Proiect verificare formală

Arpad Precup[1], Elber Da[2], Szopuch Denis[3] Radu Bogdan[4]

Universitatea de Vest din Timișoara,
Inginerie Software

1 Abstract

Acest proiect explorează procesul de instalare și rulare al MiniSat, un solver SAT de înaltă performanță, pe sisteme de operare variate, cu un focus specific pe Windows 10, Linux și pe o mașină virtuală cu Ubuntu. Problema SAT (satisfiabilitatea formulelor logice) este una dintre cele mai importante în informatică, având numeroase aplicații în domenii precum verificarea formală, inteligența artificială și teoria complexității. Proiectul documentează pașii de instalare și configurare a MiniSat, provocările întâmpinate în setările de debugging și compilare, precum și soluțiile aplicate pentru a obține o rulare funcțională a aplicației în medii de dezvoltare diferite.

MiniSat · SAT Solver · Instalare · Debugging ·

2 Introducere

Problema satisfiabilității, cunoscută sub acronimul SAT, este o problemă esențială în teoria complexității computaționale, fiind prima problemă demonstrată ca fiind NP-completă. Aceasta constă în determinarea unei atriburi de adevăr pentru variabilele unei expresii logice astfel încât întreaga expresie să fie satisfăcută. Soluționarea problemelor SAT a devenit fundamentală într-o varietate de aplicații, inclusiv verificarea formală a software-ului și hardware-ului, inteligența artificială, optimizarea și bioinformatica.

MiniSat este un solver SAT de înaltă performanță, dezvoltat pentru a participa în competițiile SAT și pentru a rezolva probleme complexe de satisfiabilitate. MiniSat utilizează algoritmi avansați, precum DPLL (Davis-Putnam-Logemann-Loveland) și CDCL (Conflict-Driven Clause Learning), pentru a explora eficient spațiul de căutare al soluțiilor posibile. Datorită eficienței sale și accesibilității ca proiect open-source, MiniSat este adesea utilizat ca reper în benchmark-urile SAT și este o alegere populară pentru studii și cercetări în domeniul satisfiabilității.

Acest proiect urmărește să documenteze procesul de instalare și configurare a MiniSat pe diverse platforme, punând accent pe provocările întâmpinate pe Windows și Linux, inclusiv rularea în mod debugging. De asemenea, include soluții pentru configurarea mediului de dezvoltare Visual Studio Code (VSCode), astfel încât să permită rularea și testarea programului într-un mod flexibil și reproductibil. Această lucrare va contribui la îmbunătățirea accesului la documentație pentru dezvoltatori și cercetători interesați de utilizarea MiniSat pe platforme multiple.

3 Descrierea Problemei SAT

Problema satisfiabilității, cunoscută sub numele de SAT, constă în determinarea valorilor de adevăr pentru variabilele unei expresii logice astfel încât întreaga expresie să fie satisfăcută, adică evaluată ca adevărată. În limbajul logicii propoziționale, o expresie este satisfiabilă dacă există o atribuire a valorilor variabilelor sale care face întreaga expresie să fie adevărată. Această problemă are o importanță majoră în teoria complexității deoarece este prima problemă cunoscută ca fiind NP-completă, fiind astfel esențială în clasificarea și înțelegerea problemelor computaționale complexe.

Algoritmii utilizați pentru rezolvarea problemelor SAT includ tehnici diverse, cei mai reprezentativi fiind algoritmul DPLL și tehnica CDCL. Algoritmul DPLL (Davis-Putnam-Logemann-Loveland) utilizează tehnici de căutare recursivă, reducând treptat problema prin atribuiri succesive ale variabilelor și eliminarea condițiilor imposibile. CDCL (Conflict-Driven Clause Learning) este o optimizare a DPLL, care învață din conflictele întâlnite în timpul căutării soluției, reducând astfel numărul de căi explorate.

Aplicațiile problemelor SAT sunt vaste și acoperă domenii variate, de la verificarea formală a corectitudinii sistemelor software și hardware până la inteligența artificială și bioinformatică. MiniSat, fiind un solver SAT foarte eficient, este utilizat adesea pentru a testa diverse benchmark-uri SAT și pentru a valida probleme complexe în cercetare și dezvoltare.

4 Instalare MiniSat

4.1 Instalare pe Windows 10

Pentru a instala MiniSat pe Windows 10, urmăm câțiva pași ce implică descărcarea codului sursă și configurarea unui mediu de dezvoltare compatibil. MiniSat este un proiect scris în C++, astfel că vom folosi Visual Studio Code sau un IDE similar pentru a permite compilarea și rularea codului C++. Este necesară instalarea MinGW, o suită de unelte care aduce funcționalitățile GNU Compiler Collection (GCC) pe Windows.

1. Descărcăm codul sursă MiniSat de pe GitHub-ul oficial al proiectului. 2. Instalăm MinGW pentru a putea compila codul C++. 3. Configurăm mediul de lucru în VSCode și adăugăm un fișier ‘launch.json‘ pentru a permite rularea în modul debugging. 4. În timpul instalării am întâmpinat erori legate de tipizare strictă; astfel, a fost necesar să folosim opțiunea ‘CXXFLAGS=”-fpermissive”‘ pentru a ignora aceste erori și a permite compilarea.

4.2 Instalare pe Linux

Instalarea MiniSat pe un sistem Linux este relativ simplă, datorită accesibilității librăriilor necesare și a suportului nativ pentru GNU Compiler Collection (GCC). Procesul se realizează prin terminal și implică actualizarea sistemului, descărcarea codului sursă, compilarea și rularea acestuia.

1. Actualizăm lista de pachete și instalăm dependențele necesare. 2. Descărcăm și compilăm codul MiniSat din repo-ul oficial. 3. Pentru rularea cu succes a compilării a fost necesară utilizarea 'CXXFLAGS="-fpermissive"', fără de care procesul întâmpina erori de tipizare strictă. 4. Configurăm un fișier 'launch.json' pentru rularea în VSCode, care facilitează debugging-ul și rularea aplicării MiniSat într-un mod ușor de gestionat.

4.3 Instalare pe o Mașină Virtuală cu Ubuntu

După ce Ubuntu este instalat pe mașina virtuală, deschidem terminalul și actualizăm pachetele sistemului cu comanda: "sudo apt update sudo apt upgrade" Apoi, instalăm dependențele necesare pentru compilare: "sudo apt install g++ make git" Clonăm codul sursă al MiniSat de pe GitHub: "git clone https://github.com/niklasso/minisat.git" Accesăm directorul minisat: "cd minisat" Și compilăm MiniSat folosind: "sudo make CXXFLAGS="-fpermissive"" După finalizare, verificăm instalarea rulând: "./minisat"

5 Provocări Întâmpinate

5.1 Instalarea pe Windows

Instalarea MiniSat pe Windows a întâmpinat mai multe provocări, în special din cauza diferențelor dintre compilatorul Windows și cerințele stricte de tipizare din codul sursă al MiniSat. Am utilizat flag-ul '-fpermissive' pentru a permite compilatorului să ignore aceste diferențe și să accepte codul.

5.2 Rularea în Mod Debugging

Pentru a permite debugging-ul, a fost necesară setarea opțiunii 'CXXFLAGS = -fpermissive -g -O0'. Aceste opțiuni au permis intervenții de depanare detaliată, iar MiniSat ar fi generat erori fără aceste flag-uri. Aceasta a făcut debugging-ul posibil pe Windows și Linux.

5.3 Crearea fișierului launch.json în VSCode

Pentru rularea MiniSat în Visual Studio Code, a fost necesară configurarea unui fișier 'launch.json' specific, astfel încât MiniSat să fie accesibil prin extensia de C++ debugging din VSCode, oferind suport pentru testare și analiză interactivă.

6 Concluzii

Acest proiect a demonstrat cum MiniSat poate fi instalat și utilizat pe platforme diferite, inclusiv Windows, Linux și Ubuntu. Provocările întâmpinate au fost legate de erorile de compilare și setările de debugging, dar soluțiile aplicate au permis obținerea unui mediu de dezvoltare stabil și eficient.