

Posudek diplomové práce

Matematicko-fyzikální fakulta Univerzity Karlovy

Autor práce Bc. Marek Dančo

Název práce The Application of SAT Solving to Finite Model Finding

Rok odevzdání 2024

Studijní program Matematika pro informační technologie

Autor posudku RNDr. Jakub Bulín, Ph.D. **Role** oponent

Pracoviště Katedra teoretické informatiky a matematické logiky

Text posudku:

Práce se zabývá problémem počítání a enumerace konečných modelů různých tříd algebraických struktur, až na izomorfismus. K tomu je použito standardního přístupu: převodu na problém splnitelnosti booleovských formulí (SAT), a využití efektivního SAT solveru implementujícího conflict-driven clause learning (zde CaDiCal), a dále nástrojů na určování počtu modelů booleovských formulí (konkrétně D4, clingo).

Algebraické struktury uvažované v práci jsou různé třídy magmat. Magma, někdy také grupoid, je algebraická struktura s jedinou, binární operací. Mezi známé typy magmat, jejichž modely jsou v práci počítány, patří grupy, lupy, kvazigrupy, pologrupy a různé jejich podtřídy dané algebraickými vlastnostmi (komutativita, levá/pravá distributivita, existence levého/pravého inverzního prvku, apod.) Myšlenky a použité postupy by ale šlo přímočaře zobecnit i na jiné typy algebraických struktur.

Na vstupu problému je dána množina klauzulí v predikátové logice prvního řádu, v jazyce obsahujícím symbol pro binární operaci, a dále další operace (typicky unární pro inverzní prvek, konstanty). Tyto vzniknou skolemizací algebraických zákonů definujících danou třídu magmat, a převodem do CNF. A dále je na vstupu dána požadovaná velikost domény. Algoritmus nejprve provede převod do výrokové logiky (flattening, grounding, a dále optimalizace pomocí clause-splitting).

Jádro práce se zabývá otázkou jak generovat neizomorfní modely. Je představena myšlenka symmetry breaking, least number heuristic, a lexleader (máme dané uspořádání tabulky binární operace, a vyžadujeme, aby sestavená tabulka byla lexikograficky nejmenší ze všech 'izomorfních' tabulek), a ukázáno, jak tyto metody zakódovat do SAT.

Zakódování symmetry breaking pro všechny permutace je ale vzhledem k jejich počtu i pro malé domény nepraktické. Standardně lze použít částečnou symmetry breaking podmínku a testovat, zda není nově zkonstruovaný model izomorfní některému už nalezenému. Specifikem magmat je ale typicky obrovské množství neizomorfních modelů, tento přístup proto selhává.

Jako alternativní řešení práce představuje myšlenku úplného symmetry breaking pomocí tzv. kanonizující množiny permutací. Jde o množinu permutací takovou, že je-li daná Cayleyho tabulka lexikograficky nejmenší vzhledem k permutacím z této množiny, je lexikograficky nejmenší vzhledem ke všem permutacím. Je představen algoritmus pro konstrukci kanonizující množiny (Algoritmus 2), za použití SAT solveru, a dále jednoduchý pokus o redukci otestováním, zda lze některou z vygenerovaných permutací zahodit (Algoritmus 3).

Myšlenka kanonizujících množin byla vyvinuta pro grafové problémy, její adaptace na hledání modelů algebraických struktur je asi největším vlastním teoretickým příspěvkem práce. Potom už následuje popis experimentů provedených na 14 různých třídách magmat, a diskuze jejich výsledků.

Textová část práce je celkově pěkná a srozumitelná, oceňuji zejména ilustraci výkladu na průběžných příkladech. Místy trochu méně promyšlená mi přišla úvodní část. Na jednu stranu začíná podrobným představením základních pojmů predikátové logiky, potom ale není nijak vysvětleno např. použití skolemizace. Chyběl mi přesný popis toho, jaké axiomy implementace akceptuje. Definice magmatu v poznámce pod čarou na str. 8 je příliš vágní, jsme omezeni na jednu unární operaci a jednu konstantu?

Implementace je v jazyce Python. Důvodům pro použití Pythonu rozumím, i tak je ale škoda, že nebyl pro takto výpočetně náročné experimenty použit efektivnější jazyk, nebo alespoň kód zrychlen pomocí Numba nebo jiných nástrojů. Problémem repozitáře je chybějící dokumentace, kvůli které byla instalace zbytečně zdlouhavá – určitě by neměl chybět alespoň základní readme (popis skriptů najdeme jen na konci textu práce), a soubor requirements.txt s výčtem potřebných balíčků a jejich verzí. Bylo by lepší organizovat kód jako package.

Celkově jde o zajímavou a kvalitní práci splňující zadání, **doporučuji ji k obhajobě**. Práci nenavrhuji na zvláštní ocenění.

Otázky k obhajobě:

1. V seznamu struktur jsem vzhledem k výzkumnému zaměření některých členů KA očekával také racky a quandly. Bylo by možné enumerovat modely i pro ně?
2. Lze změřit kolik času v experimentech stráví algoritmus na běhu SAT solveru, a kolik je ‘overhead’ implementace v Pythonu?
3. Dávalo by smysl použít v Algoritmu 2 nějakou heuristiku, např. hledat permutace vzestupně podle jejich řádu (případně s využitím ‘vypínání’ klauzulí jako v Algoritmu 3)? Mohlo by to vést ke konstrukci menších kanonizujících množin, nebo je ta intuice špatná?

V Praze dne 7. 8. 2024

Podpis: