Lecture 9
SMT Solvers II

Zvonimir Rakamarić
University of Utah

slides acknowledgements: Leonardo de Moura, Diego Caminha de Oliveira

Announcements I

- Project proposal due tonight

  Write at most 1 page per team describing your project plan. The proposal should include: project title, list of team members, short description of related work, and proposed work with basic milestones. It doesn't have to be too detailed.

- Email it as a PDF document!

Announcements II

- Project proposal presentations on Wed, Feb 13
  - Present your project idea and basic related work in class
  - Up to 10 mins per team + 3 mins for questions

Last Time

- Introduction to SMT
- Nelson-Oppen Theory Combination Procedure

Nelson-Oppen Theory Combination I

- Initial State
  - F is a conjunction of literals over $\Sigma_1 \cup \Sigma_2$
- Purification
  - Preserving satisfiability transform F into $F_1 \wedge F_2$, such that $F_i \in \Sigma_i$
- Interaction
  - Deduce an equality x = y if $F_1 \rightarrow x = y$, where x and y are common (shared) variables
  - Update $F_2 := F_2 \wedge x = y$
  - And vice-versa
  - Repeat until no further changes

Nelson-Oppen Theory Combination II

- Component solvers
  - Use individual theory solvers to decide whether F_i is satisfiable
- Return
  - If both return yes, return yes
  - No, otherwise

- Remark:

  $F_i \rightarrow x = y$ iff $F_i \wedge x \neq y$ is not satisfiable

This Time

- Internals of modern SMT solvers
  - Combining SAT with Nelson-Oppen
- Encoding Hamiltonian cycle problem into SMT using Z3Py

Eager Approach

- Translate formula into equisatisfiable propositional formula and use off-the-shelf SAT solver
- Why “eager”?
  - Search uses all theory information from the beginning
- Can use best available SAT solver
- Sophisticated encodings are need for each theory
- Sometimes translation and/or solving too slow

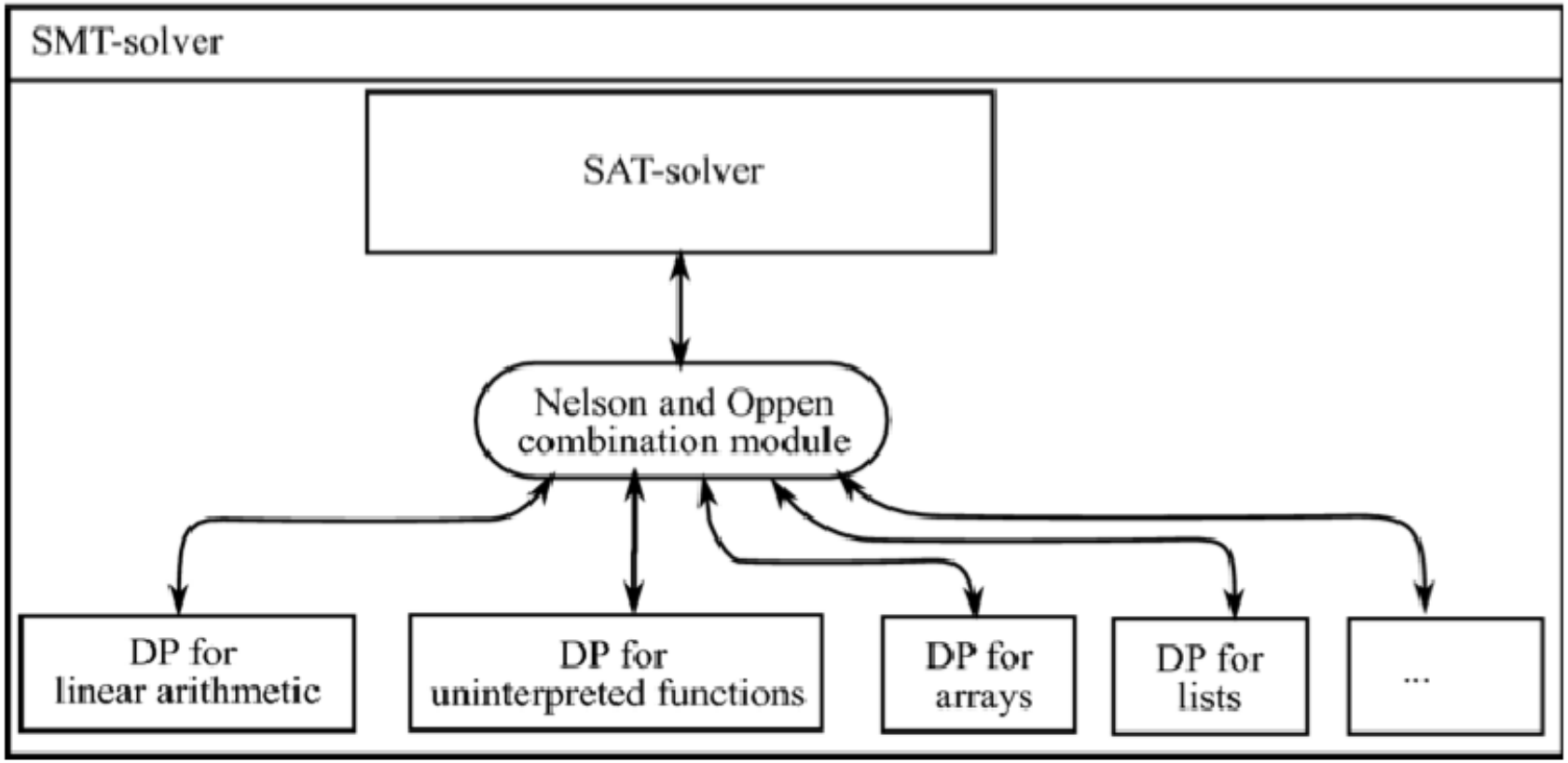
Lazy Approach: SAT + Theories I

- Independently developed by several groups
  - CVC (Stanford)
  - ICS (SRI)
  - MathSAT (Univ. Trento, Italy)
  - Verifun (HP)
- Motivated by the breakthroughs in SAT solving
  - DPLL algorithm
  - Various optimizations and heuristics

Lazy Approach: SAT + Theories II

- SAT solver
  - Manages the boolean structure and assigns truth values to the atoms in a formula
- Theory solvers
  - Efficiently validate (partial) assignments produced by the SAT solver
- When a theory solver detects unsatisfiability, a new clause (lemma) is created

Basic architecture



Naïve Approach

- Example
  - Suppose SAT solver assigns
    $\{x = y \rightarrow T,\ y = z \rightarrow T,\ f(x) = f(z) \rightarrow F\}$
  - Theory solver detects conflict
  - Lemma is created
    $\neg(x = y) \vee \neg(y = z) \vee f(x) = f(z)$
- Potential problems
  - Lemmas are imprecise (not minimal)
  - Theory solver is “passive”
    - It just detects conflicts
    - There is no propagation step
  - Backtracking is expensive
    - Restart from scratch when a conflict is detected

Theory Solvers

- Basic requirements
  - Deduce equalities between variables
  - Compute lemmas (conflict sets)
    - As precise as possible
- Extra desired features
  - Theory propagation
  - Incrementality
  - Backtracking

Equality Generation

- Combination of theories strongly relies on the propagation of deduced equalities
- Every theory solver has to support it

Precise Lemmas I

- Example
  - $\{a_1 = T, a_2 = F, a_3 = F\}$ is inconsistent
  - Lemma is $\neg a_1 \vee a_2 \vee a_3$
- An inconsistent set A is redundant if $A' \subset A$ is also inconsistent
- Redundant inconsistent sets imply
  - Imprecise lemmas
  - Ineffective pruning of the search space

Precise Lemmas II

- Noise of a redundant set is $A \setminus A_{min}$
- Imprecise lemma is useless in any partial assignment where an atom in the noise has a different assignment
- Example
  - Suppose a_1 is in the noise
  - Then $\neg a_1 \vee a_2 \vee a_3$ is useless when $a_1 = F$

Theory Propagation

- SAT solver is assigning truth values to the atoms in a formula
- Partial assignment produced by the SAT solver may imply truth values of unassigned atoms
- Example

  $x = y \wedge y = z \wedge (f(x) \neq f(z) \vee f(x) = f(w))$

  Partial assignment $\{x = y \rightarrow T, y = z \rightarrow T\}$ implies $f(x) = f(z)$
- Reduces the number of conflicts and the search space

Incrementality

- Theory solvers constantly receive new constraints and restart the process
  - Augmented partial assignments from SAT solver
  - Equalities coming from other theory solvers
- Do not restart from scratch
  - Reuse what you learned so far

Efficient Backtracking

- One of the most important improvements in SAT was efficient backtracking
- Extreme (inefficient) approach in theory solvers
  - Restart from scratch on every conflict
- Efficient approach
  - Restore to a logically equivalent state
- Backtracking should be included in the design of theory solvers

Ideal Theory Solver

- Efficient in real benchmarks
- Produces precise lemmas
- Supports theory propagation
- Incremental
- Efficient backtracking

SMT Encoding Example

- Wikipedia:
  - Hamiltonian path is a path in an undirected graph that visits each vertex exactly once
  - Hamiltonian cycle is a Hamiltonian path that is a cycle
  - Determining whether such cycles exist in graphs is the Hamiltonian cycle problem
    - NP-complete
- Task
  - Encode Hamiltonian cycle problem as SMT
  - Use Z3Py to solve it

Next Time

- You are taking over