CS402 Introduction to logic for Computer Science

DPLL Project Summary

20160400 Juhyeong Yu

Code Summary

There are three main steps in DPLL: *propagation*, *branch*, and *backtrack*. I use *unit propagation*, *VSIDS*, and *recursive call*. During implementation, I also inserted *resolution checking* to find unsatisfiable formula and *pure literal eliminating propagation* for efficient propagation. However, it did not improve performance and made worse. For decision algorithm, I use (1) VSIDS decision heuristic as referred, stands for Variable State Independent Decaying Sum. It scores for each variable by increasing if associate with conflict variable. And decaying 5% of score for each confliction.

* *init\_and\_start(num)*

If value of *num* is 0, then do DPLL solver on standard input. Otherwise, solve default input files. Initialize formula, assignment and VSIDS variables. It calls *get\_input( )* and *DPLL( )*, thus print the result of DPLL.

* *get\_input(num)*

Get inputs from file on input form. It may exit with some unexpected inputs.

* *DPLL(FA, A, lev)*

It is dually recursive function with *select\_and\_branch( )* function. Returns false if given formula is unsatisfiable otherwise true. It calls *unit\_propagation( )* for propagation and *select\_and\_branch( )* for branch.

* *unit\_propagation(new\_FA, new\_A)*

It checks whether given inputs formula has unit clause(s). If exist, append to assignment and check so that may be unsatisfiable.

* *select\_and\_branch(FA, A, lev)*

It choses new literal for branch by using ASICS method and then branch. It also backtracks if selected branch is wrong. If those two branch are both failed, return false to represent unsatisfiable.

* *add\_and\_check(\_FA, \_A, num)*

Add given num to assignment and check new assignment is valid. For valid assignment, apply \_FA under \_A by calling apply*\_A( )*

* *apply\_A(\_FA, \_A, \_var)*

Apply assignment to formula and change VSIDS values for selecting branch.