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

Időzített automaták absztrakciófinomítás-alapú verifikációja

Tézisfüzet

Tóth Tamás

Témavezető:
Majzik István, PhD (BME)

Budapest, 2021.

1. Előzmények és célok

A *beágyazott rendszerek* mindennapjainkban betöltött szerepe egyre jelentősebb. Ez magába foglalja *biztonságkritikus rendszerek* részeként történő alkalmazásukat, például vasúti, illetve autó- és repülőgépipari szakterületen. A biztonságkritikus rendszerek sokszor egyben *valósidejű rendszerek* időfüggő viselkedéssel és követelményekkel. E rendszerek helyes viselkedése kiemelt fontosságú, mivel egy rendszerszintű meghibásodás katasztrofális következményekkel, például környezeti vagy súlyos anyagi károkkal, de akár emberéletben keletkező károsodással is járhat. Ennek megfelelően az ilyen rendszereket a hibák valószínűségének és súlyosságának csökkentése érdekében részletesen specifikálják, és a specifikációnak való megfelelést alaposan ellenőrzik.

A *formális módszerek* olyan matematikai módszerek, melyek hardver-szoftver rendszerek precíz specifikációját és tipikusan tervezési idejű verifikációját célozzák. A *formális verifikációs* technikák olyan formális módszerek, melyek lehetővé teszik rendszerek egy adott specifikáció vagy tulajdonság szerint értelmezett helyességéről való érvelést. A *modellellenőrzés* [EC82; QS82] a tervezési modell állapotterének kimerítő bejárásán alapuló automatikus formális verifikációs technika, melynek a hagyományos ellenőrzési módszerekkel (például a teszteléssel) szemben előnye, hogy nemcsak hibás rendszerek hibáit képes detektálni, hanem hibamentes rendszerek helyességét is képes igazolni. A modellellenőrzés a gyakorlatban előforduló rendszerek ellenőrzésére történő alkalmazásának nehézsége ugyanakkor a módszer magas számítási költsége: egy rendszer állapotterének számossága tipikusan a rendszer viselkedését leíró bemeneti specifikáció méretében exponenciális – ez közismert nevén az *állapottérrobbanás* problémája. Ráadásul az állapottér nem is feltétlenül véges, például valósidejű rendszerek esetében, ahol az idő dimenziójú folytonos változók a specifikáció részét képezik.

Ezért a probléma kezelhetőbbé tételének érdekében a fejlett modellellenőrző eszközök gyakran alkalmaznak *szimbolikus technikákat* [Bur+92], ahol az egyes állapotok helyett állapotok halmazai képezik az állapottérbejárás alapját; valamint *absztrakciót* [CGL94], ahol az ellenőrzés a rendszer a vizsgált tulajdonság szempontjából releváns részleteire fókuszál. Az absztrakció eredménye egy egyszerűbb, az eredeti rendszer viselkedését felülbecslő rendszer, így ha az absztrakt rendszer helyes, akkor az eredeti is az. Ugyanakkor, mivel az absztrakt rendszer hamis ellenpéldákat produkálhat – azaz az eredeti rendszerben nem megfigyelhető hibás viselkedéseket – a megfelelő absztrakciós granularitás megtalálása kulcsfontosságú. Ezt a folyamatot automatizálja az *absztrakciófinomítás* [Cla+00] módszere: hamis ellenpélda esetén az absztrakció oly módon kerül hangolásra, mely kizárja a felfedezett hamis viselkedést.

Célok. Annak érdekében, hogy a modellellenőrzés egy adott rendszerre alkalmazható legyen, az ellenőrzést megelőzően szükséges a rendszer egy megfelelő modellezési formalizmusban történő modellezése. A legtöbb modellellenőrző algoritmus egy adott formalizmus valamilyen tulajdonságának vizsgálatát célozza. Ugyanakkor ahogy egyre nő az igény rendszertervek formális ellenőrzése iránt, általánosabb eszközökre is szüksége van, mivel a legmegfelelőbb formalizmus és algoritmus feladatonként változhat, és nem biztos, hogy előre ismert. Célunk egy generikus, moduláris és konfigurálható *modellellenőrző keretrendszer* biztosítása, mely támogatja különböző formalizmusok *biztonságossági tulajdonságokat* vizsgáló *absztrakciófinomítás-alapú algoritmusainak* fejlesztését és kiértékelését. Célunk továbbá valósidejű rendszerek modellellenőrzésére hatékony algoritmusokat adni e keretrendszer konkrét megpéldányosításai által. Főként a *valósidejű rendszerek* modellellenőrzésére elterjedten alkalmazott, folytonos *óraváltozókkal* rendelkező klasszikus *időzített automatákra*, valamint ezek *diszkrét változókkal* kiegészített változatára összpontosítunk. Ezen felül aszinkron üzeneteket küldő ipari valósidejű protokollok *előségi tulajdonságainak* bizonyítására adunk módszereket.

Szerkezet. A tézisfüzet hátralévő részének struktúrája az alábbiak szerint alakul. A munka során megcélzott kihívásokat az 1.1. szakasz foglalja össze. Az alkalmazott kutatási megközelítést, valamint a disszertáció új eredményeinek kivonatát a 2. szakasz foglalja össze. Az eredmények alkalmazásait a 3. szakasz írja le röviden. Végezetül a szerző a disszertáció szempontjából releváns publikációit a 4. szakasz listázza.

1.1. Kihívások összefoglalása

A disszertációban az alábbi kihívásokkal foglalkozunk.

1. **kihívás** *Konfigurálható absztrakciófinomítás-alapú modellellenőrzés.* A legtöbb eszköz egy adott formalizmushoz fejlesztett algoritmusokra fókuszál. Lehetséges olyan generikus, moduláris és konfigurálható modellellenőrző keretrendszert nyújtani, mely támogatja különböző formalizmusok elérhetőségi tulajdonságait ellenőrző algoritmusok fejlesztését, kiértékelését és alkalmazását?
2. **kihívás** *Időzített automaták absztrakciófinomítása.* Az absztrakciófinomítás módszerét elterjedten alkalmazzák modellellenőrzésre, és különösképpen szoftvermodellellenőrzésre. Lehetséges olyan absztrakciófinomítás-alapú algoritmusokat adni, melyek hatékonyan alkalmazhatóak valósidejű rendszerek modellellenőrzésére?
3. **kihívás** *Diszkrét változókkal rendelkező időzített automaták modellellenőrzése.* A gyakorlatban alkalmazott időzített rendszerek a valósértékű óraváltozókon felül tipikusan nemtriviális adatfolyam részét képző diszkrét változókkal is rendelkeznek. Lehetséges olyan módszert javasolni, mely ilyen rendszerek esetében mérsékli az állapottérrobbanás hatását?
4. **kihívás** *Ipari valósidejű rendszerek élőségi ellenőrzése.* Az ipari valósidejű rendszerek egyes követelményei gyakran élőségi tulajdonságként írhatóak le. Lehetséges módszert adni ilyen rendszerek élőségi ellenőrzésére úgy, hogy a modellekben megjelenő számos szemantikai jellemző támogatása továbbra is megmarad?

Ezen kihívások mindegyike a disszertáció egy-egy tézispontjában kerül kifejtésre.

2. Kutatási módszerek és új eredmények

Kutatásaim során az alábbi kutatási módszertant alkalmaztam. A szakirodalom alapján újszerű verifikációs módszereket és *algoritmusokat* javasoltam, valamint formális *bizonyítást* adtam ezek helyességére. A módszerek gyakorlati alkalmazhatóságának igazolására elkészítettem az algoritmusok *prototípus* implementációját. Végezetül a javasolt algoritmusok használhatóságának és teljesítményének kiértékelése végett *kísérleteket* végeztem bemeneti modellek széles skáláján.

2.1. A kutatás kontextusa

Ebben az alfejezetben röviden vázoljuk azon formalizmusokat, melyek modellellenőrzése a tézisek alapját képezi.

2.1.1. Időzített automata

Az időzített automata [AD94] a valósidejű rendszerek modellezésére és ellenőrzésére széles körben alkalmazott formalizmus. Lényegét tekintve egy időzített automata hasonlít egy vezérlési helyekkel, tranzíciókkal és egy kitüntetett kiinduló vezérlési hellyel rendelkező véges automatához, ezek mellett ugyanakkor egyúttal valós értékű óraváltozókkal – valamint az ezen óraváltozók felett értelmezett, az éleket címkéző őrfeltételekkel és értékadásokkal – is rendelkezik, melyek alkalmassá teszik időfüggő viselkedés kifejezésére.

Legyen C óraváltozók egy $\mathbb{R}_{\geq 0}$ felett értelmezett halmaza. Egy C feletti *órakényszer* $x \prec k$ vagy $x - y \prec k$ alakú atomok konjunkciója, ahol $x, y \in C$ és $k \in \mathbb{Z}$ és $\prec \in \{<, \leq\}$. A C feletti óraváltozók halmazára bevezetjük a $Constr_C$ jelölést. Szintaktikailag egy időzített automata egy $\mathcal{A} = (L, C, T, \ell_0)$ négyes, ahol

- L helyek egy véges halmaza,
- C óraváltozók egy véges halmaza,
- $T \subseteq L \times Constr_C \times \mathcal{P}(C) \times L$ a tranzíciós reláció, ahol egy $(\ell, g, R, \ell') \in T$ tranzícióhoz a g kényszer egy őrfeltételt jelöl, az R pedig a nullázandó óraváltozók halmaza, valamint
- $\ell_0 \in L$ a kiinduló hely.

Az időzített automata egy állapota az (ℓ, η) rendezett pár, ahol $\ell \in L$ és $\eta : C \to \mathbb{R}_{\geq 0}$. Legyen $\eta_0 = x \mapsto 0$. Legyen $delay_d(\eta)(x) = \eta(x) + d$ minden $x \in C$ esetén. Végezetül legyen $reset_R(\eta)(x) = 0$, ha $x \in R$, illetve $reset_R(\eta)(x) = \eta(x)$, ha $x \notin R$. Ekkor egy időzített automata operációs szemantikája egy olyan címkézett tranzíciós rendszerként írható le, melynek kezdőállapota (ℓ_0, η_0), tranzíciós relációja pedig az alábbi két szabály által meghatározott legszűkebb reláció:

$$\frac{d \geq 0}{(\ell, \eta) \xrightarrow{d} (\ell, delay_d(\eta))} \text{ delay}$$

$$\frac{t = (\ell, g, R, \ell') \qquad t \in T \qquad \eta \models g}{(\ell, \eta) \xrightarrow{t} (\ell', reset_R(\eta))} \text{ discrete}$$

Az időzített automata egy *útja* tranzíciók egy $\pi \in T^*$ szekvenciája. Egy út *megvalósítható* a.cs.a., ha megfeleltethető állapotok egy olyan, az automata szemantikája értelmében egymást követő sorozatának, mely a kezdőallapotból indul ki. A helyelérhetőségi probléma arra a kérdésre keresi a választ, hogy egy adott hibahely elérhető-e az automata (vagy automaták hálózata) egy megvalósítható útja mentén.

2.1.2. Naptárrendszer

A naptárrendszer formalizmus egy időzített rendszerek tranzíciós rendszerként történő leírására szolgáló magasabb szintű formalizmus. A formalizmus a diszkrét állapotszimulációból meríti az alapötletét: az időzített automaták elméletéből ismert óraváltozók helyett (melyek egy adott esemény bekövetkezte óta eltelt időt mérik) a naptárrendszer olyan változókkal rendelkezik, melyek a jövőben bekövetkező eseményeket tárolnak. Habár így az idő a végtelenbe tart, ezáltal végtelen állapotteret eredményezve, a formalizmus jól kezelhető indukcióval.

Legyen $\Delta = \{[a, b], (b, c], [b, c), (b, c) \mid 0 \leq a \leq b < c \text{ és } a, b, c \in \mathbb{N}\}$ és $A^? = A \cup \{\mathsf{none}\}$ és $A^! = A \setminus \{\mathsf{none}\}$. Egy naptárrendszer egy $(L, T, M, \to, \ell_0, T_0)$ hatos, ahol

- L helyek egy véges halmaza,
- T időzítések egy véges halmaza,
- M üzenetek egy véges halmaza,
- $\to \subseteq L \times Event \times Action_M \times \mathcal{P}(Action_T) \times L$ a tranzíciós reláció, ahol $Event = T \cup M$ a *kiváltó események* halmaza, $Action_M = (M \times \Delta)^?$ az *üzenetküldési akciók* halmaza, és $Action_T = T \times \Delta^?$ az *időzítésállítási akciók*,
- $\ell_0 \in L$ a kiinduló hely, és végezetül
- $T_0 : T \to \Delta^?$ egy függvény, mely az időzítésekhez azok kiinduló értékét rendeli.

A naptárrendszer kezdőállapota az (ℓ, σ) rendezett pár, ahol $\ell \in L$ és σ egy függvény, melynek értelmezési tartománya $T \cup \{C, \tau\}$ úgy, hogy $\sigma(\tau) \in \mathbb{R}_{\geq 0}$ az aktuális idő, $\sigma(x) \in \mathbb{R}^?_{\geq 0}$ az $x \in T$ időzítő aktuális értéke, valamint $\sigma(C)$ multihalmaz az *eseménynaptár*, ahol egy $(m, t) \in \sigma(C)$ elem esetén $t \in \mathbb{R}_{\geq 0}$ szám jelöli az $m \in M$ üzenet bekövetkeztét. A kezdőállapotok (ℓ_0, σ_0) alakúak, ahol $\sigma_0(\tau) = 0$ és $\sigma_0(C) = \emptyset$ és minden $x \in T$ esetén $\sigma_0(x) \in T_0(x)$

ha $T_0(x) \in \Delta$ és $\sigma_0(x) = \mathsf{none}$ egyébként. Továbbá a naptárrendszer minden $\ell \xrightarrow{e,\mu,S} \ell'$ tranzíciójához tartozik a szemantikát meghatározó tranzíciós rendszer egy $(\ell, \sigma) \xrightarrow{e} (\ell', \sigma')$ tranzíciója, ahol $\sigma'(\tau) = \min(\sigma(T)^! \cup (\mathsf{snd} \circ \sigma)(C))$, továbbá teljesülnek az alábbi feltételek.

- Minden $x \in T$ időzítés következő értékét az alábbi szabályok közül pontosan egy határozza meg.

$$\frac{(x,\delta) \in S \qquad \delta = \mathsf{none}}{\sigma'(x) = \mathsf{none}} \ \mathsf{invalidate}\ x$$

$$\frac{(x,\delta) \in S \qquad \delta \in \Delta \qquad d \in \delta}{\sigma'(x) = \sigma(x) + d} \ \mathsf{set}\ x$$

$$\frac{\forall \delta.(x,\delta) \notin S}{\sigma'(x) = \sigma(x)} \ \mathsf{skip}\ x$$

- A C naptár következő értékét az alábbi szabályok közül pontosan egy határozza meg.

$$\frac{e \in T \qquad \sigma(e) = \sigma(\tau) \qquad \mu = \mathsf{none}}{\sigma'(C) = \sigma(C)} \ e\ \mathsf{over\ /\ send\ none}$$

$$\frac{e \in T \qquad \sigma(e) = \sigma(\tau) \qquad \mu = (m,\delta) \qquad d \in \delta}{\sigma'(C) = \sigma(C) \cup \{(m, \sigma(t) + d)\}} \ e\ \mathsf{over\ /\ send}\ m$$

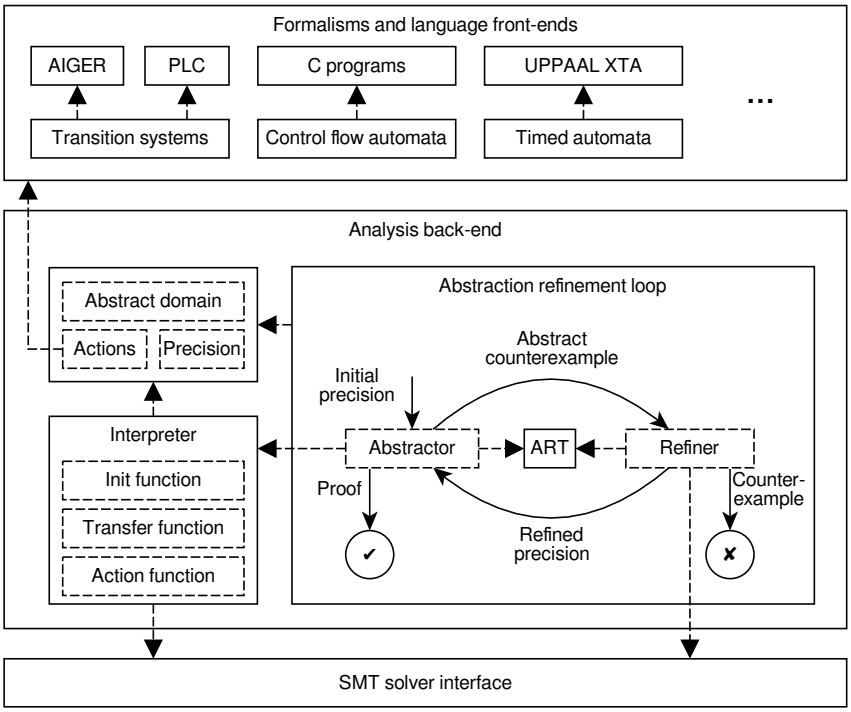
$$\frac{e \in M \qquad (e,\sigma(t)) \in \sigma(C) \qquad \mu = \mathsf{none}}{\sigma'(C) = \sigma(C) \setminus \{(e,\sigma(t))\}} \ e\ \mathsf{received\ /\ send\ none}$$

$$\frac{e \in M \qquad (e,\sigma(t)) \in \sigma(C) \qquad \mu = (m,\delta) \qquad d \in \delta}{\sigma'(C) = \sigma(C) \setminus \{(e,\sigma(t))\} \cup \{(m, \sigma(t) + d)\}} \ e\ \mathsf{received\ /\ send}\ m$$

2.2. Keretrendszer absztrakciófinomítás-alapú elérhetőségi ellenőrzéshez

Számos modellellenőrző eszköz létezik, melyek különböző formalizmusokhoz implementálnak hatékony algoritmusokat. A legtöbb eszköz egy-egy adott algoritmusra és formalizmusra fókuszál, és ezáltal alkalmas egy konkrét ellenőrzési feladat hatékony megoldására, tipikusan egy adott alaphalmaz más néven domén (domain) feletti absztrakciófinomítást alkalmazva. A predikátumabsztrakciót számos eszköz támogatja, például a Slam [BPR01], a Blast [Bey+07] és a SatAbs [Cla+05]. A lusta absztrakción alapuló eszközök, mint például az Impact [McM06] és a Wolverine [KW11] a predikátumdomén feletti absztrakció számítására költséges utóképszámítás (post-image computation) helyett Craig-interpolációt [McM03] alkalmaznak. Néhány eszköz a predikátumok helyett más alaphalmaz felett végez absztrakciófinomítást: a Dagger [Gul+08] eszköz támogatja az oktagon és poliéder doméneket, a Vinta [AGC12] algoritmus pedig intervallumok absztrakciófinomításán alapul. A CPAchecker [BK11] és az Ufo [Alb+12] keretrendszer absztrakció tekintetében konfigurálható a domén, az utókép-operátor (post-image operator), valamint a finomítási stratégia definiálhatósága által, de kizárólag szoftvermodelleket támogat. Az LTSmin eszköz több formalizmust is támogat a „Partitioned Next-State Interface” (PINS) [Kan+15] kiajánlásával, de absztrakciófinomítás helyett elsősorban szimbolikus és párhuzamos modellellenőrzési algoritmusokat implementál.

Ugyanakkor a legmegfelelőbb algoritmus és formalizmus a megoldandó probléma jellemzői függvényében változhat, ráadásul sokszor nem is ismert előre, ezért hasznos lehet egy olyan eszköz, mely egy adott formalizmust több formalizmusra is megvalósít, illetve fordítva, mely egy



1. ábra. A Theta keretrendszer architektúrája

adott formalizmushoz többféle algoritmust is megvalósít. Ennek támogatására fejlesztettük ki a Theta eszköz architektúráját. A Theta egy generikus, moduláris és konfigurálható modellellenőrző keretrendszer, melynek fő célja különböző formalizmusok absztrakciófinomítás-alapú elérhetőségi algoritmusai fejlesztésének és kiértékelésének támogatása. Az eszköz célja elsősorban kutatók számára biztosítani egy olyan keretrendszert, melyben könnyű prototipizálni, kiértékelni és összehasonlítani új komponenseket, valamint ezek kombinációját.

Az Theta eszköz legfőbb megkülönböztető jellemzője az architektúrája, mely lehetővé teszi különböző domének feletti absztrakciók és absztrakciófinomítási stratégiák kombinálását, valamint ezek alkalmazását különböző, magasabb szintű nyelvi frontendekkel támogatott formalizmusokra. A Theta eszköz architektúráját az 1. ábra szemlélteti (az ábrán a folytonos nyilak adatfolyamot, a szaggatott nyilak pedig függőségi kapcsolatot jelölnek). A keretrendszer főbb részei a nyelvi frontendek, az analízis backend, és az SMT-megoldó interfész. A keretrendszer központi eleme az analízis backend, mely további három részre bontható: az absztrakt doménre, az interpreterre, valamint az elérhetőségi vizsgálatot végző absztrakciófinomítási hurokra, ahol szigorú értelemben kizárólag az interpreter függ az elemzett formalizmustól.

Absztrakt domén. Az analízis szemantikai alapját az *absztrakt domén* biztosítja, mely absztrakt állapotok egy alaphalmazából, egy minimum (bottom) elemből, valamint egy, az absztrakt állapotok feletti kvázirendezésből (preorder) áll. Az adott analízis pontosságát formálisan a precíziók (precision) halmazának egy eleme határozza meg. (Tipikus példaként a precízió állapotváltozók azon halmaza lehet, melyek értékét az analízis során követni szeretnénk – így minél nagyobb e halmaz számossága, annál pontosabb az analízis.) A formalizmus, mely felett az analízis értelmezett, akciók egy halmazát határozza meg, ahol az egyes akciók az utóképszámítás bemenetéül szolgálnak.

Interpreter. Az *interpreter* az absztrakt domén és az akciók halmaza feletti operációs szemantikát definiál a precízió ismeretében. Az absztrakt kezdőállapotokat a *kezdőállapot-függvény* (init

function) határozza meg. Egy állapot adott akció szerinti rákövetkező állapotait a *transzferfüggvény* (transfer function) számítja ki. Az adott absztrakt állapotban engedélyezett akciók halmazát az *akciófüggvény* (action function) adja meg. Az interpreter fontos szerepet játszik a keretrendszer generikusságában, mivel szétcsatolja az (absztrakt domén által meghatározott) absztrakció és az (akciók halmaza által meghatározott) formalizmus fogalmát.

Absztrakciófinomítási hurok. Az elérhetőségi analízist az *absztrakciófinomítási hurok* (abstraction refinement loop) végzi. Az eljárás központi adatstruktúrája a lusta absztrakciós (lazy abstraction) módszerek [McM06] esetében megszokott módon egy *absztrakt elérhetőségi fa* (abstract reachability tree, ART), melynek csomópontjait az adott úton elérhető állapotok egy felülbecslését leíró absztrakt állapotok, éleit pedig akciók címkézik. Az ART manipulációját a hurok két fő komponense végzi: Az *absztraktor* (abstractor) az ART-ot az interpreter felhasználásával az aktuális precíziónak és absztrakciós stratégiának megfelelően konstruálja, utóbbit az alábbi alapműveletek határozzák meg:

- *Kifejt* (expand). Mikor szükséges egy adott csomópont kifejtése, azaz a fa növesztése az adott csomópontból a csomópont rákövetkezői hozzáadása által?
- *Fed* (cover). Milyen módon próbálja vágni (prune) a keresést az absztraktor fedő, azaz olyan csomópontokkal, melyek az aktuális csomópontnál bővebb állapothalmazt reprezentálnak?
- *Terminál* (terminate). Milyen feltételek fennállása esetén termináljon a keresés?

Amennyiben a keresés nem talál célcsomópontot – olyan csomópontot, mely a bemeneti modell alapján nem biztonságos állapotokat kódol – a konstruált ART a bemeneti modell helyességének bizonyítékaként szolgál. Ellenkező esetben a célcsomópontba vezető út megvalósíthatóságának vizsgálata céljából meghívásra kerül a *finomító* (refiner). Amennyiben az út megvalósítható, úgy a modell helyességének ellenpéldájaként szolgál. Ellenkező esetben a finomító végrehajtja a finomítási stratégiát, ezáltal biztosítva, hogy az analízis anélkül tudjon folytatódni, hogy ugyanezt a hamis ellenpéldát ismét felderítené. Ez tipikusan egy új precízió megállapításával és a fa vágásával (felülbecslés-vezérelt megközelítés), vagy bizonyos csomópontok fedésének megszüntetésével és címkézésének erősítésével (alulbecslés-vezérelt megközelítés) történhet, mindkét esetben az eljárás az ART részleges dekonstrukciójával jár.

Konkrét ellenőrző eszközök implementációja is elkészült, többek között tranzíciós rendszerek, vezérlésifolyam-automaták, valamint időzített automaták verifikációjára, melyek mindegyike különböző absztrakt doméneket (többek között predikátumokat, explicit értékeket és zónákat) és finomítási stratégiákat (többek között interpolációt és „unsat core” – egy formulahalmaz kielégíthetetlen részhalmaza – alapú finomítást) kombinál. Kiemelendő, hogy a keretrendszer segítségével – konkrétan absztrakciófinomítási hurok, valamint számos absztrakt domén és interpreter definiálásával – céloztuk meg diszkrét változókkal kiegészített időzített automaták helyelérhetőségi problémáját is. Ez alapján *formális algoritmikus keretrendszert* javaslunk, mely lehetővé teszi időzített automaták óraváltozói és diszkrét változói feletti absztrakt domének és finomítási stratégiák egységes formalizálását. A keretrendszer legfőbb előnye, hogy alkalmas a direktszorzat absztrakció fogalmára építve különböző absztrakciós módszerek zökkenőmentes *kombinálására*, ezáltal számos egyedi algoritmuskombinációt eredményezve, melyek együttesen időzítettautomata-modellek széles skálájának hatékony ellenőrzését teszik lehetővé. A keretrendszer tehát az elérhetőségi probléma megoldására generikus algoritmust definiál, mely az adott absztrakt doménben, és az e domén felett alkalmazandó konkrét absztrakciófinomítási stratégiában konfigurálható. E konfigurálhatóság tette lehetővé időzített automaták ismert hatékony lustaabsztrakció-alapú algoritmusainak a keretrendszerbe történő integrálását, és így a módszereink kombinálását és összehasonlítását a legfejlettebb módszerekkel.

1. tézis *Keretrendszer absztrakciófinomítás-alapú elérhetőségi ellenőrzéshez.* Megoldást javasoltam az absztrakciófinomítás-alapú modellellenőrzés modellezési formalizmusban, absztrakt doménben, valamint finomítási stratégiában történő konfigurálhatóságára.

1.1 *Egy konfigurálható modellellenőrző keretrendszer architektúrája.* Megterveztem a THETA, egy absztrakt domének, interpreterek, valamint absztrakciós és finomítási stratégiák kombinálását lehetővé tevő generikus, moduláris és konfigurálható modellellenőrző keretrendszer architektúráját, interfészeit, és általános algoritmikus komponenseit.

1.2 *Időzített automaták absztrakciófinomítási stratégiáinak egységes formalizálása.* Időzített automaták helyelérhetőségi problémájának megoldására absztrakt domének és absztrakciófinomítási stratégiák egységes formalizálását és kombinált használatát lehetővé tevő formális algoritmikus keretrendszert javasoltam, valamint bizonyítottam e keretrendszer helyességét.

Eredményeink újszerű algoritmusok és algoritmuskombinációk definiálását, implementálását és empirikus kiértékelését tették lehetővé. Az 1. tézis eredményeit az értekezés 3. és 4. fejezete tárgyalja. A tézishez az alábbi publikációk kapcsolódnak: [*j2*; *c6*; *c10*] .

2.3. Időzített automaták lusta elérhetőségi ellenőrzése interpolációval

Az időzített automaták [AD94] elérhetőségi problémája azzal a kérdéssel foglalkozik, hogy egy adott hibahely elérhető-e a kiinduló helyről az automata tranzíciói mentén. A probléma sztenderd megoldása az automata által indukált ú.n. zónagráfban végez előre keresést [DT98]. Ebben az esetben minden absztrakt állapot egy zóna, azaz konkrét állapotok egy olyan halmaza, mely órakényszerek egy halmazának megoldáshalmazaként írható le.

Az eljárás teljesítménye és terminálása érdekében a modellellenőrzők az automata őrfeltételeiben szereplő maximális alsó és felső korlátok (LU-bounds) alapján tipikusan általánosítják a zónákat [Beh+04]. E lépés elvégezhető közvetlenül statikus analízissel nyert korlátok [Beh+03] alapján paraméterezett extrapoláció által [Beh+04]. Alternatív megoldásként a korlátok menet közben is terjeszthetőek az automata minden tranzíciója [Her+11] vagy hamis utak mentén [HSW13], ami az absztrakció figyelembe vételével történő részhalmazvizsgálat egy hatékony implementációjával kiegészítve [HSW12] hatékony algoritmust ad időzített automaták elérhetőségi vizsgálatára. Ez utóbbi megközelítés egyfajta lusta absztrakció, ami az ellenpélda-alapú absztrakciófinomítás [Cla+03] (counterexample-guided abstraction refinement, CEGAR) egy olyan változata, ahol – az absztrakció egy tipikusan költséges absztrakt utóképszámítás segítségével történő buzgó kiszámítása helyett – az absztrakció menet közben, lokálisan kerül kiszámításra az állapottér azon útjai mentén, ahol nagyobb pontosságra van szükség.

Az időzített automaták helyelérhetőségi problémájának megoldására hasonló lusta algoritmust javaslunk. A javasolt algoritmus ugyanakkor az őrfeltételekben szereplő korlátok helyett magukat az őrfeltételeket veszi alapul: amennyiben az absztrakció túl durva felbontású egy hamis út kizárásához, az út mentén interpolánst [McM03] számítunk, azaz zónák egy olyan induktív sorozatát, mely kellően gyenge ahhoz, hogy minden elérhető állapotot magában foglaljon, és egyidejűleg kellően erős ahhoz, hogy kizárja a végrehajthatatlan tranzíciót. Hasonlóképpen alkalmazzuk az interpolációt a keresési tér hathatós vágásához az új állapotok a már korábban felfedezett állapotok által történő fedésének lehetőség szerinti kikényszerítése által. E keretrendszerben kétfajta finomítási stratégiát javaslunk: az „előre” zónainterpolációs stratégia az interpolánst előrefelé terjeszti utóképszámítással, míg a „hátra” zóna interpolációs stratégia az interpoláns komplementereként kapott hibazónákat terjeszti hátrafelé előképszámítással. Mindkét módszer az előrefelé keresés, a hátrafelé keresés, és a zónainterpoláció kombinációja, és egyúttal a zónainterpoláció tranzíciók sorozatára történő általánosításának tekinthető.

2. tézis *Időzített automaták lusta elérhetőségi ellenőrzése interpolációval.* Megoldást javasoltam időzített automaták helyelérhetőségi problémájára az alábbi lépések mentén.

- Definiáltam a zónák feletti interpoláció fogalmát, és algoritmust adtam két, egymással inkonzisztens, kanonikus DBM-ként (difference bound matrix) ábrázolt zóna interpolánsának kiszámítására.
- Időzített automaták zónaabsztrakciója felett értelmezett elő- és utóképszámítás segítségével általánosítottam a zónainterpoláció fogalmát interpolánsok sorozatára, ezáltal alkalmassá téve a módszert időzített automaták absztrakciófinomítás-alapú helyelérhetőségi ellenőrzésére.
- Az előre és hátra irányú zónainterpolációt mint a lusta absztrakciófinomítás lehetséges módszereit javasoltam.
- A javasolt absztrakciófinomítási stratégiák teljesítményét kísérleti úton kiértékeltem, és demonstráltam, hogy teljesítmény tekintetében kedvezően viszonyulnak a lusta nemkonvex absztrakciókon alapuló ismert módszerekhez.

A javasolt módszer időzített automaták nagyobb kifejezőerejű változataira is alkalmazható, például az őrfeltételekben diagonális kényszereket tartalmazó automatákra [BLR05], vagy olyan automatákra, ahol az órák értéke nem csak nullára állítható (updatable timed automata) [Bou04]. A 2. tézis eredményeit az értekezés 5. fejezete tárgyalja. A tézishez az alábbi publikációk kapcsolódnak: [*j2*; *c8*; *c9*] .

2.4. Diszkrét változókkal rendelkező időzített automaták lusta elérhetőségi ellenőrzése

Az időzített automaták ellenőrzését célzó módszerek ritkán foglalkoznak a valósidejű rendszerekben a gyakorlatban rendszerint előforduló *diszkrét adatváltozók absztrakciójával*, vagy ezt egy teljesen SMT-alapú (satisfiability modulo theories) megközelítést alkalmazva teszik, és így mind a folytonos, mind a diszkrét változók absztrakciója a mögöttes döntési eljárás hatékonyságán múlik.

Munkánk során a diszkrét változókkal rendelkező időzített automaták helyelérhetőségi problémájával foglalkozunk, és ennek megoldására egy olyan absztrakciós módszert javaslunk, mely alkalmas az ilyen specifikációkban szereplő *diszkrét változók láthatóságának lusta szabályozására*: amennyiben az absztrakció túlságosan durva felbontású ahhoz, hogy kizárjon egy nem végrehajtható tranzíciót – mely az egész útvonalat végrehajthatatlanná teszi –, akkor kiszámoljuk látható változók [Kur94; CGS04; Cha+02; Gru06] egy olyan halmazát, mely változók értékének az út mentén történő nyomonkövetése elégséges a nem végrehajtható tranzíció absztrakt állapottérből történő kizárásához. A megközelítés egy megfelelő absztrakció kiszámításához a – változóhozzárendelésekre értelmezett – interpoláció [McM03] módszerét kombinálja kényszereket hátrafelé terjesztő leggyengébbelőfeltétel-számítással (weakest precondition). E keretrendszerben kétfajta finomítási stratégiát javaslunk: az „előre” értékelésinterpolációs (valuation interpolation) stratégia az interpolánst előrefelé terjeszti utóképszámítással, míg a „hátra” értékelésinterpolációs stratégia az interpolánst karakterizáló formulát terjeszti hátrafelé leggyengébbelőfeltétel-számítással. A módszer nem igényel interpolációt támogató SMT-megoldót, és szabadon kombinálható az óraváltozókat hatékonyan kezelő (buzgó vagy lusta) zónaalapú előre kereső módszerekkel.

3. tézis *Diszkrét változókkal rendelkező időzített automaták lusta elérhetőségi ellenőrzése.* Megoldást javasoltam diszkrét változókkal rendelkező időzített automaták helyelérhetőségi problémájára az alábbi lépések mentén.

- Definiáltam egy változóértékelés és formula feletti interpoláns fogalmát, és eljárást adtam értékelésinterpolánsok kiszámítására.

- Időzített automaták tranzícióira alkalmazott leggyengébbelőfeltétel-számítás segítségével általánosítottam az értékelésinterpoláció fogalmát interpolánsok sorozatára, ezáltal alkalmassá téve a módszert időzített automaták absztrakciófinomítás-alapú helyelérhetőségi ellenőrzésére.
- Az előre és hátra irányú értékelésinterpolációt mint a lusta absztrakciófinomítás lehetséges módszereit javasoltam.
- A javasolt absztrakciófinomítási stratégiák teljesítményét kísérleti úton kiértékeltem, és demonstráltam, hogy alkalmasak a sok diszkrét változóval rendelkező időzített automaták állapottérbejárása során generált állapotok számának jelentős csökkentésére.

Mivel a javasolt módszer nem SMT-megoldáson alapul, így diszkrét változók felett tetszőleges kifejezést (például osztást vagy változók közötti szorzást) és utasítást tartalmazó időzített automatákra alkalmazható. A 3. tézis eredményeit az értekezés 6. fejezete tárgyalja. A tézishez az alábbi publikációk kapcsolódnak: [*j*1; *j*2; *c*4; *c*7; *c*11; *c*12; *e*13] .

2.5. Továbbfejlesztett módszerek ipari valósidejű protokollok élőségi ellenőrzéséhez

A biztonságossági tulajdonságok (safety properties) rendszerek olyan tulajdonságai, melyek végtelen lefutások véges prefixeinek megfigyelése által eldönthetőek. Az elérhetőségi tulajdonságok – melyek ellenőrzését a korábbi tézispontok tárgyalják – a legegyszerűbb biztonságossági tulajdonságok, melyek azt fejezik ki, hogy a rendszer lefutásai mentén nem érhetőek el rossz állapotok. A biztonságossági tulajdonságokkal ellentétben az élőségi tulajdonságok (liveness properties) olyan tulajdonságok, melyek esetében minden véges prefixum kiegészíthető egy, a tulajdonságot teljesítő végtelen lefutásra, mely tipikusan valamely, a rendszer szempontjából elvárt viselkedés előbb-utóbb történő bekövetkeztét írja le. Habár időzített rendszerek biztonságossági tulajdonságainak ellenőrzésére számos hatékony modellellenőrző létezik, az élőségi tulajdonságok vizsgálatát lehetővé tevő eszközök ritkábbak. Kutatásunk során ipari valósidejű protokollok élőségi ellenőrzésével foglalkoztunk.

Aszinkron üzenetalapú rendszerek ellenőrzésére egy k-indukció [SSS00; BC00; MRS03; ES03] alapú módszert javasoltunk. A k-indukció alapötlete a teljes indukció módszerének a rendszer állapotterében futó utak hosszára történő alkalmazása a szokásos egylépéses vizsgálat helyett a tranzíciós reláció k lépéses kigörgetésével, ami hatására az indukciós lépés nagyobb valószínűséggel jár sikerrel. Bár a k-indukció eredetileg hardvermodellek biztonságossági ellenőrzésére alkalmazott módszer, munkánk során kiterjesztettük időzített rendszerek élőségi vizsgálatára. A javasolt módszer kiindulópontja egy magasszintű rendszermodell, melyet az időzített aszinkron üzenetalapú rendszerek kényelmes modellezését lehetővé tevő naptárrendszer formalizmusban specifikálunk. Ezt a köztes modellt ezután egy szimbolikus tranzíciós rendszer alapú specifikációja fordítjuk. A specifikációt ezután monitorkomponensekkel egészítjük ki, melyek az élőségi vizsgálatot biztonságossági vizsgálattá redukálják, valamint olyan, az eredeti modellből származtatott lemmákkal bővítjük, melyek az ellenőrzendő követelmény erősítésével hozzájárulnak az indukciós lépés sikerességéhez. A teljes specifikációt ezt követően egy k-indukciós modellellenőrzővel verifikáljuk.

A biztonságkritikus rendszerek tipikusan hierarchikus felépítésűek, ahol különböző funkciók rétegesen épülnek egymásra. A valós rendszerekkel szemben támasztott követelményekre ezen felül jellemző a komplexitás abban az értelemben, hogy a követelmények rendszerint biztonságossági és élőségi tulajdonságok kombinációjaként állnak elő. Ilyen rendszerek verifikációjának elősegítése érdekében módszert javaslunk a specifikáció e rétegzett struktúrájának kisebb és kezelhetőbb verifikációs problémákra történő dekompozíciójára.

4. tézis *Továbbfejlesztett módszerek ipari valósidejű protokollok élőségi ellenőrzéséhez.* Kutatásom során továbbfejlesztett módszereket javasoltam ipari valósidejű protokollok élőségi verifikációjára.

4.1 *Valósidejű rendszerek k-indukció alapú élőségi ellenőrzése.* Definiáltam a naptárrendszer formalizmust, mely lehetővé teszi egymással kommunikáló valósidejű rendszerek kényelmes modellezését. Kiterjesztettem a k-indukció alapú modellellenőrzést naptárrendszerek mind biztonságossági, mind élőségi tulajdonságainak ellenőrzésére transzformációs lépések egy sorozata által. Ezen felül eszköztámogatást adtam a k-indukció alapú automatikus verifikáció sikerességét elősegítő lemmák modellből történő származtatására.

4.2 *Dekompozíciós módszer hierarchikus valósidejű protokollok élőségi ellenőrzésére.* Generikus dekompozíciós sémát javasoltam funkcionalitásukban hierarchikus szerkezetű valósidejű rendszerek verifikációjára. A módszer biztonságossági és élőségi tulajdonságok kombinációjának ellenőrzésekor alkalmazható.

A módszereket sikeresen alkalmaztuk egy elosztott SCADA (Supervisory Control and Data Acquisition) rendszeren belüli kommunikáció megbízhatóságát biztosítani hivatott biztonságkritikus protokoll verifikációja során. A 4. tézis eredményeit az értekezés 7. és 8. fejezete tárgyalja. A tézishez az alábbi publikációk kapcsolódnak: [*c3*; *c5*] .

3. Az új eredmények gyakorlati felhasználása

Az 1-3. tézisben javasolt algoritmusok a Theta[1] nyílt forráskódú modellellenőrző keretrendszerben kerültek implementálásra. A vonatkozó alkalmazások az alábbiak.

- A Theta eszközt sikeresen alkalmaztuk egy, a CERN-nel folytatott *ipari együttműködésben* PLC forráskódok biztonságossági ellenőrzésére.
- A Theta eszköz a tanszéki *oktatásban* mind bizonyos labortárgyak, mind önálló hallgatói munkák keretében intenzív használatban van, mint formális ellenőrző eszközök gyors prototipizálásának alapja.
- A Theta eszközhöz kapcsolódóan jelenleg is folyik (e disszertációhoz nem kapcsolódó) *kutatás*[2].
  - A Theta eszköz verifikációs backendként integrálva lett az állapottérkép-alapú modellezést és modellintegrációt támogató Gamma [Mol+18] eszközbe. Ezáltal lehetővé vált a Gamma eszközben modellezett vasúti biztosítóberendezés bizonyos protokolljainak formális verifikációja.
  - C programok ellenőrzésére szolgál a Gazer-Theta eszköz [ÁSH21]. Az eszköz részt vett a „10th International Competition on Software Verification” (SV-COMP 2021) [Bey21] versenyen, ahol 9 alkategóriában versenyzett.

A 4. tézis eredményei a többek között vasútautomatizálási szoftverek fejlesztésével foglalkozó Prolan Zrt-vel folytatott együttműködés során keletkeztek. Módszereink egy, a legmagasabb, SIL 4-es biztonságintegritási szinten (safety integrity level) alkalmazandó, fejlesztés alatt álló SCADA (Supervisory Control and Data Acquisition) rendszer egyik kommunikációs protokolljának ellenőrzése során kerültek alkalmazásra.

[1] https://github.com/FTSRG/theta
[2] A vonatkozó publikációk teljes listája elérhető az alábbi linken: https://ftsrg.mit.bme.hu/theta/publications/.

4. Publikációs lista

Publikációk száma:	16
Angol nyelvű lektorált folyóiratcikkek száma:	2
WoS vagy Scopus által indexelt folyóiratcikkek száma:	2
Angol nyelvű publikációk a szerző legalább 50%-os hozzájárulásával:	9
Lektorált publikációk száma:	16
Független hivatkozások száma:	9

4.1. Tézisekhez kapcsolt publikációk

	Folyóirat-cikkek	Nemzetközi konferenciacikkek	Helyi cikkek	Kutatási jelentések
1. tézis	[*j*2]*	[*c*6]; [*c*10]	—	—
2. tézis	[*j*2]*	[*c*8]†; [*c*9]	—	—
3. tézis	[*j*1]; [*j*2]*	[*c*4]; [*c*7]†; [*c*11]; [*c*12]	[*e*13]	—
4. tézis	—	[*c*3]; [*c*5]	—	—

* A megjelölt publikációk több tézishez is kapcsolódnak.
† A BME Méréstechnika és Információs Rendszerek által szervezett Doktorandusz Miniszimpózium a 2016-os, valamint a 2017-es évben nemzetközi részvételű volt.
A cikkek besorolása a doktori iskola publikációs pontozásához használt besorolást követi.

Folyóiratcikkek

[*j*1] Tamás Tóth és István Majzik. Formal verification of real-time systems with data processing. *Periodica Polytechnica Electrical Engineering and Computer Science* 61(2), 2017, 166–174. old. DOI: 10.3311/PPee.9766.

[*j*2] Tamás Tóth és István Majzik. Configurable verification of timed automata with discrete variables. *Acta Informatica* (online first), 2020. DOI: 10.1007/s00236-020-00393-4.

Nemzetközi konferenciacikkek

[*c*3] Tamás Tóth, András Vörös és István Majzik. K-induction based verification of real-time safety critical systems. In: *Proceedings of the 8th International Conference on Dependability and Complex Systems, DepCoS-RELCOMEX 2013*, AISC, 224. köt., 469–478. old. Springer, 2013. DOI: 10.1007/978-3-319-00945-2_43.
▷ *Saját kontribúciók (1) a naptárrendszer formalizmus (2) a modellellenőrzési módszer (3) az eszköztámogatás implementációja (4) az esettanulmány modellezése és verifikációja.*

[*c*4] Tamás Tóth, András Vörös és István Majzik. Verification of a real-time safety-critical protocol using a modelling language with formal data and behaviour semantics. In: *Computer Safety, Reliability, and Security. SAFECOMP 2014 Workshops*, LNCS, 8696. köt., 207–218. old. Springer, 2014. DOI: 10.1007/978-3-319-10557-4_24.
▷ *Saját kontribúciók (1) a modellezési formalizmus (2) a modellellenőrzési módszer (3) az eszköztámogatás implementációja (4) az esettanulmány modellezése és verifikációja.*

[*c*5] Tamás Tóth, András Vörös és István Majzik. A decomposition method for the verification of a real-time safety-critical protocol. In: *Software Engineering for Resilient Systems. 7th International Workshop, SERENE 2015*, LNCS, 9274. köt., 31–45. old. Springer, 2015. DOI:

10.1007/978-3-319-23129-7_3.
▷ *Saját kontribúciók (1) a modellellenőrzési módszer (2) az esettanulmány modellezése és verifikációja.*

[*c*6] Ákos Hajdu, Tamás Tóth, András Vörös és István Majzik. A configurable CEGAR framework with interpolation-based refinements. In: *Formal Techniques for Distributed Objects, Components, and Systems. 36th IFIP WG 6.1 International Conference, FORTE 2016*, LNCS, 9688. köt., 158–174. old. Springer, 2016. DOI: 10.1007/978-3-319-39570-8_11.
▷ *Saját kontribúciók (1) néhány meglátás a modellellenőrzési módszerrel kapcsolatban (2) az eszköztámogatás részleges implementációja.*

[*c*7] Tamás Tóth és István Majzik. Formal modeling of real-time systems with data processing. In: *Proceedings of the 23rd PhD Mini-Symposium*, 46–49. old. BME Department of Measurement és Information Systems. Accommodated by IEEE Hungary, 2016.

[*c*8] Tamás Tóth és István Majzik. Timed automata verification using interpolants. In: *Proceedings of the 24th PhD Mini-Symposium*, 82–85. old. BME Department of Measurement és Information Systems, 2017. DOI: 10.5281/zenodo.291907.

[*c*9] Tamás Tóth és István Majzik. Lazy reachability checking for timed automata using interpolants. In: *Formal Modeling and Analysis of Timed Systems. 15th International Conference, FORMATS 2017*, LNCS, 10419. köt., 264–280. old. Springer, 2017. DOI: 10.1007/978-3-319-65765-3_15.

[*c*10] Tamás Tóth, Ákos Hajdu, András Vörös, Zoltán Micskei és István Majzik. THETA: a framework for abstraction refinement-based model checking. In: *Proceedings of the 17th Conference on Formal Methods in Computer Aided Design, FMCAD 2017*, 176–179. old. FMCAD Inc., 2017. DOI: 10.23919/FMCAD.2017.8102257.
▷ *Saját kontribúciók (1) a keretrendszer architektúrájának, interfészeinek és generikus algoritmikus komponenseinek megtervezése (2) az eszköztámogatás részleges implementációja.*

[*c*11] Tamás Tóth és István Majzik. Lazy reachability checking for timed automata with discrete variables. In: *Model Checking Software. 25th International Symposium, SPIN 2018*, LNCS, 10869. köt., 235–254. old. Springer, 2018. DOI: 10.1007/978-3-319-94111-0_14.

[*c*12] Rebeka Farkas, Tamás Tóth, Ákos Hajdu és András Vörös. Backward reachability analysis for timed automata with data variables. In: *Automated Verification of Critical Systems*, Electronic Communications of the EASST, 76. köt., 1–20. old. 2018. DOI: 10.14279/tuj.eceasst.76.1076.
▷ *Saját kontribúciók (1) néhány meglátás a modellellenőrzési módszerrel kapcsolatban (2) az eszköztámogatás részleges implementációja.*

Helyi konferenciacikkek

[*e*13] Tamás Tóth és István Majzik. A framework for formal verification of real-time systems. In: *Proceedings of the 22nd PhD Mini-Symposium*, 12–13. old. BME Department of Measurement és Information Systems. Accommodated by IEEE Hungary, 2015.

Kiegészítő anyagok

[*s*14] Tamás Tóth és István Majzik. Supplementary Material for the Paper "Configurable Verification of Timed Automata with Discrete Variables". Zenodo. 2020. DOI: 10.5281/zenodo.3965792.

4.2. Tézisekhez nem kapcsolt publikációk

Nemzetközi konferenciacikkek

[c15] Gyula Sallai és Tamás Tóth. Boosting software verification with compiler optimizations. In: *Proceedings of the 24th PhD Mini-Symposium*, 66–69. old. BME Department of Measurement és Information Systems, 2017. DOI: 10.5281/zenodo.291903.

[c16] Bence Czipó, Ákos Hajdu, Tamás Tóth és István Majzik. Exploiting hierarchy in the abstraction-based verification of statecharts using SMT solvers. In: *International Workshop on Formal Engineering approaches to Software Components and Architectures, FESCA 2017*, EPTCS, 245. köt., 31–45. old. Open Publishing Association, 2017. DOI: 10.4204/EPTCS.245.3.

[c17] Gyula Sallai, Ákos Hajdu, Tamás Tóth és Zoltán Micskei. Towards evaluating size reduction techniques for software model checking. In: *Fifth International Workshop on Verification and Program Transformation, VPT 2017*, EPTCS, 253. köt., 75–91. old. Open Publishing Association, 2017. DOI: 10.4204/EPTCS.253.7.

Hivatkozások

[AD94] Rajeev Alur és David L. Dill. A theory of timed automata. *Theoretical Computer Science* 126(2), 1994, 183–235. old. DOI: 10.1016/0304-3975(94)90010-8.

[AGC12] Aws Albarghouthi, Arie Gurfinkel és Marsha Chechik. Craig interpretation. In: *Static Analysis*, LNCS, 7460. köt., 300–316. old. Springer, 2012. DOI: 10.1007/978-3-642-33125-1_21.

[Alb+12] Aws Albarghouthi, Yi Li, Arie Gurfinkel és Marsha Chechik. UFO: a framework for abstraction- and interpolation-based software verification. In: *Computer Aided Verification*, LNCS, 7358. köt., 672–678. old. Springer, 2012. DOI: 10.1007/978-3-642-31424-7_48.

[ÁSH21] Zsófia Ádám, Gyula Sallai és Ákos Hajdu. GAZER-THETA: LLVM-based verifier portfolio with BMC/CEGAR (competition contribution). In: *Tools and Algorithms for the Construction and Analysis of Systems*, LNCS, 12652. köt., 433–437. old. Springer, 2021. DOI: 10.1007/978-3-030-72013-1_27.

[BC00] Per Bjesse és Koen Claessen. SAT-based verification without state space traversal. In: *Formal Methods in Computer-Aided Design*, LNCS, 1954. köt., 409–426. old. Springer, 2000. DOI: 10.1007/3-540-40922-X_23.

[Beh+03] Gerd Behrmann, Patricia Bouyer, Emmanuel Fleury és Kim G. Larsen. Static guard analysis in timed automata verification. In: *Tools and Algorithms for the Construction and Analysis of Systems*, LNCS, 2619. köt., 254–270. old. Springer, 2003. DOI: 10.1007/3-540-36577-X_18.

[Beh+04] Gerd Behrmann, Patricia Bouyer, Kim G. Larsen és Radek Pelánek. Lower and upper bounds in zone based abstractions of timed automata. In: *Tools and Algorithms for the Construction and Analysis of Systems*, LNCS, 2988. köt., 312–326. old. Springer, 2004. DOI: 10.1007/978-3-540-24730-2_25.

[Bey+07] Dirk Beyer, Thomas A. Henzinger, Ranjit Jhala és Rupak Majumdar. The software model checker BLAST. *Software Tools for Technology Transfer* 9(5), 2007, 505–525. old. DOI: 10.1007/s10009-007-0044-z.

[Bey21] Dirk Beyer. Software verification: 10th comparative evaluation (SV-COMP 2021). In: *Tools and Algorithms for the Construction and Analysis of Systems*, LNCS, 12652. köt., 401–422. old. Springer, 2021. DOI: 10.1007/978-3-030-72013-1_24.

[BK11] Dirk Beyer és M. Erkan Keremoglu. CPAchecker: a tool for configurable software verification. In: *Computer Aided Verification*, LNCS, 6806. köt., 184–190. old. Springer, 2011. DOI: 10.1007/978-3-642-22110-1_16.

[BLR05] Patricia Bouyer, François Laroussinie és Pierre-Alain Reynier. Diagonal constraints in timed automata: forward analysis of timed systems. In: *Formal Modelling and Analysis of Timed Systems*, LNCS, 3829. köt., 112–126. old. Springer, 2005. DOI: 10.1007/11603009_10.

[Bou04] Patricia Bouyer. Forward analysis of updatable timed automata. *Formal Methods in System Design* 24(3), 2004, 281–320. old. DOI: 10.1023/B:FORM.0000026093.21513.31.

[BPR01] Thomas Ball, Andreas Podelski és Sriram K. Rajamani. Boolean and cartesian abstraction for model checking C programs. In: *Tools and Algorithms for the Construction and Analysis of Systems*, LNCS, 2031. köt., 268–283. old. Springer, 2001. DOI: 10.1007/3-540-45319-9_19.

[Bur+92] Jerry R. Burch, Edmund L. Clarke, Kenneth L. McMillan, David L. Dill és L. J. Hwang. Symbolic model checking: 10^{20} states and beyond. *Information and Computation* 98(2), 1992, 142–170. old. DOI: 10.1016/0890-5401(92)90017-A.

[CGL94] Edmund M. Clarke, Orna Grumberg és David E. Long. Model checking and abstraction. *Transactions on Programming Languages and Systems* 16(5), 1994, 1512–1542. old. DOI: 10.1145/186025.186051.

[CGS04] Edmund M. Clarke, Anubhav Gupta és Ofer Strichman. SAT-based counterexample-guided abstraction refinement. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 23(7), 2004, 1113–1123. old. DOI: 10.1109/TCAD.2004.829807.

[Cha+02] Pankaj Chauhan, Edmund M. Clarke, James Kukula, Samir Sapra, Helmut Veith és Dong Wang. Automated abstraction refinement for model checking large state spaces using SAT based conflict analysis. In: *Formal Methods in Computer-Aided Design*, LNCS, 2517. köt., 33–51. old. Springer, 2002. DOI: 10.1007/3-540-36126-X_3.

[Cla+00] Edmund M. Clarke, Orna Grumberg, Somesh Jha, Yuan Lu és Helmut Veith. Counterexample-guided abstraction refinement. In: *Computer Aided Verification*, LNCS, 1855. köt., 154–169. old. Springer, 2000. DOI: 10.1007/10722167_15.

[Cla+03] Edmund M. Clarke, Orna Grumberg, Somesh Jha, Yuan Lu és Helmut Veith. Counterexample-guided abstraction refinement for symbolic model checking. *Journal of the ACM* 50(5), 2003, 752–794. old. DOI: 10.1145/876638.876643.

[Cla+05] Edmund Clarke, Daniel Kroening, Natasha Sharygina és Karen Yorav. SatAbs: SAT-based predicate abstraction for ANSI-C. In: *Tools and Algorithms for the Construction and Analysis of Systems*, LNCS, 3440. köt., 570–574. old. Springer, 2005. DOI: 10.1007/978-3-540-31980-1_40.

[DT98] Conrado Daws és Stavros Tripakis. Model checking of real-time reachability properties using abstractions. In: *Tools and Algorithms for the Construction and Analysis of Systems*, LNCS, 1384. köt., 313–329. old. Springer, 1998. DOI: 10.1007/BFb0054180.

[EC82] E. Allen Emerson és Edmund M. Clarke. Using branching time temporal logic to synthesize synchronization skeletons. *Science of Computer Programming* 2(3), 1982, 241–266. old. DOI: 10.1016/0167-6423(83)90017-5.

[ES03] Niklas Eén és Niklas Sörensson. Temporal induction by incremental SAT solving. *Electronic Notes in Theoretical Computer Science* 89(4), 2003, 543–560. old. DOI: 10.1016/S1571-0661(05)82542-3.

[Gru06] Orna Grumberg. Abstraction and refinement in model checking. In: *Formal Methods for Components and Objects*, LNCS, 4111. köt., 219–242. old. Springer, 2006. DOI: 10.1007/11804192_11.

[Gul+08] Bhargav S. Gulavani, Supratik Chakraborty, Aditya V. Nori és Sriram K. Rajamani. Automatically refining abstract interpretations. In: *Tools and Algorithms for the Construction and Analysis of Systems*, LNCS, 4963. köt., 443–458. old. Springer, 2008. DOI: 10.1007/978-3-540-78800-3_33.

[Her+11] Frédéric Herbreteau, Dileep Kini, Balaguru Srivathsan és Igor Walukiewicz. Using non-convex approximations for efficient analysis of timed automata. In: *Foundations of Software Technology and Theoretical Computer Science*, LIPIcs, 13. köt., 78–89. old. Dagstuhl, 2011. DOI: 10.4230/LIPIcs.FSTTCS.2011.78.

[HSW12] Frédéric Herbreteau, B. Srivathsan és Igor Walukiewicz. Better abstractions for timed automata. In: *Logic in Computer Science*, 375–384. old. IEEE, 2012. DOI: 10.1109/LICS.2012.48.

[HSW13] Frédéric Herbreteau, Balaguru Srivathsan és Igor Walukiewicz. Lazy abstractions for timed automata. In: *Computer Aided Verification*, LNCS, 8044. köt., 990–1005. old. Springer, 2013. DOI: 10.1007/978-3-642-39799-8_71.

[Kan+15] Gijs Kant, Alfons Laarman, Jeroen Meijer, Jaco van de Pol, Stefan Blom és Tom van Dijk. LTSMIN: high-performance language-independent model checking. In: *Tools and Algorithms for the Construction and Analysis of Systems*, LNCS, 9035. köt., 692–707. old. Springer, 2015. DOI: 10.1007/978-3-662-46681-0_61.

[Kur94] Robert P. Kurshan. *Computer-Aided Verification of Coordinating Processes: The Automata-Theoretic Approach*. Princeton University Press, 1994. DOI: 10.1515/9781400864041.

[KW11] Daniel Kroening és Georg Weissenbacher. Interpolation-based software verification with WOLVERINE. In: *Computer Aided Verification*, LNCS, 6806. köt., 573–578. old. Springer, 2011. DOI: 10.1007/978-3-642-22110-1_45.

[McM03] Kenneth L. McMillan. Interpolation and SAT-based model checking. In: *Computer Aided Verification*, LNCS, 2725. köt., 1–13. old. Springer, 2003. DOI: 10.1007/978-3-540-45069-6_1.

[McM06] Kenneth L. McMillan. Lazy abstraction with interpolants. In: *Computer Aided Verification*, LNCS, 4144. köt., 123–136. old. Springer, 2006. DOI: 10.1007/11817963_14.

[Mol+18] Vince Molnár, Bence Graics, András Vörös, István Majzik és Dániel Varró. The GAMMA statechart composition framework: design, verification and code generation for component-based reactive systems. In: *International Conference on Software Engineering*, 113–116. old. ACM, 2018. DOI: 10.1145/3183440.3183489.

[MRS03] Leonardo de Moura, Harald Rueß és Maria Sorea. Bounded model checking and induction: from refutation to verification. In: *Computer Aided Verification*, LNCS, 2725. köt., 14–26. old. Springer, 2003. DOI: 10.1007/978-3-540-45069-6_2.

[QS82] Jean-Pierre Queille és Joseph Sifakis. Specification and verification of concurrent systems in CESAR. In: *International Symposium on Programming*, LNCS, 137. köt., 337–351. old. Springer, 1982. DOI: 10.1007/3-540-11494-7_22.

[SSS00] Mary Sheeran, Satnam Singh és Gunnar Stålmarck. Checking safety properties using induction and a SAT-solver. In: *Formal Methods in Computer-Aided Design*, LNCS, 1954. köt., 127–144. old. Springer, 2000. DOI: 10.1007/3-540-40922-X_8.