UNIWERSYTET JANA DŁUGOSZA w CZĘSTOCHOWIE

Wydział Nauk Ścisłych, Przyrodniczych i Technicznych

Kierunek: Informatyka

Specjalność: Programowanie

Tetiana Mossur

nr albumu: 76716

Efektywność SMT solverów dla klasycznych problemów NP-trudnych

Effectiveness of SMT solvers for classical NP-hard problems

Praca magisterska przygotowana

pod kierunkiem

dr hab. Andrzeja Zbrzeznego

Częstochowa 2024

Streszczenie

Celem niniejszej pracy jest analiza i ocena efektywności SMT solverów w rozwiązywaniu klasycznych problemów zaliczanych do klasy NP-trudnych. Przełom w informatyce teoretycznej sprawił, że tego rodzaju zagadnienia stały się przedmiotem intensywnych badań. Niniejsza praca skupi się na zrozumieniu, jak Satisfiability Modulo Theories (SMT) solvery, będące potężnym narzędziem w dziedzinie rozstrzygania logicznego, radzą sobie z tymi wyjątkowo wymagającymi problemami. Przedmiotem badań będą klasyczne problemy NP-trudne, a celem jest zbadanie i porównanie efektywności SMT solverów Pythona - Z3, Yices i CVC5 - w kontekście rozwiązywania konkretnych problemów. Wybór tych narzędzi wynika z ich popularności, wszechstronności i aktywnego udziału w społeczności badawczej.

Słowa kluczowe

SMT, SMT-solvery, Z3, Yices, CVC5, Python, złożoność obliczeniowa, NP-hard

Spis treści

Wstęp

Celem niniejszej pracy jest zbadanie skuteczności SMT solverów Pythona, a mianowicie Z3, Yices oraz CVC5 w rozwiązywaniu problemów NP-trudnych, takich jak: ... .

W pierwszym rozdziale zawarłam teoretyczne fundamenty problemów NP-trudnych oraz wprowadziłam kontekst, w którym operują SMT solvery. Zdefiniowałam charakterystykę tych problemów i omówiłam kluczowe koncepcje teoretyczne.

Rozdział drugi przedstawia obszerny przegląd literatury z zakresu SMT solverów oraz problemów NP-trudnych. Poprzez analizę wcześniejszych badań, podkreśliłam istotne osiągnięcia oraz ewentualne luki w dotychczasowym rozumieniu tej dziedziny, co stanowi podstawę dla dalszej analizy i eksperymentów.

W rozdziale trzecim zawarłam szczegółowy opis wybranych problemów obliczeniowych oraz SMT solverów, skupiłam się na praktycznych aspektach funkcjonowania tych narzędzi, ich mocnych stronach i potencjalnych ograniczeniach.

Rozdział czwarty poświęcony jest praktycznym eksperymentom, wykorzystując Z3, Yices i CVC5 do rozwiązania wybranych problemów NP-trudnych. Analiza wyników pozwoli określić efektywność poszczególnych solverów i wyciągnąc wnioski co do ich zastosowania.

ROZDZIAŁ 1

Teoretyczne podstawy SMT i klasy NP-hard

1.1. Złożoność obliczeniowa

1.2. Definicja klasy problemów NP-trudnych

1.3. Spełnialność

1.4. Satisfiability Modulo Theories

Satisfiability Modulo Theories (SMT) to dziedzina informatyki teoretycznej, która łączy w sobie problem spełnialności logicznej (SAT) z różnymi teoriami matematycznymi. W kontekście SMT, dany jest zestaw ograniczeń logicznych wyrażonych za pomocą formuł logiki pierwszego rzędu oraz dodatkowe ograniczenia wynikające z konkretnych teorii matematycznych, takich jak teoria liczb całkowitych, teoria równań różniczkowych, czy teoria tablic. Problematyka SMT polega na stwierdzeniu, czy istnieją wartości zmiennych spełniające zarówno ograniczenia logiczne, jak i dodatkowe ograniczenia wynikające z wybranej teorii matematycznej. W przypadku pozytywnej odpowiedzi, rozwiązaniem problemu jest znaczenie konkretnych wartości zmiennych, które spełniają wszystkie warunki.

1.4.1. Teorie

1.4.2. Lazy approach

Teoria

1.5. Definicja i zasada działania SMT-solverów

Teorie spełnialności modulo (SMT) uogólniają teorię spełnialności boole'owskiej (SAT) poprzez dodanie rozumowania równościowego, arytmetyki, bit-wektorów o stałym rozmiarze, tablic, kwantyfikatorów i innych przydatnych teorii pierwszego rzędu. SMT solver jest narzędziem do decydowania o spełnialności (lub poprawności) formuł w tych teoriach. Solvery SMT umożliwiają aplikacje, takie jak rozszerzone sprawdzanie statyczne, abstrakcja predykatów, generowanie przypadków testowych i ograniczone sprawdzanie modelu w nieskończonych dziedzinach.

ROZDZIAŁ 2

Przegląd literatury

2.1. Historia i rozwój SMT solverów

ROZDZIAŁ 3

Opis wykorzystywanych narzędzi

3.1. Z3 Solver

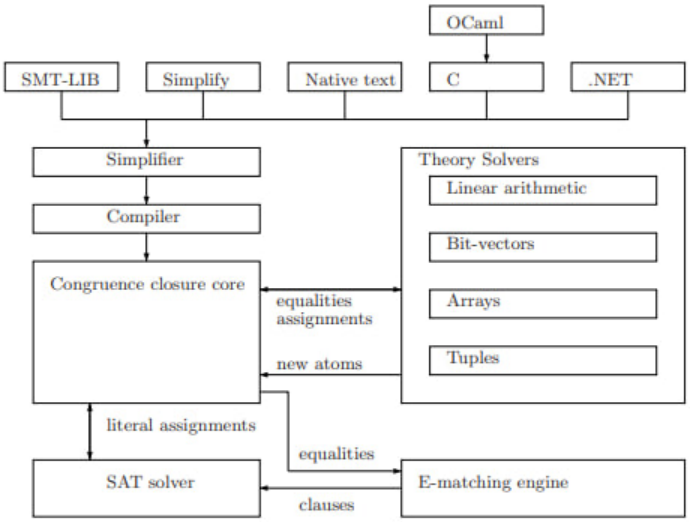
Z3 to wydajny SMT solver dostępny bezpłatnie przez Microsoft Research. Z3 jest solverem dla logiki symbolicznej, będącej podstawą wielu narzędzi inżynierii oprogramowania. Solwery SMT polegają na ścisłej integracji wyspecjalizowanych silników walidacyjnych. Każdy silnik jest elementem ogólnej struktury i implementuje wyspecjalizowane algorytmy. Przykładowo, silnik Z3 dla arytmetyki obejmuje Simplex, cięcia i rozumowanie wielomianowe, podczas gdy silnik dla obsługi ciągów znaków i wyrażeń regularnych korzysta z metod symbolicznych pochodnych języków regularnych. Wspólną cechą wielu algorytmów jest sposób, w jaki wykorzystują dwoistość między znajdowaniem rozwiązań spełniających a dowodów odrzucających. Solver ten integruje również silniki do wnioskowań globalnych i lokalnych oraz globalnej propagacji. Z3 jest używany w szerokim zakresie zastosowań inżynierii oprogramowania, obejmując weryfikację programów, walidację kompilatorów, testowanie, fuzzing przy użyciu dynamicznego wykonywania symbolicznego, rozwój oprogramowania oparty na modelach, weryfikację sieci i optymalizację. Z3 może być zbudowany przy użyciu Visual Studio, pliku Makefile lib CMake. Zapewnia obsługę wielu języków programowania, w tym .NET, C, C++, Java, OCaml, Web Assembly i Python.

3.1.1. Architektura systemu

Z3 integruje nowoczesny solver SAT oparty na DPLL, bazowy solwer dla teorii, który obsługuje równości i funkcje nieinterpretowane, specjalistyczne silniki (dla arytmetyki, tablic itp.) oraz maszynę abstrakcyjną E-matching (dla kwantyfikatorów). Z3 jest zaimplementowany w C++. Schematyczny przegląd Z3 pokazano na poniższym rysunku.

Simplifier. Formuły wejściowe są najpierw przetwarzane przy użyciu niekompletnego, ale wydajnego uproszczenia. Simplifier stosuje standardowe zasady redukcji algebraicznej, takie jak $p \wedge true \implies p$, ale także wykonuje ograniczone uproszczenie kontekstowe, identyfikując definicje równościowe w danym kontekście i redukuje pozostałą formułę przy użyciu definicji, na przykład $x = 4 \wedge q(x) \implies x = 4 \wedge q(4)$. Trywialnie spełnialny spójnik $x = 4$ nie jest kompilowany do jądra, ale zachowany poza nim na wypadek, gdyby klient wymagał modelu do obliczenia x.

Compiler. Uproszczona abstrakcyjna reprezentacja drzewa składniowego formuły jest



Rysunek 3.1.

przekształcana w inną strukturę danych, składającą się ze zbioru klauzul i węzłów domknięcia kongruencji.

Jądro domknięcia kongruencji. Jądro domknięcia kongruencji otrzymuje przypisania prawdy do atomów od solwera SAT. Atomy obejmują równości i formuły atomowe specyficzne dla danej teorii, takie jak nierówności arytmetyczne. Równości stwierdzone przez SAT solver są przekazywane przez jądro domknięcia kongruencji za pomocą struktury danych, którą nazywamy E-grafem. Węzły w E-grafie mogą wskazywać na jeden lub więcej solverów teorii. Gdy dwa węzły są połączone, zbiór odwołań do teorii są łączone, a samo złączenie jest propagowane jako równość do solverów teorii w przecięciu obu zbiorów odwołan. Jądro również propaguje efekty solverów teorii, takie jak wywnioskowane równości oraz atomy przypisane do wartości true lub false.Solvery teorii mogą także generować nowe atomowe wyrażenia w przypadku teorii niekonweksyjnych. Te atomy są następnie integrowane i zarządzane przez główny solver SAT.

Kombinacja teorii. Tradycyjne metody łączenia solverów teorii opierają się na zdolności tych solwerów do generowania wszystkich wynikających równości lub na wprowadzaniu dodatkowych literałów do przestrzeni poszukiwań na etapie wstępnego przetwarzania. Z3 używa nowej metody kombinacji teorii, która przyrostowo dostosowuje modele utrzymywane przez każdą teorię.

SAT Solver. Podziały przypadków logicznych są kontrolowane za pomocą najnowocześniejszego SAT solwera. Solver SAT integruje standardowe metody przycinania wyszukiwania, takie jak dwa obserwowane literały dla wydajnej propagacji ograniczeń boolowskich, lemma learning z wykorzystaniem klauzul konfliktowych, buforowanie faz w celu kierowania podziałami przypadków i wykonuje niechronologiczny backtracking.

Usuwanie klauzul. Instancjonowanie kwantyfikatorów ma skutek uboczny w postaci tworzenia nowych klauzul zawierających nowe atomy w przestrzeni poszukiwań. Z3 usuwa te klauzule wraz z ich atomami i termami, które były bezużyteczne w zamykaniu gałęzi. Klauzule konfliktowe i zawarte w nich literały nie są natomiast usuwane, dlatego instancje kwantyfikatorów, które były przydatne w wywołaniu konfliktów, są zachowywane jako efekt uboczny.

Propagacja relewancji. Solwery oparte na DPLL(T) przypisują wartość boolowską potencjalnie wszystkim atomom pojawiającym się w wyniku. W praktyce niektóre z tych atomów są nieistotne. Z3 ignoruje te atomy dla kosztownych teorii, jak np. wektory bitowe, i reguł wnioskowania, jak instancjonowanie kwantyfikatorów.

Instancjonowanie kwantyfikatorów z użyciem E-matchingu. Z3 wykorzystuje zaawansowaną technikę do rozumowania kwantyfikatorów, która opiera się na E-grafie. Dzięki nowym algorytmom, które identyfikują dopasowania w E-grafach w sposób skuteczny i przyrostowy, Z3 osiąga znaczną przewagę wydajności w porównaniu do innych nowoczesnych SMT solverów.

Theory Solvers. Z3 wykorzystuje liniowy solver arytmetyczny oparty na algorytmie używanym w Yices. Teoria tablic stosuje leniwe instancjonowanie aksjomatów tablicowych. Teoria wektorów bitowych o stałym rozmiarze stosuje bitowanie do wszystkich operacji na wektorach bitowych, z wyjątkiem równości.

Generowanie modeli. Z3 pozwala na generowanie modeli jako części danych wyjściowych. Modele przypisują wartości do stałych na wejściu i generują częściowe grafy dla predykatów oraz symboli funkcji.

3.2. Yices Solver

sdcssdsdf

3.3. CVC5 Solver

sdsdcfsdfcdsf

ROZDZIAŁ 4

Eksperymenty i analiza wyników

4.1. Wybór problemów NP-trudnych do implementacji

4.2. Implementacja problemów w języku Python przy użyciu z3, Yices i CVC5

4.3. Przeprowadzenie eksperymentów i zbieranie danych

4.4. Analiza zebranych danych i porównanie efektywności solverów

4.5. Identyfikacja czynników wpływających na efektywność

Zakończenie

Celem mojej pracy było zapoznanie czytelnika z … Uważam, że udało mi się zrealizować ten cel. Starałem się opisać zarówno …

DODATEK A

Tytuł załącznika jeden

Treść załącznika jeden.

DODATEK B

Tytuł załącznika dwa

Treść załącznika dwa.

Spis tablic

Spis rysunków

Spis listingów