

Použití logické syntézy pro usnadnění řešení SAT problému (BP, DP)

Cílem práce je prozkoumat vliv logické syntézy na řešení problému splnitelnosti booleovské formule (SAT). Tj. prozkoumat, zda a jaká zjednodušení (resp. transformace) booleovské formule vedou ke zrychlení řešení SATu. V literatuře se objevují zmínky o pozitivním i negativním vlivu, dosud ale tento problém nebyl zkoumán dostatečně komplexně.

Jedná se o čistě experimentální práci. Programování pravděpodobně nebude zapotřebí (kromě psaní skriptů pro spouštění experimentů). Naučíte se pracovat s výpočetním clusterem CESNET MetaCentrum (OS Linux) a nástroji pro logickou syntézu.

Výstupem práce budou příslušné statistiky a doporučení.

Exploiting Logic Synthesis in SAT-Problem Solving

The aim of this work is to explore the influence of logic synthesis on SAT solvers. This is, to explore what simplifications (transformations) of logic expressions lead to SAT-solver speed-up. Both positive and negative influences are reported in literature. However, no thorough exploration has been done yet.

It is just an experimental work. Most probably, programming will not be needed (except of writing scripts).

The outputs of the work will be respective statistics and recommendations.

Vyhodnocení efektivity SAT řešičů pro obvodový SAT (BP, DP)

Cílem práce je provést experimentální vyhodnocení efektivity (rychlosti) dostupných open-source řešičů problému splnitelnosti booleovské formule (SAT) pro instance získané transformací z obvodu (netlistu), tj. pro tzv. „circuit-SAT“. Tyto instance jsou svojí povahou specifické. Jsou sice „lehké“ (spíše se blíží 2-SATu), ale objevují se v nich těžké části. SAT řešiče se proto pro ně mohou chovat jinak, než pro běžné zkušební instance. Účinnost dostupných SAT řešičů pro tyto instance zatím nebyla dostatečně zkoumána.

Jedná se o čistě experimentální práci. Programování pravděpodobně nebude zapotřebí (kromě psaní skriptů pro spouštění experimentů). Generátor instancí je k dispozici. Naučíte se pracovat s výpočetním clusterem CESNET MetaCentrum (OS Linux).

Výstupem budou příslušné statistiky a doporučení.

Evaluation of SAT-Solvers Efficiency for Circuit-SAT

The aim of this work is to conduct an experimental evaluation of the efficiency (mainly the speed) of available open-source SAT solvers for circuit-SAT instances. These are instances produced by a transformation of a logic circuit to SAT (CNF). These instances are specific to some extent. In general they are “easy to solve”, however, there sometimes appear difficult parts. Thus, standard SAT-solvers may behave differently for these instances.

It is just an experimental work. Most probably, programming will not be needed (except of writing scripts).

The outputs of the work will be respective statistics and recommendations.

Určování míry podobnosti logických obvodů (BP, DP)

Vytvořte nástroj pro určování míry podobnosti logických obvodů. Vstupem budou dva obvody popsané logickou sítí (netlist), které jsou funkčně ekvivalentní, ale jejich struktura je odlišná. Cílem práce je navrhnout metriku "podobnosti" takovýchto log. sítí a naimplementovat algoritmus pro výpočet míry této podobnosti.

Hlubší znalost problematiky návrhu logických obvodů není nutná - jedná se spíše o řešení grafového problému.

Measuring the Similarity of Logic Circuits

Create a tool for computing of similarity of two logic circuits. The input will be two functionally equivalent (but structurally different) circuits described by a netlist. Design a metric to compute similarity of these two circuits and implement an algorithm computing it.

Deeper knowledge of logic circuits design is not needed; it is rather a graph problem.

Určování míry podobnosti logických obvodů založené na splnitelnosti booleovské formule (BP, DP)

Vytvořte nástroj pro určování míry podobnosti logických obvodů. Vstupem budou dva obvody popsané logickou sítí (netlist), které jsou funkčně ekvivalentní, ale jejich struktura je odlišná. Cílem práce je implementovat algoritmus založený na identifikaci tzv. „common-mode“ poruch a jejich kvantifikaci pomocí řešení problému splnitelnosti booleovské formule (#SAT).

Hlubší znalost problematiky návrhu logických obvodů není nutná - jedná se spíše o implementační problém.

SAT-Based Measurement of the Similarity of Logic Circuits

Create a tool for computing of similarity of two logic circuits. The input will be two functionally equivalent (but structurally different) circuits described by a netlist. The aim of this work is to implement an algorithm determining common-mode faults in the circuits and their quantification using a #SAT solver.

Deeper knowledge of logic circuits design is not needed; it is rather an implementation problem.

Randomizace algoritmů v systému ABC (BP, DP)

ABC je interaktivní open-source nástroj pro logickou syntézu vytvářený na University of California, Berkeley. Je v něm implementována řada syntézních algoritmů (jako je minimalizace, mapování na technologii, FPGA mapování, ...). Všechny tyto algoritmy jsou plně deterministické, tj. při opakovaném spuštění produkuje stejné výsledky. Na druhou stranu jsou však citlivé např. na pořadí vstupů syntetizovaného obvodu, což svědčí o přítomnosti algoritmů, do kterých by mohlo být možné vložit prvek náhodného výběru, bez narušení funkčnosti. Cílem diplomové práce je vybrané algoritmy randomizovat, tj. do míst, kde se algoritmus rozhoduje mezi „lokálně ekvivalentními“ volbami, zavést náhodný výběr. Výsledné randomizované algoritmy otestujte na standardních zkušebních úlohách a porovnejte s původními deterministickými.

Randomization of Algorithms in ABC

ABC is an interactive open-source tool from UC California, Berkeley. Numerous logic synthesis and optimization algorithms implemented in it (minimization, technology mapping, etc.). All these algorithms are almost fully deterministic, i.e., they generate equal results when run repeatedly. The aim of the thesis work is to introduce randomness into some selected algorithms and perform the experimental evaluation.

Programming language: C. Detailed knowledge of the logic synthesis problematics is not essentially necessary. Which is necessary, it the ability to understand the code.

Dekompozice logických funkcí založená na binárních rozhodovacích diagramech (DP)

Vytvořte nástroj (resp. "framework") pro dekompozici logických funkcí založený na dekompozici binárních rozhodovacích diagramů (BDD). Je možné (resp. velice vhodné) použít části dostupných zdrojových kódů podobných nástrojů. Navrhněte heuristiky pro efektivní řízení dekompozice, analyzujte různé možnosti. Vytvořený nástroj otestujte na standardních zkušebních úlohách, získané výsledky porovnejte s alternativními řešeními.

BDD-based Decomposition of Logic Functions

Design a tool for decomposition of logic functions based on binary decision diagrams (BDDs). You may (should) re-use available source code from similar existing tools (BDS). Design heuristics for effective decomposition control, analyze different possibilities. The designed tool should be tested using standard benchmark circuits.

Generování testu pro kombinační obvody s ohledem na implementaci standardními buňkami (DP)

Cílem práce je navrhnout a naimplementovat algoritmus pro automatické generování testu (ATPG) pro kombinační obvody, s ohledem na finální implementaci pomocí standardních buněk. Poruchový model je zde odlišný od klasického (trvalá 1/trvalá 0); uvažují se poruchy přímo na úrovni tranzistorů.

Klasické strukturní ATPG algoritmy pro tento poruchový model jsou známy (resp. rozšíření stávajících algoritmů je přímočaré). Tato práce by se měla zaměřit na využití principů ATPG založených na splnitelnosti booleovské formule (SAT-based ATPG) pro tento účel.

- 1) Nastudujte principy ATPG založených na splnitelnosti booleovské formule (SAT-based ATPG).
- 2) Nastudujte principy generování testu pro standardní buňky (Cell-Aware testing).
- 3) Společně s vedoucím práce navrhnete způsob, jak využít ATPG založené na splnitelnosti booleovské formule pro generování testu pro standardní buňky, případně provedte rešerši, zda takové přístupy již neexistují.
- 4) Výše zmíněné naimplementujte a otestujte.

Kompresi testu pro číslicové obvody s Illinois-Scan architekturou (DP)

Navrhnete a implementujte algoritmus pro kompresi testu pro tzv. „Illinois-scan“ dekompresní architekturu pro testování číslicových obvodů. Zde jsou testovací vektory doručovány do „scan-chainů“ paralelně, což způsobuje ztrátu pokrytí poruch. Původně navržený algoritmus se zaměřoval pouze na řazení klopných obvodů (DFF) ve „scan-chainech“, což je v současném návrhovém postupu nerealistické. Uvažujte tedy předepsané řazení DFF a soustřeďte se na výpočet vhodných testovacích vektorů, za účelem maximalizace pokrytí poruch a minimalizace délky testu. Použijte dostupné nástroje pro generování testovacích vektorů (ATPG).

Výsledný nástroj otestujte a algoritmus experimentálně vyhodnoťte z hlediska pokrytí poruch a délky testu.

Test Compression Based on the Illinois-Scan Architecture

Design and implement a test compression algorithm for the Illinois-scan decompression architecture used for testing digital circuits. Here the test vectors are delivered to multiple scan-chains in parallel, which may cause the fault-coverage loss. The originally proposed algorithm focused mainly on the flip-flops (DFFs) reordering in the scan-chains, which is infeasible in the contemporary digital design flow. Therefore, assume a pre-defined DFFs ordering, and focus on the test patterns generation instead, in order to maximize the fault coverage and minimize the test length. For this purpose, use available test generation tools (ATPGs).

Verify the functionality of the developed tool and perform experimental evaluation of the proposed algorithm, in terms of the fault coverage and the test length.

Aplikace pro vizualizaci pokročilých iterativních heuristik (DP)

Pokračujte v práci na stávající webové aplikaci pro vizualizaci pokročilých iterativních heuristik (simulované ochlazování, genetický algoritmus, Tabu prohledávání). Konkrétně:

- Začlenění externích řešičů implementovaných problémů
- Zavedení podpory skriptování aplikace
- Vytvoření zkušebních instancí, otestování algoritmů na nich
- Oprava drobných chyb, drobná vylepšení

Použité technologie: PHP, JavaScript.

Databáze zkušebních obvodů (DP, BP)

Vytvořte databázi zkušebních obvodů s webovým rozhraním. V této databázi by byly uchovávány obvody a jejich transformace, vše přehledně začleněné do skupin (sad zkušebních obvodů). Aplikace by měla umožňovat přímý přístup k souborům (obvodům), jejich agregaci, přímý přístup k metadatům, statistikám, atd. Proveďte analýzu možných řešení, vyberte vhodné (pravděpodobně to povede na nějakou NoSQL databázi). Výsledný produkt příslušně otestujte.