CSCE 990 Advanced Constraint Processing
Scribe Notes of November 24, 2009
Speakers: Chris
Scribe: Shant
Title: Building Structure into Local Search for SAT

This document summarizes a technique for exploiting structure in a SAT formula, and utilizing the structural information for simplifying problem solving with local search. The summary is based on the presentation by Chris.

This document first describes the method used for identifying the structure in a SAT sentence in CNF. Then, it explains how local search takes advantage of the structure while solving the SAT. Finally, it ends with a conclusion.

1 Finding Structure in a SAT sentence

Here, we review the tools used by the technique: modeling of the structure in SAT with logic gates, classification of the variables as independent, internal and external, and the dependency lattice structure.

1.1 Modeling of the SAT structure with logic gates

A SAT sentence (in CNF) is evaluated to true for some assignments of its Boolean variables. Such an assignment must satisfy each clause in the sentence. Dependencies between variables are relations between variables that hold for all satisfying assignments. Those relations can be represented by four types of logic gates: OR, AND, XOR and EQUIV.

For example:

- (¬a v b v c) ^ (a v ¬b) ^ (a v ¬c) is satisfied iff a=(b v c)
- (a v ¬b v ¬c) ^ (¬a v b) ^ (¬a v c) is satisfied iff a=(b ^ c)
- (¬a v ¬b v ¬c) ^ (a v b v ¬c) ^ (a v ¬b v c) ^ (¬a v b v c) is satisfied iff a=(b $\oplus$ c)
- (a v b v c) ^ (¬a v ¬b v c) ^ (¬a v b v ¬c) ^ (a v ¬b v ¬c) is satisfied iff a=(b$\Leftrightarrow$c)

Hence the above sentences can be modeled by the logic gates (OR, AND, XOR and EQUIV) with the variables 'b' and 'c' as input and the variable 'a' as output. Thus, the value of the variable 'a' is dependent on the values of the variables 'b' and 'c' in order to satisfy the sentence.

The gates can be extended to more than two variables in pairwise fashion. For example:

a=⇔ (b,c,d) is a=(d⇔ (b⇔c))

1.2 Variable classification

Variables in a SAT sentence are classified as independent, dependent or external variables.

- *Independent variables* are the ones whose values do not depend on the value of any other variable to satisfy the sentence. Those variables are never the output of any gate.
- *Dependent variables* are those whose value depend on the values of some other variables to satisfy the sentence. Hence the values of the dependent variables are determined by one or more gates.
- *The external variables* are variables that must evaluate to true to satisfy the sentence. External variables may be one of the existing variables, or newly created variables. Variables are newly created for clauses that are not 'categorized' by any gate.

1.3 Dependency lattice

After modeling the SAT sentence with the logic gates, the independent variables are identified as those variables that are not the output of any logic gate. They are input to what is called the *external* logic gates. The dependent and external variables are the outputs of the logic gates. Dependent variables can also be inputs to some logic gates that are called *internal* gates. The dependency lattice models the layers of independent variables, external gates, internal gates and external variables.

2 Solving SAT

The top layer in the dependency lattice is composed of independent variables. An appropriate assignment of truth values to only the independent variables determines a solution. Hence the problem size is reduced to $2^{|\text{independent variables}|}$ from $2^{\text{number of vars}}$.

The SAT problem is solved by local search in the following steps:

1. Instantiate the independent variables randomly
2. Repeat until a solution is found or run out of time

a. Propagate the values downward in the lattice through the gates and compute the variable sets (defined below)
b. Compute the make & break costs (defined below) for each independent variable
c. Flip the independent variable with the smallest cost computed from the make & break costs or do a random move
d. Go to step 2

Solving the SAT formula is guided by the *make cost* and the *break cost*. Given an assignment of the independent variables, a gate may change its value when one or more of the independent variables change their values. Thus, given an assignment of the independent variables, the *variable set* of a gate is the set of independent variables that will change the value of the gate. The *make cost* of flipping an independent variable is the number of external variables that will be made true. The *break cost* of flipping an independent variable is the number of external variables that will be make false.

Notice that, after a value to an independent variable is flipped, the change is propagated in the dependency lattice to update the values of the dependent variables and their variables sets. If during the propagation a gate's value and its variable set does not change, the propagation can be safely stopped.

3 Conclusion

The work introduces a technique to exploit the structural dependency between the SAT variables. This structural dependency yields a lattice that is exploited to decrease the search space, guide a local search through cost estimation heuristics, and contain the amount of propagation and updates during local search.