Interpolants from Clausal Proofs

Arie Gurfinkel
Yakir Vizel

iPRA 2014
Vienna, Austria

1. Software Engineering Institute, Carnegie Mellon University, Pittsburgh, USA
2. Computer Science Department, Technion, Israel
MiniDRUP

- SAT with DRUP proofs
- Interpolation-oriented BCP in Trim
- Learn shared-derived clauses in Replay
CDCL SAT solvers

• Check satisfiability of a CNF formula
  – CNF is conjunction of clauses and
  – Clause is a disjunction of literals

• Basic steps:
  – Arbitrary decisions for un-assigned vars
  – Propagate values (BCP)
  – Analyze conflicts and change decisions

SAT solvers can generate refutation proofs
The Implication Graph (BCP)

$$\neg a \land (a \lor \neg b) \land (b \lor c \lor \neg d)$$

Decision

$$\neg a \rightarrow \neg b$$

$$\neg c \rightarrow \neg d$$
Propositional Resolution

\[ a \lor C \quad \neg a \lor D \]

\[ C \lor D \]
Analyzing a Conflict

• Decisions made by the SAT solver may lead to a conflict
  – A clause is evaluated to false under the current assignment

• The implication graph is used to guide resolution steps

• The result is a learnt clause
  – Prevents the same conflict from re-appearing
Refutation Proofs

• A formula is UnSAT when the empty clause can be derived from the original formula.

• Resolution proof
  – A DAG that tracks resolution steps leading from the original clauses to the empty clause
    • Leaves – original clauses
    • Intermediate nodes – learnt/derived clauses

• Clausal proof
  – A sequence of learnt clauses
    • In the order they are learnt
Conflict Clauses

\[ X = \neg a \land (a \lor \neg b) \land (b \lor c \lor \neg d) \land (b \lor d) \]

Decision

\[ \neg c \rightarrow \neg d \]

Learnt clause

anchor

trivial resolution
Resolution Proof
Clausal Proof

• Record learnt clauses in the order they are learnt
  – A learnt clause is derived by Trivial Resolution from some previous clauses
    • If prior to learning c, the CNF is X, then c is derived by Trivial Resolution if running BCP on $X \land \neg c$ leads to a conflict

• for our example, clausal proof is $<X, c>$
Clausal Proof
Clausal Proof

- \(<X, (g_2 \lor g_3), (g_3)>\)
- \(X \land \neg g_2 \land \neg g_3\)
  - \(- \neg a_1\)
  - \(-g_1, \neg g_1 \rightarrow \text{conflict}\)
Clausal Proof

- \(<X, (g_2 \lor g_3), (g_3)\>
- \(X \land (g_2 \lor g_3) \land \neg g_3\)
  - \(\neg g_2\)
  - \(\neg \neg g_2 \rightarrow \text{conflict}\)
DRUP Proof

Marijn et al. FMCAD’13

• Extends a clausal proof by tracking deleted clauses
  – A SAT solver deletes learnt clauses

• \(<X, c_1, c_2, c_3, c_2^*, c_4, c_1^*, c_3^*,...>\)
  – Why?

• Introduced for SAT-solvers certification
Interpolants

• Given an unsatisfiable pair \((A, B)\) of propositional formulas
  – \(A(X, Y) \land B(Y, Z)\) is unsatisfiable

• There exists a formula \(I\) such that:
  – \(A \rightarrow I\)
  – \(I \land B\) is unsatisfiable
  – \(I\) is over the common variables of \(A\) and \(B\)
Resolution Proof

A-local variables: $a_1$
Global variables: $g_1, g_2, g_3$
McMillan’s Method

I = [(g₁ ∨ g₂) ∧ (¬g₁ ∨ g₃)] ∨ [(g₂ ∨ g₃ ∨ ¬g₄) ∧ (g₂ ∨ g₄)]
Clausal Proof

- \langle X, (g_3) \rangle
- X \land \neg g_3
  - \neg \neg g_2
  - \neg \neg a_1
  - \neg g_1, \neg \neg g_1 \rightarrow \text{conflict}
Conflict Clauses

\[\neg g_3 \lor g_2 \lor g_3 \land (\neg a_1 \lor g_2) \land (a_1 \lor g_1 \lor g_2) \land (a_1 \lor \neg g_1 \lor g_3) \land (\neg g_1 \lor g_3) \land (g_1 \lor g_2) \lor g_2 \land (g_3) \lor (\neg g_1 \lor g_3) \land (g_1 \lor g_2) \lor g_2 \]

Decision

\[\neg g_3 \rightarrow \neg g_2 \rightarrow \neg a_1 \rightarrow g_1 \]

\[\neg g_2 \rightarrow g_1 \]
Shared Derivable Clauses

• Given an unsatisfiable pair \((A,B)\) of propositional formulas

• A clause \(c\) is shared-derivable iff
  – \(c\) is over the common variables of \(A, B\)
  – \(c\) is derived using only \(A\) clauses
    • Or, \(A \Rightarrow c\)
Partial CNF Interpolants

• Given an unsatisfiable pair \((A,B)\) of propositional formulas

• Find shared-derivable clauses in the proof and
  – Log them as a CNF formula \(g\)
  – Treat them as \(B\) clauses during the computation

• Interpolant is \(I \land g\)
\[ I = (g_2 \lor g_3) \land (g_2 \lor g_4) \]
Sequence Interpolants

- Given an unsatisfiable tuple \((A, B, C)\) of propositional formulas
  - \(A(X,Y) \land B(Y,Z) \land C(Z,W)\) is unsatisfiable
- There exist formulae \(I_1, I_2\) such that:
  - \(A \rightarrow I_1\)
  - \(I_1 \land B \rightarrow I_2\)
  - \(I_2 \land C \rightarrow \text{FALSE}\)
  - \(I_1\) is over the common variables of \(A\) and \((B, C)\)
  - \(I_2\) is over the common variables of \((A, B)\) and \(C\)
Sequence Interpolants

• A sequence of partial CNFs
  – It is more complex to maintain the sequence property

• A clause is shared-derivable iff:
  – It is derived using only shared-derivable clauses from previous partitions and from clauses within its own partition
Sequence Interpolants
MINIDRUP

SAT

CNF

BCP

BCP + Learning

Trim

Clausal Proof

core proof

Replay

Interpolant

Learning
Restructuring Proofs

• Proofs generally do not have this “special” structure

• Need to force this structure on the proof
  – CNF interpolants are exponentially weaker than general interpolants
  – Must be efficient
  – We do not want to disturb the SAT solver
Restructuring Proofs

- **Observation/Intuition** - let c be a clause over shared vocabulary then one of the following must hold:
  - c is shared-derivable
  - c can be derived using shared-derivable clauses
Experiments
Info

- Visit our web site

- http://arieg.bitbucket.org/avy/

- Come to our CAV talk...
Thank You