Structured Solvers

Ken McMillan

Microsoft Research

SAT and SMT in CDCL style

- *Conflict-Driven Clause Learning* characterized by:
  - Proof of specialized goals
  - Learning (generalization) by proof transformation.
- Learning in CDCL involves two steps:
  - Decomposing the proof
  - Computing an *interpolant*

Search for a model and search for a refutation are tightly coupled. This helps to focus model search on relevant decisions and proof on relevant inferences.

Exploiting structure

- Problems often have useful modular structure

Example: a BMC formula, with many conjuncts representing successive time frames and small common vocabulary. CDCL doesn't directly exploit this structure.

- This talk: exploiting structure in CDCL
  - Combine unstructured CDCL learning with structured learning using feasible interpolation methods.

We will observe empirically that structured search and learning produces large speedups in software BMC problems.

Interpolants and feasible interpolation

An *interpolant* for a conjunction $A[X,Y] \wedge B[Y,Z]$ is $I[Y]$ such that $A \Rightarrow I$ and $B \Rightarrow \neg I$.

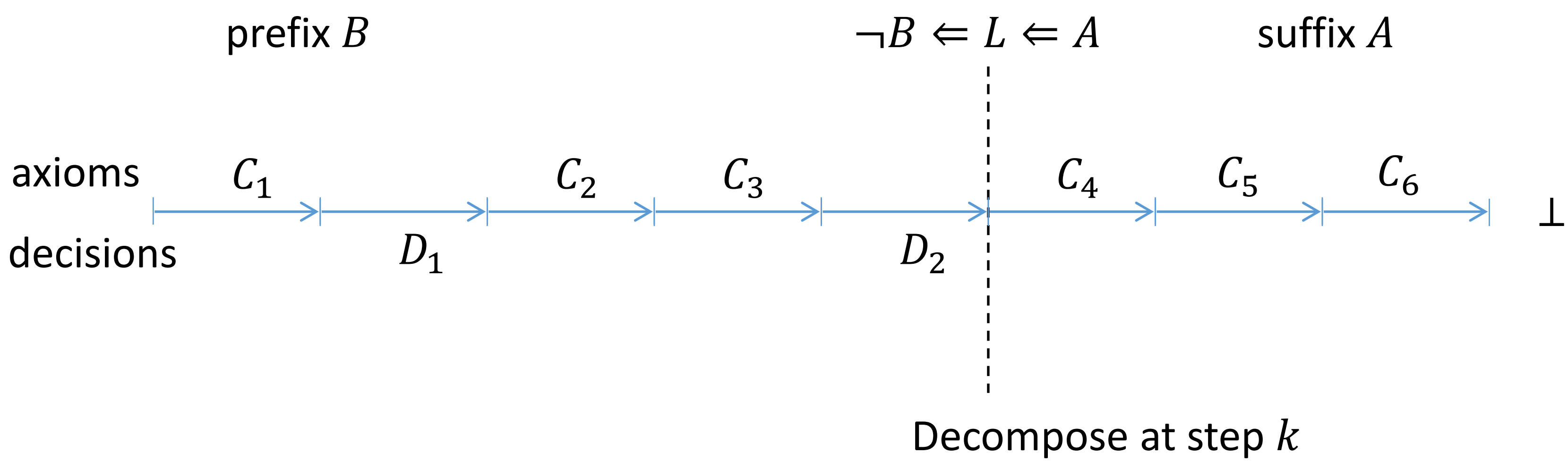
A *feasible interpolation* result for a refutation system says that we can perform this proof transformation in polynomial time:

$$A, B \vdash \bot \quad \Longrightarrow \quad \frac{A \vdash_A I \qquad B \vdash_B \neg I}{A, B \vdash \bot}$$

That is, we can transform a non-modular proof to a modular one.
CDCL uses this idea to form generalizations.

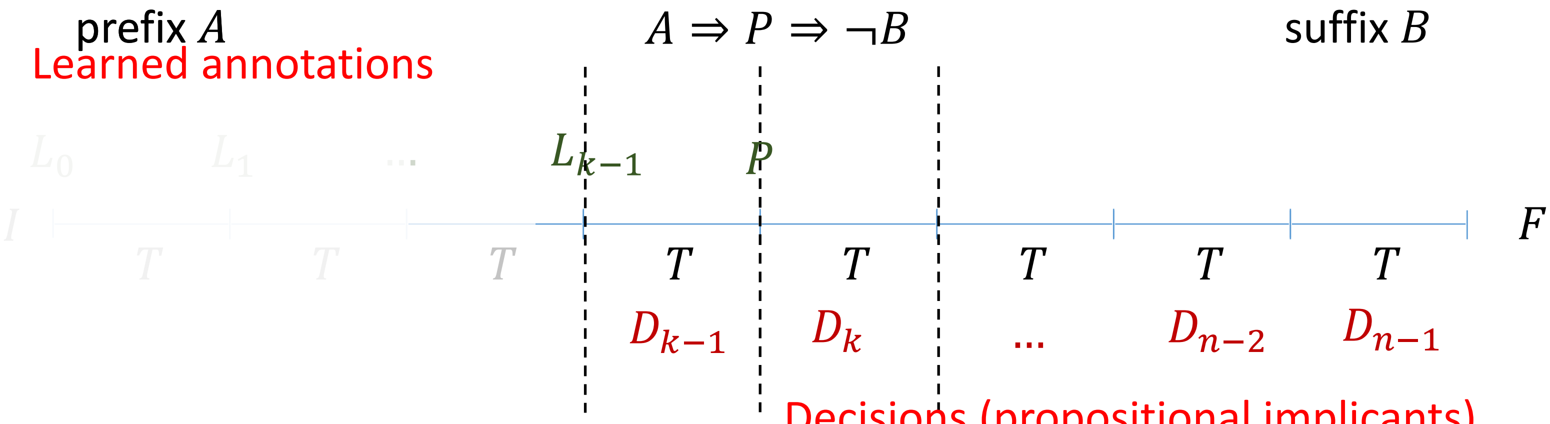
SAT/SMT interpolation strategy

- CDCL builds a specialized proof
  - Proof system is unit-resulting resolution (BCP)
  - Specializations (decisions) are units



The learned clause L is an interpolant between the prefix and suffix of the UR proof, obtained by a simple proof transformation.

Lazy Annotation



Decompose at step $k + 1$

Decompose at step k

BMC formula

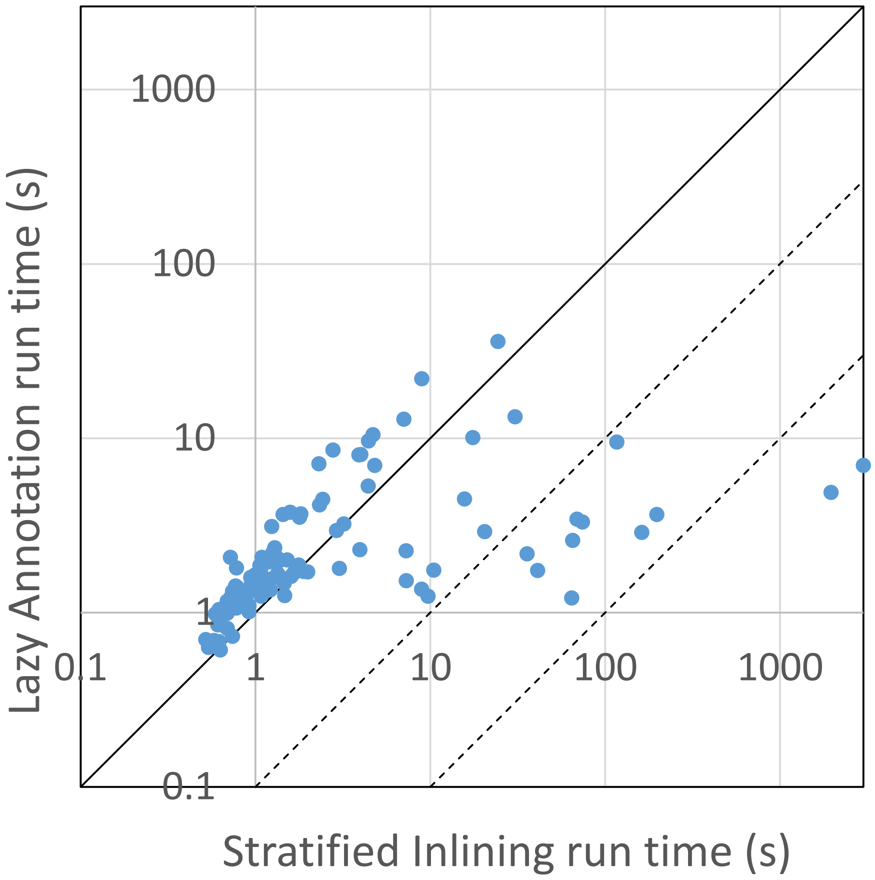
UNSAT: compute interpolant P

SAT: make a decision

Continue until $L_n \Rightarrow \neg F$, or BMC formula proved SAT

Specializing the proof goal (making decisions) makes the decision problem easier, but it might reduce relevance of the learned annotation.

Run time comparison



Learning is orders-of-magnitude slower in LA, but LA can be orders-of-magnitude faster than SI. This shows the greater effectiveness of structured learning.

Advantages of structure

- Structural facts are more re-usable
  - Re-use the summary of a procedure at every call site.
  - At some unfolding depth, the solution may become inductive.
  - Re-use facts from one CEGAR refinement to the next
- The result is fewer backtracks and more efficient search.

Conclusion

- CDCL solvers use narrowing and generalization
  - Build a specialized proof
  - Generalize by partitioning proof and interpolating
  - Model by transformational proof calculus
  - This does not result in modular proofs!
- Trick: proof structure follows problem structure
  - Problem structure embodied in Horn clauses
  - Structural learning rule using feasible interpolation

Structured learning modularizes the proof. This can result in a large gain in efficiency, and also sometimes allows us to construct inductive invariants.

Comparing structured and unstructured

- Software model checking problems
  - From device driver verification in Microsoft SDV
  - Control-oriented safety properties of drivers
- Horn representation
  - Each clause gives the semantics of one procedure
  - Each free predicate is a summary of a procedure
  - Produced by the Boogie VC generator
- The theory
  - Integer arithmetic, arrays, free functions with axioms
- Successive refinement (CEGAR)
  - Corral generates a sequence of refinements