Válasz opponensi bírálatra

Opponens: Levendoszky János, egyetemi tanár, az MTA doktora

MTA értekezés címe:
Precíz modell transzformációk tervezése és analízise a modellvezérelt fejlesztésben

Hadd köszönjem meg Levendovszky János Professzor Úr opponensi értékelését, amelyben igen mély, szemantikai kérdéseket tesz fel az értekezéssel kapcsolatban. Nagy megtiszteltetés számomra, hogy a modelltranszformációk területén elvégzett kutatásaim kimenetét magas szintű új tudományos eredménynek értékeli.

A következőkben a bírálat 3. oldalán feltett kérdésekre térek ki részletesen, egyúttal teljes mértékben elfogadva az Opponensem által az 1. oldalon megfogalmazott apróbb szerkesztési kritikát.

---

1. kérdés: Az első tézisben mi volt konkrétan a szerző saját munkája a kutatói csapat többi tagjához képest?

Az első tézisben a tetszőleges mélységű negációt és rekurziót támogató kompozícionális gráfminta alapú lekérdezőnyelvet és a ráépülő modelltranszformációs nyelvet Balogh Andrással közösen dolgoztuk ki és publikáltuk [K1,J14]. A saját kontribúcióm a gráfminták, mint önálló lekérdezőnyelv bevezetése (1/1. altézis), valamint a tetszőleges mélységű negáció és rekurzió szemantikai kérdéseinek precíz, formális specifikációja. A gráftranszformációs szabályok és absztrakt állapotgépek formalizmusának ötvözésével megalkotott modelltranszformációs nyelv alapötlete és formalizálása szintén önálló kontribúcióm (1/2. altézis). Balogh András PhD disszertációjának fő eredménye a kompozícionalitás és az újrafelhasználhatóság kérdésének vizsgálata mind a gráfminták, mind a transzformációs szabályok szintjén.

A generikus és meta-transzformációk (1/3. altézis) alapötletét PhD disszertációm témavezetőjével, Pataricza András professzorral együtt dolgoztuk ki és publikáltuk [K25]. Itt témavezetőm kontribúciója döntően a meta-transzformációk kidolgozása volt, én pedig a generikus transzformációkat vizsgáltam.

Az 1. téziscsoporthoz kapcsolódó műszaki eredmény a VIATRA2 modelltranszformációs keretrendszer, amelynek alapítója és kutatási vezetője voltam 2004-től kezdve. E szoftverrendszer közvetlen fejlesztésében több mint 10 további kutató vett részt, akik munkáját ezúton is szeretném megköszönni.

---

2. kérdés: A programozás-példák-alapján paradigma analízise (második tézis), csak a módszer leírására fókuszál és a példák generálásának automatizmusára, illetve az interaktív megközelítés szükségességére összpontosít. Nem esik szó a generálandó példák számának meghatározásáról, amely garantálja a megfelelő általánosítási képességet. Felhasználhatók-e itt a gépi tanuláselmélet eredményei (Vapnik-Chervonenkis dimenzió…etc.) a szükséges példák számának a becsléséhez, amellyel megfelelő általánosítást lehet elérni?

A tanulhatóság Leslie Valiant által javasolt PAC (Probably Approximately Correct) modellje [Valiant84] lehetővé teszi, hogy precízen vizsgálhassuk a gépi tanulás komplexitását. A tanuló algoritmusnak (fix, de ismeretlen eloszlású) példákból kiindulva kell egy általánosított hipotézist felállítania tetszőleges

(előre rögzített) precizitással és közelítéssel, ekkor nevezzük a hipotézisek egy osztályát PAC-tanulhatónak. A Vapnik-Chervonenkis (VC) dimenzió [VC71] és a kapcsolódó Sauer lemma segítségével becslést adhatunk az adott pontosság és precizitás eléréséhez szükséges példák számára [Blumer+89]. Amennyiben a VC dimenzió véges, úgy gyors konvergencia várható a hipotézis eléréséhez.

A második tézisben az induktív logikai programozás (ILP) eszközkészletét felhasználva tettem javaslatot a transzformációs szabályok automatikus származtatására példák alapján. Egyetértve Bírálóm megjegyzésével, e téziscsoportot döntően a szoftvertudomány szempontjából és nem a gépi tanulás tudományterületének szempontjából tárgyaltam. A feltett kérdés megválaszolásához célszerű megvizsgálni, hogy miként kapcsolódik az ILP tudományterülete a gépi tanulás általános elméletéhez.

Muggleton az ILP tudományterületét összegző cikkében [MR94] összefoglalta a PAC tanulhatósághoz kapcsolódó legfőbb eredményeket. Haussler fő eredménye sajnos negatív [Haussler90], mely szerint egzisztenciálisan kvantifikált elsőrendű logikai formulák általánosságban nem PAC-tanulhatók (azaz általánosságban nem várható gyors konvergencia). A kutatások ezután a logikai formalizmusok külön alosztályaira koncentráltak PAC-tanulhatóság terén, részint pozitív [Cohen93], részint negatív eredménnyel [Kietz93]. Baskiotis és kollégáinak legfrissebb eredményei pedig a deduktív-induktív rendszerek egy széles skálájának VC-dimenzióját és PAC-tanulhatóságát tárgyalja [BST2007], legtöbb esetben szintén végtelen eredménnyel a VC-dimenzióra nézve.

Az ILP rendszerek PAC-tanulhatóságának összefoglalásaként elmondható, hogy eldönthetetlen elméletek VC-dimenziója általánosságban végtelen, és csak bizonyos alosztályokra létezhet pozitív és véges VC-dimenzió. Érdekes további kutatást jelenthet a véges VC-dimenzióval rendelkező ILP alosztályok által kezelhető modelltranszformációk kategorizálása.

Az elméleti komplexitáson túl természetesen szintén kiemelt fontosságú lenne esettanulmányokon vizsgálni, hogy a gyakorlati esetekben mekkora példahalmaz szükséges a kidolgozott módszer helyes működéséhez (pl. ha a szakértők inkrementálisan, az előző szabályhalmaz futtatása után definiálnak újabb példákat). E kérdés precíz megválaszolásához további jövőbeli kutatások szükségesek.

[VC71] V.N. Vapnik, A. Chervonenkis: On the uniform convergence of relative frequencies of events to their probabilities. Theor. Probl. and its Appl.

[Valiant84] L. Valiant: A theory of the learnable. Commun. ACM 27, 11 (Nov. 1984) 1134-1142.

[Blumer+89] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth. "Learnability and the Vapnik–Chervonenkis dimension." Journal of the ACM, 36(4):929–865, 1989.

[MR94] Stephen Muggleton, Luc De Raedt: Inductive Logic Programming: Theory and Methods. J. Log. Program. 19/20: 629-679 (1994)

[Haussler90] D. Haussler: Applying Valiant's Learning Framework to AI Concept-Learning Problems. In: Y. Kodratoff, R. Michalski (Eds.), Machine Learning: An Artificial Intelligence Approach, vol. 3 Morgan Kaufmann, San Mateo, CA (1990), pp. 641–669

[Cohen93] W. Cohen: Learnability of Restricted Logic Programs. In S. Muggleton (Ed.), Proceedings of the 3rd International Workshop on Inductive Logic Programming (1993), pp. 41–72

[Kietz93] J.U. Kietz: Some Lower Bounds on the Computational Complexity of Inductive Logic Programming. In: P. Brazdil (Ed.), Proceedings of the 6th European Conference on Machine Learning, Lecture Notes in Artificial Intelligence, vol. 667, Springer-Verlag (1993), pp. 115–123

[BST2007] N. Baskiotis, M. Sebag, O. Teytaud: Inductive-deductive systems: a mathematical logic and statistical learning perspective http://hal.inria.fr/inria-00173259

---

3. kérdés: A 45. oldalon a szerző néhány feltételezést tesz. Itt érdemes lenne kifejteni az "abstraction gap" pontos definícióját, illetve amennyiben ez kvantifikálható, akkor hogyan függ ettől a "usability".

Szoftver alapú rendszerek tervezésének egyik legnagyobb kihívása, hogy a problématér és a megoldási tartomány között igen jelentős szakadék van [FR07]. A modellvezérelt tervezési folyamat során magasszintű modellezési nyelvektől indulva fokozatosan haladunk az alacsonyabb szintű programozási (és verifikációs) nyelvek majd a (még alacsonyabb szintű) végrehajtható gépi utasítások felé. Egy magasszintű nyelv használhatósága (usability) általában jobb, mint egy alacsonyabb szintű nyelvé, hiszen tömörebb specifikáció állítható elő általa.

Egy magasszintű nyelv egy konstrukciójához tipikusan egynél több, alacsonyabb szintű nyelvi konstrukció rendelhető. Például egy UML állapotgép egy állapotának a belőle generált objektum-orientált forráskódban több utasítás felel meg, és egy magasszintű programozási nyelv egy utasításához több gépi kódú utasítást rendel a fordító. Ez a hozzárendelés értelmezhető a szemantikus hézag egyfajta mennyiségi jellemzőjének, de a szoftvertervezés területén nem ismert számomra az *absztrakciós hézag* precíz, kvantitatív definíciója.

Ugyanakkor egy programozási nyelv lefordítása végrehajtható gépi kódba jellegzetesen egy automatikus lépés, mivel a két nyelv között nincs *szemantikai hézag* (formális értelemben mindkettő a Turing gép tulajdonságával és limitációjával rendelkezik). Egy magasszintű modellezési nyelv leképezése egy programozási nyelvbe azonban nem minden esetben automatizálható, mivel hiányos lehet a *rendszer környezetének leírása* (azaz, a rendszer csak adott környezeti peremfeltételek esetén nyújtja a megkívánt szolgáltatást) és *a tervezői döntések rögzítése* (azaz, a rendszermodell többféle módon is leképezhető egy alacsonyabb szintű nyelvre). Az absztrakciós ugrás során az adott alkalmazásterületre jellemző háttértudást tehát beépítjük a végrehajtható, formális algoritmusokba.

Összegzésként elmondható, hogy a bíráló által feltett kérdés, azaz a szemantikus hézag kvantifikálása izgalmas kutatási kérdéseket vet fel, amelyeket a jövőben mindenképpen érdemes alaposabban megvizsgálni.

[FR07] R. France, B. Rumpe: Model-driven Development of Complex Software: A Research Roadmap, Proc. ICSE 2007: International Conference on Software Engineering.

---

4. kérdés: A 66. oldalon a szerző a minimális feszítőfát keresi, amire számos heurisztikus "near-polynomial" komplexitású algoritmus van. Itt konkréten melyik került felhasználásra és mi indikálja ennek az alkalmazását komplexitás szempontjából?

Kutatásaink során először a tradicionális Chu-Liu / Edmonds algoritmusait [33,50] használtuk fel egy irányított gráfban minimális feszítőfa keresésére, amely négyzetes futási idejű.

A keresési terv költsége a hozzá tartozó minimális feszítőfa súlya $w(P) = \sum_{i=1}^{n} \prod_{j=1}^{i} w_j$ ahol w_j a j. él súlya a keresési terv által definiált élsorrend szerint. A minimális (irányított) feszítőfa $w(P)$ súlya

azonban csak egy becslést ad a lokális keresésen alapuló mintaillesztés várható lépésszámára, amely becslés lehet jó is és rossz is. A gyakorlati mérések azt mutatják [BKG07], hogy általában nagy a korreláció a keresési terv jósága és a mintaillesztő által végrehajtott lépések száma között (tehát kis súlyú keresési tervnél kevesebb mintaillesztési lépés szükséges).

Ugyanakkor egy kellően jó keresési terv gyors megtalálása gyakorlati esetekben jobb megoldást jelenthet, mint az optimum költségesebb megkeresése. A 66. oldalon javasolt algoritmus minden lépésben a minimális súlyú irányított éllel bővít (Kruskal irányítatlan gráfokra kidolgozott algoritmusával analóg módon). A Karlsruhei Egyetem kutatói ugyanakkor sikerrel alkalmaztak egy másik közelítő becslést [62], mely szerint a keresési terv összköltségének domináns faktora a legutolsó szorzat, tovább egyszerűsítve az elvégzendő számítást.

[BKG07] Gernot Veit Batz, Moritz Kroll, Rubino Geiß: A First Experimental Evaluation of Search Plan Driven Graph Pattern Matching. AGTIVE 2007: 471-486

---

5. kérdés: A 67. oldalon milyen "ökölszabályt" használt a Fujaba a keresési terv konstrukciójához és mi motiválja ezt a heurisztikát?

Egy keresési feladat megoldása legtöbbször akkor hatékony, ha a keresés döntési terét (állapotterét) leíró keresési fa a levelek felé „terebélyesedik", miközben a gyökér közelében még „karcsú". Így amennyiben rosszul döntünk a gráfminta csomópontjának illesztésekor, kevesebb korábbi döntés hatását kell majd visszavonnunk (backtracking / backjumping). Az egyes gráftranszformációs eszközök is különféle heurisztikákat alkalmaznak a keresési fa karcsúsítására a gráfmintaillesztés fázisában.

A gráfmintaillesztés egy olyan keresési feladat, amelyben általában kétféle műveletet használunk: egy ellenőrzés (check) jellegűt (pl. két illesztett csomópont között vezet-e a megkívánt típusú él) és egy kiterjesztés (extend) jellegűt (pl. adja vissza az összes olyan csomópontot, amely egy illesztett csomópontból adott típusú él mentén elérhető). A keresés során egy kiterjesztési művelet lényegesen drágább, mint egy ellenőrzési művelet, így a keresési tervben is nagyobb költséggel szerepel az előbbi. A gráfmintaillesztés során alkalmazott heurisztikák döntő többsége ezért a kiterjesztés műveleteinek sorrendezését végzi.

A FUJABA rendszer által konstruált keresési terveket a metamodellben (típusgráfban) lévő asszociációk (éltípusok) multiplicitása (számossága) alapján állítja elő. A FUJABA rendszer heurisztikája szerint az illesztendő gráfminta két éle közül azt célszerűbb korábban illeszteni egy kiterjesztési művelettel, amelyikhez tartozó éltípusnak (asszociációnak) kisebb a számossága. Például egy (0..1) számosságú asszociációt mindenképpen érdemes előbbre venni egy (0..*) számosságú asszociációhoz képest, hiszen az előbbi lényegében egy ellenőrzési műveletté egyszerűsödik, mivel minden csúcsból legfeljebb egy adott típusú él vezethet ki a metamodell értelmében.

---

6. kérdés: Az információvesztés miatt a terminálási eredmények a Petri hálós apparátus alapján elégséges feltételt szolgáltattak a helyességre (a döntés, vagy "helyes" , vagy "nem tudom"). Kérdés, hogy vajon konstruálható-e egy másik Petri hálós leképezés, amely pl. a "nem-helyességre" ad elégséges feltételt (a döntés, vagy "nem helyes" , vagy "nem tudom") ? Ezzel természetesen még mindig nem kapunk szükséges és elégséges feltételt (hiszen a két modell között információ eltérés van) de a modellhelyesség szempontjából informatívabban verifikáló eszköz van a kezünkben.

Az értekezésben igazoltam, hogy minden gráftranszformációs rendszerhez (GTS) származtatható olyan (számossági) Petri háló (PN), amelyik felülről becsüli a GTS által leírt viselkedést, azaz a GTS minden lépése (szabály alkalmazása) szimulálható a PN megfelelő tranzíciójának tüzelésével (lásd 90. oldal). Ebből következik, hogy a PN által specifikált helyes futási utak száma nem kevesebb, mint a GTS által specifikált helyes futási utak száma, így amennyiben a PN helyességét belátjuk minden futási útra, abból következik az eredeti GTS helyessége is.

$$\begin{array}{ccc} M_G & \xRightarrow{t_r=F(r)} & M_H \\ \uparrow_F & & \uparrow_F \\ G & \xRightarrow[(r,o)]{} & H \end{array}$$

A bíráló által felvázolt esetben egy olyan Petri hálót kellene származtatnunk minden egyes gráftranszformációs rendszerhez, amelyik alulról becsüli a GTS viselkedését, azaz a szimulációs tételt fordított irányban kellene igazolnunk: a származtatott PN bármely tranzíciójának tüzelése esetén meg kell tudnunk mutatni, hogy az eredeti GTS is képes végrehajtani a kapcsolódó lépést. Ekkor a GTS által specifikált helyes utak száma nem kevesebb, mint a PN által definiált helyes utak száma, ezért, ha egy lefutási út hibás PN-ben, az hibás lesz a GTS-ben is.

$$\begin{array}{ccc} M_G & \xRightarrow{t_r} & M_H \\ \uparrow_F & & \downarrow_{F^{-1}} \\ G & \xRightarrow[(r,o)=F(t_r)]{} & H \end{array}$$

Tisztán matematikai értelemben vizsgálva a kérdést, ha egy olyan PN-t származtatunk, amelynek kezdőállapotában egyetlen tranzíció sem engedélyezett (azaz a háló sohasem tüzelhet), így a szimulációs tulajdonság definíció szerint (triviálisan) fennáll.

Hadd vázoljak egy másik Petri hálós leképezést is, amely fontos szerepet játszik a gráftranszformációs rendszerek modellellenőrzése során.

0) Rendeljük a GTS kiindulási gráfjához (G_0-hoz) a PN egy helyét (P_0), benne egyetlen tokennel.
1) Számítsuk ki a G_0 összes lehetséges rákövetkező gráfját G_i-t (bármelyik szabály egyszeri alkalmazásával nyert gráfok).
2) Minden G_i-hez rendeljünk egy új PN helyet, és az oda vezető szabályalkalmazáshoz pedig egy PN tranzíciót P_0-ból.
3) Folytassuk a PN generálását az összes G_i-ből kiindulva egy *k* mélységben, szélességi bejárás szerint.

E Petri háló első *k* lépése során minden lépéshez hozzárendelhető az eredeti GTS egy lépése, utána pedig a Petri hálónak nincs több lépése, így onnantól kezdve is igaz marad az inverz szimulációs tulajdonság.

A fenti a módszert Petri háló kihajtogatásnak nevezzük (Petri net unfolding [McMillan95]), és formális modellek dinamikus vizsgálatának ismert módszere. Gráftranszformációs rendszerekhez Baldan és König adaptálták először [BK02], és ezen alapul az Augur modellellenőrző eszköz [KK05] is, amellyel gráftranszformációs rendszerek helyessége vizsgálható kimerítő állapottér-bejárással.

A Bíráló által felvázolt ötlet tehát nemcsak, hogy lehetséges, de nagyon hasznosnak is bizonyult a gráftranszformációs rendszerek formális ellenőrzése során.

[McMillan95] Kenneth L. McMillan: A Technique of State Space Search Based on Unfolding. Formal Methods in System Design 6(1): 45-65 (1995)

[BK02] Paolo Baldan, Barbara König: Approximating the Behaviour of Graph Transformation Systems. ICGT 2002: 14-29

[KK05] Barbara König, Vitali Kozioura: Augur - A Tool for the Analysis of Graph Transformation Systems. Bulletin of the EATCS 87: 126-137 (2005)

Végezetül szeretném ismételten megköszönni Bírálóm körültekintő és nagy szakmai mélységű munkáját. A bírálatban megfogalmazott kérdések közül is kiemelném a 2. és 3. kérdést, amely minden bizonnyal további kutatásokat eredményez majd a modelltranszformációk példák alapján történő megalkotásához.

Budapest, 2013. január 31.

Varró Dániel