Conflict-driven first-order decision procedures[1]

Maria Paola Bonacina

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

Theoretical Foundations of SAT/SMT Workshop
Satisfiability: Theory, Practice, and Beyond Program
Simons Institute for the Theory of Computing

Berkeley, California, USA, 24 March 2021

[1] Based on joint work with Sarah Winkler and joint work with David A. Plaisted

Logic-based automated reasoning

Traditional view from the decidable towards the undecidable, and from the least towards the most expressive:

- Solvers for satisfiability in propositional logic (SAT)
- Solvers for satisfiability modulo theories (SMT)
- Theorem provers for first-order reasoning (ATP)
- Proof assistants for higher-order reasoning (ITP)

Current research trends challenge the borders

Current trends in automated reasoning

Integration and hybridization, e.g.:

- At the border between higher-order and first-order logic, e.g.:
  - Solvers and provers inside or as backend to proof assistants
  - Higher-order automated theorem provers
- At the border between first-order logic and SMT/SAT, e.g.:
  - Quantifiers in SMT
  - Conflict-driven reasoning in first-order logic
- In tools for applications

This talk: conflict-driven reasoning in first-order logic

What is conflict-driven reasoning

- Procedure to determine satisfiability of a formula
- Search for a model by building candidate models
- Assignments + propagation through formulas
- Conflict btw model and formula: explain by inferences
- Learn generated lemma to avoid repetition
- Solve conflict by fixing model to satisfy learned lemma
- Nontrivial inferences on demand to respond to conflicts

Conflict-driven reasoning

- For SAT: Conflict-Driven Clause Learning (CDCL)

  [Marques Silva, Sakallah: ICCAD 1996, IEEE TOC 1999]

- For several fragments $\mathcal{T}$ of arithmetic: conflict-driven $\mathcal{T}$-satisfiability procedures

  [Korovin et al.: CP 2009] [McMillan et al.: CAV 2009] [Cotton: FORMATS 2010] [Jovanović, de Moura: JAR 2013] [Jovanović, de Moura: IJCAR 2012] [Brauße et al.: FroCoS 2019]

- For SMT: Model Constructing Satisfiability (MCSAT)

  [Jovanović, de Moura: VMCAI 2013]

- For SMT with combination of theories and SMA: Conflict-Driven Satisfiability (CDSAT)

  [MPB, Graham-Lengrand, Shankar: CADE 2017, CPP 2018, JAR 2020]

Conflict-driven reasoning

- Question: And first-order logic?
- Semantically-Guided Goal-Sensitive (SGGS) reasoning

  [MPB, David A. Plaisted: PAAR 2014, JAR 2016, JAR 2017]

This talk: can we get decision procedures from SGGS?

- SGGS decision procedures for decidable fragments of first-order logic

  [MPB, Sarah Winkler: IJCAR 2020]
- Conflict-driven and model-constructing

SGGS basics

- S: set of clauses
- Semantic guidance: a fixed Herbrand interpretation $\mathcal{I}$
  Sign-based: $\mathcal{I} = \mathcal{I}^-$ all-negative or $\mathcal{I} = \mathcal{I}^+$ all-positive
- $\mathcal{I} \not\models S$: search for a model
- SGGS works with a trail Γ: a sequence of (possibly constrained) clauses with selected literals
- Γ represents an interpretation $\mathcal{I}[\Gamma]$ that modifies $\mathcal{I}$ by satisfying the selected literals
- Get either a Γ such that $\mathcal{I}[\Gamma] \models S$
  or a contradiction $\bot$ (the empty clause)

Example I: SGGS finds a refutation

- S_1 contains $\{P(a),\ \neg P(x) \vee Q(f(y)),\ \neg P(x) \vee \neg Q(z)\}$
- $\mathcal{I} = \mathcal{I}^-$ (all-negative)
- Γ_0 is empty: $\mathcal{I}[\Gamma_0] = \mathcal{I} \not\models P(a)$
- $\Gamma_1 = [P(a)]$ by SGGS-extension with empty mgu
  where $[P(a)]$ is selected
- $\mathcal{I}[\Gamma_1] \not\models \neg P(x) \vee Q(f(y))$
- $\Gamma_2 = [P(a)],\ \neg P(a) \vee [Q(f(y))]$
  by SGGS-extension with mgu $\alpha = \{x \leftarrow a\}$
  where $[Q(f(y))]$ is selected and $\neg P(a)$ is assigned to $[P(a)]$

Example I: SGGS finds a refutation

- S_1 contains $\{P(a),\ \neg P(x) \vee Q(f(y)),\ \neg P(x) \vee \neg Q(z)\}$
- $\Gamma_2 = [P(a)],\ \neg P(a) \vee [Q(f(y))]$
- $\mathcal{I}[\Gamma_2] \not\models \neg P(x) \vee \neg Q(z)$
- $\Gamma_3 = [P(a)],\ \neg P(a) \vee [Q(f(y))],\ \neg P(a) \vee [\neg Q(f(y))]$
  by SGGS-extension with mgu $\alpha = \{x \leftarrow a, z \leftarrow f(y)\}$ where $[\neg Q(f(y))]$ is selected and $\neg P(a)$ is assigned to $[P(a)]$
- Conflict: $\mathcal{I}^-$-all-true conflict clause
  whose literals are all assigned

Example I: SGGS finds a refutation

- S_1 contains $\{P(a),\ \neg P(x) \vee Q(f(y)),\ \neg P(x) \vee \neg Q(z)\}$
- $\Gamma_3 = [P(a)],\ \neg P(a) \vee [Q(f(y))],\ \neg P(a) \vee [\neg Q(f(y))]$
- $\Gamma_4 = [P(a)],\ \neg P(a) \vee [\neg Q(f(y))],\ \neg P(a) \vee [Q(f(y))]$
  by SGGS-move: $\mathcal{I}[\Gamma_4] \models \neg Q(f(y))$
- $\Gamma_5 = [P(a)],\ \neg P(a) \vee [\neg Q(f(y))],\ [\neg P(a)]$
  by SGGS-resolution (with empty matching):
  the resolvent replaces the non-$\mathcal{I}^-$-all-true parent
- $\Gamma_6 = [\neg P(a)],\ [P(a)],\ \neg P(a) \vee [\neg Q(f(y))]$ by SGGS-move
- $\Gamma_7 = [\neg P(a)],\ \bot,\ \neg P(a) \vee [\neg Q(f(y))]$ by SGGS-resolution

Conflict-driven reasoning in SGGS

$C = L_1 \vee \ldots [L_j] \vee \ldots \vee L_k$

- Decision: SGGS-extension and literal selection
  adds all ground instances of L_j needed for $\mathcal{I}[\Gamma] \models C$
- Propagation:
  - Conflict clause: for all i, $1 \leq i \leq k$, $\mathcal{I}[\Gamma] \models \neg L_i$
  - Implied literal and justification:
    for all i, $1 \leq i \neq j \leq k$, $\mathcal{I}[\Gamma] \models \neg L_i$
- Conflict solving:
  - Conflict explanation: SGGS-resolution
  - Learning: SGGS-move

Example II: SGGS finds a model

- S_2 contains
  - $P(x, x, a)$, $\quad P(x, y, w) \vee P(y, z, w) \vee \neg P(x, z, w)$
  - $\neg P(x, x, b)$, $\quad P(x, z, w) \vee \neg P(x, y, w) \vee \neg P(y, z, w)$
- $\mathcal{I} = \mathcal{I}^-$ all-negative
- $\Gamma_1 = [P(x, x, a)]$
- $\Gamma_2 = [P(x, x, a)],\ P(x, y, a) \vee [P(y, x, a)] \vee \neg P(x, x, a)$
  by SGGS-extension with mgu $\alpha = \{z \leftarrow x, w \leftarrow a\}$
  (selecting $P(x, y, a)$ makes no difference)

Example II: SGGS finds a model

- S_2 contains
  - $P(x,x,a)$, $P(x,y,w) \vee P(y,z,w) \vee \neg P(x,z,w)$
  - $\neg P(x,x,b)$, $P(x,z,w) \vee \neg P(x,y,w) \vee \neg P(y,z,w)$
- $\Gamma_2 = [P(x,x,a)],\ P(x,y,a) \vee [P(y,x,a)] \vee \neg P(x,x,a)$
- $\Gamma_3 = [P(x,x,a)],\ P(x,x,a) \vee [P(x,x,a)] \vee \neg P(x,x,a)$,
  $y \neq x \rhd P(x,y,a) \vee [P(y,x,a)] \vee \neg P(x,x,a)$
  by SGGS-splitting to remove the intersection btw selected literals
- SGGS-splitting introduces constraints

Example II: SGGS finds a model

- S_2 contains
  - $P(x, x, a)$, $P(x, y, w) \vee P(y, z, w) \vee \neg P(x, z, w)$
  - $\neg P(x, x, b)$, $P(x, z, w) \vee \neg P(x, y, w) \vee \neg P(y, z, w)$
- $\Gamma_3 = [P(x, x, a)],\ P(x, x, a) \vee [P(x, x, a)] \vee \neg P(x, x, a),$ $y \neq x \rhd P(x, y, a) \vee [P(y, x, a)] \vee \neg P(x, x, a)$
- $\Gamma_4 = [P(x, x, a)],\ y \neq x \rhd P(x, y, a) \vee [P(y, x, a)] \vee \neg P(x, x, a)$ by SGGS-deletion as the second clause is satisfied
- $\mathcal{I}[\Gamma_4] \models S$: SGGS halts
- Is termination on this set expected? Yes and no

Why not? Because hyperresolution does not halt

- Semantic resolution: generate only resolvents false in $\mathcal{I}$

  [Slagle: JACM 1967]

- Hyperresolution: semantic resolution with $\mathcal{I}^-$ or $\mathcal{I}^+$: sign-based semantic guidance

  [Robinson: IJCM 1965]

- Positive hyperresolution: resolve a non-positive clause C with as many positive clauses as needed to resolve away with a simultaneous mgu all negative literals in C and get a positive resolvent (false in $\mathcal{I}^-$)
- Negative hyperresolution: dual with $\mathcal{I}^+$

Why not? Because hyperresolution does not halt

- S_2 contains
  - $P(x, x, a)$, $\quad P(x, y, w) \vee P(y, z, w) \vee \neg P(x, z, w)$
  - $\neg P(x, x, b)$, $\quad P(x, z, w) \vee \neg P(x, y, w) \vee \neg P(y, z, w)$
- Positive hyperresolution generates infinitely many clauses from $P(x, x, a)$ and $P(x, y, w) \vee P(y, z, w) \vee \neg P(x, z, w)$
- Negative hyperresolution generates infinitely many clauses from $\neg P(x, x, b)$ and $P(x, z, w) \vee \neg P(x, y, w) \vee \neg P(y, z, w)$

[Fermüller, Leitsch, Hustadt, Tammet: AR Handbook 2001]

[Caferra, Leitsch, Peltier: Automated Model Building book 2004]

Why yes? Because S_2 is in the Bernays-Schönfinkel class

- Also known as EPR for Effectively PRopositional
- Sentences of the form $\exists^*\forall^*\varphi$
  φ: formula with neither quantifiers nor functions (constants allowed)
- Clausal form: replace $\exists$-quantified variables by Skolem constants; no function symbols; finite Herbrand base; decidable
- Decision procedures, e.g.: DPLL(SX) [Piskac, de Moura, Bjørner: JAR 2010], NRCL [Alagi, Weidenbach: FroCoS 2015], SCL [Fiori, Weidenbach: CADE 2019]

Towards SGGS decision procedures

- Does SGGS decide EPR? Yes
- Does SGGS decide other known decidable fragments of first-order logic? Some but not all
  (with sign-based semantic guidance)
- Does SGGS allows us to discover new decidable fragments of first-order logic? Yes

How SGGS makes progress

- Disjoint prefix $dp(\Gamma)$: longest prefix of Γ with no intersection of selected literals
- Suppose $\bot \notin \Gamma$ and $\mathcal{I}[\Gamma] \not\models S$
- If $\Gamma \neq dp(\Gamma)$: remove intersection (SGGS-splitting) or solve conflict (SGGS-resolution, SGGS-move)
- If $\Gamma = dp(\Gamma)$: as $\mathcal{I}[\Gamma] \not\models C$ for some clause $C \in S$, extend Γ hence $\mathcal{I}[\Gamma]$ (SGGS-extension)
- Non-termination may come only from infinitely many SGGS-extensions

Fairness of a derivation

- Makes progress whenever possible
- Every SGGS-extension that adds a conflict clause is bundled with conflict solving
- Applies SGGS-deletion eagerly
- Does not neglect inferences on shorter prefixes to work on longer ones
- Fair search plan: all derivations are fair
- Limit Γ_∞ of a fair derivation

Fundamental theorems about SGGS

- S: input set of clauses
- A descending chain of length-bounded trails is finite
- A fair derivation is a descending chain
- SGGS is refutationally complete:
  if S is unsatisfiable, SGGS halts with a refutation
- SGGS is model-complete in the limit:
  if S is satisfiable, $\mathcal{I}[\Gamma_\infty] \models S$

SGGS decision procedures

- Refutational completeness ensures termination on unsatisfiable inputs
- In order to get a decision procedure, we need termination on satisfiable inputs:
  1. Show that the length of SGGS-trails is bounded
  2. Show that only finitely many SGGS-extensions can apply

Finite basis

- S: input set of clauses
- $\mathcal{A}$ its Herbrand base
- Finite basis: finite subset $\mathcal{B} \subseteq \mathcal{A}$
- An SGGS-derivation is in the finite basis $\mathcal{B}$ if all ground instances of all clauses ever appearing on the trail are made of atoms in $\mathcal{B}$

Termination of SGGS in a finite basis

- Finite basis $\mathcal{B}$
- **Lemma:** if a fair derivation is in $\mathcal{B}$, at all stages the length of the trail is upper bounded by $|\mathcal{B}|$
  ($|\Gamma_j| \leq |\mathcal{B}|+1$ and $|\Gamma_j| \leq |\mathcal{B}|$ if $dp(\Gamma_j) = \Gamma_j$)
- **Theorem:** a fair SGGS-derivation in a finite basis is finite

Decidability by the finite basis approach

- Fragment $\mathcal{F}$
- Show that for all clause sets S of $\mathcal{F}$ there is a finite basis $\mathcal{B}$ for SGGS
- $\mathcal{B}$ may depend on S
- Then any fair SGGS-strategy is a model-constructing decision procedure for $\mathcal{F}$

Small model property by the finite basis approach

Every satisfiable clause set S has a model whose cardinality is upper-bounded

- Finite basis $\mathcal{B}$ for SGGS
- Fair SGGS-derivation: halts with a Γ such that $\mathcal{I}[\Gamma] \models S$
- Domain of $\mathcal{I}[\Gamma]$: the Herbrand universe $\mathcal{H}$ for S
  infinite in general
- $\mathcal{H}(\mathcal{B}) \subseteq \mathcal{H}$: only the subterms of atoms in $\mathcal{B}$
- $\mathcal{H}(\mathcal{B})$ is finite as $\mathcal{B}$ is finite
- **Theorem:** S has a model of cardinality $|\mathcal{H}(\mathcal{B})| + 1$ that can be extracted from Γ

SGGS decides the stratified fragment

Stratified fragment [Abadi, Rabinovitch, Sagiv: JSC 2010]

- Well-founded ordering $<$ on sorts: if $f : s_1 \times \ldots \times s_n \rightarrow s$ then $s < s_i$
- Sort-dependency graph: arc from s_i to s
- No cycles: no series such as $a, f(a), f^2(a), f^3(a), \ldots$ or $a, f(a), g(f(a)), f(g(f(a))), \ldots$
- The finite basis $\mathcal{B}$ is the Herbrand base itself
- EPR is the special case with one sort: no function symbols
- Check stratification after Skolemization ($\exists^*\forall^*$ is ok)

Ground-preserving clauses

Clause C: C^+ positive literals; C^- negative literals

- Negatively ground-preserving: $\mathcal{V}ar(C) \subseteq \mathcal{V}ar(C^+)$

  [Kounalis, Rusinowitch: JSC 1991]
- Positively ground-preserving: $\mathcal{V}ar(C) \subseteq \mathcal{V}ar(C^-)$

  [Fermüller, Leitsch: CSL 1993] [MPB, Lynch, de Moura: JAR 2011]

  Also known as range-restricted

S positively ground-preserving: positive clauses are ground, positive hyperresolution only generates ground clauses, and **Lemma:** so does SGGS with $\mathcal{I}^-$ (suitable $\mathcal{I}$)

Restrained clauses: intuition

$S_3 = \{P(s^{10}(0), s^9(0)),\ \neg P(s(s(x)), y) \vee P(x, s(y)),\ \neg P(s(0), 0)\}$
$\mathcal{I} = \mathcal{I}^-$ all-negative

- $\Gamma_1 = [P(10, 9)]$
- $\Gamma_2 = [P(10, 9)],\ \neg P(10, 9) \vee [P(8, 10)]$
- $\Gamma_3 = [P(10, 9)],\ \neg P(10, 9) \vee [P(8, 10)],\ \neg P(8, 10) \vee [P(6, 11)]$

  ....
- $\Gamma_6 = [P(10, 9)], \ldots \neg P(2, 13) \vee [P(0, 14)]$ and $\mathcal{I}[\Gamma_6] \models S_3$

$P(s(s(x)), y) \succ P(x, s(y))$
$\succ$: KBO where s has positive weight

Restrained clauses

Restraining quasi-ordering $\succeq$:

- Stable (under substitutions)
- $\succ$ well-founded
- $\approx = \succeq \cap \preceq$ has finite equivalence classes

Clause C is (strictly) positively restrained:

- Positively ground-preserving ($\mathcal{V}ar(C) \subseteq \mathcal{V}ar(C^-)$)
- For all non-ground $L \in C^+$ there exists $M \in C^-$ such that $M \succeq L$ ($M \succ L$)

Why a quasi-ordering?
$differ(x, y) \vee \neg differ(y, x)$: $differ(x, y) \succeq_{acrpo} differ(y, x)$

SGGS decides the restrained fragment

S restrained set of clauses, $\mathcal{A}$ its Herbrand base

- $\mathcal{A}_S$: set of ground atoms in S
- Finite basis: $\mathcal{A}_S^{\preceq} = \{L \,:\, L \in \mathcal{A}, \exists M \in \mathcal{A}_S \text{ with } M \succeq L\}$: the ground atoms upper-bounded by those in S
- **Lemma:** any fair SGGS-derivation with suitable $\mathcal{I}$ is in $\mathcal{A}_S^{\preceq}$
- **Theorem:** any fair SGGS-derivation halts, is a refutation if S is unsatisfiable, and constructs a model of S if S is satisfiable
- **Corollary:** S satisfiable, model of size $|\mathcal{H}(\mathcal{A}_S^{\preceq})| + 1$

In the example, S_3 has a model of cardinality 21

More positive results

- SGGS decides the positive variable dominated (PVD) fragment, also by the finite basis approach
- Positive hyperresolution and positive ordered resolution decide the positively restrained fragment
- Negative hyperresolution and negative resolution decide the negatively restrained fragment

How to determine that a set of clauses is restrained

- Reduce restrainedness of $C \in S$ to termination of a rewrite system $(\mathcal{R}_S, \mathcal{E}_S)$ such that
- For all non-ground $L \in C^+$ there exists in $\mathcal{R}_S \cup \mathcal{E}_S$ a rewrite rule $M \rightarrow L$ for some literal $M \in C^-$
- $\mathcal{E}_S$ for permutative rules: e.g. $\mathit{differ}(x, y) \rightarrow \mathit{differ}(y, x)$
- **Lemma:**
  - $\rightarrow_{\mathcal{R}_S}$ terminating and $\mathcal{E}_S = \emptyset$: S strictly positively restrained
  - $\leftrightarrow^*_{\mathcal{E}} \circ \rightarrow_{\mathcal{R}} \circ \leftrightarrow^*_{\mathcal{E}}$ terminating, $\mathcal{V}ar(t) = \mathcal{V}ar(u)$ for all $t \rightarrow u$ in $\mathcal{E}_S$, and $\leftrightarrow^*_{\mathcal{E}}$ has finite equivalence classes, S is positively restrained
- Apply a termination tool such as $\mathsf{T_TT_2}$ or AProVE

Experimental results

- Source of clause sets: Geoff Sutcliffe's TPTP 7.2.0
- Problems in the FOF category without $\simeq$, reduced to CNF: 5,001 benchmarks
- Script StoR to generate $\mathcal{R}_S$ and $\mathcal{E}_S$ from clause set S
- Termination tool: $\mathsf{T_TT_2}$
- Either StoR or $\mathsf{T_TT_2}$ timed out on 1,539 inputs
- Out of the remaining 3,462 problems $\mathsf{T_TT_2}$ found 349 restrained, 43 of which are in no other decidable class

The Koala SGGS-based prototype theorem prover

- Written by Sarah Winkler
- Imports code for basic data structures, term indexing, and type inference from Konstantin Korovin's iProver
- Stores selected literals in a discrimination tree for unification
- Implements a fair search plan
- Recognizes stratified problems by checking acyclicity
- Picks $\mathcal{I}^-$ or $\mathcal{I}^+$ based on whether the input is positively or negatively ground-preserving

Experimental results with Koala

- Time-out: 300 sec of wall-clock time
- 349 restrained problems: 50 satisfiable and 283 unsatisfiable
- 351 PVD problems: 76 satisfiable and 232 unsatisfiable
- 1,246 stratified problems: 277 satisfiable and 643 unsatisfiable

Negative results with sign-based semantic guidance

SGGS with $\mathcal{I}^-$ or $\mathcal{I}^+$ does not decide the following fragments that admit (ordered, not hyper) resolution-based decision procedures:

- Ackermann ($\exists^*\forall\exists^*\varphi$) [Joyner: JACM 1976]
- Monadic (no functions, unary predicates) [Joyner: JACM 1976]
- FO^2 (no functions, unary predicates)

  [Scott: JSL 1962] [Grädel, Kolaitis, Vardi: BSL 1997] [Joyner: JACM 1976]
- Guarded (no functions, $\forall\bar{y}.(R(\bar{x},\bar{y}) \supset \psi[\bar{x},\bar{y}])$, $\exists\bar{y}.(R(\bar{x},\bar{y}) \wedge \psi[\bar{x},\bar{y}])$) [de Nivelle, de Rijke: JSC 2003]

Current work on SGGS decision procedures

- Relationship between SGGS and hyperresolution:
  - If clauses are ground-preserving, SGGS halts whenever hyperresolution does
  - SGGS decides the bounded depth increase (BDI) fragment
- More new decidable fragments: SGGS decides the
  - Sort-restrained fragment (restrained on cyclic sorts)
  - Sort-refined PVD fragment (PVD on cyclic sorts)
  - Controlled Horn fragment (not ground-preserving): by the second approach (finitely many SGGS-extensions)
- Modularity of termination
- Complexity of SGGS via derivation length

Future work

- More work on strategies and inner algorithms for SGGS
- Further development of the Koala prover
- Extension to equality
  - SGGS(superposition)
  - CDSAT(SGGS)
- Initial interpretations not based on sign

References

- Semantically-guided goal-sensitive reasoning: decision procedures and the Koala prover. In preparation (with Sarah Winkler).
- SGGS decision procedures. Proc. 10th IJCAR, Springer, LNAI 12166:356–374, 2020 (with Sarah Winkler).
- Semantically-guided goal-sensitive reasoning: inference system and completeness. *Journal of Automated Reasoning*, 59(2):165–218, 2017 (with David A. Plaisted).
- Semantically-guided goal-sensitive reasoning: model representation. *Journal of Automated Reasoning* 56(2):113–141, 2016. (with David A. Plaisted).
- SGGS theorem proving: an exposition. Proc. 4th PAAR Workshop, EPiC 31:25-38, 2015 (with David A. Plaisted).

Thanks

Thank you!