On first-order model-based reasoning

Maria Paola Bonacina

Dipartimento di Informatica
Università degli Studi di Verona
Verona, Italy, EU

"Logic, Rewriting, and Concurrency" Symposium
Department of Computer Science, The University of Illinois at Urbana-Champaign

24 September 2015

Motivation

- Theorem proving in FOL and first-order theories
- Proofs by refutation
- Inconsistency reveals unsatisfiability: no model
- Models are intuitive for users and relevant to applications
- Model building (not even semi-decidable in FOL)
- Decidable fragments: decision procedures
- SAT and SMT solvers: model-based
- First-order model-based reasoning

Contents of the Festschrift paper

- A survey of semantically guided and model-based methods in first-order logic (FOL) and first-order theories
  Joint work with Uli Furbach and Viorica Sofronie-Stokkermans
- A preview of a new method for first-order reasoning: SGGS for Semantically Guided Goal Sensitive reasoning
  Joint work with David A. Plaisted

Contents of this talk

- Selected key concepts used throughout the part of the survey on reasoning in first-order logic
- Selected main ideas and features of SGGS

Semantic guidance

A reasoning method is semantically guided if it employs a fixed interpretation to drive the inferences.

First example: Semantic resolution

- Given a fixed Herbrand interpretation $\mathcal{I}$
- Generate only resolvents that are false in $\mathcal{I}$
- Crux: finite representation of $\mathcal{I}$
- Examples: finite sets of literals (for finite Herbrand base), multiplication tables

[James Slagle 1967]

Second example: Hyperresolution

- $\mathcal{I}$ contains all negative literals:
  - Positive hyperresolution
  - Generate only resolvents that are positive
- $\mathcal{I}$ contains all positive literals:
  - Negative hyperresolution
  - Generate only resolvents that are negative

[J. Alan Robinson 1965]

Third example: Set of Support strategy

- $H \models^{?} \varphi$
- $H \cup \{\neg\varphi\} \vdash^{?} \bot$
- $H \cup \{\neg\varphi\} \leadsto S$ set of clauses to be refuted
- $S = T \uplus SOS$ where $\{\neg\varphi\} \leadsto SOS$ and $T = S \setminus SOS$ is consistent: $\mathcal{I} \models T$
- Allow resolution only if at least a parent is from SOS
- Add all resolvents to SOS

[Larry Wos, D. Carson, and G. Robinson 1965]

Goal sensitivity

A reasoning method is goal sensitive if it generates only clauses connected with the goal, that is, from the negation of the conjecture.

Example: The set of support strategy is goal sensitive.

[David A. Plaisted and Yunshan Zhu 1997]

Model-based reasoning

A reasoning method is model-based if it builds and transforms a candidate model and uses it to drive the inferences.

Therefore, the state of the derivation includes a representation of a candidate (partial) model.

First example: DPLL

- Model representation: trail of literals
- State of derivation: $M \parallel S$ where M is the trail and S the set of clauses to refute or satisfy
- Guess truth assignments
- Chronological backtracking upon conflict

[Martin Davis and Hilary Putnam 1960]

[Martin Davis, George Logemann, and Donald Loveland 1962]

Clausal propagation

- Conflict clause:
  $L_1 \vee L_2 \vee \ldots \vee L_n$
  for all literals the complement is in the trail
- Unit clause:
  $C = L_1 \vee L_2 \vee \ldots \vee L_j \vee \ldots \vee L_n$
  for all literals but one (L_j) the complement is in the trail
- Implied literal: add L_j to trail with C as justification

[Hantao Zhang and Mark E. Stickel 2000]

[Lintao Zhang and Sharad Malik 2002]

Second example: DPLL-CDCL or CDCL tout court

- Conflict-driven clause learning
- Explanation: conflict clause $A \vee B \vee C$ and $\neg A$ in the trail with justification $\neg A \vee D$: resolve them
- Resolvent $D \vee B \vee C$ is new conflict clause
- Any resolvent is a logical consequence and can be kept: how many? Heuristic
- Backjump: undoes at least a guess, jumps back as far as possible to state where learnt resolvent can be satisfied

[João P. Marques-Silva and Karem A. Sakallah 1997]

[Matthew W. Moskewicz, Conor F. Madigan, Ying Zhao, Lintao Zhang, and Sharad Malik 2001]

SGGS: Semantically-Guided Goal Sensitive reasoning

A new method for first-order theorem proving that is

- Semantically guided
- Goal sensitive (with flexibility)
- Model-based
- Proof confluent (No explicit backtracking)

and that

- Lifts CDCL to first-order logic
- Does not necessarily reduce to DPLL or CDCL on a propositional input

Joint work with David A. Plaisted

SGGS basics

- Set S of clauses to refute or satisfy
- Initial fixed Herbrand interpretation $\mathcal{I}$, e.g.:
  - All negative (similar to positive hyperresolution)
  - All positive (similar to negative hyperresolution)
  - $\mathcal{I} \not\models SOS$, $\mathcal{I} \models T$ (similar to set of support strategy)
  - Other (e.g., $\mathcal{I}$ satisfies the axioms of a theory)
- $\mathcal{I} \models S$: problem solved
- Otherwise: modify $\mathcal{I}$ to satisfy S
- How to represent this modified interpretation?

Semantic guidance for model-based reasoning I

- Propositional logic: P is either true or false; 2^n interpretations for n propositional variables
- First-order logic: $P(x)$ has infinitely many ground instances and there are infinitely many interpretations where each ground instance is either true or false
- That's why we need $\mathcal{I}$ as reference model to have an initial and default notion of what is true and what is false

Semantic guidance for model-based reasoning II

- Propositional logic: if L is true (e.g., it is in the trail), $\neg L$ is false; if L is false, $\neg L$ is true
- First-order logic: if L is true, $\neg L$ is false, but if L is false, we only know that there is a ground instance $L\sigma$ such that $L\sigma$ is false and $\neg L\sigma$ is true
- Uniform falsity: all ground instances false
- $\mathcal{I}$-true: true in $\mathcal{I}$; $\mathcal{I}$-false: uniformly false in $\mathcal{I}$
- If L is $\mathcal{I}$-true, $\neg L$ is $\mathcal{I}$-false
  if L is $\mathcal{I}$-false, $\neg L$ is $\mathcal{I}$-true

SGGS clause sequence

- Γ: sequence of clauses, where every literal is either $\mathcal{I}$-true or $\mathcal{I}$-false
- SGGS-derivation: $\Gamma_0 \vdash \Gamma_1 \vdash \ldots \Gamma_i \vdash \Gamma_{i+1} \vdash \ldots$
- In every clause in Γ a literal is selected:
  $C = L_1 \vee L_2 \vee \ldots \vee L \vee \ldots \vee L_n$ denoted $C[L]$
- $\mathcal{I}$-false literals are preferred for selection
- An $\mathcal{I}$-true literal is selected only in a clause whose literals are all $\mathcal{I}$-true: $\mathcal{I}$-all-true clause

Partial interpretation induced by Γ

- Get a partial interpretation $\mathcal{I}^p(\Gamma)$ by consulting Γ from left to right
- Have each clause $C_i[L_i]$ contribute the ground instances of L_i that satisfy ground instances of C_i not satisfied thus far
- Such ground instances are called proper

Partial interpretation induced by Γ

- If Γ is empty, $\mathcal{I}^p(\Gamma)$ is empty
- If $\Gamma = C_1[L_1], \ldots, C_i[L_i]$, and $I^p(\Gamma|_{i-1})$ is the partial interpretation induced by $C_1[L_1], \ldots, C_{i-1}[L_{i-1}]$, then $\mathcal{I}^p(\Gamma)$ is $I^p(\Gamma|_{i-1})$ plus the ground instances $L_i\sigma$, such that
  - $C_i\sigma$ is ground
  - $I^p(\Gamma|_{i-1}) \not\models C_i\sigma$
  - $\neg L_i\sigma \notin I^p(\Gamma|_{i-1})$

Interpretation induced by Γ

Consult first $\mathcal{I}^p(\Gamma)$ then $\mathcal{I}$:

- Ground literal L
- Determine whether $\mathcal{I}[\Gamma] \models L$:
  - If $\mathcal{I}^p(\Gamma)$ determines the truth value of L, then $\mathcal{I}[\Gamma] \models L$ iff $\mathcal{I}^p(\Gamma) \models L$
  - Otherwise, $\mathcal{I}[\Gamma] \models L$ iff $\mathcal{I} \models L$
- $\mathcal{I}[\Gamma]$ is $\mathcal{I}$ modified to satisfy the clauses in Γ by satisfying their selected literals
- $\mathcal{I}$-false selected literals makes the difference

Example

- $\mathcal{I}$ all negative
- $\Gamma = [P(x)],\ \neg P(f(y)) \vee [Q(y)],\ \neg P(f(z)) \vee \neg Q(g(z)) \vee [R(f(z), g(z))]$
- $\mathcal{I}[\Gamma]$ satisfies all ground instances of $P(x)$, $Q(y)$, and $R(f(z), g(z))$, and no other positive literal

First-order clausal propagation

- Consider an $\mathcal{I}$-false ($\mathcal{I}$-true) literal M selected in clause C_j in Γ, and an $\mathcal{I}$-true ($\mathcal{I}$-false) literal L in C_i, $i > j$:
  if all ground instances of L appear negated among the proper ground instances of M, L is uniformly false in $\mathcal{I}[\Gamma]$
- L depends on M, like $\neg L$ depends on L in propositional clausal propagation when L is in the trail

First-order clausal propagation

- Conflict clause:
  $L_1 \vee L_2 \vee \ldots \vee L_n$
  all literals are uniformly false in $\mathcal{I}[\Gamma]$
- Unit clause:
  $C = L_1 \vee L_2 \vee \ldots \vee L_j \vee \ldots \vee L_n$
  all literals but one (L_j) are uniformly false in $\mathcal{I}[\Gamma]$
- Implied literal: L_j with $C[L_j]$ as justification

Semantically-guided first-order clausal propagation

- SGGS employs assignment functions to keep track of the dependencies of $\mathcal{I}$-true literals on selected $\mathcal{I}$-false literals
- SGGS ensures that $\mathcal{I}$-all-true clauses in Γ are either conflict clauses or justifications with their selected literal as implied literal

How does SGGS build clause sequences?

- Main inference rule: SGGS-extension
- It uses the current Γ and a clause $C \in S$ to generate an instance E of C and adds it to Γ to get Γ'
- Hyperinference: it unifies literals $L_1, \ldots, L_n$ $(n \geq 1)$ of C with selected literals $M_1, \ldots, M_n$ of opposite sign in Γ
- The $M_1, \ldots, M_n$ are $\mathcal{I}$-false: inference guided by $\mathcal{I}[\Gamma]$
- Another substitution ensures that every literal in E is either $\mathcal{I}$-true or $\mathcal{I}$-false

Example

- S contains $\{P(a), \neg P(x) \vee Q(f(y)), \neg P(x) \vee \neg Q(z)\}$
- $\mathcal{I}$: all negative
- Γ_0 is empty
  $\mathcal{I}[\Gamma_0] = \mathcal{I} \not\models P(a)$
- $\Gamma_1 = [P(a)]$
  $\mathcal{I}[\Gamma_1] \not\models \neg P(x) \vee Q(f(y))$
- $\Gamma_2 = [P(a)],\ \neg P(a) \vee [Q(f(y))]$
  $\mathcal{I}[\Gamma_2] \not\models \neg P(x) \vee \neg Q(z)$
- $\Gamma_3 = [P(a)],\ \neg P(a) \vee [Q(f(y))],\ \neg P(a) \vee [\neg Q(f(w))]$
- Conflict!

Lifting theorem

- $\mathcal{I}[\Gamma] \not\models C$ for some clause $C \in S$
- $\mathcal{I}[\Gamma] \not\models C'$ for some ground instance C' of C
- Then SGGS-extension builds an instance E of C such that C' is a ground instance of E

Lifting theorem

$\mathcal{I}[\Gamma] \not\models C'$
For each literal L of C':

- Either L is $\mathcal{I}$-false and not interpreted by $\mathcal{I}^p(\Gamma)$
- Or L is $\mathcal{I}$-false and it depends on an $\mathcal{I}$-true selected literal in Γ
- Or L is $\mathcal{I}$-true and it depends on an $\mathcal{I}$-false selected literal in Γ

Lifting theorem

$\mathcal{I}[\Gamma] \not\models C'$:

- Either C' has $\mathcal{I}$-false literals and at least one of them is not interpreted by $\mathcal{I}^p(\Gamma)$
- Or C' has $\mathcal{I}$-false literals and all of them depend on selected $\mathcal{I}$-true literals in Γ
- Or C' is $\mathcal{I}$-all-true and all its literals depend on selected $\mathcal{I}$-false literals in Γ

Three kinds of SGGS-extension

The added clause E is

- Either a clause that is not in conflict and extends $\mathcal{I}[\Gamma]$ into $\mathcal{I}[\Gamma']$ by adding the proper ground instances of its selected literal
- Or a non-$\mathcal{I}$-all-true conflict clause: need to explain and solve the conflict
- Or an $\mathcal{I}$-all-true conflict clause: need to solve the conflict

First-order conflict explanation: SGGS-resolution

- It resolves a non-$\mathcal{I}$-all-true conflict clause E with a justification $D[M]$
- The literals resolved upon are an $\mathcal{I}$-false literal L of E and the $\mathcal{I}$-true selected literal M that L depends on
- Each resolvent is still a conflict clause and it replaces the previous conflict clause in Γ
- It continues until all $\mathcal{I}$-false literals in the conflict clause have been resolved away and it gets either $\square$ or an $\mathcal{I}$-all-true conflict clause
- If $\square$ arises, S is unsatisfiable

First-order conflict-solving: SGGS-move

- It moves the $\mathcal{I}$-all-true conflict clause $E[L]$ to the left of the clause $D[M]$ such that L depends on M
- It flips at once from false to true the truth value in $\mathcal{I}[\Gamma]$ of all ground instances of L
- The conflict is solved, L is implied, $E[L]$ satisfied and it becomes the justification of L

Example

- S contains $\{P(a), \neg P(x) \vee Q(f(y)), \neg P(x) \vee \neg Q(z)\}$
- $\Gamma_3 = [P(a)],\ \neg P(a) \vee [Q(f(y))],\ \neg P(a) \vee [\neg Q(f(w))]$
- $\Gamma_4 = [P(a)],\ \neg P(a) \vee [\neg Q(f(w))],\ \neg P(a) \vee [Q(f(y))]$
- $\Gamma_5 = [P(a)],\ \neg P(a) \vee [\neg Q(f(w))],\ [\neg P(a)]$
- $\Gamma_6 = [\neg P(a)],\ [P(a)],\ \neg P(a) \vee [\neg Q(f(w))]$
- $\Gamma_7 = [\neg P(a)],\ \Box,\ \neg P(a) \vee [\neg Q(f(w))]$
- Refutation!

Discussion

- SGGS is richer than what presented here: first-order literals may intersect having ground instances in common
- SGGS works with constrained clause, where SGGS constraints are a kind of Herbrand constraints (e.g., $x \not\equiv y \triangleright P(x, y)$)
- SGGS uses splitting inference rules to partition clauses and isolate intersections that can then be removed by SGGS-resolution (different sign) or SGGS-deletion (same sign)

Discussion

SGGS is

- Refutationally complete, regardless of the choice of $\mathcal{I}$
- Goal sensitive if $\mathcal{I} \not\models SOS$ and $\mathcal{I} \models T$ for $S = T \uplus SOS$

References on SGGS

- Semantically-guided goal-sensitive reasoning: model representation. *Journal of Automated Reasoning*, 29 pp., published online June 26, 2015.
- Semantically-guided goal-sensitive reasoning: inference system and completeness. To be submitted.
- SGGS theorem proving: an exposition. 4th Workshop on Practical Aspects in Automated Reasoning (PAAR), Vienna, July 2014. EPiC 31:25-38, July 2015.
- Constraint manipulation in SGGS. 28th Workshop on Unification (UNIF), Vienna, July 2014. TR 14-06, RISC, 47–54, 2014.