

# Structured Search and Learning

Kenneth L. McMillan

Microsoft Research

## Abstract

Most modern Boolean satisfiability (SAT) solvers use conflict-driven clause learning (CDCL). In this approach, search for a model and search for a refutation by resolution are tightly coupled in a way that helps to focus search on relevant decisions and resolution on relevant deductions. Decision making narrows the search by applying arbitrary constraints. When a contradiction is reached, a “learned” fact is deduced in response. This fact generalizes the conflict and constrains future decisions. The learned fact can also be viewed as a Craig interpolant. As we will see this view allows us to generalize the notion of conflict learning in useful ways.

Satisfiability Module Theories (SMT solvers) of the lazy type apply the same paradigm to first-order decision problems with certain background theories, such as linear arithmetic or the theory of arrays. In this case, the interpolants may be validities of the theory generated by “theory solvers”, but the basic conflict-driven mechanism remains the same.

A common shortcoming of these procedures, successful though they are, is that model search and conflict learning are essentially unstructured. That is, they do not take into account any modular structure that may be present in the decision problem. Decisions are made on variables regardless of their structural relationship, and consequently learned facts do not reflect the problem structure. This is in contrast to a saturation approach, in which we might order resolution so as to exploit, say, narrow tree width of the problem.

In this talk we will consider structured approaches to conflict learning. These techniques have been developed in the context of model checking, an area in which the need to exploit structure is acute. Structured learning can produce facts about reachable states of a system or summaries of procedures, which in turn can be combined to form inductive invariants. Examples of such techniques include IC3 [1] and Lazy Annotation [2].

These techniques have similar search strategies, differing primarily in their approach to computing interpolants. The approaches make different trade-offs between cost and generality, which in turn determine the usefulness of the resulting generalizations. We observe, for example, that more specialized decisions can make the learning problem easier, but possibly at the cost of reduced generality or relevance of the learned facts. Moreover, a substantial effort in generalizing the interpolants can be justified by the corresponding reduction in search.

The net effect of structured learning can be a dramatic improvement in performance, as we observe by comparing with unstructured SMT solvers on bounded software model checking problems.

## References

1. Aaron R. Bradley. SAT-based model checking without unrolling. In Ranjit Jhala and David A. Schmidt, editors, *VMCAI*, volume 6538 of *Lecture Notes in Computer Science*, pages 70–87. Springer, 2011.
2. Kenneth L. McMillan. Lazy annotation for program testing and verification. In Tayssir Touili, Byron Cook, and Paul Jackson, editors, *CAV*, volume 6174 of *Lecture Notes in Computer Science*, pages 104–118. Springer, 2010.