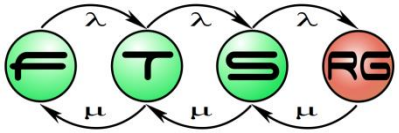
Szoftver-modellellenőrzés absztrakciós módszerekkel

Hajdu Ákos

Szoftver verifikáció és validáció

2016.11.02.



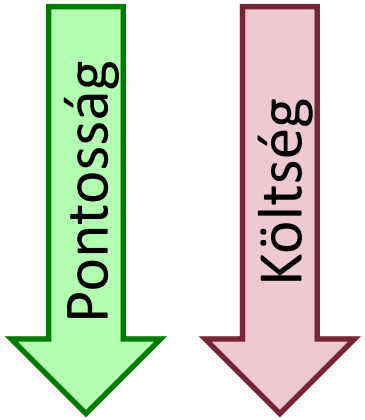
Budapesti Műszaki és Gazdaságtudományi Egyetem
Méréstechnika és Információs Rendszerek Tanszék

BEVEZETŐ

Bevezető

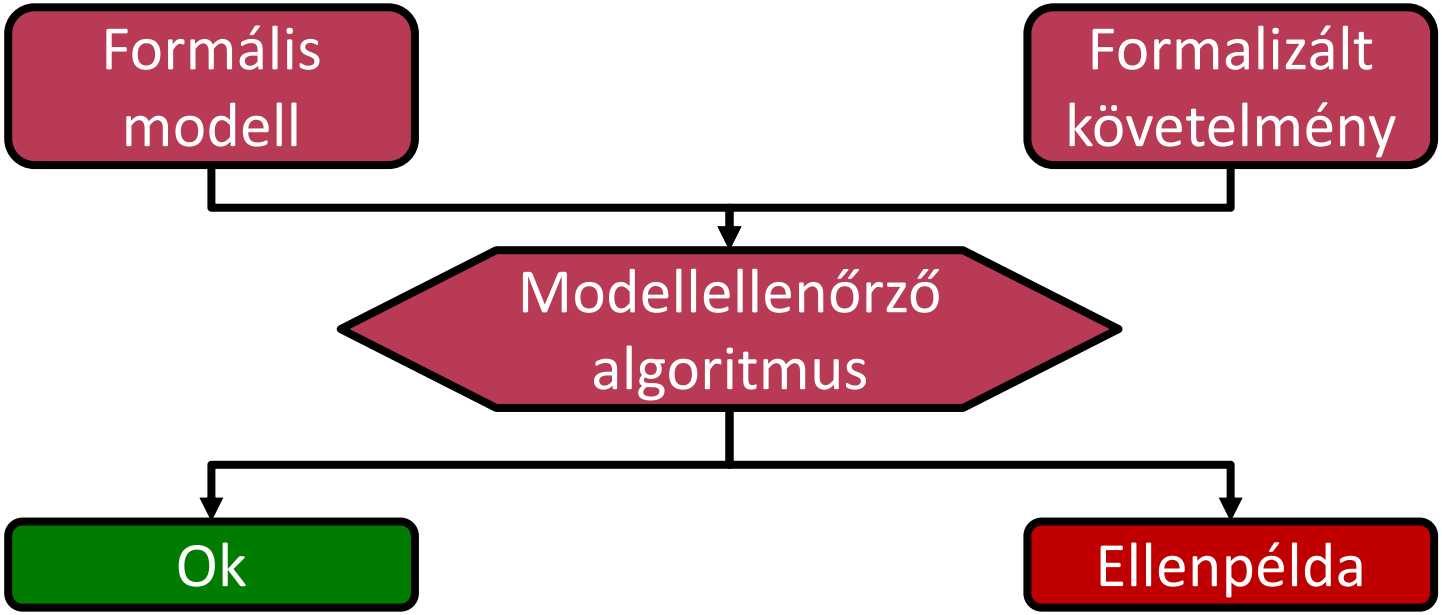
- Motiváció
  - Forráskód közvetlen ellenőrzése
  - „Gombnyomásra” működjön
    - Ne kelljen komoly háttérismeret

- Szoftverellenőrzési technikák
  - Statikus analízis
    - Hibaminta keresők
    - Absztrakt interpretáció
  - Modellellenőrzés



Bevezető

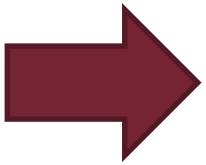
- Modellellenőrzés

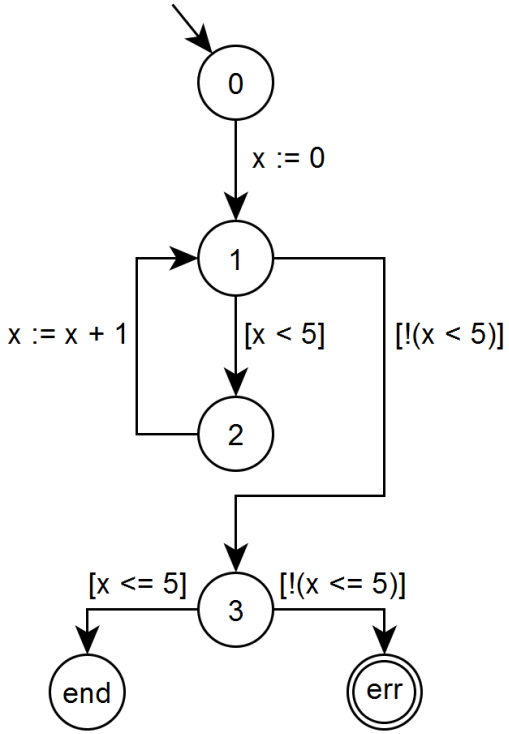


Bevezető

- Control-Flow Automata: $CFA = (X, L, l_0, G)$
  - X: Változók halmaza
  - L: Vezérlési helyek halmaza
  - l_0: Kezdő vezérlési hely
  - $G \subseteq L \times Ops \times L$: Élek halmaza
    - Ops: Műveletek a vezérlés átlépésekor
      - Pl. őrfeltétel, értékadás, …

```
    x : int
0:  x = 0
1:  while (x < 5) {
2:      x = x + 1
    }
3:  assert (x <= 5)
```

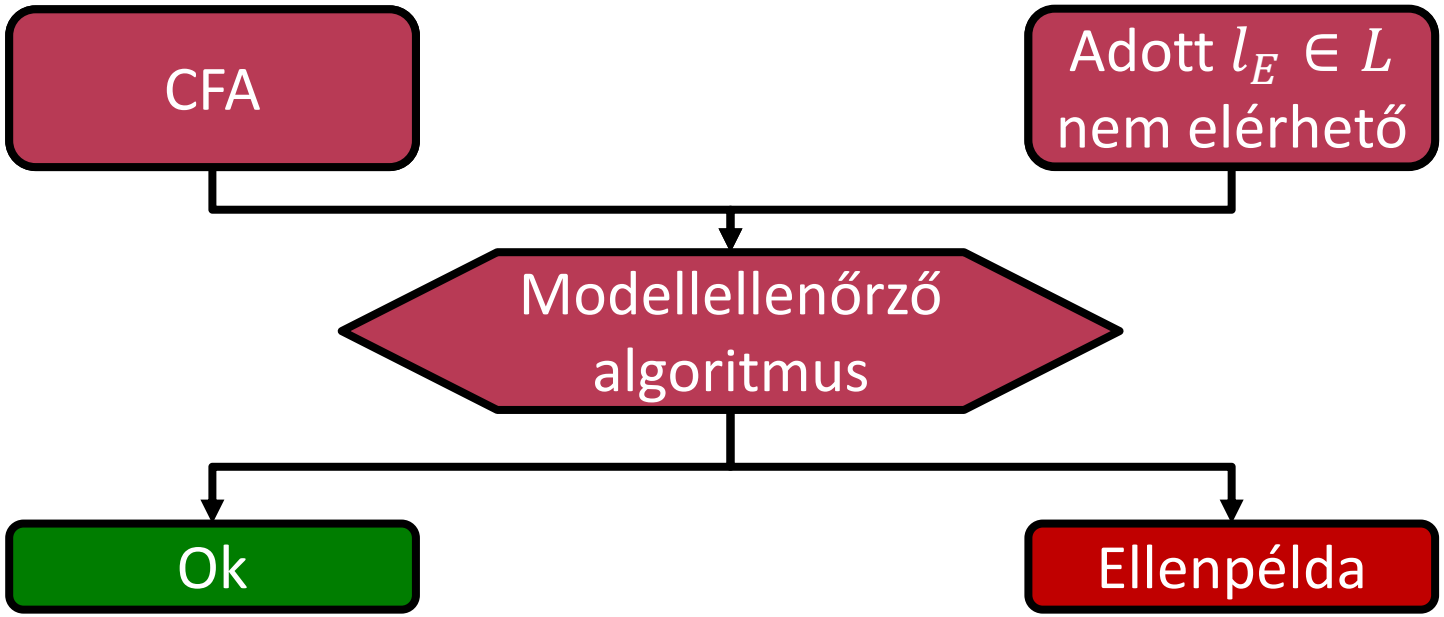




- Tipikus követelmény: „error” hely (l_E) ne legyen elérhető

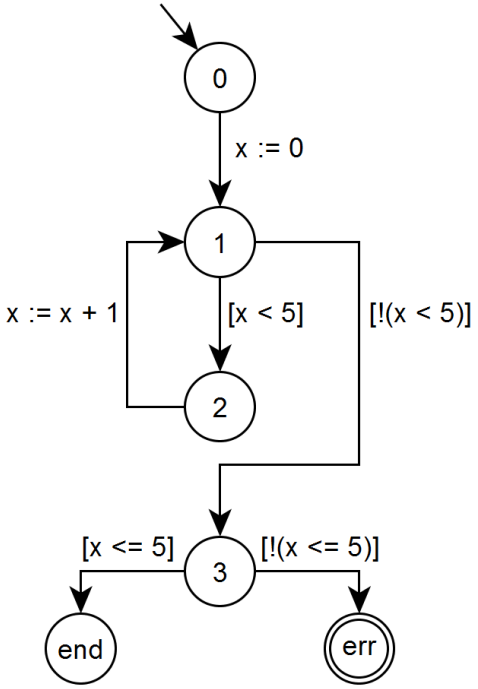
Bevezető

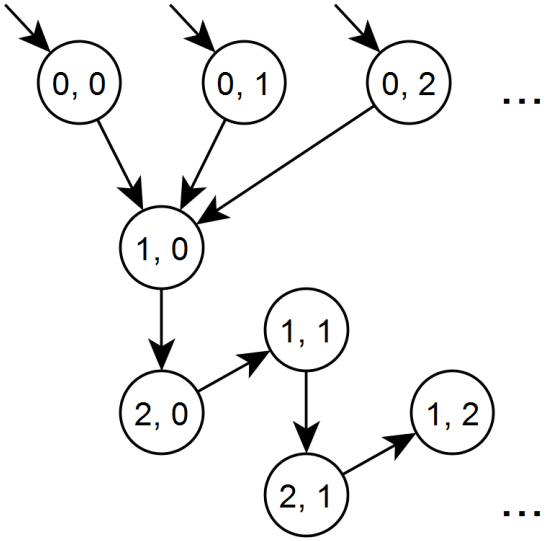
- Modellellenőrzés



Bevezető

- CFA állapottere
  - Állapot: vezérlési hely + változók értéke
    - $(l, x_1, x_2, \dots, x_n) \in L \times D_{x_1} \times D_{x_2} \times \cdots \times D_{x_n}$
    - D_{x_i}: i. változó doménje
  - Probléma: állapottér robbanás





Bevezető

- Propozicionális logika (nulladrendű)
  - Boolean 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

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

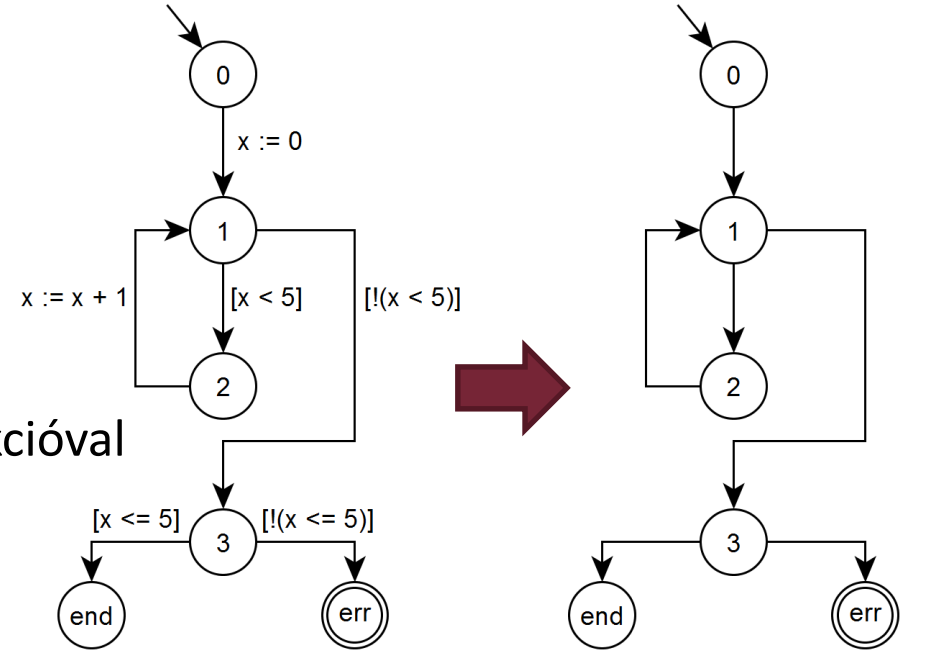
- Satisfiability Modulo Theories (SMT)
  - Elsőrendű logikai formulák
  - Interpretált szimbólumok
    - Pl. egész aritmetika

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

ABSZTRAKCIÓ

Absztrakció bevezető

- Absztrakció
  - Általános matematikai eszköz
  - Részletek elhagyása
  - Könnyebb probléma
- Példa
  - Vezérlési hely absztrakció
  - $(l, x_1, x_2, \ldots, x_n) \rightarrow (l)$
  - Önmagában általában kevés
    - Kiegészítés predikátum-absztrakcióval



Predikátumabsztrakció

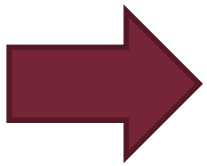
- Predikátumabsztrakció
  - Változók konkrét értékei helyett predikátumok nyilvántartása
  - Absztrakt állapot: adott vezérlési helyhez tartozó konkrét állapotok, amelyekre ugyanazok a predikátumok teljesülnek
- Absztrakció kiszámolása
  - Konkrét állapotok felsorolása és összevonása
  - Példában 3x3 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	$(x < y)$ $(y = 2)$	$(x < y)$ $(y = 2)$	$(x = y)$ $(y = 2)$

Predikátumabsztrakció

- Absztrakció kiszámolása (más módszer)
  - Csak az absztrakt állapotok felsorolása
  - P predikátumhalmaz → $|L| \cdot 2^{|P|}$ lehetséges absztrakt állapot

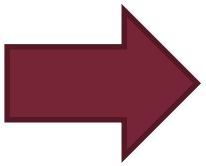
- Példa
  - 3 predikátum → 8 lehetséges absztrakt állapot (vezérlési helyenként)
  - Nem mind lehetséges
    - Szűrés SMT megoldóval
    - Pl. $(x = y) \wedge (x < y) \wedge \neg(y = 2)$

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

Predikátumabsztrakció

- Absztrakt állapotok
  - Konkrét / Absztrakt
  - $(l, x_1, \ldots, x_n) \rightarrow (l, b_1, \ldots, b_m)$
  - b_i: Bool változó: i. predikátum teljesül vagy nem
  - Jelölés: $p(b_i) = \begin{cases} p_i & \text{ha } b_i \text{ igaz} \\ \neg p_i & \text{egyébként} \end{cases}$
- Példa

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



$l\ x\ y$

$(0,0,0) \rightarrow (0, T, F, F)$

$(6,1,2) \rightarrow (6, F, T, T)$

$l\ (x = y)(x < y)(y = 2)$

Predikátumabsztrakció

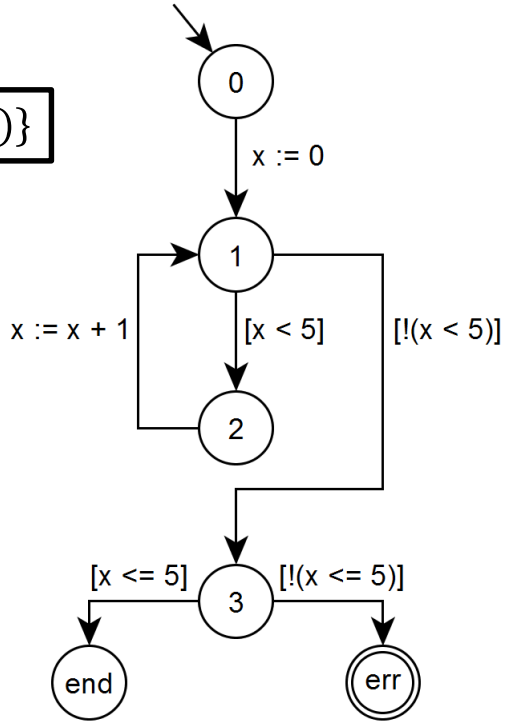
- Absztrakt kezdőállapot, hibaállapot, átmenetek
  - Absztrakt kezdőállapot: $(l, b_1, \ldots, b_m)$, ahol $l = l_0$
  - Absztrakt hibaállapot: $(l, b_1, \ldots, b_m)$, ahol $l = l_E$
  - Absztrakt átmenet: létezik legalább egy konkrét átmenet a tartalmazott konkrét állapotok között
    - Számítás SMT megoldóval (konkrét állapotok felsorolása nélkül)
    - $(l, b_1, \ldots, b_m)$ és $(l', b'_1, \ldots, b'_m)$ esetén:
      - $\exists op$: $(l, op, l') \in G$ (legyen él a két hely között a CFA-ban)
      - $p(b_1) \wedge \cdots \wedge p(b_m) \wedge op \wedge p(b'_1) \wedge \cdots \wedge p(b'_m)$ kielégíthető

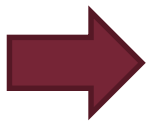
Egzisztenciális absztrakció

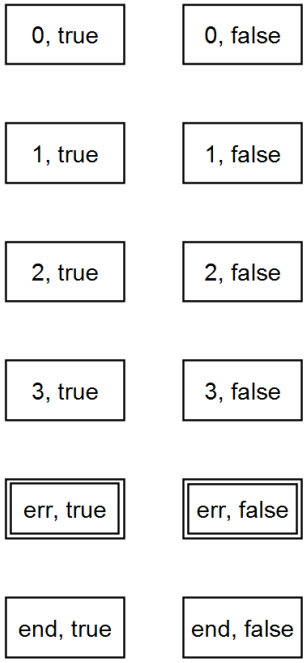
Predikátumabsztrakció

- Példa

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





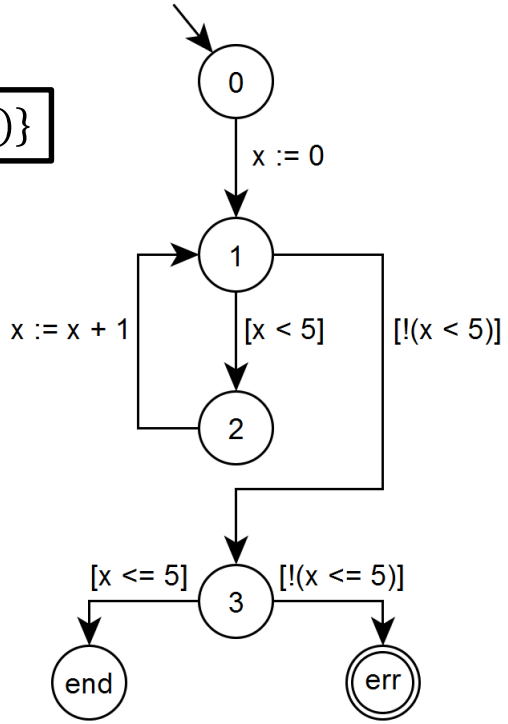


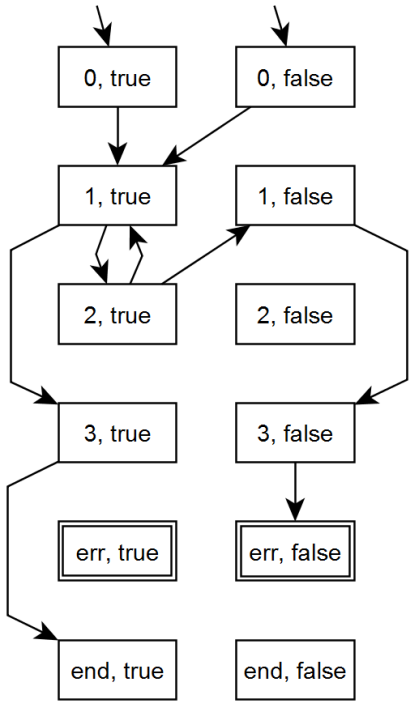
- 6 vezérlési hely, 1 predikátum → 12 absztrakt állapot

Predikátumabsztrakció

- Példa

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

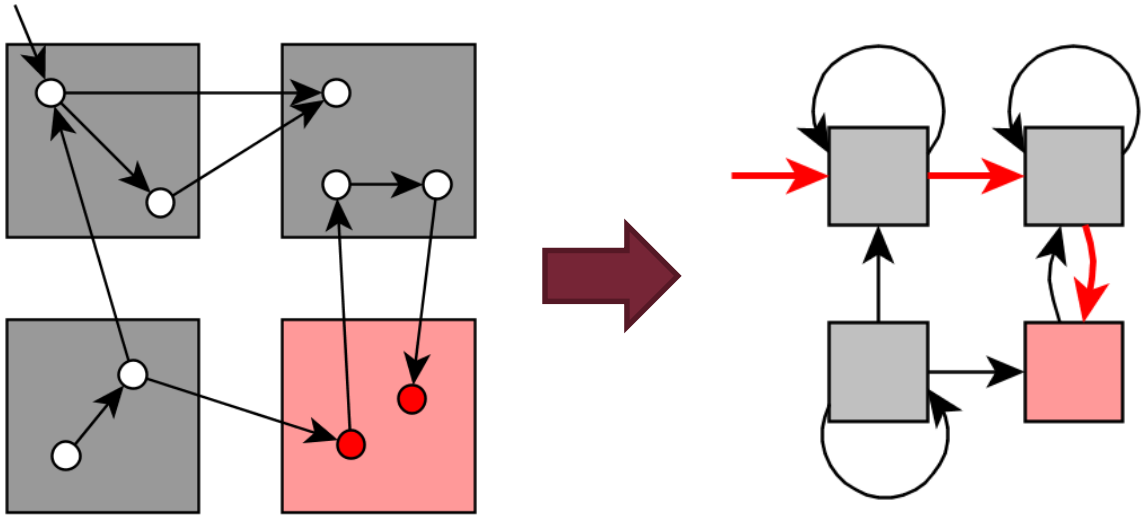




- Átmenet példák
  - $(2, \text{true}) \rightarrow (1, \text{true})$
    - $(2, x := x + 1, 1) \in G$ és $(x \leq 5) \wedge (x' = x + 1) \wedge (x' \leq 5)$ kielégíthető: $x = 0, x' = 1$
  - $(2, \text{true}) \rightarrow (1, \text{false})$
    - $(2, x := x + 1, 1) \in G$ és $(x \leq 5) \wedge (x' = x + 1) \wedge \neg(x' \leq 5)$ kielégíthető: $x = 5, x' = 6$

Egzisztenciális absztrakció

- Egzisztenciális absztrakció tulajdonságai
  - Felülbecsli az eredeti modellt
    - Minden konkrét útvonalhoz van megfelelő absztrakt útvonal
    - Univerzálisan kvantált követelmény teljesül → eredeti modellben is
      - Nem lehet eljutni hibaállapotba (AG ¬Hiba) → eredetiben sem
  - Mi történik absztrakt ellenpélda esetén?
    - Nem minden absztrakt útvonalhoz van megfelelő konkrét útvonal!

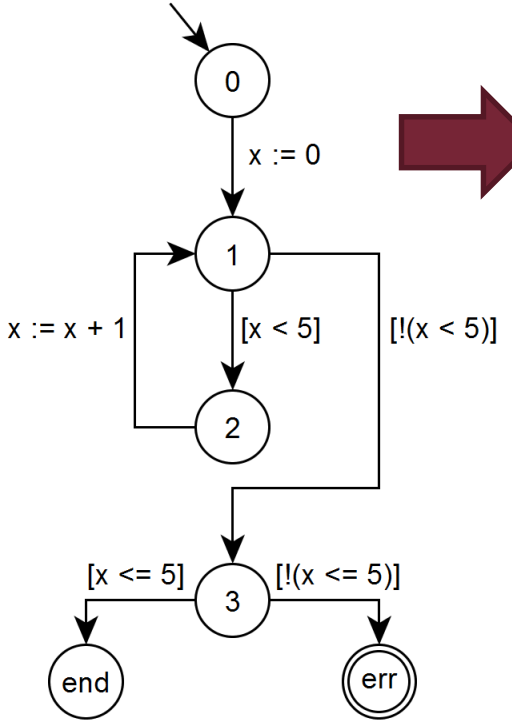


Absztrakt ellenpélda

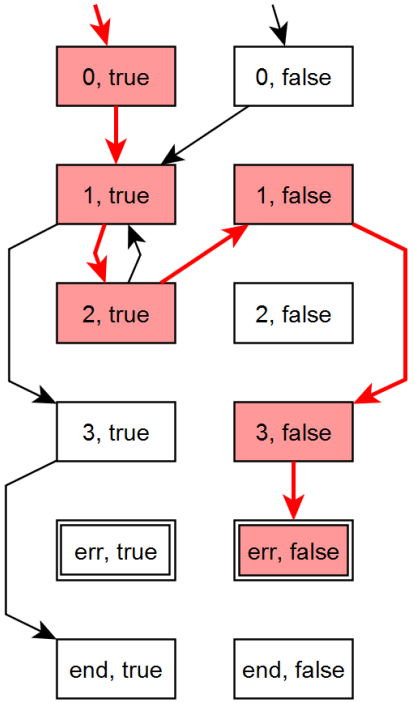
- Absztrakt ellenpélda alakja
  - Vezérlési helyek és predikátumok sorozata
  - $(l_1, b_{1,1}, \dots, b_{1,m}), (l_2, b_{2,1}, \dots, b_{2,m}), \dots, (l_n, b_{n,1}, \dots, b_{n,m})$
- Konkrét útvonal keresése → konkrét állapottér egy részének bejárása
  - Absztrakt ellenpélda által vezérelve
  - SMT megoldó segítségével
    - Korlátos modellellenőrzéshez hasonlóan
    - Egzisztenciális absztrakciónál egy átmenetre bemutatott módszer általánosítása n lépésre
- Ha van konkrét útvonal → konkrét modell is hibás
- Ha nincs konkrét útvonal → hamis ellenpélda

Absztrakt ellenpélda

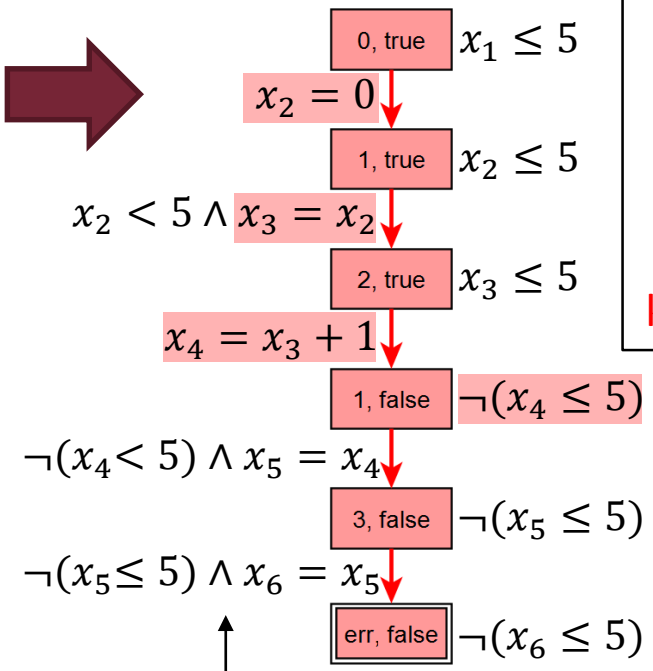
- Példa



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



Absztrakt ellenpélda (6 állapot)



Műveletek az átmeneteken

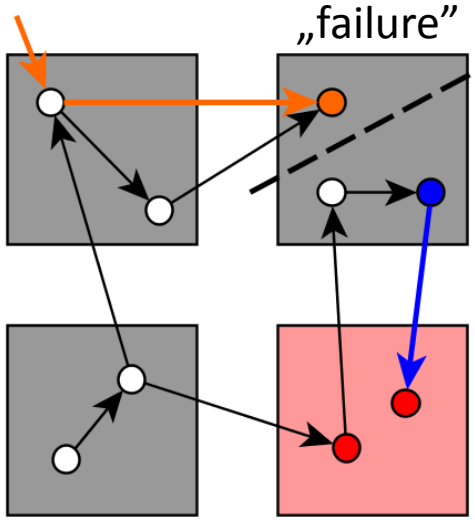
Állapotok predikátumai

$(x_1 = 3)$
$(x_2 = 0)$
$(x_3 = 0)$
$(x_4 = ?)$

Nem kielégíthető

Hamis ellenpélda

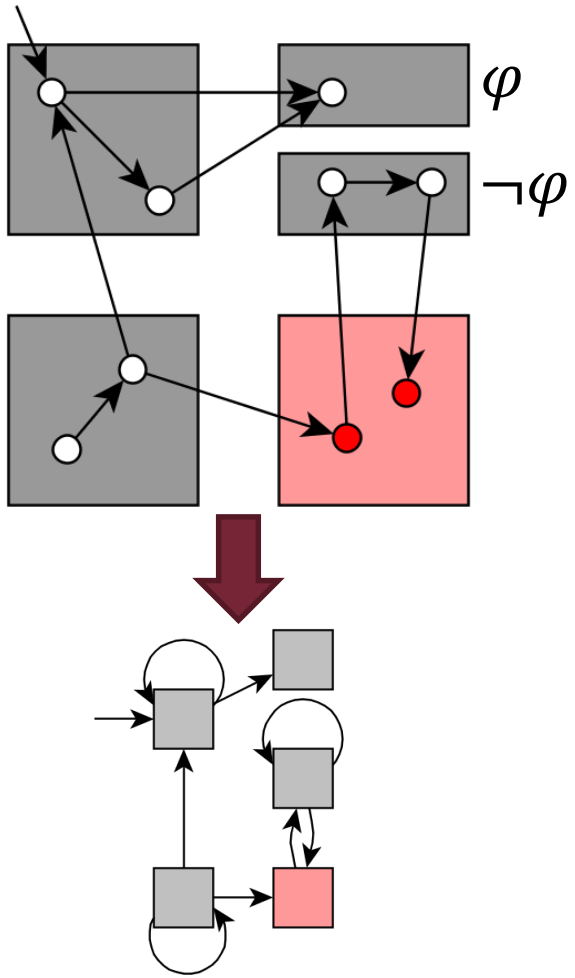
- Adott állapotig van út és onnan tovább is, de ezek különálló konkrét utak → „failure” állapot
- Konkrét állapotok csoportosítása a „failure” állapotban
  - D = „Dead-end”: elérhető
  - B = „Bad”: következő állapotra lép
  - IR = „Irrelevant”: többi



- Hamis ellenpélda oka
  - Predikátumhalmaz nem különbözteti meg D-t és B-t

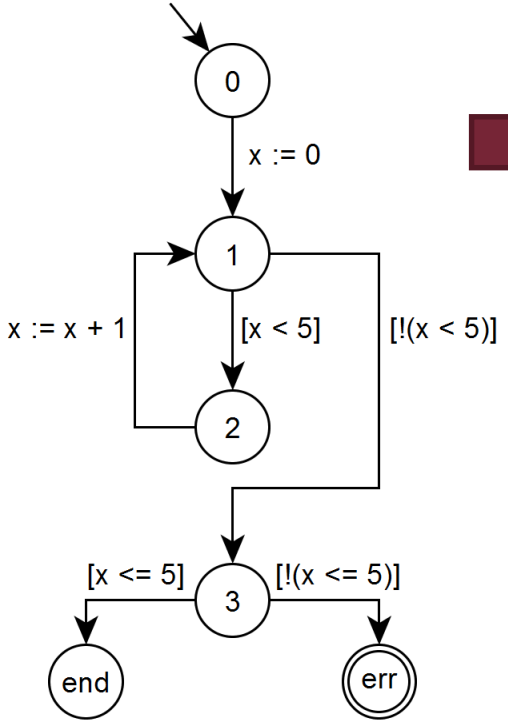
Hamis ellenpélda

- Hamis ellenpélda kiküszöbölése
  - Bővebb predikátumhalmaz (finomabb absztrakció)
  - D és B szétválasztása
    - Konkrét állapotok felsorolása nélkül
    - D és B leírása formulákkal
    - SMT megoldó képes egy φ formulát generálni, ami szétválaszt *(interpoláció)*
  - $P \cup \{\varphi\}$ predikátumhalmaz esetén ez a hamis ellenpélda megszűnik
    - Sőt, elég csak a „failure” állapotot vágni *(„lusta” absztrakció)*
- További hamis ellenpéldák
  - Újabb predikátumok

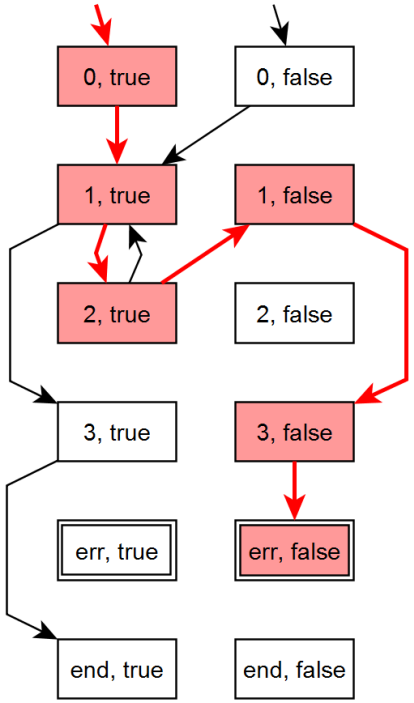


Hamis ellenpélda

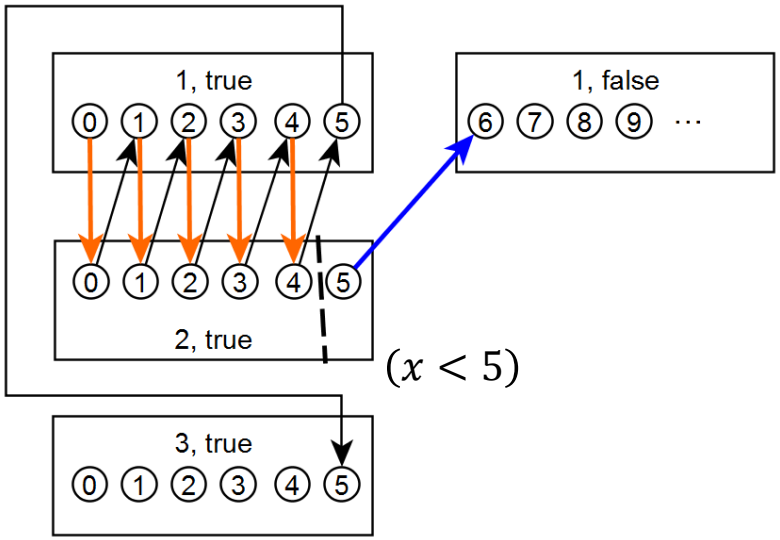
- Példa



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

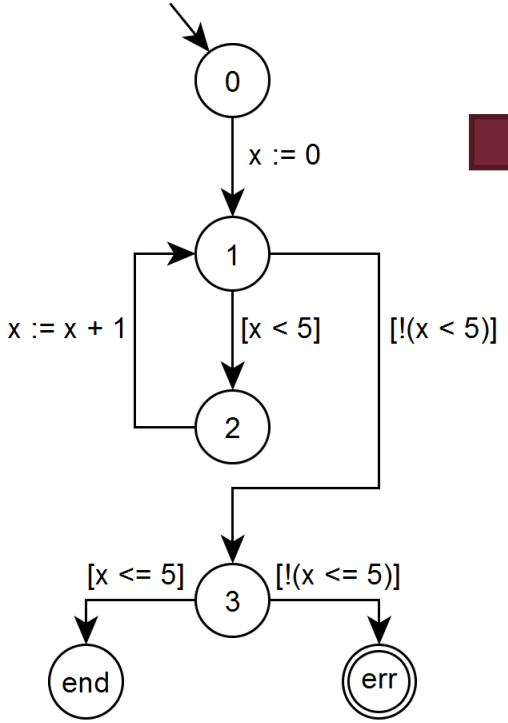


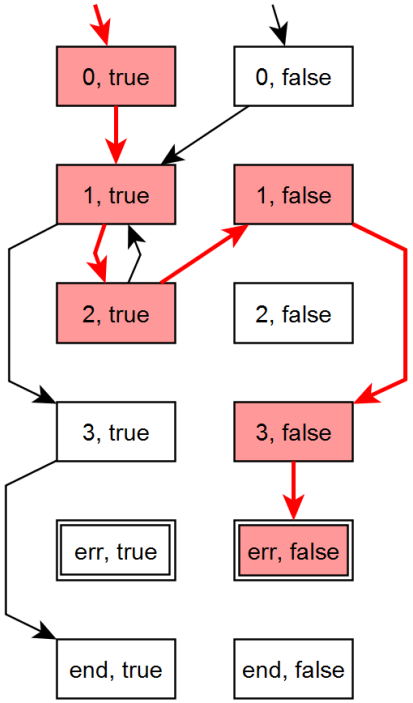
$(x_1 = 3)$
$(x_2 = 0)$
$(x_3 = 0)$
$(x_4 = ?)$

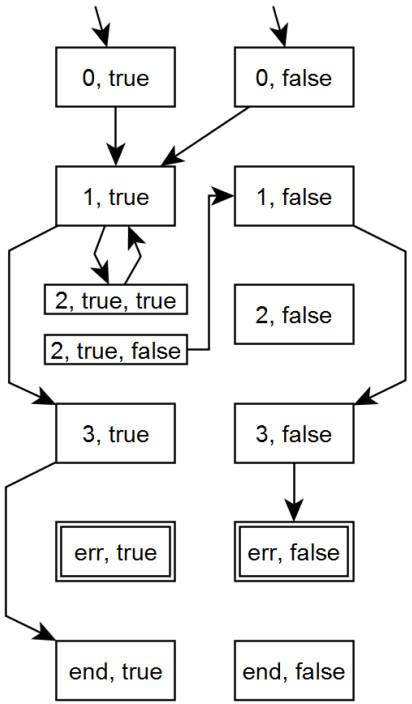


Hamis ellenpélda

- Példa



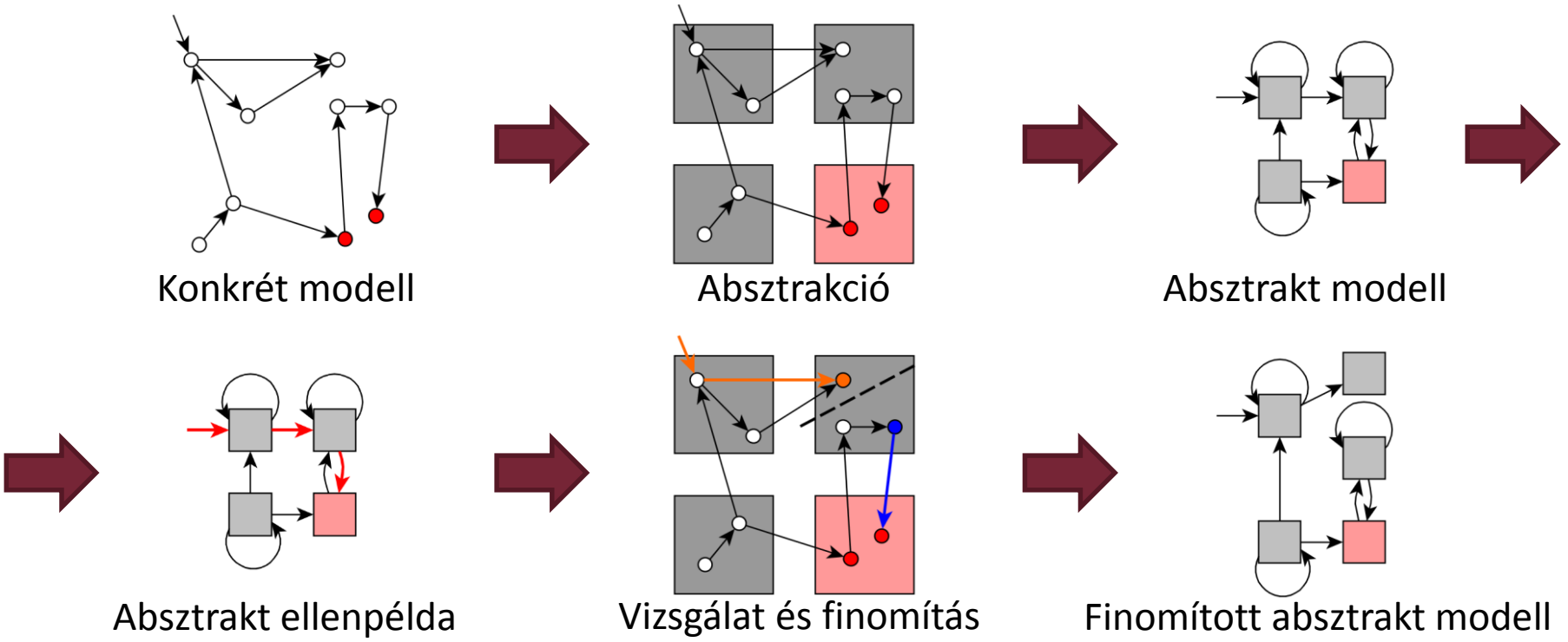


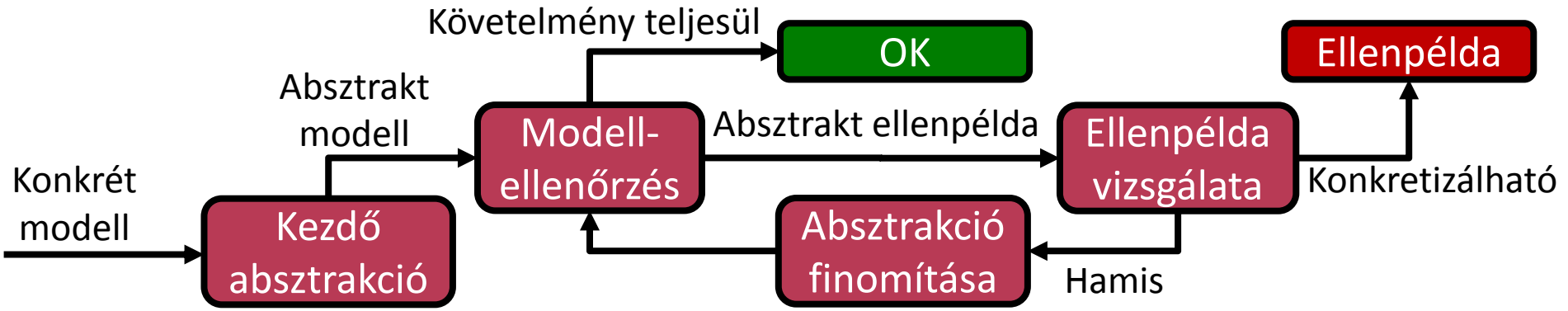


$P = \{(x \leq 5)\}$

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

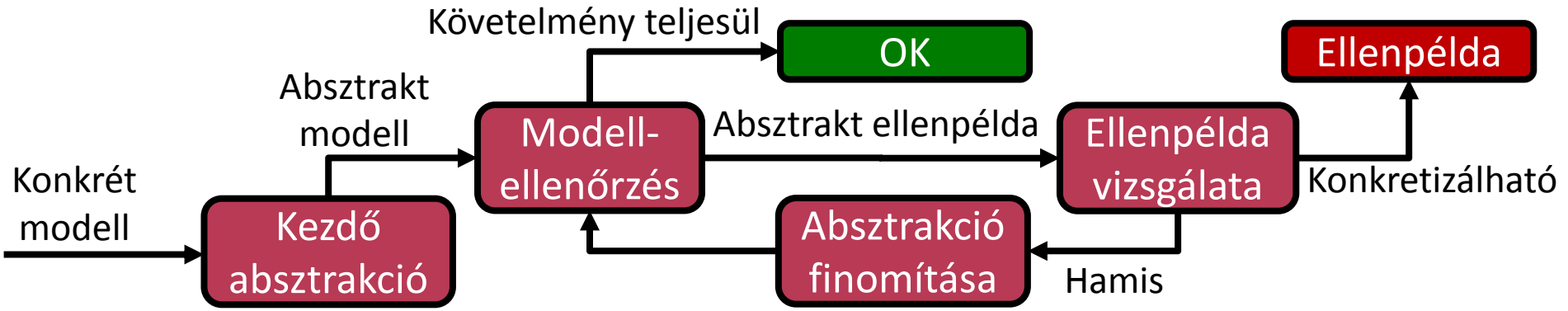
A teljes algoritmus





A teljes algoritmus

- Counterexample-Guided Abstraction Refinement (CEGAR)
  - Automatikus módszer
    - Minden lépése automatikusan működik
    - Nem szükséges formális módszerek részletes ismerete
  - És a kezdeti predikátumhalmazt ki mondja meg?
    - Lehet üres halmaz is akár
    - Programban szereplő feltételes utasításokból
    - Egyéb heurisztikák alapján



ESZKÖZÖK

Eszközök

- SLAM2
  - Static Driver Verifier Research Platform (SDVRP) része
  - Felépítése
    - Driver C kód: vizsgált komponens
    - Platform model: környezet leírása
    - Ellenőrzés: API használati szabályok betartása
  - Működése
    - Boole-program előállítása predikátumabsztrakcióval
    - Szimbolikus modellellenőrzés: BEBOP eszköz
    - CEGAR ciklus
  - research.microsoft.com/en-us/projects/slam/

Eszközök

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

Eszközök

- CPAChecker
  - The Configurable Software-Verification Platform
  - Bemenet: C program + specifikáció
    - Assertion, error címke, deadlock, null dereference, …
  - Nagymértékben konfigurálható
    - Többféle absztrakció típus (nem csak predikátum)
    - Absztrakt ellenpélda több prefixét tekinti
      - Többféle lehetséges finomítás közül választ (finomítási stratégia)
  - cpachecker.sosy-lab.org/

Eszközök

- Competition on Software Verification 2016 (SV-COMP)
  - sv-comp.sosy-lab.org/2016/
  - 35 eszköz, 6661 bemeneti verifikációs probléma (program + követelmény)
  - 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)

ÖSSZEFOGLALÁS

Összefoglalás

- Szoftver-modellellenőrzés
  - Szokásos probléma: állapottér robbanás
  - Megoldás: absztrakció
    - Vezérlési hely + predikátumok
    - Egzisztenciális absztrakció tulajdonsága
  - CEGAR: megfelelő absztrakció automatikus előállítása
    1. Kezdő absztrakt modell elkészítése
    2. Modellellenőrzés
    3. Ellenpélda vizsgálata
    4. Absztrakció finomítása
  - Eszközök