On SGGS and Horn Clauses[1]

Maria Paola Bonacina

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

8th Workshop on Practical Aspects of Automated Reasoning (PAAR)
satellite of the 11th IJCAR held at the 8th FLoC, Haifa, Israel

11 August 2022

[1] Joint work with Sarah Winkler

What is SGGS

- A satisfiability procedure that lifts CDCL to FOL: first-order conflict-driven reasoning
- Refutationally complete: semidecision procedure for validity, theorem-proving method
- Model-complete in the limit: model-building method

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

- Decision procedure for Datalog, EPR, the stratified fragment, and other fragments of FOL
- Implemented in the Koala prover

[MPB and Sarah Winkler: IJCAR 2020, JAR submitted]

SGGS in a nutshell

- Given input clause set S and initial fixed Herbrand interpretation $\mathcal{I}$
- If $\mathcal{I} \models S$ donothing, else search for a model of S
- Build trail Γ that represents a candidate model $\mathcal{I}[\Gamma]$
- Γ: sequence of (possibly constrained) clauses $A \rhd C[L]$
  - L is the selected literal: in the candidate model $\mathcal{I}[\Gamma]$
  - A is a constraint
    SGGS-constraints are a kind of Herbrand constraints $(x \not\equiv y,\ top(x) \neq f)$
- Derivation: $\Gamma_0 \vdash \Gamma_1 \vdash \ldots \Gamma_j \vdash \Gamma_{j+1} \vdash \ldots$
- S unsat: $\bot \in \Gamma_k$ for some k
- S sat: $\mathcal{I}[\Gamma_\infty] \models S$

About the SGGS trail

- Every literal in Γ must be $\mathcal{I}$-true ($\mathcal{I} \models L$) or $\mathcal{I}$-false ($\mathcal{I} \models \neg L$)
- $\mathcal{J} \models \neg L$: literal L is uniformly false in interpretation $\mathcal{J}$
- $\mathcal{I}$-all-true clause: all its literals are $\mathcal{I}$-true
- $\mathcal{I}$-all-false clause: all its literals are $\mathcal{I}$-false
- If a clause in Γ has $\mathcal{I}$-false literals, one must be selected
- An $\mathcal{I}$-true literal is selected only in an $\mathcal{I}$-all-true clause

How does an SGGS-trail represent a model?

- A clause C represents all its ground instances: $Gr(C)$
- A constrained clause $A \triangleright C[L]$ represents all the ground instances of C that satisfy A: $Gr(A \triangleright C[L])$
- Partial model $\mathcal{I}^p(\Gamma)$:
  - Read from Γ left to right
  - Each clause $A \triangleright C[L]$ adds those elements $L\sigma$ of $Gr(A \triangleright L)$ s.t.
  - $C\sigma$ not satisfied and $\neg L\sigma$ not already in
- Disjoint prefix $dp(\Gamma)$: longest prefix where every selected literal contributes to $\mathcal{I}[\Gamma]$ all its ground instances (no intersection of selected literals)
- Model $\mathcal{I}[\Gamma]$: consult $\mathcal{I}$ for ground literals undefined in $\mathcal{I}^p(\Gamma)$

How SGGS makes progress

- Suppose $\bot \notin \Gamma$ and $\mathcal{I}[\Gamma] \not\models S$
- If $\Gamma = dp(\Gamma)$: as $\mathcal{I}[\Gamma] \not\models C$ for some clause $C \in S$ extend Γ hence $\mathcal{I}[\Gamma]$ (extend)
- If $\Gamma \neq dp(\Gamma)$: expose intersection (s-split, d-split) and remove it (delete or resolve) or solve conflict (resolve, l-split, move)

The model-search rules of SGGS

- sat : $\Gamma \rightsquigarrow$ satisfiable if $\mathcal{I}[\Gamma] \models S$
- extend : $\Gamma \rightsquigarrow \Gamma, A \rhd E[L]$ (extension clause)
  - $\mathcal{I}[\Gamma] \not\models C$
  - $C' \in Gr(C)$: $\mathcal{I}[\Gamma] \not\models C'$
  - E: instance of C and C' instance of E
- delete : $\Gamma \rightsquigarrow \Gamma'$
  where Γ' is Γ with all disposable clauses removed
  C is disposable in trail $\Gamma, C, \ldots$ if $\mathcal{I}^p(\Gamma) \models C$
- Model-based, dynamic notion of redundancy

The model-search rules: SGGS-splitting

- Splits trail clause $C[L]$ into partition $C_1[L_1], \ldots, C_n[L_n]$
  $Gr(C) = \bigcup_{i=1}^{n} Gr(C_i)$ but the L_i's are pairwise disjoint
- Splits $C[L]$ to get rid of intersection btw L and M selected in another trail clause $D[M]$
- One of the L_i's contains the intersection
- s-split: $\ldots D[M] \ldots C[L] \ldots \;\rightsquigarrow\; \ldots D[M] \ldots \mathit{split}(C, D) \ldots$
  L and M have same sign
- d-split: $\ldots D[M] \ldots C[L] \ldots \;\rightsquigarrow\; \ldots D[M] \ldots \mathit{split}(C, D) \ldots$
  L and M have opposite sign

Example: a set of definite clauses

1. $P(f(a,x))$
2. $P(g(b,x))$
3. $\neg P(f(y,a)) \vee P(g(y,a))$
4. $\neg P(g(z,b)) \vee P(f(z,b))$

- If $\mathcal{I} = \mathcal{I}^+$: donothing
- If $\mathcal{I} = \mathcal{I}^-$: SGGS builds the least Herbrand model

Example: the SGGS-derivation with $\mathcal{I} = \mathcal{I}^-$

- Γ_0: ε (the empty trail)
  $\mathcal{I}[\Gamma_0] = \mathcal{I}^- \not\models P(f(a,x))$ $\qquad$ $\mathcal{I}[\Gamma_0] = \mathcal{I}^- \not\models P(g(b,x))$
- Γ_1: $[P(f(a,x))]$, $[P(g(b,x))]$
  (SGGS-extension adds the $\mathcal{I}^-$-all-false (i.e., positive) input clauses)
- $\mathcal{I}[\Gamma_1] \not\models \neg P(f(y,a)) \vee P(g(y,a))$
  Γ_2: $[P(f(a,x))]$, $[P(g(b,x))]$, $\neg P(f(a,a)) \vee [P(g(a,a))]$
  (SGGS-extension with mgu $\{y \leftarrow a,\ x \leftarrow a\}$)
- $\mathcal{I}[\Gamma_2] \not\models \neg P(g(z,b)) \vee P(f(z,b))$
  Γ_3: $[P(f(a,x))]$, $[P(g(b,x))]$, $\neg P(f(a,a)) \vee [P(g(a,a))]$,
  $\neg P(g(b,b)) \vee [P(f(b,b))]$
  (SGGS-extension with mgu $\{z \leftarrow b,\ x \leftarrow b\}$)
- $\mathcal{I}^p(\Gamma_3)$ is the least Herbrand model

The least Herbrand model of a set of definite clauses

- S: set of definite clauses
- $\mathcal{A}$: its Herbrand base
- $\mathcal{P}(\mathcal{A})$: all Herbrand interpretations as sets of atom
- $\langle \mathcal{P}(\mathcal{A}), \subseteq, \bigcap, \bigcup, \emptyset, \mathcal{A} \rangle$: complete lattice
- Least Herbrand model:
  - The intersection of all Herbrand models of S or
  - The least fixpoint of functional $T_S \colon \mathcal{P}(\mathcal{A}) \rightarrow \mathcal{P}(\mathcal{A})$:
    $L \in T_S(J)$ iff
    $L = P\sigma$ and $\{Q_1\sigma \ldots Q_m\sigma\} \subseteq J$ for some clause
    $P \vee \neg Q_1 \vee \ldots \vee \neg Q_m$ $(m \geq 0)$ and ground substitution σ
  - $lfp(T_S) = \bigcup_{k \geq 0} T_S^k(\emptyset)$

SGGS as a forward-reasoning procedure: definite clauses

- S: set of definite clauses and $\mathcal{I} = \mathcal{I}^-$
- I^- corresponds to the bottom $\emptyset$ of lattice $\mathcal{P}(\mathcal{A})$
- The first extension puts on Γ all the positive units
- If the addition of positive literals to $\mathcal{I}^p(\Gamma)$ falsifies all the negative literals in instances of mixed clauses, SGGS-extensions with mixed clauses follow
- All selected literals are positive (no choice as every clause has exactly one)
- No conflict arises: $\forall j,\ j \geq 0,\ \mathcal{I}^p(\Gamma_j) \subseteq \mathcal{I}^p(\Gamma_{j+1})$
- **Theorem**: for all fair SGGS-derivations $\mathcal{I}^p(\Gamma_\infty) = \mathit{lfp}(T_S)$

First-order clausal propagation

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

- Conflict clause: for all i, $1 \leq i \leq k$, $\mathcal{I}[\Gamma] \models \neg L_i$
- Implied literal and justification:
  for all i, $1 \leq i \neq j \leq k$, $\mathcal{I}[\Gamma] \models \neg L_i$ and $\mathcal{I}[\Gamma] \models L_j$
- All justifications are in the disjoint prefix
- $\mathcal{I}$-all-true clause: either conflict clause or justification

The SGGS analogue of a 2-watched literal scheme

- An assignment mechanism built into the rules
- The dependencies among literals that determine the propagations are stored with the clauses
- $\mathcal{I}$-true literal L in C_i made uniformly false in $\mathcal{I}[\Gamma]$ by the selection of $\mathcal{I}$-false literal M in C_j $(j < i)$:
  L assigned to C_j
- Non-selected $\mathcal{I}$-true literals must be assigned
- Selected $\mathcal{I}$-true literals must be assigned if possible
- If assigned, a selected $\mathcal{I}$-true literal is assigned rightmost

The conflict-solving rules of SGGS

- unsat: $\Gamma \ \leadsto\ $ unsatisfiable if $\bot \in \Gamma$
- resolve: $\dots D[M] \dots C[L]\ \Gamma \ \leadsto\ D[M] \dots Res(C, D) \dots \Gamma'$
  where $D[M]$ is $\mathcal{I}$-all-true and in $dp(\Gamma)$, L is $\mathcal{I}$-false, $L = \neg M\vartheta$ for some substitution ϑ, Γ' is Γ with all clauses with literals assigned to C removed
- $C[L] \in dp(\Gamma)$, $D[M]$ is $\mathcal{I}$-all-true, and M is assigned to $C[L]$:
  - move: $\dots C[L] \dots D[M] \dots \ \leadsto\ \dots D[M]\ C[L] \dots$
    if $\neg Gr(B \rhd M) = Gr(A \rhd L, \Gamma)$
  - l-split: $\dots C[L] \dots D[M] \dots \ \leadsto\ \dots split(C, D) \dots D[M] \dots$
    if $\neg Gr(B \rhd M) \subset Gr(A \rhd L, \Gamma)$

A Horn example

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

A Horn example II

- S 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 $\neg P(a)$ is assigned to $[P(a)]$ and $\neg Q(f(y))$ to $[Q(f(y))]$
- Conflict: $\neg P(a) \vee [\neg Q(f(y))]$ is an $\mathcal{I}^-$-all-true conflict clause (all its literals are assigned)

Horn example III

- S contains $\{\ P(a),\ \neg P(x) \vee Q(f(y)),\ \neg P(x) \vee \neg Q(z)\ \}$
- $\Gamma_3 = [P(a)],\ \neg P(a) \vee [Q(f(y))],\ \neg P(a) \vee [\neg Q(f(y))]$
- $\Gamma_4 = [P(a)],\ \neg P(a) \vee [\neg Q(f(y))],\ \neg P(a) \vee [Q(f(y))]$
  by SGGS-move: $\mathcal{I}[\Gamma_4] \models \neg Q(f(y))$
  Conflict: $\neg P(a) \vee [Q(f(y))]$ is a conflict clause
- $\Gamma_5 = [P(a)],\ \neg P(a) \vee [\neg Q(f(y))],\ [\neg P(a)]$ by SGGS-resolution:
  the SGGS-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

SGGS as a forward-reasoning procedure: Horn clauses

- S: set of Horn clauses and $\mathcal{I} = \mathcal{I}^-$
- $S_D \subset S$: definite clauses
- **Theorem**: for all fair SGGS-derivations, if an SGGS-extension adds to trail Γ an $\mathcal{I}^-$-all-true conflict clause C, the derivation is a refutation.
  Idea: C is in conflict with $\mathcal{I}^p(\Gamma)$ hence with a subset of $lfp(T_{S_D})$ hence with all models.
- **Theorem**: SGGS halts iff positive hyperresolution with subsumption halts.

SGGS as a backward-reasoning procedure: Horn clauses

- S: set of Horn clauses and $\mathcal{I} = \mathcal{I}^+$
- I^+ corresponds to the top $\mathcal{A}$ of lattice $\mathcal{P}(\mathcal{A})$
- $\mathcal{I}^+$ and hence SGGS is goal-sensitive
- The first extension puts on Γ all the negative clauses
- If the addition of negative literals to $\mathcal{I}^p(\Gamma)$ falsifies the positive literals in instances of mixed clauses, SGGS-extensions with mixed clauses and negative selected literals follow
- Unless a model of S is found, $\mathcal{I}^+$-all-true conflict clauses arise

Experimental results with Koala on Horn problems

- Horn problems without interpreted symbols from TPTP 7.4.0
- $\mathcal{I}^-$: 58% success rate, $\mathcal{I}^+$: 51% success rate

# sets	Koala (I^-) SAT	Koala (I^-) UNS	Koala (I^+) SAT	Koala (I^+) UNS	E 2.4 SAT	E 2.4 UNS	Vampire 4.4 SAT	Vampire 4.4 UNS	iProver 3.5 SAT	iProver 3.5 UNS
1,220	**131**	581	66	467	43	889	79	969	106	**970**

- Koala is best on satisfiable problems
- iProver is best on unsatisfiable problems

Experimental results with Koala on FOL problems

FOL problems without interpreted symbols from TPTP 7.4.0

problem class	SAT	UNS	#steps	#ext	#confl	#gen	#del	max $\|\Gamma\|$	avg time
ground	11	68	345	117	141	245	99	8	0.74
EPR	220	538	496	250	154	399	183	106	20.41
stratified	271	667	402	204	123	323	147	89	16.27
monadic	57	223	120	43	46	85	32	9	0.32
FO^2	213	371	143	75	40	113	35	46	6.30
Ackermann	14	79	295	100	120	209	84	7	0.63
guarded	124	216	506	210	187	388	182	27	7.22
PVD	74	230	553	228	206	425	201	6	7.50
sortRefinedPVD	274	699	389	198	119	313	142	87	15.74
restrained	65	313	129	53	46	96	41	19	1.32
sortRestrained	290	772	371	189	114	299	136	84	14.91
other	110	288	67	48	8	56	20	46	6.73
all	481	1,153	270	143	77	219	96	74	12.79

Comparison on unsatisfiable FOL problems

problem class	# sets	Koala	E	Vampire	iProver	CVC5	-fm	Darwin	-fm
ground	71	68	70	**71**	**71**	**71**	**71**	**71**	70
EPR	790	538	561	756	**774**	628	685	750	595
stratified	933	667	698	900	**918**	741	823	894	618
monadic	620	223	408	560	558	343	363	**590**	195
FO^2	575	372	403	518	**531**	406	492	512	283
Ackermann	84	79	83	**84**	**84**	78	**84**	**84**	73
guarded	403	216	241	385	**387**	320	347	384	258
PVD	261	230	226	**251**	**251**	219	242	248	213
sortRefinedPVD	969	699	729	932	**953**	771	855	929	622
restrained	338	313	317	**329**	328	316	310	325	216
sortRestrained	1,045	772	796	1,007	**1,029**	837	916	1,002	624
others	131	288	585	815	**870**	535	664	768	131
all	769	1,153	1,675	2,189	**2,279**	1,462	1,733	2,164	769

Koala still behind most systems, except for Darwin -fm

Comparison on satisfiable FOL problems

problem class	# sets	Koala	E	Vampire	iProver	CVC5	-fm	Darwin	-fm
ground	11	**11**	**11**	**11**	**11**	**11**	**11**	**11**	**11**
EPR	267	220	118	211	**264**	15	251	263	246
stratified	324	271	144	260	**320**	15	306	319	300
monadic	122	57	56	87	100	14	98	84	**108**
FO^2	349	213	145	240	**288**	13	271	244	287
Ackermann	18	14	**18**	**18**	**18**	13	**18**	14	**18**
guarded	164	124	85	140	**162**	15	150	161	145
PVD	84	74	44	60	**82**	13	80	81	76
sortRefinedPVD	330	274	146	262	**324**	15	311	323	303
restrained	72	65	57	66	**68**	13	67	64	65
sortRestrained	348	290	154	278	**342**	15	327	337	319
others	199	110	52	78	178	0	**200**	146	199
all	713	481	288	456	681	24	676	586	**713**

Koala solves more problems than E, CVC5, and Vampire in most classes, but behind iProver, Darwin, and CVC5 -fm

Discussion

- Equational reasoning:
  - From CDCL(T)+superposition
    [MPB, Lynch, De Moura: CADE 2009, JAR 2011]
  - To SGGS+superposition
- Conflict-driven reasoning: from propositional to first-order
  - ATP: from hyperlinking, ... Inst-Gen to SGGS
  - SMT: from CDCL($\mathcal{T}$) to CDSAT
    [MPB, Graham-Lengrand, Shankar: CADE 2017, CPP 2018, JAR 2020, JAR 2022]
  - The engineering of efficient first-order conflict-driven reasoning has yet to begin