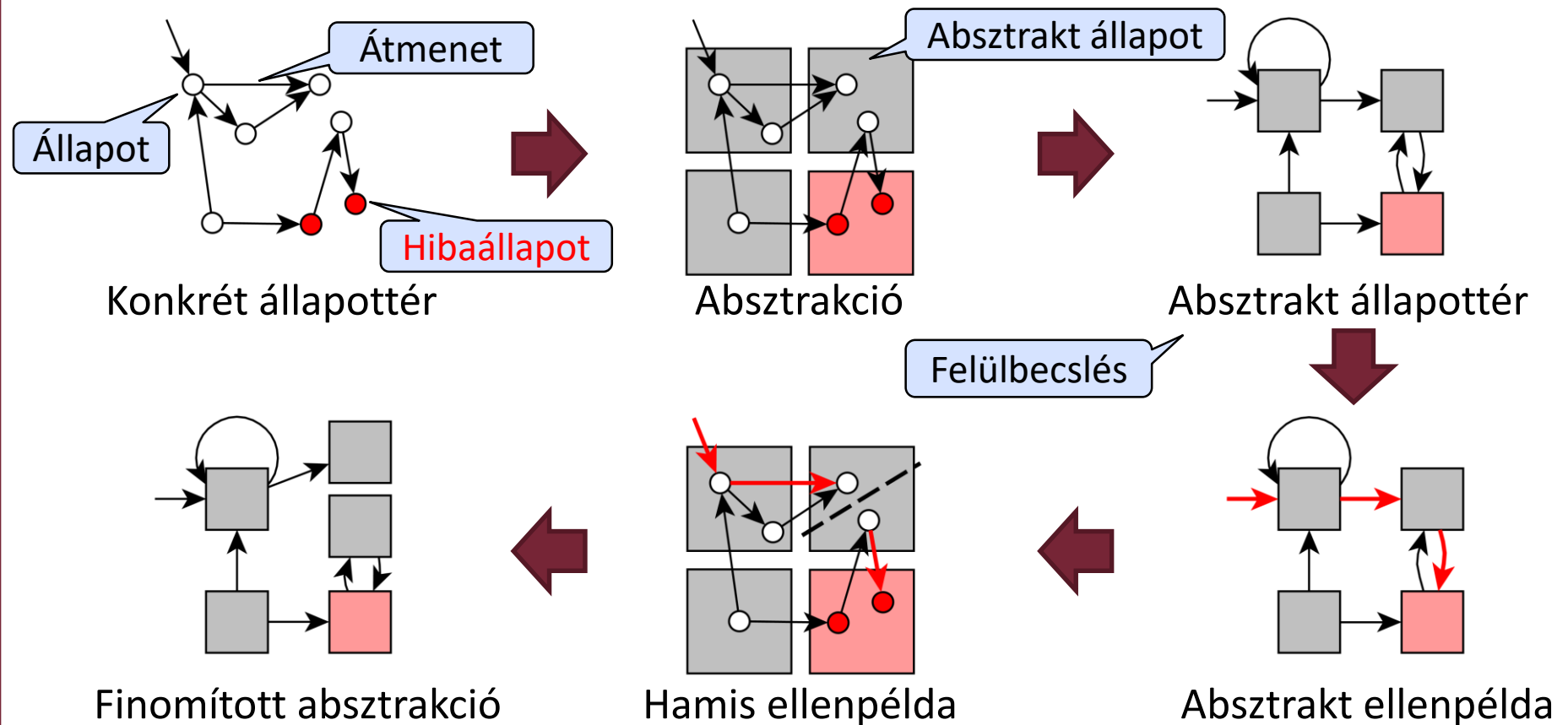
$$(x \leq y + 1) \wedge (y \geq 3)$$



# Counterexample-Guided Abstraction Refinement (CEGAR)

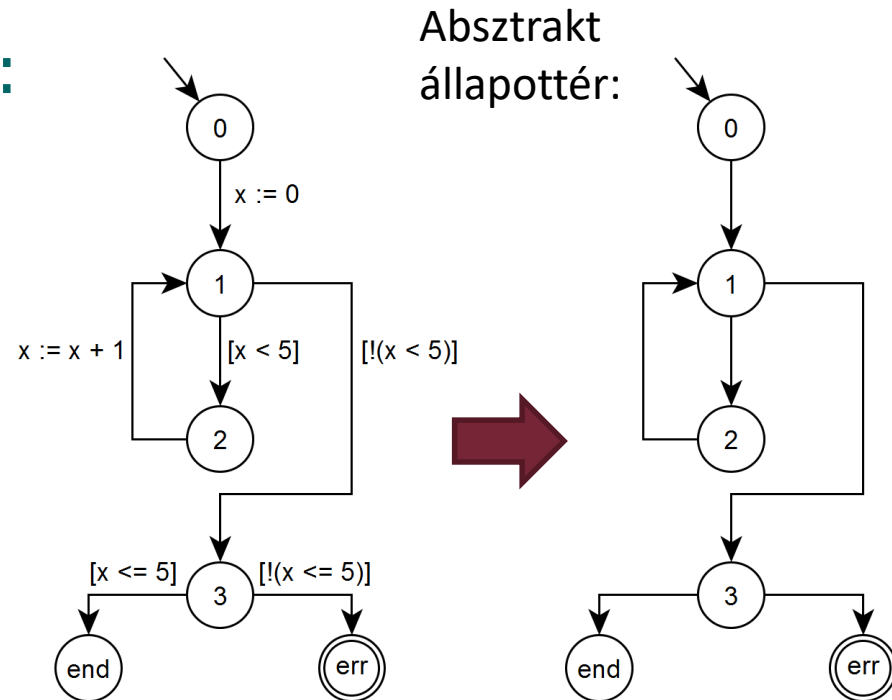
Ellenpélda-alapú absztrakció finomítás

# CEGAR – Bevezető áttekintés



# Absztrakció – Bevezető

- Mi az absztrakció?
  - Általános matematikai eszköz
  - Cél: Részletek elrejtése, fontos információ megtartása
  - Eredmény: Egyszerűbb probléma
- Példa: **Vezérlési hely absztrakció**
  - Az absztrakt modell képzése:
    - $(l, x_1, x_2, \dots, x_n) \rightarrow (l)$
  - Önmagában általában nem használható
    - Triviálisan elérhető a hibaállapot
  - Kiegészítés célszerű, például **predikátumabsztrakcióval**



# Predikátumabsztrakció

- Mi a predikátumabsztrakció?
  - Változók konkrét értékei helyett a változókon értelmezett predikátumok nyilvántartása minden vezérlési helyhez
  - Absztrakt állapot: adott vezérlési helyhez tartozó konkrét állapotok, amelyekre ugyanazok a predikátumok teljesülnek
  - Példa: 3x3 konkrét állapot, 3 predikátum  $\rightarrow$  5 absztrakt állapot
- Absztrakció kiszámítása: Első lehetőség
  - Konkrét állapotok összegyűjtése és összevonása
    - A konkrét állapotok összegyűjtése: Állapottér robbanás ☹

Változók:

$$x, y; D_x = D_y = \{0, 1, 2\}$$

Predikátumok:

$$(x = y), (x < y), (y = 2)$$



$y \backslash x$	0	1	2
0	$(x = y)$	-	-
1	$(x < y)$	$(x = y)$	-
2	$(x < y)$ $(y = 2)$	$(x < y)$ $(y = 2)$	$(x = y)$ $(y = 2)$

# Predikátumabsztrakció számítása

- Absztrakció kiszámítása: Másik lehetőség

- Csak az absztrakt állapotok felsorolása (mi lehetséges)
- $P$  predikátumhalmaz:  $|L| \cdot 2^{|P|}$  lehetséges absztrakt állapot
- Kiszűrve azokat, amik nem fordulhatnak elő

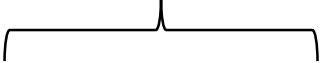
- Példa

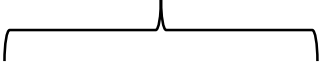
- 3 predikátum  $\rightarrow$  8 lehetséges absztrakt állapot vezérlési helyenként
- Ténylegesen nem mind fordulhat elő
  - Pl. együtt nem fordulhat elő az előbbi példa szerinti  $x, y$ -ra:  
 $(x = y) \wedge (x < y) \wedge \neg(y = 2)$
  - SMT megoldóval kiszűrhető

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

# Predikátumabsztrakció rögzítése

- Absztrakt állapotok rögzítése Boole változókkal

- Konkrét  


Absztrakt  

- $(l, x_1, \dots, x_n) \rightarrow (l, b_1, \dots, b_m)$
  - $b_i$  Boole változó:  $i$ . predikátum teljesül vagy nem
  - Jelölés:  $p(b_i) = \begin{cases} p_i & \text{predikátum, ha } b_i \text{ igaz} \\ \neg p_i & \text{negált predikátum egyébként} \end{cases}$

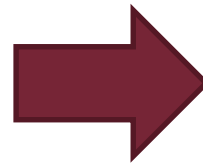
- Példa:

Változók:

$$x, y; D_x = D_y = \{0, 1, 2\}$$

Predikátumok:

$$(x = y), (x < y), (y = 2)$$



$l \quad x \quad y$		$l \quad (x = y) \quad (x < y) \quad (y = 2)$
$\downarrow \quad \downarrow \quad \downarrow$		
$(0, 0, 0)$	$\rightarrow$	$(0, T, F, F)$
$(6, 1, 2)$	$\rightarrow$	$(6, F, T, T)$

# Predikátumabsztrakció elemei

- Absztrakt kezdőállapot, hibaállapot

- Absztrakt kezdőállapot:  $(l_0, b_1, \dots, b_m)$
- Absztrakt hibaállapot:  $(l_E, b_1, \dots, b_m)$

- Absztrakt átmenetek

- Absztrakt átmenet: Létezik, ha lehetséges konkrét átmenet a tartalmazott konkrét állapotok között (azaz felülbecslő tulajdonságú az absztrakció)
- SMT megoldóval számítható a konkrét állapotok ismerete nélkül
- Tehát  $(l, b_1, \dots, b_m)$  és  $(l', b'_1, \dots, b'_m)$  között létezik átmenet, ha:
  - $\exists op: (l, op, l') \in G$ ,  
azaz van  $op$  művelettel él a két vezérlési hely között a CFA-ban, és
  - $p(b_1) \wedge \dots \wedge p(b_m) \wedge op \wedge p(b'_1) \wedge \dots \wedge p(b'_m)$  kielégíthető

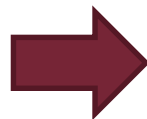
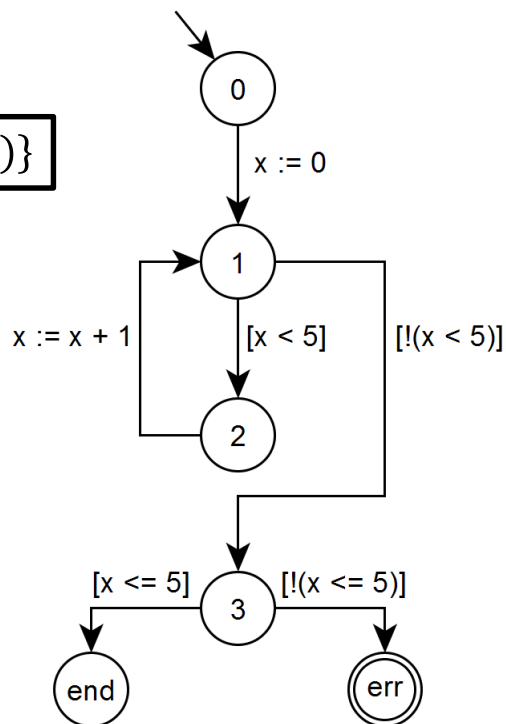
$\underbrace{\hspace{10em}}$   
Kiindulási hely predikátumai

$\underbrace{\hspace{10em}}$   
Cél hely predikátumai

# Predikátumabsztrakció

- Példa

$$P = \{(x \leq 5)\}$$



0, true	0, false
1, true	1, false
2, true	2, false
3, true	3, false
err, true	err, false
end, true	end, false

Vezérlési hely és  
predikátum

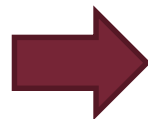
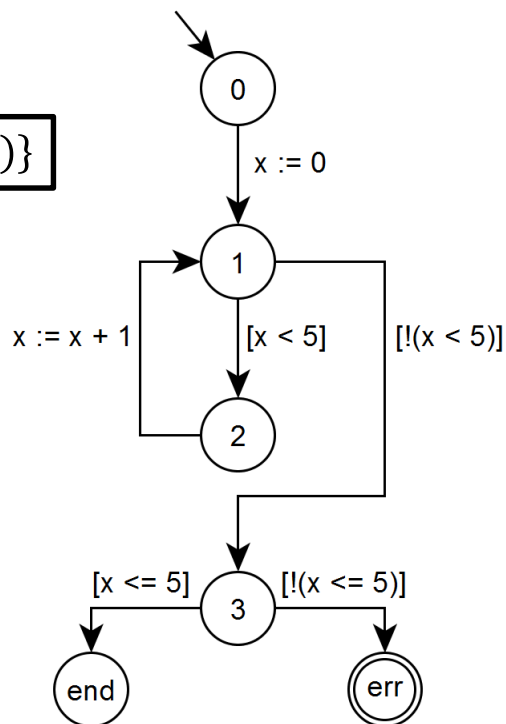
- 6 vezérlési hely, 1 predikátum:  $6 \cdot 2^1 = 12$  absztrakt állapot



# Predikátumabsztrakció

## • Példa

$$P = \{(x \leq 5)\}$$



0, true	0, false
1, true	1, false
2, true	2, false
3, true	3, false
err, true	err, false
end, true	end, false

## • Átmenet létezésének számítására példák

○  $(2, \text{true}) \rightarrow (1, \text{true})$

$(2, x := x + 1, 1) \in G$  és erre  $(x \leq 5) \wedge (x' = x + 1) \wedge (x' \leq 5)$  kielégíthető, pl.  $x = 0, x' = 1$

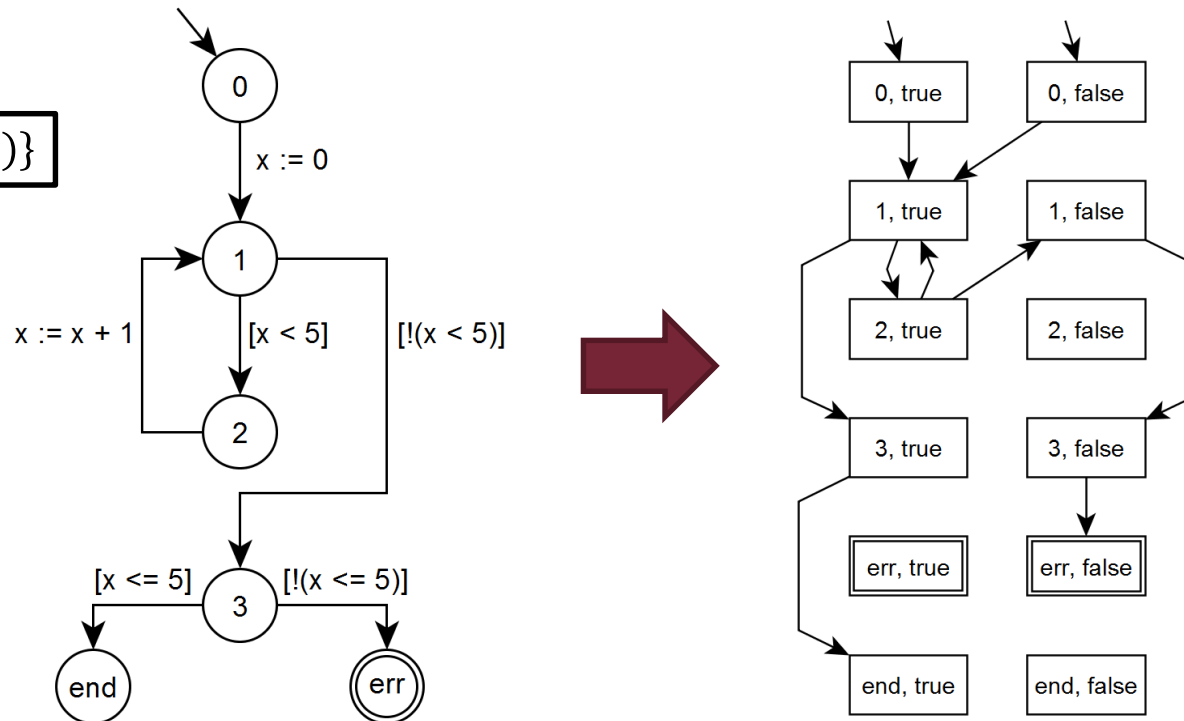
○  $(2, \text{true}) \rightarrow (1, \text{false})$

$(2, x := x + 1, 1) \in G$  és erre  $(x \leq 5) \wedge (x' = x + 1) \wedge \neg(x' \leq 5)$  kielégíthető, pl.  $x = 5, x' = 6$

# Predikátumabsztrakció

- Példa

$$P = \{(x \leq 5)\}$$



- Az összes átmenet létezése hasonló módon ellenőrizhető
  - Lokálisan, csak az absztrakt kiindulási és célállapot alapján
  - Vezérlési helyek közötti él szükséges feltétel
  - Létezik él, ha az SMT megoldó szerint a kiindulási és célállapot predikátumait lehetséges teljesíteni az él őrfeltétele és akciója alapján

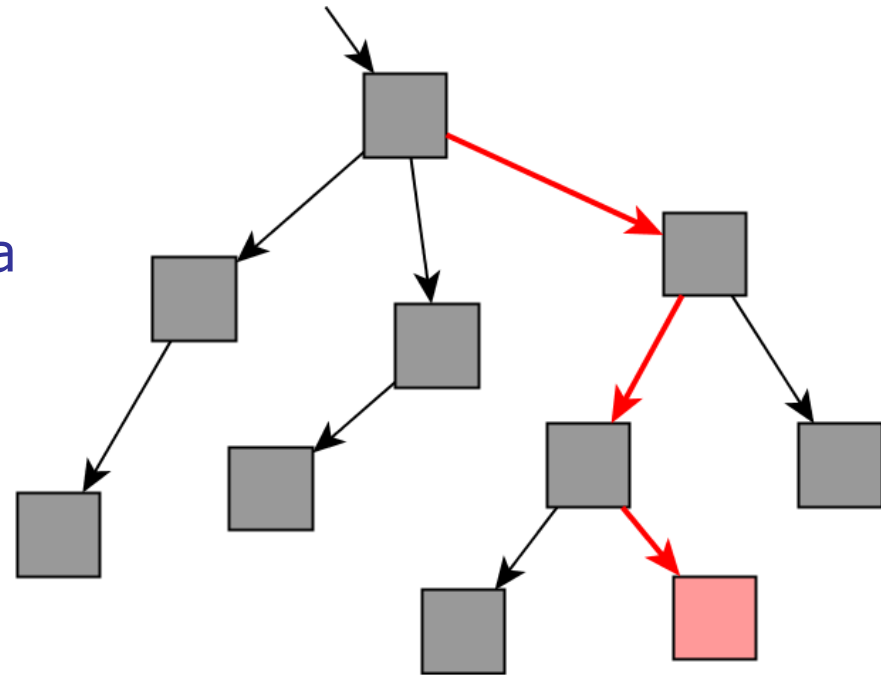
# Modellellenőrzés az absztrakt állapottérben

- Absztrakt állapottér bejárása

- Valamilyen keresési stratégiával, pl. DFS, BFS, szimbolikus
- Hibaállapot keresése

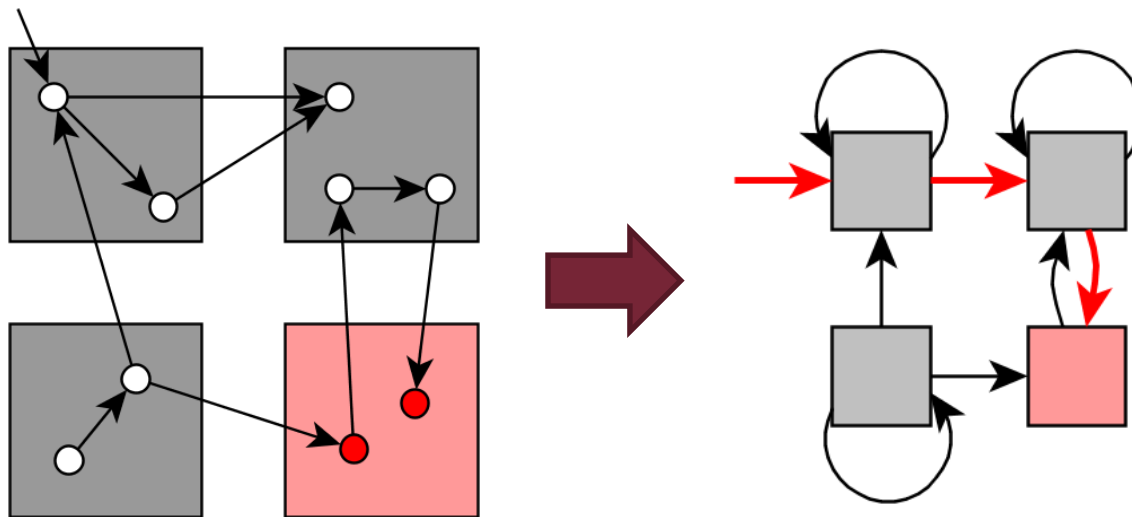
- Optimalizációk

- „On-the-fly”
  - Absztrakt állapotok kiszámítása keresés közben
- Inkrementális
  - A változatlan részeket nem kell újra bejárni, ha erre újra szükség lesz



# Jellemzők az absztrakt állapottérben

- Az absztrakció tulajdonságai
  - Felülbecsli az eredeti modell viselkedéseit (egzisztenciális)
    - Minden konkrét útvonalhoz van megfelelő absztrakt útvonal
    - Ha **nincs** absztrakt útvonal a hibaállapothoz: **nincs** konkrét útvonal sem
    - Ha **van** absztrakt útvonal a hibaállapothoz: **nem biztos**, hogy van konkrét útvonal is
  - Tehát: Az absztrakt ellenpéldát ellenőrizni kell
    - Van-e ennek megfelelő konkrét útvonal?

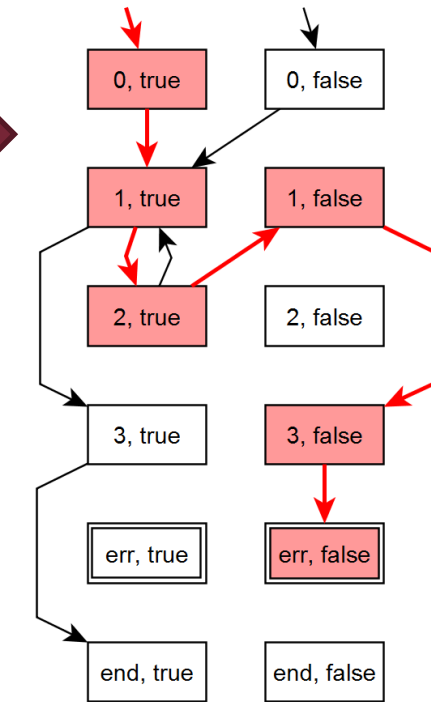
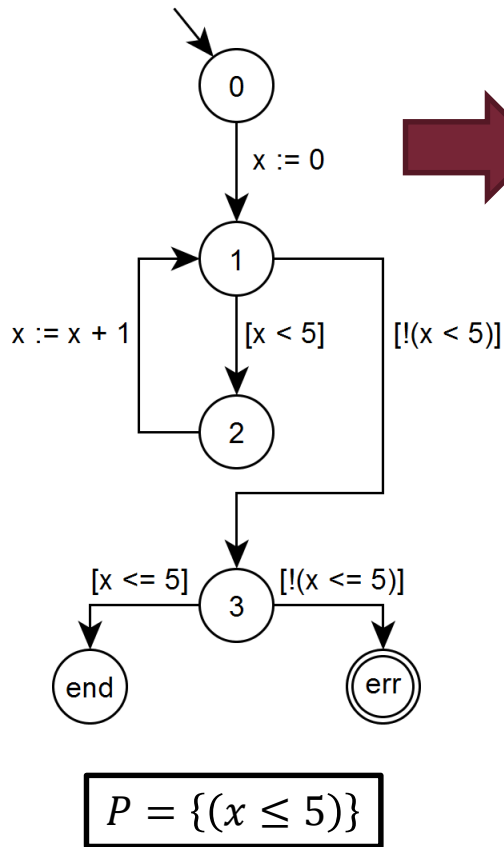


# Absztrakt ellenpélda

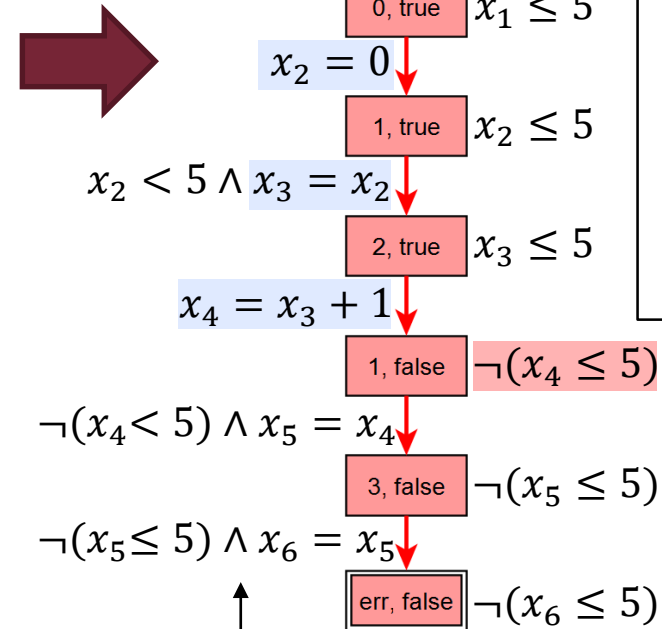
- Absztrakt ellenpélda alakja
  - Vezérlési helyek és predikátumok sorozata
$$(l_1, b_{1,1}, \dots, b_{1,m}), (l_2, b_{2,1}, \dots, b_{2,m}), \dots, (l_n, b_{n,1}, \dots, b_{n,m})$$
- Konkrét útvonal keresése: konkrét állapottér egy részének bejárása
  - Absztrakt ellenpélda által vezérelve (mit kell bejárni)
  - SMT megoldó segítségével
    - Korlátos modellellenőrzéshez hasonlóan
    - Predikátumabsztrakciónál egy átmenet létezésének számítására bemutatott módszer általánosítása  $n$  lépésre
- Ha létezik konkrét útvonal  $\rightarrow$  konkrét modell is hibás
- Ha nem létezik konkrét útvonal  $\rightarrow$  hamis ellenpélda

# Absztrakt ellenpélda

## • Példa



Absztrakt ellenpélda  
(6 állapot)



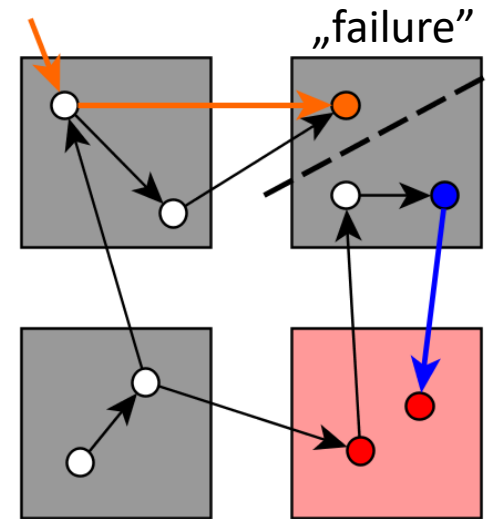
Műveletek az  
átmeneteken

Állapotok  
predikátumai

$(x_1 = int)$   
 $(x_2 = 0)$   
 $(x_3 = 0)$   
 $(x_4 = 1)$   
**Predikátum  
nem lesz  
kielégíthető**

# Hamis ellenpélda elemei

- „Failure” állapot: Eddig az absztrakt állapotig van út, és onnan tovább is, de ezek a konkrét modellben elkülönülő utak
- Konkrét állapotok csoportosítása a „failure” absztrakt állapotban
  - D = “Dead-end”: elérhető
  - B = “Bad”: következő állapotra lép
  - IR = “Irrelevant”: többi
- Hamis ellenpélda oka
  - Predikátumhalmaz nem különbözteti meg D-t és B-t



# Absztrakciófinomítás

- Cél: Hamis ellenpélda kiküszöbölése

- Bővebb predikátumhalmaz (finomabb absztrakció)

- **D** és **B** szétválasztása

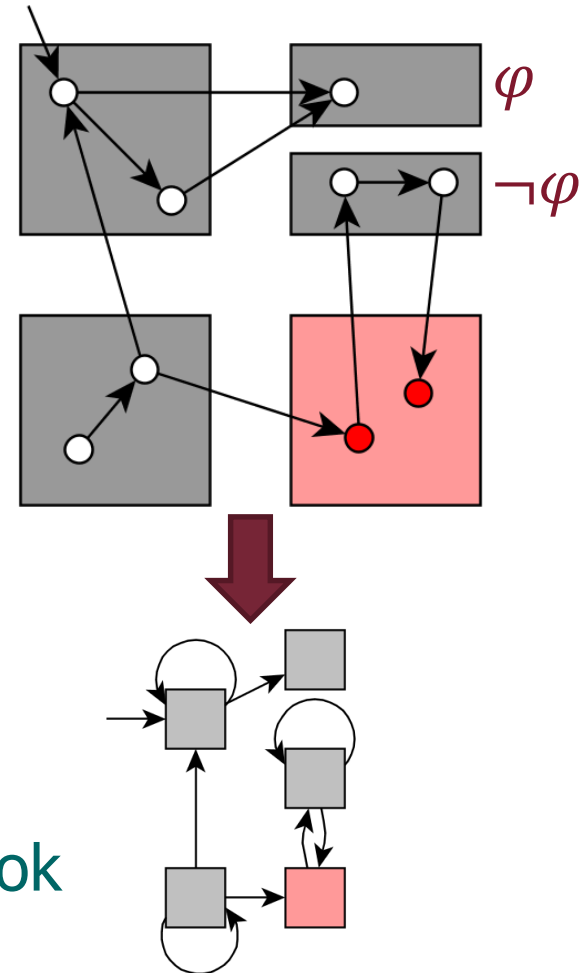
- Konkrét állapotok felsorolása nélkül kell
- **D** és **B** leírhatók formulákkal az útvonalon
- SMT megoldó képes egy  $\varphi$  formulát generálni, ami szétválasztja ezeket (egyiken igaz, a másikon nem), ez az ún. interpoláció

- $P \cup \{\varphi\}$  predikátumhalmaz esetén ez a hamis ellenpélda megszűnik

- Sőt,  $\varphi$  predikátumot elég csak a „failure” állapotban alkalmazni („lusta” absztrakció)

- További hamis ellenpéldák

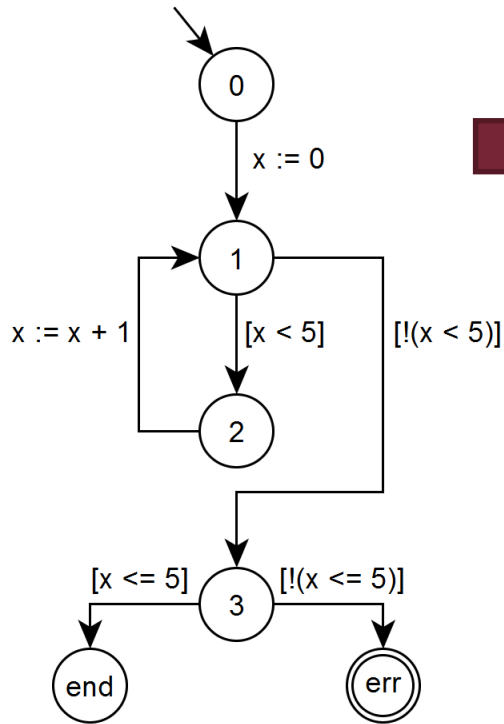
- További finomítások, újabb predikátumok





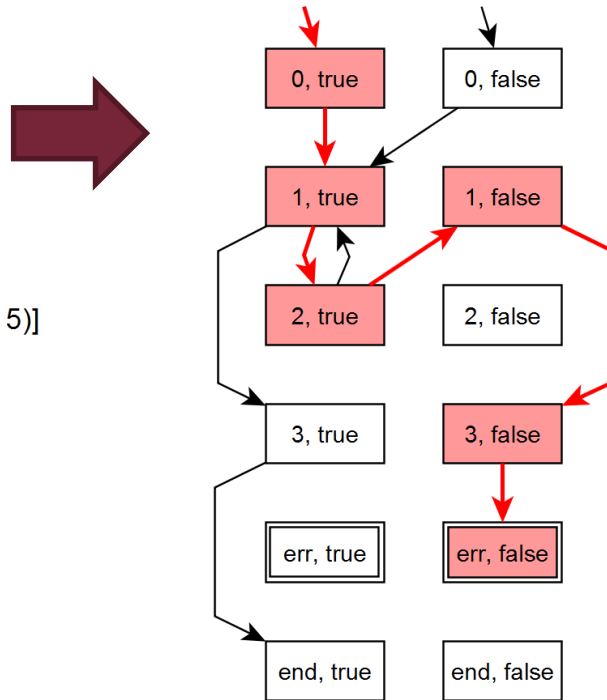
# Absztrakciófinomítás

## • Példa

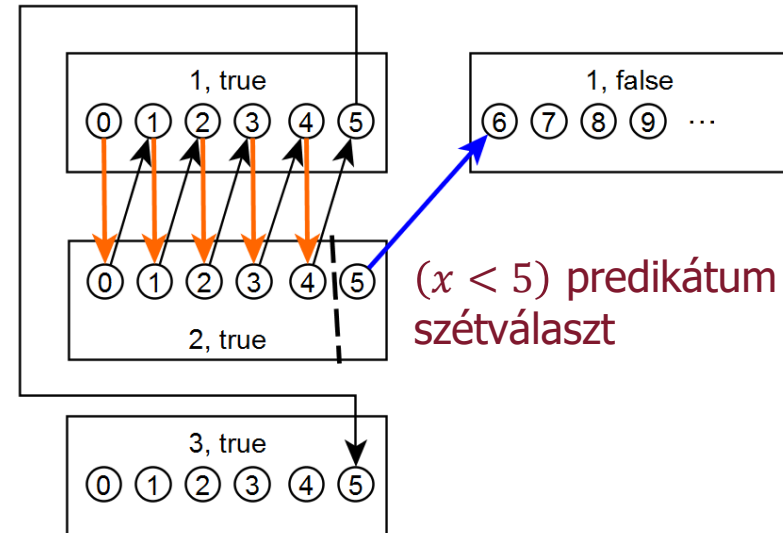


$P = \{(x \leq 5)\}$

Szétválasztás:  $(x < 5)$  predikátum

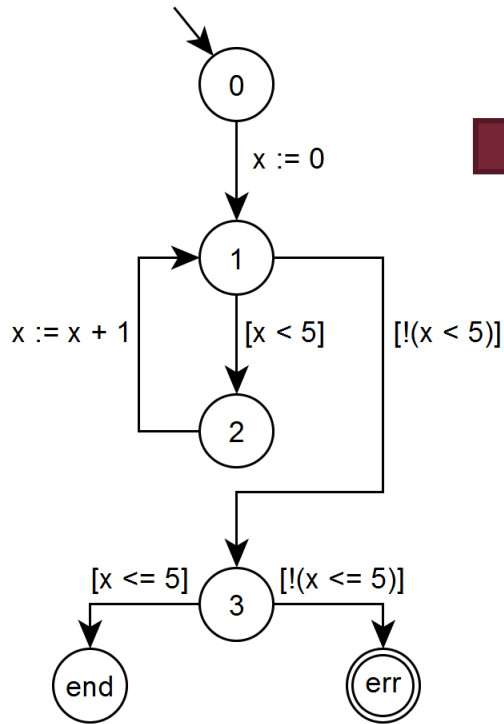


$(x_1 = int)$   
 $(x_2 = 0)$   
 $(x_3 = 0)$   
 $(x_4 = ?)$



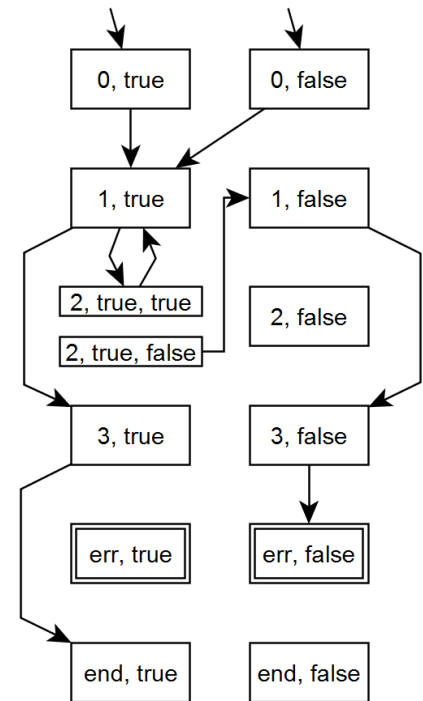
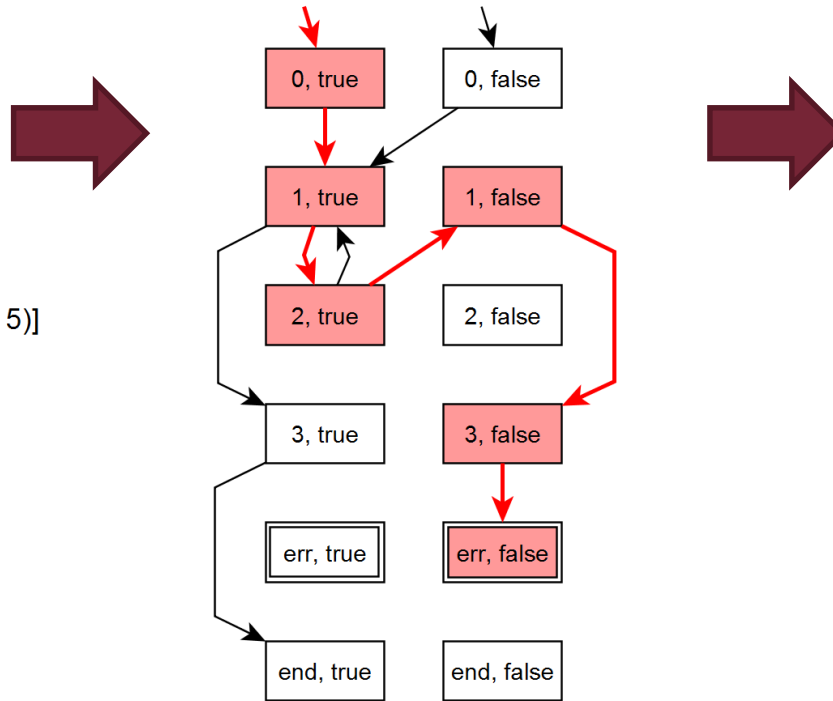
# Absztrakciófinomítás

## • Példa



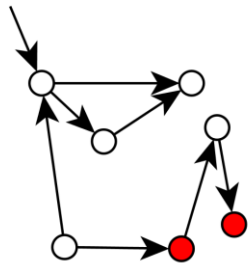
$P = \{(x \leq 5)\}$

Szétválasztás:  $(x < 5)$  predikátum

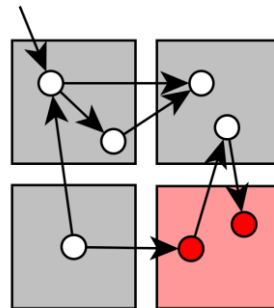


$P = \{(x \leq 5), (x < 5)\}$

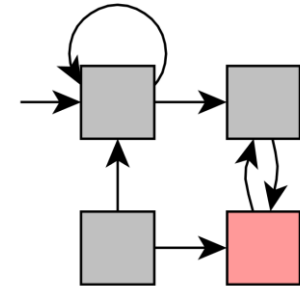
# CEGAR – Összefoglaló



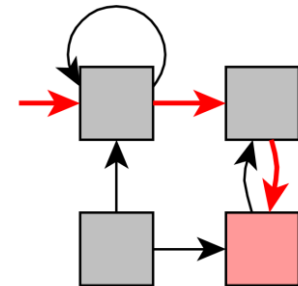
Konkrét állapottér



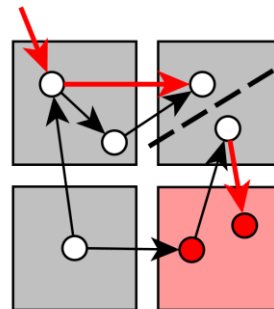
Absztrakció



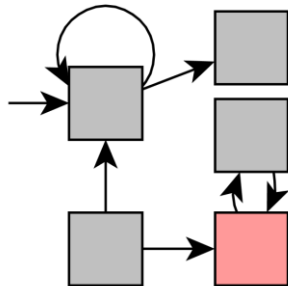
Absztrakt állapottér



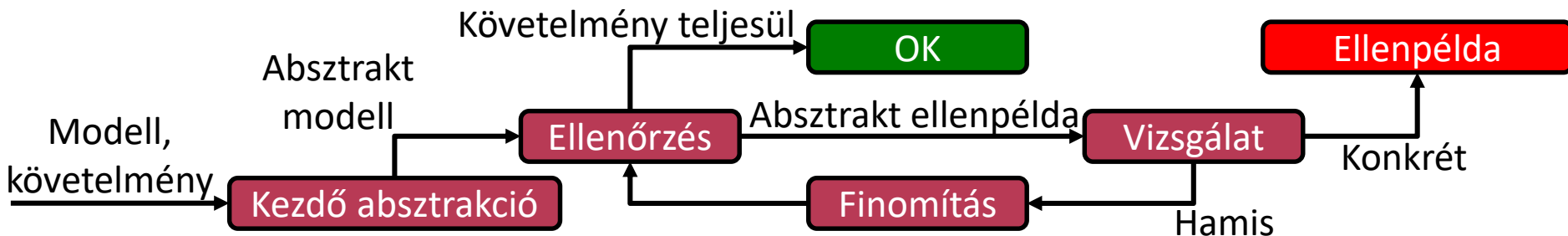
Absztrakt ellenpélda



Hamis ellenpélda

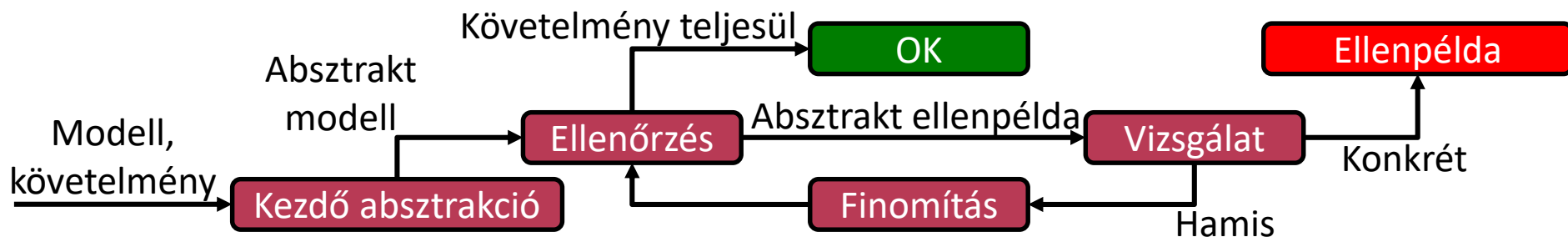


Finomított absztrakció



# A CEGAR algoritmusról

- Counterexample-Guided Abstraction Refinement (CEGAR)
  - Automatikus módszer
    - Minden lépés automatikusan működik
    - Nem szükséges a belső működés részletes ismerete
  - Ki mondja meg a kezdeti predikátumhalmazt?
    - Lehet akár üres halmaz is (a CEGAR algoritmus fogja az absztrakt állapotokat szétválasztani)
    - Programban szereplő feltételes utasítások alapján
    - Egyéb heurisztikák alapján



# Eszközök

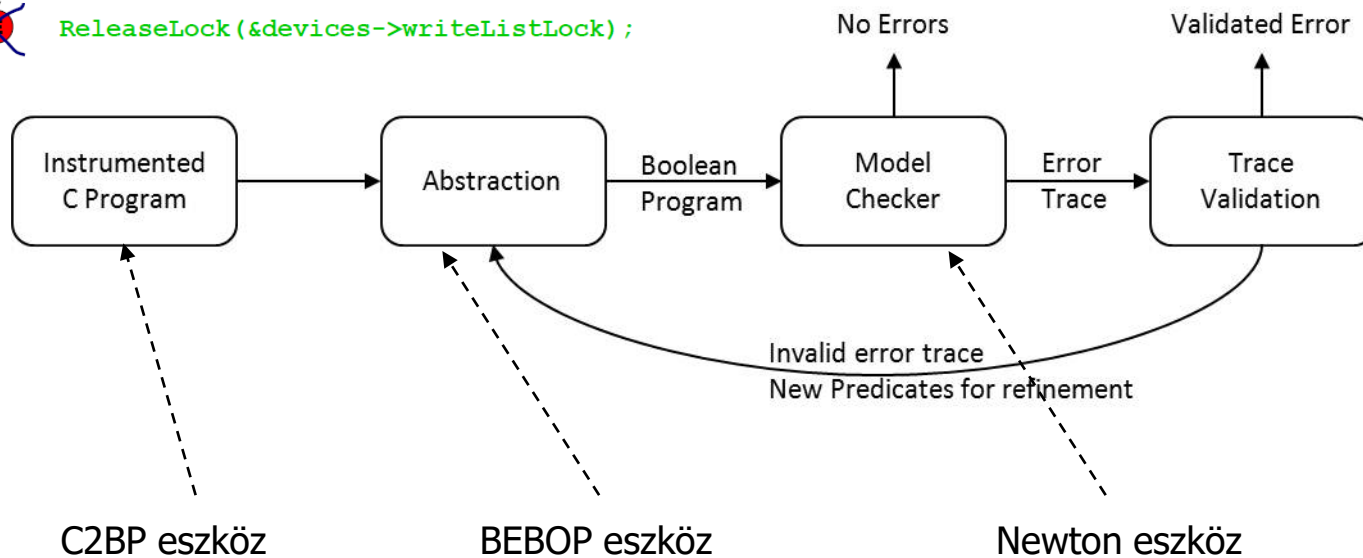
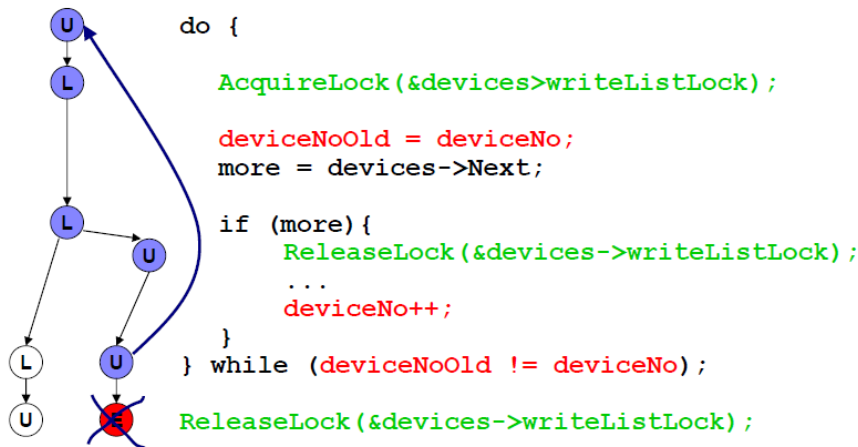
# A SLAM2 eszköz

- SLAM2

- Static Driver Verifier Research Platform (SDVRP) része
- Célkitűzések
  - Driver **C kód**: Vizsgált komponens
  - Platform modell: Környezet leírása
  - Ellenőrzés: **API használati szabályok** betartása
- Működése
  - Boole program előállítása **predikátumabsztrakcióval**
  - Szimbolikus modellellenőrzés
  - **CEGAR** ciklus
- [research.microsoft.com/en-us/projects/slam/](https://research.microsoft.com/en-us/projects/slam/)

# SLAM2 architektúra

- Static Driver Verifier Research Platform (SDVRP)



# A BLAST eszköz

- BLAST

- Berkeley Lazy Abstraction Software Verification Tool
- Bemenet: C program + követelmény (BLAST Query Language)
- Predikátumabsztrakció
  - Absztrakt elérhetőségi fa építése
- Finomítás: új predikátum(ok) interpolációval
  - „Lusta” absztrakció: új predikátum alkalmazása lokálisan
- Korlátok: szorzás, bitműveletek, túlcsordulás
- [mtc.epfl.ch/software-tools/blast/index-epfl.php](http://mtc.epfl.ch/software-tools/blast/index-epfl.php)



# A CPAchecker eszköz

- CPAchecker

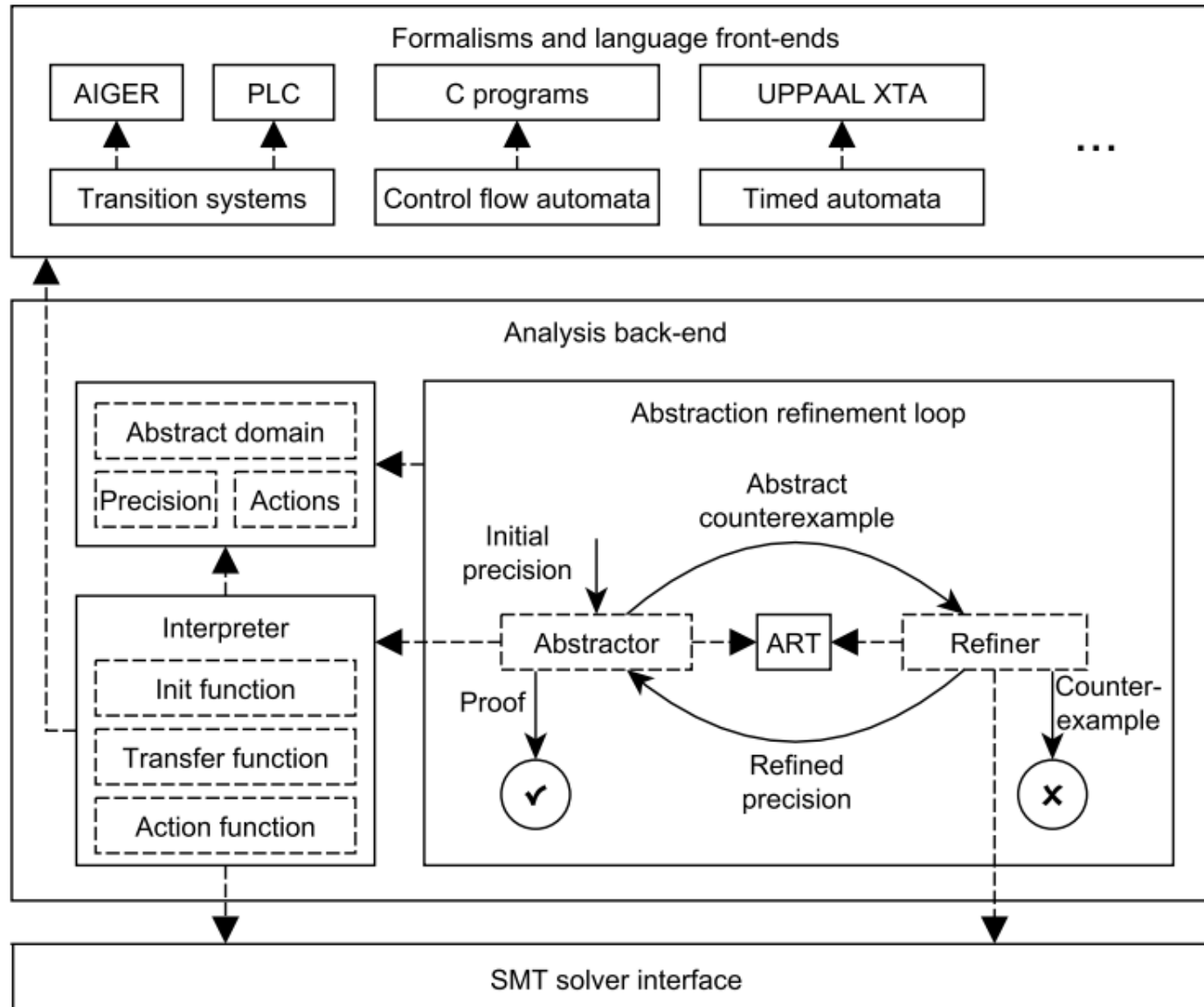
- The Configurable Software-Verification Platform
- Bemenet: C program + ellenőrzés specifikációja
  - Ellenőrzések: Assertion, error címke, deadlock, null dereference, ...
- Nagymértékben konfigurálható
  - Többféle absztrakció (nem csak predikátumok)
  - Absztrakt ellenpélda több prefixét tekinti
    - Többféle lehetséges finomítás közül választ (finomítási stratégia)
- [cpachecker.sosy-lab.org/](http://cpachecker.sosy-lab.org/)

# A Theta eszköz

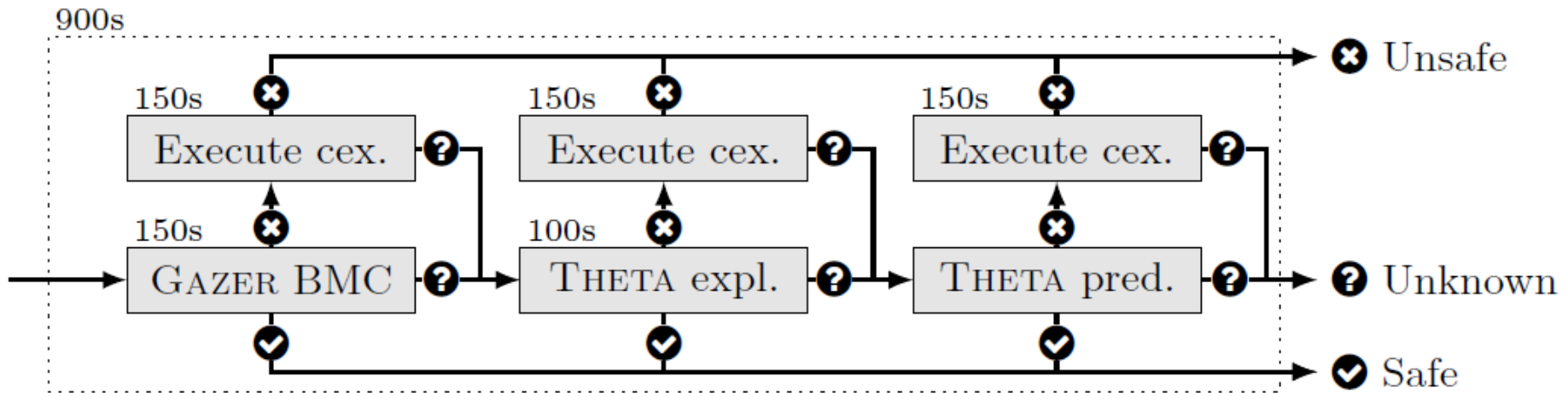
- Theta

- Általános, moduláris, konfigurálható verifikációs keretrendszer
- BME MIT saját fejlesztés
- **Általános**: különböző formális modellek támogatása
  - Tranzíciós rendszer, control flow automata, időzített automata
  - Szoftverek: CFA alapú reprezentáció
- **Moduláris**: újrafelhasználható, kombinálható modulok
  - Absztrakciók, finomítások, interpoláció, ...
- **Konfigurálható**: különböző algoritmusok és stratégiák
- **Komponálható**: (újra)próbálás más konfigurációkkal
- [github.com/FTSRG/theta](https://github.com/FTSRG/theta)

# Theta architektúra



# Theta portfólió predikátum absztrakcióhoz



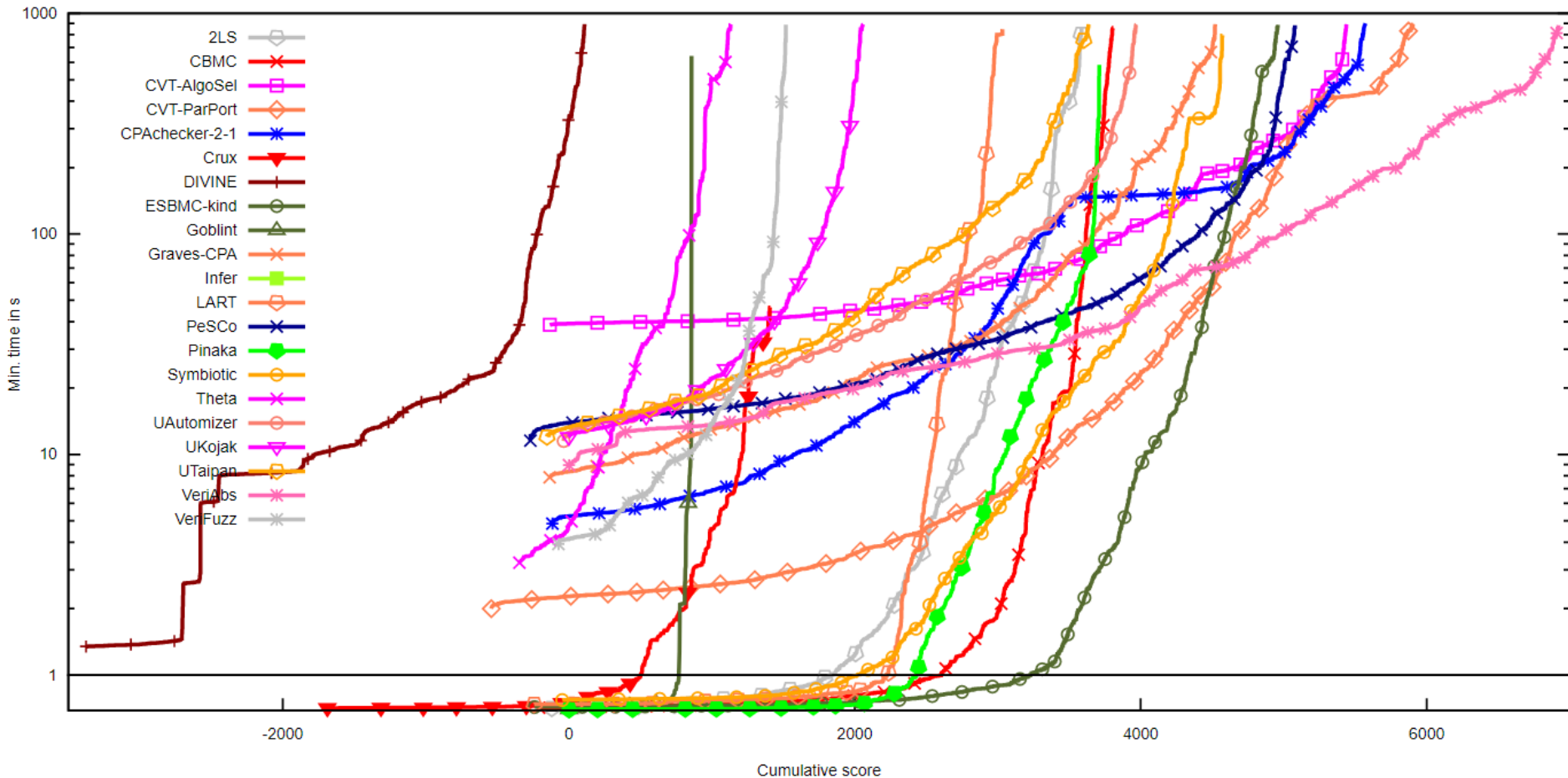
- A portfólió elemei:
  - Gazer BMC: Korlátos modellellenőrzés
  - Theta expl: explicit változó analízis
  - Theta pred: predikátum absztrakció
- Ellenpélda (cex) ellenőrzés végrehajtással
- Időtúllépés esetén továbblépés

# SV-COMP verseny

- **Competition on Software Verification (SV-COMP)**
  - [sv-comp.sosy-lab.org/](http://sv-comp.sosy-lab.org/)
  - ~30 eszköz, ~15.000 verifikációs feladat (program + követelmény)
  - Program kategóriák
    - Arrays (ArraysReach, ArraysMemSafety)
    - Bit Vectors (BitVectorsReach, Overflows)
    - Heap Data Structures (HeapReach, HeapMemSafety)
    - Floats
    - Integers and Control Flow (ControlFlow, Simple, ECA, Loops, Recursive, ProductLines, Sequentialized)
    - Termination
    - Concurrency
    - Software Systems (DeviceDriversLinux64, BusyBox)
  - Adott kategóriához megtalálható a hatékony eszköz

# Eszközök: SV-COMP 2022 verseny

- Eszközök által gyűjtött pontok (negatív is lehet)



# Összefoglalás

# Összefoglalás

- Szoftver-modellellenőrzés
  - „Gombnyomásos” módszer
    - Kezelendő probléma: Állapottér robbanás (a változók miatt)
  - Megoldás: Absztrakció
    - Vezérlési hely + predikátumok a változókon
    - Felülbecslő absztrakció az útvonalakra (hamis ellenpélda lehet)
  - CEGAR: Megfelelő absztrakció automatikus előállítás
    - 1. Kezdő absztrakt modell
    - 2. Modellellenőrzés
    - 3. Ellenpélda vizsgálata
    - 4. Absztrakció finomítása
  - Eszközök