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

K-indukciós algoritmus fejlesztése a Theta verifikációs keretrendszerben

Szakdolgozat

Készítette
Jakab Richárd Benjámin

Konzulens
Dr. Vörös András

2020. november 30.

Tartalomjegyzék

HALLGATÓI NYILATKOZAT

Alulírott *Jakab Richárd Benjámin*, szigorló hallgató kijelentem, hogy ezt a szakdolgozatot meg nem engedett segítség nélkül, saját magam készítettem, csak a megadott forrásokat (szakirodalom, eszközök stb.) használtam fel. Minden olyan részt, melyet szó szerint, vagy azonos értelemben, de átfogalmazva más forrásból átvettem, egyértelműen, a forrás megadásával megjelöltem.

Hozzájárulok, hogy a jelen munkám alapadatait (szerző(k), cím, angol és magyar nyelvű tartalmi kivonat, készítés éve, konzulens(ek) neve) a BME VIK nyilvánosan hozzáférhető elektronikus formában, a munka teljes szövegét pedig az egyetem belső hálózatán keresztül (vagy autentikált felhasználók számára) közzétegye. Kijelentem, hogy a benyújtott munka és annak elektronikus verziója megegyezik. Dékáni engedéllyel titkosított diplomatervek esetén a dolgozat szövege csak 3 év eltelte után válik hozzáférhetővé.

Budapest, 2020. november 30.

Jakab Richárd Benjámin
hallgató

Kivonat

Jelen dokumentum egy diplomaterv sablon, amely formai keretet ad a BME Villamosmérnöki és Informatikai Karán végző hallgatók által elkészítendő szakdolgozatnak és diplomatervnek. A sablon használata opcionális. Ez a sablon LaTeX alapú, a *TeXLive* TeX-implementációval és a PDF-LaTeX fordítóval működőképes.

Abstract

This document is a LaTeX-based skeleton for BSc/MSc theses of students at the Electrical Engineering and Informatics Faculty, Budapest University of Technology and Economics. The usage of this skeleton is optional. It has been tested with the *TeXLive* TeX implementation, and it requires the PDF-LaTeX compiler.

1. fejezet

Bevezetés

A körülöttünk lévő világban számos helyen találunk olyan informatikai rendszereket, melyeknél a meghibásodás (hibás működés) következménye elfogadhatatlan. Hagyományosan ilyen területek az egészségügyi alkalmazások, légi közlekedés, atomenergia ipar, fegyverrendszerek stb. Ezeket a rendszereket biztonságkritikus rendszereknek nevezzük, és létfontosságú a specifikációnak megfelelő működés ellenőrzése. Ennek ellenőrzésére különböző verifikációs technikák szolgálnak. Ezek egyike a modellellenőrzés, mey során a rendszer egy matematikai modelljét vizsgálva lehet különböző formalizált követelmények teljesülését ellenőrizni.

A munkám során a BME VIK Méréstechnika és Információs Rendszerek Tanszék[1] Hibatűrő Rendszerek Kutatócsoportja[2] által fejlesztett *Theta*[3] keretrendszert használtam. A *Theta* egy általános célú, moduláris és konfigurálható modellellenőrző keretrendszer, melyet absztrakciós finomításon alapuló algoritmusok tervezésének és értékelésének támogatására hoztak létre a különböző formalizmusok elérhetőségi elemzéséhez.

A munkámat három részre tagolhatjuk, melyet a szakdolgozatom felépítése is követ: először elmerültem a szoftververifikáció és modellezés tématerületében, kiemelten foglalkozva a k-indukció alapú szoftververifikációval, aztán a szakirodalom által bemutatott algoritmust leimplementáltam a *Theta* keretrendszerben, majd végezetül széleskörű tesztelés alá vetettem és így finomítottam az algoritmusomat.

A dolgozat az alábbi részletesebb tartalmi felosztásban tárgyalja a fentebb felvázolt folyamatot:

- A második fejezetben a szoftververifikációt mutatom be általános megközelítésben
- A harmadik fejezetben kifejtem az algoritmusom alapját adó K-indukció módszer elméleti hátterét
- A negyedik fejezetben bemutatom az algoritmusom elkészítésének folyamatait és technikai felépítését
- Az ötödik fejezetben bemutatom az algoritmusom teszteredményeit és összehasonlítom más, verifikáló algoritmusokkal

TODO: fejezet lista frissítése

[1] `https://www.mit.bme.hu/`
[2] `https://www.mit.bme.hu/research/ftsrg`
[3] `https://github.com/FTSRG/theta`

2. fejezet

Szoftververifikáció

3. fejezet

K-indukció

Az elkészített programom a k-indukció nevű matematikai módszerre [2] támaszkodik. Ebben a fejezetben ismertetem az elméleti módszert, illetve pszeudokód szinten bemutatom az ezen a matematikai módszeren alapuló algoritmus működését és bemutatom annak helyességét.

3.1. Elméleti háttér

Tekintsük az alább látható teljes indukció tételét a természetes számok halmaza fölött (kiegészítve 0-val):

$$P(0) \wedge \forall n(P(n) \Rightarrow P(n+1)) \Rightarrow \forall n P(n). \tag{3.1}$$

Lényege, hogy megnézzük az első lépésre teljesül-e a feltétel (az angol szakirodalomban ez a *base-case*). Ha igen, akkor megnézzük ennek tudatában azt, hogy az $n+1$. lépés következik-e az n. lépésből (indukciós lépés – *induction case*). Ha sikerül ezt belátnunk, akkor készen vagyunk, bebizonyítottuk az összes lépésre a feltételt.

Ezt tovább gondolva megtehetjük azt, hogy az első két lépésre nézzük meg, hogy teljesítik-e a feltételt:

$$P(0) \wedge P(1) \wedge \forall n((P(n) \wedge P(n+1)) \Rightarrow P(n+2)) \Rightarrow \forall n P(n). \tag{3.2}$$

Ezt az elvet általánosíthatjuk k lépésre, $k \geq 1$, melyet a irodalom [2] k-indukciónak nevez, formálisan:[1]

$$\left(\bigwedge_{i=0}^{k-1} P(i)\right) \wedge \forall n \left(\left(\bigwedge_{i=0}^{k-1} P(n+i)\right) \Rightarrow P(n+k)\right) \Rightarrow \forall n P(n). \tag{3.3}$$

3.2. A probléma formalizálása

Ahhoz, hogy a problémát precízebben megfogalmazhassuk, szükség van jelölések és fogalmak bevezetésére. Adott egy tranzakciós relációkból felépülő gráf, melyben $T(x,y)$-al jelöljük azt, ha létezik egy, az x állapotból az y állapotba mutató tranzakciós reláció [1]. Így már tudjuk definiálni az útvonal fogalmát, mely állapotok sorozatát jelenti T-n keresztül:

$$utvonal(s_{[0..n]}) \doteq \bigwedge_{0 \leq i < n} T(s_i, s_{i+1}) \tag{3.4}$$

[1] A k-indukció helyességének a bizonyítására a dolgozatomban nem térek ki.

Ahol $s_{[0..n]}$ rövidítés az $(s_0, s_1, \ldots, s_n)$ állapotsorozatot jelöli. Az *utvonal* n hosszúságú, ha n darab tranzakcióból áll. A nulla hosszúságú *utvonal* egy darab állapotot tartalmaz és nem értelmezzük rajta a tranzakció műveletét. Azt a megállapítást, hogy egy Q tulajdonság igaz egy útvonal összes állapotára, úgy fogjuk írni, hogy $\forall.Q(s_{[0..n]})$.

A továbbiakban lesz olyan, mikor egy útvonal alatt nem csak azt értjük, hogy az tranzakciók sorozata, hanem annak létezését is jelöli. Így, $utvonal_i(s_0, s_i)$ alatt azt jelöljük, hogy *létezik* egy útvonal s_0-ból s_i-be, mely i darab T-ből áll.

Legyen T egy tranzakciós reláció S állapothalmazán. Feltételezzük, hogy T a teljes állapottérre értelmezve van, tehát minden állapotnak (a kezdőállapotokat leszámítva) van egy szülőállapota T-n keresztül. Jelöljük I-vel a kezdőállapotokat, és azt vizsgáljuk, hogy az állapotok teljesítik-e a P tulajdonságot.

A problémát informálisan a következőképp foglalhatjuk össze: beszeretnénk azt látni, hogy ha egy kezdőállapotból elindulunk, akkor a tranzakciós relációt ismétlődően alkalmazva csak olyan állapotba fogunk eljutni, mely kielégíti P-t. Formálisan a következőt akarjuk belátni:

$$\forall i: \ \forall s_0 \ldots s_i: \ (I(s_0) \wedge utvonal(s_{[0..i]}) \rightarrow P(s_i)) \tag{3.5}$$

Ahol $i \geq 0$. Később látni fogjuk, hogy az algoritmus felhasználja ennek a megfordítottját is: a „rossz” állapotokból (hibaállapotokból) elindulunk visszafelé, és azt vizsgáljuk, hogy elérjük-e valamelyik kezdőállapotot:

$$\forall i: \ \forall s_0 \ldots s_i: \ (\neg I(s_0) \leftarrow utvonal(s_{[0..i]}) \wedge \neg P(s_i)) \tag{3.6}$$

Ahol $\neg I(s_0)$ azt jelenti, hogy s_0 nem kezdőállapot (nem teljesíti a „kezdőállapot tulajdonságot”), illetve $\neg P(s_i)$ azt, hogy s_i nem elégíti ki a P tulajdonságot. A két egyenlet ekvivalens és összetehetőek úgy, hogy azon a probléma szemléletesebb és szimmetrikusabb legyen:

$$\forall i: \ \forall s_0 \ldots s_i: \ \neg(I(s_0) \wedge utvonal(s_{[0..i]}) \wedge \neg P(s_i)) \tag{3.7}$$

Azaz szavakkal elmondva – azt akarjuk megmutatni, hogy *nem létezik* olyan útvonal, mely kezdőállapotból indul és egy *nem-P* állapotba jut.

3.3. A formalizált algoritmus

A fenti ismeretek segítségével hogy tudnánk belátni, hogy a modell a P tulajdonságra nézve biztonságos?

Például úgy, ha megnézzük, hogy a

$$\forall s_0 \ldots s_i: \ \neg(I(s_0) \wedge utvonal(s_{[0..i]}) \wedge \neg P(s_i)) \tag{3.8}$$

teljesül $i = 0$, $i = 1$, $i = 2$, … -re. Ha feltétel megsérül valamelyik elérhető állapotban, akkor ezzel a módszerrel meg fogjuk találni és az oda vezető útvonal ellenpélda lesz a modell P-biztonságosságára. Ez egy elvárható eredmény: az algoritmusnak két féle kimenetele kell, hogy legyen: vagy az, hogy a modell P-tulajdonságra nézve biztonságos (minden elérhető állapot teljesíti), vagy az, hogy a modell nem P-biztonságos, ekkor egy ellenpéldát is kell adnia, mely bizonyítja, hogy az adott kezdőállapotból elindulva, azon végighaladva valóban egy hibaállapotba kerülünk.

Ha a rendszer P-biztonságos, akkor (3.8) minden i-re igaz lesz, hiszen nem fogunk tudni találni olyan i értéket, melyre ne teljesülne. Felvetődhet a kérdés, hogy mikortól lehet azt mondani, hogy i további növelése céltalan, mert már teljes bizonyossággal kijelenthetjük, hogy a modell P-biztonságos? A $I(s_0) \wedge utvonal(s_{[0..i]})$ feltétel önmagában nem fog gyorsítást eredményezni: vagy végig megy az állapottéren amilyen hosszan csak lehetséges (ezt szeretnénk lerövidíteni), tekintve, hogy minden állapotnak van egy szülőállapota a T tranzakciós reláción keresztül, vagy végtelen ciklusba kerül.

```
Result:
i=0;
while True do
    instructions;
    if condition then
        instructions1;
        instructions2;
    else
        instructions3;
    end
end
```

Algorithm 1: Checking if model is P-safe

4. fejezet

Implementálás

4.1. Adatszerkezet

5. fejezet

Benchmark

5.1. Teszteredmények

6. fejezet

Összefoglaló

Köszönetnyilvánítás

Irodalomjegyzék

[1] Mary Sheeran – Satnam Singh – Gunnar Stålmarck: Checking safety properties using induction and a sat-solver. In Warren A. Hunt – Steven D. Johnson (szerk.): *Formal Methods in Computer-Aided Design* (konferenciaanyag). Berlin, Heidelberg, 2000, Springer Berlin Heidelberg, 127–144. p. ISBN 978-3-540-40922-9.

[2] Thomas Wahl: The k-induction principle. URL `http://www.comlab.ox.ac.uk/people/Thomas.Wahl/Publications/k-induction.pdf`.